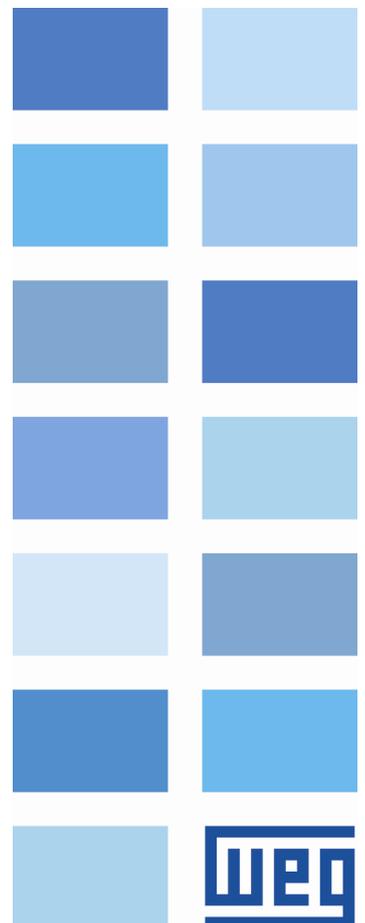


# Comunicación Serial

Soft-Starter SSW-06 V1.6X

## Manual de la Comunicación Serial

Idioma: Español  
Documento: 0899.5732 / 04





## **Manual de la Comunicación Serial**

Serie: SSW-06 V1.6X

Idioma: Español

Nº del Documento: 0899.5732 / 04

Fecha de la Publicación: 08/2009

# ÍNDICE

<b>SOBRE EL MANUAL .....</b>	<b>5</b>
CUIDADO .....	5
AVISO .....	5
TÉRMINOS UTILIZADOS .....	5
REPRESENTACIÓN NUMÉRICA .....	5
<b>1 INTRODUCCIÓN .....</b>	<b>6</b>
<b>2 PARAMETRIZACIÓN DE LA SOFT-STARTER SSW-06.....</b>	<b>7</b>
2.1 P308 – DIRECCIÓN DE LA SOFT-STARTER SSW-06 EN LA RED .....	7
2.2 P312 – TIPO DE PROTOCOLO SERIE Y TASA DE COMUNICACIÓN.....	7
2.3 P313 – ACCIÓN PARA ERROR DE COMUNICACIÓN .....	7
2.4 P314 – TIEMPO PARA TIMEOUT EN LA RECEPCIÓN DE TELEGRAMAS.....	8
2.5 P220 – SELECCIÓN FUENTE LOCAL/REMOTO.....	8
2.6 P229 – SELECCIÓN DE COMANDOS – SITUACIÓN LOCAL.....	9
2.7 P230 – SELECCIÓN DE COMANDOS – SITUACIÓN REMOTO .....	9
<b>3 DESCRIPCIÓN DE LAS INTERFACES .....</b>	<b>10</b>
3.1 RS-232.....	10
3.2 RS-485.....	10
3.2.1 Utilización del Kit RS-485 para SSW-06 .....	10
<b>4 DATOS ACCESIBLES VÍA COMUNICACIÓN SERIE.....</b>	<b>12</b>
4.1 PARÁMETROS DE LA SOFT-STARTER SSW-06.....	12
4.2 VARIABLES BÁSICAS DISPONIBLES PARA LA SOFT-STARTER SSW-06 .....	12
4.2.1 Variable Básica 0 .....	12
4.2.2 Variable Básica 1 .....	12
4.2.3 Variable Básica 3 .....	13
4.2.4 Variable Básica 6 .....	14
4.2.5 Variable Básica 7 .....	14
4.2.6 Variable Básica 8 .....	15
4.2.7 Variable Básica 9 .....	15
4.2.8 Variable Básica 10 .....	15
4.3 ALTERACIÓN DE PARÁMETROS Y VARIABLES BÁSICAS .....	16
4.4 VALORES DE LOS PARÁMETROS .....	16
<b>5 PROTOCOLO MODBUS-RTU .....</b>	<b>17</b>
5.1 MODOS DE TRANSMISIÓN.....	17
5.2 ESTRUCTURA DE LOS MENSAJES EN EL MODO RTU.....	17
5.2.1 Dirección.....	17
5.2.2 Código de la Función.....	17
5.2.3 Campo de Datos .....	18
5.2.4 CRC .....	18
5.2.5 Tiempo entre Mensajes.....	18
5.3 OPERACIÓN DE LA SOFT-STARTER SSW-06 EN LA RED MODBUS-RTU.....	18
5.3.1 Funciones Disponibles y Tiempos de Respuesta.....	19
5.3.2 Direccionamiento de los Datos y Offset.....	19
5.4 DESCRIPCIÓN DETALLADA DE LAS FUNCIONES.....	20
5.4.1 Función 01 – Read Coils.....	21
5.4.2 Función 03 – Read Holding Register.....	21
5.4.3 Función 05 – Write Single Coil .....	22
5.4.4 Función 06 – Write Single Register .....	23
5.4.5 Función 15 – Write Multiple Coils.....	23
5.4.6 Función 16 – Write Multiple Registers .....	24
5.4.7 Función 43 – Read Device Identification .....	25

---

5.4.8 Errores de Comunicación .....	26
APENDICES.....	28

# SOBRE EL MANUAL

Este manual describe la operación de la Soft-Starter SSW-06 utilizando la comunicación serie.

## CUIDADO

- Leer el manual de la Soft-Starter SSW-06 en su totalidad, antes de instalar u operar la misma.
- Seguir atentamente los cuidados y avisos de seguridad contenidos en el manual.
- Cuando hay posibilidad de daños a personas o equipamientos relacionados a los motores accionados por la Soft-Starters SSW-06, prever dispositivos de seguridad electromecánicos.

## AVISO

- Seguir atentamente los cuidados definidos en este manual, respecto a los cables de interconexión de las dos interfaces para comunicación serie.
- Equipamiento con componentes sensibles a la electricidad estática. Las tarjetas electrónicas deben ser trabajadas con los siguientes cuidados:
  - No tocar con las manos directamente sobre componentes o conexiones (conectores). Cuando necesario, tocar antes en un objeto metálico que esté puesto a tierra.
  - Utilizar aparato de soldar con puntera aterrizada.

## TÉRMINOS UTILIZADOS

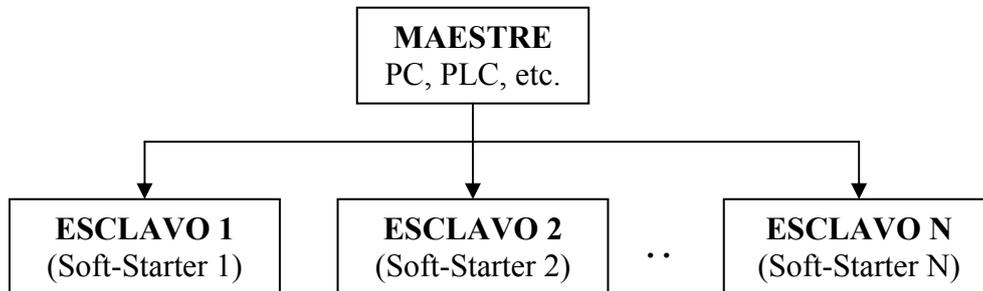
- **Parámetros:** son aquellos existentes en las Soft-Starters SSW-06 cuya visualización o alteración es posible a través de la interface hombre máquina (IHM).
- **Variables básicas:** valores internos de la Soft-Starter SSW-06 que solamente pueden ser acezados a través de la puerta serie, utilizados para monitoreo de los estados, comandos y identificación del equipamiento.
- **Registadores:** son direcciones de memoria interna de la Soft-Starter. Pueden ser utilizados para representar tanto variables básicas cuanto parámetros.
- **EEPROM:** es la memoria no volatiles que permite que la Soft-Starter SSW-06 mantenga los valores de los parámetros mismo después de apagar el equipamiento.

## 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

El objetivo básico de la comunicación serial es la conexión física entre dos o más equipamientos en una red configurada de la siguiente forma:



Utilizando esta interface, el maestro de la red puede solicitar diversos servicios para cada esclavo conectado en la red, tales como:

- IDENTIFICACION:
  - Tipo de equipamiento (convertidor de frecuencia, servoconvertidor, soft-starter)
  - Monitoreo de los estados
  - Lectura de errores
- PARAMETRIZACION
  - Lectura de los parámetros (corriente, tensión, etc.)
  - Escrita de parámetros para configuración del equipamiento
- COMANDOS
  - Habilitación
  - Sentido de giro
  - *Reset* de errores

La Soft-Starter SSW-06 utiliza el protocolo Modbus-RTU para comunicación a través de la puerta serie. Este protocolo permite la integración de la Soft-Starter SSW-06 en diferentes sistemas, una vez que posibilita su conexión a varios equipamientos, tales como:

- PC (maestro) para parametrización de una o varias Soft-Starters SSW-06 simultáneamente.
- SDCD monitoreando variables y parámetros de la Soft-Starter SSW-06.
- CLP controlando la operación del equipamiento en un proceso industrial.

## 2 PARAMETRIZACIÓN DE LA SOFT-STARTER SSW-06

A seguir serán descritos los parámetros relacionados con la comunicación serial y operación vía protocolo Modbus-RTU de la Soft-Starter SSW-06.

### 2.1 P308 – DIRECCIÓN DE LA SOFT-STARTER SSW-06 EN LA RED

Cada esclavo de la red debe de tener una dirección diferente de los demás, para que el maestro pueda enviar el telegrama deseado para un esclavo específico de la red. Este parámetro permite programar cual la dirección de la Soft-Starter SSW-06 en la red.

Rango de valores	Valor padrón	Acceso
1 ... 247	1	Lectura/ escrita

Siendo necesaria la colocación de un repetidor en el caso de que utilícense más que 30 equipamientos en una misma red de comunicación.

### 2.2 P312 – TIPO DE PROTOCOLO SERIE Y TASA DE COMUNICACIÓN

La Soft-Starter SSW-06 tiene una de las siguientes opciones para la comunicación a través de la interface serie del producto:

Rango de valores	Valor patrón	Acceso
1 = Modbus-RTU, 9600 bit/s, sin paridad	1	Lectura/ escrita
2 = Modbus-RTU, 9600 bit/s, paridad impar		
3 = Modbus-RTU, 9600 bit/s, paridad par		
4 = Modbus-RTU, 19200 bit/s, sin paridad		
5 = Modbus-RTU, 19200 bit/s, paridad impar		
6 = Modbus-RTU, 19200 bit/s, paridad par		
7 = Modbus-RTU, 38400 bit/s, sin paridad		
8 = Modbus-RTU, 38400 bit/s, paridad impar		
9 = Modbus-RTU, 38400 bit/s, paridad par		

Es necesario que todos los equipamientos que operan en la misma red tengan la misma configuración de comunicación.

### 2.3 P313 – ACCIÓN PARA ERROR DE COMUNICACIÓN

Este parámetro permite programar una acción que la Soft-starter debe tomar en caso de ocurrir el error de comunicación.

Rango de valores	Patrón	Acceso
0 = Sin acción	0	Lectura/ escrita
1 = Deshabilita		
2 = Deshabilita general		
3 = Cambia para local		
4 = Sin acción		
5 = Causa error fatal		

- **0 – Sin acción:** caso ocurra uno de los errores citados, la soft-starter permanece en el estado actual y solamente indica el error ocurrido.
- **1 – Deshabilita:** la SSW-06 será deshabilitada vía rampa de tensión en caso de error de comunicación.
- **2 – Deshabilita general:** en esta opción la soft-starter corta la energía para el motor, y este debe parar por inercia.
- **3 – Cambia para local:** caso la soft-starter esté operando en el modo remoto y ocurra un error de comunicación, ella debe ir automáticamente para el modo local.
- **4 – Sin acción:** exactamente el comportamiento de la opción 0.
- **5 – Causa error fatal:** al detectar una falla en la comunicación la Soft-Starter SSW-06 irá para el estado del error, el motor será deshabilitado y la indicación del error será quitada solamente después haver el reset de los errores del equipo.

## Parametrización de la Soft-Starter SSW-06

Para la puerta serie, solamente el error de *timeout* en la recepción de telegramas (E28 - Comunicación Serie Inactiva) es considerado como error en la comunicación. El *timeout* en la recepción de telegramas es programado a través del parámetro P314.



### ¡NOTA!

Los comandos de deshabilita y cambio para el modo local solamente pueden ser ejecutados si los mismos estuvieren siendo controlados vía puerta serie. Esta programación es hecha a través de los parámetros P220, P229 y P230.

## 2.4 P314 – TIEMPO PARA TIMEOUT EN LA RECEPCIÓN DE TELEGRAMAS

Permite programar el tiempo para detección de *timeout* en la recepción de telegramas. El valor 0 (cero) deshabilita esta función.

Caso la soft-starter esté siendo controlado vía puerta serie y ocurra un problema en la comunicación con el maestro (cables rotos, caída de tensión, etc.), no será posible enviar un comando vía puerta serie para la deshabilitación del equipamiento. En las aplicaciones donde esto representa un problema, es posible programar en el P314 un intervalo máximo dentro del cual la SSW-06 debe recibir un telegrama serie válido, caso contrario ella considera que hubo falla en la comunicación serie.

Rango de valores	Valor patrón	Acceso
0 = Función deshabilitada	0	Lectura/ escrita
0 ... 999 segundos		

Una vez programado este tiempo, caso él quede un tiempo mayor que el programado sin recibir telegramas seriales válidos, él indicará E28 y tomará la acción programada en P313. Caso la comunicación sea establecida nuevamente, la indicación de E28 será retirada.



### ¡NOTAS!

- Cuando esta función está habilitada, es necesario garantizar que el maestro de la red envíe telegramas periódicos para el esclavo, respetando el tiempo programado, para que no ocurra error de *timeout* en la comunicación.
- Cuando ocurrir el E28 también va a cerrar los valores de las variables básicas 8, 9 y 10 (ver ítem 4.2).

## 2.5 P220 – SELECCIÓN FUENTE LOCAL/REMOTO

Permite programar la fuente de comando que controla los modos local/remoto del equipamiento.

Rango de valores	Valor patrón	Acceso
0 = Siempre local	2	Lectura/ escrita
1 = Siempre remoto		
2 = Teclado (patrón local)		
3 = Teclado (patrón remoto)		
4 = DI4...DI5		
5 = Puerta Serie (patrón local)		
6 = Puerta Serie (patrón remoto)		
7 = Fieldbus (patrón local)		
8 = Fieldbus (patrón remoto)		
9 = SoftPLC (patrón local)		
10 = SoftPLC (patrón remoto)		

Caso desee controlar el modo de operación vía puerta serie, débese programar este parámetro con el valor 5 o 6. La indicación de "patrón local" o "patrón remoto" informa cual el modo de operación que debe ser activado después de la puesta en marcha del equipamiento.

## Parametrización de la Soft-Starter SSW-06

### 2.6 P229 – SELECCIÓN DE COMANDOS – SITUACIÓN LOCAL

Permite programar cual es la fuente de los comandos de la Soft-Starter SSW-06 cuando esta está en el modo local.

Rango de valores	Valor patrón	Acceso
0 = Teclado 1 = DI 2 = Puerta Serie 3 = Fieldbus 4 = SoftPLC	0	Lectura/ escrita

Caso desee controlar los comandos vía puerta serie en el modo local, débese programar este parámetro en 2.

### 2.7 P230 – SELECCIÓN DE COMANDOS – SITUACIÓN REMOTO

Permite programar cual es la fuente de los comandos de la Soft-Starter SSW-06 cuando esta está en el modo remoto.

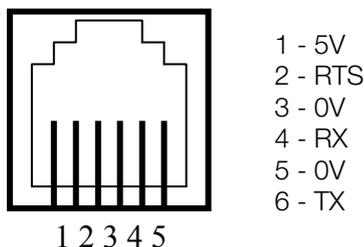
Rango de valores	Valor patrón	Acceso
0 = Teclado 1 = DI 2 = Puerta serie 3 = Fieldbus 4 = SoftPLC	0	Lectura/ escrita

Caso deséese controlar los comandos vía puerta serie en el modo remoto, débese programar este parámetro en 2.

### 3 DESCRIPCIÓN DE LAS INTERFACES

#### 3.1 RS-232

La Soft-Starter SSW-06 tiene una puerta serie RS-232C (conector X2) disponible en el producto.



**Figura 1:** Vista del conector X2 (conector modelo RJ11)

Esta interface posibilita la conexión de un maestro a una Soft-Starter SSW-06 (punto a punto) en una distancia de hasta 10m. Para comunicación con el maestro, débese utilizar un alambre para transmisión (TX), un para recepción (RX) y una referencia (0V), señales estos presentes en los bornes 4, 5 y 6. Los señales presentes en los bornes 1, 2 y 3 salen de este conector para alimentación y control del convertidor externo MIW-02, utilizado como una de las opciones para comunicación RS-485.

#### 3.2 RS-485

Hay dos posibilidades para disponibilizar una interface RS-485 en la Soft-Starter SSW-06:



##### Kit opcional RS-485 para SSW-06

- Ítem WEG: 10927208.
- Conectado internamente en el producto (en el conector XC8 de la tarjeta de control).



##### Módulo opcional MIW-02

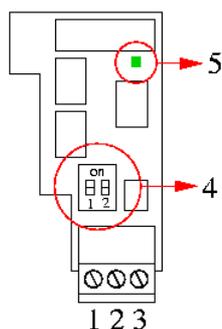
- Ítem WEG: 10051677.
- Convertidor RS-232 para RS-485.
- Módulo externo al producto, conectado en la interface RS-232 de la SSW-06.
- Consulte el manual del MIW-02 para mayores informaciones.

Utilizando la interface RS-485, el maestro puede controlar diversas soft-starters conectadas en una misma barra. El protocolo Modbus-RTU permite la conexión de hasta 247 esclavos (1 por dirección), desde que utilizados también repetidores de señal al largo de la barra. Esta interface tiene una buena inmunidad al ruido, y el largo máximo permitido del cable es de 1000 metros.

##### 3.2.1 Utilización del Kit RS-485 para SSW-06

El kit RS-485 para Soft-Starter SSW-06 es compuesto básicamente por una tarjeta de interface e instrucción para realizar la instalación en el producto. Una vez realizada la instalación en la SSW-06, la tarjeta disponibiliza los siguientes elementos:

## Descripción de las Interfaces

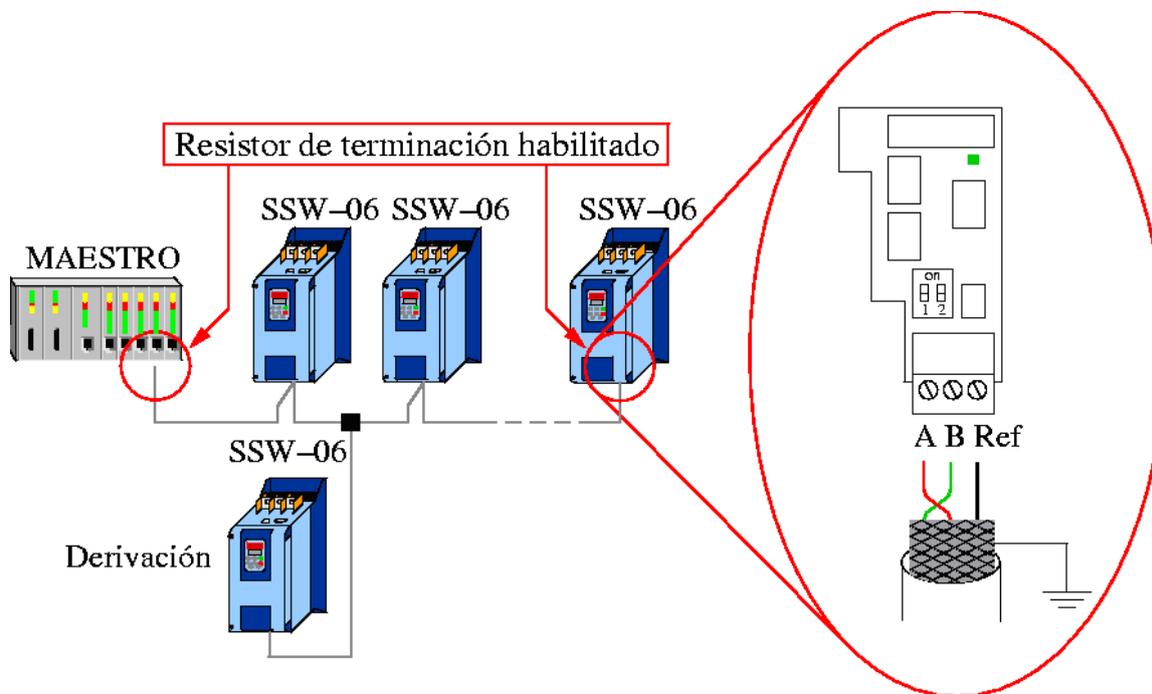


- 1 → Conector para el señal A (o Data +)
- 2 → Conector para el señal B (o Data -)
- 3 → Conector para el señal de referencia
- 4 → Selectores para habilitación del resistor de terminación
- 5 → LED de indicación TX (enciende cuando la soft-starter envía telegrama de respuesta)

Las siguientes recomendaciones deben ser observadas durante la instalación de la red utilizando esta interface:

- Generalmente utilizase un par de alambres tranzados con blindaje para la transmisión de los señales B y A. Estos señales deben ser conectados en los bornes 1 y 2 de la tarjeta.
- El borne 3 es utilizado para la conexión del señal de referencia para el circuito RS-485. Caso este señal no sea utilizado, puedes desconsiderar esta conexión.
- Es muy importante aterrizar correctamente todos los dispositivos conectados en la red RS-485, preferencialmente en el mismo punto de puesta a tierra. El blindaje del cable también debe ser aterrizada, y para esto puedes conectar el blindaje en algún punto de la carcasa de la Soft-Starter SSW-06.
- La pasaje de cable de red debe ser hecha separadamente (y si posible lejos) de los cables para alimentación de potencia.
- Es necesario disponibilizar resistores de terminación en el primero y en el último dispositivo conectado en la barra principal. La tarjeta de interface para RS-485 ya tiene selectores para habilitación de este resistor. Basta colocar ambos los selectores S1 para la posición 'on'.

La figura a seguir presenta un ejemplo de instalación de Soft-Starters SSW-06 en una red utilizando tarjetas de interface RS-485.



**Figura 2:** Soft-Starters SSW-06 en red RS-485

# 4 DATOS ACCESIBLES VÍA COMUNICACIÓN SERIE

Diversos datos pueden ser acezados vía interface serial en la Soft-Starter SSW-06, para posibilitar su parametrización, comando y monitoreo. Básicamente, estos datos pueden ser divididos en dos grupos: parámetros y variables básicas.

## 4.1 PARÁMETROS DE LA SOFT-STARTER SSW-06

Los parámetros son aquellos disponibles a través de la IHM de la Soft-Starter SSW-06. Prácticamente todos los parámetros de la soft-starter pueden ser acezados vía puerta serie, y a través de estos parámetros es posible configurar la forma como el equipamiento va a operar, bien como monitorear informaciones relevantes para la aplicación, como corriente, tensión, errores, etc..

Débase consultar el manual de la Soft-Starter SSW-06 para obtener el listado completo de los parámetros.

## 4.2 VARIABLES BÁSICAS DISPONIBLES PARA LA SOFT-STARTER SSW-06

Las variables básicas son valores internos de la Soft-Starter SSW-06 accesibles solamente a través de la puerta serie del producto. Utilizando estas variables, es posible monitorear los estados de la soft-starter bien como enviar comandos de habilitación, *reset*, etc..

Cada variable básica representa un registrador (16 bits). Para la Soft-Starter SSW-06 fueron disponibilizadas las siguientes variables básicas:

### 4.2.1 Variable Básica 0

- **Variable:** VB00 – indicación del modelo del equipamiento
- **Acceso:** solamente lectura
- **Descripción:** provee un valor fijo utilizado para identificar el tipo de equipamiento en la red. Para la Soft-Starter SSW-06, esta variable básica tiene el valor 15 (000Fh).

### 4.2.2 Variable Básica 1

- **Variable:** VB01 – estado de la Soft-Starter SSW-06
- **Acceso:** solamente lectura
- **Descripción:** indica el estado de la Soft-Starter SSW-06. Cada bit de esta palabra provee una indicación diferente:

## Datos Accesibles vía Comunicación Serie

Bit	Descripción
Bit 0	0 = motor parado. 1 = motor girando (cambia para cero cuando recibe comando de parada).
Bit 1	0 = cuando deshabilita general por cualquier uno de los medios. 1 = cuando está habilitada general por todos los medios.
Bit 2	0 = sin jog. 1 = con jog.
Bit 3	0 = no está acelerando. 1 = durante toda la aceleración.
Bit 4	0 = no está en limitación de corriente. 1 = limitación de corriente.
Bit 5	0 = sin tensión plena en el motor. 1 = con tensión plena en el motor.
Bit 6	0 = sin alarma. 1 = con alarma.
Bit 7	0 = no está decelerando. 1 = durante toda la deceleración.
Bit 8	0 = local. 1 = remoto.
Bit 9	0 = no está en frenado CC. 1 = durante el frenado CC.
Bit 10	0 = no está cambiando sentido de giro. 1 = durante el proceso de cambio del sentido de giro.
Bit 11	0 = horario. 1 = antihorario.
Bit 12	0 = con bypass abierto. 1 = con bypass cerrado.
Bit 13	Reservado
Bit 14	0 = sin alimentación de la potencia. 1 = con alimentación de la potencia en las 3 fases arriba de 15V.
Bit 15	0 = sin error. 1 = con error.

### 4.2.3 Variable Básica 3

- **Variable:** VB00 – indicación del modelo del equipamiento
- **Acceso:** lectura y escrita
- **Descripción:** permite comandar la Soft-Starter SSW-06 vía puerta serial. Tiene 16 bits, que pueden ser divididos en dos bytes para mejor entendimiento del comando:

**Byte más significativo:** funciona como la máscara de los comandos. Cada bit habilita la ejecución de un comando, y el valor efectivo del comando es transmitido en el bit menos significativo correspondiente.

Bit	Descripción
Bit 8	0 = comando inactivo. 1 = comando habilita / deshabilita activo.
Bit 9	0 = comando inactivo 1 = comando habilita general / deshabilita general activo.
Bit 10	0 = comando inactivo. 1 = comando JOG activo.
Bit 11	0 = comando inactivo. 1 = comando sentido de giro activo.
Bit 12	0 = comando inactivo. 1 = comando local / remoto activo.
Bit 13	Reservado
Bit 14	Reservado
Bit 15	0 = comando inactivo. 1 = comando de <i>reset</i> activo.

**Byte menos significativo:** tiene el valor efectivo para cada comando que se desea- ejecutar. Cada bit es responsable por ejecutar un comando, pero el comando solamente será ejecutado si el bit superior correspondiente está en 1. Caso el bit de la máscara no está con valor en 1, el valor recibido en el bit inferior correspondiente es despresado.

## Datos Accesibles vía Comunicación Serie

Bit	Descripción
Bit 0	0 = parar por rampa. 1 = girar por rampa.
Bit 1	0 = deshabilita general 1 = habilita general.
Bit 2	0 = sin JOG. 1 = con JOG.
Bit 3	0 = sentido horario. 1 = sentido antihorario.
Bit 4	0 = local. 1 = remoto.
Bit 5	Reservado
Bit 6	Reservado
Bit 7	0 = sin comando. 0 → 1 = ejecuta <i>reset</i> (caso esté en error).

Siempre que un comando es enviado para la Soft-Starter SSW-06, esta solamente aceptará y ejecutará el comando caso esté programada para recibir comandos vía puerta serial. Esta programación es hecha a través de los siguientes parámetros:

- P220 - Selección de la fuente local / remoto.
- P229 - Selección de los comandos en el modo local.
- P230 – Selección de los comandos en el modo remoto.

Débase programar estos comandos para la opción "Serial" siempre que se desea ejecutar el referido comando vía red. El comando de *reset* puede ser ejecutado vía red mismo sin esta parametrización, pero solamente si la Soft-Starter SSW-06 está en estado de error.



### ¡NOTAS!

- Errores en la tarjeta de comunicación (E28, E29 o E30) no pueden ser "reseteados" de esta forma, ya que dependen de ajustes fuera de los valores enviados vía red para que sean solucionados, y también porque en esta situación la soft-starter no está consiguiendo comunicarse con la red.
- Caso intente realizar algún comando vía red, mas que no pueda ser ejecutado por la SSW-06 (por ejemplo, un comando que no está programado para operar vía puerta serial), este comando no será ejecutado.

### 4.2.4 Variable Básica 6

- **Variable:** VB06 – indicación de los estados de los modos de ajuste
- **Acceso:** solamente lectura
- **Descripción:** permite verificar si la Soft-Starter SSW-06 está operando en algún modo especial de ajuste. Los únicos bits utilizados en esta variable son:

Bit	Descripción
Bit 0	0 = operación normal 1 = en modo de ajuste después de realizado <i>reset</i> para patrón de fábrica
Bit 1	0 = operación normal 1 = en modo de ajuste después de la alteración de P202
Bit 2 ... 15	Reservado

### 4.2.5 Variable Básica 7

- **Variable:** VB07 – comandos para los modos de ajuste
- **Acceso:** lectura y escrita
- **Descripción:** permite salir de los modos especiales de ajuste:

## Datos Accesibles vía Comunicación Serie

Bit	Descripción
Bit 0	0 = sin comando 1 = sale del modo de ajuste después del <i>reset</i> para patrón de fábrica
Bit 1	0 = sin comando 1 = sale del modo de ajuste después de la alteración de P202
Bit 2 ... 15	Reservado



### ¡NOTA!

Solamente es posible ejecutar estos comandos caso la soft-starter esté en el modo de ajuste en cuestión.

### 4.2.6 Variable Básica 8

- **Variable:** VB08 – comandos para las salidas digitales
- **Acceso:** lectura y escrita
- **Descripción:** permite comandar las salidas a relé disponibles en la Soft-Starter SSW-06. Esta palabra tiene 16 bits, donde solamente los tres primeros bits tienen función:

Bit	Descripción
Bit 0	0 = desactiva salida a relé RL1. 1 = activa salida a relé RL1.
Bit 1	0 = desactiva salida a relé RL2. 1 = activa salida a relé RL2.
Bit 2	0 = desactiva salida a relé RL3. 1 = activa salida a relé RL3.
Bit 3 ... 15	Reservado

Para que las salidas digitales puedan ser comandadas vía puerta serial, es necesario programar las funciones para la opción "Serial", en los parámetros P277, P278 y P279. Caso la salida no esté siendo controlada vía puerta serial, el valor recibido en el bit correspondiente es desconsiderado.

Caso ocurra un error de comunicación con el maestro de la red (E28), los valores para las salidas digitales serán cerrados.

### 4.2.7 Variable Básica 9

- **Variable:** VB09 – valor para la salida analógica AO1
- **Acceso:** lectura y escrita
- **Descripción:** permite comandar la salida analógica AO1.

### 4.2.8 Variable Básica 10

- **Variable:** VB10 – valor para la salida analógica AO2
- **Acceso:** lectura y escrita
- **Descripción:** permite comandar la salida analógica AO2.

Los valores recibidos vía puerta serial para las salidas analógicas AO1 y AO2 tienen resolución de 14 bits<sup>1</sup> para representar 100% del valor para la salida, o sea, el valor 0 (cero) representa 0 % del valor para la salida, mientras que el valor 16383 (3FFFh) representa 100 % del valor para la salida.

Para que sea posible controlar las salidas analógicas vía puerta serial, también es necesario programar las funciones para la opción "Serial", en los parámetros P251 y P253. Caso la salida no esté siendo controlada vía puerta serial, el valor recibido en la variable correspondiente es desconsiderado. No olvidar que aún es posible programar un gaño para las salidas, en los parámetros P252 y P254, que serán aplicados sobre el valor recibido.

<sup>1</sup> Esta resolución es utilizada solamente para transmisión de los datos. Para obtener la resolución real utilizada por las salidas analógicas, consulte el manual del equipamiento.

## Datos Accesibles vía Comunicación Serie

Caso ocurra un error de comunicación con el maestro de la red (E28), los valores para las salidas analógicas serán cerrados.

### 4.3 ALTERACIÓN DE PARÁMETROS Y VARIABLES BÁSICAS

Existen algunas particularidades cuanto al acceso de parámetros y variables básicas de la Soft-Starter SSW-06 vía serial:

- No existe clave para acceso vía serial. Es posible alterar parámetros independiente de la clave estar activa o no.
- El valor para el P000 no es salvado en la memoria no volátil del equipamiento (lo mismo ocurre vía IHM).
- Los parámetros P200 y P215 no están accesibles vía serial.
- Al alterar el valor de P202 o hacer un *reset* para el patrón de fábrica con P204 = 5, la soft-starter entra en modos especiales de ajuste. En estos modos, todos los parámetros están accesibles, pero la soft-Starter SSW-06 no podrá ser operada hasta que salga de esta condición. Esto puede ser hecho vía serial a través de la variable básica 7.
- Después de un comando de *reset* de errores o después de salir de un modo especial de operación, será hecho un *reset* completo del equipamiento. En esta situación, la Soft-Starter SSW-06 podrá no enviar respuesta para el maestro, y la comunicación quedará inactiva hasta que las rutinas de inicialización sean finalizadas.
- La escrita de comandos en la variable básicas 3 solamente será acepta si estos comandos están programados para que sean ejecutados vía serial. De la misma forma, la escrita de comandos en la variable básica 7 solamente es permitida caso la soft-starter esté en el modo de ajuste presentado en la variable básica 6.
- Si el comando de gira es enviado en la variable básica 3 durante la actuación del tiempo de P630, el comando no será acepto, pero la soft-starter no responderá con error.

### 4.4 VALORES DE LOS PARÁMETROS

Existen algunas particularidades cuanto a los valores de los parámetros de la Soft-Starter SSW-06 vía serial:

- Los valores de los parámetros son siempre enteros, sin coma.
- En los modelos de 820A y arriba, todos los parámetros relacionados con la indicación de corriente (A) y potencia (kW y kVA) son divididos por diez: P003, P010, P011, P030, P031, P032, P047, P048, P053, P060, P063, P066, P069, P072, P075.

## 5 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 de la Soft-Starter SSW-06 fue desarrollada con base en los siguientes documentos:

- MODBUS Protocol Reference Guide Rev. J, MODICON, June 1996.
- MODBUS Application Protocol Specification, MODBUS.ORG, May 8<sup>th</sup> 2002.
- MODBUS over Serial Line, MODBUS.ORG, December 2<sup>nd</sup> 2002.

En estos documentos está definido el formato de los mensajes utilizadas por los elementos que hacen parte de la red Modbus, los servicios (o funciones) que pueden estar disponibles vía red, y también como estos elementos cambian datos en la red.

### 5.1 MODOS DE TRANSMISIÓN

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

La Soft-Starter SSW-06 utiliza solamente el modo RTU para la transmisión de telegramas. Los bytes serán transmitidos en el formato hexadecimal, donde cada byte transmitido tiene 1 start bit, 8 bits de datos, 1 bit de paridad(opcional) y 1 stop bit (2 stop bits si no es utilizada paridad). La configuración del formato de los bytes es hecha a través del parámetro P312.

Start bit	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Paridad o stop bit	Stop bit
			8 bits de datos							

### 5.2 ESTRUCTURA DE LOS MENSAJES EN EL MODO RTU

La red Modbus-RTU utiliza el sistema maestro-esclavo para el cambio de mensajes. Ella puede tener hasta 247 esclavos, mas solamente un maestro. Toda comunicación empieza con el maestro haciendo una solicitud a un esclavo, y este responde al maestro lo que fue solicitado. En los dos telegramas (pregunta y respuesta), la estructura utilizada es la misma: Dirección, Código de la Función, Datos y CRC. Solamente el campo de datos podrá tener tamaño variable, dependiendo de lo que está siendo solicitado.

Maestro (telegrama de requisición):

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

Esclavo (telegrama de respuesta):

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

#### 5.2.1 Dirección

El maestro empieza la comunicación enviando un byte con la dirección del esclavo para el cual es destinado el mensaje. Al enviar la respuesta, el esclavo también empieza el telegrama con la suya propia dirección. El maestro también puede enviar un mensaje destinado a la dirección 0 (cero), lo que significa que el mensaje es destinado a todos los esclavos de la red (*broadcast*). En este caso, ningún esclavo responderá al maestro.

#### 5.2.2 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 Soft-Starter SSW-06, los datos relativos a los parámetros y variables básicas están disponibles como registradores del tipo *holding* (referenciados a partir de la dirección 40000 o '4x').

**5.2.3 Campo de Datos**

Campo con tamaño variable. El formato y 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 (ver ítem 5.4).

**5.2.4 CRC**

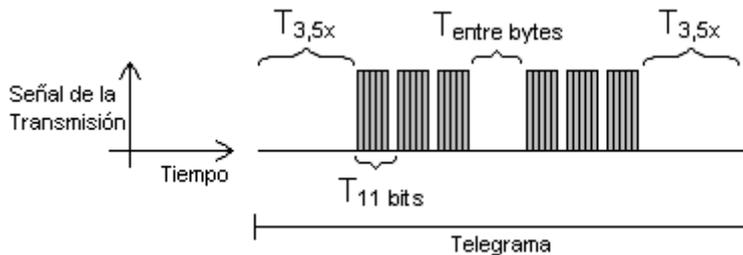
La última parte del telegrama es el campo para el chequeado de errores de transmisión. El método utilizado es el CRC-16 (Cycling Redundancy Check). Este campo es formado por dos bytes, donde primero es transmitido el byte menos significativo (CRC-), y después el más significativo (CRC+). La forma de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

**5.2.5 Tiempo entre Mensajes**

En el modo RTU no existe un carácter específico que indique el principio o el final de un telegrama. De esta forma, lo que indica cuando un nuevo mensaje empieza o cuando termina es 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 empezado después de la ocurrencia de este tiempo mínimo, los elementos de la red van a asumir que el primero carácter recibido representa el principio de un nuevo telegrama. Y de la misma forma, los elementos de la red van a asumir que el telegrama llegó al fin cuando, recibidos los bytes del telegrama, este tiempo transcurra nuevamente.

Si mientras la transmisión de un telegrama, el tiempo entre los bytes es mayor que este tiempo mínimo, el telegrama será considerado inválido, ya que la soft-starter va a descartar los bytes ya recibidos y armará un nuevo telegrama con los bytes que están siendo transmitidos.

La tabla a seguir muestra los tiempos para diferentes tasas de comunicación:



Tasa de Comunicación	T <sub>11 bits</sub>	T <sub>3,5x</sub>
9600 bits/seg	1,146 ms	4,010 ms
19200 bits/seg	573 μs	2,005 ms
38400 bits/seg	573 μs	2,005 ms

- T<sub>11 bits</sub> = Tiempo para transmitir una palabra del telegrama.
- T<sub>entre bytes</sub> = Tiempo entre bytes (no puede ser mayor que T<sub>3,5x</sub>).
- T<sub>3,5x</sub> = Intervalo mínimo para indicar principio y final de telegrama (3,5 x T<sub>11bits</sub>).

Para tasas de comunicación arriba de 19200 bits/s, son considerados los mismos tiempos que el utilizado para 19200 bits/s.

**5.3 OPERACIÓN DE LA SOFT-STARTER SSW-06 EN LA RED MODBUS-RTU**

La Soft-Starter SSW-06 tiene las siguientes características cuando operado en red Modbus-RTU:

## Protocolo Modbus-RTU

- Conexión de la red vía interface serial RS-232 o RS-485 (ver ítem 3).
- Direccionamiento, tasa de comunicación y formato de los bytes definidos a través de parámetros (ver ítem 2).
- Permite la parametrización y control del equipamiento a través del acceso a parámetros y variables básicas.

### 5.3.1 Funciones Disponibles y Tiempos de Respuesta

En la especificación del protocolo Modbus-RTU son definidas funciones utilizadas para acceder diferentes tipos de registradores. En la Soft-Starter SSW-06, tanto parámetros cuanto variables básicas fueron definidos como siendo registradores del tipo *holding*. Además de estos registradores, también es posible acceder directamente bits internos de comando y monitoreo, denominados *coils*. Para acceder estos bits y registradores, fueron disponibilizados los siguientes servicios (o funciones):

- *Read Coils*  
Descripción: Lectura de bloque de bits internos o bobinas.  
Código de la función: 01.  
Tiempo de respuesta: 5 a 10 ms.
- *Read Holding Registers*  
Descripción: Lectura de bloque de registradores del tipo *holding*.  
Código de la función: 03.  
Tiempo de respuesta: 5 a 10 ms.
- *Write Single Coil*  
Descripción: Escrita en un único bit interno o bobina.  
Código de la función: 05.  
Tiempo de respuesta: 5 a 10 ms.
- *Write Single Register*  
Descripción: Escrita en un único registrador del tipo *holding*.  
Código de la función: 06.  
Tiempo de respuesta: 5 a 10 ms.
- *Write Multiple Coils*  
Descripción: Escrita en bloque de bits internos o bobinas.  
Código de la función: 15.  
Tiempo de respuesta: 5 a 10 ms.
- *Write Multiple Registers*  
Descripción: Escrita en bloque de registradores del tipo *holding*.  
Código de la función: 16.  
Tiempo de respuesta: 10 ms para cada registrador escrito.
- *Read Device Identification*  
Descripción: Identificación del modelo de la sofá-starter.  
Código de la función: 43.  
Tiempo de respuesta: 5 a 10 ms.

### 5.3.2 Direccionamiento de los datos y Offset

El direccionamiento de los datos en la Soft-Starter SSW-06 es hecho con offset igual a cero, lo que significa que el número de la dirección es equivalente al número dado. Los parámetros son disponibilizados a partir de la dirección 0 (cero), mientras que las variables básicas son disponibles a partir de la dirección 5000. De la misma forma, los bits de estado son disponibles a partir de la dirección 0 (cero) y los bits de comando son disponibles a partir de la dirección 100. La tabla a seguir ilustra el direccionamiento de parámetros y variables básicas:

PARÁMETROS		
Número del Parámetro	Dirección Modbus	
	Decimal	Hexadecimal
P000	0	0x0000
P001	1	0x0001
⋮	⋮	⋮
P100	100	0x0064
⋮	⋮	⋮

VARIABLES BÁSICAS		
Número de la Variable Básica	Dirección Modbus	
	Decimal	Hexadecimal
V00	5000	0x1388
V01	5001	0x1389
⋮	⋮	⋮
V08	5008	0x1390

BITS DE ESTADO		
Número del bit	Dirección Modbus	
	Decimal	Hexadecimal
Bit 0	00	00h
Bit 1	01	01h
⋮	⋮	⋮
Bit 15	15	0Fh

BITS DE COMANDO		
Número del bit	Dirección Modbus	
	Decimal	Hexadecimal
Bit 100	100	64h
Bit 101	101	65h
⋮	⋮	⋮
Bit 107	107	6Bh

Los bits de estado (0 hasta 15) tienen la misma función de cada bit de la variable básica 1 (ver ítem 4.2.2), mientras que los bits de comando (100 hasta 107) tienen la misma función de los bits menos significativos de la variable básica 3, sin la necesidad de utilizar la máscara para comandar la SSW-06 (ver ítem 4.2.3).



**¡NOTA!**

Todos los registradores (parámetros y variables básicas) son tratados como registradores del tipo holding. Dependiendo del maestro utilizado, estos registradores son referenciados a partir de la dirección base 40000 o 4x. En este caso, la dirección para un parámetro o variable básica que debe ser programado en el maestro y la dirección presentado en la tabla arriba adicionado de la dirección base. De forma similar, los bits son referenciados a partir de 0000 o 0x, denominados coils. Consulte la documentación del maestro para saber como acceder registradores del tipo holding y coils.

**5.4 DESCRIPCIÓN DETALLADA DE LAS FUNCIONES**

En este ítem es hecha una descripción detallada de las funciones disponibles en el SCA-05 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 registradores son siempre representados en 16 bits. Por eso, es necesario transmitir estos campos utilizando dos bytes (*high y low*).
- Los telegramas, tanto para pregunta cuanto para respuesta, no puede ultrapasar 256 bytes.

■ Los valores transmitidos son siempre números enteros, independiente de que tengan representación con casa decimal. De esta forma, el valor 9,5 sería transmitido como siendo 95 vía serial. Consulte el manual del SCA-05 para obtener la resolución utilizada para cada parámetro.

### 5.4.1 Función 01 – Read Coils

Lee el contenido de un grupo de bits internos que necesariamente deben estar en secuencia numérica. Esta función tiene la siguiente estructura para los telegramas de lectura y respuesta (los valores son siempre hexadecimal, y cada campo representa un byte):

Pregunta (Maestro)	Respuesta (Esclavo)
Dirección del esclavo	Dirección del esclavo
Función	Función
Dirección del bit inicial (byte high)	Campo Byte Count (nr. de bytes de datos)
Dirección del bit inicial (byte low)	Byte 1
Número de bits (byte high)	Byte 2
Número de bits (byte low)	Byte 3
CRC-	etc...
CRC+	CRC-
	CRC+

Cada bit de la respuesta es colocado en una posición de los bytes de datos enviados por el esclavo. El primero byte recibe los 8 primeros bits a partir de la dirección inicial indicada por el maestro. Los demás bytes continúan la secuencia, caso el número de bits de lectura sea mayor que 8. Caso el número de bits leídos no sea múltiplo de 8, los bits restantes del último byte deben ser llenados con 0 (cero).

**Ejemplo 1:** lectura de los bits de estado de la habilitación general (bit 1) y JOG (bit 2) de la SSW-06 en la dirección 1 (supone habilitación general activa y JOG inactivo).

- Dirección: 1 = 01h (1 byte)
- Número del bit inicial: 1 = 0001h (2 bytes)
- Número de bits leídos: 2 = 0002h (2 bytes)

Pregunta (Maestro)		Respuesta (Esclavo)	
Campo	Valor	Campo	Valor
Dirección del esclavo	01h	Dirección del esclavo	01h
Función	01h	Función	01h
Bit inicial (high)	00h	Byte Count	01h
Bit inicial (low)	01h	Estado de los bits 1 y 2	01h
No. de bits (high)	00h	CRC-	D0h
No. de bits (low)	02h	CRC+	49h
CRC-	ECh		
CRC+	0Bh		

En el ejemplo, como el número de bits leídos es menor que 8, el esclavo necesitó de solamente 1 byte para la respuesta. El valor del byte es 01h, que en binario tiene la forma 0000 0001. Como el número de bits leídos es igual a 2, solamente interesa a nosotros los dos bits menos significativos, que tienen los valores 1 = habilitado general y 0 = sin JOG. Los demás bits, como no fueron solicitados, son llenados con 0 (cero).

### 5.4.2 Función 03 – Read Holding Register

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

Pregunta (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+

**Ejemplo 2:** lectura de la corriente del motor en porcentaje (P002) y corriente del motor en amperes (P003) de la Soft-Starter SSW-06 en la dirección 1 (suponiendo P002 = 50.0% y P003 = 40.0 A).

- Dirección: 1 = 01h (1 byte)
- Número del primero parámetro: 2 = 0002h (2 bytes)
- Número de parámetros leídos: 2 = 0002h (2 bytes)
- Valor leído del primero parámetro: 500 = 01F4h (2 bytes)
- Valor leído del segundo parámetro: 400 = 0190h (2 bytes)

Pregunta (Maestro)		Respuesta (Esclavo)	
Campo	Valor	Campo	Valor
Dirección del esclavo	01h	Dirección del esclavo	01h
Función	03h	Función	03h
Registrador inicial (high)	00h	Byte Count	04h
Registrador inicial (low)	02h	P002 (high)	01h
Nr. de registradores (high)	00h	P002 (low)	F4h
Nr. de registradores (low)	02h	P003 (high)	01h
CRC-	65h	P003 (low)	90h
CRC+	CBh	CRC-	BBh
		CRC+	C1h

### 5.4.3 Función 05 – Write Single Coil

Esta función es utilizada para escribir un valor para un único bit (*coil*). El valor para el bit es representado utilizando dos bytes, donde el valor FF00h representa el bit igual a 1, y el valor 0000h representa el bit igual a 0 (cero). Tiene la siguiente estructura (los valores son siempre hexadecimal, y cada campo representa un byte):

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

**Ejemplo 3:** escrita del comando de *reset* (bit 107), en una Soft-Starter en la dirección 1.

- Dirección: 1 = 01h (1 byte)
- Número del bit: 107 = 006Bh (2 bytes)
- Valor para el bit: *reset* = 1, luego el valor que debe ser escrito es FF00h

Pregunta (Maestro)		Respuesta (Esclavo)	
Campo	Valor	Campo	Valor
Dirección del esclavo	01h	Dirección del esclavo	01h
Función	05h	Función	05h
Número del bit (high)	00h	Número del bit (high)	00h
Número del bit (low)	6Bh	Número del bit (low)	6Bh
Valor para el bit (high)	FFh	Valor para el bit (high)	FFh
Valor para el bit (low)	00h	Valor para el bit (low)	00h
CRC-	FDh	CRC-	FDh
CRC+	E6h	CRC+	E6h

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

#### 5.4.4 Función 06 – Write Single Register

Esta función es utilizada para escribir un valor para un único registrador. Tiene la siguiente estructura (los valores son siempre hexadecimal, y cada campo representa un byte):

Pregunta (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+

**Ejemplo 4:** escrita del comando lógico (variable básica 3), con los comandos de habilita rampa y habilita general, para la Soft-Starter SSW-06 en la dirección 3.

- Dirección: 3 = 03h (1 byte)
- Número de la variable: VB03, direccionala en el registrador 5003 = 138Bh (2 bytes)
- Valor para la variable: habilita rampa → máscara en 1 (bit 8) y comando en 1 (bit 0)  
 habilita general → máscara en 1 (bit 9) y comando en 1 (bit 1)  
 luego, valor para el comando = 0303h (2 bytes)

Pregunta (Maestro)		Respuesta (Esclavo)	
Campo	Valor	Campo	Valor
Dirección del esclavo	03h	Dirección del esclavo	03h
Función	06h	Función	06h
Registrador (high)	13h	Registrador (high)	13h
Registrador (low)	8Bh	Registrador (low)	8Bh
Valor (high)	03h	Valor (high)	03h
Valor (low)	03h	Valor (low)	03h
CRC-	BCh	CRC-	BCh
CRC+	77h	CRC+	77h

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

#### 5.4.5 Función 15 – Write Multiple Coils

Esta función permite escribir valores para un grupo de bits (*coils*), que deben estar en secuencia numérica. También puede ser utilizada para escribir un único bit (los valores son siempre hexadecimal, y cada campo representa un byte).

Pregunta (Maestro)	Respuesta (Esclavo)
Dirección del esclavo	Dirección del esclavo
Función	Función
Dirección del bit inicial (byte high)	Dirección del bit inicial (byte high)
Dirección del bit inicial (byte low)	Dirección del bit inicial (byte low)
Número de bits (byte high)	Número de bits (byte high)
Número de bits (byte low)	Número de bits (byte low)
Campo Byte Count (no. de bytes de datos)	CRC-
Byte 1	CRC+
Byte 2	
Byte 3	
etc...	
CRC-	
CRC+	
Dirección del esclavo	

El valor de cada bit que está siendo escrito es colocado en una posición de los bytes de datos enviados por el maestro. El primero byte recibe los 8 primeros bits a partir de la dirección inicial indicada por el maestro. Los demás bytes (si el número de bits escritos es mayor que 8), continúan la secuencia. Caso el número de bits escritos no sea múltiplo de 8, los bits restantes del último byte deben ser llenados con 0 (cero).

**Ejemplo 5:** escrita de los bits 100 y 101 para habilitar la rampa y habilitar general una Soft-Starter SSW-06 en la dirección 20

- Dirección: 20 = 14h (1 byte)
- Número de primero bit: 100 = 0064h (2 bytes)
- Número de bits: 2 = 0002h (2 bytes)
- Valor para los bits: los dos bits deben ser colocados en 1, entonces valor = 03h (1 byte)

Pregunta (Maestro)		Respuesta (Esclavo)	
<i>Campo</i>	<i>Valor</i>	<i>Campo</i>	<i>Valor</i>
Dirección del esclavo	14h	Dirección del esclavo	14h
Función	0Fh	Función	0Fh
Bit inicial (byte high)	00h	Bit inicial (byte high)	00h
Bit inicial (byte low)	64h	Bit inicial (byte low)	64h
No. de bits (byte high)	00h	No. de bits (byte high)	00h
No. de bits (byte low)	02h	No. de bits (byte low)	02h
Byte Count	01h	CRC-	97h
Valor para los bits	03h	CRC+	10h
CRC-	2Eh		
CRC+	6Dh		

### 5.4.6 Función 16 – Write Multiple Registers

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

Pregunta (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+	

## Protocolo Modbus-RTU

**Ejemplo 6:** escrita del valor para las salidas analógicas AO1 y AO2, a través de las variables básicas 9 y 10, para una Soft-Starter SSW-06 en la dirección 15. La salida AO1 debe ser igual a 50% (resolución de 14 bits, 50% = 8191) y AO2 igual a 25% (resolución de 14 bits, 25% = 4095).

- Dirección: 15 = 0Fh (1 byte)
- Número de la primera variable: VB09, direccionada en el registrador 5009 = 1391h (2 bytes)
- Valor para el primero parámetro: 8191 = 1FFFh (2 bytes)
- Valor para el segundo parámetro: 4095 = 0FFFh (2 bytes)

Pregunta (Maestro)		Respuesta (Esclavo)	
Campo	Valor	Campo	Valor
Dirección del esclavo	0Fh	Dirección del esclavo	0Fh
Función	10h	Función	10h
Registrador inicial (high)	13h	Registrador (high)	13h
Registrador inicial (low)	91h	Registrador (low)	91h
No. de registradores (high)	00h	Valor (high)	00h
No. de registradores (low)	02h	Valor (low)	02h
Byte Count	04h	CRC-	15h
VB09 (high)	1Fh	CRC+	8Fh
VB09 (low)	FFh		
VB10 (high)	0Fh		
VB10 (low)	FFh		
CRC-	A4h		
CRC+	83h		

### 5.4.7 Función 43 – Read Device Identification

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

Pregunta (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 primero objeto
	Tamaño del primero objeto
	Valor del primero 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 la Soft-Starter SSW-06, solamente informaciones básicas están disponibles, formadas por tres objetos:

- Objeto 0x00 – Vendor Name: siempre 'WEG'.
- Objeto 0x01 – Product Code: formado por el código del producto (SSW-06) más la corriente nominal de la sofá-starter (ejemplo: 'SSW-06 85.0A').
- Objeto 0x02 – Major Minor Revision: indica la versión de firmware de la soft starter, no formato 'VX.XX'.

El código de lectura indica cuales las categorías de informaciones están siendo leídas, y si los objetos están siendo acezados en secuencia o individualmente. En el caso, la SSW-06 soporta los códigos 01 (informaciones básicas en secuencia), y 04 (acceso individual a los objetos). Los demás campos para la SSW-06 tienen valores fijos.

## Protocolo Modbus-RTU

**Ejemplo 7:** lectura de las informaciones básicas en secuencia, a partir del objeto 00h, de una Soft-Starter SSW-06 en la dirección 1:

Pregunta (Maestro)		Respuesta (Esclavo)	
<i>Campo</i>	<i>Valor</i>	<i>Campo</i>	<i>Valor</i>
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	00h	Conformity Level	81h
CRC-	70h	More Follows	00h
CRC+	77h	Próximo Objeto	00h
		Número de objetos	03h
		Código del Objeto	00h
		Tamaño del Objeto	03h
		Valor del Objeto	'WEG'
		Código del Objeto	01h
		Tamaño del Objeto	14h
		Valor del Objeto	'SSW-06 85.0A'
		Código del Objeto	02h
		Tamaño del Objeto	05h
		Valor del Objeto	'V1.11'
		CRC-	6Fh
		CRC+	10h

En este ejemplo, el valor de los objetos no son representado en hexadecimal, pero si utilizando los caracteres ASCII correspondientes. Por ejemplo, para el objeto 00h, el valor 'WEG' es transmitido como siendo tres caracteres ASCII, que en hexadecimal tiene los valores 57h ('W'), 45h ('E') y 47h ('G').

### 5.4.8 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, la Soft-Starter SSW-06 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, o esclavo no va a responder 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 respuesta del esclavo. En el caso de una recepción con suceso, durante el tratamiento del telegrama, la soft-starter puede detectar problemas y enviar un mensaje de error, indicando el tipo de problema encontrado:

- Función no válido (código del error = 1): la función solicitada no está implementada para el equipamiento.
- Dirección de dato no válido (código del error = 2): la dirección del dato (parámetro) no existe.
- Valor de dato no válido (código del error = 3): ocurre en las siguientes situaciones:
  - Valor está fuera del rango permitido.
  - Escrita en dato que no puede ser alterado (registrador solamente lectura).
  - Comando no está habilitado para ser ejecutado vía serial



#### ¡NOTA!

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

En el caso da 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 enviadas por el esclavo tienen a siguiente estructura:

## Protocolo Modbus-RTU

Pregunta (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 8:** maestro solicita para el esclavo en la dirección 1 es escrita en el parámetro 89 (parámetro no existe):

Pregunta (Maestro)		Respuesta (Esclavo)	
<i>Campo</i>	<i>Valor</i>	<i>Campo</i>	<i>Valor</i>
Dirección del esclavo	0x01	Dirección del esclavo	0x01
Función	0x06	Función	0x86
Registrador (high)	0x00	Código de error	0x02
Registrador (low)	0x59	CRC-	0xC3
Valor (high)	0x00	CRC+	0xA1
Valor (low)	0x00		
CRC-	0x59		
CRC+	0xD9		



## Apendices

```

    uchCRCLo = uchCRCHi ^ auchCRCHi[uIndex];
    uchCRCHi = auchCRCLo[uIndex];
}

return (uchCRCHi << 8 | uchCRCLo);
}

```

### Apendice B - Cálculo del CRC utilizando desplazamiento de registradores

En este ítem es descrito el algoritmo para el cálculo del CRC utilizado en la comunicación Modbus-RTU, a través del desplazamiento de registradores. El algoritmo fue obtenido y es explicado en los documentos referenciados en el ítem 5.

El cálculo del CRC es iniciado primeramente cargándose una variable de 16 bits (referenciado a partir de ahora como variable CRC) con el valor 0xFFFF. después ejecutase los pasos de acuerdo con la siguiente rutina:

1. Sometiese el primero byte del mensaje (solamente los bits de datos - start bit , paridad y stop bit no son utilizados) hay una lógica XOR (OU exclusivo) con los 8 bits menos significativos de la variable CRC, regresando el resultado en la propia variable CRC.
2. Entonces, la variable CRC es desplazada una posición hacia derecha, en dirección al bit menos significativo, y la posición del bit más significativo es llenada con 0 (cero).
3. Después este desplazamiento, el bit de *flag* (bit que fue desplazado para fuera de la variable CRC) es analizado, ocurriendo lo siguiente:
  - Si el valor del bit es 0 (cero), nada es hecho
  - Si el valor del bit es 1, el contenido de la variable CRC es sometido a una lógica XOR con un valor constante de 0xA001 y el resultado es retornado a la variable CRC.
4. Repiten los pasos 2 y 3 hasta que ocho desplazamientos tengan sido hechos.
5. Repiten los pasos de 1 a 4, utilizando el próximo byte del mensaje, hasta que toda el mensaje tenga sido procesado.

El contenido final de la variable CRC es el valor del campo CRC que es transmitido en el final del telegrama. La parte menos significativa es transmitida primero (CRC-) y en seguida la parte más significativa (CRC+).