

# **IMPLEMENTACIÓN EN LABORATORIO DEL PROTOCOLO MODBUS**

**Alfredy De La Espriella B**

**Oswald Mancera Guzmán**

**UNIVERSIDAD TECNOLÓGICA DE BOLIVAR**

**FACULTAD DE INGENIERIA**

**CARTAGENA DE INDIAS DT Y C**

**2005**

**IMPLEMENTACIÓN EN LABORATORIO DEL PROTOCOLO MODBUS**

**Alfredy De La Espriella B**

**Oswald Mancera Guzmán**

**MONOGRAFIA COMO REQUISITO PARA OPTAR AL TITULO DE INGENIERO  
ELECTRÓNICO**

**DIRECTOR**

**MSC Jaime Arcila Iriarte**

**UNIVERSIDAD TECNOLÓGICA DE BOLIVAR**

**FACULTAD DE INGENIERIA**

**CARTAGENA DE INDIAS DT Y C**

**2005**

**NOTA DE ACEPTACIÓN**

---

---

---

---

---

---

---

---

---

**Firma Del Director**

---

**Firma Del Jurado**

---

**Firma Del Jurado**

Cartagena, 28 De Mayo De 2005

## CONTENIDO

	pág.
INTRODUCCIÓN	10
1. GENERALIDADES	11
2. CAPAS DE APLICACIÓN Y ENLACE MODBUS	12
2.1 PRINCIPIOS DEL PROTOCOLO MAESTRO / ESCLAVO MODBUS	12
2.2 REGLAS DE DIRECCIONAMIENTO EN MODBUS	13
2.3 DIAGRAMAS DE ESTADO MAESTRO / ESCLAVO	14
2.3.1 Diagrama de Estados del Maestro	15
2.3.2 Diagrama de Estados del Esclavo	17
2.4 DIAGRAMA DE TIEMPOS DE LA COMUNICACIÓN MAESTRO / ESCLAVO	18
2.5 LOS DOS MODOS DE TRANSMISIÓN SERIE	19
2.5.1 Modo ASCII	20
2.5.2 Modo RTU	21
2.6 TRAMA DEL MENSAJE MODBUS	22
2.6.1 Trama ASCII	23
2.6.2 Trama RTU	24
2.6.3 El Campo Dirección	27
2.6.4 El Campo Función	28

2.6.5 El Campo Datos	29
2.6.6 El Campo Comprobación De Error	31
▪ ASCII	31
▪ RTU	31
2.7 CÓMO SON TRANSMITIDOS LOS CARACTERES EN SERIE	32
2.8 MÉTODOS DE COMPROBACIÓN DE ERROR	33
2.8.1 Control de Paridad	34
2.8.2 Comprobación LRC	35
2.8.3 Comprobación CRC	37
2.9 FORMATOS DE FUNCIONES MODBUS	39
2.9.1 Direcciones en los Mensajes MODBUS	39
2.9.2 Campos Contenidos En Los Mensajes MODBUS	40
2.9.3 Campos Contenidos En Los Mensajes Sobre MODBUS Plus	43
2.10 CATEGORÍAS DE LOS CÓDIGOS DE FUNCIÓN	44
2.10.1 Definición De Los Códigos De Función	46
2.11 RESPUESTAS DE EXCEPCIÓN	46
2.11.1 Campo de Código de Función	47
2.11.2 Campo de Datos	48
3. CAPA FÍSICA MODBUS	51
3.1 GENERALIDADES	51
3.2 INTERFAZ ELECTRICA	51
3.2.1 Infraestructura De Bus Serial Multipunto	51

3.2.2 Definición De Protocolo MODBUS De Dos Cables	52
3.2.3 Definición De Protocolo MODBUS De Cuatro Cables Opcional	54
3.2.4 Compatibilidad Entre Los Sistemas De Cuatro Y Dos Cables	55
3.2.5 Definición De MODBUS En RS-232	57
3.3 REQUERIMIENTOS DE SISTEMAS MULTIPUNTO	58
3.3.1 Máximo Numero De Aparatos Sin Repetidor	59
3.3.2 Topología	59
3.3.3 Longitud	59
3.3.4 Conexiones De Tierra	60
3.3.5 Terminaciones de líneas	60
3.3.6 Polarización de líneas	61
3.4 INTERFACES MECANICAS	62
3.4.1 Diagramas De Pines De Los Conectores Para 2W MODBUS	62
3.4.2 Diagramas De Pines De Los Conectores Para 4W MODBUS	64
3.4.3 Diagramas De Pines De Los Conectores Para Rs-232 MODBUS	65
3.5 CABLES	65
4. CONCLUSIONES	67
5. RECOMENDACIONES	68

## LISTA DE FIGURAS

	Pág
Figura 1: Posicionamiento De MODBUS En El Modelo OSI.	12
Figura 2: Modo único en MODBUS	13
Figura 3: Modo difusión en MODBUS.	13
Figura 4: Ejemplo De Diagrama De Estado	15
Figura 5: Diagrama De Estado Del Maestro	15
Figura 6: Diagrama De Estado Del Esclavo	17
Figura 7: Diagrama De tiempos MODBUS	19
Figura 8: Trama de mensajes ASCII	24
Figura 9: Trama de mensajes RTU	26
Figura 10: Tiempos En La Trama De Mensajes RTU	26
Figura 11: Tiempos En La Trama De Mensajes RTU con error.	25
Figura 12: Secuencia de Bits en la trama ASCII.	32
Figura 13: Secuencia de Bits en la trama RTU.	33
Figura 14: Petición Del Maestro Con Trama ASCII/RTU.	42
Figura 15: Respuesta De Un Esclavo Con Trama ASCII/RTU.	42
Figura 16: Campos En MODBUS Plus.	44
Figura 17: Categorías de los códigos de función MODBUS.	45
Figura 18: Códigos De Función Públicos MODBUS.	46

Figura 19: Petición Del Maestro Y Respuesta De Excepción Del Esclavo	49
Figura 20: Listado De Códigos De Excepción	50
Figura 21: Infraestructura Del Bus Serial.	51
Figura 22: Topología General De Dos Cables	53
Figura 23: Definición de Circuitos 2W	53
Figura 24: Topología General De Cuatro Cables	54
Figura 25: Definición de Circuitos 4W	55
Figura 26: Conexión De Un Sistema De 2W A 4W.	56
Figura 27 : Conexión De Un Sistema De 4W A 2W	57
Figura 28: Definición De Los Circuitos MODBUS En RS-232	58
Figura 29 : Conector RJ45 Para 2w MODBUS.	62
Figura 30 : Conector DB9 Para 2w MODBUS.	63
Figura 31: Pinout De Los Conectores Y Su Condicionamiento	63
Figura 32 : Conector RJ45 Para 2W MODBUS	64
Figura 33 : Conector DB9 Para 4W MODBUS.	64
Figura 34: Pinout De Los Conectores Y Su Condicionamiento	64
Figura 35: Diagrama De Pines Para MODBUS RS232.	65



## ANEXOS

	<b>Pàg</b>
Anexo A Guía de práctica del estudiante del protocolo MODBUS	70
Anexo B Guía de práctica del profesor.	89

## **RESUMEN**

El Protocolo MODBUS es un sistema de comunicaciones industrial que esta ubicado en la capa dos del modelo OSI, en este documento se desglosa todo el protocolo detallando todos los aspectos basándose en la imagen de este en el modelo OSI y detallando especialmente los aspectos del manejo de la trama MODBUS y todos los campos contenidos por ella.

Finalmente este documento termina con la formulación de un laboratorio de protocolo MODBUS usando un software simulador.

## INTRODUCCIÓN

El Presente trabajo se realizó con el fin de darle a la universidad los medios para capacitar a los estudiantes que quieran profundizar en el protocolo MODBUS por encima de lo netamente académico ó lo que se encuentra publicado, para llevarlos, a un saber teórico-práctico basado en las publicaciones del tema y los simuladores MODBUS.

Una práctica de Simuladores MODBUS es fundamental para acercarse a la realidad de los procesos industriales. Dada su versatilidad y teniendo en cuenta que este es uno de los protocolos mas antiguos, algunas empresas lo prefieren, ya que sirve como red de campo cuando los procesos están en lugares distanciados geográficamente (Pozos petroleros, de gas, ù otros). Además el protocolo MODBUS se puede indexar en redes de más alta complejidad, proveyéndole un mejor acercamiento a las tecnologías actuales.

## 1. GENERALIDADES

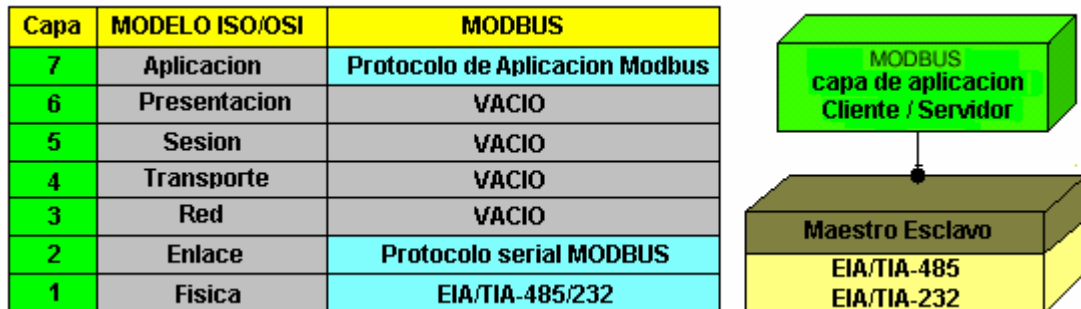
La designación MODBUS Modicon corresponde a una marca registrada por Gould Inc. Como en tantos otros casos, la designación no corresponde propiamente al estándar de red, incluyendo todos los aspectos desde el nivel físico hasta el de aplicación, sino a un protocolo de enlace (nivel OSI 2). Puede, por tanto, implementarse con diversos tipos de conexión física y cada fabricante suele suministrar un software de aplicación propio, que permite parametrizar sus productos.

EL estándar MODBUS define una capa de protocolo de mensajes, posicionada en el nivel 7 del modelo OSI que provee comunicaciones cliente / servidor entre aparatos conectados en diferentes tipos de buses ó redes. También estandariza un protocolo específico en líneas seriales para el intercambio de peticiones MODBUS entre un maestro y uno ó más esclavos.

En el nivel físico, MODBUS puede usar interfaces físicas diferentes (RS-232, RS -485 y sus variantes).

La siguiente figura da una representación de MODBUS comparada con las 7 capas del modelo OSI:

**Figura 1: Posicionamiento De MODBUS En El Modelo OSI.**



## 2. CAPAS DE APLICACIÓN Y ENLACE MODBUS

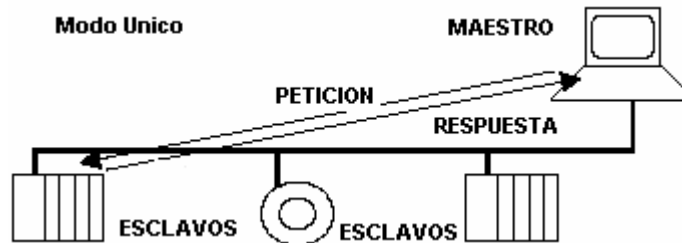
### 2.1 PRINCIPIOS DEL PROTOCOLO MAESTRO / ESCLAVO MODBUS

El protocolo MODBUS es un protocolo de línea serial maestro / esclavo. Solo un maestro es conectado al bus, y uno ó varios nodos esclavos son conectados al mismo bus serial. Una comunicación MODBUS es iniciada siempre por el maestro. Los nodos esclavos nunca podrían transmitir datos sin recibir una petición del nodo maestro. Además los nodos esclavos no pueden comunicarse entre si. Esto es que un nodo maestro inicia solo una transacción MODBUS a la vez.

El nodo maestro emite una petición MODBUS al nodo esclavo en dos modos:

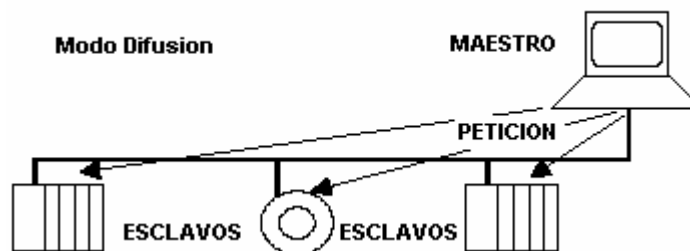
- **Modo único:** El maestro se dirige a un esclavo individual que después de recibir y procesar la petición, el esclavo devuelve un mensaje ó respuesta al maestro (figura 2).

**Figura 2: Modo único en MODBUS**



- **Modo Difusión:** El maestro envía una petición a todos los esclavos y estos no responden. Estos mensajes son comandos de escritura y todos los esclavos deben obligatoriamente aceptar la difusión para la función de escritura.

**Figura 3: Modo difusión en MODBUS.**



## **2.2 REGLAS DE DIRECCIONAMIENTO EN MODBUS.**

El direccionamiento MODBUS esta conformado por 256 direcciones diferentes a saber:

Dirección 0: Es la dirección de difusión.

Direcciones entre 1 a 247: son las direcciones de cada esclavo individual.

Direcciones entre 248 a 255: son direcciones reservadas.

La dirección 0 es una dirección reservada como la dirección de difusión y todos los esclavos tienen que reconocerla.

El nodo maestro no tiene una dirección específica, solo los esclavos tienen una dirección la cual es única en un bus serial MODBUS.

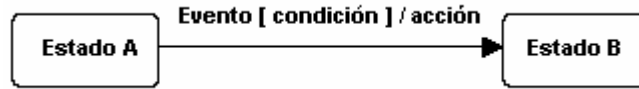
Con este aparte concluimos que el número máximo de nodos es de 248 sabiendo que uno de ellos es el maestro.

### **2.3 DIAGRAMAS DE ESTADO MAESTRO / ESCLAVO.**

La siguiente sección describe los diagramas de estado de un maestro y un esclavo y son parte importante del protocolo. Para leer los diagramas se debe entender la notación estándar de estados de la UML así:

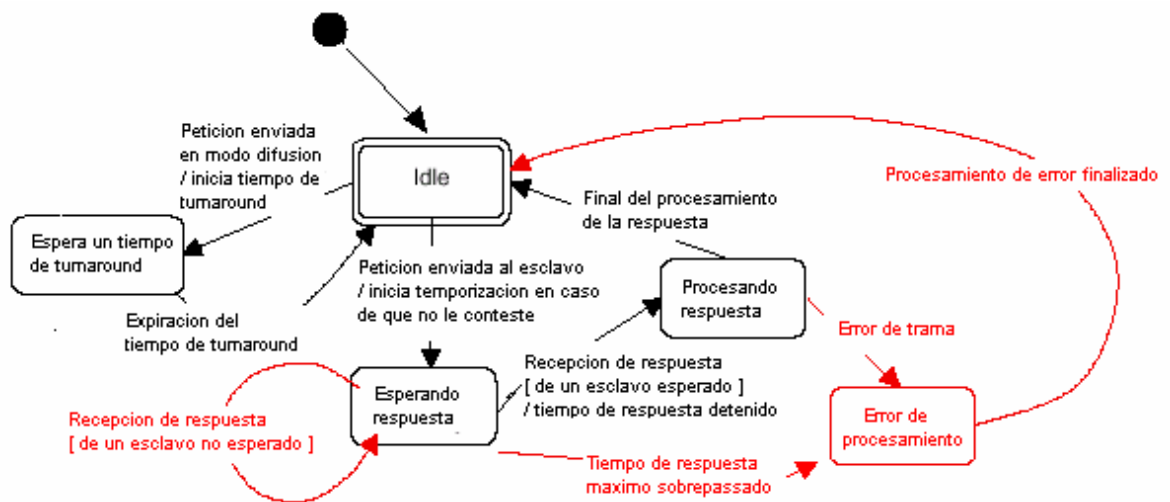
Cuando un sistema que estuvo en un “Estado A” y ocurre un evento que genera un cambio de estado a un “Estado B” es posible solo si la condición es verdadera, por lo tanto la “Acción” se lleva a cabo (figura 4).

**Figura 4: Ejemplo De Diagrama De Estado**



**2.3.1 Diagrama de Estados del Maestro** El siguiente es un diagrama que explica el comportamiento del maestro:

**Figura 5: Diagrama De Estado Del Maestro**



Explicaciones del diagrama:

- Estado Idle: es un estado donde no hay ninguna petición pendiente. Es el que asume después de poner a correr MODBUS. Después de enviar una petición el maestro sale de este estado y no puede enviar otra petición hasta volver a él.

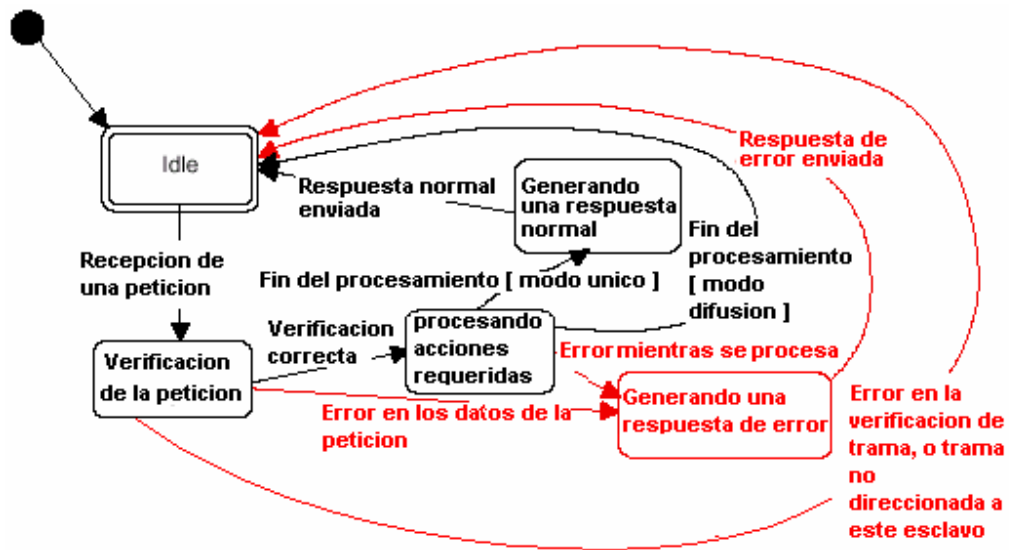


- Estado Esperando respuesta: es el estado que asume el maestro cuando envía un mensaje a un esclavo único, en este estado se inicia un temporizador que previene que el maestro este en este estado permanentemente. En este estado pueden ocurrir tres cosas: una respuesta de un esclavo no esperado, en cuyo caso el temporizado se mantiene corriendo y se mantiene el estado de espera de respuesta; una respuesta del esclavo esperado, el cual genera una parada del temporizador y posteriormente un cambio de estado al procesamiento de la respuesta; y finalmente puede finalizar el conteo del temporizador lo que genera un cambio de estado al procesamiento de error.
  
- Estado de procesamiento de la respuesta: es el estado que asume el maestro cuando ha recibido una respuesta del esclavo esperado y pueden ocurrir dos cosas: que se termine el procesamiento y llegue al estado Idle, teniendo en cuenta que la operación fue exitosa ó que haya un error dentro de la trama que generó el esclavo lo cual lleva al maestro al estado de procesamiento del error.
  
- Estado de procesamiento de error: aquí se genera una respuesta al error que sucedió a la operación no exitosa y luego llega al estado Idle.

- Estado de tiempo de Turnaround: es el que adquiere después de mandar un mensaje tipo difusión y es básicamente esperar un tiempo en el cual se garantice que todos los esclavos lo reciban y procesen, por lo general este tiempo es mucho menor al tiempo de respuesta de un esclavo.

**2.3.2 Diagrama de Estados del Esclavo** El siguiente es un diagrama que explica el comportamiento del esclavo:

**Figura 6: Diagrama De Estado Del Esclavo**



Explicaciones del diagrama.

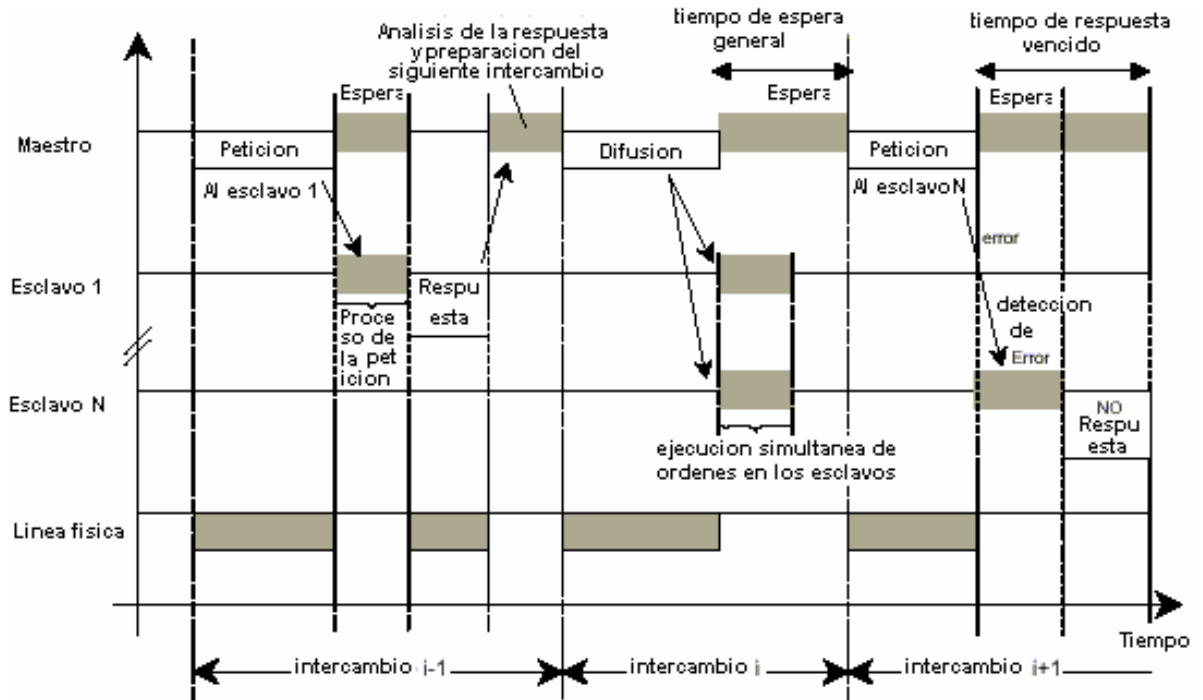
- Estado Idle: es un estado donde no hay ninguna petición pendiente. Es el que asume después de poner a correr MODBUS.

- Cuando una petición es recibida, el esclavo la verifica antes de realizar cualquier acción pedida. En este punto varios errores pueden pasar: error de formato en la petición, acción inválida, etc. En caso de error debe ser enviada una respuesta al maestro.
- Una vez que la acción requerida es completada, se requiere que sea generado un mensaje único y enviado al maestro.
- Si el esclavo detecta un error en la trama, no se envía ninguna respuesta de vuelta al maestro luego entonces el maestro genera un error por time out y se tiene que mandar otra vez la petición.

#### **2.4 DIAGRAMA DE TIEMPOS DE LA COMUNICACIÓN MAESTRO / ESCLAVO.**

El siguiente es un diagrama en el que se muestra el comportamiento del tiempo con MODBUS teniendo en cuenta los estados de cada uno de los elementos MODBUS.

**Figura 7: Diagrama De tiempos MODBUS.**



## 2.5 LOS DOS MODOS DE TRANSMISIÓN SERIE

Los controladores pueden ser configurados para comunicar sobre redes estándar MODBUS utilizando cualquiera de los dos modos de transmisión: ASCII o RTU. Los usuarios seleccionan el modo deseado, junto con los parámetros de comunicación del puerto serie (velocidad, paridad, etc), durante la configuración de cada controlador. El modo y los parámetros serie deben ser los mismos para todos los dispositivos conectados a una red MODBUS.

La selección del modo ASCII o RTU tiene que ver únicamente con redes MODBUS estándar.

Define los bits contenidos en los campos del mensaje transmitido en forma serie en esas redes.

Determina cómo debe ser empaquetada y decodificada, la información en los campos del mensaje.

En otras redes como MAP y MODBUS PLUS, los mensajes MODBUS son situados en tramas sin relación con la transmisión serie. Por ejemplo una solicitud para leer registros mantenidos (holding reg) puede ser manejada entre dos controladores en MODBUS PLUS, con independencia de la configuración actual de los puertos serie MODBUS de ambos controladores.

**2.5.1 Modo ASCII** Cuando los controladores se configuran para comunicar en una red MODBUS según el modo ASCII (American Standard Code for Information Interchange), cada byte – 8 bits - en un mensaje se envía como dos caracteres ASCII. La principal ventaja de este modo es que permite intervalos de tiempo de hasta un segundo entre caracteres sin dar lugar a error.

El formato para cada byte en modo ASCII es:

<b>Sistema de codificación:</b>	Hexadecimal, caracteres ASCII 0-9, A-F. Un carácter hexadecimal contenido en cada carácter ASCII del mensaje.
<b>Bits por byte:</b>	1 bit de arranque. 7 bits de datos, el menos significativo se envía primero. 1 bit para paridad Par o Impar; ningún bit para No paridad. 1 bit de paro si se usa paridad; 2 bits si no se usa paridad.
<b>Campo de Comprobación de error:</b>	Comprobación Longitudinal Redundante (LRC).

**2.5.2 Modo RTU** Cuando los controladores son configurados para comunicar en una red MODBUS usando el modo RTU (Remote Terminal Unit), cada byte de 8 bits en un mensaje contiene dos dígitos hexadecimales de 4 bits. La principal ventaja de este modo es que su mayor densidad de caracteres permite mejor rendimiento que el modo ASCII para la misma velocidad. Cada mensaje debe ser transmitido en un flujo continuo.

El formato para cada byte en modo RTU es:

<b>Sistema de codificación:</b>	Binario 8-bits, hexadecimal 0-9, A-F. Dos dígitos hexadecimales contenidos en cada campo de 8 bits del mensaje.
<b>Bits por byte:</b>	1 bit de arranque. 8 bits de datos, el menos significativo se envía primero. 1 bit para paridad Par o Impar; ningún bit para No paridad. 1 bit de paro si se usa paridad; 2 bits si no se usa paridad.
<b>Campo de Comprobación de error:</b>	Comprobación Cíclica Redundante (CRC).

## 2.6 TRAMA DEL MENSAJE MODBUS

En cualquiera de los modos de transmisión serie (ASCII o RTU), un mensaje MODBUS es situado por el dispositivo que transmite, en una trama que tiene un comienzo y un final conocidos. Esto permite a los dispositivos receptores comenzar en el arranque del mensaje, leer la parte de la dirección y

determinar qué dispositivo es direccionado (o todos los dispositivos si es una difusión 'dirección = 0') y conocer cuándo se ha completado el mensaje.

Mensajes parciales pueden ser detectados y establecer errores como resultado.

En redes como MAP o MODBUS Plus, el protocolo de red manipula la trama de los mensajes con delimitadores de comienzo y final que son específicos de la red.

Esos protocolos también manipulan el envío al dispositivo de destino, haciendo innecesario el campo de la dirección MODBUS integrado en el mensaje para la transmisión actual. (La dirección MODBUS es convertida a una dirección de nodo de la red y enrutada por el controlador remitente o sus adaptadores de red).

**2.6.1 Trama ASCII** En modo ASCII, los mensajes comienzan con un carácter (:) 'dos puntos' (ASCII 3A hex.) y terminan con un par de caracteres (CRLF) 'Retorno de Carro + Avance de Línea' (ASCII 0D hex. y 0A hex.).

Los caracteres a transmitir permitidos para todos los demás campos son 0-A, A-F hexadecimal.



Los dispositivos conectados en red monitorizan el bus de red continuamente para detectar un carácter 'dos puntos'. Cuando se recibe, cada dispositivo decodifica el próximo campo (el campo de dirección) para enterarse si es el dispositivo direccionado.

Pueden haber intervalos de hasta un segundo entre caracteres dentro del mensaje. Si transcurre más tiempo entre caracteres, el dispositivo receptor asume que ha ocurrido un error.

Se muestra debajo una trama de mensaje típica.

**Figura 8: Trama de mensajes ASCII.**

ARRANQUE	DIRECCION	FUNCION	DATOS	COMPROB LRC	FINAL
1 carácter ;	2 caracteres	2 caracteres	N caracteres	2 caracteres	2 caracteres CRLF

**2.6.2 Trama RTU** En modo RTU, los mensajes comienzan con un intervalo silencioso de al menos 3.5 tiempos de carácter. Esto es más fácilmente implementado como un múltiplo de tiempos de carácter a la velocidad de transmisión configurada en la red (mostrado como T1-T2-T3-T4 en la figura

4). El primer campo transmitido es entonces la dirección del dispositivo destinatario.

Los caracteres a transmitir permitidos para todos los campos son 0-A, A-F hexadecimal.

Los dispositivos conectados en red monitorizan el bus de red continuamente incluso durante los intervalos 'silencioso'. Cuando el primer campo (el campo de dirección) es recibido, cada dispositivo lo decodifica para enterarse si es el dispositivo direccionado.

Siguiendo al último carácter transmitido, un intervalo de al menos 3.5 tiempos de carácter señala el final del mensaje. Un nuevo mensaje puede comenzar después de este intervalo.

La trama completa del mensaje debe ser transmitida como un flujo continuo. Si un intervalo silencioso de más de 1.5 tiempos de carácter tiene lugar antes de completar la trama, el dispositivo receptor desecha el mensaje incompleto y asume que el próximo byte será el campo de dirección de un nuevo mensaje.

De forma similar, si un nuevo mensaje comienza antes de que transcurran 3.5 tiempos de carácter después de un mensaje previo, el dispositivo receptor lo considerará una continuación del mensaje previo. Esto dará lugar a un error, ya que el valor en el campo final CRC no será válido para el mensaje combinado.

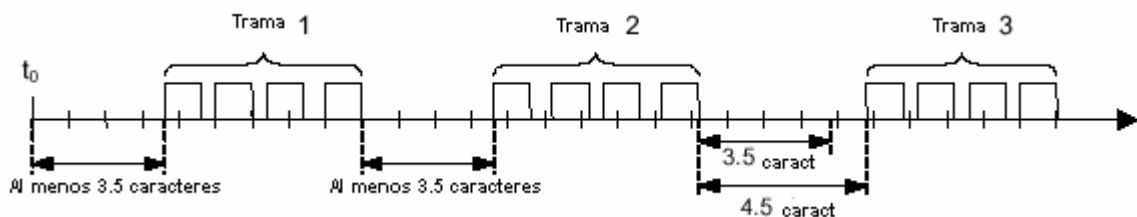
Debajo se muestra una trama de mensaje típica.

**Figura 9: Trama de mensajes RTU.**

ARRANQUE	DIRECCION	FUNCION	DATOS	COMPROB CRC	FINAL
T1-T2-T3-T4	8 BITS	8 BITS	N X 8 BITS	16 BITS	T1-T2-T3-T4

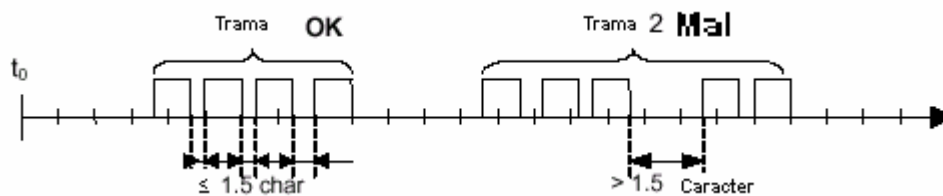
El siguiente gráfico ilustra los tiempos en la trama RTU

**Figura 10: Tiempos En La Trama De Mensajes RTU.**



Si un intervalo de silencio de más de 1.5 caracteres en tiempo ocurre entre dos caracteres, la trama es declarada incompleta y es descargada por el receptor.

**Figura 11: Tiempos En La Trama De Mensajes RTU con error.**



**2.6.3 El Campo Dirección** El campo dirección de un mensaje contiene dos caracteres (ASCII) u ocho bits (RTU). Las direcciones de esclavo válidas están en el rango de 0 – 247 decimal. Los dispositivos esclavos individuales tienen direcciones asignadas en el rango 1 – 247. Un maestro direcciona un esclavo situando la dirección del esclavo en el campo dirección del mensaje. Cuando el esclavo envía su respuesta, sitúa su propia dirección en este campo dirección de la respuesta para dar a conocer al maestro qué esclavo está respondiendo. La dirección 0 es utilizada para dirección difusión, la cual todos los dispositivos esclavos reconocen. Cuando el protocolo MODBUS es usado en redes de nivel más alto, las difusiones pueden no estar permitidas o pueden ser reemplazadas por otros métodos. Por ejemplo, MODBUS Plus

utiliza una base de datos global compartida que puede ser actualizada con cada rotación del testigo.

**2.6.4 El Campo Función** El campo código de función de una trama de mensaje contiene dos caracteres (ASCII) u ocho bits (RTU). Los códigos válidos están en el rango de 1 – 255 decimal. De esos, algunos códigos son aplicables a todos los controladores Modicon, mientras que algunos códigos se aplican sólo en algunos modelos y otros están reservados para usos futuros. Cuando un mensaje es enviado desde un maestro a un dispositivo esclavo, el campo del código de función indica al esclavo qué tipo de acción ha de ejecutar. Por ejemplo: leer los estados ON /OFF de un grupo bobinas o entradas discretas; leer el contenido de datos de un grupo de registros; leer el status de diagnóstico de un esclavo; escribir en determinadas bobinas o registros; o permitir cargar, salvar o verificar el programa dentro del esclavo. Cuando el esclavo responde al maestro, utiliza el campo del código de función para indicar bien una respuesta normal (libre de error) o que algún tipo de error ha tenido lugar (denominado respuesta de excepción). Para una respuesta normal, el esclavo simplemente replica el código de función original. Para una respuesta de excepción, el esclavo devuelve un código que es equivalente al código de función original con su bit más significativo puesto a valor 1.

Por ejemplo, un mensaje desde un maestro a un esclavo para leer un grupo de registros mantenidos tendría el siguiente código de función:

0000 0011 (Hexadecimal 03)

Si el dispositivo esclavo ejecuta la acción solicitada, sin error, devuelve el mismo código en su respuesta. Si ocurre una excepción. Devuelve:

1000 0011 (Hexadecimal 83)

Además de la modificación del código de función para una respuesta de excepción, el esclavo sitúa un único código en el campo de datos del mensaje respuesta. Esto indica al maestro qué tipo de error ha tenido lugar, o la razón para la excepción. El programa de aplicación del maestro tiene la responsabilidad de manejar las respuestas de excepción. Procedimientos típicos son: enviar subsiguientes reintentos de mensaje, intentar mensajes de diagnóstico al esclavo y notificar operadores.

**2.6.5 El Campo Datos** El campo datos se construye utilizando conjuntos de 2 dígitos hexadecimales, en el rango de 00 á FF hexadecimal. Pueden formarse a partir de un par de caracteres ASCII o desde un carácter RTU, de acuerdo al modo de transmisión serie de la red.

El campo datos de los mensajes enviados desde un maestro a un esclavo, contiene información adicional que el esclavo debe usar para tomar la acción definida por el código de función. Esto puede incluir partes como direcciones discretas y de registros, la cantidad de partes que han de ser manipuladas y el cómputo de bytes de datos contenidos en el campo. Por ejemplo, si el maestro solicita a un esclavo leer un grupo de registros mantenidos (código de función 03), el campo de datos especifica el registro de comienzo y cuántos registros han de ser leídos. Si el maestro escribe sobre un grupo de registros en el esclavo (código de función 10 hexadecimal), el campo datos especifica el registro de comienzo, cuántos registros escribir, el cómputo de bytes de datos que siguen en el campo datos y los datos que se deben escribir en los registros. Si no ocurre error, el campo datos de una respuesta desde un esclavo al maestro contiene los datos solicitados. Si ocurre un error, el campo contiene un código de excepción que la aplicación del maestro puede utilizar para determinar la próxima acción a tomar.

El campo datos puede ser inexistente (de longitud cero) en ciertos tipos de mensajes. Por ejemplo, en una petición de un dispositivo maestro a un esclavo para que responda con su anotación de eventos de comunicación (Código de función 0B hexadecimal), el esclavo no requiere ninguna información adicional. El código de función por sí solo especifica la acción.

**2.6.6 El Campo Comprobación De Error** Dos tipos de métodos de comprobación de error son utilizados para las redes MODBUS estándar. El contenido del campo Comprobación de Error depende del método que esté siendo utilizado.

- **ASCII** Cuando el modo ASCII es usado para trama de carácter, el campo Comprobación de Error contiene dos caracteres ASCII. Los caracteres de comprobación de error son el resultado de un cálculo Comprobación Longitudinal Redundante (LRC) que es realizado sobre el contenido del mensaje, excluyendo los 'dos puntos' del comienzo y los caracteres CRLF de finalización.

Los caracteres LRC son añadidos al mensaje como el último campo que precede a los caracteres CRLF.

- **RTU** Cuando el modo RTU es usado para trama de carácter, el campo Comprobación de Error contiene un valor de 16 bits implementado como dos bytes de 8 bits.

El valor de comprobación de error es el resultado de un cálculo Comprobación Cíclica Redundante (CRC) realizado sobre el contenido del mensaje. El campo CRC es añadido al mensaje como último campo del mensaje. La forma de hacerlo es, añadir primero el byte de orden bajo del



campo, seguido del byte de orden alto. El byte de orden alto del CRC es el último byte a enviar en el mensaje.

## 2.7 CÓMO SON TRANSMITIDOS LOS CARACTERES EN SERIE

Cuando los mensajes son transmitidos sobre redes serie estándar MODBUS, cada carácter o byte es enviado en este orden (izquierda a derecha):

Bit Menos Significativo (LSB)... Bit Mas Significativo (MSB)

Con trama de caracteres ASCII, la secuencia de bit es:

**Figura 12: Secuencia de Bits en la trama ASCII.**



Con trama de carácter RTU, la secuencia de bit es:

**Figura 13: Secuencia de Bits en la trama RTU.**



## 2.8 MÉTODOS DE COMPROBACIÓN DE ERROR

Las redes series estándar MODBUS utilizan dos tipos de comprobación de error. La comprobación de paridad (par o impar) que puede ser aplicada opcionalmente a cada carácter y la comprobación de la trama (LRC o CRC) que es aplicada al mensaje completo. Ambas comprobaciones, de carácter y de trama de mensaje son generadas en el dispositivo maestro y aplicadas a los contenidos del mensaje antes de la transmisión. El dispositivo esclavo comprueba cada carácter y la trama del mensaje completo durante la recepción.

El maestro es configurado por el usuario para aguardar durante un tiempo de espera predeterminado antes de abortar la transacción. Este intervalo es establecido para ser lo suficientemente largo para que cualquier esclavo responda normalmente. Si el esclavo detecta un error de transmisión, el mensaje no será tenido en cuenta. El esclavo no construirá una respuesta

para el maestro. Así el tiempo de espera expirará y permite al programa del maestro tratar el error. Observe que un mensaje direccionado a un dispositivo esclavo inexistente también causará un error de tiempo excedido - time out -.

Otras redes tales como MAP y MODBUS Plus utilizan comprobación de trama a un nivel por encima del contenido MODBUS del mensaje. En esas redes, el campo de comprobación LRC o CRC del mensaje MODBUS no se aplica. En caso de error de transmisión, el protocolo de comunicación específico a esas redes notifica al dispositivo que inició la comunicación que ha ocurrido un error y le permite reintentar o abortar de acuerdo a cómo ha sido configurado. Si el mensaje ha sido enviado, pero el dispositivo esclavo no puede responder, puede ocurrir un error de tiempo excedido que puede ser detectado por el programa del maestro.

**2.8.1 Control de Paridad** Los usuarios pueden configurar los controladores para Control de paridad Par o Impar, o Sin Control de paridad. Esto determinará cómo será iniciado el bit de paridad en cada carácter. Si se especifica cualquier control de paridad Par o Impar, se contabilizará la cantidad de bits que tienen valor 1 en la porción de datos de cada carácter (siete bits de datos para modo ACSII, u ocho para RTU). Al bit de paridad

habrá de darse valor 0 o 1, para que se obtenga finalmente un número par o impar, respectivamente, de bits con valor 1.

Por ejemplo, estos 8 bits de dato forman parte de una trama de carácter RTU:

1100 0101

La cantidad de bits de valor 1 en el dato es cuatro. Si se utiliza Control de Paridad Par, el bit de paridad de la trama debe establecerse a valor 0, haciendo que la cantidad de bits de valor 1 siga siendo un número par (cuatro). Si se utiliza Control de Paridad Impar, el bit de paridad deberá tener valor 1, resultando una cantidad de bits de valor 1, impar (cinco). Cuando el mensaje es transmitido, el bit de paridad es calculado y aplicado a la trama de cada carácter. El dispositivo receptor cuenta la cantidad de bits de valor 1 y establece un error si no coincide la paridad con la configurada para ese dispositivo (todos los dispositivos en la red MODBUS deben ser configurados para usar el mismo método de Control de paridad). Obsérvese que la comprobación de paridad sólo detecta si un número impar de bits se han alterado en una trama de caracteres durante la transmisión. Por ejemplo, si se utiliza control de paridad Impar y dos bits de valor 1 de un carácter que tiene en origen 3 bits con valor 1, han quedado falseados (pasan a valor 0)

durante la transmisión, el resultado es todavía un cómputo impar de bits de valor 1 (y por lo tanto el error no es detectado por este método).

Si se especifica control No Paridad, no se transmite bit de paridad y no se hace comprobación de paridad. Se transmite un bit de paro adicional para rellenar la trama de caracteres.

**2.8.2 Comprobación LRC** En modo ASCII, los mensajes incluyen un campo de comprobación de error que está basado en un método de Comprobación Longitudinal Redundante (LRC). El campo LRC controla el contenido del mensaje, a excepción de los ':' del comienzo y el par CRLF. Es aplicado con independencia de cualquier método de control de paridad utilizado para los caracteres individuales del mensaje.

El campo LRC es un byte, conteniendo un valor binario de ocho bits. El valor LRC es calculado por el dispositivo emisor, que añade el LRC al mensaje. El dispositivo receptor calcula el LRC durante la recepción del mensaje y compara el valor calculado con el valor recibido en el campo LRC. Si los dos valores no son iguales, resulta un error.

El valor LRC se calcula sumando entre sí los sucesivos bytes del mensaje, descartando cualquier acarreo y luego complementando a dos el valor resultante. Se realiza sobre el contenido del campo de mensaje ASCII

excluyendo el carácter ‘:’ de comienzo del mensaje y excluyendo el par CRLF de final de mensaje.

En la lógica de programación de controladores, la función CKSM calcula el LRC en base al contenido del mensaje.

**2.8.3 Comprobación CRC** En modo RTU, los mensajes incluyen un campo de comprobación de error que está basado en un método Comprobación de Redundancia Cíclica (CRC). El campo CRC controla el contenido del mensaje completo. Se aplica con independencia de cualquier método de control de paridad utilizado para los caracteres individuales del mensaje.

El campo CRC es de dos bytes, conteniendo un valor binario de 16 bits. El valor CRC es calculado por el dispositivo emisor, que añade el CRC al mensaje. El dispositivo receptor calcula el CRC durante la recepción del mensaje y compara el valor calculado con el valor recibido en el campo CRC. Si los dos valores no son iguales, resulta un error.

Para calcular el valor CRC MODBUS se precarga un registro de 16 bits, todos ellos a 1. Luego comienza un proceso que toma los sucesivos bytes del mensaje y los opera con el contenido del registro y actualiza éste con el

resultado obtenido. Sólo los 8 bits de dato de cada carácter son utilizados para generar el CRC. Los bits de arranque y paro y el bit de paridad, no se tienen en cuenta para el CRC.

Durante la generación del CRC, se efectúa una operación booleana OR exclusivo (XOR) a cada carácter de 8 bits con el contenido del registro. Entonces al resultado se le aplica un desplazamiento de bit en la dirección del bit menos significativo (LSB), rellenando la posición del bit más significativo (MSB) con un cero. El LSB es extraído y examinado. Si el LSB extraído fuese un 1, se realiza una XOR entre el registro y un valor fijo preestablecido (\*). Si el LSB fuese un 0, no se efectúa la operación.

Este proceso es repetido hasta haber cumplido 8 desplazamientos. Después del último desplazamiento (el octavo), el próximo byte es operado XOR con el valor actual del registro y el proceso se repite con ocho desplazamientos más, como se ha descrito mas arriba y así con todos los bytes del mensaje. El contenido final del registro, después de que todos los bytes del mensaje han sido procesados, es el valor del CRC.

Cuando el CRC es añadido al mensaje, primero se añade el byte de orden bajo, seguido del byte de orden alto.

En la lógica de programación de controladores, la función CKSM calcula el CRC en base al contenido del mensaje.

(\*) El valor preestablecido es A001 hex., correspondiente al polinomio generador CRC16 'Inverso', que es el que se aplica al CRC MODBUS.

## **2.9 FORMATOS DE FUNCIONES MODBUS**

**2.9.1 Direcciones en los Mensajes MODBUS** Todos las direcciones en los mensajes Modbus son referenciadas a cero. La primera unidad de cada tipo de dato es direccionada como parte número cero. Por ejemplo:

La bobina conocida como 'bobina 1' en un controlador programable es direccionada como bobina

0000 en el campo de dirección de un mensaje Modbus.

La bobina 127 decimal es direccionada como bobina 007E hex. (126 decimal).

El registro mantenido 40001 es direccionado como registro 0000 en el campo de dirección de un mensaje MODBUS. El campo código de función ya



especifica una operación sobre un 'registro mantenido'. Por lo tanto la referencia '4XXXX' está implícita.

El registro mantenido 40108 es direccionado como registro 006B hex (107 decimal).

**2.9.2 Campos Contenidos En Los Mensajes MODBUS** La figura 14 muestra un ejemplo de un mensaje de petición Modbus. La figura 15 es un ejemplo de una respuesta normal. Ambos ejemplos muestran los campos contenidos en hexadecimal y también cómo estaría distribuido en la trama, un mensaje en modo ASCII o en modo RTU.

La petición del maestro es una solicitud de Lectura de Registros Mantenedos, al dispositivo esclavo con dirección 06. El mensaje solicita datos numéricos de tres registros mantenidos, 40108 al 40110.

Observe que el mensaje especifica la dirección de comienzo como 0107 (006B hex).

La respuesta del esclavo replica el código de función, indicando que esto es una respuesta normal.

El campo Cómputo de Bytes especifica cuántas unidades de datos de 8 bits se devuelven.

Muestra la cantidad de bytes de datos que vienen a continuación, bien ASCII o RTU. En el modo ASCII, este valor representa la mitad del cómputo real de caracteres ASCII en el dato, ya que en este modo, cada dígito hexadecimal de 4 bits requiere un carácter ASCII y por lo tanto, debe haber dos caracteres ASCII para contener cada unidad de dato de 8 bits.

Por ejemplo, el dato: 63 hex se envía como un byte – ocho bits - en modo RTU (0110 0011). El mismo valor, enviado en modo ASCII requiere dos caracteres ASCII, el ASCII '6' (011 0110) y el ASCII '3' (011 0011).

El campo Cómputo de bytes contabiliza este dato como un solo dato de 8 bits, con independencia del método de trama de carácter (ASCII o RTU).

La figura 15 muestra cómo se implementa el cómputo de Bytes en una respuesta típica:

**Figura 14: Petición Del Maestro Con Trama ASCII/RTU.**

PETICION	Datos del Ejemplo (Hex)	Caracteres ASCII	Campo 8-Bits RTU
Nombre del Campo			
Cabecera		: ('dos puntos')	Ninguno
Dirección del Esclavo	06	06	0000 0110
Función	03	03	0000 0011
Dirección Comienzo Alto	00	00	0000 0000
Dirección Comienzo Bajo	6B	6B	0110 1011
No. de Registros Alto	00	00	0000 0000
No. De Registros Bajo	03	03	0000 0011
Comprobación de Error		LRC (2 caract)	CRC (16 bits)
Terminación		CR LF	Ninguno
	Total Bytes:	17	8

**Figura 15: Respuesta De Un Esclavo Con Trama ASCII/RTU.**

RESPUESTA	Datos del Ejemplo (Hex)	Caracteres ASCII	Campo 8-Bits RTU
Nombre del Campo			
Cabecera		: ('dos puntos')	Ninguno
Dirección del Esclavo	06	06	0000 0110
Función	03	03	0000 0011
Computo de Bytes	06	06	0000 0110
Dato Alto	02	02	0000 0010
Dato Bajo	2B	2B	0010 1011
Dato Alto	00	00	0000 0000
Dato Bajo	00	00	0000 0000
Dato Alto	00	00	0000 0000
Dato Bajo	63	63	0110 0011
Comprobación de Error		LRC (2caract)	CRC (16 bits)
Terminación		CR LF	Ninguno
	Total Bytes:	23	11

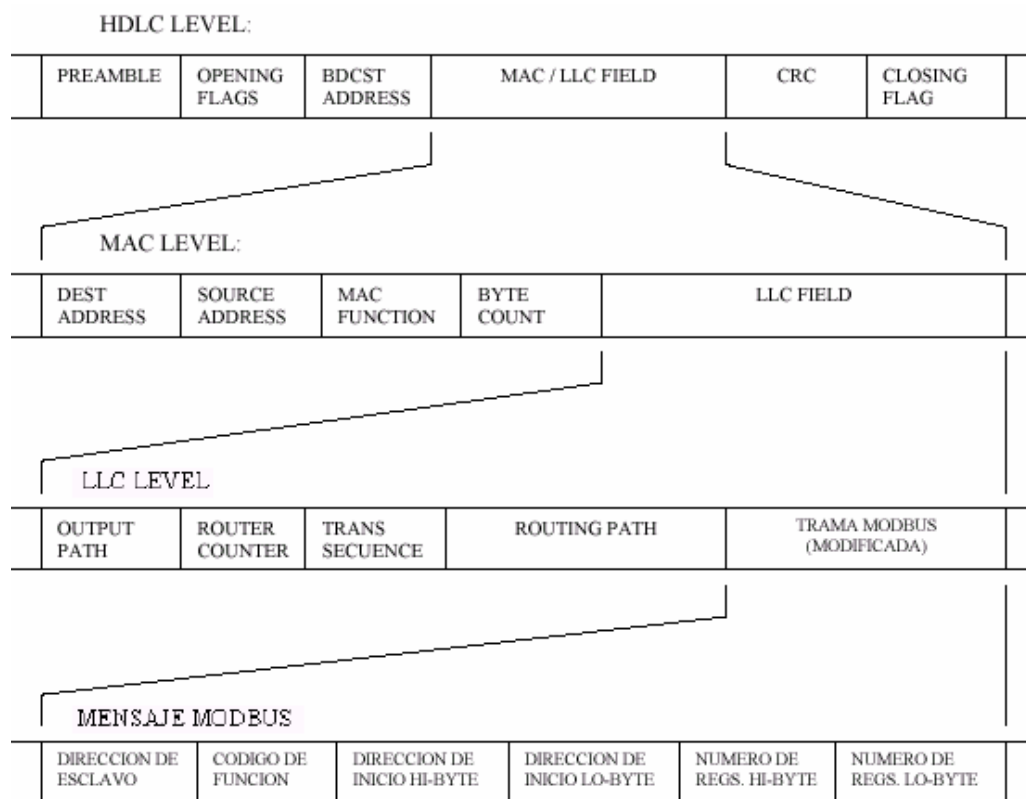
**2.9.3 Campos Contenidos En Los Mensajes Sobre MODBUS Plus** Los mensajes Modbus enviados sobre redes MODBUS Plus están integrados en la trama dentro del nivel Logical Link Control (LLC). Los campos del mensaje Modbus sobre Modbus Plus consisten en bytes de 8 bits de forma similar a la utilizada con la trama RTU.

El campo Dirección del Esclavo es convertido en un enrutamiento *ROUTING PATH* por el dispositivo emisor. El campo CRC no se envía en el mensaje Modbus, porque sería redundante con la comprobación CRC realizada en el nivel High-level Data Link Control (HDLC).

El resto del mensaje permanece como en el formato serie estándar. El software de aplicación (Bloques MSTR en los controladores, o Modcom III en ordenadores) manipula la trama del mensaje en una red de paquetes.

La figura 16 muestra cómo se integraría una petición Leer Registros Mantenedos en una trama para transmisión Modbus Plus.

**Figura 16: Campos En MODBUS Plus.**



## 2.10 CATEGORÍAS DE LOS CÓDIGOS DE FUNCIÓN

Hay tres categorías de códigos de función MODBUS y son:

### Códigos De Función Públicos:

- Son códigos de función bien definidos.
- Están garantizados para ser únicos
- Validados por la comunidad MODBUS.ORG
- Están públicamente documentados
- Tienen tests de ajuste.

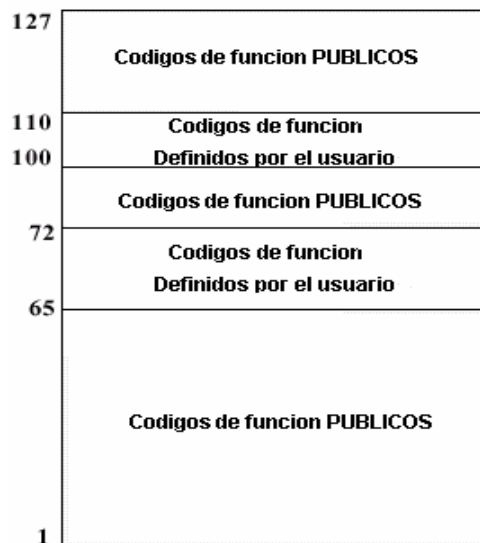
### **Códigos De Función Definidos Por El Usuario.**

- El usuario puede implementar un código de función sin la aprobación de MODBUS.ORG
- No hay garantía de que el uso de la función seleccionada podría ser única.
- El usuario podría hacer una petición para que su código sea un código público

### **Códigos De Funciones Reservados**

Son códigos de funciones usados por algunas compañías que tienen productos propietarios y no son disponibles para el uso público hay que aclarar que estos se encuentran en las mismas posiciones que las definidas por el usuario.

**Figura 17: Categorías de los códigos de función MODBUS.**



**2.10.1 Definición De Los Códigos De Función.** En La siguiente tabla podemos ver los códigos de función públicos.

**Figura 18: Códigos De Función Públicos MODBUS.**

<b>Acceso a datos</b>	<b>Acceso a Bits</b>	<b>Registro de entradas físicas</b>	<b>Leer entrada discreta</b>	<b>02</b>
		<b>Bits internos o bobinas físicas</b>	<b>Leer bobinas</b>	<b>01</b>
			<b>Escribir una bobina</b>	<b>05</b>
			<b>Escribir bobinas multiples</b>	<b>15</b>
	<b>Registro de entrada física</b>	<b>Leer registro de entrada</b>	<b>04</b>	
	<b>Acceso de 16 bits</b>	<b>Registros internos o registros de salida física</b>	<b>Leer registros multiples</b>	<b>03</b>
			<b>Escribir un solo registro</b>	<b>06</b>
			<b>Escribir multiples registros</b>	<b>16</b>
			<b>Leer/escribir multiples registros</b>	<b>23</b>
			<b>Enmascarar registro de escritura</b>	<b>22</b>
	<b>Acceso a archivos de registro</b>		<b>Leer archivo de registro</b>	<b>20</b>
		<b>Escribir archivo de registro</b>	<b>21</b>	
<b>Interfaz encapsulada</b>			<b>Leer identificacion del aparato</b>	<b>43</b>

## 2.11 RESPUESTAS DE EXCEPCIÓN

Excepto para mensajes tipo difusión, cuando un dispositivo maestro envía una petición a un dispositivo esclavo, espera una respuesta normal. Uno de cuatro posibles eventos puede ocurrir desde la petición del maestro:

Si el dispositivo esclavo recibe la petición sin error de comunicación y puede manejar la petición normalmente, devuelve una respuesta normal.

Si el esclavo no recibe la petición debido a un error de comunicación, no hay devolución de respuesta. El programa del maestro eventualmente procesará una condición de tiempo excedido - time out -, para la petición.

Si el esclavo recibe la petición, pero detecta un error de comunicación (paridad, LRC, o CRC), no hay devolución de respuesta. El programa del maestro eventualmente procesará una condición de tiempo excedido – time out -, para la petición.

Si el esclavo recibe la petición sin error de comunicación, pero no puede manejarla (por ejemplo, si la solicitud es leer una bobina o registro inexistente), el esclavo devolverá una respuesta de excepción informando al maestro de la naturaleza del error.

El mensaje de respuesta de excepción tiene dos campos que lo diferencian de una respuesta normal:

**2.11.1 Campo de Código de Función:** En una respuesta normal el esclavo replica el código de función de la petición original en el campo del código de función de la respuesta. Todos los códigos de función tienen el bit más significativo (MSB) a 0 (sus valores están por debajo de 80 hexadecimal). En una respuesta de excepción, el esclavo establece el MSB



del código de función a 1. Esto hace que el código de función en una respuesta de excepción resulte 80 hexadecimal más alto de lo que sería para una respuesta normal.

Con el MSB del código de función activado, el programa de aplicación del maestro puede reconocer la respuesta de excepción y puede examinar en el campo de datos el código de excepción.

**2.11.2 Campo de Datos:** En una respuesta normal, el esclavo puede devolver datos o estadísticas en el campo de datos (cualquier información que fuera solicitada en la petición). En una respuesta de excepción, el esclavo devuelve un código excepción en el campo de datos. Esto define la condición del esclavo que causó la excepción.

La figura 18 muestra un ejemplo de una petición del maestro y una respuesta de excepción de un esclavo. Los campos del ejemplo se muestran en hexadecimal.

**Figura 19: Petición Del Maestro Y Respuesta De Excepción Del Esclavo**

PETICION		Datos del Ejemplo (Hex)
Byte	Nombre del Campo	
1	Dirección del Esclavo	0A
2	Función	01
3	Dirección Comienzo Alto	04
4	Dirección Comienzo Bajo	A1
5	No. de Bobinas Alto	00
6	No. de Bobinas Bajo	01
7	LRC	4F
RESPUESTA DE EXCEPCION		Datos del Ejemplo (Hex)
Byte	Nombre del Campo	
1	Dirección del Esclavo	0A
2	Función	81
3	Código de Excepción	02
4	LRC	4F

En este ejemplo, el maestro direcciona una petición al dispositivo esclavo 10 (0A hex.).

El código de función (01) corresponde a una operación Leer el Estado de una Bobina. Se devuelve el estado de la bobina en la dirección 1245 (04A1 hex.).

Observe que sólo se lee una bobina, como se ha especificado en el campo del número de bobinas (0001). Si la dirección de la bobina es inexistente en el dispositivo esclavo, el esclavo devolverá la respuesta de excepción con el código de excepción mostrado (02). Eso especifica un dato de dirección

ilegal para el esclavo. Por ejemplo, si el esclavo es un controlador serie 984-385 con 512 bobinas, podría devolver este código.

**Figura 20: Listado De Códigos De Excepción**

<b>Codigo</b>	<b>Nombre</b>	<b>Significado</b>
01	ILLEGAL FUNCTION	El codigo de funcion recibido en la peticion no es una accion permitida para el esclavo. Si un comando Completar Seleccion de Programa fue notificado, este codigo indica que no le precedio funcion programa.
02	ILLEGAL DATA ADDRESS	El dato de direccion recibido en la peticion no es una direccion permitida para el esclavo
03	ILLEGAL DATA VALUE	Un valor contenido en el campo de datos de la peticion no es un valor admisible para el esclavo.
04	SLAVE DEVICE FAILURE	Ha ocurrido un error no recuperable mientras el esclavo estaba intentando ejecutar la accion solicitada.
05	ACKNOWLEDGE	El esclavo ha aceptado la peticion y está procesandola, pero requerira un tiempo largo para hacerlo. Esta respuesta se devuelve para prevenir que ocurra un error de tiempo excedido en el maestro. El maestro puede notificar a continuacion un mensaje Completar Seleccion de programa para determinar si se ha completado el procesamiento.
06	SLAVE DEVICE BUSY	El esclavo está ocupado procesando un comando de programa de larga duracion. El maestro deberia retransmitir el mensaje mas tarde cuando el esclavo esté libre.
08	MEMORY PARITY ERROR	El esclavo ha intentado leer memoria extendida, pero ha detectado un error de paridad en la memoria. El maestro puede reintentar la peticion, pero el servicio debe ser requerido al dispositivo esclavo.

### 3. CAPA FÍSICA MODBUS

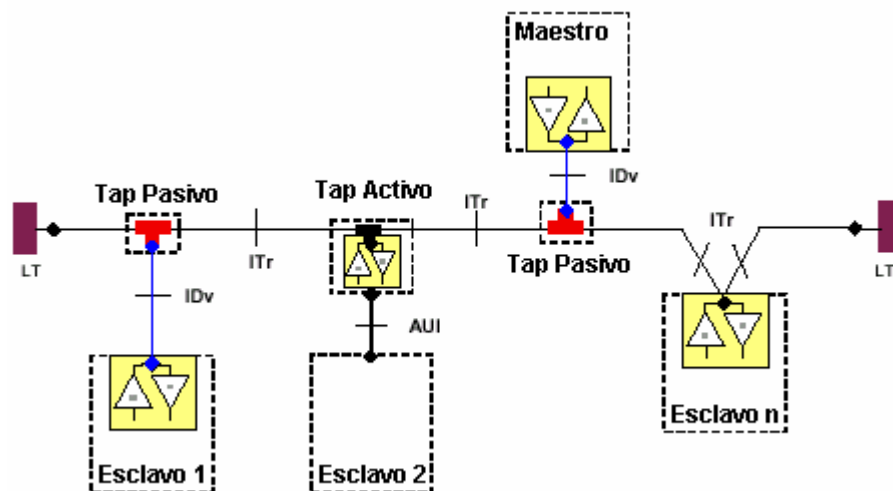
#### 3.1 GENERALIDADES

El interfaz MODBUS esta hecha para trabajar en concordancia con el estándar RS-485. Este permite sistemas multipunto en una configuración de dos cables adicionalmente algunos elementos que lo usan pueden usar una interfaz de cuatro cables e inclusive se puede manejar el interfaz RS-232.

#### 3.2 INTERFAZ ELECTRICA

**3.2.1 Infraestructura De Bus Serial Multipunto.** En la figura se presenta la infraestructura de un sistema serial MODBUS.

**Figura 21: Infraestructura Del Bus Serial.**



Un bus serie MODBUS multipunto está constituido de un cable principal, y posiblemente algunas derivaciones. Las terminaciones de líneas (LT) son necesarias en cada extremo del cable para la adaptación de impedancias.

Como es mostrado en la figura, diferentes implementaciones pueden ínter operar en el mismo sistema serial MODBUS.

- Referente al maestro y al esclavo 1:

Los aparatos tienen integrado un trceptor que es conectado al cable usando un Tap pasivo y un cable de derivación (IDv).

- Referente al esclavo 2:

El aparato no tiene integrado un trceptor y este es conectado al cable usando una interfaz de conexión unitaria y un Tap activo que tiene integrado el trceptor.

- Referente esclavo n:

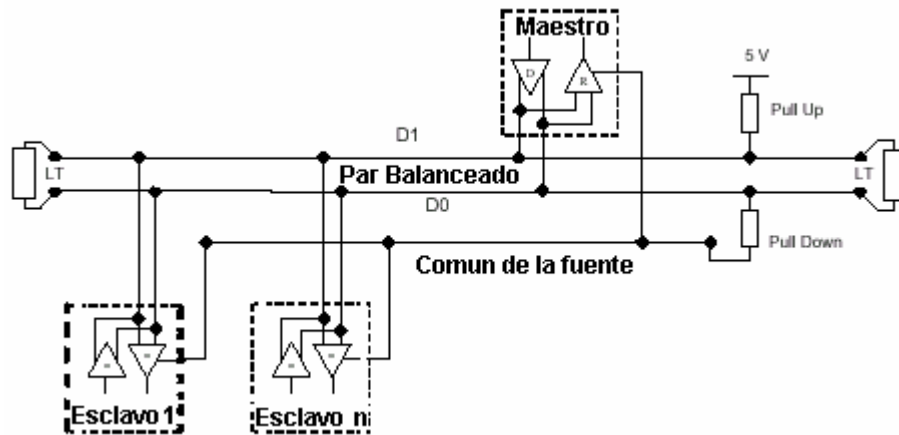
El aparato es conectado directamente al cable.

**3.2.2 Definición De Protocolo MODBUS De Dos Cables.** Una solución MODBUS sobre líneas seriales podría implementarse con una interfaz de dos cables de acuerdo con el estándar EIA/TIA-485.

En esta configuración solamente un manejador tiene el derecho a transmitir.

Realmente decir 2 cables (2W-bus) es incorrecto ya que físicamente existe siempre un cable adicional que sería el Común.

**Figura 22: Topología General De Dos Cables**



**Figura 23: Definición de Circuitos 2W**

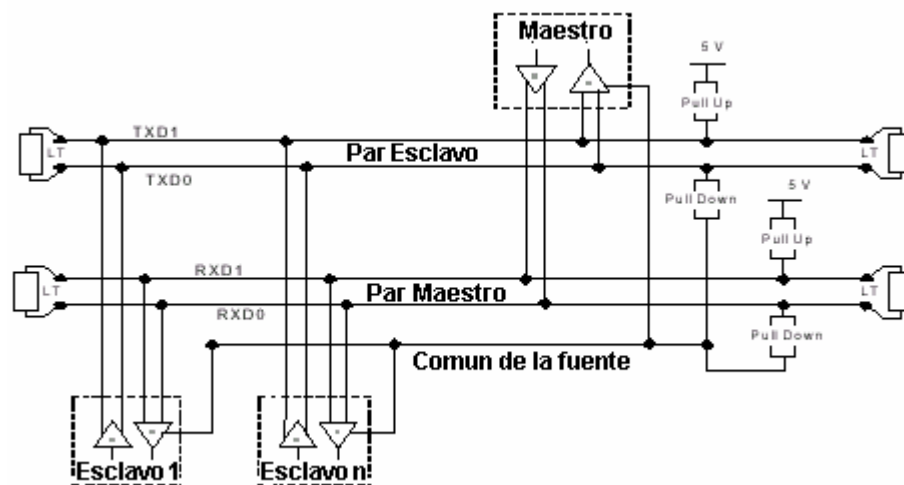
Definición De Circuitos 2W-MODBUS					
Circuitos requeridos		Para El Aparato	Requeridos En el aparato	Nombre EIA/TIA-485	Descripción
En ITr	En IDv				
D1	D1	I/O	X	B/B'	Terminal Tranceptor 1, Voltaje V1 (V1 > V0 para el binario 1 estado [OFF])
D0	D0	I/O	X	A/A'	Terminal Tranceptor 0, Voltaje V0 (V0 > V1 para el binario 0 estado [ON])
Comun	Comun	--	X	C/C'	Comun de la fuente

### 3.2.3 Definición De Protocolo MODBUS De Cuatro Cables Opcional.

Opcionalmente MODBUS también permite implementar dos pares de buses (4W) de datos monodireccional. Los datos en el par maestro solo recibidos por los esclavos; y los datos en el par esclavo son solo recibidos por el único maestro.

Aquí la discrepancia sobre el número de cables es válida ya que un quinto cable físico existe y tiene la misma identidad que en el sistema 2W, el Común. En el mismo sentido que el sistema de dos cables solo un aparato tiene el derecho de emisión.

**Figura 24: Topología General De Cuatro Cables**



**Figura 25: Definición de Circuitos 4W**

Definicion De Circuitos 4W-MODBUS					
Circuitos requeridos		Para El Aparato	Requeridos En el aparato	Nombre EIA/TIA-485	Descripcion
En ITr	En IDv				
TXD1	TXD1	OUT	X	B	Terminal Generador 1, Voltaje Vb (Vb > Va para el binario 1 estado [OFF])
TXD0	TXD0	OUT	X	A	Terminal Generador 0, Voltaje Va (Vb > Va para el binario 0 estado [ON])
RXD1	RXD1	IN	(1)	B'	Terminal Receptor 1, Voltaje Vb' (Vb' > Va' para el binario 1 estado [OFF])
RXD0	RXD0	IN	(1)	A'	Terminal Receptor 0, Voltaje Va' (Va' > Vb' para el binario 0 estado [ON])
Comun	Comun	--	X	C/C'	Comun de la fuente

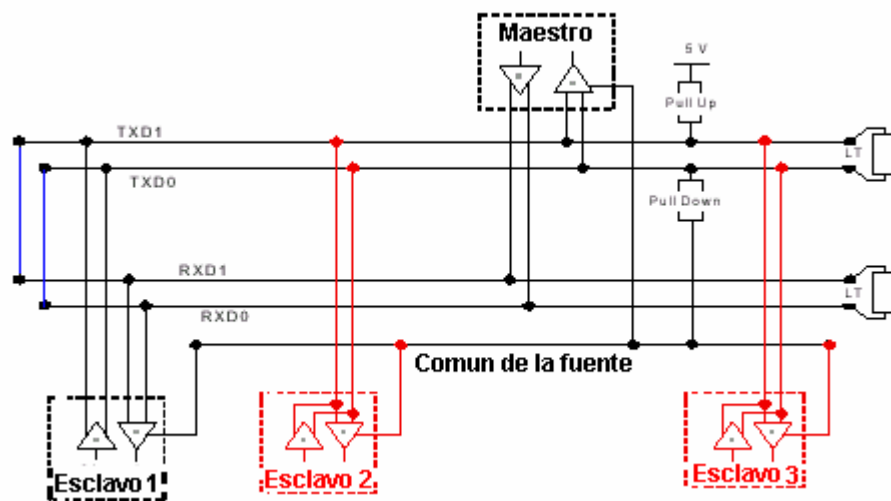
**3.2.4 Compatibilidad Entre Los Sistemas De Cuatro Y Dos Cables (4w Y 2w).** En función de conectar aparatos implementados en una interfaz de dos cables, en una existente de cuatro cables pueden ser modificados como es descrito:

- TxD0 tiene que estar conectado a RxD0 y a DO(del 2W)
- TxD1 tiene que estar conectado a RxD1 y a D1(del 2W)
- Los resistores de Pull up y Pull down y las LT deben ser adaptados a las señales DO y D1



En la siguiente figura hay un ejemplo donde los esclavos 2 y 3, que usan una interfaz 2W pueden operar con el maestro y el esclavo 1 con el uso de una interfaz 4W.

**Figura 26: Conexión De Un Sistema De 2W A 4W.**

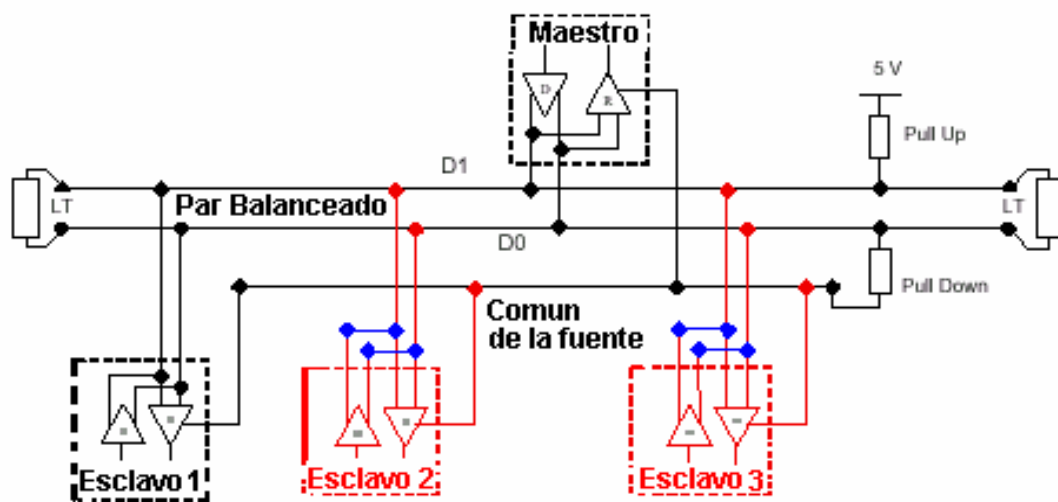


En función de conectar aparatos implementados en una interfaz de cuatro cables, en una existente de dos cables pueden ser modificados como es descrito:

- TxD0 tiene que estar conectado a RxD0 y está unido a DO(del 2W)
- TxD1 tiene que estar conectado a RxD1 y está unido a D1(del 2W)

En la siguiente figura hay un ejemplo donde los esclavos 2 y 3, que usan una interfaz 4W pueden operar con el maestro y el esclavo 1 con el uso de una interfaz 2W.

**Figura 27: Conexión De Un Sistema De 4W A 2W.**



**3.2.5 Definición De MODBUS En RS-232** Algunos aparatos Pueden implementar una interfaz RS-232 entre un DCE y un DTE.

**Figura 28: Definición De Los Circuitos MODBUS En RS-232**

Señal	Para DCE	Requerido en DCE (1)	Requerido en DTE (1)	Descripción
<b>Comun</b>	--	X	X	<b>Comun de la fuente</b>
CTS	In			Clear to Send
DCD	--			Data Carrier Detected ( from DCE to DTE )
DSR	In			Data Set Ready
DTR	Out			Data Terminal Ready
RTS	Out			Request to Send
RXD	In	X	X	<b>Received Data</b>
TXD	Out	X	X	<b>Transmitted Data</b>

Las señales anteriores pertenecen al estándar EIA/TIA 232 y las marcadas con X son requeridas para RS-232 MODBUS.

Cada Señal TXD debe ser conectada con RXD en el otro aparato.

RTS debe ser conectada con CTS en el otro aparato.

DTR debe ser conectada con DSR en el otro aparato.

### **3.3 REQUERIMIENTOS DE SISTEMAS MULTIPUNTO**

Para cualquier sistema RS-485 multipunto, en cualquiera de las configuraciones 2W ó 4W los siguientes requerimientos aplican.

**3.3.1 Máximo Numero De Aparatos Sin Repetidor.** Se tiene como ideal usar un repetidor cada 32 aparatos MODBUS dependiendo de Todas las direcciones posibles y La carga que cada aparato tenga en RS-485.

Un sistema RS-485 puede implementar un número grande de aparatos. Algunos permiten la implementación de un sistema serial MODBUS con más de 32 aparatos sin repetidor, pero para eso se necesita consultar la documentación de cada fabricante.

**3.3.2 Topología.** Una configuración MODBUS RS-485 sin repetidor tiene un cable troncal al cual los aparatos están conectados directamente ó por cables derivados cortos. Ese cable troncal es llamado bus, puede ser largo y sus dos extremos deben estar conectados a terminadores de líneas.

**3.3.3 Longitud** La longitud de extremo a extremo del cable troncal debe ser limitada. La máxima longitud es función de la rata de Baudios, el Cable usado, el número de cargas y la definición del cableado (2W ó 4W). Para un Máximo de 9600 baudios y un AWG26 ó más grande, la longitud máxima es de 1000m en el caso que se use la compatibilidad entre los sistemas de cuatro y dos cables (figura 24) La longitud máxima es de la mitad.

Las derivaciones deben ser cortas nunca más de 20m.

**3.3.4 Conexiones De Tierra.** El circuito “común” (Común de la fuente) debe ser conectado directamente a una tierra de protección, preferiblemente, en un solo punto para todo el bus.

Generalmente este punto es escogido en el aparato maestro.

**3.3.5 Terminaciones de líneas.** Una reflexión en una línea de transmisión es el resultado de una discontinuidad de impedancia que una onda ve cuando está viajando por toda la línea. Para minimizar las reflexiones desde el final del cable es requerido poner terminaciones de línea en cada extremo del bus.

Es importante que la línea sea terminada en ambos extremos cuando la propagación es bidireccional, pero no es permitido poner mas de dos terminaciones en un par balanceado esto quiere decir que no se ponen LT en los cables de derivación.

Cada LT debe ser conectado entre los dos conductores del par balanceado

Las LT deben ser una resistencia de 150 ohmios a 0.5 W.

Un condensador en serie (1nF/10V mínimo) con una resistencia de 120 ohmios a 0.25 W es una mejor opción cuando una polarización del par es implementada.

En los sistemas 4W, cada par debe ser terminado en cada final del bus.

En los sistemas RS-232, no debe haber LT.

**3.3.6 Polarización de líneas.** Cuando no hay actividad de un par RS-485, las líneas no son manejadas y son susceptibles a interferencia y ruido.

Para asegurar que los receptores están en un estado constante, cuando ninguna señal de datos está presente, algunos aparatos necesitan polarizar la red. Por eso todos los aparatos MODBUS deben traer documentación para mirar si se necesita y como se puede implementar. Si uno o varios aparatos necesitan polarización, un par de resistores deben ser conectados en el par balanceado RS-485 así:

- Una resistencia a 5V en el D1 de Pull up.
- Una resistencia a 5V en el D1 de Pull down.

Los valores de estas resistencias deben ser de 450 ó 650 ohmios; con el valor de 650 ohmios se permiten más número de aparatos conectados. Cuando se polarice una línea esta se hace en un solo punto para todo el bus. Generalmente este punto es escogido en el aparato maestro. El máximo

número de aparatos autorizados en un bus serial MODBUS es reducido a cuatro si no hay polarización.

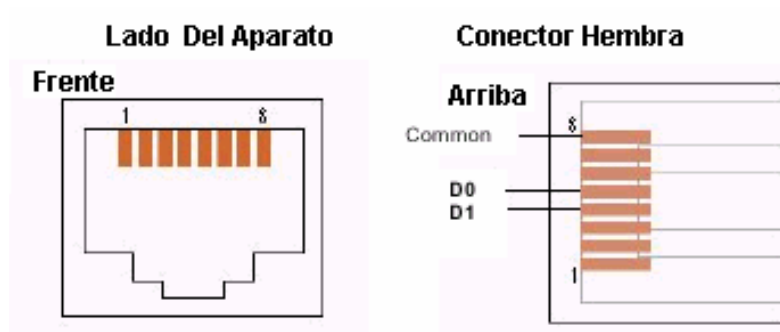
### 3.4 INTERFACES MECANICAS:

Si un terminal en RJ45 ó DB9 es usado como interfaces mecánicas, un conector hembra debe ser usado y uno macho en el otro extremo.

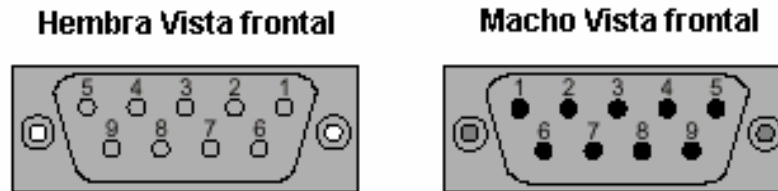
Cabe resaltar que una de las más importantes interfaces mecánicas son las terminales atornillables en las cuales cada fabricante define su Pinout.

#### 3.4.1 Diagramas De Pines De Los Conectores Para 2W MODBUS.

Figura 29 : Conector RJ45 Para 2w MODBUS.



**Figura 30 : Conector DB9 Para 2w MODBUS.**



**Figura 31: Pinout De Los Conectores Y Su Condicionamiento**

Pin on RJ45	Pin on D9-shell	nivel de requerimiento	IDv circuito	ITr circuito	RS-485 nombre	Descripcion para el IDv
3	3	opcional	PMC	--	--	Port Mode Control
4	5	requerido	D1	D1	B/B'	Terminal tranceptor 1 V1 Voltaje (V1>V0 para 1 binario estado [off])
5	9	requerido	D0	D0	A/A'	Terminal tranceptor 0 V0 Voltaje (V0>V1 para 0 binario estado [on])
7	2	recomendado	VP	--	--	Fuente positiva 5-24 VDC
8	1	requerido	comun	comun'	C/C'	comun Fuente

Estos circuitos deben ser respetados para cualquier implementación 2W MODBUS.



### 3.4.2 Diagramas De Pines De Los Conectores Para 4W MODBUS.

Figura 32 : Conector RJ45 Para 2W MODBUS.

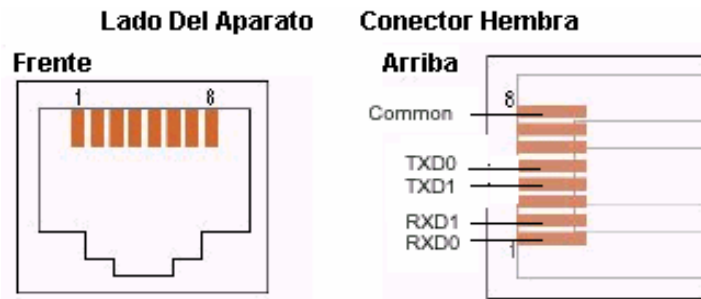


Figura 33 : Conector DB9 Para 4W MODBUS.

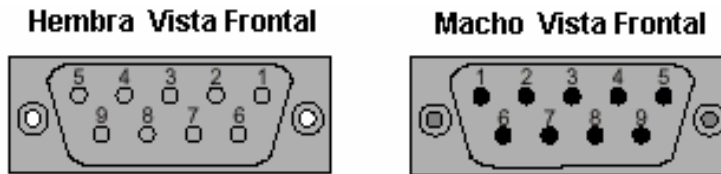


Figura 34: Pinout De Los Conectores Y Su Condicionamiento

Pin on RJ45	Pin on D9-shell	nivel de requerimiento	IDv circuito	ITr circuito	RS-485 nombre	Descripcion para el IDv
1	8	requerido	RXD0	RXD0	A'	Port Mode Control
2	4	requerido	RXD1	RXD1	B'	Terminal tranceptor 1 V1 Voltaje (V1>V0 para 1 binario estado [off])
3	3	opcional	PMC	--	--	Terminal tranceptor 0 V0 Voltaje (V0>V1 para 0 binario estado [on])
4	5	requerido	TXD1	TXD1	B	Terminal tranceptor 1 V1 Voltaje (V1>V0 para 1 binario estado [off])
5	9	requerido	TXD0	TXD0	A	Terminal tranceptor 0 V0 Voltaje (V0>V1 para 0 binario estado [on])
7	2	recomendado	VP	--	--	Fuente positiva 5-24 VDC
8	1	requerido	comun	comun'	C/C'	comun Fuente

Estos circuitos deben ser respetados para cualquier implementación 4W MODBUS.

**3.4.3 Diagramas De Pines De Los Conectores Para Rs-232 MODBUS.** En la siguiente figura se ilustran la forma de conexión ente dos aparatos MODBUS usando RS-232, mostrando los pines en cada una de las interfaces mecánicas.

**Figura 35: Diagrama De Pines Para MODBUS RS232.**

DCE			Circuito			DTE		
Pin en RJ45	Pin en D9-shell	nivel de requerimiento	Nombre	Descripción	RS232 Fuente	nivel de requerimiento	Pin en RJ45	Pin en D9-shell
1	2	Requerido	TXD	Dato Transmitido	DTE	Requerido	2	3
2	3	Requerido	RXD	Dato recibido	DCE	Requerido	1	2
3	7	Opcional	CTS	Clear to Send	DCE	Opcional	6	8
6	8	Opcional	RTS	Request to Send	DTE	Opcional	3	7
8	5	Requerido	Common	Comun de señal	--	Requerido	8	5

### **3.5 CABLES**

Un cable para soportar MODBUS debe ser blindado. En un extremo de este blindaje este debe ser conectado a una tierra de protección. Si un conector es usado en esta parte su carcasa debe estar conectada a tierra y al blindaje.

Para un sistema RS-485 MODBUS el alambre debe ser lo suficientemente grueso para permitir la máxima longitud, el calibre AWG 24 es suficiente para los datos MODBUS.

Los cables categoría 5 pueden operar para RS-485 para dar un máximo de longitud de 600m.

Para los pares balanceados usados en sistemas RS-485 una impedancia característica con un valor más alto de 100 ohms, deben ser preferidos especialmente si usan tasas de baudios mayores a 19200.

#### **4. CONCLUSIONES**

El Protocolo MODBUS es uno de los sistemas más básicos en los tipos de redes de campos pero muy útil en el sentido de trabajar muy directamente con la maquina y todas las ventajas a que esto conlleva. Además de esto es un protocolo que no exige unas capas físicas y de enlace muy complejas por consiguiente se pueden indexar estas ultimas en tecnologías de redes más complejas haciendo el alcance de MODBUS más amplio.

Tambien se podría en un futuro implementar en su totalidad, todo el protocolo en chips inteligentes como los microcontroladores, para crear un sistema Modbus completo y no simulado con computadores, ya que esto nos llevaría a conocer estar y muy cerca de las tecnologías de redes de campo actuales.

## **5. RECOMENDACIONES.**

El sistema MODBUS fue concebido básicamente para funcionar en cualquier sistema serial. Pero en esta, prima ante todo el sistema RS-485 el cual es un estándar muy usado en redes de campo actuales. En este sentido, sería muy bueno que la universidad contara con los recursos para implementar una red de campo basada en este estándar ya que se podría aprovechar muy bien para muchas practicas incluidas, por ejemplo, la practica del laboratorio aquí incluido de sistema multipunto que es generalmente la que más abunda en implementaciones industriales en nuestro medio. Con estos medios las simulaciones y laboratorios que se desarrollen con estos sistemas serán más realistas que con el actual RS-232.

## **BIBLIOGRAFÍA**

**FONSECA VELÁSQUEZ Aldemar** Y **FINO SANDOVAL Rafael**, Red De Datos Para Instrumentación, Fondo de Publicaciones Universidad Distrital francisco José de caldas, 2003 290 p.

**JIMÉNEZ BUENDÍA Manuel**, Comunicaciones Industriales Universidad Politécnica De Cartagena.

**HTTP://WWW.MOVBUS.ORG** Modbus Over Serial Line Specification And Implementation Guide V.1. 2002 44p

**HTTP://WWW.MOVBUS.ORG** Modbus Application Protocol Specification 45p

**MODICON, Industrial Automation Systems**, Modicon Modbus Protocol Reference Guide, 1996 115p

**CALTA COMPUTER SYSTEMS**, GETTING STARTED MODBUS SIMULATOR FOR WINDOWS 98/NT/2000/XP, 2004 21p

**HTTP://WWW.CAMIRESEARCH.COM**  
Data\_Com\_Basics/RS232\_standard.html

# ANEXOS