

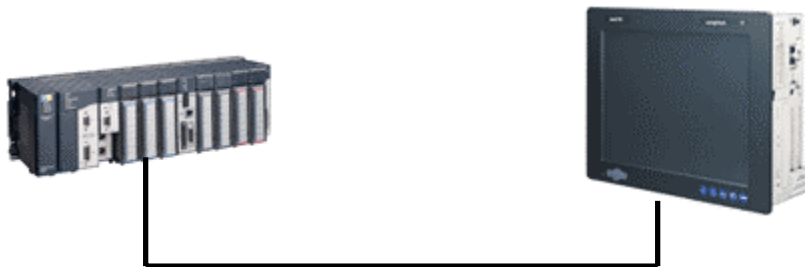
Descritivo de configuração Modbus Master para PAC

1. Equipamentos utilizados:

- IC695CHS
- IC695CMU
- Software MOD_RSsim v. 4.1 (www.adroit.co.za)
- HE693SNPCBL
- IC200CBL001

2. Descrição do sistema

O sistema montado para execução dos testes consiste de 1 conjunto PAC RX3i, o programador Cimplicity Machine Edition, o pacote de comunicação Modbus em linguagem C (disponível em <http://globalcare.gefanuc.com>) e o software de simulação de Modbus MOD_RSIM. O PC executará a simulação do dispositivo slave na rede Modbus.



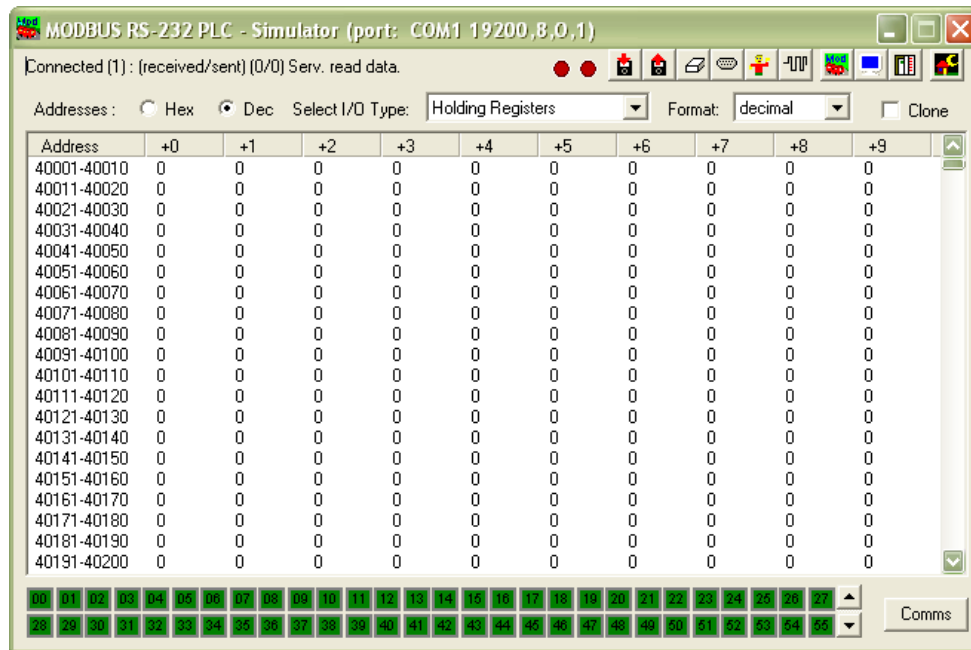
3. MOD_RSIM:

O software MOD_Rssim simula até 64 dispositivos slave em uma rede Modbus virtual, sendo que o controlador poderá solicitar informações a mais de um dispositivo ao mesmo tempo.

Nesse exemplo utilizaremos a configuração serial padrão para os produtos GE Fanuc:

19200 baud
8 data bits
1 stop bit
Odd parity

O software MOD_Rssim pode ser baixado diretamente no site do fabricante, em www.adroit.co.za. Para suporte sobre o software, favor consultar o fabricante do mesmo.



Software MOD_RSSIM

4. Modbus Master para PAC:

O controlador PAC possui duas portas seriais na própria CPU. Uma delas RS232 e a outra RS485. Com o uso de um bloco em "C" qualquer uma dessas portas podem ser usadas para comunicação Modbus, atuando como master na rede.

Além do bloco C é necessária programação adicional em Ladder para configurar a comunicação e também para estabelecer os canais de comunicação entre master e slave.

As seguintes configurações foram testadas e aprovadas:

- RS232 ponto a ponto;
- RS485 2 fios, 4 fios, ponto a ponto e multi-drop;
- Conexão por modem usando o modem SixNet VT.

As seguinte velocidades de comunicação foram testadas e aprovadas:

- 19200
- 9600
- 4800
- 2400
- 1200

As seguintes configurações NÃO foram testadas:

- Radio Modem
- Modem via celular

As seguintes funções NÃO foram implementadas:

- Função 65 – Read Scratchpad
- Funções de leitura/escrita de registros de 32 bits

Configuração de hardware:

Toda a programação do PLC será feita com o Proficy Machine Edition LD-PLC. A porta da CPU que será usada para comunicação Modbus Master deve ser configurada como SERIAL I/O, com os parâmetros de porta setados corretamente.

Parameter	Required Setting	Choices for setting
Port Mode	Serial I/O	
Data Rate**		1200, 2400, 4800, 9600, 19200
Data Bits	8	
Flow Control		None, Hardware - recommend using "None"
Parity		None, Odd, Even - slave/modems must match
Stop Bits		1, 2
Physical i/f		2 wire, 4 wire
Stop mode.....		Any setting

Nota - Se as duas portas forem utilizadas para comunicação Modbus, o switch run/stop DEVE ser habilitado

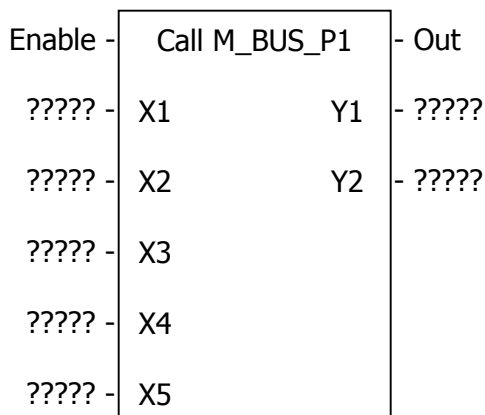
Bloco de comunicação "C":

Está disponibilizado no site <http://globalcare.gefanuc.com> um bloco pronto para comunicação em modbus. Se as duas portas forem utilizadas para essa função, então 2 blocos C deverão ser adicionados ao programa.

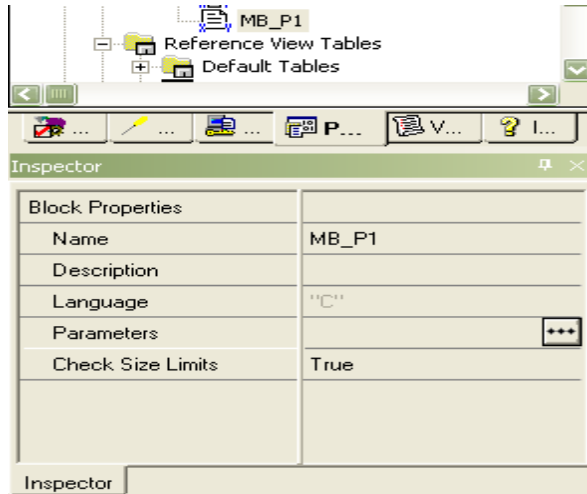
No site encontram-se 2 arquivos, um para cada porta, nomeados MP_P1.gefElf e MB_P2.gefElf. Os dois arquivos são idênticos e funcionam tanto com o RX3i como com o RX7i.

Para que esses blocos funcionem corretamente é necessário que as CPUs tenham firmware com versão 2.50 ou superior.

O bloco C tem o seguinte formato:

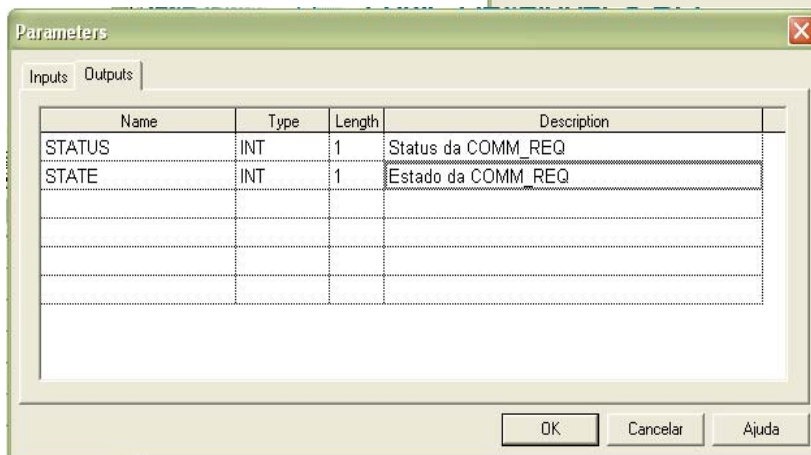


Os parâmetros do bloco devem ser configurados como segue:



Selecione o bloco MB_P1 que foi adicionado ao programa e clique no botão "...” do item *Parameters*.

Coloque os dados dos parâmetros de entrada conforme indicado na figura ao lado. Atenção para o comprimento do parâmetro CMD (8 words). Todos os itens devem ser do tipo INT.



Faça o mesmo para os parâmetros de saída do bloco. Nesse caso as saídas ocupam 1 word cada, e devem ser do tipo INT.

Explicação do bloco MB_P1

Enable:

O bloco será executado se a entrada *Enable* receber sinal lógico 1.
O bloco C deverá ser acionado a cada scan do PLC.

Parâmetros X1 (CMD):

Formato: INT

Length: 8

Função: Parâmetros da comunicação Modbus

- 1ª. Word: Comando de parametrização do protocolo
Valores válidos: 8002 apenas
O bloco C retorna 0 nessa Word a cada comando executado. O valor não deve ser carregado novamente até que a saída Y1 retorne um valor maior que 0. Se o valor 8002 for escrito novamente antes do bloco retornar o valor maior que 0 então o novo comando será ignorado até que seja detectado um valor maior que 0, só então o comando será executado.
- 2ª. Word Endereço do slave
- 3ª. Word Código de função do Modbus (ver tabela 1 a seguir)
- 4ª. Word Dependente da Word 3
- 5ª. Word Dependente da Word 3
- 6ª. Word Tipo do endereço local (RX3i) (veja tabela 2)
- 7ª. Word Endereço local inicial
- 8ª. Word Time out em milissegundos – 0 – 10000 (10s)

Parâmetros X2 (R_S)

Formato: INT

Length: 1

Função: Localização da CPU no rack e slot do sistema.

Valores válidos: 1-14 para RX3, 1 para RX7.

Parâmetros X3 (PORT)

Formato: INT

Length: 1

Função: Número da porta de comunicação

Valores válidos: 1 ou 2. 19 ou 20 também são aceitos.

Parâmetros X4 (TYPE_BUF)

Formato: INT

Length: 1

Função: Tipo do registrador utilizado para o buffer

Valores válidos: 8 (%R), 196 (%W)

Parâmetros X5 (BUF)

Formato: INT

Length: 1

Função: Endereço inicial do pacote de buffer. A função Modbus Master requer 150 registros para correto funcionamento.

Valores válidos: Qualquer valor dentro do range de registradores do PLC.

Parâmetros Y1 (STATUS)

Formato: INT

Length: 1

Função: Status da comunicação

Valores válidos: qualquer registrador (%Rxxxx, %Wxxxx)

Parâmetros Y2 (STATE)

Formato: INT

Length: 1

Função: Estado da Comm_req (idle, em processo, timeout)

Valores válidos: qualquer registrador (%Rxxxx, %Wxxxx)

Nota: não usar constantes para os parâmetros X1, Y1 e Y2. X1 é um ponteiro de um Array. Y1 e Y2 devem ser registradores do PLC de modo que o bloco possa atualizar seus valores com as informações da comunicação.

Tabela 1 – Códigos de função do Modbus

Function Code 1 – Read Outputs

Word 4 – Start address in Slave for Read

Word 5 – Numbers of Output points to read (bits); max = 2000

Word 6 – Reference Table in this master to put the values that are read. Valid Reference tables are: I, Q, T, M, G, AI, AQ, R, W - VER TABELA 2

Word 7 – Location in Reference table to put read values.

Function Code 2 – Read Inputs

Word 4 – Start address in Slave for Read

Word 5 – Numbers of Input points to read (bits); max = 2000

Word 6 – Reference Table in this master to put the values that are read. Valid Reference tables are: I, Q, T, M, G, AI, AQ, R, W - VER TABELA 2

Word 7 – Location in Reference table to put read values.

Function Code 3 – Read Holding Registers

Word 4 – Start address in Slave for Read

Word 5 – Numbers of Registers to read (16 bit words); max = 125

Word 6 – Reference Table in this master to put the values that are read. Valid Reference tables are: AI, AQ, R, W - VER TABELA 2

Word 7 – Location in Reference table to put read values.

- Function Code 4 – Read Input Registers
Word 4 – Start address in Slave for Read
Word 5 – Numbers of Registers to read (16 bit words); max = 125
Word 6 – Reference Table in this master to put the values that are read. Valid Reference tables are: AI, AQ, R, W - VER TABELA 2
Word 7 – Location in Reference table to put read values.
- Function Code 5 – Set/Clear One Coil *Broadcast allowed*
Word 4 – Coil Number in Slave
Word 5 – 1 = set coil, 0 = clear coil
Word 6 – not used
Word 7 – not used
- Function Code 6 – Write One Register *Broadcast allowed*
Word 4 – Register Number in Slave
Word 5 – Value to write to Register
Word 6 – not used
Word 7 – not used
- Function Code 7 – Read Exception Status
Word 4 – not used
Word 5 – not used
Word 6 – Reference Table in this master to put the returned exception status Valid Reference tables are: I, Q, T, M, G, AI, AQ, R, W - VER TABELA 2
Word 7 – Location in Reference table to put read values.
- Function Code 8, Subfunction 0 – Loopback
Word 4 – 0 (must be zero)
Word 5 – data pattern which is used in loopback message (any value)
Word 6 – not used
Word 7 – not used
- Function Code 8, Subfunction 1 – Restart Communication *Broadcast allowed*
Word 4 – 1 (must be one)
Word 5 – 0 (must be zero)
Word 6 – not used
Word 7 – not used
- Function Code 8, Subfunction 4 – Enter Listen Only Mode *Broadcast allowed*
Word 4 – 4 (must be four)
Word 5 – 0 (must be zero)
Word 6 – not used
Word 7 – not used
- Function Code 15 – Write Multiple Coils *Broadcast allowed*
Word 4 – Start address in Slave for Write
Word 5 – Numbers of Output points to write (bits); max = 19200
Word 6 – Reference Table in this master to get the values that are written. Valid Reference tables are: Q, R, W
Word 7 – Location in Reference table to get write values.

Function Code 16 – Write Multiple Registers *Broadcast allowed*

Word 4 – Start address in Slave for Write

Word 5 – Numbers of Registers to write (16 bit words); max = 120

Word 6 – Reference Table in this master to get the values that are written. Valid Reference tables are: AI, AQ, R, W - VER TABELA 2

Word 7 – Location in Reference table to get write values.

Function Code 17 – Report Device Type

Word 4 – not used

Word 5 – not used

Word 6 – Reference table in this master to put Device Type info.

Valid Reference tables are : AI, AQ, R, W - VER TABELA 2

Word 7 – Location in Reference table to put Device Type info

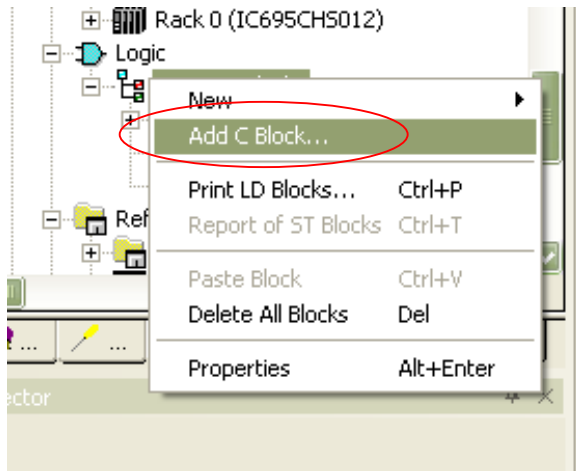
Tabela 2 – Seleção do tipo de registrador

Tipo	Valor equivalente
AI	10
AQ	12
R	8
W	196
I	16
Q	18
T	20
M	22
G	56

Programação do PLC:

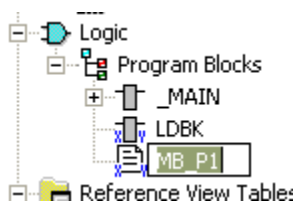
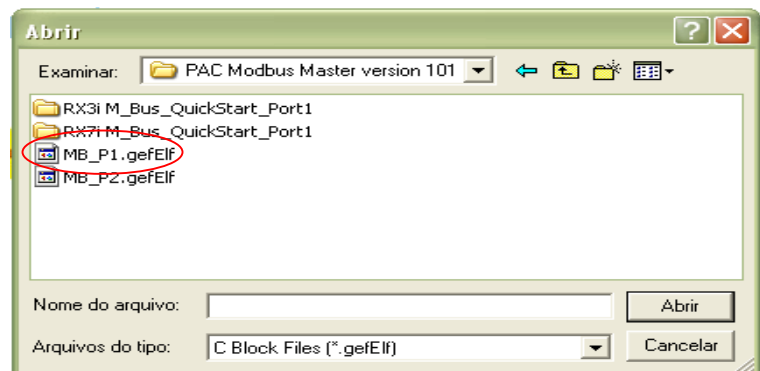
Os próximos passos demonstrarão como será feita a programação do PLC no Proficy Machine Edition.

Adicionando o bloco C:



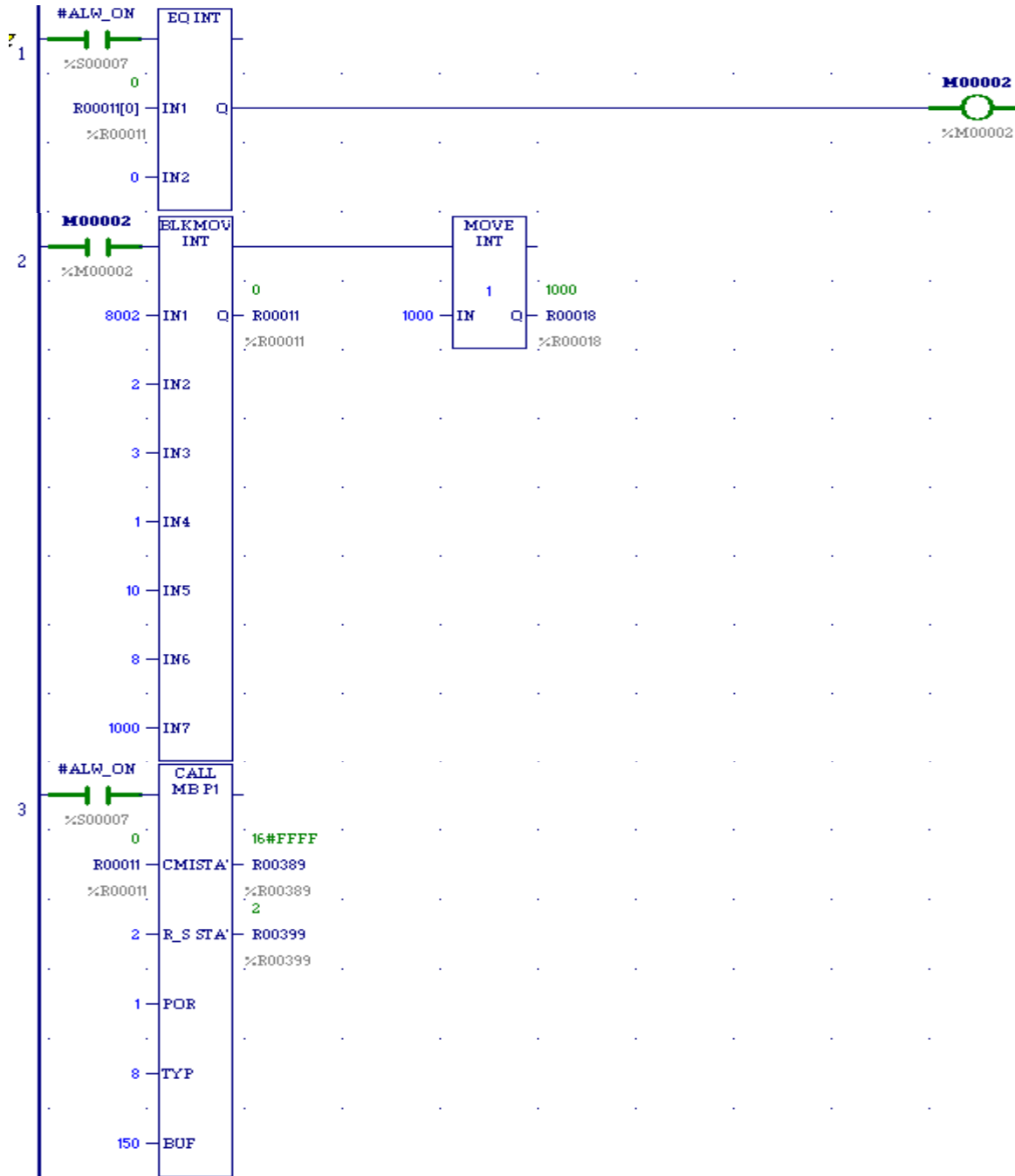
Para adicionar um novo bloco ao programa clique com o botão direito do mouse sobre o ícone *Program Blocks* e selecione a opção *Add C Block*.

O Machine Edition irá abrir uma caixa de diálogo para seleção do arquivo que será inserido no programa. Localize no seu computador o arquivo MB_P1.gefElf e clique em *Abrir*.



O bloco será inserido na lista de blocos de programa.

Com o bloco inserido é necessário desenvolver uma pequena lógica para controlar a comunicação com os slaves Modbus, como segue:



Como o programa funciona?

O programa trabalha de forma semelhante a uma comunicação por COMM_REQ. Na verdade, o bloco C executa COMM_REQs no formato serial I/O para fazer a comunicação Modbus.

Para executar um comando Modbus deve-se seguir a seqüência:

- Use um contato de transição para carregar os valores para o bloco de comando com 8 words
 - A primeira Word do bloco é o código de comando 8002 (sempre)
 - A segunda Word é o endereço do escravo Modbus na rede
 - Na terceira, quarta e quinta Words estão os códigos referentes à função do Modbus
 - A sexta Word representa o tipo de memória que será usada como buffer (8 para %R)
 - A sétima Word indica o início do bloco de buffer
 - A oitava Word é o time-out da comunicação.
- A entrada X1 deve ser a mesma do início do bloco (%R0011). Essa Word deve ser um array com 8 elementos.
- Quando o bloco C enxerga o valor 8002 na primeira Word do bloco, o comando Modbus é executado, baseado nos valores carregados no bloco.
- O bloco C verifica e valida os valores carregados e:
 - Escreve o valor 0 na primeira Word do bloco (%R0011)
 - Escreve o valor 2 na saída Y2, caso o bloco seja válido e a comunicação seja iniciada. Se houver algum erro no bloco ou no setup da porta serial um erro é escrito na saída Y1 e o valor 0 é escrito na primeira Word do bloco.
- Se o bloco de comando estiver correto, o comando modbus será enviado e o bloco C retornará a indicação de sucesso (1) ou o código do erro na saída Y1 quando a comunicação terminar ou se for detectado um time-out.
- Nesse momento, se o valor 8002 estiver carregado na primeira Word do bloco de comando, um novo ciclo de comunicação será iniciado.
- Se as duas portas forem utilizadas para comunicação, então serão necessários dois blocos C, um para cada porta, garantindo a independência de funcionamento.

Códigos de erro

A saída **Y1** do bloco C apresenta um código de status da comunicação Modbus. Este é o valor que deve ser monitorado para determinar quando uma comunicação foi finalizada e se ela foi ou não bem sucedida.

Os códigos de erro estão divididos entre o byte mais significativo (major) e o menos significativo (minor), sendo assim, é recomendado que seja monitorado no formato HEX.

Major	Minor	Descrição
0	0	COMM_REQ em andamento
0	1	Sucesso
1	1	Time-out
5	3	Numero da porta inválido – deve ser 1 ou 2 (19 ou 20 também funcionam)
6	3	Endereço do slave inválido – deve ser entre 0 a 247
7	3	Função Modbus inválida – deve ser 1,2,3,4,5,6,7,8,15,16,17
8	3	Função Broadcast inválida – Somente funções 5,6,8,sub4,15 e 16 suportam broadcast
A	3	Endereço inicial igual a Zero – o endereço inicial (word3) deve ser maior que zero
B	3	Muitos itens – o número de itens deve ser > 0
C	3	Tipo de registrador inválido – veja tabela de códigos para os valores suportados
21	3	Tipo de registrador inválido – veja tabela de códigos para os valores suportados
22	3	End. inicial inválido – o end. inicial + no. de itens deve ser < que o total de endereços
23	3	Valor da saída Set/Clear inválido – deve ser 0=clear, 1=set ou FF=Set
15	3	Configuração de porta inválida – a porta serial deve ser configurada para SERIAL I/O
16	3	Rack/Slot inválidos – A CPU não está no slot declarado no bloco de comando
17	3	Comando commreq inválido – A Word de comando 1 deve ser 8002
19	3	Código de comando inválido – O código de comando (Word 1) deve ser 8002
1	4	Erro de paridade
2	4	Erro de frame
3	4	Erro de CRC
5	4	Overflow
7	4	Erro da UART
1	5	Time-out na recepção da comunicação
2	5	Time-out na transmissão – checar sinal CTS
10	6	Tipo de endereço inválido – a Word de comando 4 deve ser 8 (%R) ou 196 (%W)
11	6	Endereço inicial muito alto – a Word 5 está muito alta. O bloco precisa de 150 words
X	8	Comando de exceção recebido. O x indica o número da exceção



A saída **Y2** do bloco C apresenta o código de estado da comunicação Modbus:

- 0 – idle
- 2 – Aguardando resposta do slave
- 3 – Time-out após envio de comando broadcast

Rodrigo Ramos

OnTime Automação e Proteção
Fone (11) 2178-1715
rodrigo.ramos@ontimeautomacao.com