

MODBUS RTU Master

S7-1200 / S7-1500 / ET200SP CPU

Denilson Pegaia

Objetivo: Prover uma visão geral de como configurar comunicação MODBUS RTU Master usando os controladores SIMATIC S7-1200 / S7-1500 / ET200 SP CPU. O documento é focado em comunicação via RS485, mas os conceitos gerais podem ser expandidos para RS 232.

Aviso: Este documento apresenta dicas e exemplos sobre o produto e supõe que o leitor possua conhecimento básico prévio sobre o mesmo. Para informações completas e atualizadas, deve-se consultar o manual do produto. O intuito deste texto é meramente didático, podendo ser alterado pela Siemens sem aviso prévio. Os exemplos devem ser adaptados ao uso final e exaustivamente testados antes de utilizados em projetos reais.

Índice

1. Introdução - Conceitos	2
Endereçamento de variáveis	2
2. Opções de hardware	3
3. Informações necessárias	4
Ligação elétrica – RS 485	5
Princípio geral.....	5
S7-1200 CB 1241 (RS485) 6ES7 241-1CH30-1XB0.....	6
S7-1200 CM 1241 (RS422/RS485) 6ES7 241-1CH32-0XB0	7
S7-1500 / ET200MP CM PtP RS 422/485 HF 6ES7541 1AB00-0AB0.....	7
ET200SP 6ES7 137 6AA00-0BA0	8
4. Configuração e Programação	9
Estrutura Geral	9
Configuração de Hardware.....	9
Modbus_Comm_Load	15
Modbus_Master.....	19
5. Técnicas de diagnóstico.....	21
6. Dicas para operar com vários escravos	25
7. Dicas gerais de configuração	30
8. Referências, exemplos e downloads.....	31
Geral.....	31
S7-1200 Rack Central	31
S7-1500, ET200SP e ET200MP	31
9. Suporte técnico e treinamento	32

1. Introdução - Conceitos

MODBUS é um protocolo de comunicação usado em sistemas de automação industrial.

Existem algumas variações de MODBUS que dizem respeito ao meio físico e a formatação do telegrama. As formas mais comuns de MODBUS encontradas são MODBUS RTU e MODBUS TCP.

Tipo de MODBUS	Meio de transmissão
MODBUS TCP	Ethernet
MODBUS RTU	Serial (em geral RS 485, podendo ser também em RS 422 ou RS 232)

No MODBUS, os participantes da comunicação podem adquirir dois tipos de papéis: os que são responsáveis por solicitar valores / mandar ordens e os que são responsáveis por responder as solicitações / acatar as ordens. As terminologias usadas para estes papéis são as seguintes:

Tipo de MODBUS	Requisita / dá ordem	Responde / acata ordens
MODBUS TCP	Cliente (<i>Client</i>)	Servidor (<i>Server</i>)
MODBUS RTU	Mestre (<i>Master</i>)	Escravo (<i>Slave</i>)

Nota: A identificação do tipo e do papel que exerce é fundamental para configuração da comunicação.

Endereçamento de variáveis

O MODBUS pode trabalhar com diferentes tipos de variáveis. Estes tipos de variáveis são denominados coils (sinais binários) e registros (palavras de 16 bits).

O padrão MODBUS não especifica a forma como as variáveis internas dos equipamentos são mapeadas (fica a cargo de cada fabricante). Por isso, a forma como os equipamentos apresentam o endereçamento das variáveis internas pode variar, tanto no conceito como nos valores:

- Alguns fabricantes começam a contar suas variáveis a partir da posição zero, enquanto outros contam a partir da posição um.
- Alguns fabricantes “quebram”, o endereçamento das variáveis em dois componentes: o “*Function code*” (que representa o tipo de acesso, leitura/escrita coil/registo) e a posição (endereço) da variável. Outros fabricantes usam um endereçamento que é, em si, uma combinação de “*Functions code*” e posição do registo.

Por exemplo, para a leitura da primeira variável de 16 bits, os fabricantes podem apresentar as nomenclaturas:

- “*Function code*” 3, registro 0 (as variáveis de 16 bits começa a ser contadas a partir da posição 0).
- “*Function code*” 3, registro 1 (as variáveis de 16 bits começa a ser contadas a partir da posição 1).
- Endereço 40.000 (40.000 representa o “*Function code*” 3 + 0 representa primeiro registro = 40.000).
- Endereço 40.001 (40.000 representa o “*Function code*” 3 + 1 representa primeiro registro = 40.001).

2. Opções de hardware

Este documento se restringe à comunicação MODBUS RTU exercida por controladores S7-1200 / S7-1500 / ET200 SP CPU usando módulos de comunicação das mesmas famílias conforme tabela abaixo:

MODBUS RTU STEP 7 V1X	Rack Central	ET200SP	ET200MP
S7-1200	CM1241 CB1241	CM PtP	CM PTP (HF)
S7-1500	CM PTP (HF)		
ET200SP CPU	CM PtP	CM PtP	CM PTP (HF)

3. Informações necessárias

Para que uma comunicação MODBUS possa ser configurada é necessário levantar as seguintes informações:

- Confirmar tipo de MODBUS que será usado (MODBUS RTU).
- Tipo de meio físico que será usado. Em especial, verifique se é RS232, RS 422 ou RS 485. Verifique também:
 - Conectores /que sinal deve ser ligado a que terminal.
 - Velocidade de comunicação
 - Número de bits por palavra
 - Número de start bits
 - Número de stop bits
 - Paridade
- O papel que cada um dos equipamentos vai exercer (quem é mestre, quem é escravo).
- Endereço das variáveis a serem trabalhadas, forma de notação usada e qual valor inicial é adotado (vide item [Endereçamento de variáveis](#)).

Ligação elétrica – RS 485

A ligação elétrica depende dos equipamentos utilizados (tanto do lado do mestre como dos escravos). Por isso é preciso obter com todos os fabricantes dos equipamentos envolvidos informações detalhadas sobre a forma de ligação de seus equipamentos.

Princípio geral

Comunicação RS 485 é baseada em um par de fios onde todos os equipamentos são ligados em paralelo. Em geral é recomendado o uso de resistores de terminação para casamento de impedância nas pontas da rede.

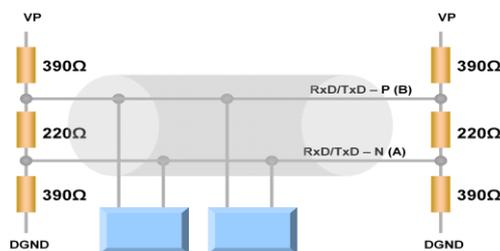


Figura 1 – No RS 485 os equipamentos são ligados em “paralelo” sobre o barramento de comunicação. Resistores de terminação promovem o casamento de impedância para altas frequências.

Os equipamentos podem ser ligados à rede RS 485 via bornes ou via conector DB9.

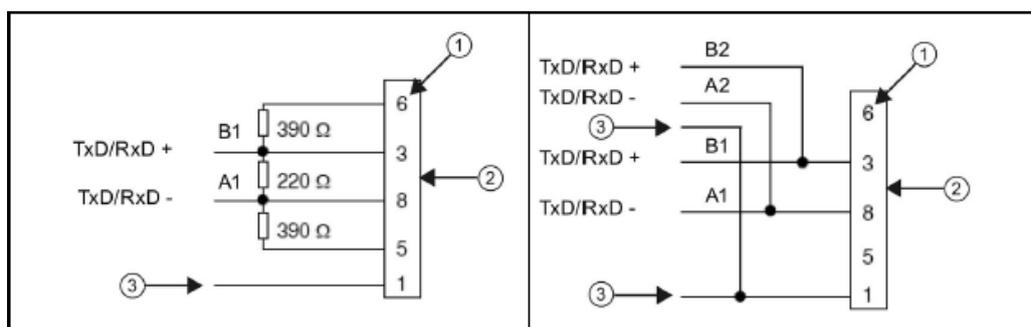


Figura 2 – Esquema de ligação para RS485 usando conectores DB9. Nas conexões das pontas de rede é necessário acrescentar resistores de terminação: (2) conector, (1) pinos do conector, (3) blindagem.

Cabo e conectores PROFIBUS são próprios para uso em RS 485 e podem ser usados em redes MODBUS também. Os conectores PROFIBUS são padrão DB9 e já tem resistores de casamento de impedância incorporados, ativados por chaves.

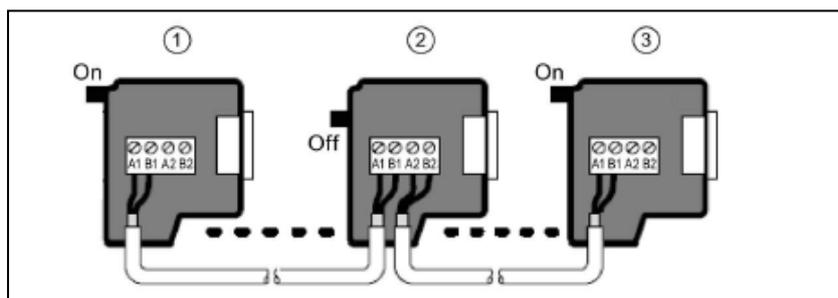


Figura 3 - Exemplo de rede RS485 usando cabos e terminações PROFIBUS – as chaves para ativar os resistores de terminação devem estar em ON nas pontas (conectores 1 e 3).

Detalhes sobre a conexão dos cabos podem ser obtidos nos capítulos abaixo ou no artigo “connect the RS485/RS422 interfaces of SIMATIC and SIPLUS modules for serial communication?” (<https://support.industry.siemens.com/cs/ww/en/view/109736665>).

S7-1200 CB 1241 (RS485) 6ES7 241-1CH30-1XB0

A CB 1241 tem conectores DB9, que permitem o uso com cabos e conectores PROFIBUS.

Como alternativa ao uso de conectores de PROFIBUS o módulo CB1241 também tem internamente os resistores de terminação. Isso possibilita o uso de conectores mais simples.

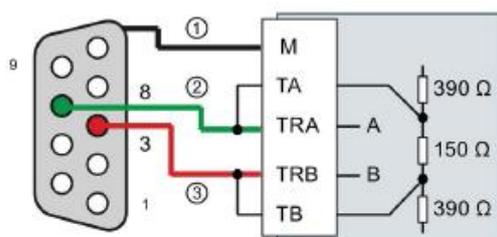


Figura 4 – Esquema de ligação de conector DB9 (não PROFIBUS) para uso de resistores de terminação internos da CB 1241. (1): malha de blindagem; (2): sinal A; (3) sinal B;

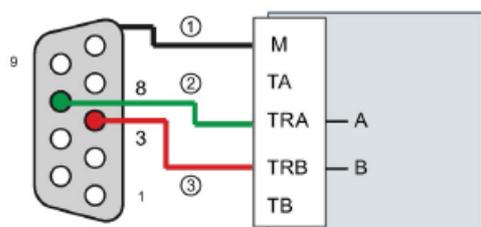


Figura 5 – Esquema de ligação de conector (não PROFIBUS) para uso sem resistores de terminação. (1): malha de blindagem; (2): sinal A; (3) sinal B;

S7-1200 CM 1241 (RS422/RS485) 6ES7 241-1CH32-0XB0

O módulo de comunicação CM 1241 (RS422/RS485) 6ES7 241-1CH32-0XB0 tem interface DB9. É possível usar conectores PROFIBUS ou montar conector próprio (quando o conector é montado no início ou fim da rede deve-se adicionar resistor de 330 Ohms).

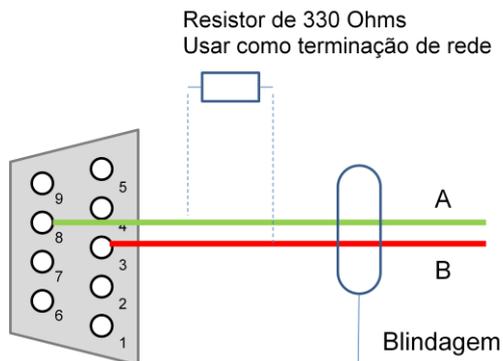


Figura 6 - Esquema de ligação de conector RS 485 DB9 para CM 1241. Um resistor de 330 Ohms deve ser acrescentado às pontas da rede.

S7-1500 / ET200MP CM PtP RS 422/485 HF 6ES7541 1AB00-0AB0

O módulo de comunicação CM PtP RS 422/485 HF 6ES7541 1AB00-0AB0 tem interface DB15. Quando o conector é montado no início ou fim da rede deve-se adicionar resistor de 330 Ohms.

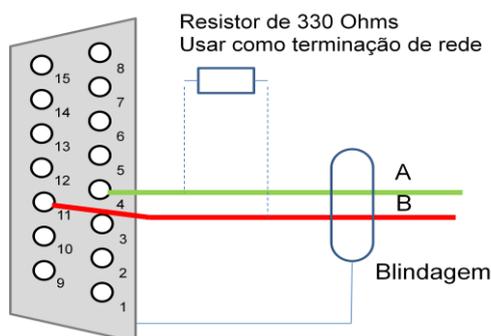


Figura 7 - Esquema de ligação de conector RS 485 DB15 para CM PtP RS 422/485 HF. Um resistor de 330 Ohms deve ser acrescentado às pontas da rede.

ET200SP 6ES7 137 6AA00-0BA0

O módulo de comunicação do ET200 SP 6ES7 137 6AA00-0BA0 tem interface através dos bornes de sua base (6ES7193-6BP00-0BA0 ou 6ES7193-6BP00-0DA0).

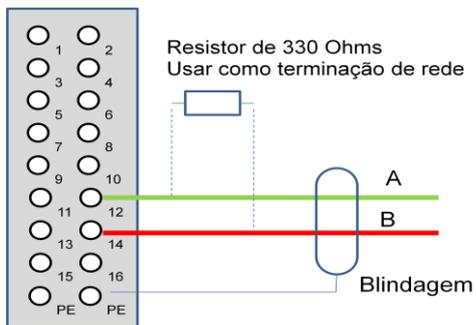


Figura 8 - Esquema de ligação para bases 6ES7193-6BP00-0BA0 / 6ES7193-6BP00-0DA0 do ET200 SP. Um resistor de 330 Ohms deve ser acrescentado quando o módulo às pontas da rede.

4. Configuração e Programação

Estrutura Geral

A comunicação MODBUS RTU é definida através da declaração e configuração dos componentes (hardware) utilizados e através da chamada de blocos de programação no programa do usuário.

A declaração e parametrização do hardware é feita na janela “Device View”.

No programa de usuário são chamados os blocos Modbus_Comm_Load e Modbus_Master.

Configuração de Hardware

A pesar dos hardwares (CM1241, CB1241, S7-1500 CM PTP HF, ET200SP CM PtP) terem detalhes específicos, eles têm características semelhantes a serem configuradas.

O primeiro passo é identificar o módulo a ser usado e arrastá-lo para o rack do controlador / remota.

O segundo passo é definir um nome simbólico ao módulo (o nome do módulo de comunicação será usado no programa do usuário). As figuras abaixo mostram onde isto é feito:

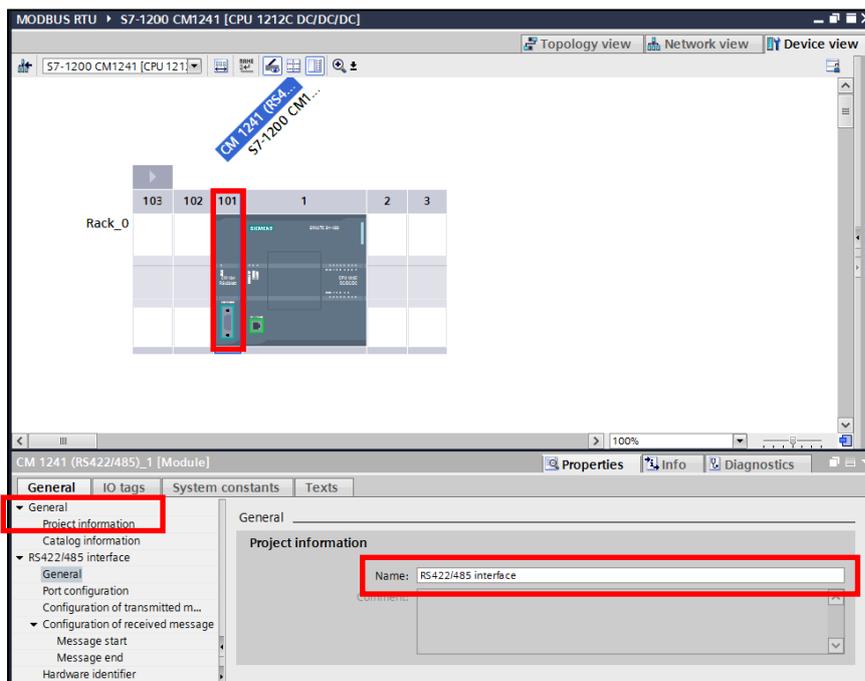


Figura 9 – CM 1241 para S7-1200 - Identificação do nome do módulo de comunicação (em vermelho).

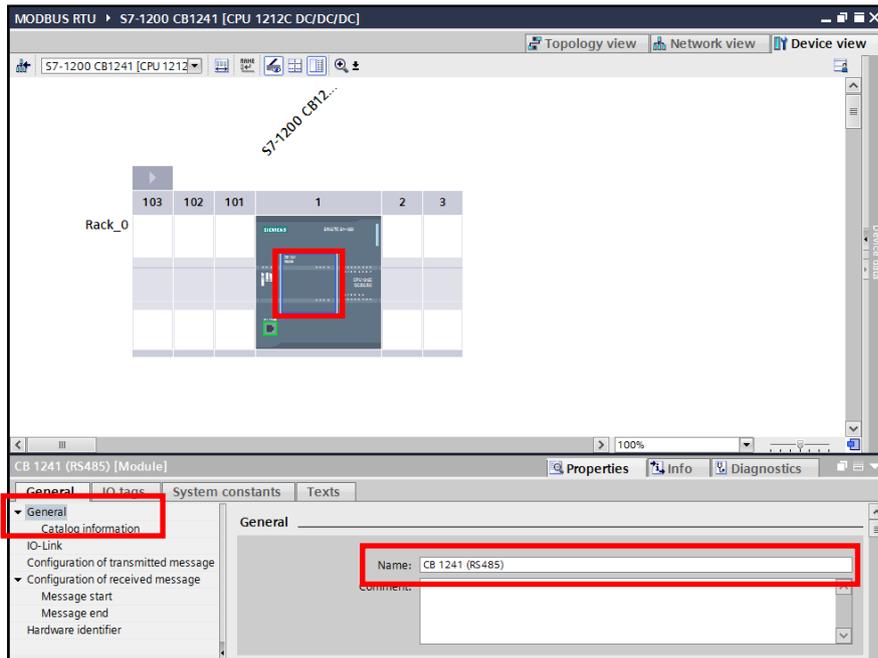


Figura 10 – CB1241 para S7-1200 - Identificação do nome do módulo de comunicação (em vermelho).

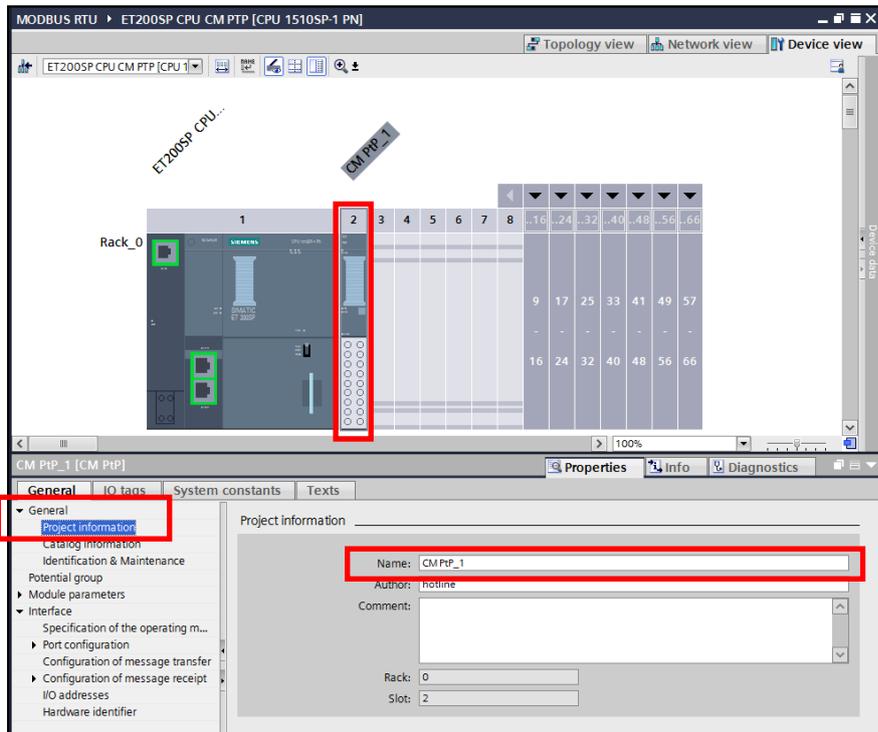


Figura 11 – CM PTP para ET200SP - Identificação do nome do módulo de comunicação (em vermelho).

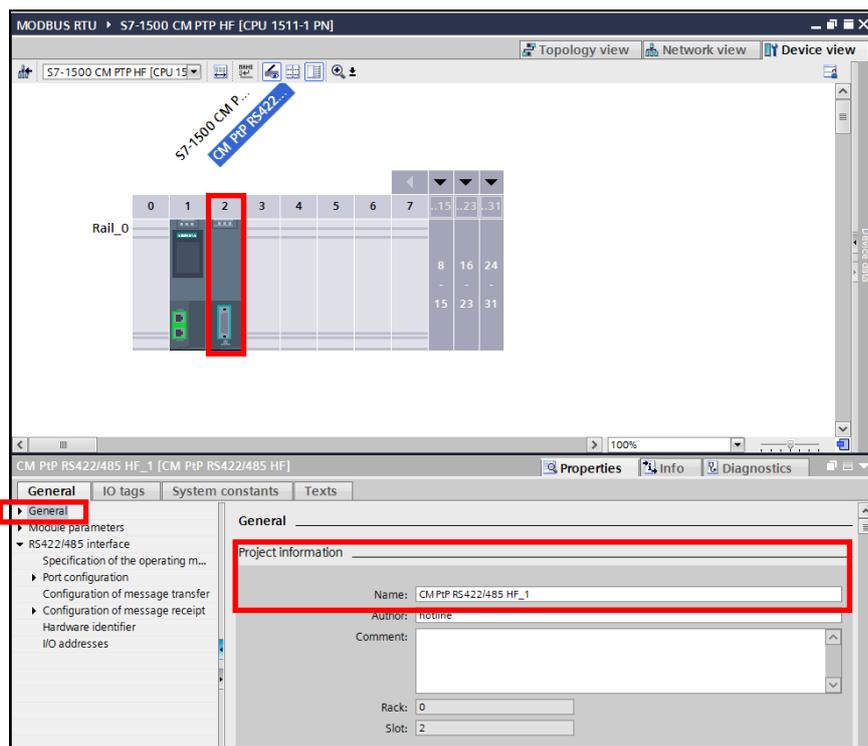


Figura 12 – CM PTP HF para S7-1500 - Identificação do nome do módulo de comunicação (em vermelho).

O ultimo passo de configuração do hardware é definir as propriedades.

No ET200SP e no S7-1500 é necessário ajustar o parâmetro Protocol para o valor “Freeport/MODBUS” (no S7-1200 esta parametrização não é necessária, este ajuste já é implícito). Os demais parâmetros do MODBUS (por exemplo, velocidade, paridade, etc.) são feitos pelas chamadas dos blocos do usuário.

Conforme o hardware usado, alguns parâmetros podem aparecer em tanto no “Device View” quanto nas chamadas dos blocos. A configuração feita através de chamada de blocos tem prioridade sobre o definido no “Device View” e por isso a configuração destes valores no “Device View” é irrelevante. Sugere-se, porém, deixar os parâmetros do “Device View”, com valores coerentes para fins de organização / documentação.

As imagens abaixo mostram as propriedades de rede do “Device View” para os diferentes hardwares:

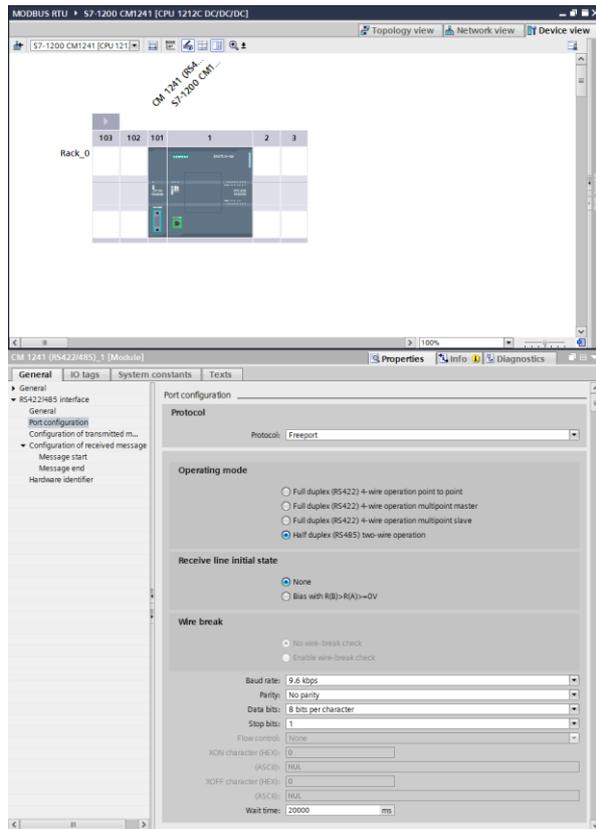


Figura 13 - CM 1241 para S7-1200 – Os parâmetros desta configuração são sobrescritos pelos parâmetros do bloco Modbus_Comm_Load. Sugere-se, porém, manter uma coerência de valores para efeitos de organização e documentação.

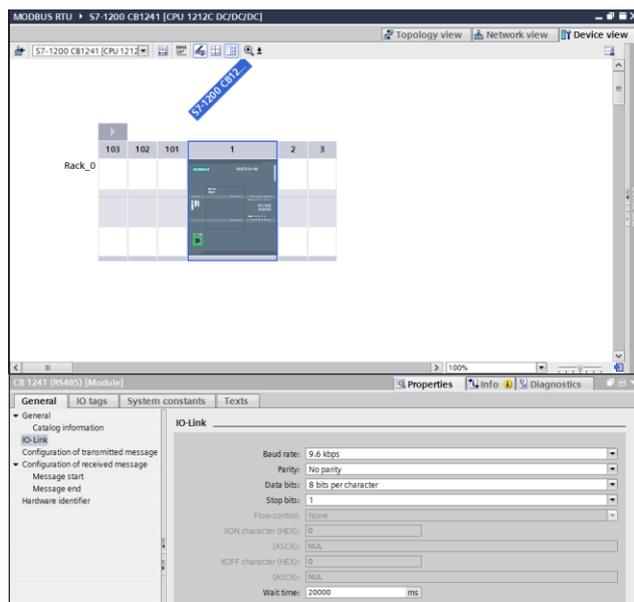


Figura 14 - CB 1241 para S7-1200 – Os parâmetros desta configuração são sobrescritos pelos parâmetros do bloco Modbus_Comm_Load. Sugere-se, porém, manter uma coerência de valores para efeitos de organização e documentação.

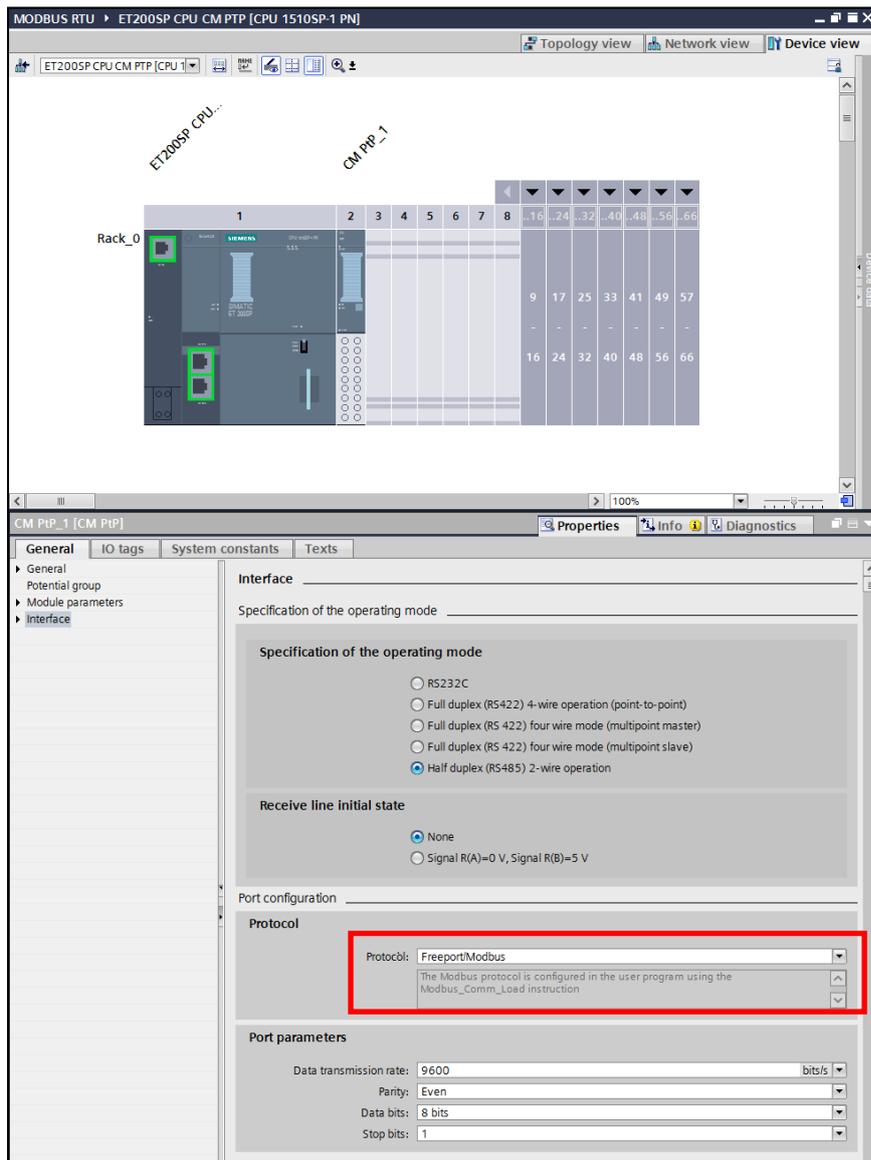


Figura 15 – CP PTP para ET200SP – Deixar parâmetro Protocol com valor “Freeport/MODBUS”. Os demais parâmetros desta configuração são sobrescritos pelos parâmetros do bloco Modbus_Comm_Load. Sugere-se, porém, manter uma coerência de valores para efeitos de organização e documentação.

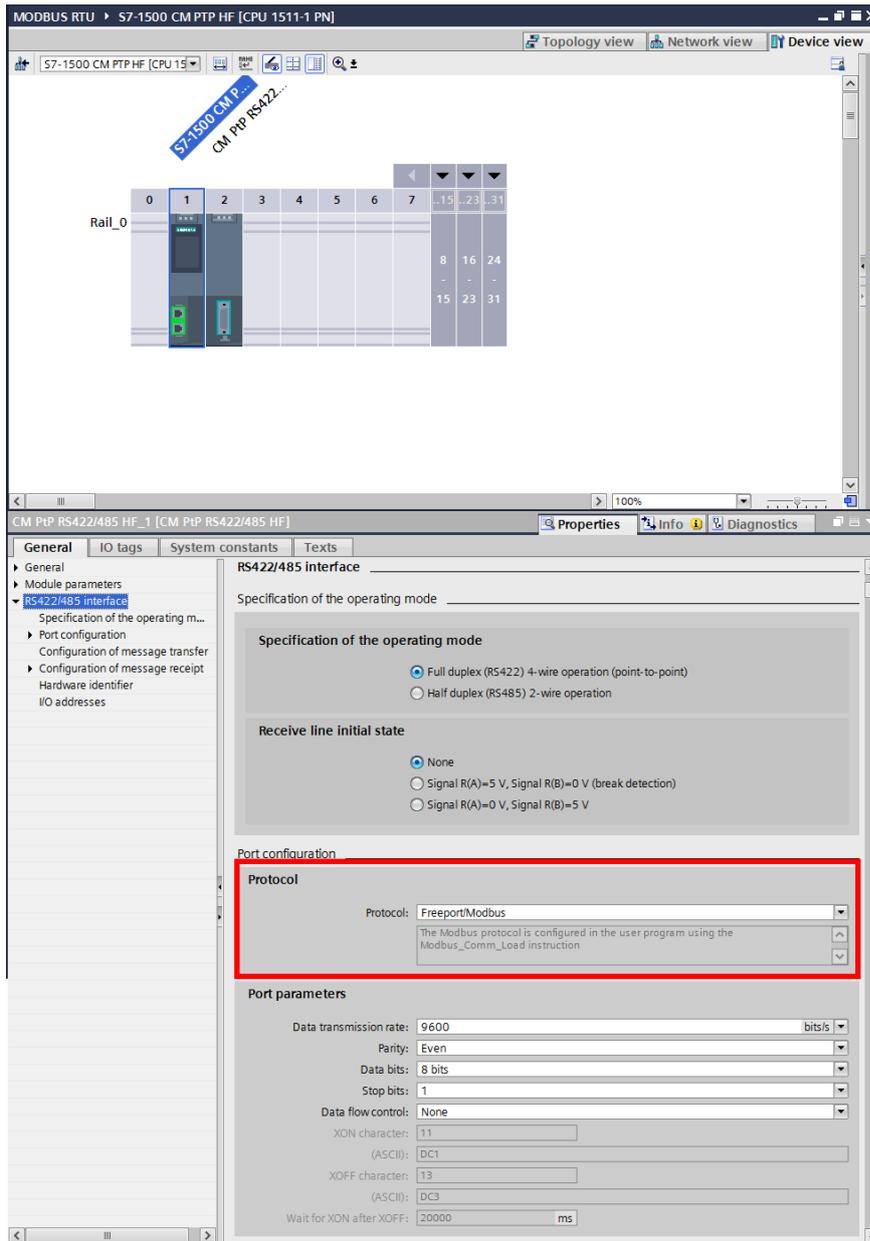


Figura 16 - CP PTP para S7-1500 – Deixar parâmetro Protocol com valor “Freeport/MODBUS”. Os demais parâmetros desta configuração são sobrescritos pelos parâmetros do bloco Modbus_Comm_Load. Sugere-se, porém, manter uma coerência de valores para efeitos de organização e documentação.

Modbus_Comm_Load

O bloco Modbus_Comm_Load é responsável pela parametrização do módulo de comunicação.

Ele está disponível no catálogo de instruções, no caminho *Communication -> Communication Processor -> MODBUS (RTU)*. Recomenda-se usar sempre a versão mais atualizada disponível para sua CPU.

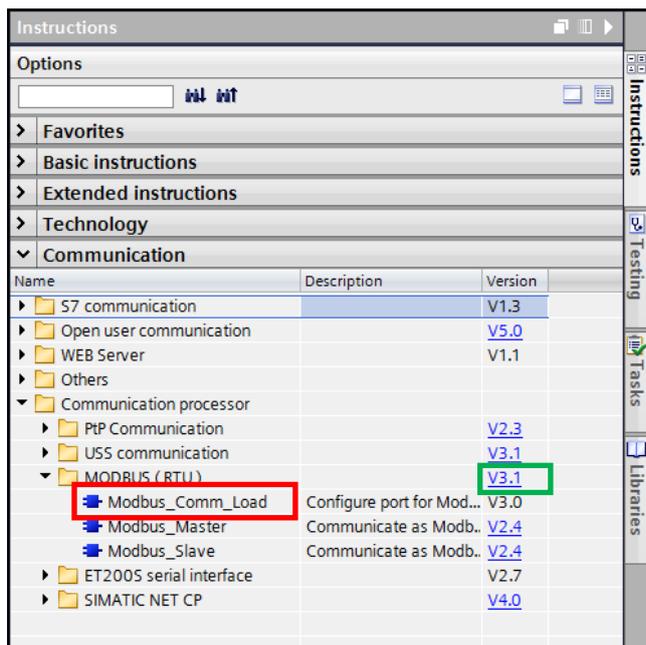


Figura 17 – Em Vermelho: Bloco Modbus_Comm_Load na janela de instruções. Em verde: na coluna Version, linha MODBUS (RTU) pode-se escolher a versão a ser usada (usar versão mais atual).

O bloco Modbus_Comm_Load deve ser chamado a todo ciclo do PLC (e.g. a partir de um OB de ciclo de programa, OB1, ou por um OB de interrupção cíclica).

Os seguintes parâmetros são relevantes para uso com RS 485:

- REQ deve receber um pulso toda vez que o usuário desejar configurar o módulo de comunicação (e.g. isto é feito no primeiro scan do OB). Recomenda-se usar a variável de sistema “first scan” nesta entrada.
- PORT deve ser configurado com o nome simbólico do módulo de comunicação (constantes de sistema). É possível dar browser nos nomes com um clique de mouse sobre a entrada. Notar que o nome simbólico do módulo é precedido por “local” quando no rack central ou pelo nome simbólico da remota.

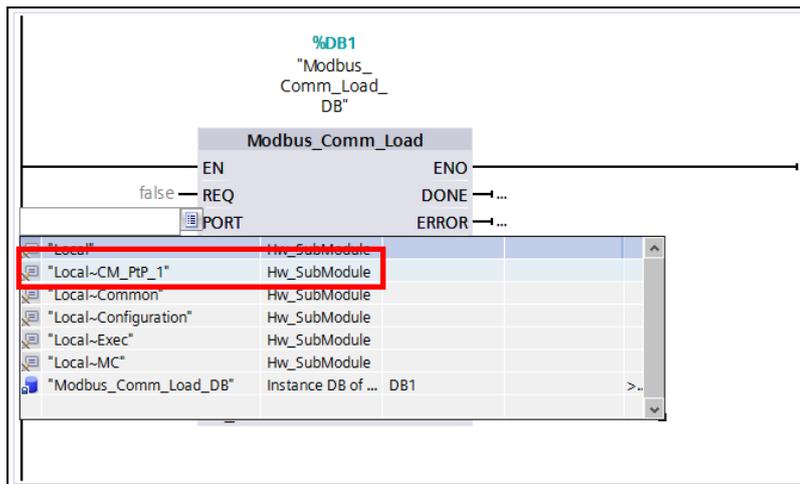


Figura 18 – Seleção do módulo através de clique na entrada PORT. Ao nome simbólico do módulo é adicionado um prefixo “Local” para módulos no rack central ou com o nome da remota onde ele está montado.

- BAUD deve ser configurado com a velocidade da rede (unidade de bits por segundo). Valor depende de características do escravo.
- PARIDADE deve receber 0 se o telegrama não usar bit de paridade, 1 se o telegrama usar um bit de paridade ou 2 se o telegrama usar dois bits de paridade. Valor depende de características do escravo.
- RESP_TO deve ser configurado com o tempo máximo (unidade milissegundo) que o MODBUS deve esperar pela resposta de um escravo antes de considerar que houve falha na comunicação por “falta de resposta”. Valor depende de características do escravo. Em geral usa-se o valor default de 1 segundo.
- MB_DB deve ser preenchida com a variável MB_DB do instance DB do bloco Modbus_Master (descrito a seguir).

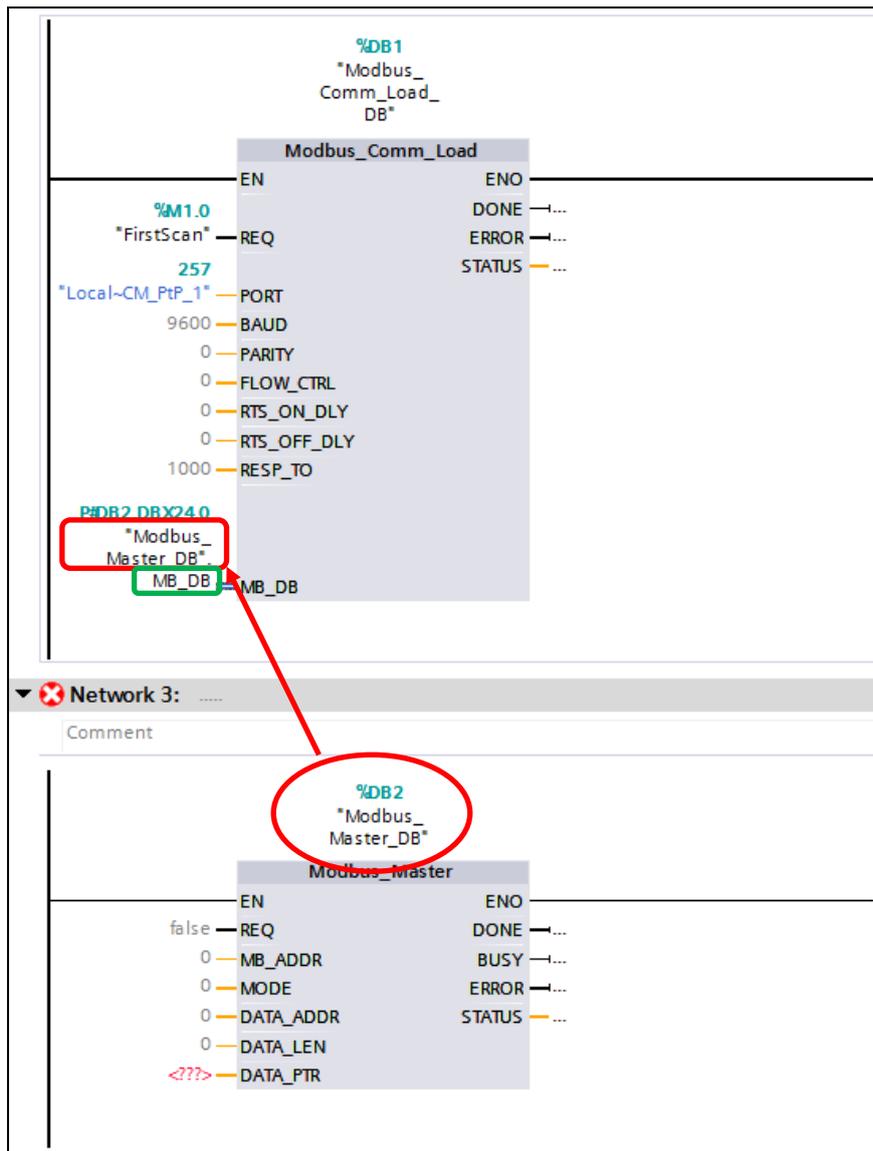


Figura 19 – O parâmetro de entrada MB_DB deve ser conectado a variável do bloco Modbus_Master (instance DB) de mesmo nome.

Além dos parâmetros de entrada do bloco, existem algumas variáveis internas que devem ser ajustadas antes da chamada do Modbus_Comm_Load (e.g. via blocos de MOVE):

- MODE = 4 para RS 485.
- STOP_BITS = número de STOP BITS da comunicação. Valor depende de características do escravo.
- LINE_PRE = polarização da rede em estado inicial. Valor depende de características do escravo. Em geral, é usado valor 0 (sem polarização). Em casos raros é usado valor 2 (A=0V, B=5V). Valor depende de características do escravo.

- RETRIES = número de tentativas de comunicação (reenvios de telegramas) antes de considerar que houve falha por “falta de resposta” (atua em conjunto com o parâmetro de entrada RESP_TO). Em geral usa-se o valor default de 2.
- ICHAR_GAP = Tempo adicional entre caracteres (unidade décimos de character delay time) somado ao valor interno de 3.5 character times. Em geral, mantém-se o valor default de 0. Valor depende de características do escravo.
- EN_SUPPLY_VOLT = habilita diagnostico por falta de alimentação.
- BRK_DET = habilita diagnóstico de quebra de fio.
- EN_DIAG_ALARM = habilita geração de interrupção de diagnostico.

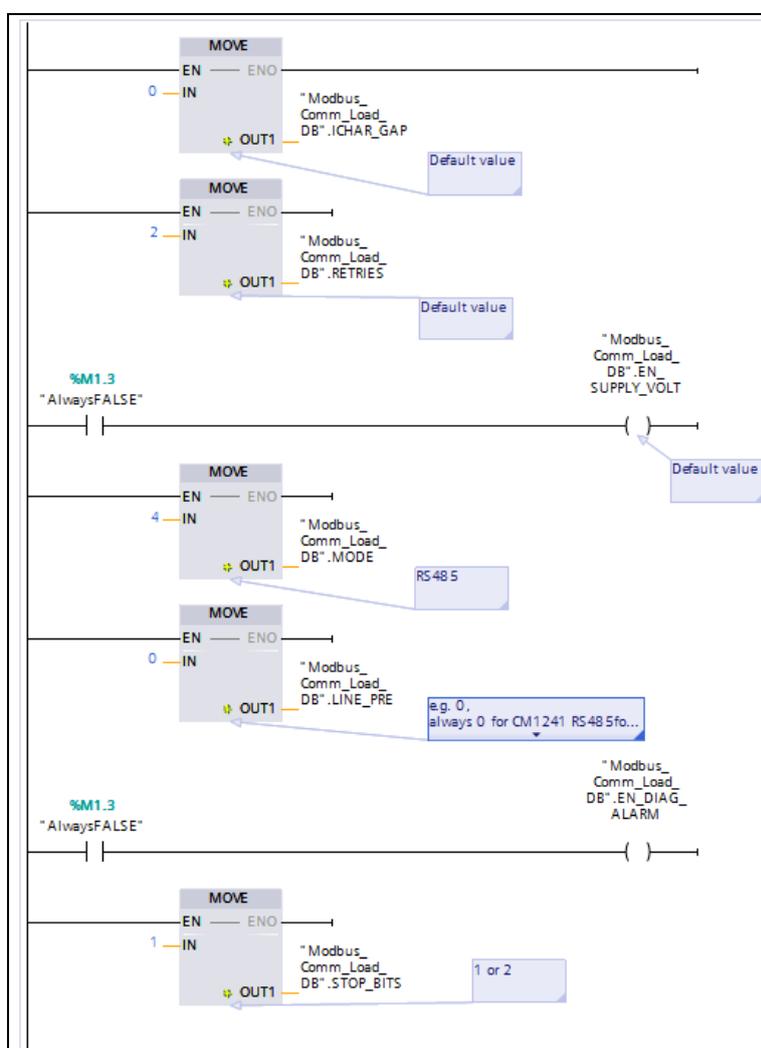


Figura 20 - Exemplos de atribuição de valores às variáveis internas do bloco Modbus_Comm_Load através de instruções.

Modbus_Master

O bloco Modbus_Master é usado para que o módulo de comunicação configurado com Modbus_Comm_Load atue como um mestre na rede MODBUS RTU.

Este bloco deve ser chamado a todo ciclo do PLC (e.g. a partir de um OB de ciclo de programa, OB1, ou por um OB de interrupção cíclica).

Segue a descrição de seus parâmetros:

- REQ quando em TRUE, envia um telegrama ao escravo. Recomenda-se usar um detector de borda em série com esta entrada para evitar um “congestionamento” de pedidos de envio de telegramas. Em geral sugere-se também que esta entrada seja ligada a um clock memory de tempo razoável para monitoração a olho nu durante a fase de testes.

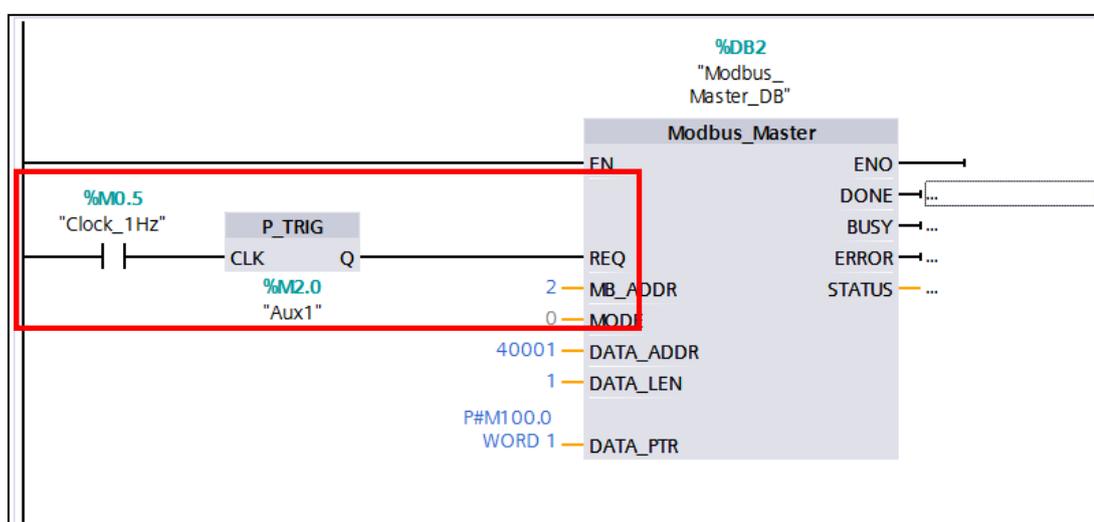


Figura 21 – Exemplo de disparo de requisições de comunicação: o bloco P_TRIG transforma o clock de 1 Hz em pulsos de duração de um ciclo e intervalo entre pulsos de 1 segundo.

Nota: quando o escravo não está presente, o bloco espera um tempo equivalente ao $RESP_TO \times RETRIES$ para concluir sua tarefa. Procure dimensionar o clock, $RESTP_TO$ e $RETRIES$ de modo a evitar gerar novas requisições de comunicação neste intervalo.

- MB_ADDR deve ser preenchido com o número do escravo na rede. Valor depende de características do escravo.
- MODE deve ser preenchido com 0 para ler valores do escravo e com 1 para escrever.
- DATA_ADDR deve ser preenchido com o número da variável do escravo a ser acessada. A formação deste endereço é uma combinação do Function Code e do endereço local da variável do escravo. Valor depende de características do escravo. Vide detalhes conceituais em [Endereçamento de variáveis](#). A tabela a seguir mostra

como MODE e DATA_ADDR são combinados para formar um endereçamento MODBUS:

Function Code	Significado	Quantidade	MB_MODE	MB_DATA_ADDR
01	Leitura de "output bits" (bits)	1 a 2000	0	1 a 9999
02	Leitura de "input bits" (bits)	1 a 2000	0	10001 a 19999
03	Leitura de "holding registers" (words)	1 a 125	0	40001 a 49999 400001 a 465535
04	Leitura de "input registers (words)	1 a 125	0	3001 a 39999
05	Escrita de um "output bits" (bit)	1	1	1 a 9999
06	Escrita de um "holding registers" (word)	1	1	40001 a 49999 400001 a 465535
15	Escrita de vários "output bits" (bits)	2 a 1968	1	40001 a 49999 400001 a 465535
16	Escrita de vários "holding registers" (words)	2 a 123	1	40001 a 49999 400001 a 465535

- DATA_LEN deve ser preenchido com a quantidade de variáveis a partir do endereço inicial descrito em DATA_ADDR que será acessada (para leitura/escrita de apenas uma variável, usar 1).
- DATA_PTR deve ser preenchido com o endereço do controlador onde as variáveis MODBUS serão lidas/escritas. Sugere-se endereços não otimizados no controlador e notação de ponteiros com endereço absoluto.

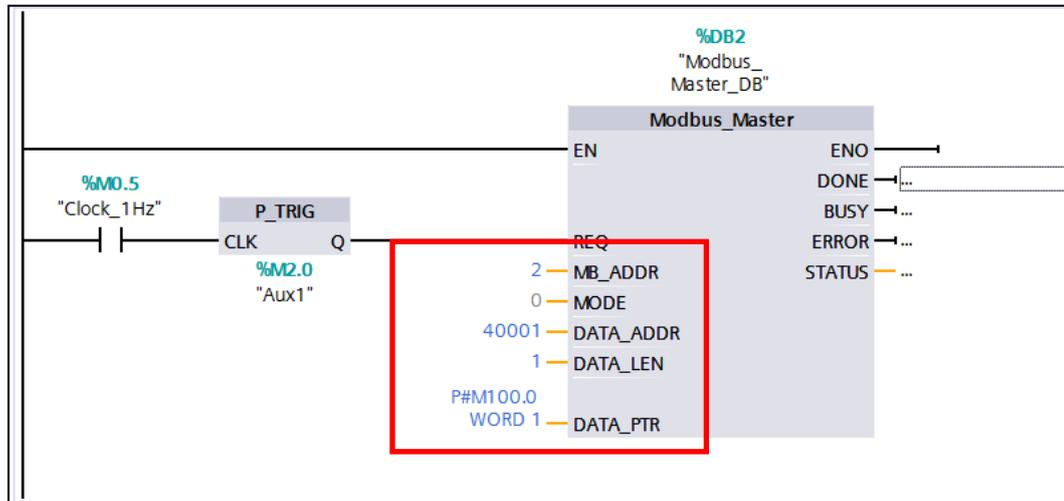


Figura 22 – Exemplo de endereçamento MODBUS: é lida (MODE =0) uma variável (DATA_LEN = 1) de endereço 40001 (o que equivale a Function code = 3 = leitura de variáveis de 16 bits, primeiro endereço disponível) e salvo no controlador no MW 100 (P#M100.0 WORD 1 = uma Word a partir do M100.0).

5. Técnicas de diagnóstico

A principal forma de diagnosticar uma comunicação MODBUS é através das variáveis de saída dos blocos Modbus_Comm_Load e Modbus_Master .

Descrição das saídas do bloco Modbus_Comm_Load:

- DONE fica em TRUE durante um ciclo (“scan”), para indicar que o pedido de reconfiguração do módulo de comunicação (REQ = TRUE) foi completado com sucesso.
- ERROR fica em TRUE durante um ciclo (“scan”) para indicar que o pedido de reconfiguração do módulo de comunicação (REQ = TRUE) não foi completado com sucesso, isto é, houve um erro.
- STATUS indica o status do bloco durante o respectivo ciclo (“scan”). Apenas valores na casa dos 8000 (hexadecimal) indicam efetivamente erros. Nota: a indicação do tipo do erro ocorrido ocorre somente durante um ciclo (“scan”) correspondente ao momento em que a variável ERROR fica em TRUE. O significado dos códigos de erro podem ser obtidos do sistema de ajuda do TIA Portal (selecionar o bloco e pressionar F1).

Descrição das saídas do bloco Modbus_Master:

- DONE fica em TRUE durante um ciclo (“scan”), para indicar que o pedido de comunicação (REQ = TRUE) foi completado com sucesso.
- BUSY fica em TRUE para indicar que um pedido de comunicação ainda está sendo processado (informação não foi completamente enviada ou ainda não se obteve uma resposta).
- ERROR fica em TRUE durante um ciclo (“scan”) para indicar que o pedido de comunicação (REQ = TRUE) não foi completado com sucesso, isto é, houve um erro.
- STATUS indica o status do bloco durante o respectivo ciclo (“scan”). Apenas valores na casa dos 8000 (hexadecimal) indicam efetivamente erros. Nota: a indicação do tipo do erro ocorrido ocorre somente durante um ciclo (“scan”) correspondente ao momento em que a variável ERROR fica em TRUE. O significado dos códigos de erro podem ser obtidos do sistema de ajuda do TIA Portal (selecionar o bloco e pressionar F1).

Os eventos registrados por estas variáveis ficam disponíveis por apenas um ciclo (“scan”) do programa. Por isso é necessário processar estes sinais pelo programa a fim de que usuário possa ter acesso a estas informações.

A sugestão é que se conte durante quantos ciclos (“scans”) o sinal de DONE ficou em TRUE, que se conte durante quantos ciclos (“scans”) o sinal de ERROR ficou em TRUE e que se salve (MOVE) em alguma memória auxiliar (não temporária) o valor de STATUS para o ciclo (“scan”) em que ERROR esteve em TRUE. Vide exemplo de lógica abaixo:

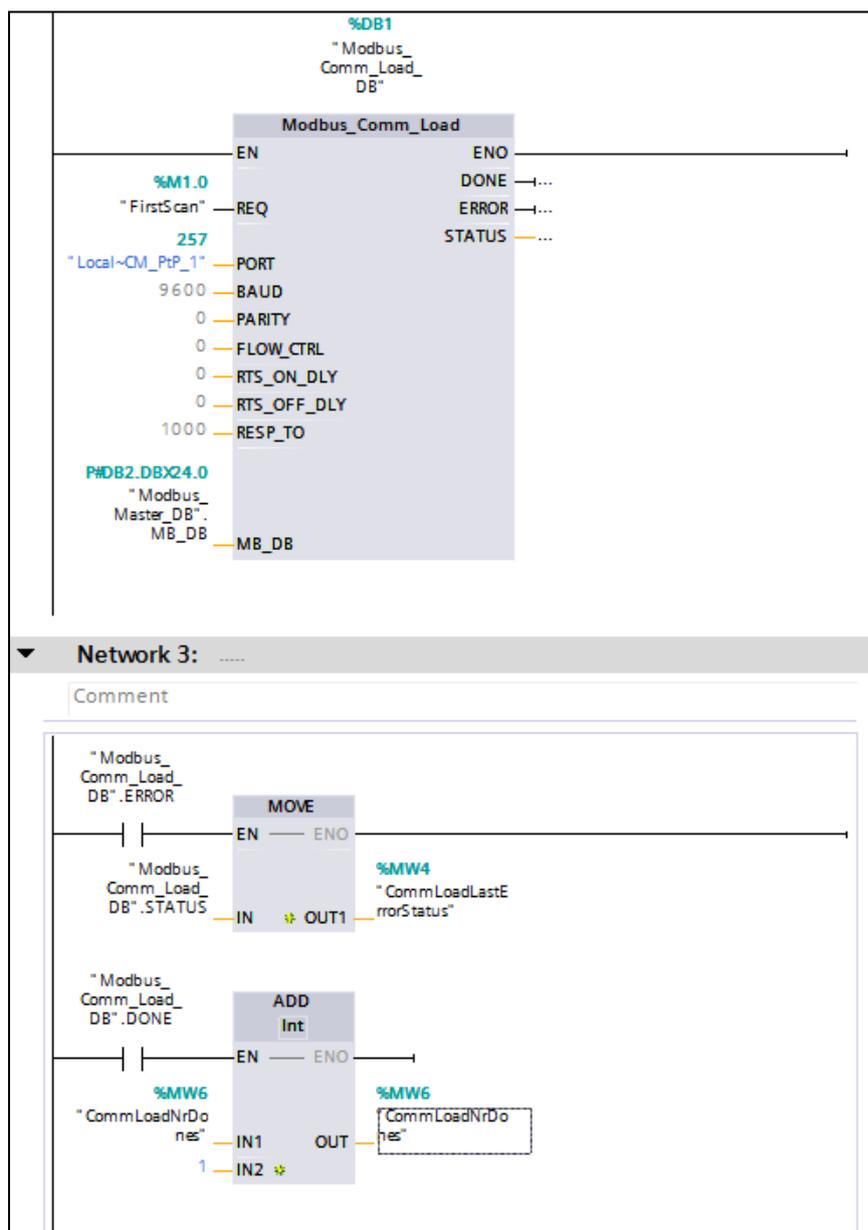


Figura 23 – Exemplo de programa para avaliação do diagnóstico para o bloco Modbus_Comm_Load memorizando o valor de STATUS para o evento de ERROR = TRUE e contando o número de ciclos (“scans”) em que DONE = TRUE (o mesmo conceito de contagem pode ser usado também para o evento de ERROR)

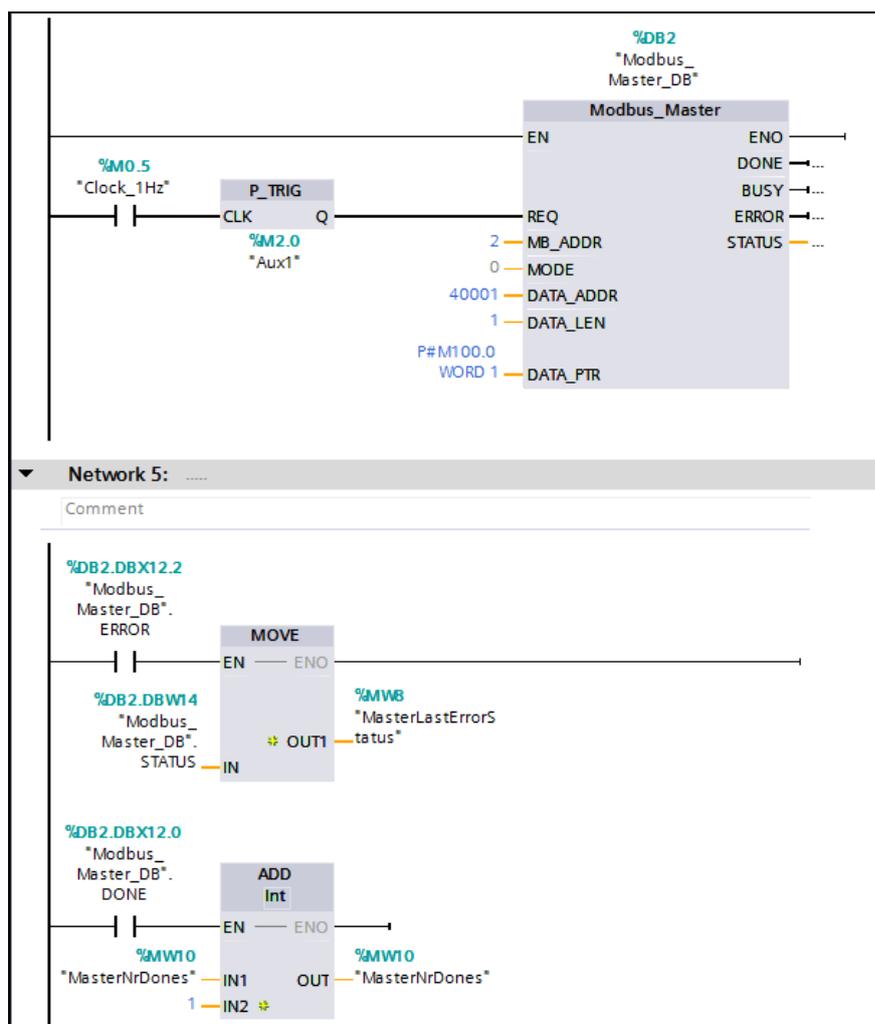


Figura 24 - Exemplo de programa para avaliação do diagnóstico para o bloco Modbus_Master memorizando o valor de STATUS para o evento de ERROR = TRUE e contando o número de ciclos (“scans”) em que DONE = TRUE (o mesmo conceito de contagem pode ser usado também).

Dica: No link <https://support.industry.siemens.com/cs/br/en/view/54377291>, existe um bloco pronto (V11) para fazer estas funções de diagnose.

Existem outras formas de obter dados de diagnose:

- Através dos LED's dos módulos. Os manuais dos módulos de comunicação descrevem o significado dos LED's, maiores detalhes, vide [Referências, exemplos e downloads](#). Através deste recurso pode-se detectar se o hardware está em falha, se telegramas estão sendo enviados ou recebidos.
- Através do TIA Portal, monitorando o hardware (e.g. Device View) online. Clicando com o botão direito sobre o módulo, pode-se ir em “Online & Diagnose” específico (maiores detalhes vide artigo publicado na internet em <https://support.industry.siemens.com/cs/us/en/view/88628706>). Através deste

recurso pode-se detectar se o hardware foi configurado corretamente, se ele se comunica com a CPU, se há alguma falha interna ou externa.

- Através do Diagnostic Buffer da CPU. Através deste recurso pode-se ver se a CPU acessa o módulo de comunicação, se este está relatando à CPU algum diagnóstico ou se existem erros de endereçamento no programa.
- Monitorando o Instance DB dos blocos de comunicação. Através deste recurso pode-se monitorar se os valores configurados efetivamente estão sendo processados. Alguns diagnósticos internos dos blocos também podem ser obtidos desta forma.

6. Dicas para operar com vários escravos

É possível usar o mesmo módulo de comunicação para realizar mais de uma tarefa de leitura / escrita com um único escravo ou para comunicar com mais de um escravo desde que se obedeça a certas regras:

Para cada módulo de comunicação usado no projeto:

- Deve-se existir apenas um instance DB do bloco Modbus_Comm_Load.
- Deve-se chamar apenas uma vez por ciclo (“scan”) o bloco Modbus_Comm_Load.
- Deve-se existir apenas um instance DB do bloco Modbus_Master.
- Deve-se chamar apenas uma vez por ciclo (“scan”) o bloco Modbus_Master.

A varredura (“pooling”) de várias tarefas (como comunicar com diferentes escravos) pode ser abordada de diferentes modos pelo programa do usuário, como por exemplo:

- Chamando o bloco Modbus_Master apenas uma vez na lógica, mas mudando o valor seus parâmetros a cada tarefa diferente.
- Usando o bloco Modbus_Master em as vezes, cada qual com valores diferentes de parâmetros mas alternado sua chamada (seja por “jumps” seja por habilitar/desabilitar a entrada EN) de modo que apenas uma chamada seja executada por ciclo.

Caso uma nova requisição de comunicação seja iniciada antes que a anterior esteja concluída (seja com sucesso, seja com falha), o bloco Modbus_Master retornará um erro.

Recomenda-se que a alternância entre requisições de comunicações (Modbus_Master , REQ =1) seja feita em “malha aberta” (tempo fixo). Este tempo deve ser superior ao tempo máximo necessário para que a tarefa comunicação anterior tenha sido concluída (seja com sucesso ou com falha). O tempo máximo para concluir a tarefa ocorre quando o parceiro não responde dentro do tempo previsto e que todas as tentativas de reenvio tenham sido esgotadas (isto é, RESP_TO x RETRIES).

Havendo necessidade de tempos de varredura mais curtos pode-se adicionar à “malha aberta” um controle por “malha fechada” (através da observação dos sinais de “DONE” e “ERROR”). Recomenda-se que neste caso use-se um conjunto de sinais de “DONE ou ERRO ou Tempo de malha aberta esgotado” para disparar o próximo evento de comunicação.

Vide, abaixo, exemplo de programa usado para fazer varredura de dois escravos:

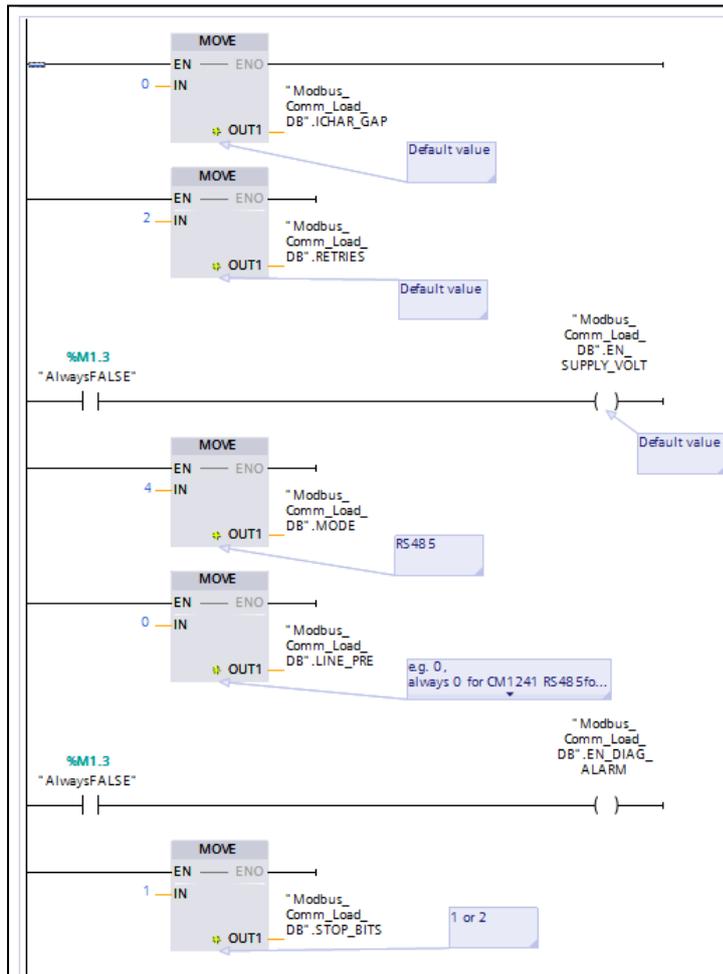


Figura 25 – Network 1 – Atribuição de valores aos parâmetros internos do bloco Modbus_Comm_Load.

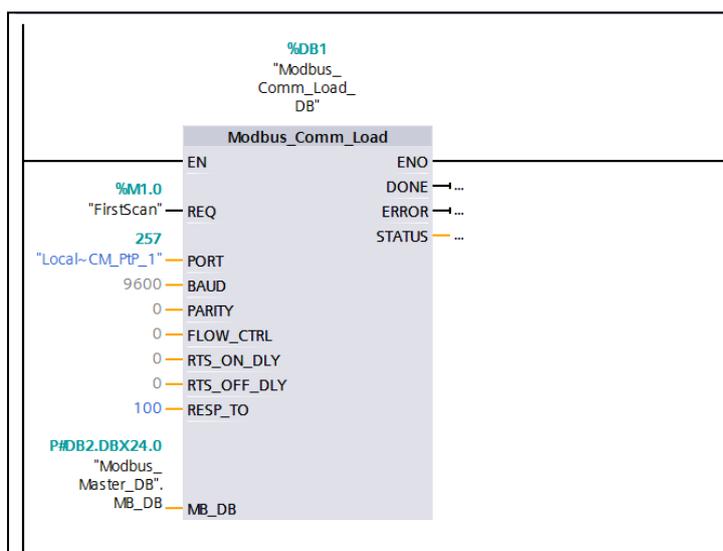


Figura 26 - Network 2 - Chamada do bloco Modbus_Comm_Load.

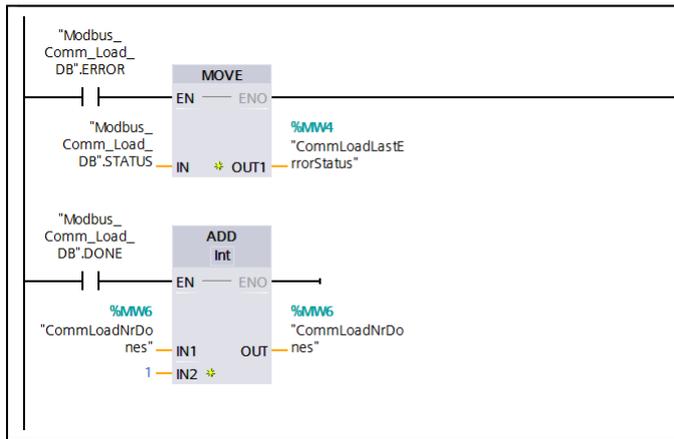


Figura 27 – Network 4 - Monitoração dos dados de diagnose do bloco Modbus_Comm_Load

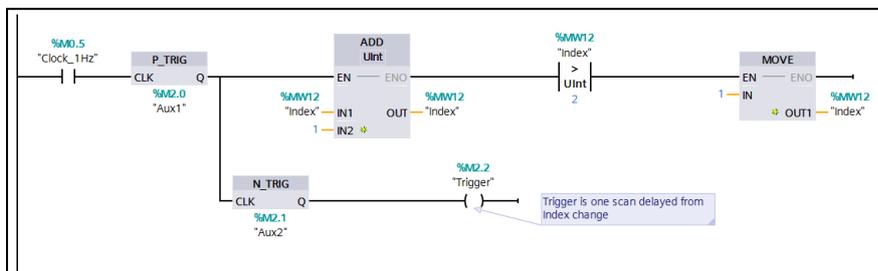


Figura 28 – Network 5 - Geração de variáveis para controle da varredura em “malha aberta” (recomendada para testes iniciais): Index varia entre 1 e 2 e trigger fica em TRUE por um ciclo, após a mudança de estado de Index.

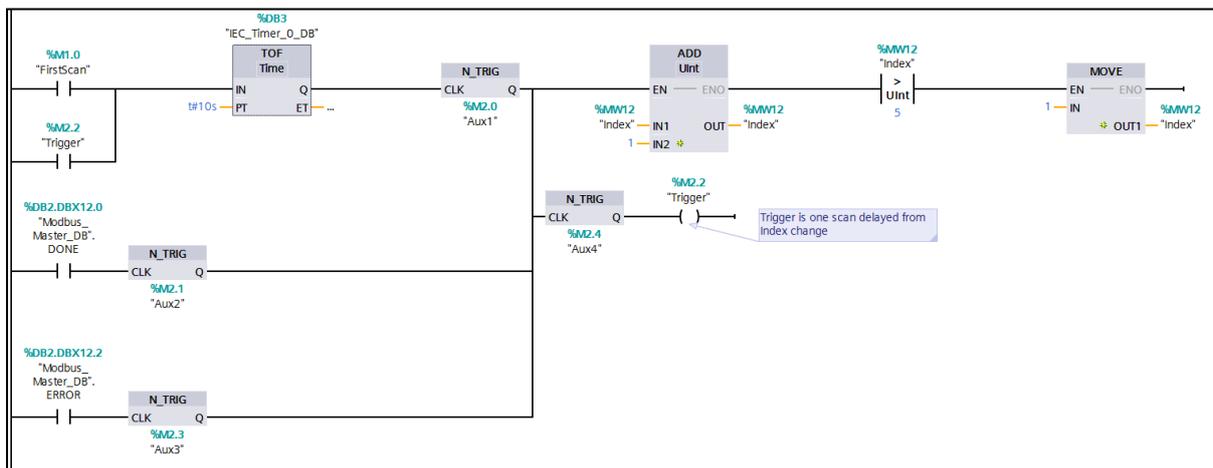


Figura 29 - Network 5 - Geração de variáveis para controle da varredura em “malha fechada” (recomendada para onde se precisa de rápidos tempos de varredura): Index varia entre 1 e 2 e trigger fica em TRUE por um ciclo, após a mudança de estado de Index.

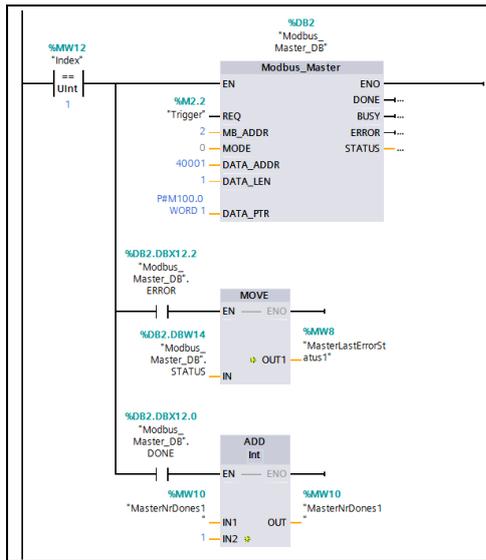


Figura 30 – Network 6 - Chamada do bloco Modbus_Master para primeiro escravo (Index = 1), seguida de avaliação dos dados de diagnóstico.

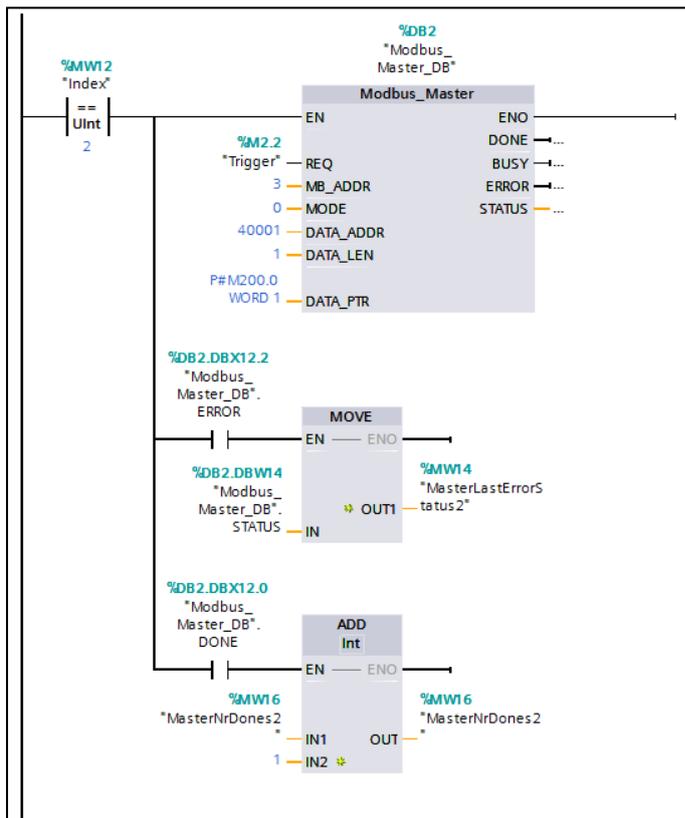


Figura 31 – Network 7 - Chamada do bloco Modbus_Master para segundo escravo (Index = 2), seguida de avaliação dos dados de diagnóstico.

Adaptação do exemplo acima para outras quantidades de escravos:

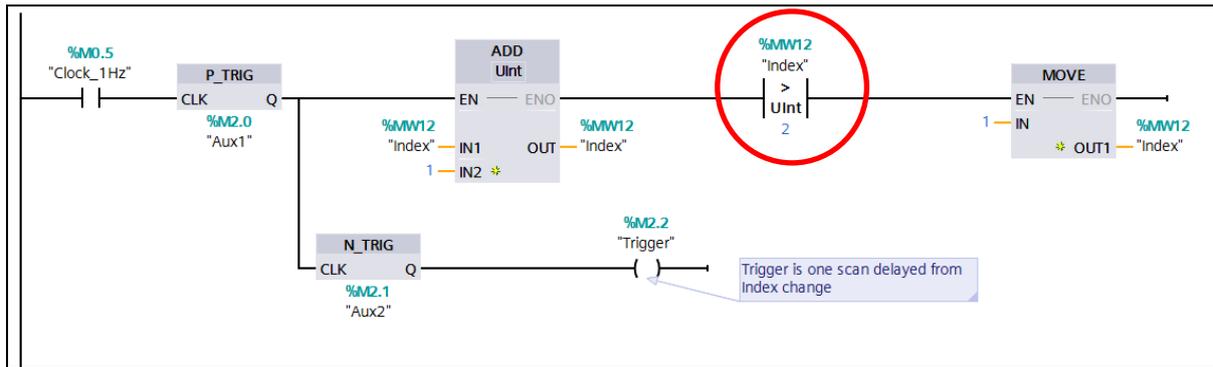


Figura 32 – A comparação destacada em vermelho deve ser alterada para um número igual ao de tarefas de comunicação a serem executadas.

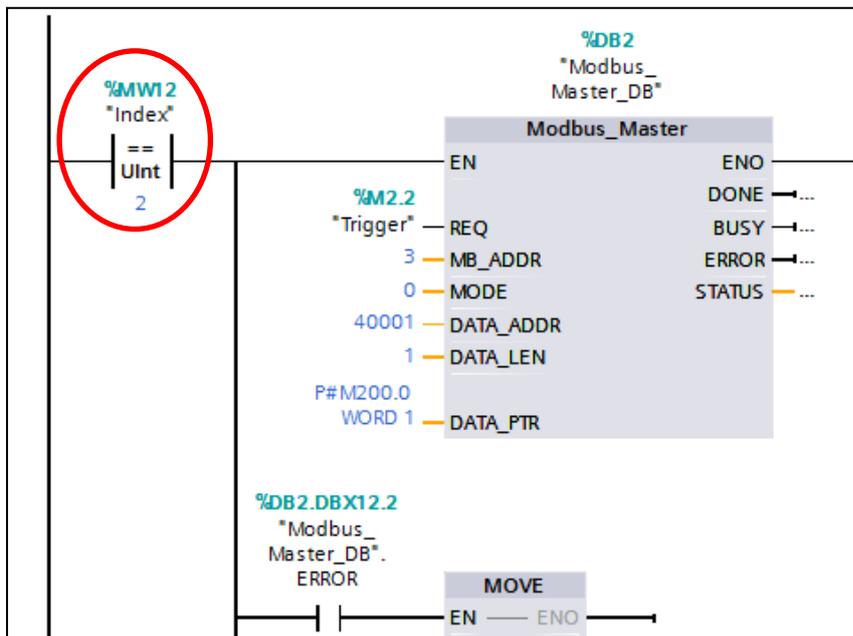


Figura 33 – Para cada nova varredura a ser executada, nova chamada do bloco deverá ser criada. Cada chamada deverá ter uma comparação do Index com um valor único. Os parâmetros do bloco Modbus_Master devem ser adaptados para cada tarefa. Variáveis de diagnóstico devem ser únicas em cada chamada também.

7. Dicas gerais de configuração

- Manter STEP 7 V1X (TIA Portal), atualizados.
- Versões mais atuais de blocos / bibliotecas, compatíveis com o HW/FW utilizado.
- Utilização correta dos projetos exemplos:
 - Os projetos exemplos, como o próprio nome diz, são exemplos. Use-os para entender como a teoria dos manuais pode ser aplicada na prática.
 - Não tenha a expectativa de que simplesmente descarregar o exemplo no controlador vai ser o suficiente para fazer sua aplicação rodar. Os exemplos precisam ser adaptados as suas condições.
 - Em algumas raras vezes, os exemplos não contém a versão mais atual dos blocos. Por isso é melhor olhar o que foi feito e realizar seu projeto copiando ideias, mas não objetos do exemplo.
- Desenvolver a aplicação em passos, partindo de uma tarefa simples e indo gradualmente para passos mais complexas:
 - Testar individualmente cada uma das partes comunicando com um simulador de MODBUS. Conferir os telegramas de cada um dos testes, para ver se coincidem. Somente após isso, unir os dois equipamentos. Isto pode ser especialmente útil para identificar os números corretos de registros a serem usados.
 - Testar primeiramente a comunicação entre o PLC e **um único** escravo. Somente teste programas lendo múltiplos equipamentos depois de ter funcionado individualmente com cada um dos deles.

8. Referências, exemplos e downloads.

Geral

- Lista de possibilidades de comunicação:
Which hardware and software components do you need to establish communication between SIMATIC S7 stations and third-party devices using the MODBUS RTU protocol?
<https://support.industry.siemens.com/cs/br/pt/view/58386780/en>
- Esquema de ligação:
How do you connect the RS485/RS422 interface of the serial modules?
<https://support.industry.siemens.com/cs/us/en/view/109736665>

S7-1200 Rack Central

- Manual do S7-1200:
SIMATIC S7 S7-1200 Programmable controller
<https://support.industry.siemens.com/cs/br/en/view/36932465>
Descrição completa dos blocos, incluindo exemplos: capítulo “13.5 Modbus communication”. Descrição de fiação: capítulo “13.2 Biasing and terminating an RS485 network connector”
- Exemplo de comunicação Mestre - Escravo com S7-1200, inclusive sugerindo como fazer o “a varredura dos escravos”:
How do you establish a MODBUS-RTU communication with STEP 7 (TIA Portal) for the SIMATIC S7-1200?
<https://support.industry.siemens.com/cs/br/en/view/47756141>

S7-1500, ET200SP e ET200MP

- Manual de CM PtP S7-1500
SIMATIC S7-1500 CM PtP RS422/485 HF
<https://support.industry.siemens.com/cs/us/en/view/59061372>
- Manual de configuração das CP ponto a ponto (S7-1500, ET200MP e ET200SP):
CM PtP - Configurations for point-to-point connections
<https://support.industry.siemens.com/cs/br/en/view/59057093>
- Exemplo de comunicação Mestre - Escravo com S7-1500, ET200MP e ET200SP,
Master Slave Communication via a CM PtP using the Modbus RTU Protocol
<https://support.industry.siemens.com/cs/br/en/view/68202723>

9. Suporte técnico e treinamento

Suporte Técnico

Requisição de suporte: www.siemens.com.br/Hotline/SR

Tel.: 0800 7 73 73 73

Portal de Suporte: www.siemens.com.br/Hotline

Homepage Brasil: www.siemens.com.br

Centro de treinamento - SITRAIN

Informações: www.siemens.com.br/sitrain

Tel.: 0800 7 73 73 73