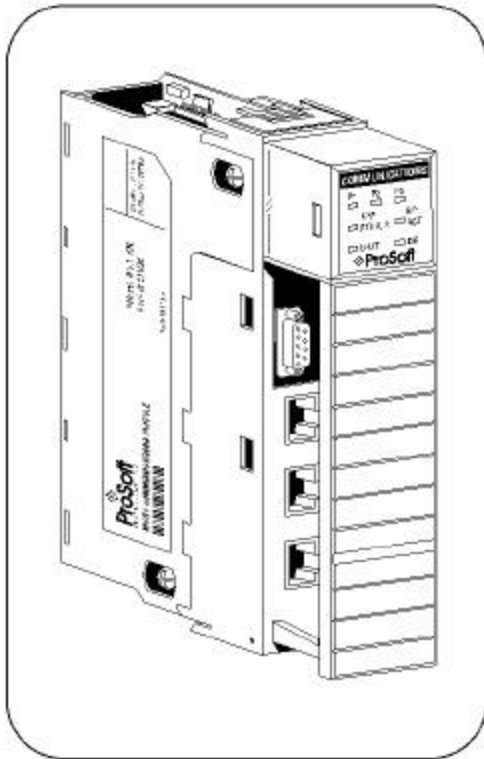


MVI56




ProSoft
TECHNOLOGY

AVISO IMPORTANTE

Uma aplicação bem sucedida deste módulo requer um razoável conhecimento do funcionamento do hardware Allen_Bradley PLC/ SLC, e da aplicação na qual a combinação será usada. Por isto, é importante que os responsáveis pela implementação garantam que a combinação atenda os requisitos da aplicação sem expor pessoal ou equipamentos a condições de operação inseguras ou não apropriadas.

Este manual pretende ajudar o usuário. Todo esforço foi feito de modo a garantir que a informação provida seja precisa e proporcione pleno entendimento dos requisitos de instalação do produto. Para garantir um pleno entendimento da operação do produto, o usuário deve ler com atenção toda documentação Allen_Bradley aplicável à operação do hardware A-B.

Em nenhuma condição, a ProSoft_Technology, Inc será passível de ser responsabilizada por danos diretos ou indiretos resultantes de uso ou aplicação de produto.

A reprodução do conteúdo deste manual, integral ou parcialmente, é expressamente proibida sem autorização por escrito da própria ProSoft_Technology, Inc

As informações contidas neste manual poderão ser modificadas ou alteradas sem prévio aviso e não representam um compromisso da parte da ProSoft_Technology, Inc e quaisquer melhoramentos ou mudanças introduzidos neste manual ou no produto poderão ser feitos em qualquer tempo. Estas mudanças serão feitas periodicamente para corrigir imprecisões técnicas ou erros tipográficos.

“This is not the latest revision. Please refer to the English version of this manual.”

ProSoft Technology, Inc.

1675 Chester Avenue

Fourth Floor

Bakersfield, CA 93301

(661) 716-5100

(661) 716-5101 Fax

www.prosoft-technology.com

ÍNDICE

- 1- Introdução**
 - 1.1 Conceitos Gerais
 - 1.2 Setup do Módulo
- 2- Entendendo a arquitetura**
 - 2.1 Loop Lógico Principal
 - 2.2 Processador ControlLogix não está em RUN
 - 2.3 Transferência de Dados pelo Barramento
 - 2.4 Transferência de Dados Normal
 - 2.4.1 Bloco de Leitura
 - 2.4.2 Bloco de Escrita
 - 2.5 Transferência de Dados de Configuração
 - 2.5.1 Dados de Configuração de Módulo
 - 2.6 Lista de Dados de Comando Mestre
 - 2.7 Blocos de Status Escravo
 - 2.8 Blocos de Controle de Comando
 - 2.8.1 Comando de Evento
 - 2.8.2 Controle de Comando
 - 2.8.3 Configuração de Escrita
 - 2.8.4 Boot A Quente
 - 2.8.5 Boot A Frio
 - 2.9 Blocos de Controle Pass_Through
 - 2.9.1 Blocos de Controle Pass_Through Não Formatados
 - 2.9.2 Blocos de Controle Pass_Through Formatados
 - 2.9.2.1 Função 5
 - 2.9.2.2 Funções 6 e 16
 - 2.9.2.3 Função 15
 - 2.9.3 Driver Escravo
 - 2.9.4 Modo de Driver Mestre
 - 2.9.4.1 Lista de Comando Mestre
- 3 Modificando a Configuração do Módulo**
 - 3.1 Energização (Power up)
 - 3.2 Mudando parâmetros durante a operação
 - 3.3 Setup do módulo
 - 3.4 Objeto de Dados do Módulo (MCMModuleDef)
 - 3.4.1 Objetos de Configuração
 - 3.4.1.1 Parâmetros de Transferência de Dados (MCMModule)
 - 3.4.1.2 Parâmetro de Porta Modbus (MCMPort)
 - 3.4.1.3 Comandos Mestre Modbus (MCMCMd)
 - 3.4.2 Objeto de Status (MCMInStat)
 - 3.5 Objetos de Dados de Usuário
 - 3.6 Controle e Status de Varredura de Escravo
 - 3.7 Dados de Mensagem Modbus
- 4 Modificando Lógica Ladder de Exemplo**
 - 4.1 Energização (Power up)
 - 4.2 MainRoutine
 - 4.3 ReadData
 - 4.4 WriteData

5 Diagnóstico e Solução de Problemas

- 5.1.1 Lendo Dados de Status do Módulo
- 5.1.2 Software Requerido
- 5.1.3 Usando a Porta
- 5.1.4 Opções de Menu
 - 5.1.4.1 A= Analisador de Dados
 - 5.1.4.1.1 1= Seleção de Porta 1
 - 5.1.4.1.2 2= Seleção de Porta 2
 - 5.1.4.1.3 5= Tiques de 1mSeg
 - 5.1.4.1.4 6= Tiques de 5 mSeg
 - 5.1.4.1.5 7= Tiques de 10mSeg
 - 5.1.4.1.6 8= Tiques de .50 mSeg
 - 5.1.4.1.7 9= Tiques de 100 mSeg
 - 5.1.4.1.8 0= Tiques de 0 mSeg
 - 5.1.4.1.9 H= Formato Hex
 - 5.1.4.1.10 A= Formato ASCII
 - 5.1.4.1.11 B= Iniciar
 - 5.1.4.1.12 S= Parar
 - 5.1.4.1.13 M= Menu Principal
 - 5.1.4.2 B= Estatística de Transferência de Blocos
 - 5.1.4.3 C=Componente Configuração de Módulo
 - 5.1.4.4 D= Visualização de Banco de Dados Modbus
 - 5.1.4.5 0-9 Registros páginas 0-9000
 - 5.1.4.6 S= Mostre Novamente
 - 5.1.4.6.1 -= Volte 5 Páginas
 - 5.1.4.6.2 P= Página Anterior
 - 5.1.4.6.3 += Avança 5 Páginas
 - 5.1.4.6.4 N= Próxima Página
 - 5.1.4.6.5 D= Tela Decimal
 - 5.1.4.6.6 H= Tela Hexadecimal
 - 5.1.4.6.7 F= Tela Flutuante
 - 5.1.4.6.8 A= Tela ASCII
 - 5.1.4.6.9 M= Menu Principal
 - 5.1.4.7 E e F = Erros de Comando Mestre (portas 1 e 2)
 - 5.1.4.7.1 S= Mostre Novamente
 - 5.1.4.7.2 -= Volte 2 Páginas
 - 5.1.4.7.3 P= Página anterior
 - 5.1.4.7.4 += Avança 2 Páginas
 - 5.1.4.7.5 N= Próxima Página
 - 5.1.4.7.6 D= Tela Decimal
 - 5.1.4.7.7 H= Tela Hexadecimal
 - 5.1.4.7.8 M= Menu Principal
 - 5.1.4.8 I e J = Lista de Comando mestre (Portas 1 e 2)
 - 5.1.4.7.8.1 S= Mostre Novamente
 - 5.1.4.7.8.2 -= Volte 5 Páginas
 - 5.1.4.7.8.3 P= Página Anterior
 - 5.1.4.7.8.4 += Avança 5 Páginas
 - 5.1.4.7.8.5 N= Próxima Página
 - 5.1.4.7.8.6 M= Menu Principal
 - 5.1.4.9 O e P = Status de Escravo (Portas 1 e 2)
 - 5.1.4.10 V= Informação de Versão
 - 5.1.4.11 W= Módulo de Boot A Quente
 - 5.1.4.12 Y= Módulo de Transferência Cfg para Processador
 - 5.1.4.13 1 e 2= status de Comunicação (Portas 1 e 2)
 - 5.1.4.14 6 e 7 = Configuração de Porta (Portas 1 e 2)
 - 5.1.4.15 Esc= Programa de Saída

- 5.2 Indicadores de Status de LED
 - 5.2.1 Eliminando Condição de Falha
 - 5.2.2 Solução de Problemas

6 Conexões de Cabo

- 6.1 Portas de Comunicação Modbus
 - 6.1.1 Conectando Cabo no Conector
 - 6.1.1.1 RS_232
 - 6.1.1.2 RS_485
 - 6.1.1.3 RS_422
 - 6.1.2 Ajustando Jumpers
- 6.2 Porta Configuração/ Debug RS_232

Apêndice A - Definição de Banco de Dados MVI56-MCM

Apêndice B - Definição de Dados de Status MVI56-MCM

Apêndice C - Definição de Dados de Configuração MVI56-MCM

- Setup de Barramento
- Setup da Porta 1
- Setup da Porta 2
- Comandos da Porta 1
- Comandos da Porta 2
- Status de Misc
- Controle de Comando

Apêndice D – Controle de Comando MVI56-MCM

Apêndice E – Especificações de Produto

Especificações Gerais

- Especificações Funcionais de Escravo
- Especificações Funcionais de Mestre Modbus
- Físico
- Interface ControlLogix
- Especificações de Hardware

Suporte, Serviço, e Garantia

1 Introdução

O produto MVI56-MCM (“Módulo de Comunicação Modbus”) permite a processadores compatíveis E/S Allen_Bradley ControlLogix a facilmente interfacearem outros dispositivos compatíveis com protocolo Modbus. Dispositivos compatíveis incluem não apenas PLC’s Modicon (todos suportando Modbus), mas também uma ampla variedade de dispositivos terminais.

O módulo MVI56_MCM atua como gateway entre a rede Modbus e o barramento Allen_Bradley. A transferência de dados a partir do processador ControlLogix é assíncrona em relação às ações na rede Modbus. Um espaço de registro de 5000 palavras no módulo é usado para trocar dados entre processador e rede Modbus.

1.1 Conceitos Gerais

A discussão acima cobre diversos conceitos chave para entender a operação do módulo MVI56-MCM

Na energização o módulo começa a realizar as seguintes funções lógicas:

1. Inicializa componentes de hardware
 - a. Inicializa driver de barramento ControlLogix
 - b. Testa e limpa toda RAM
 - c. Inicializa portas de comunicação lógicas
2. Aguarda por configuração do módulo a partir do processador ControlLogix
3. Inicializa espaço de Registro de Módulo
4. Habilita Driver Mestre em portas selecionadas
5. Habilita Driver Escravo em portas selecionadas.

Uma vez que o módulo tenha recebido o Bloco de Configuração de módulo do processador, o módulo começa a se comunicar com outros nós na rede, dependendo da configuração.

1.2 Set up do Módulo

Uma vez instalado o módulo, você pode começar o processo de modificar configuração mesmo e da Lógica Ladder. Antes de começar este processo, você deve entender a arquitetura, mostrada na próxima seção. As demais seções explicam como fazer modificações no arquivo existente **.cfg** e na Lógica Ladder de exemplo.

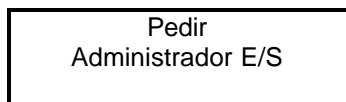
2 Entendendo a Arquitetura

Esta seção dá ao leitor uma visão geral funcional do módulo MVI56-MCM. Os detalhes associados à memória Ladder e ao mapa de memória não serão cobertos nesta seção (dirija-se à seção Setup do módulo). Um entendimento completo da informação contida neste documento é requerido para uma implementação bem sucedida do módulo para uma aplicação de usuário. Se você entendeu o conteúdo desta seção, dirija-se à seção Setup de Módulo para fazer o setup do módulo e colocá-lo em funcionamento. Se você ainda não está familiarizado com operações de protocolo Modbus e transferência de dados, leia este documento antes de fazer o setup do módulo.

2.1 Loop Lógico Principal

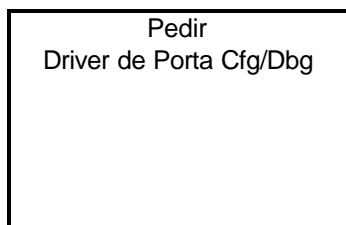
Ao completar o processo de configuração de energização, o módulo entra em loop infinito, realizando as seguintes funções

A partir da lógica de energização



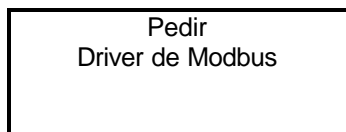
Pedir Operador E/S

Transfere dados entre módulo e processador (usuário, status, configuração, etc.)



Pedir Driver de Porta Cfg/Dbg (Configuração/ Debug)

- Rotinas de buffer Rx e Tx são acionadas por interrupção
- Pedir às rotinas de porta serial verificarem se há algum dado no buffer e dependendo do valor, atender o buffer ou aguardar mais caracteres



Pedir Driver Plus Modbus

Se Porta Modbus Mestre, varrer escravos usando lista de comando
Se Porta escravo Modbus, responder aos comandos recebidos

2.2 Processador ControlLogix não se encontra em RUN

A qualquer momento que o módulo detecte que o processador saiu do modo RUN (i.e., em caso de Falha ou PGM) as portas Modbus podem ser fechadas, como prescrito na configuração de usuário. Quando o processador volta para o estado RUN, o módulo reinicia comunicação na rede.

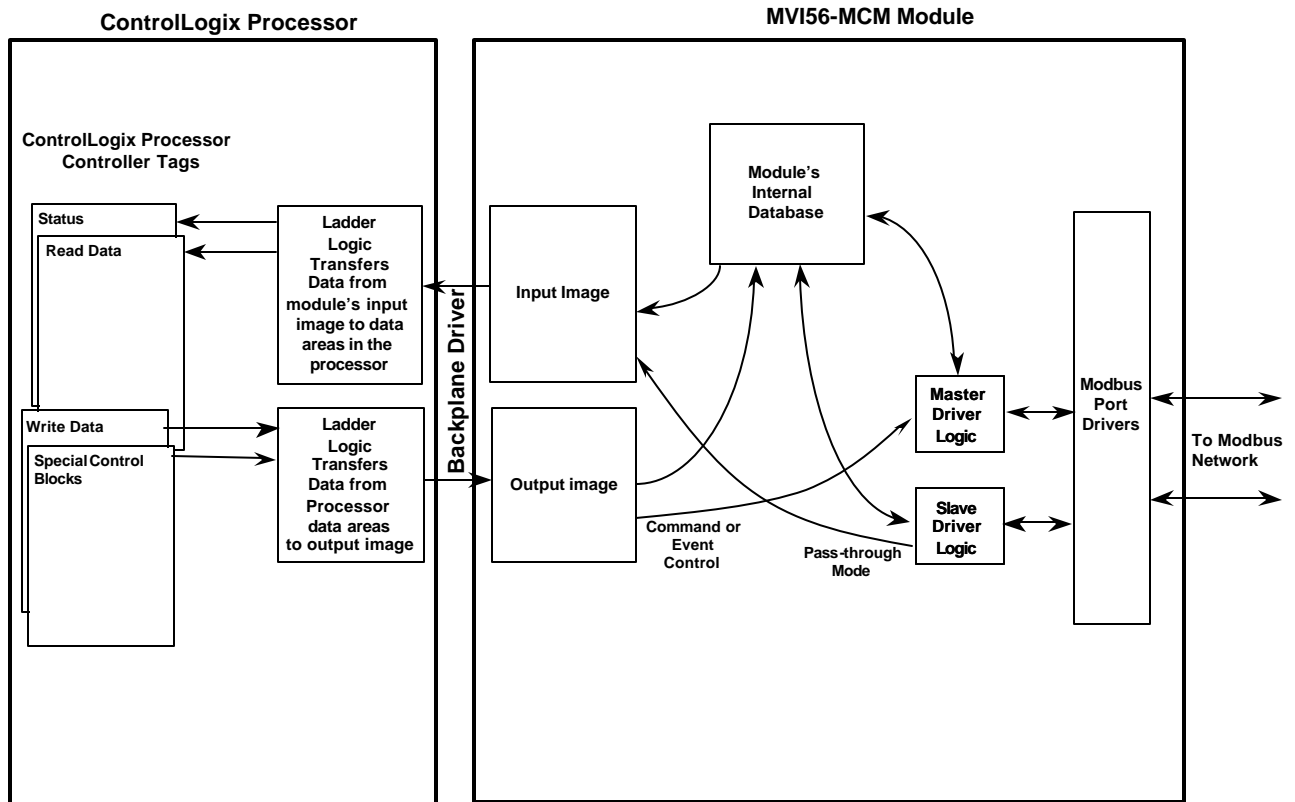
2.3 Transferência de Dados pelo Barramento

O módulo MVI56-MCM é único no qual o barramento ControlLogix é utilizado. O Dado é paginado entre módulo e o processador ControlLogix através do barramento usando as imagens de entrada e saída do módulo. A frequência de atualização da imagem é determinada pela razão de escaneamento programada definida pelo usuário para o módulo e a carga de comunicação no módulo. Atualizações típicas se encontram entre 2 a 10 milissegundos.

Esta transferência bidirecional de dados é realizada pelo módulo preenchendo dados na imagem de entrada de módulo a ser enviada para o processador. O dado na imagem de entrada é colocado em Tags Controladores no processador pela Lógica Ladder. A imagem de entrada para o módulo é ajustada em 250 palavras. Atualizações típicas permitem transferir dados rapidamente entre módulo e processador.

O processador insere dados para imagem de saída de módulo a ser transferida para o módulo. O programa do módulo extrai os dados e os coloca no banco de dados interno do módulo. A imagem de saída do módulo é ajustada em 248 palavras. Esta ampla área de dados permite uma rápida transferência de dados do processador para o módulo.

O diagrama a seguir mostra o método de transferência de dados usado para mover dados entre processador ControlLogix, módulo MVI56-MCM, e rede Modbus.

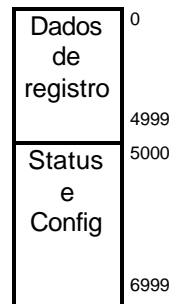


Como mostrado no diagrama acima, todos os dados são transferidos entre módulo e processador pelo barramento através das imagens de saída. A Lógica ladder deve ser escrita no processador ControlLogix para interfacear os dados de imagem de entrada e saída com dados definidos nos Tags Controladores. Todos dados usados pelo módulo são armazenados no banco de dados interno. Este banco de dados é definido como tabela de dados Modbus virtual com endereços de 0 (Modbus 40001) a 6999 (Modbus 47000). O diagrama abaixo mostra layout do banco de dados.

Estrutura de Banco de Dados Interno do Módulo

5000 registros para dados de usuário

2000 palavras de configuração e dados de status



Os dados contidos neste banco de dados são paginados nas imagens de entrada e saída por coordenação da Lógica Ladder do ControlLogix e programa do módulo MVI56-MCM.

Até 248 palavras de dados podem ser transferidas do módulo para o processador de uma vez. Até 247 palavras podem ser transferidas do processador para o módulo. Cada imagem tem uma estrutura definida dependendo do conteúdo de dados e da função da transferência de dados como definido abaixo.

2.4 Transferência Normal de Dados

A transferência normal de dados inclui paginação dos dados de usuário encontrada no banco de dados interno do módulo em registros 0 a 4999 e dados de status. Estes dados são transferidos através de blocos de leitura (imagem de entrada) e escrita (imagem de saída). Dirija-se à seção “**Setup do Módulo**” onde pode ser encontrada a descrição de objetos de dados usados com os blocos e a Lógica Ladder. Estrutura e função de cada bloco serão discutidas abaixo.

2.4.1 Bloco de Leitura

Estes blocos de dados são usados para transferir dados do módulo para o processador ControlLogix. A estrutura da imagem de entrada usada para transferir estes dados se encontra na tabela abaixo:

Offset	Descrição	Comprimento
0	Reservado	1
1	ID do Bloco de Escrita	1
2-201	Dado de Leitura	200
202	Contador de Escaneamento de Programa	1
203-204	Código de Produto	2
205-206	Versão de Produto	2
207-208	Sistema Operacional	2
209-210	Número de RUN	2
211-217	Status de Erro da Porta 1	7
218-224	Status de Erro da Porta 2	7
225-230	Status de Transferência de Erro	6
231	Índice/ Erro Corrente da Porta 1	1
232	Índice/ Último Erro da Porta 1	1
233	Índice/ Erro Corrente da Porta 2	1
234	Índice/ Último Erro da Porta 2	1
235-248	Reserva	14
249	ID do Bloco de Leitura	1

O ID do bloco de Leitura é um valor de índice usado para determinar a localização de onde o dado será colocado no arranjo de Tag controlador de processador ControlLogix do dado de leitura do módulo. Cada transferência pode mover até 200 palavras (Offsets de bloco 2 a 201). Adicionalmente a mover os dados de usuário, o bloco também contém dados de status para o módulo. Este último jogo de dados é transferido com cada novo bloco de dado e é usado para um movimento de dados em alta velocidade.. O ID do Bloco de Escrita associado ao bloco é usado para pedir dados do processador ControlLogix. Em operação normal do programa, o módulo envia seqüencialmente blocos de leitura e pede blocos de escrita. Por exemplo, se três blocos de leitura e dois blocos de escrita forem usados com a aplicação, a seqüência será a seguinte:

R1W1 ↯ R2W2 ↯ R3W1 ↯ R1W2 ↯ R2W1 ↯ R3W2 ↯ R1W1 ↯

A seqüência continua até ser interrompida por outro bloco de escrita enviada pelo controlador ou por um pedido de comando a partir de um nó na rede Modbus ou controle do operador através da porta Configuração/Debug do módulo.

2.4.2 Bloco de Escrita

Estes blocos de dados são usados para transferir informação do processador ControlLogix para o módulo. A estrutura da imagem de saída usada para transferir dados está mostrada na tabela abaixo:

Offset	Descrição	Comprimento
0	ID do Bloco de Escrita	1
1-200	Dado de Escrita	200
201-247	Reserva	47

O ID do Bloco de Escrita é um valor de índice usado para determinar a localização no banco de dados do módulo onde o dado será colocado. Cada transferência pode mover até 250 palavras (offsets 1 a 200) de dados.

2.5 Transferência de Dados de Configuração

Quando o módulo realiza uma operação reiniciar, será solicitada informação de configuração a partir do processador ControlLogix. Este dado será transferido para o módulo em blocos de escrita especialmente formatados (imagem de saída). O módulo **varre** cada bloco estabelecendo o número de bloco de escrita requerido. Dirija-se à seção **Setup de Módulo** quanto à descrição dos objetos de dados usados com os blocos e a lógica ladder requerida. O formato dos blocos para configuração será dado nas seções que se seguem.

2.5.1 Dados de Configuração de Módulo

Este bloco é usado para enviar informação de configuração geral do processador para o módulo. O dado é transferido em um bloco com um código de identificação. A estrutura do bloco está mostrada na tabela abaixo:

Offset	Descrição	Comprimento
0	9000	1
1-6	Setup do Barramento	6
7-31	Configuração da Porta 1	25
32-56	Configuração da Porta 2	25
57-59	Configuração Auxiliar da Porta 1	3
60-62	Configuração Auxiliar da Porta 2	3
63-247	Reserva	185

O bloco de leitura usado para pedir configuração tem a seguinte estrutura.

Offset	Descrição	Comprimento
0	Reservado	1
1	9000	1
2	Erros de Configuração de Módulo	1
3	Erros de configuração de Porta 1	1
4	Erros de Configuração de Porta 2	1
5-248	Reserva	244
249	-2 ou -3	1

Se houver qualquer erro na configuração, o bit associado ao erro será colocado em uma das três palavras de erro de configuração. O erro deve ser corrigido antes de o módulo iniciar modo normal de operação

2-6 Lista de Dados de Comando Mestre

Cada porta no módulo pode ser configurada como dispositivo mestre Modbus contendo a própria lista de cem comandos. Os comandos são lidos a partir do processador usando os seguintes IDs dos Blocos de Escrita: Porta Modbus 1 de 6000 a 60003; Porta Modbus 2 de 6100 a 6103. O módulo seqüencialmente varre cada bloco a partir do processador. A Lógica Ladder deve ser escrita para lidar com cada uma das transferências de dados. A estrutura de cada bloco pode ser vista na tabela abaixo:

Offset	Descrição	Comprimento
0	6000 a 60003 e 6100 a 6103	1
1-8	Definição de Comando	8
9-16	Definição de Comando	8
17-24	Definição de Comando	8
25-32	Definição de Comando	8
33-40	Definição de Comando	8
41-48	Definição de Comando	8
49-56	Definição de Comando	8
57-64	Definição de Comando	8
65-72	Definição de Comando	8
73-80	Definição de Comando	8
81-88	Definição de Comando	8
89-96	Definição de Comando	8
97-104	Definição de Comando	8
105-112	Definição de Comando	8
113-120	Definição de Comando	8
121-128	Definição de Comando	8
129-136	Definição de Comando	8
137-144	Definição de Comando	8
145-152	Definição de Comando	8
153-160	Definição de Comando	8
161-168	Definição de Comando	8
169-176	Definição de Comando	8
177-184	Definição de Comando	8
185-192	Definição de Comando	8
193-200	Definição de Comando	8

2.7 Blocos de Status de Escravo.

Blocos de Status de Escravo são usados para enviar informação de status de cada dispositivo escravo em uma porta mestre. Escravos anexos à porta mestre podem estar nos seguintes estados:

0	O escravo está inativo e não definido na Lista de Comando para porta Mestre.
1	O escravo está ativamente em varredura ou sendo controlado pela porta mestre e a comunicação é bem sucedida.
2	A porta mestre não conseguiu estabelecer comunicação com o dispositivo escravo. As comunicações com o escravo são suspensas por um período definido pelo usuário com base no escaneamento da Lista de Comando.
3	Comunicações com o escravo foram desabilitadas pela Lógica Ladder. Nenhuma comunicação deverá ocorrer com o escravo até que este estado seja cancelado pela Lógica Ladder.

Escravos são definidos para o sistema, quando o módulo inicializa a Lista de Comando Mestre. Cada escravo definido será colocado no estado de um nesta etapa inicial. Se a porta mestre não estabelecer comunicação com um dispositivo escravo (quantidade de repetição terminada com comando), o mestre coloca o estado do escravo no valor 2 na tabela de estado. Isto suspende comunicação com dispositivo escravo de acordo com a quantidade de escaneamento especificada por usuário (valor **ErrorDelayCntr** no objeto **MCMPort** para cada porta). Cada vez que é escaneado um comando na lista, com endereço de um escravo suspenso, o valor do contador de atraso será decrementado. Quando este valor chegar a zero, o estado do escravo será colocado em zero. Isto habilita varredura no escravo.

ID do Bloco	Descrição
3002	Pedido para os primeiros 128 valores de status de escravo para a Porta 1
3003	Pedido para os primeiros 128 valores de status de escravo para a Porta 1
3102	Pedido para os primeiros 128 valores de status de escravo para a Porta 2
3103	Pedido para os primeiros 128 valores de status de escravo para a Porta 2

O formato destes blocos está mostrado na tabela abaixo:

Offset	Descrição	Comprimento
0	3002- 3003 ou 3102- 3103	1
1-247	Reserva	246

O módulo reconhecerá o pedido recebendo o código de bloco de escrita especial e responderá com um bloco de escrita com o seguinte formato:

Offset	Descrição	Comprimento
0	Reservado	1
1	ID do Bloco de Escrita	1
2-129	Dado de status de varredura de Escravo	128
130-248	Reserva	119
249	3002- 3003 ou 3102 - 3103	1

A Lógica Ladder pode ser escrita de modo a sobrepor o valor na tabela de status de escravo, e pode desabilitar (valor de estado 3) enviando um bloco especial de dado a partir do processador para o escravo. Os escravos da Porta 1 são desabilitados com bloco 3000 e os escravos da Porta 2 são desabilitados com bloco 3100. Cada bloco contém os endereços de nós de escravo de desabilitar. A estrutura do bloco está mostrada abaixo:

Offset	Descrição	Comprimento
0	3000 ou 3100	1
1	Número de Escravos no Bloco	1
2-201	Índice de escravo	200
202-247	Reserva	46

O módulo responderá com um bloco usando o mesmo código de identificação recebido e indicará o número de escravos que foi atuado com o bloco. O formato deste bloco de resposta está mostrado abaixo:

Offset	Descrição	Comprimento
0	Reservado	1
1	ID do Bloco de Escrita	1

2	Número de Escravos processados	1
3-248	Reserva	246
249	3000 ou 3100	1

A Lógica Ladder pode ser escrita de modo a sobrepor o valor na tabela de status de escravo para habilitar o escravo (valor de estado 1) enviando um bloco especial. Os escravos da porta 1 são habilitados usando bloco 3001, e os escravos da porta 2 são habilitados usando bloco 3101. Cada bloco contém os endereços de nó de escravo. O formato do bloco está mostrado abaixo:

Offset	Descrição	Comprimento
0	3001- ou 3101	1
1	Número de Escravos no Bloco	1
2-201	Índice de Escravo	200
202-247	Reserva	46

O módulo responderá com um bloco com o mesmo código de identificação recebido e indicará o número de escravos que foram atuados com o bloco. O formato deste bloco de resposta está mostrado abaixo::

Offset	Descrição	Comprimento
0	Reservado	1
1	ID do Bloco de Escrita	1
2	Número de Escravos processados	1
3-248	Reserva	246
249	3001 ou 3101	1

2.8 Blocos de Controle de Comando

Os blocos de controle e comando são blocos especiais usados para controlar o módulo ou dados especiais de pedido a partir do módulo. A versão corrente do software suporta cinco blocos de controle de comando, a saber: controle de comando de evento, controle de comando, configuração de escrita, boot a quente, e boot a frio.

2.8.1 Comando de Evento

Blocos de comando de evento são usados para enviar comandos diretamente da Lógica Ladder a uma das portas mestre. O formato destes blocos está mostrado abaixo:

Offset	Descrição	Comprimento
0	1000-1255 ou 2000-2255	1
1	Endereço BD Interno	1
2	Quantidade de Pontos	1
3	Código Swap	1
4	Código "Modbus Function"	1
5	Endereço de Banco de Dados do Dispositivo	1
6-247	Reserva	242

O número de bloco define a porta Modbus a ser considerada e o nó de escravo a ser acessado. Os blocos na gama 1000 são direcionados para a Porta 1 de Modbus, e blocos na gama 2000 são direcionados para a Porta Modbus 2. O endereço de escravo é representado no número de bloco na gama 0 a 255. A soma destes dois valores determina o número de bloco. Os outros parâmetros passados com o bloco são usados para construir o comando. O parâmetro **endereço BD Interno**, especifica a localização do banco de dados do módulo para associar ao comando. **"Quantidade de Pontos"** (Point Count) define o número de pontos ou registros para o comando. O **código Swap** é usado com pedidos de função Modbus 3 para mudar ordem de palavra ou byte. O **código "Modbus Function"** assume um dos seguintes valores 1, 2, 3, 4, 5, 6, 15, ou 16. **Endereço do Banco de Dados do Dispositivo** é o registro ou ponto no dispositivo escravo

remoto a ser associado ao comando. Quando o comando recebe o bloco ele o processará e o colocará na fila de comando. O módulo responderá a cada bloco de comando de evento com bloco de leitura tendo o seguinte formato

Offset	Descrição	Comprimento
0	Reservado	1
1	ID do Bloco de Escrita	1
2	0= Falha; 1= Sucesso	1
3-248	Reserva	246
249	1000-1255 ou 2000-2255	1

A palavra dois do bloco pode ser usada pela Lógica Ladder para determinar se o comando foi adicionado à fila de comando do módulo. O comando somente falhará, se a fila do comando da porta estiver completa (100 comandos para cada fila).

2.8.2 Controle de Comando

Os blocos de controle de comando são usados para colocar comandos na Lista de Comando da fila de comando. Cada porta tem uma fila de comando de até 100 comandos. O módulo serve comandos na fila antes da Lista de Comando Mestre. Isto confere uma alta prioridade aos comandos na fila. Os comandos colocados na fila com este mecanismo devem ser definidos na Lista de Comando Mestre. Na execução normal da Lista de Comando, o módulo somente executa comandos com o parâmetro Habilitado em um ou dois. Se o valor for colocado em zero, o comando será desconsiderado. Comandos podem ser colocados na Lista de Comando com parâmetro Habilitado colocado em zero. Estes comandos então podem ser executados usando blocos de controle de comando.

De um a seis comandos podem ser colocados na fila de comando com um único pedido. O formato do bloco está mostrado na tabela abaixo:

Offset	Descrição	Comprimento
0	5001-5006 ou 5101-5106	1
1	Índice de Comando (MCM.P1.CMD [valor de índice de comando])	1
2	Índice de Comando (MCM.P1.CMD [valor de índice de comando])	1
3	Índice de Comando (MCM.P1.CMD [valor de índice de comando])	1
4	Índice de Comando (MCM.P1.CMD [valor de índice de comando])	1
5	Índice de Comando (MCM.P1.CMD [valor de índice de comando])	1
6	Índice de Comando (MCM.P1.CMD [valor de índice de comando])	1
7-247	Reserva	241

Blocos na gama 5001-5006 são usados para a Porta Modbus 1, e blocos na gama 5101 a 5106 são usados para a Porta Modbus 2. O último dígito no código de bloco define o número de comandos a processar no bloco. Por exemplo, um código de bloco 5003 contém 3 índices de comando que devem ser usados com a Porta Modbus 1. Os parâmetros de índice de Comando no bloco têm uma gama de 0 a 99 e correspondem às entradas de Lista de Comando Mestre.

O módulo responde a um bloco de controle de comando com um bloco tendo o número de comandos adicionados à fila de comando para a porta. O formato do bloco está mostrado abaixo:

Offset	Descrição	Comprimento
0	Reservado	1
1	ID do bloco de Escrita	1
2	Número de comandos adicionados à fila de comando	1
3-248	Reserva	246
249	5000-5006 ou 5100 -5106	1

2.8.3.1.1 Configuração de Escrita

O bloco é enviado a partir do processador ControlLogix para o módulo de modo a forçar o módulo a escrever sua configuração corrente para o processador. Esta função é usada quando a configuração do módulo tiver sido alterada remotamente usando operações de escrita do banco de dados. O módulo responderá com blocos contendo dados de configuração do módulo. A Lógica Ladder deve ser escrita para lidar com a recepção destes blocos. Os blocos transferidos dos módulos são os seguintes:

Bloco – 9000, Dados de Configuração Geral

Offset	Descrição	Comprimento
0	Reservado	1
1	-9000	1
2-7	Setup do Barramento	6
8-32	Configuração da Porta 1	25
33-57	Configuração da Porta 2	25
58-60	Configuração da Porta 1 Aux	3
61-63	Configuração da Porta 2 Aux.	3
64-248	Reserva	185
249	-9000	1

Blocos -6000 a -6003 e -6100 a 6103, Dados da Lista de Comando Mestre para Portas 1 e 2, respectivamente:

Offset	Descrição	Comprimento
0	Reservado	1
1	-6000 a -6003 e -6100 a -6103	1
2-9	Definição de Comando	8
10-17	Definição de Comando	8
18-25	Definição de Comando	8
26-33	Definição de Comando	8
34-41	Definição de Comando	8
42-49	Definição de Comando	8
50-57	Definição de Comando	8
58-65	Definição de Comando	8
66-73	Definição de Comando	8
74-81	Definição de Comando	8
82-89	Definição de Comando	8
90-97	Definição de Comando	8
98-105	Definição de Comando	8
106-113	Definição de Comando	8
114-121	Definição de Comando	8
122-129	Definição de Comando	8
130-137	Definição de Comando	8
138-145	Definição de Comando	8

146-153	Definição de Comando	8
154-161	Definição de Comando	8
162-169	Definição de Comando	8
170-177	Definição de Comando	8
178-185	Definição de Comando	8
186-193	Definição de Comando	8
194-201	Definição de Comando	8
202-248	Definição de Comando	47
249	-6000 a -6003 e -6100 a -6103	1

Cada bloco deve ser trabalhado pela Lógica Ladder para operação apropriada do módulo.

2.8.4 Boot a Quente

Este bloco é enviado do processador ControlLogix para o módulo (imagem de saída) quando se requer que o módulo realize uma operação de boot a quente. (reinicialização de software). Este bloco em geral é enviado para o módulo sempre que modificações de dados de configuração são feitas na área de dados dos Tags Controladores. Isto força o módulo a ler uma nova informação de configuração e reiniciar. A estrutura do bloco de controle está mostrada abaixo:

Offset	Descrição	Comprimento
0	9998	1
1-247	Reserva	247

2.8.5 Boot a Frio

Este bloco é enviado do processador ControlLogix para o módulo (imagem de saída) quando se requer que o módulo realize operação de boot a frio (reinicialização de hardware). Este bloco é enviado para o módulo quando a Lógica Ladder detecta um problema de hardware que requeira reinicialização do hardware. A estrutura do bloco de controle está mostrada abaixo:

Offset	Descrição	Comprimento
0	9999	1
1-247	Reserva	247

2.9 Blocos de Controle Pass_Through

2.9.1 Blocos de Controle Pass_Through Não Formatado

Se uma ou mais portas escravo no módulo estiverem configuradas no modo Pass_Through Não-Formatado de operação, o módulo passa blocos com códigos de identificação 9996 para o processador a cada comando de escrita recebido. Qualquer comando de função Modbus 5, 6, 15 e 16 será passado da porta para o processador usando este número de identificação de bloco. A Lógica Ladder deve ser escrita para lidar com a recepção de todas funções de escrita Modbus para o processador e responder como esperado aos comandos emitidos pelo dispositivo mestre Modbus remoto. A estrutura do bloco de controle Pass_Through Não-Formatado está mostrada abaixo:

Offset	Descrição	Comprimento
0	0	1
1	996	1
2	Número de bytes na mensagem Modbus	1
3-248	Mensagem Modbus recebida	246
249	9996	1

A Lógica Ladder deve copiar de forma subdividida a mensagem recebida e controlar o processador como esperado pelo dispositivo mestre. O processador deve responder ao bloco de controle Pass_Through com um bloco de escrita no seguinte formato:

Offset	Descrição	Comprimento
0	996	1
1-247	Reserva	247

Isto informa ao módulo que o comando foi processado e pode ser eliminado da fila Pass_Through.

2.9.2 Blocos de Controle Pass_Through Formatado

Se uma ou mais portas escravo forem configuradas para o modo Pass_Through Formatado de operação, o módulo passará blocos com códigos de identificação de 9996 para o processador a cada comando de escrita recebido. Qualquer comando de função Modbus 5, 6, 15, 16 será passado da porta para o processador usando este número de identificação de bloco. A Lógica Ladder deve ser escrita de modo a lidar com recepção de todas funções de escrita Modbus para o processador e responder como esperado aos comandos emitidos pelo dispositivo mestre Modbus. A estrutura do bloco de controle Pass_Through Formatado está mostrada abaixo:

2.9.2.1 Função 5

Offset	Descrição	Comprimento
0	0	1
1	9958	1
2	Endereço de Bit	1
3	1	1
4	Dado	1
5-248	Mensagem Modbus recebida	244
249	9958	1

A Lógica Ladder deve copiar de forma subdividida a mensagem recebida e controlar o processador como esperado pelo dispositivo Mestre. O processador deve responder ao bloco de controle Pass_Through com um bloco de escrita com o seguinte formato:

Offset	Descrição	Comprimento
0	958	1
1-247	Reserva	247

Isto informa ao módulo que o comando foi processado e pode ser eliminado da fila Pass_Through.

2.9.2.2 Funções 6 e 16

Offset	Descrição	Comprimento
0	996	1
1	9956/ 9957 (Ponto Flutuante)	1
2	Número de Palavras de dados	1
3	Endereço de Dado	1
4-248	Dado	244
249	9956/ 9957	1

A Lógica Ladder deve processar copiar de forma subdividida mensagem recebida e controlar o processador como esperado pelo dispositivo mestre. O processador deve responder ao bloco de controle Pass_Through com um bloco de escrita com o seguinte formato:

Offset	Descrição	Comprimento
0	9956/ 9957	1
1-247	Reserva	247

Isto informará ao módulo que o comando foi processado e pode ser eliminado da fila Pass_Through.

2.9.2.3 Função 15

Quando o módulo recebe um código de função 15, quando no modo Pass_Through, o módulo escreverá o dado usando ID do bloco 9959 para dados de múltiplos bits. Em primeiro lugar, a máscara de bit é usada para eliminar os bits a serem atualizados. Isto é realizado acrescentando (AND_{ing}) a máscara invertida com o dado existente. A seguir, o novo dado acrescentado (AND_{ed}) com máscara é colocado como alternativa (OR_{ed}) para o dado existente. Isto protege os outros bits nos registros INT de serem afetados.

Offset	Descrição	Comprimento
0	0	1
1	9959	1
2	Número de palavras	1
3	Endereço de palavra	1
4-53	Dado	50
54-103	Máscara	50
104-248	Reserva	145
249	9959	1

A Lógica Ladder deve copiar de forma subdividida a mensagem recebida e controlar o processador como esperado pelo dispositivo mestre. O processador deve responder ao bloco de controle Pass_Through com um bloco de escrita com o seguinte formato:

Offset	Descrição	Comprimento
0	9959	1
1-247	Reserva	247

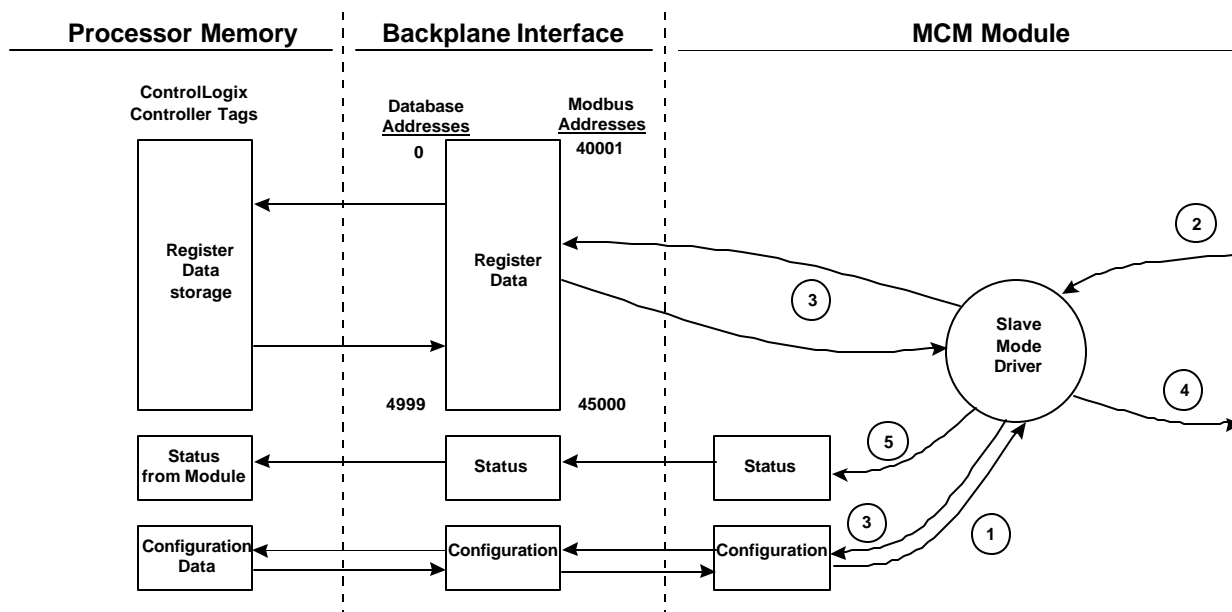
Isto informará ao módulo que o comando foi processado e pode ser eliminado da fila Pass_Through.

Fluxo de Dados entre Módulo MVI56-MCM e processador ControlLogix

A discussão a seguir detalha o fluxo de dados entre duas peças de hardware (processador ControlLogix e módulo MVI56-MCM) e outros nós da rede Modbus em diferentes modos de operação do módulo. Cada porta no módulo é configurada para emular um dispositivo Mestre Modbus ou dispositivo escravo Modbus. A operação de cada porta depende desta configuração. A seção abaixo trata a operação de cada modo.

2.9.3 Driver Escravo

O modo driver Escravo permite ao módulo MVI56-MCM responder aos comandos de leitura e escrita de dados emitidos por um mestre na rede Modbus. O fluxograma tabela associada abaixo detalham o fluxo de dados que entram e saem do módulo.

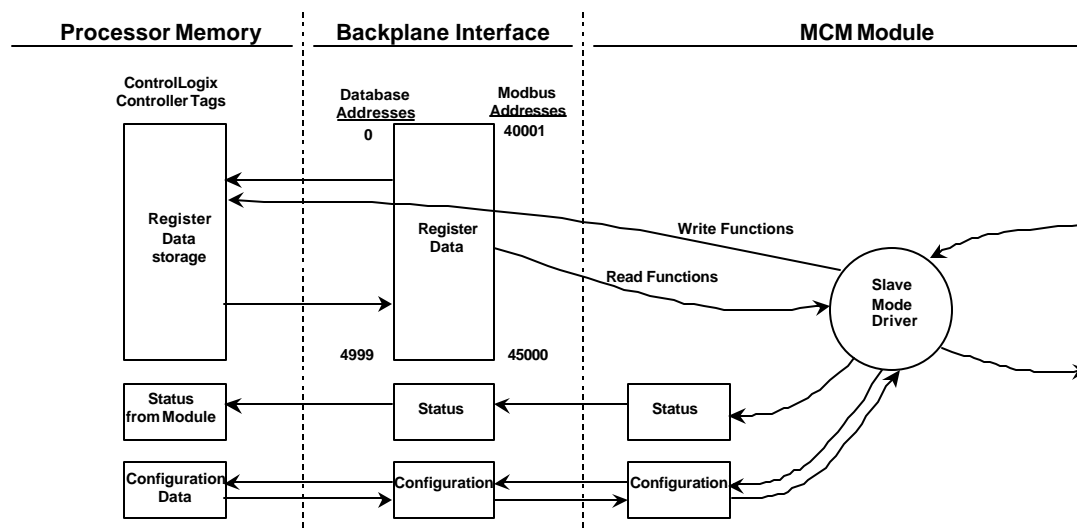


Etapa	Descrição
1	O driver Modbus escravo recebe a informação de configuração a partir do processador ControlLogix. Esta informação é usada para configurar a porta serial e definir as características de nó do escravo. Adicionalmente, a informação de configuração contém dados que podem ser usados para deslocar dados no banco de dados para endereços solicitados nas mensagens recebidas a partir das unidades mestre.
2	Um dispositivo host, tal como um Modicon PLC ou pacote MMI, emite um comando de leitura ou escrita para o endereço de nó do módulo. O driver de porta qualifica a mensagem antes de aceitá-la no módulo.
3	Uma vez que o módulo aceite o comando, o dado é imediatamente transferido de/para o banco de dados interno no módulo. Se o comando é um comando de leitura, o dado é lido fora do banco de dados e emitida uma mensagem de resposta. Se o comando é um comando de escrita, o dado é escrito diretamente no banco de dados e emitida uma mensagem de resposta.
4	Uma vez que o processamento tenha sido completado, na etapa 2 a resposta é emitida para o nó mestre de origem.
5	Contadores são disponíveis no bloco de status permitindo que o programa de Lógica Ladder determine o nível de atividade do Driver Escravo.

Dirija-se à seção **Set up de Módulo** para uma lista completa dos parâmetros que devem ser definidos para uma Porta Escravo.

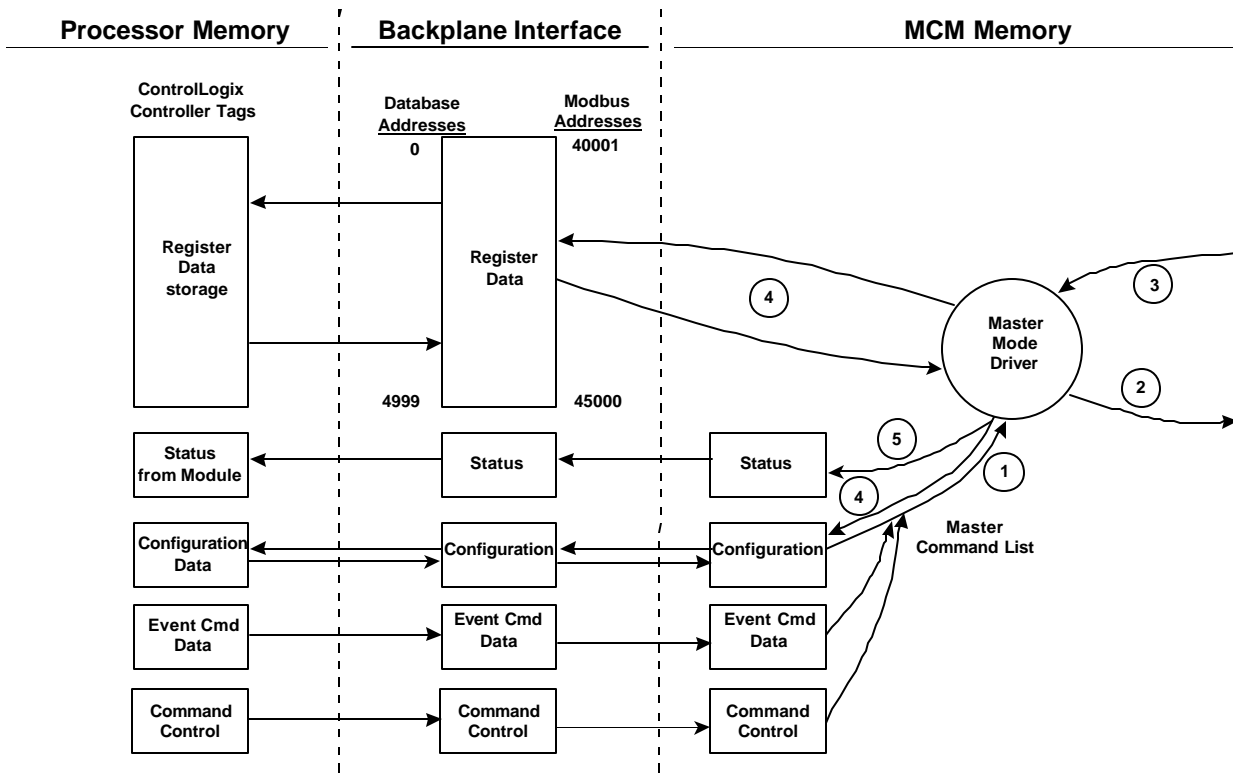
Uma exceção para este modo normal de operação ocorre quando o modo Pass_Through está implementado. Neste modo, pedidos escritos passam diretamente para o processador e não são colocados no banco de dados. Isto permite um controle remoto direto do processador sem

um banco de dados intermediário. Este modo é especialmente útil para dispositivos Mestre que não enviam ambos estados de controle. Por exemplo, um sistema SCADA pode somente enviar um comando “ON” a um ponto de controle digital. e nunca o estado “Eliminar”. O sistema SCADA espera que a lógica local restabeleça o bit de controle. O modo Pass_Through deve ser usado para simular este modo de operação. O diagrama abaixo ilustra o fluxo de dados para Porta Escravo com modo Pass_Through Habilitado.



2.9.4 Modo de Driver Mestre

No modo mestre de operação, o módulo MVI56-MCM é responsável pela emissão de comandos de leitura ou escrita a dispositivos escravo na rede Modbus.. Estes comandos são configurados pelo usuário no módulo com a Lista de Comando Mestre recebida do processador ControlLogix ou emitida diretamente do processador ControlLogix (Controle de Comando de Evento). O status de comando retorna para o processador a cada comando individual no bloco de status da Lista de Comando. A localização deste bloco de status no banco de dados interno é definida pelo usuário. O diagrama e tabela associada detalham o fluxo de dados para dentro e para fora do módulo.



(ver anexo)

Etapa	Descrição
1	O driver Modbus escravo obtém dado de configuração a partir do processador ControlLogix. O dado de configuração obtido inclui o número de comandos e Lista de Comando Mestre. Estes valores são usados pelo driver Mestre para determinar o tipo de comandos a serem emitidos para outros nós na rede Modbus (dirija-se à Guia de Setup do módulo MVI56-MCM)
2	Uma vez configurado, o driver Mestre começa a transmitir comandos de leitura e/ou escrita a outros nós na rede. Se os dados forem escritos em outro nó, os dados para o comando de escrita serão obtidos a partir do banco de dados interno do módulo para construir o comando.
3	Presumindo um processamento bem sucedido pelo nó especificado no comando, uma mensagem de resposta é recebida no driver Mestre para processamento.
4	O dado recebido a partir do nó na rede é passado para o banco de dados interno do módulo, assumindo um comando de leitura.
5	Status retorna para o processador ControlLogix a cada comando na Lista de Comando Mestre.

Dirija-se à seção **Setup de Módulo** onde pode ser vista uma descrição completa dos parâmetros requeridos para definir a porta Mestre Modbus. Dirija-se à documentação **Driver MCM** onde pode ser vista discussão completa da estrutura e conteúdo de cada comando.

Deve se tomar cuidado ao construir cada comando na lista para uma operação previsível do módulo. Se dois comandos forem escritos para o mesmo endereço de banco de dados interno do módulo, o resultado não será o desejado. Todos os comandos contendo dados inválidos serão ignorados pelo módulo.

2.9.4.1 Lista de Comando Mestre

Para funcionar no Modo Mestre, a Lista de Comando Mestre do módulo deve ser definida. Esta lista contém até 100 entradas individuais, sendo que cada entrada contém a informação requerida para construir o comando válido. Incluindo:

- Modo de comando Habilitado ((0) Desabilitado, (1) Contínuo ou (2) Condicional.)
- Endereço de Nó de Escravo
- Tipo de Comando – Escrita ou Leitura até 125 palavras (16000 bits) por comando
- Fonte de Banco de Dados e Endereço de Registro de Destino- Determina onde os dados serão colocados e/ou obtidos.
- Quantidade: Selecione o número de palavras a transferir- 1 a 100 em FC 3, 4 ou 16. Selecione o número de bits em FC 1, 2, 15.

Como a lista é lida a partir do processador e como os comandos são processados, um valor de erro é mantido no módulo para cada comando. Esta lista de erro pode ser transferida para o processador. Os erros gerados pelo módulo são mostrados abaixo:

Erros de Protocolo Modbus Padrão

Código	Descrição
1	Função ilegal
2	Endereço de Dado Ilegal
3	Valor de Dado Ilegal
4	Falha no Dispositivo Associado
5	Reconhecimento
6	Ocupado, Mensagem Rejeitada

Códigos de Erro de Comunicação de Módulo

Código	Descrição
-1	Linha de controle de modem CTS não estabelecida antes da transmissão
-2	Timeout enquanto transmite mensagem
-11	Timeout aguardando resposta depois do pedido
253	Endereço escravo incorreto na resposta
254	Código de função incorreto na resposta
255	Valor CRC/LRC inválido na resposta

Erros de Entrada na Lista de Comando

Código	Descrição
-41	Código Habilitado inválido
-42	Endereço Interno > Endereço máximo
-43	Endereço de nó inválido (<0 ou > 255)
-44	Parâmetro de Quantidade em zero
-45	Código de Função Inválido
-45	Código Swap Inválido

3- Modificando Configuração de Módulo

Para o módulo MVI56-MCM funcionar, uma quantidade mínima de dados de configuração deve ser transferida para o módulo. A tabela abaixo provê uma visão geral dos diferentes tipos de dados de configuração, dependendo dos modos operacionais a serem suportados.

Endereço de Registro de Modo	Modos funcionais afetados	Nome	Descrição
5000-5009	Transferência de dados	Configuração de Módulo Geral	Esta seção do dado de configuração contém dados de configuração de módulo que definem a transferência de dados entre módulo e processador ControlLogix
5010-5039 e 5040-5069	Mestre e Escravo	Configuração de Porta	Estas seções são usadas para definir as características de cada uma das portas de comunicação serial no módulo. Estes parâmetros devem ser corretamente estabelecidos para uma operação adequada do módulo.
5070-5869 e 5870-6669	Mestre	Lista de Comando Mestre	Se a funcionalidade do modo Mestre do módulo deve ser suportada em uma porta, a Lista de Comando Mestre deve ser implementada.
6750-6770	Mestre e Escravo	Configuração de Porta Aux.	Estas seções são usadas para definir as características de cada uma das portas de comunicação serial Modbus no módulo. Estes parâmetros devem ser corretamente implementados para uma operação adequada do módulo.

Dirija-se à seção “**Setup do Módulo**”, onde pode ser visto uma descrição da configuração do módulo. O módulo deve ser configurado pelo menos uma vez quando a placa for energizada pela primeira vez, e sempre que os parâmetros devem ser alterados.

3.1 Energização

Na energização, o módulo entra em loop lógico aguardando os dados de configuração. Quando recebe, o módulo começa executar Lista de Comando, se existir.

3.2 Mudando Parâmetros Durante Operação

Uma cópia dos dados de configuração do módulo é mapeada no banco de dados do módulo como mostrado na tabela acima. Estes valores são inicializados quando o módulo recebe pela primeira vez sua configuração a partir do processador ControlLogix. Qualquer nó na rede pode alterar este dado. Uma porta Mestre no módulo pode fazer varredura de um escravo para o dado ou uma porta escravo pode receber o dado de uma unidade Mestre remota. Este módulo não usará este dado até receber o comando. A Lógica Ladder pode ser escrita para emitir um bloco de comando de Configuração de Escrita (9997) para o módulo. Um dispositivo remoto pode estabelecer um valor de 9997 no endereço 6800 no módulo para download da configuração no processador. Alternativamente a porta configuração/ debug no módulo pode ser usada para emitir o comando diretamente para o módulo. Todos estes três métodos forçam o módulo a descarregar (download) a configuração no processador ControlLogix. A Lógica Ladder deve estar no processador para aceitar os blocos enviados pelo módulo. Se tudo estiver corretamente configurado, o módulo pode receber sua configuração de um dispositivo remoto.

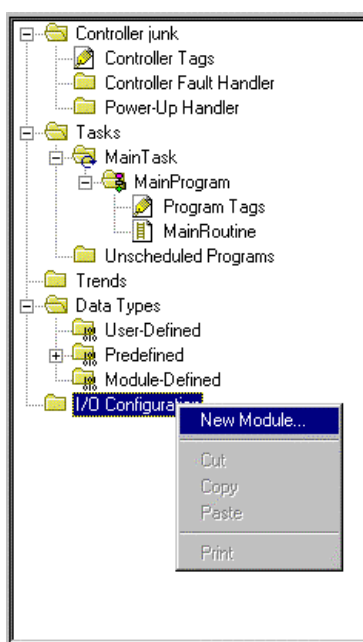
3.3 Setup do Módulo

O setup do módulo MVI56-MCM requer somente configuração de software usando programa RSLogix 5000. O método mais fácil para implementar o módulo é iniciar com o exemplo provido com o módulo (MVI56-MCM -EX1.ACD).

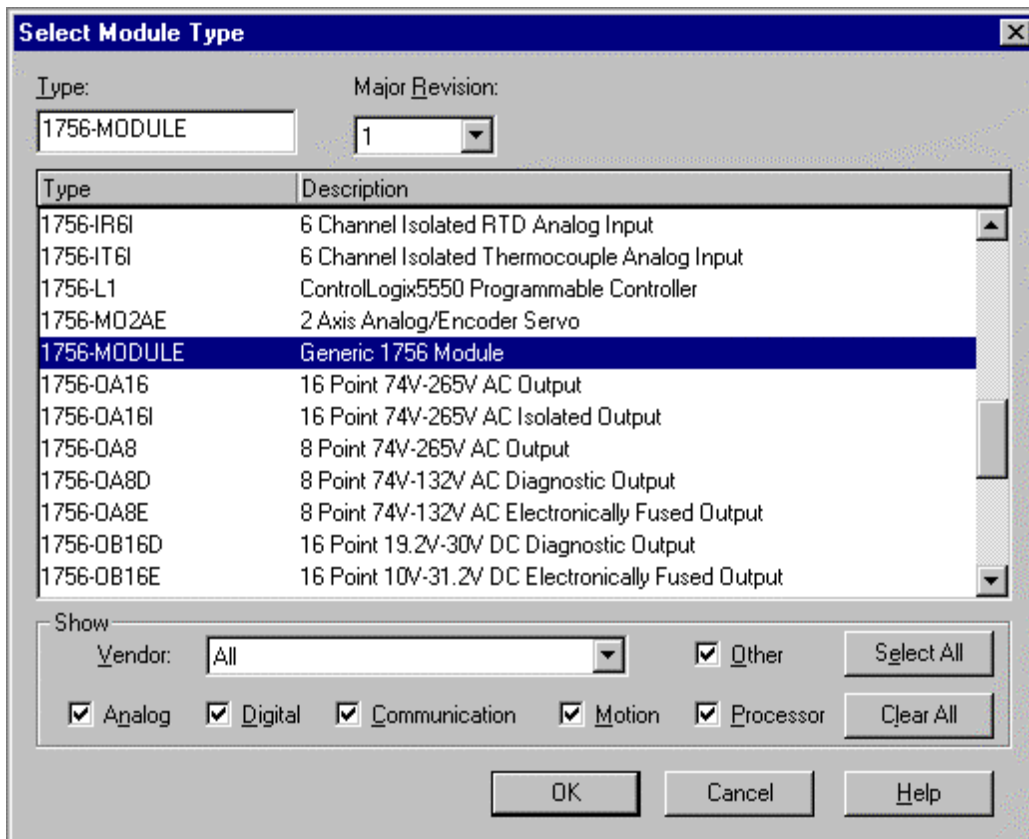
Se você estiver instalando este módulo em uma aplicação existente, você simplesmente pode copiar os elementos requeridos da Lógica Ladder de exemplo para sua aplicação.

Nota: O módulo apenas poderá ser adicionado a um projeto usando o software no modo off_line.

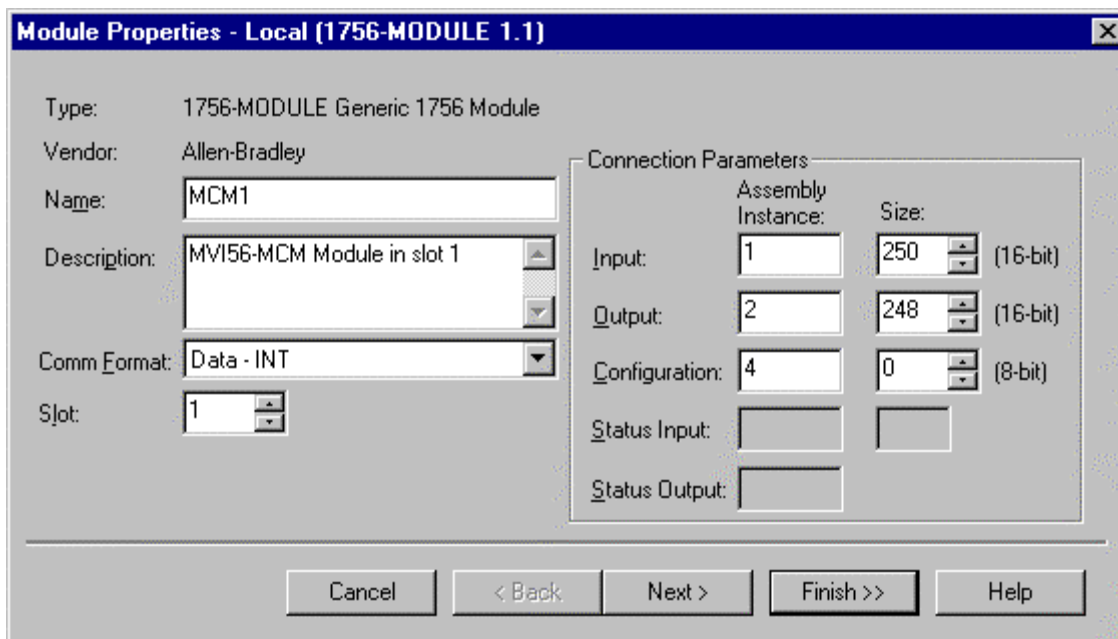
A primeira etapa no setup do módulo é definir o módulo para o sistema. Clique o botão da direita no mouse na opção **"I/O Configuration"** na janela **"Controller Organization"** para aparecer uma janela pop_up de menu. Selecione a opção **"New Module"** do menu **"I/O Configuration"**.



Isto faz o programa mostrar a seguinte caixa de diálogo:

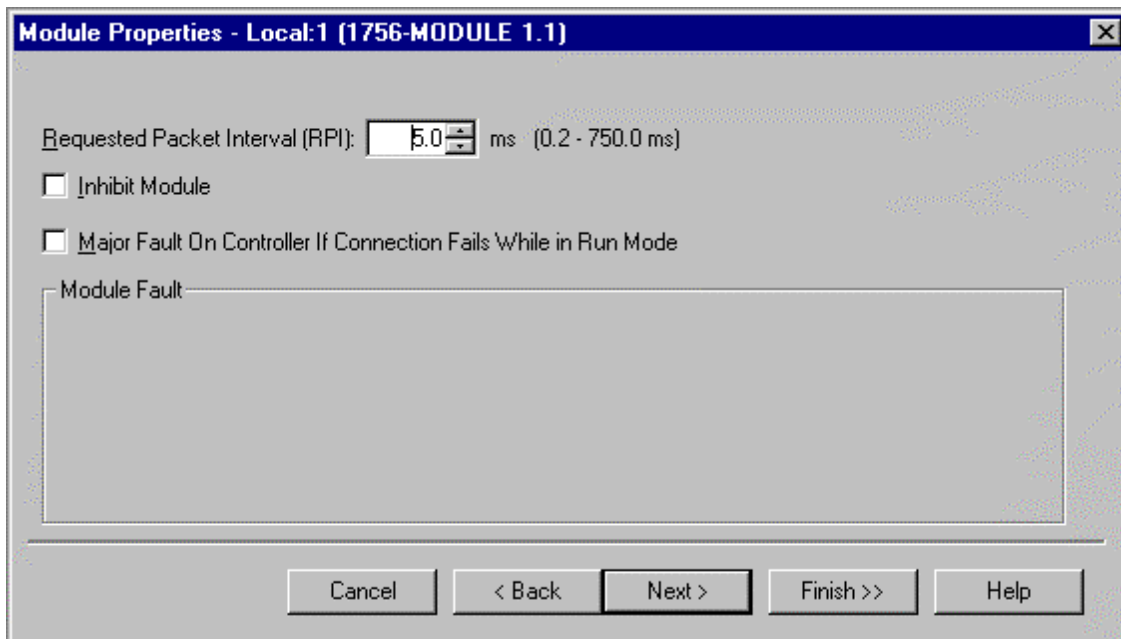


Selecione “1756 Module” (1756 Module genérico) da lista e selecione o botão **OK**. Aparece a seguinte caixa de diálogo.



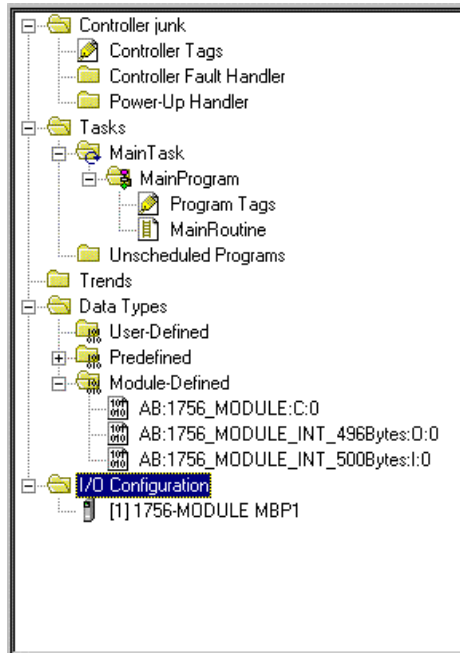
Preencha os campos **Name**, **Description**, **Slot** nas caixas de diálogo para sua aplicação. Assegure-se de selecionar **COMM_Format** como **Data_INT** na caixa de diálogo. Se os valores

Assembly Instance e **Size** não forem preenchidos corretamente resulta um módulo que não se comunica pelo barramento do Rack ControlLogix. Selecione **"Next"** para a seguinte a caixa de diálogo:

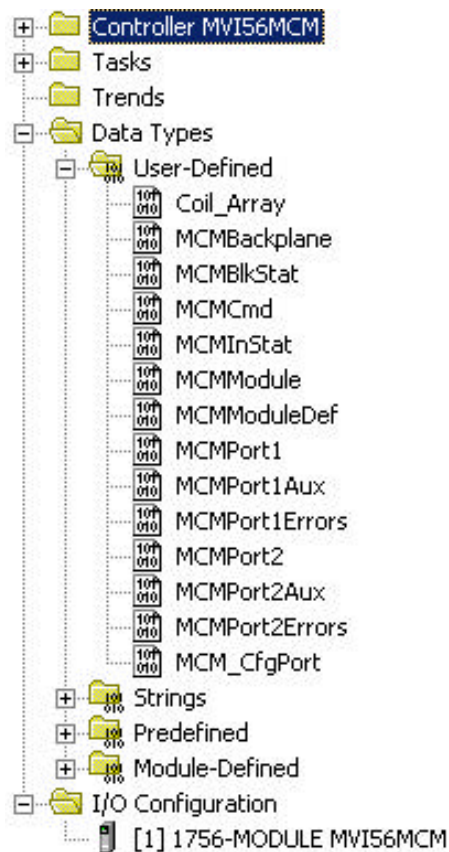


Selecione o valor de intervalo **Request Packet Internal** para escanear E/S no módulo. Este valor representa a frequência mínima na qual o módulo lida com eventos programados. Este valor não é menor que 1 milissegundo. Valores entre 1 e 10 milissegundos são aplicáveis na maior parte das aplicações.

Depois de completar o setup do módulo, a janela **"Controller Organization"** mostra a presença do módulo. O dado requerido para o módulo então é definido para aplicação e objetos são alocados na área de Tags Controladores. Um exemplo da janela **"Controller Organization"** pode ser visto abaixo:



A próxima etapa no setup do módulo é definir os tipos de dados definidos por usuário a serem usados com o módulo. Copie estes dados da Lógica Ladder se você não estiver usando o exemplo. Eles estão definidos, se você estiver começando a partir da Lógica Ladder de exemplo. A janela “**Controller Organization**” mostra os tipos de dados definidos por usuário como pode ser visto no exemplo abaixo:



A próxima etapa no setup do módulo é definir os dados a serem usados para interfacear módulo e Lógica Ladder.

Abra a caixa de diálogo **“Controller Tags Edit Tags”** e forneça os valores mostrados no exemplo a seguir: O módulo MVI56-MCM é definido no exemplo como MCM1.

Você pode colocar qualquer nome para **Tag Name**.

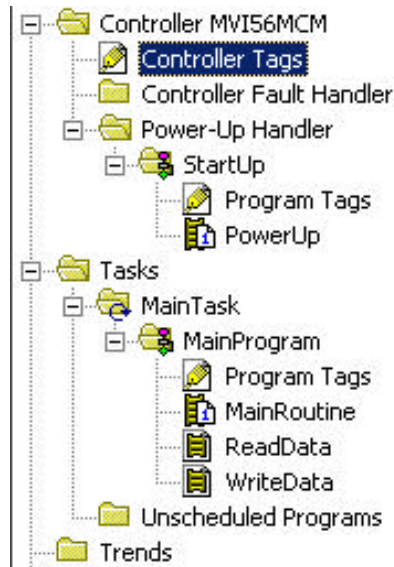
Se você estiver usando a Lógica Ladder de exemplo esta etapa já foi realizada

The screenshot shows a software interface window titled 'Controller Tags Edit Tags'. At the top, there are dropdown menus for 'Scope' (set to 'MVI56MCM/control'), 'Style' (set to 'Show All'), and 'Seg' (set to 'TagName'). Below these is a table with the following columns: Tag Name, Value, Force Mask, Style, Type, and Description. The table contains the following data:

Tag Name	Value	Force Mask	Style	Type	Description
CmdControl	0		Decimal	BOOL	
ColdBoot	2#0000_0000		Binary	BOOL	
EventCmd	0		Decimal	BOOL	
Local 1:C	{...}	{...}		AB-1756_MODULE_I/O	
Local 1:I	{...}	{...}		AB-1756_MODULE_INT...	
Local 1:O	{...}	{...}		AB-1756_MODULE_INT...	
MBData	{...}	{...}		Col_Array	
MBControl1	{...}	{...}		CONTROL	
MBControl2	{...}	{...}		CONTROL	
MBMsg	{...}	{...}	Decimal	SINT[500]	
MBMsgLen	200		Decimal	INT	
MBOffset	0		Decimal	INT	
MBOffsetBit	0		Decimal	INT	
MBScratch	{...}	{...}	Decimal	INT[3]	
MCM	{...}	{...}		MCMModuleDef	
M_FAULTS	{...}	{...}	Decimal	DINT[12]	
PortSlave0Read	2#0000_0000		Binary	BOOL	
PortSlave128Read	2#0000_0000		Binary	BOOL	
PortSlave0Read	2#0000_0000		Binary	BOOL	
PortSlave128Read	2#0000_0000		Binary	BOOL	
WarmBoot	2#0000_0000		Binary	BOOL	

Neste ponto, pegue o tempo para preencher os valores de configuração na tabelas de dados MCM e ajuste os tamanhos de array. Dirija-se à seção Objeto de Dado de Módulo deste documento para obter informação de como configurar este módulo.

A última etapa no setup do módulo é adicionar a Lógica Ladder. Se você estiver usando a Lógica Ladder de exemplo ajuste o ladder à sua aplicação. Se você não estiver usando Ladder de exemplo, copie a Lógica Ladder mostrada na janela **“Controller Organization”** sob sua aplicação.



O setup do módulo está completo e pronto para ser usado em sua aplicação.

Descarregue a nova aplicação para o processador e coloque o processador no modo RUN. Se todos parâmetros de configuração estiverem corretos e o módulo conectado a uma rede Modbus, o LED de aplicação do módulo (APP LED) deve permanecer **off** e o LED de atividade do barramento (BP LED) deve piscar rapidamente. Dirija-se à seção Diagnóstico e Solução de Problemas neste manual em caso de erro. Conecte um computador ou terminal à porta Debug/Configuração no módulo e verifique o status do módulo usando o “debugger” residente no módulo.

3.4 Objeto de Dado do Módulo (MCMModuleDef)

Todos dados com relação a MVI56-MCM estão armazenados em um tipo de dado definido por usuário. Um exemplo do tipo de dado é requerido antes de o módulo poder ser usado. Isto é feito apenas declarando uma variável do tipo de dado na caixa “**Controller Tags Edit Tags**”. A estrutura do objeto está mostrada na figura a seguir:

Warning: This structure is being referenced. Modifications will result in loss of data.

Name:

Description:

Members: Data Type Size: ?? byte(s)

	Name	Data Type	Style	Description
<input type="checkbox"/>	ModDef	MCModule		Settings for data to be read and written to the module
<input type="checkbox"/>	CFG_Port	MCM_CfgPort		Settings for debug port
<input type="checkbox"/>	Port1	MCMPort1		Modbus settings for P1
<input type="checkbox"/>	Port2	MCMPort2		Modbus settings for P2
<input type="checkbox"/>	P1Cmd	MCMCmd[100]		Master Command List for Port 1
<input type="checkbox"/>	P2Cmd	MCMCmd[100]		Master Command List for Port 2
<input type="checkbox"/>	InStat	MCMInStat		Status information in each read block
<input type="checkbox"/>	ReadData	INT[600]	Decimal	Data read from module
<input type="checkbox"/>	WriteData	INT[600]	Decimal	Data to write to module
<input type="checkbox"/>	BP	MCMBackplane		Data to handle backplane logic
<input type="checkbox"/>	P1Slaves	INT[256]	Decimal	Port 1 slave status values
<input type="checkbox"/>	P2Slaves	INT[256]	Decimal	Port 2 slave status values
<input type="checkbox"/>	Port1Aux	MCMPort1Aux		MB auxilliary settings for Port 1
<input type="checkbox"/>	Port2Aux	MCMPort2Aux		MB auxilliary settings for Port 2
<input type="checkbox"/>	*			

Este objeto contém objetos que definem configuração, dados de usuário, e dados de controle de comando e status relativos ao módulo. Cada um destes tipos de objeto será discutido nas seções que se seguem deste documento.

3.4.1 Objetos de Configuração

A configuração do módulo é realizada simplesmente preenchendo nos valores no objeto de módulo na caixa de diálogo “**Controller Tags Edit Tags**”. Cada parâmetro requerido pelo módulo tem uma localização definida no objeto. As seguintes tabelas e discussões descrevem os parâmetros colocados na caixa de diálogo. Você pode ver estas tabelas abrindo tipo de dado na opção “**User Defined Data Type**” na janela “**Controller Organization**”.

3.4.1.1 Parâmetros de Transferência de Dados (MCModule)

Name	Data Type	Style	Description
WriteStartReg	INT	Decimal	Start reg to transfer from PLC to module
WriteRegCnt	INT	Decimal	Number of registers to write from PLC
ReadStartReg	INT	Decimal	Start reg to transfer from module to PLC
ReadRegCnt	INT	Decimal	Number of registers to transfer from module
BPFail	INT	Decimal	Determines module operation if BP fails 0=continue,>0=number of retries
ErrStatPtr	INT	Decimal	Internal DB start register for status data (-1=ignore)
*			

Este objeto é usado para definir parâmetros para movimentar dados entre módulo e processador. Os valores fornecidos determinam a Lógica Ladder e o tamanho de dado requerido para aplicação. Os arrays **ReadData** e **WriteData** devem ser iguais ou maiores que os valores de quantidade fornecidos. A Lógica Ladder deve ser escrita para processar o número de blocos de dados a serem transferidos. O número de blocos é computado da seguinte forma:

$$\text{BlockCnt} = \text{INT}(\text{RegCnt}/200) + \text{if}(\text{MOD}(\text{RegCnt},200), 1,0)$$

Se a quantidade de registros resultar em divisão exata por 200, o número de blocos é facilmente computável e o Ladder fica muito simples de escrever. Se o número não resultar em divisão exata por 200, um procedimento especial para o último bloco de dado deverá ser desenvolvido, para transferir menos que 200 palavras. Recomendadas quantidades divisíveis por 200.

O parâmetro **BPFail** é usado para determinar se o módulo deve continuar comunicando na rede Modbus quando falha a operação de transferência de barramento. Um valor de **zero** indica que o módulo deve continuar se comunicando quando o barramento não estiver operacional. Se o valor for maior que zero, o barramento tenta um certo número de vezes antes de uma falha ser reportada e cessar a comunicação nas portas. Quando a configuração pelo barramento for restabelecida, o módulo começa comunicação na rede. Por exemplo, se você fornecer um valor de 10 para o parâmetro, o módulo interrompe todas comunicações Modbus se forem reconhecidos sucessivos 10 erros de barramento. Quando for reconhecida uma transferência bem sucedida, o módulo reinicia comunicação na rede.

O parâmetro **ErrStatPtr** é usado para definir a localização do banco de dados do módulo onde é armazenado o dado erro/ status. Se for dado valor -1, o dado não será armazenado na área de dado do usuário. Um valor entre 0 e 4939 faz o programa do módulo armazenar o dado na localização especificada.

3.4.1.2 Parâmetros de Porta Modbus (MCMPort)

Name:

Description:

Members: Data Type Size: 12 byte(s)

Name	Data Type	Style	Description
Baudrate	DINT	Decimal	Baudrate for P1 debug port. Fixed value and cannot be changed with this setting.
Parity	INT	Decimal	0=None. Fixed value
DataBits	INT	Decimal	8 data bits. Fixed value
StopBits	INT	Decimal	1 stop bits. Fixed value
*			

Name:

Description:

Members: Data Type Size: 52 byte(s)

Name	Data Type	Style	Description
Enabled	INT	Decimal	0=Port Disabled, 1=Port Enabled
Type	INT	Decimal	0=Master, 1=Slave, 2=Slave, pass-through, 3=Slave, formatted pass-through/Slave swapped, 4=Slave, form, pass-through
FloatFlag	INT	Decimal	0=No floating-point data, 1=Use floating-point data
FloatStat	INT	Decimal	Register offset in message for floating-point data
FloatOffset	INT	Decimal	Internal DB offset to start of floating-point data
Protocol	INT	Decimal	0=Modbus RTU, 1=Modbus ASCII
Baudrate	INT	Decimal	Baudrate for port (110 to 115.2K)
Parity	INT	Decimal	0=None, 1=Odd, 2=Even, 3=Mark, 4=Space
DataBits	INT	Decimal	5 to 8 data bits
StopBits	INT	Decimal	1 or 2 stop bits
RTSDn	INT	Decimal	0-65535 mSec delay before data
RTSDH	INT	Decimal	0-65535 mSec delay after data
MinResp	INT	Decimal	0-65535 mSec minimum time before response to request
UseCTS	INT	Decimal	0=No, 1=Yes to use CTS modem line
SlaveID	INT	Decimal	1-255 Modbus Node Address (Slave)
BinInOffset	INT	Decimal	Internal DB offset to bin input data (Slave)
WordInOffset	INT	Decimal	Internal DB offset to word input data (Slave)
OutOffset	INT	Decimal	Internal DB offset to bin output data (Slave)
HoldOffset	INT	Decimal	Internal DB offset to holding register data (Slave)
CmdCount	INT	Decimal	Command list count (Master)
MinCmdDelay	INT	Decimal	0-65535 mSec minimum time between each command (Master)
CmdErrPt	INT	Decimal	Internal DB location to place command error list (Master)
RespT0	INT	Decimal	0-65535 mSec response timeout for command (Master)
Retry_Count	INT	Decimal	Retry count for failed request (Master)
ErrrDelayCnt	INT	Decimal	0-65535 Command cycle count if error (Master)
*			

Este objeto é usado para definir os parâmetros de operação de cada porta Modbus no módulo. Dirija-se para o Apêndice C para definição de cada parâmetro.

3.4.1.3 Comandos de Mestre Modbus (MCMCmd)

Este objeto é usado para definir os parâmetros de cada comando na Lista de Comando Mestre. O objeto **MCMModuleDef** contém um array destes objetos que definem a lista completa para cada porta. A definição de cada parâmetro requerido para cada comando é dada abaixo.

Parâmetro	Descrição
Habilitado (Enable)	Este parâmetro é usado para definir se o comando será executado ou desconsiderado. Os seguintes valores são válidos: 0= desabilita comando e não executa; 1= o comando será considerado na execução de cada escaneamento da Lista de Comando e controlado pelo parâmetro Polling ; 2= o comando somente executa se o dado associado ao comando tiver sido alterado desde a última vez em que o comando foi emitido. Esta opção somente é disponível para comandos de escrita.;
(Endereço Interno) IntAddress	Este parâmetro especifica o endereço de registro interno de início a ser associado ao comando. A entrada válida para este parâmetro é 0 a 4999 registros ou 0 a 65535 bits quando endereçando comando em nível de bit.
Varredura Int PollInt	Este parâmetro define o número mínimo de segundos a aguardar entre a execução de comandos contínuos (Habilitado =1). Este Comando de intervalo de varredura pode ser usado para aliviar a carga de comunicação em uma rede carregada. A entrada válida é 0 a 6535.
Quantidade	Este parâmetro define o número de registros a ser considerado pelo comando. A entrada válida é 1 a 125 palavras ou 16000 bits.
Swap	Este parâmetro que é usado para especificar o dado usado no comando deve ser alterado quando um código de função Modbus 3 for usado para dado de leitura a partir de um nó na rede. Os valores que podem ser designados são os seguintes: 0= sem troca de dados, 2= troca valores de byte e palavra e 3= troca valores de byte esta opção é usada quando se interfaceia o módulo com ASCII e dado de Ponto Flutuante em outros dispositivos.

Dispositivo	Este parâmetro é usado para designar um endereço de nó de escravo Modbus para o módulo alcançar com o comando na rede Modbus. A este parâmetro pode ser designado valores de 0 a 255. A maior parte das redes Modbus limitam o valor superior em 247.
Função	Este parâmetro especifica a função Modbus a ser realizada pelo comando. Entradas válidas são 1, 2, 3, 4, 5, 6, 15 e 16.
End de Dispositivo (DevAddress)	Este parâmetro define o endereço de início no dispositivo considerado pelo modo. Os valores dados para este campo dependem da definição de banco de dados do fabricante. Dirija-se à definição de banco de dados do fabricante do dispositivo para determinar a localização do dado a ser interfaceado.

3.4.2 Objeto de Status (MCMInStat)

Este objeto é usado para ver o status do módulo. O objeto **MCMInStat** mostrado abaixo é atualizado cada vez que um bloco de leitura é recebido pelo processador. Este dado pode ser usado para monitorar o status do módulo em tempo real.

Name:

Description:

Members: Data Type Size: 72 byte(s)

Name	Data Type	Style	Description
PassCnt	INT	Decimal	Program cycle counter
Product	INT[2]	Hex	Product Name
Rev	INT[2]	Hex	Revision Level Number
OP	INT[2]	Hex	Operating Level Number
Run	INT[2]	Hex	Run Number
[-] Prt1Errs	MCMPort1Errors		Port error statistics
[-] CmdReq	INT	Decimal	Total number of command list requests sent
[-] CmdResp	INT	Decimal	Total number of command list responses received
[-] CmdErr	INT	Decimal	Total number of command list errors
[-] Requests	INT	Decimal	Total number of requests for port
[-] Responses	INT	Decimal	Total number of responses for port
[-] ErrSent	INT	Decimal	Total number of errors sent
[-] ErrRec	INT	Decimal	Total number of errors received
[-] Prt2Errs	MCMPort2Errors		
[-] CmdReq	INT	Decimal	Total number of command list requests sent
[-] CmdResp	INT	Decimal	Total number of command list responses received
[-] CmdErr	INT	Decimal	Total number of command list errors
[-] Requests	INT	Decimal	Total number of requests for port
[-] Responses	INT	Decimal	Total number of responses for port
[-] ErrSent	INT	Decimal	Total number of errors sent
[-] ErrRec	INT	Decimal	Total number of errors received
[-] BlkErrs	MCMBlkStat		Block transfer statistics
[-] Read	INT	Decimal	Total number of read block transfers
[-] Write	INT	Decimal	Total number of write block transfers
[-] Parse	INT	Decimal	Total number of blocks parsed
[-] Event	INT	Decimal	Total number of event blocks received
[-] Cmd	INT	Decimal	Total number of command blocks received
[-] Err	INT	Decimal	Total number of block transfer errors
Port1CurErr	INT	Decimal	Current error/index for Port 1
Port1LErr	INT	Decimal	Last error/index for Port 1
Port2CurErr	INT	Decimal	Current error/index for Port 2
Port2LErr	INT	Decimal	Last error/index for Port 2
*			

Dirija-se ao Apêndice B para uma listagem completa dos dados armazenados neste objeto.

3.5 Objetos de Dados de Usuário

Estes objetos são usados para segurar dados a serem transferidos entre processador e módulo MVI56-MCM. Os dados de usuário são dados de leitura e escrita transferidos entre processador e módulo como páginas de dados até um comprimento de 200 palavras.

ReadData	INT[600]	Decimal	Data read from module
WriteData	INT[600]	Decimal	Data to write to module

Os dados de leitura (**ReadData**) é um array que deve corresponder com o valor dado ao parâmetro **ReadRegCnt** no objeto **MCModule**. Para maior facilidade de uso, este array deve ser dimensionado com incrementos iguais de 200 palavras. Este dado é paginado até 200 palavras de uma vez do módulo para o processador. A tarefa **ReadData** é responsável por colocar o dado recebido na posição adequada no array de dado de leitura.

Este dado pode ser usado para status e controle na Lógica Ladder do processador.

O dado de escrita (**WriteData**) é um array que corresponde ao valor dado no parâmetro **WriteRegCnt** do objeto **MCModule**. Para facilidade de uso, este array deve ser dimensionado como incrementos de 200 palavras. Este dado é paginado até 200 palavras de uma vez do processador para o módulo. A tarefa **WriteData** é responsável por colocar o dado de escrita na imagem de saída para transferir para o módulo. Este dado é transferido do processador para o módulo para informação de status e controle para ser usado em outros nós na rede.

Se este array for maior que 600 registros, mude o valor **LIM** superior na linha **ReadData 2** e linha **ReadData 10** do arquivo ladder.

3.6 Status e Controle de Varredura de Escravo

Dois arrays são alocados no objeto primário do módulo para guardar status de varredura de cada escravo nas portas mestre. Os dados de status podem ser usados para determinar quais escravos estão correntemente ativos na porta, apresentam erro de comunicação, ou têm sua varredura suspensa ou estão desabilitados. A Lógica Ladder no processador pode ser escrita para monitorar e controlar o status de cada escravo em uma porta Mestre. Os objetos usados estão mostrados no diagrama a seguir:

Escravos P1	INT (256)	Decimal	Valores de status de Porta 1 escravo
Escravos P2	INT (256)	Decimal	Valores de status de Porta 2 escravo

Usando blocos especiais, o processador pode pedir o dado corrente aos escravos. Usando outros blocos, o processador pode habilitar ou desabilitar a varredura de escravos selecionados.

3.7 Dados de Mensagem Modbus

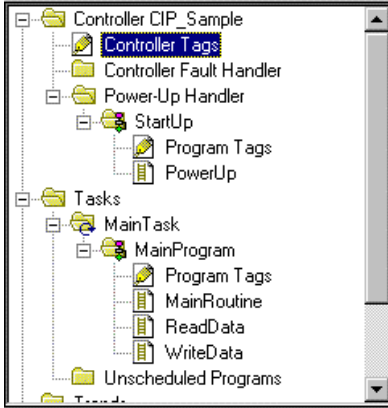
Esta nova versão do programa do módulo inclui o modo **Pass_Through** de operação. Neste modo, mensagens de escrita enviadas para uma porta escravo são transferidas diretamente através do processador. A Lógica Ladder é responsável em processar a mensagem recebida usando esta característica. Dois objetos de dados são requeridos para este modo de operação. Uma variável para guardar o comprimento da mensagem e um buffer para guardar a mensagem.

Esta informação é transferida do módulo para o processador usando um código de identificação de bloco 9996. A palavra 2 deste bloco contém o comprimento da mensagem e a mensagem começa na palavra 3. Outros Tags Controladores são requeridos para armazenar os valores controlados contidos nestas mensagens. O protocolo Modbus suporta controlador de saída binária (coils - funções 5 e 15) e registros (funções 6 e 16).

4 Modificando Lógica Ladder de Exemplo

A Lógica Ladder é requerida para aplicação do módulo MVI56-MCM. As tarefas que devem ser manuseadas pela Lógica Ladder são as seguintes: configuração de módulo, transferência de dados, manuseio de bloco especial, e recepção de dado de status. Esta seção discute cada aspecto da Lógica Ladder como requerido pelo módulo. Adicionalmente, o operador de energização deve ser escrito para manusear a inicialização de dado do módulo e eliminar qualquer condição de falha do processador.

A janela “**Controller Organization**” da Lógica Ladder de exemplo para o módulo MVI56-MCM está mostrada no exemplo abaixo:



4.1 Energização

A Lógica Ladder de energização é usada para inicializar os objetos de dado usados pelo módulo MVI56-MCM e recuperar falhas do controlador durante energização inicial do processador.

A Lógica Ladder requerida para realizar estas tarefas está mostrada nos parágrafos que se seguem.

Esta linha é usada para recuperação em um processador uma condição de falha por queda de energia e reinicializar quando o processador se encontra no modo RUN. Você também pode ter que enfrentar outras condições de falha. Adicionalmente, um operador de falha pode ser escrito para o processador lidar com outras falhas. Este objeto **MJFAULT** deve ser definido nos Tags de Controlador antes de ser usado nesta lógica.



Esta linha é usada para inicializar os últimos valores de leitura e escrita, a imagem de saída para o módulo MVI56-MCM e a área de dado de escrita para zero. Os últimos valores de leitura e escrita (**MCM.BP.LastRead**) e (**MCM.BP.LastWrite**) são usados na lógica de transferência de dados. A imagem de saída para o módulo MVI56-MCM (**Local:1:0.Data[_]**) é usada para transferir dados do processador para o módulo. A área de dados de escrita (**MCM.WiteData[_]**) é usada para armazenar os dados do processador a serem escritos no módulo usando a imagem de saída.



4.2 MainRoutine

A tarefa “**MainRoutine**” é usada para reconhecer a presença de novos dados de leitura a partir do módulo para o processador. O módulo cicla através de sua lista de blocos de leitura para transferir dados do módulo para o processador. Sempre que um novo dado se encontra disponível, o módulo coloca o valor para o bloco na imagem de entrada do módulo (**Local:1Data[249]**). A Lógica Ladder deve constantemente escanear esta palavra de entrada a busca de um novo valor. Quando houver um novo valor, a Lógica Ladder realiza as tarefas **ReadData** e **WriteData**, nesta ordem.

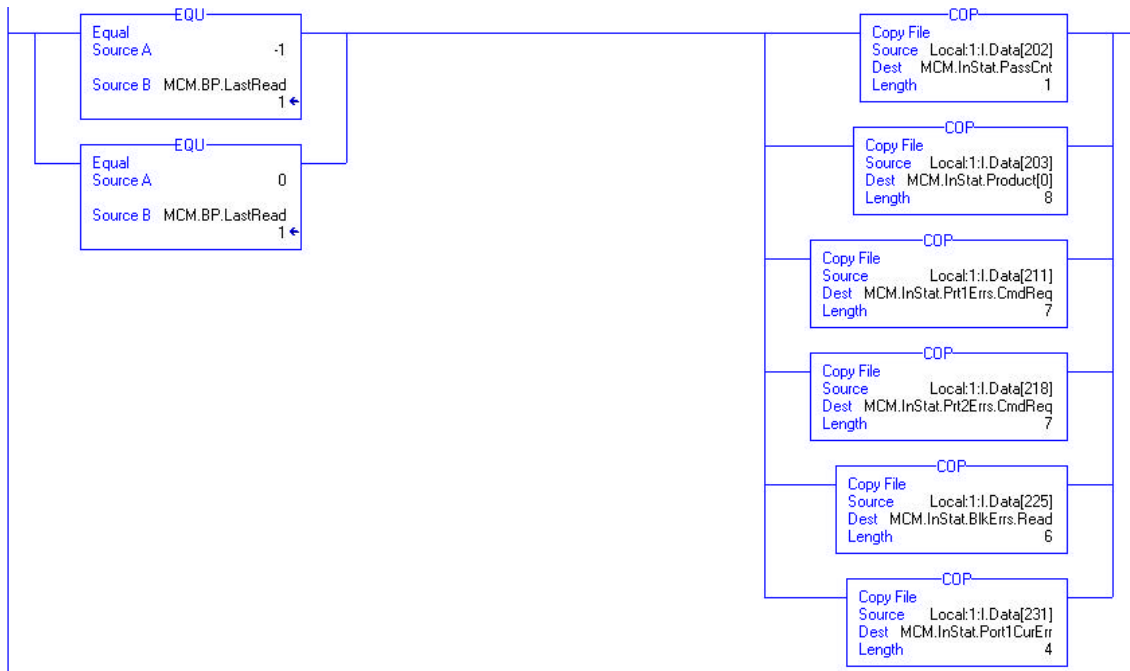


4.3 ReadData

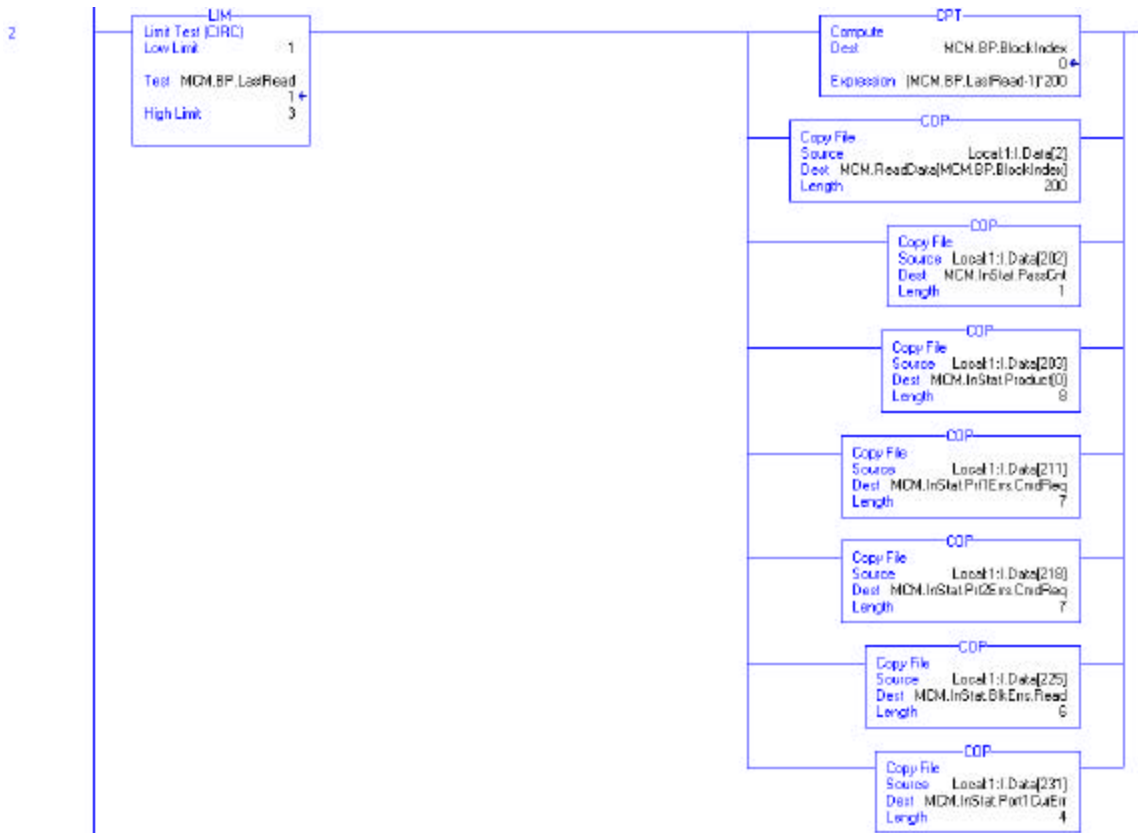
A tarefa **ReadData** é responsável por manusear todos novos dados recebidos do módulo e colocá-los em uma localização adequada no processador. O dado é transferido para o processador usando a imagem de entrada do módulo (**Local:1:I:Data[_]**). A primeira linha da tarefa coloca o último número de bloco de leitura (**MCM1.BP.LastRead**) no número de bloco corrente enviado a partir do módulo (**Local:1:I:Data[249]**).



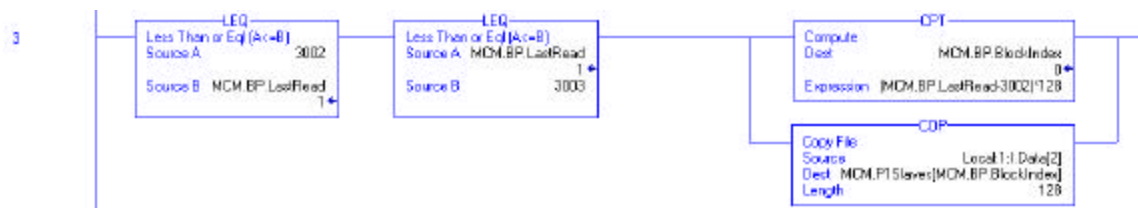
Se o módulo estiver configurado para um ou zero blocos, ele envia blocos com códigos de identificação de zero e -1. Nenhum dado de usuário será incluído nestes blocos. Eles apenas conterão os dados de status. A linha abaixo mostra a lógica usada para trabalhar estes blocos.



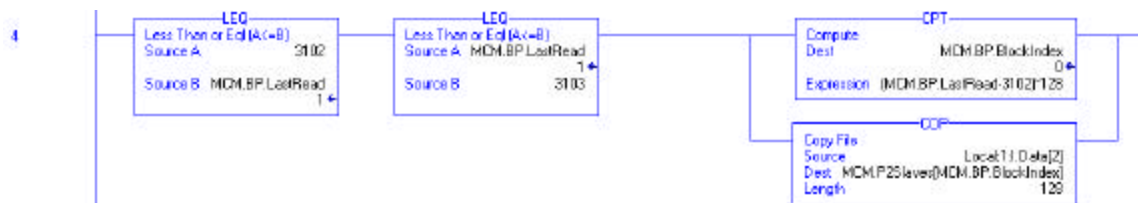
A próxima linha da Lógica Ladder determina se o novo dado recebido na imagem é dado de usuário. Se houver dado de usuário, a Lógica Ladder coloca o dado na localização correta na área de dado de leitura do processador (**MCM.ReadData[_]**). Até 200 palavras de dado podem ser transferidas em cada transferência de bloco. Adicionalmente ao dado de usuário, o bloco também contém importantes dados de status. Estes dados devem ser copiados na área de dado correta no módulo (**MCM.InStat**). Este dado de status pode ser usado para determinar a “saúde” do módulo MVI56-MCM.



As próximas duas linhas de Lógica Ladder são usadas para manusear a recepção do dado de status do nó escravo. Estes blocos são requeridos pelo processador no **WriteData** e enviados do módulo para o processador. As duas linhas abaixo mostram a lógica requerida para processar estes blocos.

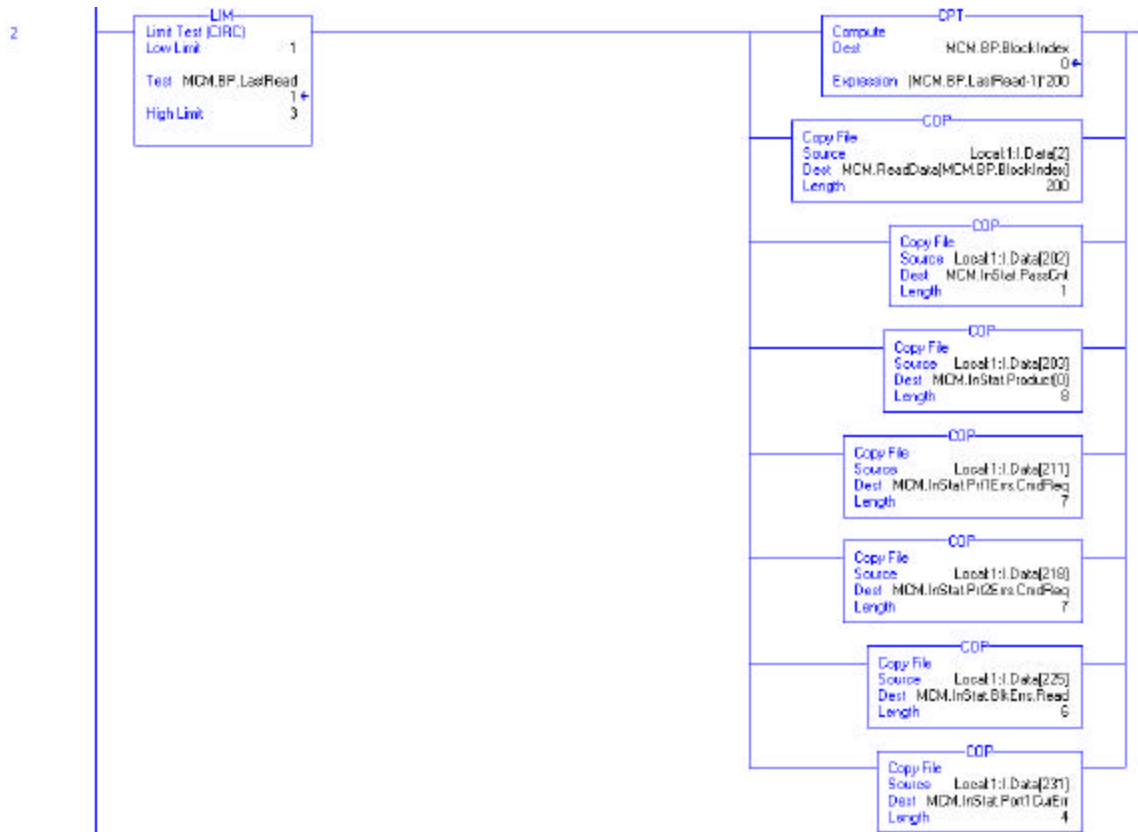


Esta linha é usada para trabalhar escravos anexos à Porta Modbus 1. Dois blocos de 128 escravos cada são processados pela linha e o dado é armazenado na localização apropriada do array.

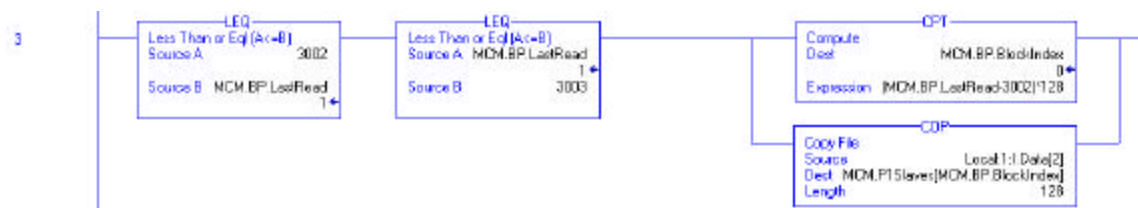


Esta linha é usada para manusear escravos anexos à Porta Modbus 2.

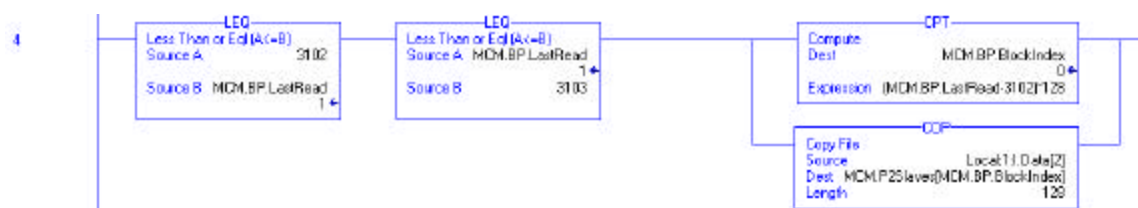
Se o processador deve receber a configuração do módulo de uma fonte remota através do banco de dados do módulo, ele deve ser programado para manusear blocos especiais. A informação de configuração é transferida do módulo para o processador através dos blocos -9000, -6000 a -6003 e -6100 a -6103. A Lógica Ladder para esta função está mostrada abaixo:



Esta linha é usada para processar a recepção da informação de configuração geral para o módulo.

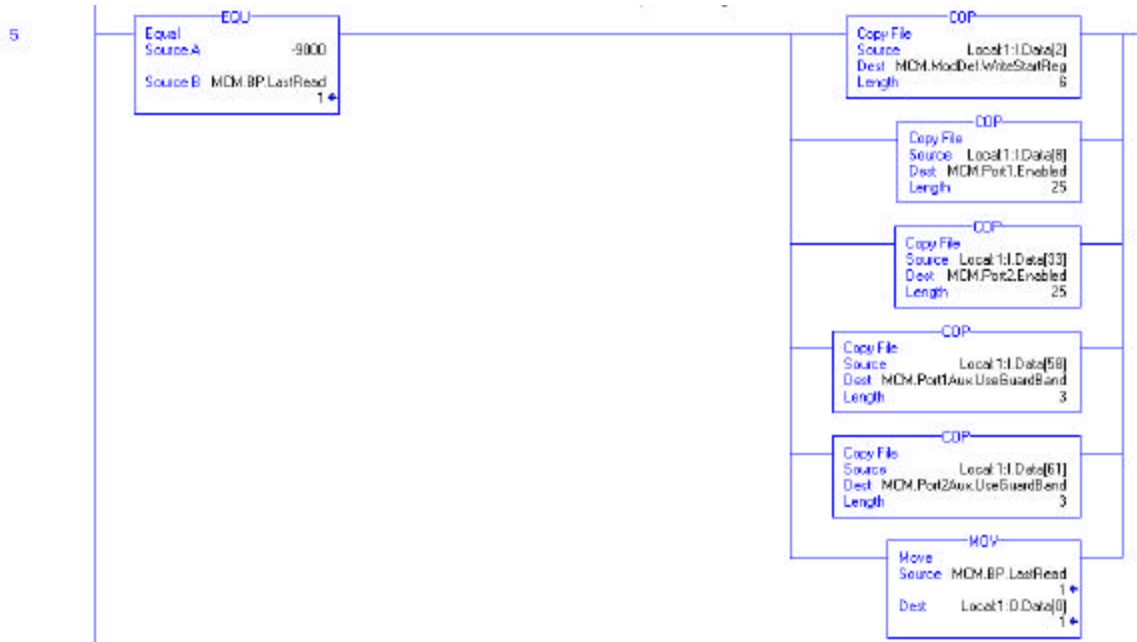


Esta linha é usada para manusear a recepção do dado da Lista de Comando Mestre para a Porta Modbus 2.

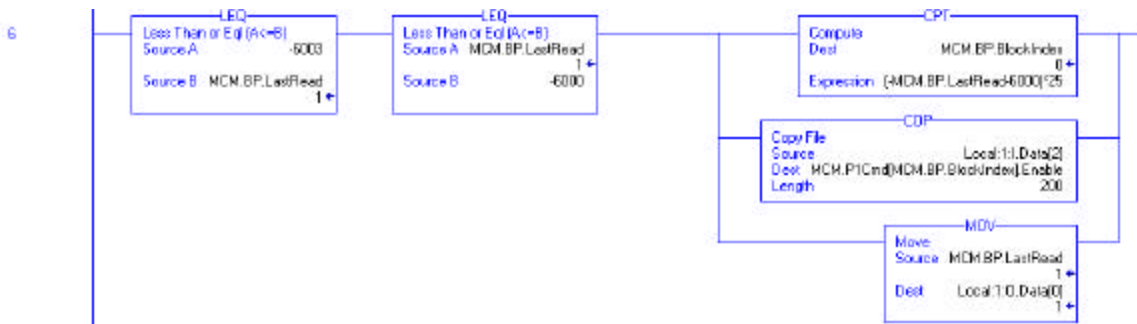


Esta linha é usada para manusear a recepção do dado da Lista de Comando Mestre para a Porta Modbus 3. Outros blocos poderão ser manuseados em futuros upgrades do produto.

A seguinte linha é usada para copiar uma mensagem que foi transferida para o processador diretamente do host remoto através do módulo (modo Pass_Through).

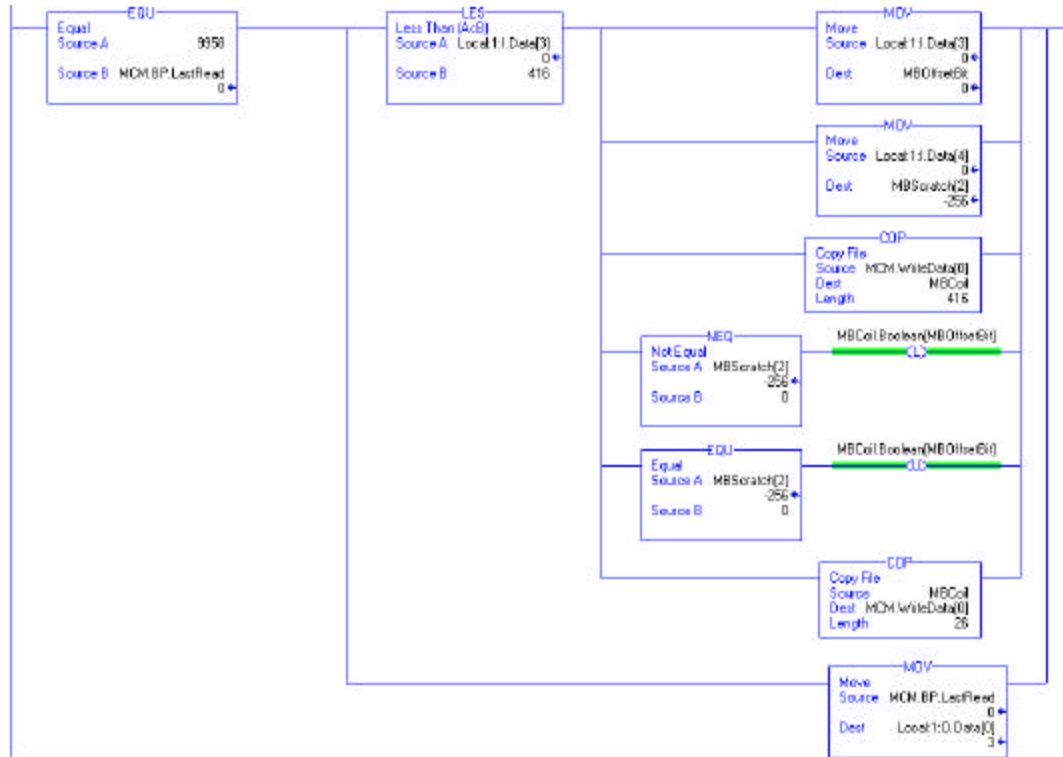


Esta linha é usada para manusear pedidos de Código de Função 6 e 16, quando o módulo estiver sendo usado no modo Pass_Through formatado.



Esta linha é usada para manusear pedidos de Código de Função 5, quando o módulo estiver sendo usado como escravo no modo Pass_Through formatado.

10



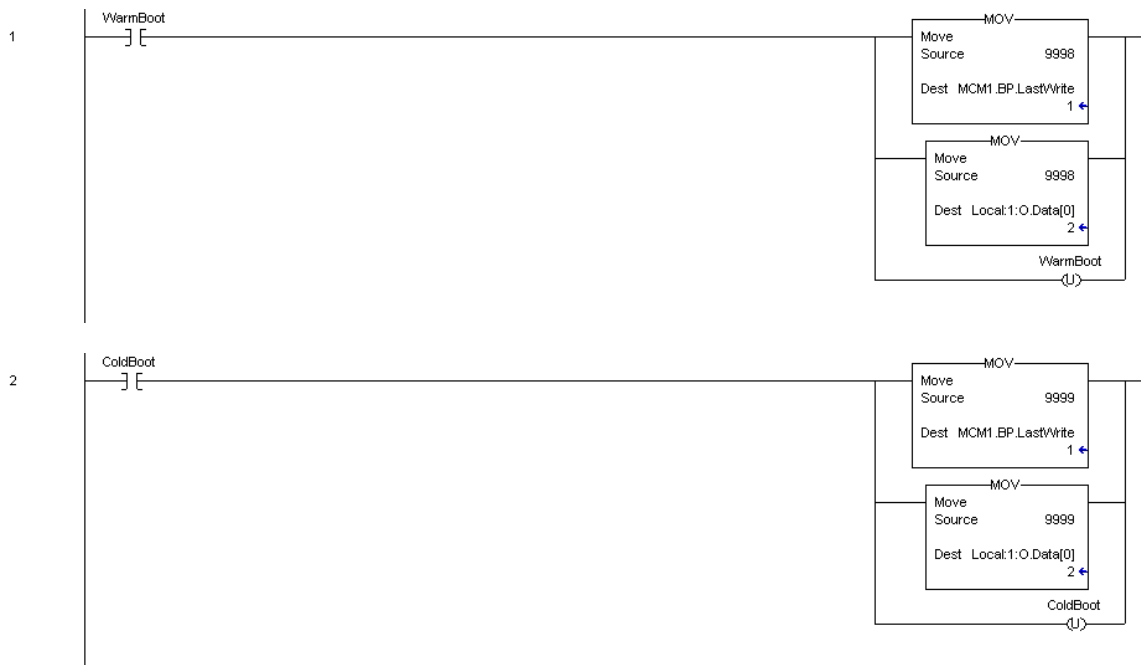
Nota: A linha 11 (não mostrada) é usada para a lógica Pass_Through para o código de função 15. Esta linha não deve ser alterada.

4.4 WriteData

A tarefa **WriteData** é responsável pelo envio de dado do processador para o módulo MVI56-MCM. O dado é transferido do processador para o módulo usando a imagem de saída corrente (**Local:1:O:Data[_]**). A primeira linha é usada para armazenar o dado correntemente requisitado no objeto de dado **MCM.BP.LastWrite**. Este objeto é usado em toda a Lógica Ladder subsequente no caso de a palavra de entrada (**Local:1:I:Data[1]**) mudar durante o processamento.



As próximas duas linhas são usadas para manusear o controle do processador do módulo usando os números de bloco de controle de boot a quente e boot a frio. Quando o processador requer que o módulo realize uma destas operações, ele simplesmente copia o número de bloco na imagem de saída do módulo e o módulo realiza a operação. Se certifique de colocar o número de bloco requerido no último objeto de escrita para impedir um processamento adicional na tarefa **WriteData**. Exemplos de cada controle são providos nas linhas seguintes:



As próximas quatro linhas são usadas para o dado de status de nó escravo associado a cada porta mestre. Dois pedidos são requeridos para cada porta para obter o dado para os 256 endereços potenciais em uma porta. A seguinte Lógica Ladder mostra o que é requerido para obter o dado para a Porta Modbus 1.



As próximas duas linhas mostram a lógica para o dado de status/ controle de escravo da Porta Modbus 2.

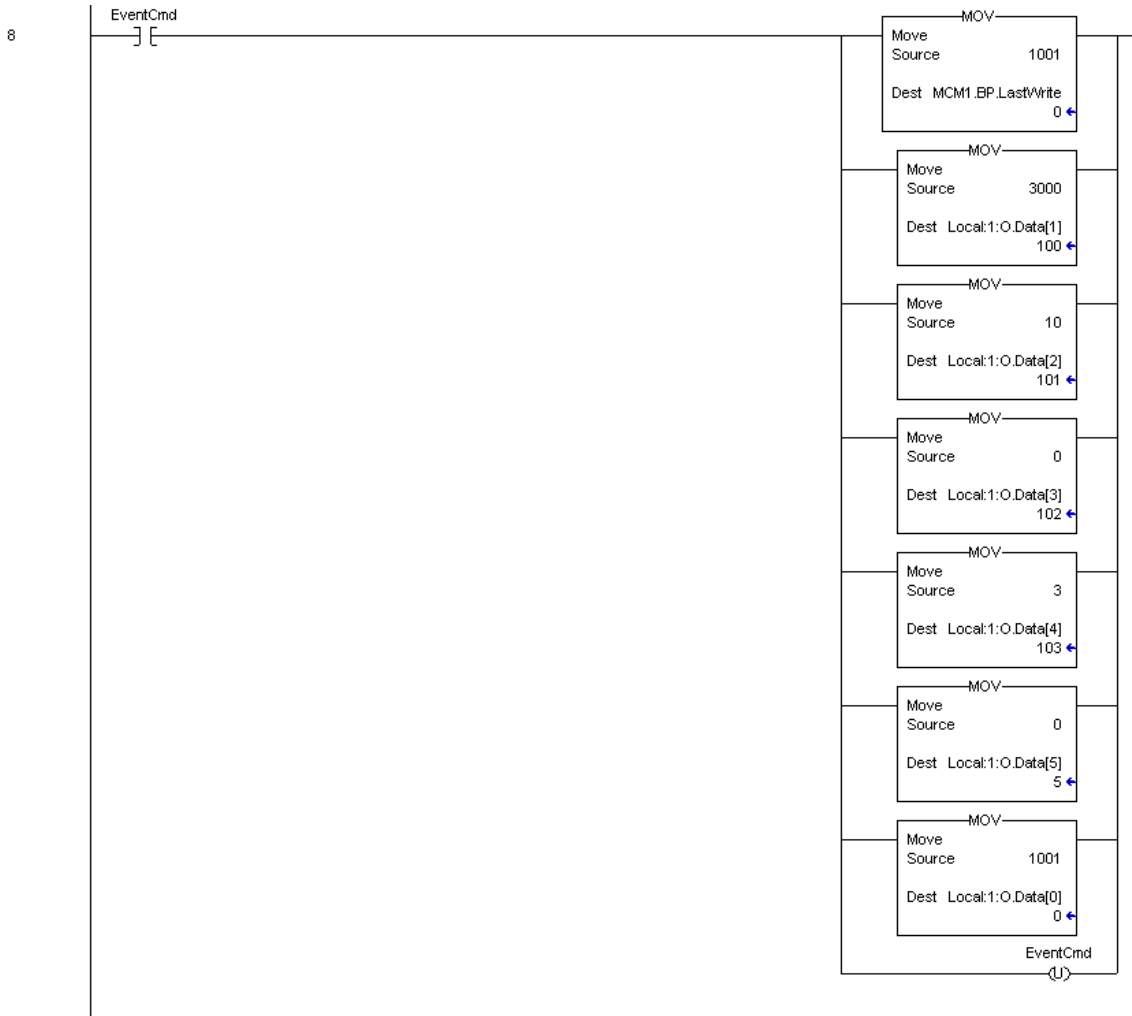


A próxima linha mostra um controle de comando de exemplo. Este bloco de dado é transferido do processador para o módulo para executar um comando em uma Lista de Comando da porta mestre.



Com o bit **CmdControl** colocado, o comando mestre da Porta 1 (índice 0) será colocado na fila de comando e executado. Até seis comandos podem ser transferidos da Lista de Comando para a fila de comando com um pedido..

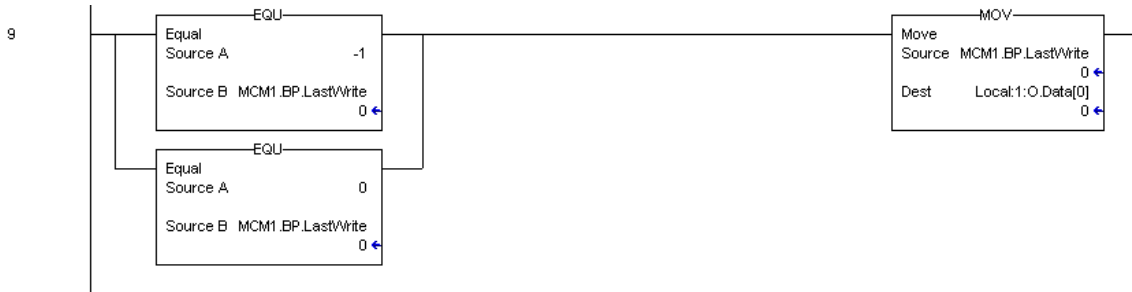
A próxima linha é usada como mensagem de evento (mensagem construída por usuário) na Porta 1.



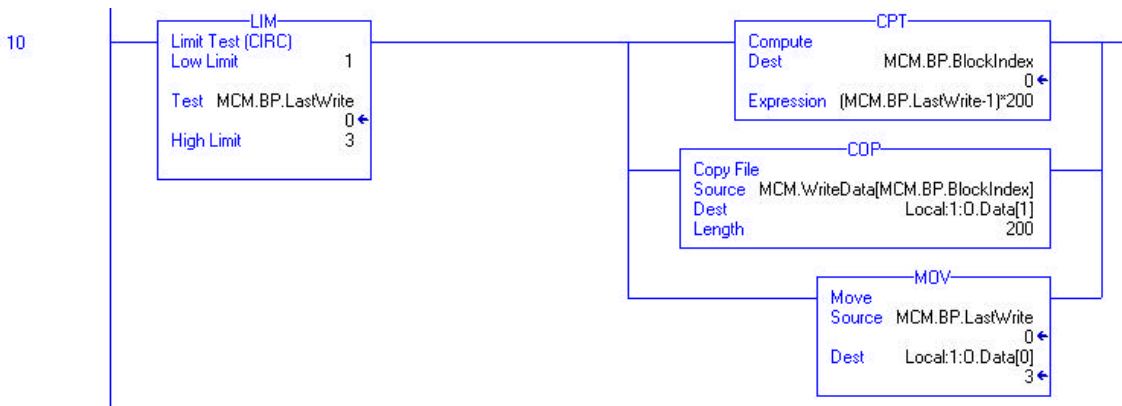
Com o bit **EventCmd** colocado, a linha será executada. Ele coloca o comando contido na linha na fila de comando para ser executado. Esta técnica pode ser usada para emitir comandos em uma porta sem construir uma Lista de Comando Mestre, ou executar comandos que devem ser emitidos em condições especiais (i.e., modo reset executado uma vez por dia, semana, etc.).

Se o módulo estiver configurado para nenhuma ou apenas uma transferência de bloco, será requerido um processamento especial. O módulo deve observar a primeira palavra da imagem de saída do módulo mudando de ordem para reconhecer a recepção do novo dado. Se o valor nunca muda, o módulo não processa o dado. Isto causa um problema quando menos que dois blocos devem ser transferidos para o módulo a partir do processador. Para superar este problema, o módulo envia -1 e 0 na palavra de entrada. Quando o módulo estiver configurado para zero blocos de escrita, a seguinte seqüência de pedido de bloco será usada: 1, 0, 1, 0, 1, 0.

A linha a seguir é requerida para manusear estas condições:

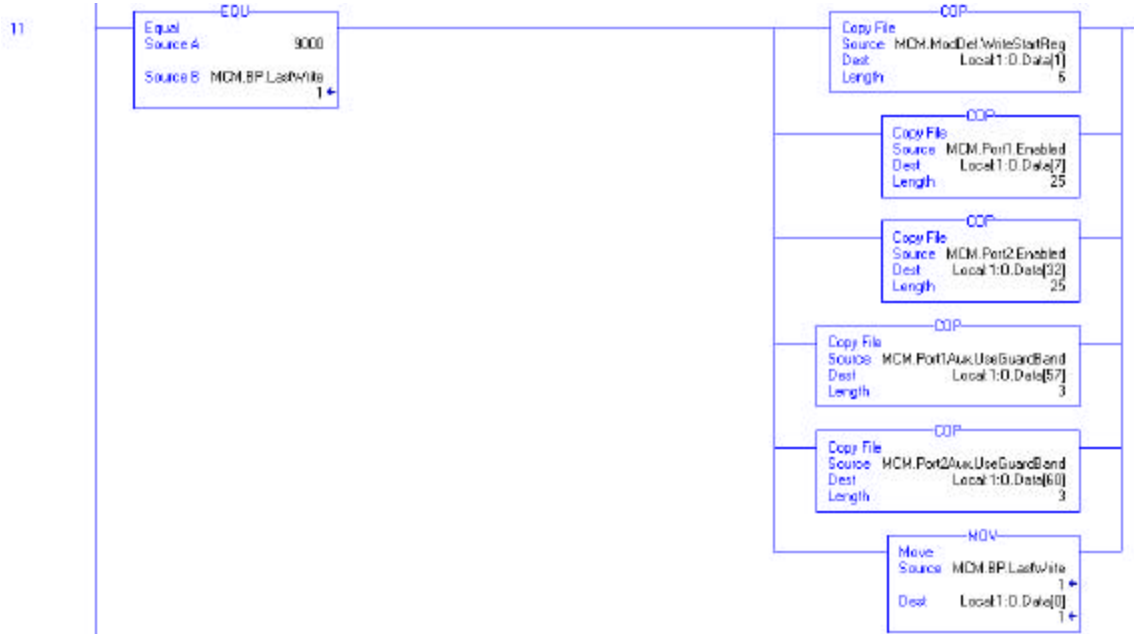


A próxima linha na Lógica Ladder é a mais importante. Ela manuseia a transferência do dado do processador para o módulo. Até 200 palavras de dado de usuário no processador (**MCM.WriteData[]**) podem ser transferidas para o módulo de uma vez.

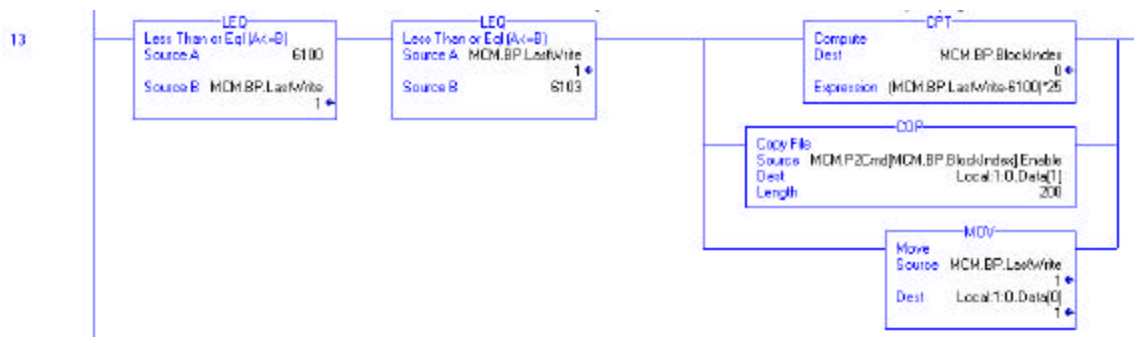
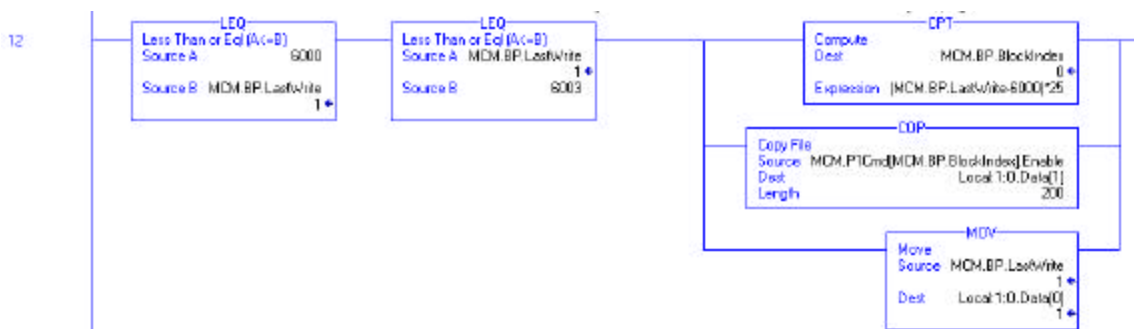


Para configurar o módulo, a configuração deve ser transferida da tabela de dado do módulo para o módulo. Diversos blocos são requeridos para transferir todas as informações requeridas pelo módulo. Cada um destes módulos deve ser programado e manuseado para o módulo ser executado.

O primeiro bloco de configuração tem um valor de código igual a 9000. este bloco é usado para transferir informação de tamanho de bloco de dado (**MCM.ModDef**) e informação de configuração de Porta Modbus (**MCM.Port[]**). Este é o primeiro conjunto de dados requerido pelo módulo no boot.



O último conjunto de informação de configuração requerida a partir do módulo é a Lista de Comando Mestre para cada porta. Esta lista é transferida para o módulo a uma razão de 25 comandos por vez. A Lógica Ladder para transferir a Lista de Comando para o módulo está mostrada nas linhas que se seguem:



5. Diagnóstico e Solução de Problemas

O módulo provê informação de diagnóstico em três formas para o usuário

- 1 Valores de dados transferidos do módulo para os arquivos de dados definidos pelo processador ControlLogix.
- 2 Todos os dados contidos no módulo podem ser vistos através da porta Configuração / Debug a um emulador de terminal anexo.
- 3 Indicadores de status LED na frente do módulo fornecem informação com respeito ao status dos módulos.

As seções a seguir explicam como obter dados de status do módulo e o significado de cada LED no módulo.

5.1 Lendo Dados de Status do Módulo

O módulo MVI56-MCM retorna um bloco de Dados de Status de 29 palavras que pode ser usado para determinar o status de operação do módulo. Este dado está localizado no banco de dados do módulo nos registros 6670 a 6698 na localização especificada na configuração. Este dado é transferido para o processador ControlLogix continuamente com cada bloco de leitura.

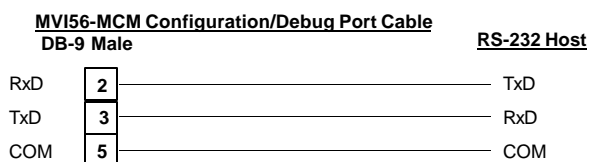
Para ver a listagem completa do objeto de dado de status, dirija-se à seção **Setup de Módulo**.

5.1.1 Hardware Requerido

Os requisitos de hardware para interfacear a porta de Configuração/ Debugger não são muito exigentes. Um microcomputador com porta serial já seria suficiente. Para um desempenho ótimo, o mínimo requerido seria:

Processador tipo 80468 (Pentium de preferência)
1 megabyte de memória
pelo menos uma porta de comunicação serial disponível.

Adicionalmente, um modem de cabo é fornecido com o módulo para prover conexão entre PC e porta. A porta do módulo tem um conector macho DB-9 na extremidade de um Pig_Tail RJ_45 para DB9. A extremidade RJ_45 do cabo deve ser colocada no conector de porta 1 do MVI56-MCM (porta superior). O cabo requerido está mostrado no diagrama abaixo:



5.1.2 Software Requerido

O software requerido no PC para interfacear a porta Configuração/ Debugger depende do sistema operacional. Softwares testados incluem:

DOS	ProComm, PS-Term e outros programas para emulação de terminal
Window 3.1	Terminal
Window 95/98	Hyper Terminal e PS Term
Window NT	Hyper Terminal
LINUX	MInicom

Qualquer software emulador de terminal ASCII provido com seu sistema operacional deve funcionar desde que configurado da seguinte forma:

Baud_Rate	57600
Paridade	Nenhum
Data Bits	8
Stop Bits	1
Protocolo de Transferência de Arquivo	Zmodem

5.1.3 Usando a Porta

As seguintes etapas são requeridas para interfacear Porta Configuração/ Debugger

- 1 Conecte se computador à porta do módulo usando um cabo modem
- 2- Inicie o programa emulador de terminal no computador e configure os parâmetros de comunicação de acordo com a seção de Software Requerido.
- 3 Digite o caractere “?” em seu computador. Se tudo OK, aparece o menu da porta

Se não houver resposta do módulo, verifique o setup de comunicação e o cabo. Adicionalmente, se assegure que conectou a porta correta no computador e no módulo

5.1.4 Opções de Menu

As facilidades disponíveis com o uso da porta Configuração/ Debugger no módulo MVI56-MCM são todos ativados usando teclas de atalho no computador. Há um único menu principal e diversos sub-menus na porta. Para ver as seleções disponíveis tecle “?”.

Se você estiver no menu Principal, o seguinte menu deve aparecer:

```
MODBUS MASTER/SLAVE COMMUNICATION MODULE (MVI56-MCM) MENU
?=Display Menu
A=Data Analyzer
B=Block Transfer Statistics
C=Module Configuration
D=Modbus Database View
Master Command Errors : E=Port 1   F=Port 2
Master Command List   : I=Port 1   J=Port 2
Slave Status List     : O=Port 1   P=Port 2
U=Version Information
W=Warm Boot Module
Y=Transfer Module Cfg to Processor
Communication Status : 1=Port 1   2=Port 2
Port Configuration   : 6=Port 1   7=Port 2

Esc=Exit Program
```

Se este menu não aparecer, tecle “M” para chamar o menu principal. Todas facilidades oferecidas pela Configuração / Debugger são mostradas no menu Principal. Cada opção será discutida nas seções que se seguem.

5.1.4.1 A= Analisador de Dado

A seleção desta opção de menu coloca o programa no modo de menu analisador. Este modo de operação é usado para mostrar mensagens Modbus geradas e recebidas pelo módulo.

Para ver as opções de menu disponíveis neste modo, tecle “?”, o seguinte menu deve aparecer.

```
Data Analyzer Mode Selected
MODBUS DATA ANALYZER VIEW MENU
?=Display Menu
1=Select Port 1
2=Select Port 2
5=1 mSec Ticks
6=5 mSec Ticks
7=10 mSec Ticks
8=50 mSec Ticks
9=100 mSec Ticks
0=No mSec Ticks
H=Hex Format
A=ASCII Format
B=Start
S=Stop
M=Main Menu

Port = 1, Format=HEX, Tick=10
```

Esta ferramenta é extremamente útil para determinar a operação do módulo e dos nós na rede de cada porta. Os parâmetros na base da tela mostram os atuais ajustes do analisador. Cada uma das opções de menu será discutida nas seções que se seguem.

5.1.4.1.1 1= Selecciona Porta 1

Esta opção selecciona Porta Modbus 1 para análise. Os dados mostrados no modo analisador, se referem à Porta 1.

5.1.4.1.2 2= Selecciona Porta 2

Esta opção selecciona Porta Modbus 2 para análise. Os dados mostrados, no modo analisador, se referem à Porta 2.

5.1.4.1.3 5= Tiques de 1 mSeg

Esta opção gera marcas de temporização de 1 milissegundo na tela. Isto pode ajudar a determinar as características de temporização de comunicação.

5.1.4.1.4 6= Tiques de 5 mSeg

Esta opção gera marcas de temporização de 5 milissegundos na tela. Isto pode ajudar a determinar as características de temporização de comunicação.

5.1.4.1.5 7= Tiques de 10 mSeg

Esta opção gera marcas de temporização de 10 milissegundos na tela. Isto pode ajudar a determinar as características de temporização de comunicação.

5.1.4.1.6 8= Tiques de 50 mSeg

Esta opção gera marcas de temporização de 50 milissegundos na tela. Isto pode ajudar a determinar as características de temporização de comunicação.

5.1.4.1.7 9= Tiques de 100 mSeg

Esta opção gera marcas de temporização de 100 milisegundos na tela. Isto pode ajudar a determinar as características de temporização de comunicação.

5.1.4.1.8 0=Sem Tiques de Temporização

Esta opção desliga a tela de marcas de temporização.

5.1.4.1.9 H= Formato Hexadecimal

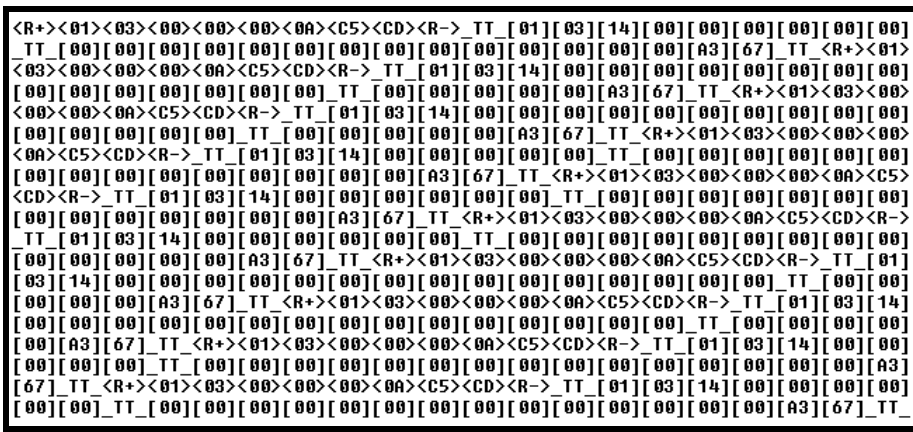
Esta opção seleciona a tela de dados no formato hexadecimal. Este formato é mais útil na visualização de mensagens no protocolo RTU Modbus.

5.1.4.1.10 A= Formato ASCII

Esta opção seleciona a tela de dados no formato hexadecimal. Este formato é mais útil na visualização de mensagens no protocolo ASCII Modbus.

5.1.4.1.11 B= Início

Esta opção inicia o analisador de dados. Depois de teclada, todos dados transmitidos e recebidos na porta correntemente selecionada serão mostrados. Uma tela de exemplo é mostrada a seguir:



A tabela abaixo descreve os caracteres especiais usados na tela:

[]	Dados nestes caracteres representam os dados recebidos na porta.
<>	Dados nestes caracteres representam os dados transmitidos na porta
<R+>	Estes caracteres são inseridos quando a linha RS é elevada na porta
<R->	Estes caracteres são inseridos quando a linha RS é baixada na porta.
<CS>	Estes caracteres são mostrados quando a linha CTS é reconhecida como elevada
TT	Estes caracteres são mostrados quando o intervalo de marca de temporização tiver sido alcançado. Este parâmetro é definido por usuário.

5.1.4.1.12 S= Stop

Esta opção interrompe o analisador. Use esta opção para congelar a tela de modo que o dado possa ser analisado. Para reiniciar o analisador, tecla "B".

Aviso- No modo analisador, a velocidade execução do programa será reduzida. Somente use esta ferramenta na sessão "**Solução de Problema**". Desabilite o analisador antes de voltar o módulo para o modo normal.

5.1.4.1.13 Menu Principal

O menu principal é usado para voltar para o modo de menu principal.

5.1.4.2 Estatísticas de Transferência de Blocos

Esta opção de menu mostra a configuração de estatística das operações de transferência de dados de barramento. Depois de selecionada esta opção, a tela abaixo é mostrada: Selecione esta opção em intervalos de 1 segundo para determinar o número de blocos transferidos a cada segundo.

```
BACKPLANE STATISTICS:
DATA TRANSFER CONFIGURATION:
WRITE DATA TRANSFER:
  Start : 0      Count : 400    Max Blocks : 2    Last : 2
READ DATA TRANSFER:
  Start : 0      Count : 600    Max Blocks : 3    Last : 2
BLOCK COUNTS:
  Retry : 20     Failed: 0      Fail Cnt: 0
  Read  : 43009  Write : 43021  Parsing : 43008
  Error : 59833  Event : 0      Command : 0
```

5.1.4.3 C= Configuração de Módulo

Esta opção mostra a informação de configuração de módulo geral para o módulo MVI56-MCM. Depois de selecionada a opção, aparece a seguinte tela:

```
MODULE CONFIGURATION:
MVI56-MCM, ProSoft Technology, Inc.
DATABASE:
  Err/Stat Blk Pointer : 600
BLOCK TRANSFER:
  READ  -- Start : 0      Count : 600    Max : 3
  WRITE -- Start : 0      Count : 400    Max : 2
  FAIL COUNT : 20
```

5.1.4.4 D= Visualização do Banco de Dados Modbus

A seleção desta opção coloca o programa no modo de menu de visualização de banco de dados. Este modo de opção mostra os valores disponíveis de banco de dados internos do módulo neste modo. Para ver as opções de menu disponíveis tecla "?" e aparece o seguinte menu:

```
MODBUS DATABASE VIEW MENU
?=Display Menu
0-6=Display 0-6000
S=Show Again
-=Back 5 Pages
P=Previous Page
+=Skip 5 Pages
N=Next Page
D=Decimal Display
H=Hexadecimal Display
F=Float Display
A=ASCII Display
M=Main Menu
```

Todos os banco de dados contidos no banco de dados do módulo ficam disponíveis para serem visualizados usando as opções de menu. Cada opção disponível do menu será discutida nas seções que se seguem.

5.1.4.5 0-9 Páginas de Registros 0-9000

Esta opção de menu avança para um conjunto específico de registros no banco de dados e mostra os dados. As teclas têm as seguintes funções:

Chave	Função
0	Mostra- Registros 0 to 99
1	Mostra- Registros 1000 to 1099
2	Mostra- Registros 2000 to 2099
3	Mostra- Registros 3000 to 3099
4	Mostra- Registros 4000 to 4099
5	Mostra- Registros 5000 to 5099
6	Mostra- Registros 6000 to 6099
7	Mostra- Registros 7000 to 7099
8	Mostra- Registros 8000 to 8099
9	Mostra- Registros 9000 to 9099

5.1.4.6 S= Mostra Novamente

Esta opção de menu mostra a página corrente de 100 registros no banco de dados. O exemplo mostra a tela de banco de dados no exemplo abaixo:

MODBUS DATABASE DISPLAY 0 TO 99 (DECIMAL)

```

0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0

```

5.1.4.6.1 - = Volta 5 Páginas

Esta opção de menu volta os 500 registros anteriores para visualizar e mostrar os dados.

5.1.4.6.2 P = Página Anterior

Esta opção de menu seleciona e mostra 100 registros anteriores de dados.

5.1.4.6.3 + = Avança 5 Páginas

Esta opção de menu avança 500 registros de dados e mostra a nova página de dados.

5.1.4.6.4 N = Próxima Página

Esta opção de menu seleciona os próximos 100 registros para visualização e mostra os dados.

5.1.4.6.5 D = Tela Decimal

Esta opção de menu mostra os dados da página atual no formato decimal.

5.1.4.6.6 H = Tela Hexadecimal

Esta opção de menu mostra os dados na página atual no formato hexadecimal.

5.1.4.6.7 F = Tela de Flutuante

Esta opção de menu mostra os dados na página corrente no formato de Ponto Flutuante. O programa assume que os valores estejam alinhados na extremidade do registro. Se os valores de Ponto Flutuante não estiverem alinhados, eles não serão mostrados apropriadamente.

5.1.6.4.8 A = Tela ASCII

Esta opção de menu mostra os dados na página atual no formato ASCII. Isto é útil para regiões do banco de dados que contêm dados ASCII.

5.1.6.4.9 M = Menu Principal

Esta opção de menu faz voltar para o modo de menu principal.

5.1.4.7 E e F=Erros de Comando Mestre (Portas 1 and 2)

Selecionando estas opções de menu coloca o programa no modo de menu de comando mestre para a porta especificada. Este modo de opção mostra múltiplas páginas de dados de erro/ status da lista de comando mestre. Para ver as opções de menu disponíveis neste modo tecle "?" e aparece o menu abaixo:

```
COMMAND ERROR LIST MENU (MASTER Port 1)
?=Display Menu
S=Show Again
-=Back 2 Pages
P=Previous Page
+=Skip 2 Pages
N=Next Page
D=Decimal Display
H=Hexadecimal Display
M=Main Menu
```

Cada opção de menu será discutida nas seções que se seguem.

5.1.4.7.1 S= Mostra Novamente

Esta opção mostra a página corrente dos dados de comando erro/ status. Selecionando esta opção, aparece a seguinte tela:

```
COMMAND ERROR LIST FOR PORT 1, COMMANDS 0 TO 19 (DECIMAL)
```

```
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
```

Cada valor mostrado na tela corresponde ao código erro/ status para o índice de Lista de Comando mestre associada. Dirija-se à seção **Setup de Módulo** onde pode ser vista a listagem completa dos códigos.

5.1.4.7.2 - = Volta 2 Páginas

Esta opção volta 20 comandos e mostra os dados

5.1.4.7.3 P= Página Anterior

Esta opção mostra a página anterior de dados.

5.1.4.7.4 + = Avança 2 Páginas

Esta opção avança os próximos 20 comandos e mostra os dados.

5.1.4.7.5 N= Próxima Página

Esta opção mostra a próxima página de dados erro de status da Lista de Comando Mestre.

5.1.4.7.6 D= Tela Decimal

Esta opção muda a tela de dados para formato decimal

5.1.4.7.7 H= Tela Hexadecimal

Esta opção muda a tela de dados para formato hexadecimal

5.1.4.7.8 M= Menu Principal

Esta opção volta o programa para modo de menu principal.

5.1.4.8 I e J= Lista de Comando Mestre (Portas 1 e 2)

Selecionar esta opção de menu coloca o programa no modo de menu da Lista de Comando Mestre para a porta especificada. Este modo de opção é usado para mostrar múltiplas páginas de dados de Lista de Comando Mestre. Para visualizar as opções de menu disponíveis neste modo tecle “?” e aparece o menu abaixo:

```
MASTER COMMAND LIST MENU (Port 1)
?=Display Menu
S=Show Again
-=Back 5 Pages
P=Previous Page
+=Skip 5 Pages
N=Next Page
M=Main Menu
```

Cada opção no menu será discutida nas seções que se seguem.

5.1.4.8.1 S= Mostra novamente

Esta opção mostra a página corrente de comandos mestre. 10 comandos são mostrados em cada página, como mostrado no seguinte exemplo:

COMMAND LIST FOR PORT 1-- COMMANDS 0 TO 9								
EN	MBREG	POLLINT	COUNT	SWAP	NODE	FUNC	ADDRS	LASTERR
1	400	0	10	0	1	3	0	0X0000
0	0	0	0	0	0	0	0	0X0000
0	0	0	0	0	0	0	0	0X0000
0	0	0	0	0	0	0	0	0X0000
0	0	0	0	0	0	0	0	0X0000
0	0	0	0	0	0	0	0	0X0000
0	0	0	0	0	0	0	0	0X0000
0	0	0	0	0	0	0	0	0X0000
0	0	0	0	0	0	0	0	0X0000
0	0	0	0	0	0	0	0	0X0000

5.1.4.8.2 - = Volta 5 Páginas

Esta opção de menu mostra os dados da Lista de Comando Mestre depois de voltar 50 comandos anteriores.

5.1.4.8.3 P = Página Anterior

Esta opção de menu mostra a página anterior de dados da Lista de Comando Mestre.

5.1.4.8.4 += Avança 5 Páginas

Esta opção de menu mostra os dados de Lista de Comando Mestre depois de avançar os próximos 50 comandos.

5.1.4.8.5 N= Próxima Página

Esta opção de menu mostra a próxima página de dados da Lista de Comando Mestre.

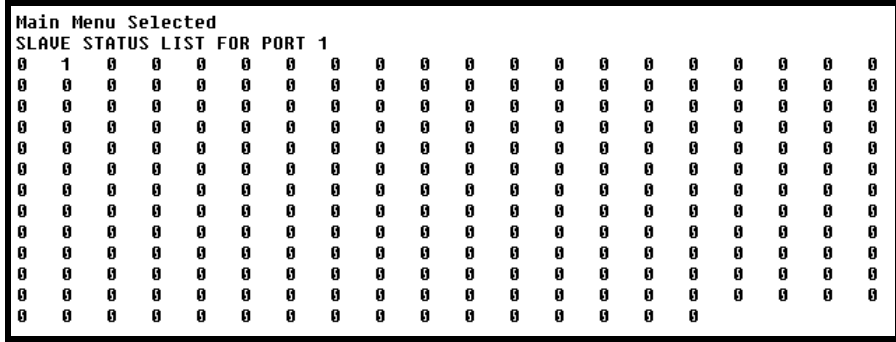
5.1.4.8.6 M= Menu Principal

Esta opção faz voltar para o modo de menu principal de operação.

5.1.4.9 O e P = Lista de Status de Escravo

Selecionar estas opções de menu mostra os 256 valores de status escravo associados às portas. Os valores mostrados têm as seguintes definições:

- 0_ escravo não é usado
- 1_ escravo ativamente sofrendo varredura,
- 2_ escravo suspenso e
- 3_ escravo desabilitado.



```
Main Menu Selected
SLAVE STATUS LIST FOR PORT 1
0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

5.1.4.10 V= Informação de Versão

Esta opção mostra a versão corrente do software do módulo e outros importantes valores. Selecionando a opção, aparece a tela abaixo:



```
VERSION INFORMATION:
MODBUS MASTER/SLAVE COMMUNICATION MODULE (MUI56-MCM)
(c) 1999, ProSoft Technology, Inc.
PRODUCT NAME CODE      : MCM
SOFTWARE REVISION LEVEL : 0.00
OPERATING SYSTEM REVISION : 0100
RUN NUMBER              : 2602
PROGRAM SCAN COUNTER    : 26961
```

Esta informação pode ser solicitada quando um suporte técnico para o produto é chamado. Os valores na base da tela são importantes para determinar a opção do módulo. O valor **Program Scan Counter** é incrementado cada vez que se completa um ciclo de programa do módulo. Este valor pode ser usado para determinar a frequência da execução do programa tecendo "V" para intervalos de 1 segundo.

5.1.4.11 W= Módulo de Boot A Quente

Esta opção é usada quando uma opção de boot a quente é requerida do módulo. Esta solicitação é usualmente feita depois de as mudanças de configuração terem sido feitas na área de dados de Tags Controladores do processador ControlLogix para implementar as mudanças. Depois de selecionar a opção, aparece a tela abaixo.

```

Press 'Y' key to confirm warm boot!

Warm booting module....

Reloading Program Values....

Read Configuration...
Read Port 1 Command Configuration...
MODBUS MASTER/SLAVE COMMUNICATION MODULE (MUI56-MCM)
(c) 1999, ProSoft Technology, Inc.

    PRODUCT NAME CODE       : MCM
    SOFTWARE REVISION LEVEL  : 0.00
    OPERATING SYSTEM REVISION : 0100
    RUN NUMBER               : 2602

Press ? for menu help.

```

5.1.4.12 Y= Transfere Cfg do Módulo para o Processador.

Esta opção transfere dados de configuração correntes no módulo para o processador ControlLogix. A Lógica Ladder deve estar no processador para esta opção ser implementada com sucesso. Selecionando esta opção, aparece a seguinte tela indicando que a operação foi bem sucedida:

```

Press 'Y' key to confirm transfer option!

Sending configuration to processor....

Send complete....Error code = 0.

Warm booting module....

Reloading Program Values....

Read Configuration...
Read Port 1 Command Configuration...
MODBUS MASTER/SLAVE COMMUNICATION MODULE (MUI56-MCM)
(c) 1999, ProSoft Technology, Inc.

    PRODUCT NAME CODE       : MCM
    SOFTWARE REVISION LEVEL  : 0.00
    OPERATING SYSTEM REVISION : 0100
    RUN NUMBER               : 2602

Press ? for menu help.

```

Se a operação não foi bem sucedida, retorna um código de erro. Os códigos de erro são:

Código	Descrição
0	Transferência bem sucedida
-1	Erro na transferência de dados de configuração de módulo. (bloco-9000)
-2	Erro na transferência de dados da lista de comando mestre para Porta 1 (blocos -6000 to -6003)
-3	Erro na transferência de dados da lista de comando mestre para Porta 2 (blocos -6100 to -6103)

Depois de uma transferência bem sucedida, o módulo realiza uma operação de boot a quente para leitura no novo dado.

5.1.4.13 1 e 2= Status de Configuração (Portas 1 e 2)

Esta opção mostra o status de comunicação e estatística da porta Modbus especificada. Esta informação pode ser útil na busca de solução de problemas de rede. Selecionando a opção, aparece a seguinte informação:

```
PORT 1 MODBUS STATUS:
Enabled : Y
Retries : 0      Cur Cmd : 0      State : 100
ComState: 0

Number of Command Requests: 5431
Number of Cmd Responses : 5430
Number of Command Errors : 0
Number of Requests : 5430
Number of Responses : 5430
Number of Errors Received : 0
Number of Errors Sent : 0
```

5.1.4.14 6 e 7= Configuração de Porta (Portas 1 e 2)

Estas opções são usadas para mostrar informação de configuração para a porta Modbus selecionada. Selecionando a opção, aparece a seguinte informação:

```
PORT 1 CONFIGURATION:
Enabled : Y      Port Type : (0) - MASTER
SLAVE SETUP:
  Modbus Slave ID: 0      Pass-Through = DISABLED
  Offsets:
    BitIn: 0      WordIn: 0      Output: 0      Holding: 0
  Floating-point Data:
    Flag: N      Start: 0      Offset: 0
  Route Count : 0
  Function 99 Offset : 0
  Use Packet Gap : N      Packet Gap Delay : 0
MASTER SETUP:
  Command Count : 3      Cmd Delay: 0      Cmd Offset : 350
  Response Timeout: 1000      Retries : 0      Delay Count: 0
COMMUNICATION PARAMETERS:
  Protocol: 1 (Modbus ASCII)
  Baud: 38400      Parity: NONE      Databits: 8      Stopbits: 1
  RTS On: 0      RTS Off: 0      Use CTS Line: N
```

5.1.4.15 Esc= Programa de Saída

Esta opção faz sair do programa e mostra o prompt do sistema operacional. Esta opção deve ser selecionada somente se instruída pelo Grupo de Suporte Técnico da ProSoft. Quando você seleciona esta opção, o módulo cessa operação. Os dados não serão mais transferidos entre as portas Modbus e o módulo e entre o processador ControlLogix e o módulo. Isto pode causar um transtorno ao programa em execução.

5.2 LEDs Indicadores de Status

Os LEDs indicam o status de operação do módulo como segue:

Módulo ProSoft	Cor	Status	Indicação
P1	Verde	On	Dados estão sendo transferidos entre o módulo e terminal remoto usando a porta Configuração/ Debug
		Off	Nenhum dado está sendo transferido na porta Configuração/ Debug
P2	Verde	On	Dados estão sendo transferidos entre o módulo e a rede Modbus em sua Porta Modbus 2
		Off	Nenhum dado está sendo transferido na porta.
P3	Verde	On	Dados estão sendo transferidos entre o módulo e a rede Modbus em sua Porta Modbus 3.
		Off	Nenhum dado está sendo transferido na porta.
APP	Ambar	On	MVI56-MCM está operando normalmente
		Off	O programa do módulo MVI56-MCM reconheceu um erro de comunicação em uma das portas Modbus
BP ACT	Ambar	On	O LED está aceso, quando o módulo realiza uma operação de escrita no barramento.
		Off	O LED está apagado quando o módulo realiza uma operação de leitura no barramento. Em condições normais, o LED deve piscar rapidamente
OK	Verm// Verde	Off	A placa não energiza e não está firmemente conectada ao rack
		Verde	O módulo está operando normalmente
		Verm	O programa detectou um erro ou está sendo configurado. Se o LED permanece vermelho mais que 10 segundos, o programa provavelmente travou. Remova e reinsira a placa do rack para reiniciar o programa do módulo.
BAT	Verm	Off	A voltagem da bateria está OK e a bateria está funcionando
		On	A voltagem da bateria está baixa ou não tem bateria. Substitua a bateria

Durante configuração do módulo, OK será vermelho e os LEDs APP e BP ACT devem ficar acesos. Se os LEDs ficarem presos neste modo por um longo período de tempo, verifique as palavras de erro de configuração no bloco de pedido de configuração.

A estrutura dos blocos está mostrada abaixo:

Offset	Descrição	Comprimento
0	Reservado	1
1	9000	1
2	Erros de Configuração de Módulo	1
3	Erros de Configuração de Porta 1	1
4	Erros de Configuração de Porta 2	1
5 – 248	Reserva	244
249	-2 ou -3	1

Os bits em cada palavra de configuração estão mostrados na seguinte tabela. A palavra de erro de configuração de módulo tem a seguinte definição:

Bit	Descrição	Valor
0	O valor de início de bloco de leitura é maior que o tamanho do banco de dados.	0x0001
1	O valor de início do bloco de leitura é menor que zero	0x0002
2	Quantidade de blocos de leitura menor que zero	0x0004
3	O valor início + quantidade de blocos de leitura maior que o tamanho do banco de dados	0x0008
4	O valor de início do bloco de escrita maior que o tamanho do banco de dados	0x0010
5	O valor de início do bloco de escrita menor que zero	0x0020
6	Quantidade de blocos de escrita menor que zero	0x0040
7	Quantidade + Início de blocos de escrita maior que o tamanho do banco de dados	0x0080
8		0x0100
9		0x0200
10		0x0400
11		0x0800
12		0x1000
13		0x2000
14		0x4000
15		0x8000

As palavras de erro de configuração de porta têm as seguintes definições:

Bit	Descrição	Valor
0	Código de tipo não é válido. Entre com valor 0 (mestre) a 1 (escravo).	0x0001
1	O parâmetro Flag de Flutuante (Float Flag) não é válido	0x0002
2	O parâmetro Flag de Flutuante de início não é válido	0x0004
3	O parâmetro Offset de Flag de Flutuante não é válido	0x0008
4	O parâmetro Protocolo não é válido	0x0010
5	O parâmetro Baud_Rate não é válido	0x0020
6	O parâmetro Paridade não é válido	0x0040
7	O parâmetro Data_Bits não é válido	0x0080
8	O parâmetro Stop_Bits não é válido	0x0100
9	ID escravo não é válido	0x0200
10	bit ou palavra de entrada, palavra de saída e/ou offset de registro armazenado não são válidos	0x0400
11	O parâmetro Command_Count não é válido	0x0800
12	Reserva	0x1000
13	Reserva	0x2000
14	Reserva	0x4000
15	Reserva	0x8000

Corrija qualquer dado inválido na configuração para uma operação de módulo apropriada. Quando a configuração contém um conjunto de parâmetros válidos, todos os bits nas palavras de configuração serão eliminados. Isto não indica que a configuração é válida para a aplicação do usuário. Fique certo que cada parâmetro esteja corretamente colocado para aplicação específica. Se os LEDs APP, BP ACT e OK piscam rapidamente (a cada segundo), chame a ProSoft. Há um problema sério com o módulo e ele deverá ser enviado para ProSoft.

5.2.1 Eliminando Condição de Falha

Tipicamente, se o LED OK na frente do módulo ficar vermelho por 10 segundos, um problema de hardware foi detectado no módulo ou o programa caiu. Para reparar esta condição:

- 1_ Remova e reinsira a placa no Rack
- 2_ Verifique os dados de configuração que estão sendo transferidos para o módulo a partir do processador ControlLogix.

Se o LED do módulo não passa para verde, se assegure que o módulo está inserido corretamente no Rack e um programa Ladder válido foi baixado no processador (o processador deve estar no modo RUN). Se isto não resolver, contate a fábrica.

5.2.2 Solução de Problemas

A tabela abaixo tem o propósito de ajudá-lo a solucionar problemas no módulo.

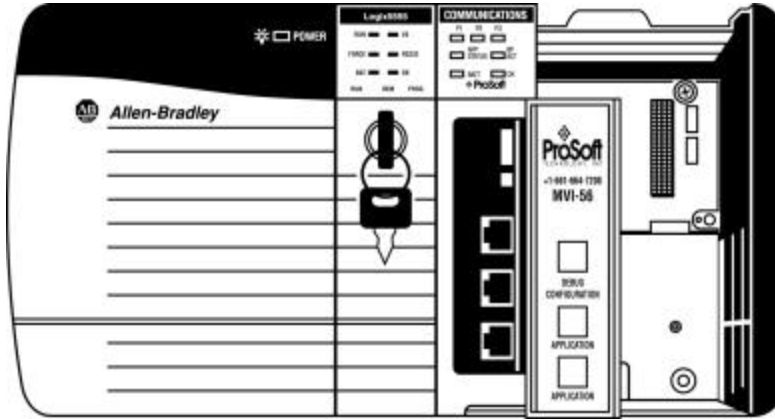
Por favor, use esta tabela para tentar corrigir problemas, mas se houver qualquer dúvida, não hesite em nos contactar.

Descrição do Problema	Ações a serem tomadas
Falha do processador	1_ Assegure-se que o módulo está conectado no slot configurado para o módulo MVI56-MCM. 2_ Assegure-se que o setup do slot na configuração do rack foi feito corretamente
LED E/S do processador pisca	Isto indica que há um problema na comunicação do barramento. Fique certo que este e todos os módulos no Rack estão configurados no processador.
LED BP ACT permanece apagado ou pisca lentamente.	Isto indica uma falha nas operações de transferência do barramento. Use a facilidade porta Configuração/Debug para estabelecer comunicações no barramento, se assegure que: 1_ O driver do barramento está carregado no módulo. 2_ O módulo está configurado para transferir dados de bloco de leitura e escrita. 3_ A Lógica Ladder lida com todas situações dos blocos de leitura e escrita. 4_ O módulo está configurado no processador.
LED OK permanece apagado	O programa travou ou ocorreu um erro crítico. Conecte a porta Configuração/Debug para ver se o módulo está executando. Se o programa travou, remova e reinsira a placa no Rack

6 Conexões de Cabo

O módulo MVI56-MCM tem as seguintes conexões de comunicação no módulo:

- Duas portas de comunicação Modbus (Conector RJ45).
- Uma porta de Configuração/Debug RS_232 (Conector RJ45).



6.1 Porta de Comunicação Modbus

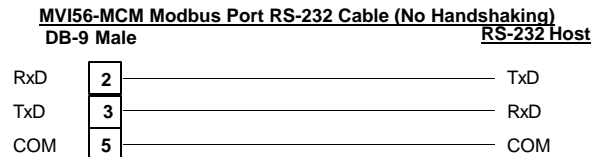
O módulo MVI56-MCM tem dois conectores Modbus físicos, com um plugue RJ45 localizado na frente do módulo.

6.1.1 Conectando Cabo ao Conector

A ProSoft provê dois Pig_Tails RJ45 para DB_9 para permitir uma interface simples com outros dispositivos. As Portas Modbus do módulo podem ser configuradas para operar no modo RS_232, RS_422, ou RS_485. A interface a ser associada à porta é ajustada com jumpers no módulo. Há um jumper para cada uma das duas portas. Adicionalmente, o uso de linhas de controle do modem é definido pelo usuário configurando as variáveis MCMPORT1.UseCTS e MCMPORT2.UseCTS para o valor de um (1). As seções que se seguem descrevem cada interface.

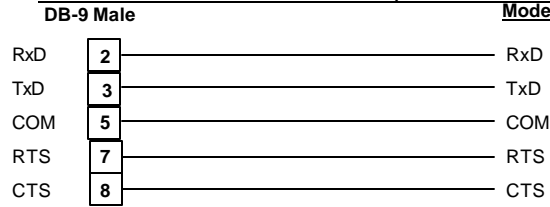
6.1.1.1 RS_232

Quando a interface RS_232 for usada, o uso de linhas de controle de modem será definido por usuário. Se não forem usadas linhas de controle de modem, o cabo para conectar a porta deve ficar como mostrado no exemplo abaixo:



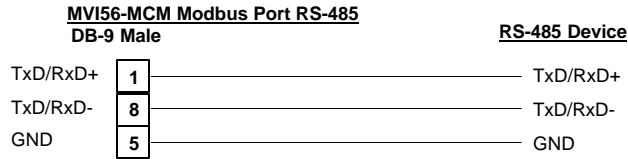
A linha RTS é controlada pelo RTS com parâmetros ON e OFF ajustados para a porta. Se a linha CTS for usada (usualmente requerida para um modem half-duplex), as linhas RTS e CTS devem ser conectadas uma a outra ou ao modem. O diagrama abaixo mostra o cabo requerido quando a porta é conectada ao modem.

MVI56-MCM Modbus Port RS-232 Cable (Use CTS Line and Modem)



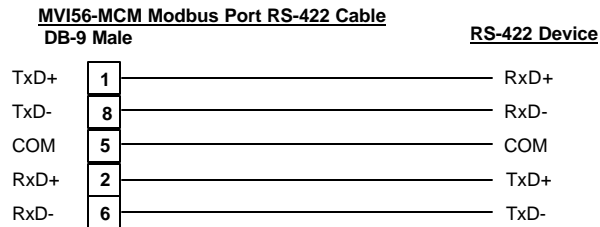
6.1.1.2 RS_485

Quando a interface RS_485 for usada, um, dois, ou três cabos serão requeridos. O uso de terra é opcional e depende da rede RS_485. O cabo requerido para esta interface está mostrado no diagrama abaixo:



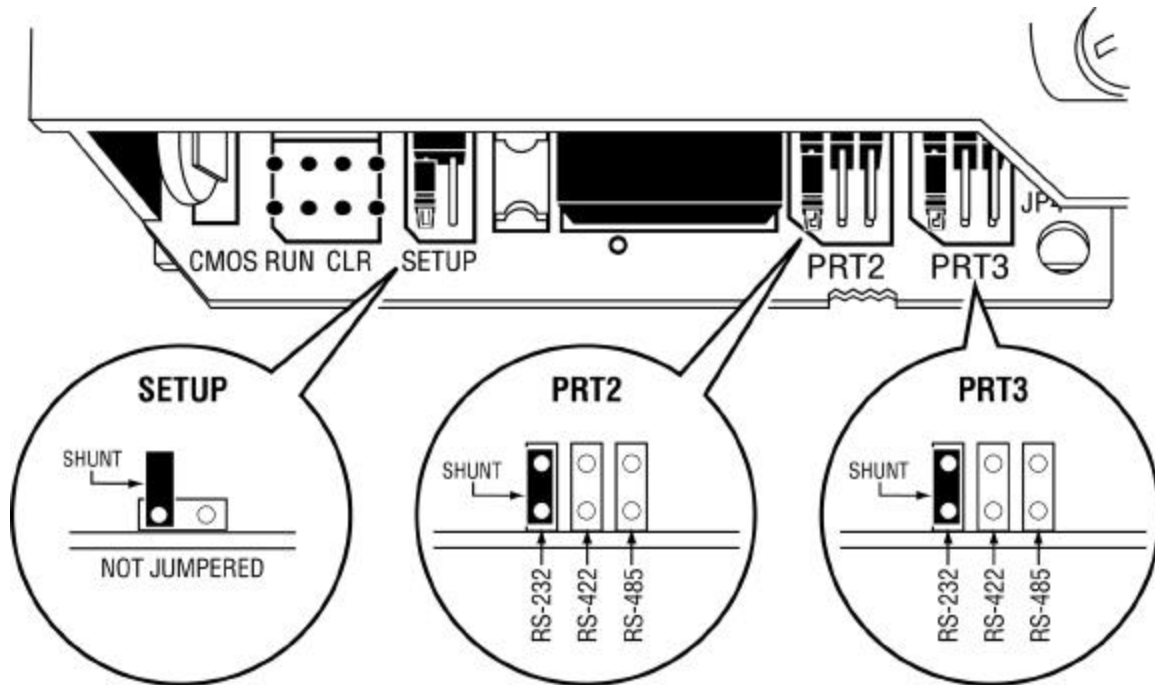
6.1.1.3 RS_422

Quando a interface RS_422 for usada, quatro ou cinco cabos serão requeridos. O uso de terra é opcional e depende da rede RS_422. O cabo requerido para esta interface está mostrado no diagrama abaixo:



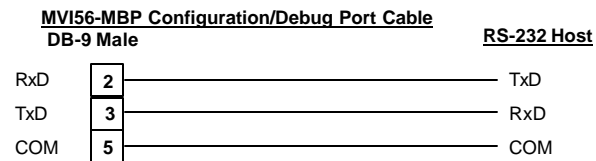
6.1.2 Ajustando Jumpers

Se você usar uma interface diferente de RS_232 (default), a configuração de jumper deverá ser outra para se ajustar à interface. O diagrama mostra a configuração de jumper MVI56-MCM.



6.2 Porta Configuração/Debug

Esta porta é fisicamente uma conexão RJ_45. Um cabo Pig_Tail RJ45 para DB9 é fornecido junto com o módulo. Esta porta permite a um programa de emulação de terminal PC visualizar os dados de status de configuração no módulo e controlar o módulo. O cabo para comunicação nesta porta está mostrado abaixo:



Apêndice A -

Definição de Banco de Dados MVI56-MCM

Este apêndice contém uma listagem do banco de dados interno do módulo MVI56-MCM. Esta informação pode ser usada para interfacear outros dispositivos para os dados contidos no módulo.

Gama de Registro	Modbus Reduzido	Modbus Elevado	Conteúdo	Tamanho
0-4999	40001	45000	Dado de usuário	5000
5000-5009	45001	45010	Configuração de Barramento	10
5010-5039	45011	45040	Setup de Porta 1	30
5040-5069	45041	45070	Setup de Porta 2	30
5070-5869	45071	46070	Comandos de Porta 1	800
5870-6669	46071	47070	Comandos de Porta 2	800
6750-6752	46751	46753	Setup de Porta 1 Aux.	3
6760-6762	46761	46763	Setup de Porta 2 Aux.	3
7600-7632	47601	47633	Dados de Status de Miscelânea	33
7200-7232	47801	7999	Controle de Comando	200

A área de dado de usuário é usada para manter os dados colhidos de outros nós na rede (comandos de leitura mestre) ou dados recebidos do processador (blocos de escrita).

Adicionalmente, esta área de dados é usada como fonte de dados para o processador (blocos de leitura) ou outros nós na rede (comandos de escrita).

Uma definição detalhada da área de status de miscelânea pode ser encontrada no apêndice B.

A definição das áreas de dados de configuração poderá ser encontrada na seção de definição de dados deste documento e no Apêndice C.

O Apêndice D contém uma discussão da seção de controle de comando do banco de dados.

Apêndice B

Definição de Dados de status MVI56-MCM

Este apêndice contém a descrição dos membros presentes no objeto **MCMInStat**.
Estes dados são transferidos do módulo para o processador como parte de cada bloco de leitura.

Offset	Conteúdo	Descrição
202	Quantidade de Escaneamento de Programa	Este valor é incrementado cada vez que um ciclo de programa é completado.
203–204	Código de Produto	Estes dois registros contêm o código de produto de "MCM"
205–206	Versão de Produto	Estes dois registros contêm a versão de Produto para o software correntemente em execução.
207–208	Sistema Operacional	Estes dois registros contêm os valores de mês e ano para o sistema operacional do programa.
209–210	Número RUN	Estes dois registros contêm o valor do número RUN para o software correntemente em execução.
211	Pedido de Lista de Comando de Porta 1	Este campo contém os números de pedidos feitos a partir desta porta para dispositivos escravo na rede.
212	Resposta de Lista de Comando de Porta 1	Este campo contém o número de mensagens de resposta de escravo recebidas na porta.
213	Erro de Lista de Comando de Porta 1	Este campo contém o número de erros de comando processados na porta. Estes erros podem ser provocados por má resposta ou mau comando.
214	Pedido de Porta 1	Este campo contém o número total de mensagens enviado da Porta.
215	Resposta de Porta 1	Este campo contém o número total de mensagens recebido na Porta.
216	Erros Enviados de Porta 1	Este campo contém o número total de erros de mensagem enviado da Porta.
217	Erros Recebidos de Porta 1	Este campo contém o número total de erros de mensagem recebido na Porta.
218	Pedido de Lista de Comando de Porta 2	Este campo contém o número de pedidos feitos a partir desta porta para dispositivo.s escravo na rede.
219	Resposta de Lista de Comando de Porta 2	Este campo contém o número de mensagens de resposta de escravo recebidas na Porta.
220	Erros de Lista de Comando de Porta 2	Este campo contém o número de erros de comando processados na porta. Estes erros podem ser provocados por má resposta ou mau comando
221	Pedido de Porta 2	Este campo contém o número total de mensagens enviado da Porta.
222	Resposta de Porta 2	Este campo contém o número total de mensagens recebido na Porta.
223	Erros Enviados de Porta 2	Este campo contém o número total de erros de mensagem enviado da Porta.
224	Erros Recebidos de Porta 2	Este campo contém número total de erros de mensagem recebido na Porta.
225	Quantidade de Blocos de Leit.	Este campo contém o número total de blocos de leitura transferido do módulo para o processador,

226	Quantidade de Blocos de Escrita	Este campo contém o número total de blocos de escrita transferido do módulo para o processador.
227	Quantidade de Blocos Subdivididos	Este campo contém o número total de blocos subdivididos corretamente recebido do processador.
228	Quantidade de Blocos de Evento de Comando	Este campo contém o número total de blocos de Evento de Comando recebido do processador.
229	Quantidade de Blocos de Comando	Este campo contém o número total de blocos de comando recebido do processador.
230	Quantidade de Blocos de Erro	Este campo contém o número total de erros de bloco reconhecido pelo módulo
231	Erro Corrente da Porta 1	Para uma Porta Escravo, este campo contém o valor do código de erro corrente que retornou. Para uma Porta Mestre, este campo contém o índice do comando correntemente em execução
232	Último Erro da Porta 1	Para uma Porta Escravo, este campo contém o valor do último código de erro que retornou. Para uma Porta Mestre, este campo contém o índice do comando com erro.
233	Erro Corrente da Porta 2	Para uma Porta escravo, este campo contém o valor do código de erro corrente que retornou. Para uma Porta Mestre, este campo contém o índice do comando correntemente em execução.
234	Último Erro da Porta 2	Para uma Porta Escravo, este campo contém o último código de erro que retornou. Para uma Porta Mestre, este campo contém o índice do comando com erro.

Apêndice C

Definição de Dados de Configuração MVI56-MCM

Este apêndice contém a listagem do banco de dados do módulo MVI56-MCM relativa à configuração do módulo. Este dado é disponível a qualquer nó na rede e pode ser lido a partir do processador ControlLogix quando o módulo for inicializado pela primeira vez.

Setup do Barramento

Registro	Conteúdo	Descrição
5000	Registro de Início de Escrita	O parâmetro especifica o registro de início, onde o dado transferido do processador será colocado. A gama válida para este parâmetro vai de 0 a 4999
5001	Quantidade de Registros de Escrita	O parâmetro especifica o número de registros a serem transferidos do processador para o módulo. A entrada válida para este parâmetro vai de 0 a 5000.
5002	Registro de Início de Leitura	O parâmetro especifica o registro de início no módulo onde o dado será transferido do módulo para o processador. Uma gama válida para este parâmetro vai de 0 a 4999.
50003	Quantidade de Registros de Leitura	O parâmetro especifica o número de registros a serem transferidos do módulo para o processador. Uma gama válida para este parâmetro vai de 0 a 5000.
5004	Falha de Barramento	O parâmetro especifica o número de sucessivos erros de transferência que devem ocorrer antes de as portas de configuração serem fechadas. Se for colocado o valor de zero, as portas de configuração continuarão operando sob quaisquer condições. Se o valor colocado for maior que zero (1-65535), as configurações cessam se ocorrer o número especificado de falhas.
5005	Ponteiro de Status de Erro	O parâmetro especifica a localização do registro no banco de dados do módulo onde o dado de status do módulo será armazenado. Se for colocado um valor menor que zero, o dado não será armazenado no banco de dados. Se o valor especificado estiver entre 0 e 4940, o dado será colocado na área de dados do usuário.
5006	Reserva	
5007	Reserva	
5008	Reserva	
5009	Reserva	

Setup da Porta 1

Registro	Conteúdo	Descrição
5010	Habilitado	O parâmetro é usado para definir se esta porta Modbus será usada. Se o parâmetro for colocado em zero, a porta será desabilitada, enquanto o valor um habilita a Porta.
5011	Tipo	O parâmetro especifica a porta que emula o dispositivo mestre Modbus (0), um dispositivo escravo Modbus sem Pass_Through (1), ou um dispositivo escravo com Pass_Through não formatado (2), ou um dispositivo escravo Modbus com Pass_Through formatado e inversão de dados (3).
5012	Flag de Flutuante (Float Flag)	Este Flag especifica se a funcionalidade de acesso de dado de Ponto Flutuante deve ser implementada. Se o Float_Flag for colocado em 1, as funções Modbus 3, 6, 16 interpretam valores de Ponto Flutuante a registros como os especificados pelos dois parâmetros que se seguem.
5013	Início de Flutuante	O parâmetro define o primeiro registro do dado de Ponto Flutuante. Todos os pedidos com valores de registro maiores ou iguais a este valor serão considerados pedidos de dado de Ponto Flutuante. Este parâmetro somente será usado se o Float Flag estiver habilitado.
5014	Offset de Flutuante	O parâmetro define o registro de início para o dado de Ponto Flutuante no banco de dados interno. Este parâmetro somente será usado se Flag de Flutuante estiver habilitado.
5015	Protocolo	O parâmetro especifica o protocolo Modbus a ser usado na porta. Os protocolos válidos são: 0= Modbus RTU e 1= Modbus ASCII.
5016	Baud_Rate	Baud_Rate a ser usado na porta. Entre Baud_Rate como valor. Por exemplo, para selecionar 19K Baud entre com 19200. Entradas válidas são 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 28800, 38400, 57600, e 115.
5017	Paridade	Este é o código de paridade a ser usado para a porta. Os valores são: Nenhum, Ímpar, Par
5018	Data Blts	O parâmetro estabelece o número de data bits a cada palavra usada pelo protocolo. Entradas válidas para este campo se encontram entre 5 e 8.
5019	Stop Bits	O valor estabelece o número de stop bits a ser usado com cada valor de dado enviado. Entradas válidas para este campo se encontram entre 1 e 2.
5020	RTS On	O parâmetro estabelece o número de milissegundos de atraso depois de o RTS ter sido declarado e antes de o dado ser transmitido. Os valores válidos se encontram entre 0 e 65535 segundos.
5021	RTS Off	P parâmetro estabelece o número de milissegundos de atraso depois de o último byte de dados ter sido enviado e antes de o sinal de modem RTS ser reduzido. Os valores válidos se encontram entre 0 e 65535.

Registro	Conteúdo	Descrição
5022	Tempo de Resposta MIn	O parâmetro especifica o número mínimo de milisegundos de atraso antes de responder a uma mensagem de pedido. Este atraso no pré-envio é aplicado antes de o RTS estar ativo. Isto pode ser requerido na comunicação com dispositivos lentos.
5023	Linha CTS de Uso	O parâmetro especifica se o a linha de controle do modem CTS deve ser usada. Se o parâmetro for colocado em zero, a linha CTS não será monitorada. Se o parâmetro for colocado em um , a linha CTS será monitorada e deve ser elevada antes de o módulo enviar dados. Este parâmetro normalmente será requerido apenas quando se usa modems half-duplex. (2-fios)
5024	ID Escravo	O parâmetro define o endereço de escravo Modbus virtual para o banco de dados interno. Todos os pedidos recebidos pela porta com este endereço são processados pelo módulo. Se assegure que cada dispositivo tenha um único endereço em uma rede. A gama válida para este parâmetro se encontra entre 1 e 255 (247 em algumas redes).
5025	Bit in Offset	O parâmetro especifica o endereço offset no banco de dados Modbus interno que deve ser usado com pedido de rede para comando de função Modbus 2 . Por exemplo, se o valor for colocado em 150, um pedido de endereço de zero retorna o valor para o registro 150 no banco de dados.
5026	Palavra in Offset	O parâmetro especifica o endereço de offset no banco de dados Modbus interno que deve ser usado com pedidos para comandos de função Modbus 4 . Por exemplo, se o valor for colocado em 150, um pedido de endereço de zero retorna o valor para o registro 150 no banco de dados.
5027	Out in Offset	O parâmetro especifica o endereço de offset no banco de dados Modbus interno que deve ser usado com pedidos para comandos de função Modbus 1, 5, 5 . Por exemplo, se o valor for colocado em 100, um pedido de endereço de zero retorna o valor para o registro 50 no banco de dados.
5028	Offset de Registro de Armazenamento	O parâmetro especifica um endereço de offset no banco de dados Modbus interno que deve ser usado com os pedidos de rede para comandos de função Modbus 3, 6, 16 . Por exemplo, se o valor for colocado em 50, um pedido de endereço zero corresponderá ao registro 50 no banco de dados.
5029	Quantidade de Cmd	O parâmetro especifica o número de comandos a serem processados pela porta mestre Modbus.
5030	Atraso de Cmd Min	O parâmetro especifica quantos milisegundos deve se aguardar entre emissões de comandos . Este valor de atraso não se aplica a repetições (retries).
5031	Ponteiro de Cmd de Erro	O parâmetro coloca o endereço no banco de dados interno onde o erro de comando será colocado. Se o valor estabelecido for 1, o dado não será transferido para o banco de dados. A gama válida de valores para este parâmetro se encontra entre 1 e 4999.

Registro	Conteúdo	Descrição
5032	Timeout de Resposta	O parâmetro representa o período de timeout de resposta de mensagem em incrementos de 1 milissegundo. Este é o tempo em que a porta configurada como mestre deve aguardar antes de retransmitir um comando se nenhuma resposta for recebida do escravo endereçado. O valor é colocado dependendo da rede de comunicação usada e o do tempo de resposta esperado do dispositivo mais lento na rede.
5033	Quantidade de repetições	O parâmetro especifica o número de vezes que um comando deve ser repetido se ele falha . Se a porta mestre não recebe uma resposta depois da última repetição, a comunicação dos dispositivos escravos serão suspensa na porta pelo escaneamento do Error Delay Counter .
5034	Quantidade de Atraso de Erro	O parâmetro especifica o número de varreduras a avançar antes de tentar restabelecer comunicação. Depois de o escravo falhar em responder, o mestre avança os comandos a serem enviados para o escravo no número de vezes estabelecido neste parâmetro.
5035	Reserva	
5036	Reserva	
5037	Reserva	
5038	Reserva	
5039	Reserva	

Setup da Porta 2

Registro	Conteúdo	Descrição
5040	Habilitado	O parâmetro é usado para definir se esta porta Modbus será usada. Se o parâmetro for colocado em zero , a porta será desabilitada, enquanto o valor um habilita a porta.
5041	Tipo	O parâmetro especifica se a porta emula um dispositivo mestre Modbus (0), um dispositivo escravo Modbus sem Pass_Through (1), ou um dispositivo escravo Modbus com Pass_Through não-formatado (2), ou um dispositivo escravo Modbus com Pass_Through formatado e inversão de dados (3).
5042	Flag de Flutuante (Float Flag)	Este Flag especifica se a funcionalidade de acesso a dado de Ponto Flutuante deve ser implementada. Se o Flag de Flutuante for colocado em um , as funções Modbus 3, 6, 16 interpretam os valores de ponto flutuante nos registros, como especificado pelos dois parâmetros que se seguem
5043	Início de Flutuante	Este parâmetro define o primeiro registro de dados de Ponto Flutuante. Todos os pedidos de registro maiores ou iguais a este valor serão considerados pedidos de dados ponto flutuante. O parâmetro somente será usado se o Flag de Flutuante estiver habilitado.

Registro	Conteúdo	Descrição
5044	Offset de Flutuante	Este parâmetro define o registro de início para o dado de ponto flutuante no banco de dados interno . Este parâmetro somente será usado se o Flag de Flutuante estiver habilitado
5045	Protocolo	O parâmetro especifica o protocolo Modbus a ser usado na porta. Os protocolos válidos são: 0= Modbus RTU e 1= Modbus ASCII
5046	Baud_Rate	Baud_Rate a ser usado na porta. Entre Baud_Rate como valor. Por exemplo, para selecionar 19k, entre com 19200. Entradas válidas são 110, 300, 600, 1200, 2400, 4800, 9600, 19200, 28800, 38400, 57600 e 115
5047	Paridade	O parâmetro é um código de paridade a ser usado para a porta. Os valores válidos são Nenhum, Ímpar, Par
5048	Data Blts	O parâmetro estabelece o número de data bits para cada palavra usada pelo protocolo. Entradas válidas para este campo se encontram entre 5 e 8.
5049	Stop Bits	O parâmetro estabelece o número de stop bits a ser usado com cada valor de dado enviado. As entradas válidas são 1 e 2.
5050	RTS On	O parâmetro estabelece o número de milissegundos de atraso depois de RTS ter sido declarado antes de o dado ser transmitido. Os valores válidos se encontram entre 0 e 65535 milissegundos.
5051	RTS Off	O parâmetro estabelece o número de milissegundos de atraso antes de o último byte de dado ter sido enviado antes de o sinal de modem RTSD ser reduzido. Valores válidos se encontram entre 0 e 6535 milissegundos
5052	Tempo de Resposta Min	O parâmetro especifica o número mínimo de milissegundos de atraso antes de responder a uma mensagem de pedido. Este atraso pré-enviado é aplicado antes de RTS. Isto pode ser requerido na comunicação com dispositivos lentos.
5053	Linha CTS de Uso	Este parâmetro especifica se a linha de controle CTS deve ser usada. Se o parâmetro for colocado em zero , a linha CTS não deve monitorada, e se o parâmetro for colocado em um , a linha CTS será monitorada e deverá ser elevada antes de o módulo enviar dados. Este parâmetro normalmente será requerido quando forem usados modems half-duplex (2 fios).
5054	ID de Escravo	Este parâmetro define o endereço de escravo Modbus virtual para o banco de dados interno. Todos os pedidos recebidos na porta com este endereço são processados pelo módulo. Assegure-se que cada dispositivo tenha um único endereço em uma rede. A gama válida para este parâmetro vai de -1 a 255 (247 em certas redes).

Registro	Conteúdo	Descrição
5055	Bit em offset	Este parâmetro especifica o endereço offset no banco de dados Modbus interno que deve ser usado com pedidos de rede para comandos de função Modbus 2. Por exemplo, se o valor for colocado em 150, um pedido de endereço de zero deve retornar o valor para o registro 150 no banco de dados.
5056	Palavra em offset	Este parâmetro especifica o endereço de offset no banco de dados Modbus interno que deve ser usado com pedido de rede para comandos de função Modbus 4. Por exemplo, se o valor for colocado em 150, um pedido de endereço de zero deve retornar o valor para o registro 150 no banco de dados.
5057	Out in Offset	O parâmetro especifica o endereço offset no banco de dados Modbus interno que deve ser usado com pedidos de rede para comandos de função Modbus 1, 5, 15. Por exemplo, se o valor for colocado em 100, um pedido de endereço de zero deve corresponder ao registro 50 no banco de dados.
5058	Offset do Registro de Armazenamento	O parâmetro especifica o endereço de offset no banco de dados Modbus interno que deve ser usado com pedidos de rede para comandos de função Modbus 3, 6, 46. Por exemplo, se o valor for colocado em 50, um pedido para endereço zero corresponde ao registro 50 no banco de dados.
5059	Quantidade de Cmd	O parâmetro especifica o número de comandos a serem processados pela porta mestre Modbus.
5060	Atraso de Cmd MIn	Este parâmetro especifica o número de milissegundos a aguardar entre emissões de cada comando. Este valor de atraso não se aplica a repetições (retries).
5061	Ponteiro de Erro de Cmd	Este parâmetro estabelece o endereço no banco de dados Modbus interno, onde o erro de comando será colocado. Se o valor for colocado em -1, o dado não será transferido para o banco de dados. A gama válida de valores para este parâmetro vai de -1 e 4999
5062	Timeout de Resposta	Este parâmetro representa um período de timeout de resposta de mensagem em incrementos de 1 milissegundo. Este é o tempo no qual a porta configurada como mestre aguarda antes de retransmitir um comando, se nenhuma resposta for recebida do escravo endereçado. O valor é estabelecido dependendo da rede de comunicação usada e do tempo de resposta esperado do dispositivo mais lento na rede.
5063	Quantidade de Repetições	Este parâmetro especifica o número de vezes que um comando é repetido em caso de falha. Se a porta Mestre não recebe uma resposta depois da última repetição, a comunicação de dispositivos escravos será suspensa na porta pelo escaneamento da "Quantidade de Atraso de Erro"
5064	Quantidade de Atraso de Erro	Este parâmetro especifica o número de varreduras a avançar no escravo tentando restabelecer comunicação. Depois de o escravo falhar em responder, o mestre deve saltar os comandos a serem enviados para o escravo o número de vezes estabelecido para este parâmetro.

Registro	Conteúdo	Descrição
5065	Reserva	
5066	Reserva	
5067	Reserva	
5068	Reserva	
5069	Reserva	

Comandos da Porta 1

Registro	Conteúdo	Descrição
5070-5777	Comando # 1	Este jogo de registros contém os parâmetros para o primeiro comando na lista de comando mestre. A estrutura desta área de dados pode ser vista na seção de objeto de dados da documentação.
5878-5085	Comando # 2	Jogo de Dados de Comando # 2.
5852-5859	Comando #100	Jogo de Dados de Comando # 100.

Comandos da Porta 2

Registro	Conteúdo	Descrição
5870-5877	Comando # 1	Este jogo de registros contém os parâmetros para o primeiro comando na lista de comando mestre. A estrutura desta área de dados pode ser vista na seção de objeto de dados da documentação.
5878-5885	Comando # 2	Jogo de Dados de Comando # 2.
6662-6669	Comando # 100	Jogo de Dados de Comando # 100.

Status de Misc

Registro	Conteúdo	Descrição
6670	Quantidade de Escaneamento de Pgm	Este valor é incrementado cada vez que é completado um ciclo de programa no módulo.
6671-6672	Código de Produto	Estes dois registros contêm o código de produto MCM
6673-6674	Versão de Produto	Estes dois registros contêm a versão de produto para o software correntemente em execução.
6675-6676	Sistema Operacional	Estes dois registros contêm valores de mês e ano para o sistema operacional do programa.
6677-6678	Número RUN	Estes dois registros contêm o valor do número RUN para o software em execução corrente.

Registro	Conteúdo	Descrição
6679	Pedidos da Lista de Comando da Porta 1	Este campo contém o número de pedidos feitos a partir desta porta para os dispositivos escravos na rede.
6680	Resposta da Lista de Comando da Porta 1	Este campo contém o número de mensagens de resposta de escravo recebidas na porta.
6681	Erros da Lista de Comando da Porta 1	Este campo contém o número de erros de comando processados na porta. Estes erros podem ser provocados por má resposta ou mau comando.
6682	Pedidos da Porta 1	Este campo contém o número total de mensagens enviado da porta .
6683	Respostas da porta 1	Este campo contém o número total de mensagens recebido na porta.
6684	Erros Enviados da Porta 1	Este campo contém o número total de erros de mensagem enviado da porta.
6885	Erros recebidos na Porta 1	Este campo contém o número total de erros de mensagem recebido na porta.
6686	Pedidos da Lista de Comando da Porta 2	Este campo contém o número de pedidos feitos desta porta para os dispositivos escravo na rede.
6687	Respostas da Lista de Comando da Porta 2	Este campo contém o número de mensagens de resposta de escravo recebidas na porta.
6688	Erros da Lista de Comando da Porta 2.	Este campo contém o número de erros de comando processados na porta. Estes erros podem ser provocados por má resposta ou mau comando.
6689	Pedidos da Porta 2	Este campo contém o número total de mensagens enviado da porta.
6690	Respostas da Porta 2	Este campo contém o número total de mensagens recebido na porta.
6691	Erros enviados da Porta 2	Este campo contém o número total de erros de mensagem enviado da porta.
6692	Erros recebidos na Porta 2	Este campo contém o número total de erros de mensagem recebido na porta.
6693	Quantidade de Blocos de Lei.	Este campo contém o número total de blocos de leitura transferido do módulo para o processador.
6694	Quantidade de Blocos de Escrita	Este campo contém o número total de blocos de escrita transferido do módulo para o processador.
6695	Quantidade de Blocos Subdivididos	Este bloco contém o número total de blocos subdivididos corretamente recebido do processador.
6696	Quantidade de Blocos de Evento de Cmd	Este campo contém o número total de blocos de evento de comando recebido do processador.
6697	Quantidade de Blocos de Cmd	Este campo contém o número total de blocos de comando recebido do processador.
6698	Quantidade de Blocos de Erro	Este campo contém o número total de erros de bloco reconhecido pelo módulo.
6699	Erro Corrente da Porta 1	Para uma porta escravo, este campo contém o valor do código de erro corrente que retornou. Para uma porta mestre, este campo contém o índice do comando em execução corrente.

6700	Último Erro da Porta 1	Para uma porta escravo, este campo contém o valor do último erro que retornou. Para uma porta mestre, este campo contém o índice do comando com erro.
6701	Erro Corrente da Porta 2	Para uma porta escravo, este campo contém o valor do código de erro corrente que retornou. Para uma porta mestre, este campo contém o índice do comando em execução corrente
6702	Último Erro da Porta 2	Para uma porta escravo, este campo contém o valor do código de erro corrente que retornou. Para uma porta mestre, este campo contém o índice do comando com erro.
6703	Reserva	
6749	Reserva	
6750	Banda de Proteção de Uso da Porta 1	Use timeout de intervalo de pacote para mensagens (Sim ou Não). Use somente em aplicações multi-drop
6751	Tempo de Banda de Proteção da Porta 1	Um valor de zero usa a taxa de Baud_Rate default ou você pode estabelecer um valor de timeout em milisegundos
6752	Offset Fcn 99 da Porta 1	Offset de BD interno para Contador de Função 99
6753	Reserva	
6759	Reserva	
6760	Banda de Proteção de Uso da Porta 2	Use timeout de intervalo de pacote para mensagens (Sim ou Não). Use somente em aplicações multi-drop
6761	Port 2 Tempo de Banda de Proteção da Porta 2	Um valor de zero usa a taxa de Baud default ou você pode estabelecer um valor de timeout em milisegundos
6762	Offset Fcn 99 da Porta 1	Offset de BD interno para Contador da Função 99
6763	Reserva	
6799	Reserva	

Controle de Comando

Registro	Conteúdo	Descrição
6800	Código de Cmd	Entre com um dos códigos de comando de controle válidos neste registro para controlar o módulo (9997, 9998, ou 9999). Dirija-se para o Apêndice D para maiores informações.
6801	Dado de Cmd	Não usado
6999	Dado de Cmd	Não Usado

Apêndice D- Controle de Comando MVI56-MCM

Dados de Controle de Comando recebidos de outros nós na rede podem controlar o módulo MVI56-MCM. Valores específicos são escritos nas regiões deste bloco para controlar o módulo. Correntemente, o módulo é programado para manusear a recepção dos seguintes pedidos: configuração escrita para processador, boot a quente, e boot a frio.

O nó remoto controla o módulo escrevendo um dos seguintes valores no registro 7800 (endereço Modbus 47801).

9997	Configuração escrita no banco de dados para o processador e Proceder a Boot A Quente no módulo
9998	Proceder a Boot A Quente no módulo
9999	Proceder a Boot A Frio no módulo

O registro de controle é limpo (valor zero) depois de a operação ter sido realizada, com exceção do comando 9997. Se o módulo falhar na transferência da configuração para o processador, um código de erro retorna para o registro de controle da seguinte forma:

0	Nenhum erro, transferência bem sucedida
-1	Erro na transferência de Informação de configuração geral
-2	Erro na Transferência de Lista de Comando Mestre de Porta 1 Modbus
-3	Erro na Transferência de Lista de Comando Mestre de Porta 2 Modbus

A Lógica Ladder deve ser escrita para manusear o comando 9997.

A Lógica Ladder não é requerida quando se utilizam comandos Boot A Quente e Boot A Frio.

Apêndice E – Especificações do Produto

Especificações Gerais

O módulo MVI56-MCM atua como gateway entre uma rede Modbus e barramento Allen_Bradley. A transferência de dados do processador ControlLogix é assíncrona a partir das ações na rede Modbus. Um espaço de registro de 5000 palavras no módulo é usado para trocar dados entre processador e rede Modbus.

Algumas das especificações gerais incluem:

- ?? Suporte para armazenamento e transferência de até 5000 registros de/ para arquivos de dados do processador ControlLogix.
- ?? Utilização de memória do módulo que é completamente definida por usuário.
- ?? Duas Portas para emular qualquer combinação de dispositivo mestre ou escravo Modbus.
- ?? Parâmetros configuráveis incluem:

Protocolo	RTU ou ASCII
Baud Rate	110 a 115200
Paridade	Nenhuma, Ímpar, Par
Data Bits	5 a 8
Stop Bits	1 ou 2
RTS ON e OFF Timing	0 a 65535 milisegundos
Atraso de Resposta Min	0 a 65535 milisegundos
Linha CTS de Uso	Sim ou Não
Suporte de Ponto Flutuante de trajetórias de Roteamento de Dispositivo	1 a 64

Especificações Funcionais de Escravo

O módulo MVI56-MCM aceita comandos de código de função Modbus 1, 2, 3, 4, 5, 6, 8, 15, 16, 17, 22 e 23 de uma unidade mestre Modbus anexa. Uma porta configurada como escravo Modbus permite a um mestre remoto interagir com todos dados contidos no módulo. Os dados podem ser procedentes de outros dispositivos escravo Modbus na rede através de uma porta mestre ou a partir de processador ControlLogix. O módulo pode ser configurado de modo a passar comandos escritos (funções 5, 6, 15, 16, 22 e 23) diretamente do host remoto para o processador. Este modo de operação é chamado modo Pass_Through.

Especificações Funcionais de mestre Modbus

A porta configurada como dispositivo mestre Modbus virtual no módulo MVI56-MCM ativamente emitirá comandos Modbus a outros nós na rede Modbus. Cem comandos são suportados em cada porta. Adicionalmente, as portas mestre têm característica de varredura otimizada que deve varrer escravos com problemas de configuração menos freqüentes. O processador ControlLogix pode ser programado de modo a controlar atividade na porta selecionando ativamente comandos da Lista de Comando para executar ou emitir comandos diretamente da Lógica Ladder. O processador ControlLogix também tem a capacidade de controlar o escaneamento de escravos na porta. A varredura (Polling) de escravos individuais pode ser seletivamente controlada (Habilitado/ Desabilitado) por Lógica Ladder .

Físico

Este módulo é projetada pela ProSoft_Technology e incorpora tecnologia licenciada da Allen_Bradley (Tecnologia de Barramento ControlLogix)

- ?? Fator de Forma ControlLogix – Slot Único

- ?? Conexões
- 2- conectores RJ45 para suporte de interfaces RS_232, RS_422, RS_485
- 1- RJ_45, RS_232 Conector de Ferramenta de Configuração

Interface ControlLogix

- ?? Operação via Lógica Ladder simples
- ?? Complete setup e monitoramento do módulo com software RSLogix 5000
- ?? Interface de barramento ControlLogix via Acesso E/S
- ?? Todos dados relativos ao módulo estão contidos em um único Tag Controlador com objetos definidos para facilitar configuração, monitoramento e interfaceamento com módulo.
- ?? Dados de configuração de módulo e configuração de comunicação são transferidos para MVI56-MCM via um pré-definido tipo de dado de usuário no processador.

Especificações de hardware

O módulo MVI56-MCM é projetado pela ProSoft_Technology e incorpora tecnologia licenciada da Allen_Bradley (Tecnologia de Barramento ControlLogix)

- ?? Cargas Correntes 800 ma @ 5V (a partir do barramento)
- ?? Temperatura de Operação 0 a 60°C
- ?? Temperatura de Armazenamento -40 a 85°C
- ?? Umidade Relativa 5-95% (sem condensação)
- ?? Conector de Porta Modbus dois Conectores RJ45 (cabo RJ45 a DB9 fornecido com a unidade) suportando interfaces RS_232, RS_422, RS_485 (cabos RJ45 a DB9 fornecidos com a unidade).
- ?? Conector de Configuração: Conector RJ45 (fornecido cabo RJ45 a DB9 com a unidade).

Suporte, Manutenção, e Garantia

Suporte técnico

ProSoft_Technology sobrevive por sua capacidade de prover suporte integral a seus clientes. Qualquer questão ou problema que surgir, por favor nos chame:

Fábrica/ Suporte Técnico
ProSoft_Technology
9801 Camino Media, Suite 105
Bakersfield CA 93311
(661)664-7208
(800)326-7066
(661)664-7233 (fax)
prosoft@prosoft-technology.com
www.prosoft-technology.com

No entanto, antes de ligar para o suporte, se prepare. Para prover o suporte melhor e mais rápido, pedimos as seguintes informações (que poderão ser enviados por fax antes da consulta por telefone propriamente dita).

- 1 Número da Versão do Produto
- 2 Hierarquia do Sistema
- 3 Configuração do Módulo e conteúdo do arquivo MCM.CFG
- 4 Operação do Módulo
Informação de status de Configuração/Debug
Padrões de LED
- 5 Informação com respeito a processador e áreas de dados visto através do RSLogix 500 e padrões de LED no processador
- 6 Detalhes com respeito à rede serial

Um sistema automático depois do horário (número de Bakersfield) dá acesso por Pager a um de nossos técnicos e/ou engenheiros de suporte a qualquer tempo para responder questões urgentes.

Manutenção e Reparo de Módulo

A placa MVI56-MCM é um produto eletrônico, projetado e fabricado para funcionar sob condições adversas. Mas como qualquer produto, com envelhecimento, má aplicação, ou outro motivo qualquer, pode precisar de reparo em algum momento.

Quando comprado da ProSoft_Technology, o módulo oferece garantia de um ano para peças e mão de obra de acordo com os limites especificados na garantia. Substituição e/ou retorno devem ser dirigidos para o distribuidor de quem o produto foi comprado. Se a placa precisar retornar para reparo, será preciso antes obter o número RMA da ProSoft_Technology. Favor contatar a fábrica para obtê-lo e uma vez disponível, coloque-o de modo bem visível na embalagem.

Política de Garantia Geral

A ProSoft_Technology Inc (doravante simplesmente ProSoft) garante que o Produto se encontra em conformidade e tem o desempenho de acordo com especificações técnicas incluídas no material anexo, e não deve apresentar

defeitos de material e/ou mão de obra, no período indicado, sendo que este período de garantia começa a contar na data de recebimento do produto.

Esta garantia é limitada a reparos e substituição, a critério da ProSoft, de produtos com defeitos ou não-conformidades, e a ProSoft não será responsável por falhas do produto em realizar funções especificadas ou outras não-conformidades causadas ou atribuída a: (a) qualquer mau uso do Produto; (b) falha do Cliente em cumprir qualquer especificação ou instrução da ProSoft; (c) negligência, abuso, ou acidente infligido ao Produto; ou (d) qualquer equipamento ou software associado ou complementar não fornecido pela ProSoft.

Um serviço de garantia limitado pode ser obtido entregando o Produto à ProSoft com a respectiva prova de compra ou recibo. O Cliente concorda em garantir o Produto ou assumir riscos de perda ou dano em trânsito e pagar adiantado as taxas de embarque e transporte à ProSoft e usar embalagem original ou equivalente. Contate Serviço ao Cliente da ProSoft para mais informações.

Limite de Responsabilidade

EXCETO COMO EXPRESSAMENTE ESPECIFICADO AQUI, A **PROSOFT** NÃO DÁ GARANTIA DE QUALQUER TIPO, EXPRESSO OU SUBTENDIDO, COM RESPEITO A QUALQUER EQUIPAMENTO, PARTE, OU SERVIÇO PROVIDO DE ACORDO COM ESTE TERMO, INCLUINDO MAS NÃO SE LIMITANDO À GARANTIA DE ADEQUAÇÃO E COMERCIAL PARA UM PARTICULAR PROPÓSITO, NEM A **PROSOFT** E SEU REPRESENTANTE SERÃO RESPONSÁVEIS POR QUALQUER DANO, INCLUINDO, MAS NÃO SE LIMITANDO A, DANOS DIRETOS, INDIRETOS, ACIDENTAIS, ESPECIAIS, OU PROVOCADOS, QUER POR CONTRATO OU POR CULPA INCLUINDO NEGLIGÊNCIA E RESPONSABILIDADE ESTRITA TAL COMO, MAS NÃO SE LIMITANDO A, PERDA, LUCROS CESSANTES OU BENEFÍCIOS RESULTANTES, PROPICIADAS POR, OU EM CONEXÃO COM, USO OU FORNECIMENTO DO EQUIPAMENTO, PARTE, OU SERVIÇO DESCRITO AQUI OU DESEMPENHO, USO, INABILIDADE NO USO DO MESMO, MESMO SE A RESPONSABILIDADE TOTAL DA **PROSOFT** OU SEU REPRESENTANTE EXCEDER O PREÇO PAGO PELO PRODUTO.

Onde estabelecido por Lei Estadual, algumas das exclusões ou limitações acima poderão não ser aplicáveis em alguns Estados. Esta garantia provê direitos legais específicos; outros direitos que variem de Estado para Estado ainda poderão existir. Esta garantia não será aplicada na extensão em que qualquer provisão desta garantia seja proibida por qualquer Lei Federal, Estadual, ou Municipal que não possa ser pressuposta.

Detalhes de Garantia para o Hardware

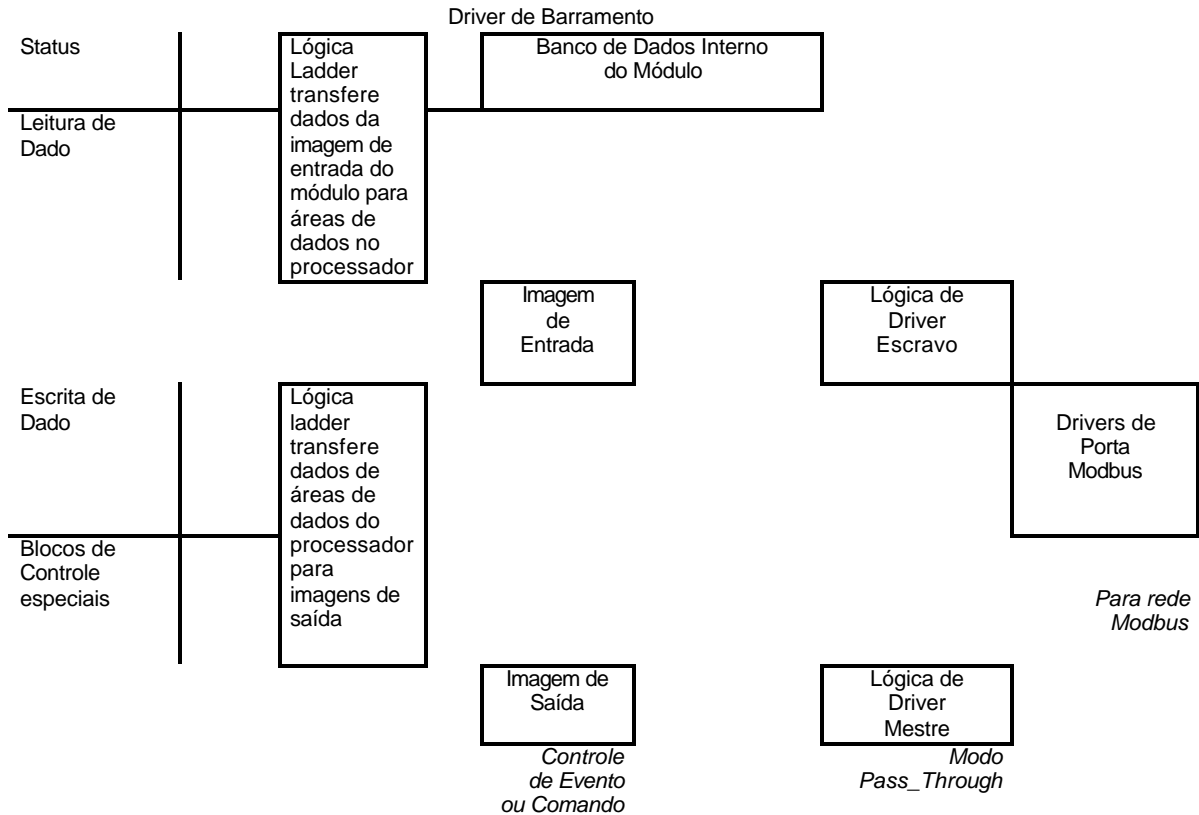
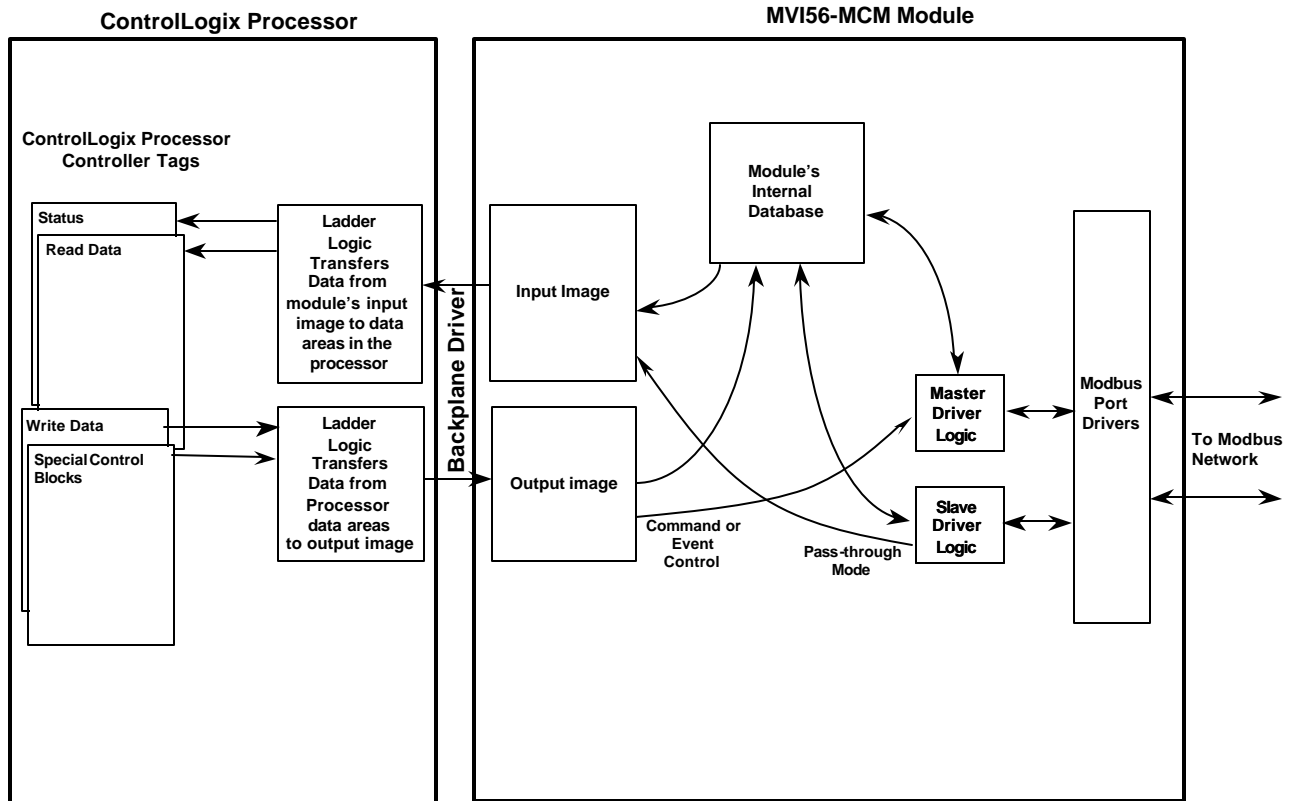
Período de Garantia:

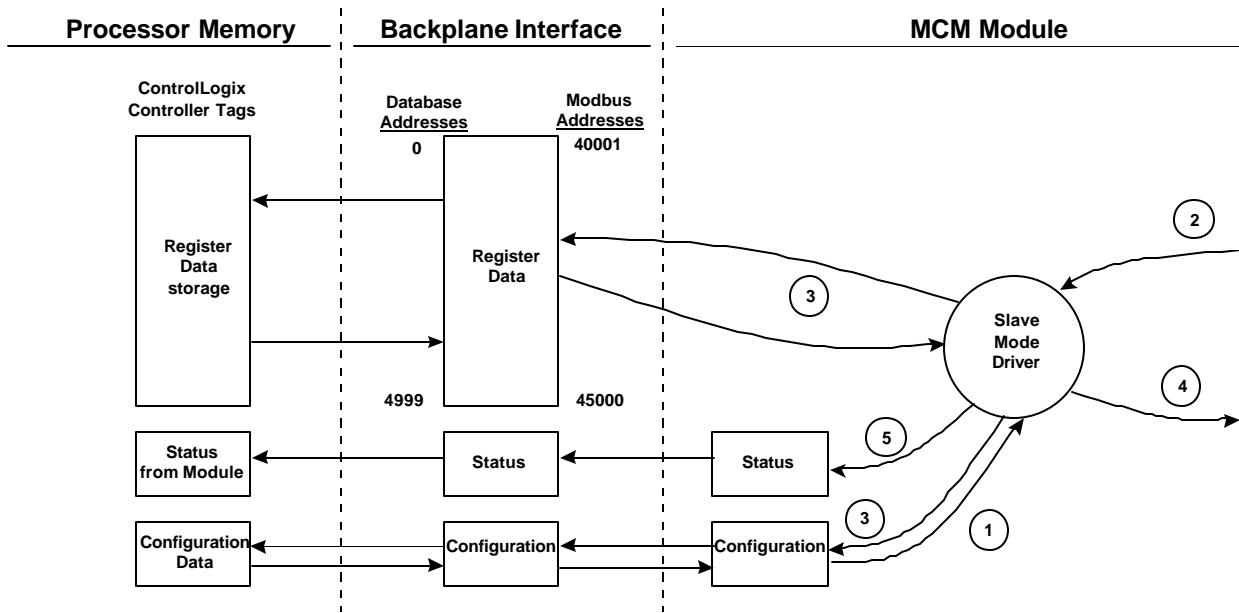
A ProSoft garante hardware pelo período de 1 (um) ano.

Procedimento de Garantia

No retorno do Produto de Hardware a ProSoft deve, a seu critério, reparar ou substituir o Produto sem nenhum custo adicional, exceto como estabelecido acima. O reparo de Partes ou substituição do Produto é provido com base de troca por Partes ou Produtos novos ou reconicionados. Produtos e Partes substituídos se tornam propriedades da ProSoft.

Se a ProSoft determinar que o Produto não tem garantia, a ProSoft, a critério do Cliente, pode reparar o Produto usando custos correntes para Partes e Mão de Obra, retornando o Produto para coleta do transportador





Memória de processador

Interface de Barramento

Módulo MCM

Endereços de banco de dados Endereços Modbus de dados

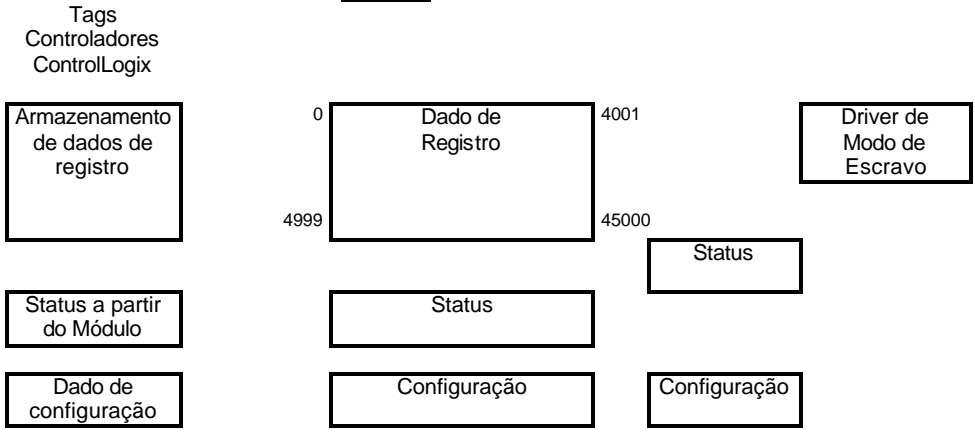


DIAGRAMA III 2.9.3- Página 19

