Comunicação Modbus RTU
AC10 e AC310

TECNOLOG

www.tecnolog.com.br

Av. Pernambuco, 2623, | Conj. 101 | Porto Alegre - RS

Telefone: (51) 3076.7800

E-mail: vendas@tecnolog.ind.br

Comunicação Modbus RTU dos inversores AC10 e AC310

Funcionamento:

A comunicação Modbus RTU a partir de um dispositivo mestre como CLP ou IHM permite monitorar e alterar o valor dos parâmetros dos inversores (escravos) AC10 (preto) e AC310 (vermelho). Cada fabricante de ihm ou clp possui o seu próprio driver de comunicação Modbus.

Podemos alterar a frequência do motor, comandar a partida e parada do motor, alterar as rampas de aceleração e desaceleração e monitorar ou alterar todos os demais parâmetros através do acesso ao endereço Modbus de cada parâmetro.

O inversor antes precisa ser preparado para permitir a comunicação serial:

Parametrização do inversor para receber comandos via Modbus RTU:

F01.01: Comando de partida e parada do motor > **2** (RS485) **(F00.02 = 2)**
F01.02: Comando de frequência > **6** (RS485) **(F00.03 = 6)**

Preparação da porta serial do inversor para operação em Modbus RTU:

F12.00: Define se mestre ou escravo (**0:** escravo e **1:** mestre) **(F13.00)**
F12.01: Endereço de rede (1: estação 1) **(F13.01 = 1)**
F12.02: Velocidade da comunicação RS485 (**3**: 9600 bps) **(F13.02 = 3)** ou outra
F12.03: Formatação do *frame* (**0**: sem paridade, 8 data bits e 1 stop bit) **(F13.03 = 0)**

Existem outros parâmetros para a configuração da comunicação, mas na maior parte são utilizados em casos excepcionais e devem ser deixados no modo padrão.

Mapa da memória:

A comunicação Modbus é controlada pelo mestre que executa comandos através da porta serial para leitura ou escrita da memória dos inversores (escravos) através do acesso ao endereço de comunicação de cada parâmetro, onde cada endereço é um **registrador de 16 bits** composto por 2 bytes (alto e baixo) e a notação é com **números inteiros hexadecimais.**

Todos os grupos de parâmetros dos inversores possuem uma faixa de endereços organizada de forma sequencial com 4 dígitos (2 bytes), onde o **byte mais alto** identifica o **grupo** e o mais baixo (xx) a sequência de parâmetros deste grupo, começando pelo **zero.**

No manual dos inversores o endereço Modbus de cada parâmetro é sempre indicado entre parênteses, conforme abaixo, no caso grupo 12, parâmetro 12.01 = **0C01h.**

Code (Addr.)	Name	Content	Factory setting (Setting range)
F12.01 (0x0C01)			

Lista de endereços de comunicação de todos os grupos do inversor AC10:

AC10 Address field high byte definition:

Parameter group code number	*Parameter address of this group*
F00 environment application parameter group	*0x00xx (not stored in EEPROM) 0x10xx (stored in EEPROM)*
F01 basic parameter group	*0x01xx (not stored in EEPROM)0x11xx (stored in EEPROM)*
F02 motor 1 parameter group	*0x02xx (not stored in EEPROM)0x12xx (stored in EEPROM)*
F03 vector control parameter group	*0x03xx (not stored in EEPROM)0x13xx (stored in EEPROM)*
F04 V/F control parameter group	*0x04xx (not stored in EEPROM)0x14xx (stored in EEPROM)*
F05 input terminal parameter group	*0x05xx (not stored in EEPROM)0x15xx (stored in EEPROM)*
F06 output terminal parameter group	*0x06xx (not stored in EEPROM)0x16xx (stored in EEPROM)*
F07 operation control parameter group	*0x07xx (not stored in EEPROM)0x17xx (stored in EEPROM)*
F08 auxiliary control 1 parameter group	*0x08xx (not stored in EEPROM)0x18xx (stored in EEPROM)*
F09 Auxiliary Control 2 Parameter Group	*0x09xx (not stored in EEPROM)0x19xx (stored in EEPROM)*
F10 protection parameter group	*0x0Axx (not stored in EEPROM)0x1Axx (stored in EEPROM)*
F11 keyboard parameter group	*0x0Bxx (not stored in EEPROM)0x1Bxx (stored in EEPROM)*
F12 communication parameter group	*0x0Cxx (not stored in EEPROM)0x1Cxx (stored in EEPROM)*
F13 Process PID Control Parameter Group	*0x0Dxx (not stored in EEPROM)0x1Dxx (stored in EEPROM)*
F14 multi-speed and simple PLC function	*0x0Exx (not stored in EEPROM)0x1Exx (stored in EEPROM)*
C00 basic monitoring parameter group	*0x2100*
C01 fault monitoring parameter group	*0x2200*
C02 application monitoring parameter group	*0x2300*
C03 Maintenance Monitoring Parameter Group	0x2400
MODBUS communication control parameter group	0x30xx or 0x20xx
Input and output interface communication group	0x34xx
Extended fault and power down parameter set	0x36xx

Devemos observar que cada **grupo** possui 2 faixas de endereços: 0x**00**xx e 0X**10**xx assim como também o grupo **MODBUS communication control parameter**: 0x**20**xx e 0x**30**xx**.**
A faixa maior permite acesso à memória EEPROM (não volátil) e a menor à memória RAM (volátil**)** de cada parâmetro. **Precisamos ter cuidado ao escrever na memória EEPROM** para não comprometer a vida útil do inversor pois **existe um limite na quantidade de escritas** suportadas pela memória EEPROM de cada parâmetro.

O grupo **MODBUS communication control parameter** com endereços **x20xx** e **0x30xx reúne os 32 endereços mais importantes** dos inversores que estão distribuídos em outros grupos, facilitando muito a comunicação, pois como são sequenciais podemos acessa-los numa única comunicação (em bloco).

Exemplo: Endereço da frequência (velocidade) do inversor

Address	Name	Read/Write (R/W)	Dimension (range)	Description
0x2000 /0x3000	Given frequency	R/W	0.01Hz (0.00~320.00)	Communication given frequency

Este endereço **0x2000** pode ser lido ou escrito (Read/Write) dependendo do comando utilizado pelo mestre e representa o primeiro endereço (00) deste grupo:

MODBUS communication control parameter group	**0x30xx or 0x20xx**

Além do endereço devemos sempre observar a faixa de valores permitida no parâmetro, no caso do **0x2000** a faixa é de 0.00 a 320.00, que também deve ser informada no bloco de comunicação no **formato hexadecimal**.

No mapa de endereços *Modbus RTU* padrão os endereços são transmitidos no formato **hexadecimal,** e devido a uma peculiaridade da norma, **é necessário somar um ao endereço desejado**, então, para acessar 2000h devemos solicitar **2001h.**

Em alguns drivers de comunicação como exemplo o da Weintek existe o driver **Modbus Zero based** que elimina este problema.

Mensagem (frame) de comunicação MODBUS

A mensagem Modbus é composta por uma sequência de bytes:

Endereço, comando, endereço inicial alto, endereço inicial baixo, quantidade de registros alto e baixo, dados e CRC alto e baixo

Existem comandos Modbus para leitura e escrita de registros (palavras) e de bits, sendo os mais utilizados:

1 (01h) Ler estado de bits
2 (02h) Escrever o estado de bits
3 (03h) Ler múltiplos registradores
6 (06h) Escrever um registrador
16 (10h) Escrever múltiplos registradores

Exemplo: Leitura do setpoint de velocidade do inversor: 2000h (somando +1 = **2001h**)

01h endereço (byte)
03h comando (byte) Ler múltiplos registradores
20h endereço alto
01h endereço baixo
00h número de registros alto
01h número de registros baixo
CRC código de checagem da mensagem (2 bytes)

A resposta do escravo para o mestre será:

01h endereço (byte)
03h comando (byte)
20h endereço alto
01h endereço baixo
XXh valor alto
XXh valor baixo
CRC código de checagem da mensagem (2 bytes)

Atenção: Em muitos clps o driver de comunicação Modbus **adiciona um número** antes do endereço formando novas **famílias de endereços** conforme o formato da variável:

0x = Coil = **0**0001-**0**9999 (bits)
1x = Discrete Input = **1**0001-**1**9999 (bits)
3x = Input Register = **3**0001-**3**9999 (registradores)
4x = Holding Register = **4**0001-**4**9999 (registradores)

Isto é uma **apenas uma convenção** que consiste em **somar** um **endereço base** ao **endereço original** desejado.

Assim para fazermos a leitura do registrador **2000h** deveremos soma-lo ao endereço base **40001h** (40000 mais o 1 devido a falta do zero) para indicar o endereço correto.

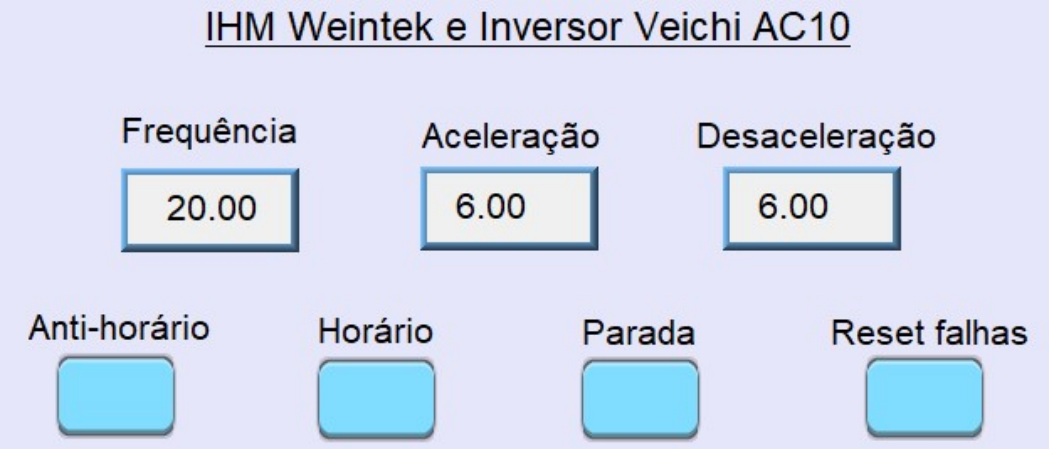
Então temos que 40001h + 2000h = **42001h**

Mapa dos endereços mais comuns dos inversores e seus respectivos valores:

2000h: Frequência do motor (valor xx)
2001h: Controle do motor: Girar anti-horário (1), horário(2), parar (3), reset falha (5)
200Eh: Rampa de aceleração (valor xx) **(14)**
200Fh: Rampa de desaceleração (valor xx) **(15)**

Exemplo de aplicação:

Abaixo temos a tela de uma IHM Weintek com 7 objetos do tipo "*numeric input*" para introduzirmos a frequência do motor, rampas de aceleração e desaceleração e objetos "*Set word*" para fazer os comandos de partida anti-horária, horária e parada.



No software EasyBuilder-Pro vamos selecionar o driver **Modbus zero based** para resolver o problema do endereçamento eliminando a necessidade de somarmos um ao endereço e vamos acessar os parâmetros do grupo **MODBUS communication control parameter group.**

Precisaremos converter os endereços de hexadecimal para decimal para utiliza-los na ihm, então o endereço da frequência que é **2000h**, convertido para decimal fica **8192,** e para escrita no inversor vamos utilizar o comando Modbus 6 **(6x)**.

Address	Name	Read/Write (R/W)	Dimension (range)	Description
0x2000 /0x3000	Given frequency	R/W	0.01Hz (0.00~320.00)	Communication given frequency

O comando de parada e partida do motor será feito na palavra de controle endereço **2001h** (decimal = **8193)**. Serão utilizados 4 objetos "*Set Word"* que vão escrever 4 valores neste endereço: **1** para a partida anti-horária, **2** para a partida horária, **5** para a parada do motor e **7** para reset de falhas, conforme especificado no parâmetro abaixo:

Address	Name	Read/Write (R/W)	Dimension (range)	Description
0x2001 /0x3001	Command given	W	0x0000 (0x0~0x0103)	0x0000: invalid 0x0001: Forward run 0x0002: Reverse run 0x0003: Forward jog 0x0004: Reverse jog 0x0005: Deceleration stop 0x0006: Free stop 0x0007: Reset command 0x0008: Run prohibition command Communication writes to the 3001 address, the inverter is free to stop, and it needs to write 9 to 3001 or re-power on before it can run. 0x0009: Run permission command 0x0101: Equivalent to F2.07 =1 [rotation parameter auto-tuning], plus run command 0x0102: Equivalent to F5.07 = 2 [static parameter auto-tuning], plus run command 0x0103: Equivalent to F5.07 = 3 [stator resistance setting], plus run command

Os endereços das rampas de aceleração (**200E**h) e desaceleração (**200F**h) devem ser convertidos para decimal, resultando **8206** para aceleração e **8207** para desaceleração.

Address	Name	Read/Write (R/W)	Dimension (range)	Description
0x200E /0x300E	Acceleration time 1	R/W	0.00s (0.00~600.00s)	Read and write the value of F01.22
0x200F /0x300F	Deceleration time 1	R/W	0.00s (0.00~600.00s)	Read and write the value of F01.23

<u>Mapeamento dos endereços dos 7 objetos da IHM e seus respectivos valores:</u>

8192 (2000h): Frequência do motor (valor xx)
8206 (200Eh): Rampa de aceleração (valor xx) **(14)**
8207 (200Fh): Rampa de desaceleração (valor xx) **(15)**
8193 (2001h): Partida anti-horária (valor 1)
8193 (2001h): Partida horária (valor 2)
8193 (2001h): Parada (valor 5)
8193 (2001h): Reset falhas (valor 7)

<u>Inversor como mestre (host) da rede Modbus RTU (opcional):</u>

Existe a possibilidade de parametrizar um inversor (mestre) para operar em *Broadcast* com outros inversores na mesma rede, neste caso o mestre envia a sua frequência atual e comando de partida (**F12.10 = 0031**) para que os escravos mantenham sincronismo de partida/parada e velocidade com ele.

Parametrização dos inversores:

Mestre: **F12.00: 1** (mestre) ou **(F13.00 = 1)**

Escravos: **F01.01 = 2** (comando de partida e parada via RS485) **(F00.02 = 2)**
F01.02 = 6 (comando de frequência via RS485) **(F00.03 = 6)**
F12.01 = 0 (estação 0) **(F13.01 = 0)** escravo em *Broadcast*

TECNOLOG