



Direto ao **Ponto**

**Comunicação Modbus RTU
mestre utilizando-se o módulo
C24 IQ-R**
Rev. A





Revisões

Data da Revisão	Nome do Arquivo	Revisão
Junh0/2018 (A)	DAP-PLCIQR- 08(A)_ModbusRTU_C24	Primeira edição



1. OBJETIVO

O objetivo deste documento é fornecer orientação básica de como realizar a comunicação Modbus RTU utilizando o Módulo RJ71C24 do CLP IQ-R como mestre da rede.

2. HARDWARE

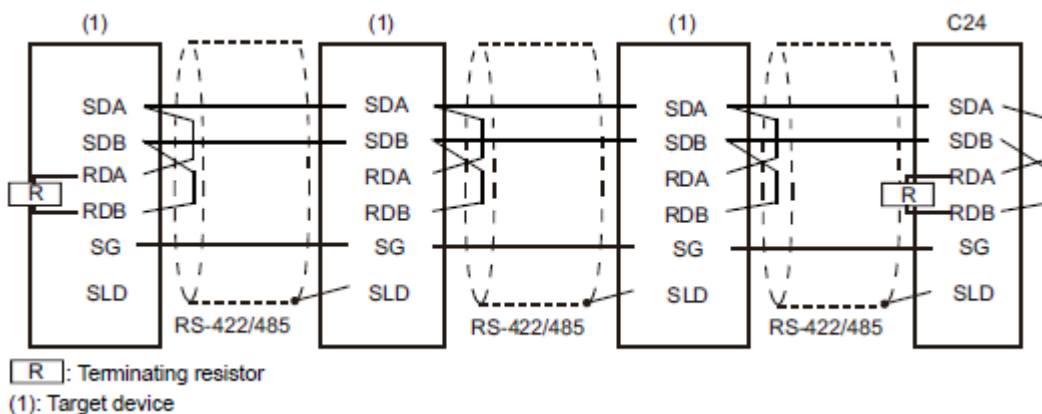
1 CLP IQ-R com módulo RJ71C24 instalado.

1 Computador com o Software GX Works 3.

3. PREPARAÇÃO

3.1. Itens a preparar

Fazer a conexão física do equipamento escravo ao módulo RJ71C24 via padrão RS-422/485 conforme imagem abaixo.

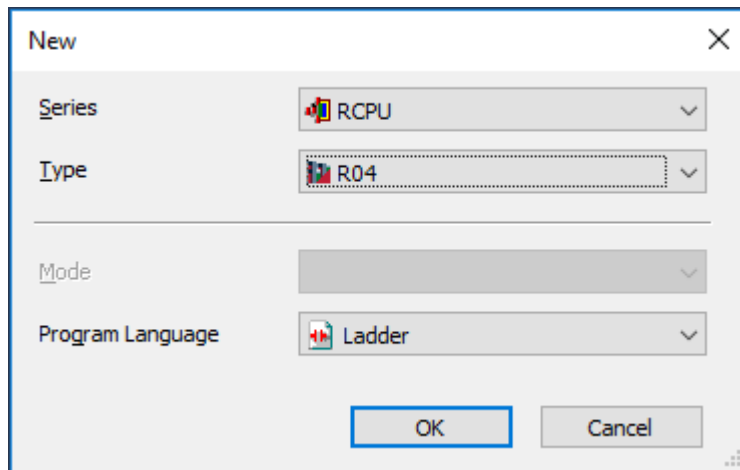


3.2. Configurações Básicas no CLP/Módulo Serial

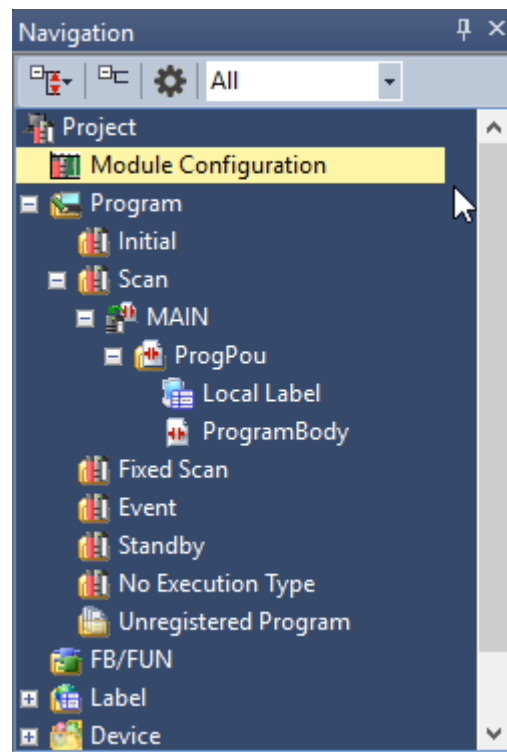
Primeiramente, deve-se criar um novo projeto no GX Works 3 e adicionar o hardware a configuração.



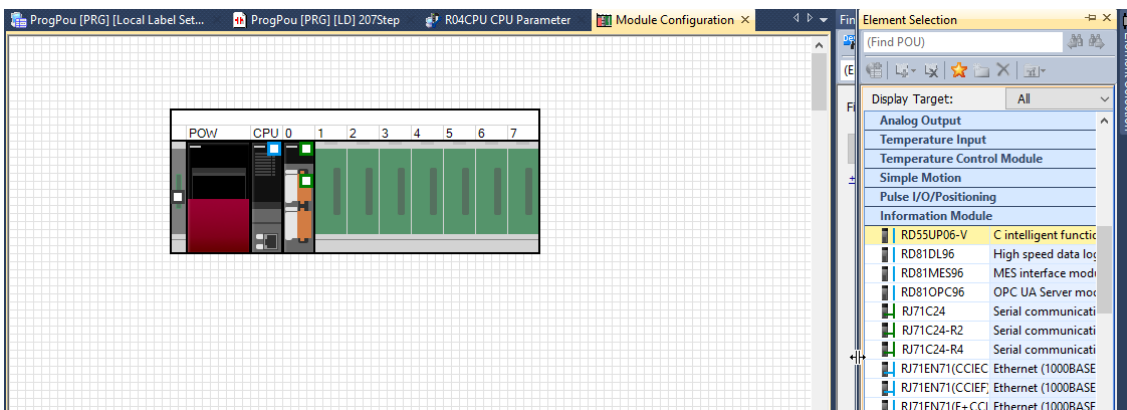
Criar projeto escolhendo família IQ-R (RCPU) e selecionar a CPU que está sendo usada:



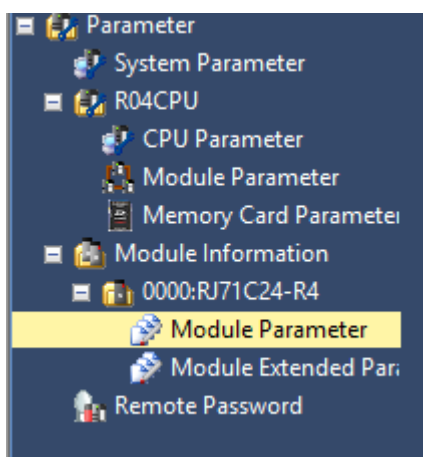
Acessar o menu “Module Configuration” que fica na árvore do projeto a esquerda, para adicionar o hardware do CLP.



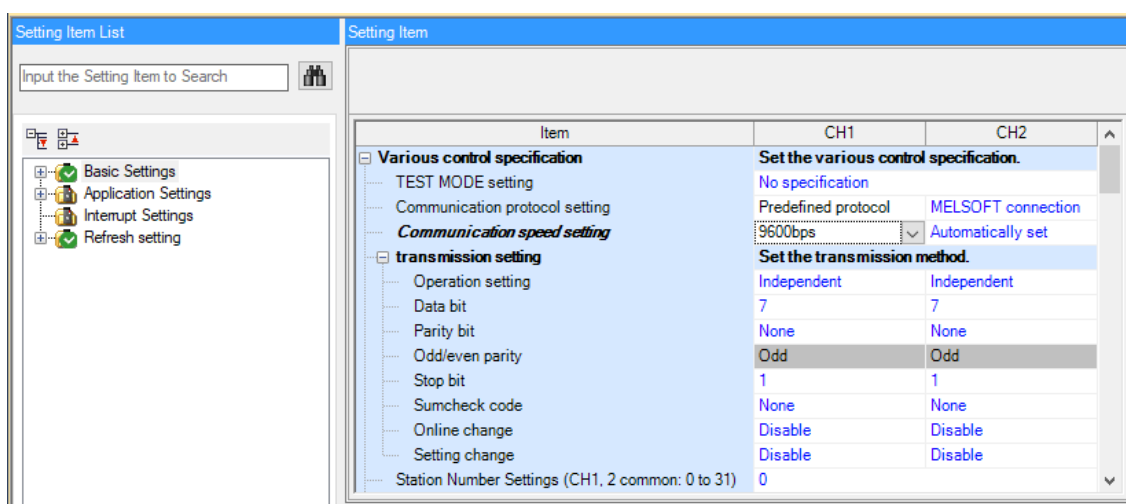
Basta selecionar CPU, rack, fonte e módulos que serão usados na lista a direita, e arrastá-los para a tela, ao final da configuração clicar com botão direito na tela: [Parameter] > [Fix(s)]



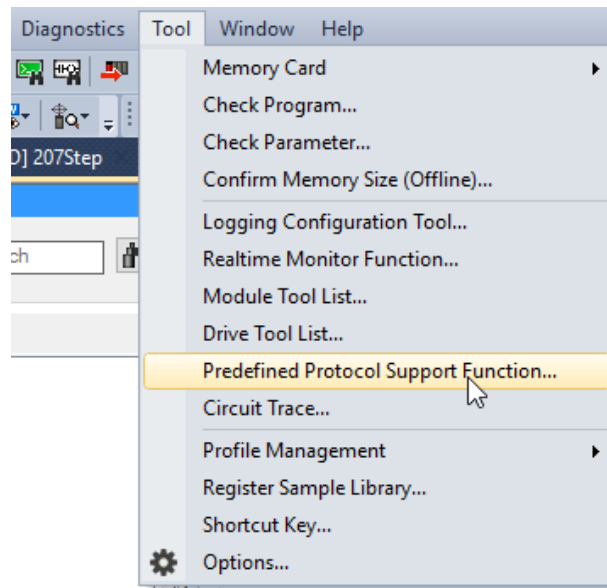
Após adicionado o hardware, na árvore do projeto clicar em “Module Parameter” do módulo serial para configurá-lo:



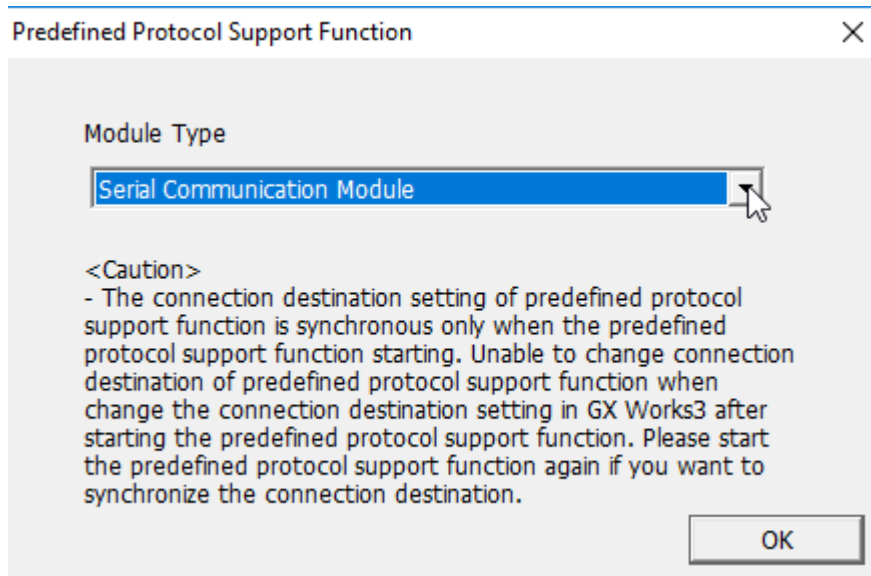
Alterar no canal desejado, a opção de “Communication protocol setting” para “Predefined protocol” e alterar velocidade da transmissão para ficar igual da rede.



Após configurar o módulo serial é necessário configurar o protocolo Modbus RTU na ferramenta “Predefined Protocol Support Function”, para isso vá em “Tool” e selecione a ferramenta.

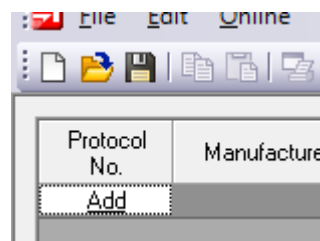


Para Modbus RTU vamos selecionar a opção de Serial communication Module

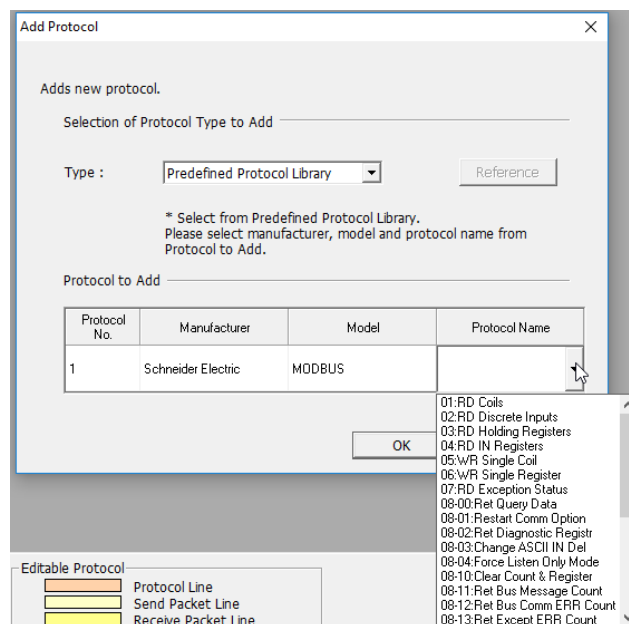


Criando um novo protocolo

Crie um novo arquivo em File -> New, e depois clique em Add



Escolher na coluna de fabricante a opção Schneider Electric, em Model a opção MODBUS, e em protocol name é onde escolheremos o tipo de operação que queremos, de leitura ou escrita, se serão bobinas ou registradores e assim por diante.

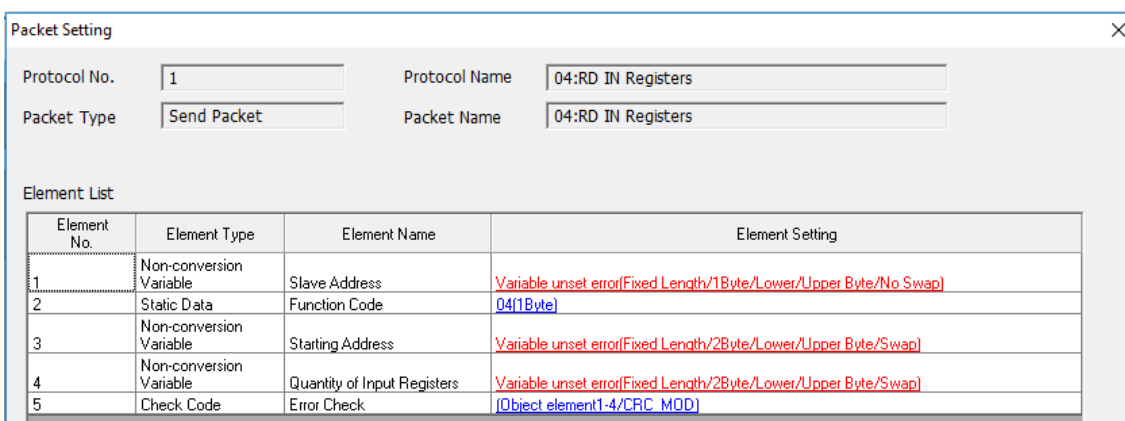


Para nosso exemplo vamos escolher a função Modbus 04 : RD IN Registers

Protocol No.	Manufacturer	Model	Protocol Name	Communication Type	-> Send		Packet Name	Packet Setting
					<- Receive			
1	Schneider Electric	MODBUS	04:RD IN Registers	Send&Receive	>	<	04:RD IN Registers	Variable Unset
					<- (1)	<- (2)	NOR/04:RD IN Registers	Variable Unset
							ERR/04:RD IN Registers	Variable Unset
2	Schneider Electric	MODBUS	06:WR Single Register	Send&Receive	>	<	06:WR Single Register	Variable Unset
					<- (1)	<- (2)	NOR/06:WR Single Register	Variable Unset
							ERR/06:WR Single Register	Variable Unset

É preciso configurar as áreas de memórias que usaremos para alocar os dados necessários nas trocas da comunicação Modbus RTU, cada um dos itens em vermelho indica que falta configurar os registros, abaixo cada um deles:

O primeiro é onde configuramos os registros que serão usados para configurar a mensagem que será enviada ao escravo.

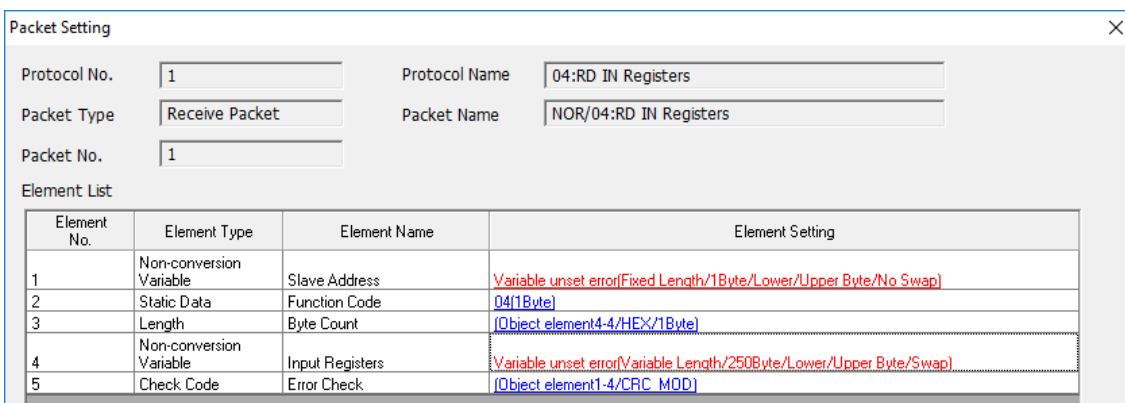


Slave Address: Configurar o registro onde será armazenado o ID do escravo na rede

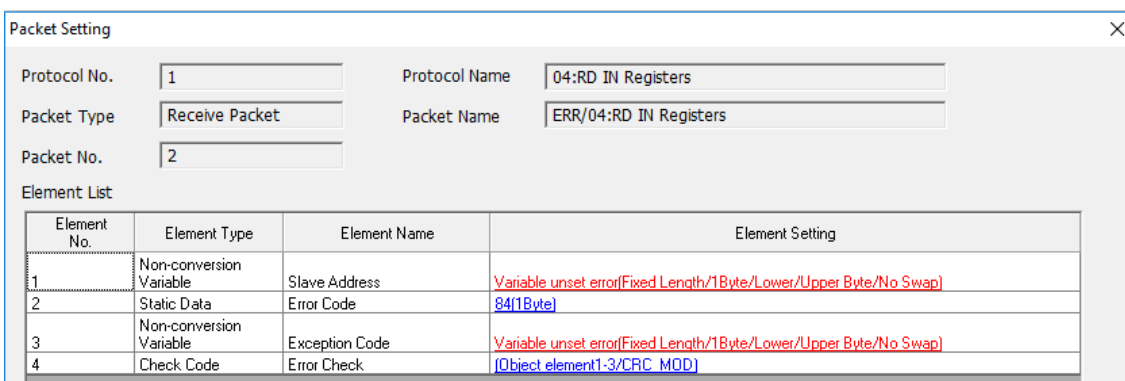
Starting Address: Registro onde será armazenado o endereço Modbus, ou o primeiro endereço para o caso de a leitura ou escrita for de vários endereços sequenciais.



Quantity of input Registers: Quantidade de endereços seguidos que serão lidos ou escritos.

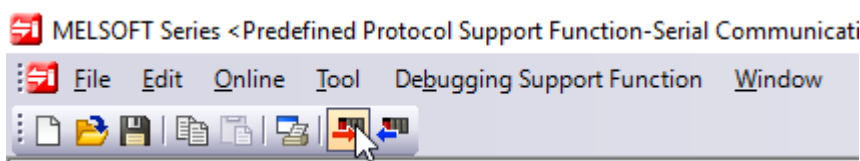


Input Registers: Registro onde começara a ser armazenado os dados que serão lidos ou escritos, o primeiro registro configurado receberá a quantidade de bytes que estão sendo trocados, e a partir do próximo é que os dados são armazenados.



Exception code: Registro onde ficará armazenado o código de erro caso aconteça algum na execução do protocolo.

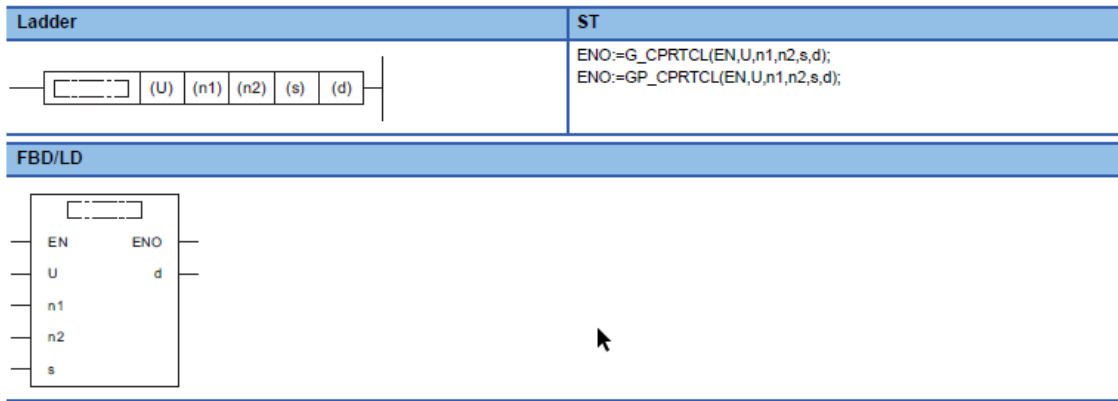
Após configurar o protocolo, é preciso descarregar o mesmo para o IQ-R, colocando o mesmo em STOP e descarregando o arquivo que ficará salvo na Flash ROM.



3.3 Lógica Ladder

Usaremos a instrução GP.CPRTCL para executar o protocolo Modbus RTU que foi configurado.

G(P).CPRTCL



Execução

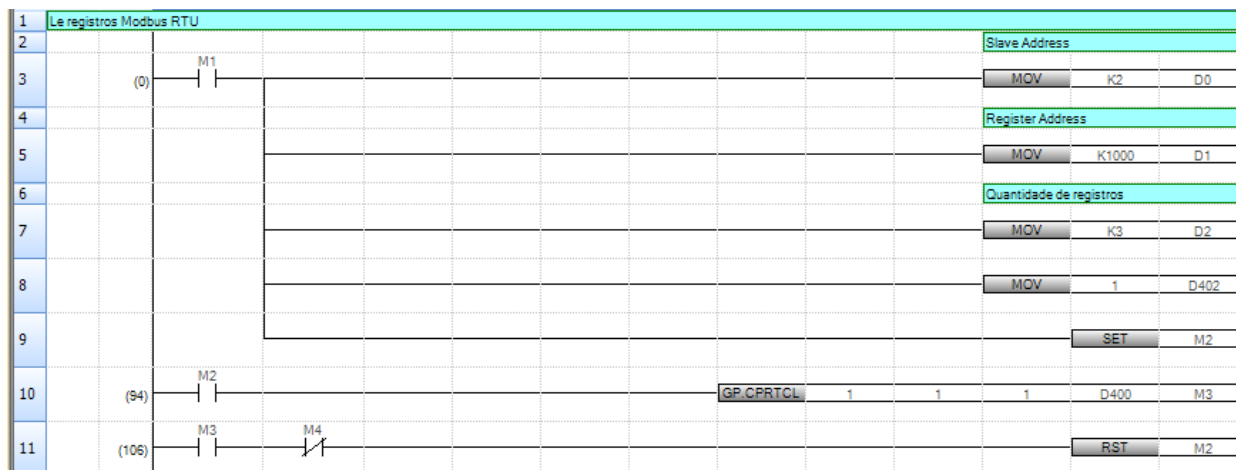
Instruction	Execution condition
G_CPRTCL	
GP_CPRTCL	

- U:** Posição do módulo no rack em Hexadecimal
- N1:** Canal que será usado para comunicação
- N2:** Número de protocolos a serem executados em sequência
- S:** Registro onde foi armazenado o começo da palavras de controle
- D:** Bit que ficará ligado por 1 scan após completar a instrução, e d+1 o bit que ficará ligado em caso de erro na execução

Palavra de controle (s)

s + 0	Resultado da execução	0 : Normal , outro valor : código do erro
s + 1	Protocolos sendo executados	Quantidade de protocolos sendo executados
s + 2	Número do protocolo a ser executado	Número do primeiro protocolo a ser executado
...		...
s + 9		Número do oitavo protocolo a ser executado
s + 10	Pacote de confirmação de comunicação	Guarda o número de confirmação que o primeiro protocolo foi executado corretamente, se o valor for 0 um erro ocorreu no respectivo protocolo
...		...
s + 17		Guarda o número de confirmação que o oitavo protocolo foi executado corretamente, se o valor for 0 um erro ocorreu no respectivo protocolo

No exemplo a seguir foi feita uma rotina para ler os Input Registers, usando o protocolo que foi configurado anteriormente.



M1 foi usado para mover os valores de ID do escravo, endereço onde começara a leitura e quantidade de registros a serem lidos e número do protocolo a ser executado respectivamente em D0, D1, D2 e D402. M1 também setou M2 que será usado para executar a função GP.CPRTCL.

Na instrução GP.CPRTCL foi configurada com o módulo serial na posição 1 após a CPU, canal 1, para executar apenas 1 protocolo (é possível até 8 de uma vez), D400 é o registro onde começa as palavras de controle S, e na posição S+2 ficará guardado o número do protocolo (linha) que será executado, portanto em D402 moveremos o valor 1. Poderia ser usado um array de 18 posições, onde cada posição seria uma palavra de S.

M3 ficará em ON por 1 scan caso a instrução seja completada, e M4 ficará em 0 caso não ocorra erro, e com isso a instrução RST será executada, para evitar que M2 fique ligado e fique executando a instrução a todo momento, o que pode gerar um erro.

Como são 3 registros sendo lidos, D3 receberá o valor 6 (6 bytes, 2 para cada registro), e D4, D5 e D5 receberão o valor dos IN registers lidos.