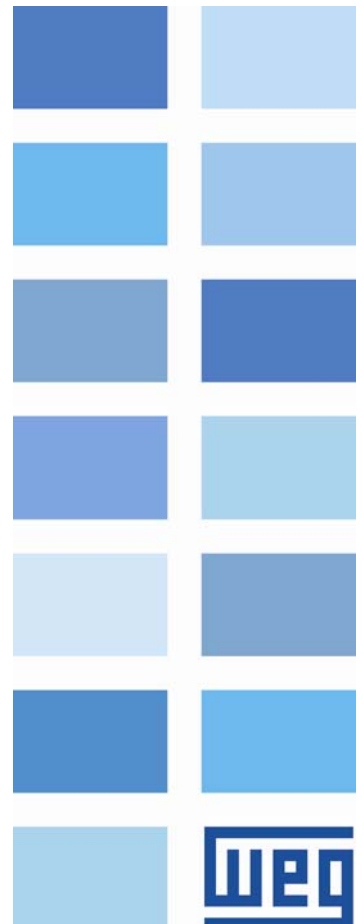


Modbus RTU

CFW701

Manual del Usuario





Manual del Usuario Modbus RTU

Serie: CFW701

Idioma: Español

N ° del Documento: 10001538637 / 00

Fecha de la Publicación: 02/2012

CONTENIDOS

CONTENIDOS.....	3
A RESPECTO DEL MANUAL	5
ABREVIACIONES Y DEFINICIONES	5
REPRESENTACIÓN NUMÉRICA.....	5
1 INTRODUCCIÓN A LA COMUNICACIÓN SERIAL	6
2 DESCRIPCIÓN DE LAS INTERFACES	7
2.1 RS485	7
2.1.1 Características de la interfaz RS485.....	7
2.1.2 Terminales del Conector	7
2.1.3 Resistor de terminación	7
2.1.4 Conexiones con la red RS485.....	7
3 PARAMETRIZACIÓN.....	9
3.1 SÍMBOLOS PARA DESCRIPCIÓN DE LAS PROPIEDADES	9
P0105 – SELECCIÓN 1ª/2ª RAMPA.....	9
P0220 – SELECCIÓN FUENTE LOCAL/REMOTO.....	9
P0221 – SELECCIÓN REFERENCIA LOCAL	9
P0222 – SELECCIÓN REFERENCIA REMOTA.....	9
P0223 – SELECCIÓN GIRO LOCAL	9
P0224 – SELECCIÓN GIRA/PARA LOCAL	9
P0225 – SELECCIÓN JOG LOCAL	9
P0226 – SELECCIÓN GIRO REMOTO.....	9
P0227 – SELECCIÓN GIRA/PARA REMOTO.....	9
P0228 – SELECCIÓN JOG REMOTO	9
P0308 – DIRECCIÓN SERIAL	9
P0310 – TASA DE COMUNICAÇÃO SERIAL	10
P0311 – CONFIGURACIÓN DE LOS BYTES DE LA INTERFAZ SERIAL	10
P0313 – ACCIÓN PARA ERROR DE COMUNICACIÓN	10
P0314 – WATCHDOG SERIAL	11
P0316 – ESTADO DE LA INTERFAZ SERIAL.....	11
P0680 – ESTADO LÓGICO	12
P0681 – VELOCIDAD DEL MOTOR EN 13 BITS.....	13
P0682 – PALABRA DE CONTROL VÍA SERIAL	13
P0683 – REFERENCIA DE VELOCIDAD VÍA SERIAL.....	14
P0695 – VALOR PARA LAS SALIDAS DIGITALES	15
P0696 – VALOR 1 PARA SALIDAS ANALÓGICAS	15
P0697 – VALOR 2 PARA SALIDAS ANALÓGICAS	15
4 PROTOCOLO MODBUS RTU	17
4.1 MODOS DE TRANSMISIÓN.....	17
4.2 ESTRUCTURA DE LOS MENSAJES EN EL MODO RTU.....	17
4.2.1 Dirección.....	17
4.2.2 Código de la Función.....	17
4.2.3 Campo de Datos	17
4.2.4 CRC	18
4.2.5 Tiempo entre Mensajes.....	18
5 OPERACIÓN EN LA RED MODBUS RTU – MODO ESCLAVO	19
5.1 FUNCIONES DISPONIBLES Y TIEMPOS DE RESPUESTA.....	19

5.2	MAPA DE MEMORIA.....	19
6	DESCRIPCIÓN DETALLADA DE LAS FUNCIONES	21
6.1	FUNCIÓN 03 – READ HOLDING REGISTER.....	21
6.2	FUNCIÓN 06 – WRITE SINGLE REGISTER.....	21
6.3	FUNCIÓN 16 – WRITE MULTIPLE REGISTERS	22
6.4	FUNCIÓN 43 – READ DEVICE IDENTIFICATION	22
6.5	ERRORES DE COMUNICACIÓN	23
7	FALLAS Y ALARMAS RELACIONADAS CON LA COMUNICACIÓN MODBUS RTU	
	25	
	A128/F228 – TIMEOUT EN LA RECEPCIÓN DE TELEGRAMAS	25
I.	APÉNDICES	26
	APÉNDICE A. TABLA ASCII	26
	APÉNDICE B. CÁLCULO DEL CRC UTILIZANDO TABLAS	27

A RESPECTO DEL MANUAL

Este manual provee la descripción necesaria para la operación del convertidor de frecuencia CFW701 utilizando las interfaces serie RS232 o RS485. Este manual debe ser utilizado en conjunto con el manual del usuario del CFW701.

ABREVIACIONES Y DEFINICIONES

ASCII	American Standard Code for Information Interchange
CRC	Cycling Redundancy Check
EIA	Electronic Industries Alliance
TIA	Telecommunications Industry Association
RTU	Remote Terminal Unit

REPRESENTACIÓN NUMÉRICA

Números decimales son representados a través de dígitos sin sufijo. Números hexadecimales son representados con la letra 'h' después del número.

1 INTRODUCCIÓN A LA COMUNICACIÓN SERIAL

En una interfaz serial los bits de datos son enviados de modo secuencial a través de un canal de comunicación o bus. Diversas tecnologías utilizan comunicación serial para la transferencia de datos, incluyendo las interfaces RS232 y RS485.

Las normas que especifican los padrones RS232 y RS485, sin embargo, no especifican el formato ni la secuencia de caracteres para la transmisión y recepción de datos. En este sentido, además de la interface, es necesario identificar también el protocolo utilizado para la comunicación. Entre los diversos protocolos existentes, un protocolo muy utilizado en la industria es el protocolo Modbus RTU.

2 DESCRIPCIÓN DE LAS INTERFACES

El convertidor de frecuencia CFW701 posee una interfaz RS485 estándar en el producto. A seguir son presentadas informaciones sobre la conexión e instalación del equipamiento en red.

2.1 RS485

2.1.1 Características de la interfaz RS485

- Interfaz sigue el estándar EIA-485.
- Puede operar como esclavo de la red Modbus RTU.
- Posibilita comunicación utilizando tasas de 9600 hasta 57600 Kbit/s.
- Interfaz aislada galvánicamente y con señal diferencial, confiriendo mayor robustez contra interferencia electromagnética.
- Permite la conexión de hasta 32 dispositivos en el mismo segmento. Una cantidad mayor de dispositivos puede ser conectada con el uso de repetidores¹.
- Longitud máxima del bus: 1000 metros.

2.1.2 Terminales del Conector

La conexión para la interfaz RS485 está disponible a través del conector XC1 utilizando los siguientes terminales:

Tabla 2.1: Terminales del conector para RS485

Terminal	Nombre	Función
10	A-Line (-)	RxD/TxD negativo
9	B-Line (+)	RxD/TxD positivo
8	GND	0V aislado del circuito RS485

2.1.3 Resistor de terminación

Para cada segmento de la red RS485, es necesario habilitar una resistencia de terminación en los puntos extremos del bus principal. El convertidor de frecuencia CFW701 posee llaves que pueden activarse (colocando ambas las llaves S2 en la posición ON) para habilitar la resistencia de terminación.

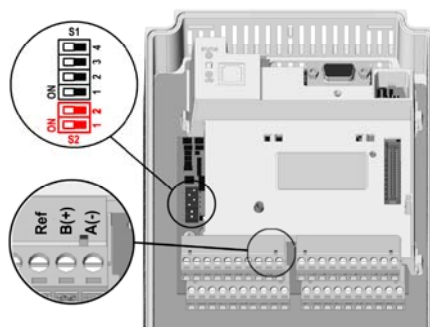


Figura 2.1: Resistencia de terminación y conector de la interfaz RS485

2.1.4 Conexiones con la red RS485

Para la conexión del convertidor de frecuencia CFW701 utilizando la interfaz RS485, los siguientes puntos deben ser observados:

- Es recomendado el uso de un cable con par tranzado blindado.
- Se recomienda también que el cable posea más un conductor para la conexión de la señal de referencia (GND). Caso el cable no posea el conductor adicional, se debe dejar la señal GND desconectado.
- La instalación del cable debe ser separado (y si posible lejos) del cableados de potencia.

¹ El número límite de equipos que pueden ser conectados en la red también depende del protocolo utilizado.

- Todos los dispositivos de la red deben estar debidamente puestos a tierra, de preferencia en la misma conexión con a tierra. El blindaje del cable también debe ser puesto a tierra.
- Habilitar los resistores de terminación solo en dos puntos, en los extremos del bus principal, mismo que existan derivaciones a partir del bus.

3 PARAMETRIZACIÓN

A seguir es presentado solo os parámetros del convertidor de frecuencia CFW701 que poseen relación con la comunicación Modbus RTU.

3.1 SÍMBOLOS PARA DESCRIPCIÓN DE LAS PROPIEDADES

RO	Parámetro solamente de lectura.
CFG	Parámetro solamente modificado con el motor parado.

P0105 – SELECCIÓN 1ª/2ª RAMPA

P0220 – SELECCIÓN FUENTE LOCAL/REMOTO

P0221 – SELECCIÓN REFERENCIA LOCAL

P0222 – SELECCIÓN REFERENCIA REMOTA

P0223 – SELECCIÓN GIRO LOCAL

P0224 – SELECCIÓN GIRA/PARA LOCAL

P0225 – SELECCIÓN JOG LOCAL

P0226 – SELECCIÓN GIRO REMOTO

P0227 – SELECCIÓN GIRA/PARA REMOTO

P0228 – SELECCIÓN JOG REMOTO

Estos parámetros son utilizados en la configuración de la fuente de los comandos para los modos de operación local y remota del convertidor de frecuencia CFW701. Para que el equipo sea controlado a través de la interfaz Modbus RTU, se debe seleccionar una de las opciones 'serial' disponibles en los parámetros.

La descripción detallada de estos parámetros se encuentra en el manual de programación del convertidor de frecuencia CFW701.

P0308 – DIRECCIÓN SERIAL

Rango de	0 a 255	Padrón: 1
Valores:		
Propiedades:	CFG	
Grupo de acceso vía HMI:	NET	

Descripción:

Permite programar la dirección utilizada para la comunicación serial del convertidor de frecuencia. Es necesario que cada equipo de la red posea una dirección distinta de las demás. Las direcciones válidas para este parámetro dependen del protocolo programado en el P0312:

- P0312 = 2 (Modbus RTU) → direcciones válidas: 1 a 247.
- P0312 = 3 (BACnet MS/TP) → direcciones válidas: 0 a 254.
- P0312 = 4 (N2) → direcciones válidas: 1 a 255.



¡NOTE!

El equipo debe ser inicializado cuando la dirección serial es cambiada.

P0310 – TASA DE COMUNICACIÓN SERIAL

Rango de Valores:	0 = 9600 bit/s 1 = 19200 bit/s 2 = 38400 bit/s 3 = 57600 bit/s	Padrón: 0
Propiedades:	CFG	
Grupo de acceso vía HMI:	NET	

Descripción:

Permite programar el valor deseado para la tasa de comunicación de la interfaz serial, en bits por segundo. Esta tasa debe ser la misma para todos los equipos conectados en la red.

P0311 – CONFIGURACIÓN DE LOS BYTES DE LA INTERFAZ SERIAL

Rango de Valores:	0 = 8 bits de datos, sin paridad, 1 stop bit 1 = 8 bits de datos, paridad par, 1 stop bit 2 = 8 bits de datos, paridad impar, 1 stop bit 3 = 8 bits de datos, sin paridad, 2 stop bits 4 = 8 bits de datos, paridad par, 2 stop bits 5 = 8 bits de datos, paridad impar, 2 stop bits	Padrón: 0
Propiedades:	CFG	
Grupo de acceso vía HMI:	NET	

Descripción:

Permite la configuración del número de bits de datos, paridad y stop bits en los bytes de la interfaz serial. Esta configuración debe ser la misma para todos los equipos conectados en la red.


¡NOTA!

Para el protocolo BACnet y N2 se debe seleccionar la opción 0 (padrón).

P0313 – ACCIÓN PARA ERROR DE COMUNICACIÓN

Rango de Valores:	0 = Inactivo 1 = Para por Rampa 2 = Deshabilita General 3 = Va para modo Local 4 = Va para modo Local y mantiene comandos y referencia 5 = Causa Falla	Padrón:
Propiedades:	CFG	
Grupo de acceso vía HMI:	NET	

Descripción:

Este parámetro permite seleccionar cual es la acción que debe ser ejecutada por el equipo, caso este sea controlado vía red y un error de comunicación sea detectado.

Tabla 3.1: Valores de lo parámetro P0313

Opciones	Descripción
0 = Inactivo	Ninguna acción es tomada, el convertidor de frecuencia permanece en el estado actual.
1 = Para por Rampa	El comando de parada por rampa es ejecutado, y el motor para de acuerdo con la rampa de desaceleración programada.
2 = Deshabilita General	El convertidor de frecuencia es deshabilitado general, y el motor para por inercia.
3 = Va para modo Local	El convertidor de frecuencia es comandado para el modo local.
4 = Va para modo Local y mantiene comandos y referencia	El convertidor es comandado para el modo local, más los comandos de habilita y de referencia de velocidad recibidos vía red son mantenidos en modo local, desde que el convertidor de frecuencia sea programado para utilizar, en modo local, comandos y la referencia de velocidad vía HMI.
5 = Causa Falla	En el lugar de alarma, un error de comunicación causa un falla en el convertidor de frecuencia; siendo necesario hacer el reset de fallas en el convertidor de frecuencia para que el mismo regrese a su operación normal.

Se considera errores de comunicación los siguientes eventos:

Comunicación Serial (RS485):

- Alarma A128/Falla F228: *timeout* de la interfaz serial.

Las acciones descritas en este parámetro son ejecutadas a través de la escrita automática de los respectivos bits en el parámetro de control de la interfaz de red que corresponde a la falla detectada. De esta forma, para que los comandos escritos en este parámetro tengan efecto, es necesario que el equipo se encuentre programado para ser controlado vía la interfaz de red utilizada (a excepción de la opción "Causa Falla", que bloquea el convertidor aunque el mismo no sea controlado vía red). Esta programación es hecha a través de los parámetros P0220 hasta P0228.

P0314 – WATCHDOG SERIAL

Rango de Valores:	0,0 a 999,0s	Padrón: 0,0
Propiedades:	CFG	
Grupo de acceso vía HMI:	NET	

Descripción:

Permite programar un tiempo para la detección de error de comunicación vía interfaz serial. Caso el convertidor de frecuencia se queda sin recibir telegramas válidos por un tiempo mayor del que el programado en este parámetro, será considerado que ha ocurrido un error de comunicación, señalizando el alarma A128 en la HMI (o falla F228, dependiendo de la programación hecha en el P0313) y la acción programada en el P0313 será ejecutada.

Luego de energizado, el convertidor de frecuencia empezará a contar este tiempo a partir del primero telegrama válido recibido. El valor 0,0 deshabilita esta función.

P0316 – ESTADO DE LA INTERFAZ SERIAL

Rango de Valores:	0 = Inactivo 1 = Activo 2 = Error de Watchdog	Padrón: -
Propiedades:	RO	
Grupo de acceso vía HMI:	NET	

Descripción:

Permite identificar si la tarjeta de interfaz serial RS485 está debidamente instalado, y si la comunicación serial presenta errores.

Tabla 3.2: Valores de lo parámetro P0316

Opciones	Descripción
0 = Inactivo	Interfaz serial inactiva. Ocurre cuando el equipo no posee tarjeta de interfaz RS485 instalado. No utilizado en CFW701.
1 = Activo	Tarjeta de interfaz RS485 instalada y reconocida.
2 = Error de Watchdog	Interfaz serial activa, más detectado error de comunicación serial – alarma A128 / falla F228.

P0680 – ESTADO LÓGICO
Rango de 0000h a FFFFh

Padrón: -

Valores:
Propiedades: RO

Grupo de acceso vía HMI: NET

Descripción:

Permite el monitoreo del estado del drive. Cada bit representa un estado:

Bits	15	14	13	12	11	10	9	8	7	6	5	4 a 0
Función	En Falla	Reservado	Subtensión	LOC/REM	JOG	Sentido de Giro	Habilitado General	Motor Girando	En Alarma	En modo de configuración	Segunda Rampa	Reservado

Tabla 3.3: Función de los bits para el parámetro P0680

Bits	Valores
Bits 0 a 4	Reservado.
Bit 5 Segunda Rampa	0: Drive configurado para rampa de aceleración y de desaceleración del motor vía primera rampa; valores programados en los parámetros P0100 y P0101. 1: Drive configurado para rampa de aceleración y de desaceleración del motor vía segunda rampa; valores programados en los parámetros P0102 y P0103.
Bit 6 En Modo de Configuración	0: Drive operando normalmente. 1: Drive en modo de configuración. Indica una condición especial en la cual el convertidor de frecuencia no puede ser habilitado: <ul style="list-style-type: none"> ▪ Ejecutando la rutina de autoajuste. ▪ Ejecutando la rutina de puesta en marcha (start-up) orientada. ▪ Ejecutando la función copy de la HMI. ▪ Ejecutando la rutina auto-guiada de la tarjeta de memoria flash. ▪ Posee incompatibilidad de parametrización. ▪ Sin alimentación en el circuito de potencia del drive.
Bit 7 En Alarma	0: Drive no está en el estado de alarma. 1: Drive está en el estado de alarma. Observación: el número de la alarma puede ser leído a través del parámetro P0048 – Alarma Actual.
Bit 8 Motor Girando	0: Motor está parado. 1: Drive está girando el eje del motor a la velocidad de referencia, o ejecutando rampa de aceleración o desaceleración.
Bit 9 Habilitado General	0: Drive está deshabilitado general. 1: Drive está habilitado general y listo para girar el eje del motor.
Bit 10 Sentido de Giro	0: Motor girando en el sentido reverso. 1: Motor girando en el sentido directo.
Bit 11 JOG	0: Función JOG inactiva. 1: Función JOG activa.
Bit 12 LOC/REM	0: Drive en modo local. 1: Drive en modo remoto.
Bit 13 Subtensión	0: Sin subtensión. 1: Con subtensión.
Bit 14	Reservado
Bit 15 En Falla	0: Drive no está en el estado de falla. 1: Algún falla registrado por el drive. Observación: El número del falla puede ser leído a través del parámetro P0049 – Falla Actual.

P0681 – VELOCIDAD DEL MOTOR EN 13 BITS

Rango de Valores:	- 32768 a 32767	Padrón: -
Propiedades:	RO	
Grupo de acceso vía HMI:	NET	

Descripción:

Permite monitorear la velocidad del motor. Esta palabra utiliza resolución de 13 bits con señal para representar la rotación sincrónica del motor:

P0681 = 0000h (0 decimal) velocidad del motor = 0

P0681 = 2000h (8192 decimal) velocidad del motor = rotación sincrónica

Valores de velocidad intermedios o superiores pueden ser obtenidos utilizando esta escala. Por ejemplo, para un motor de 4 polos y 1800 rpm de rotación sincrónica, caso el valor leído sea 2048 (0800h), para obtener el valor en rpm se debe calcular:

8192 => 1800 rpm
2048 => Velocidad en rpm

Velocidad en rpm = $\frac{1800 \times 2048}{8192}$
--

Velocidad en rpm = 450 rpm

Valores negativos para este parámetro indican motor girando en el sentido antihorario.

P0682 – PALABRA DE CONTROL VÍA SERIAL

Rango de Valores:	0000h a FFFFh	Padrón: 0000h
Propiedades:	-	
Grupo de acceso vía HMI:	NET	

Descripción:

Palabra de comando del drive vía interfaz Modbus RTU. Este parámetro solamente puede ser modificado vía interfaz serial. Para las demás fuentes (HMI, etc.) ele se comporta como un parámetro solamente de lectura.

Para que los comandos escritos en este parámetro sean ejecutados, es necesario que el convertidor de frecuencia se encuentre programado para ser controlado vía serial. Esta programación es hecha a través de los parámetros P0105 y P0220 hasta P0228.

Cada bit de esta palabra representa un comando que puede ser ejecutado en el convertidor.

Bits	15	14	13	12 a 8	7	6	5	4	3	2	1	0
Función	Controlador PID externo 2	Controlador PID externo 1	Controlador PID principal	Reservado	Reset de Fallas	Parada Rápida	Utiliza Segunda Rampa	LOC/REM	JOG	Sentido de Giro	Habilita General	Gira/Para

Tabla 3.4: Función de los bits para el parámetro P0682

Bits	Valores
Bit 0 Gira/Para	0: Para el eje del motor por rampa de desaceleración. 1: Gira el eje del motor de acuerdo con la rampa de aceleración hasta alcanzar el valor de la referencia de velocidad.
Bit 1 Habilita General	0: Deshabilita general el drive de frecuencia, interrumpiendo la alimentación para el motor. 1: Habilita general el drive, permitiendo la operación del motor.
Bit 2 Sentido de Giro	0: Girar el eje del motor en el sentido opuesto al de la referencia. 1: Girar el eje del motor en el sentido indicado en la referencia.
Bit 3 JOG	0: Deshabilita la función JOG. 1: Habilita la función JOG.
Bit 4 LOC/REM	0: Drive va para el modo local. 1: Drive va para el modo remoto.
Bit 5 Utiliza Segunda Rampa	0: Drive configurado para rampa de aceleración y de desaceleración del motor vía primera rampa; valores programados en los parámetros P0100 y P0101. 1: Drive configurado para rampa de aceleración y de desaceleración del motor vía segunda rampa; valores programados en los parámetros P0102 y P0103.
Bit 6 Parada Rápida	0: No ejecuta el comando de parada rápida. 1: Ejecuta el comando de parada rápida. Observación: cuando el tipo de control (P0202) es V/F o VVW no se recomienda la utilización de esta función.
Bit 7 Reset de Fallas	0: Sin función. 1: Si en estado de falla, ejecuta el reset del drive.
Bits 8 a 12	Reservado.
Bit 13 Controlador PID principal	0: Automático. 1: Manual
Bit 14 Controlador PID externo 1	0: Automático. 1: Manual
Bit 15] Controlador PID externo 2	0: Automático. 1: Manual

P0683 – REFERENCIA DE VELOCIDAD VÍA SERIAL

Rango de	-32768 a 32767	Padrón: 0
Valores:		
Propiedades:	-	
Grupo de acceso vía HMI:	NET	

Descripción:

Permite programar la referencia de velocidad para el motor vía interfaz Modbus RTU. Este parámetro solamente puede ser modificado vía interfaz serial. Para las demás fuentes (HMI, etc.) elle se comporta como un parámetro solamente de lectura.

Para que la referencia escrita en este parámetro sea utilizada, es necesario que el equipo se encuentre programado para utilizar la referencia de velocidad vía serial. Esta programación es hecha a través de los parámetros P0221 y P0222.

Esta palabra utiliza resolución de 13 bits con señal para representar la rotación sincrónica del motor:

- P0683 = 0000h (0 decimal)→ velocidad del motor = 0
- P0683 = 2000h (8192 decimal)→ velocidad del motor = rotación sincrónica

Valores de referencias intermediarias o superiores pueden ser programados utilizando esta escala. Por ejemplo, para un motor de 4 polos y 1800 rpm de rotación sincrónica, caso se dese una referencia de 900 rpm, se debe calcular:

$$1800 \text{ rpm} \Rightarrow 8192$$

$$900 \text{ rpm} \Rightarrow \text{Referencia en 13 bits}$$

$$\text{Referencia en 13 bits} = \frac{900 \times 8192}{1800}$$

$$\text{Referencia en 13 bits} = 4096 \Rightarrow \text{Valor correspondiente a 900 rpm en la escala de 13 bits}$$

Este parámetro también acepta valores negativos para cambiar el sentido de la rotación del motor. El sentido de la rotación de la referencia, sin embargo, depende también del valor del bit 2 de la palabra de control – P0682:

- Bit 2 = 1 y P0683 > 0: referencia para el sentido directo
- Bit 2 = 1 y P0683 < 0: referencia para el sentido reverso
- Bit 2 = 0 y P0683 > 0: referencia para el sentido reverso
- Bit 2 = 0 y P0683 < 0: referencia para el sentido directo

P0695 – VALOR PARA LAS SALIDAS DIGITALES

Rango de Valores:	0000h a 001Fh	Padrón: 0000h
Propiedades:	RW	
Grupo de acceso vía HMI:	NET	

Descripción:

Posibilita el control de las salidas digitales a través de la interfaz de red. Este parámetro no puede ser modificado a través de la HMI.

Cada bit de este parámetro corresponde al valor deseado para una salida digital. Para que la salida digital correspondiente pueda ser controlada de acuerdo con este contenido, es necesaria que su función sea programada para “Contenido P0695”, en los parámetros P0275 a P0279.

P0696 – VALOR 1 PARA SALIDAS ANALÓGICAS

P0697 – VALOR 2 PARA SALIDAS ANALÓGICAS

Rango de Valores:	-32768 a 32767	Padrón: 0
Propiedades:	RW	
Grupo de acceso vía HMI:	NET	

Descripción:

Posibilita el control de las salidas analógicas a través del interfaz de red . Estos parámetros no pueden ser modificados a través de la HMI.

El valor escrito en estos parámetros es utilizado como valor para la salida analógica, desde que la función de la salida analógica deseada sea programada para “Contenido P0696 / P0697”, en los parámetros P0251, P0254.

El valor debe ser escrito en una escala de 15 bits (7FFFh = 32767)² para representar 100% del valor deseado para la salida, o sea:

- P0696 = 0000h (0 decimal)→ valor para la salida analógica = 0 %
- P0696 = 7FFFh (32767 decimal)→ valor para la salida analógica = 100 %

En este ejemplo fue presentado el parámetro P0696, más la misma escala es utilizada para los parámetros P0697. Por ejemplo, se desea controlar el valor de la salida analógica 1 a través del serial. En este caso se debe proceder la siguiente programación:

- Elegir un de los parámetros P0696, P0697 para ser el valor utilizado por la salida analógica 1. En este ejemplo, vamos elegir el P0696.
- Programar, en la función de la salida analógica 1 (P0254), la opción “Contenido P0696”.
- A través del interfaz de red, escribir en el P0696 el valor deseado para la salida analógica 1, entre 0 y 100%, de acuerdo con la escala del parámetro.

² Para la resolución real de la salida, consulte el manual del producto.

**¡NOTA!**

Caso la salida analógica sea programada para operar de -10V hasta 10V valores negativos para estos parámetros deben ser utilizados para comandar la salida con valores negativos de tensión; o sea, -32768 hasta 32767 que representa una variación de -10V hasta 10V en la salida analógica.

4 PROTOCOLO MODBUS RTU

El protocolo Modbus fue inicialmente desarrollado en 1979. Actualmente, es un protocolo abierto ampliamente difundido, utilizado por varios fabricantes en diversos equipamientos. La comunicación Modbus RTU del convertidor de frecuencia CFW701 fue desarrollada con base en los siguientes documentos:

- MODBUS Protocol Reference Guide Rev. J, MODICON, June 1996.
- MODBUS Application Protocol Specification, MODBUS.ORG, December 28th 2006.
- MODBUS over Serial Line, MODBUS.ORG, December 20th 2006.

En estos documentos están definidos los formatos de los mensajes utilizados por los elementos que hacen parte de la red Modbus, los servicios (o funciones) que pueden ser estar disponibles vía red, y también como estos elementos intercambian datos en la red.

4.1 MODOS DE TRANSMISIÓN

En la especificación del protocolo están definidos dos modos de transmisión: ASCII y RTU. Los modos definen la forma como son transmitidos los bytes del mensaje. No es posible utilizar los dos modos de transmisión en la misma red.

El convertidor de frecuencia CFW701 utiliza solamente el modo RTU para la transmisión de telegramas. Los bytes son transmitidos en el formato hexadecimal, y su configuración depende de la programación hecha a través del .

4.2 ESTRUCTURA DE LOS MENSAJES EN EL MODO RTU

La red Modbus RTU utiliza el sistema maestro-esclavo para el intercambio de mensajes. Permite hasta 247 esclavos, más solamente un maestro. Toda comunicación inicia con el maestro haciendo una solicitud a un esclavo, y este contesta al maestro el que fue solicitado. En ambos los telegramas (pregunta y respuesta), la estructura utilizada es la misma: Dirección, Código de la Función, Datos y CRC. Luego el campo de datos podrá tener tamaño variable, dependiendo del que está siendo solicitado.

Maestro (telegrama de solicitud):

Dirección (1 byte)	Función (1 byte)	Datos de la solicitud (n bytes)	CRC (2 bytes)
-----------------------	---------------------	------------------------------------	------------------

Esclavo (telegrama de respuesta):

Dirección (1 byte)	Función (1 byte)	Datos de la respuesta (n bytes)	CRC (2 bytes)
-----------------------	---------------------	------------------------------------	------------------

4.2.1 Dirección

El maestro inicia la comunicación enviando un byte con la dirección del esclavo para el cual se destina el mensaje. Al enviar la respuesta, el esclavo también inicia el telegrama con el su propia dirección. El maestro también puede enviar un mensaje destinado a la dirección "0" (cero), lo que significa que el mensaje es destinada a todos los esclavos de la red (broadcast). En este caso, ninguno esclavo irá contestar al maestro.

4.2.2 Código de la Función

Este campo también contiene un único byte, donde el maestro especifica el tipo de servicio o función solicitada al esclavo (lectura, escrita, etc.). De acuerdo con el protocolo, cada función es utilizada para acceder un tipo específico de dato.

Para la lista de funciones disponibles para acceso a los datos, consulte el ítem 5.

4.2.3 Campo de Datos

Campo con tamaño variable. El formato y el contenido de este campo dependen de la función utilizada y de los valores transmitidos. Este campo está descrito juntamente con la descripción de las funciones (consultar ítem 5).

4.2.4 CRC

La última parte del telegrama es el campo para el chequeo de errores de transmisión. El método utilizado es el CRC-16 (Cycling Redundancy Check). Este campo es formado por dos bytes, donde primero es transmitido el byte menos significativo (CRC-), y después el más significativo (CRC+). La forma de cálculo del CRC es descrita en la especificación del protocolo, sin embargo informaciones para su implementación también son suministradas en el Apéndice B.

4.2.5 Tiempo entre Mensajes

En el modo RTU no existe un carácter específico que indique el inicio o el fin de un telegrama. La indicación de cuando un nuevo mensaje empieza o cuando él termina es hecha por la ausencia de transmisión de datos en la red, por un tiempo mínimo de 3,5 veces el tiempo de transmisión de un byte de datos (11 bits). Siendo así, caso un telegrama tenga iniciado luego de transcurrido este tiempo mínimo, los elementos de la red irán asumir que el primero carácter recibido representa el inicio de un nuevo telegrama. Y de la misma forma, los elementos de la red irán asumir que el telegrama ha llegado al fin cuando, recibidos los bytes del telegrama, este tiempo transcurrir nuevamente.

Si durante la transmisión de un telegrama, el tiempo entre los bytes fue mayor que este tiempo mínimo, el telegrama será considerado inválido, pues el convertidor de frecuencia irá rechazar los bytes ya recibidos y montará un nuevo telegrama con los bytes que estuvieren siendo transmitidos.

Para tasas de comunicación superiores a 19200 bits/s, los tiempos utilizados son los mismos que para esta tasa. La tabla a seguir preséntanos los tiempos para distintas tasas de comunicación:

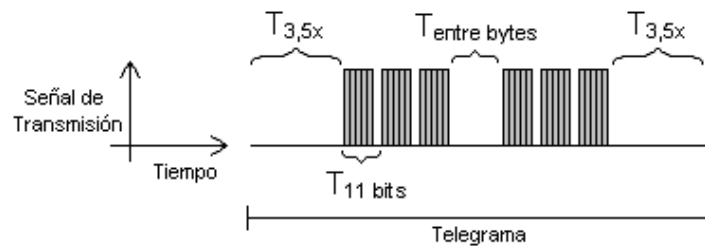


Tabela 4.1: Tasas de comunicación y tiempos involucrados en la transmisión de telegramas

Tasa de Comunicación	T11 bits	T3,5x
1200 bits/s	9,167 ms	32,083 ms
2400 bits/s	4,583 ms	16,042 ms
4800 bits/s	2,292 ms	8,021 ms
9600 bits/s	1,146 ms	4,010 ms
19200 bits/s	573 μ s	2,005 ms
38400 bits/s	573 μ s	2,005 ms
57600 bits/s	573 μ s	2,005 ms

- $T_{11 \text{ bits}}$ = Tiempo para transmitir una palabra del telegrama.
- $T_{\text{entre bytes}}$ = Tiempo entre bytes.
- $T_{3,5x}$ = Intervalo mínimo para indicar el inicio y el fin de telegrama ($3,5 \times T_{11 \text{ bits}}$).

5 OPERACIÓN EN LA RED MODBUS RTU – MODO ESCLAVO

El convertidor de frecuencia CFW701 posee las siguientes características cuando operado como esclavo en red Modbus RTU:

- Conexión de la red vía interfaz serial RS485.
- La dirección, tasa de comunicación y formato de los bytes definidos a través de parámetros.
- Permite la parametrización y control del convertidor de frecuencia a través del acceso a parámetros.

5.1 FUNCIONES DISPONIBLES Y TIEMPOS DE RESPUESTA

En la especificación del protocolo Modbus RTU son definidas funciones utilizadas para acceder diferentes tipos de datos. En el CFW701, los parámetros fueran definidos como siendo registradores del tipo holding. Para acceder estos registradores, fueran colocados disponibles los siguientes servicios (o funciones):

- Read Coils³
Descripción: lectura de bloque bits del tipo coil.
Código de la función: 01.
- Read Discrete Inputs¹
Descripción: lectura de bloque bits del tipo entradas discretas.
Código de la función: 02.
- Read Holding Registers
Descripción: lectura de bloque de registradores del tipo holding.
Código de la función: 03.
- Read Input Registers¹
Descripción: lectura de bloque de registradores del tipo input.
Código de la función: 04.
- Write Single Coil¹
Descripción: escrita en un único bit del tipo coil.
Código de la función: 05.
- Write Single Register
Descripción: escrita en un único registrador del tipo holding.
Código de la función: 06.
- Write Multiple Coils¹
Descripción: escrita en bloque de bit del tipo coil.
Código de la función: 15.
- Write Multiple Registers
Descripción: escrita en bloque de registradores del tipo holding.
Código de la función: 16.
- Read Device Identification
Descripción: identificación del modelo del equipo.
Código de la función: 43.

El tiempo de respuesta, fin de la transmisión del maestro hasta el inicio de la respuesta del esclavo, varía de 2 a 10 ms, para cualquier una de las funciones arriba.

5.2 MAPA DE MEMORIA

La comunicación Modbus para el convertidor de frecuencia CFW701 es basada en la lectura/escritura de parámetros del equipamiento. Toda la lista de parámetros del equipamiento es disponibilizada como registradores de 16 bits del tipo holding. El direccionamiento de los datos es realizado con offset igual a cero, lo

³ Funciones utilizadas para acceder a los datos utilizados por la función SoftPLC.

que significa que el número del parámetro equivale al número del registrador. La tabla a seguir ilustra el direccionamiento de los parámetros, que pueden accesarse como registradores del tipo holding.

Tabla 5.1: Mapa de memoria para la interfaz Modbus RTU

Número del Parámetro	Dirección del dato Modbus	
	Decimal	Hexadecimal
P0000	0	0000h
P0001	1	0001h
⋮	⋮	⋮
P0100	100	0064h
⋮	⋮	⋮

Para la operación del equipamiento, es necesario conocer la lista de parámetros del producto. De esta forma se pueden identificar cuales datos son necesarios para monitoreo de los estados y control de las funciones. Dentro de los principales parámetros se pueden citar:

Monitoreo (lectura):

- P0680 (holding register 680): Palabra de estado
- P0681 (holding register 681): Velocidad del motor

Comando (escritura):

- P0682 (holding register 682): Palabra de comando
- P0683 (holding register 683): Referencia de velocidad

Consulte el manual de programación para la lista completa de parámetros del equipamiento.



NOTA!

- Todos los parámetros son tratados como registradores del tipo holding. Dependiendo del maestro utilizado, estos registradores son referenciados a partir del endereço base 40000 o 4x. En este caso, la dirección para un parámetro que debe ser programado en el maestro es la dirección presentada en la tabla arriba adicionado a la dirección base. Consulte la documentación del maestro para saber como acceder registradores del tipo holding.
- Se debe observar que parámetros con la propiedad de solamente lectura apenas pueden ser leídos del equipamiento, mientras que demás parámetros pueden leerse y escribirse a través de la red.
- Además de los parámetros, otros tipos de datos como marcadores de bit, word o float también pueden ser accedidos utilizando la interfaz Modbus RTU. Estos marcadores son utilizados principalmente por la función SoftPLC disponible para el CFW701. Para la descripción de estos marcadores, bien como la dirección para accederlos vía Modbus, se debe consultar el Manual de la SoftPLC.

6 DESCRIPCIÓN DETALLADA DE LAS FUNCIONES

En este ítem es hecha una descripción detallada de las funciones disponibles en el convertidor de frecuencia CFW701 para comunicación Modbus RTU. Para la elaboración de los telegramas, es importante observar lo siguiente:

- Los valores son siempre transmitidos en hexadecimal.
- La dirección de un dato, el número de datos y el valor de los registradores son siempre representados en 16 bits. Por eso, es necesario transmitir estos campos utilizando dos bytes – superior (high) e inferior (low).
- Los telegramas, tanto para pregunta cuanto para respuesta, no pueden ultrapasar 64 bytes.

6.1 FUNCIÓN 03 – READ HOLDING REGISTER

Lee el contenido de un grupo de registradores, que necesariamente deben estar en secuencia numérica. Esta función posee la siguiente estructura para los telegramas de lectura y respuesta (cada campo representa un byte):

Solicitud (Maestro)	Respuesta (Esclavo)
Dirección del esclavo	Dirección del esclavo
Función	Función
Dirección del registrador inicial (byte high)	Campo Byte Count
Dirección del registrador inicial (byte low)	Dato 1 (high)
Número de registradores (byte high)	Dato 1 (low)
Número de registradores (byte low)	Dato 2 (high)
CRC-	Dato 2 (low)
CRC+	Etc...
	CRC-
	CRC+

6.2 FUNCIÓN 06 – WRITE SINGLE REGISTER

Esta función es utilizada para escribir un valor para un único registrador. Posee la siguiente estructura (cada campo representa un byte):

Solicitud (Maestro)	Respuesta (Esclavo)
Dirección del esclavo	Dirección del esclavo
Función	Función
Dirección del registrador (byte high)	Dirección del registrador (byte high)
Dirección del registrador (byte low)	Dirección del registrador (byte low)
Valor para el registrador (byte high)	Valor para el registrador (byte high)
Valor para el registrador (byte low)	Valor para el registrador (byte low)
CRC-	CRC-
CRC+	CRC+

Note que para esta función, la respuesta del esclavo es una copia idéntica de la solicitud hecha por el maestro.

6.3 FUNCIÓN 16 – WRITE MULTIPLE REGISTERS

Esta función permite escribir valores para un grupo de registradores, que deben estar en secuencia numérica. También puede ser usada para escribir un único registrador (cada campo representa un byte).

Solicitud (Maestro)	Respuesta (Esclavo)
Dirección del esclavo	Dirección del esclavo
Función	Función
Dirección del registrador inicial (byte high)	Dirección del registrador inicial (byte high)
Dirección del registrador inicial (byte low)	Dirección del registrador inicial (byte low)
Número de registradores (byte high)	Número de registradores (byte high)
Número de registradores (byte low)	Número de registradores (byte low)
Campo Byte Count (nº de bytes de datos)	CRC-
Dato 1 (high)	CRC+
Dato 1 (low)	
Dato 2 (high)	
Dato 2 (low)	
etc...	
CRC-	
CRC+	

6.4 FUNCIÓN 43 – READ DEVICE IDENTIFICATION

Función auxiliar, que permite la lectura del fabricante, modelo y versión de firmware del producto. Posee la siguiente estructura:

Solicitud (Maestro)	Respuesta (Esclavo)
Dirección del esclavo	Dirección del esclavo
Función	Función
MEI Type	MEI Type
Código de lectura	Conformity Level
Número del objeto	More Follows
CRC-	Próximo objeto
CRC+	Número de objetos
	Código del primer objeto
	Tamaño del primer objeto
	Valor del primer objeto (n bytes)
	Código del segundo objeto
	Tamaño del segundo objeto
	Valor del segundo objeto (n bytes)
	etc...
	CRC-
	CRC+

Esta función permite la lectura de tres categorías de informaciones: Básica, Regular y Extendida, y cada categoría es formada por un grupo de objetos. Cada objeto es formado por una secuencia de caracteres ASCII. Para el CFW701, solo informaciones básicas están disponibles, formadas por tres objetos:

- Objeto 00h – VendorName: representa el nombre del fabricante del producto.
- Objeto 01h – ProductCode: Formado por el código del producto (CFW701).
- Objeto 02h – MajorMinorRevision: indica la versión de firmware del producto, en el formato 'VX.XX'.

El código de lectura indica cuales las categorías de informaciones son leídas, y si los objetos son accedidos en secuencia o individualmente. En el caso, el CFW701 soporta los códigos 01 (informaciones básicas en secuencia), y 04 (acceso individual a los objetos). Los demás campos son especificados por el protocolo y para el CFW701 poseen valores fijos.

Ejemplo 4: lectura de las informaciones básicas en secuencia, a partir del objeto 02h, de un esclavo en la dirección 1:

Solicitud (Maestro)		Respuesta (Esclavo)	
Campo	Valor	Campo	Valor
Dirección del esclavo	01h	Dirección del esclavo	01h
Función	2Bh	Función	2Bh
MEI Type	0Eh	MEI Type	0Eh
Código de lectura	01h	Código de lectura	01h
Número del objeto	02h	Conformity Level	81h
CRC-	70h	More Follows	00h
CRC+	77h	Próximo Objeto	00h
		Número de objetos	01h
		Código del objeto	02h
		Tamaño del objeto	05h
		Valor del objeto	'V1.00'
		CRC-	3Ch
		CRC+	53h

En este ejemplo, el valor de los objetos no fue representado en hexadecimal, más sí utilizando los caracteres ASCII correspondientes. Por ejemplo, para el objeto 02h, el valor 'V1.00' fue transmitido como siendo cinco caracteres ASCII, que en hexadecimal poseen los valores 56h ('V'), 31h ('1'), 2Eh ('.'), 30h ('0') y 30h ('0').

6.5 ERRORES DE COMUNICACIÓN

Errores de comunicación pueden ocurrir tanto en la transmisión de los telegramas cuanto en el contenido de los telegramas transmitidos. De acuerdo con el tipo de error, el CFW701 podrá o no enviar respuesta para el maestro.

Cuando el maestro envía un mensaje para un esclavo configurado en una determinada dirección de la red, este no irá contestar al maestro caso ocurra:

- Error en el bit de paridad.
- Error en el CRC.
- Timeout entre los bytes transmitidos (3,5 veces el tiempo de transmisión de un byte).

En estos casos, el maestro deberá detectar la ocurrencia del error por el timeout en la espera de la contestación del esclavo. En el caso de una recepción con suceso, durante el tratamiento del telegrama, el esclavo puede detectar problemas y enviar un mensaje de error, indicando el tipo de problema encontrado:

- Función inválida (código del error = 1): la función solicitada no está implementada para el equipamiento.
- Dirección de dato inválido (código del error = 2): la dirección del dato no existe.
- Valor de dato inválido (código del error = 3): ocurre en las siguientes situaciones:
 - Valor está fuera del rango permitido.
 - Escrita en dato que no puede ser modificado (registrador solamente de lectura).



¡NOTA!

Es importante que sea posible identificar en el maestro cual el tipo de error ocurrido para que se pueda diagnosticar problemas durante la comunicación.

En el caso de la ocurrencia de algún de estos errores, el esclavo debe retornar un mensaje para el maestro que indica el tipo de error ocurrido. Los mensajes de error enviados por el esclavo poseen la siguiente estructura:

Solicitud (Maestro)	Respuesta (Esclavo)
Dirección del esclavo	Dirección del esclavo
Función	Función (con el bit más significativo en 1)
Datos	Código del error
CRC-	CRC-
CRC+	CRC+

Ejemplo 5: maestro solicita para el esclavo de la dirección 1 la escrita en el registrador 2900 (suponiendo registrador 2900 como siendo inexistente):

Solicitud (Maestro)		Respuesta (Esclavo)	
<i>Campo</i>	<i>Valor</i>	<i>Campo</i>	<i>Valor</i>
Dirección del esclavo	01h	Dirección del esclavo	01h
Función	06h	Función	86h
Registrador (high)	0Bh	Código del error	02h
Registrador (low)	54h	CRC-	C3h
Valor (high)	00h	CRC+	A1h
Valor (low)	00h		
CRC-	CAh		
CRC+	3Eh		

7 FALLAS Y ALARMAS RELACIONADAS CON LA COMUNICACIÓN MODBUS RTU

A128/F228 – TIMEOUT EN LA RECEPCIÓN DE TELEGRAMAS

Descripción:

Único alarma/falla relacionado con la comunicación serial. Indica que el producto ha parado de recibir telegramas seriales válidos por un período mayor del que el programado en el P0314.

Actuación:

El parámetro P0314 permite programar un tiempo dentro del cual el esclavo deberá recibir al menos un telegrama válido vía interface serial RS485 – con dirección y campo de chequeo de errores correctos – caso contrario será considerado que ha ocurrido algún problema en la comunicación serial. El conteo del tiempo es iniciado luego de la recepción del primer telegrama válido.

Después de identificado el timeout en la comunicación serial, será señalizada a través del HMI el mensaje de alarma A128 – o falla F228, dependiendo de la programación hecha en el parámetro P0313. En condición de alarma, caso la comunicación sea restablecida y nuevos telegramas válidos sean recibidos, la señalización de falla en el IHM desaparecerá automáticamente.

Posibles Causas/Corrección:

- Verificar factores que puedan provocar fallas en la comunicación (cables, instalación, puesta a tierra).
- Garantizar que el maestro envíe telegramas para el esclavo siempre en un tiempo menor que el programado en el P0314.
- Deshabilitar esta función en el P0314.

I. APÉNDICES

APÉNDICE A. TABLA ASCII

Tabla I.1: Caracteres ASCII

Dec	Hex	Chr	Dec	Hex	Chr	Dec	Hex	Chr	Dec	Hex	Chr
0	00	NUL (Null char.)	32	20	Sp	64	40	@	96	60	`
1	01	SOH (Start of Header)	33	21	!	65	41	A	97	61	a
2	02	STX (Start of Text)	34	22	"	66	42	B	98	62	b
3	03	ETX (End of Text)	35	23	#	67	43	C	99	63	c
4	04	EOT (End of Transmission)	36	24	\$	68	44	D	100	64	d
5	05	ENQ (Enquiry)	37	25	%	69	45	E	101	65	e
6	06	ACK (Acknowledgment)	38	26	&	70	46	F	102	66	f
7	07	BEL (Bell)	39	27	'	71	47	G	103	67	g
8	08	BS (Backspace)	40	28	(72	48	H	104	68	h
9	09	HT (Horizontal Tab)	41	29)	73	49	I	105	69	i
10	0A	LF (Line Feed)	42	2A	*	74	4A	J	106	6A	j
11	0B	VT (Vertical Tab)	43	2B	+	75	4B	K	107	6B	k
12	0C	FF (Form Feed)	44	2C	,	76	4C	L	108	6C	l
13	0D	CR (Carriage Return)	45	2D	-	77	4D	M	109	6D	m
14	0E	SO (Shift Out)	46	2E	.	78	4E	N	110	6E	n
15	0F	SI (Shift In)	47	2F	/	79	4F	O	111	6F	o
16	10	DLE (Data Link Escape)	48	30	0	80	50	P	112	70	p
17	11	DC1 (Device Control 1)	49	31	1	81	51	Q	113	71	q
18	12	DC2 (Device Control 2)	50	32	2	82	52	R	114	72	r
19	13	DC3 (Device Control 3)	51	33	3	83	53	S	115	73	s
20	14	DC4 (Device Control 4)	52	34	4	84	54	T	116	74	t
21	15	NAK (Negative Acknowledgement)	53	35	5	85	55	U	117	75	u
22	16	SYN (Synchronous Idle)	54	36	6	86	56	V	118	76	v
23	17	ETB (End of Trans. Block)	55	37	7	87	57	W	119	77	w
24	18	CAN (Cancel)	56	38	8	88	58	X	120	78	x
25	19	EM (End of Medium)	57	39	9	89	59	Y	121	79	y
26	1A	SUB (Substitute)	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC (Escape)	59	3B	;	91	5B	[123	7B	{
28	1C	FS (File Separator)	60	3C	<	92	5C	\	124	7C	
29	1D	GS (Group Separator)	61	3D	=	93	5D]	125	7D	}
30	1E	RS (Record Separator)	62	3E	>	94	5E	^	126	7E	~
31	1F	US (Unit Separator)	63	3F	?	95	5F	_	127	7F	DEL

APÉNDICE B. CÁLCULO DEL CRC UTILIZANDO TABLAS

A seguir es presentada una función, utilizando lenguaje de programación "C", que implementa el cálculo del CRC para el protocolo Modbus RTU. El cálculo utiliza dos tablas para suministrar valores precalculados de los desplazamientos necesarios para la realización del cálculo. El algoritmo fue obtenido y es explicado en los documentos referenciados en el ítem 4.

```

/* Table of CRC values for high-order byte */
static unsigned char auchCRChi[] = {
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41 };

/* Table of CRC values for low-order byte */
static char auchCRCLo[] = {
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4, 0x04,
0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09, 0x08, 0xC8,
0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0xDE, 0xDF, 0x1F, 0xDD, 0x1D, 0x1C, 0xDC,
0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3, 0x11, 0xD1, 0xD0, 0x10,
0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36, 0xF6, 0xF7, 0x37, 0xF5, 0x35, 0x34, 0xF4,
0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A, 0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38,
0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE, 0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C,
0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26, 0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0xE0, 0x20,
0xA0, 0x60, 0x61, 0xA1, 0x63, 0xA3, 0xA2, 0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4,
0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F, 0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68,
0x78, 0xB8, 0xB9, 0x79, 0xBB, 0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C,
0xB4, 0x74, 0x75, 0xB5, 0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0xB0, 0x70,
0x50, 0x90, 0x91, 0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54,
0x9C, 0x5C, 0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98,
0x88, 0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80, 0x40 };

/* The function returns the CRC as a unsigned short type */
unsigned short CRC16(puchMsg, usDataLen)
unsigned char *puchMsg; /* message to calculate CRC upon */
unsigned short usDataLen; /* quantity of bytes in message */
{
    unsigned char uchCRChi = 0xFF; /* high byte of CRC initialized */
    unsigned char uchCRCLo = 0xFF; /* low byte of CRC initialized */
    unsigned uIndex; /* will index into CRC lookup table */
    while (usDataLen--) /* pass through message buffer */
    {
        uIndex = uchCRCLo ^ *puchMsg++; /* calculate the CRC */
        uchCRCLo = uchCRChi ^ auchCRChi[uIndex];
        uchCRChi = auchCRCLo[uIndex];
    }
    return (uchCRChi << 8 | uchCRCLo);
}

```