

**LABORATORIO DE REDES INDUSTRIALES CON PLCs SIEMENS S7-200**

NORBERTO ALONSO ANGEL CABARCAS

WALBERTO JULIO CANTILLO PÁJARO

FACULTAD DE INGENIERÍAS

INGENIERIA ELECTRICA ELECTRÓNICA

UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR

CARTAGENA DE INDIAS D.T. Y C.

2008

**LABORATORIO DE REDES INDUSTRIALES CON PLCs SIEMENS S7-200**

NORBERTO ALONSO ANGEL CABARCAS

WALBERTO JULIO CANTILLO PÁJARO

Monografía presentada

Como requisito para optar al título de

Ingeniero Electrónico.

DIRECTOR

JORGE DUQUE

MSc. INGENIERIA ELECTRONICA

FACULTAD DE INGENIERÍAS

PROGRAMA DE INGENIERIAS ELECTRICA Y ELECTRÓNICA

UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR

CARTAGENA DE INDIAS D.T. Y C.

2008

**Cartagena, Mayo de 2008**

**Señores:**

**UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR**

Comité de evaluación de proyectos

La ciudad

Respetados señores:

De la forma mas cordial y mucho agrado, me permito presentar a ustees la monografía titulada: “**LABORATORIO DE REDES INDUSTRIALES CON PLCs SIEMENS S7-200**”, desarrollada por los estudiantes de Ingenieria Electronica, **NORBERTO ALONSO ANGEL CABARCAS Y WALBERTO JULIO CANTILLO PAJARO**.

Al respecto de dicho trabajo, el cual he dirigido, lo considero de gran importancia y utilidad para futuras practicas y aplicaciones.

Sinceramente,

---

Jorge Eliécer Duque

MSc. Ingeniería Electrónica

**Cartagena, Mayo de 2008**

**Señores:**

**UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR**

Comité de evaluación de proyectos

La ciudad

Respetados señores:

De la forma mas cordial y mucho agrado, nos permitimos presentar a ustedes la monografía titulada: “**LABORATORIO DE REDES INDUSTRIALES CON PLCs SIEMENS S7-200**” , para su estudio, consideración y aprobación, como requisito para obtener el titulo de ingeniero electrónico, además para la aprobación del Minor de Automatización Industrial.

En espera que se cumpla con las normas pertinentes establecidas por la institución.

Sinceramente,

---

Norberto Alonso Angel Cabarcas

Codigo: 00 04 065

---

Walberto Julio Cantillo Pajaro

Codigo: 00 04 024

## **AUTORIZACION**

Cartagena de Indias D. T. y C., Mayo del 2008

Nosotros, NORBERTO ALONSO ANGEL CABARCAS y WALBERTO JULIO CANTILLO PAJARO, identificados con numero de cedula 73.191.358 de Cartagena y 9.296.046 de Turbaco, autorizamos a la UNIVERCIDAD TECNOLOGICA DE BOLIVAR para hacer uso de nuestro trabajo de grado y publicarlo en el catalogo Online de la Biblioteca.

---

Norberto Alonso Angel Cabarcas  
c.c 73.191.358 de Cartagena

---

Walberto Julio Cantillo Pájaro  
c.c 9.296.046 de Turbaco

*A Dios por su infinita bondad y Amor incondicional.*

*A mi madre por que siempre me dio fuerzas para seguir adelante.*

*A mi padre, ese hombre que todo me ha dado que casi de sol a sol a trabajado, que su único afán es sacarme a mi adelante, que me a dado todo su apoyo, que es lo más importante. Al cual agradezco todo su amor, bondad y su mal genio. Que en los momentos más difíciles además de ser un padre es un amigo. Gracias por los consejos que me has dado y que no olvidare jamás, a ti es a quien tengo que agradecer todo esto y mucho más.*

*A mis hermanas las cuales me impulsaban con cada uno de sus logras a continuar para así ser cada vez mejor.*

*Nunca podré pagarles todo lo que han hecho por mi.*

*Norberto Alonso Angel Cabarcas.*

**Nota de aceptación**

---

---

---

---

---

---

**Presidente del jurado**

---

**Jurado**

---

**Jurado**

## Tabla de contenidos:

1	Introducción.....	1
2.	Comunicación en redes de PLCs con S7-200.....	2
	Protocolo PPI (Interfaz punto a punto).....	2
2.2	Protocolo Modbus.....	3
3.	Implementación de redes de PLCs Siemens S7-200.....	5
3.1	Red PPI.....	5
	3.1.1 Bloques de funciones.....	6
	3.1.1.1 NETR.....	6
	3.1.1.2 NETW.....	7
	3.1.2 Cables y conectores.....	7
	3.1.2.1 Cables.....	8
	3.1.2.2 Conectores.....	8
	3.1.2.3 Circuitos para los conectores de bus.....	9
3.2	Red Modbus.....	11
	3.2.1 Librerías Modbus.....	12
	3.2.1.1 MBUS_INIT.....	14
	3.2.1.2 MBUS_SLAVE.....	17
	3.2.2 Cables y conectores.....	18
4.	Prácticas de laboratorio.....	19
4.1	Practica N.1: Comunicación Maestro – Esclavo en una red PPI.....	19



4.1.1	Objetivos.....	19
4.1.2	Ejercicio.....	19
4.1.3	Procedimiento.....	19
4.1.3.1	Comprobación de comunicaciones.....	20
4.1.3.2	Selección del modo Maestro/Esclavo y contenido del búfer de Lectura/Escritura.....	22
4.1.3.3	Creación del programa de la práctica.....	23
4.1.3.3.1	Tabla de lectura.....	24
4.1.3.3.2	Tabla de escritura.....	28
4.1.3.3.3	Creación del programa esclavo.....	29
4.2	Practica N.2: Comunicación Maestro – Esclavo en una Red Modbus.....	31
4.2.2	Objetivos.....	31
4.2.3	Ejercicio.....	31
4.2.4	Procedimiento.....	31
4.2.4.1	Modbus Poll.....	31
4.2.4.2	Programa Modbus esclavo.....	34
4.3	Practicas de Redes propuestas.....	41
5.	Conclusiones.....	44
6.	Bibliografía.....	45
6.1	Anexo A: Marcas especiales.....	46
6.2	Anexo B: Bloques de funciones; Movimientos, creación de tablas.....	49
6.3	Anexo C. Punteros.....	52

## Lista de figuras.

Figura 1. Red Token Ring para el protocolo PPI de Siemens.....	3
Figura 2. Configuración de una red Modbus con PLCs S7-200.....	4
Figura 3. Protocolo PPI, conexión en bus, estrella lógica.....	5
Figura 4. Árbol de funciones NETR y NETW.....	6
Figura 5. Bloque NETR.....	6
Figura 6. Bloque NETW.....	7
Figura 7. Cables y conectores de los PLCs S7-200.....	7
Figura 8. Conexión de los hilos al RJ45.....	9
Figura 9. Circuito de conexión de las cajas de final de línea o circuito.....	10
Figura 10. Circuito de conexión de las cajas de interconexión o intermedias.....	11
Figura 11. Conexión Red Modbus PLCs esclavos, PC maestro.....	12
Figura 12. Librerías Free Port y Modbus Slave.....	13
Figura 13. Bloque MBUS_INIT.....	17
Figura 14. Bloque MBUS_SLAVE.....	18
Figura 15. Conexión de dos PLCs, uno maestro y otro esclavo.....	20
Figura 16. Configuración de los puertos de comunicación.....	21
Figura 17. Ventana de comunicación, mostrando dos PLCs conectados a la red.....	21
Figura 18. Habilitación del modo PPI maestro.....	22
Figura 19. Creación de las tablas y activación del modo maestro PPI.....	24
Figura 20. Escritura de la dirección del PLC esclavo.....	25
Figura 21. Escritura del puntero de lectura.....	26

Figura 22. Determinación de la longitud de datos.....	26
Figura 23. Llenado de la tabla de lectura a nivel de programación.....	27
Figura 24. Movimientos del Byte de lectura al registro de salida.....	27
Figura 25. Llenado de tabla de escritura.....	29
Figura 26. Programa esclavo.....	30
Figura 27. Ventana de trabajo del software Modbus Poll.....	32
Figura 28. Definición de la lectura y escritura en Modbus Poll.....	32
Figura 29. Escritura en las Bobinas o salidas del PLC desde Modbus Poll.....	33
Figura 30. Escritura en los Registros del PLC desde Modbus Poll.....	33
Figura 31. Inicio de la programación Modbus esclavo.....	35
Figura 32. Parámetros de configuración de MBUS_INIT.....	36
Figura 33. Ventana de resultados – Errores falta de memoria.....	36
Figura 34. Árbol del bloque de programa, Librería.....	37
Figura 35. Ventana de asignación de memoria.....	37
Figura 36. Ventana de resultados, compilado con 0 errores.....	38
Figura 37. Programación del Bloque MBUS_SLAVE.....	38
Figura 38. Programa Modbus esclavo.....	39
Figura 39. Ventana de Modbus Poll conectado a la red.....	40
Figura 40. Esquema de un semáforo, para peatones y vehículos.....	42
Figura 41. Esquema del garaje, con control de capacidad.....	43

## Lista de tablas

Tabla 1; Funciones mas usadas en Protocolo Modbus.....	4
Tabla 2; Conexión Del conector DB9 Para las redes PPI y ModBus.....	10
Tabla 3; Errores de ejecución.....	16
Tabla 4; Tabla de posicionamiento de bytes para la lectura o escritura.....	23
Tabla 5; Tabla de lectura del maestro PPI.....	25
Tabla 6; Tabla de escritura PPI.....	28

## **1. Introducción**

En el campo de la automatización industrial y las comunicaciones las redes industriales están en constante avance, ya que la evolución tecnológica nunca cesa, y cada día adquieren mayor importancia en el sector debido a la necesidad de comunicar entre sí a todos los instrumentos y los equipos de control de una planta.

El desarrollo tecnológico experimentado por la sociedad en las últimas décadas, impone un gran desafío a los profesionales que trabajan en el área de redes y comunicaciones para convertirse en agentes de cambio, aceptando los retos impuestos por la vertiginosa renovación tecnológica, permitiendo que las organizaciones hagan uso adecuado de los recursos de hardware y software, para gestionar la información y a la vez proyectarse con eficiencia en un ambiente altamente competitivo y globalizado.

En el área de las redes industriales nos topamos con los buses de campos los cuales se caracterizan por contener los protocolos de comunicación entre los elementos y el controlador, ya que transmiten en un mismo medio físico de forma bidireccional la información. Estos también designados a la comunicación para el control de redes industriales, interconectando computadores, PLC, sensores, actuadores y demás equipos dentro de la jerarquía de la red del proceso.

Entre los protocolos contenidos en los buses de campo podemos encontrar protocolos abiertos y protocolos tipo propietario, estos últimos son de uso exclusivo de un fabricante específico. Los protocolos de redes industriales se clasifican ya sean por su topología, medio físico, velocidad de transferencia, entre otros. Estas características se pueden observar en el anexo D.

Por ello la Universidad Tecnológica de Bolívar, implementa un laboratorio de redes industriales, el cual trabaja con los PLCs S7- 200 de Siemens, que se encuentran disponibles en los laboratorios de dicha entidad, con estos PLCs se crean redes basadas en los protocolos PPI y Modbus.

Los laboratorios contenidos en este trabajo serán realizados de manera didáctica para que el lector comprenda su funcionamiento y pueda realizar experiencias de mayor grado de complejidad después de haber concluido la lectura y comprendido el contenido de este.

## **2. Comunicación en redes de PLCs con S7-200**

Una red industrial con PLCs consiste en un conjunto de PLCs interconectados entre si por medio de cables y que se comunican a través de un protocolo común. Las redes industriales manejan diferentes tipos de protocolos de comunicación los cuales dependen de su utilización, facilidad de manejo y velocidad de transmisión. Específicamente, en los PLCs S7-200 de Siemens se pueden utilizar varios tipos de protocolos, de los cuales se utilizaron para este trabajo: PPI (Punto a punto) y Modbus.

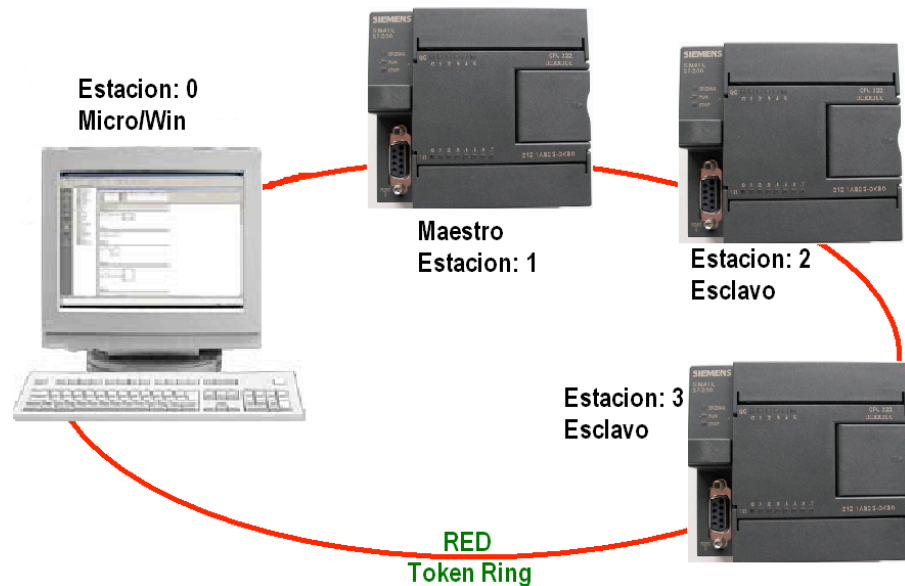
El protocolo PPI o interfaz punto a punto, se usa para la programación y comunicación de los PLCs Siemens S7-200 con un computador. Esta interfaz es de uso exclusivo para los S7-200 ya que otros PLCs Siemens utilizan protocolos e interfaces diferentes.

### **2.1 Protocolo PPI (Interfaz punto a punto)**

El protocolo PPI es un protocolo maestro-esclavo. Este protocolo se basa en las redes token ring (redes de anillo con testigo circulante), lo que quiere decir que el maestro pasa el control de la red haciendo circular un token (o testigo) en el orden de las direcciones de las estaciones que componen la red. El token le permite a una estación transmitir datos. Una sola estación a la vez puede tener el token en su poder. Los esclavos no pueden tener nunca el token en su poder, por lo que no tienen permiso de transmitir datos. En otras palabras, los esclavos no pueden iniciar mensajes, sino que deben esperar a que un maestro les envíe una petición o que les solicite una respuesta, como se muestra en la figura 1. En el protocolo

PPI no se limitan el número de maestros que se pueden comunicar con un mismo esclavo. Sin embargo la red no puede comprender más de 32 maestros.

Figura 1. Red Token Ring para el protocolo PPI de Siemens

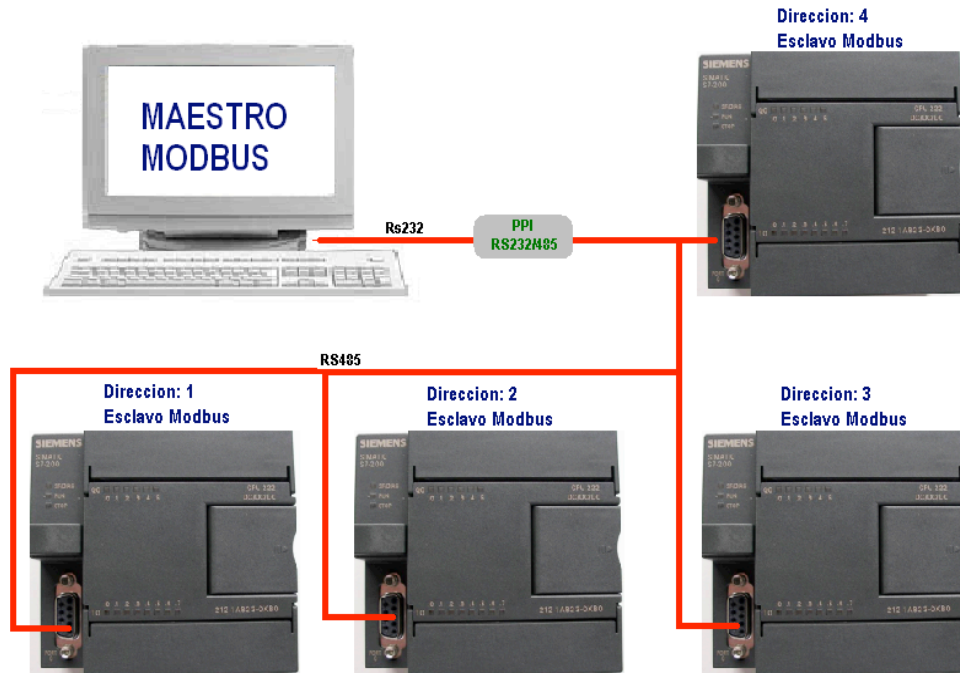


## 2.2 Protocolo Modbus

Este protocolo fue desarrollado originalmente por Modicon para la comunicación entre PLCs y actualmente está soportado por la organización Modbus-IDA. Debido a su simplicidad y especificación abierta, actualmente está ampliamente difundido y es utilizado por diferentes fabricantes en todo el mundo. Entre los dispositivos que utilizan dicho protocolo se encuentran PLCs, terminales HMI, RTUs (**R**emote **T**ransmisión **U**nit), drivers, sensores y actuadores remotos. Existen dos variantes del protocolo Modbus: ASCII (*American Standard Code for Information Interchange*) y RTU (*R*emote *T*ransmisión *U*nit). En esta monografía se trabajara con el protocolo Modbus RTU, para el cual Siemens tiene unas librerías prediseñadas, las cuales serán descritas mas adelante.



Figura 2. Configuración de una red Modbus con PLCs S7-200



En los PLCs s7-200 las librerías Modbus soportan las funciones en modo esclavo mostradas en la tabla 1.

Tabla 1. Funciones mas usadas en Protocolo Modbus

Código	Nombre
01	Leer status de bobinas
02	Leer status de entradas
03	Leer registros de retención
04	Leer registros de entrada
05	Forzar una bobina
06	Seleccionar una bobina
15	Forzar varias bobinas
16	Reset múltiples registros

### 3. Implementación de redes de PLCs Siemens S7-200.

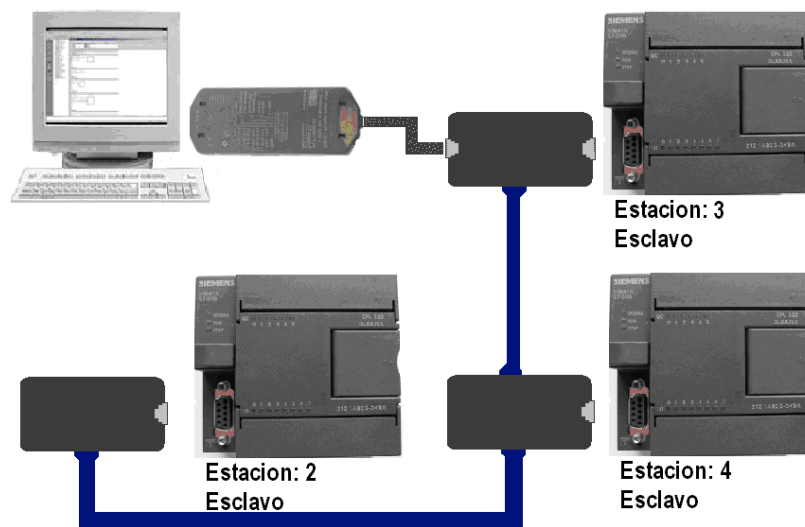
Para la implementación de redes industriales con PLCs S7-200 y protocolos PPI y ModBus se debe tener en cuenta las características físicas de la red y las de la interfaz de usuario; También se debe tener en cuenta, que ofrece una empresa como Siemens para una red ModBus, que es una interfaz abierta de otra compañía como Modicom, y sacar el mejor provecho a estas redes.

#### 3.1 Red PPI.

El protocolo PPI como se menciona anteriormente es el protocolo utilizado por Siemens para la programación de los PLC de la familia S7-200, por ende, su configuración es un poco sencilla y fácil de manipular en este tipo de PLCs.

Así como se muestra en la figura 3 se conectan los PLCs en bus pero con una arquitectura de red lógica tipo anillo para cumplir con el estándar del Token Ring.

Figura 3. Protocolo PPI, conexión en bus, estrella lógica



### 3.1.1 Bloques de funciones:

Los bloques de funciones utilizados por el protocolo PPI son el NETW (Escribir en la Red) y NETR (Leer en la Red). Estos bloques de funciones se encuentran anclados en el árbol de operaciones así como se muestra en la figura 4.

Figura 4. Árbol de funciones NETR y NETW



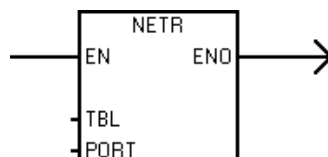
Las características de estos bloques se dan a continuación.

#### 3.1.1.1 NETR

La operación Leer de la red (NETR) inicia una comunicación para registrar datos de una estación remota a través del puerto indicado (PORT), según se define en la tabla (TBL).

Con la operación NETR se pueden leer hasta 16 bytes de información de una estación remota. El bloque de esta función es mostrado en la figura 5

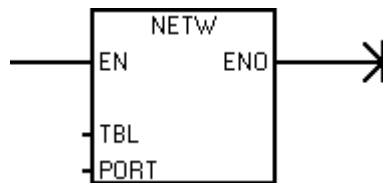
Figura 5; Bloque NETR



### 3.1.1.2 NETW

La operación Escribir en la red (NETW) inicia una comunicación para escribir datos en una estación remota a través de la interfaz indicada (PORT), según se define en la tabla (TBL). Con la operación NETW se pueden escribir hasta 16 bytes de información en una estación remota. El bloque de esta función es mostrado en la figura 6.

Figura 6; Bloque NETW



*Nota:* El programa puede contener un número cualquiera de operaciones NETR/NETW, pero sólo ocho de ellas (en total) pueden estar activadas simultáneamente.

### 3.1.2 Cables y conectores

Los cables y conectores juegan un papel muy importante en toda red ya que gracias a estos se realiza la conexión a nivel físico o hardware de la red así como se observa en la figura 7.

Figura 7; Cables y conectores de los PLCs S7-200



### **3.1.2.1 Cables**

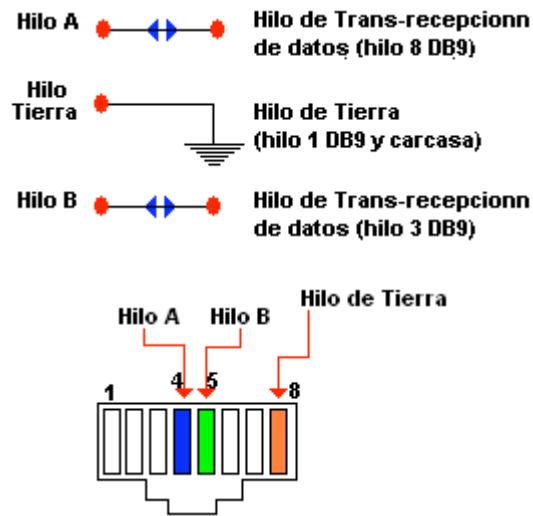
Los cables que se deben usar en una red industrial deben cumplir con ciertas características como son la impedancia, capacitancia, atenuación, alma del cable, diámetro del cable, entre otras. Todas estas características son tomadas dependiendo del tipo de datos y velocidad de transmisión a la que se desee trabajar. En el caso de este trabajo, como hay que cumplir con las especificaciones del protocolo PPI y Modbus se trabajara con cable UTP categoría 5 en adelante de 8 (conectores de red) y 9 Hilos (conectores DB9), todos estos cables son uno a uno.

### **3.1.2.2 Conectores**

Entre los conectores se encuentra el DB9 o conector de 9 pines, el cual es usado para conectarse al puerto 0 del PLC, dicho conector es usado por el cable PPI para conectar el PLC al PC y para conectar el PLC a los conectores de bus. También se encuentran conectores tipo RJ45, los cuales fueron seleccionados por su fácil manejo e implementación, también se seleccionaron estos conectores por su impedancia, que concuerda con la utilizada para el protocolo PPI y Modbus.

Para la conexión de estos se utiliza la topología de conexión mostrada en la figura 8, esta conexión se selecciona de acuerdo con el estándar EIA/TIA-485, para dos hilos Modbus en el conector RJ45

Figura 8. Conexión de los hilos al RJ45

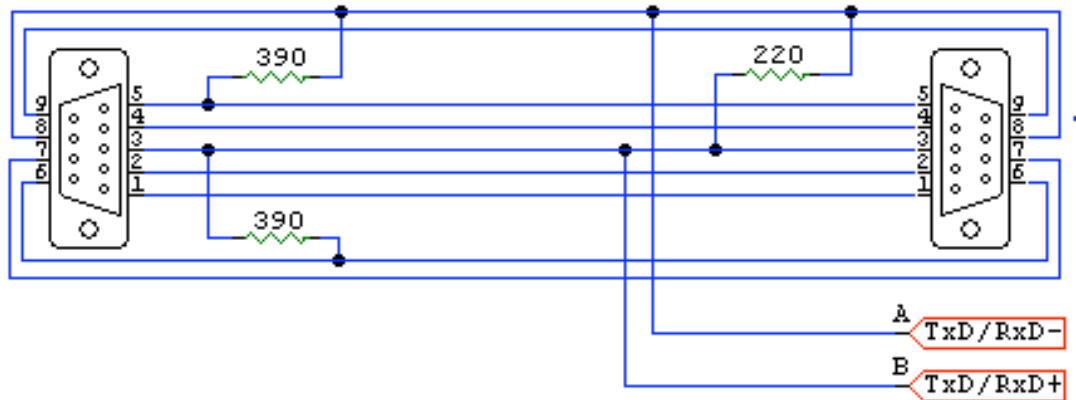


En donde se observa la conexión de los conectores y como van los hilos en dicho conector. Esto es para la caja de los conectores de bus y no para los cables ya que estos están conectados uno a uno.

### 3.1.2.3 Circuitos para los conectores de bus.

Los conectores de bus, son los encargados de enlazar los diferentes dispositivos de la red a través de los cables de conexión, indicar las posiciones de los PLCs en una red, y de definir cual es el punto de inicio de la red y el final de la misma. Dependiendo de su polarización y circuito de conexión como se observa en la figura 9 y 10.

Figura 9; Circuito de conexión de las cajas de final de línea o circuito



Este circuito indica la asignación de pines y su número en cada uno de los extremos, los cables “+TxD/RxD” y “-TxD/RxD” son los cables que van conectados a las cajas o conectores subsiguientes, y el pin uno es el apantallamiento o tierra, igual que la carcasa del conector, cada uno de estos pines tiene una función la cual se describe en la tabla 2.

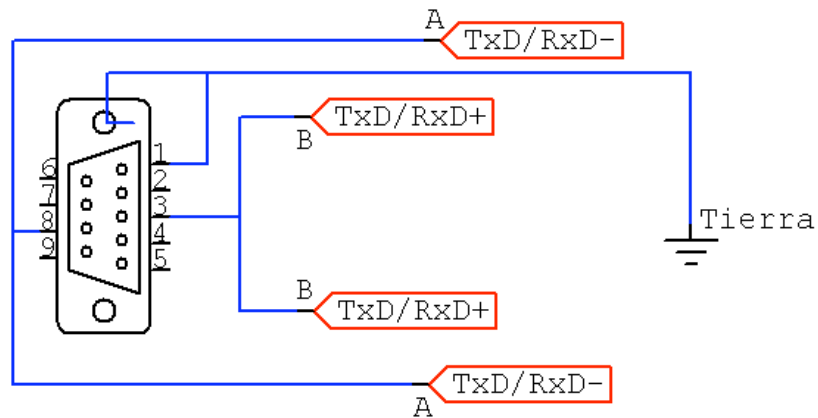
Tabla 2; Conexión Del conector DB9 Para las redes PPI y ModBus

Nº de Pin	Red PPI o Protocolo Modbus	Conector DB 9 o de 9 pines
1	Tierra	
2	Hilo logico	
3	Señal B RS-485	
4	RTS (TTL)	
5	Hilo logico	
6	+5 V, 100 Ω resistor en serie	
7	+24 V	
8	Señal A RS-485	
9	Selección protocolo de 10 bits (entrada)	
Carcasa del conector	Tierra	

Para conectar los PLCs intermedios se debe tener en cuenta que los conectores de bus

tienen una topología de circuito más sencilla. Y esta es la mostrada en la figura 10, la cual no tiene polarización.

Figura 10; Circuito de conexión de las cajas de interconexión o intermedias



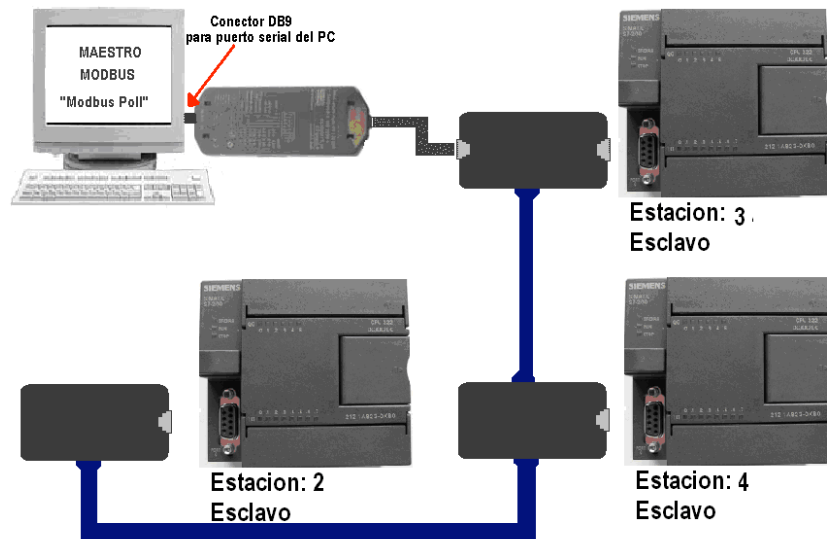
Estos mismos conectores, poseen un cable conector colocado para mayor facilidad de conexión a los PLCs, el cual tiene un conector DB9 macho en su parte final, que se conecta al PLC.

### 3.2 Red Modbus

El protocolo Modbus es un protocolo utilizado por Siemens para la comunicación con otros dispositivos Modbus o para conectarse a una red Modbus con un computador para el caso específico de los S7-200; CPU222 se utiliza un simulador de un dispositivo Modbus maestro en el PC y los PLC se configuran como esclavos, así como se ve en la figura 11



Figura 11; Conexión Red Modbus PLCs esclavos, PC maestro



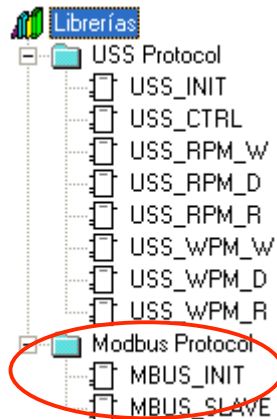
Para la utilización de dicho protocolo en los PLCs siemens S7-200 es necesario tener en cuenta ciertos parámetros como son:

- El protocolo ModBus es un protocolo abierto pero no es Siemens su creador. Por ende con el programa básico Step7/MicroWin no se pueden crear redes ModBus fácilmente.
- Siemens ofrece la librería Modbus (6ES7 8302BC000YX0) la cual facilita la programación en este protocolo.

### 3.2.1 Librerías Modbus

La librería "Modbus Protocol" la cual se adquiere por separado, se instala introduciendo el CD que contiene dicha librería y automáticamente el Software la reconoce como se ve en la figura 12. Donde las operaciones Modbus se instalan en la carpeta Librerías.

Figura 12; Librerías Free Port y Modbus Slave



Estas operaciones permiten que el S7-200 actúe de dispositivo Modbus. Cuando se dispone una operación Modbus en el programa, se agregan automáticamente al proyecto una o más subrutinas asociadas, que facilitan la comunicación con los maestros y esclavos Modbus.

Para la comunicación mediante el protocolo modbus se debe tener en cuenta los siguientes parámetros o requisitos:

- La inicialización del protocolo para esclavos Modbus utiliza el puerto 0 para la comunicación Modbus.
- Si el puerto 0 se está utilizando para la comunicación vía el protocolo de esclavos Modbus, no se podrá usar para ninguna otra función, incluyendo la comunicación con Step7/MicroWin.
- Las operaciones del protocolo para esclavos Modbus afectan a todas las direcciones de marcas especiales (SM) asociadas a la comunicación Freeport por el puerto 0.
- Las operaciones del protocolo para esclavos Modbus utilizan 3 subrutinas y 2 rutinas de interrupción.

- Las operaciones del protocolo para esclavos Modbus necesitan 1857 bytes de espacio en el programa para las dos operaciones Modbus y las rutinas de soporte.
- Las variables de las operaciones del protocolo de esclavos Modbus necesitan un bloque de 779 bytes de la memoria V. El usuario asigna la dirección inicial de este bloque, que se reserva para las variables Modbus.

**Nota:** Para conmutar el puerto de comunicación de la CPU a modo PPI después de haber activado el modo Modbus, de manera que pueda comunicarse con STEP 7--Micro/WIN, utilice uno de los métodos siguientes:

- Active una entrada la cual coloque el PLC en modo stop (Eje: I0.0→(stop)).
- Coloque el selector de modo del S7--200 en posición STOP.

Cualquiera de estos dos métodos conmuta el puerto de la CPU para poder establecer la comunicación con STEP 7--Micro/WIN.

Para la programación de un esclavo ModBus es necesario tener en cuenta los siguientes bloques:

### **3.2.1.1 MBUS\_INIT**

Esta operación sirve para activar e inicializar, o bien para desactivar la comunicación Modbus. MBUS\_INIT se deberá ejecutar sin errores antes de poder utilizar la operación MBUS\_SLAVE. La operación se finaliza y el bit “done” se pone a 1 inmediatamente, antes de continuar con la siguiente operación. La operación se ejecuta en cada ciclo cuando está activada la entrada “EN”. Ejecute la operación MBUS\_INIT sólo una vez por cada cambio

del estado de la comunicación. Por tanto, la entrada “EN” deberá permanecer activada al detectarse un flanco positivo, o bien ejecutarse sólo en el primer ciclo.

El valor de la entrada “Mode” selecciona el protocolo de comunicación. Si la entrada tiene el valor “1”, el puerto 0 se asignará al protocolo Modbus y se habilitará el protocolo. Si la entrada tiene el valor “0”, el puerto 0 se asignará a PPI y se inhibirá el protocolo Modbus. El parámetro “Baud” permite ajustar la velocidad de transferencia a 1200, 2400, 4800, 9600, 19200, 38400, 57600, o bien a 115200 bit/s. El parámetro “Addr” ajusta la dirección a valores comprendidos entre 1 y 247.

El parámetro “Parity” se ajusta de manera que concuerde con la paridad del maestro ModBus. Se aceptan los valores siguientes:

- 0 Sin paridad
- 1 Paridad impar
- 2 Paridad par

El parámetro “Delay” retarda el timeout de fin de mensaje ModBus estándar, sumando el número indicado de milisegundos al timeout de mensajes ModBus estándar. En redes cableadas, el valor típico de este parámetro debería ser “0”.

El parámetro “MaxIQ” ajusta el número de entradas (I) y salidas (Q) disponibles para las direcciones Modbus 00xxxx y 01xxxx a valores comprendidos entre 0 y 128. Un valor de “0” inhibe todas las lecturas y escrituras de las entradas y salidas. Es recomendable ajustar el valor de “MaxIQ” a 128, con objeto de poder acceder a todas las entradas y salidas del S7200.

El parámetro “MaxAI” ajusta el número de registros de entradas analógicas (AI) disponibles para la dirección Modbus 03xxx a valores comprendidos entre 0 y 32. Un valor

de “0” inhibe la lectura de las entradas analógicas. Para poder acceder a todas las entradas analógicas del S7200, es recomendable ajustar el valor de “MaxAI” de la manera siguiente:

16 para la CPU 222

El parámetro “MaxHold” ajusta el número de registros de retención en la memoria V disponibles para la dirección Modbus 04xxx. Por ejemplo, para que el maestro pueda acceder a 2000 bytes de la memoria V, ajuste el valor de “MaxHold” a 1000 palabras (registros de retención).

El parámetro “HoldStart” es la dirección inicial de los registros de retención en la memoria V. Por lo general, este valor se ajusta a VB0, de manera que el parámetro “HoldStart” se ajuste a &VB0 (dirección de VB0). Como dirección inicial de los registros de retención en la memoria V se puede indicar también una dirección diferente, con objeto de poder utilizar VB0 en otra parte del proyecto. El maestro Modbus tiene acceso al número “MaxHold” de palabras de la memoria V, comenzando en “HoldStart”.

La salida “Done” se activa una vez finalizada la operación MBUS\_INIT. El byte de salida “Error” contiene el resultado de ejecución de la operación. Donde los errores se siguen de la tabla 3:

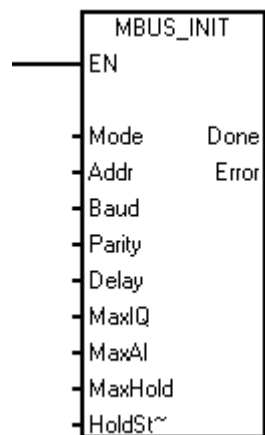
Tabla 3; Errores de ejecución.

ERROR	Descripción
-------	-------------

0	No se ha presentado ningún error
1	Error de rango de memoria
2	Velocidad de transferencia o paridad no valida
3	Dirección de esclavo no valida
4	Valor no valido para un parámetro ModBus
5	Los registros de retención ocultan los símbolos de los esclavos ModBus
6	Error de paridad en recepción
7	Error CRC de recepción
8	Petición no valida/ función no soportada
9	Dirección no valida en una petición
10	Función de esclavo no habilitada

El bloque de esta función se muestra en la figura 13.

Figura 13; Bloque MBUS\_INIT



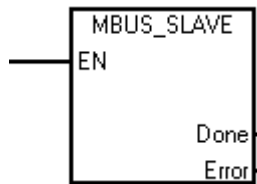
Donde cada uno de los parámetros se explicaron anteriormente.

### 3.2.1.2 MBUS\_SLAVE

Esta operación se utiliza para procesar una petición del maestro Modbus, debiendo ejecutar en cada ciclo para poder comprobar y responder a las peticiones Modbus. La operación se ejecuta en cada ciclo cuando está activada la entrada EN. MBUS\_SLAVE no tiene parámetros de entrada. La salida “Done” se activa cuando la operación MBUS\_SLAVE responde a una petición Modbus y se desactiva si no se ha procesado ninguna petición. La

salida “Error” contiene el resultado de ejecución de la operación. Esta salida sólo será válida si está activada la salida “Done”. Si “Done” está desactivada, no cambiará el parámetro de error. El bloque de esta función es el mostrado en la figura 14.

Figura 14; Bloque MBUS\_SLAVE



### 3.2.2 Cables y conectores

Para la utilización de los cables y conectores, se usan las mismas características físicas que se utilizan para el protocolo PPI, los mismos conectores y la misma topología de la red, teniendo el mismo anillo lógico conectado en un bus físico, como se explico en la sección 3.1.2.

## 4. Prácticas de laboratorio:

#### **4.1 Practica N.1: Comunicación Maestro – Esclavo en una red PPI**

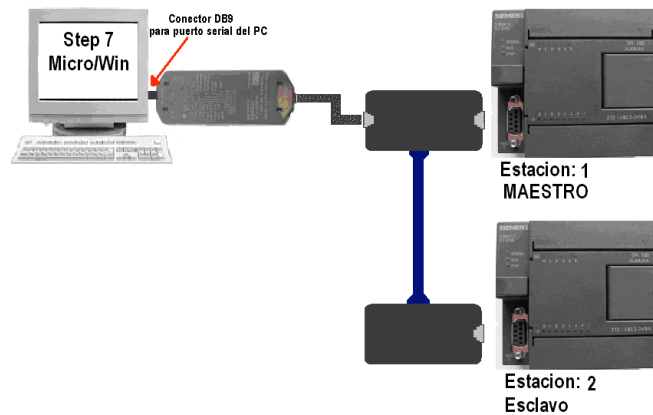
**4.1.1 Objetivos:** Conocer el funcionamiento Básico-Practico de las redes PPI de Siemens y sus funciones de lectura y escritura en una red de los registros de entradas y salidas.

**4.1.2 Ejercicio:** Activar y desactivar las salidas de un PLC esclavo con las entradas de un PLC maestro. Y activar y desactivar las salidas del PLC maestro leyendo las entradas del PLC esclavo.

4. **Procedimiento:** Es muy importante tener en cuenta las conexiones físicas en los conectores de bus a los PLCs y al cable PPI. Ya sean estos conectores de final de red o de conexión intermedia. Así como se mostró en las figuras 7 y 11, en este caso solo se van a conectar dos PLCs como se muestra en la figura 15 y luego se procede a la comprobación de comunicaciones a nivel de software.

Figura 15; conexión de dos PLCs, uno maestro y otro esclavo





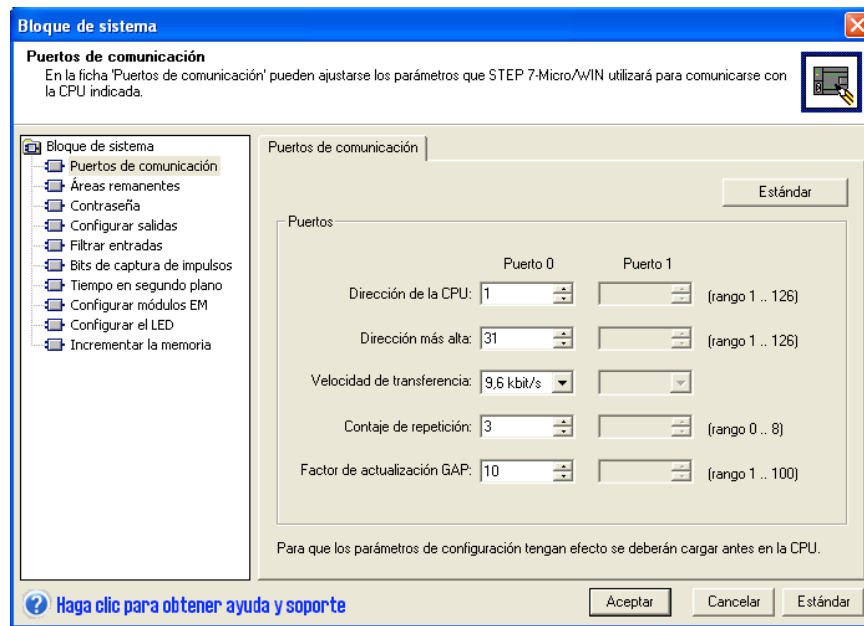
#### 4.1.3.1 Comprobación de comunicaciones

Para la comprobación de la conexión es necesario realizar un cambio de las direcciones remotas de los PLCs que se utilizan, puesto que estos traen por default (configuración de fábrica) la dirección remota 2 y si no se cambia, todos los PLCs tendrán la misma dirección y responderán a las mismas peticiones, lo que puede ocasionar problemas de comunicación.

Para evitar eso ello se realizan los pasos que siguientes:

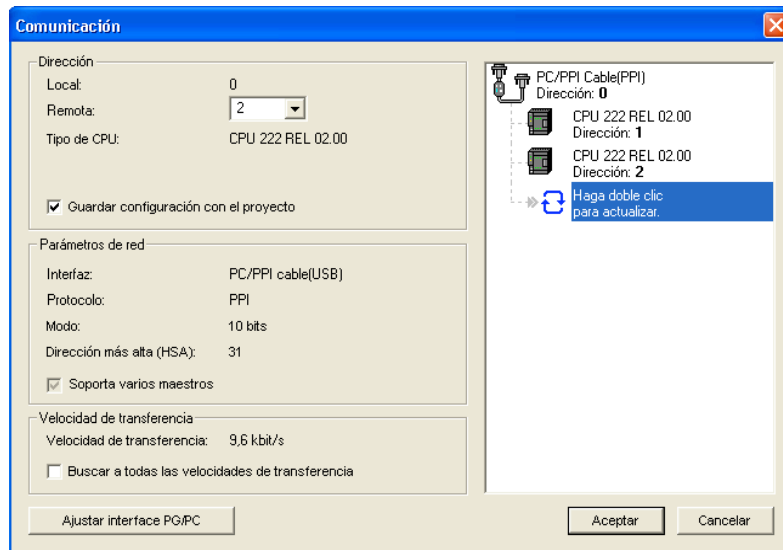
1. Se realiza el cambio de dirección en el árbol de funciones como se muestra en la figura 16, y luego se cargan los datos al PLC para así tener las direcciones nuevas en cada uno de los PLCs que se estén usando, en este caso la dirección 1 y la dirección 2. Estos cambios se realizan en la ventana de Bloques de sistemas (ver figura 16) que se encuentra en el árbol de comunicaciones.

Figura 16; Configuración de los puertos de comunicación



- Se realiza la comprobación de comunicación en donde deberían salir todos los PLCs que se tengan conectados con sus respectivas direcciones remotas. Así como se muestra en la figura 17 para 2 PLCs

Figura 17; Ventana de comunicación, mostrando dos PLCs conectados a la red.



#### 4.1.4.2 Selección del modo Maestro/Esclavo y contenido del búfer de Lectura/Escritura

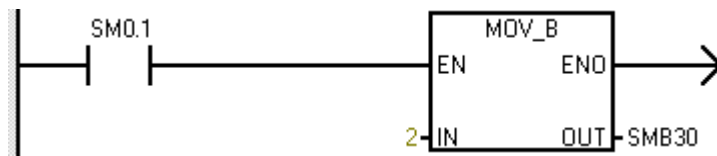
La selección de maestros y esclavos son posibles dependiendo de los datos almacenados en dos bits de marcas especiales SM30.0 y SM30.1 (Ver anexo A)

Valor bits 00 (Binario = 0 Decimal): modo PPI / esclavo

Valor bits 10 (Binario = 2 Decimal): modo PPI / maestro

Así una vez habilitado el modo maestro PPI (ver figura 18), la comunicación se establece por el PLC maestro haciendo uso de las instrucciones NETR y NETW (ver secciones 3.1.1.1 y 3.1.1.2).

Figura 18. Habilitación del modo PPI maestro



En la figura 18 se observa el uso de marcas especiales (ver anexo A) y el bloque de transferencia de bytes (ver anexo B), los cuales activan el modo PPI maestro.

Para realizar la comunicación con las instrucciones NETR y NETW, es preciso previamente definir 7 bytes de datos por instrucción, los cuales conforman los búfer de lectura y escritura, que se guardan en la memoria de la CPU maestra, donde aparece toda la información relativa a la comunicación que se desea establecer. La función de cada uno de estos bytes es la indicada en la tabla 4, esta tabla muestra cada una de las posiciones a

llenar cada vez que se desee implementar una de las instrucciones lectura o escritura así un PLC esclavo.

Tabla 4; Tabla de posicionamiento de bytes para la lectura o escritura

BYTE	CONTENIDO
0	Define códigos de error y estado de la comunicación. No hace falta programarse pero si se debe considerar.
1	Dirección de la estación remota: dirección de la CPU a cuyos datos se desea acceder.
2	Puntero indirecto a la zona de datos en la estación remota que se desea acceder (I, Q, M, S, V) para lectura o escritura.
3	
4	
5	
6	Cantidad de bytes a los que voy a acceder (leer o escribir) desde 0 a 16.
7 al 22	A partir de este byte comienzan a introducirse los datos con NETR o a copiar los datos con NETW.

Esta tabla puede comenzar en cualquier dirección del buffer de memoria, pero una vez definido el byte de inicio los demás quedan definidos a continuación.

Como se explico anteriormente se le cambiaran las direcciones a cada uno de estos PLCs, y así se tiene el maestro (dirección 1) y el esclavo (dirección 2), como se observo en la 16.

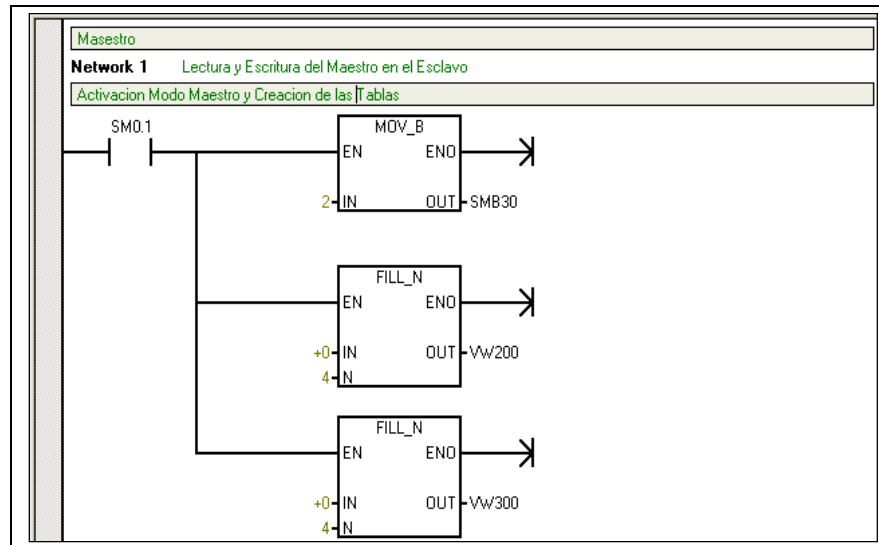
#### 4.1.4.2 Creación del programa de la práctica

Para ello se deben crear dos tablas en el PLC maestro, las cuales serán para la lectura y escritura como se menciona anteriormente.

Se debe recordar que las operaciones de lectura y escritura están definidas mediante 7 a 22 bytes de datos en la estación maestra, dependiendo de lo que se desee enviar o recibir, en este caso las tablas de lectura y escritura tendrá 8 bytes a partir de la dirección que se escoja, en este caso VB200 para lectura y VB300 para escritura (Variables escogidas

aleatoriamente). Esto se realiza como se muestra en la figura 19 en la cual se puede observar la activación del modo maestro y la creación de las tablas.

Figura 19; Creación de las tablas y activación del modo maestro PPI.



Para la creación de las tablas se usa la función FILL\_N (ver anexo B), la cual crea una tabla comenzando en el punto 0 y creando 4 palabra subsiguientes al byte de inicio (+0 → IN).

Para el llenado de dichas tablas se divide en llenado de tabla de lectura y de tabla de escritura.

#### 4.1.4.3 Tabla de lectura.

Para la tabla de lectura, la cual permite copiar y direccionar los registros que se leen, se realiza con la utilización de operaciones y funciones las cuales se encuentran en el árbol de funciones del programa Step7/MicroWin. La tabla 5 muestra detalladamente el contenido del buffer de lectura reservado en la memoria de la estación maestra.

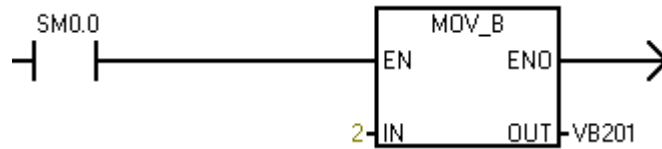
Tabla 5; Tabla de lectura del maestro PPI

Posición de la variable VB200 (consecutiva)	Descripción
VB300	Códigos de error no definidos
VB301	Dirección remota = 3
VB302	Puntero (ver anexo C) al área de datos en el PLC remoto = & IB0
VB303	
VB304	
VB305	
VB306	Longitud de datos = 1 byte
VB307	Lectura al PLC Esclavo

Como se puede observar en la tabla que se va a crear en el programa, los códigos de errores no definidos, los cuales no se colocan en el programa ya que no es necesario la visualización de ellos, la dirección remota, en la cual se escribe la dirección del PLC esclavo que se desea en este caso leer, la zona del puntero (ver anexo C) del área de datos, es la zona a la cual apuntar para la lectura de los datos, la longitud de los datos que en este caso es 1 puesto que las entradas solo tienen un byte de espacio en memoria, y la posición donde se ubica la lectura tomada del PLC remoto.

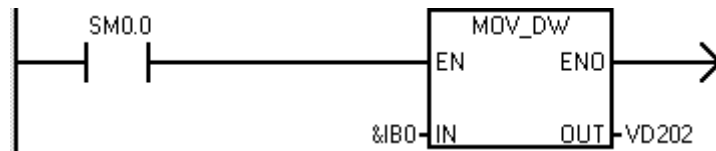
Luego de realizar la tabla, se procede a la programación del PLC. En donde el programa de llenado de la tabla de lectura, comienza con la posición de la tabla VB201 en la cual se coloca la dirección del PLC esclavo (En este caso 2), como se muestra en la figura 20. en donde se coloca una marca especial (ver anexo A) par que permanezca activada la transferencia de la dirección del PLC esclavo.

Figura 20. Escritura de la dirección del PLC esclavo



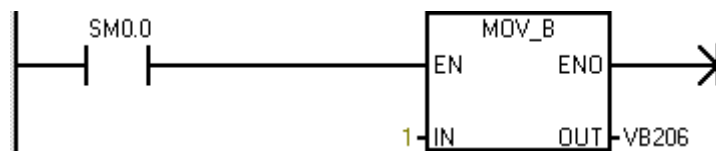
Para el llenado de las posiciones del puntero (ver anexo C), se realiza un movimiento de doble palabra, llenando los espacios seleccionado para esto en la tabla de lectura. Como se muestra en la figura 21, donde se muestra el uso del movimiento de doble palabra y la marca especial que lo activa en todo ciclo.

Figura 21. Escritura del puntero de lectura



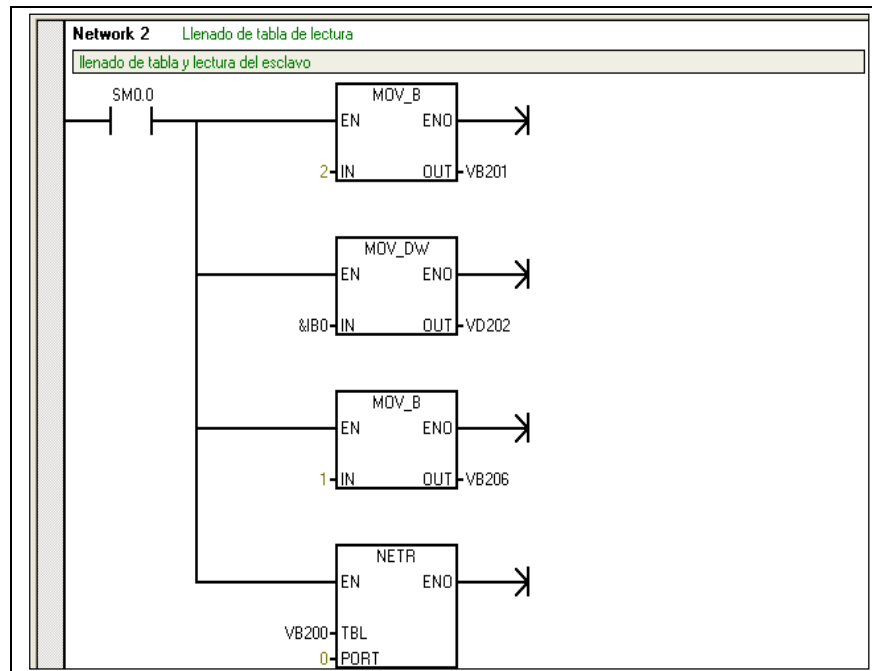
En la posición de “Longitud de datos” se debe colocar el numero de bytes que se van a utilizar como se muestra en la figura 22 y automáticamente se crean las posiciones de los bytes de uso, en este caso por ejemplo, la longitud de datos es uno ya que solo se usara un solo byte el del registro de las entradas, por ello se crea la posición VB207 en la cual se escribirá este registro.

Figura 22. Determinación de la longitud de datos



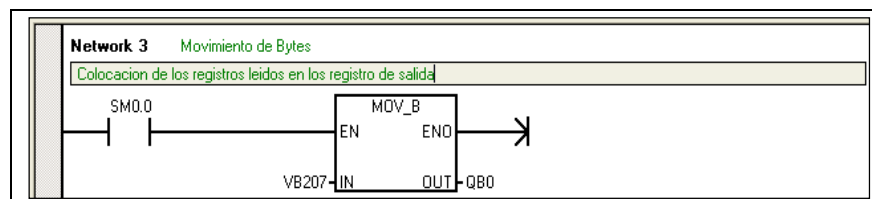
En la figura 23, se observa la implementación de todas estas funciones de red, usadas de manera secuencial para el llenado de la tabla de lectura

Figura 23; Llenado de la tabla de lectura a nivel de programación.



Después de leído el Byte que se guarda en la posición VB207, se direcciona adonde se desee escribir en este caso el registro de salidas del PLC maestro. Como se muestra en la figura 24 donde todo el Byte de entrada del PLC esclavo, es enviado a la salida.

Figura 24; Movimientos del Byte de lectura a el registro de salida.





#### 4.1.4.4 Tabla de escritura

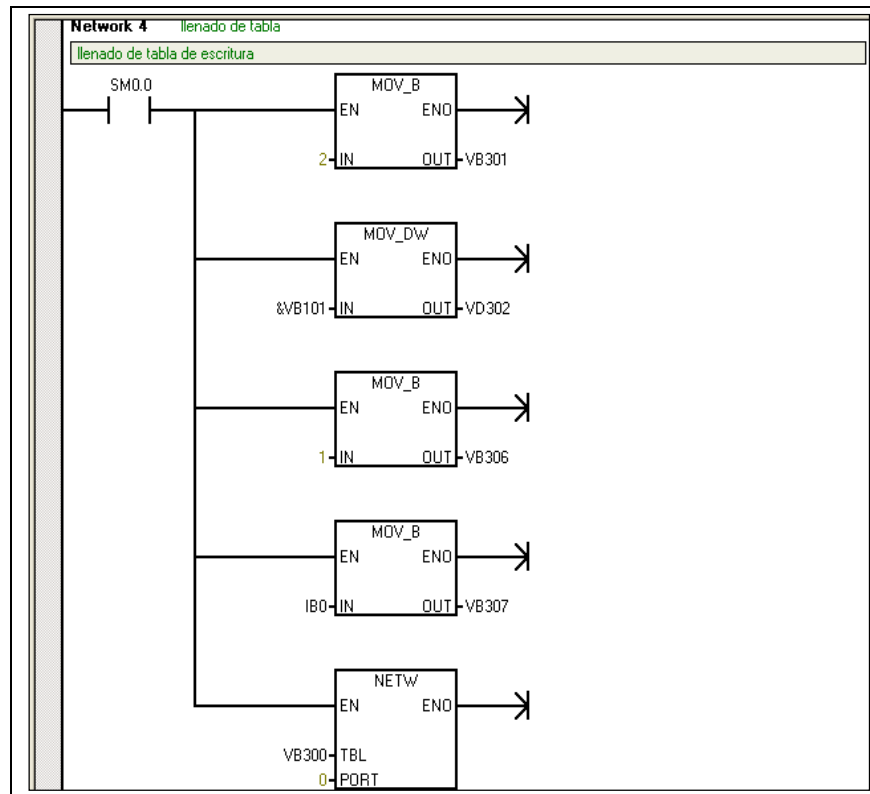
De igual manera como se realizo para la creación de la tabla de lectura se reserva el uso del búfer de escritura en la memoria del PLC maestro, se escriben los datos como se muestra en la tabla 6:

Tabla 6; Tabla de escritura PPI

Posición de la variable VB300 (consecutiva)	Descripción
VB300	Códigos de error no definidos
VB301	Dirección remota = 3
VB302	Puntero al área de datos en el PLC remoto = &VB101
VB303	
VB304	
VB305	
VB306	Longitud de datos = 1 byte
VB307	Escritura del PLC Maestro

Se realizan los mismos procedimientos que en la tabla de lectura con pequeñas variaciones como es la posición de la memoria del puntero, el movimiento del byte de escritura a la posición VB307 de la tabla. Así como se muestra en la figura 25.

Figura 25; Llenado de tabla de escritura.



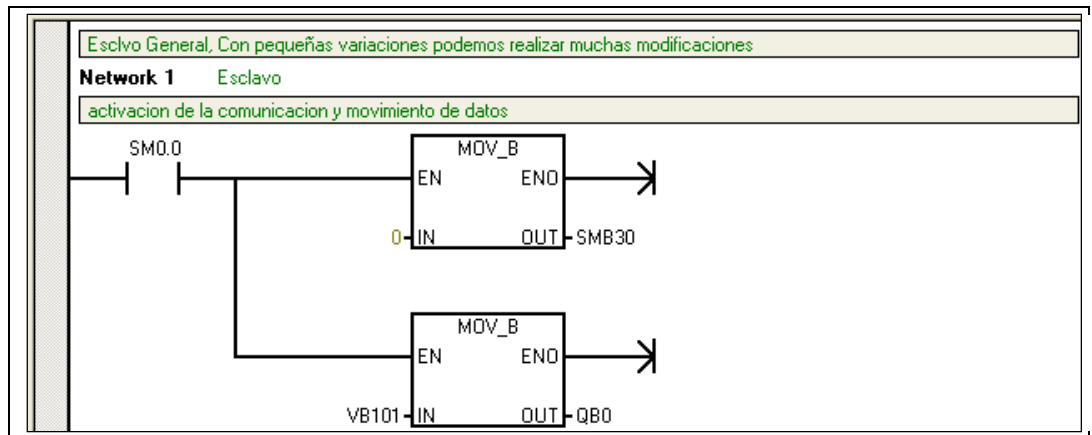
Como se ve en el llenado de esta tabla es muy parecido al de la tabla de lectura, por ello se omite la repetición de los pasos anteriormente explicados.

#### 4.1.4.5 Creación del programa esclavo

Este programa debe permitir la lectura y escritura de y en sus registros. Para ello solo se debe activar el protocolo PPI en modo esclavo, y luego se pasan los registros escritos en el a sus salidas de la manera como se muestra en la figura 21, ya que como se noto en el programa del maestro sus entradas fueron escritas en un registro de memoria (VB101), y de

hay pasan directamente al esclavo, donde en este son copiadas en el registro de salidas. Así como se muestra en la figura 26.

Figura 26; Programa esclavo.



Este programa con pocas modificaciones puede tener grandes variaciones, con las cuales se pueden generar grandes cambios en el programa.

## **4.2 Practica N.2: Comunicación Maestro – Esclavo en una Red Modbus**

**4.2.1 Objetivos:** Conocer el funcionamiento Básico-Practico de las redes Modbus con los PLCs Siemens s7-200 y sus funciones de esclavos Modbus y la simulación del maestro Modbus en el computador.

**4.2.2 Ejercicio:** Activar y desactivar las salidas de un PLC Esclavo Modbus, desde un computador con el software Modbus Poll y visualizar en este, el comportamiento de las entradas y las salidas.

### **4.2.2 Procedimiento**

Para la creación de una red Modbus con los PLCs s7-200; CPU222 se hace necesario la utilización de un software que simule un maestro Modbus, para lo cual se utilizara el “Modbus Poll”.

#### **4.2.2.1 Modbus Poll**

Modbus Poll es un simulador de un maestro Modbus, creado principalmente para ayudar a los desarrolladores de dispositivos Modbus esclavos como los s7-200. Ofrece la posibilidad de visualizar múltiples ventanas de trabajo (Ver figura 27) con las cuales se pueden supervisar varios esclavos Modbus y/o áreas de datos al mismo tiempo. Para cada ventana de trabajo basta con que se especifique: la función, la dirección, el tamaño y el tipo de petición para así poder leer y escribir registros y bobinas en cualquier ventana de trabajo como se muestran en las figuras 28, 29, 30.

Figura 27. Ventana de trabajo del software Modbus Poll

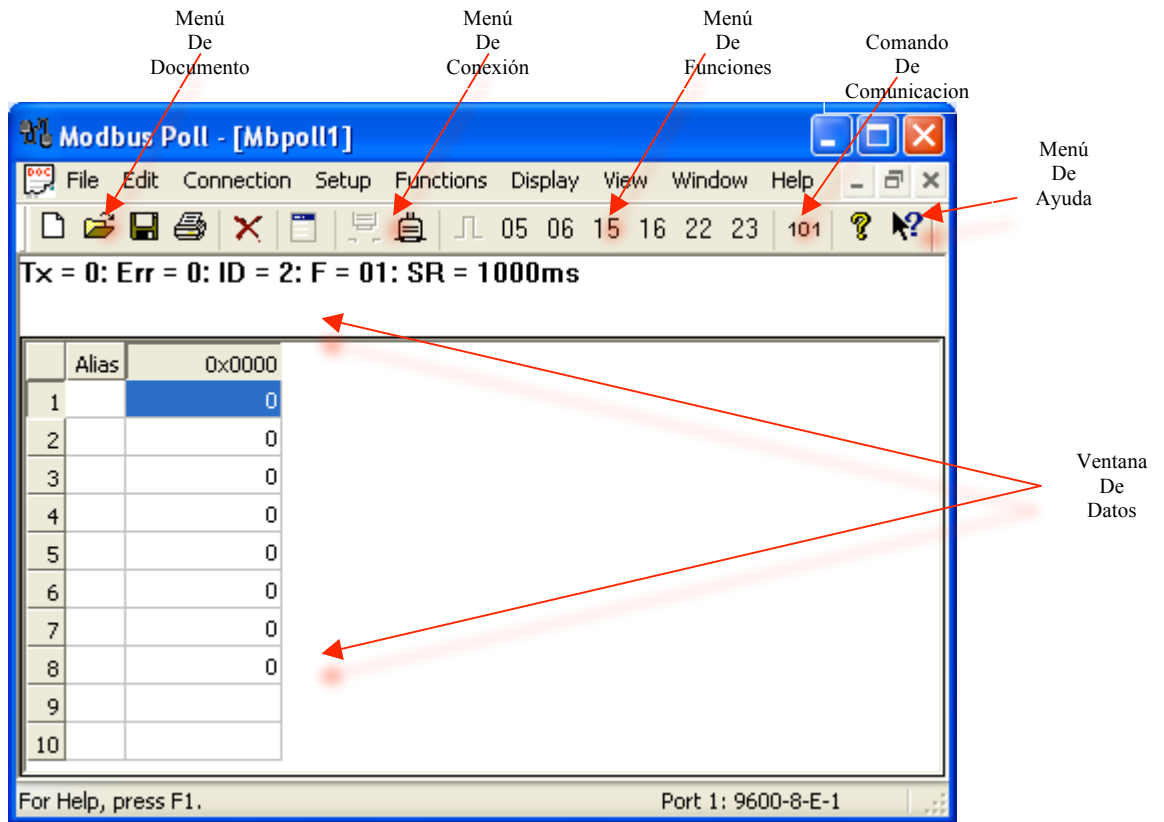


Figura 28. Definición de la lectura y escritura en Modbus Poll

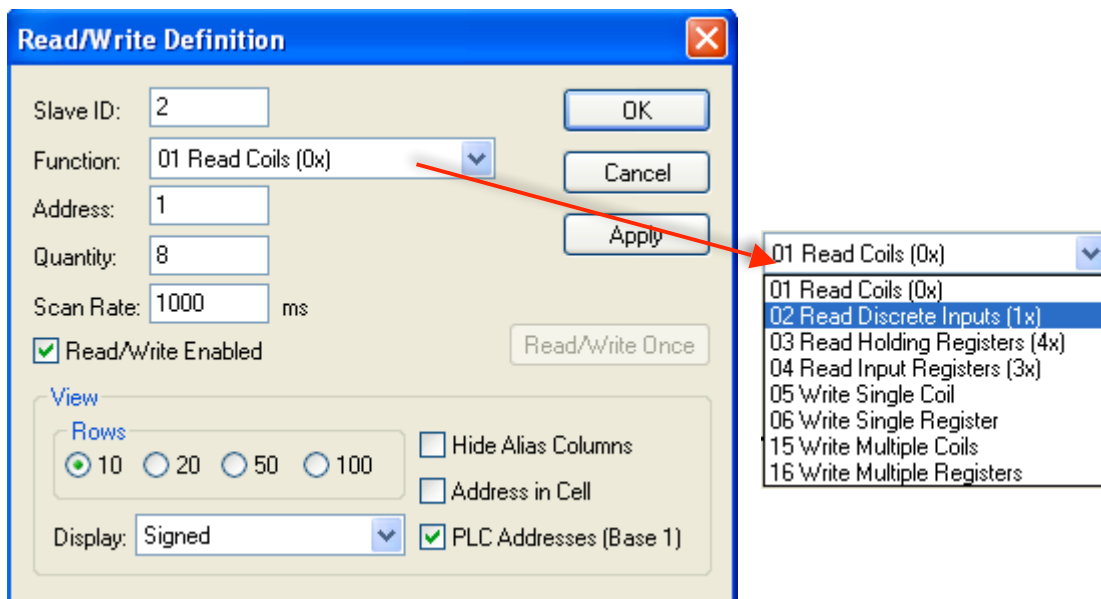


Figura 29; Escritura en las Bobinas o salidas del PLC desde Modbus Poll

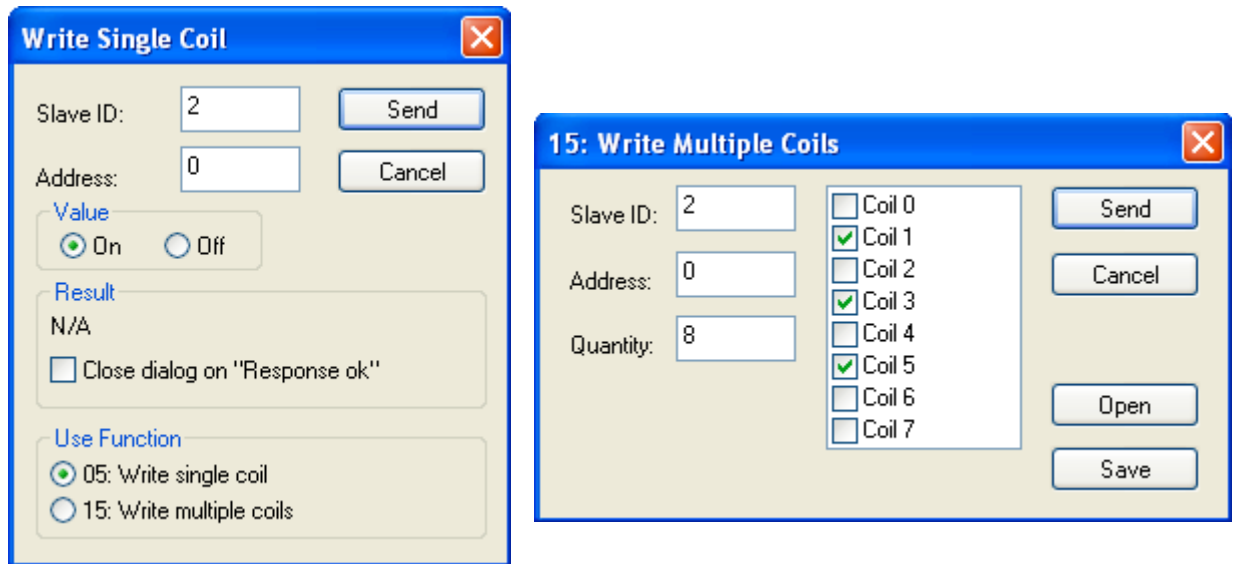
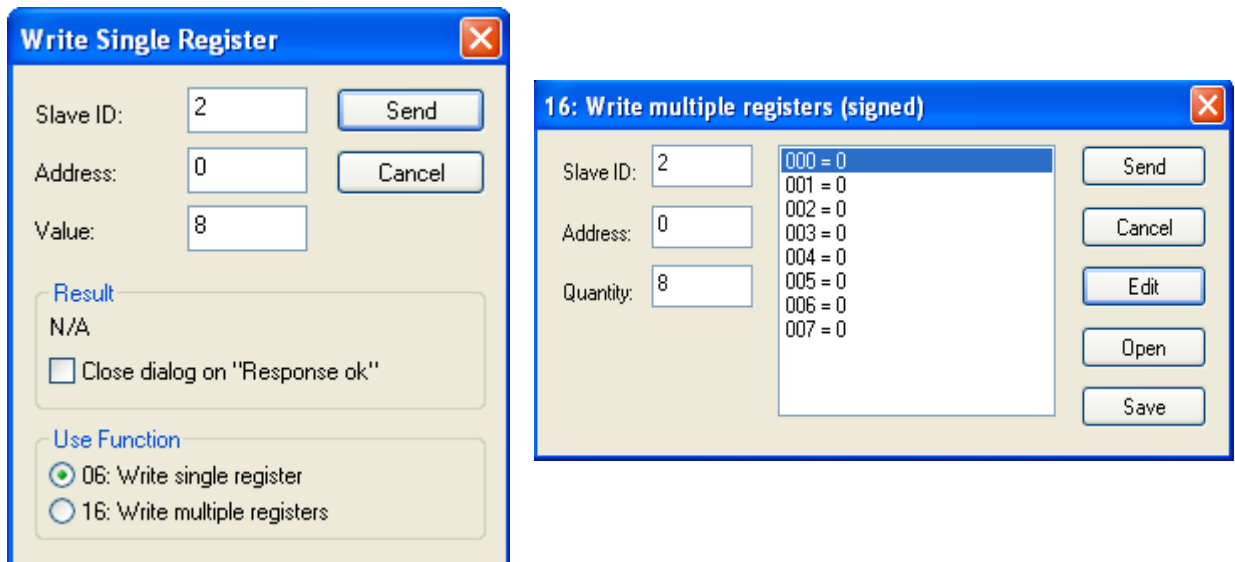


Figura 30; Escritura en los Registros del PLC desde Modbus Poll



Si se desea cambiar un solo registro, simplemente se hace doble clic en el valor, o bien, puede cambiar múltiples registros y bobinas haciendo clic en estas y luego en Send (enviar). Este programa cuenta con las siguientes características:

- Reconocimiento del modo RTU.
- Reconocimiento del modo Modbus TCP/IP.
- Lectura / Escritura de hasta 125 Registros.
- Lectura / Escritura de hasta 2000 Entradas / Salidas.
- Centro de pruebas (Componer sus propias cadenas de prueba).

Y soporta las siguientes funciones del protocolo Modbus:

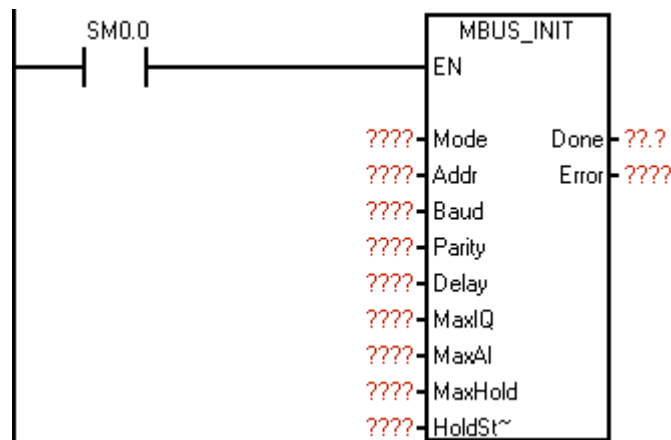
- 01: Leer estado bobina
- 02: Leer estado de entrada.
- 03: Leer registro de retención.
- 05: Escribe en una bobina (salida).
- 06: Escribe en un registro único.
- 15: Escribe en varias bobinas.
- 16: Escribe en múltiples registros.

Después de observar el simulador del Modbus maestro se procede a realizar el programa de Modbus esclavo en el PLC.

#### **4.2.2.2 Programa Modbus esclavo**

Para ello se adhiere el bloque MBUS\_INI de la librería “Modbus Protocol” como se mostró en la figura 12, en donde se muestra el bloque en la figura 31 y así empezar a colocar los datos de inicio del protocolo modbus esclavo.

Figura 31; Inicio de la programación Modbus esclavo.



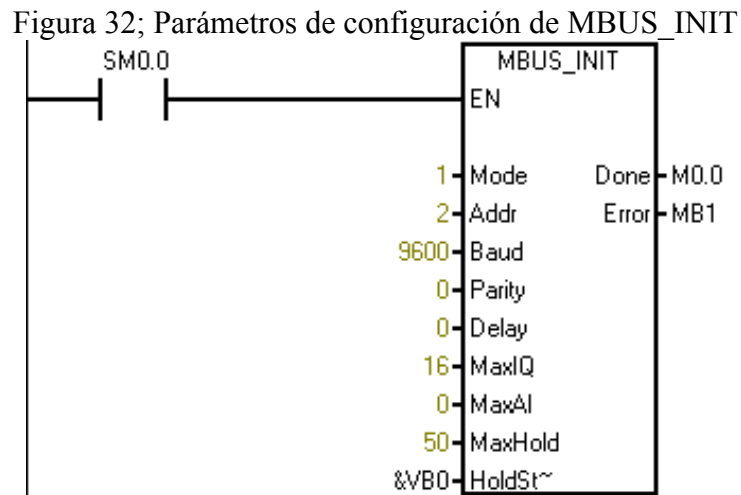
En este bloque se escriben cada una de las características del esclavo modbus que se implementara en esta practica, y se carga de la siguiente manera:

- Mode: 1; Modo de comunicación (0 = PPI & 1 = Modbus)
- Addr: 2; Dirección del esclavo Modbus, preferiblemente que concuerde con la dada en el bloque de sistema puerto de comunicación (ver figura 16).
- Baud: 9600; Velocidad de transferencia de datos.
- Parity: 0; Paridad de la comunicación puede ser: 0 = ninguna, 1 = Impar, 2 = Par, esta se ajusta para que concuerde con la del maestro modbus.
- Delay: 0; Retarda el timeout de fin de mensaje Modbus estándar.
- MaxIQ: 16; Ajusta el número de entradas (I) y salidas (Q) disponibles para las direcciones Modbus.
- MaxAI: 0; Ajusta el número de registros de entradas analógicas (AI) disponibles para la dirección Modbus.
- MaxHold: 50; Ajusta el número de registros de retención en la memoria V disponibles para la dirección Modbus.



- HoldStart: &VB0; Es la dirección inicial de los registros de retención en la memoria V.
- Done: M0.0; Se activa una vez finalizada la operación.
- Error: MB1; Contiene el resultado de ejecución de la operación.

Después de introducido estos parámetros quedaría el bloque como se muestra en la figura 32.



Se realiza la comprobación de errores a través de la compilación la cual debe sugerir que se presentan una cantidad de 20 errores como se muestra en la ventana de resultados del programa (ver figura 33).

Figura 33; Ventana de resultados – Errores falta de memoria

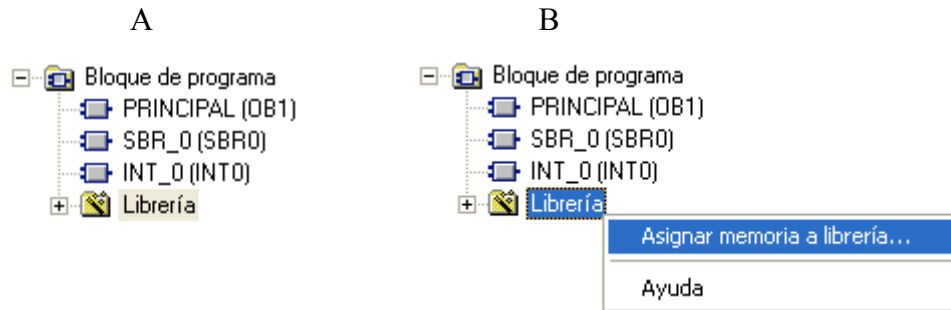


Esto debido a que se le debe asignar la memoria a la librería, lo cual se realiza de la siguiente manera:

- 1 Se abre el árbol de bloque de programa (ver figura 34-A) y se selecciona con el clic

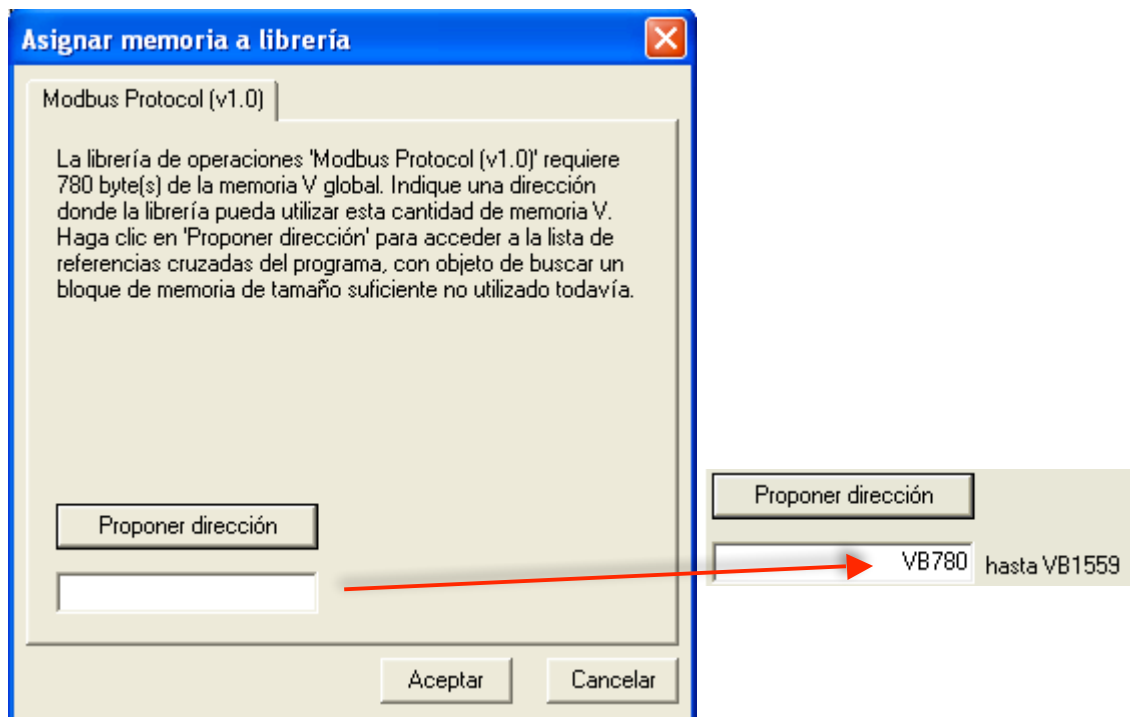
derecho Librería (ver figura 34-B).

Figura 34; Árbol del bloque de programa, Librería.



2. Se abre la ventana Asignar memoria a librería como se muestra en la figura 32, en donde la memoria se asigna dependiendo del HoldStart seleccionado, para este caso se selecciona una memoria desde VB780 hasta VB1559 como se muestra en la figura 35.

Figura 35; Ventana de asignación de memoria.



- .3 Se realiza la compilación y se comprueba nuevamente los errores que existan de compilación. Esto hasta que en la ventan de resultados se muestre Total de errores = 0 como se muestra en la figura 36.

Figura 36; Ventana de resultados, compilado con 0 errores

```
Compilando el bloque del programa...
PRINCIPAL (OB1)
SBR_0 (SBR0)
MBUS_INIT (SBR1)
INT_0 (INT0)
MBUS2 (INT1)
MBUS3 (INT2)
Tamaño del bloque = 767 (bytes), 0 errores

Compilando el bloque de datos...
Tamaño del bloque = 0 (bytes), 0 errores

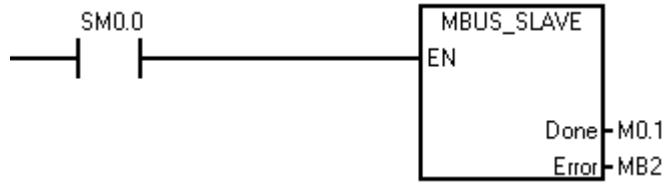
Compilando el bloque de sistema...
Bloque compilado con 0 errores, 0 advertencias

Total errores: 0
```

Después de realizado el proceso de asignación de memoria se introduce el programa al PLC de la manera convencional, y se activa el modo “RUN” y se vuelve a comprobar la comunicación, comprobando de esta manera que no exista solapamiento de las memorias.

Después de comprobar la ausencia de errores se procede a adherir el bloque esclavo, el cual es el encargado de permitir la lectura y escritura en el PLC esclavo, como se muestra en la figura 37, este bloque se utiliza para procesar una petición del maestro Modbus, debiéndose ejecutar en cada ciclo para poder comprobar y responder a las peticiones Modbus.

Figura 37; Programación del Bloque MBUS\_SLAVE

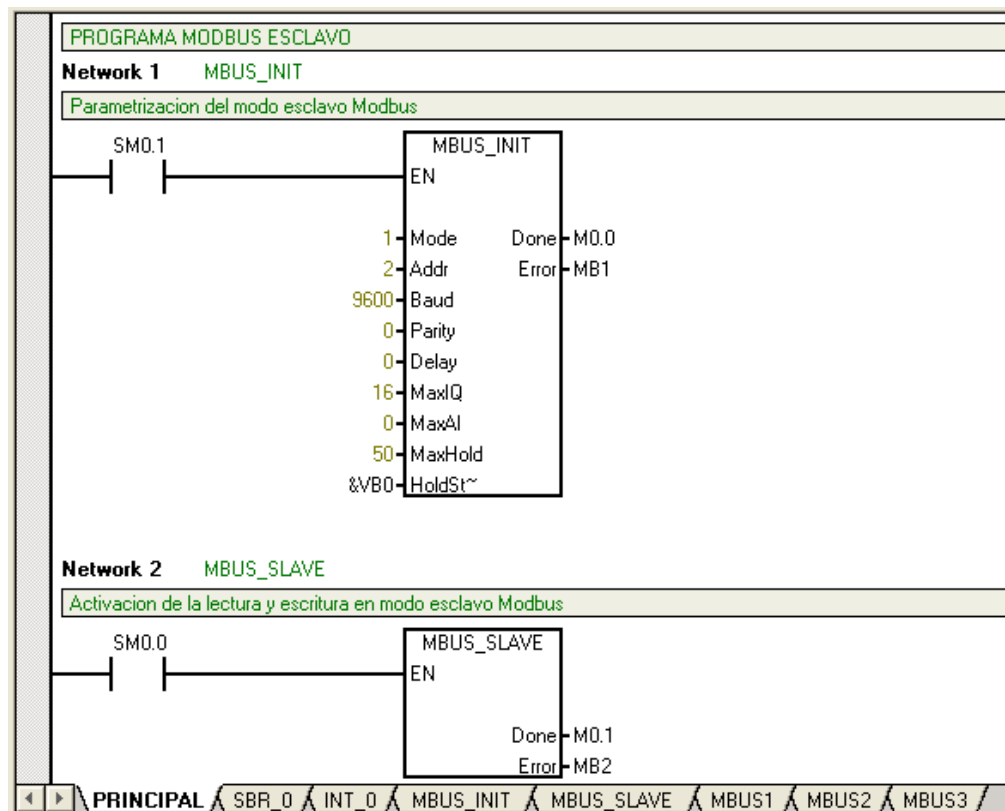


Se realiza la parametrización de este bloque:

Done: M0.1; Se activa cuando la operación responde a una petición Modbus y se desactiva si no se ha procesado ninguna petición

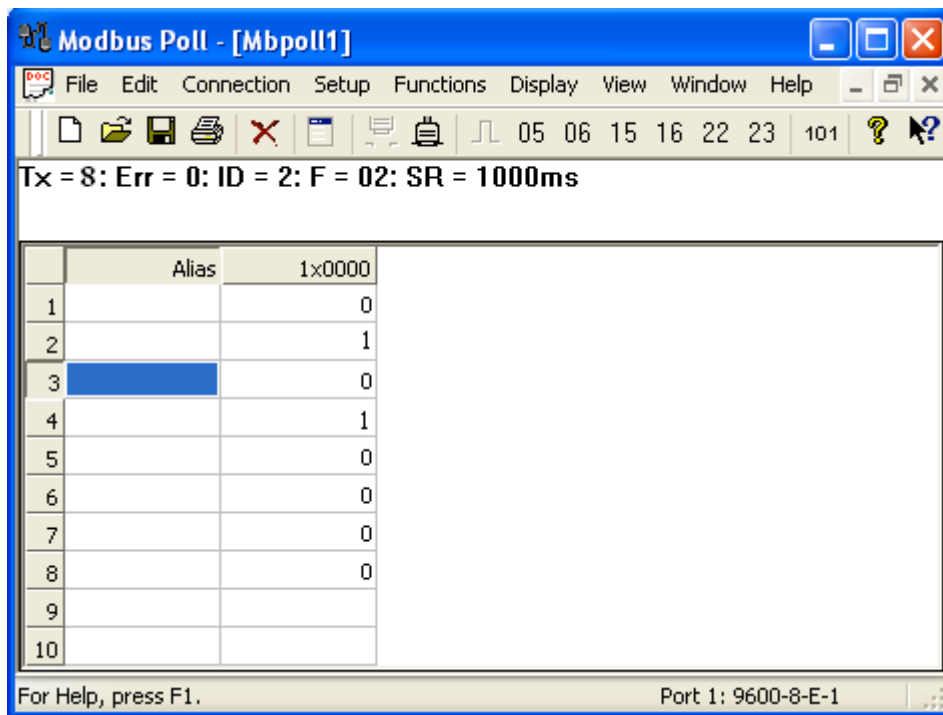
Error: MB2; Contiene el resultado de ejecución de la operación. Esta salida sólo será válida si está activada la salida "Done". Se carga el programa completo (ver figura 38) en la memoria del PLC.

Figura 38; Programa Modbus esclavo



Luego se procede a la visualización del esclavo en el simulador del maestro Modbus (Modbus Poll). Como se muestra en la figura 39. Donde se ven las entradas activas en la ventana de datos, la dirección del esclavo modbus (ID = 2), la función que se esta realizando en este caso 2 (F= 02→lectura de las entradas digitales), los datos transmitidos (Tx = 8), los errores emitidos (Err = 0), y la rata de muestreo (SR = 1000ms).

Figura 39. Ventana de Modbus Poll conectado a la red



**Nota:** Para la simulación o puesta en marcha de un dispositivo modbus maestro, se pueden utilizar diferentes programas, no solo “Modbus Poll”, entre estos se pueden encontrar programas de sistemas SCADA (LabView, DAQFactory, etc.), DCSs, entre otros.

## 4.2 Practicas de Redes propuestas

**Protocolo PPI:** Mediante un PLC esclavo (acceso a un garaje (Mirar ciclo y especificaciones mas adelante)) se desea controlar el ciclo de un semáforo regulado por otro PLC, que actúa como maestro. Ambos autómatas se encuentran conectados a través de sus puertos de comunicaciones.

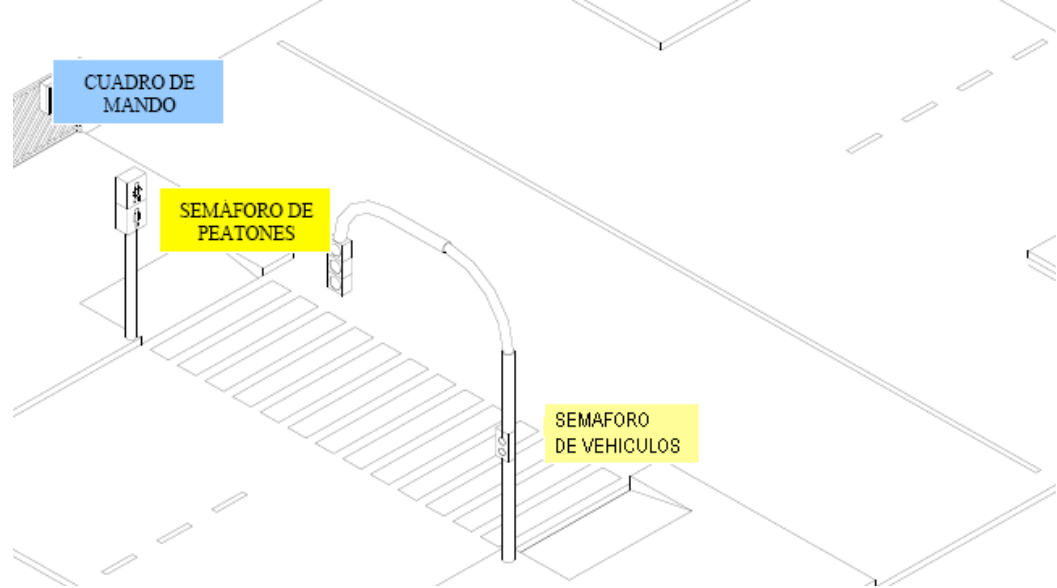
El PLC maestro controla el ciclo de un semáforo (Mirar ciclo y especificaciones mas adelante) para paso de peatones situado antes de la puerta de un garaje para vehículos. El autómata esclavo controla el acceso a un garaje para vehículos. Además el PLC maestro envía continuamente peticiones de lectura a través de su interfaz de comunicaciones al PLC esclavo, controlando continuamente si algún vehículo dentro del parqueadero desea salir.

Ante la salida de un vehículo del parqueadero, el PLC maestro responde iniciando nuevamente la secuencia del semáforo; de forma que el semáforo cambie inmediatamente a “rojo vehículos, verde peatones” y así el vehiculo que desea salir del garaje pueda hacerlo sin ningún peligro. El semáforo continuará ejecutando su ciclo normal hasta que un nuevo vehiculo desee salir del garaje volviéndose a repetir el proceso

- Ciclo de un semáforo, para peatones y vehículos con el fin de regular el paso. Durante un ciclo normal, el semáforo funciona para dar paso a los vehículos y a los peatones mediante sucesivas temporizaciones, realizándose el cambio de luces de amarillo, rojo y verde para los vehículos, y de rojo y verde para los peatones, de tal forma que si está el semáforo en verde para peatones, proceden a pasar éstos y no los vehículos y viceversa.

Como se muestra en la figura 40

Figura 40; Esquema de un semáforo, para peatones y vehículos.



El sistema posee un control del funcionamiento de las luces de forma que permita su mantenimiento en caso de fundirse cualquiera de ellas. Mediante pulsadores se pueden detener el ciclo de funcionamiento normal, pasando a funcionar todas a la vez pudiendo detectar averías.

Además el automatismo permite mediante un pulsador el cambio a intermitente para vehículos, anulándose la secuencia semafórica del ciclo normal. Permitiendo así regular el tráfico de forma manual.

El cambio entre los tres modos de funcionamiento se realiza por pulsadores situados en el cuadro de control del semáforo. Para su ayuda se brinda un cronograma del ciclo normal de un semáforo el cual se muestra a continuación:

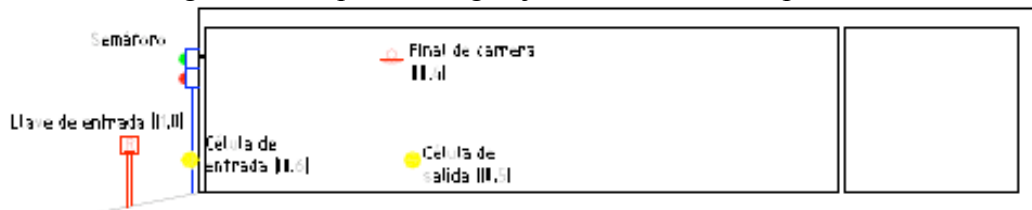
El tiempo total para el ciclo de este semáforo es de 86 segundos, el semáforo se encuentra:

- Entre los segundos 0 y 4 seg: amarillo para vehículos, rojo peatones
- Entre los segundos 10 y 13 seg: rojo para vehículos, rojo para peatones

- Entre los segundos 13 y 43 seg: rojo para vehículos, verde para peatones.
- Entre los segundos 44 y 49 seg: rojo para vehículos, verde intermitente para peatones.
- Entre los segundos 50 y 55 seg: rojo para vehículos, rojo para peatones.
- Entre los segundos 56 y 86 seg: verde para vehículos, rojo para peatones

En otro PLC se realizara el siguiente programa: El control del acceso a un garaje con capacidad máxima de seis vehículos en su interior. El garaje dispone de una sola puerta y dos sensores que detectan el paso al entrar o al salir. Un pulsador de apertura exterior permitirá la entrada y otro pulsador de apertura desde el interior, la salida. Como se muestra en la figura 41.

Figura 41; Esquema del garaje, con control de capacidad.



Cuando un vehículo desee entrar la puerta de garaje se abrirá, siempre que el garaje no se encuentre lleno (con 6 vehículos en su interior), permaneciendo completamente abierta un tiempo de 10 s; transcurrido ese tiempo procederá a cerrarse. Si un vehículo es detectado por el sensor fotoeléctrico exterior o interior cuando la puerta se cierra, esta se detiene y después de un tiempo de espera, vuelve a abrirse.

Se incorpora un semáforo de entrada, en rojo siempre que el garaje se encuentre lleno o desee salir algún vehículo, verde en el resto de situaciones. Empleando un motor



trifásico con inversor de giro para poder abrir y cerrar la puerta, que a su vez dispone de finales de carrera superior e inferior que controlan su posición.

## **5. Conclusiones:**

Entre los temas tratados se pueden destacar las características y funcionamiento del Protocolo ModBus. Ya que es un protocolo muy utilizado actualmente a nivel industrial, y como se observo entre sus características es un protocolo abierto y de implementación para la comunicación en red.

La programación de los PLC a través del protocolo PPI, resulta más sencilla y mas practica al trabajar con dispositivos de Siemens, dicho protocolo es del tipo propietario el cual Siemens utiliza para interconectar sus PLCs s7-200.

La programación de un maestro PPI, es relativamente menos compleja que la de un maestro Modbus, ya que este ultimo requiere de características especiales, que los S7-200; CPU222 no poseen, por ello su implementación no fue posible.

Se puede comprender que el simulador Modbus utilizado (Modbus Poll) es muy poco práctico a nivel industrial, pero existen otros programas como Labview, DAQFactory, entre otros, que pueden brindar mejores resultados a la hora de interactuar con los dispositivos esclavos.

## **6. Bibliografía**

- **De La Espriella B., Alfredo, Mancera Guzmá, Oswald,** Implementación en laboratorio del protocolo Modbus; Universidad Tecnológica de Bolívar.
- **Hernandez Zakzuk, Julian J.; Gutierrez Rapalino, Alfredo R.** Implementación de una red Modbus con Labview; Universidad Tecnológica de Bolívar.
- **Gándara Jadid, Mauricio José; Pérez Romero, Pedro Julio;** Redes y buses de campo; Universidad Tecnológica de Bolívar.
- **Modicon, Modbus Protocol, Reference Guide MODICON, Inc.,** Industrial Automation Systems One High Street North Andover, Massachusetts 01845.
- **Manual Del Sistema De Automatización S7-200;** Siemens AG Bereich Automation and Drives Geschaeftsgebiet Industrial Automation Systems Postfach 4848, D 90327 Nuremberg.