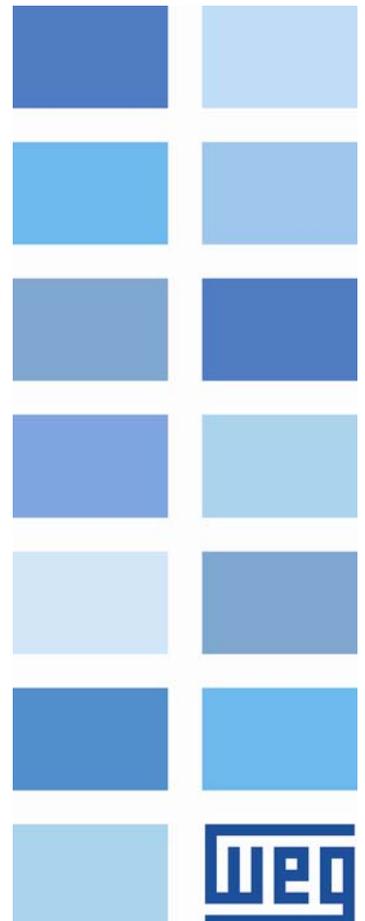


# WEGTP

## SCA06

### Manual do Usuário





# **Manual do Usuário WEGTP**

Série: SCA06

Idioma: Português

N ° do Documento: 10001626006 / 00

Data da Publicação: 02/2012

# SUMÁRIO

<b>SUMÁRIO</b> .....	<b>3</b>
<b>1 SOBRE O MANUAL</b> .....	<b>4</b>
<b>ABREVIações E DEFINIções</b> .....	4
<b>REPRESENTAÇÃO NUMÉRICA</b> .....	4
<b>2 INTRODUÇÃO À COMUNICAÇÃO SERIAL</b> .....	<b>5</b>
<b>3 DESCRIÇÃO DAS INTERFACES</b> .....	<b>6</b>
<b>3.1 MÓDULO DE EXPANÇÃO E COMUNICAÇÃO RS232 E RS485 ECO1</b> .....	6
<b>3.2 RS232</b> .....	6
3.2.1 Indicações.....	6
3.2.2 Conexão com a Rede RS232.....	6
3.2.3 Cabos para Ligação em RS232.....	6
3.2.4 Pinagem do Conector.....	6
<b>3.3 RS485</b> .....	7
3.3.1 Indicações.....	7
3.3.2 Características da interface RS485.....	7
3.3.3 Pinagem do Conector.....	7
3.3.4 Resistor de terminação.....	8
3.3.5 Conexão com a Rede RS485.....	8
<b>4 PARAMETRIZAÇÃO</b> .....	<b>9</b>
<b>4.1 SÍMBOLOS PARA DESCRIÇÃO DAS PROPRIEDADES</b> .....	9
P00650 – ENDEREÇO DO SERVOCONVERSOR NA COMUNICAÇÃO SERIAL 1 – RS232.....	9
P00656 – ENDEREÇO DO SERVOCONVERSOR NA COMUNICAÇÃO SERIAL 2 – RS485.....	9
P00652 – BIT RATE SERIAL 1 – RS232.....	9
P00658 – BIT RATE SERIAL 2 – RS485.....	9
P00653 – CONFIGURAÇÃO SERIAL 1 – RS232.....	10
P00659 – CONFIGURAÇÃO SERIAL 2 – RS485.....	10
P00654 – SELECIONA PROTOCOLO SERIAL 1 – RS232.....	10
P00660 – SELECIONA PROTOCOLO SERIAL 2 – RS485.....	10
P00662 – AÇÃO PARA ERRO DE COMUNICAÇÃO.....	10
P00663 – WATCHDOG SERIAL.....	11
P00664 – SALVA PARÂMETROS EM MEMÓRIA NÃO VOLÁTIL.....	11
P00667 – SALVA EM MARCADORES.....	11
<b>5 PROTOCOLO WEGTP</b> .....	<b>13</b>
5.1 ENDEREÇAMENTO NO PROTOCOLO WEGTP.....	13
5.2 CAMPOS DO PROTOCOLO.....	13
5.3 FORMATO DOS TELEGRAMAS.....	14
5.3.1 Telegrama de leitura.....	15
5.3.2 Telegrama de escrita.....	15
5.4 EXEMPLO DE TELEGRAMAS UTILIZANDO O PROTOCOLO WEGTP.....	15
<b>6 FALHAS E ALARMES RELACIONADOS COM A COMUNICAÇÃO WEGTP</b> .....	<b>17</b>
A00128/F00028 – TIMEOUT NA RECEPÇÃO DE TELEGRAMAS.....	17
<b>I. APÊNDICES</b> .....	<b>18</b>
<b>APÊNDICE A. TABELA ASCII</b> .....	18

## **1 SOBRE O MANUAL**

Este manual fornece a descrição necessária para a operação do servoconversor SCA06 utilizando o protocolo WEGTP. Este manual deve ser utilizado em conjunto com manual do usuário do SCA06.

### **ABREVIações E DEFINIções**

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

### **REPRESENTAÇÃO NUMÉRICA**

Números decimais são representados através de dígitos sem sufixo. Números hexadecimais são representados com a letra 'h' depois do número.

## **2 INTRODUÇÃO À COMUNICAÇÃO SERIAL**

Em uma interface serial os bits de dados são enviados sequencialmente através de um canal de comunicação ou barramento. Diversas tecnologias utilizam comunicação serial para transferência de dados, incluindo as interfaces RS232 e RS485.

As normas que especificam os padrões RS232 e RS485, no entanto, não especificam o formato nem a sequência de caracteres para a transmissão e recepção de dados. Neste sentido, além da interface, é necessário identificar também o protocolo utilizado para comunicação. Dentre os diversos protocolos existentes, um protocolo muito utilizado na indústria é o protocolo WEGTP.

A seguir serão apresentadas características das interfaces seriais RS232 e RS485 disponíveis para o produto, bem como a descrição detalhada dos protocolos para utilização destas interfaces.

### 3 DESCRIÇÃO DAS INTERFACES

Para possibilitar a comunicação serial no servoconversor SCA06, é necessário utilizar o módulo de expansão e comunicação ECO1 descrito a seguir. Informações sobre a instalação deste módulo podem ser obtidas no guia que acompanha o acessório.

#### 3.1 MÓDULO DE EXPANÇÃO E COMUNICAÇÃO RS232 E RS485 ECO1



- Item WEG: 11330271.
- Composto pelo módulo de comunicação ECO1 (figura ao lado) e um guia de montagem.
- Interface isolada galvanicamente e com sinal diferencial, conferindo maior robustez contra interferência eletromagnética.
- Os sinais RS232 e RS485 são canais independentes, podendo ser utilizados simultaneamente.

#### 3.2 RS232

##### 3.2.1 Indicações

O led LA121 indica (acende) quando há transmissão de dados através da comunicação RS232.

##### 3.2.2 Conexão com a Rede RS232

- Os sinais RX e TX do inversor devem ser ligados respectivamente aos sinais TX e RX do mestre, além da conexão do sinal de referência (GND).
- A interface RS232 é muito susceptível a interferências. Por este motivo, o cabo utilizado para comunicação deve ser o mais curto possível – sempre menor que 10 metros – e deve ser colocado em separado da fiação de potência que alimenta o inversor e motor.

##### 3.2.3 Cabos para Ligação em RS232

Caso seja desejado, estão disponíveis itens dos seguintes cabos para ligação em RS232 entre o servoconversor e um mestre da rede, como um PC:

Cabo	Item
Cabo RS232 blindado com conectores DB9 Comprimento: 3 metros	10050328
Cabo RS232 blindado com conectores DB9 fêmea Comprimento: 10 metros	10191117

Outros cabos, porém, podem ser encontrados no mercado – em geral denominados null-modem – ou montados de acordo com o desejado para a instalação.

##### 3.2.4 Pinagem do Conector

A conexão para a interface RS232 está disponível através dos conectores XA121 e XA122 utilizando a seguinte pinagem:

**Tabela 3.1:** Pinagem do conector para RS232 XA121

Pino	Função
1	Terra
2	RX_232
3	TX_232
4	Reservado <sup>1</sup>
5	GND
6	Reservado
7	232 RTS
8	A
9	B

**Tabela 3.2:** Pinagem do conector para RS232 XA122

Pino	Função
1	NC
2	RX_232
3	TX_232
4	Reservado
5	GND
6	NC
7	232 RTS
8	NC
9	NC
Caraça	Terra

### 3.3 RS485

#### 3.3.1 Indicações

O led LA122 indica (acende) quando há transmissão de dados através da comunicação RS485.

#### 3.3.2 Características da interface RS485

- Interface segue o padrão EIA/TIA-485.
- Pode operar como escravo da rede WEGTP.
- Possibilita comunicação utilizando taxas de 9600 até 57600 Kbit/s.
- Interface isolada galvanicamente e com sinal diferencial, conferindo maior robustez contra interferência eletromagnética.
- Permite a conexão de até 32 dispositivos no mesmo segmento. Uma quantidade maior de dispositivos pode ser conectada com o uso de repetidores.<sup>2</sup>
- Comprimento máximo do barramento de 1000 metros.

#### 3.3.3 Pinagem do Conector

A conexão para a interface RS485 está disponível através do conector XC1 utilizando a seguinte pinagem:

**Tabela 3.3:** Pinagem do conector para RS485 XA121

Pino	Função
1	Terra
2	RX_232
3	TX_232
4	Reservado
5	GND
6	Reservado
7	232 RTS
8	A
9	B

<sup>1</sup> Não conectar os pinos reservados.

<sup>2</sup> O número limite de equipamentos que podem ser conectados na rede também depende do protocolo utilizado.

**Tabela 3.4:** Pinagem do conector para RS485 XA123

Pino	Função
1	NC
2	NC
3	NC
4	NC
5	GND
6	Reservado
7	NC
8	A (data -)
9	B (data +)
Carcaça	Terra

### 3.3.4 Resistor de terminação

Para cada segmento da rede RS485, é necessário habilitar um resistor de terminação nos pontos extremos do barramento principal. O acessório ECO1 possui duas chaves DIP Switch que podem ser ativadas (colocando ambas as chaves SA121 na posição ON) para habilitar o resistor de terminação.


**Figura 3.1:** Resistor de terminação

### 3.3.5 Conexão com a Rede RS485

Para a ligação do servoconversor SCA06 utilizando a interface RS485, os seguintes pontos devem ser observados:

- É recomendado o uso de um cabo com par trançado blindado.
- Recomenda-se também que o cabo possua mais um fio para ligação do sinal de referência (GND). Caso o cabo não possua o fio adicional, deve-se deixar o sinal GND desconectado.
- A passagem do cabo deve ser feita separadamente (e se possível distante) dos cabos para alimentação de potência.
- Todos os dispositivos da rede devem estar devidamente aterrados, preferencialmente na mesma ligação com o terra. A blindagem do cabo também deve ser aterrada.
- Habilitar os resistores de terminação apenas em dois pontos, nos extremos do barramento principal, mesmo que existam derivações a partir do barramento.

## 4 PARAMETRIZAÇÃO

A seguir serão apresentados apenas os parâmetros do servoconversor SCA06 que possuem relação direta com a comunicação WEGTP.

### 4.1 SÍMBOLOS PARA DESCRIÇÃO DAS PROPRIEDADES

RW	Parâmetro de leitura e escrita
AC	Parâmetro visível na HMI somente quando o acessório correspondente esta conectado

#### P00650 – ENDEREÇO DO SERVOCONVERSOR NA COMUNICAÇÃO SERIAL 1 – RS232

#### P00656 – ENDEREÇO DO SERVOCONVERSOR NA COMUNICAÇÃO SERIAL 2 – RS485

<b>Faixa de Valores:</b>	1 a 247	<b>Padrão:</b> 1
<b>Propriedades:</b>	RW AC	

#### Descrição:

Permite programar o endereço utilizado para comunicação serial do equipamento. É necessário que cada equipamento da rede possua um endereço diferente dos demais. Os endereços válidos para este parâmetro dependem do protocolo programado no servoconversor.

- P00654/P00660 = 1 (WEGTP) → endereços válidos: 1 to 30.
- P00654/P00660 = 2 (Modbus RTU) → endereços válidos: 1 to 247.

#### P00652 – BIT RATE SERIAL 1 – RS232

#### P00658 – BIT RATE SERIAL 2 – RS485

<b>Faixa de Valores:</b>	0 = 4800 bits/s 1 = 9600 bits/s 2 = 14400 bits/s 3 = 19200 bits/s 4 = 24000 bits/s 5 = 28800 bits/s 6 = 33600 bits/s 7 = 38400 bits/s 8 = 43200 bits/s 9 = 48000 bits/s 10 = 52800 bits/s 11 = 57600 bits/s	<b>Padrão:</b> 1
<b>Propriedades:</b>	RW AC	

#### Descrição:

Permite programar o valor desejado para a taxa de comunicação da interface serial, em bits por segundo. Esta taxa deve ser a mesma para todos os equipamentos conectados na rede.

**P00653 – CONFIGURAÇÃO SERIAL 1 – RS232**
**P00659 – CONFIGURAÇÃO SERIAL 2 – RS485**

<b>Faixa de Valores:</b>	0 = 8 bits de dados, sem paridade, 1 stop bit 1 = 8 bits de dados, paridade par, 1 stop bit 2 = 8 bits de dados, paridade ímpar, 1 stop bit 3 = 8 bits de dados, sem paridade, 2 stop bits 4 = 8 bits de dados, paridade par, 2 stop bits 5 = 8 bits de dados, paridade ímpar, 2 stop bits 6 = 7 bits de dados, sem paridade, 1 stop bits 7 = 7 bits de dados, paridade par, 1 stop bits 8 = 7 bits de dados, paridade ímpar, 1 stop bits 9 = 7 bits de dados, sem paridade, 2 stop bits 10 = 7 bits de dados, paridade par, 2 stop bits 11 = 7 bits de dados, paridade ímpar, 2 stop bits	<b>Padrão: 3</b>
<b>Propriedades:</b> RW AC		

**Descrição:**

Permite a configuração do número de bits de dados, paridade e *stop* bits nos bytes da interface serial. Esta configuração deve ser a mesma para todos os equipamentos conectados na rede.

**P00654 – SELECIONA PROTOCOLO SERIAL 1 – RS232**
**P00660 – SELECIONA PROTOCOLO SERIAL 2 – RS485**

<b>Faixa de Valores:</b>	1 = WEGTP 2 = Modbus RTU	<b>Padrão: 1</b>
<b>Propriedades:</b> RW		

**Descrição:**

Permite selecionar o protocolo desejado para a interface serial. A descrição detalhada do protocolo é feita no item 5 deste manual.

**P00662 – AÇÃO PARA ERRO DE COMUNICAÇÃO**

<b>Faixa de Valores:</b>	0 = Inativo 1 = Causa Falha 2 = Causa alarme e Stop	<b>Padrão: 1</b>
<b>Propriedades:</b> RW		

**Descrição:**

Este parâmetro permite selecionar qual a ação deve ser executada pelo equipamento, caso ele seja controlado via rede e um erro de comunicação seja detectado.

**Tabela 4.1:** Opções para o parâmetro P00662

Opção	Descrição
0 = Apenas indica alarme	Apenas mostra o código de alarme na HMI do servoconversor.
1 = Causa Falha	Causa falha e o servoconversor só volta a operar caso seja feito reset de falhas.
2 = Indica alarme e executa STOP	Será feita a indicação de alarme juntamente com a execução do comando STOP. Para que o servo saia desta condição, será necessário realizar o reset de falhas ou desabilitar o drive.
3 = Indica alarme e desabilita drive	Será feita a indicação de alarme juntamente com a execução do comando desabilita.

São considerados erros de comunicação os seguintes eventos:

Comunicação Serial (RS232/RS485):

- Alarme A00128/Falha F00028: *timeout* da interface serial.

### P00663 – WATCHDOG SERIAL

**Faixa de Valores:** 0,0 a 999,0s **Padrão:** 0,0  
**Propriedades:** RW

**Descrição:**

Permite programar um tempo para a detecção de erro de comunicação via interface serial. Caso o inversor fique sem receber telegramas válidos por um tempo maior do que o programado neste parâmetro, será considerado que ocorreu um erro de comunicação, mostrado o alarme A00128 na HMI (ou falha F00028, dependendo da programação feita no P00662) e a ação programada no P00662 será executada.

Após energizado, o inversor começará a contar este tempo a partir do primeiro telegrama válido recebido. O valor 0,0 desabilita esta função.

### P00664 – SALVA PARÂMETROS EM MEMÓRIA NÃO VOLÁTIL

**Faixa de Valores:** 0 = Não salva parâmetro na memória não volátil **Padrão:** 1  
 1 = Salva parâmetro na memória volátil  
**Propriedades:** RW

**Descrição:**

Permite selecionar se a escrita de parâmetros via serial deve ou não salvar o conteúdo dos parâmetros em memória não volátil (EEPROM). Quando utilizado o protocolo ModBus e apenas esse parâmetro que determina se os parâmetros escritos via serial serão ou não salvos na memória não volátil. Porém, quando utilizado o protocolo WegTP, deve-se observar que no byte de código do telegrama consta a informação sobre salvar ou não o parâmetro na EEPROM. Para que via WegTP os mesmos sejam salvos em memória não volátil, e necessário que as duas informações, o byte de código do telegrama e o parâmetro P00664, sejam verdadeiras.



**NOTA!**

Este tipo de memória possui um numero limite de escritas (100.000 vezes). Dependendo da aplicação, este limite pode ser ultrapassado, caso alguns parâmetros sejam escritos ciclicamente via serial (referencia de velocidade, torque, etc.). Nestes casos, pode ser desejado que, durante a operação do servoconversor, a escrita via serial não salve o conteúdo dos parâmetros em memória não volátil, para não ultrapassar o limite de escritas no servoconversor.



**NOTA!**

Esse parâmetro não se aplica quando a escrita e feita utilizando a interface USB.

### P00667 – SALVA EM MARCADORES

**Faixa de Valores:** 0 = Lê e escreve normalmente o conteúdo no parâmetro correspondente **Padrão:** 0  
 1 = Lê e escreve conteúdo em marcadores de WORD volátil a partir do MW13000  
**Propriedades:** RW

**Descrição:**

Propriedade verificada quando parâmetro e escrito e lido via serial. Seleciona se quer que conteúdo a ser escrito/lido seja salvo em parâmetro ou em marcador de Word volátil.

**NOTA!**

Sendo este parâmetro  $P00667 = 1$ , ao escrever via serial no parâmetro  $P00105 = 30$ , o conteúdo do parâmetro será armazenado no marcador de Word 13105 ( $MWinicial + Numero\_par \Rightarrow 13000 + 105$ ). Portanto,  $MW13105 = 30$ .

Observação: Uma vez que  $P00667 = 1$ , o mesmo não poderá ser alterado via serial. Pois na tentativa de escrever no parâmetro  $P00667$  estará escrevendo no marcador de Word  $P13667$ .

## 5 PROTOCOLO WEGTP

O WEGTP foi desenvolvido com o objetivo de possibilitar a comunicação com CLPs da linha TP. Mas devido a sua flexibilidade e facilidade de uso, tem sido utilizado em outras aplicações, sendo muitas vezes implementado em CLPs e outros sistemas para controle e monitoração dos equipamentos WEG.

Nestes documentos estão definidos os formatos das mensagens utilizados pelos elementos que fazem parte da rede WEGTP, os serviços (ou funções) que podem ser disponibilizados via rede, e também como estes elementos trocam dados na rede.

### 5.1 ENDEREÇAMENTO NO PROTOCOLO WEGTP

Para o protocolo WEGTP, durante a transmissão de telegramas, o endereço selecionado no parâmetro endereço do servoconversor na comunicação serial é representado por um caractere ASCII, de acordo com a tabela a seguir:

**Tabela 5.1:** Endereço para o protocolo WEGTP

Endereço	ASCII	hexadecimal	Endereço	ASCII	hexadecimal
0	@	0x40	16	P	0x50
1	A	0x41	17	Q	0x51
2	B	0x42	18	R	0x52
3	C	0x43	19	S	0x53
4	D	0x44	20	T	0x54
5	E	0x45	21	U	0x55
6	F	0x46	22	V	0x56
7	G	0x47	23	W	0x57
8	H	0x48	24	X	0x58
9	I	0x49	25	Y	0x59
10	J	0x4A	26	Z	0x5A
11	K	0x4B	27	[	0x5B
12	L	0x4C	28	\	0x5C
13	M	0x4D	29	]	0x5D
14	N	0x4E	30	^	0x5E
15	O	0x4F	31	_	0x5F

Endereços para executar tarefas especiais:

- Endereço 0: qualquer servoconversor da rede é consultado, independentemente de seu endereço. Deve-se ter apenas um servoconversor ligado a rede (ponto-a-ponto) para que não ocorram curto-circuitos nas linhas de interface.
- Endereço 31: um comando pode ser transmitido simultaneamente para todos os servoconversores da rede, sem reconhecimento de aceitação.

### 5.2 CAMPOS DO PROTOCOLO

- **STX**: Byte de "Start of Transmition": Valor: 0x02 (hexadecimal); 2 (decimal).
- **ETX**: Byte de "End of Transmition": Valor: 0x03 (hexadecimal); 3 (decimal).
- **ADR**: Byte do endereço do servoconversor na rede  
Faixa de Valores: 0x41 (hexadecimal); 65 (decimal); 'A' (ASCII) ... 0x5E (hexadecimal); 94 (decimal); '^' (ASCII) → Representam os valores de 1 ... 30 programados no parâmetro do endereço do servoconversor.  
Especial 1: 0x40 (hexadecimal); 64 (decimal); '@' (ASCII) → Permite escrita ou leitura de todos os equipamentos conectados a rede.  
Especial 2: 0x5F (hexadecimal); 95 (decimal); '\_' (ASCII) → Permite SOMENTE escrita em todos os equipamentos conectados a rede sem resposta de aceitação ou rejeição.
- **COD**: Byte do Código do Telegrama  
Leitura: 0x3C (hexadecimal); 60 (decimal); '<' (ASCII)  
Escrita: 0x3D (hexadecimal); 61 (decimal); '=' (ASCII) sem salvar o parâmetro na EEPROM  
Escrita: 0x3E (hexadecimal); 62 (decimal); '>' (ASCII) salvando o parâmetro na EEPROM

- **BCC:** Byte de Checksum longitudinal do telegrama, ou seja, OU EXCLUSIVO entre todos os bytes do telegrama. Tamanho de 1 byte (0x00 ... 0xFF hexadecimal)
- **DMW:** “Data Master Write”. São 4 bytes de escrita que o mestre envia ao escravo, sendo que os 2 primeiros representam o parâmetro e/ou a variável básica e os 2 últimos o valor a ser escrito neste parâmetro.  
PHi: Byte representando a parte alta do parâmetro  
PLo: Byte representando a parte baixa do parâmetro  
VHi: Byte representando a parte alta do valor a ser escrito  
VLo: Byte representando a parte baixa do valor a ser escrito  
*Exemplo:* Escrever 2000 rpm na referência de velocidade (P0121) → PHi = 0x00 (hexadecimal), PLo = 0x79 (hexadecimal), VHi = 0x07 (hexadecimal), VLo = 0xD0 (hexadecimal).
- **DMR:** “Data Master Read”. São 2 bytes de leitura que o mestre envia ao escravo que representam o parâmetro a ser lido.  
PHi: Byte representando a parte alta do parâmetro  
PLo: Byte representando a parte baixa do parâmetro  
*Exemplo:* Ler o valor contido no parâmetro do estado das DIs (P0008) → PHi = 0x00 (hexadecimal), PLo = 0x08 (hexadecimal).
- **NUM:** Byte que representa o número de DMW ou DMR a serem transmitidos, conforme o COD do telegrama.  
 Faixa de valores: 1 ... 6 (decimal)
- **DSV:** “Data Slave Value”. São 2 bytes que o escravo envia ao mestre após uma solicitação de um telegrama de leitura do mestre, representando o valor contido no parâmetro solicitado.  
VHi: Byte representando a parte alta do valor a ser escrito  
VLo: Byte representando a parte baixa do valor a ser escrito  
*Exemplo:* Resposta a solicitação de leitura do parâmetro de habilitação (P0099) → VHi = 0x00 (hexadecimal), VLo = 0x01 (hexadecimal), informando que o servoconversor está habilitado.
- **ACK:** Byte de aceitação do escravo após uma escrita do mestre  
 Valor: 0x06 (hexadecimal); 6 (decimal);
- **NAK:** Byte de rejeição do escravo após uma leitura ou escrita do mestre. Pode ocorrer quando o mestre solicita uma escrita ou leitura de um parâmetro inexistente ou o valor a ser escrito no parâmetro está fora da faixa de valores permitida,  
 Valor: 0x15 (hexadecimal); 21 (decimal);

### 5.3 FORMATO DOS TELEGRAMAS

A seguir serão apresentados os formatos dos telegramas de leitura e escrita em parâmetros. É importante observar que cada telegrama no protocolo WEGTP permite realizar a leitura ou escrita de até 6 parâmetros por vez. Telegramas que possuem erro no formato ou BCC incorreto serão ignorados pelo servoconversor, que não enviará resposta para o mestre.



**NOTA!**

O tempo de escrita na EEPROM é de 10ms por parâmetro portanto é necessário cuidar-se para não sobrecarregar o servoconversor com muitos telegramas seguidos pois isto pode fazer com que o servoconversor ignore os últimos telegramas de modo a ter tempo de escrever todos parâmetros na EEPROM (quando isto ocorre o servoconversor indica alarme 107).



**NOTA!**

O número de escritas na memória EEPROM limita a vida útil desta, portanto é recomendável não se salvar na EEPROM parâmetros que são escritos muitas vezes por dia, o usuário deve salvar na EEPROM apenas aqueles parâmetros em que isto é realmente necessário.

### 5.3.1 Telegrama de leitura

Mestre:

STX	ADR	COD	NUM	DMR	...	DMR	ETX	BCC
-----	-----	-----	-----	-----	-----	-----	-----	-----

- COD: código para leitura → 0x3C (hexadecimal); 60 (decimal); '<' (ASCII)
- NUM: número de parâmetros lidos. Faixa de 1 ... 6.
- DMR: número do parâmetro solicitado. O número de DMRs deve ser igual ao valor configurado no byte NUM.

Escravo:

ADR	DSV	...	DSV	BCC	ou	ADR	NAK
-----	-----	-----	-----	-----	----	-----	-----

- DSV: valor do parâmetro solicitado. O número de DSVs é igual ao valor configurado no byte NUM

Lembrando que:

DMR		DSV	
PHi	PLo	VHi	VLo

### 5.3.2 Telegrama de escrita

Mestre:

STX	ADR	COD	NUM	DMW	...	DMW	ETX	BCC
-----	-----	-----	-----	-----	-----	-----	-----	-----

- COD: código para escrita
  - 0x3E (hexadecimal); 62 (decimal); '>' (ASCII) → salvando na EEPROM
  - 0x3D (hexadecimal); 61 (decimal); '=' (ASCII) → sem salvar na EEPROM
- NUM: número de parâmetros escritos. Faixa de 1 ... 6.
- DMW: número e conteúdo para o parâmetro. O número de DMWs deve ser igual ao valor configurado no byte NUM.

Escravo:

ADR	ACK	ou	ADR	NAK
-----	-----	----	-----	-----

Lembrando que:

DMW			
PHi	PLo	VHi	VLo

## 5.4 EXEMPLO DE TELEGRAMAS UTILIZANDO O PROTOCOLO WEGTP

Todos exemplos a seguir consideram que o servoconversor está programado com o endereço 1, logo o campo ADR é setado em 41.

Exemplo 1: leitura de dois parâmetros do servoconversor:

- Velocidade do servomotor - P0002 (supondo P0002 em 1200rpm = 0x04B0).
- Estado do servoconversor - P0006 (supondo P0006 em 1 = 0x0001).

Mestre:

0x02	0x41	0x3C	0x02	0x00	0x02	0x00	0x06	0x03	0x7A
STX	ADR	COD	NUM	DMR:P0002		DMR:P0006		ETX	BCC
				Parâmetro		Parâmetro			

Escravo:

0x41	0x04	0xB0	0x00	0x01	0xF4
ADR	DSV:1200		DSV:1		BCC
	Valor		Valor		

Exemplo 2: Mudar o servoconversor para modo ladder:

- Para isto é necessário colocar o parâmetro P0202 em 4.
- Telegrama de escrita salvando na EEPROM.
- P0202 = 4 (202 em decimal = 0x00CA, 4 em decimal = 0x0004)

Mestre:

0x02	0x41	0x3E	0x01	0x00	0xCA	0x00	0x04	0x03	0xB1
STX	ADR	COD	NUM	DMW: P0202 = 4				ETX	BCC
				Parâmetro		Valor			

Escravo:

0x41	0x06
ADR	ACK

## **6 FALHAS E ALARMES RELACIONADOS COM A COMUNICAÇÃO WEGTP**

### **A00128/F00028 – TIMEOUT NA RECEPÇÃO DE TELEGRAMAS**

#### **Descrição:**

Alarme que indica falha na comunicação serial. Indica que o equipamento parou de receber telegramas seriais válidos por um período maior do que o programado no P00663.

#### **Atuação:**

O parâmetro P00663 permite programar um tempo dentro do qual o servoconversor deverá receber ao menos um telegrama válido via interface serial RS485 – com endereço e campo de checagem de erros corretos – caso contrário será considerado que houve algum problema na comunicação serial. A contagem do tempo é iniciada após a recepção do primeiro telegrama válido. Esta função pode ser utilizada para qualquer protocolo serial suportado pelo servoconversor.

Depois de identificado o timeout na comunicação serial, será sinalizada através da HMI a mensagem de alarme A00128 – ou falha F00028, dependendo da programação feita no P00662. Para alarmes, caso a comunicação seja restabelecida, a indicação do alarme será retirada da HMI.

#### **Possíveis Causas/Correção:**

- Verificar instalação da rede, cabo rompido ou falha/mal contato nas conexões com a rede, aterramento.
- Garantir que o mestre envie telegramas para o equipamento sempre em um tempo menor que o programado no P00663.
- Desabilitar esta função no P00663.

# I. APÊNDICES

## APÊNDICE A. TABELA ASCII

**Tabela I.1:** Caracteres ASCII

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