

**DISEÑO DE UN PROGRAMA DE CAPACITACIÓN EN EL SOFTWARE
INCONTROL E IMPLEMENTACIÓN DE UNA APLICACIÓN EN TIEMPO REAL**

**DEICY MILENA GUERRERO ALARCÓN
REMBERTO CARLOS MORENO HERAZO**

**CORPORACIÓN UNIVERSITARIA TECNOLÓGICA DE BOLÍVAR
FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA
CARTAGENA DE INDIAS**

2003

**DISEÑO DE UN PROGRAMA DE CAPACITACIÓN EN EL SOFTWARE
INCONTROL E IMPLEMENTACIÓN DE UNA APLICACIÓN EN TIEMPO REAL**

**GUERRERO A, DEICY MILENA
MORENO H, REMBERTO CARLOS**

Trabajo de grado presentado para optar al título de Ingeniero Electrónico

**Director
DAVID SENIOR
Ingeniero Electrónico**

**CORPORACIÓN UNIVERSITARIA TECNOLÓGICA. DE BOLÍVAR
FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA
CARTAGENA**

2003

Nota de aceptación -----

Firma de presidente del jurado

Firma del Jurado

Firma del jurado

Cartagena, 21 de Mayo de 2003

CONTENIDO

	Pág.
INTRODUCCIÓN	
1. MARCO TEÓRICO	3
1.1 RESEÑA HISTÓRICA	3
1.1.1 Aplicaciones con Wonderware	4
1.1.2 Factorysuite 2000 (FS200)	5
1.1.3 Colsein Ltda.	6
1.2 INCONTROL	8
1.2.1 Características de InControl	9
1.2.2 Lenguajes de Programación	10
1.2.3 Factory Object Editor (FOE)	13
1.2.4 Niveles de seguridad	15
1.2.5 Requerimientos del sistema	15
1.2.6 Instalación	17
1.2.7 Ambiente InControl	18
1.2.7.1 Ventana de Desarrollo	18
1.2.7.2 Ventana de Ejecución	20
1.2.8 Ejecución de InControl	22
1.3 INTOUCH	24
1.3.1 Características del InTouch	25

1.3.2 Ventanas en el ambiente InTouch	26
1.3.3 Términos comunes	28
1.4 HERRAMIENTAS QUE SOPORTAN EL TUTORIAL DE MANEJO DEL SOFTWARE INCONTROL	31
1.4.1 Lenguajes y tecnologías	32
1.4.2 Herramientas de dibujo	34
1.4.3 Herramientas para animaciones	37
1.5 PLC SIMATIC S7-300 cpu314c-2DP	40
1.5.1 Características	40
1.5.2 Funcionalidades	41
1.6 REDES DE COMUNICACIONES INDUSTRIALES	42
1.6.1 FIELDBUS	45
1.6.2 Protocolo Profibus	46
1.6.3 Sistema de transmisión Profibus	50
1.7 TARJETA SMS CIF30-DPM	53
1.7.1 Características	54
2. ANÁLISIS	56
2.1 CONSIDERACIONES Y REQUERIMIENTOS DEL CURSO DE CAPACITACIÓN EN EL SOFTWARE INCONTROL	56
2.1.1 Consideraciones y requerimientos de diseño de las ayudas del curso de capacitación en el software InControl	57
2.1.2 Consideraciones y requerimientos de diseño para el tutorial de manejo del software InControl	58

2.1.2.1 Herramientas a utilizar	59
2.2 CONSIDERACIONES Y REQUERIMIENTOS DE DISEÑO DE LA APLICACIÓN EN EL SOFTWARE INCONTROL	60
2.2.1 Descripción del banco control de nivel existente	60
2.2.1.1 Sistema Control de Nivel	61
2.2.1.2 Estrategias de control	62
2.2.1.3 Descripción del sistema de control	63
2.2.1.4 Modelamiento de la planta	65
2.2.1.5 Instrumentos, equipos y circuitos	70
2.2.1.5.1 Estructura física	70
2.2.1.5.2 Bomba hidráulica	70
2.2.1.5.3 Válvula proporcional servoactuada	51
2.2.1.5.4 Transmisor de nivel	74
2.2.1.5.5 Circuitos	75
2.2.2 Problemas o inconvenientes actuales del banco control de nivel	76
2.2.3 Requerimientos y consideraciones de diseño para la aplicación	77
2.2.4 Requerimientos y consideraciones de diseño para el software de supervisión	80
3. DISEÑO E IMPLEMENTACIÓN	81
3.1 DISEÑO E IMPLEMENTACIÓN DEL CURSO DE CAPACITACIÓN EN EL SOFTWARE INCONTROL	81

3.1.1 Audiencia	82
3.1.2 Objetivos	83
3.1.3 Prerrequisitos	84
3.1.4 Programa del curso	85
3.1.5 Diseño del tutorial de manejo del software InControl	91
3.1.5.1 Secciones de la pantalla principal o de inicio del tutorial	95
3.1.5.2 Contenido del tutorial	98
3.2 DISEÑO DE LA APLICACIÓN DEL SOFTWARE	
INCONTROL EN EL BANCO CONTROL DE NIVEL	103
3.2.1 Sistema de control	105
3.2.1.1 Descripción del nuevo sistema de control	106
3.2.2 Diseño de las estrategias de control (on-off, PID)	
en el software InControl	109
3.2.2.1 Control de nivel en el tanque 1 y sintonización del PID	110
3.2.2.2 Diseño del control PID	116
3.2.2.3 Diseño del control ON-OFF	132
3.2.2.4 Configuración de la tarjeta SMS CIF30-DPM como maestro DP	143
3.2.2.5 Programación del PLC S7-300 como esclavo DP	165
3.2.2.5.1 Definición de las Señales de Entrada y Salida en el PLC	165
3.2.2.5.2 Configuración del PLC como Esclavo	165

3.2.2.5.3 Elaboración del programa en el software Step 7	171
3.2.2.6 Cableado y Equipo Requerido	176
3.2.3 Visualización del sistema en el software InTouch	181
3.2.3.1 Creación y configuración de la aplicación	181
3.2.3.1.1 Creación y configuración de la ventana Banco de Nivel	185
3.2.3.1.2 Creación de los gráficos de la ventana Banco de Nivel	187
3.2.3.1.3 Configuración de tagnames de la ventana Banco de nivel	198
3.2.3.1.4 Creación y configuración de la ventana Tablero de Control	203
3.2.3.1.5 Creación de los gráficos de la ventana Tablero de Control	205
3.2.3.1.6 Configuración de tagnames en la ventana Tablero de Control	209
3.2.3.1.7 Creación y configuración de la ventana Alarma	213
3.2.3.1.8 Creación y configuración de la ventana Válvula Proporcional	214
3.2.3.1.9 Creación de Scripts	215
3.2.3.2 Descripción del manejo del mímico	220
3.2.3.3 Creación de tagnames desde InTouch para leer símbolos InControl	226
3.2.3.4 Ejecución del supervisorío y de la estrategia de control	230

3.3 FUTURO PROYECTO: DISEÑO DE UNA RED PROFIBUS CON LOS BANCOS DEL LABORATORIO DE CONTROL DE LA TECNOLÓGICA DE BOLÍVAR.	238
3.3.1 Diseño y descripción de la red PROFIBUS	239
3.3.2 Cableado y equipo requerido	242
CONCLUSIONES	247
GLOSARIO	251
BIBLIOGRAFÍA	255
ANEXOS	

LISTA DE CUADROS

	Pag
Cuadro 1. Requerimientos del sistema	16
Cuadro 2. Elementos de la ventana de desarrollo	19
Cuadro 3. Elementos de la ventana de ejecución	21
Cuadro 4. Tipos de versiones Profibus	49
Cuadro 5. Especificaciones de la CIF30 – DPM	55
Cuadro 6. Segmentos de bus y cable de derivación para el estandar profibus	79
Cuadro 7. Tareas de los iconos de la barra estándar	96
Cuadro 8. Fórmulas para calcular los parámetros del PID según Ziegler – Nichols	114
Cuadro 9. Salidas y entradas del PLC	165
Cuadro 10. Especificaciones de cables y conectores (1)	178
Cuadro 11. Especificaciones de cables y conectores (2)	243
Cuadro 12. Lista de símbolos y tagnames de la ventana Banco de Nivel	202
Cuadro 13. Lista de símbolos y tagnames de la ventana Banco de Nivel	211
Cuadro 14. Cuadro de tareas de la ventana Banco de Nivel	223

Cuadro 15. Cuadro de tareas de la ventana Tablero de Control	224
Cuadro 16. Tagnames de InTouch y Símbolos InControl	
Vinculados	229

LISTA DE TABLAS

	Pág.
Tabla 1. Velocidad de transmisión	50
Tabla 2. Características de la bomba hidráulica	71
Tabla 3. Especificaciones de línea tipo A	179
Tabla 4. Estándar EN 50170	244
Tabla 5. Otras especificaciones Línea Tipo A	244

LISTA DE ANEXOS

Pág

ANEXO A. Manual de usuario del software InControl	
ANEXO B. Curso de entrenamiento en InControl (diapositivas)	
ANEXO C. Manual de manejo del tutorial del software InControl	
ANEXO D. Guías de laboratorio	
ANEXO E. Manual de usuario del PLC Simatic S7-300	
ANEXO F. Manual de hardware de la tarjeta SMS-30	
ANEXO G. Buses de campo aplicados al control de procesos industriales	

LISTA DE FIGURAS

	Pag.
Figura 1. Ejemplo de una línea de programa en lenguaje RLL	11
Figura 2. Ejemplo de un programa en lenguaje SFC	12
Figura 3. Ejemplo de un programa en lenguaje STL	13
Figura 4. Pantalla de instalación del InControl	17
Figura 5. Ventana de Desarrollo	19
Figura 6. Ventana de Ejecución	21
Figura 7. Iniciar InControl	23
Figura 8. InControl Project Manager	24
Figura 9. Application Manager de InTouch	29
Figura 10. Window Maker de InTouch	27
Figura 11. Window Viewer de InTouch	28
Figura 12. PLC SIMATIC S7- 300	40
Figura 13. Conexión física industrial	43
Figura 14. Conversión de señales analógicas a digitales	44
Figura 15. Comunicación digital bidireccional	45
Figura 16. Comunicación ProfiBus y sus diferentes interfaces de usuario	47
Figura 17. Tarjeta SMS	55

Figura 18. Diagrama de bloques del sistema	61
Figura 19. Diagrama esquemático del sistema	64
Figura 20. Entrada y salida de la planta	65
Figura 21. Diagrama de control para el sistema de primer orden	67
Figura 22. Diagrama de control para el sistema de segundo orden	69
Figura 23. Curva de desempeño de la bomba hidráulica	71
Figura 24. Válvula de control proporcional	72
Figura 25. Señal de acción de la válvula proporcional	73
Figura 26 . Circuito filtro pasabajos	75
Figura 27. Circuito amplificador	76
Figura 28. Formato general de las páginas del tutorial	94
Figura 29. Pantalla de inicio del Tutorial	95
Figura 30. Iconos de Acceso	97
Figura 31. Diagrama de conexión física del sistema	104
Figura 32. Diagrama de bloques del sistema	105
Figura 33. Diagrama esquemático del sistema de control	107
Figura 34. Partes del PLC S7-300	108
Figura 35. Comunicación Profibus DP	109
Figura 36. Sistema de control de lazo cerrado	110
Figura 37. Respuesta escalón unitario de una planta	112
Figura 38. Curva de respuesta en forma de S	113
Figura 39. Señales y parámetros del controlador PID	113

Figura 40. Curva de reacción para determinación de constantes	114
Figura 41. Respuesta del sistema de segundo orden con Control PI	115
Figura 42. Creación de proyectos InControl	117
Figura 43. InControl Project Manager	118
Figura 44. Cuadro New	119
Figura 45. Cuadro Select Factory Object	120
Figura 46. Propiedades de Wonderware PID Control	121
Figura 47. Propiedades de Wonderware PID Control etiqueta variable de proceso	122
Figura 48. Propiedades de Wonderware PID Control etiqueta Setpoint	123
Figura 49. Propiedades de Wonderware PID Control etiqueta Output	124
Figura 50. Propiedades de Wonderware PID Control etiqueta Tuning	125
Figura 51. Ventana del FOE PID	126
Figura 52. Cuadro new	127
Figura 53. Symbol Manager	128
Figura 54. Programa de encendido de la bomba	128
Figura 55. Selección del I/O Driver Profibus (Synergetic) en InControl	129
Figura 56. Cuadro de dialogo Profibus (Synergetic/Hilscher)	130

Figura 57. Sistema de control de lazo cerrado	131
Figura 58. Creación de proyectos InControl	133
Figura 59. Presentación del proyecto ya creado	134
Figura 60. Cuadro New	135
Figura 61. Cuadro de diálogo para guardar el controlador On/OFF	136
Figura 62. Symbol Manager del Programa NivelONOFF	137
Figura 63. Presentación del programa NivelONOFF	138
Figura 64. Programa que fija la brecha diferencial o Banda muerta del controlador ON-OFF	139
Figura 65. Selección del I/O Driver Profibus (Synergetic) en InControl	140
Figura 66. Cuadro de diálogo Profibus (Synergetic/Hilscher)	141
Figura 67. Cuadro de diálogo CIF Device Driver Setup	146
Figura 68. SyCon Sistem Configurator	147
Figura 69. Pantalla de inicio del software SyCon System Configurator	148
Figura 70. Cuadro Select FieldBus	149
Figura 71. Cuadro de diálogo Insert Master	150
Figura 72. Cuadro de diálogo Device Assignment CIF Device Driver	150
Figura 73. Cuadro de diálogo CIF DEVICE DRIVER	152
Figura 74. Cuadro de diálogo Insert Slave	153
Figura 75. Vista del proyecto PROFIBUS	154

Figura 76. Selección del Profibus (Synergetic) I/O Driver en InControl	156
Figura 77. Cuadro de diálogo Profibus (Synergetic/Hilscher)	157
Figura 78. Cuadro de dialogo PROFIBUS Module Configuration	159
Figura 79. Cuadro de dialogo PROFIBUS Digital Port Configuration	161
Figura 80. Cuadro de dialogo PROFIBUS Analog/Byte Port Configuration	162
Figura 81. Cuadro de diálogo PROFIBUS Analog/Byte Port Configuration	164
Figura 82. Pantalla de inicio del proyecto creado	166
Figura 83. Ventana de inicio del software HW Config	167
Figura 84. Ventana de presentación de hardware para nuestro proyecto	168
Figura 85. Primer paso para la configuración del modo operativo	169
Figura 86. Segundo paso para la configuración del modo operativo	169
Figura 87. Tercer paso para la configuración del modo operativo	170
Figura 88. Cuarto paso para la configuración del modo operativo	170
Figura 89. Tabla de símbolos	171
Figura 90. Programa KOP	172
Figura 91. Normalización de la señal del sensor	173
Figura 92. Programa del bloque FC1	174

Figura 93. Continuación del programa del bloque FC1	175
Figura 94. Bloque de datos	176
Figura 95. Diagrama de conexión del sistema de red Profibus	180
Figura 98. Ejecución del InTouch	182
Figura 99. Ventana Application Manager	182
Figura 100. Creación de una nueva aplicación	183
Figura 101. Ingreso de la dirección de la aplicación	184
Figura 102 Nombre y descripción de la aplicación	184
Figura 103. Presentación de la aplicación en la lista de proyectos de InTouch	185
Figura 104. Ambiente de trabajo del InTouch (Window Maker)	186
Figura 105. Configuración de las propiedades de la ventana principal “Banco de Nivel”	186
Figura 106. Barras de herramientas para la creación de gráficos	187
Figura 107. Estructura del banco	188
Figura 108 . Elementos gráficos	189
Figura 109. Conversión de la figura del computador a celda	190
Figura 110. Conversión de la pantalla del computador a símbolo	191
Figura 111. Ejemplos de figuras, objetos y símbolos	191
Figura 112. Insertar el Mapa de bits sobre la pantalla	192
Figura 113. Seleccionar la imagen a importar	193
Figura 114. Imagen Importada	193
Figura 115. Imágenes importadas tipo Bitmap	194

Figura 116 . Insertar botones empleando wizards	95
Figura117. Configuración de botones	196
Figura 118. Gráficos Wizards	196
Figura 119. Configuración de wizards	197
Figura 120. Presentación final de la ventana Banco de Nivel	198
Figura 121. Animation Links Library	199
Figura 122. Configuración de una etiqueta	200
Figura 123. Selección del tipo de tagname	200
Figura 124. Configuración de una animación de llenado	201
Figura 125. Vincular un texto a un tagname	202
Figura 126. Creación de la ventana Tablero de Control	204
Figura 127. Trazo de bordes de la ventana Tablero de Control	205
Figura 128. Insertar Frames	206
Figura 129. Objetos de control creados	206
Figura 130. Elección de botones de control	207
Figura 131. Iconos para gráficas de tendencias	207
Figura 132. Insertar gráficas de tendencias	208
Figura 133. Elección de los botones de control	209
Figura 134. Configuración de las gráficas de tendencias	210
Figura 135. Presentación final de la pantalla Tablero de Control	211
Figura 136. Configuración de la ventana Alarma	213
Figura 137. Configuración de la ventana Válvula proporcional	214
Figura 138. Ingreso a los scripts	215

Figura 139. Ventana de un Script de aplicación	216
Figura 140. Ventana Application Manager	220
Figura 141. Ventana Window Maker	221
Figura 142. Ventana Banco de Nivel	222
Figura 143. Ventana Tablero de Control	224
Figura 144. Ventana Válvula proporcional	225
Figura 145. Ventana Alarma	226
Figura 146. Elección del wizard InControl Runtime Add Tag	227
Figura 147. Elección de las etiquetas a vincular	228
Figura 148. Vinculo de los tagname con los símbolos de InControl	228
Figura 149. Confirmación de conversión de tagname	229
Figura 150. Ruta de Ejecución InTouch	230
Figura 151. Ventana Application Manager	231
Figura 152. Ventana Window Maker de la aplicación	232
Figura 153. Ventana Window Viewer	234
Figura 154. Ventana del proyecto control PID	235
Figura 155. ventana del proyecto control ON-OFF	235
Figura 156. InControl Project Manager	236
Figura 157. Ventana tablero de control activa	237
Figura 158. Número de estaciones – tiempo de bus del ciclo	241
Figura 159. Diagrama de bloques del sistema laboratorio de control	245

RESUMEN

Con este trabajo se busca continuar la línea de proyectos que se iniciaron en la CUTB en el campo del control industrial, concretamente en el aprendizaje de un software para automatización industrial como es la FactorySuite de Wonderware, con el diseño e implementación del curso de capacitación en el software InControl éste junto con el diseño de una aplicación en el banco control de nivel fue el punto de inicio de este plan.

El siguiente paso se ha dado aquí con el curso de capacitación en el software InControl, con lo cual se seguirá la secuencia que permitirá suscitar el interés desde el punto de vista didáctico ya que por medio de este curso se ayudará a los estudiantes a entender las funciones del InControl para crear soluciones de control y aplicaciones para una variedad de procesos de automatización, también para diseñar, programar e implementar con mucha exactitud aplicaciones de control por computador utilizando los lenguajes de programación disponibles en InControl 7.1.

Para un mejor aprovechamiento del desarrollo y diseño de este curso de capacitación su presentación se ha dispuesto de la siguiente manera: la primer parte se muestra una reseña histórica de Wonderware, FactorySuite e InControl, seguido de un pequeño grupo ejemplos de aplicaciones, los conceptos básicos del software InControl y sus características más importantes, tales como sus

funciones InControl, la barra de herramientas Runtime y Standar, drivers I/O, tipos de datos y variables soportadas por InControl, diferentes lenguajes de programación soportados por InControl, además se hace una breve descripción de lo que es el InTouch, sus principales características, sus funciones mas importantes y sus dos ventanas principales (*WindowMaker* y *WindowViewer*), seguido a esto se hace una pequeña descripción de las principales herramientas de diseño del tutorial de manejo en el software InControl, también una descripción muy general de las principales características del PLC S7-300, de la tarjeta CIF30-DPM y por ultimo se hace una reseña de las redes de comunicaciones industriales.

La segunda parte esta compuesta por un análisis detallado de los principales requerimientos y consideraciones acerca del diseño del curso de capacitación compuesto por un tutorial de manejo en formato HTML, un manual de usuario en formato PDF, ayudas audiovisuales consistentes en diapositivas y por ultimo se presenta el análisis del diseño de la aplicación concerniente al sistema de control en el banco Control de Nivel.

La tercera parte se refiere al diseño e implementación del curso de capacitación en el *software* InControl en este se describe las características y las herramientas usadas en el diseño de tutorial de manejo, asi como el diseño de la aplicación en el banco Control de Nivel, describiendose el nuevo sistema de control (teniendo en cuenta las condiciones de comunicación del software) y se explica paso a paso

cómo se realizaran las estrategias de control implementadas (ON-OFF y PID) al igual que el diseño del sistema supervisorío (Monitoreo de variables) en el software InTouch. Además se presentan los pasos necesarios para la ejecución de la estrategia de control y del sistema de visualización en InTouch. Por último se presenta el diseño de un futuro proyecto consistente en una red Profibus con los bancos del laboratorio de control automático de la tecnológica de Bolívar, en este se enuncian los dispositivos necesarios para llevar a cabo la implementación de esta aplicación, así como la información técnica concerniente a dicha aplicación.

La cuarta parte consistente en los anexos está formada por un manual de usuario de InControl en formato PDF, compuesto de un libro que se encuentra en impreso y a la vez contenido en un CD anexo a este libro.

Además en esta parte se presentan las ayudas audiovisuales (Diapositivas) correspondientes a la capacitación del curso en el software InControl, que servirán de guía a la persona que realice dicha capacitación, así mismo se presentan los manuales de hardware de la tarjeta CIF30-DPM y del PLC S7-300 dispositivos utilizados en el diseño de la aplicación, información técnica concerniente a las redes de comunicaciones industriales, un manual de usuario de manejo del tutorial de manejo del InControl que está acompañado de un CD que contiene el tutorial de manejo del software InControl y por último se presentan unas prácticas de laboratorio.

1. MARCO TEÓRICO

1.1 RESEÑA HISTÓRICA

Wonderware ocupa un lugar importante en la Industria en lo que se refiere a la implementación de soluciones de automatización basadas en los sistemas operativos *Microsoft Windows 95, Windows 98 y Windows NT*, al igual que en la *Suite Microsoft* de soluciones para servidores *BackOffice*. Fue pionero en el uso de *Microsoft Windows* en el área de producción de una planta en 1989, con la construcción de un revolucionario software de visualización y control HMI (*Human Machine Interface*) conocido como *InTouch*.

Actualmente *Wonderware* ofrece el software **FactorySuite 2000**, una suite de productos que abarcan a la empresa en su conjunto, desde el área de producción de la fábrica hasta las oficinas ejecutivas, permite aplicaciones de control supervisorio y adquisición de datos, administración de la producción y seguimiento de recursos, administración flexible de procesos por lotes, diagnóstico de maquinaria y productos de conectividad para vincularlos a dispositivos de control y a redes de computación.

1.1.1 Aplicaciones con Wonderware. Los siguientes enunciados son soluciones *Wonderware* en diferentes empresas de varios países.

- El Centro de Investigaciones de la NASA emplea *InTouch* para el monitoreo de túneles de viento.
- Experimentos de la NASA contienen *InTouch* para unir los transbordadores espaciales con la cabina de mando terrestre.
- M. Hiber usa Industrial SQL Server y *Database* para la producción en línea de productos *strand board*.
- Caterpillar usa *InTouch* con *VAX Host* para revisar la línea de producción moderna de tractores.
- Motores Delta usa *InTrack* para proporcionar datos en tiempo real de sus pisos de planta con sistemas SAP ERP.
- PCS – *Nitrogen* usa tecnología *Thin Client* con *Wonderware* para modernizar operaciones de planta y control.

- La suite Wonderware es usada para automatizar la producción de unidades de láminas en 3-D en PCS – *Nitrogen*.
- La Hidroeléctrica Bangor usa *Wonderware* para el control remoto en red de subestaciones de energía.

1.1.2 FactoySuite 2000 (FS2000). Es un sistema HMI integral que incluye un conjunto de componentes de información y automatización. Estos componentes ofrecen funciones indispensables, tales como:

- *InTouch*: proporciona una sola visión integrada de todos los recursos de control e información.
- *InControl*: para control de procesos y maquinaria, basados en Windows NT.
- *Industrial SQL server*: la primera base de datos relacional en tiempo real para el área de producción de la planta.
- *Scout*: una herramienta *Intranet / Internet* para la visualización de datos remotos.
- *Intrack*: para administración de recursos.

- *InBatch*: para administración flexible de procesos por lotes.

Estos componentes tienen una base de datos en común, además trabajan juntos de manera perfectamente integrada, comparten información entre aplicaciones, tienen la misma apariencia y soportan mecanismos de comunicación comunes.

FactorySuite 2000 es uno de las herramientas mas implementadas en la Industria Nacional, es el caso de empresas como: Colgate Palmolive, Aguas de Santa Marta, Bavaria, ECOPETROL y AAA. En el sector de Cartagena se encuentran: Aguas de Cartagena S.A., Propileno del Caribe S.A, Biofilm S.A, y Abonos Colombianos S.A.

Una de estas empresas, Abonos Colombianos S.A, aplica el *FactorySuite 2000* para monitorear el sistema de control de presión de un compresor multiservicio que anteriormente no estaba en funcionamiento debido a las lecturas erróneas que entregaba al antiguo sistema de control y al mal estado de los monitores, pero gracias a FS2000 actualmente poseen un sistema de control mas seguro, al mismo tiempo que disminuyeron la lista de elementos empleados, reduciendo el costo de mantenimiento y la posibilidad de introducir algunas modificaciones a dicho sistema de control.

1.1.3 Colsein Ltda. Es una empresa fundada en 1.987 la cual provee servicios y equipos de automatización que le resuelven a la industria sus necesidades dentro

del marco de la pirámide de automatización Industrial. Su enfoque está orientado al campo de los sensores, actuadores, interfaces de proceso, periféricos de control, controladores de proceso, y un software de supervisión y control de procesos llamado *FactorySuite 2000* de *Wonderware*.

La empresa esta conformada por equipos de ingenieros y técnicos. Entre sus servicios se encuentran:

- Líneas de interfaces y técnicas de conexión, diseñados para ser montados en los tableros eléctricos.
- Equipos periféricos como los indicadores de panel digital, controladores de lazo simple y computadores industriales también se encuentran en su gama de productos.
- Software de supervisión y control que diagnostica, no sólo el estado del sistema, sino el de todos y cada uno de los elementos de campo instalados en su planta.

Todo esto les da la pauta de estar en los más importantes proyectos de automatización que se desarrollan en Colombia.

Dentro de las soluciones Industriales, Colsein ha prestado su servicio con el producto *FactorySuite 2000* en:

- Federación Nacional de Cafeteros.
- ECOPETROL.
- Colgate Palmolive.
- Procter & Gamble Colombia.
- Bavaria S.A.

1.2 INCONTROL

InControl es un sistema de control de arquitectura abierta en tiempo real basado en el sistema operativo *Windows NT de Microsoft*, que permite diseñar, crear, probar y correr programas de aplicación para controlar procesos de manera rápida.

Se pueden crear soluciones propias de automatización en una variedad de lenguajes a base de gráficos y textos. *InControl* además, soporta interfaces directas a una variedad de dispositivos I/O, motores y otros equipos de fábrica, y soporta tanto interfaces tradicionales como las basadas en los estándares actuales.

InControl ofrece una solución de control integrada que sustituye a los sistemas de control propietarios, ofreciendo una arquitectura de control a un costo más bajo con conectividad integrada, una poderosa capacidad de procesamiento y una fácil capacidad de expansión. Al igual que todos los productos *Wonderware*, *InControl* se encuentra perfectamente integrado con *InTouch* y con los componentes de *FactorySuite 2000*, ofreciendo una productividad sin precedentes.

1.2.1 Características de InControl. Gracias a sus características logra aumentar la productividad en los procesos a diferencia de otros sistemas de control tradicionales:

- **Arquitectura abierta:** puede ser utilizado en cualquier plataforma que soporte el sistema operativo *Windows NT* de *Microsoft*, incluyendo estaciones de trabajo industriales "*Flat Panel*", servidores SMP y CPU's de PLC's tipo "*Open Controllers*"
- **Control distribuido:** se conecta a cualquier software de ejecución remoto, descarga programas y estados del monitor, todo desde una estación de desarrollo.

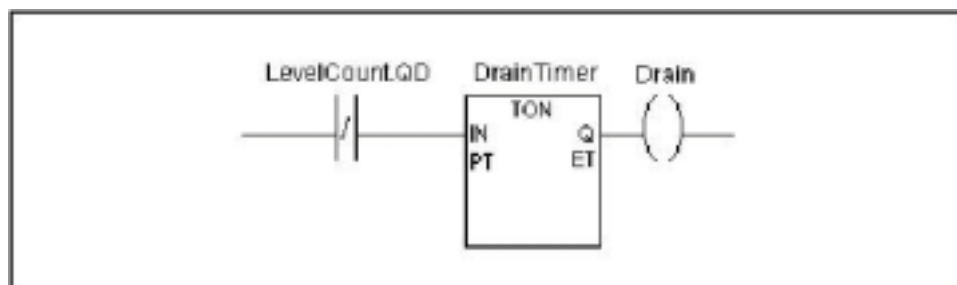
- **Normas Internacionales:** compatible con la especificación de lenguajes de programación internacional IEC-61131-3, OMAC, RS-274D y *Open Device Network*
- **Lenguajes soportados:** Lógica de Escalera por Conmutación (*Relay Ladder Logic*), Organigrama de Funciones Secuenciales(*Sequential Function Chart*) y lenguaje de Texto Estructurado (*Structured Text Language*). El desarrollo de una aplicación se puede realizar combinando los diferentes lenguajes.
- **Interfaces:** soporte para una amplia gama de tarjetas I/O para adquisición y control de datos como: *DeviceNet, PMAC, AB 1771 Remote I/O, PCDIO, GE 90/30, Interbus-S o Profibus DP*.
- **Conexiones:** integración con el software SCADA InTouch®, comunicación con clientes *SuiteLink* y capacidad como servidor

1.2.2 Lenguajes de programación. InControl permite crear soluciones de automatización en la industria con ayuda de diversos lenguajes basados en texto y gráficos. También existe una herramienta que permite usar controles dentro de un proyecto, estos son los FOEs. Los lenguajes que soporta InControl son:

- **Lógica en Escalera por Conmutación (*Relay Ladder Logic*) RLL:** es un programa basado en gráficos que esta compuesto de dos líneas verticales (barras de energía) conectadas por una o mas líneas horizontales (escalones RLL). La barra izquierda representa la fuente de energía y la barra derecha representa la línea de descarga. Los elementos básicos de los programas RLL, contactos y bobinas, son localizados en los escalones y representan los componentes actuales del *hardware* (Interruptores de límite, Bobinas, Solenoides, etc) y la localización interna de los bit en la memoria.

Para mas información acerca de los elementos y el uso del editor de este tipo de programación, refiérase al Anexo A, MANUAL DE USUARIO INCONTROL en el capítulo 8 (Elementos del programa RLL) y el capítulo 7 (Uso del editor RLL) respectivamente.

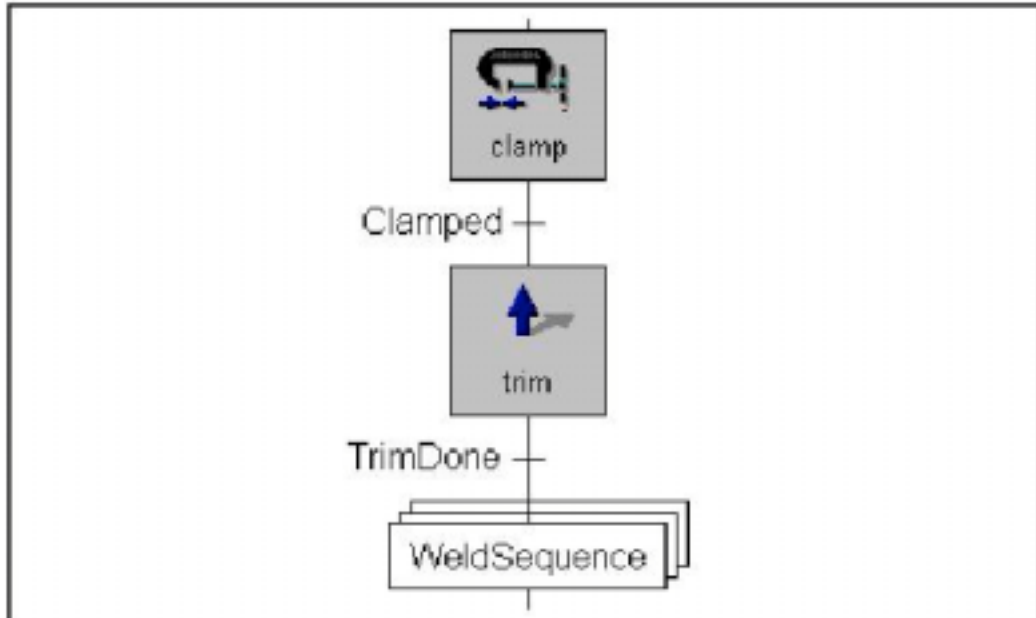
Figura 1. Ejemplo de una línea de programa en lenguaje RLL



- **Organigrama de Funciones Secuenciales (*Sequential Function Chart*) (SFC)** es un programa de aplicación que se representa como una serie de estados secuenciales o Pasos. Un Paso representa una condición en la cual la ejecución del programa sigue un conjunto de reglas definidas por Acciones y Funciones asociadas con un Paso.

Los elementos de programación SFC y el uso del editor SFC se encuentran descritos en los capítulos 8 y 7 del MANUAL DE USUARIO INCONTROL, Anexo A.

Figura 2. Ejemplo de un programa en lenguaje SFC



- **Lenguaje de Texto Estructurado (*Structured Text Language*) (STL)** es un subconjunto del conjunto de instrucciones basadas en texto que siguen la norma IEC-61131. Estas instrucciones están diseñadas para la fácil creación de operaciones tanto lógicas como matemáticas.

La descripción acerca de cómo crear un programa de aplicación usando el Lenguaje de Texto Estructurado, se encuentra en el capítulo 11 del MANUAL DE USUARIO INCONTROL, Anexo A. El tema acerca del el uso del editor esta expuesto en el capítulo 12 del mismo anexo.

Figura 3. Ejemplo de un programa en lenguaje STL

```
      ("Sets PID output to 0 if PID is in manual mode")
      IF Loop1.InManual      THEN
          Loop1.Out := 0 ;
      END IF
      ("Cycle PID Loop between setpoints of 0 and 100")
      IF ABS (Loop1.error) < 0.5 AND Loop1.InAuto THEN
          IF Loop1.sp >= 100 THEN
              Loop1.Sp := 0 ;
          ELSE
              Loop1.Sp := 100 ;
          END IF ;
      END IF ;
```

1.2.3 Factory Object Editor (FOEs). Los Editores de Objetos de Fábrica (FOE) son contenedores *ActiveX*, cuyas características habilita el uso de controles dentro

de un proyecto *InControl*. *InControl* es compatible con las especificaciones del Servidor *ActiveX*.

Un control *ActiveX* debe ser instalado dentro de *InControl* antes que pueda ser configurado y ejecutado. Después de la instalación, es referido como un *Factory Object (FOE)* de *InControl*. Tal como otros programas *InControl*, un FOE puede correr independientemente. También puede llamarse para ejecución desde otro programa.

La instalación del *InControl* incluye algunos FOEs. Los tipos de FOEs son los siguientes:

- **PID *InControl* FOE:** para manejar funciones de lazo PID.
- ***Analog Alarm* FOE:** para monitorear una señal de entrada análoga en condiciones de alarma.
- ***Project Information* FOE:** para enviar proyectos e información de nodos a otras aplicaciones, así como *InTouch*, y así ejecutar varios comandos *runtime* o para eliminar fallas.

En el capítulo 13 del Anexo A ,MANUAL DE USUARIO INCONTROL, se halla una introducción en el editor de Objetos de Fábrica (FOE) de *InControl* y describe cómo usarlo para adicionar controles *ActiveX* a un proyecto *InControl*.

1.2.4 Niveles de seguridad. El acceso al ambiente *InControl* está restringido para asegurar que solo personal autorizado y/o calificado pueda interactuar con los procesos de fábrica. Existen tres niveles de seguridad para el acceso: Administrador, Ingeniero y Operador. Inmediatamente después que se instala *InControl* y se corre el programa, se adquiere el nivel de seguridad de Administrador.

La información acerca de cómo asignar o remover *passwords* para el ambiente de desarrollo y de ejecución *InControl* está contenida en el Capítulo 3 del anexo A, MANUAL DE USUARIO INCONTROL.

1.2.5 Requerimientos del sistema. El software *InControl* está diseñado para correr sobre cualquier computador IBM PC compatible. Antes de instalar el *InControl*, se debe verificar que el sistema reúna los requerimientos especificados en el cuadro 1 de la siguiente página.

Para ejecutar el *Runtime Engine* sobre sistemas *Windows NT*, se debe tener en cuenta lo siguiente:

- Por lo menos 32 MB en RAM (48 MB es lo recomendado) y un mínimo de 20 MB de espacio en disco duro es lo requerido para el *software* de ejecución.

Para ejecutar el *Runtime Engine* sobre sistemas *Windows CE*, se debe tener en cuenta lo siguiente:

- *InControl* soporta aplicaciones específicas del proveedor en sistemas *Windows CE*, basados en procesadores fabricados por Intel (X86, StrongArm), Itachi, y MIPS.

El capítulo 1, Inicio Rápido con InControl, del Anexo A contiene más información acerca de los requerimientos de software y hardware para el InControl.

Cuadro 1. Requerimientos del sistema

Elementos del sistema	Requerimientos Mínimos	Requerimientos recomendados
Procesador	<i>Pentium Pro</i> 200 MHZ	Superior a 200 MHZ
Memoria RAM	32 MB	48 MB y 128 MB para muchos símbolos.
Disco Duro	80 MB	128 MB
Monitor Colores resolución	VGA 256 640x480 pixel	VGA 256 800x600 pixel
Sistema Operativo	Microsoft Windows NT (Versión 4.x, Service Pack 5), Windows 2000.	
Requerimientos de Hardware		
Drive CD ROM <i>Mouse</i> u otro componente compatible con el sistema operativo Windows NT		

1.2.6 Instalación. El *InControl* está distribuido en un disco compacto que hace parte de la *Wonderware FactorySuite*. Cuando se ejecuta el programa de configuración el sistema hace lo siguiente:

- Crea los directorios necesarios en el disco duro, y copia los archivos dentro de los directorios apropiados.
- Establece la carpeta de programas e íconos.
- Modifica el registro del sistema.

Figura 4. Pantalla de instalación del InControl.



Antes de intentar instalar el programa se recomienda lo siguiente: cerrar cualquier programa, particularmente aplicaciones *Wonderware* que estén corriendo actualmente y desinstalar cualquier versión previa de *InControl*. Si se pretende usar *InTouch* con *InControl* sobre un mismo sistema, es recomendable que se instale *InTouch* antes de instalar *InControl*.

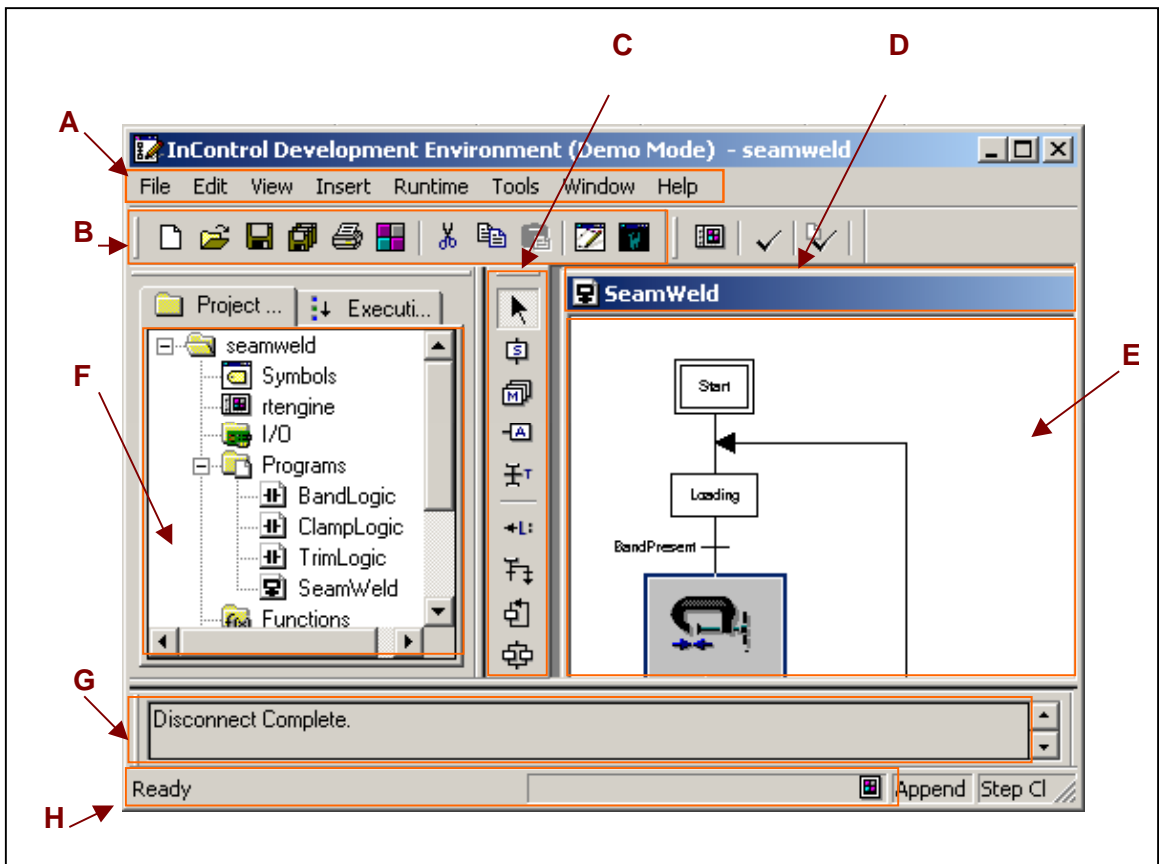
En los temas Línea Directa de Instalación y Desinstalación de InControl en el capítulo 1 del anexo A , MANUAL DE USUARIO INCONTROL, están contenidas las instrucciones para instalar y correr el InControl.

1.2.7 Ambiente InControl. El InControl consta de dos conjuntos de ventanas: La Ventana de Desarrollo, donde se pueden crear programas de aplicación, y la Ventana de Ejecución donde se ejecutan y monitorean los programas creados.

En el capítulo 2, Ambiente InControl, del anexo A está contenida la descripción detallada del ambiente InControl: Uso de las barras de herramientas, opciones de menú, campos de texto, íconos, botones, etc.

1.2.7.1 Ventana de Desarrollo. El esquema típico para la ventana de desarrollo es como se muestra en la Figura 5.

Figura 5. Ventana de Desarrollo



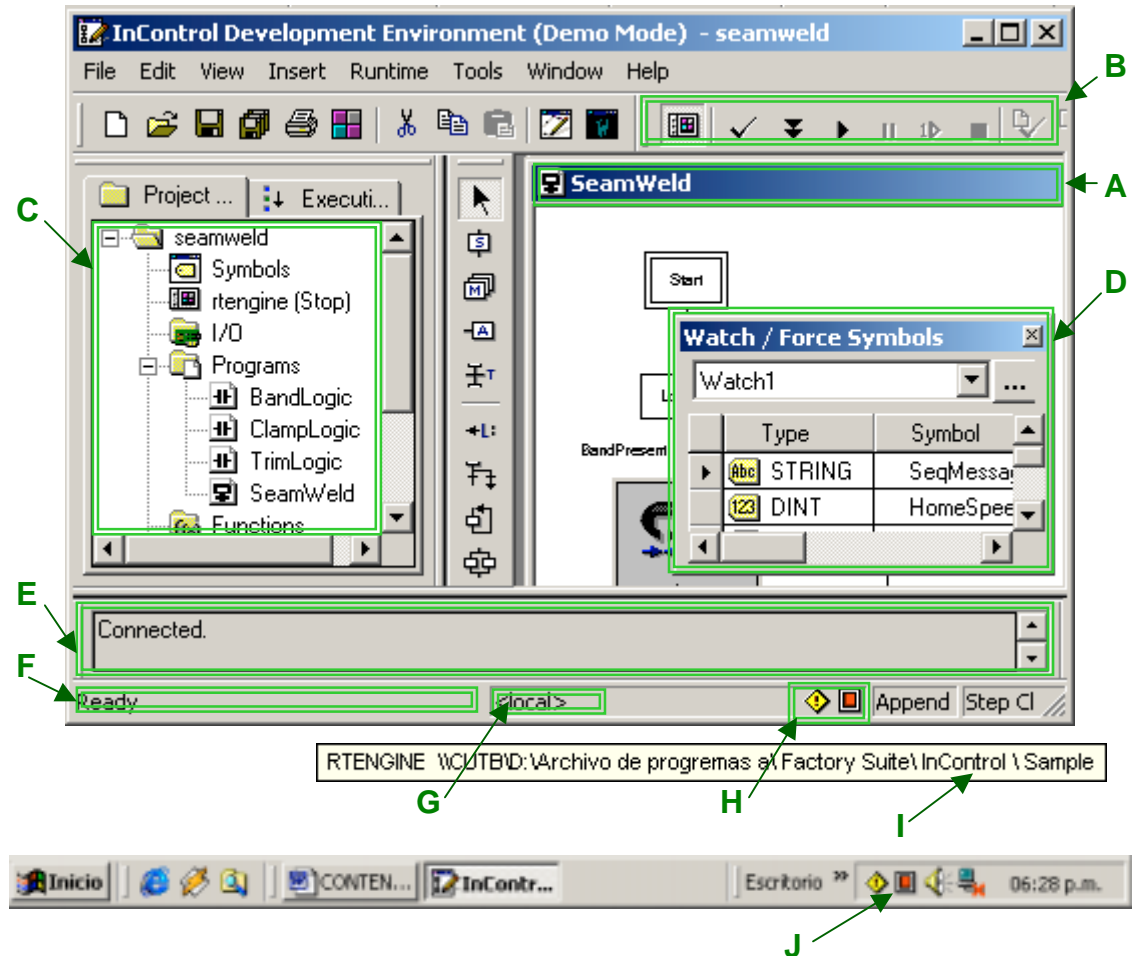
Cuadro 2. Elementos de la Ventana de Desarrollo

Objetos	Elementos de la Pantalla	Función
A	Barra de Menú	Muestra las funciones estándar en formato texto.
B	Barra de herramientas Estándar	Muestra las funciones estándar como iconos.
C	Barra de Herramientas del Editor	Muestra las herramientas usadas para adicionar elementos de programa a un programa. Esta barra cambia, dependiendo del tipo de programa que esta siendo editado.

Objetos	Elementos de la Pantalla	Función
D	Barra de Título de la Ventana de Edición	Muestra el nombre del programa. Un asterisco en el nombre del programa indica que el programa ha sido modificado, pero no guardado, y/o es diferente de la copia que está corriendo en el <i>Runtime Engine</i> . Si una copia del programa está corriendo, su modo (Correr, Pausa, Detenido) también es mostrado. Para funciones, bloques de funciones y objetos de fábrica InControl que son cargados en el <i>Runtime Engine</i> , el modo cargado es mostrado
E	Ventana de Edición	Área de trabajo para la edición.
F	Ventana de Proyectos	Muestra las funciones específicas de un proyecto: programas, organización I/O y prioridad de ejecución.
G	Ventana de Salida	Muestra los mensajes (incluyendo mensajes de error), del <i>Runtime Engine</i> , el compilador y otros elementos de programas. Esa información también es escrita en el <i>Wonderware Logger</i>
H	Barra de Estados	Muestra información del programa o de ciertos íconos.

1.2.7.2 Ventana de Ejecución. El esquema típico para la ventana de ejecución es como se muestra en la Figura 6.

Figura 6. Ventana de Ejecución



Cuadro 3. Elementos de la Ventana de Ejecución

Objetos	Elementos de la Pantalla	Función
A	Barra de Título del Programa	Muestra el nombre del programa. Si una copia del programa está corriendo, su modo (<i>Run, Pause, Stopped, etc</i>) también es mostrado.
B	Barra de herramientas de Ejecución	Muestra las funciones de ejecución estándar como iconos.

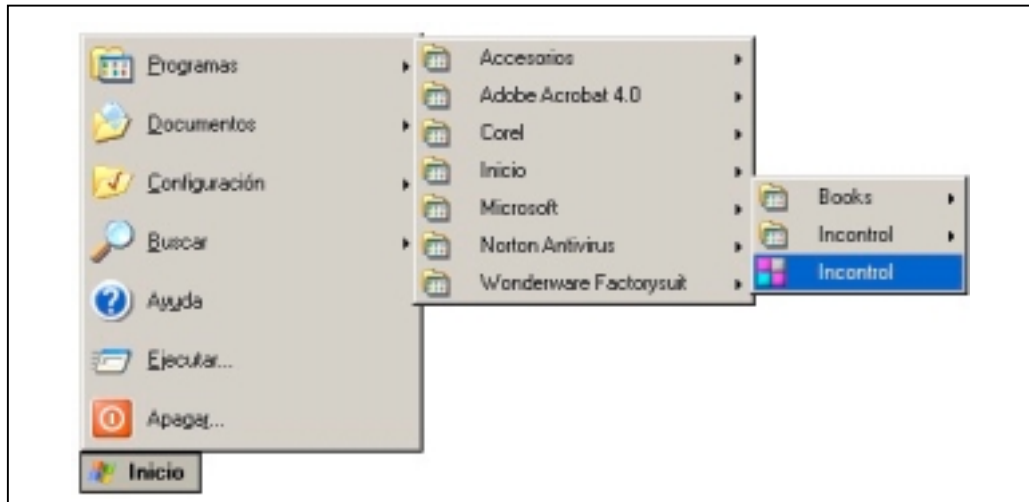
Objetos	Elementos de la Pantalla	Función
C	Ventana de Proyectos	Muestra las funciones específicas de un proyecto: programas configuración E/S, organización E/S y prioridad de ejecución.
D	Ventana Watch (Observación)	Muestra las variables (símbolos) y su estado de ejecución. Usted puede usar independientemente la ventana <i>Watch</i> si no quiere abrir el ambiente de Desarrollo o Ejecución.
E	Ventana de Salida	Muestra los mensajes del <i>Runtime Engine</i> , el compilador, los elementos del programa, etc.
F	Barra de Estado	Muestra información del programa.
G	Nodo Conectado	Muestra el nodo al cual el ambiente de Desarrollo esta conectado.
H	Icono del RTE Conectado	Muestra el estado del <i>Runtime Engine</i> en el nodo local o remoto, al correr el proyecto.
I	Proyecto Cargado	Herramienta de información que reporta el nombre del proyecto cargado en el <i>Runtime Engine</i> .
J	Icono de Monitoreo del Runtime Engine	Monitorear el estado del <i>Runtime Engine</i> en el nodo local. Puede ser usado para controlar el modo del proyecto y ajustar el tiempo de exploración o análisis.

1.2.8 Ejecución de InControl. Los siguientes pasos se deben seguir para iniciar

InControl:

1. Hacer clic en **Inicio** sobre la barra de tareas. Luego seleccionar **Programas \ Wonderware FactorySuite**

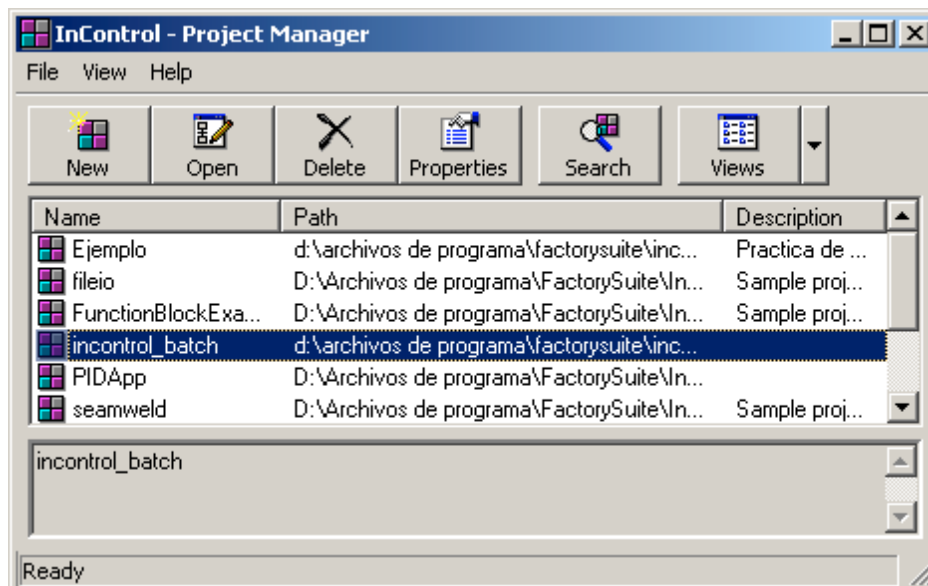
Figura 7. Iniciar InControl



2. Pulsar en **InControl**. Aparecerá el cuadro de diálogo **InControl Project Manager** donde se podrá abrir un proyecto o crear uno nuevo

Desde esta la ventana del Project manager se puede crear, acceder, eliminar o conocer las propiedades de un proyecto.

Figura 8. InControl Project Manager



1.3 INTOUCH

InTouch es uno de los componentes *Wonderware FactorySuite* donde se pueden crear de manera fácil y rápida aplicaciones de interfase hombre-máquina (HMI) en los sistemas operativos Microsoft Windows 95 (o siguientes) y Windows NT 4.0 SP4.

Las aplicaciones de *InTouch* son globales, se encuentran en una multitud de mercados incluso en el procesamiento de comida, semiconductores, aceite y gas, automotores, químicos, farmacéutico, la pulpa y el papel, transporte, utilidades, y más. Dentro del país se encuentran los siguientes clientes: British Petroleum,

Federación Nacional de Cafeteros, Bavaria, Aguas de Cartagena, ECOPETROL y ABOCOL S.A.

Con la utilización del *InTouch*, se pueden crear aplicaciones poderosas, que aprovechan las características importantes de *Microsoft Windows*, incluso los controles de ActiveX, OLE, gráficos, conexión de una red de computadoras y más.

InTouch también puede extenderse agregando controles comunes de *ActiveX*, *wizards*, objetos genéricos, y creando las extensiones de *InTouch QuickScript*.

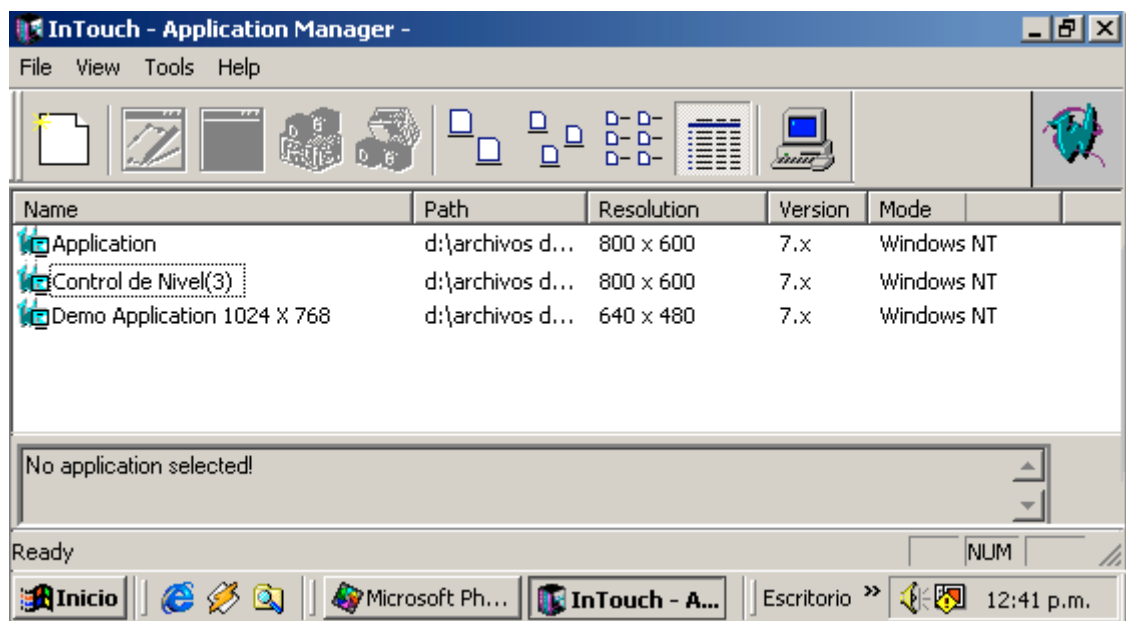
1.3.1 Características del InTouch. Estas son las principales características que conforman las aplicaciones en el *InTouch*:

- **Gráficas orientadas a objetos:** por medio de esto se pueden realizar operaciones de fácil configuración y animación.
- **Links de Animación:** incluye entradas discretas, análogas, deslizadores horizontales y verticales, botones de acción, etc.
- **Distribución de alarmas:** maneja varias alarmas simultáneamente y de localizaciones remotas.

- **Distribución histórica:** especifica la historia de la fuente de archivo de datos.

1.3.2 Ventanas en el ambiente InTouch. *InTouch* está constituido por tres grandes programas, el *InTouch Application Manager*, *WindowMaker* y *WindowViewer*. *InTouch* también incluye los diagnósticos del programa *Wonderware Logger*.

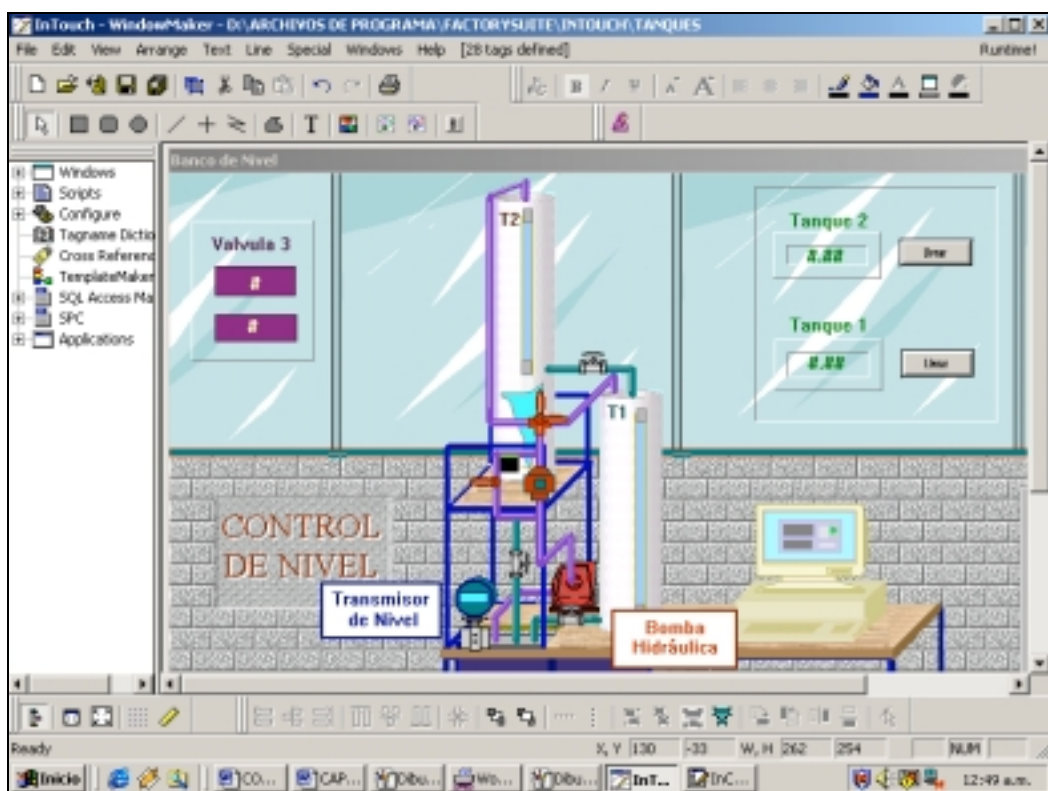
Figura 9. Application Manager de InTouch



El *InTouch Application Manager* (Figura 9) organiza las aplicaciones que se crean. También es usado para configurar al *WindowViewer* (Figura 11) como un servidor de NT, configurar *Network Application Development* (NAD) para las arquitecturas

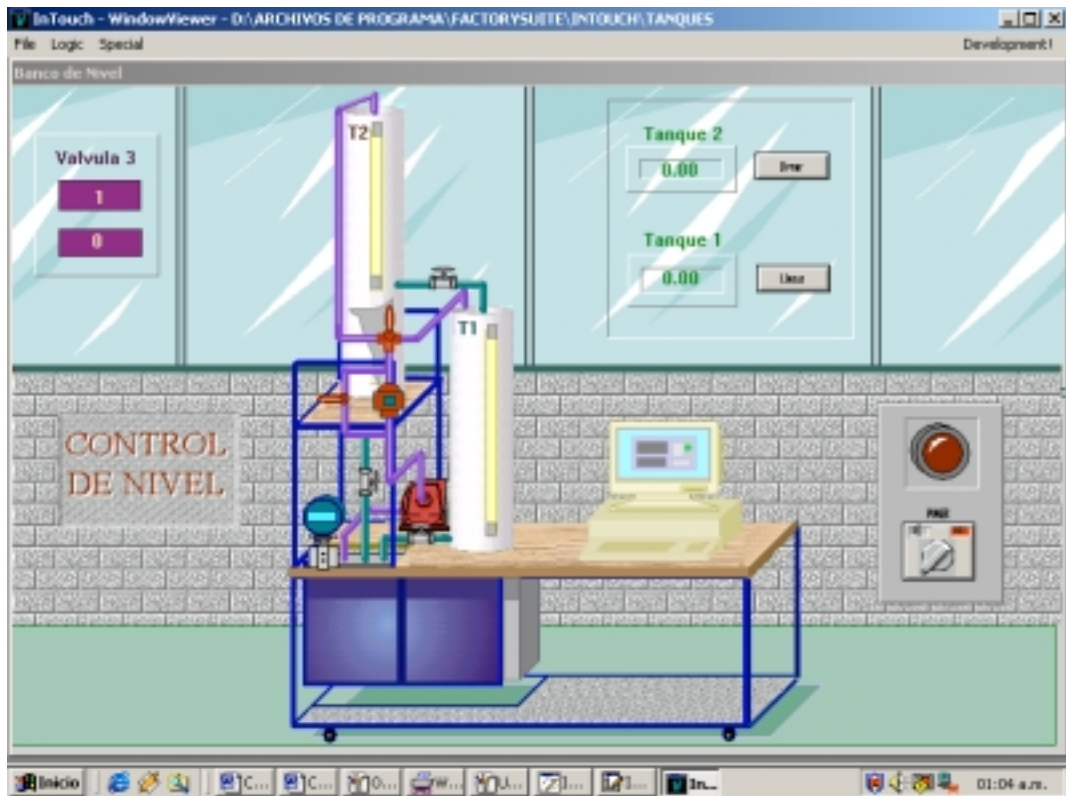
cliente-base y servidor-base, y para configurar la Conversión de la Resolución Dinámica (*Dynamic Resolution Conversión. DRC*) y/o alarmas distribuidas. Las bases de datos DBDump y DBLoad también se utilizan para empezar el *Application Manager*.

Figura 10. Window Maker de InTouch



WindowMaker (Figura 10) es el ambiente de desarrollo, donde los gráficos de orientación de objetos son usados para crear animaciones y desplegar ventanas. Estas ventanas desplegadas pueden conectarse a los sistemas industriales I/O y a otras aplicaciones de Microsoft Windows.

Figura 11. Window Viewer de InTouch



WindowViewer (Figura 11) es el ambiente *runtime* usado para el despliegue de las ventanas gráficas creadas en *WindowMaker*. *WindowViewer* ejecuta *InTouch QuickScripts*, realiza registros e informes de datos históricos, registra e informa las alarmas de procesos, y puede funcionar como un cliente y un servidor para DDE y protocolos de comunicación *SuiteLink*.

1.3.3 Términos comunes. Posteriormente se definen los términos más comunes en el desarrollo de las aplicaciones:

- **Gráficas Orientadas a Objetos.** Sus poderosas herramientas de diseño orientadas a objetos hacen que el dibujar, localizar, alinear, colocar objetos unos sobre otro, espaciar, rotar, invertir, duplicar, cortar, copiar, pegar y borrar objetos resulte sumamente fácil. El gráfico que resultado de la utilización de estas herramientas se conoce como **mímico**.
- **Links de animación.** Es posible combinar vínculos de animación para lograr cambios complejos de tamaño, color, movimiento, y/o cambios de posición. Los vínculos de animación incluyen datos de entrada discreta, análoga y de texto; barras deslizables horizontales y verticales; botones discretos y de acción.
- **Wizards.** Ellos ahorran una cantidad considerable de tiempo durante el desarrollo de la aplicación. Los *wizard* son fáciles de usar y de configurar. Cuando se selecciona un *wizard*, se pega en una ventana y luego, se hace doble clic donde aparece una caja de diálogo para su configuración.

Usando los *wizards*, no se pasa el tiempo dibujando los componentes individuales para el objeto, o entrando los rangos de valor, o animando el objeto.

- **Application Explorer.** InTouch incluye el poderoso *Application Explorer* (Explorador de Aplicaciones). El Explorador de Aplicaciones es un despliegue jerárquico de los objetos que componen la aplicación de InTouch. Permite un acceso rápido, fácil, e intuitivo a todos los parámetros de aplicaciones incluyendo: ventanas, QuickScript, Diccionario de Tagname, SuperTag, acceso a SQL, SPC o SPCPro, y más. El Explorador de Aplicaciones permite ejecutar desde WindowMaker cualquier aplicación de Windows, como es el caso de otros componentes de FactorySuite, Word o Excel de Microsoft, o paquetes de programación de PLCs. Esto permite al WindowMaker convertirse en la herramienta de desarrollo central para toda una aplicación de automatización.
- **Tendencias históricas distribuidas.** InTouch permite especificar de manera dinámica diferentes fuentes de datos de archivos históricos para cada una de las trazas en una grafica de tendencias. Estas fuentes de archivos históricos pueden ser otras bases de datos de InTouch o cualquier base de datos de IndustrialSQL Server.

Dado que InTouch permite el uso de hasta 8 trazas por grafica de tendencias, los usuarios pueden tener una cantidad sin precedentes de datos históricos disponibles para su visualización en cualquier momento dado.

- **Scripts.** Dentro de los *script* se escriben comandos y operaciones lógicas basadas en un evento específico que pueden ser: Ejecución de la aplicación (*Application*), pulso de un botón (*Key*), cambio del valor de un tagname (*Data Change*) o abertura de una ventana (*Window*).

El lenguaje de creación de “*Scripts*” de *InTouch*, es tan poderoso, flexible y fácil de usar que se puede crear la lógica en “*Scripts*” con tan solo apuntar y hacer click sin jamás tocar el teclado. El *QuickScript* permite crear las llamadas *QuickFunctions*. Un ejemplo del uso de los *QuickScripts* es abrir una ventana, el cambio de una válvula, etc.

1.4 HERRAMIENTAS QUE SOPORTAN EL TUTORIAL DE MANEJO DEL SOFTWARE INCONTROL

Las herramientas utilizadas en el diseño del Tutorial de Manejo del *software InControl* permiten crear un entorno dinámico y sencillo, esta acompañado de animaciones interactivas que le resultan familiar al usuario y que sobretodo en el momento de ejecutar una pagina no hicieran de ésta algo muy complejo al momento de su presentación.

El tutorial fue creado bajo los lenguajes de programación HTML y JavaScript que en la actualidad son las tecnologías mas usadas en la WWW (*World Wide Web*)

en el INTERNET. Para una mejor presentación e interactividad se emplearon herramientas gráficas y otras herramientas de animaciones las cuales se describirán posteriormente.

1.4.1 Lenguajes y tecnologías. Debido al creciente auge que ha tomado la WWW (*World Wide Web*), se han creado una serie de técnicas sobre la publicación de cualquier tipo de información, a través de la creación de paginas Web por medio del lenguaje **HTML** y otras herramientas especializadas.

El HTML es un lenguaje descriptivo con el cual es posible codificar un hipertexto de manera tal que los programas navegadores lo entienden. Esta codificación se lleva a cabo a través de determinadas etiquetas, más conocidas como tags, que dan forma al documento, es decir, cómo debe aparecer la información (texto y gráficos) y cómo se ejecutan otros tipos de archivos (por ejemplo, archivos de sonido y vídeo).

FrontPage es un programa para la edición de páginas web de *Microsoft*. Está orientado a personas inexpertas y sin conocimientos profundos en HTML. Ofrece un conjunto de herramientas completo y de fácil manejo para la creación, instalación y mantenimiento de sitios web. Sus capacidades son semejantes a las de otros editores, como el crear mapas de imágenes, gestionar la arborescencia de las páginas del sitio, etc.

Otra de las Tecnologías muy utilizadas en la creación de sitios Web es el **JavaScript**, ya que le aporta dinamismo del lado del usuario (Visitante de la Página), permitiéndole interactuar con la página de diferentes formas.

JavaScript es un Lenguaje de alto nivel, basado en objetos con el cual se pueden generar documentos Web interactivos de un modo sencillo. Ofrece todas las características básicas de un lenguaje orientado a objetos sin la complejidad de lenguajes como C++ o Java.

Gracias a su vocabulario limitado facilita su comprensión y permite una interacción dinámica entre el usuario y la página HTML. También posee un conjunto de herramientas compactas propias de JavaScript que permiten responder a eventos tales como: pulsación del ratón, entradas de formulario, navegación interna en la página, entre otros.

Como la mayoría de los lenguajes *Script*; *Javascript* es interpretado en el momento de ejecución por el navegador antes de compilarse. Pero a su vez se convierte en una desventaja ya que es necesario esperar a que se ejecute el código para ver su respuesta al evento. Pero cabe anotar que son fáciles de utilizar. La gran ventaja de *JavaScript* es que se adhiere directamente al código HTML permitiendo que algunos elementos HTML reaccionen a un número limitado de acciones que genera el usuario. Tales acciones pueden ser el cambio de una

imagen cuando el puntero del ratón pasa sobre ella o al pulsar sobre un enlace (Hipervínculo) en una imagen mapeada (Imagen seccionada por hipervínculos).

1.4.2 Herramientas de dibujo. La herramienta **CorelDraw Paint 9** es un programa de dibujo vectorial que facilita la creación de ilustraciones profesionales: desde simples logotipos a complejas ilustraciones técnicas. *CorelDraw* proporciona una variedad de herramientas y efectos que le permiten trabajar de manera eficiente para producir gráficos de alta calidad. Asimismo, permite enviar una ilustración a un servicio de filmación para su impresión o para la publicación de su documento en Internet.

Los programas informáticos de imágenes están basados en la creación de gráficos vectoriales o de imágenes de mapa de bits.

- **Mapas de bits.** Los programas de edición de fotografías como Corel PHOTO-PAINT operan con imágenes de mapa de bits. Si se trabaja con este tipo de imágenes, se pueden depurar pequeños detalles, introducir cambios radicales y dar mayor intensidad a los efectos. Las imágenes de mapa de bits, también conocidas como imágenes ráster, están compuestas de puntos individuales denominados píxeles dispuestos y coloreados de formas diversas para conformar un patrón.

- **Gráficos Vectoriales.** Las imágenes vectoriales, también llamadas imágenes orientadas al objeto o imágenes de dibujo, se definen matemáticamente como una serie de puntos unidos por líneas. Los elementos gráficos presentes en un archivo vectorial se denominan objetos. Cada objeto es una entidad completa con propiedades tales como color, forma, contorno, tamaño y posición en la pantalla, que están incluidas en su definición.

Considerando que cada objeto es una entidad completa, se pueden mover y cambiar sus propiedades una y otra vez manteniendo la claridad y nitidez originales, sin afectar a los restantes objetos de la ilustración. Estas características hacen que los programas vectoriales sean idóneos para la ilustración, en la que el proceso de diseño requiere a menudo la creación y manipulación de objetos individuales.

Corel PHOTO-PAINT es un programa de edición de imágenes de mapa de bits que resulta idóneo para realizar retoques fotográficos o crear gráficos originales. Corel PHOTO-PAINT proporciona herramientas y medios de un estudio de diseño gráfico profesional. Con sólo hacer clic con el ratón, se pueden elegir de un vasto conjunto de medios y texturas, infinidad de colores, pinceles de todos los tamaños y formas, y una biblioteca de imágenes ya preparadas. Crea imágenes nuevas, retoque fotografías, añada texto y efectos especiales, o cambia la iluminación de

los objetos. Corel PHOTO-PAINT incluye cientos de funciones que se pueden utilizar para imitar técnicas de pintura y fotografía, o para desarrollar un estilo propio. También puede animar las imágenes y compartirlas publicándolas en Internet.

La herramienta de dibujo **Paint** puede utilizarse para crear dibujos sencillos o complicados. Estos dibujos pueden ser en blanco y negro o en color, y pueden guardarse como un archivo de mapa de bits. Se puede imprimir el dibujo, usarlo como fondo del escritorio o pegarlo en otro documento. Incluso es posible utilizar Paint para ver y modificar fotografías digitalizadas.

El programa **Photoshop 4.0** es una aplicación integrada tanto en el entorno Windows como en el Macintosh. Este programa permite la creación y modificación de imágenes a través de un entorno sencillo de uso y cuyos resultados son unos ficheros de pequeño tamaño, que permite agilizar su visualización e incluso su inserción en Internet. El Photoshop 4.0 incluye una serie de ventajas respecto de su versión anterior, como son el mejor manejo de las selecciones, transparencias, paletas de colores, una barra de tareas mucho más intuitiva, así como una nueva gama de filtros más potentes, eficaces y a la vez accesibles. Además, introduce una serie de opciones nuevas, como por ejemplo, las acciones, el trabajo automático por capas y otros.

1.4.3 Herramientas para animaciones. La herramienta mas utilizada actualmente para tipo de animaciones en la web, es **Flash Macromedia 5**. *Flash* debe sus raíces a una pequeña compañía llamada *FutureSplash* que fue adquirida por *Macromedia* en 1997 para complementar su programa director que sirve para la creación de producciones multimedia interactivas, títulos de CD/DVD, etc., cuando deseaban darle un enfoque para la Web.

La clave de *Flash* es que es un programa de animación vectorial. Esto significa que se pueden crear animaciones complejas: aumentar y reducir elementos de la animación, mover de posición estos objetos, y otras tareas sin que la animación cambie su contorno original y ocupe mucho espacio en el disco. Los vectores con los que trabaja *Flash* sólo son, por decirlo de alguna manera, siluetas que casi no ocupan espacio y se pueden modificar fácilmente y sin gasto de memoria en disco.

Flash es el software de creación más avanzado para crear animación interactiva escalable para la Web. Mediante él se crean logotipos animados, controles de navegación de sitios Web, animaciones de gran formato o sitios Web completos de *Flash*.

Flash 5 ofrece nuevas funciones para la creación de sitios Web interactivos interesantes y atractivos. Entre las nuevas funciones de reproducción se incluyen: flujos de sonido MP3, campos de texto para que el usuario pueda incluir texto editable en una película y mejoras en las acciones de *Flash* que permiten crear

juegos, formularios y cuestionarios. Entre las nuevas funciones de creación se incluyen: un proceso de publicación fluido, una ventana de biblioteca con diseño nuevo, varios inspectores nuevos y una interfaz gráfica mejorada.

Un programa que sirve de ayuda para las animaciones *Flash* llamado **Flax**. Este es un programa muy útil pues con él se pueden crear efectos de texto espectaculares para las animaciones *Flash* y de una manera extremadamente fácil, con una interfaz muy amigable escribiendo el texto que se desee animar y escoger la animación adecuada (hay más de 22 predefinidas). También es altamente configurable, y se puede cambiar la mayoría de las opciones, como los colores y los tipos y tamaños de letras, pero también puedes configurar los efectos a tus gustos y necesidades. También se pueden cambiar la velocidad de movimiento de las letras, grados de rotación, etc. Cada efecto tiene sus propias opciones configurables.

Este programa también es bueno para profesionales, pues permite ahorrar de una forma extraordinaria el tiempo. En lugar de perder el tiempo creando efectos en Flash (que la mayoría sólo es posible con animación FRAME-BY-FRAME) se utiliza *Flax* y se trabajan en las otras partes de tus películas, y por consiguiente se ahorra tiempo.

El **Swish 2** es un editor de animaciones *Flash* pero que no ocupa tanto como Macromedia Flash y con el que se pueden crear animaciones en un momento sin

tener ni idea de cómo se utiliza el programa *Flash*. Simplemente se elige el texto a poner, se retoca un poco y se añaden efectos.

Permite crear producciones de animaciones complejas, imágenes, gráficos y sonido. Tiene mas de 150 efectos incorporados como destellos, vórtices, vueltas 3D, serpientes y muchos más. *Swish* tiene herramientas para crear líneas, rectángulos, elipses, curvas, y trayectorias en movimiento, esta herramienta, exporta el formato del archivo de SWF usado por el flash de *Macromedia*; así que su animación se ejecutará en cualquier computador que tenga *Flash*. Las animaciones de SWISH se pueden incorporar en cualquier pagina Web o importar en *Flash*. Pueden también ser enviadas por medio e-mail, ser encajadas en una presentación de Microsoft PowerPoint o ser incluidas en un documento de Microsoft.

A la animación creada se le llama película. Dentro de cada película está una colección de escenas y cada escena tiene un tiempo. Durante el tiempo de cada escena, se colocan los objetos (texto, imágenes, etc.) a los cuales se le aplican efectos. Estos efectos comenzarán y se detendrán en los capítulos particulares y pueden ser controlados insertando acciones y acontecimientos. En las escenas, los acontecimientos ocurren cuando la película alcanza cierto tiempo.

1.5 PLC SIMATIC S7-300 CPU314C-2DP

Es un sistema miniautómata desarrollado por SIMATIC que se caracteriza por su tamaño reducido y compacto, rapidez, múltiples tareas y variedad de prestaciones.

Figura 12. PLC SIMATIC S7- 300



1.5.1 Características. Posee una memoria de programa con 16 k instrucciones como capacidad máxima, 1024 entradas / salidas digitales y 32 módulos dentro de un solo sistema (para tareas especiales se ofrecen módulos específicos), alta potencia de cálculos aritméticos con 32 bits en coma flotante y una interface multipunto o puerto MPI.

Otras características de SIMATIC S7-300 que pueden destacarse son la modularidad (numerosos módulos de extensión), la comunicación por bus, las

funcionalidades integradas de visualización y operación así como su lenguaje de programación bajo entorno Windows.

El miniautómata S7-300 se encuentra en procesos de evolución rápida y en aplicaciones que exigen tratamiento de datos en paralelo a las funciones de automatismo. Por ejemplo, en maquinaria de embalaje y en máquinas de herramienta, en el sector agroalimentario o en la industria química o farmacéutica.

1.5.2 Funcionalidades. SIMATIC S7-300 viene dotado de funcionalidades complejas. El juego de instrucciones de cálculos tratamiento de bits con números de 32 bits.

Las operaciones aritméticas, tales como la multiplicación y la división, están integradas, lo mismo que la evaluación de flancos y otras funciones similares.

La interfaz de programación multipunto, (MPI) de 187.5 kbits/s, soporta comunicación simultánea con equipos de programación, paneles de operador y otros autómatas sin necesidad de hardware suplementario. Esto trae como consecuencia la reducción del número de interfaces opcionales.

Sin embargo, la mayor ventaja en materia de potencia es la que aportan los módulos de función, cada módulo incorpora 32 canales con separación galvánica.

Así pues, pueden lograrse más funciones en menos espacio. La conexión frontal se establece por conectores.

La CPU siempre está perfectamente informada de la configuración del autómata. Ella la almacena automáticamente en memoria y, si lo desea, procederá a su verificación utilizando un mecanismo automático.

En una configuración de autómatas en red, el puesto central de mando puede acceder directamente a cualquier CPU y a cualquier módulo de función, a cualquier panel de operador y a cualquier procesador de comunicaciones de la red, todo ello sin hardware ni software adicional.

El SIMATIC S7-300 es integrable en un automatismo existente por medio de PROFIBUS-DP, el bus de campo PROFIBUS de Siemens. Para una pequeña red de automatización, cada S7-300 dispone del mecanismo de comunicación, "Datos globales", que permite realizar de forma muy simple el intercambio rápido de pequeños volúmenes. Ver anexo G, Manual de hardware del PLC S7-300.

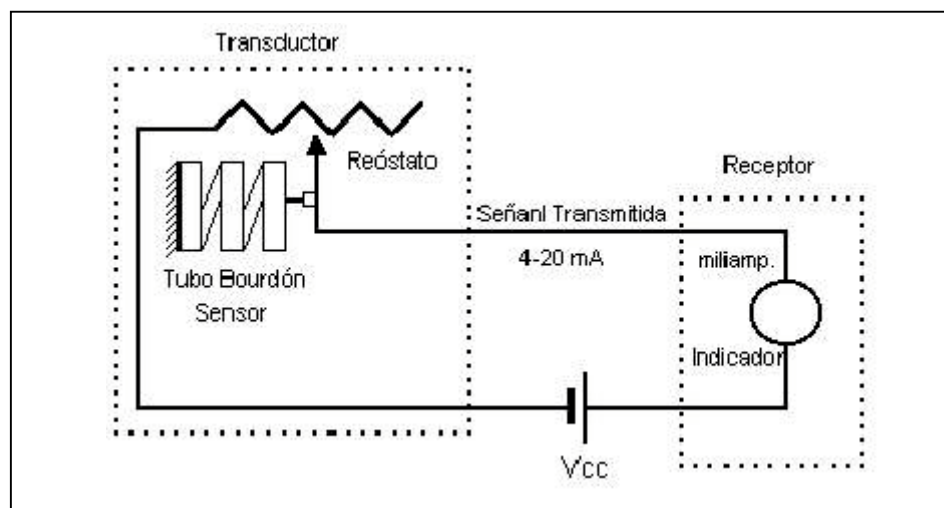
1.6 REDES DE COMUNICACIONES INDUSTRIALES

En la búsqueda de la integración de las comunicaciones industriales, fueron desarrolladas las Redes de Comunicaciones Industriales (RCI). Las RCI, tienen su origen en los estudios efectuados por la fundación *FieldBus*, que buscaba la

creación y desarrollo de esquemas de comunicaciones universales y de arquitectura abierta.

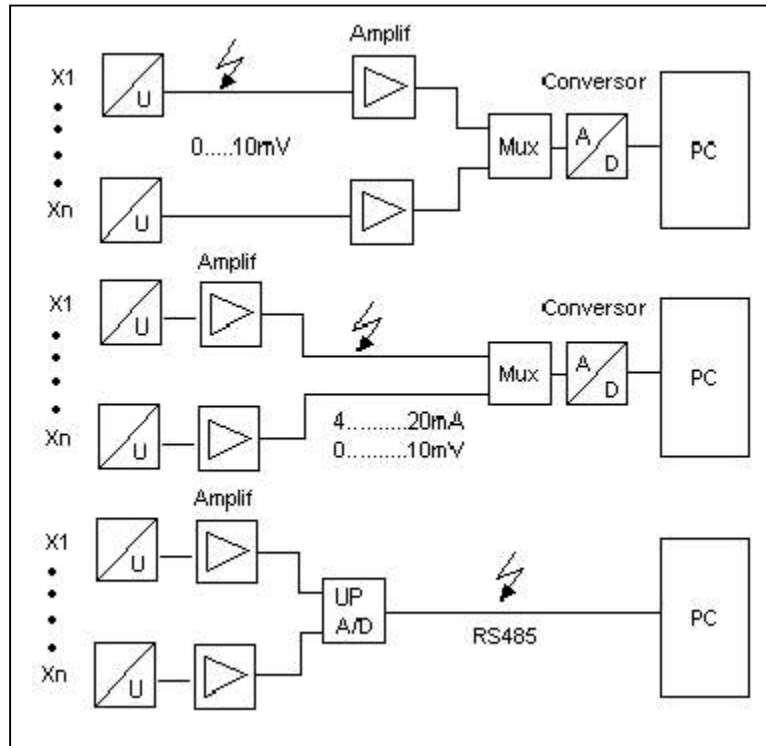
La comunicación industrial se ha venido realizando mediante una conexión física (cable) que conecta exclusivamente cada sensor o cada actuador a su equipamiento de control, donde la información se transmite por una señal analógica (generalmente 4-20 mA).

Figura 13. Conexión física industrial



Ahora se sustituye la transmisión analógica punto a punto por una digital. Los dispositivos de campo (sensores y actuadores) y dispositivos de control comparten una única línea bidireccional para transmitir información entre ellos (Bus de Campo/ Fieldbus). Las señales analógicas son convertidas a digital en los mismos dispositivos de campo.

Figura 14. Conversión de señales analógicas a digitales



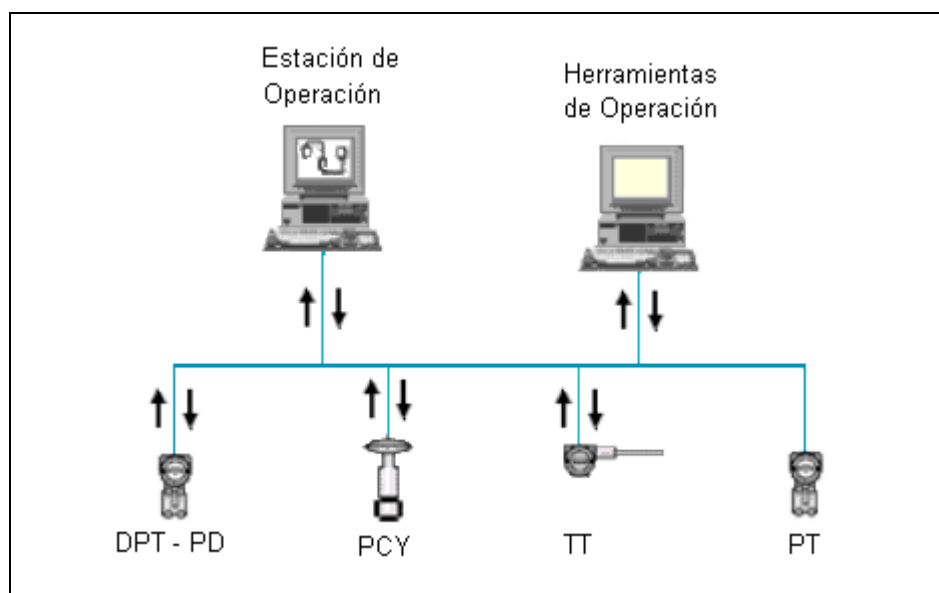
La comunicación digital, soportada por componentes de hardware y software, ha permitido implementar prestaciones de gran trascendencia en el diseño de los sistemas de automatización en plantas, pudiéndose mencionar economía de cableado, programación a distancia de los dispositivos de campo, recibir información de diagnóstico, distribuir funciones de control entre los dispositivos (control real distribuido), facilidades de sustitución y modularidad, disponibilidad de información para mantenimiento predictivo, etc. La distribución de funciones hace más confiable al sistema y disminuye el costo de los tradicionales dispositivos de control centralizados como PC's o PLC's, disminuyendo sus capacidades de

procesamiento y memoria. Toda la información generada en el proceso de fabricación puede ahora archivarse en una base de datos de planta, la que a su vez puede integrarse con el sistema administrativo.

La red *Profibus* esta basada en los estándares de *FieldBus*, este realiza el proceso de adquisición de datos y los transmite a niveles gerenciales.

1.6.1 FIELDBUS. Es un sistema de comunicación digital bidireccional (Figura 15) que permite la interconexión en red de múltiples instrumentos directamente en el campo, realizando funciones de control, monitoreo de procesos y estaciones de operación (HMI) a través de un software de supervisión.

Figura 15. Comunicación digital bidireccional



1.6.2 Protocolo Profibus. La fundación FIELDBUS, desarrolló un protocolo de comunicación, para la medición y control de procesos donde todos los instrumentos puedan comunicarse en una misma plataforma. *FIELDBUS* permite disponer de una nueva tecnología para una nueva generación de sistemas de control y automatización, físicamente más simple, donde toda la rutina de control regulatorio y control lógico, es efectuado por dispositivos de campos, posibilitando además una arquitectura abierta donde cualquier fabricante de equipos de instrumentación pueda integrarse a la red de campo existen en una fabrica o empresa.

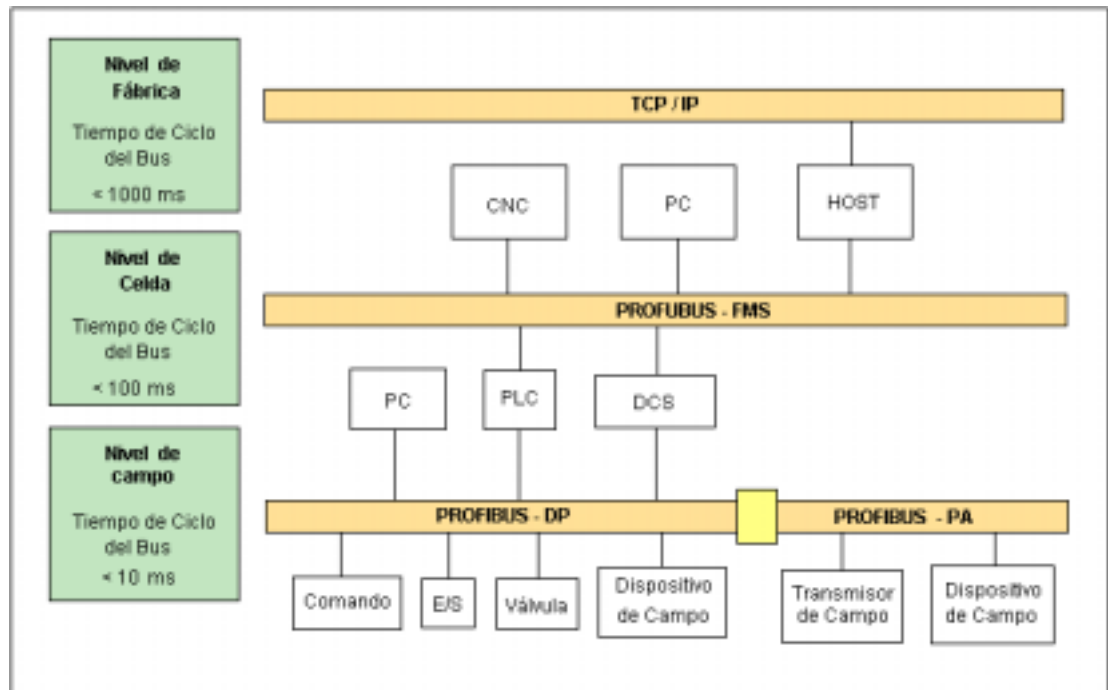
En la mayoría de las fábricas se requieren de estaciones remotas, estas deben comunicarse a través de un bus de comunicación con las computadoras ubicadas en las diferentes salas de control, para así conocer cómo está funcionando la planta. Los usuarios (industrias o fabricas) requieren un sistema de bus de campo con las siguientes características:

- Aptitud universal para los distintos equipos, sectores y aplicaciones.
- Normalización en ISO, DIN u organismo de normalización semejante.

PROFIBUS cumple con los anteriores requisitos. Normalizado con EN 50 170, tomo 2, norma PROFIBUS, este bus ofrece interfaces de usuario tanto para comunicaciones rápidas con dispositivos de campo, por ejemplo, estaciones

periféricas o descentralizadas o accionamientos, como para un amplio intercambio de dato entre equipos maestros. Posee distintas interfaces de usuario (Figura 16):

Figura 16. Comunicación ProfiBus y sus diferentes interfaces de usuario



- **PROFIBUS- FMS:** (Sistema de Bus de Campo para Especificación de Mensajes de Bus de Campo) ofrece servicios de usuarios estructurados para la comunicación abierta en pequeñas células (valores característicos de 10-15 equipos de automatización como autómatas SIMATIC o PC's). En estas configuraciones, lo principal es el voluminoso intercambio de información y no el tiempo de respuesta de los mismos.

- **PROFIBUS-DP:** (Sistema de Bus de Campo con Periféricos Descentralizados) es la interfaz de usuario para el acoplamiento de dispositivos de campo, por ejemplo, accionamiento, estaciones periféricas descentralizadas ET200, isletas de válvulas).
- **PROFIBUS-PA:** (Sistema de Bus de Campo para procesos de automatización) se utiliza para la automatización de procesos en recintos expuestos al peligro de explosiones (áreas clasificadas). El proceso de transmisión cumple la norma internacional IEC 1158-2, el perfil de protocolo es PROFIBUS - FMS.

El PROFIBUS ofrece, además de ello, el interfaz optimizado SEND/RECEIVE para permitir una sencilla comunicación entre sistemas SIMATIC (equipos automáticos fabricados por SIEMENS).

Al igual que en el industrial Ethernet, también PROFIBUS, permite la creación de una red con cable bifilar o cables de fibra óptica.

Cuadro 4. Tipos de versiones Profibus

Protocolo	Aplicable para	Interfaces de Usuario
FMS	SIMATIC.S5/S7 PG/PC, HMI	Funcionalidad elevada
DP	Dispositivo de campo binario y analógicos inteligentes.	Optimizada para comunicaciones con dispositivos de campo.
SEND/RECEIVE	SIMATIC,S5/S7 PG/PC, HMI	Funcionalidad elemental
Funciones S7	SIMATIC,S7 PG/PC, HMI	Funcionalidad elevada para com. Con SIMATIC S7

El método de acceso a PROFIBUS funciona por el procedimiento "Token Passing con maestro-esclavo subyacente" según EN 50 170, tomo 2. En este método se distingue entre aparatos (estaciones) de red activos y pasivos. El "Token" lo reciben únicamente los aparatos activos acoplados al bus. Este Token es el derecho a emisión que un aparato activo acoplado al bus transmite al siguiente dentro de un período de tiempo predefinido.

Se identifica automáticamente si ha fallado un aparato acoplado al bus o si se ha incorporado un aparato más. Todos los aparatos que integran la red deben estar configurados a idéntica velocidad de transmisión.

1.6.3 Sistemas de transmisión Profibus. Profibus ofrece un amplio espectro de componentes de red para sistemas de transmisión eléctricos y ópticos.

- **Red eléctrica:** la red eléctrica utiliza un cable bifilar trenzado apantallado como medio de transmisión. La Interfaz RS-485 funciona con diferencia de tensión. Por este motivo, es más inmune a las interferencias que una interfaz de tensión o de corriente. En PROFIBUS los aparatos pertenecientes al bus están conectados a éste a través de un terminal de bus o un enchufe de conexión a bus (máximo 32 equipos acoplados por segmento). Los distintos segmentos se conectan a través de repetidores.

La velocidad de transmisión puede configurarse por nivel desde 9.6 Kbits/seg hasta 1.5 Mbits/seg según PROFIBUS para aplicaciones en DP. La máxima longitud del cable (trenzado y apantallado) depende de la velocidad de transmisión.

Tabla 1. Velocidades de transmisión

Velocidad (Kb/s)	9.6	19.2	93.75	187.5	500	1500	12000
Distancia/Segmento (m)	1200	1200	1200	100	400	200	100

- **Red óptica:** la red óptica de PROFIBUS, utiliza un cable de fibra óptica como medio de transmisión. La variante del cable de fibra óptica es inmune a las interferencias electromagnéticas, es apta para grandes distancias de transmisión (cables de fibra opcionalmente de plástico o vidrio). La velocidad de transmisión puede configurarse por niveles desde 9.6 Kbits/seg hasta 1.5 Mbits/seg (Según EN 50 170, tomo 2, PROFIBUS). La longitud máxima de segmento en la variante óptica del PROFIBUS es independiente de la velocidad de transmisión (excepción anillos ópticos redundantes.) La configuración de las redes de fibra óptica se realiza mediante OLM (Optical Link Module) con cable de fibra óptica de vidrio o de plástico.

Con OLM es posible crear una red óptica con topología lineal, en anillo o en estrella. Con la ayuda de OLP (Optical Link Plugs) se puede realizar anillos monofásicos sencillos de plástico.

El aparato terminal se conecta directamente a OLM u OLP. La conexión de aparatos terminales a OLP se realiza únicamente con estaciones pasivas PROFIBUS (esclavos DP/FMS).

Los anillos ópticos pueden configurarse como anillos monofásicos (económicos) o como anillos bifásicos (superior disponibilidad de la red).

- **Red combinada:** son posibles estructuras mixtas de red PROFIBUS eléctrica y óptica. La transición entre ambos soportes se realiza a través del OLM. En la comunicación entre los aparatos acoplados al bus no existe ninguna diferencia entre los que están interconectados a través de un sistema eléctrico y lo que están a través de fibra óptica. Como máximo pueden conectarse 127 aparatos a una red PROFIBUS (Siemens, Catálogo, IK 10, 1997).

Las redes de comunicación industrial, permiten conocer todo lo referente a un proceso industrial a través de las variables fundamentales medidas por instrumentos instalados en campo, permitiendo a la gerencia saber como está funcionando su empresa. Además otro aspecto fundamental es que permite controlar a grandes distancia la planta. Para realizar el control y poder integrar cada uno de los instrumentos de campo es necesario tener un estándar para que puedan ellos comunicarse.

Profibus, basada en el estándar de la FielBus, permite acoplar diferentes equipos de marcas distintas. Puede formar diferente topología (estrella, bus lineal o token) y siempre con arquitectura abierta.

1.7 TARJETA SMS CIF 30-DPM

La tarjeta CIF30 DPM PROFIBUS para slot PC/ISA y tiene un microprocesador AMD80188EM de 33Mhz para manejar las comunicaciones PROFIBUS y la interfase de memoria de puerto dual. El maestro DP soporta comunicaciones para tasa de datos por encima de 12 Mbaudios.

La interfase de comunicación CIF30 es una tarjeta inteligente PC/ISA. Es usada para manejar protocolos de transferencia de datos entre un PC y varias unidades de entradas y salidas de controladores industriales. Construido con un microprocesador capaz de manejar procesos de transferencia de datos independientemente y de este modo evitar funciones de tiempo crítico en PC. El intercambio de datos entre el PC y la CIF30 se realiza a través de la memoria de puerto dual (DPM) de 2k o 8k. Esta memoria es accesada simultáneamente por el PC y el microprocesador en la CIF. En cada Cif30 existe un chip FIELDBUS que interconecta directamente la red FIELDBUS. En adición, la CIF30 tiene una interfase serial de diagnostico. La CIF30 corre sobre un sistema operativo en tiempo real, el cual maneja la transferencia de datos por medio de la DPM y controla las secuencias de tiempo.

En modo de operación maestro, el dato de salida es ingresado en la memoria de puerto dual desde el PC y enviado al esclavo a través de la red PROFIBUS. La transmisión puede cíclicamente o puede ser introducida a través de una variable

de sincronización por el usuario. En paralelo con la salida el dato de entrada PROFIBUS es leído y está disponible para el usuario en la memoria de puerto dual. El usuario puede ser notificado del ciclo de transmisión por una interrupción de hardware o por sensado mismo de la tarjeta.

1.7.1 Características.

- Memoria de puerto dual flexible para manejar DP (Periféricos descentralizados), también trabaja con otros maestros CIF.
- Coprocesador diseñados para alto desempeño.
- Puerto de diagnostico RS-232.
- Watchdog timer.
- Memoria flash para fácil configuración.
- Interrupciones u operación por sondeo.

Cuadro 5. Especificaciones de la CIF30 - DPM

Especificaciones	
Capacidad I/O	1KB / 8192 puntos
Requerimientos de memoria de puerto dual para el PC	2K
Interrupciones soportadas	De 3 hasta 15
Desempeño de operación de la tarjeta	Maneja interrupciones o por sondeo.
Máxima Tasa de baudios	12 MB RS-485 aislado ópticamente.
Diagnóstico	4 LED's y puerto de diagnostico (RS-232)
Voltaje de operación	5 VDC, máximo 600mA
Temperatura de operación	0 - 50 °C
Tamaño PC/ISA	5.25 pulgadas
Información clasificada	
Tarjeta Profibus	SMS-CIF30-DPM
cable de diagnostico	SMS-CAB-SRV
Cable Profibus	SMS-CAB-PB

2. ANÁLISIS

2.1 CONSIDERACIONES Y REQUERIMIENTOS DEL CURSO DE CAPACITACIÓN EN EL SOFTWARE INCONTROL

El curso de entrenamiento en InControl debe abarcar cada uno de los temas que componen el aprendizaje del manejo de este software, esto corresponde al contenido presente en el Manual de Usuario del Software InControl. Este software emplea una gran cantidad de imágenes por sus características de editor de programas, como es el uso de barras de herramientas, iconos, menús, comandos y ventanas; se ve la necesidad de utilizar medios audiovisuales, como las diapositivas, que permitan adquirir el total dominio de este software. De igual forma, se necesita el apoyo de un tutorial en formato HTML de manejo del software InControl y un manual de usuario donde esté contenida toda la información concerniente a la configuración, programación y desarrollo de proyectos con el InControl; todo esto se complementará con el diseño de una aplicación en tiempo real con este software.

El objetivo principal de este curso es que el estudiante obtenga total dominio sobre el manejo del software InControl y así poder crear programas de aplicación para sus soluciones de control, para esto el curso sigue un orden de trabajo:

- Al iniciar el curso se introduce al estudiante en los aspectos principales del InControl: sus características, conceptos básicos, su instalación, los componentes de su sistema y su ambiente de trabajo junto con cada una de las herramientas (barras, menús, comandos) que le acompañan.
- Posteriormente se trabaja sobre proyectos ya creados para comprobar cada una de las tareas (carga, descarga, ejecución, eliminación, verificación, etc.) que el software InControl puede ejecutar en las aplicaciones.
- Luego se explican las herramientas de trabajo y editores correspondientes a la creación de nuevos proyectos para que el estudiante empiece a crear sus propios diseños, entre los que se encuentra el Symbol Manager para la creación y manejo de variables y usar cada uno de los editores de los lenguajes de programación.
- Como complemento a cada tema desarrollado se realizará una práctica de laboratorio.

2.1.1 Consideraciones y requerimientos de diseño de las ayudas del curso de capacitación en el software InControl. Entre los medios audiovisuales existentes se decidió hacer uso de diapositivas diseñadas por medio de la

herramienta *Microsoft PowerPoint*, debido a que permite el desarrollo del curso por medio de módulos que conllevan al estudiante a una mayor asimilación y entendimiento de los temas propuestos en este curso. Estas ayudas audiovisuales se utilizarán a la hora de impartir el curso de capacitación en el software InControl, en éstas se tiene como principal característica que el contenido del curso estará desarrollado en forma de módulos correspondientes a cada capítulo, además de ejemplos prácticos con el software InControl que complementaran el aprendizaje por parte del estudiante. Estas diapositivas se realizarán, de tal manera, que permitirán al estudiante ir complementando lo aprendido con el desarrollo de trabajo práctico en el software InControl.

Existen capítulos dentro del contenido del curso que hacen referencia a los lenguajes de programación de InControl: RLL, SFC y STL, los cuales se explican de forma general para que el estudiante empiece a diseñar programas sencillos en el momento de usar sus editores de programación.

2.1.2 Consideraciones y requerimientos de diseño para el tutorial de manejo del software InControl. Para llegar a manejar el software InControl en óptimas condiciones se precisa del conocimiento de una extensa información acerca del manejo, debido a que el InControl permite una amplia gama de tareas, entre ellas:

Escribir programas en tres distintos lenguajes, establecer seguridad, depurar, correr y examinar programas de aplicación, configurar FOE's y Drivers,

comunicar con InTouch y definir diversos estilos de variables. Además éste software emplea una gran cantidad de imágenes por sus características de editor de programas, tales como el uso de barras, iconos, menús, comandos y ventanas.

Por las características anteriormente mencionadas, se requiere de un tutorial que contenga información de cada una de las tareas principales soportadas por InControl para introducir al usuario en la creación de proyectos; además, dicho tutorial debe ir acompañado de un buen contenido de imágenes junto con dinámicas interactivas para facilitar el aprendizaje en el ambiente de trabajo del software InControl. Este será un tutorial interactivo donde se le enseñará al estudiante o la persona que reciba el curso, en forma fácil y rápida, el manejo de este software y cómo desarrollar sistemas de control para cualquier sistema o planta. También es necesario que tenga un sistema de navegación sencillo y asequible en todo momento para que sirva como medio de consulta mientras se trabaja sobre el software InControl.

Algunos de los ejercicios dentro del tutorial pueden ser explicados sin necesidad de que el usuario tenga interactividad con los elementos de la pantalla de trabajo, así que se muestran solo visualmente a través de películas.

2.1.2.1 Herramientas a utilizar. Se requieren de herramientas que permitan crear un entorno dinámico, por tal motivo se hará uso del programa **FrontPage** que cumple con ésta condición y del programa **JavaScript** que le aporta mayor

dinamismo a las páginas, gracias a que ofrece las características básicas de un lenguaje orientado a objetos. En el tutorial también deben existir elementos que reaccionen a un número limitado de acciones, éstos son los ejemplos que acompañan a cada uno de los temas del contenido y en el cual se hará uso del programa **Flash Macromedia 5**.

El tratamiento de las imágenes (cambios en color, forma, contorno, tamaño y posición en la pantalla y depurar detalles) puede realizarse con la herramienta de dibujo **Paint** y para poder cambiar las propiedades de las imágenes una y otra vez manteniendo la claridad y nitidez originales se hará uso de la herramienta **CorelDraw 9**.

La elaboración de las películas se diseñarán empleando la herramienta **Corel PhotoPaint 9** para capturar las imágenes de la pantalla y el **Photoshop 4.0** para insertar textos.

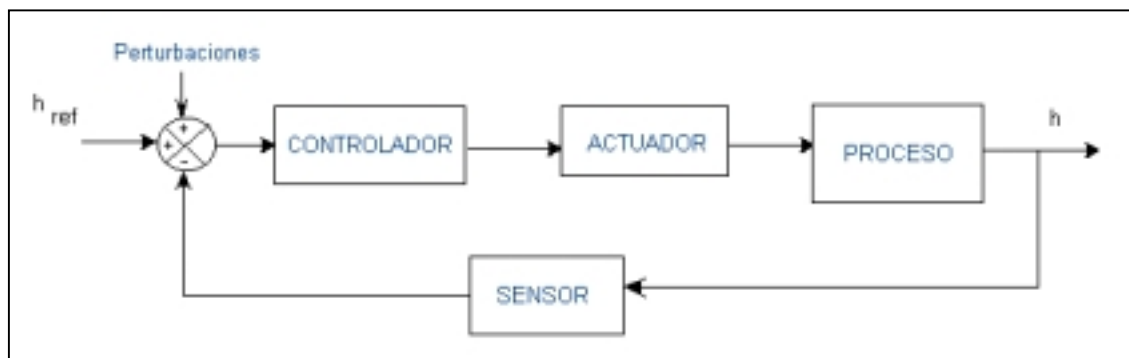
2.2 CONSIDERACIONES Y REQUERIMIENTOS DE DISEÑO DE LA APLICACIÓN EN EL SOFTWARE INCONTROL

2.2.1 Descripción del banco Control de Nivel existente. El banco o planta sobre el cual se desea implementar el software InControl posee un sistema de control como lo muestra la figura 18, con el programa Genie como controlador.

Es necesario conocer su diseño y cada uno de los dispositivos que los conforman para instalar el nuevo controlador, software InControl.

2.2.1.1 Sistema Control de Nivel. El banco “Control de Nivel” existente presenta un sistema de realimentación (Figura 18) para el control de nivel de un tanque, el cual se mantendrá para la aplicación del InControl.

Figura 18. Diagrama de bloques del sistema



En este sistema se tiene:

- **Proceso:** tanques en cascada
- **Variable controlada:** nivel de agua del tanque 1.
- **Actuador:** válvula solenoide proporcional.
- **Controlador:** computador + tarjeta de adquisición de datos.
- **Perturbación:** válvula manual de by-pass.
- **Sensor:** transmisor de presión diferencial.

2.2.1.2 Estrategias de control. Se tienen dos configuraciones, el control de nivel de un solo tanque (sistema de primer orden) y control de nivel de dos tanques en cascada (sistema de segundo orden).

Un sensor de nivel detecta la variable controlada (altura) y la convierte en una señal de corriente de 4 a 20 mA. Esta señal es acondicionada para enviarla a través de la entrada análoga de la tarjeta de adquisición de datos PLC-818 al computador. Existe un programa de control realizado en el software de adquisición de datos, Genie, el cual compara la señal de salida con la referencia introducida por el operador y genera una señal de error.

La señal de error es procesada dependiendo del modo de control utilizado y se usa para corregir la desviación de la variable controlada del punto de consigna. A su vez la salida del controlador se envía a través de la salida análoga de la tarjeta de adquisición de datos a la válvula solenoide proporcional, que actúa abriendo o cerrando para mantener la variable controlada en el set-point.

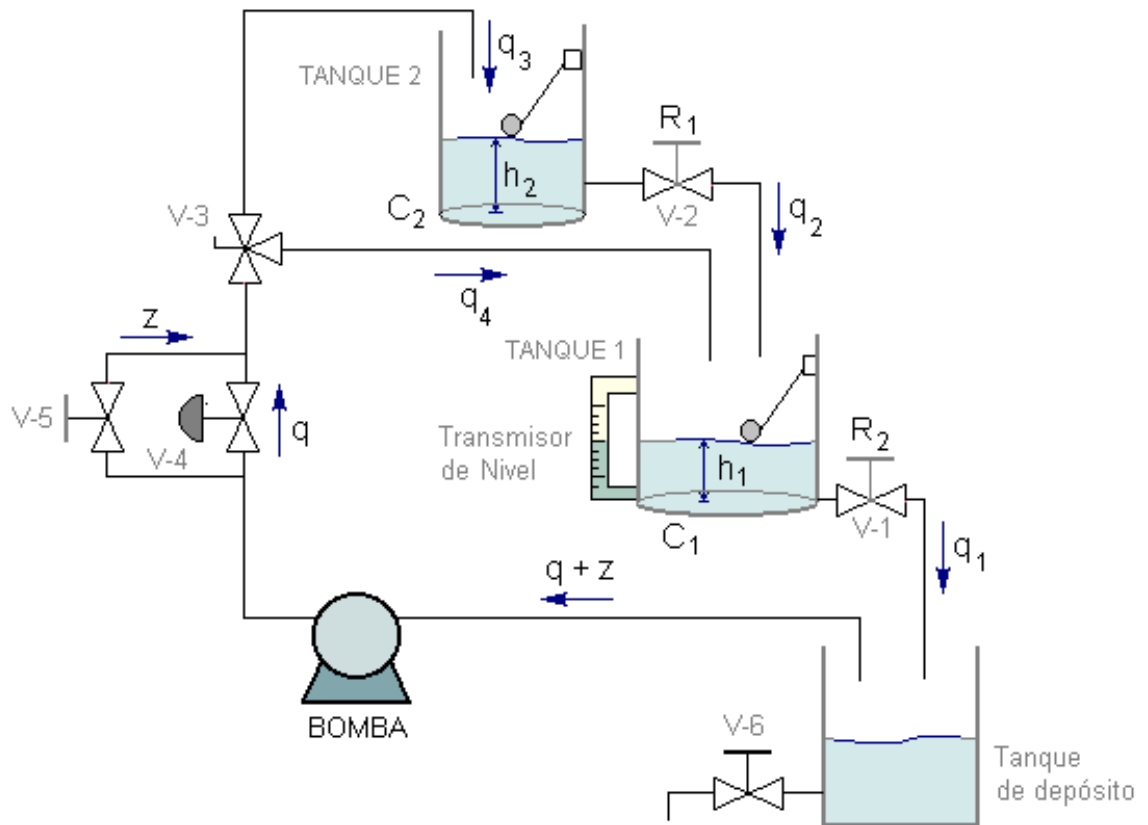
En el sistema se contempla como medida de seguridad del proceso, un interruptor de nivel en cada tanque que envía una señal de detención a la bomba en caso de alcanzar un nivel muy alto de líquido en el tanque.

2.2.1.3 Descripción del sistema de control. El diagrama de la figura 19 de la siguiente página, muestra el diagrama esquemático del banco control de nivel de un depósito de agua.

El nivel de agua que se controla es el del *tanque 1*. Seguidamente está ubicado un transmisor de precisión diferencial inteligente (ROSEMOUNT), el cual se encarga de sensar el nivel de agua del tanque 1. Esta señal sensada la convierte en una señal de corriente, para luego ser enviada al computador o controlador, el cual compara dicha señal con la señal deseada, arrojando un error (si existe) y corrigiéndolo, al enviar una señal a la *válvula proporcional V-4*, compuesta por una bobina solenoide que permite el paso o bloqueo del líquido, con el fin de mantener el nivel requerido o set-point.

El agua es transportada del depósito a los tanques, utilizando una bomba hidráulica de succión, la cual es accionada por un motor eléctrico de 0.5 HP. En el proceso también es necesaria la utilización de válvulas manuales para permitir la entrada o salida de agua en los tanques.

Figura 19. Diagrama esquemático del sistema

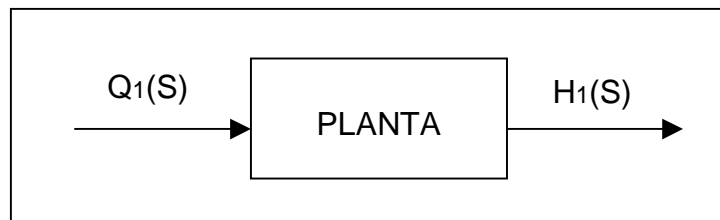


Donde:

- V-1: Válvula de Cierre Manual (esférica).
- V-2: Válvula de Cierre Manual(esférica)
- V-3: Válvula manual de 3 Vías.
- V-4: Válvula de Control Proporcional.
- V-5: Válvula manual de By-Pass (perturbaciones).
- V-6: Válvula Cheque (antiretorno).

2.2.1.4 Modelamiento de la planta. La válvula V-3 permite que el agua se dirija al tanque 1 o al tanque 2, de esta forma tenemos dos sistemas: Control de nivel de un solo tanque (sistema de primer orden) y control de nivel de dos tanques en cascada (sistema de segundo orden). A continuación se determina la función de transferencia de la planta para cada sistema, $H_1(s) / Q_1(s)$ (Figura 20).

Figura 20. Entrada y salida de la planta



Para el **Control de nivel de un solo tanque (sistema de primer orden)**, según la ecuación de la capacitancia y resistencia para el tanque 1 (Figura 19) tenemos:

$$q + z - q_1 = C_1 \frac{dh_1}{dt} \quad (1)$$

$$R = \frac{h_1}{q_1} \quad (2)$$

Transformada

$$Q(s) + Z(s) - Q_1(s) = sC_1 H_1(s) \quad (3)$$

$$Q_1(s) = \frac{H_1(s)}{R_1} \quad (4)$$

Remplazando (4) en (3)

$$Q(s) + Z(s) - \frac{H_1(s)}{R_1} = sC_1H_1(s)$$

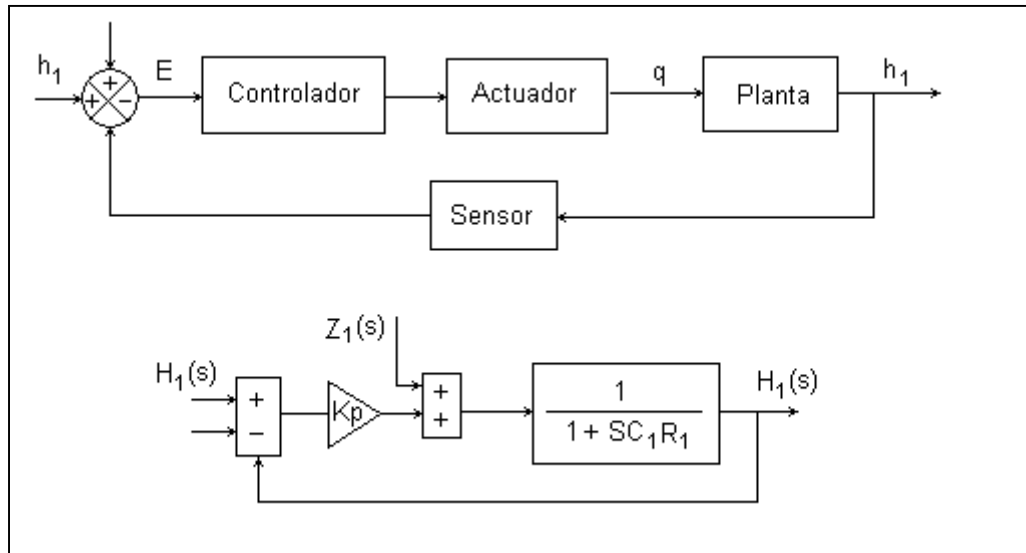
$$Q(s) + Z(s) = \frac{H_1(s)}{R_1} + sC_1H_1(s)$$

$$H_1(s) \left(\frac{1}{R_1} + sC_1 \right) = Q(s) + Z(s)$$

$$H_1(s) \left(\frac{1 + sC_1R_1}{R_1} \right) = Q(s) + Z(s)$$

$$H_1(s) = (Q(s) + Z(s)) \left(\frac{R_1}{1 + sC_1R_1} \right)$$

Figura 21. Diagrama de control para el sistema de primer orden



Control de nivel de dos tanques en cascada (sistema de segundo orden):

Ecuaciones para el tanque 1(Figura 19):

$$q_2 - q_1 = C_1 \frac{dh_1}{dt} \quad (1)$$

$$q_1 \frac{h_1}{R_1} \quad (2)$$

Ecuaciones para el tanque 2 (Figura 19):

$$q + z - q_2 = C_2 \frac{dh_2}{dt} \quad (3)$$

$$q_2 = \frac{h_2}{R_2} \quad (4)$$

Transformada:

$$Q_2(s) - Q_1(s) = sC_1H_1(s) \quad (5)$$

$$Q_1(s) = \frac{H_1(s)}{R_1} \quad (6)$$

$$Q(s) - Z(s) - Q_2(s) = sC_2H_2(s) \quad (7)$$

$$Q_2(s) = \frac{H_2(s)}{R_2} \quad (8)$$

Despejando $Q_2(s)$ de la ecuación (5):

$$Q_2(s) = Q_1(s) + sC_1H_1(s) \quad (9)$$

Despejando $Q_2(s)$ de (7)

$$Q(s) + Z(s) - Q_2(s) = sC_2R_2Q_2(s)$$

$$Q(s) + Z(s) = Q_2(s) + sC_2R_2Q_2(s)$$

$$Q_2(s)(1 + sC_2R_2) = Q(s) + Z(s)$$

$$Q_2(s) = \frac{Q(s) + Z(s)}{1 + sC_2R_2} \quad (10)$$

Igualando las ecuaciones (9) y (10):

$$Q_1(s) + sC_1H_1(s) = \frac{Q(s) + Z(s)}{1 + sC_2R_2}$$

$$\frac{H_1(s)}{R_1} + sC_1H_1(s) = \frac{Q(s) + Z(s)}{1 + sC_2R_2}$$

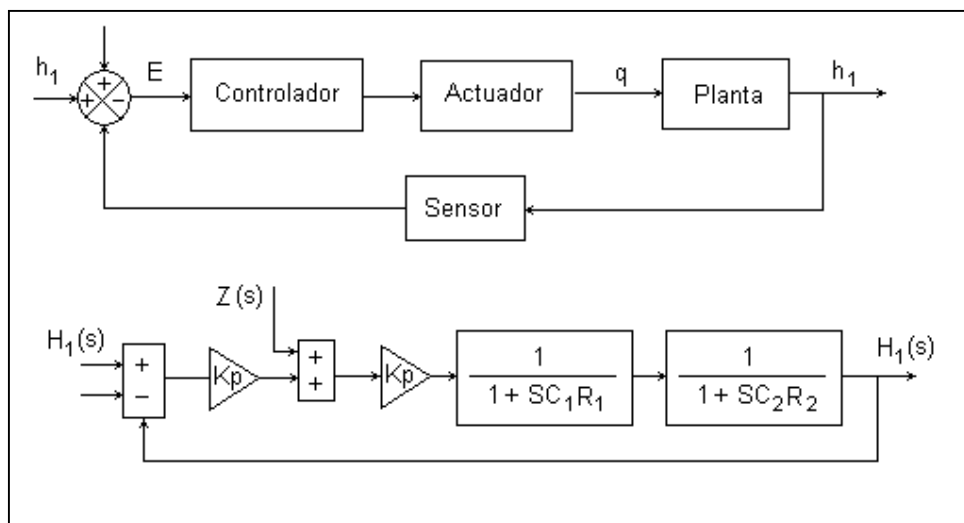
Despejando $H_1(s)$:

$$H_1(s) \left(\frac{1}{R_1} + sC_1 \right) = \frac{Q(s) + Z(s)}{1 + sC_2R_2}$$

$$H_1(s) \left(\frac{1 + sC_1R_1}{R_1} \right) = \frac{Q(s) + Z(s)}{1 + sC_2R_2}$$

$$H_1(s) = (Q(s) + Z(s))R_1 \left(\frac{1}{(1 + sC_1R_1)(1 + sC_2R_2)} \right)$$

Figura 22. Diagrama de control para el sistema de segundo orden



2.2.1.5 Instrumentos, equipos y circuitos. En el laboratorio de control de la Tecnológica de Bolívar se encuentra el banco control de nivel compuesto de los siguiente equipos e instrumentos:

2.2.1.5.1 Estructura física. Dos tanques construidos en PVC de 8 pulgadas de diámetro y altura de 1,0 m y 0.8 m para el tanque 1 y 2 respectivamente. Un depósito de almacenamiento en lámina galvanizada.

2.2.1.5.2 Bomba hidráulica. Bomba de succión accionada por motor eléctrico marca PEDROLLO con las siguientes características:

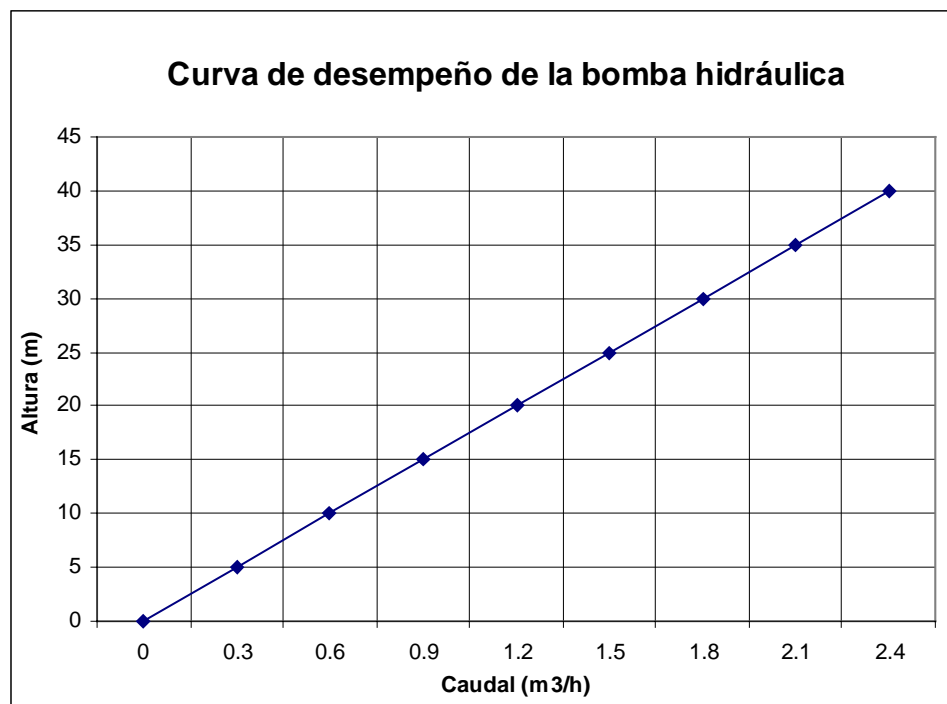
- **Rendimiento:** elevadas alturas manométricas en relación con la potencia pedida.
- **Curvas de funcionamiento:** estables, es decir, caracterizadas por pequeñas variaciones del caudal suministrado, frente a considerables variaciones de la presión pedida.
- **Elevada insensibilidad** a la presencia de aire mezclado con el fluido bombeado.
- **Potencia:** 0,5 HP (0,35 Kw), monofásico.

La tabla de la siguiente página muestra la caracterización (Tabla 2) de la bomba según fábrica y su curva de desempeño (Figura 23):

Tabla 2. Características de la bomba hidráulica.

CAUDAL (M ³)	0	0.3	0.6	0.9	1.2	1.5	1.8	2.1	2.4
ALTURA (M)	0	5	10	15	20	25	30	35	40

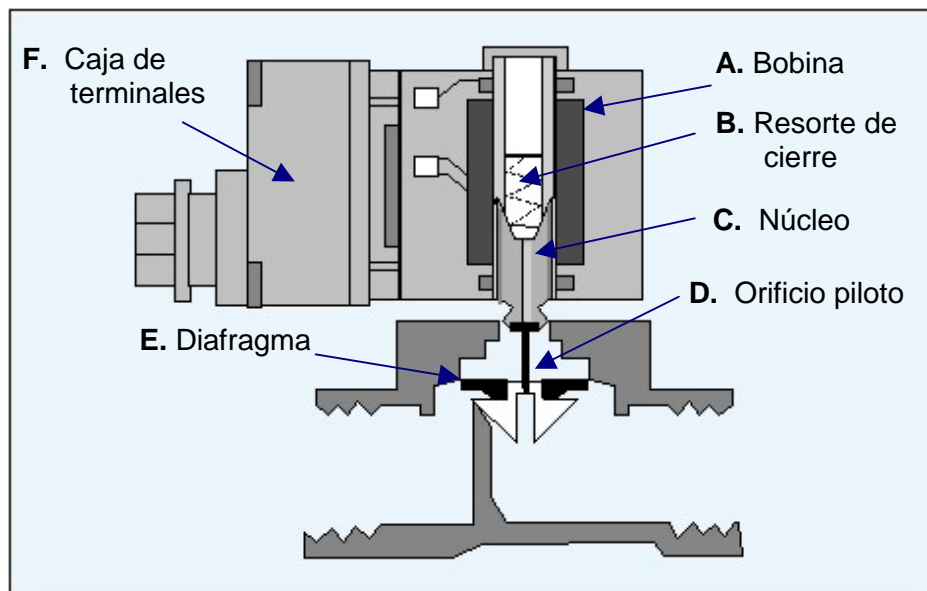
Figura 23. Curva de desempeño de la bomba hidráulica.



2.2.1.5.3 Válvula proporcional servoactuada. Es una válvula de dos vías para líquidos neutros. La regulación proporcional de la misma, para la apertura y el cierre, se logra mediante la regulación progresiva de la corriente que absorbe la bobina, y como consecuencia de la fuerza de tracción de la bobina solenoide. Al

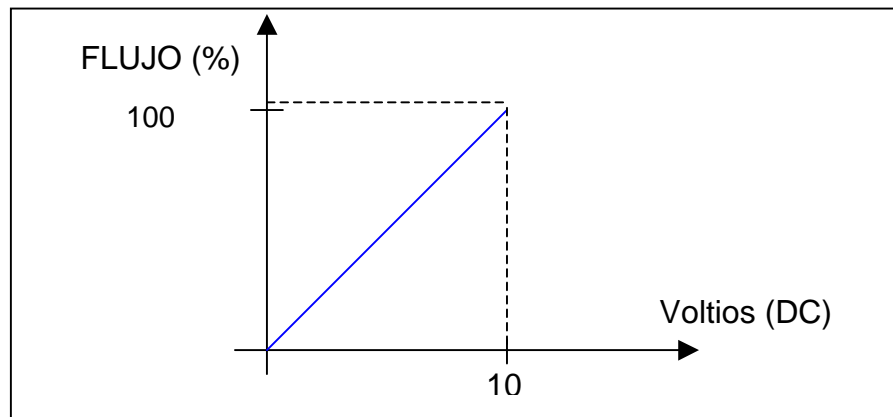
aumentar la corriente de la bobina a partir de cierto punto, la fuerza de tracción de la bobina (A) será mayor que la fuerza opuesta al resorte de cierre (B). El núcleo (C) se eleva y abre el orificio piloto (D) en el diafragma (E), que debido al efecto servo sigue moviendo el núcleo. Cuando la corriente de la bobina alcanza su valor máximo la válvula está completamente abierta (Figura 24).

Figura 24. Válvula de control proporcional.



Para la **señal de acción** se encuentra implementado un convertidor de señales (para el caudal) y una señal piloto de 0-10 voltios. La relación entre la señal piloto y el caudal es directamente proporcional (Figura 25).

Figura 25. Señal de acción de la válvula proporcional



La válvula tiene una conexión de $\frac{1}{2}$ pulgada y un $K_v = 2,1 \text{ m}^3/\text{h}$, la presión diferencial permitida es de 0,5 bares (mínimo) y de 10 bares (máximo), la tensión de alimentación puede variar entre 21 y 30 voltios D.C y la señal piloto puede variar entre 0 y 10 voltios. Puede ser utilizada en plantas industriales para regulación progresiva del caudal.

Los siguientes puntos son las características de funcionamiento de la válvula:

- Tiempo de acción es corto.
- Características lineales a lo largo del campo de regulación.
- Es normalmente cerrada, es decir, se cierra en caso de interrupción de la energía.
- Tensión de alimentación de 24 voltios DC.
- Maneja una señal piloto estándar: 4 – 20 mA o 0- 10 voltios DC.

- Puede usarse en agua, aceite y otros líquidos neutros similares.
- Régimen de trabajo continuo.
- Temperatura ambiente entre 25 y 50 °C.
- Cuerpo de la válvula en bronce, el resto en acero inoxidable.

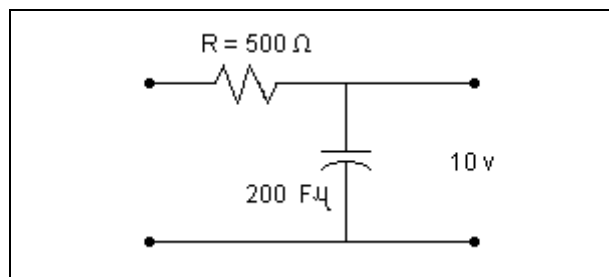
2.2.1.5.4 Transmisor de nivel. Los transmisores son instrumentos que captan la variable del proceso y la transmiten a distancia a un elemento receptor, indicador, registrador, controlador o combinación de éstos. Los transmisores pueden manejar varios tipos de señales: neumática, electrónica, digitales, hidráulica y telemétrica. En cuanto a la medición del nivel está aplicado el método de presión diferencial.

El **método de presión diferencial** consiste en detectar la diferencia de presión en el fondo del líquido y en la parte superior del líquido, (ocasionada por el peso que origina el nivel del líquido). El extremo con que se detecta la presión en el fondo del líquido se conoce como el extremo de alta presión, y el que se utiliza para detectar la presión en la parte superior del líquido, como extremo de baja presión. Una vez se conoce el diferencial de presión y la densidad del líquido, se puede obtener el nivel. Par este tipo de medición existe el transmisor de presión inteligente (ROSEMOUNT) modelo 1151.

2.2.1.5.5 Circuitos. La señal generada en el transmisor de presión pasa a una tarjeta de adquisición de datos que forma parte de los equipos básicos del sistema.

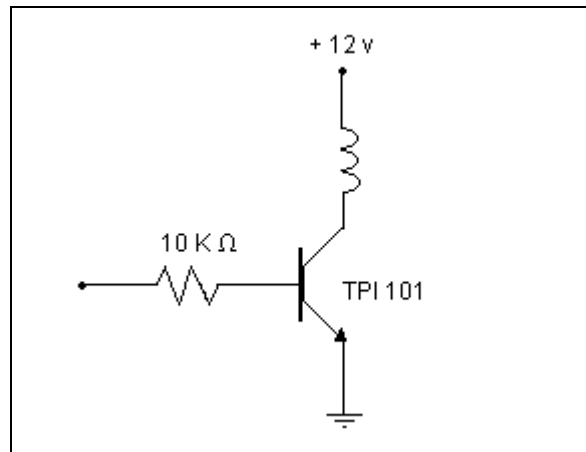
Se encuentra implementado un filtro pasabajos (Figura 26) con una frecuencia de corte igual a 60 Hz. Debido a que el transmisor entrega una señal de 4 a 20 mA y cómo la tarjeta esta configurada para recibir señales de 0 a 10 voltios, está adecuada mediante una resistencia.

Figura 26 . Circuito filtro pasabajos



Existe una amplificación (Figura 27) de la salida digital de la tarjeta, para poder manejar la señal relé que arranca, y para la bomba hidráulica ya que la tarjeta en nivel alto solo entrega 0.8 mA.

Figura 27. Circuito amplificador



2.2.2 Problemas o inconvenientes actuales del banco control de nivel. El banco control de nivel en las condiciones actuales presenta los siguientes inconvenientes:

- La tarjeta de adquisición de datos (**PCL818L**) instalada en este banco no es compatible con el *software* InControl, ya que el InControl no posee un *driver* para esta tarjeta; de hay que ésta no pueda ser instalada en un sistema de control desarrollado con este software y además no se puede establecer una comunicación directa entre el proceso y el *software* InControl, lo que quiere decir que no se pueden tomar las variables del proceso y realizar su respectivo procesamiento y hacer el respectivo control del proceso.

- Las características de la válvula están afectadas por factores tales como la caída de presión que existe en los codos, tubería y reducciones; además del comportamiento de la bomba, presentando una característica no lineal en el sistema.

2.2.3 Requerimientos y consideraciones de diseño para la aplicación. Por los mismos inconvenientes físicos del Banco se necesita de un dispositivo que permita tomar las variables del proceso y a su vez realice el control del proceso, pero principalmente que esté disponible en el laboratorio de control de la Institución.

Por lo anterior se decidió optar por utilizar el **PLC S7-300** de la marca **Siemens** en el desarrollo del proyecto. Este PLC fue adquirido por la Tecnológica de Bolívar para el laboratorio de control y posee las siguientes características muy útiles para el desarrollo de este proyecto:

- Es un dispositivo que permite adquirir y enviar las señales (análogas y digitales) del proceso para posteriormente realizar su respectivo procesamiento y visualización.

- Este es un PLC que Permite enviar y recibir datos procesados a través del protocolo de comunicación Profibus DP, pero el InControl no posee un *driver* capaz de comunicarse directamente con este PLC.

Habiendo hallado el dispositivo que permite tomar las variables del proceso, ahora se necesitaba de un dispositivo que permitiera comunicar el PLC con el software InControl; para esto se consultó la guía de drivers que aparecen en la pagina Web de Wonderware **www.Wonderware.com** y se encontró que el InControl posee un *driver* para una tarjeta que contiene un puerto de comunicación Profibus DP capaz de comunicarse con el PLC a través de este puerto, esta tarjeta es la **CIF30-DPM** de la marca **Synergetic** con un puerto Profibus DP embebido que establece comunicación con dispositivos de campo configurados como esclavos DP con base al protocolo de comunicación Profibus DP, lográndose así obtener las variables del proceso y realizar su procesamiento y el control del proceso. Esta es una de las tarjetas mas usadas para aplicaciones con el software InControl y con la cual se puede establecer una comunicación directa entre ésta y el InControl por medio del Driver *Profibus/Hilscher* de éste.

Para la conexión entre el PLC y la tarjeta CIF30-DPM, el componente de media de red Profibus es una línea de transmisión balanceada correspondiente con el estándar **EIA RS-485**, es decir un cable par trenzado categoría 5 definido por la EIA/TIA/ANSI 568, terminado en ambos extremos, en este caso para aplicaciones

Profibus DP es conveniente usar tipos de línea A (Cuadro 6) para la especificación RS-485.

Cuadro 6. Segmentos de bus y cable de derivación para el estándar Profibus.

Características	Segmentos de bus y cables de derivación			
	Requisitos de línea A	Requisitos de línea B	Capacidad total	
Impedancia	135 – 165 Ω (3-29 MHz)	100 – 130 Ω ($f > 100$ Hz)		
Capacitancia	< 20 pf / m	< 60 pF/m		
Resistencia	< 110 Ω / Km	-		
Calibre del cable	> 0.64 mm	> 0.53 mm		
Área de conductor	> 0.34 mm ²	> 0.22 mm ²		
Longitud máx 1 con vel. en baudios (bits/s) de	≤ 19.2 K	1200 m	1200 m ²	≤ 15 nf ²
	93.75 K	1200 m	1200 m ²	≤ 3 nf ²
	187.5 K	1000 m	600 m ²	≤ 1 nf ²
	500 K	400 m	200 m ²	≤ 0.6 nf ²
	1.5 M	200 m	NA	≤ 0.2 nf ²
.6 y 12M	100 m	NA	≤ 0.2 nf ² sin cables de derivación.	
NA : no aplicable				
¹ Si usa una combinación de ambos tipos de línea, dividida entre dos longitudes mostradas.				
² Esta es la suma de todos los segmentos de bus y longitudes de cable de derivación.				

Otra consideración importante al momento de diseñar la estrategia de control es que debido a la limitación de la abertura de la válvula, que introduce una no-linealidad de saturación al lazo de control, y por ser este un sistema de respuesta rápida es conveniente aplicar los métodos de sintonización de controladores PID de *Ziegler-Nichols* para un control PI, ya que este se aplica a procesos rápidos tales como: nivel, presión, caudal, etc.

2.2.4 Requerimientos y consideraciones de diseño para el software de supervisión. El InControl es un software que hace el papel de un controlador: recibe datos, los procesa y genera señales. Su desventaja es la limitada visualización que brinda, pues aunque posee FOEs (Editor de Objetos de fábrica) como los PID y Alarmas Análogas que muestran señales, así como también posee una ventana de observación de variables, no permite crear animaciones que se asemejen a la realidad del proceso controlado. El InTouch es un componente de la familia FactorySuite2000 al igual que el *software* InControl, compatibles entre sí y además comparten información bajo una misma base de datos. Por tal razón el monitoreo del sistema se hará bajo el software InTouch.

Gracias a las herramientas gráficas y al banco de objetos gráficos de fábrica que posee el InTouch, es posible crear un escenario muy parecido al banco control de nivel existente, en el cual se distinguirán cada uno de los elementos, valores y modificaciones de nivel.

3. DISEÑO E IMPLEMENTACIÓN

3.1 DISEÑO E IMPLEMENTACIÓN DEL CURSO DE CAPACITACIÓN EN EL SOFTWARE INCONTROL.

El curso de capacitación está compuesto por:

- **Material audiovisual:** diapositivas hechas en PowerPoint (35).
- **Manual de usuario:** Manual de Usuario del Software InControl.
- **Tutorial de manejo:** Tutorial de Manejo del Software InControl contenido en un CD con su respectivo Manual de Manejo..
- **Prácticas de laboratorio:**
- **Diseño de una aplicación en tiempo real:** aplicación sobre el banco Control de Nivel que se encuentra en los laboratorios de la Universidad.

El curso va dirigido a operadores, ingenieros de planta, administradores, Integradores, diseñadores de sistemas, estudiantes y profesores de la Institución Universitaria Tecnológica de Bolívar que quieran implementar o utilizar InControl en el diseño de sus procesos o como parte de la solución de un sistema de control. Su intensidad horaria es de 8 horas al día por dos días.

Este curso de dos días ayudará a los estudiantes a entender las funciones del InControl para crear soluciones de control y aplicaciones para una variedad de procesos de automatización, también para diseñar, programar e implementar con mucha exactitud aplicaciones de control por computador utilizando los lenguajes de programación disponibles en InControl 7.1.

Los ejercicios en clase guiarán al estudiante a través de los ambientes de desarrollo y ejecución del InControl, incluyendo la ejecución del Runtime Engine (Software de Ejecución). Los estudiantes crearán programas de aplicación usando Lógica en escalera conmutada (RLL), Cartas de Funciones Secuenciales (SFC), Texto Estructurado (ST), y Objetos de Fabrica (FO). El curso también cubre todos los tipos de variables globales y locales de InControl, pre-configuración de Objetos de Fabrica (Controles ActiveX), configuración de tarjetas de exploración o análisis I/O, y uso de dispositivos de redes para control de datos en tiempo real. Una vez terminado, los estudiantes aprenderán como se interconecta el InControl con Wonderware® InTouch™ y además serán capaces de configurar y desarrollar un verdadero sistema de control por computador dentro del ambiente Windows NT.

3.1.1 Audiencia. Operadores e ingenieros de planta, administradores, integradores y diseñadores de sistema y otras personas que quieran implementar o utilizar InControl en el diseño de sus procesos o como parte de la solución de un sistema de control.

3.1.2 Objetivos. Una vez terminado este curso, los participantes serán capaces de:

- Entender las funciones InControl para crear soluciones de control y aplicaciones para una variedad de áreas de automatización.
- Comprender los conceptos básicos de la estructura de las soluciones de automatización para su fácil manejo y corrección de errores.
- Saber trabajar en el ambiente de desarrollo y ejecución InControl.
- Utilizar completamente las herramientas de manejo de proyectos suministradas por InControl para manejar sus soluciones de automatización.
- Crear programas de aplicación usando Lógica en Escalera Conmutada (RLL-Relay Ladder Logic), Cartas de Funciones Secuenciales(SFC-Sequential Function Chart), y Lenguaje de Texto Estructurado(STL-Structured Text Language).
- Definir todos los tipos de variables globales y locales de InControl, incluyendo los tipos de datos definidos por el usuario.
- Trabajar con la variedad de Objetos de Fabrica (FOE) pre-configurados (Controles ActiveX) que usted puede usar en sus proyectos.

- Comprender las técnicas generales para configurar las tarjetas de exploración I/O y usar los dispositivos de red para el control de datos en tiempo real.
- Configurar las características de ejecución del Runtime Engine(Software de ejecución) de InControl.
- Implementar las funciones de monitoreo y corrección de errores del sistema InControl.
- Entender como el InControl se interconecta con InTouch.

3.1.3 Prerrequisitos.

- Conocimiento de los sistemas operativos Microsoft Windows NT y Windows 2000.
- Experiencia en el diseño Industrial.
- Familiaridad con los sistemas de control y sus respectivos lenguajes programación.
- Familiaridad con los estilos populares de PLC, redes I/O y dispositivos.

3.1.4 Programa del curso.

➤ DIA 1(Intensidad horaria = 8 horas)

Modulo 1 – Que es InControl?

Sección 1 - Introducción

- Aplicaciones InControl
- ventajas de la Arquitectura abierta
- Aplicaciones de Control

Sección 2 - Configuración I/O

- Drivers I/O
- Driver I/O - Servidor de Comunicaciones I/O

Modulo 2 - Exploración del InControl

Sección 1 - Requerimientos del Sistema InControl

- Requerimientos del Sistema InControl
- Consideraciones de la Instalación
- Notas de Instalación
- Procedimiento de Instalación

Sección 2 – ingresar al InControl

- La Ventana de Desarrollo y Ejecución
- La Barra de herramientas Standar y Runtime
- Funcionalidad de la Barra de Menú
- Ventana de Observación (Watch Window)
- Ventana de Proyectos (Project Window)

Modulo 3 - Creando, Descargando y Ejecutando un Programa o Proyecto

Sección 1 - Creando, Modificando, y Removiendo un Proyecto

- Consideraciones InControl
- Creando un Proyecto
- Modificando la Lista de Directorio

Sección 2 – Validando, Descargando, y Ejecutando

- *Validando, Descargando, y Ejecutando*
- Validando un Programa o Proyecto
- Descargando un Programa o Proyecto
- Ejecutando un Programa o Proyecto
- Detención de un Programa o Proyecto

Sección 3 – Configuración del Runtime Engine

- Consideraciones del Ambiente de Desarrollo
- Configurando el Runtime Engine
- Elementos del cuadro de dialogo “Runtime Engine Properties”

Sección 4 - Configuraciones I/O(Entradas / Salidas)

- Configuraciones I/O
- Cuadro de dialogo de configuración de drivers

Modulo 4 -- Symbol Manager

Sección 1 – Ingresando al Symbol Manager

- Symbol Manager

Sección 2 – Definiendo Variables

- Acerca de Variables (Símbolos)
- Variables Locales
- Variables Globales
- Tipos de Datos de las Variables
- Variables de Sistema General
- Variables del Runtime Engine
- Arreglos de Variables
- Variables User-Defined (definidas por usuario)

Sección 3 – Watch/Force Symbols

- Ventana de observación (Watch/Force Symbols Window)
- Barra de herramientas de ventana de observación
- Adicionando Variables a la ventana de observación (Watch Window)
- Modificando Valores de Variables
- * *Lab 1 – Creación de un Proyecto*
- * *Lab 2 – Creación de Variables (Símbolos)*

➤ **DIA 2 (Intensidad horaria = 8 horas)**

Modulo 5 – Lógica en escalera conmutada-RLL (Relay Ladder Logic)

Sección 1 – Programas RLL

- Programación en RLL

Sección 2 – Barra de herramientas RLL

- Barra de herramientas RLL

Sección 3 – Contactos y Bobinas

- Contactos RLL
- Tipos de Contactos
- Bobinas RLL
- Consejos de Edición
- * *Lab 3 - Contactos / bobinas y Lógica And/Orc*
- * *Lab 4 – Temporizadores(Timers)*
- * *Lab 5 – Contadores(Counters)*
- * *Lab 6 - Bloques Matemáticos*

Sección 4 – Bloques de Funciones

- Funciones y Bloques de Función Predefinidos y User-Defined
- La Paleta de Bloques de Función
- Adición de Funciones y Bloques de Función
- Bloques de Función de operadores de bit
- Comparación de Bloques de Función
- Conversión de Bloques de Función
- Bloques de Función de Contadores /Temporizadores
- Bloques de Función de archivos
- Bloques de Función Matemáticos
- Bloques de Función de cadenas de caracteres
- Bloques de Función Trigonométricos /Logarítmicos
- Bloques de Función Trigger (disparador)

Modulo 6 – Cartas de Funciones Secuenciales-SFC (Sequential Function Chart)

Sección 1 – Cartas de Funciones Secuenciales- SFC

- Programación SFC
- Lenguaje de Programación SFC

Sección 2 – Barra de herramientas SFC

- Barra de herramientas SFC

Sección 3 – Creación de un Programa SFC

- Creación de un Programa SFC

Sección 4 – Asistente InControl dentro InTouch

- Asistente InControl
- * *Lab 7 – Adición de una Divergencia Selección*
- * *Lab 8 – Adicionando una Detención de emergencia(Emergency Stop)*
- * *Lab 9 – Conectando con InTouch*

Modulo 7 – Lenguaje de Texto Estructurado-STL (Structured Text Language)

Sección 1 – Lenguaje de Texto Estructurado

- Elementos de Texto Estructurado
- Expresiones
- Operadores

Sección 2 – Barra de herramientas de Texto Estructurado

- Barra de herramientas de Texto Estructurado

Sección 3 – Creando un Programa de Texto Estructurado

- Creación de un Programa de Texto Estructurado

Sección 4 – Código del Programa de Texto Estructurado

- Tipos de Ordenes
- Funciones y Procedimientos de llamadas
- Grupos de Funciones y Bloques funciones
- Descripción de la Sintaxis de una Función

** Lab 10 – Creación de un Programa de Texto Estructurado*

Modulo 8 – Objetos de Fabrica(Factory Objects)

Sección 1 – Objetos de Fabrica

- Objetos de Fabrica
- Integración de Objetos de Fabrica con InControl

Sección 2 – Adición y configuración de Objetos de Fabrica

- Instalación de Objetos de Fabrica
- Configuración de Objetos de Fabrica
- cuadro de Dialogo “PID Control Properties”
- Objetos de Fabrica PID

** Lab 11 – Adicionando, Configurando y ejecutando un Objeto de Fabrica*

Modulo 9 -- Seguridad

Sección 1 – Niveles de Seguridad

- Seguridad

Sección 2 – Registrando y Adicionando un Nuevo Usuario

- Registro en el sistema
- Adición de Usuarios

3.1.5 Diseño del tutorial de manejo del software InControl. Este tutorial está contenido en un CD anexo a este libro con el nombre TUTORIAL DE MANEJO DEL SOFTWARE INCONTROL, además viene acompañado de un manual de manejo donde se indica cómo iniciar el tutorial y describe las características principales de su diseño.

El principal objetivo del tutorial es ser una guía para quienes deseen aprender a manejar el software InControl y orientar en la implementación de aplicaciones de control utilizando los lenguajes de programación disponibles, junto con las herramientas que el mismo software nos proporciona.

En forma general éste tutorial de manejo consta de cinco capítulos donde se describe el manejo del *software* InControl acompañado de ejemplos con accesos directos a su entorno de trabajo. A continuación se describe el contenido de cada capítulo:

- **Introducción:** brinda información acerca de las características del InControl, de la marca Wonderware y el producto FactorySuite a la cual pertenece.
- **Configuración y Ambiente InControl:** éste capítulo describe los requerimientos de software y hardware para el InControl, instrucciones para instalar y correr el InControl, cómo conseguir iniciar la programación y la

información de soporte técnico. También describe el ambiente InControl: elementos de la barra de herramientas, menú de opciones y campos de pantalla, etc. Explica cómo configurar el sistema de seguridad para el ambiente de desarrollo y ejecución InControl. Los subcapítulos que lo componen son:

- Instalación
 - Ambiente InControl
 - Seguridad
-
- **Manejo, Creación y Organización de Proyectos:** éste capítulo define cada una de las unidades de programación que emplea el InControl. Muestra el funcionamiento y uso de los distintos editores correspondientes a los lenguajes de programación: Lógica de Escalera Conmutada (RLL), Organigrama de Funciones Secuenciales (SFC), y Lenguaje de Texto Estructurado (STL). Indica cómo definir los tipos de variables globales y locales de InControl, incluyendo los tipos de datos definidos por el usuario en la creación de sus programas. Además le enseña a trabajar con la variedad de objetos de fábrica (FOE) pre-configurados dentro de sus proyectos. Está compuesto por los siguientes temas:

- POU's
- Manejo De Proyectos

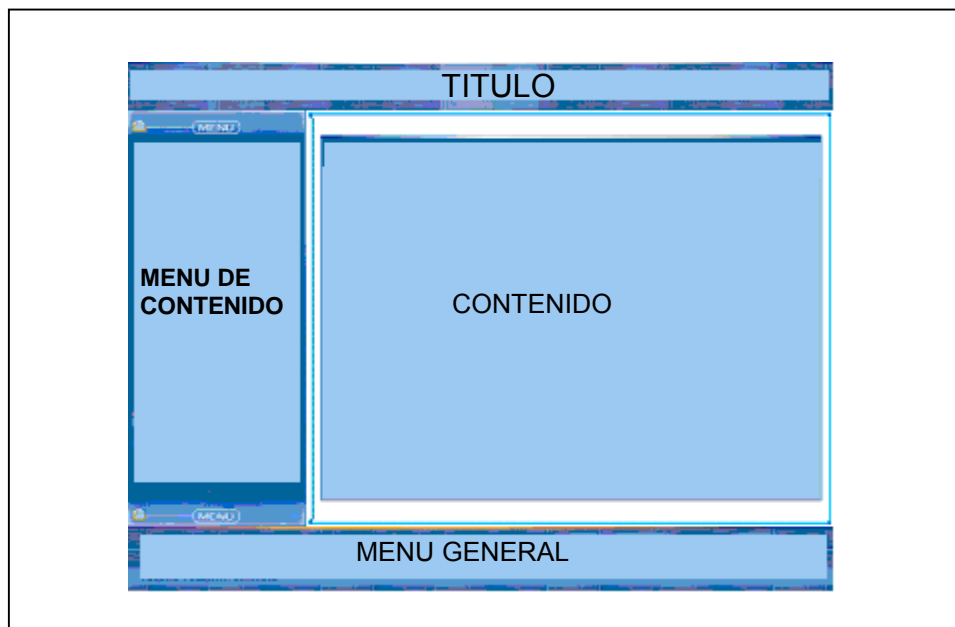
- Definición De Variables
- Creación De Programas

- **Ejecución de Proyectos:** muestra los métodos de ejecución de los proyectos y programas, e identifica los iconos que hacen parte de esta tarea.

- **Comunicaciones:** explica en forma general la configuración de tarjetas escáner I/O para la comunicación entre InControl y los equipos I/O, cómo llegar a usar un driver I/O con InControl, y el uso de wizards especiales para la comunicación InControl e InTouch. Esta compuesto por los siguientes temas:
 - Tarjetas y Drivers I/O
 - InControl e InTouch

El estilo de las páginas del tutorial consta de las siguientes secciones (Figura 28) para facilitar la navegación sobre el contenido.

Figura 28. Formato general de las páginas del tutorial

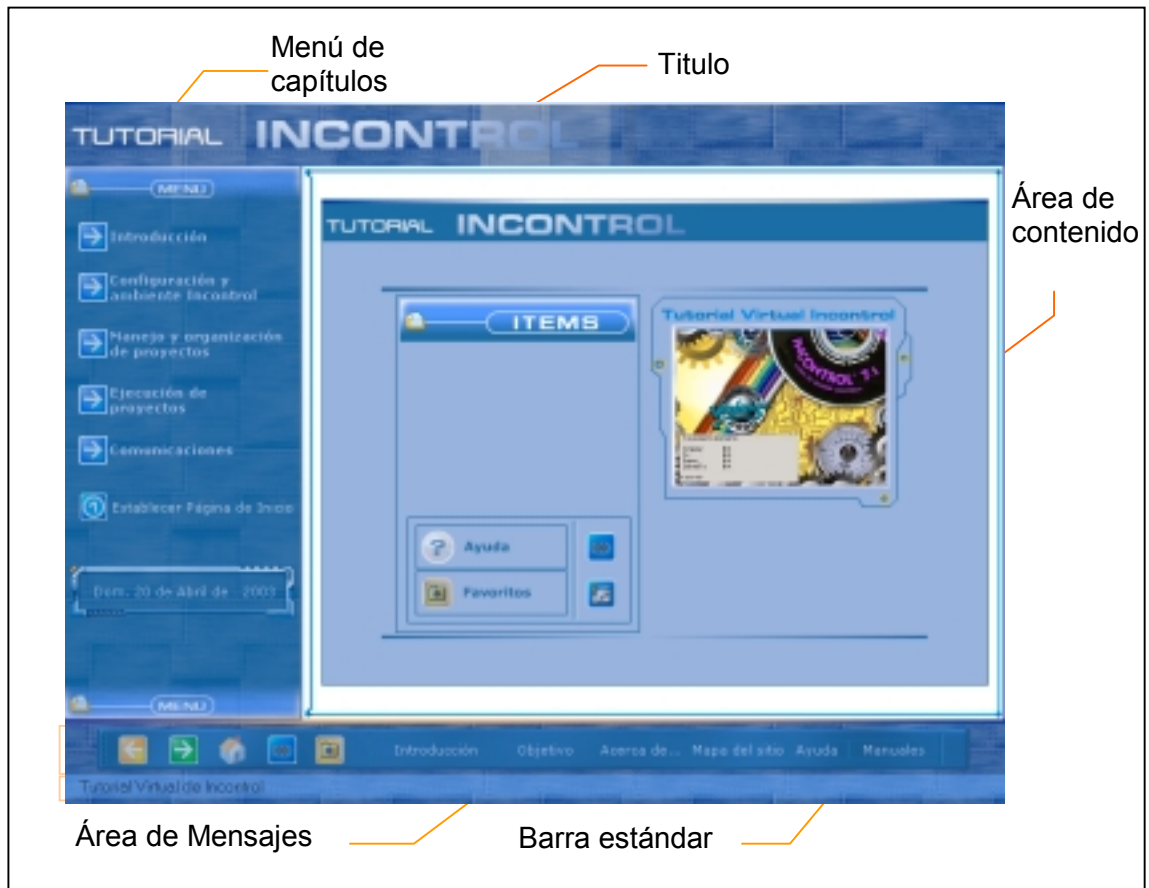


Su color principal es azul junto con otros colores complementarios como el color verde, y el anaranjado para los iconos de desplazamiento y subtítulos; se escogió el color azul debido a que es el mas utilizado en asuntos académicos, ya que es un color que representa la inteligencia, sabiduría y elegancia. El verde y el naranja son colores complementarios, y el blanco porque es un color que no cansa la vista, mas que todo si va acompañado de un gris.

En la primera página y durante todo el tutorial, la pantalla se encuentra dividida en cinco secciones: título, menú de capítulos, área del contenido, la barra estándar y área de mensajes. La siguiente gráfica presenta cada una de estas secciones.

3.1.5.1 Secciones de la pantalla principal o de inicio del Tutorial








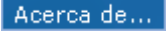
Figura 29. Pantalla de inicio del Tutorial.



- **Título:** muestra el título del programa “TUTORIAL INCONTROL”.
- **Menú de capítulos:** desde aquí tiene acceso a cada uno de los capítulos que conforman el tutorial con solo pulsar una vez sobre cada uno de los títulos. La opción “Establecer página de inicio”, le permite establecer la página actual cómo su página de inicio predeterminada. El cuadro de la parte inferior le indica la fecha actual de trabajo.

- **Área de mensajes:** si ubica el *mouse* sobre cada uno de los íconos de la pantalla, ésta área le informa de antemano la tarea que podrá realizar.
- **Barra estándar:** puede ejecutar las tareas principales de toda página web y conseguir información referente al tutorial.

Cuadro 7. Tareas de los iconos de la barra estándar.

Íconos	Tareas
	Retrocede a la página anterior.
	Avanza a la siguiente página (visitada).
	Abre la página de inicio.
Íconos	Tareas
	Abre la página de vínculos.
	Agrega la página actual a la lista de favoritos.
	Abre la ventana que contiene la introducción de este tutorial virtual.
	Abre la página de los objetivos.
	Abre la ventana referente a la producción del tutorial virtual InControl.

Mapa del sitio	Muestra el Mapa de desplazamiento de todo el contenido de los capítulos que conforman el tutorial.
Ayuda	Ingresa a la Ayuda para la óptima navegación del tutorial.
Manuales	Permite el acceso a los manuales de apoyo para usuarios y webmaster.

Otra opción para disminuir la extensión de las páginas es el uso de iconos que permiten acceder a películas y a páginas en formato PDF.

Figura 30. Iconos de Acceso



3.1.5.2 contenido del tutorial.

- INTRODUCCIÓN
 - Antecedentes
 - Wonderware
 - Acerca de incontrol
 - Características principales
 - Otras Características y Mejoras
- CONFIGURACIÓN Y AMBIENTE INCONTROL
 - INSTALACIÓN

- Requerimientos del sistema
- Procedimientos de instalación
- Procedimiento de desinstalación
- AMBIENTE INCONTROL
 - Ejecución del incontrol
 - Ventanas de desarrollo y ejecución
 - Barra de herramientas
 - Barra de menú
- SEGURIDAD
 - Niveles De Seguridad
 - Verificación del nombre
 - Registro y Cambio de Password
 - Libre de Registro
 - Manejo de la Seguridad
 - Bloqueo de algoritmos SFC
 - Seguridad windows NT
- MANEJO, CREACIÓN Y ORGANIZACIÓN DE PROYECTOS
 - POU's
 - Definición y características de los pou's.
 - Diferencias entre funciones y bloques de funciones
 - Denominación de variables.
 - MANEJO DE PROYECTOS

- Ventana de proyectos
- Creación de un proyecto
- Como remover un proyecto
- Adición de un proyecto
- Modificación del nombre y la descripción de un proyecto
- Manejo de pous
- Organización de un proyecto
- Definición de bloques de funciones
 - Ajuste de Parámetros y Variables
 - Código para Llamar un Bloque de Funciones
 - Ejemplos de Bloques de Funciones
- Definición de Funciones
 - Ajuste de Parámetros y Variables
 - Especificación del tipo de dato para el valor de retorno de una función
 - Código para Llamar una Función
- Asignación De La Plataforma De Hardware
- DEFINICIÓN DE VARIABLES
 - Características de los Nombres de Variables
 - Tipos De Variables
 - Tipos de datos de variables
 - Symbol manager
 - Creación de una variable

- Tipo de dato user-defined
- Transferencia de bases de datos de símbolos
- CREACIÓN DE PROGRAMAS
 - Introducción
 - Uso Del Editor De Objetos De Fabrica InControl
 - Definición de Objetos de Fábrica.
 - Instalación de un FOE.
 - Creación de categorías
 - Ingreso a las propiedades de los FOEs.
 - Uso del FOE PID
 - Uso del FOE de Alarmas Análogas
 - Uso del FOE de Información de Proyectos
 - Uso del editor RLLI
 - Creación de un programa RLL
 - Barra de elementos RLL
 - Adición de elementos y comentarios al programa.
 - Uso del Editor SFC
 - Creación de Programas SFC
 - Barra de herramientas SFC
 - Adición y Edición de Elementos de Programa
 - Uso del Editor STL
 - Creación de un Programa STL
 - Herramientas de Texto Estructurado

- Ingreso del Código de Programa
 - Uso del Editor De Objetos De Fabrica InControl
 - Definición de Objetos de Fábrica.
 - Instalación de un FOE.
 - Creación de categorías
 - Uso del FOE PID
 - Uso del FOE de Alarmas Análogas
 - Uso del FOE de Información de Proyectos
- EJECUCIÓN DE PROYECTOS
 - Iconos del Runtime Engine
 - Verificación de la Conexión del Nodo
 - Icono de Monitoreo del Runtime Engine
 - Símbolos y Códigos
 - Uso de los comandos del icono RTE
 - Salida y Restauración del monitor RTE
 - Validación y Descarga
- COMUNICACIONES
 - TARJETAS Y DRIVERS I/O (10 Pag)
 - Elementos de comunicación
 - Conexiones con drivers i/o
 - Uso del driver i/o con incontrol
 - Simulación i/o

- INCONTROL E INTOUCH (15 pag)
 - Recomendaciones para su uso
 - Wizards incontrol
 - Uso de dos Wizards InControl
 - Vínculos entre símbolos y etiquetas

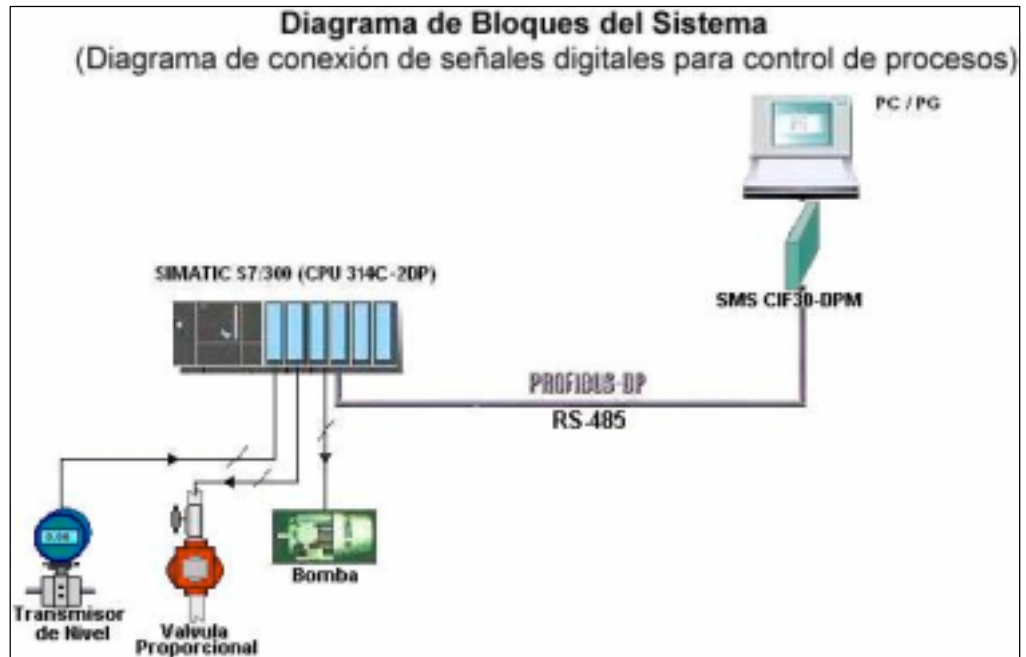
3.2 DISEÑO DE LA APLICACIÓN DEL SOFTWARE INCONTROL EN EL BANCO CONTROL DE NIVEL

La aplicación del software InControl es planteada sobre el Banco Control de Nivel ubicado en el Laboratorio de Control de la Tecnológica de Bolívar. El diseño de ésta nueva aplicación contiene con el mismo sistema de realimentación y las mismas estrategias de control (ON-OFF y PID) presentes en el banco, pero con modificaciones en su sistema de adquisición de datos porque se requieren añadir otros dispositivos capaces de comunicar al software InControl con el proceso.

Para el desarrollo de la aplicación, la estrategia de control fue programada con el software InControl y el software InTouch para supervisar y monitorear las variables del proceso.

A continuación se muestra el diagrama del sistema de conexión descentralizado de las señales de entrada y salida tanto análoga como digital de control y supervisión del proceso (Figura 31).

Figura 31. Diagrama de conexión física del sistema

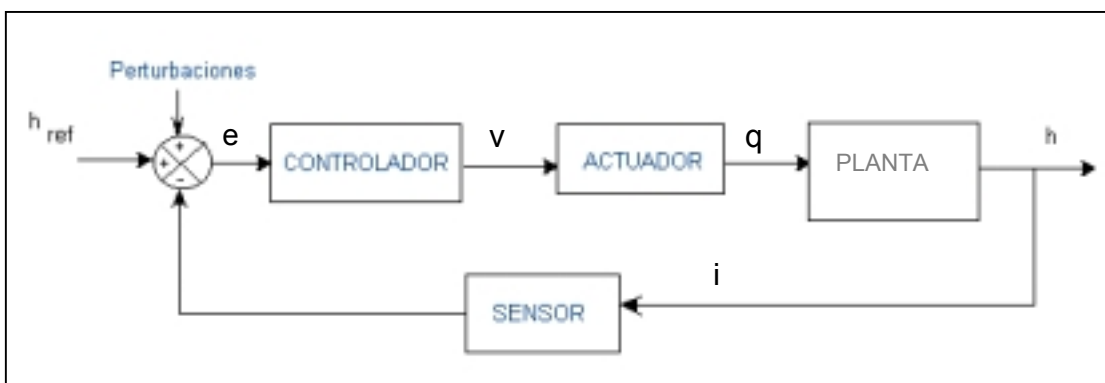


Nuestra aplicación se desarrolla a través de los siguientes pasos, el primero será el diseño de las estrategias de control (ON-OFF, PID) que se realizara en el software InControl. Para monitorear o visualizar las variables del proceso se usa **Wacth Window** (ventana de observación y monitoreo de variables de InControl) que evita el uso de un *software* supervisorio (InTouch, LabView, etc.) en aplicaciones desarrolladas con InControl, pero para hacer que nuestra aplicación fuera muy similar a los sistemas automatizados de control industrial integrado se decidió realizar la visualización y monitoreo de las variables del sistema con el *software* InTouch.

3.2.1 Sistema de control. El sistema de control (realimentación) existente en el banco, consiste en manipular un flujo de alimentación (q), ver figura 32, para regular automáticamente el nivel de agua de un tanque (h), comparando esta señal de salida (h) con una señal de referencia (h_{ref}) y así generar una señal de error el cual utiliza un controlador para corregirlo de acuerdo con una acción de control programada y enviar una respuesta (v) al actuador para que controle el flujo de alimentación (q).

Para el control de nivel de los tanques se tienen dos configuraciones: control de nivel de un solo tanque (sistema de primer orden) y control de nivel de dos tanques en cascada (sistema de segundo orden).

Figura 32. Diagrama de bloques del sistema



Donde:

- **Variable de entrada (h_{ref}):** nivel de agua deseada.

- **Variable de salida (h):** nivel de agua del tanque 1.
- **Variable controlada (q):** flujo que alimenta los tanques.
- **Señal de control (v):** voltaje de la válvula.
- **Controlador:** software *InControl*.
- **Actuador:** Válvula Proporcional Servoactuada.
- **Planta:** Tanques en cascada.
- **Sensor:** Transmisor de nivel diferencial.
- **Perturbación:** disminución del flujo de alimentación.

3.2.1.1 Descripción del nuevo sistema de control. El transmisor de nivel (Figura 32) detectará la variable controlada (**h**) y la convertirá en una señal de corriente (**i**) de 4 a 20 mA. Esta señal de corriente será recibida por el PLC S7-300 por medio de uno de sus puertos de entrada analógico, el PLC tomará ésta señal analógica y la convertirá en una señal digital y la enviará por el puerto *Profibus* DP (Figura 34) a través del segmento de bus (cable bifilar trenzado blindado) conectado al puerto *Profibus* DP de la tarjeta CIF30-PM que recibe estos datos para que el *InControl* los procese y genere una señal de salida. Esta señal de salida (voltaje (**v**)) del controlador será dirigida a través del puerto *Profibus* DP de la tarjeta CIF30-DPM por medio del segmento de bus al puerto *Profibus* DP del PLC S7-300 (Figura 34) el cual le asigna a esta un valor de voltaje (0-10 voltios), que la enviará por una de las salidas análogas a la válvula solenoide proporcional para que ésta varíe su porcentaje de apertura. El voltaje generado por el

controlador (v) modificará el cierre o apertura de la válvula proporcional y esta a su vez controlará el flujo de alimentación (q) del tanque, para mantener la variable controlada en el *setpoint*. La variación automática permite que haya un mayor o menor flujo de agua por el tubo, lo cual hace que aumente o disminuya el nivel en el tanque hasta alcanzar el nivel deseado (equilibrio de masas).

Figura 33. Diagrama esquemático del sistema de control.

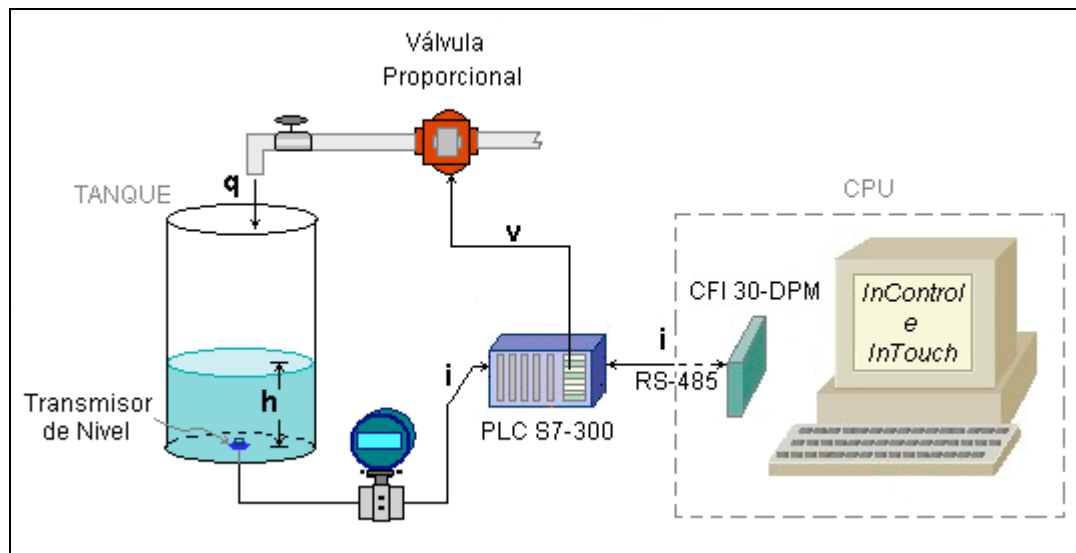
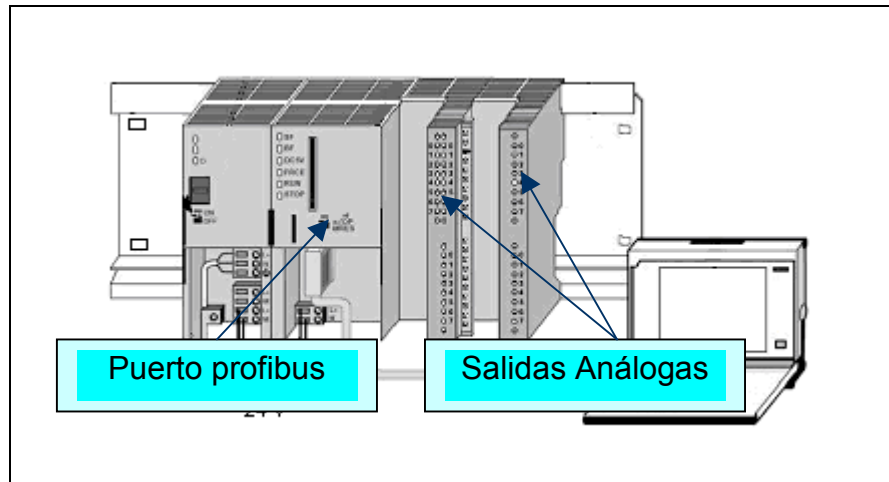


Figura 34. Partes del PLC S7-300

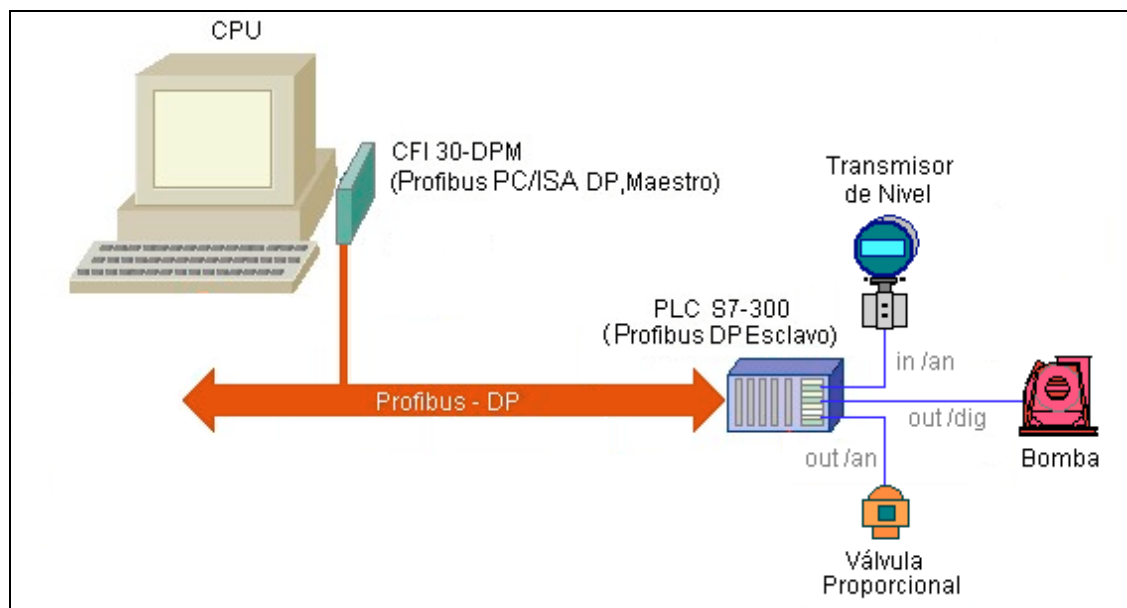


El programa de control que realizado en el *software InControl*, comparará el valor equivalente a la señal de corriente del transmisor (**h**) con la referencia introducida (**href**) por el operador (Figura 32) y generará una señal de error (**e**). La señal de error será procesada dependiendo del modo de control utilizado (PID u ON-OFF) y se usará para corregir la desviación de la variable controlada del punto de consigna. Al mismo tiempo las señales de altura del tanque (**i**), el error (**e**), y voltaje a la válvula (**v**) serán enviadas al *software InTouch* para su visualización. Desde el InTouch se consigue monitorear el encendido del sistema, seleccionar el modo de control a utilizar y determinar el nivel (**href**) de agua deseado o setpoint.

En el sistema se encuentra como medida de seguridad del proceso, un interruptor de nivel en cada tanque que envía una señal de detención a la bomba en caso de alcanzar un nivel muy alto de líquido en el tanque (nivel máximo: 80 cm). La señal

de corriente recibida por el PLC S7-300 desde el transmisor de nivel se convertirá a una señal digital y se enviará por el puerto *Profibus DP* a través del protocolo de comunicaciones *Profibus DP* (Figura 35).

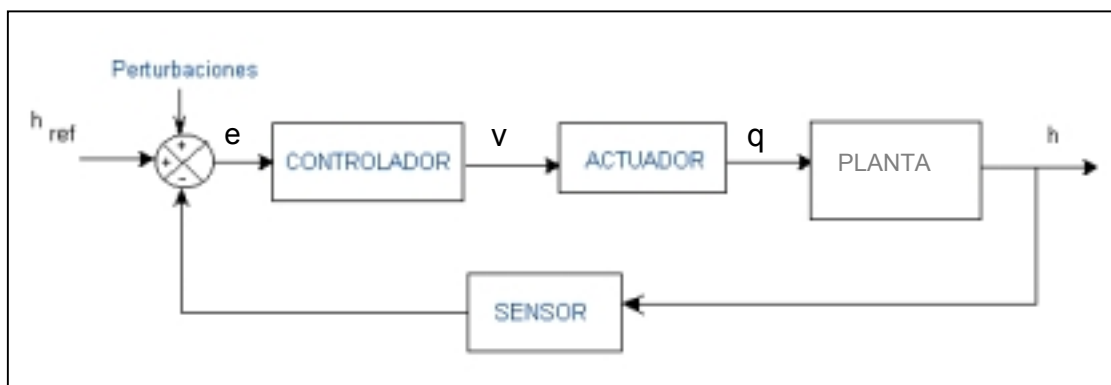
Figura 35. Comunicación Profibus DP



3.2.2 Diseño de las estrategias de control (ON-OFF, PID) en el software InControl. En las estrategias de control ON-OFF y PID diseñadas para el sistema control de nivel, es importante tener en cuenta que por ser éste un sistema de respuesta rápida es conveniente aplicar los métodos de sintonización de controladores PID de *Ziegler-Nichols* para un control PI, ya que este se aplica a procesos rápidos tales como: nivel, presión, caudal, etc.

3.2.2.1 Control de nivel en el tanque 1 y sintonización del PID. El control de nivel en el tanque 1 se realiza a través de la manipulación del caudal de agua que fluye por la válvula (Actuador). El sistema de control utilizado es el de lazo cerrado, tal como se muestra en la siguiente figura 36.

Figura 36. Sistema de control de lazo cerrado



El nivel real será medido por el transmisor de nivel (sensor) que está instalado en el banco, desde allí este envía una señal de corriente (4 a 20 mA) proporcional al nivel del tanque, al PLC este a su vez envía la información recibida a través del bus de campo establecida como una señal digital (protocolo PROFIBUS DP) al software InControl (controlador) por medio de la tarjeta CIF30-DPM.

Este a su vez compara este nivel real con el nivel deseado y si el nivel real difiere del deseado el controlador envía una señal a la válvula proporcional para que esta varíe su porcentaje de apertura. Esta variación automática permite que haya un

mayor o menor flujo de agua por el tubo, lo cual hace que aumente o disminuya el nivel en el tanque hasta alcanzar el nivel deseado (equilibrio de masas).

El controlador usado es un PID digital el cual se implementa a través del software InControl. Este PID viene incluido como una de las herramientas internas de trabajo del InControl y aparece con el nombre FOE PID por lo que no hace falta diseñar el algoritmo PID de control, solo basta con parametrizar el FOE PID y ponerlo a trabajar, este hace parte importante de los FOE (Factory Object Editor) que son controles ActiveX que trae desarrollado el InControl y que son utilizados para el desarrollo de sistemas de control industrial.

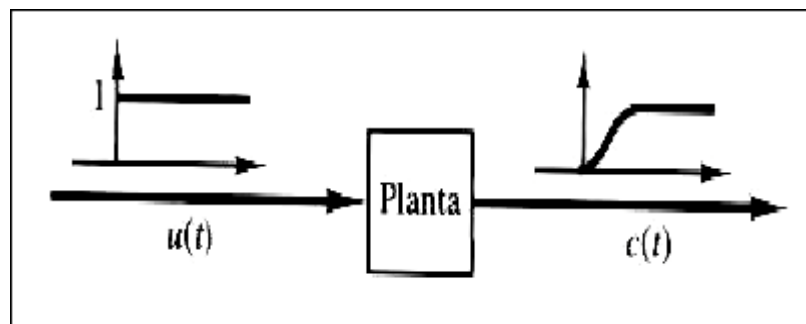
Entre los parámetros mínimos necesarios para poner a funcionar el controlador tenemos: obviamente los valores de las constantes K_p , T_i , T_d , el valor del setpoint deseado y el tiempo de muestreo.

Entre otra de las características que posee el **FOE PID** esta la posibilidad de conmutar entre modo automático y modo manual. En este último la magnitud manipulada sigue un valor ajustado manualmente. También permite la desconexión y conexión independiente de las acciones proporcional, integral y derivativa lo que permite la configuración de controladores P, PI, PID.

La sintonización del controlador PID del sistema de control de nivel se hace utilizando los métodos de sintonización de **Ziegler-Nichols**.

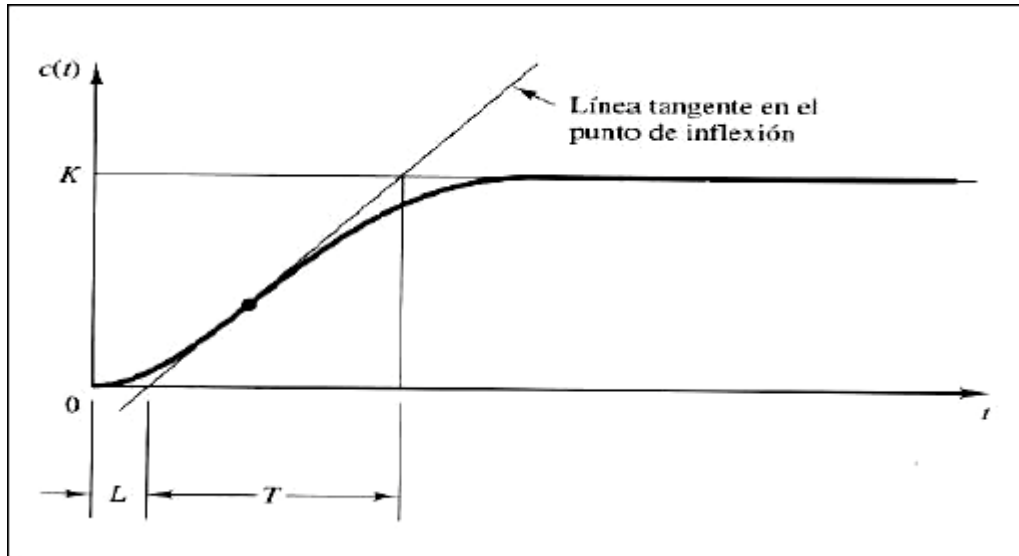
El método que se recomienda utilizar para la sintonización de los controladores PID es el de curva de reacción. En este método se aplica una entrada escalón a la planta, lo cual significa un cambio abrupto de 0 a 100% en el valor de la variable manipulada en el proceso de control y se observa la respuesta de la variable controlada, tal como se muestra en la figura 37.

Figura 37. Respuesta escalón unitario de una planta



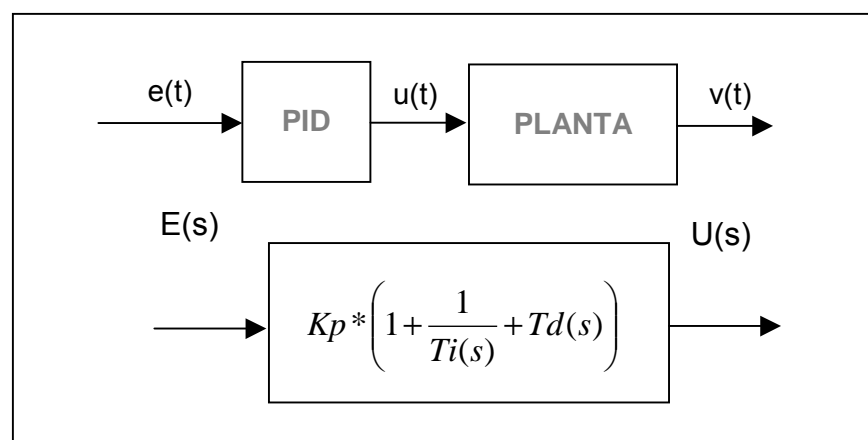
Si la respuesta de la variable controlada ante la entrada escalón tiene la forma de **S** se toma de ésta el tiempo de retardo L y la constante de tiempo T , las cuales se determinan dibujando una recta tangente en el punto de inflexión de la curva en forma de **S** y determinando las intersecciones de esta tangente con el eje de tiempo y la línea $C(t) = K$ como se muestra en al figura 38.

Figura 38. Curva de respuesta en forma de S



Luego, de acuerdo a los valores de L y T obtenidos de la curva de respuesta en forma de S, se calculan los valores los parámetros del controlador PID; K_p , T_i y T_d (Figura 39) según las formulas que aparecen en el cuadro 8.

Figura 39. Señales y parámetros del controlador PID

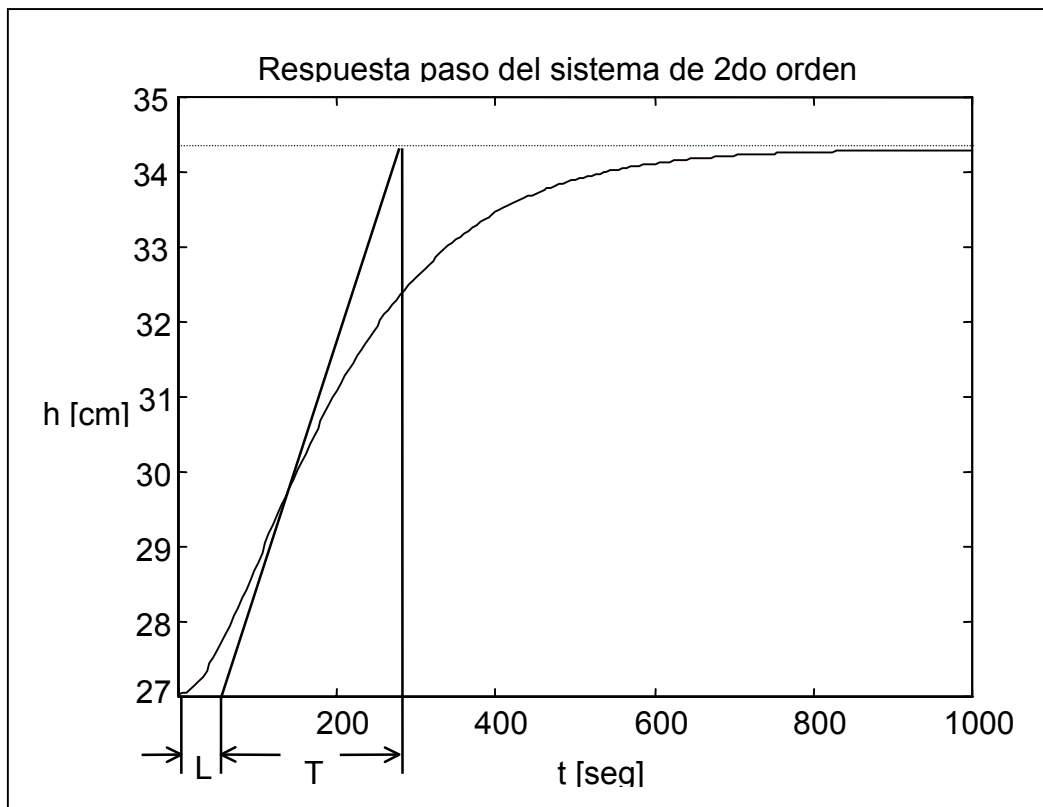


Cuadro 8. Fórmulas para calcular los parámetros del PID según Ziegler-Nichols

Tipo de Controlador	Kp	Ti	Td
PID	$1.2 \frac{T}{L}$	2L	0.5L

De acuerdo con el Método de la curva de reacción de Ziegler - Nichols, se obtuvo la respuesta ante un paso de 1V (Figura 40):

Figura 40. Curva de reacción para determinación de constantes



De la gráfica: $L = 41$ seg y $T = 280$ seg.

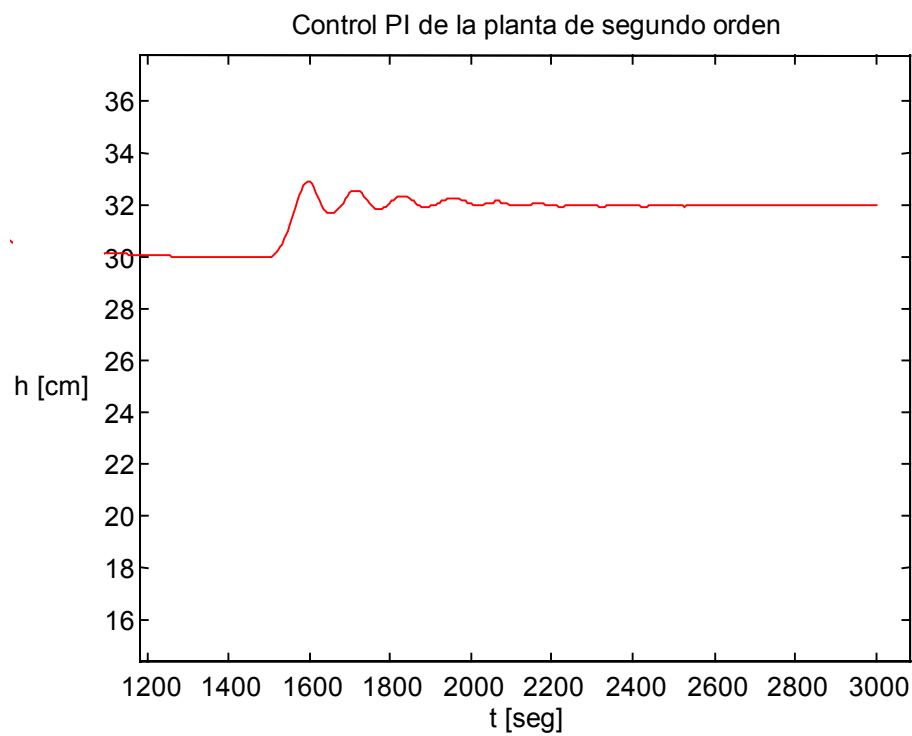
De acuerdo con las reglas de sintonización de Ziegler-Nichols para controlador PI:

$$K_p = 0.9 \cdot T/L = 6.146341463$$

$$T_i = L/0.3 = 136.6666667$$

Las constantes que se introducen en el programa de simulación Simulink y en el de control **ControlPID** de InControl son: $P = K_p = 6.14$ e $I = K_p / T_i = 0.045$. Los resultados obtenidos se muestran a continuación (Figura 41):

Figura 41. Respuesta del sistema de segundo orden con Control PI.



El paso dado a la planta es de 2 cm, debido a que para valores mayores, la válvula de control presenta saturación, en el sistema actual.

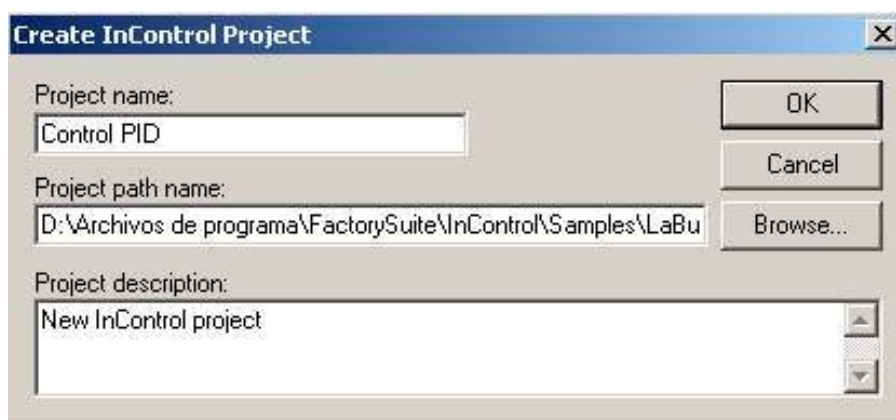
3.2.2.2 Diseño del control PID. El primer paso en el desarrollo de la estrategia de control **PID** fue crear el proyecto en el software InControl concerniente al proceso (control de nivel), lo cual será descrito en los siguientes pasos:

Nota: es necesario poseer el archivo de licencia del software InControl para poder realizar algunos de los pasos que se muestran a continuación, ya que el InControl se instala en modo demo (24 horas), presentando restricciones en los drivers I/O, y no permite la carga de los FOE (controles ActiveX) mas especialmente de los parámetros de configuración para su ejecución en aplicaciones.

Después de iniciar el InControl:

1. En el menú **File** del cuadro de diálogo **InControl Project Manager**, se hace clic en **New**. Aparece el cuadro de diálogo **Create InControl Project** (Figura 42).

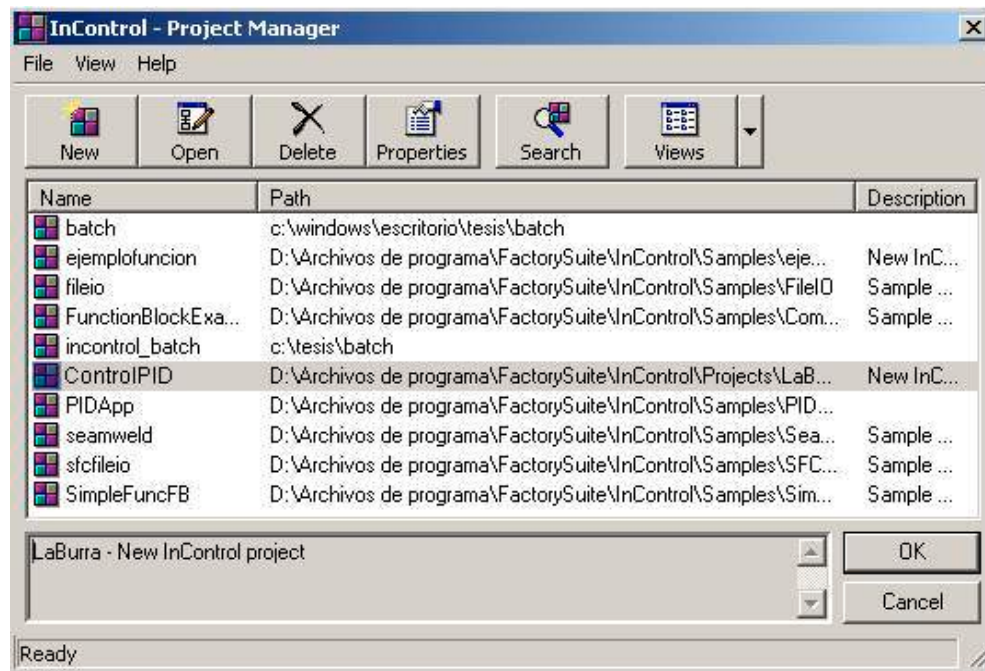
Figura 42. Creación de proyectos InControl



2. Se ingresa el nombre del proyecto (**Control PID**). Se escribe el nombre del proyecto usando las convenciones para nombres de Windows NT, se selecciona una ruta, y se pulsa **OK**. Un directorio con este nombre se crea en el disco duro y el nuevo proyecto se adiciona a la lista de proyectos de InControl. En la figura 43 de la página siguiente, el proyecto **Control PID** es creado.

El proyecto que se crea aparece en la ventana de proyectos del ambiente de desarrollo. Después de creado el proyecto, se inicia la creación de los programas necesarios para la ejecución de la estrategia de control PID.

Figura 43. InControl Project Manager

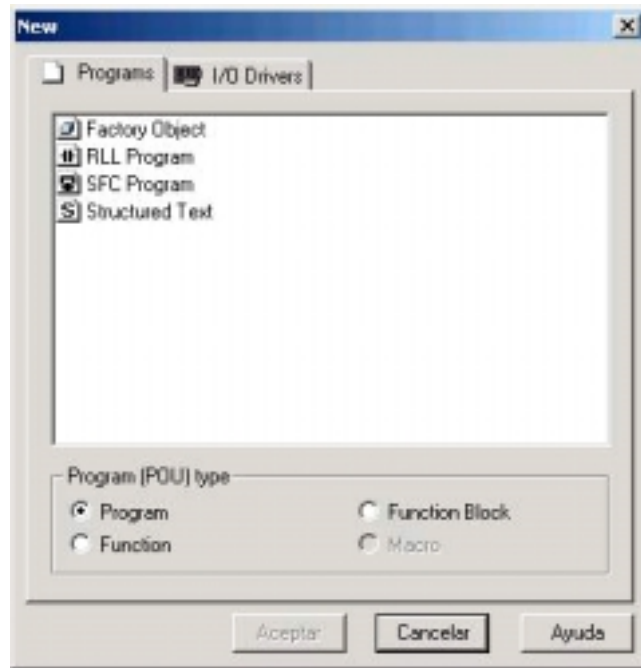


Para la implementación de la estrategia de control PID se utiliza el regulador PID digital (**FOE PID**) que viene incluido como una de las herramientas internas de trabajo del InControl.

Lo primero es anexar el FOE PID al proyecto ya creado (**Control PID**), esto se hace, haciendo doble clic sobre el nombre del proyecto (**Control PID**) en el **Project Manager**, para abrirlo en el ambiente de desarrollo:

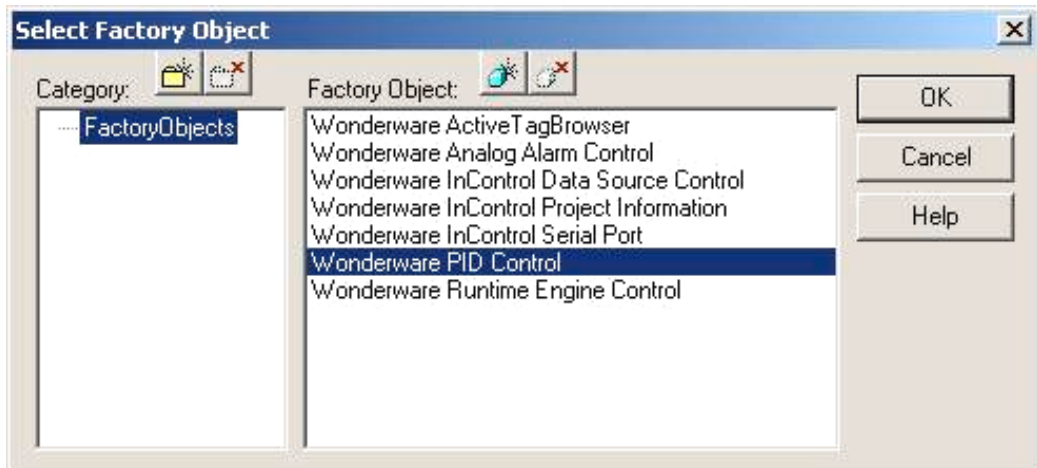
1. Una vez en el ambiente desarrollo, sobre el menú **File**, se hace clic en **New**. Aparecerá el cuadro de diálogo **New** (Figura 44).

Figura 44. Cuadro New



2. Se selecciona la etiqueta **Programs** en el cuadro de diálogo **New**.
3. Se selecciona el tipo de programa **Factory Object** (Objetos de Fabrica). Después de escogida esta opción, se pulsa **Aceptar**.
4. Seguido a esto aparecerá el cuadro de dialogo **Select Factory Object** (figura 45), donde se escogerá el regulador digital (FOE) **Wonderware PID Control** y se pulsa **OK**.

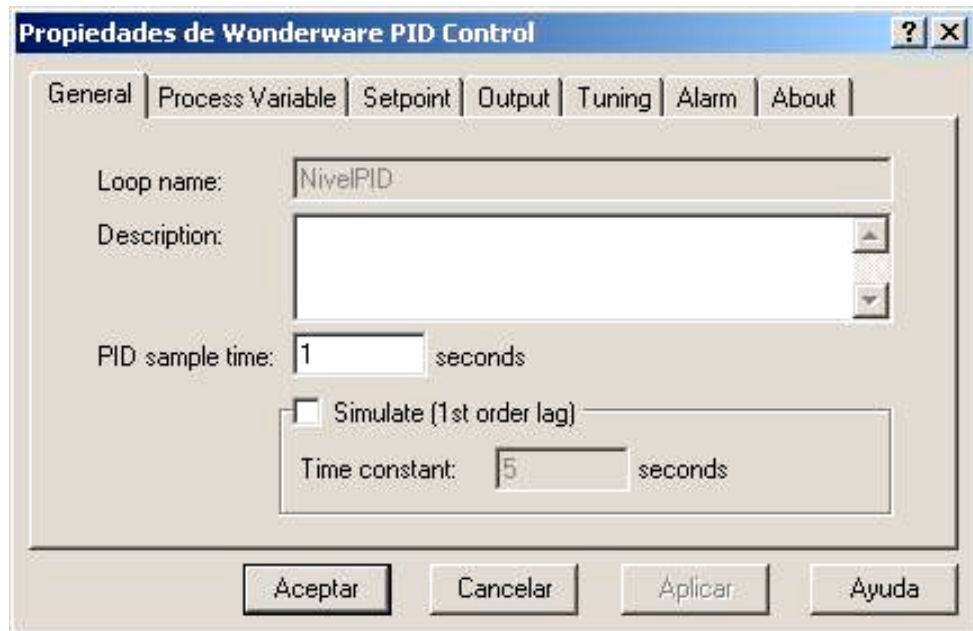
Figura 45. Cuadro Select Factory Object



5. Luego se ingresa el nombre del FOE (**NiveIPID**) en el cuadro de diálogo **Guardar como**.

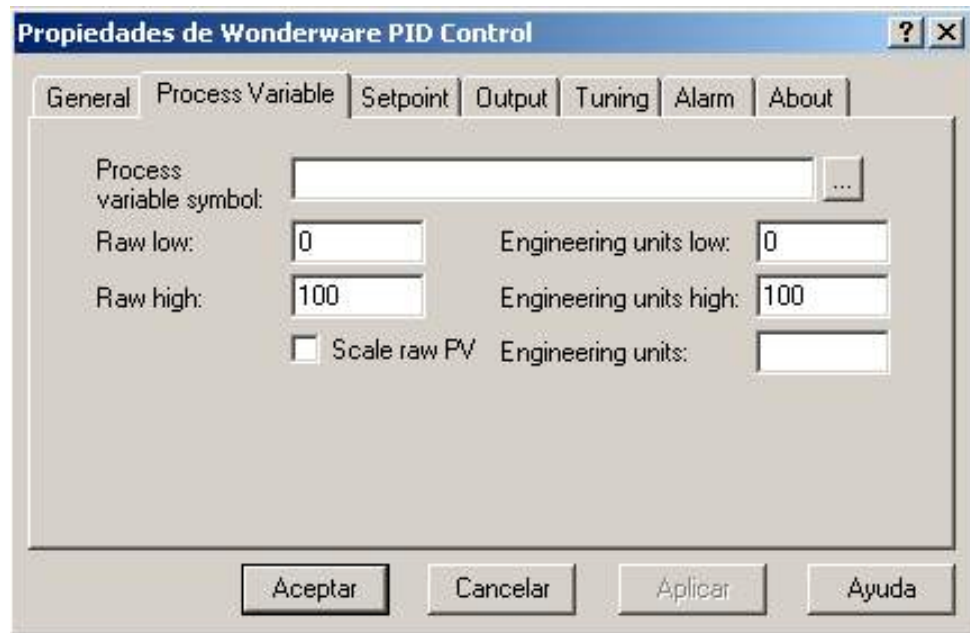
Seguido a estos pasos en la **Project View** aparecerá el nuevo programa (FOE) **NiveIPID**, en la carpeta **Programs**. Después de esto se configuraran los parámetros del FOE (**NiveIPID**) como son tiempo de muestreo, limite y unidades de la variable de proceso, valores mínimos y máximos del setpoint, la variable de salida y los valores de sintonización (Proporcional, integral y derivativo), en el cuadro de diálogo **Propiedades de Wonderware PID Control** (Figura 46), el cual se habilita haciendo doble clic sobre la ventana del FOE.

Figura 46. Propiedades de Wonderware PID Control



Se fija un tiempo de muestreo de 0.1 segundo para el tiempo de muestreo del PID. Para la variable de proceso (figura 47) se escoge el símbolo que representa el nivel real del tanque “**Nivel_tanque**” suministrado por el PLC a través del puerto de entrada análoga proveniente del transmisor de nivel y los valores límites para este dato.

Figura 47. Propiedades de Wonderware PID Control etiqueta variable de proceso



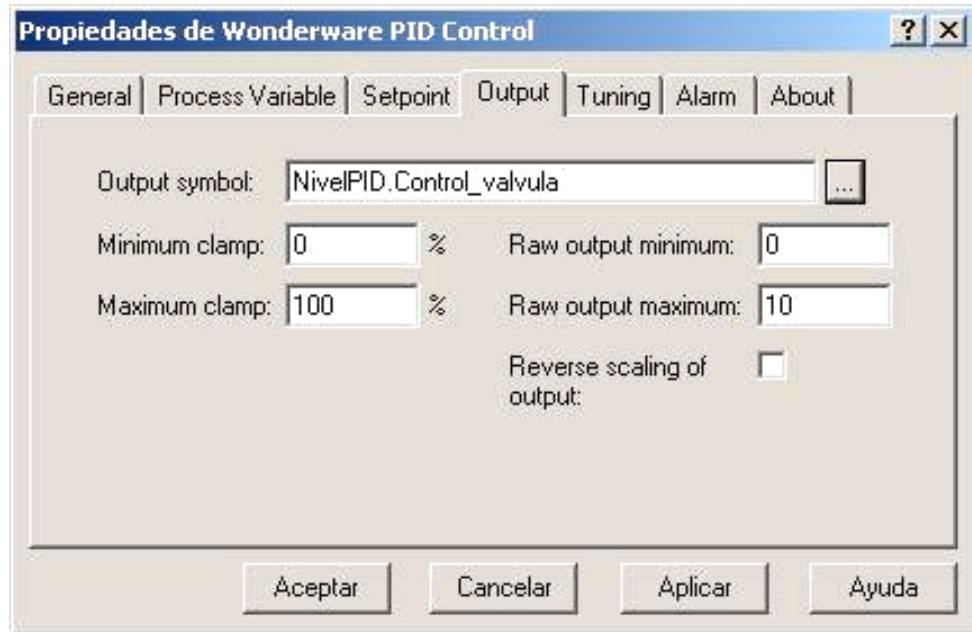
En la etiqueta **Setpoint** (Figura 48) se asigna el símbolo que representa este dato en el programa "**NiveIPID.Sp**" y sus límites (valor máximo y valor mínimo).

Figura 48. Propiedades de Wonderware PID Control etiqueta Setpoint



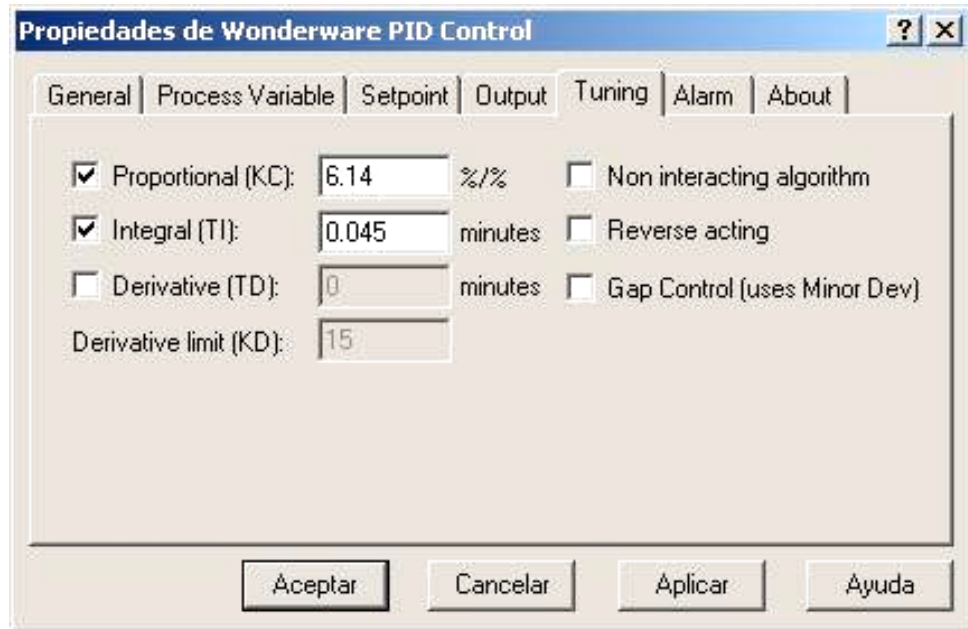
En la etiqueta **Output** (figura 49) correspondiente a la salida del PID se asigna el símbolo que representa este valor en el programa “**NivelPID.Control_valvula**” y sus límites (valor máximo y valor mínimo).

Figura 49. Propiedades de Wonderware PID Control etiqueta Output



En la etiqueta **Tuning** (Figura 50) correspondiente a los parámetros de sintonización del PID se asigna los valores de los parámetros (K_c , T_i , T_d y K_d).

Figura 50. Propiedades de Wonderware PID Control etiqueta Tuning



A continuación se muestra el FOE PID (Figura 51) después de ser configurado y en plena ejecución para un setpoint (línea blanca) de 65 cm y un valor de nivel (línea verde) de 20 cm y valor de voltaje (línea azul) de salida de 10 voltios.

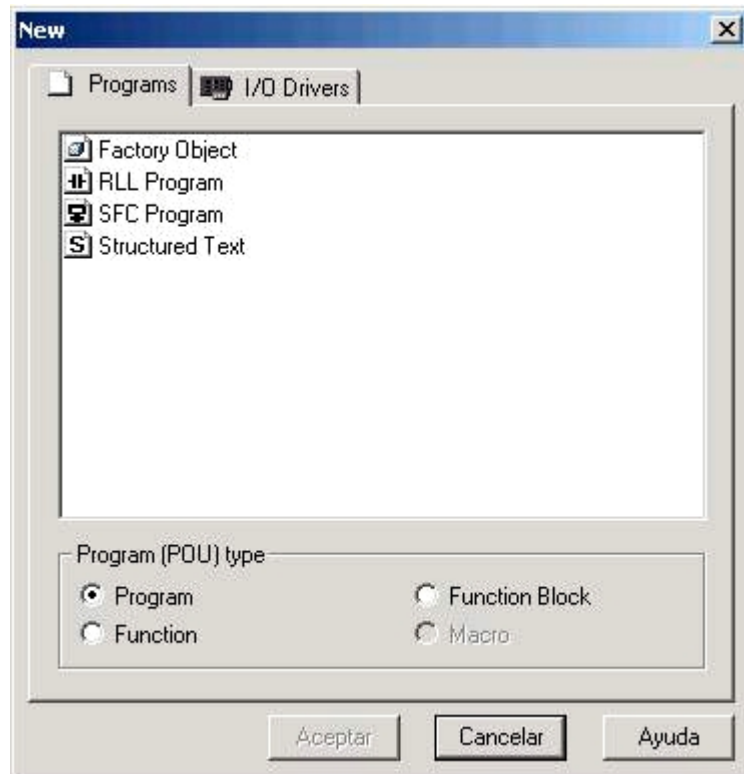
Figura 51. Ventana del FOE PID



Después de configurar el FOE **NivePID**, se crean los programas necesarios para ejecutar el encendido de la bomba y de configuración del modo de ejecución (automático, manual) del FOE, estos programas son creados en nuestro proyecto **ControlPID**, de la misma manera como se crea el programa **NivePID**, y son mostrados a continuación.

1. Sobre el menú **File**, pulse **New**. Aparecerá el cuadro de diálogo **New** (Figura 52).

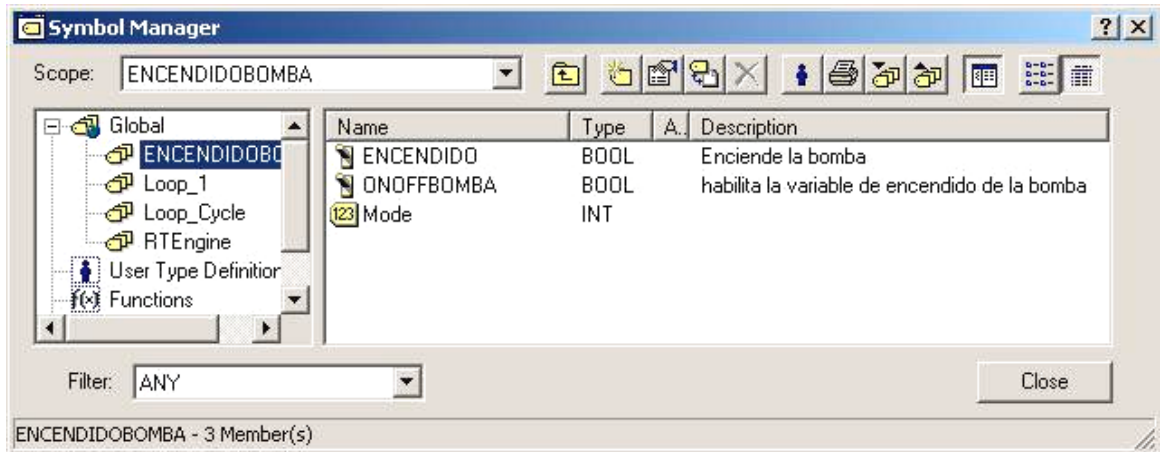
Figura 52. Cuadro new



2. Se selecciona la etiqueta **Programs** en el cuadro de diálogo **New**.
3. Se pulsa en **RLL Program**. Luego se ingresa el nombre del programa **ENCENDIDOBOMBA** en el cuadro de diálogo **Guardar como**.

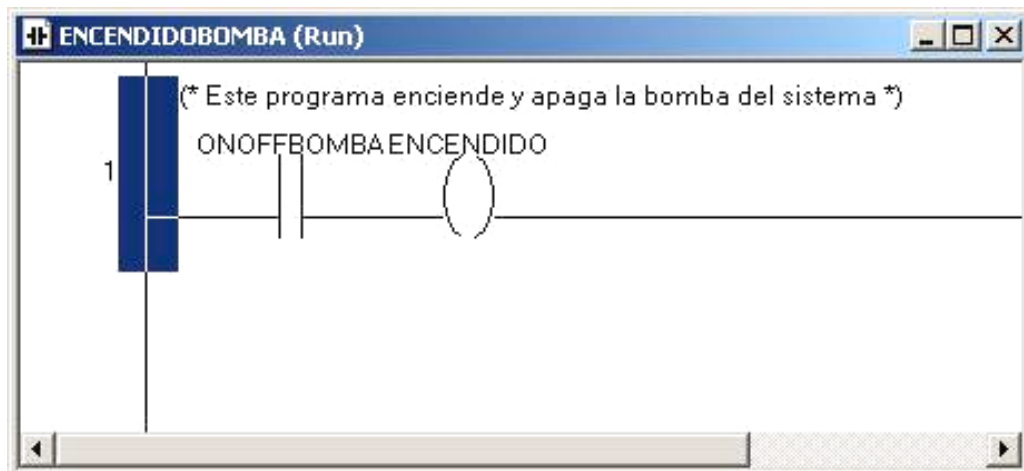
El programa a continuación es el encargado de encender y apagar la bomba, este fue creado en el lenguaje de programación RLL (Relay Ladder Logic) que posee el InControl. El primer paso será definir las variables necesarias para la ejecución del programa, en el Symbol Manager, las cuales se muestran en la figura 53.

Figura 53. Symbol Manager



Y está compuesto de las siguientes órdenes lógicas, representadas por medio de contactos y bobinas.

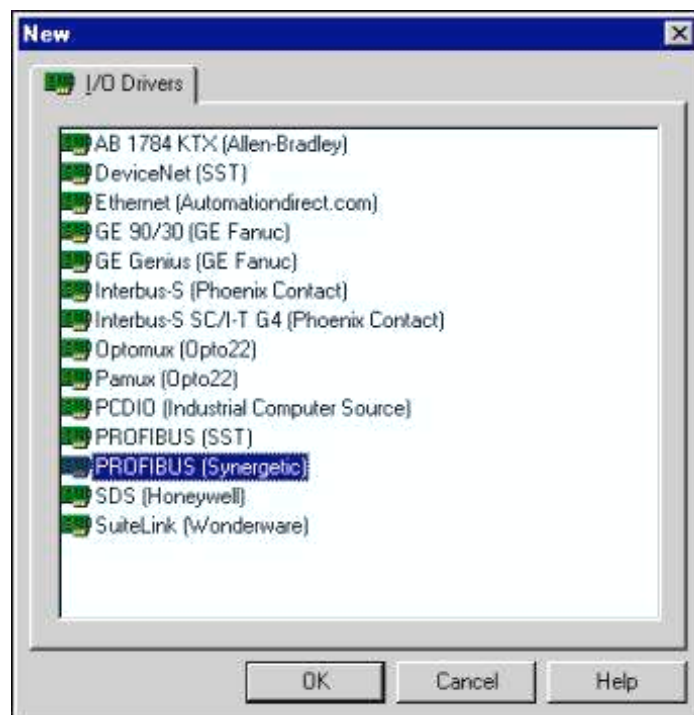
Figura 54. Programa de encendido de la bomba



Después de creados los programas se anexa el driver a nuestro proyecto. Para la adición del driver InControl Profibus (Synergetic) I/O Driver de la tarjeta CIF30-DPM a nuestro proyecto **Control PID**, se realiza en InControl los siguientes procedimientos:

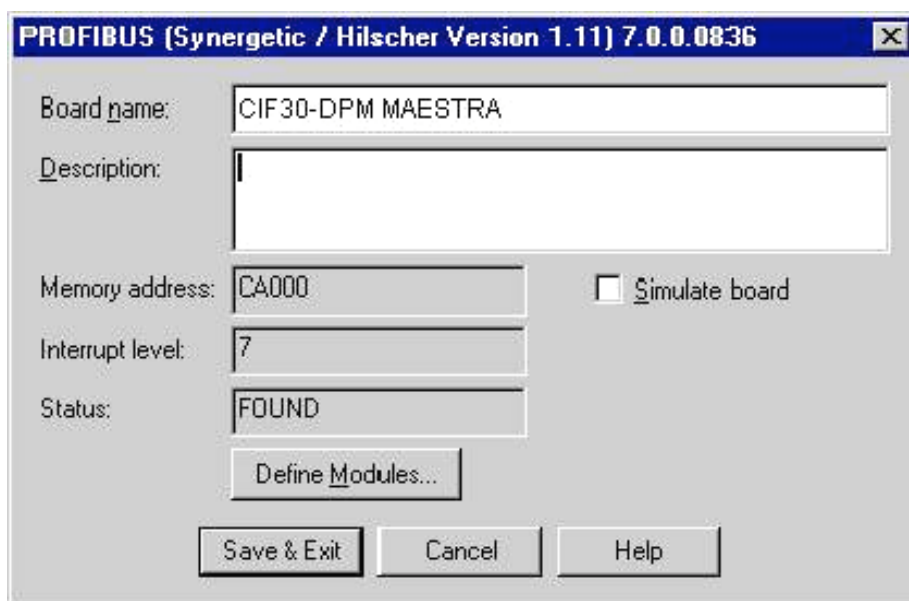
1. En el ambiente de desarrollo de InControl se hace clic derecho en la carpeta **I/O** de la ventana **Project View** y se selecciona en **New I/O**.
2. Se selecciona el driver correspondiente a la tarjeta **CIF30-DPM** que es el **PROFIBUS (Synergetic)** y se pulsa **OK** (Figura 55).

Figura 55. Selección del I/O Driver Profibus (Synergetic) en InControl



Para verificar que el driver fue instalado en la carpeta I/O se hace doble clic sobre el driver **PROFIBUS (Synergetic/Hilscher)**, aparece el cuadro de dialogo **PROFIBUS (Synergetic/Hilscher)**.

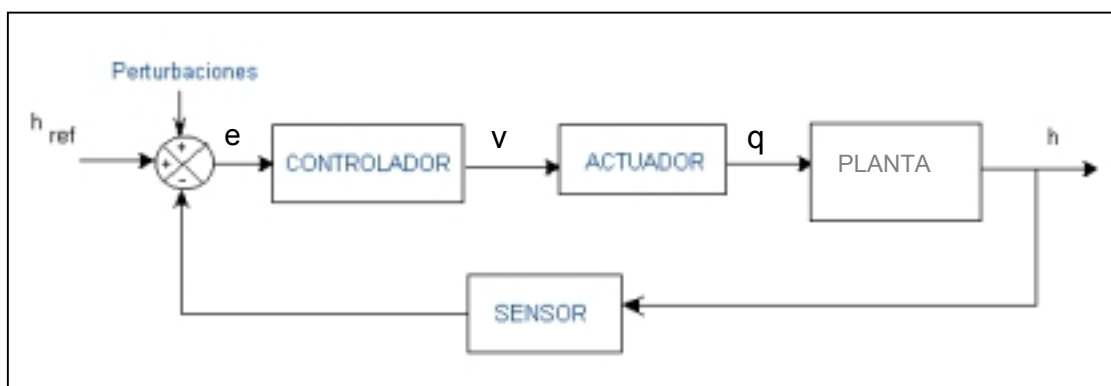
Figura 56. Cuadro de dialogo Profibus (Synergetic/Hilscher).



Se nota que en la etiqueta **Status** dice **"FOUND"**. Esto significa que el InControl ha encontrado la tarjeta y está lista para ser configurada. Los pasos de configuración del driver **Profibus (Synergetic/Hilscher)** de la tarjeta **SMS CIF30-DPM** están descritos en el tema **"CONFIGURACIÓN DE LA TARJETA SMS CIF30-DPM"**.

Descripción de la estrategia de control PID: El control de nivel en el tanque 1 se realiza a través de la manipulación del caudal de agua que fluye por la válvula (actuador). El sistema de control utilizado es el de lazo cerrado, tal como se muestra en la siguiente figura.

Figura 57. Sistema de control de lazo cerrado



El nivel real será medido por el transmisor de nivel (Sensor) que está instalado en el banco, desde allí este envía una señal de corriente (4 a 20 mA) proporcional al nivel del tanque, al PLC este a su vez envía la información recibida a través del bus de campo establecida como una señal digital (protocolo PROFIBUS DP) al software InControl (controlador) por medio de la tarjeta CIF30-DPM.

El FOE PID toma el valor de nivel real que le entrega el PLC, proveniente del transmisor de nivel y lo compara con el valor de nivel deseado (setpoint), obteniendo así una diferencia o señal de error, luego esta señal de error es procesada por el algoritmo PID y el resultado de este procesamiento es finalmente

convertido en una señal análoga que llega al actuador (válvula proporcional servoactuada) de la variable manipulada por medio de la salida análoga del PLC.

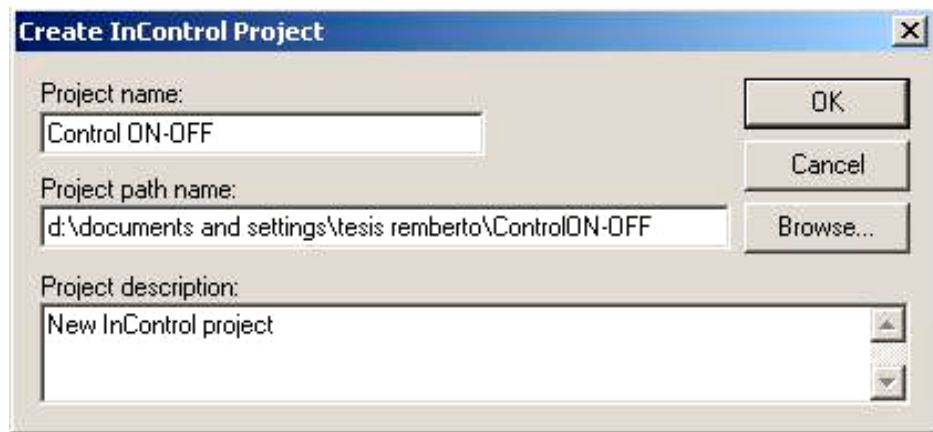
Esta señal llega a la válvula proporcional para que esta varíe su porcentaje de apertura. Esta variación automática permite que haya un mayor o menor flujo de agua por el tubo, lo cual hace que aumente o disminuya el nivel en el tanque hasta alcanzar el nivel deseado (equilibrio de masas).

3.2.2.3 Diseño del control ON-OFF: El primer paso en el desarrollo de la estrategia de control **ON-OFF** es crear el proyecto en InControl concerniente al proceso (control de nivel), lo cual está descrito en los siguientes pasos:

Después de iniciar el InControl:

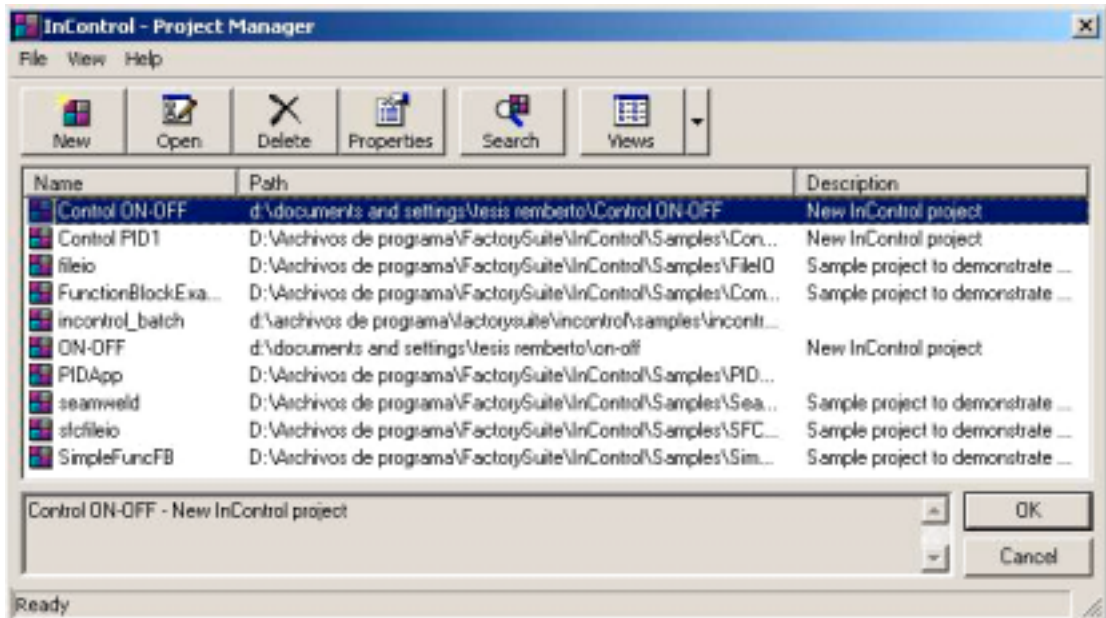
1. En el menú **File** del cuadro de diálogo **InControl Project Manager**, se hace clic en **New**. Aparece el cuadro de diálogo **Create InControl Project**.

Figura 58. Creación de proyectos InControl



2. Se ingresa el nombre del proyecto (**Control ON-OFF**). Se escribe el nombre del proyecto usando las convenciones para nombres de Windows NT, se selecciona una ruta, y se pulsa **OK**. Un directorio con este nombre es creado en el disco duro y el nuevo proyecto será adicionado a la lista de proyectos de InControl. En la siguiente figura, el proyecto **Control ON-OFF** es creado.

Figura 59. Presentación del proyecto ya creado

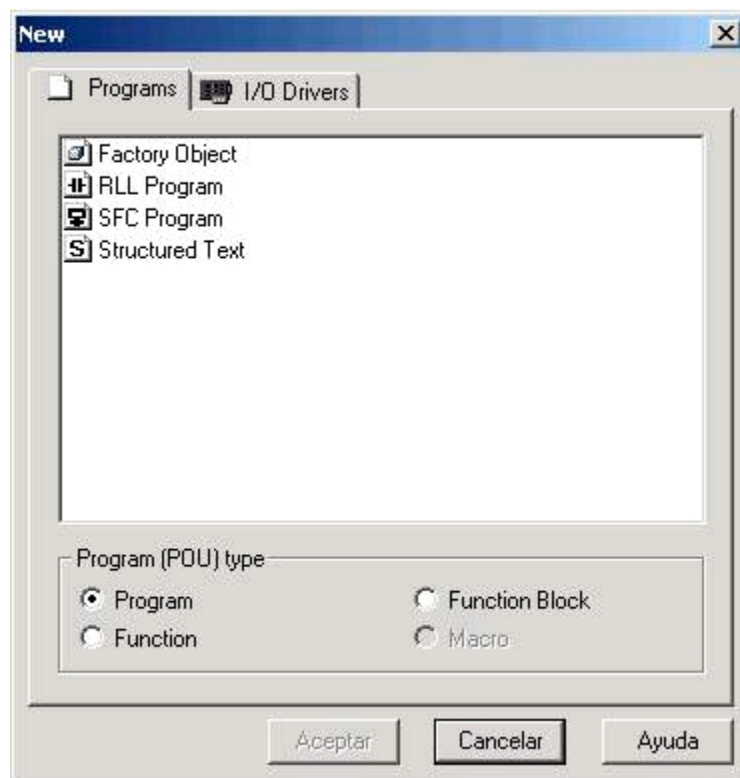


El proyecto que se crea aparece en la ventana de Proyectos del ambiente de desarrollo. Después de crear el proyecto, se inicia la creación de los programas necesarios para la ejecución de la estrategia de control ON-OFF. Para la implementación de la estrategia de control ON-OFF se utiliza el lenguaje de programación que viene incluido como una de las herramientas internas de trabajo del InControl.

Lo primero es crear el programa encargado de la estrategia de control ON-OFF en el proyecto ya creado (**Control ON-OFF**), esto se hace, haciendo doble clic sobre el nombre del proyecto **Control ON-OFF** en el **Project Manager**, para abrirlo en el ambiente de desarrollo:

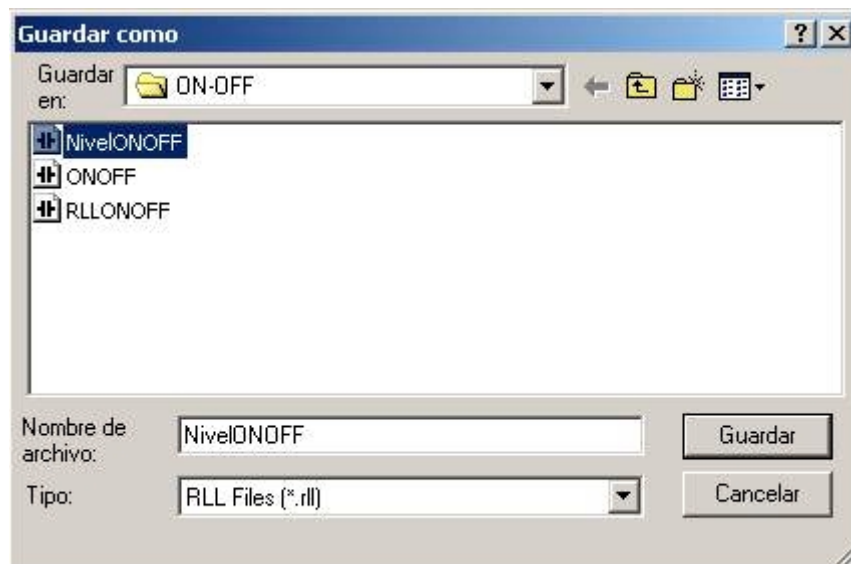
1. Una vez en el ambiente desarrollo, sobre el menú **File**, se pulsa en **New**. Aparece el cuadro de diálogo **New**.

Figura 60. Cuadro New



2. Se selecciona la etiqueta **Programs** en el cuadro de dialogo **New**.
3. Se selecciona el tipo de programa **RLL Program** (Programas RLL). Después de escogida esta opción, se pulsa **Aceptar**.

Figura 61. Cuadro de diálogo para guardar el controlador On/OFF



4. Luego aparece el cuadro de diálogo **Guardar como**, donde se ingresara el nombre del programa RLL (**NivelONOFF**).

El editor RLL muestra un nuevo archivo RLL con dos barras de energía y un escalón vacío. Lo cual indica que todo está listo para editar el programa. En la **Project View** en la carpeta **Programs** aparece el nuevo programa **NivelONOFF**.

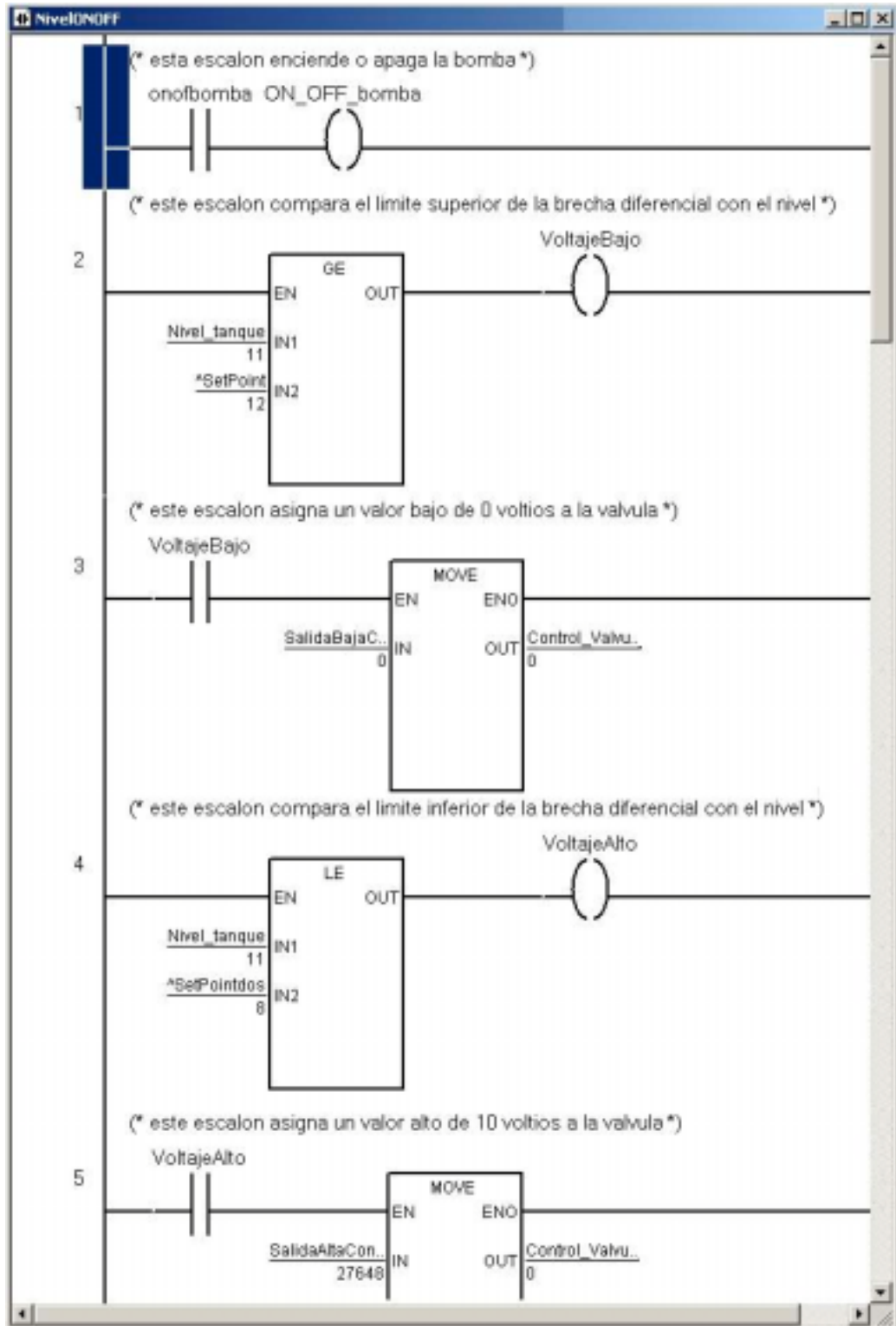
El programa a continuación (Figura 63) será el encargado de efectuar la estrategia de control **ON-OFF** y además de encender y apagar la bomba, este es creado en el lenguaje de programación RLL (Relay Ladder Logic) que posee el InControl. El primer paso es definir las variables necesarias para la ejecución del programa, en el Symbol Manager, entre estas tenemos:

Figura 62. Symbol Manager del Programa NivelONOFF



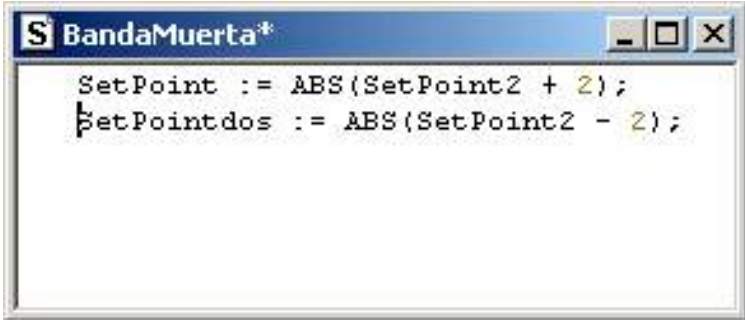
Y esta compuesto de las siguientes órdenes lógicas, representadas por medio de contactos y bobinas.

Figura 63. Presentación del programa NivelONOFF



Además se creara un programa **BandaMuerta** (Figura 64) en lenguaje de programación STL (Structured Text Language) encargado de fijar la zona de banda muerta (brecha diferencial) para el control ON-OFF, que en nuestro caso es de +/-2 cm debido a que entre mas pequeña es esta brecha diferencial el actuador (válvula) tiene una frecuencia de trabajo muy elevada y así evitar que la Válvula presente características de saturación o un mal desempeño de sus funciones, esta consideración se tomo del análisis previo del comportamiento real del sistema actual.

Figura 64. Programa que fija la brecha diferencial o Banda muerta del controlador ON-OFF

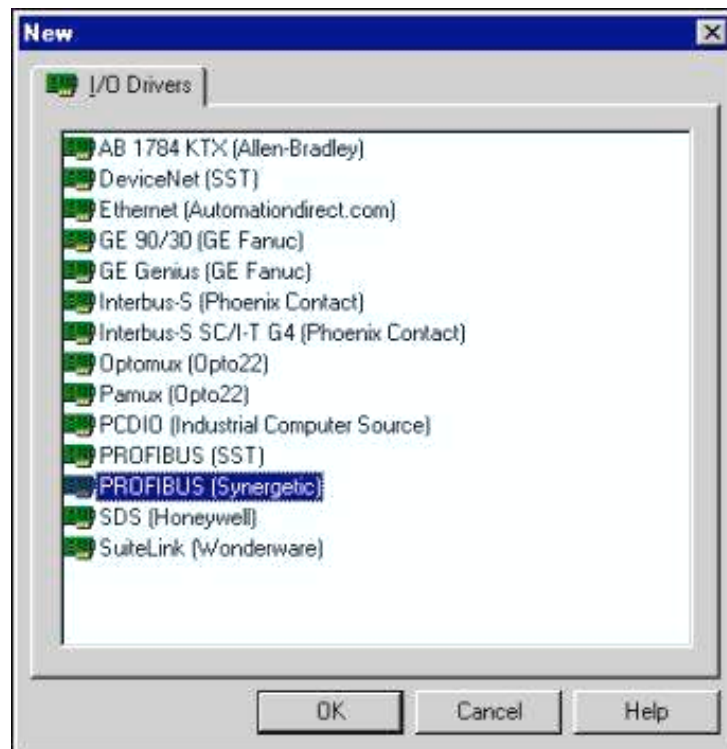


```
SetPoint := ABS(SetPoint2 + 2);  
SetPointdos := ABS(SetPoint2 - 2);
```

Después de crear los programas se anexa el driver a nuestro proyecto. Para la adición del driver InControl Profibus (Synergetic) I/O Driver de la tarjeta CIF30-DPM a nuestro proyecto **Control PID** se realiza en InControl los siguientes procedimientos:

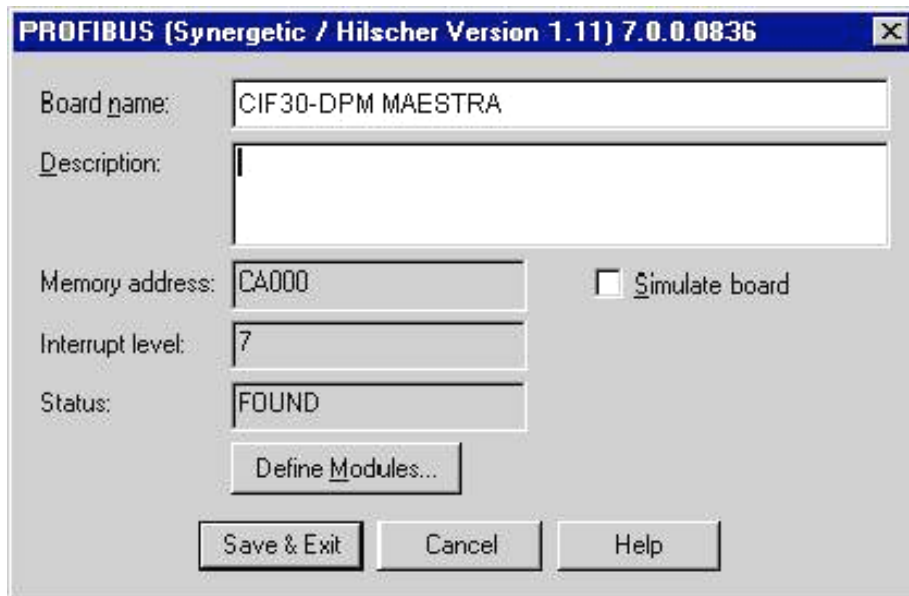
1. En el ambiente de desarrollo de InControl se hace clic derecho en la carpeta **I/O** de la ventana **Project View** y se seleccionara **New I/O**.
 - a. Se seleccionara el driver correspondiente a la tarjeta **CIF30-DPM** que es el **PROFIBUS (Synergetic)** y se pulsa en **OK** (Figura 65).

Figura 65. Selección del I/O Driver Profibus (Synergetic) en InControl



Para verificar que el driver fue instalado en la carpeta I/O se hace doble clic sobre el driver **PROFIBUS (Synergetic/Hilscher)**, aparece el cuadro de diálogo **PROFIBUS (Synergetic/Hilscher)** (Figura 66).

Figura 66. Cuadro de diálogo Profibus (Synergetic/Hilscher)



Se nota que en la etiqueta **Status** dice **“FOUND”**. Esto significa que el InControl ha encontrado la tarjeta y esta lista para ser configurada. Los pasos de configuración del driver **Profibus (Synergetic/Hilscher)** de la tarjeta **SMS CIF30-DPM** están descritos en el tema **“CONFIGURACIÓN DE LA TARJETA SMS CIF30-DPM”**.

Descripción de la estrategia de control ON-OFF: El control ON-OFF toma el valor de nivel real que le entrega el PLC, proveniente del transmisor de nivel y lo compara con el valor de nivel deseado (setpoint), si este valor real no esta dentro del rango de valores de setpoint permitidos en la brecha diferencial o banda muerta, se envía una señal análoga que llega al actuador (válvula proporcional

servoactuada) de la variable manipulada por medio de la salida análoga del PLC. Esta señal llega a la válvula proporcional para que varíe su porcentaje de abertura. Esta variación automática permite que haya un mayor o menor flujo de agua por el tubo, lo cual hace que aumente o disminuya el nivel en el tanque hasta alcanzar el nivel deseado (equilibrio de masas).

En el controlador ON-OFF (todo o nada), el elemento de control sólo proporciona dos niveles de control: total y nulo. Además se observa que en éste tipo de control el seguimiento del setpoint no es muy preciso. Durante la simulación en Matlab de este control la respuesta del controlador ON-OFF presenta una oscilación sostenida alrededor del setpoint, propia de éste tipo de control. La banda muerta se fija en ± 2 cm, ya que el actuador tiene una frecuencia de trabajo elevada a medida que se disminuye la brecha diferencial (Banda muerta). Se observara que en éste tipo de control el seguimiento del setpoint no es muy preciso, la respuesta del controlador ON-OFF presentara una oscilación sostenida alrededor del setpoint, propia de éste tipo de control. La actuación del control ON-OFF se selecciono así:

Sobre él limite = la salida es cero = 0 voltios (Nulo: se cierra la válvula).

Bajo él limite = la salida es 1 = 10 voltios (Total: se abre la válvula).

3.2.2.4 Configuración de la tarjeta SMS CIF30-DPM como maestro DP.

Dispositivos necesarios:

- **Tarjeta SMS-CIF30-DPM Profibus ISA DP Maestro**
- **Software de configuración SYCON-DP**
- **Archivos GSD Profibus del PLC S7-300**

Los siguientes pasos definen los procedimientos generales que se realizarán para llegar a usar la tarjeta **PROFIBUS I/O (SMS-CIF30-DPM)** como componente maestro en una red Profibus.

- Instalación de la tarjeta I/O (SMS-CIF30-DPM) en la unidad de hardware usada para correr InControl, esto es después de adquirir la tarjeta.
- Uso del software de configuración SyCon-DP de Synergetic para desarrollar la configuración PROFIBUS DP.
- Descarga de la configuración a la tarjeta SMS-CIF30-DPM de Synergetic.
- Instalación del driver para tarjeta I/O (SMS-CIF30-DPM).

- Adición del driver **Profibus (Synergetic)** al *Project View* (vista de proyectos) de la ventana *Project* en InControl.
- Ingreso al cuadro de diálogo de configuración **Profibus (Synergetic/Hilscher)**.
- Ingreso de la información de configuración para la tarjeta I/O.

A continuación se describe paso a paso la instalación del hardware y software correspondiente a la tarjeta **SMS-CIF30-DPM**, así como también el software de configuración **SYCON** para esta tarjeta. En éstos se describe la configuración de la tarjeta **SMS-CIF30-DPM** como maestro DP en un red **PROFIBUS** bajo un sistema de conexión Maestro/Esclavo con el **PLC S7-300**, ésta configuración nos permitirá establecer la comunicación necesaria entre el PLC y la tarjeta, y de esta manera poder recibir los datos y realizar el control del proceso, solucionando los inconvenientes presentados a nivel de comunicación con el banco Control de Nivel.

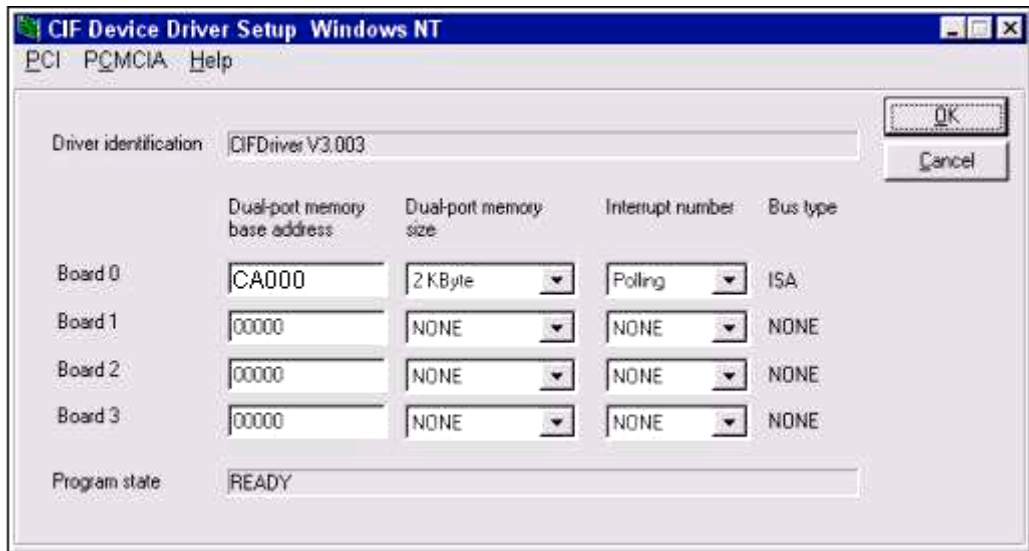
Nota: Para la realización de algunos de los pasos que se describe a continuación es necesario poseer la tarjeta **SMS-CIF30-DPM** instalada en el computador. Para poder llevarlos a cabo fue necesaria la colaboración de la empresa Colsein.Ltda en Barranquilla, ya que ellos han adquirido posteriormente dicha tarjeta y realizado

montajes para empresas del sector industrial en Colombia, y poseen los drivers necesarios para efectuar dicha configuración (el nombre de la empresa es información clasificada por parte de colsein.Ltda.

Paso 1: Instalación y Configuración del programa CIF Device Driver Setup de la tarjeta I/O. Con el PC apagado y ya adquirida la tarjeta, se instala la tarjeta **SMS-CIF30-DPM** dentro de un Slot ISA libre en el computador donde se desarrolló la aplicación, después se dará inicio al programa **CIF Device Driver Setup** (configuración del driver de la tarjeta CIF30-DPM), ingresando la información correspondiente a la memoria de puerto dual, al tamaño de la memoria y al número de interrupciones. La Figura 67 muestra el cuadro de diálogo **CIF Device Driver Setup**. Para información mas detallada sobre la instalación de esta tarjeta ver el anexo “**MANUAL DE HARDWARE DE LA SMS CIF30-DPM**”.

- a. La tarjeta **SMS-CIF30-DPM** usada en este ejemplo tiene un tamaño de memoria de 2K.
- b. El número de interrupciones se definirá por **Polling** (escaneo de interrupciones de los módulos conectados al bus para procedimiento maestro-esclavo) por la configuración misma del bus usada para manejar la comunicación entre componentes maestros y sus componentes esclavos.
- c. Se pulsa en **OK** para aceptar la configuración y reiniciar el computador.

Figura 67. Cuadro de diálogo CIF Device Driver Setup



Paso 2: Configuración del SyCon System Configurator. Los siguientes procedimientos describen cómo se configura el programa de configuración **SYCON-DP** de **Synergetic**.

1. Después de reiniciar el computador se instala el software de configuración **SYCON-DP** descargado de la pagina Web www.synergetic.com o desde el CD de instalación que suministra el proveedor del dispositivo (CIF30-DPM). Además se adicionan los archivos **GSD** (archivo de base de datos que posee descripción del componente esclavo) que definen los componentes Profibus de la red, estos archivos son suministrados por el proveedor o pueden estar contenidos en el componente maestro de la red, en este caso la tarjeta **SMS-CIF30-DPM** de **Synergetic**, pero ésta no posee estos

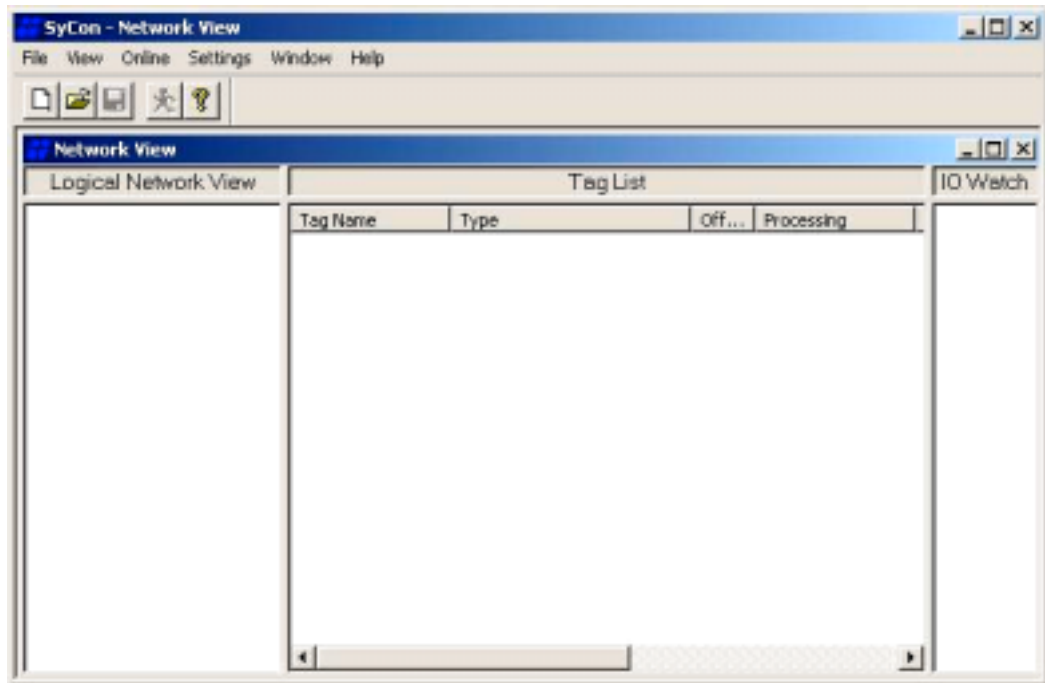
archivos para el **PLC S7-300** de Siemens, por lo tanto se descargara este archivo de la pagina Web de Siemens. Luego el archivo (**S7-314C-2DP : SIE_802F.GSD**) se importa dentro de la base de datos de **SYCON** (C:\ArchivosdePrograma\Synergetic\SYCON\Fieldbus\carpetaGSD) para ser usado en la configuración.

Figura 68. SyCon System Configurator



2. Después de instalar el software **SYCON** se da inicio a su ejecución, haciendo clic sobre el icono **SYCON**, en la barra de tareas de Inicio\Programas\SyCon System Configurator. La pantalla por defecto aun sin configurar se muestra a continuación (Figura 69).

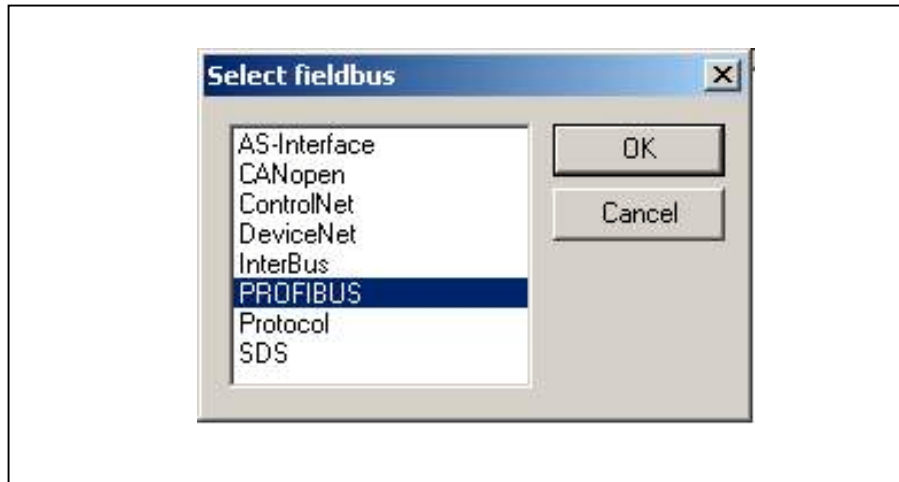
Figura 69. Pantalla de inicio del software SyCon System Configurator



Después de iniciar el software Sycon, estos son los pasos necesarios para crear un nuevo proyecto:

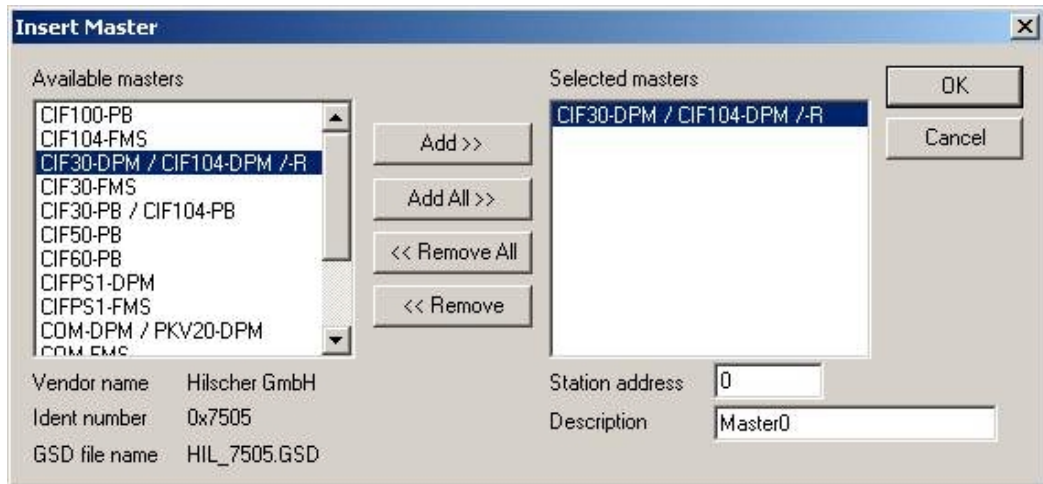
- a. Se hace clic en **File/New** para crear un nuevo archivo.
- b. Se selecciona el bus de campo **Profibus** en el menú **Select Field Bus** (Figura 70), y se hace clic en **OK**.

Figura 70. Cuadro Select FieldBus



- c. En el menú **Insert** se selecciona la opción **Master**.
- d. Se señala y se hace clic una sola vez con la flecha en el rectángulo, debajo de la barra de menú. Aparece el cuadro de diálogo **Insert Master** como se muestra en la Figura 71.
- e. Desde el cuadro de diálogo **Insert Master**, se selecciona el maestro correcto de la sección derecha de maestros disponibles y se pulsa en el botón **Add**. En este caso nuestro maestro fue la tarjeta **CIF30-DPM**.

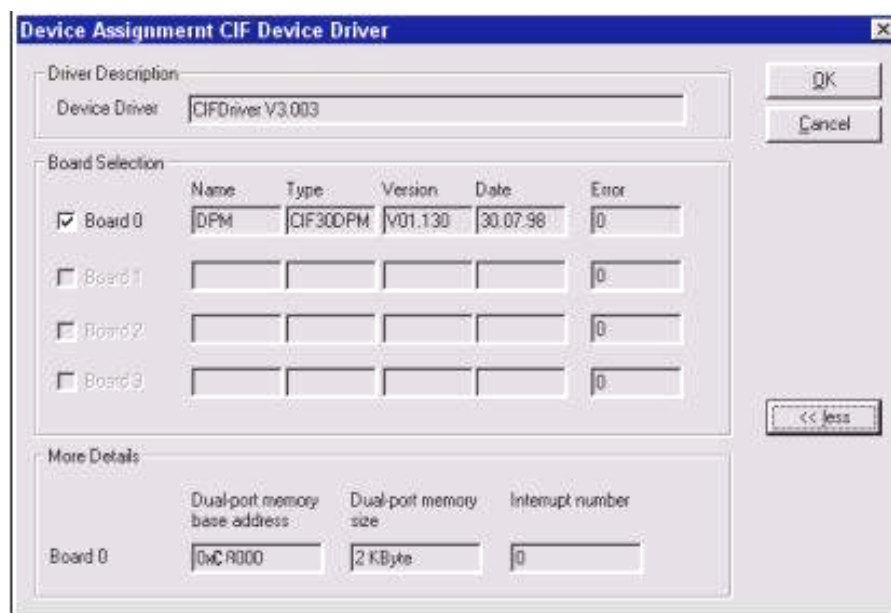
Figura 71. Cuadro de diálogo Insert Master



3. Se hace clic en el menú **Settings** y se selecciona **Device Assignment**.

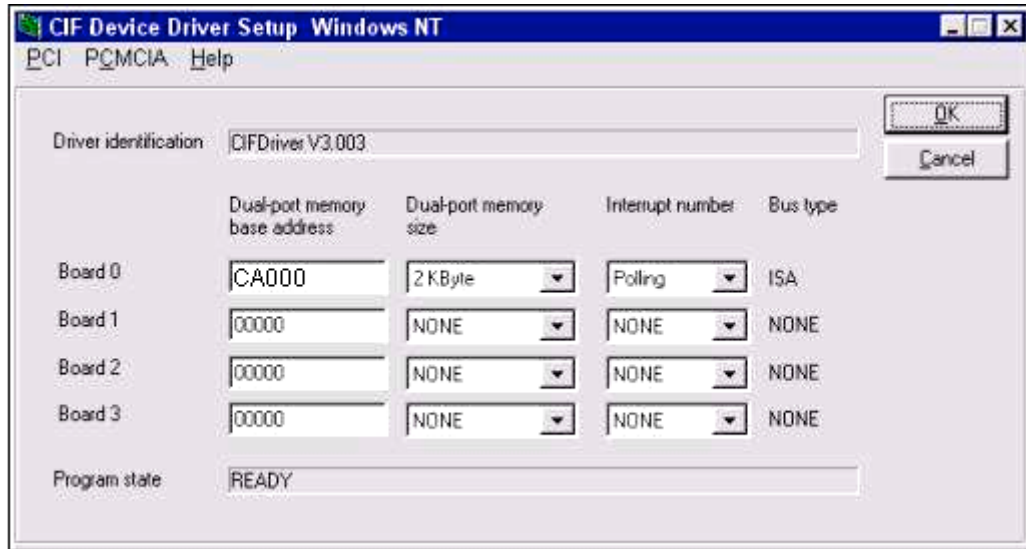
Aparece el cuadro de diálogo **Device Assignment CIF Device Driver**.

Figura 72. Cuadro de diálogo Device Assignment CIF Device Driver



4. Se habilita la opción **Board 0** en el área **Board Selection**.
- a. Se presentara información sobre la tarjeta en la figura 72 “**cuadro de dialogo Device Assignment CIF Device Driver**”.
 - b. Se revisa el código de error en el cuadro de diálogo. El cual no debe estar vacío.
 - Un código de error de 10 indica que la memoria de puerto dual de la tarjeta tiene un conflicto con otro componente en el computador.
 - Si está correcta se debe escoger otra dirección de memoria y reiniciar el computador, para que estos cambios tengan lugar.
 - c. Se hace clic en **OK** para salir.

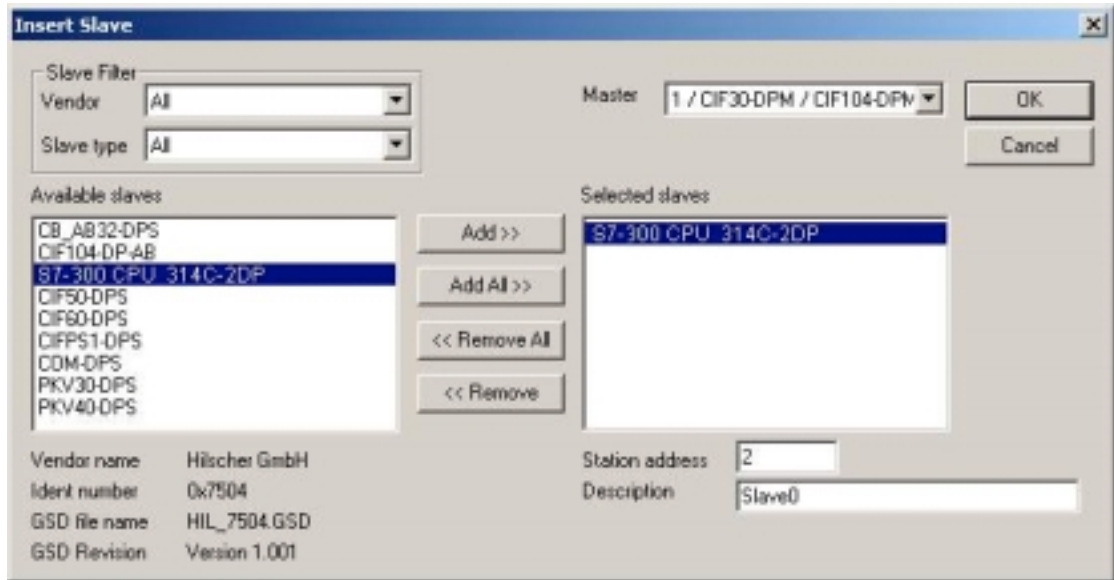
Figura 73. Cuadro de diálogo CIF DEVICE DRIVER



5. Luego de seleccionar el esclavo desde el menú **Insert**, aparecerá cuadro de dialogo **Insert Slave** (Figura 74).

- a. Se apunta y se hace clic con la flecha una vez en el bus, debajo del maestro.
- b. Se selecciona el componente esclavo correcto y se pulsa en el botón **Add**, lo cual adiciona el componente esclavo seleccionado al área en el lado derecho.
- c. Se cambia la dirección de estación en el nodo de dirección correspondiente a la dirección del bus.

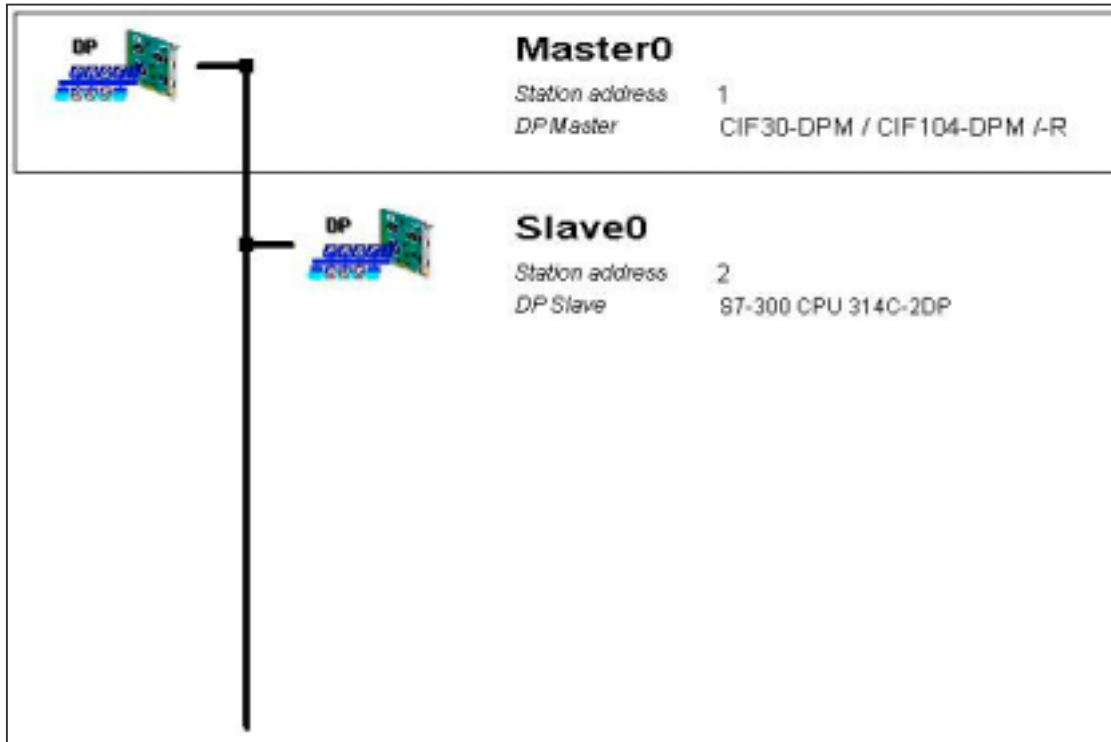
Figura 74. Cuadro de diálogo Insert Slave



d. Se pulsa en **OK** para aceptar la selección.

- A la vez, se observa al maestro y al esclavo anexados al bus Profibus (Figura 75).

Figura 75. Vista del proyecto PROFIBUS



6. En la ventana de proyectos **SYCON**, se hace clic en **Settings** y se selecciona en **Slave Configuration**.
 - a. Manualmente se ingresaran los atributos correctos para todos los módulos I/O en el sistema, y se pulsa en **OK** para confirmar.
 - b. Desde el menú **Online**, se selecciona **Download** para transferir los datos de configuración al maestro y al esclavo.

7. Por ultimo se configuraran los parámetros del bus, como son la velocidad de transmisión de datos y el tipo de optimización. En la ventana de proyectos SYCON, se hace clic en **Settings** y se selecciona la opción **Bus Paramater**.

a. Debido a que la longitud del cable RS-485 es menor de 100 m la velocidad del bus debe ser de 3 Mbps según la tabla contenida en el tema "**Cableado Y Equipo Requerido**", de hay que esta es la establecida durante la configuración del PLC S7-300, es importante que todos los componentes del bus tengan la misma velocidad de transmisión de datos (rata de baudios).

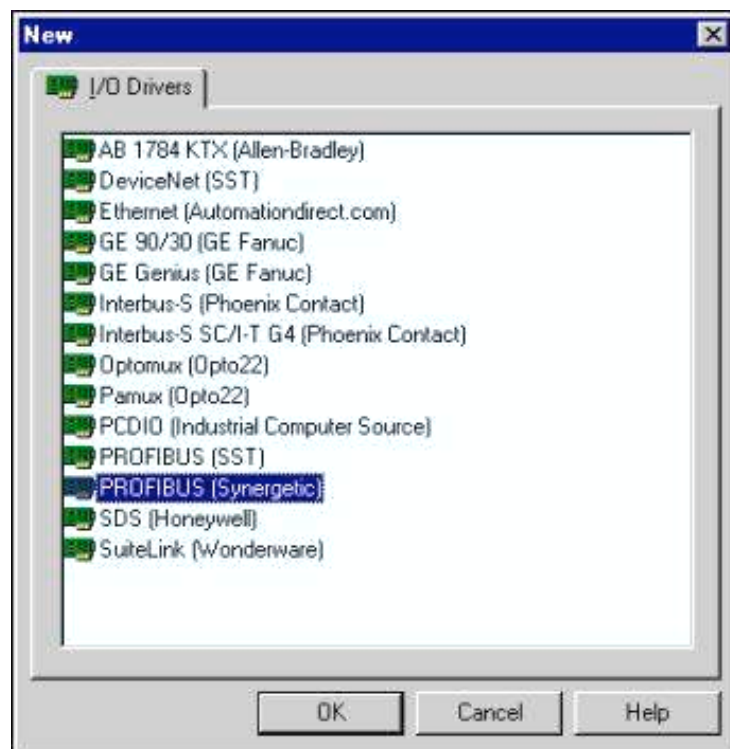
8. Para verificar si los dispositivos están conectados al bus correctamente. Se hace clic en **Online** en la barra de herramientas y seleccione **Star Mode debug**. Mientras el debug este corriendo, de esta manera los componentes esclavo y maestros serán conectados en su pantalla por una línea verde si la configuración y el cableado son correctos.

Paso 3: Adición del driver de Configuración del InControl Profibus (Synergetic) I/O Driver a la Project View en la ventana Project en InControl.

Los siguientes procedimientos describirán como es configurado el driver I/O Profibus de InControl.

1. En InControl se hace clic derecho en la carpeta **I/O** en la ventana **Project** y se selecciona **New I/O**.
 - b. Se selecciona **PROFIBUS (Synergetic)** y se pulsa en **OK** (Figura 76).

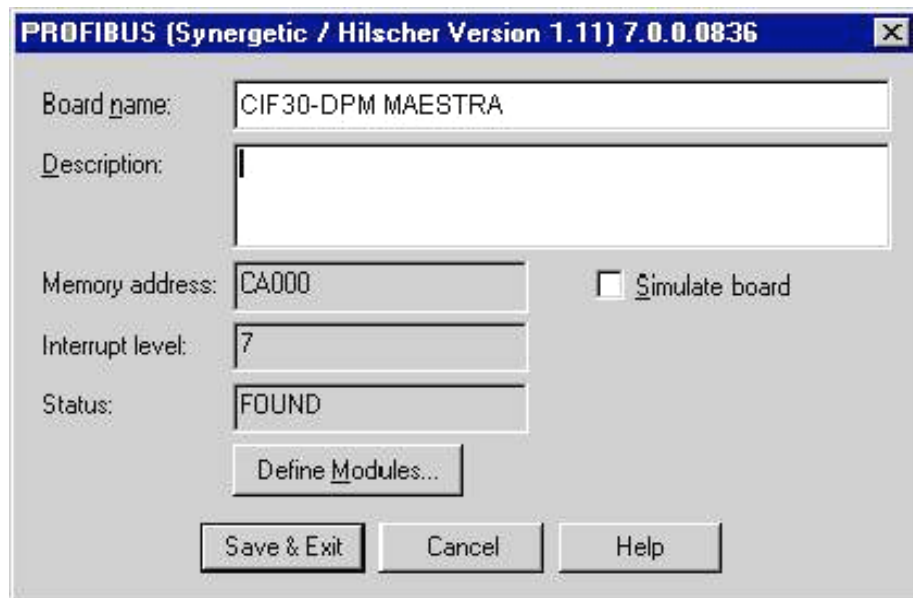
Figura 76. Selección del Profibus (Synergetic) I/O Driver en InControl



Paso 4: Ingreso y Configuración del Cuadro de dialogo PROFIBUS (Synergetic/Hilscher).

1. Se hace doble clic en la tarjeta PROFIBUS en la **Project View**, aparecerá el cuadro de dialogo **PROFIBUS (Synergetic/Hilscher)**.

Figura 77. Cuadro de diálogo Profibus (Synergetic/Hilscher)



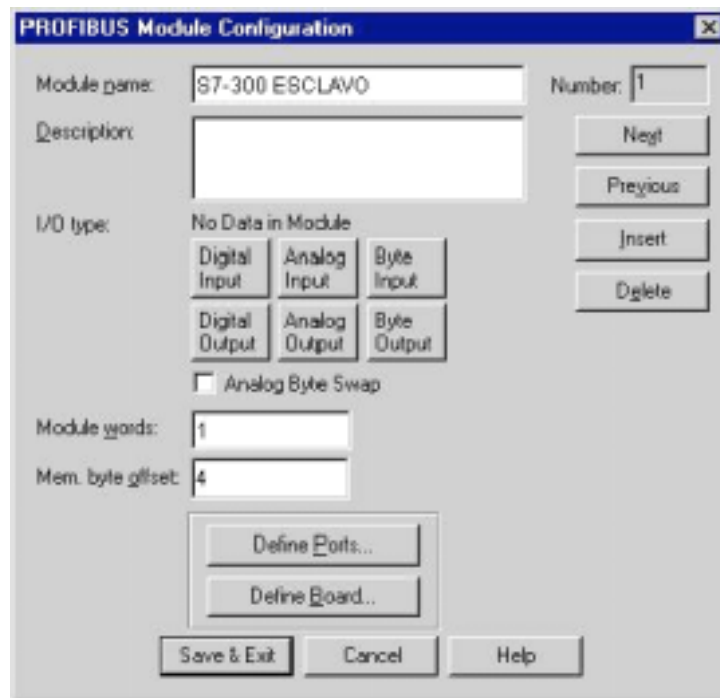
Note que el estado dice “FOUND”. Esto significa que el InControl ha encontrado la tarjeta y esta lista para ser configurada.

2. Se pulsa en el botón **Define Module** en el cuadro de diálogo de configuración **PROFIBUS (Synergetic/Hilscher)**. Este cuadro permite al usuario definir los TAG NAMES para los bits de entrada individual. Estos

Tag Names son parte de la base de datos universal que va ser usada entre InControl e InTouch, y que a su vez fueron creados en nuestros programas de aplicación. Aparece el cuadro de dialogo **PROFIBUS Module Configuration** (Figura 78).

- a. Se ingresa el nombre del modulo (S7-300 ESCLAVO), la descripción, el numero de datos tipo **word(16 bits)** asociados con el modulo (en nuestro caso : 3 word), el numero de byte de offset de memoria asociados con el sistema.
- b. Tenga en cuenta que la opción **Module words** es el numero de datos tipo word suministrado o habilitado por el módulo, y la opción **Memory byte offset** son los byte de datos que el modulo representa.
- c. Se pulsa el botón de recuadro (**I/O type**) apropiado, para seleccionar el tipo de I/O que utiliza el sistema (Digital Output, Analog Input, Analog Output).

Figura 78. Cuadro de dialogo PROFIBUS Module Configuration



Para la configuración de salidas digitales:

1. Para definir la salida digital se pulsa en el botón **Digital Output** y se define un valor de 1 en el cuadro **Module Word**. Esto denota que el modulo tomara 1 palabra de dato (Word).
2. Luego se pulsa en el botón **Define Ports** para definir los puertos de salida digital a utilizar en nuestra aplicación. Con esta opción se incorporan los símbolos I/O creados en nuestros programas InControl (**PID, ON-OFF**).

- La figura 79 muestra el cuadro de diálogo **PROFIBUS Digital Port Configuration**. Este cuadro permite al usuario definir los TAG NAMES (símbolos InControl) para los bits de entrada individual. Estos Tag Names son parte de la base de datos universal que se usa entre InControl e InTouch.

- a. En el campo seleccionado se ingresa un nombre significativo para el puerto (**OutPortM10**) y los símbolos (Encendido) de salida (1 salida digital).

Figura 79. Cuadro de dialogo PROFIBUS Digital Port Configuration

The image shows a software dialog box titled "PROFIBUS Digital Port Configuration". The main section is labeled "Digital Outputs". It includes the following fields and controls:

- Module name:** S7-300 ESCLAVO
- Number:** 1
- Word:** 0
- Port name:** OutPortM10
- Buttons:** "Next Port" and "Previous Port" are located to the right of the module name and number fields respectively.
- Output Slots:** A grid of 16 numbered slots (1-16) for digital outputs. Slot 1 is filled with the text "Encendido". Slots 2 through 16 are empty.
- Bottom Buttons:** "Define Module...", "Save & Exit", "Cancel", and "Help".

- b. Se pulsa en **Save & Exit** para guardar y completar la configuración I/O digital, se pulsa en **Define Modules** para regresar al cuadro de diálogo **PROFIBUS Module Configuration** para continuar con la configuración del resto de puertos I/O.

Para la configuración de salidas análogas:

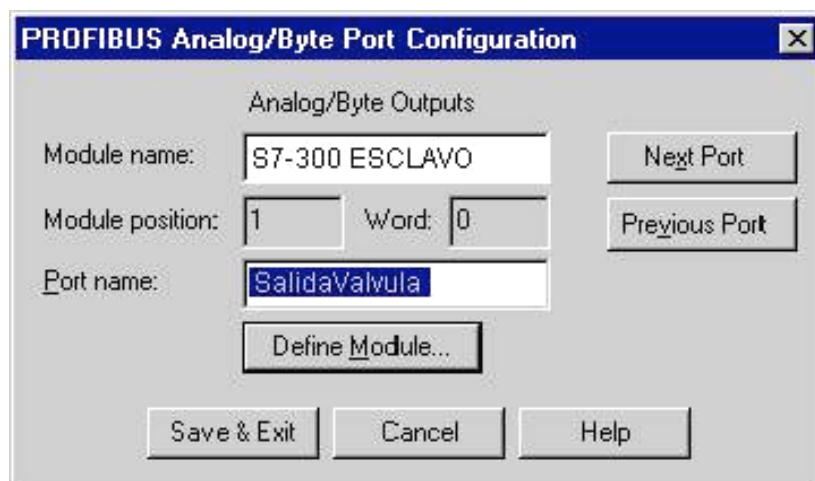
1. Para definir nuestra salida análoga se pulsa en el botón **Analog Output** y se definió un valor de 1 en el cuadro **Module Word**.

2. Luego se pulsa en el botón **Define Ports** para definir los puertos de salida análogo a utilizar en nuestra aplicación. Con esta opción se incorporaron los símbolos I/O creados en nuestros programas InControl (**PID, ON-OFF**).

La figura 80 muestra el cuadro de diálogo **PROFIBUS Analog Port Configuration** para salidas análogas.

- a. Se ingresa un nombre significativo para el puerto (**OutPortM11**) que contiene el nombre del tag name (SalidaValvula) para el canal de salida análogo.

Figura 80. Cuadro de dialogo PROFIBUS Analog/Byte Port Configuration

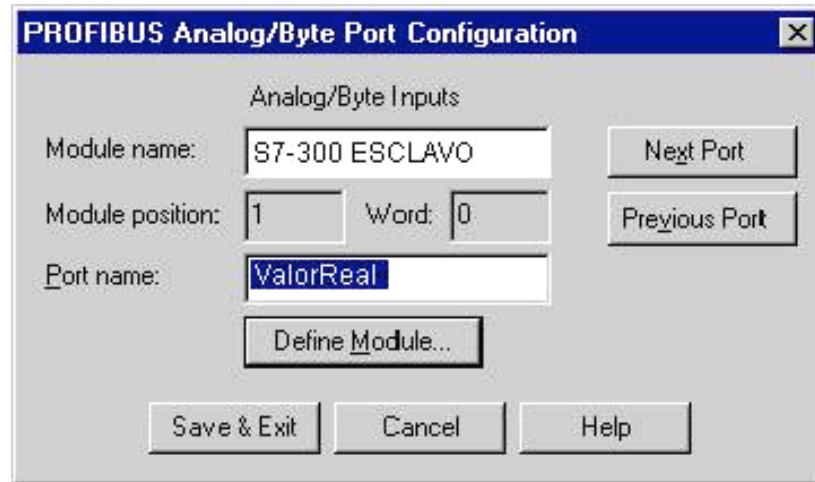


- b. Se pulsa en **Save & Exit** para guardar y completar la configuración I/O digital. Se pulsa en **Define Modules** para regresar al cuadro de dialogo **PROFIBUS Module Configuration** para continuar con la configuración del resto de puertos I/O.

Para la configuración de entradas análogas:

1. Para definir nuestra entrada análoga se pulsa en el botón **Analog Input** y se definió un valor de 1 en el cuadro **Module Word**.
2. Luego se pulsa en el botón **Define Ports** para definir los puertos de entrada análoga a utilizar en nuestra aplicación. Con esta opción se incorporaron los símbolos I/O creados en nuestros programas InControl (**PID, ON-OFF**). La figura 81 muestra el cuadro de dialogo **PROFIBUS Analog/Byte Port Configuration** para entradas análogas.
 - a. Se ingresa un nombre significativo para el puerto (**InPortM10**) que contiene el nombre del tag name (ValorReal) para el canal de entrada análogo.

Figura 81. Cuadro de diálogo PROFIBUS Analog/Byte Port Configuration



- b. Se pulsa en **Save & Exit** para guardar y completar la configuración I/O digital. Se pulsa en **Define Modules** para regresar al cuadro de diálogo **PROFIBUS Module Configuration** para continuar con la configuración del resto de puertos I/O.

Definidos todos los Tag Names análogos y digitales para nuestra aplicación, pueden ser ahora utilizados en nuestros programas (ON-OFF, PID).

3.2.2.5 Programación del PLC S7-300 como esclavo DP. Dispositivos necesarios:

- PLC S7-300 CPU314C-2DP mas cables de configuración y conexión a bus.
- Software de configuración STEP 7 Versión 5.1 + Paquete de servicio 2.

3.2.2.5.1 Definición de las Señales de Entrada y Salida en el PLC. Para el control del sistema a través del PLC se necesitan las siguientes entradas y salidas (digitales o análogas).

Cuadro 9. Salida y entradas del PLC

Salidas y Entradas				
Nº	Nombre	Descripción	Tipo	I/O
1	Bomba	Salida digital hacia el rele que alimenta o desconecta la bomba	Sal. Digital	A0.0
2	Señal_control_valvula	Señal de control de la válvula (4 - 20mA)	Sal. Análoga	PAW2
3	Señal_sensor_nivel	Señal análoga del sensor de nivel	Ent. Análoga	PEW0

3.2.2.5.2 Configuración del PLC como Esclavo.

Materiales necesarios:

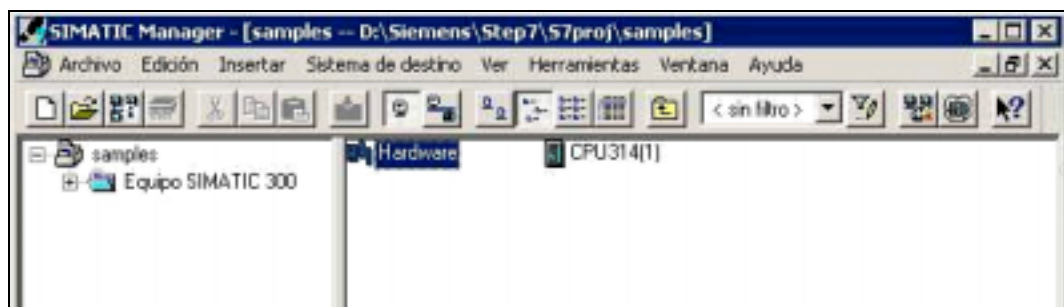
- **PLC S7-300 CPU 314C-2DP cables de configuración.**
- **Software STEP7 versión 3.1**

Después de configurada la tarjeta SMS CIF30-DPM como maestro procede a configurar el PLC S7-300 con CPU 314C–2DP como esclavo, lo cual permitirá establecer una comunicación entre este y el puerto Profibus DP embebido en la CPU correspondiente a la tarjeta SMS CIF30-DPM.

Antes de haber configurado el PLC como esclavo, se define la parte de hardware de este, esto se realiza en el software de configuración de Siemens (**HW config**) como se muestra a continuación:

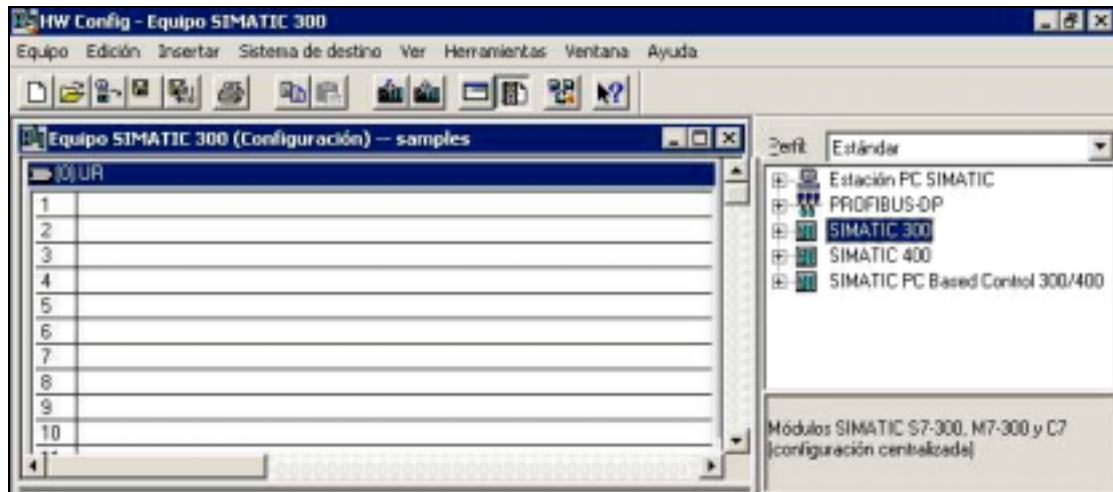
El primer paso es iniciar el Programa **Step 7** y crear un nuevo proyecto (**Samples**), una vez creado este, se hace doble clic en la opción **Hardware** para su configuración.

Figura 82. Pantalla de inicio del proyecto creado



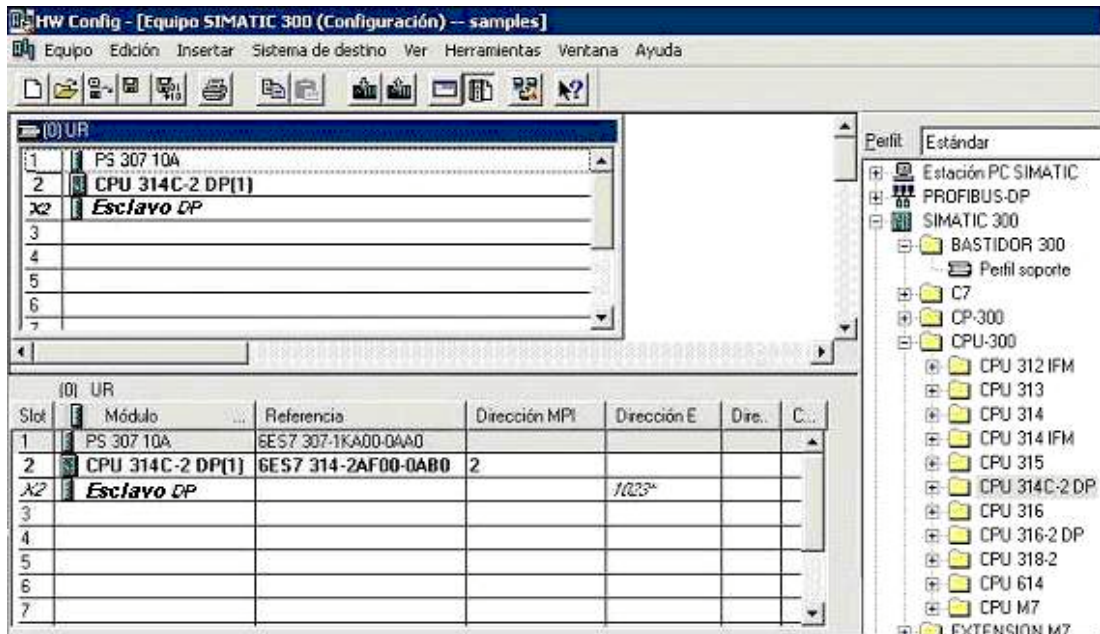
Aparece la siguiente ventana de inicio del software de configuración de Siemens (HW Config).

Figura 83. Ventana de inicio del software HW Config



Se seleccionan los elementos de hardware que hacen parte de nuestro PLC como son: la fuente de alimentación (**PS 307**), la CPU (**314C-2 DP**) y el tipo de protocolo de mensajes (**Esclavo DP**) (Figura 84).

Figura 84. Ventana de presentación de hardware para nuestro proyecto.



Una vez escogido el hardware perteneciente a nuestro PLC, se hace clic derecho sobre el tipo de mensajería a trabajar en nuestro proyecto (**Esclavo DP**) donde se establecen los parámetros del PLC para una configuración DP.

Como primer paso se define la dirección física del PLC a trabajar (**2**), el tipo de enlace de transmisión (**DP**) y la tasa de transmisión (**3 Mbit/s**) (Figura 85).

Como segundo paso se define el modo operativo del PLC esclavo DP (Figura 86).

Figura 85. Primer paso para la configuración del modo operativo

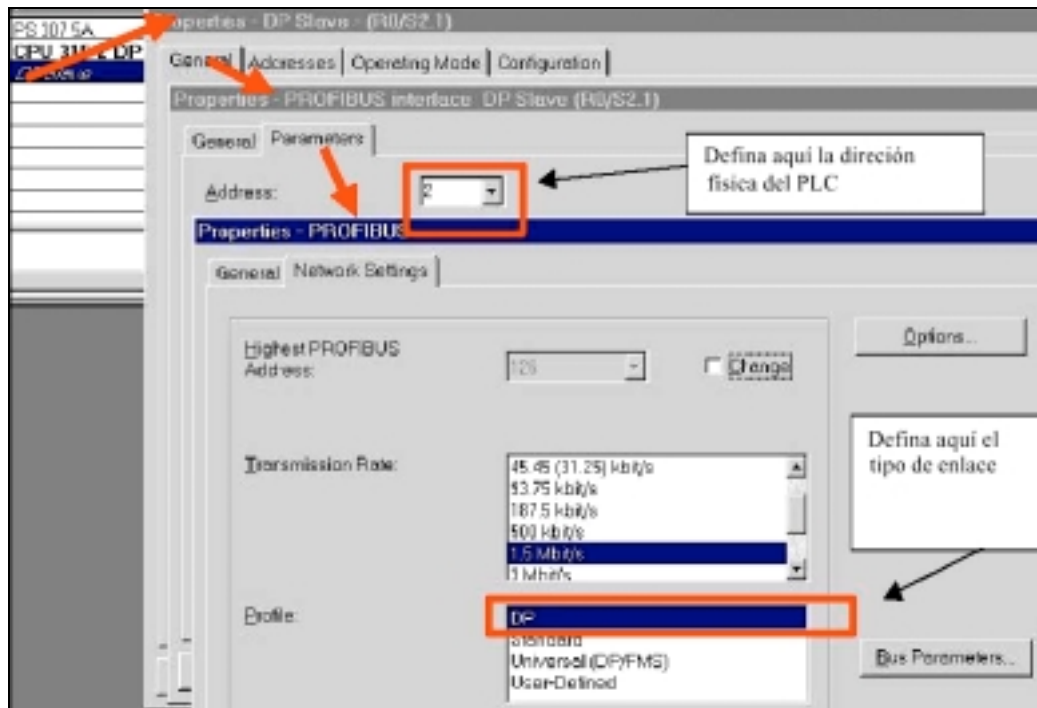
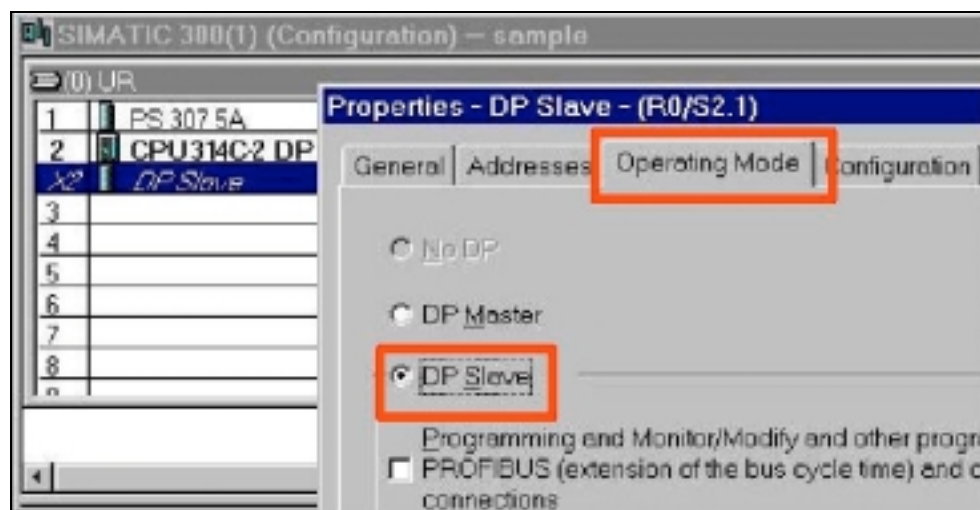
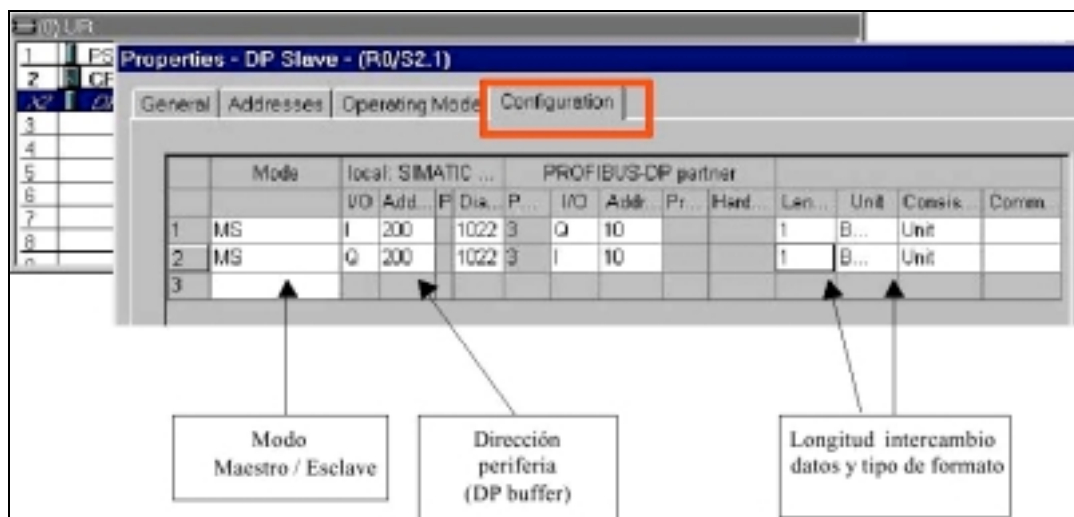


Figura 86. Segundo paso para la configuración del modo operativo.



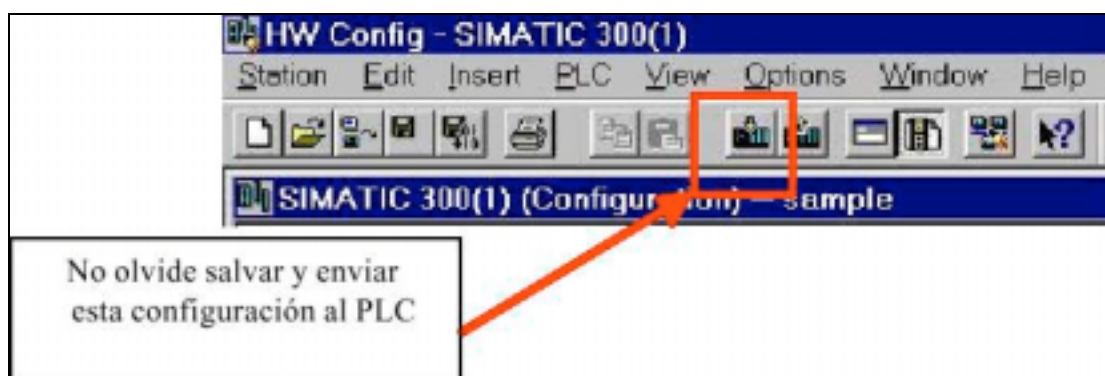
Como tercer paso se realiza la configuración del modo operativo.

Figura 87. Tercer paso para la configuración del modo operativo.



Por ultimo se guarda y envía la configuración al PLC S7-300.

Figura 88. Cuarto paso para la configuración del modo operativo



3.2.2.5.3 Elaboración del programa en el software Step 7. Después de realizada la configuración del PLC S7-300 como esclavo DP se debe realizar un programa que permita copiar los buffers de INPUT y OUTPUT. Los programas son creados con ayuda del lenguaje KOP perteneciente al software **Step 7**.

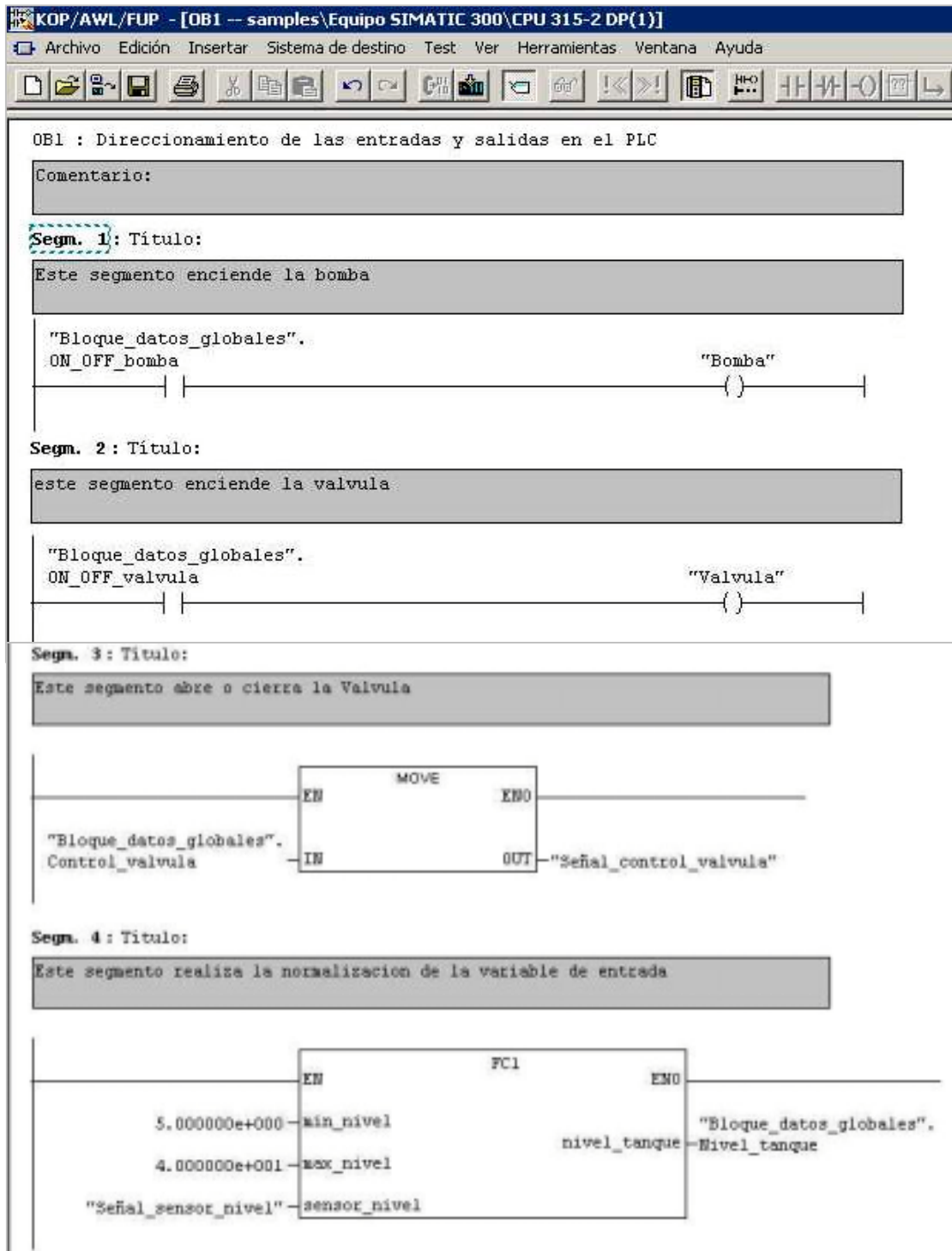
Programa de direccionamiento de las entradas del PLC. Se diseña el siguiente programa para poder leer y escribir en los puertos físicos del PLC a través de la tarjeta SMS CIF30-DPM. Primero se definen las variables de entrada y salida por medio de los símbolos a usar en el programa, descritos en una tabla de símbolos (Figura 89) dentro del programa, donde se definieron los tipos de datos y sus direcciones.

Figura 89. Tabla de símbolos

	Símbolo	Dirección	Tipo de datos	Comentario
1	Cycle Execution	OB 1	OB 1	Programa principal
2	Bomba	A 0.0	BOOL	Salida digital hacia el relé que alimenta o desconecta la bomba
3	Valvula	A 0.1	BOOL	Salida digital hacia el relé que alimenta o desconecta la valvula
4	Bloque_datos_globales	DB 1	DB 1	El DB1 almacena datos que proviene del programa INCONTROL y del PLC
5	Bloque_normalizacion	FB 1	FB 1	El FB1 normaliza la señal de corriente del sensor a valores en centímetros
6	MD0	MD 0	DINT	Almacena la señal del sensor de nivel como un valor Entero Doble
7	MD4	MD 4	REAL	Almacena la señal del sensor de nivel como un valor Real
8	Señal_control_valvula	PAW 2	WORD	Señal de control de la valvula, la cual debe ser sólo 4 o 20mA
9	Señal_sensor_nivel	PEW 0	INT	Señal analoga del sensor de nivel
10				

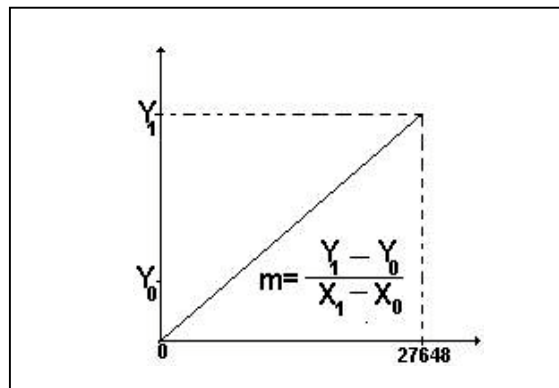
Después se realiza el programa en KOP (Figura 90), para recibir y enviar datos a través de los puertos físicos del PLC.

Figura 90. Programa KOP



Programa de Normalización de la señal del sensor de Nivel por medio del Bloque FC1. Se crea el programa correspondiente a la función de Normalización (Figura 92) de la señal del sensor de nivel representado por el bloque **FC1**, éste bloque convierte unidades Siemens (0-27648) en valores reales correspondientes al nivel del tanque (0-80cm) respectivamente, esto porque el PLC internamente representa de esta forma cualquier valor que recibe o envía.

Figura 91. Normalización de la señal del sensor



$$Y = m \cdot X + b$$

Y = Valor real del nivel del tanque.

$$m = 0,00289351851.$$

$$b = 0.$$

$Y_0 = \text{Min_Nivel}$ (setpoint mínimo aceptado por el sistema).

$Y_1 = \text{Max_Nivel}$ (setpoint máximo aceptado por el sistema).

X_0 y X_1 valores máximos y mínimos del sensor de nivel representados en unidades Siemens (0 a 27648) respectivamente.

Figura 92. Programa del bloque FC1

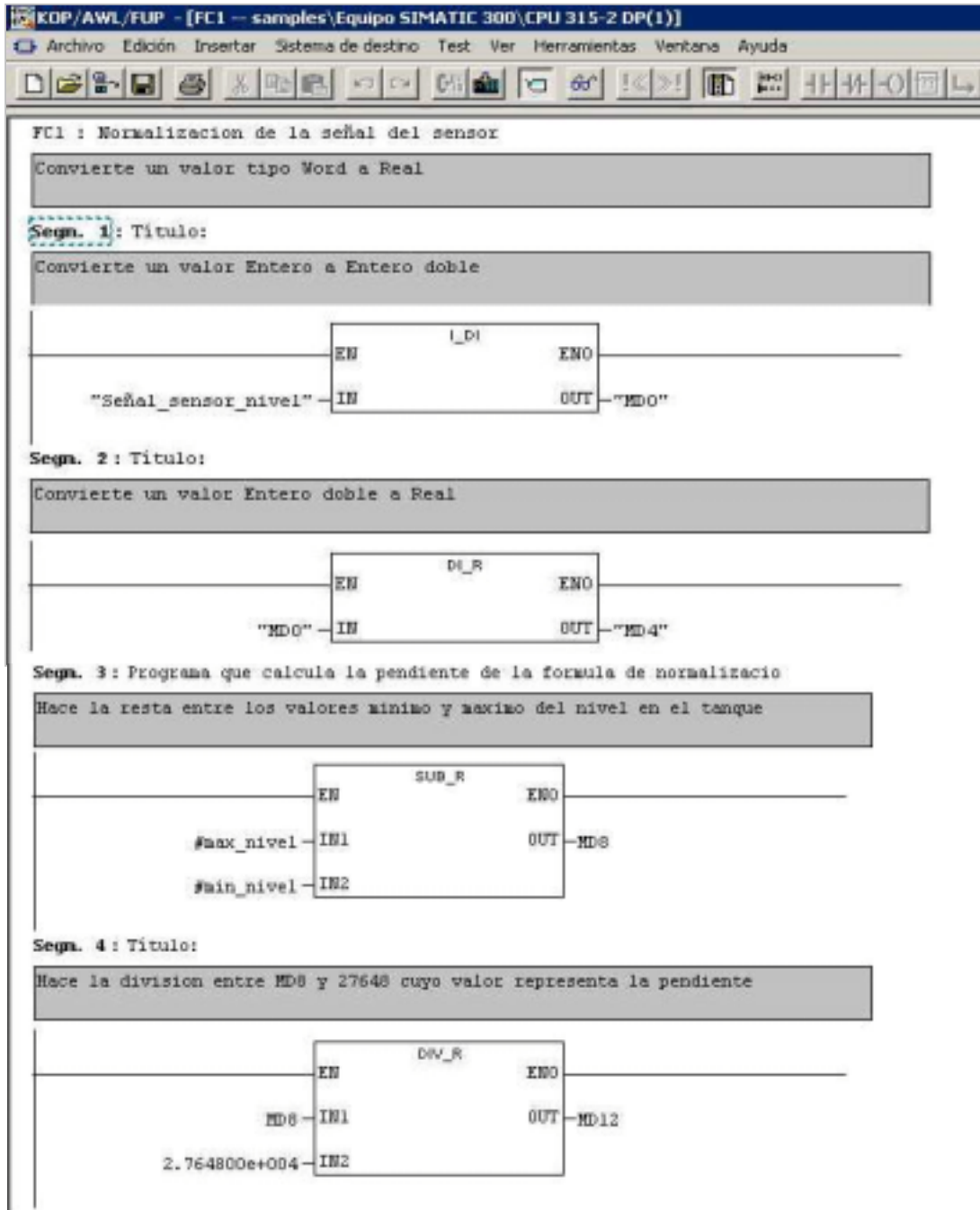
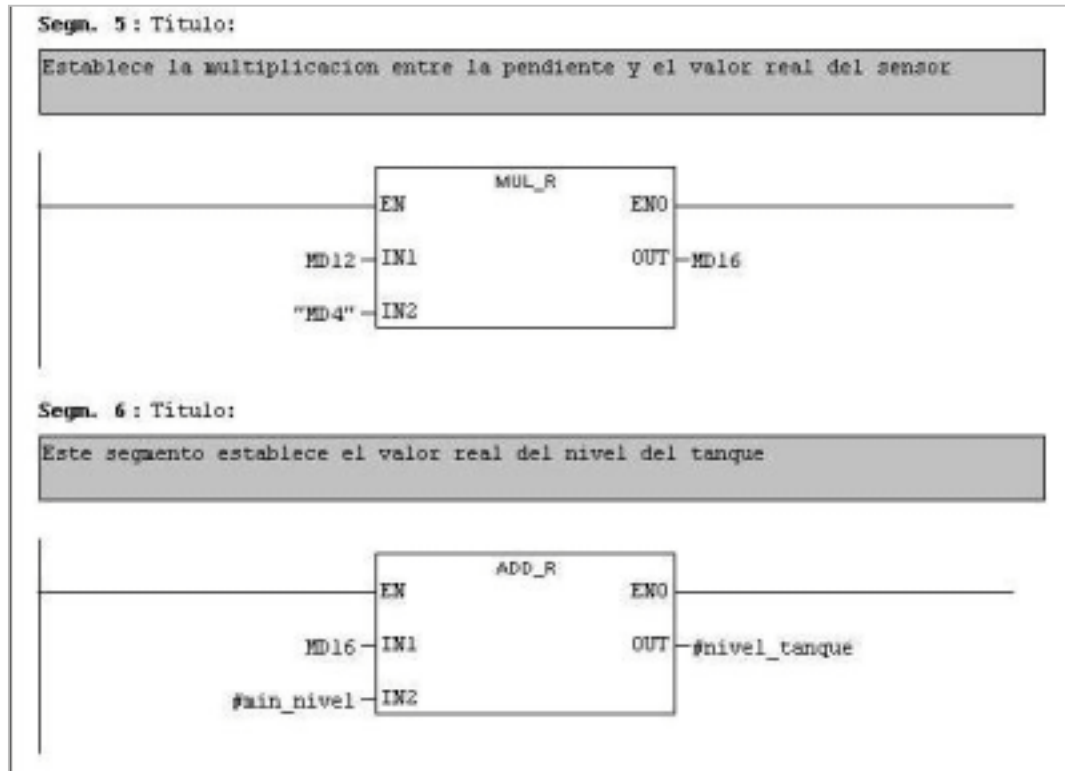


Figura 93. Continuación del programa del bloque **FC1**



Con este programa la variable de salida **"Nivel_tanque"** es tomada y procesada por el programa creado en InControl para realizar el respectivo control (ON-OFF ó PID).

Por último se crea el bloque de datos perteneciente a los programas **"Programa de direccionamiento de las entradas del PLC"** y **"Programa de normalización de la señal del sensor de nivel por medio del bloque FC1"**.

Figura 94. Bloque de datos

Dirección	Nombre	Tipo	Valor inicial	Comentario
0.0		STRUCT		
+0.0	ON_OFF_bomba	BOOL	FALSE	Estado de la bomba (ON/OFF)
+0.1	ON_OFF_valvula	BOOL	FALSE	Estado de la válvula (ON/OFF)
+2.0	Control_valvula	WORD	W#16#0	Control de la válvula (máx o mín apertura)
+4.0	Nivel_tanque	REAL	0.000000e+000	Nivel actual del tanque
+8.0		END_STRUCT		

3.2.2.6 Cableado y Equipo Requerido. Materiales necesarios para el diseño del cable de media de red Profibus:

- Cable RS-485 Par trenzado blindado de cobre con un par conductor.
- Cables de derivación (preferiblemente de 1m o menos (precher+Shuh))
- Conectores de junta en T (precher+Shuh).
- Cables de segmento de bus (Srecher+Shuh).
- Conector (precher+Shuh).

Un cable Par trenzado blindado de cobre con un par conductor (interfase RS-485), es necesario para conectar la tarjeta CIF30-DPM al PLC S7-300, y así poder recibir los datos provenientes del sistema de control (proceso).

RS-485 o EIA-485-A, es un estándar eléctrico diferencial o balanceado, porque usa diferencias de voltaje para determinar las señales lógicas. No obstante, RS-

485 es una configuración multipunto. Esto es, puede soportar múltiples drivers y múltiples receptores. Es una interfase bidireccional que puede soportar más de 32 unidades de carga, donde una carga es de $4K\Omega$. RS-485 es la mejor opción para aplicaciones de red Profibus. La tasa de datos y la longitud del cable del RS-485 son similares al estándar RS-422. RS-485 puede enviar datos por encima de 10 Mbps y a más de 4000 pies.

Ventajas del RS-485:

- Aplicaciones Multipuntos
- Bajo costo
- Inmunidad al ruido

Desventajas del RS-485:

- Requiere más cableado
- No es muy común

Cables. Se necesita un cable doble trenzado blindado para hacer el cableado (segmentos de bus o cables de derivación). Se puede usar cualquier cable doble trenzado blindado, los cuales pueden obtenerse en establecimientos comerciales, para conectar el banco a la red PROFIBUS y así mismo al PLC. Se recomienda el cable Sprecher+Schuh, tal como se muestra a continuación en la siguiente tabla.

Conectores en junta en T. Se necesitaran conectores de junta en T para conectar los cables de derivación a los segmentos de bus. Se puede usar cualquier junta en T, las cuales se pueden obtener en cualquier establecimiento comercial, sin embargo recomendamos conectores Sprecher+Schuh, tal como se muestra en el cuadro 10.

Conectores de bus. Se necesitara un conector de bus para conectar el banco a la red PROFIBUS DP. Puede usar cualquier conector de bus compatible, los cuales pueden obtenerse en establecimientos comerciales.

Cuadro 10. Especificaciones de cables y conectores 1.

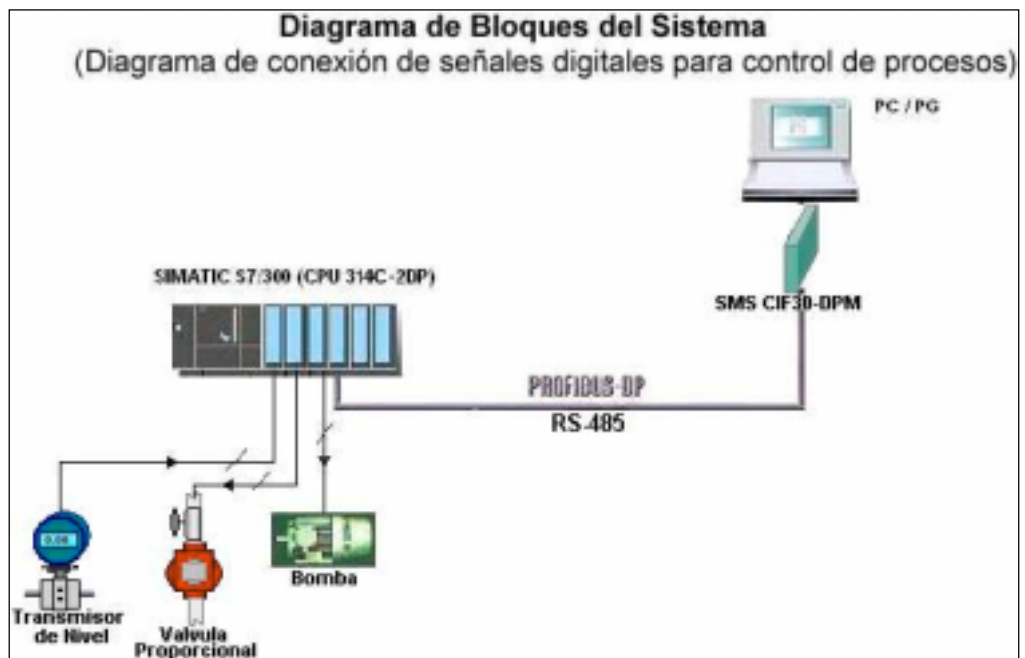
Equipo	Tipo	Numero de catalogo	Numero de parte
Cable de derivación	Belden Sprecher+Schuh	3079A PDC-2, PDC-4	87.890.282-10
Conector de junta en T	Sprecher+Schuh	PTS-D	97.890.276-01
Cable de segmento de bus	Sprecher+Schuh	Ninguno	299.257.001
Segmento de bus 1m	Sprecher+Schuh	PCB-10	87.890.281-10
Conector	Sprecher+Schuh	Ninguno	87.890.283-01

En nuestra red PROFIBUS (Figura 95) el banco control de nivel estará conectado al bus a través de un terminal de bus o un conector de conexión a bus. La tecnología de transmisión RS-485 es muy fácil de manejar, la instalación del cable par trenzado no requiere un conocimiento experto. El estándar EN 50170 especifica usar línea tipo A de cable par trenzado blindado con las siguientes especificaciones:

Tabla 3. Especificaciones de línea tipo A

Parámetros	Valor
Impedancia	135 a 165 Ω
Capacitancia por unidad de longitud	< 30 pF/m
Resistencia de lazo	110 Ω /km
Diámetro del núcleo	0.64 mm

Figura 95. Diagrama de conexión del sistema de red Profibus.



3.2.3 Visualización del sistema en el software InTouch. Para monitorear y visualizar el sistema Control de Nivel se utiliza el *software* de supervisión *InTouch*. Se crea el mímico del banco de control y sus respectivas animaciones para luego comunicarlo con el *software InControl* (el controlador), el cual recibe los datos del PLC, los procesa y los envía al *InTouch* para que éste muestre dichos datos y los resultados.

Los siguientes pasos se siguieron para la elaboración de la aplicación y cada ventana que la conforma:

1. Creación y configuración de la aplicación.
2. Creación y configuración de las ventanas.
3. Creación de los gráficos o mímico.
4. Vinculación de animaciones y configuración de *tagnames*.
5. Creación de *scripts*.

3.2.3.1 Creación y configuración de la aplicación. Primero se ejecuta el software *InTouch* desde la barra de tareas de inicio en el icono *InTouch* (Figura 98), de esta manera se ingresa a la ventana “*Application Manager*” (Figura 99) desde donde se crean y administran las aplicaciones de *InTouch*.

Figura 98. Ejecución del InTouch

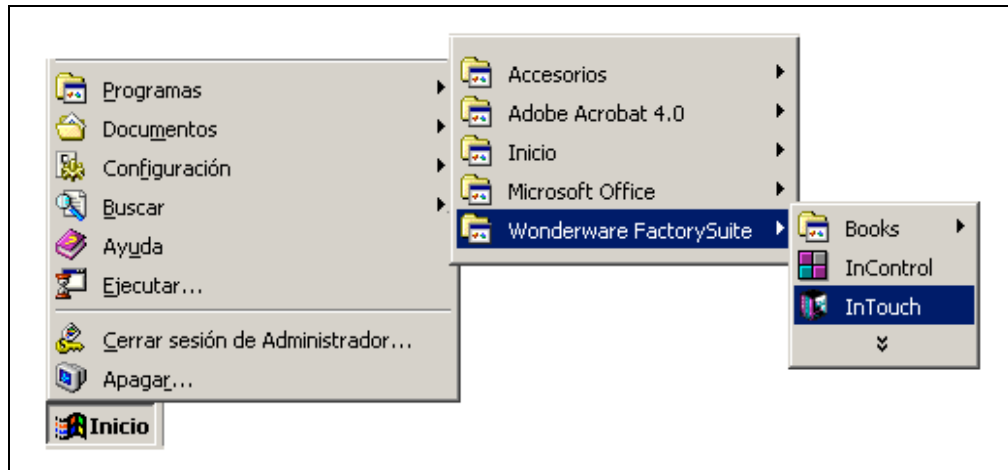
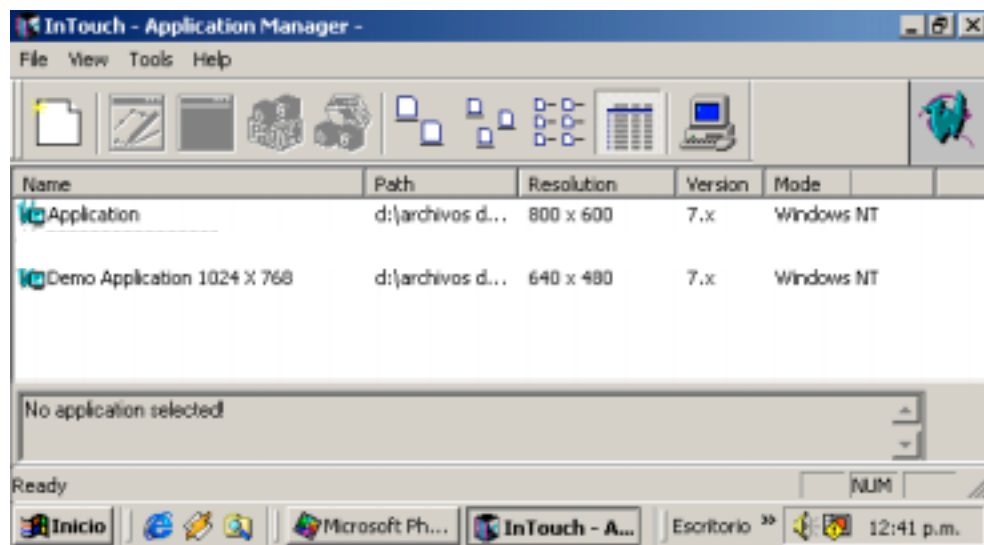
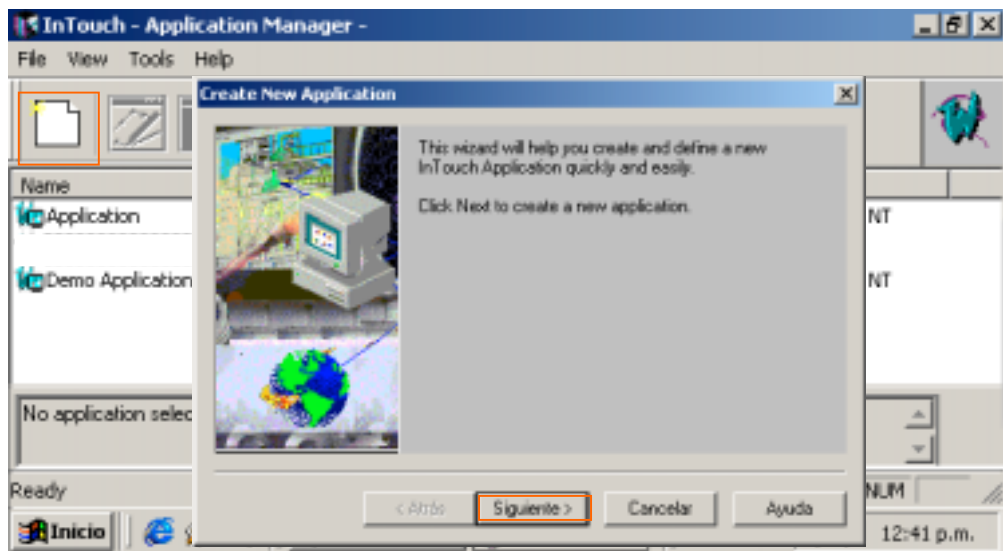


Figura 99. Ventana Application Manager



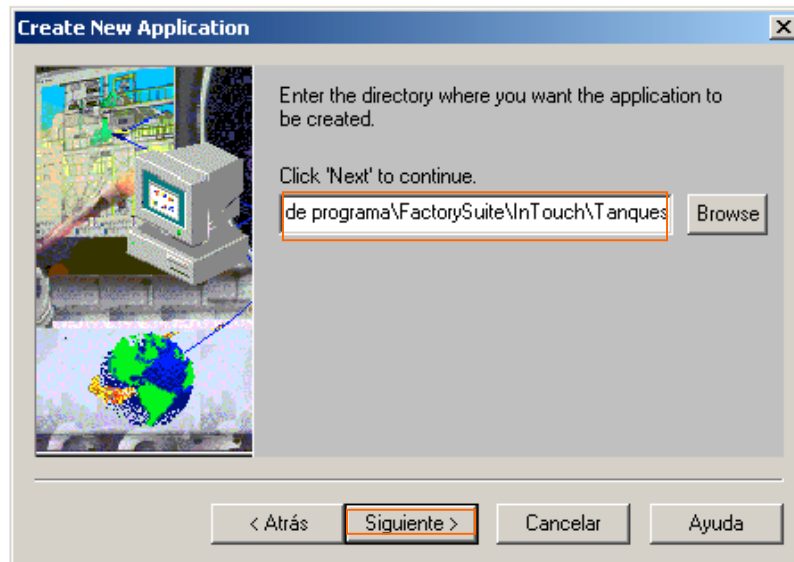
Para crear la aplicación dentro del *Application Manager* de InTouch se pulsa el botón *New* (Figura 100), luego aparece la ventana *Create New Application*, un wizard que permite definir el nombre y la dirección de la aplicación dentro del sistema. Luego de pulsar en el icono **siguiente** de la ventana (Figura 100) se ingresa la dirección donde se quiere que la aplicación sea creada (Figura 101).

Figura 100. Creación de una nueva aplicación



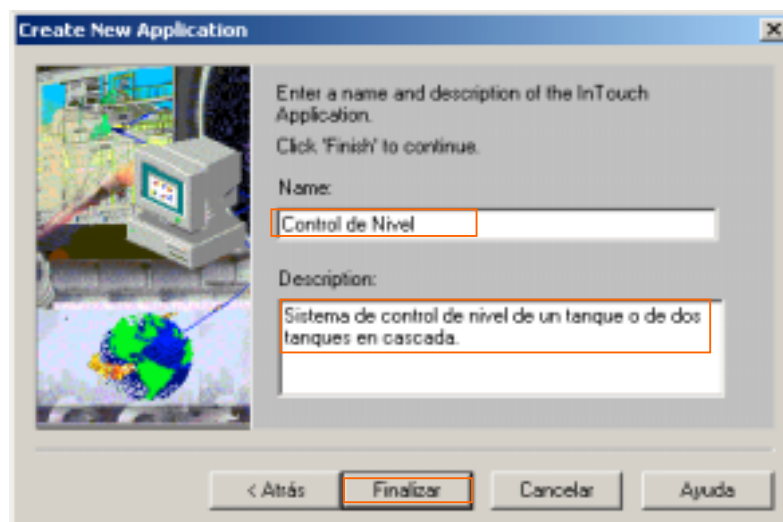
La dirección definida es “**C:\Archivos de programa\FactorySuite\InTouch\Tanques**”(Figura 101). Pulsando de nuevo el icono **siguiente** se llega a definir el nombre y una descripción para la aplicación.

Figura 101. Ingreso de la dirección de la aplicación



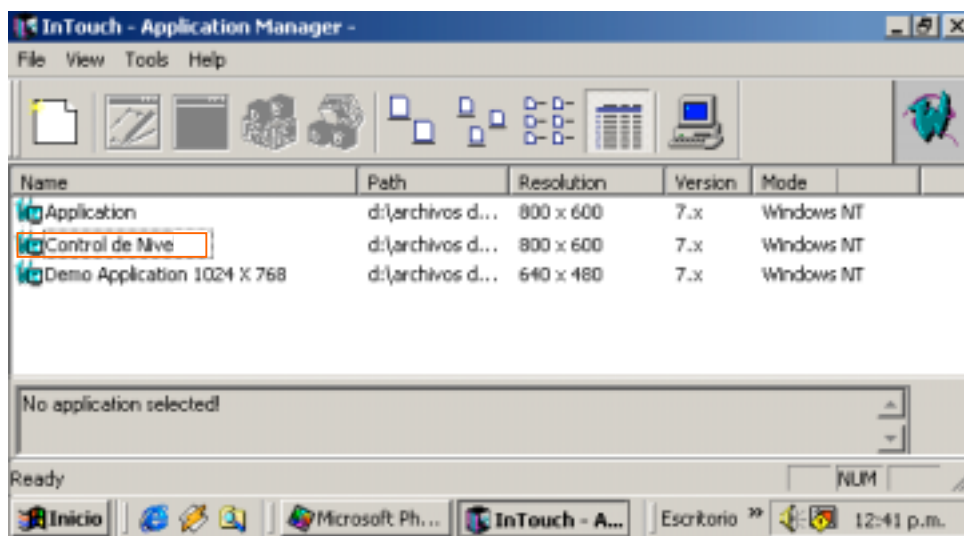
El nombre de la aplicación es “Control de nivel”. Para terminar con la configuración se hace clic en el icono **Finalizar** (Figura 102).

Figura 102. Nombre y descripción de la aplicación



De esta forma la aplicación “**Control de nivel**” entra a hacer parte de la lista de proyectos de InTouch (Figura 103) y del sistema.

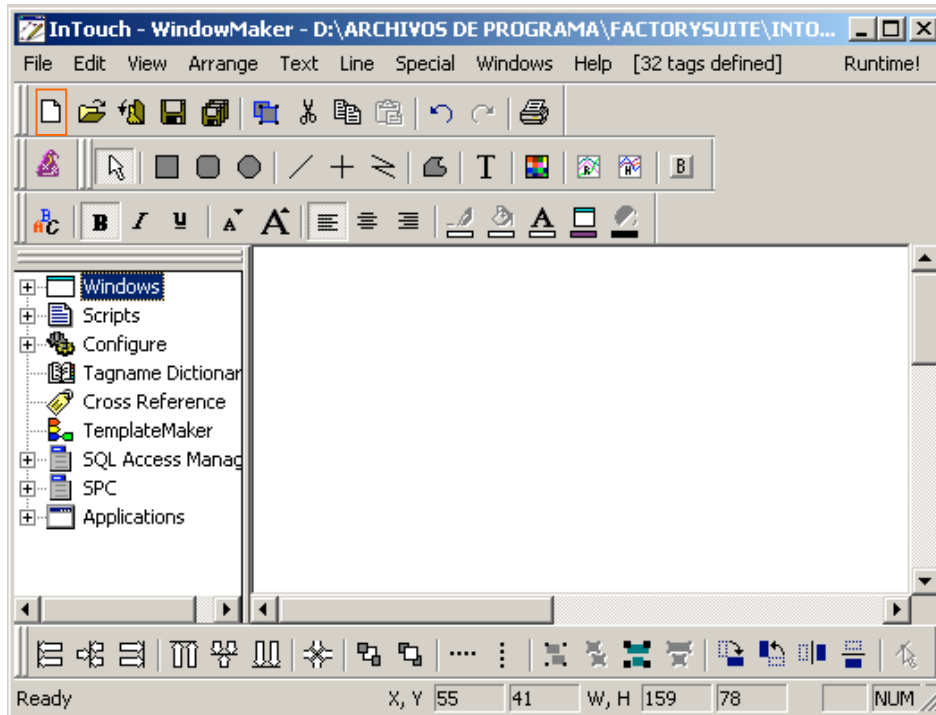
Figura 103. Presentación de la aplicación en la lista de proyectos de InTouch.



3.2.3.1.1 Creación y configuración de la ventana Banco de Nivel. Luego se ingresa al ambiente de trabajo (*WindowMaker*) de InTouch (Figura 104) pulsando dos veces sobre el nombre de la aplicación “**Control de Nivel**” y se da inicio a la creación de las ventanas que conforman la aplicación.

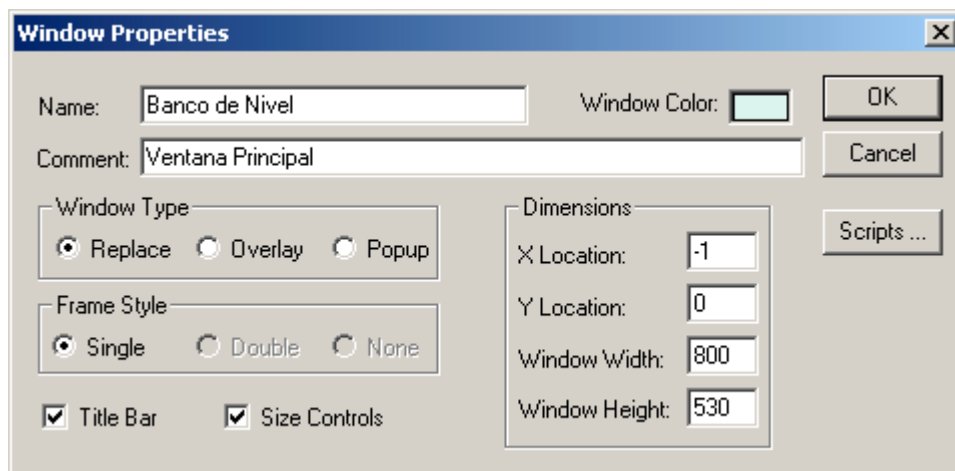
Desde el *Window Maker* se pulsa el botón **New**, posteriormente se muestra el cuadro de diálogo *Window Properties* (Figura 105) para la configuración de la ventana.

Figura 104. Ambiente de trabajo del InTouch (Window Maker)



La siguiente figura muestra la configuración de la ventana.

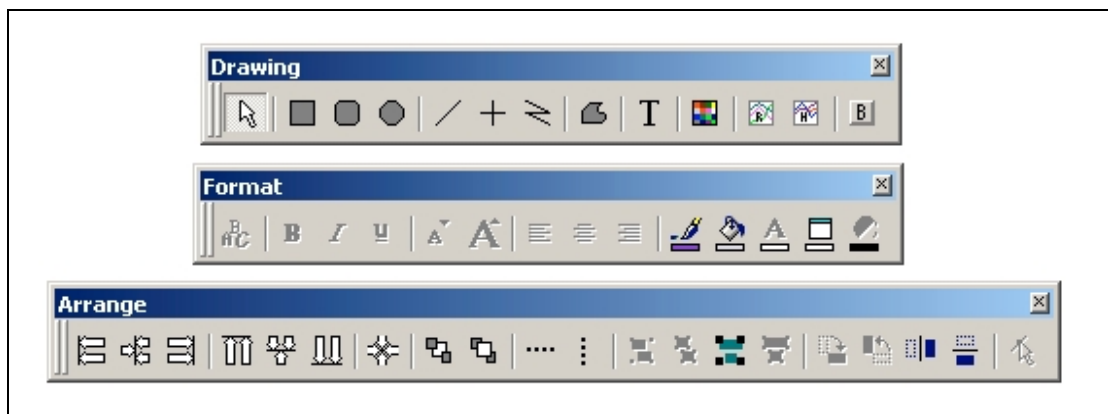
Figura 105. Configuración de las propiedades de la ventana principal "Banco de Nivel"



El nombre de la ventana principal (*name*) está definida como “Banco de Nivel”, el tipo de ventana seleccionado es *replace*, es decir, al momento de ejecutar la aplicación, la ventana “Banco de Nivel” siempre estará sobre la pantalla aunque se esté trabajando sobre otra ventana y mostrará un color de fondo verde (**Window Color**). El título de la ventana estará activa durante la ejecución (**Title Bar**) y su tamaño podrá ser modificado (**Size Controls**). Su área (**Window Width - height**) ocupa toda la pantalla del computador con 800x530 pixeles.

3.2.3.1.2 Creación de los gráficos de la ventana Banco de Nivel. Definida la ventana principal se procede a dibujar el banco tal cual como se ve en su vista frontal. Se hizo uso de las barras de herramientas *Drawing*, *Format* y *Arrange* (Figura 106) para la creación de los objetos y gráficos.

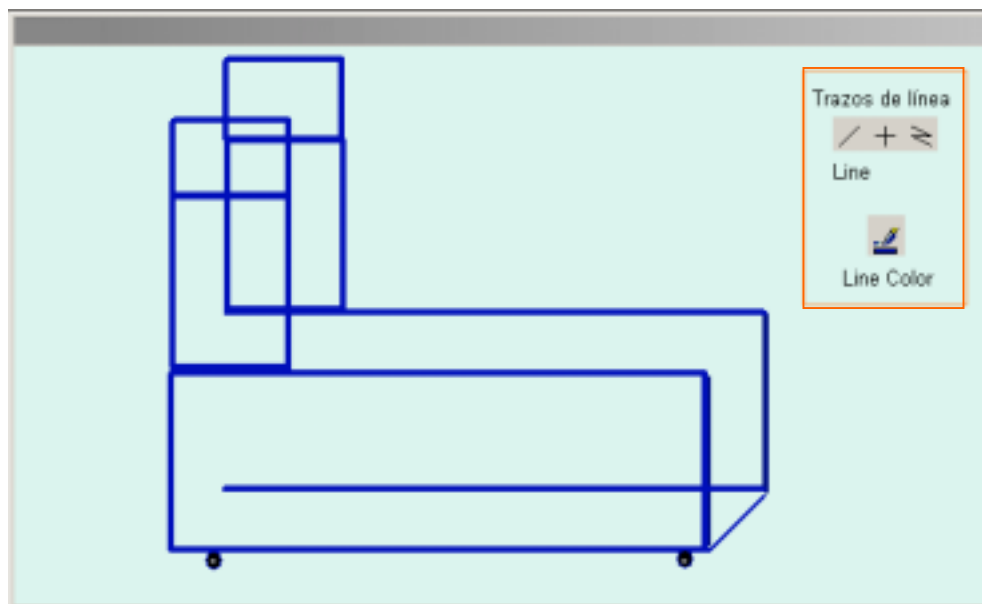
Figura 106. Barras de herramientas para la creación de gráficos.



Con la barra *Format* (Figura 106) se seleccionan los colores de relleno y contorno de las figuras, el fondo de la ventana y los textos establecidos. La barra *Arrange* (Figura 106) permite hacer los arreglos correspondientes a las posiciones de varias figuras que se seleccionan, también envía objetos delante o detrás de otros y agrupa un conjunto de figuras definiéndolas como símbolo o celda. El proceso de creación de los gráficos es el siguiente :

- Primero se traza el armazón (Figura 107) sobre el cual están montados los elementos: se pulsa el botón de trazos de línea, **Line**, ubicado en la barra de herramientas *Drawing*, se elige el icono **line color** para seleccionar el color y se apunta con el *mouse* la posición y la longitud de ésta en el área de trabajo.

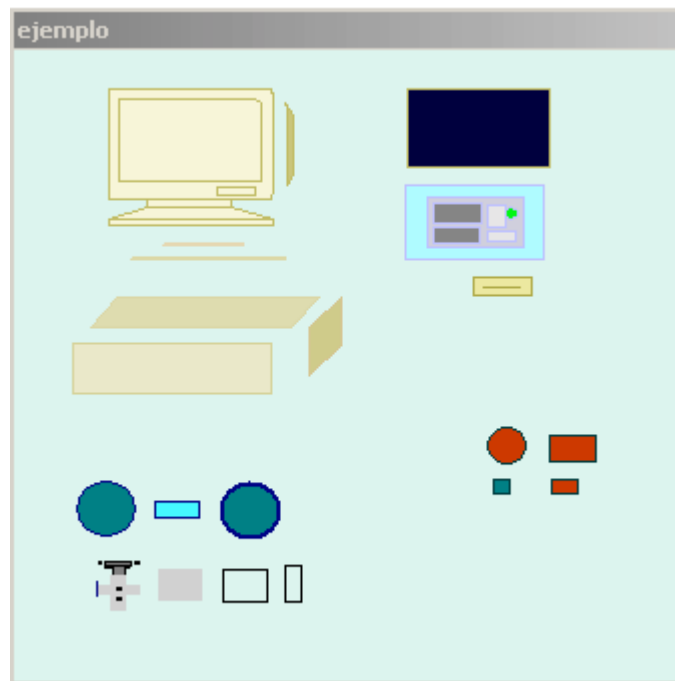
Figura 107. Estructura del banco



- Posteriormente se dispone a crear los diferentes gráficos que conforman la aplicación, los cuales son cada uno de los elementos del banco (Transmisor de Nivel, la bomba de succión, la Válvula proporcional, los textos, el computador, la estructura física y la tubería del banco) .

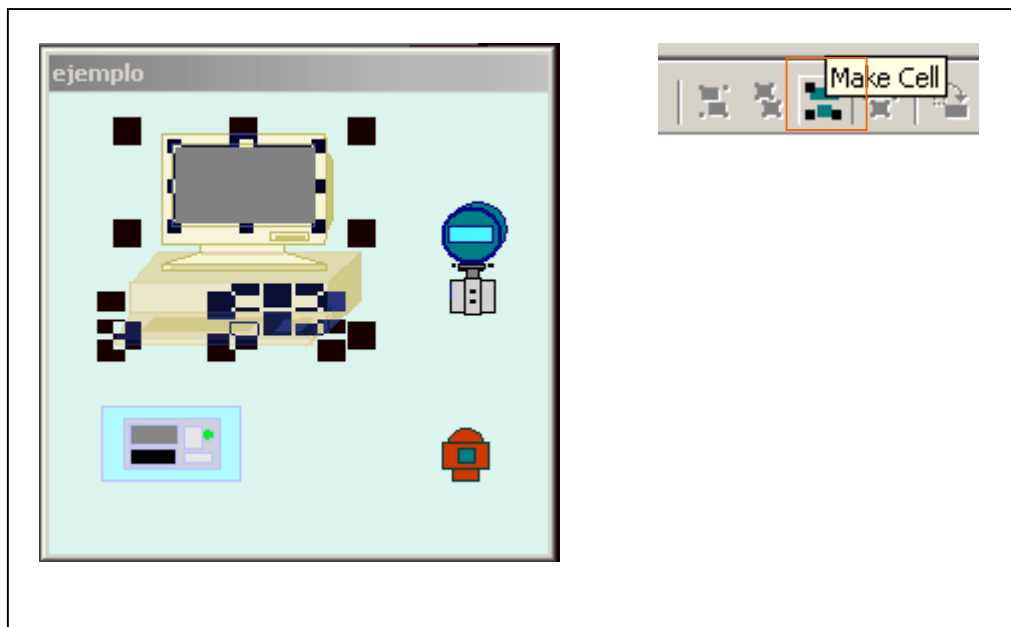
Estos gráficos se crean seleccionando las figuras de la barra *Drawing*, insertándolos sobre el área de trabajo y agrupándolos con la opción de *Make Cell*. La siguiente figura muestra los contornos necesarios para la creación del computador, la válvula y el transmisor.

Figura 108. Elementos gráficos



- Las figuras se reubican de tal forma que construyan los diversos elementos del banco de control. Posteriormente se selecciona el conjunto de figuras y se pulsa **Make Cell** para agruparlos en una misma celda y convertirlo en un objeto no animado.

Figura 109. Conversión de la figura del computador a celda.



La pantalla del computador, en cambio, necesita de una animación, de modo que se pulsa el icono **Make Symbol** para convertirlo en un símbolo y permitir la animación (Figura 110). De igual forma los objetos como el transmisor, la válvula proporcional, la válvula de tres vías, y los medidores de nivel son convertidos a símbolos. Objetos como las tuberías y el armazón se agrupan en celdas (Figura 111).

Figura 110. Conversión de la pantalla del computador a símbolo

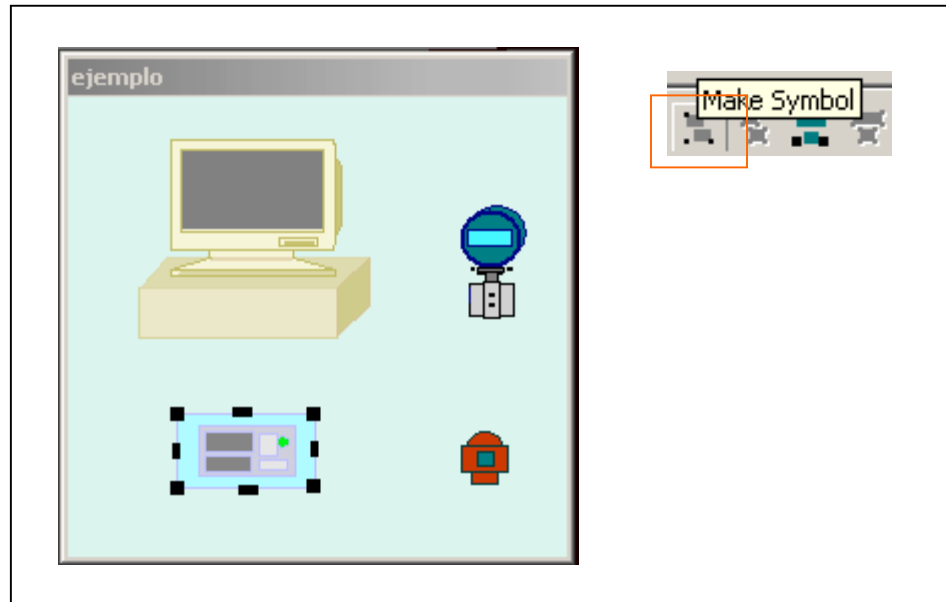
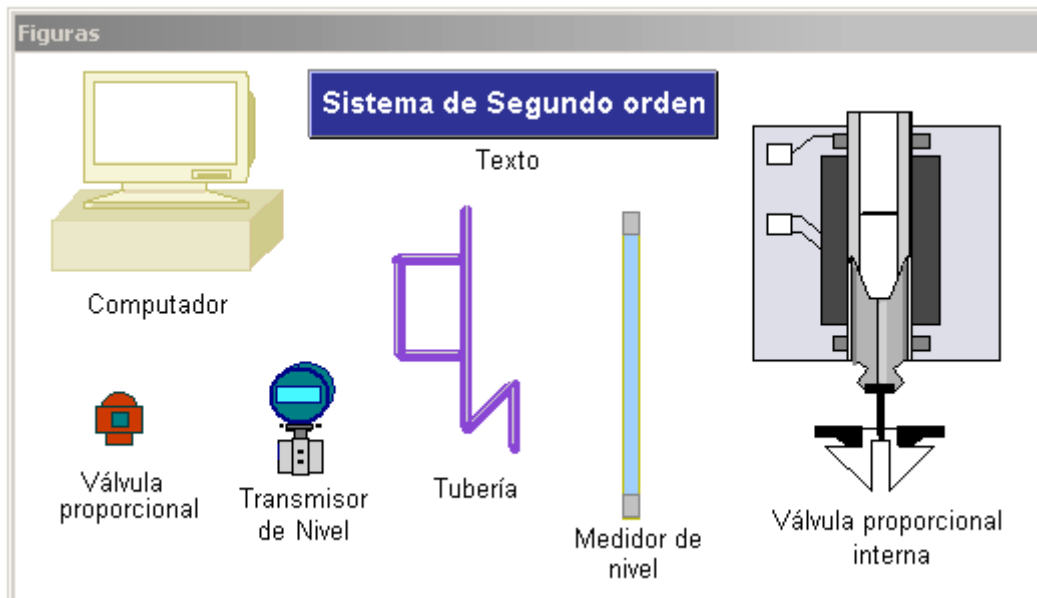


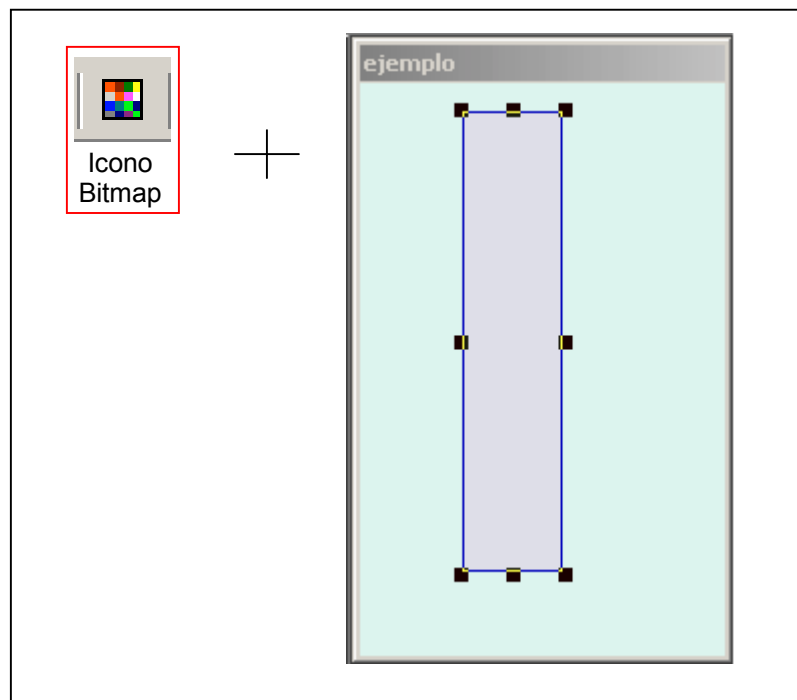
Figura 111. Ejemplos de figuras, objetos y símbolos



Posteriormente fueron insertados los gráficos de los tanques, la mesa y el tanque de depósito (Figura 112) que se crean originalmente dentro de las herramientas *Paint* y *CorelDrawn*. Su procedimiento es el siguiente:

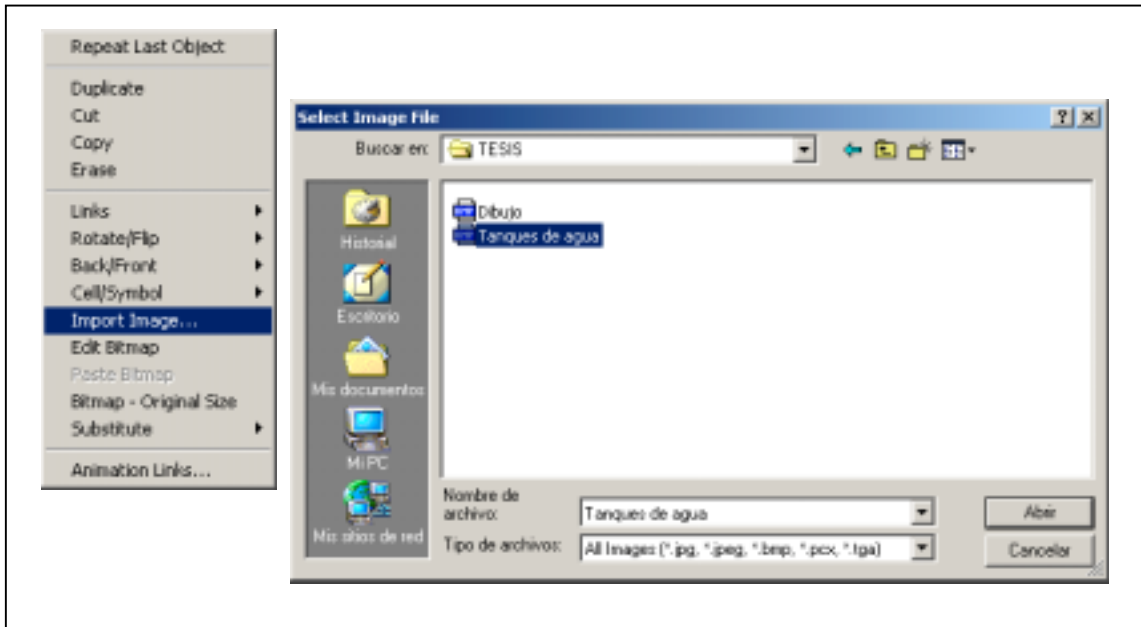
- Se pulsa el icono **Bitmap** (Figura 112).
- Se pega el cuadro sugerido por el programa sobre el área de trabajo a un tamaño deseado.

Figura 112. Insertar el Mapa de bits sobre la pantalla



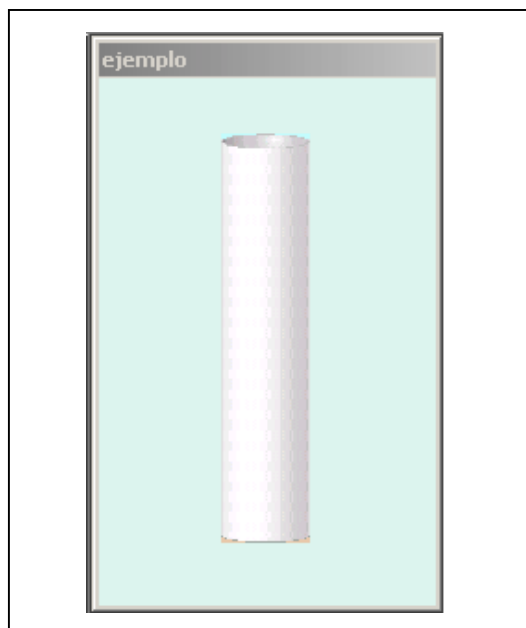
- Se pulsa el botón derecho del mouse sobre el cuadro y se selecciona la opción *Import Image* (Figura 113).

Figura 113. Seleccionar la imagen a importar



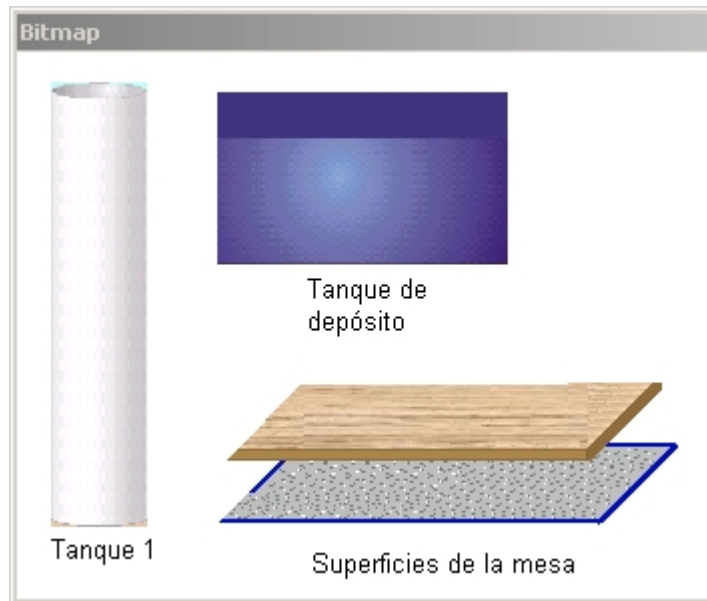
- La nueva imagen (Figura 114) reemplaza al cuadro seleccionado.

Figura 114. Imagen Importada



Las siguientes figuras corresponden a otras imágenes importadas sobre esta ventana.

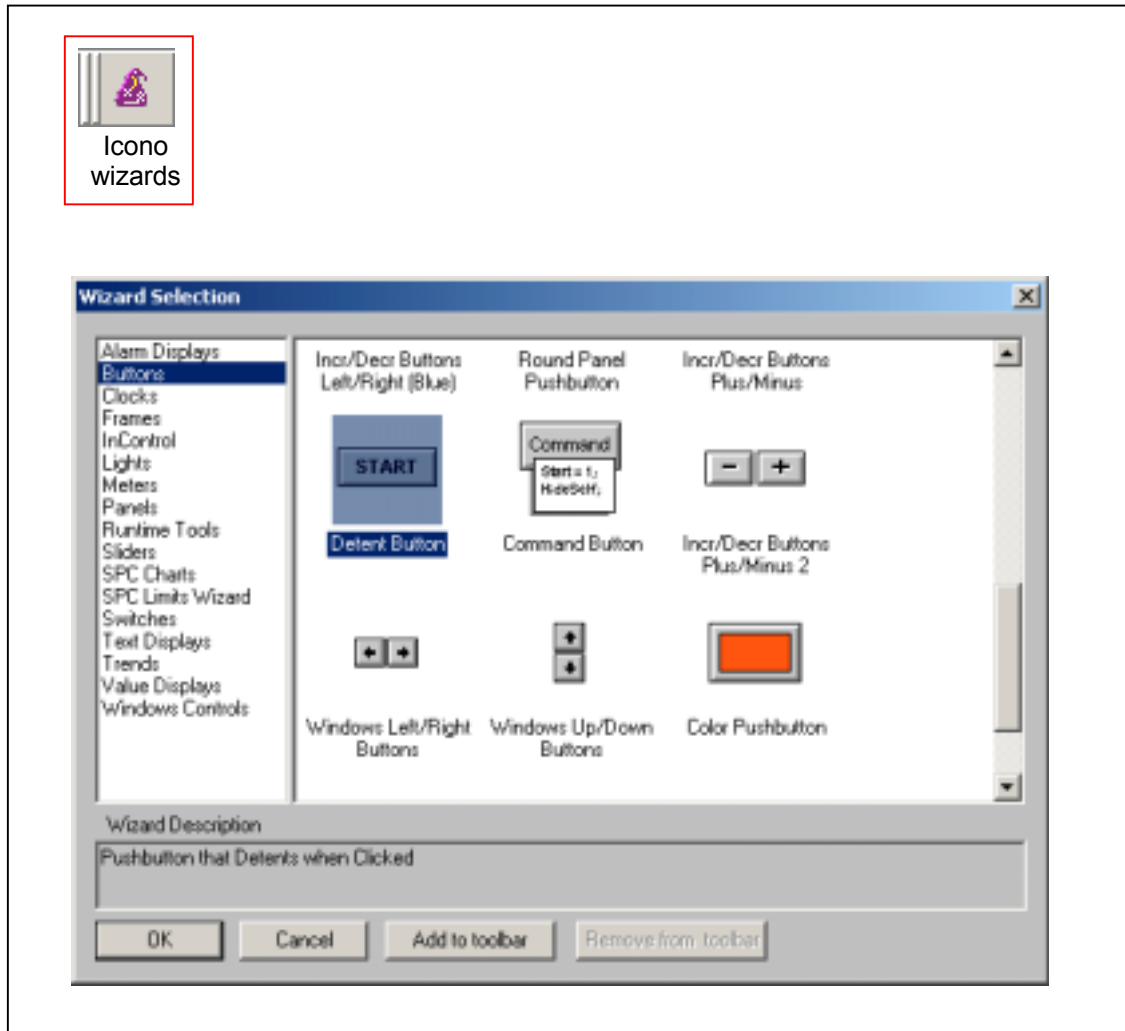
Figura 115. Imágenes importadas tipo Bitmap.



El resto de gráficos del mímico son elementos típicos de aplicación Industrial, por esta razón se emplean los gráficos prediseñados **Wizards** de InTouch.

- Pulsando el icono **wizards**, que muestra la figura 116, aparece un banco de dibujos donde se encuentran estilos de alarmas, bordes, botones, Displays, paneles, switches, objetos de fábrica como válvulas, tanques, camiones y tubos, útiles para nuestra aplicación. Un ejemplo de wizards son los botones de llenado y vaciado.

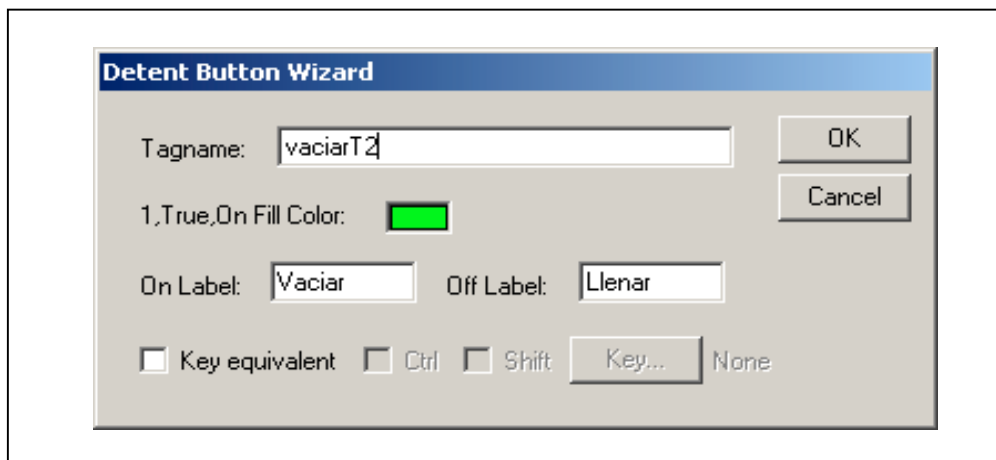
Figura 116. Insertar botones empleando wizards



- Después de seleccionar los botones de la librería de wizards, se pulsa **OK** y se ubica el cursor en el lugar donde se desea insertar el objeto.
- Se hace doble clic sobre el objeto para configurarlo, de tal manera que al pulsar sobre éste el tagname del tanque 2, etiqueta *vaciarT2*, modifique su

valor, también que el mensaje sobre él cambie de *llenar a vaciando* y de color verde, indicando que el tanque está disponible para llenar o se está vaciando.

Figura 116. Configuración de botones.



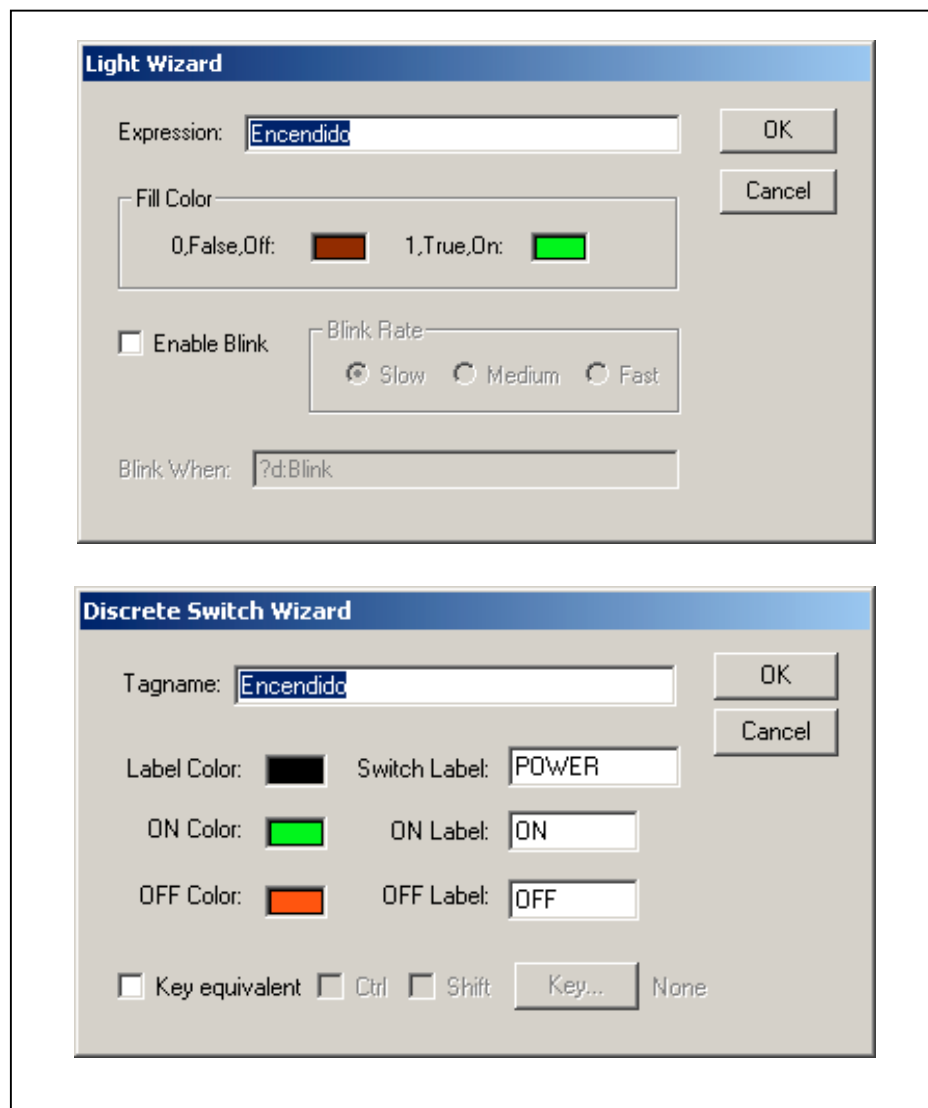
Otros wizards insertados en la aplicación son los mostrados en la siguiente figura.

Figura 35. Gráficos Wizards



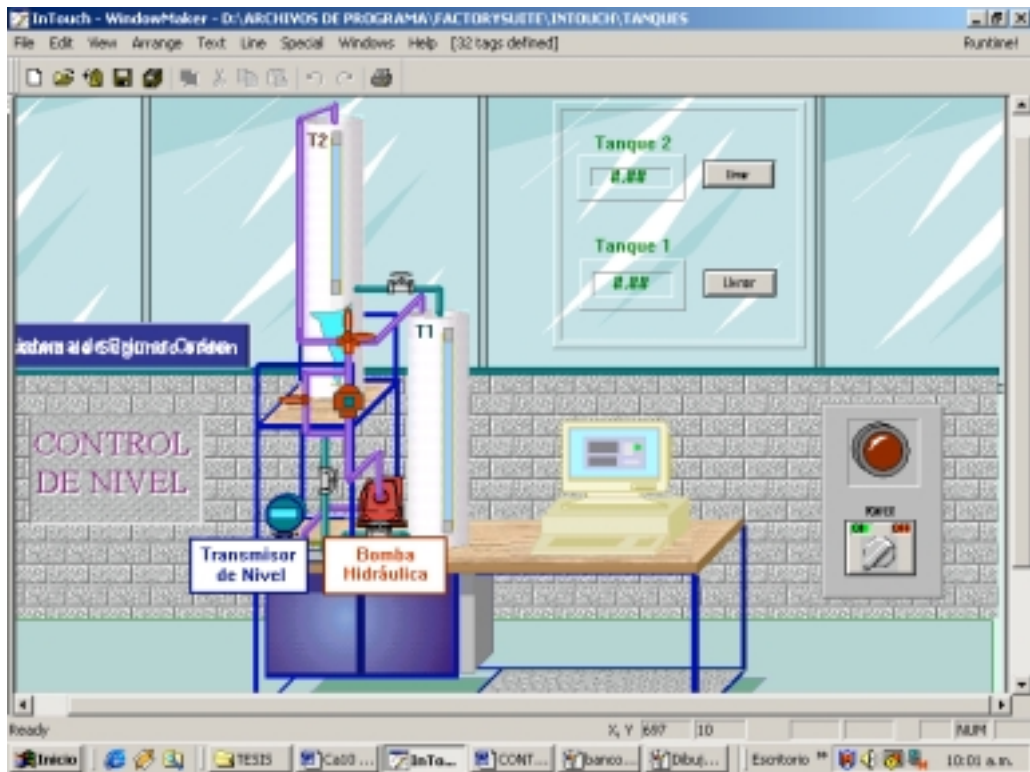
Las luces y el encendido tienen la siguiente configuración respectivamente (Figura 119).

Figura 119. Configuración de wizards



La presentación final de la ventana desde el *Window Maker* al terminar de ubicar los objetos se ve de la siguiente manera:

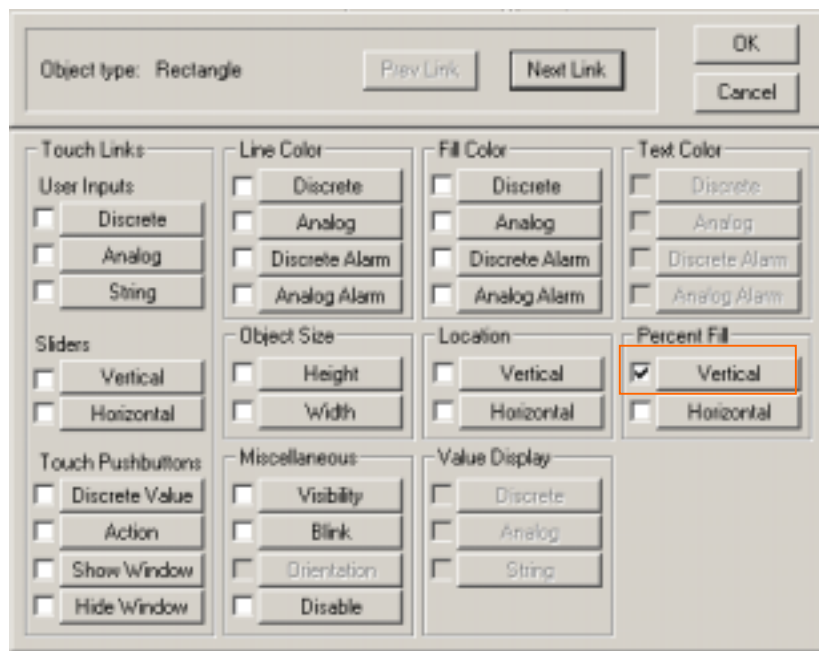
Figura 120. presentación final de la ventana banco de nivel



3.2.3.1.3 Configuración de tagnames en la ventana Banco de Nivel. Para darle vida a los objetos creados se utilizan los cuadros *Animation Links Library* y los *scripts*. El proceso general es pulsar dos veces sobre un objeto para ingresar a su librería de animaciones (Figura 122) y vincular este objeto a un *tagname* (Figura 122) para que cada vez que cambiara el valor del *tagname* las características (color, tamaño, localización, niveles de llenado, visibilidad, valor, etc.) del objeto fueran modificadas de acuerdo a la aplicación. A continuación está descrito un ejemplo de un objeto animado.

Al hacer doble clic sobre la barra de llenado del tanque aparece su cuadro **Animation Links Library** (Figura 121) correspondiente, para seleccionar los tipos de animación que son vinculados, en su caso *Porcent Fill / Vertical*.

Figura 121. Animation Links Library



Para la primera animación, el programa pide definir el nombre, la descripción y el tipo de *tagname* (Figura 123). El nombre del **tagname** designado es “**sensor1**” de tipo real como se observa en las figuras 122 y 123.

Figura 122. Configuración de una etiqueta

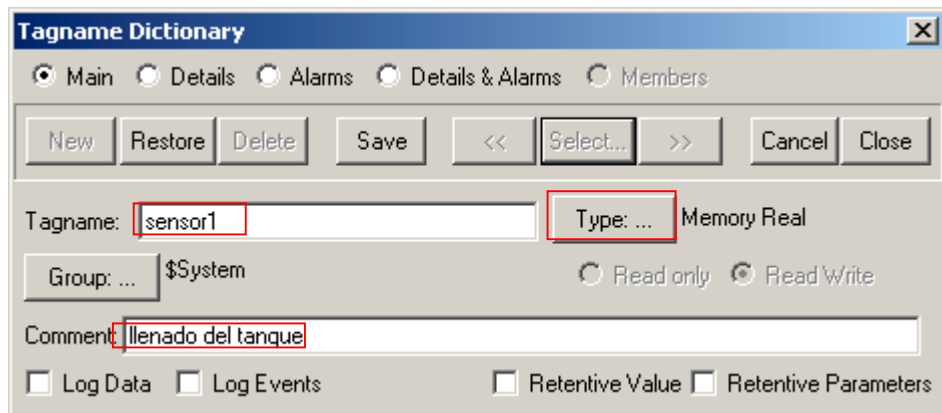
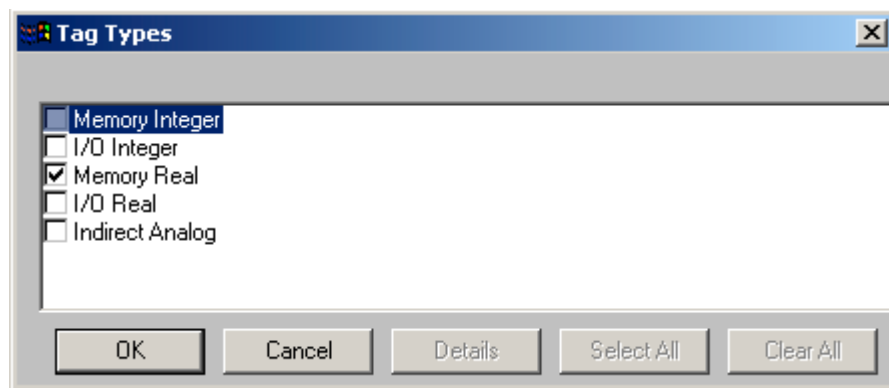
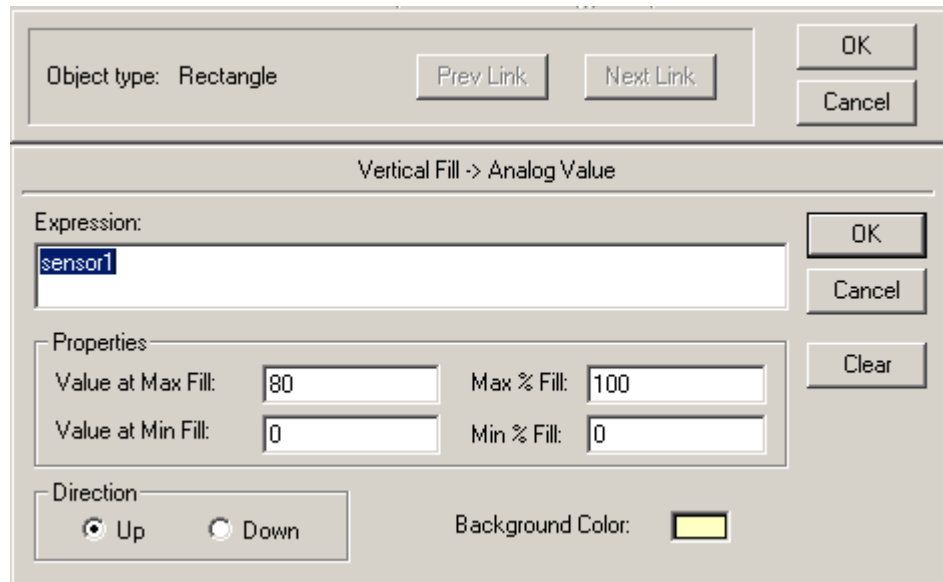


Figura 123. Selección del tipo de tagname



Luego de definir el *tagname* se sigue con la configuración de la animación como lo muestra la figura 124.

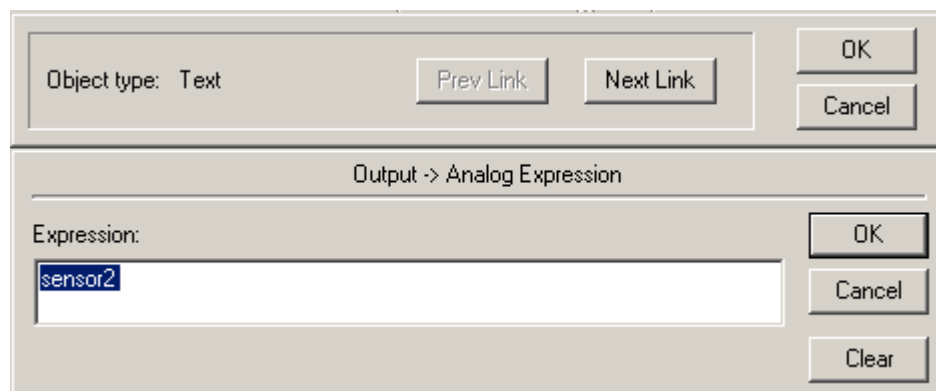
Figura 124. Configuración de una animación de llenado



De acuerdo con la configuración, el valor máximo de llenado es 80, correspondiente a la altura real del tanque, su valor mínimo de llenado es 0 que corresponde al tanque cuando esta vacío. A medida que el *software InTouch* recibe estos valores a través del PLC y del *software InControl*, su porcentaje de llenado aumenta (hasta un 100%) a medida que su valor aumenta (hasta 80) y si el valor disminuye su porcentaje disminuye en igual proporción.

Para que el valor numérico del tanque sea visualizado, se inserta un texto y se vincula al *tagname* sensor1 haciendo doble clic sobre el texto, seleccionando la animación **Value Display / Analog** desde el *animation Links* y configurándolo como se muestra en la figura 125.

Figura 125. Vincular un texto a un tagname.



El siguiente cuadro muestra los *tagnames* creados dentro de la ventana “Banco de Nivel”, su tipo de gráfico y la animación que se vinculó a ésta.

Cuadro 12. Lista de símbolos y tagnames de la ventana “Banco de Nivel”







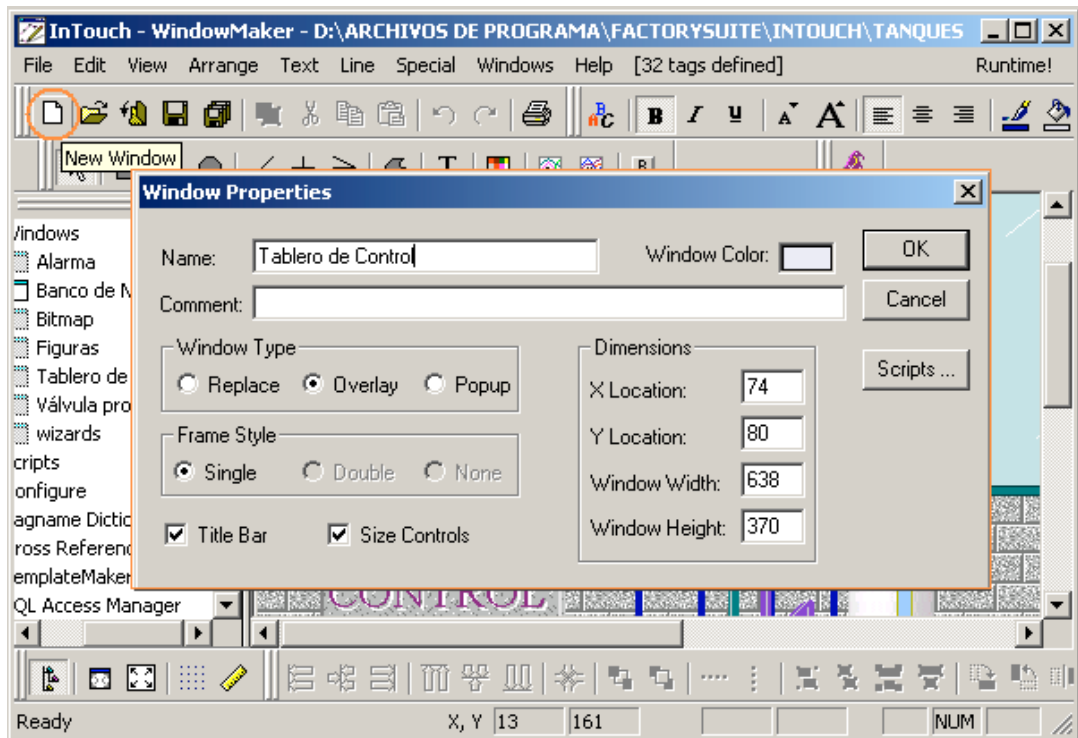
Figura	Tagname	Animation Links	Wizards	Figuras vinculadas
	Encendido		Discrete Witch	Luces, Application Script
###	sensor1 sensor2	Value Display Analog, Miscellaneous / Blink		Barra de llenado vertical
	BombaHidra	Touch Pushbotton/ Action, Miscellaneous / Blink		Encendido y texto
###	Error	Value Display Analog, Miscellaneous / Blink		Application Script

Figura	Tagname	Animation Links	Wizards	Figuras vinculadas
	nuevo	Porcent Fill / Vertical		Aplication Script
###	SetPoint	Value Display Analog,		
	TransNivel	Miscellaneous/ Visibility		
	VaciarT1 Vaciar T2		Detent Button	Aplication Script
	Válvula_3	Miscellaneous/ Visibility Touch Pushbotton/ Action		Aplication Script, texto
	Válvula3	Miscellaneous/ Visibility Touch Pushbotton/ Action		Aplication Script, texto

3.2.3.1.4 Creación y configuración de la ventana Tablero de Control. Luego de terminar el diseño de la animación del banco se da inicio a la creación de una ventana que permita controlar el proceso, seleccionar el tipo de control y visualizar las señales de entrada y salida.

- Se pulsa el botón **New Window** para configurar la nueva ventana como lo muestra la figura 126.

Figura 126. Creación de la ventana tablero de control



El nombre de la ventana (**name**) se define como “Tablero de Control”, el tipo de ventana seleccionado es **overlay**, es decir, la ventana puede ocultarse si se pulsa fuera de ésta. El título de la ventana estará activa durante la ejecución si se pulsa la pantalla de la figura del computador (**Title Bar**) y su tamaño podrá ser modificado (**Size Controls**). Su área (**Window Width - height**) ocupa 638x370 pixeles.

3.2.3.1.5 Creación de los gráficos de la ventana Tablero de Control.

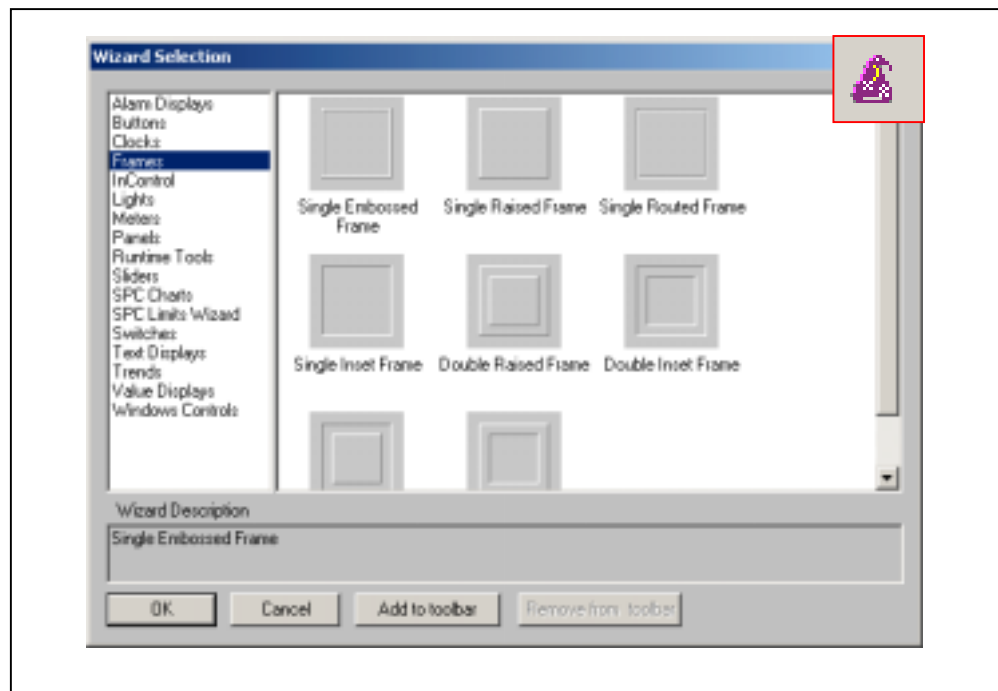
- Primero se trazan las áreas de trabajo con sus respectivos bordes (Figura 127), separando los gráficos históricos de los botones que controlan el proceso. Para esto se hace uso de la herramienta wizard llamada **Frames**.

Figura 127. trazo de bordes del la ventana tablero de control



- La elección de los frames se realizarán desde el cuadro de diálogo *Wizard Selection* (Figura 128), pulsando **OK** y ubicando el mouse sobre el lugar de la ventana de trabajo.

Figura 128. Insertar Frames



- Se copia el *wizard* de encendido (Figura 129) ubicado en la ventana principal para pegarlo sobre la nueva ventana. Los botones son insertados desde la ventana *Wizard Selection*. (Figura 130).

Figura 129. Objetos de control creados

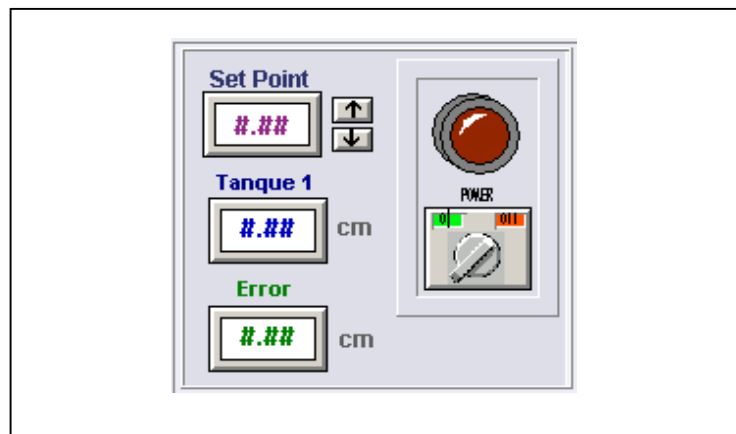
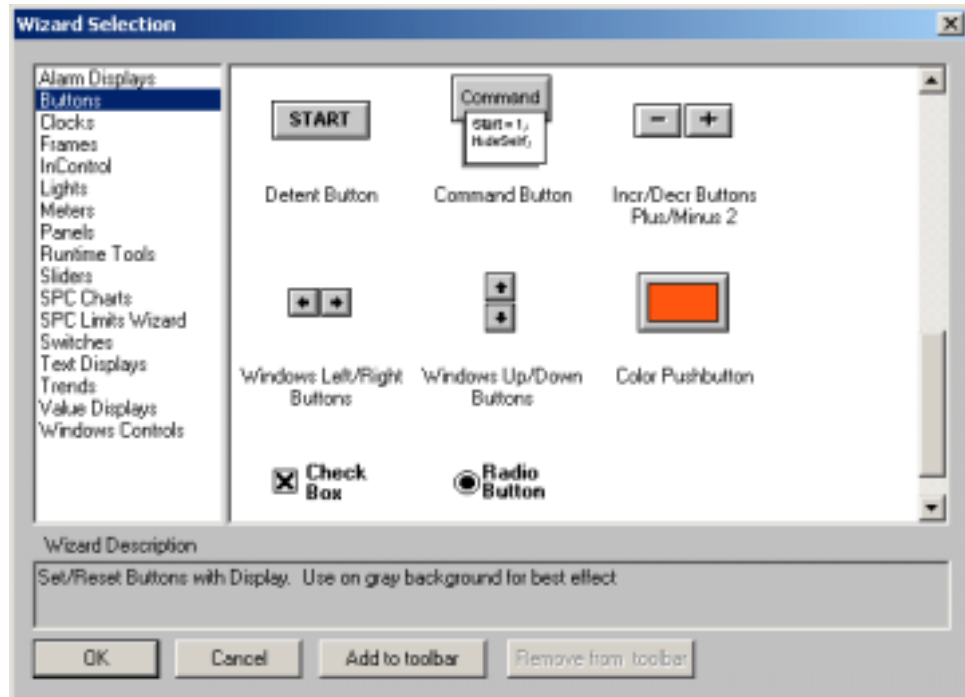


Figura 130. Elección de botones de control



- Las gráficas de tendencias son insertadas desde la barra de herramientas *Drawing* (Figura 131), pulsando el botón **Real-Time Trend** y arrastrando el *mouse* sobre la pantalla, escogiendo al mismo tiempo el tamaño de la gráfica (Figura 132).

Figura 131. Iconos para gráficas de tendencias

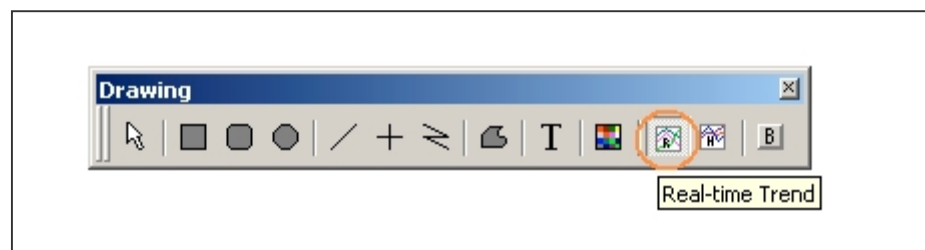
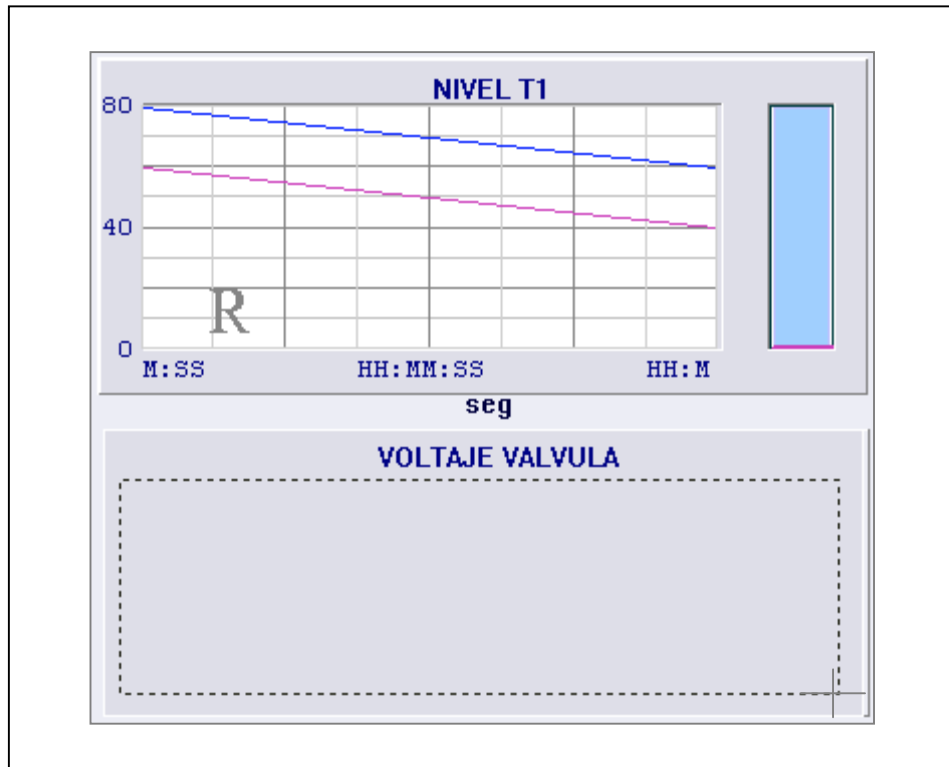
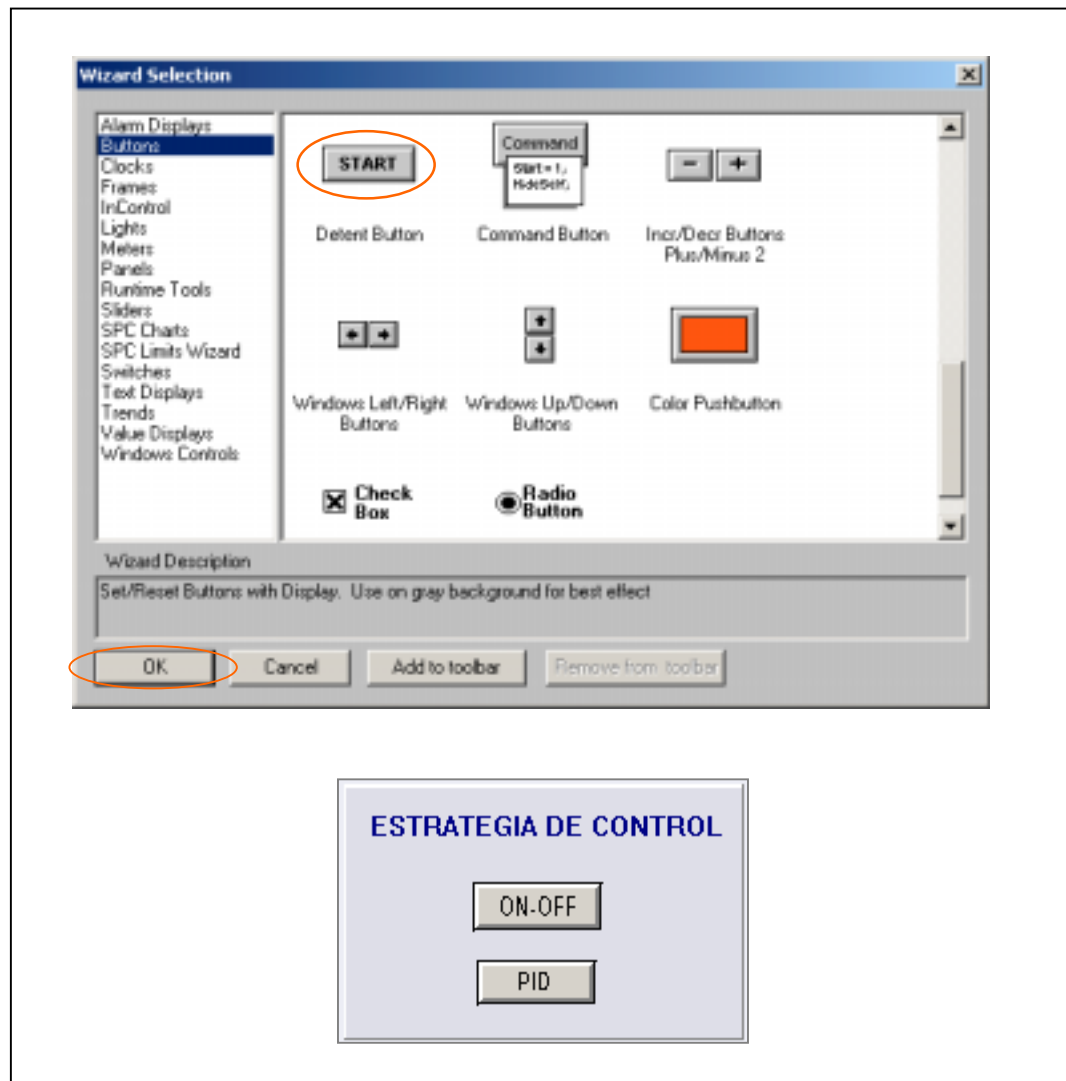


Figura 132. Insertar gráfica de tendencias



- Los botones para la selección del tipo de control son insertados desde el *wizard* llamado **Buttons** (Figura 133). Luego de pulsar sobre el icono insertado se modifica el nombre (ON-OFF o PID).

Figura 133. Elección de los botones de control



3.2.3.1.6 Configuración de tagnames en la ventana Tablero de Control. La configuración de la gráfica de tendencia en tiempo real del nivel del tanque 1 y la gráfica de la señal generada por el controlador que recibe la válvula se muestran en las figuras de la siguiente página. Se hace doble clic sobre cada gráfica de tendencia y se muestra los cuadros de diálogo como en la siguiente página

Figura 134. Configuración de las gráficas de tendencias

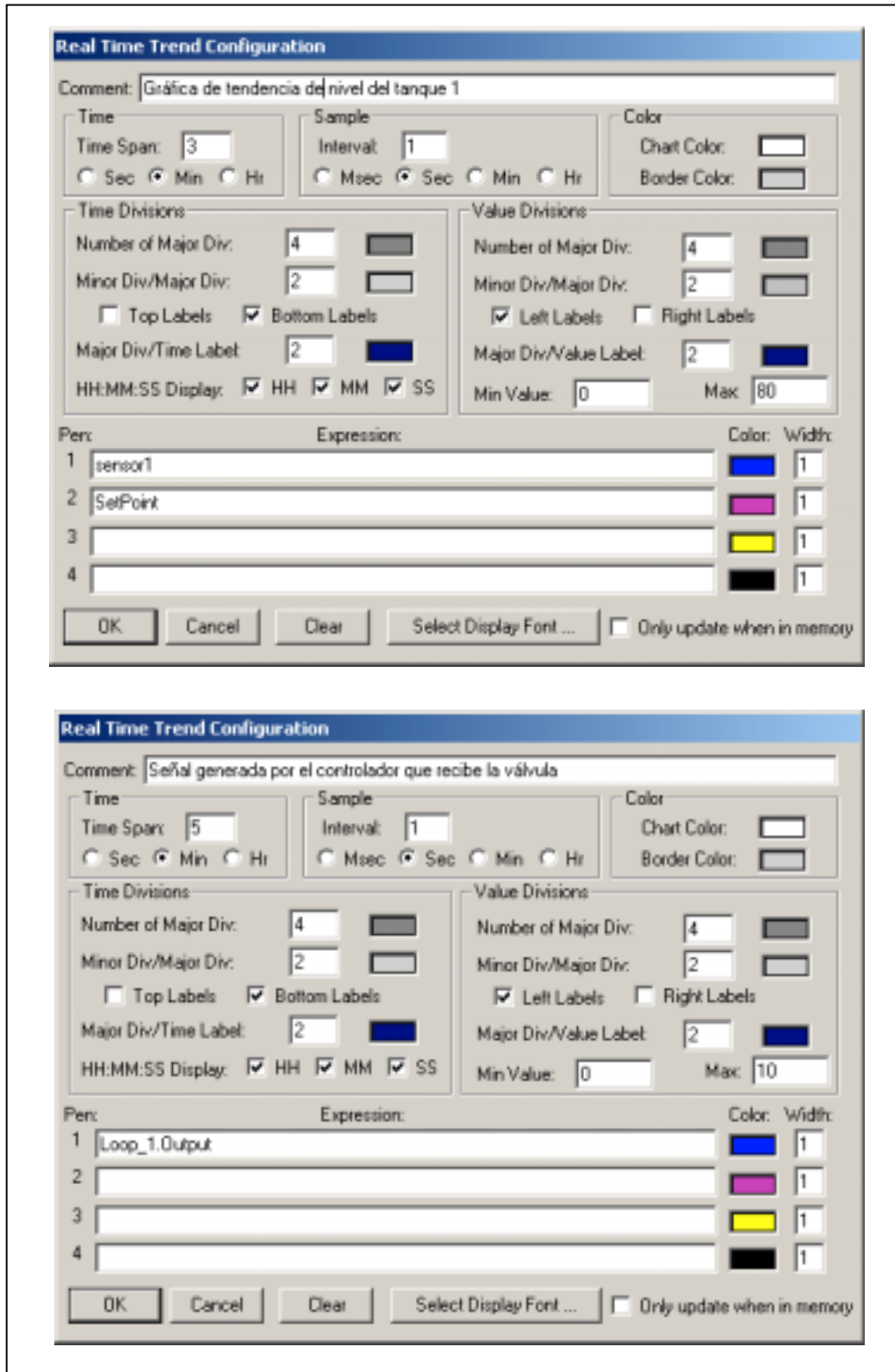
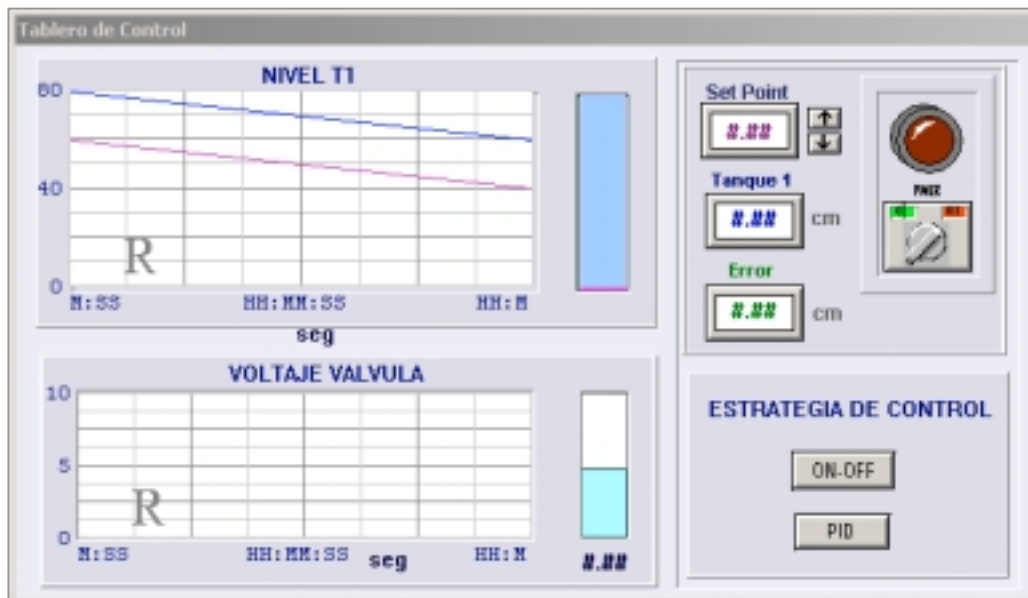


Figura 135. Presentación final de la pantalla tablero de control



El siguiente cuadro muestra los *tagnames* creados dentro de la ventana “Tablero de Control”, su tipo de gráfico, y la animación que se vinculó a ésta.

Cuadro13. Lista de símbolos y tagmanes de la ventana banco de nivel










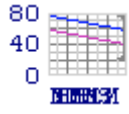
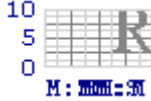
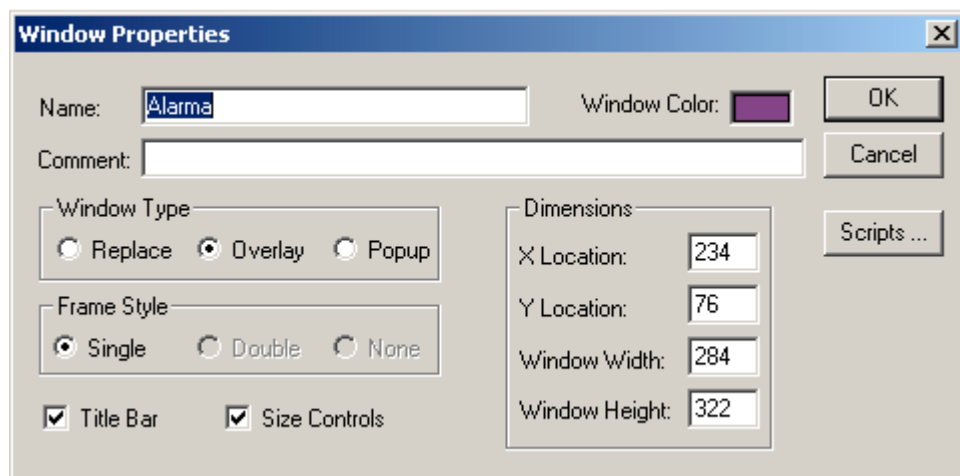
Figura	Tagname vinculado	Animation Links	Wizards	Figuras vinculadas
	Encendido		Discrete Witch	Luces, Aplication Script
	Encendido		Light wizard	Aplication Script

Figura	Tagname vinculado	Animation Links	Wizards	Figuras vinculadas
	SetPoint	Touch Links / Analog,		Aplication Script, Real-Time.
	sensor1	Miscellaneous / Blink Value Display / Analog		Barra de llenado t Ap.Script, gráfica Real-time.
	Error	Value Display / Analog		Application Script.
		Detent button wizard		
		Detent button wizard		
	sensor1	Porcent Fill / Vertical		Application Script, Real-Time, valor #.
	Voltaje	Porcent Fill / Vertical		Real-time, valor#. Script.
	sensor1 SetPoint	Real time trend configuration		Barra de llenado, script, valor#.
	Voltaje	Real time trend configuration		Barra de llenado, script, valor#.

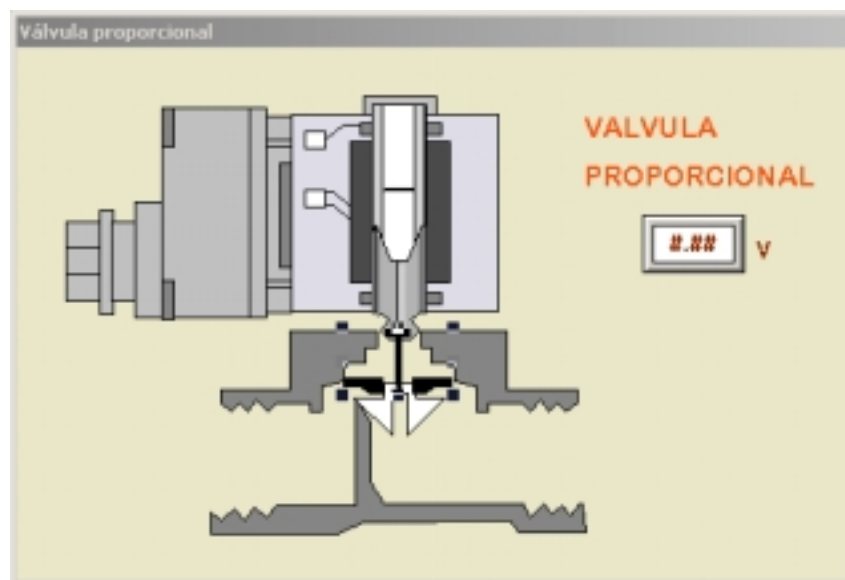
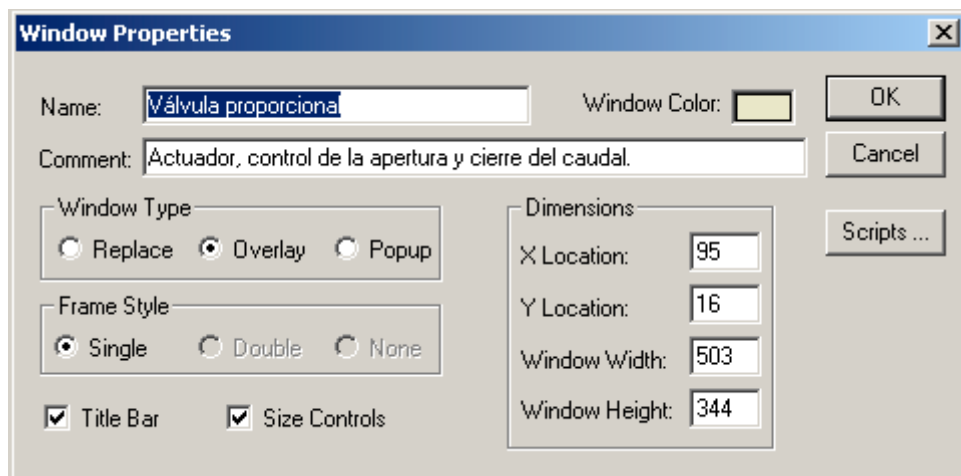
3.2.3.1.7 Creación y configuración de la ventana Alarma. La siguiente figura muestra la configuración de la ventana de alarma la cual aparece cuando el nivel de alguno de los tanques de llenado está cerca del valor de 80 cm.

Figura 136. Configuración de la ventana Alarma



3.2.3.1.8 Creación y configuración de la ventana Válvula proporcional. La siguiente figura muestra la configuración de la ventana Válvula proporcional desde donde se puede observar el comportamiento interno de la válvula.

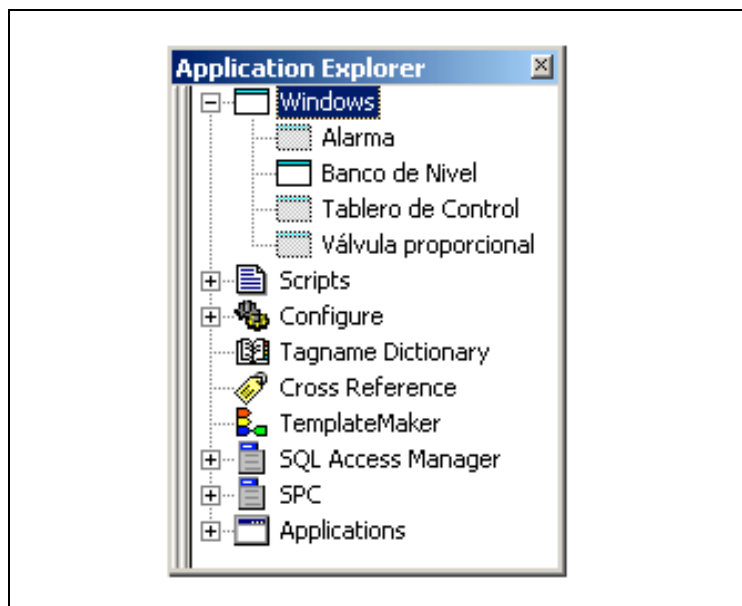
Figura 137. Configuración de la ventana válvula proporcional



3.2.3.1.9 Creación de Scripts. Hasta aquí los objetos tienen animaciones particulares, pero se necesita relacionar entre sí las animaciones de los *tagnames* existentes, por ejemplo, que el *tagname* que controla la luz de encendido del sistema controlara al mismo tiempo los *tagnames* de llenado de los tanques, la elección del tipo de sistema (primer o segundo orden), y que fuese controlado por un valor de máximo de nivel. Esto puede hacerse escribiendo una expresión lógica dentro de los cuadros de configuración de los *Links* y diseñando un **Scripts**.

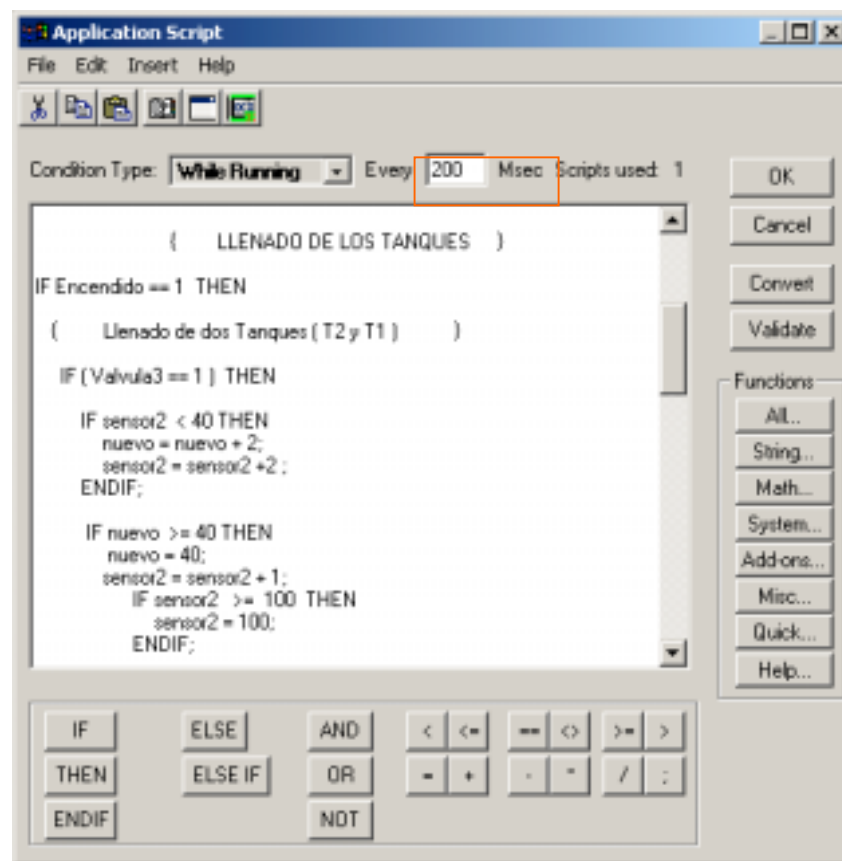
Los scripts que se desarrollan están basados en los eventos de ejecución de la aplicación Control de Nivel (*Application*) y del pulso de la válvula manual de tres vías (*Key*). Estos eventos se determinan eligiendo el *script* correspondiente desde la ventana *Application Explorer* (figura 138).

Figura 138. Ingreso a los scripts



Nota: Debido a que el sistema no es realmente implementado se diseñó un *script*, del tipo *Aplicación*, que simulara el comportamiento del proceso. El script es programado para que se ejecute cada 200 Msec mientras la aplicación esté corriendo, y aunque no es el tiempo real de llenado de los tanques es un valor aproximado donde se puede comprobar la estrategia de control (ON-OFF y PID).

Figura 139. Ventana de un Script de aplicación



{ACTIVACIÓN DE LA ALARMA, si alguno de los tanques se acerca a su desborde, aparece una ventana de advertencia}

```

IF sensor2>=78 OR sensor1>=78 THEN

    Show "Alarma";

ENDIF;

{Elige automáticamente el sistema de primer orden como estado inicial}

IF ( Valvula3 == 0 ) AND ( Valvula_3 == 0 ) THEN

    Valvula3 = 1;

ENDIF;

{ LLENADO DE LOS TANQUES, Llenado de dos Tanques ( T2 y T1 ) si el
sistema es de segundo orden, se empezara por el llenado de la rotura del tanque
2 (nuevo) }
IF Encendido == 1 THEN

    IF ( Valvula3 == 1 ) THEN

        IF sensor2 < 40 THEN

            nuevo = nuevo + 2;

            sensor2 = sensor2 +2 ;

        ENDIF;

        IF nuevo >= 40 THEN

            nuevo = 40;

            sensor2 = sensor2 + 1;

            IF sensor2 >= 100 THEN

                sensor2 = 100;

            ENDIF;

        ENDIF;

        sensor1 = sensor1 + 1;

        IF sensor1 >= 80 THEN

```

```

        sensor1 = 80 ;
        Encendido = 0;
ENDIF;

    ENDIF;

ENDIF;

{    Llenado de un Tanque T1    para el sistema de primer orden    }

IF Valvula_3 ==1 THEN

    sensor1 = sensor1 + 2;

        IF sensor1 >= 80 THEN

            sensor1 = 80 ;

            Encendido = 0;

ENDIF;

    ENDIF;

ENDIF;

{    Vaciar el Tanque2 al pulsar "vaciar"    }

IF vaciarT2 == 1 THEN

    sensor2 = sensor2 - 1;

        IF sensor2 <= 40 THEN

            nuevo = nuevo - 1;

                IF nuevo <= 0 THEN

                    nuevo = 0 ;

                ENDIF;

                IF sensor2 <= 0 THEN

```



```

        sensor2 = 0;
    ENDIF;
ENDIF;
ENDIF;

{ Vaciar el Tanque1 al pulsar "vaciar" }
IF vaciarT1 == 1 THEN
    sensor1 = sensor1 - 1;
    IF sensor1 <= 0 THEN
        sensor1 = 0;
    ENDIF;
ENDIF;

{ Error de Control}
Error = Abs(SetPoint - sensor1);

```

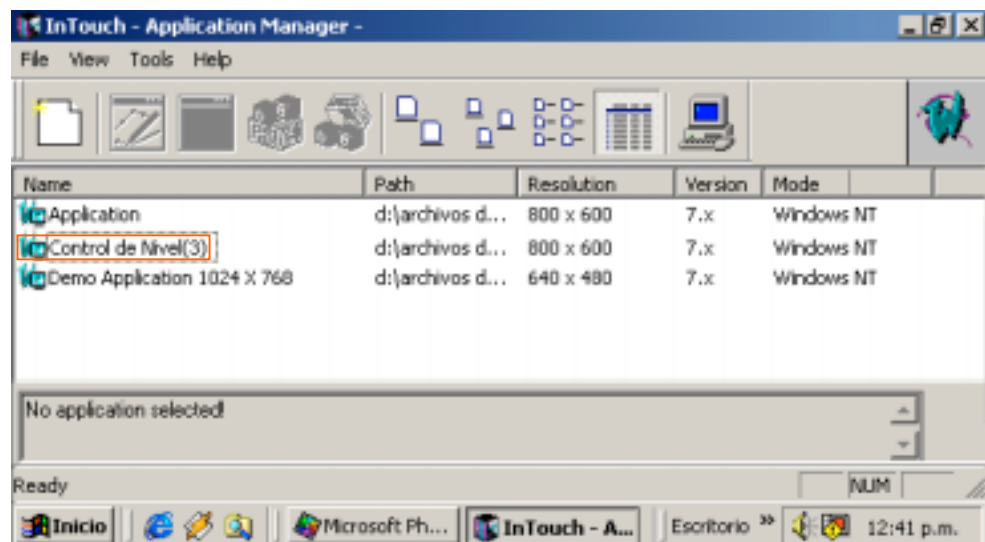
El anterior *script* permite elegir el tipo de sistema para que simule el llenado de uno o los dos tanques mientras el banco este encendido, además deja ver el vaciado de los tanques mientras el banco se encuentre apagado. El valor del nivel de los tanques esta determinado por las expresiones de incremento o decremento que incluye los *tagnames* sensor 1 y sensor2, entre ellos, $sensor1 = sensor1+2$, $sensor1 = sensor1-1$ y $sensor 2 = sensor2 +1$. El tiempo de llenado lo determina la frecuencia en el cual el script se ejecuta (200 Msc).

Después que el sistema de control que se plantea se implemente, el *script* de aplicación se reduce a:

```
IF sensor1>=80 THEN  
    Show "Alarma";  
ENDIF;  
{ Error de Control}  
Error = Abs(SetPoint - sensor1);
```

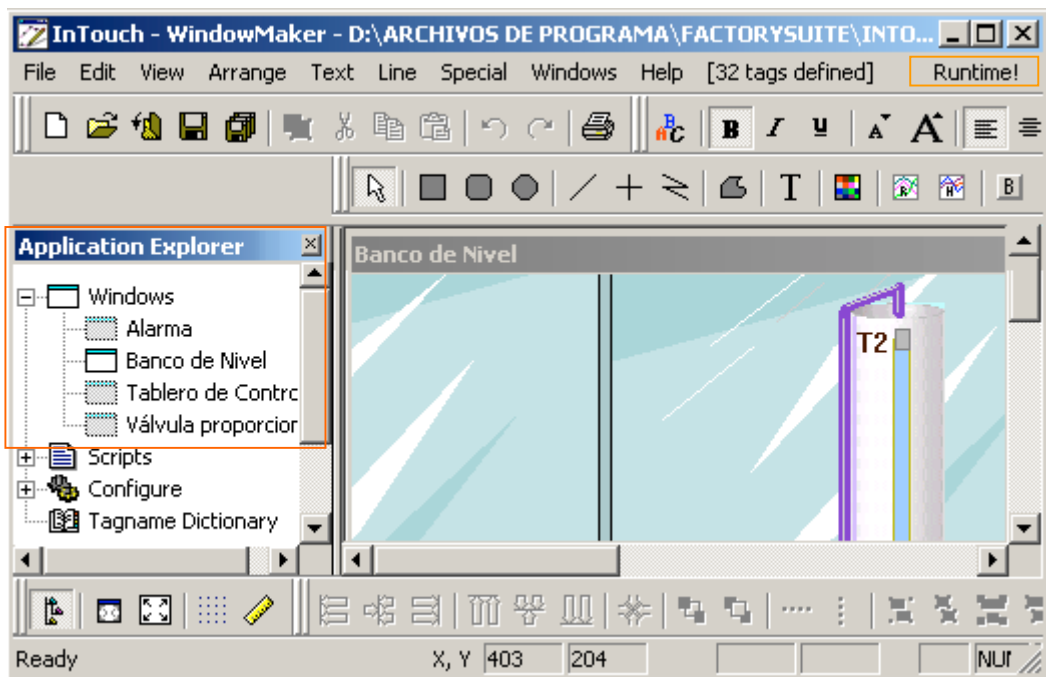
3.2.3.2 Descripción del manejo del mímico. La aplicación del sistema lleva el nombre “Control de Nivel”, como lo muestra la ventana *Application Manager* (Figura 140) que aparece cuando se ejecuta el programa *InTouch*.

Figura 140. Ventana Application Manager



Esta aplicación está conformada por cuatro ventanas según lo confirma el administrador *Application Explorer* (Figura 141), éstas son: Banco de Nivel, Válvula proporcional, Alarma y Tablero de Control, cada una de ellas con una función especial dentro de la aplicación. Para su ejecución se pulsa el botón *runtime* dentro de *WindowMaker*.

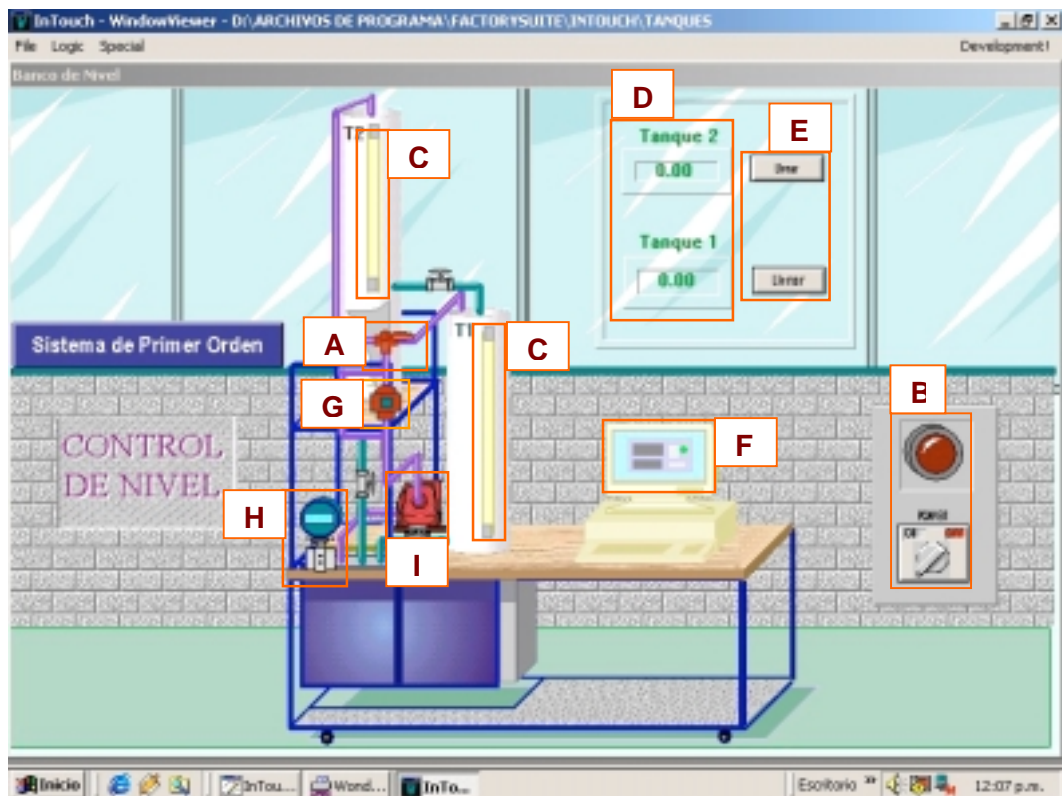
Figura 141. Ventana Window Maker



Esta es la ventana principal de la aplicación llamada **Banco de Nivel** (Figura 142) donde se ubican cada uno de los elementos que conforman físicamente el banco, desde aquí se vigila el comportamiento del llenado de los tanques y se ven los

valores sentidos, se monitorea el sistema, y se abren las ventanas restantes. Cuando se ejecuta el proyecto ésta es la ventana que se encuentra activa.

Figura 142. Ventana Banco de Nivel



Cada una de los recuadros de la figura anterior indica una animación o tarea específica que se puede realizar dentro de esta ventana.

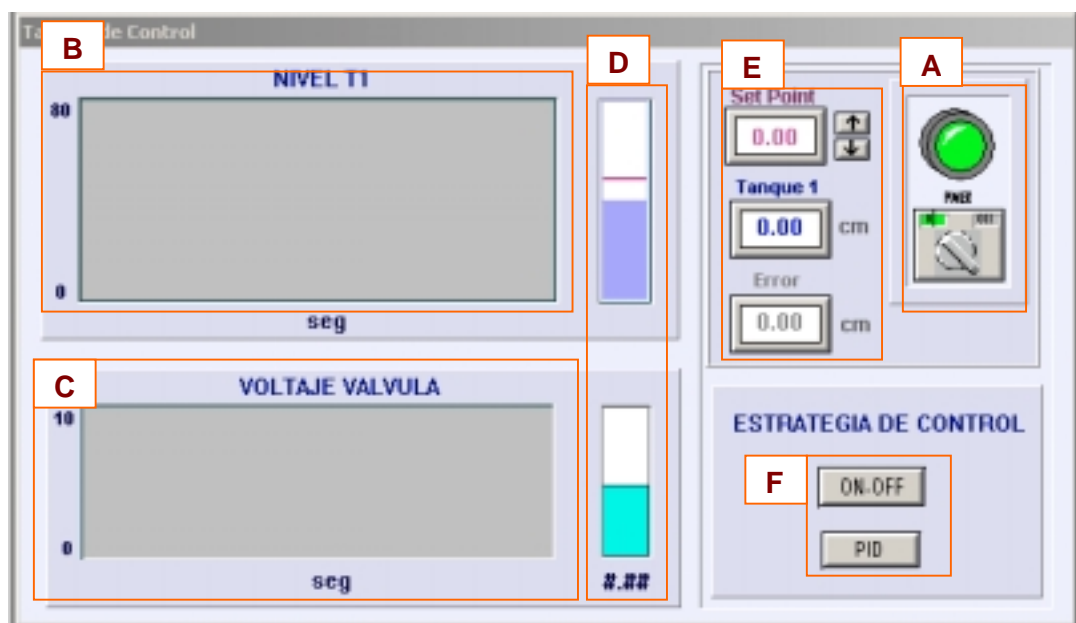
Cuadro 14. Cuadro de tareas de la ventana Banco de Nivel

Cuadro	Animaciones y Tareas
A	Pulse sobre esta gráfica para elegir la configuración del sistema, es decir, si es de primer orden (un solo tanque) o segundo orden (tanques en cascada).
B	Enciende o apaga el sistema pulsando sobre la gráfica <i>POWER</i> , así mismo se puede dar inicio o detención al llenado de los tanques.
C	Puede ver la animación de llenado de los tanques para vigilar el estado del sistema.
D	Muestra el valor en centímetros del nivel de agua de los tanques.
E	Al pulsar sobre estos botones puede modificar el estado de vaciado o llenado de los tanques, mientras el sistema esta encendido.
F	Abre la ventana “Tablero de Control” pulsando sobre la pantalla del computador.
G	Abre la ventana “Válvula proporcional”.
H	Identifica al elemento al pulsarlo como “Transmisor de nivel”.
I	Muestra el estado actual de la bomba hidráulica y la identifica al pulsar sobre ella.

Como su nombre lo indica la ventana **Tablero de Control** se encarga de configurar el sistema de control, a ésta se accede haciendo clic sobre la pantalla del computador que se encuentra en la ventana principal “Banco de Nivel”. Desde

aquí se muestran las gráficas de tendencia de las señales que recibiría del *software InControl* (sensor1, sensor2. Encendido) y las generadas por éste (Error, SetPoint). También se puede seleccionar la estrategia de control a utilizar y fijar el valor del set-point.

Figura 143. Ventana Tablero de Control



Cuadro 15. Cuadro de tareas de la ventana Tablero de Control.

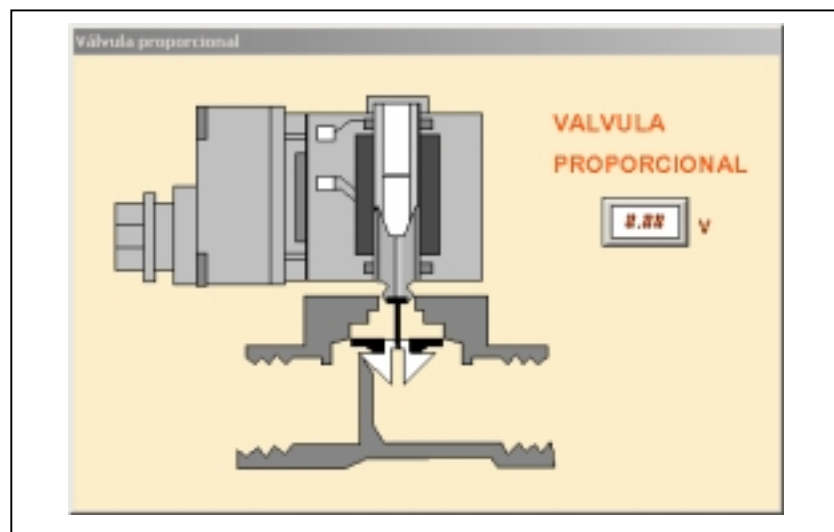
I

Cuadro	Animaciones y Tareas
A	Para encender o apagar el sistema pulsando sobre la gráfica <i>POWER</i> , así mismo se puede dar inicio o detención al llenado de los tanques.
B	Gráfica de tendencia del nivel del tanque sensado y el valor del set-point elegido.

Cuadro	Animaciones y Tareas
C	Gráfica de tendencia de la señal de voltaje que genera el controlador para modificar la apertura de la válvula proporcional.
D	Representación en diagrama de barra de la altura del nivel y el voltaje de la válvula con su indicación digital. Además del valor del setpoint representado por una línea horizontal.
E	Elección del valor del set-point escribiendo sobre la pantalla o pulsando sobre los iconos. En la parte inferior se muestran los valores del nivel del tanque a controlar y el error a corregir.
F	Elección de la estrategia de control.

La ventana **Válvula proporcional** (Figura 144) muestra la animación del comportamiento de la válvula bajo la señal de voltaje enviada por el controlador.

Figura 144. Ventana Válvula proporcional.



La siguiente ventana, **Alarma** (Figura 145), aparece si el nivel del tanque a controlar se encuentra cerca del desbordamiento al mismo tiempo que se apaga el sistema.

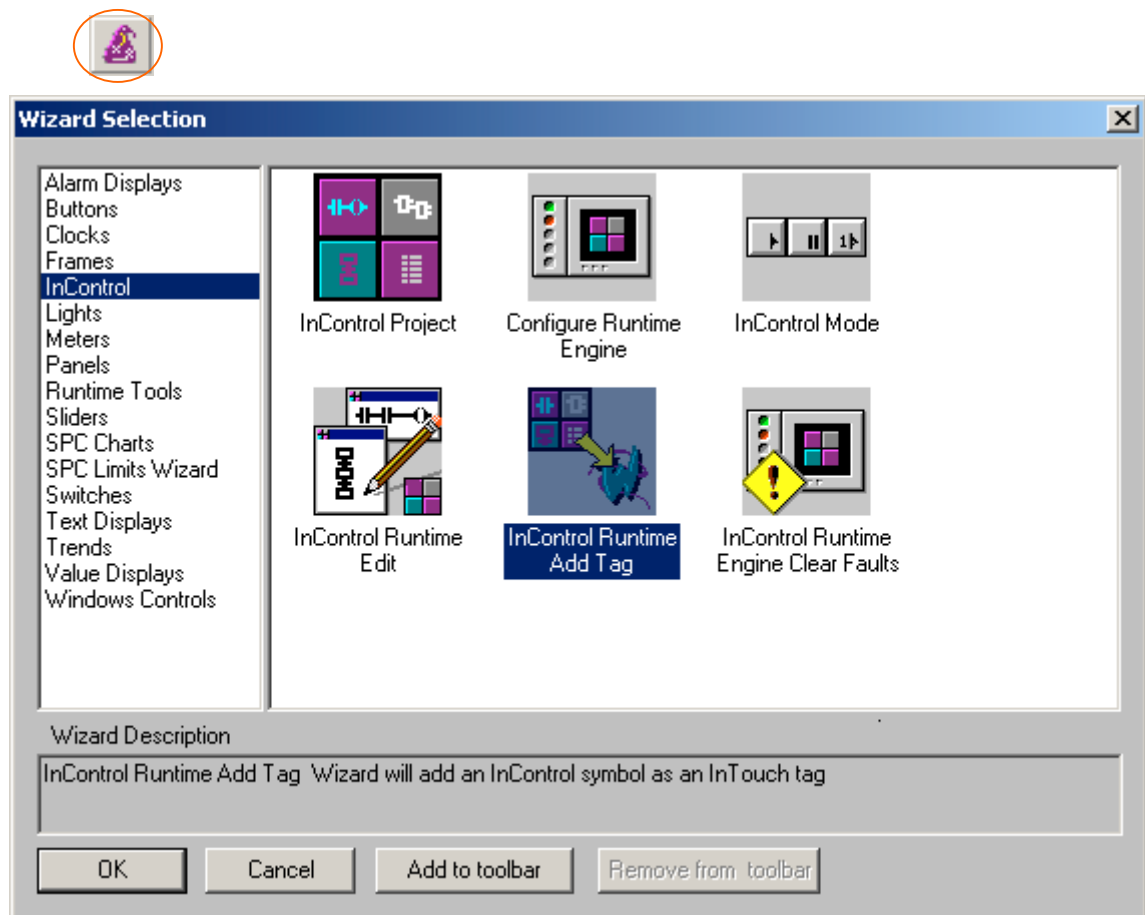
Figura 145. Ventana Alarma



3.2.3.3 Creación de tagnames desde InTouch para leer símbolos InControl.

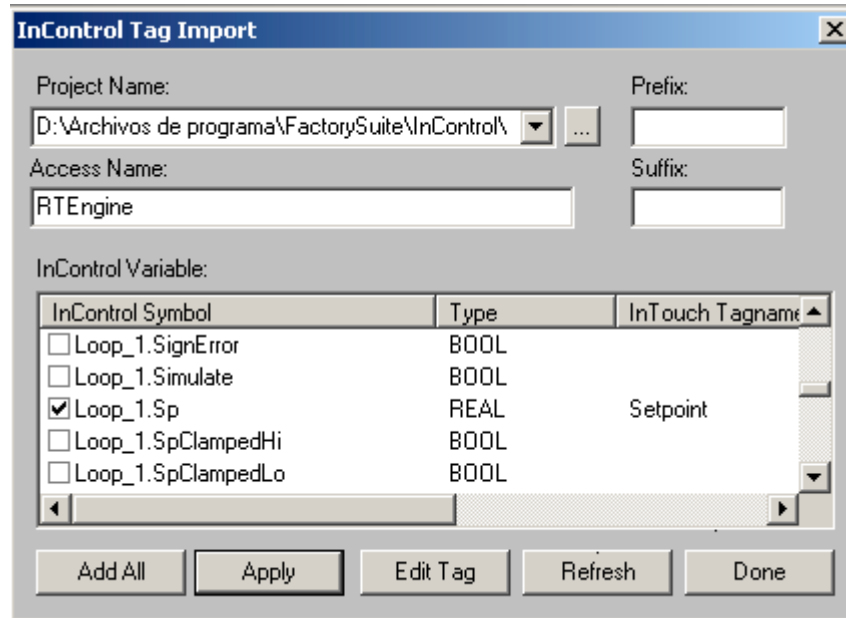
Abra la aplicación "Control de Nivel" y pulse sobre el icono de wizards. Luego de aparecer la ventana *Wizard Selection*, seleccione el tipo de wizard *InControl*, escoja el icono *InControl Runtime Add Tag* y pulse **OK**.

Figura 146. Elección del wizard InControl Add Tag



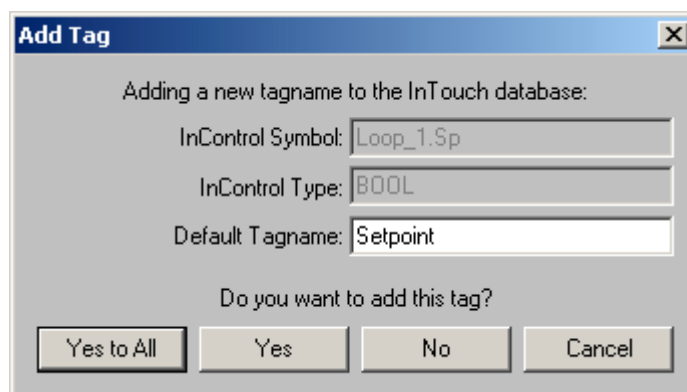
El cuadro de diálogo *InControl Tag Import* muestra los símbolos InControl de la aplicación seleccionada en el campo *Project Name* . Se selecciona la dirección de la aplicación y en el campo *Access Name* se deja en *RTEngine* porque los símbolos InControl se encuentran en el mismo computador.

Figura 147. Elección de las etiquetas a vincular



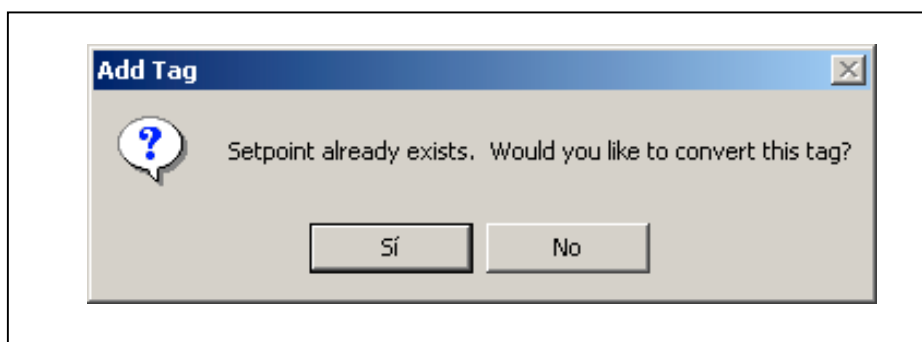
Seleccionando el símbolo Loop_1.Sp y pulsando *Apply* aparece el siguiente cuadro de diálogo Add Tag. Aquí se escribe el nombre del símbolo InTouch y se pulsa **Yes**.

Figura 148. Vínculo de los tagnames con los vínculos de InControl



Como la etiqueta ya ha sido creada el programa pregunta si se desea convertir la etiqueta o Tag, lo cual se confirma pulsando **Si**.

Figura 149. Confirmación de conversión de tagnames



Los siguientes son los símbolos InControl vinculados con los *tagnames* de InTouch.

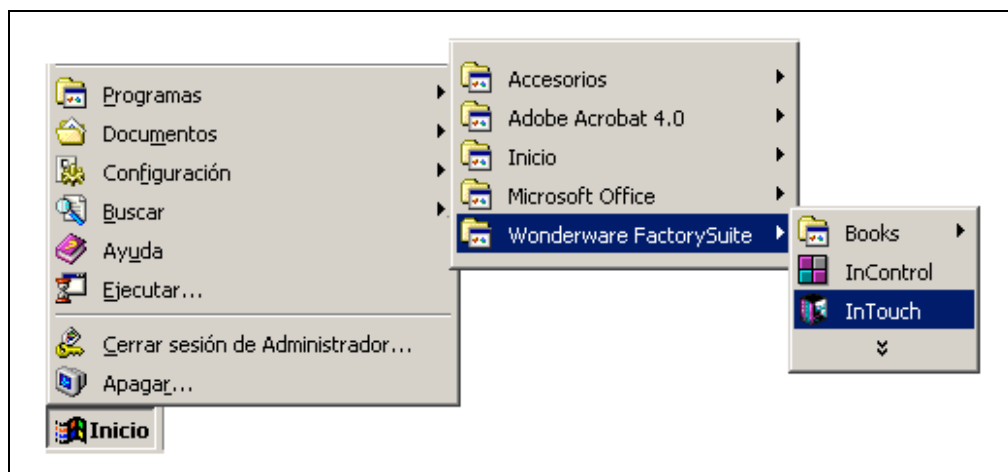
Cuadro 16. Tagnames de InTouch y símbolos InControl vinculados

Símbolo InControl	Tagname InTouch
Nivel.PID.Sp	Setpoint
NivelPID.control_valvula	Voltaje
Nivel PID.nivel_tanque	sensor1

3.2.3.4 Ejecución del supervisorío y de la estrategia de control. Los siguientes pasos describen cómo ingresar al software de supervisión para monitorear y controlar el sistema del banco Control de Nivel.

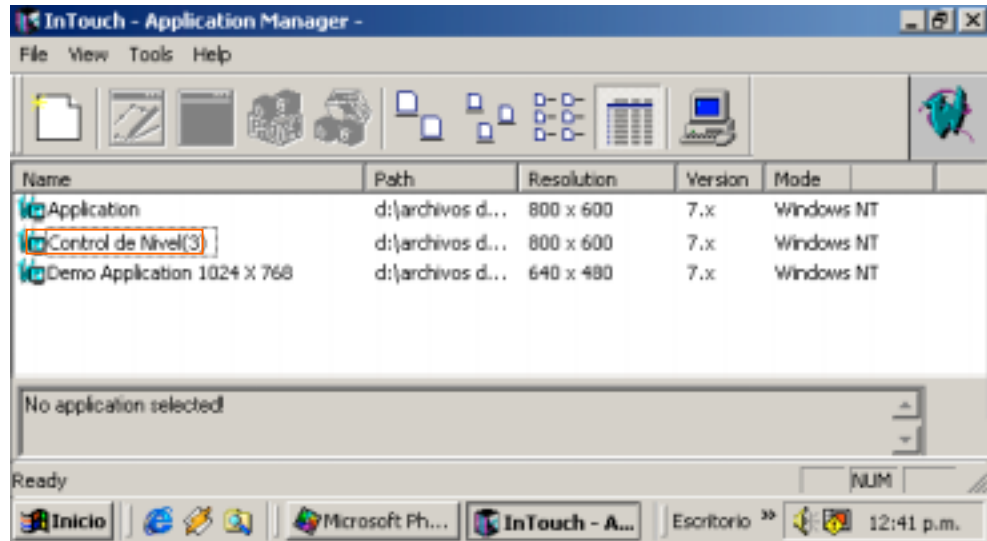
- Se pulsa el botón de inicio desde la barra de tareas de Windows y seleccione **inicio / WonderwareFactorySuite** (Figura 150) para ingresar al *Application Manager* de *InTouch* (Figura 151).

Figura 150. Ruta de ejecución InTouch



- Desde la ventana *Application Manager* se escoge el proyecto a ejecutar, en nuestro caso se ingresa a la aplicación denominada “**Control de Nivel**” haciendo doble clic sobre su nombre (Figura 151) para abrirlo en el ambiente de trabajo de proyectos o *Window Maker* (Figura 152).

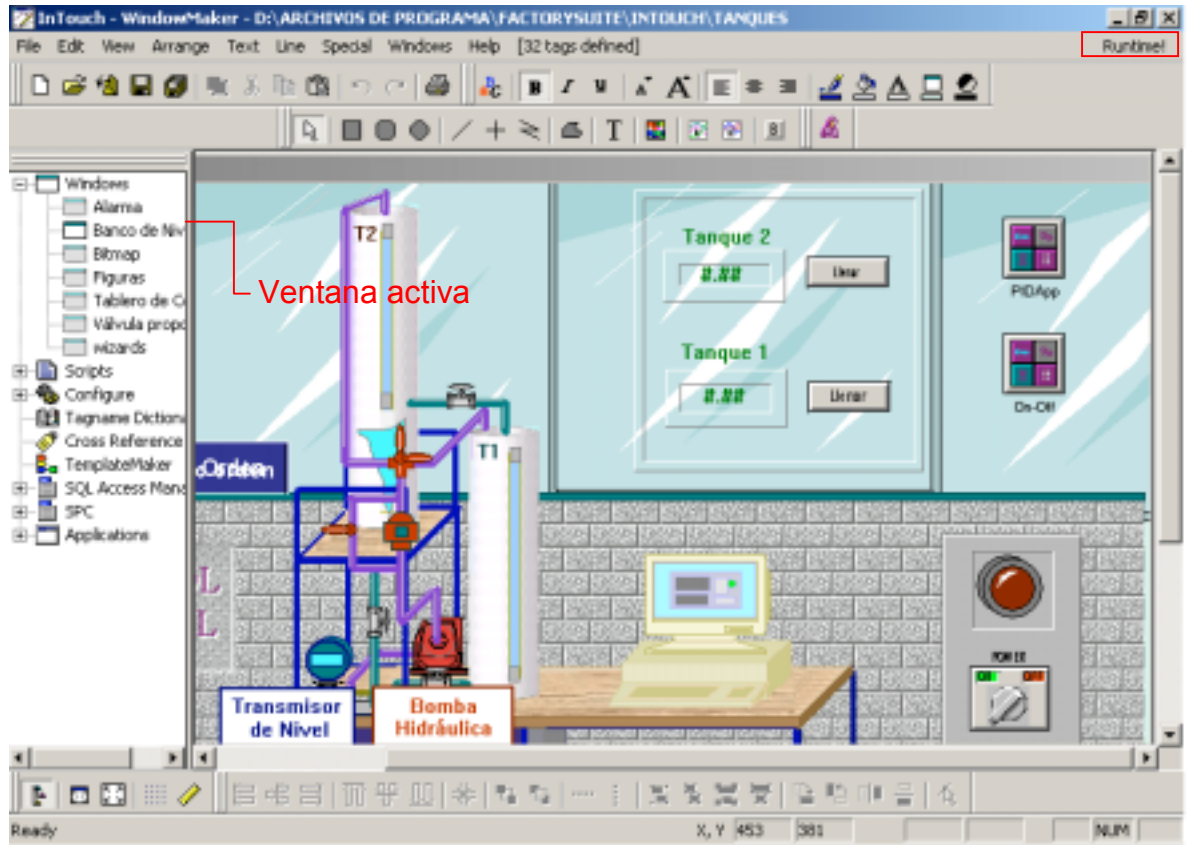
Figura 151. Ventana Application Manager



- Dentro del *WindowMaker* se pulsa en el icono *Runtime* ubicado en la parte superior derecha de la pantalla como se indica en la figura 152, esto para ingresar a la ventana de ejecución o *Window Viewer* (Figura 153) desde donde se ejecuta, monitorea y controla nuestro sistema de control.

Se debe tener en cuenta que las ventanas que se encuentren activas en la ventana *Application Manager* (Figura 152) serán ejecutadas al momento en que la *Window Viewer* se ejecute.

Figura 152. Ventana Window Maker de la aplicación



Estando dentro de la ventana de ejecución (Figura 153) se selecciona el tipo del sistema con el cual se desea trabajar, es decir si es de primer orden (un tanque) o de segundo orden (dos tanque en cascada) y se selecciona el tipo de control ON-OFF o PID con que se desea trabajar. Los siguientes pasos describen la forma correcta de ejecutar el control del sistema después de haber iniciado el software InTouch e InControl:

1. Encienda el sistema pulsando el botón de encendido ya sea desde la ventana **“Banco de Nivel”** o **“Tablero de Control”**.
2. Seleccione el tipo de sistema (De segundo orden o de primer orden) ubicando el cursor sobre la válvula de tres vías (Figura 153).
3. Haga doble clic sobre la pantalla del computador que se presenta en la ventana principal de nuestra aplicación (Figura 153) para abrir la ventana **“Tablero de Control”** (Figura 154) donde se selecciona el tipo de control pulsando sobre los iconos ON-OFF y PID.
4. Escoja el nivel del tanque deseado escribiendo sobre el campo SetPoint (Figura 157).

Existe la opción de acceder a los proyectos de control creados en InControl desde la ventana InTouch pulsando en los iconos PIDAPP y ON-OFF (Figura 153) o ingresando directamente desde InControl, y ejecutando la respectiva estrategia de control (ON-OFF ó PID) correspondiente a los proyectos ControlPID (Figura 154) y ControlON-OFF (Figura 155).

Figura 153. Ventana Window Viewer

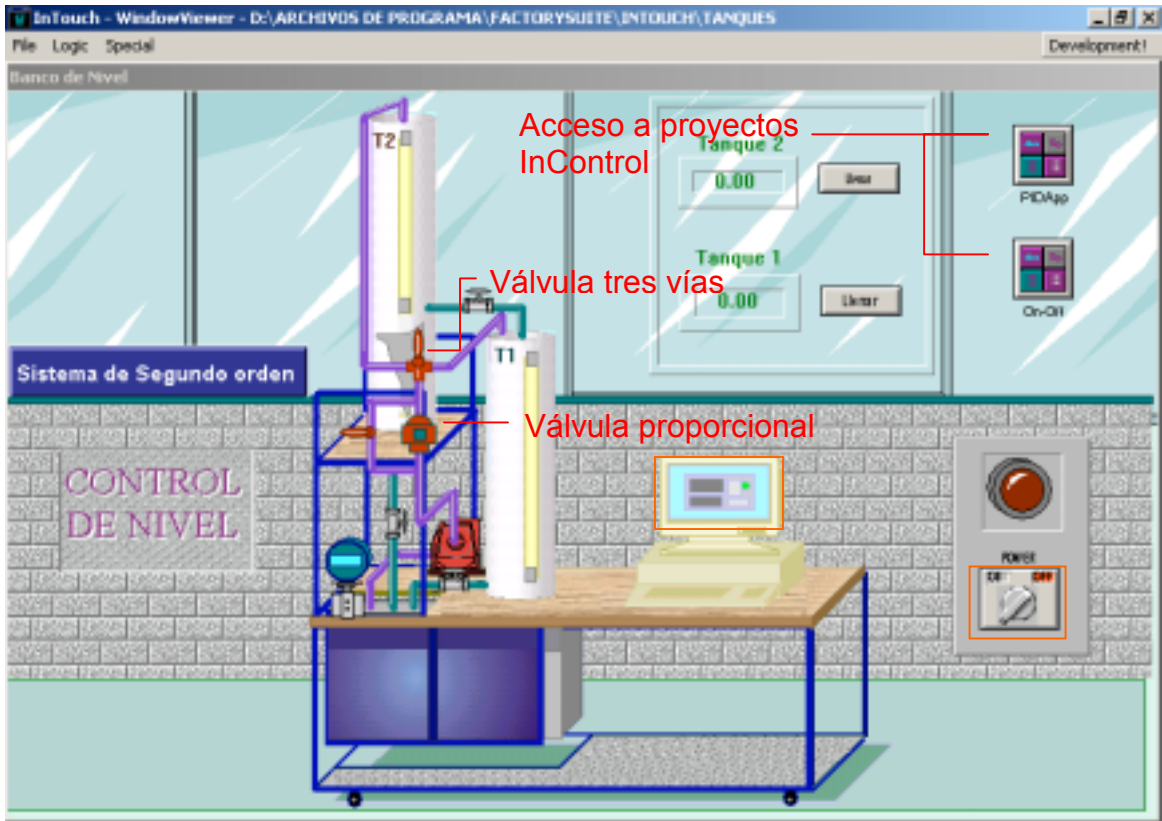


Figura 154. Ventana del proyecto ControlPID.

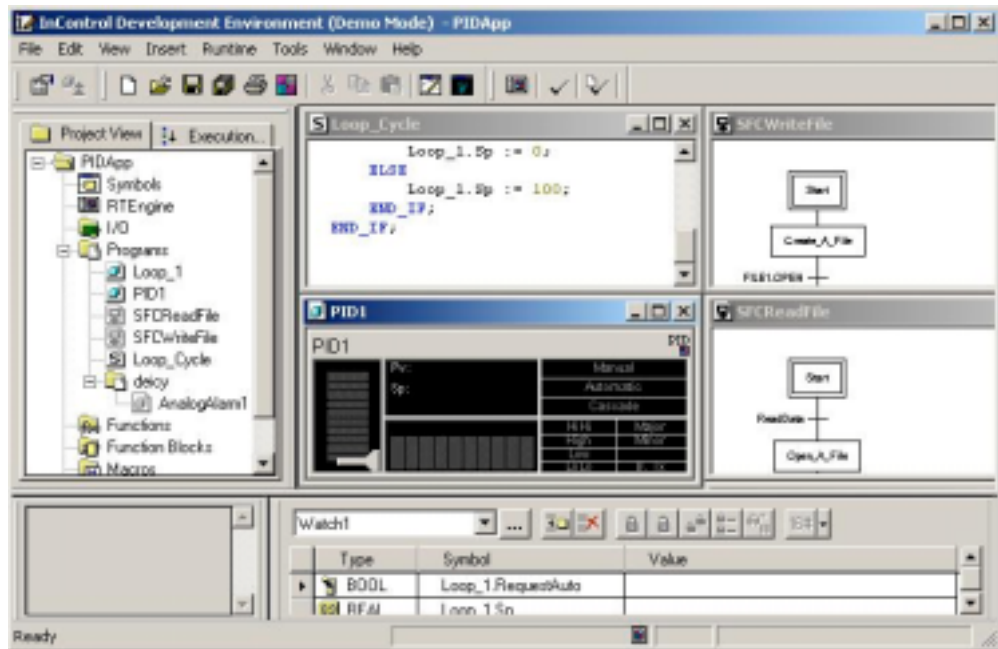
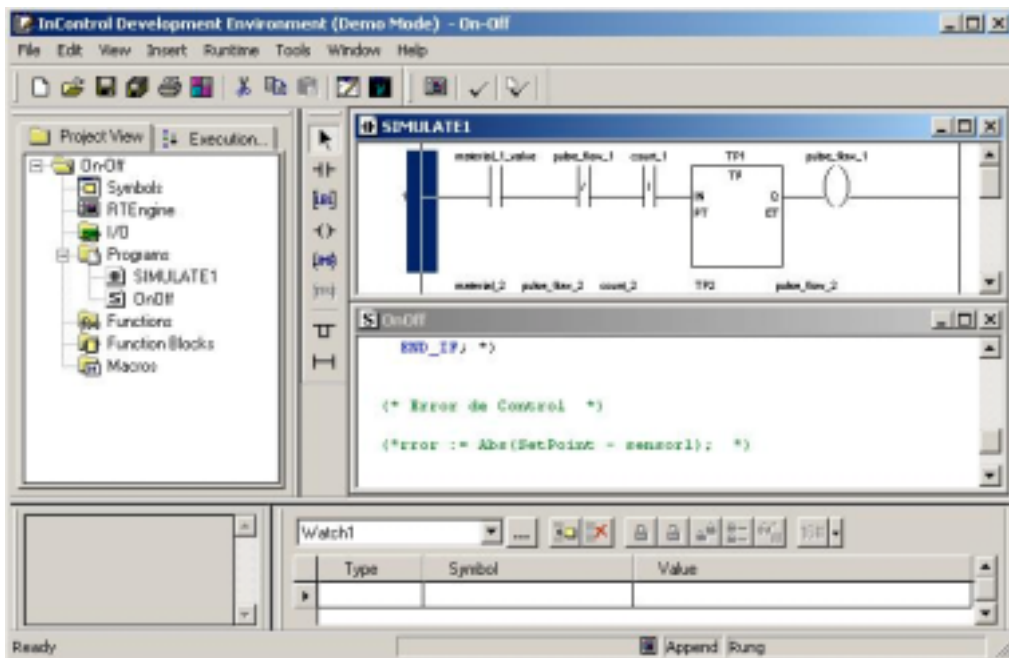
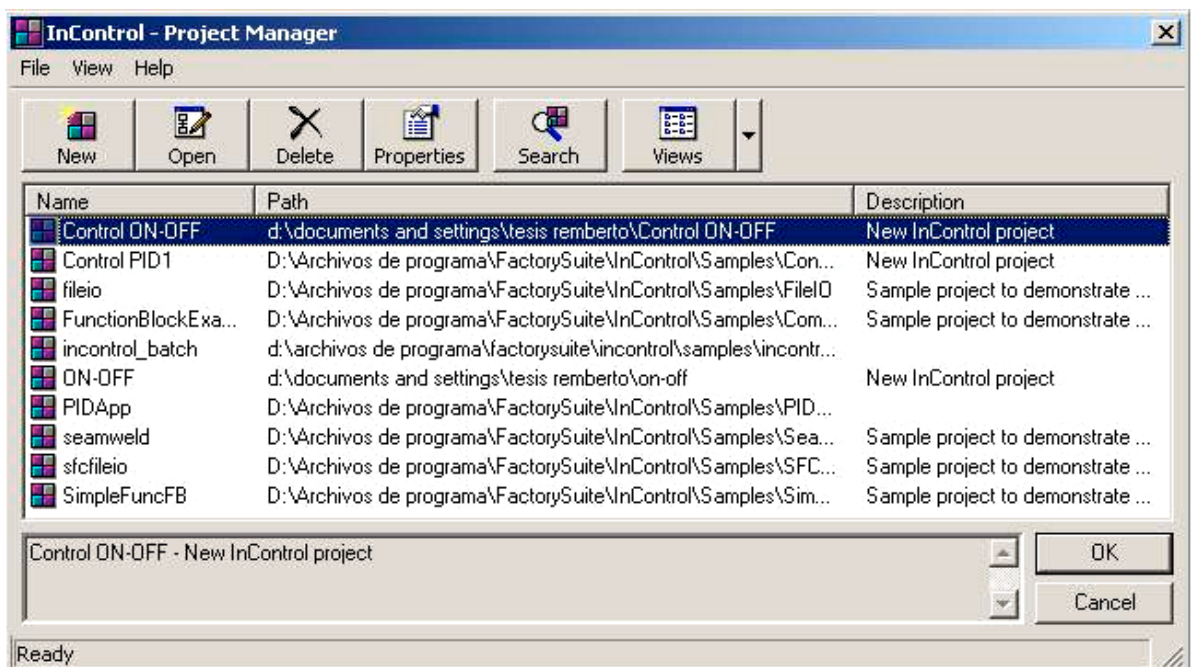


Figura 155. Ventana del proyecto Control ON-OFF.



Si se quiere ejecutar solamente la estrategia de control (InControl) sin necesidad del supervisorío (InTouch) usted puede ingresar directamente desde la barra de inicio y hacer clic en el icono InControl para ejecutar el proyecto de control (ON-OFF ó PID) desde el InControl Project Manager (Figura 156) o seguir los pasos de la figura 150.

Figura 156. InControl Project Manager.



Para observar el comportamiento de la válvula se puede hacer clic sobre ella en la figura 153 y ver los nombres de algunos elementos como el Transmisor de Nivel y la bomba hidráulica (Figura 153).

De esta manera el programa muestra las señales de tendencia del nivel del Tanque 1 y la señal del voltaje que genera el controlador (Figura 157), además de los valores numéricos de error y nivel del tanque a controlar.

Figura 157. Ventana Tablero de control activa



5. Pulse los botones **Vaciar** de la ventana principal si desea vaciar los tanques y apagar el sistema.
6. Pulse en el icono **Development** ubicado en la parte superior derecha de la pantalla del InTouch para regresar a su ambiente de trabajo (*Window Maker*)
7. Salga de la aplicación pulsando el comando **Exit** del menú **File**.

Con los anteriores pasos usted puede ejecutar todos los proyectos de aplicación que usted cree en InControl.

3.3 FUTURO PROYECTO: DISEÑO DE UNA RED PROFIBUS CON LOS BANCOS DEL LABORATORIO DE CONTROL DE LA TECNOLÓGICA DE BOLÍVAR.

Materiales necesarios para el diseño de la red Profibus:

- Tarjeta SMS CIF30-DPM Synergetic.
- Tarjeta SMS CIF30-DPS Synergetic (6 unidades).
- Archivos GSD del PLC S7-300 de Siemens.
- Cable RS-485 Par trenzado blindado de cobre con un par conductor.
- Cables de derivación (preferiblemente de 1m o menos (precher+Shuh))
- Conectores de junta en T (precher+Shuh).
- Cables de segmento de bus (Srecher+Shuh).
- Conector (precher+Shuh).
- PC con un procesador >800 MHz.
- Windows NT 3.51, Windows NT 4.0.
- RAM; mínima de 64MB.
- Resolución grafica: 800x600 píxel o más.

Por medio de una red Profibus, los dispositivos de diferentes fabricantes pueden comunicarse sin ajustarse a una interfase especial. El protocolo PROFIBUS distingue entre dispositivos maestros y esclavos. Los dispositivos maestros determinan la comunicación de los datos en el bus. Los dispositivos esclavos son

dispositivos periféricos, típicamente un esclavo incluye dispositivos I/O, válvulas y transmisores de medida o un PC propiamente.

3.3.1 Diseño y Descripción de la red PROFIBUS. La red Profibus DP con los bancos del laboratorio será configurada con el software SyCon System Configurator des Synergetic instalado en el computador que será nuestra estación central o maestra, por lo tanto se necesita un computador con las especificaciones anteriores.

El primer paso es configurar la red Profibus, partiendo de la instalación de cada una de las tarjetas (SMS CIF30-DPS, SMS CIF30-DPM) en los PC de cada uno de los bancos de laboratorio, necesarias para el montaje de esta red y de la respectiva conexión del cable de derivación al segmento de bus por medio del conector de junta en T, según las especificaciones del cuadro 10.

Una vez establecida la conexión física de cada esclavo (PC de cada banco de laboratorio) y del dispositivo maestro (SMS CIF30-DPM en un PC del banco de laboratorio) se procede a configurar a cada una de las tarjetas SMS CIF30-DPS como dispositivos esclavos a través del software SyCon, terminado esto se continua con la configuración de la tarjeta SMS CIF30-DPM como dispositivo maestro y al mismo tiempo se anexaran cada uno de los dispositivos que actuaran como esclavos en nuestra red.

El segundo paso es programar cada software supervisorio de cada banco para que establezca una comunicación con la tarjeta SMS CIF30-DPS instalada en cada uno de estos y así poder recibir los datos o variables necesarias para realizar el monitoreo y control de cada banco desde la estación maestra (PC con dispositivo maestro).

El ultimo paso es configurar el software InControl para que reciba y procese los datos provenientes de cada uno de los bancos, por tal motivo será necesario crear un proyecto de aplicación con su respectiva estrategia de control para cada banco conectado a la red, además se configurara el software InTouch para poder realizar la visualización y monitoreo de cada uno de los bancos y de sus respectivas variables de proceso, lo cual hace necesario crear un proyecto de aplicación para cada banco en este software supervisorio.

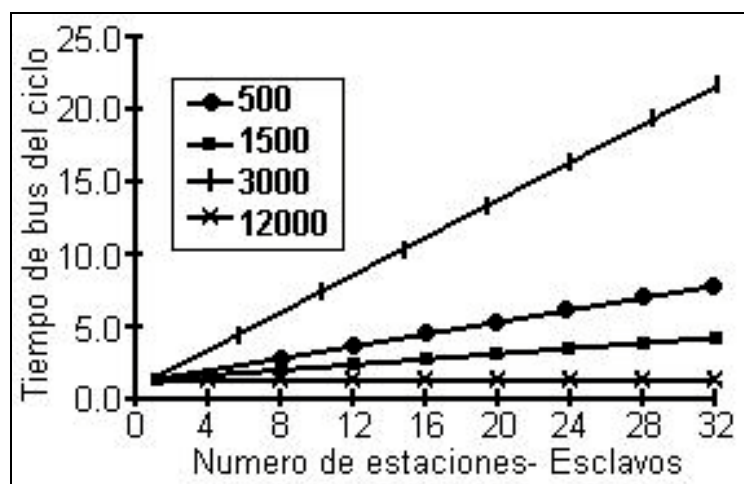
Terminados todos los pasos anteriores se está en capacidad de realizar monitoreo y control de todos los bancos de laboratorio conectados a la red Profibus. Por lo tanto desde la estación maestra (Dispositivo maestro) serán monitoreados todos los bancos conectados a la red Profibus (RS-485), los datos son tomados a través de las tarjetas SMS CIF30-DPS configuradas como esclavos e instaladas en cada banco conectado a la red permitiendo conocer todo lo referente a un proceso específico en cada banco a través de las variables fundamentales medidas por los instrumentos o dispositivos de medida en cada uno de los bancos. Otro aspecto fundamental es que esta red permite controlar a grandes distancias cada uno de

los procesos de cada banco, integrando cada uno de los instrumentos por medio de un estándar de comunicación (Profibus DP) único para cada banco.

Se escoge la tarjeta SMS CIF30-DPS debido a que posee un puerto Profibus DP esclavo el cual se puede conectar al puerto Profibus DP Maestro de la Tarjeta SMS CIF30-DPM instalada en el computador maestro, ésto por medio del estándar RS-485 o sea un par trenzado blindado; y además por su interoperabilidad muy eficaz a la hora de realizar la transmisión de datos por medio del procedimiento **Polling** usado para manejar la comunicación entre dispositivos maestros y sus dispositivos esclavos designados.

Para un número de esclavos de 8 el tiempo del ciclo de transmisión es aproximadamente 6 ms para distancias menores a 500 metros, este se obtiene de la siguiente grafica.

Figura 96. Numero de estaciones- tiempo de bus del ciclo



Una única velocidad de transmisión es seleccionada para todos los dispositivos en el bus cuando el sistema esta comisionado o configurado.

3.3.2 Cableado y equipo requerido:

- **Cables.** Se necesita un cable doble trenzado blindado para hacer el cableado (segmentos de bus o cables de derivación). Se puede usar cualquier cable doble trenzado blindado, los cuales pueden obtenerse en establecimientos comerciales, para conectar los bancos a la red PROFIBUS. Se recomienda el cable Sprecher+Schuh, tal como se muestra a continuación en el cuadro 11.
- **Conectores en junta en T.** Se necesitan conectores de junta en T para conectar los cables de derivación a los segmentos de bus. Se puede usar cualquier junta en T, las cuales se pueden obtener en cualquier establecimiento comercial, sin embargo se recomienda conectores Sprecher+Schuh, tal como se muestra en el cuadro 11.
- **Conectores de bus.** Se necesita un conector de bus para conectar su equipo a la red PROFIBUS DP. Puede usarse cualquier conector de bus compatible, los cuales pueden obtenerse en establecimientos comerciales.

Cuadro 11. Especificaciones de cables y conectores 2

Equipo	Tipo	Numero de catalogo	Numero de parte
Cable de derivación	Belden Sprecher+Schuh	3079A PDC-2, PDC-4	87.890.282-10
Conector de junta en T	Sprecher+Schuh	PTS-D	97.890.276-01
Cable de segmento de bus	Sprecher+Schuh	Ninguno	299.257.001
Segmento de bus 1m	Sprecher+Schuh	PCB-10	87.890.281-10
Conector	Sprecher+Schuh	Ninguno	87.890.283-01

En la red PROFIBUS los bancos de laboratorio son conectados al bus a través de un terminal de bus o un conector de conexión a bus. La tecnología de transmisión RS-485 es muy fácil de manejar, la instalación del cable par trenzado no requiere un conocimiento experto.

Todos los bancos son conectados a una estructura de bus, más de 32 estaciones (Maestro o Esclavos) pueden ser conectadas en un segmento. El estándar EN 50170 especifica usar línea tipo A de cable par trenzado blindado con las siguientes especificaciones:

Tabla 4. Estandar EN 50170

Parámetros	Valor
Impedancia	135 a 165 Ω
Capacitancia por unidad de longitud	< 30 pF/m
Resistencia de lazo	110 Ω /km
Diámetro del núcleo	0.64 mm
Corte transversal del núcleo	>0.34 mm ²

La máxima longitud del cable depende de la velocidad de transmisión. La longitud del cable especificada en la tabla de abajo esta basada en la línea tipo A, para nuestra red, que maneja distancias menores a 100 m la velocidad escogida es de 3 Mbps.

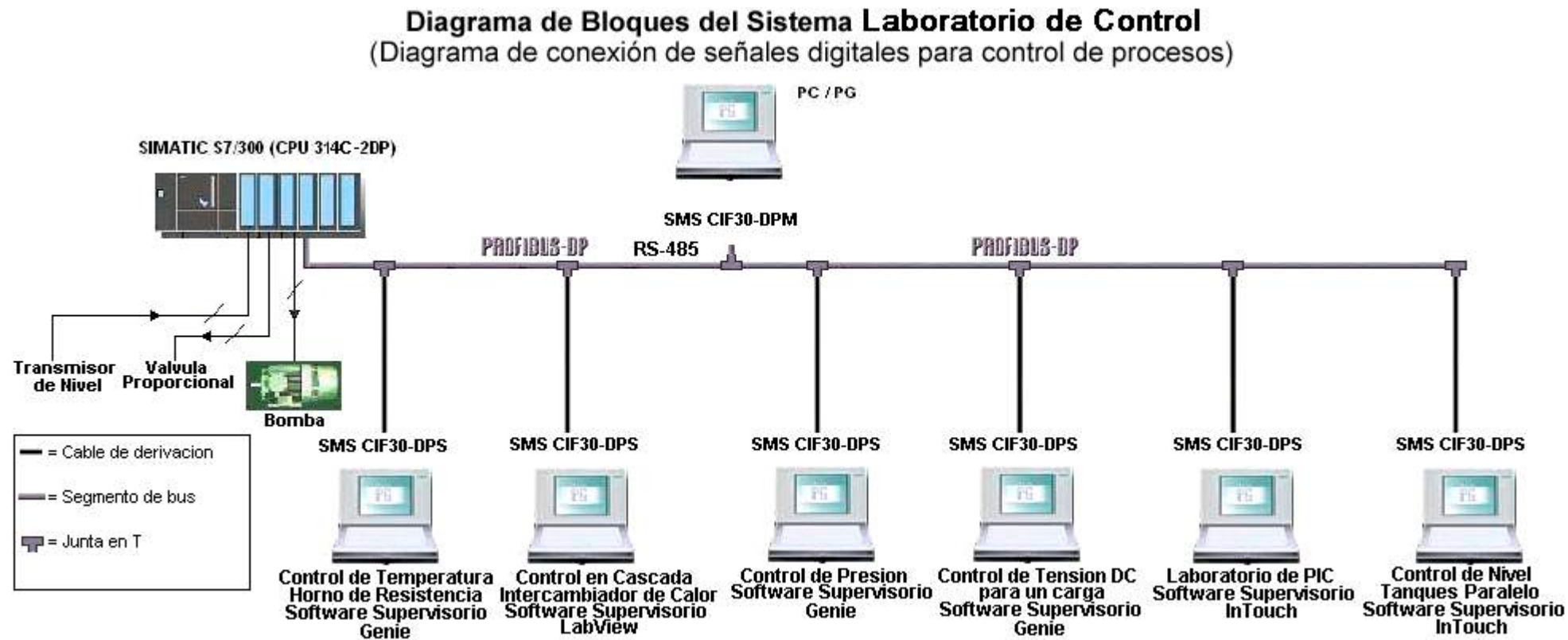
Tabla 5. Otras especificaciones línea tipo A

Tasa de baudios (kbit/s)	Distancia / Segmento (m)
9.6, 19.2, 93.75	1200
197.5	1000
500	400

Tasa de baudios (kbit/s)	Distancia / Segmento (m)
1500	200
3000, 6000, 12000	100

A continuación se muestra el diagrama del sistema de conexión descentralizado de las señales de entrada y salida tanto analógica como digital de control y supervisión del proceso (red Profibus del laboratorio) (Figura 97).

Figura 97. Diagrama de bloques del sistema Laboratorio de Control



CONCLUSIONES

- Con el diseño e implementación de este curso de capacitación se buscó que los estudiantes entendieran las funciones del InControl para crear soluciones de control para aplicaciones en la industria, convirtiendo en profesionales mas competitivos en el ámbito del control industrial, con esto se logra que nuestros egresados cumplan con las necesidades de la creciente demanda de personal capacitado en el manejo de software industriales por partes de las industrias del sector y del resto del país.
- El proyecto que se ha presentado como trabajo de grado busca cumplir con los objetivos planteados al iniciar el mismo, este tiene unos fines didácticos y educativos claramente marcados: comprender y analizar el funcionamiento del software InControl y desarrollar estrategias de control basados en sus lenguajes de programación (RLL, STL, SFC).
- Pensando en el interés que produciría un proyecto como éste desde el punto de vista didáctico, ya que permitirá acercar el aspecto teórico al aspecto practico, los autores se sienten satisfechos de poder contribuir al fortalecimiento de la enseñanza en el análisis y diseño de los sistemas de automatización industrial en la CUTB.

- Para poder desarrollar la implementación de un sistema de control basado en el software InControl se debe adquirir su licencia (archivo de licencia), necesaria para poder instalar correctamente todas las funciones y utilidades que posee el InControl para correr aplicaciones de control en la solución de sus procesos; ya que sin éste el InControl se instalará en modo demo (ejecución por 24 horas) y algunos FOE's no se ejecutarán bien y presentarán restricciones a la hora de su ejecución, todo esto acompañado de una serie de restricciones I/O que no permiten el procesamiento de datos de los puertos I/O, así como también de no permitir instalación de los drivers para las tarjetas I/O soportadas por InControl.

- El InControl por ser un software de control orientado a las redes industriales es muy complejo a la hora de realizar la implementación de un sistema de control, ya que resulta muy costoso adquirir las tarjetas necesarias para la adquisición de datos provenientes del proceso, así como también de los dispositivos y cableado necesario para el montaje de dicho sistema.

- Al momento de configurar la velocidad de transmisión de datos del PLC S7-300 y de la tarjeta CIF30-DPM es importante tener en cuenta la longitud del cable RS-485 utilizado en el montaje de la red Profibus por que ésta determina la velocidad de transmisión (Bits / segundos), para nuestro diseño en una longitud de cable RS-485 menor de 100 metros se establece una velocidad de 12 Mbps

hasta 3 Mbps, pero como en nuestro proceso no se procesarán demasiados datos se decidió por el valor menor de este rango ósea 3 Mbps.

- El controlador ON-OFF es una buena solución en los procesos que no requieren alta exactitud por eso es importante escoger un valor de brecha diferencial adecuado para evitar que el actuador (válvula proporcional) tenga una frecuencia de trabajo elevada y así prevenir que este entre en saturación y ocurra un mal desempeño del sistema, esto trae como consecuencia la limitación de la vida útil del elemento final (actuador).

- Es importante tener presente que al momento de realizar la implementación del sistema de control de nivel con el software InControl es de consideración la limitación que presenta la abertura de la válvula montada actualmente, la cual introduce una no linealidad de saturación en el lazo de control, lo cual no permite aplicar los métodos de sintonización de PID de ziegler-nichols.

- El software InControl es uno de los software mas avanzados y prácticos a la hora de realizar soluciones de control en grandes procesos debido a su orientación hacia las redes de comunicaciones industriales mas especialmente al estándar Fieldbus y a sus diferentes versiones (Profibus, Devicenet, Optomux, Pamux, etc.).

- Es de gran relevancia saber que para la realización de algunos de los pasos de configuración presentados en este trabajo es necesario contar con el archivo de licencia del software y de la tarjeta CIF30-DPM, así como también del PLC S7-300, de hay que nos vimos en la necesidad de contar con el apoyo y asesoría de la empresa Colsein Ltda. de la ciudad de Barranquilla.
- Mas que una ayuda autodidáctica, el tutorial por su extensión puede ser empleado como medio de consulta para quienes ya tienen conocimiento del software InControl.
- Los ejercicios interactivos del tutorial que acompañan a sus respectivas tareas no siguen un orden para realizar un proyecto específico, en cambio, logra familiarizar al estudiante con el ambiente y las herramientas empleadas en el InControl con mayor base teórica y permite conocer más sus alcances en la creación de proyectos.
- La aplicación final en el InTouch es bastante pesada (9.85 MB) para un sistema sencillo como lo es el Control de Nivel, pero por mostrar una buena presentación se hizo uso de numerosos bitmap lo que hace que el computador requiera mayor capacidad. Al momento de diseñar este tipo de aplicaciones en una empresa es necesario conocer primero las capacidades de hardware y software del computador o de todo el sistema donde se encuentre la planta para luego plantear el diseño.

GLOSARIO

Archivo Activo (Active File): archivo del programa que está contenido en la cabeza de la ventana Editor de programas con su barra de título resaltada. Los comandos que son ejecutados desde los menús o pulsando sobre el botón de la barra de herramientas se llevan a cabo en el archivo activo.

Factory Suite Factory Suite 2000 es el primero en el mundo que ha integrado componentes basados en sistemas MMI. Con Factory Suite 2000, tiene acceso a toda la información que necesita para correr su diseño de fábrica (visualización, optimización y control, colección de datos de tipos datos) y hacer su planta verdaderamente productiva.

FOE: InControl es compatible con las especificaciones del servidor ActiveX. El editor de Objetos de fábrica InControl (FOE) es un contenedor ActiveX, el cual lo habilita para usar controles ActiveX dentro de un proyecto InControl. Un control ActiveX debe estar instalado dentro del InControl antes que usted pueda configurarlo y correrlo. Después de la instalación, es llamado como un objeto de fábrica InControl (FOE). Como otro programas InControl, un FOE se puede ejecutar independientemente. También puede llamarlo para ejecución desde otro programa.

Tarjeta I/O: se introduce dentro de una de las hendiduras de expansión de la unidad de sistemas InControl y se conecta a las rejillas E/S periféricas y módulos. También llamado tarjeta (card) o tablero (board).

Etiqueta (Label): Elemento de programación gráfica en un programa SFC o RLL que identifica dónde el flujo del programa debe continuar desde su correspondiente salto.

POU El POU (unidad de organización de programas) está definido por el estándar IEC 61131-3 como la unidad de programación básica. InControl soporta estos POU: programas, función, bloque de función, y macro.

Proyecto (project) Organiza y agrupa los programas de aplicación y configura archivos para una aplicación en un subdirectorío aparte.

Lógica en Escalera por Relevos: RLL es un lenguaje de programación gráfica usado para describir la lógica de programas de aplicación basado en contactos de relevos eléctricos y bobinas análogas.

Runtime Engine: Módulo responsable para la programación y ejecución de la lógica del programa asociado con el código fuente del proyecto, e.g., SFC, RLL,

etc. Este módulo también se desempeña como un Suite Link o servidor DDE para Suite Link o clientes DDE tales como InTouch.

Gráfica de Funciones Secuenciales: SFC es un lenguaje de programación gráfica para esquematizar lógica secuencial usando Pasos, Transiciones y Acciones.

Paso Elemento gráfico en un SFC que representa un estado o lapso de tiempo en la ejecución del programa durante el cual las acciones y funciones asociadas con el Paso son realizadas.

Lenguaje de Texto Estructurado El lenguaje de programación de Texto Estructurado es un subconjunto de un conjunto de instrucciones basado en texto compatible con IEC – 61131.

Suite Link El driver Suite Link provee fiabilidad, alta velocidad de recopilación de datos para controladores, tarjetas PC y otros equipos. A diferencia de los DDE, el cual transfiere solamente un mensaje a la vez, Suite Link transfiere bloques de mensajes, permitiendo mas rapidez y transferencia de datos eficiente.

Tagname (Etiqueta): El nombre asignado a una variable en la base de datos InTouch para representar una variable InControl.

Variable Localización de la memoria interna que contiene datos del proyecto. El contenido de la información está definido por el tipo y pueden ser números reales, enteros, strings de caracteres, etc. El Administrador de Símbolos (Symbol Manager) es usado para definir una variable y asignarle un nombre simbólico y tipo de dato. También llamado *Symbol*.

BIBLIOGRAFÍA

- OGATA, katsuhiko. Ingeniería de Control Moderna, Prentice Hall - segunda edición: Hispanoamericana, S.A. 1970. p. 636-641.
- J. MALONEY, Timothy. Electrónica Industrial Moderna, Prentice Hall - tercera edición: p. 75-114.
- ALARCÓN, Sandra Milena. Curso de capacitación en el software InTouch (2002).
- Wonderware Corporation. FactorySuite System Administrator's Guide.
- Wonderware Corporation. InControl Serial Port Manual.
- Wonderware Corporation. InControl - SuiteLink (Wonderware) Manual.
- Wonderware Corporation. InControl I/O Scanner Board.
- Wonderware Corporation. InControl User's Guide.

- Wonderware Corporation. InTouch 7.1 Precourse.

- WONDERWARE, About us. http://wonderware.com/about/indsdas/mml_appstories.htm.

- ORTEGA, Diana. Entorno de supervisión y control de un sistema SCADA vía Internet. <http://www.etsit.ulpgc.es/guia/proyecto25.htm>.

- I/O drivers InControl. www.integratedsoftsolution.com

- Estándar RS-485. [www.Arcelet.com /RS485.info_tutorial](http://www.Arcelet.com/RS485.info_tutorial)

Wonderware[®] FactorySuite[™] InControl[™]

Guía de Usuario

Revisión E

Julio, 1999

Wonderware Corporation

CONTENIDO

	Pág.
Capitulo 1 - INICIO RÁPIDO CON INCONTROL.....	1-1
INTRODUCCIÓN.....	1-2
Lenguajes de Programación	1-2
Soporte para Múltiples Plataformas.....	1-5
Requerimientos del Sistema.....	1-6
SOPORTE TÉCNICO.....	1-7
LÍNEA DIRECTA DE INSTALACIÓN.....	1-11
Antes de Empezar	1-11
Selección de las Opciones de Configuración.....	1-13
Adición de Componentes	1-15
DESINSTALACIÓN DEL INCONTROL.....	1-16
ACERCA DE LA GUÍA DE USUARIO.....	1-16
EJECUCIÓN DEL INCONTROL: INICIACIÓN.....	1-17
Creación de un Proyecto.....	1-17
QUE SIGUE?	1-19
Capitulo 2 - AMBIENTE INCONTROL.....	2-1
VENTANAS DE DESARROLLO Y EJECUCIÓN.....	2-2
Ventana de Desarrollo.....	2-2
Ventana de Ejecución.....	2-3
BARRA DE HERRAMIENTAS.....	2-7
Barra de Herramientas Standard.....	2-7
Barra de Herramientas Runtime.....	2-8
BARRA DE MENÚ.....	2-10
Comandos Insert.....	2-10
Comandos File.....	2-11
Comandos Edit.....	2-12
Comandos View.....	2-13
Comandos Runtime.....	2-15

Comandos Tools.....	2-18
Comandos Window.....	2-19
Comandos Help.....	2-19
Capitulo 3 - ESTABLECIMIENTO DE LA SEGURIDAD.....	3-1
INTRODUCCIÓN.....	3-2
REGISTRO DE UN USUARIO.....	3-4
Registro de un nuevo usuario	3-4
Eliminación del registro a un usuario.....	3-5
Cambio de la contraseña.....	3-6
MANEJO DE LA SEGURIDAD.....	3-6
Adición de un nombre de usuario.....	3-6
Cambio de la clave y eliminación de un nombre de usuario.....	3-7
BLOQUEO DE ALGORITMOS SFC.....	3-8
LA SEGURIDAD DE WINDOWS NT.....	3-10
Capitulo 4 - CONFIGURACIÓN I/O.....	4-1
INTRODUCCIÓN.....	4-2
ADICIÓN DE DRIVERS.....	4-7
Adición de un driver a la ventana de proyectos.....	4-7
Remover o eliminar un driver.....	4-8
VERIFICACIÓN DE LA VERSIÓN DEL DRIVER.....	4-9
CONFIGURACIÓN I/O.....	4-11
SIMULACIÓN I/O.....	4-12
Capitulo 5 - MANEJO DE PROYECTOS.....	5-1
INTRODUCCIÓN.....	5-2
Programas.....	5-2
Bloques de Funciones.....	5-3
Funciones.....	5-5
Macros.....	5-5
MANEJO DE PROYECTOS.....	5-7
Creación de un proyectos.....	5-8

Eliminación de un proyectos.....	5-10
Adición de un proyectos.....	5-11
Modificación del nombre y la descripción de un proyectos.....	5-12
Adición de una nueva POU a un proyectos.....	5-13
Adición de una POU existente a un proyecto.....	5-15
Como remover una POU.....	5-17
Como renombrar a una POU.....	5-18
Organización de un proyecto.....	5-19
Definición de Bloques de Funciones.....	5-21
Definición de Funciones.....	5-26
Adición y organización de drivers I/O.....	5-31
Configuración del Runtime Engine (software de ejecución)	5-32
Ingreso al Symbol Manager (administrador de símbolos)	5-34
Designación de la plataforma del hardware.....	5-35
Capitulo 6 - Defining Variables.....	6-1
INTRODUCCIÓN.....	6-2
Definición de Variables.....	6-2
Variables locales y globales.....	6-2
Variables asignadas con un valor constante.....	6-4
Variables retentivas.....	6-5
TIPOS DE DATOS DE LAS VARIABLES.....	6-7
LREAL.....	6-8
REAL.....	6-8
DINT	6-9
INT.....	6-9
SINT.....	6-9
Números enteros sin registro.....	6-9
DWORD.....	6-10
WORD.....	6-10
BYTE.....	6-11
BOOL.....	6-11
DT	6-11
DATE.....	6-12

TOD.....	6-12
Uso de tipos de datos basados en tiempo y fecha en expresiones.....	6-13
TIME.....	6-14
TMR	6-15
FILE.....	6-17
STRING.....	6-19
USER-DEFINED.....	6-20
Conversión de tipos de datos.....	6-21
SYMBOL MANAGER (ADMINISTRADOR DE SÍMBOLOS).....	6-22
Uso de la barra de tarea del Symbol Manager.....	6-25
Consejos de edición - menú de contexto.....	6-26
Consejos de edición– Cambio en el orden de los miembros.....	6-27
Consejos de edición– copiar, pegar y mover símbolos.....	6-27
CREACIÓN DE UNA VARIABLE.....	6-29
CREACIÓN DE UN ARREGLO DE VARIABLES.....	6-31
Referenciar arreglos.....	6-32
ASIGNACIÓN DE UN NOMBRE A UN BIT EN UNA VARIABLE.....	6-33
CREACIÓN DE UN TIPO DE DATO USER-DEFINED (DEFINIDO POR EL	
USUARIO).....	6-36
Diseño de un de tipo de dato propio.....	6-36
Uso del tipo de dato user-defined.....	6-38
PRESENTACIÓN DE INFORMACIÓN DE LAS VARIABLES.....	6-39
VARIABLES DE SISTEMA.....	6-40
Variables de sistemas del Runtime Engine.....	6-42
TRANSFERENCIA DE BASES DE DATOS DE SÍMBOLOS.....	6-44
Intercambio de Símbolos Entre Proyectos InControl.....	6-44
Reportes De Referencias Cruzadas De Símbolos.....	6-45
Intercambio de Símbolos Entre InControl e InTouch.....	6-45
Importación y Exportación de Símbolos.....	6-46
Formato CSV de InControl para Archivos.....	6-48
Edición de archivos de símbolos.....	6-50
 Capítulo 7-USO DEL EDITOR RLL.....	 7-1
CREACIÓN DE UN PROGRAMA RLL.....	7-2

HERRAMIENTAS RLL.....	7-5
Uso de las herramientas RLL y la barra de Menú.....	7-5
Consejos de edición.....	7-6
ADICIÓN DE ELEMENTOS DE PROGRAMA.....	7-9
Adición de un contacto.....	7-9
Adición de una bobina.....	7-11
Adición de un escalón.....	7-13
Adición de un ramal o derivación OR.....	7-15
Eliminación de un ramal o bifurcación OR.....	7-17
Adición de una etiqueta.....	7-18
Adición de una bobina de salto.....	7-20
Adición de un bloque de función.....	7-21
Adición de una bobina de transición SFC.....	7-24
EDICIÓN DE ELEMENTOS DE PROGRAMA.....	7-25
ADICIÓN DE COMENTARIOS.....	7-25

Capítulo 8-ELEMENTOS DE PROGRAMAS RLL.....8-1

FLUJO DE ENERGIA EN UN PROGRAMA RLL.....	8-2
Resolución de una Lógica Sencilla de Contactos y Bobinas.....	8-2
Bloques de Funciones y Flujo de Energía.....	8-2
TIPOS DE CONTACTOS RLL.....	8-3
Contacto Abierto.....	8-3
Contacto Cerrado.....	8-4
Contacto de Transición Positiva.....	8-4
Contacto de Transición Negativa.....	8-4
TIPOS DE BOBINAS RLL.....	8-5
Bobina de Salida.....	8-5
Bobina de Salida Negativa.....	8-5
Bobina de Cierre (Latch).....	8-6
Bobina de Reset (Unlatch).....	8-6
Bobina de Transición Positiva.....	8-6
Bobina de Transición Negativa.....	8-7
BOBINA DE SALTOS / ETIQUETAS.....	8-8

Capítulo 9 - USO DEL EDITOR SFC.....	9-1
CREACIÓN DE UN PROGRAMA SFC.....	9-2
HERRAMIENTAS SFC.....	9-4
Uso de las herramientas SFC y la barra de Menú.....	9-4
Consejos de edición.....	9-6
ADICIÓN DE ELEMENTOS DE PROGRAMA.....	9-8
Adición de un paso(Step)	9-8
Adición de una Transición(Transition)	9-10
Adición de un Macro Paso(Macro Step)	9-14
Adición de una Acción(Action)	9-15
Adición de un Salto(Jump)	9-18
Adición de una etiqueta(Label)	9-20
Adición de un Lazo(Loop)	9-21
Adición de una Divergencia de Selección	9-22
Adición de una Divergencia Paralela	9-24
Adición de un Paso de Librería	9-26
Adición de un Paso desde la Librería de Pasos	9-29
Adición de comentarios al programa	9-30
Capitulo 10- ELEMENTOS DE PROGRAMAS SFC	10-1
QUE ES UN SFC?	10-2
Elementos de un SFC	10-2
Elementos de un SFC	10-2
PASOS (STEP)	10-5
Funciones	10-5
Parámetros	10-6
Código	10-8
Capitulo 11- LENGUAJE DE TEXTO ESTRUCTURADO.....	11-1
ELEMENTOS DE TEXTO ESTRUCTURADO	11-2
EXPRESIONES	11-3
Operadores	11-3
Tipos de Datos	11-5

TIPOS DE ORDENES	11-6
Asignación	11-7
BREAK	11-8
CASE	11-8
Comment (Comentario)	11-10
EXIT	11-11
FOR	11-12
Llamadas de Funciones y Procedimientos	11-14
IF	11-15
INCLUDE	11-16
LABEL (Etiqueta)	11-17
REPEAT	11-17
RETURN	11-18
SCAN	11-18
WHILE	11-19
#Pragma	11-20
FUNCIONES Y PROCEDIMIENTOS DE TEXTO ESTRUCTURADO	11-21
ABORT_ALL	11-24
ABS	11-24
ACOS	11-25
ARRAY_TO_STRING	11-25
ASIN	11-27
ATAN	11-28
BCD_TO_INT	11-28
CLOSEFILE	11-29
CONCAT	11-34
COPYFILE	11-34
COS	11-36
DATE_TO_REAL	11-36
DATE_TO_STRING	11-37
DELETE	11-37
DELETEFILE	11-38
EXP	11-39
FIND.....	11-40

INSERT	11-41
INT_TO_BCD	11-42
INT_TO_STRING	11-43
LEFT	11-44
LEN	11-44
LN	11-45
LOG	11-46
MAX	11-46
MID	11-47
MIN	11-47
MSGWND	11-48
NEWFILE	11-49
OPENFILE	11-50
READFILE	11-52
REAL_TO_DATE	11-54
REAL_TO_STRING	11-54
REAL_TO_TIME	11-55
REPLACE	11-56
REWINDFILE	11-57
RIGHT	11-58
ROL	11-59
ROR	11-60
SHL	11-61
SHR	11-62
SIN	11-63
SQRT	11-63
STRING_TO_ARRAY	11-64
STRING_TO_DATE	11-65
STRING_TO_INT	11-65
TAN	11-67
TIME_TO_REAL	11-68
TIME_TO_STRING	11-69
TRUNC	11-70
WRITEFILE	11-71

Capitulo 12-USO DEL EDITOR DE TEXTO ESTRUCTURADO.....	12-1
CREACIÓN DE UN PROGRAMA STL	12-2
HERRAMIENTAS DE TEXTO ESTRUCTURADO	12-4
Las herramientas de Texto Estructurado y la barra de Menú	12-4
Consejos de edición	12-6
INGRESO DEL CÓDIGO DEL PROGRAMA	12-7
Capitulo 13- EDITOR DE OBJETOS DE FÁBRICA INCONTROL.....	13-1
DEFINICIÓN DE OBJETOS DE FABRICA.....	13-2
INTEGRACIÓN DE CONTROLES ACTIVEX EN INCONTROL.....	13-4
Instalando Controles ActiveX.....	13-4
Organizando FOEs.....	13-6
Adicionando FOEs a un Proyecto.....	13-7
CONFIGURACIÓN DE OBJETOS FABRICA.....	13-7
USANDO LA BARRA DE HERRAMIENTAS Y DE MENÚ.....	13-9
EJECUCIÓN DE OBJETOS FABRICA INCONTROL.....	13-10
Corriendo y Controlando FOEs.....	13-10
Ejecución de Animaciones.....	13-11
Cargando Parámetros.....	13-12
Usando el resto de los FOEs.....	13-12
Referencia de un Objeto de Fábrica InControl.....	13-13
USO DEL FOE PID DE WONDERWARE.....	13-15
Operación del software de ejecución (Runtime).....	13-15
Cargando Parámetros.....	13-16
Ventana de presentación del PID.....	13-17
Parámetros Generales de Configuración.....	13-18
Configurando Parámetros de variables de proceso.....	13-20
Configurando Parámetros de Setpoint.....	13-22
Configurando parámetros de Salida.....	13-24
Configurando Parámetros de Afinación (Tuning)	13-26
Control Gap	13-28
Configurando Alarmas de Lazo	13-28
PV Alarms and Deviation	13-31

Acción Histéresis	13-32
Revisando Versión FOE	13-32
SÍMBOLOS Y MÉTODOS PID	13-33
TEORÍA DE OPERACIÓN PID	13-39
Algoritmo PID	13-39
Modos de lazo	13-42
Reset Windup	13-43
USO DEL FOE DE ALARMA ANÁLOGA DE Wonderware	13-45
Cargando Parámetros	13-46
Ventana principal de Alarma Análoga	13-47
Configurando Parámetros Generales	13-48
Configurando Parámetros de Variable del proceso	13-49
Configurando los parámetros del Setpoint Objetivo	13-51
Configurando Alarmas	13-52
Chequeando la versión del FOE	13-55
SÍMBOLOS Y MÉTODOS DE ALARMAS ANÁLOGAS	13-55
USO DEL FOE INFORMACIÓN DE PROYECTO	13-59
Configurando el FOE de Información de Proyectos	13-59
Enviando Información de Nodos y proyectos	13-59
Reportando y Eliminando Fallas	13-60
Programas de Pasos individuales	13-61
Guardando Valores Retentivos y Forzados	13-62
Capítulo 14 – EJECUCIÓN DE PROYECTOS.....	14-1
SELECCIÓN DE OPCIONES DE EJECUCIÓN.....	14-2
CONEXIÓN AL RUNTIME ENGINE.....	14-5
Verificación del Nodo conectado.....	14-6
Uso del Monitor del Runtime Engine.....	14-7
Uso de Comandos del Monitor del Runtime Engine.....	14-8
Ejecución y desactivación del Monitor del Runtime Engine.....	14-10
VALIDACIÓN Y DESCARGA.....	14-11
Validación de Proyectos.....	14-11
Descarga de un proyecto.....	14-13
Validación de un Programa Individual.....	14-17

Descarga de un Programa Individual.....	14-18
EJECUCIÓN DE PROYECTOS Y PROGRAMAS.....	14-21
Ejecución de un proyecto.....	14-21
Ejecución de un Programa Individual.....	14-24
DETENCIÓN DE LA EJECUCIÓN DE PROYECTOS Y DE PROGRAMAS.....	14-62
Detención de la ejecución de un Proyecto.....	14-26
Detención de la Ejecución de un Programa	14-27
ORDEN DE EJECUCIÓN DE PROYECTOS Y PROGRAMAS.....	14-29
Orden de los Programas en la Ventana de Ejecución.....	14-29
Prioridad de los Programas en la Execution View.....	14-30
DEPURACIÓN DE UN PROGRAMA.....	14-31
Verificación de la Barra de Estados	14-31
Verificación del Modo del Programa.....	14-31
Análisis Sencillo de un Proyecto y Programa.....	14-32
Uso de BreakPoints (Punto de detención)	14-32
Ejecución de un Programa Paso a Paso.....	14-33
Monitoreando variables de programas.....	14-35
Adicionando Múltiples Variables a la Ventana Watch.....	14-37
Eliminación de una variable.....	14-38
Adición de una Tabla de Valores a la Ventana Watch.....	14-38
Modificación y Forzado de una Variable.....	14-40
Valores validos para la variable de sistema Mode	14-43
Ajustando la Tasa de Actualización	14-44
Pausa en la actualización de la ventana Watch	14-45
Variables no Forzadas	14-45
Mostrando Variables Forzadas	14-45
Uso del menú de la ventana Watch	14-46
Uso de la ventana Watch autónoma	14-48
Uso de las Variables del Sistema Runtime Engine	14-51
Verificación del Wonderware Logger (registro)	14-53
Capitulo 15 - InControl e InTouch	15-1
FUNCIONES INCONTROL SOPORTADAS POR INTOUCH.....	15-2
Instalación de los wizards InControl.....	15-2

USO DEL WIZARD DE PROYECTOS INCONTROL.....	15-3
USO DEL WIZARD DE CONFIGURACIÓN DEL RUNTIME ENGINE.....	15-5
USO DEL WIZARD DE MODO INCONTROL.....	15-6
USO DEL WIZARD EDITOR DE EJECUCIÓN INCONTROL.....	15-7
USO DEL WIZARD DE ELIMINACIÓN DE FALLAS INCONTROL.....	15-9
USO DEL WIZARD DE ADICIÓN DE ETIQUETAS DE EJECUCIÓN INCONTROL.....	15-10
Muestra de los símbolos InControl.....	15-10
Vinculación de Símbolos InControl a Etiquetas InTouch.....	15-12
Vinculación de Etiquetas Sobre Sistemas Remotos.....	15-14
USO DEL VISUALIZADOR DE ETIQUETAS DE INTOUCH.....	15-15
NODOS Y NOMBRES DE PROYECTOS.....	15-16
FUNCIONES DE PROGRAMACIÓN INCONTROL.....	15-16

Capítulo 16 - ADMINISTRACIÓN DEL SISTEMA INCONTROL.....16-1

DESCRIPCIÓN DE LOS EVENTOS DEL RUNTIME ENGINE.....	16-2
Ejecución SFC.....	16-5
Ejecución de programa de texto estructurado.....	16-6
Ejecución de RLL.....	16-6
Ejecución de FOE.....	16-6
Ejecución de Programas y Fijando programas.....	16-7
CONFIGURACIÓN DEL RUNTIME ENGINE.....	16-8
Acceso a las propiedades del RUNTIME ENGINE.....	16-8
Revisando las Propiedades Generales.....	16-9
Fijando Tiempo de Análisis.....	16-9
REVISIÓN EN LA OPERACIÓN DEL RUNTIME ENGINE.....	16-11
Observación de los Datos de Estado.....	16-11
Ajustando el Tiempo de Análisis.....	16-11
Ejemplos de Scan Normal y de Saltos.....	16-14
Operaciones Scan y el Watchdog Timer.....	16-18
Consideraciones I/O.....	16-18
Buscando en el Logger Data.....	16-19
Revisando Nodos Remotos.....	16-20
Eliminando modos de falla y condiciones de error.....	16-22

Modo Fault del Runtime Engine.....	16-22
Programa en Modo Fault.....	16-23
Errores I/O y de Hardware.....	16-24
Configuración del Inicio del Servicio del Runtime Engine.....	16-24
MANEJO DE FALLAS DE ENERGÍA.....	16-24
Usando un Suministro de Energía Continuo.....	16-24
Uso de las Variables del Sistema.....	16-25
Variable Forzadas y Retentivas, y Fallas de Energía.....	16-36
Reinicio Automático de Proyectos.....	16-27
Configuración del Modo de Reinicio (Restart).....	16-27
Copia de seguridad de Variables Forzadas y Retentivas.....	16-28
EJECUCIÓN DE MÚLTIPLES PROYECTOS.....	16-29
Configuración de una Conexión al Nodo Remoto.....	16-32
Transferencia y Registro de Datos de Proyecto	16-34
Configuración I/O en el Nodo Remoto	16-35
CAMBIO DE LA CLAVE EN EL REGISTRO DEL SISTEMA	16-36
Cambios en la fijación de registros FOE	16-36
Mostrando Advertencia de Compilador	16-37
SOPORTES DE CONTENIDO, MODALIDAD Y TIEMPO	16-39
INGRESO DE LA CONFIGURACIÓN DEL VISOR DE SUCESOS	16-40
Apendice A – PALABRAS RESERVADAS.....	A-1
Apendice B - TIPOS DE DATOS.....	B-1
CATEGORÍA DE TIPOS DE DATOS.....	B-2
RANGOS TIPOS DE DATOS.....	B-3
Apendice C- MONITOREO DE DATOS POR DDE/SuiteLink.....	C-1
INTRODUCCIÓN	C-2
MONITOREO DE VARIABLES DESDE INTOUCH.....	C-3
MONITOREO DE VARIABLES DESDE EXCEL.....	C-5
Apendice D - EXTENSIONES IEC 61131-3	D-1

SFC Y PROGRAMAS DE TEXTO ESTRUCTURADO.....	D-2
Consideraciones SFC.....	D-2
Consideraciones STL.....	D-3
PROGRAMAS RLL.....	D-4
TIPOS DE DATOS.....	D-5
Tipos de Datos no Soportados.....	D-5
Conversión Tipos de Datos.....	D-5
ESPECIFICACIÓN PARÁMETROS PARA INCONTROL.....	D-6
Condiciones de Error.....	D-9
Apendice E - TECLAS DE ACCESO DIRECTO.....	E-1
OPERACIONES GENERALES.....	E-2
Operaciones de ventanas.....	E-3
VENTANA DE PROYECTOS.....	E-3
VENTANA OUTPUT(SALIDA).....	E-3
PROJECT MANAGER.....	E-4
VENTANA WATCH.....	E-4
EDITOR DE PROGRAMAS.....	E-5
SYMBOL MANAGER	E-7
GLOSARIO DE TÉRMINOS	G-1

INICIO RÁPIDO CON INCONTROL

Este capítulo contiene los requerimientos de software y hardware para el InControl, instrucciones para instalar y correr el InControl, cómo conseguir iniciar la programación y la información de soporte técnico.



CONTENIDO

- INTRODUCCIÓN.
- SOPORTE TÉCNICO.
- LÍNEA DIRECTA DE INSTALACIÓN.
- DESINSTALACIÓN DEL INCONTROL.
- ACERCA DE LA GUÍA DE USUARIO.
- EJECUCIÓN DEL INCONTROL: INICIO RÁPIDO.
- QUÉ SIGUE?

This product is licensed to:

Company:	N/A
SN:	N/A
Expires:	N/A
Locked To:	N/A

7.1.0.1337

INTRODUCCIÓN

El InControl es un componente de la familia de herramientas incluidas en la FactorySuite (Suite de Fábrica) de Wonderware en la solución de procesos de control para aplicaciones de planta en fábricas. InControl es un paquete de arquitectura abierta que permite diseñar, crear, medir y correr programas de aplicación para controlar sus procesos. Está diseñado para la integración cerrada con otros componentes de la FactorySuite de Wonderware, entre ellos InTouch, InBatch, InTrack e IndustrialSQL Server.

InControl incorpora los últimos estándares internacionales para el diseño de soluciones de automatización. Es compatible con la especificación de lenguajes de programación internacional IEC-61131-3.

Lenguajes de Programación

InControl crea soluciones de automatización en la industria basado en los siguientes lenguajes de texto y gráfico.

➤ **Lógica en Escalera Conmutada (Relay Ladder Logic) RLL.**

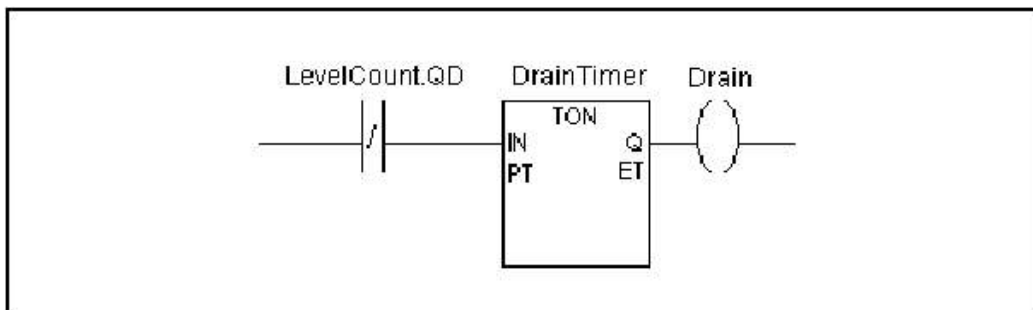


Figura 1. Lógica en escalera conmutada.

➤ **Cartas de Funciones Secuenciales (Sequential Function Chart) SFC.**

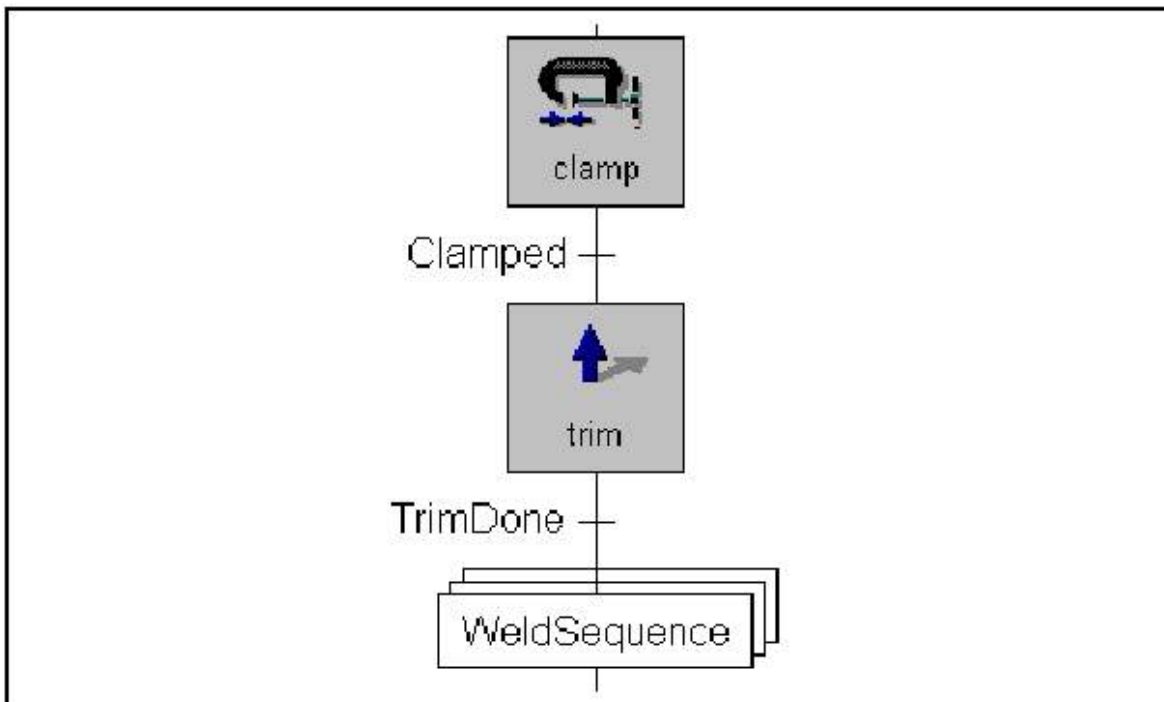


Figura 2. Cartas de funciones secuenciales.

➤ **Lenguaje de Texto Estructurado (Structured Text Language) STL.**

```
S Loop_Cycle.stl
(^Sets PID output to 0 if PID is in manual mode^)
IF Loop1.InManual THEN
  Loop1.Out := 0;
END IF;

(^Cycle PID Loop between setpoints of 0 and 100 ^)
IF ABS (Loop1.error) < 0.5 AND Loop1.InAuto THEN
  IF Loop1.sp >= 100 THEN
    Loop1.Sp := 0;
  ELSE
    Loop1.Sp := 100;
  END IF;
END IF;
```

Figura 3. Lenguaje de texto estructurado.

- **Objetos de Fábrica InControl.** InControl es compatible con la especificación de servidores ActiveX. El Editor de Objetos de Fábrica (**F**actory **O**bject **E**ditor), **FOE** de InControl es un contenedor ActiveX, el cual le permite agregar controles ActiveX al proyecto. Los siguientes FOE's están descritos en este manual:
- Use el FOE PID de InControl (PID InControl FOE) para manejar las funciones de lazo PID.
 - Use el FOE de Alarmas Análogas de InControl (Analog Alarm InControl FOE) para monitorear una señal de entrada análoga en condiciones de alarma.
 - Use el FOE de Información de Proyectos de InControl (Project Information InControl FOE) para enviar información del proyecto y del nodo a otra aplicación, así como el InTouch, para ejecutar varios comandos de ejecución, para borrar faltas, o para guardar valores forzados y retentivos de las variables.

Estos **Objetos de Fabrica** están descritos en el capítulo "USO DEL EDITOR DE OBJETOS DE FÁBRICA DE INCONTROL".

Soporte para Múltiples Plataformas

Algunas versiones de InControl soportan múltiples plataformas de hardware para su ejecución. Si ha adquirido una de estas versiones, entonces cuando cree un nuevo proyecto, InControl le sugiere seleccionar la plataforma (la designada) donde intenta correr el proyecto. Esto le permite desarrollar un proyecto en el ambiente Windows NT, y luego cargar y correr el proyecto sobre cualquiera de las diferentes plataformas, incluso en sistemas Windows CE integrados. Para más información diríjase a su distribuidor Wonderware.

Usted puede convertir un proyecto que fue desarrollado para un objetivo y correrlo sobre otro objetivo designado. Note que algunas tarjetas I/O (Entrada / Salida) no pueden convertir si ellas no soportan la plataforma de hardware designada. Además, para la conversión de proyectos, se pueden insertar archivos de programas, los cuales son desarrollados en un proyecto para un determinado objetivo, dentro de un proyecto con un objetivo diferente.

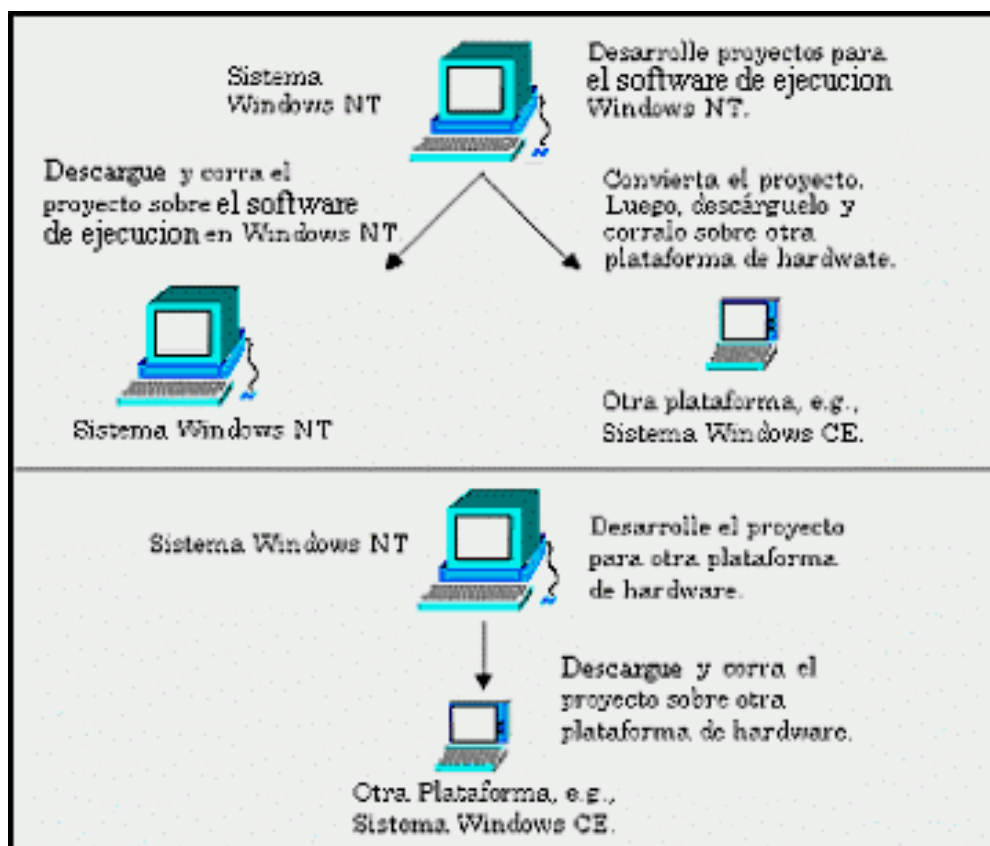


Figura 4. Ejecución de proyectos sobre otra plataforma

Requerimientos del Sistema

El software InControl esta diseñado para correr sobre cualquier computador IBM. Antes de instalar el InControl, verifique que su sistema reúna los siguientes requerimientos.

➤ Para el Desarrollo de Proyectos InControl

- Sistema basado en un procesador Pentium (procesador Pentium de 200 MHz o más rápido es recomendado).
- Al menos 32 MB de memoria RAM (48 MB es recomendable y 128 MB es lo recomendado para muchos programas y símbolos).
- Unidad de CD-ROM.
- Sistema operativo Microsoft Windows NT (versión 4.x, paquete de servicio 5).
- Por lo menos 80 MB de espacio en disco duro es lo necesario en disponibilidad (128 MB es lo recomendado).
- Características del monitor: VGA con 256 colores y 640 x 480 píxeles de resolución. (256 colores y 800 x 600 píxeles de resolución es lo recomendado).

Software de Ejecución InControl (InControl Runtime Engine) — Sistema Windows NT. Por lo menos 32 MB en RAM (48 MB es lo recomendado) y un mínimo de 20 MB de espacio en disco duro es lo requerido para el software de ejecución.

Software de Ejecución InControl (InControl Runtime Engine) — Sistema Windows CE. InControl soporta aplicaciones específicas de proveedor en sistemas Windows CE, basados en procesos fabricados por Intel (X86, StrongArm), Itachi, y MIPS. Para información más detallada, contacte su distribuidor Wonderware.

SOPORTE TÉCNICO

El soporte técnico Wonderware ofrece una variedad de opciones de soporte para responder cualquier pregunta sobre los productos Wonderware y su implementación. Consultar los capítulos pertinentes en su documentación InControl para una posible solución de cualquier problema que pueda encontrar. Si cree necesario contactar el soporte técnico de InControl para su asistencia, por favor tenga en cuenta la siguiente información disponible:

- El número del serial y el número de la versión del software. Pulse en **About InControl** sobre el menú **Help**, o en **Configure** sobre el menú **Runtime** para determinar la versión del InControl. También puede hacer clic en **About Engine Monitor** sobre el icono de la pantalla del Runtime Engine.

Si el InControl no está corriendo, puede usar el explorador de Windows NT para revisar las propiedades de la versión (pulse el botón derecho sobre el nombre del archivo) del archivo ICDev.exe o RTEngine.exe, localizado en el directorio donde InControl esta instalado.

Si el InControl no esta instalado, puede determinar la versión revisando el archivo Versión.txt, localizado en el subdirectorio InControl del disco compacto distribuido.

El número de la versión del Driver I/O aparece en la barra de título del primer cuadro de diálogo de configuración que aparece durante la configuración.

- Número del contrato de soporte (Support Contract Number). Su nombre debe ser uno de los tres nombres de contactos especificado en el contrato.
- Redacción exacta del mensaje de error encontrado en el sistema.
- Naturaleza del problema y detalles que le indiquen que debe hacer para resolver el problema y resultados.
- Información de configuración del sistema. Consulte "[Requerimientos del Sistema](#)"(Pagina 1-5).

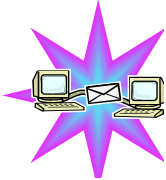
Obtención de Soporte Técnico InControl



Teléfono — En U.S.A, llame al 949-727-3299. En Europa, llame al 31-20-642-0514. En Asia, llame al 65-276-1508.



Internet— Wonderware provee acceso e-mail a través de Internet. Para acceder a la página de Soporte Técnico Wonderware, use <http://www.wonderware.com>.



E-Mail—Para obtener información de Wonderware por e-Mail, elija las siguientes direcciones:

Wondertech.wonderware.com

Sup_europe@wonderware.com

Sup_asia@wonderware.com



FAX— Puede enviar por Fax su pregunta a Soporte Técnico al 949-727-9733 en U.S.A, al 49-89-45-0558-222 en Europa, o 65-276-1506 en Asia (Singapur).

**Servicio de Boletines (Bulletin Board Service) BBS —**

Información técnica disponible las 24 horas al día en este servicio de boletines al número 949-727-0726. Use esta comunicación: 8 bits de datos, 1 BIT de detención, ninguna paridad.

Para soporte técnico sobre los módulos de instalación y uso de Wonderware FactorySuite o FactorySuite Plus, llame a su distribuidor Wonderware. Su distribuidor está entrenado para proporcionarle soporte local personalizado en FactorySuite y FactorySuitePlus. Para determinar su distribuidor local, llame a la oficina de ventas regional Wonderware más cercana:

Wonderware Corporation

Irvine, California, U.S.A.
949-727-3200

Wonderware Italia

Milan, Italia
39-332-298-444

Wonderware Argentina

Buenos Aires, Argentina
54-11-4-328-8150

Wonderware México

Ciudad de México, México
5-511-79-62

Wonderware GmbH

Munich, Alemania
49-89-4505580

Wonderware Korea

Seul, Korea
82-2-565-3807

Wonderware Francia

Paris, Francia

Wonderware Japon

Tokyo, Japon
81-3-3275-0699



Distribuidor en Colombia
Barranquilla

Cra 51B No. 82-254 Oficina 57 Piso 4
Telf: 378 4127 - 378 3497
Fax: 378 4005
E-mail: infobaq@colsein.com.co
Centro Comercial Bahía

Cali

Carrera 34 No. 4D -80
Edificio San Fernando Of. 205
Telf: 5560702 Telfax: 5572091
E-mail: infocali@colsein.com.co

Medellín

Carrera 43A No. 1sur-188
Torre Empresarial Bancafé
Oficina 609
Tel. (574) 3119766
Fax. (574) 3119790
E-mail: infomed@colsein.com.co
Celular (033) 233 5440

Bucaramanga

Calle 48 No. 32-14 Edificio Tempo I
Telf: 097-6478512
Telfax: 097-6477435

LÍNEA DIRECTA DE INSTALACIÓN

Para una descripción completa del procedimiento de instalación, consulte la *Guía del Administrador del sistema FactorySuite*.

El InControl esta distribuido en un disco compacto que hace parte de la Wonderware FactorySuite (suite de fábrica wonderware). Cuando se ejecuta el programa de configuración, el sistema hace lo siguiente:

- Crea los directorios necesarios en su disco duro, y copia los archivos dentro de los directorios apropiados.
- Establece la carpeta de programas e iconos.
- Modifica el registro del sistema.



Antes de Empezar

Uso de InTouch con InControl

Si intenta usar InTouch con InControl en iguales sistemas, es recomendable que instale InTouch antes de instalar InControl. Si instala primero InControl, debe correr el programa de configuración del InControl de nuevo después de la instalación de InTouch para instalar las extensiones de InControl. Para hacer esto, reinicie el programa de configuración. Haga clic en **Add/Remove** y entonces seleccione **InTouch Extensions**.

Si intenta usar InTouch e InControl sobre sistemas separados y necesita ver los símbolos de InControl desde el InTouch, instale los archivos para las extensiones sobre el sistema donde InTouch esta localizado. Corra el programa de configuración de InControl sobre el sistema de InTouch y seleccione **InTouch Extensions** en el cuadro de diálogo **Selects Components**. No es necesario instalar otros componentes.

Ejecución de Proyectos sobre Múltiples Plataformas

Si ha adquirido una versión de InControl que soporte múltiples plataformas de hardware de ejecución, entonces el programa le indicara que plataforma debe seleccionar.

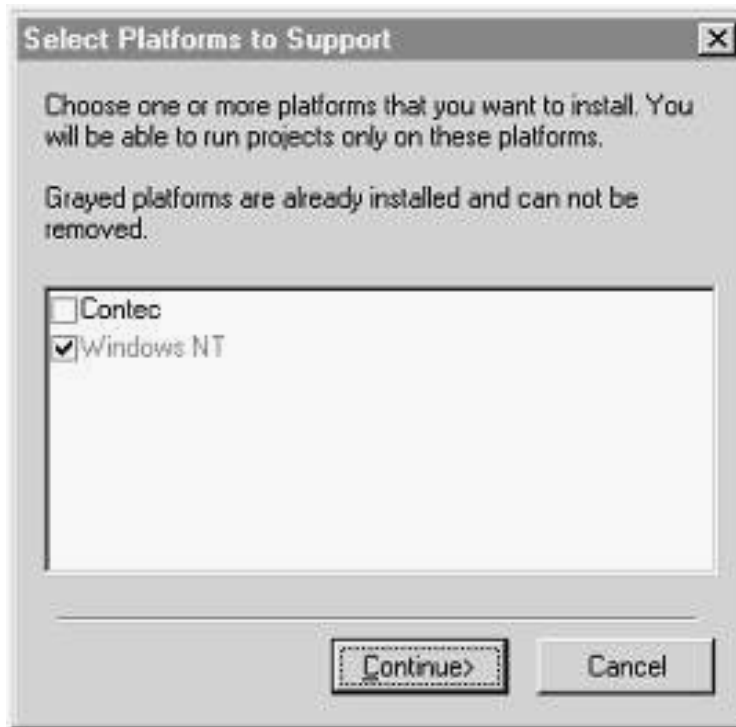


Figura 5. Selección de la plataforma

Antes de correr el programa de configuración, verifique la plataforma en la cual quiere correr el proyecto.

Recomendaciones Adicionales

Las Recomendaciones adicionales incluyen las siguientes:

- Antes de instalar el InControl, inicie su sistema Windows NT usando una cuenta con el administrador de privilegios.
- Cierre cualquier programa, particularmente las aplicaciones con Wonderware, que están corriendo actualmente.
- Desinstale cualquier versión previa de InControl.

Selección de las Opciones de Configuración

Después de correr el programa de instalación y que aparezca el cuadro de diálogo **Selects Components** del InControl, puede seleccionar los componentes que quiera instalar.

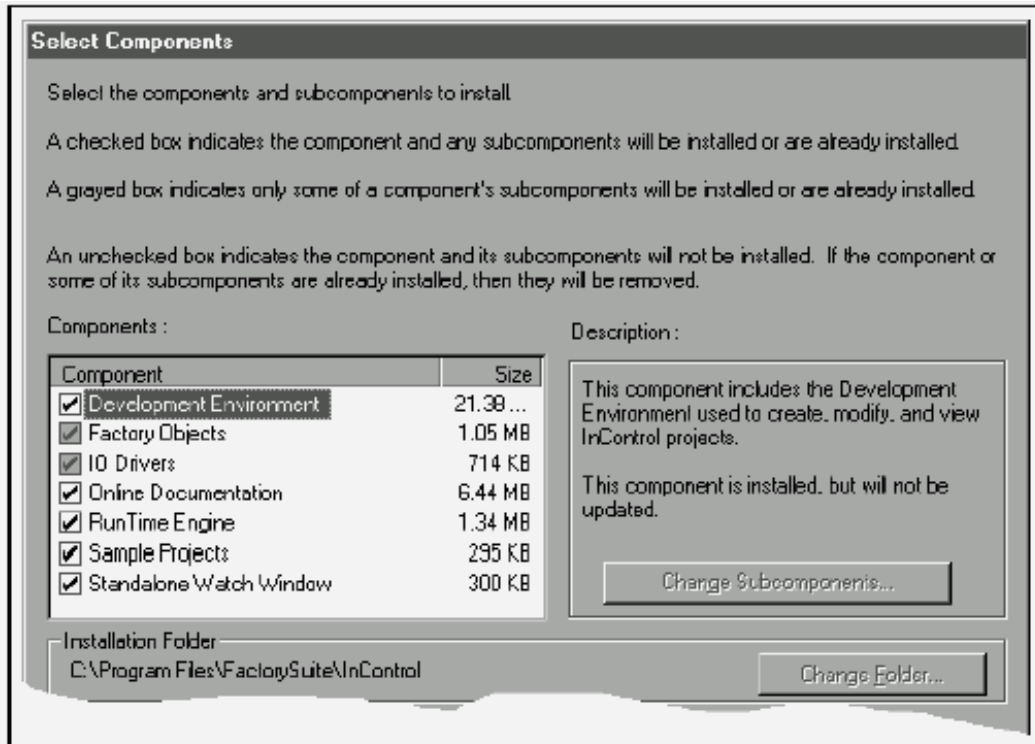


Figura 6. Selección de los componentes

El programa de configuración usa los siguientes indicadores:

- Indica que los componentes y cualquier subcomponente están instalados o serán instalados.
- Indica que algunos de los subcomponentes de los componentes están instalados o serán instalados.
- Indica que los componentes y sus subcomponentes no serán instalados o serán removidos.

Nota: Si un componente no está instalado, ninguno de sus subcomponentes serán instalados. Si uno de los componentes está siendo removido, todos sus subcomponentes serán removidos.

Escoja del listado siguiente los componentes que desee instalar. Puede escoger los subcomponentes de un componente, haciendo clic en el icono **Choose Subcomponents** cuando esta opción este disponible.

- **Development Environment** (Ambiente de Desarrollo)...Archivos de programa ejecutables necesarios para crear, modificar y ver los proyectos InControl.
- **Factory Objects** (Objetos de Fabrica).....FOE's Wonderware. Estos componentes tienen subcomponentes.
- **I/O Drivers**los drivers I/O. Incluyendo Suitelink. Estos componentes tienen subcomponentes.
- **InTouch Extensions** (Extensiones InTouch).....los asistentes (wizards) de InControl y las etiquetas de navegación para InTouch, usada para hojear las etiquetas de InControl. Sus componentes no aparecen como una opción a menos que InTouch ya esté instalado.
- **Online Documentation** (Documentación en Línea)...La guía de usuario en línea. Guías de usuario para drivers I/O individuales no son copiados si los subcomponentes de los drivers no son verificados.

El programa de instalación copia la **Guía de Usuario de InControl** y la **guía de usuario** para los drivers I/O en la subcarpeta **Books** en la carpeta **Wonderware Factory Suite**.

- **Runtime Engine** (software de ejecución)...Software de ejecución Windows NT, el cual permite correr proyectos InControl.
- **Sample Projects** (Ejemplos de Proyectos)...Algunos ejemplos de proyectos InControl. Este componente tiene subcomponentes.
- **Standalone Watch Window** (Ventana de Observación Autónoma)...Esta utilidad le permite monitorear variables durante el tiempo de ejecución sin tener que abrir el ambiente de desarrollo.

Cuando la instalación está completa usted tiene la oportunidad de leer las notas de introducción antes de correr el InControl. Debe instalar la licencia del InControl para tener acceso a todas las características. Para mas información sobre instalación de la licencia, ver la *Guía del Administrador de Sistemas FactorySuite*.

Adición de Componentes

Para adicionar o remover un componente InControl, se debe correr el programa de instalación.

➤ **Para adicionar o remover componentes:**

1. Corra el programa de instalación del InControl.
2. Haga clic en el icono **Add/Remove**. Aparecerá el cuadro diálogo **Select Components**.
3. Revise o quite el cuadro de comprobación para los componentes que se necesitan.
4. Haga clic en **OK/Continue**. El programa de instalación agregará o removerá los componentes, como se muestra en la siguiente figura, donde un **driver I/O** ha sido seleccionado para ser removido.

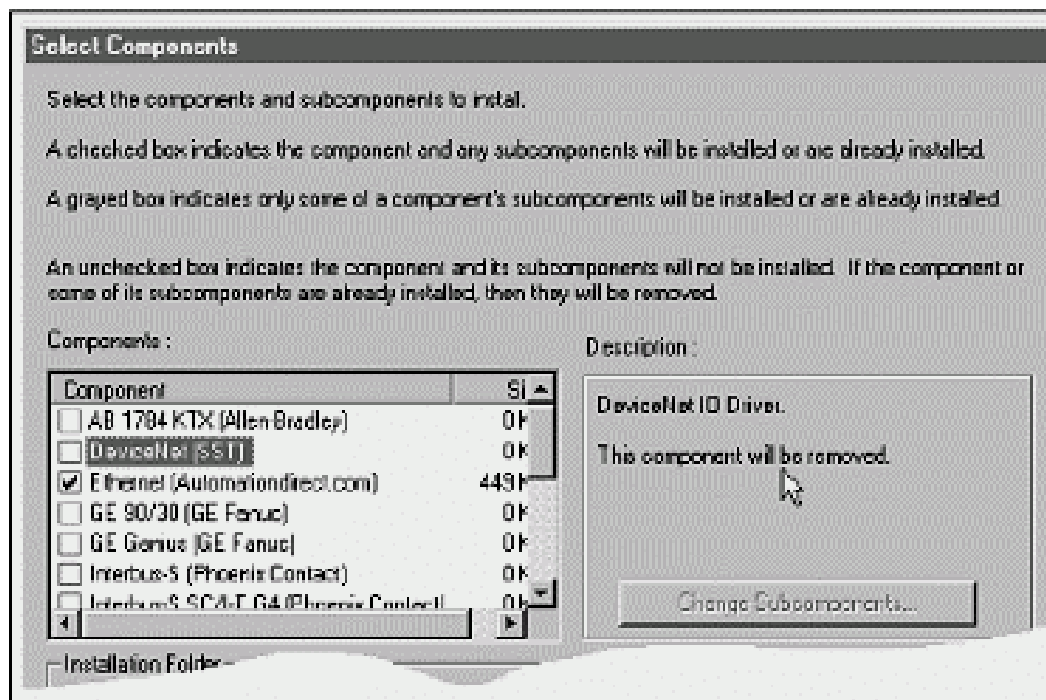


Figura 7. Selección de componentes

DESINSTALACIÓN DEL INCONTROL

Use la utilidad de desinstalación del InControl para remover todos sus archivos e iconos asociados y sus componentes de su sistema.

➤ **Para remover el InControl:**

1. Haga clic en **Start** sobre **Taskbar** (la barra de tareas). Luego señale **Programs\Wonderware Factorysuite**.
2. Señale **Uninstall** debajo de **InControl** y haga clic.

Es posible que no todos los archivos sean removidos. Usted puede revisar el registro de FactorySuite para ver una lista de los archivos que fueron removidos.

ACERCA DE LA GUÍA DE USUARIO

El programa de instalación copia la **Guía de Usuario de InControl** y la **Guía de usuario individual** para los drivers I/O en la subcarpeta **Books** debajo de la carpeta **Factorysuite**.

Contenido de la Guía de Usuario:

- **RLL** ver los capítulos “**Usando el editor RLL**” y “**Elementos de programa RLL**”.
- **SFC** ver los capítulos “**Usando el editor SFC**” y “**Elementos de programa SFC**”.
- **Texto Estructurado** ver los capítulos “**Usando el editor de Texto Estructurado**” y “**Lenguajes de Texto Estructurado**”.
- **Objeto de Fabrica InControl** ver los capítulos “**Usando el editor de Objetos de Fabrica InControl**”.

Contenido de la Guía de Usuario I/O: Para información específica sobre los drivers I/O tal como adición y configuración de módulos, componentes, etiquetas, etc., referirse a los manuales para drivers individuales.

EJECUCIÓN DEL INCONTROL: INICIACIÓN

➤ **Para iniciar el InControl:**

1. Haga clic en **Start** sobre la barra de tarea de Windows NT para mostrar el menú de inicio.
2. Señale en **Programs\Wonderware Factorysuite**.
3. Haga clic en **InControl**. Aparecerá el cuadro de diálogo **InControl Project Manager** (Administrador de Proyectos InControl).

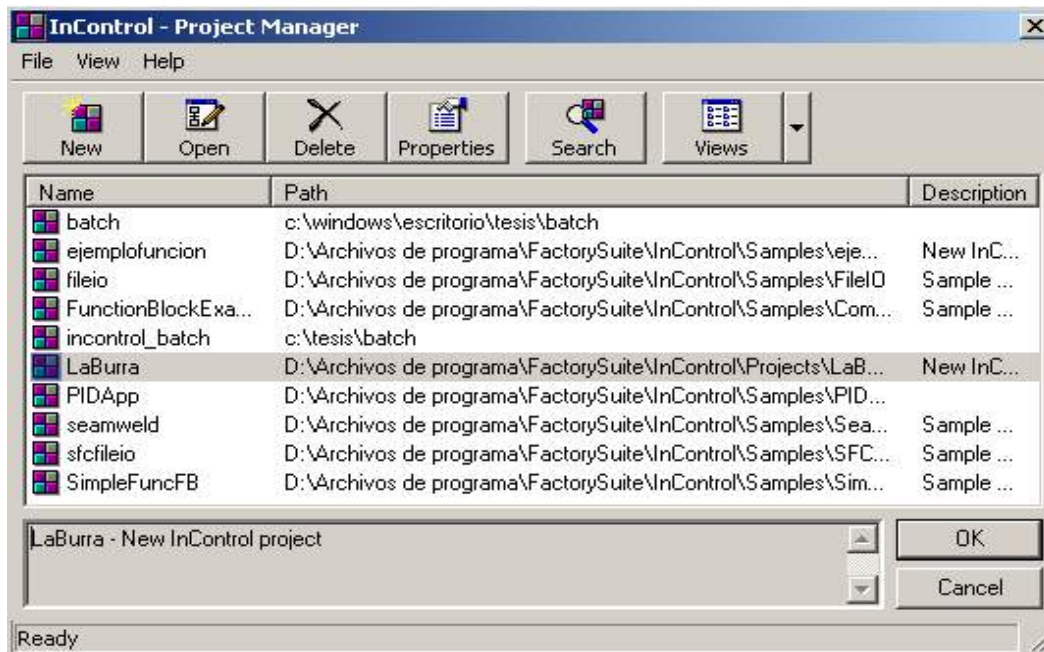


Figura 8. InControl Project Manager

Creación de un Proyecto

El InControl le permite crear grupos de programas, llamados proyectos. Diez proyectos aparecen en la figura de arriba, entre ellos: **Batch**, **EjemploFuncion** y **PIDApp**. Todos los programas dentro de un proyecto pueden ser ejecutados simultáneamente, y puede coordinarlos para manejar sus procesos. Antes de escribir un programa, se debe crear un proyecto.

➤ **Para crear un proyecto después de iniciar el InControl:**

1. En el menú **File** del cuadro de diálogo **InControl Project Manager**, haga clic en **New**. Aparecerá el cuadro de diálogo **Create InControl Project**.

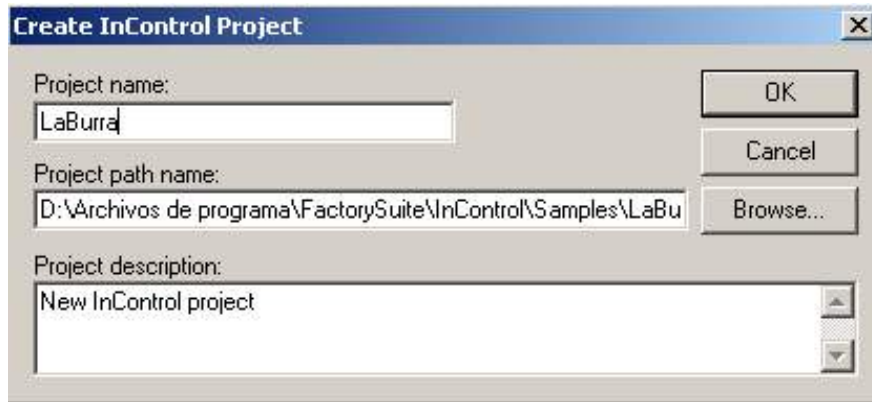


Figura 9. Creación de proyectos InControl

2. Ingrese el nombre de un proyecto. Seleccione una ruta y haga clic en **OK**. En la siguiente figura, el proyecto **LaBuma** ha sido creado.

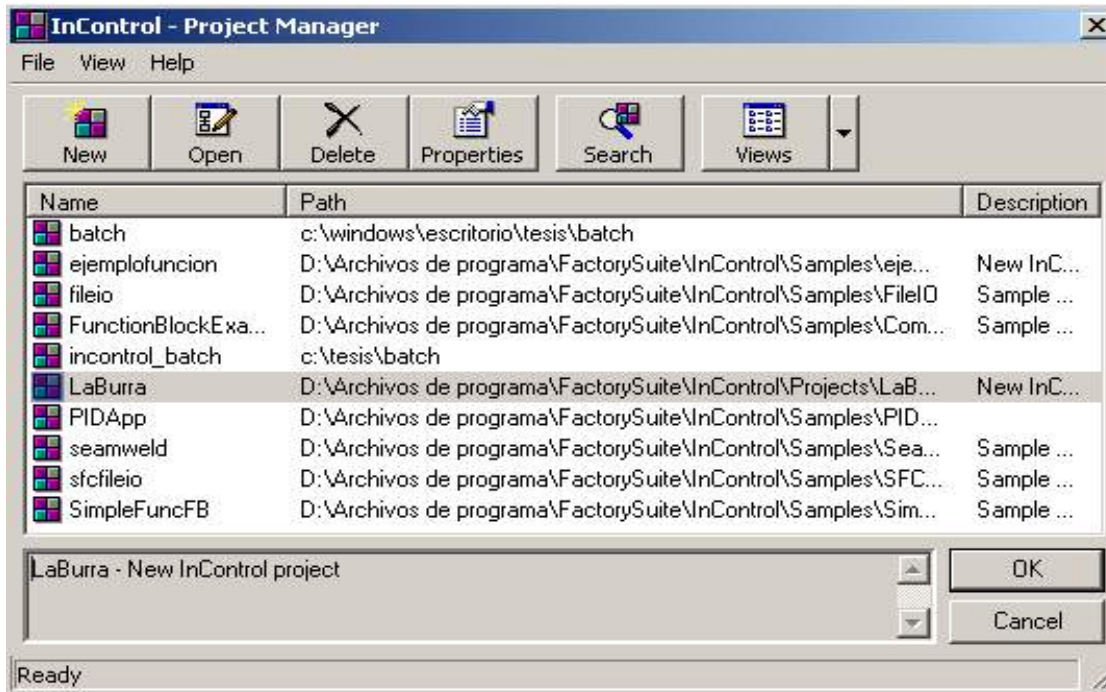


Figura 10. Presentación del proyecto ya creado

3. Haga doble clic sobre el nombre del proyecto para abrir el proyecto en el ambiente de desarrollo. Aparecerá el cuadro de diálogo **Runtime Engine Target**.

QUE SIGUE?

Aquí no hay un método predeterminado para hacer las tareas de configuración y programación en InControl. En general, El siguiente orden de tareas es recomendado para la mayoría de las aplicaciones.

Orden	Tareas	Capítulos de la Guía de Usuario InControl
1	Adición de Drivers I/O para la ventana I/O.	“Configuración I/O”
2	Definición de puntos I/O	Consulte la guía de usuario I/O en línea en el Directorio de Manuales InControl.
3	Definición de variables	“Definición de Variables”
4	Escritura de programas de aplicación	“Manejo y Organización de Proyectos ” “Relay Ladder Logic (RLL)” “Sequential Function Chart (SFC)” “Structured Text Language (STL)” “Uso del Editor de Objetos de Fabrica InControl”
5	Depuración de Programas de aplicación	“Ejecución de un Proyecto”
6	Ejecución y Examen de Programas de	“Ejecución de un Proyecto”
7	Aplicación	“Establecimiento de la Seguridad”
8	Configuración de Seguridad	
	Diseño de interfase HMI	“InControl e InTouch” además, consulte la Guía de Usuario InTouch.

AMBIENTE INCONTROL

Este capitulo describe el ambiente InControl: elementos de la barra de herramientas, menú de opciones y campos de pantalla, etc.

CONTENIDO

- VENTANAS DE DESARROLLO Y EJECUCIÓN
- BARRA DE HERRAMIENTAS
- BARRA DE MENÚ

This product is licensed to:

Company:	N/A
SN:	N/A
Expires:	N/A
Locked To:	N/A

7.1.0.1337

VENTANAS DE DESARROLLO Y EJECUCIÓN

El InControl consta de dos conjuntos de ventanas: la **Ventana de Desarrollo** donde se pueden crear programas de aplicación, y la **Ventana de Ejecución** donde se ejecutan y se encarga de monitorear los programas creados. Usted puede controlar varias ventanas ocultándolas o mostrándolas.

Ventana de Desarrollo

El esquema típico para la ventana de desarrollo es como se muestra en la siguiente figura.

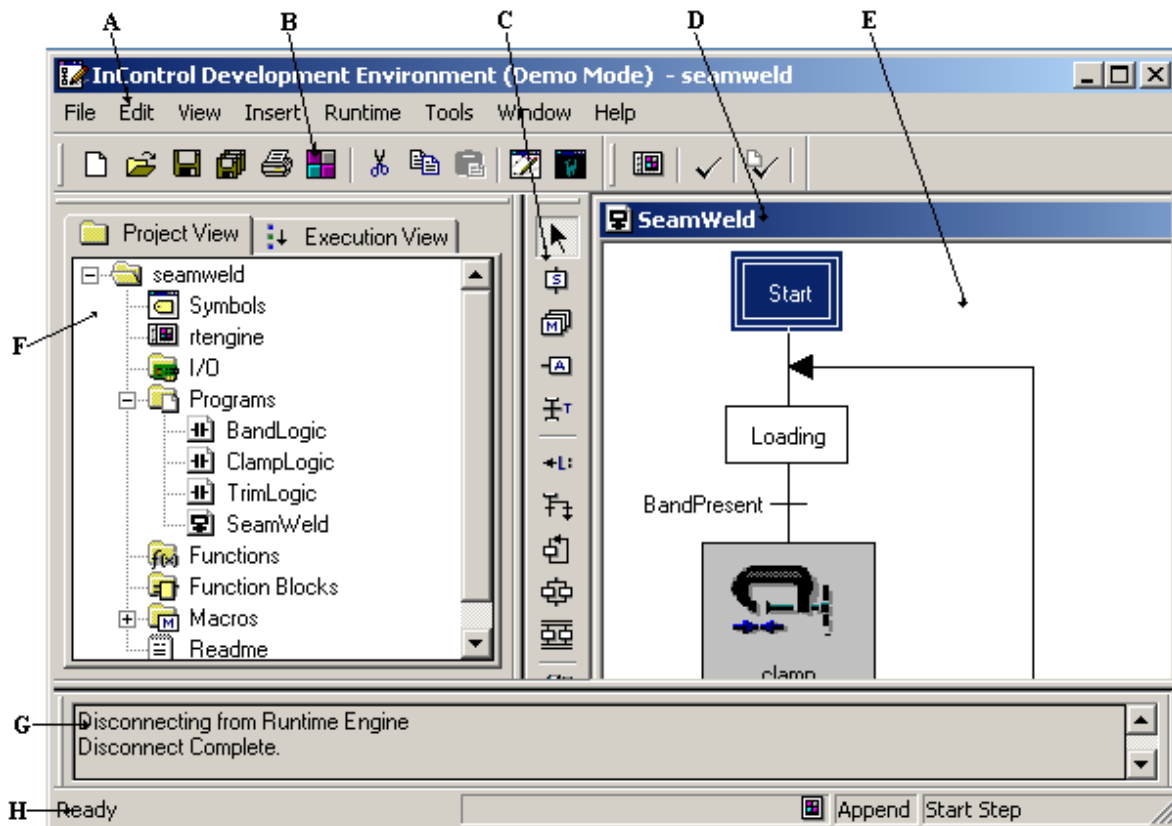


Figura 1. Ventana de desarrollo

Elementos de la ventana de desarrollo

Objetos	Elementos de la Pantalla	Función
A	Barra de Menú	Muestra las funciones estándar en formato texto.
B	Barra de herramientas Estándar	Muestra las funciones estándar como iconos.
C	Barra de Herramientas del Editor	Muestra las herramientas usadas para adicionar elementos de programa a un programa. Esta barra cambia, dependiendo del tipo de programa que esta siendo editado.
D	Barra de Título de la Ventana de Edición	Muestra el nombre del programa. Un asterisco en el nombre del programa indica que el programa ha sido modificado, pero no guardado, y/o es diferente de la copia que esta corriendo en el Runtime Engine. Si una copia del programa esta corriendo, su modo (Correr, Pausa, Detenido) también es mostrado. para funciones, bloques de funciones y objetos de fabrica InControl que son cargados en el Runtime Engine, el modo cargado es mostrado.
E	Ventana de Edición	Área de trabajo para la edición.
F	Ventana de Proyectos	Muestra las funciones específicas de un proyecto: programas, organización E/S y prioridad de ejecución.
G	Ventana de Salida	Muestra los mensajes (incluyendo mensajes de error), del Runtime Engine, el compilador y otros elementos de programas. Esa información también es escrita en el Wonderware Logger.
H	Barra de Estados	Muestra información del programa o de ciertos iconos.

Ventana de Ejecución

El esquema típico para la ventana de ejecución es como se muestra en la siguiente figura.

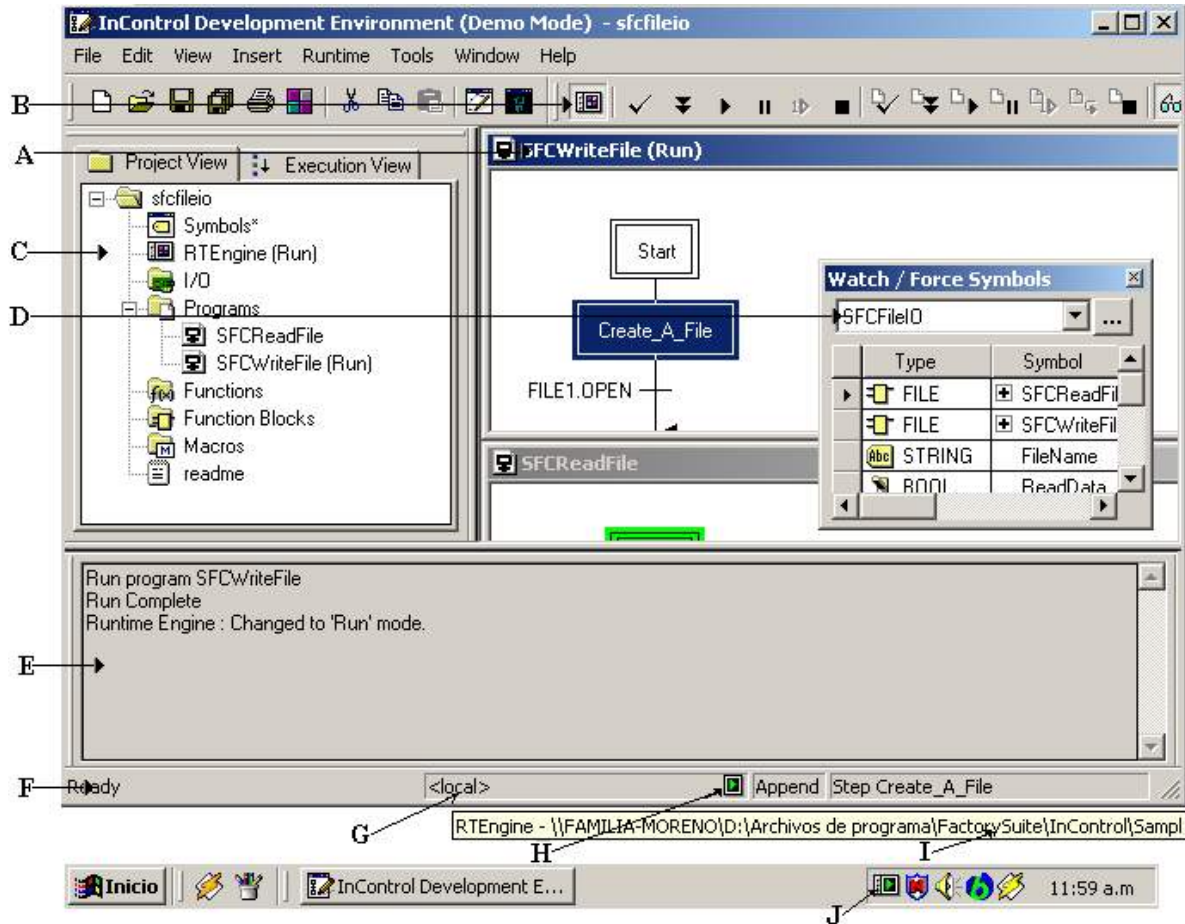








Figura 2. Ventana de ejecución

Elementos de la ventana de ejecución

Objetos	Elementos de la Pantalla	Función
A	Barra de Título del Programa	Muestra el nombre del programa. Si una copia del programa esta corriendo, su modo (Run, Pause, Stopped, etc) también mostrado. Un asterisco en el nombre del programa indica lo siguiente: - No esta conectado al ambiente de ejecución: el programa ha sido modificado, pero no guardado. - Conectado al ambiente de ejecución: el programa es diferente de la copia que esta corriendo en el Runtime Engine.
B	Barra de herramientas de Ejecución	Muestra las funciones de ejecución estándar como iconos.
C	Ventana de Proyectos	Muestra las funciones específicas de un proyecto: programas configuración E/S, organización E/S y prioridad de ejecución. Estas funciones son descritas en el capítulo “MANEJO Y ORGANIZACIÓN DE PROYECTOS” .
D	Ventana Watch (Observación)	Muestra las variables (símbolos) y su estado de ejecución. Usted puede usar independientemente la ventana Watch si no quiere abrir el ambiente de Desarrollo o Ejecución.
E	Ventana de Salida	Muestra los mensajes del Runtime Engine, el compilador, los elementos del programa, etc.
F	Barra de Estado	Muestra información del programa.
G	Nodo Conectado	Muestra el nodo al cual el ambiente de Desarrollo esta conectado.
H	Icono del RTE Conectado	Muestra el estado del Runtime Engine en el nodo local o remoto, al correr el proyecto.
I	Proyecto Cargado	Herramienta de información que reporta el nombre del proyecto cargado en el Runtime Engine.
J	Icono de Monitoreo del Runtime Engine	Monitorear el estado del Runtime Engine en el nodo local. Puede ser usado para controlar el modo del proyecto y ajustar el tiempo de exploración o análisis.

Iconos del Runtime Engine

Los siguientes iconos están asociados con el icono de monitoreo del **Runtime Engine** y también aparecen en la barra de Estado:

Icono	Estado
	Indica que el Runtime Engine esta en el modo Run .
	Indica que el Runtime Engine esta en el modo Pause .
	Indica que el Runtime Engine esta en el modo Stop .
	Indica que el Runtime Engine esta en el modo Fault . Note que este icono no indica que un programa esta en el modo Fault .
	Indica que un mensaje ha sido enviado a la ventana de salida y al Wonderware Logger . El icono también aparece cuando un programa ingresa al modo Fault o cuando se ejecutan las funciones RLL MSGW o las funciones de Texto Estructurado MSGWND .
	Indica que una o más variables están forzadas.

BARRA DE HERRAMIENTAS

Las funciones y comandos pueden ser seleccionados desde las barras de herramientas de la ventana de desarrollo de InControl. Para ser visualizadas deben estar seleccionados desde el comando **View** en la barra de menú. Estas barras de herramientas son: **Standar** y **Runtime**.

Barra de Herramientas Standard

Algunas de las funciones InControl pueden ser seleccionadas desde esta barra de herramientas **Standard**.

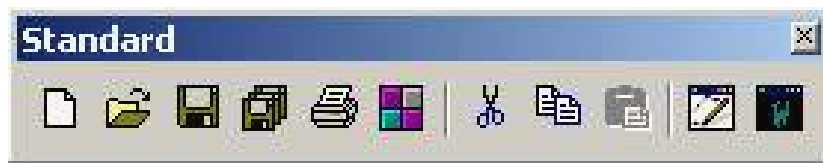













Figura 3. Barra de herramientas Standard

Icono	Opción en la barra de menú	Función
	New en el menú File .	Crea un nuevo programa.
	Open en el menú File .	Abre un programa existente.
	Save en el menú File .	Guarda el programa activo.
	Save All en el menú File .	Guarda todos los archivos abiertos.
	Print en el menú File .	Imprime un programa.
	Project en el menú File .	Abre el Project Manager.
	Cut en el menú Edit .	Corta el objeto seleccionado y lo ubica en el portapapeles.

Icono	Opción en la barra de menú	Función
	Copy en el menú Edit .	Copia el objeto seleccionado y lo ubica en el portapapel.
	Paste en el menú Edit .	Pega el contenido del portapapel.
	Windows Maker en el menú Tools .	Accede al InTouch Window Maker (Ventana de creación)
	Windows Viewer en el menú Tools .	Accede al InTouch Window Viewer (Ventana de visualización)

Barra de Herramientas Runtime

Los comandos de ejecución de InControl pueden ser seleccionados desde la barra de herramientas **Runtime** (Ejecución).

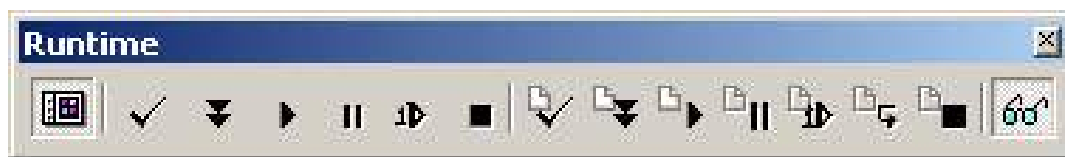









Figura 4. Barra de herramientas Runtime

Icono	Comando en el menú Runtime	Descripción
	Connect / Disconnect (Conecta / Desconecta)	Conecta el ambiente de desarrollo al Runtime Engine.
	Validate Project (Valida el Proyecto)	Valida todos los programas en un proyecto.
	Download Project (Descarga el Proyecto)	Descarga todos los programas de un proyecto al Runtime Engine.
	Run Project (Ejecuta el Proyecto)	Corre todos los programas en un proyecto.
	Pause (Pausa)	Interrumpe todos los programas que actualmente están corriendo por el Runtime Engine.
	Single Scan (Análisis Sencillo)	Ejecuta un análisis sencillo del Runtime Engine.
	Stop (Detención)	Detiene todos los programas de un proyecto que actualmente están corriendo por el Runtime Engine.

Icono	Comando en el menú Runtime	Descripción
	Validate Program (Valida el Programa)	Valida el programa seleccionado.
	Download Program (Descarga el Programa)	Descarga el programa al Runtime Engine.
	Run Program (Ejecuta el Programa)	Corre el programa seleccionado.
	Pause Program (Pausa el Programa)	Interrumpe un programa que actualmente esta corriendo por el Runtime Engine.
	Single Scan Program (Análisis Sencillo del Programa)	Ejecuta un análisis sencillo del programa.
	Step Program (Ejecución de un Paso en un Programa)	<p>RLL: ejecuta un escalón.</p> <p>Structured Text: ejecuta una línea de código.</p> <p>SFC: Ejecuta una línea de código de cada Step activo.</p> <p>FOE: No es soportado.</p>
	Stop Program (Detiene un Programa)	Detiene un programa y los descarga desde la memoria.
	View Menu: Watch/Force variables (Menú de observación: Observar y Forzar variables)	Muestra la ventana Watch. Usted puede usar la ventana Watch de forma individual si usted no quiere abrir el Ambiente de Desarrollo o Ejecución.

BARRA DE MENÚ

Usted puede escoger cualquiera de las funciones InControl desde la barra de menú, la cual se muestra en la siguiente figura.









File Edit View Insert Runtime Tools Window Help

Comandos Insert

Los comandos **Insert** le permiten colocar elementos de programa dentro de un programa desde la barra de **Menú**, en lugar de usar la barra de herramientas de programas. Para mas información acerca de elementos de programa, ver el capítulo apropiado para el tipo de programa.

Comandos File

Use estos comandos para operaciones con archivos, tales como abrir, cerrar, e imprimir archivos, y adicionar o remover archivos de programa del proyecto.

Comando del menú File	Icono en la barra de herramienta	Función
New		Crea un nuevo archivo de programa.
Open		Abre un archivo de programa existente.
Close	n/a	Cierra todas las ventanas asociadas con el archivo de programa activo.
Save		Guarda el archivo de programa activo.
Save As	n/a	Guarda el programa activo con un nombre diferente.
Save All		Guarda todos los archivos de programas abiertos y la información relacionada con el proyecto.
Project		Abre el Project Manager.
Add File to Project	n/a	Adiciona un archivo de programa al proyecto.
Remove File From Project	n/a	Remueve un archivo de programa desde el proyecto.
Print		Imprime un programa.
Print Xref ¹ (References cross)	n/a	Imprime de las variables de programa, la información de donde y cuan a menudo son usadas en el programa.
Print Setup	n/a	Le permite cambiar las opciones de impresión y la impresora.
Exit	n/a	Cierra el Ambiente de Desarrollo. Si no guarda los programas que están abiertos, usted le indicara que los guarde.

¹ Usted también puede imprimir referencias cruzadas desde el **Symbol Manager**. Para mas información, ver el capítulo “DEFINICIÓN DE VARIABLES”.

Comandos Edit

El menú **Edit** aparece solamente cuando un programa es siendo editado. El InControl le indica que un comando ha sido seleccionado por que se coloca una marca o señal en este.

Comandos del menú Edit	Programa específico	Función
Undo	Todos	Deshace la ultima acción.
Redo	RLL, SFC, STL	Rehace previamente una acción deshecha que involucra un elemento de programa.
Cut	Todos	Corta el objeto seleccionado y lo coloca en el portapapeles.
Copy	Todos	Copia el objeto seleccionado y lo coloca en el portapapeles.
Paste	Todos	Pega el contenido del portapapeles.
Delete	Todos	Borra el objeto seleccionado.
Select All	SFC-Stp STL	Selecciona todas las líneas de código.
Mark line	SFC-Stp STL	Se usa para seleccionar una o más líneas de código.
Edit Element	SFC, RLL	Abre el cuadro de dialogo para el elemento de programa seleccionado.
Step Properties	SFC	Abre el cuadro de dialogo para un paso.
Find	Todos	Encuentra el texto especificado.
Find Next	Todos	Encuentra el siguiente caso del texto especificado.
Replace	Todos	Reemplaza el texto especificado con un nuevo texto.
Go To	SFC-Stp RLL	Muestra la localización seleccionada en el programa.
Go To Coil	RLL	Muestra el caso siguiente de la bobina seleccionada.
Boolean Transition	SFC-Stp	Ajusta por defecto la transición para todos los programas SFC abiertos, al tipo Boolean , en lugar del tipo RLL . Este comando no cambia transiciones existentes.
Lock Algorithms	SFC	Se usa para agregar la clave de protección para los algoritmos. Cuando un algoritmo esta bloqueado, este comando es " Unlock Algorithms ".
Set Bookmark	SFC-Stp,STL	Marca una o más localizaciones en un programa y usa el comando Go To para saltar entre estos.
Properties	FOE	Muestra el cuadro de dialogo de configuración.

RLL = Lógica en escalera por relevos **FOE** = Objeto de fabrica InControl

STL = Lenguaje de texto Estructurado **SFC-Stp** = Paso SFC

SFC = Cartas de funciones secuenciales

Comandos View

El menú **View** muestra en la pantalla elementos que usted puede mostrar o esconder. Los elementos cambian dependiendo del tipo de programa que este en la actual ventana activa. El InControl le indica que un elemento ha sido seleccionado por colocar una marca o señal en este.

Comando del menú View	Programa específico	Función
Toolbar	Todos	Muestra las funciones de programas como iconos. La barra de herramientas estándar es mostrada en “Ventana de Desarrollo” (Pagina 2-2).
Runtime Toolbar	Todos	Muestra las funciones de ejecución como iconos. La barra de herramienta aparece automáticamente cuando se conecta el Runtime Engine. La barra de herramientas de ejecución es mostrada en “La Barra de Herramientas Runtime” (pagina 2-7).
Status Bar	Todos	Muestra el modo actual del Runtime Engine (Corriendo, Detenido, Suspendido), el modo de edición (insertar o añadir), la actual localización del cursor en el programa que actualmente esta siendo editado, y la información de ayuda cuando el cursor esta encima de un botón. La barra de estado es mostrada en “Ventana de Ejecución” (pagina 2-4)
Contact/Coil Bar	RLL	Muestra los elementos de programa como iconos. La barra de Contactos/Bobinas es descrita en el capítulo “USO DEL EDITOR RLL” .
SFC Bar	SFC	Muestra los elementos de programa como iconos. La barra SFC es descrita en el capítulo “USO DEL EDITOR SFC” .
Structured Text Bar	SFC-Stp STL	Muestra los elementos de programa como iconos. La barra de Texto Estructurado es descrita en el capítulo “USO DEL EDITOR DE TEXTO ESTRUCTURADO” .
Factory Object Bar	FOE	Muestra el cuadro de dialogo de configuración. La barra de Objetos de Fabrica es descrita en el capítulo “USO DEL EDITOR DE OBJETOS DE FABRICA INCONTROL” .
Function Block Palette	RLL	Muestra los bloques de funciones que pueden ser usados en el programa.
Project	Todos	Muestra el área de trabajo para las funciones en la ventana de Proyectos. Las funciones de la ventana de Proyectos son descritas en el capítulo “MANEJO Y ORGANIZACIÓN DE PROYECTOS” .

Comandos del menú View	Programa específico	Función
Symbol Addresses	RLL SFC	Muestra las direcciones de los puntos I/O en los programas RLL y acciones SFC .
Watch/Force Variables	Todos	Muestra las variables seleccionadas y sus actuales valores de ejecución. Se usa para especificar nuevos valores para las variables para ayudar a la depuración del programa.
Logger	Todos	Muestra el Wonderware Logger , el cual mantiene un registro de los mensajes de ejecución. Estos mensajes también aparecen en la ventana de salida (Output).
Program Comments	SFC RLL	Muestra u oculta cualquier comentario que usted ingresa dentro del programa. Esta disponible solamente cuando un programa esta abierto para su edición. Para programas RLL usted debe escoger esta opción antes de ingresar un nuevo comentario.
Function Block Details	RLL	Muestra las variables de los bloques de funciones RLL . Al correr el programa, se actualizan las Variables.
All Steps	SFC	Muestra los Pasos para presentar sus nombres, descripciones, iconos asociados, o sus códigos de programas.
Runtime Highlighting	SFC RLL STL	Habilita la ejecución de una animación para un programa. Ajusta la frecuencia de actualización para la animación en el cuadro de dialogo Runtime Engine Properties .
Zoom	RLL SFC	Zoom hacia dentro o zoom hacia fuera del programa.

RLL = Lógica en escalera por relevos **FOE = Objeto de fabrica InControl**

STL = Lenguaje de texto Estructurado **SFC-Stp = Paso SFC**











SFC = Cartas de funciones secuenciales


Comandos Runtime

Los comandos **Runtime** le permiten a usted ajustar los parámetros de ejecución, observe el dato de estado, y enviar comandos directamente al **Runtime Engine**.

Ver el capítulo “EJECUCIÓN DE UN PROYECTO” para mas información.

Comando del menú Runtime	Icono de la Barra de Herramienta	Descripción
Connect/ Disconnect		<p>Conecta el ambiente de desarrollo al Runtime Engine. El Engine corre continuamente como un servicio Windows NT, y si actualmente ejecuta un proyecto, depende de su modo de operación (Run, Stop, Pause, etc.).</p> <p>Cuando el Runtime Engine esta conectado, el icono se mantiene presionado y la nueva opción es “Desconectar”, la cual desconecta el ambiente de Desarrollo del Runtime Engine. Si usted cierra el ambiente de Desarrollo el Runtime Engine continua corriendo.</p>
Configure	n/a	<p>Muestra el cuadro de dialogo Offline Runtime Engine Properties si no esta conectado al Runtime Engine.</p> <p>Muestra el cuadro de dialogo Online Runtime Engine Properties si esta conectado al Runtime Engine.</p>
Report Faults	n/a	Muestra los programas fallidos y las faltas I/O, en la ventana de Salida y en el Wonderware Logger .
Clear Faults	n/a	Coloca los programas fallidos en el modo Pause , borra las faltas, y borra los bits de estado de error, asi como RTEngine.ScanOverrun .
Validate Project		Valida todos los programas en un proyecto. Todos los programas modificados son guardados en el disco duro.
Download Project ¹		Descarga todos los programas de un proyecto al Runtime Engine. Los programas modificados son guardados en el disco duro. Los programas son validados si es necesario.

Comando en el menú Runtime	Icono de la Barra de Herramienta	Descripción
Run Project ¹		Corre todos los programas de un proyecto. Los programas son validados y descargados si es necesario. Todos los programas modificados son guardados en el disco duro.
Pause		Interrumpe todos los programas que actualmente están corriendo por el Runtime Engine.
Single Scan		Ejecuta un Scan (análisis) sencillo del Runtime Engine. El I/O es actualizado, entonces todos los programas en un proyecto que actualmente son descargados al Runtime Engine ejecutan un análisis. <i>Ver el capítulo "EJECUCIÓN DE UN PROYECTO" para mas información.</i>
Stop		Detiene todos los programas en un proyecto que actualmente están siendo ejecutados. Los programas no son cargados desde la memoria. El I/O va al estado definido en la configuración para cada tarjeta I/O.
Validate Program		Valida el programa seleccionado. Si el programa fue modificado, es guardado en el disco duro.
Download Program		Descarga el programa seleccionado al Runtime Engine. Si el programa fue modificado, es guardado en el disco duro.
Run Program		Corre el programa seleccionado. Si el programa fue modificado, es guardado en el disco duro.
Pause Program		Interrumpe un programa que este siendo ejecutado. El I/O continua su actualización.
Single Scan Program		Ejecuta un análisis sencillo del programa. Solamente puede hacerse mientras el programa este detenido.
Stop Program		Detiene y descarga un programa desde la memoria. Otros programas en el proyecto y los puntos I/O no son afectados.

Comando en el menú Runtime	Icono de la Barra de Herramienta	Descripción
Step Program		<p>RLL: ejecuta un escalón.</p> <p>Structure Text: ejecuta una línea de código.</p> <p>SFC: Ejecuta una línea de código de cada Paso activo. Todas las Acciones activas son ejecutadas completamente.</p> <p>FOE: No es soportado.</p> <p>RLL, Structured Text, y programas SFC deben ser compilados con la opción Debug Enabled.</p>
Toggle Breakpoint	n/a	<p>Pausa el flujo del programa en la línea seleccionada en el código del programa de Texto Estructurado. La línea no es ejecutada hasta que usted ejecuta o reanude el programa.</p>
Clear All Breakpoint	n/a	<p>Remueve todos los puntos de detención o corte situados en el programa de Texto Estructurado. No esta disponible a menos que el programa de Texto Estructurado este abierto.</p>

Comandos Tools

Use los comandos **Tools** para llamar a otros programas o utilidades para su ejecución.

Comandos del menú Tools	Programa específico	Función
Symbol Manager	Todos	Acceso al Symbol Manager; se usa para crear y editar variables.
Action Manager	SFC	Acceso al Action Manager; se usa para renombrar y borrar acciones SFC.
RLL Transition Manager	SFC	Acceso al Transition Manager; se usa para renombrar y borrar Transiciones SFC a RLL.
Step Library	SFC	Acceso a la Step Library (Librería de pasos); se usa para crear, editar y borrar pasos predefinidos en la librería.
InTouch	Todos	Muestra el menú de apertura para InTouch.
WindowMaker	Todos	Acceso a la InTouch WindowMaker (Ventana de creación).
WindowViewer	Todos	Acceso a la InTouch WindowViewer (Ventana de observación).
Security	Todos	Acceso al Administrador de la Seguridad; se usa para configurar la seguridad del sistema.
Configure Colors	SFC, STL, RLL	Acceso al cuadro de dialogo de selección de color; se usa para especificar los colores para la edición y ejecución de los programas.
Configure RLL/SFC Font Configure Text Editor Font	RLL, SFC, STL	Acceso al cuadro de dialogo de selección del tipo de letra; se usa para especificar el tipo de letra usada por los programas en su edición y ejecución.

Comandos Window

Use los comandos **Windows** para organizar las ventanas. El menú **Window** no aparece a menos que usted abra una ventana de edición.

Comandos del menú Window	Función
Cascade	Organiza las ventanas de tal modo que queden superpuestas.
Tile	Organiza las ventanas en recuadros no superpuestos.
Arrange Icons	Mueve los iconos a la parte inferior de la Ventana activa.
Close All	Cierra todas las ventanas de edición. No salva los programas guardados.

Comandos Help

Use los comandos **Help** para mostrar información del sistema y las opciones de ayuda.

Comandos del menú Help	Función
InControl Help Topics	Muestra la tabla de contenido para el sistema de ayuda.
SFC Editor Help	Muestra la ayuda para el editor SFC.
RLL Editor Help	Muestra la ayuda para el editor RLL.
Structured Text Editor Help	Muestra la ayuda para el editor de Texto Estructurado.
Wonderware Via Internet	Para sistemas con accesos a la Internet, conectarse al pagina Web de Wonderware.
About InControl	Muestra el numero del serial y la actual versión de InControl.
About SFC Editor	Cuando una ventana SFC esta activa, muestra la versión actual del editor SFC.
About RLL Editor	Cuando una ventana RLL esta activa, muestra la versión actual del editor RLL.
About Structured Text Editor	Cuando una ventana de Texto Estructurado esta activa, muestra la versión actual del editor de Texto Estructurado.
About Factory Object Editor	Cuando una ventana FOE esta activa, muestra la versión actual del editor FOE.

ESTABLECIMIENTO DE LA SEGURIDAD

Este capítulo describe como asignar y remover la contraseña para el ambiente de desarrollo y de ejecución(runtime) InControl.

CONTENIDO

- INTRODUCCIÓN
- REGISTRO DE UN USUARIO
- MANEJO DE LA SEGURIDAD
- BLOQUEO DE ALGORITMOS SFC
- LA SEGURIDAD DE WINDOWS NT

This product is licensed to:

Company:	N/A
SN:	N/A
Expires:	N/A
Locked To:	N/A

7.1.0.1337

INTRODUCCIÓN

El acceso al ambiente InControl se restringe para asegurar que solo personal autorizado y/o calificado puede interactuar con los procesos de la fabrica. Los tres niveles de seguridad disponibles para el acceso son: Administrador, Ingeniero, y operador. La siguiente tabla muestra los niveles de seguridad requeridos para hacer las tareas mostradas. Una "Y" indica el nivel de seguridad que es requerido para hacer la tarea.

Tareas	Nivel de seguridad		
	Administrador	Ingeniero	Operador
Editar Programas	Y	Y	
Editar Configuración I/O	Y	Y	
Editar Símbolos	Y	Y	
Correr y Detener programas	Y	Y	Y
Cargar Programas	Y	Y	
Cambiar Clave (Passwords)	Y		
Agregar y Borrar nombres de Usuarios	Y		
Modificar y Establecer Valores	Y	Y	

El InControl es instalado con un nombre de usuario predefinido: Administrador.

La contraseña prefijada es una cadena de caracteres vacía. El administrador del sistema, con el administrador del nivel de seguridad, asigna todos los usuarios y contraseñas para el sistema. Es muy recomendable que la clave sea cambiada por el cliente después de que se ha instalado el software InControl.

Advertencia. Usuarios quienes tengan acceso ilimitado a todas las tareas de configuración de InControl puede ser capaces de producir cambios al sistema impropios y desautorizados. Esto podría causar operaciones imprevisibles para el controlador, lo cual puede resultar en muerte o lesiones al personal y/o averías al equipo. Es seguro que la clave predefinida sea cambiada por el administrador de nivel de seguridad y aconséjeles a todos los usuarios que protejan sus contraseñas asignadas.

La seguridad del InControl está diseñada para trabajar con las características de seguridad que son parte del sistema operativo Windows NT. Su administrador de sistema puede ayudar a garantizar la integridad de todo el sistema configurando la identificación del usuario para que solo usuarios autorizados puedan tener acceso a los proyectos InControl y al programa de ejecución (runtime engine). Observe la documentación por el sistema operativo Windows NT para información acerca de cómo configurar estos niveles de seguridad.

Para comprobar el nivel de seguridad del usuario que esta actualmente registrado, pulse sobre **Security** en el menú **Tools**. El usuario y nivel de seguridad aparece en el menú, como se muestra la siguiente figura.

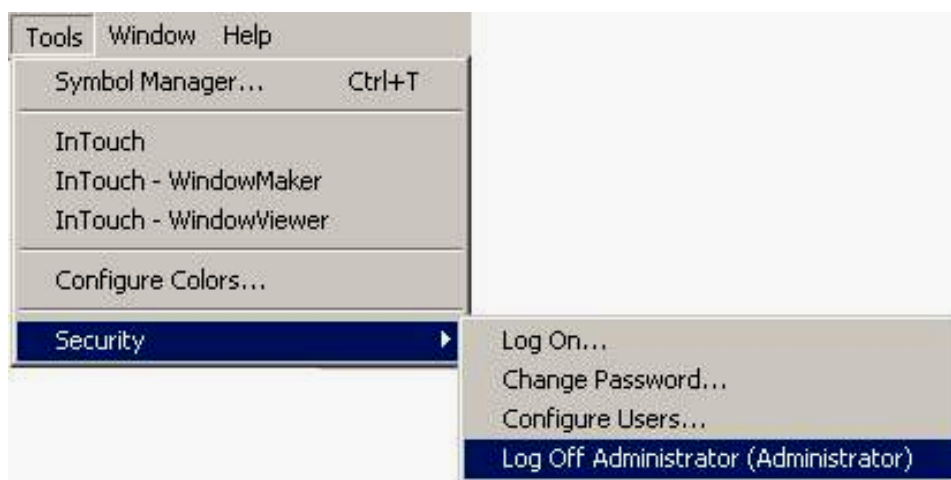


Figura 1. Nivel de seguridad

REGISTRO DE UN USUARIO

Esta sección describe las tareas de seguridad que son típicamente efectuadas por cualquier usuario, incluyendo un operador.

Registro de un nuevo usuario

➤ **Para Registrarlo:**

1. Sobre el menú **Tools**, señale la etiqueta **Security** y haga click sobre **Log On**. Aparecerá el cuadro de dialogo **Log On**.



Figura 2. Cuadro de registro

2. Escriba su nombre y clave(password) y haga clic en **Ok**. Si se registra un nuevo usuario, el anterior usuario es automáticamente eliminado o borrado.

Eliminación del registro a un usuario

➤ **Para borrarlo:**

1. En el menú **Tools**, señale la etiqueta **Security**.
2. Haga Click en **log Off xxx**. Si otro usuario es registrado, esta opción muestra el nombre del usuario actual en el campo xxx. Si el usuario no esta registrado, la opción **Log Off** no estará disponible.

Cambio de la contraseña

Un usuario puede cambiar la clave usada por el nombre de usuario registrado inicialmente.

- **Para cambiar la clave por un nombre de usuario:**
 1. Registre el nombre de usuario por el cual usted quiere cambiar la clave(password).
 2. Sobre el menú **Tool**, señale la etiqueta **Security** y haga click en **Change Password**. Aparecerá el cuadro de dialogo **Change Password**.
 3. Ingrese la clave actual.
 4. Ingrese la nueva clave y entonces vuelva a ingresarla para verificar.
 5. Pulse **OK** para cerrar el cuadro de dialogo.

MANEJO DE LA SEGURIDAD

Esta sección describe las tareas que son típicamente efectuadas por el administrador del sistema. Inmediatamente después de instalar el InControl y correr el programa, usted es el administrador del acceso al nivel de seguridad. Usted puede empezar haciendo tareas normales, como configuración, escribir y editar programas, sin registrarse. El sistema permanece en este estado hasta que usted este libre de registro o elabore cambios en la seguridad. Después de esto, usted debe registrarse antes de que pueda hacer cualquier cambio.

Adición de un nombre de usuario

Para hacer cambios en la configuración de seguridad, usted debe tener acceso a las tareas de seguridad de cambios de clave.

➤ **Para adicionar un nombre de usuario:**

1. Regístrese como administrador.
2. Sobre el menú **Tools**, señale la etiqueta **Security**, y haga Click en **Configure Users**. Aparecerá el cuadro de dialogo **Configure Users**.



Figura 3. Cuadro de configuración del usuario

3. Ingrese el nuevo nombre de usuario en el campo **User Name**.
4. Ingrese la clave en el campo **Password**.
5. Seleccione el nivel de acceso en el campo **Access**.(Administrador, Ingeniero u Operador.
6. Haga clic en **Require logon at startup** para pedir al usuario que se registre cuando inicie la sesión con InControl. Note que esta opción afecta todos los usuarios, así como al usuario que se adiciona.
7. Haga clic en **Add**. El nuevo nombre de usuario es adicionado a la lista nivel de seguridad.

Cambio de la clave y eliminación de un nombre de usuario

Para hacer cambios en la configuración de seguridad, usted debe tener acceso a los cambios de clave de las tareas de seguridad.

➤ **Para cambiar claves o niveles de seguridad de acceso de los usuarios:**

1. Regístrese como administrador.
2. Sobre el menú **Tools**, señale la etiqueta **Security** y haga clic en **Configure Users**. Aparecerá el cuadro de dialogo **Configure Users**.
3. Pulse sobre el nombre de usuario que esta cambiando.
4. Si usted está cambiando la clave, ingrese la nueva clave en el campo **Password**.
5. Si usted está cambiando el nivel de acceso de seguridad, seleccione el nuevo nivel de acceso en al campo **Access**.
6. Pulse **Update**. La nueva información será adicionada a la lista nivel de seguridad de acceso.
7. Pulse **Ok** para cerrar el cuadro de dialogo.

➤ **Para eliminar el nombre de usuario:**

1. Regístrese como administrador.
2. Sobre el menú **Tools**, señale la etiqueta **Security** y haga clic en **Configure Users**. Aparecerá el cuadro de dialogo **Configure Users**.
3. Pulse sobre el nombre de usuario que usted está eliminando.
4. Pulse **Delete**. El nombre de usuario será removido de la lista nivel de seguridad de acceso.
5. Pulse **Ok** para cerrar el cuadro de dialogo.

BLOQUEO DE ALGORITMOS SFC

InControl permite proteger el código del programa dentro de una SFC, de cambios no autorizados. Seleccione el comando **Lock Algorithms** en el menú **Edit** y asigne una clave. Para bloquear el código **SFC**, usted debe tener acceso al editor de Programas.

➤ **Para bloquear los algoritmos SFC:**

1. Abra el programa **SFC**.
2. Sobre el menú **Edit**, pulse en **Lock Algorithms**. Aparecerá el cuadro de dialogo **Enter Lock Password**.
3. Ingrese la clave en el campo **Password**. Use una clave que sea diferente de la contraseña de seguridad.
4. Confirme la nueva clave y pulse **OK**.
5. Para guardar y bloquear el archivo activo, pulse **YES**.

Bloqueando los algoritmos se impide que usuarios desautorizados cambien el código del Texto Estructurado dentro de los Pasos del programa. Esto no les impide editar otros elementos en el programa, por ejemplo, Transiciones, Acciones, lazos, etc., y no les impide a los usuarios desautorizados de borrar los pasos.

Advertencia: El código de los pasos SFC que han sido bloqueados no pueden cambiarse sin la contraseña. Si olvida la clave, usted no podrá editar el código para ningún Paso en el programa. Guarde una copia de la clave en un lugar seguro.

➤ **Para desbloquear los algoritmos SFC:**

1. Abra el programa SFC.
2. Sobre el menú **Edit**, pulse en **Unlock Algorithms**, aparecerá el cuadro de diálogo **Validate Unlock Password**.
3. Ingrese la clave en el campo **Password** y pulse en **OK**.

Desbloquear los algoritmos permite a todos los usuarios con acceso al Editor de Programas editar el Texto Estructurado dentro de los pasos del programa.

LA SEGURIDAD DE WINDOWS NT

Usted puede restringir el acceso a la unidad de hardware pidiendo a todos los operadores y otros usuarios registrarse en el sistema operativo Windows NT. Su administrador del sistema Windows NT puede configurar la identificación del usuario para que solo clientes autorizados puedan correr programas y acceder a directorios seleccionados. Con la configuración de seguridad apropiada, el menú de la pantalla del Software de Ejecución, que permite el acceso al Software de Ejecución, no es visualizado y puede ser accesado sin la clave apropiada.

Vea la documentación del sistema Windows NT para información acerca de cómo configurar estos niveles de seguridad.

CONFIGURACIÓN I/O

Este capítulo describe un método general para configurar tarjetas de pruebas I/O (Entrada / Salida). Para información mas detallada, consulte los documentos sobre Configuración I/O y la documentación de usuario que acompaña la tarjeta I/O.

CONTENIDO

- INTRODUCCIÓN
- ADICIÓN DE DRIVERS
- VERIFICACIÓN DE LA VERSIÓN DEL DRIVER
- CONFIGURACIÓN I/O
- SIMULACIÓN I/O

This product is licensed to:

Company:	N/A
SN:	N/A
Expires:	N/A
Locked To:	N/A

7.1.0.1337

INTRODUCCIÓN

Típicamente, una tarjeta debe ser instalada en su sistema InControl para comunicarse con los módulos I/O (Entrada / salida) de la planta. La siguiente figura muestra una tercera tarjeta de análisis o prueba I/O instalada en la unidad de hardware InControl para examinar los módulos I/O.

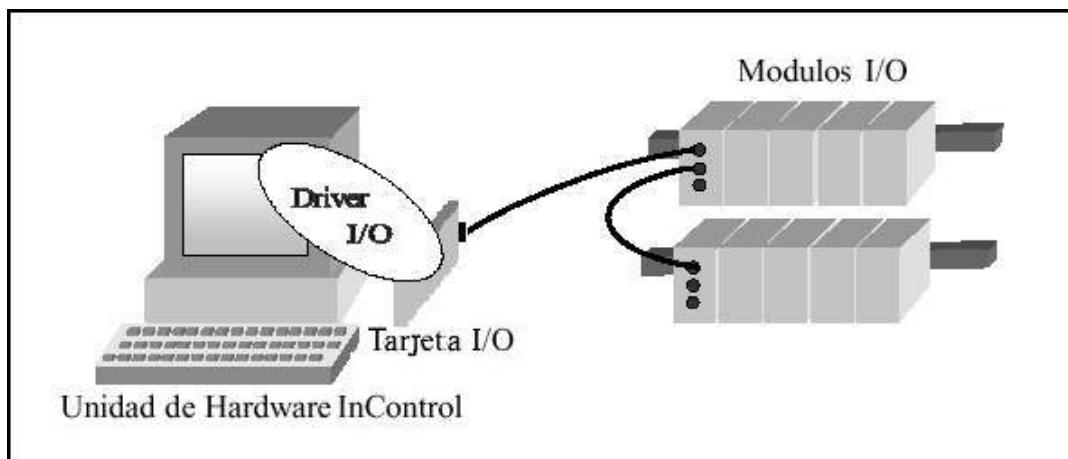


Figura 1. Sistema InControl

Dependiendo del modelo de la tarjeta de prueba, debe ser capaz para instalar tarjetas múltiples del mismo modelo en la unidad de hardware del InControl, como se muestra en la siguiente figura.

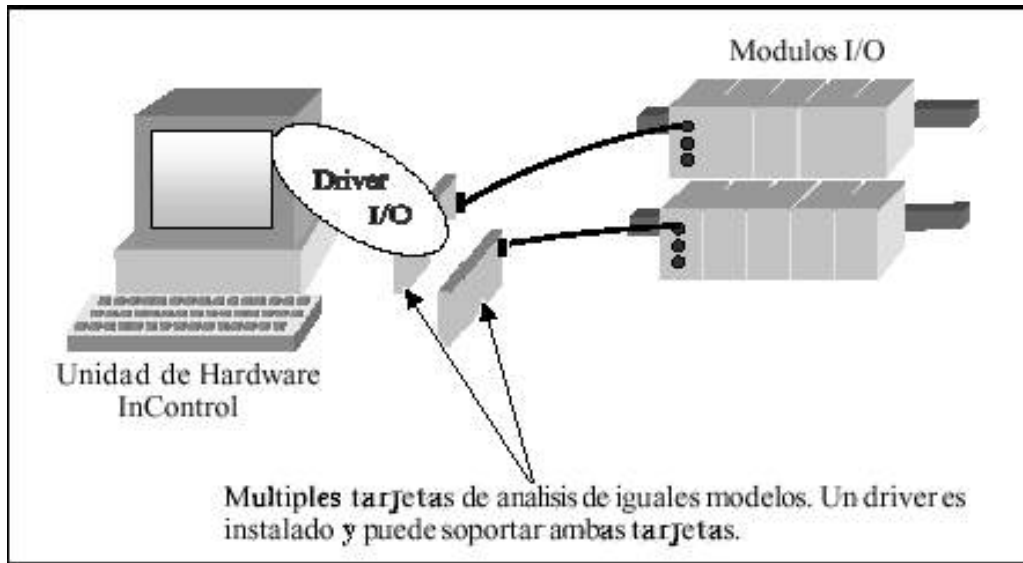


Figura 2. Sistema InControl con múltiples tarjetas I/O

Para iguales modelos de tarjetas, usted puede ser capaz de instalar tarjetas de diferentes fabricantes en la unidad de hardware InControl, como lo muestra la siguiente figura.

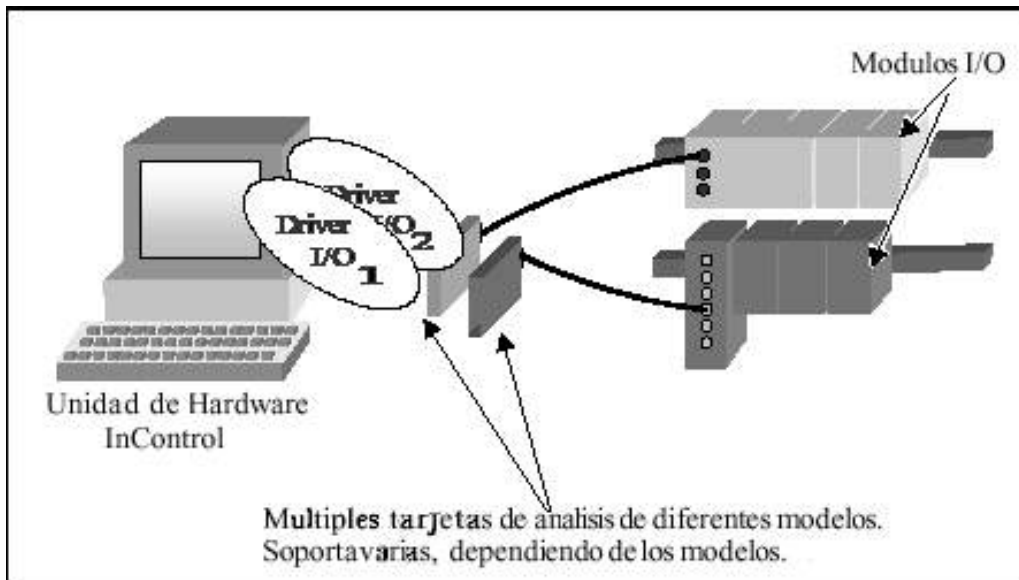


Figura 3. Sistema InControl con diferentes tarjetas I/O

Nota: Verifique con el fabricante para determinar si su tarjeta de prueba puede operar en igual unidad de hardware con otra tarjeta de análisis.

Así mismo los Drivers I/O no requieren una tarjeta de prueba específica para ser instalado en la unidad de hardware InControl. La siguiente figura ilustra como un driver I/O puede comunicarse con un módulo inteligente I/O directamente a través de la tarjeta serial.

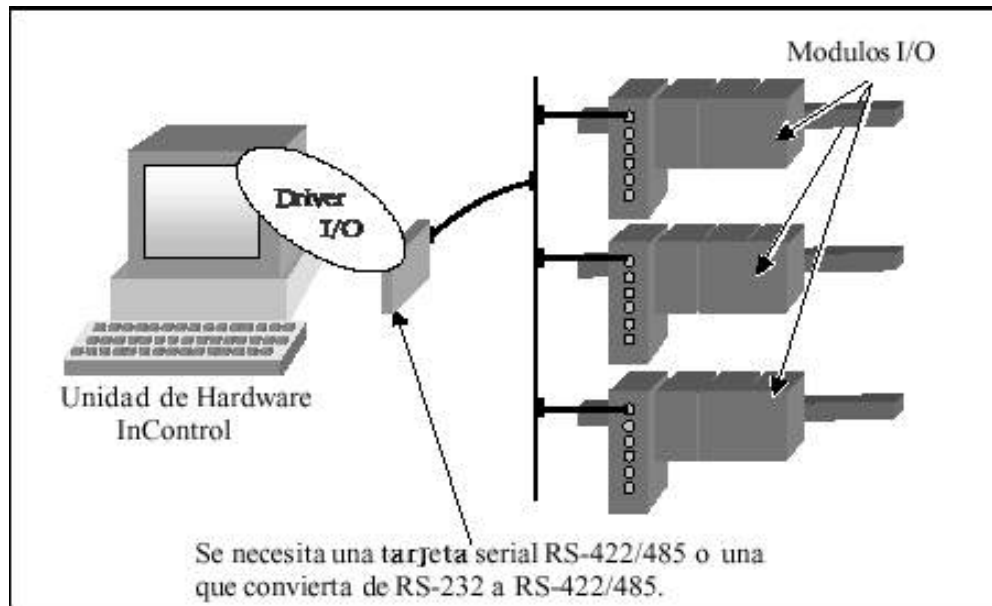


Figura 4. Comunicación de un driver I/O con los módulos I/O

Diferente a otros drivers I/O, el driver SuiteLink no requiere un hardware especial para su instalación. La siguiente figura muestra ejemplos de conexiones SuiteLink que usted puede hacer mientras opera InControl como un cliente.

- A.** InControl, un servidor sobre de un computador, se comunica con InControl corriendo como un cliente sobre otro computador.
- B.** InControl, un cliente en un computador, se comunica con InTouch corriendo como un servidor sobre otro computador.
- C.** InControl, operando como un cliente, se comunica con InTouch corriendo como un servidor sobre el mismo computador.
- D.** InControl, operando como un cliente, comunica con cualquier servidor Wonderware de entrada y salida corriendo sobre el mismo computador.

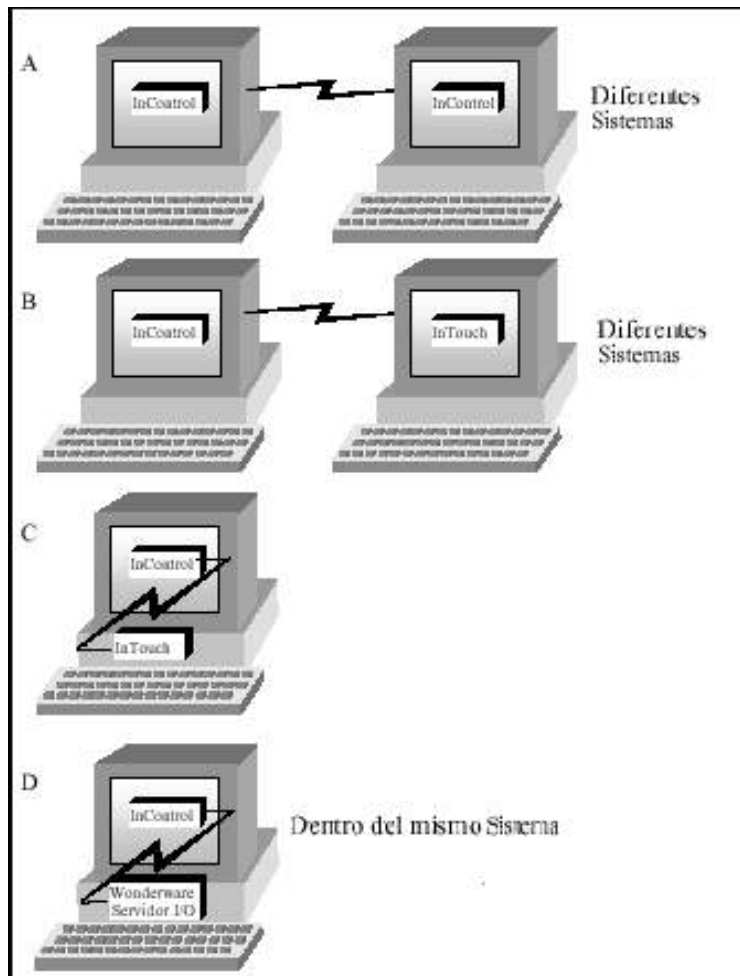


Figura 5. Conexiones SuiteLink de InControl

Varios modelos y diseños de tarjetas de pruebas están disponibles para permitir la comunicación hacia puntos I/O discretos y análogos.

4-6 CAPITULO 4

Para usar un driver I/O con InControl, siga estos procedimientos generales.

✓ Tenga en cuenta la configuración para la unidad de hardware usada para correr InControl: las direcciones de memoria disponibles, las interrupciones, direcciones de puerto.

✓ Consulte el manual de usuario para la tarjeta I/O, y configure e instale la tarjeta en la unidad de hardware.

✓ Instale el driver para la tarjeta I/O.

✓ Adicione el driver a la ventana de proyectos.

✓ Ingrese la configuración I/O (Entrada / Salida).

Información detallada, tal como compensación de memoria, base, puntos, etc., acerca de instalación y operación de una tarjeta I/O está disponible en la documentación de usuario que viene con la tarjeta. Para información acerca de la configuración de tarjetas individuales, consulte la **Guía de Usuario I/O**.

ADICIÓN DE DRIVERS

Después de instalar los drivers y la tarjeta I/O, usted necesita adicionar el driver a la ventana de proyectos antes de configurar las Entradas y Salidas.

Adición de un driver a la ventana de proyectos

- **Para adicionar un driver la ventana de proyectos:**
 1. Pulse **New** en el menú **File**. Aparecerá el cuadro de dialogo **New**.
 2. Pulse la etiqueta **I/O Drivers** para mostrar la lista de drivers instalados.



Figura 6. Drivers Instalados

3. Haga doble clic en el nombre del driver.
4. Ingrese un único nombre para el driver y comience la configuración.

Nota: El driver del cliente SuiteLink de Wonderware, no esta asociado con la tarjeta de prueba, sin embargo, adiciónelo a la ventana de proyectos de la misma manera como usted haría con cualquiera de los otros drivers.

Remover o eliminar un driver

Cuando usted remueve o elimina la configuración de un driver desde la ventana de proyectos, cualquier variable que sea trazada hacia los puntos I/O (Entrada / Salida) será eliminada.

➤ **Para remover o eliminar la configuración de un driver desde la ventana de proyectos:**

1. Seleccione el driver y haga clic-derecho como se muestra abajo.

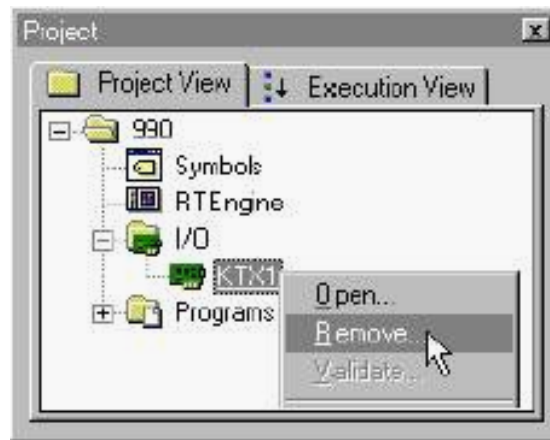


Figura 7. Como remover un driver

2. Pulse **Remove**, Cuando aparezca el cuadro de dialogo **Remove** y le indique para confirmar, pulse **Yes**.

Para eliminar la configuración del driver I/O desde el disco duro, consulte el cuadro de verificación **Delete Associated Files** en el cuadro de dialogo **Remove**.

VERIFICACIÓN DE LA VERSIÓN DEL DRIVER

Usted puede determinar el número de la versión del driver comprobando la barra de título del primer cuadro de diálogo que aparece cuando acceda a la configuración del driver. Para algunos driver, como el driver InterBus-S, el cuadro de diálogo de configuración, es como se muestra en la siguiente figura.

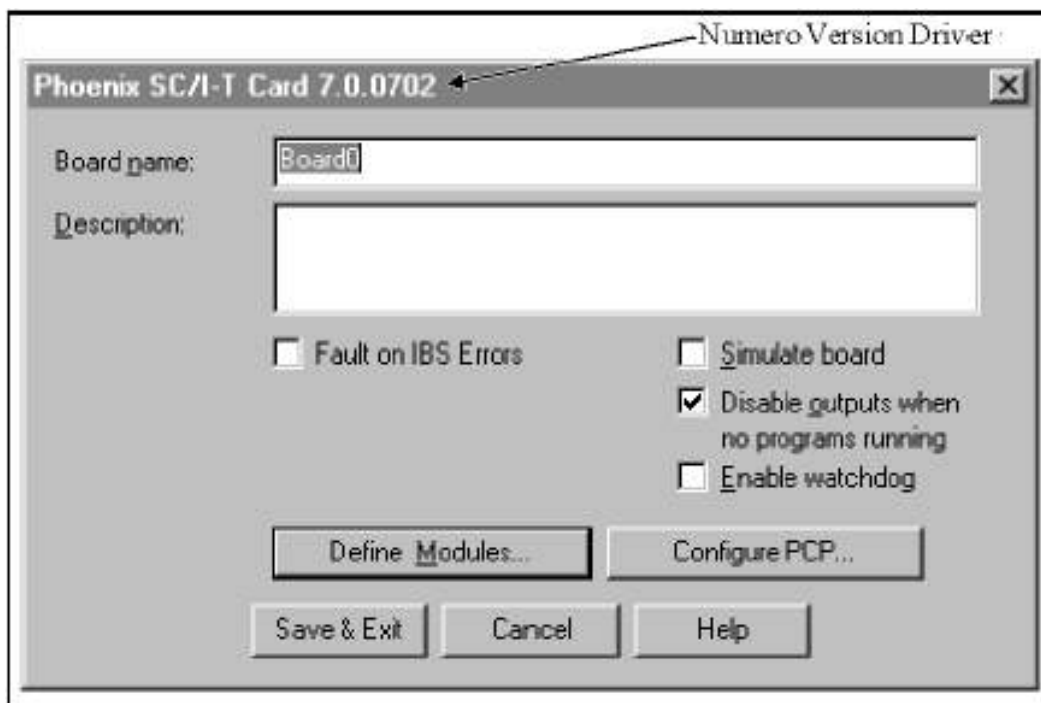


Figura 8. Cuadro de configuración del driver

4-10 CAPITULO 4

Para otros driver, el número de la versión del driver aparece en la presentación de este, el cual muestra todos los objetos asociados con él. Un ejemplo es el mostrado en la figura de abajo.

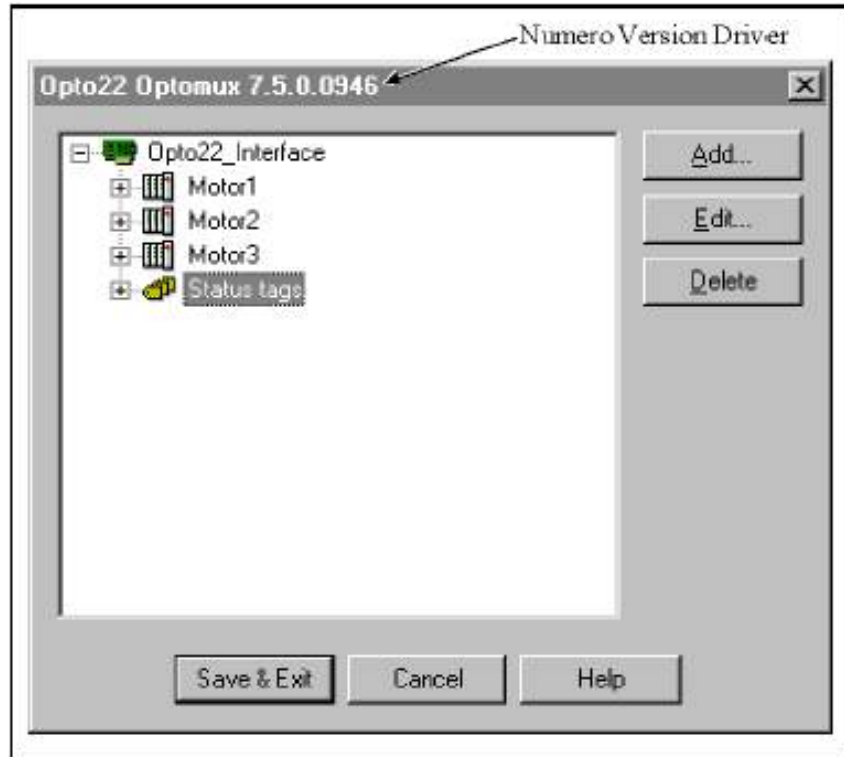


Figura 9. Cuadro de presentación del driver

CONFIGURACIÓN I/O

Información acerca de la configuración de las tarjetas y asignación de los nombres simbólicos para los puntos I/O (Entrada / Salida) esta descrito en la guía de usuario I/O. Que son copiadas en el disco duro cuando se instala el InControl. Algunos de los drivers I/O mantenidos actualmente requieren de una configuración de proveedor específico útil para definir la configuración de entrada y salida. Revise sus CD's distribuidos para estas utilidades e instálelos cuando este usando estos drivers.

Algunos de los drivers I/O soportados actualmente le permiten abrir y editar la configuración I/O de un proyecto que correrá sobre un nodo remoto.

Los drivers I/O deben ser instalados tanto en el nodo remoto como en el local, donde usted hace la configuración, además, para algunos drivers, usted puede necesitar instalar la tarjeta de prueba en el nodo local así como en el nodo remoto.

Nota: varios de los drivers I/O proporcionan utilidades que puede usar para hacer una configuración automática y/o ejecutar un diagnóstico en línea. Algunos de estos drivers requieren que usted use estas utilidades sobre el mismo nodo remoto. Los nuevos drivers soportan configuración automática remota y diagnóstico en línea. Para lograr esto, ellos mismos pueden cargarse el programa remoto de ejecución. En este caso, el driver ingresa el modo cargado.

Tarjetas de control, arquitectura de buses de control, y módulos I/O, todos tienen diferentes requerimientos de tiempo, los cuales son a menudo implementaciones específicas. Es posible fijar un tiempo de análisis total dentro del InControl que es más rápido que el que manejan los dispositivos. Para información acerca de cómo manejar esta situación, vea "**Ajustando el tiempo de análisis**" en el capítulo "**Administración del Sistema InControl**".

SIMULACIÓN I/O

La Simulación I/O (Entradas / Salidas) le proporciona los siguientes beneficios cuando usted desarrolla un proyecto.

- Usted puede definir etiquetas de Entrada y Salida antes de que su tarjeta de prueba asociada haya sido instalada.
- Usted puede probar el código del programa que relaciona las etiquetas de Entrada y Salida sin enviar realmente señales a los dispositivos de Entrada y Salida.

Muchos de los drivers que soporta el InControl tienen un cuadro de verificación “**Simulate**” que usted puede usar para propósitos de simulación I/O. A menudo esto le permite hacer las pruebas que usted necesita. Sin embargo, algo más de la tercera-Parte de los drivers pueden requerir que la tarjeta este instalada, incluso cuando ellos estén operando en un modo de simulación. Otra tercera parte de los drivers no permiten hacer la prueba apropiada para su aplicación desde el modo de simulación.

Para una forma alternativa de simulación I/O usted puede excluir un driver I/O desde la carga del proyecto. Si es necesario, usted puede excluir todos los drivers. La exclusión le permite probar etiquetas de Entrada y Salida, incluso para esos driver que requieren que la tarjeta de prueba o análisis sea instalada cuando usted acceda al driver.

➤ **Para excluir etiquetas de Entrada y Salida para todos los drivers I/O:**

1. Desde la **Project View**, haga clic derecho en la carpeta **I/O** como se muestra abajo.

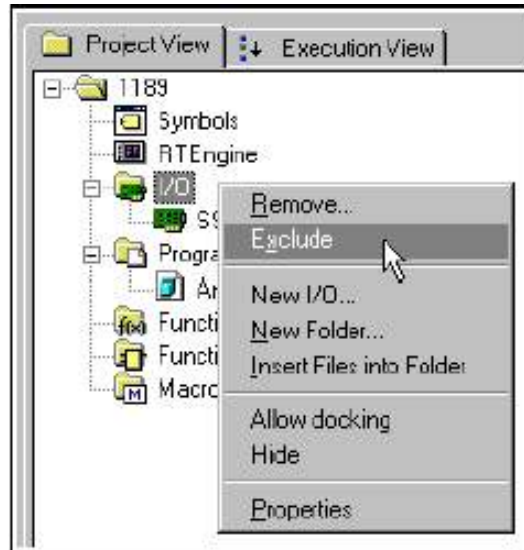


Figura 10. Project View

2. Pulse la etiqueta **Exclude**.

➤ **Para simular etiquetas de Entrada y Salida para un solo driver:**

1. Desde la **Project View**, pulse con el botón derecho sobre el driver a utilizar.
2. Pulse la etiqueta **Exclude**.

MANEJO Y ORGANIZACIÓN DE PROYECTOS

Este capítulo describe la organización de los proyectos en términos del concepto IEC-61131-1 acerca de la unidad de organización de programas (POU).

CONTENIDO

- INTRODUCCIÓN
- MANEJO DE PROYECTOS

This product is licensed to:

Company:	N/A
SN:	N/A
Expires:	N/A
Locked To:	N/A

7.1.0.1337

INTRODUCCIÓN

InControl se sigue de los requerimientos de la norma IEC-61131-1 para el diseño de códigos en las unidades de organización de programas (POU's), en este capítulo las POU's soportadas por el InControl son las descritas a continuación:

- Programas.
- Bloques de funciones.
- Funciones.
- Macros.

Usted puede diseñar POU's en una variedad de lenguajes de programación, incluyendo RLL, Texto Estructurado y SFC. Desde que el editor de objetos de fabrica del InControl es un contenedor ActiveX, usted también puede adherir controles ActiveX y objetos de fabrica (FOE's) a los proyectos. Los lenguajes de programación, variables, y tipos de datos que usted puede usar en sus códigos son descritos en los siguientes capítulos.

Programas

Un programa es un bloque de códigos que puede ser programado para ejecutar automáticamente cada análisis. Estos programas tienen un mecanismo principal para ejecutar su lógica de control, que tiene las siguientes características:

- Los programas se ejecutan automáticamente, esto es, usted no tiene que llamar al programa para su ejecución, es como si llamaras a una función o bloques de funciones.
- Un programa es definido por sus lenguajes de programación: RLL, texto estructurado o SFC, en adición, el InControl es compatible con las especificaciones de los servidores ActiveX. El editor de objetos de fabrica del InControl es un contenedor ActiveX, que permite adherir controles ActiveX y objetos de fabrica (FOE's) a los proyectos.

- Los programas pueden contener declaraciones de variables locales y globales. Las variables globales pueden ser leídas o escritas por otros programas, funciones, y FOE's. Con unas cuantas excepciones, las variables locales solamente pueden ser leídas o escritas junto al programa donde fueron definidas.
- El InControl soporta solamente ciertos programas, como los definidos en la especificación IEC 61131-3. el InControl no soporta programas de tipo definición.
- Los programas no pueden contener partes de otros programas.

Bloques de Funciones

Los bloques de funciones POU consisten de un conjunto de instrucciones de programación que pueden ser llamadas para ejecución por otra POU. Los bloques de funciones tienen las siguientes características:

- El InControl soporta bloques de funciones de tipo definición. Usted puede crear una o más casos de un tipo de bloques de funciones.
- Una función de bloque de este tipo es descrita por su lenguaje de programación. Actualmente, el InControl soporta bloques de funciones que están escritos en texto estructurado.
- Existen casos de bloques de funciones que no se ejecutan automáticamente. Ejemplos de bloques de funciones que son ejecutadas cuando el orden del texto estructurado relacionado con ellos es ejecutado.
- Bloques de funciones de tipo Definición pueden contener declaraciones de variables locales y de parámetros de entrada y salidas.

Las variables locales dentro de un ejemplo de bloques de funciones mantienen sus valores entre las llamadas de estas. Esto le permite a usted mantener un conteo, por ejemplo, cada vez que el tipo de bloques de funciones es llamado. Sin embargo, las variables locales no pueden ser leídas o escritas por otras POU's. Los Parámetros de entrada y salida también mantienen sus valores entre llamadas. Otros POU's pueden relacionarse con estos parámetros.

- Las llamadas a los bloques de funciones no pueden ser repetitivas. Un tipo de bloques de funciones puede llamar otro tipo de bloques de funciones para su ejecución, pero no puede llamarse así mismo. Esto es, una función de bloques **X** no puede llamarse a sí misma o a otra función de bloques **Y**, la cual a su vez llamara a la función de bloques **X**.
- Todo lazo constructor continuará su ejecución hasta que se haya completado.
- El examinar y cambiar el orden del texto estructurado y el permitir depurar, son tareas que no son aprobadas en estos tipos de bloques de funciones.

Funciones

La Función POU consiste en un conjunto de instrucciones de programación que pueden ser llamadas para ejecución de un programa, una función de bloque u otra función. Estas funciones tienen las siguientes características:

- Una función es descrita por su lenguaje de programación. Actualmente, el InControl soporta funciones que son escritas en lenguaje de texto estructurado.
- Las funciones no se ejecutan automáticamente. Las funciones son ejecutadas cuando el orden del texto estructurado que se relaciona a ellas es ejecutado.
- Las funciones pueden contener variables locales y parámetros de entrada y salida. El espacio en memoria para las variables locales y los parámetros es asignado cada vez que la función es llamada, y es liberado cuando la función finaliza su ejecución. Las variables solamente existen en el instante en que la función se está ejecutando y no pueden ser leídas o escritas por otras POU's.
- La función puede tener una opción que regrese el valor que contenía el resultado de la función después de la ejecución.
- Llamar a las funciones puede no ser repetitivo. Una función puede llamar a otra función para su ejecución, pero esta no puede llamarse a sí misma. Esto es, la función **X** no puede llamarse a sí misma o a otra función **Y**, la cual sucesivamente llama la función **X**.
- Todos los lazos constructores continuaran su ejecución hasta que estén completos.

Diferencias entre Funciones y Bloques de Funciones

Tres características que diferencian los bloques de funciones y las funciones:

- Usted puede crear una o más ejemplos de un tipo de función de bloques.
- Los parámetros y las variables locales se guardan de modo diferente..

Para el caso de bloques de funciones: las variables locales mantienen sus valores entre las llamadas.

Para las funciones: el espacio en memoria para las variables locales y los parámetros es asignado cada vez que la función es llamada, y es liberado cuando la función finaliza su ejecución. Las variables solamente existen para en el instante en que la función se está ejecutando.

- Las funciones pueden tener una opción que regrese el valor que contiene el resultado de la función después de la última ejecución.

Macros

El macro es una POU especializada que provee una manera de incluir un SFC, el hijo, para su ejecución a partir de un paso en otro SFC, el padre. El macro representa una mejora para la IEC-61131-3. tiene las siguientes características:

- Los macros son ejecutados automáticamente.
- Un macro SFC no puede ser incluido a sí mismo o a su SFC padre.
- Usted no puede anidar SFC's. Esto es, un macro SFC no puede llamar a otro macro SFC.
- Usted puede llamar a un mismo macro desde múltiples puntos, llamado pasos de macros, dentro de una SFC, y desde múltiples programas.
- Los macros pueden tener variables locales.

Denominación de variables

Se usan Solamente caracteres alfanuméricos y acentuados(subrayados) para nombrar a una función POU. Los nombres pueden comenzar con un carácter acentuado (subrayado) o un alfanumérico, pero no un carácter numérico. La longitud máxima de un nombre son las siguientes:

- Tarjetas I/O (Entrada / Salida): 31 caracteres.
- Programas, funciones y bloques de funciones: 31 caracteres.
- Proyectos: 255 caracteres.
- Otros símbolos: la longitud máxima recomendada es de 100 caracteres.

MANEJO DE PROYECTOS

Todas las funciones POU's que usted crea aparecerán en la ventana de proyectos. Esta ventana tiene dos etiquetas: la etiqueta **Project View** que muestra los programas, los tipos de bloques de funciones que existen, las funciones y los macros que están asociados con un proyecto.

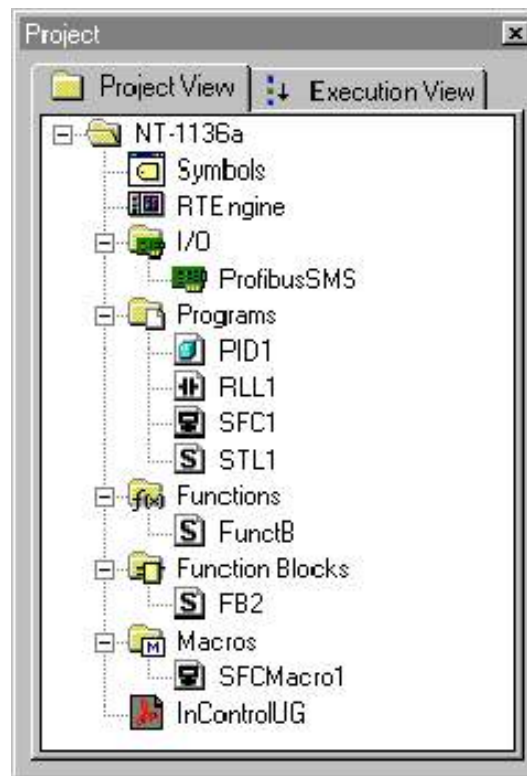


Figura 1. La ventana de proyectos.

Usted puede también manejar símbolos, gestiones de entrada y salida y configuración del tiempo de ejecución de un proyecto a través de la etiqueta Project View. Y por ultimo la etiqueta **Execution View** que muestra los niveles de prioridad para programas(exploración normal, prioridad baja) y el orden en el cual los programas son ejecutados.

Creación de un proyecto

Antes de usted crear el conjunto de POU's, símbolos y archivos de configuración relacionados, que manejan la ejecución de una aplicación de control lógico, usted primero debe crear un proyecto.

➤ Creación de un proyecto después de iniciar el InControl:

1. Sobre el menú **File**, pulse **Project**. Aparecerá el **Project Manager**.

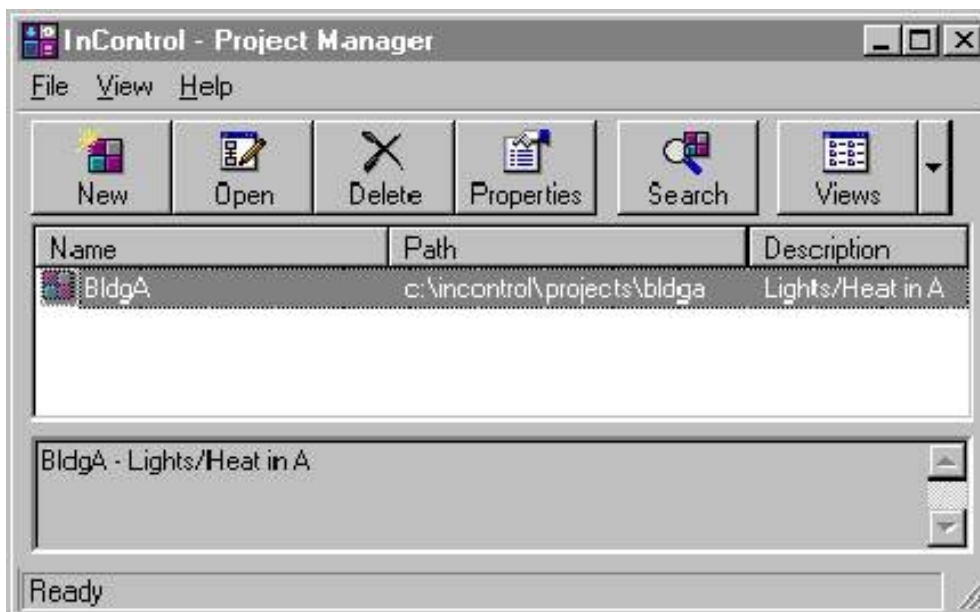


Figura 2. El InControl Project Manager.

2. Sobre el menú **File** del **Project Manager**, pulse en **New**. Aparecerá el cuadro de dialogo **Create InControl Project**.

3. Escriba el nombre del proyecto usando las convenciones para nombres de Windows NT, seleccione una ruta, y pulse **OK**. Un directorio con este nombre es creado en el disco duro y el nuevo proyecto es adicionado a la lista de proyectos de InControl. En la siguiente figura, se muestra el proyecto de nombre “EjemploFuncion-Bloque” que fue creado.

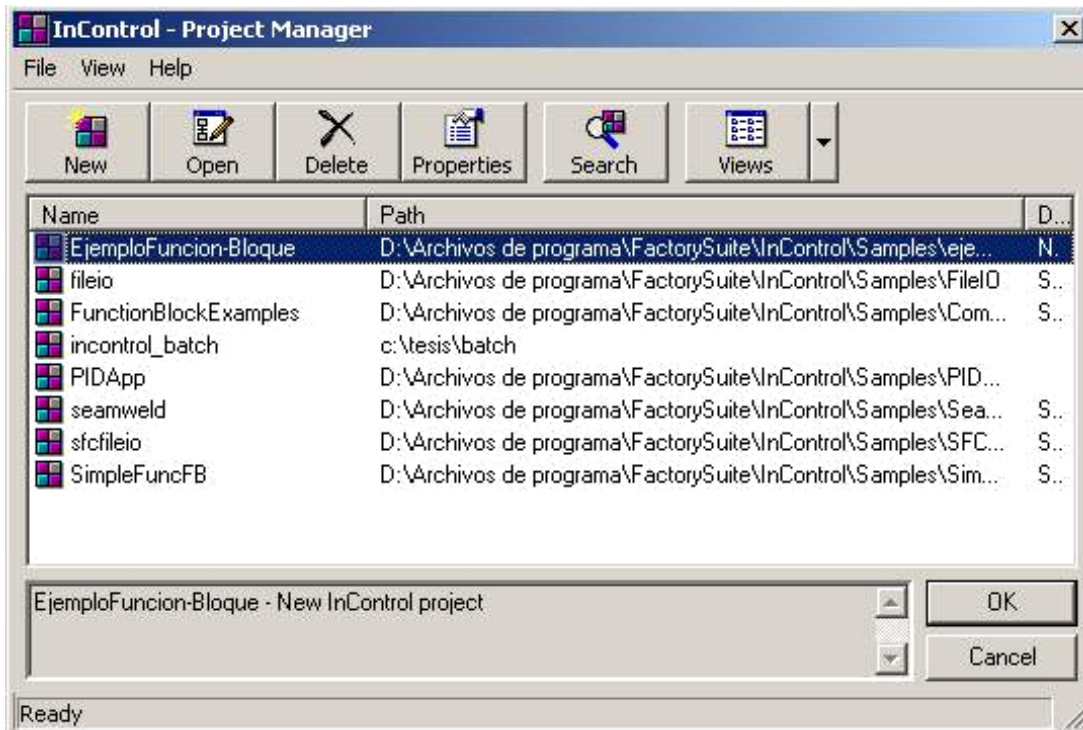


Figura 3. Presentación de un nuevo proyecto.

4. Haga doble clic en el nombre del proyecto para abrirlo en el ambiente de desarrollo. Si su versión de InControl soporta mas de una versión del Runtime Engine, aparecerá el cuadro de dialogo **Runtime Engine Target**.
5. Seleccione el propósito de la plataforma de hardware y pulse **OK**. A menos que intente correr programas sobre otra plataforma de hardware, seleccione Windows NT.

El proyecto que usted creó aparece en la ventana de Proyectos del ambiente de desarrollo.

Nota: Usted puede convertir un proyecto que fue desarrollado para un objetivo correrlo para otro objetivo. Sin embargo, algunas tarjetas de Entrada y Salida no puede convertirse si ellas no soportan el objetivo del sistema operativo. Consulte "**Designación de la plataforma de hardware**" pagina (5-35). Además de convertir proyectos, usted puede insertar archivos de programa que son desarrollados en un proyecto para un objetivo, dentro de un proyecto con un objetivo diferente.

Eliminación de un proyecto

Para remover un proyecto, usted debe mostrar el **Project Manager**.

➤ **Para remover o borrar un proyecto:**

1. Sobre el menú **File**, pulse **Project**. Aparecerá el **Project Manager**.

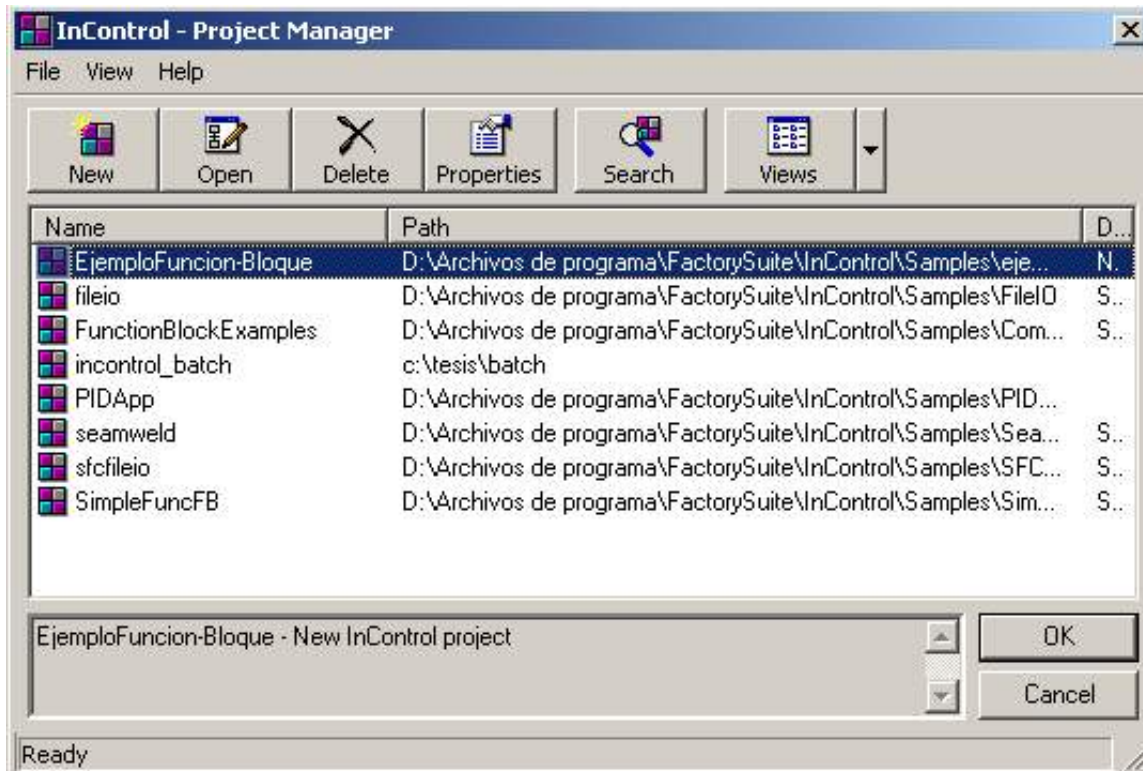


Figura 4. El Project Manager

2. Seleccione el proyecto, y sobre el menú **File** del **Project Manager** pulse en **Delete**. Cuando aparezca el cuadro de dialogo **Delete** y le indique para confirmar pulse **Yes**.

Note que removiendo un proyecto no se borran sus archivos del disco duro. Para borrar todos los archivos asociados con el proyecto, revise el cuadro de verificación **Delete Associated Files** en el cuadro de dialogo **Delete**.

Nota : Usted puede borrar archivos de proyectos individuales, tal como programas o drivers de entrada y salida, dentro del ambiente InControl. Ver “**Como remover una POU**”.

Adición de un proyecto

Usted puede adicionar un proyecto que usted haya removido desde el **Project Manager**.

➤ Para Adicionar un proyecto:

1. Sobre el menú **File**, pulse **Project**. Aparecerá el **Project Manager**.
2. Sobre el menú **File**, del **Project Manager**, pulse en **Search**. Aparecerá el cuadro de dialogo **Buscar Carpeta**.

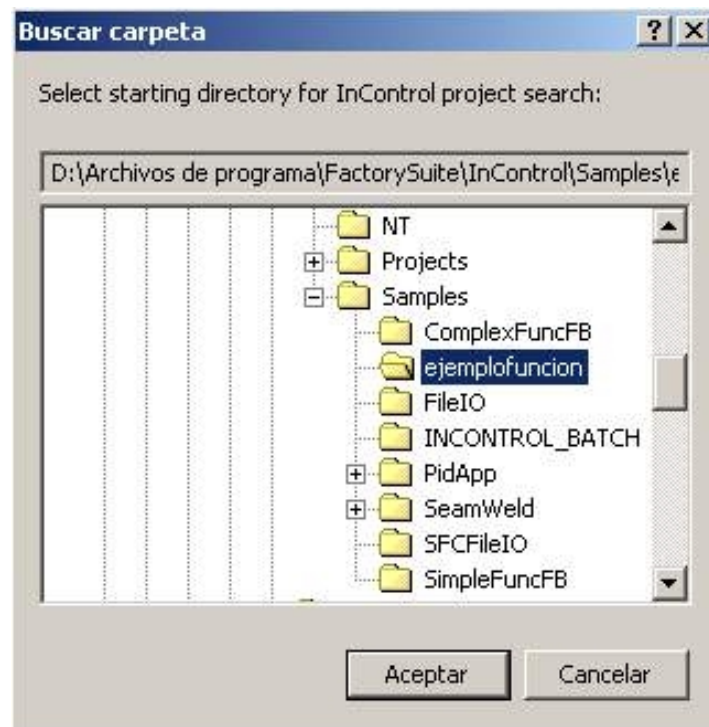


Figura 5. Selección de la carpeta del nuevo proyecto.

3. Seleccione el proyecto y pulse en **OK**. El proyecto será adicionado a la lista de proyectos.

Nota: usted puede agregar un proyecto (y sus archivos de programa asociado) solamente si usted no lo ha borrado de su disco duro.

Modificación del nombre y la descripción de un proyecto

- **Para modificar el nombre o la descripción de un proyecto:**
1. Sobre el menú **File**, pulse **Project**. Aparecerá el **Project Manager**.
 2. Seleccione el proyecto y sobre el menú **File** del **Project Manager**. Pulse en **Properties**. Aparecerá el cuadro de dialogo **Modify InControl Project**.



Figura 6. Modificación del nombre y la descripción de un proyecto.

3. Ingrese la nueva descripción o el nombre del proyecto, entonces pulse **OK**. La nueva descripción aparecerá solamente en rojo en el campo **Description** del cuadro de dialogo **InControl Projects**.

Adición de una nueva POU a un proyecto

Cuando usted adiciona un programa, tipos de bloques de funciones, funciones o macros al proyecto, estos objetos aparecerán sobre sus respectivas carpetas en la Project View. Para organizar los componentes de su proyecto, usted puede crear subcarpetas si es necesario.

➤ Para agregar una nueva POU al proyecto:

1. En el menú **File**, pulse en **New**.
2. Seleccione la etiqueta de programas en el cuadro de dialogo **New**.
3. Seleccione el tipo de programa(Objetos de Fabrica, RLL, SFC, Texto Estructurado), y el tipo de POU(Programa, Función, Bloque de Función, Macro).
4. Escoja un nombre (encima de 31 caracteres).
5. Si usted adiciona una nueva función o función de bloques usted necesita definir parámetros y variables.

Consulte el apéndice “**EJEMPLOS DE PROGRAMAS**” para ejemplos de procedimientos de escritura de programas, funciones y bloques de funciones. Una nueva POU aparece en un proyecto, lista para ser editada, al hacer doble clic sobre la POU se cierra la orden de edición.

Seleccione una POU y al hacer clic derecho se muestra lo siguiente:

- **Open:** Inicia la sesión de edición.
- **Symbols:** Para acceder al Symbol Manager.
- **Remove:** Para remover la POU desde la Project View.
- **Exclude:** Para excluir la POU desde el proyecto cuando usted cargue el proyecto, usted puede todavía cargar POU's excluidas individualmente. Esta característica es útil para agregar códigos de simulación en su proyecto. Cuando una POU es excluida, su icono se desactiva(esta cambia a un color opaco).
- **Validate:** Para Validar el código(revisa la sintaxis).
- **Download:** Carga la POU dentro del Runtime Engine.
- **Run:** Ejecuta la POU en el Runtime Engine.

- **Pause:** Suspended the execution of the POU.
- **Stop:** Stops the execution of the POU and downloads it from the Runtime Engine.
- **Properties:** Reviews the properties of the following codes:
Registers descriptive information about the POU in the dialog box **Properties**.
Enables the error debugger.
Excludes the POU from the project, described previously.
Verifies the following dates: the date when the POU was modified, the date when the POU was compiled, and the date when the POU was compiled in the Runtime Engine.

The InControl indicates the modes or states of a program in the Project View. The following figure shows these states of the programs: the program **PID4** is in execution mode; the **RLL5** is in stopped mode; **SFC10** is in pause mode (the asterisk indicates that the program has been modified and is not saved, or is different from the copy in the Runtime Engine). Note that if the mode is not shown, the program is not stopped, not loaded in the Runtime Engine, the development environment is not connected to the Runtime Engine, or the project is different from the copy in the Runtime Engine.



Figura 7. Estados de los programas en la carpeta Programs.

The InControl indicates the functions and the function blocks that are loaded in the Runtime Engine, to show their mode in the Project View as **Loaded**.

Adición de una POU existente a un proyecto

➤ Para adherir una POU existente a un proyecto:

1. Sobre el menú **Insert**, haga clic en **Files Into Project**. Aparecerá el cuadro de dialogo **Insert Files Into Project**.

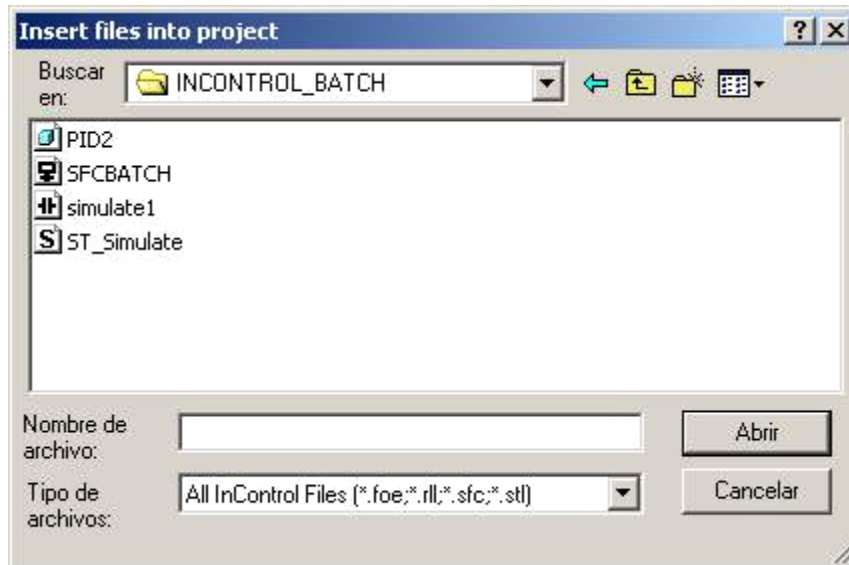


Figura 8. Adición de una POU al proyecto.

2. Localice la POU que usted quiere adicionar.
3. Haga clic en **OK**. La POU es adherida a la Project View.

Nota: Todas la POU's son insertadas en la carpeta de programas de la ventana de proyectos. Usted debe mover las funciones hacia la carpeta de funciones. Los bloques de funciones hacia la carpeta de bloques de funciones, y los macros hacia la carpeta de macros para que el proyecto los compile correctamente. Si usted abre un proyecto desarrollado sobre InControl 7.0, usted tiene la opción de convertir los archivos hacia un proyecto de InControl 7.1. cualquier macro en el proyecto aparecerá en la carpeta de programas después de la conversión. Usted puede mover estos macros hacia la carpeta de macros. Pero esto no es requerido para la compilación del proyecto.

Usted puede agregar archivos de programa que no son parte del InControl, tal como, un archivo **Readme.txt**, de la misma forma en que se agrega una POU.

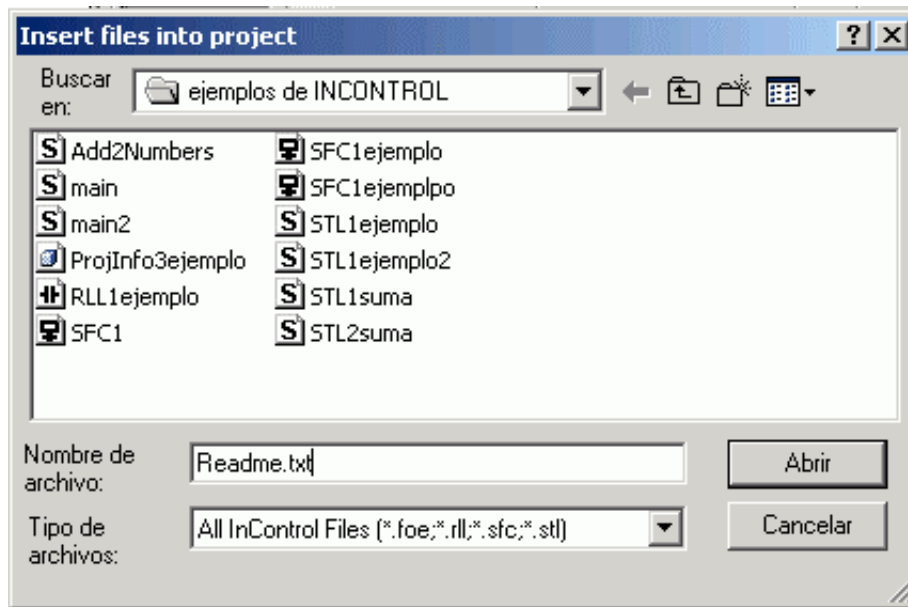


Figura 9. Adición de un archivo de texto al proyecto.

Al hacer doble clic sobre el archivo **Readme.txt** en la Project View, se abre la utilidad **Bloc de notas** de **Windows NT** que muestra a este archivo en especial. Si usted adiciona un archivo ejecutable, tal como la utilidad bloc de notas de **Windows NT (wordpad.exe)**, la hacer doble clic sobre el nombre del archivo en la Project View correrá el archivo ejecutable.

Nota: Usted puede adherir un archivo a un proyecto por arrastre y bajada desde cualquier aplicación que soporte las características de arrastrar y bajar, tal como el explorador de **Windows NT**.

Como remover una POU

Para remover o borrar una POU de un proyecto, usted debe mostrar la etiqueta Project View.

➤ **Para remover o borrar una POU de un proyecto:**

1. Muestre la etiqueta Project View.
2. Seleccione la POU y haga clic derecho como se muestra abajo.

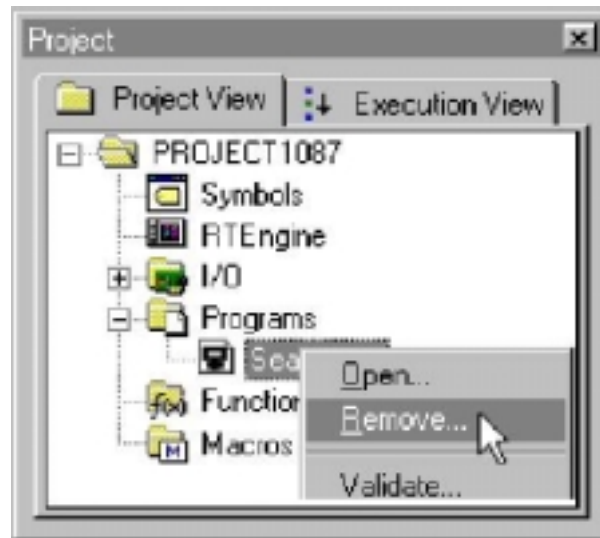


Figura 10. Como remover una POU del proyecto.

3. Haga clic en **Remove**. Cuando aparezca el cuadro de dialogo **Remove** y le indique para confirmar, haga clic en **Yes**.

Note que removiendo una POU no se borran sus archivos del disco duro. Para borrar sus archivos asociados, revise el cuadro de verificación **Delete Associated Files** en el cuadro de dialogo **Remove**.

Como renombrar a una POU

Para renombrar a una POU, use la opción **Save As** en el menú **File**.

➤ **Para renombrar a una POU:**

1. Muestre la etiqueta Project View.
2. Seleccione la POU.

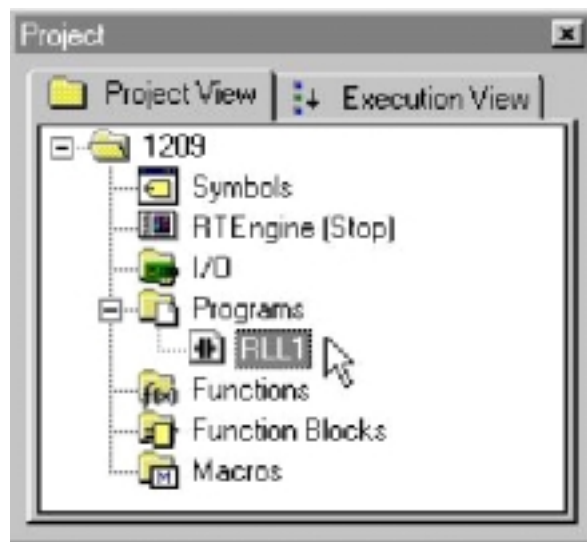


Figura 11. Selección de la POU a renombrar.

3. Pulse en la etiqueta **Save As** en el menú **File**.
4. Escoja un nombre (por encima de 31 caracteres). Un cuadro de mensaje le indicara para confirmar así como para adherir la POU al proyecto.

Nota: La POU original permanece en el proyecto después que usted la guarda con un nombre diferente.

Organización de un proyecto

La ventana de proyectos de InControl es una herramienta versátil que le permite a usted una fácil organización de los componentes de su proyecto. La siguiente figura muestra en la ventana de proyectos la configuración de las carpetas de un nuevo proyecto.

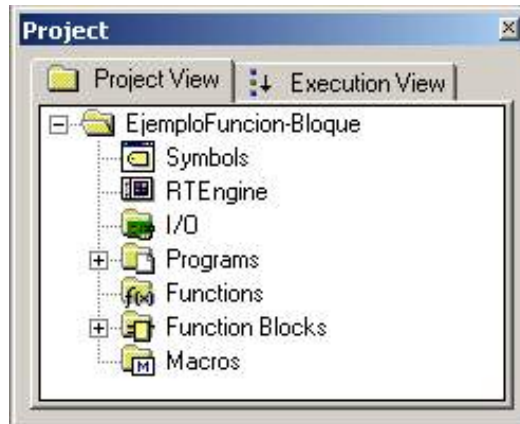


Figura 12. Configuración de las carpetas en la ventana de proyectos.

Cuando usted adiciona POU's y configuraciones I/O, los nuevos objetos aparecen en las carpetas apropiadas. En la siguiente figura la carpeta I/O muestra dos configuraciones I/O, La carpeta de programas muestra dos programas, la carpeta de funciones muestra una función, la carpeta de bloques de funciones muestra dos tipos de bloques de funciones y la carpeta de macros muestra un macro.

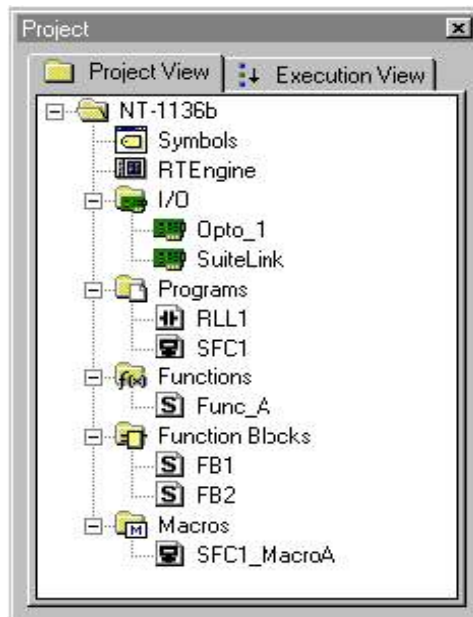


Figura 13. Organización de las POU's en la ventana de proyectos.

Si usted quiere grupos de POU's por unidades de áreas de la planta en la fabrica, usted puede hacer clic derecho sobre el nombre del proyecto y crear una nueva carpeta. Haga clic derecho en la nueva carpeta y adicione las POU's y las configuraciones I/O que son apropiadas para esas unidades de la fabrica. En la siguiente figura, dos configuraciones I/O, dos programas, una función, dos tipos de bloques de funciones aparecen sobre la carpeta **Conveyor_A**. La otra unidad de fabrica, es una caldera(boiler) con una carpeta llamada **Boiler_East_Wing**, es representada por una configuración I/O y dos programas.

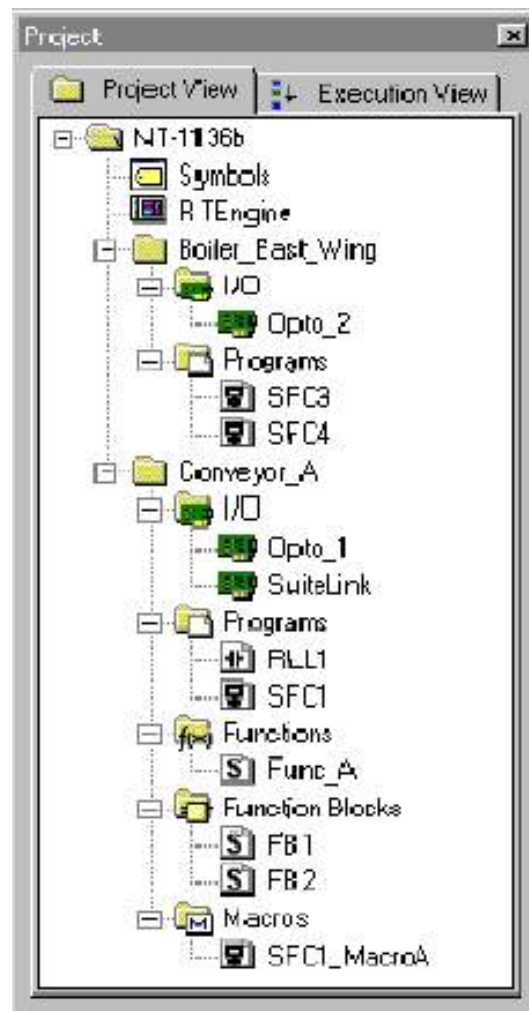


Figura 14. Organización de POU's por unidades de áreas.

Usted puede arrastrar POU's y configuraciones I/O existentes en carpetas si usted quiere reconfigurar la organización del proyecto. Si la subcarpeta apropiada no existe (usted arrastrara la función dentro de una carpeta que no contiene una carpeta de funciones lista), el InControl creara la subcarpeta automáticamente.

Usted puede renombrar cualquiera de las carpetas, incluyendo la carpeta de proyectos estándar (Programas, Funciones, Macros, etc.).

Definición de Bloques de Funciones

Después de crear un nuevo tipo de bloques de funciones, usted necesitará especificar sus parámetros y las variables.

Ajuste de parámetros y variables

Usted define los parámetros de entrada y salida y las variables para un tipo de bloques de funciones en el **Symbol Manager**. Las variables locales de una función de bloques son propias de esa función de bloques y no pueden ser relacionadas por cualquier otra POU. Dentro de este caso de bloques de funciones, los parámetros de entrada son solamente leídos.

- **Para definir un parámetro o una variable para un tipo de bloques de funciones:**
 1. Agregue un bloque de función al proyecto. Para información acerca de crear bloques de funciones, ver “**Adición de una nueva POU al proyecto**”.
 2. Sobre el menú **Tools**, pulse en **Symbol Manager**.
 3. Haga clic sobre el tipo de bloque de función a seleccionar.

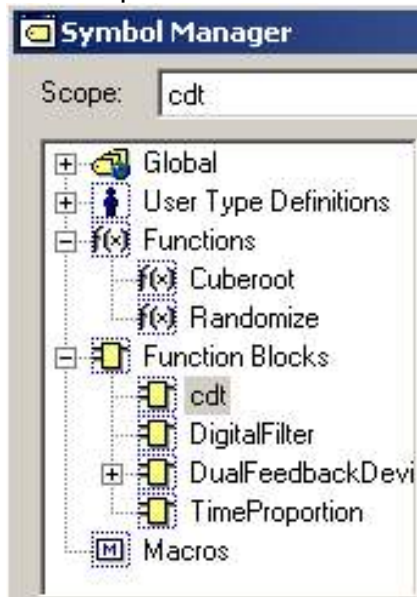


Figura 15. Selección del tipo de bloque de función.

- Haga clic en **New** en la barra de tareas del **Symbol Manager**.



Figura 16. Herramienta para insertar Parámetros o variables.

Aparecerá el cuadro de dialogo **Symbol Properties**.

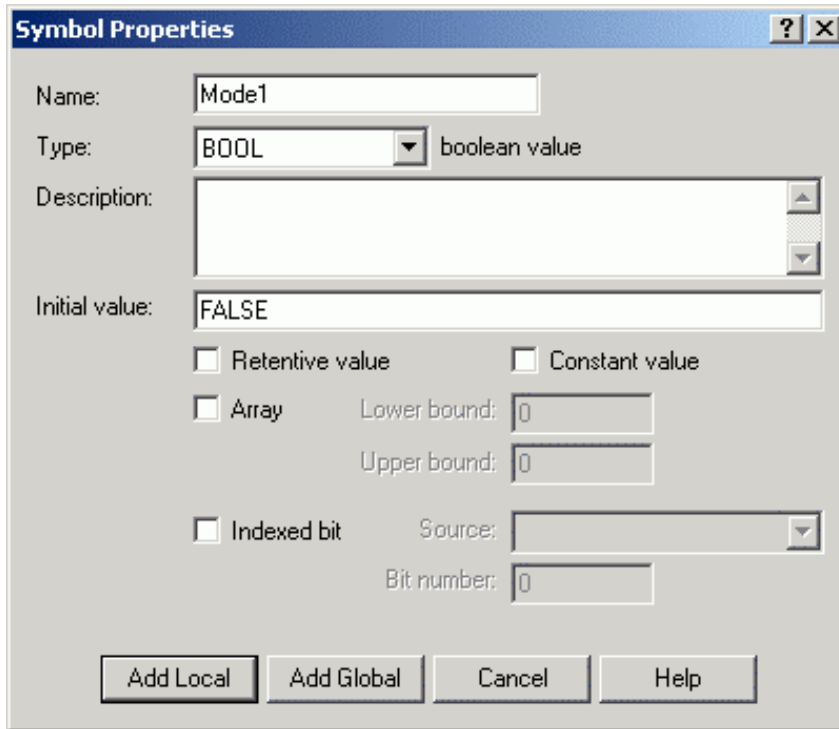


Figura 17. El cuadro **Symbol Properties**.

- Ingrese el nombre del parámetro o la variable dentro del campo **Name**(mas de 100 caracteres). Use solamente caracteres alfanuméricos y subrayados.
- Seleccione el tipo de dato en el campo **Type**.
- Ingrese la descripción opcional dentro del campo **Description**.
- Escoja la variable o el tipo de símbolo en el campo **In/Out**, como se muestra abajo.

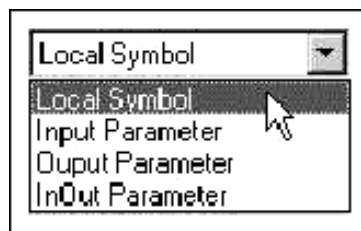


Figura 18. Selección del tipo de variable o símbolo.

9. Si el parámetro o la variable requieren de un valor inicial, ingrese el valor dentro del campo **Initial Value**.
10. Pulse en el cuadro de comprobación **Retentive Value** de la variable, es para retener su valor en el caso de una falla de energía.
El InControl periódicamente guarda variables retenidas en el disco duro. Por defecto la frecuencia es cero. Este intervalo es configurable y usted puede fijarlo en el cuadro de dialogo **Runtime Engine Properties** descrito en “Ajustando el Tiempo de Exploración” del capítulo “**ADMINISTRACIÓN DE SISTEMAS INCONTROL**”.
11. Si usted está definiendo los parámetros o variables como un arreglo o serie. Revise el cuadro de comprobación **Array**(arreglo o serie). Esto causa que los campos **Lower Bound**(límite inferior) y **Upper Bound** (límite superior) permanezcan activos. Ingrese el valor más bajo y el más alto en los campos apropiados.
12. Pulse en la etiqueta **Add Local** para completar la definición de los parámetros o variables.

Nota: Usted puede cambiar el orden de los parámetros en una función de bloque existente con solo hacer clic derecho sobre el parámetro, y cambiar el acceso a un tipo de función de bloque en el Symbol Manager con solo hacer clic en las opciones **Decrease Address**(Disminuir dirección) o **Increase Address**(incrementar dirección).

Ejemplos de bloques de funciones

Usted puede definir un caso o tipo de un tipo de bloques de funciones en el Symbol Manager.

➤ **Para definir un tipo de bloques de funciones:**

1. Ingrese al Symbol Manager.
2. Seleccione el alcance apropiado, por ejemplo, global, un programa individual, otra función de bloque de función, etc.

3. Pulse en **New** en la barra de tareas del Symbol Manager.



Figura 19. Herramienta para insertar el nuevo bloque de función.

Aparecerá el cuadro de dialogo **Symbol Properties**(propiedades de los símbolos).

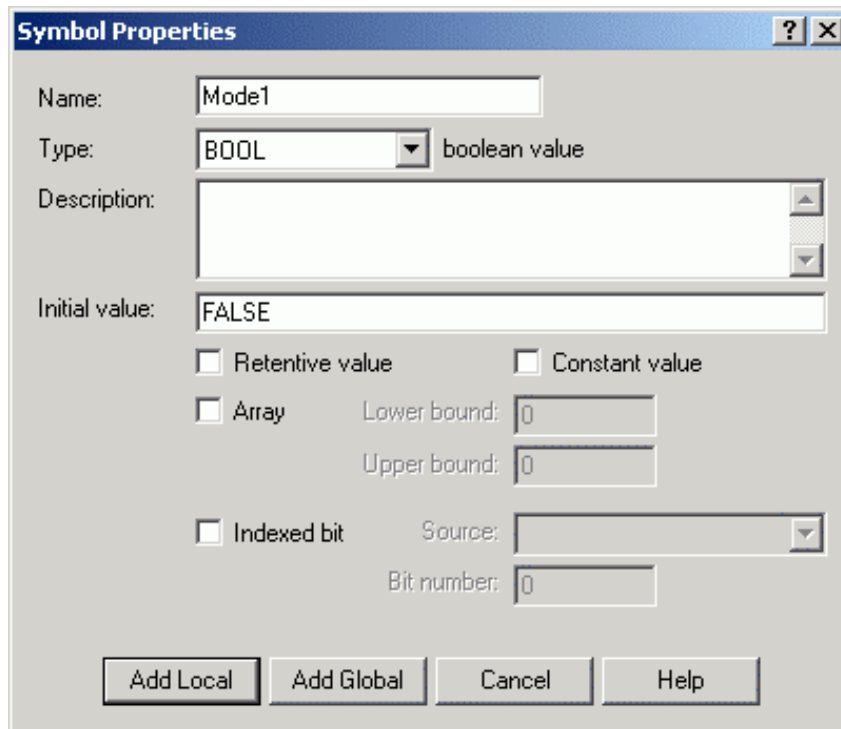


Figura 20. El cuadro Symbol Properties.

4. Ingrese el nombre dentro del campo **Name**(mas de 100 caracteres). Use solamente caracteres alfanuméricos y subrayados.
5. Seleccione el tipo de función de bloque en el campo **Type**.

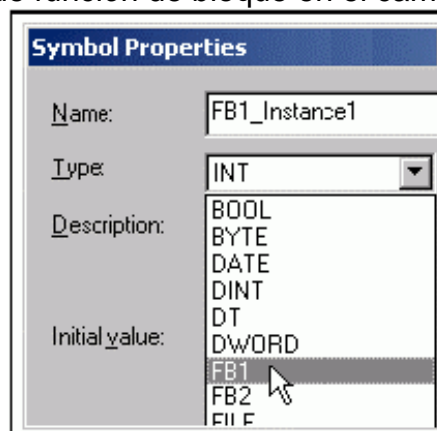


Figura 21. Selección del tipo de función de bloque.

6. Ingrese la descripción opcional dentro del campo **Description**.
7. Puesto que todas las propiedades permanentes están definidas por el tipo de función de bloques, pulse clic en **Add Local(adición local)** o **Add Global(adición global)** para completar la definición para el caso.

Código para llamar a un bloque de funciones

Usted puede llamar a una función de bloque para su ejecución desde cualquier otra POU que este escrita en Lenguaje de Texto Estructurado. Esta sección le da dos ejemplos de sintaxis que usted puede usar.

```
<FB_Instance>      (<Parameter1Value>,      <Parameter2Value>,
<Parameter3Value>,.....) ;
<result>: (<FB_Instance.ParameterOut>);
```

Por ejemplo, usted ha definido un tipo de función de bloque llamado CALC que esta basada en el calculo de dos parámetros de entrada llamados IN1 y IN2 y se escribe en un parámetro de salida llamado OUT. Usted puede crear un caso llamado CALC1 y si quiere escribe el resultado en una variable llamada CALCRESULT.

La sintaxis es la siguiente:

```
CALC1 (IN1:= 55.55, IN2:=66.66);
```

```
CALCRESULT := CALC1.OUT;
```

El siguiente código da igual resultado:

```
<FB_Instance.ParameterIn1>:_ <Value1>;
```

```
<FB_Instance.ParameterIn2>:_ <Value2>;
```

```
<FB_Instance.ParameterIn3>:_ <Value3>;
```

```
<FB_Instance> ( );
```

```
<result> :_ (<FB_Instance.ParameterOut>);
```

Usando el mismo ejemplo citado anteriormente, la sintaxis es la siguiente:

```
CALC1 . IN1:= 55.55
```

```
CALC1 . IN2:= 66.66
```

```
CALC ( );
```

```
CALCRESULT:_ CALC1.OUT;
```

Definición de Funciones

Después que usted crea una nueva función, usted necesita especificar sus parámetros y las variables, en caso de que la función regrese un valor. Usted necesita especificar sus tipos de datos.

Ver apéndice “**EJEMPLOS DE PROGRAMAS**” para ejemplos como designar funciones.

Ajuste de parámetros y variables

Usted define los parámetros de entrada y salida y las variables de una función en el Symbol Manager. Las variables de una función son locales para esa función y no pueden ser relacionadas por cualquier otra POU, excepto dentro del contexto de llamar a una función. Dentro de la función, los parámetros de entrada son solamente leídos. Los parámetros de salida le deben ser asignados valores a través de orden de asignación.

- **Para definir un parámetro o una variable para un tipo de funciones:**
 1. Agregue una función al proyecto. Para información acerca de crear funciones, ver “Adición de una nueva POU al proyecto”(pagina 5-12).
 2. Sobre el menú **Tools**, pulse en la etiqueta **Symbol Manager**(Administrador de Símbolos).
 3. Haga clic sobre el tipo de función a seleccionar como se muestra en la siguiente figura.

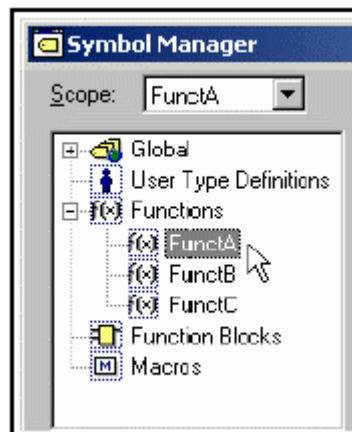


Figura 22. Selección de la función.

4. Pulse en la etiqueta **New** en la barra de tareas del Symbol Manager.



Figura 23. Herramienta para insertar la nueva función.

Aparecerá el cuadro de diálogo **Symbol Properties**(Propiedades de los símbolos).

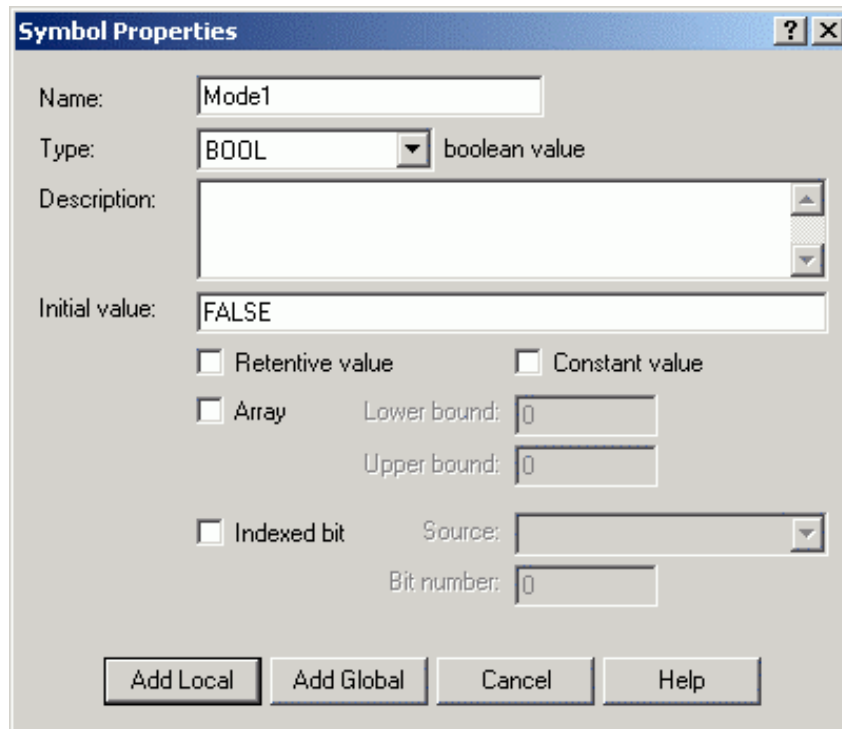


Figura 24. El cuadro Symbol Properties.

5. Ingrese el nombre del parámetro o la variable dentro del campo **Name**(mas de 100 caracteres). Use solamente caracteres alfanuméricos y subrayados.
6. Seleccione el tipo de dato en el campo **Type**.
7. Ingrese la descripción opcional dentro del campo **Description**.
8. Escoja la variable o el tipo de símbolo en el campo **In/Out**, como se muestra abajo.

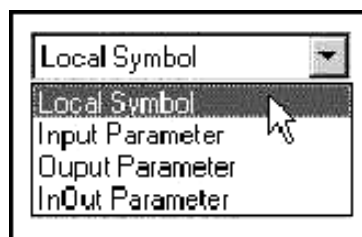


Figura 25. Selección del tipo de variable o símbolo.

9. Si el parámetro o la variable requieren de un valor inicial, ingrese el valor dentro del campo **Initial Value**.
10. Si usted está definiendo los parámetros o variables como un arreglo o serie. Revise el cuadro de comprobación **Array**(arreglo o serie). Esto causa que los campos **Lower Bound**(salto hacia abajo) y **Upper Bound** (salto hacia arriba) permanezcan activos. Ingrese el valor más bajo y el más alto en los campos apropiados.
11. Pulse en la etiqueta **Add Local** para completar la definición de los parámetros o las variables.

Especificación del tipo de dato para el valor de retorno de una función

Usted especifica el tipo de retorno para una función en el Symbol Manager.

➤ **Para especificar el tipo de dato de retorno para una función:**

1. Ingrese al **Symbol Manager**.
2. Haga clic derecho sobre la función y seleccione la etiqueta **Properties**. Aparecerá el cuadro de diálogo **Symbol Properties**.

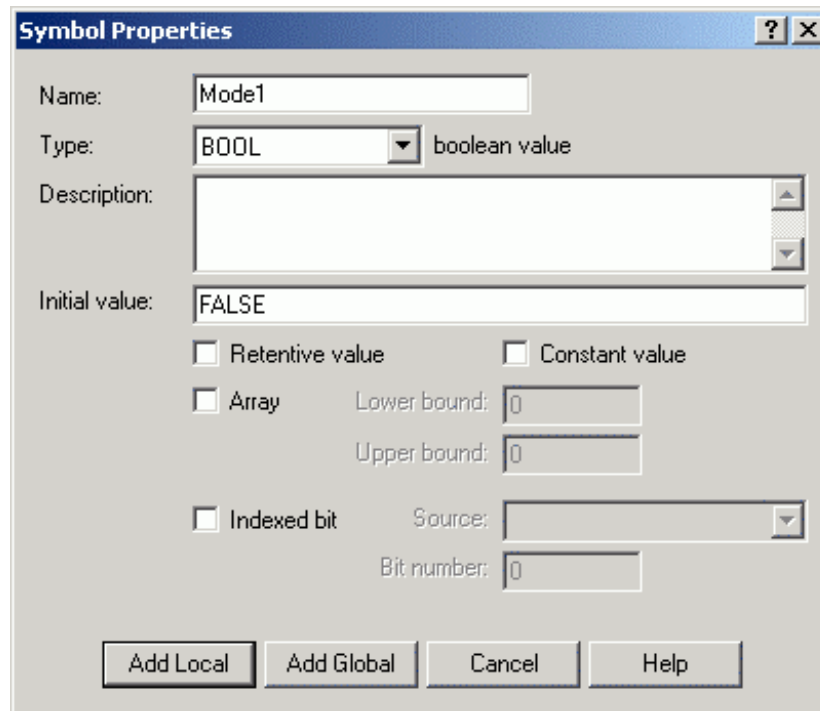


Figura 26. El cuadro Symbol Properties.

3. Para una función, selecciona el tipo de dato en el campo **Return Type** (Tipo de Retorno) como se muestra en la siguiente figura.

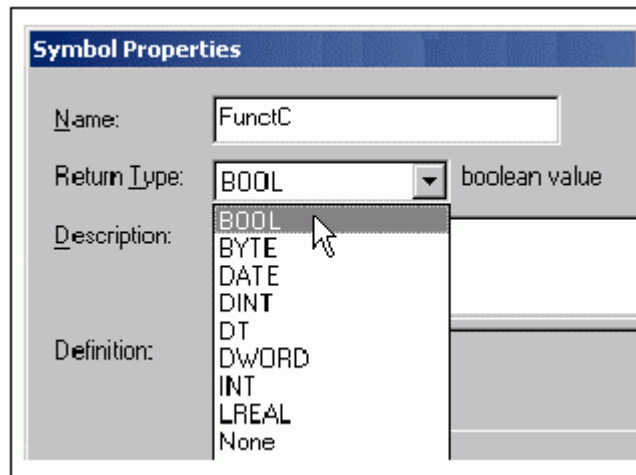


Figura 27. Selección del tipo de dato de retorno.

Para continuar, seleccione la opción **None** (ninguno) como el tipo de retorno. Las funciones solamente regresan tipos de datos simples. Estas no pueden regresar arreglos o series, estructuras o bloques de funciones.

Código para llamar a una función

Usted puede llamar a una función para su ejecución desde cualquier otra POU que este escrita en lenguaje de texto estructurado. El formato para este tipo de funciones es el siguiente:

<Function> (<Parameter1Value>, <Parameter2Value>, <Parameter3Value>,...;

Por ejemplo, usted ha definido una función llamada CALC que esta basada en el calculo de dos parámetros de entrada llamados IN1 Y IN2 y esta escrita en un parámetro de salida llamado OUT. La sintaxis es la siguiente:

CALC1 (IN1:= 55.55, IN2:=66.66, OUT:=CALCRESULT);

Usted puede especificar un valor de retorno para una función. El formato para este tipo de funciones es el siguiente:

<Result>:=<Function> (<Parameter1Value>, <Parameter2Value>,...);

Ajustar el valor de retorno incluyendo el código en la función que asigna el valor para el nombre de la función.

Por ejemplo, usted ha definido una función llamada CALC que esta basada en el calculo de dos parámetros de entrada llamados IN1 Y IN2 y que regresa el valor de un tipo de dato LREAL. La sintaxis de llamar a la función es la siguiente:

```
CALCRESULT := CALC1 (IN1:= 55.55, IN2:=66.66 );
```

El código actual en la función es el siguiente:

```
CALC := IN1 * IN2;
```

Note como usted ajusta el valor de retorno para ubicar el nombre de la función CALC en el lado izquierdo del orden de asignación. Formas alternativas en la sintaxis para llamar a una función le permite usar orden de asignación por parámetros, o por lista de variables o valores literales por parámetros en un orden definido por sus direcciones en el Symbol Manager.

En la siguiente sintaxis, los parámetros reciben sus valores a través de un orden de asignación. En esta forma, los parámetros pueden ser en cualquier orden.

```
(<Parameter1:= Value1>, <Parameter2:= Value2>, <Parameter3:= Value3>,..);
```

En el siguiente ejemplo, los parámetros aparecen en cualquier orden.

```
CALC1 (OUT:= CALCRESULT, IN2:= 5.5, IN1:=6.6 );
```

En la siguiente sintaxis, los parámetros reciben sus valores de acuerdo con el orden de estos en el Symbol Manager.

```
(< Value1>, < Value2>, < Value3>,.. . . );
```

En el siguiente ejemplo, el orden de los parámetros no cambia.

```
CALC ( 55.55, 66.66, CALCRESULT );
```

Usted debe ingresar todos los parámetros. Usted no puede saltarse un parámetro por usar una coma extra. Usted debe usar la misma sintaxis para todos los parámetros usados en la función. Esto es, no usar un orden de asignación para un parámetro y una variable o un valor literal para otros.

Adición y organización de drivers I/O

Cuando usted adiciona drivers I/O al proyecto, los drivers aparecen sobre la carpeta I/O en la Project View. Para organizar los drivers, usted puede crear carpetas para varios drivers sobre la carpeta I/O y ubicar los drivers dentro de esta si es necesario, esto haciendo clic en la carpeta I/O y clic derecho. Seleccione la etiqueta **New Fólder** para crear una carpeta sobre la carpeta I/O.

➤ **Para adicionar un driver I/O al proyecto:**

1. Pulse en la etiqueta **New** sobre el menú **File**.
2. Seleccione la etiqueta Drivers I/O en el cuadro de dialogo **New**.
3. Seleccione el Driver I/O y pulse en **OK**.

Un doble clic sobre un driver I/O para configurarlo. Seleccione un driver y haga clic derecho, para ingresar información descriptiva a cerca del driver en el cuadro de dialogo **Properties**. Además, usted puede con un clic derecho configurar o remover un driver, o excluirlo desde el proyecto cuando usted este cargando el proyecto.

Cuando usted excluye un driver I/O, los símbolos I/O son cargados, pero el driver no es ejecutado. Esta característica es usada para simular I/O (entradas y salidas) sin tener hardware I/O físicamente presente.

Nota: Removiendo una configuración I/O no se borran los archivos desde el disco duro. Para borrar la configuración del driver I/O, revisar el cuadro de comprobación **Delete Associated Files** en el cuadro de dialogo **Remove**.

Cuando usted remueve o borra una configuración de un driver I/O desde la Project View, cualquier variable que este en los puntos de I/O será borrada.

Configuración del Runtime Engine (software de ejecución)

Usted puede ajustar el tiempo de análisis y otros parámetros de ejecución. Tal como recarga automática para el Runtime Engine. Haciendo doble clic sobre el icono **RTEngine** en la Project View, aparecerá el cuadro de dialogo del Runtime Engine.

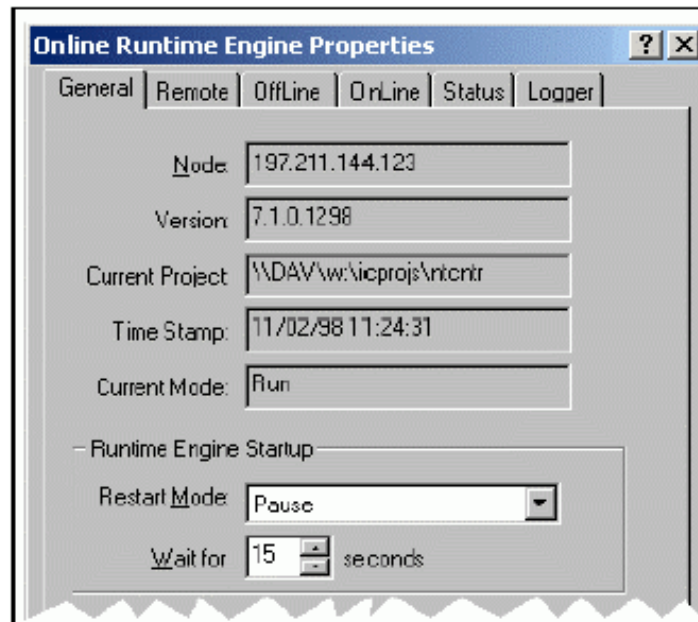


Figura 28. Cuadro de dialogo del Runtime Engine.

Nota: Las etiquetas **Remote**(Remoto), **Online**(En Línea) y **Status**(Estado) no aparecen cuando el ambiente de desarrollo no esta conectado al Runtime Engine. La etiqueta **Logger**(Registrador) no aparece cuando el Runtime Engine esta sobre una plataforma de hardware que no esta corriendo con el sistema operativo Windows NT.

Haga clic derecho sobre el icono **RTEngine** en la Project View para hacer lo siguiente:

- **Connect/Disconnect:** Conecta y desconecta el ambiente de desarrollo y el Runtime Engine.
- **Configure:** Muestra el cuadro de dialogo de configuración **Offline Runtime Engine Properties**, si el ambiente de desarrollo no esta conectado al Runtime Engine. Muestra el cuadro de dialogo **Online Runtime Engine Properties** si el ambiente de desarrollo esta conectado al Runtime Engine.
- **Report faults:** Muestra en una lista las fallas de los programas y las fallas I/O. En la ventana de salida y en el registro de Wonderware(Wonderware Logger).
- **Clear faults :** Depura las fallas o ajusta los programas que fallaron en el modo pausa. Depura las fallas de I/O y elimina los errores en los bits de estado del Runtime Engine, tal como un análisis excesivo del Runtime Engine.
- **Pause:** Ajusta el Runtime Engine en el modo pausa.
- **Single Scan:** Ejecuta un análisis simple del Runtime Engine.
- **Stop:** Ajusta el Runtime Engine en el modo de suspensión o parada.
- **Properties:** Revisa el siguiente código de propiedades:
Ingresa información descriptiva sobre el Runtime Engine en el cuadro de dialogo **Properties**.

Cambia la versión del Runtime Engine. Esta es usada cuando usted desarrolla y examina un proyecto en el computador usando el sistema operativo Windows NT pero intentando cargar y correr el proyecto en un computador que usa otro sistema operativo.

Revisa las siguientes fechas: la fecha cuando el Runtime Engine fue modificado, y la fecha en que el proyecto fue cargado en el Runtime Engine.

Para información más detallada acerca del Runtime Engine, consulte los siguientes capítulos:

“EJECUCIÓN DE UN PROYECTO” y **“ ADMINISTRACIÓN DEL SISTEMA INCONTROL”**.

Ingreso al Symbol Manager (administrador de símbolos)

Con un doble clic sobre el icono **Symbols** en la **Project View** se ingresa al **Symbol Manager**. Usted crea o edita las variables de las POU en el Symbol Manager. Con un clic derecho sobre el icono Symbols se ingresa información descriptiva sobre la configuración de sus símbolos en el cuadro de dialogo **Properties**. El Symbol Manager es como se muestra en la siguiente figura:

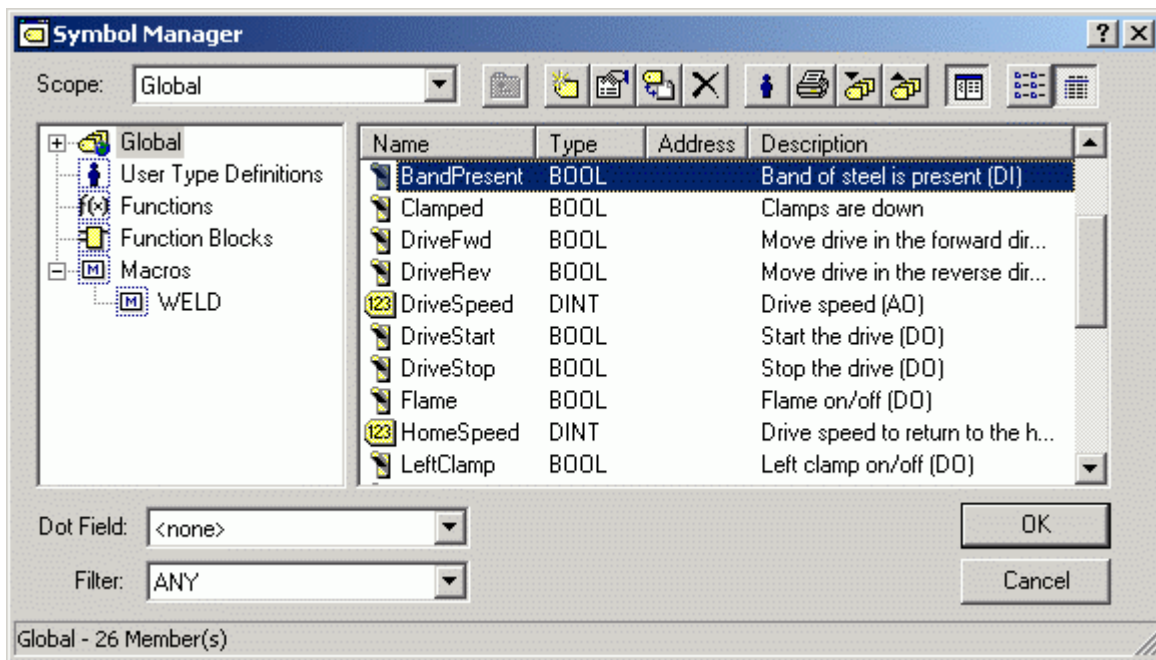


Figura 29. El Symbol Manager.

Para mas información detallada sobre el Symbol Manager, consulte el capítulo “DEFINICIÓN DE VARIABLES”.

Designación de la plataforma del hardware

Algunas versiones de InControl soportan múltiples plataformas de hardware de ejecución. Si usted ha adquirido una de estas versiones, entonces cuando usted creó una nueva versión de un proyecto; El InControl le indicara para seleccionar la plataforma donde usted intentara correr el proyecto. Esto le permite desarrollar un proyecto sobre el ambiente Windows NT, y entonces cargar y correr el proyecto sobre cualquiera plataforma de hardware. Usted puede convertir un proyecto que fue desarrollado por un propósito y correrlo para otro propósito.

➤ **Para cambiar el destino del Runtime Engine:**

1. Seleccione el icono **Runtime Engine** en la Project View y pulse clic derecho, aparecerá el cuadro de dialogo **Properties** del Runtime Engine.

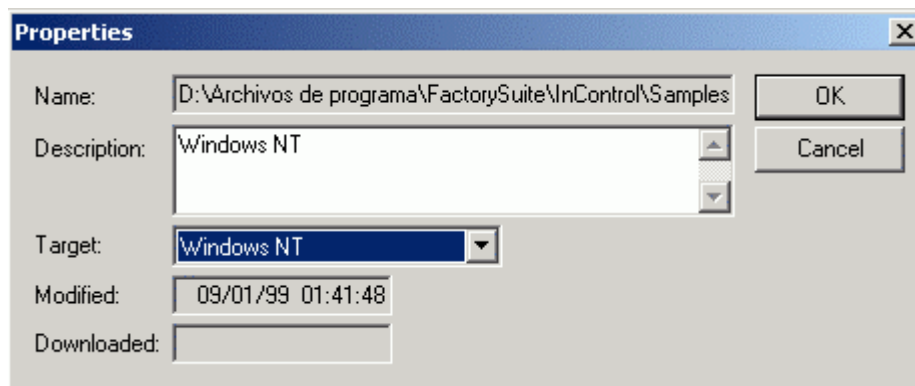


Figura 30. Cuadro de propiedades del Runtime Engine.

2. La versión actual es mostrada en el campo **Target**. Seleccione la nueva versión y haga clic en **Ok**.

Nota: Algunas tarjetas de entrada y salida (I/O) pueden no convertirse si ellas no soportan la versión del sistema operativo. Después de esta conversión, estas tarjetas son desactivadas en la ventana de proyectos.

Cambio en la prioridad y el orden de ejecución del programa

Use la Execution View para especificar el nivel de prioridad de los programas (Análisis Normal, Prioridad Baja), y cambiar el orden en el cual ellos son ejecutados. Si usted no fija el orden, los programas serán ejecutados en el orden en el cual usted los creó. Para cambiar el orden o la ejecución, haga clic sobre el nombre del programa y arrástrelo a otro lugar en la lista de programas. En la siguiente figura RLL5 ha sido movido mas arriba de STL1.

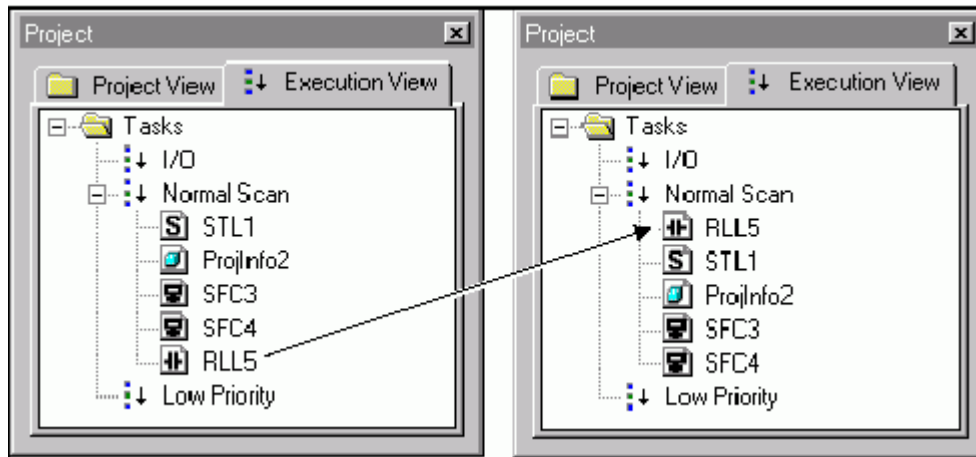


Figura 31. Cambio del orden o la ejecución de un programa.

Para cambiar la prioridad, haga clic sobre el nombre del programa y arrástrelo a otro lugar en la lista de programas. En la siguiente figura, **SFC4** ha sido asignado a Prioridad Baja.

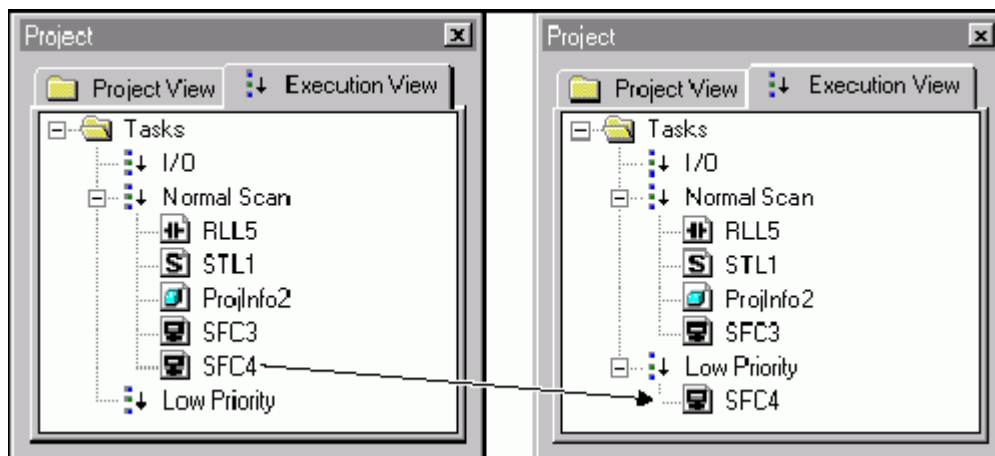


Figura 32. Cambio en la prioridad de un programa.

Para mas información sobre el nivel de prioridad, consulte el tema “LÍNEA DE TIEMPO DEL RUNTIME ENGINE” del capítulo “ADMINISTRACIÓN DEL SISTEMA INCONTROL”.

DEFINICIÓN DE VARIABLES

Este capítulo describe como definir los símbolos que usted usara en sus programas de aplicación.

CONTENIDO

- INTRODUCCIÓN
- TIPOS DE DATOS
- SYMBOL MANAGER (ADMINISTRADOR DE SÍMBOLOS)
- CREACIÓN DE UNA VARIABLE
- CREACIÓN DE UN ARREGLO DE VARIABLES
- ASIGNACIÓN DE UN NOMBRE A UN BIT EN UNA VARIABLE
- CREACIÓN UN TIPO DE DATO DEFINIDO POR EL USUARIO
- PRESENTACIÓN DE INFORMACIÓN DE VARIABLES
- VARIABLES DE SISTEMA
- TRANSFERENCIA DE BASES DE DATOS DE SÍMBOLOS

This product is licensed to:
 Company: N/A
 SN: N/A
 Locked To: N/A
 7.1.0.1337

INTRODUCCIÓN

Las herramientas de programación de InControl le permiten definir y usar variables, las cuales están localizadas en la memoria interna que contiene los datos del proyecto. El contenido de la información esta definido por el tipo de dato y pueden ser números reales, números enteros, cadenas de caracteres, etc. Use el **Symbol Manager** para definir una variable, asignándole un nombre de un simbólico y un tipo de dato, y designándolo en el ámbito local o global.

Definición de variables

Use solamente caracteres alfanuméricos y subrayados para el nombre de una variable. Los nombres pueden comenzar con un carácter alfabético o uno subrayado, pero no un carácter numérico. La longitud máxima recomendada para el nombre de una variable es de 100 caracteres.

Variables locales y globales

Las variables pueden ser de alcance local o global.

- **Variable global:** Una variable global puede ser usada y referenciada por todos los proyectos dentro de todos los programas, incluyendo los objetos de fabrica InControl (FOE's). usted puede usar variables globales dentro de la **Suitelink** (Paquetes Integrados de Enlace) y operaciones **DDE** (Intercambio Dinámico de Datos), y usted puede referenciar los puntos I/O como variables globales.

• **Variables locales:** Una variable local es usada y referenciada solamente dentro del programa en cual esta definida. Excepto para variable locales definidas para funciones, usted puede referenciar una variable local a través de **Suitelink** (Paquetes Integrados de Enlace) y operaciones **DDE** (Intercambio Dinámico de Datos). Para ayudar a coordinar los programas de ejecución, usted puede referenciar las siguientes variables locales de sistema en otros programas:

Mode: Contiene el modo actual de un programa. Use esta sintaxis para referenciar la variable Mode:

<Nombre del programa> . Mode

Variables FOE's: los FOE's y otros controles ActiveX adicionan variables locales automáticamente cuando estos son instalados. Usted no puede editar estas variables desde dentro del **Symbol Manager**. Use esta sintaxis para referenciar las variables FOE's:

<Nombre del FOE> . <Nombre de la variable>

Para más información acerca de usar la SuiteLink y DDE para referenciar variables locales y globales, ver el apéndice "**Monitoreo de datos por DDE/SuiteLink**" y la "**Guía de Usuario SuiteLink de Wonderware InControl**".

En contraste para las variables, que son localizaciones internas de memoria, los puntos de I/O son localizaciones externas de memoria, asociadas con puntos físicos. Porque usted puede referenciar a estos en un programa al igual que las variables. Los puntos de entrada son abarcados como las variables locales: Usted puede referenciarlos desde cualquier programa, y usted puede usarlos en operaciones de **SuiteLink**. Usted no puede editar un punto de entrada y salida desde el **Symbol Manager**.

VARIABLES ASIGNADAS CON UN VALOR CONSTANTE

Revise el cuadro de comprobación **Constant Value** en el cuadro de dialogo **Symbol Properties** de las variables, si usted quiere definir una variable nombrada y darle un valor constante. El cuadro de dialogo **Symbol Properties** es mostrado a continuación.

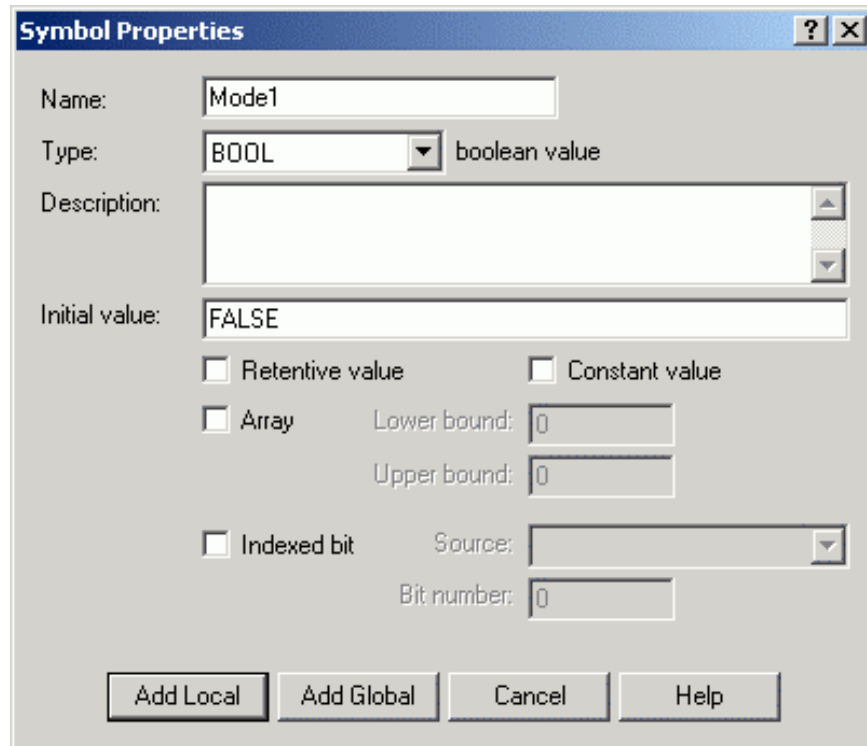


Figura 1. Asignación de un valor constante a una variable.

Asignar un valor constante a una variable que usted necesita, tal como **pi** (π), adquiere un valor literal invariable al ser usado en su programa. Para algunas variables, tales como ingredientes de formulas, usted puede necesitar probar un numero de valores hasta que usted determine uno para usar en la variable.

En este caso, usted puede ingresar al cuadro de dialogo **Symbol Properties** para las variables, asignar un valor inicial basado en sus pruebas, y entonces revisar el cuadro de comprobación **Constant Value**.

- Si usted adiciona una variable que ha sido definida como una constante en la ventana de Observación, la variable aparecerá con un valor resaltado. Usted no puede modificar la variable en la ventana de Observación.
- Usted no puede directamente monitorear un variable, que tiene un valor constante, desde una aplicación externa, tal como InTouch. Asigne el valor de una variable a otra variable que no es constante, y entonces podrá monitorear la segunda variable.

VARIABLES RETENTIVAS

Revise el cuadro de comprobación **Retentive Value** del cuadro de dialogo **Symbol Properties** de las variables si usted quiere la opción, de hacer una copia de seguridad del valor de una variable en el disco duro. El cuadro de dialogo **Symbol Properties** es mostrado a continuación.

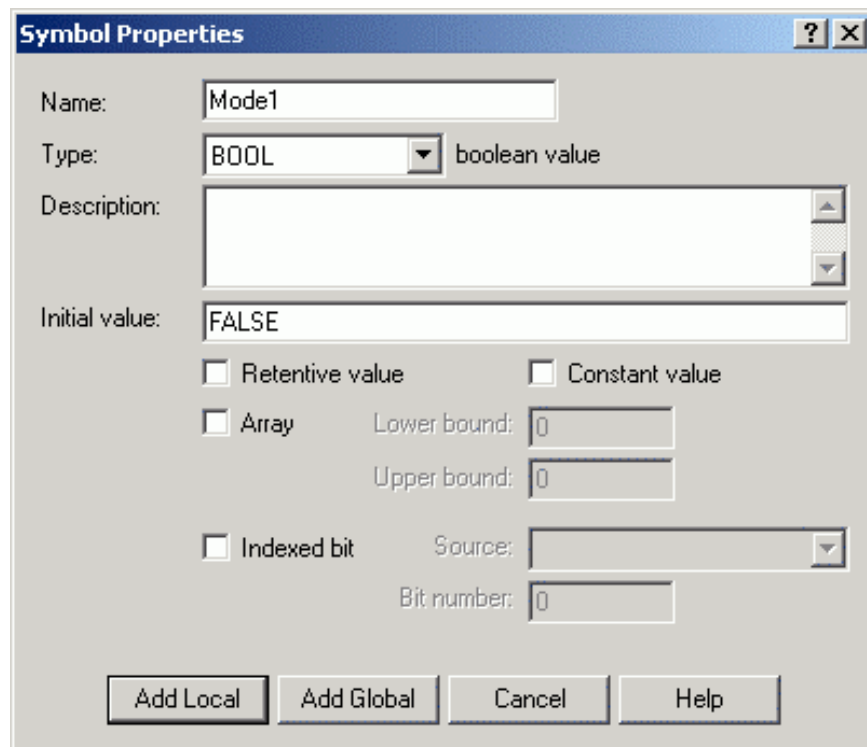


Figura 2. Asignación de un valor retentivo a una variable.

El InControl suministra tres maneras por las cuales usted puede especificar para que la copia de seguridad ocurra.

- Si el Runtime Engine (software de ejecución) se cierra durante una falla de energía, el valor de la variable retenida es copiado en el disco duro. Cuando el software de ejecución (Runtime Engine) se reinicia, este valor es restaurado en la variable. Note que los valores de cualquier variable que haya sido forzada o utilizada, también son guardados durante la falla de energía.

Los valores de las variables retenidas y aplicadas no son guardados a menos que usted este usando una UPS inteligente con el sistema y que usted haya configurado al InControl para que señale la falla de energía.

Para mas información a cerca de preparar las fallas de energía, ver “**MANEJO DE FALLAS DE POTENCIA**” en el capítulo “**ADMINISTRACIÓN DEL SISTEMA INCONTROL**”.

- Usted puede configurar al InControl para guardar los valores de las variables retenidas y forzadas en el disco duro periódicamente. La frecuencia de fallo de cero desactiva esta característica. Usted puede cambiar esto en el cuadro de dialogo **Runtime Engine Properties**.

Se llevan dentro de una cuenta el número de variables retenidas en su proyecto cuando usted escoge la frecuencia de actualización. Un corto intervalo de actualización puede degradar el funcionamiento de su sistema cuando un gran número de variables son señaladas como retenidas.

- Usted puede diseñar un código para guardar el valor de las variables retenidas y forzadas.

Los valores son restaurados cuando usted recarga el proyecto o si usted ha configurado el Runtime Engine (software de ejecución) para reiniciarse automáticamente (modo pausa o arranque) después de reiniciar su sistema.

TIPOS DE DATOS DE LAS VARIABLES

La siguiente tabla muestra los tipos de datos de la especificación IEC-61131-3 que son soportados por el InControl. Individualmente estos tipos de datos son descritos en la siguiente página.

Tipos de datos y categorías

Todos los tipos	Grupo	Subgrupo	Tipo de Dato
ANY	ANY_NUM	ANY_REAL	LREAL
			REAL
		ANY_INT	<u>DINT</u>
			<u>INT</u>
			SINT
			DWORD
			WORD
			BYTE
	ANY_BIT	DWORD	
		WORD	
		BYTE	
		BOOL	
	ANY_DATE	DT (fecha y tiempo)	
		DATE	
		TOD	
			STRING
			TIME
		FILE ¹	
		TMR ¹	
		User-Defined	
1	Mejora de la especificación IEC 61131-3.		

Advertencia: la especificación IEC-61131-3 no soporta la combinación de números registrados y no registrados (Tipos de Datos **ANY_NUM**) en cálculos matemáticos, los resultados de estas operaciones matemáticas pueden no ser los que usted espera, lo cual puede tener un potencial de riesgo de muerte y daño a los equipos. Evitar usar estas expresiones que combinan números registrados y no registrados.

LREAL

El tipo de dato LREAL es parte del tipo de datos ANY_REAL. El tipo de dato LREAL son validos en cualquier instrucción o función de bloque que acepte un tipo de datos ANY, ANY_NUM, ANY_REAL o REAL. LREAL es un tipo de dato numérico de un valor de 64 bits compuesto de uno o mas dígitos (0-9), que esta registrado, y contiene un punto decimal. El rango para lo números LREAL es el siguiente: -1.79769313486231 E308 (negativo) hasta +1.79769313486231 E308 (positivo), incluyendo el cero.

El formato de la IEEE el usado para representar los tipos de datos LREAL.

Nota: Cuando usted se comunica con el software de ejecución usando la interfase Suitelink/DDE, los tipos de datos LREAL de 64-bits son transmitidos con una precisión de 32-bits.

REAL

El tipo de dato REAL es parte del tipo de datos ANY_REAL. El tipo de dato REAL es valido en cualquier instrucción o función de bloque que acepte un tipo de datos ANY, ANY_NUM, ANY_REAL o REAL. REAL es un tipo de dato numérico de un valor de 32 bits, compuesto de uno o mas dígitos (0-9), que esta registrado, y contiene un punto decimal. El rango para los números REAL es el siguiente: -3.402823 E38 (negativo) hasta +3.402823 E38 (positivo), incluyendo el cero.

El formato de la IEEE es el usado para representar los tipos de datos REAL.

DINT

El tipo de dato DINT es parte del tipo de datos ANY_INT. El tipo de dato DINT es valido en cualquier instrucción o función de bloque que acepte un tipo de datos ANY, ANY_NUM, ANY_INT o DINT. DINT es un número entero, que esta compuesto de uno o mas dígitos (0-9), y no puede contener un punto decimal. El DINT tiene una longitud de 32-bits y tiene un rango de: -2147483648 hasta +2147483647.

INT

El tipo de dato INT es parte del tipo de datos ANY_INT. El tipo de dato INT es valido en cualquier instrucción o función de bloque que acepte un tipo de datos ANY, ANY_NUM, ANY_INT o INT. INT es un número entero, que esta compuesto de uno o mas dígitos (0-9), y no puede contener un punto decimal. El INT tiene una longitud de 16-bits y tiene un rango de: -32768 hasta +327683647.

SINT

El tipo de dato SINT es parte del tipo de datos ANY_INT. El tipo de dato SINT es valido en cualquier instrucción o función de bloque que acepte un tipo de datos ANY, ANY_NUM, ANY_INT o SINT. SINT es un número entero muy pequeño, que esta compuesto de uno o mas dígitos (0-9), y no puede contener un punto decimal. El SINT tiene una longitud de 8-bits y tiene un rango de: -128 hasta +127.

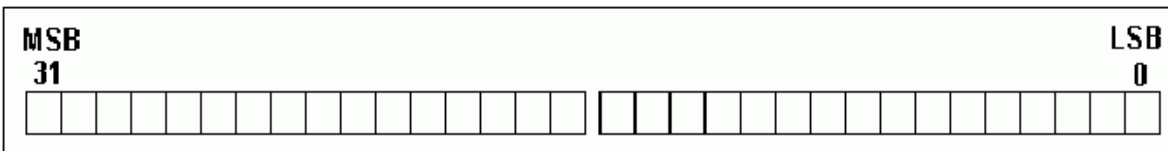
Números enteros sin registro

Los tipos de datos UDINT, UINT y el USINT son equivalentes a los tipos de datos DWORD, WORD y BYTE respectivamente. El InControl mejoro el tipo de dato ANY_BIT haciendo a UDINT, UINT y USINT innecesarios.

DWORD

El tipo de dato DWORD es parte del tipo de datos ANY_BIT y ANY_INT. El tipo de dato DWORD es valido en cualquier instrucción o función de bloque que acepte un tipo de datos ANY, ANY_BIT o DWORD. DWORD es un número entero sin registro, esta compuesto de uno o mas dígitos (0-9), y no puede contener un punto decimal. El DWORD tiene una longitud de 32-bits y tiene un rango de: 0 hasta +4294967295.

El formato de DWORD es mostrado en la siguiente figura:

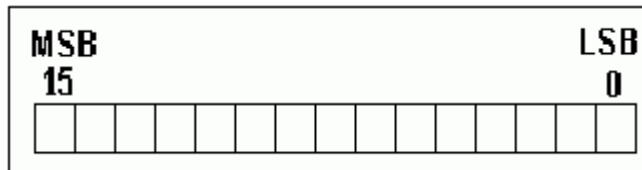


Usted puede referenciar un bit individual dentro de una variable DWORD. Ver **“Asignación de un nombre a un bit en una variable”**(pagina 6-13).

WORD

El tipo de dato WORD es parte del tipo de datos ANY_BIT y ANY_INT. El tipo de dato WORD es valido en cualquier instrucción o función de bloque que acepte un tipo de datos ANY, ANY_BIT o WORD. WORD es un número entero sin registro, esta compuesto de uno o más dígitos (0-9), y no puede contener un punto decimal. El WORD tiene una longitud de 16-bits y tiene un rango de: 0 hasta 65535.

El formato de WORD es el mostrado en la siguiente figura:

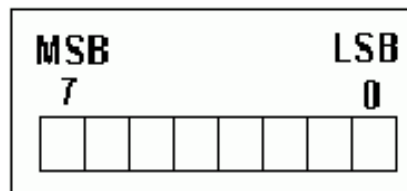


Usted puede referenciar un bit individual dentro de una variable WORD. Ver **“Asignación de un nombre a un bit en una variable”**(pagina 6-13).

BYTE

El tipo de dato BYTE es parte del tipo de datos ANY_BIT y ANY_INT. El tipo de dato BYTE es valido en cualquier instrucción o función de bloque que acepte un tipo de datos ANY, ANY_BIT o BYTE. Un BYTE es un número entero sin registro, esta compuesto de uno o mas dígitos (0-9), y no puede contener un punto decimal. Un BYTE tiene una longitud de 8-bits y tiene un rango de: 0 hasta 255.

El formato de BYTE es el mostrado en la siguiente figura:



Usted puede referenciar un bit individual dentro de una variable BYTE. Ver “**Asignación de un nombre a un bit en una variable**”(pagina 6-13).

BOOL

El tipo de dato BOOL es parte del tipo de datos ANY_BIT. El tipo de dato BOOL es valido en cualquier instrucción o función de bloque que acepte un tipo de datos ANY, ANY_BIT o BOOL. Un BOOL tiene una longitud de 1-bit y puede tener uno de dos valore posibles: TRUE (1, o ON) o FALSE (0, o OFF).

DT

El tipo de dato DT es parte del tipo de datos ANY_DATE. El tipo de dato DT es valido en cualquier instrucción o función de bloque que acepte un tipo de datos ANY, ANY_DATE o DT.

El formato de DT es el siguiente:

DATE_AND_TIME | date_and_time | DT | dt#YYYY-MM-DD-HH:MM: S.S, donde YYYY (100-2100) es el año, MM (1-12) es el mes, DD (1-31) es el día del mes, HH (0-23) es la hora, MM (0-59) son los minutos, y S.S (0.0-59.0) son los segundos.

Si usted crea una expresión de tipos de datos DT, las entradas y las salidas deben ser los tipos de datos mostrados en la tabla “**Usando tipos de datos basados en tiempo y fecha en expresiones**”(pagina 6-13).

DATE

El tipo de dato DATE es parte del tipo de datos ANY_DATE. El tipo de dato DATE es valido en cualquier instrucción o función de bloque que acepte un tipo de datos ANY, ANY_DATE o DATE.

El formato de DATE es el siguiente:

DAT | date | D | d#YYYY-MM-DD, donde **YYYY** (100-2100) es el año, **MM** (1-12) es el mes, **DD** (1-31) es el día del mes.

Si usted crea una expresión con los tipos de datos DATE, las entradas y las salidas deben ser con los tipos de datos mostrados en la tabla “**Usando tipos de datos basados en tiempo y fecha en expresiones**”. La variable de sistema TODAY es un tipo de dato DATE que contiene la fecha actual del sistema y que puede ser usada para determinar cuando un evento tuvo lugar. Estos operadores pueden ser usados con TODAY: EQ, LT, GT, LE, GE, y NE. Para leer el valor de TODAY, use el orden de asignación o el comando MOVE para mover el valor hacia una variable.

TOD

El tipo de dato TOD es parte del tipo de datos ANY_DATE. El tipo de dato TOD es valido en cualquier instrucción o función de bloque que acepte un tipo de datos ANY, ANY_DATE o TOD.

El formato de TOD es el siguiente:

TIME_OF_DAY | time_of_day | TOD | tod#HH:MM: S.S, donde **HH** (0-23) es la hora, **MM** (0-59) son los minutos, **S.S** (0.0-59.0) son los segundos.

La variable de sistema NOW es un tipo de dato TOD que contiene el tiempo actual del sistema y que puede ser usada para determinar cuando un evento tuvo lugar. Estos operadores pueden ser usados con TODAY: EQ, LT, GT, LE, GE, y NE. Para leer el valor de NOW, use el orden de asignación o el comando MOVE para mover el valor hacia una variable. Si usted crea una expresión con datos TOD, los tipos de datos de entrada y salida deben ser los mostrados en la tabla “**Usando tipos de datos basados en tiempo y fecha en expresiones**”(pagina 6-13).

Uso de tipos de datos basados en tiempo y fecha en expresiones

Si usted crea una expresión con tipos de datos ANY_DATE, las entradas y las salidas deben ser los mostrados en la siguiente tabla.

Tipos de datos basados en tiempo usados en expresiones

Operación	Entrada1	Entrada2	Salida
Adición	TIME	TIME	TIME
Adición	TIME_OF_DAY	TIME	TIME_OF_DAY
Adición	DATE_AND_TIME	TIME	DATE_AND_TIME
Adición	DATE	TIME	DATE
Adición	DATE	TIME_OF_DAY	DATE_AND_TIME
Resta	TIME	TIME	TIME
Resta	DATE	TIME	DATE
Resta	DATE	DATE	TIME
Resta	TIME_OF_DAY	TIME	TIME_OF_DAY
Resta	TIME_OF_DAY	TIME_OF_DAY	TIME
Resta	DATE_AND_TIME	TIME	DATE_AND_TIME
Resta	DATE_AND_TIME	DATE_AND_TIME	TIME
Resta	DATE_AND_TIME	DATE	TIME_OF_DAY
Resta	DATE_AND_TIME	TIME_OF_DAY	DATE

TIME

El tipo de dato TIME es parte del tipo de datos ANY. El tipo de dato TIME es valido en cualquier instrucción o función de bloque que acepte un tipo de datos ANY o TIME. TIME es una variable que representa la duración del tiempo.

El formato de TIME es el siguiente:

TIME | **time** | **T** | **t#** seguido por una secuencia de uno o mas números y unidades de tiempo específicas.

Las unidades de tiempo específicas, rangos y ejemplos de sus convenciones son mostrados abajo. Usted puede separar especificaciones con un carácter subrayado: t:5m_45s.

- **D** o **d** = Días (0-1000000) **T#D2h**: 1 día y 2 horas
- **H** o **h** = Horas (0-23) **t#20h**: 20 horas
- **M** o **m** = Minutos (0-59) **t#5m45s**: 15 minutos y 45 segundos
- **S** o **s** = Segundos (0-59) **t#26S200MS**: 26 segundos y 200 ms
- **MS** o **ms** = Milisegundos (0-999) **T#45.325ms**: 45.325 milisegundos

Cuando una variable del tipo de dato TIME es convertida hacia un tipo de dato ANY_NUM, el valor es convertido en un numero de segundos. Por ejemplo, cuando 3 minutos y 24 milisegundos es convertido a un numero REAL, el valor es 180.024 segundos.

Usted puede usar equivalentes decimales de días, horas, etc., pero solamente por la unidad menos significativa puede ser fraccionada. Por ejemplo, **t#1.5d** es equivalente a **t#1d12h**. **t#1d1.5h** es equivalente a **t#1d1h30min**. Sin embargo, **t#1.5d6h** no es valido.

Si usted crea una expresión con tipos de datos **TIME**, los tipos de datos de entrada y salida deben ser los mostrados en la tabla “**Uso de tipos de datos basados en tiempo y fecha en expresiones**” (pagina 6-13).

TMR

El tipo de dato TMR es parte del tipo de datos ANY. El tipo de dato TMR es valido en cualquier instrucción o función de bloque que acepte un tipo de datos ANY o TMR. El TMR posee cuatro variables de sistemas, las cuales se identifican por el nombre del timer o contador más una extensión:

- **Tmr_name.PT** contiene el valor del tiempo a programar y es un tipo de dato **TIME**. Esta es una variable retentiva (su valor es retenido durante una falla de energía). Para mas información sobre variables retentivas, ver “**Variables Retentivas**”(pagina 6-5).

Usted puede especificar el valor inicial de esta variable de la siguiente manera: Ingrese un valor en el cuadro de dialogo **Symbol Properties** en el **Symbol Manager** (Administrador de Símbolos).

Use el orden de asignación (Programación de Texto Estructurado) o la función de bloque MOVE (Programación RLL) para asignar un valor inicial a esta variable.

- **Tmr_name.EN** arranca y para el TMR y es un tipo de dato **BOOLEAN**.
- **Tmr_name.ET** contiene el tiempo del TMR en segundos y es un tipo de dato **TIME**.
- **Tmr_name.Q** representa el TMR de salida y es un tipo de dato **BOOLEAN**.

La operación de TMR es la siguiente:

- Cuando la transición del **tmr_name.EN** va de **FALSE** o **TRUE**, el **tmr_name.ET** esta ajustado a cero, el **tmr_name.Q** esta ajustado a **FALSE**, y el contador comienza a medir el tiempo.
- Cuando el **tmr_name.ET** es igual al **tmr_name.PT**, entonces el **tmr_name.EN** esta ajustado en **FALSE**, el **tmr_name.Q** esta ajustado en **TRUE**. Usted puede diseñar un programa para resetear el **tmr_name.EN** primero que el tiempo a programar.
- Si el **tmr_name.EN** es conservado en **TRUE**, por un orden de asignación, por ejemplo, el **tmr_name.Q** será **TRUE** por la duración de una exploración dentro de un ciclo de periodo del **tmr_name.PT**.
Usted puede usar el **tmr_name.Q** con una condición **IF** para causar que el código se ejecute cíclicamente. El ciclo se mantendrá por aproximadamente una exploración.
- Si **tmr_name.EN** se ajusta en **FALSE**, el **tmr_name.ET** se mantendrá en el ultimo valor y **tmr_name.Q** permanecerá en **FALSE**.
- En el tiempo transcurrido **tmr_name.ET** puede ser leído cuando quiera.

Nota: Los Contadores evalúan el tiempo transcurrido y no son afectados por ajustar un programa o por que el Runtime Engine este en modo pausa.

FILE

El tipo de dato FILE es parte del tipo de datos ANY. FILE es una estructura que esta diseñada solamente para variables de control de archivos usadas en funciones RLL y archivos de Texto Estructurado.

El tipo de dato FILE posee varias variables de sistemas, las cuales se identifican por el nombre de la función de bloque de control más una extensión. Tres de estas variables especifican el estado de un archivo después que este es abierto: lectura/escritura, si el dato puede ser anexado, y si otras aplicaciones pueden ingresar al archivo. Ocho variables proveen significados de los errores de monitoreo, si un archivo esta en uso, o cuando una operación se completo, etc.

Las tres variables de entrada son mostradas en la siguiente tabla.

Variables de entrada de control de archivos

Variable	Descripción
fcb.ACCESS¹	El Byte de la variable, especifica el estado lectura-escritura del archivo después de que es abierto. 0 = (por defecto) El archivo esta abierto para operaciones de lectura- escritura. 1 = El archivo esta abierto solamente para operaciones de lectura. 2 = El archivo esta abierto solamente para operaciones de escritura.
fcb.APPEND¹	Las variables Booleanas especifican si un dato puede ser anexado a un archivo después de abierto. Solo es valido cuando el archivo esta abierto en el estado de escritura. Esto es, la variable ACCES = 0 o 2. TRUE = El dato será anexado al archivo. FALSE = El dato no puede ser anexado al archivo.
fcb.SHARE¹	El Byte de la variable especifica como otras aplicaciones pueden entrar al archivo cuando esta abierto. 0 = (por defecto) otras aplicaciones pueden entrar al archivo para operaciones de lectura-escritura. 1 = Otras aplicaciones pueden entrar al archivo solamente para operaciones de lectura. 2 = Otras aplicaciones pueden entrar al archivo solamente para operaciones de escritura. 3 = Otras aplicaciones no pueden entrar al archivo.

¹ Estas variables son leídas y toman efecto solamente cuando las funciones **STL OPENFILE** y **NEWFILE** o las funciones de bloque **RLL FOPEN** y **FNEW** son ejecutadas.

Las ocho variables que manejan las operaciones de los archivos son mostradas en la siguiente tabla.

Variables de salida de control de archivos

Variable	Descripción
fcb.BUSY	Variable booleana que indica que el archivo esta siendo utilizado. El sistema ajusta la variable Control de Archivos Activos en TRUE cuando el archivo esta siendo utilizado por otra función de archivos. Si usted intenta ejecutar un tipo de función de archivos mientras esta la variable en TRUE , ocurrirá un error (código de error 15).
fcb.CLSD	Variable booleana que indica que el archivo ha sido cerrado. El sistema ajusta la variable Cerrar Archivos en TRUE cuando ha sido cerrado el archivo.
fcb.EFLAG	Variable booleana que indica cuando un error ocurre. Si un error ocurre durante la operación de un archivo, el sistema ajusta la variable de Error de Archivos en TRUE , esta variable no se resetea automáticamente; el programa debe resetear la variable. Usted puede también resetearla manualmente a través de la Ventana de Observación. Un tipo de función de archivos no puede ser ejecutado mientras esta variable este en TRUE . El programa no pasara al modo de fallo cuando ocurra un error.
fcb.EOG	Variable booleana que indica que el sistema ha encontrado un archivo de Terminación de Ficheros. El sistema ajusta la variable Terminación de Ficheros en TRUE cuando encuentra el EOF .
fcb.ERR	Variable número entero que contiene el código del error que ha ocurrido. Si un error ocurre durante la operación de un archivo, el sistema guarda el código en el Número Entero de Códigos de Errores de Archivos. La tabla que sigue muestra la lista de códigos de errores.
fcb.OPEN	Variable booleana que indica que el archivo ha sido abierto. El sistema ajusta la variable Apertura de Archivos en TRUE cuando ha sido abierto el archivo.
fcb.RDN	Variable booleana que indica que una operación de lectura ha sido concluida. El sistema ajusta la variable Finalización de la lectura de Archivos en TRUE cuando la operación de lectura se ha terminado.
fcb.WDN	Variable booleana que indica que una operación de escritura ha sido concluida. El sistema ajusta la variable Finalización de la escritura de Archivos en TRUE cuando la operación de escritura se ha terminado.

STRING

El tipo de dato STRING es parte del tipo de datos ANY. El tipo de dato STRING es valido en cualquier instrucción o función de bloque que acepte un tipo de datos ANY o STRING.

El formato de STRING consiste de una serie de caracteres ASCII en comillas sencillas. Por ejemplo 'Esta serie es valida'. La longitud máxima del tipo de dato STRING es 1024 caracteres. Una serie esta terminada por un carácter nulo en aquellas operaciones con funciones STL y RLL excepto para conversiones serie/arreglo. Para más información ver los capítulos "**ELEMENTOS DE PROGRAMAS RLL**" y "**LENGUAJE DE TEXTO ESTRUCTURADO**".

El InControl interpreta el símbolo \$ seguido por dos dígitos hexadecimales, encerrados entre comillas sencillas, como la representación hexadecimal del código de caracteres de 8 bits.

Por ejemplo: '\$41 \$42 \$43' es interpretado como **A B C**.

Para designar caracteres especiales en una serie, estos deben estar precedidos del signo peso "\$", como se muestra en los siguientes ejemplos:

- Signo Peso = '\$\$'
- Comilla sencilla = '\$' '
- Línea nueva = '\$N' o '\$n'
- Suministro de Datos = '\$P' o '\$p'
- Comilla doble = '\$" ' '
- Carácter de retorno = '\$R' o '\$r'
- Alimentación de Línea = '\$L' '\$I'
- Etiqueta = '\$T' o '\$t'

USER-DEFINED

El tipo de dato USER-DEFINED es parte del tipo de datos ANY. USER-DEFINED es valido en cualquier instrucción o función de bloque que acepte un tipo de datos ANY o USER DEFINED.

El tipo de dato USER-DEFINED es una estructura que representa un conjunto de datos predefinidos (INTEGER, REAL, otros tipos de datos USER-DEFINED, etc.), los cuales son llamados miembros. Para cada estructura de USER-DEFINED, usted puede especificar el número de miembros y el tipo de dato para cada una. Los tipos de datos TMR y FILE, y funciones y funciones de bloques no son miembros validos para el tipo de dato USER-DEFINED.

Los miembros no deben ser iguales, por ejemplo, usted puede crear un tipo de dato USER -DEFINED llamado Device_Status, compuesto por tres miembros llamados Running, Stopped, y Speed. Running y Stopped son tipos de datos BOOL, y Speed es un tipo de dato REAL. Usted puede entonces definir una variable, por ejemplo: Motor1, el cual es un tipo de dato Device_Status, y este automáticamente tiene tres miembros asociados Motor1.Running, Motor1.Stopped, y Motor1.Speed. En un programa RLL, una bobina llamada Motor1.Running controla el arranque del Motor1, una bobina llamada Motor1.Stopped para el Motor1; y la velocidad del Motor1 esta determinada por el valor contenido en Motor1.Speed.

Si usted quiere Device_Status puede tener un código asociado con este, definido como una función de bloque. Para mas información, ver “**BLOQUES DE FUNCIONES**” en el capitulo “**MANEJO Y ORGANIZACIÓN DE PROYECTOS**”.

Conversión de tipos de datos

Cuando diversas operaciones con tipos de datos son encontradas durante la compilación de un programa, el compilador debe convertir los tipos de datos en un grandísimo tipo numérico. Advertencias deben ser generadas cuando el compilador convierte los tipos de datos. Usted puede evitar algunas de estas advertencias usando funciones de conversión. Al intentar mezclar tipos de datos incompatibles cuando la conversión no es válida genera un mensaje de error.

La conversión es manejada para este tipo de datos:

- BYTE**
- WORD**
- DWORD**
- DINT**
- INT**
- SINT**
- REAL**
- LREAL**
- BOOL** (Alcance limitado)

Más de una función de bloque acepta una variedad de tipos de datos de entrada y salida y convierte los que necesita. La conversión trabaja operación por operación y debe ocultar los resultados intermedios que estén fuera del rango, si estos son significativos, entonces usted necesita convertir uno o ambos operándos en un gran tipo de dato numérico. Por ejemplo, si usted usa la función de bloque RLL ADD para adicionar dos WORDs, la suma debe requerir un DWORD. Por que el resultado no se ajusta a un WORD. En este caso cambiar uno o ambos operándos en DWORDs producirá un valor diferente. Es necesario convertir uno o ambos WORDs en DWORDs antes de sumarlos.

SYMBOL MANAGER (ADMINISTRADOR DE SÍMBOLOS)

Use el Administrador de Símbolos para crear o editar variables de programa. Usted puede entrar al Administrador de Símbolos desde varios puntos del desarrollo del programa, algunos de los cuales se describen abajo. Los campos y botones del Administrador de Símbolos son descritos en la tabla Cuadro de Dialogo del Administrador de Símbolos.

➤ **Para entrar al Administrador de Símbolos:**

1. Sobre el menú **Tools**, haga clic en **Symbol Manager**.



Figura 3. Ejecución del Symbol Manager desde la barra Tools.

2. En la Ventana de Proyectos, haga doble clic en **Symbols**.

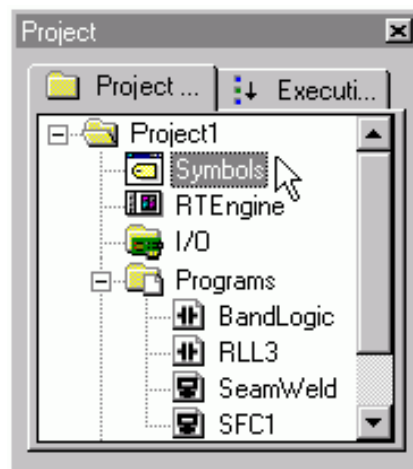


Figura 4. Ejecución del Symbol Manager desde la ventana de proyectos.

3. En la Ventana Watch(Observación), haga clic en **Add Symbol**.



Figura 5. Ejecución del Symbol Manager desde la ventan Watch.

4. En los cuadros de dialogo **Edit Contact** y **Edit Coil**, y las funciones de bloques **RLL**, haga clic en el icono **Symbol Manager**.



Figura 6. Ejecución del Symbol Manager desde el cuadro Edit Contact.

5. Haga doble clic sobre cualquier variable (o sobre cualquier área en blanco) en el programa de Texto Estructurado.

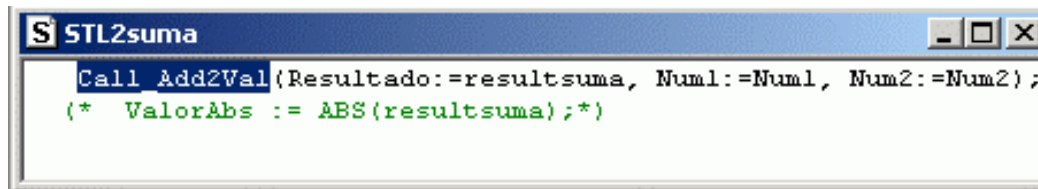


Figura 7. Ejecución del Symbol Manager desde

6. Haga doble clic en el campo de la variable seleccionada en el cuadro de configuración para algunos **FOE's**.

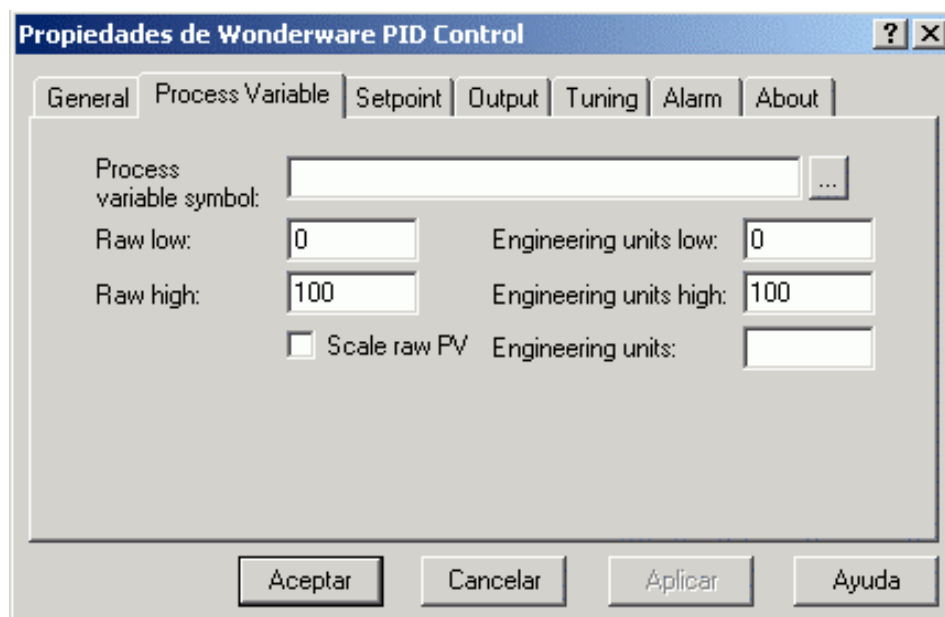


Figura 8. Ejecución del Symbol Manager desde el FOE PID.

El Symbol Manager es mostrado en la siguiente figura:

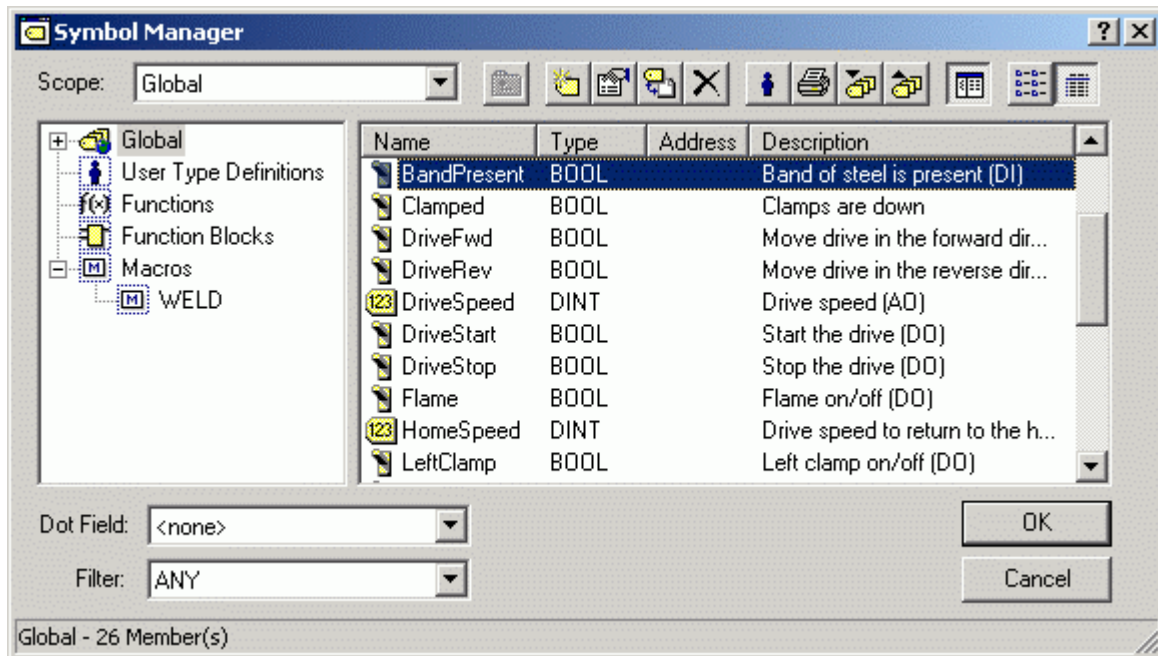



Figura 9. El Symbol Manager.

➤ **Cuadro de dialogo del Symbol Manager.**

Opción/Botones Descripción

Scope

Haga clic en el icono  para mostrar el alcance de la variable: variables globales, variables de programa individuales o locales, funciones, bloques de funciones, macros, variables de sistema de Runtime Engine.

Name

Muestra los nombres de todas las variables en el alcance seleccionado.

Type

Muestra el tipo de dato de la variable.

Address

Para Drivers I/O, especifica la localización del punto I/O.
Para funciones definidas por usuario, funciones de bloques, y tipos de datos, especifica el orden de la estructura.

Description

Muestra la descripción de las variables.

Filter

Usa el filtro para mostrar solamente los tipos de datos seleccionados.











Uso de la barra de tarea del Symbol Manager

La barra de tareas del Administrador de Símbolos muestra las herramientas usadas para crear y manejar los símbolos.



Figura 10. Barra de tareas del Symbol Manager.

Las herramientas del Symbol Manager son mostradas en la siguiente tabla.

Opción/Botones	Descripción
	Haga clic en Up One Level para mover un nivel arriba el alcance en la jerarquía.
	Haga clic en New para adicionar una variable al Symbol Manager.
	Haga clic en Properties para editar una variable existente.
	Haga clic en Cross Referente para mostrar variables, donde y como son usadas en el programa.
	Haga clic en Delete para remover una variable del Symbol Manager.
	Haga clic en Edit User Types para crear/editar un tipo de dato definido por el usuario.
	Haga clic en Print Symbols para imprimir una lista de las variables del Symbol Manager .
	Haga clic en Import para leer un archivo ASCII de variables dentro de la base de datos del InControl.
	Haga clic en Export para crear un archivo ASCII (formato de variables separadas por coma) de las variables InControl.
	Haga clic en Show/Hide Tree para mostrar u ocultar los símbolos de la estructura de árbol.

Opción/Botones	Descripción
	Haga clic en List View para mostrar solamente una variable de la lista.
	Haga clic en Detailed View para mostrar el nombre de una variable, tipo de dato, dirección y descripción.



Consejos de edición - menú de contexto

Durante la sesión de edición de un símbolo, usted puede hacer clic derecho para mostrar rápidamente las opciones de edición del Symbol Manager.

- Con el símbolo seleccionado, haga clic derecho para mostrar el siguiente menú:

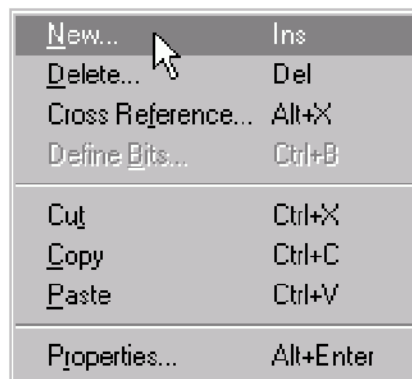


Figura 11. Menú de edición del Symbol Manager.

- Con el símbolo sin seleccionar, haga clic derecho para mostrar algunas de las opciones de la barra de herramientas del Symbol Manager.



Figura 12. Opciones de edición del Symbol Manager.

Consejos de edición– Cambio en el orden de los miembros

Usted puede cambiar el orden de los miembros en un tipo de dato o en funciones de parámetros o en funciones de bloques definidas por el usuario.

- Con el objeto seleccionado, haga clic derecho y use las opciones de **Decrease / Increase Address**.

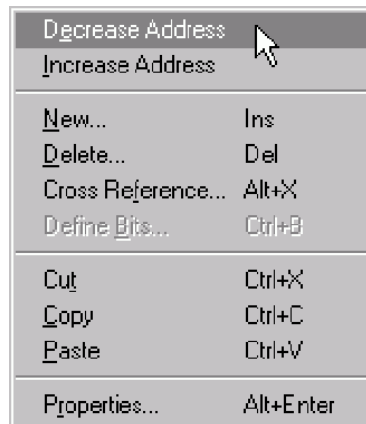


Figura 13. Cambio en el orden de los miembros en una función.

Consejos de edición– copiar, pegar y mover símbolos

Usted puede copiar y pegar símbolos como es descrito a continuación.

- En medio del alcance local y global dentro de un programa, usted puede también arrastrar símbolos entre los alcances.
- Entre el **Symbol Manager** y otras aplicaciones.

- Entre los **Symbol Manager** de dos InControl. Cuando usted selecciona un símbolo, déjelo caer en el alcance apropiado (Global, nombre del programa, o Usuario de Tipo Definición), no en la lista de símbolos. Ver la siguiente figura.

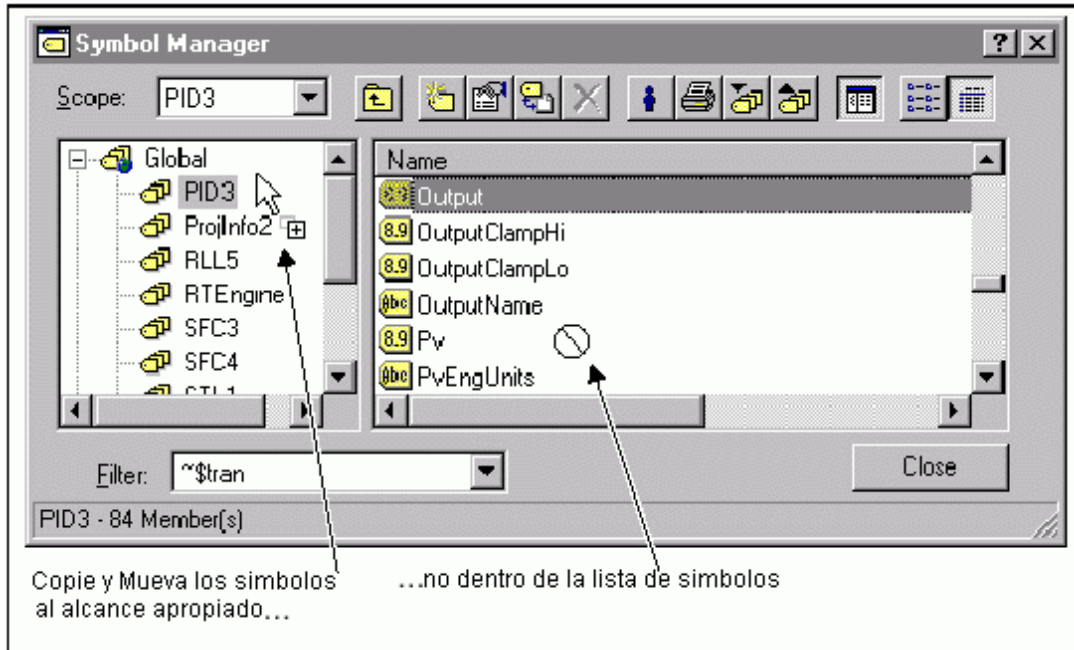


Figura 14. Copiar símbolos dentro del Symbol Manager.

CREACIÓN DE UNA VARIABLE

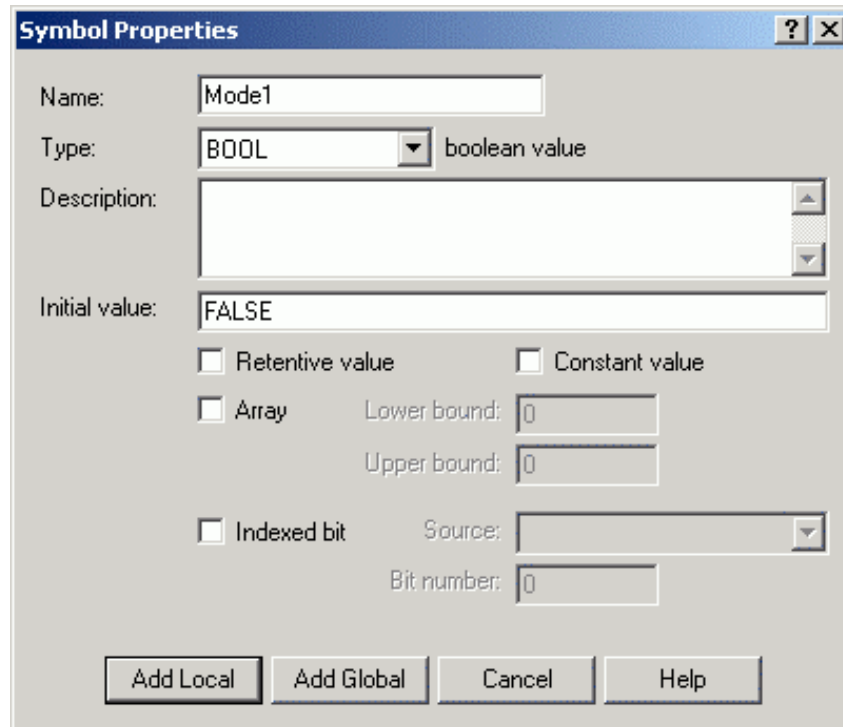
➤ **Para crear una variable de programa:**

1. Entre al **Symbol Manager**.
2. Seleccione el alcance apropiado. Por ejemplo, global, local (un programa individual), funciones de bloques, etc.
3. Haga clic en **New** en la barra de herramientas del Administrador de Símbolos.



Figura 15. Creación de una nueva variable.

Aparecerá el cuadro de dialogo **Symbol Properties** mostrando las mismas propiedades así como para las variables anteriores de la lista.

A screenshot of the 'Symbol Properties' dialog box. The title bar says 'Symbol Properties' with a question mark and a close button. The dialog has several fields and options:

- Name: 'Mode1'
- Type: 'BOOL' (dropdown) with 'boolean value' to the right.
- Description: an empty text area.
- Initial value: 'FALSE'
- Retentive value:
- Constant value:
- Array: Lower bound: '0' Upper bound: '0'
- Indexed bit: Source: (dropdown) Bit number: '0'

At the bottom are four buttons: 'Add Local', 'Add Global', 'Cancel', and 'Help'.

Figura 16. El cuadro Symbol Properties para una variable.

4. Ingrese el nombre de la variable dentro del campo **Name**. Use solamente caracteres alfanuméricos y subrayados. La longitud máxima del nombre de la variable es de 100 caracteres.

Las variables locales deben ser únicas. Esto es, dos variables con igual nombre no pueden estar en el mismo programa. Además, una variable local no puede tener el mismo nombre que una variable global.

5. Seleccione el tipo de dato en el campo **Type**.

6. Ingrese la descripción opcional dentro del campo **Description**.

7. Si la variable requiere un valor inicial, ingrese el valor dentro del campo **Inicial Value**.

8. Revise el cuadro de comprobación **Constant Value** si usted quiere que el valor para esta variable permanezca constante. Este seguro de ingresar el valor en el campo **Initial Value**.

9. Haga clic en el cuadro de comprobación **Retentive Value** para que la variable retenga este valor en caso de una falla de potencia.

El InControl periódicamente guarda variables retenidas y forzadas en el disco duro. La frecuencia de cero deshabilita esta característica. Este intervalo es configurable y usted puede ajustarlo en el cuadro de dialogo “**Runtime Engine Properties**”, descrito en “**Ajuste del tiempo de exploración**” en el capítulo “**ADMINISTRACIÓN DE SISTEMAS INCONTROL**”.

10. Haga clic en **Add Local** o **Add Global**. La nueva variable aparecerá en el campo **Symbol List**.

➤ **Para remover una variable existente:**

1. Haga clic en el nombre de la variable.

2. Haga clic en **Delete** en la barra de tareas del Administrador de Símbolos.



Figura 17. Eliminación de una variable.

Usted también puede utilizar la tecla **Del** en el teclado.

CREACIÓN DE UN ARREGLO DE VARIABLES

➤ Para crear un arreglo de variables:

1. Ingrese al **Symbol Manager**.
2. Seleccione el alcance: global, local (un programa individual), funciones de bloques, etc.
3. Haga clic en **New** en la barra de tareas del **Symbol Manager**.



Figura 18. Creación de un arreglo de variables.

Aparecerá el cuadro de dialogo **Symbol Properties** mostrando las mismas propiedades así como para las variables anteriores de la lista.

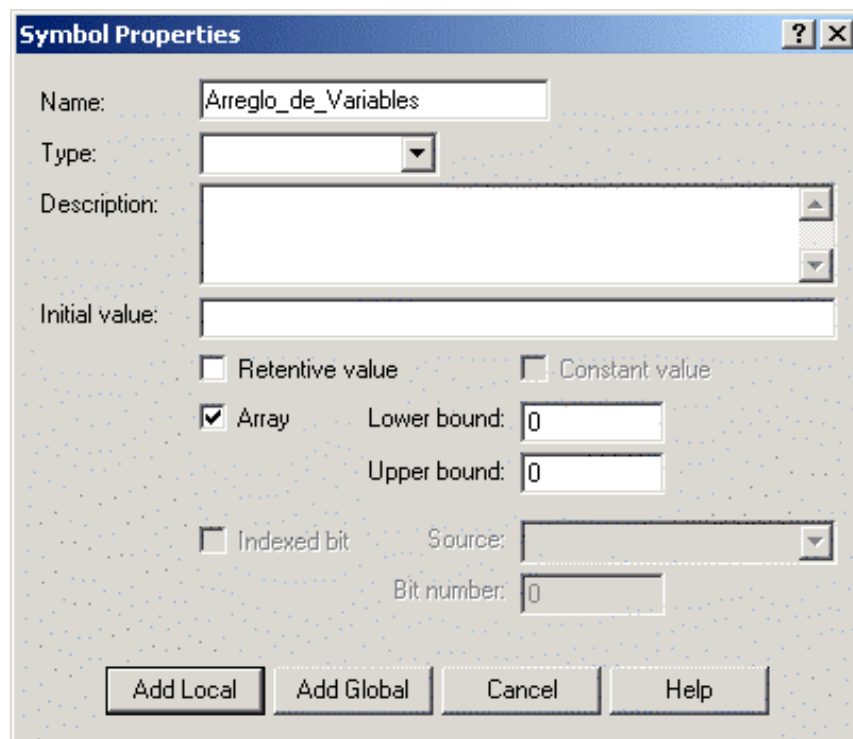


Figura 19. El cuadro Symbol Properties para un arreglo de variables.

4. Seleccione el tipo de dato para los elementos del arreglo en el campo **Type**.

5. Ingrese el nombre del arreglo dentro del campo **Name**. Use solamente caracteres alfanuméricos y subrayados. La longitud máxima del nombre del arreglo es de 100 caracteres.

Las variables locales deben ser únicas. Esto es, dos variables con igual nombre no pueden estar en un mismo programa. En adición, una variable local no puede tener el mismo nombre de una variable global.

6. Ingrese la descripción opcional dentro del campo **Description**.

7. Ingrese un valor en el campo **Initial Value**. Este valor será signado para todos los elementos del arreglo.

8. Revise el cuadro de comprobación **Array**. Este selecciona para un arreglo de variables y activa los campos **Lower Bound** y **Upper Bound**.

9. Ingrese los valores más bajos y más altos para el arreglo, dentro de los campos **Lower Bound** y **Upper Bound**.

10. Haga clic en el cuadro de comprobación **Retentiva Value** si a los elementos del arreglo le serán retenidos sus elementos en caso de una falla de energía.

11. Haga clic en **Add Local** o **Add Global**.

➤ **Para remover un arreglo existente:**

1. Haga clic en el nombre del arreglo.

2. Haga clic en **Delete** en la barra de tareas del **Symbol Manager**.



Figura 20. Eliminación de un arreglo de variables.

Usted también puede utilizar la tecla **Del** en el teclado.

Referenciar arreglos

Use valores literales o expresiones entre corchetes para referenciar los elementos de un arreglo. En el siguiente ejemplo, un elemento es referenciado con un valor literal.

```
Array_of_Values [ 0 ] := 42;
```

En el siguiente ejemplo, un elemento es referenciado con una expresión.

```
Int1 := Array_of_Values [ i + 1 ];
```

Nota: El compilador del InControl genera un error fuera del rango si usted usa un valor literal para el índice del arreglo y si este no está dentro de los límites especificados. Si usted usa una expresión para definir el índice, y resolver las expresiones para un valor que está fuera del rango de ejecución, el programa entrará en el modo de fallo.

ASIGNACIÓN DE UN NOMBRE A UN BIT EN UNA VARIABLE

La característica de bit indexado le permite referenciar un bit específico dentro de un **BYTE**, **WORD**, o una variable **DWORD**. El alcance debe ser el siguiente:

- Si la variable origen es global, el bit puede ser local o global.
- Si la variable origen es local (abarca un programa, o una variable del software de ejecución, por ejemplo) el bit debe tener igual alcance.
- Si la variable origen es parte de un tipo de dato de usuario definido, el bit debe ser parte de este mismo tipo de dato.

➤ **Para asignar nombres a los bits en una variable seleccionada:**

1. Ingrese al **Symbol Manager**.
2. Haga clic derecho en la variable origen (**tipo de datos BYTE, WORD, o DWORD**) y haga clic entonces en **Define Bits**. Aparecerá el cuadro dialogo **Define Bits**.

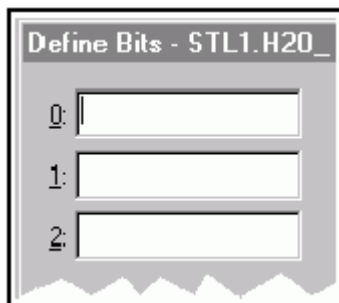


Figura 21. Asignación de nombres a los bits de una variable.

3. Ingrese el nombre para los bits.

➤ **Para definir un bit y especificar su variable origen:**

1. Ingrese al Symbol Manager.
2. Verifique que la variable origen este ya definida. Esta debe ser un tipo de datos **BYTE**, **WORD** o **DWORD** y tener el mismo nivel de alcance.
3. Seleccione el alcance: global, local (un programa individual), funciones de bloques, etc.
4. Haga clic en **New** en la barra de herramientas del Administrador de Símbolos.



Figura 22. Edición de un bit en una variable.

Aparecerá el cuadro de dialogo **Symbol Properties**.

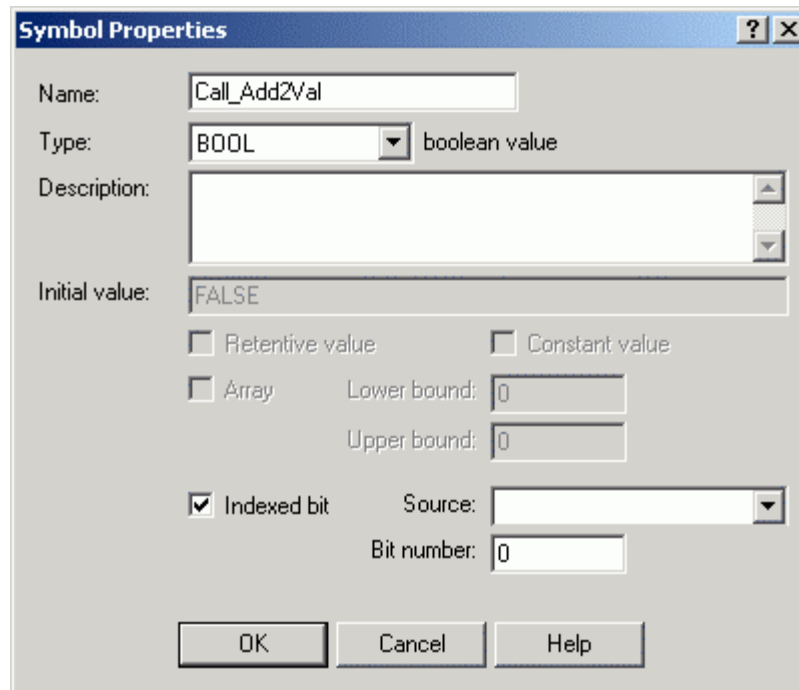


Figura 23. Definición de bits a una variable.

5. Ingrese el nombre del bit dentro del campo **Name**. Use solamente caracteres alfanuméricos y subrayados. La longitud máxima del nombre de un bit es de 100 caracteres. Las variables locales deben ser únicas. Esto es, dos variables con igual nombre no deben estar en un mismo programa,

además, una variable local no puede tener el mismo nombre de una variable global.

6. Seleccione el tipo de dato **Booleano** en el campo **Type**, e ingrese la descripción opcional dentro del campo **Description**.
7. Revise el cuadro de comprobación **Indexed Bit**.
8. Ingrese el nombre de la variable en el campo **Source**.
9. Ingrese el numero del bit en el campo **Bit #**. Valores validos son 0-7 para el **BYTE**, 0-15 para **WORDS**, y 0-31 para **DWORDS**.
Si el origen es un arreglo, ingrese el número del bit basado en su posición en el arreglo. Por ejemplo, para indexar el último bit en un arreglo de cuatro bytes, ingrese 31 para el número del bit.
10. Haga clic en **Add Local** o **Add Global**.

➤ **Para remover un bit indexado existente:**

1. Haga clic en el nombre del bit.
2. Haga clic en **Delete** en la barra de herramientas del Administrador de Símbolos.



Figura 24. Eliminación de un bit en una variable.

Usted también puede presionar la tecla **Del**.

CREACIÓN DE UN TIPO DE DATO USER-DEFINED (DEFINIDO POR EL USUARIO)

Usted puede diseñar un estilo de tipo de datos para aplicaciones específicas. Primero, usted creara la definición del tipo de usuario y entonces agregara variables a su programa que esta basado en esta definición. La definición del tipo de usuario es una estructura que consiste de un conjunto de tipos de datos existentes (**INT**, **WORD**, **REAL**, otros tipos de datos definidos por el usuario, etc.), los cuales son llamados miembros. Los tipos de datos **TMR** y **FILE**, funciones y funciones de bloques no son miembros validos para los tipos de datos **USER-DEFINED**(definido por el usuario). Para cada definición del tipo de dato **USER-DEFINED**, usted especificara el número de miembros y el tipo de dato para cada miembro. Los miembros de la definición del tipo de usuario no tienen que ser iguales a los tipos de datos.

Diseño de un de tipo de dato propio

➤ **Para crear una definición de un tipo de usuario:**

1. Entre al Administrador de Símbolos.
2. Haga clic en **Edit User Types** en la barra de herramientas del Administrador de Símbolos.



Figura 25. Creación de un tipo de dato definido por el usuario.

3. Haga clic en **New** en la barra de herramientas del Administrador de Símbolos.



Figura 26. Creación de un tipo de dato en el Symbol Manager.

Aparecerá el cuadro de dialogo **User Type Definition**.

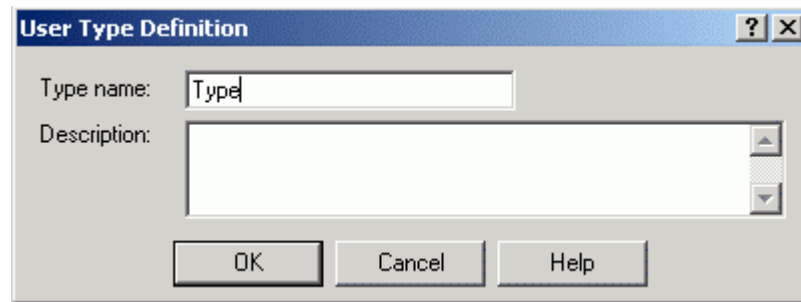


Figura 27. Definición del tipo de dato definido por el usuario.

4. Ingrese un nombre en el campo **Type Name**, y una descripción (opcional). Entonces haga clic en **OK**.
5. Para comenzar a adicionar miembros a la estructura, haga clic en **New** en la barra de tareas del Administrador de Símbolos. Aparecerá el cuadro de dialogo **Symbol Properties**.
6. Ingrese la información apropiada, como esta descrito en “**CREACIÓN DE UNA VARIABLE**”(pagina 6-28).
7. Haga clic **Add Member**. Entonces repita los pasos 5-7 para agregar miembros adicionales a la estructura.

Usted puede cambiar el orden de los miembros en un tipo de datos **USER-DEFINED**. Con el tipo de dato seleccionado, haga clic derecho y use las opciones **Decrease / Increase Address**. Esto le permite modificar un tipo de dato **USER-DEFINED** existente en vez de definir uno nuevo.



Figura 28. Cambio en el orden del tipo de dato definido por el usuario.

Nota: Si usted hace cambios en la definición del tipo de usuario, como adicionar o remover un miembro, en todos los casos la definición es actualizada automáticamente. A menudo si usted especifica un valor inicial para un miembro y después cambia este valor, los valores iniciales de cualquier variable que este basado en el tipo de dato definido originalmente no se actualizan. Tales cambios, incluyen un cambio en la dirección del orden de los miembros, requiere una recarga total del proyecto antes que usted pueda correrlo.

Uso del tipo de dato user-defined

La nueva definición del tipo de usuario aparecerá una vez que seleccione en el campo **Type** del cuadro de dialogo **Symbol Properties**, junto con los otros tipos de datos. Cuando usted define una nueva variable en el Administrador de Símbolos, usted puede escoger la nueva definición del tipo de usuario como un tipo de dato para la variable.

Los siguientes ejemplos demuestran varias maneras de referenciar tipos de datos **USER-DEFINED**.

Referenciando un miembro individual:

```
User_Type1 . Int1 := 42;
```

Referenciando un arreglo de tipos **USER-DEFINED**:

```
User_Type2 [ 6 ] . Int1 := 11;
```

Referenciando un arreglo de tipos de datos **USER-DEFINED** con un miembro que es un arreglo:

```
User_Type3 [ 9 ] . Int2 [ 3 ] := 77;
```

Nota: El compilador del InControl genera un error fuera del rango si usted usa un valor literal para indexar un arreglo y si este no esta dentro de los límites especificados. Si usted usa una expresión para definir el índice, y resolver las expresiones para un valor que esta fuera del rango del tiempo de ejecución, el programa entrara en el modo de fallo.

PRESENTACIÓN DE INFORMACIÓN DE LAS VARIABLES

Usted puede imprimir los siguientes tipos de informes desde el Administrador de Símbolos.

- Detalles de los símbolos—Tipos de datos, descripción, y un valor inicial para cada variable.
- Referencias de los programas—Todas las variables que son referenciadas por cada programa.
- Referencias cruzadas—Todos los programas que referencian cada variable.
- Sin referenciar—Variables que han sido definidas pero que no están referenciadas.

El compilador del InControl genera la información que se ha de imprimir. Por lo tanto usted debe validar el proyecto antes de imprimir un informe.

➤ **Para imprimir información de las variables:**

1. Valide el proyecto.
2. Entre al Administrador de Símbolos.
3. Haga clic en **Print Symbol** en la barra de herramientas del Administrador de Símbolos.



Figura 29. Impresión de información de las variables.

Aparecerá el cuadro de dialogo **Print Select Report type**.

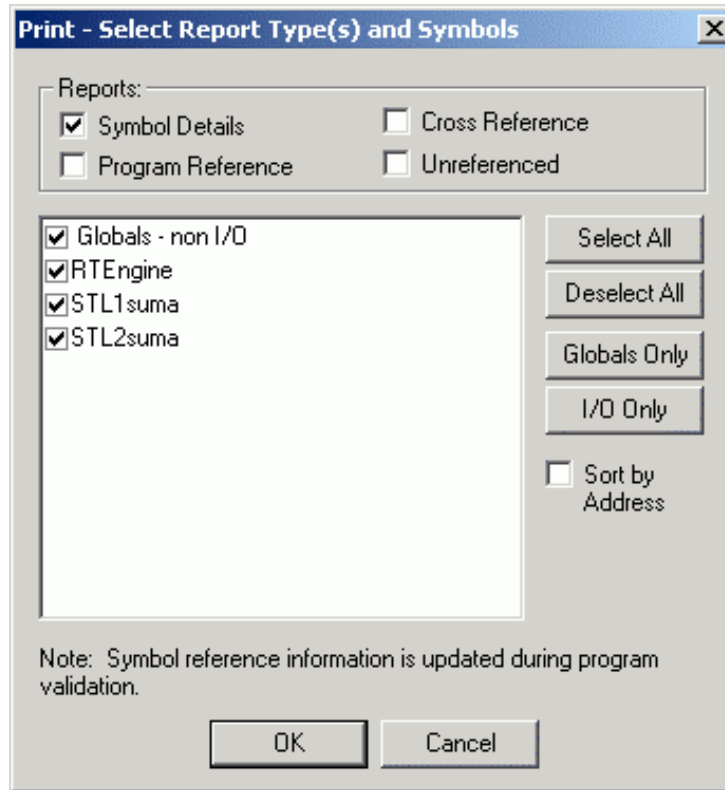


Figura 30. Selección del tipo de informe a imprimir.

4. Seleccione la clase de reporte y las variables (global, local I/O, etc.).

Nota: Si usted imprime mas de un tipo de informe, usted le indicara para seleccionar la impresora para cada informe.

VARIABLES DE SISTEMA

El InControl provee variables de sistemas que pueden ser usadas para monitorear y controlar varias funciones del sistema.

Variables de Sistema

Variable	Descripción
Variabes de Contadores	Estas variables contiene información de estados para los contadores RLL (CTD, CTU, y CTUD) , para mas información ver el capitulo “ ELEMENTOS DE PROGRAMA RLL ”.
Variabes de Control de Archivos	Estas variables contienen información de estados para operaciones de archivos. Para mas información ver el capitulo “ ELEMENTOS DE PROGRAMAS RLL ”.
Mode ¹	Una variable Mode (INT) es creada por cada programa y también por cada configuración I/O. Estas variables pueden tener los siguientes valores: 0 = Descargado desde el runtime engine 1 = Detenido. El programa (I/O) esta detenido. 2 = Pausado. El programa (I/O) esta pausado. 3 = Exploración Simple. El programa (I/O) esta en el modo exploración simple. 4 = Programa. El programa (I/O) esta siendo cargado en el Runtime Engine . 5 = Ejecutar. El programa (I/O) se esta ejecutando. 6 = Falla. El programa (I/O) tiene un error. Para borrar la falla, ver “ Borrar Modos de Fallo y Condiciones de Error ” del Capitulo “ ADMINISTRACIÓN DEL SISTEMA INCONTROL ”. 7 = El programa SFC ha finalizado su ejecución. 8 = Para programas: el programa es cargado en el y esta listo para ejecutarse cuando sea llamado por otro programa. Para configuraciones I/O: la configuración ha sido cargada por un nodo remoto.
NOW	Contiene el tiempo actual del sistema.
Variabes de Temporizadores	Estas variables contienen información de estados para los temporizadores RLL (TOF, TON, y TP) .
TODAY	Contiene la fecha actual del sistema.
T ²	Contiene el tiempo transcurrido durante la ejecución de un paso SFC .
X ²	Contiene los estados activo / inactivo de un paso SFC .
DN ²	Para un SFC , indica cuando este ha finalizado su ejecución. Para un Paso , indica cuando el código dentro del paso ha finalizado su ejecución.

¹ En el programa las variables **Mode** son mostradas en el Administrador de Símbolos, pero en la tarjeta I/O las variables **Mode** no se encuentran.

² Para mas información ver el capitulo “**ELEMENTOS DE PROGRAMA SFC**”.

Nota: Las variables de sistemas generadas por el Runtime Engine no aparecen en el nuevo sistema instalado o cuando usted detenga el Runtime Engine (haga clic en **Stop** en el menú **Runtime**). En estas situaciones, usted no puede adicionarlas a la Ventana de Observación y los clientes **SuiteLink/DDE** no pueden leerlas.

VARIABLES DE SISTEMAS DEL RUNTIME ENGINE

Variable	Descripción
RTEngine.DivideZero	Valor Booleano . TRUE indica una división por cero; ajustado por el Runtime Engine (RTE).
RTEngine.Error	Valor Booleano . TRUE indica una condición de error que ha ocurrido en el Runtime Engine. Consulte el registro Wonderware para más información.
RTEngine.ExecAvg	Valor TIME . Contiene el tiempo promedio de ejecución del programa que esta siendo ejecutado.
RTEngine.ExecLast	Valor TIME (solo lectura). Contiene el último tiempo de ejecución del programa que esta siendo ejecutado.
RTEngine.ExecMax	Valor TIME . Contiene el tiempo máximo de ejecución del programa que esta siendo ejecutado.
RTEngine.FirstScan	Valor Booleano (solo lectura). TRUE indica el acontecimiento de la primera exploración del programa.
RTEngine.FirstScanOnAutoStart	Valor Booleano (solo lectura). TRUE indica el acontecimiento de la primera exploración del programa después del inicio automático seguido del reinicio del sistema.
RTEngine.IOAvg	Valor TIME . Contiene el tiempo promedio de exploración I/O.
RTEngine.IOLast	Valor TIME (solo lectura). Contiene el ultimo tiempo de exploración I/O.
RTEngine.IOMax	Valor TIME . Contiene el tiempo máximo de exploración I/O.

Variable	Descripción
RTEngine.Mode	Valor INT . Indica el modo actual del Runtime Engine: 0 = Todos los programas son descargados desde el Runtime Engine. 1 = Detenido. Los programas en un proyecto están detenidos. 2 = Pausado. Los programas en un proyecto están pausados. 3 = Exploración Simple. Los proyectos están en el modo exploración sencilla. 4 = Programa. El proyecto esta siendo cargado en el Runtime Engine. 5 = Ejecutar. Por lo menos un programa en un proyecto sé esta ejecutando. 6 = Falla. El Runtime Engine no puede correr un proyecto.
RTEngine.PowerFail	Valor Booleano que indica una falla de energía cuando es TRUE . Una configuración UPS es requerida. Nota: usted debe resetear esta variable manualmente, desde la ventana de observación.
RTEngine.RelativeTime	Valor TIME (solo lectura) que contiene la duración del tiempo ha estado corriendo desde que el sistema fue iniciado. Este valor es independiente del reloj del sistema y puede ser usado en un programa para calcular los intervalos medidos.
RTEngine.ScanAvg	Valor TIME que contiene el promedio de exploración.
RTEngine.ScanLast	Valor TIME (solo lectura) que contiene la duración de la ultima exploración.
RTEngine.ScanMax	Valor TIME que contiene el tiempo máximo de exploración.
RTEngine.ScanOverrun	Valor Booleano que indica una exploración excesiva cuando es TRUE .
RTEngine.ScanTime	Valor LREAL que contiene el tiempo actual de exploración asignado por el usuario ajustado en milisegundos. Valores escritos en esta variable cambiaran el tiempo de exploración del software de Ejecución.

TRANSFERENCIA DE BASES DE DATOS DE SÍMBOLOS

El InControl soporta el intercambio de símbolos entre proyectos de InControl y entre el InControl e InTouch. La utilidad de Exportar / Importar puede escribir y leer un archivo de texto **ASCII** que tiene el formato de separación por comas (**CSV**). Usted puede fácilmente editar un archivo de símbolos usando un editor de textos **ASCII** o una hoja de cálculo, como **Excel**.

La utilidad de exportar le da las siguientes opciones para exportar información de símbolos.

- Formato InControl.
- Referencia Cruzada InControl.
- Formato InTouch.
- Súper Etiquetas InTouch.

Intercambio de Símbolos Entre Proyectos InControl

Cuando usted transfiere símbolos entre proyectos InControl, considere los siguientes puntos.

- Use el archivo de formato InControl cuando exporte símbolos. Cuando usted importe símbolos, este formato es usado automáticamente.
- Los símbolos de InControl son creados por un número de diferentes clases de objetos. Algunos de estos objetos no permiten importación de símbolos, por ejemplo, **ActiveX**, **FOE's** y **objetos I/O**.
- Usted puede cargar nuevos símbolos locales dentro de un programa, pero el programa ya debe existir. Por ejemplo, para cargar un símbolo local dentro de un programa llamado **RLL1**, usted debe crear un programa llamado **RLL1** Antes de importar los símbolos. Los símbolos globales pueden ser importados sin esta limitación.

Reportes De Referencias Cruzadas De Símbolos

Usted puede crear un informe de símbolos de referencias cruzadas que tiene el formato de separación por comas. Usted puede abrir el informe con cualquier editor de texto **ASCII**, como el bloc de notas, o con una hoja de cálculo, como Excel.

Intercambio de Símbolos Entre InControl e InTouch

Cuando usted transfiera símbolos entre el InControl e InTouch, considere los siguientes puntos.

- Use el formato de archivos InTouch o el formato de archivos de Súper Etiquetas de InTouch cuando usted exporte símbolos hacia InTouch. Escoja el formato de Súper Etiquetas para crear súper etiquetas InTouch de grupos lógicos de símbolos InControl. Los siguientes grupos son ejemplos de símbolos InControl que pueden ser exportados como súper etiquetas:
 - Elementos de un Arreglo.
 - Parámetros de Temporizadores o Contadores.
 - Parámetros de cualquier función o funciones de bloques.
 - Variables de Sistema del Software de Ejecucion.
- Símbolos exportados desde InTouch solamente pueden ser importados como símbolos globales al InControl.
- Cuando importe símbolos hacia el InTouch, los nombres de los símbolos serán traducidos en un formato compatible con InTouch en igual manera el asistente lo hará en InControl. Todos los puntos y corchetes serán convertidos en subrayados. Esto es, **GArray[1]** será **GArray_1** y **RTEngine.ScanMax** será **RTEngine_ScanMax**.

Importación y Exportación de Símbolos

➤ **Para exportar símbolos:**

1. Ingrese al Administrador de Símbolos y haga clic sobre **Export** en la barra de herramientas del Administrador de Símbolos.



Figura 31. Herramienta de exportación de símbolos.
Aparecerá el cuadro de dialogo **Export-Select Report Types**.

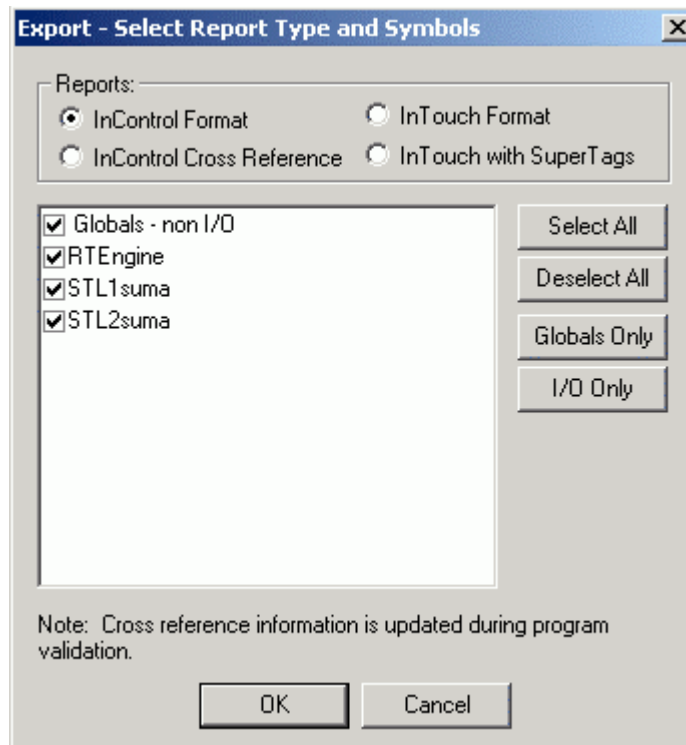


Figura 32. Selección del tipo de informe y los símbolos a exportar.

2. Seleccione el tipo de reporte y las variables (globales, locales, I/O, etc.).
Para exportar a otro proyecto InControl, haga clic sobre **InControl Format**.
Para exportar hacia un proyecto InTouch, haga clic sobre **InTouch Format** o **InTouch SuperTags**.

Para generar un archivo de símbolos de referencias cruzadas, haga clic en **InControl Cross Reference**.

3. Haga clic en **OK**. Aparecerá el cuadro de dialogo **Save As**.
4. Ingrese el nombre del archivo y haga clic en **Save**. El nuevo archivo es creado en el formato **CSV**.

➤ **Para importar símbolos:**

1. Ingrese al Administrador de Símbolos y haga clic sobre **Import** en la barra de herramientas del Administrador de Símbolos.



Figura 33. Herramienta de importación de símbolos.

Aparecerá el cuadro de dialogo **Abrir**.

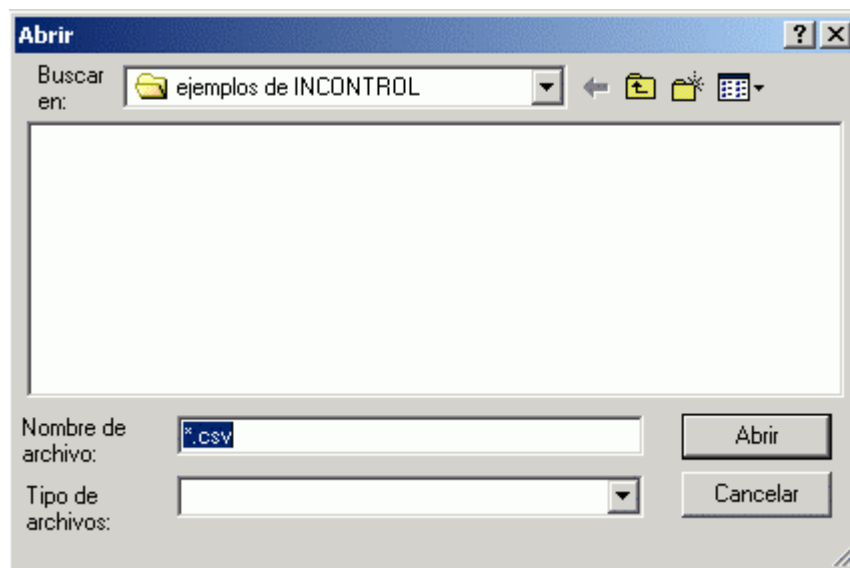


Figura 34. Selección de símbolos a importar.

2. Seleccione el archivo de símbolos y haga clic en **Open**. Los símbolos serán importados y mostrados en el Administrador de Símbolos.

Formato CSV de InControl para Archivos

El formato CSV de InControl para archivos es mostrado en la figura **Formato CSV InControl**(pagina 6-50).

Las etiquetas usadas en el formato **CSV** son las mostradas abajo:

- **ParentObject** – nombre del objeto que creo el símbolo. Si es global, este es Global.
- **ParentObjectType** – Global, Programa, ActiveX, o **IO**.
- **Name** – nombre del símbolo.
- **DataType** – **BOOL, INT, DINT, BYTE, WORD, DWORD, REAL, LREAL, STRING, TIME, DATE, TOD**, o un tipo **USER-DEFINED**.
- **InitialValue** – valor inicial del símbolo.
- **Attributes** – atributos del símbolo, separados por el carácter |. Esto es necesario para mantener vínculos internos dentro del InControl, y para no ser editado normalmente. Si usted esta creando nuevos símbolos, es recomendable que usted haga una copia de los símbolos existentes con las características requeridas y entonces haga los cambios necesarios.
- **ArrayDefinition** – si el símbolo es un arreglo, este campo contiene los valores más bajos y más altos del arreglo. Si el campo esta vacío, la etiqueta no es un arreglo.
- **Comment** – este campo contiene una descripción definida por el usuario del símbolo.
- **IOTitle** – si el símbolo es un punto **IO**, este campo contiene la cadena de caracteres de salida para el driver que describe el símbolo que representa, por ejemplo, **PROFIBUS-DP borrad.Module.Port**.
- **IOName** -- si el símbolo es un punto **IO**, este campo contiene la cadena de caracteres de salida para el driver que contiene el nombre correspondiente al IOTitle, por ejemplo, **SMSProfi.OutPutMod.OutPortMs**. Si el símbolo es un bit indexado dentro de otro símbolo, este campo contiene el nombre del símbolo fuente y el bit de desplazamiento (offset), por ejemplo, **GWord.0, GWord.1**, etc.

- **BitIndexSource** -- Si el símbolo es un bit indexado dentro de otro símbolo, este campo contiene el nombre del símbolo fuente.
- **BitIndexPosition** -- Si el símbolo es un bit indexado dentro de otro símbolo, este campo contiene la posición del bit dentro de la fuente.
- **Address** – la dirección del símbolo. La mayoría de los símbolos no tiene una dirección.
- **Order** – contiene el orden (hexadecimal) de un miembro dentro de la estructura de un tipo de dato definido por el usuario.

El formato del archivo **CSV InControl** es mostrado en la siguiente figura.

FORMATO CSV INCONTROL

```
ParentObject,ParentObjectType,Name,DataType,InitialValue,Attributes,ArrayDefinition,Comment,
Global,Global,MyBool,BOOL,FALSE,PUBLIC|USER_CREATED,,a global Boolean value,,,,,
Global,Global,MyBoolArray,BOOL,TRUE,PUBLIC|USER_CREATED|IEC_TYPE,1-5,array 1-5,,,,,
Global,Global,MyComment,LREAL,0,USER_CREATED|IEC_TYPE,,comment has comma in it,,,,,
Global,Global,MyInt,INT,34,PUBLIC|USER_CREATED|IEC_TYPE,,integer of init value 34,,,,,
RTEngine,Program,DivideZero,BOOL,FALSE,PUBLIC|IEC_TYPE|READ_ONLY,,,,,
RTEngine,Program,Error,BOOL,FALSE,PUBLIC|IEC_TYPE|READ_ONLY,,,,,
RTEngine,Program,ExecAvg,TIME,T#0ms,PUBLIC|IEC_TYPE|READ_ONLY,,,,,
RTEngine,Program,ExecLast,TIME,T#0ms,PUBLIC|IEC_TYPE|READ_ONLY,,,,,
RTEngine,Program,ExecMax,TIME,T#0ms,PUBLIC|IEC_TYPE|READ_ONLY,,,,,
RTEngine,Program,FirstScan,BOOL,FALSE,PUBLIC|IEC_TYPE|READ_ONLY,,,,,
RTEngine,Program,IOAvg,TIME,T#0ms,PUBLIC|IEC_TYPE|READ_ONLY,,,,,
RTEngine,Program,IOLast,TIME,T#0ms,PUBLIC|IEC_TYPE|READ_ONLY,,,,,
RTEngine,Program,IOMax,TIME,T#0ms,PUBLIC|IEC_TYPE|READ_ONLY,,,,,
RTEngine,Program,Mode,INT,0,PUBLIC|IEC_TYPE|READ_ONLY,,,,,
RTEngine,Program,PowerFail,BOOL,FALSE,PUBLIC|IEC_TYPE|READ_ONLY,,,,,
RTEngine,Program,RelativeTime,TIME,T#0ms,PUBLIC|IEC_TYPE|READ_ONLY,,,,,
RTEngine,Program,ScanAvg,TIME,T#0ms,PUBLIC|IEC_TYPE|READ_ONLY,,,,,
RTEngine,Program,ScanLast,TIME,T#0ms,PUBLIC|IEC_TYPE|READ_ONLY,,,,,
RTEngine,Program,ScanMax,TIME,T#0ms,PUBLIC|IEC_TYPE|READ_ONLY,,,,,
RTEngine,Program,ScanOverRun,BOOL,FALSE,PUBLIC|IEC_TYPE|READ_ONLY,,,,,
RTEngine,Program,ScanTime,LREAL,0,PUBLIC|IEC_TYPE|READ_ONLY,,,,,
```

```
IOTitle,IOName,BitIndexSource,BitIndexPosition,Address,Order
```

Edición de archivos de símbolos

Usted puede editar las definiciones de los símbolos con una hoja de cálculo, como Excel, o un editor de texto ASCII y entonces reimportarlos dentro del InControl. El InControl automáticamente combina cualquier cambio con las definiciones de símbolos existentes. Es recomendable que usted no haga cualquier cambio a los atributos.

En general, si una cadena de caracteres contiene una coma, la cadena de caracteres debe ser encerrada en doble comillas (“este comentario tendrá una coma, dentro de este”).

El orden de los registros dentro de un archivo **CSV** tiene estas restricciones para los símbolos definidos por el usuario y para los símbolos de bit indexado:

Símbolos User-Defined o definidos por el usuario

- El tipo definido por el usuario debe ser primero.
- Los miembros para el tipo definido por el usuario aparecen en segundo lugar.
- Los símbolos individuales del tipo definido por el usuario aparecen de último.

Símbolos de Bit Indexado

- El símbolo fuente debe ser primero.
- Los bits indexados deben seguir al símbolo fuente.

USO DEL EDITOR RLL

Este capítulo nos muestra el funcionamiento del editor RLL (Lógica en Escalera por relevos) y describe como poder usarlo para crear nuevos programas y para adicionar elementos (contactos, bobinas, bobinas de salto, bloques de funciones, etc.) a estos. Para información mas detallada acerca de estos elementos, referirse al capítulo “ELEMENTOS DE PROGRAMAS RLL”.

CONTENIDO

- CREACIÓN DE UN PROGRAMA RLL
- HERRAMIENTAS RLL
- ADICIÓN DE ELEMENTOS DE PROGRAMA
- EDICIÓN DE ELEMENTOS DE PROGRAMA
- ADICIÓN DE UN COMENTARIO

This product is licensed to:

Company:	N/A
SN:	N/A
Expires:	N/A
Locked To:	N/A

7.1.0.1337

CREACIÓN DE UN PROGRAMA RLL

Después de iniciar el InControl, usted puede crear un nuevo programa **RLL** o editar uno existente.

➤ **Para crear un nuevo programa RLL:**

1. Sobre el menú **File**, haga clic en **New**. Aparecerá el menú de tipos de programas soportados por InControl.
2. Seleccione **RLL Program** y haga clic en **Aceptar**. Aparecerá el cuadro de dialogo **Save As**.
3. Escoja un nombre (Hasta 31 caracteres) y el directorio (Proyecto) para el programa y haga clic en **Save**. Aparecerá un nuevo programa RLL, mostrando dos barras de energía y un escalón. El nuevo programa aparecerá en la ventana de proyectos.

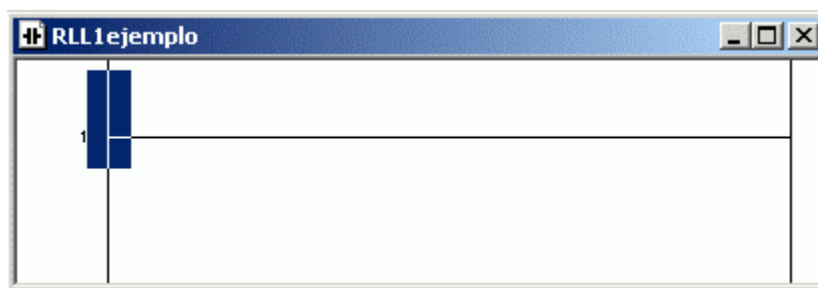


Figura 1. Presentación y definición del programa

4. Para adicionar elementos de programa, ver “**ADICIÓN DE ELEMENTOS DE PROGRAMA**” (Pagina 7-8).

- **Para editar un programa RLL existente:**
1. Si la ventana de proyectos no esta abierta, haga clic en **Project** en el menú **View**. Aparecerá la ventana de proyectos.
 2. Haga doble clic en el nombre del programa que usted quiere editar.

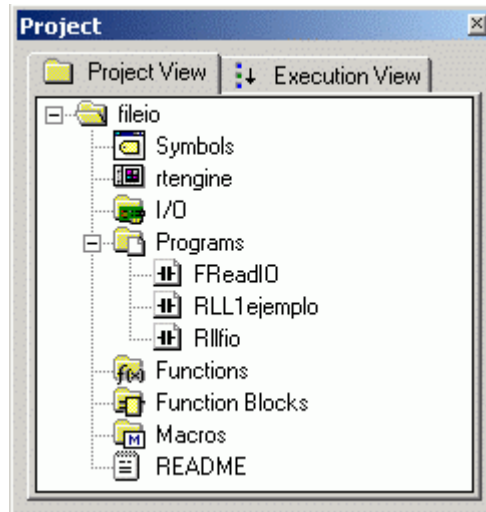


Figura 2. Presentación del programa en la Project View

Se abre el editor **RLL**, mostrando el programa seleccionado.

Usted también puede hacer clic en **Open** en el menú **File** para abrir un programa existente para editarlo. Cuando aparezca el cuadro de dialogo **Open**, seleccione el programa ha abrir. Si un programa no es parte del actual proyecto, usted puede adicionarlo.

Usted puede hacer clic en **Files Into Project** en el menú **Insert** para adicionar cualquier **POU** (Programa, Función, Bloque de Función, etc.) al proyecto. En la siguiente figura, el programa **STL1ejemplo**, que se muestra en el cuadro de dialogo **Insert Files Into Project**, es seleccionado y puede ser adicionado a **PIDApp**. Note que el archivo propiamente dicho no es copiado o movido cuando es adicionado a otro proyecto.

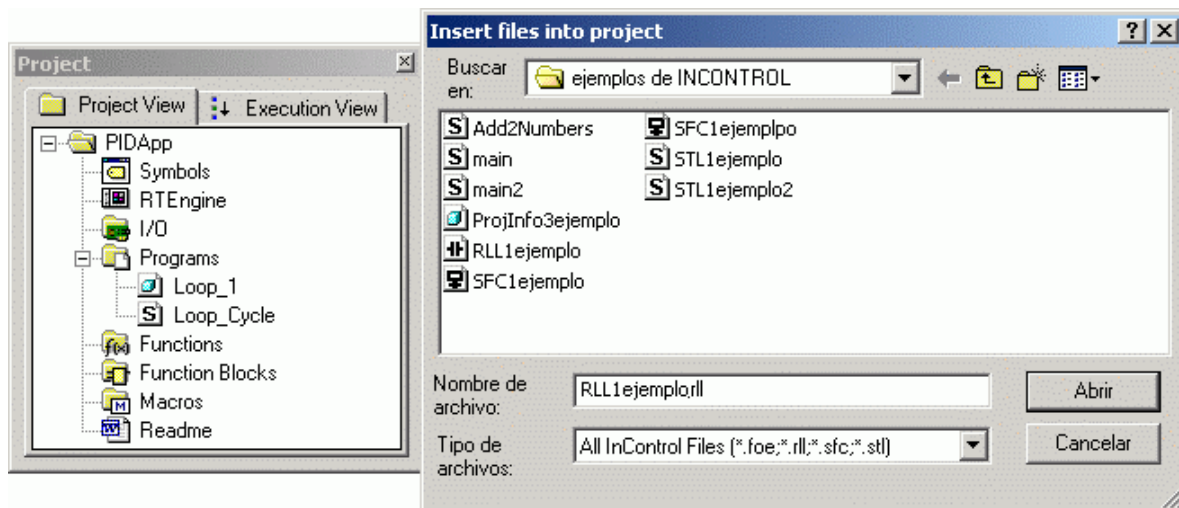


Figura 3. Adición de POUs a un proyecto.

Nota: Todas las **POUs** son insertadas debajo de la carpeta **Programs** de la ventana de proyectos. Usted debe mover las funciones a la carpeta **Functions**, los tipos de bloque se funciones a la carpeta **Function Block**, y los macros a la carpeta **Macros** para la correcta compilación del proyecto.

Si usted abre un proyecto desarrollado bajo **InControl 7.0**, usted tiene la opción de convertir los archivos a un proyecto en **InControl 7.1**. Cualquier macro en un proyecto aparecerá en la carpeta **Programs** después de la conversión. Usted puede mover estos macros a la carpeta **Macros**, pero esto no es necesario para compilar el proyecto.

HERRAMIENTAS RLL

Esta sección describe la barra de herramientas **RLL** y muestra algunos consejos útiles para la edición de programas.

Uso de las herramientas RLL y la barra de Menú

La barra **RLL** muestra las herramientas usadas para crear programas **RLL**.



Figura 4. La barra de herramientas RLL

Las opciones de la barra de herramientas RLL son descritas en la siguiente tabla.

Tabla 1. Iconos de la barra de herramientas RLL

Icono	Opción en la barra de menú	Función
	n/a	Le permite seleccionar elementos de programa.
	Contacto	Adiciona un contacto al programa.
	Etiqueta	Adiciona una etiqueta al programa.
	Bobina	Adiciona una bobina al programa.
	Bobina de Salto	Adiciona una bobina de salto al programa.
	Bobina de Transición SFC	Adiciona una bobina de transición al programa. Esta opción solo esta disponible cuando el programa esta siendo editado.
	Ramal o Derivación	Adiciona un ramal OR al programa.
	Escalón	Adiciona un nuevo escalón al programa.

Consejos de edición

Estos consejos pueden ayudarlo a editar sus programas.

- Este capítulo describe como usar estas herramientas, basándose en selecciones que usted hará desde la barra de herramientas RLL. Usted también puede escoger estas herramientas seleccionándolas desde el menú **Insert**, el cual se muestra en la siguiente figura. Para evitar confusión, solo un método es descrito en este capítulo.



Figura 5. Uso de las herramientas RLL desde el menú Insert

Cuando usted inserta un elemento de programa desde la barra de **Menú**, el elemento es insertado en la localización actual del cursor dentro del programa. Cuando usted inserta un elemento de programa usando la barra de herramientas, usted puede mover el cursor a localización en el programa donde usted quiera colocar el elemento.

- Use el menú **View** para mostrar los objetos que usted necesita ver durante una sesión de edición. Por ejemplo, si usted prefiere adicionar elementos de programa desde la barra de **Menú**, en lugar de la barra de herramientas **RLL**, usted puede ocultar la barra de herramientas **RLL**.

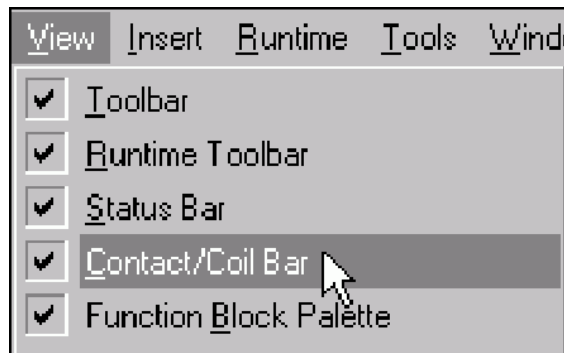


Figura 6. Desactivando la barra RLL

- Durante una sesión de edición, usted puede hacer clic derecho para mostrar rápidamente algunas de las opciones de edición que aparecen en la barra de menú.

Con el cursor encima de un elemento RLL, haga clic derecho para mostrar el siguiente menú:



Figura 7. Opción de edición de un elemento RLL

Con el cursor en la ventana de edición, pero no encima de un elemento RLL, haga clic derecho para mostrar el siguiente menú:

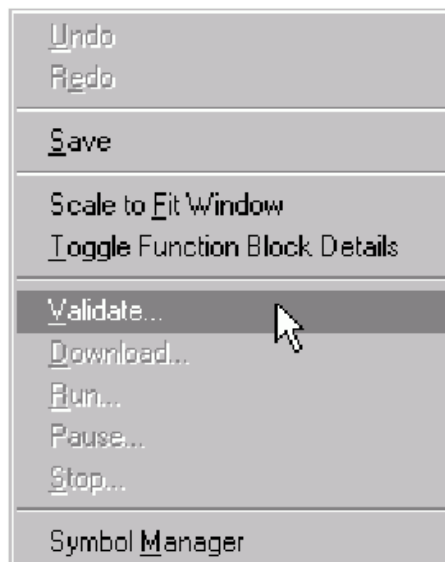


Figura 8. Opción de edición de un elemento RLL

Estas opciones de menú son descritas en el capítulo “**AMBIENTE INCONTROL**”.

- Para imprimir un Programa RLL, incluyendo datos de configuración para los bloques de funciones, seleccione **Function Block Details** sobre el menú **View**. Luego imprima el programa.

ADICIÓN DE ELEMENTOS DE PROGRAMA

Esta sección describe como adicionar elementos de programa usando las herramientas de la barra RLL.

Adición de un contacto

➤ Para adicionar un contacto a un programa:

1. Pulse en **Contact Tool** sobre la barra de herramientas RLL.

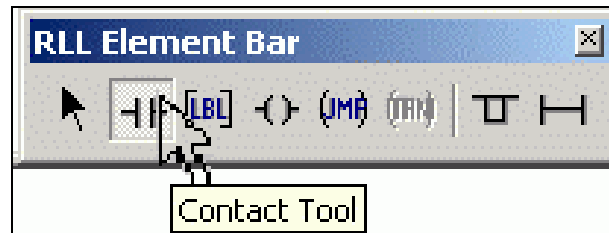


Figura 9. Adición de un contacto al programa.

2. Mueva el cursor a la localización sobre el escalón donde usted quiere colocar el nuevo contacto.

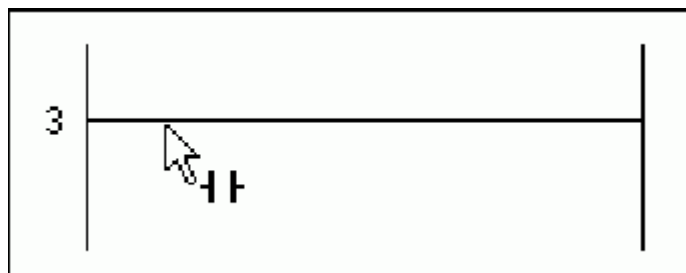


Figura 10. Ubicación del contacto en el escalón.

3. Pulse el botón izquierdo del mouse. Aparecerá el cuadro de dialogo **Edit Contact**.

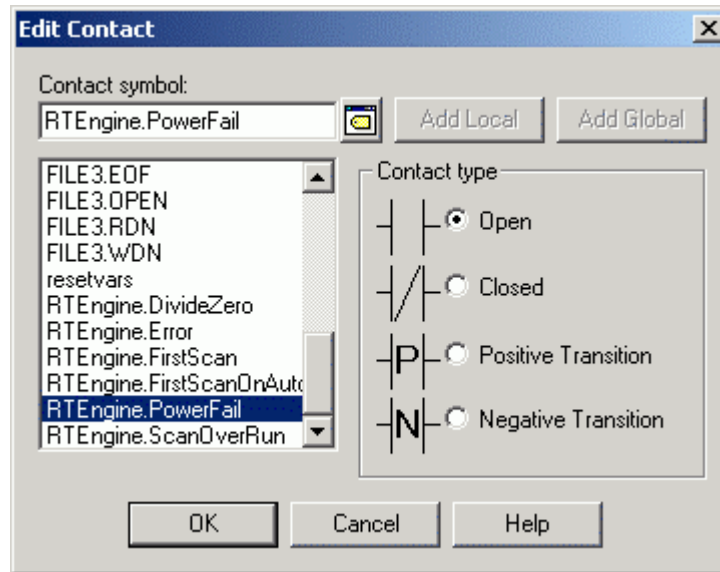


Figura 11. Cuadro de dialogo de edición de contactos.

4. Si usted ya ha definido los nombres de las variables para su sistema, pulse sobre el nombre de la variable que representa el contacto (**RTEngine.PowerFail** en la figura).

Si usted no ha definido el nombre de la variable para este contacto, ingrese un nombre en el campo **Contact Symbol**. Pulse en **Add Local** o **Add Global** para adicionar el nuevo nombre de la variable al **Symbol Manager** como variable local o global. Para mas información acerca de la definición de variables ver **“CREACIÓN DE VARIABLES”** en el capítulo **“DEFINICIÓN DE VARIABLES”**.

Si usted pulsa **OK** sin adicionar el símbolo al **Symbol Manager**, el editor acepta el nombre, pero usted, a pesar de esto debe adicionar la variable al **Symbol Manager** antes de que el programa sea compilado.

5. Seleccione el tipo de contacto (Abierto, Cerrado, etc.) y pulse **OK**. Aparecerá el nuevo contacto en el escalón, en la localización que usted especificó.

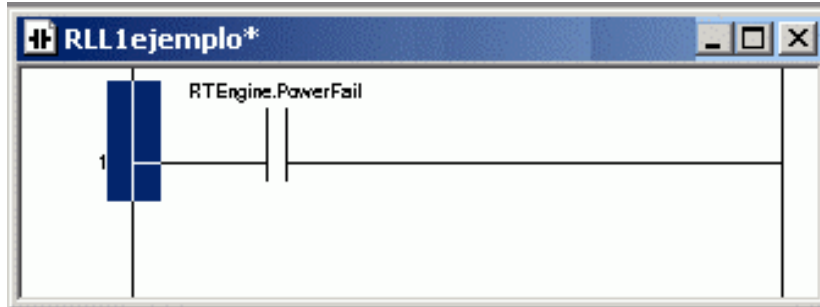


Figura 12. Presentación del nuevo contacto en el programa.

Nota: Usted puede crear un contacto que siempre sea TRUE o siempre sea FALSE. Ingrese cualquiera de los dos, TRUE o FALSE en el campo Contact Symbol y pulse OK. El contacto operara como si este fuera TRUE (encendido) o FALSE (apagado) forzado.

Adición de una bobina

- Para adicionar una bobina a un programa:

1. Pulse en **Coil Tool** sobre la barra de herramientas RLL.

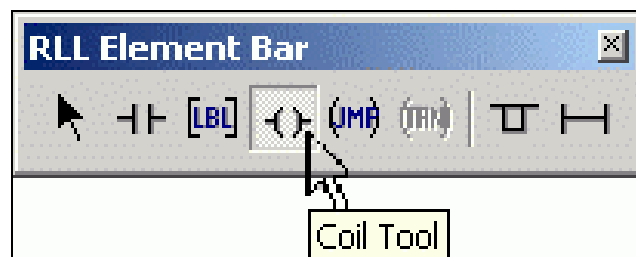


Figura 13. Adición de una bobina al programa.

2. Mueva el cursor a la localización sobre el escalón donde usted quiere colocar la nueva bobina.

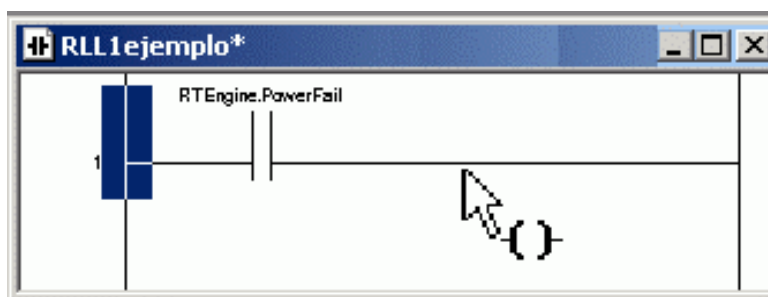


Figura 14. Ubicación de la bobina en el escalón.

3. Pulse el botón izquierdo del mouse. Aparecerá el cuadro de dialogo **Edit Coil**.

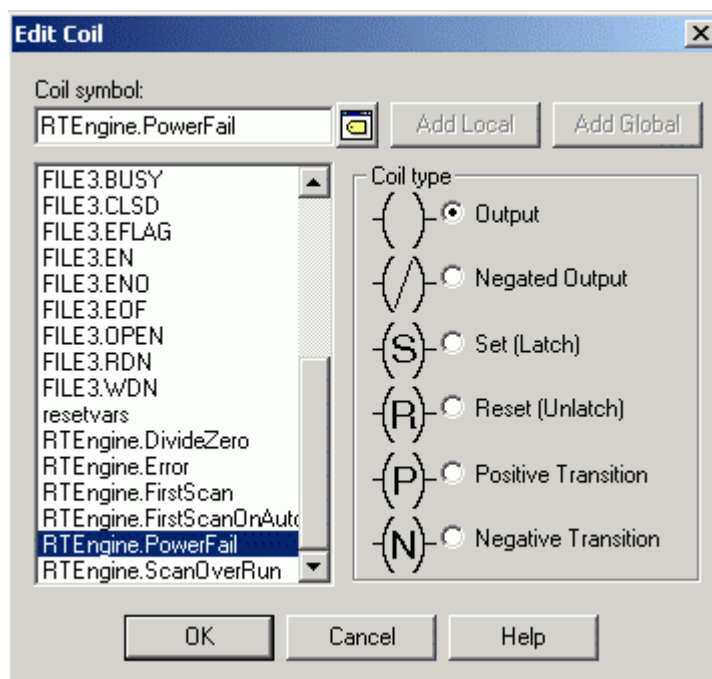


Figura 15. Cuadro de dialogo de edición de bobinas.

4. Si usted ya ha definido los nombres de las variables para su sistema, pulse sobre el nombre de la variable que representa la bobina (**RTEngine.PowerFail** en la figura).

Si usted no ha definido el nombre de la variable para esta bobina, ingrese un nombre en el campo **Coil Symbol**. Pulse en **Add Local** o **Add Global** para adicionar el nuevo nombre de la variable al **Symbol Manager** como variable local o global. Si usted pulsa **OK** sin adicionar el símbolo al **Symbol Manager**, el editor acepta el nombre, pero usted, a pesar de esto debe adicionar la variable al **Symbol Manager** antes de que el programa sea compilado.

5. Seleccione el tipo de bobina (Salida, Salida Negada, etc.) y pulse **OK**. Aparecerá la nueva bobina en el escalón, en la localización que usted especifico.

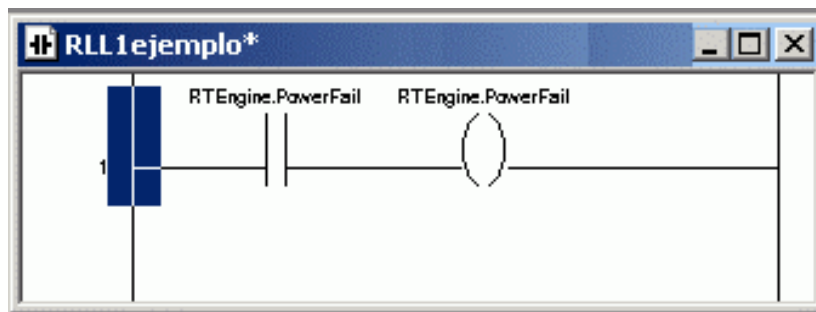


Figura 16. Presentación de la nueva bobina en el programa.

Usted puede localizar una bobina de salida en cualquier escalón, incluyendo a la derecha de una bobina de entrada. Una bobina de salida almacena el resultado de la lógica evaluada mas arriba de su localización en el escalón.

Adición de un escalón

- Para adicionar un escalón a un programa:

1. Pulse en **Rung Tool** sobre la barra de herramientas RLL.



Figura 17. Adición de un escalón al Programa.

2. Mueva el cursor a la localización sobre la barra de energía izquierda donde usted quiere insertar el nuevo escalón.

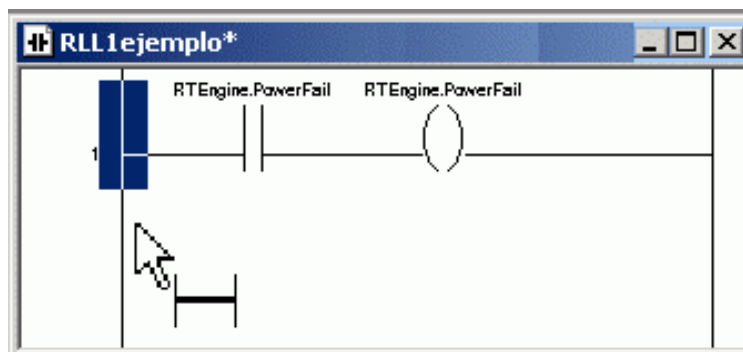


Figura 18. Ubicación de un nuevo escalón.

3. Pulse el botón izquierdo del mouse. El editor insertara el escalón en la localización especificada.

Adición de un ramal o derivación OR

- Para adicionar un ramal OR a un programa:

1. Pulse en **OR Branch Tool** sobre la barra de herramientas RLL.



Figura 19. Adición de un ramal OR al programa.

2. Mueva el cursor a la localización sobre el escalón donde usted quiere insertar el ramal OR.

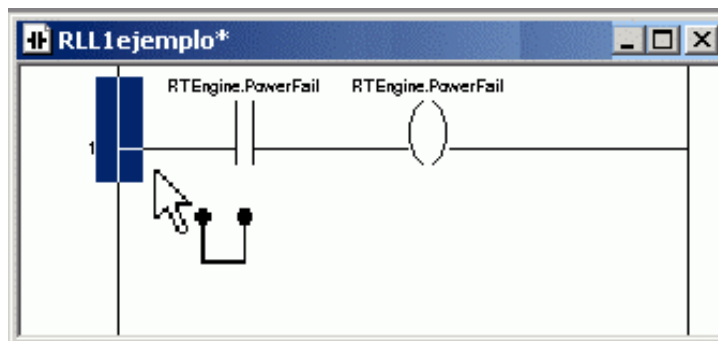


Figura 20. Ubicación del ramal OR en el programa.

3. Pulse el botón izquierdo del mouse. El editor insertara el ramal OR en la localización especificada.

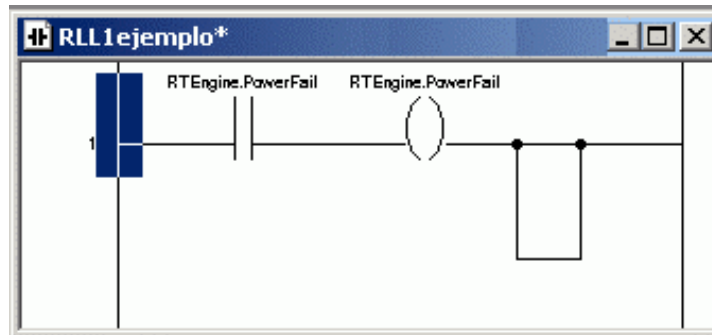


Figura 21. Presentación del ramal OR en el programa.

Después de insertar el ramal o derivación OR, usted puede ajustar los puntos de contacto que necesite.

➤ **Para mover los puntos de contacto de un ramal OR:**

1. Pulse en **Select Tool**.

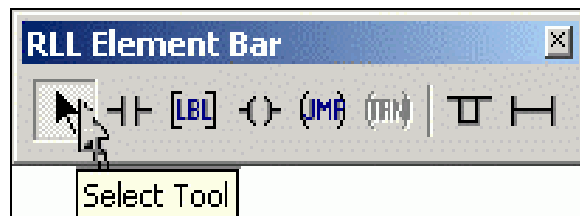


Figura 22. Herramienta de selección de un elemento de programa.

2. Pulse en el punto de contacto que usted quiere mover.
3. Arrastre el punto de contacto a la nueva localización en el escalón. El editor conecta el ramal OR a la nueva localización en el escalón.

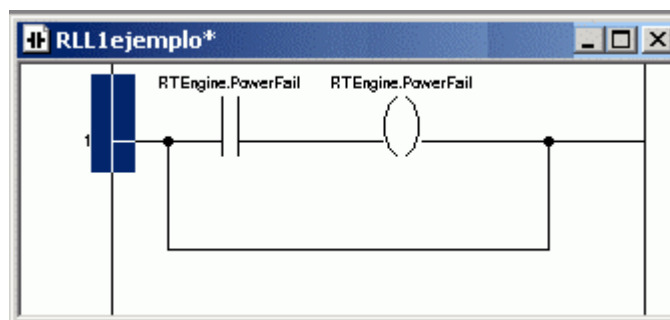


Figura 23. Nueva ubicación del punto de contacto del ramal OR.

Para determinar donde usted puede mover un punto de contacto, localice el cursor sobre el punto de contacto y haga doble clic. Una serie de signos de interrogación aparecerán para mostrar los lugares validos en el escalón, como se muestra en la siguiente figura. Esto puede ser muy útil cuando un ramal OR esta anidado.

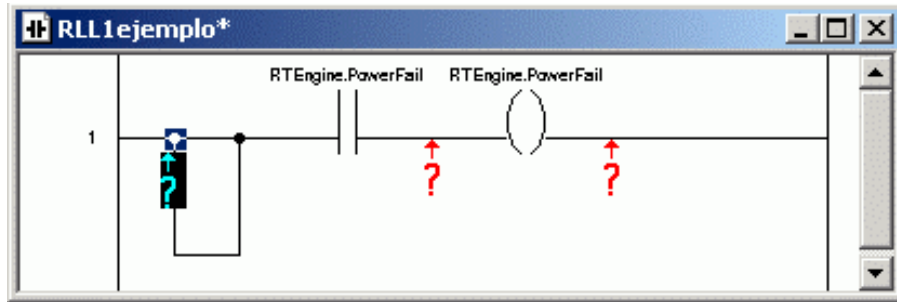


Figura 24. Lugares validos en el escalón para mover un punto de contacto.

Pulse en uno de los signos de interrogación para regresar al modo normal de edición. El punto de contacto seleccionado se mueve al lugar del signo de interrogación seleccionado.

Como una mejora de la especificación IEC 61131-3, usted puede insertar un ramal OR que no contenga lógica (una desviación). Usted puede usar una desviación para desactivar temporalmente una parte de la lógica sin borrarla del programa. Se usa con un contacto que corta el flujo de energía a la lógica en cuestión, la desviación mantiene el flujo de energía a través del resto del escalón. Esta característica es muy útil para corregir errores en los programas. Usted también puede mover un punto de contacto de un escalón a otro sin tener que borrar la lógica contenida dentro del ramal OR.

Nota: Si usted edita un escalón RLL contenido dentro una transición SFC RLL, usted no puede adicionar un segundo escalón. Para mas información sobre SFC, ver él capítulo “**ELEMENTOS DE PROGRAMAS SFC**”.

Eliminación de un ramal o bifurcación OR

La operación de **Cut Tool** esta basada en seleccionar un objeto, para luego pulsar en **Cut Tool** para borrar el objeto. Para borrar un ramal OR, siga uno de estos procedimientos.

➤ **Para eliminar un ramal OR Que no contenga elementos:**

1. Pulse en **Select Tool**.

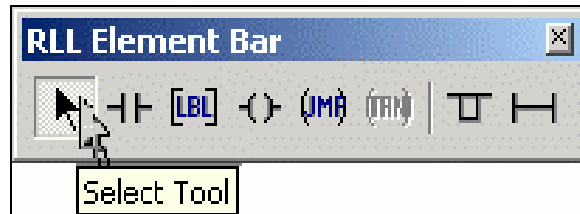


Figura 25. Herramienta para seleccionar un elemento de programa.

2. Mueva el cursor a la mitad del ramal OR y pulse.

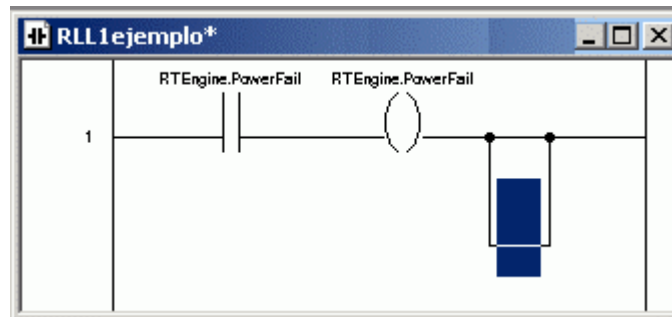


Figura 26. Selección de un ramal OR

3. Pulse en **Cut Tool** para borrar el ramal OR.



➤ **Para eliminar un ramal OR que contiene uno o más elementos:**

1. Pulse en **Select Tool**.

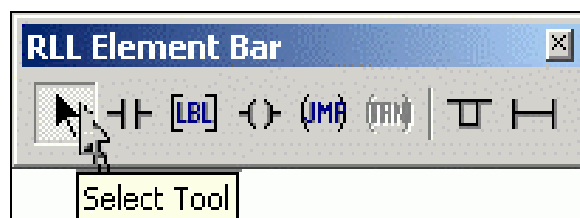


Figura 27. Herramienta de selección de un elemento de programa.

2. Arrastre un área que incluya el ramal OR completo y sus puntos de conexión.

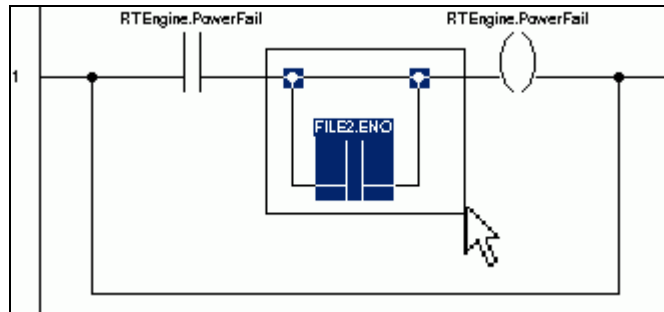


Figura 28. Selección de un ramal OR

3. Pulse en **Cut Tool** para borrar el ramal OR.



Adición de una etiqueta

Usted debe usar la bobina de salto y la etiqueta juntas. Una bobina de salto sin una etiqueta causara un error cuando usted compile el programa.

➤ Para adicionar una etiqueta a un programa:

1. Pulse en **Label Tool** sobre la barra de herramientas RLL.

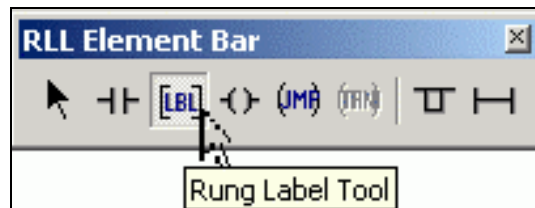


Figura 29. Adición de una etiqueta al programa.

- Mueva el cursor a la localización donde usted quiere insertar la nueva etiqueta.

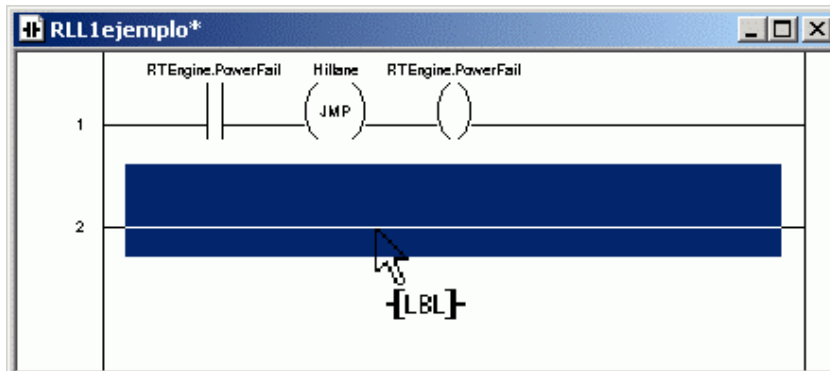


Figura 30. Ubicación de la nueva etiqueta en el programa.

- Pulse el botón izquierdo del mouse. Aparecerá el cuadro de dialogo **Edit Rung Label**.

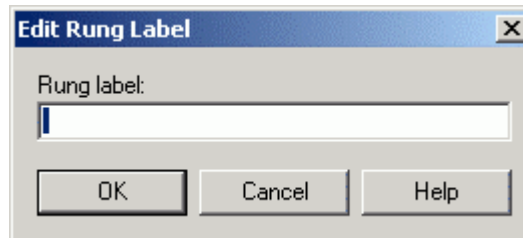


Figura 31. Cuadro de dialogo de edición de etiquetas.

- Ingrese una etiqueta y pulse **OK**. El nombre de la etiqueta no puede contener ningún espacio.

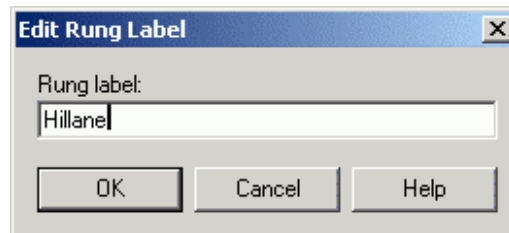


Figura 32. Definición de la etiqueta.

El editor insertara la etiqueta en el lugar especificado.

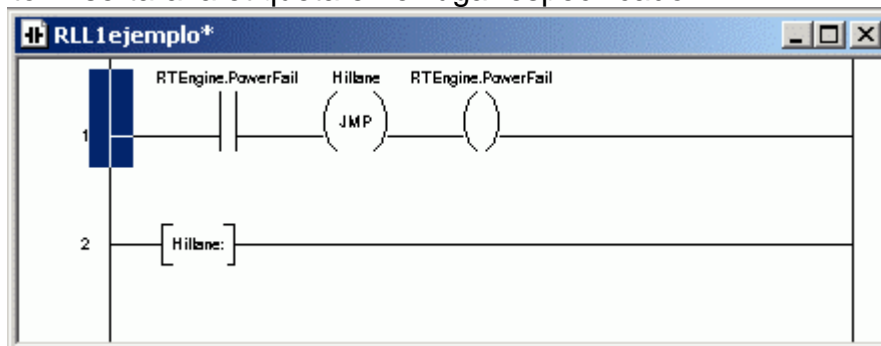


Figura 33. Presentación de la nueva etiqueta en el programa.

Adición de una bobina de salto

Usted debe usar la bobina de salto y la etiqueta juntas. Una bobina de salto sin una etiqueta causara un error cuando usted compile el programa.

➤ **Para adicionar una bobina de salto a un programa:**

1. Pulse en **Jump Coil Tool** sobre la barra de herramientas RLL.

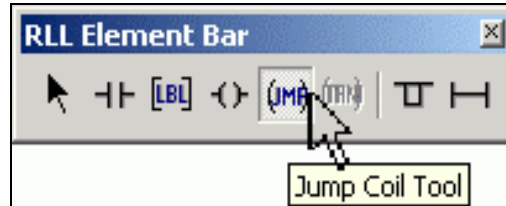


Figura 34. Adición de una bobina de salto al programa.

2. Mueva el cursor a la localización donde usted quiere insertar la nueva bobina de salto.

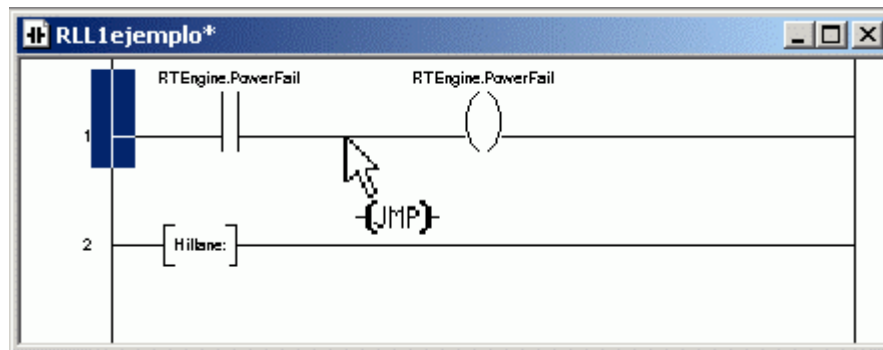


Figura 35. Ubicación de la bobina de salto en el escalón.

3. Pulse el botón izquierdo del mouse. Aparecerá el cuadro de dialogo **Edit Jump Coil**.

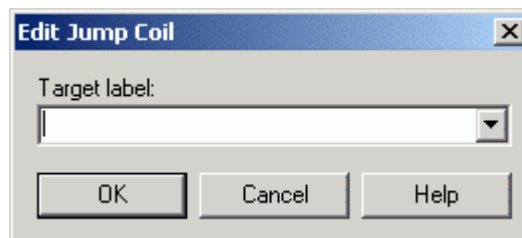


Figura 36. Cuadro de dialogo de edición de bobinas de salto.

4. Ingrese la etiqueta destino y pulse **OK**. El nombre de la etiqueta destino no puede contener ningún espacio.

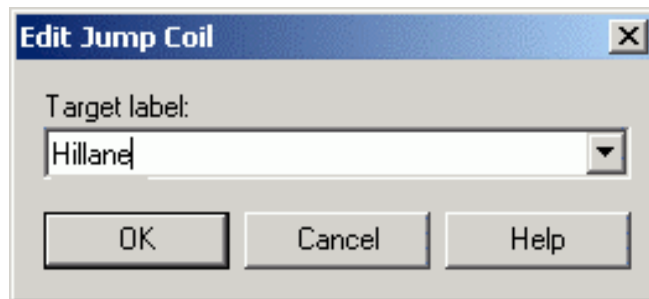


Figura 37. Definición de la bobina de salto.

El editor insertará la bobina de salto en el lugar especificado.

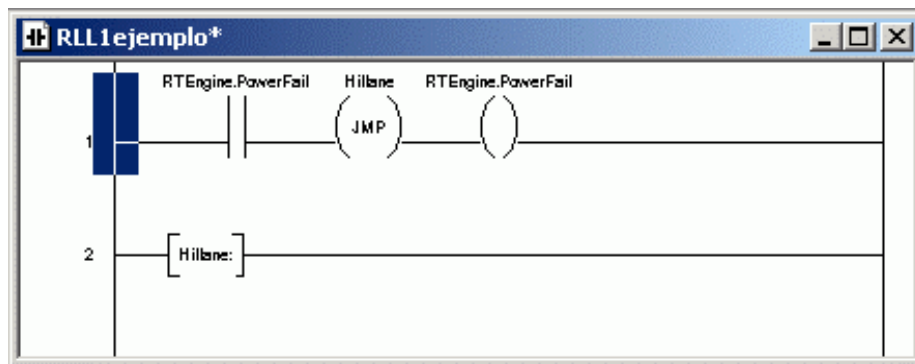


Figura 38. Presentación de la bobina de salto en el programa.

Adición de un bloque de función

El InControl provee varios algoritmos predefinidos llamados bloques de funciones que usted puede usar en un programa RLL. Usted puede habilitar bloques de funciones con entradas desde un escalón RLL, permitiendo hacer operaciones tales como trigonométricas, matemáticas, funciones lógicas, operaciones con desplazamiento de bits, operaciones con archivos, etc., y luego enviar los resultados a una salida que los almacena dentro de otro elemento en el escalón RLL.

➤ **Para adicionar un bloque de función a un programa:**

1. Si la paleta Bloque de Función no esta siendo mostrada, pulse en **Function Block Palette** en el menú **View**.

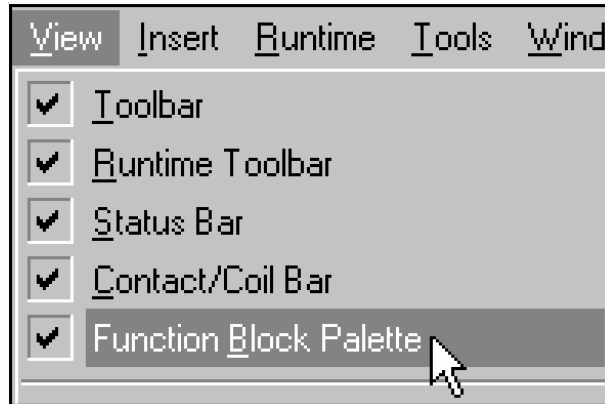


Figura 39. Habilitación de la Paleta de bloques de función.

El editor muestra la Paleta de Bloques de Funciones.

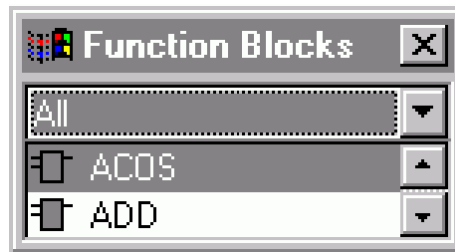


Figura 40. Paleta de bloques de funciones.

2. Seleccione el bloque de función específico que usted quiere adicionar, tal como un bloque OR.

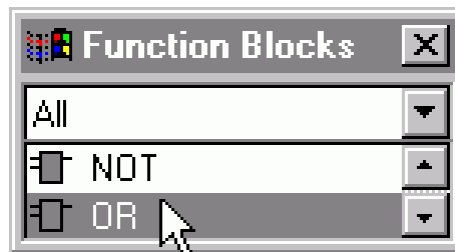


Figura 41. Selección del bloque de función OR.

3. Arrastre el bloque de función hacia el escalón.

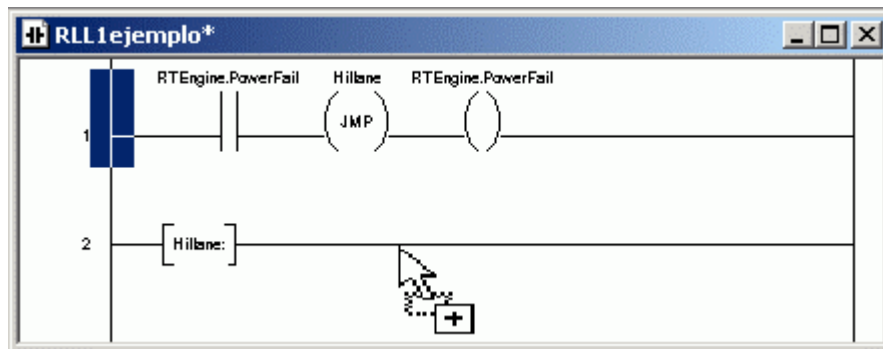


Figura 42. Ubicación del bloque de función en el escalón.

Aparecerá el cuadro de dialogo del bloque. Por ejemplo en la siguiente figura se muestra el cuadro de dialogo del Bloque de Función OR.

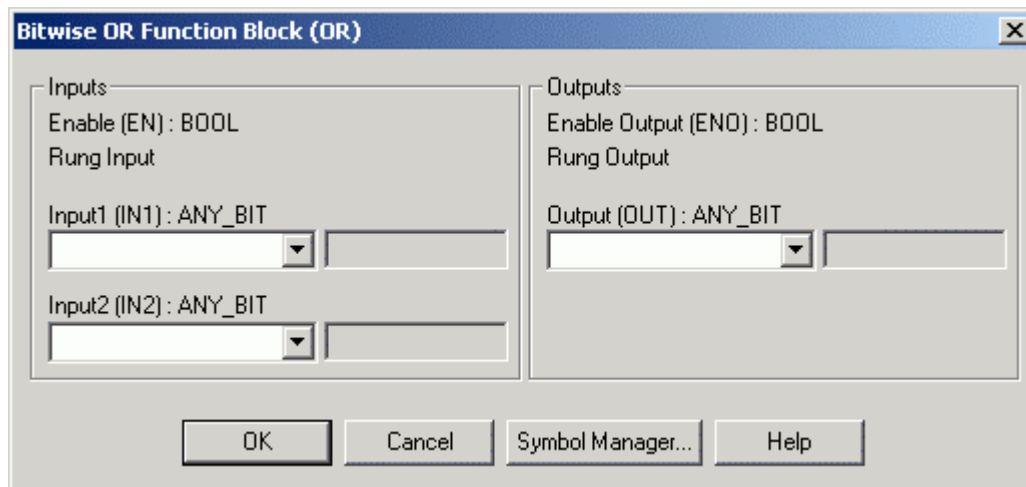


Figura 43. Cuadro de dialogo del bloque de función OR.

4. Llene la información apropiada para el bloque de función.
5. Cuando usted ha finalizado de llenar el cuadro de dialogo, pulse **OK**. El editor insertara el bloque en el lugar especificado.

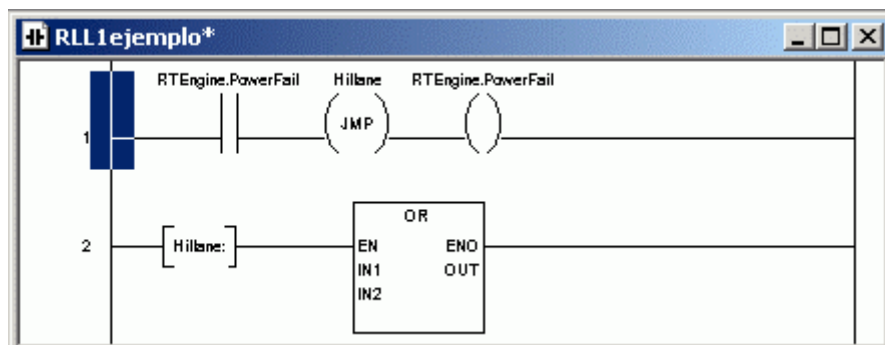


Figura 44. Presentación del nuevo bloque de función OR.

Adición de una bobina de transición SFC

La bobina de transición SFC elemento de programa RLL que usted debe usar solamente bajo condiciones específicas (dentro de una acción SFC) en un programa SFC. Para más información acerca de bobinas de transición y SFC's, ver el capítulo "ELEMENTOS DE PROGRAMAS SFC". Para adicionar una bobina de transición SFC a un programa, usted debe editar una Acción en un programa SFC. Con una Acción abierta para editar, siga estos pasos.

1. Pulse en **SFC Transition Coil Tool** sobre la barra de herramientas RLL.

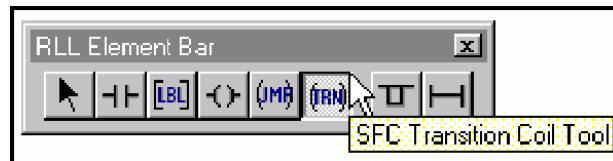


Figura 45. Adición de una bobina de transición SFC al programa.

2. Mueva el cursor a la localización sobre el escalón donde usted quiere colocar la bobina de transición SFC.

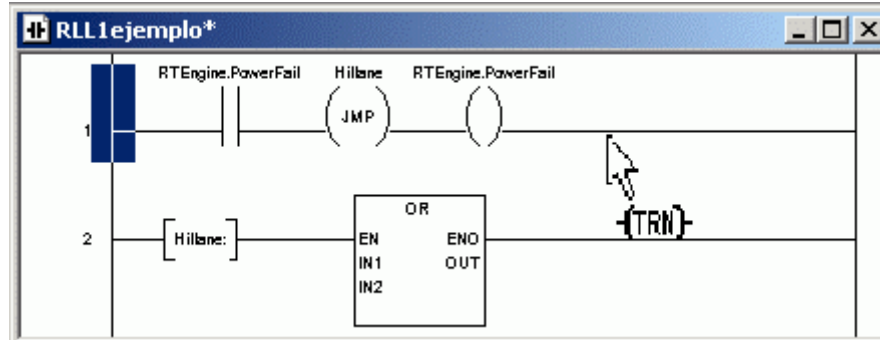


Figura 46. localización de la nueva bobina de transición SFC.

3. Pulse el botón izquierdo del mouse. Aparecerá el cuadro de dialogo **Edit SFC Transition Coil**.

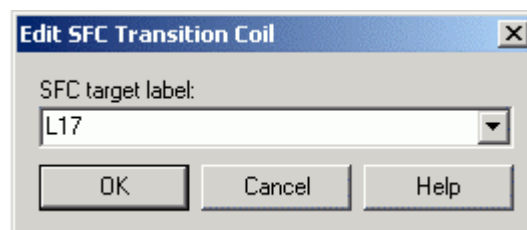


Figura 47. Definición de la bobina de transición SFC.

4. Ingrese el nombre del SFC destino y pulse **OK**. El nombre de la etiqueta SFC destino no puede contener ningún espacio. El editor insertara la bobina de transición SFC en el lugar especificado

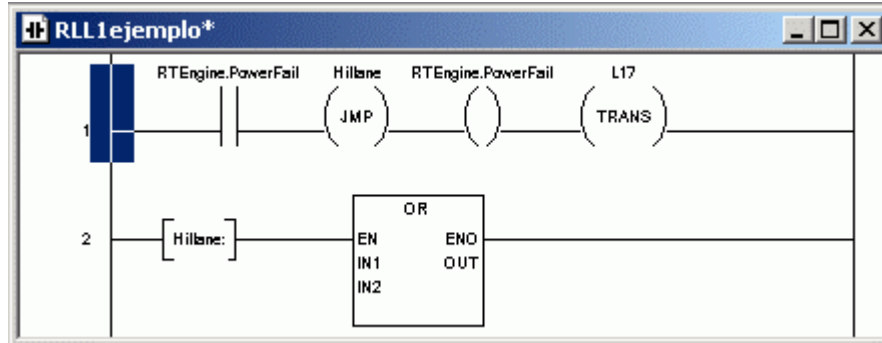


Figura 48. Presentación de la bobina de transición SFC en el programa.

EDICIÓN DE ELEMENTOS DE PROGRAMA

- Para editar un elemento de programa existente:

1. Pulse en **Select Tool**.

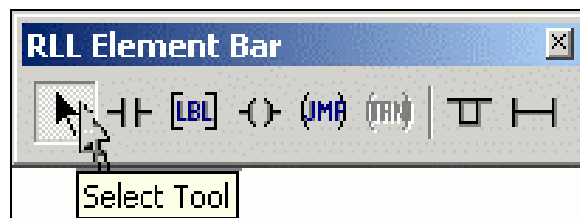


Figura 49. Herramienta de selección de un elemento de programa.

2. Haga doble clic sobre el elemento (contacto, bobina, bloque de función, etc.). Aparecerá el cuadro de dialogo apropiado para el elemento (**Edit Contact**, **Edit Jump Coil**, **Bitwise OR**, etc.). Si un elemento es resaltado, usted también puede abrir el cuadro de dialogo de edición, presionando la tecla **Enter**.
3. Haga cambios en el cuadro de dialogo requerido.

ADICIÓN DE COMENTARIOS

Usted puede ingresar un comentario descriptivo para cada escalón en el programa. Los comentarios pueden tener varias líneas si es necesario.

➤ **Para ingresar un comentario para un escalón:**

1. Abra un programa RLL. Si la opción **Program Comments** ha sido seleccionada en el menú **View**, aparecerá el texto “**Rung Comment**(Comentario del escalón)” entre cada escalón.

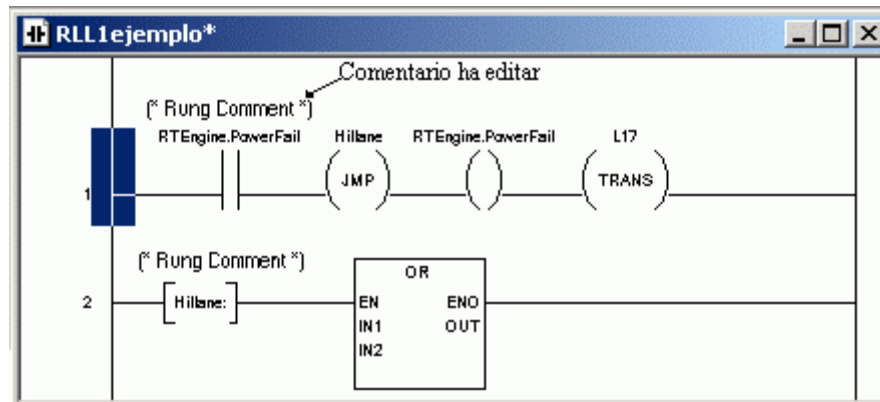


Figura 50. Adición de comentarios al programa.

2. Si el texto “**Rung Comment**” no aparece, pulse en **Program Comments** sobre el menú **View**. Aparecerá el texto “**Rung Comment**” entre cada escalón en el programa que no tenga un comentario.
3. Haga doble clic sobre el texto “**Rung Comment**” que usted quiere editar. Aparecerá el cuadro de dialogo **Program Comment**.

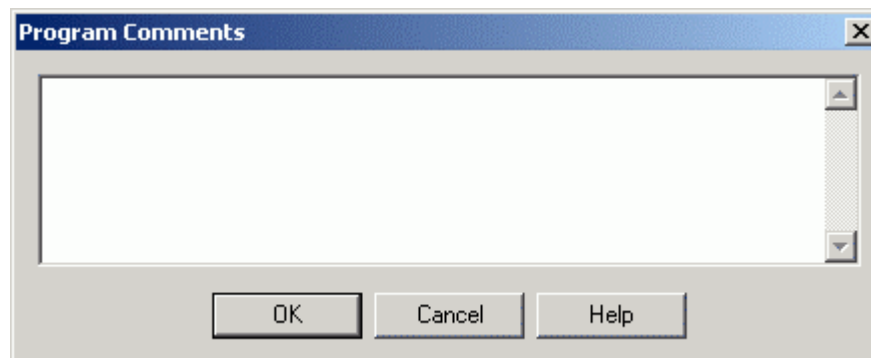


Figura 51. Cuadro de edición de comentarios del programa.

- 4. Ingrese el comentario y pulse **OK**. Su comentario aparecerá dentro del programa.

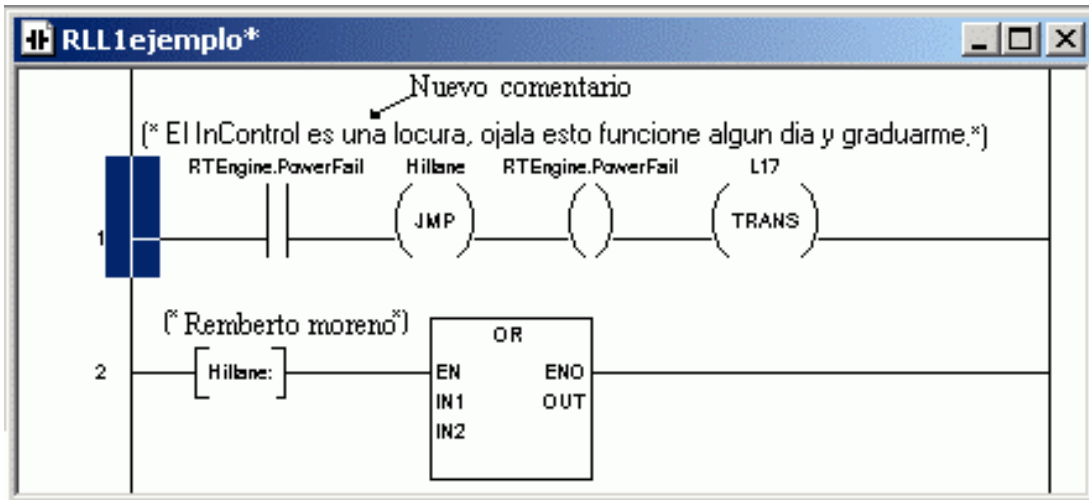


Figura 52. Presentación de los comentarios en el programa.

ELEMENTOS DE PROGRAMAS RLL

Este capítulo describe los elementos de programación que usted usara en un programa RLL.

CONTENIDO

- FLUJO DE NERGIA EN PROGRAMA RLL.
- TIPOS DE CONTACTOS RLL.
- TIPOS DE BOBINAS RLL.
- BOBINAS DE SALTOS / ETIQUETAS.
- BOBINA DE TRANSICION SFC.
- BLOQUES DE FUNCIONES.

This product is licensed to:

Company:	N/A
SN:	N/A
Expires:	N/A
Locked To:	N/A

7.1.0.1337

FLUJO DE ENERGIA EN UN PROGRAMA RLL

Esta sección describe como InControl revisa o analiza un programa RLL y resuelve su lógica.

Resolución De Una Lógica Sencilla De Contactos Y Bobinas

Un programa RLL esta compuesto de dos líneas verticales (barras de energía) que están conectados por una o mas líneas horizontales (escalones RLL), la barra izquierda representa la fuente de energía y la barra derecha representa el sumidero o línea de descarga de energía.

Los elementos básicos de los programas RLL, contactos y bobinas son ubicados en los escalones y representan los componentes actuales de hardware (interruptores de limite, bobinas solenoides, luces, etc.) y bits de localización de memoria, el sistema lee las entradas y la salidas físicas, escribe en estas y resuelve la lógica RLL. el flujo de energía y la solución de la lógica de un programa RLL, dentro de un programa RLL es siempre desde la parte superior hasta la parte inferior y de izquierda a derecha.

Bloques de funciones y flujo de Energía

Los bloques de función RLL son paquetes programados que pueden también ser ubicados en un escalón RLL. ellos proveen un mecanismo para resolver

problemas mas complejos, de difícil manejo por medio de contactos y bobinas, operaciones matemáticas, funciones lógicas, temporizadores, contadores, etc. Las entradas de los bloques de funciones reciben un flujo de energía desde el escalón y lo transfieren al siguiente elemento del escalón a través de sus salidas.

TIPOS DE CONTACTOS RLL

El contacto típicamente representa un punto de entrada discreto, semejante a un interruptor de limite. Un contacto puede también representar una localización de interna memoria y como tal, es representado en términos booleanos, puede tener uno de dos estados: TRUE o FALSE. Usted representa en sus programas al contacto por medio de un símbolo o variable asignado en el Symbol Manager; InControl soporta cuatro tipos de contactos descritos a continuación:

Contacto Abierto

El contacto normalmente abierto opera de la siguiente forma:

- El contacto permite el flujo de energía si el punto que el representa es TRUE.
- El contacto no permite el flujo de energía si el punto que el representa es FALSE.

Contacto Cerrado

El contacto normalmente cerrado opera de la siguiente forma:

- El contacto permite el flujo de energía si el punto que el representa es FALSE.
- El contacto no permite el flujo de energía si el punto que el representa es TRUE.

Contacto de Transición Positiva

El contacto de transición positiva opera de la siguiente forma:

- El contacto el estado del contacto es evaluado, el estado actual del punto que el representa es comparado por un análisis previo. Si el punto es TRUE en el análisis actual, pero fue FALSE en la exploración anterior, el contacto permite el flujo de energía de lo contrario no permite el flujo de energía.
- El contacto no permite el flujo de energía de nuevo hasta que el punto que este representa pase de FALSE a TRUE de nuevo.

Contacto de Transición Negativa

El contacto de transición negativa opera de la siguiente forma:

- El contacto el estado del contacto es evaluado, el estado actual del punto que el representa es comparado por un análisis previo. Si el punto es FALSE en el análisis actual, pero fue TRUE en la exploración anterior, el contacto permite el flujo de energía, de lo contrario no permite el flujo de energía.
- El contacto no permite el flujo de energía de nuevo hasta que el punto que este representa pase de TRUE a FALSE de nuevo.

TIPOS DE BOBINAS RLL

La bobina representa un punto de salida discreto, semejante a un solenoide. Una ventaja con respecto al resto de los lenguajes basados en la especificación IEC61131-3 es que se pueden ubicar múltiples bobinas en un solo escalón y el estado de una bobina no afecta el estado de las otras. El InControl soporta seis tipos de bobinas las cuales son descritas a continuación:

Bobina de Salida

Estas bobinas operan de la siguiente manera:

- La bobina ajusta el punto que esta representa a TRUE cuando la bobina tiene un flujo de energía.
- La bobina ajusta el punto que ella representa a FALSE cuando ella no tiene flujo de energía.

Bobina de Salida Negativa

Estas bobinas operan de la siguiente manera:

- La bobina ajusta el punto que esta representa a TRUE cuando la bobina no tiene un flujo de energía.
- La bobina ajusta el punto que ella representa a FALSE cuando ella tiene flujo de energía.

Bobina de Cierre (Latch)

Estas bobinas operan de la siguiente manera:

- La bobina ajusta el punto que esta representa a TRUE cuando la bobina tiene un flujo de energía.
- El punto continua en TRUE (el punto es ajustado o fijado) siempre y cuando la bobina no tenga flujo de energía mayor.
- El punto que ella representa puede ser ajustado a FALSE por un reset en la bobina.

Bobina de Reset (Unlatch)

Estas bobinas operan de la siguiente manera:

- La bobina ajusta el punto que esta representa a FALSE cuando la bobina tiene un flujo de energía.
- El punto permanece en FALSE (el punto es reseteado) siempre y cuando la bobina no tenga flujo de energía mayor.
- El punto que ella representa puede ser ajustado a TRUE por un reset en la bobina.

Bobina de Transición Positiva

Estas bobinas operan de la siguiente manera:

- Cuando el estado de la bobina es evaluado, el estado del flujo de energía dentro de la bobina durante este análisis es comparado con su estado en la análisis previo. Si el flujo de energía es TRUE en el actual análisis, pero fue

FALSE en el análisis previo, los pulsos de la bobina ajustan el punto que esta representa a TRUE.

- El punto permanece en TRUE, a menos que sea ajustado a FALSE, por la duración del análisis de la lógica en escalera.
- Después que la bobina pulso, el flujo de energía debe ser FALSE por lo menos un análisis antes que la bobina pueda pulsar de nuevo.

Bobina de Transición Negativa

Estas bobinas operan de la siguiente manera:

- Cuando el estado de la bobina es evaluado, el estado del flujo de energía dentro de la bobina durante este análisis es comparado con su estado en la análisis previo. Si el flujo de energía es FALSE en el actual análisis, pero fue TRUE en el análisis previo, los pulsos de la bobina ajustan el punto que esta representa a TRUE.
- El punto permanece en TRUE, a menos que sea ajustado a FALSE, por la duración del análisis de la lógica en escalera.
- Después que la bobina pulso, el flujo de energía debe ser TRUE por lo menos un análisis antes que la bobina pueda pulsar de nuevo.

BOBINA DE SALTOS / ETIQUETAS

Use los elementos bobinas de salto y etiquetas para deshabilitar secciones de código de programas temporalmente. Usted debe usar la bobina de salto y etiqueta juntos . una bobina de salto sin una etiqueta causara un error cuando usted compile el programa. La bina de salto y al etiqueta operan de la siguiente manera:

- Cuando la bobina recibe el flujo de energía, la ejecución del programa ignora toda la lógica entre el salto y su correspondiente etiqueta.
- Cuando la bobina de salto es activada se detiene la lógica, las salidas entre la bobina de salto y la etiqueta no son activadas.
- Toda la lógica entre la bobina de salto y la etiqueta es ejecutada normalmente cuando al bobina de salto no recibe flujo de energía.
- La ejecución del programa no puede saltar hacia atrás a un escalón previo.

BOBINA DE TRANSICIÓN SFC

Esta bobina de transición SFC es un elemento de programas RLL que solamente se puede usar bajo ciertas condiciones especificas: dentro de un programa SFC, dentro de una acción que esta asociada con un Paso o un Macro Paso. La bobina de transición SFC, se ejecuta al igual que una bobina de salto.

Bloques de Funciones

Grupo	Tipo	Descripción
BITWISE (operaciones con bit)	AND	Calcula el AND de bits de dos números.
	NOT	Calcula el complemento en bits de un numero.
	OR	Calcula el OR de dos números.
	ROR	Rota la entrada a la derecha para él numero de bits especificado por él numero de traslado.
	ROL	Rota la entrada a la izquierda para él numero de bits especificado por él numero de traslado.
	SHL	Invierte la entrada a la izquierda para él numero de bits especificado por él numero de traslado.
	SHR	Invierte la entrada a la derecha para él numero de bits especificado por él numero de traslado.
	XOR	Calcula el OR exclusivo en bits de dos números.
COMPARACIÓN	EQ	Examina la igualdad de dos números.
	GE	Examina si la primera entrada es más grande que o igual a la segunda entrada.
	GT	Examina si la primera entrada es más grande que la segunda entrada.
	LE	Examina si la primera entrada es menor que o igual a la segunda entrada.
	LT	Examina si la primera entrada es menor que la segunda entrada.
	NE	Examina la igualdad de dos entradas.
CONVERSION	BATOS	Toma el arreglo de bytes de entrada y los almacena en una cadena de caracteres.
	ITOA	Convierte un numero entero de entrada en una cadena ASCII.
	RTOA	Convierte un numero real de entrada en una cadena ASCII.
	STOBA	Toma una cadena de caracteres de entrada y los almacena en un arreglo de bytes.
COUNTER (Contador)	CTD	Cuenta eventos por decremento en uno.
	CTE	Cuenta eventos por incremento en uno.
	CTUD	Cuenta eventos de arriba hacia abajo.
	FCOPY	Copia un archivo.
	FDEL	Borra un archivo.
	FNEW	Crea un nuevo archivo.
	FOPEN	Abre un archivo existente.
	FREAD	Lee datos desde un archivo.

Grupo	Tipo	Descripción
COUNTER (Contador)	FRWND	Devuelve un archivo al comienzo.
	FWRIT	Escribe datos en un archivo.
MATH (Matemáticos)	ABS	Calcula el valor absoluto de un numero.
	ADD	Suma dos números.
	DIV	Divide un numero por otro numero.
	EXPT	Eleva el primer numero a la potencia especificada por el segundo numero.
	MOD	Divide un numero por otro y almacena el residuo.
	MOVE	Copia un dato desde una localización a otra.
	MUL	Multiplica dos números.
	NEG	Niega (invierte) el numero de entradas.
	SQRT	Calcula la raíz3 cuadrada de un numero.
	SUB	Resta un numero a otro.
MISCELANEA	ABTAL	Aborta o cierra todos los programas que se están ejecutando.
	MSGW	Muestra un mensaje en la ventana de salida.
STRING (Cadena)	CAT	Concatena la segunda cadena de entrada con el final de la segunda cadena de entrada.
	DEL	Borra caracteres desde la mitad de la cadena de entrada.
	FIND	Busca una cadena de entrada dentro de otra cadena.
	INS	Inserta una cadena de entrada dentro de otra cadena de entrada.
	LEFT	Copia el carácter mas a la izquierda de la cadena de entrada.
	LEN	Almacena la longitud de una cadena de entrada.
	MID	Copia caracteres desde la mitad de la cadena de entrada.
	RGHT	Copia el carácter mas a la derecha de la cadena de entrada.
	RPLC	Reemplaza caracteres en una cadena de entrada con otra cadena de entrada.
TIMER (temporizadores)	TOP	Provee un tiempo de retardo al final de un evento.
	TON	Provee un tiempo de retardo al inicio de un evento.
	TP	Activado por un pulso, provee un tiempo de retardo al final de un evento.

Grupo	Tipo	Descripción
TRIG / LOG (Trigonométricos / Logarítmicos)	ACOS	Calcula el arco coseno de un numero.
	ASIN	Calcula el arco seno de un numero.
	ATAN	Calcula el arco tangente de un numero.
	COS	Calcula el coseno de un numero.
	EXP	Calcula el logaritmo exponencial de un numero.
	LN	Calcula el logaritmo natural.
	LOG	Calcula el logaritmo en base 10 de un numero.
	SIN	Calcula el seno de un numero.
	TAN	Calcula la tangente de un numero.
TRIGGER	F_TRIG	Enciende una salida cuando hay un falla en el trigger.
	R_TRIG	Enciende una salida cuando hay un pico en el trigger.

El formato general de presentación para los bloques de funciones es el siguiente:

Campo	Tipo	Descripción
Enable (EN)	—	Escalón de entrada que habilita el escalón de entrada del bloque de funciones (BOOL) para su ejecución cuando es TRUE.
Input (IN) entrada	Any_Num	Contiene el valor o dato a procesar en la ejecución del bloque de función.
Enable Output (ENO)	—	Escalón de salida (BOOL) indica excepciones cuando es FALSE.
Output (OUT) salida	Any_Num	Contiene el resultado de la ejecución del bloque de función.

USO DEL EDITOR SFC

Este capítulo nos muestra el funcionamiento del editor **SFC** (Cartas de Funciones Secuenciales) y describe como poder usarlo para crear nuevos programas y para adicionar elementos (Pasos(Steps), Transiciones(Transitions), Saltos(Jumps), Macro Pasos(Macro Steps), etc.) a estos. Para información mas detallada acerca de estos elementos, ver él capítulo “**ELEMENTOS DE PROGRAMAS SFC**”.

CONTENIDO

- CREACIÓN DE UN PROGRAMA SFC
- HERRAMIENTAS SFC
- ADICIÓN DE ELEMENTOS DE PROGRAMA
- EDICIÓN DE ELEMENTOS DE PROGRAMA

This product is licensed to:

Company:	N/A
SN:	N/A
Expires:	N/A
Locked To:	N/A

7.1.0.1337

CREACIÓN DE UN PROGRAMA SFC

Después de iniciar el InControl, usted puede crear un nuevo programa **SFC** o editar uno existente.

➤ **Para crear un nuevo programa SFC:**

1. Sobre el menú **File**, haga clic en **New**. Aparecerá el menú de tipos de programas soportados por InControl.
2. Seleccione **SFC Program** y pulse **Aceptar**. Aparecerá el cuadro de dialogo **Save As**.
3. Escoja un nombre (Hasta 31 caracteres) y el directorio (Proyecto) para el programa y haga clic en **Save**. Aparecerá un nuevo programa SFC, mostrando un paso de inicio(Start) y un paso de finalización(End).

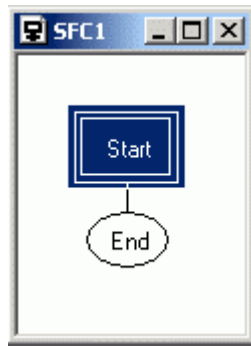


Figura 1. Presentación y definición del programa

4. Para adicionar elementos de programa, ver **“ADICIÓN DE ELEMENTOS DE PROGRAMA”** (Pagina 9-).

➤ **Para editar un programa SFC existente:**

1. Si la ventana de proyectos no esta abierta, haga clic en **Project** en el menú **View**. Aparecerá la ventana de proyectos.
2. Haga doble clic en el nombre del programa que usted quiere editar.

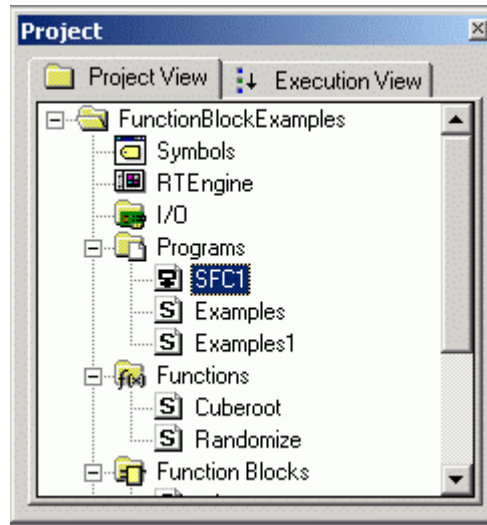


Figura 2. Presentación del programa en la Project View

Se abre el editor **SFC**, mostrando el programa seleccionado.

Usted también puede pulsar en **Open** en el menú **File** para abrir un programa existente para editarlo. Cuando aparezca el cuadro de dialogo **Open**, seleccione el programa ha abrir. Si un programa no es parte del actual proyecto, usted puede adicionarlo.

Usted puede hacer clic en **Files Into Project** en el menú **Insert** para adicionar cualquier **POU** (Programa, Función, Bloque de Función, etc.) al proyecto. En la siguiente figura, el programa **SFC1ejemplo**, que se muestra en el cuadro de dialogo **Insert Files Into Project**, es seleccionado y puede ser adicionado a **PIDApp**. Note que el archivo propiamente dicho no es copiado o movido cuando es adicionado a otro proyecto.

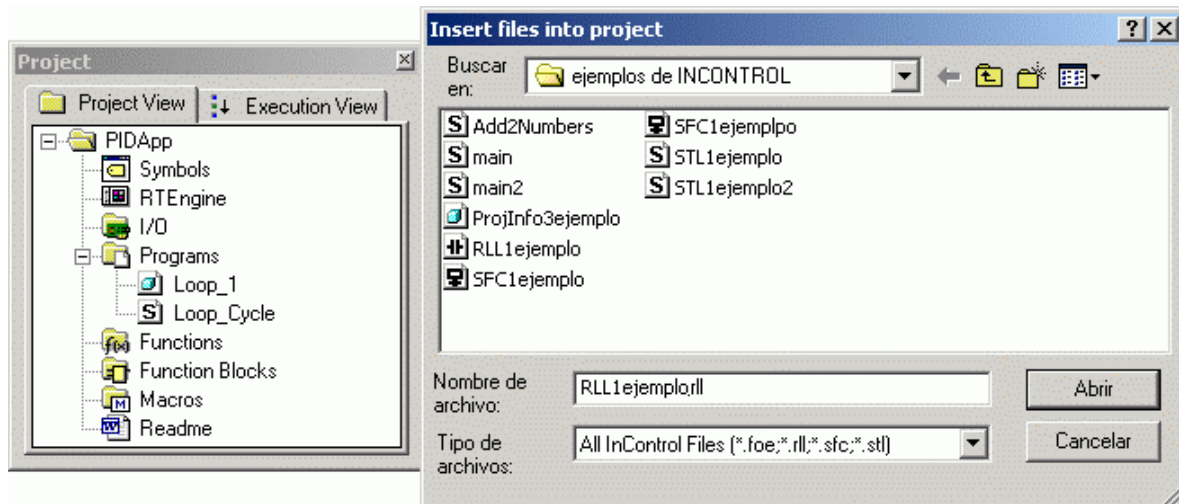


Figura 3. Adición de POU a un proyecto

Nota: Todas las **POUs** son insertadas debajo de la carpeta **Programs** de la ventana de proyectos. Usted debe mover las funciones a la carpeta **Functions**, los tipos de bloque se funciones a la carpeta **Function Block**, y los macros a la carpeta **Macros** para la correcta compilación del proyecto.

Si usted abre un proyecto desarrollado bajo **InControl 7.0**, usted tiene la opción de convertir los archivos a un proyecto en **InControl 7.1**. Cualquier macro en un proyecto aparecerá en la carpeta **Programs** después de la conversión. Usted puede mover estos macros a la carpeta **Macros**, pero esto no es necesario para compilar el proyecto.

HERRAMIENTAS SFC

Esta sección describe la barra de herramientas **SFC** y muestra algunos consejos útiles para la edición de programas.

Uso de las herramientas SFC y la barra de Menú

La barra **SFC** muestra las herramientas usadas para crear programas **SFC**.

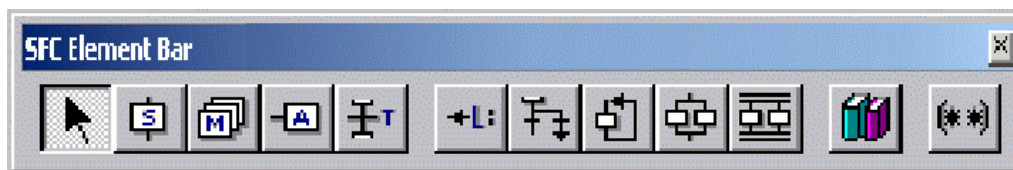



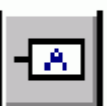










Figura 4. La barra de herramientas SFC

Las opciones de la barra de herramientas SFC son descritas en la siguiente tabla.

Iconos de la barra de herramientas SFC

Icono	Opción en la barra de menú	Función
	n/a	Le permite seleccionar elementos de programa.
	Insertar/Step	Adiciona un Step al programa.
	Insertar/Macro Step	Adiciona un Macro Step al programa.
	Insertar/ Acción	Adiciona una Acción al programa.
	Insertar/ Transición	Adiciona una Transición al programa.
	Insertar/ Etiqueta	Adiciona una etiqueta al programa.
	Insertar/Salto	Adiciona un salto al programa.
	Insertar/Lazo	Adiciona un lazo al programa.
	Insertar/Divergencia de Selección	Adiciona una Divergencia de Selección al programa.
	Insertar/Divergencia Paralela	Adiciona una Divergencia Paralela al programa.
	Insertar/Librería de Pasos	Adiciona un Paso predefinido desde la Librería de Pasos al programa.
	Insertar/Comentario	Permite adicionar comentarios al programa.

Consejos de edición

Estos consejos pueden ayudarlo a editar sus programas.

- Este capítulo describe como usar estas herramientas, basándose en selecciones que usted hará desde la barra de herramientas SFC. Usted también puede escoger estas herramientas seleccionándolas desde el menú **Insert**, el cual se muestra en la siguiente figura. Para evitar confusión, solo un método es descrito en este capítulo.

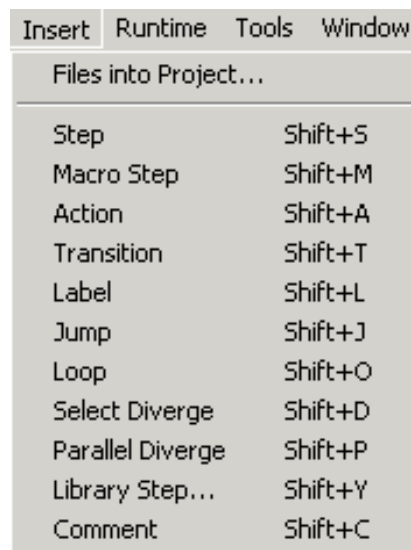


Figura 5. Uso de las herramientas SFC desde el menú Insert

Cuando usted inserta un elemento de programa desde la barra de **Menú**, el elemento es insertado en la localización actual del cursor dentro del programa. Cuando usted inserta un elemento de programa usando la barra de herramientas, usted puede mover el cursor a localización en el programa donde usted quiera colocar el elemento.

- Use el menú **View** para mostrar los objetos que usted necesita ver durante una sesión de edición. Por ejemplo, si usted prefiere adicionar elementos de programa desde la barra de **Menú**, en lugar de la barra de herramientas SFC, usted puede ocultar la barra de herramientas SFC.



Figura 6. Desactivando la barra SFC

- Durante una sesión de edición, usted puede hacer clic derecho sobre la ventana de edición SFC para mostrar rápidamente algunas de las opciones de edición que aparecen en la barra de menú.
- Para imprimir el código usado en los pasos SFC, seleccione **Structured Text** en el cuadro de dialogo **Edit Step** para los pasos apropiados. Entonces imprima el programa.

Cuando usted ingrese el código usado en los pasos SFC, use las mismas herramientas y las opciones de menú que están en el editor STL. Para mas información acerca del editor STL, ver el capítulo “**USO DEL EDITOR DE TEXTO ESTRUCTURADO**”.

ADICIÓN DE ELEMENTOS DE PROGRAMA

Esta sección describe como adicionar elementos de programa SFC usando las herramientas de la barra SFC.

Adición de un paso(Step)

Un Paso representa una condición en la cual el comportamiento del sistema sigue un conjunto de reglas definidas por las Acciones y las funciones asociadas con el Paso. Para información mas detallada acerca del uso de los Pasos, ver “**PASOS**” en el capítulo “**ELEMENTOS DE PROGRAMAS SFC**”

➤ **Para adicionar un paso a un programa:**

1. Pulse en **Step Tool** sobre la barra de herramientas SFC.



Figura 7. Adición de un Step al programa.

2. Mueva el cursor a la localización sobre el programa donde usted quiere colocar el nuevo Paso y pulse **OK**. Aparecerá el cuadro de dialogo **Edit Step**.

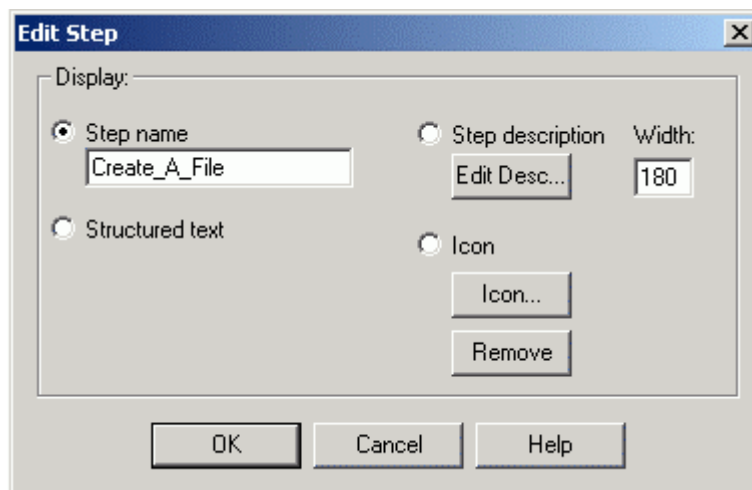


Figura 8. Definición del nuevo Paso o Step.

3. Ingrese la información apropiada para mostrar el Paso en el SFC como se describe en “PASOS” en el capítulo “ELEMENTOS DE PROGRAMAS SFC”. Luego pulse **OK**. El nuevo Paso aparecerá en el lugar que usted especifico en el programa.

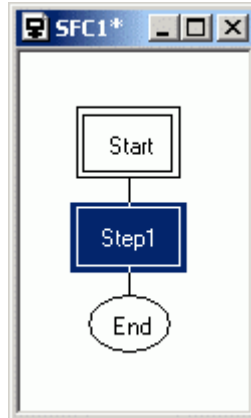


Figura 9. Presentación del nuevo paso o sep en el programa.

4. Para ingresar el código de Texto Estructurado para un Paso, haga doble clic sobre el Paso. Aparecerá la ventana de edición de Texto Estructurado.



Figura 10. Ingreso del código de Texto Estructurado para el nuevo Paso.

5. Ingrese el código del programa, descrito en el capítulo “LENGUAJE DE TEXTO ESTRUCTURADO”.

Nota: Usted también puede adicionar un Paso desde la Librería de Pasos. Un modelo de Paso contiene un código definido por el usuario que hace una función específica y un icono que es apropiado para la función. Ver “**Adición de una Librería de Pasos**” (Página 9-).

El InControl le permite proteger sus códigos de programas dentro de un Paso SFC de cambios desautorizados. Seleccione el comando **Lock Algorithms** en el menú **Edit** y asigne una clave o password. Para bloquear el código SFC, usted debe tener acceso a las tareas de seguridad de edición de programas.

Para mas información sobre bloqueo de algoritmos, ver “**BLOQUEO DE ALGORITMOS SFC**” en él capitulo “**CONFIGURACIÓN DE LA SEGURIDAD**”.

Adición de una Transición(Transition)

Una Transición representa la condición que controla la razón o causa para pasar de uno o más Pasos que preceden a la Transición hacia uno o más Pasos sucesivos que le siguen a esta. Para información detallada acerca del uso de las Transiciones en programas ver “**TRANSICIONES**” en él capitulo “**ELEMENTOS DE PROGRAMAS SFC**”.

Usted puede escoger de dos tipos de Transiciones para un programa SFC.

- La Transición RLL esta basada en un código RLL.
- La Transición Boleana esta basada en una lógica Boleana y en expresiones Boleanas de Texto Estructurado.

➤ Para adicionar una Transición RLL a un programa:

1. Si la opción **Boolean Transition** sobre el menú **Edit** esta activada, pulse sobre esta opción para desactivarla.
2. Pulse en **Transition Tool** sobre la barra de herramientas SFC.



Figura 11. Adición de una transición RLL al programa.

3. Mueva el cursor a la localización sobre el programa donde usted quiere colocar la nueva Transición y haga clic. La nueva Transición aparecerá en el programa.

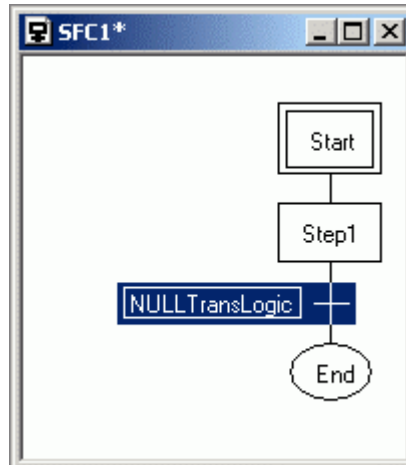


Figura 12. Presentación de la nueva transición RLL en el programa.

4. Ingrese el código RLL, haga doble clic sobre la Transición. Aparecerá el cuadro de dialogo **Select RLL Transition Logic**.

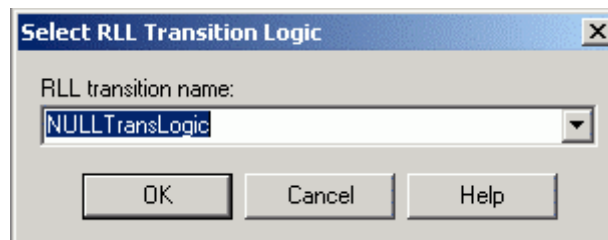


Figura 13. Definición de la nueva transición RLL.

5. Ingrese un nombre significativo para la Transición. El nombre de la Transición debe tener una longitud máxima de 100 caracteres.

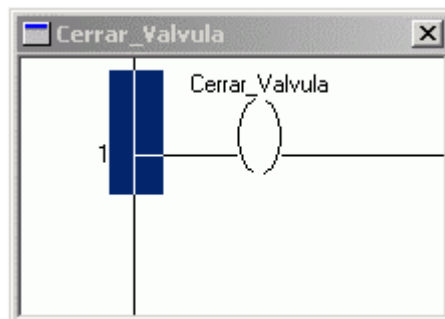



Figura 14. Adición del código RLL.

6. Adicione el código RLL, descrito en el capítulo **“ELEMENTOS DE PROGRAMAS RLL”**.
7. Cierre la ventana de edición.

➤ **Para eliminar permanentemente una Transición RLL:**

1. Seleccione la Transición en el SFC y Pulse en **Cut Tool** para borrar la Transición. 
2. Sobre el menú **Tools**, pulse en **RLL Transition Manager**. Aparecerá el cuadro de dialogo **RLL Transition Manager**.
3. Seleccione la Transición ha eliminar y pulse el botón **Delete**. La Transición es borrada del **RLL Transition Manager**.

Nota: Usted debe borrar la Transición del **RLL Transition Manager** y del SFC, para remover completamente el código de la Transición de su programa.

➤ **Para renombrar una Transición RLL existente:**

1. Sobre el menú **Tools**, pulse en **RLL Transition Manager**. Aparecerá el cuadro de dialogo **RLL Transition Manager**.
2. Seleccione la Transición ha renombrar y pulse en el botón **Rename**. La Transición es renombrada

Nota: Usted debe renombrar las Transiciones RLL individualmente, después de hacer un cambio de nombre en el **RLL Transition Manager**.

➤ **Para adicionar una Transición Boleana a un programa:**

1. Si la opción **Boolean Transition** sobre el menú **Edit** no esta activada, pulse sobre esta opción para activarla.
2. Pulse en **Transition Tool** sobre la barra de herramientas SFC.

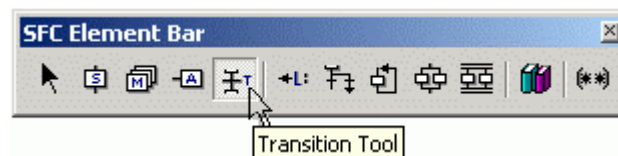


Figura 15. Adición de una transición Boleana al programa.

3. Mueva el cursor a la localización sobre el programa donde usted quiere colocar la nueva Transición y haga clic. La nueva Transición aparecerá en el programa.

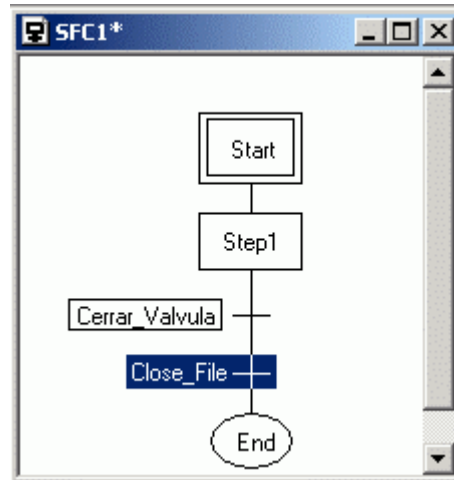


Figura 16. Presentación de la nueva transición booleana en el programa.

4. Para ingresar el código Boleano, haga doble clic sobre la Transición. Aparecerá el cuadro de dialogo **Edit Transition Logic**.

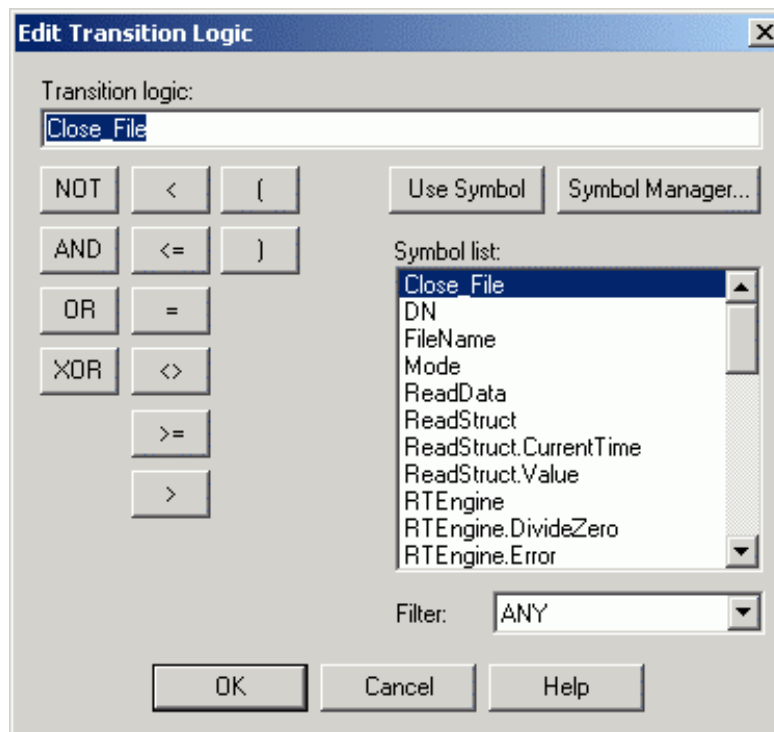


Figura 17. Definición de la nueva transición.

5. Para ingresar el código Boleano para la Transición, escríbalo directamente en esta o pulse el botón operador y seleccione símbolo que aparece en el campo **Symbol List**.
6. Pulse **OK** para guardar su trabajo y cerrar el cuadro de dialogo.

Adición de un Macro Paso(Macro Step)

➤ Para adicionar un macro Paso a un programa:

1. Pulse en **Macro Step Tool** sobre la barra de herramientas SFC.



Figura 18. Adición de un Macro Paso al programa.

2. Mueva el cursor al lugar en el programa donde usted quiere colocar el nuevo macro Paso y haga clic. Aparecerá el cuadro de dialogo **Edit Macro Step**.

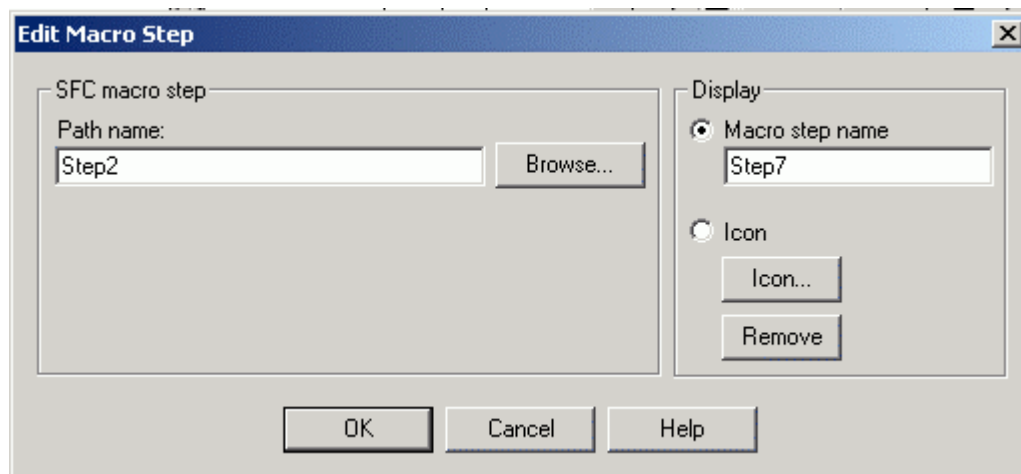


Figura 19. Definición del nuevo Macro Paso.

3. Ingrese la información para mostrar en el Macro Paso y para seleccionar en el macro SFC, descrito en "**MACRO PASOS**" en el capítulo "**ELEMENTOS DE PROGRAMAS SFC**". Luego pulse **OK**.

El nuevo Macro Paso aparecerá en el programa.

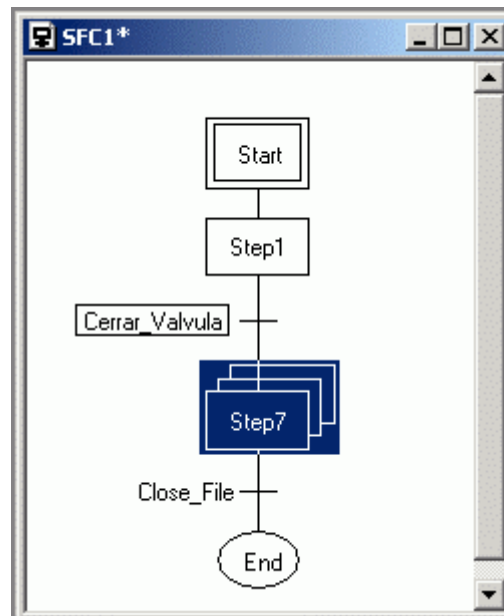


Figura 20. Presentación del nuevo Macro Paso en el programa.

4. Haga doble clic en el Macro Paso para editar el macro SFC. Si usted hace clic derecho en el Macro Paso y selecciona **Step Properties** usted puede cambiar el nombre del Macro Paso o como este se presenta en el SFC.

Adición de una Acción(Action)

Una Acción consiste de una o más secciones del código RLL que esta asociado con un Paso o un Macro Paso. El sistema ejecuta una Acción cuando su Paso asociado esta activo. Para información mas detallada sobre acciones, ver “ACCIONES” en él capitulo “ELEMENTOS DE PROGRAMAS SFC”. Usted debe crear un Paso o un Macro Paso antes de adicionar una Acción a un programa.

➤ Para adicionar una Acción a un programa:

1. Pulse en **Action Tool** sobre la barra de herramientas SFC.

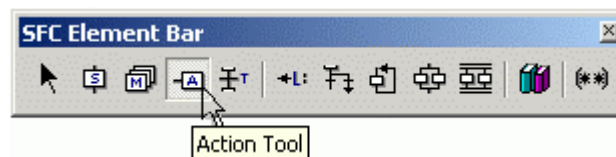


Figura 21. Adición de una Acción al programa.

2. Mueva el cursor al lugar en el programa donde usted quiere colocar la nueva Acción y haga clic. Aparecerá la nueva Acción en el programa.

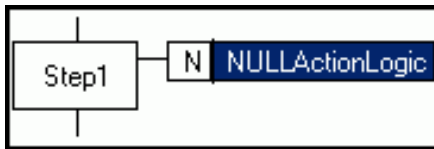


Figura 22. Presentación de la nueva Acción en el programa.

Usted puede adicionar mas de una Acción a un Paso o macro Paso ubicando el cursor encima de una Acción existente.

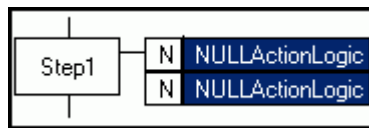


Figura 23. Adición de mas de una Acción a un programa.

➤ **Para editar una Acción:**

1. Haga doble clic en la Acción. Aparecerá el cuadro de dialogo **Edit Action Association**.

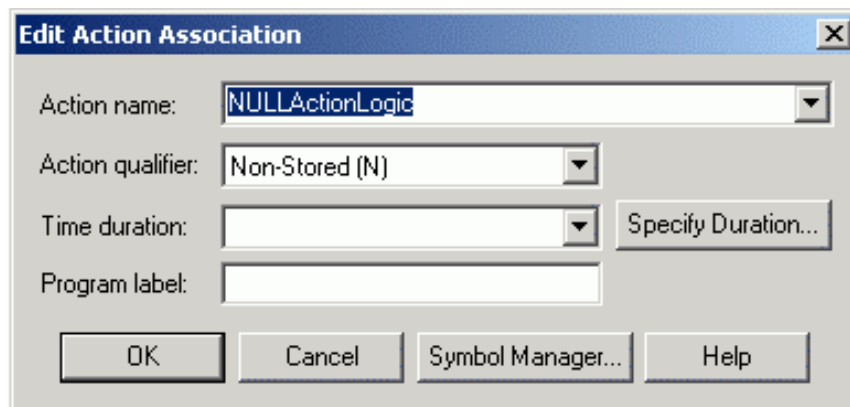


Figura 24. Edición de una acción.

2. Ingrese la información para configurar la Acción, descrita en **"ACCIONES"** en el capítulo **"ELEMENTOS DE PROGRAMAS SFC"**. Luego pulse **OK** para guardar sus cambios. El sistema cierra el cuadro de dialogo y luego muestra un escalón RLL vacío listo para editar.
3. Ingrese el código RLL, descrito en el capítulo **"ELEMENTOS DE PROGRAMAS RLL"**.

➤ **Para editar el código RLL de una Acción existente:**

1. Haga doble clic en el lado derecho de la Acción, como se muestra en la siguiente figura.

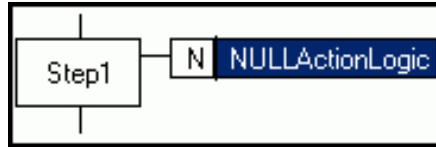


Figura 25. Edición Del código RLL de una acción.

Aparecerá el código RLL de la Acción.

2. Ingrese el código RLL, descrito en el capítulo “ELEMENTOS DE PROGRAMAS RLL”.

➤ **Para editar los parámetros de configuración de una Acción existente:**

1. Haga doble clic en el lado izquierdo de la Acción, como se muestra en la siguiente figura.

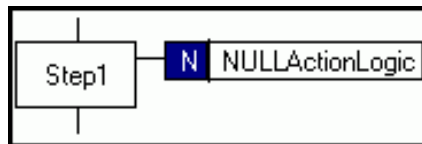


Figura 26. Edición de los parámetros de una acción.

El sistema mostrará el cuadro de diálogo **Edit Action Association** para la Acción.

2. Ingrese la información para configurar la Acción, como se describe en “ACCIONES” en el capítulo “ELEMENTOS DE PROGRAMAS SFC”. Luego pulse **OK** para guardar los cambios.

➤ **Para borrar permanentemente una Acción existente:**

1. Seleccione la Acción en el SFC y Pulse en **Cut Tool** para borrar la

Acción. 

2. Sobre el menú **Tools**, pulse en **Action Manager**. Aparecerá el cuadro de diálogo **Action Manager**.
3. Seleccione la Acción a eliminar y pulse el botón **Delete**. La Acción es borrada del **Action Manager**.

➤ **Para renombrar una Acción existente:**

1. Sobre el menú **Tools**, pulse en **Action Manager**. Aparecerá el cuadro de dialogo **Action Manager**.

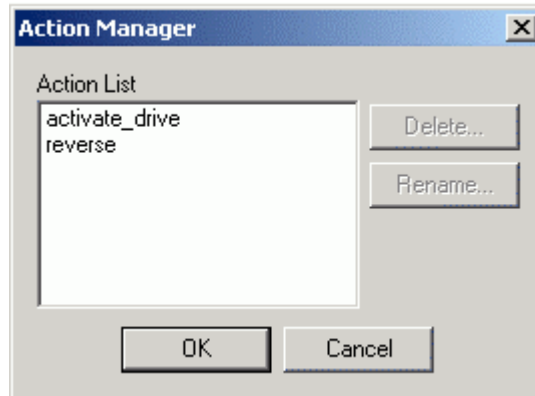


Figura 27. Cuadro Action Manager para renombrar una acción.

2. Seleccione la Acción ha renombrar y pulse el botón **Rename**.
3. Ingrese el nuevo nombre.

Nota: Usted debe renombrar las Acciones individualmente, después de hacer un cambio de nombre en el **Action Manager**.

Adición de un Salto(Jump)

Una combinación disponible de Salto-Etiqueta permite en el SFC la ejecución de una transferencia a cualquier lugar indicado por la etiqueta. Para información mas detallada sobre el uso de Saltos y Etiquetas, ver “**FLUJO DEL PROGRAMA: SALTOS y ETIQUETAS**” en él capitulo “**ELEMENTOS DE PROGRAMAS SFC**”.

➤ **Para adicionar un Salto a un programa:**

1. Seleccione o deseleccione la opción **Boolean Transition** sobre el menú **Edit** para escoger el tipo de Transición a usar con el Salto: RLL o Boléano.
2. Pulse en **Jump Tool** en la barra de herramientas SFC.



Figura 28. Adición de un salto al programa.

- Mueva el cursor al lugar en el programa donde usted quiere colocar el nuevo Salto y pulse. El nuevo Salto y dos transiciones aparecerán en el programa.

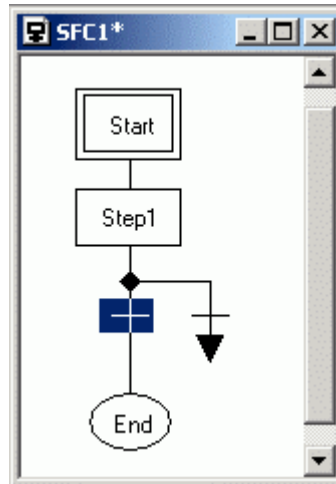


Figura 29. Presentación del nuevo salto en el programa.

- Haga doble clic en la punta de flecha del Salto.

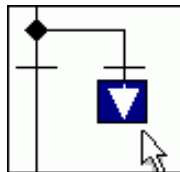


Figura 30. Selección de un Salto para su edición.

Aparecerá el cuadro de dialogo **Edit Jump Target**.

Figura 31. Edición de un Salto.

- Ingrese la Etiqueta a la cual el Salto transferirá el flujo del programa. Las Etiquetas deben iniciar con un carácter alfabético y seguidas por cualquier carácter alfanumérico y/o subrayado. En las Etiquetas no se distinguen las mayúsculas de las minúsculas y pueden tener una longitud de mas de 100 caracteres.
- Edite las dos Transiciones como se describe en “**Adición de una Transición**” (Pagina 9-)

Adición de una etiqueta(Label)

Una combinación disponible de Salto-Etiqueta permite en el SFC la ejecución de una transferencia a cualquier lugar indicado por la etiqueta. Para información mas detallada sobre el uso de Saltos y Etiquetas, ver “**SALTOS / ETIQUETAS: FLUJO DEL PROGRAMA**” en el capítulo “**ELEMENTOS DE PROGRAMAS SFC**”.

➤ **Para adicionar una etiqueta a un programa:**

1. Pulse en **Label Tool** sobre la barra de herramientas SFC.

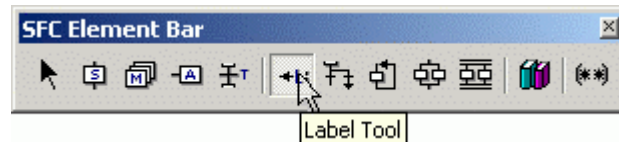


Figura 32. Adición de una etiqueta al programa.

2. Mueva el cursor al lugar en el programa donde usted quiere insertar la nueva etiqueta y pulse. La nueva Etiqueta aparecerá en el programa.

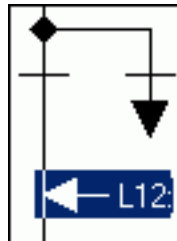


Figura 33. Presentación de la nueva etiqueta en el programa.

3. Haga doble clic en la Etiqueta. Aparecerá el cuadro de dialogo **Edit Label**.

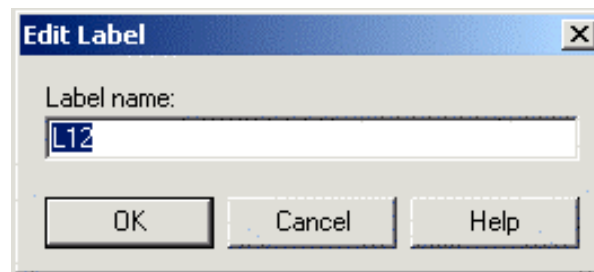


Figura 34. Edición de la nueva etiqueta.

4. Ingrese una etiqueta significativa. Las Etiquetas deben iniciar con un carácter alfabético y seguidas por cualquier carácter alfanumérico y/o subrayado. En las Etiquetas no se distinguen las mayúsculas de las minúsculas y pueden tener una longitud de mas de 100 caracteres.

Adición de un Lazo(Loop)

Un lazo permite que la ejecución del programa SFC regrese a un lugar anterior en el orden del programa para repetir una serie de Pasos. Para información mas detallada sobre el uso de Lazos, ver “**FLUJO DEL PROGRAMA: LAZOS**” en el capítulo “**ELEMENTOS DE PROGRAMAS SFC**”.

➤ **Para adicionar un lazo a un programa:**

1. Seleccione o deseleccione la opción **Boolean Transition** sobre el menú **Edit** para escoger el tipo de Transición a usar con el Lazo: RLL o Boléano.
2. Pulse en **Loop Tool** en la barra de herramientas SFC.

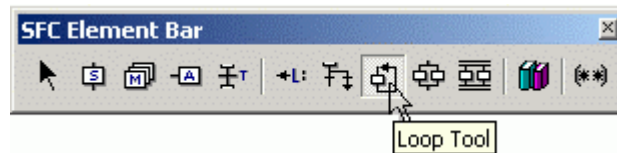


Figura 35. Adición de un Lazo al programa.

3. Mueva el cursor al lugar en el programa donde usted quiere colocar el extremo inferior del lazo y pulse. El nuevo lazo aparecerá en el programa.

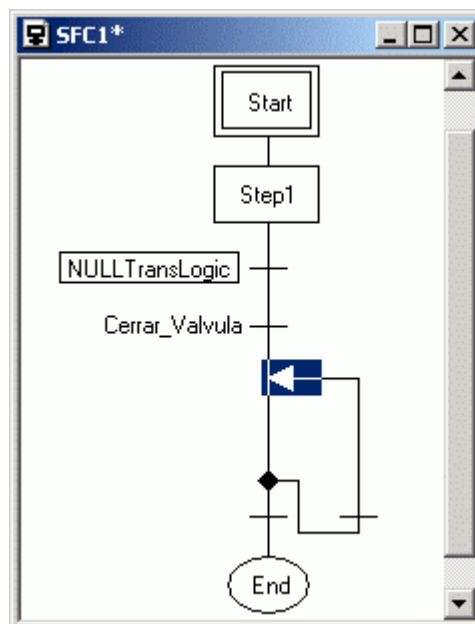


Figura 36. Presentación del nuevo lazo en el programa.

4. Arrastre la flecha del lazo al punto, donde el extremo superior del lazo será localizado.

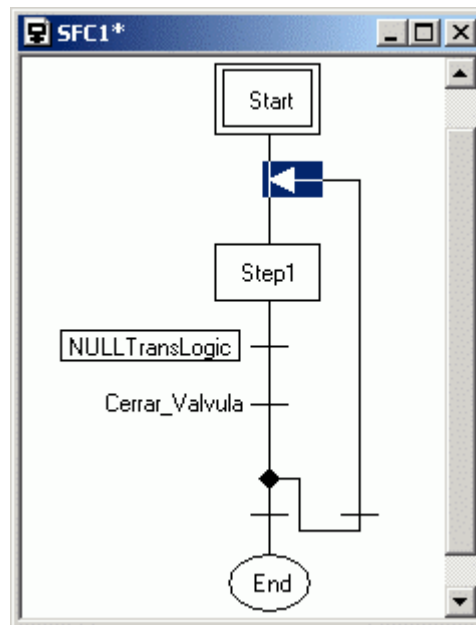


Figura 37. Ubicación del flujo del lazo en el programa.

5. Edite las dos Transiciones como se describe en “Adición de una Transición” (Pagina 9-)

Adición de una Divergencia de Selección

Una Divergencia de Selección permite que la ejecución del programa SFC siga una de dos o más rutas de control. Para información mas detallada sobre el uso de Divergencias de Selección, ver “DIVERGENCIA DE SELECCIÓN: FLUJO DEL PROGRAMA” en el capítulo “ELEMENTOS DE PROGRAMAS SFC”.

➤ Para adicionar una Divergencia de Selección a un programa:

1. Seleccione o deseleccione la opción **Boolean Transition** sobre el menú **Edit** para escoger el tipo de Transición a usar con la Divergencia: RLL o Boléana.
2. Pulse en **Select Diverge Tool** en la barra de herramientas SFC.

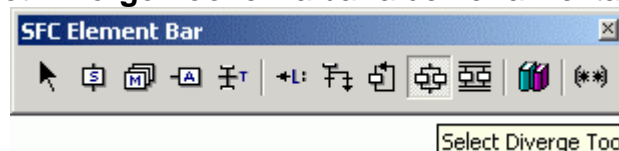


Figura 38. Adición de una divergencia de selección en el programa.

3. Mueva el cursor al lugar en el programa donde usted quiere colocar la Divergencia de Selección y pulse. La Divergencia de Selección aparecerá en el programa.

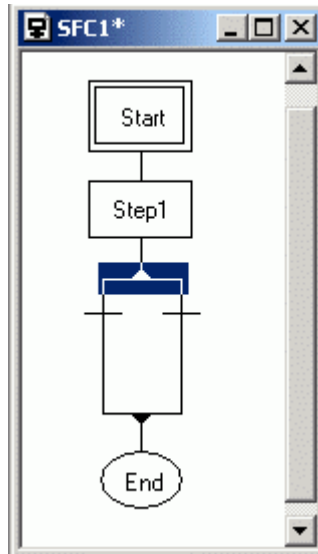


Figura 39. Presentación de la nueva divergencia de selección.

4. Edite las dos Transiciones como se describe en “Adición de una Transición” (Pagina 9-)

➤ **Para adicionar otra ruta a la Divergencia de Selección:**

1. Pulse arriba de la Divergencia de Selección.

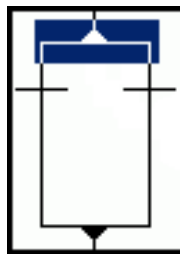


Figura 40. Selección de la divergencia.

2. Pulse en **Select Diverge Tool** y coloque el cursor en la parte superior de la divergencia.

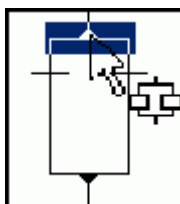


Figura 41. Localización de la nueva divergencia de selección.

3. Pulse y aparecerá otra ruta en la divergencia.

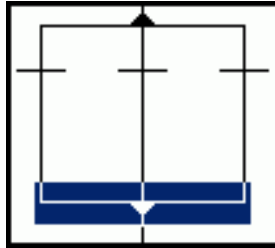



Figura 42. Presentación de la nueva ruta en la divergencia.

- **Para borrar una ruta en la divergencia:**

- Seleccione la ruta y Pulse en **Cut Tool** para borrarla. 

- **Para borrar la entrada de la divergencia:**

1. Pulse bien sea en la parte superior o inferior de la divergencia.
2. Pulse en **Cut Tool**.

Adición de una Divergencia Paralela

Una Divergencia Paralela permite que la ejecución del programa SFC siga dos o más rutas de control. La ejecución a lo largo de cada ruta debe ser terminada para que la ejecución del programa continúe mas allá de la Divergencia Paralela. Para información mas detallada sobre el uso de Divergencias Paralelas, ver “**FLUJO DEL PROGRAMA: DIVERGENCIA PARALELA**” en el capítulo “**ELEMENTOS DE PROGRAMAS SFC**”.

- **Para adicionar una Divergencia Paralela a un programa:**

1. Pulse en **Parallel Diverge Tool** en la barra de herramientas SFC.

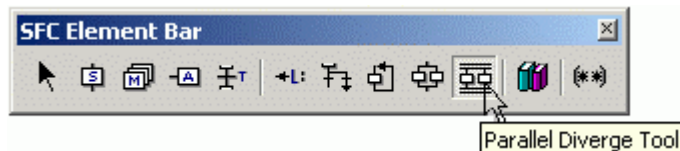


Figura 43. Adición de una divergencia Paralela en el programa.

2. Mueva el cursor al lugar en el programa donde usted quiere colocar la Divergencia Paralela y pulse. La Divergencia Paralela aparecerá en el programa.

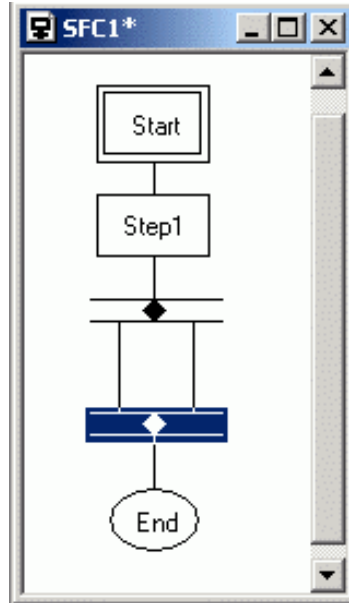


Figura 44. Presentación de la nueva divergencia paralela.

➤ Para adicionar otra ruta a la Divergencia Paralela:

1. Pulse arriba de la Divergencia Paralela.

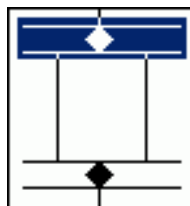


Figura 45. Selección de la divergencia paralela.

2. Pulse en **Parallel Diverge Tool** y coloque el cursor en la parte superior de la divergencia.

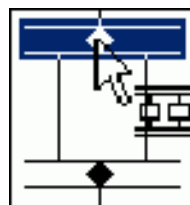


Figura 46. Localización de la nueva divergencia paralela.

3. Pulse y aparecerá otra ruta en la divergencia paralela.

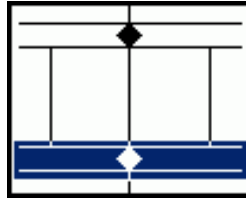


Figura 47. Presentación de la nueva ruta en la divergencia paralela.

➤ **Para borrar una ruta en la divergencia:**

- Seleccione la ruta y Pulse en **Cut Tool** para borrarla.



➤ **Para borrar la entrada de la divergencia:**

1. Pulse bien sea en la parte superior o inferior de la divergencia.
2. Pulse en **Cut Tool**.

Adición de un Paso de Librería

Un Paso de Librería es un Paso de programa que contiene la plantilla de un código, el cual es definido por el usuario para una función específica, y un icono, el cual es apropiado para la función. El Paso opera así mismo como otro Paso, basado en un código de programa que usted usa en este. El InControl provee una librería de varios iconos que puede ser usada dentro de un SFC. Para información mas detallada acerca del uso de los Pasos, ver “**PASOS**” en el capítulo “**ELEMENTOS DE PROGRAMAS SFC**”. Usted debe crear un Paso para la librería antes de adicionarlo al programa.

➤ **Para crear un nuevo Paso de librería:**

1. Sobre el menú **Tools**, pulse en **Step Library**, y luego pulse en **New Library Step**. Aparecerá la paleta de iconos.

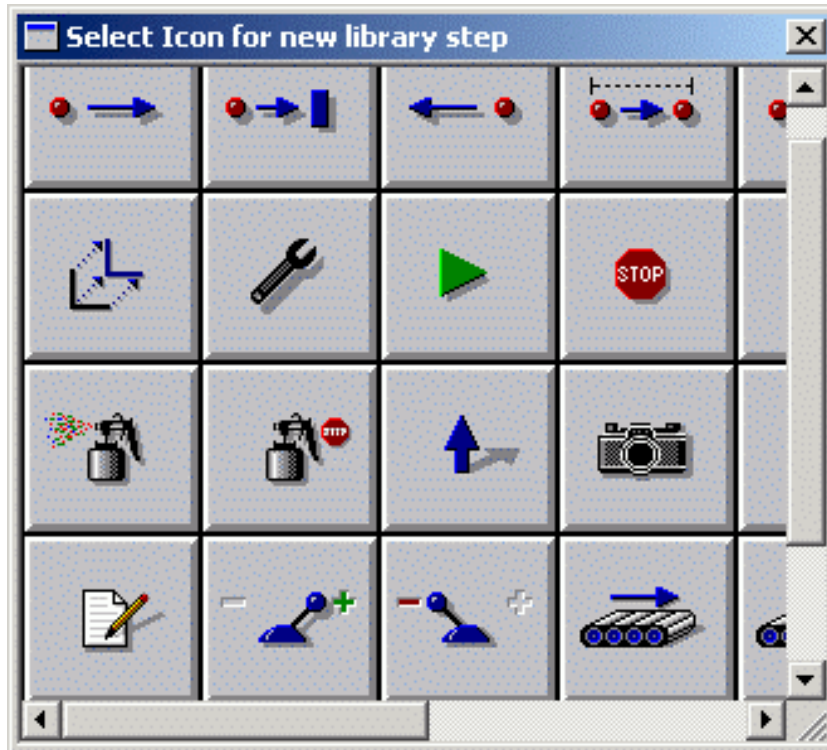


Figura 48. Creación de un Paso de librería.

2. Pulse en el icono que representa el Paso. Aparecerá el cuadro de dialogo **Library Step Title**.

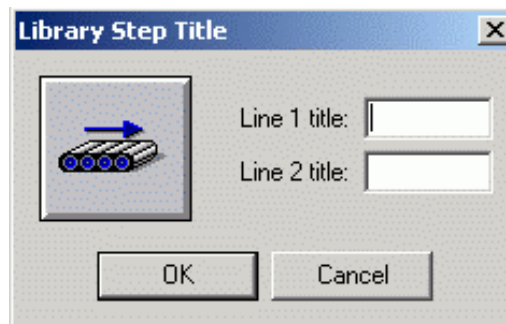


Figura 49. Definición del Paso.

- Ingrese un título para el Paso y pulse **OK**. Aparecerá el cuadro de dialogo **Edit Step**.

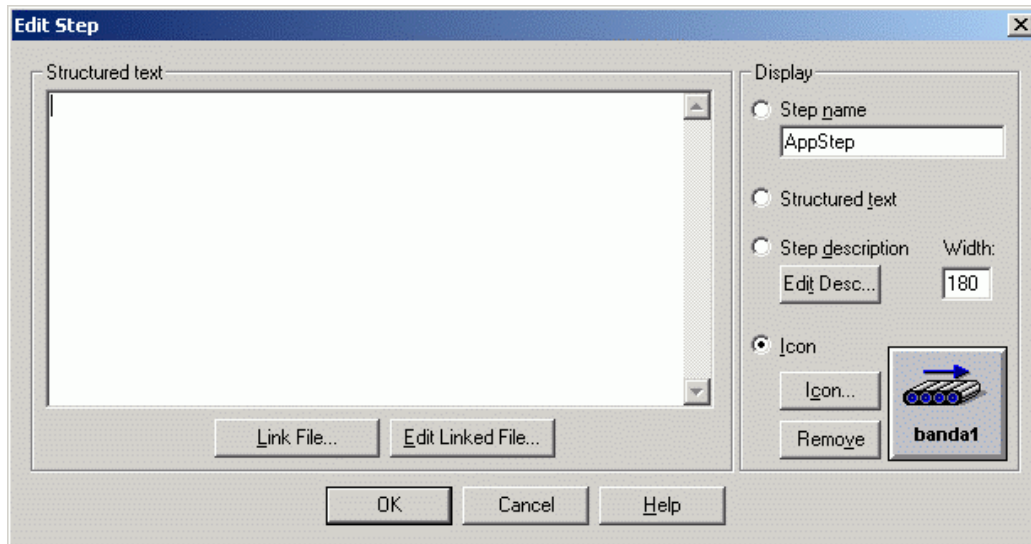


Figura 50. Cuadro de edición del nuevo Paso de librería.

- Personalización del Paso:

Ingrese el Código de Texto Estructurado.

Ingrese el nombre del Paso.

Escoja como usted quiere mostrar el Paso: Icono, Nombre del Paso, Código, etc.

- Pulse **OK**. El nuevo Paso es adicionado a la librería, y aparece la paleta de iconos. Bien sea cree un nuevo Paso o cierre la paleta.

➤ **Para editar un Paso en la librería:**

- Sobre el menú **Tools**, pulse en **Step Library**, y luego pulse en **Edit Library Step**. Aparecerá la ventana **Edit Library Step**.

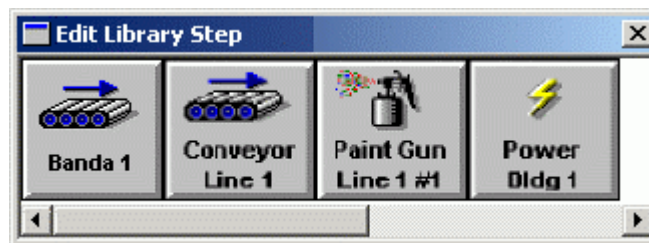


Figura 51. Edición de un Paso de librería.

- Pulse en el Paso para editarlo.

3. Después que aparece el cuadro de dialogo **Edit Step**, haga sus cambios y luego pulse **OK**. Los cambios son adicionados al Paso.
 4. Aparece la paleta de iconos **Edit Library Step**. Bien sea edite otro Paso o cierre la paleta.
- **Para borrar un Paso desde la librería:**
1. Sobre el menú **Tools**, pulse en **Step Library**, y luego pulse en **Delete Library Step**. Aparecerá la ventana **Delete Library Step**.
 2. Pulse en el Paso para borrarlo. Tenga en cuenta que usted no ha sido notificado para confirmar.
 3. El Paso seleccionado es removido de la librería.
 4. Aparece la ventana **Delete Library Step**. Bien sea borre otro Paso o cierre la ventana.

Adición de un Paso desde la Librería de Pasos

Usted debe crear un Paso para la librería antes de adicionarlo al programa.

- **Para adicionar un Paso desde la librería al programa:**
1. Pulse en **Library Step** sobre la barra de herramientas SFC.

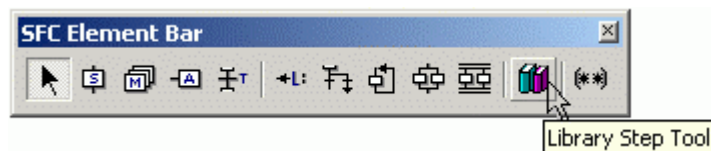


Figura 52. Adición de un Paso de librería.

Aparece la ventana **Select Library Step Tool**.

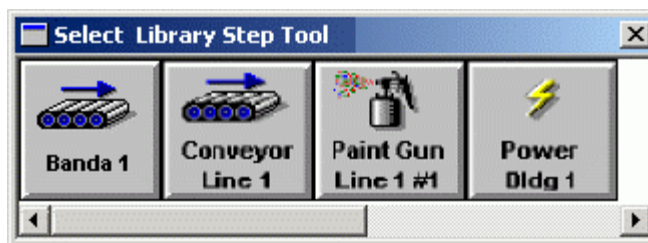


Figura 53. Selección de un Paso desde la librería.

2. Pulse en el Paso apropiado. La ventana **Select Library Step Tool** se cierra.

3. Mueva el cursor al lugar en el programa donde usted quiere colocar el Paso y pulse. Aparecerá el cuadro de dialogo **Edit Step**.

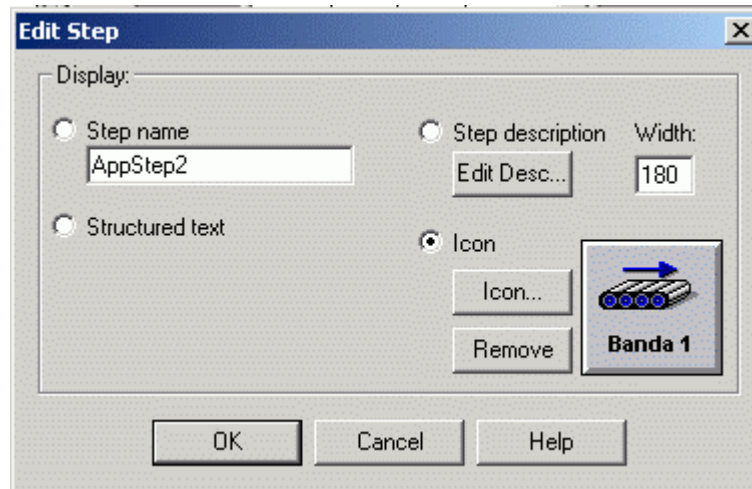


Figura 54. Cuadro de edición de un Paso ya creado.

4. Ingrese la información apropiada para mostrar el Paso en el SFC y Luego pulse **OK**.

Adición de comentarios al programa

Un comentario puede consistir de cualquier descripción significativa que usted quiera mostrar al lado del elemento de programa. Usted puede escoger ya sea que el sistema muestre los comentarios o los oculte.

➤ Para ingresar un comentario en el programa:

1. Pulse en **Comment Tool** sobre la barra de herramientas SFC.

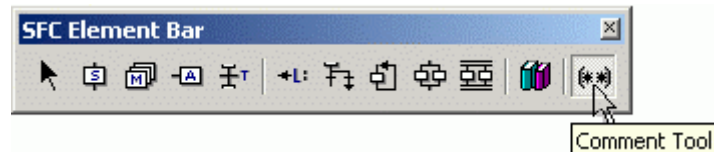


Figura 55. Adición de un comentario al programa.

2. Mueva el cursor al lugar en el programa donde usted quiere colocar el comentario y pulse. El nuevo comentario aparecerá en el programa.

- Haga doble clic sobre el comentario. Aparecerá el cuadro de dialogo **Program Comments**.

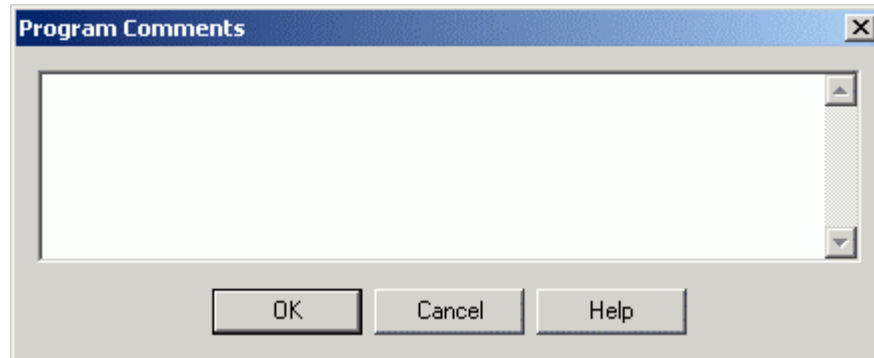


Figura 56. Ingreso del texto del comentario.

- Ingrese el comentario y pulse **OK**. Su comentario aparecerá dentro del programa.

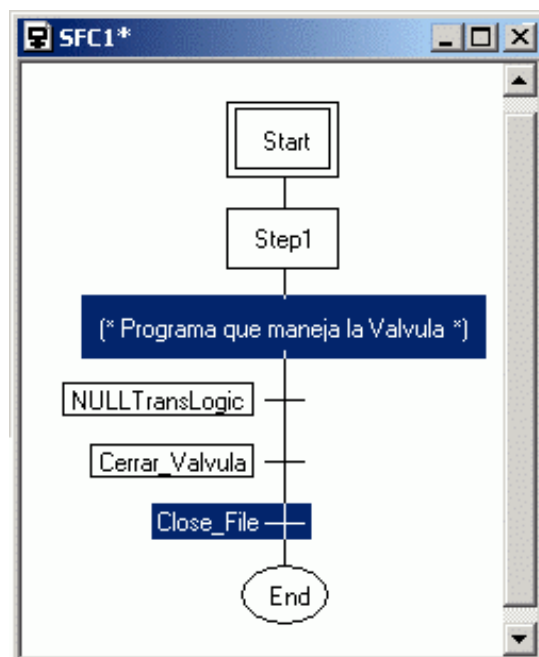


Figura 57. Presentación del comentario en el Programa.

EDICIÓN DE ELEMENTOS DE PROGRAMA

➤ Para editar un elemento de programa existente:

1. Pulse en **Select Tool**.

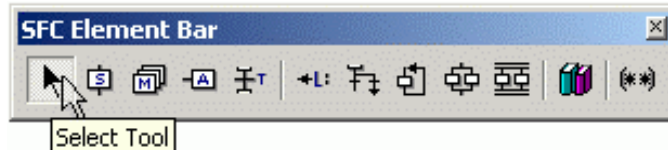


Figura 58. Edición de un elemento de programa existente.

2. Haga doble clic sobre el elemento (Paso, Transición, Etiqueta, etc.). Aparecerá el cuadro de dialogo apropiado para el elemento (**Edit Step**, **Select RLL Transition Logic**, **Bypass Jump Transitio Logic**, etc.). Si un elemento ya esta seleccionado, usted también puede abrir el cuadro de dialogo de edición, presionando la tecla **Enter**.
3. Haga cambios en el cuadro de dialogo requerido.

ELEMENTOS DE PROGRAMAS SFC

Este capitulo presenta los elementos de programación que usted usara en las Cartas de Funciones Secuenciales.

CONTENIDO

- QUE ES UN SFC?.
- PASOS (STEP).

This product is licensed to:

Company:	N/A
SN:	N/A
Expires:	N/A
Locked To:	N/A

7.1.0.1337



QUE ES UN SFC?

Elementos de un SFC

Las Cartas de Funciones Secuenciales (SFC) es un programa de aplicación que se representa como una serie de estados secuenciales o Pasos. Un Paso representa una condición en la cual la ejecución del programa sigue un conjunto de reglas definidas por Acciones y Funciones asociadas con el Paso.

La flexibilidad en la sintaxis del lenguaje SFC le permite a usted llamar a un SFC de entrada (SFC hijo o destino) para su ejecución desde dentro de un Paso sencillo, llamado Macro Paso. Cuando el SFC hijo se ha completado, el control del programa regresa al Macro Paso que hizo la llamada. Usted puede diseñar múltiples ramas en su SFC. El ramal de la Divergencia de Selección puede consistir de dos o más rutas, en este caso a la ejecución del programa solo le es permitida seguir una de estas rutas. El ramal de la Divergencia Paralela puede consistir de múltiples rutas, pero la ejecución del programa debe proceder por todas las rutas.

Flujo del programa

El flujo del programa se mueves desde la parte superior hasta la parte inferior, como se muestra en la siguiente figura. El código dentro de cada Paso es ejecutado, y cuando este se ha completado, el flujo del Programa se mueve hacia el próximo elemento de programa. Si el próximo elemento es un Paso, el código dentro del paso es ejecutado. Si el proximo elemento es una Transición, el flujo del programa continua hasta cuando la Transición sea TRUE. Una Transición es una condición de lenguaje Booleano o RLL que se resuelve para un estado de TRUE o FALSE.

Figura

Cuando el flujo del programa alcanza el Paso de Finalización(End), el modo del SFC cambia de ejecución a terminación. El programa debe ser reiniciado antes

que este pueda ejecutarse de nuevo. En el siguiente ejemplo, una Transición sigue a otra. El flujo del programa aun se mueve desde la parte superior hasta la parte inferior, y la ejecución de los elementos de programa no comienza hasta que los elementos anteriores se hayan completado. Después que la primera Transición, T1, se hace TRUE, la segunda Transición T2 se activa. Después de la segunda Transición, T2 se hace TRUE, el **Step2** se activa.

Figura

Usted puede usar una combinación de Salto a Etiqueta para saltarse al código que su aplicación requiera. Usted también puede agregar Lazos para reejecutar una sección de código. El InControl le permite ejecutar múltiples programas de múltiples tipos. Esto es, usted puede correr dos programas RLL, por ejemplo, al mismo tiempo que tres SFC son ejecutados. Usted puede coordinar la ejecución de programas a través de símbolos globales, los cuales son admitidos por todos los tipos de programas. La variable de sistema Mode, la cual está asociada con un programa individual, es un símbolo local que usted también puede usar para coordinar la ejecución de programas.

PASOS (STEP)

Funciones

El SFC es un programa de aplicación que se representa como una serie de Pasos secuenciales. Un Paso representa una condición en la cual el comportamiento de los procesos de fábrica es definido por el código de Texto Estructurado y las Acciones asociadas con dicho Paso. Mientras que el SFC está siendo ejecutado, un Paso está activo o inactivo, y en un momento determinado, el estado de los procesos de la fábrica estarán definidos por el Paso activo y los valores de sus variables internas o de salida. En la figura siguiente, por ejemplo, el **Step3** es el Paso activo, y solo el código dentro del Paso está siendo ejecutado. Usted puede crear SFC que tengan múltiples rutas, sin embargo, y esto es posible para que más de un SFC esté activo a la vez.

Figura

Un Paso esta representado dentro de un SFC como un cuadro que contiene el identificador del Paso. Cuando usted crea un nuevo SFC, el sistema automáticamente genera el primer Paso, llamado **Start**(inicio), y el ultimo Paso, llamado **End**(final). Usted no puede editar estos Pasos; ellos simplemente representan el comienzo y el final del SFC.

Típicamente, usted separa los Pasos en un SFC con Transiciones, las cuales son elementos de programas descritos en “**TRANSICIONES**” (pagina 10-). Como una mejora de la especificación IEC-61131, el InControl permite localizar un Paso inmediatamente antes o después de otro Paso, lo cual implica que debe existir una Transición (TRUE) entre ellos.

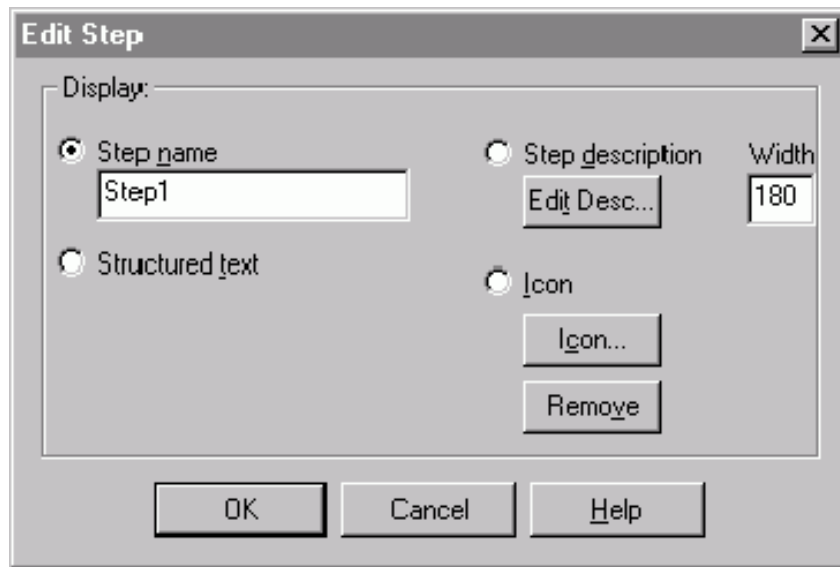
El código de Texto Estructurado en un Paso es ejecutado una sola vez, cuando el Paso este activo. El código no es ejecutado otra vez, hasta que el Paso que esta inactivo, se active. Esto ocurre en una exploración o análisis (Scan); sin embargo la presencia de lazos, operación de archivos, u ordenes **SCAN** pueden causar que la ejecución tome mas de una exploración o análisis. Si usted quiere ejecutar un código continuamente mientras un Paso este activo, usted debe colocar en este una Acción. Una Acción es otro elemento de programa que usted puede usar para coordinar la ejecución de un código dentro de un Paso con otro código de programa. Para mas información, ver “**ACCIONES**” (pagina 10-).

Parámetros

Los parámetros del Paso definen como el Paso es mostrado en el programa.

➤ Para editar los parámetros de un Paso:

- Sobre el menú **Edit**, pulse en **Step Properties**. Aparecerá el cuadro de dialogo **Edit Step**. Usted también puede hacer clic derecho sobre el Paso.



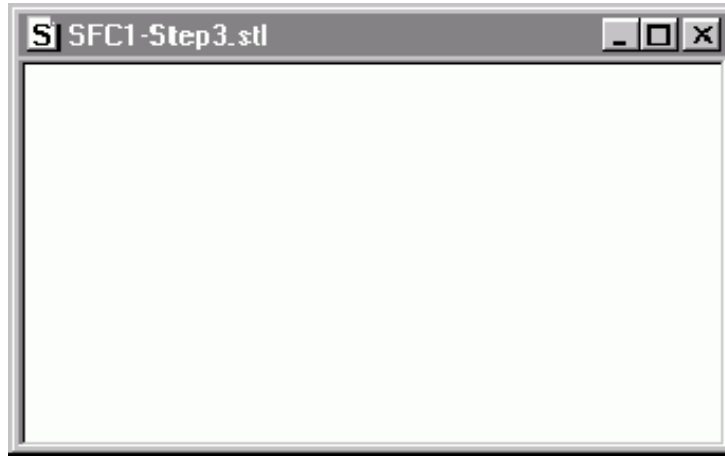
Cuadro de diálogo de edición de Pasos

Campo / Botón	Descripción
Step Name (Nombre del Paso)	Ingrese un nombre para el Paso. La longitud máxima del nombre del Paso es de 100 caracteres. Pulse en el botón de opción Step Name para mostrar el nombre del Paso en el SFC.
Structured Text (Texto Estructurado)	Pulse en el botón de opción Structured Text para mostrar el código del Paso en el SFC. Para una serie grande de comandos, se puede aumentar el tamaño del Paso.
Step Description (Descripción del Paso)	Pulse en el botón de opción Step Description para mostrar la descripción del Paso en el SFC.
Edit Description (Cuadro de edición)	Pulse en Edit Description para ingresar una descripción del Paso.
Width (Ancho)	Ingrese el grosor en píxeles para la descripción del Paso.
Icon (Botón de opción)	Pulse en el botón de opción Icon para mostrar el Paso en el SFC como un icono.
Icon (Botón)	Pulse en el botón Icon para mostrar la paleta de iconos de la cual se va a escoger.
Remove (Borrar)	Pulse en Remove para borrar un icono desde el Paso, si uno está asignado.

Código

➤ Para ingresar el código en un Paso:

1. Haga doble clic en el Paso. Se abre la ventana de edición de Texto Estructurado.




2. Ingrese el código de programa usando el lenguaje de Texto Estructurado.

Uso de Pasos de librería

Un Paso de librería es un Paso en el programa que contiene un modelo de código, el cual es definido por el usuario para una función específica, y un icono, el cual es apropiado para la función. El Paso opera así mismo como otro Paso, basado en un código de programa que usted usa en este. El InControl provee una librería de varios iconos que puede ser usada dentro de un SFC.

LENGUAJE DE TEXTO ESTRUCTURADO

Este capítulo describe cómo crear un programa de aplicación usando el lenguaje de Programación de Texto Estructurado (Structured Text Lenguaje, STL).



CONTENIDO

- ELEMENTOS DE TEXTO ESTRUCTURADO
- EXPRESIONES
- TIPOS DE ORDENES
- FUNCIONES Y PROCEDIMIENTOS DE TEXTO ESTRUCTURADO

This product is licensed to:

Company:	N/A
SN:	N/A
Expires:	N/A
Locked To:	N/A

7.1.0.1337

ELEMENTOS DE TEXTO ESTRUCTURADO

El lenguaje de programación de Texto Estructurado (**STL**) es un subconjunto del conjunto de instrucciones basadas en texto que siguen la norma IEC-61131. Estas instrucciones están diseñadas para la fácil creación de operaciones tanto lógicas como matemáticas.

- Use el Texto Estructurado para crear el código de aplicación para un paso **SFC**, como se ilustra a continuación. Cuando el **SFC** es ejecutado, el código de Texto estructurado que incorporó dentro de cada paso es procesado cuando el paso es activado.

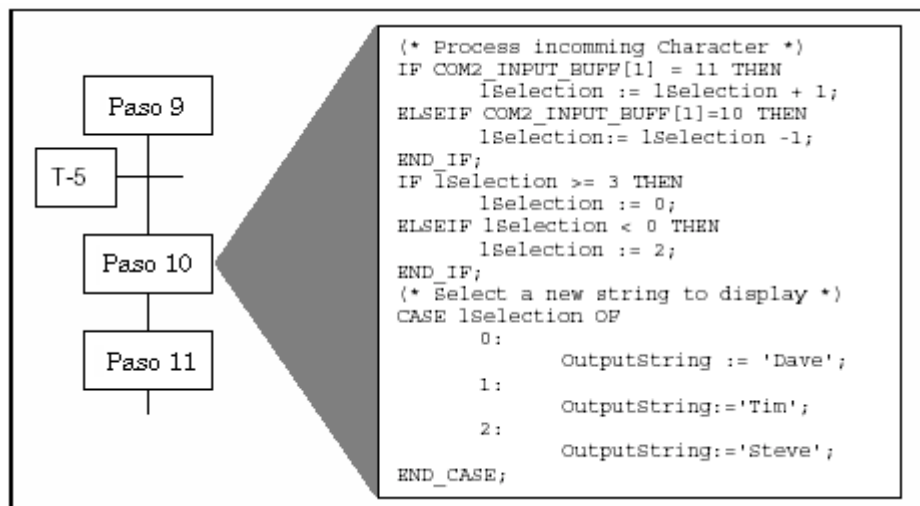


Figura 1. Código STL del paso SFC.

- Puede crear un programa **STL** independiente, como se muestra a continuación.

```

SFC1-Step1.stl
vari_a := 50;
vari_b := 10;
vari_c := 0;
WHILE vari_b <> vari_a DO
    vari_b := vari_b + vari_a/10;
END_WHILE;

```

Figura 2. Programa STL independiente de un Paso SFC.

- Actualmente, las funciones definidas por el usuario (user-defined), procedimientos o bloques de funciones deben estar escritos en lenguaje STL.

Cuando diseñe el programa, tenga en cuenta los siguientes puntos:

- El cuerpo del código de STL consiste de **Expresiones**(pagina 11-), **Tipos de Ordenes**(pagina 11-), y **LLamadas a Procedimientos y Funciones**(pagina 11-), descritos en este capítulo. Los operadores y tipos de datos válidos en STL están descritos en el tema “**Expresiones**”(pagina 11-4).
- Haga cualquier declaración que necesite en el programa desde el Symbol Manager, como está descrito en “**SYMBOL MANAGER**” en el capítulo “**DEFINICIÓN DE VARIABLES**”.
- Las palabras reservadas en InControl, las cuales están mostradas en el apéndice “**PALABRAS RESERVADAS**”, tienen especial significado dentro del lenguaje de Texto Estructurado. No use las palabras reservadas como nombres de variables.

EXPRESIONES

Una expresión está definida como una combinación de operadores (matemáticos, lógicos o de relación) y operando (constantes, variables, valores literales o expresiones) que pueden ser evaluados, aceptando un valor de un tipo de dato específico, número entero, número real, etc. Los operadores y los tipos de datos están descritos en esta sección.

Operadores

En la tabla de abajo están listados los operadores que usted puede usar dentro de una expresión. El orden de prioridad determina la secuencia en la cual serán ejecutados dentro de la expresión. El operador con la mas alta prioridad es aplicado primero, seguido por el operador con la siguiente mas alta prioridad. Los operadores de igual prioridad son evaluados de izquierda a derecha.

Operador	Símbolo	Prioridad
Paréntesis	()	1
Función de Evaluación	identificadores (lista de argumento) ejemplo: LN (A), ABS (X)	2
Exponencial	**	3
Negativo	-	4
Complemento	NOT	4
Multiplicación	*	5
División	/	5
Módulo	MOD	5
Adición	+	6
Sustracción	-	6
Comparación ¹	<,>, <=, >=	7
Igualdad	=	8
Desigualdad	<>	8
Boolean / Operador AND	AND	9
Boolean / Operador OR exclusivo	XOR	10
Boolean / Operador OR	OR	11

1 En general, es recomendable que evite hacer una comparación por igualdad (o desigualdad) con números reales. Si usted hace este tipo de comparación usando una constante (literal) y una variable real, la variable debe ser un tipo de dato LREAL para garantizar que puede recibir el resultado esperado.

Estos símbolos tienen las siguientes funciones:

- **:=** Asigna el valor de una expresión a una variable.
- **;** El punto y coma es requerido para designar el final de una declaración.
- **[]** Los corchetes son usados para filas o series indexadas cuando la fila indexada es un integrador. Por ejemplo, la siguiente orden pone al primer elemento de una serie el valor $j + 3$: **array [1]:= j + 3;**

- :: Dos puntos dobles designan un nivel. Los niveles deben estar seguidos por una declaración en la misma o en la siguiente línea. Por ejemplo, esto especifica un nivel:

spray_on::

- (* *) Designa un comentario. Ejemplo : (*Este es un comentario*)

Tipos de Datos

Cuando una expresión es evaluada, el resultado debe ser uno de los tipos de datos soportados por el InControl. Estos tipos de datos están mostrados en la siguiente tabla. Y están descritos detalladamente en el capítulo “DEFINICIÓN DE VARIABLES”.

Todos los tipos	Grupo	Subgrupo	Tipo de Dato
ANY	ANY_NUM	ANY_REAL	LREAL
			REAL
		ANY_INT	DINT
			INT
			SINT
			DWORD
			WORD
			BYTE
	ANY_BIT	DWORD	
		WORD	
		BYTE	
		BOOL	
	ANY_DATE	DT (fecha y tiempo)	
		DATE	
		TOD	
			STRING
		TIME	
		FILE ¹	
		TMR ¹	
		User-Defined	
1	Mejora de la especificación IEC 61131-3.		

TIPOS DE ORDENES

Las Ordenes del Lenguaje de Texto Estructurado, las cuales permiten la ejecución del programa actual, consisten de los siguientes tipos:

- **Asignación** Le pone a un objeto un valor específico (Pág. 11-8)
- **BREAK** Detiene la ejecución del programa si ha habilitado la opción “**Debugging**” (depuración) (Pág. 11-9).
- **CASE** Para la ejecución condicional de una parte de la declaración (Pág. 11- 10).
- **Comentario** Provee los comentarios que serán incluidos dentro del programa (Pág. 11-11).
- **EXIT** Termina las interacciones antes de que la condición terminal llegue a ser verdadera TRUE (Pág. 11-12).
- **FOR** Indica que una secuencia de declaración será ejecutada repetidamente basada en el valor de una variable de control (Pág. 11-12).
- **Llamadas de Funciones y Procedimientos** Llama a una función o procedimiento para que sea ejecutado (Pág. 11-14).
- **IF** Especifica que una o mas declaraciones serán ejecutadas condicionalmente (Pág. 11-14).
- **INCLUDE** Ejecuta una parte de las declaraciones contenidas dentro de un archivo externo (Pág. 11-15).
- **LABEL** Usado dentro de una Acción SFC para controlar cuando el código RLL asociado con un paso o un Macro paso es ejecutado (Pág. 11-16)
- **REPEAT** Indica que una secuencia de la declaración será ejecutada repetidamente hasta que una expresión Boolean sea evaluada como verdadera (**TRUE**). Ver (Pág. 11-16).
- **RETURN** Usada dentro de una función, procedimiento o bloque de función para hacer que el programa corra de nuevo en la POU que hace el llamado de función (Pág. 11-17).
- **SCAN** Causa que la ejecución del Texto Estructurado sea suspendida momentáneamente mientras una entrada o salida es examinada (Pág. 11-17).
- **WHILE** Indica QUE una secuencia declarada será ejecutada repetidas veces hasta que una expresión Boolean sea evaluada como falsa (Pág. 11-18).
- **#pragma** Modifica las características y operaciones de revisión del compilador (Pág. 11-19).

En la descripción de la sintaxis de las declaraciones que sigue, los signos <> indican los elementos que sustituyen un valor. Los corchetes [] indican que su contenido es opcional dentro de la declaración. Los elementos que usted debe teclear aparecen exactamente en: `typewriter font`.

Asignación

Use la declaración de asignación para reemplazar el valor de un objeto con el resultado de la expresión evaluada. El formato para la declaración de asignación es el siguiente:

<object> := <expression>;

Donde **<object>** es una variable, fila de elementos, etc. y **<expression>** es un valor único o expresión.

Mover o asignar estructuras y filas no es recomendable. Este tipo de operación se extiende a una serie de cambios y asignaciones que pueden causar exceso de revisiones o análisis. Para trasladar una estructura (**Structure1:=Structure2**), el tamaño y el tipo de dato de los miembros de la estructura deben ser exactamente iguales. La conversión de tipos de datos no es soportada para traslados complejos.

Para trasladar una fila o arreglo (**Array1:=Array2**), el tamaño y tipo de dato de los elementos de las filas deben ser exactamente iguales. Cada elemento de la fila 2 (Array2) es trasladado a la fila 1 (Array1).

Los siguientes ejemplos son declaraciones de asignación Boleanas:

Boolitem1:= TRUE; Boolitem2 := (val <= 75);

En el siguiente ejemplo pone en un elemento de una fila el valor resultante de la evaluación de una expresión con número real:

RealArray[13] := (r1a / r1b)* 13.41574;

El siguiente ejemplo pone en una variable string (cadena de caracteres) al valor de un string. Asegúrese de encerrar el string entre comillas simples (' ').

String_Val := 'Es un string literal';

El siguiente ejemplo asigna en un valor la variable del proceso para el objeto de fábrica (FOE) InControl **PID** llamado BoilerTempControl.

```
BoilerTempControl.PV := 500.0;
```

BREAK

La declaración **BREAK** ocasiona que el flujo del programa se detenga y es útil para depurar un programa. Para habilitar las declaraciones **BREAK** usadas en un programa, pulse **Enable Debug** sobre el cuadro de diálogo **Validate Project** o **Validate Program** o sobre el cuadro de diálogo **Properties** del programa. Para programas de texto estructurado considere usar el punto de ruptura (breakpoint), ver ("**Uso de Breakpoints**" en el capítulo "**EJECUCIÓN DE UN PROYECTO**"). El formato para la declaración **BREAK** es el siguiente:

```
BREAK;
```

El siguiente es un ejemplo de una declaración **BREAK**.

```
Intout := BCD_TO_INT(bcd_in);  
BREAK;  
Max_num := MAX (num1, num2);
```

Nota: Las siguientes declaraciones no pueden aparecer en la misma línea: **BREAK**, **SCAN**, **END_FOR**, **END_FOR_NOWAIT**, **END_WHILE**, y **END_WHILE_NOWAIT**.

La declaración **BREAK** no esta permitida en funciones, procedimientos, o bloques de funciones.

CASE

Use la declaración **CASE** para diseñar la ejecución de una conjunto de declaración basado en el valor de una variable. La construcción de la declaración **CASE** consiste en lo siguiente:

- Una expresión que evalúa un valor de tipo **ANY_INT**.
- Una lista de grupos de declaraciones, con cada grupo clasificado por uno o más enteros o rangos de enteros.

Cuando el nivel para un grupo de declaraciones se iguala al valor de la expresión ANY_INT, las declaraciones en estos grupos son ejecutados. Si el nivel consiste de mas de un integrador o rango de integración, la igualdad puede ocurrir con cualquiera de los integradores contenidos en el nivel. Si no ocurre la acción de igualdad o emparejamiento y ha incluido una declaración ELSE, la declaración siguiente al ELSE es ejecutada. Si no hay declaración ELSE, ninguna declaración es ejecutada en la declaración CASE.

El formato para la declaración CASE es el siguiente:

```

CASE      <expression>   OF
          <INT>:
              <statement list>
          <INT>,<INT>,<INT>:
              <statement list>
          <INT>..<INT>:
              <statement list>
ELSE
          <statement list>
END_CASE;
```

Donde **<expression>** evalúa el tipo de dato ANY_INT, **<INT>** es un valor literal ANY_INT, y **<Statement List>** es cualquier juego de declaraciones de Texto estructurado válido.

El siguiente ejemplo selecciona un nuevo string (cadena de caracteres) para visualizar:

```

CASE ASelection OF
  0:
      OutputString := 'Dave';
  1:
      OutputString := 'Tim';
  2,3,4:
      OutputString := 'Steve';
  5..9:
      OutputString := 'Alan';
ELSE
      OutputString := 'Empty';
END_CASE;
```

Comment (Comentario)

Use la declaración COMMENT para incorporar anotaciones útiles dentro de su código de programas. El formato para un comentario es el siguiente:

```
(*<texto libre de forma>*)
```

El siguiente es un ejemplo para la orden **Comment**:

```
(* Selección de un nuevo mensaje a mostrar *)
```

```
CASE ISelection OF
```

```
  0:
```

```
(* Este selecciona "Dave" como la salida *)
```

```
  OutputString := 'Dave' ;
```

```
  1:
```

```
(*Este selecciona "Shiela" como la salida *)
```

```
  OutputString := 'Shiela' ;
```

```
END_CASE;
```

También puede hacer comentarios de mas de una línea:

```
(* Fija intvarA a 3
```

```
  Luego fija intvarB a 4 *)
```

Nota: Si usted inserta un comentario dentro de otro comentario ocurrirá un error.

EXIT

Use la declaración EXIT para salir de un proceso iterativo o repetitivo, e.g. una declaración FOR o WHILE, antes que la terminación normal del proceso ocurra.

Su formato es el siguiente:

```
<condition for exiting> EXIT;
```

Donde **<condition for exiting>** es una expresión que determina si se realizará una terminación temprana.

Si usa la declaración EXIT con una iteración anidada, la salida ocurre desde el lazo en el cual el EXIT esta localizado, y el flujo del programa continua después de la declaración que normalmente finaliza la iteración o declaraciones repetitivas, por ejemplo: END_FOR, END_WHILE, etc.

El siguiente ejemplo muestra la operación de una orden EXIT. Cuando la variable llamada **cancel** es igual a 0, entonces la variable llamada **tally** se hace 0; cuando la variable llamada **cancel** no es igual a 0, **tally** es igual a 46.

```
tally :=0;
FOR counta := 1 TO 4 DO
  FOR countb := 1 TO 3 DO
    FOR countc := 1 TO 2 DO
      IF cancel = 0 THEN EXIT; END_IF;
      tally := tally + countc;
    END_FOR;
  END_FOR;
  tally := tally + counta;
END_FOR;
```

Nota: Típicamente, un lazo bien diseñado no requiere una orden EXIT. Use esta orden escasamente. Para evitar una condición de lazo interminable, asegúrese que todas las condiciones para completar cualquier iteración anidada estén satisfechas después que un EXIT es ejecutado.

FOR

Use esta orden para ejecutar una serie de ordenes repetitivas, con el número de repeticiones basadas en el valor de una variable de control. Cada iteración cambia el valor de la variable de control (el valor predefinido es 1), el cuál puede ser una expresión, y que sigue la porción BY de la orden. La variable de control puede ser positiva o negativa. La orden FOR revisa la variable de control antes de cada iteración y las ordenes dentro de FOR/END_FOR no son ejecutados cuando el valor actual de la variable de control ha alcanzado (o excedido) el límite.

El formato para la orden FOR es el siguiente:

```
FOR <INT variable> := <expression> TO <expression> [BY
<expression>] DO
    <statement list>
END_FOR;
```

Donde **<INT variable>** es un tipo de dato ANY_INT, **<expression>** resuelve un tipo de dato ANY_INT, y **<statement list>** es cualquier grupo de ordenes en Texto Estructurado. Para evitar la posibilidad de diseñar un lazo que no tenga fin, se sugiere que no cambie el valor de la variable de control del lazo dentro del lazo.

La orden END_FOR hace que el sistema espere para análisis I/O al final de cada ciclo del lazo FOR. Como una mejora en las especificaciones del IEC-61131-3 usted puede usar la orden END_FOR_NOWAIT para volver al lazo sin continuar las otras tareas programadas (las entradas y salidas de otros programas no serán tenidas en cuenta). Usted puede usar EXIT para finaliza el FOR antes de su normal terminación.

Advertencia: Si la orden END_FOR_NOWAIT ocasiona una condición de lazo interminable, el Watchdog Timeout expirara. Esto causa que el servicio del software de ejecución se interrumpa y suspenda todos los programas, con un riesgo potencial de muerte o lesiones al personal y/o daños al equipo. Diseñe y revise su código para verificar que una condición de lazo interminable no ocurra. Si usa una orden EXIT dentro de una iteración anidada, asegúrese que todas las condiciones para completar la iteración estén satisfechas para evitar este tipo de lazo infinito.

No use una variable TMR <TMRname>.Q dentro de una condición de sección FOR si también usa la orden END_FOR_NOWAIT. Esta variable no es procesada hasta que el lazo FOR termine, y el lazo no puede terminar la ejecución hasta que la variable sea procesada.

Nota: Las siguientes ordenes no pueden aparecer en la misma línea: BREAK, SCAN, END_FOR, END_FOR_NOWAIT, END_WHILE, y END_WHILE_NOWAIT. La orden END_FOR no puede ser usada en funciones o bloques de funciones. Use END_FOR_NOWAIT para completar la orden FOR.

El siguiente ejemplo muestra la operación de una orden FOR.

```
total :=0;
FOR count := 1 TO 100 DO
    total := total + 1;
END_FOR;
total:=0;
FOR t := 10 TO 1 BY -1 DO
    total := total + 5;
END_FOR;
```

Llamadas de Funciones y Procedimientos

Estas llamadas ejecutan uno de los algoritmos predefinidos que le permitan hacer cálculos, funciones lógicas, operaciones de traslado de bits, etc. También puede usar estas llamadas para ejecutar métodos FOE y funciones de usuario definidas o bloques de funciones.

Una función que retorne un valor opera como una función verdadera y se usa al lado derecho de la orden de asignación. El formato para la llamada de función es la siguiente:

```
<result> := <function name> (parameter_1, parameter_2, ...);
```

El siguiente ejemplo muestra la sintaxis de la llama a la función TAN.

```
TrigAnswer := TAN (input);
```

Una función que no retorne un valor opera como un procedimiento. El formato para una llamada de procedimiento es la siguiente:

```
<procedure name> (parameter_1, parameter_2, . . .) ;
```

El siguiente ejemplo muestra la sintaxis del procedimiento OPENFILE:

```
OPENFILE (FCB:= <fcb>, FILE:= <filename>);
```

Si un parámetro no es del tipo de dato correcto, ocurre un error de validación.

Para mas información acerca del diseño de una función definida por el usuario o bloque de funciones, ver el capítulo “**ORGANIZACIÓN Y MANEJO DE PROYECTOS**”. Ejemplos de llamadas para bloques de funciones también se encuentran en este capítulo.

Todos los FOEs, los cuales no tienen un método programado para ejecutar automáticamente, y todas las funciones y bloques de funciones ingresan al modo Loaded (cargado) cuando los descarga al software de ejecución. Un FOE, función, o bloque de función que este en modo Loaded, ha sido cargado en el software de ejecución y corre cuando es llamado para su ejecución.

IF

Use la orden IF para designar la ejecución de un juego de ordenes solamente cuando una variable Booleana o expresión sea TRUE (verdadera). La construcción de la orden IF consiste de lo siguiente:

- Una expresión Boolean precedida por el IF, y seguida por THEN, esto ocasiona que el conjunto de declaraciones se ejecuten cuando la expresión sea TRUE.
- Una segunda expresión Boolean (opcional) precedida por ELSEIF, y seguida por THEN, esto ocasiona que un segundo grupo de declaraciones se ejecuten si el primer grupo de declaraciones no corre y la segunda condición es TRUE.
- Un grupo de declaraciones precedida por ELSE (opcional) esto es ejecutado si las declaraciones IF y ELSEIF no se efectúan.
- Una declaración END_IF es requerida para cerrar la declaración IF.

Si ninguna expresión Booleana es TRUE y usted ha incluido una orden ELSE, las ordenes siguientes a ELSE seran ejecutadas. Si no hay orden ELSE, ninguna orden es ejecutada dentro de la construcción IF.

El formato para la orden IF es el siguiente:

```
IF   <Boolean expression>      THEN
      <statement list>
[ ELSEIF <Boolean expression>    THEN
      <statement list> ]
[ ELSE
      <statement list> ]
END_IF;
```

Donde **<Boolean expression>** es cualquier expresión que evalúa un valor Boolean, y **<statement list>** es cualquier conjunto de declaraciones de Texto estructurado válido.

El siguiente es un ejemplo de la orden IF:

```
com_value := com_input_BUFF[11];
IF COM_VALUE = 11 THEN
lSelection := lSelection + 1;
ELSEIF COM_VALUE = 10 THEN
lSelection:= lSelection -1;
ELSE
lSelection:= 0;
END_IF;
```

INCLUDE

Use la declaración INCLUDE para llamar un archivo externo y ejecutar un grupo de declaraciones contenidas dentro de un archivo.

El formato para la declaración INCLUDE es el siguiente:

```
INCLUDE '< string >';
```

Donde **<string>** especifica el camino y el archivo en donde esta contenida la declaración a ser ejecutada.

Puede usar mas de una orden INCLUDE si necesita llamar múltiples archivos. Los comandos del programa pueden ser mezclados con la orden INCLUDE. Para simplificar la codificación y el mantenimiento, se sugiere no usar la declaración INCLUDE en sus programas.

El siguiente es un ejemplo del uso de la declaración INCLUDE.

```
INCLUDE 'C:\ST_FILES\STFILE1.TXT';
```

LABEL (Etiqueta)

Puede usar un LABEL en un paso SFC para control cuando el código RLL asociado con un Paso o MacroPaso es ejecutado. Si una etiqueta es especificada, el RLL no corre hasta que la etiqueta en el código del paso sea encontrada.

Note que la orden LABEL no tiene significado en un programa de Texto Estructurado, solamente dentro de un paso SFC.

El formato para la declaración LABEL consiste de un nombre de etiqueta seguido por dos puntos, como se muestra abajo:

```
Label_A::
```

Para mas información acerca de los LABELs y Actions, ver el capítulo “ELEMENTOS DE PROGRAMAS SFC”.

REPEAT

Use la declaración REPEAT para diseñar una ejecución repetida de un grupo de declaraciones hasta que una condición Boolean se vuelva TRUE. Las declaraciones siempre se ejecutan en un tiempo mínimo antes que la expresión Boolean sea evaluada.

La declaración END_REPEAT causa que el sistema examine las entradas y salidas al final de cada ciclo de un lazo REPEAT. Puede usar el EXIT para finalizar el REPEAT antes de su culminación normal.

El formato para la declaración REPEAT es el siguiente:

```
REPEAT
    <statement list>
UNTIL    <Boolean expression>    END_REPEAT;
```

Donde **<Boolean expression>** es cualquier expresión que resuelve a un valor Boolean, y **<statement list>** es cualquier conjunto de declaraciones en Texto Estructurado.

Nota: Las siguientes declaraciones no pueden aparecer sobre la misma línea: BREAK, SCAN, END_FOR, END_FOR_NOWAIT, END_WHILE, y END_WHILE_NOWAIT.

La declaración END_REPEAT no puede ser usada en funciones o bloques de funciones. Use END_REPEAT_NOWAIT para completar la declaración REPEAT.

El siguiente es un ejemplo con la declaración REPEAT.

```
ZZ := 0;
REPEAT
    ZZ := ZZ+5;
UNTIL ZZ >= 100 END_REPEAT;
```

RETURN

Use RETURN en una función, procedimiento, o bloque de funciones para provocar que el flujo del programa continúe en la POU que llamó a la función, procedimiento o bloque de función para ejecutar.

El formato para la declaración RETURN es el siguiente:

```
RETURN;
```

SCAN

Use la declaración SCAN para suspender la ejecución de declaraciones en texto estructurado hasta después del próximo I/O Scan. El formato para la declaración SCAN es la siguiente:

```
SCAN;
```

Nota: Las siguientes ordenes no pueden aparecer sobre la misma línea: BREAK, SCAN, END_FOR, END_FOR_NOWAIT, END_WHILE, y END_WHILE_NOWAIT.

Las declaraciones con SCAN no son permitidas en funciones o bloques de funciones.

El siguiente es un ejemplo de una declaración con SCAN.

```
ASelection := ASelection + 1;
SCAN;
IF ASelection >= 3 THEN
    ASelection := 0;
ELSEIF ASelection < 0 THEN
    ASelection := 2;
END_IF;
```

WHILE

Use la declaración WHILE para diseñar la ejecución repetida de un juego de ordenes hasta que una condición Boolean se convierta en FALSE. Si la condición Boolean es inicialmente FALSE, entonces las ordenes no son ejecutadas del todo.

La declaración END_WHILE ocasiona que el sistema haga una revisión de las entradas y salidas (I/O Scan) al final de cada ciclo del lazo REPEAT. Como una mejora al IEC-61131-3, puede usar el END_REPEAT_NOWAIT para cerrar el lazo sin continuar las otras tareas en la misma línea del tiempo. Puede usar el comando EXIT para finalizar el REPEAT antes de su normal culminación.

El formato para la declaración WHILE es la siguiente:

```
WHILE <Boolean expression> DO
    <statement list>
END_WHILE;
```

Donde <Boolean expression> es cualquier expresión que resuelve un valor Boolean, y <statement list> es cualquier juego de declaraciones de texto

estructurado válido. Siempre escriba un punto y coma después de la declaración END_WHILE y de cada declaración.

El siguiente es un ejemplo de la declaración WHILE.

```
VAL_A := 1;
WHILE (VAL_A <= 100) DO
    VAL_A := VAL_A*5;
END_WHILE;
```

#Pragma

Use la declaración #pragma para cambiar la forma del mensaje transmitido por el compilador cuando un programa es validado y hay una o más peticiones de escribir un valor de salida al símbolo I/O (símbolos de entrada y salida) que ha sido definido como una entrada. El formato para la declaración #pragma es la siguiente:

```
#pragma <parameter>
```

Donde <parameter> puede ser uno de estos valores: **IOWriteError**, **IOWriteWarn**, **IOWriteIgnore**. Una declaración #pragma debe empezar en la primera columna, ser el único ítem sobre la línea, y no llevar punto u coma. La declaración #pragma afecta solamente el programa en el cual el es usado.

El siguiente es un ejemplo del uso de la declaración #pragma. Una advertencia es generada si el programa incluye una petición de escribir un valor de salida a una entrada I/O.

```
#pragma IOWriteWarn
```

El siguiente es otro ejemplo del uso de #pragma. Un mensaje no es generado si el programa incluye una instancia de escribir un valor de salida a la entrada I/O.

```
#pragma IOWriteIgnore
```

FUNCIONES Y PROCEDIMIENTOS DE TEXTO ESTRUCTURADO

Las funciones y procedimientos en Texto Estructurado son algoritmos predefinidos que llevan a cabo operaciones singulares.

- Para funciones, el resultado de la operación se asigna al objeto que se encuentra a la izquierda de una declaración asignada.
- Los procedimientos requieren que el código lea las salidas de la operación para acceder al resultado.

Las funciones y procedimientos de Texto Estructurado están listados en la siguiente tabla.

Funciones y Procedimientos por Grupo

Grupo	Función/ Procedimiento	Descripción	Página
Bitwise	ROL	Rota a la izquierda la entrada según el número de bits especificado por el número de desplazamiento.	11-61
	ROR	Rota a la derecha la entrada según el número de bits especificado por el número de desplazamiento.	11-62
	SHL	Mueve a la izquierda la entrada el número de bits especificado por el número de desplazamiento.	11-63
	SHR	Mueve a la derecha la entrada según el número de bits especificado por el número de desplazamiento.	11-64
Conversión	ARRAY_TO_STRING	Convierte una fila de bits en una cadena.	11-26
	BCD_TO_INT	Convierte el valor BCD de la entrada a un numero entero.	11-30
	DATE_TO_REAL	Convierte el contenido de un tipo de dato DATE a un tipo de dato REAL.	11-36
	DATE_TO_STRING	Convierte el contenido de un tipo de dato DATE a una cadena de caracteres.	11-37
	INT_TO_BCD	Convierte un numero entero a su valor equivalente BCD.	11-44
	INT_TO_STRING	Convierte un numero entero a una cadena de caracteres ASCII.	11-45

Grupo	Función/ Procedimiento	Descripción	Página
Conversión	REAL_TO_DATE	Convierte la entrada real a un tipo de dato DATE.	11-55
	REAL_TO_STRING	Convierte la entrada real a una cadena ASCII.	11-56
	REAL_TO_TIME	Convierte el contenido de un tipo REAL a un tipo de dato TIME.	11-57
	STRING_TO_ARRAY	Lleva una cadena de caracteres entrada y almacena los caracteres de una cadena en un arreglo de bytes.	11-66
	STRING_TO_DATE	Convierte el contenido de una cadena ASCII a un tipo de dato DATE.	11-67
	STRING_TO_INT	Convierte el contenido de un ASCII a un tipo de dato ANY_INT.	11-68
	STRING_REAL	Convierte el contenido de una cadena ASCII a un tipo de dato REAL.	11-69
	STRING_TO_REAL	Convierte el contenido de un tipo de dato TIME a un tipo de dato REAL.	11-70
	TIME_TO_STRING	Convierte el contenido de un tipo de dato TIME a un tipo de dato ASCII.	11-72
File	CLOSEFILE	Cierra y archivo.	11-74
	COPYFILE	Copia un archivo.	11-31
	DELETEFILE	Elimina un archivo.	11-35
	NEWFILE	Crea un Nuevo archivo.	11-39
	OPENFILE	Abre un archivo existente.	11-52
	READFILE	Lee un dato desde un archivo.	11-53
	REWINDFILE	Pone el puntero del archivo interno al principio del archivo.	11-59
	WRITEFILE	Escribe el dato al archivo.	11-76
Math	ABS	Calcula el valor absoluto de un número.	11-24
	EXPT	Eleva el primer número a la potencia especificada por el segundo número.	11-41
	MAX	Determina el máximo de dos entradas.	11-49
	MIN	Determina el mínimo de dos entradas.	11-51
	SQRT	Calcula la raíz cuadrada de un número.	11-65
	TRUNC	Remueve uno o más de los dígitos significativos de un tipo de dato REAL.	11-75

Grupo	Función/ Procedimiento	Descripción	Página
Math	ABS	Calcula el valor absoluto de un número.	11-24
	EXPT	Eleva el primer número a la potencia especificado por el segundo número.	11-41
	MAX	Determina el máximo de dos entradas.	11-49
	MIN	Determina el mínimo de dos entradas.	11-51
	SQRT	Calcula la raíz cuadrada de un número.	11-65
	TRUNC	Remueve uno o más de los dígitos significativos de un tipo de dato REAL.	11-75
Variado	ABORT_ALL	Aborta todos los programas que estén corriendo.	11-24
	MSGWND	Visualiza un mensaje en la ventana de salida y el Wonderware Logger.	11-51
String	CONCAC	Concatena dos cadenas de caracteres.	11-34
	DELETE	Borra caracteres desde la mitad de la cadena de caracteres.	11-38
	FIND	Busca una cadena de caracteres dentro de otra cadena de caracteres.	11-42
	INSERT	Inserta una cadena de caracteres dentro de otra cadena de caracteres.	11-43
	LEFT	Copia el carácter mas a la izquierda de la cadena de caracteres de entrada.	11-46
	LEN	Almacena la longitud de una cadena de caracteres de entrada.	11-47
	MID	Copia caracteres desde la mitad de una cadena de caracteres de entrada.	11-50
	REPLACE	Reemplaza caracteres en una cadena de caracteres de entrada con otra cadena de caracteres de entrada.	11-58
	RIGHT	Copia los caracteres mas a la izquierda de la cadena de caracteres de entrada.	11-60
Trig/Log	ACOS	Calcula el arc coseno de un número.	11-25
	ASIN	Calcula el arc seno de un número.	11-28
	ATAN	Calcula el arc tangente de un número.	11-29
	COS	Calcula el coseno de un número.	1-36
	EXP	Calcula el exponente natural de un número.	11-40
	LN	Calcula el logaritmo natural de n número.	11-48

Grupo	Función/ Procedimiento	Descripción	Página
Trig/Log	SIN	Calcula el seno de un número.	11-65
	TAN	Calcula la tangente de un número.	11-71

ABORT_ALL

Este procedimiento aborta todos los programas que estén corriendo en el software de ejecución.

El formato para el procedimiento ABORT_ALL es el siguiente:

```
ABORT_ALL ( ) ;
```

La operación es como sigue:

- ABORT_ALL aborta todos los programas (RLL, SFC, Texto Estructurado), cambia el icono del monitor del software de ejecución a la condición de error y fija el software de ejecución al modo Fault (falla).
- Ninguna otra orden que siga el ABORT_ALL es ejecutado.

Advertencia: El procedimiento ABORT_ALL detiene todos los programas en un proyecto. Una Suspensión no deseable de todos los programas puede causar operaciones impredecibles por los equipos de salida, los cuales pueden resultar en lesiones o muerte de personal y/o daños a equipos. Diseñe su código de programa muy cuidadosamente de manera que el ABORT_ALL sea solo ejecutado las condiciones cuidadosamente controladas.

ABS

La función ABS calcula el valor absoluto de la entrada.

El formato para la función ABS es la siguiente:

```
<result> := ABS(<number>);
```

Parámetros	Descripción
------------	-------------

<result>	Un tipo de dato variable ANY_NUM.
----------	-----------------------------------

<number>	Cualquier número, variable, o expresión que resuelva un tipo de dato ANY_NUM
----------	--

El siguiente es un ejemplo de una función ABS.

```
absNum := ABS(num);
If num = -99, then absNum = 99.
```

ACOS

Esta función calcula el arc coseno de la entrada. El resultado es en radianes. El formato para la función ACOS es la siguiente:

```
<result> := ACOS(<number>);
```

Parámetros	Descripción
<result>	Un tipo de dato variable ANY_REAL.
<number>	Cualquier número, variable, o expresión que resuelva un tipo de dato ANY_REAL. El rango válido es -1.0 hasta +1.0.

El siguiente es un ejemplo de la función ACOS.

```
angle := ACOS(num);
```

Si el número es 0.7071067, entonces el ángulo es 0.7853975.

ARRAY_TO_STRING

El procedimiento ARRAY_TO_STRING convierte una fila de entrada de bytes a un a longitud de string.

El formato del procedimiento ARRAY_TO_STRING es el siguiente:

```
ARRAY_TO_STRING(OUT := <string> , IN := <array>);
```

Parámetro	Descripción
<string>	Una variable (tipo de dato STRING), la longitud igual al tamaño de la fila.
<array>	Un arreglo de tipos de datos BYTE. Cada byte en el arreglo contiene un código decimal de caracteres ASCII.

La operación es la siguiente. ARRAY_TO_STRING convierte cada byte dentro de la fila especificada por la entrada (IN) a un string (cadena de caracteres) y almacena el resultado en la localización especificada por el parámetro de salida OUT.

El siguiente es un ejemplo del procedimiento ARRAY_TO_STRING.

```
ARRAY_TO_STRING(OUT := <string> ,IN := <array>);
```

Si byte_array consiste de 12 elementos con los valores que se muestran en la siguiente figura, entonces Cnv_string contiene INCONTROL.

Variables mostradas en la ventana Watch	
Elementos del arreglo de bytes	Valor
1	73
2	78
3	67
4	79
5	78
6	84
7	82
8	79
9	76
10	0
11	0
12	0
Símbolo	Valor
Cnv_string	INCONTROL

Debido a que la cadena de caracteres es de longitud fija, la salida del ARRAY_TO_STRING es de igual longitud que el tamaño del arreglo. Si la declaración RIGHT es ejecutada sobre la salida del ARRAY_TO_STRING, el cálculo empieza en el elemento mas a la derecha en la salida. Esto es, si la siguiente orden RIGHT es ejecutada en el ejemplo:

```
get_string_rght_chars := RIGHT (IN:= Cnv_string, L:= 2);
```

La cadena de caracteres que es creada contiene el valor OL cuando los elementos 8-9 de la fila contienen 82 y 79, respectivamente. Ver la siguiente figura.

Variables mostradas en la ventana Watch	
Elementos del arreglo de bytes	Valor
1	73
2	78
3	67
4	79
5	78
6	84
7	82
8	79
9	76
10	00
11	66
12	67
Símbolo	Valor
Cnv_string	INCONTROL
Get_string_rght_chars	OL

ASIN

La función ASIN calcula el arc seno de la entrada. El resultado es en radianes.

El formato para la función ASIN es la siguiente:

<result> := ASIN(<number>);

Parámetro	Descripción
-----------	-------------

<result>	Una variable ANY_REAL tipo de dato (REAL, LREAL).
-----------------------	---

<number>	Un número, variable, o expresión que resuelva un tipo de dato ANY_REAL (REAL, LREAL). El rango válido es -1.0 hasta +1.0.
-----------------------	---

El siguiente es un ejemplo de la función ASIN.

```
angle := ASIN(num);
```

Si num = 0.5, entonces angle = 0.5235983.

ATAN

La función ATAN calcula el arc tangente de la entrada. El resultado es en radianes. El formato para la función ATAN es la siguiente:

```
<result> := ATAN(<number>);
```

Parámetro	Descripción
<result>	Una variable ANY_REAL (REAL, LREAL).
<number>	Un número, variable, o expresión que resuelva un tipo de dato ANY_REAL (REAL, LREAL). El rango válido es -1.0 hasta +1.0.

El siguiente es un ejemplo de la función ATAN:

```
angle := ATAN(num);
```

Si num = 1.0, Entonces angle = 0.78539816.

BCD_TO_INT

La función BCD_TO_INT convierte una entrada decimal de valor BCD (código binario decimal) a uno integer.

El formato es:

```
<result> := BCD_TO_INT (<number>);
```

Parámetro	Descripción
<result>	Una variable tipo dato ANY_INT (SINT, INT, DINT, BYTE, WORD, DWORD)
<number>	Un número BCD, variable o expresión que resuelve a un tipo de dato ANY_INT (SINT, INT, DINT, BYTE, WORD, DORD).

La operación es la siguiente. BCD_TO_INT convierte la representación integer de la entrada BCD a un integer con el valor de la entrada BCD. Si <number> es una entrada BCD inválida, <result> se pone en -1.

El siguiente es un ejemplo de la función BCD_TO_INT.

```
intout := BCD_TO_INT(bcd_in);
```

Si bcd_in = 0311 entonces intout = 311 (0000 0001 0011 0111), como se muestra en la siguiente figura.

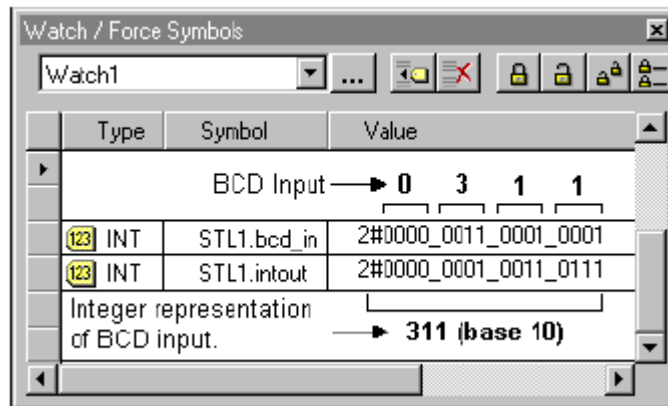


Figura 3. Representación en números enteros de una entrada BCD.

CLOSEFILE

El procedimiento CLOSEFILE cierra un archivo que ha sido abierto por el procedimiento OPENFILE. El CLOSEFILE es uno de los ocho procedimientos que archiva operaciones o funcionamientos. Note que estos procedimientos no están diseñados para ejecuciones de alta velocidad (high-speed) I/O, Transferencia de datos de bloques grandes de información, o aplicaciones para control.

El formato para el procedimiento CLOSEFILE es el siguiente:

```
CLOSEFILE (<fcb>);
```

Parámetro Descripción

Parámetro	Descripción
<fcb>	Nombre del bloque de control del archivo que maneja operaciones para este archivo. Declare <fcb> en el Symbol Manager como un tipo de dato FILE.

La operación es la siguiente. Cierre el archivo CLOSEFILE que esta asociado con el bloque de mando, el cual usted especifica con el parámetro <fcb>. Note que no diseña el archivo por su nombre de archivo, solamente por el nombre del bloque de control. Todos los procedimientos del archivo que operan en el mismo archivo deben usar el mismo parámetro de bloque de control (File Control Block).

Si un error ocurre, la variable de error del archivo se pone en TRUE, y un mensaje aparece en la ventana de salida (Output) y en el Wonderware Logger.

Todos los procedimientos tipo archivo usan variables de bloques de función para manejar control de archivos. Tres de estas variables especifica el estado del archivo después que se abre: lectura, escritura, si un dato puede o no ser añadido, o si otra aplicación puede o no acceder al archivo. Ocho variables suministran un significado de error de monitoreo, si un archivo esta en uso, cuando una operación es completada, etc. Puede usar cualquiera de estas variables en una expresión, contacto o bobina en vez de un símbolo del mismo tipo. Para referenciar una variable, entre el nombre del bloque de control seguido por un periodo y el nombre específico de la variable. Por ejemplo, **FILEA.BUSY** se refiere a la variable de bloque de control **BUSY** para el archivo referenciado por el bloque de control **FILEA**.

Para las tres variables que especifican los estados después de que se abren, puede especificar valores iniciales para las variables en el cuadro de diálogo **Symbol Properties** en el Symbol Manager. Si prefiere, puede usar declaraciones de asignación en su programa de texto estructurado para asignar valores a las variables antes que abra el archivo. Estas tres variables de entrada están mostradas en la tabla Variables de Entrada de Control de Archivos.

Variables de Entrada de Control de Archivos

Variable	Descripción
fcb.ACCESS¹	Variable byte que especifica estados de lectura o escritura del archivo después de abierto. 0 = El archivo es abierto para operaciones de lectura o escritura. 1 = El archivo se abre para operaciones de solo lectura. 2 = El archivo se abre para operaciones de solo escritura.
fcb.APPEND¹	Variable Boolean que especifica si o no un dato puede se añadido a el archivo después de abierto. Solo es válido cuando el archivo esta abierto con estados de escritura. Esto es, la variable ACCESS = 0 o 2. TRUE = el dato será añadido al archivo. FALSE= (predefinido) el dato no puede ser añadido a el archivo.
fcb.SHARE¹	Variable byte que especifica cómo otra aplicación puede acceder al archivo después de abierto. 0 = (predefinido) otras aplicaciones pueden acceder al archivo para operaciones de lectura y escritura. 1 = Otras aplicaciones pueden acceder al archivo para aplicaciones de solo lectura. 2 = Otras aplicaciones pueden acceder al archivo para aplicaciones de solo escritura. 3 = Otras aplicaciones no pueden acceder al archivo.

1. *Estas variables son leídas y se llevan a cabo solamente cuando los procedimientos OPENFILE y NEWFILE son ejecutados.*

Las ocho variables que manejan operaciones de archivos están listadas en la siguiente tabla.

Variables de Salida de Control de Archivos

Variable	Descripción
fcb.OPEN	Variable Boolean que indica el archivo que ha sido abierto. El sistema fija la variable File Open a TRUE cuando el archivo es abierto.
fcb.BUSY	Variable Boolean que indica que el archivo esta siendo accedido. El sistema pone la variable File Control Busy a TRUE cuando el archivo esta siendo accedido por otro archivo de función. Si usted intentó ejecutar un archivo tipo función no puede ejecutar mientras esta variable es TRUE, un error ocurre (código del error 15).
fcb.EFLAG	Variable Boolean que indica cuando un error ocurre. Si un error ocurre durante una operación de archivo, el sistema fija la variable File Error a TRUE. Esta variable no se restablece (reset) automáticamente; el programa debe restablecer la variable. Usted también puede restablecerlo manualmente a través de la ventana Watch. Un archivo tipo función no puede ejecutar mientras esta variable es TRUE. El programa no va dentro del modo Fault cuando un error ocurre.
fcb.RDN	Variable Boolean que indica que una operación de lectura ha sido completada. El sistema fija la variable File Read Done a TRUE cuando la operación de lectura es terminada.
fcb.WDN	Variable Boolean que indica que una operación de escritura ha sido completada. El sistema fija la variable File Write Done a TRUE cuando la operación de escritura es terminada.
fcb.CLSD	Variable Boolean que indica que un archivo ha sido cerrado. El sistema fija la variable File Close a TRUE cuando el ha cerrado el archivo.
fcb.EOF	Variable Boolean que indica que el sistema encontró un final de archivo (End Of File). El sistema fija el End Of File a TRUE el encuentra el EOF.
fcb.ERR	Variable Integer que encierra el código de error si un error ocurre. Si un error ocurre durante una operación de archivo, el sistema escribe un código de error para el File Error Code integer. La tabla que sigue lista los códigos de error.

Códigos de Error para Procedimientos con Archivos

Código de Error	Descripción de Error
15	Bloque de Control de archivo esta ocupado.
16	Nombre de archivo no especificado.
17	El archivo no ha sido abierto.
18	Archivo no encontrado.
19	Disco lleno.
20	Falla de lectura.
21	Falla de copia de archivo.
22	Falla de escritura.
25	Falla de cierre de archivo.
26	Archivo ya existente.
27	Archivo esta abierto.
28	Archivo de solo lectura.
29	Falla abertura de archivo.
30	Error general.
31	Alcance final de archivo.
32	Violación al compartir. Otra operación tiene cerrado el archivo.
33	Acceso denegado. Puede ocurrir si pide un tipo de acceso y usa otro.
34	Memoria insuficiente. El tamaño máximo del campo de dato que puede escribir o leer es 1KB.
35	Dato malo. Puede ocurrir si los tipos de datos especificados son desiguales.

Puede hacer solamente una operación de archivo para cada File Control Block en un momento. Note que las operaciones I/O de control de archivos tienen lugar ejecución del programa.

El siguiente es un ejemplo del procedimiento CLOSEFILE.

```
CLOSEFILE (FCB:= datrpt);
```

El sistema cierra el archivo asociado con el bloque de función de archivo llamado datrpt.

Para un ejemplo que use el CLOSEFILE con algunos otros procedimientos File, ver WRITEFILE (pag 11-76).

CONCAT

La función CONCAT concatena un string de entrada al final de otro string de entrada.

El formato para la función CONCATENATE es la siguiente:

```
<result> := CONCAT (<stringa>, <stringb>);
```

Parámetro	Descripción
-----------	-------------

<result>	Una variable (tipo de dato STRING)
----------	--

<stringa>	Cualquier valor STRING, o variable o expresión que resuelve un
-----------	--

<stringb>	tipo de dato STRING.
-----------	----------------------

Si la suma de las longitudes de las dos cadenas de caracteres es mayor que 2048,

<result> se ajusta como una cadena de caracteres vacío.

El siguiente es un ejemplo de la función CONCAT.

```
fullname := CONCAT (firstname,lastname);
```

Si firstname = sara_ y lastname = svensen, entonces el fullname = sara_svensen, como se muestra en la siguiente figura:

Variables mostradas en la ventana Watch	
Símbolo	Valor
STL3.firstname	Sara_
STL3.lastname	Svensen
STL3.fullname	Sara_svensen

COPYFILE

El procedimiento COPYFILE copia un archivo existente. El COPYFILE es uno de los ocho procedimientos que hacen operaciones de archivos. Note que estos procedimientos no están diseñados para velocidades altas de ejecución I/O, transferencia de datos de bloques grandes de información, o para aplicaciones de control.

El formato para el procedimiento COPYFILE es el siguiente:

```
COPYFILE(FCB:= <fcb>,OUT:= <tofilename> IN:= <fromfilename>);
```


Parámetros	Descripción
<fcb>	Nombre del bloque de control del archivo que maneja operaciones para este archivo. Declare <fcb> en el Symbol Manager como un tipo de dato FILE.
<tofilename>	Contiene el nombre del archivo para el cual la fuente es copiada. Una variable tipo de dato ANY.
<fromfilename>	Contiene el nombre de archivo que será copiado. Una variable tipo de dato ANY.

La operación es la siguiente. COPYFILE copia el archivo especificado por **<fromfilename>** al nuevo archivo y le asigna el nombre que esta especificado por **<tofilename>**. El bloque de control del archivo maneja el acceso al archivo y condiciones de error. Todos los procedimientos de archivo que operan en el mismo archivo deben usar el mismo nombre del bloque de control del archivo. Además, así como diseñe su código, debe usar las variables de control del archivo para manejar el control. Para una descripción detallada de las variables de control de archivos, ver el procedimiento CLOSEFILE.

Si un error ocurre, la variable de error del archivo se pone en verdadero, y un mensaje aparece en la ventana de salida y en el Wonderware Logger. Intentar copiar sobre un archivo existente genera un error (código de error 26). Para una lista de todos los códigos de error, ver la tabla de Códigos de Error para procedimientos tipo Archivo.

Puede hacer solamente una operación de archivo por cada Bloque de Control de Archivo a la vez. Note que las operaciones E/S de control de archivo tienen lugar para programar ejecuciones asíncronas.

El siguiente es un ejemplo del procedimiento COPYFILE.

```
COPYFILE(FCB:= datrpt, OUT:= "newdatcopy" IN:= "olddatcopy");
```

El sistema hace una copia del archivo llamado olddatcopy y lo nombra newdatcopy.

COS

La función COS calcula el coseno de una entrada, el cual debe estar en radianes.

El formato para la función COS es la siguiente:

```
<result> := COS(<number>);
```

Parámetros	Descripción
<result>	Una variable tipo de dato ANY_REAL (REAL, LREAL).
<number>	Cualquier número, variable, o expresión que resuelve un tipo de dato ANY_REAL (REAL, LREAL).

El siguiente es un ejemplo de la función COS.

```
cosangle := COS(angle);
```

Si angle = 6.0, entonces cosangle = 0.96017029.

DATE_TO_REAL

La función DATE_TO_REAL convierte el contenido de un tipo de dato ANY_DATE a tipo de dato REAL.

El formato para esta función es:

```
<result> := DATE_TO_REAL (<inputdate>);
```

Parámetros	Descripción
<result>	Una variable tipo de dato ANY_REAL (REAL, LREAL).
<inputdate>	Cualquier número, variable, o expresión que resuelve un tipo de dato ANY_DATE (DT, DATE, TOD).

La operación es la siguiente. DATE_TO_REAL convierte el valor representado por el contenido de <inputdate> y almacena el resultado como un tipo de dato LREAL en <result>. El siguiente es un ejemplo de la función DATE_TO_REAL.

```
realvar := DATE_TO_REAL (datevar);
```

Si datevar contiene D#1999-12-31, entonces realvar equivale 36525.

Nota Para ayudar a asegurar exactitud, use el tipo de dato LREAL para el resultado cuando convierte un tipo de dato DT.

DATE_TO_STRING

La función DATE_TO_STRING convierte el contenido del tipo de dato DATE a un tipo de dato STRING.

El formato para la función DATE_TO_STRING es el siguiente:

```
<result> := DATE_TO_STRING (<inputdate>);
```

Parámetros	Descripción
<result>	Una variable (tipo dato STRING).
<inputdate>	Cualquier número, variable, o expresión que resuelve un tipo dato ANY_DATE(DT, DATE, TOD).

La operación es la siguiente. DATE_TO_STRING convierte el contenido de <inputdate> a un carácter ASCII y lo almacena dentro del string <result> con el mismo tipo de dato que <inputdate>.

El siguiente es un ejemplo de la función DATE_TO_STRING.

```
stringvar := DATE_TO_STRING (datevar);
```

Si datevar contiene D#1999-05-06, entonces stringvar equivale a 'D#1999-05-06'.

DELETE

La función DELETE borra un número especificado de caracteres desde una posición en la mitad de una entrada string.

El formato para la función DELETE es el siguiente:

```
<result>:=DELETE (IN:=<stringa>,L:= <length>, P:= <position>);
```

Parámetros	Descripción
<result>	Una variable (tipo dato STRING).
<stringa>	Cualquier valor, o variable o expresión que resuelva un tipo de dato STRING.
<length>	Cualquier número, variable, o expresión que resuelva un tipo de dato ANY_INT (SINT, INT, DINT, BYTE, WORD, DWORD).
<position>	

La operación es la siguiente. DELETE remueve de <stringa> el número de caracteres especificado por <length>, empezando en el carácter especificado por <position>. DELETE almacena el resultado en <result>. Su el valor de P o L es negativo, <result> se coloca en un string vacío.

El siguiente es un ejemplo de la función DELETE:

```
areacode := DELETE (IN := phonenum, L := 9, P := 4);
```

Si phonenum es '567-445-9999' entonces areacode es '567' como es muestra en la siguiente figura.

Variables mostradas en la ventana Watch	
Símbolo	Valor
STL3.areacode	567
STL3.phonenum	567-445-9999

DELETEFILE

El procedimiento DELETEFILE borra un archivo existente. El DELETEFILE es uno de los ocho procedimientos que hacen operaciones de archivo. Note que estos procedimientos no están diseñados para ejecuciones E/S de altas velocidades, transferencia de datos de bloques grandes de información, o para aplicaciones de control.

El formato para el procedimiento DELETEFILE es el siguiente:

```
DELETEFILE (FCB:= <fcb>, IN:= <filename>);
```

Parámetros	Descripción
<fcb>	El nombre del bloque de control de archivo que maneja operaciones para este archivo. El nombre <fcb> no debe chocar con ningún otro nombre de variable local o global. Declare <fcb> en el Symbol Manager como un tipo de dato FILE.
<filename>	Contiene el nombre del archivo que será eliminado. Cualquier variable tipo de dato ANY.

La operación es la siguiente. DELETEFILE borra el archivo especificado por <filename>. El bloque de control de archivo maneja el acceso al archivo y condiciones de error. Todos los procedimientos de archivo que operan sobre el mismo archivo debe usar el mismo nombre File Control Block (bloque de control de archivo). Además, así como diseñó su código, debe usar las variables de control de archivos para manejar control de archivos. Para una descripción detallada de las variables de control de archivo, ver el procedimiento CLOSEFILE (Pág. 11-31).

Si un error ocurre, la variable error del archivo se ajusta a TRUE(verdadero), y un mensaje aparece en la ventana de salida y en el Wonderware Logger. Intentar borrar un archivo abierto genera un error (código de error 27). Para una lista de los códigos de error, ver la tabla Códigos de Error para Procedimientos con Archivos (Pág. 11-33).

Solamente puede hacer una operación de archivo por cada File Control Block a la vez. Note que las operaciones I/O de control de archivos tienen lugar asincrónicamente a la ejecución del programa.

El siguiente es un ejemplo del procedimiento DELETEFILE.

```
DELETEFILE (FCB:= datrpt IN:= "datareport");
```

El sistema borra el archivo llamado datereport.

EXP

La función EXP calcula el exponente natural de un valor.

El formato para la función EXP es el siguiente:

```
<result> := EXP (<number>);
```

Parámetros	Descripción
------------	-------------

<result>	Cualquier variable tipo dato ANY_REAL (REAL, LREAL).
----------	--

<number>	Cualquier variable o expresión que resuelve un tipo dato ANY_REAL (REAL, LREAL).
----------	--

La operación es la siguiente. EXP eleva e a la potencia del valor especificado por <number> y almacena el resultado en <result>.

El siguiente es un ejemplo de la función EXP:

```
newval := EXP (val);
```

Si val = 2.5 entonces newval = 12.182494.

EXPT

La función EXPT eleva un valor a la potencia especificada por el segundo valor.

El formato para la función EXPT es el siguiente:

```
<result> := EXPT (<number> , <power>);
```

Parámetros	Descripción
<result>	Una variable tipo dato ANY_NUM (SINT, INT, DINT, BYTE, WORD, DWRORD, REAL, O LREAL).
<number>	Cualquier número, variable, o expresión que resuelve un tipo dato ANY_NUM (SINT, INT, DINT, BYTE, WORD, DWORD, REAL, LREAL).
<power>	Cualquier número, variable, o expresión que resuelve un tipo dato ANY_NUM (SINT, INT, DINT, BYTE, WORD, DWORD, REAL, LREAL).

La operación es la siguiente. EXPT eleva el valor especificado por <number> a la potencia del valor especificado por <power> y almacena el resultado en <result>. Para el caso específico en el cual ambos <number> y <power> es igual a cero, <result> es igual a uno.

El siguiente es un ejemplo de la función EXPT

```
numexp := EXPT (vala, valb);
```

Si vala = 2.5, y valb = 5, entonces numexp = 97.656250.

FIND

La función FIND busca una cadena de caracteres dentro de otra cadena o string.

El formato para la función FIND es la siguiente:

```
<result> := FIND (IN1:= <stringa>, IN2:= <stringb>);
```

Parámetros	Descripción
<result>	Una variable tipo dato ANY_INT (SINT, INT, DINT, BYTE, WORD, DWORD).
<stringa>	Cualquier valor STRING, o variable o expresión que resuelva un tipo de dato STRING.
<stringb>	Cualquier valor STRING, o variable o expresión que resuelva un tipo de dato STRING.

La operación es la siguiente. Si <stringa> contiene <stringb> entonces FIND escribe la posición del carácter inicial en <result>. De otra manera, FIND escribe un cero en <result>.

El siguiente es un ejemplo de la función FIND.

```
position := FIND (IN1 := fullname, IN2 := midname);
```

Si fullname es 'ann marie williams' midname es 'marie', entonces la posición es 5 como se muestra en la siguiente figura.

Variables mostradas en la ventana Watch	
Símbolo	Valor
STL3.position	5
STL3.fullname	Ann marie willians
STL3.midname	marie

INSERT

La función INSERT inserta una cadena de caracteres dentro de otra cadena o string.

El formato para la función INSERT es el siguiente:

```
<result>:=INSERT( IN1:=<stringa>, IN2:=<stringb>, P:=<position>);
```

Parámetros	Descripción
<result>	Una variable (tipo dato STRING).
<stringa> <stringb>	Cualquier valor STRING, o variable o expresión que resuelva un tipo de dato STRING.
<position>	Cualquier número, variable, o expresión que resuelva un tipo de dato (SINT, INT, DINT, BYTE, WORD, DWORD).

La operación es la siguiente. INSERT ubica el string especificado por <stringb> dentro del string especificado por <stringa> en la posición especificada por <position> y almacena el resultado en <result>. Si el valor de P es negativo, o si la suma de las longitudes de los dos string es mayor que 2048, <result>. Se coloca en un string vacío. Si P es mayor que 2048 o que la longitud de <stringa>, los dos string son concatenados.

El siguiente es un ejemplo de la función INSERT.

```
fullname:=INSERT(IN1:=first_last, IN2:= middle, P:= position);
```

Si `first_last` es 'ann_brown', `middle` es 'marie_', y `position` es 4, entonces `fullname` = 'ann_marie_brown' como se muestra en el Ejemplo 1 de la siguiente figura.

Ejemplo 1:

Variables mostradas en la ventana Watch	
Símbolo	Valor
STL3.fullname	Ann_marie_brown
STL3.firstname	Ann_brown
STL3.middle	Marie_

Ejemplo 2:

Variables mostradas en la ventana Watch	
Símbolo	Valor
STL3.fullname	Marie_ann_brown
STL3.first_last	Ann_brown
STL3.middle	Marie_

Para insertar **<stringb>** antes del primer carácter en **<stringa>**, fije `P = 0`. En este caso, `fullname` = `marie_ann_brown`, como se muestra en el ejemplo 2 de la figura.

INT_TO_BCD

La función `INT_TO_BCD` convierte un valor integer a la equivalente representación Código Binario decimal (BCD) del valor.

El formato para la función `INT_TO_BCD` es la siguiente:

```
<result> := INT_TO_BCD (<number>);
```

Parámetros	Descripción
<result>	Una variable BCD que resuelve a un tipo de dato ANY_BIT (BYTE, WORD, DWORD).
<number>	Un número, variable o expresión que resuelve un tipo de dato ANY_INT (SINT, DINT, BYTE, WORD, DWORD).

La operación es la siguiente. `INT_TO_BCD` convierte el número entero de entrada a su representación BCD.

```
Bcd_out := INT_TO_BCD(int_in);
```


Si `int_in = 1234` (0100 1101 0010 en binario), entonces `bcd_out = 0001 0010 0011 0100`, el cual es la representación de 1234 como se muestra en la siguiente figura.

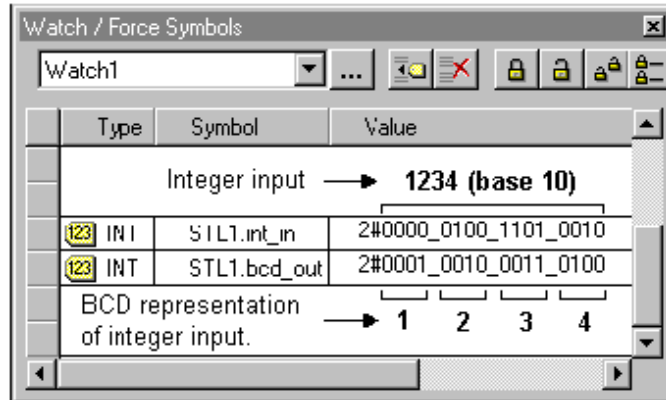


Figura 4. Representación BCD de un numero entero.

INT_TO_STRING

La función `INT_TO_STRING` convierte un integer a un string de caracteres ASCII.

El formato para la función `INT_TO_STRING` es:

```
<result> := INT_TO_STRING (<number>);
```

Parámetros	Descripción
------------	-------------

<result>	Una variable (tipo de dato STRING).
-----------------------	-------------------------------------

<number>	Cualquier número, variable o expresión que resuelve un tipo de dato ANY_INT (SINT, INT, DINT, BYTE, WORD, DWORD).
-----------------------	---

`INT_TO_STRING` convierte el numero entero dentro de una representación de caracteres ASCII del valor del numero entero.

```
intstring := INT_TO_STRING (num);
```

Si `num = 11`, entonces `intstring` contiene la cadena de caracteres ASCII '11'.

LEFT

La función LEFT crea un string de caracteres de un número especificado de los caracteres mas a la izquierda de otro string de caracteres.

```
<result> := LEFT (IN:= <stringa> , L:= <length>);
```

Parámetros	Descripción
<result>	Una variable (tipo de dato STRING).
<stringa>	Cualquier valor STRING, o variable o expresión que resuelva un tipo de dato STRING.
<length>;	Cualquier número, variable, o expresión que resuelve un tipo de dato ANY_INT (SINT, INT, DINT, BYTE, WORD, DWORD).

LEFT copia el número de caracteres especificado por **<length>**, empezando desde la izquierda de un string especificado por **<stringa>**, a la variable string especificada por **<result>**.

```
firstname := LEFT (IN:=fullname, L:= length);
```

Si fullname es 'john_willians', y su longitud es 4, entonces firstname es 'john' como se muestra en la figura de abajo.

Variables mostradas en la ventana Watch	
Símbolo	Valor
STL3.firstname	John
STL3.fullname	John_willians

LEN

La función LEN retorna la longitud de un string.

El formato para la función LEN es la siguiente:

```
<result> := LEN (<stringa>);
```

Parámetros	Descripción
<result>	Una variable tipo dato ANY_INT (SINT, INT, DINT, BYTE, WORD, DWORD).
<stringa>	Cualquier valor STRING, variable o expresión que resuelva a un tipo de dato STRING.

La operación es la siguiente. LEN copia el número de caracteres contenido en **<stringa>** en la variable integer especificada por **<result>**.

El siguiente es un ejemplo de la función LEN.

```
namelength := LEN (fullname);
```

Si fullname es 'john willians', entonces namelenght es 13 así como se muestra en la figura.

Variables mostradas en la ventana Watch	
Símbolo	Valor
STL3.namelenght	13
STL3.fullname	John willians

Nota: Si **<stringa>** contiene un carácter o caracteres nulos incluido, LEN devuelve el número de caracteres a el primer carácter nulo. Ver ARRAY_TO_STRING.

LN

La función LN calcula el logaritmo natural de un valor.

El formato para la función es el siguiente:

```
<result> := LN (<number>);
```

Parámetros	Descripción
<result>	Cualquier variable tipo dato ANY_REAL (REAL, LREAL).
<number>	Cualquier número, variable, o expresión que resuelva un tipo dato ANY_REAL (REAL, LREAL).

La operación es la siguiente. LN calcula el logaritmo natural del valor especificado por **<number>** y almacena el resultado en **<result>**.

El siguiente es un ejemplo de la función LN.

```
ln_num := LN (num);
```

Sí num = 674.3, entonces ln_num = 6.513675.

LOG

La función LOG calcula el logaritmo en base 10 de un valor.

El formato para la función LOG es la siguiente:

<result> := LOG (<number>);

Parámetros	Descripción
<result>	Cualquier variable tipo dato ANY_REAL (REAL, LREAL).
<number>	Cualquier número, variable, o expresión que resuelva un tipo dato ANY_REAL (REAL, LREAL).

La operación es la siguiente. LOG calcula el logaritmo de el valor especificado por **<number>** y almacena el resultado en **<result>**.

El siguiente es un ejemplo de la función LOG.

lognum := LOG (num);

Si num = 14.0, entonces lognum = 1.146128.

MAX

La función MAX determina el máximo de dos valores.

El formato para la función MAX es la siguiente:

<result> := MAX (<numbera> , <numberb>);

Parámetros	Descripción
<result>	Una variable tipo dato ANY excepto FILE, TMR, y definido por usuario.
<numbera>	Cualquier número, variables, o expresiones que resuelve a
<numberb>	un tipo de dato ANY excepto FILE, TMR, y usuario definido.

La operación es la siguiente. MAX compara el valor especificado por **<numbera>** al valor especificado por **<numberb>** y almacena el valor más grande en **<result>**.

El siguiente es un ejemplo de la función MAX.

result := MAX (vala, valb);

Si vala = 14, y valb = 5, entonces el result = 14.

MID

La función MID crea un string de caracteres de un número especificado de caracteres desde la mitad de otro string de caracteres.

El formato para la función MID es el siguiente:

```
<result>:=MID(IN:= <stringa>, L:= <length> , P:= <position>);
```

Parámetros	Descripción
<result>	Una variable (tipo dato STRING).
<stringa>	Cualquier valor STRING, o variable o expresión que resuelva un tipo de dato STRING.
<length>	Cualquier número, variable o expresión que resuelva un tipo dato ANY_INT (SINT, INT, BTE, WORD, DWORD)
<position>	

La operación es la siguiente. MID copia el número de caracteres en <stringa> especificado por <length> y empieza en la posición dada por <position>. MID almacena el resultado en <result>.

El siguiente es un ejemplo de la función MID.

```
midname := MID (IN:= fullname, L:= 5, P:= 5);
```

Si fullname es 'ann marie', entonces midname es 'marie' como se muestra en la siguiente figura.

Variables mostradas en la ventana Watch	
Símbolo	Valor
STL3.midname	Marie
STL3.fullname	Ann marie willians

MIN

La función MIN determina el mínimo de dos valores.

El formato para la función MIN es el siguiente:

```
<result> := MIN (<numbera> , <numberb>);
```

Parámetros	Descripción
<result>	Una variable ANY excepto FILE, TMR, o definida por usuario.
<numbera>	Cualquier número, variable o expresión que resuelve un tipo dato ANY excepto FILE, TMR, o definido por usuario.
<numberb>	

La operación es la siguiente. MIN compara el valor especificado por **<numbera>** con el valor especificado por **<numberb>** y almacena el valor mas pequeño en **<result>**.

El siguiente es un ejemplo de la función MIN.

```
result := MIN (vala, valb);
```

Si vala = 3.7415, y valb = 5.752, entonces result = 3.7415.


MSGWND

El procedimiento MSGWND visualiza un mensaje en la ventana de salida y en el Wonderware Logger.

El formato para el procedimiento MSGWND es el siguiente:

```
MSGWND (<stringa> , [<stringb>]);
```

Parámetros	Descripción
<stringa>	Cualquier valor string, o variable o expresión que resuelva un tipo de dato STRING. Si usa valores string, encierre el string entre comillas simples (' ').
<stringb>	

La operación es la siguiente. MSGWND muestra el contenido de **<stringa>** en la ventana de salida Output. El parámetro opcional **<stringb>** contiene el título, el cual aparece adelante del mensaje seguido por dos puntos. La pantalla del Runtime Engine también muestra el diamante de advertencia amarillo: .

El siguiente es un ejemplo de la función MSGWND.

```
MSGWND ('Open Valves', Phase);
```

La siguiente figura muestra la ventana Output cuando la fase (Phase) es 'Drain Phase'.

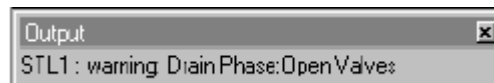


Figura 5. La ventana Output cuando la Fase es “Drain Phase”.

NEWFILE

Los procedimientos NEWFILE crean un nuevo archivo. El NEWFILE es uno de los ocho procedimientos que hacen operaciones de archivo. Note que estos procedimientos no están diseñados para ejecuciones E/S de alta velocidad, transferencia de datos de bloques grandes de información, o para aplicaciones de control.

El formato para el procedimiento NEWFILE es el siguiente:

```
NEWFILE (FCB:= <fcb>, FILE:= <filename>);
```

Parámetros	Descripción
<fcb>	Nombre del bloque de control del archivo que maneja operaciones para este archivo. Declare <fcb> en el Symbol Manager como un tipo de dato FILE.
<filename>	Contiene el nombre de el archivo a ser creado. Una variable tipo de dato ANY. La ruta predefinida es la misma para el software de ejecución (RTEngine.exe). Si necesita especificar una ruta diferente, no use un nombre UNC (Universal Naming Convention) Convención de Nombramiento Universal. UNC no es soportado.

La operación es la siguiente. NEWFILE crea el archivo especificado por <filename>, si un archivo con este mismo nombre no ha sido creado. Si un archivo con aquel nombre existe, NEWFILE lo sobrescribe. Después que el archivo ha sido abierto satisfactoriamente, el puntero de la posición del archivo interno se coloca en el principio de el archivo. El bloque de control de archivo maneja el acceso al archivo y las condiciones de error. Todos los procedimientos de archivo que operan sobre el mismo archivo debe usar el mismo nombre de bloque de control de archivo File Control Block. Además, cuando diseñe si código, debe usar variables de control de archivos para manejar control de archivos. Las tres variables de entrada de control de archivo son de particular importancia cuando los archivos están abiertos. Usted necesita verificar si sus valores definidos son o no son apropiados para su aplicación. Para una descripción detallada de las variables de control de archivos, ver el procedimiento CLOSEFILE (pag 11-31).

Si un error ocurre, la variable de error de archivo se pone en verdadero, y un mensaje aparece en la ventana de salida Output y en el Wonderware. Si un archivo que siendo manejado por el bloque de control de archivo ya esta abierto, NEWFILE cierra el primero y crea el segundo. Para una lista de todos los códigos de error, ver la tabla Códigos de Error para Procedimientos Tipo Archivo (pag 11-33). Puede hacer solamente una operación de archivo para cada bloque de control de archivo a la vez. Note que las operaciones E/S de control de archivo tienen lugar para programas ejecuciones asíncronas.

El siguiente es un ejemplo de el procedimiento NEWFILE.

```
NEWFILE (FCB:= "datrpt", FILE:= "datareport");
```

El sistema crea un Nuevo archivo llamado datareport.

OPENFILE

El procedimiento OPENFILE abre un archivo para operaciones, así como para lectura o escritura. El OPENFILE es uno de los ocho procedimientos que hacen operaciones de archivo. Note que estos procedimientos no están diseñados para ejecuciones E/S de alta velocidad, transferencia de datos de bloques grandes de información, o para aplicaciones de control.

El formato para el procedimiento OPENFILE es el siguiente:

```
OPENFILE (FCB:= <fcb>, FILE:= <filename>);
```

Parámetros	Descripción
<fcb>	Nombre del bloque de control del archivo que maneja operaciones para este archivo. Declare <fcb> en el Symbol Manager como un tipo de dato FILE.
<filename>	Contiene el nombre del archivo a ser creado. Una variable tipo de dato ANY. La ruta predefinida es la misma para el software de ejecución (RTEngine.exe). Si necesita especificar una ruta diferente, no use un nombre UNC (Universal Naming Convention) Convención de Nombramiento Universal. UNC no es soportado.

Las operaciones la siguiente. OPENFILE abre el archivo especificado por **<filename>**. Después que el archivo ha sido abierto satisfactoriamente, el puntero de la posición del archivo interno se coloca en el principio del archivo. El bloque de control de archivo maneja el acceso al archivo y las condiciones de error. Todos los procedimientos de archivo que operan sobre el mismo archivo deben usar el mismo nombre de bloque de control de archivo File Control Block. Además, cuando diseñe su código, debe usar variables de control de archivos para manejar control de archivos. Las tres variables de entrada de control de archivo son de particular importancia cuando los archivos están abiertos. Usted necesita verificar si sus valores definidos son o no son apropiados para su aplicación. Para una descripción detallada de las variables de control de archivos, ver el procedimiento CLOSEFILE (pag 11-31).

Si un error ocurre, la variable de error de archivo se pone en verdadero, y un mensaje aparece en la ventana de salida Output y en el Wonderware. Intentar abrir un archivo que no existe genera el código de error 18. Si un archivo que siendo manejado por el bloque de control de archivo ya esta abierto, OPENFILE cierra el primero y crea el segundo. Para una lista de todos los códigos de error, ver la tabla Códigos de Error para Procedimientos Tipo Archivo (pag 11-33).

Puede hacer solamente una operación de archivo para cada bloque de control de archivo a la vez. Note que las operaciones E/S de control de archivo tienen lugar para programas ejecuciones asíncronas.

El siguiente es un ejemplo del procedimiento OPENFILE.

```
OPENFILE (FCB:= "datrpt", FILE:= "data_report");
```

El sistema abre el archivo llamado called dat_report.

READFILE

El procedimiento READFILE lee datos desde un archivo. El READFILE es uno de los ocho procedimientos que hacen operaciones de archivo. Note que estos procedimientos no están diseñados para ejecuciones E/S de alta velocidad, transferencia de datos de bloques grandes de información, o para aplicaciones de control.

El formato para el procedimiento READFILE es el siguiente:

READFILE

```
FCB: =<fcb>, OUT: =<variable>, [ F: =<fieldsep> ], [ S: =<stringsep> ], [
T: =<eol> ] ) ;
```

Parámetros	Descripción
<fcb>	Nombre del bloque de control de archivo que maneja operaciones para este archivo. Declare <fcb> en el Symbol Manager como un tipo de dato FILE.
<variable>	Nombre de la variable que va a contener el dato leído desde el archivo. Una variable tipo dato ANY.
<fieldsep> ¹	Opcional. Caracter String usado para separar campos en el archivo. El valor predefinido el caracter de espacio. Si usted crea el archivo usando otra aplicación, así como un editor de texto en vez de el procedimiento WRITEFILE, asegúrese de usar separadores de campo entre valores.
<stringsep> ¹	Opcional. Caracter String usado para delimitar strings en el archivo. El valor predefinido es el carácter de comillas dobles ("). El string para delimitar no es requerido. Sin embargo, si usted crea el archivo usando otra aplicación, así como un editor de texto estructurado en vez de WRITEFILE, asegúrese de que el dato este estructurado para que READFILE lo lea correctamente.
<eol> ¹	Opcional. Es el carácter String usado para indicar el final de una línea en un archivo. El signo predefindo es una línea inferior. InControl trata al delimitador EOL como un separador de campo. Esto le permite al procedimiento READFILE leer desde mas de una línea a la ves. Note que como el separador de campo. El carácter EOL previene al comienzo un valor sobre una línea y lo continua sobre el próximo.

Elija los delimitadores muy cuidadosamente para evitar conflictos con caracteres String contenidos dentro del archivo. Ver WRITEFILE para mayor información (pag 11-76). La operación es la siguiente. READFILE lee el archivo asociado con el bloque de control **<fcb>** y almacena el dato en la variable especificada por **<variable>**. Después de cada operación de lectura el puntero interno de posición de archivo es adelantado por uno. El bloque de control de archivo maneja el acceso al archivo y condiciones de error. Todos los procedimientos de archivo que operan sobre el mismo archivo deben usar el mismo nombre File Control Block. Además, así como usted diseñe su código, debe usar las variables de control de archivo para manejar control de archivo. Para una descripción detallada de las variables de control de archivo, ver el procedimiento CLOSEFILE (pag 11-31).

Note que el procedimiento WRITEFILE escribe datos del tipo de dato TIME, DATE, DT, y TOD en formato STRING en vez de formato numérico. El procedimiento READFILE puede leer este formato de dato, siempre y cuando esté correctamente estructurado siguiendo las especificaciones IEC-61131.

Puede hacer una sola operación de archivo por cada bloque de control de archivo al tiempo. Las operaciones de E/S de control de archivo tienen lugar en la ejecución de programas asincrónicamente. Si un error ocurre, la variable de error de archivo es puesta en verdadero y un mensaje de error aparece en la ventana de salida Output. Para ver la Lista de Códigos de Errores ver la tabla de la pag 11-33.

El siguiente es un ejemplo del procedimiento READFILE.

```
READFILE(FCB:=datrpt,OUT:=datout,F:=",",S:="$",T:="$n");
```

El sistema accede al archivo referenciado por el bloque de control de archivo llamado datrpt y copia su contenido en la variable especificada por daout. La coma separa campos, las coma separa campos, los strings son encerrados dentro de comillas dobles y el final de la línea es indicada por \$n.

REAL_TO_DATE

La función REAL_TO_DATE convierte el contenido de un tipo de dato ANY_REAL a un tipo de dato ANY_DATE.

El formato para esta función es el siguiente:

```
<result> := REAL_TO_DATE (<number>);
```

Parámetros	Descripción
<number>	Cualquier variable, valor, o expresión que resuelve un tipo de dato ANY_REAL (REAL, LREAL).
<number>	Cualquier variable, valor, o expresión que resuelve un tipo de dato ANY_REAL (REAL, LREAL).

La operación es la siguiente. REAL_TO_DATE convierte el contenido de <number> a un tipo de dato ANY_DATE y almacena el resultado en <result>.

El siguiente es un ejemplo de la función REAL_TO_DATE.

```
datevar := REAL_TO_DATE (realvar);
```

Si realvar contiene 93467.3, entonces datevar equivale a DT#2155-11-25-07:12:00.000.

Nota: Para asegurar la precisión, use un dato tipo LREAL para el parámetro de entrada cuando aplique la conversión al tipo de dato DT.

REAL_TO_STRING

La función REAL_TO_STRING convierte un número real a un carácter string ASCII.

El formato para esta función es la siguiente:

```
<result> := REAL_TO_STRING (<number>);
```

Parámetros	Descripción
<result>	Una variable tipo dato STRING.
<number>	Cualquier valor, variable, o expresión que resuelve un tipo de dato ANY_REAL (REAL, LREAL).

La operación es la siguiente. REAL_TO_STRING convierte el número real **<number>** dentro de una representación de carácter ASCII de el valor de el número y almacena el resultado en **<result>**.

El siguiente es un ejemplo de la función REAL_TO_STRING.

```
realstring := REAL_TO_STRING (num);
```

Si num = 13.2288. entonces realstring contiene el caracter string ASCII '13.2288'.

REAL_TO_TIME

La función REAL_TO_TIME convierte el contenido de un tipo de dato REAL, el cual representa un tiempo en segundos, a un tipo de dato TIME.

El formato para la función REAL_TO_TIME es el siguiente:

```
<result> := REAL_TO_TIME (<number>);
```

Parámetros	Descripción
-------------------	--------------------

<result>	Una variable tipo de dato TIME.
-----------------------	---------------------------------

<number>	Cualquier valor, variable, o expresión que resuelva un tipo de dato ANY_REAL (REAL, LREAL).
-----------------------	---

La operación es la siguiente. REAL_TO_TIME convierte el contenido de **<number>** a un tipo de dato TIME y almacena el resultado como un valor en segundos en **<result>**.

El siguiente es un ejemplo de la función REAL_TO_TIME.

```
timevar := REAL_TO_TIME (realvar);
```

Si realvar contiene 62.98, entonces timevar equivale a T#1m2s980ms.

REPLACE

La función REPLACE reemplaza caracteres en una cadena de caracteres con un número especificado de caracteres de otra cadena de caracteres.

El formato para esta función es la siguiente:

```
<result>:=REPLACE( IN1:=<stringa> , IN2:=<stringb> , L:=<length> ,
:=<position> );
```

Parámetros	Descripción
<result>	Una variable (tipo dato STRING).
<stringa>	Cualquier valor STRING, o variables o expresiones que resuelvan un tipo de dato STRING.
<stringb>	resuelvan un tipo de dato STRING.
<length>	Cualquier, número, variable, o expresión que resuelve Un
<position>	tipo de dato ANY_INT (SINT, INT, DINT, BYTE, WORD, DWORD).

La operación es la siguiente. REPLACE reemplaza el número de caracteres especificado por <length> en <stringa> con caracteres de <stringb>. REPLACE reemplaza caracteres comenzando de la posición especificado por <position>, y almacena el resultado en <result>.

El siguiente es un ejemplo de la función.

```
newname := REPLACE(IN1:=fullname, IN2:= newmid, L:= 7, P:= 8);
```

Su fullname es 'Thomas Willians Sims', y newmid es 'Alan', entonces después de la operación. Newname es 'Thomas Alan Sims' así como se muestra en la siguiente figura.

Variables mostradas en la ventana Watch	
Símbolo	Valor
STL3.newname	Thomas Alan Sims
STL3.fullname	Thomas Willians Sims
STL3.NEWMD	Alan

REWINDFILE

El procedimiento REWINDFILE ubica el puntero de archivo interno al principio de un archivo. El REWINDFILE es uno de los ocho procedimientos que hacen operaciones de archivo. Note que estos procedimientos no están diseñados para ejecuciones E/S de alta velocidad, transferencia de bloques grandes de información, o para aplicaciones de control.

El formato para el procedimiento REWINDFILE es el siguiente:

```
REWINDFILE (<fcb>);
```

Parámetros	Descripción
<fcb>	Nombre de el bloque de control de archivos que manejan operaciones para estos archivos. Declare <fcb> en el Symbol Manager como un tipo de dato File.

La operación es la siguiente. REWINDFILE pone el puntero del archivo al principio del archivo asociado con el **<fcb>**. Esto permite al próximo operación de archivo empezar en el inicio del archivo. Esta operación se hace automáticamente cuando usted abre un archivo con OPENFILE o NEWFILE. Todas las operaciones de procedimientos de archivo sobre el mismo archivo debe usar el mismo nombre File Control Block. Además, así como diseña su código, debe usar las variables de control de archivo para manejar el control de archivo. Para una descripción detallada de las variables de control de archivo, ver el procedimiento CLOSEFILE (pag 11-31).

Si un error ocurre, las variables de error de archivo se colocan en verdadero, y un mensaje aparece en la ventana Output y en el Wonderware Logger. Para una lista de los códigos de error, ver la tabla Códigos de Error para Procedimientos Tipo Archivo (pag 11-33).

Puede hacer solamente una operación de archivo para cada File Control Block a la vez. Note que las operaciones E/S de control de archivo tienen lugar para programar ejecuciones asíncronas.

El siguiente es un ejemplo del procedimiento REWINDFILE.

```
REWINDFILE (FCB:= rpt);
```

El sistema rebobina o envuelve el archivo referenciado por el bloque de control de archivo llamado rpt.

RIGHT

La función RIGHT crea una cadena de caracteres desde un número especificado de caracteres ubicados a la derecha de otra cadena de caracteres.

El formato para la función RIGHT es la siguiente:

```
<result> := RIGHT (IN:= <stringa> , L:= <length>);
```

Parámetros	Descripción
<result>	Una variable (tipo de dato STRING).
<stringa>	Cualquier valor STRING, o variable o expresión que resuelva un tipo de dato STRING.
<length>	Cualquier número, variable, o expresión que resuelva un tipo de dato ANY_INT (SINT. INT, DINT, BYTE, WORD, DWORD).

La operación es la siguiente. RIGHT copia el número de caracteres especificado por <length> empezando desde el último a la derecha de un string especificado por <stringa> a la variable string especificado por <result>.

El siguiente es un ejemplo de la función RIGHT.

```
lastname := RIGHT (IN:=fullname, L:= length);
```

Si fullname es 'john willians', y length = 8, entonces lastname es 'willians' como se muestra en la siguiente figura.

Variables mostradas en la ventana Watch	
Símbolo	Valor
STL3.lastname	Willians
STL3.fullname	John willians

ROL

La función ROL rota los bits individuales de un valor un número especificado posiciones a la izquierda.

El formato para la función ROL es la siguiente:

```
<result> := ROL(IN:= <number>, N:= <shift>);
```

Parámetros	Descripción
<result>	Cualquier variable tipo dato ANY_BIT (BYTE, WORD, DWORD).
<number>	Cualquier número o variable que resuelva un tipo de dato ANY_BIT (BYTE, WORD, DWORD).
<shift>	Cualquier número, variable, o expresión, que resuelve un tipo de dato ANY_INT (SINT, INT, DINT, BYTE, WORD, DWORD).

La operación es la siguiente. ROL examina <number> en su forma binaria, mueve cada bit hacia la izquierda según el número de posiciones especificado por <shift>, y almacena el resultado en <result>. El bit más significativo se mueve hasta la posición del bit menos significativo.

El siguiente es un ejemplo de la función ROL.

```
num1rotl := ROL (IN:= num1, N:= num2);
```

Su num1 es un byte = 130 (10000010), y num2 = 1, entonces después que la operación de rotación ha finalizado, num1rotl = 5 (00000101) como se muestra en la siguiente figura.

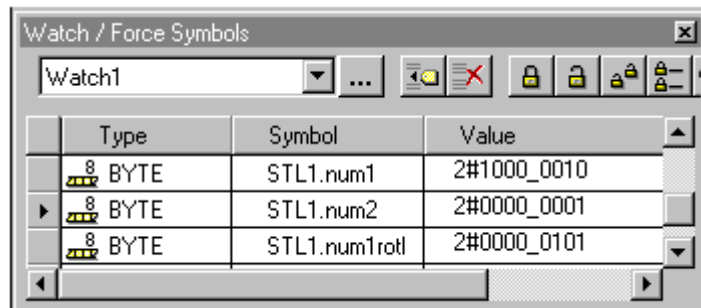


Figura 6. Ejecución de la orden ROL.

ROR

La función ROR rota bits particulares de un valor un número especificado de posiciones a la derecha.

El formato para la función ROR es la siguiente:

```
<result> := ROR(IN:= <number>, N:= <shift>);
```

Parámetros	Descripción
<result>	Cualquier variable tipo dato ANY_BIT (BYTE, WORD, DWORD).
<number>	Cualquier número o variable que resuelva un tipo de dato ANY_BIT (BYTE, WORD, DWORD).
<shift>	Cualquier número, variable, o expresión, que resuelve un tipo de dato ANY_INT (SINT, INT, DINT, BYTE, WORD, DWORD).

La operación es la siguiente. ROL examina <number> en su forma binaria, mueve cada bit hacia la derecha según el número de posiciones especificado por <shift>, y almacena el resultado en <result>. El bit menos significativo se mueve hasta la posición del bit más significativo.

El siguiente es un ejemplo de la función ROR.

```
num1rotr := ROR (IN:= num1, N:= num2);
```

Si num1 es un byte = 9 (00001001), y num2 = 1, entonces después que la operación de rotación finalice, num1rotr = 132 (10000100) como se muestra en la siguiente figura.

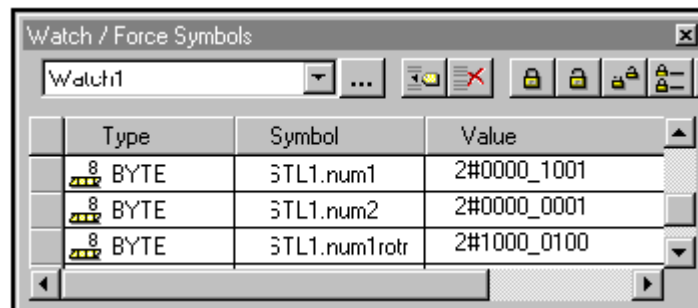


Figura 7. Ejecución de la orden ROR.

SHL

La función SHL mueve los bits individuales de un valor hacia la izquierda.

El formato para la función SHL es la siguiente:

<result> := SHL(**<number>**, **<shift>**);

Parámetros	Descripción
<result>	Cualquier variable tipo dato ANY_BIT (BYTE, WORD, DWORD).
<number>	Cualquier número o variable que resuelva un tipo de dato ANY_BIT (BYTE, WORD, DWORD).
<shift>	Cualquier número, variable, o expresión, que resuelve un tipo de dato ANY_INT (SINT, INT, DINT, BYTE, WORD, DWORD).

La operación es la siguiente. SHL examina **<number>** en su forma binaria y luego mueve cada bit a la izquierda de acuerdo al número de posiciones especificadas por **<shift>**. Los bits mas significativos se pierden después del cambio; los bits menos significativos, son remplazados por cero.

Si **<shift>** especifica un valor que es mayor que el número de bits en **<number>**, entonces SHL pone a **<result>** en cero.

El siguiente es un ejemplo de la función SHL.

num1shl := SHL (num1, num2);

Si num1 es un byte = 132 (10000100) y num2 = 2, entonces después que la operación de cambio ha terminado, num1shl = 13 (00010000) como se muestra en la siguiente figura.

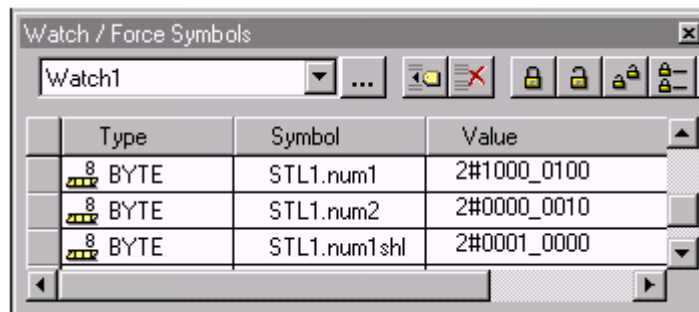


Figura 8. Representación de los bits de una variable.

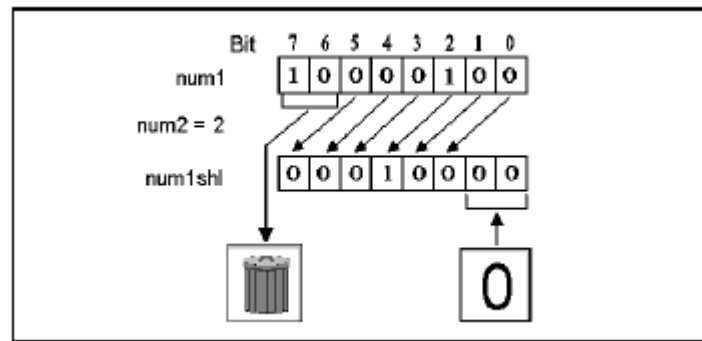


Figura 9. Ejecución de la orden SHL.

SHR

La función SHR mueve los bits individuales de un valor a la derecha.

El formato para la función SHR es la siguiente:

```
<result> := SHR(<number>, <shift>);
```

Parámetros	Descripción
<result>	Cualquier variable tipo dato ANY_BIT (BYTE, WORD, DWORD).
<number>	Cualquier número o variable que resuelva un tipo de dato ANY_BIT (BYTE, WORD, DWORD).
<shift>	Cualquier número, variable, o expresión, que resuelve un tipo de dato ANY_INT (SINT, INT, DINT, BYTE, WORD, DWORD).

La operación es la siguiente. SHL examina **<number>** en su forma binaria y luego mueve cada bit a la izquierda de acuerdo al número de posiciones especificadas por **<shift>**. Los bits más significativos se pierden después del cambio; los bits menos significativos, son remplazados por cero.

Si **<shift>** especifica un valor que es mayor que el número de bits en **<number>**, luego SHR pone a **<result>** en cero.

El siguiente es un ejemplo de la función SHR.

```
num1shr := SHR (num1, num2);
```

Si num1 es un byte = 162 (10100010) y num2 = 4, luego que la operación shift ha terminado, num1shr = 10 (00001010) así como se muestra en la siguiente figura.

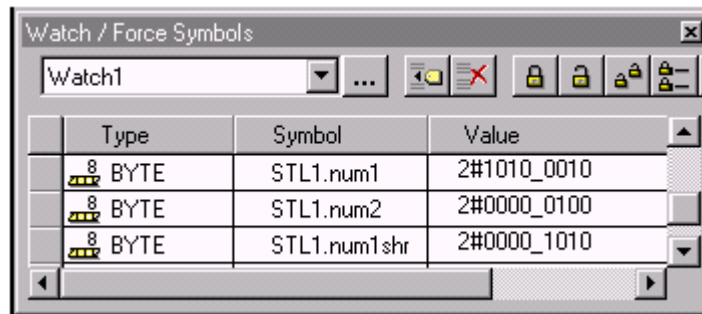


Figura 10. Representación de los bits de una variable.

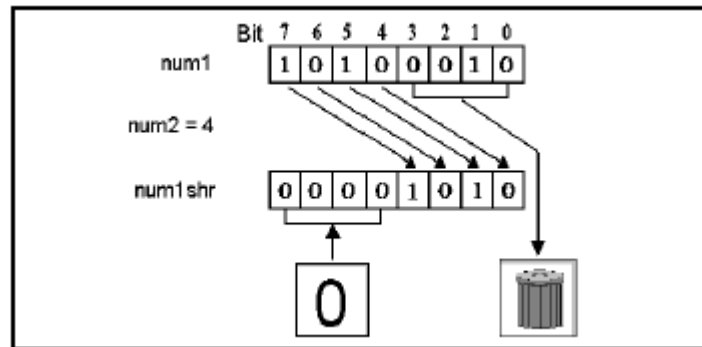


Figura 11. Ejecución de la orden SHR.

SIN

La función SIN calcula el seno de la entrada, el cual debe estar en radianes.

El formato para la función SIN es la siguiente:

```
<result> := SIN(<number>);
```

Parámetros	Descripción
<result>	Cualquier variable tipo dato ANY_BIT (BYTE, WORD, DWORD).
<number>	Cualquier número o variable que resuelva un tipo de dato ANY_REAL (REAL, LREAL).

El siguiente es un ejemplo de la función SIN.

```
sinangle := SIN(angle);
```

Si angle = 6.0, entonces sinangle = -0.27941550.

SQRT

La función SQRT calcula la raíz cuadrada de un valor.

El formato de la función SQRT es la siguiente:

```
<result> := SQRT(<number>);
```

Parámetros	Descripción
<result>	Cualquier variable tipo dato ANY_NUM (SINT, INT, DINT, BYTE, WORD, DWORD, REAL, LREAL).
<number>	Cualquier número, variable, o expresión que resuelva un tipo de dato ANY_NUM (SINT, INT, DINT, BYTE, WORD, DWORD, REAL, LREAL).

Si **<result>** es negativo, se pone igual a **<number>**.

El siguiente es un ejemplo de la función SQRT:

```
sqrtNum := SQRT(num);
```

Si num y sqrtNum son números enteros, entonces sqrtNum = 11 cuando num = 132. Si num y sqrtNum son números reales, entonces sqrtNum = 11.489125 cuando num = 132.0.

STRING_TO_ARRAY

El procedimiento STRING_TO_ARRAY toma la cadena de entrada ASCII y almacena el carácter de la cadena en una fila de byte.

El formato para el procedimiento STRING_TO_ARRAY es el siguiente:

```
STRING_TO_ARRAY(OUT := <array> , IN := <string>);
```

Parámetros	Descripción
<array>	un arreglo o fila de datos tipo byte.
<string>	una string, variable o expresión que resuelva a un tipo de dato string.

La operación es como sigue. STRING_TO_ARRAY convierte cada carácter de la cadena **<string>** a su valor decimal y almacena el resultado en **<array>** incluyendo el último elemento del arreglo, o hasta que el final de el string sea alcanzado.

El siguiente es un ejemplo del procedimiento STRING_TO_ARRAY.

```
STRING_TO_ARRAY (OUT:= byte_array, IN:= string1);
```

Si string1 contiene 'Sara', entonces el arreglo consiste de cinco bytes, y byte_array[0] = 83, byte_array[1] = 97, byte_array[2] = 114, byte_array[3] = 97, y byte_array[4] = 0.

STRING_TO_DATE

La función `STRING_TO_DATE` toma el string de entrada ASCII y almacena el carácter del string como un tipo de dato `DATE`.

El formato para la función `STRING_TO_DATE` es el siguiente:

```
<result> := STRING_TO_DATE (<string>);
```

Parámetros	Descripción
<result>	Una variable tipo dato <code>ANY_DATE</code> (<code>DT</code> , <code>DATE</code> , <code>TOD</code>).
<string>	Cualquier <code>STRING</code> , o variable o expresión que resuelva a un tipo de dato <code>STRING</code> . El contenido de <string> debe igualar el formato de el tipo de dato <code>DATE</code> , <code>DT</code> , o <code>TOD</code> .

La operación es como se sigue. `STRING_TO_DATE` convierte cada carácter en el string **<string>** y almacena el resultado en **<result>** conforme al tipo de dato `DATE`, descrito anteriormente.

Si el contenido de **<string>** no iguala el formato del tipo de dato `DATE`, **<result>** se pone en cero.

El siguiente es un ejemplo de la función `STRING_TO_DATE`.

```
datevar := STRING_TO_DATE (string1);
```

Si `string1` contiene 'D#06- 13-48', entonces `datevar` equivale `D#06-13-48`.

Si `string1` contiene '03 – 13 - 69', entonces `datevar` equivale a `DT#1899-12-30-00 : 00: 01.1001`.

STRING_TO_INT

La función `STRING_TO_INT` toma el string de entrada ASCII y almacena el carácter del string en un integer.

El formato para la función `STRING_TO_INT` es la siguiente:

```
<result> := STRING_TO_INT (<string>);
```

Parámetros	Descripción
<result>	Cualquier variable tipo dato <code>ANY_BIT</code> (<code>BYTE</code> , <code>WORD</code> , <code>DWORD</code>).
<string>	Cualquier <code>STRING</code> . O una variable o expresión que resuelva un tipo de dato <code>STRING</code> .

La operación es la siguiente. `STRING_TO_INT` convierte cada carácter numérico, hasta el primer carácter no-numérico, (si lo hay) de la cadena **<string>** en su valor decimal y lo muestra en **<result>**.

El siguiente es un ejemplo de la función `STRING_TO_INT`.

```
intvar := STRING_TO_INT (string1);
```

Si `string1` contiene '12345', entonces `intvar` equivale 12345.

Si `string1` contiene '-12', entonces `intvar` equivale -12.

Si `string1` contiene '123AB', entonces `intvar` equivale 123.

STRING_TO_TIME

La función `STRING_TO_TIME` toma la entrada string ASCII y almacena el carácter del string en un tipo de dato `TIME`.

El formato para la función `STRING_TO_TIME` es la siguiente:

```
<result> := STRING_TO_TIME (<string>);
```

Parámetros	Descripción
<result>	Una variable tipo dato <code>TIME</code> .
<string>	Un <code>STRING</code> o una variable o expresión que resuelva a un tipo de dato <code>STRING</code> . El contenido de <string> debe igualar el formato del tipo de dato <code>TIME</code> : <code>TIME</code> tiempo T t# seguido por una secuencia de uno o más números y unidades de tiempo específicas. Las unidades de tiempo específicas, rangos, y ejemplos de su uso están listados en la siguiente tabla.

Especificaciones	Rango	Ejemplo
D or d = Dias	0-1000000	T#1D2h: 1 dia y 2 horas
H or h = Horas	0-23	t#20H: 20 hours
M or m = Minutos	0-59	t#5m45s: 5 minutos y 45 segundos.
S or s = Segundos	0-59	t#26S200MS: 26 segundos y 200 milisegundos.
MS or ms = Milisegundos	0-999	T#45.325ms: 45.325 milisegundos

La operación es la siguiente. STRING_TO_TIME convierte cada carácter de la cadena **<string>** a su valor decimal y almacena el resultado en **<result>** de acuerdo al tipo de dato TIME, descrito anteriormente.

Si el contenido de **<string>** no es igual al formato del tipo de dato TIME, **<result>** se pone en cero.

El siguiente es un ejemplo de la función STRING_TO_TIME.

timevar := STRING_TO_TIME (string1);

Si string1 contiene 't#12D_5H_13M_5s', entonces timevar equivale a t#12D_5H_13M_5s.

Si string1 contiene '03-13-69', entonces timevar equivale a T#0ms.

TAN

La función TAN calcula la tangente de la entrada, el cual debe estar en radianes.

El formato para la función TAN es la siguiente:

<result> := TAN(**<number>**);

Parámetros	Descripción
<result>	Cualquier variable tipo dato ANY_REAL (REAL, LREAL).
<number>	Cualquier número o variable que resuelva un tipo de dato ANY_REAL (REAL, LREAL).

El siguiente es un ejemplo de la función TAN.

tanangle := TAN(angle);

Si angle = 6.0, entonces tanangle = -0.2910062.

TIME_TO_REAL

La función TIME_TO_REAL convierte el contenido del tipo dato TIME a un tipo dato REAL, el cual representa el valor del tiempo en segundos.

El formato para la función TIME_TO_REAL es el siguiente:

```
<result> := TIME_TO_REAL (<inputtime>);
```

Parámetros	Descripción
<result>	Una variable tipo dato ANY_REAL (REAL, LREAL.)
<inputtime>	Un valor, variable, o expresión que resuelve a un tipo de dato TIME. El contenido de <inputtime> debe igualar el formato del tipo de dato TIME: TIME tiempo T t# seguido por una secuencia de uno o más números y unidades de tiempo específicas. Las unidades de tiempo específicas, rangos, y ejemplos de su uso están listados en la siguiente tabla.

Especificaciones	Rango	Ejemplo
D o d = Dias.	0-1000000	T#1D2h: 1 días y 2 horas.
H o h = Horas.	0-23	t#20H: 20 horas.
M o m = Minutos	0-59	t#5m45s: 5 minutos y 45 segundos.
S o s = Segundos.	0-59	t#26S200MS: 26 segundos y 200 milisegundos.
MS o ms = Milisegundos.	0-999	T#45.325ms: 45.325 milisegundos.

La operación es la siguiente. TIME_TO_REAL calcula el valor en segundos representado por el contenido de <inputtime> y almacena el resultado como un tipo de dato REAL en <result>.

El siguiente es un ejemplo de la función TIME_TO_REAL.

```
realvar := TIME_TO_REAL (timevar);
```

Si timevar contiene T#5d6m49ms, entonces realvar equivale 432360.049.

TIME_TO_STRING

La función TIME_TO_STRING convierte el contenido del tipo de dato TIME a un tipo dato STRING.

El formato para la función TIME_TO_STRING es el siguiente:

```
<result> := TIME_TO_STRING (<inputtime>);
```

Parámetros	Descripción
------------	-------------

<result>	Una variable tipo STRING.
-----------------------	---------------------------

<inputtime>	Un valor, variable, o expresión que resuelve a un tipo de dato TIME. El contenido de <inputtime> debe igualar el formato del tipo de dato TIME: TIME tiempo T t# seguido por una secuencia de uno o más números y unidades de tiempo específicas. Las unidades de tiempo específicas, rangos, y ejemplos de su uso están listados en la siguiente tabla.
--------------------------	--

Especificaciones	Rango	Ejemplo
D o d = Dias.	0-1000000	T#1D2h: 1 días y 2 horas.
H o h = Horas.	0-23	t#20H: 20 horas.
M o m = Minutos	0-59	t#5m45s: 5 minutos y 45 segundos.
S o s = Segundos.	0-59	t#26S200MS: 26 segundos y 200 milisegundos.
MS o ms = Milisegundos.	0-999	T#45.325ms: 45.325 milisegundos.

La operación es la siguiente. TIME_TO_STRING convierte el contenido de **<inputtime>** a un carácter ASCII y los almacena en el string **<result>** usando el formato del tipo de dato TIME.

El siguiente es un ejemplo de la función TIME_TO_STRING.

```
stringvar := TIME_TO_STRING (timevar);
```

Si timevar contiene T#5d6m49ms, entonces stringvar equivale 'T#5d6m49ms'.

TRUNC

El TRUNC remueve uno o mas de los dígitos menos significantes de un tipo de dato REAL.

El formato para la función TRUNC es la siguiente:

```
<intnumber> := TRUNC(<realnumber>);
```

Parámetros	Descripción
<intnumber>	Cualquier variable tipo dato ANY_INT (SINT, INT, DINT, BYTE, WORD, DWORD).
<realnumber>	Cualquier número, variable, o expresión que resuelva un tipo de dato ANY_REAL (REAL, LREAL).

La operación es la siguiente. TRUNC remueve la parte fraccional de **<realnumber>** y almacena el valor integer del resultado en **<intnumber>**.

El siguiente es un ejemplo de la función TRUNC.

```
intresult := TRUNC(realinput);
```

Sí realinput = 56.789, entonces intresult = 56.

WRITEFILE

El procedimiento WRITEFILE escribe datos a un archivo. El WRITEFILE es uno de los ocho procedimientos que hacen operaciones de archivo. Note que estos procedimientos no están diseñados para ejecuciones E/S de alta velocidad, transferencia de datos de bloques grandes e información, o para aplicaciones de control.

El formato para el procedimiento WRITEFILE es el siguiente:

```
WRITEFILE(FCB: =<fcb>, IN: =<variable>, [F: =<fieldsep>],
[S: =<stringsep>], [T: = <eol>]);
```

Parámetro	Descripción
<fcb>	Nombre del bloque de control de archivo que maneja operaciones para este archivo. Declare <fcb> en el Symbol Manager como un tipo de dato File.
<variable>	Nombre de la variable que contiene el dato escrito al archivo. Una variable tipo dato ANY.
<fieldsep> ¹	Opcional. Caracter string usado para separar campos en el archivo. El carácter predefinido es el espacio. Mientras el archivo escribe operaciones, InControl ubica un campo separador entre valores.
<stringsep> ¹	Opcional. El carácter string usado para delimitar strings dentro del archivo. El carácter predefinido es la doble comilla. WRITEFILE ubica el string delimitador sobre cada extremo de un string previo a escritura para el archivo.
<eol> ¹	Opcional. El carácter string usado para indicar el final de una línea de archivo. El carácter predefinido es el linefeed (␣). Una línea puede ser hasta de 1K de longitud. Usted no puede escribir mas de 1K de dato para cada ejecución del WRITEFILE. Cuando una operación escrita es finalizada y todos los valores son escritos al archivo, InControl ata un delimitador EOL.

La operación es la siguiente. WRITEFILE lee el dato en la variable especificada por **<variable>** y lo escribe en el archivo asociado con el bloque de control **<fcb>**. Después de cada operación, la posición del puntero del archivo interno es avanzada en uno. El bloque de control de archivo maneja el acceso al archivo y condiciones de error. Todos los procedimientos de archivo que operan sobre el mismo archivo debe usar el mismo nombre de bloque de control del archivo (File Control Block). Además, así como usted diseñe su código, debe usar las variables de control de archivo para manejar control de archivo. Para una descripción detallada de las variables de control de archivo, ver el procedimiento CLOSEFILE (pag -)

Note que el dato datos tipos TIME, DATE, DT, y TOD es escrito a la salida en formato STRING en vez de un formato numérico. El procedimiento READFILE puede leer este formato del dato, con tal que sea estructurado correctamente seguido por la especificación IEC – 61131.

Si usted usa WRITEFILE para añadir datos a el archivo (el archivo fue abierto con el File Control Append Input Variable puesto en TRUE), entonces las operaciones leídas subsecuentes sobre el archivo generará un End o error File (code 31). Asegúrese reenrollar el archivo procurando leerlo de nuevo.

Si un error ocurre, la variable de error de archivo se pone en verdadero, y un mensaje aparece en la ventana de saluda Output y el Wonderware Logger. Para una lista de todos los códigos de error, ver la tabla de códigos de error para procedimientos con archivos (pag 11-33).

El siguiente es un ejemplo del procedimiento WRITEFILE. También note la costumbre de los procedimientos NEWFILE, REWINDFILE, READFILE y CLOSEFILE.

```
NEWFILE (rpt, "my_file");  
WRITEFILE (rpt, data_in);  
REWINDFILE (rpt);  
READFILE (FCB:= rpt, OUT:= data_out);  
CLOSEFILE (rpt);
```

1. **NEWFILE** : crea un archivo llamado my_file. La localización predefinida para my_file es el mismo directorio donde RTEngine.exe esta localizado.
2. **WRITEFILE**: copia el contenido de la variable STRING data_in en my_file.
3. **REWINDFILE**: ajusta el archivo del puntero al inicio de my_file.
4. **READFILE**: Copia el contenido de my_file en la variable STRING data_out.
5. **CLOSEFILE**: cierra my_file.

Precaución: Cada operación de escritura de archivo imprime una o más líneas de texto ASCII al archivo. Escribiendo datos al archivo en múltiples tiempos puede potencialmente corromper el contenido del archivo, con el riesgo de perder información. Esto puede ocurrir cuando usted reenrolla (rewind) y dentro de la siguiente operación de escritura, la longitud de la nueva línea de información es diferente que la de la línea sobrescrita. Cuando diseñe operaciones de escritura de archivo tenga en cuenta que las nuevas líneas de datos sean de igual longitud que las líneas que han sido reemplazadas. De otra manera, borre un archivo y reemplácelo, si es necesario.

USO DEL EDITOR DE TEXTO ESTRUCTURADO

Este capítulo nos muestra el funcionamiento del editor de Texto Estructurado y como poder usarlo para crear nuevos programas y para adicionar códigos de Texto Estructurado. Para detalles acerca del Lenguaje de Texto Estructurado, referirse al capítulo “Lenguaje de Texto Estructurado”.



CONTENIDO

- **CREACIÓN DE UN PROGRAMA STL**
- **HERRAMIENTAS DE TEXTO ESTRUCTURADO**
- **INGRESO DEL CÓDIGO DE PROGRAMA**

This product is licensed to:

Company:	N/A
SN:	N/A
Expires:	N/A
Locked To:	N/A

7.1.0.1337

CREACIÓN DE UN PROGRAMA STL

Después de iniciar el InControl, usted puede crear un nuevo programa de texto estructurado o editar uno existente.

➤ **Para crear un nuevo programa de Texto Estructurado:**

1. Sobre el menú **File**, haga clic en **New**. Aparecerá el menú de tipos de programas soportados por InControl.
2. Seleccione **Structured Text**, escoja el tipo de programa (Programa, Bloque de Función, Función) y haga clic en **Aceptar**. Aparecerá el cuadro de dialogo **Save As**.
3. Escoja un nombre (Hasta 31 caracteres) y el directorio (Proyecto) para el programa y haga clic en **Save**.



Figura 1. Presentación y definición del programa

4. Para ingresar el código del programa, ver “**INGRESO DEL CÓDIGO DEL PROGRAMA**” (Pagina 12-7).

Nota: Todas las **POUs** son insertadas debajo de la carpeta **Programs** de la ventana de proyectos. Usted debe mover las funciones a la carpeta **Functions**, los tipos de bloque se funciones a la carpeta **Function Block**, y los macros a la carpeta **Macros** para la correcta compilación del proyecto.

Si usted abre un proyecto desarrollado bajo **InControl 7.0**, usted tiene la opción de convertir los archivos a un proyecto en **InControl 7.1**. Cualquier macro en un proyecto aparecerá en la carpeta **Programs** después de la conversión. Usted puede mover estos macros a la carpeta **Macros**, pero esto no es necesario para compilar el proyecto.

➤ **Para editar un programa de Texto Estructurado existente:**

1. Si la ventana de proyectos no esta abierta, haga clic en **Project** en el menú **View**. Aparecerá la ventana de proyectos.
2. Haga doble clic en el nombre del programa ha editar.

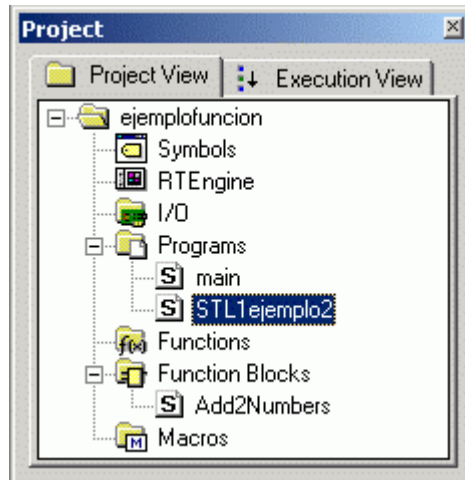


Figura 2. Presentación del programa en la Project View

Se abre el editor de **Texto Estructurado**, mostrando el programa seleccionado.

Usted también puede hacer clic en **Open** en el menú **File** para abrir un programa existente para editarlo.

Usted puede hacer clic en **Files Into Project** en el menú **Insert** para adicionar cualquier **POU** (Programa, Función, Bloque de Función, etc.) al proyecto. En la siguiente figura, el programa **STL1ejemplo**, que se muestra en el cuadro de dialogo **Insert Files Into Project**, es seleccionado y puede ser adicionado a **PIDApp**. Note que el archivo propiamente dicho no es copiado o movido cuando es adicionado a otro proyecto.

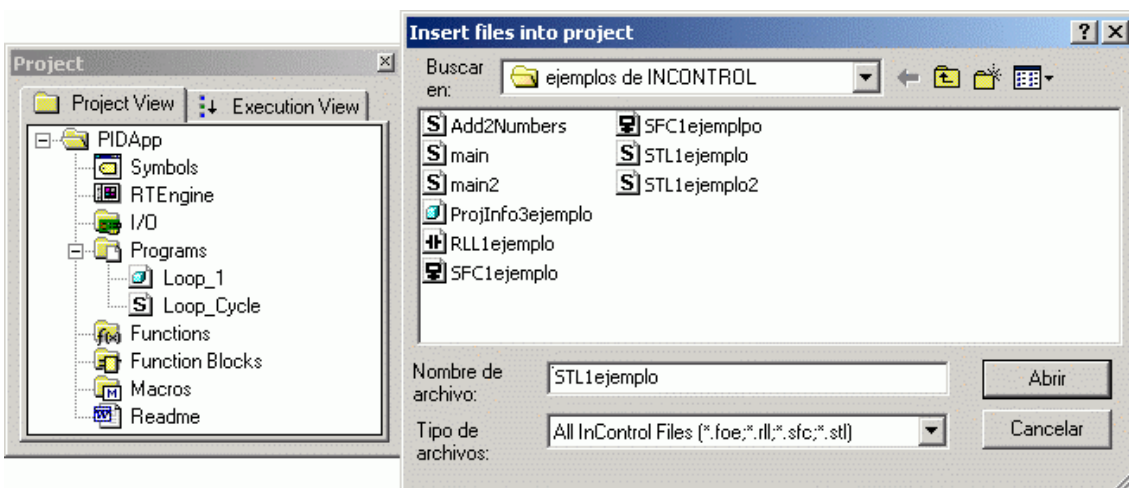


Figura 3. Adición de POU a un proyecto

HERRAMIENTAS DE TEXTO ESTRUCTURADO

Esta sección describe la barra de herramientas de Texto Estructurado y muestra algunos consejos útiles para la edición de programas.

Las herramientas de Texto Estructurado y la barra de Menú

La barra de Texto Estructurado muestra las herramientas usadas para crear programas de Texto Estructurado. Estas herramientas proveen un acceso directo para ingresar elementos de programa. Usted también puede escribir directamente el código de Texto Estructurado.



Figura 4. La barra de herramientas de Texto Estructurado

Iconos de la Barra de Herramientas de Texto Estructurado

Icono	Función	Opción de la Herramienta
	Pulse para insertar de ordenes preformateadas.	If, If Else, If Elself Else, Else, Elself, EndIf For, For (no wait) Repeat Until, Repeat Until (no wait) While Do, While Do (no wait) Case, Case Item Assignment Exit Include Scan Label Break
	Pulse para insertar funciones.	Ver “ PROCEDIMIENTOS Y FUNCIONES DE TEXTO ESTRUCTURADO ” en el capítulo “ LENGUAJE DE TEXTO ESTRUCTURADO ” para un listado de las funciones.
	Pulse para insertar operadores matemáticos.	Suma + Resta, Negación - Multiplicación * División / Modulación MOD Exponenciación ** Asignación := Menor que < Menor o igual que ≤ Igual = Desigualdad <> Mayor o igual que ≥ Mayor que >
	Pulse para insertar operadores de comparación.	
	Pulse para insertar operadores booleanos.	Operador booleano AND Operador booleano OR Operador booleano XOR OR exclusiva Complemento NOT
	Pulse para insertar un símbolo.	N/A
	Pulse para insertar un comentario.	N/A

Consejos de edición

Estos consejos pueden ayudarlo a editar sus programas.

- Usted también puede ingresar el código de Texto Estructurado seleccionando desde el menú **Insert**, el cual se muestra en la siguiente figura.

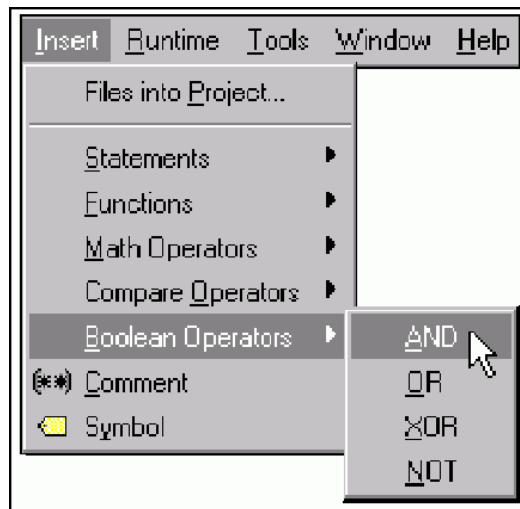


Figura 5. Ingresando código desde el menú Insert

- Use el menú **View** para mostrar los objetos que usted necesita ver durante una sesión de edición. Por ejemplo, si usted prefiere adicionar elementos de programa desde la barra de **Menú**, en lugar de la barra de herramientas **STL**, usted puede ocultar la barra de herramientas **STL**.

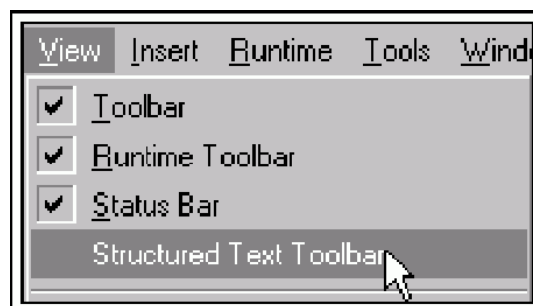


Figura 6. Desactivando la barra STL

- Durante una sesión de edición, usted puede hacer clic derecho para mostrar rápidamente algunas de las opciones de edición que aparecen en la barra de menú.

INGRESO DEL CÓDIGO DEL PROGRAMA

Usted puede ingresar el código de Texto Estructurado dentro de la ventana de edición por cualquiera de estos métodos.

- Escriba líneas de código directamente dentro de la ventana de edición.
- Pulse la selección en la barra de herramientas de Texto Estructurado.
- Pulse en la barra de menú **Insert** y entonces pulse la selección.

Para información detallada sobre las ordenes de Texto Estructurado, funciones, operadores, etc., ver “**ELEMENTOS DE TEXTO ESTRUCTURADO**” en el capítulo “**LENGUAJE DE TEXTO ESTRUCTURADO**”. Para una descripción de las herramientas útiles de edición, tal como Find, Replace, Go To, Set Bookmark, etc., ver el capítulo “**AMBIENTE INCONTROL**”.

Cuando usted selecciona un bloque de código y pulsa en **Comment** en el menú **Insert**, el código es señalado como un comentario.

Para usar símbolos en un programa, usted puede escoger cualquiera de estos métodos:

- Si un símbolo no ha sido definido en el **Symbol Manager**, usted puede ingresar un nuevo símbolo en el código y hacer doble clic en el nombre del símbolo. Esto abre el cuadro de dialogo **Symbol Properties**, y usted puede entonces ingresar los datos de configuración para el símbolo. El símbolo es automáticamente adicionado al **Symbol Manager**.
- Usted puede hacer doble clic en cualquier parte dentro del código de Texto Estructurado y abrir el **Symbol Manager**.

Si usted hace doble clic en el nombre de un símbolo aparecerá el **Symbol Manager**, el símbolo es insertado en la localización del cursor en el programa. Si usted hace doble clic en un método **FOE**, la sintaxis completa para la función de llamada del **FOE** es insertada en la localización del cursor en el programa.

- Usted puede hacer doble clic en cualquier nombre de un símbolo que aparezca en el programa y abrir el **Symbol Manager**. Si el símbolo ya ha sido definido, abra el **Symbol Manager** en el nivel correcto de alcance con el nombre del símbolo seleccionado.

Cuando usted valide un programa, las líneas con errores serán marcadas como se muestra en la siguiente figura.

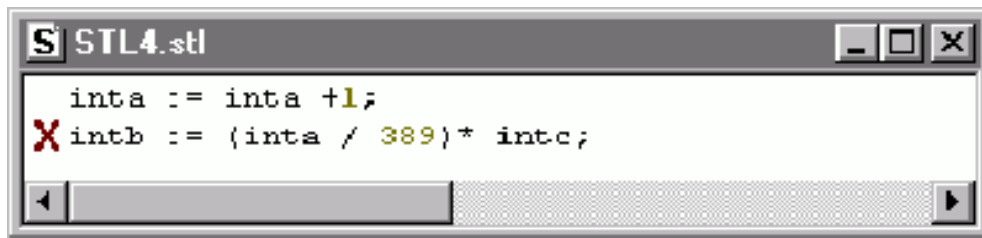


Figura 7. Marcación de errores en una línea de código

Usted puede revisar la ventana **Output**(Salida) para los mensajes de error que pueden ayudarlo a solucionar problemas del programa.

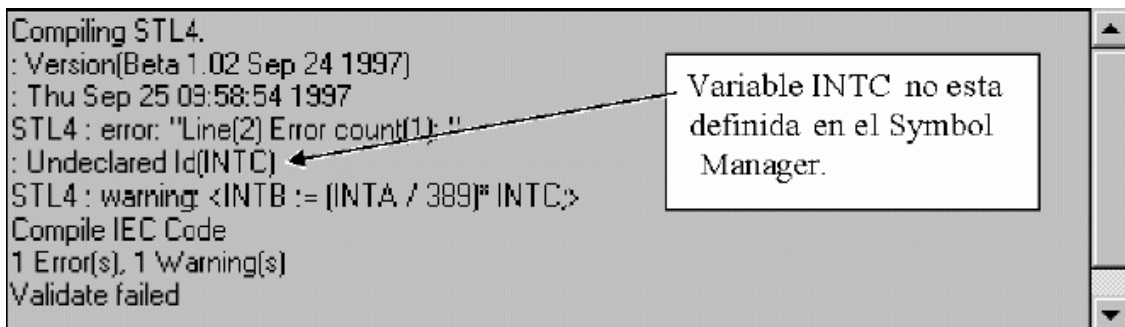


Figura 8. Mensajes de error en la ventana de salida

Nota: El editor de Texto Estructurado abre automáticamente desde un Paso SFC cuando usted hace doble clic sobre el Paso.

USO DEL EDITOR DE OBJETOS DE FÁBRICA INCONTROL

Este capítulo lo introduce en el editor de Objetos de Fábrica (FOE) de InControl y le dice cómo usarlo para adicionar controles ActiveX a un proyecto InControl.



CONTENIDO

- DEFINICIÓN DE OBJETOS FABRICA.
- INTEGRACIÓN DE CONTROLES ACTIVE X EN INCONTROL.
- CONFIGURACIÓN DE OBJETOS FABRICA.
- USANDO LA BARRA DE HERRAMIENTAS Y DE MENÚ.
- EJECUCIÓN DE OBJETOS FABRICA INCONTROL.
- USO DEL FOE PID DE WONDERWARE.
- SÍMBOLOS Y MÉTODOS PID.
- TEORÍA DE OPERACIÓN PID.
- USO DEL FOE DE ALARMAS ANÁLOGAS DE WONDERWARE.
- MÉTODOS Y SÍMBOLOS DE ALARMAS ANÁLOGAS.
- USO DEL FOE DE INFORMACIÓN DE PROYECTO.

This product is licensed to:
Company: N/A
SN: N/A
Expires: N/A
Locked To: N/A
7.1.0.1337

DEFINICIÓN DE OBJETOS DE FABRICA

InControl es compatible con las especificaciones del Servidor ActiveX. El FOE de InControl es un contenedor ActiveX, el cual lo habilita para usar controles dentro de un proyecto InControl.

Un control ActiveX debe ser instalado dentro de InControl antes que pueda configurarlo y correrlo. Después de la instalación, esta referido como un factory Object (FOE) de InControl. Tal como otros programas InControl, un FOE puede correr independientemente. También puede llamarlo para ejecución desde otro programa.

La actual descarga de InControl incluye algunos FOEs. Los siguientes FOEs estan descritos en este manual:

- Use el PID InControl FOE para manejar funciones de lazo PID.

Para mas información acerca de PID FOE. Ver "USANDO EL FOE PID DE WONDERWARE" (page -).

- Use el Analog Alarm (alarmas análogas) FOE de InControl para monitorear una señal de entrada análoga para condiciones de alarma.

Información acerca de su configuración, VER "USANDO ALARMAS ANÁLOGAS FOE DE WONDERWARE" (page -).

- Use el Project Information FOE para enviar proyectos e información de nodos a otras aplicaciones, así como InTouch, o para ejecutar varios comandos runtime o para limpiar fallas.

Para mas información acerca del Project Information FOE. VER “USANDO EL PROJECT INFORMATION FOE”. Para información adicional de acerca de los objetos de fábrica Wonderware, contacte a su distribuidor.

Para instalar un FOE en InControl, siga estos procedimientos generales.

Copie el FOE a la unidad de hardware InControl.

Los FOEs son instalados cuando usted instala InControl.

Instale el FOE en InControl.

Adicione una instancia de el FOE a el proyecto.

Configure el FOE

Note que Los FOEs de InControl siempre verifican la licencia Wonderware apropiada antes que sean ejecutados. Tipicamente, controles ActiveX estan autorizados y no cargan dentro de InControl si la autorización esta perdida.

INTEGRACIÓN DE CONTROLES ACTIVEX EN INCONTROL

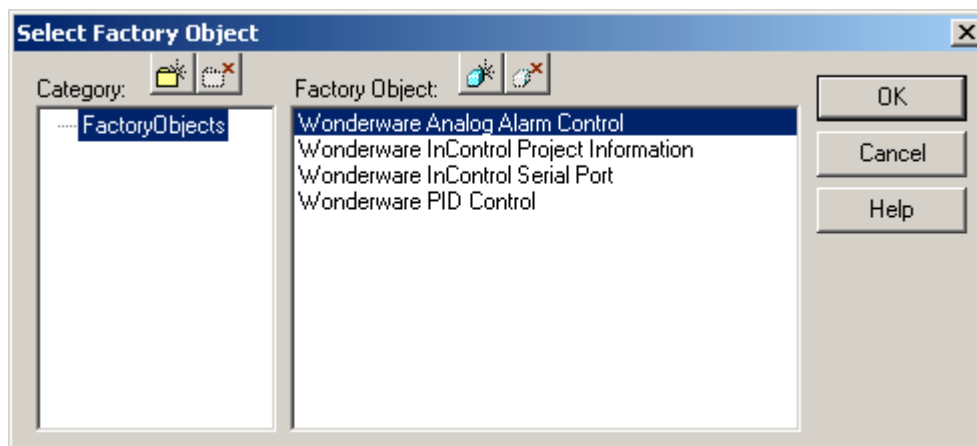
Uno o mas controles ActiveX pueden ser presentados sobre su unidad de hardware. Note que no todos ellos son apropiados para correr dentro de un proyecto InControl.

Instalando Controles ActiveX

Debe instalar el control ActiveX antes que pueda adicionarlo a la ventana de proyectos. Los FOEs son instalados automáticamente cuando instala InControl.

➤ Para instalar un Control ActiveX:

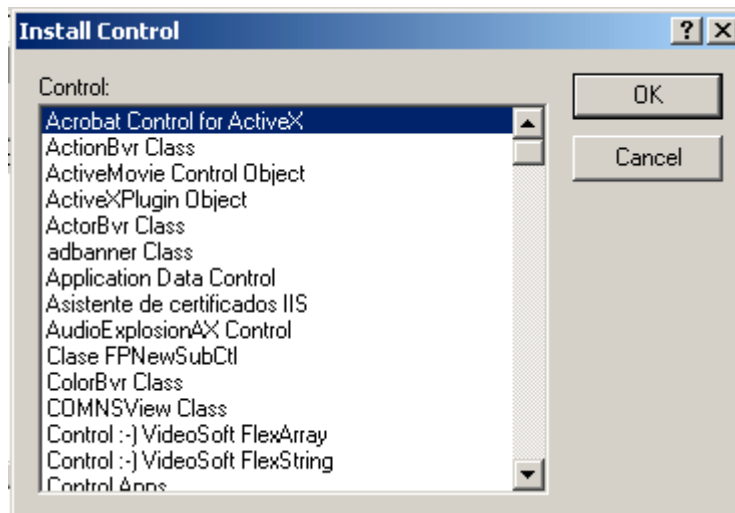
1. Copie el control ActiveX en la unidad de hardware de InControl.
2. Sobre el menú **File**, pulse **New**. La nueva el cuadro de diálogo, el cual muestra tipos de programas soportados por InControl, aparece.
3. Seleccione los Objetos de Fabrica y pulse OK. El cuadro de diálogo **Select Factory Object** aparece. Note los FOEs que ya están instalados.



- Opcional. Si quiere agrupar FOEs en categorías, elabore la categoría (“Organizando FOEs”) y pulse el nombre de la categoría antes de la instalación.
- Pulse **Install Factory Object** sobre la barra de herramientas.



El cuadro de diálogo **Install Control** aparece.



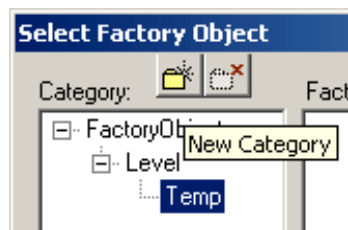
- Seleccione el control ActiveX y pulse OK. El cuadro de diálogo **Select Control Type** aparece. Si el control ActiveX está sobre su sistema pero no aparece en la muestra, puede hacer un cambio en el registro del sistema. Ver “Cambiando botones de registro del sistema” en el capítulo “Administrador de Sistemas de InControl”.
- Seleccione el tipo de Control y pulse OK. El objeto es instalado dentro de InControl y aparece en la muestra Factory Object.

Si pulsa OK para cerrar el cuadro de dialogo, se le sugiere entrar un nombre de archivo y gravar. Esto añade el FOE a el proyecto, también esta descrito en “Adding FOEs a el proyecto”. Si quiere instalar FOEs adicionales repita los pasos 4 – 6 antes de cerrar el cuadro de dialogo.

Organizando FOEs

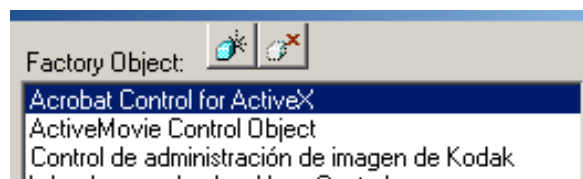
Usted puede organizar FOEs debajo de una o mas categorías y puede desinstalar FOEs que no necesite.

- Para crear una nueva categoría:
 1. Pulse una categoría debajo del cual quiere crear uno nuevo, y pulse **New category** sobre la barra de herramientas.



2. Cuando esta cuadro **New Category** aparece, entre el nombre y pulse **OK**. La nueva categoría aparece en el campo Category.

- Para instalar desinstalar un FOE:
 1. Pulse el nombre FOE.
 2. Pulse **Uninstall Factory Object** sobre la barra de herramientas.



Adicionando FOEs a un Proyecto

Después de instalar un control ActiveX como un FOE InControl, puede añadir uno o mas instancias de el a un proyecto.

➤ Para adicionar un FOE a un Proyecto:

1. Sobre el menú **File** pulse **New**.

El cuadro de diálogo **New**, el cual muestra tipos de programas soportados por InControl, aparece.

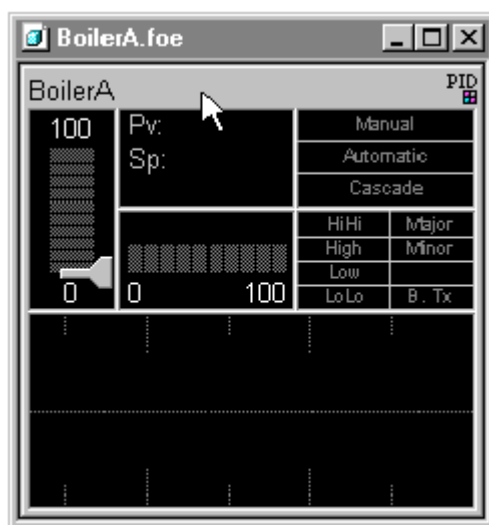
2. Seleccione **Factory Object** y pulse **OK**. El cuadro de diálogo **Select Factory Object** aparece.
3. Seleccione el FOE y pulse **OK**.
4. Cuando el cuadro de diálogo **Save as**, entre un nombre (hasta 31 caracteres) y pulse **Save**. El FOE es adicionado a el proyecto.

CONFIGURACIÓN DE OBJETOS FABRICA.

Después de adicionar un FOE a un proyecto, puede abrirlo para configurarlo.

➤ Para abrir y editar un FOE existente:

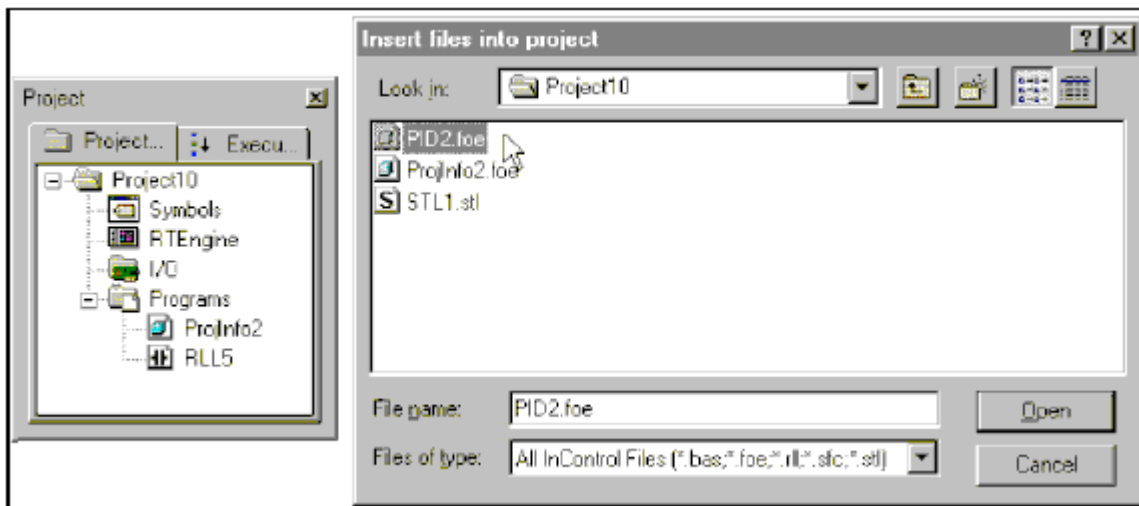
1. Si la ventana de proyectos no se abre, pulse **Project** en el menú **View** para abrirlo.
2. Pulse dos veces el nombre de el FOE. La ventana FOE se abre.
3. Para configurar el FOE, pulse dos veces dentro de la ventana FOE.



4. El cuadro de diálogo de Configuración aparece. Para mas información acerca de un third – party FOE, consulte el documento específico para este objeto. El PID FOE esta descrito en “Usando la Alarma Análoga FOE de Wonderware ”(Page -) y el Project Information FOE esta descrito en “Usando el Project information FOE”. (-).

Para mas información acerca de la Interface de Puerto Serial FOE, ver la guia de Usuario de Interface de Puerto serial de InControl Wonderware.

También puede pulsar **Open** en el menu **File** para abrir un programa existente para edición. Cuando el cuadro de diálogo **Open** aparece, seleccione el programa para abrir. Si un programa no es parte de un proyecto activo, puede adicionarlo. Puede pulsar **Files into Project** en el menú **Insert** para adicionar cualquier POU (programa, funciones, bloques de funciones, etc) para un proyecto. En la siguiente figura, el programa PID2, muestra el cuadro de diálogo Insert files Project, esta seleccionado y puede ser añadido a el Project10.



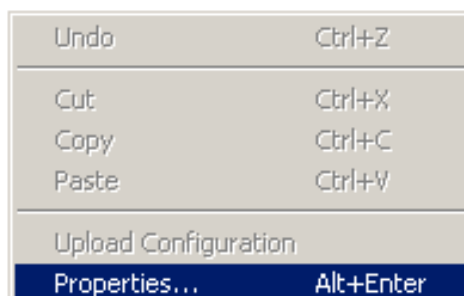
USANDO LA BARRA DE HERRAMIENTAS Y DE MENÚ

Para configurar un FOE, edite sus propiedades. Puede abrir el cuadro de diálogo para la propiedad por varios métodos.

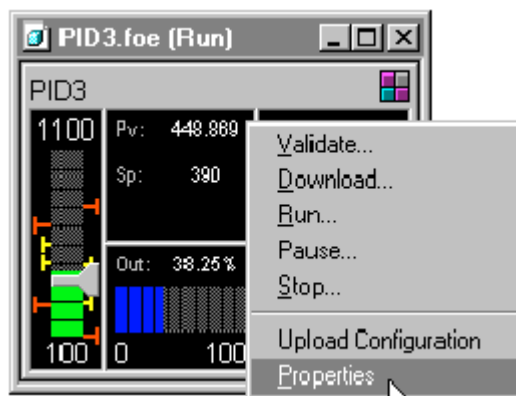
- Mostrar el cuadro de diálogo para las propiedades:
 - Pulse el FOE y luego pulse **Properties** sobre la barra de herramientas **Factory Object**.



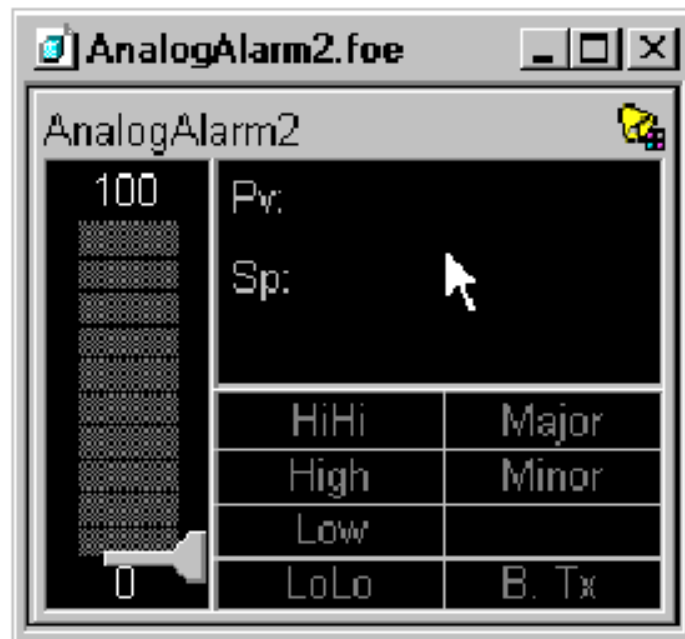
- Pulse el FOE y luego pulse **Properties** sobre el menú **Edit**.



- Pulse el botón izquierdo del mouse en el FOE y luego pulse **Propertier**.



Haga doble click en FOE. Note que si el FOE esta corriendo y usted está viendo su animación, no puede acceder a las propiedades haciendo doble – click en el FOE. Use uno de los otros métodos para acceder las propiedades.



EJECUCIÓN DE OBJETOS FABRICA INCONTROL

Siga las líneas de guía en esta sección para el uso de FOEs en un proyecto.

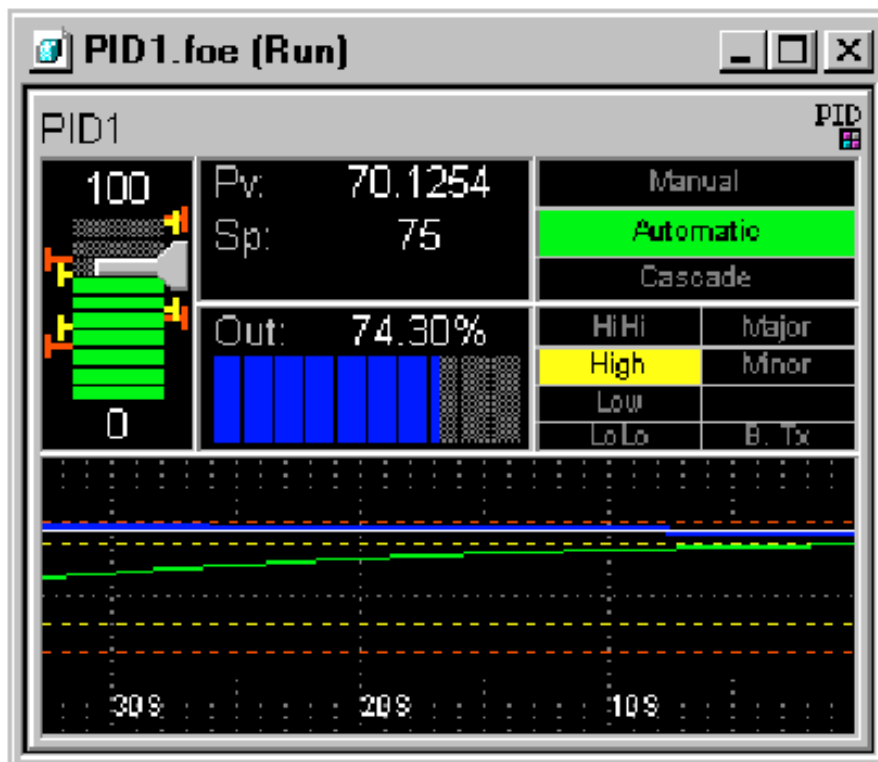
Corriendo y Controlando FOEs

Algunos FOEs, así como el PID FOE, están diseñados para correr automáticamente. Ellos tienen un método que es llamado una vez por cada análisis del Runtime Engine. Para el PID FOE, este método es el método DoControl. Cuando cargó en del Runtime Engine, el modo para esos FOEs esta indicado como Run, Pause, Stop, etc.

Algunos FOEs no tienen un método diseñado para correr automáticamente por cada análisis. Para esos FOEs, necesitan incluir códigos en un SFC o programa de texto estructurado para llamar un método para que el FOE se ejecute. Cuando descarga al Runtime Engine, el modo para estos FOEs está indicado en el Loaded, el cual está mostrado en la ventana Project. Estos FOEs están listos para ejecutar métodos y fijar propiedades cuando llame a ejecutar.

Ejecución de Animaciones

Algunos FOE aparecen en un modo animación de ejecución. Si usted muestra uno de esos FOEs en el Ambiente de Desarrollo, seleccionó parámetros que están actualizados cuando ellos cambian, como ilustrados por el PID FOE como aparece abajo. Para habilitar animaciones en runtime, sobre el menú **View**, pulse **Runtime Higliting**.



Cargando Parámetros

Algunos FOEs le permite cargar parámetros desde en el runtime engine a el Ambiente de Desarrollo. Este puede ser útil cuando quiere determinar empíricamente ciertos valores, y entonces cargar estos valores a la hoja de propiedades para el FOE. Refiérase a la documentación para el FOEs individual para los símbolos específicos que están cargados.

Usando el resto de los FOEs

Algunos Third-party FOEs tal vez no puedan ser diseñados para un control de fábrica en tiempo real. InControl FOEs necesita operar rápidamente y no causar perdida de memoria sobre períodos de tiempo extendidos. Si intentó usar el FOE a ejecución, necesita verificar que conoce las siguientes líneas de guía.

- No use códigos que hagan llamadas de funciones que requieran interacción humana, así como un método que muestre una el cuadro de diálogo.
- No llame un método que requiera un periodo relativamente largo de tiempo en ejecución, como sonar al portavoz de la unidad de hardware que causa un sonido del mensaje.
- Verifique que la memoria este asignada o desasignada correctamente por string que son pasados como parámetros a métodos o devueltos como valores desde métodos.
- Es muy recomendable que revise third-party FOEs antes de usarlos en el proceso de fábrica.
- Es recomendable que el modo threading para FOEs sea fijado a "Both". FOEs con modelos treathing que no esten fijados en "Both" y que son usados en ejecución, pueden causar una degradación seria en el

desempeño. El acceso a los valores de datos o métodos de el FOE pueden ser hasta 20 veces mas lento.

Referencia de un Objeto de Fábrica InControl

Para todos los controles ActiveX (incluyendo controles no diseñados como objetos de fábrica InControl), las propiedades y métodos pueden ser trazados a un tipo de datos existente IEC-61131 que está adicionado como símbolos desde la ventana Watch InControl, así como desde su programa de aplicación.

Para referirse a estos símbolos, use la siguiente formato denominado para propiedades:

<FOE name>.<property name>

Por ejemplo, para denominar variables llamadas Temp y Error en el FOE nombrado TempControl, use.

TempControl.Temp and TempControl.Error.

Para propiedades con parámetros, use el siguiente formato:

<FOE name>.<property name> (parameter list).

Use una declaración de Asignación para acceder una propiedad con parámetros así como usted hace con una función. Por ejemplo:

FOE1.param(3):=100;

Use el siguiente formato de llamada para métodos:

<FOE name>.<method name> (parameter list)

Por ejemplo, para llamar un método llamado Square en el FOE nombrado Math, el cual lleva un parámetro, use:

Math.Square(6);

Variables FOE están congeladas en sus valores pasados cuando el FOE es fijo al modo Stop. Por ejemplo, la variable InCascade usada en el PID FOE puede ser TRUE incluyendo cuando su lazo asociado no esta corriendo. Cuando esta monitoreando una variable, este seguro de monitorear la variable del sistema en Modo program, también debe verificar que el contenido de la variable es correcto. Excepto para valores forzados, valores FOE son inicializados cuando son transmitidos, inclusive durante una descarga inteligente. Valores forzados permanecen forzados durante una descarga inteligente. Valores retentivos de el FOE están recargados con sus valores salvados anteriormente sobre un inicio automático o cuando un proyecto es recargado.

Nota. Para ayudas acerca del diseño y desarrollo de FOEs, contacte su distribuidor para obtener el Equipo de herramientas de objetos de Fábrica InControl.

USO DEL FOE PID DE WONDERWARE

El PID FOE le permite manejar combinaciones de control proporcional, integral, y derivativo para sus aplicaciones de lazo. Ambos modos de afinación entrelazados y no entrelazados están soportados. El límite de ganancia derivativo es implementado para mejorar la ejecución de el modo derivativo de el lazo. Usted puede cambiar la acción integral de ON o OFF con una variable Boolean.

Además para el algoritmo de control PID, el soporte es proporcionado para automático, manual, en operación modo cascada , dos tipos de transferencia de menos choques para el modo automático, un transmisor de alarma interrumpida, alarma de desviación de error, sujetamiento de resultados y setpoint. Hay también varios parámetros de configuración que le permiten controlar el comportamiento de el PID durante la ejecución.

La hoja de propiedades que usa para configurar estos FOE están descritos, iniciando en “Parámetros generales de Configuración”. (-)

Operación en Tiempo de ejecución (Runtime)

Para fijar el modo PID, necesita entrar el código en un SFC o programa de Texto Estructurado que llame uno de los siguientes métodos:

- Manual
- Auto
- Cascade

Por ejemplo, para fijar el PID llamado TempControl a modo automático, use la siguiente función : TempControl.Auto ();

Para fijar el modo **PID** desde un programa **RLL** o una aplicación externa, así como InTouch, necesita cambiar la condición de una de las siguientes variables Boolean:

- RequestManual
- RequestAuto
- RequestCascade

También puede monitorear y cambiar el estado de la variable PID en la ventana Watch Window de InControl. Métodos y propiedades FOE aparecen en el Administrador de Símbolos (Symbol Manager). Si usted selecciona su propio nombre de símbolos para la variable del proceso, setpoint (punto fijo), y output (salida), el PID FOE automáticamente lee y escribe valores a estos símbolos como los apropiados. Por ejemplo, si TempSet contiene el dato setpoint de un caldero, y entra el TempSet en el campo **Remote Setpoint Symbol**, el PID usa el valor de TempSet en sus cálculos de la salida en modo cascada. Si usted no selecciona su propio nombre de símbolo para este término, el algoritmo PID usa símbolos basados sobre el nombre del lazo, e.g., PIDa.Pv, PID1.Rsp, PID.Out. Luego debe designar su propio código de programa para mover los valores correctos de estos términos hasta y desde estos símbolo como apropiados.

Todos los otros nombres de símbolos PID están basados sobre el nombre del lazo, por ejemplo, PID1.Mode, PID1.Kc, PID1.InAuto, etc.

Para habilitar animaciones de ejecución, sobre el menú View, pulse **Runtime Highlighting**.

Cargando Parámetros

El PID FOE le permite cargar parámetros desde el Runtime Engine hasta el ambiente de desarrollo. Esto puede ser útil cuando quiera determinar empíricamente los valores para afinación y alarma, por ejemplo, y luego cargar estos valores a la hoja de propiedades para el PID. La tabla "Symbols Used by the PID FOE" de la página 13-33 muestra los símbolo que son cargados.

➤ **Para cargar parámetros PID:**

Conecte al Runtime Engine.

Fije el PID a el modo Run.

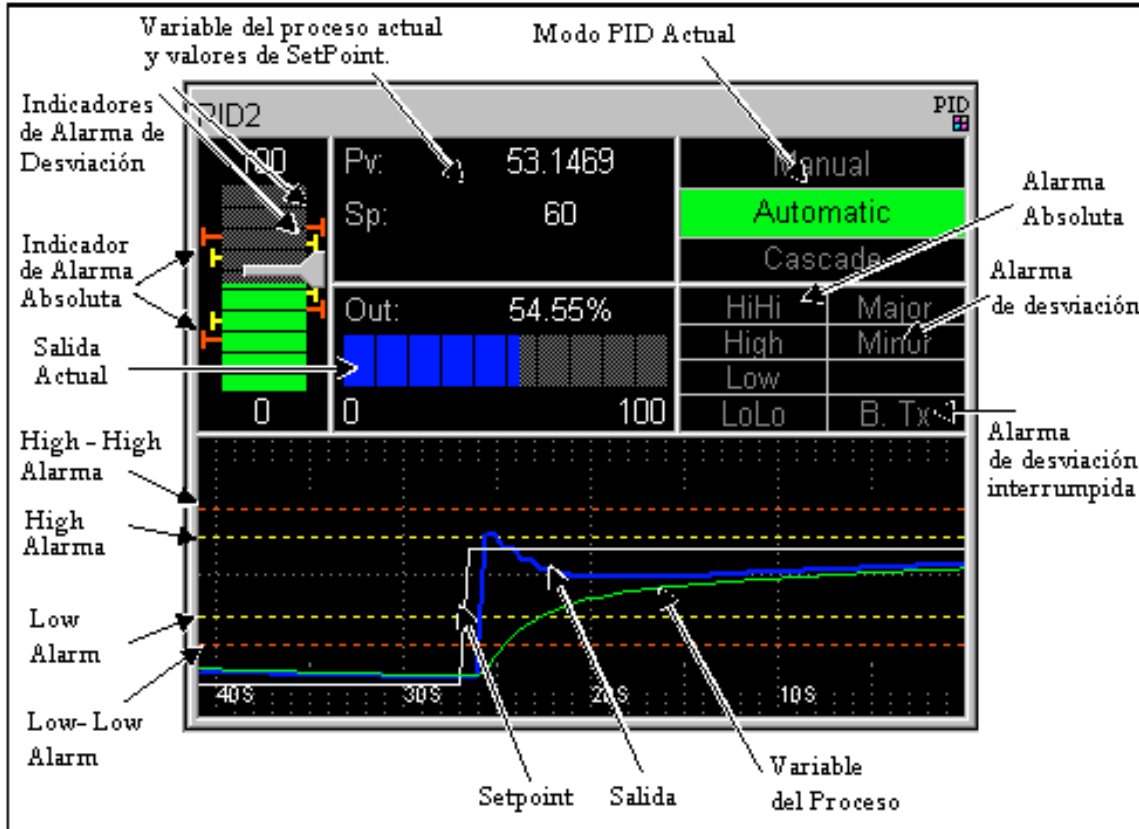
Abra el PID en el editor FOE.

Sobre el menú **Edit**, pulse **Upload Configuration**.

Usted puede cargar parámetros solamente a el PID en el proyecto donde el PID fue originalmente configurado. Esto es, usted no puede cerrar el proyecto, abrir un proyecto diferente, y luego cargar a un PID en el segundo proyecto. Si usted necesita un segundo PID FOE con parámetros idénticos a el primero, debe hacer una copia.

Ventana de presentación del PID

La siguiente figura ilustra la sección del frente de ventana principal del PID en Runtime.



Especifique la frecuencia en la cual la ventana principal FOE es cargado en el cuadro de diálogo **Runtime Engine Properties**.

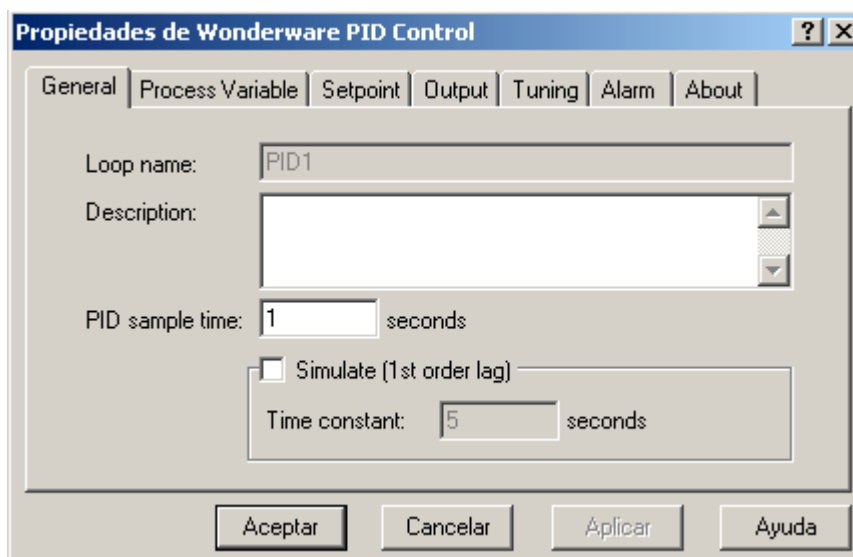
La ventana principal PID muestra datos de tendencia para la variable de procesos, setpoint, y output. Si las alarmas absolutas han sido configuradas y habilitadas, son mostradas también en la tendencia.

- Los valores que usted asigna a las unidades de diseño de la variable del proceso (símbolos: PvEuHi y PvEuLo) determina la escala vertical.
- El valor que asigna a el tiempo de muestra (symbol Sample Time) determina la escala horizontal.

Parámetros Generales de Configuración

Las siete etiquetas que comprenden el cuadro de diálogo de propiedades para el PID FOE están descritas en la página siguiente.

Abra el cuadro de diálogo de propiedades (right-click en el FOE y pulse **properties**). Luego pulse la etiqueta **General** de el cuadro de diálogo **Wonderware PID Control Properties** para entrar información básica acerca del lazo.



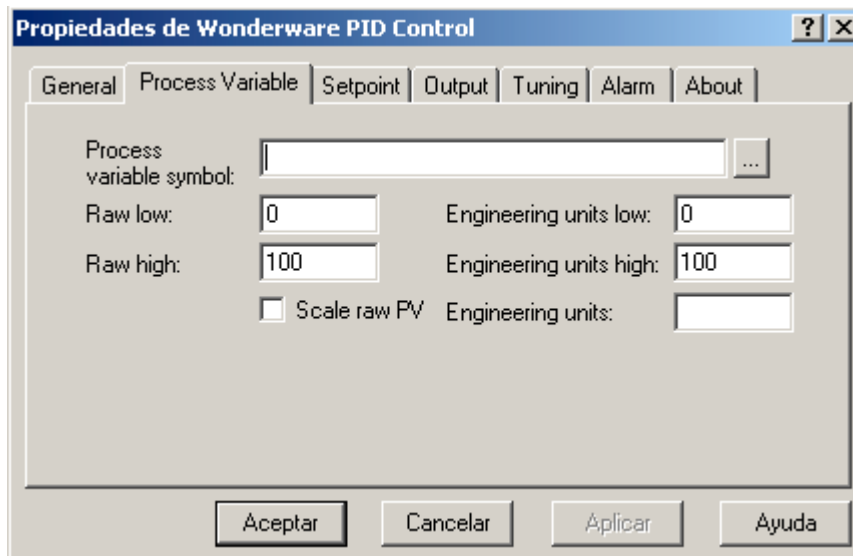
Propiedades de Control PID**El cuadro de chequeo**

El cuadro de chequeo	Símbolo	Descripción
Loop Name	LoopName	El campo solo lectura muestra el nombre de lazo. El nombre también aparece sobre el ventana principal de PID.
Loop Description	LoopDesc	Da al lazo una descripción significativa.
PID Sample Time	SampleTime	Entre un tiempo en segundos ($0.002 \leq \text{REAL value} \leq 172800.0$). La rata de muestreo determina cómo frecuentemente el algoritmo de el lazo evalúa la variable del proceso y calcula una nueva salida. El valor que asigna al tiempo de muestra determina la escala horizontal de la pantalla de tendencia.
Simulate	Simulate	Revise el cuadro de control Simulate para crear una simulación First Order Lag para revisar su programa. Este tiempo constante first-order causa un cambio gradual en la variable de el proceso después de un cambio inmediato en la salida. La ecuación usada en la simulación es la siguiente: $PV = a \times PV(st-1) + (1-a) \times \text{Output}$ $a = \left[\frac{\tau}{\tau + s\tau} \right]$ t = tiempo en segundo para la variable del proceso lograr un 63% de su valor final. st = Tiempo de Muestreo en segundos.
Time Constant	TimeConstant	Para una única simulación: Entre un tiempo en segundos (valor REAL, ≥ 0.002) para la variable del proceso para alcanzar un 63%de

de su valor final.

Configurando Parámetros de variables de proceso

Abra el cuadro de diálogo de propiedades (haga right-click en el FOE y pulse **Properties**). Luego pulse la etiqueta **Process Variable** de el cuadro de diálogo de Propiedades de Control PID Wonderware para entrar los parámetros de la variable del proceso.



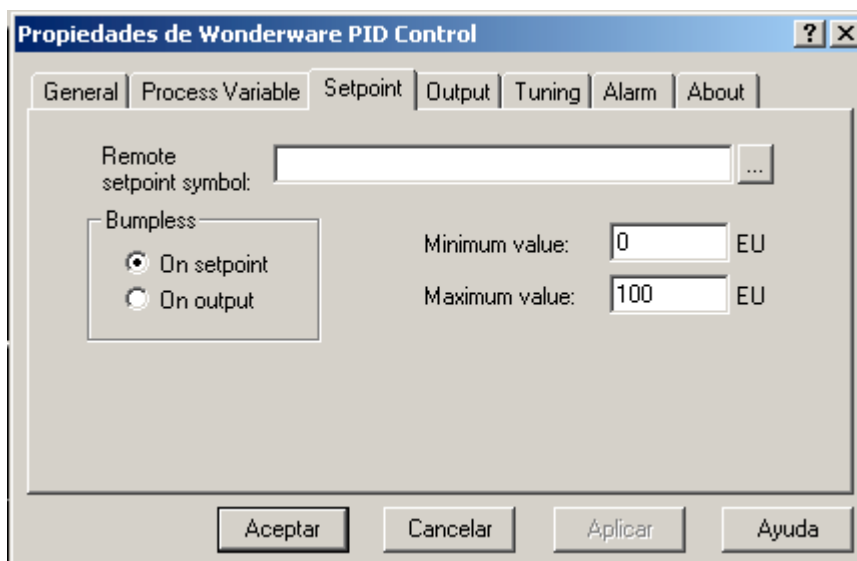
Propiedades de Control PID - Etiqueta de Variable del Proceso

El cuadro de Chequeo	Símbolo	Descripción
Process Variable Symbol	PvName	Entre un nombre para representar la variable del proceso (o la variable del proceso crudo si el cuadro de diálogo Scale Raw PV es revisada). Usted debe adicionar este símbolo a el Administrador de Símbolos. Ubique el cursor en el campo y haga doble click para abrir el Administrador de Símbolos (Symbol Manager) El PID FOE automáticamente lee valores desde este símbolo. Por ejemplo, si WaterTemp contiene la variable del proceso que lee el dato desde una caldera, y usted entra WaterTemp en el campo Process Variable Symbol , el PID usa el valor de WaterTemp en su cálculo de el resultado. Si usted no entra un nombre en el campo Process Variable Symbol, el algoritmo PID usa un símbolo predefinido basado en el nombre de el lazo, e.g., PID1.Pv. Luego debe designar su código de programa para mover valores a este símbolo predefinido.
Raw Low ¹	PvRawLo	Si usted revisa el cuadro de dialogo Scale Raw PV , entre un valor inferior para la variable del proceso crudo (raw). El valor superior debe ser mayor que el valor inferior.
Raw High ¹	PcRawHi	Si usted revisa el cuadro de dialogo Scale Raw PV , entre un valor superior para la variable del proceso crudo (raw). El valor superior debe ser mayor que el valor inferior.
Scale Raw PC	ScaleRawPv	Revisa el cuadro de diálogo Scale Raw PV para calcular la variable del proceso desde la entrada de la variable del proceso raw.

El cuadro de Chequeo	Símbolo	Descripción
Engineering Units Low ²	PvEuLo	Entre el valor inferior (valor REAL) para la variable del proceso en las unidades de proceso. El valor superior debe ser mayor que el valor inferior.
Engineering Units High ²	PvEuHi	Entre el valor superior (valor REAL) para la variable del proceso en las unidades de diseño. El valor superior debe ser mayor que el valor inferior.
Engineering Units	PvEngUnits	Entre las unidades de diseño. Este es un texto string ASCII que representa la unidad de medida, esto es, grados centígrados, PSI, Pounds/Hr, etc.

Configurando Parámetros de Setpoint

Abra el cuadro de diálogo de propiedades (right – clic el FOE y pulse Propertier). Luego pulse la etiqueta **Setpoint** de el cuadro de diálogo **Wonderware PID Control Propertier** para entrar parámetros a el setpoint de lazo.



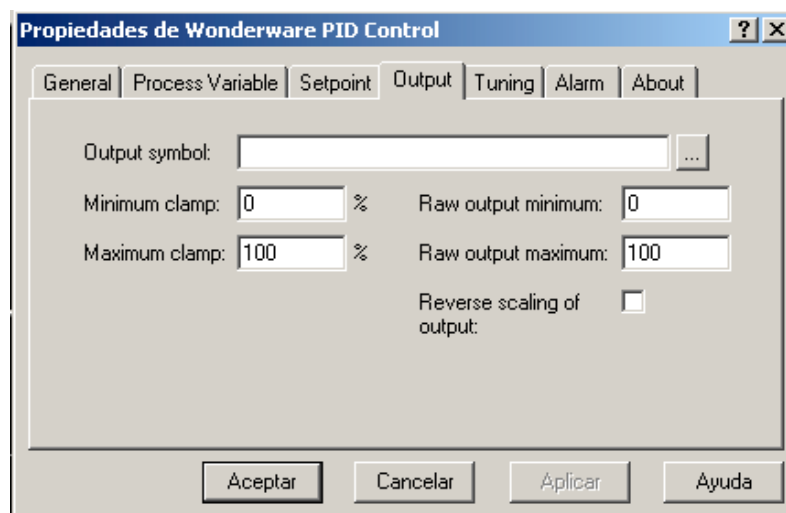
Propiedades de Control PID – Etiquetas SetPoint

El cuadro de Chequeo	Símbolo	Descripción
Remote Setpoint Symbol	RemoteSpName	<p>Cuando usted esta en modo lazo en cascada entre un nombre para representar el setpoint remoto. Usted puede ubicar el cursor en el campo y hacer doble clic para abrir el Administrador de Símbolos. El PID FOE automáticamente lee y escribe valores para estos símbolos.</p> <p>Por ejemplo, si TempSet contiene el dato setpoint para una caldera, y usted entra TempSet en el campo Remote Setpoint Symbol, el PID usa el valor de TempSet en su cálculo de output en modo cascada. Si usa al nombre de otro lazo PID, este lazo (el esclavo) usa la salida de otro lazo (el maestro) como el setpoint en modo cascada. El lazo maestro no podrá cambiar a modo automático a menos que el lazo esclavo este en modo cascada.</p>
Bumpless on Setpoint	Bumpless	<p>Seleccione Bumpless on Setpoint (1) para inicializar el setpoint a el valor activo de la variable de el proceso, y (2) para fijar el termino integral a el valor activo de la salida cuando el lazo descarga en modo automático. El modo automático inicializa con error cero y de esta manera la salida del lazo mantiene inalterado a través de la transición. Seleccione esta opción, fije el símbolo PID <loop name>. Bumpless a cero.</p>

El cuadro de Chequeo	Símbolo	Descripción
Bumpless on Output	Bumpless	Seleccione Bumpless o Output para calcular el término integral usando el valor inicial de el error y la ganancia proporcional de modo que la salida no cambie. El setpoint no es cambiado.
Minimum Value	SpLo	Entre el valor mínimo (valor REAL) para el setpoint. El valor mínimo debe ser mayor que o igual al valor inferior usado por la variable del proceso, y debe ser menor que el valor máximo de setpoint.
Maximum Value	SpHi	Entre El valor máximo debe ser menor que o igual a el valor superior usado por la variable del proceso, y debe ser mayor que el valor de setpoint mínimo.

Configurando parámetros de Salida

Abra el cuadro de diálogo de propiedades Luego pulse la etiqueta **Output** de el cuadro de diálogo **Wonderware PID Control Propertier** para entrar parámetros para la salida de el de lazo.

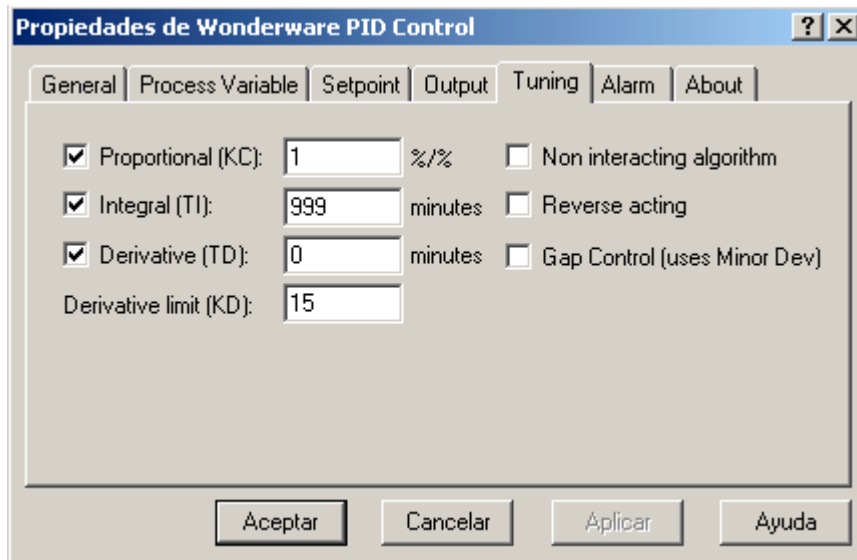


Propiedades de Control PID – Etiqueta de Salida

El cuadro de Chequeo	Símbolo	Descripción
Output Symbol	OutputName	Entre un nombre para representar la salida. Puede ubicar el cursor en el campo y hacer doble clic para abrir el Administrador de Símbolos. El PID FOE automáticamente lee y escribe valores a este símbolo. Por ejemplo, si TempOut contiene el dato de salida para una caldera, y usted entra TempOut en el campo Output Symbol , el PID escribe el valor de la salida calculada a TempOut. Si no entra un nombre en el campo Output Symbol , el algoritmo PID usa un símbolo predefinido basado sobre el nombre de el lazo, e.g., PID1.Output. Luego debe diseñar su código de programa para mover el valor correcto de este término hasta y desde este símbolo predefinido.
Minimum Clamp ¹	OutputClampLo	Entre el valor mínimo (valor REAL, 0 hasta un valor menor que la señal máxima) para la salida.
Maximum Clamp ¹	OutputClampHi	Entre el valor máximo (valor REAL, mayor que la señal mínima, hasta 100).
Raw Output Minimum ²	RawOutputMin	Entre el mínimo valor (valor REAL), para el cual la salida del lazo no es descargada
Raw Output Maximum ²	RawOutputMax	Entre el valor máximo (valor REAL) para el cual la salida de el lazo no es descargada.
Reverse scaling of output	ReverseOutput	Revise el cuadro de dialogo Reverse Scaling of Output para hacer un escalamiento inverso de la salida para la salida del raw.

Configurando Parámetros de Afinación (Tuning)

Abra el cuadro de propiedades (right – clic a el FOE y pulse Properties). Entonces pulse la etiqueta Tuning de la caj de diálogo **Wonderware PID Control Propertier** para entrar los parámetros de afinación.



Propiedades de Control PID – Etiquetas de Afinación

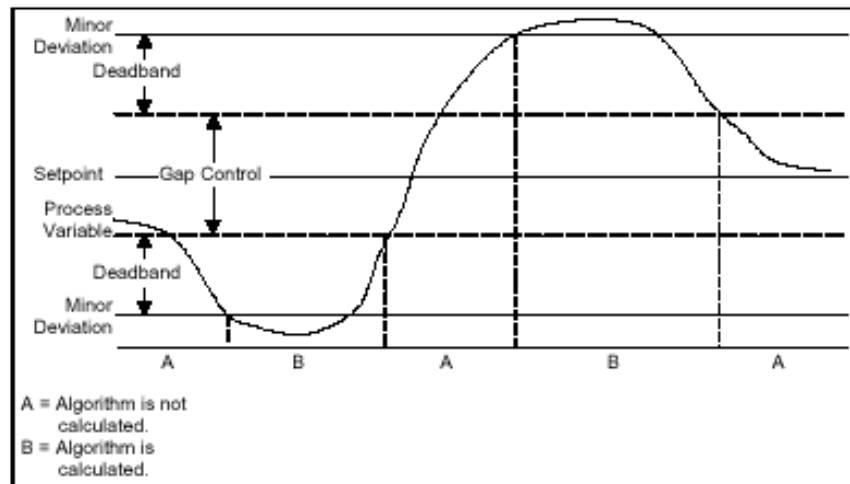
El cuadro de Chequeo

El cuadro de Chequeo	Símbolo	Descripción
Proportional Checkbox *	EnableProportional	Revisa el cuadro de chequeo para habilitar el término Proporcional.
Proportional Field.	Kc	Entre un valor (valor REAL, 0 para 999) para la ganancia en % / %.
Integral Checkbox *	EnableIntegral	Revise el cuadro de dialogo Integral para validar el termino Integral.
Integral	Ti	Entre un valor (valor REAL, 0.001 hasta 999) para restablecer el tiempo en minutos. Internamente, 999 es convertido a $1e^{39}$ para los cálculos.
El cuadro de chequeo Derivativo	Enable derivative	Revisa el cuadro de dialogo Derivative para validar el término Derivative.

El cuadro de Chequeo	Símbolo	Descripción
Derivative	Td	Entre un valor (valor REAL, 0.0 hasta 999) para la tasa tiempo derivativo en minutos.
Derivative Limit	Kd	Entre un valor (valor REAL, 10.0 hasta 20.0) para el coeficiente limitante de la ganancia derivativa. Este límite ayuda a mantener la estabilidad del lazo y desechar ruido.
Non – Interacting Algorithm	NonInteracting	Revisa el cuadro de chequeo Non-Interacting Algorithm para los modos integral y derivativo para actuar independientemente del modo proporcional. Esto es, la ganancia es fijada para 1 por los modos integral y derivativo.
Reverse Acting	ReverseActing	Revise el cuadro de diálogo Reverse Acting para seleccionar un lazo que genere un resultado de ganancia negativa. Un cambio positivo en el error rinde un cambio en la salida, y como el valor positivo de el error aumenta, la señal de salida decrece.
Gap Control checkbox	EnableGabControl	Revise el cuadro de diálogo Gab Control para habilitar el control gap Esté seguro al entrar los valores en los campos Minor Deviation y Alarm Deadband sobre la etiqueta Alarms . Estos valores son usados para usar calcular control gap, como se ilustra después.

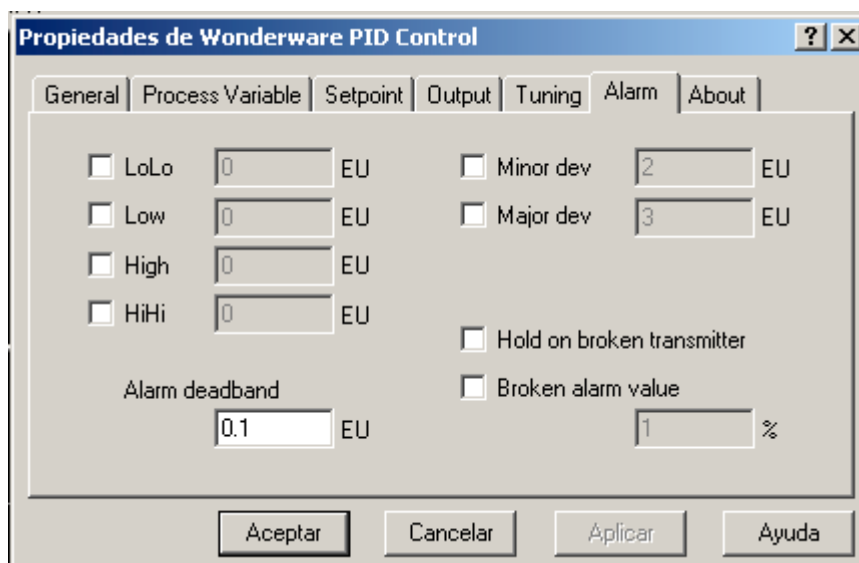
Control Gap

La siguiente figura ilustra como la desviación del control gap previene vibraciones cuando la variable del proceso se acerca a un límite de desviación.



Configurando Alarmas de Lazo

Abra el cuadro de diálogo right-click del FOE u pulse **Propertier**). Luego pulse la etiqueta **Alarms** de el cuadro de diálogo **PID Control Propertier** para entrar los parámetros de alarma. Ver “PV Alarms and Deviations” (page 13-30) para un ejemplo de la relación de las alarmas y las desviaciones para el setpoint y los límites de la variable del proceso.



Propiedades de Control PID – Etiquetas de Alarma

El cuadro de Chequeo	Símbolo	Descripción
LoLo Checkbox ¹	MonLoLoAlarm	Revisa el cuadro de chequeo LoLo para activar la alarma low-low.
LoLo Field	LoLoAlarm	Entra el valor (valor REAL menor que o igual al valor mas bajo) para la alarma low-low.
Low Checkbox ¹	MonLowAlarm	Revisa el cuadro de dialogo Low para activar el cuadro de dialogo inferior.
Low Field	LowAlarm	Entra el valor (valor REAL menor que el valor superior) para la alarma inferior.
High Field	HighAlarm	Entra el valor (valor REAL menor que el valor high-high) para la alarma superior.
Hi Hi Checkbox ¹	MonHiHiAlarm	Revisa el cuadro de chequeo para activar la alarma high-high.
Hi Field	HiHiAlarm	Entra el valor (Valor REAL mayor o igual al valor superior) para la alarma high-high.
Alarm Deadband Field ^{2,3}	AlarmDeadBand	Entra el valor (valor REAL dentro de un espacio de diseño) en las unidades de diseño.
Minor Deviation Alarm Checkbox ^{1,2}	MonMinorDevAlarm	Revisa el cuadro de dialogo Minor Dev para activar la alarma de desviación secundaria (menor).
Minor Deviation Alarm Field ^{1,2}	MinorDevAlarm	Entra el valor (valor REAL dentro del espacio de diseño, y menor que el valor de la alarma desviación principal) en unidades de diseño. el valor máximo de el error que dispara la alarma de desviación principal.

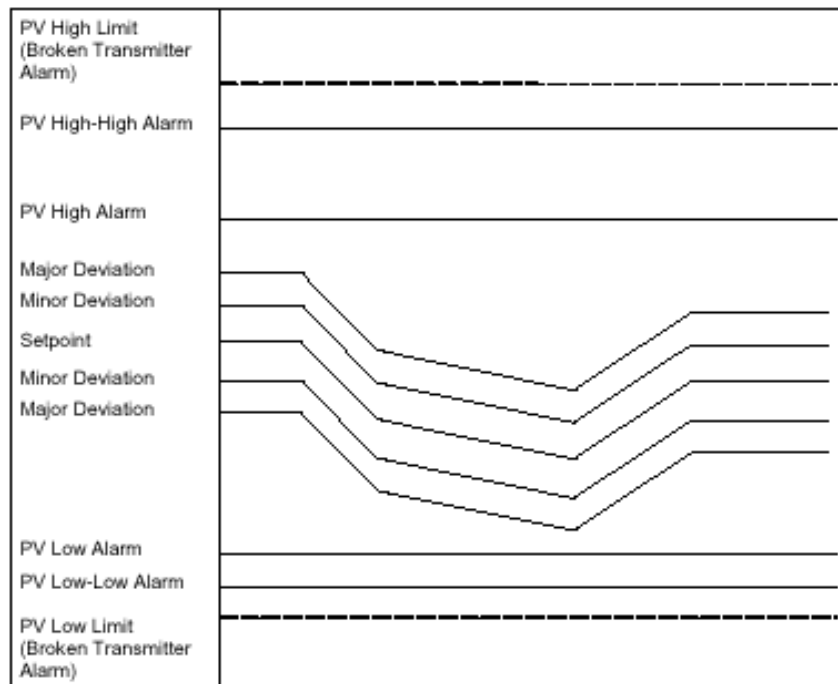
El cuadro de Chequeo	Símbolo	Descripción
Major deviation 9Alarm Checkbox ¹	MonMajorDevAlarm	Revise el cuadro de dialogo Major Dev para activar la alarma de desviación principal.
Major Deviation Alarm Field	MajorDevAlarm	Entre un valor (valor REAL dentro del espacio de diseño, y mayor que la alarma de desviación secundaria) en unidades de diseño. Este es el valor máximo de error que dispara la alarma de desviación principal.
Hold on BrokenTransmitter Checkbox ¹	HoldPvOnBrokenTx	Revise el cuadro de chequeo Hold on Broken Descargar para retener la variable del proceso, pero no la variable del proceso raw, a el valor aceptable anterior cuando recientemente valores calculados están fuera del rango especificado en el campo Broken Alarm .
Broken Alarm Value Checkbox ¹	MonBrokenTx	Revise el cuadro de dialogo Broken Alarm Value para monitorear la alarma del transmisor.
Broken Alarm Value Field	BrokenTxValue	Entre el valor (valor REAL) 0 – 100%) para representar el punto de alarma por encima y por debajo del espacio de la variable del proceso raw que dispara la alarma de el transmisor roto.

PV Alarms and Deviation

La siguiente figura ilustra la relación de las alarmas y desviaciones para el setpoint y los límites de la variable del proceso.

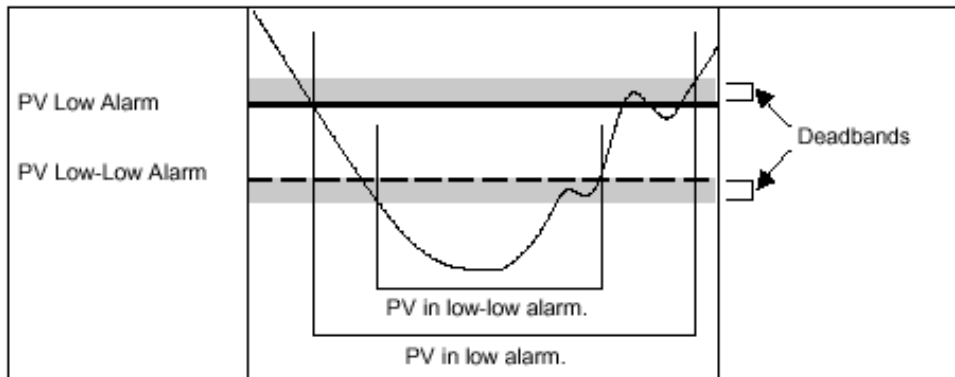
Nota: Las alarmas de la variable del proceso son preparadas, mientras las desviaciones son vinculadas al setpoint. Cuando el operador cambia el setpoint, las desviaciones mayores y menores siguen de acuerdo al setpoint.

Las alarmas absolutas y de desviación aparecen sobre el ventana principal PID. Ver “PID ventana principal” para una ilustración del ventana principal.



Acción Histéresis

La siguiente figura ilustra como una alarma deadband previene vibraciones cuando la variable del proceso se aproxima a un límite de alarma.



Revisando Versión FOE

Abra el cuadro de diálogo de propiedades (right-click el FOE y pulse **Properties**). Luego pulse la etiqueta **About** de el cuadro de diálogo Wonderware PID Control Propertier para revisar la versión de el FOE.



SÍMBOLOS Y MÉTODOS PID

La siguiente tabla muestra todos los símbolos que son usados por el FOE PID.

Símbolos usados por el FOE PID

Símbolo	Descripción	Read/Write
AlarmDeadband ¹	Valor de histéresis de la alarma (unidades de diseño)	RW REAL
BrokenTx ²	Variable del proceso que esta dentro del estado de la alarma del transmisor roto.	RO BOLEAN
BrokenTxValue ¹	Valor fraccionado del espacio de variable usado para determinar si la variable del proceso esta en estado de transmisor roto.	RW REAL
Bumpless ¹	0 = Fija el setpoint igual a la variable del proceso sobre descarga para modo automático o de cascada. 1 = no cambie el Setpoint en la transición a modos automáticos o de cascada. En cambio retrocede cálculos de prejuicio.	RW INT
CascadedLoop ²	Setpoint remoto es un lazo PID.	RO BOLEAN
Description ¹	Descripción del FOE.	RW STRING
EnableDerivate ²	Algoritmo PID derivativo usado.	RO BOLEAN
EnableGapControl ²	Habilita Control Gap.	RO BOLEAN
EnableIntegral ²	Algoritmo PID Integral usado.	RO BOLEAN
EnableProportional ²	Algoritmo PID proporcional usado.	RO BOLEAN
Error ²	SP – PV (variable del proceso menos que setpoint).	RO REAL
FreezeOutput	Detiene PID con salida mantenida a el valor anterior.	RW BOLEAN
HighAlarm ¹	Alarma de altura (unidades de diseño)	RW REAL

Símbolo	Descripción	Read/Write
HiHi Alarm ¹	Alarma de altura alta (unidades de diseño).	RW REAL
HoldPvOnBrokenTx ¹	Mantiene la variable del proceso en el anterior valor aceptable en el evento del transmisor .	RW BOOLEAN
InAuto ²	PID esta en modo automático. (PID es activo)	RO BOOLEAN
InCascade ²	PID esta en modo cascada (PID esta activo, setpoint remoto).	RO BOOLEAN
InHighAlarm ²	PV esta sobre de la alarma superior.	BOOLEAN
InHiHiAlarm ²	PV esta sobre la alarma superior alta.	RO BOOLEAN
InLoLoAlarm ²	PV esta debajo de la alarma inferior baja	RO BOOLEAN
InMajor DevAlarm ²	Valor absoluto de Error esta de bajo de MajorDevAlarm.	9RO BOOLEAN
InManual ²	PID esta en modo manual (PID esta inactivo, fuera de manual).	RO BOOLEAN
InMinorDevAlarm ²	Valor Absoluto de Error esta sobre MinorDevAlarm.	RO BOOLEAN
Integral	PID: error de salida Cero en automático o modo cascada (%).	RW REAL
Interlock ¹	PID no puede ser fijado a automático o cascada.	RW BOOLEAN
Kc ¹	Ganancia proporcional (%/%)	RW REAL
Kd ¹	Límite de Ganancia derivativa (no unidades)	RW REAL
LoLoAlarm ¹	Alarma inferior baja (unidades de diseño)	RW REAL
InMinorDevAlarm ²	Valor absoluto de error esta sobre el MinorDevAlarm	RO BOOLEAN
Integral	Bias (sesgo) PID : error de salida cero en automático o modo cascada (%).	RW REAL

Símbolo	Descripción	Read/Write
Interlock ¹	PID no puede ser puesto en automático o cascada.	RW BOOLEAN
Kc ¹	Ganancia proporcional (%/%).	RW REAL
Kd ¹	Límite de Ganancia derivativo.	RW REAL
LoLoAlarm ¹	Alarma inferior baja (unidades de diseño.)	RW REAL
LoopName ²	Nombre.	RO STRING
Low Alarm ¹	Alarma menor (unidades de diseño).	RW REAL
MajorDevAlarm ¹	Alarma de desviación de error principal (unidades de diseño).	RW REAL
MinorDevAlarm ¹	Alarma de desviación de error menor (unidades de diseño).	RW REAL
Mode	Indica modos activos del PID. La propiedad no es disponible si el control Actives no es usado con InControl. 0= Descargado desde el Runtime Engine. 1= suspendido 2= pausa 3= Examinar 4= programa 5= ejecutar 6= Falla 7= Completo 8= Cargado a en el runtime engine.	RW INT
MonBrokenTx ¹	Supervisa el estado del transmisor roto.	RW BOOLEAN
MonHighAlarm ¹	Supervisa el estado de la alarma superior.	RW BOOLEAN
MonHiHiAlarm ¹	Supervisa el estado de la alarma superior alta.	RW BOOLEAN

Símbolo	Descripción	Read/Write
MonLoLoAlarm ¹	Supervisa el estado de la alarma inferior baja.	RW BOOLEAN
MonLowAlarm ¹	Supervisa el estado de la alarma inferior.	RW BOOLEAN
MonMajorDevAlarm	Monitorea el estado de la alarma de desviación principal.	RW BOOLEAN
MonMinorDevAlarm	Monitorea el estado de la alarma de desviación secundario.	RW BOOLEAN
NoDevAlarm ²	Valor absoluto de el error es menor que MinorDevAlarm.	RO BOOLEAN
NonInteracting ¹	Modos PID son independientes.	RW BOOLEAN.
Out ²	Resultados en formato 0-100%	RW REAL
OutClampedHi ²	La salida esta en el valor OutputClampHi	RO BOOLEAN
OutClampedLo ²	La salida esta en el valor OutputClampLo	RO BOOLEAN
Output ²	Salida Raw dentro de un rango de usuario definido.	RW REAL
OutputClampHi ¹	Límite de alerta para la salida.	RW REAL
OutputClampLo ¹	Límite de alerta para la salida.	RW REAL
OutputName ²	Nombre de la salida	RO STRING
Pv ²	Variable de proceso en unidades de diseño.	RW REAL
PvEngUnits ¹	Unidades de diseño de la variable del proceso.	RW STRING
PvEuHi ¹	Rango superior de unidades de diseño.	RW REAL

Símbolo	Descripción	Read/Write
PvEuLo ¹	Rango inferior de unidades de diseño.	RW REAL
PvName ²	Nombre de la variable de el proceso.	RO STRING
PvRaw ²	Variable del proceso Rw desde el campo.	RW REAL
PvRawHi ¹	Valor máximo para la variable del proceso Raw.	RW REAL
PvRawLo ¹	Valor mínimo para la variable del proceso Raw.	RW REAL
RawOutputMax ¹	Valor máximo para la salida raw.	RW REAL
RawOutputMin ¹	Valor mínimo para la salida raw.	RW REAL
RemoteSpName ²	Nombre del setpoint en modo cascada.	RO STRING
RequestAuto	Pide que el modo se ponga en automático.	RW BOOLEAN
RequestCascade	Pide que el modo se ponga en cascada.	RW BOOLEAN
RequestManual	Pide que el modo se ponga en manual.	RW BOOLEAN
ReverseActing ¹	Fija el modo PID a acción reversa.	RW BOOLEAN
Reverse Output ¹	Fija la salida a un máximo de 0% y un mínimo de 100%.	RW BOOLEAN
SampleTime ¹	Tiempo de muestra para computación PID (segundos).	RW REAL
ScaledError ²	Error Normalizado.	RO REAL
ScaleRawPv ¹	Usa la variable del proceso para cálculo del valor de la variable del proceso. Cuando TRUE, PvRaw es habilitado para escribir y Pv es solo lectura.	RO BOOLEAN
ScanRate ²	Tiempo actual entre ejecuciones (segundos).	RO REAL

Símbolo	Descripción	Read/Write
SignError ²	0= señal de error es positivo.	RO BOOLEAN
Simulate	Simula la variable del proceso con un primer retraso del orden interno.	RO BOOLEAN
Sp	Setpoint PID	RW REAL
SpClampedHi ¹	SetpPoint esta cerca o sobre el valor SpHi.	RO BOOLEAN
SpClampedLo ¹	SetPoint esta cerca o por debajo del vaor SpLo	RO BOOLEAN
SpHi	Valor máximo de setpoint (unidades de diseño).	RW REAL
SpLo	Valor mínimo de setpoint (unidades de diseño).	RW REAL
Td ¹	Tiempo derivativo (minutos).	RW REAL
Ti ¹	Tiempo Integral (minutos).	RW REAL
TimeConstant	Constante de tiempo de simulación (segundos)	RW REAL
TransitionToAuto ²	Indica que PID esta en estado de transición.	RO BOOLEAN
TransitionCascade ²	Indica que PID esta en estado de transición.	RO BOOLEAN
TransitionToManual ²	Indica que PID esta en estado de transición.	RO BOOLEAN

Métodos asociados con el FOE PID

Símbolo	Descripción.
Auto ()	Fija el lazo a modo automático.
DoControl ()	Ejecuta el algoritmo PID. Llamado en cada análisis. Si el PID FOE no es usado con InControl usted debe diseñar un código de programa para llamar el método DoControl para que el PID FOE opere correctamente.
Manual ()	Fija el lazo a modo manual.
Cascade ()	Fija el lazo a modo cascada.

TEORÍA DE OPERACIÓN PID

Esta sección define el algoritmo PID y describe sus ciclos de operación.

Algoritmo PID

El algoritmo PID es calculado durante cada tiempo de muestreo. Si el tiempo de muestreo no es un múltiplo del tiempo de análisis del Runtime Engine, el algoritmo es calculado durante la próxima ejecución del FOE. Esto es, si el tiempo de muestreo es 500ms y el tiempo de análisis es 300ms, el algoritmo es calculado en el siguiente intervalo: 0ms, 600ms, 1200ms, 1500ms, etc.

El FOE revisa los cambios de modo, durante cada análisis del Runtime Engine. Para un FOE PID que se ejecuta con una prioridad normal, el FOE hace cálculos de alarma durante cada análisis. Si el FOE ha sido asignado a prioridad baja, el FOE hace cálculos de alarma cada vez que este corre.

Para mas información acerca de prioridades de programa, ver “Línea de tiempo del ” en el capítulo “Administración de sistemas InControl”.

Ecuación PID

Salida = (Proportional + Integral + Derivative x 100)

Los componentes del algoritmo PID son calculados como se muestra a continuación:

Término proporcional

Term Proporcional = error medido x Kc

Para un lazo de realimentación, Kc es multiplicado por -1.

Término Integral

Term Integral = Integral + $\left(\frac{Kc \times \text{Tiempo de muestreo}}{60 \times Ti} \right) \times \text{Error medido}$

Para un lazo de realimentación, Kc es multiplicado por -1.

Para una acción no integral, Kc es multiplicado por 0.

El término integral es condicionado a los límites superior e inferior.

Para un modo de no interacción, Kc es igual a 1.

Término Derivativo

Término derivativo = -Kc x Kd x $\left(\frac{Pv - Pv \text{ filtrado}}{PvEuHi - PvEuLo} \right)$

Para una acción no integral, Kc es multiplicado por 0.

Para un modo de no interacción, Kc es igual a 1.

Variable ScaledError (error medido)

ScaledError = $\left(\frac{Sp - Pv}{PvEuHi - PvEuLo} \right)$

Variable FilteredPv (Pv Filtrado)

$$\text{FilteredPv} = \text{FilteredPv} + (\text{Pv} - \text{FilteredPv}) \times \left(\frac{\text{Tiempo de Muestreo}}{\text{Tiempo de Muestreo} + \frac{(\text{Td} \times 60.0)}{\text{Kd}}} \right)$$

Definición de Variables de Algoritmos

Variable	Definición
EnableDerivative	Término derivativo que es usado en el algoritmo.
EnableIntegral	Término integral que es usado dentro del algoritmo.
EnableProportional	Término proporcional que es usado en el algoritmo.
Kc	Ganancia proporcional (%/%).
Kd	Límite de ganancia derivativa (adimensional).
NonInteracting	Modos PID que son independientes.
Out	Salida en formato 0 – 100%.
Pv	Variable del proceso en unidades de diseño.
PvEuHi	Rango superior para unidades de diseño.
PvEuLo	Rango inferior para unidades de diseño.
ReverseActing	Convierte el modo de PID en acción de realimentación.
SampleTime	Tiempo de muestreo para el cálculo PID.
ScaledError	Error Normalizado.
ScanRate	Tiempo actual entre ejecuciones (segundos).
ScanTime	Frecuencia en que el PID es inicialmente llamado (segundos).
Sp	PID Setpoint.
Td	Tiempo Derivativo (Minutos).
Ti	Tiempo integral (Minutos).

Modos de lazo

El FOE PID soporta tres modos de operación: manual, automático, y cascada. En cascada opera exactamente como el modo automático, pero el setpoint es leído desde una localización remota.

En modo automático, la salida es computada por la vía del algoritmo PID. En el modo manual, la ecuación PID no está activa y el operador puede fijar la salida del lazo directamente, por escritura, para cualquiera de los siguientes símbolos:

- `<loop name>.Out` Este símbolo siempre oscila entre el valor desde 0% hasta 100%.
- `<loop name>.Output`. Este símbolo lleva valores que son especificados en él `<loop name>.RawOutputMax` y `<loop name>.RawOutputMin`.
- El nombre del símbolo introducido en el campo **Output Symbol** de la etiqueta **Output**.

Con el modo manual en el ciclo, si cambia el valor de cualquiera de estos dos símbolos: `<loop name>.Out` o `<loop name>.Output`, el otro símbolo refleja aquel cambio. Los valores de salida Raw máximo y mínimo determinan cómo el ciclo escribe para presentar los dispositivos. Por ejemplo, una tarjeta I/O acepta los valores de 819 hasta 4096 para una señal de 4-20 mA. Necesita fijar el mínimo de salida Raw a 819 y un máximo de salida Raw a 4096 para que iguale 819 con 0% e iguale 4096 con 100%.

El FOE PID soporta asegurar las salidas cuando el lazo está en cualquier modo. Automático, cascada, o manual. Si la salida va más allá de cualquier valor del símbolo `<loop name>.OutputClampLo` o del valor del símbolo `<loop name>.OutputClampHi`, la salida está sujeta al valor superior o inferior. Sin embargo, el símbolo `<loop name>.Output` está en uno de los límites, entonces el indicador `<loop name>.OutClampedLo` o `<loop name>.OutClampedHi` Boolean es TRUE.

El FOE PID también tiene la capacidad para dar escala – reversible a la salida. Esto le permite usar el lazo con actuadores de acción reversible, así como válvulas de control (fail –open) fallas-abiertas. Cuando seleccione la escala reversible, Las siguientes acciones ocurren:

- Cuando el símbolo **<loop name>.Out** iguala a cero, **<loop name>.Output** es fijado al valor contenido por el símbolo **<loop name>.RawOutputMax**.
- Cuando el símbolo **<loop name>.Out** iguala 100%, El símbolo **<loop name>.Output** es fijado al valor contenido por el símbolo **<loop name>.RawOutputMin**.

La opción de escala reversible es útil en la autorización de un cero en el símbolo **<loop name>.Out** para indicar que la válvula de control asociada es inicialmente ordenada a una posición cerrada.

El símbolo **<loop name>.Integral** es también afectado por el valor de las variables de salida. En modo manual, el símbolo **<loop name>.Integral** rastrea el valor del símbolo **<loop name>.Out**. El símbolo **<loop name>.Integral** representa la salida de estado estable del lazo en modo automático o cascada cuando el error del lazo es cero.

El término “bumpless transfer” se refiere a como el lazo maneja los cambios del modo desde manual a automático. El FOE PID soporta dos métodos de transferencia de bumpless a automático.

- Bumpless on setpoint: tal como las transiciones del lazo dentro del automático, el setpoint es inicializado al valor actual de la variable del proceso y el valor del símbolo del **<loop name>.Integral** es fijado al valor actual de **<loop name>.Out**. El modo automático inicializa con cero error, y así la salida del lazo se mantiene inalterada a través de la transición.

- **Bumpless on Output:** El setpoint no es cambiado. El integral es computado usando el valor inicial del error y la ganancia proporcional, y la salida permanece invariable. Desde que exista un error, la acción integral empieza cambiando la salida a la rampa de la variable del proceso para el setpoint.

Símbolos de programas están congelados en sus valores anteriores cuando un programa es fijado al modo Stop. Por ejemplo, el símbolo InCascade puede ser TRUE inclusive cuando su lazo (ciclo) asociado no esta corriendo. Cuando está monitoreando un símbolo de programa, esté seguro para monitorear la variable del sistema Mode program también, para verificar que el contenido del símbolo es válido.

Reset Windup

Restaurar windup es también crítico para aplicaciones de control de procesos. La protección PID FOE windup maneja el comportamiento restringido de la salida del lazo por congelación del integral y sujetamiento de la salida en el valor máximo o mínimo especificado por usuario. El algoritmo también le permite al integral mover la salida lentamente hacia atrás desde la represión como determinado por el signo de la integración.

USO DEL FOE DE ALARMA ANÁLOGA DE WONDERWARE

El FOE de alarma análoga le permite monitorear para una condición de alarma de una señal de entrada análoga. Puede especificar rangos de alarma para la variable del proceso y para el setpoint designado, y también monitoreo para una condición de descarga interrumpida.

Las láminas de propiedades que usa para configurar estos FOE están descritas, empezando en “Parámetros de Configuración general” (-).

Operación de Ejecución

Puede empezar el FOE de Alarma Análoga ejecutando en forma manual o de programa.

➤ Para correr el FOE de Alarma Análoga:

- Desde el menú **Runtime**, pulse **Run Project** o seleccione el FOE de Alarma Análoga y desde el menú **Runtime** pulse **Run Program**
- Entre el código en otro programa que escriba un valor de 5 a la variable del Modo programa del FOE de Alarma Análoga.

Otros valores son válidos. Ver la tabla “Valores de Variable del Sistema Modo Valido” en el capítulo “Proyectar un funcionamiento”.

Después de introducir el modo Run, el FOE automáticamente empieza monitoreando alarmas. Por ejemplo, si TempMon contiene el dato setpoint designado para una caldera, y usted entra TempMon en el campo **Remote Setpoint Symbol**, la alarma análoga usa el valor de TempMon como la base para activación de alarmas. La tasa de actualización es siempre de 0.5 segundos.

Si usted no entra un nombre dentro del campo **Remote Setpoint Symbol**, el FOE compara la variable del proceso con el setpoint mínimo (el valor entrado por la alarma baja). Si no entra un nombre dentro del campo **Process variable Symbol**,

el FOE compara el valor entrado por las unidades de diseño bajas con el valor setpoint.

Usted puede monitorear y cambiar los estados de las variables de alarma análoga desde la ventana InControl Watch. Todos los métodos y propiedades aparecen en el Symbol Manager.

Todos los nombres de símbolos de alarmas análogas están basados sobre el nombre de alarma análoga, por ejemplo, AA1.InHighAlarm, TempMon13.BrokenTx, WaterTank.PV.

Para habilitar animaciones de ejecución, sobre el menú **View**, pulse **Runtime Highlighting**.

Cargando Parámetros

El FOE de alarma análoga le permite cargar parámetros desde el Runtime Engine hasta el Ambiente de Desarrollo. Este puede ser útil cuando se quiere determinar empíricamente los valores para alarma, por ejemplo, y cargar estos valores a la lámina de propiedades para la alarma análoga. La tabla “Símbolos usados por el FOE de alarmas análogas” muestra los símbolos que son cargados.

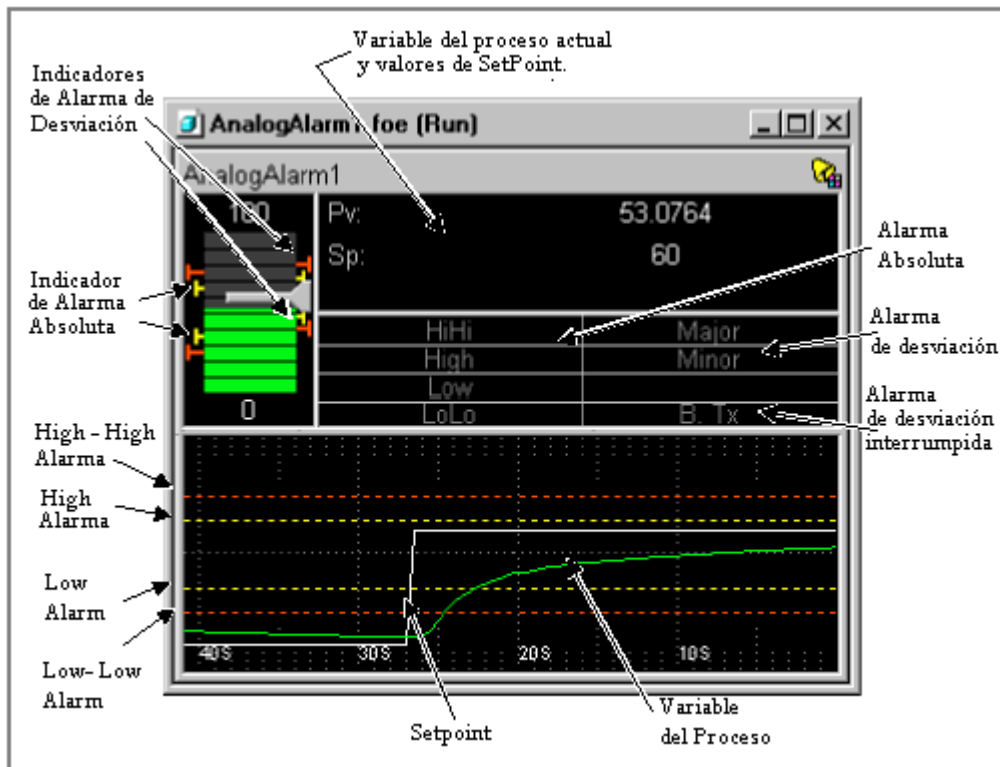
- **Para cargar parámetros de alarma análoga:**

1. Conecte al Runtime Engine.
2. Fije el FOE de Alarma Análoga a modo **Run**.
3. Abra la alarma análoga en el editor FOE.
4. Sobre el menú **Edit**, pulse **Upload Configuration**.

Puede cargar parámetros solo a la alarma análoga en el proyecto donde la alarma análoga fue originalmente configurada. Esto es, no puede cerrar el proyecto, abrir un proyecto diferente, y luego cargar una alarma análoga en el segundo proyecto. Si usted necesita un segundo FOE de Alarma análoga con parámetros idénticos al primero, debe hacer una copia.

Ventana principal de Alarma Análoga

La siguiente figura ilustra la ventana principal de alarma análoga tal como esta aparece al ejecutarla.



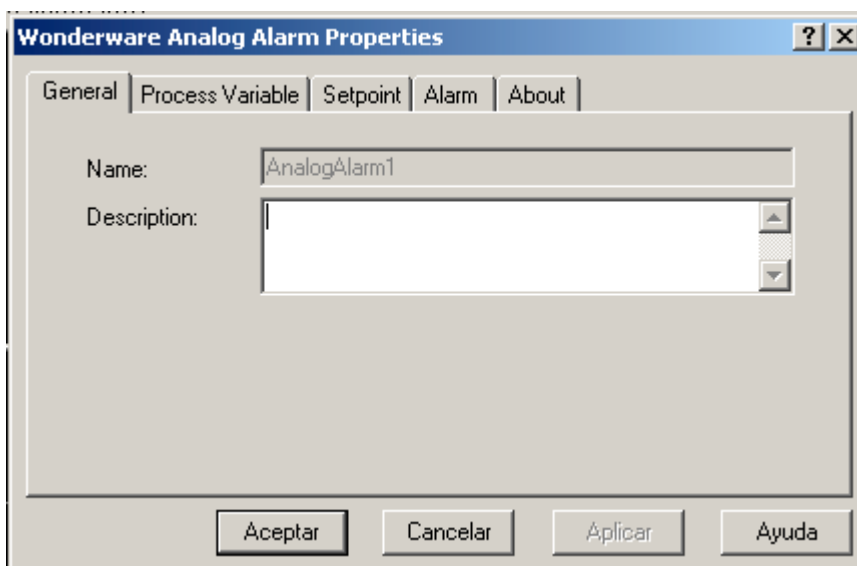
Especifique la frecuencia a la cual la ventana principal FOE es actualizada en el cuadro de diálogo **Engine Properties**.

La ventana principal de alarma muestra datos de tendencia para la variable del proceso y setpoint. Si la alarma absoluta ha sido configurada y habilitada, también son mostrados en la tendencia. Defina la escala vertical en la pantalla trend (tendencia) por el ajuste de los valores para las unidades de diseño de variable del proceso (símbolos: PvEuHi y PvEuLo). La escala horizontal es definida automáticamente por InControl, basado sobre un 0.5 segundos de tiempo actual.

Configurando Parámetros Generales

Las cinco etiquetas que comprenden la el cuadro de diálogo de propiedades para el FOE de alarmas Análogas están descritas en la página que sigue.

Abra el cuadro de diálogo de propiedades (right- clic en el FOE y pulse propiedades). Luego pulse la etiqueta **General** del cuadro de diálogo **Wonderware Analog Alarm Control Properties** para entrara información básica acerca de la alarma.

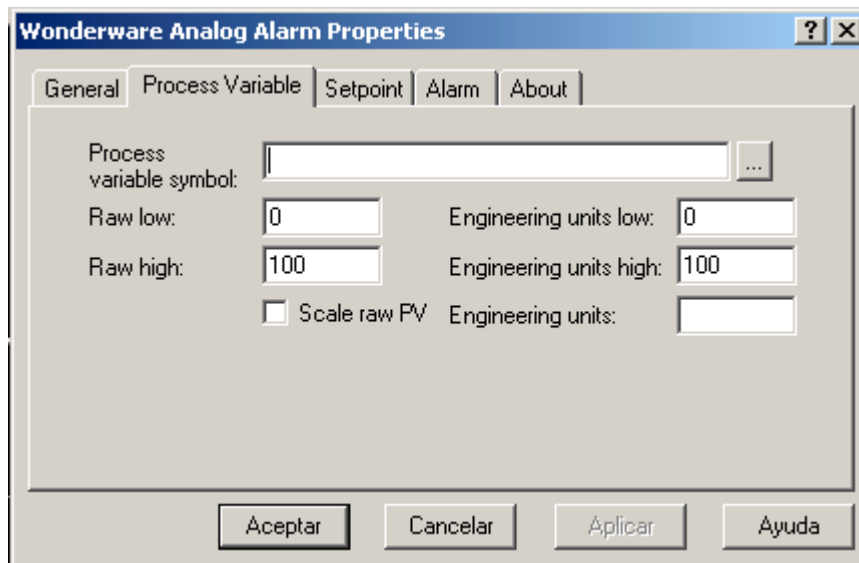


Etiqueta General de Propiedades de Alarma Análoga

Botón / Campo	Símbolo	Descripción
Name	Name	Campo de solo lectura que muestra el nombre de la alarma análoga. El nombre también aparece sobre la ventana principal de alarma análoga.
Description	Description	Opcional. Le da a la alarma análoga una descripción significativa.

Configurando Parámetros de Variable del proceso

Abra el cuadro de diálogo de propiedades (haga clic derecho en el FOE y pulse propiedades). Luego pulse en la etiqueta general **Process Variable** de el cuadro de diálogo **Wonderware Analog Alarm Control properties** para entrara información básica acerca de la alarma.



Etiqueta de Variable de Proceso

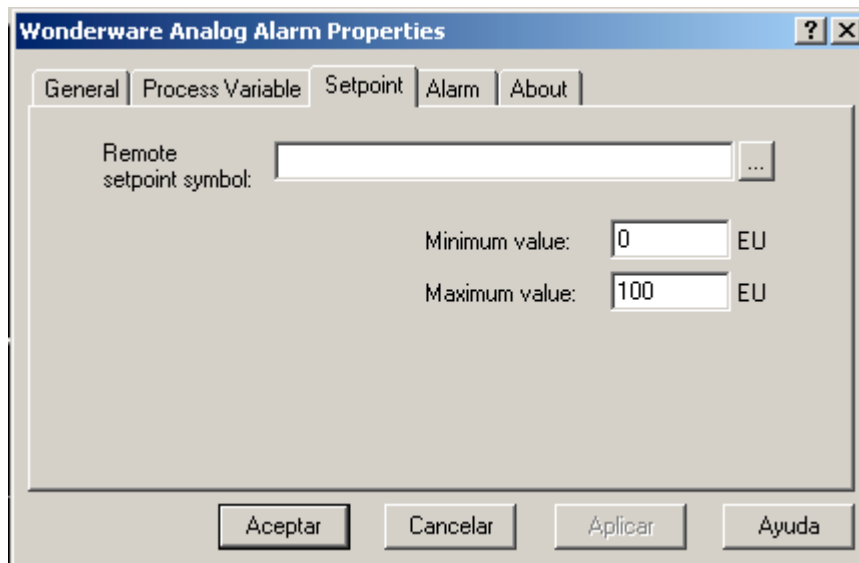
Botón / Campo	Símbolo	Descripción
Process Variable Symbol	PvName	<p>Ingrese el nombre del símbolo de la variable del proceso (o variable del proceso Raw sí el cuadro de diálogo Scale Raw PV es revisado) para el FOE a monitorear. Usted puede ubicar el cursor en el campo y hacer doble clic para abrir el Symbol Manager.</p> <p>Si no ingresa un nombre en este campo, el FOE compara el valor introducido en el campo Engineering Units Low con el valor setpoint.</p>

Botón / Campo	Símbolo	Descripción
---------------	---------	-------------

Raw low	PvRawLo	Si revisa el cuadro Scale Raw Pv , ingrese el valor mas bajo para la variable del proceso Raw. El valor alto debe ser mayor que el valor bajo.
RawHigh ¹	PvRawHi	Si revisa el cuadro Scale Raw Pv , ingrese el valor alto para la variable del proceso Raw. El valor alto debe ser mayor que el valor bajo.
Raw Low ¹	PvRawLo	Si usted revisa el cuadro Scale Raw Pv , ingrese el valor mas bajo para la variable del proceso Raw.
Raw High ¹	PvRawHi	Si usted revisa el cuadro Scale Raw Pv , ingrese el valor alto para la variable del proceso Raw.
Scale Raw Pv	ScaleRawPv	Revise el cuadro Scale Raw Pv para calcular la variable del proceso desde la entrada de la variable del proceso.
Engineering Units Low ²	PvEuLo	Entre el valor bajo (valor REAL) para la variable del proceso en unidades de diseño. El valor alto debe ser mayor que el valor bajo.
Engineering Units High ²	PvEuHi	Entre el valor alto (valor REAL) para la variable del proceso en unidades de diseño. El valor alto debe ser mayor que el valor bajo.
Engineering Units	PvEngUnits	Entra las unidades de diseño. Este es un texto con una cadena de caracteres ASCII que representa la unidad de medida, esto es, grados centígrados, PSI, Libra , etc.

Configurando los parámetros del Setpoint Objetivo

Abra el cuadro de diálogo de propiedades (haga clic derecho en el FOE y pulse propiedades). Luego pulse la etiqueta general **Setpoint** del cuadro de diálogo **Wonderware Analog Alarm Control properties** para ingresar los parámetros para el setpoint a alcanzar.



Propiedades de la Etiqueta Setpoint

Campo/

Cuadro de verificación	Símbolo	Descripción
Remote Setpoint Symbol	RemoteSpName	Ingrese el nombre del símbolo del setpoint para el FOE a monitorear. Puede ubicar el cursor en el campo y hacer doble clic para abrir el Symbol Manager. Si usted no ingresa un nombre en este campo, el FOE compara la variable del proceso con el setpoint mínimo (el valor introducido en el campo Minimum Value).
Minimum Value	SpLo	Ingrese el valor mínimo (valor REAL) para

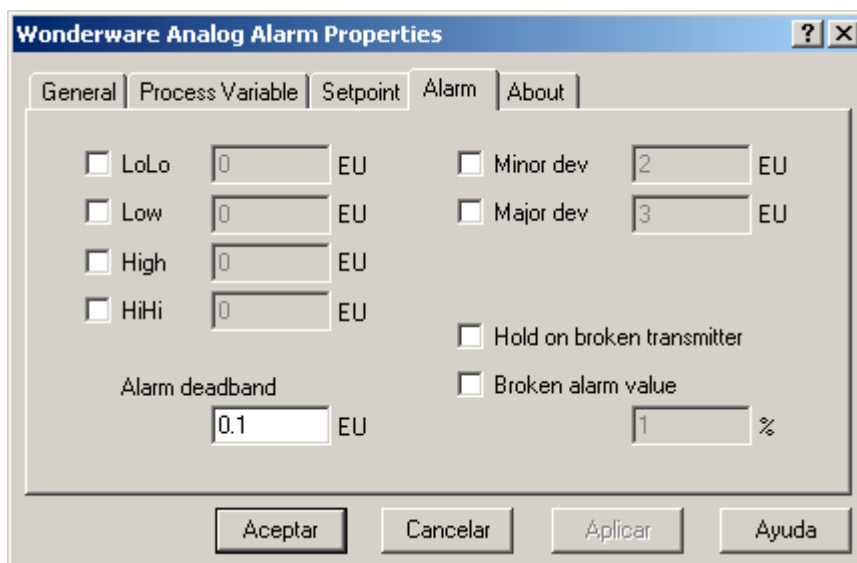
el setpoint. El valor mínimo debe ser mayor o igual al valor bajo usado por la variable del proceso, y debe ser menor que el valor máximo del setpoint.

Maximum Value SpHi

Entre el valor máximo (valor REAL) para el setpoint. El valor máximo debe ser menor o igual al valor mas alto usado por la variable del proceso, y debe ser mayor que el valor mínimo del setpoint.

Configurando Alarmas

Abra el cuadro de diálogo de propiedades (haga clic derecho en el FOE y pulse Properties). Luego pulse la etiqueta **Alarms** del cuadro de diálogo **Wonderware Analog Alarm Control Properties** para ingresar los parámetros de entrada. Ver **“Alarmas y desviaciones PV”** (-) por un ejemplo de la relación de las alarmas y desviaciones para el setpoint y los límites de la variable del proceso.



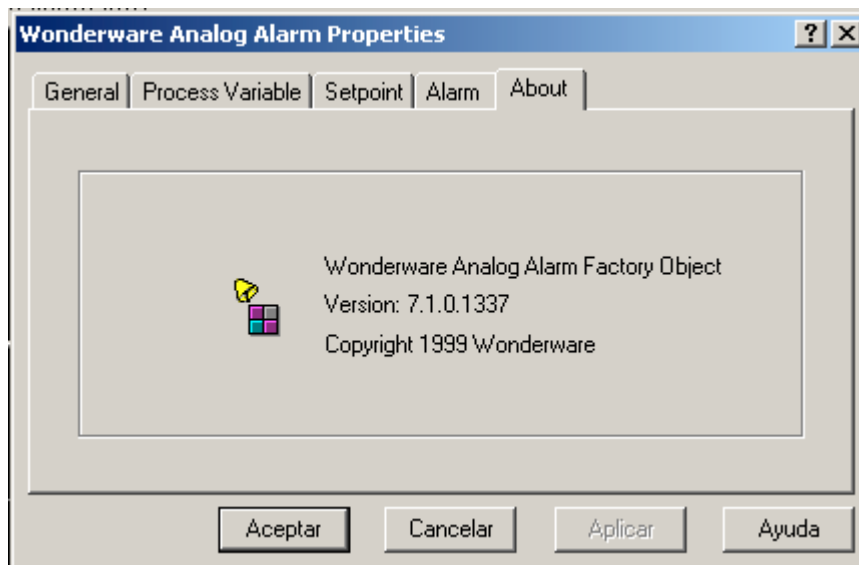
Propiedades de Alarma

Cuadro de verificación	Símbolo	Descripción
LoLo checkbox ¹	MonLoLoAlarm	Revise el cuadro LoLo para habilitar la alarma low-low.
LoLo Field	LoLoAlarm	Ingrese el valor (valor REAL menor o igual que el valor bajo) para la alarma low-low.
Low Checkbox ¹	MonLowAlarm	Revise el cuadro Low para habilitar la alarma Low.
Low Field	LowAlarm	Entre el valor (valor REAL menor que el valor alto) para la alarma baja.
High Checkbox ¹	MonHighAlarm	Revise el cuadro High para habilitar la alarma alta.
High Field	HighAlarm	Ingrese el valor (valor REAL menor que el valor high-high) para la alarma alta.
HiHi Checkbox ¹	MonHiHiAlarm	Revise el cuadro HI para habilitar la alarma high-high.
HiHi Field	HiHiAlarm	Ingrese el valor (valor REAL mayor o igual al valor alto) para la alarma high-high.
Alarm deadband	AlarmDeadband	Entre el valor (valor REAL dentro del espacio de diseño) dentro de las unidades de diseño. Este proporciona histéresis sobre alarmas para prevenir vibraciones cuando la variable del proceso se aproxima al límite.
Minor Deviation Alarm Checkbox ¹	MonMinorDevAlarm	Revise el cuadro Minor Dev para habilitar la alarma de desviación menor.

Cuadro de verificación	Símbolo	Descripción
	MinorDevAlarm	Entre el valor (valor REAL dentro del espacio de diseño, y menor que la alarma de desviación mayor) dentro de las unidades de diseño. Esto es el valor máximo del error de disparo de la alarma de desviación menor.
Major Deviation Alarm Checkbox ¹	MonMajorDevAlarm	Revise el cuadro Minor Dev para habilitar la alarma de desviación mayor.
Minor Deviation Alarm Field	MajorDevAlarm	Entre el valor (valor REAL dentro del espacio de diseño, y mayor que la alarma de desviación menor) dentro de las unidades de diseño. Esto es el valor máximo del error de disparo de la alarma de desviación mayor.
Hold on Broken Transmitter Checkbox ¹	HoldPvOnBrokenTx	Revise el cuadro Hold on Broken Transmitter para mantener la variable del proceso, pero no la variable del proceso Raw, en el valor anteriormente aceptable cuando recientemente valores calculados estén fuera del rango especificado en el campo Broken Alarm .
Broken Alarm Value Checkbox ¹	MonBrokenTx	Revise el cuadro Broken Alarm Value para monitorear la alarma de descarga interrumpida.
Broken Alarm Value Field	BrokenTxValue	Ingrese un valor (valor REAL, 0 – 100%) para representar el punto de alarma por encima y por debajo del alcance (span) de la variable del proceso Raw que dispara la alarma de descarga Interrumpida.

Chequeando la versión del FOE

Abra el cuadro de diálogo de propiedades (haga clic derecho en el FOE y pulse **Properties**). Luego pulse la etiqueta **About** del cuadro de diálogo **Wonderware Analog Alarm Properties** para revisar la versión del FOE.



SÍMBOLOS Y MÉTODOS DE ALARMAS ANÁLOGAS

La siguiente tabla muestra todos los símbolos que son usados por el FOE de alarma análoga.

Símbolos usados por el FOE de Alarma Análoga

Símbolo	Descripción	Leer / escribir
AlarmDeadband ¹	Valor de histéresis de alarma (unidades de diseño).	RW REAL
BrokenTx ²	Variable de proceso esta en el estado de alarma de transmisor.	RO BOLEAN

Símbolo	Descripción	Leer / escribir
BrokenTxValue ¹	Valor fraccionado del espacio de la variable de proceso usado para determinar si la variable del proceso esta en estado de descarga interrumpida.	RW REAL
Description ¹	Descripción del FOE.	RW STRING
Error ²	SP – PV (Setpoint--Variable de Proceso).	RO REAL
HighAlarm ¹	Alarma alta (unidades de diseño).	RW REAL
HiHiAlarm ¹	Alarma high – high (unidades de diseño).	RW REAL
HoldPvOnBrokenTx ¹	Mantiene la variable del proceso en el valor aceptable anterior en el evento que una descarga se interrumpa.	RW BOOLEAN
InHighAlarm ²	PV esta debajo de la alarma alta.	RO BOOLEAN
InHiHiAlarm ²	PV esta debajo de la alarma high-high.	RO BOOLEAN
InLoLoAlarm ²	PV esta debajo de la alarma low-low.	RO BOOLEAN
InLowAlarm ²	PV esta debajo de la alarma low.	RO BOOLEAN
InMajorDevAlarm ²	Valor absoluto de Error que esta debajo de MajorDevAlarm.	RO BOOLEAN
ImMinorDevAlarm ²	Valor absoluto de Error que esta debajo de MinorDevAlarm.	RO BOOLEAN
LoLoAlarm ¹	Alarma Low-Low (unidades de diseño).	RW REAL
LowAlarm ¹	Alarma baja (unidades de diseño).	RW REAL
MajorDevAlarm ¹	Alarma de desviación de error mayor (unidades de diseño).	RW REAL

Símbolo	Descripción	Leer / escribir
MinorDevAlarm ¹	Alarma de desviación de error menor (unidades de diseño).	RW REAL
Mode	Indica modos activos del FOE. La propiedad no esta disponible si el control ActiveX no es usado con InControl. 0= Descargado desde el Runtime Engine. 1= suspendido 2= pausa 3= Examinar 4= programa 5= ejecutar 6= Falla 7= Completo 8= Cargado al Runtime Engine.	RW INT
MonBrokenTx ¹	Supervisa el estado del transmisor roto.	RW BOOLEAN
MonHighAlarm ¹	Supervisa el estado de la alarma superior.	RW BOOLEAN
MonHiHiAlarm ¹	Supervisa el estado de la alarma superior alta.	RW BOLEAN
MonLoLoAlarm ¹	Supervisa el estado de la alarma inferior baja.	RW BOOLEAN
MonLowAlarm ¹	Supervisa el estado de la alarma inferior.	RW BOOLEAN
MonMajorDevAlarm	Monitorea el estado de la alarma de desviación principal.	RW BOLEAN
MonMinorDevAlarm	Monitorea el estado de la alarma de desviación secundaria.	RW BOOLEAN

Símbolo	Descripción	Leer / escribir
Name ²	Nombre	RO STRING
Pv ²	Variable de proceso en unidades de diseño.	RW REAL
PvEngUnits ¹	Unidades de diseño de la variable del proceso.	RW STRING
PvEuHi ¹	Rango superior de unidades de diseño.	RW REAL
PvEuLo ¹	Rango inferior de unidades de diseño.	RW REAL
PvName ²	Nombre de la variable del proceso.	RO STRING
PvRaw ²	Variable del proceso Rw desde el campo.	RW REAL
PvRawHi ¹	Valor máximo para la variable del proceso Raw.	RW REAL
PvRawLo ¹	Valor mínimo para la variable del proceso Raw.	RW REAL
RemoteSpName ²	Nombre del setpoint remoto.	RO STRING
ScaleRawPv ¹	Usa la variable del proceso para cálculo del valor de la variable del proceso. Cuando es TRUE, PvRaw es habilitado para escribir y Pv es de solo lectura.	RO BOOLEAN
Sp	Setpoint de Alarma Análoga	RW REAL
Sp	Setpoint PID	RW REAL
SpClampedHi ¹	SetpPoint esta cerca o sobre el valor SpHi.	RO BOOLEAN
SpClampedLo ¹	SetPoint esta cerca o por debajo del valor SpLo	RO BOOLEAN
SpHi	Valor máximo del setpoint (unidades de diseño).	RW REAL
SpLo	Valor mínimo del setpoint (unidades de diseño).	RW REAL

Métodos asociados con el FOE de Alarma Análoga

Símbolo	Descripción
DoControl ()	Este método actualiza la alarma análoga. Llama a cada análisis. Si el FOE de Alarma Análoga no es usado con InControl usted debe diseñar un código de programa para llamar al método DoControl para que el FOE pueda operar correctamente.

USO DEL FOE INFORMACIÓN DE PROYECTO

el FOE de información de proyectos le permite hacer lo siguiente:

- Enviar información de proyectos y nodos a un MI, así como a InTouch, o a un programa InControl.
- Reporta y elimina fallas desde un programa InControl.
- Pasos individuales en uno o más programas desde otro programa InControl.
- Devuelve los valores de variables retentivas y forzadas al disco duro.

Configurando el FOE de Información de Proyecto

No tiene el cuadro de diálogo de propiedades y no requiere configuración. Use las propiedades y métodos asociados con el FOE para enviar comandos e información.

Enviando Información de Nodos y proyectos

Dos propiedades están asociadas con este FOE: NodeName y ProjectName. Use el nombre del símbolo para estas propiedades, el cuál aparece en el Symbol Manager, para pasar nombres de proyectos y nodos a otro programa InControl o a un HMI.

Por ejemplo para mostrar el nombre de un proyecto (llamado 923) y el nodo donde este es ejecutado (llamado DYB2), en la Watch Window, adicione los siguientes símbolos a la Watch Window: ProjInfo2.ProjectName y ProjInfo2.NodeName, donde projInfo2 es el nombre del Project Information FOE. La siguiente figura muestra los símbolos como ellos aparecen en la Watch Window.

Type	Symbol	Value
Ab: STRING	ProjInfo2.NodeName	DYB2
Ab: STRING	ProjInfo2.ProjectName	\\DYB2\C:\PROGRAM FILES\FACTORYSUITE\INCONTROL\923

Realizando Comandos de ejecución

Dos métodos están asociados con el Project information FOE: ExecProgramCommand y ExecProjectCommand. Para ejecutar estos métodos, llámelos desde otro programa InControl (Texto Estructurado o SFC). Los valores de comandos usados por estos métodos están resumidos en la tabla de abajo.

Valores de Comandos de Información de proyectos

Método	Valores de Comando	Resultado de Comando
ExecProgramComand	6	Paso único a un programa
ExecProgramComand or ExecProgramComand	12	Elimina fallas (programa u objetos)
ExecProgramComand or ExecProgramComand	13	Reporta fallas (programa u objetos)
ExecProgramComand	19	Salva valores retentivos para un proyecto.

Reportando y Eliminando Fallas

Usted puede usar el método Project Information para reportar y limpiar fallas para un programa u objeto. Para un programa, use la siguiente sintaxis:

<FOE name> . ExecProgramCommand ("program name", <command>)

Para un proyecto, use la siguiente sintaxis:

<FOE name> . ExecProjectCommand (**<command>**)

Donde **<command>** equivale a 13 para reportar fallas, y **<command>** equivale a 12 para limpiar fallas.

Ejemplo 1: reportar fallas para un programa llamado STL1 usando el método de un Project Information FOE llamado ProjInfoA, entre la siguiente línea en un programa:

```
ProjInfo1.ExecProgramCommand("STL1",13);
```

Asegúrese de usar comillas alrededor del nombre del programa.

Ejemplo 2: Limpiar fallas para un proyecto llamado Bldg_A usando los métodos de un objeto Project Information llamado ProjInfoA, entre la siguiente línea en el programa:

```
ProjInfoA.ExecProjectCommand(12);
```

Programas de Pasos individuales

Puede usar los métodos Project Information para pasos individuales de uno o más programas. Use la siguiente sintaxis:

<FOE name> . ExecProgramCommand ("**program name**", 6)

Por ejemplo, para pasos individuales de tres programas, llamados STL1, STL2 y RLL5, usando los métodos de Project Information FOE llamado ProjInfo1, entre la siguiente línea en otro programa que este usando para arreglar su código:

```
ProjInfo1.ExecProgramCommand("STL1",6);
```

```
ProjInfo1.ExecProgramCommand("STL2",6);
```

```
ProjInfo1.ExecProgramCommand("RLL5",6);
```

Guardando Valores Retentivos y Forzados

Puede usar estos métodos para guardar los valores de variables retentivas y forzadas en un proyecto. Para una variable forzada, ambos el valor y el estado forzado están guardados en el disco duro. Use la siguiente sintaxis:

<FOE name> . ExecProjectCommand (19)

Ejemplo: Para guardar los valores de variables retentivas y variables que han sido forzadas en un proyecto, use los métodos de un Project Information FOE llamado ProjInfoA, ente la siguiente línea de códigos:

```
ProjInfoA.ExecProjectCommand(19);
```

Los valores son restaurados cuando cargue el proyecto o si tiene configurado el Runtime Engine para reiniciar automáticamente (Last, Pause, Run mode) después que un sistema reinicia. Para mas información, ver “**Restauración de proyectos Automáticamente**” en el capítulo “**ADMINISTRACION DE SISTEMAS INCONTROL**”.

EJECUCIÓN DE PROYECTOS

Este capítulo describe el ambiente de ejecución de InControl: opciones de menú, campos de pantalla, elementos de la barra de herramientas, etc.



CONTENIDO

- SELECCIÓN DE OPCIONES DE EJECUCIÓN
- CONEXIÓN AL RUNTIME ENGINE
- VALIDACIÓN Y DESCARGA
- EJECUCIÓN DE PROYECTOS Y PROGRAMAS
- DETENCIÓN DE LA EJECUCIÓN DE PROYECTOS Y DE PROGRAMA

ORDEN DE EJECUCIÓN DE PROYECTOS Y PROGRAMAS

Company:	N/A
SN:	N/A
Expires:	N/A
Locked To:	N/A

7.1.0.1337

SELECCIÓN DE OPCIONES DE EJECUCIÓN

Después del diseño de programas de aplicación dentro del ambiente de desarrollo InControl, ejecute los programas dentro de un ambiente de ejecución. Los programas atraviesan tres fases cuando ellos entran al ambiente de ejecución:

- **Validación** ---- un programa es revisado por errores de sintaxis y compilados.
- **Descarga**----- un programa es cargado en el Runtime Engine.
- **Ejecución**----- un programa es ejecutado por el Runtime Engine.











Usted puede validar, descargar, y correr un programa individual o un proyecto entero. Una unidad de hardware InControl soporta una petición de una Runtime Engine, el cual puede correr un proyecto a la vez. Sin embargo, puede descargar proyectos al Runtime Engine que son ejecutadas sobre múltiples unidades de hardware.

Para mas información acerca de múltiples proyectos, ver “**EJECUCIÓN DE MÚLTIPLES PROYECTOS**” en el capítulo “**ADMINISTRADOR DE SISTEMAS INCONTROL**”.

Puede seleccionar todas las opciones de ejecución desde la barra de menú o desde la barra de herramientas de ejecución, la cual aparece cuando se ha conectado al Runtime Engine. Este capítulo describe como usar estas herramientas basadas en selecciones que hace desde la barra de menú. Para evitar confusión, solo un método es descrito en este manual.

Comandos de la barra de herramientas Runtime

Comando del menú Runtime	Icono de la Barra de Herramienta	Descripción
Connect/ Disconnect		<p>Conecta el ambiente de desarrollo al Runtime Engine. El Engine corre continuamente como un servicio Windows NT, y si actualmente ejecuta un proyecto, depende de su modo de operación (Run, Stop, Pause, etc.).</p> <p>Cuando el Runtime Engine esta conectado, el icono se mantiene presionado y la nueva opción es “Desconectar”, la cual desconecta el ambiente de Desarrollo del Runtime Engine. Si usted cierra el ambiente de Desarrollo el Runtime Engine continua corriendo.</p>
Configure	n/a	<p>Muestra el cuadro de dialogo Offline Runtime Engine Properties si no esta conectado al Runtime Engine.</p> <p>Muestra el cuadro de dialogo Online Runtime Engine Properties si esta conectado al Runtime Engine.</p>
Report Faults	n/a	Muestra los programas fallidos y las faltas I/O, en la ventana de Salida y en el Wonderware Logger .
Clear Faults	n/a	Coloca los programas fallidos en el modo Pause , borra las faltas, y borra los bits de estado de error, asi como RTEngine.ScanOverrun .
Validate Project		Valida todos los programas en un proyecto. Todos los programas modificados son guardados en el disco duro.
Download Project ¹		Descarga todos los programas de un proyecto al Runtime Engine. Los programas modificados son guardados en el disco duro. Los programas son validados si es necesario.

Comando en el menú Runtime	Icono de la Barra de Herramienta	Descripción
Run Project ¹		Corre todos los programas de un proyecto. Los programas son validados y descargados si es necesario. Todos los programas modificados son guardados en el disco duro.
Pause		Interrumpe todos los programas que actualmente están corriendo por el Runtime Engine.
Single Scan		Ejecuta un Scan (análisis) sencillo del Runtime Engine. El I/O es actualizado, entonces todos los programas en un proyecto que actualmente son descargados al Runtime Engine ejecutan un análisis. <i>Ver el capítulo "EJECUCIÓN DE UN PROYECTO" para mas información.</i>
Stop		Detiene todos los programas en un proyecto que actualmente están siendo ejecutados. Los programas no son cargados desde la memoria. El I/O va al estado definido en la configuración para cada tarjeta I/O.
Validate Program		Valida el programa seleccionado. Si el programa fue modificado, es guardado en el disco duro.
Download Program		Descarga el programa seleccionado al Runtime Engine. Si el programa fue modificado, es guardado en el disco duro.
Run Program		Corre el programa seleccionado. Si el programa fue modificado, es guardado en el disco duro.
Pause Program		Interrumpe un programa que este siendo ejecutado. El I/O continua su actualización.
Single Scan Program		Ejecuta un análisis sencillo del programa. Solamente puede hacerse mientras el programa este detenido.
Stop Program		Detiene y descarga un programa desde la memoria. Otros programas en el proyecto y los puntos I/O no son afectados.

CONEXIÓN AL RUNTIME ENGINE


El ambiente de desarrollo debe ser conectado al Runtime Engine antes de que descargue o corra un programa o proyecto.

Nota: El Runtime Engine es un servicio Windows NT y empieza automáticamente cuando enciende la unidad de hardware. Para mas información acerca de cambiar el servicio automático de inicio del Runtime Engine, ver la ***Guía de Administración de sistemas de FactorySuite***.

El Runtime Engine al cual usted se conecta puede ser localizado sobre cualquier nodo local o remoto. Usted especifica el nodo en el cuadro de diálogo **Runtime Engine Properties**, descrito en “**CONFIGURACIÓN DEL RUNTIME ENGINE**” en el capítulo “**ADMINISTRACIÓN DE SISTEMAS INCONTROL**”. Para revisar la versión de la plataforma actual, haga clic derecho en **RTEngine** en la ventana de Proyectos y seleccione **Properties**.

➤ **Para conectarse al Runtime Engine:**

- En el menú Runtime, pulse en **Connect**. El mensaje cambiará a **Disconnect**.

También puede pulsar **Connect Runtime Engine** en la barra de herramientas de ejecución  para conectarse al Runtime Engine.

Verificación del Nodo conectado

Después de conectarse al Runtime Engine, puede revisar la barra de estado y verificar el nodo (computador) en el cual el ambiente de desarrollo esta conectado. En la siguiente figura, el Runtime Engine esta corriendo en un nodo llamado **NDYB1**.



Figura 1. Verificación del nodo conectado.

Si ubica el cursor sobre el nombre del nodo, el ayudante muestra el nombre del proyecto que esta corriendo. En la siguiente figura, el proyecto llamado **957**, el cual fue desarrollado en un nodo remoto **YB2** y que esta corriendo sobre un nodo local.

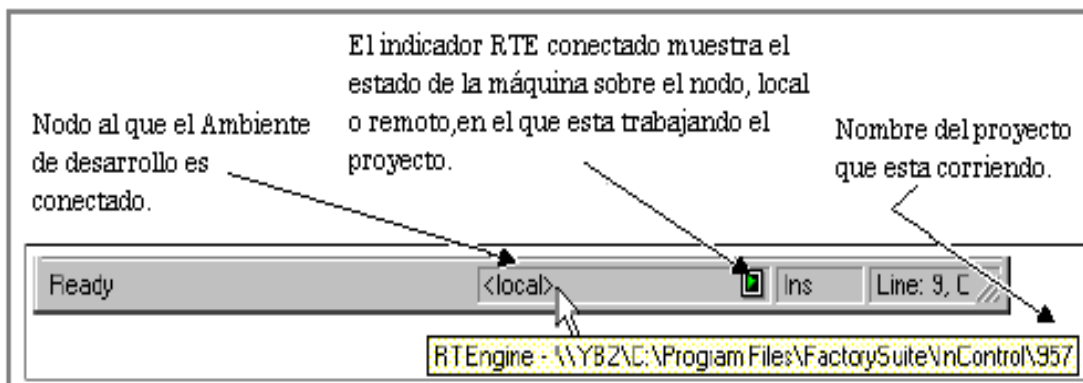


Figura 2. Verificación del Proyecto que sé esta ejecutando.

Uso del Monitor del Runtime Engine

El monitor del Runtime Engine esta asociado con el Runtime Engine sobre el computador local. Puede usar el monitor para enviar comandos y revisar el estado del Runtime Engine local. El icono del monitor del Runtime Engine de ejecución esta localizado en la barra de tareas, como se muestra abajo.

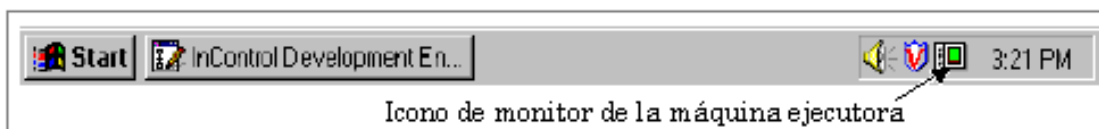


Figura 3. Verificación del nodo conectado.

No confunda el monitor del Runtime Engine para el nodo local con el indicador RTE conectado del Runtime Engine conectado al ambiente de desarrollo. El monitor del Runtime Engine, el cual esta ubicado en la barra de tareas, esta asociado solamente con el Runtime Engine local. El indicador RTE conectado, el cual esta localizado en la Barra de Estado de InControl, esta asociado con el Runtime Engine al cual el Ambiente de Desarrollo esta conectado. Este puede ser cualquier nodo local o remoto.

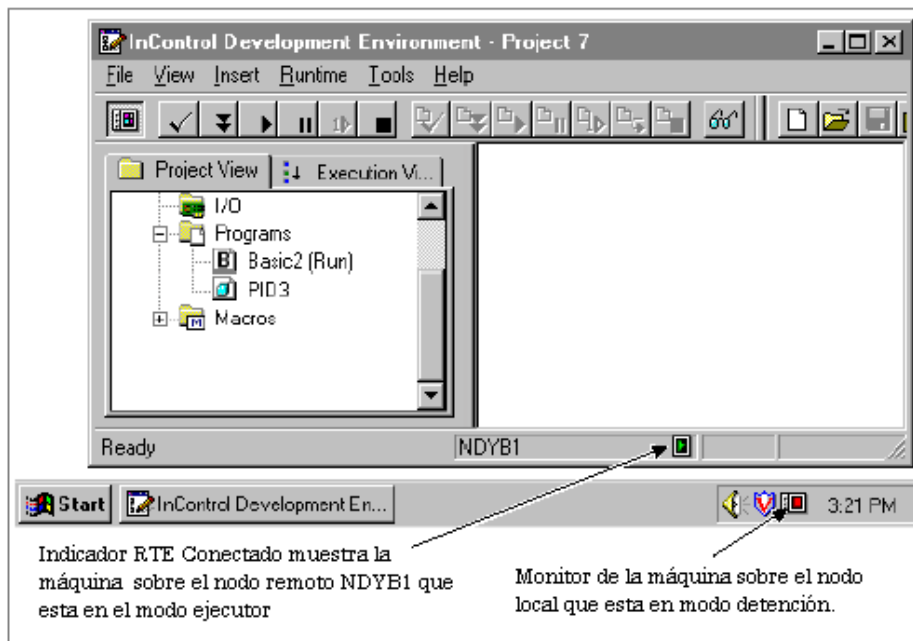


Figura 4. .


Uso del Icono del Runtime Engine


Este icono y el indicador RTE para el Runtime Engine el cuál el Ambiente de Desarrollo esta conectado los siguientes códigos de colores:

- **Rojo** (cuadrado), indica modo **Stop**.
- **Amarillo** (pausa), indica modo **Pause**.
- **Verde** (flecha), indica modo **Run**.

Cuando las variables están forzadas, este icono tiene el siguiente símbolo:



Con una señal de admiración amarilla InControl indica que un mensaje de error o advertencia ha sido enviado a la ventana Output y al Woderware Logger. Este icono también aparece cuando el programa entra al modo Fault o cuando el RLL MSGW o Texto estructurado MSGWND ejecuta funciones. 

InControl Indica una condición de falla con el siguiente símbolo: 

Uso d Comandos del Monitor del Runtime Engine

Puede ubicar el cursor encima del icono y hacer clic derecho. Aparecerá El siguiente menú.

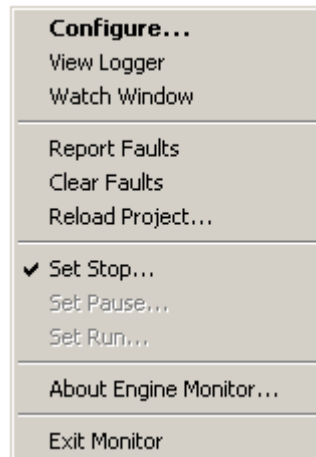


Figura 5. Comandos del monitor del Runtime Engine.

Comandos del Menú del Monitor del Runtime Engine

Opciones de Menú	Descripción
Configure	Muestra el cuadro de diálogo Online Engine Properties . Descrito en el “Fijando Tiempos de Análisis” en el capítulo “Administración de Sistemas InControl”.
View Logger	Muestra el Wonderware Logger, el cual guarda un registro de mensajes ejecutados. Estos mensajes también aparecen en la ventana Output. Ver “Revisando Wonderware Logger”.
Watch Windows	Muestra la ventana Watch. Esta es una versión stand – alone (puesto único) de la ventana Watch y no es necesariamente para abrir el Ambiente de Desarrollo y accederlo. Ver “Usando Stan- Alone de la ventana Watch”.
Report faults	Lista programas fallados y fallas I/O, en la ventana Output y el Wonderware Logger.
Clear Faults	Fija programas fallados a modo Pause, limpia fallas I/O, y limpia bits de estados de error de el Runtime Engine, así como RTEngine.Scan Overrun.
Reload Project	Carga el proyecto anterior que fue exitosamente descargado, incluyendo cualquier cambio.
Set Stop	Suspende todos los programas que están corriendo. El I/O va a el estado definido en la configuración para cada tablero I/O. Los programas son descargados desde la memoria. Para correrlos de nuevo, pulse Reload Project sobre el menú Runtime Engine Monitor y luego pulse Set Run.
Set Pause	Detiene todos los programas que estén corriendo. Para ejecutarlos puede pulsar Run sobre el menú Runtime Engine Monitor o Run Project/Program sobre el menú Runtime de la barra estándar.
Set Run	Corre todos los programas actualmente cargados en el Runtime Engine.
About Engine Monitor	Muestra el cuadro de diálogo About Runtime Engine .
Exit Monitor	Remueve el icono desde la pantalla y coloca el Runtime Engine en modo Stop.

Ejecución y desactivación del Monitor del Runtime Engine

El monitor del Runtime Engine aparece en el directorio **Startup** y empieza su ejecución automáticamente si el servicio del Runtime Engine esta corriendo cuando usted registra en el sistema operativo Windows NT. Si usted sale de este monitor, siga uno de los procedimientos para reiniciarlo.

➤ **Inicio del monitor desde la barra de menú:**

1. Conecte el Runtime Engine y pulse **Download Project** o **Run Project** sobre el menú Runtime.

➤ **Inicio del monitor desde la barra de tareas:**

1. Pulse **Start** sobre la barra de tareas, luego pulse Runtime Engine Monitor debajo de InControl.

Iniciando el servicio del Runtime Engine también inicia su monitor. Cuando sale del monitor, cualquier programa que este corriendo se detiene. El icono siempre es mostrado cuando los programas son cargados o ejecutados y cuando usted es registrado en el sistema de operativo Windows NT.

➤ **Desactivación del monitor del Runtime Engine:**

1. Haga clic derecho sobre el icono del monitor del Runtime Engine para mostrar el menú.
2. Haga clic en **Exit Monitor**. Si el programa está corriendo, se le pedirá confirmar que este se detuvo.

El icono es removido.

VALIDACIÓN Y DESCARGA

Usted puede validar un proyecto o programa, sin tener en cuenta el sistema operativo usado por la tarjeta del Runtime Engine. Note que puede cambiar la tarjeta del Runtime Engine antes de la validación (haga clic derecho al icono del Runtime Engine en la ventana de proyectos para mostrar el cuadro de diálogo **Properties**). Esto es útil cuando desarrolle y pruebe un proyecto sobre el computador usando el sistema operativo Windows NT pero teniendo la intención de descargar y correr el proyecto sobre un computador que use un sistema operativo diferente.

La operación del sistema es ligeramente diferente si usted valida y descarga un programa individual en lugar de un proyecto. Cuando usted trabaja en el nivel de proyecto, todos los programas en el proyecto son afectados. Cuando usted trabaja en el nivel programa, las acciones que usted hace en el programa individual actúa sobre aquel programa y probablemente afecte otros programas dentro del proyecto. Por consiguiente, proceda por validación y descarga con respecto a programas y proyectos que están descritos separadamente.

- **Validación de Proyectos** esta descrito en “**Validación de Proyectos**” (pagina 14-11).
- **Validación de Programas** esta descrito en “**Validación de un Programa Individual**” (pagina 14-).

Validación de Proyectos

Usted puede validar todos los programas dentro de un proyecto cuando quiera durante el desarrollo del proyecto. No es necesario que el ambiente de desarrollo este conectado al Runtime Engine cuando valide un proyecto. Esto le permite aprobar un proyecto que este desarrollando sobre un sistema Windows NT pero que intente correr sobre un computador usando otro sistema operativo.

Comúnmente, use la opción de Validación de Proyectos para revisar los errores de sintaxis en programas y para compilarlos sin tener que descargarlos o ejecutarlos.

➤ **Para validar un proyecto:**

1. Pulse en **Validate Project** sobre el menú Runtime. Aparecerá el cuadro de diálogo **Validate Project**.

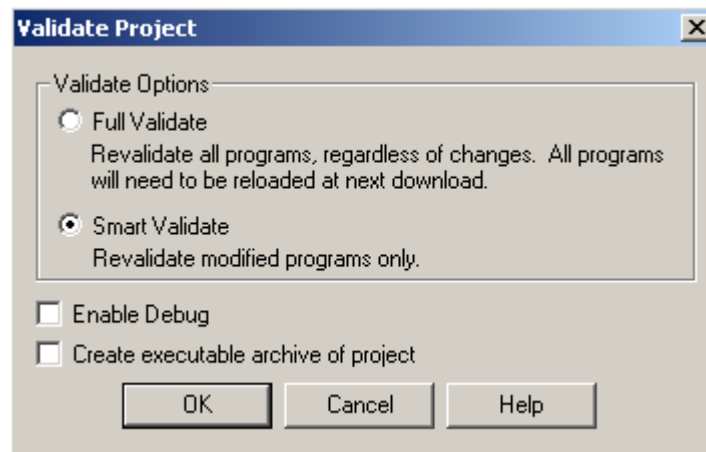


Figura 6. Cuadro de dialogo Validate Project.

2. Seleccione la opción de validación apropiada como se muestra a continuación. Los programas son guardados y validados como usted los necesite.
- Para validar todos los programas que hayan sido cambiados, seleccione **Full Validate**.
 - Para validar solamente los programas que hayan sido cambiados, seleccione **Smart Validate**.

Opcional. Verifique el cuadro **Enable Debug** para habilitar cualquier función **BREAK** de Texto estructurado que haya sido programada, o para usar la característica **Breakpoint** en programas de Texto Estructurado. La función **BREAK** ocasiona que un programa SFC o STL se detenga cuando el flujo del programa encuentre la orden **BREAK**. También debe verificar el cuadro **Enable Debug** si quiere Pasos simples en un programa. Note que el tiempo de ejecución es mas lento cuando la característica de depuración es habilitada. Verifique el cuadro **Create Executable Archive of Project** para crear un archivo que contenga el proyecto compilado. Puede usar el archivo como un registro o copiarlo en un nodo donde pueda correrlo.

3. Revise la ventana de salida para mensajes. Un mensaje del sistema reporta una validación suspendida y muestra la localización de los errores, como se muestra en la figura:

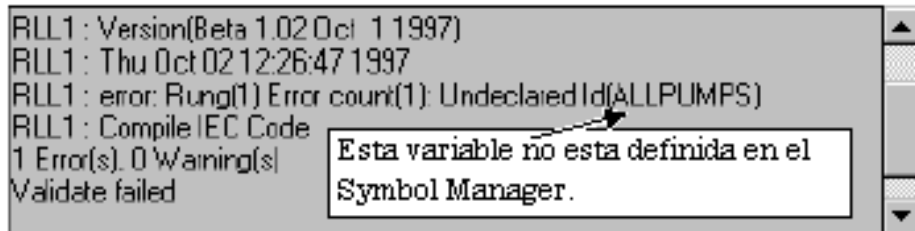


Figura 7. Suspensión de una validación y presentación de un error.

Si un programa falla en su validación, refiérase a los mensajes de error en la ventana de salida y haga los cambios apropiados en su programa de códigos. Note que todos los mensajes no son visibles, y puede necesitar desplazarse para ver mensajes adicionales.

Nota: La característica de **Smart Validation** no remueve ningún símbolo desde el Runtime Engine cuando ellos ya han sido descargados. Los símbolos son removidos solamente cuando usted fija un proyecto a el modo Stop o hace un **Full Restart** (reinicio completo). Por lo tanto los símbolos que usted eliminó en el Symbol Manager aún están visibles en la ventana Watch. Usted puede modificarlos en la ventana Watch aunque no tendrán efecto sobre la ejecución de su lógica.

Descarga de un proyecto

Cuando descarga un proyecto debe conectarse primero al Runtime Engine.

➤ **Para descargar un proyecto:**

1. Conéctese al Runtime Engine.
2. Pulse **Download Project** sobre el menú Runtime.

Aparecerá el cuadro de diálogo **Download Project**.

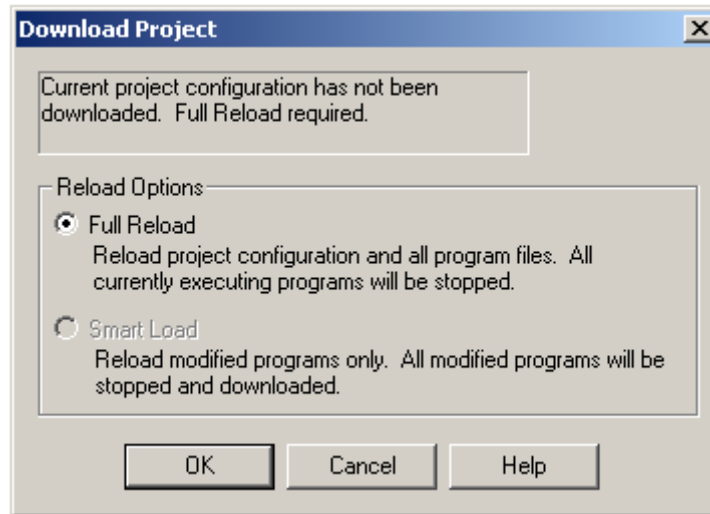
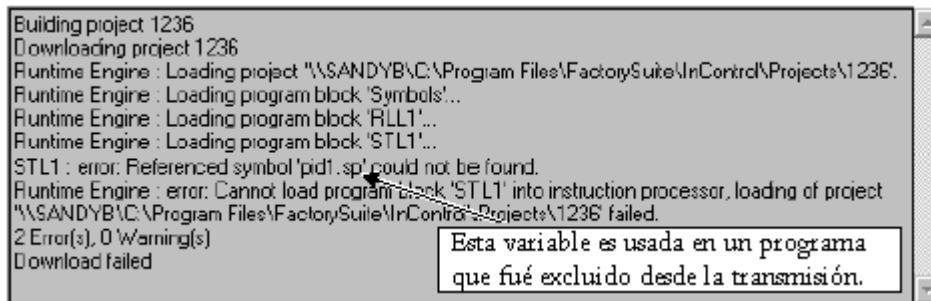


Figura 8. Cuadro de diálogo Download Project.

3. Seleccione la opción de descarga adecuada, como se describe abajo, y pulse OK. Los programas son guardados y validados, si es necesario, por último descargados.
- Las siguientes acciones requieren de **Full Reload**(recarga completa):
 - Descarga de una configuración de proyecto (configuración I/O o configuración del Runtime Engine) que no haya sido descargada antes.
 - Edición de una configuración.
 - Realización de modificaciones a variables, incluyendo tipos definidos por el usuario(user-defined), que ya han sido descargadas: cambio del tipo de dato, el elemento inferior de un arreglo, o la información de bits indexados.
 - Cambio de la configuración del Runtime Engine de fuera de línea.
 - Si los datos de configuración de proyecto ya han sido cargados durante una descarga previa y no quiere interrumpir los programas que están corriendo actualmente, seleccione **Smart Load**. Todos los programas nuevos son validados y descargados. Los programas que están corriendo, y que han sido modificados, son detenidos y reemplazados por la versión modificada; estos programas permanecen detenidos. Otros programas, los cuales están corriendo, son inalterados. Programas detenidos son validados y descargados.

- Si un programa esta corriendo y usted descarga una versión modificada del programa, el programa original es detenido, incluso si la nueva versión falla al descargar. Si la descarga es exitosa, el asterisco del nombre del programa en el título es removido.
 - Si usted no quiere que un programa no sea detenido después que una nueva versión es descargada, pulse Run Project, no Download Project, sobre el menú Runtime. El programa es descargado y fijado el modo Run si la descarga es exitosa.
4. Revise la ventana de salida para mensajes. La siguiente figura muestra el reporte de una descarga fallida de un sistema y muestra la localización del error.



```
Building project 1236
Downloading project 1236
Runtime Engine : Loading project "\SANDYB\C\Program Files\FactorySuite\InControl\Projects\1236".
Runtime Engine : Loading program block 'Symbols'...
Runtime Engine : Loading program block 'RLL1'...
Runtime Engine : Loading program block 'STL1'...
STL1 : error: Referenced symbol 'pid1.sp' could not be found.
Runtime Engine : error: Cannot load program block 'STL1' into instruction processor, loading of project
"\SANDYB\C\Program Files\FactorySuite\InControl\Projects\1236" failed.
2 Error(s), 0 Warning(s)
Download failed
```

Esta variable es usada en un programa que fué excluido desde la transmisión.

Figura 9. Falla de una Descarga y presentación de un error.

Si una descarga falla, refiérase al mensaje de error en la ventana de salida y haga los cambios apropiados en su programa de códigos. Note que todos los mensajes no pueden ser visibles, y usted puede deslizarse para ver los mensajes adicionales.

Nota: Si ha incluido cualquier código para probar o simular su proceso y el escribe a las variables de entrada de I/O, un mensaje de advertencia aparece. Usted puede cambiar esto para un mensaje de error o prevenir sus apariciones por completo. Ver “**VISUALIZACIÓN DE ADVERTENCIAS DEL COMPILADOR**” en el capítulo “**ADMINISTRACIÓN DEL SISTEMA DE INCONTROL**”.

Si descarga un proyecto usando la opción **Smart Load**, variables locales y globales que ya fueron descargadas retiene sus valores actuales y no son reiniciados. SFC's son restaurados.

Nota: La característica **Smart Load** no remueve ningún símbolo desde el Runtime Engine cuando ellos ya han sido descargados. Los símbolos son removidos solamente cuando usted fija un proyecto a el modo Stop o hace una reinicio completo (**Full Restart**). Por consiguiente, los símbolos que usted eliminó en el Symbol manager aún están visibles en la ventana Watch. Usted puede modificarlos en la ventana Watch aunque no tendrán efecto sobre la ejecución de su lógica.

Usted puede excluir un programa desde la descarga del proyecto. Haga clic derecho a un programa en la Project View y seleccione **Exclude**. Esto es útil para programas de simulación que normalmente no necesita excluir, pero el cual quiere almacenar con el proyecto.

Para fijar el programa del proyecto a modo Run, ver “**Ejecución de un programa**” (pagina 14-).

Validación de un Programa Individual

Puede validar un programa, sin tener en cuenta el sistema operativo usado por el Runtime Engine. Además, la versión del Runtime Engine (clic derecho al icono del Runtime Engine en la ventana de proyectos para visualizar el cuadro de diálogo **Properties**). Esto es útil cuando desarrolla y ensaya un programa sobre un computador usando el sistema operativo Windows NT pero intentó descargar y correr el programa sobre un computador que usa un sistema operativo diferente.

➤ **Para validar un programa:**

1. Pulse **Validate Program** en el menú Runtime. Aparecerá el cuadro de dialogo **Validate Program**.

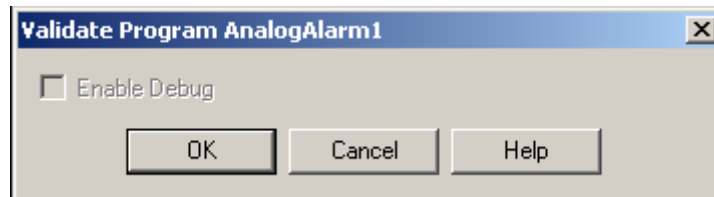


Figura 10. Cuadro de dialogo **Validate Program**.

2. Revise el cuadro **Enable Debug** para habilitar cualquier función BREAK de Texto estructurado que haya sido programado, o para usar la característica Breakpoint en programas de Texto Estructurado. La función BREAK ocasiona que un programa SFC o STL se detenga cuando un programa fluye encontrándose con BREAK. También debe revisar el cuadro **Enable Debug** si quiere single-step un programa. Note que el tiempo de ejecución es mas lento cuando la característica de depuración es habilitada.
3. Pulse **OK**. El programa es guardado, si es necesario, y validado.
4. Revise la ventana de salida para los mensajes. La siguiente figura muestra el reporte de una descarga fallida de un sistema y muestra la localización del error.

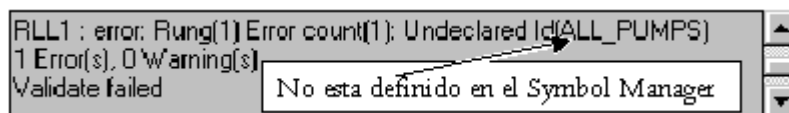


Figura 11. Descarga fallida de un sistema y localización de un error.

Si una validación falla, refiérase al mensaje de error en la ventana de salida y haga los cambios apropiados en su programa de códigos. Note que todos los mensajes no pueden ser visibles, y usted puede deslizarse para ver los mensajes adicionales.

Descarga de un Programa Individual

Cuando usted descarga un programa, primero debe conectarse al Runtime Engine. Cualquier driver I/O configurado también son descargados a lo largo con el programa.

➤ **Para descargar un programa:**

1. Conéctese al Runtime Engine.
2. Descargue cualquier otro POU's (funciones, bloques de funciones, otros programas, etc.) en el cual el programa depende. De otra manera el programa fallará en la descarga.
3. Pulse **Download Program** en el menú **Runtime**. Aparecerá el cuadro de dialogo **Download Program**.

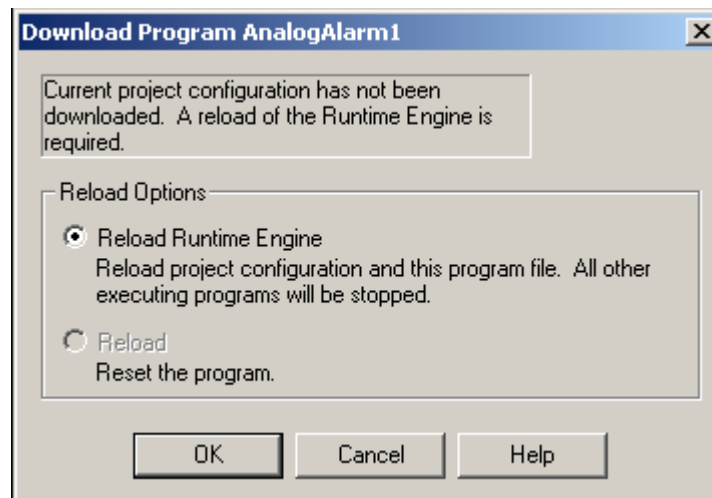


Figura 12. Cuadro de dialogo Download Program.

4. Seleccione la opción de descarga apropiada y pulse **OK**. El programa es guardado y validado, si se necesita, y descargado.

- Las siguientes acciones requieren que usted seleccione **Reload Runtime Engine**. Note que esta opción descarga todos los programas actualmente cargados y drivers I/O.
 - Descarga de la configuración de un proyecto (configuración I/O o configuración del Runtime Engine) que no ha sido descargada antes.
 - Edición de una configuración I/O.
 - Realización de modificaciones a variables, que ya han sido descargados: cambio del tipo de dato, el menor elemento de un arreglo, o información de bits indexados.
 - Cambio en la configuración del Runtime Engine.
- Si los datos de configuración de un proyecto (configuración I/O o configuración del Runtime Engine) ya han sido cargados durante una descarga previa y no quiere interrumpir los programas que están corriendo actualmente, seleccione **Reload**.
 - Si un programa está corriendo y usted descargo una versión modificada del programa, el programa original es descargado, inclusive si la nueva versión falla al descargar. Si la descarga es exitosa, el asterisco del nombre del programa en la barra de título es removida.
 - Si no quiere que el programa de ejecución sea interrumpido, pulse **Run Program**, no **Download Program**, en el menú **Runtime**. El programa es descargado e inicia su ejecución hasta completar la descarga.
- 5. Revise la ventana de salidas para ver los mensajes. La siguiente figura reporta una falla en la descarga y muestra su localización:

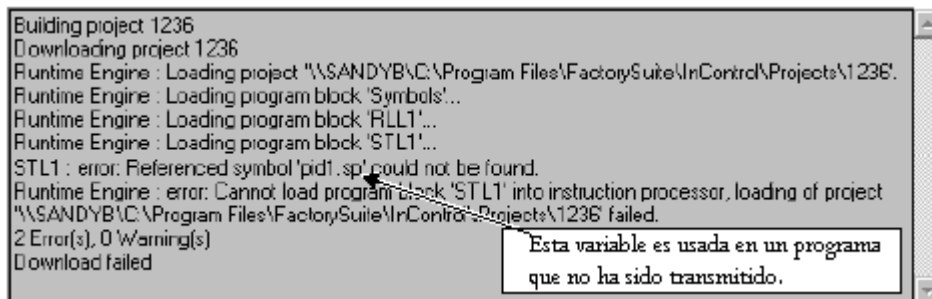


Figura 13. Reporte de una falla en la descarga de un programa.

Si una descarga de programa falla, revise el mensaje de error en la ventana de salida y haga los cambios apropiados en su código de programas.

Nota: Si ha incluido cualquier código para probar o simular su proceso y este escribe a variables de entrada I/O, un mensaje de advertencia aparece. Usted puede cambiar esto para mensaje de error o prevenir su aparición. Ver **“Visualización de Advertencias de Compilador”** en el capítulo **“ADMINISTRACIÓN DEL SISTEMA INCONTROL”**.

Si recargó un programa que fue descargado previamente, variables locales y globales que ya fueron descargadas, retienen sus valores actuales y no son reiniciadas.

Para ver un programa en modo Run, diríjase a **“EJECUCIÓN DE UN PROGRAMA INDIVIDUAL”**(pagina 14-).

EJECUCIÓN DE PROYECTOS Y PROGRAMAS

La operación del sistema es diferente si usted corre un proyecto en lugar de un programa individual. Por lo tanto, procedimientos para correr programas y proyectos son descritos separadamente.

Ejecución de un proyecto

Después que los programas en el proyecto están validados y descargados correctamente al Runtime Engine, son fijados en el modo Pause. Note que si usted pulsa **Run Project** en lugar de **Download Project**, los programas son fijados a modo Run después que son descargados.

InControl indica los modos o estados de los programas en la ventana de proyectos como se muestra en la siguiente figura (todos los programas están corriendo).

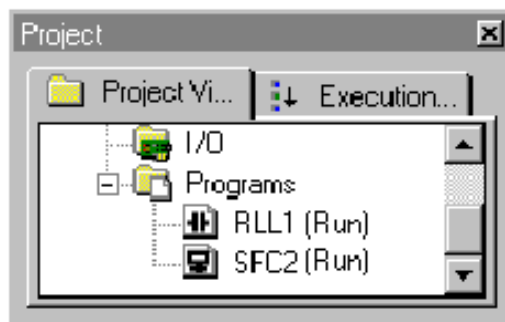


Figura 14. Estados de los programas en la ventana de proyectos.

Cuando un programa esta abierto en un editor, InControl también indica el modo del programa en la barra de título, así como se muestra en la siguiente figura.

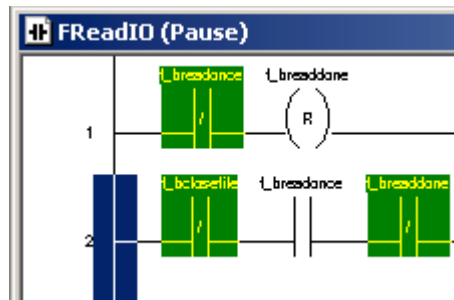


Figura 15. Estado de un programa desde la barra de título.

Advertencia La ejecución de un programa que no ha sido completamente revisado en un sistema conectado para equipos de campo puede causar operaciones impredecibles por los equipos. Operaciones impredecibles de los componentes de campo puede causar lesiones o muerte y/o daño a operadores. Es recomendable que revise su programa antes de correrlo en un sistema que controla un proceso de fábrica. Verifique la operación de cada elemento del programa o línea de código. Note que algunas condiciones de errores no son detectadas hasta que se corra el programa y puede desactivar todo o parte de su programa lógico.

➤ **Ejecución de un programa en un proyecto:**

1. Pulse **Run Project** en el menú Runtime. Aparecerá el cuadro de dialogo **Run Project**.

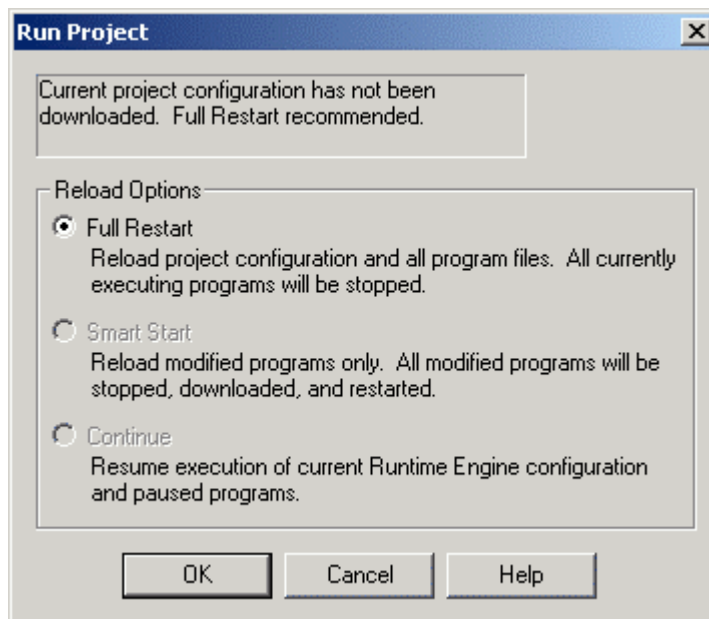


Figura 16. Cuadro de dialogo Run Project.

2. Seleccione la opción apropiada para la ejecución.
- Si la configuración del proyecto (configuration I/O o configuración del Runtime Engine) ha cambiado, seleccione **Full Restart**. Note que todos los programas son reiniciados, incluyendo SFCs que ha completado la ejecución.

- Para recargar solamente programas en un proyecto que usted ha modificado, seleccione **Smart Start**. Esta opción no tiene efecto en otros programas que estén corriendo, pero puede seleccionarlo solamente si no ha hecho cambios a la configuración I/O o a la configuración del Runtime Engine. Las variables en los programas no están inicializadas excepto por aquellas usadas por FOEs.
- Para ignorar cambios hechos en cualquier programa, seleccione **Continue**.

3. Pulse **OK** para confirmar. Todos los programas modificados son guardados, validados, descargados, si es necesario, y fijados a modo Run. Un asterisco en el nombre del programa en la ventana de proyectos o en la barra de título indica que el programa no ha sido guardado, y/o es diferente desde la copia que fue ejecutada en el Runtime Engine.

Cuando el programa en un proyecto cambia a modo Run, el indicador RTE, localizado en la barra de estado de InControl, se vuelve verde.

Nota: Solamente un proyecto puede correr en el Runtime Engine a la vez. Sin embargo, puede crear y editar programas en un proyecto mientras los programas en otro proyecto están corriendo. Además, usted puede correr proyectos en el Runtime Engine instalada en otro computador como se describe en “**Ejecución De Múltiples Proyectos**” en el capítulo “**ADMINISTRACIÓN DEL SISTEMA INCONTROL**”.

Ejecución de un Programa Individual

Después que un programa es exitosamente validado y descargado al Runtime Engine, el modo del programa es fijado a Pause. Note que si usted pulsa **Run Program** en lugar de **Download Program**, el programa es fijado a modo Run después de su descarga.

InControl indica el modo de un programa en la ventana de proyectos, así como se muestra en la siguiente figura: (FreadI0 esta en modo Run(ejecución),Rllfio esta en modo Pause).

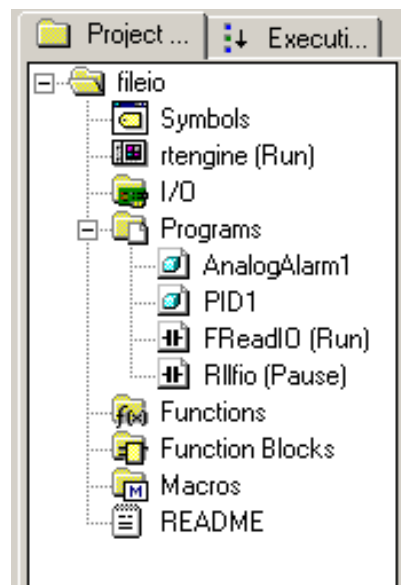


Figura 17. Estado de un programa en la ventana de proyectos.

Cuando un programa esta abierto en un editor, InControl también indica el modo del programa en la barra de título:



Figura 18. Estado de un programa desde la barra de título.

➤ **Para correr un programa:**

1. Seleccione el programa (pulse el nombre del programa en la ventana de proyectos) y pulse en **Run Program** en el menú **Runtime**. Aparecerá el cuadro de dialogo **Run Program**.

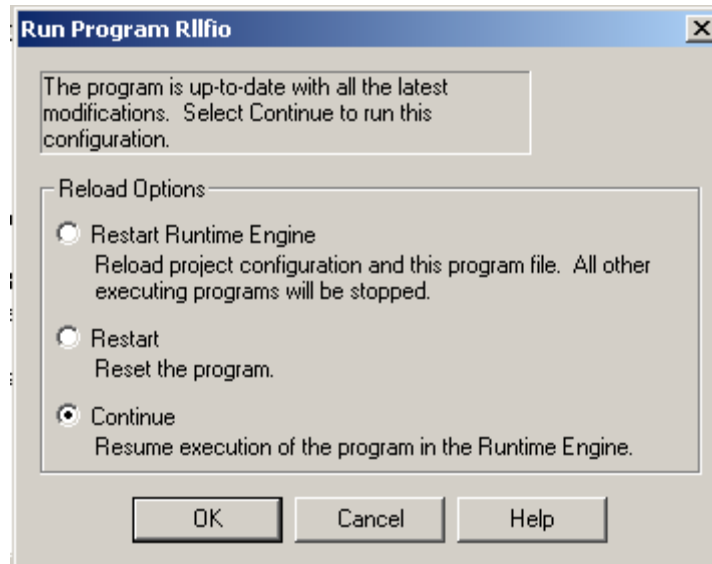


Figura 19. Cuadro de dialogo Run Program.

2. Seleccione la opción apropiada.
 - Si la configuración del proyecto se ha cambiado, seleccione **Restart Runtime Engine**. Esta opción detiene todos los programas que están corriendo y solamente restaura que esta siendo descargado, más I/O.
 - Si quiere que las variables de los programas mantengan sus valores actuales, seleccione **Restart**. Las variables en los programas no son inicializados excepto por aquellos usados por FOEs. Esta opción no afecta cualquier otro programa que este corriendo.
 - Para continuar ejecutando la versión del programa ya en el runtime engine, seleccione **Continue**. Cualquier modificación que fue hecha en el archivo de programa es guardada, pero no-descargada.
3. Pulse OK para confirmar.

Cuando el programa cambia al modo Run el icono del Runtime Engine, localizado en la barra de tareas, es verde para indicar que esta en modo Run.

Cualquier driver I/O configurado son también descargados al Runtime Engine y empiezan a ejecutarse.

Nota: Puede editar un programa, u otros programas, mientras el programa este corriendo. Además, puede correr proyectos sobre el Runtime Engine instalado en otro computador, como se describe en “**EJECUCIÓN DE MÚLTIPLES PROYECTOS**” en el capítulo “**ADMINISTRACIÓN DEL SISTEMA INCONTROL**”.


DETENCIÓN DE LA EJECUCIÓN DE PROYECTOS Y DE PROGRAMAS

Esta sección describe cómo detener la ejecución de un proyecto o programa individual.


Detención de la ejecución de un Proyecto

Cuando usted detiene un proyecto, todos los programas son descargados desde el Runtime Engine las puntas de salida I/O son deshabilitadas. El equipo I/O individual determina el estado actual de la salida.

➤ **Para detener la ejecución de un proyecto:**

- Pulse **Stop** en el menú **Runtime**.
- Pulse **Stop Runtime** en la barra de herramientas de Runtime. 
- Pulse **Set Stop** en el menú del icono del monitor del Runtime Engine. Aparecerá el cuadro de dialogo **Stop Runtime Engine**.

Cuando usted pulsa **OK** todos los programas son suspendidos. Debido a que ellos están descargados del Runtime Engine, usted debe descargarlos desde el ambiente de desarrollo otra vez antes que pueda correrlos. Pulse **Run Project** en el menú Runtime o en la barra de herramientas de Runtime.

También puede pulsar **Reload Project** en el menú del monitor del Runtime Engine. 

Nota: La opción **Set Run** en el menú para el icono de monitoreo del Runtime Engine solamente corre proyectos que han sido pausados, no proyectos que han sido suspendidos.

Las variables de los programas son congelados en sus valores anteriores cuando un programa se pone en modo Stop. Por ejemplo, la variable InCascade usada en el **FOE PID** podría ser TRUE inclusive cuando su lazo asociado no esta corriendo. Cuando usted está monitoreando una variable de programa, asegúrese de supervisar en el programa la variable de sistema Mode, debe verificar que el contenido de la variable sea válido.


InControl automáticamente fija el valor de la variable de sistema del Runtime Engine **RTEngine.Mode** a 1 cuando el Runtime Engine es suspendido. Usted puede cambiar el modo de un proyecto desde la ventana Watch u otro programa externo escribiendo uno de los siguientes valores a la variable de sistema **RTEngine.Mode**: 2(Pause), 3(Single Scan), 5(Run), o 6(Fault). Sin embargo, no puede escribir 1(Stop) o cualquier otro valor a **RTEngine.mode**.


Detención de la Ejecución de un Programa

Cuando suspende un programa, el programa es descargado desde el Runtime Engine. Los puntos I/O no son detectados.

➤ Para detener la ejecución de un programa:

Si el programa esta abierto en la ventana editor, selecciónelo.

Pulse **Stop program** en el menú Runtime, o **Stop Program** en la barra de herramientas Runtime 

Cuando usted pulsa **Ok** el programa es detenido. Debido a que el es descargado desde el Runtime Engine, usted debe descargarlo de nuevo antes que pueda correrlo. Pulse cualquiera, Run Program en el menú Runtime o pulse Run Program en la barra de herramientas Runtime: 

Nota: La opción Set Run en el menú del icono del monitor del Runtime Engine solamente corre proyectos que han sido pausados, no proyectos que han sido suspendidos.

También puede suspender un programa seleccionándolo en la **Project View** antes de pulsar **Stop program**. Si otro programa es abierto y seleccionado en la ventana de edición, este será suspendido.

Las variables de los programas le son congelados sus valores anteriores cuando el programa se fija en modo Stop. Por ejemplo, la variable InCascade usada en el FOE PID podría ser TRUE inclusive cuando su lazo asociado no esta corriendo. Cuando usted está monitoreando una variable de un programa, asegúrese de supervisar en el programa la variable de sistema Mode, debe verificar que el contenido de la variable sea válido.

El valor de la variable de sistema MODE en el programa es ajustado a 1 cuando el programa esta detenido.

ORDEN DE EJECUCIÓN DE PROYECTOS Y PROGRAMAS

Usted puede especificar el orden, dentro del análisis, en el cual los programas son ejecutados y asignarles un nivel de prioridad a cada programa: **Normal Scan** y **Low Priority**(prioridad baja).

Orden de los Programas en la Ventana de Ejecución

Puede especificar el orden de análisis en el cual los programas serán ejecutados si usted descarga los programas como un proyecto. Cuando usted seleccione la **Execution View** en la ventana de Proyectos, los programas son mostrados en el orden en que son ejecutados.

➤ **Para cambiar el orden de ejecución:**

1. Muestre la **Execution View**.
2. Pulse en el nombre del programa.
3. Arrastre el programa a otra localización en la lista de programas.

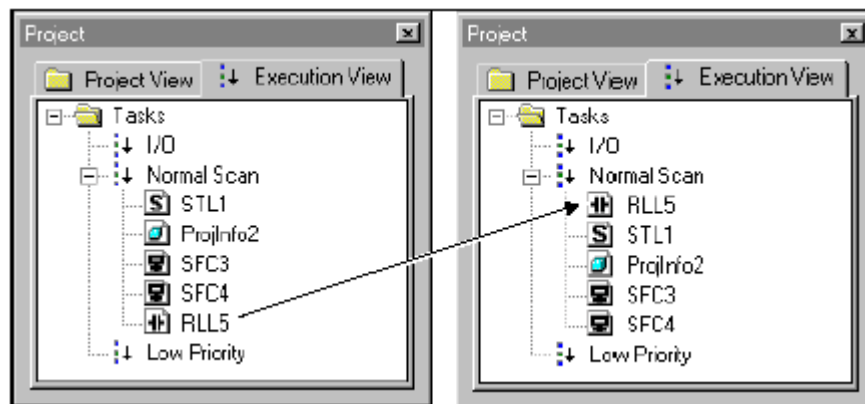


Figura 20. Orden de ejecución de los programas en la Execution View.

Tenga en cuenta estas sugerencias:

- Si descarga programas individualmente, en lugar de un proyecto, los programas son ejecutados en el orden en que usted los descarga. Si más tarde descarga el proyecto, el orden de ejecución es determinado por el orden de los programas en la Execution View.

- Si cambia el orden del programa en la Execution View, usted debe descargar el proyecto para el orden a ser descargado. Use la opción **Smart load** o **Smart Start**.
- Si usted descarga un proyecto y luego descarga una versión de un programa que esta corriendo actualmente en el proyecto, el orden de ejecución para el programa es preservado. Sin embargo, si usted crea un nuevo programa en el proyecto y luego descarga el nuevo programa, él va a lo último de la fila y corre después de otros programas en el proyecto.

Prioridad de los Programas en la Execution View

Usted puede especificar la prioridad en la ejecución de un programa: Normal Scan(análisis normal) o Low Priority(Prioridad baja). InControl corre un programa fijado en Low Priority solamente si hay tiempo restante después de que todas las otras tareas se han realizado. Durante el lapso de tiempo de prioridad baja de análisis, muchos programas son ejecutados como sea posible, con la ejecución de un programa reiniciado previamente después que un programa de prioridad baja ejecutado se ha completado.

➤ **Para cambiar la prioridad de ejecución:**

1. Muestre la Execution View.
2. Pulse en el nombre del programa.
3. Arrastre el programa a la nueva posición sobre la lista de tareas.

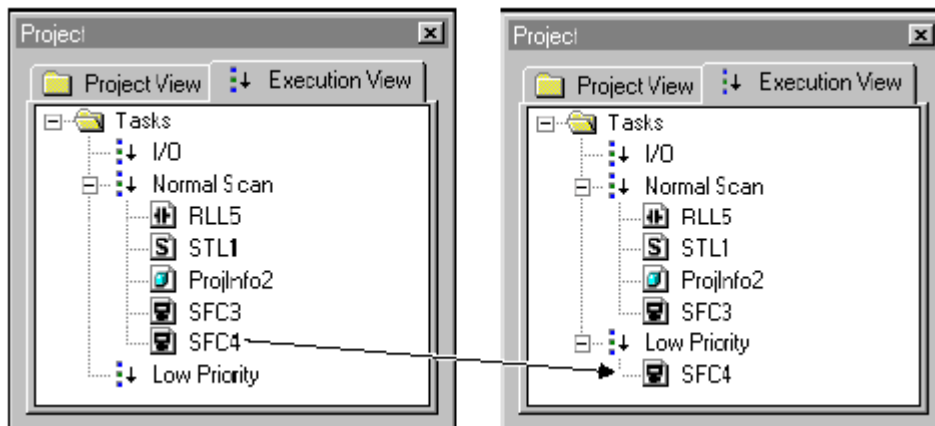


Figura 21. Prioridad de ejecución de los programas en la Execution View.

DEPURACIÓN DE UN PROGRAMA

Use la opción descrita en esta sección para revisar el programa que va ejecutar.


Verificación de la Barra de Estados

La barra de estados, presentada en “**Ventana Runtime**” en el capítulo “**AMBIENTE INCONTROL**”, muestra los estados del Runtime Engine en el que se está corriendo el proyecto, el nombre del proyecto descargado al Runtime Engine, el nodo para el cual el ambiente de desarrollo es conectado, así como otra información de programas.

Verificación del Modo del Programa

Durante el curso de su ejecución, un programa podría hacer una operación ilegal y entrar en modo Fault(falla).

Si un programa entra a modo de falla, aparece “**Fault**” en la barra de título del programa y en la Project View. InControl fija el Modo de la variable de sistema del programa en 6, escribe un mensaje en la ventana de salida y el Wonderware Logger, y superpone el siguiente símbolo al icono de monitoreo del Runtime

Engine: 

➤ Para reiniciar un programa en el modo Fault:

1. Elimine las fallas. En el menú del **Runtime Engine Monitor** pulse en **Clear Faults**. Esto fija al programa en modo pausa. Si otros programas están corriendo, continúan y el Runtime Engine mantiene su modo Run. También puede fijar el programa a modo Run o modo Pause para eliminar las fallas.
2. Fije el programa a modo Run.
3. Si el programa retorna a modo Fault, necesita revisar la lógica del programa y posiblemente la integridad de su hardware. Si cambia el programa, descárguelo y ejecútelo de nuevo.

Nota: InControl escribe un mensaje en la ventana de salida “Output” y en el Wonderware Logger cuando un programa entra a modo Fault.

Análisis Sencillo de un Proyecto y Programa

Esta opción le permite ejecutar un programa único o todos los programas en un proyecto para un análisis completo por el Runtime Engine. La ejecución sigue el horario que está descrito en “**Línea de tiempo del Runtime Engine**” del capítulo “**ADMINISTRACIÓN DEL SISTEMA INCONTROL**”. El orden de ejecución está basado según la Execution View. Un programa o proyecto debe estar en modo Pause antes que pueda ejecutar un análisis sencillo(Single Scan).

➤ **Para ejecutar un single scan:**

1. Fije el proyecto o programa a modo Pause.
2. Abra la ventana Watch, y adicione las variables que usted necesite monitorear en el programa.
3. Pulse **Single Scan** (o **Single Scan Program**) en el menú **Runtime**. Todos los programas detenidos entran al modo Run durante el próximo análisis del Runtime Engine, mantienen el modo Run durante el análisis, luego vuelve a entrar al modo pause.

Uso de Breakpoints(Puntos de Detención)

Usted puede detener la ejecución de un programa en una línea seleccionada del código escrita en el programa de Texto estructurado. El programa detiene su flujo antes de ejecutar la línea seleccionada. Para que el flujo continúe usted puede run(ejecutar), unpaused(reiniciar), Single Scan(análisis sencillo), o Single Step(Pasos simples) al programa. Puede ubicar el breakpoint (punto de detención o depuración) en múltiples líneas del código, y cuando el programa siga el flujo, se detiene en el próximo breakpoint.

Debe revisar el cuadro de diálogo **Enable Debug** en cualquiera de uno de los siguientes cuadros de diálogos para validar el programa antes que pueda poner un

breakpoint en el programa de texto estructurado. Verificando un cuadro de comprobación automáticamente revisa los otros.

- El cuadro de diálogo **Validate Project** o el cuadro de diálogo **Validate Program**. En el menú Runtime pulse **Validade Program** para mostrar este cuadro de diálogo.
- El cuadro de diálogo **Properties** para cada paso que quiera del programa. En la Project View haga clic derecho al programa y pulse **Properties** para mostrar este cuadro de diálogo.

Un breakpoint opera de la misma manera como una declaración BREAK de Texto Estructurado. Sin embargo, debe remover la declaración BREAK del programa o no comprobar uno de los cuadros **Enable Debug** cuando haya arreglado completamente el programa. Usted puede fácilmente adicionar o remover breakpoints en la ejecución pulsando **Toggle Breakpoint** y **Clear All Breakpoints** en la barra de herramientas **Runtime**. Note que el tiempo de ejecución es mas lento cuando la característica de depuración esta habilitada.

Nota: Los Breakpoints no están permitidos en funciones, procedimientos, o bloques de funciones.

Ejecución de un Programa Paso a Paso

La opción **Step Program** le permite ejecutar una línea lógica en un programa (RLL, SFC, o Texto Estructurado). Un programa debe estar en modo Pause antes que pueda usar la opción **Step**. Debe revisar el cuadro de diálogo **Enable debug** en cada uno de los siguientes cuadros para validación del programa antes que pueda hacer un “**Step**” a través de estos programas.

- Pulse **Validate Project** o **Validate Program** en el Runtime.
- En la **Project View**, haga clic derecho al programa y pulse en **Properties** para mostrar este cuadro.

Cada vez que usa la opción **Step** en un programa, los programas se ejecutan de la siguiente manera:

- Para un programa RLL, un escalón de lógica es ejecutado.
- Para un programa de texto estructurado, una línea de código es ejecutada.
- Para un programa SFC, una línea de código de cada **Step** activo es ejecutada. Todas las Actions(acciones) activas son ejecutadas completamente.
- Para los FOE's, El **Step Program** no es soportado.

Nota: En cualquier programa la opción Single-Stepping (ejecutado paso por paso) podría causar un orden de ejecución diferente de la ejecución normal del programa. Esto es debido a la ejecución continua de I/O, otros programas, y en el caso de SFCs, Actions.

➤ **Para hacer un Step(ejecutar una línea de código) en un programa:**

1. Seleccione el programa y fíjelo en modo Pause.
2. Abra la ventana Watch y adicione cualquier variable que necesite monitorear cuando el programa se ejecute.
3. Pulse **Step Program** en el menú **Runtime**. El programa entra a modo Run durante el próximo análisis del Runtime Engine, se mantiene en modo Run para un análisis, luego vuelve a entrar a modo Pause. Cualquier exploración I/O controlado por el programa es actualizada cuando al programa se le hace un Step.

Monitoreando variables de programas

La ventana Watch, mostrada en la siguiente figura, muestra las variables y sus estados en ejecución. Puede abrir la ventana para cada proyecto, y para cada ventana donde puede crear una o más tablas de variables para ser monitoreadas. Dentro de una tabla, puede monitorear variables desde uno o más programas, incluyendo local, global, y variables de sistemas, o combinaciones de cada una. No puede monitorear variables o parámetros usados en una función o procedimiento.

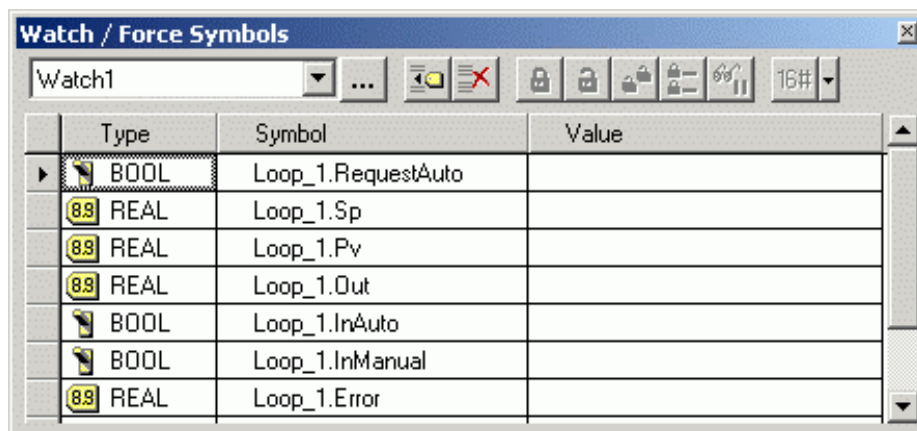



Figura 22. La ventana Watch.

Muestre o oculte la ventana Watch pulsando **Watch/ Force Variables** en el menú **View** o pulsando en el icono **Watch window** de la barra de herramientas **Runtime**. 

También puede correr una versión autónoma de la ventana Watch. No es necesario abrir el ambiente de desarrollo para acceder a ella. En el icono de monitoreo del Runtime Engine, pulse **Watch Window**. También esta disponible en el menú **Start** en la barra de tareas de Windows NT.

Adición de una Variable a la Ventana Watch

➤ **Para adicionar una variable en la ventana watch:**

1. Pulse **Add Symbol** en la barra de herramientas de la ventana Watch.



Figura 23. Adición de una variable en la ventana Watch.

Aparecerá el cuadro de dialogo **Symbol Manager**.

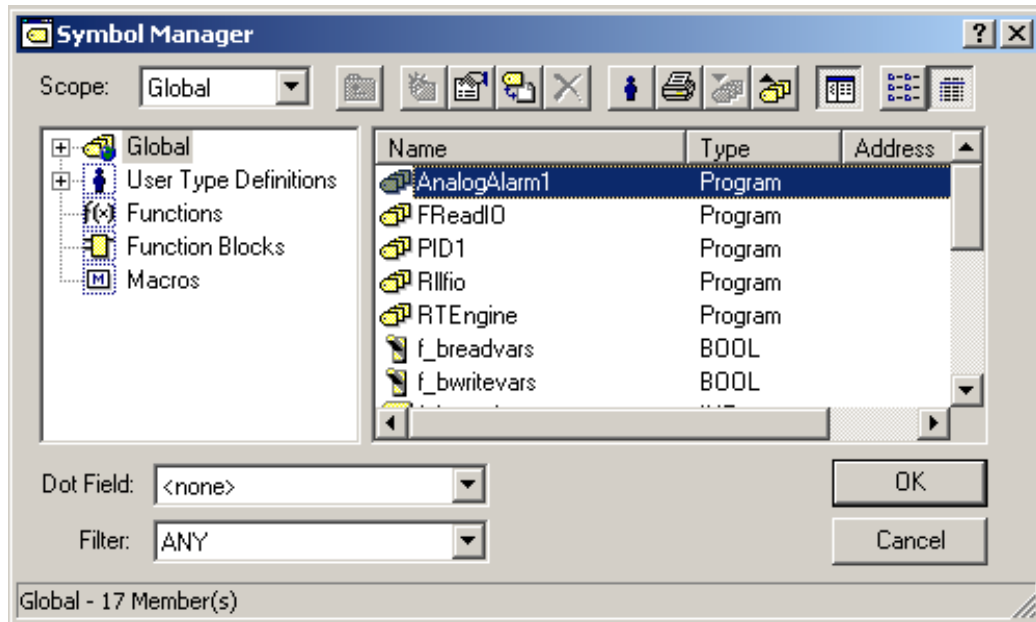


Figura 24. El Symbol Manager.

2. Pulse el nombre de una o más variables para adicionar. Puede hacer clic derecho en una variable para ver una imagen de solo lectura de sus propiedades.
3. Pulse **OK**. La variable seleccionada aparecerá en la ventana watch.

Para BYTES, WORDs, DWORDs, SINTs, REALs, y LREALs, puede escoger visualizar el valor en binario, octal, decimal, o formato hexadecimal. Pulse en el nombre de la variable y luego seleccione el formato desde la herramienta **Format** en la barra de herramientas de la ventana Watch.

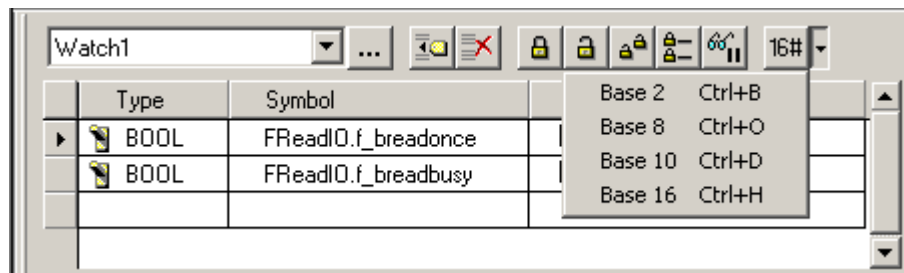


Figura 25. Selección del tipo de formato de las variables.

Si usted adiciona una variables que han sido definidas como una constante en la ventana Watch, la variable aparece como un valor oscurecido. Usted no puede modificar esta variable en la ventana Watch.

Si una variable tiene un valor inicial en el Symbol Manager. Por ejemplo, si el valor inicial es 2#1101_0110, entonces el formato predefinido en la ventana Watch es binario.

Adicionando Múltiples Variables a la Ventana Watch

En lugar de estar seleccionando variable individuales para mostrar en la ventana Watch, usted puede adicionar variables que estén relacionadas por grupo. Por ejemplo, usted puede adicionar todas las variables usadas en un programa, o todas las variables asociadas a una estructura.

- **Para mostrar todas las variables de programas en la ventana Watch:**
 1. Ubique el cursor en el campo **Symbol** y teclee el nombre del programa.
 2. Presione **Enter**. Las variables de programa aparecen en una estructura de árbol.

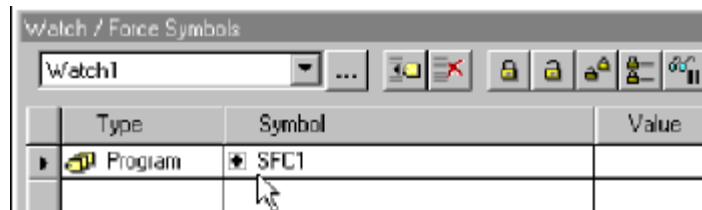


Figura 26. Estructura de las variables en la ventana Watch.

3. Extienda el árbol y todas las variables del programa aparecerán.

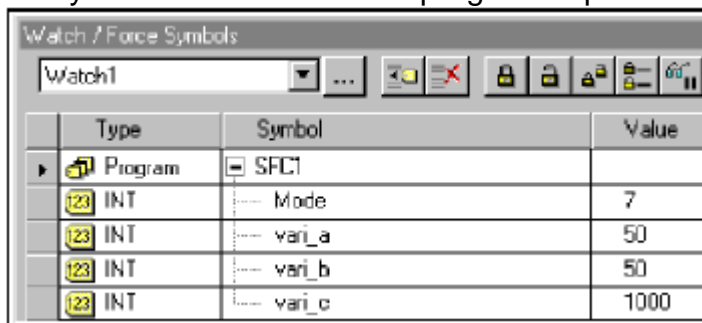


Figura 27. Todas las variables del programa en la ventana Watch.

Eliminación de una variable

➤ Para remover una variable desde la ventana Watch:

1. Pulse en el nombre de la variable.
2. Pulse en **Remove Symbol** en la barra de herramientas de la ventana Watch.



Figura 28. Eliminación de un símbolo desde la ventana Wacht.

Adición de una Tabla de Valores a la Ventana Watch

➤ Para adicionar una tabla a la ventana watch:

1. Pulse **Save Current table** en la barra de herramientas de la ventana Watch.



Figura 29. Adición de una tabla de valores a la ventana Wacht.

Aparecerá el cuadro de dialogo **Watch Windows Tables** mostrando la tabla predefinida llamada **Watch1**.

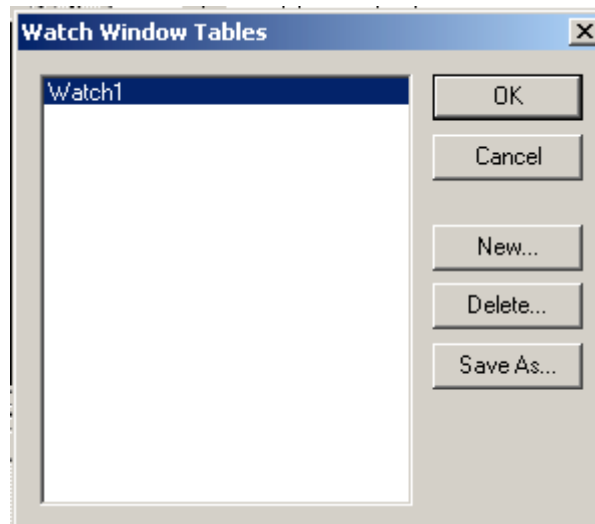


Figura 30. Cuadro Watch Windows Tables.

2. Pulse **New** para adicionar la tabla. Cuando aparezca el cuadro de diálogo **New** entre un nombre para la nueva tabla y pulse **OK**. Pulse **OK** de nuevo para retornar a la ventana Watch.

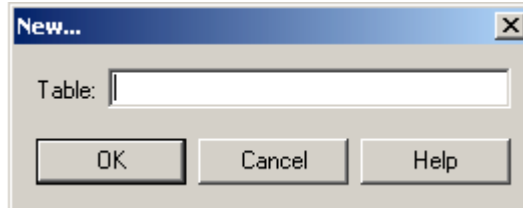


Figura 31. Definición de la nueva tabla de variables.

3. Adicione las variables a la ventana Watch.

Como una alternativa para crear las tablas dentro de la ventana Watch, usted puede crear un archivo ASCII con cualquier editor de texto.

➤ **Para crear una tabla con un editor de texto ASCII:**

1. Abra el editor de texto.
2. Entre el nombre de la variable usando los siguientes formatos.

Para una variable local, use el nombre del programa, un periodo, y luego el nombre de la variable. Para las variables locales en el ejemplo de abajo, los nombres de los programas son Bldg1 y B3.

Para una variable global, use solamente el nombre de la variable.

Para una variable del sistema de Runtime Engine en ejecución, use el nombre del sistema, RTEngine, seguido por un periodo y el nombre de la variable.

Para variables de usuario definido, use el nombre de la variable del tipo de dato de usuario definido, seguido por un periodo y el nombre del asociado.

La siguiente figura muestra un ejemplo.

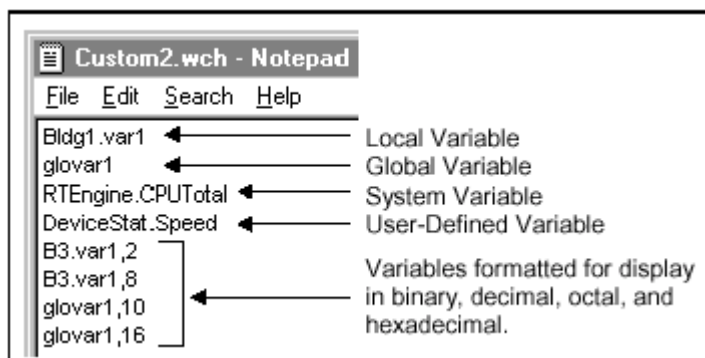


Figura 32. Creación de una tabla de variables desde un editor de texto.

3. Guarde el archivo con la extensión. **WCH** y ubíquelo en el directorio de proyecto. El nombre del archivo aparecerá en la lista de la tabla con el otro nombre de tabla.

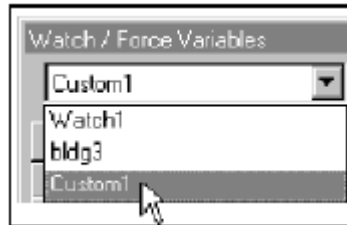


Figura 33. Presentación de la nueva tabla de variables.

Modificación y Forzado de una Variable

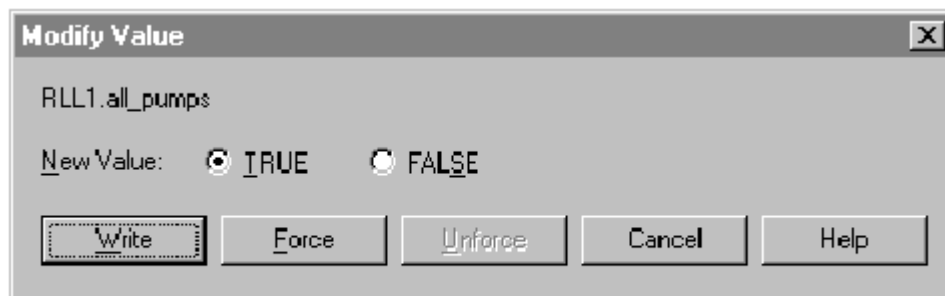
➤ **Para modificar o forzar una variable:**

1. Pulse el nombre de la variable.
2. Pulse en **Modify Value** en la barra de herramientas de la ventana watch.



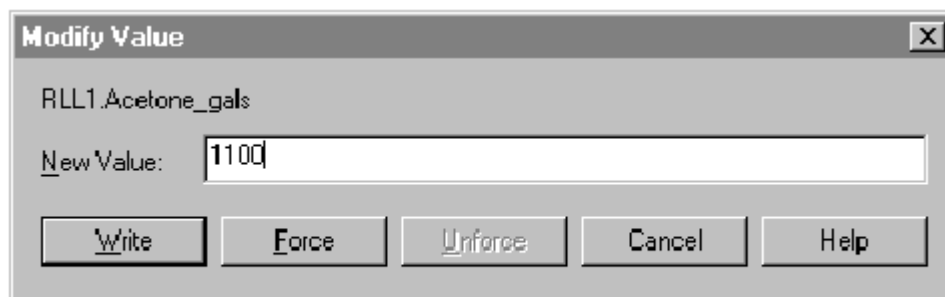
Figura 34. Opción de modificación del valor de una variable.

Aparecerá el cuadro de diálogo **Modify Value**.



Modifying Boolean Data Types

Figura 35. Cuadro de modificación del valor de una variable.




Modifying all other Data Types

Figura 36. Ingreso del nuevo valor de la variable.

3. Para escribir un nuevo valor a la variable, entre el valor (seleccione TRUE o FALSE para un tipo de dato Boleano) y pulse **Write**. El nuevo valor es escrito a la variable. Note que el valor puede cambiar cuando el programa corre.

Para forzar una variable a un valor, entre el valor (seleccione TRUE o FALSE para un tipo de dato Boleano) y pulse **Force**. La variable es forzada al nuevo valor, y el valor no hace cambios cuando el programa corre hasta que usted lo obligue a un nuevo valor o no le exiga.

Cuando usted entra un valor, puede separa caracteres con una guión (subraya). Para números hexadecimales y binarios, la ventana Watch separa cada cuatro caracteres con un guión para mejorar legibilidad.

Cuando las variables son forzadas, el icono del monitor del Runtime Engine tiene un símbolo con cerradura: 

La barra de Estado también muestra un símbolo cuando la variable es forzada:



Figura 37. Símbolo que indica que la variable esta forzada.

Tenga en cuenta los siguientes puntos adicionales:

- Si usted esfuerza un bit que ha sido puesto en un índice (descrito en “**Asignando un nombre a un bit**” en el capítulo “**DEFINICIÓN DE VARIABLES**”), la variable referida y todos los otros símbolos que están en índice con la misma variable, también son forzados. Por ejemplo, si Bit0 y Bit1 están indexados dentro de Word1, al esforzar Bit0 causa que ambos Bit1 y Word1 también sean forzados.
- Si usted adiciona una variable que ha sido definida como una constante para la ventana Watch, la variable aparece como un valor oscurecido. No puede modificar la variable en la ventana Watch.

- Si usted ha adicionado una variable para la ventana Watch y luego la borra desde el Symbol manager, un **Smart Download** o **Smart Restart** no remueve la variable desde la ventana Watch o el Runtime Engine en ejecución. Además, la ventana Watch continua para permitirle modificar la variable. Note que si su programa es afectado por el símbolo de tachadura, el fallará la validación hasta que usted modifique apropiadamente la lógica.

Nota: El nivel de Administrador o Ingeniero es requerido para forzar o modificar una variable. Un privilegio de seguridad de usuario aplica a el nodo local donde el usuario es registrado en cualquier nodo remoto para el cual el usuario puede conectarse.

Usted tiene la opción de retroceder el valor de una variable y su estado forzado al disco duro. InControl proporciona tres maneras por el cual usted puede especificar para el apoyo a ocurrir.

- Si el Runtime Engine se cierra durante una falla de energía, el valor de una variable forzada es copiada en el disco duro. Cuando el Runtime Engine reinicia, este valor (y el estado forzado) es restaurado a la variable. Note que los valores de cualquier variable retentiva también son guardados durante una falla de energía.
- Usted puede configurar InControl para gravar variables retentivas y forzadas al disco duro periódicamente. La frecuencia predefinida de cero desactiva esta característica. Usted puede cambiarlo en el cuadro de diálogo Runtime Engine Properties, descrito en “**Configuración Del tiempo de análisis**” en el capítulo “**ADMINISTRACIÓN DEL SISTEMA INCONTROL**”.
- Usted puede diseñar códigos en un programa para gravar el valor de variables retentivas y forzadas en demanda. Para mas información, ver “**USO DEL FOE DE INFORMACIÓN DE PROYECTOS**” en el capítulo “**USO DEL EDITOR DE OBJETOS DE FÁBRICA INCONTROL**”.

Los valores son restaurados cuando recarga el proyecto o si ha configurado el Runtime Engine para reiniciar automáticamente (Last, Pause, Run mode) después que un sistema rebote. Para mas información “**Restauración de proyectos Automáticamente**” en el capítulo “**ADMINISTRACIÓN DEL SISTEMA INCONTROL**”.

Valores validos para la variable de sistema Mode

Usted puede escribir valores a la variable de sistema **Mode** para un programa y, por consiguiente, controlar el modo del programa desde otro programa en el mismo proyecto. La siguiente tabla lista los valores válidos que pueden ser escritos a la variable de Modo sistema.

Valor	Acción que realiza la variable Mode
2	Fija el programa en modo Pause.
3	Análisis único a un programa. El programa primero debe ser pausado.
5	Fija el programa a modo Run.
6	Fija el programa a modo Fault. Asegúrese que este completamente informado de los procesos que se encuentren bajo el control del programa antes de la fijación del modo Fault.
7	Fije el modo del programa a Complete. Funcionalmente, los modos Pause y Complete son equivalentes.
1	Si usted está monitoreando la variable en Modo programa, los siguientes valores adicionales pueden ser leídos, aunque no pueden ser escritos a esta variable: <ul style="list-style-type: none"> 0 El programa es recargado desde el Runtime Engine. 1 El programa es suspendido. 4 El programa esta siendo cargado al Runtime Engine. 8 Para programas: El programa es cargado al Runtime Engine y es preparada para correr cuando es llamada por otro programa. Aplicable a funciones, bloques funcionales, y a FOEs que requiere funciones que llamen un método para correr. Note que el modo de estos POU's no cambia al correr, incluso después que son llamados.
2	Información acerca de la justificación de la falla un programa.

Advertencia: Cuando modifique modalidades de variables en un programa que use las variables para controlar otro programa, asegúrese de estar consciente de los cambios de modo que toman lugar en todos los programas. Cambios de modos de programas inesperados pueden causar lesiones o muerte y/o daños a equipos.

Ajustando la Tasa de Actualización

Fije la tasa de actualización para la ventana Watch en el cuadro de diálogo **Runtime Engine Configuration**, Online(en línea) u Offline(fuera de línea).

- Para fijar la tasa de actualización de la ventana Watch:
 1. Pulse **Configure** en el menú **Runtime**.
 2. Seleccione cualquier etiqueta **Online** o **Offline**. Los cambios hechos en la etiqueta **Online** son retenidos para sesiones vigentes solamente. Cambios hechos en la etiqueta **Offline** son guardados.
 3. Entre la tasa de actualización en el campo **Update Interval** como se muestra en la siguiente figura:

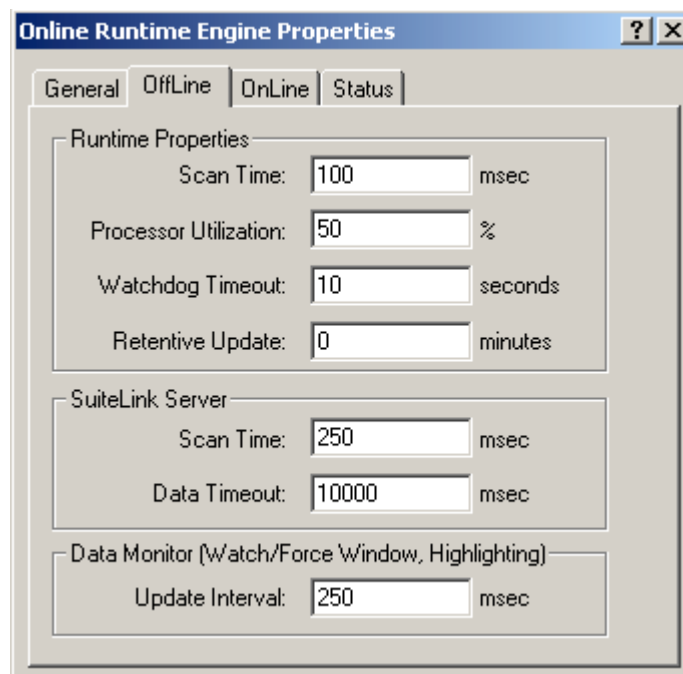


Figura 38. Ingreso de la tasa de actualización para la ventana Watch.

Pausa en la actualización de la ventana Watch

Esta puede pausar la actualización de la ventana Watch pulsando en la barra de herramientas de la ventana Watch en **Pause Watch Window**. Los valores son ensombrecidos cuando la ventana Watch es pausada.



Figura 39. Pausa en actualización de la ventana Watch.

Variables no Forzadas

Para deshacer variables forzadas, pulse en el nombre de la variable y pulse **Unforce Symbol**.

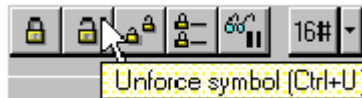


Figura 40. Eliminación de la opción de forzar una variable.

También puede dejar de forzar un variable desde el cuadro de diálogo **Modify Value**, descrita en “**Modificación y Forzado una Variable**”(pagina 14-).

Para desforzar todas las variables forzadas, incluyendo variables que han sido forzadas en otras tablas, use **Unforce All Symbol**.



Figura 41. Desforzando todas las variables forzadas.

Mostrando Variables Forzadas

Para visualizar todas las variables que han sido forzadas, incluyendo variables que han sido forzadas en otras tablas, pulse **Show All Forces**.



Figura 42. Mostrando todas las variables forzadas del programa.

Todas las variables en el proyecto que han sido forzadas son mostradas en la parte superior de la tabla que se muestre actualmente.

Note que la visualización de otras tablas y la terminación de InControl causa la tabla mostrada actualmente para ser salvada, incluyendo la lista de variables forzadas.

Uso del menú de la ventana Watch

Usted puede seleccionar todas las funciones desde un menú. Ubique el cursor dentro de la ventana Watch y haga clic derecho. Pulse la función apropiada para seleccionarlo.

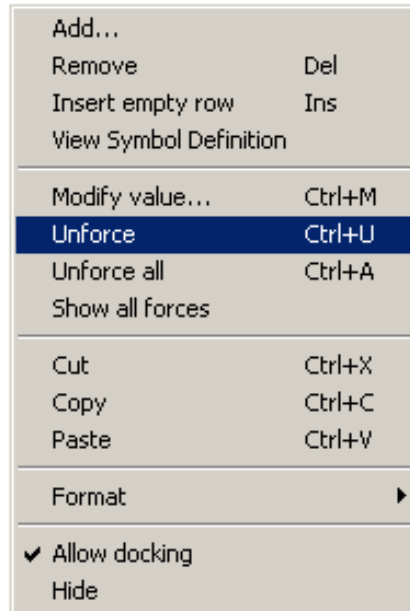



Figura 43. Menú de la ventana Watch.

Comandos de la Ventana Watch

Barra de menú	Icono	Descripción
Add		Adiciona símbolos a la ventana Watch.
Remove		Remueve símbolos desde la ventana Watch.
Insert Empty Row	N/a	Inserta una línea entre las variables.
View Symbol Definition	N/a	Visualiza la definición de los símbolos en le Symbol Manager.
Modify Value		Escribe un valor para la variable seleccionada.
Unforce		Desforza una variable seleccionada.
Unforce All		Desforza todas las variables que han sido forzadas.
Show All Forces		Visualiza todas las variables que han sido forzadas.
Cut	N/a	Corta el elemento seleccionado y lo ubica en el portapapel.

Barra de menú	Icono	Descripción
Copy	N/a	Copia el elemento seleccionado y lo ubica en el portapapel
Paste	N/a	Pega el contenido del portapapel.
Format		Seleccione el formato (binario, octal, decimal, o hexadecimal) para tipos de datos ANY_INT (BUTEs, WORDs, DWORDs, etc.). Si usted cambia el formato mientras la ventana Watch es pausada, el valor actualmente visualizado es reformateado (reestructurado). Esto es, un nuevo valor no es leído desde el Runtime Engine antes que la reestructuración no sea hecha.
Allow Docking ¹	N/a	Pulse para enmarcar o desenmarcar la ventana Watch.
Hide ¹	N/a	Remueve la ventana Watch

1 Esta opción no aparece en el menú cuando accedió desde la ventana Watch autónoma.

La ventana Watch soporta las características de **corta/copia/pega**. Esto le permite hacer lo siguiente:

- Duplica o mueve símbolos dentro de la ventana Watch.
- Copia símbolos desde el Symbol Manager para la ventana Watch.
- Copia símbolos desde una aplicación externa, así como Excel, o bloc de notas.
- Copia símbolos entre múltiples instancias de la ventana watch.

Uso de la ventana Watch autónoma

Usted puede correr una versión autónoma de la ventana Watch. Esta ventana opera muy similar a la ventana Watch que abre desde el ambiente de desarrollo. Sin embargo, con la ventana Watch independiente, usted puede monitorear variables durante la ejecución, sin abrir el Ambiente de Desarrollo. Además, puede correr múltiples instancias de la ventana Watch y monitorear las variables en uno o mas nodos.

➤ Para ejecutar la ventana Watch:

1. Sobre el icono de monitoreo del Runtime Engine, pulse en Watch window. Usted también puede correrlo desde el menú Start de la barra de tareas.
2. Si el cuadro de diálogo le sugiere un nodo, ingrese el nombre del nodo que quiere monitorear. Dependiendo del sistema designado, puede necesitar entrar la dirección TCP/IP. Para el nodo local, deje el campo Node en blanco.
3. Verifique el proyecto que intentó monitorear. El nombre del nodo y el proyecto aparece en la barra de Estado en la parte inferior de la ventana Watch. Si ubica el cursor en la barra de estado, el ayudante también muestra el nombre del proyecto.

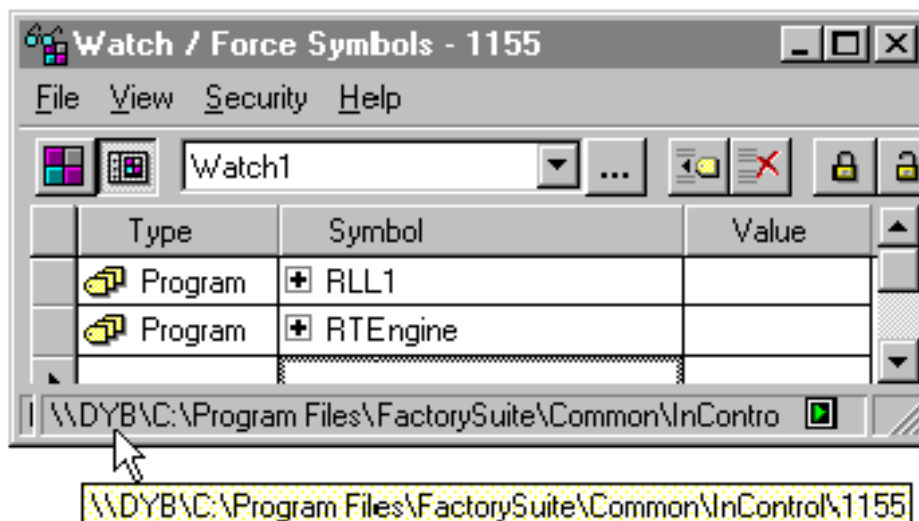











Figura 44. Visualización del nombre del Proyecto desde la barra de estado.

Los comandos del menú para la ventana Watch autónoma están descritos en la siguiente tabla.

Comandos de la ventana Watch independiente

Opción del menú	Comando	Icono de la Toolbar	Descripción
File	Project		Accede al Project Manager.
File	Connect / Disconnect		Conecta y desconecta la ventana del ambiente de desarrollo.
File	Add		Adiciona símbolo a la Watch window.
File	Remove		Remueve los símbolos desde la ventana Watch window.
File	Insert Empty Row	N/a	Inserta una fila entre las variables.
File	View Symbol Definition	N/a	Muestra la definición de los símbolos en el Symbol Manager.
File	Modify Value		Escribe o fuerza un valor a una variable.
File	Unforce		Cede variable seleccionada.
File	Unforce All		Cede todas las variables que han sido forzadas.
File	Show All Forces		Visualiza todas las variables que han sido forzadas.
File	Exit	N/a	Cierra las ventanas Watch.
View	Toolbar	N/a	Expone la barra de herramientas de la ventana Watch.
View	Status Bar	N/a	Expone la barra de estado de la ventana watch.
View	Format		Selecciona el formato (binario, octal, decimal, o hexadecimal)
Security ¹	Log On	N/a	Accede al cuadro de diálogo Log On.
Security ¹	Change Password	N/a	Accede al cuadro de diálogo Change Password.
Security ¹	Configure Users	N/a	Accede a el cuadro de diálogo Configure Users.
Security ¹	Log Off	N/a	Accede al cuadro de diálogo Log Off.
Help			Expone ayuda Online para el standalone de la ventana Watch.

Usted también puede seleccionar funciones para la ventana Watch ubicando el cursor dentro de la ventana y pulsando el botón izquierdo del mouse. Estas funciones están descritas en “**Uso del menú de la ventana Watch**”(pagina 14-).

Nota: El nivel de seguridad del administrador o Ingeniero es requerido para forzar o modificar una variable.

Un privilegio de la seguridad de usuario aplica a cualquier nodo remoto para el cual el usuario puede conectarse, también como para un nodo local donde el usuario es registrado.

Uso de las Variables del Sistema Runtime Engine

Estas variables proporcionan información adicional acerca del sistema, así como análisis y tiempo de ejecución de los programas, fallas de energía, etc. La siguiente tabla muestra las variables del sistema Runtime Engine.


Variables del Sistema del Runtime Engine	
Variable	Descripción
RTEngine.DivideZero	Valor Boolean que indica una división por cero cuando es TRUE; puesto por el Runtime Engine (RTE).
RTEngine.Error	Valor Boolean indica que una condición de error ha ocurrido en el Runtime Engine. Revise el WWLogger para mas información.
RTEngine.ExecAvg	Valor TIME que contiene el tiempo de ejecución promedio para la lógica del programa que esta siendo ejecutada.
RTEngine.ExecLast	Valor TIME (solo lectura) contiene el tiempo de ejecución anterior para la lógica del programa que esta siendo ejecutada.
RTEngine.ExecMax	Valor TIME que contiene el tiempo de ejecución máximo para la lógica del programa que esta siendo ejecutada.
RTEngine.ExecScan	Valor Boolean (solo lectura) indica un suceso del primer análisis de la lógica del programa cuando es TRUE.
RTEngine.ExecScanOn AutoStart	Valor Boolean (solo lectura). TRUE indica una ocurrencia de primer análisis de la lógica del programa después de un inicio automático seguido de un sistema reboto.
RTEngine.IOAvg	Valor TIME contiene en tiempo de análisis I/O promedio.
RTEngine.IOLast	Valor TIME (solo lectura) que contiene el tiempo de análisis I/O anterior.
RTEngine.IOMax	Valor TIME que contiene el tiempo de análisis máximo IO.

Variable	Descripción
RTEngine..Mode	<p>Valor INT que indica el modo actual del Runtime Engine:</p> <p>0 = Todos los programas son descargados.</p> <p>1 = Stop. Los programas de un proyecto son suspendidos.</p> <p>2 = Pause. Los programas de un proyecto son pausados.</p> <p>3 = Scan. El proyecto esta en modo de análisis único.</p> <p>4 = Program. Proyecto que actualmente esta siendo cargado al Runtime Engine.</p> <p>5= Run. Por lo menos un programa en in proyecto esta corriendo.</p> <p>6 = Fault. El Runtime Engine no puede correr el proyecto.</p>
RTEngine.PowerFail	<p>Valor Boolean (solo lectura) indica una falla de energía cuando es TRUE. Una configuración UPS es requerida. Note Usted debe restablecer estas variables manualmente, desde la ventana Watch.</p>
RTEngine.RelativeTime	<p>Valor TIME (solo lectura) que contiene el tiempo de duración que el Runtime Engine ha estado corriendo desde que el sistema fue booted. Este valor es independiente del reloj del sistema y puede ser usado en un programa para calcular intervalos cronometrados.</p>
RTEngine.ScanAvg	<p>Valor TIME que contiene el análisis medio.</p>
RTEngine.ScanLast	<p>Valor TIME (solo lectura) contiene la duración del último análisis.</p>
RTEngine.ScanMax	<p>Valor TIME que contiene el tiempo de análisis máximo.</p>
RTEngine.ScanOverrun	<p>Valor Boolean que indica un desbordamiento de análisis cuando es TRUE.</p>
RTEngine.ScanTime	<p>Valor LREAL que contiene la fijación de tiempo de análisis asignado por el usuario en milisegundos. Valores escritos a esta variable cambiarán el tiempo de análisis del Runtime Engine.</p>

Verificación del Wonderware Logger (registro)

El **Wonderware Logger** guarda información con respecto a la actividad hecha en el computador, por ejemplo: datos de inicio, condiciones de error, información del Servidor SuiteLink, etc. El Wonderware Logger es un servicio Windows NT y es configurado para iniciarse automáticamente cuando enciende la unidad de hardware.

Si un problema ocurre, por ejemplo, el Runtime Engine indica una condición de error, este seguro de revisar el Wonderware Logger antes de justificar el error. Siempre revise el Wonderware Logger para mensaje de errores antes de llamar un soporte técnico.

El icono de monitoreo del Runtime Engine que se muestra a continuación indica una condición de error y que la información ha sido escrita en el Wonderware Logger. 

Note que alguna información puede ser escrita en el Wonderware Logger inclusive si ocurre una condición de error, por ejemplo que cambie un modo.

Hay dos maneras por donde usted puede visualizar el Wonderware Logger.

- Pulse **inicio** en la barra de tareas, luego seleccione **WWLogger** debajo de **Common**, como se muestra abajo.

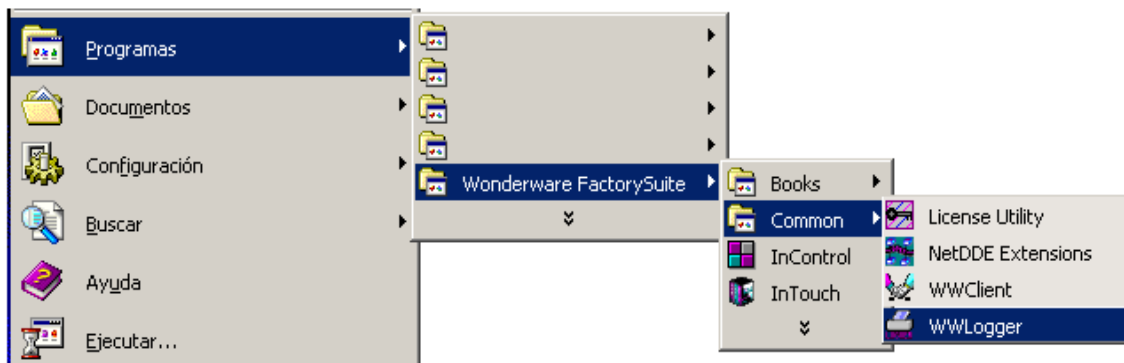


Figura 45. Ejecución del Woderware Logger.

- Haga clic derecho en el icono de monitoreo del Runtime Engine, y seleccione **View Logger**, como se muestra en la siguiente figura.

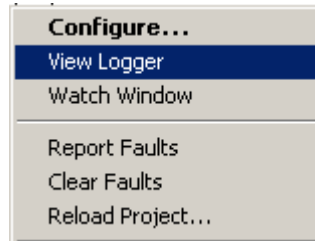


Figura 46. Inicio del Woderware Logger desde el Runtime Engine.

El **Window NT Event Viewer**(Visor de eventos) también monitorea eventos en su unidad de hardware. Para mas información acerca del **Event Viewer**, ver su documentación Windows NT.

INCONTROL E INTOUCH

Este capítulo describe las funciones de los wizards o ayudantes InControl usados en el ambiente InTouch.

CONTENIDO

- FUNCIONES INCONTROL SOPORTADAS POR INTOUCH.
- USO DEL WIZARD DE PROYECTOS INCONTROL.
- USO DEL WIZARD DE CONFIGURACION DEL RUNTIME ENGINE.
- USO DEL WIZARD DE MODO INCONTROL.
- USO DEL WIZARD EDITOR DE EJECUCIÓN INCONTROL.
- USO DEL WIZARD DE ELIMINACIÓN DE FALLAS DE INCONTROL.
- USO DEL WIZARD DE ADICIÓN DE ETIQUETAS DE EJECUCIÓN INCONTROL.
- USO DEL VISUALIZADOR DE ETIQUETAS DE INTOUCH.
- NODOS Y NOMBRES DE PROYECTOS.
- FUNCIONES DE PROGRAMACIÓN INCONTROL.

This product is licensed to:

Company: N/A
SN: N/A
Expires: N/A
Locked To: N/A

7.1.0.1337

FUNCIONES INCONTROL SOPORTADAS POR INTOUCH

InControl proporciona los siguientes seis Wizards o ayudantes que pueden ser ubicados en una ventana InTouch. Estos wizards le permiten una interacción rápida y efectiva entre InControl e InTouch.

- **InControl Project:** habilita al operador lanzar un proyecto InControl. Este empieza con InControl para el proyecto especificado y le permite al operador usar todas las funciones InControl para editar, compilar, transmitir y correr los programas en aquel proyecto. El ambiente de desarrollo InControl debe ser instalado en el sistema de hardware del operador.
- **Configure Runtime Engine:** Le permite al operador hacer cambios en línea de la configuración del Runtime Engine en el nodo especificado.
- **InControl Mode:** Le permite al operador fijar el modo (Run, Pause, Single Scan) para el proyecto que es transmitido al nodo especificado.
- **InControl Edit:** Le permite al operador lanzar un programa individual en un proyecto. Este inicia en el ambiente de desarrollo de InControl, en la línea especificada dentro del programa especificado. El operador puede usar todas las herramientas disponibles en el ambiente de desarrollo para editar, compilar, transmitir, y correr el programa. El ambiente de desarrollo InControl debe estar instalado en el sistema de hardware del operador.
- **InControl Clear Faults:** permite al operador justificar cualquier máquina que falla en el nodo especificado en ejecución.
- **InControl Runtime Add Tag:** Vincula etiquetas InTouch con símbolos(variables) InControl. El ambiente de desarrollo InControl debe estar instalado en el sistema de hardware del operador.

Instalación de los wizards InControl

Para las instrucciones acerca de la adición de los wizards InControl a la caja de herramientas y su colocación en la ventana InTouch, ver Guía de usuario InTouch.

USO DEL WIZARD DE PROYECTOS INCONTROL

Ubique el wizard **InControl Project**(Ayudante de Proyectos InControl) en una ventana InTouch para posibilitarle al operador iniciar la ejecución de un Proyecto. Pulsando este wizard desde la **WindowViewer** inicia InControl dentro del ambiente de ejecución para el proyecto especificado. El operador puede usar todas las funciones del ambiente de desarrollo de InControl para editar, validar, descargar, y correr los programas en aquel proyecto. El ambiente de desarrollo de InControl debe estar instalado en el sistema de hardware del operador.



Figura 1. El wizard InControl Project.

Si usted hace doble clic en el wizard InControl Project desde el **WindowMaker** de InTouch, aparecerá el cuadro de diálogo **InControl Project**.

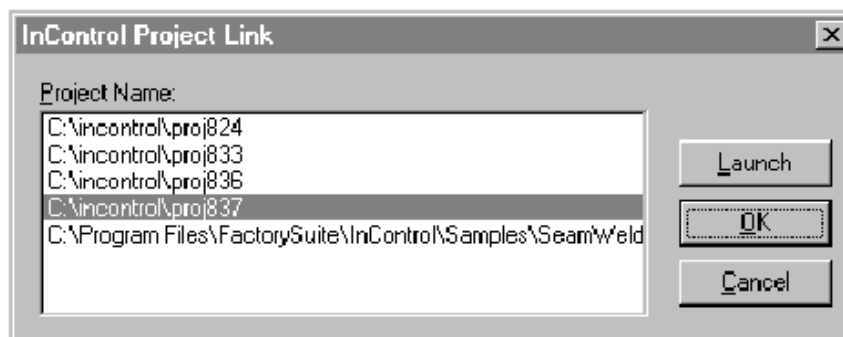


Figura 2. Cuadro InControl Project.

Cuadro de Diálogo InControl Project Link

Campo	Descripción
Project Name	Seleccione el proyecto predefinido para ser arrojado en el Ambiente de Desarrollo. El operador puede seleccionar cualquiera de los otros proyectos que aparecen en el campo Project Name. El ambiente de desarrollo InControl debe estar instalado en el sistema de hardware del operador antes que el operador pueda editar cualquier archivo del proyecto.
Launch	Pulse Launch para correr el proyecto InControl especificado desde la WindowMarker .
OK	Acepta el proyecto seleccionado para ser lanzado en el ambiente de desarrollo.
Cancel	Cierra el cuadro de diálogo sin guardar los ingresos.

USO DEL WIZARD DE CONFIGURACIÓN DEL RUNTIME ENGINE

Ubique el wizard **Configure Runtime Engine** (Ayudante de Configuración del Runtime Engine) en una ventana InTouch para habilitarle al operador hacer cambios de configuración al Runtime Engine en el nodo especificado para ejecutar.



Figura 3. El wizard Configure Runtime Engine.

- Para especificar el nodo por el cuál el operador configura el Runtime Engine:
 1. Haga doble clic sobre el wizard. Aparecerá el cuadro de dialogo **InControl Runtime Node**.

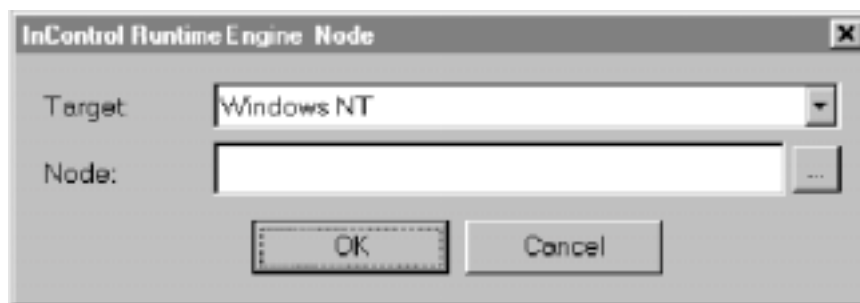


Figura 4. Cuadro InControl Runtime Engine Node.

2. Especifique la versión de la plataforma de hardware.
3. Para especificar al Runtime Engine el nodo local, deje el campo **Node** en blanco.

Para especificar al Runtime Engine el nodo remoto, entre el nombre del nodo en el campo **Node**.

Cuando el operador pulsa el wizard desde la ventana InTouch en la ejecución, aparecerá el cuadro de dialogo **Online Runtime Engine Properties**.

USO DEL WIZARD DE MODO INCONTROL.

Ubique el wizard **InControl Mode**(Ayudante de Modos InControl) en la ventana para permitirle al operador cambiar el modo (**Run Project, Single Scan**) para el proyecto que ha sido transmitido a la máquina ejecutora.



Figura 5. El wizard InControl Mode.

- Para especificar el nodo controlado por el wizard **InControl Mode**:
 1. Haga doble clic sobre este wizard. Aparecerá el cuadro de dialogo **InControl Runtime Node**.

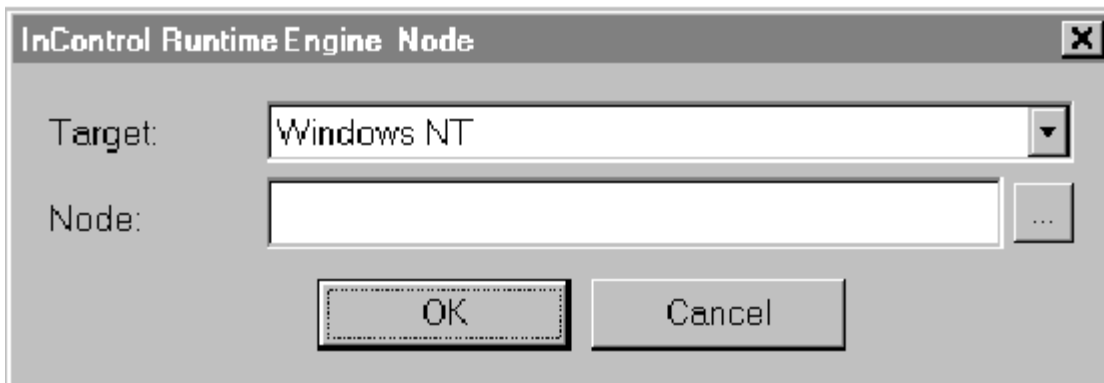


Figura 6. Cuadro InControl Runtime Engine Node.

2. Especifique la plataforma del hardware de la tarjeta.
3. Para especificar al Runtime Engine el nodo local, deje el campo **Node** en blanco.

Para especificar al Runtime Engine el nodo remoto, entre el nombre del nodo en el campo **Node**.

Para mas información acerca del modo del programa, ver “**SELECCIÓN DE OPCIONES DE EJECUCIÓN**” del capítulo “**EJECUCIÓN DE UN PROYECTO**”.

USO DEL WIZARD EDITOR DE EJECUCIÓN INCONTROL

Ubique el Wizard **InControl Runtime Edit** (Ayudante Editor de Ejecución InControl) en la ventana de InTouch para habilitar el operador al abrir un programa InControl en un punto específico del programa. Después que el editor abre el programa, el operador puede usar todas las funciones del ambiente de desarrollo de InControl para editar, compilar, transmitir, y correr el programa. Si hay otros programas en el proyecto, el operador puede acceder a ellos. El ambiente de desarrollo InControl debe ser instalado en el sistema de hardware de operador antes que el operador pueda editar cualquier archivo de proyecto.



Figura 7. El wizard InControl Runtime Edit.

Cuando el programa es abierto, el editor muestra el programa en la última localidad del cursor cuando el programa fue cerrado. Si usted quiere dirigir la atención del operador al código que controla una válvula, por ejemplo, el cuál esta sobre la ventana Windows, ubique el wizard editor cerca de la válvula. Abra el programa, mueva el cursor al código de la válvula, y luego cierre el programa.

Si usted hace doble clic en el Wizard **InControl Runtime Edit** desde la **WindowMaker** de InTouch, aparecerá el cuadro de dialogo **InControl Editor**.

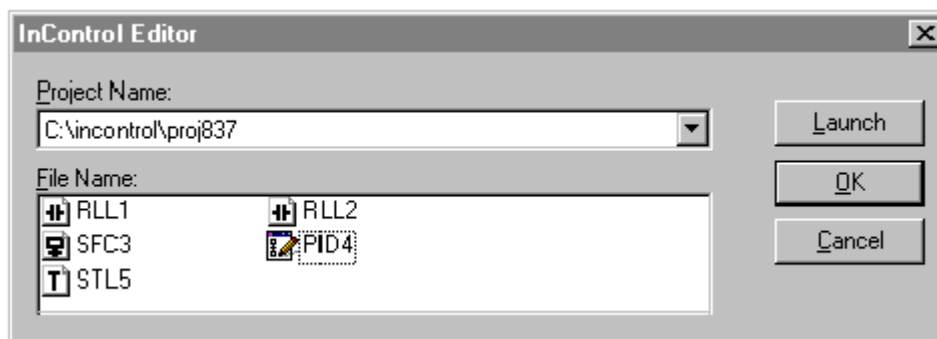


Figura 8. Cuadro InControl Editor.

Cuadro de Diálogo InControl Editor

Campo	Descripción
Project Name	Selecciona el proyecto InControl contenido en el programa para editar.
File Name	Selecciona el programa a ser editado.
Launch	Pulse Launch para correr el programa InControl especificado desde el WindowMaker .
OK	Acepta el programa seleccionado a ser editado.
Cancel	Cierra el cuadro de diálogo sin guardar las entradas.

USO DEL WIZARD DE ELIMINACIÓN DE FALLAS INCONTROL

Ubique el wizard **InControl Clear Faults**(Ayudante de Eliminación de Fallas InControl) sobre la ventana InTouch para habilitarle al operador eliminar cualquier falla de ejecución sobre el nodo especificado. Cuando el operador pulse el wizard en la ejecución, los bits de estados del Runtime Engine, como el **RTEngine.ScanOverrun**, son eliminados.



Figura 9. El wizard InControl Runtime Engine Clear Faults.

- Para especificar el nodo por el cual el operador eliminara las fallas:
 1. Haga doble clic al wizard. Aparecerá el cuadro de diálogo **InControl Runtime Node**.

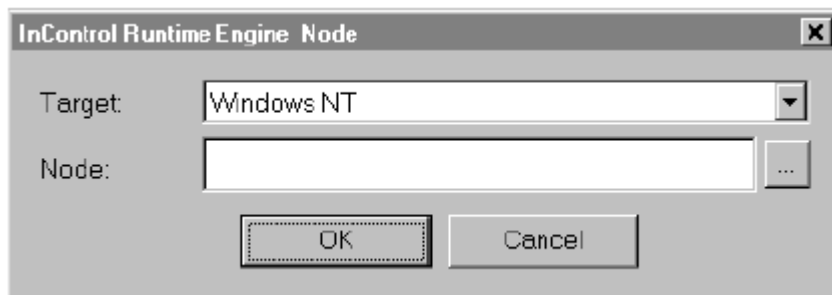


Figura 10. Cuadro InControl Runtime Engine Node.

2. Especifique la plataforma de hardware de la tarjeta.
3. Para especificar al Runtime Engine el nodo local, deje el campo **Node** en blanco.

Para especificar al Runtime Engine el nodo remoto, ingrese el nombre del nodo en el campo **Node**.

USO DEL WIZARD DE ADICIÓN DE ETIQUETAS DE EJECUCIÓN INCONTROL

Use el wizard **InControl Runtime Add Tag** (Ayudante de Adición de Etiquetas de Ejecución InControl) para vincular etiquetas InTouch a los símbolos (variables) usados en un proyecto InControl. El ambiente de desarrollo InControl debe ser instalado donde este corriendo InTouch para que pueda vincular símbolos InControl a etiquetas InTouch.



Figura 11. El wizard InControl Runtime Add Tag.

Muestra de los símbolos InControl

Cuando usted pulsa este wizard desde el WindowMarker, el cuadro de diálogo **InControl Tag Import** aparece. Todos los símbolos usados en los programas dentro del proyecto están listados. Para vincular un símbolo a una etiqueta, ver “Vincular Símbolos InControl a Etiquetas InTouch”.

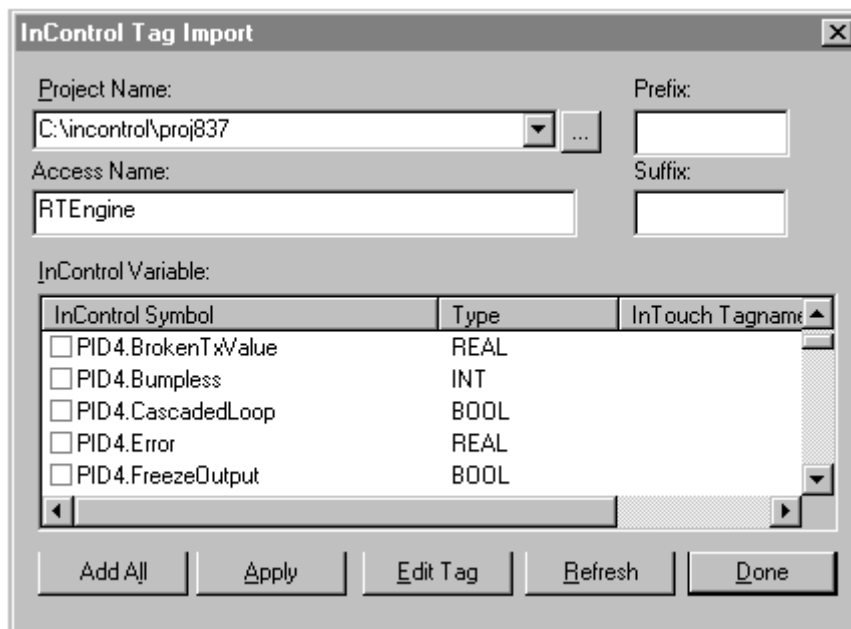


Figura 12. Cuadro de diálogo InControl Tag Import.

Cuadro de Diálogo InControl Tag Import

Campos	Descripción
Project Name	Selecciona el proyecto con los símbolos para ser vinculados a las etiquetas. Los proyectos InControl listados en este campo son los mismos que unos configurados en el ambiente de desarrollo InControl, si esta instalado en el nodo local. Si InControl no esta instalado en el nodo local, esta lista queda en blanco. Usted puede examinar la red y adicionar proyectos InControl a la lista.
Access Name	Especifica el nombre de acceso a ser usado para acceder etiquetas que vincule. Para el nodo local, use el valor RTEngine predefinido. Para un nodo remoto, ingrese la definición del nombre del acceso mostrado en el cuadro de diálogo de InTouch Access Name para el nodo. (pulse Access Name sobre el menu Special para abrir este cuadro de diálogo).
Prefix / Suffix	Opcional. Entre un prefijo o un sufijo para los nombres de etiquetas que son importadas dentro del InTouch. Esto lo habilita para agrupar etiquetas, para proyectos por ejemplo, si usted esta observando varios grupos en una ventana InTouch.
InControl Symbol	Nombre del símbolo usado en InControl. Haga doble clic en el nombre del símbolo para mostrar el cuadro de diálogo Add Tag ¹ . Si las etiquetas ya han sido adicionadas a la base de datos InTouch, haga doble clic al nombre de el símbolo para mostrar el cuadro de diálogo Tagname Dictionary .
Type	Campo de solo lectura que muestra el tipo de dato para el símbolo.

Campos	Descripción
InTouch	Nombre de la etiqueta a ser usada en InTouch. El campo esta en blanco hasta que el símbolo InControl haya sido vinculado a la etiqueta InTouch.
Tagname	Vincula todos los símbolos del proyecto a las etiquetas InTouch.
Add All	Vincula todos los símbolos InControl seleccionados a las etiquetas InTouch.
Apply	Muestra el cuadro de diálogo Add Tag si la etiqueta no ha sido adicionada a la base de datos InTouch.
Edit Tag	Muestra el Tagname Dictionary si la etiqueta ya ha sido adicionada a la base de datos InTouch
Refresh	Muestra de nuevo toda la información en el cuadro de diálogo.
Done	Cierra el cuadro de diálogo.

Vinculación de Símbolos InControl a Etiquetas InTouch

Cuando usted muestra el cuadro de diálogo **InControl Tag Import**, todos los símbolos usados en los programas dentro de los proyectos son mostrados. Seleccione uno o más símbolos y pulse **Apply**. Aparecerá el cuadro de diálogo **Add Tag**. También puede hacer doble clic en un símbolo individual para mostrar el cuadro de diálogo **Add Tag**.



Figura 13. Cuadro de diálogo Add Tag.

Cuadro de diálogo Add Tag

Campo	Descripción
InControl Symbol	Campo de solo lectura que muestra el nombre del símbolo a ser vinculado.
InControl Type	Campo de solo lectura que muestra el tipo de dato de el símbolo.
Default Tagname	Nombre de la etiqueta a ser usada por InTouch. El nombre del símbolo InControl es el predefinido.
Yes to All	Vincula todos los símbolos seleccionados a las etiquetas InTouch.
Yes	Vincula el símbolo seleccionado a la etiqueta InTouch.
No	Cierra el cuadro de diálogo sin vincular el símbolo a la etiqueta. Si símbolos múltiples están seleccionados, el próximo símbolo es mostrado.
Cancel	Cierra el cuadro de diálogo sin guardar las entradas.

La siguiente figura muestra el cuadro de diálogo **InControl Tag Import** después que dos símbolos han sido vinculados a las etiquetas.

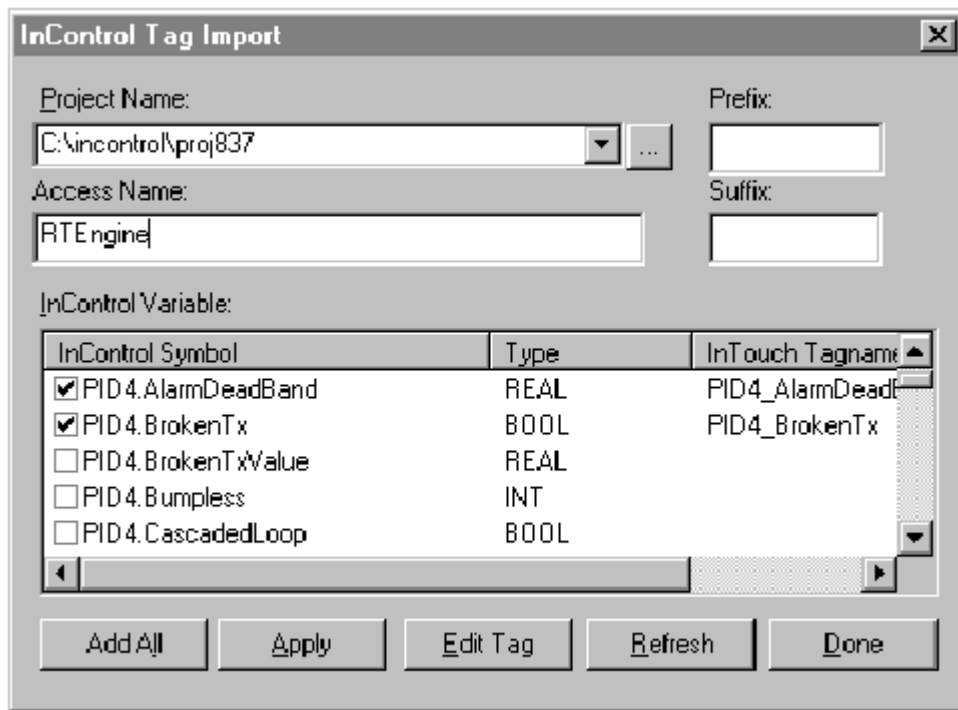


Figura 14. Cuadro de diálogo InControl Tag Import.

Vinculación de Etiquetas Sobre Sistemas Remotos

Usted puede vincular símbolos InControl, los cuales son usados en proyectos locales sobre nodos remotos, a etiquetas InTouch. Los sistemas remotos deben contener una copia de InControl, y debe estar conectado a una red con los sistemas en los que esta corriendo InTouch. Si usted esta usando DDE debe implementar NetDDE entre los dos sistemas NetDDE. Si esta usando Suite Link, no es necesario usar DDE.

➤ Para vincular los símbolos de un nodo remoto a etiquetas InTouch sobre el sistema local:

1. Pulse el wizard **InControl Runtime Add Tag** desde el Window Marker.

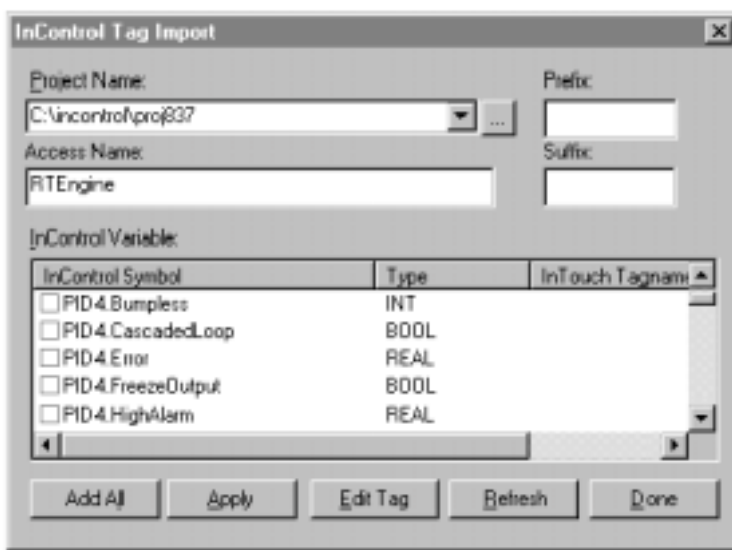


Figura 15. Cuadro InControl Runtime Engine Node.

2. Pulse el botón que se encuentra a la derecha del campo **Project Name** para mostrar todos los proyectos en lista de proyectos.



Figura 16. Campo Project Name.

3. Seleccione el proyecto remoto. El cuadro **InControl Tag** muestra los símbolos para el proyecto remoto.
4. Vincule los símbolos a las etiquetas InControl, descritos en el capítulo “**Vinculación de símbolos InControl a Etiquetas InTouch**”.

USO DEL VISUALIZADOR DE ETIQUETAS DE INTOUCH

Una alternativa para importar símbolos InControl dentro del InTouch es usando el **Tag Browser**(Visualizador de Etiquetas) de InTouch. Es muy útil cuando esta escribiendo un **Script**(archivos de comandos) y necesita referenciar un símbolo InControl.

Note que también puede usar herramientas de importación y exportación sobre la barra de herramientas del Symbol Manager para intercambiar símbolos entre InTouch e InControl. El Formato de Coma Separada (**CSV**) es usado, lo cual significa que usted puede editar fácilmente un archivo o dato de símbolo usando un editor de texto ASCII o Excel.

Para mas información acerca de las herramientas de Importación y exportación, ver “**TRANSFERENCIA DE BASES DE DATOS DE SÍMBOLOS**” en el capítulo “**DEFINICIÓN DE VARIABLES**”.

Si usted intentó usar InTouch e InControl sobre sistemas separados y necesita observar los símbolos de InControl desde el InTouch, debe instalar los archivos del **Tag Browser** de InControl sobre el sistema donde InTouch esta localizado. Correr el programa setup de InControl sobre el sistema InTouch y seleccione **InTouch Extensions** cuando sugirió la opción setup.

NODOS Y NOMBRES DE PROYECTOS

Usted puede usar el **FOE de Información de Proyectos** para enviar el nombre de un proyecto y el nodo en el cual el proyecto se esta ejecutando para un HMI, como InTouch.

Para información acerca del uso del **FOE de Información de Proyectos** para enviar información de proyectos a un HMI, ver “**USO DEL OBJETO DE INFORMACIÓN DE PROYECTOS**” en el capítulo “**USO DEL EDITOR DE OBJETOS DE FÁBRICA DE INCONTROL**”.

FUNCIONES DE PROGRAMACIÓN INCONTROL

Las capacidades del **InTouch QuickScript** le permiten al operador ejecutar comandos y operaciones lógicas basadas en un criterio especificado. Por ejemplo, puede definir un botón que el operador seleccione para abrir un proyecto InControl o eliminar fallas en el Runtime Engine.

InControl soporta dos funciones QuickScript que puede usar dentro de un InTouch script : **InControl()** e **InControlRuntimeEngine()**.

InControl()

Abre un proyecto, programa, Configuraciones I/O, entre otros, de InControl.

Sintaxis `InControl ("Project", "Command", "Param");`

Parámetro	Descripción	Valor
Project	Trayectoria de un proyecto InControl.	<Especificado por usuario>.
Command	Operación a ser ejecutada	Launch
Param	Nombre del objeto para ser ejecutado.	<Especificado por usuario>.

Ejemplo `InControl("c:\Projects\SeamWeld", "Launch", "BandLogic");`

Abre el programa llamado **bandLogic** del proyecto llamado **SeamWeld** en el ambiente de desarrollo de InControl.

InControlRuntimeEngine()

Envía un comando al Runtime Engine.

Sintaxis `InControlRuntimeEngine("Nodenamet", "launch", "BandLogic");`

Parámetro	Descripción	Valor
Nodename	Nombre del nodo donde la máquina es ejecutada.	<especificado por usuario>.
Command	Operación a ser ejecutada.	Mode Configure ClearFaults
Param	Usado solamente con el comando Run Mode; especifica el modo de la máquina.	Run Pause Single Scan Stop

Ejemplo `InControlRuntimeEngine("Node75", "Mode", "Stop");`


Detiene el proyecto que esta corriendo en el nodo llamado Node75.

`InControlRuntimeEngine("Node42", "Configure", "");`

Muestra el cuadro de diálogo **Online Runtime Engine Properties** para el Runtime Engine que esta corriendo en el nodo llamado **Node42**.

ADMINISTRACIÓN DEL SISTEMA INCONTROL

Este capítulo describe las funciones de nivel de sistema InControl.



CONTENIDO

- DESCRIPCIÓN DE LOS EVENTOS DEL RUNTIME ENGINE
- CONFIGURACIÓN DEL RUNTIME ENGINE
- REVISIÓN EN LA OPERACIÓN DEL RUNTIME ENGINE
- MANEJO DE FALLAS DE ENERGÍA.
- EJECUCIÓN DE MÚLTIPLES PROYECTOS.
- CAMBIO DE LA CLAVE EN EL REGISTRO DEL SISTEMA.
- SOPORTES DE CONTENIDO, MODALIDAD Y TIEMPO.
- INGRESO A LA CONFIGURACIÓN DEL VISOR DE SUCESOS.

This product is licensed to:

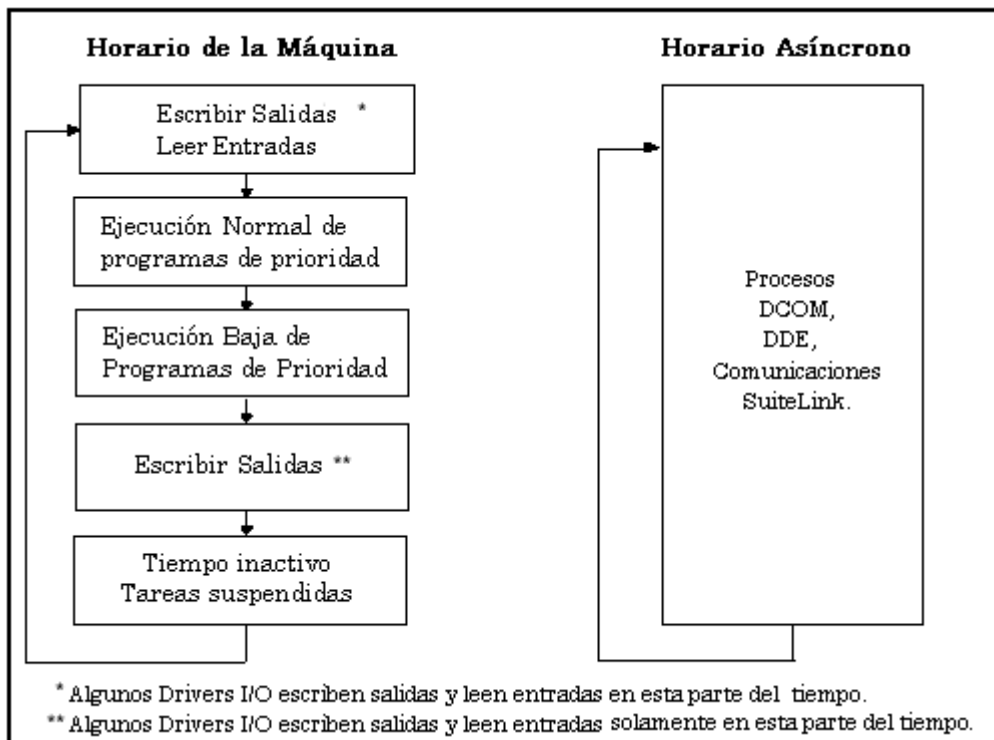
Company:	N/A
SN:	N/A
Expires:	N/A
Locked To:	N/A

7.1.0.1337

DESCRIPCIÓN DE LOS EVENTOS DEL RUNTIME ENGINE

El software de ejecución lee y escribe a el I/O , ejecuta programas de aplicación, y hace tareas superiores y comunicaciones SuiteLink/DDE. El tiempo requerido para hacer todas estas tareas, llamado Total Scan Time (tiempo de análisis total), es ajustable a del cuadro de diálogo **Runtime Engine Properties**.

Un diagrama de un análisis completo para el Runtime Engine (el horario) se muestra en la siguiente figura.



Lecturas y escrituras inmediatas se llevan a cabo por variables locales y globales que no sean variables I/O. Usted puede usar DCOM e interfaces SuiteLink para leer y escribir valores inmediatos por lógica RRL y SFC. Esto es hecho asincrónicamente, como se muestra en la figura. Operaciones de lectura y

escritura SuiteLink desde estas interfaces ocurren inmediatamente, en cualquier momento que sean requeridos por la aplicación externa, e.g., InTouch. La frecuencia de un "Aviso" DCOM esta determinada por Data Monitor Update Interval (Intervalo de actualización del monitor de datos). Usted especifica ambos valores en el cuadro de diálogo **Runtime Engine Properties**.

Tarjetas de unidades de control, arquitectura de bus de control, y módulos I / O todos tienen diferentes tiempos de cronómetro, los cuales son a menudo implementaciones específicas. Es posible fijar un tiempo de análisis total dentro de InControl que sea más rápido que lo que aquellos equipos pueden manejar. Para más información acerca de cómo manejar esta situación, ver "Ajustando Tiempo de Análisis".

Porque la ejecución del Runtime Engine tiene una alta prioridad dentro del sistema operativo Windows NT, apropias otras operaciones, así como el movimiento del mouse, por ejemplo. Disminución en el tiempo de análisis podría causar que el sistema responda más lentamente para entradas de usuario desde el mouse o teclado.

Detalles del horario o de las líneas de tiempo son mostrados a continuación. La ejecución de programas está descrito en las páginas que siguen.

1. Actualizar el I/O para la primera tarjeta listada en el I/O View. Una actualización está basada en la implementación de los Driver I/O; algunas tarjetas leen y escriben I/O en este espacio de tiempo, el cual otras tarjetas solamente leen entradas. Actualización adicional de las tarjetas en el mismo orden en que son listadas en el I/O View.
2. Ejecutar programas de prioridad normal en el orden en que aparecen en el Execution View (Visualizador de Ejecución). * Por ejemplo, especifique el orden de ejecución en el Execution View, como se describe en "project / Program Execution Order" en el capítulo "Corriendo proyectos". Si usted cambia el orden en el Execution View, el nuevo orden no tiene efecto hasta el próximo proyecto descargado.

Programas de prioridad Normal corren en una relación de tiempo real (asíncrono) con el I/O. El Runtime Engine siempre ejecuta estos programas. Si el tiempo de análisis total es fijado demasiado lento, los programas corren, pero la variable del sistema RTEngine,ScanOverrun se pone en TRUE.

3. Los programas de prioridad baja se ejecutan en el orden en que son descargados. * Como con los programa de prioridad Normal, usted puede especificar el orden de descarga en el Execution View. Si un programa es descargado después que un proyecto es descargado, su orden de ejecución como lo definió en el Execution View es preservado.

Programas de prioridad baja corren solamente si el tiempo se mantiene después que todas las otras tareas en el timeline han sido ejecutadas. Si todos los programas de prioridad baja no son ejecutados en un tiempo único, la ejecución empieza después del último programa de prioridad baja que corrió durante el anterior análisis.

Note que los programas de prioridad baja se ejecutan como Windows NT high – priority real-time threads. (hilos de tiempo real de Windows NT de alta prioridad).

4. Tarjetas I/O que solamente leen entradas en el escalón 1 escribe salidas o resultados en esta fracción de tiempo.
5. Tareas suspendidas. Si el Runtime Engine empieza a usar mas que que el porcentaje de asignación especificado en el campo **Processor Utilization** (“Etiquetas en línea y fuera de línea en las propiedades del Runtime Engine ”), empieza a saltar análisis. Este diseño le impide al Runtime Engine consumir mucho tiempo del procesador para que la interacción del usuario no sea bloqueada. Si el análisis no se completa antes que el watchdog timer, especificado en el campo **Watchdog Timeout** es restaurado (reset), el servicio del Runtime Engine es detenido.

*Cuando el Runtime Engine esta en modo Pause, los programas de prioridad normal y baja no son ejecutados.

Ejecución SFC

El orden de ejecución general para los elementos del programa de un SFC es como se sigue:

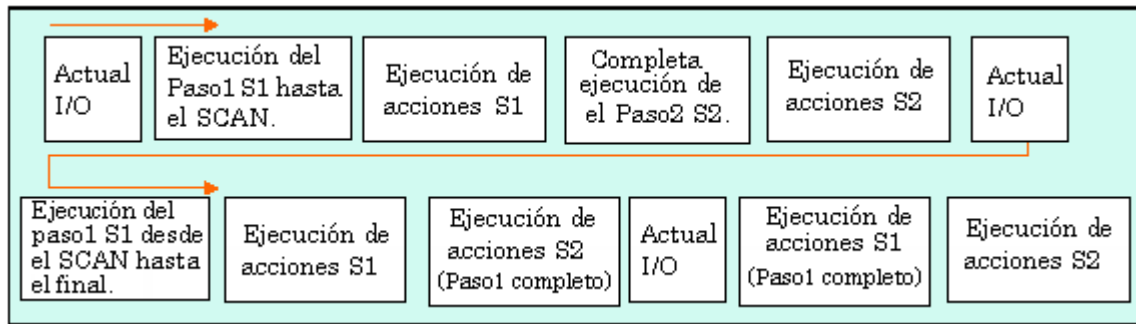
1. Evalua todas las transiciones activas. Actions y Step (acciones y pasos) son programados para ejecución.
2. Ejecuta tareas programadas: Step code (Structured Text o Macros).
3. Ejecuta Actions (acciones).

El código de Texto Estructurado en un Step es ejecutado una vez a la realización cuando el Step es activado primero. Excepciones pueden ocurrir en operaciones largas, así como lazos (FOR, WHILE; REPEAT), y operaciones de acceso a archivos. Para lazos, ejecuciones de texto estructurado es suspendido hasta el próximo análisis, a menos que el lazo este terminado con una de las declaraciones NOWAIT. La declaración SCAN obliga a la ejecución a suspender hasta después del próximo análisis.

El SFC no hace transición hasta que el código de Texto estructurado este completo. Además, después que el código este completo, el Step no corre de nuevo hasta que el Step haya salido y haya vuelto a entrar.

Cuando una acción termina, es ejecutada una vez mas en el análisis siguiente, con la entrada del escalón puesta en FALSE. Esto permite “enrollar” (coils) cronómetros, contadores y resultados a restaurar. Si usted quiere que la lógica adicional sea ejecutada cuando la acción es terminada, use el bloque de función F_TRIG para negar el flujo de energía FALSE dentro del escalón.

En el siguiente ejemplo, el proyecto contiene dos SFCs, S1 y S2, cada uno con un Step. Si S1 tiene una petición SCAN a medio camino a través de su ejecución, obtiene el siguiente orden:



Ejecución de programa de texto estructurado

El código de Texto Estructurado en un Paso (Step) es ejecutado una vez a la realización cuando el Step es activado primero. Excepciones pueden ocurrir en operaciones largas, así como lazos (FOR, WHILE; REPEAT), y operaciones de acceso a archivos. Para lazos, ejecuciones de texto estructurado es suspendido hasta el próximo análisis, a menos que el lazo este terminado con una de las declaraciones NOWAIT. La declaración SCAN obliga a la ejecución a suspender hasta después del próximo análisis.

Ejecución de RLL

La ejecución de los programas RLL consisten de un completo análisis del programa.

Ejecución de FOE

La ejecución de un FOE consiste en llamar su método de control una vez por análisis. Para controles ActiveX que no están en InControl, una instrucción SFC o Texto Estructurado debe llamar al control ActiveX donde quiera que este para que el control corra.

Algunos FOEs, así como el PID FOE, están designados para ser ejecutados automáticamente. Ellos tienen un método que es llamado una vez por cada análisis de I el Runtime Engine . Cuando cargo a el Runtime Engine , el modo para aquellos FOEs pueden ser Run, Pause, Stop, etc.

Algunos FOEs no tienen un método designado para correr automáticamente por cada análisis. Para estos FOEs, usted necesita incluir códigos en un SFC o programa de Texto estructurado para llamar un método y que el FOE ejecute correctamente. Cuando cargó al Runtime Engine , el modo para aquellos FOEs esta indicado como estando en Loaded.

Ejecución de Programas y Fijando programas

La opción del Setp Program le permite ejecutar una línea de la lógica único. Un programa debe estar en modo Pause antes que usted pueda andarlo paso por paso.

Cada vez que pulsa **Step Program** sobre la barra de herramientas, los programas son ejecutados como sigue:

- Para un programa RLL, un escalón de lógica es ejecutado.
- Para un programa de Texto estructurado, una línea de código s ejecutado.
- Para un programa SFC, una línea de código de un activo Step, es ejecutado. Si mas de un Step esta activo, cada Step es dado a paso a su vez. Todos los Actions activos son ejecutados completamente.
- Para un FOE, la opción Step Program no es soportada.

Ejecución de Programas y Proyectos y Single Scanning

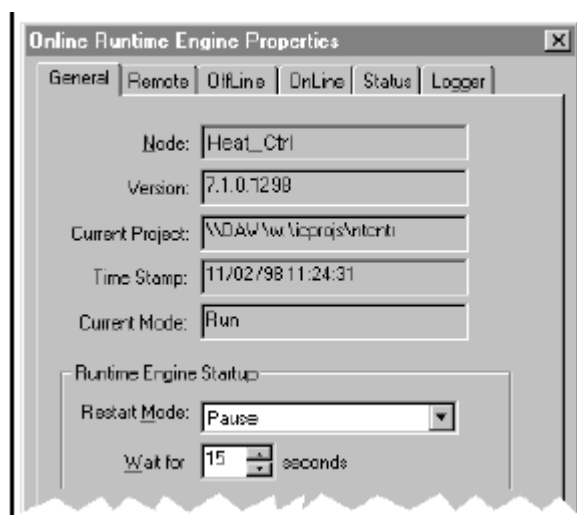
La opción Single Scan (análisis único) le permite ejecutar todos los programas en un proyecto para un análisis completo, siguiendo el cronograma (timeline) descrito en "Horario de la El Runtime Engine ". El orden de la ejecución del programa esta basado en el orden de los programas en el Execution View. Un project/programa debe estar en el modo Pause antes que pueda ejecutar un análisis único.

Nota El Single-stepping puede causar orden de ejecución para diferenciarse de un programa que corre normalmente. Este se debe a la ejecución continua de I/O, otros programas, y en el caso de SFCs, Actions.

CONFIGURACIÓN DEL RUNTIME ENGINE

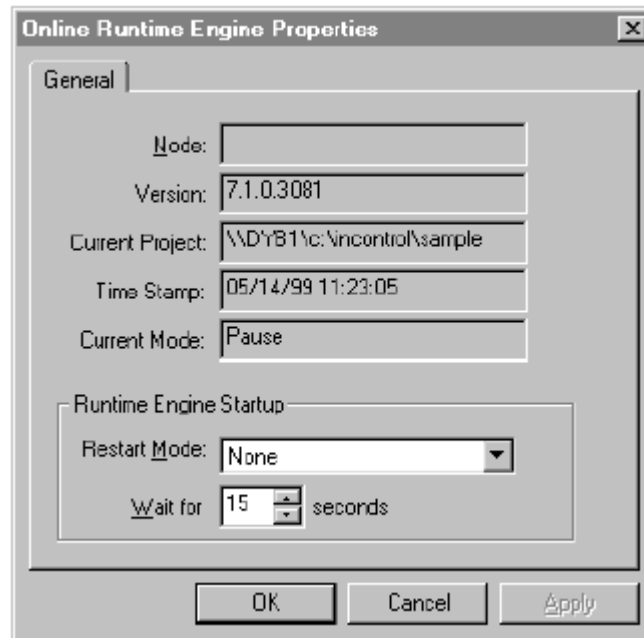
Acceso a las propiedades del RUNTIME ENGINE

Para configurar el Runtime Engine , pulse **Configure** en el menú **Runtime**. Una el cuadro de diálogo de diálogo aparece, dependiendo de si el ambiente de desarrollo no esta conectado, del cuadro de diálogo **Offline Runtime Engine Properties** aparece. Si el ambiente desarrollo esta conectado aparece del cuadro de diálogo **Online Runtime Engine Properties** aparece. Además las etiquetas aparecen el la anterior el cuadro de diálogo de diálogo como se muestra abajo.



Revisando las Propiedades Generales

La etiqueta general del cuadro de diálogo **Runtime Engine Properties** se muestra en la siguiente figura.



La configuración podría variar para diferentes plataformas de hardware. Para mas información acerca de la configuración del Runtime Engine sobre otras plataformas, ver la guía de usuario apropiada para la plataforma de hardware.

Fijando Tiempo de Análisis

Use las etiquetas de Offline o Online del cuadro de diálogo **Runtime Engine Properties** para designar la utilización del procesador y fijar intervalos de tiempo, así como tiempo de análisis, y tiempos de actualización para el Watch Window.

- Etiqueta Offline -- Los cambios que usted haga en esta etiqueta tienen efecto la próxima vez que afectan el tiempo del próximo o programa individual que descargue. Offline también aparece cuando pulsa **Configure** en el menú **Runtime** o doble click en el ícono **RTEngine** en el Project View.
- Etiqueta Online – Los cambios que usted hace tienen efecto tan pronto como los aplique. Sin embargo, la próxima vez que cargue un proyecto o programa e incluya la configuración del proyecto (Seleccione **Full Restart** sobre del cuadro

16-10 **CAPITULO 16**

de diálogo **Run Project**, por ejemplo) las fijaciones de la etiqueta fuera de línea sobre escriben las fijaciones de la etiqueta en línea.

La etiqueta Online aparece cuando ubica el cursor sobre el icono del monitor del Runtime Engine en la barra en tareas, pulse right-clic, y pulse **Configure**. Si el ambiente de desarrollo esta conectado al Runtime Engine , pulse **Configure** sobre el menú o haga doble clic al icono **RTEngine** en el Prject View.

Los campos de las etiquetas Offline y Online del cuadro de diálogo **Runtime Enine Properties** son mostrados en la siguiente figura:



La configuración tal vez pueda variar para diferentes tarjetas de plataformas de hardware. Para mas información acerca de la configuración del Runtime Engine sobre otras plataformas, ver la guía de usuario apropiada para la plataforma de hardware.

Advertencia: La fijación del tiempo de análisis demasiado bajo puede ocasionar que al Runtime Engine salte scans.

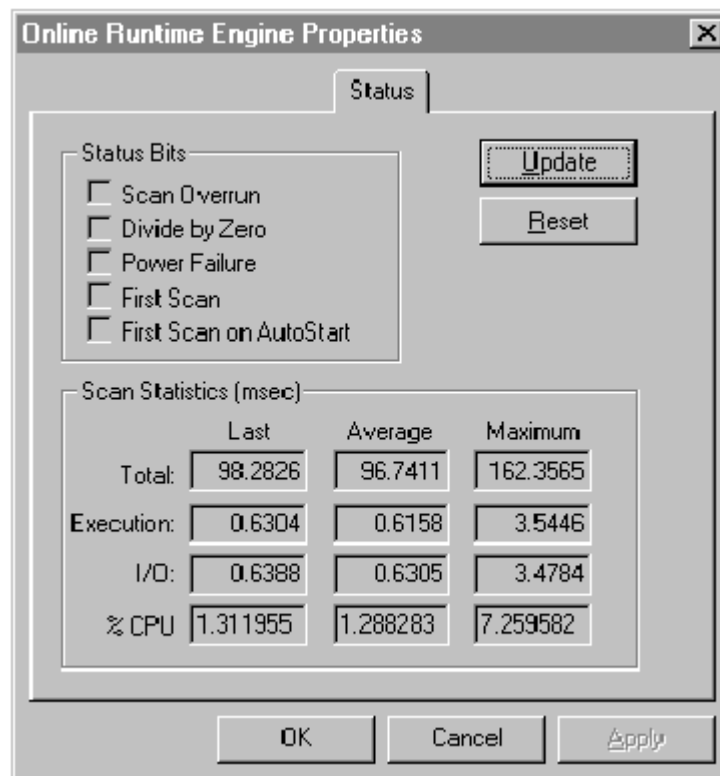
Esto podría resultar en operaciones impredecibles por los equipos I/O y ocasionar herida o muerte y/o daños a equipos. Asegúrese de seleccionar un tiempo de análisis apropiado para su aplicación. Ver "Ajustando el Tiempo de análisis" para mas información.

REVISIÓN EN LA OPERACIÓN DEL RUNTIME ENGINE.

Esta sección describe como revisar estadísticas para la operación del Runtime Engine , ajuste del tiempo de análisis y limpiar condiciones de falla.

Observación de los Datos de Estado

Las etiquetas de estado del cuadro de diálogo **Runtime Engine Properties** se muestra en la siguiente figura. Use esta hoja de propiedades para revisar o restaurar tiempos de scan, bits de estados, y uso de memoria asignada por InControl.



Ajustando el Tiempo de Análisis

Esta sección provee algunas líneas para especificar el más eficiente tiempo de análisis para su unidad de hardware.

Utilización de la CPU

Es importante ajustar al Runtime Engine para balancear sus requerimientos de CPU con aquellos de otras aplicaciones, incluyendo el sistema operativo. Estos ayudaran a evitar consumir todos los recursos de la CPU, el cual pudo bloquear el sistema. InControl provee dos propiedades en el cuadro de diálogo **Runtime Engine Properties**. Use estas propiedades para determinar la cantidad de tiempo de procesador que tiene al Runtime Engine para ejecutar la lógica y procesar I/O.

- El Scan Time determina que tan a menudo al Runtime Engine ejecuta la lógica.
- La propiedad Processor Utilization determina el máximo porcentaje del tiempo de análisis que puede ser usado por al Runtime Engine en la ejecución de la lógica del programa y procesos I/O.

Usted debe especificar apropiadamente los valores para estos dos parámetros según sus necesidades y las necesidades del Runtime Engine .

Sobre cada Scan, al Runtime Engine hace el siguiente cálculo para determinar un porcentaje CPU:

$$\text{CPU\%} = \frac{\text{Execution Time (normal priority programas) + I/O Time}}{\text{Scan Time}}$$

Cuando al Runtime Engine hace estos cálculo, ella revisa para ver si el porcentaje CPU excedel valor que usted tiene especificado. Si su valor es excedido, entonces al Runtime Engine saltará análisis para regresar el porcentaje CPU observado a su valor

Por ejemplo, si usted especifica el 20% de utilización CPU y 100 ms de tasa de scan, entonces 20 ms estarán asignados a cada análisis para la ejecución de un programa de prioridad normal y Scan I/O. Si los programas e I/O consistentemente llevan 30 ms, entonces al Runtime Engine empezará saltando cada ocho scans para devolver el porcentaje CPU a 20%. Si los programas de prioridad normal e

I/O consistentemente llevan 80 ms, entonces al Runtime Engine correrá un scan y luego saltará tres scans para devolver el porcentaje CPU a 20%. En ambos de estos casos, si usted especifica 80% para la Utilización CPU, ningún scan es saltado.

Considere que al Runtime Engine tiene un porcentaje designado de uso CPU, y hará los cálculos necesarios y saltará los scans necesarios para alcanzar aquel objetivo.

Advertencia Saltos Scan pudo resultar en operaciones impredecibles por los equipos I/O y ocasionar daños o muerte a personal y / o daños a equipo. Para ayudar a evitar saltos de scans, asegúrese de seleccionar un tiempo de análisis y utilización CPU que es apropiado para su aplicación.

Normalmente, ocurren algunas fluctuaciones en ambos tiempos de ejecución de programas y tiempos de ejecución I/O de scan a scan . Por lo tanto, es recomendable que deje algún sitio adicional en el calculo de porcentaje CPU para acomodar aquellas fluctuaciones normales.

Usando el tiempo scan y porcentaje CPU para ahogar al Runtime Engine tiene las siguientes metas:

- Permitir otras aplicaciones (con algunas limitaciones) correr en la misma unidad de hardware como al Runtime Engine .
- Prevenir al sistema de bloqueos en caso de un imprevisto sobre uso de la CPU por la lógica del programa o scan I/O.

El mecanismo para este estrangulamiento esta diseñado para ayudar en situaciones donde programas e I/O no están operando normalmente. Estos saltos no deberían ocurrir durante una operación normal.

Programas de prioridad baja son tratados en un modo diferente. Si al Runtime Engine no ha usado su intervalo de tiempo designado, entonces un programa de prioridad baja correrá. Este programa puede correr mas allá de los límites de porcentaje CPU, y no será tenido en cuenta como parte del porcentaje CPU del

16-14 **CAPITULO 16**

Runtime Engine . Si el primer programa de prioridad baja termina la ejecución y algunas veces se mantiene en el intervalo del Runtime Engine durante aquel scan, entonces otro programa de prioridad baja será iniciado.

Nota Sobre un sistema de procesador único, es recomendable que evite fijación de la utilización de la CPU a un 100%. En general, dependiendo del sistema y la aplicación sobre el que usted esta corriendo, no fije el CPU Utilization mayor que 90%. Para un sistema multiprocesador, un CPU Utilization de 100% es recomendable.

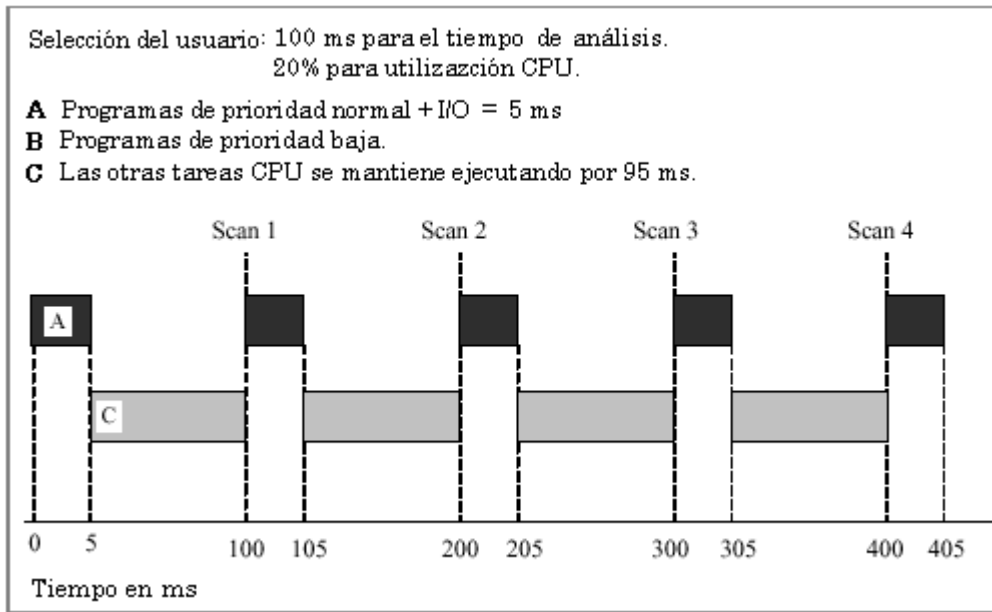
Las figuras en las próximas páginas ilustran varios escenarios de ejecuciones de programas y muestran ejemplos de operación normal y operaciones con saltos scans.

Ejemplos de Scan Normal y de Saltos

En la siguiente figura, la razón del actual CPU% y a utilización de la CPU especificado esta definido por la siguiente relación:

$$0 < \text{razón} < 1$$

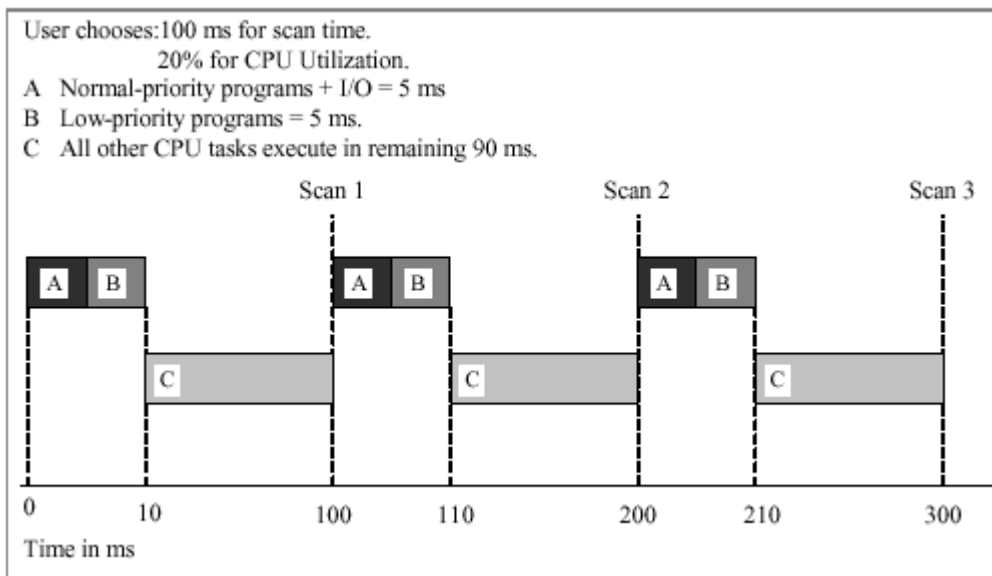
- La operación es normal.
- Ningún scan es saltado.
- No hay errores watchdog.
- No hay programas de prioridad baja.



En la siguiente figura, la razón del actual CPU% y el CPU Utilization especificado esta definido por la siguiente relación:

$$0 < \text{razón} < 1$$

- La operación es normal.
- Ningún scan es saltado.
- No hay errores watchdog.
- Programas de prioridad baja son programados para correr.

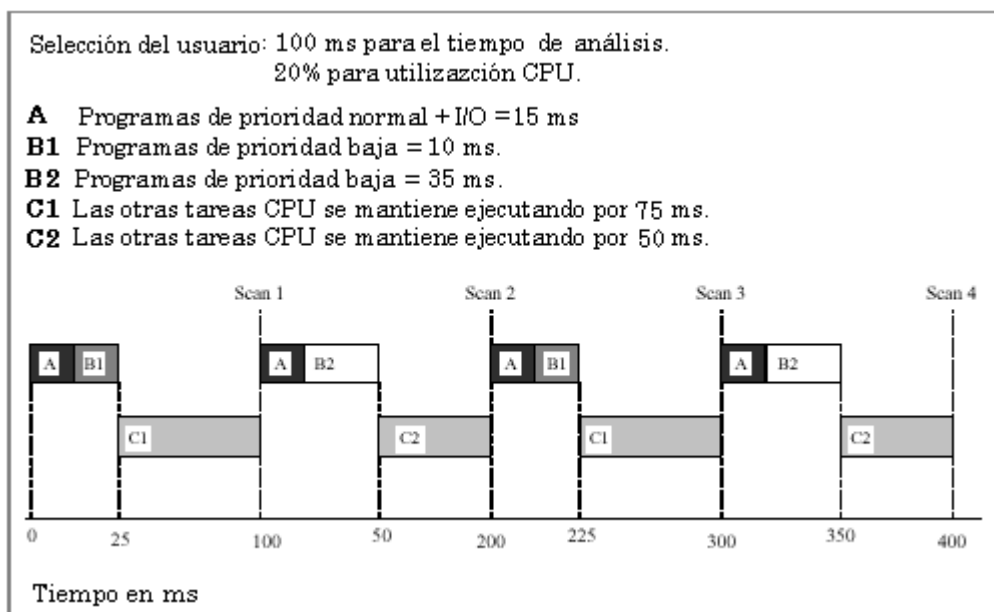


16-16 **CAPITULO 16**

En la siguiente figura, la razón del actual CPU% y el CPU Utilization especificado esta definido por la siguiente relación:

$0 < \text{razón} > 1$

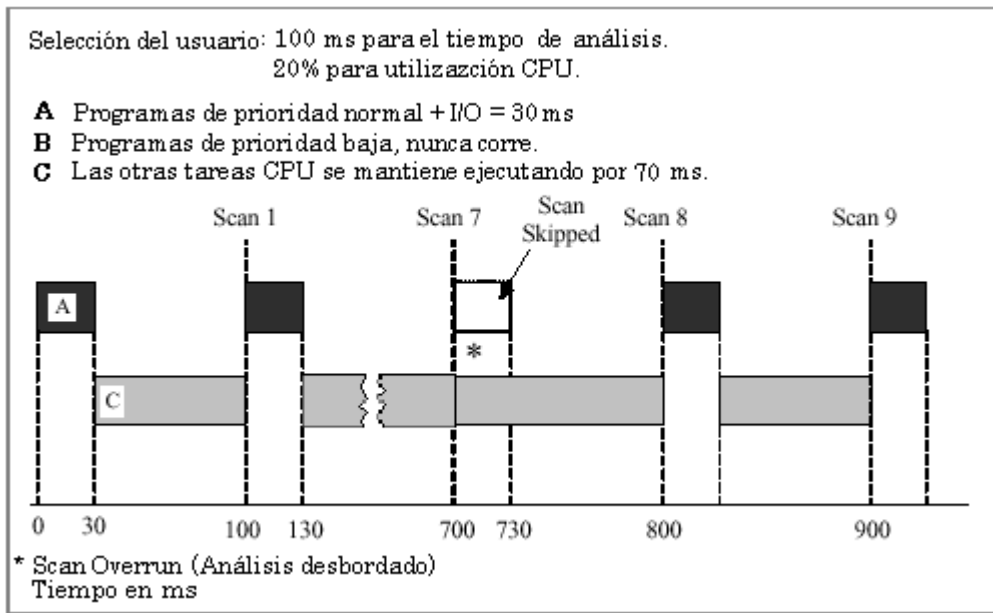
- La operación es normal.
- Ningún scan es saltado.
- No hay errores watchdog.
- Programas de prioridad baja son programados para correr y extendidos el scan.



En la siguiente figura, la razón del actual CPU% y el CPU Utilization especificado esta definido por la siguiente relación:

$1 < \text{razón} > 2$

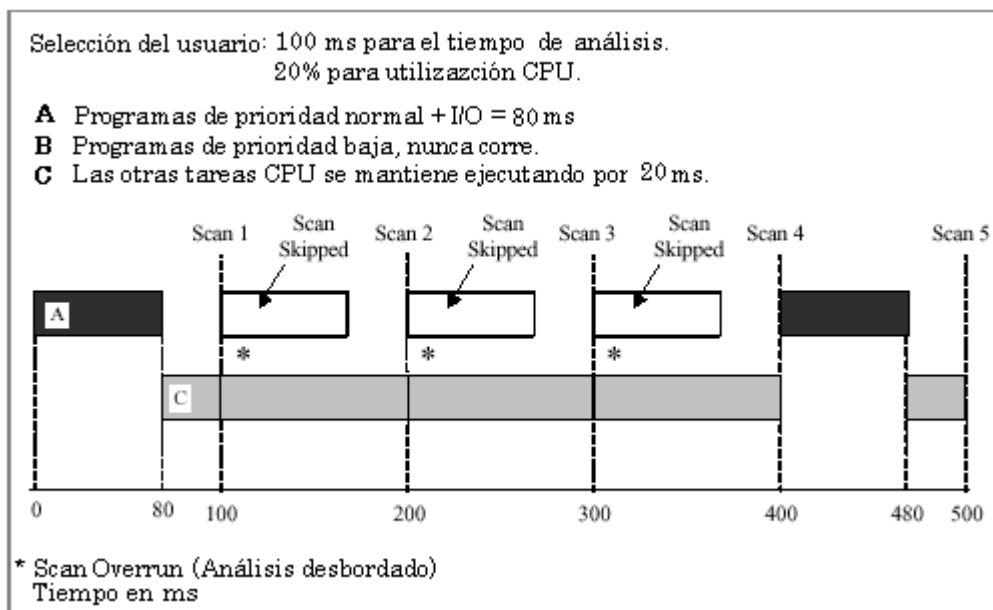
- La operación no es normal.
- Scans son saltados.
- No hay errores watchdog.
- Programas de baja prioridad, cualquiera, nunca correrá.



En la siguiente figura, la razón del actual CPU% y el CPU Utilization especificado esta definido por la siguiente relación:

$$\text{razón} = > 2$$

- La operación no es normal.
- Scans son saltados mas progresivamente.
- No hay errores watchdog.
- Programas de baja prioridad, cualquiera, nunca correrá.



Operaciones Scan y el Watchdog Timer

Un error watchdog ocurre cuando un scan único de la lógica del programa e I/O excedió el intervalo watchdog especificado. Al Runtime Engine hace este chequeo a cada intervalo de scan. Si el watchdog alguna vez expira, entonces al Runtime Engine atenderá para fijar el I/O a un estado seguro y luego tapa abajo. El intervalo watchdog definido es de 10 segundos.

Consideraciones I/O

Tarjetas de la unidad de control, arquitectura del bus de control, y módulos I/O tienen diferentes requerimientos de tiempo, las cuales a menudo son implementaciones específicas. Es posible fijar un tiempo de análisis total dentro de InControl que es mas rápido que estos equipos pueden manejar. Esto es particularmente cierto cuando el tiempo de análisis es menor que 10 ms. Por ejemplo, puede fijar InControl para actualizar un módulo 300Hz I/O en 1000Hz, pero sólo 300 de estas actualizaciones realmente se registran con el módulo de I/O.

Para tiempos de análisis por debajo de 10 ms, es recomendable que realice la configuración inicial y la comprobación en el tiempo de análisis por encima de 10 ms. Luego pruebe ejecutar exitosamente en el tiempo de análisis mas rápido progresivamente, hasta el tiempo de análisis requerido sea archivado.

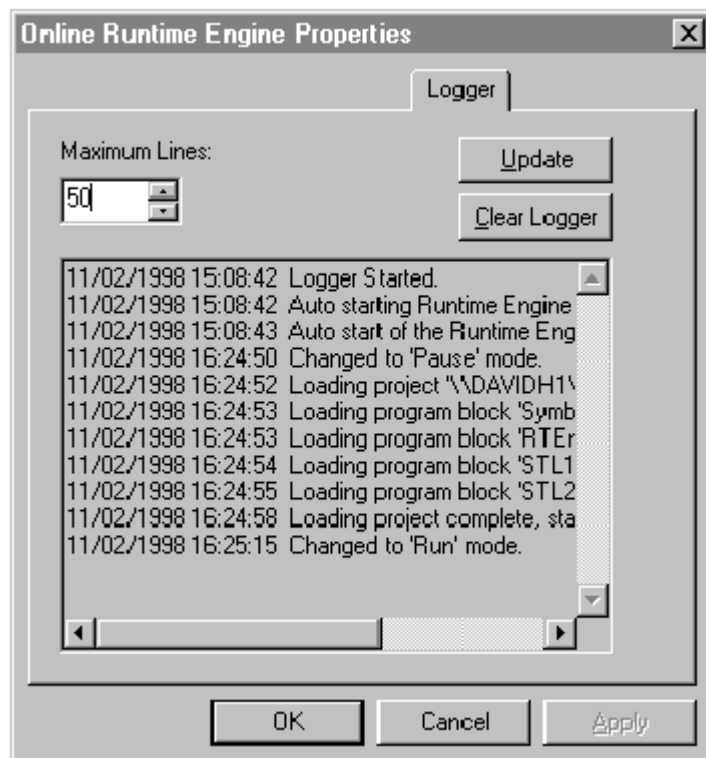
Revise el tiempo de análisis actual en la etiqueta Status del cuadro de diálogo **Runtime Engine Properties**. Reduzca el tiempo de análisis total y el porcentaje CPU hasta que el valor promedio sea lo mas bajo necesario. El tiempo de análisis se fija con relación a la cantidad de tiempo CPU que usa en su proceso. La relación para determinar el tiempo de análisis apropiado esta dado en "Targeting CPU Utilization".

En tiempo de análisis muy rápido (menores que 10ms) la maquina lleva la prioridad de sistema de la llamada del sistema a un timer de ejecución del sistema. Consecuentemente, las variables del sistema RTEngine.ScanLast y RTEngine.ScanMax no son el reflejo exacto del tiempo de análisis I/O actual en una

razón de análisis muy rápida. Además buses y latencias I/O para módulos I/O pueden estar en el rango de 1 - 10 ms. Cuando su aplicación requiera análisis de esta velocidad siempre conecte a un osciloscopio al I/O para obtener medidas de reflejo de ejecución, en ves de contara sobre las variables del sistema para este dato. Contadores InVontrol solamente reflejan tiempo interno.

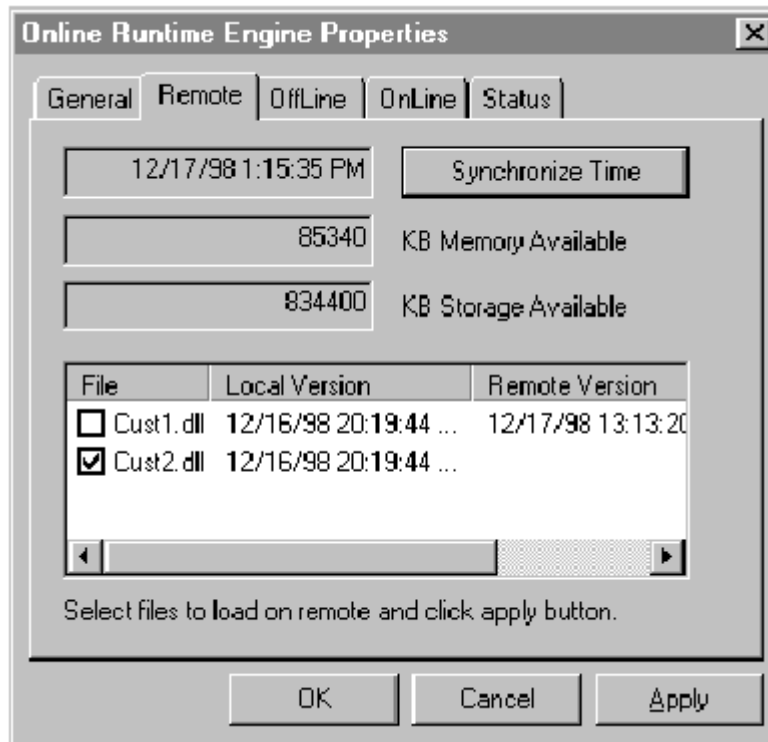
Buscando en el Logger Data

El Logger tab del cuadro de diálogo **Runtime Engine Properties** que se muestra en la siguiente figura. Esta etiqueta aparece en el cuadro de diálogo **Online Runtime properties** cuando la tarjeta en el Runtime Engine esta sobre la plataforma del hardware que no esta usando el sistema operativo Windows NT. Cuando la plataforma del hardware no soporta el Wonderware Logger, toda la información toda la información que al Runtime Engine normalmente envía al Wonderware Logger aparece en cambio en el Runtime Engine Logger.



Revisando Nodos Remotos

El Remote tab del cuadro de diálogo **Runtime Engine Properties** se muestra en la siguiente figura. Esta etiqueta aparece en el cuadro de diálogo **Online Runtime Engine properties** cuando la tarjeta del Runtime Engine esta puesta en nodo remoto.



En el evento que usted necesite transmitir archivos a el nodo remoto desde el nodo local, puede usar el Remote tag para revisar versiones y ara seleccionar archivos para ser descargados. Por ejemplo, usted puede tener un objeto de fábrica diseñado para cliente que quiera usar en el nodo remoto; o podría querer actualizar un driver I/O en el nodo remoto.

Antes que pueda descargar cualquier archivo, ubique copias de los archivos dentro de un subdirectorio del directorio InControl del nodo local. Este subdirectorio debe tener el mismo nombre que el nombre de la tarjeta del nodo remoto. Por ejemplo, su la ruta para InControl en el sistema local es el siguiente:

C:\Program Files\FactorySuite\Common\InControl

Y la tarjeta esta en una plataforma Windows NT, ubique los archivos en el siguiente directorio:

C:\Program Files\FactorySuite\Common\InControl\NT

Si el sistema de la tarjeta es una plataforma Contec, por ejemplo, ubique los archivos en el siguiente directorio:

C:\Program Files\FactorySuite\Common\InControl\Contec

Después que usted copie un archivo dentro del directorio apropiado, el nombre del archivo aparece en el campo **Field** del Remote tab. Seleccione los archivos que usted quiere descargar y pulse **Apply**. Archivos ejecutables con extensión .OCX, .DLL, y :EXE, son registrados automáticamente en el nodo remoto cuando ellos son descargados.


Un archivo no puede ser transferido si esta en uso. Para archivos de programa que están corriendo en el Runtime Engine sobre un nodo remoto, debe detener el programa antes de que pueda transferirlo. Dependiendo del programa y de factores como la longitud del período de tiempo de salida, usted puede tener que esperar varios minutos para que el programa sea descargado.

Nota Un mensaje de error puede aparecer, dependiendo del archivo que esta siendo descargado, la tarjeta del sistema operativo y la manera como este sistema operativo maneja los registros de archivos. Es probable que la descarga y el registro ocurran correctamente; sin embargo, asegúrese de revisar el documento de usuario apropiado de la tarjeta del sistema operativo para mas información.

Eliminando modos de falla y condiciones de error

Esta sección describe el modo de Fault del programa y el Runtime Engine y como limpiar estas condiciones de error.

Modo Fault del Runtime Engine

El monitor del Runtime Engine revisa los estados de la máquina cada 10 segundos. Si el monitor es incapaz de obtener los estados del Runtime Engine, un símbolo de falla es superpuesto sobre el icono del monitor del Runtime Engine. En caso de que el Runtime Engine entre al modo Fault, la variable del sistema RTEngine.Mode es fijado a un valor de seis, y el siguiente símbolo de falla es superpuesto sobre el icono del monitor del Runtime Engine : 

➤ Para limpiar el modo Fault del Runtime Engine :

1. Revise el Wonderware Logger o la ventana Output para mensajes de error.
2. Pulse el monitor del Runtime Engine y seleccione **Exit Monitor**.
3. Restaure el servicio del Runtime Engine .

Para mas información acerca de la modificación de inicio automático del servicio del Runtime Engine, ver Guía de Administración del Sistema de FactorySuite.

4. Descargue el proyecto.

Si un mensaje de error o advertencia es enviado al Wonderware Logger, la variable del sistema RTEngine.Error es fijada en TRUE.

El bloque de funciones RLL ABTAL y el procedimiento de texto estructurado ABORT_ALL también puede fijar el Runtime Engine a modo Fault. Para eliminar el modo Fault en este caso, usted puede fijar 2 y 3 pasos en el procedimiento anterior. En lugar de 4 pasos, fije el Runtime Engine a modo Pause o modo Run.

Programa en Modo Fault

Durante el curso de su ejecución, un programa puede hacer una operación ilegal y entrar a modo Fault. El proceso de validación de InControl le ayuda a evitar algunos errores de programación, así como errores de sintaxis, que resultan en operaciones ilegales. Sin embargo, algunos problemas pueden ocasionar que un programa entre a modo Fault. Por ejemplo, en un índice de serie (array index) puede estar asignado un valor que este fuera de los límites. Un Tirad-party FOE puede desarrollar un valor interno. Un valor de seis puede ser escrito a una variable de sistema program's Mode (desde la ventana Watch, por lógica de programa, etc.) y causar que el programa entre a modo fault.

Si un programa entra a modo Fault, "Fault" parece en la barra de título del programa y en el Project View. InControl fija la variable del sistema Mode para el programa a seis, escribe un mensaje a la ventana Output y el Wonderware Logger, y superpone el siguiente símbolo sobre el icono del Runtime Engine:



Cuando un programa esta en el modo fault, su lógica no es una ejecución prolongada. Un programa fallado no afecta directamente la ejecución de otros programas o POU's del proyecto.

➤ Para restaurar un programa en el modo Fault:

1. Eliminar Fallas. Sobre el menú **Runtime Engine Monitor**, pulse **Clear Faults**. Este fija el programa en modo Pause. Si otros programas están corriendo, al Runtime Engine permanece en modo Run.
2. Fije el programa a modo Run.
3. Si el programa retorna a modo Fault, usted necesita revisar la lógica del programa y posiblemente la integridad del hardware. Si usted cambia el programa, descárguelo de nuevo y córralo.

Para mas información acerca de Información de proyectos FOE para eliminar fallas, ver "Usando el Objeto de Información de Proyecto " en el capítulo "Usando el Editor de Objetos de InControl Factory".

Errores I/O y de Hardware.

Para errores I/O y otros errores de hardware, corrija la condición de falla y luego pulse **Clear Faults** sobre el menú **Runtime**.

Configuración del Inicio del Servicio del Runtime Engine

Al Runtime Engine esta en servicio Windows NT y es configurado para empezar automáticamente cuando enciende la unidad de hardware. Típicamente, no es necesario detener el servicio del Runtime Engine . Si ningún proyecto esta corriendo, al Runtime Engine está inactiva y el procesador consume muy poco tiempo. Sin embargo, es necesario detener el servicio del Runtime Engine para instalar el software, arreglos de problemas del sistema, energizar el ciclo del sistema sin reiniciar automáticamente al Runtime Engine , etc. Usted puede configurar sus sistema para que al Runtime Engine no inicie automáticamente después que usted reinicie el sistema operativo Windows NT.

MANEJO DE FALLAS DE ENERGÍA

Usted puede configurar InControl para que cierre y reinicie automáticamente de una manera ordenada. Esto le permite diseñar una reasunción desatendida de control de su proceso de fábrica en el evento de una falla de energía.

Usando un Suministro de Energía Continuo

InControl provee dos utilidades (**ICPwFail** e **ICPwOn**) y una variable del sistema (**RTEngine.PowerFail**) para reportar la pérdida y restauración de energía a el sistema. Estas características son diseñadas para interactuar con cualquier suministro de energía continuo thid-party (UPS) que es capaz de reportar una pérdida de energía.

- La utilidad ICPwFail fija la RTEngine.PowerFail a TRUE cuando ICPwFail recibe notificación de la UPS de que la energía se ha perdido.

- La utilidad ICPwFail fija la RTEngine.PowerFail a FALSE cuando ICPwFail recibe notificación de la UPS de que la energía ha continuado.

Es importante notar que no es suficiente simplemente conectar el sistema InControl a un UPS para monitorear el estado de energía dentro de su aplicación. El UPS debe tener una capacidad de programación que puede ser usada para darle instrucciones para responder a cambios de estados de energía de la siguiente manera.

- Cuando el software UPS determina que la energía ha fallado, el ejecuta la utilidad ICPwFail. La utilidad comunica con InControl, permitiendo cualquier acción que tenga programado para suceder.
- Cuando el software UPS determina que la energía ha sido restaurada, el ejecuta la utilidad ICPwOn. La utilidad comunica este cambio de estado a InControl, permitiendo cualquier acción que usted haya programado para que sucediera.

El servicio UPS que es parte del sistema operativo Windows NT 4.0 provee solamente la primera capacidad, corriendo un archivo de comando en respuesta a una falla de energía. Es más, el solamente ejecuta el archivo de comandos varios segundos antes que el sistema se cierre. En mas instancias, esto no provee suficiente tiempo para ser útil, excepto por el propósito de automatización industrial simple. Un número de productos tirad-party están disponibles para proveer la capacidad de correr ambos archivos de comandos.

Uso de las Variables del Sistema

Use la variable del sistema **RTEngine.PowerFail** para coordinar el cierre y el inicio. La variable Mode automáticamente creado para cada programa y para cada tablero I/ O también puede ser útil en la planificación de un cierre ordenado. Use las variables del sistema **RTEngine.FirstScan** y/o **RTEngine.ScanOnAutoStart** para automatizar la restauración control de procesos por causa del flujo de programas a una sección para procedimientos de inicialización. Configure al Runtime Engine

para reasumir la ejecución en el modo apropiado: último modo, pausa, detención, etc., como se describe en “Revisando propiedades Generales”.

Usted puede escribir o forzar la variable **RTEngine,PowerFail** desde la ventana Watch, y cualquier programa que use la variable reflejará el nuevo valor en su ejecución. Note que debe restaurar esta variable manualmente, desde la ventana Watch.

Variable Forzadas y Retentivas, y Fallas de Energía

Si usted revisa del cuadro de diálogo **Retentive Value** sobre del cuadro de diálogo **Symbol Properties** de diálogo de la variable, usted tiene la opción de copiar el valor de una variable a el disco duro.

InControl proporciona tres maneras por el cual puede especificar para que ocurra la copia.

- Si al Runtime Engine se cierra durante la falla de energía, los valores de retención y variables forzadas son copiadas a el disco duro. Cuando al Runtime Engine reinicia, estos valores son restaurados a las variables.
- Los valores de variables forzadas y retentivas no son guardadas a menos que este usando un UPS inteligente con el sistema y lo tenga configurado para señalar una falla de energía de InControl.
- Usted puede configurar InControl para guardar variables retentivas y forzadas a el disco duro periódicamente. La frecuencia predefinida de cero descactiva esta característica. Usted puede cambiarla en el cuadro de diálogo **Runtime Engine Properties**, descrito en “Fijando tiempos de análisis”.
- Usted puede diseñar códigos en un programa para gravar el valor de retención y la variable forzada en demanda. Para mas información ver “Usando el Project Information FOE” en el capítulo “Usando el editor de Objetos de Fábrica de InControl”.

Los valores son restaurados cuando usted recarga el proyecto o si tiene configurada el Runtime Engine para reiniciar automáticamente (modos Last, Pause,

Run) después que un sistema reboto. Para mas información, ver “Backing Up Retentive / Forced Variables”.

Reinicio Automático de Proyectos

En el momento en que la unidad de hardware de InControl pierde energía, el servicio del Runtime Engine empieza automáticamente cuando la energía es reasumida. Al Runtime Engine retorna a el modo especificado en la etiqueta General del cuadro de diálogo **Runtime Engine Properties**. Esta el cuadro de diálogo se muestra abajo y se describe en “Revisando Propiedades Generales”.



Configuración del Modo de Reinicio (Restart)

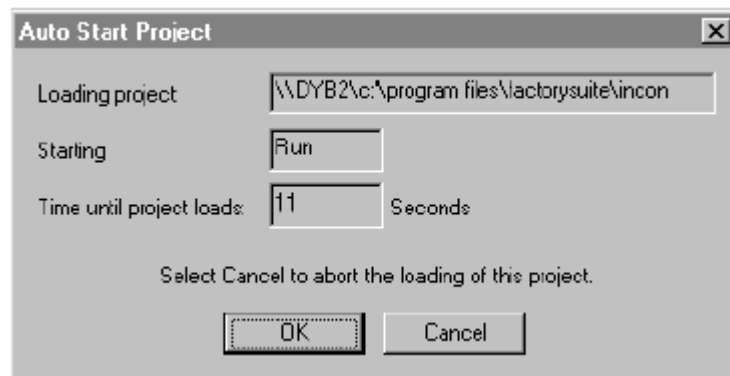
Escoja de las siguientes opciones, cuál esta seleccionada en el archivo **Restart Mode**:

- Last – Recargua el proyecto anterior. Al Runtime Engine entra al modo que estuvo antes que la unidad de hardware fuera booted.
- Note que cuando usted hace un cierre manual (comparado al cierre ocurrido debido a la falla de energía) InControl automáticamente fijal Runtime Engine en Stop. Si usted quiere que al Runtime Engine recargue y corre el último proyecto, seleccione la opción Run.

16-28 **CAPITULO 16**

- None – El servicio del Runtime Engine empieza, pero ningún programa es cargado en el Runtime Engine .
- Pause – Recarga el último proyecto. Al Runtime Engine entra a modo Pausa.
- Run – Recarga el último proyecto. Al Runtime Engine continua corriendo el proyecto que estuvo corriendo antes que la unidad de hardware fuera booted.

Si al Runtime Engine empieza a introducir un modo un después que la unidad del hardware reinicies, tiene la opción de cancelar el inicio automático de un proyecto. Cuando el servicio del Runtime Engine reinicia, del cuadro de diálogo **Auto Start project** aparece.



Si usted no quiere que el proyecto reasuma la ejecución, pulse **Cancel** para fijar al Runtime Engine a modo Stop.

Copia de seguridad de Variables Forzadas y Retentivas

Si al Runtime Engine cierra durante una fallas de energía, los valores de variables forzadas y retentivas son copiadas en el disco duro. Cuando al Runtime Engine reinicia, estos valores son restaurados a la variable.

Los valores de variables forzadas y retentivas no son guardadas a menos que este usando una UPS inteligente con el sistema y lo tiene configurado para señalar fallas de energía.

Para mas información acerca de las falla de energía, ver “Manejo de Fallas de Energía”.

EJECUCIÓN DE MÚLTIPLES PROYECTOS

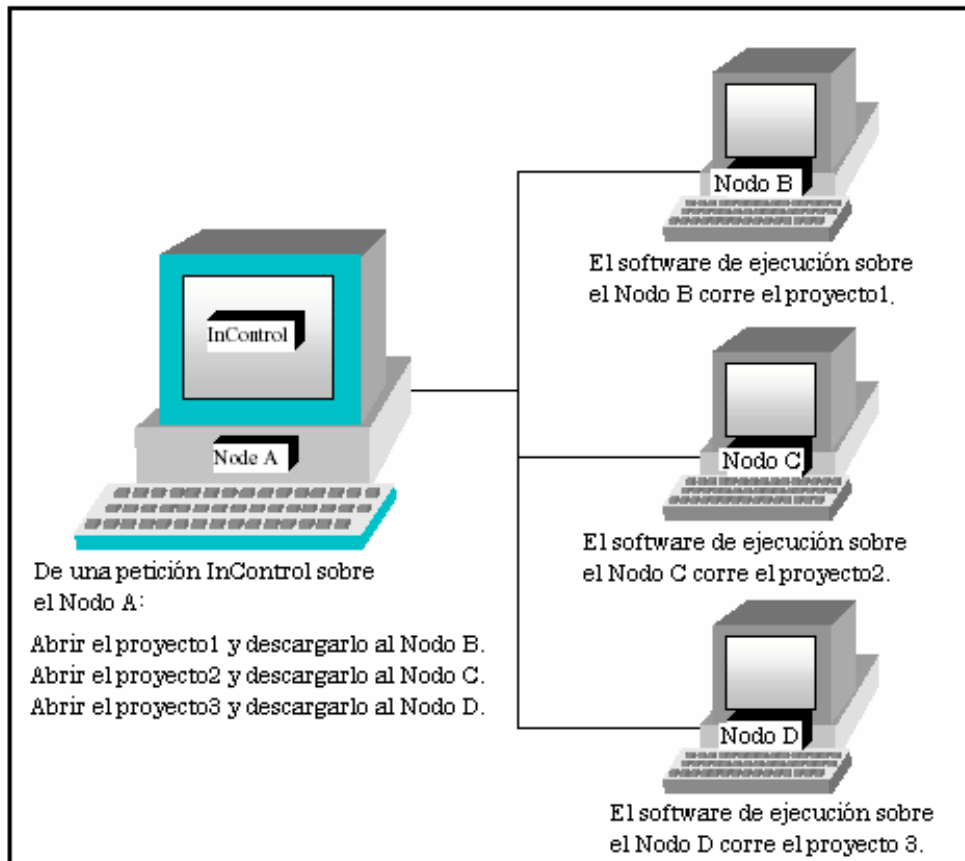
InControl soporta control distribuido, el cual lo habilita para descargar y correr un proyecto sobre una el Runtime Engine localizada en otro computador. Estas líneas solicitan:

- Un proyecto puede ser abierto para editar en un ambiente de desarrollo a la vez. Esto previene la corrupción de archivos de proyectos. Una instancia del ambiente de desarrollo InControl puede abrir solamente un proyecto a la vez.
- El ambiente de desarrollo sobre un nodo puede descargar un proyecto al Runtime Engine localizada en el nodo conectado.
- El ambiente de desarrollo sobre cualquier nodo puede conectar una ejecución de proyecto sobre otro nodo, monitorear las variables desde la ventana Watch, detener e iniciar el proyecto.
- Un nodo soporta una instancia del Runtime Engine .

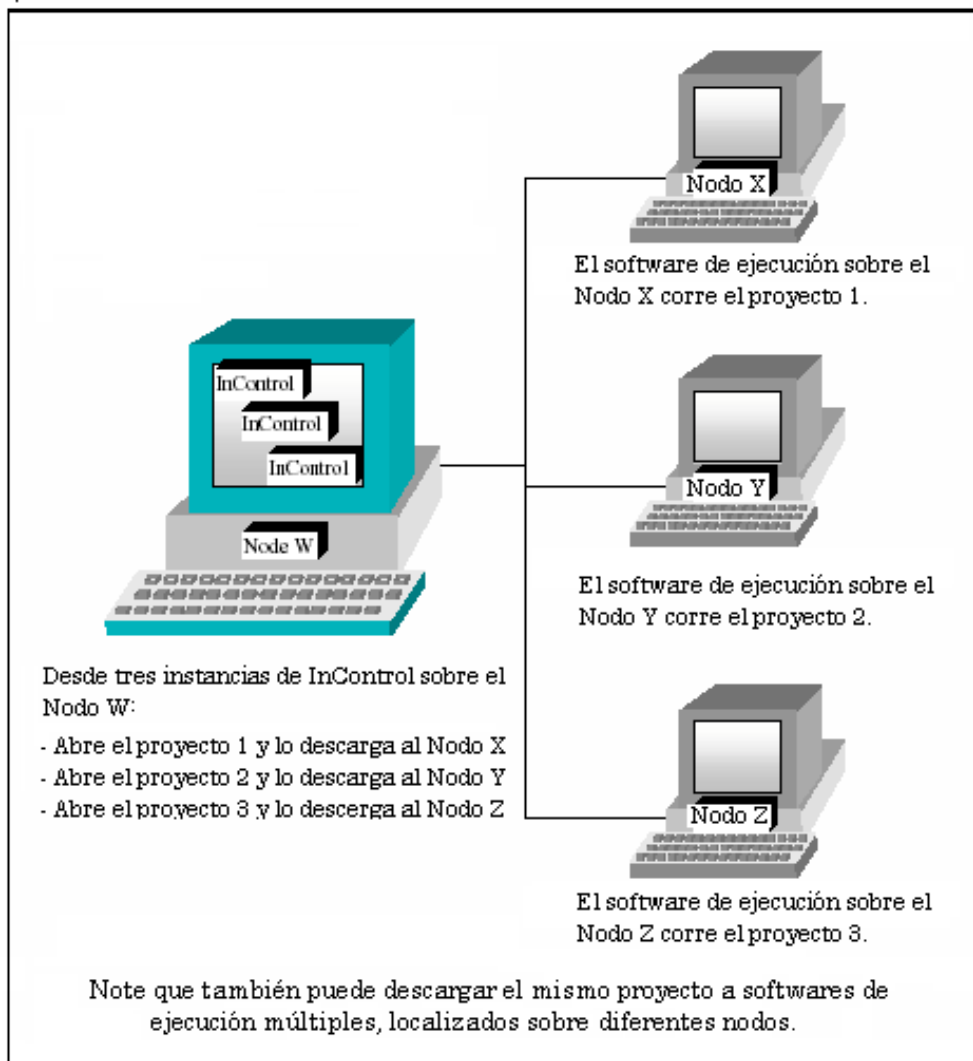
Nota Durante la instalación, usted puede elegir instalar solamente al Runtime Engine sobre un nodo.

Algunos ejemplos de control distribuido están ilustrados en las siguientes figuras. Otros escenarios son posibles.

- Sobre el nodo A, usted puede abrir un proyecto y descargarlo a el nodo remoto B. Entonces, después de la conmutación a un segundo proyecto (pulse **Project** sobre el menú **File** y seleccione el proyecto), usted puede descargar este proyecto a el nodo remoto C. Luego puede abrir otro proyecto y descargarlo a D.



Sobre el nodo W, usted puede correr una instancia de InControl, abrir un proyecto, y descargarlo a el nodo remoto X. Luego puede correr una segunda instancia de InControl. Pulse **Start** sobre la barra de tareas, pulse **Programs**. Debajo de **Wonderware Factory Suite**, seleccione **InControl**. Abra un Segundo proyecto y descargue este proyecto al nodo remoto Y. Siga el mismo procedimiento para el nodo Z.



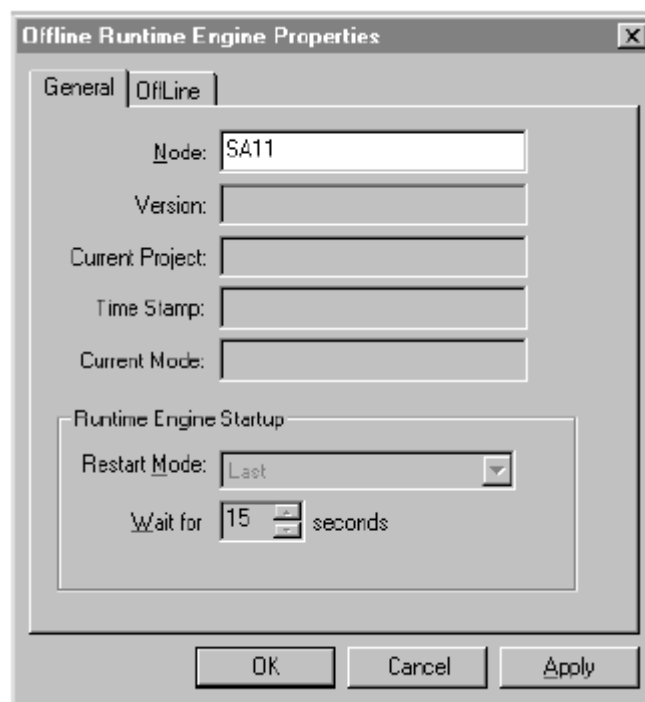
Es recomendable que planee cuidadosamente cómo almacenar los archivos fuentes para un proyecto. Si usted descarga un proyecto desde el nodo a una el Runtime Engine sobre el nodo B, solamente un archivo binario es cargado al nodo B. Si luego borra el proyecto sobre el nodo A, el código fuente es va y el proyecto no puede ser editado.

Considere tener todos los proyectos almacenados o archivados en un servidor central y revisar una copia cuando necesite hacer cambios. Como una alternativa, podría hacer una copia de seguridad de un proyecto en el mismo nodo para el cuál usted descarga el proyecto.

Configuración de una Conexión al Nodo Remoto

Designe el nodo remoto sobre la etiqueta General del cuadro de diálogo **Runtime Engine Properties**.

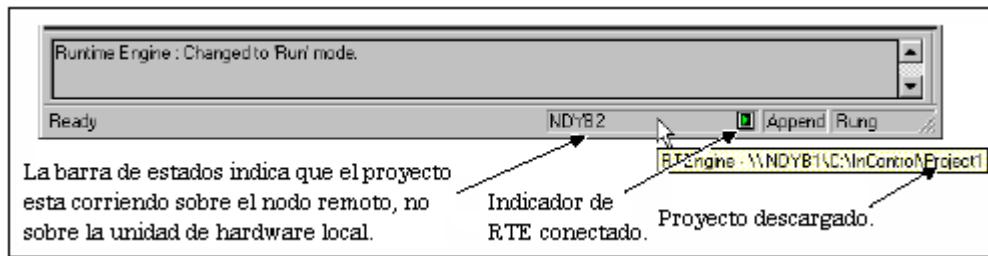
- Para designar un nodo remoto:
 1. Desconecte el ambiente de desarrollo del Runtime Engine
 2. Sobre el menú **Runtime**, pulse **Configure**.
 3. Seleccione la etiqueta **General** para visualizar las propiedades de desconexión.



4. Entre el nombre del nodo remoto en el campo **Node** y pulse **OK**.

Nota: Deje el campo en blanco para indicar al computador local.

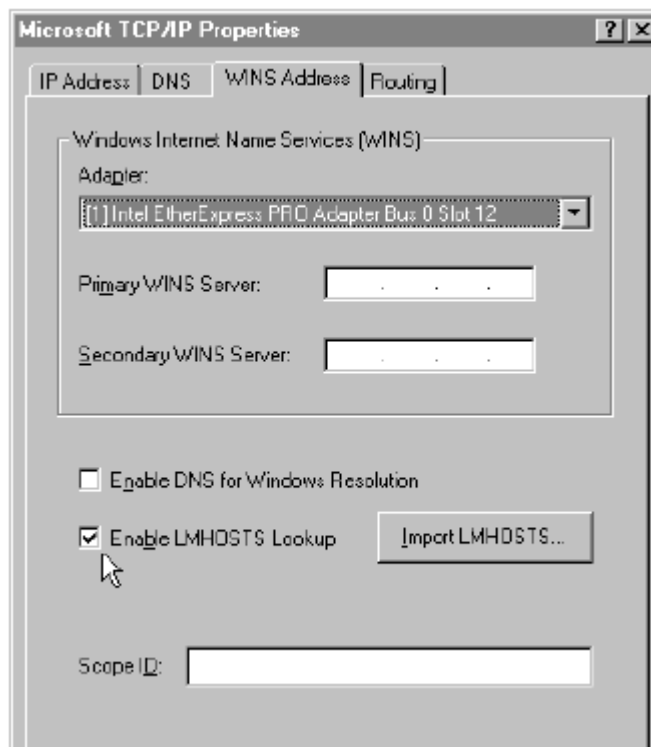
La próxima vez que pulse **Connect** en el menú Runtime, el ambiente de desarrollo se conecta al Runtime Engine localizada en el nodo remoto.



Usted puede entrar el nombre del Runtime Engine o la dirección TCP/ IP del computador en el campo Node del cuadro de diálogo **Runtime Engine Properties**. Si usted usa la dirección TCP/ IP y tiene dificultades en la conexión al nodo remoto, habilite la red Microsoft TCP/ IP LMHosts.

➤ Para habilitar LMHosts:

1. Pulse **Start** en la barra de tareas, luego seleccione **Settings** y pulse **Control Panel**.
2. Pulse **Network** y seleccione la etiqueta Protocols.
3. Seleccione TCP/ IP y pulse Properties.
4. Sobre la etiqueta WIN Ardes, revise del cuadro de diálogo de supervisión Enable LMHosts Lookup, como se muestra en la siguiente figura.



5. Reinicie el computador para que la nueva fijación sea efectiva.

Transferencia y Registro de Datos de Proyecto

Cuando los nodos están enlazados por conexiones de red, la descarga de un proyecto a un nodo remoto es cumplida por el ambiente de desarrollo InControl. El proceso es automático y requiere solamente que usted especifique un nodo remoto en el cuadro de diálogo de dialogo **Properties**.

La utilidad de validación InControl crea un archivo llamado **RTEngine.dat** que contiene todos los datos compilados de un proyecto. Usted puede usar el archivo para los siguiente propósitos:

- Haga una copia de **RTEngine.dat** y guárdelo para propósitos de almacenamiento.
- Copie **RTEngine.dat** a otro nodo donde pueda cargarlo y correrlo. Esto le permite distribuir proyectos pre-built que no incluye el código fuente. Cuando los nodos no son vinculados, puede transferir **RTEngine.dat** a un nodo remoto usando un proceso manual.

Tenga en cuenta las siguientes consideraciones.

- No es necesario conectar al Runtime Engine antes de la creación de RTEngine.dat.
 - Cuando crea un RTEngine.dat, es almacenado en un nodo local en el mismo directorio como un proyecto.
 - Antes que pueda cargar el archivo al Runtime Engine (en cualquier nodo local o nodo remoto), usted debe moverlo al directorio de manejo del Runtime Engine . Para plataformas NT, este es el subdirectorio NT, el cual esta inmediatamente debajo del directorio dondel archivo RTEngine.exe esta localizado.
- Para hacer un archivo de almacenamiento:
1. Valide el proyecto. En el cuadro de diálogo **Validate Project**, revise del cuadro de diálogo de chequeo Create **Exacutable Archiveof Project**. Este crea el archivo (almacenado) llamado RTEngine.dat.
 2. Localiza el archivo, el cuál esta almacenado en el mismo subdirectorio como un proyecto.

- Para cargar el archivo en al Runtime Engine :
 1. Verifique que el nodo este en la plataforma apropiada.
 2. Fije al Runtime Engine a el modo Stop.
 3. Mueva el archivo al directorio de manejo, definido abajo.
 4. haga right-click al icono del monitor del Runtime Engine y pulse **Reload Project**.

Por ejemplo, si el proyecto llamado **BldCtrl** tiene la siguiente parte:

C:\Program Files\FactorySuite\InControl\Projects\BldgCtrl

Usted puede localizar el RTEngine.dat en el subdirectorío BldgCtrl después de validar el proyecto.

Para una plataforma Windows NT, debe ubicar RTEngine.dat en el siguiente directorio antes que pueda cargarlo:

..\InControl\NT

Configuración I/O en el Nodo Remoto

Algunos de los drivers I/O soportados actualmente le permiten abrir y editar la configuración I/ O de un proyecto que correrá sobre un nodo remoto. El driver debe estar instalado en ambos nodos, el remoto y el local donde hace la configuración. Además, para algunos drivers, prodrías necesitar instalar la scanner board en el nodo local también como en el nodo remoto.

Nota: Algunos drivers I/ O proveen utilidades que puede usar para hacer una configuración automática y/o correr diagnósticos en línea. Algunos de estos driver lo requieren a usted para usar estas utilidades en el nodo remoto el mismo. Drivers más nuevos soportan configuración automática remota y diagnósticos en línea. Para llevar a cabo esto, aquellos drivers pueden descargar ellos mismos al Runtime Engine remota. En este caso, el driver entra a modo Loaded (Mode system variable = 8.)

CAMBIO DE LA CLAVE EN EL REGISTRO DEL SISTEMA

Esta sección describe cambios que puede decidir hacer en el registro del sistema.

Advertencia: Hacer cambios incorrectos en el registro Windows NT puede resultar en operaciones impredecibles por InControl y equipos I/O. Esto tiene el peligro potencial de lesiones o muerte de personal y/o daños al equipo. Siempre verifique que los cambios que usted hace en las claves de registro no afectará adversariamente la operación del computador o la aplicación que intentó correr antes probar equipos de campo de fábrica de control.

Asegúrese que Wonderware no provee soporte técnico para problemas que puedan surgir como un resultado de cambios que hace a el registro.

Cambios en la fijación de registros FOE

Si la clave ThreadingModel para un FOE no tiene el valor string correcto, usted no puede instalar el objeto en un proyecto InControl. Por otra parte, la clave ThreadingModel debe ser "Both." Para un FOE y así operar eficientemente InControl. Si la clave es "Apartment", el acceso a las propiedades será significativamente mas lento. Adicionalmente, es mas probable que el FOE afecte adversamente el determinismo del Runtime Engine . Esta sección describe cambiar la clave Threading.

Advertencia: Si el FOE no esta designado para correr bajo el modelo free-threaded, cambiar el valor del ThreadingModel puede ocasionar que el objeto corra de una manera impredecible. Por ejemplo, puede ocurrir un bloqueo, o valores adulterados pueden ser leídos, con un riesgo potencial de muerte o lesiones al personal y/o daño al equipo. Revise y verifique que los objetos corran correctamente después del cambio de valore ThreadingModel antes de usar para controlar la maquinaria de piso de fábrica. Porque incluso revisiones exhaustivas no pueden revelar problemas que pueden ocurrir en un ambiente del Runtime Engine , es altamente recomendable que consulte con el diseñador del FOE para verificar que lo operará correctamente con InControl.

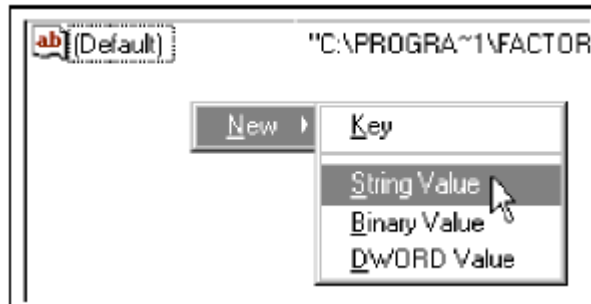
- Para editar el registro:
 1. Pulse **Start** sobre la barra de tareas, luego pulse **Run**.
 2. Teclee en **regedit** en el cuadro de diálogo **Run**. El Registry Editor se abre.
 3. Busque el nombre del FOE. Por ejemplo, para buscar PID FOE, pulse **Find** sobre el menú **Edit** y entre “Wonderware PID” en el campo **Find What**.
 4. Expanda el fólder abierto en el árbol de registro y haga doble clic en la clave **ImprocServer32**.
 5. Verifique que el valor string de Threading es “Both” como se muestra bajo.



6. Si necesita editar el valor, haga doble clic en el nombre del valor y entre **Both** en el campo **Value Data**.

Si un valor ThreadingModel no existe, usted puede crear uno nuevo.

1. Haga right-click y crea un nuevo valor string, com se muestra abajo.



2. Entre ThreadingModel para el nombre del valor.
3. Haga doble clic en el nuevo valor y entre **Both** en el campo **Value Data**.

Mostrando Advertencia de Compilador

Si su programa tiene cualquier código para revisar o simular su proceso y le escribe a variables de entrada I/O, un mensaje de advertencia aparece cuando los programas son validados. Usted puede tener mensajes de error en problemas de compilación ene vez de una advertencia. Esta puede ser hecha para un único

programa, o un hecho específico a una cuenta de usuario y afectar todos los proyectos.

Para hacer el cambio en el nivel de programa, incluya la siguiente declaración #pragma en un SFC o programa de Texto estructurado:

```
#pragma IOWriteError
```

Para mas información acerca de la declaración #pragma, ver “#pragma” en el capítulo “**LENGUAJE DE TEXTO ESTRUCTURADO**”.

Hacer el cambio para una cuenta de usuario, usted necesita fijar un bit en la clave CPOption debajo de la siguiente parte en el registro.

```
HKEY_CURRENT_USER\  
    SOFTWARE\  
        WONDERWARE\  
            INCONTROL
```

Advertencia: Cambios incorrectos en el registro Windows NT puede resultar en operaciones impredecibles por InControl o equipos I/O. Esto tiene el riesgo potencial de lesiones o muerte del personal y/o daños al equipo. Siempre verifique que los cambios que hace en las claves de registro no afectará adversamente la operación para controlar equipos de campo de fábrica. Tenga en cuenta que Wonderware no provee soporte técnico para problemas que pueden surgir como un resultado de cambios que usted hace al registro.

Si la tecla CPOption no existe, crea una nueva tecla (valor DWORD) y etiquételo CPOption. Para que el compilador emita una advertencia, fije el bit 6 en 0. este es el valor predeterminado. Para que el compilador emita un error, fije el bit 6 en 1.

Por ejemplo, asumiendo que ningunos otros bits están fijados en 1, escriba 20H a la clave CPOption para hacer que el compilador emita un mensaje de error en vez de una advertencia. Usted puede no tener todos los mensajes emitidos, pero esto puede ser hecho solamente con la declaración #pragma.

SOPORTES DE CONTENIDO, MODALIDAD Y TIEMPO

Al Runtime Engine reporta información de Valor/ Tiempo/ Calidad (V/T/Q). Aplicaciones, como InTouch e IndustrialSQL, que puedan conectarse a InControl con el SuiteLink o protocolo DDE 3, podrá obtener estos datos.

Los valores de datos del Runtime Engine , así como etiquetas de memoria, tienen un timestamp (tiempo de sello) basado en el último scan (análisis). La calidad es reportada como se sigue:

- Good indica que al Runtime Engine no está en modo Fault.
- Bad indica que al Runtime Engine está en modo Fault.

El **timestamp** y calidad para valores de datos desde un procesador de instrucción (IP), así como un control ActiveX o un driver I/O, es determinado por el IP. La calidad es determinada como a continuación, basado en modo program:

- Modo Stopped – La calidad es reportada como Bad.
- Modo Fault – La calidad es reportada como Bad y el equipo tiene fallas.
- Idle -- Calidad es reportada como Uncertain, y Stale (gastado).

Cuando usted está usando el protocolo DDE 3, actualizaciones son enviadas a los clientes cuando cualquiera de los valores o la calidad cambia. Sin embargo, si usted está usando el protocolo DDE 2, solamente cambios de valores son transmitidos.

INGRESO DE LA CONFIGURACIÓN DEL VISOR DE SUCESOS

El Event Viewer de Windows NT monitorea eventos como ellos ocurren en su unidad de hardware. El recibe mensajes desde el Runtime Engine y drivers I/O, por ejemplo. Si el espacio del Buffer que ha localizado para mantener los mensajes se llena, el servicio del Runtime Engine puede detenerse.

Advertencia: Autorizar al registro Event Viewer ponerse lleno e inhabilitado para recibir mensajes de sistema podría causar que el servicio del Runtime Engine y todos los programas se detengan inesperadamente. Esto puede resultar en una operación impredecible para equipos y causar muerte o lesiones al personal y/o daños al equipo. Si usted recibe desde el Event Viewer indicando se esta llenando, lleve la acción limpiar al registro. Es recomendable que seleccione la opción **Overwrite Evnt as Needed** en el cuadro de diálogo **Event Log Estting**.

PALABRAS RESERVADAS

Las palabras mostradas en la siguiente tabla son palabras reservadas InControl. Evite usar cualquiera de estas palabras reservadas en sus programas.

ABORT_ALL	ABS	ABTAL
ACOS	ACTION	ADD
AND	ANDN	ANDT
ANDTN	ANY	ANY_BIT
ANY_DATE	ANY_INT	ANY_NUM
ANY_REAL	APPENDFILE	ARRAY
ARRAY_TO_STRING	ASIN	AT
ATAN		
BCD_TO_INT	BEGIN	BEGIN_IL
BOOL	BREAK	BY
BYTE		
CAL	CALC	CALCN
CASE	CD	CLK
CLOSEFILE	CONCAT	CONFIGURATION
CONSTANT	COPYFILE	COS
CTD	CTU	CTUD
CU		
D	DATE	DATE_AND_TIME
DATE_TO_REAL	DELETE	DELETEFILE
DINT	DIV	DO
DS	DT	DWORD

ELSE	ELSEIF	ELSIF
EN	END	END_ACTION
END_CASE	END_CONFIGURATION	END_FOR
END_FOR_NOWAIT	END_FUNCTION	END_FUNCTION_BLOCK
END_IF	END_IL	END_PROGRAM
END_REPEAT	END_REPEAT_NOWAIT	END_RESOURCE
END_STEP	END_STRUCT	END_TRANSITION
END_TYPE	END_VAR	END_WHILE
END_WHILE_NOWAIT	EQ	EXIT
EXP	EXPT	
F_EDGE	F_TRIG	FALSE
FB	FILE	FIND
FOR	FROM	FUNCTION
FUNCTION_BLOCK		
GE	GLOBAL	GOTO
GT	-	-
IF	IN	INCLUDE
INITIAL_STEP	INSERT	INT
INT_TO_BCD	INT_TO_REAL	INT_TO_STRING
INTERVAL		
JMP	JMPC	JMPCN
L	LD	LDN
LE	LEFT	LEN
LIMIT	LINT	LN
LOG	LREAL	LT
LWORD	-	-
MACROSTEP	MAX	METHOD
MID	MIN	MOD
MODE	MOVE	MSGWND
MUL	MUX	

N	NE	NEWFILE
NOT	NOW	
OF	ON	OPENFILE
OR	ORN	ORT
ORTN	OUT	
P	POW	PRIORITY
PROGRAM	PT	PV
PW		
QD	QU	
R	R_EDGE	R_TRIG
R1	READ_ONLY	READ_WRITE
READFILE	REAL	REAL_TO_DATE
REAL_TO_INT	REAL_TO_STRING	REAL_TO_TIME
REPEAT	REPLACE	RESOURCE
RET	RETAIN	RETC
RETCN	RETURN	REWINDFILE
RIGHT	ROL	ROR
RS	RUNG	
S	S1	SCAN
SD	SEL	SHL
SHR	SIN	SINGLE
SINT	SL	SQRT
SR	ST	START
STEP	STN	STRING
STRING_TO_ARRAY	STRUCT	STT
STTN	SUB	
T	TAN	TASK
THEN	TIME	TIME_OF_DAY
TIME_TO_REAL	TMR	TO
TOD	TODAY	TOF

TON	TP	TRANS
TRANSITION	TRUE	TRUNC
TYPE	UDINT	UINT
ULINT	UNTIL	USINT
VAR	VAR_ACCESS	VAR_EXTERNAL
VAR_GLOBAL	VAR_IN_OUT	VAR_INPUT
VAR_OUTPUT		-
WHILE	WITH	WORD
WRITEFILE		
XOR	XORN	-

TIPOS DE DATOS

Este apéndice proporciona una referencia rápida a los tipos de datos soportados por InControl. Para mas información acerca de los tipos de datos, ver “Tipos de datos de Variables” en el capítulo “Definiendo Variables”.



CONTENIDO

- CATEGORÍAS DE TIPOS DE DATOS
- RANGOS DE TIPOS DE DATOS

This product is licensed to:

Company:	N/A
SN:	N/A
Expires:	N/A
Locked To:	N/A

7.1.0.1337

CATEGORÍA DE TIPOS DE DATOS

Todos los tipos	Grupo	Subgrupo	Tipo de Dato	
ANY	ANY_NUM	ANY_REAL	LREAL	
			REAL	
		ANY_INT		DINT
				INT
				SINT
				DWORD ¹
				WORD ¹
		BYTE ¹		
	ANY_BIT		DWORD	
			WORD	
			BYTE	
			BOOL	
	ANY_DATE		DT (fecha-tiempo)	
			DATE	
			TOD	
			STRING	
			TIME	
		FILE ²		
		TMR ²		
		User-Defined		
<p>Nota: Los tipos de datos <i>LINT</i>, <i>ULINT</i>, y <i>LWORD</i> no están actualmente soportados por InControl.</p> <p>1 Los tipos de datos <i>UDINT</i>, <i>UINT</i> y <i>USINT</i> son equivalentes a los tipos de datos <i>DWORD</i>, <i>WORD</i> y <i>BYTE</i> respectivamente. Una mejora de InControl a los tipos de datos <i>ANY_BIT</i> hace que los tipos datos <i>UDINT</i>, <i>UINT</i> y <i>USINT</i> sean innecesarios.</p> <p>2 Mejora de la especificación IEC 61131-3.</p>				

Advertencia: **IEC-61131** no soporta la combinación de números enteros registrados y no registrados en los cálculos matemáticos. Si usted combina números enteros registrados y no registrados, el resultado de la operación matemática tal vez no sea la que usted espera, lo que puede tener un riesgo potencial de muerte o lesiones al personal y/o daños al equipo. Evite el uso de expresiones que combine números enteros registrados y no registrados.

RANGOS TIPOS DE DATOS

Tipo de Dato	Rango
LREAL	-1.79769313486231 E308 (negativo) a +1.79769313486231 E308 (positivo) e incluye cero. Cuando usted se comunice con el Runtime Engine usando una interfase SuiteLink/DDE, incluyendo InTouch, los tipos datos LREAL de 64 bits son transmitidos con una precisión de 32 bits.
REAL	-3.402823 E38 (negativo), to +3.402823 E38 (positivo), e incluyendo el cero.
DINT	-2147483648 to +2147483647
INT	-32768 to +32767
SINT	-128 a +127
DWORD	0 a 4294967295
WORD	0 a 65535
BYTE	0 a 255
BOOL	TRUE o FALSE
DT	Formato: DATE_AND_TIME date_and_time DT dt#YYYY-MM-DD-HH:MM:S.S YYYY (100-2100) = año MM (1-12) = mes DD (1-31) = día del mes HH (0-23) = hora MM (0-59) = minuto S.S (0.0-59.0) = segundos (número real)
DATE	Formato: DATE date D d#YYYY-MM-DD YYYY (100-2100) = año MM (1-12) = mes DD (1-31) = día del mes.

Tipo de Dato	Rango
FILE	<p>Tiene tres sistemas de variables de entrada, los cuáles son identificados por el nombre del bloque de control de la función (fcb) más una extensión.</p> <p>fcb.ACCESS Variable Byte que especifica el estado de lectura / escritura del archivo después que este se ha abierto. 0 = (predefinido) el archivo es abierto para operaciones de lectura/escritura. 1 = archivo es abierto para operaciones de solo lectura.. 2 = archivo es abierto para operaciones de solo lectura.</p> <p>fcb.APPEND Variable Boolean especifica si o no los datos pueden ser añadidos al archivo después que se abre. Solamente se valida cuando el archivo se abre con estado de escritura. Esto es ACCESS variable = 0 o 2. TRUE = el dato será añadido a el archivo. FALSE = (predefinido) el dato no puede ser añadido al archivo.</p> <p>fcb.SHARE Variable Byte especifica cómo otras aplicaciones pueden acceder al archivo después de el se abre. 0 = (predeterminado) otras aplicaciones acceden al archivo para operaciones de lectura y escritura. 1 = otras aplicaciones acceden al archivo para operaciones de solo lectura. 2 = otras aplicaciones acceden al archivo para operaciones de solo lectura. 3 = otras aplicaciones no pueden acceder al archivo.</p> <p>Tiene ocho sistemas de variables de salida, los cuáles están identificados por el nombre del bloque de control de la función masa una extensión.</p> <p>Fcb.OPEN Variable Boolean que indica que el archivo ha sido abierto.</p> <p>Fcb.BUSY Variable Boolean indicando que el archivo esta siendo accesado.</p> <p>Fcb.EFLANG Variable Boolean que indica cuando ocurre un error.</p> <p>Fcb.RDN Variable Boolean que indica que una operación de lectura ha sido completada.</p> <p>Fcb.WDN Variable Boolean que indica que una operación de escritura ha sido completada.</p> <p>Fcb.CLSD Variable Boolean que indica que el archivo ha sido cerrado.</p> <p>Fcb.EOF Variable Boolean que indica que el sistema encontró un End Of File (terminación de archivo).</p> <p>Fcb.ERR Variable Integer que contiene el código de error si un error ocurre.</p>

Tipo de Dato	Rango
STRING	<p>Formato: ' Character_representation '</p> <p>Character_representation = cualquier carácter imprimible hasta 1024 caracteres.</p> <p>Un \$ seguido por dígitos hexadecimal, encerrado en comillas es interpretada como una representación hexadecimal de código de caracteres de ocho bits.</p> <p>Ejemplo: '\$41 \$42 \$43' es interpretado como A B C.</p> <p>Para designar estos caracteres, use el siguiente formato:</p> <p>signo Dollar = '\$\$'</p> <p>Single quote = '\$'</p> <p>Line feed = '\$L' or '\$l'</p> <p>New line = '\$N' or '\$n'</p> <p>Form feed = '\$P' or '\$p'</p> <p>Carriage return = '\$R' or '\$r'</p> <p>Tab = '\$T' or '\$t'</p>
TIME	<p>Formato: TIME tiempo T t #Days Horas Minutos Segundos Milisegundos</p> <p>Dias = xD, donde x = un integer (0-1000000), e.g. 5D.</p> <p>Horas = xH, donde x = un integer (0-23), e.g., 7H.</p> <p>Minutos = xM, donde x = un integer (0-59), e.g., 5M.</p> <p>Segundos = xS, donde x = un integer (0-59), e.g., 5S.</p> <p>Milisegundos = x MS, donde x = un integer (0-999), e.g., 200MS.</p> <p>Ejemplo: T#1D20H13M15S500MS es interpretado como 1 día, 20 horas, 13 minutos, 15 segundos, y 500 milisegundos.</p> <p>Ejemplo: t#1d2h31M6s5ms es interpretado como 1 día, 2 horas, 31 minutos, 6 segundos, y 5 milisegundos.</p>

Tipo de Dato	Rango
TMR	<p>Tiene cuatro variables de sistema, los cuales son identificados por el nombre del timer mas una extensión.</p> <p>Tmr_name.PT contiene el valor del tiempo presente. Esta variable es retentiva. Esto es , que retiene su valor durante una perdida de energía.</p> <p>Tmr_name.EN inicia y detiene el TMR y es un tipo de dato BOOLEAN .</p> <p>Tmr_name.ET contiene el lapso de tiempo del TMR en segundos.</p> <p>Tmr_name.Q contiene la salida del TMR.</p>
TOD	<p>Formato: TIME_OF_DAY time_of_day TOD tod#HH:MM:SS</p> <p>HH (0-23) = hora MM (0-59) = minuto S.S (0.0-59.0) = segundos</p>
User - defined	<p>Consiste en un juego de tipos de datos que funcionan como un grupo. Los miembros no tienen que ser del mismo tipo de datos.</p>

MONITOREO DE DATOS POR DDE / SUITELINK

Este apéndice proporciona información acerca de cómo monitorear variables InControl desde otras aplicaciones.



CONTENIDO

- INTRODUCCIÓN.
- MONITOREO DE VARIABLES DESDE INTOUCH.
- MONITOREO DE VARIABLES DESDE EXCEL.

This product is licensed to:

Company:	N/A
SN:	N/A
Expires:	N/A
Locked To:	N/A

7.1.0.1337

INTRODUCCIÓN

Usted puede monitorear variables desde otras aplicaciones, así como InTouch o Excel, por ejemplo. Puede usar cualquiera de los protocolos DDE o SuiteLink. Este apéndice describe cómo usar DDE e InControl para operar como un servidor DDE.

Si la aplicación está sobre un nodo remoto, necesita configurar acciones DDE para aquel nodo. Información acerca de la configuración de acciones DDE, ver Guía de Administrador de Sistemas FactorySuite.

Para información acerca del uso de SuiteLink, el cual le permite a InControl operar un cliente SuiteLink, ver Guía de Usuario de Wonderware InControl SuiteLink.

Nota: Algunas aplicaciones, como Excel, no podrán monitorear elementos en serie o miembros de estructuras.

MONITOREO DE VARIABLES DESDE INTOUCH

InTouch puede recibir datos desde InControl cuando usted crea elementos en el diccionario de datos etiquetados de InTouch. Para instrucciones detalladas acerca de cómo usar una conexión DDE para monitorear una variable InControl desde InTouch, ver la Guía de Usuario de InTouch.

Cuando usted crea un elemento DDE, necesita la siguiente información:

Para el nombre de aplicación, use **RTEngine**.

Para el tema, use **TagName**.

Para el nombre de los elementos, use el nombre de la variable que quiera monitorear, siga las siguientes reglas:

- Para referenciar una variable global InControl, use solamente el nombre de la variable.
- Para referenciar una variable local InControl, precede el nombre de la variable con el nombre del programa seguido por un punto. Por ejemplo, si el nombre del programa es SeamWeld, y el nombre de la variable es weld_done, use Seamweld.weld_done como el nombre del elemento DDE.
- Para referenciar un tipo de dato user-defined, precede el nombre del miembro con el nombre de la variable del tipo de dato user-defined seguido por un punto. Por ejemplo si el tipo de dato user-defined es Device_Status y el miembro de la variable Device_Status que quiere monitorear es Speed, use Device_Status.Speed como el nombre del elemento DDE.

Cuando usted visualiza etiquetas SuiteLink en la ventana Watch de InControl, puede observar que sus valores están cambiando, incluso cuando las etiquetas han sido forzadas. Esto se debe a la reflexión de golpes por el servidor SuiteLink a otros clientes. Usted puede prevenir esto haciendo un cambio al registro Windows NT.

Advertencia: Hacer cambios incorrectos en el registro Windows NT puede resultar en operaciones impredecibles para los equipos I/O e InControl. Esto tiene un riesgo potencial de lesiones o muerte a personal y/o daños a equipos. Siempre verifique que los cambios que hace en claves de registro no afectarán adversamente la operación del computador o las aplicaciones que intentó correr antes de intentar controlar equipos de campo de fábrica.

Tenga en cuenta que Wonderware no provee soporte técnico para problemas que pueden surgir como un resultado de los cambios que usted hace al registro.

➤ **Para cambiar el registro:**

1. Edite el registro en la siguiente localización.

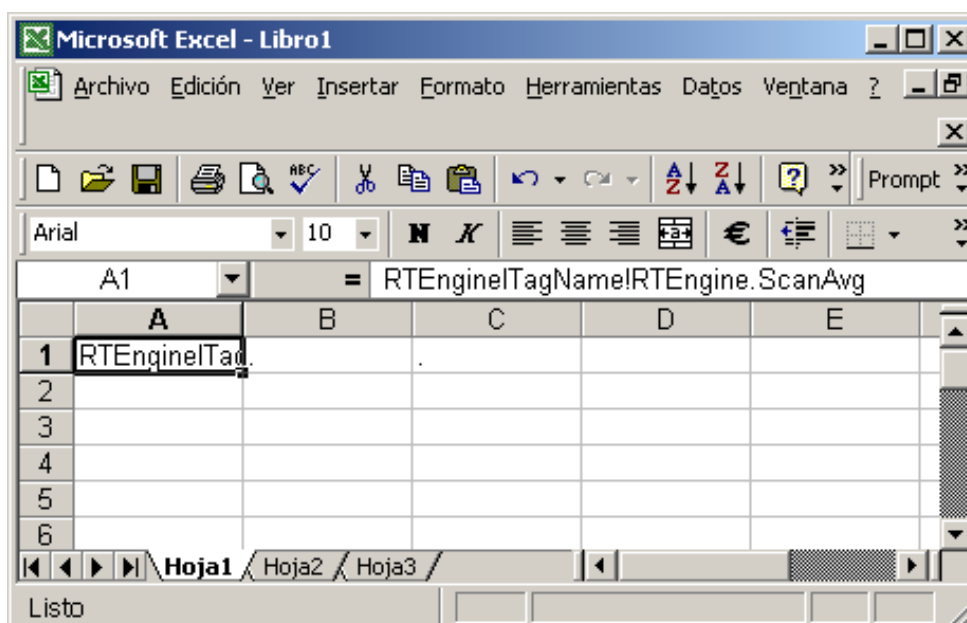
**HKEY_LOCAL_MACHINE\
SOFTWARE\
WONDERWARE\
INCONTROL**

2. Cree una nueva clave llamada **SuiteLinkReflectPokes**. Esta debe ser un tipo de dato DWORD.
3. Asigne a la clave un valor de 0. Esto previene al servidor de rebotes recibidos vía SuiteLink hacia otros clientes.

MONITOREO DE VARIABLES DESDE EXCEL

Excel puede leer datos desde InControl cuando crea una referencia remota DDE en una celda.

Nota: Algunas aplicaciones así como Excel, no podrán monitorear los elementos en serie o miembros de estructuras.



La fórmula en A1 es

=RTEngine|TagName!RTEngine.ScanAvg

Donde:

RTEngine es el nombre de la aplicación.

| Es el símbolo usado para separar el nombre de aplicación del nombre del tema.

TagName Es el nombre del tema (llamado como se muestra en el ejemplo).

! Es el que delimita el nombre del tema y el nombre de la variable, y

RTEngine.ScanAvg es el nombre de la variable que está siendo monitoreada.

Usted puede mostrar cualquier nombre de variable aquí.

Para referenciar una variable InControl local, precede el nombre de la variable con el nombre del programa seguido por un punto. Por ejemplo, si el nombre del programa es PowerUp, y el nombre de la variable es lights_on, use PowerUp.lights_on como el nombre de elemento DDE.

Para referenciar un tipo dato user-defined, precede el nombre del miembro con el nombre del tipo de dato seguido por un punto. Por ejemplo si el tipo de dato user-defined es Time y el miembro de la estructura Time que quiere monitorear es Hour, use Time.Hour como el nombre del elemento DDE.

Esta fórmula es dinámica y actualiza el valor en A1 continuamente. Para mas información acerca de entradas de fórmulas dentro de una hoja de cálculo Excel, ver la documentación de usuario Excel.

Usted debe configurar las acciones DDE para la aplicación "RTEngine.". Para mas información acerca de la configuración de las acciones DDE, ver La Guía de Administrador de Sistema de Wonderware FactorySuite.

EXTENSIONES IEC 61131-3

Este apéndice describe el perfeccionamiento y otras extensiones a la especificación IEC 61131-3. InControl cumple con IEC 61131-3 excepto donde lo notará en este apéndice.



CONTENIDO

- SFC Y PROGRAMAS DE TEXTO ESTRUCTURADO
- PROGRAMAS RLL
- TIPOS DE DATOS
- PARÁMETROS INCONTROL

This product is licensed to:

Company:	N/A
SN:	N/A
Expires:	N/A
Locked To:	N/A

7.1.0.1337

SFC Y PROGRAMAS DE TEXTO ESTRUCTURADO

Para mas información acerca de la programación SFC, ver los capítulos:

“USO DEL EDITOR SFC”

“ELEMENTOS DE PROGRAMAS SFC”

“LENGUAJE DE TEXTO ESTRUCTURADO”

Consideraciones SFC

Transiciones

Típicamente, los Steps (pasos) dentro de un SFC están separados con transiciones. InControl permite que usted coloque una Transición después de otra, sin pasos entre ellas.

Cuando dos Steps no son separados por una Transición, InControl inserta una transición TRUE invisible entre ellos. Cuando dos transiciones no están separadas por un Step, InControl inserta un Step vacío invisible entre ellos.

Todas las funciones dentro de todo los Pasos o Steps que preceden una transición deben haber sido terminados antes que el sistema evalúe una transición.

Representación de Step o Pasos

Los Steps pueden ser representados por una caja que contiene el nombre del Step, el código programado dentro del Step, o por un icono.

Bobina de transición (Transition Coil)

Usted puede usar un Bobina de transición dentro de la lógica de una acción (Action) para detener todos los Pasos asociados y transferir el mando a la etiqueta (Label) asociada.

Macro Step (Macro Paso)

Usted puede usar el Macro Paso para llamar a un SFC, para la ejecución de un Paso desde otro SFC. El flujo del programa se transfiere al SFC que fue llamado (SFC-hijo). Cuando el SFC hijo ha completado la ejecución, el flujo del programa retorna al SFC padre y reasume o continua después del Macro Paso.

Consideraciones STL

Parámetros

Los parámetros para funciones de Texto Estructurado pueden ser listados en cualquier orden con tal que los nombres del parámetro formal se den como lo especificado por IEC-61131-3.

Orden FOR

Usted puede usar la declaración END_FOR_NOWAIT para regresar (loop back) sin un análisis I/O.

Orden REPEAT

Usted puede usar la declaración END_REPEAT_NOWAIT para regresar (loop back) sin un análisis I/O.

Orden SCAN

Usted puede usar la declaración SCAN para suspender la ejecución de la declaración de Texto Estructurado hasta después del próximo I/O Scan.

Orden BREAK

Usted puede usar la declaración BREAK para detener el flujo del programa. BREAK es útil para depurar un programa.

Funciones no Soportadas

InControl no soporta las siguientes funciones, las cuales están definidas en la especificación IEC 61131-3.

- LIMIT
- MUX
- SEL

PROGRAMAS RLL

Para mas información acerca de la programación RLL, ver este capítulo:

“USANDO EL EDITOR RLL”

“ELEMENTOS DE PROGRAMAS RLL”

Bobinas de Salida(Output Coils)

Usted puede colocar un Output Coils en cualquier parte sobre un escalón, incluyendo a la izquierda de una espirales (coil) de entrada o dentro de una rama OR. Usted puede colocar múltiples espirales sobre un único escalón, y el estado de un espiral no es afectado por los estados de los otros. Un Output Coil (espiral de salida) almacena los resultados lógicos de la lógica evaluada hasta su localización sobre el escalón.

Ramal OR (OR Branches)

Usted puede insertar una rama OR que no contenga una lógica (desviación). Usted puede usar una desviación temporalmente para desactivar una sección de lógica sin borrarla del programa. Usado con un contacto que apaga el flujo de energía para la lógica en cuestión, la desviación mantiene el flujo de energía por el resto del escalón.

Parámetros del Contador (Counter Parameters)

Los parámetros de Valor Presente y Valor Actual usados en el bloque de funciones del contador RLL son tipos dato DINT.

Bloques de Funciones Matemáticas

Los siguientes bloques de funciones matemáticas tienen entradas y salidas que aceptan cualquiera de los tipos de datos Any_Bit, excepto los tipo BOOL:

- ADD
- DIV
- SUB
- MUL
- MOD

TIPOS DE DATOS

Para mas información acerca de los tipos de datos, ver “Tipos de Datos de Variables” en el capítulo “Definiendo Variables”.

Tipos de Datos no Soportados

InControl no soporta los siguientes tipos de datos, los cuales están definidos en la especificación IEC 61131-3.

- LINT
- ULINT
- LWORD

Conversión Tipos de Datos

InControl hace una conversión automática limitada de tipos de datos. No es necesario usar las funciones INT_TO_REL y REAL_TO_INT especificadas en el IEC, porque la conversión es manejada internamente por los bloques de funciones RLL MOVE y declaraciones Structured Text Assignment.

Para mas información acerca de cómo InControl maneja conversión de tipos de datos, ver “Conversión tipos de datos” en el capítulo “Definiendo variables”.

ESPECIFICACIÓN PARÁMETROS PARA INCONTROL

Los parámetros InControl están listados en la siguiente tabla.

Implementación - Parámetros Dependientes

Cláusula	Parámetro	Implementación
1.5.1	Procedimientos de manejo de error	Ver tabla Condiciones de error (Pagina D-9)
2.1.1	Caracteres nacionales usados	Ninguno
2.1.2	Longitud máxima de identificadores	100
2.1.5	Longitud de comentario máximo.	No limitada.
2.2.3.1	Rango de valores de duración	Rango de REAL en ms.
2.3.1	Rango de valores para variables de tipo dato TIME.	Rango de REAL en ms.
2.3.1	Precisión de representación de segundos en tipos datos TIME_OF_DAY/DATE_AND_TIME.	Rango de REAL en ms.
2.3.3	Número máximo de subíndice de series.	1
2.3.3	Máximo tamaño de serie.	Limitado por el espacio usado por todos los símbolos.
2.3.3	Número máximo de elementos de la estructura.	Limitado por el espacio usado por todos los símbolos.
2.3.3	Máximo tamaño de la estructura.	Limitado por el espacio usado por todos los símbolos.
2.3.3	Número máximo de variables por declaración.	Limitado por el espacio usado por todos los símbolos.
2.3.3.1	Número máximo de valores enumerados.	ninguna
2.3.3.2	Longitud máxima prefijada de variables STRING.	1024
2.3.3.2	Longitud máxima permitida de variables STRING.	1024
2.4.1.1	Número máximo de niveles jerárquicos	Ninguno.

Cláusula	Parámetro	Implementación
2.4.1.1	Cartografía lógica o física.	Ninguna.
2.4.1.2	Número máximo de subíndices.	1
2.4.1.2	Máximo rango de valores de subíndices.	No limitado.
2.4.1.2	Máximo número de niveles de estructuras.	No limitado.
2.4.2	Iniciación de entradas de sistemas.	Lo mismo como todas las variables.
2.4.3	Número máximo de variables por declaración.	Ninguna.
2.5.1.1	Método de representación de funciones.	Nombres y bloques de función.
2.5.1.3	Número máximo de especificaciones de función.	Limitado por el espacio usado por todos los símbolos.
2.5.1.5	Número máximo de entradas de funciones extensibles.	2
2.5.1.5.1	Efectos de tipo conversión exactitud.	por Ninguna.
2.5.1.5.2	Exactitud de funciones de variable.	una Lo mismo que la exactitud de la función de la librería C.
2.5.2	Número máximo de especificaciones de bloque de función.	Ninguna.
2.5.3.3	PVmin, PVmax de contadores.	De tipo DINT
2.5.3	Limitaciones del tamaño del programa.	No limitado.
2.6.2	Precisión del tiempo transcurrido en Steps.	Ms
2.6.2	Número máximo de Steps por SFC.	No limitado
2.6.3	Número máximo de transiciones por SFC y por Step (paso).	No limitado.
2.6.4	Mecanismos de mando de acción.	Declaraciones gráficas en lenguaje RLL.
2.6.4.2	Número máximo de bloques de Actino por Step.	No limitado.
2.6.5	Indicaciones gráficas del estado sep.	El Step es resaltado.
2.6.5	Tiempo de liquidación de transición.	Un scan.
2.6.5	Anchura máxima de construcciones divergir / convergir.	No limitada.

Cláusula	Parámetro	Implementación
2.7.1	Contenidos de bibliotecas del recurso.	Ninguna.
2.7.2	Número máximo de tareas.	No limitado.
2.7.2	Resolución del intervalo de la tarea.	Ms.
2.7.2	Planificación preventiva o non-preventiva.	No preventiva
3.3.1	Longitud máxima de expresiones.	No limitada.
3.3.1	Evaluación parcial de expresiones Boolean.	Evaluación completa.
3.3.2	Longitud máxima de declaraciones.	No limitada.
3.3.2.3	Número máximo de selecciones CASE	No limitada
3.3.2.4	Valor de la variable de control en la terminación del lazo FOR.	Uno mas ayá del último valor.
4.1.1	Representaciones gráficas y semigráficas.	Gráficas.
4.1.1	Restricciones sobre la topología de la red.	Emparejar convergencias y divergencias.
4.1.3	Orden de evaluación de lazos de realimentación.	Ninguna.

Condiciones de Error

Las condiciones de error definidas en IEC 61131-3 están listadas en la siguiente tabla. El método de detección (preparación de programas o ejecución de programas) esta listado.

Clausula	Condición de Error	Detección
2.3.3.1	Valor de la variable excede el subrango especificado.	Preparación y ejecución.
2.4.2	La longitud de lista de la inicialización no empareja el número de entradas de la serie.	N/a
2.5.1.5.1	Errores de conversión.	N/a
2.5.1.5.2	El resultado numérico excede el rango para el tipo del datos.	Ejecución.
2.5.1.5.2	División por cero.	Ejecución.
2.5.1.5.4	Los tipos de datos de entrada mezclados a una función de la selección.	N/a
2.5.1.5.4	Selector (K) fuera del rango de la función MUX.	N/a
2.5.1.5.5	Posición de caracter inválida.	Ejecución
2.5.1.5.5	Resultado excede la longitud de string máxima.	Ejecución.
2.5.1.5.6	Resultado excede el rango para el tipo de dato.	Ejecución.
2.6.2	Zero o mas de un Steps inicial en la red SFC.	Preparación.
2.6.2	El programa del usuario intenta modificar estado o tiempo del Paso.	Preparación.
2.6.2.5	Simultáneamente corrige, transiciones non-priorizadas en una divergencia de selección.	N/a
2.6.3	Efectos laterales en la evaluación de condición de Transición.	N/a
2.6.4.5	Error de contención de la acción de control.	Ejecución.
2.6.5	SFC inseguro o inalcanzable.	Preparación.

Clausula	Condición de Error	Detección
2.7.1	Conflicto de tipo de datos en VAR_ACCESS.	Preparación.
2.7.2	La tarea requiere demasiados recursos de procesador.	Ejecución.
2.7.2	La fecha límite de la ejecución no se encontró.	Ejecución.
2.7.2	Otros conflictos de planificación de tarea.	Ejecución.
3.2.2	Resultado numérico excede el rango para el tipo de dato.	Ejecución.
3.3.1	División por cero.	Ejecución.
3.3.1	Tipo de dato inválido para ejecución.	Ejecución.
3.3.2.1	Retorno de la función sin valor asignado.	N/a
3.3.2.4	La iteración no termina.	Ninguna.
4.1.1	El mismo identificador usado como etiqueta del conector y nombre del elemento.	Preparación.
4.1.4	Variable de realimentación no inicializada.	N/a



TECLAS DE ACCESO DIRECTO

El apéndice lista las combinaciones de teclas para herramientas InControl y opciones de menú. Estas combinaciones están basadas sobre el teclado americano (U.S). Si esta usando un teclado de diferente lenguaje, pudiese necesitar cambiar sus combinaciones de teclas.

CONTENIDO

- OPERACIONES GENERALES
- VENTANA DE PROYECTOS (PROJECT WINDOW)
- VENTANA DE SALIDA (OUTPUT WINDOW)
- ADMINISTRADOR DE PROYECTOS (PROJECT MANAGER)
- VENTANA DE OBSERVACIÓN (WATCH WINDOW)
- EDITOR DE PROGRAMAS (PROGRAM EDITORS)
- ADMINISTRADOR DE SÍMBOLOS (SYMBOL MANAGER)

This product is not for sale in the following countries:

Company:	N/A
SN:	N/A
Expires:	N/A
Locked To:	N/A

7.1.0.1337

OPERACIONES GENERALES

Accesos Directos a la Barras de Herramientas

Operación

Tecla o Combinación de Tecla

Abre un nuevo programa.	Ctrl N
Abre un programa existente.	Ctrl O
Imprime un programa seleccionado.	Ctrl P
Guarda el programa seleccionado.	Ctrl S
Copia el elemento en el portapapeles.	Ctrl C
Pega el contenido del portapapeles.	Ctrl V
Corta el elemento seleccionado y lo ubica en la papelera.	Ctrl X
Deshace la última operación.	Ctrl Z
Rehace o repite la última operación.	Ctrl Y
Busca el elemento seleccionado.	Ctrl F
Busca la próxima ocurrencia del elemento seleccionado.	F3
Reemplaza el elemento seleccionado.	Ctrl H
Elimina el elemento seleccionado.	DEL
Abre el Symbol Manager.	Ctrl T
Muestra la ayuda en línea.	F1
Pone al runtime Engine en modo Stop.	Ctrl Break
Ejecuta un paso del programa.	F10
Valida el proyecto.	F4
Valida el programa.	Shift F4
Ejecuta el proyecto.	F5
Ejecuta el programa.	Shift F5
Descarga el proyecto.	Ctrl F5
Descarga el programa	Shift Ctrl F5
Detiene el proyecto.	F7
Detiene el programa.	Shift F7
Análisis único al proyecto.	F8
Análisis único al programa.	Shift F8
Barra de punto de ruptura.	F9
Limpia todas las barras de punto de ruptura.	Shift Ctrl F9

Operaciones de Ventanas

Operación	Combinación de teclas
Interruptor de enfoque entre el la ventana de Project, Watch, Ventana, Ventana de Salida (Output), y la ventana Development / Runtime.	ALT F6
Muestra la ventana Project.	Ctrl J
Muestra la ventana Watch.	Ctrl W
Interruptor de enfoque entre los programas abiertos en la ventana de edición.	Ctrl F6

VENTANA DE PROYECTOS

acceso directo Ventanas y Archivos

Operación	Combinación de Teclas
Elimina el programa seleccionado.	DEL
Abre el programa seleccionado para edición.	Enter
Muestra las propiedades para el programa seleccionado.	ALT Enter

VENTANA OUTPUT(SALIDA)

acceso directo a las ventanas

Operación	Combinación de teclas
Copia el texto seleccionado al portapapeles.	Ctrl C
Limpia la ventana de toda la iformación, si o no algun texto este seleccionado.	DEL

PROJECT MANAGER

acceso directo a Archivos

Operaciones	Combinación de teclas
Crea un nuevo proyecto	Ctrl N
Abre el proyecto seleccionado	Enter <i>or</i> Ctrl O
Elimina el proyecto seleccionado	DEL
Muestra las propiedades para el proyecto seleccionado	ALT Enter

VENTANA WATCH

acceso directo a la Barras de herramientas

Operación	Combinación de Teclas
Copia el elemento seleccionado al portapapeles.	Ctrl C
Pega el contenido del protapapeles.	Ctrl V
Corta el elemento seleccionado y lo ubica en el portapapeles.	Ctrl X
Remueve el símbolo seleccionado.	DEL
Modifica el valor seleccionado.	Ctrl M
Suelta el símbolo seleccionado.	Ctrl U
Suelta todos los símbolos	Ctrl A
Pone la base del número para el símbolo seleccionado a binario.	Ctrl B 1
Pone la base del número para el símbolo seleccionado a octal.	Ctrl O 1
Pone la base del número para el símbolo seleccionado a decimal.	Ctrl D 1
Pone la base del número para el símbolo seleccionado a hexadecimal.	Ctrl H 1
Inserta una fila	INS
Cambia el enfoque entre la lista de símbolos y la lista de la tabla de la ventana Watch.	TAB

Operación	Combinación de Teclas
Muestra las tablas en la lista de la tabla de la ventana Watch.	ALT Down Arrow
Extiende un árbol colapsado de símbolos.	Right Arrow
Mueve el cursor a la derecha de un único símbolo o un árbol expandido de símbolos.	Right Arrow
Colapsa un árbol expandido de símbolos.	Left Arrow
Hace doble clic en cualquier celda.	F2
<p><i>1 Si un arreglo o estructura es expandida y el elemento madre tiene enfoque, el cambio en la base de el número es aplicado a todos los elementos del arreglo o estructura.</i></p>	

EDITOR DE PROGRAMAS

Acceso directo a los programas RLL

Operación	Combinación de Teclas
Inserta una rama.	Shift B
Inserta un nuevo escalón.	Shift R
Inserta un contacto.	Shift C
Inserta un espiral.	Shift O
Edita un elemento de escalón seleccionado.	F2
Va a un escalón especificado.	Ctrl G

Acceso directo al Texto Estructurado

Operación	Combinación de Teclas
Va a la línea especificada o marcador de libros.	Ctrl G
Selecciona todo el texto.	Ctrl A
Marca la línea actual.	Ctrl L
Abre el Symbol Manager.	Ctrl T
Abre el Symbol manager.	Shift Ctrl T
Selecciona un símbolo y pulsa OK para insertar el símbolo seleccionado dentro del programa.	

Acceso directo al SFC**Operación****Combinación
de teclas**

Inserta un acción.

Shift A

Inserta un comentario.

Shift C

Inserta una divergencia selecta.

Shift D

Inserta un salto.

Shift J

Inserta una etiqueta.

Shift L

Inserta un Paso Macro.

Shift M

Inserta un lazo.

Shift O

Inserta una divergencia simultánea.

Shift P

Inserta un Paso.

Shift S

Edita propiedades de Paso.

ALT Enter

Inserta una Transición.

Shift T

Inserta un Paso de Librería.

Shift Y

Edita un elemento de programa seleccionado.

F2

Abre el administrador de acciones (Actino Manager).

Ctrl M

Cierre una acción.

ESC

Abre el administrador de transición (Transition
Manager).

Ctrl R

Acceso directo a los FOE**Operación****Combinación
de Teclas**Cierra la configuración sin guardar los cambios
ingresados en la etiqueta actual.

ESC

Edita propiedades FOE.

ALT Enter

SYMBOL MANAGER

Acceso directo a Objetos y Ventanas

Operación	Combinación de Teclas
Abre el cuadro de diálogo Export Select Symbols (Exportación de Símbolos Selectos).	ALT E
Abre el cuadro de diálogo Open para importar símbolos.	ALT I
Abre el cuadro de diálogo Imprimir Símbolos Selectos (Print Select Symbols).	ALT P
Selecciona el nivel próximo mas alto de alcance.	Backspace
Añade un símbolo.	Insert
Elimina un símbolo.	DEL
Muestra las propiedades para el símbolo seleccionado.	ALT Enter

GLOSARIO DE TÉRMINOS

Action (Acción):	Colección de operaciones, con nombre asignado, asociada con uno o más Pasos en un SFC.
Action Manager (Administrador de Acciones):	Se usa para renombrar y eliminar Acciones SFC.
Action Qualifier (Calificador de Acciones):	Elemento de programación gráfico dentro de un SFC relacionado con cada bloque de acción que controla la ejecución de la acción lógica relativa al período durante el cual el Paso asociado es activo.
Active File (Archivo Activo):	Archivo del programa que está contenido en la cabeza de la ventana Editor de programas con su barra de título resaltada. Los comandos que son ejecutados desde los menús o pulsando sobre el botón de la barra de herramientas se llevan a cabo en el archivo activo.
ActiveX:	Ver <i>FOE</i> .
Alarma Análoga:	FOE de InControl que monitorea señales de entrada análoga para condiciones de alarma.
ANY:	Tipo de dato genérico que puede representar cualquiera de los tipos de datos soportados por InControl.
ANY_BIT:	Tipo de dato genérico que puede representar estos tipos de datos: DWORD, WORD, BYTE, BOOL, incluyendo un bit individual dentro de estos tipos de datos.
ANY_DATE:	Tipo de dato genérico que puede representar estos tipos de datos: DT, DATE, TOD.

ANY_INT:	Tipo de dato que puede representar el tipo dato INT.
ANY_NUM:	Tipo dato genérico que puede representar estos tipos de datos: ANY_REAL y ANY_INT.
ANY_REAL:	Nombre tipo genérico que puede representar el tipo de dato REAL.
Array (Arreglo):	Conjunto de valores de datos, todos del mismo tipo. Elementos individuales pueden ser referenciados por una expresión que consiste de nombre de fila y una expresión de posicionamiento.
BOOL	Miembro del grupo tipos de datos ANY_BIT. Tipos de datos BOOL son válidos en cualquier instrucción InControl o bloque de función que acepta un tipo de dato ANY, ANY_BIT, o BOOL. Un BOOL es de un bit de longitud y puede tener uno de dos valores: TRUE (1, o ON) o FALSE (0, o OFF).
Boolean Transition	Tipo de transición en un SFC representado por una expresión que consiste de variables y operadores que evalúan un resultado booleano único.
Boolean	Elemento lógico o expresión que evalúa cualquier TRUE o FALSE.
BYTE	Miembro del grupo tipo dato ANY_BIT. Tipos de dato BYTE son válidos en cualquier instrucción InControl o bloque de función que acepta un tipo dato ANY, ANY_BIT, o BYTE. Un BYTE es un tipo de dato numero entero sin registrar que esta compuesto de uno o más de los dígitos (0-9) y no puede contener un punto decimal. Un BYTE es de 8 bits de longitud y tiene un rango de 0 a 255.

Card (Tarjeta)	<i>Ver Tarjeta E/S.</i>
Child SFC	El macro es un POU especializado que provee un medio de incluir un SFC, el SFC hijo, para la ejecución de un Paso en otro SFC, el SFC origen o padre.
Coil	Elemento de programación gráfico RLL que representa una variable de salida Boolean.
Conector	Grupo de puertos E/S usualmente enrutados a un conector físico sobre una tarjeta.
Elemento de lazo de control	<i>Ver Lazo.</i>
DATE	Miembro del grupo tipo dato ANY_DATE. Tipos de dato DATE son válidos en cualquier instrucción o bloque de función que acepta un tipo dato ANY, ANY_DATE, o DATE.
DDE	<i>Ver Intercambio de Datos Dinámicos.</i>
DINT	Miembro del grupo tipo dato ANY_NUM. Tipo de dato DINT son validados en cualquier instrucción o bloque de función que acepte un tipo de dato ANY, ANY_NUM, ANY_INT, o DINT. El DINT es un tipo de dato integer registrado que esta compuesto de uno o mas de los dígitos (0-9) y no puede contener un punto decimal. El DINT tiene 32 bits de longitud y tiene un rango de -2147483648 hasta +2147483647.

DT

Miembro del grupo tipo dato ANY_DATE. Tipos de dato DT son validados dentro de cualquier instrucción o bloque de instrucción que acepte un tipo de dato ANY. ANY_DATE, o DT.

Formato: DT#YYYY-MM-DD-HH:MM:S.S

YYYY(100-2100)= año

MM(1-12)= mes

DD(1-31) = día del mes

HH(0-23) = hora

MM (0-59) = minuto

S.S(0.0-59.0) = segundos (número real)

File

El tipo de dato FILE es un miembro del grupo tipo de datos ANY. FILE es una estructura que es diseñada solamente para las variables de control de archivos usado con RLL y funciones de archivo de Texto estructurado.

DWORD

Miembro del grupo tipo dato ANY_BIT. Tipos de dato DWORD son válidos en cualquier instrucción InControl o bloque de función que acepta un tipo dato ANY, ANY_BIT, o DWORD. Un DWORD es un tipo de dato integer sin registro que esta compuesto de uno o mas de los dígitos (0-9) y no puede contener un punto decimal. Un DWORD tiene una longitud de 32 bits y tiene un rango de 0 hasta 4294967295.

Intercambio de Datos Dinámicos DDE es el paso de datos entre aplicaciones, llevado a cabo sin involucramiento o monitoreo de usuario. En el ambiente Windows, DDE se consigue a través de un conjunto de tipos de mensajes, procedimientos recomendados (protocolos) para procesar estos tipos de mensajes, y algunos tipos de datos definidos recientemente. Siguiendo los protocolos, las aplicaciones que fueron escritas independientemente de los otros pueden pasar datos entre ellos mismos sin involucramiento por parte del usuario, ejemplo: InTouch.

Objetos de Fábrica *Ver FOE.*

Factory Suite FactorySuite 2000 es el primero en el mundo que ha integrado componentes basados en sistemas MMI. Con FactorySuite 2000, tiene acceso a toda la información que necesita para correr su diseño de fábrica (visualización, optimización y control, colección de datos de tipos datos) y hacer su planta verdaderamente productiva.

FOE InControl es compatible con las especificaciones del servidor ActiveX. El editor de Objetos de fábrica InControl (FOE) es un contenedor ActiveX, el cual lo habilita para usar controles ActiveX dentro de un proyecto InControl. Un control ActiveX debe estar instalado dentro del InControl antes que usted pueda configurarlo y correrlo. Después de la instalación, es llamado como un objeto de fábrica InControl (FOE). Como otros programas InControl, un FOE se puede ejecutar independientemente. También puede llamarlo para ejecución desde otro programa.

- Force** Utilidad de la ventana Watch que pone una variable a un valor que no cambia mientras el programa que corre hasta que usted lo suelte.
- Function** Un POU consiste de un conjunto de instrucciones de programación que pueden ser llamados para ejecución por otros POU. Las funciones son algoritmos predefinidos que cargan una operación particular, así como Raíz Cuadrada, Rotar a la Izquierda, Tangente, etc. Una función real retorna un valor y es usado al lado derecho de una declaración de asignación. *Ver procedimiento.*
- Bloque de Función** Un POU consiste de un conjunto de instrucciones de programación que pueden ser llamados para ejecución por otros POU. A diferencia de una función, una o más instancias de un tipo bloque de función puede ser creado; y las variables locales mantener sus valores entre llamadas.
- INT** Miembro de un grupo de dato ANY_NUM. Tipo de dato INT son válidos en cualquier instrucción o bloque de función que acepta un tipo dato ANY, ANY_NUM, ANY_INT, o INT. El INT es un tipo dato integer registrado que esta compuesto de uno o mas de los dígitos (0-9) y no pueden contener un punto decimal. El INT tiene 16 bits de longitud y tiene un rango de -32768 hasta +32767.
- Tarjeta I/O** Se introduce dentro de una de las hendiduras de expansión de la unidad de sistemas InControl y se conecta a las rejillas E/S periféricas y módulos. También llamado tarjeta (card o board).

Jump (Salto)	Elemento de programación gráfico dentro de un programa SFC o RLL que dirige el flujo del programa a otra localidad en el programa identificado por una etiqueta.
Label (Etiqueta)	Elemento de programación gráfica en un programa SFC o RLL que identifica dónde el flujo del programa debe continuar desde su correspondiente salto.
Loop (Lazo)	Elemento de programación gráfico en un diagrama SFC. El elemento Lazo contiene dos condiciones de transición: uno dirige el flujo del programa en la dirección descendente, y el otro dirige el flujo del programa hacia atrás. Elementos de lazo múltiples pueden ser anidados dentro de estos mismos pero no pueden cruzarse unos con otros y no pueden ingresar divergencias de selección o paralelas. <i>Ver también lazo PID.</i>
Padre SFC	El macro es un POU especializado que provee una manera de incluir un SFC, el SFC hijo, para la ejecución desde un Paso en otro SFC, el SFC origen.
LREAL	Miembro del grupo tipo de dato ANY_NUM. Tipos de datos REAL son válidos en cualquier instrucción o bloque de unión que acepta un tipo dato ANY, ANY_NUM, ANY_REAL, o LREAL. Un número tipo dato LREAL es un valor de 64 bits compuesto de uno o mas de los dígitos (0-9), es registrado, y contiene un punto decimal. El rango para los números LREAL es el siguiente: -1.79769313486 231 E308 (negativo) hasta + 1.79769313486231 E308 (positivo), incluyendo el cero.

Macro	Un POU especializado que proporciona un medio de incluir un SFC, el SFC hijo, para la ejecución desde un Paso en otro SFC, el SFC origen o padre. El Macro Paso es el elemento gráfico en un SFC que representa la inclusión de otro SFC entero como un Paso particular. El SFC incluido empieza la ejecución en su Paso de inicio (Start Step) cuando el Paso Macro que lo llama esta activo. La ejecución en el Paso Macro es completado cuando el SFC incluido alcanza su paso final (End Step).
Divergencia Paralela	Elemento de programación gráfica SFC que separa una ruta de control dentro de dos o mas rutas paralelas. Cuando la ejecución del programa alcanza el inicio de una divergencia paralela, todas las rutas de control subsiguientes llegan a activarse en paralelo. Estas rutas de control continúan siendo activas hasta que todas las rutas de control dentro de una Divergencia Paralela alcanza el punto de convergencia. En estos puntos, todas las rutas dentro de la divergencia paralela son desactivadas y la ruta de control debajo del punto de convergencia será activa.
Lazo PID	FOE InControl que maneja control de retroalimentación derivativo, integral y proporcional.
Port	Puerto E/S, usualmente consiste de 8 bits de E/S.
POU	El POU (unidad de organización de programas) esta definido por el estandar IEC 61131-3 como la unidad de programación básica. InControl soporta estos POU: programas, función, bloque de función, y macro.

Procedimiento	Algoritmo predefinido que carga hacia fuera una operación particular, así como Eliminar archivo (Delete File), Abortar todo (Abort All), rebobinar archivo (Rewind File), etc. Un procedimiento no retorna un valor. <i>Ver también función.</i>
Programa	Un POU consiste de un bloque de código que puede ser programado para ejecutar automáticamente cada análisis (scan).
Proyecto (project)	Organiza y agrupa los programas de aplicación y configura archivos para una aplicación en un subdirectorío aparte.
REAL	Miembro del grupo tipo dato ANY_NUM. Tipos de dato REAL son válidos en cualquier instrucción o bloque de función que acepta un tipo dato ANY o REAL. Un número tipo dato REAL es un valor de 32 bits compuesto de uno o mas los dígitos (0-9), es registrado, y contiene un punto decimal. El rango para los números REAL es el siguiente: -3.0402823 E38 (negativo), hasta -3.402823 E38 (positivo), incluyendo el cero.
Lógica en Escalera por Relevo	RLL es un lenguaje de programación gráfica usado para describir la lógica de programas de aplicación basado en contactos de relevos eléctrico y bobinas análoga.
Variable Retentiva	Una variable que retiene sus valores dentro del evento de una perdida de energía.
Transición RLL	Transición en un SFC que es programada usando el lenguaje RLL. La transición RLL consiste de un escalón RLL particular con una bobina de salida que tiene el mismo nombre tal como la transición RLL.

Runtime Engine	Modulo responsable para la programación y ejecución de la lógica del programa asociado con el código fuente del proyecto, e.g., SFC, RLL, etc. Este módulo también se desempeña como un Suite Link o servidor DDE para Suite Link o clientes DDE tales como InTouch.
Divergencia de Selección	Elemento de programación gráfica SFC que separa una ruta de control particular dentro de dos o mas rutas. La ruta de control seleccionada para la ejecución esta determinada por las condiciones de transición que están localizadas en el principio de cada una de las nuevas rutas de control.
Gráfica de Funciones Secuenciales	SFC es un lenguaje de programación gráfica para esquematizar lógica secuencial usando Pasos, Transiciones y Acciones.
Bobina de Transición SFC	Un elemento de programación gráfica SFC que puede ser asociado con un Paso o un macro Paso y proporciona control del flujo del programa. Asociado con un paso, la bobina de transición SFC puede abortar el Paso y dirigir el flujo del programa a otro Paso. Asociado con un Macro Paso, la bobina de transición SFC puede abortar el SFC hijo y dirigir el flujo del programa a otro Paso en el SFC origen o padre.
SINT	Miembro del grupo tipo dato ANY_NUM. Tipos de datos INT son válidos en cualquier instrucción o bloque de función que acepta un tipo dato ANY o SINT. El SINT es un tipo dato integer señalado que esta compuesto de uno o mas dígitos (0-9) y no contiene un punto decimal. El INT es de 8 bits de longitud y un rango de -128 hasta +127.

Paso	Elemento gráfico en un SFC que representa un estado o lapso de tiempo en la ejecución del programa durante el cual las acciones y funciones asociadas con el Paso son realizadas.
STL	Ver Lenguaje de Texto Estructurado.
STRING	Miembro del grupo tipo dato ANY. Tipos de datos STRING son válidos en cualquier instrucción o bloque de función que acepta un tipo dato STRING o ANY. El formato para un tipo dato STRING consiste de un string de hasta 1024 caracteres ASCII dentro de comillas simples.
STL (Lenguaje de Texto Estructurado)	El lenguaje de programación de Texto Estructurado es un subconjunto de un conjunto de instrucciones basado en texto compatible con IEC – 61131.
Suite Link	El driver Suite Link provee fiabilidad, alta velocidad de recopilación de datos para controladores, tarjetas PC y otros equipos. A diferencia de los DDE, el cual transfiere solamente un mensaje a la vez, Suite Link transfiere bloques de mensajes, permitiendo mas rapidez y transferencia de datos eficiente.
Tagname (Etiqueta)	El nombre asignado a una variable en la base de datos InTouch para representar una variable InControl.
Task View	Muestra la gráfica de los programas en un proyecto, ordenados por prioridad de ejecución.
TMR	El tipo de dato TMR es un miembro del grupo tipo de dato ANY. Tipos de datos TMR son válidos en cualquier instrucción o bloque de función que acepta un tipo de dato ANY o TMR.

TIME

Miembro del grupo tipo dato ANY. Tipos de dato TIME son válidos en cualquier instrucción o bloque de función que acepta un tipo dato ANY o TIME. El formato del tipo dato TIME consiste de un T# o t# seguido por una secuencia de uno o mas números y unidades de tiempo específicas.

Ejemplo:

T# 1D2h = 1 día y 2 horas

t#26H = 26 horas

t#5m45s = 5 minutos y 45 segundos

t#26S2000MS = 26 segundos y 200 milisegundos

T#900ms = 900 milisegundos.

TOD

El tipo de dato TOD es un miembro del grupo tipo dato ANY_DATE. Tipos de datos TOD son válidos en cualquier instrucción o bloque de función que acepta un ANY, ANY_DATE, tipo dato TOD.

Transición

Elemento gráfico en un SFC que evalúa un resultado Boolean. Este resultado Boolean determina cuando el flujo del programa es pasado o entregado desde el Paso(s) precedente a la transición hasta el Paso(s) siguiente a la transición.

UDINT

El tipo dato UDINT es equivalente al tipo dato DWORD. Una mejora de InControl a los tipos de datos ANY_BIT hacen que al tipo dato UDINT (integer doble no señalado) innecesario.

UINT	El tipo de dato UINT es equivalente al tipo de dato WORD. Una mejora InControl a los tipos de datos ANY_BIT hacen al tipo de dato UINT(integer no señalado) innecesario.
Unforce	Utilidad de la ventana Watch que remueve los estados forzados (Force) desde una variable, permitiendo al programa escribir valores en el. <i>Ver también Force.</i>
USER_DEFINED	Miembro del grupo tipo dato ANY. Tipos de datos USER-DEFINED consisten de un grupo de tipos de datos (integers, Booleans, strings, etc.) que funcionan como un grupo. No tienen que ser del mismo tipo, pero deben ser globales.
USINT	El tipo de dato USINT es equivalente al tipo dato BYTE. Una mejora InControl al tipo de dato ANY_BIT hace al tipo dato USINT innecesario.
Variable	Localización de la memoria interna que contiene datos del proyecto. El contenido de la información esta definido por el tipo y pueden ser números reales, integers, strings, etc. El Administrador de Símbolos (Symbol Manager) es usado para definir una variable y asignarle un nombre simbólico y tipo de dato. También llamado <i>Symbol</i> .
WORD	Miembro del grupo tipo dato ANY_BIT. Tipos de dato WORD son válidos en cualquier instrucción o bloque de función que acepta un tipo dato ANY o WORD. Un WORD es un tipo dato integer no señalado que esta compuesto de uno o mas de los dígitos (0-9) y no pueden contener un punto decimal. Un WORD es de 16 bits de longitud y tiene un rango de 0 hasta 65535.

CURSO DE ENTRENAMIENTO EN INCONTROL




CORPORACIÓN UNIVERSITARIA TECNOLÓGICA DE BOLÍVAR

2003

INTRODUCCIÓN

InControl es la más nueva herramienta de la FactorySuite de Wonderware en la solución de procesos de control para aplicaciones en la industria. InControl es un paquete de control de arquitectura abierta que le permite diseñar, crear, medir y correr aplicaciones de control para controlar sus procesos. Usted puede crear sus propias soluciones de automatización en una variedad de lenguajes basados en texto y gráficos.



JUSTIFICACIÓN DEL CURSO

DIRIGIDO A:

- ◆ Operadores e ingenieros de planta.
- ◆ Administradores.
- ◆ Integradores y diseñadores de sistema.
- ◆ Y otras personas que quieran implementar o utilizar InControl en el diseño de sus procesos o como parte de la solución de un sistema de control.

OBJETIVOS

- ◆ Entender las funciones InControl para crear soluciones de control y aplicaciones para una variedad de áreas de automatización.
- ◆ Comprender los conceptos básicos de la estructura de las soluciones de automatización para su fácil manejo y corrección de errores.
- ◆ Saber trabajar en el ambiente de desarrollo y ejecución InControl.
- ◆ Utilizar completamente las herramientas de manejo de proyectos suministradas por InControl para manejar sus soluciones de automatización.

OBJETIVOS

- ◆ Crear programas de aplicación usando lógica en escalera conmutada (RLL-Relay Ladder Logic), Cartas de Funciones Secuenciales (SFC-Sequential Function Chart), y Lenguaje de Texto Estructurado (STL-Structured Text Language).
- ◆ Definir todos los tipos de variables globales y locales de InControl, incluyendo los tipos de datos definidos por el usuario.
- ◆ Trabajar con la variedad de Objetos de Fabrica (FOE) pre-configurados (Controle ActiveX) que usted puede usar en sus proyectos.

OBJETIVOS

- ◆ Comprender las técnicas generales para configurar las tarjetas de exploración I/O y usar los dispositivos de red para el control de datos en tiempo real.
- ◆ Configurar las características de ejecución del Runtime Engine (Software de ejecución) de InControl.
- ◆ Implementar las funciones de monitoreo y corrección de errores del sistema InControl.
- ◆ Entender como el InControl se interconecta con InTouch.

PRERREQUISITOS

- ◆ Conocimiento de los sistemas operativos Microsoft Windows NT y Windows 2000.
- ◆ Experiencia en el diseño industrial.
- ◆ Familiaridad con los sistemas de control y sus respectivos lenguajes de programación.
- ◆ Familiaridad con los estilos populares de PLC, redes I/O y dispositivos.

MODULO 1

¿Qué es el InControl de Wonderware?

Es un sistema de control de arquitectura abierta en tiempo real basada en Windows NT/CE que permite diseñar, crear, probar y ejecutar programas de aplicación para controlar su proceso – más rápido que nunca.

Usted puede crear su propia solución automática en una gran variedad de lenguajes basados en graficas y textos. InControl soporta directamente interfases a una variedad de módulos I/O, de motores, sensores y otros equipos de fábrica y dan soporte a interfases antiguas así como a nuevas interfases de aparatos.

Aplicaciones InControl

Arquitectura abierta.

InControl se puede usar en cualquier plataforma que soporte el sistema operativo Windows NT, incluyendo las estaciones de trabajo industriales, los servidores SMP y controladores industriales abiertos. El sistema operativo Windows NT proporciona ventajas superiores en extensibilidad y control en tiempo real.

El InControl ha sido también pre-incrustado en plataformas seleccionadas basadas en Windows CE para bajo costo e implementación del control altamente flexible. InControl da soporte a controles distribuidos vía DCOM, con comunicación colega-a-colega integrada en el producto.

MODULO 2

REQUERIMIENTOS DEL SISTEMA INCONTROL

Sistema basado en un procesador Pentium (procesador Pentium de 200 MHz o más rápido es recomendado).

Al menos 32 MB de memoria RAM (48 MB es recomendable y 128 MB es lo recomendado para muchos programas y símbolos).

Unidad de CD-ROM.

Sistema operativo Microsoft Windows NT (versión 4.x, paquete de servicio 5).

Por lo menos 80 MB de espacio en disco duro es lo necesario en disponibilidad (128 MB es lo recomendado).

Características del monitor: VGA con 256 colores y 640 x 480 píxeles de resolución. (256 colores y 800 x 600 píxeles de resolución es lo recomendado).

Software de ejecución InControl (InControl Runtime Engine) — Sistema Windows NT. Por lo menos 32 MB en RAM (48 MB es lo recomendado) y un mínimo de 20 MB de espacio en disco duro es lo requerido para el software de ejecución.).

Software de ejecución InControl (InControl Runtime Engine) — Sistema Windows CE. InControl soporta aplicaciones específicas de proveedor en sistemas Windows CE, basados en procesos fabricados por Intel, Itachi y MIPS.

CONSIDERACIONES DE LA INSTALACIÓN

Antes de instalar el InControl, inicie su sistema Windows NT usando una cuenta con el administrador de privilegios.

Cierre cualquier programa, particularmente las aplicaciones con Wonderware, que están corriendo actualmente.

Desinstale cualquier versión previa de InControl.

INSTALACIÓN

Uso de InTouch con InControl

Si intenta usar InTouch con InControl en iguales sistemas, es recomendable que instale InTouch antes de instalar InControl.

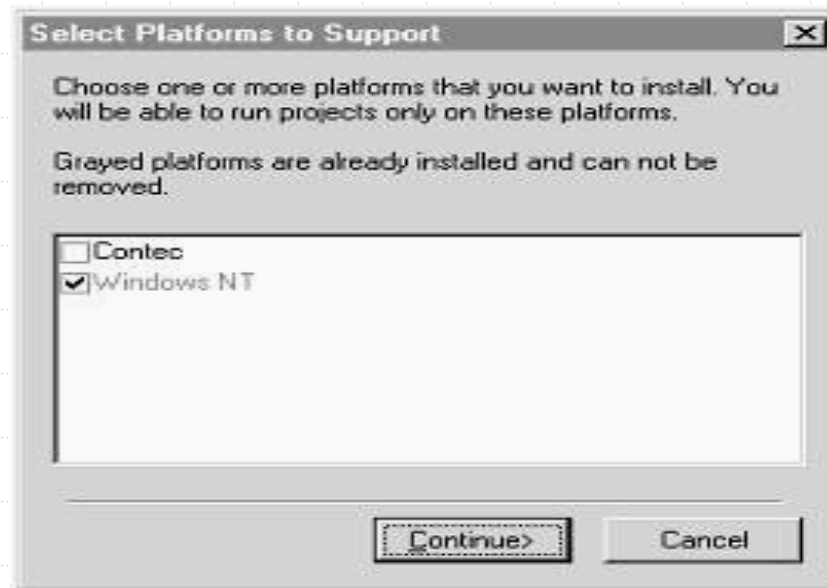
Si intenta usar InTouch e InControl sobre sistemas separados y necesita ver los símbolos de InControl desde el InTouch, instale los archivos para las extensiones sobre el sistema donde InTouch está localizado.

Corra el programa de configuración de InControl sobre el sistema de InTouch y seleccione InTouch Extensions en el box de diálogo SelectsComponents.

No es necesario instalar otros componentes.

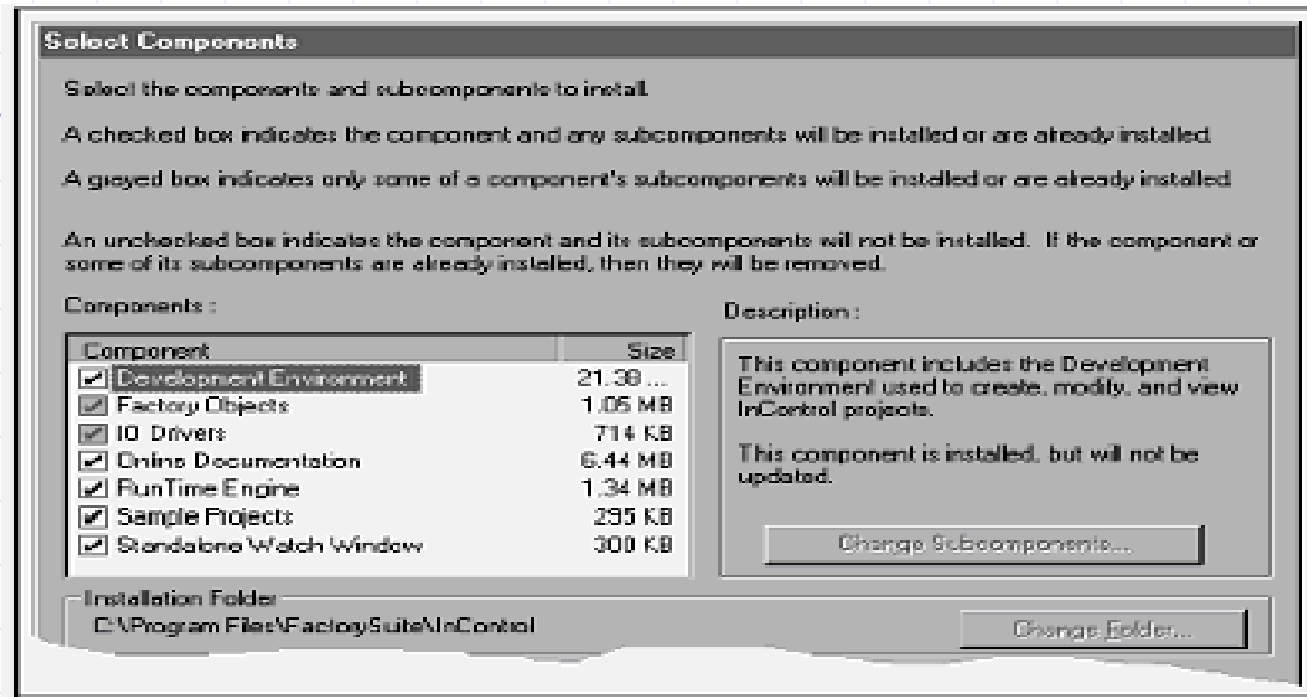
Ejecución de Proyectos sobre Múltiples Plataformas

Si ha adquirido una versión de InControl que soporte múltiples plataformas de hardware de ejecución, entonces el programa le indicará que plataforma debe seleccionar..



Selección de la plataforma

PROCEDIMIENTO DE INSTALACIÓN



Para Instalar el InControl:

1. Corra el programa de instalación de FactorySuite (SETUP.EXE) en el directorio origen del CD de InControl.
2. Complete la instalación del FactorySuite y la instalación de los Componentes comunes.

Aparecerá el cuadro de Diálogo Selección de componentes.

El programa de configuración usa los indicadores:

- Indica que los componentes y cualquier subcomponente están instalados o serán instalados.
- Indica que algunos de los subcomponentes de los componentes están instalados o serán instalados.
- Indica que los componentes y sus subcomp. No serán instalados o serán removidos.

3. En la ventana Components, seleccione los componentes que desea instalar. Puede escoger los subcomponentes de un componente, haciendo clic en el icono Choose Subcomponents cuando esta opción este disponible.

Development Environment
(Ambiente de Desarrollo)
Archivos de programa ejecutables necesarios para crear, modificar y ver los proyectos InControl.

Factory Objects
(Objetos de Fabrica)
FOE's Wonderware.
Tienen subcomponentes.

I/O Drivers
Incluyen Suitelink.
Estos componentes tienen subcomponentes.

Online Documentation (Documentación en Línea)

La guía de usuario en línea. Guías de usuario para drivers I/O individuales no son copiados si los subcomponentes de los drivers no son verificados.

Runtime Engine
(software de ejecución)
Software de ejecución Windows NT, el cual permite correr proyectos InControl.

Standalone Watch Window
(Ventana de Observación Autónoma)
Esta utilidad le permite monitorear variables durante el tiempo de ejecución sin tener que abrir el ambiente de desarrollo.

InTouch Extensions
(Extensiones InTouch)
Los asistentes (wizards) de InControl y las etiquetas de navegación para InTouch, usada para hojear las etiquetas de InControl. Sus componentes no aparecen como una opción a menos que InTouch ya esté instalado.

Sample Projects
(Ejemplos de Proyectos)
Algunos ejemplos de proyectos InControl. Tiene subcomponentes.

Continuacion Procedimiento de Instalación

4. Para cambiar un subcomponente de un componente seleccionado, haga clic en Change Subcomponents. En el cuadro de dialogo que aparece, seleccione los subcomponentes que desea, y luego haga clic en OK.
5. La ruta para la instalación del InControl aparece en el grupo de carpetas de instalación. Para especificar un directorio diferente, haga clic en Change Folder y luego vaya a la carpeta destino deseada.
6. Haga clic en Continue.
Aparecerá el cuadro de diálogo de Start Copying Files.
7. Verifique que los archivos se hayan instalado. Si la configuración es aceptable, haga clic en Next. Todos los archivos de InControl serán copiados en el disco duro.
8. Cuando se ha completado la instalación, usted tiene la oportunidad de leer las notas de introducción antes de correr el InControl. Debe instalar la licencia del InControl para tener acceso a todas las características. Para mas información sobre instalación de la licencia, ver la Guía del Administrador de Sistemas FactorySuite.

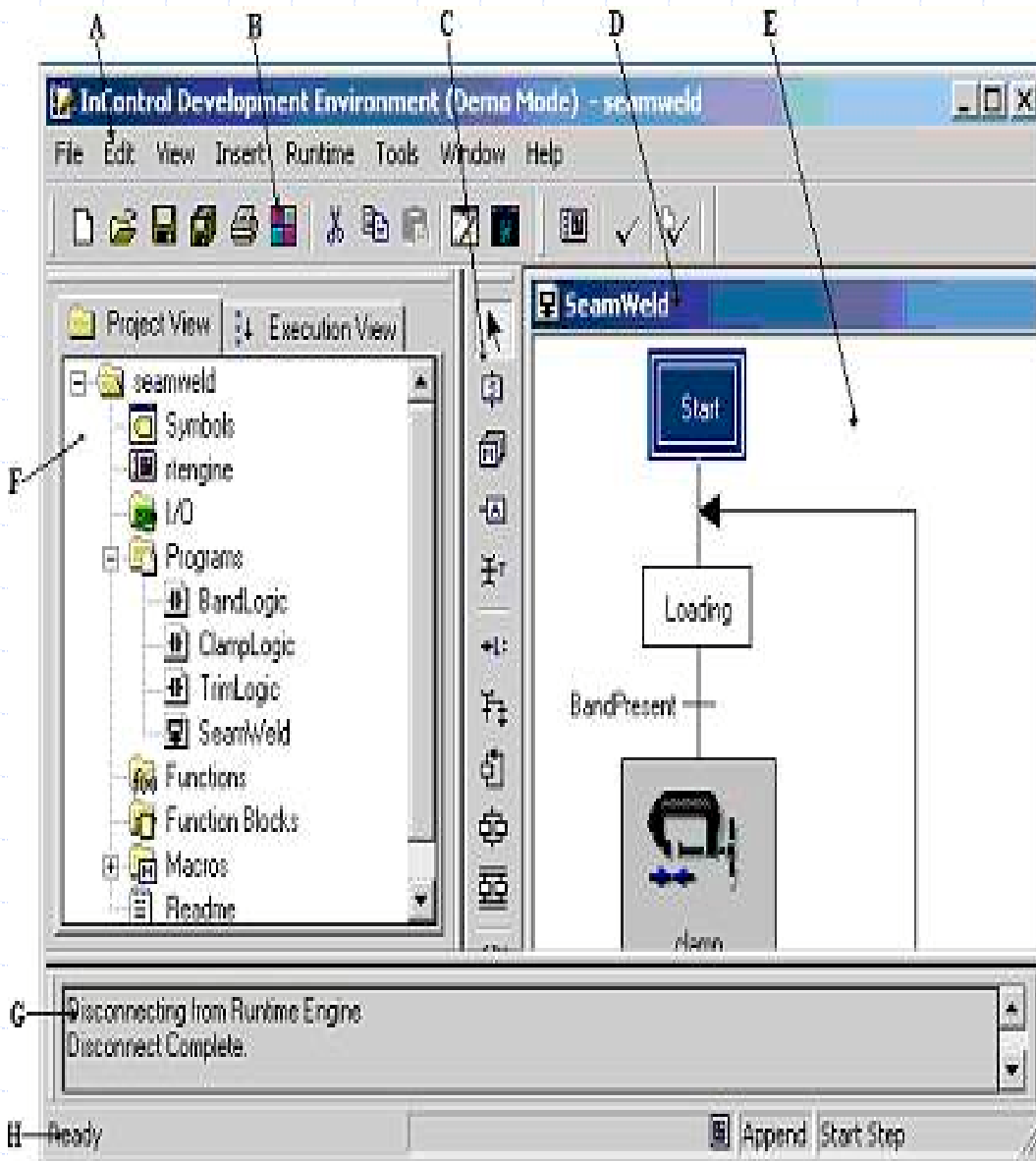
VENTANAS DE DESARROLLO Y EJECUCIÓN

El InControl consta de dos conjuntos de ventanas:

La Ventana de Desarrollo donde se pueden crear programas de aplicación, y la Ventana de Ejecución donde se ejecutan y se encarga de monitorear los programas creados.

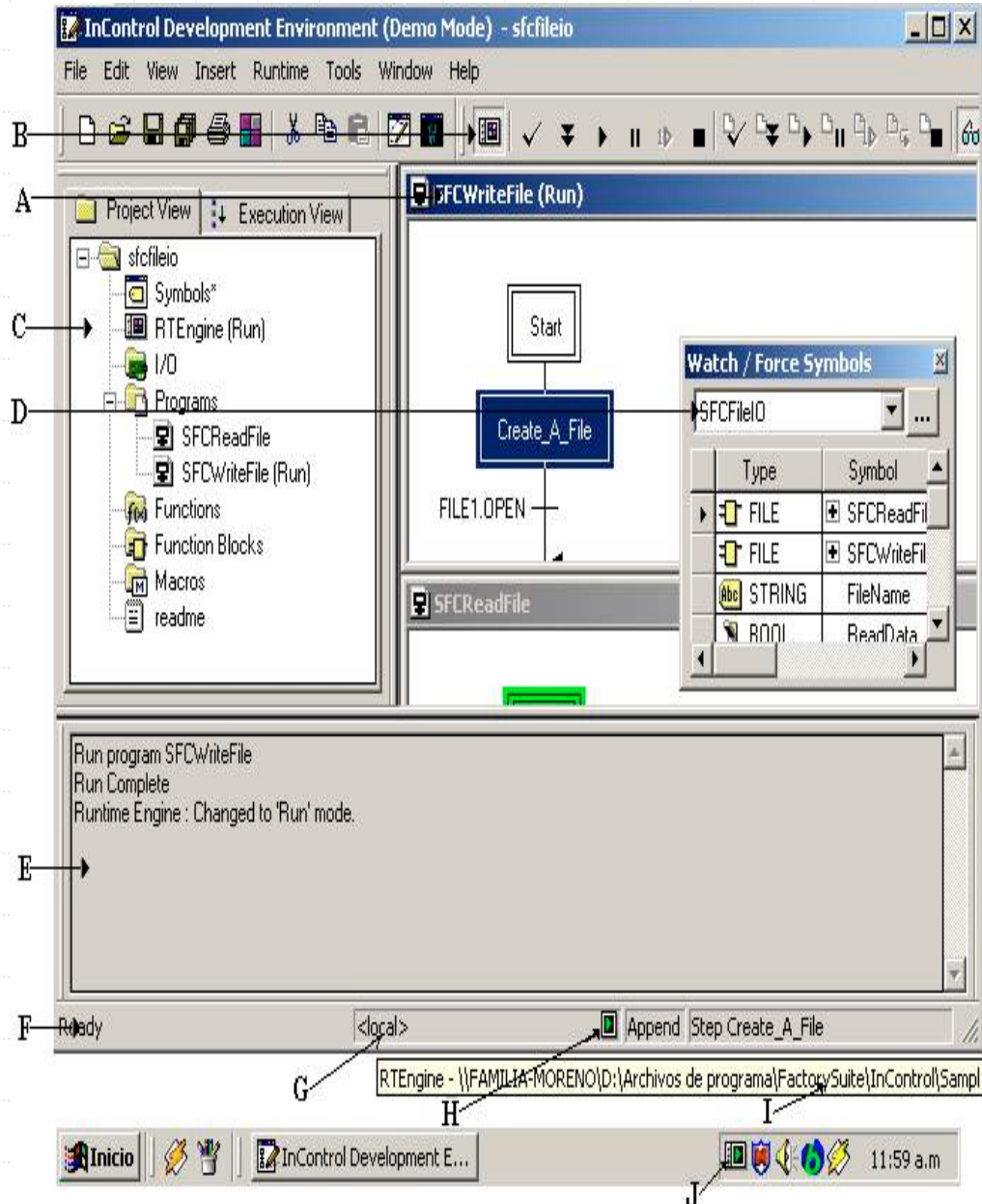
Usted puede controlar varias ventanas ocultándolas o mostrándolas.

VENTANA DE DESARROLLO



Objeto	Elementos de la Pantalla	Función
A	Barra de Menú	Muestra las funciones estándar en formato texto.
B	Barra de herramientas Estándar	Muestra las funciones estándar como iconos.
C	Barra de Herramientas del Editor	Muestra las herramientas usadas para adicionar elementos de programa a un programa.
D	Barra de Título de la Ventana de Edición	Muestra el nombre del programa.
E	Ventana de Edición	Área de trabajo para la edición.
F	Ventana de Proyectos	Muestra las funciones específicas de un proyecto: programas, organización E/S y prioridad de ejecución.
G	Ventana de Salida	Muestra los mensajes (incluyendo mensajes de error), del Runtime Engine, el compilador y otros elementos de programas.
H	Barra de Estados	Muestra información del programa o de ciertos iconos.







VENTANA DE EJECUCIÓN



Objeto	Elementos de la Pantalla	Función
A	Barra de Título del Programa	Muestra el nombre del programa.
B	Barra de herramientas de Ejecución	Muestra las funciones de ejecución estándar como iconos
C	Ventana de Proyectos	Muestra las funciones específicas de un proyecto: programas configuración E/S, organización E/S y prioridad de ejecución. .
D	Ventana Watch (Observación)	Muestra las variables (símbolos) y su estado de ejecución .
E	Ventana de Salida	Muestra los mensajes del Runtime Engine, el compilador, los elementos del programa, etc .
F	Barra de Estado	Muestra información del programa .
G	Nodo Conectado	Muestra el nodo al cual el ambiente de Desarrollo esta conectado.
H	Icono del RTE Conectado	Muestra el estado del Runtime Engine en el nodo local o remoto, al correr el proyecto.
I	Proyecto Cargado	Herramienta de información que reporta el nombre del proyecto cargado en el Runtime Engine .
J	Icono de Monitoreo del Runtime Eng.	Monitorear el estado del Runtime Engine en el nodo local

ICONOS DEL RUNTIME ENGINE

Los siguientes iconos están asociados con el icono de monitoreo del Runtime Engine y también aparecen en la barra de Estado:

Icono	Estado
	Indica que el Runtime Engine esta en el modo Run.
	Indica que el Runtime Engine esta en el modo Pause.
	Indica que el Runtime Engine esta en el modo Stop.
	Indica que el Runtime Engine esta en el modo Fault. Note que este icono no indica que un programa esta en el modo Fault.
	Indica que un mensaje ha sido enviado a la ventana de salida y al Wonderware Logger.
	Indica que una o más variables están forzadas.

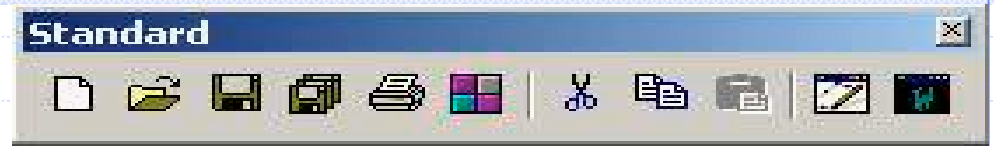
BARRA DE HERRAMIENTAS












Las funciones y comandos pueden ser seleccionados desde las barras de herramientas de la ventana de desarrollo de InControl.

Para ser visualizadas deben estar seleccionados desde el comando **View** en la barra de menú.

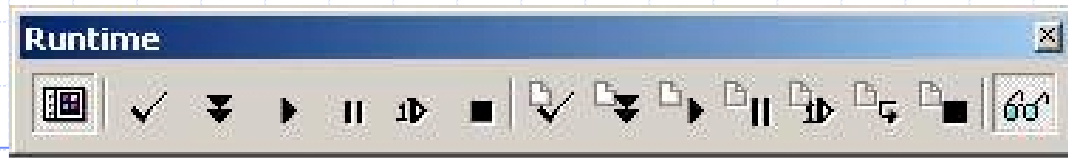
Estas barras de herramientas son: **Standar** y **Runtime**.









Barra de Herramientas Standard



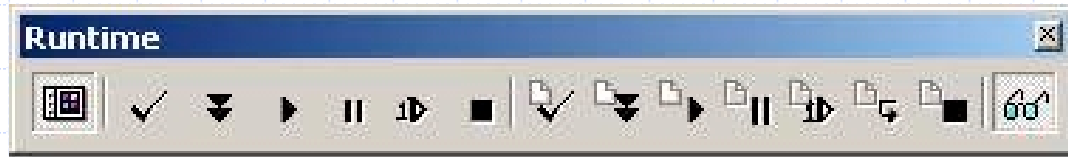
Icono	Opción en la barra de menú	Función
	New en el menú File	Crea un nuevo programa.
	Open en el menú File.	Abre un programa existente.
	Save en el menú File.	Guarda el programa activo.
	Save All en el menú File.	Guarda todos los archivos abiertos.
	Print en el menú File.	Imprime un programa.
	Project en el menú File.	Abre el Project Manager.
	Cut en el menú Edit.	Corta el objeto seleccionado y lo ubica en el portapapeles.
	Copy en el menú Edit.	Copia el objeto seleccionado y lo ubique en el portapapel.
	Paste en el menú Edit.	Pega el contenido del portapapel.
	Windows Maker en el menú Tools.	Accede al InTouch Window Maker (Ventana de creación)
	Windows Viewer en el menú Tools.	Accede al InTouch Window Viewer (Ventana de visualización)








Barra de Herramientas Runtime



Icono	Comando en el menú Runtime	Descripción
	Connect / Disconnect (Conecta / Desconecta)	Conecta el ambiente de desarrollo al Runtime Engine.
	Validate Project (Valida el Proyecto)	Valida todos los programas en un proyecto.
	Download Project (Descarga el Proyecto)	Descarga todos los programas de un proyecto al Runtime Engine.
	Run Project (Ejecuta el Proyecto)	Corre todos los programas en un proyecto.
	Pause (Pausa)	Interrumpe todos los programas que actualmente están corriendo por el Runtime Engine.
	Single Scan (Análisis Sencillo)	Ejecuta un análisis sencillo del Runtime Engine.
	Stop (Detención)	Detiene todos los programas de un proyecto que actualmente están corriendo por el Runtime Engine.
	Validate Program (Valida el Programa)	Valida el programa seleccionado.

Barra de Herramientas Runtime



Icono	Comando en el menú Runtime	Descripción
	Download Program (Descarga el Programa)	Descarga el programa al Runtime Engine.
	Run Program (Ejecuta el Programa)	Corre el programa seleccionado.
	Pause Program (Pausa el Programa)	Interrumpe un programa que actualmente esta corriendo por el Runtime Engine.
	Single Scan Program (Análisis Sencillo del Programa)	Ejecuta un análisis sencillo del programa.
	Step Program (Ejecución de un Paso en un Programa)	RLL: ejecuta un escalón. Structured Text: ejecuta una línea de código. SFC: Ejecuta una línea de código de cada Step activo. FOE: No es soportado
	Stop Program (Detiene un Programa)	Detiene un programa y los descarga desde la memoria.
	View Menu: Watch/Force variables (Menú de observación: Observar y Forzar variables)	Muestra la ventana Watch. Usted puede usar la ventana Watch de forma individual si usted no quiere abrir el Ambiente de Desarrollo o Ejecución.

FUNCIONALIDAD DE LA BARRA DE MENÚ








File Edit View Insert Runtime Tools Window Help


Usted puede escoger cualquiera de las funciones InControl desde la barra de menú, la cual se muestra en la siguiente figura.

COMANDOS FILE

Use estos comandos para operaciones con archivos, tales como abrir, cerrar, e imprimir archivos, y adicionar o remover archivos de programa del proyecto.

Comando del menú File	Icono en la barra de herramienta	Función
New		Crea un nuevo archivo de programa.
Open		Abre un archivo de programa existente.
Close	n/a	Cierra todas las ventanas asociadas con el archivo de programa activo.
Save		Guarda el archivo de programa activo.
Save As	n/a	Guarda el programa activo con un nombre diferente.
Save All		Guarda todos los archivos de programas abiertos y la información relacionada con el proyecto.
Project		Abre el Project Manager.

COMANDOS FILE

Comando del menú File	Icono en la barra de herramienta	Función
Add File to Project	n/a	Adiciona un archivo de programa al proyecto.
Remove File From Project	n/a	Remueve un archivo de programa desde el proyecto.
Print		Imprime un programa.
Print Xref (References cross)	n/a	Imprime de las variables de programa, la información de donde y cuan a menudo son usadas en el programa.
Print Setup	n/a	Le permite cambiar las opciones de impresión y la impresora.
Exit	n/a	Cierra el Ambiente de Desarrollo. Si no guarda los programas que están abiertos, usted le indicara que los guarde.

1 Usted también puede imprimir referencias cruzadas desde el Symbol Manager. Para mas información, ver el capítulo “DEFINICIÓN DE VARIABLES”.

COMANDOS EDIT

El menú **Edit** aparece solamente cuando un programa es abierto para su edición. Los comandos cambian dependiendo del tipo de programa que este en la actual ventana activa. El InControl le indica que un comando ha sido seleccionado por que se coloca una marca o señal en este.

Comandos del menú Edit	Icono en la barra de herramienta	Función
Undo	Todos	Deshace la ultima acción.
Redo	RLL, SFC,	Rehace previamente una acción deshecha que involucra un elemento de programa.
Cut	STL	Corta el objeto seleccionado y lo coloca en el portapapeles.
Copy	Todos	Copia el objeto seleccionado y lo coloca en el portapapeles.
Paste	Todos	Pega el contenido del portapapeles.
Delete	Todos	Borra el objeto seleccionado.

COMANDOS EDIT

Comandos menú Edit	Icono en la barra de herramienta	Función
Select All	SFC-Stp	Selecciona todas las líneas de código.
Mark line	STL	Se usa para seleccionar una o más líneas de código.
Edit Element	SFC-Stp	Abre el cuadro de dialogo para el elemento de programa seleccionado.
Step Properties	STL	Abre el cuadro de dialogo para un paso.
Find	SFC, RLL	Encuentra el texto especificado.
Find Next	SFC	Encuentra el siguiente caso del texto especificado.
Replace	Todos	Reemplaza el texto especificado con un nuevo texto.
Go To	Todos	Muestra la localización seleccionada en el programa.
Go To Coil	Todos	Muestra el caso siguiente de la bobina seleccionada.
Boolean Transition	SFC-Stp,RLL	Ajusta por defecto la transición para todos los programas SFC abiertos, al tipo Boolean, en lugar del tipo RLL.
Lock Algorithms	SFC	Se usa para agregar la clave de protección para los algoritmos.
Set Bookmark	SFC-Stp,STL	Marca una o más localizaciones en un programa y usa el comando Go To para saltar entre estos.
Set Bookmark	FOE	Muestra el cuadro de dialogo de configuración.

RLL = Lógica en escalera por relevos FOE = Objeto de fabrica InControl STL = Lenguaje de texto Estructurado SFC-Stp = Paso SFC
 SFC = Cartas de funciones secuenciales

Comandos Insert

Los comandos **Insert** (Insertar) le permiten colocar elementos de programa herramientas de programas. dentro de un programa desde la barra de Menú, en lugar de usar la barra de

Comandos View

El menú **View** muestra en la pantalla elementos que usted puede mostrar o esconder. Los elementos cambian dependiendo del tipo de programa que este en la actual ventana activa. El InControl le indica que un elemento ha sido seleccionado por colocar una marca o señal en este.

Comandos Runtime

Los comandos **Runtime** le permiten a usted ajustar los parámetros de ejecución, observe el dato de estado, y enviar comandos directamente al Runtime Engine.

Comandos Tools

Use los comandos **Tools** para llamar a otros programas o utilidades para su ejecución.

Comandos Window

Use los comandos **Windows** para organizar las ventanas.

El menú **Window** no aparece a menos que usted abra una ventana de edición.

Comandos Help

Use los comandos **Help** para mostrar información del sistema y las opciones de ayuda.

Uso de la Watch Window (Ventana de observación)

Usted puede correr una versión independiente de la ventana Watch.

Esta ventana opera muy similar a la ventana Watch que abre desde el ambiente de desarrollo. Sin embargo, con la ventana Watch independiente, usted puede monitorear variables durante la ejecución, sin abrir el Ambiente de Desarrollo.

Además, puede correr múltiples instancias de la ventana Watch y monitorear las variables en uno o mas nodos.

Para correr la Ventana de Observacion (Watch):

- ◆ Sobre el icono del monitor de la máquina, pulse **Watch window**. Usted también puede correrlo desde el menú **Start** de la barra de tareas.
- ◆ Si la caja de diálogo le sugiere un nodo, entre el nombre de el nodo que quiere monitorear. Dependiendo del sistema designado, puede necesitar entrar la dirección TCP/IP. Para el nodo local, deje el campo **Node** en blanco.
- ◆ Verifique el proyecto que intentó monitorear. El nombre del nodo y el proyecto aparece en la barra de Estado en la parte inferior de la ventana **watch**. Si ubica el cursor en la barra de estado, el ayudante también muestra el nombre del proyecto

WATCH WINDOW (VENTANA DE OBSERVACIÓN)

Watch / Force Symbols - 1155

File View Security Help







Watch1

Type	Symbol	Value
Program	+ RLL1	
Program	+ RTEngine	


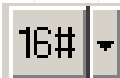
\\DYB\C:\Program Files\FactorySuite\Common\InContro

\\DYB\C:\Program Files\FactorySuite\Common\InControl\1155

Los comandos del menú para la ventana Watch:

Opción de la barra de menú	Comando	Icono de la barra de herramientas	Descripción
File	Project		Accede al Project Manager.
File	Connect /Disconnect		Conecta y desconecta la ventana del ambiente de desarrollo.
File	Add		Adiciona símbolo a la Watch window.
File	Remove		Remueve los símbolos desde la ventana Watch window.
File	Insert Empty Row	N/a	Inserta una fila entre las variables.
File	View Symbol Definition	N/a	Muestra la definición de los símbolos en el Symbol Manager.
File	Modify Value		Escribe o forzar un valor a una variable.
File	Unforce		Cede variable seleccionada.
File	Unforce All		Cede todas las variables que han sido forzadas.

Los comandos del menú para la ventana Watch:

Opción de la barra de menú	Comando	Icono de la barra de herramientas	Descripción
File	Show All Forces		Visualiza todas las variables que han sido forzadas.
File	Exit	N/a	Cierra las ventanas Watch.
View	Toolbar	N/a	Expone la barra de herramientas de la ventana Watch.
View	Status Bar	N/a	Expone la barra de estado de la ventana watch.
View	Format		Selecciona el formato (binario, octal, decimal, o hexadecimal) para tipos de datos ANY_INT (BYTES, WORDs, etc) o tipos de datos ANY_REAL (REAL, LREAL).
Security ¹	Log On	N/a	Accede a la caja de diálogo Log On.
	Change Password	N/a	Accede a la caja de diálogo Change Password.
Security ¹	Configure Users	N/a	Accede a la caja de diálogo Configure Users.
Security ¹	Log Off	N/a	Accede a la caja de diálogo Log Off.
Help			Expone ayuda Online para el standalone de la ventana Watch.
1 La configuración de la seguridad está descrita en el capítulo “Fijación de la seguridad”.			

También se pueden seleccionar funciones para la ventana Watch ubicando el cursor dentro de la ventana y pulsando el botón izquierdo del mouse.

- Ventana de Proyectos (Project Window)
- Explorando y leyendo entradas por el Runtime Engine

MODULO 3

CREANDO, DESCARGANDO Y EJECUTANDO UN PROGRAMA O PROYECTO

- SECCIÓN 1: Creando, Modificando y Removiendo un Proyecto
- SECCIÓN 2: Validando, Descargando y Ejecutando
- SECCIÓN 3: Configuración del Runtime Engine
- SECCIÓN 4: Configuraciones I/O

SECCIÓN 1: Creando, Modificando y Removiendo un Proyecto

- Consideraciones InControl
- Creando un proyecto
- Modificando la lista de directorio

SECCIÓN 2: Validando, Descargando y Ejecutando

- Validando, descargando y ejecutando
- Validando un programa o proyecto
- Descargando un programa o proyecto
- Ejecutando un programa o proyecto
- Detección de un programa o proyecto

SECCIÓN 3: Configuración del Runtime Engine

- Consideraciones del Ambiente de Desarrollo
- Configurando el Runtime engine
- Elementos del cuadro "Runtime Engine Properties"

SECCIÓN 4: Configuraciones I/O (entrada/salida)

- Configuraciones I/O
- Cuadro de dialogo de configuración de drivers.

MODULO 4

SYMBOL MANAGER

- SECCIÓN 1: Ingresando al Symbol Manager
- SECCIÓN 2: Definiendo variables
- SECCIÓN 3: Watch/Force Symbol

SECCIÓN 1: Ingresando al Symbol Manager

- Symbol Manager

SECCIÓN 2: Definiendo variables

- Acerca de variables (Símbolos)
- Variables locales
- Variables globales
- Tipos de datos de las variables
- Variables de sistema general
- Variables del runtime Engine
- Arreglos de variables
- Variables definidas por usuario

SECCIÓN 3: Watch/Force Symbol

- Watch/Force Symbol Window
- Watch/Force Symbol Windows Barra de Herramientas
- Creación de una Watch/Force Symbol Window
- Adicionando variables a la ventana de observación (Watch Window)
- Modificando valores de variables

• **LAB 1: Creación de un proyecto**

• **LAB 2: Creación de variables (símbolos)**

MODULO 5

LOGICA EN ESCALERA CONMUTADA

RLL (Relay Ladder Logic)

- SECCIÓN 1: Programas RLL
- SECCIÓN 2: Barra de herramientas RLL
- SECCIÓN 3: Contactos y bobinas
- SECCION 4: Bloques de funciones

SECCIÓN 1: Programas RLL

- Programación en RLL

SECCIÓN 2: Barra de herramientas RLL

- Barra de herramientas RLL

SECCIÓN 3: Contactos y bobinas

- Contactos RLL
- Tipos de contactos
- RLL bobinas
- Consejos de edición

LABORATORIOS MODULO 5

- LAB 3 - Contactos/Bobinas y Logica And/Or
- LAB 4 - Temporizadores (Timers)
- LAB 5 - Contadores (Counters)
- LAB 6 - Bloques Matemáticos

SECCIÓN 4: Bloques de funciones

- Funciones y bloques de función predefinidos y de uso definido.
- Paleta de bloques de función.
- Adicionado de funciones.
- Bloques de función de operadores de bit .
- Comparación de bloques de función.
- Convesión de bloques de función.
- Bloques de función de contadores/temporizadores.
- Bloques de función de Archivos.
- Bloques de función Matematicos.
- Bloques de función de cadenas de caracteres.
- Bloques de función trigonométricos/logarítmicos.
- Bloques de función Trigger (disparador).

MODULO 6

CARTAS DE FUNCIONES SECUENCIALES - SFC

- SECCIÓN 1: Cartas de funciones secuenciales SFC
- SECCIÓN 2: Barra de herramientas SFC
- SECCIÓN 3: Creación de un programa SFC
- SECCION 4: Asistente InControl en InTouch

SECCIÓN 1: Cartas de funciones secuenciales SFC

- Programación SFC
- Lenguaje de Programación SFC.

SECCIÓN 2: Barra de herramientas SFC

- Barra de herramientas SFC

SECCIÓN 3: Creación de un programa SFC

- Creación de un programa SFC

SECCIÓN 4: Asistente InControl en InTouch

- Asistente InControl

LABORATORIOS

- LAB 7 – La fabrica de chocolates Milk.
- LAB 8 – Adición de una divergencia.
- LAB 9 – Adición de una detección de emergencia.
- LAB 10 – Utilizando funciones de archivo.
- LAB 11 – Conectando con InTouch.

MODULO 7

LENGUAJE DE TEXTO ESTRUCTURADO STL

- SECCIÓN 1: Lenguaje de texto estructurado
- SECCIÓN 2: Barra de herramientas de texto estructurado
- SECCIÓN 3: Creando un programa de texto estructurado
- SECCION 4: Código del programa de texto estructurado

SECCIÓN 1: Lenguaje de texto estructurado

- Elementos de STL
- Expresiones
- Operadores
- Tipos de datos

SECCIÓN 2: Barra de herramientas de texto estructurado

- Barra de herramientas STL

SECCIÓN 3: Creación de un programa STL

- Creación de un programa STL

SECCIÓN 4: Código del programa de STL

- Tipos de ordenes
- Funciones y procedimientos de llamadas
- Grupos de funciones y bloques funcionales
- Descripción de la sintaxis de una función

LABORATORIOS

- LAB 12 – Creación de un programa de texto estructurado.

MODULO 8

OBJETOS DE FABRICA (FACTORY OBJECTS)

- SECCIÓN 1: Objetos de fabrica
- SECCIÓN 2: Adición y configuración de objetos de fabrica

SECCIÓN 1: Objetos de fabrica

- Objetos de fabrica
- Integración de objetos de fabrica con InControl

SECCIÓN 2: Adición y configuración de objetos de fabrica

- Instalando Objetos de fabrica
- Configurando Objetos de fabrica
- Cuadro de dialogo "PID Control Properties"
- Objetos de fabrica PID

LABORATORIO

- LAB 13 – Adicionando, configurando y ejecutando un Objeto de Fabrica.

MODULO 9

SEGURIDAD

- SECCIÓN 1: Niveles de seguridad
- SECCIÓN 2: Registrando y Adicionando un nuevo usuario

SECCIÓN 1: Niveles de seguridad

- Seguridad

SECCIÓN 2: Registrando y Adicionando un nuevo usuario

- Registro en el sistema
- Adición de usuarios



CONCLUSIONES

FIN DE LA PRESENTACIÓN



CORPORACIÓN UNIVERSITARIA TECNOLÓGICA DE BOLÍVAR
MMIII®

**TUTORIAL DE MANEJO DEL SOFTWARE
INCONTROL**

MANUAL DE MANEJO

Mayo, 2003

CONTENIDO

	Pag.
INTRODUCCIÓN	
OBJETIVOS	
1. REQUERIMIENTOS DE INSTALACIÓN	3
2. EJECUCIÓN DEL TUTORIAL	4
2.1 PÁGINA DE PRESENTACIÓN	4
3. PANTALLA PRINCIPAL	6
3.1 SECCIONES DE LA PÁGINA PRINCIPAL	7
4. ESTILOS DE PÁGINAS Y VENTANAS	11
4.1 TEXTO Y GRÁFICOS	11
4.2 ANIMACIONES	12
4.3 PELÍCULAS	14
4.4 ACCESO A PDF	16
4.5 VENTANAS AUXILIARES	16
4.5 VENTANAS DE LA BARRA ESTÁNDAR	17
5. RECOMENDACIONES PARA EL USO DEL TUTORIAL	19

LISTA DE FIGURAS

	Pag.
Figura 1. Pantalla de presentación	4
Figura 2. Página principal	6
Figura 3. Área del contenido	9
Figura 4. Estilo texto y gráfico	12
Figura 5. Ejemplo de ejercicio con ventana animada	13
Figura 6. Ejercicio insertado directamente a la página	14
Figura 7. Presentación de una película	15
Figura 8. Icono de acceso a las películas	15
Figura 9. Icono de acceso a los manuales PDF	16
Figura 10. Ejemplo de una ventana auxiliar	17
Figura 11. Formato de una ventana de la barra estándar	18

LISTA DE CUADROS

	Pag.
Cuadro 1. Requerimientos de hardware y software	3
Cuadro 2. Tareas de la barra estándar	8

INTRODUCCIÓN

Este manual contiene la información necesaria para que el usuario tenga un mejor aprovechamiento del uso del **TUTORIAL DE MANEJO DEL SOFTWARE INCONTROL**. Describe cada una de las herramientas que conforman las páginas (íconos, ventanas y animaciones), sus funciones y modo de empleo, los requerimientos de instalación, y recomendaciones para una navegación más eficiente a través del contenido del Tutorial.

OBJETIVOS

- Introducir al usuario en el ambiente de trabajo del TUTORIAL DE MANEJO DEL SOFTWARE INCONTROL.
- Facilitar el manejo del tutorial para un mejor aprendizaje de este.

1. REQUERIMIENTOS DE INSTALACIÓN

Los siguientes cuadros muestran las características técnicas físicas y lógicas necesarias para ejecutar el tutorial de una manera completa y satisfactorias.

Cuadro 1. Requerimientos de hardware y software

DETALLES	REQUERIMIENTOS MÍNIMOS	R. RECOMENDADOS
Hardware		
Procesador	Pentium 333 MHz	Pentium IV
Memoria RAM	64 MB	258 MB
Espacio disco duro	60 MB	2.5 GB
Tarjeta de video	8MB	32 MB
Tarjeta de sonido	10 bits	16 bits
Monitor	VGA	SVGA
Software		
Sistema operativo	Windows 98	Windows 98,2000, Mellenium, XP Linux (Suse, Mandrake)
Software Adicional	Acrobat 3.0 Flash	Acrobat 4.0, Flash
Browser	Internet Explorer 5.5 Nescape 4.0	Internet explorer 6.0 Nescape 6.0, Opera 5.0

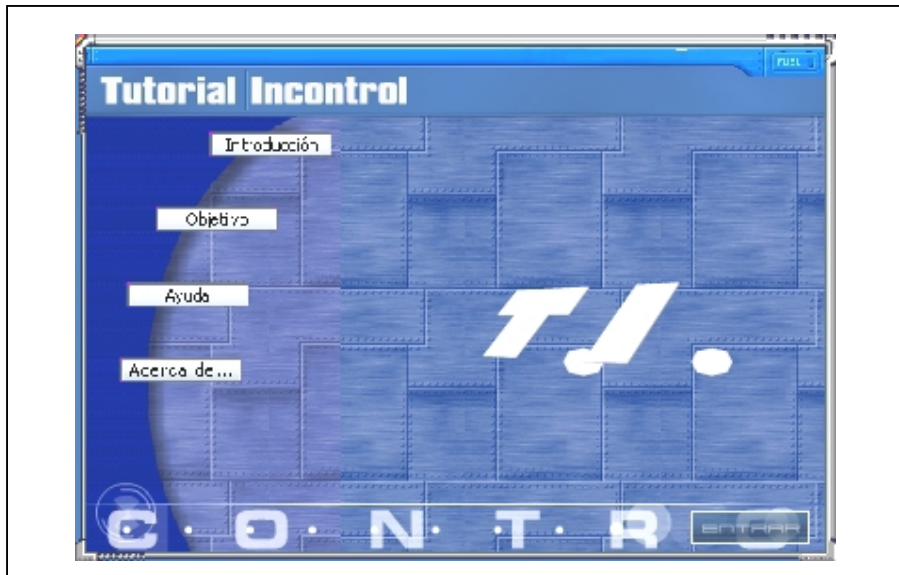
2. EJECUCIÓN DEL TUTORIAL

El tutorial esta incluido en un CD titulado “**TUTORIAL DE MANEJO DEL SOFTWARE INCONTROL**” y para acceder a su contenido debe ejecutar el archivo **presentación** el cual aparece en el momento de abrir el disco.

2.1 PÁGINA DE PRESENTACIÓN

La siguiente figura muestra la página de presentación que logra ver luego de ejecutar el tutorial.

Figura 1. Pantalla de presentación

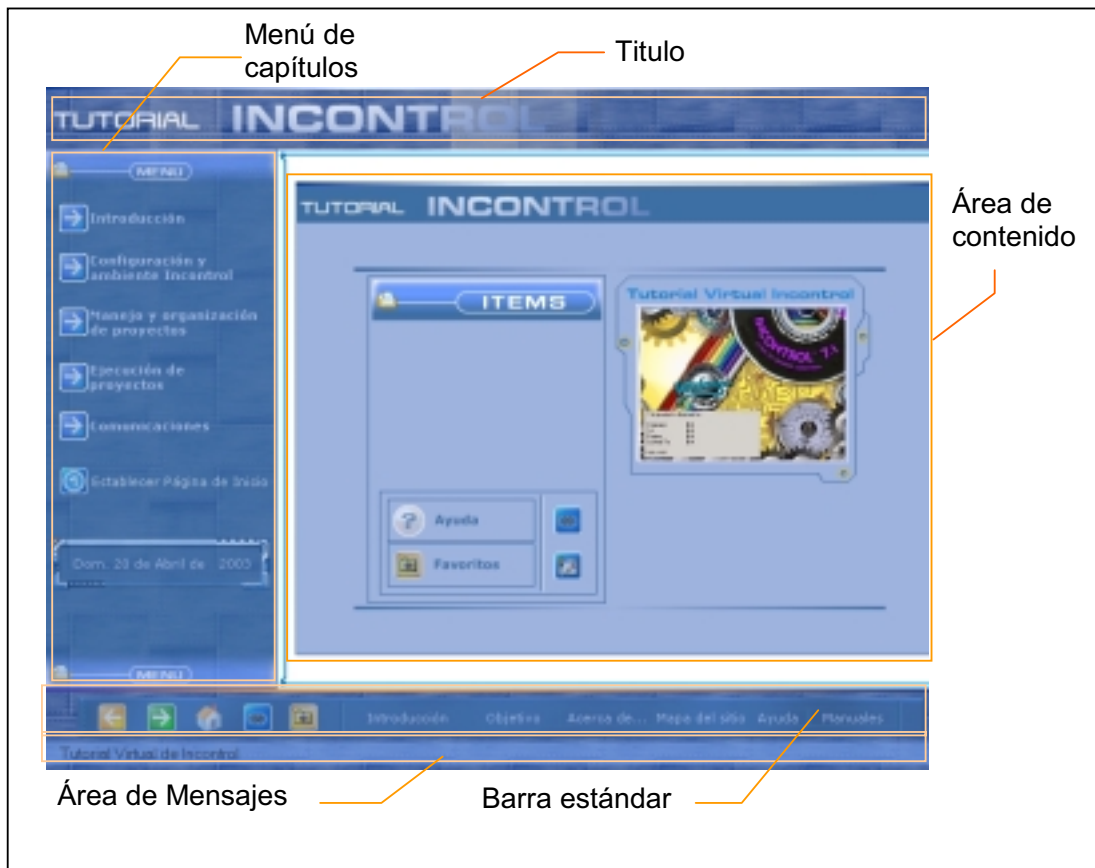


Durante la animación de la presentación podrá leer los **requerimientos** de software necesarios para una correcta ejecución del tutorial. También aparecerá la sugerencia de dirigirse a la **Ayuda** para tener un conocimiento general del manejo de este tutorial antes de empezar a trabajar en él.

3. PANTALLA PRINCIPAL

En la primera página y durante todo el tutorial, la pantalla se encuentra dividida en cinco secciones: título, menú de capítulos, área del contenido, la barra estándar y área de mensajes. La siguiente gráfica presenta cada una de estas secciones.






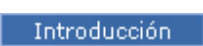




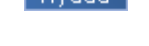
Figura 2. Página principal



3.1 SECCIONES DE LA PANTALLA PRINCIPAL.

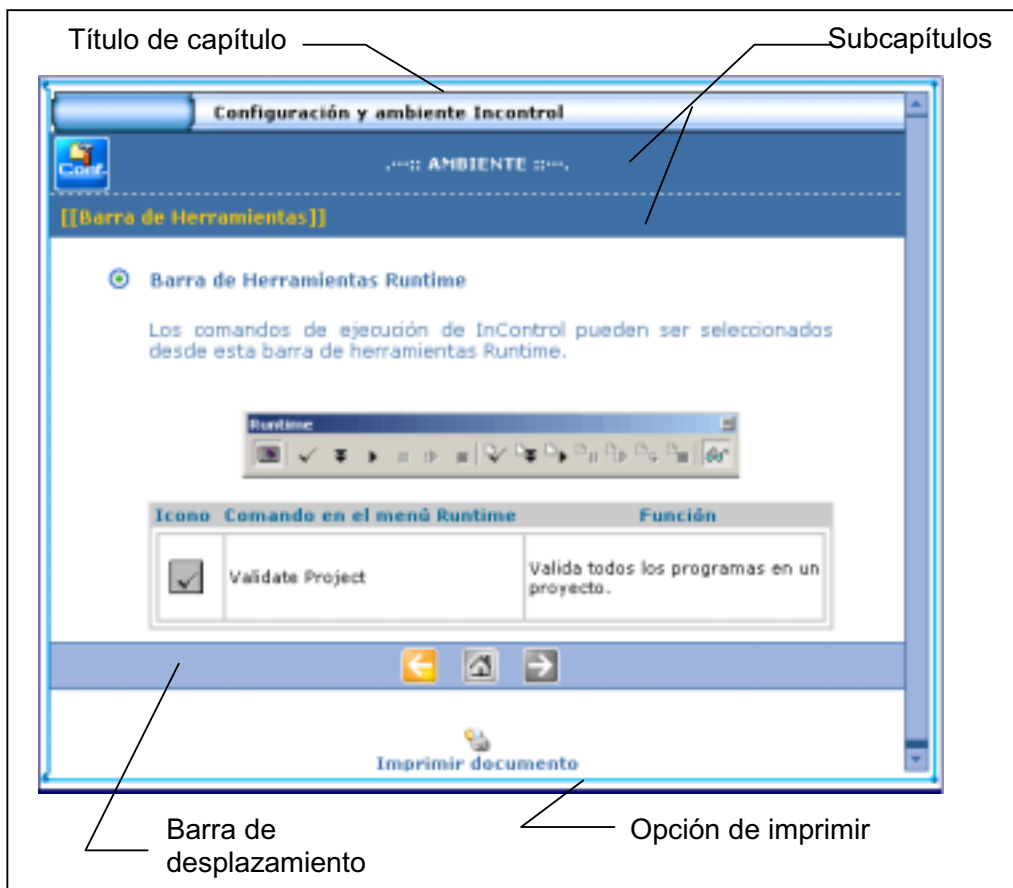
- **Título:** muestra el título del programa “TUTORIAL INCONTROL”.
- **Menú de capítulos:** desde aquí tiene acceso a cada uno de los capítulos que conforman el tutorial con solo pulsar una vez sobre cada uno de títulos. Estos son: Introducción, Configuración y Ambiente InControl, Manejo y organización de proyectos, Ejecución de proyectos y Comunicaciones. La opción “Establecer página de inicio”, le permite establecer la página actual cómo su página de inicio predeterminada. El cuadro en la parte inferior de ésta área le indica la fecha actual de trabajo.
- **Área de mensajes:** si ubica el *mouse* sobre cada uno de los íconos de la página, ésta área le informa de antemano la tarea que podrá realizar.
- **Barra estándar:** puede ejecutar las tareas principales de toda página web y conseguir información referente al tutorial.

Cuadro 2. Tareas de la barra estándar

Íconos	Tareas
	Retrocede a la página anterior.
	Avanza a la siguiente página (visitada).
	Abre la página de inicio.
	Abre la página de vínculos.
	Agrega la página actual a la lista de favoritos.
	Abre la ventana que contiene la introducción de este tutorial virtual.
	Abre la página de los objetivos.
	Abre la ventana referente a la producción del tutorial virtual InControl.
	Muestra el Mapa de desplazamiento de todo el contenido de los capítulos que conforman el tutorial.
	Ingresa a Ayuda para la óptima navegación la adel tutorial.
	Permite el acceso a los manuales de apoyo para usuarios y webmater.

- **Área del contenido:** es el lugar donde podrá ver las páginas que conforman el contenido de los capítulos del tutorial. Después de ingresar a una de las páginas del contenido desde el menú, observe de arriba hacia abajo para distinguir el nombre del capítulo al que pertenece esta página (Configuración y ambiente InControl), luego sobre un fondo azul aparece el subtítulo (Ambiente) y mas abajo un subtítulo con un orden inferior (Barra de herramientas). Con esto podrá conocer exactamente su posición respecto al todo el contenido y así ubicarse en el Mapa de desplazamiento.

Figura 3. Área del contenido



Posteriormente puede leer el tema seleccionado. En la parte inferior de esta área se encuentra una barra de desplazamientos (Figura 3) con tres íconos los cuales le permitirán moverse entre las páginas anteriores, posteriores y la página de inicio. Debajo de esta barra puede utilizar la opción de “imprimir documento” para imprimir la página que se encuentra abierta actualmente.

4. ESTILOS DE PÁGINAS Y VENTANAS

Mientras avance dentro del área de contenido del TUTORIAL DE MANEJO DEL SOFTWARE INCONTROL utilizando las barras de desplazamiento o el menú de capítulos podrá distinguir varios estilos de página, las cuales se diferencian por las diversas opciones de trabajo que contienen, estos son:

- Textos y gráficos.
- Animaciones.
- Películas.
- Acceso a los manuales en PDF.
- Ventanas auxiliares.
- Ventanas de la barra estándar.

4.1 TEXTO Y GRÁFICOS

Este es el estilo mas sencillo dentro del tutorial y puede contener solo texto o estar acompañado de gráficos. No poseen ningún tipo de animación y lo podemos encontrar en el primer capítulo o en algunos ejercicios que requieran de una sola imagen.

Figura 4. Estilo texto y gráfico

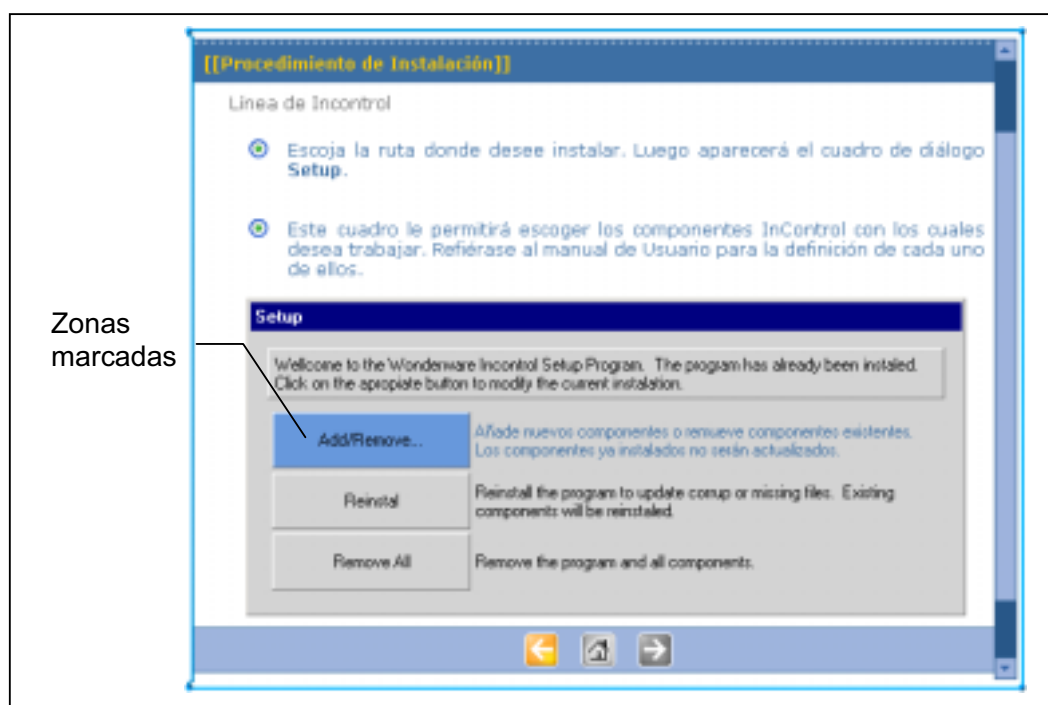


4.2 ANIMACIONES

Para disminuir la extensión de las páginas y evitar el uso de los *scrolling* existen objetos (iconos, textos, figuras) que abren ventanas con ejercicios animados con solo pulsar sobre ésta. Es necesario que lea el texto que acompaña la figura antes de abrir la ventana, luego siga las zonas marcadas, sino encuentra inicialmente zonas marcadas mueva el cursor sobre la pantalla.

Algunas veces no será necesario abrir una nueva ventana sino que podrá trabajar directamente sobre la pantalla principal. Para esto lea detenidamente los pasos descritos y siga las zonas marcadas, sino encuentra inicialmente zonas marcadas mueva el cursor sobre la pantalla.

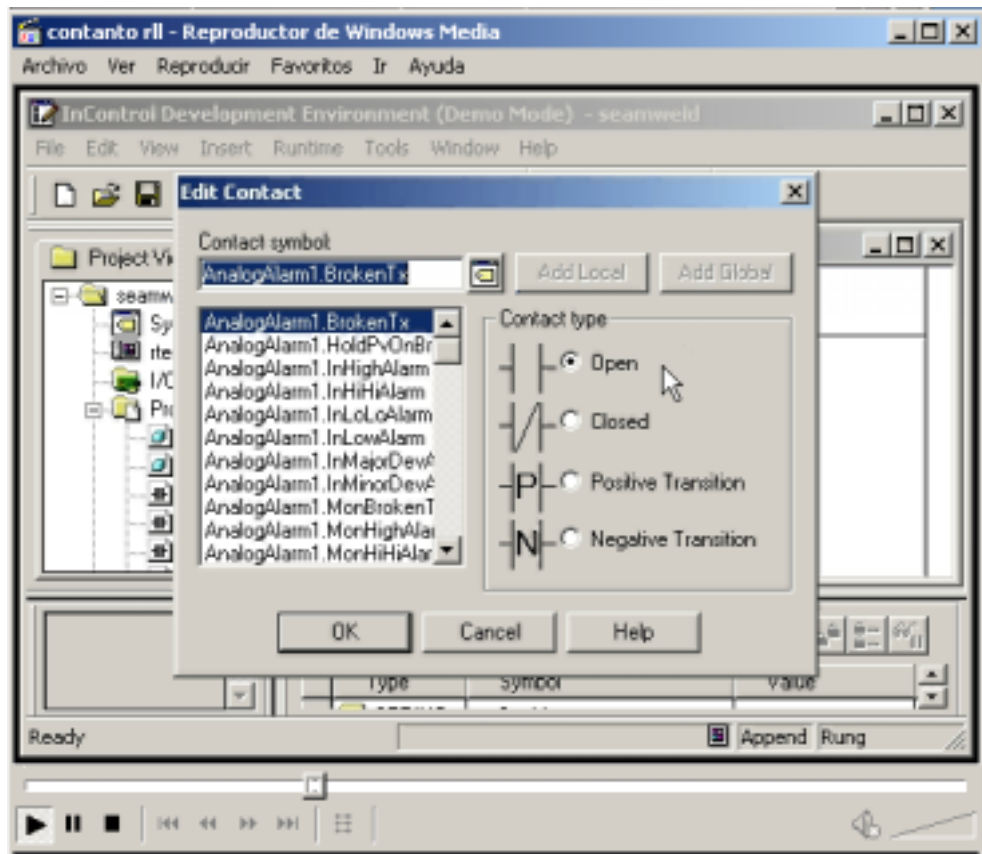
Figura 6. Ejercicio insertado directamente a la página



4.3 PELÍCULAS

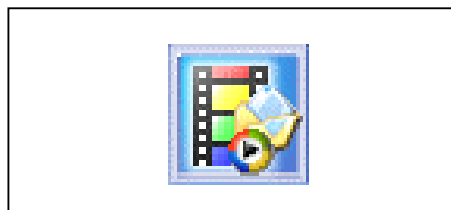
Otra opción para disminuir la extensión de las páginas es el uso de películas, las cuales visualizan los diferentes pasos que se deben seguir para una tarea determinada dentro del software InControl.

Figura 7. Presentación de una película



Dichas películas son reproducidas al pulsar sobre un icono como el de la figura 7.

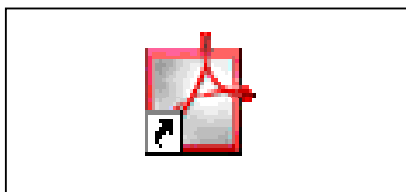
Figura 8. Icono de acceso a las películas



4.4 ACCESO A PDF

Si pulsa el icono que muestra la siguiente figura dentro del área de contenido, tendrá acceso directo a un capítulo específico de los manuales de usuario que acompañan al Tutorial y que está relacionada con el tema donde se encuentra el icono.

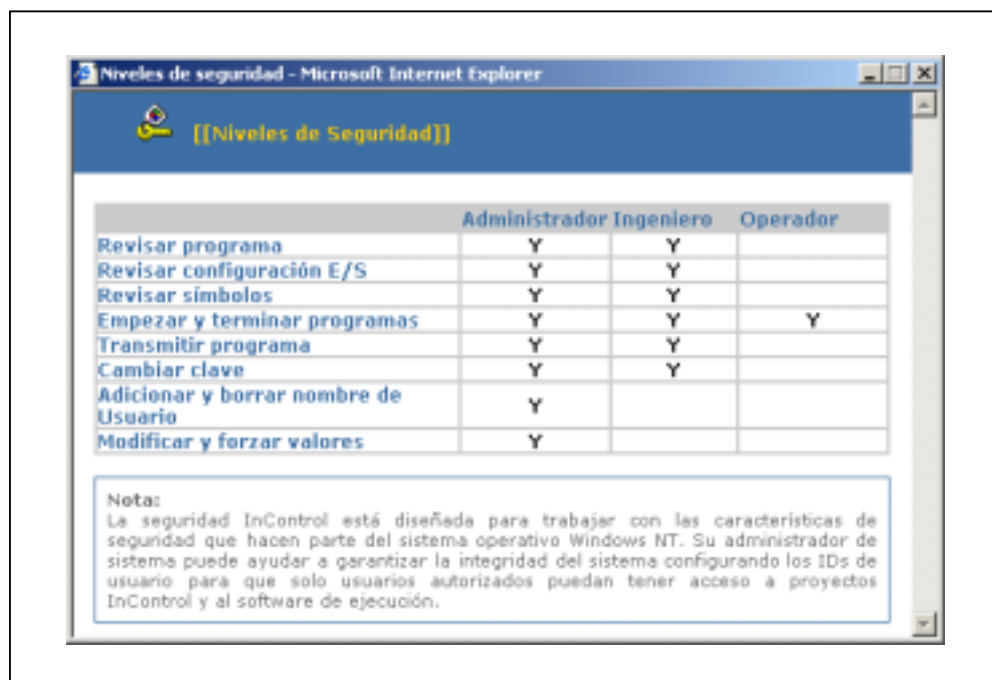
Figura 9. Icono de acceso a los manuales PDF



4.5 VENTANAS AUXILIARES

Estas ventanas guardan información complementaria de ciertos términos que necesitan ser definidos para una mejor comprensión del tema. Para acceder a ellas debe pulsar dicho término dentro del área del contenido, de esta manera se despliega una nueva ventana. La siguiente figura muestra la ventana correspondiente al tema **niveles de seguridad**.

Figura 10. Ejemplo de una ventana auxiliar



4.6 VENTANAS DE LA BARRA ESTÁNDAR

Esta es el estilo de presentación de las ventanas que conforman la barra estándar, debe pulsar el icono correspondiente y para cerrar pulse el botón de color rojo.

Figura 11. Formato de una ventana de la barra estándar



5. RECOMENDACIONES PARA EL USO DEL TUTORIAL

Las siguientes son recomendaciones para una navegación mas eficiente dentro del sistema del tutorial.

- Lea detenidamente los íconos **Introducción**, **Objetivos** y **Ayuda** de la **Barra Estándar**.
- Para estudiar el contenido siga el mismo orden que muestra el menú de los capítulos.
- Si no tiene conocimientos previos del software InControl, lea detenidamente el capítulo de **Introducción**.
- Debe Instalar el software InControl en su sistema para alternar los ejercicios de las animaciones del tutorial con las practicas en su computador y así tener mayor dominio del ambiente. Instale todos los componentes que le sugiera el cuadro de diálogo *Select Component* según los pasos de **línea de instalación /Instalación /Configuración y Ambiente InControl** del tutorial.

- Abra uno de los proyectos creados desde el Project Manager según se describe en el tema **Ejecución del InControl / Ambiente** del capítulo **Configuración y Ambiente InControl**, para trabajar los siguientes temas del tutorial sobre dicho proyecto.
- Registre su sistema de seguridad en el nivel de **Administrator**, según lo indica el tema de **Instalación** del capítulo **Configuración y Ambiente InControl**, para tener acceso todas las tareas disponibles en el software InControl y que son descritas en los capítulos posteriores del tutorial.
- Trabaje cada una de las tareas descritas en los capítulos **Manejo y Organización de Proyectos** y **Ejecución de proyectos** sobre los proyectos ya creados. Haga uso de los **Manuales** para mas información acerca de estos temas.

LABORATORIO #1

CREACIÓN DE PROYECTOS

OBJETIVOS.

- Conocer el punto de partida y los pasos necesarios en la implementación de una estrategia de control en tiempo real con base en el software InControl.
- Identificar los principales elementos que conforman un proyecto.

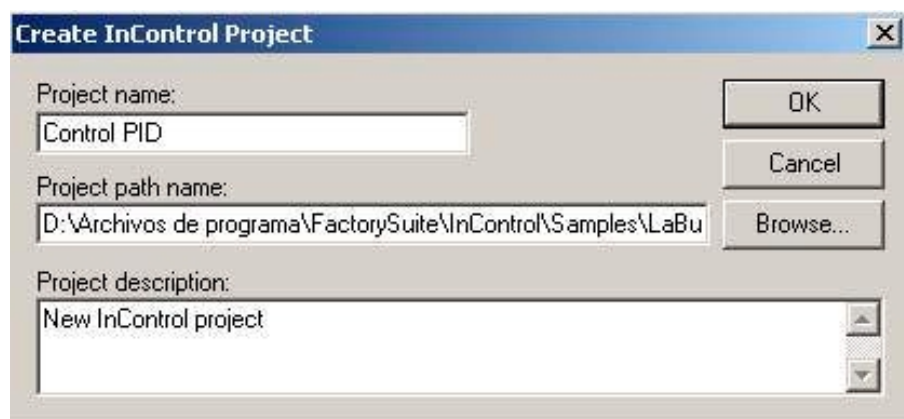
PRELIMINARES: se enuncia lo que se debe hacer para realizar este laboratorio, como los temas que debe estudiar o leer y que lo primordial es que debe haber recibido la aplicación. El estudiante o la persona que va a realizar este laboratorio tiene como requisito primordial haber recibido la capacitación en el software InControl o tener algún conocimiento sobre el manejo de este mismo.

Para un buen desarrollo de este laboratorio el estudiante debe haber leído el Capítulo “**MANEJO Y ORGANIZACIÓN DE PROYECTOS**” del manual de Usuario InControl, además debe conocer los tipos de programas, funciones, bloques de funciones y macros que conforman un proyecto, así como de manejar los tipos de lenguajes que soporta el InControl.

DESARROLLO:

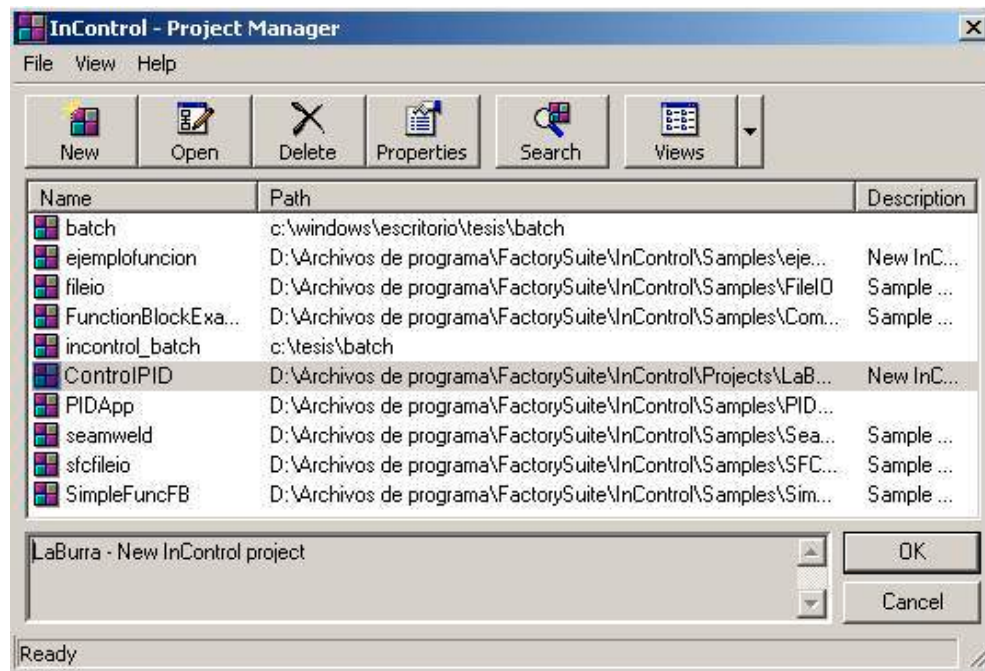
1. Después de iniciar el InControl:
2. En el menú **File** del cuadro de diálogo **InControl Project Manager**, se hace clic en **New**. Aparece el cuadro de diálogo **Create InControl Project** (Figura 1).

Figura 1. Creación de proyectos InControl



3. Se ingresa el nombre del proyecto (**Control PID**). Se escribe el nombre del proyecto usando las convenciones para nombres de Windows NT, se selecciona una ruta, y se pulsa **OK**. Un directorio con este nombre se crea en el disco duro y el nuevo proyecto se adiciona a la lista de proyectos de InControl. En la figura 43 de la página siguiente, el proyecto **Control PID** es creado.
4. El proyecto que se crea aparece en la ventana de proyectos del ambiente de desarrollo. Después de creado el proyecto, se inicia la creación de los programas necesarios para la ejecución de la estrategia de control PID.

Figura 2. InControl Project Manager



Cómo Remover Un Proyecto

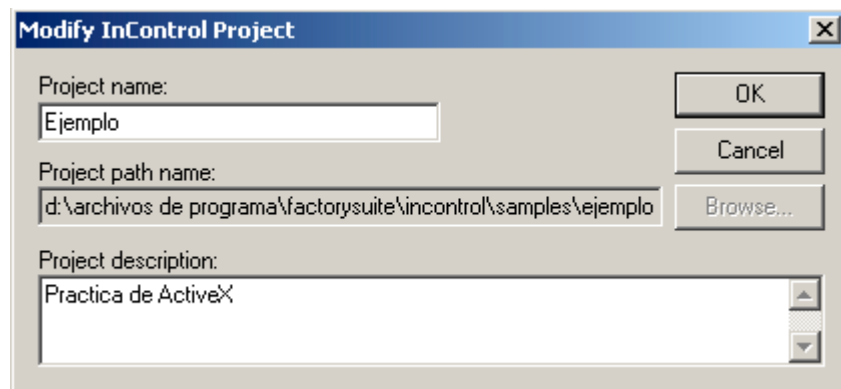
Para remover un proyecto, debe mostrar el **Project Manager**.

➤ Para remover o borrar un proyecto:

1. Seleccione el proyecto desde el **Project Manager**, y sobre el menú **File** pulse en **Delete**. Cuando aparezca el cuadro de dialogo **Delete** y le indique para confirmar pulse **OK**.

Modificación del Nombre y la Descripción de un Proyecto

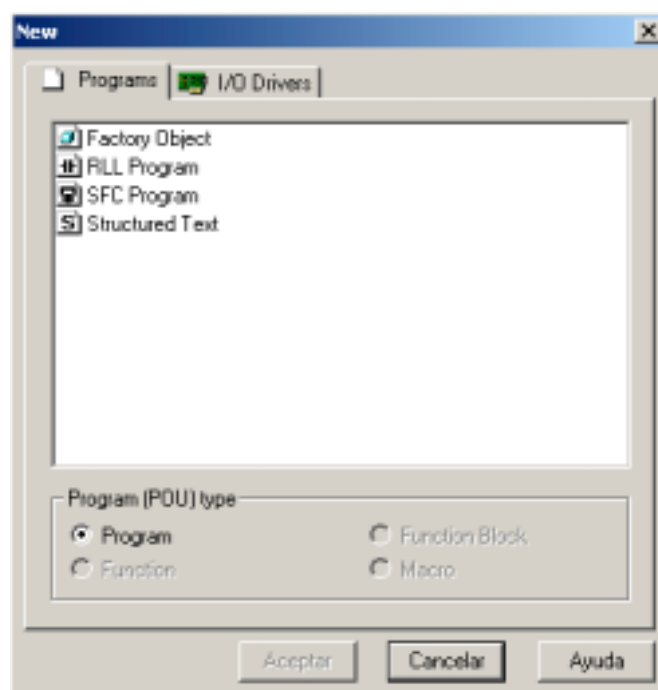
1. Seleccione el proyecto y sobre el menú **File** del **Project Manager** pulse en **Properties**. Aparecerá el cuadro de diálogo **Modify InControl Project**.



2. Ingrese la nueva descripción o el nombre del proyecto, entonces pulse **OK**. La nueva descripción aparecerá solamente en rojo en el campo **Description** del cuadro de dialogo **InControl Projects**.

➤ Para agregar una POU al proyecto:

1. En el menú **File**, pulse en **New**.



2. Seleccione la etiqueta de programas en el cuadro de dialogo **New**.
3. Seleccione el tipo de programa (Objetos de Fabrica, RLL, SFC, Texto Estructurado), y el tipo de POU(Programa, Función, Bloque de Función, Macro). Luego pulse **Aceptar**.
4. Escoja un nombre no mayor de 31 caracteres, dentro del cuadro de **Guardar como** que aparece.

EVALUACIÓN:

Al finalizar este laboratorio el estudiante estará en capacidad de crear un proyecto y realizar las tareas asociadas a este como son adición y eliminación de programas, funciones, bloques de funciones y macros, y modificación en las características de un proyecto.

LABORATORIO #2

CREACIÓN DE UNA VARIABLE

OBJETIVOS.

- Crear una variable en un programa y establecer su alcance.
- Identificar los tipos de variables que son soportadas por el InControl, así como su funcionalidad en un programa.
- Definir todos los tipos de variables globales y locales de InControl, incluyendo los tipos de datos definidos por el usuario

PRELIMINARES:

El estudiante o la persona que va a realizar este laboratorio tiene como requisito primordial haber recibido la capacitación en el software InControl o tener algún conocimiento sobre el manejo de este mismo.

Para un buen desarrollo de este laboratorio el estudiante debe haber leído el Capítulo “**DEFINICIÓN DE VARIABLES**” del manual de Usuario InControl, además debe conocer los tipos de variables, su alcance, y como poder definir las, así como manejar los tipos de lenguajes que soporta el InControl.

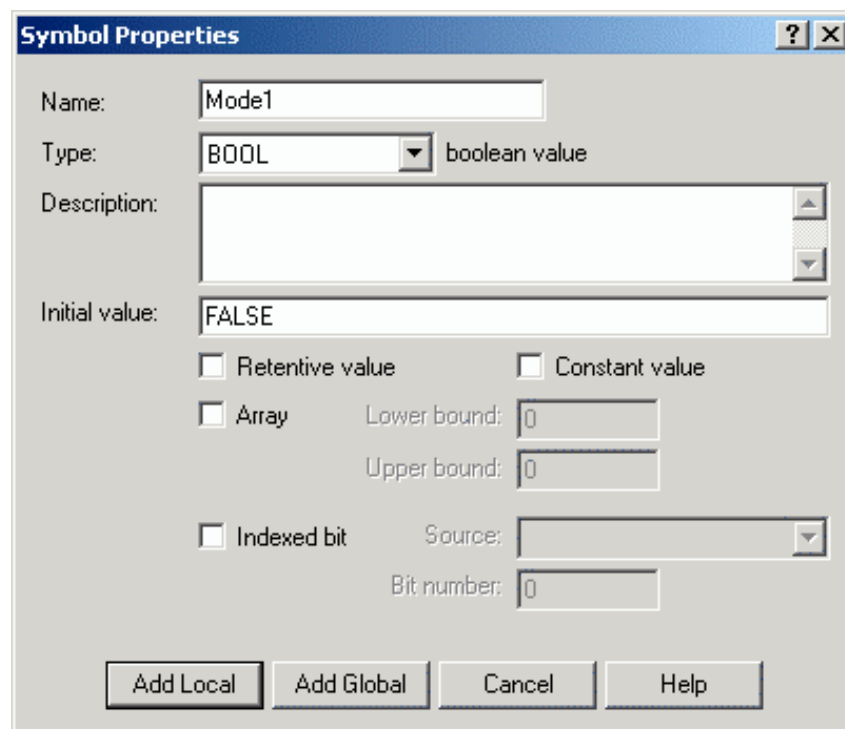
DESARROLLO:

➤ Para crear una variable de programa:

1. Entre al **Symbol Manager**.
2. Seleccione el alcance apropiado. Por ejemplo, global, local (un programa individual), funciones de bloques, etc.
3. Haga clic en **New** en la barra de herramientas del Administrador de Símbolos.



Aparecerá el cuadro de dialogo **Symbol Properties** mostrando las mismas propiedades así como para las variables anteriores de la lista.

A screenshot of the 'Symbol Properties' dialog box. The dialog has a title bar with a question mark and a close button. The fields are: Name: 'Mode1'; Type: 'BOOL' (dropdown) with 'boolean value' to its right; Description: an empty text area; Initial value: 'FALSE'. Below these are several options: 'Retentive value' (checkbox), 'Constant value' (checkbox), 'Array' (checkbox) with 'Lower bound: 0' and 'Upper bound: 0' input fields, and 'Indexed bit' (checkbox) with 'Source:' dropdown and 'Bit number: 0' input field. At the bottom are four buttons: 'Add Local', 'Add Global', 'Cancel', and 'Help'.

4. Ingrese el nombre de la variable dentro del campo **Name**. Use solamente caracteres alfanuméricos y subrayados. La longitud máxima del nombre de la variable es de 100 caracteres.

Las variables locales deben ser únicas. Esto es, dos variables con igual nombre no pueden estar en el mismo programa. Además, una variable local no puede tener el mismo nombre que una variable global.

5. Seleccione el tipo de dato en el campo **Type**.
6. Ingrese la descripción opcional dentro del campo **Description**.
7. Si la variable requiere un valor inicial, ingrese el valor dentro del campo **Inicial Value**.
8. Revise el cuadro de comprobación **Constant Value** si usted quiere que el valor para esta variable permanezca constante. Este seguro de ingresar el valor en el campo **Initial Value**.
9. Haga clic en el cuadro de comprobación **Retentive Value** para que la variable retenga este valor en caso de una falla de potencia.

El InControl periódicamente guarda variables retenidas y forzadas en el disco duro. La frecuencia de cero deshabilita esta característica. Este intervalo es configurable y usted puede ajustarlo en el cuadro de dialogo “**Runtime Engine Properties**”, descrito en “**Ajuste del tiempo de exploración**” en el capítulo “**ADMINISTRACIÓN DE SISTEMAS INCONTROL**”.

10. Haga clic en **Add Local** o **Add Global**. La nueva variable aparecerá en el campo **Symbol List**.

➤ **Para remover una variable existente:**

1. Haga clic en el nombre de la variable.
2. Haga clic en **Delete** en la barra de tareas del Administrador de Símbolos.



Usted también puede utilizar la tecla **Del** en el teclado.

EVALUACIÓN:

Al finalizar este laboratorio el estudiante estará en capacidad de crear y diferenciar los tipos de variables que se usan en un programa, así como definir su alcance y su funcionalidad en un proyecto. Además el estudiante será capaz de crear arreglos de variables y tipos de variables diseñadas para un uso específico en un programa.

LABORATORIO #3

CREACIÓN DE UN PROGRAMA RLL

OBJETIVOS.

- Aprender el funcionamiento del editor **RLL** (Lógica en Escalera por relevos) y identificar como poder usarlo para crear nuevos programas y para adicionar elementos (contactos, bobinas, bobinas de salto, bloques de funciones, etc.) a estos.
- Crear programas de aplicación para sus proyectos futuros basados en ordenes y en la lógica RLL , y saber definir que tipos de ordenes debe realizar en su programa.

PRELIMINARES:

El estudiante o la persona que va a realizar este laboratorio tiene como requisito primordial haber recibido la capacitación en el software InControl o tener algún conocimiento sobre el manejo de este mismo.

Para un buen desarrollo de este laboratorio el estudiante o la persona debe haber leído los capítulos “**ELEMENTOS DE PROGRAMACIÓN RLL**” y “**USO DEL EDITOR RLL**” del Manual de Usuario InControl, además debe conocer los diferentes de lenguajes de programación de PLC que existen, sus codigos, y como

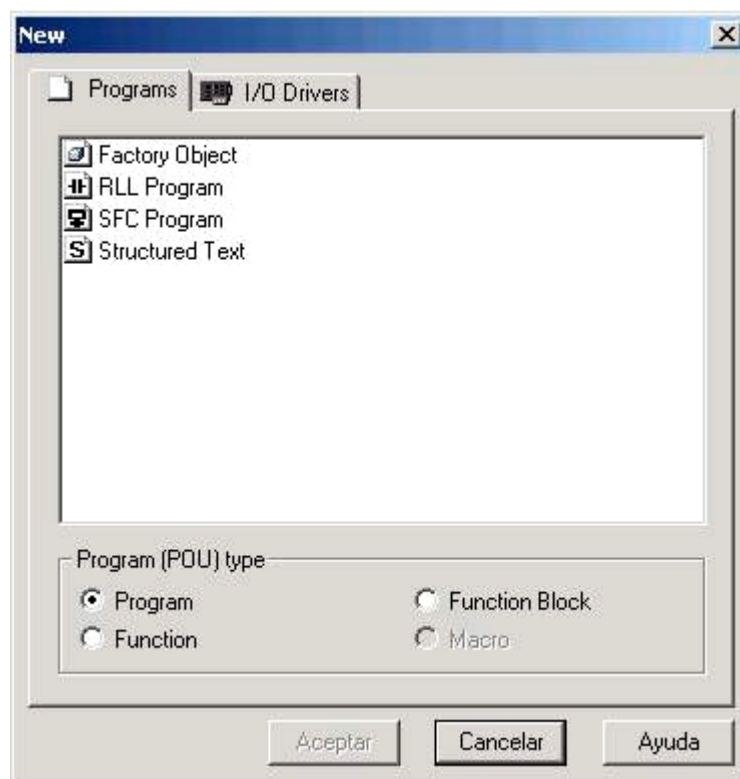
poder aplicarlos a un programa de control, así como manejar los tipos de lenguajes que soporta el InControl.

DESARROLLO:

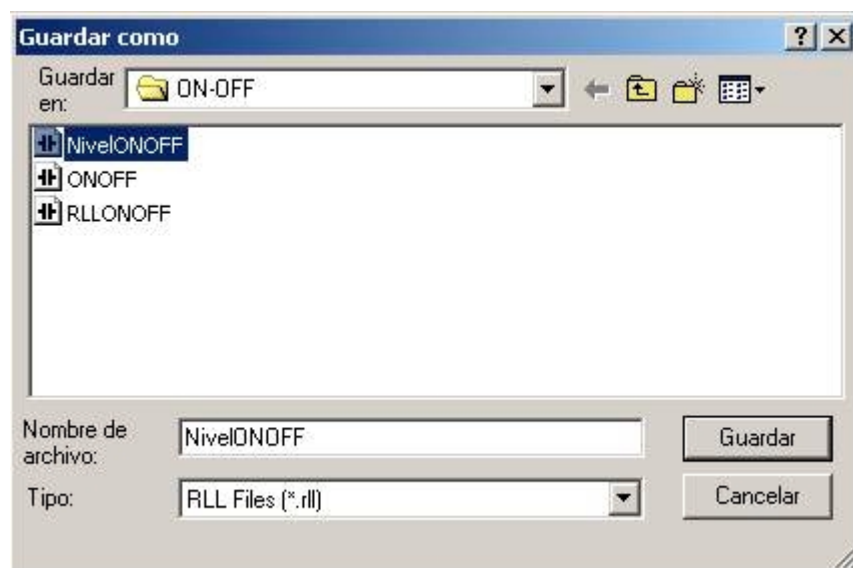
Para el desarrollo de esta practica tomaremos como ejemplo el desarrollo del programa encargado de la estrategia de control ON-OFF en el proyecto (**Control ON-OFF**) de nuestra aplicación, esto se hace, haciendo doble clic sobre el nombre del proyecto **Control ON-OFF** en el **Project Manager**, para abrirlo en el ambiente de desarrollo:

1. Una vez en el ambiente desarrollo, sobre el menú **File**, se pulsa en **New**.

9Aparecerá el cuadro de diálogo **New**.



2. Se selecciona la etiqueta **Programs** en el cuadro de dialogo **New**.
3. Se selecciona el tipo de programa **RLL Program** (Programas RLL). Después de escogida esta opción, se pulsa **Aceptar**.
4. Luego aparece el cuadro de diálogo **Guardar como**, donde se ingresara el nombre del programa RLL (**NivelONOFF**).



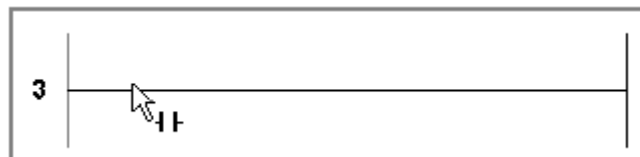
El editor RLL muestra un nuevo archivo RLL con dos barras de energía y un escalón vacío. Lo cual indica que todo está listo para editar el programa. En la **Project View** en la carpeta **Programs** aparece el nuevo programa **NivelONOFF**. A continuación se describen los pasos necesarios para la adición de los elementos RLL usados en nuestro programa **NivelONOFF**.

Para adicionar un contacto al escalón:

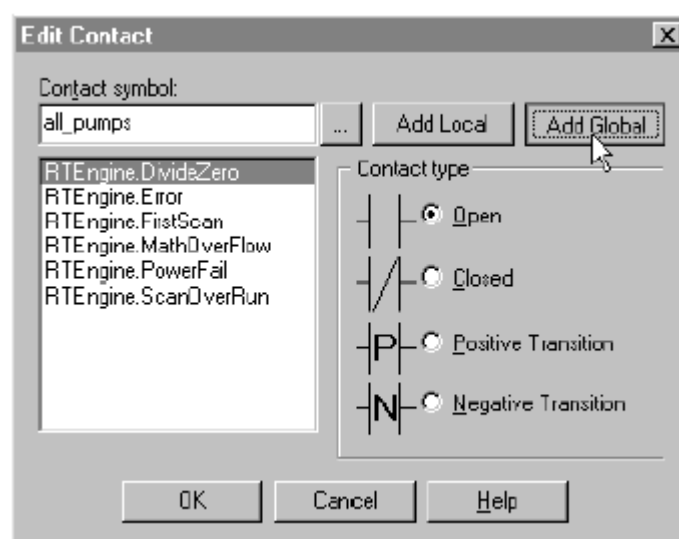
1. Pulse **Contact Tool** sobre la barra de herramientas RLL.



2. Note que la barra de herramientas debe ser acomodada en orientación vertical.
3. Mueva el cursor sobre el escalón.



4. Pulse el botón izquierdo del mouse. Aparecerá el cuadro de diálogo **Edit Contact**.
5. Ingrese un nombre en el campo **Contact Symbol**. Solamente caracteres alfanuméricos y caracteres subrayados son válidos. No use espacios y no empiece el nombre con un número. Nombres de símbolos no son casos sensibles. En la figura, **all_pumps** es la nueva variable.



6. Pulse **Add Global** para adicionar el nuevo nombre de la variable a el Symbol Manager como una variable Global. Este asigna una localización de memoria interna para representar el nuevo contacto.
7. Pulse **Open** para especificar el tipo de contacto. Luego pulse **OK**. El nuevo contacto parece sobre el escalón en la localización que usted especificó.

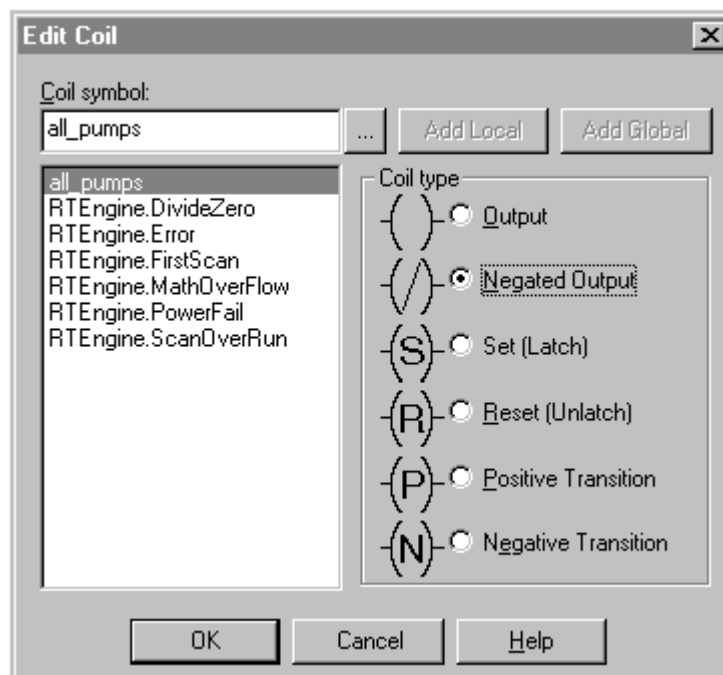
Para adicionar una bobina sobre el escalón:

1. Pulse **Coil Tool** sobre la barra de herramientas RLL.



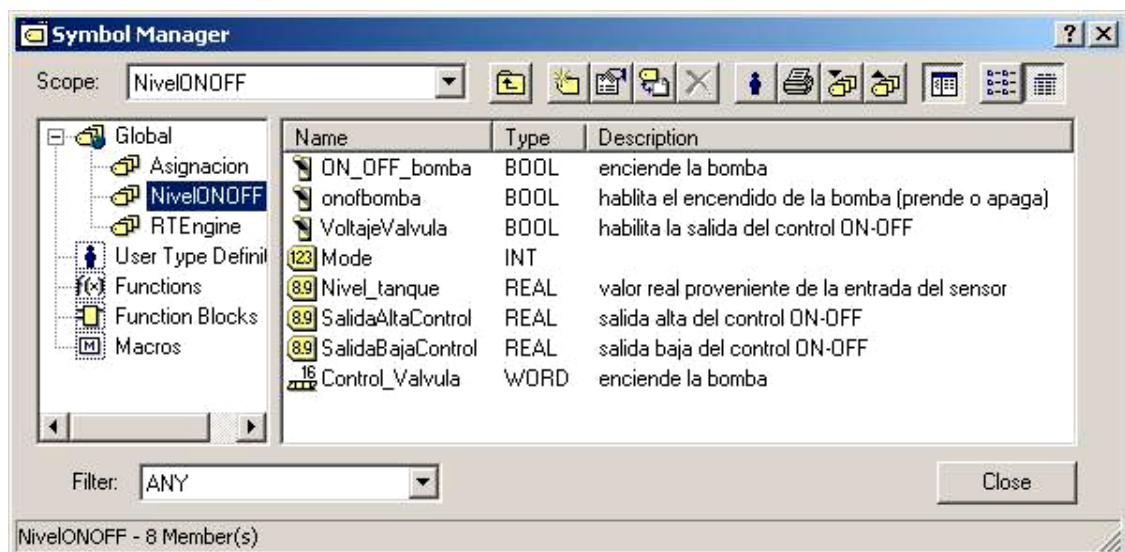
2. Mueva el cursor sobre la localización del escalón.
3. Pulse el botón izquierdo del mouse. La caja de diálogo **Edit Coil** aparece.

Note que la variable definida para el contacto aparece en la lista de símbolos.



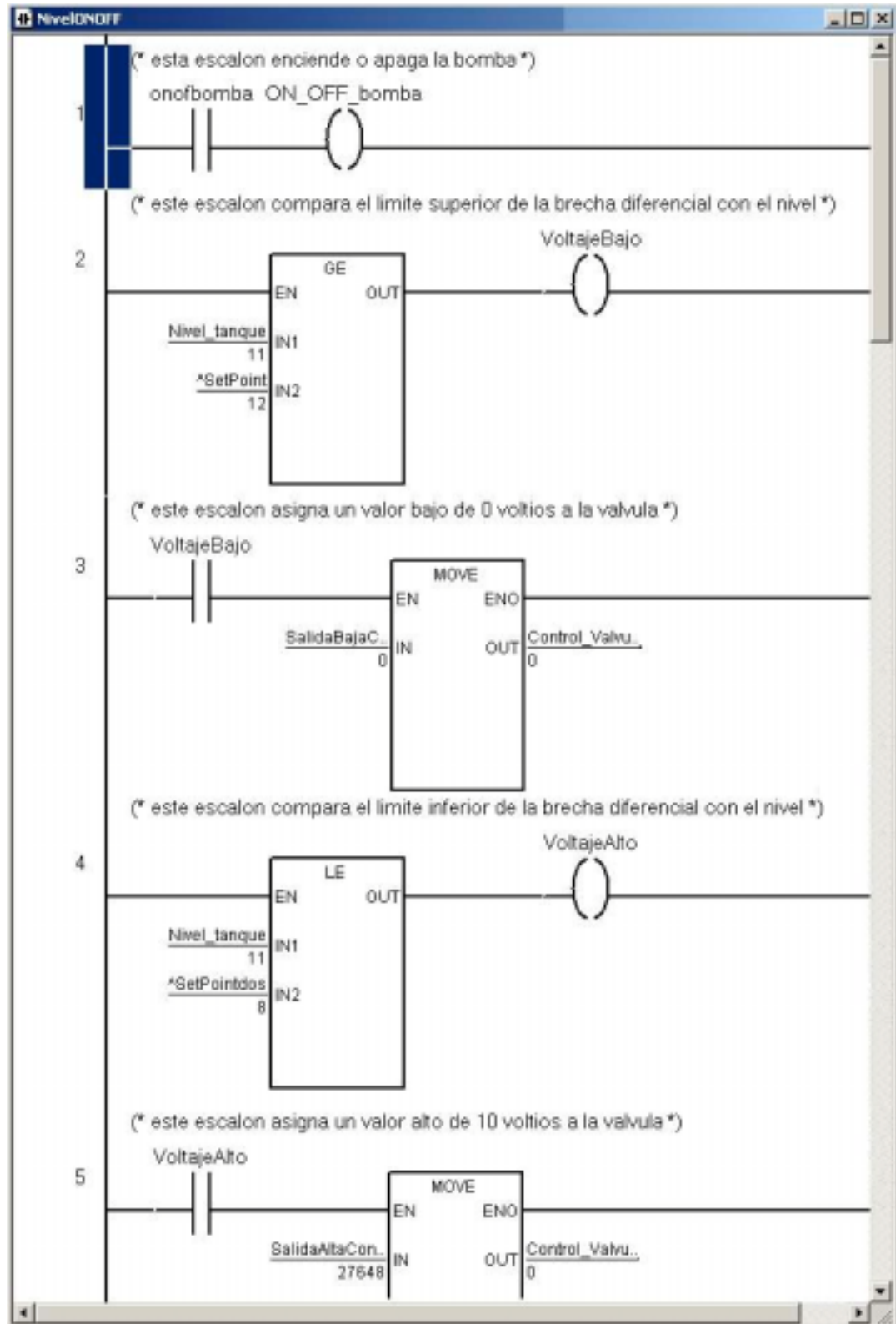
4. Debido a que el ejemplo usa la misma variable para el contacto y la nueva bobina, no es necesario adicionar otra variable al Symbol Manager. Seleccione **all_pumps** como el nombre de variable para la bobina.
5. Pulse **Negated Output** para especificar el tipo de bobina. Luego pulse OK. La nueva bobina, Negated Output Coil, aparece sobre el escalón.

El programa a continuación (Figura 3) será el encargado de efectuar la estrategia de control **ON-OFF** y además de encender y apagar la bomba, este es creado en el lenguaje de programación RLL (Relay Ladder Logic) que posee el InControl. El primer paso es definir las variables necesarias para la ejecución del programa, en el Symbol Manager, entre estas tenemos:



Y esta compuesto de las siguientes órdenes lógicas, representadas por medio de contactos y bobinas.

Figura 3. Presentación del programa **NivelONOFF**.



EVALUACIÓN:

Al finalizar este laboratorio usted estará en capacidad de desarrollar cualquier estrategia de control basado en el lenguaje RLL, así como el funcionamiento de la lógica de bobinas y contactos y funciones de bloques, que le permitirán representar y simular sus procesos tal como en la realidad. Además estará en capacidad de emitir conceptos acerca del tipo de programación a realizar en su proyecto de aplicación, es decir que tipo de elemento RLL ha de utilizar en su programa.

LABORATORIO #4

ADICIÓN, CONFIGURACIÓN Y EJECUCIÓN DEL FOE PID

OBJETIVOS.

- Aprender el funcionamiento del editor de objetos de fabrica InControl (FOE) y identificar como poder usarlo para crear nuevos programas de aplicación y como adicionarlos a la estrategia de control en el proyecto que usted desarrolla.
- Saber definir que tipos de parámetros son necesarios para el correcto desempeño del FOE PID en su proyecto y como debe configurarlo para su ejecución.

PRELIMINARES:

El estudiante o la persona que va a realizar este laboratorio tiene como requisito primordial haber recibido la capacitación en el software InControl o tener algún conocimiento sobre el manejo de este mismo.

Para un buen desarrollo de este laboratorio el estudiante la persona debe haber leído los capítulos “**USO DEL EDITOR DE OBJETOS DE FABRICA INCONTROL**” del Manual de Usuario InControl, además debe conocer los diferentes de lenguajes de programación de PLC que existen, sus códigos, y como poder aplicarlos a un programa de control, así como de saber manejar los tipos de lenguajes que soporta el InControl.

DESARROLLO:

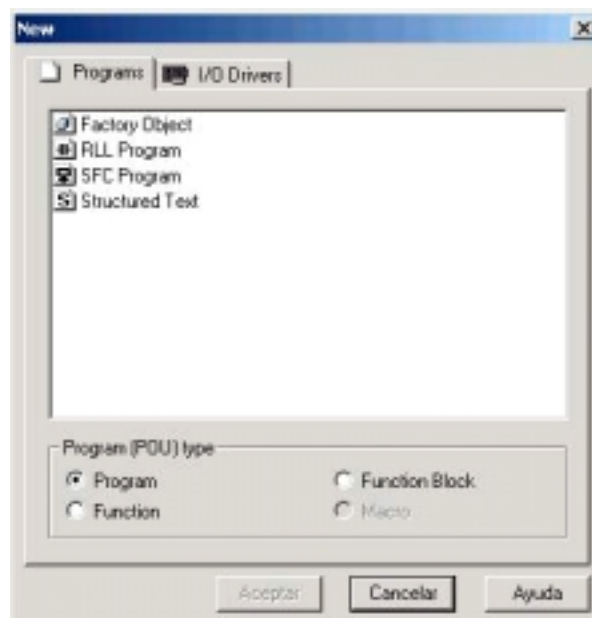
Para el desarrollo de esta practica tomaremos como ejemplo la implementación de la estrategia de control PID desarrollada para nuestro proyecto (**Control ON-OFF**) para lo cual se utilizara el regulador PID digital (**FOE PID**) que viene incluido como una de las herramientas internas de trabajo del InControl.

Lo primero es anexar el FOE PID al proyecto ya creado (**Control PID**), esto se hace, haciendo doble clic sobre el nombre del proyecto (**Control PID**) en el **Project Manager**, para abrirlo en el ambiente de desarrollo:

1. Una vez en el ambiente desarrollo, sobre el menú **File**, se hace clic en **New**.

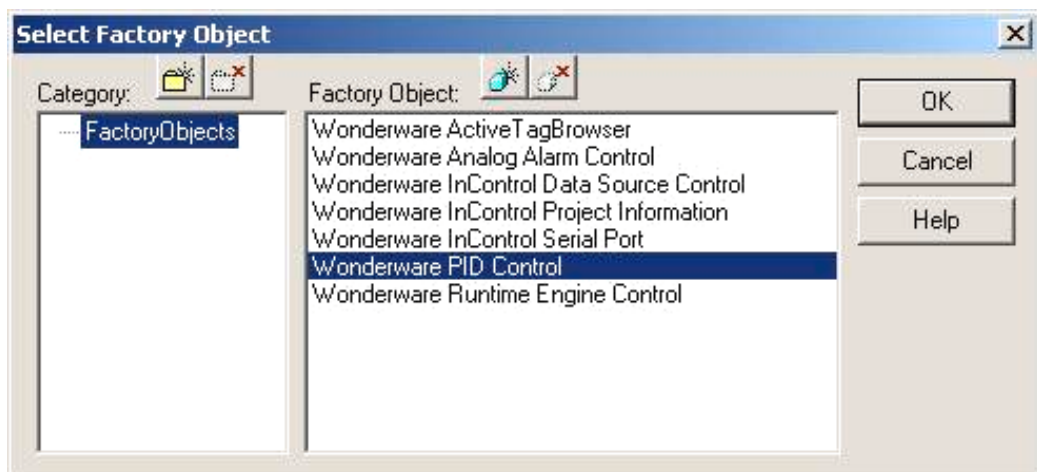
Aparecerá el cuadro de diálogo **New** (Figura 1).

Figura 1. Cuadro New.



2. Se selecciona la etiqueta **Programs** en el cuadro de diálogo **New**.
3. Se selecciona el tipo de programa **Factory Object** (Objetos de Fabrica). Después de escogida esta opción, se pulsa **Aceptar**.
4. Seguido a esto aparecerá el cuadro de dialogo **Select Factory Object** (figura 2), donde se escogerá el regulador digital (FOE) **Wonderware PID Control** y se pulsa **OK**.

Figura 2. Cuadro Select Factory Object.



5. Luego se ingresa el nombre del FOE (**NiveIPID**) en el cuadro de diálogo **Guardar como**.

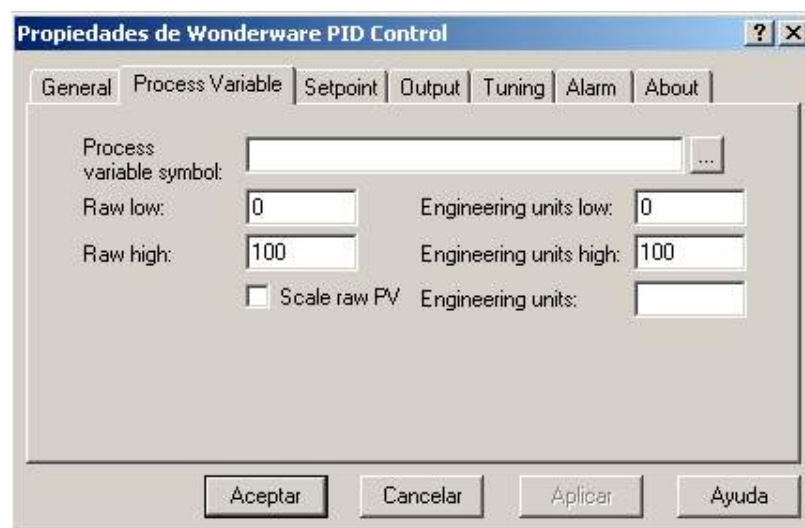
Seguido a estos pasos en la **Project View** aparecerá el nuevo programa (FOE) **NiveIPID**, en la carpeta **Programs**. Después de esto se configuraran los parámetros del FOE (**NiveIPID**) como son tiempo de muestreo, limite y unidades de la variable de proceso, valores mínimos y máximos del setpoint, la variable de salida y los valores de sintonización (Proporcional, integral y derivativo), en el cuadro de diálogo **Propiedades de Wonderware PID Control** (Figura 3), el cual se habilita haciendo doble clic sobre la ventana del FOE.

Figura 3. Propiedades de Wonderware PID Control



Se fija un tiempo de muestreo de 0.1 segundo para el tiempo de muestreo del PID. Para la variable de proceso (figura 4) se escoge el símbolo que representa el nivel real del tanque “**Nivel_tanque**” suministrado por el PLC a través del puerto de entrada análoga proveniente del transmisor de nivel y los valores limites para este dato.

Figura 4. Propiedades de Wonderware PID Control etiqueta variable de proceso



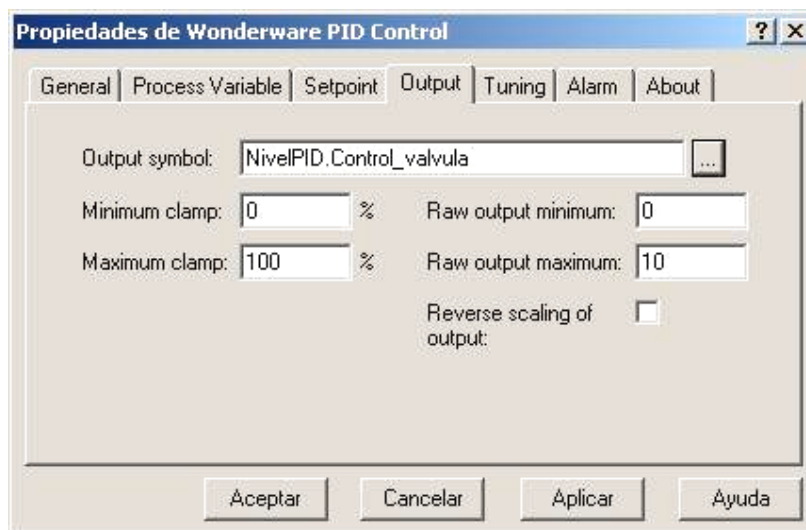
En la etiqueta **Setpoint** (Figura 5) se asigna el símbolo que representa este dato en el programa “**NivelPID.Sp**” y sus límites (valor máximo y valor mínimo).

Figura 5. Propiedades de Wonderware PID Control etiqueta Setpoint



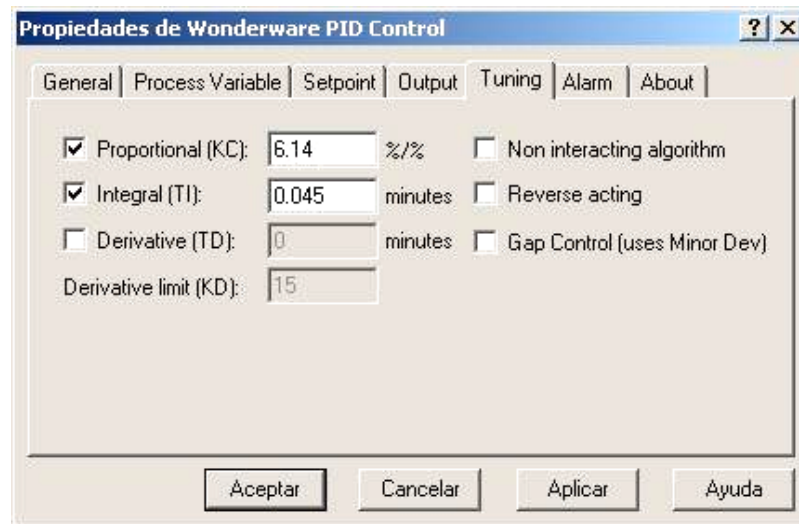
En la etiqueta **Output** (figura 6) correspondiente a la salida del PID se asigna el símbolo que representa este valor en el programa “**NivelPID.Control_valvula**” y sus límites (valor máximo y valor mínimo).

Figura 6. Propiedades de Wonderware PID Control etiqueta Output



En la etiqueta **Tuning** (Figura 7) correspondiente a los parámetros de sintonización del PID se asigna los valores de los parámetros (K_c , T_i , T_d y K_d).

Figura 7. Propiedades de Wonderware PID Control etiqueta Tuning



A continuación se muestra el FOE PID (Figura 8) después de ser configurado y en plena ejecución para un setpoint (línea blanca) de 65 cm y un valor de nivel (línea verde) de 20 cm y valor de voltaje (línea azul) de salida de 10 voltios.

Figura 8. Ventana del FOE PID



Después de configurar el FOE **NiveIPID**, estará listo para la ejecución en nuestro proyecto **ControlPID** o para ejecutarlo en cualquiera estrategia de control.

EVALUACIÓN:

Al finalizar este laboratorio usted estará en capacidad de configurar cualquiera de los FOE's que trae incluido el InControl para poder utilizarlos en sus estrategias de control y en cualquier proyecto que usted crea necesario de utilizarlos por la importancia de estos para el desarrollo de sus aplicaciones de control industrial o automatización.

LABORATORIO #5

CREACIÓN DE UNA FUNCIÓN DE BLOQUE

OBJETIVOS.

- Identificar cuales son los pasos y requisitos necesarios para la creación de una función de bloque en el software InControl.
- Crear los tipos de variables necesarias para la ejecución de un programa y establecer su alcance en la función de bloque.
- Identificar los tipos de variables que son soportadas por el InControl, así como su funcionalidad en un programa.
- Definir todos los tipos de variables globales y locales de InControl, incluyendo los tipos de datos definidos por el usuario.
- Definir que tipo de programa es necesario para la realización de la de un código de programa en una función de bloque.

PRELIMINARES:

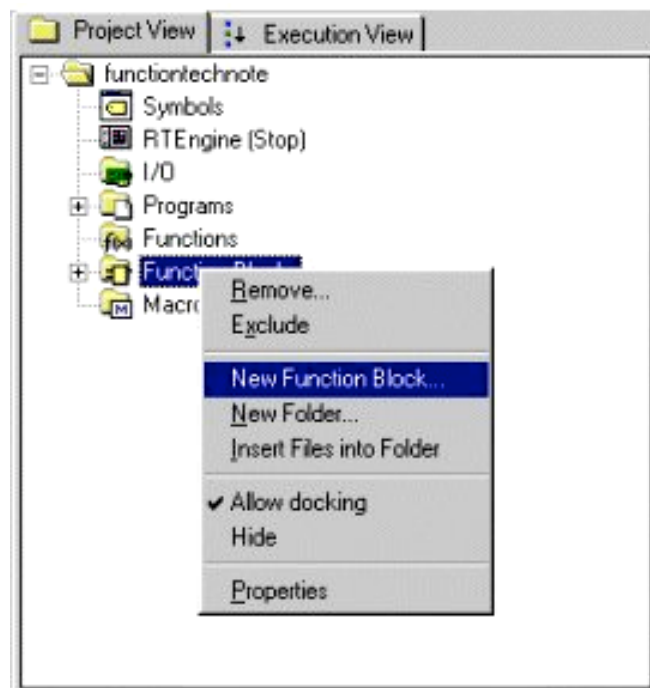
El estudiante o la persona que va a realizar este laboratorio tiene como requisito primordial haber recibido la capacitación en el software InControl o tener algún conocimiento sobre el manejo de este mismo.

Para un buen desarrollo de este laboratorio el estudiante o la persona debe haber leído los Capítulos 5 **“DEFINICIÓN DE VARIABLES”** y **“MANEJO Y ORGANIZACIÓN DE PROYECTOS”** del manual de Usuario InControl, Además para un buen desarrollo de este laboratorio el estudiante o a la persona debe conocer los tipos de variables, su alcance, y como poder definir las, así como de manejar los tipos de lenguajes que soporta el InControl, además debe conocer los tipos de programas, funciones, bloques de funciones y macros que conforman un proyecto, así como de saber manejar los tipos de lenguajes que soporta el InControl.

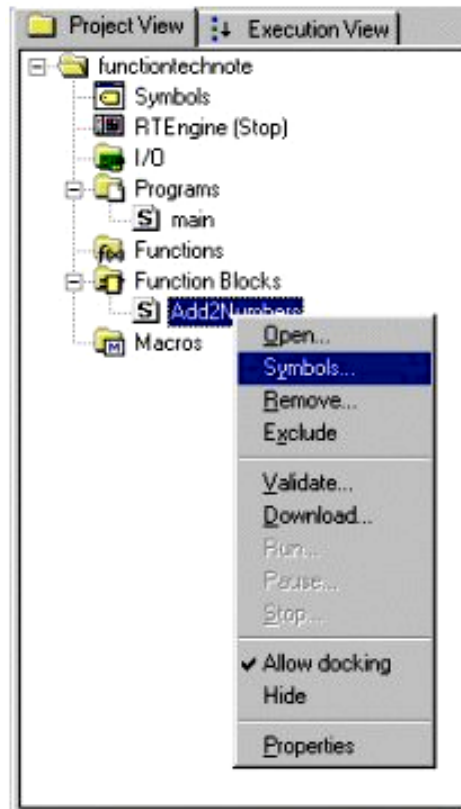
DESARROLLO:

En este laboratorio se evaluará el resto de laboratorios realizados, ya que es necesario su realización para posteriormente poder realizar esta práctica, es decir si aun no ha realizado las demás prácticas no podrá realizar un buen desempeño de esta práctica de laboratorio ya que en esta se pone a prueba lo aprendido con las demás.

1. Abra el InControl Project Manager y cree un nuevo proyecto. Abra el nuevo proyecto en el ambiente de desarrollo de InControl.
2. Sobre la etiqueta **Project View**, haga clic derecho en **Function Blocks**, y seleccione una en **New Function Block**. Cree una nueva función de bloque e ingrésele por nombre **Add2Numbers**.

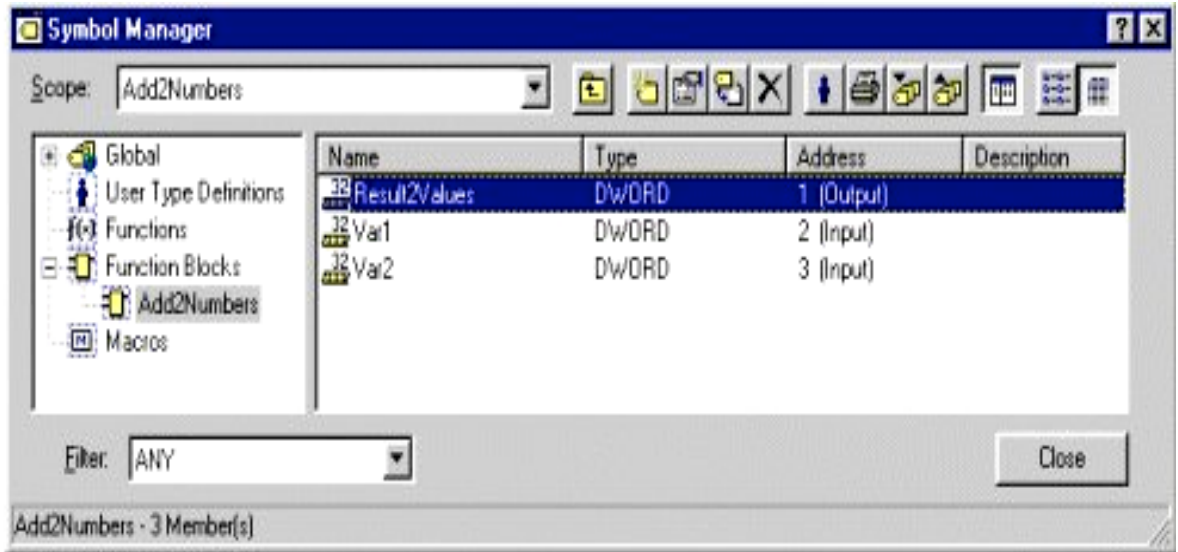


3. Después sobre la etiqueta **Project View**, haga clic derecho sobre **Add2Numbers**, y seleccione en **Symbols**.

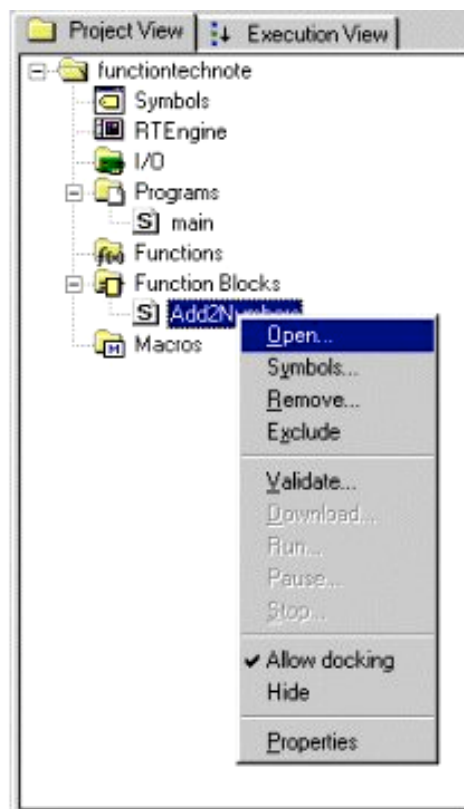


4. Para este ejemplo adicione los siguientes símbolos locales, como se muestra en la figura 3. cuando finalice haga clic en el botón **Close**.

<u>Nombre</u>	<u>Tipo</u>	<u>Dirección</u>
Result2Values	DWORD	1 (Salida)
Var1	DWORD	2 (Entrada)
Var2	DWORD	3 (Entrada)

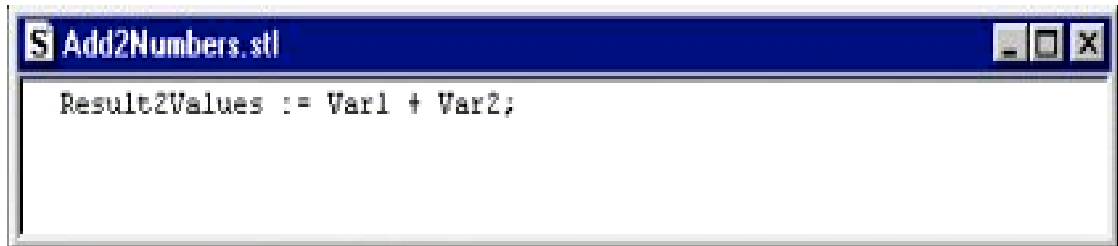


- De nuevo sobre la etiqueta **Project View** haga clic derecho en **Add2Numbers** y seleccione en **Open**.



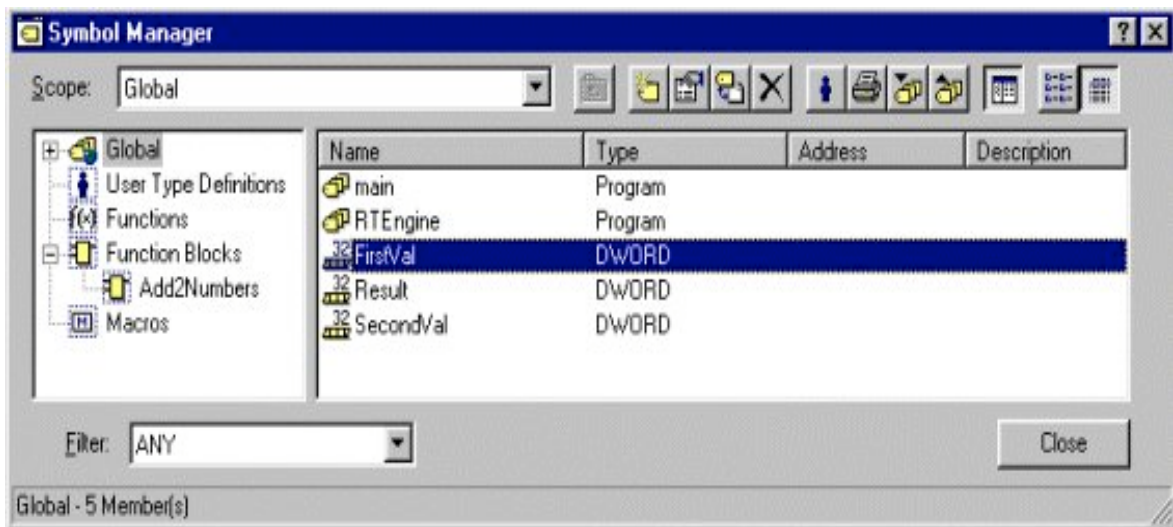
6. Cree el siguiente programa en texto estructurado para la función de bloque:

```
Result2Values := Var1 + Var2;
```

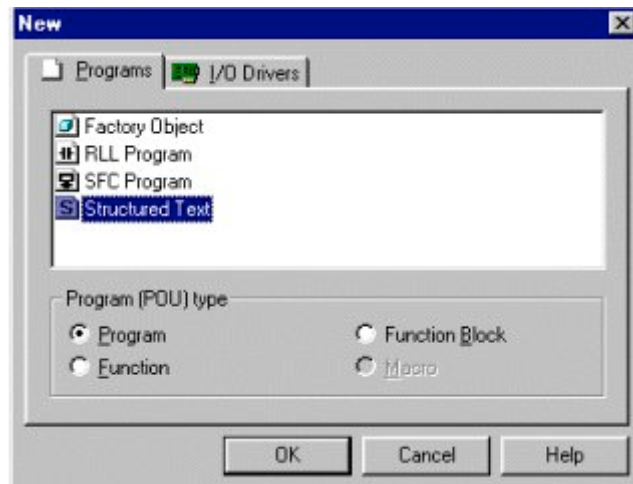


7. Use el **Symbol Manager**, para adicionar los siguientes símbolos globales como se muestra en la siguiente figura. Cuando finalice pulse el botón **Close**.

<u>Name</u>	<u>Type</u>
Result	DWORD
FirstVal	DWORD
SecondVal	DWORD

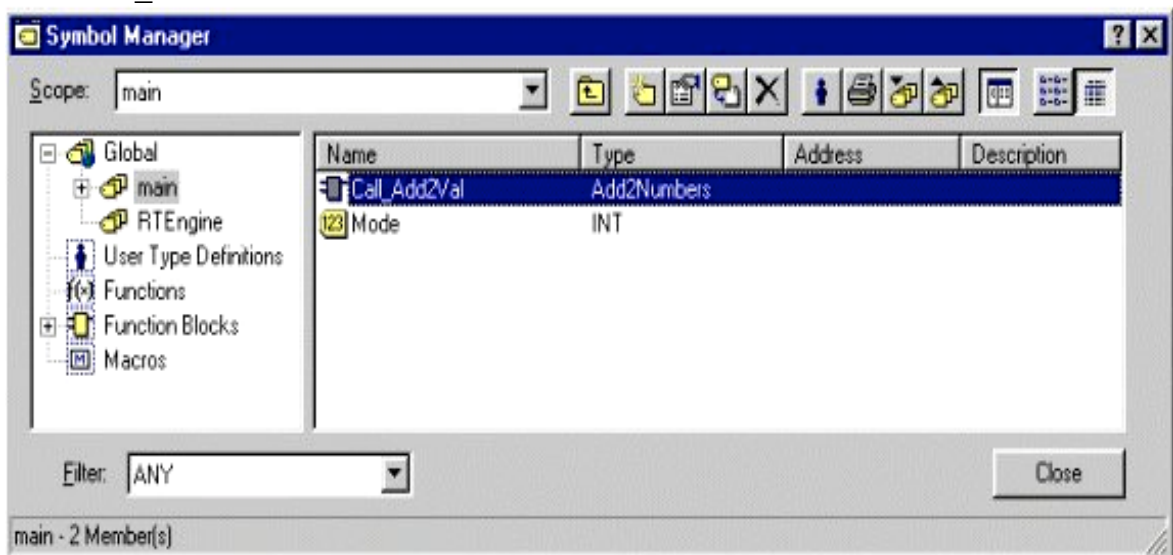


8. Desde la barra de menú seleccione en **File/New** y cree un nuevo programa de texto estructurado llamado **Main**.

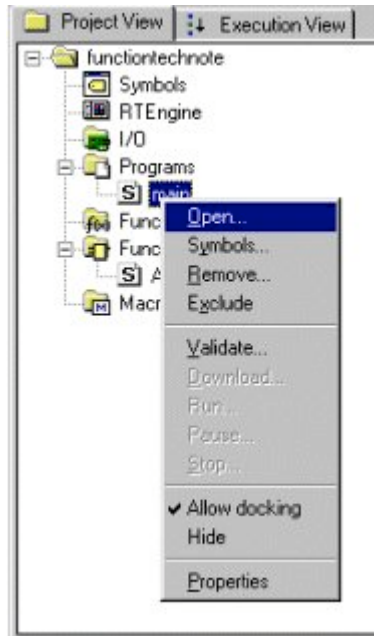


9. Use el **Symbol Manager**, para crear un tipo de función de bloque para adicionar un símbolo local al programa **Main** que se define as continuación cuando finalice pulse el botón **Close**.

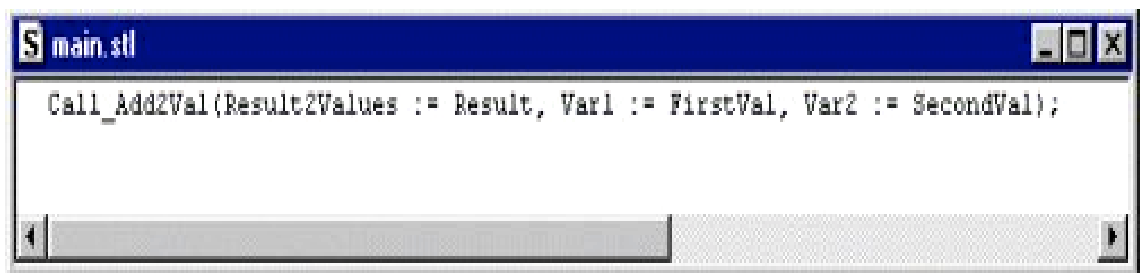
Nombre Tipo
Call_Add2Val Add2Numbers



11. Sobre la etiqueta **Project View/Programs**, haga clic derecho en la etiqueta **Main**, y seleccione en **Open**.

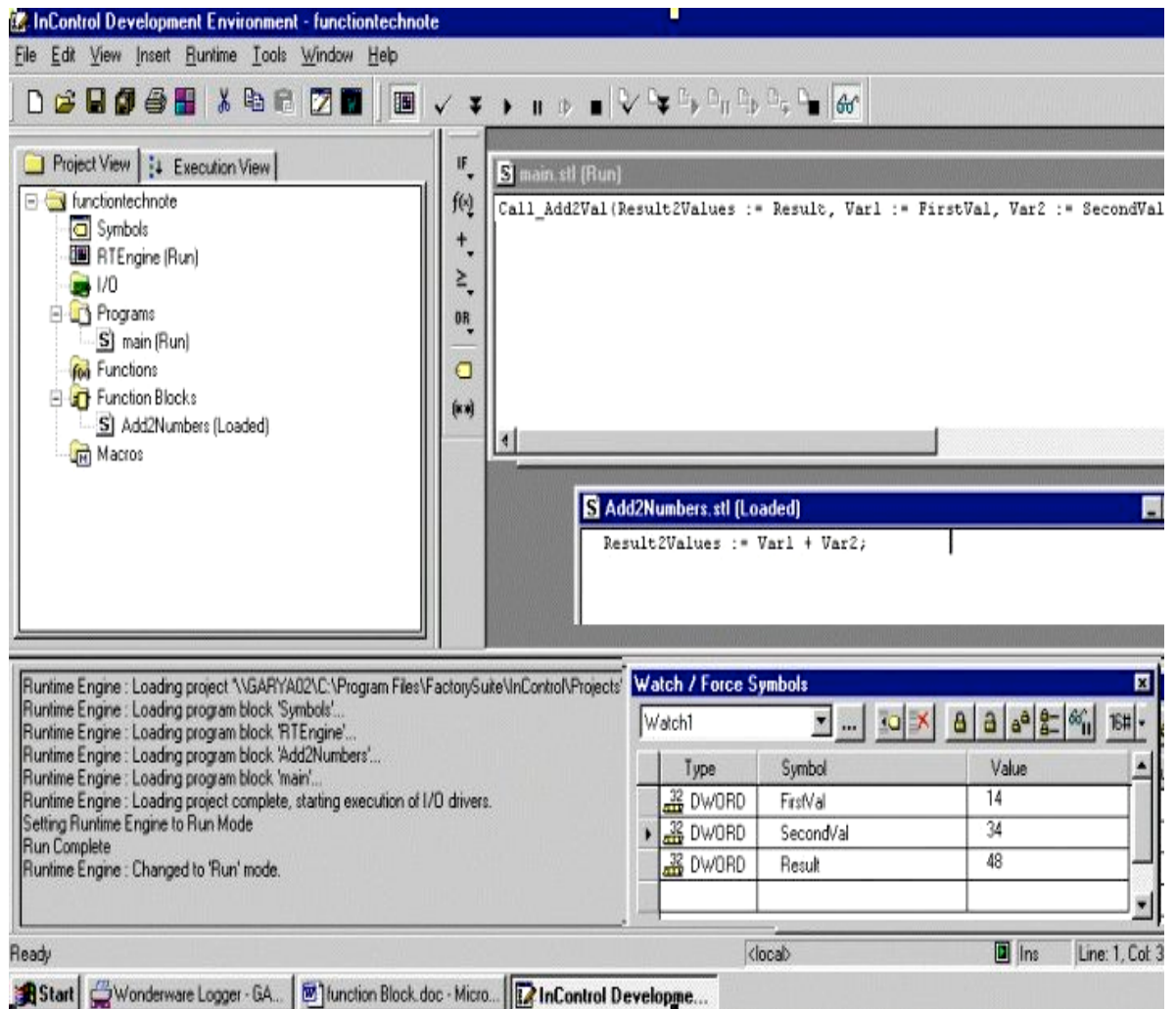


11. Cree el siguiente programa de texto estructurado para el programa **Main**:
Call_Add2Val(Result2Values:=Result,Var1:=FirstVal, Var2:=SecondVal);



12. Corra la aplicación del InControl e ingrese los valores para **FirstVal** y **SecondVal** usando la **Watch Window**.

El símbolo **Result** contendrá la suma de los símbolos **FirstVal** y **SecondVal**.



EVALUACIÓN:

Al finalizar este laboratorio el estudiante estará en capacidad de crear y diferenciar los tipos de variables que se usan en un programa, así como definir su alcance y su funcionalidad en un proyecto y como poder definir las para su uso en una función de bloque cualquiera. Además el estudiante será capaz de crear arreglos de variables y tipos de variables diseñadas para un uso específico en un programa. Además el estudiante estará en capacidad de crear un proyecto y realizar las tareas asociadas a este como son adición y eliminación de programas, funciones, bloques de funciones y macros, y modificación en las características de un proyecto.

Hardware Manual

SMS-CIF30-DPM
SMS-CIF30-DPS
SMS-CIF30-PB
SMS-CIF30-DNM
SMS-CIF30-COM
SMS-CIF30-SDSM
SMS-CIF30-IBM
SMS-CIF30-IBS
SMS-CIF30-CNS

PC/ISA Communications

Lantronix
1841 Centre Point Circle, Suite 143
Naperville, Illinois 60563 USA
PHONE: +1 630/245-1770
FAX: +1 630/245-1717
EMAIL: sales@synergetic.com
WEB: www.synergetic.com

Table of Contents

INTRODUCTION	3
HARDWARE DESCRIPTION	4
CONFIGURATION / JUMPERS	5
DISPLAY / LEDS	7
FIELDBUS CONNECTION	8
PROFIBUS INTERFACES (SMS-CIF30-DPM, SMS-CIF30-PB, SMS-CIF30-DPS).....	8
DEVICENET INTERFACE (SMS-CIF30-DNM).....	9
SDS INTERFACE (SMS-CIF30-SDSM).....	10
CANOPEN INTERFACE (SMS-CIF30-COM).....	11
INTERBUS INTERFACE (SMS-CIF30-IBM,SMS-CIF30-IBS).....	12
CONTROLNET INTERFACE (SMS-CIF30-CNS).....	13
DIAGNOSTIC RS232 CONNECTION	14
NETWORK CONFIGURATION	15
FIRST CHECK OUT THE CARD.....	15
INSTALL THE SOFTWARE.....	15
FIRMWARE.....	15

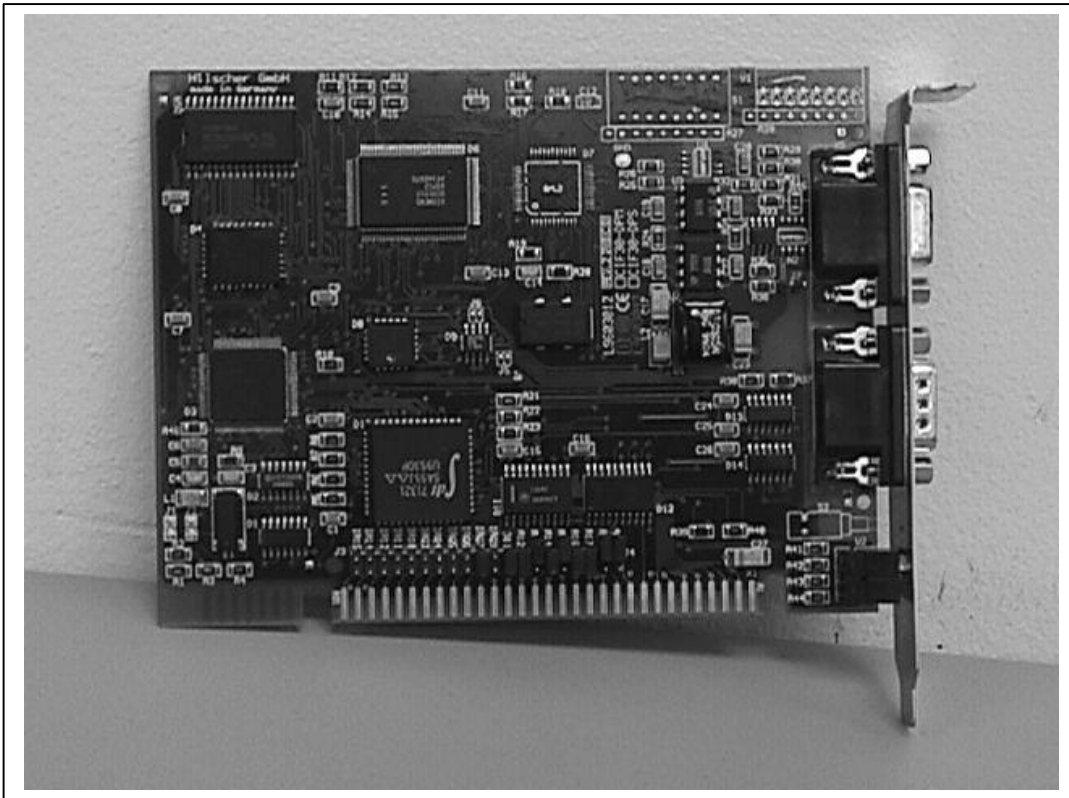
Introduction

The CIF 30 communication interface is an intelligent PC/ISA card. It is used to manage data transfer protocols between a PC and various industrial controllers input/output units. With its built-in microprocessor, it handles the complete data transfer process independently, and thus relieves the PC of time-critical functions.

The data is exchanged between the PC and CIF 30 via a 2k or 8k byte dual-port memory (DPM). This memory is accessed simultaneously by the PC and the microprocessor in the CIF. On each of the CIF 30 board there is a fieldbus chip that interfaces directly to the fieldbus network. In addition, the CIF 30 has a serial diagnostic interface.

The CIF 30 runs a real time operating system, which manages the data transfer via the DPM and controls the time sequences. The fieldbus protocols (Profibus, DeviceNet, InterBus, SDS, CanOpen, ControlNet, Modbus, ...) functions are implemented in three independent tasks. These tasks are the physical layer and the USER interface. The physical access on the medium and the data protection is done by the ASIC.

The CIF 30 can be used with any AT compatible PC that has an ISA slot.



Hardware Description

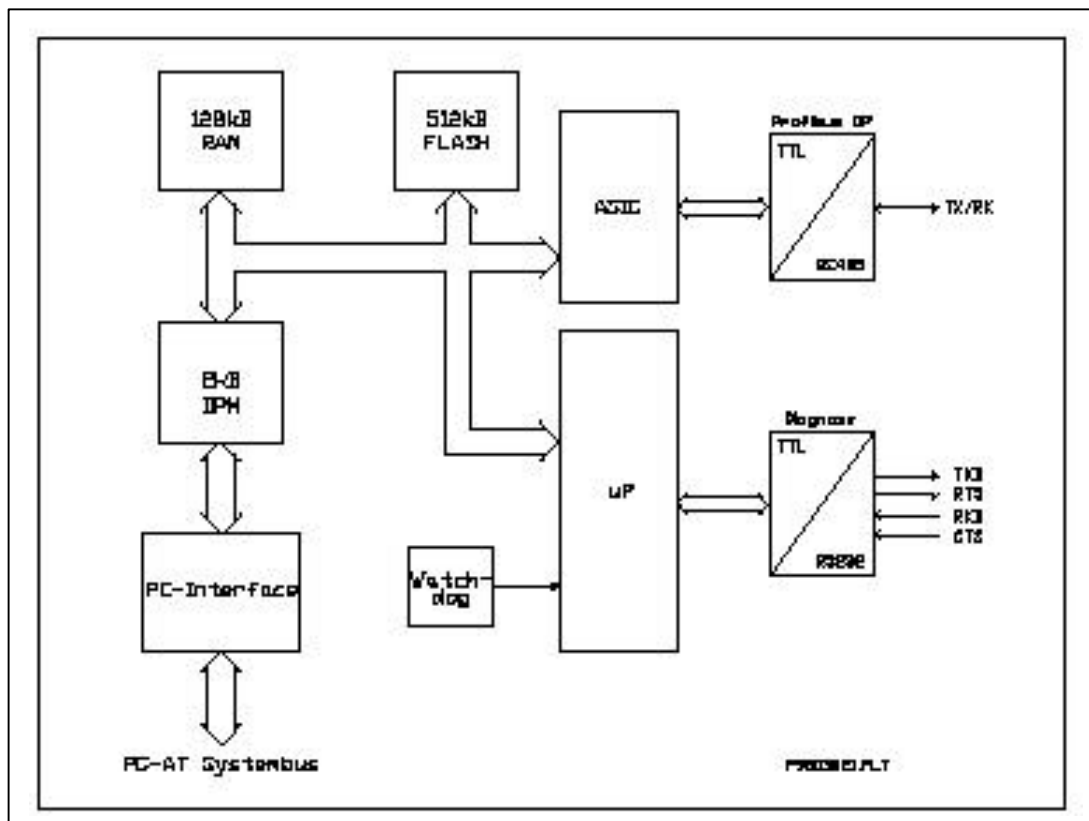
The heart of the CIF 30 computer is the microprocessor with RAM, Flash EPROM and dual port memory. The dual port memory is 2k or 8k bytes in size (depending on the type of CIF 30 card), and can be set to any addresses required within the address range of the PC jumpers. Interrupts can be triggered between the CIF 30 and PC by writing to a particular cell in the dual-port memory. The interrupt used for the PC can be set with a jumper.

A special function CIF 30 is a watchdog circuit. If this circuit is activated, a reset is triggered off on the card and the card is reset.

The diagnostic interface uses a serial RS232C asynchronous interface of the microprocessor and is permanently set as a non-isolated RS232C interface.

The interface to the PC has a bus data width of 8 bits, and is compatible to a standard PC/ISA bus interface.

The entire circuit is designed as a plug-in card and uses only an 8 or 16 bit PC/ISA slot. The card is supplied with its operating voltages via this slot. The internal auxiliary voltages are generated by a local DC/DC converter.



Configuration / Jumpers

The hardware of the CIF 30 is configured by address and interrupt jumpers. These are designated with the code letter J and consecutively numbered. The dual port memory jumpers have labels A19 through A11. A 2k-byte card has A19...A11, a 8k byte card as only A19...A13. The interrupt jumpers are labeled IRQ followed by a number (3,4,5,6,7,9,10,11,12,14,15). The memory address occupies a 2k or 8k byte block of memory depending on the card used.

Card Name	Network Type	DPM Size
SMS-CIF30-DPM	Profibus DP Master	2k
SMS-CIF30-DPS	Profibus DP Slave	2k
SMS-CIF30-PB	Profibus DP/FMS Master	8k
SMS-CIF30-DNM	DeviceNet Master	8k
SMS-CIF30-SDSM	SDS Master	8k
SMS-CIF30-COM	CANOpen Master	8k
SMS-CIF30-IBM	InterBus Master	2k
SMS-CIF30-IBS	InterBus Slave	2k
SMS-CIF30-CNS	ControlNet Adapter / Slave	2k

The location is in binary, however by using the chart below, you can quickly and easily configure the card for any of the supported locations. Selecting three nibble addresses configures the memory address of the board, these represent the upper three hex address values. To select the value for each nibble look up the desired values on the left column and install or remove the jumpers in that memory address location (A19...A11).

The default location for the CIF 30 series of cards is CA00:00, the clear areas in Figure 1 indicate this default location, while figure 2 shows how the board is jumpered from the factory.

Memory Address Selection Chart

	Nibble 1				Nibble 2				Nibble 3
	A19	A18	A17	A16	A15	A14	A13	A12	A11
0	I	I	I	I	I	I	I	I	I
1	I	I	I	B	I	I	I	B	B
2	I	I	B	I	I	I	B	I	
3	I	I	B	B	I	I	B	B	
4	I	B	I	I	I	B	I	I	
5	I	B	I	B	I	B	I	B	
6	I	B	B	I	I	B	B	I	
7	I	B	B	B	I	B	B	B	
8	B	I	I	I	B	I	I	I	
9	B	I	I	B	B	I	I	B	
A	B	I	B	I	B	I	B	I	
B	B	I	B	B	B	I	B	B	
C	B	B	I	I	B	B	I	I	
D	B	B	I	B	B	B	I	B	
E	B	B	B	I	B	B	B	I	
F	B	B	B	B	B	B	B	B	

I= Installed B= Blank

Figure 1: Dual port memory address block

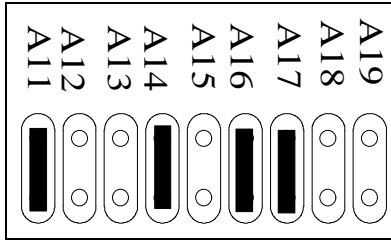


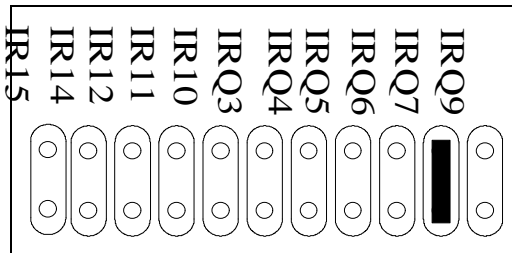
Figure 2:

IRQ Jumpers

The CIF series of cards support IRQ locations 3, 4, 5, 6, 7, 9, 10, 11, 12, 14 and 15 as well as polled operation. All the locations are designated on the silk screen of the board with the abbreviation of “IR” or “IRQ” followed by the number. For a specific IRQ, jumper the IRQ position desired while leaving all the other positions open. The factory default is IRQ 7 as shown in Figure 3.

For polled operation, leave all IRQ positions open.

IRQ Jumper Block



Display / LEDs

4 LED's are available on the CIF 30 for status display:

Display	Color	State	Description
RDY	Yellow	On Flashing Cyclic Flashing non-Cyclic Off	CIF ready Bootstrap loader active Hardware or system error Hardware error
RUN	Green	On Flashing Cyclic Flashing non-Cyclic Off	Communications running Ready for communications Parameter error No communications
ERR	Red	On	Error on communication line
STA	Yellow		State depends on firmware

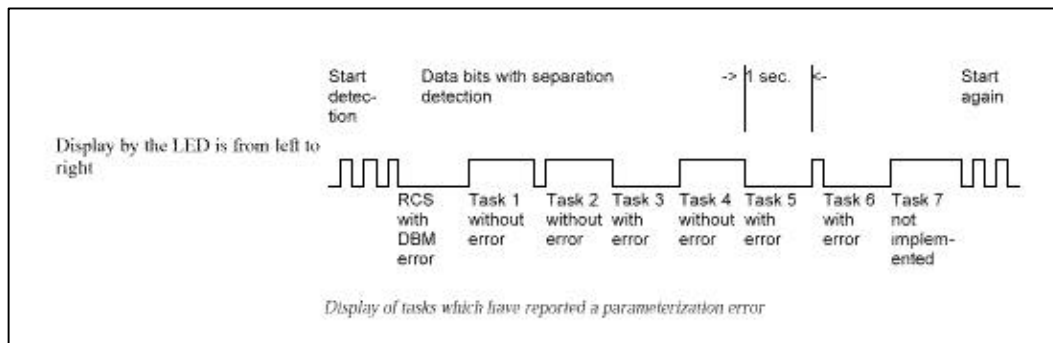
When switched on, the CIF 30 performs a self-test. If this is performed satisfactory, the yellow RDY LED is switched on. Otherwise, the LED starts to flash, and further running of the program is aborted. An error number is entered in the RcsError variable in the DPM. If the LED remains off, the card is defective.

If no firmware has been loaded on the CIF 30, the bootstrap loader displays this by regular flashing of the RDY LED at one-second intervals. The flash frequency is increased to approx. 2 Hz while the firmware is being loaded.

If a parameterization error is detected by a protocol task, the RUN LED as shown in the illustration below displays the task. If no error occurs and communication has started, the RUN LED is switched on. If this LED blinking cyclic, no parameterization error has been detected, but the communication on the bus has not been established. If communication is blocked, e.g. by the 'System start' parameter, the RUN LED remains off.

The red ERR LED displays data transfer errors at the communications interface.

The yellow status LED STA is activated when the CIF 30 holds the token.

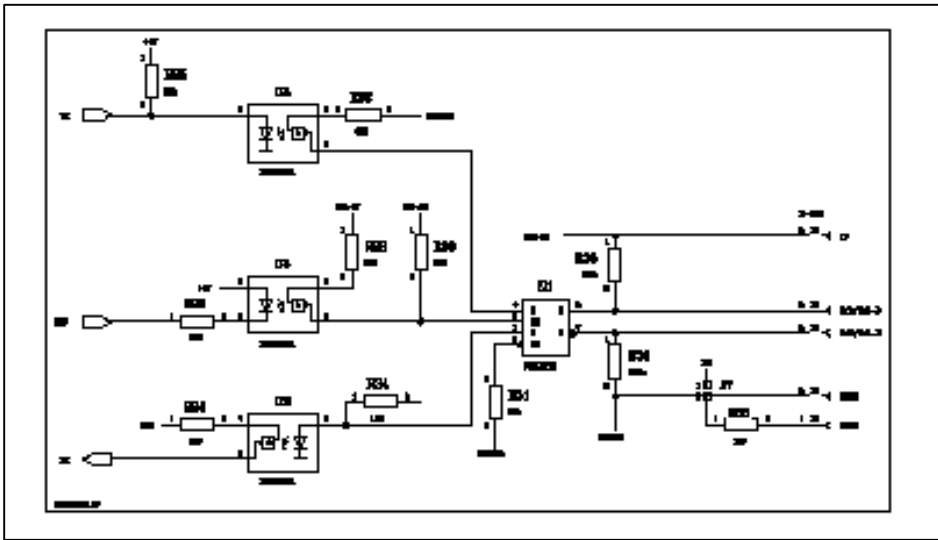


Fieldbus Connection

The CIF 30 has two connectors attached to the metal faceplate. The top connector is the fieldbus connector and is connected directly to the fieldbus (Profibus, DeviceNet,...). The bottom connector is the diagnostic RS232 connector and is used for diagnostics or remote configurations. The CIF 30 status LED's appear below the diagnostic connector. Each of the different type of CIF 30's fieldbus interfaces is described below.

PROFIBUS Interfaces (SMS-CIF30-DPM, SMS-CIF30-PB, SMS-CIF30-DPS)

The serial communication interface is installed as a electrically isolated floating RS485. The DC/DC converter supplies all necessary auxiliary voltage.

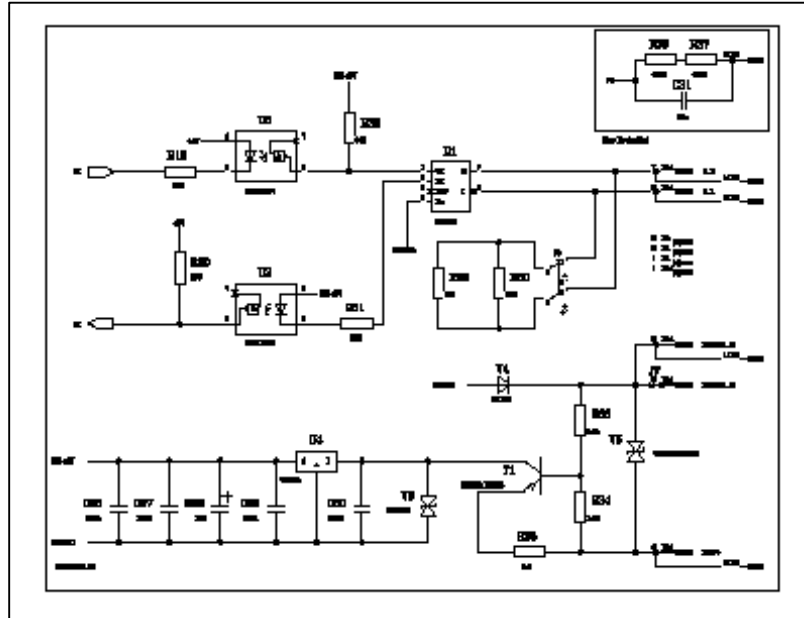


Jumper J7 has the purpose to connect the D-Sub plug housing to ISO GND (Pin 5). Resistors R35 and R36 set a closed level on the RS485 bus. This ensures a defined level in the bus transfer phases and when the plugs are removed. The cable shield of the bus plug is directly connected to the front of the PC main card. The grounding follows over the chassis of the PC.

Pin	Signal	Symbol	Type
1	Reference potential over 100 ohm resistor	RGND	
2	Not connected		
3	Send/Receive data-Positive	RxD/TxD-P	RS485
4	Not connected		
5	Reference potential	DGND	
6	Positive supply voltage	VP	
7	Not connected		
8	Send/Receive data-Negative	RxD/TxD-N	RS485
9	Not connected		
shield	Grounding Potential, bracket	PE	

DeviceNet Interface (SMS-CIF30-DNM)

The serial communication interface is isolated by optocouplers. A local DC/DC converter generates all internal voltages on the SMS-CIF30-DNM. The DC voltage for the fieldbus interface of the SMS-CIF30-DNM is over the bus cable.

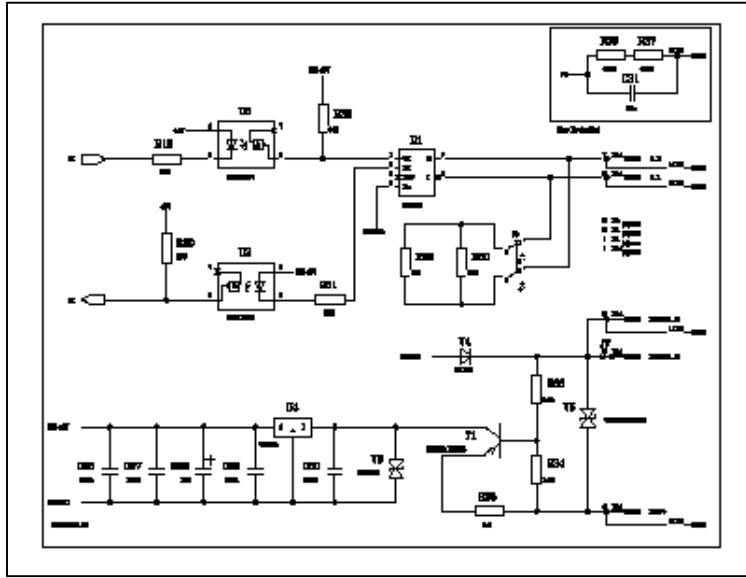


The SMS-CIF30-DNM has a 5 position COMBICON connector. The shield is connected with the PC bracket over a RC combination (1M Ω , 10nF). The following table shows the pinout diagram of the SMS-CIF30-DNM. It is possible to turn on the bus termination (121 Ohm, 0.5W) by the switch J8.

Pin	Signal	Symbol	Type
2	Receive- / Send data low	CAN-L	ISO-High Speed
1	Data Ground	DGND	
4	Receive / Send data high	CAN-H	ISO-High Speed
5	+24V Power supply	+V	
3	Shield over 1M Ω / 10nF to PC bracket	PE	

SDS Interface (SMS-CIF30-SDSM)

The serial communication interface is isolated by optocouplers. A local DC/DC converter generates all internal voltages on the SMS-CIF30-SDSM. The DC voltage for the fieldbus interface of the SMS-CIF30-SDS is over the bus cable.

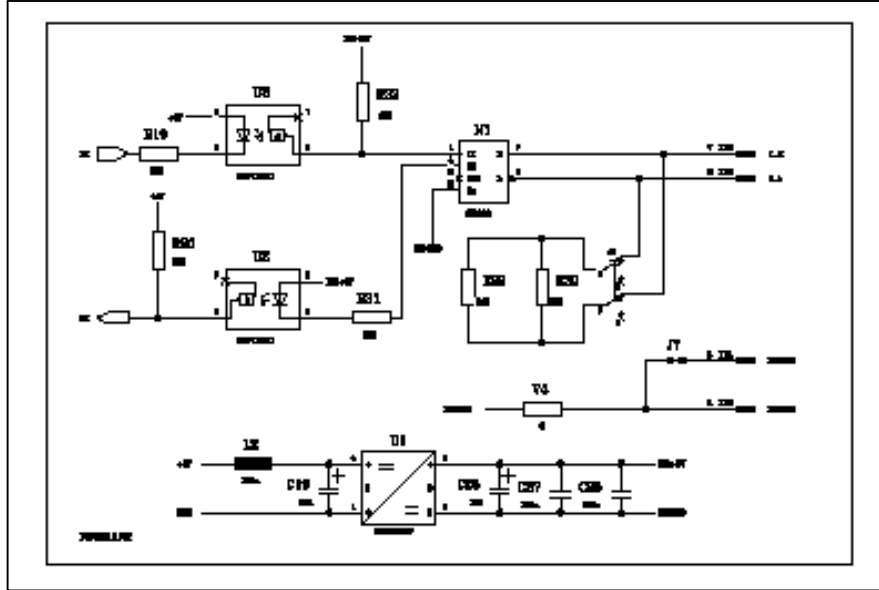


The SMS-CIF30-SDSM has a 9 way D-Sub male connector for the fieldbus and diagnostic interface. The shield of the connector is directly connected to the PC bracket. The following table shows the pinout diagram of the SMS-CIF30-SDSM. It is possible to turn on the bus termination (121 Ohm, 0.5W) by the switch J8.

Pin	Signal	Symbol	Type
2	Receive- / Send data low	CAN-L	ISO-High Speed
3	Data Ground	DGND	
6	Data Ground	DGND	
7	Receive+ / Send data high	CAN-H	ISO-High Speed
9	+24V Power supply	+V	
Shield	Shield over 1Mohm / 10nF to PC bracket	PE	

CANOpen Interface (SMS-CIF30-COM)

The serial communication interface is isolated by optocouplers. A local DC/DC converter generates all internal voltages on the SMS-CIF30-COM.

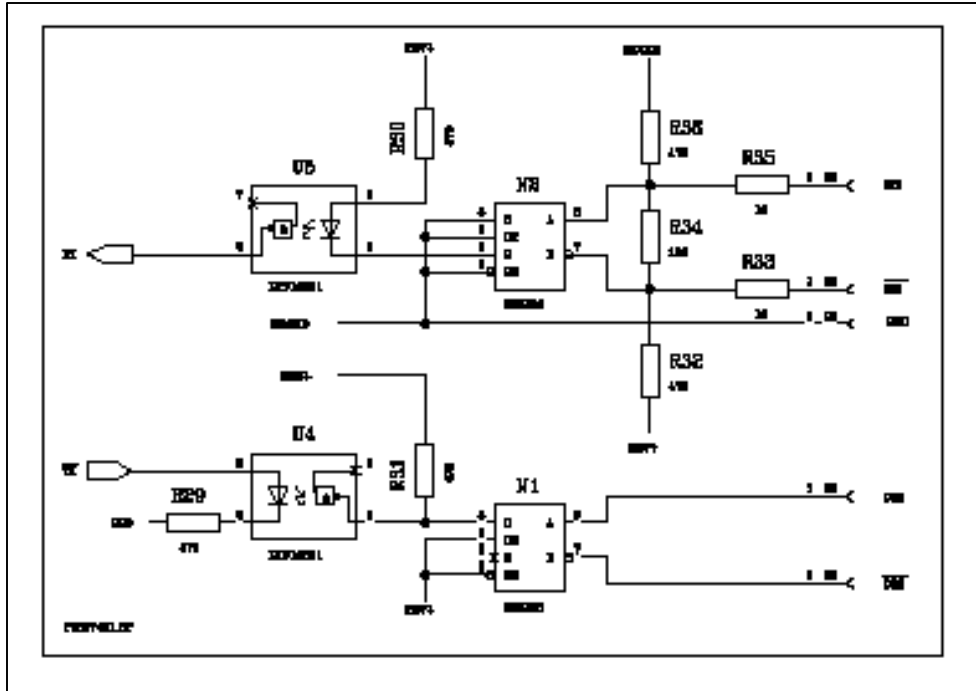


The SMS-CIF30-COM has a 9 way D-Sub male connector for the fieldbus and diagnostic interface. The shield of the connector is directly connected to the cardholder of the PC. The following table shows the pinout diagram of the SMS-CIF30-COM. It is possible to turn on the bus termination (121 Ohm, 0.5W) by the switch J8.

Pin	Signal	Symbol	Type
2	Receive- / Send data low	CAN-L	ISO-High Speed
3	Data Ground	DGND	
6	Data Ground	DGND	
7	Receive+ / Send data high	CAN-H	ISO-High Speed
9	+24V Power supply	+V	
Shield	Shield over 1Mohm / 10nF to PC bracket	PE	

InterBus Interface (SMS-CIF30-IBM,SMS-CIF30-IBS)

The complete interface for InterBus is implemented using a special fieldbus controller. The following figure shows the RS422 interface circuit:



For InterBus it is necessary to use a double twisted pair cable and one ground cable with a wave resistance of 100 - 130 Ohm. The shield of the fieldbus connector is directly connected with the front panel of the PC-host card. So earth ground is possible over the PC case.

Pin	Signal	Symbol	Type
1	Transmit data +	DO2	RS485
2	Receive data +	DI2	RS485
3	Isolated 0V Ground	ISOGND2	
4	Not connected	NC	
5	Isolated 5V power	ISOV*2	
6	Transmit data -	/DO2	RS485
7	Receive data -	/DI2	RS485
8	Not connected	NC	
9	Not connected	NC	
Shield	Earth ground, PC bracket	PE	

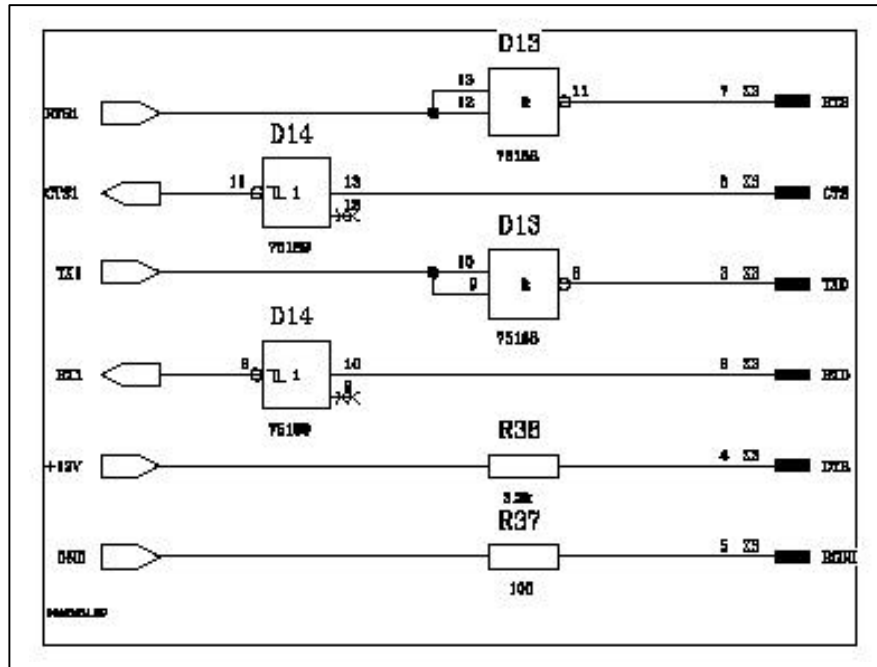
ControlNet Interface (SMS-CIF30-CNS)

The complete interface for ControlNet is implemented using the Rockwell ControlNet Interface controller. The pinout for the two redundant coaxial ControlNet connections are specified by the ControlNet specification. The simple RJ configuration port is also available on the back plane.

Diagnostic RS232 Connection

The serial RS232 interface is designed as a non-electrically-isolated RS232C interface. The circuit diagram for the interface is shown below. The signal DTR has +12V. After the initialization of the CIF the signal RTS is switched on. The signal CTS is not evaluated. This means that there is no synchronization with control lines.

The ControlNet Adapter as the serial port connection via a 10-pin header on the top half of the PC board. A simple ribbon cable to DB9 cable must be used to utilize the diagnostic serial port.



Pin	Input/Output	Signal CCITT	Signal DIN	Signal Name
2	Input	RXD	D2	Transmit data
3	Output	TXD	D1	Receive data
4	Output	DTR	S1.2	Data terminal ready
5	Output	GND	E2	Ground
7		RTS	S2	Ready to send
8	Input	CTS	M2	Clear to send

Network Configuration

First Check out the Card

Plug in the card in a free slot of the PC. Please be sure that the PC is powered off. Power on the PC and watch the LED's at the front of the card. After an initialization time of 2-3 seconds, both lower LED has to turn off and then turn the upper yellow RDY-LED on. If the yellow RDY-LED of the card remains off or starts flashing, the card is defective.

If a parameterization error is detected, the yellow RDY-LED is on and the green Run-LED is flashing uncyclic. In this case at least one protocol task has detected a parameterization error, but the CIF 30 respond to the diagnostic program. If the initialization finishes successfully, the green Run-LED flashes cyclic. It shines continuous, when there is communication to at least one slave and process data can exchange with this slave.

Install the software

Each CIF 30 needs configuration software to give it the exact network configuration. The SYCON 2 software provides this configuration in all cases except the Modbus Plus and ControlNet adapter cases. After installing the SYCON 2 software the user can check the firmware version number and date on the card. If successful this proves that the software and operating system driver are installed properly. If not see the SYCON 2 manual.

Firmware

All cards come with the firmware loaded and tested. For some reason if the firmware needs to be updated the SYCON software tools can be used to update the firmware. The firmware has the extension '.HXX' and is loadable in flash EPROM

ANÁLISIS DEL ESTADO DEL ARTE DE LOS BUSES DE CAMPO APLICADOS AL CONTROL DE PROCESOS INDUSTRIALES

Dr.-Ing. Héctor Kaschel C.

Ing. Ernesto Pinto L.

Fac. de Ingeniería, Depto. de Ingeniería Eléctrica

Universidad de Santiago de Chile

Avda Ecuador 3519, Estación Central. Santiago, CHILE

Fonos: (56) 2-77866417 (56) 2-6762452 Fax. (56) 2-6819079, Santiago, CHILE

E-mail: hkaschel@lauca.usach.cl

e.pinto@ieeee.org

RESUMEN

El desarrollo del control distribuido que está formado por una gran variada de campos va paralelo al de las comunicaciones. Esto está evocado a diferentes niveles de abstracción sobre integración y producción de acuerdo a la filosofía de la "Computer Integrated Manufacturing" -CIM. Cada vez es más necesario disponer de dispositivos inteligentes para realizar el control o la supervisión remota. Un bus de campo transfiere información secuencial y serial por un número limitado de líneas o cables. Hay muchos tipos diferentes de buses en uso y muchos son altamente dependientes de las aplicaciones. Este artículo se analiza el estado de avance en la tecnología de la comunicación de los buses de campo aplicados al control de procesos industriales.

1. INTRODUCCIÓN

Un bus de campo es un sistema de transmisión de información (datos) que simplifica enormemente la instalación y operación de máquinas y equipamientos industriales utilizados en procesos de producción. El objetivo de un bus de campo es sustituir las conexiones punto a punto entre los elementos de campo y el equipo de control a través del tradicional bucle de corriente de 4-20mA. Típicamente son redes digitales, bidireccionales, multipunto, montadas sobre un bus serie, que conectan dispositivos de campo como PLCs, transductores, actuadores y sensores. Cada dispositivo de campo incorpora cierta capacidad de proceso, que lo convierte en un dispositivo inteligente, manteniendo siempre un costo bajo. Cada uno de estos elementos será capaz de ejecutar funciones simples de diagnóstico, control o mantenimiento, así como de comunicarse bidireccionalmente a través del bus.

El objetivo es reemplazar los sistemas de control centralizados por redes de control distribuido mediante el cual permita mejorar la calidad del producto, reducir los costos y mejorar la eficiencia. Para ello se basa en

que la información que envían y/o reciben los dispositivos de campo es digital, lo que resulta mucho más preciso que si se recurre a métodos analógicos. Además, cada dispositivo de campo es un dispositivo inteligente y puede llevar a cabo funciones propias de control, mantenimiento y diagnóstico. De esta forma, cada nodo de la red puede informar en caso de fallo del dispositivo asociado, y en general sobre cualquier anomalía asociada al dispositivo. Esta monitorización permite aumentar la eficiencia del sistema y reducir la cantidad de horas de mantenimiento necesarias.

2. VENTAJAS DE LOS BUSES DE CAMPO

La principal ventaja que ofrecen los buses de campo, y la que los hace más atractivos a los usuarios finales, es la reducción de costos. El ahorro proviene fundamentalmente de tres fuentes: ahorro en costo de instalación, ahorro en el costo de mantenimiento y ahorros derivados de la mejora del funcionamiento del sistema. Una de las principales características de los buses de campo es su significativa reducción en el cableado necesario para el control de una instalación. Cada componente sólo requiere un cable para la conexión de los diversos nodos. Se estima que puede ofrecer una reducción de 5 a 1 en los costos de cableado. En comparación con otros tipos de redes, dispone de herramientas de administración del bus que permiten la reducción del número de horas necesarias para la instalación y puesta en marcha.

El hecho de que los buses de campo sean más sencillos que otras redes de uso industrial como por ejemplo MAP, hace que las necesidades de mantenimiento de la red sean menores, de modo que la fiabilidad del sistema a largo plazo aumenta. Además, los buses de campo permiten a los operadores monitorizar todos los dispositivos que integran el sistema e interpretar fácilmente las interacciones entre ellos. De esta forma, la detección de las fuentes de problemas en la planta y su corrección resulta mucho más sencilla, reduciendo

los costos de mantenimiento y el tiempo de parada de la planta.

Los buses de campo ofrecen mayor flexibilidad al usuario en el diseño del sistema. Algunos algoritmos y procedimientos de control que con sistemas de comunicación tradicionales debían incluirse en los propios algoritmos de control, radican ahora en los propios dispositivos de campo, simplificando el sistema de control y sus posibles ampliaciones.

También hay que tener en cuenta que las prestaciones del sistema mejoran con el uso de la tecnología de los buses de campo debido a la simplificación en la forma de obtener información de la planta desde los distintos sensores. Las mediciones de los distintos elementos de la red están disponibles para todos los demás dispositivos. La simplificación en la obtención de datos permitirá el diseño de sistemas de control más eficientes.

Con la tecnología de los buses de campo, se permite la comunicación bidireccional entre los dispositivos de campo y los sistemas de control, pero también entre los propios dispositivos de campo.

Otra ventaja de los buses de campo es que sólo incluyen 3 capas (Física, Enlace y Aplicación), y un conjunto de servicios de administración. El usuario no tiene que preocuparse de las capas de enlace o de aplicación. Sólo necesita saber cual es funcionalidad. Al usuario sólo se le exige tener un conocimiento mínimo de los servicios de administración de la red, ya que parte de la información generada por dichos servicios puede ser necesaria para la reparación de averías en el sistema. De hecho, prácticamente, el usuario sólo debe preocuparse de la capa física y la capa de usuario.

3. BUSES DE CAMPO EXISTENTES

Debido a la falta de estándares, diferentes compañías han desarrollado diferentes soluciones, cada una de ellas con diferentes prestaciones y campos de aplicación. En una primera clasificación tenemos los siguientes grupos:

3.1. Buses de alta velocidad y baja funcionalidad

Están diseñados para integrar dispositivos simples como finales de carrera, fotocélulas, relés y actuadores simples, funcionando en aplicaciones de tiempo real, y agrupados en una pequeña zona de la planta, típicamente una máquina. Básicamente comprenden las capas física y de enlace del modelo OSI, es decir,

señales físicas y patrones de bits de las tramas. Algunos ejemplos son:

- CAN: Diseñado originalmente para su aplicación en vehículos.
- SDS: Bus para la integración de sensores y actuadores, basado en CAN
- ASI: Bus serie diseñado por Siemens para la integración de sensores y actuadores.

3.2. Buses de alta velocidad y funcionalidad media

Se basan en el diseño de una capa de enlace para el envío eficiente de bloques de datos de tamaño medio. Estos mensajes permiten que el dispositivo tenga mayor funcionalidad de modo que permite incluir aspectos como la configuración, calibración o programación del dispositivo. Son buses capaces de controlar dispositivos de campo complejos, de forma eficiente y a bajo costo. Normalmente incluyen la especificación completa de la capa de aplicación, lo que significa que se dispone de funciones utilizables desde programas basados en PCs para acceder, cambiar y controlar los diversos dispositivos que constituyen el sistema. Algunos incluyen funciones estándar para distintos tipos de dispositivos (perfiles) que facilitan la interoperabilidad de dispositivos de distintos fabricantes. Algunos ejemplos son:

- DeviceNet: Desarrollado por Allen-Bradley, utiliza como base el bus CAN, e incorpora una capa de aplicación orientada a objetos.
- LONWorks: Red desarrollada por Echelon.
- BitBus: Red desarrollada por INTEL.
- DIN MessBus: Estándar alemán de bus de instrumentación, basado en comunicación RS-232.
- InterBus-S: Bus de campo alemán de uso común en aplicaciones medias.

3.2. Buses de altas prestaciones

Son capaces de soportar comunicaciones a nivel de todos los niveles de la producción CIM. Aunque se basan en buses de alta velocidad, algunos presentan problemas debido a la sobrecarga necesaria para alcanzar las características funcionales y de seguridad que se les exigen. La capa de aplicación tiene un gran número de servicios a la capa de usuario, habitualmente un subconjunto del estándar MMS (Manufacturing Message Specification). Entre sus características incluyen:

- Redes multi-maestro con redundancia.

- Comunicación maestro-esclavo según el esquema pregunta-respuesta.
- Recuperación de datos desde el esclavo con un límite máximo de tiempo
- Capacidad de direccionamiento unicast, multicast y broadcast,
- Petición de servicios a los esclavos basada en eventos.
- Comunicación de variables y bloques de datos orientada a objetos.
- Descarga y ejecución remota de programas.
- Altos niveles de seguridad de la red, opcionalmente con procedimientos de autenticación.
- Conjunto completo de funciones de administración de la red.

Algunos ejemplos son:

- Profibus
- WorldFIP
- Fieldbus Foundation

3.4. Buses para áreas de seguridad intrínseca

Incluyen modificaciones en la capa física para cumplir con los requisitos específicos de seguridad intrínseca en ambientes con atmósferas explosivas. La seguridad intrínseca es un tipo de protección por la que el componente en cuestión no tiene posibilidad de provocar una explosión en la atmósfera circundante. Un circuito eléctrico o una parte de un circuito tienen seguridad intrínseca, cuando alguna chispa o efecto térmico en este circuito producidos en las condiciones de prueba establecidas por un estándar (dentro del cual figuran las condiciones de operación normal y de fallo específicas) no puede ocasionar una ignición. Algunos ejemplos son HART, Profibus PA o WorldFIP.

4. ALGUNOS BUSES ESTANDARIZADOS

4.1 PROFIBUS

Profibus se desarrolló bajo un proyecto financiado por el gobierno alemán. Está normalizado en Alemania por DIN E 19245 y en Europa por EN 50170. El desarrollo y posterior comercialización ha contado con el apoyo de importantes fabricantes como ABB, AEG, Siemens, Klöckner-Moeller, ... Está controlado por la PNO (Profibus User Organisation) y la PTO (Profibus Trade Organisation).

Existen tres perfiles:

- Profibus DP (Decentralized Periphery). Orientado a sensores/actuadores enlazados a procesadores (PLCS) o terminales.
- Profibus PA (Process Automation). Para control de proceso y cumpliendo normas especiales de seguridad para la industria química (IEC 1 / 15 8-2, seguridad intrínseca).
- Profibus FMS (Fieldbus Message Specification). Para comunicación entre células de proceso o equipos de automatización. La evolución de Profibus hacia la utilización de protocolos TCP/IP para enlace al nivel de proceso hace que este perfil esté perdiendo importancia.

Utiliza diferentes capas físicas. La más importante, en PROFIBUS DP, está basada en EIA RS-485. Profibus PA utiliza la norma IEC 11158-2 (norma de comunicación síncrona entre sensores de campo que utiliza modulación sobre la propia línea de alimentación de los dispositivos y puede utilizar los antiguos cableados de instrumentación 4-20 mA) y para el nivel de proceso se tiende a la utilización de Ethernet. También se contempla la utilización de enlaces de fibra óptica. Existen puentes para enlace entre diferentes medios, además de gateways que permiten el enlace entre perfiles y con otros protocolos.

Se distingue entre dispositivos tipo maestro y dispositivos esclavo. El acceso al medio entre maestros se arbitra por paso de testigo, el acceso a los esclavos desde un maestro es un proceso de interrogación cíclico (polling). Se pueden configurar sistemas multimaestro o sistemas más simples maestro-esclavo.

En Profibus DP se distingue entre: maestro clase 1 (estaciones de monitorización y diagnóstico), maestro clase 2 (elementos centralizadores de información como PLCS, PCs, etc.), esclavo (sensores, actuadores).

El transporte en Profibus-DP se realiza por medio de tramas según IEC 870-5-1. La comunicación se realiza por medio de datagramas en modo broadcast o multicast. Se utiliza comunicación serie asíncrona por lo que es utilizable una UART genérica.

Profibus DP prescinde de los niveles ISO 3 a 6 y la capa de aplicación ofrece una amplia gama de servicios de diagnóstico, seguridad, protecciones etc. Es una capa de aplicación relativamente compleja debido a la

necesidad de mantener la integridad en el proceso de paso de testigo (un y sólo un testigo)

Profibus FMS es una compleja capa de aplicación que permite la gestión distribuida de procesos al nivel de relación entre células con posibilidad de acceso a objetos, ejecución remota de procesos etc. Los dispositivos se definen como dispositivos de campo virtuales, cada uno incluye un diccionario de objetos que enumera los objetos de comunicación. Los servicios disponibles son un subconjunto de los definidos en MMS (ISO 9506)

Las plataformas hardware utilizadas para soportar Profibus se basan en microprocesadores de 16 bits más procesadores de comunicaciones especializados o circuitos ASIC como el LSPM2 de Siemens. La PNO se encarga de comprobar y certificar el cumplimiento de las especificaciones PROFIBUS.

Entre sus perspectivas de futuro se encuentra la integración sobre la base de redes Ethernet al nivel de planta y la utilización de conceptos de tiempo real y filosofía productor-consumidor en la comunicación entre dispositivos de campo.

Las distancias potenciales de bus van de 100 m a 24 Km (con repetidores y fibra óptica). La velocidad de comunicación puede ir de 9600 bps a 12 Mbps. Utiliza mensajes de hasta 244 bytes de datos.

Profibus se ha difundido ampliamente en Europa y también tiene un mercado importante en América y Asia. El conjunto Profibus DP- Profibus PA cubre la automatización de plantas de proceso discontinuo y proceso continuo cubriendo normas de seguridad intrínseca.

4.2. INTERBUS

Protocolo propietario, inicialmente, de la empresa Phoenix Contact GmbH, aunque posteriormente ha sido abierta su especificación. Normalizado bajo DIN 19258, norma europea EN 50 254. Fue introducido en el año 1984.

Utiliza una topología en anillo y comunicación mediante un registro de desplazamiento en cada nodo. Se pueden enlazar buses periféricos al principal.

Capa física basada en RS-485. Cada dispositivo actúa como repetidor. Así se puede alcanzar una distancia entre nodos de 400 m para 500Kbps y una distancia

total de 12 KM. Es posible utilizar también enlaces de fibra óptica.

Capa de transporte basada en una trama única que circula por el anillo (trama de suma)

La información de direccionamiento no se incluye en los mensajes, los datos se hacen circular por la red. Alta eficiencia. Para aplicaciones de pocos nodos y un pequeño conjunto de entradas/salidas por nodo, pocos buses pueden ser tan rápidos y eficientes como INTERBUS.

Físicamente tiene la impresión de seguir una topología en estrella, pero realmente cada nodo tiene un punto de entrada y otro de salida hacia el siguiente nodo.

Es muy sensible a corte completo de comunicación al abrirse el anillo en cualquiera de los nodos. Por otra parte, la estructura en anillo permite una fácil localización de fallos y diagnóstico.

Es muy apropiado para comunicación determinista a alta velocidad, es muy difícil una filosofía de comunicación orientada a eventos.

4.3. DeviceNet

Bus basado en CAN. Su capa física y capa de enlace se basan en ISO 11898, y en la especificación de Bosch 2.0. DeviceNet define una de las más sofisticadas capas de aplicaciones industriales sobre bus CAN.

DeviceNet fue desarrollado por Allen-Bradley a mediados de los noventa, posteriormente pasó a ser una especificación abierta soportada en la ODVA (Open DeviceNet Vendor Association), Cualquier fabricante puede asociarse a esta organización y obtener especificaciones, homologar productos, etc.

Es posible la conexión de hasta 64 nodos con velocidades de 125 Kbps a 500 Kbps en distancias de 100 a 500 m.

Utiliza una definición basada en orientación a objetos para modelar los servicios de comunicación y el comportamiento externo de los nodos. Define mensajes y conexiones para funcionamiento maestro-esclavo, interrogación cíclica, "strobing" o lanzamiento de interrogación general de dispositivos, mensajes espontáneos de cambio de estado, comunicación uno-

uno, modelo productor-consumidor, carga y descarga de bloques de datos y ficheros etc.

DeviceNet ha conseguido una significativa cuota de mercado. Existen más de 300 productos homologados y se indica que el número de nodos instalados superaba los 300.000 en 1998, Está soportado por numerosos fabricantes: Allen-Bradley, ABB, Danfoss, Crouzet, Bosh, Control Techniques, Festo, Omron, .etc.

4.4. FOUNDATION FIELDBUS

Un bus orientado sobre todo a la interconexión de dispositivos en industrias de proceso continuo. Su desarrollo ha sido apoyado por importantes fabricantes de instrumentación (Fisher-Rosemount, Foxboro,...). En la actualidad existe una asociación de fabricantes que utilizan este bus, que gestiona el esfuerzo normalizador, la Fieldbus Foundation. Normalizado como ISA SP50, IEC-ISO 61158 (ISA es la asociación internacional de fabricantes de dispositivos de instrumentación de proceso).

En su nivel H1 (uno) de la capa física sigue la norma IEC 11158-2 para comunicación a 31,25 Kbps, es por tanto, compatible con Profibus PA, su principal contendiente. Presta especial atención a las versiones que cumplen normas de seguridad intrínseca para industrias de proceso en ambientes combustibles o explosivos. Se soporta sobre par trenzado y es posible la reutilización de los antiguos cableados de instrumentación analógica 4-20 mA. Se utiliza comunicación síncrona con codificación Manchester Bifase-L.

La capa de aplicación utiliza un protocolo sofisticado, orientado a objetos con múltiples formatos de mensaje. Distingue entre dispositivos con capacidad de arbitraje (Link Master) y normales. En cada momento un solo Link master arbitra el bus, puede ser sustituido por otro en caso de fallo. Utiliza diversos mensajes para gestionar comunicación por paso de testigo, comunicación cliente-servidor, modelo productor-consumidor etc. Existen servicios para configuración, gestión de diccionario de objetos en nodos, acceso a variables, eventos, carga descarga de ficheros y aplicaciones, ejecución de aplicaciones, etc. La codificación de mensajes se define según ASN.1

El nivel H2 (dos) está basado en Ethernet de alta velocidad (100 Mbps) y orientado al nivel de control de la red industrial.

4.5. FIP- WorldFIP

Desarrollado en Francia a finales de los ochenta y normalizado por EN 50170, que también cubre Profibus. Sus capas física y de aplicación son análogas a las de Foundation Fieldbus H1 y Profibus PA. La división Norteamérica de WorldFIP se unió a mediados de los noventa a la Fieldbus Foundation en el esfuerzo por la normalización de un bus industrial común.

Utiliza un modelo productor-consumidor con gestión de variables cíclicas, eventos y mensajes genéricos.

4.6. LONWORKS

La empresa Echelon, localizada en California, fue fundada en 1988. Comercializa el bus de campo LonWorks basado en el protocolo LonTalk y soportado sobre el NeuronChip. Alrededor de estas marcas ha construido toda una estructura de productos y servicios, hábilmente comercializados, dirigidos al mercado del control distribuido en domótica, edificios inteligentes, control industrial etc. Asegura que varios miles de empresas trabajan con LonWorks, que cientos de empresas comercializan productos basados en su bus y que se han instalado millones de nodos.

El protocolo LonTalk cubre todas las capas OSI. El protocolo se soporta en hardware y firmware sobre el NeuronChip. Se trata de un microcontrolador que incluye el controlador de comunicaciones y toda una capa de firmware que, además de implementar el protocolo, ofrece una serie de servicios que permiten el desarrollo de aplicaciones en el lenguaje Neuron C, una variante de ANSI C. Motorola y Toshiba fabrican el NeuronChip, además Echelon ofrece la posibilidad de abrir la implementación de LonWorks a otros procesadores.

La red Lonworks ofrece una variada selección de medios físicos y topologías de red: par trenzado en bus, anillo y topología libre, fibra óptica, radio, transmisión sobre red eléctrica etc. El soporte más usual es par trenzado a 38 o 78 Kbps. Se ofrece una amplia gama de servicios de red que permiten la construcción de extensas arquitecturas con multitud de nodos, dominios y grupos, típicas de grandes edificios inteligentes.

El método de comparación de medio es acceso CSMA predictivo e incluye servicios de prioridad de mensajes.

Echelon ofrece herramientas de desarrollo, formación, documentación y soporte técnico. Echelon basa su negocio en la comercialización del bus, medios, herramientas y soporte,

4.7. SDS

SDS ("Smart Distributed System") es, junto con DeviceNet y CANOpen, uno de los buses de campo basados en CAN más extendidos. Fue desarrollado por el fabricante de sensores industriales Honeywell en 1989.

Se ha utilizado sobre todo en aplicaciones de sistemas de almacenamiento, empaquetado y clasificación automática. Se define una capa física que incluye alimentación de dispositivos en las conexiones. La capa de aplicación define autodiagnóstico de nodos, comunicación por eventos y prioridades de alta velocidad.

4.8. CANOpen

Bus de campo basado en CAN. Fue el resultado de un proyecto de investigación financiado por la Comunidad Europea y se está extendiendo de forma importante entre fabricantes de maquinaria e integradores de célula de proceso. Está soportado por la organización CiA (CAN In Automation), organización de fabricantes y usuarios de CAN que también apoya DeviceNet, SDS etc. Al final de este trabajo se describirá con más detalle este bus, como ejemplo de bus de campo normalizado soportado sobre CAN.

4.9.- MODBUS

En su definición inicial Modbus era una especificación de tramas, mensajes y funciones utilizada para la comunicación con los PLCs Modicon. Modbus puede implementarse sobre cualquier línea de comunicación serie y permite la comunicación por medio de tramas binarias o ASCII con un proceso interrogación-respuesta simple. Debido a que fue incluido en los PLCs de la prestigiosa firma Modicon en 1979, ha resultado un estándar de facto para el enlace serie entre dispositivos industriales.

Modbus Plus define un completo bus de campo basado en técnica de paso de testigo. Se utiliza como soporte físico el par-trenzado o fibra óptica.

En la actualidad Modbus es soportado por el grupo de automatización Schneider (Telemecanique, Modicon,...).

4.10.- INDUSTRIAL ETHERNET

La norma IEEE 802.3 basada en la red Ethernet de Xerox se ha convertido en el método más extendido para interconexión de computadores personales en redes de proceso de datos. En la actualidad se vive una auténtica revolución en cuanto a su desplazamiento hacia las redes industriales. Es indudable esa penetración. Diversos buses de campo establecidos como Profibus, Modbus etc. han adoptado Ethernet como la red apropiada para los niveles superiores. En todo caso se buscan soluciones a los principales inconvenientes de Ethernet como soporte para comunicaciones industriales:

- El intrínseco indeterminismo de Ethernet se aborda por medio de topologías basadas en conmutadores. En todo caso esas opciones no son gratuitas.
- Se han de aplicar normas especiales para conectores, blindajes, rangos de temperatura etc. La tarjeta adaptadora Ethernet empieza a encarecerse cuando se la dota de robustez para un entorno industrial

Parece difícil que Ethernet tenga futuro a nivel de sensor, aunque puede aplicarse en nodos que engloban conexiones múltiples de entrada-salida.

Como conclusión Ethernet está ocupando un área importante entre las opciones para redes industriales, pero parece aventurado afirmar, como se ha llegado a hacer, que pueda llegar a penetrar en los niveles bajos de la pirámide CIM.

4.11. ASI

AS-I (Actuator Sensor Interface) es un bus de campo desarrollado inicialmente por Siemens, para la interconexión de actuadores y sensores binarios. Actualmente está recogido por el estándar IEC TG 17B.

A nivel físico, la red puede adoptar cualquier tipo de topología: estructura en bus, en árbol, en estrella o en anillo. Permite la interconexión de un máximo de 31 esclavos. La longitud máxima de cada segmento es de 100 metros. Dispone de repetidores que permiten la unión de hasta tres segmentos, y de puentes hacia redes Profibus. Como medio físico de transmisión, emplea un único cable que permite tanto la transmisión de datos como la alimentación de los dispositivos conectados a la red. Su diseño evita errores de polaridad al conectar nuevos dispositivos a la red. La incorporación o eliminación de elementos de la red no requiere la modificación del cable.

El cable consta de dos hilos sin apantallamiento. Para lograr inmunidad al ruido, la transmisión se hace basándose en una codificación Manchester

Cada esclavo dispone de hasta 4 entradas/salidas, lo que hace que la red pueda controlar hasta 124 E/S digitales. La comunicación sigue un esquema maestro-esclavo, en la cual el maestro interroga a las estaciones enviándoles mensajes (llamados telegramas) de 14 bits y el esclavo responde con un mensaje de 7 bits. La duración de cada ciclo pregunta respuesta es de 150 μ s. En cada ciclo de comunicación se deben consultar todos los esclavos, añadiendo dos ciclos extras para operaciones de administración del bus (detección de fallos). El resultado es un tiempo de ciclo máximo de 5ms.

4.12 BITBUS

Introducido por Intel a principios de los 80. Es un bus maestro-esclavo soportado sobre RS485 y normalizado en IEEE- 1118. Debido a su sencillez ha sido adoptado en redes de pequeños fabricantes o integradores. En su capa de aplicación se contempla la gestión de tareas distribuidas, es decir es, en cierto modo, un sistema multitarea distribuido. Existe una organización europea de soporte (Bitbus European User's Group).

4.13 ARCNet

Originalmente desarrollada como red para proceso de datos en los años '70 ARCNet ha encontrado aplicación en el mundo industrial. Su técnica de paso de testigo hace que sea predecible, determinista y robusta. Está normalizada como ANSI/ATA 878. 1. La velocidad de comunicación es de 2,5 Mbps con paquetes del 0 a 512 bytes. Soporta topología en bus y estrella y diversos medios físicos (cable coaxial, par trenzado, fibra óptica).

Es una red muy apropiada para un nivel intermedio en la jerarquía CIM. Algunos fabricantes proponen como jerarquía ideal para control industrial una basada en Ethernet en el nivel superior, ArcNET en el intermedio y CAN al nivel de celda de fabricación.

4.14 CONTROLNET

Bus de alta velocidad (5 Mbps) y distancia (hasta 5 Km), muy seguro y robusto promovido por Allen-Bradley. Utiliza cable RG6/U (utilizado en televisión por cable) y se basa en un controlador ASIC de Rockwell.

No es soportado por muchos fabricantes y resulta de elevado precio por nodo. Se ha utilizado para

interconexión de redes de PLCs y computadores industriales en aplicaciones de alta velocidad y ambientes muy críticos.

4.15. HART

Es un protocolo para bus de campo soportado por la HART Communication Foundation y la Fieldbus Foundation, Su campo de aplicación básico es la comunicación digital sobre las líneas analógicas clásicas de los sistemas de instrumentación, manteniendo éstas en servicio. Sus prestaciones como bus de campo son reducidas.

Utiliza el bus analógico estándar 4-20 mA sobre el que transmite una señal digital modulada en frecuencia (modulación FSK 1200-2200 Hz). Transmite a 1200 bps manteniendo compatibilidad con la aplicación analógica inicial y sobre distancias de hasta 3 Km.

Normalmente funciona en modo maestro-esclavo.

5. LA GUERRA DE LOS BUSES.

Ante la variedad de opciones existente, parece razonable pensar que fabricantes y usuarios hicieran un esfuerzo en la búsqueda de normativas comunes para la interconexión de sistemas industriales.

Lo que ha venido llamándose "la guerra de los buses" tiene que ver con la permanente confusión reinante en los entornos normalizadores en los que se debate la especificación del supuesto "bus de campo universal". Desde mediados de los años '80 la Comisión Electrotécnica Internacional (IEC-CEI) y la Sociedad de Instrumentación Americana (ISA) ha sido escenario del supuesto esfuerzo de los fabricantes para lograr el establecimiento de una norma única de bus de campo de uso general. En 1992 surgieron dos grupos, el ISP (Interoperable Systems Project) y WorldFIP cada uno promoviendo su propia versión del bus de campo. En el primer grupo estaban fabricantes como Siemens, Fisher-Rosemount, Foxboro y Yokogawa. En el segundo Allen-Bradley, HoneyWell, Square D y diversas empresas francesa. En 1994 ambos grupos se unieron en la Fieldbus Foundation. El debate se trasladó luego, y continua en la actualidad, a la conjunción de Fieldbus y el mundo Profibus. Los años pasan, la norma del supuesto bus universal nunca se acaba de *generar* y en *el camino aparecen nuevas opciones* como CAN, LonWorks, Ethernet. Incluso el debate es confuso y totalmente incomprensible, otras empresas participantes en el debate generaban en paralelo soluciones propias, es el caso de Allen-Bradley con DeviceNet y

HoneyWell con SDS. La realidad es que sólo los usuarios están realmente interesados en la obtención de normas de uso general. Los fabricantes luchan por su cuota de mercado y, en general, sólo están a favor de una norma cuando ésta recoge las características de su propia opción, lo cual es comprensible dadas las fuertes inversiones necesarias para el desarrollo de un bus industrial normalizado. El debate sigue abierto.

6. CONCLUSIONES

Se han presentado algunas características funcionales de los Buses de Campo detallando algunos de mayor uso en diferentes modalidades de ambiente de trabajo. Además, se incluyeron criterios útiles de la estructuración dentro del concepto CIM-OSI. Estos antecedentes pueden ayudar a los profesionales para encontrar soluciones racionales y bien encaminadas para los problemas de comunicaciones a los niveles de producción, a pesar la falta de normas internacionales definitivas, dentro de la gama de buses de campo existentes.

BIBLIOGRAFIA

- [1] Tindell K, "Calculating CAN response times", University of York, England 1995.
- [2] Mc Farlane Andy, "Tutorial: Fieldbus review", Sensor Review, Vol 17, Num 3, 1997.
- [3] Sirgo J.A., "Redes locales en entornos industriales: Buses de campo", Universidad de Oviedo, 1997.
- [4] Quezada J., "Bus CAN: Estado de buses industriales y aplicaciones" Escuela Técnica Superior de Ingenieros Industriales", 1999.
- [5] www.fieldbus.org, Fieldbus Organization.
- [6] www.bosh.de/KB/can, Página de la Bosh dedicada a CAN.
- [7] www.Kvaser.se. KingDom CAN applications.