

MasterTool® Programming Manual de Programação

Rev. H 09/2004

Cód. Doc: 6399-100.4

Nenhuma parte deste documento pode ser copiada ou reproduzida de alguma forma sem o consentimento prévio e por escrito da ALTUS Sistemas de Informática S.A., que reserva-se o direito de efetuar alterações sem prévio comunicado.

Conforme legislação vigente no Brasil, do Código de Defesa do Consumidor, informamos os seguintes aspectos relacionados com a segurança de pessoas e instalações do cliente:

Os equipamentos de automação industrial, fabricados pela ALTUS, são robustos e confiáveis devido ao rígido controle de qualidade a que são submetidos. No entanto, equipamentos eletrônicos de controle industrial (controladores programáveis, comandos numéricos, etc.) podem causar danos às máquinas ou processos por eles controlados, no caso de defeito em suas partes e peças, erros de programação ou instalação, podendo inclusive colocar em risco vidas humanas.

O usuário deve analisar as possíveis consequências destes defeitos e providenciar instalações adicionais externas de segurança que, em caso de necessidade, atuem no sentido de preservar a segurança do sistema, principalmente nos casos da instalação inicial e de testes.

É imprescindível a leitura completa dos manuais e/ou características técnicas do produto, antes da instalação ou utilização do mesmo.

A ALTUS garante os seus equipamentos contra defeitos reais de fabricação pelo prazo de doze meses a partir da data da emissão da nota fiscal. Esta garantia é dada em termos de manutenção de fábrica, ou seja, o transporte de envio e retorno do equipamento até a fábrica da ALTUS, em Porto Alegre, RS, Brasil, ocorrerá por conta do cliente. A garantia será automaticamente suspensa caso sejam introduzidas modificações nos equipamentos por pessoal não autorizado pela ALTUS. A ALTUS exime-se de quaisquer ônus referentes a reparos ou substituições em virtude de falhas provocadas por agentes externos aos equipamentos, pelo uso indevido dos mesmos, bem como resultantes de caso fortuito ou por força maior.

A ALTUS garante que seus equipamentos funcionam de acordo com as descrições contidas explicitamente em seus manuais e/ou características técnicas, não garantindo a satisfação de algum tipo particular de aplicação dos equipamentos.

A ALTUS desconsiderará qualquer outra garantia, direta ou implícita, principalmente quando se tratar de fornecimento de terceiros.

Pedidos de informações adicionais sobre o fornecimento e/ou características dos equipamentos e serviços ALTUS, devem ser feitos por escrito. A ALTUS não se responsabiliza por informações fornecidas sobre seus equipamentos sem registro formal.

DIREITOS AUTORAIS

Série Ponto, MasterTool e QUARK são marcas registradas da ALTUS Sistemas de Informática S.A. IBM é marca registrada da International Business Machines Corporation.

Sumário

PREFÁCIO	1
DESCRIÇÃO DESTE MANUAL	1
MANUAIS RELACIONADOS	1
TERMINOLOGIA	2
CONVENÇÕES UTILIZADAS	3
CONVENÇÕES PARA UTILIZAÇÃO COM MOUSE	3
SUPORTE TÉCNICO	5
REVISÕES DESTE MANUAL	6
INTRODUÇÃO	8
A LINGUAGEM DE PROGRAMAÇÃO	8
LINGUAGEM DE DIAGRAMAS DE RELÉS	9
ELEMENTOS DE PROGRAMAÇÃO	9
ORGANIZAÇÃO DE MEMÓRIA DOS CPS ALTUS	9
LÓGICAS	11
OPERANDOS	12
IDENTIFICAÇÃO DE UM OPERANDO PELO ENDEREÇO	12
IDENTIFICAÇÃO DE UM OPERANDO PELO TAG	12
OPERANDOS UTILIZADOS NO MASTERTOOL	13
IDENTIFICAÇÃO DOS OPERANDOS SIMPLES	14
IDENTIFICAÇÃO DOS OPERANDOS CONSTANTE	15
IDENTIFICAÇÃO DOS OPERANDOS TABELA	15
OPERANDOS %E - RELÉS DE ENTRADA	16
OPERANDOS %S - RELÉS DE SAÍDA	16
OPERANDOS %A - RELÉS AUXILIARES	17
OPERANDOS %R - ENDEREÇOS NO BARRAMENTO	17
OPERANDOS %M - MEMÓRIAS	19
OPERANDOS %D - DECIMAIS	20
OPERANDOS %F - REAIS	21
OPERANDOS %I - INTEIRO	21
OPERANDOS %KM, %KI, %KD E %KF - CONSTANTES	22
OPERANDOS %TM, %TI, %TD E %TF - TABELAS	23
ACESSO INDIRETO	24
DECLARAÇÃO DE OPERANDOS	25
OPERANDOS RETENTIVOS	26
INSTRUÇÕES	27
RESTRIÇÕES QUANTO AO USO DE INSTRUÇÕES NOS CPS	29
REPRESENTAÇÃO GRÁFICA DAS INSTRUÇÕES	30
DESCRIÇÃO DA SINTAXE DAS INSTRUÇÕES	31
RESTRIÇÕES QUANTO AO POSICIONAMENTO DAS INSTRUÇÕES	32
PROJETO DE PROGRAMAÇÃO	34
ESTRUTURAÇÃO DE UM PROJETO DE PROGRAMAÇÃO	34
ESTADOS DE OPERAÇÃO DO CP	38

EXECUÇÃO DO PROJETO DE PROGRAMAÇÃO	40
ELABORAÇÃO DE PROJETOS DE PROGRAMAÇÃO	41
DEPURAÇÃO DE PROJETOS DE PROGRAMAÇÃO	47
TEMPOS DE CICLO DE EXECUÇÃO DO PROGRAMA	55
NÍVEIS DE PROTEÇÃO DO CP	58
INTERTRAVAMENTO DE COMANDOS NO CP	59
PROJETO DE ROTEADOR	62
ESTRUTURAÇÃO DE UM PROJETO DE ROTEADOR	62
ESTADOS DE OPERAÇÃO DO ROTEADOR	63

REFERÊNCIA DAS INSTRUÇÕES 65

LISTA DAS INSTRUÇÕES	65
CONVENÇÕES UTILIZADAS	65
INSTRUÇÕES DO GRUPO RELÉS	67
CONTATOS	68
BOBINAS	69
SLT - BOBINA DE SALTO	70
PLS - RELÉ DE PULSO	72
RM, FRM - RELÉ MESTRE, FIM DE RELÉ MESTRE	73
INSTRUÇÕES DO GRUPO MOVIMENTADORES	74
MOV - MOVIMENTAÇÃO DE OPERANDOS SIMPLES	75
MOP - MOVIMENTAÇÃO DE PARTES (SUBDIVISÕES) DE OPERANDOS	76
MOB - MOVIMENTAÇÃO DE BLOCOS DE OPERANDOS	78
MOT - MOVIMENTAÇÃO DE TABELAS	80
MES - MOVIMENTAÇÃO DE ENTRADAS/SAÍDAS	82
CES - CONVERSÃO DE ENTRADAS/SAÍDAS	84
AES - ATUALIZA ENTRADAS/SAÍDAS	85
CAB - CARREGA BLOCO	86
INSTRUÇÕES DO GRUPO ARITMÉTICAS	90
SOM - ADIÇÃO	91
SUB - SUBTRAÇÃO	92
MUL - MULTIPLICAÇÃO	93
DIV - DIVISÃO	94
AND - E BINÁRIO ENTRE OPERANDOS	95
OR - OU BINÁRIO ENTRE OPERANDOS	97
XOR - OU EXCLUSIVO ENTRE OPERANDOS	99
CAR - CARREGA OPERANDOS	101
INSTRUÇÕES DE COMPARAÇÃO DE OPERANDOS - IGUAL, MAIOR E MENOR	102
INSTRUÇÕES DO GRUPO CONTADORES	105
CON - CONTADOR SIMPLES	106
COB - CONTADOR BIDIRECIONAL	107
TEE - TEMPORIZADOR NA ENERGIZAÇÃO	109
TED - TEMPORIZADOR NA DESENERGIZAÇÃO	111
INSTRUÇÕES DO GRUPO CONVERSORES	113
B/D - CONVERSÃO BINÁRIO-DECIMAL	114
D/B - CONVERSÃO DECIMAL-BINÁRIO	115
A/D - CONVERSÃO ANALÓGICO-DIGITAL	116
D/A - CONVERSÃO DIGITAL-ANALÓGICO	118
INSTRUÇÕES DO GRUPO GERAL	120
LDI - LIGA/DESLIGA INDEXADO	121
TEI - TESTE DE ESTADO INDEXADO	123
SEQ - SEQÜENCIADOR	125
CHP - CHAMA MÓDULO PROCEDIMENTO	130
CHF - CHAMA MÓDULO FUNÇÃO	131

ECR - ESCRITA DE OPERANDOS EM OUTRO CP	134
LTR - LEITURA DE OPERANDOS DE OUTRO CP	142
LAI - LIBERA ATUALIZAÇÃO DE IMAGENS DOS OPERANDOS	144
INSTRUÇÕES DO GRUPO LIGAÇÕES	145
LGH - LIGAÇÃO HORIZONTAL	146
LGN - LIGAÇÃO NEGADA	146
LGV - LIGAÇÃO VERTICAL	146

REFERÊNCIA DOS MÓDULOS FUNÇÃO	147
--------------------------------------	------------

F-RELOG.000 - FUNÇÃO PARA ACESSO A MÓDULO RELÓGIO DE TEMPO REAL	150
INTRODUÇÃO	150
PROGRAMAÇÃO	150
F-LEDS.001 - FUNÇÃO PARA ACESSO A MÓDULO PAINEL DE LEDS	153
INTRODUÇÃO	153
PROGRAMAÇÃO	153
F-PT100.002 - FUNÇÃO PARA LEITURA DE MÓDULO PT-100	155
INTRODUÇÃO	155
PROGRAMAÇÃO	155
F-TERMO.003 - FUNÇÃO PARA LEITURA DE MÓDULO TERMOPAR	158
INTRODUÇÃO	158
PROGRAMAÇÃO	158
F-CONTR.004 - FUNÇÃO PARA ACESSO A MÓDULO CONTADOR RÁPIDO	161
PARA A SÉRIE GRANO:	161
INTRODUÇÃO	161
PARAMETRIZAÇÃO	161
PROGRAMAÇÃO	162
PARA DEMAIS SÉRIES:	165
INTRODUÇÃO	166
PROGRAMAÇÃO	166
F-CONT.005 - FUNÇÃO PARA ACESSO ÀS ENTRADAS DE CONTAGEM RÁPIDA	168
INTRODUÇÃO	168
PROGRAMAÇÃO	168
DESCRIÇÃO DO FUNCIONAMENTO	170
F-ANLOG.006 - FUNÇÃO PARA CONVERSÃO A/D OU D/A INTEGRADOS	171
INTRODUÇÃO	171
PROGRAMAÇÃO	171
F- SAIDR.009 - FUNÇÃO PARA ACESSO ÀS SAIDAS RÁPIDAS	173
INTRODUÇÃO	173
PROGRAMAÇÃO	173
F-EVENT.017 - FUNÇÃO PARA ACESSO AO MÓDULO REGISTRO DE EVENTOS	176
INTRODUÇÃO	176
PROGRAMAÇÃO	176
F-ALNET2.032 - FUNÇÃO LEITURA DE ESTATÍSTICAS DA REDE ALNET II	187
INTRODUÇÃO	187
PROGRAMAÇÃO	187
F-PID.033 - FUNÇÃO CONTROLE PID	191
INTRODUÇÃO	191
PROGRAMAÇÃO	192
F-RAIZN.034 - FUNÇÃO RAIZ QUADRADA	197
INTRODUÇÃO	197
PROGRAMAÇÃO	197
F-ARQ2.035 A F-ARQ31.042 - FUNÇÕES ARQUIVO DE DADOS	199
INTRODUÇÃO	199
PROGRAMAÇÃO	199

F-MOBT.043 - FUNÇÃO PARA MOVIMENTAÇÃO DE BLOCOS DE OPERANDOS TABELA	203
INTRODUÇÃO	203
PROGRAMAÇÃO	203
F-STMOD.045 - FUNÇÃO ESTADO DOS BARRAMENTOS E MÓDULOS DE E/S	205
INTRODUÇÃO	205
PROGRAMAÇÃO	205
F-RELG.048 - FUNÇÃO PARA ACESSO AO RELÓGIO DE TEMPO REAL	210
INTRODUÇÃO	210
PROGRAMAÇÃO	210
F-SINC.049 - FUNÇÃO PARA ACESSO AO RELÓGIO DE TEMPO REAL SINCRONIZADO	213
INTRODUÇÃO	213
PROGRAMAÇÃO	213
F-PID16.056 - MÓDULO F PARA CONTROLE PID	217
INTRODUÇÃO	217
PROGRAMAÇÃO	219
OPERANDOS	219
ENTRADAS E SAÍDAS	219
CARACTERÍSTICAS DO FUNCIONAMENTO	220
DESSATURAÇÃO DA AÇÃO INTEGRAL	220
MODOS MANUAIS	220
CONTROLE DIRETO E REVERSO	220
INTERVALO DE AMOSTRAGEM	221
TEMPO DE EXECUÇÃO	221
DESCRIÇÃO DAS POSIÇÕES DA TABELA DE PARÂMETROS	221
DESCRIÇÃO DO OPERANDO %A DE CONTROLE	222
 NOTAS DE APLICAÇÃO	 224
 SELEÇÃO DO TEMPO DE AMOSTRAGEM	 224
FEEDFORWARD/BIAS	225
CONTROLE EM CASCATA	225
CONSIDERAÇÕES IMPORTANTES	226
SUGESTÕES PARA AJUSTES DO CONTROLADOR PID	227
DETERMINAÇÃO DAS CONSTANTES DO CONTROLADOR ATRAVÉS DO PERÍODO E DO GANHO CRÍTICO	227
DETERMINAÇÃO DAS CONSTANTES DO CONTROLADOR ATRAVÉS DAS CONSTANTES DO PROCESSO	228
GANHOS X ESCALAS	230
 EXEMPLO DE APLICAÇÃO	 233
 DESCRIÇÃO DO PROCESSO	 233
DESCRIÇÃO DOS MÓDULOS ANALÓGICOS	234
PONTO DE AJUSTE	234
BLOCODIAGRAMA GERAL E VALORES LIMITES	234
PARÂMETROS DO PROCESSO	235
SINTONIA DO CONTROLADOR	236
UTILIZAÇÃO DA F-PID16.056	236
COMPARAÇÃO COM O F-PID.033	237
F-CTRL.059 - MÓDULO F PARA CONTROLE AVANÇADO	238
INTRODUÇÃO	238
PROGRAMAÇÃO	242
CARACTERÍSTICAS DE FUNCIONAMENTO	243
F-RELG.061 - FUNÇÃO PARA ACESSO AO RELÓGIO DE TEMPO REAL DO QK801 E QK2000	244
INTRODUÇÃO	244
PROGRAMAÇÃO	244

F-ALNET1.062 - FUNÇÃO INTERPRETADOR DO PROTOCOLO ALNET I PARA QK801	247
INTRODUÇÃO	247
PROGRAMAÇÃO	247
F-IMP.063 - FUNÇÃO PARA IMPRESSÃO DE CARACTERES ASCII	251
INTRODUÇÃO	251
PROGRAMAÇÃO	251
F-RECEP.064 - FUNÇÃO PARA RECEPÇÃO DE CARACTERES ASCII	254
INTRODUÇÃO	254
PROGRAMAÇÃO	254
F-UTR_S.068 - FUNÇÃO PARA AÇIONAMENTO DE SAÍDAS EM UTRS	256
INTRODUÇÃO	256
PROGRAMAÇÃO	256
F-NORM.071 - FUNÇÃO PARA NORMALIZAÇÃO	263
INTRODUÇÃO	263
PROGRAMAÇÃO	263
OPERAÇÃO	264
ENTRADAS E SAÍDAS	264
UTILIZAÇÃO	264
F-COMPF.072 - FUNÇÃO PARA MÚLTIPLAS COMPARAÇÕES	266
INTRODUÇÃO	266
PROGRAMAÇÃO	266
F-ANDT.090, F-ORT.091 E F-XORT.092 - FUNÇÕES DE OPERAÇÕES LÓGICAS ENTRE OPERANDOS	
TABELA	268
INTRODUÇÃO	268
PROGRAMAÇÃO	268
F-ETHDG.089 – FUNÇÃO DE DIAGNÓSTICO DE ETHERNET	270
INTRODUÇÃO	270
DESCRIÇÃO DOS OPERANDOS:	270
DESCRIÇÃO DAS ENTRADAS:	270
DESCRIÇÃO DAS SAÍDAS:	270
OBSERVAÇÕES IMPORTANTES:	271
F-NEGT.093 - FUNÇÃO PARA NEGAÇÃO LÓGICA DE OPERANDOS TABELA	276
INTRODUÇÃO	276
PROGRAMAÇÃO	276

APÊNDICE A TEMPOS DE EXECUÇÃO DAS INSTRUÇÕES **278**

DESCRIÇÃO DOS TEMPOS DE EXECUÇÃO	278
RELÉS	279
MOVIMENTADORES	281
ARITMÉTICAS	283
CONTADORES	285
CONVERSORES	285
GERAL	286

APÊNDICE B TEMPOS DE EXECUÇÃO DOS MÓDULOS FUNÇÃO **289**

DESCRIÇÃO DOS TEMPOS DE EXECUÇÃO	289
---	------------

GLOSSÁRIO **296**

GLOSSÁRIO DE REDES	296
GLOSSÁRIO REDES PROFIBUS	297

GLOSSÁRIO GERAL	297
PRINCIPAIS ABREVIATURAS	299

Prefácio

Descrição deste Manual

Este manual apresenta a linguagem de programação utilizada nos controladores programáveis ALTUS, bem como orientações para a elaboração de programas aplicativos. Foi escrito supondo-se familiaridade com a utilização de microcomputadores padrão IBM-PC® e ambiente operacional Windows™.

O software programador MT4000 ou MT4100, referido a partir deste ponto como MasterTool® Programming ou simplesmente MasterTool, foi desenvolvido para a programação em linguagem de relés e blocos das séries de controladores programáveis ALTUS AL-600, AL-2000, AL-3000, QUARK® e PICCOLO, bem como a configuração dos dispositivos roteadores AL-2400/S, AL-2401, QK2400 e QK2401.

Este manual está dividido em 4 capítulos e 3 apêndices.

O capítulo 1, **Introdução**, apresenta as características principais da programação de CPs e roteadores ALTUS.

O capítulo 2, **Linguagem de Diagrama de Relés**, apresenta os componentes da linguagem.

O capítulo 3, **Referência das Instruções**, descreve a função e a sintaxe de todas as instruções da linguagem.

O capítulo 4, **referência dos Módulos Função**, descreve a função e a programação dos parâmetros de entrada e saída dos módulos função fornecidos pela ALTUS.

O apêndice A, **Tempos de Execução das Instruções**, contém uma lista com os tempos de execução de cada instrução.

O apêndice B, **Tempos de Execução dos Módulos Função**, contém uma lista com os tempos de execução dos módulos função que acompanham o MasterTool.

O apêndice C apresenta um **Glossário** de termos técnicos utilizados neste manual..

Manuais Relacionados

Para maiores informações sobre o MasterTool Programming, as séries de CPs ALTUS e as redes ALNET I e ALNET II, recomendam-se os seguintes manuais:

- Manual de Utilização AL-3830
- Manual de Utilização AL-600
- Manual de Utilização AL-2000/MSP
- Manual de Utilização AL-2002/MSP
- Manual de Utilização AL-2003
- Manual de Utilização AL-3000
- Manual de Utilização dos CPs da Série QUARK
- Manual de Utilização dos CPs da Série PICCOLO
- Manual de Utilização ALNET II
- Manual de Utilização FOCOS
- Manual de Características Técnicas

- NT-031: PROTOCOLO ALNET I

Terminologia

Neste manual, as palavras “software” e “hardware” são empregadas livremente, por sua generalidade e frequência de uso. Por este motivo, apesar de serem vocábulos em inglês, aparecerão no texto sem aspas.

A abreviatura MSP significa "Multi Station Processor", ou seja, corresponde à capacidade do CP em realizar processamentos distribuídos em diversas estações.

O nome MasterTool Programming identifica o programa ALTUS para microcomputador padrão IBM-PC®, executado no ambiente operacional Windows™ 3.1 ou superior, Windows™ for Workgroups versão 3.11 ou superior ou no Windows 95/98/NT/2000, que permite o desenvolvimento de aplicativos para os CPs das séries AL-600, AL-2000, AL-3000, QUARK, PICCOLO e os dispositivos roteadores AL-2400/S, AL-2401, QK2400 e QK2401. Ao longo do manual, este programa será referido pela própria sigla ou como "programador MasterTool".

A palavra "módulo", quando se referir a hardware, é utilizada para denominar cada um dos componentes de um equipamento.

A palavra "módulo", quando se referir a software, é utilizada para denominar cada um dos componentes de um programa aplicativo.

Convenções Utilizadas

Os símbolos utilizados ao longo deste manual possuem os seguintes significados:

- Este marcador indica uma lista de itens ou tópicos.

maiúsculas PEQUENAS indicam nomes de teclas, por exemplo ENTER.

TECLA1+TECLA2 é usado para teclas a serem pressionadas simultaneamente. Por exemplo, a digitação simultânea das teclas CTRL e END é indicada como CTRL+END.

TECLA1, TECLA2 é usado para teclas a serem pressionadas sequencialmente. Por exemplo, a mensagem “Digite ALT, F10” significa que a tecla ALT deve ser pressionada e liberada e então a tecla F10 pressionada e liberada.

MAIÚSCULAS GRANDES indicam nomes de arquivos e diretórios.

Itálico indica palavras e caracteres que são digitados no teclado ou vistos na tela. Por exemplo, se for solicitado a digitar A:MASTERTOOL, estes caracteres devem ser digitados exatamente como aparecem no manual.

NEGRITO é usado para nomes de comandos ou opções, ou para enfatizar partes importantes do texto.

As mensagens de advertência apresentam os seguintes formatos e significados:

PERIGO:

O rótulo **PERIGO** indica que risco de vida, danos pessoais graves ou prejuízos materiais substanciais resultarão se as precauções necessárias não forem tomadas.

CUIDADO:

O rótulo **CUIDADO** indica que risco de vida, danos pessoais graves ou prejuízos materiais substanciais podem resultar se as precauções necessárias não forem tomadas.

ATENÇÃO:

O rótulo **ATENÇÃO** indica que danos pessoais ou prejuízos materiais mínimos podem resultar se as precauções necessárias não forem tomadas.

Contém informações importantes sobre o produto, sua operação ou uma parte do texto para a qual se deve dar atenção especial.

☺DICA:

O rótulo **DICA** indica a melhor maneira de realizar uma tarefa.

Convenções para Utilização com Mouse

Apesar do MasterTool Programming poder ser executado apenas com o uso do teclado, pode-se obter uma maior eficiência com o uso de um mouse.

Alguns termos são utilizados para descrever a ação a ser executada com o mouse para a realização de uma tarefa específica.

Termo	Significado
Clicar	Pressionar o botão principal do mouse. Normalmente o botão principal é o esquerdo, mas pode ser alterado para uso de pessoas sinistras através do Painel de Controle do Windows, através do comando Configurações, Mouse .
Clicar duas vezes ou duplo clique	Pressionar o botão duas vezes com um curto intervalo de tempo. Este intervalo de tempo pode ser configurado no Painel de Controle do Windows, através do comando Configurações, Mouse .
Arrastar	Pressionar o botão principal do mouse, mover o mouse para a posição desejada mantendo o botão pressionado e soltá-lo.

Suporte Técnico

Para acessar o Suporte Técnico ligue para (51) 589-9500 em São Leopoldo, RS, ou para o Suporte Técnico mais próximo conforme a página da Altus na INTERNET:

- www.altus.com.br
- E-MAIL: altus@altus.com.br

Caso o equipamento já esteja instalado, é aconselhável providenciar as seguintes informações antes de entrar em contato:

- Modelos de equipamentos utilizados e configuração do sistema instalado
- Número de série da UCP, revisão do equipamento e versão do software executivo, constantes na etiqueta fixada na sua lateral
- Informações do modo de operação da UCP, obtidas através do programador MASTERTOOL
- Conteúdo do programa aplicativo (módulos), obtido através do programador MASTERTOOL
- Versão do programador utilizado

Revisões deste Manual

O código de referência, da revisão e a data do presente manual estão indicados na capa. A mudança da revisão pode significar alterações da especificação funcional ou melhorias no manual.

O histórico a seguir lista as alterações correspondentes a cada revisão deste manual:

Revisão: A	Data: 11/1997
Aprovação: Luiz Gerbase	

Observações:

- Revisão inicial do manual.

Revisão: B	Data: 08/2000
Aprovação: Luiz Gerbase	
Autor: Joaquim Souza	

Observações:

- Inclusão do PL104 e PL105.
- Inclusão de novas funções.
- Retirada da hardkey e inserção do contrato de software.
- Correções de erros.

Revisão: C	Data: 08/2002
Aprovação: Luiz Gerbase	
Autor: Dimitrius Biroth Rocha	

Observações:

- Inclusão do Canal Ethernet

Revisão: D	Data: 02/2003
Aprovação: Luiz Gerbase	
Autor : Joaquim Souza, Rosana Casais	

Observações:

- Inclusão do AL-2004
- Inclusão dos operandos reais (%F, %TF e %KF)
- Inclusão do Módulo C Estendido
- Inclusão dos Módulos F-PID.056 e F-CTRL.059.

Revisão: E	Data: 11/2003
Aprovação: Luiz Gerbase	
Autor : Dimitrius Biroth Rocha	

Observações:

- Inclusão das UCPs PO3042 e PO3142;

Revisão: F	Data: 02/2004
Aprovação: Luiz Gerbase	
Autor : Dimitrius Biroth Rocha	

Observações:

- Inclusão das UCPs PO3242, PO3342, GR310, GR316 e GR330;

Revisão: G Aprovação: Luiz Gerbase Autor : Dimitrius Biroth Rocha	Data: 06/2004
---	---------------

Observações:

- Inclusão das UCPs GR350, GR351, GR370 e GR371;
- Inclusão dos módulos F-CONTR.004 e F-SAIDR.009 para as UCP da série Grano

Revisão: H Aprovação: Luiz Gerbase Autor : Dimitrius Biroth Rocha	Data: 09/2004
---	---------------

Observações:

- Inclusão dos operandos inteiros (%I, %TI e %KI)

Introdução

Bem-vindo à Linguagem de Relés e Blocos ALTUS, a linguagem que permite a construção de programas aplicativos para os CPs ALTUS a partir do MasterTool Programming.

O programa aplicativo tem como objetivo a execução de tarefas de controle. Este programa, quando carregado no controlador programável (CP), faz com que este passe a exercer as funções de controle da máquina ou processo para o qual está sendo programado.

A Linguagem de Programação

Os controladores programáveis surgiram para substituir painéis de controle a relés. Neste contexto, uma linguagem de programação que mais se aproximasse da experiência de técnicos e engenheiros seria a solução mais adequada para desenvolvimento de programas aplicativos de CPs.

Em vista disso, as instruções disponíveis para a construção do programa aplicativo no MasterTool são programadas em linguagem de relés e blocos, muito semelhante à linguagem de contatos elétricos e bobinas, utilizadas na descrição dos painéis de controle a relé.

A principal vantagem da utilização deste tipo de linguagem é seu rápido aprendizado, pois assemelha-se muito com os esquemas elétricos convencionais.

O acompanhamento e verificação de funcionamento de um programa aplicativo é similar ao de um esquema elétrico, com a vantagem de visualizar o estado dos contatos e bobinas na janela do MasterTool.

Linguagem de Diagramas de Relés

Este capítulo descreve a linguagem de Relés e Blocos ALTUS detalhando os elementos da linguagem, a estruturação modular de um programa aplicativo e a função de cada módulo.

Ao final da leitura deste capítulo será possível estruturar um programa aplicativo bem como realizar a configuração de CPs e roteadores.

Elementos de Programação

Um programa aplicativo é composto por 4 elementos básicos:

- módulos
- lógicas
- instruções
- operandos

Um programa aplicativo é composto por diversos **módulos**, permitindo uma melhor estruturação das rotinas de acordo com as suas funções. Os módulos são programados em linguagem de relés, seguindo a tendência mundial de normatização nesta área.

Um módulo de programa aplicativo é dividido em **lógicas de programação**. O formato de uma lógica de programa aplicativo utilizado nos CPs das séries AL-600, AL-2000, AL-3000, QUARK e PICCOLO permite até oito elementos em série e até quatro caminhos em paralelo.

As **instruções** são utilizadas para executar determinadas tarefas por meio de leituras e/ou alterações do valor dos operandos.

Os **operandos** identificam diversos tipos de variáveis e constantes utilizadas na elaboração de um programa aplicativo, podendo ter seu valor modificado de acordo com a programação realizada. Como exemplo de variáveis pode-se citar pontos de E/S e memórias contadoras.

Cada elemento componente do programa aplicativo é explicado em detalhes nas seções seguintes.

Organização de Memória dos CPs ALTUS

O programa aplicativo é armazenado no controlador em uma área de memória dividida em **bancos**. Podem existir um ou mais bancos de memória RAM e EPROM, conforme o modelo do CP e a sua configuração de memória, cada banco possuindo 16, 32 ou 64 Kbytes. A memória EPROM pode existir na forma de cartucho removível ou como flash EPROM interna ao CP.

Neste manual, na ajuda do MasterTool e no programador MasterTool, o nome EPROM refere-se indistintamente à memória para gravação permanente do programa aplicativo utilizada no CP, seja do tipo cartucho de EPROM ou flash EPROM.

Na janela do diretório de módulos do CP (opções **Comunicação, Módulos**) é possível visualizar a quantidade de memória livre em cada banco, para cada tipo existente no controlador. Ver **Opção Módulos** na seção **Comando Comunicação** no capítulo 4.

A tabela 2-1 mostra a capacidade de memória de programa aplicativo para cada controlador. Nesta tabela a memória flash EPROM (flash) foi distinguida do cartucho de EPROM (EPROM) para indicar com maior precisão o tipo de memória utilizado em cada modelo de CP, embora ambas sejam equivalentes quanto à operação.

Controlador		Capacidade Padrão de Bancos				Capacidade Máxima de Bancos			
		0	1	2	3	0	1	2	3
AL-600	flash	-	-	-	-	32K	32K	32K	32K
	RAM	16K	-	-	-	16K	32K	32K	32K
AL-3003	EPROM	32K	-	-	-	32K	32K	-	-
	RAM	32K	-	-	-	32K	32K	-	-
AL-3004	EPROM	16K	-	-	-	32K	-	-	-
	RAM	16K	-	-	-	16K	-	-	-
AL-2000/MSP	flash	32K	32K	-	-	32K	32K	32K	32K
	RAM	32K	-	-	-	32K	32K	32K	32K
AL-2002/MSP	flash	32K	32K	-	-	32K	32K	32K	32K
	RAM	32K	-	-	-	32K	32K	32K	32K
AL-2003	flash	16 bancos de 64 Kbytes				16 bancos de 64 Kbytes			
	RAM	64K	64K	-	-	64K	64K	-	-
AL-2004	flash	16 bancos de 64 Kbytes				16 bancos de 64 Kbytes			
	RAM	64K	64K	-	-	64K	64K	-	-
PL101, PL102,	flash	-	-	-	-	-	-	-	-
PL103	RAM	16K	-	-	-	16K	-	-	-
PL104, PL105	flash	32K	-	-	-	32K	-	-	-
	RAM	32K	32K	-	-	32K	32K	-	-
QK600	flash	-	-	-	-	32K	32K	32K	32K
	RAM	16K	-	-	-	16K	32K	32K	32K
QK800	flash	32K	-	-	-	-	-	-	-
	RAM	32K	-	-	-	-	-	-	-
QK801	flash	32K	32K	-	-	32K	32K	32K	32K
	RAM	32K	-	-	-	32K	32K	32K	32K
QK2000	flash	32K	32K	-	-	32K	32K	32K	32K
	RAM	32K	-	-	-	32K	32K	32K	32K

Tabela 2-1 Capacidade de memória de Programa Aplicativo dos CPs

Os valores dos operandos numéricos (%M, %D, %F, %TM, %TD, %TF) são armazenados em área separada do programa, com diferentes tamanhos de acordo com o modelo do CP. Pode-se consultar a quantidade de memória de operandos livre na janela de edição do módulo C no quadro de operandos. Para maiores informações sobre a janela de edição do módulo C, ver seção **Janelas de Edição**, no capítulo 3 do Manual de Utilização.

Os operandos binários (%E, %S e %A) possuem área permanentemente reservada para os seus valores na memória interna do microprocessador.

O uso da memória de operandos é apresentada em detalhes na seção **Declaração de Operandos**, neste mesmo capítulo.

Para maiores informações sobre as capacidades e a organização de memória de cada controlador, consultar os seus respectivos Manuais de Utilização (ver seção **Manuais Relacionados**, no prefácio deste manual).

Lógicas

Chama-se **lógica** a matriz de programação formada por 32 células (elementos da matriz) dispostas em 4 linhas (0 a 3) e 8 colunas (0 a 7). Em cada uma das células podem ser colocadas instruções, podendo-se programar até 32 instruções em uma mesma lógica.

Cada lógica presente no programa simula um pequeno trecho de um diagrama de relés real. A figura 2-1 mostra o formato de uma lógica do programa aplicativo.

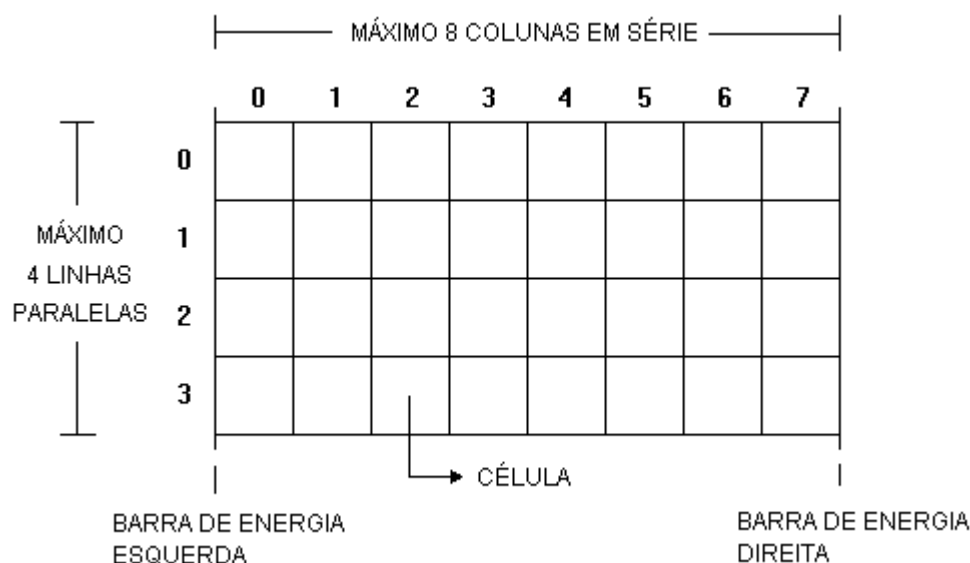


Figura 2-1 Formato de uma Lógica

As duas linhas laterais da lógica representam barras de energia entre as quais são colocadas as instruções a serem executadas.

Estão disponíveis para a programação instruções simbólicas tipicamente encontradas em diagramas, tais como contatos, bobinas, ligações e instruções representadas em caixas, como temporizadores, contadores e aritméticas.

A lógica deve ser programada de forma que bobinas e entradas das instruções de caixas sejam "energizadas" a partir do fechamento de um fluxo de "corrente" da esquerda para a direita entre as duas barras, através de contatos ou das saídas das caixas interligados. Entretanto, o fluxo de "corrente elétrica" simulado em uma lógica flui somente no sentido da barra de energia esquerda para a direita, diferentemente dos esquemas elétricos reais. O conceito utilizado simplifica sobremaneira o projeto lógico de relés, uma vez que não é necessário a preocupação com caminhos de fuga de corrente.

O processamento das instruções de uma lógica é realizado em colunas, desde a coluna 0 até a 7. Uma coluna é processada na ordem sequencial de suas linhas, desde a linha 0 até a linha 3. A figura 2-2 mostra a ordem de processamento das células da lógica. O número existente dentro de cada célula indica a sua ordem de processamento.

	0	1	2	3	4	5	6	7
0	1	5	9	13	17	21	25	29
1	2	6	10	14	18	22	26	30
2	3	7	11	15	19	23	27	31
3	4	8	12	16	20	24	28	32

Figura 2-2 Ordem de Processamento das Células na Lógica

Operandos

Operandos são elementos utilizados pelas instruções do MasterTool na elaboração de um programa aplicativo. Os operandos podem definir valores constantes, definidos no momento da programação, ou variáveis, identificadas através de um endereço ou de um tag, com valores possíveis de serem alterados durante a execução do programa aplicativo.

Identificação de um Operando pelo Endereço

A identificação e utilização de um operando pelo seu endereço é caracterizada pelo caractere % como primeiro caractere do nome. O restante do nome utilizado deve seguir às regras de formatação de endereço de operandos.

O formato de cada operando pode ser visto na seção **Identificação de Operandos Simples** e nas subseqüentes, neste mesmo capítulo.

Identificação de um Operando pelo Tag

A identificação e utilização de um operando pelo seu tag é caracterizada pela utilização de um nome, com até 7 caracteres (alfanuméricos), que pode ser atribuído a qualquer operando, exceto constantes. Este nome passa a representar o operando nos processos de programação, monitoração, depuração e documentação de um programa aplicativo.

O MasterTool não permite a utilização de TAGs para operandos do tipo constante (%KM ou %KD).

Ex.:

Atribui-se o tag **CONT1** ao operando **%M0000**.

Sempre que o operando **%M0000** necessite ser utilizado na edição do programa aplicativo, pode-se utilizar o seu tag **CONT1**.

©DICA:

A escolha do nome do tag para o operando deve refletir ao máximo a função que o conteúdo do operando executa no programa aplicativo. Ex.: **TANQUE1**, armazena o volume do tanque 1.

A identificação de um operando pelo seu endereço poderá ser feita sempre, uma vez que todo operando possui um endereço. A identificação de um operando pelo seu tag, somente poderá ser feita após à atribuição do tag a um operando.

A atribuição de tags à operandos pode ser feita através do comando **Operandos** do menu **Relatório** ou diretamente no momento da programação. No segundo caso, ao preencher o nome de um operando de uma instrução com um tag não existente, é indicada a inexistência da definição do tag, e solicitado o tipo de operando para qual o tag deve ser criado.

Para maiores informações sobre criação e atribuição de tags para operandos, ver seções sobre o comando **Relatório, Operandos**, no capítulo 4 e **Inserindo Tags e Comentários de Operandos**, no capítulo 5 do Manual de Utilização do MasterTool

Os operandos também podem ser visualizados através de seu wire-info associado. No entanto, um operando não pode ser forçado ou monitorado digitando-se o wire-info ao invés do tag ou endereço.

Operandos Utilizados no MasterTool

São mostrados na tabela 2-2 os operandos disponíveis no MasterTool:

Tipo	Operando
%E	Relés de Entrada
%S	Relés de Saída
%R	Endereço no Barramento
%A	Relés Auxiliares
%M	Memórias
%D	Decimais
%F	Reais
%KM	Constantes Memórias
%KD	Constantes Decimais
%KF	Constantes Reais
%TM	Tabelas Memórias
%TD	Tabelas Decimais
%TF	Tabelas Reais

Tabela 2-2 Operandos Utilizados no MasterTool

Os operandos dividem-se em 3 grupos:

- operandos simples
- operandos constante
- operandos tabela

Identificação dos Operandos Simples

Os operandos simples são utilizados como variáveis de armazenamento de valores no programa aplicativo. Conforme a instrução que os utilizam, eles podem ser referenciados na sua totalidade ou em uma subdivisão (uma parte do operando). As subdivisões de operandos podem ser **palavra**, **octeto**, **nibble** ou **ponto**.

O formato geral de um operando simples pode ser visto na figura 2-3.

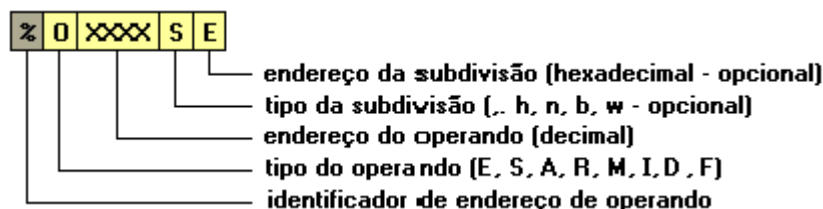


Figura 2-3 Formato de um Operando Simples

Tipo do operando:

- %E - entrada
- %S - saída
- %A - auxiliar
- %M – memória
- %I - inteiro
- %D - decimal
- %F - real

Tipo da subdivisão:

- . - ponto da palavra baixa (1 ponto)
- h - ponto da palavra alta (1 ponto)
- n - nibble (4 pontos)
- b - octeto (8 pontos)
- w - palavra (16 pontos)

Exemplos de Endereços:

- %E0002.3 - ponto 3 do operando de entrada 2
- %S0004.7 - ponto 7 do operando de saída 4
- %A0039n1 - nibble 1 do operando auxiliar 39
- %A0045 - octeto auxiliar 45
- %I0234 – operando inteiro 234
- %M0205 - operando memória 205
- %M0205b0 - octeto 0 da memória 205
- %D0029 - operando decimal 29
- %D0034w1 - palavra 1 do operando decimal 34
- %F0001 – operando real 1

Exemplos de tags:

- FORNO
- LIMSUP
- CHAVE1

Identificação dos Operandos Constante

Os operandos constante são utilizados para a definição de valores fixos durante a edição do programa aplicativo.

O formato geral de um operando constante pode ser visto na figura a seguir.

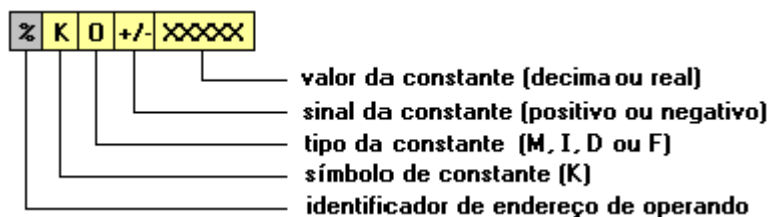


Figura 2-4 Formato de um Operando Constante

Tipo da constante:

- %M memória
- %I inteiro
- %D decimal
- %F real

Exemplos:

- %KM+05172 - constante memória positiva
- %KI-1 – constante inteira negativa
- %KD-0974231 - constante decimal negativa
- %KF+0153.78 – constante real positiva

Identificação dos Operandos Tabela

Tabelas de operandos são conjuntos de operandos simples, constituindo arranjos unidimensionais. São utilizados índices para determinar a posição da tabela que se deseja ler ou alterar. São possíveis tabelas de operandos memória ou decimal.

O formato geral de um operando tabela pode ser visto na figura 2-5.

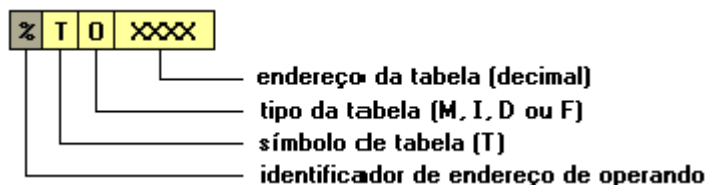


Figura 2-5 Formato de um Operando Tabela

Tipo da tabela:

- %TM memória
- %TI inteiro
- %TD decimal
- %TF real

Exemplos:

- %TM0026 - tabela memória 26
- %TI0020 – tabela inteiro 20
- %TD0015 - tabela decimal 15
- %TF0069 – tabela real 69

Operandos %E - Relés de Entrada

São operandos usados para referenciar pontos de módulos digitais de entrada. Sua quantidade é determinada pelo número de módulos de E/S que estão dispostos nos bastidores que compõem o sistema. Ver item **Configurando o Barramento** na seção **Configurando o Módulo C** no capítulo 5 do Manual de Utilização do MasterTool.

Os operandos %E são normalmente utilizados em instruções binárias (contatos, bobinas) e de movimentação. Ocupam um byte de memória (8 bits), armazenando os valores dos pontos diretamente em cada bit. Os valores dos operandos são armazenados na memória interna do microprocessador, não utilizando o espaço disponível ao programa aplicativo.

Os formatos dos operandos %E podem ser vistos na figura 2-6.

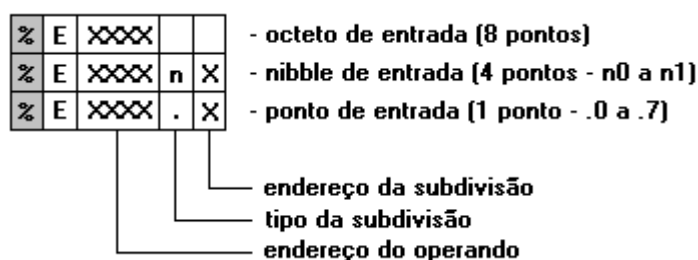


Figura 2-6 Formatos dos Operandos %E

Exemplos:

- %E0018.6 - ponto 6 do octeto de entrada 18
- %E0021n0 - nibble 0 do octeto de entrada 21
- %E0025 - octeto de entrada 25

Operandos %S - Relés de Saída

São operandos usados para referenciar pontos de módulos digitais de saída. Sua quantidade é determinada pelo número de módulos de E/S que estão dispostos nos bastidores que compõem o sistema. Ver item **Configurando o Barramento** na seção **Configurando o Módulo C** no capítulo 5 do Manual de Utilização do MasterTool.

Os operandos %S são utilizados em instruções binárias (contatos, bobinas) e de movimentação. Ocupam um byte de memória (8 bits), armazenando os valores dos pontos diretamente em cada bit.

Os valores dos operandos são armazenados na memória interna do microprocessador, não utilizando o espaço disponível ao programa aplicativo.

Os formatos dos operandos %S podem ser vistos na figura 2-7.

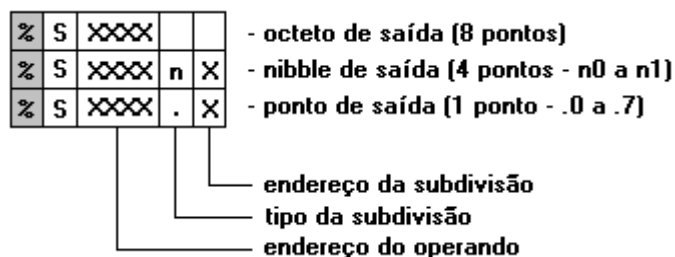


Figura 2-7 Formatos dos Operandos %S

Exemplos:

- %S0011.2 - ponto 2 do octeto de saída 11
- %S0010n1 - nibble 1 do octeto de saída 10
- %S0015 - octeto de saída 15

Operandos %A - Relés Auxiliares

Os relés auxiliares são operandos usados para armazenamento e manipulação de valores binários intermediários no processamento do programa aplicativo. Sua quantidade nos controladores é fixa (ver seção **Declaração de Operandos** neste mesmo capítulo).

Operandos %A são utilizados em instruções binárias (contatos, bobinas) e de movimentação. Ocupam um byte de memória (8 bits), armazenando valores diretamente em cada bit. Os valores dos operandos são armazenados na memória interna do microprocessador, não utilizando o espaço disponível ao programa aplicativo.

Os formatos dos Operandos %A podem ser vistos na figura 2-8

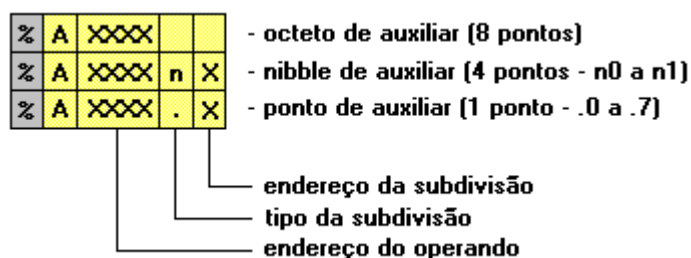


Figura 2-8 Formatos dos Operandos %A

Exemplos:

- %A0032.7 - ponto 7 do auxiliar de saída 32
- %A0087n1 - nibble 1 do auxiliar de saída 87
- %A0024 - octeto auxiliar 24

Operandos %R - Endereços no Barramento

São operandos usados para referenciar pontos ou octetos no barramento de módulos de entrada e saída do controlador. Estes operandos representam apenas endereços do barramento, não

armazenando valores nem ocupando espaço de memória. São utilizados em algumas instruções ou funções que realizam acessos a módulos.

Os formatos dos operandos %R podem ser vistos na figura 2-9.

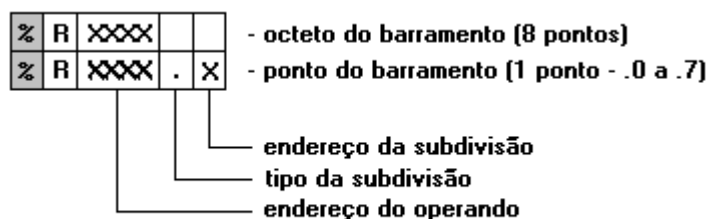


Figura 2-9 Formatos dos Operandos %R

Nos CPs **AL-3003, AL-3004** e nos **barramentos 0 e 1 do AL-2002/MSP, AL-2003 e do AL-2004**, cada posição no barramento corresponde a 8 octetos de operandos %R. Desta forma, na posição 0 tem-se os operandos %R0000 a %R0007; na posição 1, %R0008 a %R0015, e assim por diante.

Para se obter o primeiro operando de uma posição do barramento, basta calcular:

Endereço do Octeto = Posição no Barramento X 8

Nos CPs **PL101, PL102 e PL103**, cada posição no barramento corresponde a 4 octetos de operandos %R. Desta forma, na posição 0 tem-se os operandos %R0000 a %R0003; na posição 1, %R0004 e %R0007, e assim por diante.

Para se obter o primeiro operando de determinada posição do barramento, basta calcular:

Endereço do Octeto = Posição no Barramento X 4

Nos CPs **AL-600, AL-2000/MSP, QK800, QK801, QK2000/MSP** e nos **barramentos de 2 a 9 do AL-2002/MSP, AL-2003 e do AL-2004**, cada posição no barramento corresponde a 2 octetos de operandos %R. Desta forma, na posição 0 tem-se os operandos %R0000 e %R0001; na posição 1, %R0002 e %R0003, e assim por diante.

Para se obter o primeiro operando de determinada posição do barramento, basta calcular:

Endereço do Octeto = Posição no Barramento X 2

Os endereços para cada posição do barramento são automaticamente mostrados na janela de declaração do barramento na coluna **Endereços** (ver item **Configurando o Barramento** na seção **Configurando o Módulo C** no capítulo 5 do Manual de Utilização do MasterTool).

Exemplos:

- %R0026 - octeto 26 do barramento
- %R0015.7 - ponto 7 do octeto 15 do barramento

Operandos %M - Memórias

Os operandos %M são usados para processamento numérico, armazenando valores em precisão simples, com sinal.

Os formatos dos operandos %M podem ser vistos na figura 2-10.

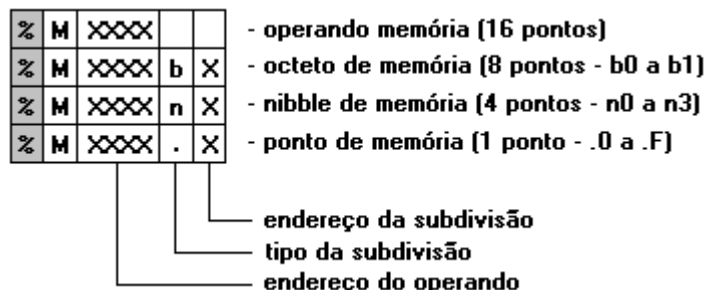


Figura 2-10 Formatos dos Operandos %M

A quantidade de operandos memória é configurável na declaração do módulo C, sendo o limite máximo dependente do modelo de CP em uso (ver seção **Declaração de Operandos** neste mesmo capítulo).

Os operandos %M são utilizados em instruções de movimentação, comparação, aritméticas, contagem, temporização e de conversão. Podem ser utilizados em contatos, da mesma forma que os operandos %E, %S e %A. Estes operandos ocupam dois bytes de memória (16 bits), armazenando o valor no formato complemento de dois (2'), conforme a figura 2-11.

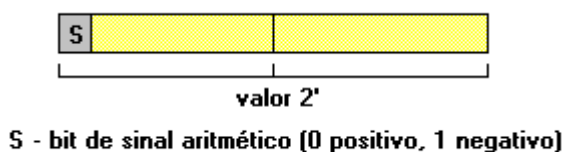


Figura 2-11 Formato do Operando Memória

Exemplos:

- %M0032 - memória 32
- %M0072n1 - nibble 1 da memória 72
- %M0084.F - ponto 15 da memória 84

Operandos %D - Decimais

Os operandos %D são usados para processamento numérico, armazenando valores em formato BCD com até 7 dígitos e sinal.

Os formatos dos operandos %D podem ser vistos na figura 2-12.

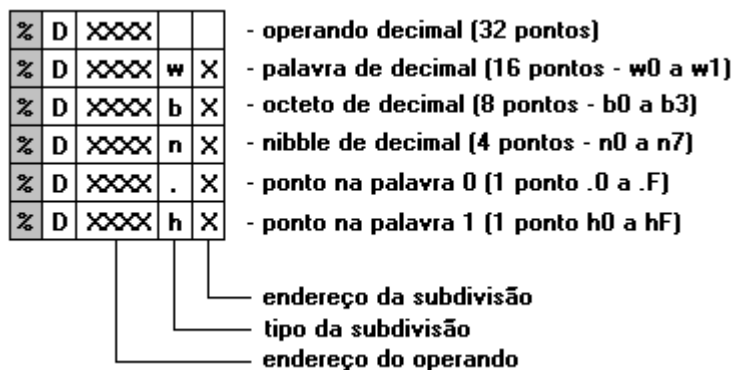


Figura 2-12 Formatos dos Operandos %D

A quantidade de operandos decimal é configurável na declaração do módulo C, sendo o limite máximo dependente do modelo de CP em uso (ver seção **Declaração de Operandos** neste mesmo capítulo).

Os operandos %D são utilizados em instruções de movimentação, comparação, aritméticas, e conversão. Podem ser utilizados em contatos, da mesma forma que os operandos %E, %S e %A. Estes operandos ocupam quatro bytes de memória (32 bits), armazenando o valor no formato BCD (cada dígito ocupa 4 bits), com sinal, conforme a figura 2-13.



Figura 2-13 Formato do Operando Decimal

Exemplos:

- %D0041 - decimal 41
- %D0023b2 - octeto 2 do decimal 23
- %D0059n6 - nibble 6 da memória 59
- %D0172hA - ponto 10 da palavra 1 da memória 172

Operandos %F - Reais

Os formatos dos operandos %F podem ser vistos na figura a seguir:

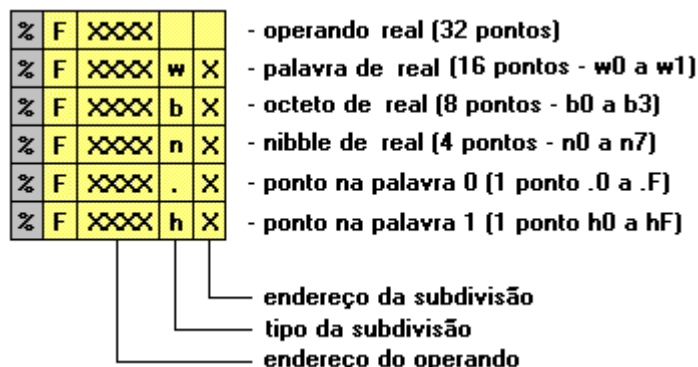


Figura 2-14 Formatos dos Operandos %F

A quantidade de operandos real é configurável na declaração do módulo C, sendo o limite máximo dependente do modelo de CP em uso (ver seção **Declaração de Operandos** neste mesmo capítulo).

Os operandos %F são usados para processamento numérico, armazenando valores em 32 bits com ponto flutuante, precisão simples e sinal, conforme a norma IEEE 754. Estes operandos ocupam quatro bytes de memória (32 bits), armazenando o valor conforme a figura a seguir:

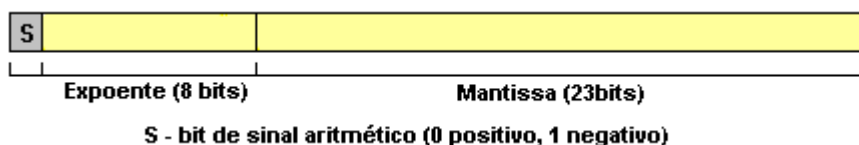


Figura 2-15 Formato do Operando Real

O valor de um operando real (%F) é obtido pela seguinte expressão:

$$\text{Valor} = \text{Sinal} \times 1.\text{Mantissa} \times 2^{(\text{Expoente} - 127)}$$

Sendo assim, a faixa de valores armazenáveis vai de -3,4028234663852886E+38 a 3,4028234663852886E+38.

Valores cujo módulo é maior que zero e menor que 1,1754943508222875E-38, são tratados como zero pelos CPs, por serem muito pequenos. Os CPs não tratam os seguintes casos especiais previstos na norma: números não normalizados, infinito e NaNs (not a number).

Exemplo:

- %F0032 – real 32

Operandos %l - Inteiro

Os operandos %I são usados para processamento numérico, armazenando valores em precisão simples, com sinal. A diferença básica entre este tipo de operando e o operando Memória %M, é que o operando Inteiro %I possui 32 bits.

Os formatos dos operandos %I podem ser vistos na figura a seguir.

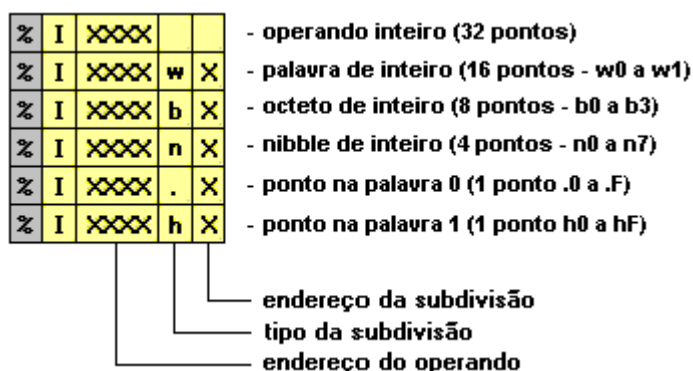


Figura 2-16 Formatos dos Operandos %I

A quantidade de operandos inteiro é configurável na declaração do módulo C, sendo o limite máximo dependente do modelo de CP em uso (ver seção Declaração de Operandos neste mesmo capítulo).

Os operandos %I são utilizados em instruções de movimentação, comparação, aritméticas, e conversão. Estes operandos ocupam quatro bytes de memória (32 bits), com sinal, conforme a figura a seguir:



Figura 2-17 Formato do Operando Inteiro

Exemplos:

- %I0041 - inteiro 41
- %I0023b2 - octeto 2 do inteiro 23
- %I0059n6 - nibble 6 do inteiro 59
- %I0172hA - ponto 10 da palavra 1 do inteiro 172

Operandos %KM, %KI, %KD e %KF - Constantes

São operandos usados para a definição de valores fixos na elaboração do programa aplicativo. Existem tipos de constante, %KM, %KI, %KD e %KF, cada um seguindo um formato diferente de representação de valores, sendo idênticos aos operandos %M, %I, %D e %F, respectivamente.

O formato dos operandos constantes pode ser visto na figura a seguir.

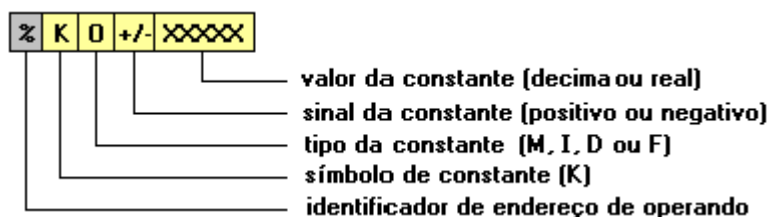


Figura 2-18 Formato dos Operandos Constantes

Estes operandos são utilizados em instruções de movimentação, comparação, aritméticas, contagem e de temporização.

Exemplos:

- %KM+00241 - constante memória + 241
- %KI+2000000000 – constante inteiro 2 bilhões ou 2×10^9
- %KD-0019372 - constante decimal - 19.372
- %KF+0125.78 – constante real + 125.78
- %KF+3.1415E23 – constante real 3.1415×10^{23}

As constantes reais podem conter até 8 dígitos significativos.

Operandos %TM, %TI, %TD e %TF - Tabelas

Tabelas de operandos são conjuntos de operandos simples, constituindo arranjos unidimensionais com a finalidade de armazenar valores numéricos. Cada tabela possui uma quantidade de posições configurável, onde cada posição pode conter exatamente os mesmos valores de um operando %M, %I, %D ou %F, se a tabela for do tipo %TM, %TI, %TD ou %TF, respectivamente.

O formato dos operandos tabelas pode ser visto na figura a seguir:

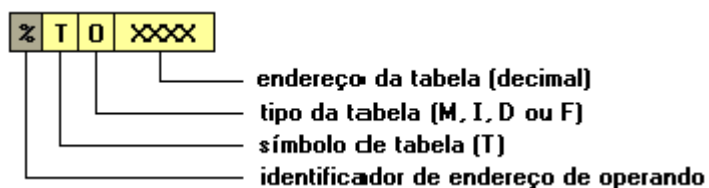


Figura 2-19 Formato dos Operandos Tabelas

A quantidade de tabelas e o número de posições de cada uma é configurável na declaração do módulo C. Podem ser definidas até 255 tabelas totais e até o máximo de 255 posições em cada tabela, respeitando-se o limite da memória de operandos do CP.

As tabelas são utilizadas em instruções de movimentação.

Acesso Indireto

Esta forma de acesso é usada em conjunto com um operando memória %M para referenciar indiretamente outros operandos do sistema.

O sinal *, colocado na frente de um tipo de operando, indica que este é referenciado pelo endereço contido na memória especificada à esquerda do sinal.

O formato do acesso indireto pode ser visto na figura a seguir:

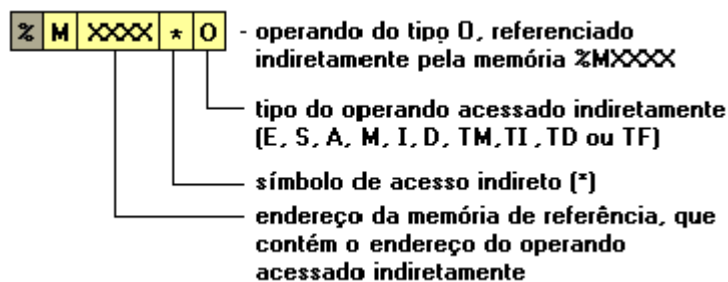


Figura 2-20 Formato de um Acesso Indireto

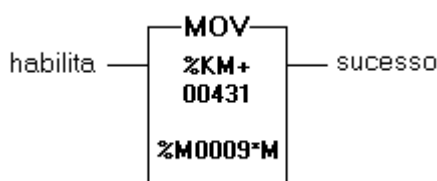
No MasterTool, o acesso indireto às tabelas é representado sem o asterisco.

O acesso indireto é utilizado em instruções de movimentação, comparação, contagem e de temporização.

Exemplos:

- %M0043*E - octeto de entrada referenciada indiretamente pela memória 43
- %M1824*A - octeto auxiliar referenciado indiretamente pela memória 1824
- %M0371TD - tabela de decimais referenciada indiretamente pela memória 371
- %M0009*M - operando memória referenciado indiretamente pela memória 9
- %F0045*M - operando real referenciado indiretamente pela memória 45

Exemplo:



Esta instrução movimenta o valor +431 para o operando memória cujo endereço é o valor correntemente armazenado em %M0009. Se %M0009 contiver o valor 32, então o valor +431 será armazenado em %M0032. Se %M0009 contiver o valor 12 então o valor constante será armazenado em %M0012.

ATENÇÃO:

É responsabilidade do programa aplicativo que o valor contido na memória de referência (%M0009, no exemplo) represente endereços válidos, não contendo valores negativos ou acima dos endereços existentes para o tipo de operando referenciado indiretamente. As instruções não realizam os acessos indiretos inválidos, normalmente possuindo um sinal de saída para a indicação do erro.

Se no programa do exemplo anterior houvessem sido declarados 256 operandos %M, o valor de %M0009 deveria estar entre 0 e 255 para que a instrução fosse corretamente executada. Caso o valor não estivesse nesta faixa, o acesso não seria realizado.

Declaração de Operandos

Os operandos %E, %S e %A ocupam áreas de memórias próprias, permanentemente reservadas no microprocessador do CP. A quantidade destes operandos nos controladores, portanto, é constante.

Os operandos %R não ocupam espaço em memória, sendo apenas endereços para o acesso aos barramentos.

Por representarem valores fixos, os operandos constante (%KM, %KI, %KD e %KF) também não ocupam espaço em memória, sendo armazenados no próprio programa aplicativo na etapa de programação. Não há limites no número de operandos constante utilizados no programa.

Pode-se declarar a quantidade de operandos %M, %I, %D, %F, %TM, %TI, %TD e %TF, ocupando estes uma área de memória RAM própria da UCP em uso. A tabela a seguir mostra a capacidade máxima de memória para o armazenamento destes operandos em cada controlador. Os operandos %E, %S e %A não ocupam esta área.

Controlador	Memória de Operandos %M, %D, %TM e %TD
AL-600	8 Kbytes
AL-3003	11,5 Kbytes
AL-3004	11,5 Kbytes
AL-2000/MSP	15,5 Kbytes
AL-2002/MSP	15,5 Kbytes
AL-2003	48 Kbytes
AL-2004	48 Kbytes
QK600	8 Kbytes
QK800	15,5 Kbytes
QK801	15,5 Kbytes
QK2000	15,5 Kbytes
PL101, PL102, PL103	11,5 Kbytes
PL104, PL105	15,5 Kbytes

Tabela 2-3 Capacidade de Memória dos Operandos Numéricos do CP

A declaração dos operandos é realizada através da janela de edição do módulo C do MasterTool, sendo armazenada no módulo C. A quantidade de operandos declarada deve se adequar à capacidade máxima de memória disponível. Ver itens **Configurando Operandos Simples**, **Configurando Operandos Tabelas** e **Configurando Operandos Retentivos** na seção **Configurando o Módulo C** no Manual de Utilização do MasterTool.

Deve-se declarar uma quantidade mínima de operandos memória (%M) que comporte os bytes de diagnóstico utilizados nos módulos do barramento.

A reserva dos operandos %M, %I, %D e %F é realizada em blocos de 256 bytes. No caso de operandos memória, esta quantidade corresponde a 128 operandos. Em operandos decimais e reais, correspondem a 64 operandos.

Os operandos %TM, %TI, %TD e %TF são declarados informando-se o número de tabelas necessário para cada tipo e o número de posições que cada tabela contém. É possível a definição de até 255 tabelas totais e até 255 posições para uma tabela, respeitando-se o limite da memória RAM de operandos.

A tabela a seguir, mostra o espaço de memória ocupado por cada tipo de operando e onde os seus valores são armazenados.

Operando	Ocupação de Memória	Localização
%E - entrada	1 byte	Microprocessador
%S - saída	1 byte	Microprocessador
%A - auxiliar	1 byte	Microprocessador
%KM - constante M	-	-
%KI - constante I	-	-
%KD - constante D	-	-
%KF - constante F	-	-
%M - memória	2 bytes	RAM de operandos
%I - inteiro	4 bytes	RAM de operandos
%D - decimal	4 bytes	RAM de operandos
%F - real	4 bytes	RAM de operandos
%TM - tabela M	2 bytes por posição	RAM de operandos
%TI - tabela I	4 bytes por posição	RAM de operandos
%TD - tabela D	4 bytes por posição	RAM de operandos
%TF - tabela F	4 bytes por posição	RAM de operandos

Tabela 2-4 Ocupação de Memória e Localização dos Operandos

Operandos Retentivos

Operandos retentivos são operandos que têm seus valores preservados quando a UCP é desenergizada (desligada). Os operandos não retentivos têm o seu valor zerado no momento em que o controlador programável é ligado.

Todos os operandos tabela são sempre retentivos. É possível configurar a quantidade de operandos %M (memória), %I (inteiro), %D (decimal), %F (real), %S (saída) e %A (auxiliar) retentivos.

Os operandos retentivos são configurados a partir dos últimos endereços até os iniciais, obedecendo a mesma regra dos operandos simples. Ou seja, a reserva é realizada em blocos de 256 bytes para operandos numéricos. A declaração dos operandos %S e %A é realizada de octeto em octeto.

Por exemplo, se existem 512 operandos %M declarados (%M0000 a %M0511) e deseja-se que 128 desses operandos sejam retentivos, serão considerados retentivos os operandos %M0384 ao %M0511.

Ver item **Configurando Operandos Retentivos** na seção **Configurando o Módulo C** no capítulo 5 do Manual de Utilização do MasterTool.

Instruções

Os CPs ALTUS utilizam a linguagem de relés e blocos para a elaboração do programa aplicativo, cuja principal vantagem, além de sua representação gráfica, é ser similar a diagramas de relés convencionais.

A programação desta linguagem, realizada através do MasterTool, utiliza um conjunto de poderosas instruções apresentadas no capítulo 3 **Referência das Instruções**, neste manual.

As instruções do MasterTool podem ser divididas em 7 grupos:

- **RELÉS** contendo as instruções:
 - **RNA** contato normalmente aberto
 - **RNF** contato normalmente fechado
 - **BOB** bobina simples
 - **BBL** bobina liga
 - **BBD** bobina desliga
 - **SLT** bobina de salto
 - **PLS** relé de pulso
 - **RM** relé mestre
 - **FRM** fim de relé mestre
- **MOVIMENTADORES** contendo as instruções:
 - **MOV** movimentação de operandos simples
 - **MOP** movimentação de partes de operandos
 - **MOB** movimentação de blocos de operandos
 - **MOT** movimentação de tabelas de operandos
 - **MES** movimentação de entradas ou saídas
 - **CES** conversão de entradas ou saídas
 - **AES** atualização de entradas ou saídas
 - **CAB** carrega bloco de constantes
- **ARITMÉTICOS** contendo as instruções:
 - **SOM** soma
 - **SUB** subtração
 - **MUL** multiplicação
 - **DIV** divisão
 - **AND** função "e" binário entre operandos
 - **OR** função "ou" binário entre operandos
 - **XOR** função "ou exclusivo" binário entre operandos
 - **CAR** carrega operando
 - **=** igual
 - **<** menor
 - **>** maior

- **CONTADORES** contendo as instruções:
 - **CON** contador simples
 - **COB** contador bidirecional
 - **TEE** temporizador na energização
 - **TED** temporizador na desenergização


- **CONVERSORES** contendo as instruções:
 - **B/D** conversão binário - decimal
 - **D/B** conversão decimal - binário
 - **A/D** conversão analógico - digital
 - **D/A** conversão digital – analógico


- **GERAIS** contendo as instruções:
 - **LDI** liga ou desliga pontos indexados
 - **TEI** teste de estado de pontos indexados
 - **SEQ** seqüenciador
 - **CHP** chama módulo procedimento
 - **CHF** chama módulo função
 - **ECR** escrita de operandos em outro CP
 - **LTR** leitura de operandos de outro CP
 - **LAI** libera atualização de imagem de operandos

- **LIGAÇÕES** contendo as instruções:
 - **LGH** ligação horizontal
 - **LGV** ligação vertical
 - **LGN** ligação negada

Restrições Quanto ao Uso de Instruções nos CPs

A linguagem de relés e blocos é perfeitamente compatível entre os CPs programados pelo MasterTool. Devido a características de funcionamento, contudo, algumas instruções não estão disponíveis em todos os controladores. Na tabela 2-9 estão relacionadas as instruções e os controladores nos quais as mesmas não podem ser utilizadas.

 - indica que o CP possui a instrução

 - indica que o CP não possui a instrução

UCPs Instrução	AL-600	AL-3003 AL-3004	QK600, QK800, QK801	PL101, PL102, PL103, PL104, PL105	AL-2000, AL-2002, AL-2003, AL-2004, QK2000
CES					
A/D					
D/A					
ECR					
LTR					
LAI					

Tabela 2-5 Instruções Inexistentes em Determinados CPs

O MasterTool não permite inserir no programa aplicativo uma instrução que não possa ser executada no CP para o qual está configurado.

ATENÇÃO:

Ao editar um módulo de programa aplicativo, o tipo de UCP declarado no item **Modelo de UCP** da janela de edição do módulo C deve ser o da UCP onde o programa será executado.

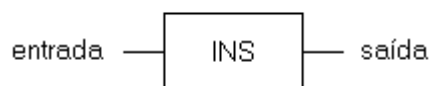
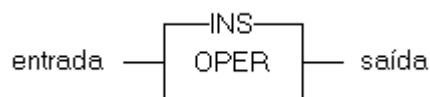
ATENÇÃO:

Caso se deseje mudar o tipo de UCP para outro, após o programa estar editado, deve-se procurar e remover as instruções que não podem ser utilizadas no novo tipo de UCP. Este procedimento deve ser realizado em todos os módulos do programa.

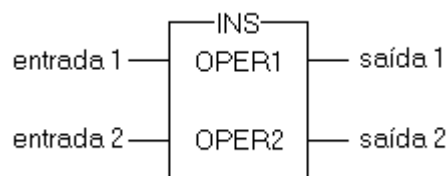
Representação Gráfica das Instruções

Nas figuras a seguir estão representadas as configurações máximas de entradas e saídas em cada tipo, não sendo necessariamente todas utilizadas em uma determinada instrução.

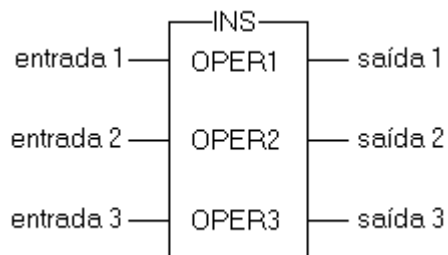
Instruções com uma célula



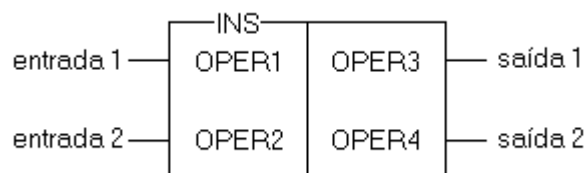
Instruções com duas células



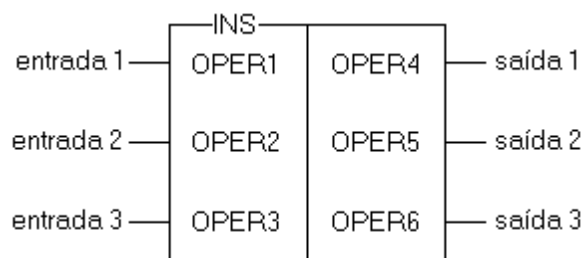
Instruções com três células



Instruções com quatro células



Instruções com seis células



Descrição da Sintaxe das Instruções

A descrição dos operandos possíveis de serem programados nas células de cada instrução é realizada de acordo com o formato mostrado na figura a seguir:

OPER1	OPER2	OPER3
LISTA DE OPERANDOS POSSÍVEIS NA CÉLULA	LISTA DE OPERANDOS POSSÍVEIS NA CÉLULA	LISTA DE OPERANDOS POSSÍVEIS NA CÉLULA

Figura 2-21 Formato da Sintaxe das Instruções

Várias combinações diferentes de operandos podem ser especificadas para uma mesma instrução.

Exemplo:

OPER1	OPER2	OPER1	OPER2
%M	%M %M*M %KM	%D	%D %M*D %KD

Esta declaração de sintaxe significa que, como primeiro operando, podem ser utilizados %M ou %D. Se o primeiro operando for %M, o segundo somente poderá ser %KM, %M ou %M * M (acesso indireto a memória). Se o primeiro for %D, o segundo somente poderá ser %KD, %D ou %M * D (acesso indireto a decimal).

Restrições Quanto ao Posicionamento das Instruções

Existem regras a serem respeitadas quanto ao posicionamento das instruções nas 8 colunas da lógica. Pode-se dividir as instruções em três categorias:

- Instruções que podem ser editadas somente na coluna 7:
 - **BOB** bobina simples
 - **BBL** bobina liga
 - **BBD** bobina desliga
 - **SLT** bobina de salto
 - **RM** relé mestre
 - **FRM** fim de relé mestre

- Instruções que podem ser editadas nas colunas 0 a 6:
 - **RNA** relé normalmente aberto
 - **RNF** relé normalmente fechado
 - **PLS** relé de pulso
 - **LGH** ligação horizontal
 - **LGV** ligação vertical
 - **LGN** ligação negada
 - **DIV** divisão
 - **MOB** movimentação de blocos de operandos
 - **>** maior
 - **<** menor
 - **=** igual
 - **SEQ** seqüenciador
 - **CHF** chama módulo função
 - **ECR** escrita de operandos em outro CP
 - **LTR** leitura de operandos de outro CP

- Instruções que podem ser editadas em todas as colunas:
 - **MOV** movimentação de operandos simples
 - **MOP** movimentação de partes de operandos
 - **MOT** movimentação de tabelas de operandos
 - **MES** movimentação de entradas ou saídas
 - **CES** conversão de entradas ou saídas
 - **AES** atualização de entradas ou saídas
 - **CAB** carrega bloco de constantes
 - **SOM** soma
 - **SUB** subtração
 - **MUL** multiplicação
 - **AND** função 'e' binário entre operandos

• OR	função 'ou' binário entre operandos
• XOR	função 'ou exclusivo' binário entre operandos
• CON	contador simples
• COB	contador bidirecional
• TEE	temporizador na energização
• TED	temporizador na desenergização
• B/D	conversão binário - decimal
• D/B	conversão decimal - binário
• CAR	carrega operando
• LDI	liga ou desliga pontos indexados
• TEI	teste de estado de pontos indexados
• CHP	chama módulo procedimento
• LAI	libera atualização de imagem
• A/D	conversão analógico - digital
• D/A	conversão digital - analógico

Projeto de Programação

Estruturação de um Projeto de Programação

Funcionalmente, um projeto de programação, pode ser visto como uma coleção de módulos utilizados para realizar uma tarefa específica, também conhecido como programa aplicativo. Isto permite uma visão hierárquica do projeto com a criação de sub-rotinas e funções.

Os módulos são chamados para a execução pelo software executivo (sistema operacional do CP) ou por outros módulos, através de instruções apropriadas. Quando armazenado em disco, o projeto de programação corresponde a um conjunto de arquivos, onde cada arquivo contém um módulo, denominados como mostra a figura a seguir:

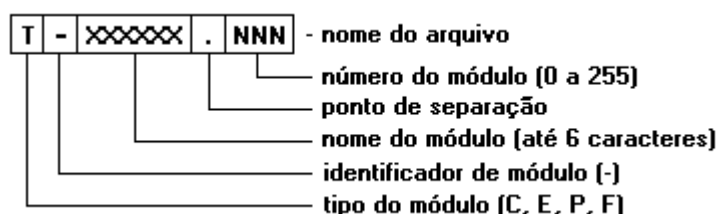


Figura 2-22 Formato do Nome dos Módulos em Arquivo

Exemplo: **F-PID.033**

Em alguns locais deste manual e na Ajuda os módulos de programa são referenciados somente pelo seu tipo e número, quando não for relevante o nome utilizado no mesmo.

Exemplo: **E018**

ATENÇÃO:

O nome do arquivo correspondente a um módulo de programa não deve ser alterado através de outro aplicativo do Windows™. Para alterar o nome de um arquivo, deve-se ler e salvar o mesmo com o nome desejado através do MasterTool. Ver seção **Salvando um Módulo com Outro Nome** no capítulo 5 do Manual de Utilização do MasterTool.

Se o nome do arquivo for modificado através de outro aplicativo do Windows™, poderá ser atribuído um nome inválido para o mesmo, não podendo mais ser lido para o MasterTool ou carregado no CP.

Existem 4 tipos de módulos que podem fazer parte de um projeto de programação:

- **Módulo C** (Configuração): existe um módulo de configuração por projeto, contendo os parâmetros de configuração do CP (C000).
 - **Módulo C Estendido** (Configuração): este módulo de configuração existe quando o usuário utiliza em seu projeto uma determinada característica da UCP que necessita de um módulo de configuração estendido. Para maiores informações consultar o manual de utilização do MasterTool Programming (C003 a C009).
- **Módulo E** (Execução): podem existir até 4 módulos de execução por projeto. Os mesmos são chamados somente pelo sistema operacional do CP (E000, E001, E018 e E020).
- **Módulo P** (Procedimento): podem existir até 112 módulos procedimento por projeto. Eles contêm trechos de programa aplicativo, sendo chamados por instruções colocadas em módulos de execução, procedimento ou função. Após serem executados, o processamento retorna para a instrução seguinte à de chamada. Os módulos P funcionam como sub-rotinas, não permitindo a passagem de parâmetros para o módulo chamado (P000 a P111).

- **Módulo F (Função):** podem existir até 112 módulos função por projeto. Eles contêm trechos de programa aplicativo escritos de forma genérica, permitindo a passagem de parâmetros para o módulo chamado, de forma a poderem ser reaproveitados em vários programas aplicativos diferentes. São semelhantes a instruções, podendo ser chamados por módulos de execução, procedimento ou função. (F000 a F111).

Módulo C - Configuração

O módulo C contém os parâmetros de configuração do CP. Sua criação é pré-requisito para a edição dos demais módulos do projeto de programação no MasterTool. A definição dos parâmetros contidos no mesmo é realizada através da janela de edição de módulo C. Para maiores detalhes sobre como configurar um módulo C, ver seção **Configurando o Módulo C** no capítulo 5 do Manual de Utilização do MasterTool.

Há somente um módulo C por projeto de programação, tendo como nome o próprio nome do projeto e o número 000.

Conteúdo de um módulo C:

- **Declaração do Barramento de módulos de E/S:** especifica a configuração dos módulos de E/S a serem utilizados no controlador programável, indicando a distribuição dos mesmos e módulos especiais no barramento do CP. A declaração dos módulos define, desta forma, o número de pontos e os endereços de E/S a serem utilizados no programa aplicativo. A mesma é realizada através da janela de edição do módulo C. Para maiores informações sobre como configurar o barramento, ver o item **Configurando o Barramento** na seção **Configurando o Módulo C** no capítulo 5 do Manual de Utilização do MasterTool.
- **Declaração de Operandos:** especifica o número de operandos simples e tabelas de operandos que serão utilizados no projeto de programação, dentro de cada tipo disponível. Permite também a definição da retentividade dos operandos, ou seja, quais operandos devem manter seu conteúdo com a falta de energia do sistema.
 - **Declaração de Operandos Simples:** permite a definição da quantidade de operandos Memória (%M), Inteiro (%I), Decimal (%D) e Real (%F). É realizada através da janela de edição de módulo C. Para maiores informações sobre como declarar operandos simples, ver o item **Configurando Operandos Simples** na seção **Configurando o Módulo C** no Manual de Utilização do MasterTool.
 - **Declaração de Operandos Tabela:** permite a definição do número de tabelas de operandos Memória (%TM), de operandos Inteiro (%TI), de operandos Decimal (%TD), de operandos Real (%TF) e do número de posições de cada uma. Uma tabela representa um conjunto de operandos, sendo a sua definição realizada através da janela de edição de módulo C. Para maiores informações sobre como configurar operandos tabela, ver o item **Configurando Operandos Tabela** na seção **Configurando o Módulo C** no Manual de Utilização do MasterTool.
 - **Declaração de Operandos Retentivos:** especifica o número de operandos simples que são retentivos, dentro dos operandos já declarados. Operandos retentivos são aqueles que continuam com o seu conteúdo inalterado com a falta de energia do CP, sendo os não retentivos zerados com a reinicialização do sistema. Os operandos tabela são todos retentivos. A declaração é realizada através da janela de edição de módulo C. Para maiores informações sobre como configurar operandos retentivos, ver o item **Configurando Operandos Retentivos** na seção **Configurando Operandos Retentivos** no capítulo 5 do Manual de Utilização do MasterTool.
- **Declaração dos Parâmetros Gerais da UCP:** são parâmetros genéricos necessários para o funcionamento do controlador programável, tais como o tipo de UCP na qual o programa aplicativo será carregado, o período de chamada dos módulos acionados por interrupção e o tempo máximo de ciclo de varredura. Estes parâmetros são declarados através da janela de edição de módulo C. Para maiores informações sobre como configurar os parâmetros gerais, ver seção **Configurando o Módulo C** no capítulo 5 do Manual de Utilização do MasterTool.

- **Declaração dos Parâmetros da Rede ALNET I:** especifica os diversos parâmetros necessários ao funcionamento da comunicação em rede ALNET I. Estes parâmetros são configurados na janela de edição de módulo C. Para maiores informações sobre como configurar parâmetros da rede ALNET I, ver item **Configurando Parâmetros da Rede ALNET I** na seção **Configurando o Módulo C** no capítulo 5 do Manual de Utilização do MasterTool.
- **Declaração dos Parâmetros da Rede ALNET II:** especifica os diversos parâmetros necessários ao funcionamento da comunicação em rede ALNET II, para os controladores programáveis que permitem o seu uso. Estes parâmetros são configurados na janela de edição de módulo C. Para maiores informações sobre como configurar parâmetros da rede ALNET II, ver item **Configurando Parâmetros da Rede ALNET II** na seção **Configurando o Módulo C** no capítulo 5 do Manual de Utilização do MasterTool
- **Declaração dos Parâmetros da Rede Ethernet:** especifica os diversos parâmetros necessários ao funcionamento da comunicação em rede Ethernet, para os controladores programáveis que permitem o seu uso. Estes parâmetros são configurados na janela de edição de módulo C. Para maiores informações sobre como configurar parâmetros da rede Ethernet, ver item **Configurando Parâmetros da Rede Ethernet** na seção **Configurando o Módulo C** no capítulo 5 do Manual de Utilização do MasterTool
- **Declaração dos Parâmetros da Rede de Sincronismo:** especifica os diversos parâmetros necessários ao funcionamento da comunicação com rede de sincronismo, para os controladores programáveis que permitem o seu uso. Estes parâmetros são configurados na janela de edição de módulo C. Para maiores informações sobre como configurar parâmetros da rede de sincronismo, ver item **Configurando Parâmetros da Rede de Sincronismo** na seção **Configurando o Módulo C** no capítulo 5 do Manual de Utilização do MasterTool

Módulo C Estendido – Configuração

Estes módulos contém configurações de determinadas características das UCPs. Estes módulos são totalmente gerenciados pelo usuário, isto é, deve ser criado e apagado conforme necessidade do usuário. Isto se deve ao fato de que a quantidade deste tipo de módulo varia de acordo com cada aplicação, podendo não ter nenhum a ter até 7 módulos (C003 a C009).

Para maiores informações consultar Manual de Utilização do MasterTool.

Módulo E - Execução

Os módulos E contém trechos do programa aplicativo, sendo chamados para a execução pelo software executivo. Existem diversos módulos E, diferenciando-se entre si pelo modo como são chamados à execução, conforme o seu número.

Tipos de módulos E:

- **E000 - Módulo de Inicialização:** é executado uma única vez, ao se energizar o CP ou na passagem de modo programação para execução com o MasterTool, antes da execução cíclica do módulo E001.
- **E001 - Módulo Sequencial de Programa Aplicativo:** contém o trecho principal do programa aplicativo, sendo executado ciclicamente.
- **E018 - Módulo Acionado por Interrupção de Tempo:** o trecho de programa aplicativo colocado neste módulo é chamado para a execução em intervalos de tempo periódicos. Define-se o período de chamada do mesmo nos parâmetros gerais do módulo C, podendo ser escolhido entre 50 ms, 25 ms, 10 ms, 5 ms, 3,125 ms, 2,5 ms, 1,25 ms e 0,625 ms. Ao ser transcorrido o tempo programado, a execução sequencial do programa aplicativo é interrompida e o módulo E018 é executado. Após o seu final, o sistema retorna a execução para o ponto do processamento sequencial onde o módulo E0001 havia sido interrompido. O tempo continua a ser contado durante a chamada do módulo E018, devendo a sua execução ser o mais breve possível para não haver o aumento excessivo no tempo de ciclo do módulo E001.

ATENÇÃO:

O tempo de execução do módulo E018 não pode ser maior ou igual ao período de chamada. Caso isto aconteça, o CP entra em modo erro sendo exibida a mensagem **Reentrada no módulo E018**, na janela **Informações** (comando **Comunicação, Estado, Informações**).

- **E020 - Módulo Acionado pela Entrada de Interrupção:** o trecho de programa aplicativo colocado neste módulo é executado com o acionamento da entrada de interrupção dos CPs AL-600/4, AL-600/8, AL-600/16, QK600, PL102 ou PL103. Quando ocorrer uma transição de subida no sinal presente nesta entrada, a execução sequencial do programa aplicativo é interrompida e o módulo E020 é executado. Após o seu final, o sistema retorna a execução para o ponto do processamento sequencial onde o módulo E0001 havia sido interrompido. Se a entrada for acionada com muita frequência, o tempo de execução do módulo E020 deve ser o mais breve possível, para não haver o aumento excessivo no tempo de ciclo do módulo E001.

ATENÇÃO:

O tempo de execução do módulo E020 não pode ser maior ou igual ao período de chamada. Caso isto aconteça, o CP entra em modo erro sendo exibida a mensagem **Reentrada no módulo E020**, na janela **Informações** (comando **Comunicação, Estado, Informações**).

O módulo E020 atua somente nos CPs AL-600/4, AL-600/8, AL-600/16, a partir da versão 1.20 do software executivo, bem como nos CPs QK600, PL102 e PL103. Somente estes CPs possuem a entrada rápida de interrupção que aciona o E020.

©DICA:

Caso o modelo de CP definido no módulo C permita o uso de determinado tipo de módulo mas o MasterTool não esteja habilitando sua criação, pode-se utilizar o módulo de execução genérico, definindo-se seu número de acordo com a necessidade (E-.018 ou E-.020).

Módulo P - Procedimento

Os módulos P contêm trechos de programas aplicativos chamados a partir de módulos E, P ou F através da instrução CHP (Chama Procedimento).

Este tipo de módulo não possui passagem de parâmetros, sendo similar ao conceito de sub-rotina.

O número máximo de módulos deste tipo é 112 (P000 a P111).

O módulo P é útil para conter trechos de programas aplicativos que devem ser repetidos várias vezes no programa principal, sendo assim programados uma só vez e chamados quando necessário, economizando memória de programa.

Podem ser usados também para uma melhor estruturação do programa principal, dividindo-o em segmentos de acordo com a sua função e declarando-os em diversos módulos P. Neste caso, o módulo de execução contínua E001 somente chama os módulos P na sequência desejada.

Exemplos:

- P-MECAN.000 - realiza o intertravamento mecânico da máquina
- P-TEMPER.001 - realiza o controle de temperaturas
- P-VIDEO.002 - realiza o interfaceamento homem-máquina
- P-IMPRES.003 - gerência a impressão de relatórios

Módulo F - Função

Os módulos F contêm trechos de programas aplicativos chamados a partir de módulos E, P ou F, através da instrução CHF (Chama Função).

Na chamada dos módulos F é possível a passagem de valores como parâmetros para o módulo chamado. Estes módulos são usualmente escritos de forma genérica para serem aproveitados por vários programas aplicativos, em linguagem de relés ou de máquina, sendo semelhantes às instruções da linguagem de relés. Os valores dos parâmetros são enviados e devolvidos através de listas de operandos existentes na instrução de chamada e no módulo F.

Na edição de uma instrução CHF, devem ser definidas 2 listas de operandos que são utilizadas para:

- enviar parâmetros para execução do módulo função (Entrada)
- receber os valores retornados pelo módulo função (Saída)

Na edição do módulo função, também devem ser definidas 2 listas de operandos, utilizando o comando **Edição, Editar, Parâmetros**, que são utilizados para:

- receber parâmetros da instrução CHF (Entrada)
- enviar valores de retorno para a instrução CHF (Saída)

A passagem de parâmetros é realizada através da cópia dos valores dos operandos declarados (passagem de parâmetros por valor). A figura a seguir apresenta o fluxo de dados entre a instrução CHF e o módulo função:

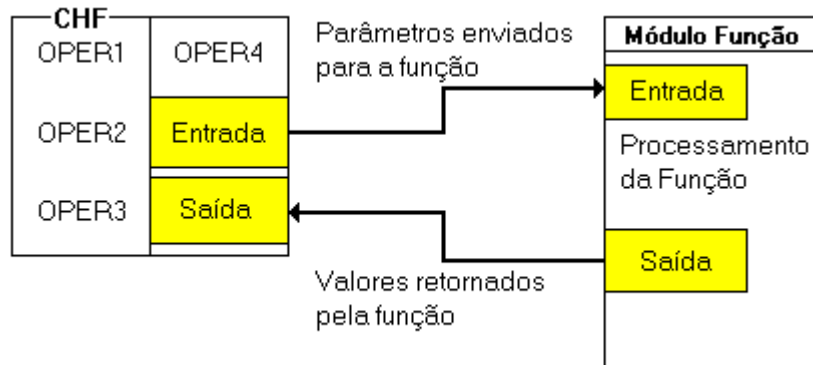


Figura 2-23 Passagem de Parâmetros para o Módulo F

Maiores informações a respeito da passagem de parâmetros podem ser encontradas na descrição da instrução CHF neste mesmo manual. É permitida a passagem de todos os tipos de operandos.

Exemplos:

- F-LINEAR.002 - executa a linearização de valores lidos de um sensor
- F-PID.033 - realiza cálculos para implementação de laço PID de controle

Estados de Operação do CP

Existem cinco estados ou modos de operação do CP: inicialização, execução, programação, ciclado e erro. O estado em que o controlador programável se encontra é indicado nos LEDs do painel frontal da UCP, podendo também ser consultado pelo MasterTool, através da caixa de diálogo **Estado** (opções **Comunicação, Estado**, a partir do menu principal). Para obter informações específicas sobre os modos de operação, consultar o manual de utilização do controlador utilizado.

- **Estado Inicialização:** o CP inicializa as diversas estruturas de dados de uso do programa executivo e realiza consistências no projeto de programação presente na memória. Este estado ocorre após a energização do controlador, passando após alguns segundos para o estado execução. Caso não exista programa aplicativo na memória, o CP passa para modo erro.

Enquanto o CP está inicializando, pode-se acionar o comando **Comunicação, Estado, Programação**, ou atalho equivalente na barra de ferramentas, fazendo com que o CP passe diretamente para o estado de programação, ao invés de executar o programa aplicativo. Este procedimento é útil para a reinicialização de CPs com programas contendo erros graves de programação.

Por exemplo, um módulo com um laço infinito de execução, programado com uma instrução de salto para uma lógica anterior, provoca o acionamento do circuito de cão-de-guarda da UCP sempre que for ligada, após o estado de inicialização. Executando-se o procedimento anterior logo após a energização, o CP passa para o estado programação após inicializar, permitindo o apagamento ou a substituição do programa.

- **Estado Execução:** normalmente o controlador programável se encontra neste estado, varrendo continuamente os pontos de entrada e atualizando os pontos de saída de acordo com a lógica programada. Este estado indica que o CP está executando corretamente um programa aplicativo.
- **Estado Programação:** o programa aplicativo não é executado, não havendo a leitura dos pontos de entrada, sendo as saídas desativadas e a memória do CP é compactada. O CP permanece inoperante, esperando comandos do MasterTool. Este modo normalmente é utilizado para a carga dos módulos do projeto de programação pelo MasterTool, através do canal serial. Ao passar para estado execução ou ciclado a partir do estado programação, os operandos são zerados.
- **Estado Ciclado:** quando em modo ciclado, o controlador programável não executa ciclicamente o módulo E001, permanecendo à espera de comandos do MasterTool. Cada comando **executa ciclo** acionado no MasterTool (opções **Comunicação, Estado, Executa Ciclo** a partir do menu principal ou atalho equivalente) dispara uma única varredura do programa aplicativo (módulo E001), permanecendo o CP à espera de um novo comando após a execução da mesma. Quando o CP passa para modo ciclado, a contagem de tempo nos temporizadores pára, sendo os mesmos incrementados de uma unidade de tempo a cada duas varreduras executadas. As chamadas para o módulo de interrupção de tempo E018 não são realizadas neste modo. O módulo E020, acionado pela entrada de interrupção externa, continua sendo chamado neste modo.
- **Estado de Erro:** indica que houve alguma anomalia no CP durante o processamento do projeto de programação. O tipo de erro ocorrido pode ser consultado através da caixa de diálogo (opções **Comunicação, Estado, Informações** a partir do menu principal), enquanto o CP estiver neste estado. A saída do estado de erro somente é possível passando-se o controlador programável para modo programação.

Em condições normais, o controlador programável pode estar nos modos execução, programação e ciclado, sendo esses modos acionados através de comandos do MasterTool (opções **Execução, Programação e Ciclado** da caixa de diálogo **Estado**, ou seus atalhos equivalentes na **Barra de Ferramentas**). Na ocorrência de alguma situação de funcionamento errôneo nestes modos, o CP passa para estado de erro. A recuperação do modo erro somente é possível passando-se o controlador programável para modo programação. A figura a seguir apresenta as possibilidades de troca de estados:

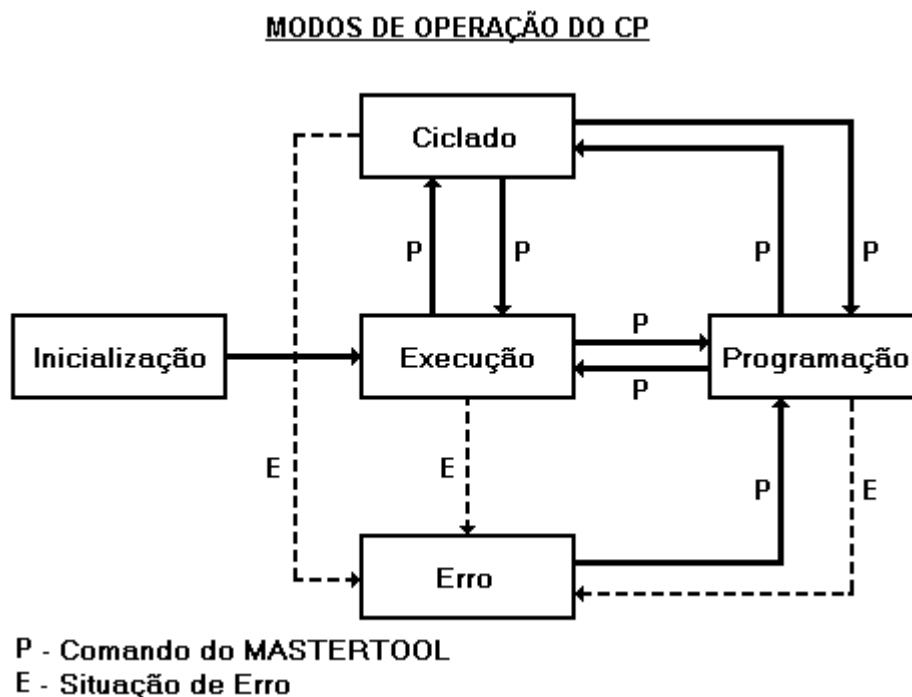


Figura 2-24 Estados de Operação do CP

Nos modos execução, programação e ciclado é possível carregar e ler módulos do projeto de programação pelo canal serial do controlador programável, bem como monitorar e forçar quaisquer operandos utilizados. Essas operações não são possíveis caso o CP esteja em modo erro.

Os operandos que não são retentivos são zerados na passagem de modo programação para execução ou programação para ciclado, permanecendo os demais inalterados.

Execução do Projeto de Programação

Quando o CP é energizado ou após a passagem para modo execução, as inicializações do sistema são realizadas de acordo com o conteúdo do módulo C, sendo logo após executado o módulo E000 uma única vez. O controlador programável passa então para o processamento cíclico do módulo E001, atualizando as entradas e saídas e chamando o módulo E018, quando existente, a cada período de tempo de interrupção programado. Nos CPs AL-600/4, AL-600/8, AL-600/16, QK600, PL102 e PL103, o módulo E020 é chamado, quando existente, com o acionamento da entrada de interrupção. A figura 2-22 mostra esquematicamente a execução do programa aplicativo.

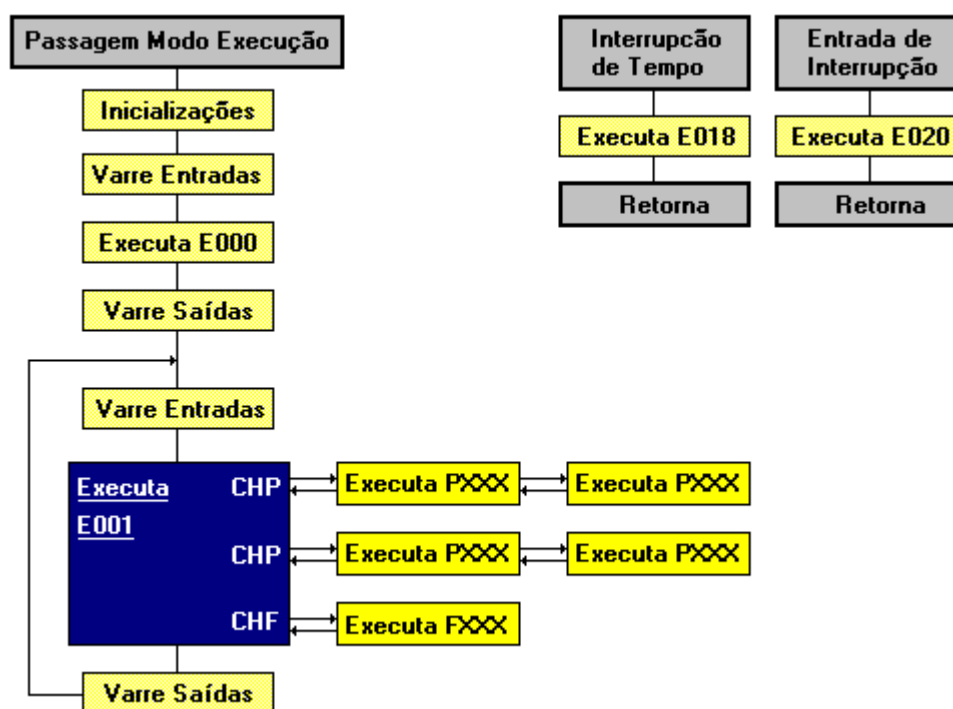


Figura 2-25 Execução do Projeto de Programação

Elaboração de Projetos de Programação

Considerações Gerais

Um projeto de programação é composto ao menos por um módulo C (configuração) e um módulo E001 (execução). A condição mínima para a execução de um projeto de programação é a presença destes dois módulos na UCP do controlador programável.

O primeiro passo para a edição de um projeto de programação no MasterTool é a criação ou a leitura de um projeto. O módulo de configuração do projeto é criado automaticamente na criação de um novo projeto, uma vez que neste módulo estão contidas as declarações dos módulos de entrada e saída e operandos utilizados em todo o projeto. Cada módulo que contenha trechos de programa aplicativo (E, P ou F) necessita que o módulo C esteja presente no MasterTool para que possa ser editado.

Após a criação ou leitura de um projeto, pode-se editar o mesmo adicionando módulos já existentes, criando módulos novos para o projeto ou excluindo módulos que já façam parte do projeto.

O MasterTool permite que vários módulos sejam carregados e permaneçam simultaneamente em sua memória

Considerações sobre Operandos

Os diversos módulos que compõem um projeto de programação devem preferencialmente ter sido programados utilizando-se o mesmo módulo C. Caso um módulo já programado necessite ser usado em outro projeto de programação, os operandos utilizados pelo módulo devem obrigatoriamente estar declarados no módulo C do novo projeto.

Os operandos disponíveis no controlador programável são de uso comum a todos os módulos do projeto de programação presentes no CP (operandos globais). Em consequência deste fato, dois módulos quaisquer podem estar inadvertidamente acessando o mesmo operando, ocorrendo erros no funcionamento de ambos.

Ao elaborar um projeto de programação, deve-se reservar operandos em número suficiente para o mesmo, preferencialmente separando-os em grupos, cada grupo utilizado somente por um módulo. Os operandos utilizados nos módulos F programados em linguagem de relés e blocos também podem ser acessados por quaisquer outros módulos de programa presentes no CP, mesmo os operandos utilizados na passagem de parâmetros. Para garantir o seu caráter genérico, cada módulo F deve utilizar um grupo de operandos diferente dos demais utilizados no programa aplicativo.

Utilização dos Módulo P e F

Dentro de um módulo do projeto de programação podem ser colocadas as instruções de chamada de outros módulos. As instruções CHP e CHF chamam, respectivamente, módulos de procedimento e função. Elas realizam o gerenciamento das chamadas dos módulos, verificando a existência ou não dos mesmos no diretório do controlador programável, baseadas em seus tipos e números.

Nas UCPs AL-600, AL-600/4, AL-600/8, AL-600/16, AL-3003, AL-3004, AL-2000/MSP, AL-2002/MSP, AL-2003, AL-2004, QK600, QK800, QK801, QK2000/MSP, PL101, PL102 e PL103 existem 18 níveis de chamada, desde os módulos E (nível mais alto). No CP AL-2003 e AL-2004 existem 32 níveis de chamada. Ou seja, podem ser executadas até 18 (ou 32, no caso do AL-2003 ou AL-2004) chamadas consecutivas de módulos sem ser finalizada a execução de nenhum. Deve-se considerar que o módulo E018 (se existir) e os módulos por ele chamados também ocupam níveis de chamada.

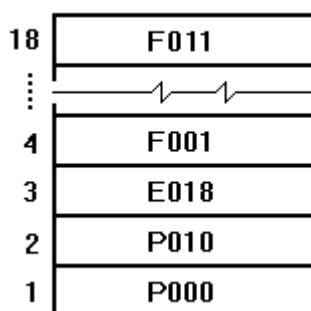


Figura 2-26 Número Máximo de Níveis de Chamada de Módulos

Quando o número máximo de chamadas acumuladas sem retorno for ultrapassado, o sistema não mais as realizará, prosseguindo com a execução normal do programa aplicativo. Nos casos de ocorrência de chamadas para módulos inexistentes ou o excesso do número de chamadas totais, são mostradas mensagens de advertência na janela **Informações** (opções **Comunicação, Estado, Informações** a partir do menu principal), pois estas situações poderão causar erros no processamento conforme a lógica programada.

É possível a chamada de um módulo por ele mesmo (programação por recursividade) tomando-se os cuidados necessários, ou seja, deve ser previsto no trecho de programa aplicativo com recursividade um momento em que não há mais chamadas para o mesmo módulo. Embora seja possível, o uso de tal procedimento não é aconselhável em controladores programáveis, devido ao grande tempo de processamento que um pequeno trecho de programa aplicativo pode necessitar para ser executado e à facilidade da ocorrência de laços infinitos de execução (loops).

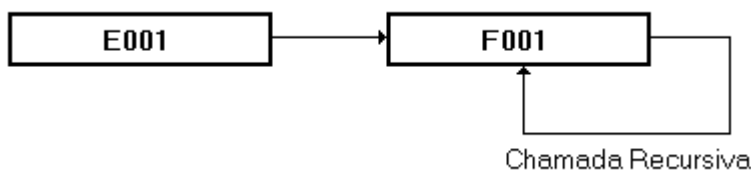


Figura 2-27 Chamada Recursiva de Módulos

Deve-se evitar a programação indevida de laços de chamada de módulos sem fim ("dead locks"). Caso um módulo do projeto de programação chame outro e este também realize uma chamada para o primeiro, se as instruções de chamada nos dois módulos não forem desabilitadas ambos permanecerão chamando-se mutuamente até a passagem do controlador programável para modo erro, por excesso de tempo de execução do programa aplicativo.

A mesma situação pode ocorrer com chamadas encadeadas entre diversos módulos, quando um módulo chamado volte a chamar algum módulo inicial da cadeia. Por exemplo, se o módulo P005 chamar o P002, este chamar P007 e este chamar novamente o P005, o processamento poderá permanecer neste laço se nenhuma instrução de chamada for desabilitada.

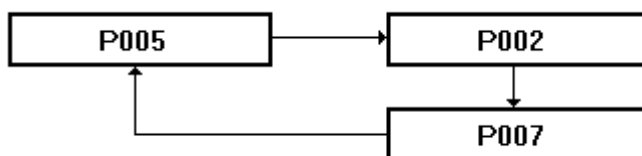


Figura 2-28 Laço de Chamada de Módulos

Utilização do Módulo E018

O módulo E018 deve ser utilizado quando for necessário um processamento rápido para alguns pontos de entrada e saída do controlador programável, como por exemplo, no sensoreamento de fins-de-curso em sistemas de posicionamento rápido. Para este caso deve ser empregada a instrução de atualização de pontos de E/S (AES), realizando dentro do módulo E018 um processamento semelhante a um laço completo de execução do programa principal. As entradas são lidas, o trecho de programa aplicativo desejado é executado e as saídas são atualizadas.

Da mesma forma, este módulo torna-se útil quando se deseja uma resposta dos acionamentos de saída após um tempo fixo dos estímulos das entradas, independente do tempo de varredura do programa principal, que pode ser variável. Essa característica é importante também em sistemas de controle de posição.

Outra aplicação para o módulo E018 é a geração de bases de tempo menores que 100 ms para o programa principal. Podem ser gerados temporizadores com precisão de 50 ms, 10 ms ou menos, se necessário, através do uso de instruções contadoras dentro do módulo de interrupção de tempo.

Este módulo é útil quando se deseja um controle preciso de tempos no processamento do CP.

Utilização do Módulo E020

A entrada rápida de interrupção dos CPs das séries AL-600 e PICCOLO pode ser usada para um processamento imediato de um ponto de entrada, sendo útil para o controle rápido de posicionamentos. Com o seu acionamento, o módulo E020 é chamado para a execução, realizando o processamento necessário e a atualização de pontos de saída através da instrução AES.

O módulo E020 também pode ser utilizado no acionamento de dispositivos ou procedimentos de segurança, dispositivos de frenagem ou outras aplicações que necessitem rapidez de atuação.

A entrada de interrupção também é utilizada como a segunda entrada de contagem rápida nos CPs das séries AL-600 e PICCOLO e na UCP QK600, não sendo necessário nenhum ajuste no equipamento para selecionar sua função. Caso o módulo E020 esteja presente no CP, este é chamado a cada acionamento da entrada. Se o programa aplicativo chamar o módulo F-CONT.005, este realiza a leitura e escrita do valor de contagem, incrementado a cada acionamento da entrada. Se desejado, pode-se utilizar esta entrada com ambas as funções, com o módulo F-CONT.005 contando o número de vezes que o módulo E020 foi acionado.

O módulo E020 atua somente nos CPs AL-600/4, AL-600/8 e AL-600/16, a partir da versão 1.20 do software executivo, e nos CPs QK600, PL102 e PL103. Somente estes CPs possuem a entrada rápida de interrupção que aciona o E020.

Cuidados na Utilização do Módulo E018

Alguns cuidados especiais são necessários na programação do módulo E018. Como o mesmo é chamado de modo síncrono a cada período fixo de tempo, interrompendo o processamento do módulo E001, o seu tempo de execução deve ser o mais breve possível para não aumentar excessivamente o tempo de ciclo total do programa aplicativo.

Se o intervalo entre as chamadas do módulo E018 for programado para 25 ms, por exemplo, e o seu tempo de execução for 20 ms, restarão somente 5 ms para a execução do programa principal antes que o E018 seja chamado novamente. Esta situação aumenta de forma considerável o tempo de ciclo do módulo E001.

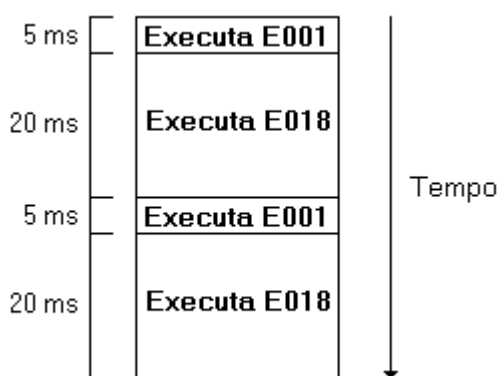


Figura 2-29 Cuidados na Utilização do Módulo E018

Caso a execução do módulo E018 demore mais do que o intervalo de tempo programado para suas chamadas, o CP passa para o estado de erro, sendo exibida a mensagem "Reentrada no módulo E018" na janela **Informações** (opções **Comunicação, Estado, Informações** a partir do menu principal). Nesta situação, deve-se aumentar o período de chamada utilizado ou diminuir o tempo de execução do módulo E018 para que o projeto de programação possa ser executado corretamente.

As instruções permanecem com o mesmo comportamento quando executadas dentro do módulo E018, exceto em relação a algumas características particulares. Os temporizadores (TEE e TED) continuam a contar o tempo a cada 100 ms, qualquer que seja o período de acionamento programado para o módulo, exatamente como na execução cíclica. O relé de pulso (PLS) aciona a sua saída durante uma execução do módulo E018, zerando a mesma na próxima chamada. As instruções CHP e CHF podem ser usadas da mesma forma como no programa principal, devendo os módulos acionados pelas mesmas obedecerem às mesmas regras de programação válidas para o módulo E018 propriamente dito. O número máximo de níveis de chamada de módulos utilizados dentro do módulo E018 deve ser acrescentado ao máximo nível empregado em E001, devendo esta soma ser menor que o limite do sistema (18 níveis).

Cuidados na Programação do Módulo E020

Alguns cuidados especiais são necessários na programação do módulo E020. O seu tempo de processamento deve ser breve, principalmente se a entrada de interrupção for acionada de forma freqüente, para não aumentar de forma excessiva o tempo de ciclo total do programa aplicativo.

Se a entrada de interrupção for acionada de forma periódica a cada 30 ms, por exemplo, e o tempo de execução do E020 for 25 ms, só restarão 5 ms para a execução do programa principal antes que o

módulo seja chamado novamente. Esta situação aumenta de forma considerável o tempo de ciclo do módulo E001.

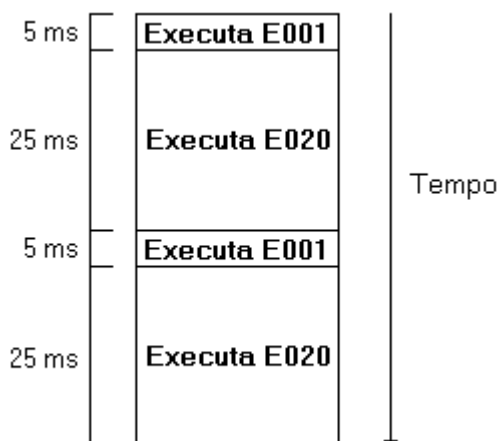


Figura 2-30 Cuidados na Utilização do Módulo E020

Caso o módulo E020 esteja sendo executado e ocorra novo acionamento na entrada de interrupção do CP, este acionamento é desconsiderado, continuando normalmente a execução do módulo. Esta situação não provoca a mudança para modo erro, permanecendo o CP em execução normal. Portanto, o CP ignora acionamentos da entrada rápida de interrupção que ocorram em tempos menores que o tempo de execução do E020.

As instruções continuam a ter o mesmo comportamento quando executadas dentro do módulo E020, exceto em relação a algumas características particulares.

A chamada do módulo depende do processo que está sendo controlado, não ocorrendo de forma periódica. Esta característica inviabiliza o uso dos temporizadores no E020, ou seja, as instruções TEE e TED não devem ser utilizadas no mesmo. O relé de pulso (PLS) aciona a sua saída durante uma execução do módulo E020, zerando a mesma na próxima chamada. As instruções CHP e CHF podem ser usadas da mesma forma como no programa principal, devendo os módulos acionados pelas mesmas obedecerem às mesmas regras de programação válidas para o módulo E020 propriamente dito. O número máximo de níveis de chamada de módulos utilizados dentro do módulo E020 deve ser acrescentado ao máximo nível empregado em E001 e E018, devendo esta soma ser menor que o limite do sistema (18 níveis).

Utilização dos Operandos na Programação dos Módulos E018 e E020

Outro cuidado necessário diz respeito ao compartilhamento de dados entre os módulos E018 ou E020 e os demais presentes no controlador programável. As interrupções podem ocorrer em qualquer ponto do programa principal de execução cíclica (módulo E001 ou módulos P ou F chamados pelo mesmo), inclusive durante o processamento das suas instruções. Como os operandos são todos de uso comum a qualquer módulo do projeto de programação, deve-se tomar o cuidado para não utilizar inadvertidamente nos módulos E018 ou E020 algum operando que seja utilizado em outro módulo do projeto de programação, pois este uso, conforme o caso, pode ocasionar o funcionamento incorreto. Quando o módulo E018 e E020 são utilizados simultaneamente, ambos devem empregar operandos exclusivos.

Para possibilitar o compartilhamento de dados entre os módulos E018, E020 e outro módulo qualquer de execução cíclica devem ser utilizadas as instruções MOV (movimentação de operandos simples) e MOB (movimentação de blocos de operandos), para criar uma imagem dos operandos que contém os dados a serem compartilhados. Estas instruções devem ser utilizadas nos módulos pertencentes ao ciclo normal de execução e não nos módulos E018 ou E020.

Por exemplo, se for necessário que o módulo E018 utilize o valor contido em uma memória usada no programa principal, deve-se passar o valor desta memória para outra através da instrução MOV, devendo o módulo E018 utilizar somente esta última. A instrução MOV deve estar no programa principal, e não no módulo E018.

O fluxo contrário de dados também exige a criação de operandos imagem. Se o módulo E020 manipula uma tabela e o programa principal precisa utilizar os valores desta tabela, esses valores devem ser copiados para uma segunda tabela de uso exclusivo do programa principal, através de uma instrução MOB. A instrução MOB deve estar no programa principal, e não no módulo E020.

Uma situação semelhante ocorre para as instruções bobinas. Se algum ponto de um operando é modificado no programa principal por uma bobina, não é permitida a alteração de qualquer ponto pertencente a todo octeto do mesmo operando nos módulos E018 ou E020. Esta restrição não existe quando os octetos utilizados pertencem à faixa %S0000 a %S0015.

Entretanto, é possível que pontos de um operando sejam alterados nos módulos E018 ou E020 por uma bobina e sejam somente testados por outro módulo com instruções contatos, por exemplo. A situação contrária também é permitida, ou seja, os pontos de um operando modificados no programa principal por bobinas podem ser testados nos módulos E018 ou E020 por contatos.

Outro cuidado a ser tomado diz respeito à atualização das entradas e saídas dentro dos módulos E018 ou E020.

Preferencialmente devem ser atualizadas dentro destes módulos somente as entradas utilizadas no seu processamento, utilizando-se a instrução AES. Como o programa aplicativo de execução cíclica pode ser interrompido em qualquer local por estes módulos, se neles forem atualizadas as imagens das entradas do programa principal, estas poderão assumir valores diversos em pontos diferentes do programa aplicativo durante o mesmo ciclo de execução. Este fato pode provocar erros se um operando de entrada for utilizado em vários lugares do programa principal, pois normalmente é suposto que seu valor permaneça inalterado na mesma varredura.

Devido a este fato, é aconselhável o uso de octetos de entrada exclusivos para os módulos E018 ou E020, se for necessária a sua atualização dentro do mesmo, não sendo estes octetos utilizados no programa principal.

Caso seja necessária a atualização de entradas utilizadas simultaneamente nas interrupções e no processamento cíclico, o valor das mesmas pode ser copiado para operandos auxiliares ou memórias no início do programa principal, sendo estes operandos utilizados no restante do mesmo. Pode-se também não atualizar as imagens das entradas nos módulos E018 ou E020 com a instrução AES, mas somente ler diretamente os valores dos módulos de E/S para operandos memória através da instrução MES, e utilizar estas memórias em contatos para realizar o processamento nos módulos de interrupção.

A atualização de octetos de saída nos módulos E018 ou E020 (através da instrução AES) é possível, desde que os pontos pertencentes a estes octetos sejam acionados por bobinas somente dentro destes módulos.

Nos módulos E018 e E020, não se deve escrever valores com a instrução MES em módulos de saída declarados no barramento através do MasterTool, pois a varredura das saídas também realiza atualização de valores nos mesmos.

Quando um módulo E018 ou E020 está sendo executado e a compactação for disparada, os mesmos poderão ser transferidos para outra posição na memória pela rotina de compactação. Durante esta transferência a sua chamada será desabilitada, podendo algumas interrupções ocorrerem sem que os

módulos E018 ou E020 sejam processados. Deve-se atentar para este efeito da compactação sobre a execução do módulo acionado por interrupção. Durante a compactação dos demais módulos, todavia, os módulos E018 ou E020 continuarão sendo executados.

Utilização Simultânea dos Módulos E018 e E020

É possível a utilização simultânea do módulo E018 (acionado periodicamente pela interrupção de tempo) com o módulo E020 (acionado pela entrada rápida de interrupção) nos CPs AL-600/4, AL-600/8, AL-600/16, QK600, PL102 e PL103.

Não existem prioridades de execução quanto às interrupções dos dois módulos. Ou seja, se estiver sendo executado o módulo E020 e ocorrer a próxima interrupção de tempo, o processamento do E020 é interrompido, é executado o módulo E018, retornando após para a execução interrompida do E020.

Da mesma forma, se estiver sendo executado o módulo E018 e for acionada a entrada de interrupção, o processamento do E018 é interrompido, é executado o módulo E020, retornando após para a execução interrompida do E018.

Alguns cuidados devem ser observados na utilização simultânea dos dois módulos. Deve-se somar o tempo de execução do módulo E018 ao E020, devendo este total ser menor do que o período de chamada programado para o E018, evitando-se o erro de reentrada de execução no mesmo.

Deve-se também somar o número máximo de níveis de chamada dos módulos E001, E018 e E020, devendo o resultado ser menor ou igual ao número máximo permitido (18 níveis).

Os dois módulos devem utilizar operandos exclusivos, obedecendo às regras da seção **Utilização dos Operandos na Programação dos Módulos E018 e E020**, neste mesmo capítulo.

Depuração de Projetos de Programação

Várias facilidades estão previstas no controlador programável para auxiliar a depuração dos projetos de programação, sendo descritas a seguir.

Informações do Estado do CP

Diversas informações sobre o estado do controlador podem ser obtidas com o acionamento das opções **Comunicação, Estado, Informações** no MasterTool.

Atalho:



- **Modelo da UCP** - indica o tipo do controlador com o qual o MasterTool está comunicando.
- **Versão do Executivo** - mostra o número da versão do programa executivo que o CP contém.
- **Modo de Operação** - indica o modo de operação atual do CP: execução, programação, ciclado ou erro.
- **Mensagem de Erro/Advertência** - se o CP estiver em modo erro, é apresentada uma mensagem indicando a causa do erro ocorrido. Caso o CP esteja em qualquer outro modo, a mensagem indica a existência de problemas que não causam a mudança para modo erro (por exemplo, a bateria do CP descarregada). Ver **Mensagens de Erro**, apêndice A do Manual de Utilização do MasterTool.
- **Saídas** - indica se as saídas estão habilitadas ou desabilitadas.
- **Relés Forçados** - indica se existe algum ponto de entrada ou saída forçado.
- **Troca de Módulos com CP Energizado** - indica a possibilidade de troca de módulos com CP energizado.
- **Compactando RAM** - indica se o CP está compactando a memória RAM do programa aplicativo.
- **Copiando Módulo** - indica se algum módulo está sendo carregado no CP, transferido da RAM para a flash EPROM ou da flash EPROM para a RAM, ou se o CP está apagando a memória flash.

- **Nível de Proteção** - mostra o nível de proteção atual do CP.
- **Tempos de Ciclo** - mostra o valor instantâneo, médio, máximo e mínimo do tempo de ciclo do programa aplicativo. Ver seção **Tempos de Ciclo de Execução do Programa** neste mesmo capítulo.
- **Período de Acionamento de E018** - mostra o período de chamada do módulo acionado por interrupção de tempo E018, se estiver presente no CP.

As janelas de estado do CP (opções Comunicação, Estado, Informações), diretório de módulos (opções **Comunicação, Módulos**) e monitoração (opções Comunicação, Monitorar Operandos ou Monitorar Bloco de Operandos ou Monitorar Tabelas) fornecem diversas informações úteis para a verificação do bom funcionamento do controlador. Estas informações podem ser obtidas à distância, caso o CP esteja ligado em rede. Sempre que o MasterTool for conectado a algum CP, aconselha-se a consulta dessas informações como o primeiro procedimento a ser tomado.

Monitoração

Através do MasterTool é possível monitorar os valores de um ou mais operandos do CP em qualquer modo de operação, exceto em modo erro.

Os valores dos operandos contidos em uma lógica de programa aplicativo também podem ser visualizados diretamente sobre a mesma facilitando a verificação de seu funcionamento.

Para maiores informações sobre como realizar a monitoração, ver itens **Monitorando Operandos Simples, Monitorando Operandos Tabelas e Monitorando Programas** na seção **Comunicando com o CP ou Roteador** no capítulo 5 do Manual de Utilização do MasterTool.

A monitoração de operandos no CP ocorre no final do ciclo de execução do programa aplicativo. Devido a este fato, situações incoerentes podem ser visualizadas na monitoração de lógicas, se os valores dos operandos forem modificados nas lógicas posteriores à monitorada.

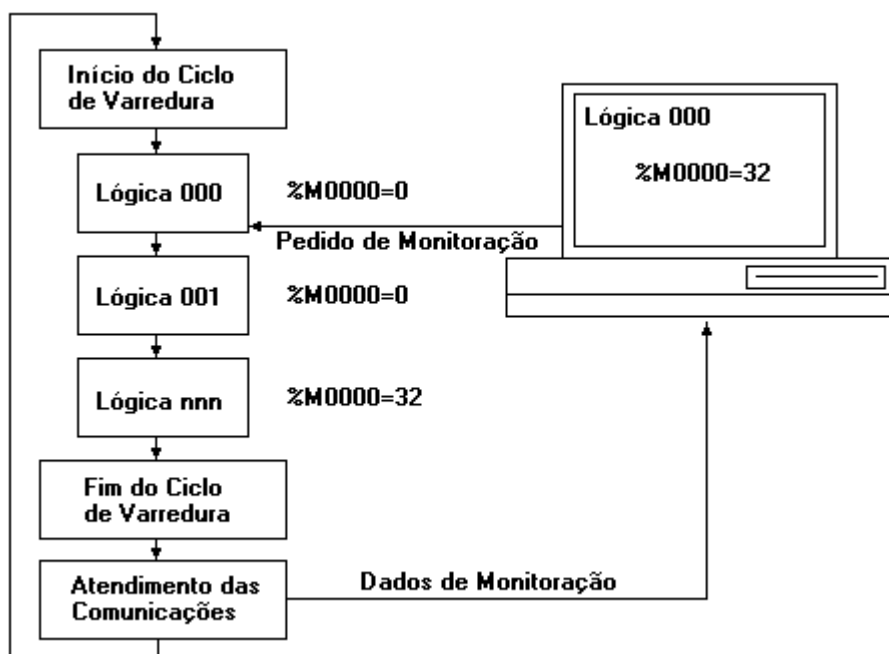


Figura 2-31 Situação Incoerente na Monitoração de Lógicas

Forçamento

Os valores dos operandos também podem ser forçados com o MasterTool, ou seja, pode-se modificar o conteúdo de qualquer operando do projeto de programação. O forçamento de operandos é

permitido em qualquer modo de operação, exceto em modo erro. Ver itens **Forçando Operandos Simples** e **Forçando Operandos Tabela** na seção **Comunicando com o CP ou Roteador** no capítulo 5 do Manual de Utilização do MasterTool.

Os operandos %A, %M, %I, %D, %F, %TM, %TI, %TD e %TF têm o seu valor alterado somente por uma varredura, logo após o envio do comando ao CP. Para que o valor forçado permaneça no operando, não pode haver no programa nenhuma instrução que o modifique.

O forçamento dos operandos %E e %S é realizado de forma permanente no controlador. Após o envio do comando ao CP, o valor é forçado em todas as varreduras do programa aplicativo, até que o operando seja liberado. O LED FC no painel da UCP permanece ligado se houver algum operando %E ou %S forçado.

Os valores forçados em operandos %E sobrepõem os obtidos na leitura dos módulos de entrada, antes do início de cada ciclo de execução do programa aplicativo. O programa é executado com o valor forçado, como se o ponto de entrada correspondente estivesse com este valor, podendo ser visualizado na monitoração.

Por exemplo, caso o operando %E0002.5 esteja forçado com o valor 1, o programa aplicativo será executado com este valor para este operando, não importando o estado do ponto no módulo de entrada correspondente. A monitoração de %E0002.5 mostra sempre o valor 1.

Os valores forçados nos operandos %S são diretamente enviados para os módulos de saída, independente do valor obtido após a execução do programa aplicativo. A monitoração mostra o valor forçado, que corresponde ao valor assumido pelo ponto correspondente ao operando no módulo de saída.

Por exemplo, caso o operando %S0024.3 esteja forçado com o valor 0, o ponto respectivo no módulo de saída permanecerá desligado, não importando o estado da bobina que contenha o operando no programa aplicativo. A monitoração de %S0024.3 mostra sempre o valor 0.

ATENÇÃO:

Podem ser visualizadas situações incoerentes na monitoração de lógicas com operandos %S forçados. Isto ocorre porque o valor monitorado pode ser diferente do valor realmente obtido pelo programa aplicativo.

ATENÇÃO:

Todos os forçamentos de operandos %E e %S são removidos com a desenergização do CP. O forçamento destes operandos deve ser utilizado de forma temporária, somente para auxiliar a depuração do projeto de programação. Não devem ser deixados operandos %E ou %S forçados em caráter permanente, pois serão liberados com a desenergização e posterior energização do controlador.

Os operandos %E e %S deixam de ser forçados pelo CP através do comando de liberação de forçamento. A liberação consiste em anular o forçamento anteriormente determinado. Os operandos %E voltam a ter seus valores atualizados de acordo com os módulos de entrada, enquanto que os módulos de saída recebem os valores obtidos no processamento do programa aplicativo.

ATENÇÃO:

A operação de forçamento não atua sobre operandos %E ou %S atualizados com a instrução AES. Esta instrução executa a leitura para operandos %E ou a escrita de operandos %S no momento em que é executada, não considerando os efeitos de forçamento sobre os mesmos. Por este motivo, recomenda-se que não sejam forçados os operandos atualizados pela instrução AES que estejam ativas no programa.

Para maiores informações sobre como liberar operandos forçados, ver item **Liberando Operandos Forçados** na seção **Comunicando com o CP ou Roteador** no capítulo 5 do Manual de Utilização do MasterTool.

Desabilitação das Saídas

Para a segurança na posta-em-marcha quando se utiliza o programa aplicativo diretamente na máquina, os acionamentos das saídas do controlador programável podem ser desabilitados através do comando desabilita. O programa aplicativo continua a ser executado no CP, com a varredura das entradas e cálculo dos valores das saídas, porém com todos os pontos de saída mantidos desacionados. Os operandos %S podem ser monitorados e conferidos com os valores esperados para os mesmos.

Para maiores informações sobre como desabilitar as saídas, ver item **Habilitando e Desabilitando as Saídas** na seção **Comunicando com o CP ou Roteador** no capítulo 5 do Manual de Utilização do MasterTool.

ATENÇÃO:

Se o CP for desenergizado, a desabilitação dos pontos de saída é removida. Ou seja, quando o CP for novamente energizado, o estado dos operandos da memória será normalmente transferido, ao final de cada varredura, para os pontos de saída. A desabilitação deve ser utilizada de forma temporária, somente para auxiliar a depuração do projeto de programação.

Modificações no Programa

A carga de módulos durante a execução do projeto de programação (carga "on line") possibilita sucessivas modificações e envios do módulo em depuração para o controlador programável. Deste modo não é necessária a reinicialização do programa aplicativo de controle nem a troca do estado do controlador programável a cada alteração efetuada em um módulo.

ATENÇÃO:

Após qualquer modificação realizada no módulo C do projeto de programação, o mesmo deve ser enviado para o CP.

ATENÇÃO:

Se a declaração de operandos simples ou tabelas for modificada, aconselha-se reinicializar o CP, passando para modo programação, carregando o módulo C e retornando para modo execução. Podem ocorrer erros no funcionamento alterando-se a configuração de operandos e enviando o módulo C com o controlador em modo execução.

Após um certo número de cargas sucessivas em modo execução, entretanto, pode se tornar necessária a compactação da memória RAM de programa pelos motivos explicados na seção **Gerenciamento de Módulos do Projeto de Programação no CP**, neste capítulo. Este tipo de carga somente é possível se houver memória livre suficiente no CP para armazenamento do módulo a ser enviado.

Ao acabar a depuração de um módulo de programa, sugere-se a transferência do mesmo para a memória flash EPROM ou a sua gravação no cartucho de EPROM, liberando o espaço disponível na memória RAM de programa.

Modo Ciclado

A execução do projeto de programação em modo ciclado torna-se útil na verificação do funcionamento de intertravamentos rápidos no programa aplicativo. As demais facilidades de depuração continuam atuando da mesma forma como no modo execução (monitoração, forçamento, carga e outras operações com módulos).

Em modo ciclado, os valores dos operandos permanecem constantes entre os ciclos, exceto os pontos de entrada (%E) que continuam sendo continuamente atualizados, mostrando seus valores reais.

Gerenciamento de Módulos do Projeto de Programação

Os módulos que compõem o programa aplicativo são independentes entre si, não necessitando de ligação ("link") através de programas auxiliares. A carga dos módulos no controlador programável pelo canal serial pode ser realizada em qualquer ordem, permitindo que somente o módulo alterado seja carregado no CP, em caso de manutenção de projetos de programação.

ATENÇÃO:

Para a UCP do sistema, somente o tipo do módulo e seu número são relevantes, sendo o seu nome desconsiderado. Se dois módulos com tipo e número iguais mas com nomes diferentes forem carregados no CP, somente o último carregado será considerado.

O controlador programável organiza um diretório interno onde são armazenadas diversas informações a respeito dos módulos contidos no mesmo, podendo ser consultado pelo MasterTool através do comando diretório de módulos (opções **Comunicação, Módulos** a partir do menu principal). Quando este comando for acionado, uma caixa de diálogo é aberta, mostrando na sua parte superior, dois quadros chamados **Módulos em RAM** e **Módulos em EPROM** com a lista dos nomes e a ocupação de memória de cada módulo presente no CP.

No quadro **Memória Ocupada** é informado o número total de módulos e o espaço de memória total ocupado pelos mesmos (soma de todas as ocupações individuais), além do espaço total ocupado em RAM ou EPROM.

No quadro **Memória Livre** estão expostas as quantidades de memória RAM e EPROM disponível para a carga de novos módulos, em cada banco de memória existente no controlador programável.

Somente os módulos presentes no diretório são considerados válidos para a execução no CP.

Um módulo de programa presente no diretório do CP pode estar somente em um tipo de memória, RAM ou EPROM, nunca simultaneamente em ambas. Os módulos carregados pelo canal serial são sempre armazenados na memória RAM de programa aplicativo.

Compactação

A memória de programa do controlador programável está dividida em um ou mais bancos, dependendo do modelo da UCP utilizada (ver tabela 2-1 na seção **Organização de Memória dos CPs** neste capítulo).

À medida que os módulos que compõem o projeto de programação são enviados para o CP através do canal serial, os mesmos ocupam o primeiro banco de memória RAM, desde o seu início até o seu final. Quando o espaço restante no primeiro banco não for suficiente para carregar o próximo módulo, este será carregado no banco seguinte, se este existir.

A cada carga de novo módulo no controlador programável, o software executivo testa se há espaço suficiente para o mesmo desde o primeiro até o último banco disponível. A carga de um novo módulo somente é possível se houver memória livre à disposição para o seu armazenamento.

Dentro de um banco de memória RAM, a carga de um módulo é realizada sempre a partir da primeira posição após o último módulo presente. Se um módulo no início do banco for removido, os módulos que estão após o mesmo devem ser transferidos para ocupar o seu espaço de memória, para que este espaço esteja disponível no final do banco para a carga de outros módulos. Este procedimento denomina-se **compactação da memória RAM** do programa aplicativo.

Exemplo:

Suponha-se que o primeiro banco de memória do controlador programável esteja inicialmente com os seguintes módulos:

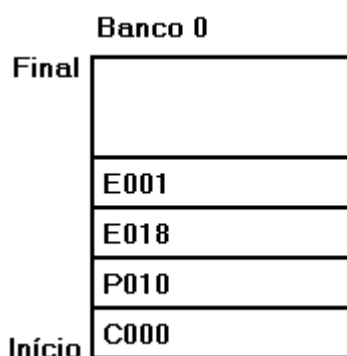


Figura 2-32 Compactação de Memória RAM

Se o módulo P010 for removido do CP o banco 0 passará a ter a seguinte organização:

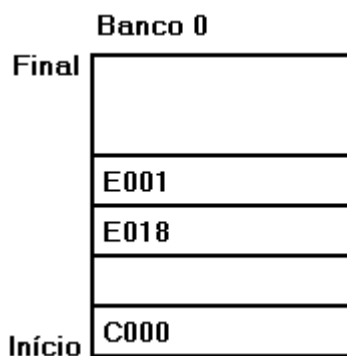


Figura 2-33 Compactação da Memória RAM - 2

O espaço anteriormente ocupado por P010 não é aproveitável pelo controlador programável, pois a carga de um novo módulo somente é possível após o último, o módulo E001. Após realizar a compactação de memória do CP, o banco 0 passa para a seguinte configuração:

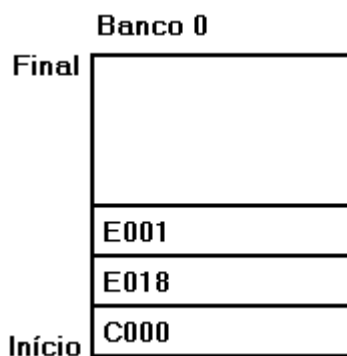


Figura 2-34 Compactação da Memória RAM - 3

Os módulos E018 e E001 são transferidos para o espaço anteriormente ocupado pelo módulo P010, tornando este espaço disponível ao final da memória do banco para a carga de outro módulo.

Se o controlador programável estiver em modo programação ou ciclado, os bancos de memória RAM de programa são mantidos automaticamente compactados pelo programa executivo. Em modo execução, todavia, deve-se disparar a compactação manualmente, através do MasterTool (opções **Comunicação, Módulos, Compactar RAM** desde o menu principal). Este procedimento é comum quando são realizadas diversas cargas de módulos em modo execução (cargas "on line"), tipicamente quando um módulo está sendo depurado, necessitando de sucessivas alterações e transmissões para o CP.

ATENÇÃO:

Dependendo da localização dos módulos na memória, o procedimento de compactação pode aumentar em muito o tempo de alguns ciclos do programa aplicativo, quando realizado em modo execução. Deve-se estar consciente dos efeitos deste aumento de tempo de processamento. Aconselha-se que a compactação não seja disparada caso a máquina sob controle esteja em operação ou com seus acionamentos principais ativos.

Devido a este mecanismo de gerenciamento de módulos no controlador programável, é possível que a soma da memória disponível nos bancos do CP com o valor ocupado pelos módulos seja menor que a memória de programa total da UCP, se esta estiver em modo execução. Este fato significa que a memória de programa não está compactada. Após a compactação, entretanto, a soma dos valores ocupados com a memória livre deve ser igual à memória total.

No MasterTool não existe a Compactação de Flash. O método para se "compactar" a Flash é carregar os módulos para a RAM, limpar a Flash e recarregar os módulos para a Flash.

Utilização da Memória EPROM ou Flash EPROM

Os controladores podem conter dois tipos diferentes de memória para gravação permanente do programa aplicativo:

- **Memória EPROM** - é apresentada na forma de cartuchos, conectados no painel frontal do CP. O projeto de programação é gravado nos cartuchos através do gravador AL-2860, ligado ao MasterTool, e removido com apagadores apropriados, utilizando luz ultravioleta. Utilizada nos controladores AL-3003 e AL-3004.
- **Memória flash EPROM** - é colocada na placa do circuito do CP, não sendo necessário removê-la para gravar ou apagar programas. Estas operações são realizadas pelo próprio controlador, através de comandos do MasterTool. Utilizada nos controladores das séries AL-600, AL-2000, PL104, PL105 e QUARK.

Ambos os tipos possuem características peculiares: podem ser gravadas parcialmente, porém não permitem o apagamento parcial do seu conteúdo. Ou seja, somente é possível desgravar todo o conteúdo da memória no seu apagamento.

Cada CP somente utiliza um dos dois tipos citados anteriormente. Nenhum CP possui ambos os tipos de memória EPROM.

A configuração de memória de cada modelo de CP é apresentada na seção **Organização de Memória dos CPs** neste capítulo.

Neste manual, na Ajuda e no programador MasterTool, o nome EPROM refere-se indistintamente à memória utilizada para gravação permanente do projeto de programação no CP, seja do tipo cartucho de EPROM ou flash EPROM.

Transferência de módulos de RAM para Flash:

Após serem carregados na memória RAM de programa, através do canal serial do CP, os módulos do projeto de programação podem ser transferidos para a flash EPROM. Este comando somente é utilizável nos CPs que possuam memória flash. Para maiores informações sobre como transferir módulos de RAM para flash EPROM, ver item **Transferindo Módulos de RAM para Flash EPROM** na seção **Comunicando com o CP ou Roteador** no capítulo 5 do Manual de Utilização do MasterTool.

Pode-se transferir um único módulo ou um conjunto de módulos, mesmo com o CP executando o programa. A transferência em modo execução é realizada parcialmente em cada varredura, podendo demorar vários segundos até ser completada, principalmente se houver alto tempo de ciclo de execução. No final da transferência, o módulo em RAM é automaticamente apagado e as informações do diretório modificadas.

O gerenciamento da carga do módulo na flash EPROM é idêntico ao da memória RAM, mostrada na seção anterior **Compactação**. Ou seja, o módulo da RAM é gravado no primeiro banco de flash que possua espaço livre suficiente para o conter, após o último módulo do banco. A pesquisa do espaço livre ocorre na ordem sequencial dos bancos (0, 1, 2 e 3).

A memória EPROM do PL101, PL102 ou PL103 possibilita a realização de "backup" da memória RAM. Somente é possível transferir todos os módulos de RAM para EPROM ou todos os módulos de EPROM para RAM. Não é possível visualizar-se o conteúdo da memória EPROM.

Caso um CP PL101, PL102 ou PL103 seja energizado sem que existam módulos de programa na memória RAM, todos os módulos de programa existentes na memória EPROM são transferidos para a RAM.

Nas UCPs PL104 e PL105 a memória Flash EPROM também é utilizada para se gravar o programa executivo.

Transferência de módulos de EPROM para RAM:

Os módulos presentes na memória flash EPROM ou no cartucho de EPROM também podem ser transferidos para a memória RAM de programa. Para maiores informações sobre como transferir módulos de EPROM para RAM, ver item **Transferindo Módulos de EPROM para RAM** na seção **Comunicando com o CP ou Roteador** no capítulo 5 do Manual de Utilização do MasterTool.

Pode-se transferir um único módulo ou um conjunto de módulos, mesmo com o CP executando o programa. A transferência em modo execução é realizada parcialmente em cada varredura, podendo

demorar vários segundos até ser completada, principalmente se houver alto tempo de ciclo de execução. No final da transferência, as informações do diretório são modificadas.

O gerenciamento da carga do módulo na flash EPROM é idêntico ao da memória RAM, mostrada na seção anterior **Compactação**.

Apagamento e reabilitação de módulos em EPROM:

O comando de apagamento pode ser utilizado para módulos armazenados na memória EPROM do CP. Como o apagamento de EPROMs somente é possível para todo o seu conteúdo, este comando apenas retira as informações do diretório de módulos, não realizando um apagamento real da memória.

O mesmo ocorre se um módulo gravado em EPROM for substituído por um novo módulo de mesmo tipo e número carregado pelo canal serial. O novo módulo é armazenado em RAM, permanecendo o antigo em EPROM, sendo mostrado no diretório apenas o novo em RAM.

O módulo removido através do comando de apagamento ou substituído com carga de um novo módulo pode ser recuperado para o diretório, pois o seu conteúdo ainda está gravado na memória EPROM. Esta recuperação é possível com o comando de reabilitação de módulos.

A reabilitação faz com que um módulo inexistente no diretório reapareça situado em EPROM, ou que um já existente em RAM seja substituído por um antigo em EPROM.

Para maiores informações sobre como apagar ou reabilitar módulos, ver itens **Apagando Módulos do CP ou Roteador** e **Reabilitando Módulos em EPROM** na seção **Comunicando com o CP ou Roteador** no capítulo 5 do Manual de Utilização do MasterTool.

Apagamento da memória EPROM:

Com o apagamento total da memória EPROM, todos os módulos são removidos, ficando todo o seu espaço disponível para a gravação de novos módulos.

Para apagar o cartucho de EPROM deve ser utilizado um dispositivo apagador apropriado, após a remoção do cartucho do CP.

Para apagar a memória flash EPROM, utiliza-se as opções **Comunicação, Módulos, Apaga Flash** estando o CP em modo programação. O apagamento pode demorar vários segundos, dependendo da capacidade da memória flash utilizada no CP. Para maiores informações sobre como apagar a memória flash, ver item **Apagando a Memória Flash EPROM** na seção **Comunicando com o CP ou Roteador** no capítulo 5 do Manual de Utilização do MasterTool.

Tempos de Ciclo de Execução do Programa

O tempo máximo possível para a execução de um ciclo completo do programa aplicativo no controlador programável é configurável de 100 ms a 800 ms. Ou seja, a execução completa de uma varredura do módulo E001 não pode se estender por mais do que o valor configurado, incluindo as chamadas para módulos P e F e acionamentos do módulo de interrupção de tempo E018. O software executivo realiza uma verificação contínua no tempo de ciclo, passando automaticamente para estado de erro caso este limite seja ultrapassado.

Pode-se verificar os tempos de execução do programa aplicativo através da janela de informações do CP (opções **Comunicação, Estado, Informações** a partir do menu principal), sendo informados diversos tempos de ciclos de execução, especificados a seguir:

- **Tempo de ciclo instantâneo:** indica o tempo de ciclo da última varredura executada pelo CP antes de enviar as informações do seu estado para o MasterTool. Este item é útil em modo ciclado, quando mostra o tempo de execução do último ciclo disparado no controlador programável.
- **Tempo de ciclo médio:** indica a média de tempos de execução das últimas 256 varreduras realizadas pelo CP. Em modo execução este parâmetro fornece uma idéia geral do tempo de processamento do programa

aplicativo, ao contrário do tempo de ciclo instantâneo, que pode estar mostrando um valor atípico de uma varredura isoladamente. Como este tempo é calculado somente a cada 256 varreduras, por vezes o seu valor necessita de alguns segundos para ser atualizado, principalmente em caso de aumento brusco no tempo de execução (inclusão de novos módulos no controlador programável, por exemplo).

- **Tempo de ciclo máximo:** indica o maior tempo entre todos os ciclos realizados desde a passagem do CP para modo execução ou ciclado.
- **Tempo de ciclo mínimo:** indica o menor tempo entre todos os ciclos realizados desde a passagem do CP para modo execução ou ciclado.

Os tempos de ciclo são indicados em milisegundos (ms), sendo as contagens inicializadas na passagem de modo programação para execução ou programação para ciclado.

O atendimento da comunicação serial com o MasterTool aumenta o tempo de ciclo do programa aplicativo no CP, podendo, em alguns casos, ultrapassar o tempo de ciclo máximo selecionado. Caso o tempo de execução limite for ultrapassado somente devido aos comandos da comunicação serial (monitoração, forçamento e demais), o CP não entrará em estado de erro. É possível, portanto, a indicação de um tempo de ciclo máximo maior que o selecionado sem que o controlador programável tenha entrado em modo erro.

O procedimento de compactação da memória de programa do controlador programável também segue a regra anterior. Em alguns casos, a rotina de compactação necessita copiar um módulo muito extenso na memória entre dois ciclos do programa aplicativo, aumentando exageradamente o tempo de execução de uma varredura. Nesta situação o CP não entrará em estado de erro.

Deve-se tomar cuidados especiais quando os tempos de ciclo de execução se aproximam do tempo máximo selecionado. O simples fato do programa aplicativo estar executando corretamente com as condições mais comuns dos pontos de entrada não garante que seu tempo de varredura, em condições reais de funcionamento da máquina, permaneça dentro do valor limite.

ATENÇÃO:

Cada projeto de programação deve ser examinado cuidadosamente em busca de situações que irão provocar os maiores tempos de execução. Essas situações devem ser simuladas e os tempos medidos, verificando se não são excessivos. Este procedimento deve ser realizado mesmo em projeto de programação com tempos de ciclo bem abaixo do limite, para assegurar o seu bom funcionamento.

É possível que em algumas varreduras isoladas o tempo de ciclo exceda o tempo máximo selecionado sem que o CP passe para modo erro, caso estas varreduras esporádicas não causem atrasos nos temporizadores do sistema.

ATENÇÃO:

Se o CP indicar o tempo de ciclo máximo maior que o selecionado sem que tenha havido uma compactação de memória, mesmo que continue normalmente em modo execução, deve-se examinar o programa para diminuir o seu tempo de ciclo nas situações que causem maiores tempos.

☺DICA:

Existem alguns procedimentos típicos para diminuir o tempo de execução de programas aplicativos muito extensos. Um bom gerenciamento na chamada de módulos pode diminuir sensivelmente o tempo de ciclo total, sendo realizadas chamadas de poucos módulos do programa aplicativo em cada varredura, não permitindo que todos sejam disparados em um mesmo ciclo. O uso de instruções de salto dentro dos módulos, diminui o tempo de execução dos mesmos, pois um trecho de programa aplicativo saltado é desconsiderado pelo software executivo. As instruções relé mestre e fim de relé mestre (RM e FRM) não possuem esta propriedade, pois o segmento de programa aplicativo delimitado pelas mesmas continua a ser executado mesmo quando a bobina RM o desabilita.

☺DICA:

Deve-se realizar inicializações de valores em operandos ou tabelas dentro do módulo E000, idealizado especialmente para este propósito. A execução do módulo E000, por não ser cíclica, pode demorar mais que o tempo máximo, sendo este tempo desconsiderado na contagem do tempo da primeira varredura do módulo E001. Pelo modo como é executado, torna-se sem sentido a programação de temporizadores (TEE, TED) no módulo E000.

Níveis de Proteção do CP

As UCPs das séries AL-600, AL-2000, QUARK e Piccolo possuem um mecanismo de proteção do projeto de programação e dos operandos, permitindo o bloqueio da carga de módulos de programa, forçamentos de valores ou mesmo leituras de módulos e monitoração por operadores não autorizados.

Estas características são interessantes em processos críticos, para evitar modificações acidentais nos dados ou no programa de controle ou na necessidade de sigilo dos mesmos.

O bloqueio de operações é realizado através de níveis de proteção, que podem ser definidos apenas por operadores que conheçam uma senha preestabelecida. O controlador pode operar em quatro diferentes níveis de proteção:

- **Nível 0** - todas as operações no CP são permitidas.
- **Nível 1** - não é possível alterar o projeto de programação (apagar ou carregar novos módulos de programa) ou mudar o estado do CP. Pode-se forçar e monitorar operandos e ler módulos do programa.
- **Nível 2** – Possui as mesmas restrições do nível 1 e também não é possível ler módulos do programa.
- **Nível 3** – Possui as mesmas restrições dos níveis 1 e 2 e também não é possível monitorar ou forçar operandos nem mudar o estado do CP. Também não será permitido consultar o diretório de módulos nem liberar forçamento de operandos.

A troca do nível de proteção é realizada com as opções **Comunicação, Estado, Proteção** no MasterTool, devendo-se digitar a senha de acesso correta para efetivá-la. O nível de proteção do CP pode ser consultado com o MasterTool através das opções **Comunicação, Estado, Informações**.

A utilização de níveis de proteção diferentes de zero permite que somente pessoas autorizadas, que conheçam a senha, modifiquem o programa ou os dados do CP. Operadores não autorizados, mesmo dispondo do MasterTool, ficam impedidos de realizar alterações inadvertidas.

A senha de acesso pode ter de um a oito caracteres alfanuméricos. É definida ou trocada com as opções **Comunicação, Estado, Senha**, devendo-se digitar a senha anterior e a nova senha duas vezes, para ser confirmada a mudança.

O CP é fornecido sem senha. Não é necessário digitar qualquer valor no campo senha anterior para a definição da primeira senha.

ATENÇÃO:

A senha deve ser escrita e guardada em lugar seguro. Em caso de perda da senha programada no CP, entrar em contato com a ALTUS.

A proteção do CP atua não somente para as operações realizadas com o MasterTool, mas também para os comandos recebidos pelas redes ALNET I e ALNET II, com as mesmas características definidas para cada nível.

Para maiores informações sobre como alterar o nível de proteção e a senha do CP, ver itens **Alterando o Nível de Proteção** e **Alterando a Senha** na seção **Comunicando com o CP ou Roteador** no capítulo 5 do Manual de Utilização do MasterTool.

Intertravamento de Comandos no CP

Na série AL-2000 e na UCP QK2000/MSP da série QUARK pode-se utilizar as redes de comunicação ALNET I e ALNET II em conjunto. Quando interligado desta forma, é possível a recepção simultânea de dois comandos cuja execução concorrente seja indesejável, devido às suas características. Por exemplo, o CP pode receber um comando de transferência de módulo da EPROM para a RAM pela ALNET II enquanto o mesmo módulo está sendo carregado na ALNET I.

Situações semelhantes ocorrem com os comandos de transferência de módulos de programa da memória EPROM para RAM, de RAM para flash ou de apagamento da memória flash. A execução destes comandos pode estender-se por vários segundos, durante os quais o CP pode receber outros comandos que entrem em conflito com a operação em curso. Por exemplo, o CP pode receber um comando para apagar a memória flash enquanto um módulo estava sendo transferido para a mesma memória.

Para resolver as possíveis situações de conflito, há um mecanismo de intertravamento para a execução de alguns comandos disponíveis no CP. Estes comandos não podem ser executados caso o CP esteja realizando uma operação específica. Existem dois sinais internos, **carregando módulo** (CM) e **compactando RAM** (CR), que são usados para este propósito. As tabelas 2-9 e 2-10 mostram os comandos que utilizam o intertravamento e o acionamento dos sinais.

O estado dos sinais carregando módulo e compactando RAM pode ser verificado na janela de informações do CP, opções **Comunicação, Estado, Informações** no MasterTool. Enquanto qualquer um dos sinais estiver acionado, o LED FC do painel do CP permanece aceso.

Operação realizada no CP	Comando bloqueado (ALNET I, ALNET II)	Sinal ligado
Carga de Módulos	Carga de módulos Transferência de EPROM para RAM Transferência de RAM para Flash Pedido de carga de módulos Reabilitação de módulos em EPROM Apagamento de Flash EPROM Compactação	CM
Transferência de EPROM para RAM	Carga de módulos Transferência de EPROM para RAM Transferência de RAM para Flash Pedido de carga de módulos Reabilitação de módulos em EPROM Apagamento de Flash EPROM Compactação	CM
Transferência de RAM para Flash	Carga de módulos Transferência de EPROM para RAM Transferência de RAM para Flash Pedido de carga de módulos Reabilitação de módulos em EPROM Apagamento de Flash EPROM Compactação	CM
Apagamento de Flash EPROM	Carga de módulos Transferência de EPROM para RAM Transferência de RAM para Flash Pedido de carga de módulos Reabilitação de módulos em EPROM Apagamento de Flash EPROM Compactação	CM
Legenda: CM - Carregando Módulo		

Tabela 2-6 Intertravamento de Comandos no CP (carregando módulo)

Operação realizada no CP	Comando bloqueado (ALNET I, ALNET II)	Sinal ligado
Compactação	Carga de módulos Transferência de EPROM para RAM Transferência de RAM para Flash Pedido de carga de módulos Reabilitação de módulos em EPROM Remoção de módulos Compactação	CR
Legenda: CR - Compactando RAM		

Tabela 2-7 Intertravamento de Comandos no CP (Compactando RAM)

Por exemplo, enquanto um módulo está sendo carregado no CP pela rede ALNET I ou ALNET II, os comandos de carga de módulos, transferência de EPROM para RAM, transferência de RAM para flash, pedido de carga de módulos, reabilitação de módulos em EPROM, apagamento de Flash EPROM e compactação não podem ser executados, caso sejam recebidos através da outra rede. Se forem recebidos pelo CP, uma resposta indicando a impossibilidade de sua execução é transmitida para o solicitante.

ATENÇÃO:

O intertravamento dos comandos não ocorre para os CPs AL-3003 e AL-3004. O LED FC do painel frontal não é ligado na ocorrência das operações das tabelas 2-8 e 2-9.

Projeto de Roteador

Estruturação de um Projeto de Roteador

Um projeto de roteador constitui-se de um único módulo de configuração de redes (módulo R) que possui os parâmetros necessários à configuração do roteamento da rede.

O módulo de configuração de rede é chamado para a execução pelo software executivo (sistema operacional do roteador). Quando armazenado em disco, o projeto de programação corresponde a um arquivo, denominado como mostra a figura 2-32.

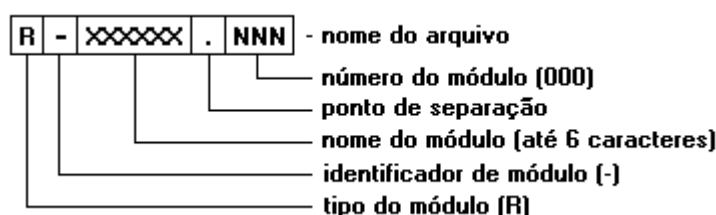


Figura 2-35 Formato do Nome do Módulo R em arquivo

Exemplo: **R-GAT1.000**

Em alguns locais deste manual e na Ajuda os módulos de programa são referenciados somente pelo seu tipo e número, quando não for relevante o nome utilizado no mesmo.

Exemplo: **R000**

ATENÇÃO:

O nome do arquivo correspondente a um módulo de programa não deve ser alterado através de outro aplicativo do Windows™. Para alterar o nome de um arquivo, deve-se ler e salvar o mesmo com o nome desejado através do MasterTool. Ver seção **Salvando um Módulo com Outro Nome** no capítulo 5 do Manual de Utilização do MasterTool.

Se o nome do arquivo for modificado através de outro aplicativo do Windows™, poderá ser atribuído um nome inválido para o mesmo, não podendo mais ser lido para o MasterTool ou carregado no CP.

Módulo R - Configuração de Roteador

O módulo R contém os parâmetros de configuração de roteador. A definição dos parâmetros contidos no mesmo é realizada através da janela de edição de módulo R, ver seção **Configurando o Módulo R**, no capítulo 5 do Manual de Utilização do MasterTool.

Há somente um módulo R por projeto de roteador, tendo como nome o próprio nome do projeto, para uma melhor identificação do mesmo.

Conteúdo de um módulo R:

- **Declaração do Modelo de UCP:** especifica o modelo de UCP na qual o projeto de roteador deve ser executado. A edição deste parâmetro é realizada na janela de edição do módulo R. Para maiores informações, ver o item **Configurando o Modelo de UCP**, na seção **Configurando o Módulo R** no capítulo 5 do Manual de Utilização do MasterTool.
- **Declaração dos Parâmetros do Canal:** especifica os parâmetros de configuração relativo a cada canal do roteador. Para maiores informações sobre a função de cada canal do roteador, ver o manual do roteador

utilizado. Os parâmetros do canal são declarados na janela de edição de módulo R. Para maiores informações sobre como declarar os parâmetros, ver o item **Configurando os Parâmetros do Canal**, na seção **Configurando o Módulo R** no capítulo 5 do Manual de Utilização do MasterTool.

- **Declaração do Roteamento do Canal:** especifica a tabela de roteamento do canal, ou seja, para qual sub-rede deve ser repassado o comando para que chegue à sub-rede destino. Esta tabela é declarada na janela de edição do módulo R. Para maiores informações sobre como declarar a tabela de roteamento, ver o item **Configurando o Roteamento do Canal**, na seção **Configurando o Módulo R** no capítulo 5 do Manual de Utilização do MasterTool.
- **Declaração da Redundância do Canal:** especifica os parâmetros de redundância do canal: habilitação ou não da redundância, período de teste para a conexão ativa e atraso para comutação. Devem ser declarados na janela de edição do módulo R. Para maiores informações sobre como declarar os parâmetros de redundância, ver o item **Configurando a Redundância do Canal**, na seção **Configurando o Módulo R** no capítulo 5 do Manual de Utilização do MasterTool.

Estados de Operação do Roteador

Existem quatro estados ou modos de operação do roteador: inicialização, execução, programação, e erro. O estado em que o roteador se encontra é indicado nos LEDs do painel frontal da UCP, podendo também ser consultado pelo MasterTool, através da caixa de diálogo **Estado** (opções

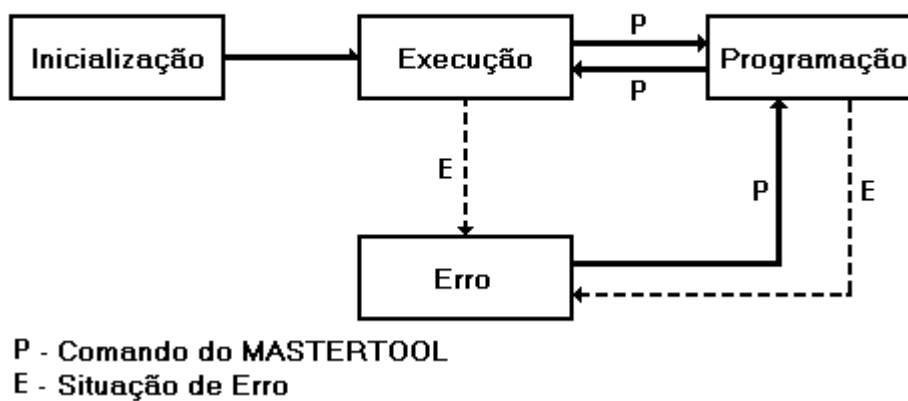
Comunicação, Estado, a partir do menu principal). Para obter informações específicas sobre os modos de operação, consultar o manual de utilização do roteador utilizado.

- **Estado Inicialização:** o roteador inicializa as diversas estruturas de dados de uso do programa executivo e realiza consistências no projeto de roteador presente na memória. Este estado ocorre após a energização do roteador, passando após alguns segundos para o estado execução. Caso não exista um módulo R na memória, o roteador passa para modo erro.

Enquanto o roteador está inicializando, pode-se acionar o comando **Comunicação, Estado, Programação**, ou atalho equivalente na barra de ferramentas, fazendo com que o roteador passe diretamente para o estado de programação, ao invés de passar para modo execução. Este procedimento é útil para a reinicialização de roteadores com módulos R contendo erros graves de configuração.

- **Estado Execução:** normalmente o roteador se encontra neste estado, executando o roteamento conforme o módulo R. Este estado indica que o roteador está executando corretamente o roteamento.
- **Estado Programação:** o roteamento não é executado. O roteador permanece inoperante, esperando comandos do MasterTool. Este modo normalmente é utilizado para a carga do módulo do projeto de roteador pelo MasterTool, através do canal serial.
- **Estado de Erro:** indica que houve alguma anomalia no roteador durante o processamento do projeto de roteador. O tipo de erro ocorrido pode ser consultado através da caixa de diálogo (opções **Comunicação, Estado, Informações** a partir do menu principal), enquanto o roteador estiver neste estado. A saída do estado de erro somente é possível passando-se o roteador para modo programação.

Em condições normais, o roteador pode estar nos modos execução e programação, sendo esses modos acionados através de comandos do MasterTool (opções **Execução, Programação e Ciclado** da caixa de diálogo **Estado**, ou seus atalhos equivalentes na **Barra de Ferramentas**). Na ocorrência de alguma situação de funcionamento errôneo nestes modos, o roteador passa para estado de erro. A recuperação do modo erro somente é possível passando-se o roteador para modo programação. A figura 2-33 apresenta as possibilidades de troca de estados.

MODOS DE OPERAÇÃO DO ROTEADOR**Figura 2-36 Estados de Operação do Roteador**

Nos modos execução e programação, é possível carregar e ler módulos R pelo canal serial do roteador, bem como monitorar e forçar quaisquer operandos utilizados. Essas operações não são possíveis caso o roteador esteja em modo erro.

Referência das Instruções

Este capítulo apresenta a lista de instruções integrantes da Linguagem de Diagramas e Relés ALTUS, descrevendo o formato, o uso, a sintaxe e fornecendo exemplos de cada instrução.

Lista das Instruções

Os CPs ALTUS utilizam a linguagem de relés e blocos para a elaboração do programa aplicativo, cuja principal vantagem, além de sua representação gráfica, é ser similar a diagramas de relés convencionais.

A programação desta linguagem, realizada através do MasterTool, utiliza um conjunto de poderosas instruções apresentadas nas seções seguintes.

As instruções do MasterTool podem ser divididas em 7 grupos:

- **RELÉS**
- **MOVIMENTADORES**
- **ARITMÉTICOS**
- **CONTADORES**
- **CONVERSÕES**
- **GERAIS**
- **LIGAÇÕES**

Convenções Utilizadas

Foram utilizadas diversas convenções para a apresentação dos grupos e instruções tornando melhor a visualização e reconhecimento dos itens descritos, visando com isto um aprendizado mais simples e uma fonte de consulta direta aos tópicos desejados.

Apresentação dos Grupos

A descrição de cada grupo segue o seguinte roteiro.

1. O grupo é descrito com um título contendo o nome do grupo.
2. Logo após o título, é realizada uma breve descrição das características comuns às instruções do grupo.
3. Finalizando a apresentação do grupo, é exibida uma tabela contendo na primeira coluna o nome da instrução, na segunda coluna a descrição do nome da instrução e na terceira coluna a sequência de teclas para realizar a inserção da instrução diretamente pelo teclado.

Exemplo:

Instruções do Grupo Relés

As instruções do grupo **Relés** são utilizadas para o processamento lógico dos diagramas de relés. Através das mesmas pode-se manipular os valores dos pontos digitais de entrada (%E) e saída (%S), bem como pontos de operandos auxiliares (%A), memória (%M) e decimal (%D).

São usadas também para desvio do fluxo e controle do processamento do programa aplicativo.

Nome	Descrição do Nome	Seqüência de Edição	Barra de Ferramentas
<u>RNA</u>	contato normalmente aberto	ALT, R, A	
<u>RNF</u>	contato normalmente fechado	ALT, R, F	
<u>BOB</u>	bobina simples	ALT, R, B	
<u>SLT</u>	bobina de salto	ALT, R, S	
<u>BBL</u>	bobina liga	ALT, R, L	
<u>BBD</u>	bobina desliga	ALT, R, D	
<u>PLS</u>	relé de pulso	ALT, R, P	
<u>FRM</u>	fim de relé mestre	ALT, R, M	
<u>RM</u>	relé mestre	ALT, R, R	

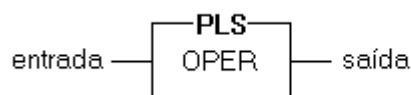
Apresentação das Instruções

A descrição de cada instrução é feita da seguinte maneira.

1. A instrução é descrita com um título contendo o nome da instrução e a descrição do nome. Uma figura apresentando como a instrução é visualizada no diagrama de relés contendo seus operandos, entradas e saídas. Abaixo da figura, é exibida uma breve descrição do significado de cada operando.
2. O item **Descrição** contém informações descrevendo o funcionamento da instrução conforme as entradas habilitadas e os tipos de operando utilizados. Neste item também são descritas as saídas que serão acionadas após a execução da instrução.
3. O item **Sintaxe** descreve as combinações de operandos que podem ser utilizados na instrução. Este item somente está presente nas instruções que possuam operandos.
4. O item **Exemplo** fornece um exemplo de utilização da instrução descrevendo seu comportamento. Este item somente está presente nas instruções que requeiram um detalhamento maior de seu funcionamento.
5. Podem existir outros itens descrevendo uma característica específica da instrução caso haja necessidade.

Exemplo:

PLS - Relé de Pulso



Descrição:

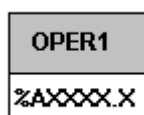
A instrução relé de pulso gera um pulso de uma varredura em sua saída, ou seja, permanece energizada durante uma varredura do programa aplicativo quando o estado da sua entrada passar de desenergizado para energizado.

O relé auxiliar declarado serve como memorizador, evitando limitações quanto ao número de instruções de pulso presentes no programa aplicativo.

ATENÇÃO:

O valor do relé auxiliar não deve ser utilizado em nenhum outro ponto do programa aplicativo.

Sintaxe:



Instruções do Grupo Relés

As instruções do grupo **Relés** são utilizadas para o processamento lógico dos diagramas de relés. Através das mesmas pode-se manipular os valores dos pontos digitais de entrada (%E) e saída (%S), bem como pontos de operandos auxiliares (%A), memória (%M) e decimal (%D).

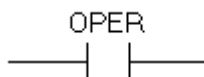
São usadas também para desvio do fluxo e controle do processamento do programa aplicativo.

Nome	Descrição do Nome	Seqüência de Edição	Barra de Ferramentas
<u>RNA</u>	contato normalmente aberto	ALT, R, A	
<u>RNF</u>	contato normalmente fechado	ALT, R, F	
<u>BOB</u>	bobina simples	ALT, R, B	
<u>SLT</u>	bobina de salto	ALT, R, S	
<u>BBL</u>	bobina liga	ALT, R, L	
<u>BBD</u>	bobina desliga	ALT, R, D	
<u>PLS</u>	relé de pulso	ALT, R, P	
<u>FRM</u>	fim de relé mestre	ALT, R, M	
<u>RM</u>	relé mestre	ALT, R, R	

Tabela 3-1 Instruções do Grupo Relés

Contatos

- **RNA** contato normalmente aberto



- **RNF** contato normalmente fechado



Descrição:

Estas instruções refletem, logicamente, o comportamento real de um contato elétrico de um relé no programa aplicativo.

O contato normalmente aberto fecha conforme o estado do seu operando associado. Caso o ponto do operando esteja no estado lógico **1** ou **0**, o contato normalmente aberto está fechado ou aberto, respectivamente.

O contato normalmente fechado possui comportamento oposto ao normalmente aberto. Caso o ponto do operando associado esteja no estado lógico **1** ou **0**, o contato normalmente fechado está aberto ou fechado, respectivamente.

Quando um contato está fechado, a instrução transmite o estado lógico da sua entrada para a sua saída. Se estiver aberto, o valor da entrada não é colocado na saída.

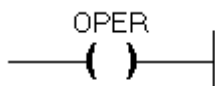
Sintaxe:

OPER1
%EXXXX.X
%SXXXX.X
%AXXXX.X
%MXXXX.X
%DXXXX.X
%DXXXXhX

Tabela 3-2 Sintaxe das Instruções RNA e RNF

Bobinas

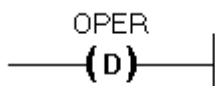
- **BOB** Bobina Simples



- **BBL** Bobina Liga



- **BBD** Bobina Desliga



Descrição:

As instruções bobina modificam o estado lógico do operando na memória imagem do controlador programável, conforme o estado da linha de acionamento das mesmas.

A bobina simples liga ou desliga o ponto do operando conforme a linha de acionamento, enquanto que as bobinas do tipo liga e do tipo desliga somente ligam ou desligam os operandos quando a linha está energizada ("set"/"reset").

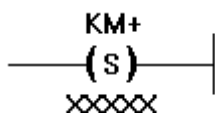
Estas instruções somente podem ser posicionadas na coluna 7 da lógica.

Sintaxe:

OPER1
%SXXXX.X
%AXXXX.X
%MXXXX.X
%DXXXX.X
%DXXXXhX

Tabela 3-3 Sintaxe das Instruções BOB, BBL e BBD

SLT - Bobina de Salto



Descrição:

A instrução bobina de salto serve para controlar a seqüência de execução de um programa aplicativo, sendo usada para desviar o processamento do mesmo para uma lógica determinada.

Seu operando é uma constante que determina o número de lógicas a serem saltadas a partir da energização da bobina. A determinação da lógica destino é realizada pela soma da constante que acompanha a instrução com o número da lógica onde a mesma se encontra.

Quando a linha de acionamento da bobina de salto estiver desenergizada, o salto não ocorre, e a instrução seguinte àquela em que esta bobina está declarada é executada.

Exemplo:

Supondo que a instrução a seguir esteja na lógica 2, a execução do programa aplicativo é desviada para a lógica 7 se a linha de acionamento estiver energizada, ou seja, se o valor do ponto %A0009.3 for **1**. Se o valor deste ponto for **0**, a execução continuará normalmente na lógica 003.

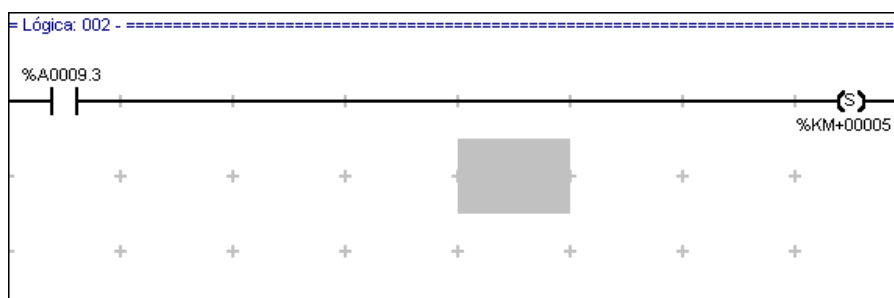


Figura 3-1 Exemplo da Instrução SLT

Esta instrução pode ser posicionada somente na coluna 7 da lógica.

Pode ser utilizada nesta instrução uma constante %KM com valor zero ou mesmo com valor negativo. Se programada com o valor zero, a lógica destino é a mesma que contém a bobina de salto, quando esta é energizada. Ou seja, o processamento é desviado para o início da própria lógica da bobina. Se o valor programado é negativo, o processamento é desviado para uma lógica anterior à lógica que contém a bobina de salto.

ATENÇÃO:

O uso de constante zero ou negativa corresponde a um uso não convencional da instrução. Caso deseje-se utilizá-la, deve-se tomar os cuidados necessários para evitar a entrada em laço infinito de execução "loop" ou o aumento excessivo do tempo de ciclo do programa aplicativo. Recomenda-se, contudo, a utilização da bobina de salto somente com constantes positivas maiores que zero.

O controle da execução nestas situações deve ser realizado através de um intertravamento que desligue o salto para a lógica anterior, após um certo número de laços executados no trecho de retorno.

Caso a lógica destino ultrapasse a última lógica do programa aplicativo, o CP salta para o final do programa e continua seu ciclo normal.

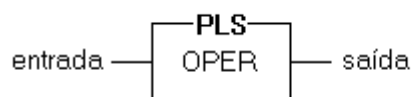
Caso a lógica destino de um salto de retorno seja menor do que a primeira lógica do programa aplicativo, a execução é reiniciada a partir da lógica 0.

Sintaxe:

OPER1
%KM+XXXXX
%KM-XXXXX

Tabela 3-4 Sintaxe da Instrução SLT

PLS - Relé de Pulso



Descrição:

A instrução relé de pulso gera um pulso de uma varredura em sua saída, ou seja, permanece energizada durante uma varredura do programa aplicativo quando o estado da sua entrada passar de desenergizado para energizado.

O relé auxiliar declarado serve como memorizador, evitando limitações quanto ao número de instruções de pulso presentes no programa aplicativo.

ATENÇÃO:

O valor do relé auxiliar não deve ser modificado em nenhum outro ponto do programa aplicativo.

Sintaxe:

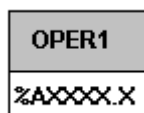


Tabela 3-5 Sintaxe da Instrução PLS

RM, FRM - Relé Mestre, Fim de Relé Mestre

- **RM** Relé Mestre



- **FRM** Fim de Relé Mestre

**Descrição:**

As instruções relé mestre e fim de relé mestre são utilizadas para delimitar trechos de programas aplicativos, energizando ou não a barra lógica de alimentação nos mesmos, conforme o estado da sua linha de acionamento.

Estas instruções não necessitam de operandos, podendo ser posicionadas somente na coluna 7 da lógica.

Quando a entrada da instrução RM estiver desenergizada, a barra lógica de alimentação é desenergizada desde a lógica seguinte até a lógica que contém a instrução FRM.

Como estas instruções atuam sempre na lógica seguinte a que estão contidas é aconselhável o seu posicionamento sempre como últimas instruções da lógica em que estiverem presentes. Assim sendo, o trecho de programa aplicativo delimitado visualmente pelas instruções no diagrama corresponde exatamente ao controlado pelas mesmas, evitando assim má interpretação de seu funcionamento.

ATENÇÃO:

As instruções CON, COB, TEE e TED contém saídas energizadas mesmo sem o acionamento das suas entradas. Estas saídas permanecem energizadas mesmo dentro de um trecho sob comando de um relé mestre desenergizado, podendo causar acionamentos indesejáveis.

Instruções do Grupo Movimentadores

Estas instruções são utilizadas para a manipulação e transferência de valores numéricos entre constantes, operandos simples ou tabelas de operandos.









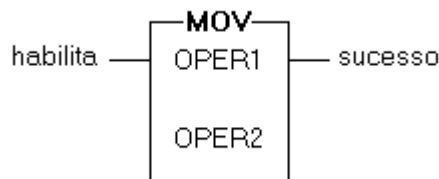
Nome	Descrição do Nome	Seqüência de Edição	Barra de Ferramentas
<u>MOV</u>	movimentação de operandos simples	ALT, M, V	
<u>MOP</u>	movimentação de partes de operandos	ALT, M, P	
<u>MOB</u>	movimentação de blocos de operandos	ALT, M, B	
<u>MOT</u>	movimentação de tabelas de operandos	ALT, M, T	
<u>MES</u>	movimentação de entradas ou saídas	ALT, M, E	
<u>AES</u>	atualização de entradas ou saídas	ALT, M, A	
<u>CES</u>	conversão de entradas ou saídas	ALT, M, S	
<u>CAB</u>	carrega bloco	ALT, M, C	

Tabela 3-6 Instruções do Grupo Movimentadores

MOV - Movimentação de Operandos Simples



OPER1 - operando origem

OPER2 - operando destino

Descrição:

Esta instrução move o conteúdo de operandos simples, quando a entrada **habilita** é acionada.

O operando que ocupa a primeira célula da instrução (OPER1) é o operando origem, cujo valor é movimentado para o operando destino, especificado na segunda célula (OPER2).

Se o formato do operando destino for menor que o do origem, os octetos mais significativos do valor origem são desprezados. Se o formato do destino for maior, seus octetos mais significativos são zerados. Se a movimentação for realizada, a saída **sucesso** é acionada.

Se os índices indiretos excederem os limites de operandos declarados no módulo de configuração, a movimentação não é efetuada e a saída **sucesso** não é ligada.

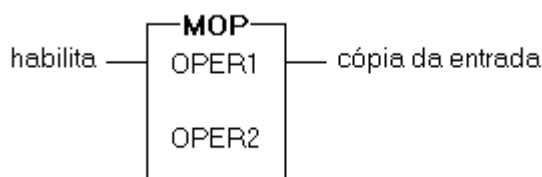
Não é permitida a movimentação de subdivisões de operandos. Para isto, deve ser usada a instrução MOP.

Quando o operando destino da instrução é um inteiro (%M) e pelo menos um dos demais operandos da instrução é um real (%F) o resultado armazenado será truncado, ou seja, armazena-se no operando M apenas a parte inteira do resultado da operação, desprezando-se a parte fracionária. **Sintaxe da Instrução:**

OPER1	OPER2	OPER1	OPER2
%E	%E		
%S	%S		
%A	%A		
%M	%M	%M	%M
%I	%I	%F	%M
%D	%D	%I	%F
%M*E	%M*E	%M*M	%I
%M*S	%M*S	%M*F	%M*M
%M*A	%M*A	%M*I	%M*F
%M*M	%M*M	%KM	%M*I
%M*I	%M*I	%KF	
%M*D	%M*I	%KI	
%KM	%M*D		
%KD			

Tabela 3-7 Sintaxe da Instrução MOV

MOP - Movimentação de Partes (Subdivisões) de Operandos



OPER1 - operando origem

OPER2 - operando destino

Descrição:

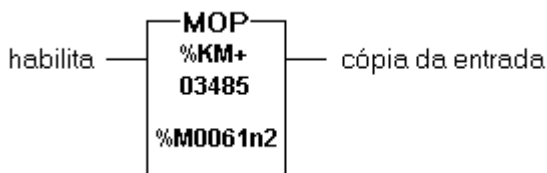
Esta instrução move conteúdos de partes de operandos simples (palavras, octetos, "nibbles", pontos) quando a entrada habilita é energizada. Não é realizada a conversão entre tipos de operandos, apenas a movimentação dos valores.

O operando que ocupa a primeira célula da instrução (OPER1) é o operando origem, cujo valor é movimentado para o operando destino especificado na segunda célula (OPER2). O tipo de subdivisão usado no primeiro operando deve ser o mesmo do segundo.

ATENÇÃO:

Se a movimentação é realizada de uma constante para um operando, é considerada sempre a subdivisão menos significativa da constante igual à declarada no operando destino. Devido a esta característica, sugere-se que sempre seja declarado na constante origem o valor real a ser movimentado, para maior clareza do programa.

Exemplo:



O operando destino da instrução está declarado com subdivisão de nibble. Portanto, o nibble menos significativo da constante origem (com valor igual a 1101 em binário, 13 em decimal) será movido para o nibble 2 da memória M0061.

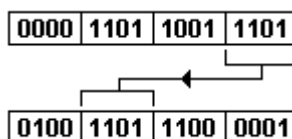


Figura 3-2 Exemplo da Instrução MOP

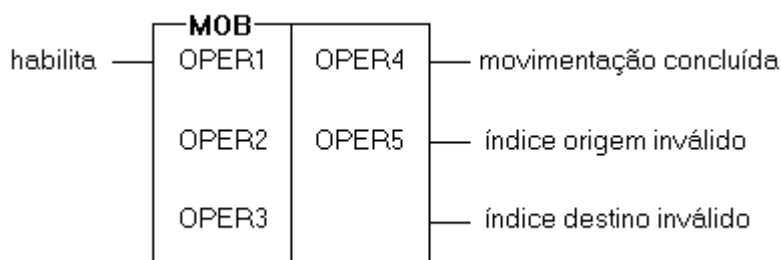
Os demais bits que compõem a constante são desprezados, ou seja, o resultado da movimentação seria idêntico utilizando-se uma constante %KM00013. O exemplo apresentado utiliza um valor excedente ao da movimentação para melhor ilustrar o funcionamento da MOP. Para melhor interpretação do programa deve-se utilizar o valor %KM00013.

Sintaxe:

OPER1	OPER2	OPER1	OPER2
%EXXX.X %SXX.X %AXX.X %MXX.X %DXX.X %DXXhX %FXX.X %FXXhX %IXXX.X %IXXXhX %KMXXXXX %KDXXXXX	%EXXX.X %SXX.X %AXX.X %MXX.X %DXX.X %DXXhX %FXX.X %FXXhX %IXXX.X %IXXXhX	%MXXbX %DXXbX %FXXbX %IXXXbX %EXXX %SXX %AXX %KMXXXXX %KDXXXXX	%MXXbX %DXXbX %FXXbX %IXXXbX
OPER1	OPER2	OPER1	OPER2
%EXXXnX %SXXnX %AXXnX %MXXnX %DXXnX %FXXnX %IXXXnX %KMXXXXX %KDXXXXX	%EXXXnX %SXXnX %AXXnX %MXXnX %DXXnX %FXXnX %IXXXnX	%MXXbX %DXXbX %FXXbX %IXXXbX	%EXXX %SXX %AXX
OPER1	OPER2	OPER1	OPER2
%DXXwX %FXXwX %IXXXwX %MXX %KMXXXXX %KDXXXXX	%DXXwX %FXXwX %IXXXwX	%DXXwX %FXXwX %IXXXwX	%MXX

Tabela 3-8 Sintaxes da Instrução MOP

MOB - Movimentação de Blocos de Operandos



OPER1 - primeiro operando do bloco origem
 OPER2 - número de transferências a realizar
 OPER3 - operando de controle
 OPER4 - primeiro operando do bloco destino
 OPER5 - número de transferências por varredura

Descrição:

Esta instrução realiza a cópia dos valores de um bloco de operandos origem para um bloco destino.

Especifica-se o primeiro operando do bloco origem em OPER1 e o primeiro operando do bloco destino em OPER4. O número total de transferências a serem realizadas é declarado no parâmetro OPER2, devendo também ser especificados o número de transferências por varredura (OPER5) e uma memória acumuladora para a contagem do número de transferências (OPER3).

Se o bloco origem ou destino for uma tabela, a transferência tem início na sua primeira posição.

Se o formato do operando destino for menor do que o origem, os octetos mais significativos do valor origem são desprezados. Caso contrário, os octetos mais significativos do destino são zerados.

O número de transferências por varredura é limitado em 255 operandos. Na medida do possível deve-se evitar um número elevado de transferências na mesma varredura, para diminuir o tempo de execução do programa.

Em cada instrução MOB é utilizada uma memória como operando de controle (OPER3), que deve estar zerada antes da primeira execução.

ATENÇÃO:

O operando de controle não deve ter seu conteúdo alterado em nenhuma parte do programa aplicativo, sob pena de prejudicar a execução correta da instrução. Cada ocorrência desta instrução no programa deve possuir um operando de controle exclusivo, diferente dos demais. Este operando não pode ser retentivo.

Quando ligadas, as saídas da segunda e terceira células indicam, respectivamente, que pelo menos um dos operandos componentes do bloco origem ou destino tem endereço superior ao número máximo declarado para o operando ou tabela utilizado, não sendo realizada nenhuma movimentação. Caso o valor do segundo operando seja negativo, a saída **índice origem inválido** é acionada.

A saída da primeira célula é acionada na varredura em que a movimentação for completada.

ATENÇÃO:

A entrada **habilita** deve permanecer ativa até que a movimentação esteja concluída. Como esta instrução é executada em múltiplos ciclos de execução, não deve ser saltada enquanto não estiver terminada a movimentação.

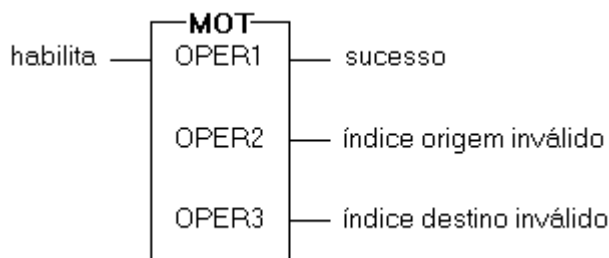
Sintaxe:

OPER1	OPER2	OPER3	OPER4	OPER5
%E %S %A %M %I %D %TM %TD	%KM	%M	%E %S %A %M %I %D %TM %TD	%KM

OPER1	OPER2	OPER3	OPER4	OPER5
%F %TF	%KM	%M	%F %TF	%KM

Tabela 3-9 Sintaxe da Instrução MOB

MOT - Movimentação de Tabelas



OPER1 - tabela origem ou operando origem

OPER2 - índice da tabela

OPER3 - operando destino ou tabela destino

Descrição:

Esta instrução permite duas operações: transferir o valor de uma posição de tabela para um operando simples ou de um operando simples para uma posição de tabela.

O operando que ocupa a primeira célula da instrução (OPER1) é o operando origem, cujo valor é movimentado para o operando destino especificado na terceira célula (OPER3). OPER2 contém a posição da tabela declarada em OPER1 ou OPER3.

Leitura de conteúdo de tabela:

Permite ler o conteúdo de uma posição de tabela e carregá-lo em um operando memória ou decimal.

A instrução é programada da seguinte forma:

- OPER1 - especifica o endereço da tabela a ser lida
- OPER2 - especifica a posição (%KM) a ser lida ou a memória (%M) que contém esta posição
- OPER3 - especifica para onde o conteúdo da posição de tabela deve ser transferido

Se o primeiro operando referenciar indiretamente uma tabela não especificada, ou se o valor do segundo operando for negativo ou maior que a última posição definida para a tabela, a transferência não é realizada e a saída **índice origem inválido** é acionada. Se o terceiro operando referenciar indiretamente um operando não declarado, a transferência não é realizada e a saída **índice destino inválido** é acionada.

Escrita de valores em tabela:

Permite escrever um valor constante ou o conteúdo de um operando memória ou decimal em uma posição de tabela.

A instrução é programada da seguinte forma:

- OPER1 - especifica o operando origem
- OPER2 - especifica a posição (%KM) a ser escrita na tabela ou a memória (%M) que contém esta posição
- OPER3 - especifica o endereço da tabela para onde é transferido o conteúdo

Se o primeiro operando referenciar indiretamente um operando não declarado, a transferência do conteúdo não é realizada e a saída **índice origem inválido** é acionada. Se o valor do segundo operando for negativo ou maior que a última posição definida para a tabela, ou se o terceiro operando referenciar indiretamente uma tabela não especificada, a transferência do conteúdo não é realizada e a saída **índice destino inválido** é acionada.

Esta instrução simplifica a programação de uma série de algoritmos envolvendo decodificações, seqüenciamentos, geração de curvas, armazenamento e comparação de valores, entre outros.

Sintaxe:

Leitura:

OPER1	OPER2	OPER3
%TM %M*TM	%KM %M	%M %M*M

Escrita:

OPER1	OPER2	OPER3
%KM %M %M*M	%KM %M	%TM %M*TM

OPER1	OPER2	OPER3
%TD %M*TD	%KM %M	%D %M*D

OPER1	OPER2	OPER3
%KD %D %M*D	%KM %M	%TD %M*TD

OPER1	OPER2	OPER3
%TF %M*TF	%KM %M	%F %M*F

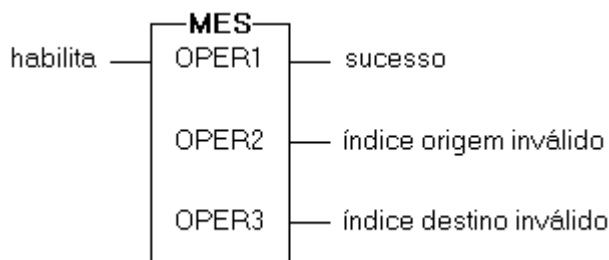
OPER1	OPER2	OPER3
%KF %F %M*F	%KM %M	%TF %M*TF

OPER1	OPER2	OPER3
%TI %M*TI	%KM %M	%I %M*I

OPER1	OPER2	OPER3
%KI %I %M*I	%KM %M	%TI %M*TI

Tabela 3-10 Sintaxes da Instrução MOT

MES - Movimentação de Entradas/Saídas



OPER1 - primeiro operando origem
 OPER2 - número de octetos a transferir
 OPER3 - primeiro operando destino

Descrição:

Esta instrução é utilizada para a transferência de dados diretamente entre operandos memória e octetos do barramento de módulos de entrada e saída. É possível realizar leituras de valores dos octetos do barramento ou escritas no mesmo, conforme os operandos programados na instrução.

O operando que ocupa a primeira célula da instrução (OPER1) é o operando origem, cujo conteúdo será movimentado para o operando destino especificado na terceira célula (OPER3). OPER2 define o número de octetos a serem transferidos a partir do primeiro origem e destino especificados.

ATENÇÃO:

O número de octetos a serem transferidos está limitado em 255.

Caso seja programado uma constante na primeira célula (escrita de valor no barramento), o seu valor é movido para todos os octetos do barramento especificados pelos operandos da segunda e terceira células.

Sempre que a entrada **habilita** está energizada, uma das saídas da instrução é energizada, conforme as regras a seguir.

A saída **índice origem inválido** é energizada em 3 situações:

- o número de transferências especificado em OPER2 for negativo, zero, maior do que o número máximo de octetos no barramento do CP utilizado (leitura do barramento) ou que o limite de memórias configurado (escrita no barramento)
- a primeira posição lida for maior do que o número máximo de octetos no barramento do CP utilizado (%M*R programado em OPER1)
- o primeiro endereço de memória a ser escrito for negativo ou maior do que o último endereço de memória configurado (%M*M programado em OPER1)

A saída **índice destino inválido** é energizada quando:

- o número de transferências especificado em OPER2 for maior do que o limite de memórias configurado (leitura do barramento) ou que o número máximo de octetos no barramento do CP utilizado (escrita no barramento)
- a primeira posição escrita for maior do que o número máximo de octetos no barramento do CP utilizado (%M*R programado em OPER3)
- o primeiro endereço de memória a ser lido for negativo ou maior do que o último endereço de memória configurado (%M*M programado em OPER3)

A saída **sucesso** é energizada quando as saídas **índice origem inválido** e **índice destino inválido** não forem energizadas.

Esta instrução é utilizada somente para acessos especiais ao barramento. Para o seu uso, deve-se saber exatamente que módulo de E/S está colocado na posição física do barramento lida ou escrita pela MES e como acessá-lo. Como os módulos de entrada e saída fornecidos pela ALTUS possuem instruções específicas para o seu acesso, a instrução MES não é necessária na maioria dos programas aplicativos.

Não é possível escrever valores em octetos de módulos digitais de entrada ou ler valores de octetos de módulos digitais de saída com a MES.

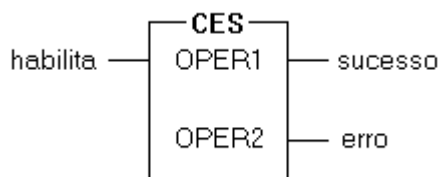
Sintaxe:

OPER1	OPER2	OPER3
%R %M*R	%KM %M	%M %M*M

OPER1	OPER2	OPER3
%KM %M %M*M	%KM %M	%R %M*R

Tabela 3-11 Sintaxes da Instrução MES

CES - Conversão de Entradas/Saídas



OPER1 - operando origem

OPER2 - operando destino

Descrição:

Esta instrução é utilizada para a transferência de dados diretamente entre operandos memória e octetos no barramento, convertendo os valores de binário para BCD, em caso de escrita no barramento, ou BCD para binário, em caso de leitura.

Caso se deseje converter octetos do barramento para uma memória, deve-se programar em OPER1 o octeto inicial e em OPER2 a memória a receber os valores convertidos. A instrução concatena o valor do octeto especificado com o octeto seguinte, converte do formato BCD para binário e armazena o valor convertido na memória destino.

Caso se deseje converter valores de uma memória ou constante memória para o barramento, deve-se especificar em OPER1 o valor a ser convertido em OPER2 o octeto inicial a receber os valores. A instrução converte o valor para o formato BCD e escreve o mesmo no octeto especificado e no seguinte. Se o valor movimentado para o barramento possuir mais do que 4 dígitos, os dígitos mais significativos excedentes serão descartados.

Exemplo:

Movimentar o conteúdo de %M0100 para %R0010:

- valor de %M0100=21947, equivalente a 101010110111011 no formato binário
- valor de %M0100=21947, convertido para 0010 0001 1001 0100 0111 no formato BCD
- valor movido para %R0010=47 no formato BCD, equivalente a 0100 0111 escrito no octeto
- valor movido para %R0011=19 no formato BCD, equivalente a 0001 1001 escrito no octeto

A instrução é executada sempre que a entrada **habilita** é energizada. A saída **sucesso** é energizada caso a instrução tenha sido executada corretamente.

A saída **erro** é energizada quando se faz um acesso inválido a algum operando referenciado indiretamente por uma memória.

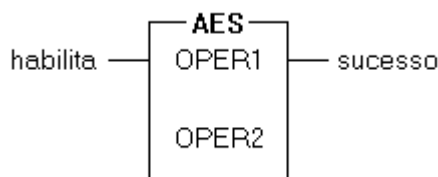
Sintaxe:

OPER1	OPER2	OPER1	OPER2
%R	%M	%KM	%R
%M*R	%M*M	%M	%M*R
		%M*M	

Tabela 3-12 Sintaxes da Instrução CES

Esta instrução não está disponível para as UCPs AL-3003 e AL-3004.

AES - Atualiza Entradas/Saídas



OPER1 - primeiro octeto de operandos a atualizar

OPER2 - número de octetos a atualizar

Descrição:

Esta instrução executa uma atualização imediata na memória imagem para os operandos especificados. Sua atuação é idêntica à varredura dos pontos de E/S realizada pelo programa executivo ao final de cada varredura, porém com o número de operandos limitados.

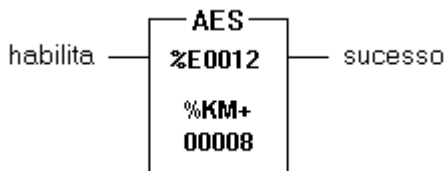
O primeiro operando (OPER1) contém o primeiro octeto de operandos a ser atualizado, enquanto que o segundo operando (OPER2) indica o número total de octetos a atualizar. Os operandos %E (entrada) são lidos do barramento para a memória imagem e os operandos %S (saída) são escritos da memória imagem para o barramento quando a instrução é executada.

Se o número de operandos a atualizar ultrapassar o número de operandos declarados, são atualizados somente os possíveis dentro do tipo declarado.

Se nenhum octeto for atualizado pela instrução, a saída **sucesso** é desenergizada.

A instrução AES deve ser usada somente em processamentos especiais, nos quais um tempo de resposta muito rápido ou constante é exigido do CP. Em programas aplicativos relativamente pequenos, com baixo tempo de varredura e tarefas de controle comuns, a mesma não necessita ser utilizada.

Exemplo:



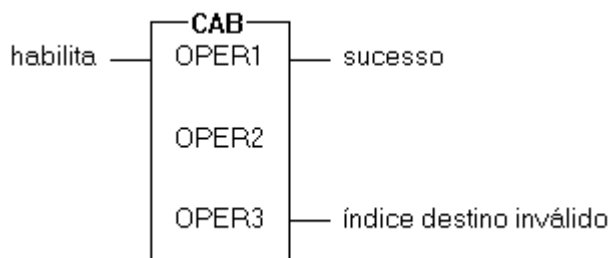
Caso a configuração do CP seja 16 octetos de entrada (%E0000 a %E0015) e 8 octetos de saída (%S0016 a %S0023), a instrução apresentada atualizaria apenas 4 octetos (%E0012 a %E0015). Nenhum octeto de saída é atualizado.

Sintaxe:

OPER1	OPER2
%E %S	%KM %M

Tabela 3-13 Sintaxes da Instrução AES

CAB - Carrega Bloco



OPER1 - operando inicial ou tabela a ser carregada

OPER2 - número de operandos ou posições de tabela

OPER3 - tabela de constantes a serem carregadas

Descrição:

Esta instrução permite a carga de até 255 valores constantes em um bloco de operandos ou em tabelas.

O operando inicial ou tabela a ser carregada é especificado no primeiro parâmetro (OPER1), o número de operandos ou posições da tabela a serem carregados no segundo operando (OPER2) e o valor das constantes no terceiro (OPER3).

O valor do segundo operando deve ser positivo, menor ou igual a %KM+128.

O terceiro operando (OPER3) é composto por uma tabela de valores constantes a serem carregados. Estes valores são declarados selecionando o botão **Bloco**, sendo aberta uma janela de edição no MasterTool. As constantes são do tipo %KM se o tipo do primeiro operando for %E, %S, %A, %M, %TM ou são do tipo %KD se o primeiro operando for %D ou %TD e do tipo %KF se o primeiro operando for %F ou %TF. Caso o primeiro operando seja um octeto (%E, %S ou %A), somente serão movimentados os valores dos octetos menos significativos de cada constante declarada.

Também é possível realizar a declaração dos valores da tabela em ASCII. Este modo permite que seja digitado um texto que será carregado na tabela com os valores ASCII relativos a cada caractere. Neste modo é possível a inserção de endereços ou tags de operandos que devem representar o seu valor no momento em que a instrução for executada. O endereço ou tag do operando deve ser digitado entre chaves ({}).

Ex.: Supondo que %M0000 tenha o valor 35 e que tenha sido carregado o seguinte texto em ASCII "Valor de %M0000: {%M0000}.".O texto fica o seguinte:

Valor de %M0000: 00035.

Quando o botão **Bloco** é selecionado é exibida a caixa de diálogo **CAB - Valores**:

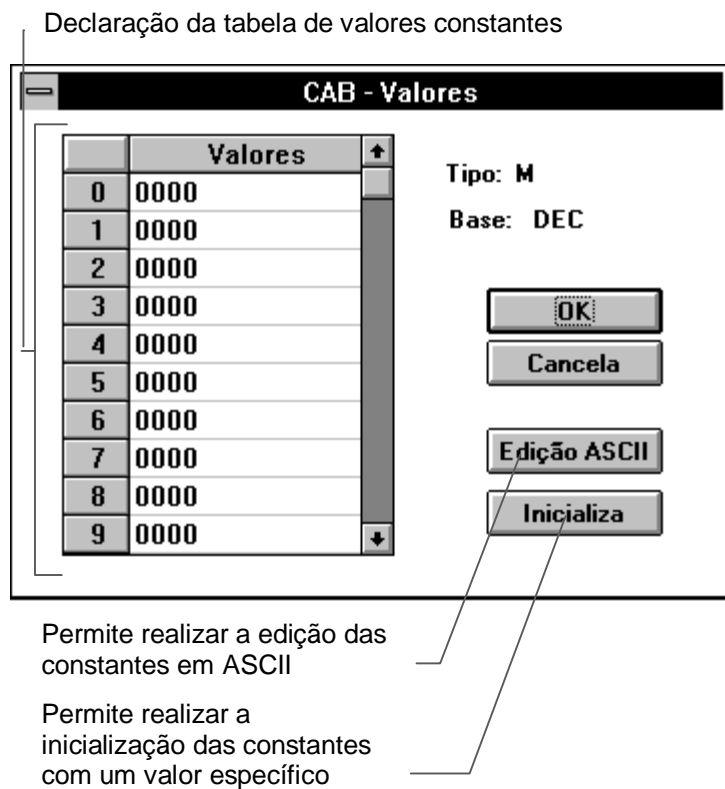


Figura 3-3 Caixa de Diálogo CAB - Valores

Para realizar a edição das constantes

1. Posicionar o cursor no índice a ser editado. Caso seja necessário rolar as páginas podem ser utilizadas as teclas PAGE DOWN e PAGE UP ou a barra de rolagem vertical.
2. Digitar o valor da constante.

Para realizar a edição em ASCII

1. Selecionar o botão **Edição ASCII**. É exibida a caixa de diálogo **CAB - Edição em ASCII**.
2. Digitar o texto que se deseja carregar nas constantes da CAB e selecionar o botão Ok.

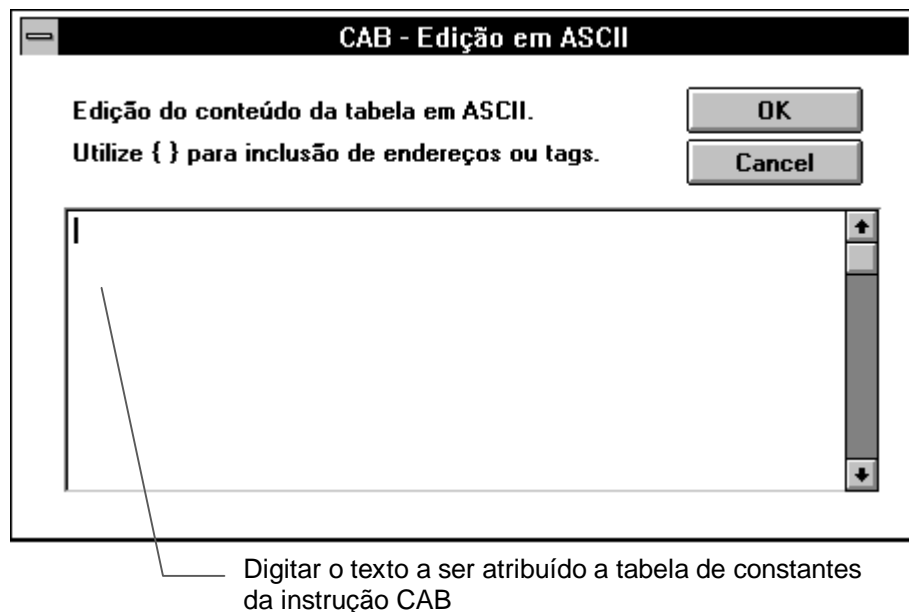


Figura 3-4 Caixa de Diálogo CAB - Edição em ASCII

Para inicializar as constantes com um valor específico

1. Selecionar o botão **Inicializar**. É exibida a janela **CAB - Inicializa tabela**.
2. No item **Valor**, digitar o valor a ser inicializado nas constantes.
3. No item **Posição inicial**, digitar o número da primeira posição a receber o valor de inicialização.
4. No item **Posição final**, digitar o número da última posição a receber o valor de inicialização.
5. Selecionar o botão **Ok**.

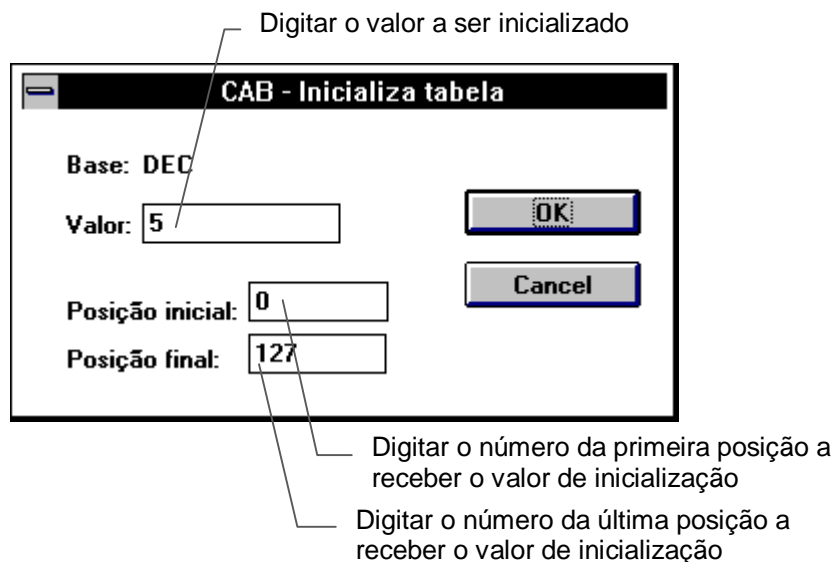


Figura 3-5 CAB - Inicializa Tabela

A saída **índice destino inválido** é acionada quando algum operando não puder ser acessado ou uma posição de tabela não existir. A saída **sucesso** é acionada sempre que a instrução for executada corretamente. Se a saída **índice destino inválido** foi acionada, nenhuma carga de constantes ocorreu.

A carga dos valores constantes é inteiramente realizada em uma só varredura do programa aplicativo, podendo ocasionar um tempo de ciclo excessivo quando o mesmo for extenso. Na maior parte dos programas aplicativos, a instrução CAB pode ser executada somente na inicialização do mesmo (carga de tabelas cujos conteúdos serão somente lidos) ou em alguns momentos especiais, não precisando ser chamada em todas as varreduras. Nestes casos, recomenda-se a sua programação no módulo de programa aplicativo de inicialização (E000) ou que seja acionada apenas nos momentos de carga necessários.

Sintaxe:

OPER1	OPER2	OPER3
%E %S %A %M %TM %M*E %M*S %M*A %M*M %M*TM	%KM	TABELA DE VALORES MEMÓRIA

OPER1	OPER2	OPER3
%D %TD %M*D %M*TD	%KM	TABELA DE VALORES DECIMAIS

OPER1	OPER2	OPER3
%F %TF %M*F %M*TF	%KM	TABELA DE VALORES REAIS

OPER1	OPER2	OPER3
%I %TI %M*I %M*TI	%KM	TABELA DE VALORES INTEIRO

Tabela 3-14 Sintaxes da Instrução CAB

Instruções do Grupo Aritméticas

As instruções aritméticas modificam os valores dos operandos numéricos, permitindo a realização de cálculos aritméticos e lógicos entre os mesmos. Permitem também comparações entre valores de operandos.












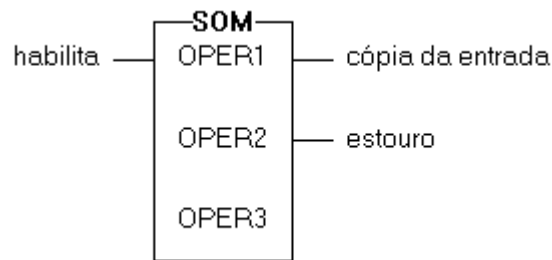
Nome	Descrição do Nome	Seqüência de Edição	Barra de Ferramentas
<u>S</u>OM	adição	ALT, A, S	
<u>S</u>UB	subtração	ALT, A, B	
<u>M</u>UL	multiplicação	ALT, A, M	
<u>D</u>IV	divisão	ALT, A, D	
<u>A</u>ND	função E binário entre operandos	ALT, A, A	
<u>O</u>R	função OU binário entre operandos	ALT, A, O	
<u>X</u>OR	função OU EXCLUSIVO entre operandos	ALT, A, X	
<u>C</u>AR	carrega operandos	ALT, A, C	
<u>I</u>GUAL	igual	ALT, A, I	
<u>M</u>ENOR	menor	ALT, A, N	
<u>M</u>AIOR	maior	ALT, A, R	

Tabela 3-15 Instruções do Grupo Aritméticas

SOM - Adição



OPER1 - primeira parcela
 OPER2 - segunda parcela
 OPER3 - total

Descrição:

Esta instrução realiza a soma aritmética de operandos. Quando a entrada **habilita** é energizada, os valores dos operandos especificados nas duas primeiras células são somados e o resultado armazenado no operando da terceira célula.

Se o resultado da operação for maior ou menor do que o armazenável, a saída **estouro** é energizada e o máximo ou mínimo valor armazenável é atribuído a variável total como resultado.

Se a entrada **habilita** não está energizada, todas as saídas são desenergizadas e o valor de OPER3 não é alterado.

Quando o operando destino da instrução é um inteiro (%M) e pelo menos um dos demais operandos da instrução é um real (%F) o resultado armazenado será truncado, ou seja, armazena-se no operando M apenas a parte inteira do resultado da operação, desprezando-se a parte fracionária.

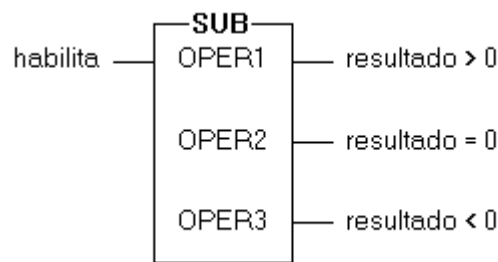
Exs: **9.999999 (%F) + 1.000000(%F) = 10 (%M)** **Sintaxe:**

OPER1	OPER2	OPER3
%KD %D	%KD %D	%D

OPER1	OPER2	OPER3
%KF %F %KM %M %KI %I	%KF %F %KM %M %KI %I	%F %M %I

Tabela 3-16 Sintaxes da Instrução SOM

SUB – Subtração



OPER1 - primeira parcela

OPER2 - segunda parcela

OPER3 - resultado

Esta instrução realiza a subtração aritmética entre operandos. Quando **habilita** é energizada, o valor do operando da segunda célula é subtraído do valor do operando da primeira célula. O resultado é armazenado na memória especificada na terceira célula.

As linhas de saída **resultado > 0**, **resultado = 0** e **resultado < 0** podem ser usadas para comparações e são acionadas de acordo com o resultado da subtração.

Se a entrada **habilita** não está energizada, todas as saídas são desenergizadas e OPER3 permanece inalterado.

Se o resultado da operação excede o maior ou menor valor armazenável no operando, o respectivo valor limite é considerado como resultado.

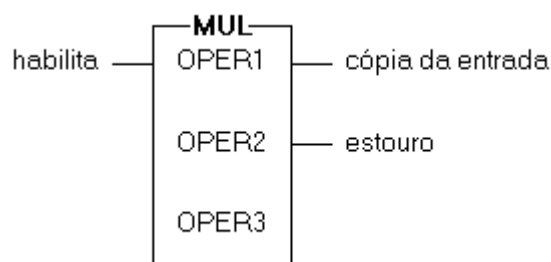
Quando o operando destino da instrução é um inteiro (%M) e pelo menos um dos demais operandos da instrução é um real (%F) o resultado armazenado será truncado, ou seja, armazena-se no operando M apenas a parte inteira do resultado da operação, desprezando-se a parte fracionária. Sintaxe:

OPER1	OPER2	OPER3
%KD %D	%KD %D	%D

OPER1	OPER2	OPER3
%KF %F %KM %M %KI %I	%KF %F %KM %M %KI %I	%F %M %I

Tabela 3-17 Sintaxes da Instrução SUB

MUL - Multiplicação



OPER1 - multiplicando

OPER2 - multiplicador

OPER3 - produto

Descrição:

Esta instrução realiza a multiplicação aritmética de operandos. Quando a entrada **habilita** está energizada, ocorre a multiplicação do conteúdo do operando especificado na primeira célula pelo especificado na segunda.

O resultado é armazenado na memória especificada na terceira célula. Caso este exceda o valor máximo armazenável em uma memória, o resultado final é este valor e a saída **estouro** é energizada. Se a entrada **habilita** é desenergizada, nenhuma saída é ligada e OPER3 permanecerá inalterado.

Quando o operando destino da instrução é um inteiro (%M) e pelo menos um dos demais operandos da instrução é um real (%F) o resultado armazenado será truncado, ou seja, armazena-se no operando M apenas a parte inteira do resultado da operação, desprezando-se a parte fracionária.

Exs: $9.999999 (\%F) * 1.000000 (\%F) = 9 (\%M)$

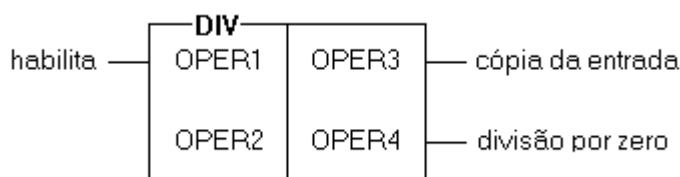
$9.999999 (\%F) * 1 (\%M) = 9 (\%M)$

Sintaxe:

OPER1	OPER2	OPER3
%KF	%KF	
%F	%F	%F
%KM	%KM	%M
%M	%M	%I
%KI	%KI	
%I	%I	

Tabela 3-18 Sintaxe da Instrução MUL

DIV - Divisão



OPER1 - dividendo
 OPER2 - divisor
 OPER3 - quociente
 OPER4 - resto

Descrição:

Esta instrução realiza a divisão aritmética de operandos. Quando a entrada **habilita** está energizada, ocorre a divisão do valor do operando da primeira célula pelo da segunda, sendo o resultado armazenado na memória especificada na terceira célula e o resto da operação colocado no quarto operando. Os operandos da primeira e segunda células podem ser do tipo memória ou constante.

Se o valor do segundo operando for zero, a saída divisão por zero é acionada e em OPER3 é colocado o valor máximo ou mínimo armazenável no operando, conforme o sinal de OPER1. Neste caso, em OPER4 (resto) será armazenado zero. As saídas da instrução somente são energizadas se a entrada **habilita** estiver acionada. Se não estiver acionada, OPER3 e OPER4 permanecerão inalterados.

Sintaxe:

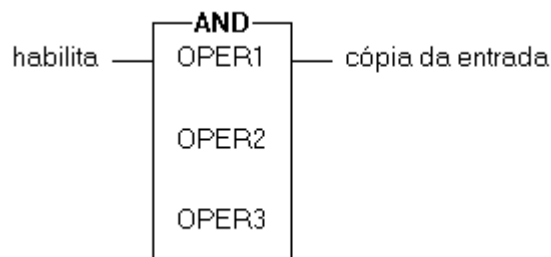
OPER1	OPER2	OPER3	OPER4
%KM %M %KI %I	%KM %M %KI %I	%M %I	%M %I

OPER1	OPER2	OPER3	OPER4
%KF %F %KM %M %KI %I	%KF %F %KM %M %KI %I	%F %M %I	%M (NU)

Tabela 3-19 Sintaxe da Instrução DIV

NU= Não Utilizado, apenas preencher com uma memória qualquer.

AND - E Binário entre Operandos



OPER1 - primeiro operando

OPER2 - segundo operando

OPER3 - resultado

Descrição:

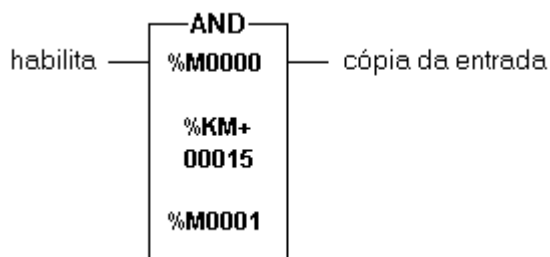
Esta instrução realiza a operação "e" binário entre os dois primeiros operandos, armazenando o resultado no terceiro.

A operação é realizada ponto a ponto entre os operandos. A tabela a seguir mostra as combinações da operação "e" ponto a ponto possíveis.

ponto OPER1	ponto OPER2	ponto OPER3 (resultado)
0	0	0
0	1	0
1	0	0
1	1	1

Tabela 3-20 Operações Ponto a Ponto (AND)

Exemplo:



Neste exemplo deseja-se preservar o valor do nibble menos significativo de %M0000, zerando o resto do operando. Se %M0000 contém 215 (11010111 binário), o resultado do "e" binário com 15 (00001111 binário) é 7 (00000111 binário).

	Decimal		Binário
	215		00000000 11010111 (conteúdo de %M0000)
AND	15	AND	00000000 00001111 (valor de %KM+00015)
	7		00000000 00000111 (resultado em %M0001)

Portanto, o valor 7 decimal é armazenado em %M0001.

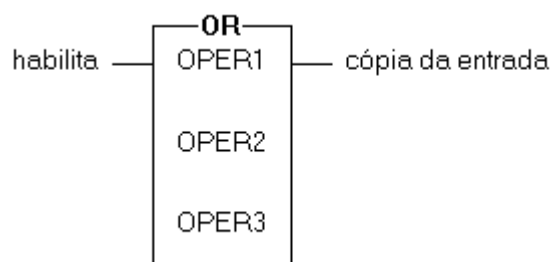
Sintaxe:

OPER1	OPER2	OPER3	OPER1	OPER2	OPER3
%KM %M	%KM %M	%M	%KD %D	%KD %D	%D

OPER1	OPER2	OPER3
%I %KI	%I %KI	%I

Tabela 3-21 Sintaxes da Instrução AND

OR - Ou Binário entre Operandos



OPER1 - primeiro operando

OPER2 - segundo operando

OPER3 - resultado

Descrição:

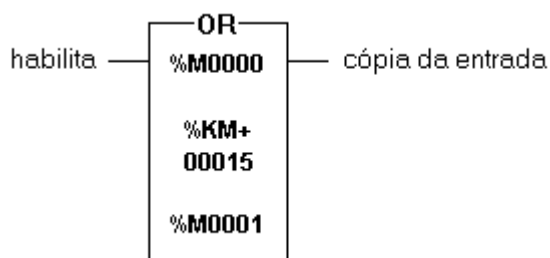
Esta instrução realiza a operação "ou" binário entre os valores dos dois primeiros operandos, armazenando o resultado no terceiro.

A operação é realizada ponto a ponto entre os operandos. A tabela a seguir mostra as combinações da operação "ou" ponto a ponto possíveis.

ponto OPER1	ponto OPER2	ponto OPER3 (resultado)
0	0	0
0	1	1
1	0	1
1	1	1

Tabela 3-22 Operações Ponto a Ponto (OR)

Exemplo:



Neste exemplo deseja-se forçar o nibble menos significativo de %M0000 para 1, preservando-se o valor nos outros nibbles. Se %M0000 contém 28277 (0110111001110101 binário) o resultado é 28287 (0110111001111111 binário).

Decimal	Binário
28277	01101110 01110101 (conteúdo de %M0000)
OR 15	OR 00000000 00001111 (valor de %KM+00015)
28287	01101110 01111111 (resultado em %M0001)

Sintaxe:

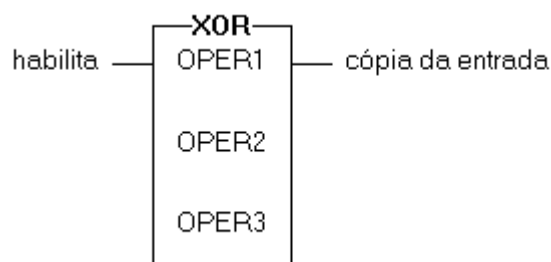
OPER1	OPER2	OPER3
%KM %M	%KM %M	%M

OPER1	OPER2	OPER3
%KD %D	%KD %D	%D

OPER1	OPER2	OPER3
%I %KI	%I %KI	%I

Tabela 3-23 Sintaxes da Instrução OR

XOR - Ou Exclusivo entre Operandos



OPER1 - primeiro operando

OPER2 - segundo operando

OPER3 - resultado

Descrição:

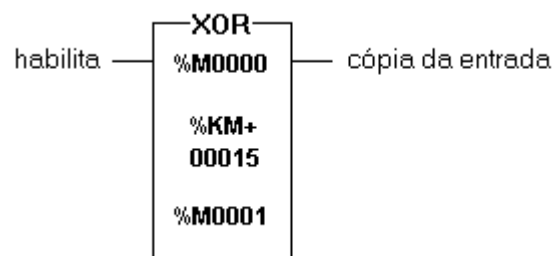
Esta instrução realiza a operação "ou exclusivo" binário entre os dois primeiros operandos, armazenando o resultado no terceiro.

A operação é realizada ponto a ponto entre os operandos. A tabela a seguir mostra as combinações da operação "ou exclusivo" ponto a ponto possíveis.

ponto OPER1	ponto OPER2	ponto OPER3 (resultado)
0	0	0
0	1	1
1	0	1
1	1	0

Tabela 3-24 Operações Ponto a Ponto (XOR)

Exemplo:



Neste exemplo deseja-se inverter os pontos contidos no nibble menos significativo de %M0000, preservando o resto do operando. Se %M0000 contém 1612 (0000011001001100 binário), o resultado é 1603 (0000011001000011 binário)

Decimal	Binário
1612	00000110 01001100 (conteúdo de %M0000)
XOR 15	XOR 00000000 00001111 (valor de %KM+00015)
1603	00000110 01000011 (resultado em %M0001)

Portanto, o valor 1603 decimal é armazenado em M001.

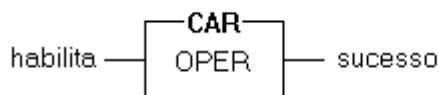
Sintaxe:

OPER1	OPER2	OPER3	OPER1	OPER2	OPER3
%KM %M	%KM %M	%M	%KD %D	%KD %D	%D

OPER1	OPER2	OPER3
%I %KI	%I %KI	%I

Tabela 3-25 Sintaxes da Instrução XOR

CAR - Carrega Operandos



OPER - operando a ser carregado

Descrição:

A instrução carrega operando realiza a carga do valor do operando especificado em registrador especial interno ao CP, para subsequente uso das instruções de comparação (maior, menor, igual). O operando permanece carregado até a próxima instrução de carga, podendo ser utilizado por várias lógicas, inclusive em ciclos de varredura subsequentes.

A saída **sucesso** é acionada se a carga for realizada. Se algum acesso indireto a operando não for possível (índice inválido), a saída **sucesso** não é acionada.

Não pode ser feita a comparação entre operandos decimais e operandos flutuante.

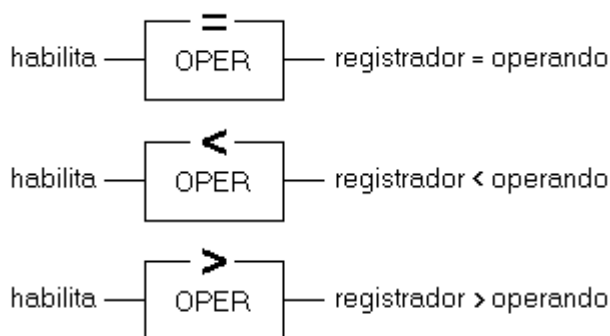
Ver considerações e exemplos apresentados na seção seguinte, **Instruções de Comparação de Operandos**.

Sintaxe:

OPER1
%E
%S
%A
%M
%D
%F
%I
%KM
%KD
%KF
%KI
%M*E
%M*S
%M*A
%M*M
%M*D
%M*F
%M*I

Tabela 3-26 Sintaxe da Instrução CAR

Instruções de Comparação de Operandos - Igual, Maior e Menor



OPER - operando a ser comparado

Descrição:

As instruções maior, menor e igual realizam comparações do operando especificado com o valor previamente carregado no registrador interno com a instrução CAR (Carrega Operando), fornecendo o resultado da comparação em suas saídas. Caso algum acesso indireto seja inválido, a saída é desacionada.

Por exemplo, a instrução maior energiza a sua saída se o valor do operando presente na última instrução CAR ativa for maior que o valor do seu operando. As instruções igual e menor operam de forma idêntica, mudando apenas o tipo de comparação realizada.

Se os operandos a serem comparados são do mesmo tipo, são comparados conforme o seu formato de armazenamento (considerando o seu sinal). Se não são do mesmo tipo, são comparados ponto a ponto (como valores binários sem sinal). Se algum dos operandos diferentes for do tipo real, o operando de menor precisão é convertido para real e após é efetuada a mesma comparação ponto a ponto.

ATENÇÃO:

Sugere-se que sempre sejam comparados operandos de tipos iguais, para evitar má interpretação nos resultados quando os operandos possuírem valores negativos. Ver o exemplo a seguir.

Não pode ser feita a comparação entre operandos decimais e operandos flutuante.

Exemplo:

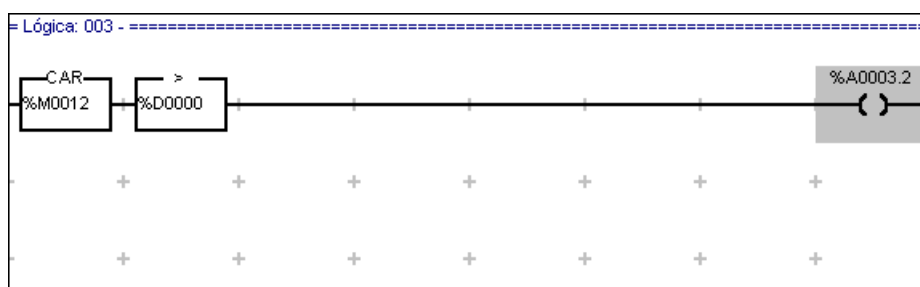


Figura 3-6 Exemplo das Instruções de Comparação

Como os tipos dos operandos são diferentes (%M e %D), a comparação é realizada ponto a ponto, sem considerar os sinais aritméticos. Devido a este fato, se %M0012 possuir valor -45 e %D0010

possuir valor +21, o operando %A0003.2 será energizado, como se o valor de %M0012 fosse maior que %D0010, o que não ocorre na realidade.

%M0012	= -45	1111	1111	1101	0011
%D0000	= +21	0000	0000	0000	0001

Para considerar os sinais na comparação do exemplo, deve-se converter o valor do operando memória para um decimal, utilizando este último na instrução CAR, como mostrado na lógica a seguir:

O valor 1111 1111 1101 0011 (%M0012) é maior que 10 0001 (%D0010) na comparação ponto a ponto, mesmo representando um valor negativo.

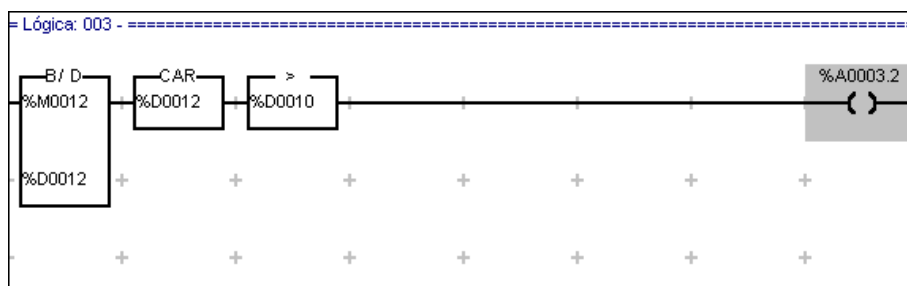


Figura 3-7 Exemplo das Instruções de Comparação

ATENÇÃO:

Devido à ordem de processamento das instruções na lógica, deve-se cuidar o posicionamento das instruções de comparação para evitar erros na interpretação no seu funcionamento. Ver seção **Lógicas** neste mesmo capítulo e o exemplo a seguir.

Exemplo:

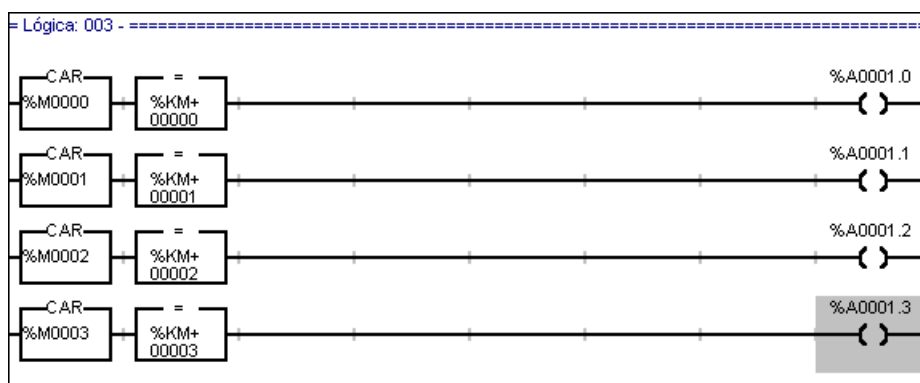


Figura 3-8 Utilização Incorreta da Instrução CAR

Na lógica apresentada, deseja-se comparar os valores dos operandos %M0000, %M0001, %M0002 e %M0003 com as constantes %KM00000, %KM00001, %KM00002 e %KM00003, respectivamente. Entretanto, o funcionamento ocorre de forma diversa do que o aspecto visual sugere. Como o processamento da lógica ocorre em colunas, no final da execução da coluna 0 estará carregado o

valor de %M0003 para as comparações na coluna 1. Na realidade, somente o valor do operando %M0003 será comparado com as constantes presentes na coluna 1.

Para o funcionamento desejado, a lógica deve ser programada da seguinte forma:

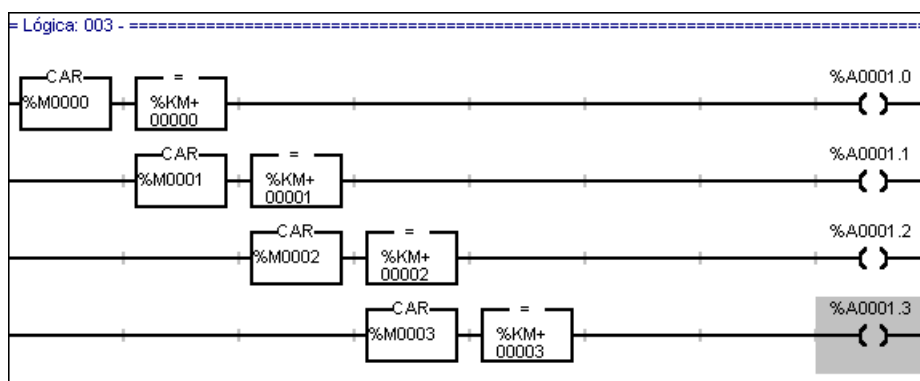


Figura 3-9 Utilização Correta da Instrução CAR

ATENÇÃO:

Para evitar interpretações errôneas no funcionamento das instruções de comparação, sugere-se o uso de apenas uma instrução CAR por coluna da lógica.

Sintaxe:

OPER1
%E
%S
%A
%M
%D
%F
%I
%KM
%KD
%KF
%KI
%M*E
%M*S
%M*A
%M*M
%M*D
%M*F
%M*I

Tabela 3-27 Sintaxe das Instruções Maior, Igual e Menor

Instruções do Grupo Contadores

As instruções contadoras são utilizadas para realizar contagens de eventos ou de tempo no programa aplicativo.





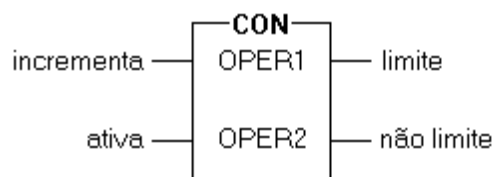
Nome	Descrição do Nome	Seqüência de Edição	Barra de Ferramentas
CON	contador simples	ALT, C, N	
COB	contador bidirecional	ALT, C, B	
TEE	temporizador na energização	ALT, C, T	
TED	temporizador na desenergização	ALT, C, D	

Tabela 3-28 Instruções do Grupo Contadores

CON - Contador Simples



OPER1 - contador

OPER2 - limite de contagem

Descrição:

Esta instrução realiza contagens simples, com o incremento de uma unidade em cada acionamento.

A instrução contador simples possui dois operandos. O primeiro, sempre do tipo %M, especifica a memória que contabiliza os eventos. O segundo estabelece o valor limite de contagem para energização da saída da célula superior e pode ser do tipo %KM, %M ou operando %M referenciado indiretamente.

Se a entrada **ativa** esta desenergizada, a memória em OPER1 é zerada, a saída **não limite** energizada e a saída **limite** desenergizada.

Quando a entrada **ativa** está energizada, cada transição de ligação na entrada **incrementa** aumenta o valor do operando contador (OPER1) de uma unidade.

Se o valor do primeiro operando igualar-se ao do segundo operando, a saída **limite** é energizada. A variável contadora não é incrementada com novas transições na entrada **incrementa**, permanecendo com o valor limite. Se for menor, a saída **limite** é desenergizada. O estado lógico da saída **não limite** é exatamente o oposto da saída **limite**, mesmo estando a instrução desativada.

Em caso de acesso indireto inválido para o segundo operando da instrução, a saída não limite é energizada.

ATENÇÃO:

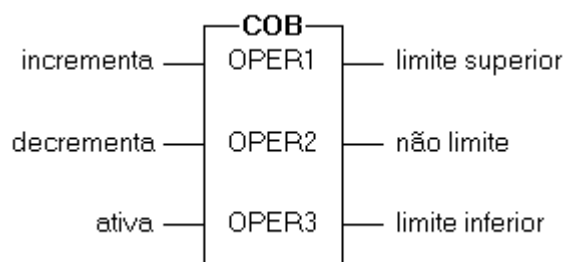
Com a entrada **ativa** desativada, a saída **não limite** permanece sempre energizada, mesmo quando a instrução estiver em um trecho comandado pela instrução RM (relé mestre). Deve-se ter cuidado para não realizar acionamentos indesejáveis na lógica devido a este fato.

Sintaxe:

OPER1	OPER2
%M	%KM %M %M*M

Tabela 3-29 Sintaxe da Instrução CON

COB - Contador Bidirecional



OPER1 - contador

OPER2 - passo de contagem

OPER3 - limite de contagem

Descrição:

Esta instrução realiza contagens com o valor de incremento ou decremento definido por um operando. A instrução contador bidirecional permite contagens em ambos os sentidos, isto é, incrementa ou decrementa o conteúdo de um operando do tipo memória.

O primeiro operando contém a memória acumuladora do valor contado, enquanto que o segundo especifica o valor do incremento ou decremento desejado. O terceiro operando contém o valor limite da contagem.

A contagem ocorre sempre que a entrada **ativa** está energizada e as entradas **incrementa** ou **decrementa** sofrerem uma transição de desligadas para ligadas. Se ambas as entradas sofrem a transição no mesmo ciclo de varredura do programa, não há incremento nem decremento no valor da memória declarada em OPER1.

Caso o valor do incremento seja negativo, a entrada **incrementa** provoca decrementos e a entrada **decremento** provoca incrementos no valor da contagem.

Se o valor do primeiro operando tornar-se maior ou igual ao do terceiro operando, a saída **limite superior** é energizada, não havendo incremento.

Se o valor do primeiro operando tornar-se igual ou inferior a zero, a saída **limite inferior** é acionada, sendo armazenado zero no primeiro operando.

Se o valor do primeiro operando está entre zero e o limite, a saída **não limite** é acionada. Se a entrada **ativa** não está energizada, a saída **limite inferior** é energizada e o primeiro operando é zerado.

Em caso de acesso indireto inválido para qualquer um dos operandos da instrução, a saída limite inferior é energizada.

ATENÇÃO:

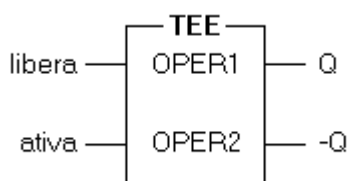
Com a entrada **ativa** desativada, a saída **limite inferior** permanece sempre energizada, mesmo quando a instrução estiver em um trecho comandado pela instrução RM (relé mestre). Deve-se ter cuidado para não realizar acionamentos indesejáveis na lógica devido a este fato.

Sintaxe:

OPER1	OPER2	OPER3
% M	% M	% M
% M*M	% M*M	% M*M
	% KM	% KM

Tabela 3-30 Sintaxe da Instrução COB

TEE - Temporizador na Energização



OPER1 - acumulador de tempo

OPER2 - limite de tempo (décimos de segundos)

Descrição:

Esta instrução realiza contagens de tempo com a energização das suas entradas de acionamento.

A instrução TEE possui dois operandos. O primeiro (OPER1) especifica a memória acumuladora da contagem de tempo. O segundo operando (OPER2) indica o tempo máximo a ser acumulado. A contagem de tempo é realizada em décimos de segundos, ou seja, cada unidade incrementada em OPER1 corresponde a 0,1 segundo.

Enquanto as entradas **libera** e **ativa** estiverem simultaneamente energizadas, o operando OPER1 é incrementado a cada décimo de segundo. Quando OPER1 for maior ou igual a OPER2, a saída **Q** é energizada e **-Q** desenergizada, permanecendo OPER1 com o mesmo valor de OPER2.

Desacionando-se a entrada **libera**, há a interrupção na contagem do tempo, permanecendo OPER1 com o mesmo valor. Desacionando-se a entrada **ativa**, o valor em OPER1 é zerado.

Se OPER2 for negativo ou o acesso indireto for inválido, OPER1 é zerado e a saída **-Q** é energizada.

O estado lógico da saída **Q** é exatamente o oposto da saída **-Q**, mesmo estando a instrução desativada.

ATENÇÃO:

Com a entrada **ativa** desativada, a saída **-Q** permanece sempre energizada, mesmo quando a instrução estiver em um trecho comandado pela instrução RM (relé mestre). Deve-se ter cuidado para não realizar acionamentos indesejáveis na lógica devido a este fato.

Diagrama de Tempos:

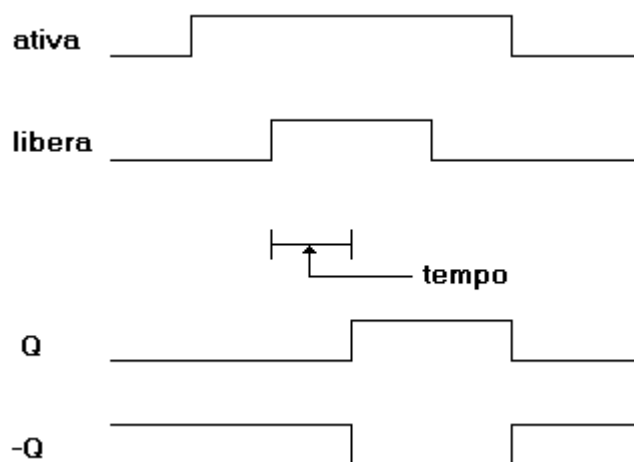


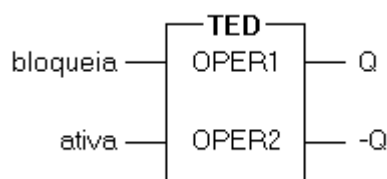
Figura 3-10 Diagrama de Tempos da Instrução TEE

Sintaxe:

OPER1	OPER2
%M	%M %M*M %KM

Tabela 3-31 Sintaxe da Instrução TEE

TED - Temporizador na Desenergização



OPER1 - acumulador de tempo

OPER2 - limite de tempo (décimos de segundo)

Descrição:

Esta instrução realiza contagens de tempo com a desenergização da sua entrada de acionamento.

A instrução TED possui dois operandos. O primeiro (OPER1) especifica a memória acumuladora da contagem de tempo. O segundo operando (OPER2) indica o tempo máximo a ser acumulado. A contagem de tempo é realizada em décimos de segundos, ou seja, cada unidade incrementada em OPER1 corresponde a 0,1 segundo.

Enquanto a entrada **ativa** estiver energizada e a entrada **bloqueia** desenergizada, o operando OPER1 é incrementado a cada décimo de segundo. Quando OPER1 for maior ou igual a OPER2, a saída **Q** é desenergizada e **-Q** energizada, permanecendo OPER1 com o mesmo valor de OPER2.

A saída **Q** fica energizada sempre que a entrada **ativa** estiver energizada e OPER1 for menor do que OPER2. Acionando-se a entrada **bloqueia**, há a interrupção na contagem do tempo, enquanto que desacionando a entrada **ativa**, o tempo do acumulador é zerado e a saída **Q** é desacionada.

Se OPER2 for negativo ou o acesso indireto for inválido, OPER1 é zerado e a saída **Q** é energizada. O estado lógico da saída **-Q** é exatamente o oposto da saída **Q**, mesmo estando a instrução desativada.

ATENÇÃO:

Com a entrada **ativa** desativada, a saída **-Q** permanece sempre energizada, mesmo quando a instrução estiver em um trecho comandado pela instrução RM (relé mestre). Deve-se ter cuidado para não realizar acionamentos indesejáveis na lógica devido a este fato.

Diagrama de Tempos:

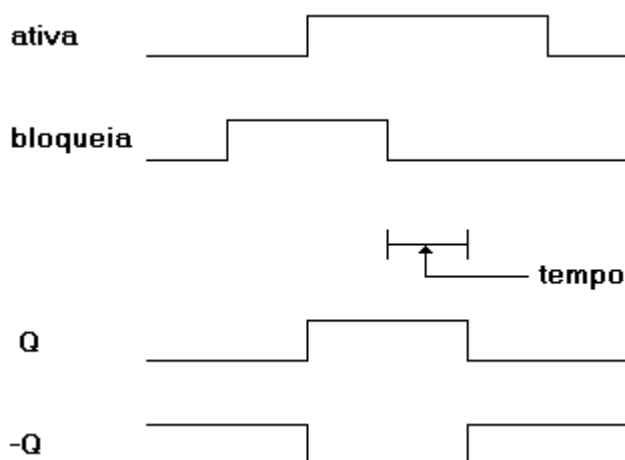


Figura 3-11 Diagrama de Tempos da Instrução TED

Sintaxe:

OPER1	OPER2
%M	%M %M*M %KM

Tabela 3-32 Sintaxe da Instrução TED

Instruções do Grupo Conversores

Esse grupo possui instruções que permitem a conversão entre os formatos de armazenamento dos valores utilizados nos operandos do programa aplicativo e acessos a módulos analógicos no barramento de entrada e saída.





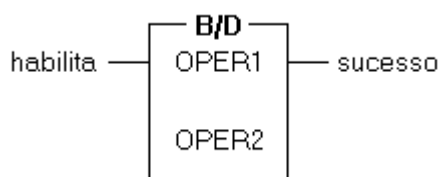
Nome	Descrição do Nome	Seqüência de Edição	Barra de Ferramentas
<u>B</u> IN/DEC	conversão binário-decimal	ALT, V, B	
<u>D</u> EC/BIN	conversão decimal-binário	ALT, V, D	
<u>A</u> NA/DIG	conversão analógico-digital	ALT, V, A	
<u>D</u> IG/ANA	conversão digital-analógico	ALT, V, G	

Tabela 3-33 Instruções do Grupo Conversores

B/D - Conversão Binário-Decimal



OPER1 - origem

OPER2 - destino

Descrição:

Esta instrução converte valores armazenados em formato binário, contidos em operandos memória (%M), para o formato decimal (BCD), armazenando-os em operandos decimais (%D).

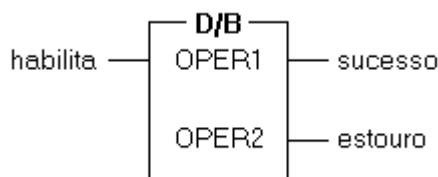
O valor binário contido no primeiro operando (OPER1) é convertido para valor decimal e armazenado no segundo operando (OPER2). A saída **sucesso** é acionada se a conversão for corretamente realizada. Se algum acesso indireto inválido a operando ocorrer, a saída **sucesso** não é energizada.

Sintaxe:

OPER1	OPER2
%M	%D
%M*M	%M*D

Tabela 3-34 Sintaxe da Instrução B/D

D/B - Conversão Decimal-Binário



OPER1 - origem

OPER2 - destino

Descrição:

Esta instrução converte valores armazenados em formato decimal, contidos em operandos decimais (%D), para o formato binário, armazenando-os em operandos memórias (%M).

O valor decimal contido no primeiro operando (OPER1) é convertido para valor binário e armazenado no segundo operando (OPER2). A saída **sucesso** é acionada se a conversão for corretamente realizada. Se algum acesso indireto inválido a operando ocorrer, a saída **sucesso** não é energizada.

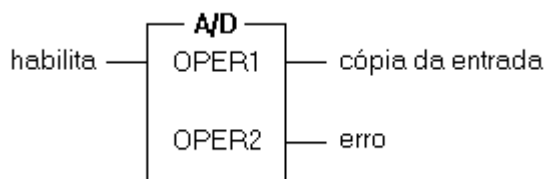
Caso o valor convertido resultar em um valor maior do que os máximos armazenáveis em operandos %M, a saída **sucesso** não é energizada, sendo armazenado o valor limite no operando destino. Neste caso, a saída **estouro** é energizada.

Sintaxe:

OPER1	OPER2
%D	%M
%M*D	%M*M

Tabela 3-35 Sintaxe da Instrução D/B

A/D - Conversão Analógico-Digital



OPER1 - endereço do módulo no barramento / número de canais a converter

OPER2 - primeiro operando a receber o valor convertido

Descrição:

Esta instrução converte os valores lidos de um módulo de entrada analógica para valores numéricos armazenados em operandos.

É possível efetuar a leitura de 1 ou 8 canais alterando-se apenas a especificação do primeiro operando, o qual indica o endereço no barramento ocupado pelo módulo A/D. Este módulo deve estar especificado na declaração do barramento, realizada no MasterTool. O endereço a ser programado em OPER1 pode ser obtido diretamente no MasterTool. Os valores convertidos são colocados em operandos do tipo memória, definidos em OPER2.

A conversão é realizada apenas se a entrada **habilita** estiver energizada.

Caso OPER1 seja especificado com subdivisão do tipo ponto (%RXXXX.X), a conversão é realizada somente para o canal do módulo relativo ao ponto. Os pontos .0 a .7 do operando correspondem aos canais 0 a 7 do módulo, respectivamente. Neste formato, o tempo de execução da instrução é significativamente menor do que a conversão dos 8 canais, sendo adequada, por exemplo, para uso em módulos de programa E018, acionados por interrupção de tempo.

Se OPER1 for especificado como %RXXXX (conversão de 8 canais), os valores convertidos são colocados na memória declarada em OPER2 e nas 7 memórias subsequentes.

Se OPER1 for especificado como %RXXXX.X (conversão de 1 canal), o valor já convertido é colocado na memória declarada em OPER2.

Os módulos disponíveis para realizar a conversão A/D são mostrados a seguir. Os valores convertidos pela instrução pertencem a uma faixa relacionada com a característica de cada módulo:

- AL-1103 (10 bits) : valores de 0000 a 1023
- AL-1116 (12 bits) : valores de 0000 a 4095
- AL-1119 e QK1119 (12 bits) : valores de 0000 a 4095
- QK1136 (12 bits) : valores de 0000 a 3999, com indicação de overflow (4000 a 4095)
- AL-1139 e QK1139 (12 bits) : valores de 0000 a 3999, com indicação de overflow (4000 a 4095)

A saída de erro da instrução é ativada em alguma das seguintes situações:

- módulo declarado no barramento é inválido para a instrução (não é um dos módulos relacionados anteriormente)
- tentativa de acesso a operandos não declarados
- erro de conversão (exceto AL-1103)

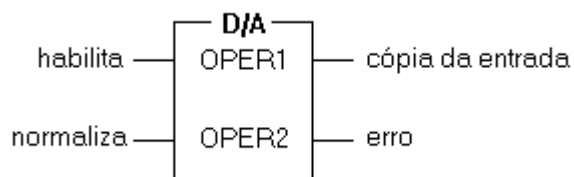
Sintaxe:

OPER1	OPER2
%RXXXX %RXXXX.X	%M

Tabela 3-36 Sintaxe da Instrução A/D

Esta instrução não está disponível para as UCPs AL-3003, AL-3004, PL101, PL102, PL103, PL104 e PL105.

D/A - Conversão Digital-Analógico



OPER1 - primeiro operando com os valores a serem convertidos

OPER2 - endereço do módulo no barramento / número de canais a converter

Descrição:

Esta instrução converte os valores numéricos de memórias para sinais analógicos. Os valores são convertidos através dos cartões de saída analógica AL-1203, AL-1214, AL-1222 ou QK1222 sendo possível a conversão de 1 ou 4 canais utilizando-se apenas uma instrução D/A.

O primeiro operando especifica a primeira memória com o valor a converter.

O segundo operando indica o endereço do módulo D/A no barramento de módulos. O módulo deve estar especificado na declaração do barramento, realizada no MasterTool. O endereço a ser programado em OPER2 pode ser obtido diretamente através do MasterTool.

A conversão é efetuada apenas se a entrada **habilita** estiver energizada.

Caso OPER2 seja especificado com subdivisão do tipo ponto (%RXXXX.X), a conversão é realizada do operando declarado em OPER1 para o canal do módulo correspondente ao ponto. Os pontos .0 a .3 do operando correspondem aos canais 0 a 3 do módulo, respectivamente.

Se OPER2 for especificado como %RXXXX (conversão de 4 canais), os valores a serem convertidos são obtidos da memória declarada em OPER1 e das 3 subseqüentes.

Os módulos disponíveis para realizar a conversão D/A são mostrados a seguir. Os valores convertidos pela instrução pertencem a uma faixa relacionada com a característica de cada módulo:

Saída em Tensão

Módulo	Resolução	Normalização	Faixa
AL-1203	10 bits	não utilizada	0000 a 1000
AL-1214	10 bits	não utilizada	0000 a 1000
AL-1222, QK1222	12 bits	desligada	0000 a 4000
AL-1222, QK1222	12 bits	ligada	-2000 a +2000

Tabela 3-37 Instrução D/A - Saída em Tensão

Saída em Corrente

Módulo	Resolução	Normalização	Faixa
AL-1203	10 bits	não utilizada	0000 a 1000
AL-1214	10 bits	não utilizada	0000 a 1000
AL-1222, QK1222	11 bits	desligada	0000 a 2000

Tabela 3-38 Instrução D/A - Saída em Corrente

Os valores convertidos dos AL-1222 e QK1222 dependem ainda da entrada **normaliza**, que converte valores simétricos quando energizada. Isto torna-se útil quando se necessita trabalhar com valores negativos, por exemplo na faixa de tensão de $\pm 10V$.

Não existe normalização para os módulos AL-1203 e AL-1214, apenas para os AL-1222 e QK1222. Entretanto, a normalização somente é possível para operação em modo tensão.

ATENÇÃO:

Em modo corrente a entrada **normaliza** não deve ser energizada.

Os AL-1222 e QK1222 podem trabalhar com as 4 saídas em modo tensão ou modo corrente, ou os dois modos simultaneamente. A seleção do modo de operação é feita pelo usuário através da programação do endereçamento do módulo em OPER2:

- Se %RXXXX for par, converte corrente
- Se %RXXXX for ímpar, converte tensão

Exemplo:

Caso o módulo seja colocado no endereço %R0024 do barramento, se a instrução for programada com %R0024, os AL-1222 e QK1222 irão operar em modo corrente. Se for programada com %R0025, irá operar em modo tensão.

ATENÇÃO:

A instrução não pode ser saltada durante a execução do programa aplicativo sob pena dos valores amostrados estarem incorretos.

A saída de erro da instrução é ativada em alguma das seguintes situações:

- módulo declarado no barramento é inválido para a instrução (não é um dos módulos relacionados anteriormente)
- tentativa de acesso a operandos não declarados

Sintaxe:

OPER1	OPER2
%M	%RXXXX %RXXXX.X

Tabela 3-39 Sintaxe da Instrução D/A

Esta instrução não está disponível para as UCPs AL-3003, AL-3004, PL101, PL102, PL103, PL104 e PL105.

Instruções do Grupo Geral

As instruções do grupo gerais permitem teste e acionamentos de pontos indiretamente, implementações de máquinas de estado, chamadas de procedimentos e funções e escrita e leitura de operandos na rede ALNET II.









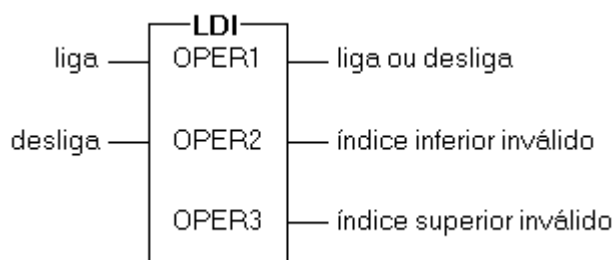
Nome	Descrição do Nome	Seqüência de Edição	Barra de Ferramentas
<u>LDI</u>	liga ou desliga pontos indexados	ALT, G, L	
<u>TEI</u>	teste de estado de pontos indexados	ALT, G, T	
<u>SEQ</u>	seqüenciador	ALT, G, S	
<u>CHP</u>	chama módulo procedimento	ALT, G, P	
<u>CHF</u>	chama módulo função	ALT, G, F	
<u>ECR</u>	escrita de operandos em outro CP	ALT, G, E	
<u>LTR</u>	leitura de operandos de outro CP	ALT, G, T, T	
<u>LAI</u>	libera atualização de imagens	ALT, G, I	

Tabela 3-40 Instruções do Grupo Geral

LDI - Liga/Desliga Indexado



OPER1 - endereço do ponto a ser ligado ou desligado

OPER2 - endereço limite inferior

OPER3 - endereço limite superior

Descrição:

Esta instrução é utilizada para ligar ou desligar pontos indexados por uma memória, delimitados por operandos de limite inferior e superior.

O primeiro operando especifica a memória cujo conteúdo referencia o operando auxiliar, entrada ou saída a ser ligado ou desligado. Deve ser declarado como operando de acesso indireto a operando %E ou %A (%MXXXX*E ou %MXXXX*A). Mesmo quando a instrução for utilizada para ligar ou desligar pontos de saída (%S), a representação deste operando será como acesso indireto a entrada (%MXXXX*E).

O segundo operando especifica o endereço do primeiro relé de saída ou auxiliar válido na instrução. Deve ser especificado com subdivisão de ponto (%RXXXX.X, %SXXXX.X ou %AXXXX.X).

O terceiro operando especifica o endereço do último relé de saída ou auxiliar válido na instrução. Deve ser especificado com subdivisão de ponto (%EXXXX.X, %SXXXX.X ou %AXXXX.X).

Se as entradas **liga** ou **desliga** forem acionadas, o ponto especificado pelo valor contido no operando memória (OPER1) é ligado ou desligado se estiver dentro da área de endereços limitada por OPER2 e OPER3. Por exemplo, se estes operandos correspondem a %S0003.3 e %S0004.5, respectivamente, esta instrução só atua para os elementos de %S0003.3 a %S0003.7 e de %S0004.0 a %S0004.5.

Se o relé ou auxiliar apontado pela memória índice estiver fora dos limites definidos pelos parâmetros da segunda e terceira células, a saída **índice superior inválido** ou **índice inferior inválido** é ligada. A saída da primeira célula é acionada se qualquer uma das entradas **liga** ou **desliga** é energizada e o acesso for corretamente realizado.

Caso as entradas permaneçam desacionadas, todas as saídas da instrução permanecem desenergizadas.

Se ambas as entradas forem energizadas simultaneamente, nenhuma operação é realizada, e todas as saídas da instrução são desenergizadas.

Em OPER1 deve ser carregado um valor que especifique o ponto desejado para ligar ou desligar, de acordo com a seguinte fórmula:

$$\text{VALOR OPER1} = (\text{OCTETO} * 8) + \text{PONTO}$$

Exemplo:

Por exemplo, se S0010.5 é o ponto que se deseja ligar indiretamente, então:

$$\text{OCTETO} = 10$$

$$\text{PONTO} = 5$$

VALOR OPER1 = $(10 * 8) + 5 = 85$

O valor a ser carregado em OPER1 é 85.

ATENÇÃO:

Esta instrução permite ligar ou desligar indiretamente pontos de operandos %E, sobrepondo o valor da varredura dos módulos de entrada após a sua execução.

Sintaxe:

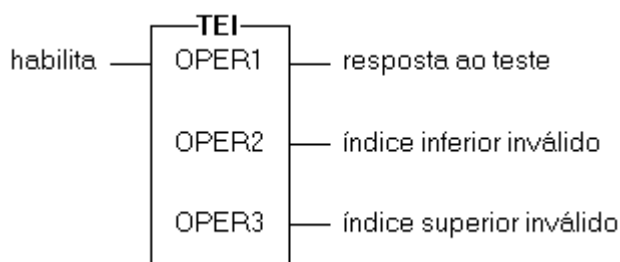
OPER1	OPER2	OPER3
%M*E	%E	%E

OPER1	OPER2	OPER3
%M*S	%S	%S

OPER1	OPER2	OPER3
%M*A	%A	%A

Tabela 3-41 Sintaxes da Instrução LDI

TEI - Teste de Estado Indexado



OPER1 - endereço do ponto a ser testado

OPER2 - endereço limite inferior

OPER3 - endereço limite superior

Descrição:

Esta instrução é utilizada para testar o estado de pontos indexados por uma memória, delimitados por operandos de limite inferior e superior.

O primeiro operando especifica a memória cujo conteúdo referencia o operando auxiliar ou relé de saída a ser testado. Deve ser declarado como operando de acesso indireto a operando %E ou %A (%MXXXX*E ou %MXXXX*A). Mesmo quando a instrução for utilizada para testar pontos de saída (%S), a representação deste operando será como acesso indireto a entrada (%MXXXX*E).

O segundo operando especifica o endereço do primeiro relé de saída ou auxiliar válido na instrução. Deve ser especificado com subdivisão de ponto (%EXXXX.X, %SXXXX.X ou %AXXXX.X).

O terceiro operando especifica o endereço do último relé de saída ou auxiliar válido na instrução. Deve ser especificado com subdivisão de ponto (%EXXXX.X, %SXXXX.X ou %AXXXX.X).

Se a entrada **habilita** estiver energizada, o estado do relé ou auxiliar especificado pelo valor contido na memória índice (OPER1) é examinado. Conforme esteja em 1 ou 0, a saída **resposta** é ligada ou não.

O ponto indexado pela memória é testado se estiver dentro da área de endereços limitada por OPER2 e OPER3. Por exemplo, se estes operandos correspondem a %S0003.3 e %S0004.5, respectivamente, esta instrução só atua para os elementos de %S0003.3 a %S0003.7 e de %S0004.0 a %S0004.5.

Se o relé ou auxiliar apontado pela memória índice estiver fora dos limites definidos pelos parâmetros da segunda e terceira células, a saída **índice superior inválido** ou **índice inferior inválido** é ligada e a saída da primeira célula desligada. Esta verificação é feita somente no momento em que a entrada **habilita** é energizada.

O cálculo do valor a ser armazenado no primeiro operando, para referência do ponto desejado, é o mesmo especificado na instrução **LDI**.

Sintaxe:

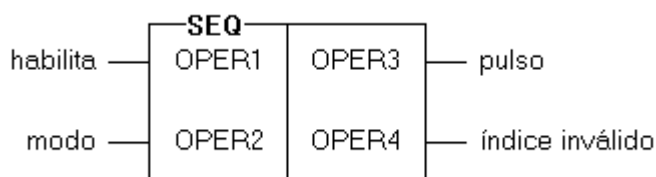
OPER1	OPER2	OPER3
% M *E	%E	%E

OPER1	OPER2	OPER3
% M *S	%S	%S

OPER1	OPER2	OPER3
% M *A	%A	%A

Tabela 3-42 Sintaxes da Instrução TEI

SEQ - Seqüenciador



OPER1 - tabela de condições ou primeira tabela de estados

OPER2 - índice da(s) tabela(s) (estado atual)

OPER3 - operando base da primeira série de condições

OPER4 - operando base da segunda série de condições

Descrição:

Esta instrução permite a programação de seqüenciamentos complexos com condições de evolução específicas para cada estado. Sua forma de programação é semelhante a "máquinas de estado".

A instrução pode ser executada em dois modos: o modo 1000 e o modo 3000. Quando a entrada **modo** está desenergizada, a instrução é executada no modo 1000, e quando ela está energizada, a instrução é executada no modo 3000. No modo 3000 seqüenciamentos mais complexos podem ser programados.

Modo 1000:

Neste modo ocorre uma seqüência de evolução fixa dos estados. A evolução sempre ocorre do estado atual para o seguinte, e do último para o primeiro.

O primeiro operando especifica uma tabela onde cada posição contém o endereço de um ponto de operando auxiliar que é testado como condição de evolução para o próximo estado.

O segundo operando especifica uma memória que armazena o estado atual e serve de índice para a tabela especificada no primeiro operando.

O terceiro operando é irrelevante, porém deve ser especificado um operando do tipo memória ou auxiliar nesta célula, pois o MasterTool realiza a consistência conforme o modo 3000.

O quarto operando é irrelevante, porém deve ser especificado um operando do tipo memória ou auxiliar nesta célula, pois o MasterTool realiza a consistência conforme o modo 3000.

Quando a entrada **habilita** está desenergizada, as saídas **pulso** e **índice inválido** ficam desenergizadas, independente de qualquer outra condição. Quando a entrada **habilita** estiver energizada, a saída de **pulso** fica normalmente energizada, e a saída de **índice inválido** fica normalmente desenergizada.

Além disso, quando a entrada **habilita** está energizada, a posição da tabela (OPER1) indexada pelo estado atual (OPER2) é acessada e o ponto de operando auxiliar referenciado nesta posição da tabela é examinado. Se este ponto estiver energizado, o conteúdo de OPER2 é incrementado (ou zerado, se estiver apontando para a última posição da tabela OPER1) e na saída **pulso** ocorre um pulso de desenergização com duração de um ciclo de programa. Se o ponto examinado estiver desenergizado nada ocorre e o valor da memória em OPER2 permanece inalterado.

A saída **índice inválido** é ativada se a memória OPER2 (estado atual) contiver um valor que indexa uma posição não existente na tabela especificada em OPER1. Isto pode ocorrer modificando-se a memória OPER2 em um ponto do programa aplicativo fora da instrução SEQ (na inicialização de OPER2, por exemplo). Deve-se ter o cuidado de definir e inicializar a tabela especificada em OPER1 com valores legais.

Na tabela especificada em OPER1 devem ser carregados valores em formato decimal que especifiquem pontos de operandos auxiliares que devem ser testados como condições de evolução. O cálculo destes valores é especificado pela equação:

$$\text{VALOR} = (\text{endereço do operando} * 8) + \text{endereço da subdivisão}$$

Exemplo:

Se %A0030.2 é o ponto que se deseja usar como condição de evolução a partir do estado 4, então:

endereço do operando = 30

Endereço da subdivisão = 2

$$\text{VALOR} = (30 * 8) + 2 = 242$$

O valor a ser carregado na posição 4 da tabela OPER1 deve ser 242 para que o ponto %A0030.2 cause a evolução para o próximo estado, que é o estado 5 (ou o estado 0, se a tabela tiver 5 posições).

Modo 3000:

Neste modo é possível definir a sequência de evolução e escolher um entre dois caminhos a partir do estado atual. Portanto, 2 graus de liberdade a mais são oferecidos em relação ao modo 1000, permitindo-se implementar máquinas de estado bem mais complexas.

Existe, entretanto, menos liberdade para escolher as condições de evolução em relação ao modo 1000, além de ser necessário o uso de mais memória (tabelas) no modo 3000.

O primeiro operando especifica a primeira de duas tabelas subsequentes que são utilizadas pela instrução. As duas tabelas devem ter o mesmo tamanho. Cada posição da primeira tabela contém o próximo estado caso a condição associada ao operando 3 esteja energizada. Cada posição da segunda tabela contém o próximo estado caso a condição associada ao operando 4 esteja energizada.

O segundo operando especifica uma memória que indica qual o estado atual e serve de índice para as tabelas especificadas no primeiro operando.

O terceiro operando especifica um operando que serve de base para determinar a condição de evolução a partir do estado OPER2 para o estado indexado por OPER2 na primeira tabela.

O quarto operando especifica um operando que serve de base para determinar a condição de evolução a partir do estado OPER2 para o estado indexado por OPER2 na segunda tabela.

Quando a entrada **habilita** está desenergizada, as saídas **pulso** e **índice inválido** ficam desenergizadas, independente de qualquer outra condição. Quando a entrada **habilita** está energizada, a saída de **pulso** fica normalmente energizada, e a saída de **índice inválido** fica normalmente desenergizada.

Além disso, quando a entrada **habilita** está energizada, a instrução busca o valor da memória OPER2 (estado atual) e testa a respectiva condição de evolução com base em OPER3. Se esta condição estiver energizada, o operando OPER2 é carregado com um novo estado, indexado pelo próprio operando OPER2 na primeira tabela especificada por OPER1. Caso a condição de evolução associada a OPER2 e com base em OPER3 estiver desenergizada, testa-se a condição de evolução associada a OPER2 e com base em OPER4. Se esta última condição estiver energizada, o operando OPER2 é carregado com um novo estado, indexado pelo próprio operando OPER2 na segunda tabela especificada por OPER1. Se pelo menos uma das 2 condições acima estiver energizada, uma transição de estado ocorrerá, e um pulso de desenergização com duração de um ciclo de programa aplicativo ocorrerá na saída **pulso** da instrução. Se nenhuma das 2 condições estiver energizada, nada acontece, e o valor da memória OPER2 (estado atual) permanece inalterado, bem como a saída **pulso** continua energizada.

A saída **índice inválido** é ativada se a memória OPER2 contiver um valor que indexa uma posição não existente nas tabelas especificadas em OPER1. Isto pode ocorrer modificando-se a memória OPER2 em um ponto do programa aplicativo fora da instrução SEQ (na inicialização de OPER2, por exemplo) ou dentro da própria instrução SEQ, caso algumas das posições das tabelas especificadas em OPER1 contenham valores inválidos para serem o próximo estado. Deve-se ter o cuidado de definir as 2 tabelas especificadas por OPER1 com o mesmo tamanho, e deve-se inicializá-las com valores legais (exemplo: se as tabelas tiverem 10 posições, somente valores entre 0 e 9 devem ser carregados em posições desta tabela, pois somente estes podem ser estados legais).

As condições de evolução associadas ao estado atual (OPER2) são determinadas com base em OPER3 (próximo estado é carregado a partir da primeira tabela) ou com base em OPER4 (próximo estado é carregado a partir da segunda tabela). Sabendo-se que os operandos OPER3 e OPER4 são do tipo memória (16 bits) ou do tipo auxiliar (8 bits), suponha-se o seguinte:

ESTADO = conteúdo do operando OPER2 (estado atual)

END3 = endereço de OPER3

END4 = endereço de OPER4

END1 = endereço do ponto a ser testado, com base em OPER3

SUB1 = subdivisão do ponto a ser testado, com base em OPER3

END2 = endereço do ponto a ser testado, com base em OPER4

SUB2 = subdivisão do ponto a ser testado, com base em OPER4

Os pontos testados como condição de evolução associada a cada tabela serão:

M<END1>.<SUB1> ou A<END1>.<SUB1> (primeira tabela)

e M<END2>.<SUB2> ou A<END2>.<SUB2> (segunda tabela)

onde:

$END1 = END3 + ESTADO / 16$ (se operando %M)

$END1 = END3 + ESTADO / 8$ (se operando %A)

$SUB1 = RESTO (ESTADO / 16)$ (se operando %M)

$SUB1 = RESTO (ESTADO / 8)$ (se operando %A)

$END2 = END4 + ESTADO / 16$ (se operando %M)

$END2 = END4 + ESTADO / 8$ (se operando %A)

$SUB2 = RESTO (ESTADO / 16)$ (se operando %M)

$SUB2 = RESTO (ESTADO / 8)$ (se operando %A)

Exemplo:

Sejam:

OPER1 = %TM000

OPER2 = %M0010

OPER3 = %M0100

OPER4 = %A0020

Onde:

%TM000	Posição	Valor
	000	00001
	001	00002
	002	00004
	003	00001
	004	00000

%TM001	Posição	Valor
	000	00001
	001	00003
	002	00001
	003	00004
	004	00000

%M0010 = 00001

%M0100	XXXXXX
%M0101	XXXXXX
%M0102	XXXXXX
%M...	...

%A0020	XXXXXX
%A0021	XXXXXX
%A0022	XXXXXX
%A...	...

Então as condições de evolução a partir do estado 1 serão:

Para a primeira tabela:

- $100 + 1/16 = 100$
- $\text{resto}(1/16) = 1$
- ponto a ser testado = %M0100.1

Para a segunda tabela:

- $20 + 1/8 = 20$

- $\text{resto}(1/8) = 1$
- ponto a ser testado = %A0020.1

Baseado nas condições de %M0100.1 e %A0020.1 teremos, a partir de uma das tabelas, o novo estado do operando %M0010:

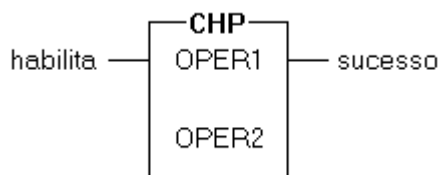
%M0100.1	%A0020.1	%M0010	Observação
0	0	00001	não há mudança de estado
0	1	00003	mudança de estado conforme %TM001
1	0	00002	mudança de estado conforme %TM000
1	1	00002	mudança de estado conforme %TM000 (OPER3 tem prioridade sobre OPER4)

Sintaxe:

OPER1	OPER2	OPER3	OPER4
%TM %M*TM	%M %M*M	%M %A	%M %A

Tabela 3-43 Sintaxe da Instrução SEQ

CHP - Chama Módulo Procedimento



OPER1 - nome do módulo a chamar

OPER2 - número do módulo a chamar

Descrição:

Esta instrução realiza o desvio do processamento do módulo corrente para módulo Procedimento especificado nos seus operandos, se o mesmo estiver presente no CP. Ao final da execução do módulo chamado, o processamento retorna para a instrução seguinte à CHP. Não há passagem de parâmetros para o módulo chamado.

O primeiro operando (OPER1) é documental e especifica o nome do módulo a ser chamado. O segundo operando (OPER2) especifica o número deste módulo, sendo implícito o fato do módulo chamado ser do tipo procedimento.

Caso o módulo chamado não exista, a saída **sucesso** é desenergizada e a execução continua normalmente após a instrução. O nome do módulo não é considerado pelo CP para a chamada mas apenas o seu número. Caso exista o módulo P com o mesmo número do módulo chamado, porém com nome diferente, este módulo mesmo assim é executado.

Ver seção **Utilização dos Módulos P e F** no capítulo 2 deste manual.

Exemplo:

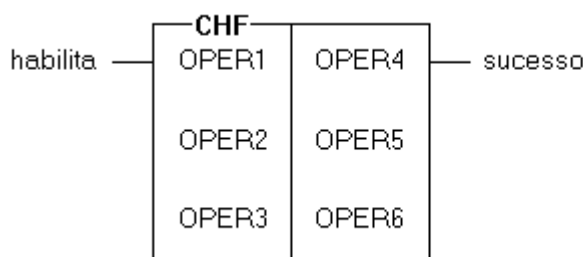


Sintaxe:

OPER1	OPER2
NOME	NÚMERO

Tabela 3-44 Sintaxe da Instrução CHP

CHF - Chama Módulo Função



- OPER1 - nome do módulo a chamar
- OPER2 - número de parâmetros a enviar
- OPER3 - número de parâmetros a retornar
- OPER4 - número do módulo a chamar
- OPER5 - lista dos parâmetros a enviar
- OPER6 - lista dos parâmetros a retornar

Descrição:

A instrução chama módulo função realiza o desvio do processamento do módulo corrente para o módulo especificado na mesma, se este estiver presente no CP. Ao final da execução do módulo chamado, o processamento retorna para a instrução seguinte à CHF.

Devem ser declarados o nome e o número do módulo como operandos OPER1 e OPER4, respectivamente, sendo implícito o fato do módulo chamado ser do tipo função. Caso o módulo chamado não exista no controlador, a execução continua normalmente após a instrução de chamada, com a saída sucesso da mesma desligada. O nome do módulo não é levado em consideração pelo CP, estando no programa aplicativo apenas como referência documental, sendo considerados para a chamada apenas o seu tipo e número. Caso exista o módulo F com o mesmo número do módulo chamado mas com nome diferente, este módulo é executado.

É possível a passagem de valores de operandos (parâmetros) para o módulo chamado e no sentido inverso, após a sua execução. Na quinta célula da instrução (OPER5) é especificada uma lista de operandos a serem enviados para o módulo chamado. Antes da execução do módulo, os valores destes operandos são copiados para os operandos especificados na lista de parâmetros de entrada do módulo F, declarados na opção **Parâmetros** do MasterTool quando o mesmo foi programado.

Após a execução do módulo F chamado, os valores dos operandos declarados na sua lista de parâmetros de saída (opção **Parâmetros** do MasterTool na sua programação) são copiados para os operandos declarados na lista de operandos a retornar da instrução CHF (OPER6). Finalizada a cópia de retorno, o processamento continua na instrução seguinte à de chamada.

ATENÇÃO:

O MasterTool não realiza nenhuma consistência em relação aos operandos programados como parâmetros, tanto na instrução CHF como no módulo F.

A lista de operandos a serem enviados para o módulo F deve conter o mesmo número de operandos com o mesmo tipo dos declarados como parâmetros de entrada do módulo, para que a cópia dos seus valores seja realizada corretamente. A cópia dos operandos é realizada na mesma ordem em que os mesmos estão dispostos nas listas. Caso uma das duas listas possua menos operandos em relação à outra, os valores dos operandos excedentes não serão copiados. Se os operandos possuírem tipos diferentes, a cópia dos valores será realizada com as mesmas regras usadas na instrução MOV (movimentação simples de operandos). Este princípio é válido também para as listas de parâmetros de retorno do módulo F.

A passagem de parâmetros é realizada com a cópia dos valores dos operandos declarados (passagem de parâmetros por valor), embora esses operandos ainda continuem de uso global, utilizáveis por qualquer módulo presente no CP. Os módulos F podem ser programados de forma genérica, para serem reutilizados em diversos programas aplicativos como novas instruções. É aconselhável que os mesmos empreguem operandos próprios, não usados por nenhum outro módulo presente no programa aplicativo, evitando alterações inadvertidas em operandos utilizados em outros módulos.

É possível a passagem de operandos simples e constantes para o módulo F. A passagem de tabelas como parâmetros não é permitida, devido ao grande tempo que seria necessário para a cópia do conteúdo das mesmas. Entretanto, o endereço de uma tabela pode ser passado para o módulo F contido em um operando memória e dentro deste módulo serem realizados acessos indiretos à mesma.

Não é permitida a passagem de operandos com subdivisão para o módulo F, tais como %M004.2, %A0021n1, etc. Devem ser usados somente operandos simples para esta passagem.

Para realizar a edição de parâmetros.

1. Declarar o número de parâmetros a enviar e retornar em OPER2 e OPER3, limitados em 10 para cada um (%KM+00000 a %KM+00010).
2. Selecionar o botão **Entrada**. É exibida a janela **CHF - Parâmetros de Entrada**.
3. Posicionar o cursor no índice a ser editado e digitar o endereço ou tag do operando desejado para aquela posição.
4. Repetir o passo 3 até que todos os operandos utilizados como parâmetros de entrada tenham sido editados.
5. Selecionar o botão **Ok**.
6. Para editar os parâmetros de saída da CHF, repetir o passo 2 selecionando o botão **Saída**, e após repetir os passos 3, 4 e 5.

	Operandos
0	%M0000
1	%M0000
2	%M0000
3	%M0000
4	%M0000
5	%M0000
6	%M0000
7	%M0000
8	%M0000
9	%M0000

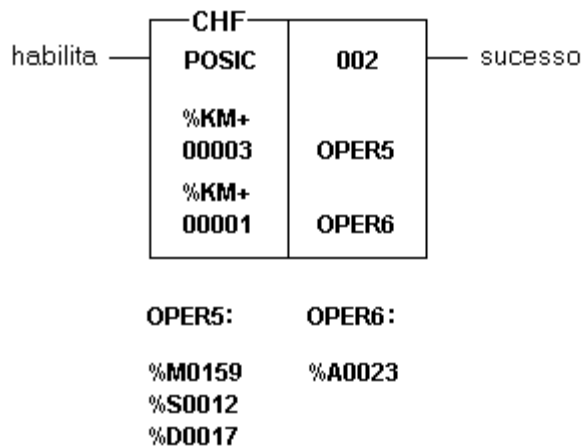
Digitar os operandos utilizados como parâmetros de entrada

Figura 3-12 Caixa de Diálogo CHF - Parâmetros de Entrada

Ver seção **Utilização dos Módulos P e F** no capítulo 2 deste manual.

Caso o valor de OPER2 ou OPER3 seja maior do que 10, o MasterTool considera tal valor como igual a 10 (%KM+00010).

Exemplo:

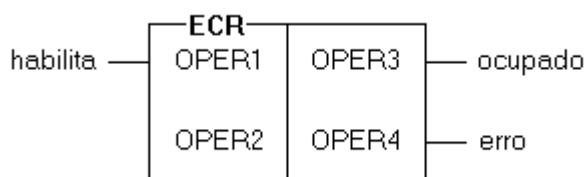


Sintaxe:

OPER1	OPER2	OPER3	OPER4	OPER5	OPER6
NOME	%KM	%KM	NÚMERO	%KM %KD %KF %TM %TD %TF %M %D %F %E %S %A %R	%KM %KD %KF %TM %TD %TF %M %D %F %E %S %A %R

Tabela 3-45 Sintaxe da Instrução CHF

ECR - Escrita de Operandos em Outro CP



OPER1 - endereço de nodo do controlador remoto

OPER2 - endereço da sub-rede do controlador remoto

OPER3 - operando de controle da instrução

OPER4 - janela de edição dos operandos

Descrição:

Esta instrução realiza a escrita de valores de operandos do controlador onde está sendo executada em operandos presentes em outros CPs, através da rede ALNET II de comunicação. Para o seu uso, portanto, é imprescindível que o controlador que a execute esteja conectado a outros CPs pela ALNET II.

Através da ECR podem ser transferidos valores de operandos individuais ou de conjuntos de operandos, sendo possível a programação de até 6 comunicações diferentes em uma mesma instrução.

A ECR pode ser programada para ser prioritária, enviando uma comunicação urgente, processada pelos "bridges" e pelo CP destino antes das comunicações comuns. A ECR prioritária permite somente uma comunicação, sendo útil para a sinalização de alarmes ou situações de emergência entre CPs.

Para programar a instrução, deve-se declarar na primeira e segunda células (OPER1 e OPER2) os endereços de nodo e sub-rede do controlador programável destino que irá receber os valores escritos. Estes operandos são programados como constantes do tipo memória (%KM) e possuem o mesmo significado dos endereços configurados nas opções **Comunicação, Endereço, Sub-rede e Comunicação, Endereço, Nó**.

A tabela 3-46 apresenta os valores possíveis para endereços de nó e sub-rede.

Sub-rede	Nó	Tipo de Comunicação
001 a 063	001 a 031	ALNET II com um nó
064	xxx	Endereço IP, onde xxx especifica outro endereço IP dentro da mesma subrede.
100	001 a 015	ALNET II com grupo de multicast em todas as sub-redes
101 a 163	001 a 015	ALNET II com grupo de multicast em uma sub-rede específica
200	xxx	ALNET II em broadcast para todas as sub-redes
201 a 263	xxx	ALNET II em broadcast para uma sub-rede específica

Tabela 3-46 Endereços de Nó e Sub-rede

O endereço de sub-rede entre os valores 001 e 063 indica que a comunicação é realizada utilizando a rede ALNET II e que é destinada a um único nó indicado na opção **Nó**.

O endereço de sub-rede 100 indica que a comunicação é realizada utilizando a rede ALNET II e destinada a todos os nós de todas as sub-redes que pertençam ao grupo de multicast especificado na opção **Nó** (multicast global).

O endereço de sub-rede entre 101 e 163 indica que a comunicação é realizada utilizando a rede ALNET II e que é destinada a todos os nós da sub-rede indicada na opção **Sub-rede menos 100** que pertençam ao grupo de multicast especificado na opção **Nó** (multicast local).

O endereço de sub-rede 200 indica que a comunicação é realizada utilizando a rede ALNET II e é destinado a todos os nós de todas as sub-redes (broadcast global). O valor contido na opção **Nó** não é relevante nesta opção.

O endereço de sub-rede entre 201 e 263 indica que a comunicação é realizada utilizando a rede ALNET II e destinada a todos os nós da sub-rede indicada na opção **Sub-rede menos 200** (broadcast local). O valor contido na opção **Nó** não é relevante nesta opção.

Na terceira célula (OPER3) deve ser declarado um operando decimal (%D) para ser utilizado pela própria instrução no controle do seu processamento.

ATENÇÃO:

O operando %D programado em OPER3 não pode ter o seu valor modificado em nenhum outro ponto do programa aplicativo para o correto funcionamento da ECR. Conseqüentemente, cada nova instrução ECR ou LTR inserida no programa aplicativo deve utilizar um operando %D diferente das demais. Este operando não pode ser retentivo.

Para realizar a edição dos parâmetros da ECR

1. Selecionar o botão **CP**. É exibida a caixa de diálogo **ECR - Parâmetros**.

ECR - Parâmetros			
<input type="checkbox"/> Mensagem Prioritária			
CP LOCAL			CP REMOTO
P/1		%M0000	a
2		LIMINF1	a
3			a
4			a
5			a
6			a

Figura 3-13 Caixa de Diálogo ECR - Parâmetros

Esta caixa de diálogo está dividida em duas partes: CP LOCAL e CP REMOTO, cada qual contendo três colunas. Nas três colunas que compõem o CP local pode-se definir o operando ou o grupo de operandos cujos valores serão enviados para o controlador programável destino. Nas colunas pertencentes ao CP remoto, são declarados os operandos que irão receber os valores no controlador destino, podendo ser de tipos diferentes do CP local. A caixa de diálogo possui seis linhas, permitindo que sejam definidas até seis comunicações diferentes na mesma instrução ECR para o mesmo controlador destino.

O item **Mensagem Prioritária** permite a edição de uma ECR prioritária quando estiver selecionado.

Na edição de uma ECR prioritária, somente a linha para a edição da comunicação, reconhecida pela inicial **P/1** é válida, enquanto que em uma ECR não prioritária as comunicações **P/1, 2, 3, 4, 5 e 6** são válidas. Se durante a troca entre ECR prioritária e não prioritária já existam operandos editados, a

comunicação de número **P/1** passa a ser a comunicação da ECR prioritária e vice-versa. Os demais operandos são editados da mesma maneira que em uma ECR não prioritária.

Os operandos especificados para o CP local são consistidos pelo MasterTool de acordo com as declarações constantes no módulo C presente no mesmo, por pertencerem ao programa aplicativo que está sendo editado. Os operandos declarados para o CP remoto não sofrem consistências quanto a tipo e endereços, por pertencerem a um programa aplicativo de outro controlador programável. Entretanto, o número de bytes ocupados pelo bloco de operandos declarados no CP local deve ser igual ao número de bytes ocupado pelos operandos do CP remoto em cada comunicação, para que a escrita seja realizada corretamente. O número máximo de bytes possível de ser ocupado por um bloco de operandos em cada comunicação é limitado em 220.

A seguir estão relacionados os tipos de operandos possíveis de serem programados para o CP local e remoto, com a disposição correta dos mesmos nas colunas de edição e os seus respectivos significados.

CP LOCAL ou CP REMOTO	Significado
%EXXXX	Operando individual %EXXXX
%SXXXX	Operando individual %SXXXX
%AXXXX	Operando individual %AXXXX
%MXXXX	Operando individual %MXXXX
%DXXXX	Operando individual %DXXXX
%FXXXX	Operando individual %FXXXX
%EXXXX .. %EYYYY	Grupo de operandos %EXXXX a %EYYYY
%SXXXX .. %SYYYY	Grupo de operandos %SXXXX a %SYYYY
%AXXXX .. %AYYYY	Grupo de operandos %AXXXX a %AYYYY
%MXXXX .. %MYYYY	Grupo de operandos %MXXXX a %MYYYY
%DXXXX .. %DYYYY	Grupo de operandos %DXXXX a %DYYYY
%FXXXX .. %FYYYY	Grupo de operandos %FXXXX a %FYYYY
%TMXXXX YYY	Tabela %TMXXXX posição YYY
%TDXXXX YYY	Tabela %TDXXXX posição YYY
%TFXXXX YYY	Tabela %TFXXXX posição YYY
%TMXXXX III .. FFF	Tabela %TMXXXX posições III a FFF
%TDXXXX III .. FFF	Tabela %TDXXXX posições III a FFF
%TFXXXX III .. FFF	Tabela %TFXXXX posições III a FFF

Tabela 3-47 Operandos para CP Local e Remoto em ECR

O MasterTool permite a livre edição dos operandos dentro de uma mesma linha, possibilitando a troca de colunas com o auxílio das teclas de setas de movimentação horizontais. As consistências são realizadas na tentativa de troca de linha (setas verticais) ou confirmação do conteúdo editado na janela com a tecla ENTER. Pode-se desistir das alterações realizadas acionando-se a tecla ESC, permanecendo a instrução com o conteúdo anterior à abertura da janela de edição.

A tabela seguinte mostra o número de octetos ocupado por cada tipo de operando possível de ser programado nas definições de escritas.

Operando	Número de bytes
%E	1
%S	1
%A	1
%M	2
%F	4
%D	4
%TM	2 por posição
%TD	4 por posição
%TF	4 por posição

Tabela 3-48 Ocupação em Bytes dos Operandos da ECR

O cálculo do número de bytes ocupado nas declarações de CP local e remoto é realizado multiplicando-se o número de operandos declarados pelo número de octetos do tipo correspondente. Na tabela a seguir são mostrados alguns exemplos.

CP LOCAL ou CP REMOTO	Cálculo	Bytes
%E0004	1 operando x 1 byte	1
%S0020	1 operando x 1 byte	1
%A0018	1 operando x 1 byte	1
%M0197	1 operando x 2 bytes	2
%D0037	1 operando x 4 bytes	4
%E0005 .. %E0008	4 operandos x 1 byte	4
%S0024 .. %S0031	8 operandos x 1 byte	8
%A0089 .. %A0090	2 operandos x 1 byte	2
%M0002 .. %M0040	39 operandos x 2 bytes	78
%D0009 .. %D0018	10 operandos x 4 bytes	40
%TM0031 101	1 posição x 2 bytes	2
%TD0002 043	1 posição x 4 bytes	4
%TM0000 000 .. 002	3 posições x 2 bytes	6
%TD0007 021 .. 025	5 posições x 4 bytes	20

Tabela 3-49 Exemplos de Ocupação em Bytes

Ao ser acionada a entrada **habilita**, é disparada a comunicação da primeira escrita presente na ECR, sendo energizada a saída **ocupado** da mesma. No momento em que esta comunicação for completada, a instrução dispara a próxima escrita, independentemente do estado da entrada de habilitação, repetindo este procedimento para as demais comunicações existentes nesta instrução. Ao final da última escrita, a saída **ocupado** da ECR é desenergizada, com o disparo de um pulso com duração de uma varredura na saída **erro** caso não tenha sido possível realizar alguma comunicação.

Nos seis primeiros nibbles do operando D programado em OPER3 são colocados os estados das seis comunicações da instrução. Os últimos dois nibbles são utilizados para o controle do seu processamento.

Operando %D programado em OPER3 nas instruções ECR e LTR

7	6	5	4	3	2	1	0
		1	2	3	4	5	6

Controle da Instrução Estado das Comunicações

Estado da comunicação 1 - Nibble 5
 Estado da comunicação 2 - Nibble 4
 Estado da comunicação 3 - Nibble 3
 Estado da comunicação 4 - Nibble 2
 Estado da comunicação 5 - Nibble 1
 Estado da comunicação 6 - Nibble 0

Figura 3-14 Operando de Controle da Instrução ECR e LTR

O estado da comunicação armazenado em cada nibble é codificado da seguinte forma:

- 0 - comunicação com sucesso
- 1 - operando não definido
- 2 - endereço do controlador local igual ao remoto (comunicação para o próprio CP)
- 3 - bloco de operando inválido
- 4 - tipo de operando inválido
- 5 - time-out de transmissão de pacote
- 6 - não há espaço na fila de espera de transmissão
- 7 - falta de buffer de transmissão
- 8 - time-out de requisição
- 9 - erro de hardware
- 10 - CP remoto protegido

Em resumo, ao ser disparada a execução de uma instrução ECR todas as comunicações existentes na mesma são realizadas, mesmo que a sua entrada de habilitação seja desenergizada. Quando todas as escritas forem completadas, a próxima instrução ECR ou LTR encontrada no programa aplicativo com a entrada **habilita** energizada torna-se ativa, começando a processar as suas comunicações.

ATENÇÃO:

O programa aplicativo não pode realizar saltos sobre a instrução ECR ativa ou deixar de executar o módulo que a contém, para assegurar o seu correto processamento.

Em um programa aplicativo sendo executado no CP, em um dado momento, somente uma instrução de acesso à rede ALNET II (ECR ou LTR) é considerada ativa, mesmo que existam várias instruções com entrada **habilita** acionadas. A saída ocupado determina qual a instrução ativa, podendo ser utilizada para sincronizar as comunicações com o programa aplicativo. Para evitar sobrecargas no tráfego de informações na rede, aconselha-se disparar as instruções ECR periodicamente, evitando mantê-las permanentemente habilitadas no programa aplicativo, se possível. Um procedimento recomendado é desligar a entrada **habilita** logo que a saída **ocupado** for energizada, evitando um novo disparo da instrução após seu término.

A ECR prioritária não segue a ordem de processamento das ECRs não prioritárias, sendo processada e transmitindo seus dados o mais rápido possível, ao ser habilitada. Por esse motivo uma ECR prioritária não deve ficar permanentemente habilitada, devendo ser disparada somente em situações de alarme ou periodicamente. Caso contrário, pode impedir que as demais ECRs do programa realizem as suas comunicações ou causar o esgotamento de buffers de recepção do CP destino.

Se a instrução for programada especificando-se o endereço de nodo igual ao endereço do próprio controlador que a executa (escrita de valores de si próprio), a saída erro é energizada.

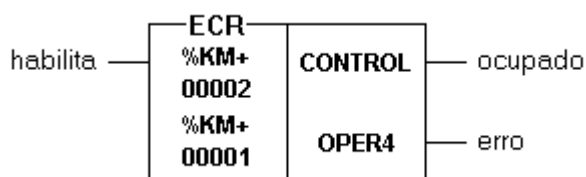
Caso nenhum operando tenha sido definido em OPER4, as saídas **erro** e **ocupado** ficam desenergizadas.

Sintaxe da Instrução:

OPER1	OPER2	OPER3	OPER4
%KM	%KM	%D	COMUNICAÇÕES

Tabela 3-50 Sintaxe da Instrução ECR

Exemplo:



Conteúdo da janela de edição em OPER4 da ECR não prioritária

COM	CP Local				CP Remoto			
1	%M0004				%A0014	..	%A0015	
2	%S0038 .. %S0041				%D0027			
3	%TD0007	028	..	030	%M0009	..	%M0014	
4	%M0006				%M0018			
5	%A0013 .. %A0020				%D0003	..	%D0004	
6	%TM0019	000	..	004	%TM0032	018	..	022

Esta instrução realiza escritas no controlador programável com o endereço de nó igual a 2 na sub-rede 1. São definidas seis comunicações para a mesma, transferindo dados de diversos tipos entre os CPs. A comunicação 0 envia o conteúdo de um operando memória no CP local para dois operandos auxiliares no CP remoto, sendo transferidos 2 octetos. As comunicações 1, 2, 3, 4 e 5 transferem, respectivamente, 4, 12, 2, 8 e 10 octetos entre os controladores.

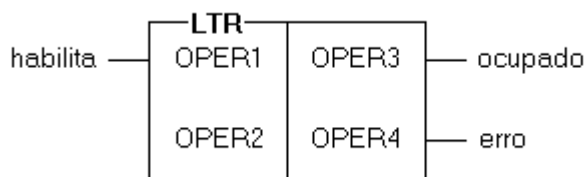
Conteúdo da janela de edição em OPER4 em uma ECR prioritária:

COM	CP Local	CP Remoto
P/1	%M0004	%A0014 .. %A0015

Esta instrução realiza escritas no controlador programável com o endereço de nó igual a 2 na sub-rede 1. É definida uma comunicação prioritária para a mesma. A comunicação P/1 envia o conteúdo de um operando memória no CP local para dois operandos auxiliares no CP remoto, sendo transferidos 2 octetos.

Esta instrução somente pode ser utilizada nas UCPs AL-2000/MSP, AL-2002/MSP, AL-2003, AL-2004 e QK2000/MSP.

LTR - Leitura de Operandos de Outro CP



OPER1 - endereço de nodo do controlador remoto
 OPER2 - endereço de sub-rede do controlador remoto
 OPER3 - operando de controle da instrução
 OPER4 - janela de janela de edição dos operandos

Descrição:

Esta instrução realiza a leitura de valores de operandos presentes em outros controladores programáveis para operandos do controlador programável onde está sendo executada, através da rede ALNET II de comunicação. Para o seu uso, portanto, é imprescindível que o CP que a execute esteja conectado a outros CPs pela ALNET II.

Através da LTR podem ser lidos valores de operandos individuais ou de conjuntos de operandos, sendo possível a programação de até 6 comunicações de leitura diferentes em uma mesma instrução.

A programação da instrução LTR é idêntica à ECR, observando as mesmas restrições. Na LTR, a transferência dos valores ocorre dos operandos declarados no CP remoto para o CP local, sendo esta a única diferença entre ambas.

ATENÇÃO:

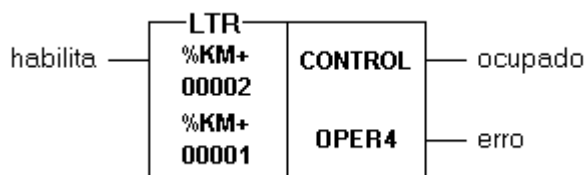
A instrução LTR difere da ECR quanto a possibilidade de mensagens prioritárias, ou seja, não é possível editar uma LTR prioritária.

Sintaxe da Instrução:

OPER1	OPER2	OPER3	OPER4
%KM	%KM	%D	COMUNICAÇÕES

Tabela 3-51 Sintaxe da Instrução LTR

Exemplo:



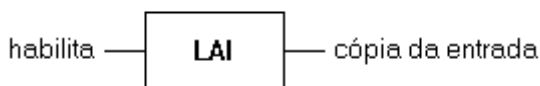
Conteúdo da janela de edição em OPER4 em uma LTR:

COM	CP Local				CP Remoto			
1	%M0004				%A0014	..	%A0015	
2	%S0038 .. %S0041				%D0027			
3	%TD0007	028	..	030	%M0009	..	%M0014	
4	%M0006				%M0018			
5	%A0013 .. %A0020				%D0003	..	%D0004	
6	%TM0019	000	..	004	%TM0032	018	..	022

Esta instrução realiza leituras no controlador programável com o endereço de nó igual a 2 na sub-rede 1. São definidas seis comunicações para a mesma, transferindo dados de diversos tipos entre os CPs. A comunicação 0 lê o conteúdo de dois operandos auxiliares no CP remoto para um operando memória no CP local, sendo transferidos 2 octetos. As comunicações 1, 2, 3, 4 e 5 transferem, respectivamente, 4, 12, 2, 8 e 10 octetos entre os controladores programáveis.

Esta instrução somente pode ser utilizada nas UCPs AL-2000/MSP, AL-2002/MSP, AL-2003, AL-2004 e QK2000/MSP.

LAI - Libera Atualização de Imagens dos Operandos



Descrição:

A instrução libera atualização da imagem de operandos realiza o processamento das comunicações pendentes da rede ALNET II para o CP local.

Ao retornar para o processamento do software executivo, ao final de cada varredura, o CP processa as requisições de leitura e outros serviços que tenham sido solicitados para o mesmo por outros CPs presentes na rede, durante a execução do programa aplicativo.

O controlador programável possui uma área de memória reservada para o armazenamento de até 32 comunicações recebidas durante o laço de execução do programa aplicativo, enquanto o software executivo não as processa. Caso o programa aplicativo possua tempo de execução relativamente alto e o controlador programável receba muitas requisições de serviços da rede, pode ocorrer a situação do CP não conseguir atendê-las, chegando ao limite de 32 comunicações pendentes à espera de processamento. Neste caso, o CP devolve uma resposta ao requisitante indicando a impossibilidade de atender a sua comunicação.

A instrução LAI executa o processamento de recepções e transmissões pendentes no CP, diminuindo a possibilidade de ocorrência da situação descrita anteriormente e reduzindo o tempo de atendimento às requisições. É recomendado o seu uso em programas aplicativos com alto tempo de ciclo, devendo ser inserida em pontos intermediários dos módulos, dividindo-os em trechos com aproximadamente 20 ms de tempo de execução.

ATENÇÃO:

Os valores dos operandos do programa aplicativo podem ser modificados após a execução de uma LAI, pois outro equipamento ligado à rede pode estar solicitando escritas nos mesmos. Deve-se considerar a influência deste fato ao se inserir esta instrução no programa aplicativo.

Esta instrução somente pode ser utilizada nas UCPs AL-2000/MSP, AL-2002/MSP, AL-2003, AL-2004 e QK2000/MSP.

Instruções do Grupo Ligações

As instruções do grupo ligações permitem a construção de caminhos em série e em paralelo bem como a inversão do sinal.




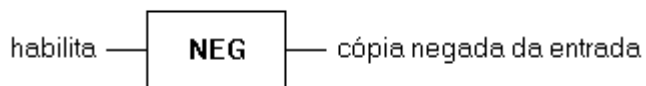
Nome	Descrição do Nome	Seqüência de Edição	Barra de Ferramentas
LGH	ligação horizontal	ALT, L, H	
LGN	ligação negada	ALT, L, N	
LGV	ligação vertical	ALT, L, V	

Tabela 3-52 Instruções do Grupo Ligações

LGH - Ligação Horizontal

LGN - Ligação Negada



LGV - Ligação Vertical

|

Descrição:

As ligações são elementos auxiliares na construção dos diagramas de relés, para interligar as demais instruções.

A ligação negada inverte na sua saída o estado lógico da sua entrada.

Referência dos Módulos Função

Este capítulo contém a descrição dos módulos Função (F) que acompanham o MasterTool, disponíveis para os controladores programáveis das séries AL-600, AL-2000, QUARK e PICCOLO.

Os módulos Função implementam diversas rotinas de uso específico ou para o acesso aos módulos especiais de E/S pelo programa aplicativo, sendo similares às instruções, porém carregados como módulos do programa. A sua execução é disparada por outros módulos através da instrução CHF.

Os módulos que acompanham o MasterTool são programados em linguagem de máquina, não podendo ser lidos para o programador e visualizados como os módulos em diagrama de relés. Devem ser carregados diretamente do disco para o CP (opções **Comunicação, Ler/Enviar Módulo**).

Cada modelo de UCP possui um conjunto de módulos F próprio, contidos em subdiretórios correspondentes no MasterTool. A tabela 4-1 mostra as funções existentes para cada UCP, bem como os módulos especiais de E/S que são acessados através das mesmas.

As UCPs PL102 e PL103 da série PICCOLO possui somente os módulos função F-CONT.005, F-ANLOG.006 e F-PID.033.
--

A UCP PL104 da série PICCOLO possui somente os módulos função F-CONT.005, F-ANLOG.006, F-PID.033 e F-RELG.048.
--

A UCP PL105 da série PICCOLO possui somente os módulos função F-PID.033 e F-RELG.048.

Função	AL-600 QK600	AL-2000	AL-2002 AL-2003 AL-2004	QK800	QK801	PL104	PL105	QK2000	Módulo de E/S
F-RELOG.000									AL-1420
F-LEDS.001									AL-1460
F-PT100.002									AL-1117 QK1117
F-TERMO.003									AL-1109 QK1109
F-CONTR.004									AL-1440 AL-1450 QK1450
F-CONT.005									AL-600
F-ANLOG.006									AL-600
F-EVENT.017									AL-3130, AL-3132, AL-3138
F-ALNET2.032									
F-PID.033									
F-RAIZN.034									
F-ARQ2.035									
F-ARQ4.036									
F-ARQ8.037									
F-ARQ12.038									
F-ARQ15.039									
F-ARQ16.040									
F-ARQ24.041									
F-ARQ31.042									
F-MOBT.043									
F-STMOD.045									
F-RELG.048									

Função	AL-600, QK600	AL-2000	AL-2002, AL-2003, AL-2004	QK800	QK801	PL104	PL105	QK2000	Módulo de E/S
F-SINC.049									
F-PID16.056									
F-CTRL.059									
F-RELG.061									
F-ALNET1.062									
F-IMP.063									
F-RECEP.064									
F-UTR_S.068									AL-3202
F-NORM.071									
F-COMPF.072									
F-ANDT.090									
F-ORT.091									
F-XORT.092									
F-NEGT.093									

Tabela 4-1 Lista dos Módulos Função Fornecidos pela ALTUS

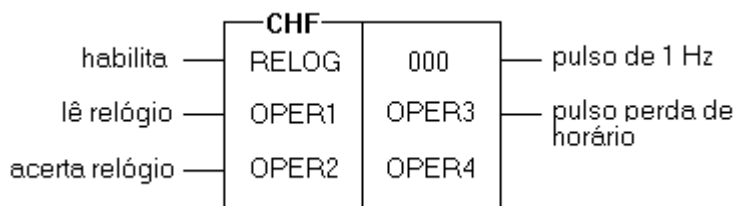
Durante a instalação do MasterTool são copiados diversos módulos com o mesmo nome, sendo armazenados em subdiretórios diferentes, conforme o tipo de UCP ao qual se destinam. Mesmo possuindo o mesmo nome, estes módulos diferem no seu conteúdo. Por exemplo, o módulo F-RELOG.000 está presente nos subdiretórios \AL600, \AL2000, \AL2002, \AL2003 e \AL-2004, porém sendo quatro arquivos diferentes, cada um destinado a uma UCP em particular.

ATENÇÃO:

Os arquivos contidos no subdiretório de um CP não devem ser copiados para o de outro CP, sob o risco de perda de módulos. Deve-se carregar no controlador somente os módulos contidos no subdiretório correspondente à UCP utilizada.

Em caso de dúvidas sobre o tipo de UCP para o qual o módulo foi programado, utilizar o comando de informações de arquivos (comando **Arquivo, Informações do Módulo** no MasterTool).

F-RELOG.000 - Função para Acesso a Módulo Relógio de Tempo Real



Introdução

A função F-RELOG.000 realiza o acesso ao módulo relógio de tempo real AL-1420. Este módulo implementa um relógio e calendário completos com alta precisão, permitindo o desenvolvimento de programas aplicativos que dependam de bases de tempo muito estáveis. O módulo continua a manter a informação de tempo mesmo com a falta de alimentação do sistema, pois é alimentado por baterias.

Programação

Operandos

As células da instrução CHF utilizada para a chamada da função são programadas do seguinte modo:

- **OPER1** - Especifica o número de parâmetros que são passados para a função em OPER3. Este operando deverá ser obrigatoriamente uma constante memória com valor 2 (%KM+00002).
- **OPER2** - Deve ser um operando do tipo constante memória com valor 0 (%KM+00000). Determina o número de parâmetros possíveis de serem programados na janela de edição de OPER4. Como esta função não necessita de nenhum parâmetro em OPER4, o valor de OPER2 é 0.
- **OPER3** - Contém os parâmetros que são passados para a função, declarados através de uma janela visualizada no MasterTool quando a instrução CHF for editada. O número de parâmetros editáveis é especificado em OPER1, sendo fixo em 2 para este módulo:

—* **%RXXXX** - Endereço do barramento onde está alojado o módulo AL-1420.

—* **%MXXXX ou %TMXXXX** - Especificação dos operandos para onde são lidos ou de onde são acertados os valores do relógio. Se este parâmetro for especificado como memória, os valores são lidos ou acertados para a memória declarada e as seis subsequentes. Se for especificado como tabela, os valores são colocados ou acertados a partir da posição 0 até a 6. Caso os operandos não estejam declarados, a leitura ou acerto dos valores de tempo não é realizada e as saídas da instrução são desligadas. É possível o uso de tabelas com mais de 7 posições, sendo que a função ignora as posições excedentes. Os valores são lidos ou acertados nos operandos na seguinte sequência:

Operando	Posição Tabela	Conteúdo	Formato
%MXXXX	0	Segundos	000XX
%MXXXX+1	1	Minutos	000XX
%MXXXX+2	2	Horas	000XX
%MXXXX+3	3	Dia do mês	000XX
%MXXXX+4	4	Mês	000XX
%MXXXX+5	5	Ano	000XX
%MXXXX+6	6	Dia da semana	000XX

Tabela 4-2 Valores Lidos do Relógio (F-RELOG.000)

O conteúdo destes operandos pode ser lido ou modificado a qualquer momento, mas são atualizados com a hora real do módulo apenas quando a instrução for executada. É utilizado o formato 24 horas na contagem do tempo.

- **OPER4** - Não utilizado.

Entradas e Saídas

Descrição das entradas:

- **habilita** - quando esta entrada está energizada a função é chamada, sendo analisados os parâmetros programados na instrução CHF. Caso os mesmos estejam incorretos, todas as saídas da instrução são desenergizadas. Se estiverem corretos, a saída pulso 1 Hz é ligada por uma varredura a cada segundo.
- **lê relógio** - quando energizada, os valores de tempo do módulo são transferidos para os operandos memória ou para a tabela declarada como segundo parâmetro em OPER3.
- **acerta relógio** - quando energizada, os valores contidos nos operandos memória ou tabela declarados são transferidos para o módulo.

ATENÇÃO:

Se as duas últimas entradas forem energizadas simultaneamente, é realizado o acerto dos valores do módulo.

Descrição das saídas:

- **pulso 1 Hz** - indica se houve uma mudança no contador de segundos do relógio. O pulso dura uma varredura e pode ser usado para sincronizar eventos do programa aplicativo que utilizem o Relógio de Tempo Real. Este pulso também pode ser usado para realizar a leitura do relógio somente quando houver uma mudança de segundos, economizando-se tempo de execução, pois a leitura do pulso isoladamente é processada de forma mais rápida.
- **pulso perda de horário** - esta saída é ligada na primeira varredura após a energização do CP caso o relógio tenha ficado sem a alimentação da bateria durante falha na alimentação principal. Para que a mesma seja acionada, é necessário que a instrução esteja habilitada durante a primeira varredura do controlador.

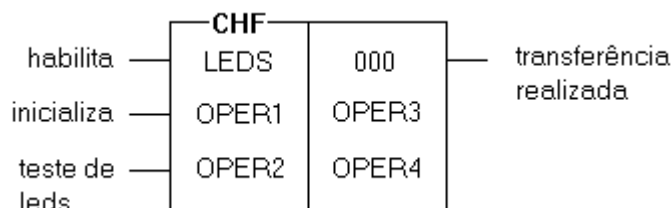
ATENÇÃO:

Quando o módulo permanece sem alimentação de bateria e com o CP desligado, valores inválidos podem ser obtidos na leitura do horário, sem haver contagem de tempo e sem acionamento da saída pulso 1 Hz. Para que o relógio volte a funcionar corretamente, basta realizar o acerto do tempo, programando um novo horário.

Utilização

Esta função pode ser utilizada nas UCPs AL-600, AL-2000/MSP, AL-2002/MSP, AL-2003 e AL-2004.

F-LEDS.001 - Função para Acesso a Módulo Painel de LEDs



Introdução

A função F-LEDS.001 realiza o interfaceamento do programa aplicativo com o módulo multiplexador de LEDs AL-1460, permitindo enviar para este o estado de octetos de pontos digitais do sistema.

Programação

Operandos

As células da instrução CHF utilizada para a chamada da função são programadas do seguinte modo:

- **OPER1** - Especifica o número de parâmetros que serão passados para a função em OPER3. Este operando deverá ser obrigatoriamente uma constante memória com valor 4 (%KM+00004).
- **OPER2** - Deve ser um operando do tipo constante memória com valor 0 (%KM+00000). Determina o número de parâmetros possíveis de serem programados na janela de edição de OPER4. Como esta função não necessita de nenhum parâmetro em OPER4, o valor de OPER2 é 0.
- **OPER3** - Contém os parâmetros que são passados para a função, declarados através de uma janela visualizada no MasterTool quando a instrução CHF for editada. O número de parâmetros editáveis é especificado em OPER1, sendo fixo em 4 para este módulo:

—* **%RXXXX** - Endereço do barramento onde está alojado o módulo AL-1460.

—* **%AXXXX, %EXXXX ou %SXXXX** - Endereço do octeto inicial a partir do qual serão feitas as movimentações para o painel de LEDs.

—* **%KM+XXXXX** - Número de octetos a serem transferidos.

—* **%KM+XXXXX** - Número do octeto inicial do painel de LEDs para onde serão feitas as movimentações. Deve estar compreendido no intervalo entre 0 e 31.

ATENÇÃO:

O espaço de endereçamento dos relés auxiliares (%AXXXX) encontra-se em seguida ao dos relés de entrada e saída (%EXXXX e %SXXXX). Desta forma, se o octeto inicial a ser transferido for %S0062 e o número de transferências for de 5, os octetos transferidos serão %S0062, %S0063, %A0000, %A0001 e %A0002.

- **OPER4** - Não utilizado

Entradas e Saídas

Descrição das entradas:

- **habilita** - quando esta entrada está energizada, a função realiza a transferência de octetos para o painel de LEDs. Os octetos a serem transferidos são definidos pelo segundo e terceiro parâmetros de OPER3, e o destino da transferência pelo quarto parâmetro.
- **inicializa** - quando esta entrada é energizada todos os LEDs do painel são apagados. A entrada pode ser utilizada durante a execução do programa aplicativo para apagar todos os LEDs sem a necessidade de se realizar modificações nos valores dos octetos.

ATENÇÃO:

Para que os valores sejam visualizados no painel, a entrada inicializa deve estar necessariamente desenergizada.

- **teste de leds** - quando esta entrada é energizada, a instrução liga todos os LEDs do painel, permitindo a verificação do funcionamento dos mesmos. Esta entrada pode ser energizada durante a execução do programa aplicativo com a finalidade de acender todos os LEDs sem a necessidade de se realizar modificações nos valores dos octetos.

ATENÇÃO:

Caso as entradas inicializa e teste de leds estejam energizadas ao mesmo tempo, é realizada a inicialização do módulo, resultando no apagamento dos LEDs do painel.

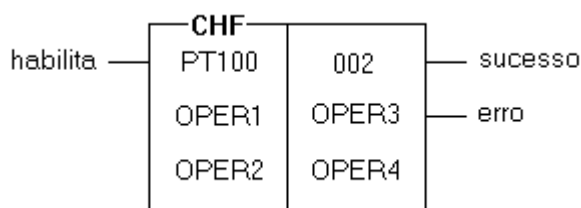
Descrição das saídas:

- **transferência realizada** - após todos os octetos serem transferidos ou caso a entrada teste de leds esteja ativada, esta saída é energizada. Se o número de transferências e o endereço inicial dos octetos fonte ou destino forem incompatíveis, a instrução não é realizada e a saída é desenergizada.

Utilização

Esta função pode ser utilizada nas UCPs AL-600, AL-2000/MSP, AL-2002/MSP, AL-2003, AL-2004, QK800, QK801 e QK2000/MSP.

F-PT100.002 - Função para Leitura de Módulo Pt-100



Introdução

A função F-PT100.002 realiza a leitura de temperaturas fornecidas pelos módulos AL-1117 e QK1117, módulo interface com até 4 sensores do tipo Pt-100. Os valores lidos podem ser linearizados ou não, sendo possível a leitura de 1 ou 4 canais, alterando-se apenas a programação dos parâmetros utilizados na sua chamada.

Programação

Operandos

As células da instrução CHF utilizada para a chamada da função são programadas do seguinte modo:

- **OPER1** - Especifica o número de parâmetros que são passados para a função em OPER3. Este operando deve ser obrigatoriamente uma constante memória com valor 4 (%KM+00004).
- **OPER2** - Deve ser um operando do tipo constante memória com valor 0 (%KM+00000). Determina o número de parâmetros possíveis de serem programados na janela de edição de OPER4. Como esta função não necessita de nenhum parâmetro em OPER4, o valor de OPER2 é 0.
- **OPER3** - Contém os parâmetros que são passados para a função, declarados através de uma janela visualizada no MasterTool quando a instrução CHF for editada. O número de parâmetros editáveis é especificado em OPER1, sendo fixo em 4 para este módulo:

—* **%RXXXX ou %RXXXX.X** - Endereço do barramento onde está alojado o módulo AL-1117 ou QK1117. Caso seja especificado com subdivisão de ponto (%RXXXX.X), a leitura é realizada somente para o canal do módulo correspondente ao ponto (pontos .0 a .3 do operando correspondendo aos canais 0 a 3 do módulo, respectivamente). No caso de não ser feita a especificação com subdivisão de ponto (%RXXXX), todos os 4 canais serão lidos (0 a 3).

—* **%KM+XXXXX** - Especificação do tipo de linearização a ser executada (ver tabela 6-2 para ajuste dos módulos AL-1117 e QK1117). Os seguintes tipos são válidos:

—* **%KM+00000** - a função não executa qualquer linearização, apresentando como resultado a saída do conversor com valores entre 0 e 4095.

—* **%KM+00001** - a função executa linearização para a faixa de temperatura de -30,00 a +50,00 °C, representada em valores de +0000 a +8000 (Valor armazenado = $(T + 30) \times 100$).

—* **%KM+00002** - a função executa linearização para a faixa de temperatura de -30,00 a +370,00 °C, representada em valores de +0000 a +4000 (Valor armazenado = $(T + 30) \times 10$).

—* **%KM+00003** - a função executa linearização para a faixa de temperatura de -30,00 a +770,00 °C, representada em valores de +0000 a +8000 (Valor armazenado = $(T + 30) \times 10$).

—* **%KM+00004** - a função executa linearização para a faixa de temperatura de -30,00 a +50,00 °C, representada em valores de -3000 a +5000.

—* **%KM+00005** - a função executa linearização para a faixa de temperatura de -30,00 a +370,00 °C, representada em valores de -0300 a +3700.

—* **%KM+00006** - a função executa linearização para a faixa de temperatura de -30,00 a +770,00 °C, representada em valores de -0300 a +7700.

Constante de Linearização	Faixa de Medida	Valor Armazenado	PA5	PA6
%KM+00000	qualquer	0000 a +4095	0/1	0/1
%KM+00001	-30 °C a +50 °C	0000 a +8000	0	0
%KM+00002	-30 °C a +370 °C	0000 a +4000	1	1
%KM+00003	-30 °C a +770 °C	0000 a +8000	2	1
%KM+00004	-30 °C a +50 °C	-3000 a +5000	0	0
%KM+00005	-30 °C a +370 °C	-0300 a +3700	1	1
%KM+00006	-30 °C a +770 °C	-0300 a +7700	2	1

Tabela 4-3 Linearização e Configuração dos Módulos AL-1117 e QK1117

ATENÇÃO:

Caso a temperatura do sensor exceda a faixa de medida, o valor 9999 será armazenado no canal correspondente.

—* **%MXXXX** - Especificação do operando onde são armazenados os valores dos canais após a leitura e linearização. Se o primeiro parâmetro for especificado como %RXXXX.X (leitura de um canal), apenas a posição de memória declarada no parâmetro 3 é atualizada. Se o primeiro parâmetro for especificado como %RXXXX (leitura de 4 canais), é utilizada a memória declarada no parâmetro 3 e as 3 consecutivas à mesma.

—* **%MXXXX** - Operando utilizado pela função para o controle interno do seu processamento.

ATENÇÃO:

O operando de controle não deve ter seu conteúdo alterado em nenhuma parte do programa aplicativo, sob pena de prejudicar a execução correta da função. Cada CHF para este módulo F deve possuir um operando de controle exclusivo, diferente dos demais. O operando de controle não deve ser retentivo.

- **OPER4** - Não utilizado.

Entradas e Saídas

Descrição das entradas:

- **habilita** - quando esta entrada está energizada a função é chamada, sendo analisados os parâmetros programados na instrução CHF.

Descrição das saídas:

- **sucesso** - é energizada quando a função foi corretamente executada.
- **erro** - esta saída é energizada sempre que ocorre um dos seguintes erros:

- * ruptura da ligação com o sensor Pt-100
- * curto-circuito na ligação com o sensor Pt-100
- * o módulo declarado no barramento não é AL-1117 ou QK1117
- * erro na especificação dos operandos ou tentativa de acesso a operandos não declarados

Nos dois primeiros erros, o valor do operando correspondente ao canal recebe o valor 9999.

ATENÇÃO:

A saída de erro está implementada a partir da versão 1.10 de F-PT100.002.

Utilização

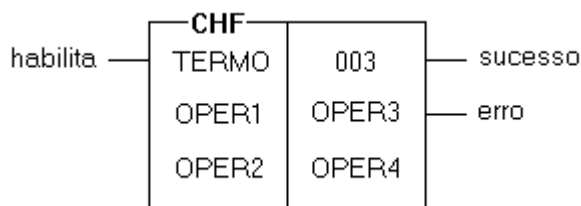
Esta função pode ser utilizado nas UCPs AL-600, AL-2002/MSP, AL-2002/MSP, AL-2003, AL-2004, QK800, QK801 e QK2000/MSP.

ATENÇÃO:

O tempo de atualização para cada canal é de 400 ms. Este tempo é contabilizado pela própria função. Desta forma, a instrução CHF utilizada para a chamada do módulo F não deve ser saltada, sob pena de aumentar-se o tempo de conversão.

A função não pode ser chamada para um outro canal antes de ser encerrada a conversão do canal corrente.

F-TERMO.003 - Função para Leitura de Módulo Termopar



Introdução

A função F-TERMO.003 realiza a leitura de temperaturas fornecidas pelos módulos AL-1109 e QK1109, módulo interface com até 4 sensores do tipo termopar. Os valores lidos podem ser linearizados ou não sendo possível a leitura de 1 ou 4 canais, alterando-se apenas a programação dos parâmetros utilizados na sua chamada.

Programação

Operandos

As células da instrução CHF utilizada para a chamada da função são programadas do seguinte modo:

- **OPER1** - Especifica o número de parâmetros que são passados para a função em OPER3. Este operando deve ser obrigatoriamente uma constante memória com valor 5 (%KM+00005).
- **OPER2** - Deve ser um operando do tipo constante memória com valor 0 (%KM+00000). Determina o número de parâmetros possíveis de serem programados na janela de edição de OPER4. Como esta função não necessita de nenhum parâmetro em OPER4, o valor de OPER2 é 0.
- **OPER3** - Contém os parâmetros que são passados para a função, declarados através de uma janela visualizada no MasterTool quando a instrução CHF for editada. O número de parâmetros editáveis é especificado em OPER1, sendo fixo em 5 para este módulo:

—* **%RXXXX ou %RXXXX.X** - Endereço do barramento onde está alojado o módulo AL-1109 ou QK1109. Caso seja especificado com subdivisão de ponto (%RXXXX.X), a leitura é realizada somente para o canal do módulo correspondente ao ponto (pontos .0 a .3 do operando, correspondendo aos canais 0 a 3 do módulo, respectivamente). No caso de não ser feita a especificação com subdivisão de ponto (%RXXXX), todos os 4 canais são lidos (0 a 3).

—* **%KM+XXXXX** - Especificação do tipo de termopar:

—* **%KM+00000** - valor fornecido pelo módulo sem linearização

—* **%KM+00001** - termopar tipo J

—* **%KM+00002** - termopar tipo R

—* **%KM+00003** - termopar tipo S

—* **%KM+00004** - termopar tipo K

—* **%KM+00005** - termopar tipo B

—* **%KM+XXXXX** - Define se os valores serão apresentados normalizados (0000 a 1000) ou em graus (ver tabela 4-3):

—* **%KM+00000** - resultado em graus Célsius

—* **%KM+00001** - resultado normalizado

Tipo de Termopar	Resultado em graus Célsius (oC)	Resultado Normalizado
J	0000 a 1000	0000 a 1000
R	0000 a 1500	0000 a 1000
S	0000 a 1500	0000 a 1000
K	0000 a 1250	0000 a 1000
B	0000 a 1800	0000 a 1000

Tabela 4-4 Valores Lidos dos Módulos AL-1109 e QK1109

ATENÇÃO:

Caso a temperatura do sensor exceda a faixa de medida, o valor 9999 será armazenado no canal correspondente.

—* **%MXXXX** - Especificação do operando onde são armazenados os valores dos canais após a leitura e normalização. Se o primeiro parâmetro for especificado como **%RXXXX.X** (leitura de um canal) apenas a posição de memória declarada no parâmetro 3 é utilizada. Se o primeiro parâmetro for especificado como **%RXXXX** (leitura de 4 canais), é utilizada a memória declarada no parâmetro 3 e as 3 consecutivas à mesma.

—* **%MXXXX** - Operando utilizado pela função para o controle interno do seu processamento.

ATENÇÃO:

O operando de controle não deve ter seu conteúdo alterado em nenhuma parte do programa aplicativo, sob pena de prejudicar a execução correta da função. Cada CHF para este módulo F deve possuir um operando de controle exclusivo, diferente dos demais.

- **OPER4** - Não utilizado.

Entradas e Saídas

Descrição das entradas:

- **habilita** - quando esta entrada está energizada a função é chamada, sendo analisados os parâmetros programados na instrução CHF.

Descrição das saídas:

- **sucesso** - é energizada quando a função foi corretamente executada.
- **erro** - esta saída é energizada sempre que ocorre um dos seguintes erros:

- * ruptura da ligação com o sensor termopar
- * o módulo declarado no barramento não é AL-1109 ou QK1109
- * erro na especificação dos operandos ou tentativa de acesso a operandos não declarados

No primeiro erro, o valor do operando correspondente ao canal recebe o valor 9999.

ATENÇÃO:

A saída de erro está implementada a partir da versão 1.10 de F-TERMO.003.

Utilização

Esta função pode ser utilizada nas UCPs AL-600, AL-2000/MSP, AL-2002/MSP, AL-2003, AL-2004, QK800, QK801 e QK2000/MSP.

ATENÇÃO:

A partir da versão 1.10 do módulo F-TERMO.003, a saída de erro é ligada e o valor 9999 é colocado nos canais na primeira varredura de execução do programa. O tempo de atualização de um canal foi reduzido de 400 ms para 100 ms a partir desta mesma versão.

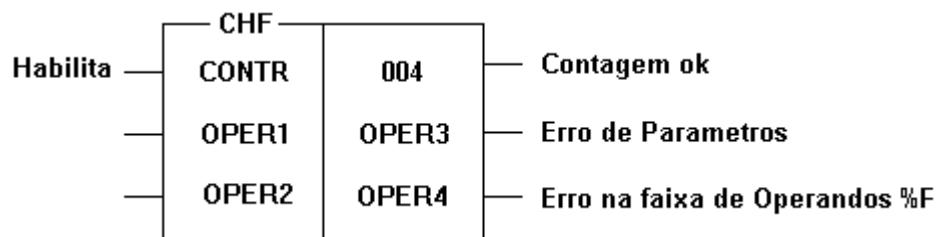
ATENÇÃO:

O tempo de atualização para cada canal é de 100 ms. Este tempo é contabilizado pela própria função. Desta forma, a instrução CHF utilizada para a chamada do módulo F não deve ser saltada, sob pena de aumentar-se o tempo de conversão.

A função não pode ser chamada para um outro canal antes de ser encerrada a conversão do canal corrente.

F-CONTR.004 - Função para Acesso a Módulo Contador Rápido

Para a Série Grano:



Introdução

A função F-CONTR.004 realiza o interfaceamento do programa aplicativo com o Contador Rápido 24 bits do HardFlex modelo GR900 dos controladores da Série Grano. Esta função também é utilizada para interfaceamento com outros modelos de HardFlex que possuam contadores rápidos.

Parametrização

O contador rápido é parametrizado através de um byte utilizando a tabela a seguir:

Bits do byte de parametrização								Descrição
7	6	5	4	3	2	1	0	
						0	0	Modo 0 – A incrementa e B decrementa
						0	1	Modo 1 – A informa o sentido e B conta
						1	0	Modo 2 – Quadratura, 4 contagens por período
						1	1	Modo 3 – Quadratura, 2 contagens por período
					0			Desabilita saída física do comparador 1 (saída T0)
					1			Habilita saída física do comparador 1 (saída T0)
				0				Desabilita saída física do comparador 2 (saída T1)
				1				Habilita saída física do comparador 2 (saída T1)
			0					Desabilita saída física do comparador Zero (saída T2)
			1					Habilita saída física do comparador Zero (saída T2)
		0						Desabilita entrada física de zeramento (entrada I3)
		1						Habilita entrada física de zeramento (entrada I3)
	0							Desabilita entrada física de congelamento (entrada I4)
	1							Habilita entrada física de congelamento (entrada I4)
0								Zera o contador ao atingir os valores de overflow e underflow
1								Zera contador ao atingir valor do registro de comparação 2

Notas da Tabela:

Bits 0 e 1 – Estes bits selecionam o modo de operação do contador rápido:

- 00 – Pulso na entrada Contagem A -> Incrementa contador; Pulso em Contagem B -> Decrementa o contador;
- 01 – Nível do sinal na entrada Contagem A dá o sentido da contagem e pulsos na entrada Contagem B incrementam ou decrementam a contagem;
- 10 – Uso com transdutor de posição. Pulsos em quadratura provenientes do transdutor de posição são inseridos nas entradas Contagem A e Contagem B, a decodificação dos pulsos gera quatro contagens por período;
- 11 – Idem ao modo 2, mas com duas contagens por período.

Bit 2 – Quando em 1 habilita a saída física do comparador 1 no borne da saída T0. Toda vez que o valor do contador for igual ao valor do comparador 1 será gerado na saída T0 um pulso com duração mínima de 50ms.

Bit 3 – Quando em 1 habilita a saída física do comparador 2 no borne da saída T1. Toda vez que o valor do contador for igual ao valor do comparador 2 será gerado na saída T1 um pulso com duração mínima de 50ms.

Bit 4 – Quando em 1 habilita a saída física do comparador Zero no borne da saída T2. Toda vez que o valor do contador for igual a zero será gerado na saída T2 um pulso com duração mínima de 50ms.

Bit 5 – Quando em 1 habilita a entrada de zeramento externa do contador via borne de entrada I3. O contador é zerado (resetado) sempre que for aplicado nível alto de sinal na entrada I3.

Bit 6 – Quando em 1 habilita a entrada de congelamento de contagem externa do contador via borne de entrada I4. A contagem permanecerá congelada enquanto for aplicado nível alto de sinal na entrada I4.

Bit 7 – Quando em 1 habilita o zeramento (reset) do contador sempre que o mesmo atingir valor de contagem igual ao valor do comparador 2. Quando em 0 o contador somente será zerado após atingir valores de overflow (+8.388.607) ou underflow (-8.388.608).

Programação**Operandos**

As células da instrução CHF utilizada para a chamada da função são programadas do seguinte modo:

- **OPER1** - Especifica o número de parâmetros que são passados para a função em OPER3. Este operando deve ser obrigatoriamente uma constante memória com valor 6 (%KM+00006).
- **OPER2** - Deve ser um operando do tipo constante memória com valor 0 (%KM+00000). Determina o número de parâmetros possíveis de serem programados na janela de edição de OPER4. Como esta função não necessita de nenhum parâmetro em OPER4, o valor de OPER2 é 0.
- **OPER3** - Contém os parâmetros que são passados para a função, declarados quando a instrução CHF for editada. O número de parâmetros editáveis é especificado em OPER1, sendo fixo em 5 para este módulo:

* **%KMXXXX** – Posição do módulo no barramento Grano. Para contadores internos ao controlador (Hardflex) este valor deve ser %KM0000.

É importante que este parâmetro seja configurado corretamente, pois uma configuração errada comprometerá o funcionamento deste dispositivo e de outros nestes mesmo HardFlex.

* **%KMXXXX** – Modelo de HardFlex ou módulo no qual está o contador ao qual se deseja executar algum comando.

%KM0001 – GR900

%KM0002 – GR901

%KM0003 – GR902

%KM0004 – GR903

* **%KMXXXX** – Número do contador no qual se deseja executar algum comando. No caso do GR900 existe apenas um contador, ou seja qualquer comando é dado para este e o parâmetro deve ser sempre %KM0001. Outros modelos de HardFlex poderão conter mais de um contador o que implica na necessidade deste byte para um módulo F que busca ser genérico. Ex.: Se o comando é para o segundo contador do HardFlex, o valor será %KM0002.

É importante que este parâmetro seja configurado corretamente, pois uma configuração errada comprometerá o funcionamento deste dispositivo e de outros nestes mesmo HardFlex.

* **%FXXXX ou %TFXXXX** – Primeiro operando de leitura e escrita do contador. São quatro operandos na seguinte ordem:

Valor lido do Contador.

Valor a ser escrito no Contador.

Valor a ser escrito no Comparador 1.

Valor a ser escrito no Comparador 2.

Em caso de %TFXXXX esta ordem representa o numero do índice da tabela que deve ter 4 posições. Para %FXXXX este representa Valor lido do Contador, enquanto as outras opções estão nos 3 operandos logo após este.

* **%AXXXX ou %MXXXX** - Operando onde está armazenado o status do contador.

Byte de Status (%A ou %M)								Descrição
7	6	5	4	3	2	1	0	
							0	Contagem normal
							1	Overflow na contagem
							0	Contagem normal
							1	Underflow na contagem
					0			Direção de contagem DOWN
					1			Direção de contagem UP
				0				Contagem diferente do comparador 1
				1				Contagem igual ao comparador 1
			0					Contagem diferente do comparador 2
			1					Contagem igual ao comparador 2
		0						Contagem diferente de zero
		1						Contagem igual a zero
0	0							Sempre zeros

Notas da Tabela:

Bit 0 – É acionado sempre que o valor do contador estourar positivamente, ou seja, passar por +8.388.607. Deve ser zerado pela aplicação através do bit 2 do 2o byte de comando.

Bit 1 – É acionado sempre que o valor do contador estourar negativamente, ou seja, passar por - 8.388.608. Deve ser zerado pela aplicação através do bit 2 do 2o byte de comando.

Bit 2 – Mostra em que direção ocorreu a última contagem. Quando em 1 significa que a última contagem ocorreu na direção UP, ou seja, o contador foi incrementado. Quando em 0 significa que a última contagem ocorreu na direção DOWN, ou seja, o contador foi decrementado.

Bit 3 – É acionado sempre que o valor do contador for igual ao valor do comparador 1. Este bit será desacionado assim que houver a primeira leitura do byte de status por parte do aplicativo, desde que o valor do contador não seja mais igual ao valor do comparador 1.

Bit 4 – É acionado sempre que o valor do contador for igual ao valor do comparador 2. Este bit será desacionado assim que houver a primeira leitura do byte de status por parte do aplicativo, desde que o valor do contador não seja mais igual ao valor do comparador 2.

Bit 5 – É acionado sempre que o valor do contador for igual a zero. Este bit será desacionado assim que houver a primeira leitura do byte de status por parte do aplicativo, desde que o valor do contador não seja mais igual a zero.

%AXXXX ou %MXXXX – Comando que diz qual o procedimento que será executado na chamada atual da função. Possui dois bytes e no caso de %AXXXX, o operando declarado é o mais significativo e o seguinte o menos significativo.

1º Byte de Comando(%A ou %M)								Descrição
7	6	5	4	3	2	1	0	
							0	Contagem normal
							1	Escreve no contador
						0		Contagem normal
						1		Escreve no comparador 1
					0			Contagem normal
					1			Escreve no comparador 2
0	0	0	0	0				Sempre zeros

2º Byte de Comando(%A ou %M)								Descrição
7	6	5	4	3	2	1	0	
							0	Contagem normal
							1	Inibe contagem (congela valor do contador)
						0		Contagem normal
						1		Zera registrador de contagem
					0			Contagem normal
					1			Zera bit de status de overflow/underflow
0	0	0	0	0				Sempre zeros

Notas das tabelas:

1 - O comando de zeramento do contador (bit 1 do segundo byte de comando) tem prioridade sobre os outros comandos. Se houver comandos de congelamento, zeramento e escrita do valor do contador simultaneamente por exemplo, o contador será somente zerado, não mantendo seu valor atual nem recebendo o novo valor;

- 2 - Os comandos de congelamento de contagem e escrita no contador não são conflitantes, podendo ser executados simultaneamente;
- 3 - Os comandos de escrita nos comparadores são independentes entre si e do contador, podendo ser executados simultaneamente a outros comandos;
- 4 - Os comandos do 2o byte estendem-se até uma nova chamada da função solicitando contagem normal.

Entradas e Saídas

Descrição das entradas:

- **Habilita** - quando esta entrada está energizada a função é chamada, sendo analisados os parâmetros programados na instrução CHF. Caso os mesmos estejam incorretos, a respectiva saída de erro da função é energizada. Se estiverem corretos, os comandos contidos nos bytes auxiliares %XXXXX são executados, realizando as operações de leitura e escrita conforme especificado. Quando a entrada habilita estiver desenergizada, as instruções anteriormente enviadas ao módulo são mantidas, não executando nenhuma operação de leitura ou escrita no mesmo.

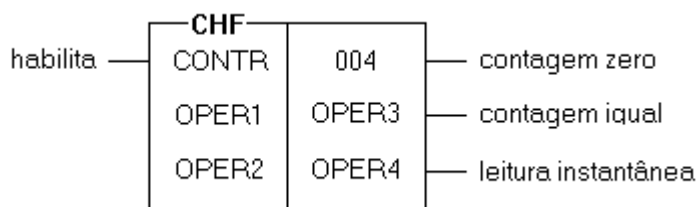
Descrição das saídas:

- **Contagem ok** – é ativada para indicar sucesso na tentativa de realizar a operação solicitada. Sempre que a operação for concluída com sucesso essa saída é ligada e as restantes são desligadas. Se esta saída estiver desligada significa que algo de errado ocorreu no processo, sendo que o resultado pode ser ou não indicado numa das outras duas saídas.
- **Erro parâmetros** - é ativada quando existe alguma inconsistência nos parâmetros passados para a função.
- **Erro na faixa de operandos %F** - é ativada quando um de um operando %F de escrita do modo contagem ultrapassar a faixa de 24 bits com sinal, ou seja, o valor for maior que +8.388.607 ou menor que -8.388.608.
- **Todas as saídas desligadas** - indica erro na comunicação com os dispositivos selecionados.

Utilização

Esta função pode ser utilizada nas UCPs GR350, GR351, GR370 e GR371.

Para demais Séries:



Introdução

A função F-CONTR.004 realiza o interfaceamento do programa aplicativo com o módulo contador rápido AL-1440 e contador rápido com interface para transdutores óticos AL-1450 e QK1450.

Programação

Operandos

As células da instrução CHF utilizada para a chamada da função são programadas do seguinte modo:

- **OPER1** - Especifica o número de parâmetros que são passados para a função em OPER3. Este operando deve ser obrigatoriamente uma constante memória com valor 4 (%KM+00004).
- **OPER2** - Deve ser um operando do tipo constante memória com valor 0 (%KM+00000). Determina o número de parâmetros possíveis de serem programados na janela de edição de OPER4. Como esta função não necessita de nenhum parâmetro em OPER4, o valor de OPER2 é 0.
- **OPER3** - Contém os parâmetros que são passados para a função, declarados através de uma janela visualizada no MasterTool quando a instrução CHF for editada. O número de parâmetros editáveis é especificado em OPER1, sendo fixo em 4 para este módulo:

—* **%RXXXX** - Endereço do barramento onde está alojado o módulo AL-1440, AL-1450 ou QK1450.

—* **%DXXXX** - Operando com o valor a ser lido ou escrito no registrador de contagem do módulo.

—* **%DXXXX** - Operando com o valor a ser lido ou escrito no registrador de comparação do módulo.

—* **%AXXXX** - Octeto que contém um conjunto de instruções para o módulo, descrito a seguir:

- **%AXXXX.0** - inibe contagem
- **%AXXXX.1** - zera registrador contador
- **%AXXXX.2** - habilita relés de saída (comparação por hardware)
- **%AXXXX.3** - habilita entrada de referência
- **%AXXXX.4** - executa leitura do contador
- **%AXXXX.5** - executa escrita no contador
- **%AXXXX.6** - executa escrita no comparador
- **%AXXXX.7** - não utilizado
- **OPER4** - Não utilizado.

Entradas e Saídas

Descrição das entradas:

- **habilita** - quando esta entrada está energizada a função é chamada, sendo analisados os parâmetros programados na instrução CHF. Caso os mesmos estejam incorretos, todas as saídas da instrução são desenergizadas. Se estiverem corretos, as instruções contidas no octeto auxiliar %AXXXX são executadas, realizando as operações de leitura e escrita conforme especificado. Quando a entrada habilita estiver desenergizada, as instruções anteriormente enviadas ao módulo são mantidas, não executando nenhuma operação de leitura ou escrita no mesmo.

Descrição das saídas:

- **contagem zero** - é ativada quando o registrador contador atinge o valor zero. A sua ativação é realizada ao menos durante uma varredura do programa aplicativo, caso o contador passe por zero e não permaneça com este valor.
- **contagem igual** - é ativada quando o valor de contagem atinge o valor igual ao do registrador comparador. Assim como a saída anterior, é ativada por ao menos uma varredura de programa aplicativo.
- **leitura instantânea** - é ativada somente durante a varredura seguinte ao pedido de leitura, caso esta não esteja desabilitada por ponte de ajuste do módulo. Neste caso, a próxima leitura do contador fornece o valor deste no momento em que foi solicitada a leitura instantânea.

ATENÇÃO:

Caso sejam solicitadas simultaneamente a leitura e a escrita no módulo contador, apenas é executada a escrita. Independentemente desta situação, a escrita no registrador é executada sempre que solicitada, se a instrução estiver habilitada. Deve-se levar em conta o fato de que cada operação de leitura ou escrita é relativamente lenta. Desta forma, devem ser evitadas operações desnecessárias a fim de não comprometer o tempo de varredura do programa aplicativo.

ATENÇÃO:

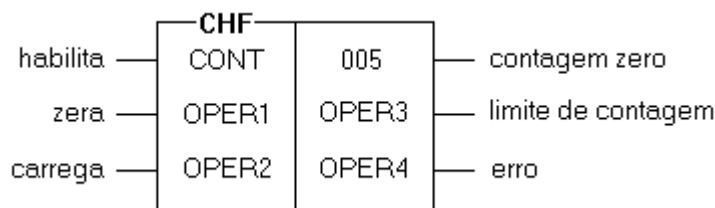
Após o CP ser ligado, o valor dos registradores de contagem e comparação é aleatório, devendo ser inicializado convenientemente.

Maiores informações podem ser obtidas nas Características Técnicas dos módulos AL-1440, AL-1450 e QK1450.

Utilização

Esta função pode ser utilizada nas UCPs AL-600, AL-2000/MSP, AL-2002/MSP, AL-2003, AL-2004, QK800, QK801 e QK2000/MSP.

F-CONT.005 - Função para Acesso às Entradas de Contagem Rápida



Introdução

A função F-CONT.005 realiza o interfaceamento do programa aplicativo com as entradas de contagem rápida integradas nas UCP AL-600, QK600, PL102, PL103 e PL104.

As UCPs AL-600, QK600, PL102, PL103 e PL104 possuem no seu painel frontal duas entradas de contagem rápida, possibilitando a contagem de pulsos com elevada frequência (até 10 KHz), quando se torna inadequada a contagem por meio de pontos de entrada convencionais.

Programação

Operandos

As células da instrução CHF utilizada para a chamada da função são programadas do seguinte modo:

- **OPER1** - Especifica o número de parâmetros que são passados para a função em OPER3. Este operando deve ser obrigatoriamente uma constante memória com valor 3 (%KM+00003).
- **OPER2** - Deve ser um operando do tipo constante memória com valor 0 (%KM+00000). Determina o número de parâmetros possíveis de serem programados na janela de edição de OPER4. Como esta função não necessita de nenhum parâmetro em OPER4, o valor de OPER2 é 0.
- **OPER3** - Contém os parâmetros que são passados para a função, declarados através de uma janela visualizada no MasterTool quando a instrução CHF for editada. O número de parâmetros editáveis é especificado em OPER1, sendo fixo em 3 para este módulo:

—* **%KM+XXXXX** - Número da entrada contadora (0 ou 1).

—* **%DXXXX** - Operando que armazena o valor de contagem.

—* **%MXXXX** - Operando utilizado pela função para o controle interno do seu processamento.

ATENÇÃO:

O operando de controle não deve ter seu conteúdo alterado em nenhuma parte do programa aplicativo, sob pena de prejudicar a execução correta da função. Cada CHF para este módulo F deve possuir um operando de controle exclusivo, diferente dos demais.

- **OPER4** - Não utilizado.

Entradas e Saídas

Descrição das entradas:

- **habilita** - quando esta entrada está energizada a função é chamada, sendo analisados os parâmetros programados na instrução CHF. Caso os mesmos estejam incorretos, a saída de erro é energizada.
- **zera** - causa o zeramento do valor de contagem, quando habilitada.
- **carrega** - quando ativada, faz com que o valor armazenado no operando %D seja o novo valor de contagem.

Descrição das saídas:

- **contagem zero** - é energizada quando o valor do operando %D de contagem possui valor zero.
- **limite de contagem** - é energizada quando o valor do operando %D de contagem possui valor +9999999.
- **erro** - é energizada caso ocorra erro na especificação dos operandos ou tentativa de acesso a operandos não declarados.

Quando se passa o CP de programação para execução a entrada ZERA das funções F-CONT.005 devem ser acionadas por uma varredura, de forma a permitir o referenciamento da função, visto que seus operandos de contagem e de controle foram zerado na troca de estados.

Utilização

Esta função pode ser utilizada somente nas UCPs AL-600, QK600, PL102, PL103 e PL104.

Descrição do Funcionamento

Cada contador realiza uma contagem incremental, de 0 a +9999999 pulsos, armazenada em um operando %D. Quando a contagem atinge o valor limite, o operando não é mais incrementado, ligando a saída limite de contagem na instrução CHF.

O valor no operando de contagem pode ser inicializado com o acionamento da entrada zera da instrução. Para começar a contagem com um valor diferente de zero, basta mover o valor desejado para o operando %D e acionar a entrada carrega da instrução.

A função deve ser chamada periodicamente, no ciclo de varredura normal ou no módulo executado por interrupção de tempo E018. A frequência máxima de contagem depende do período de chamada, sendo mostrada na tabela 4-4.

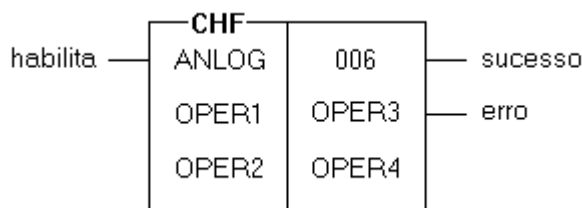
Frequência Contagem	Máxima de	Período de Chamada da Função
2,5 KHz		100 ms
3,4 KHz		75 ms
5,0 KHz		50 ms
10,4 KHz		25 ms

Tabela 4-5 Frequência de Contagem AL-600

A função pode ser chamada várias vezes em programas com alto tempo de ciclo de varredura, permitindo a contagem de frequências mais elevadas.

Por exemplo, caso o tempo de ciclo do programa aplicativo seja 85 ms e seja necessário a contagem de pulsos até 7 KHz, deve-se repetir a chamada da função 4 vezes ao longo do programa ou incluí-la no módulo E018 com frequência de execução de 25 ms.

F-ANLOG.006 - Função para Conversão A/D ou D/A Integrados



Introdução

A função F-ANLOG.006 realiza a conversão A/D (analógico/digital) ou D/A (digital/analógico) dos canais analógicos integrados disponíveis nas UCPs AL-600, QK600, PL102, PL103 e PL104 (DAC 1 e DAC 2).

Utilizando-se duas instruções CHF, é possível realizar a conversão A/D em um dos canais e D/A no outro ou o mesmo tipo de conversão em ambos.

Programação

Operandos

As células da instrução CHF utilizada para a chamada da função são programadas do seguinte modo:

- **OPER1** - Especifica o número de parâmetros que são passados para a função em OPER3. Este operando deve ser obrigatoriamente uma constante memória com valor 3 (%KM+00003).
- **OPER2** - Deve ser um operando do tipo constante memória com valor 0 (%KM+00000). Determina o número de parâmetros possíveis de serem programados na janela de edição de OPER4. Como esta função não necessita de nenhum parâmetro em OPER4, o valor de OPER2 é 0.
- **OPER3** - Contém os parâmetros que são passados para a função, declarados através de uma janela visualizada no MasterTool quando a instrução CHF for editada. O número de parâmetros editáveis é especificado em OPER1, sendo fixo em 3 para este módulo:

—* **%KM+XXXXX** - Especificação do canal a ser convertido. Deve-se utilizar %KM+00000 para DAC 1 e %KM+00001 para DAC 2.

—* **%KM+XXXXX** - Tipo de conversão a ser realizada no canal definido pelo parâmetro anterior. Deve-se utilizar %KM+00000 para conversão A/D e %KM+00001 para conversão D/A.

—* **%MXXXXX** - Especificação do operando onde é armazenado o valor a ser escrito no conversor em caso de conversão D/A ou valor lido em caso de conversão A/D.

- **OPER 4** - Não utilizado.

Entradas e Saídas

Descrição das entradas:

- **habilita** - quando esta entrada está energizada a função é chamada, sendo analisados os parâmetros programados na instrução CHF.

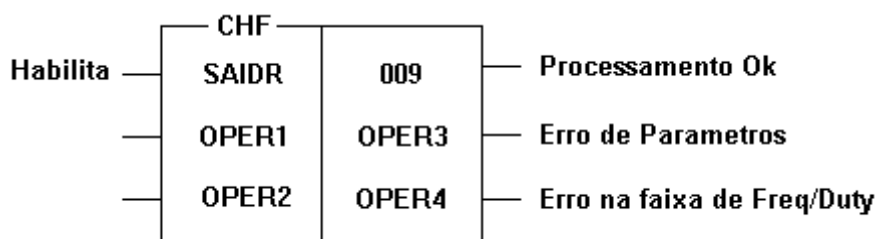
Descrição das saídas:

- **sucesso** - é energizada quando a função foi corretamente executada.
- **erro** - é energizada caso ocorra erro na especificação dos operandos ou tentativa de acesso a operandos não declarados.

Utilização

Esta função pode ser utilizada somente nas UCPs AL-600, QK600, PL102, PL103 e PL104.

F- SAIDR.009 - Função para Acesso às Saídas Rápidas



Introdução

A função F-SAIDR.009 realiza o interfaceamento do programa aplicativo com as saídas rápidas do HardFlex modelo GR900 dos controladores da Série Grano. Esta função também é utilizada para interfaceamento com outros modelos de HardFlex que possuam saídas rápidas.

Programação

Operandos

As células da instrução CHF utilizada para a chamada da função são programadas do seguinte modo:

- **OPER1** - Especifica o número de parâmetros que são passados para a função em OPER3. Este operando deve ser obrigatoriamente uma constante memória com valor 5 (%KM+00005).
- **OPER2** - Deve ser um operando do tipo constante memória com valor 0 (%KM+00000). Determina o número de parâmetros possíveis de serem programados na janela de edição de OPER4. Como esta função não necessita de nenhum parâmetro em OPER4, o valor de OPER2 é 0.
- **OPER3** - Contém os parâmetros que são passados para a função, declarados quando a instrução CHF for editada. O número de parâmetros editáveis é especificado em OPER1, sendo fixo em 5 para este módulo:

* **%KMXXXX** – Posição do módulo no barramento Grano. Para saídas rápidas internas ao controlador (Hardflex) este valor deve ser KM0000.

É importante que este parâmetro seja configurado corretamente, pois uma configuração errada comprometerá o funcionamento deste dispositivo e de outros nestes mesmo HardFlex.

* **%KMXXXX** – Modelo de HardFlex ou módulo no qual está a saída rápida a qual se deseja executar algum comando.

%KM0001 – GR900
 %KM0002 – GR901
 %KM0003 – GR902
 %KM0004 – GR903

* **%KMXXXX** – Número do bloco de saídas rápidas no qual deseja executar algum comando. Ex.: O HardFlex GR900 possui um bloco de duas saídas VFO. Estas estão no primeiro bloco de saídas rápidas deste HardFlex. Sendo assim o número do bloco é KM0001. Outros HardFlex podem ter mais blocos de saídas rápidas. É por isso que existe esse parâmetro na função que visa ser genérica para qualquer HardFlex que possua estes blocos.

É importante que este parâmetro seja configurado corretamente, pois uma configuração errada comprometerá o funcionamento deste dispositivo e de outros nestes mesmo HardFlex.

* **%MXXXX ou %TMXXXX** – Primeiro operando de escrita da saída rápida. São quatro operandos na seguinte ordem:

Valor de frequência da saída rápida VFO 0.

Valor de duty cycle da saída rápida VFO 0.

Valor de frequência da saída rápida VFO 1.

Valor de duty cycle da saída rápida VFO 1.

Em caso de %TMXXXX esta ordem representa o número do índice da tabela que deve ter 4 posições. Para %MXXXX este representa Valor de frequência da saída rápida 0, enquanto as outras opções estão nos 3 operandos logo após este.

* **%AXXXX ou %MXXXX** – Comando que diz qual o procedimento será executado nesta varredura da função. Possui dois bytes e no caso de %AXXXX, o operando declarado é o mais significativo e o seguinte o menos significativo.

1º Byte de Comando(%A ou %M)								Descrição
7	6	5	4	3	2	1	0	
							0	Operação normal
							1	Escreve frequência e duty cycle da saída VFO 0
						0		Operação normal
						1		Escreve frequência e duty cycle da saída VFO 1
0	0	0	0	0	0			Sempre zeros

2º Byte de Comando(%A ou %M)								Descrição
7	6	5	4	3	2	1	0	
							0	Desabilita saída VFO 0 (saída parada)
							1	Habilita saída VFO 0
						0		Desabilita saída VFO 1 (saída parada)
						1		Habilita saída VFO 1
0	0	0	0	0	0			Sempre zeros

Notas das tabelas:

1 – Se comandos de escrita de frequência e duty cycle são feitos nas duas saídas antes de habilitar as mesmas, quando elas forem habilitadas ambas iniciarão em fase. Porém se já estiverem habilitadas e uma nova escrita for feita, mesmo sendo os comandos executados na mesma chamada da função, as saídas poderão sair de fase. Isto se deve ao fato de cada escrita ser realizada em momentos distintos dentro da função e no momento da escrita os novos valores já entram em operação.

Entradas e Saídas

Descrição das entradas:

- **Habilita** - quando esta entrada está energizada a função é chamada, sendo analisados os parâmetros programados na instrução CHF. Caso os mesmos estejam incorretos, a respectiva saída de erro da função é energizada. Se estiverem corretos, os comandos contidos nos bytes auxiliares %XXXXX são executados, realizando as operações de leitura e escrita conforme especificado. Quando a entrada habilita estiver desenergizada, as instruções anteriormente enviadas ao módulo são mantidas, não executando nenhuma operação de leitura ou escrita no mesmo.

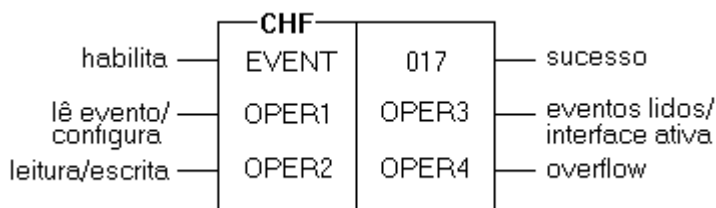
Descrição das saídas:

- **Processamento ok** – é ativada para indicar sucesso na tentativa de realizar a operação solicitada. Sempre que a operação for concluída com sucesso com sucesso essa saída é ligada e as restantes são desligadas. Se esta saída estiver desligada significa que algo de errado ocorreu no processo, sendo que o resultado pode ser ou não indicado numa das outras duas saídas.
- **Erro parâmetros** - é ativada quando existe alguma inconsistência nos parâmetros passados para a função.
- **Erro na faixa de Freq/Duty** - é ativada quando um de um operando um valor de frequência estiver fora da faixa de 1 Hz a 20 kHz ou quando um valor de duty cycle estiver fora da faixa de 0 a 100%.
- **Todas as saídas desligadas** - indica erro na comunicação com os dispositivos selecionados.

Utilização

Esta função pode ser utilizada nas UCPs GR350, GR351, GR370 e GR371.

F-EVENT.017 - Função para Acesso ao Módulo Registro de Eventos



Introdução

A função **F-EVENT.017** realiza o acesso às interfaces AL-3130, AL-3132 e AL-3138, que possuem 32 entradas digitais com registro de eventos.

As interfaces AL-3130, AL-3132 e AL-3138 são equivalentes para o software. Nesta descrição da **F-EVENT** toda a referência à AL-3130 vale também para as demais interfaces, exceto quando houver referência explícita.

A interface AL-3130 registra as variações de estado nas suas entradas com precisão de 1 milissegundo, permitindo a monitoração de eventos digitais em tempo real. A interface armazena os eventos em uma memória local, de forma independente, enquanto a UCP processa o programa aplicativo. A função **F-EVENT** transfere esses eventos da memória da interface para uma tabela, definida na instrução de chamada CHF.

Cada AL3130 presente no barramento do CP contém um relógio interno sincronizado com o relógio da UCP AL-2002, AL-2003 ou AL-2004, eliminando diferenças nos horários dos eventos registrados em interfaces diferentes. Esta sincronização ocorre de forma transparente ao programa aplicativo.

Caso diversos CPs estejam interligados através da rede de sincronismo e da rede ALNET II, os relógios de todas as AL-3130 do sistema mantêm o horário do relógio do CP gerador do sincronismo.

A interface também registra mudanças no horário do seu relógio, ocorridas devido a eventuais acertos no relógio da UCP através das funções **F-RELG.048** ou **F-SINC.049**, para a correta avaliação dos dados dos eventos registrados.

Para maiores informações a respeito da sincronização dos controladores, consultar o Manual de Utilização das UCPs AL-2002, AL-2003 e AL-2004.

Programação

Antes de programar a chamada da função, deve-se declarar a interface na posição do barramento em que está alojada, através da janela de edição do módulo C. Pode-se declará-la com dois códigos diferentes:

- * AL-3130 / AL-3132 / AL-3138 - **somente eventos**
- * AL-3131 / AL-3133 / AL-3139 - **eventos e entradas**

Se declarada como AL-3130 (ou AL-3132, AL-3138), a interface não é acessada pela varredura de E/S do CP, não sendo reservados operandos %E para os seus octetos de entrada. Desta forma, a interface somente registra as transições das suas entradas e o programa aplicativo não pode processar os valores dos pontos como em um módulo de entrada digital.

Caso seja declarado como AL-3131 (ou AL-3133, AL-3139), operandos %E são associados aos pontos de entrada, e o CP os acessa em sua varredura de E/S. Assim, além de registrar eventos, os valores dos pontos de entrada do módulo podem ser usados pelo programa aplicativo.

ATENÇÃO:

A atualização dos pontos de entrada do módulo para operandos %E somente é possível a partir da versão 1.10 do software executivo nas AL-3130. Nas AL-3132 ou AL-3138 sempre é possível.

A instrução CHF possui dois tipos de chamada da função **F-EVENT**:

- chamada de configuração
- chamada de leitura de eventos

A chamada de configuração permite que o programa aplicativo especifique ou leia os parâmetros de funcionamento da interface.

A chamada de leitura de eventos transfere os registros de eventos da interface para a tabela especificada.

Chamada de Configuração

A chamada de configuração é usada para programar ou ler parâmetros de funcionamento do módulo.

Os parâmetros de configuração são:

- as máscaras de desabilitação do registro de eventos
- o tempo de "debounce"
- opções de "overflow" e filtro

Máscaras:

As máscaras de desabilitação permitem a desativação do registro de eventos para pontos usados como entrada normal, evitando que eventos indesejáveis sejam gerados, o que obrigaria o programa a identificá-los e descartá-los. Existem 4 máscaras, cada uma especificando um octeto da interface, cada bit correspondendo a um ponto de entrada. O bit ligado inibe a entrada correspondente para os eventos.

Debounce:

O tempo de "debounce" especifica, em milissegundos, quanto tempo a interface deve ignorar as variações de uma entrada **após** sua mudança de estado. O "debounce" é útil para eliminar ruídos provenientes de rebatimento de contatos. Caso uma entrada seja acionada por um contato seco de relé, por exemplo, evita-se que a vibração deste gere vários eventos na interface, ao invés de um único.

O "debounce" funciona como um filtro, desprezando o ruído após o contato fechar ou abrir. O instante do fechamento ou abertura do contato é registrado, sendo a seguir desprezadas as variações do estado da entrada durante o tempo de "debounce". O tempo pode ser especificado entre 0 e 255 milissegundos, sendo válido para todas as entradas da interface.

Caso se utilize o Filtro Digital (explicado a seguir) juntamente com o "debounce", o tempo especificado para o "debounce" passa a contar a partir do tempo do filtro. Exemplo: caso se especifique um filtro de 10 ms e um debounce de 50 ms, as entradas ficarão mascaradas por um tempo de 60 ms após a ocorrência do evento.

O tempo de debounce também afeta as entradas digitais, caso a interface seja declarada como AL-3131/AL-3133/AL-3139.

A precisão de programação do debounce é +- 10% adicionado de +- 3 milissegundos..

Overflow:

As opções de "overflow" permitem atuar sobre o modo de funcionamento quando novos eventos ocorrerem após o número máximo de eventos armazenáveis na interface for ultrapassado (memória da interface totalmente preenchida). Podem ser selecionadas duas opções:

- opção 0 - mantém eventos antigos
- opção 1 - mantém eventos novos

A opção 0, configurada automaticamente com a energização da AL-3130, mantém na interface os eventos mais antigos. Os novos eventos que ocorrerem após o preenchimento total da memória serão desprezados.

Na opção 1, os novos eventos são preservados, sendo descartados os mais antigos para o armazenamento dos mais novos.

ATENÇÃO:

A memória de armazenamento dos eventos nas interfaces AL-3130 são esvaziadas com a leitura dos eventos pelo programa aplicativo, através da função **F-EVENT.017**. O programa deve realizar leituras periódicas, para evitar que a memória da interface seja totalmente preenchida, com a possível perda do registro de eventos.

Filtro:

As interfaces AL-3130 (32 ou 38) com versão de software executivo igual ou maior que 1.55 possuem um Filtro Digital que pode ser programado para rejeitar pulsos de até 254 ms. Os eventos que passam pelo filtro tem sua estampa de tempo mantida, com resolução de 1 ms. O filtro pode ser programado para entre 10 e 254 ms, podendo serem programados apenas constantes **pares**.

A constante 0 desliga a função filtro. Constantes entre 1 e 9 definem sempre um filtro de 10 ms (valor mínimo da filtragem).

Quando o filtro é utilizado os operandos %EXXXX gerados pela placa são também afetados pelo filtro, sofrendo um atraso proporcional ao tempo do filtro utilizado.

A precisão de programação do filtro é +- 10% adicionado de +- 3 milissegundos.

Configuração

As células da instrução CHF utilizada para a chamada para configuração são programadas do seguinte modo:

- **OPER1**-Especifica o número de parâmetros que são passados para a função em OPER3. Este operando deverá ser obrigatoriamente uma constante memória com valor 2 (%KM+00002).
- **OPER2**-Deve ser um operando do tipo constante memória com valor 0 (%KM+00000). Determina o número de parâmetros possíveis de serem programados na janela de edição de OPER4. Como esta função não necessita de nenhum parâmetro em OPER4, o valor de OPER2 é 0.
- **OPER3**-Contém os parâmetros que são passados para a função, declarados através de uma janela visualizada no MasterTool quando a instrução CHF for editada. O número de parâmetros editáveis é especificado em OPER1, sendo fixo em 2 para esta chamada:

—* _____ %RXXXX -Endereço do barramento onde está alojado a interface AL-3130.

—* _____ %TMXXXX -Especificação da tabela memória onde estão colocados os parâmetros de configuração da interface. Os valores são lidos ou acertados da posição 0 até 5 da tabela. Caso a tabela não esteja declarada ou possua menos que 6 posições, a leitura ou acerto dos valores de configuração não é realizada e todas as saídas da

instrução são desligadas. É possível o uso de tabelas com mais de 6 posições, sendo que a função ignora as posições excedentes. Os valores são lidos ou acertados nos operandos na seguinte seqüência:

Posição Tabela	Conteúdo	Formato
0	Máscara desabilitação octeto 0	00XXX
1	Máscara desabilitação octeto 1	00XXX
2	Máscara desabilitação octeto 2	00XXX
3	Máscara desabilitação octeto 3	00XXX
4	Debounce	00XXX
5	Overflow e filtro	00XXX

Tabela 4-6 Parâmetros de Configuração da Interface

Formatos:

Máscaras de desabilitação do registro de eventos:

Bit da máscara: 7 6 5 4 3 2 1 0

Ponto do octeto: 7 6 5 4 3 2 1 0

Valor do bit: 0 - registro de eventos habilitado

1 - registro de eventos desabilitado

Debounce:

Valor: 0 a 255 milissegundos

Overflow e filtro:

O filtro e a opção de overflow são programados na posição 5 da tabela de configuração, conforme segue:

Posição 5	12	8	4	0
		F F F F	F F F V	
Campos:				
V	- tipo de overflow			0 – mantém eventos antigos 1 – mantém eventos novos
FFFFFFF	- filtro			0 – sem filtro 2 a 10 – filtro de 10 ms >10 – filtro de xx ms (somente valores pares)

- **OPER4** - Não utilizado.

Leitura de Eventos

Na chamada como leitura de eventos, a função transfere para uma tabela memória os dados lidos da interface AL-3130.

As células da instrução CHF utilizada para a chamada da função para leitura de eventos são programadas do seguinte modo:

- **OPER1**-Especifica o número de parâmetros que são passados para a função em OPER3. Este operando deverá ser obrigatoriamente uma constante memória com valor 2 (%KM+00002).
- **OPER2**-Deve ser um operando do tipo constante memória com valor 0 (%KM+00000). Determina o número de parâmetros possíveis de serem programados na janela de edição de OPER4. Como esta função não necessita de nenhum parâmetro em OPER4, o valor de OPER2 é 0.
- **OPER3**-Contém os parâmetros que são passados para a função, declarados através de uma janela visualizada no MasterTool quando a instrução CHF for editada. O número de parâmetros editáveis é especificado em OPER1, sendo fixo em 2 para esta chamada:

—* %RXXXX -Endereço do barramento onde está alojado a interface AL-3130.

—* %TMXXXX -Tabela memória onde são colocados os registros dos eventos lidos da AL-3130. Os registros de evento são colocados a partir da posição definida por um "Apontador" (ver lay-out da tabela %TMXXXX).

Caso a tabela não esteja declarada, a leitura não é realizada e as 3 saídas da instrução são desligadas. A tabela deve possuir no mínimo 12 posições, mas o valor recomendado afim de otimizar a transmissão dos dados em rede é de 64 posições.

- **OPER4** - Não utilizado.

Tamanho da tabela de eventos

Embora a função F-EVENT possa trabalhar com tabelas de 12 até 255 posições, recomenda-se o uso de tabelas de tamanho 64 bytes, conveniente para a transmissão da tabela pela rede.

Formato da tabela de eventos

O "lay-out" para os registros de eventos segue o formato:

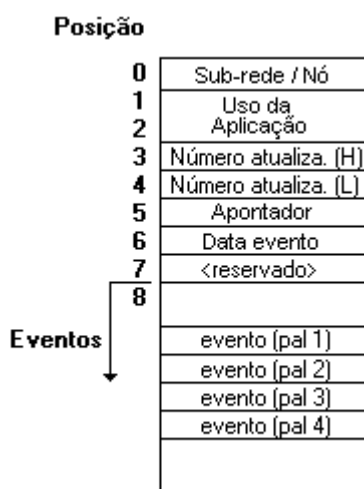


Figura 4-1 "Lay-out" para os Registros de Eventos

—* Na primeira posição a aplicação deve colocar a identificação do CP para o supervisor: sub-rede e nó:

	12				8				4				0			
Ident. CP	0	0	S	S	S	S	S	S	0	0	0	N	N	N	N	N

CAMPOS:

SSSSSS - sub rede(1 a 64)

NNNN - nó (1 a 31)

—* As posições 2 e 3 da tabela são reservadas para uso da aplicação/supervisório.

—* Número atualização: número sequencial que identifica a "atualização" da tabela. É um número de 32 bits que é incrementado pela F-EVENT a cada atualização da tabela. É usado pelos supervisórios ou aplicação para validar os dados lidos.

—* Apontador: posição da tabela que aponta para a próxima posição disponível. Seu valor mínimo é 8, e seu valor máximo é o tamanho da tabela (tabela 100% preenchida). É atualizado pela F-EVENT a cada novo evento colocado na tabela. Deve ser zerado pela aplicação ou supervisor afim de liberar a tabela para novos eventos.

—* Data dos eventos: contém a data (dia e mês) dos eventos carregados na tabela. A função não trata o ano que pode ser lido diretamente da UCP. A posição é mantida pela **F-EVENT** e deve ser inicializada com **zero**. A data tem o formato:

	12				8				4				0			
Data dos eventos	0	0	0	0	MM	MM	0	0	0	D	D	D	D	D	D	D

CAMPOS:

MMMM - mes dos eventos (1 a 12)

DDDDDD - dia dos eventos (1 a 31)

ATENÇÃO:

A data é atualizada pela F-EVENT de acordo com a data atual da UCP e dos eventos gravados nos módulos AL-3130.

—* Eventos: os eventos ocupam quatro posições contíguas da tabela.

Os eventos têm o formato:

				12				8				4				0
Posição n	0	0	1	0	1	0	1	1	0	b	b	b	e	e	e	e
																0
Posição n+1	0	0	0	0	0	0	0	0	0	r	r	r	r	r	r	r
								8								0
Posição n+2	A	D	0	h	h	h	h	h	0	0	m	m	m	m	m	m
								10								0
Posição n+3	s	s	s	s	s	s	l	l	l	l	l	l	l	l	l	l

CAMPOS:

bbb - bit do octeto (0-7)

eeee - estado do evento:

0 - estado 0 da entrada

1 - estado 1 da entrada

14 - mudança no horário: hora anterior

15 - mudança no horário: hora atual

rrrrrr - endereço da AL-3130 (%RXXXX) + N° octeto

A - Atraso de data (somente eeee = 15)

D - Mudança de data (somente eeee = 15)

hhhhh - hora do evento (0 a 23)

mmmmm - minutos do evento (0 a 59)

sssss - segundos do evento (0 a 59)

lllllll - milisegundos (0 a 999)

Obs.1: os valores 14 e 15 do estado não significam que houve uma transição de algum ponto de entrada, mas sim que houve um acerto de horário do relógio da 3130. Neste caso, o valor dos campos llllllll e bbb é sempre igual a zero.

Obs.2: os campos “A” e “D” só são usados nos eventos tipo 15 (acerto de hora / hora atual) e são usados para avisar quando houve um acerto de data/hora que altere o dia (D = 1). “A” (atraso), indica que o acerto foi de atraso (dia anterior > dia atual).

Obs.3: o acerto de hora no CP é feito imediatamente, mas nos CPs escravos em rede e nas placas AL-3130 o acerto acontece 6 segundos após o comando. Eventos gerados nos seis segundos que antecedem ao acerto na rede ficam com a hora anterior.

Obs.4: os eventos tipo “acerto” não atualizam a data na tabela dos eventos. Apenas eventos de entrada o fazem.

EX. 1:

Um evento no bit 5, estado "1", no terceiro octeto da AL-3130 da posição R024, às 14 horas, 30 minutos, 10 segundos e 456 milisegundos, geraria o registro:

Pal 0 0010101101010001

Pal 1 0000000000011010

Pal 2 0000111000011110

Pal 3 0010100111001000

EX. 2:

Um acerto de hora gera dois registros de eventos: o primeiro com a hora anterior e o segundo com a hora atual. Um evento de "hora atual" na AL-3130 do endereço R016, 08:32:25, milisegundos 000, geraria o registro:

Pal 0 0 0 1 0 1 0 1 1 0 0 0 0 1 1 1 1

Pal 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0

Pal 2 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0

Pal 3 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0

Entradas e Saídas

Descrição das entradas:

- **habilita** -quando esta entrada está energizada a função é chamada, sendo analisados os parâmetros programados na instrução CHF. Caso os mesmos estejam incorretos, todas as saídas da instrução são desenergizadas. Se estiverem corretos, a saída sucesso retorna ligada.

ATENÇÃO:

A função realiza procedimentos de inicialização e acerto de horário da interface de forma automática. Devido a este fato, deve ser chamada a cada ciclo de varredura do programa aplicativo, mantendo-se a entrada habilita da instrução CHP sempre energizada.

- **lê evento / configura** -quando energizada, a função realiza a configuração da interface. Se desenergizada, realiza a leitura dos eventos.

IMPORTANTE:

A AL-3130 só gera eventos se receber o sinal de sincronismo da UCP. Para que a UCP envie o sincronismo, deve ser declarada como "geradora" ou "receptora" de sincronismo (Declara/Comunicação/Sincronismo).

- **leitura / escrita**

- em modo configuração - quando energizada, os valores contidos na tabela de configuração são transferidos para a interface. Quando desenergizada a configuração é lida da interface para a tabela.

- em modo leitura de eventos - quando energizada, os registros de eventos são transferidos para a tabela especificada, caso existam eventos e caso haja posições livres. Quando desenergizada, os eventos não são transferidos. Esta entrada pode ser usada para inibir a leitura de eventos enquanto se processa a tabela, permitindo assim que a função seja chamada em cada ciclo.

Descrição das saídas:

- **sucesso** - indica que os parâmetros da chamada estão corretos e que a função foi corretamente executada.
 - em modo leitura de eventos: indica também que a interface está ativa.
- **eventos lidos / interface ativa**
 - em modo configuração - se energizada, indica que a interface está ativa.

- em modo leitura de eventos - se energizada, indica que pelo menos um registro de evento foi lido para a tabela na varredura atual. Caso desenergizada, nenhum registro de evento foi lido na varredura atual.

- **overflow**-se ligada, indica que a memória interna do AL-3130 foi totalmente preenchida. Operações de leitura subsequentes, realizadas com sucesso, desligam esta saída. A saída de overflow somente é acionada em modo leitura de eventos.

Diagnósticos

O estado de "interface ativa", acima referenciado, é consequência de vários fatores que devem ser verificados, em caso de haver dificuldades na programação da **F-EVENT**:

- chave troca a quente desligada (posição "RUN")
- a interface não está em erro (led ERR apagado)
- AL-3130 está recebendo sincronismo da UCP (UCP corretamente configurada e conectada)
- AL-3130 está declarado no barramento do CP
- função **F-EVENT** está carregada na UCP

Se a interface não está ativa, as saídas **sucesso** (chamada de eventos) e **interface ativa** (chamada de configuração) não ligam. Caso a **F-EVENT** não tenha sido carregada na UCP, a saída **sucesso** da chamada de configuração também não liga.

Caso falte sincronismo da UCP, os Leds "ERR" das placas AL-3130 piscam em intervalos de dois segundos. Para corrigir a falta de sincronismo, utilizar a janela "sincronismo" do módulo C-xx.000, no MasterTool.

As interfaces AL-3130, AL-3132 ou AL-3138 adquirem eventos mesmo que estejam com a chave de troca quente desligada. Para as interfaces poderem adquirir eventos basta que a UCP envie o sinal de sincronismo.

Utilização

Esta função pode ser utilizada somente na UCP AL-2002/MSP, AL-2003 e AL-2004.

Exemplo de Aplicação

Declaração no barramento:

A interface pode ser declarada de dois modos: Como AL-3130/AL-3132/AL-3138 ou como AL-3131/AL-3133/AL-3139. Se for declarada como **par**, a interface só pode ser usada para tratamento de eventos. Se for declarada como **ímpar**, a interface também pode ser usada como interface de entrada, sem perder a função dos eventos.

Posição	Modelo	Entradas	Saídas	Endereço
00	AL-3131	%E0000-%E0003		%R000
01	AL-3132			%R008

Tabela 4-7 Declaração do Módulo AL-3130 no Barramento

Uso no diagrama de relés:

No exemplo abaixo, a interface é configurada na lógica 000. É interessante que a configuração seja feita sempre a cada varredura, pois a interface pode ser trocada com o CP ligado (troca quente).

O formato da **%TM0000** na chamada de configuração segue o anteriormente descrito.

O relé **%A0000.0** liga sempre que a função é chamada e os parâmetros estão corretos.

O relé %A0000.1 liga quando a interface está ativa. %A0000.1 desligado significa interface inativa ou em erro.

Na lógica 001 a configuração da interface é lida. As posições da tabela %TM0001 tem significado idêntico à %TM0000.

O relé %A0001.0 liga sempre que a função é chamada e os parâmetros estão corretos.

O relé %A0001.1 liga quando a interface está ativa. %A0001.1 desligado significa interface inativa ou em erro.

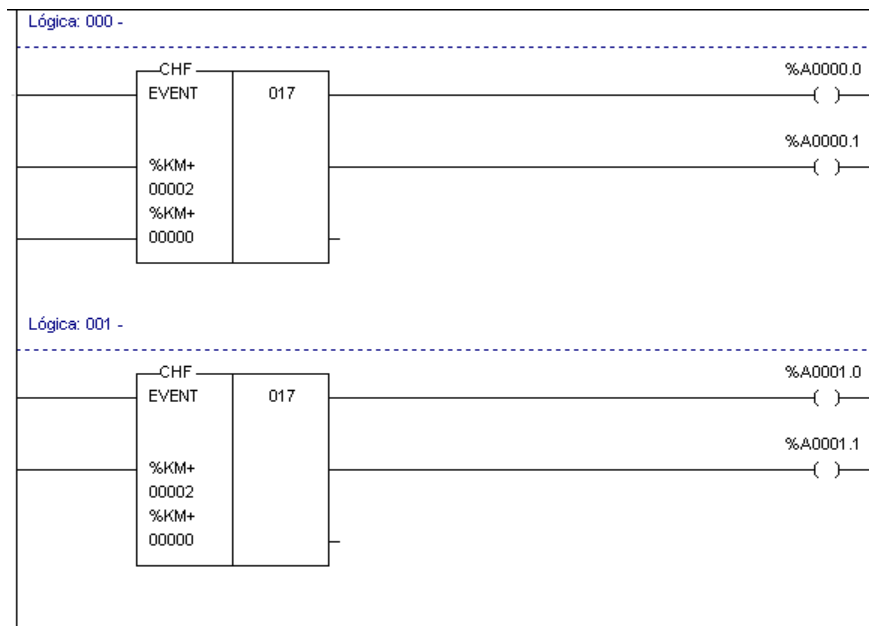


Figura 4-2 Exemplo 1 de Uso do Módulo F-EVENT.017

Lógica 000 - Módulo F-EVENT.017 :

Entrada	Saída
%R0000	
%TM0000	

Lógica 001 - Módulo F-EVENT.017 :

Entrada	Saída
%R0000	
%TM0001	

A lógica 2 mostra a leitura dos pontos da interface. A leitura é possível desde que a interface seja declarado como AL-3131 ou AL-3133 no barramento. A leitura dos pontos é independente das máscaras de configuração, mas são afetados pelo valor do debounce.

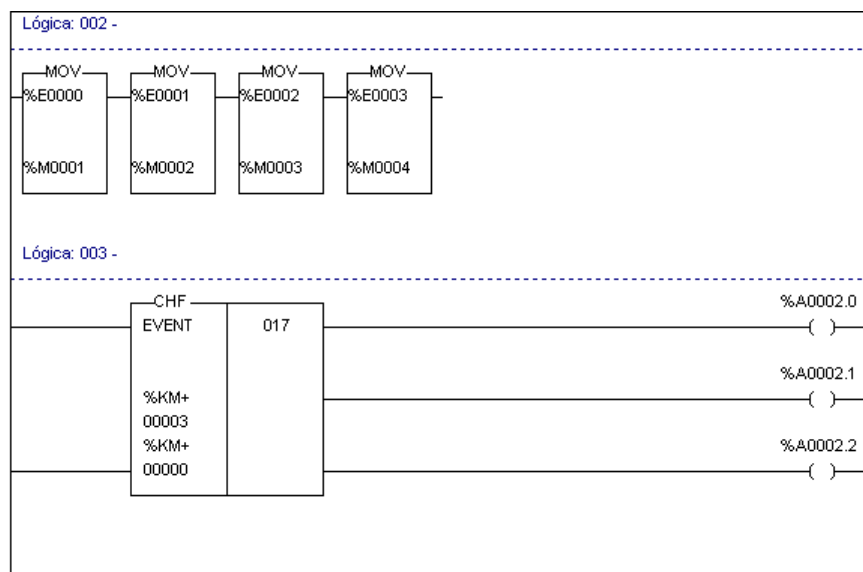


Figura 4-3 Exemplo 2 de Uso do Módulo F-EVENT.017

Lógica 003 - Módulo F-EVENT.017 :

Entrada Saída

%R0000

%TM0002

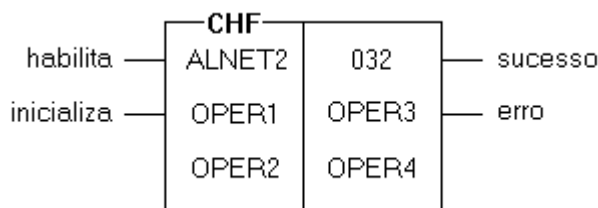
Na lógica 3 do exemplo é mostrada a leitura dos eventos registrados pelo AL-3130. Os eventos são transferidos do AL-3130 para a tabela %TM0002. A posição 5 da %TM0002 aponta sempre a próxima posição disponível da tabela

O relé %A0002.0 liga sempre que a função é chamada, os parâmetros estão corretos e a interface está ativa.

O relé %A0002.1 liga quando uma atualização foi realizada na tabela de eventos na varredura em questão.

O relé %A0002.2 liga quando ocorreu "overflow" na memória interna da interface. O "overflow" desliga à medida que os eventos são retirados da interface para a tabela %TM0002.

F-ALNET2.032 - Função Leitura de Estatísticas da Rede ALNET II



Introdução

A função F-ALNET2.032 permite a leitura e a escrita dos valores dos parâmetros de configuração e das estatísticas de funcionamento do CP na rede ALNET II pelo programa aplicativo.

Programação

Operandos

As células da instrução CHF utilizada para a chamada da função são programadas do seguinte modo:

- **OPER1** - Especifica o número de parâmetros que são passados para a função em OPER3. Este operando deverá ser obrigatoriamente uma constante memória com valor 1 (%KM+00001).
- **OPER2** - Deve ser um operando do tipo constante memória com valor 0 (%KM+00000). Determina o número de parâmetros possíveis de serem programados na janela de edição de OPER4. Como esta função não necessita de nenhum parâmetro em OPER4, o valor de OPER2 é 0.
- **OPER3** - Contém os parâmetros que são passados para a função, declarados através de uma janela visualizada no MasterTool quando a instrução CHF for editada. O número de parâmetros editáveis é especificado em OPER1, sendo fixo em 1 para este módulo:

—* — %MXXXX ou %TMXXXX - Operando memória ou tabela que recebe os valores de estatísticas e parâmetros. Se for utilizado um operando %M, devem estar definidos no mínimo 41 operandos a partir do declarado (incluindo o mesmo) para que a função seja executada corretamente. Caso seja um operando tabela, esta deve possuir pelo menos 41 posições.

- **OPER4** - Não utilizado.

Entradas e Saídas

Descrição das entradas:

- **habilita** - quando esta entrada está energizada a função é chamada, sendo analisados os parâmetros programados na instrução CHF. Caso os mesmos estejam incorretos, todas as saídas da instrução são desenergizadas. Se estiverem corretos, os valores são copiados, sendo acionada a saída sucesso.
- **inicializa** - quando energizada, zera os valores das estatísticas.
- **escreve** - se energizada, transfere os valores dos 3 últimos parâmetros da tabela para as variáveis do CP, forçando o estado das conexões físicas.

Descrição das saídas:

- **sucesso** - é energizada quando a função foi corretamente executada.
- **erro** - é energizada caso ocorra erro na especificação dos operandos ou tentativa de acesso a operandos não declarados.

ATENÇÃO:

A entrada escreve está implementada a partir da versão 1.20 da F-ALNET2.032. A saída de erro está implementada a partir da versão 1.10.

Descrição dos Valores das Estatísticas e Parâmetros

Operando	Posição Tabela	Conteúdo
Estatísticas de transmissões		
%MXXXX	0	Número de transmissões sem erro
%MXXXX+1	1	Número de transmissões com erro de colisão
%MXXXX+2	2	Número de transmissões com erro de "underrun"
%MXXXX+3	3	Número de transmissões sem recepção de ACK de hardware
%MXXXX+4	4	Número de transmissões canceladas por fim de retentativas
%MXXXX+5	5	Número de "time-outs" de serviço
%MXXXX+6	6	Número de faltas de "buffers" de transmissão
%MXXXX+7	7	Não utilizado
%MXXXX+8	8	Não utilizado
Estatísticas de recepções		
%MXXXX+9	9	Número de recepções sem erro
%MXXXX+10	10	Número de recepções com erro de colisão
%MXXXX+11	11	Número de recepções com erro de "overrun"
%MXXXX+12	12	Número de recepções com erro de CRC
%MXXXX+13	13	Número de recepções com erro de alinhamento
%MXXXX+14	14	Número de recepções de pacotes com erro de tamanho
%MXXXX+15	15	Número de "time-outs" de pacote
%MXXXX+16	16	Número de faltas de "buffers" de recepção
%MXXXX+17	17	Não utilizado
%MXXXX+18	18	Velocidade de comunicação (ver tabela a seguir)
%MXXXX+19	19	Endereço da estação
%MXXXX+20	20	Endereço da sub-rede local
%MXXXX+21	21	Grupos de "multicast"
%MXXXX+22	22	Endereço do gateway 1
%MXXXX+23	23	Endereço do gateway 2
%MXXXX+24	24	"Time-out" intrabarramento (décimos de segundo)
%MXXXX+25	25	"Time-out" interbarramento (décimos de segundo)
%MXXXX+26	26	"Time-out" de pacote (décimos de segundo)
%MXXXX+27	27	Número de retentativas de retransmissão
%MXXXX+28	28	Tipo de conexão física (0 - elétrica, 1 - ótica)
%MXXXX+29	29	Redundância de conexões físicas (0 - sem, 1 - com)
%MXXXX+30	30	Período para teste da redundância (segundos)

Operando	Posição Tabela	Conteúdo
Parâmetros de configuração		
%MXXXX+31	31	Tempo para comutação da conexão física (segundos)
%MXXXX+32	32	Não utilizado
%MXXXX+33	33	Não utilizado
%MXXXX+34	34	Conexão física selecionada (1 ou 2)
%MXXXX+35	35	Estado da conexão 1 (0 - normal, 1 - falha)
%MXXXX+36	36	Estado da conexão 2 (0 - normal, 1 - falha)
%MXXXX+37	37	Não utilizado
Parâmetros de configuração possíveis de serem escritos		
%MXXXX+38	38	Conexão física forçada (0 - não forçada, 1 ou 2)
%MXXXX+39	39	Estado forçado da conexão 1 (0 - normal, 1 - falha)
%MXXXX+40	40	Estado forçado da conexão 2 (0 - normal, 1 - falha)

Tabela 4-8 Descrição dos Valores das Estatísticas e Parâmetros

Velocidade de comunicação:	0	-	2 Mbaud
	1	-	1 Mbaud
	2	-	500 Kbaud
	3	-	250 Kbaud
	4	-	125 Kbaud
	5	-	64 Kbaud
	6	-	25 Kbaud

Maiores informações sobre o significado das estatísticas e parâmetros podem ser encontrados no Manual de Utilização ALNET II.

Utilização

Esta função pode ser utilizada nas UCPs AL-2000/MSP, AL-2002/MSP, AL-2003, AL-2004 e QK2000/MSP.

Exemplo de Aplicação

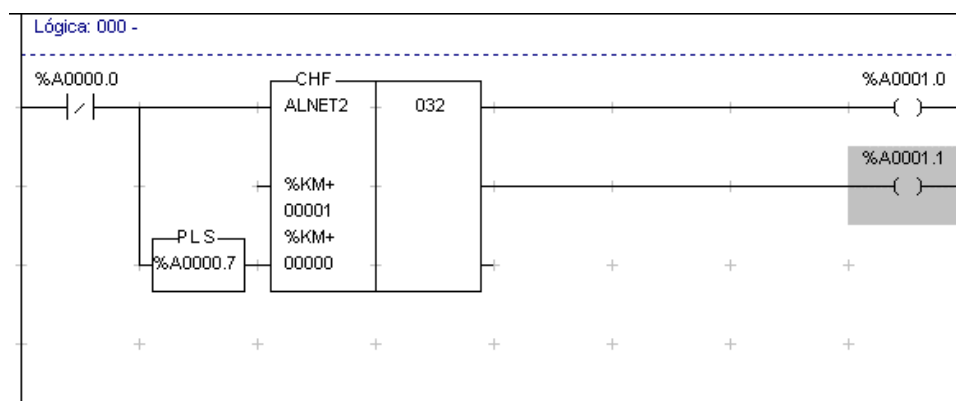


Figura 4-4 Exemplo de Uso do Módulo Função F-ALNET.032

Lógica 000 - Modulo F-ALNET2.032 :

Entrada

Saída

%TM0000

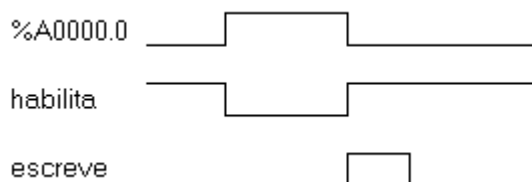


Figura 4-5 Diagrama de Tempos do Exemplo de F-ALNET.032

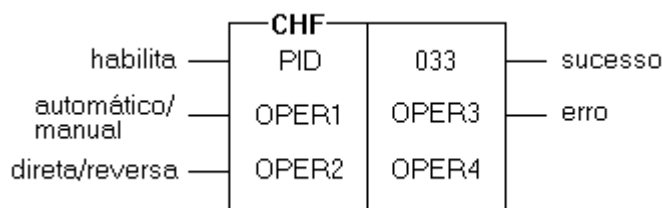
Seja a %TM0000 uma tabela com 41 posições.

Para que seja possível escrever os 3 únicos parâmetros de estado da rede ALNET II que podem ser alterados pelo programa aplicativo (ver parâmetros 38, 39, 40), deve-se primeiro retirar o sinal de habilitação da função.

A seguir, colocam-se os valores desejados nas posições correspondentes aos parâmetros na %TM0000 (posições 38, 39 e 40). Neste momento, o sinal de habilitação é novamente ativado, quando é gerado um pulso de escrita por uma varredura.

O pulso de escrita causará a alteração dos parâmetros desejados. Independentemente do pulso de escrita, enquanto o sinal de habilitação estiver ativo, a %TM0000 estará sendo atualizada com o conjunto de estatísticas e parâmetros da rede ALNET II.

F-PID.033 - Função Controle PID



Introdução

A função F-PID.033 implementa o algoritmo de controle proporcional, integral e derivativo. A partir de um valor medido (VM) e do ponto de ajuste desejado (PA) a função calcula o valor de atuação (VA) para o sistema controlado. Este valor é calculado periodicamente, levando em consideração os fatores proporcionais, integrais e derivativos programados. O diagrama em blocos da função é mostrado na figura 4-6.

As características mais importantes apresentadas pelo laço de controle implementado são:

- desaturação da ação integral (anti-reset windup)
- acompanhamento da saída no modo manual e comutação manual/automática balanceada (output tracking e bumpless transfer)
- ação direta ou reversa
- limites de saída máximo e mínimo ajustáveis
- ação derivativa calculada sobre várias amostragens
- capacidade de realizar integral discreta
- deslocamento com sinal
- tempo de execução de 1,6 ms no pior caso
- resolução de saída de 1:1000

—* **%MXXXX** - Memória que contém o valor de atuação no processo, geralmente acionando uma instrução D/A.

—* **%AXXXX** - Octeto auxiliar que contém pontos de controle da função PID.

- **OPER4** - Não utilizado.

Entradas e Saídas

Descrição das entradas:

- **habilita** - quando esta entrada está energizada a função é chamada, sendo analisados os parâmetros programados na instrução CHF. Caso o número de parâmetros ou seu tipo sejam diferentes das necessidades da função, todas as saídas da instrução são desenergizadas. Se estiverem corretos, o cálculo do controle PID é realizado.
- **automático(0)/manual(1)** - quando energizada, o operando de atuação não recebe o valor calculado pela função (modo manual).
- **direta(0)/reversa(1)** - especifica a forma de ação do controle.

Descrição das saídas:

- **sucesso** - é energizada quando a função foi corretamente executada.
- **erro** - é energizada caso ocorra erro na especificação dos operandos ou tentativa de acesso a operandos não declarados.

Parâmetros Adicionais

Além dos operandos programados na instrução de chamada CHF, outros parâmetros devem ser carregados na tabela declarada em OPER3. Esta tabela deve conter 16 posições, sendo utilizada para definir os parâmetros utilizados pelo algoritmo de controle e armazenar resultados intermediários. A tabela 4-9 apresenta os parâmetros que devem ser carregados em cada posição de tabela, bem como seus valores mínimos e máximos.

Po	Parâmetro armazenado	Fórm.	Variação permitida	Valor tab
00	Ganho proporcional x 10	GP x 10	GP: 1,0 a 100,0	10 a 1000
01	Fat integral - parte frac.	dt / GI	GI: 1 a 1000 s/rep	0,0001 a 10,000
02	Fat integral - parte int		dt: 0,1 a 10 s	
03	Fat derivativo - parte frac.	GD / 3dt	GD: 1 a 1000 s	0,0333 a 3333,3333
04	Fat derivativo - parte int		dt: 0,1 a 10 s	
05	Deslocamento	DE	0 a 1000	0 a 1000
06	Valor mínimo da saída		0 a 1000	0 a 1000
07	Valor máximo da saída		0 a 1000	0 a 1000
08	Não utilizada			
09	Variável medida N - 1			0 a 1000
10	Variável medida N - 2			0 a 1000
11	Variável medida N - 3			0 a 1000
12	Erro			0 a 1000
13	Ação proporcional x 10			0 a 65535
14	Ação integral - parte frac x 10			0 a 65535
15	Ação integral - parte int x 10			0 a 65535

Tabela 4-9 Parâmetros Adicionais da PID

Para possibilitar uma maior velocidade de execução, alguns parâmetros devem ser carregados na tabela já pré-calculados. Sendo valores relativamente fixos, evita-se desta forma que sejam recalculados a cada chamada da função.

Os parâmetros que devem ser pré-calculados são:

- Ganho proporcional X 10 (posição 0) - É calculado multiplicando-se o ganho proporcional desejado por 10.
- Fator multiplicativo integral (posições 1 e 2) - É calculado dividindo-se o intervalo de amostragem (dt) pelo ganho integral desejado. A unidade de dt é segundos, sendo o seu valor mínimo de 0,1 segundos e máximo de 10,0 segundos e deve ser igual ao intervalo de tempo em que a rotina é executada. A unidade de GI é segundos/repetição, podendo variar de 1 até 1000 segundos/repetição. GI igual a 1 segundo/repetição significa o máximo efeito integral.
- Fator multiplicativo derivativo (posições 3 e 4) - É calculado dividindo-se o ganho derivativo (GD) pelo intervalo de amostragem (dt) e pelo valor 3. A unidade de GD é segundos, podendo variar de 1 até 1000 segundos. GD igual a 1000 segundos significa máximo efeito derivativo. Recomenda-se que quanto maior o valor de GD, maior deve ser o intervalo de amostragem. Mesmo para valores de GD = 1 segundo, o intervalo de amostragem deve ser maior que 0,2 segundos. Caso não seja tomado este cuidado, o termo derivativo produzirá apenas "ruído" e a ação de controle será muito brusca.
- Deslocamento (posição 5) - Permite que seja introduzido um deslocamento ("bias") no valor de atuação, evitando que erros negativos causem saturação no valor mínimo de saída. Geralmente este valor é ajustado para 50% (500) ou igual ao ponto de ajuste, se o ganho proporcional é pequeno.
- Valores mínimo e máximo de saída (posições 6 e 7) - São valores opcionais que limitam a excursão do valor de atuação, podendo ser modificados dinamicamente em função das condições operacionais. Se o valor máximo for maior ou igual a 1000 e o valor mínimo igual a 0, não é realizada nenhuma limitação.

O valor medido, o valor de atuação, o deslocamento, os valores máximo e mínimo têm como variação a faixa de 0 a 1000, o que corresponde a uma variação de 0 a 100% nas variáveis do processo.

As demais posições da tabela são utilizadas exclusivamente pela função PID, não devendo ser modificadas pelo programa aplicativo. A posição 12 (erro) pode ser consultada pelo programa. As posições 14 e 15 acumulam o fator integral, podendo serem zeradas, se necessário. Recomenda-se que estas posições estejam zeradas no início do processamento para evitar que valores aleatórios fiquem armazenados.

Além da tabela de parâmetros, alguns pontos de controle são utilizados pela função, contidos no octeto auxiliar especificado (%XXXXX).

- %XXXXX.4 - Sinal da ação integral - É utilizado pela função PID. Quando desenergizado, o termo integral é positivo, caso contrário é negativo. Pode ser lido pelo programa, se desejado.
- %XXXXX.5 - Sinal do deslocamento Indica para a função qual é o sinal do deslocamento, devendo ser acionado pelo programa. O ponto desenergizado indica deslocamento positivo. Quando energizado, o deslocamento é negativo.
- %XXXXX.6 - Inibe ação derivativa - Quando energizado, a função não executa a ação derivativa.
- %XXXXX.7 - Inibe ação integral - Quando energizado, a ação integral não é calculada, permanecendo a sua atribuição como o último valor calculado antes da inibição, a menos que os valores limites sejam excedidos.

Características do Funcionamento

A desaturação da ação integral (anti-reset windup) é feita de modo a evitar que o termo integral continue a acumular erro quando um distúrbio no processo causa a saturação da saída do controlador em alguns dos limites. No momento em que o valor de saída atinge algum dos limites (máximo ou mínimo), o termo integral é fixado em seu valor corrente, impedindo o seu crescimento indefinido, sem influenciar na saída. Isto assegura que haverá uma resposta do controlador tão logo desapareça o distúrbio que o levou a saturar a saída.

A função pode ser executada em modo manual, energizando-se a segunda entrada da instrução CHF. Neste modo, a rotina não mais modifica o valor da saída de atuação, mas o acompanha (output tracking). Isto é, em função do valor da saída fixo e do valor medido do processo, os termos proporcional e derivativo são calculados e o termo integral é forçado para um valor adequado, de modo que, quando ocorrer a transição de manual para automático, a rotina possa reassumir o controle com o valor inicial da saída igual ao último valor da saída no modo manual. Chama-se este fato de comutação manual/automática balanceada (bumpless transfer).

A forma de controle pode ser direta ou reversa. Esta seleção é realizada desenergizando ou energizando a terceira entrada da instrução CHF. Caso o processo seja tal que o valor medido cresce quando o valor da saída de atuação cresce, a ação direta deve ser selecionada. Se o valor medido decresce com o aumento da saída de atuação, então a ação reversa deve ser utilizada.

O intervalo entre amostragens de um laço PID pode variar de 0,1 a 10,0 segundos. É de responsabilidade do usuário programar um "disparador" da função, ou seja, um trecho de programa aplicativo que somente habilite a rotina PID nos intervalos de tempo desejados. Note-se ainda que o valor do intervalo de amostragem usado para o cálculo dos fatores multiplicativos integral e derivativo deve coincidir com o intervalo de tempo das chamadas do "disparador". Como cada execução da rotina pode despendar até 3 ms, é aconselhável que cada laço de controle diferente seja disparado em diferentes varreduras do programa.

Exemplo de Aplicação

Como exemplo de utilização, sejam os seguintes valores de ajuste desejados para um laço de controle:

$$\begin{aligned} \text{PA} &= 62 \\ \text{GP} &= 5 \quad (\text{GP} = 100 / \text{banda proporcional em \%}) \end{aligned}$$

GI	=	100 segundos/repetição
GD	=	5 segundos
dt	=	1 segundo
DES	=	50%
MAX	=	80%
MIN	=	0%

Os valores que devem ser carregados na tabela de parâmetros são:

Posição	Valor	
0	50	GP X 10 (50)
1	100	dt / GI (0,0100)
2	0	
3	6666	GD / 3dt (1,6666)
4	1	
5	500	DES
6	0	MIN
7	800	MAX
8	620	PA

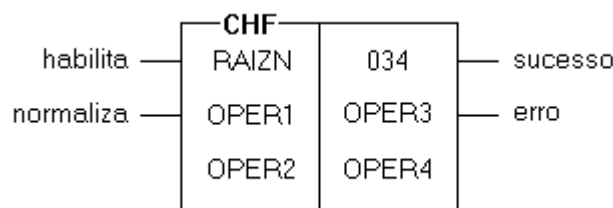
Utilização

Esta função pode ser utilizada nas UCPs AL-600, AL-2000/MSP, AL-2002/MSP, AL-2003, AL-2004, QK800, QK801, QK2000/MSP, PL102, PL103, PL104 e PL105.

ATENÇÃO:

O módulo F-PID.033 pode ser utilizado no CP AL-2000/MSP somente a partir da versão 1.10 do software executivo.

F-RAIZN.034 - Função Raiz Quadrada



Introdução

A função F-RAIZN.034 extrai a raiz quadrada de um valor fornecido em um operando memória ou real. No caso de operandos memória, o resultado pode ser normalizado para uma escala previamente definida.

O cálculo realizado corresponde à seguinte expressão:

$$\text{Op Destino} = \text{Raiz Quadrada (Op Fonte)} * \text{Constante de Normalização} / 256$$

A normalização executada em conjunto com o processamento da raiz quadrada garante uma boa precisão dos resultados, pois são utilizadas variáveis internas com maior capacidade de armazenamento do que os operandos memória.

Esta função é tipicamente utilizada na linearização das leituras de transdutores que fornecem valores em escala quadrática, ou seja, com a saída proporcional ao quadrado do sinal medido.

Programação

Operandos

As células da instrução CHF utilizada para a chamada da função são programadas do seguinte modo:

- **OPER1** - Especifica o número de parâmetros que são passados para a função em OPER3. Este operando deverá ser obrigatoriamente uma constante memória com valor 3 (%KM+00003).
- **OPER2** - Deve ser um operando do tipo constante memória com valor 0 (%KM+00000). Determina o número de parâmetros possíveis de serem programados na janela de edição de **OPER4**. Como esta função não necessita de nenhum parâmetro em OPER4, o valor de OPER2 é 0.
- **OPER3** - Contém os parâmetros que são passados para a função, declarados através de uma janela visualizada no MasterTool quando a instrução CHF for editada. O número de parâmetros editáveis é especificado em OPER1, sendo fixo em 3 para este módulo:

—* %MXXXX ou %FXXXX - Operando com o valor a ser extraída a raiz quadrada (fonte). Este valor deve ser positivo para que o cálculo seja realizado.

—* %MXXXX ou %KM+XXXXX - Operando memória ou constante para a normalização de fundo de escala da raiz quadrada extraída. O valor programado é dividido por 256 e multiplicado pela raiz do operando fonte, gerando o valor do operando destino, quando a segunda entrada da instrução estiver energizada. Este operando não é utilizado quando os operandos fonte e destino são do tipo real.

—* %MXXXX ou %FXXXX - Operando que recebe o resultado da raiz quadrada normalizada (destino). Deve necessariamente ser do mesmo tipo que o operando fonte.

- **OPER4** - Não utilizado.

Entradas e Saídas

Descrição das entradas:

- **habilita** - quando esta entrada está energizada a função é chamada, sendo analisados os parâmetros programados na instrução CHF. Caso os mesmos estejam incorretos, todas as saídas da instrução são desenergizadas. Se estiverem corretos, os cálculos são realizados, sendo acionadas as saídas sucesso ou erro.
- **normaliza** - quando energizada, realiza o ajuste de fundo de escala para o valor da raiz quadrada obtido. Se desenergizada, o valor do operando memória destino receberá simplesmente a raiz quadrada do operando fonte.

Descrição das saídas:

- **sucesso** - indica que o cálculo da raiz e a sua normalização foram realizados corretamente. Quando desenergizada, indica que a entrada habilita não está acionada, o módulo não está carregado no CP, os operandos não foram corretamente definidos ou existem valores negativos armazenados nos mesmos.
- **erro** - esta saída é energizada sempre que ocorre um dos seguintes erros:
 - * existem valores negativos no operando fonte ou na constante de normalização
 - * erro na especificação dos operandos ou tentativa de acesso a operandos não declarados
 - * operando fonte com tipo diferente do operando destino

ATENÇÃO:

Na versão 1.00 de F-RAIZN.034 a saída de erro não é acionada na tentativa de acesso a operandos não declarados.

Utilização

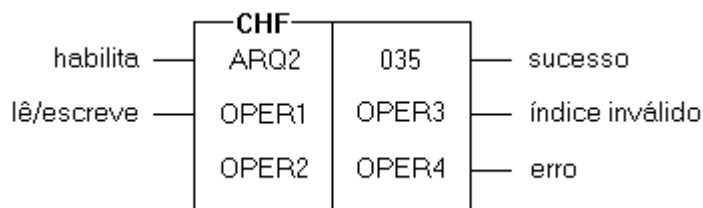
Esta função pode ser utilizada nas UCPs AL-600, AL-2000/MSP, AL-2002/MSP, AL-2003, AL-2004, QK800, QK801 e QK2000/MSP.

Exemplo de Aplicação

Para normalizar o valor do operando destino de forma que tenha a mesma escala do operando fonte, o valor a ser declarado no operando de normalização deve ser igual à raiz quadrada do valor do operando fonte multiplicado por 256.

Por exemplo, seja o caso de um transdutor que fornece valores de 0 a 1024, proporcionais ao quadrado de uma vazão, e deseja-se que estes valores sejam linearizados para a mesma escala de valores (0 a 1024). A constante de normalização programada é 8192 (raiz quadrada (1024) * 256).

F-ARQ2.035 a F-ARQ31.042 - Funções Arquivo de Dados



Introdução

As funções arquivo de dados permitem o uso da memória do programa aplicativo para armazenar grandes quantidades de informações, utilizando conceitos de registros e campos. Desta forma obtém-se grande flexibilidade no aproveitamento dos bancos de memória do CP, além de um aumento substancial na capacidade de armazenamento de dados.

Existem diversas funções que implementam arquivos de dados, sendo idênticas no modo de programação e no funcionamento, diferindo-se apenas quanto à capacidade de armazenamento. Os módulos disponíveis são:

- F-ARQ2.035 - Função arquivo com 2 Kbytes de dados
- F-ARQ4.036 - Função arquivo com 4 Kbytes de dados
- F-ARQ8.037 - Função arquivo com 8 Kbytes de dados
- F-ARQ12.038 - Função arquivo com 12 Kbytes de dados
- F-ARQ15.039 - Função arquivo com 15 Kbytes de dados
- F-ARQ16.040 - Função arquivo com 16 Kbytes de dados
- F-ARQ24.041 - Função arquivo com 24 Kbytes de dados
- F-ARQ31.042 - Função arquivo com 31 Kbytes de dados

Cada arquivo pode possuir até 255 registros, numerados de 0 a 254, sendo que cada registro pode possuir até 255 campos, também numerados de 0 a 254. Note-se, entretanto, que a quantidade total de memória ocupada não pode exceder a capacidade do módulo.

Cada campo ocupa o mesmo número de bytes do operando onde serão realizadas as escritas ou leituras do arquivo.

Programação

Operandos

As células da instrução CHF utilizada para a chamada da função são programadas do seguinte modo:

- **OPER1** - Especifica o número de parâmetros que são passados para a função em OPER3. Este operando deverá ser obrigatoriamente uma constante memória com valor 5 (%KM+00005).
- **OPER2** - Especifica o número de parâmetros que são passados para a função em OPER4. Este operando deverá ser obrigatoriamente uma constante memória com valor 0 (%KM+00000).
- **OPER3** - Contém os parâmetros que são passados para a função, declarados através de uma janela visualizada no MasterTool quando a instrução CHF for editada. O número de parâmetros editáveis é especificado em OPER1, sendo fixo em 5 para este módulo:

—* %MXXXX, %DXXXX, %AXXXX, %EXXXX,
%SXXXX, %TMXXXX, %TDXXXX, %KM+XXXXXX ou %KD+XXXXXXX - Operando de onde os dados

serão lidos nas operações de escrita no arquivo ou para onde os dados serão copiados nas leituras do arquivo (parâmetro 1).

—* %MXXXX - Número do registro de/para o qual será copiado o dado (parâmetro 2). Deve estar compreendido entre 0 e o número de registros totais menos 1.

—* %MXXXX - Número do campo de/para o qual será copiado o dado (parâmetro 3). Deve estar compreendido entre 0 e o número de campos totais menos 1.

—* %KM+XXXXX - Número de registros totais (1 a 255) desejados para o arquivo (parâmetro 4).

—* %KM+XXXXX - Número de campos totais (1 a 255) desejados para o arquivo (parâmetro 5).

- **OPER4** - Não utilizado.

Entradas e Saídas

Descrição das entradas:

- **habilita** - quando esta entrada está energizada a função é chamada, sendo analisados os parâmetros programados na instrução CHF. Caso o número de parâmetros ou seu tipo sejam diferentes das necessidades do módulo, a saída de erro é energizada. Se estiverem corretos, é realizada uma tentativa de acesso ao arquivo.
- **lê/escrive** - quando energizada, o valor do primeiro parâmetro é copiado para o registro e o campo especificados no segundo e terceiros parâmetros. Se estiver desenergizada, o valor é lido do campo e copiado para o primeiro parâmetro.

Descrição das saídas:

- **sucesso** - indica que o acesso ao arquivo de dados foi corretamente realizado.
- **índice inválido** - esta saída é ligada se:
 - * o campo a ser lido ou escrito não foi especificado
 - * a declaração de registros e campos excede a capacidade de memória do módulo
 - * houver tentativa de leitura quando o primeiro parâmetro for uma constante
 - * houver tentativa de escrita estando o módulo armazenado em memória EPROM
- **erro** - é energizada caso ocorra erro na especificação dos parâmetros ou tentativa de acesso a operandos não declarados.

Utilização

Esta função pode ser utilizada nas UCPs AL-600, AL-2000/MSP, AL-2002/MSP, AL-2003, AL-2004, QK800, QK801, QK2000/MSP, PL104 e PL105.

Descrição do Funcionamento

Para declaração correta do número de campos e registros do arquivo, deve ser realizado o seguinte cálculo:

$$\text{Ocupação do arquivo} = \text{Núm. registros} \times \text{Núm. campos} \times \text{Núm. bytes por campo}$$

(parâmetro 4)

(parâmetro 5)

O número de bytes por campo ocupado por cada tipo de operando pode ser obtido da tabela 4-10.

Parâmetro 1	Número de bytes por campo
%MXXXX	2
%DXXXX	4
%AXXXX	1
%EXXXX	1
%SXXXX	1
%TMXXXX	2
%TDXXXX	4
%KM+XXXXX	2
%KD+XXXXXXXX	4

Tabela 4-10 Ocupação dos Campos dos Arquivos

O valor obtido no cálculo anterior deve ser menor ou igual à capacidade total da função utilizada, conforme a tabela 4-11.

Função	Capacidade (bytes)
F-ARQ2.035	2048
F-ARQ4.036	4096
F-ARQ8.037	8192
F-ARQ12.038	12288
F-ARQ15.039	15360
F-ARQ16.040	16384
F-ARQ24.041	24576
F-ARQ31.042	31744

Tabela 4-11 Capacidade das Funções Arquivos de Dados

ATENÇÃO:

Em um mesmo programa podem ser declaradas diversas instruções CHF para acesso ao mesmo arquivo. Em todas estas instruções os operandos com os valores a serem escritos ou que recebem os valores lidos (parâmetro 1 em OPER3) devem possuir o mesmo número de bytes por campo (ver tabela 4-5).

Portanto, é possível ler ou escrever indistintamente operandos %E, %S e %A de um arquivo ou %KM, %M e %TM de outro. Contudo, não devem ser realizados acessos com operandos %M ou %D em um mesmo arquivo.

Se o primeiro parâmetro for uma tabela (%TM ou %TD), todos os campos do registro indicado no segundo parâmetro são copiados, ou seja, a transferência de dados é realizada entre o registro e a tabela, sendo que o valor do terceiro parâmetro (número do campo) é ignorado.

Caso a tabela possua menos posições do que o número de campos do registro, serão transferidos somente os campos que correspondem às posições existentes. Caso a tabela possua mais posições do que o número de campos do registro, serão transferidos somente os campos existentes.

A operação de escrita de dados copia-os para a própria área de memória ocupada pelo módulo função.

ATENÇÃO:

Caso o módulo F-ARQ esteja armazenado em Flash EPROM, não é possível escrever dados no arquivo, somente ler dados. Para que a escrita de dados nos arquivos ser realizada, os módulos F correspondentes aos mesmos devem estar na memória RAM do programa aplicativo.

ATENÇÃO:

Durante a leitura de um módulo arquivo de dados do CP com o programador MasterTool ou durante a sua transferência de RAM para Flash, não deve ser realizada nenhuma escrita de dados no mesmo

Isto porque a escrita de dados modifica o módulo lido, sendo considerado inválido pelo programador ou pelo CP devido à alteração do seu "checksum".

As funções arquivos de dados são módulos de programa aplicativo podendo ser carregados ou lidos do CP e armazenados em disquetes. Por exemplo, seja o caso de um CP controlando uma máquina injetora, armazenando diversos parâmetros de configuração em um módulo F-ARQ8.037. Depois que os parâmetros forem armazenados, este módulo F pode ser lido e armazenado em um disquete, para carga em outras máquinas injetoras iguais.

Exemplo de Aplicação

Como exemplo, caso seja desejado um arquivo com 120 registros e com 8 campos por registro para armazenar operandos %D, a ocupação de memória será:

Ocupação do arquivo = 120 registros X 8 campos/registro X 4 bytes/campo

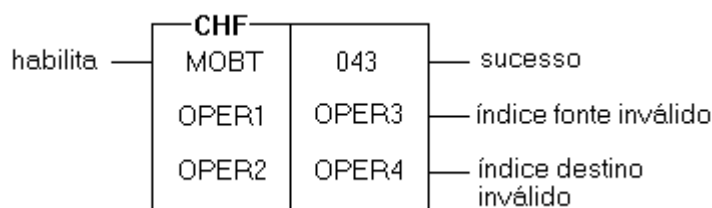
Ocupação do arquivo = 3840

A configuração desejada ocupa 3840 bytes, devendo ser utilizado o módulo F-ARQ4.036, pois o mesmo permite o armazenamento de até 4096 bytes.

Os parâmetros programados em OPER3 da instrução CHF utilizada para o acesso ao arquivo são:

- %D0020 - operando para onde será lido ou com o valor a ser escrito no arquivo
- %M0100 - contém o número do registro a ser lido ou escrito, devendo estar compreendido entre 0 e 119 (120 registros totais)
- %M0101 - contém o número do campo a ser lido ou escrito, devendo estar compreendido entre 0 e 7 (8 registros totais)
- %KM+00120 - número total de registros
- %KM+00008 - número total de campos

F-MOBT.043 - Função para Movimentação de Blocos de Operandos Tabela



Introdução

A função F-MOBT.043 realiza a cópia de blocos de operandos numéricos (%M ou %D) ou posições de tabelas (%TM ou %TD). Podem ser copiados até 255 valores de operandos simples para tabelas e vice-versa, transferindo-se também posições de uma tabela para outra. É possível especificar a posição inicial do bloco a ser copiado na tabela fonte e na tabela destino.

Programação

Operandos

As células da instrução CHF utilizada para a chamada da função são programadas do seguinte modo:

- **OPER1** - Especifica o número de parâmetros que são passados para a função em OPER3. Este operando deverá ser obrigatoriamente uma constante memória com valor 5 (%KM+00005).
- **OPER2** - Deve ser um operando do tipo constante memória com valor 0 (%KM+00000). Determina o número de parâmetros possíveis de serem programados na janela de edição de OPER4. Como esta função não necessita de nenhum parâmetro em OPER4, o valor de OPER2 é 0.
- **OPER3** - Contém os parâmetros que são passados para a função, declarados através de uma janela visualizada no MasterTool quando a instrução CHF for editada. O número de parâmetros editáveis é especificado em OPER1, sendo fixo em 5 para este módulo:

—*_____ %MXXXX, %DXXXX, %TMXXXX ou %TDXXXX - Operando inicial de onde os valores são copiados (operando fonte).

—*_____ %KMXXXX - Posição inicial a ser transferida da tabela fonte. Este parâmetro é desconsiderado se o operando fonte for um operando simples (%M ou %D).

—*— %MXXXX, %DXXXX, %TMXXXX ou %TDXXXX - Operando inicial para onde os valores são copiados (operando destino).

—*_____ %KMXXXX - Posição inicial para onde serão copiados os valores na tabela destino. Este parâmetro é desconsiderado se o operando destino for um operando simples (%M ou %D).

—*_____ %KMXXXX - Número de operandos simples ou posições de tabela a serem transferidos a partir do operando ou da posição inicial declarados nos parâmetros anteriores. Deve ser menor ou igual a 255.

- **OPER4** - Não utilizado.

Entradas e Saídas

Descrição das entradas:

- **habilita** - quando esta entrada está energizada a função é chamada, sendo analisados os parâmetros programados na instrução CHF. Caso os mesmos estejam incorretos, são acionadas as saídas de índice inválido.

Descrição das saídas:

- **sucesso** - indica que a movimentação foi corretamente realizada.
- **índice fonte inválido** - indica que houve erro na especificação do operando fonte:
 - * o operando não está declarado no módulo C
 - * o tipo do parâmetro 2 não é %KM
 - * não existe a posição inicial programada, se o operando fonte for tabela
 - * não existem operandos ou posições de tabela suficientes para realizar a movimentação
- **índice destino inválido** - indica que houve erro na especificação do operando destino:
 - * o operando não está declarado no módulo C
 - * o tipo do parâmetro 4 não é %KM
 - * não existe a posição inicial programada, se o operando destino for tabela
 - * não existem operandos ou posições de tabela suficientes para realizar a movimentação

Caso as duas saídas de índice inválido sejam acionadas simultaneamente, ocorreu algum dos seguintes erros:

- o número de parâmetros programado em OPER1 é diferente de cinco
- o tipo do parâmetro 5 não é %KM
- o número total de posições a serem transferidas é maior que 255

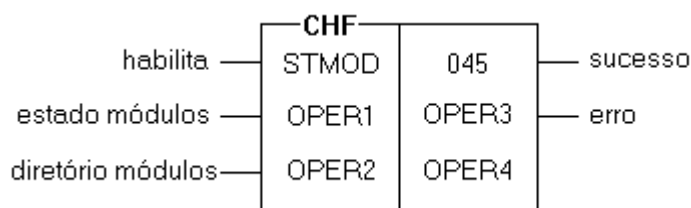
Utilização

Esta função pode ser utilizada nas UCPs AL-600, AL-2000/MSP, AL-2002/MSP, AL-2003, AL-2004, QK800, QK801 e QK2000/MSP.

ATENÇÃO:

Esta função permite a movimentação de um grande número de operandos em uma única varredura. Deve-se utilizá-la com cuidado para que o tempo máximo de ciclo do programa não seja excedido.

F-STMOD.045 - Função Estado dos Barramentos e Módulos de E/S



Introdução

A função F-STMOD.045 possibilita a leitura do estado dos barramentos, dos octetos e dos módulos de E/S do CP. Ela permite que sejam programadas ações e procedimentos especiais no programa aplicativo, para o caso de troca a quente ou erro em algum módulo ou barramento. Também coloca à disposição do programa aplicativo a configuração de módulos de E/S utilizada pelo CP.

Programação

Operandos

As células da instrução CHF utilizada para a chamada da função são programadas do seguinte modo:

- **OPER1** - Especifica o número de parâmetros que são passados para a função em OPER3. Este operando deverá ser obrigatoriamente uma constante memória com valor 4 (%KM+00004).
- **OPER2** - Deve ser um operando do tipo constante memória com valor 0 (%KM+00000). Determina o número de parâmetros possíveis de serem programados na janela de edição de OPER4. Como esta função não necessita de nenhum parâmetro em OPER4, o valor de OPER2 é 0.
- **OPER3** - Contém os parâmetros que são passados para a função, declarados através de uma janela visualizada no MasterTool quando a instrução CHF for editada. O número de parâmetros editáveis é especificado em OPER1, sendo fixo em 4 para este módulo:

—* **%MXXXX** ou **%TMXXXX** - Operando memória ou tabela que recebe os valores do estado dos octetos de E/S. Se for utilizado um operando %M, devem estar definidos no mínimo 16 operandos a partir do declarado (incluindo o mesmo) para que a função seja executada corretamente. Caso seja um operando tabela, esta deve possuir pelo menos 16 posições. Cada ponto do operando %M ou da posição de tabela representa o estado de um operando %E ou %S (apenas os pontos do seu byte menos significativo, .0 a .7). O valor 0 no ponto indica que o octeto está sendo atualizado normalmente pela varredura de E/S, enquanto que o valor 1 indica que o módulo ou barramento correspondente ao octeto está com erro ou desativado para troca a quente. A tabela a seguir mostra os pontos dos operandos correspondentes ao estado dos octetos de E/S.

ATENÇÃO: Para a utilização com o CP AL-2003 ou AL-2004 devem estar definidos no mínimo 32 operandos memória, ou 32 posições de tabela, como primeiro parâmetro, para visualização dos estados dos 256 octetos de E/S.

Operando	Posição Tabela	Octeto de E/S Associado ao Ponto do Operando ou da Posição da Tabela							
		.7	.6	.5	.4	.3	.2	.1	.0
%MXXXX	0	%E0007	%E0006	%E0005	%E0004	%E0003	%E0002	%E0001	%E0000
%MXXXX+1	1	%E0015	%E0014	%E0013	%E0012	%E0011	%E0010	%E0009	%E0008
%MXXXX+2	2	%E0023	%E0022	%E0021	%E0020	%E0019	%E0018	%E0017	%E0016
%MXXXX+3	3	%E0031	%E0030	%E0029	%E0028	%E0027	%E0026	%E0025	%E0024
%MXXXX+4	4	%E0039	%E0038	%E0037	%E0036	%E0035	%E0034	%E0033	%E0032
%MXXXX+5	5	%E0047	%E0046	%E0045	%E0044	%E0043	%E0042	%E0041	%E0040
%MXXXX+6	6	%E0055	%E0054	%E0053	%E0052	%E0051	%E0050	%E0049	%E0048
%MXXXX+7	7	%E0063	%E0062	%E0061	%E0060	%E0059	%E0058	%E0057	%E0056

Tabela 4-12 Formato de Armazenamento do Estado das E/S para o AL2002/MSP

Operando	Posição Tabela	Octeto de E/S Associado ao Ponto do Operando ou da Posição da Tabela							
		.7	.6	.5	.4	.3	.2	.1	.0
%MXXXX	0	%E0007	%E0006	%E0005	%E0004	%E0003	%E0002	%E0001	%E0000
%MXXXX+1	1	%E0015	%E0014	%E0013	%E0012	%E0011	%E0010	%E0009	%E0008
%MXXXX+2	2	%E0023	%E0022	%E0021	%E0020	%E0019	%E0018	%E0017	%E0016
%MXXXX+3	3	%E0031	%E0030	%E0029	%E0028	%E0027	%E0026	%E0025	%E0024
%MXXXX+4	4	%E0039	%E0038	%E0037	%E0036	%E0035	%E0034	%E0033	%E0032
%MXXXX+5	5	%E0047	%E0046	%E0045	%E0044	%E0043	%E0042	%E0041	%E0040
%MXXXX+6	6	%E0055	%E0054	%E0053	%E0052	%E0051	%E0050	%E0049	%E0048
%MXXXX+7	7	%E0063	%E0062	%E0061	%E0060	%E0059	%E0058	%E0057	%E0056
%MXXXX+8	8	%E0071	%E0070	%E0069	%E0068	%E0067	%E0066	%E0065	%E0064
%MXXXX+9	9	%E0079	%E0078	%E0077	%E0076	%E0075	%E0074	%E0073	%E0072
%MXXX+10	10	%E0087	%E0086	%E0085	%E0084	%E0083	%E0082	%E0081	%E0080
%MXXX+11	11	%E0095	%E0094	%E0093	%E0092	%E0091	%E0090	%E0089	%E0088
%MXXX+12	12	%E0103	%E0102	%E0101	%E0100	%E0099	%E0098	%E0097	%E0096
%MXXX+13	13	%E0111	%E0110	%E0109	%E0108	%E0107	%E0106	%E0105	%E0104
%MXXX+14	14	%E0119	%E0118	%E0117	%E0116	%E0115	%E0114	%E0113	%E0112
%MXXX+15	15	%E0127	%E0126	%E0125	%E0124	%E0123	%E0122	%E0121	%E0120
%MXXX+16	16	%E0135	%E0134	%E0133	%E0132	%E0131	%E0130	%E0129	%E0128
%MXXX+17	17	%E0143	%E0142	%E0141	%E0140	%E0139	%E0138	%E0137	%E0136
%MXXX+18	18	%E0151	%E0150	%E0149	%E0148	%E0147	%E0146	%E0145	%E0144
%MXXX+19	19	%E0159	%E0158	%E0157	%E0156	%E0155	%E0154	%E0153	%E0152
%MXXX+20	20	%E0167	%E0166	%E0165	%E0164	%E0163	%E0162	%E0161	%E0160
%MXXX+21	21	%E0175	%E0174	%E0173	%E0172	%E0171	%E0170	%E0169	%E0168
%MXXX+22	22	%E0183	%E0182	%E0181	%E0180	%E0179	%E0178	%E0177	%E0176
%MXXX+23	23	%E0191	%E0190	%E0189	%E0188	%E0187	%E0186	%E0185	%E0184
%MXXX+24	24	%E0199	%E0198	%E0197	%E0196	%E0195	%E0194	%E0193	%E0192
%MXXX+25	25	%E0207	%E0206	%E0205	%E0204	%E0203	%E0202	%E0201	%E0200
%MXXX+26	26	%E0215	%E0214	%E0213	%E0212	%E0211	%E0210	%E0209	%E0208
%MXXX+27	27	%E0223	%E0222	%E0221	%E0220	%E0219	%E0218	%E0217	%E0216
%MXXX+28	28	%E0231	%E0230	%E0229	%E0228	%E0227	%E0226	%E0225	%E0224
%MXXX+29	29	%E0239	%E0238	%E0237	%E0236	%E0235	%E0234	%E0233	%E0232
%MXXX+30	30	%E0247	%E0246	%E0245	%E0244	%E0243	%E0242	%E0241	%E0240
%MXXX+31	31	%E0255	%E0254	%E0253	%E0252	%E0251	%E0250	%E0249	%E0248

Tabela 4-13 Formato de Armazenamento do Estado das E/S para o AL2003 e AL-2004

Obs.:

- Os octetos foram todos representados como entradas (%E). A partir de determinado endereço devem ser considerados saídas (%S), dependendo da configuração de módulos utilizada nos barramentos.

- No CP AL-2002, os últimos 8 operandos %M ou as últimas 8 posições da tabela são reserva para uso futuro, devendo serem declarados para a correta execução da função.

- No caso do CP ser um AL-2003 ou AL-2004, todos os operandos ou 32 posições de tabela são utilizados para armazenamento do estado das E/S.

- Valor do ponto:

- 0 - octeto sendo atualizado de forma normal com o módulo de E/S correspondente ou não utilizado na configuração do barramento
- 1 - octeto não atualizado, barramento ou módulo com erro ou desativado para troca a quente

—* **%MXXXX** ou **%TMXXXX** - Operando memória ou tabela que recebe os valores do estado do barramento. Se for utilizado um operando %M, devem estar definidos no mínimo 10 operandos a partir do declarado (incluindo o mesmo) para que a função seja executada corretamente. Caso seja um operando tabela, esta deve possuir pelo menos 10 posições. Cada operando ou posição de tabela corresponde ao estado de um barramento.

Operando	Posição Tabela	Barramento Associado
%MXXXX	0	Estado do barramento 0
%MXXXX + 1	1	Estado do barramento 1
%MXXXX + 2	2	Estado do barramento 2
%MXXXX + 3	3	Estado do barramento 3
%MXXXX + 4	4	Estado do barramento 4
%MXXXX + 5	5	Estado do barramento 5
%MXXXX + 6	6	Estado do barramento 6
%MXXXX + 7	7	Estado do barramento 7
%MXXXX + 8	8	Estado do barramento 8
%MXXXX + 9	9	Estado do barramento 9

Tabela 4-14 Formato de Armazenamento do Estado dos Barramentos

Obs.:

- Bits de estado:
 - %MXXXX = 0 - barramento não utilizado
 - %MXXXX.4 (bit 4) - barramento com erro
 - %MXXXX.6 (bit 6) - barramento desativado para troca a quente de módulo
 - %MXXXX.7 (bit 7) - barramento funcionando normalmente

—* **%MXXXX** ou **%TMXXXX** - Operando memória ou tabela que recebe os valores do estado dos módulos. Se for utilizado um operando %M, devem estar definidos no mínimo 160 operandos a partir do declarado (incluindo o mesmo) para que a função seja executada corretamente. Caso seja um operando tabela, esta deve possuir pelo menos 160 posições. Cada operando %M ou posição de tabela corresponde a um módulo do barramento, com o seu estado representado pelos seguintes valores:

- %MXXXX = 0 - módulo inexistente
- %MXXXX.4 (bit 4) - módulo com erro
- %MXXXX.6 (bit 6) - módulo desativado para troca a quente
- %MXXXX.7 (bit 7) - módulo funcionando normalmente

—* **%MXXXX** ou **%TMXXXX** - Operando memória ou tabela que recebe os valores do diretório dos módulos. Se for utilizado um operando %M, devem estar definidos no mínimo 160 operandos a partir do declarado (incluindo o mesmo) para que a função seja executada corretamente. Caso seja um operando tabela, esta deve possuir pelo menos 160 posições. Cada operando %M ou posição de tabela corresponde a um módulo do barramento, contendo o seu código respectivo de identificação. Este código pode ser obtido na janela de edição dos parâmetros dos módulos de E/S no MasterTool (opções *Opções, Módulos de E/S*), no campo Tipo.

- **OPER4** - Não utilizado.

Entradas e Saídas

Descrição das entradas:

- habilita - quando esta entrada está energizada a função é chamada, sendo analisados os parâmetros programados na instrução CHF. Caso os mesmos estejam incorretos, a saída sucesso é desenergizada. Se estiverem corretos, a função realiza a leitura do estado dos octetos e do estado do barramento.
- estado módulos - quando esta entrada é energizada, a função realiza a leitura do estado dos módulos para o operando declarado em OPER3.
- diretório módulos - quando esta entrada é energizada, a função realiza a leitura do diretório dos módulos para o operando declarado em OPER3.

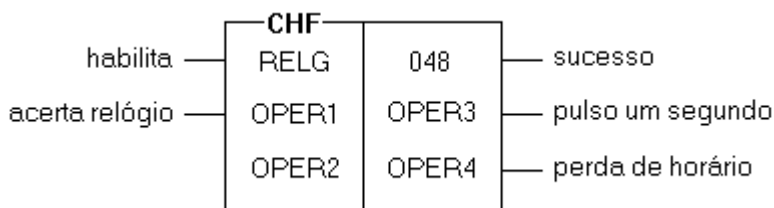
Descrição das saídas:

- sucesso - esta saída é energizada quando a função foi corretamente executada.
- erro - é energizada caso ocorra erro na especificação dos operandos ou tentativa de acesso a operandos não declarados.

Utilização

Esta função pode ser utilizada somente nas UCPs AL-2002/MSP, AL-2003 e AL-2004.

F-RELG.048 - Função para Acesso ao Relógio de Tempo Real



Introdução

A função F-RELG.048 realiza o acesso ao relógio de tempo real contido na UCP AL-2002. O relógio possui horário e calendário completos, permitindo o desenvolvimento de programas aplicativos que dependam de bases de tempo precisas. A informação de tempo é mantida mesmo com a falta de alimentação do sistema, pois a UCP é alimentada por baterias.

Esta função possui características semelhantes à função F-SINC.049, pois ambas executam acessos ao mesmo relógio, diferindo apenas quanto aos métodos de acerto. Elas podem ser utilizadas simultaneamente em um mesmo programa, caso necessário.

Programação

Operandos

As células da instrução CHF utilizada para a chamada da função são programadas do seguinte modo:

- **OPER1** - Especifica o número de parâmetros que são passados para a função em OPER3. Este operando deverá ser obrigatoriamente uma constante memória com valor 2 (%KM+00002).
- **OPER2** - Deve ser um operando do tipo constante memória com valor 0 (%KM+00000). Determina o número de parâmetros possíveis de serem programados na janela de edição de OPER4. Como esta função não necessita de nenhum parâmetro em OPER4, o valor de OPER2 é 0.
- **OPER3** - Contém os parâmetros que são passados para a função, declarados através de uma janela visualizada no MasterTool quando a instrução CHF for editada. O número de parâmetros editáveis é especificado em OPER1, sendo fixo em 2 para este módulo:

—* %MXXXX ou %TMXXXX - Especificação dos operandos para onde são lidos os valores do relógio. Se este parâmetro for especificado como memória, os valores são lidos para a memória declarada e as 6 subseqüentes. Se for especificado como tabela, os valores são colocados a partir da posição 0 até 6. Caso os operandos não estejam declarados, a leitura dos valores de tempo não é realizada e as saídas da instrução são desligadas. É possível o uso de tabelas com mais de 7 posições, sendo que a função ignora as posições excedentes. Os valores são lidos dos operandos na seguinte seqüência:

Operando	Posição Tabela	Conteúdo	Formato
%MXXXX	0	Segundos	000XX
%MXXXX + 1	1	Minutos	000XX
%MXXXX + 2	2	Horas	000XX
%MXXXX + 3	3	Dia do mês	000XX
%MXXXX + 4	4	Mês	000XX
%MXXXX + 5	5	Ano	000XX
%MXXXX + 6	6	Dia da semana	000XX

Tabela 4-15 Valores Lidos do Relógio (F-REL.G.048)

O conteúdo destes operandos pode ser lido a qualquer momento, mas são atualizados com a hora real do relógio apenas quando a instrução for executada. Não devem ser modificadas em nenhum outro ponto do programa aplicativo. É utilizado o formato 24 horas na contagem do tempo. Os dias da semana são contados com valores de 1 a 7:

Valor	Dia da Semana
1	Domingo
2	Segunda-feira
3	Terça-feira
4	Quarta-feira
5	Quinta-feira
6	Sexta-feira
7	Sábado

Tabela 4-16 Valores dos Dias da Semana (F-REL.G.048)

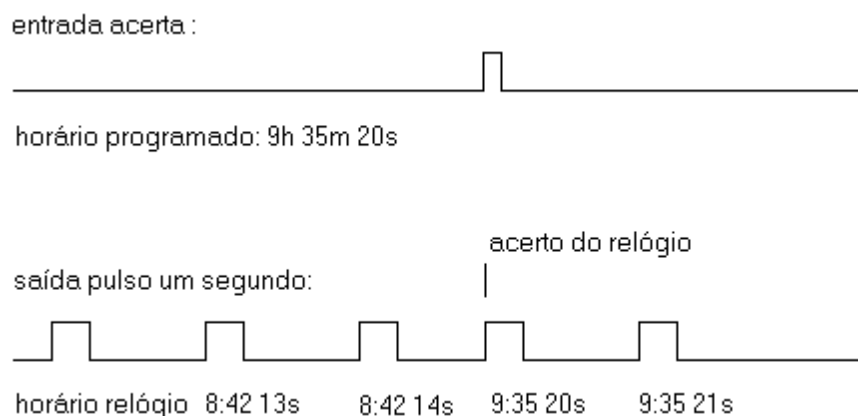
—* %MXXXX ou %TMXXXX - Especificação dos operandos de onde são acertados os valores do relógio, com o acionamento de alguma das entradas de acerto da função. Se este parâmetro for especificado como memória, os valores são copiados da memória declarada e as 6 subseqüentes. Se for especificado como tabela, os valores são copiados da posição 0 até 6. Caso os operandos não estejam declarados, o acerto não é realizado e as saídas da instrução são desligadas. Os valores a serem copiados para o relógio devem ser colocados nos operandos na mesma seqüência dos operandos de leitura (segundos, minutos, horas, dia do mês, mês, ano e dia da semana).

- **OPER4** - Não utilizado.

Entradas e Saídas

Descrição das entradas:

- **habilita** - quando esta entrada está energizada a função é chamada, sendo analisados os parâmetros programados na instrução CHF. Caso os mesmos estejam incorretos, todas as saídas da instrução são desenergizadas. Se estiverem corretos, os valores de tempo do relógio são transferidos para os operandos memória ou para a tabela declarada como primeiro parâmetro em OPER3, a saída sucesso é energizada e a saída pulso um segundo é ligada por uma varredura a cada segundo.
- **acerta relógio** - quando energizada, os valores dos operandos declarados como segundo parâmetro em OPER3 são acertados no relógio, caso estejam com valores corretos. Enquanto esta entrada estiver acionada o tempo não é contado, permanecendo a saída pulso um segundo desenergizada.

Exemplo:**Figura 4-7 Exemplo de Diagrama de Tempos da Entrada Acerta**

Descrição das saídas:

- **sucesso** - é energizada quando a função foi corretamente executada.
- **pulso um segundo** - indica se houve uma mudança no contador de segundos do relógio. O pulso dura uma varredura e pode ser usado para sincronizar eventos do programa aplicativo.
- **perda de horário** - esta saída é ligada caso o relógio tenha ficado sem a alimentação da bateria durante falha na alimentação principal. É desacionada com o acerto do relógio.

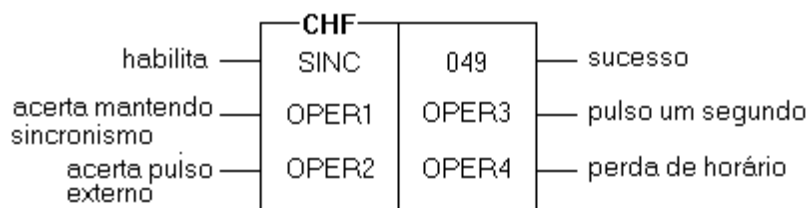
Utilização

Esta função pode ser utilizada somente nas UCPs AL-2002/MSP, AL-2003, AL-2004, PL104 e PL105.

ATENÇÃO:

A alimentação de bateria é fornecida para a UCP através do barramento. Portanto, a remoção da UCP AL-2002/MSP do bastidor causa a perda do horário do relógio.

F-SINC.049 - Função para Acesso ao Relógio de Tempo Real Sincronizado



Introdução

A função F-SINC.049 realiza o acesso ao relógio de tempo real contido na UCP AL-2002/MSP. O relógio possui horário e calendário completos, permitindo o desenvolvimento de programas aplicativos que dependam de bases de tempo precisas. A informação de tempo é mantida mesmo com a falta de alimentação do sistema, pois a UCP é alimentada por baterias.

Os relógios de vários controladores AL-2002 podem operar de forma sincronizada com precisão de 1 milissegundo, através das redes de sincronismo e ALNET II. A função F-SINC.049 possui características úteis para o uso em controladores que operam com seus relógios sincronizados, especialmente no CP gerador do sincronismo.

Esta função possui características semelhantes à função F-RELG.048, pois ambas executam acessos ao mesmo relógio, diferindo apenas quanto aos métodos de acerto. Elas podem ser utilizadas simultaneamente em um mesmo programa, caso necessário.

Para maiores informações a respeito da sincronização dos controladores, consultar o Manual de Utilização AL-2002 e o Manual de Redes ALTUS.

Programação

Operandos

As células da instrução CHF utilizada para a chamada da função são programadas do seguinte modo:

- **OPER1** - Especifica o número de parâmetros que são passados para a função em OPER3. Este operando deverá ser obrigatoriamente uma constante memória com valor 2 (%KM+00002).
- **OPER2** - Deve ser um operando do tipo constante memória com valor 0 (%KM+00000). Determina o número de parâmetros possíveis de serem programados na janela de edição de OPER4. Como esta função não necessita de nenhum parâmetro em OPER4, o valor de OPER2 é 0.
- **OPER3** - Contém os parâmetros que são passados para a função, declarados através de uma janela visualizada no MasterTool quando a instrução CHF for editada. O número de parâmetros editáveis é especificado em OPER1, sendo fixo em 2 para este módulo:

—* %MXXXX ou %TMXXXX - Especificação dos operandos para onde são lidos os valores do relógio. Se este parâmetro for especificado como memória, os valores são lidos para a memória declarada e as 6 subseqüentes. Se for especificado como tabela, os valores são colocados a partir da posição 0 até 6. Caso os operandos não estejam declarados, a leitura dos valores de tempo não é realizada e as saídas da instrução são desligadas. É possível o uso de tabelas com mais de 7 posições, sendo que a função ignora as posições excedentes. Os valores são lidos dos operandos na seguinte sequência:

Operando	Posição Tabela	Conteúdo	Formato
%MXXXX	0	Segundos	000XX
%MXXXX + 1	1	Minutos	000XX
%MXXXX + 2	2	Horas	000XX
%MXXXX + 3	3	Dia do mês	000XX
%MXXXX + 4	4	Mês	000XX
%MXXXX + 5	5	Ano	000XX
%MXXXX + 6	6	Dia da semana	000XX

Tabela 4-17 Valores Lidos do Relógio (F-SINC.049)

O conteúdo destes operandos pode ser lido a qualquer momento, mas são atualizados com a hora real do relógio apenas quando a instrução for executada. É utilizado o formato 24 horas na contagem do tempo. Os dias da semana são contados com valores de 1 a 7:

Valor	Dia da Semana
1	Domingo
2	Segunda-feira
3	Terça-feira
4	Quarta-feira
5	Quinta-feira
6	Sexta-feira
7	Sábado

Tabela 4-18 Valores dos Dias da Semana (F-SINC.049)

—* %MXXXX ou %TMXXXX - Especificação dos operandos de onde são acertados os valores do relógio, com o acionamento de alguma das entradas de acerto da função. Se este parâmetro for especificado como memória, os valores são copiados da memória declarada e as 6 subseqüentes. Se for especificado como tabela, os valores são copiados da posição 0 até 6. Caso os operandos não estejam declarados, o acerto não é realizado e as saídas da instrução são desligadas. Os valores a serem copiados para o relógio devem ser colocados nos operandos na mesma seqüência dos operandos de leitura (segundos, minutos, horas, dia do mês, mês, ano e dia da semana).

- **OPER4** - Não utilizado.

Entradas e Saídas

Descrição das entradas:

- **habilita** - quando esta entrada está energizada a função é chamada, sendo analisados os parâmetros programados na instrução CHF. Caso os mesmos estejam incorretos, todas as saídas da instrução são desenergizadas. Se estiverem corretos, os valores de tempo do relógio são transferidos para os operandos memória ou para a tabela declarada como primeiro parâmetro em OPER3, a saída sucesso é energizada e a saída pulso um segundo é ligada por uma varredura a cada segundo.
- **acerta mantendo sincronismo** - quando energizada, os valores dos operandos declarados como segundo parâmetro em OPER3 são acertados no relógio, caso estejam corretos, sendo mantida a seqüência de incremento dos segundos. Desta forma, a seqüência dos pulsos de segundo não é alterada, somente o valor

do tempo contido no relógio. Esta característica é útil para o acerto do CP gerador do sincronismo, pois não modifica a base de tempo dos demais controladores do sistema, apenas os valores do horário.

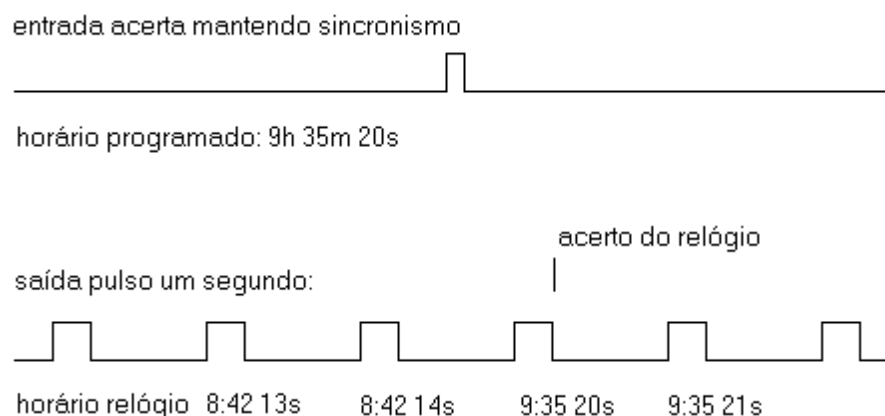


Figura 4-8 Ex. de Diagrama de Tempos da Ent. Acerta Mantendo Sincronismo

- **acerto pulso externo** - quando energizada, os valores dos operandos declarados como segundo parâmetro em OPER3 são programados para serem acertados no relógio no próximo pulso na entrada de sincronismo do CP, se o CP estiver configurado como gerador do sincronismo. Quando ocorrer o pulso na entrada de sincronismo, os novos valores do horário são assumidos, caso estejam com valores corretos. Além do acerto do horário, a contagem dos segundos é inicializada no instante do acionamento do pulso externo, modificando a sequência dos pulsos de segundo gerada. Desta forma é possível acertar o controlador que gera o sincronismo do sistema de controle por relógios externos ou outros sistemas. Não é necessário que a entrada seja mantida energizada até o acionamento do pulso para que este acerto ocorra.

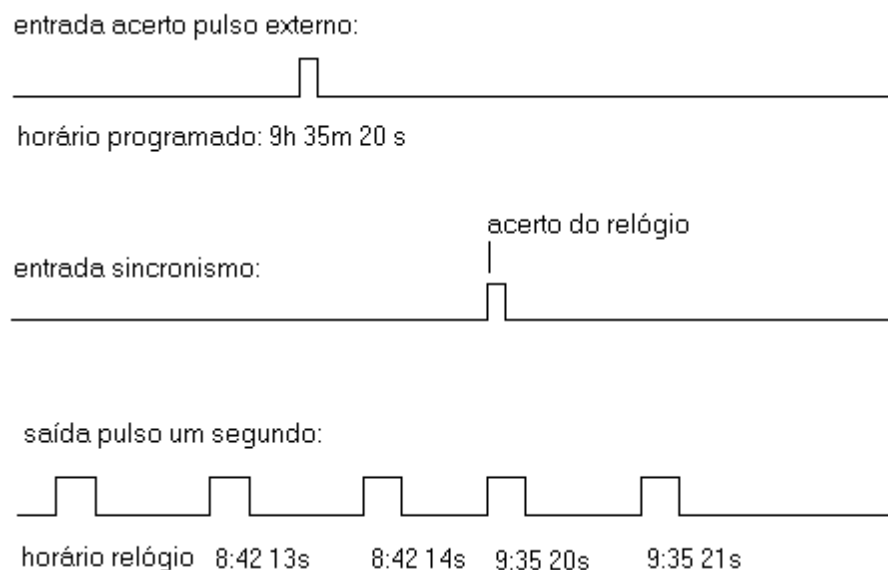


Figura 4-9 Ex. de Diagrama de Tempos da Ent. Acerto Pulso Externo

ATENÇÃO:

Este acerto somente é possível para CPs configurados como geradores do sincronismo. O pulso externo deve ocorrer em até 6 segundos após a entrada ter sido acionada. Depois deste período, os valores do horário não serão modificados com o pulso externo, inicializando-se somente a contagem dos segundos.

Caso o CP esteja configurado como gerador do sincronismo, o acionamento simultâneo das entradas acerta mantendo sincronismo e acerto pulso externo faz com que o CP gere um acerto dos valores do horário em todos os demais CPs ligados à rede de sincronismo (acerto absoluto), não modificando o horário do seu relógio. Para que este acerto ocorra, além dos CPs estarem conectados à rede de sincronismo, devem estar interligados pela rede ALNET II. Este acerto é idêntico ao realizado automaticamente a cada minuto pelo CP gerador do sincronismo.

Descrição das saídas:

- **sucesso** - é energizada quando a função foi corretamente executada.
- **pulso um segundo** - indica se houve uma mudança no contador de segundos do relógio. O pulso dura uma varredura e pode ser usado para sincronizar eventos do programa aplicativo.
- **perda de horário** - esta saída é ligada caso o relógio tenha ficado sem a alimentação da bateria durante falha na alimentação principal. É desacionada com o acerto do relógio.

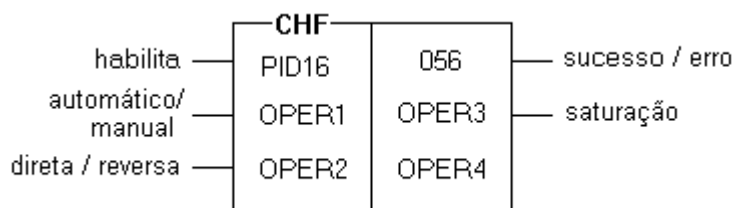
Utilização

Esta função pode ser utilizada somente nas UCPs AL-2002/MSP, AL-2003 e AL-2004.

ATENÇÃO:

A alimentação de bateria é fornecida para a UCP através do barramento. Portanto, a remoção da UCP AL-2002/MSP do bastidor causa a perda do horário do relógio.

F-PID16.056 - Módulo F para controle PID



Introdução

A função F-PID16.056, disponível para as UCPs AL-2003, AL-2004, PO3145, PO3142 e PO3242, implementa o algoritmo de controle proporcional, integral e derivativo. A partir de um valor medido (VM) e do ponto de ajuste desejado (PA) a função calcula o valor de atuação (VA) para o processo controlado. Este valor é calculado periodicamente, levando em consideração os fatores proporcional, integral e derivativo programados. Trata-se de um algoritmo de controle PID tipo ISA onde o ganho proporcional é o ganho do controlador, aplicado tanto ao erro como às parcelas integral e derivativa do controlador.

O módulo F pode ser representado pelo diagrama de blocos mostrado na figura 4-10.

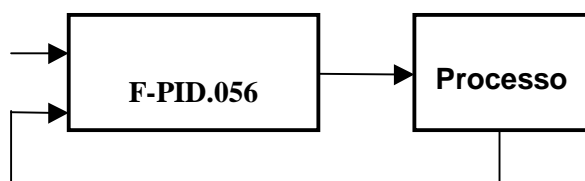


Figura 4-10 Diagrama de blocos do F-PID16.056

Detalhes do diagrama de blocos do controlador PID podem ser observados na figura 4-11.

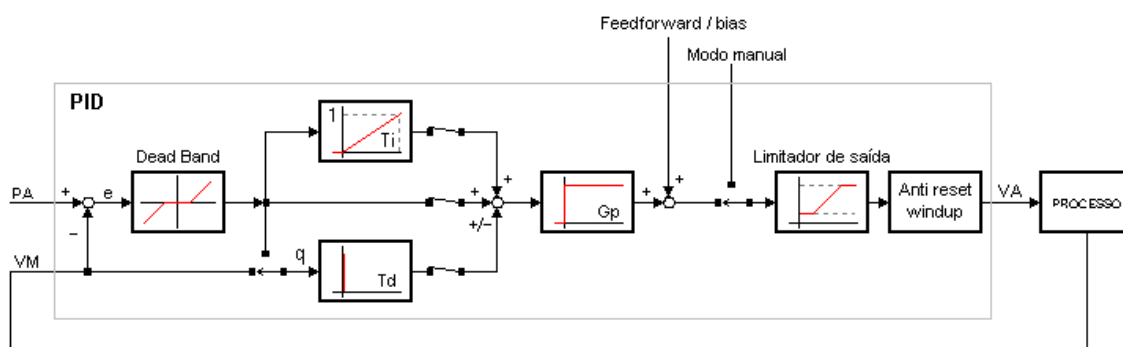


Figura 4-11 Diagrama de blocos detalhado do F-PID16.056

O algoritmo de controle básico utilizado para o controlador PID é descrito pela equação abaixo.

$$VA = Gp \cdot (e + \frac{1}{Ti} \cdot \int e \cdot dt \pm Td \cdot \frac{\partial q}{\partial t}) + Bias$$

Onde:

- VA é a valor de atuação;
- Gp é o ganho proporcional do controlador;
- e é o erro do sistema ($PA-VM$);
- Ti é a constante de tempo integral (s/rep);
- Td é a constante de tempo derivativo em (s);
- dt é o período de amostragem;
- q representa o erro do sistema (+) ou a variável medida (-), conforme seleção;
- PA é o ponto de ajuste;
- VM é a variável medida no processo que está sendo controlado;
- $Bias$ é um deslocamento inserido através de um ponto de soma após o cálculo do algoritmo.

Além do algoritmo de controle básico descrito anteriormente, o módulo F-PID16.056 possui as seguintes características:

- Operação em 4 quadrantes (valores positivos e negativos em entradas e saídas);
- Pode ser usado em modo cascata, implementando algoritmos de controle complexos;
- Operação em 16 bits;
- Uso de parâmetros (Gp , Ti , Td) diretamente no formato ISA;
- Seleção de termo derivativo agindo em função do erro (positivo) ou da variável medida (negativo);
- Inibição individual dos termos derivativo, integral ou proporcional;
- Ação derivativa calculada sobre três amostragens (filtro);
- Ação direta ou reversa;
- Limites de saída ajustáveis;
- Desaturação da ação integral (“anti reset windup”);
- Limitação da taxa de crescimento;
- “Feedforward / bias”;
- Modo manual ou automático;
- “Output tracking” para transição suave (bumpless) do modo manual para o modo automático;
- Zona morta configurável aplicada sobre o erro.

Programação

Operandos

As células da instrução CHF (figura 3) utilizadas para a chamada da função são programadas do seguinte modo:

- OPER1 - Especifica o número de parâmetros que são passados para a função em OPER3. Este operando deverá ser obrigatoriamente uma constante memória com valor 6 (%KM+00006).
- OPER2 - Especifica o número de parâmetros que são passados para a função em OPER4. Este operando deverá ser obrigatoriamente uma constante memória com valor 0 (%KM+00000).
- OPER3 - Contém os parâmetros que são passados para a função, declarados quando a instrução CHF for editada. O número de parâmetros editáveis é especificado em OPER1, sendo fixo em 6 para este módulo:
 - %TMXXXX - Tabela que contém os parâmetros utilizados pelo algoritmo de controle. Deve conter pelo menos 30 posições.
 - %MXXXX - Memória que contém o valor medido do processo (VM), normalmente obtido através de uma entrada analógica.
 - %MXXXX - Contém o ponto de ajuste (PA), que é o valor desejado para a variável medida ("set point").
 - %MXXXX - Memória que contém o valor de atuação (VA) gerado pelo algoritmo de controle. Em modo manual o algoritmo de controle não atua sobre esta variável, que pode ser manipulada pelo usuário.
 - %MXXXX - Memória para "feedforward/bias". O valor deste operando será somado ao valor de saída do controlador PID antes da limitação (limites superior e inferior declarados na tabela de parâmetros).
 - %AXXXX - Octeto auxiliar que contém pontos de controle da função PID.
- OPER4 - Não utilizado.

Entradas e Saídas

Descrição das entradas:

- habilita - quando esta entrada está energizada a função é chamada, sendo analisados os parâmetros programados na instrução CHF. Caso o número de parâmetros ou seu tipo sejam diferentes das necessidades da função, haverá a desenergização da saída sucesso/erro. Se estiverem corretos, o cálculo do controle PID é realizado.
- automático(0)/manual(1) - quando energizada, a variável de atuação não é alterada pela função podendo ser alterada manualmente (modo manual).
- reversa (0)/ direta (1) - especifica a forma como o controlador atuará sobre o processo.

ATENÇÃO:

O conceito de modo direto e reverso do módulo F-PID.033 está invertido em relação ao F-PID16.056.

Descrição das saídas:

- sucesso(1)/erro(0)- é energizada quando a função foi corretamente executada. Sempre que ocorrerem erros na especificação dos operandos, tentativa de acesso a operandos não declarados ou parâmetros inválidos, esta saída não é energizada, indicando erro.
- saturação – quando energizada indica que a saída do controlador atingiu a saturação, ou no limite máximo, ou no limite mínimo.

Utilização

Este módulo F está disponível para os CPs AL-2003, AL-2004, PO3145, PO3142 e PO3242.

Características de Funcionamento

Características do Funcionamento

Dessaturação da Ação Integral

A dessaturação da ação integral (“anti-reset windup”) é feita de modo a evitar que o termo integral continue a acumular erro quando um distúrbio no processo causa a saturação da saída do controlador em um dos limites. No momento em que o valor de saída atinge um dos limites (máximo ou mínimo), o termo integral é fixado em seu valor corrente, impedindo o seu crescimento indefinido. Isto assegura que haverá uma resposta do controlador tão logo desapareça o distúrbio que o levou a saturar a saída.

Modo Manual

A função pode ser executada em modo manual, energizando-se a segunda entrada da instrução CHF. Neste modo, a rotina não modifica o valor da saída de atuação, mas o acompanha (“output tracking”). Ou seja, em função do valor da saída no modo manual e do valor medido do processo, os termos proporcional e derivativo são calculados e o termo integral é forçado para um valor adequado de modo que, quando ocorrer a transição de manual para automático, a rotina possa reassumir o controle com o valor inicial da saída igual ao último valor da saída no modo manual. Chama-se este comportamento de comutação manual/automática balanceada (“bumpless transfer”). É importante observar que este recurso somente funciona quando a ação integral está habilitada. Embora em modo manual, a saída de atuação não pode assumir valores maiores que os limites declarados na tabela de parâmetros. Quando isto ocorrer o valor da saída de atuação será forçado para o limite mais próximo.

Controle Direto e Reverso

A forma de controle pode ser direta ou reversa. Esta seleção é realizada desenergizando ou energizando a terceira entrada da instrução CHF.

Em caso de controle direto, caso ocorra um aumento no valor medido (VM), o controlador deve aumentar a saída de atuação (VA) a fim de controlar o processo.

Em caso de controle reverso, caso ocorra um aumento no valor medido (VM), o controlador deve diminuir a saída de atuação (VA) a fim de controlar o processo.

Considerando-se dois exemplos que utilizam a mesma válvula, controlada por uma saída analógica 4-20 mA (VA varia de 0 a 4095). Suponha que com VA = 0 (4 mA) a válvula esteja totalmente fechada, e que com VA = 4095 (20 mA) a válvula esteja totalmente aberta.

No primeiro exemplo, deseja-se controlar o nível de um tanque (VM = nível do tanque), através de uma válvula de esgotamento do tanque. Portanto, quanto mais se abrir a válvula, mais rápido o nível

do tanque diminui. Neste caso, caso aumente o nível (VM) do tanque, o controlador deve aumentar VA para abrir a válvula. Portanto, trata-se de um controle direto.

No segundo exemplo, deseja-se controlar a vazão através da válvula. Portanto, quanto mais se abrir a válvula, maior será a vazão. Neste caso, caso aumente a vazão (VM) através da válvula, o controlador deve diminuir VA para fechar a válvula. Portanto, trata-se de um controle reverso.

Intervalo de amostragem

O intervalo entre amostragens de um laço PID pode variar de 0,01 a 10 segundos. É de responsabilidade do usuário programar um "disparador" da função, ou seja, um trecho de programa aplicativo que somente habilite a rotina F-PID16.056 nos intervalos de tempo desejados. Aconselha-se utilizar um módulo E018, pois este módulo é executado dentro de um intervalo de tempo fixo que pode ser utilizado para gerar uma ou mais bases de tempo para a execução de um ou mais laços PID. Nota-se ainda que o valor do intervalo de amostragem (dt) declarado na tabela de parâmetros do controlador deve coincidir com o intervalo de tempo das chamadas do "disparador".

Tempo de execução

O pior caso de execução de um laço de controle com o F-PID16.056 atinge o tempo de 360 μ s. Este tempo é válido para as UCPs AL-2003, PO3145, PO3142 e PO3242.

Descrição das Posições da Tabela de Parâmetros

A tabela com os parâmetros do controlador também é utilizada para armazenar variáveis de uso interno, o somatório da ação integral e as variáveis medidas ou erros de ciclos anteriores para o cálculo da ação derivativa. Cada posição desta tabela está descrita abaixo, na tabela 1.

Posição da tabela	Parâmetros	Descrição
00	GP (x10)	Ganho proporcional (sem unidade). Os valores possíveis estão dentro do intervalo de 0,1 a 3000. O ganho proporcional deve ser multiplicado por 10 para ser declarado neste campo, assumindo o intervalo de 1 a 30000. Valores fora desta faixa farão com que o módulo F entre em modo erro e não execute o algoritmo de controle.
01	Ti (x10)	Constante de tempo integral (s/repetição). Os valores possíveis estão dentro do intervalo de 0,1 a 3200 s/rep. A constante de tempo integral deve ser multiplicada por 10 para ser declarada neste campo, assumindo o intervalo de 1 a 32000. Observa-se aqui que quanto menor o valor de Ti, maior será a ação integral. Valores fora desta faixa farão com que o módulo F entre em modo erro e não execute o algoritmo de controle.
02	Td (x100)	Constante de tempo derivativa (s). Os valores possíveis estão dentro do intervalo de 0 a 320 s em unidades de 0,01s. A constante de tempo derivativo deve ser multiplicada por 100 para ser declarada neste campo, assumindo o intervalo de 0 a 32000. Valores fora desta faixa farão com que o módulo F entre em modo erro e não execute o algoritmo de controle. Se for atribuído o valor zero à constante de tempo derivativa, a ação será calculada com valor zero, não influenciando na saída de atuação. Recomenda-se desabilitar a ação derivativa quando esta não é utilizada.
03	dt (x100)	Intervalo de amostragem do processo que está sendo controlado (s). Os valores possíveis estão dentro do intervalo de 0,01 a 10 s. O intervalo de amostragem deve ser multiplicado por um fator de 100 para ser declarado neste campo, assumindo o intervalo de 1 a 1000. Valores fora desta faixa farão com que o módulo F entre em modo erro e não execute o algoritmo de controle. É responsabilidade do usuário disparar o módulo F neste intervalo de tempo.
04	Valor máximo de saída	Valor máximo de saída permitido. Pode assumir valores de -30000 a +30000. Deve necessariamente ser maior que o valor mínimo de saída. Valores fora desta faixa farão com que o módulo F entre em modo erro e não

		execute o algoritmo de controle.
05	Valor mínimo de saída	Valor mínimo de saída permitido. Pode assumir valores de -30000 a +30000. Deve necessariamente ser menor que o valor máximo de saída. Valores fora desta faixa farão com que o módulo F entre em modo erro e não execute o algoritmo de controle.
06	Zona morta	Zona morta. Pode assumir valores de 0 a +30000. Sempre que o valor absoluto do erro for menor que o valor definido neste campo, o controlador será executado considerando o erro como zero. Para desabilitar este recurso basta declarar o valor zero para a zona morta. Valores fora desta faixa farão com que o módulo F entre em modo erro e não execute o algoritmo de controle.
07	Variação máxima permitida.	O valor declarado neste campo indica o valor absoluto da variação máxima que a saída do controlador pode ter a cada intervalo de amostragem (dt). Pode assumir valores de 1 a 30000. O valor 1 representa uma variação muito pequena enquanto que o valor 30000 representa uma grande variação a cada ciclo de amostragem. Declarando zero neste campo a variação máxima permitida não é verificada, permitindo-se qualquer variação. Valores fora desta faixa farão com que o módulo F entre em modo erro e não execute o algoritmo de controle.
08	AI acumulada x100 (Hi)	O valor presente neste campo é a ação integral acumulada. Visando obter mais resolução numérica, a ação integral acumulada é armazenada multiplicada por um fator de 100. São utilizados três operandos de 16 bits para guardar a parte alta, baixa e parte fracionária da ação integral. Estes campos devem ser inicializados com zero para evitar que algum valor aleatório fique armazenado.
09	AI acumulada x100 (Lo)	
10	AI acumulada x100 (frac)	
11	Valor de atuação anterior (VA)	Este campo é de uso restrito do módulo F e não deve ter seu conteúdo alterado. O valor deste campo é a variável de atuação do ciclo anterior, utilizada para limitar a variação máxima.
12-13	Reservado	Operando reservados.
14	VM(t-3) ou erro(t-3)	Este campo é de uso restrito do módulo F e não deve ter seu conteúdo alterado. Histórico dos três últimos erros ou variáveis medidas, utilizadas para cálculo do termo derivativo. A ação derivativa pode agir tanto em função do erro como da variável medida, porém jamais deve-se trocar a seleção de erro para variável medida ou vice-versa durante o processo de controle.
15	VM(t-2) ou erro(t-2)	
16	VM(t-1) ou erro(t-1)	
17-27	Uso interno	Estas posições da tabela são utilizadas exclusivamente pela função PID, não devendo ser modificadas pelo programa aplicativo.
28-29	Reservado	Operandos reservados.

Tabela 8: Descrição das posições da tabela de parâmetros do F-PID16.056

Sempre que ocorrer alguma alteração nos parâmetros GP, T_i , T_d ou dt o módulo F-PID16.056 necessita um ciclo de execução para adaptar o controlador aos novos parâmetros, não executando o controle neste ciclo e mantendo a variável de atuação (VA) inalterada.

Descrição do Operando %A de Controle

O operando auxiliar de controle do módulo F é utilizado conforme a tabela 2.

Bit	Descrição
0	inibe(1) / habilita(0) ação integral
1	inibe(1) / habilita(0) ação derivativa
2	inibe(1) / habilita(0) ação proporcional
3	Ação derivativa em função do erro (1) ou da variável do processo (0)
4	Reservado
5	Uso interno
6	Uso interno
7	Uso interno

Tabela 9: Descrição do operando auxiliar de controle

Através do operando auxiliar de controle da função F-PID16.056 é possível desabilitar a ação proporcional, integral e/ou derivativa e também selecionar a ação derivativa agindo em função do erro ou da variável medida no processo. Quando alguma ação de controle (seja ela proporcional, integral ou derivativa) não for utilizada, esta deve ser desabilitada ligando o bit correspondente.

Desabilitando a ação de controle proporcional deixa-se de gerar ação de controle proporcional ao erro, mas o ganho do sistema continua sendo aplicado sobre as ações integral e derivativa. Para um integrador puro, por exemplo, deve-se habilitar somente a ação integral, ajustar a constante de tempo T_i desejada e atribuir à constante GP (ganho proporcional) um ganho unitário.

A ação derivativa agindo em função da variável medida é recomendada para a maioria das aplicações, pois evita grandes variações na saída VA quando o ponto de ajuste PA é modificado. Para aplicações especiais existe a possibilidade de seleção da ação derivativa em função do erro do sistema.

Os bits de uso interno são de uso exclusivo da função F-PID16.056 e não devem ter seu conteúdo alterado.

Notas de Aplicação

Seleção do Tempo de Amostragem

A eficiência do controlador digital está diretamente relacionada com o intervalo de amostragem utilizado. A medida que este intervalo diminui, o resultado do controlador digital aproxima-se do resultado de um controlador analógico. Aconselha-se utilizar um tempo de amostragem da ordem de um décimo da constante de tempo do sistema, ou seja

$$T_A = \frac{T}{10}$$

onde T_A é o tempo de amostragem utilizado e T é a constante de tempo do sistema.

Exemplo: Pode-se obter a constante de tempo de um sistema de primeira ordem a partir do seu gráfico da resposta da variável de atuação (VA) a um degrau no ponto de ajuste PA com o laço de controle aberto (PID desabilitado ou em modo manual), conforme a figura 4.

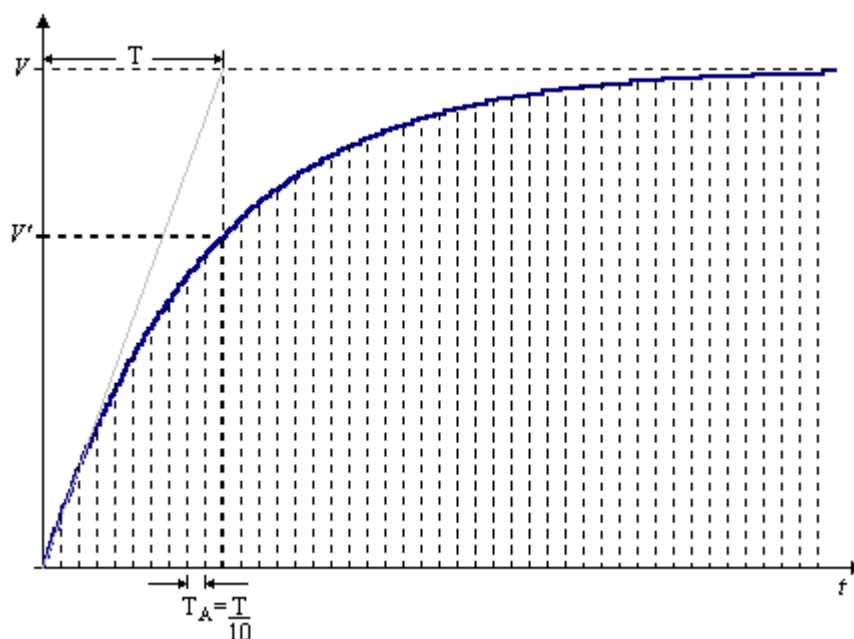


Figura 37: Constante de tempo do sistema e intervalo de amostragem

Esta figura demonstra a obtenção da constante de tempo do sistema por dois modos distintos. O mais usual é tomar como constante de tempo do sistema o tempo necessário para o sistema atingir 63,212% do valor final. Outro modo é traçar a primeira derivada da curva da resposta ao degrau, a constante de tempo é aquela onde esta reta cruza o valor final da resposta do sistema.

Uma vez definida a constante de tempo, basta definir o intervalo de amostragem da ordem de um décimo deste valor.

É importante lembrar que na Série Ponto a atualização das entradas e saídas ocorre na mesma ordem de tempo de um ciclo do CP. Sempre que o tempo de ciclo do CP for maior que o tempo de amostragem aconselha-se o uso da instrução F-AES.087.

Feedforward/Bias

Através do operando memória utilizado para *feedforward/bias* é possível injetar alguma variável do sistema na saída do controlador e/ou aplicar um deslocamento na mesma.

O objetivo do feedforward é medir os principais distúrbios do processo e calcular a mudança necessária na variável de atuação para compensá-los antes que estes causem alterações na variável controlada. A manipulação dos distúrbios do processo pode ser feita através dos blocos de controle avançado (F-CTRL.059) que disponibilizam blocos avanço-atraso, derivador com retardo temporal e retardo temporal de primeira ordem.

Pode-se citar como exemplo, um sistema onde a variável a ser controlada é a temperatura de uma mistura quente. Numa determinada fase do processo é necessário derramar água fria nesta mistura. Sem o feedforward, seria necessário esperar a água fria mudar o estado da mistura para então o controlador gerar a ação corretiva. Utilizando o *feedforward*, um valor associado à temperatura da água fria seria injetado na saída do controlador, fazendo com que este tome uma “ação corretiva” antes mesmo da água fria começar a alterar o estado da mistura quente, agilizando a resposta do controlador.

O “bias” é utilizado sempre que se deseja aplicar algum deslocamento sobre a saída do controlador.

Controle em Cascata

Provavelmente o controle em cascata é uma das técnicas de controle avançado mais utilizadas na prática. É composto por pelo menos duas malhas de controle. A figura 5 mostra um controlador em cascata com duas malhas.

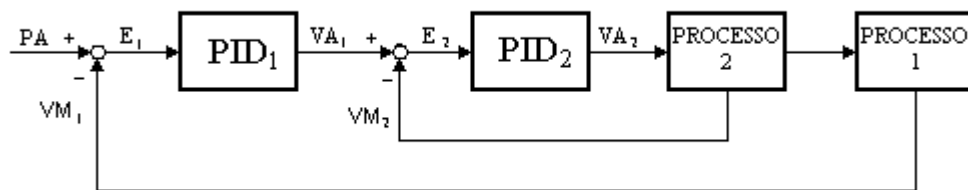


Figura 38: Controlador em cascata com duas malhas

A malha externa é chamada de controlador mestre e a malha interna de controlador escravo. O controlador mestre tem seu ponto de ajuste fixo e sua saída fornece o ponto de ajuste do controlador escravo (VA_1). A variável de atuação do controlador escravo (VA_2) atuará sobre o processo 2 que, por sua vez, atuará sobre o processo 1, fechando a malha do controlador mestre.

Este tipo de controlador é aplicado, por exemplo, no controle de temperatura pela injeção de vapor. Além da variação da temperatura, que deve ser controlada, o sistema está sujeito a variações de pressão na linha de vapor. Torna-se então desejável um controlador de vazão escravo atuando em função das variações de pressão e um controlador mestre para manipular a referência do escravo controlando então a temperatura do processo. Este exemplo pode ser representado graficamente conforme a figura 6.

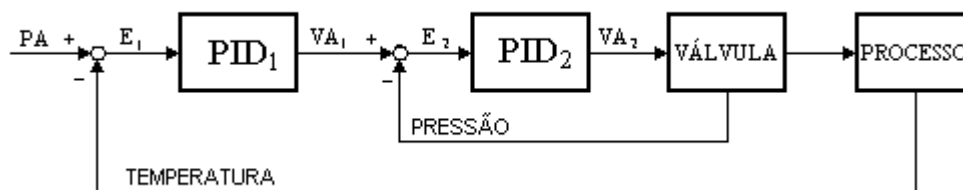


Figura 39: Aplicação de um controlador em cascata

Caso fosse utilizado somente um controlador de temperatura atuando diretamente sobre a válvula de vapor, não haveria como compensar eventuais variações de pressão na linha de vapor.

Existem três principais vantagens no uso de controladores em cascata:

- Qualquer distúrbio que afete o controlador escravo é detectado e compensado por este controlador antes de afetar a variável controlada pelo controlador mestre;
- Aumento da controlabilidade do sistema. No caso do controle de temperatura pela injeção de vapor, a resposta do sistema é melhorada devido ao controlador de vazão aumentando a controlabilidade do laço principal.
- Não linearidades de um laço interno são manipuladas dentro deste laço e não percebidos pelo laço externo. No exemplo anterior, as variações de pressão são compensadas pelo controlador escravo e o controlador mestre “enxerga” apenas uma relação linear entre a válvula e a temperatura.

Considerações Importantes

Para se utilizar controladores em cascata deve-se tomar os seguintes cuidados:

- Como o ponto de ajuste dos controladores escravos é manipulado conforme a saída dos controladores mestres, poderão ocorrer variações bruscas no erro do controlador escravo. Se os controladores escravos estiverem com a ação derivativa agindo em função do erro surgirão ações derivativas com grandes valores. Portanto aconselha-se utilizar os controladores escravos com a ação derivativa em função da variável medida.
- O controlador escravo deve ser rápido o suficiente para eliminar os distúrbios de seu laço antes que estes afetem o laço do controlador mestre.

Sugestões para Ajustes do Controlador PID

A seguir são apresentados dois métodos para a determinação das constantes do controlador PID. O primeiro método consiste na determinação das constantes em função do período de oscilação e do ganho crítico, enquanto que o segundo determina as constantes do controlador em função da constante de tempo (T), do tempo morto (T_m) e do ganho estático do sistema (K). Para maiores detalhes aconselha-se a leitura da literatura referenciada.

ATENÇÃO:

A Altus Sistemas de Informática não se responsabiliza por eventuais danos causados por erros de configuração das constantes do controlador ou parametrização. Recomenda-se que pessoa devidamente qualificada execute esta tarefa.

Determinação das Constantes do Controlador Através do Período e do Ganho Crítico

Este método gera uma resposta amortecida cuja taxa de amortecimento é igual a $1/4$. Isto é, depois de sintonizar um laço através deste método, espera-se um resposta como mostrada na figura 7:

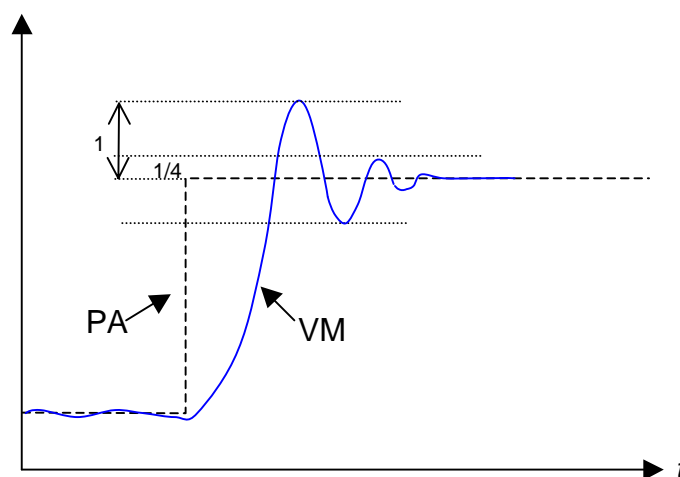


Figura 40:Resposta com Amortecimento de 1/4

O ganho crítico é definido como o ganho de um controlador proporcional que gera uma oscilação de amplitude constante no sistema em malha fechada enquanto que o período crítico é o período desta oscilação. O ganho crítico é uma medida de controlabilidade do sistema, ou seja, quanto maior o ganho crítico mais fácil será o controle do sistema. O período crítico de oscilação é uma medida da velocidade de resposta do sistema em malha fechada, ou seja, quanto maior o período de oscilação mais lento será o sistema. No decorrer deste capítulo o ganho crítico será denominado como G_{Pc} e o período crítico como T_c .

É importante lembrar que ganhos ligeiramente menores que G_{Pc} geram oscilações cujo período decresce com o tempo, enquanto que ganhos maiores que G_{Pc} geram oscilações cuja amplitude cresce com o tempo. No caso de ganhos maiores que G_{Pc} é preciso ter cuidado para não tornar o sistema criticamente instável.

O processo para a determinar G_{Pc} e T_c consiste em fechar a malha com o controlador em modo automático desabilitando a ação integral e a derivativa. Os passos são os seguintes.

- Remover a ação integral e derivativa através do operando %A de controle;

- Aumentar o ganho proporcional com pequenos incrementos. Depois de cada incremento inserir um pequeno distúrbio no sistema através de um pequeno degrau no ponto de ajuste (PA). Verificar o comportamento do sistema (VM), a amplitude de oscilação deve aumentar à medida que o ganho aumenta. O ganho crítico (GP_c) será aquele que gerar oscilações com amplitude constante (ou quase constante) conforme a figura 8;
- Medir o período destas oscilações (T_c).

Para determinar as constantes do controlador basta aplicar os valores de GP_c e T_c nas fórmulas da tabela 3.

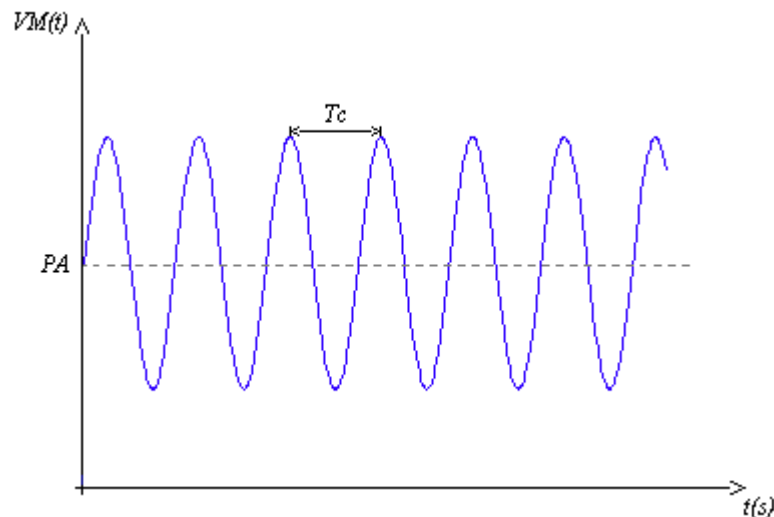


Figura 41: Representação gráfica de um sistema oscilando quando sujeito ao GP_c

Tipo de Controlador	Constantes
Proporcional (P)	$GP = 0,5 \cdot GP_c$
Proporcional e Integral (PI)	$GP = 0,45 \cdot GP_c$ $Ti = \frac{T_c}{1,2}$
Proporcional, Integral e Derivativo (PID)	$GP = 0,75 \cdot GP_c$ $Ti = \frac{T_c}{1,6}$ $Td = \frac{T_c}{10}$

Tabela 10: Fórmulas para determinar as constantes do controlador

Determinação das Constantes do Controlador Através das Constantes do Processo

Este método se aplica bem a processos lineares, de primeira ordem (similar a um circuito RC) e com tempo morto. Na prática, muitos processos industriais se adaptam a este modelo.

O método requer, inicialmente, determinar as seguintes características do processo em laço aberto:

- K : Ganho estático do processo. Definido como a razão entre uma variação de VM e uma variação de VA, ou seja, $K = \Delta VM / \Delta VA$;
- T_m : Tempo morto, definido como o tempo entre o início de uma variação na saída VA (t_0) e o início da reação do sistema.

- T : Constante de tempo do sistema, definido como o tempo que a variável medida leva para excursionar 63,212% de seu valor final.

Além disso, o método requer dois parâmetros adicionais, que não são características do processo em si, e devem ser informados pelo usuário:

T_r : tempo de resposta desejado após a sintonia do laço. Trata-se de uma característica interessante, pois através deste parâmetro o usuário pode informar um requisito de performance do laço controlado.

dt : tempo de amostragem em segundos, isto é, o período de chamada da F-PID16.056 e atualização da entrada VM e saída VA. A constante dt simboliza um tempo morto adicional, que deve ser somado a T_m . Na prática, soma-se $dt/2$ ao valor de T_m , pois este é o tempo morto médio inserido.

O tempo de resposta T_r pode ser comparado com uma “constante de tempo” do laço fechado, conforme ilustra a figura 9.

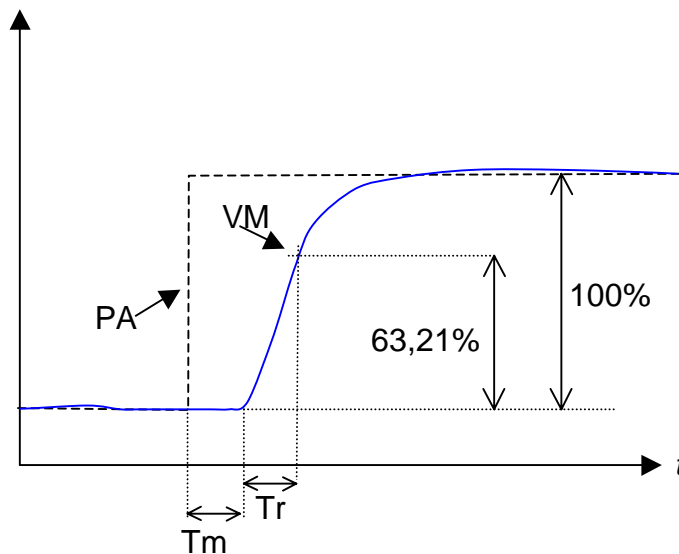


Figura 42: Especificação do Tempo de Resposta T_r

O parâmetro T_r , na figura 9, mostra o tempo de resposta desejado. Trata-se do tempo medido entre o início da resposta do sistema (após o tempo morto T_m), e o momento em que VM atinge 63,21% de sua excursão total. Através de T_r o usuário pode especificar um “requisito de performance” para o laço controlado. Deve-se ter o cuidado de não especificar tempos de resposta menores que um décimo da constante de tempo do sistema, pois do contrário o sistema pode ficar instável. Quanto menor o valor de T_r , maior o ganho necessário.

A seguir, descreve-se como determinar, através de um teste de laço aberto, os demais parâmetros (K , T_m e T), que caracterizam o processo. Um modo simples para determinar estas constantes do processo é colocar o módulo F-PID16.056 em modo manual, gerar um pequeno degrau em VA e plotar a resposta de VM no tempo. Para processos lentos isto pode ser feito manualmente, mas para processos rápidos aconselha-se o uso de um osciloscópio ou qualquer outro dispositivo que monitore a variação de VM. O degrau em VA deve ser grande o suficiente para causar uma variação perceptível em VM.

As figuras 10 e 11 representam respectivamente um degrau na saída VA, aplicado no instante t_0 , e a resposta de um sistema linear de primeira ordem com tempo morto.

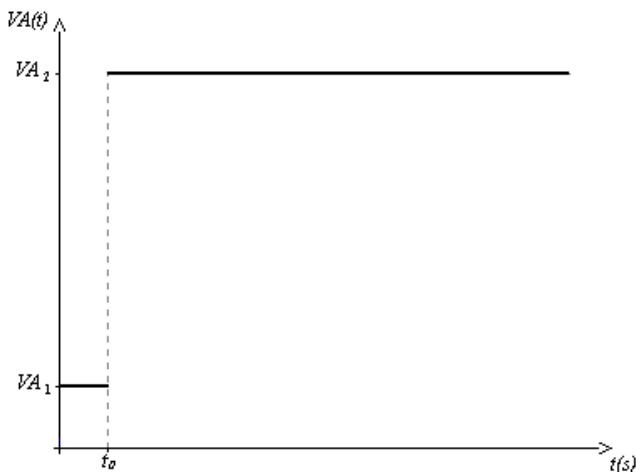


Figura 43: Degrau em VA

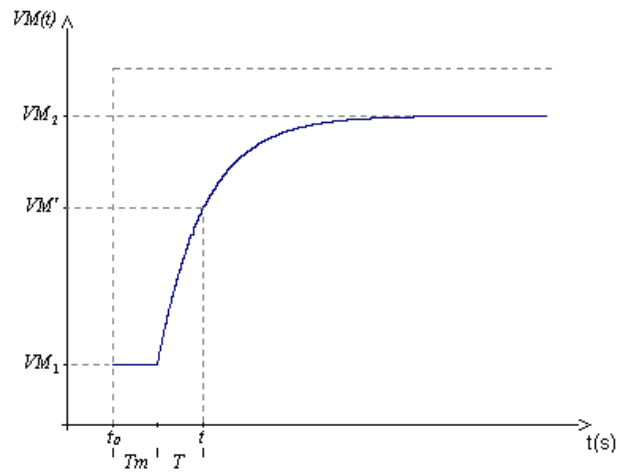


Figura 44: Resposta do sistema ao degrau

Através das figuras 10 e 11 pode-se obter todas as constantes necessárias para a determinação dos parâmetros do controlador. O ganho estático do processo é obtido através da razão entre a variação da variável medida e a variação da variável de atuação, ou seja:

$$K = \frac{VM_2 - VM_1}{VA_2 - VA_1}$$

O tempo morto, Tm, é o tempo entre o momento de aplicação do degrau em VA (t0) e o início da resposta do sistema.

A constante de tempo do sistema, T, é o tempo entre o início da reação do sistema e 63,212% do valor final de VM (VM'), isto é:

$$0,63212 = \frac{VM' - VM_1}{VM_2 - VM_1}$$

A partir das constantes do sistema, K, Tm e T, pode-se obter os parâmetros do controlador utilizando as fórmulas da tabela 4.

Tipo de Controlador	Constantes
Proporcional, Integral e Derivativo (PID)	$GP = \frac{T}{K * (Tr + Tm + dt/2)}$ $Ti = T$ $Td = Tm/2 + dt/4$

Tabela 11: Fórmulas para determinação dos parâmetros do controlador

Ganhos X Escalas

É importante lembrar que o ganho proporcional somente executará sua ação de modo correto quando tanto a entrada como a saída do sistema utilizarem as mesmas escalas. Por exemplo, um controlador proporcional com ganho unitário e entrada (VM) utilizando a faixa de 0 a 1000 somente será realmente unitário se a faixa de saída (VA) também for de 0 a 1000.

Em muitos casos as escalas de entrada e saída são diferentes. Pode-se citar como exemplo um sistema onde o cartão de entrada analógica é de 4-20 mA, onde 4 mA corresponde ao valor 0, e 20 mA corresponde ao valor 30000. E o cartão de saída analógica é de 0V a 10 V, onde 0 V corresponde ao valor 0, e 10V corresponde ao valor 1000. Em casos como o deste exemplo, o ajuste de escalas pode ser feito através do ganho proporcional ao invés de uma normalização dos valores de entrada ou de saída.

Uma estratégia que pode ser adotada é, inicialmente, determinar o ganho em termos percentuais (independente de escalas), sem se preocupar com o tipo de módulos de entrada e saída analógicas utilizados. Posteriormente, após determinado este ganho, deve-se executar a correção de escalas, antes de introduzir o ganho proporcional no módulo F-PID16.056.

A estratégia consiste em determinar o ganho proporcional do sistema utilizando a faixa percentual (0% a 100%) tanto da variável medida (VM) como do valor de atuação (VA), sem levar em consideração os valores absolutos, tanto de VM como de VA.

Isto levará à determinação de um ganho proporcional denominado GP%. Este ganho GP% não pode ser utilizado diretamente na F-PID16.056. Antes é necessário fazer uma correção de escalas, que considere os valores absolutos destas variáveis.

Atenção:

Na seção anterior, *Sugestões para Ajustes do Controlador PID*, são sugeridos métodos de ajuste nos quais a correção de escalas é implícita ao método, não devendo ser considerada. No capítulo seguinte, *Exemplo de Aplicação*, a correção de escalas também é desnecessária, pois utilizou-se um dos métodos abordados na seção *Sugestões para Ajustes do Controlador PID*.

A correção de escalas é ilustrada a partir de um exemplo descrito a seguir.

Considere um sistema de ar condicionado onde o módulo de entrada analógica está lendo um resistor PTC (coeficiente térmico positivo) e o módulo de saída analógica gera uma tensão de 0 a 10V para atuar sobre a válvula responsável pela circulação da água que resfria o ar insuflado.

O módulo de entrada trabalha com uma faixa de 0 a 30000, porém a faixa útil é de 6634 a 8706 com o seguinte significado:

- $EA_0 = 6634 = 0\% = 884,6\Omega$ (corresponde a mínima temperatura que pode ser medida)
- $EA_1 = 8706 = 100\% = 1160,9\Omega$ (corresponde a máxima temperatura que pode ser medida)

O módulo de saída utiliza a mesma faixa de 0 a 30000 sem restrições e com o seguinte significado:

- $SA_0 = 0 = 0\% = 0V$ (corresponde a mínima vazão de água pela válvula)
- $SA_1 = 30000 = 100\% = 10V$ (corresponde a máxima vazão de água pela válvula)

Supondo que o ganho GP% foi previamente determinado, o ganho GP pode ser calculado pela seguinte equação:

$$GP = GP\% * R$$

onde:

$$R = \frac{SA_1 - SA_0}{EA_1 - EA_0}$$

Para o exemplo anterior:

$$R = \frac{30000 - 0}{8706 - 6634} = 14,478$$

Esta razão R é uma constante que, quando multiplicada pelo ganho proporcional do controlador, compensa as diferenças entre as faixas de entrada e saída sem a necessidade de uma normalização direta.

Exemplo de Aplicação

Neste capítulo, será mostrado um exemplo prático de utilização do módulo F-PID16.056, abrangendo diversas fases do projeto do processo e do seu sistema de controle.

Descrição do Processo

O processo exemplo tem como objetivo o fornecimento de água aquecida, com temperatura controlada, para um consumidor. O aquecimento será feito através de um queimador de gás, sendo controlado a partir da variação de vazão de gás através de uma válvula.

A figura 12 ilustra este processo.

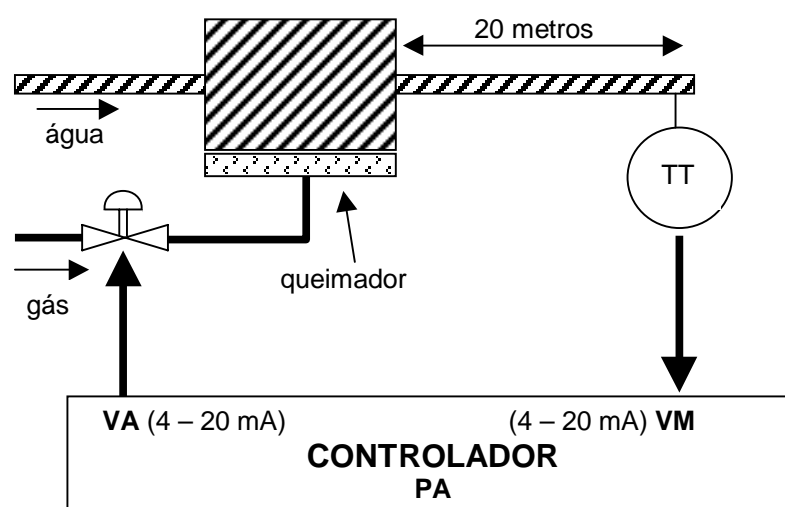


Figura 45: Processo de Aquecimento de Água

Na figura, observa-se que o transmissor de temperatura (TT) fica perto do consumidor, que fica a 20 metros do ponto de aquecimento da água. Processos como este são bons exemplos de como podem ser introduzidos “tempos mortos”. Isto porque a água aquecida no ponto de aquecimento leva algum tempo para percorrer a distância até o ponto de medição junto do consumidor. Tempos mortos foram discutidos anteriormente (ver figura 11).

Algumas hipóteses foram assumidas no modelamento deste processo:

Assume-se que a água que chega ao ponto de aquecimento sobre o queimador tem temperatura fixa, de 30 °C.

Assume-se que a vazão de água é constante.

Assume-se que a pressão do gás é constante.

A seguir define-se algumas características deste processo e dos elementos utilizados:

A água aquecida deve ter sua temperatura programável entre 50 °C e 80 °C.

O transmissor de temperatura TT tem saída de 4 – 20 mA, e se comporta de forma linear, de tal maneira que 4 mA correspondem a 30 °C e 20 mA correspondem a 130 °C.

Assume-se que, para aumentar em 10 °C a temperatura da água, é necessário injetar 1 m³/h de gás. Este comportamento é linear.

A válvula de gás se fecha com 4 mA, injetando 0 m³/h de gás. Por outro lado, com 20 mA, ela injeta 8 m³/h de gás.

Descrição dos Módulos Analógicos

Conforme pode ser visto na figura 12, necessita-se de uma saída analógica 4 – 20 mA, e de uma entrada analógica de 4 – 20 mA, como interfaces entre o controlador e o processo.

Internamente ao controlador, estas faixas de 4 – 20 mA correspondem a faixas numéricas em operandos M (VM e VA). Estas faixas de valores numéricos podem variar em função dos módulos de entrada e saída analógica selecionados. Neste exemplo, assume-se o seguinte:

entrada analógica VM (0 a 30000):

VM = 0 ---> 4 mA ---> 30 °C

VM = 30000 ---> 20 mA ---> 130 °C

saída analógica VA (0 a 10000):

VA = 0 ---> 4 mA = 0 ---> 0 m³/h

VA = 10000 ---> 20 mA ---> 8 m³/h

Ponto de Ajuste

O operando PA deve ser utilizado para programar a temperatura desejada, entre 50 °C e 80 °C.

Como este operando deve ser comparado com VM, ele deve ter a mesma faixa numérica de VM, ou seja:

PA = 0 ---> 30 °C

PA = 30000 ---> 130 °C

Ou para restringir a faixa entre 50 °C e 80 °C:

PA = 6000 ---> 50 °C

PA = 15000 ---> 80 °C

Blocodiagrama Geral e Valores Limites

A figura 13 mostra um blocodiagrama geral do sistema (controlador + processo), onde dentro do controlador mostra-se o módulo F-PID16.056. Observar que PA, VM e VA são operandos M.

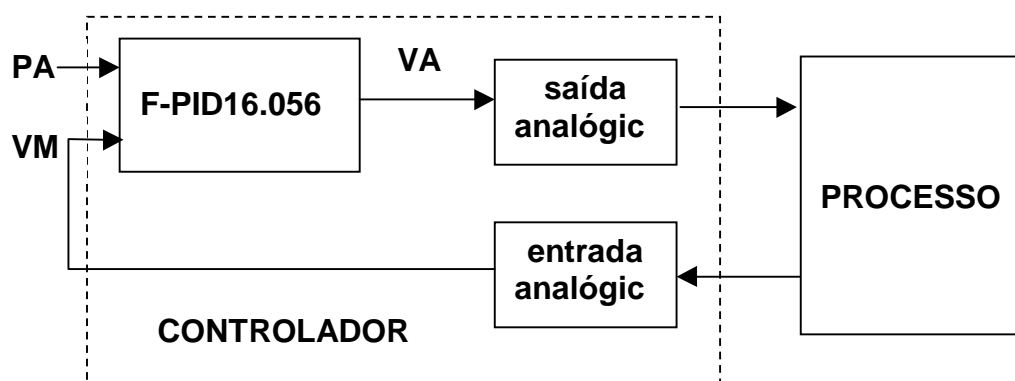


Figura 46: Blocodiagrama Geral

PA:

mínimo = 6000 (50 °C)

máximo = 15000 (80 °C)

VM:

mínimo = 0 (30 °C)

máximo = 30000 (130 °C)

VA:

mínimo = 0 (0 m³/h)

máximo = 7500 ---> (6 m³/h)

Observa-se que no caso de VA, embora a válvula tenha capacidade de injetar 8 m³/h, deseja-se limitar esta vazão em 6 m³/h.

Parâmetros do Processo

A figura 14 mostra o resultado de um teste de malha aberta sobre o processo. Para executar este teste, utilizou-se diretamente as variáveis VA e VM, com suas unidades internas.

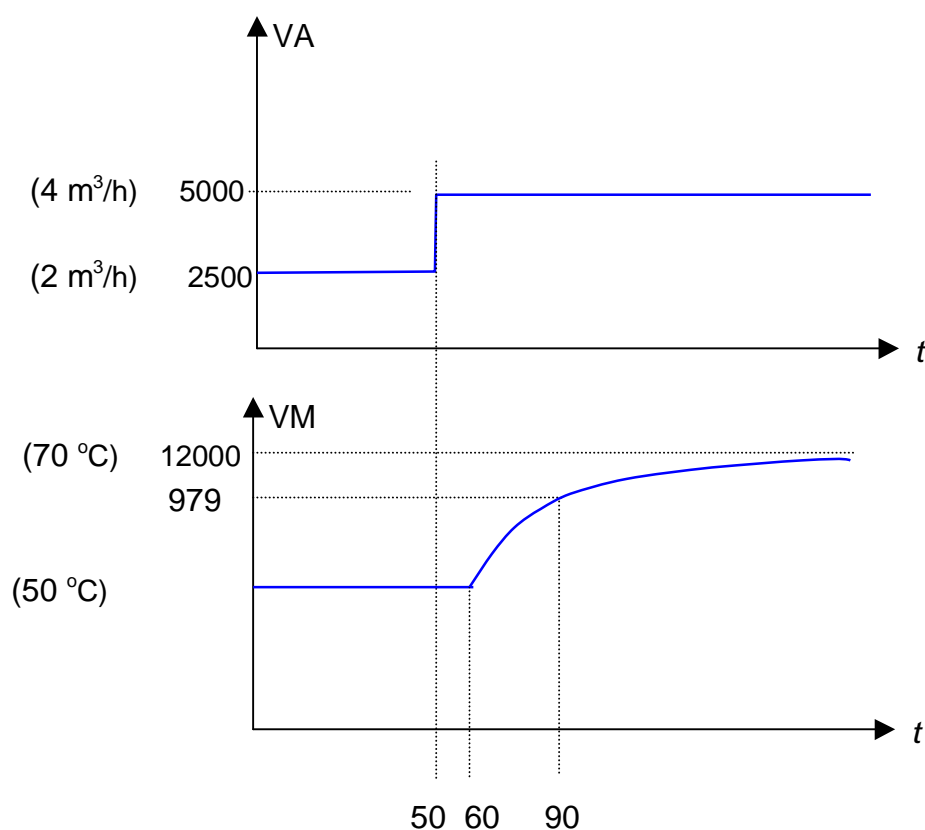


Figura 47: Teste de Malha Aberta

A partir desta figura pode-se determinar os 3 parâmetros básicos, conforme explicado anteriormente no capítulo **Notas de Aplicação**.

$T_m = 10$ segundos (tempo morto, visto que o degrau foi aplicado em $t = 50$ s e a resposta iniciou em $t = 60$ s).

$T = 30$ segundos (constante de tempo, visto que a resposta iniciou em $t = 60$ s, e atingiu 63,21% da excursão em $t = 90$ s:

$$9792 = 6000 + (12000 - 6000) * 0,6321.$$

$K = 2.4$ (ganho estático do processo)

$$2.4 = 12000 - 6000$$

$$5000 - 2500$$

Sintonia do Controlador

Já que foi realizado o teste de malha aberta, será utilizado o segundo método de sintonia descrito no capítulo **Notas de Aplicação**.

Para utilizar este método, além dos parâmetros do processo determinados na seção anterior (T_m , T e K), também é necessário que o usuário informe outros 2 parâmetros:

T_r , ou tempo de resposta desejado. Neste exemplo, será arbitrado em 10 segundos (um terço da constante de tempo em malha aberta).

dt , ou tempo de ciclo da F-PID16.056. Conforme comentado anteriormente, este tempo deve ser 10 vezes menor do que a constante de tempo em malha aberta, ou ainda menor. Portanto, o valor deve ser menor que 3 segundos. Selecionou-se $dt = 1$ segundo.

Agora, é possível aplicar as equações do método:

$$GP = T / (K * (T_r + T_m + Dt/2)) = 30 / (2.4 * (10 + 10 + 1/2)) = 0,609$$

$$T_i = T = 30 \text{ s/rep}$$

$$T_d = T_m/2 + Dt/4 = 10/2 + 1/2 = 5.25 \text{ s}$$

Utilização da F-PID16.056

A cada um segundo, deve-se executar a F-PID16.056, acionando sua entrada HABILITA durante uma única varredura.

A entrada AUTOMÁTICO/MANUAL pode ser controlada durante a operação do processo. Normalmente o processo estará em automático.

Para este processo, a entrada REVERSA/MANUAL deverá estar no estado 0 (reversa). O processo exige controle reverso pois, no caso de um aumento de VM, o controlador deve diminuir VA a fim de controlar o processo. Em outros termos, se a temperatura aumenta, deve-se fechar a válvula.

Operando TMXXXX:

$$\text{posição } 0 = GP \times 10 = 6$$

$$\text{posição } 1 = T_i \times 10 = 300$$

$$\text{posição } 2 = T_d \times 100 = 525$$

$$\text{posição } 3 = dt \times 100 = 100$$

$$\text{posição } 4 = \text{valor máximo da saída} = 7500$$

$$\text{posição } 5 = \text{valor mínimo da saída} = 0$$

posição 6 = zona morta = 0 (desabilitada)

posição 7 = variação máxima permitida = 0 (desabilitada)

posições 8 a 29, inicializar com zeros somente na energização do CP

Operando AXXXX de controle: zerar todos os bits na inicialização.

Comparação com o F-PID.033

O módulo F-PID16.056 foi desenvolvido visando melhorar a interface com o usuário, otimizar o tempo de execução e torná-lo compatível com variáveis de 16 bits ou menos resolução.

As principais alterações foram:

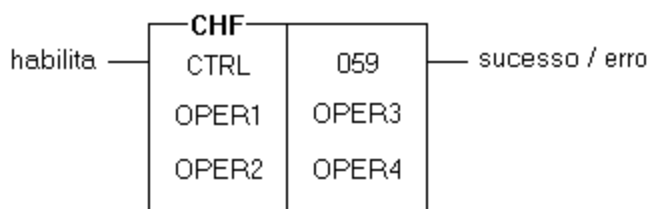
- Entradas e saídas com intervalo (“range”) de -30000 a 30000;
- Entrada de parâmetros sem pré-cálculo (entrada direta de G_p , T_i , T_d e dt);
- Intervalo da amostragem (dt) de 10ms a 10s, enquanto que no F-PID.033 o limite mínimo é de 100ms.

Juntamente com estas alterações, um conjunto de novas características foi acrescentado ao F-PID anterior. A seguir a tabela 5 traz uma comparação das características entre o módulo F-PID16.056 e o F-PID.033.

CARACTERÍSTICA	F-PID.033	Novo PID
Parâmetros programados diretamente no formato ISA (G_p , T_i , T_d)		X
Ação derivativa calculada em função do erro ou da variável medida.		X
Ação derivativa calculada sobre 3 amostragens	X	X
Ação direta ou reversa	X	X
Intervalo de entrada e saída de -100% a +100%		X
Dead band		X
Dessaturação da ação integral (“anti-reset windup”)	X	X
Entrada para <i>feedforward</i> / <i>bias</i> (deslocamento)	X	X
Inibição do termo derivativo	X	X
Inibição do termo integral	X	X
Inibição do termo proporcional		X
Limitação da taxa de crescimento		X
Limites de saída ajustáveis	X	X
Modo manual / automático	X	X
Acompanhamento da saída no modo manual e comutação manual/automática balanceada (“Output tracking” e “bumpless transfer”).	X	X

Tabela 12: Comparação entre o F-PID.033 e o F-PID16.056

F-CTRL.059 - Módulo F para Controle Avançado



Introdução

O módulo função F-CTRL.059 implementa os algoritmos de controle avanço/atraso (lead/lag), retardo de primeira ordem e derivador com retardo de primeira ordem. Cada modo de operação (algoritmo) é selecionado através de um índice no módulo F-CTRL.059.

A partir de um valor de entrada, o módulo calcula o valor de saída em função do algoritmo selecionado. Todos os módulos utilizam duas constantes, uma constante de tempo T e uma segunda constante K cuja função varia conforme o algoritmo selecionado. Os algoritmos são executados de modo discreto, necessitando que o tempo de disparo da função seja declarado juntamente com os parâmetros.

Estas funções são utilizadas em algoritmos de controle avançado para otimização da malha de controle. São geralmente utilizadas em conjunto com uma função PID.

Retardo de 1ª Ordem

Quando selecionado o algoritmo de retardo de 1ª ordem, o módulo F aplica sobre o valor do operando de entrada (V_i) um retardo temporal de 1ª ordem. O valor de saída (V_o) desta função é proporcional à entrada, porém, retardado conforme uma função exponencial.

Este algoritmo necessita de duas constantes. Uma constante de tempo T que, numa analogia com um circuito RC, representa sua constante de tempo de carga (63,212% do valor final) e uma constante de ganho proporcional K .

No domínio frequência (s), o retardo temporal de 1ª ordem efetua a seguinte função de transferência:

$$V_o(s) = \frac{K}{1 + T \cdot s} \times V_i(s)$$

Onde $V_i(s)$ e $V_o(s)$ são as transformadas de Laplace dos sinais de entrada e saída.

Sua resposta ao degrau é representada pela figura 4-23, onde pode ser observada a sua constante de tempo T associada ao valor V' , que representa 63,212% da diferença entre o valor inicial e o valor final.

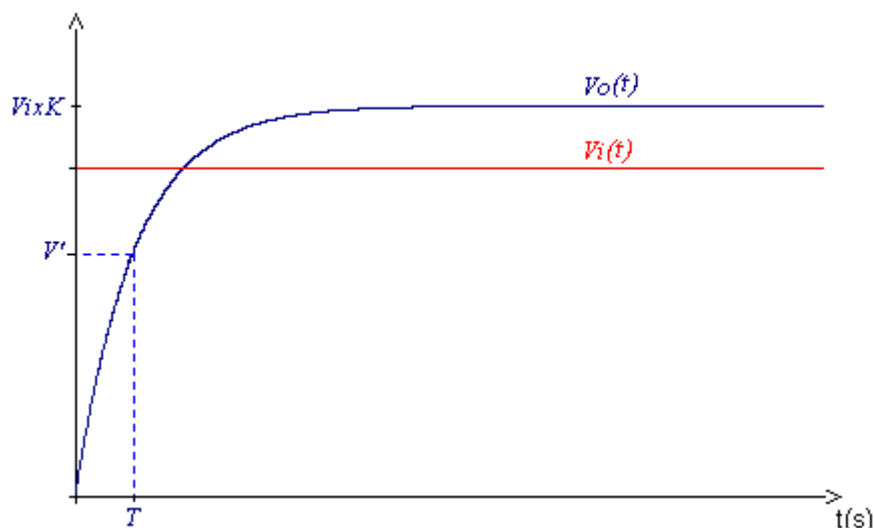


Figura 4-23 Retardo Temporal de 1ª Ordem

Derivador com Retardo de 1ª Ordem

Quando este algoritmo é selecionado, o módulo F aplica sobre o valor do operando de entrada a sua derivada juntamente com um retardo temporal de 1ª ordem. O valor de saída é a derivada da entrada V_i com amortecimento retardado conforme uma função exponencial.

Este algoritmo necessita de duas constantes. Uma constante de tempo T que pode ser entendida, na mesma analogia do retardo temporal de primeira ordem, como a constante de tempo de descarga de um circuito RC. A segunda constante é a constante derivativa K que, dividida pela constante T , determinará uma outra constante que pode ser tratada como um ganho.

No domínio frequência (s), o derivador com retardo temporal de 1ª ordem efetua a seguinte função de transferência:

$$V_o(s) = \frac{K \cdot s}{1 + T \cdot s} \times V_i(s)$$

Onde $V_i(s)$ e $V_o(s)$ são as transformadas de Laplace dos sinais de entrada e saída.

Sua resposta ao degrau é representada pela figura 4-24. No instante inicial ($t = t_0$) observa-se que a saída desta função (V_o) é o degrau de entrada, com amplitude A , multiplicada pela razão K/T . No instante $t = t_0 + T$, a saída do sistema equivale a V' , ou seja, 36,788% de $A \times K/T$. Quando a entrada V_i fica constante, a saída desta função retorna a zero com um retardo temporal de 1ª ordem.

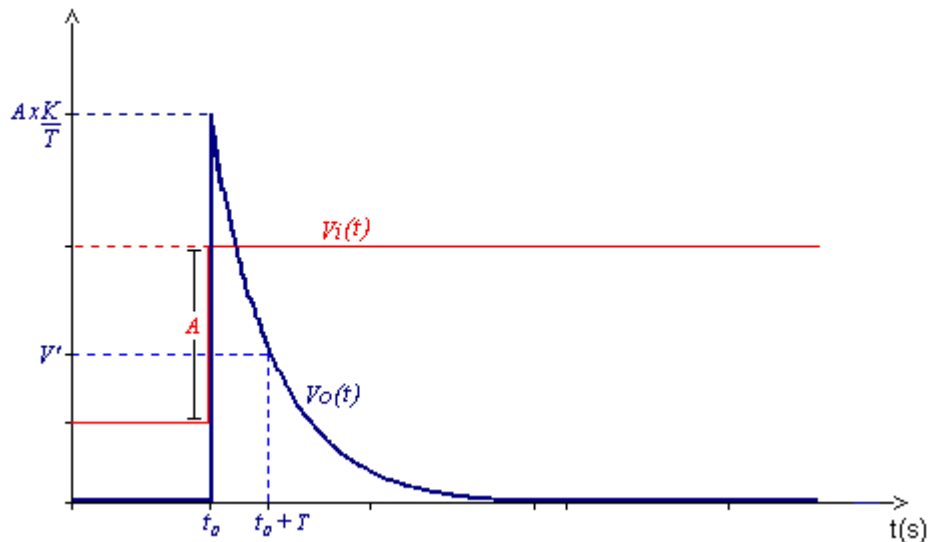


Figura 4-24 Derivador com Retardo Temporal de 1ª Ordem

É importante lembrar que o degrau não é visto pelo módulo F-CTRL.059 como uma variação instantânea, mas sim como uma variação entre duas amostragens. Caso contrário sua derivada teria valor infinito.

Avanço/ Atraso (*lead/lag*)

Quando este algoritmo é selecionado, o módulo F aplica sobre o valor do operando de entrada o avanço ou o atraso conforme a relação entre as constantes declaradas.

O algoritmo necessita de duas constantes. Uma constante de tempo T que, da mesma forma que os algoritmos anteriores, pode ser entendida como a constante de tempo de um circuito RC. E uma constante K que, juntamente com a constante T , definirá o comportamento do algoritmo como avanço ou atraso.

Sempre que a constante de tempo T for maior que a constante K , o algoritmo se comportará como atraso. Quando K for maior que T seu comportamento será de avanço. As constantes K e T são também conhecidas como constantes de avanço e atraso, respectivamente.

No domínio frequência (s), o avanço/atraso efetua a seguinte função de transferência:

$$Vo(s) = \frac{1 + K \cdot s}{1 + T \cdot s} \times Vi(s)$$

Onde $Vi(s)$ e $Vo(s)$ são as transformadas de Laplace dos sinais de entrada e saída.

A resposta ao degrau do avanço é representada pela figura 4-25. No instante $t = t_0$ observa-se que a saída desta função $Vo(t_0)$ equivale a V'' que pode ser descrita como

$$V'' = Vi(t) + A \times \frac{K}{T}, \text{ para } t < t_0,$$

ou seja, o valor da entrada antes do salto somado ao produto da amplitude do degrau aplicado na entrada (A) pela razão K/T . No instante $t = t_0 + T$, a saída do sistema equivale a V' , ou seja, 36,788% da diferença entre o valor máximo (V'') e o valor de $Vi(t)$ para $t > t_0$ somado a um deslocamento igual a $Vi(t_0)$.

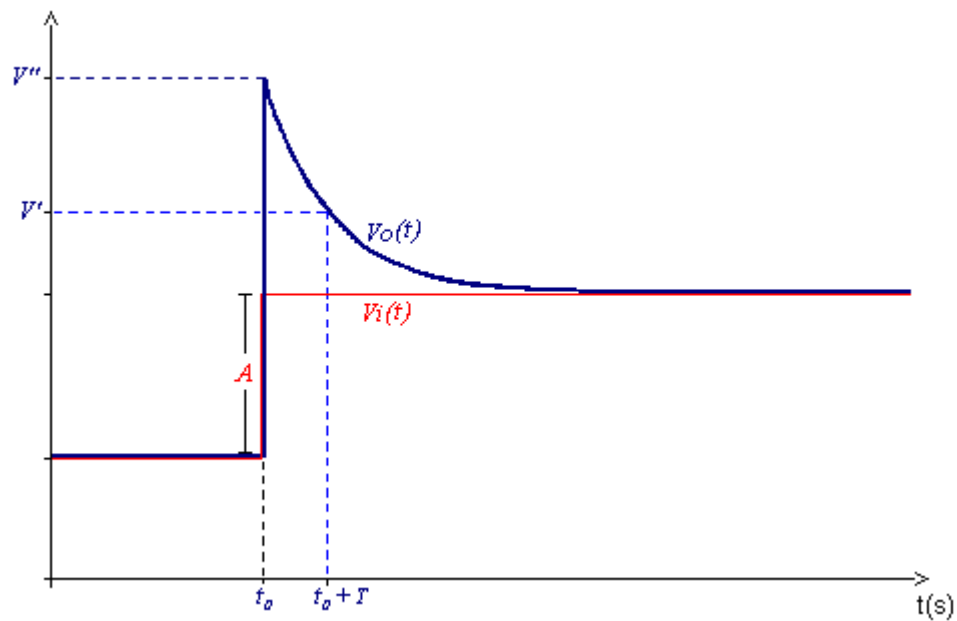


Figura 4-25 Avanço

A resposta ao degrau do atraso é representada pela figura 4-26. No instante $t = t_0$ observa-se que a saída desta função $V_o(t_0)$ equivale a V'' que, da mesma forma que o avanço, pode ser descrita como:

$$V'' = V_i(t) + A \times \frac{K}{T}, \text{ para } t < t_0,$$

diferindo-se do gráfico do avanço porque K é menor que T . No instante $t = t_0 + T$, a saída do sistema equivale a V' , ou seja, 63,212% da diferença entre o valor de $V_i(t)$, para $t > t_0$, e o valor de V'' , somado a um deslocamento igual a $V_i(t_0)$.

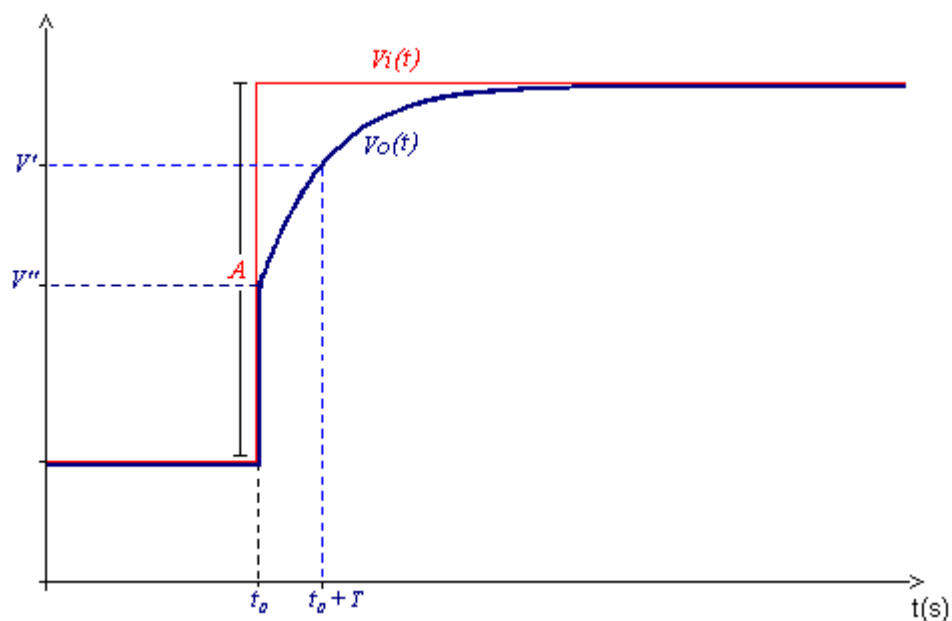


Figura 4-26 Atraso

Programação

Operandos

As células da instrução CHF utilizada para a chamada da função são programadas do seguinte modo:

- **OPER1** - Especifica o número de parâmetros que são passados para a função em OPER3. Este operando deverá ser obrigatoriamente uma constante memória com valor 8 (%KM+00008).
- **OPER2** - Especifica o número de parâmetros que são passados para a função em OPER4. Este operando deverá ser obrigatoriamente uma constante memória com valor 0 (%KM+00000).
- **OPER3** - Contém os parâmetros que são passados para a função, declarados quando a instrução CHF for editada. O número de parâmetros editáveis é especificado em OPER1, sendo fixo em 8 para este módulo:
 - **%KMXXXX** – Constante memória que indica qual o algoritmo será utilizado, pode assumir os seguintes valores:
 - %KM0000 – retardo temporal de primeira ordem;
 - %KM0001 – derivativo com retardo temporal de primeira ordem;
 - %KM0002 – avanço/atraso (lead/lag) .
 - **% KMXXXX** – Constante com o valor do intervalo de amostragem . Assume valores de 0,01 a 10s, devendo ser multiplicado por 100 para ser declarado neste campo.
 - **%MXXXX** – Memória com o valor da constante de tempo T . Assume valores de 0,01 a 320s, devendo ser multiplicado por 100 para ser declarado neste campo.
 - **%MXXXX** – Memória com o valor da constante K . Assume valores de 0,01 a 320, devendo ser multiplicado por 100 para ser declarado neste campo.
 - **%MXXXX** – Memória com o valor de entrada num intervalo de -32768 a +32767.
 - **%MXXXX** – Memória com o valor de saída num intervalo de -30000 a +30000.
 - **%MXXXX** – Uso interno. Não deve ser alterado.
 - **%MXXXX** – Uso interno. Não deve ser alterado.
- **OPER4** - Não utilizado.

Entradas e Saídas

Descrição das entradas:

- habilita - quando esta entrada está energizada a função é chamada, sendo analisados os parâmetros programados na instrução CHF. Caso o número de parâmetros ou seu tipo sejam diferentes das necessidades da função, haverá a desenergização da saída sucesso/erro. Se estiverem corretos, o cálculo do algoritmo selecionado é realizado.

Descrição das saídas:

- sucesso(1) /erro (0) - É energizada quando a função foi corretamente executada. Não é energizada caso ocorram erros na especificação dos operandos, tentativa de acesso a operandos não declarados ou parâmetros inválidos.

Utilização

Este módulo F está disponível para os CPs AL-2003, AL-2004, PO3145, PO3142 e PO3242.

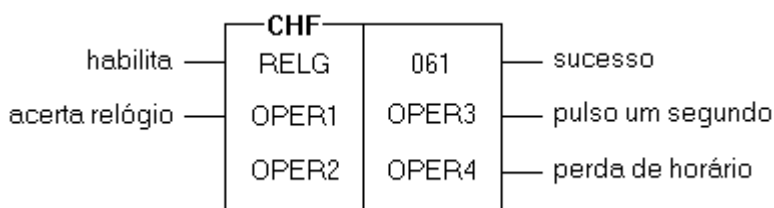
Características de Funcionamento

A cada intervalo de amostragem o valor de entrada da função é aplicado sobre o algoritmo e a saída da função atualizada.

Observa-se que o algoritmo é aplicado de forma discreta de modo que o tempo de amostragem (dt) deve ser da ordem de 10 vezes menor que a constante de tempo T para se obter um resultado satisfatório. O intervalo entre amostragens de um laço do módulo F-CTRL.059 pode variar de 0,01 a 10 segundos. É de responsabilidade do usuário programar um “disparador” da função, ou seja, um trecho de programa aplicativo que somente habilite o módulo F nos intervalos de tempo desejados. Aconselha-se utilizar um módulo E018, este módulo é executado dentro de um intervalo de tempo fixo que pode ser utilizado para gerar uma ou mais bases de tempo para a execução de um ou mais laços do F-CTRL.059. Nota-se ainda que o valor do intervalo de amostragem declarado nos parâmetros deve coincidir com o intervalo de tempo das chamadas do "disparador".

É importante lembrar que a atualização das entradas e saídas ocorrem na mesma ordem de tempo de um ciclo do CP. Sempre que o tempo de ciclo do CP for maior que o tempo de amostragem aconselha-se o uso da instrução AES para o CP AL-2003 e AL-2004 ou F-AES para a Série Ponto.

F-RELG.061 - Função para Acesso ao Relógio de Tempo Real do QK801 e QK2000



Introdução

A função F-RELG.061 realiza o acesso ao relógio de tempo real contido nas UCPs QK2000 e QK801. O relógio possui horário e calendário completos, permitindo o desenvolvimento de programas aplicativos que dependam de bases de tempo precisas. A informação de tempo é mantida mesmo com a falta de alimentação da UCP, pois o relógio permanece alimentado por bateria.

Programação

Operandos

As células da instrução CHF utilizada para a chamada da função são programadas do seguinte modo:

- **OPER1** - Especifica o número de parâmetros que são passados para a função em OPER3. Este operando deverá ser obrigatoriamente uma constante memória com valor 2 (%KM+00002).
- **OPER2** - Deve ser um operando do tipo constante memória com valor 0 (%KM+00000). Determina o número de parâmetros possíveis de serem programados na janela de edição de OPER4. Como esta função não necessita de nenhum parâmetro em OPER4, o valor de OPER2 é zero.
- **OPER3** - Contém os parâmetros que são passados para a função, declarados através de uma janela visualizada no MasterTool quando a instrução CHF for editada. O número de parâmetros editáveis é especificado em OPER1, sendo fixo em 2 para este módulo:

—* **%MXXXX** ou **%TMXXXX** - Especificação dos operandos para onde são lidos os valores do relógio. Se este parâmetro for especificado como memória, os valores são lidos para a memória declarada e as 6 subseqüentes. Se for especificado como tabela, os valores são colocados a partir da posição 0 até 6. Caso os operandos não estejam declarados, a leitura dos valores de tempo não é realizada e as saídas da instrução são desligadas. É possível o uso de tabelas com mais de 7 posições, sendo que a função ignora as posições excedentes. Os valores são lidos dos operandos na seguinte seqüência:

Operando	Posição Tabela	Conteúdo	Formato
%MXXXX	0	Segundos	00XX
%MXXXX + 1	1	Minutos	00XX
%MXXXX + 2	2	Horas	00XX
%MXXXX + 3	3	Dia do mês	00XX
%MXXXX + 4	4	Mês	00XX
%MXXXX + 5	5	Ano	00XX
%MXXXX + 6	6	Dia da semana	00XX

Tabela 4-24 Valores Lidos do Relógio (F-RELG.061)

O conteúdo destes operandos pode ser utilizado a qualquer momento, mas são atualizados com a hora real do relógio apenas quando a função for executada. É utilizado o formato 24 horas na contagem do tempo e os dias da semana são contados com valores de 1 a 7:

Valor	Dia da Semana
1	Domingo
2	Segunda-feira
3	Terça-feira
4	Quarta-feira
5	Quinta-feira
6	Sexta-feira
7	Sábado

Tabela 4-25 Valores dos Dias da Semana (F-REL G.061)

—* %MXXXX ou %TMXXXX - Especificação dos operandos de onde são acertados os valores do relógio, com o acionamento da entrada de acerto da função. Se este parâmetro for especificado como memória, os valores são copiados da memória declarada e as 6 subseqüentes. Se for especificado como tabela, os valores são copiados da posição 0 até 6. Caso os operandos não estejam declarados, o acerto não é realizado e as saídas da instrução são desligadas. Os valores a serem copiados para o relógio devem ser colocados nos operandos na mesma seqüência dos operandos de leitura (segundos, minutos, horas, dia do mês, mês, ano e dia da semana).

- **OPER4** - Não utilizado.

Entradas e Saídas

Descrição das entradas:

- **habilita** - quando esta entrada está energizada a função é chamada, sendo analisados os parâmetros programados na instrução CHF. Caso os mesmos estejam incorretos, todas as saídas da instrução são desenergizadas. Se estiverem corretos, os valores de tempo do relógio são transferidos para os operandos memória ou para a tabela declarada como primeiro parâmetro em OPER3, a saída sucesso é energizada e a saída pulso um segundo é ligada por uma varredura a cada segundo.
- **acerta relógio** - quando energizada, os valores dos operandos declarados como segundo parâmetro em OPER3 são acertados no relógio, caso estejam com valores corretos. Enquanto esta entrada estiver acionada o tempo não é contado, permanecendo a saída pulso um segundo desenergizada.

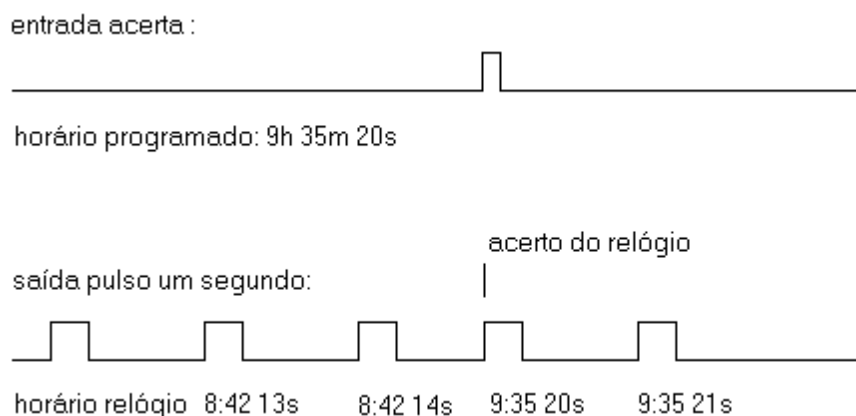


Figura 4-27 Ex. de Diagrama de Tempos da Entrada Acerta

Descrição das saídas:

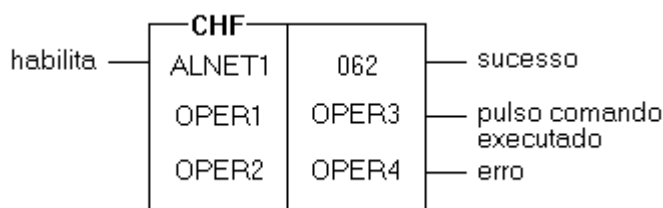
- **sucesso** - é energizada quando a função foi corretamente executada.

- **pulso um segundo** - indica se houve uma mudança no contador de segundos do relógio. O pulso dura uma varredura e pode ser usado para sincronizar eventos do programa aplicativo.
- **perda de horário** - esta saída é ligada caso o relógio tenha ficado sem a alimentação da bateria durante falha na alimentação principal. É desacionada com o acerto do relógio.

Utilização

Este módulo pode ser utilizado nas UCPs QK801 e QK2000/MSP.

F-ALNET1.062 - Função Interpretador do Protocolo ALNET I para QK801



Introdução

A função F-ALNET1.062 implementa a comunicação no canal serial secundário do controlador QK801, permitindo que o mesmo receba e execute comandos do protocolo ALNET I como um dispositivo escravo. Desta forma, o CP QK801 pode ser conectado a uma rede de supervisão ALNET I ou a equipamentos periféricos através deste canal.

Programação

Operandos

As células da instrução CHF utilizada para a chamada da função são programadas do seguinte modo:

- **OPER1** - Especifica o número de parâmetros que são passados para a função em OPER3. Este operando deverá ser obrigatoriamente uma constante memória com valor 2 (%KM+00002).
- **OPER2** - Deve ser um operando do tipo constante memória com valor 0 (%KM+00000). Determina o número de parâmetros possíveis de serem programados na janela de edição de OPER4. Como esta função não necessita de nenhum parâmetro em OPER4, o valor de OPER2 é zero.
- **OPER3** - Contém os parâmetros que são passados para a função, declarados através de uma janela visualizada no MasterTool quando a instrução CHF for editada. O número de parâmetros editáveis é especificado em OPER1, sendo fixo em 2 para este módulo:

—* **%KMXXXX** - Especificação do baudrate da comunicação no canal serial secundário. O valor da constante corresponde diretamente ao número de bauds, podendo assumir os valores 9600, 4800, 2400, 1200, 600 ou 300.

—* **%KMXXXX** - Especificação do uso dos sinais de MODEM (RTS, CTS, DTR e DSR), quando o canal serial secundário for utilizado no padrão RS-232. Se programada com o valor 0, a comunicação não utiliza os sinais de MODEM. Se programada com o valor 1, estes sinais são utilizados. Esta constante deve ser programada com o valor 0 quando o canal serial secundário seguir o padrão RS-485.

- **OPER4** - Não utilizado.

Entradas e Saídas

Descrição das entradas:

- **habilita** - quando esta entrada está energizada a função é chamada, sendo analisados os parâmetros programados na instrução CHF. Caso algum esteja incorreto, a saída de erro é energizada. Se estiverem corretos, a função interpreta, executa e responde os comandos do protocolo ALNET I recebidos no canal serial secundário do CP.

ATENÇÃO:

Para o correto funcionamento, a entrada habilita deve estar energizada no primeiro ciclo de execução do programa aplicativo. No primeiro ciclo é executada a configuração do canal serial secundário do OK801.

Descrição das saídas:

- **sucesso** - é energizada quando a função foi chamada (está presente no controlador).
- **pulso comando executado** - indica que foi processado um comando recebido pelo canal serial, sendo energizada por uma varredura.
- **erro** - é energizada permanentemente caso ocorra erro na programação das constantes da função ou por uma varredura caso ocorra erro na comunicação serial.

Protocolo ALNET I

O segundo canal serial do CP QK801 é orientado para o uso com sistemas supervisórios ou interfaces homem-máquina, não executando os comandos referentes aos módulos do programa aplicativo, nem os comandos de mudança de estado do CP.

Os comandos ALNET I executados pelo módulo F são mostrados na tabela a seguir, indicando-se a versão do protocolo ao qual pertencem. Os comandos da versão 1.00 são utilizados em redes que contenham controladores da série AL-1000 ou AL-500. Os comandos da versão 2.00 podem ser usados em redes que contenham somente controladores das séries AL-600 ,AL-2000, AL-3000 ou QUARK, não possuindo nenhum controlador das séries AL-1000 ou AL-500.

Núm.	Descrição do comando	V1.00	V2.00
002	Força operando simples AL-1000	x	
004	Libera todos os forçamentos	x	x
006	Monitora operando simples AL-1000	x	
009	Desabilita saídas digitais	x	x
010	Habilita saídas digitais	x	x
012	Força posição de tabela AL-1000	x	
013	Monitora posição de tabela AL-1000	x	
014	Força bloco de tabela AL-1000	x	
015	Monitora bloco de tabela AL-1000	x	
032	Recebe módulo de programa		x
037	Lê status		x
038	Lê diretório de módulos de programa		x
039	Lê status de módulo de programa		x
040	Monitora operandos simples		x
041	Monitora operandos tabela		x
042	Lê status de forçamentos		x
129	Força operandos simples		x
130	Força operandos tabela		x
131	Libera operandos		x
133	Escreve operandos simples		x
134	Muda o nível de proteção		x
135	Muda a senha		x

Tabela 4-26 Comandos Executados pelo Módulo F-ALNET1.062

A tabela a seguir apresenta os comandos que não são executados pelo módulo F-ALNET1.062:

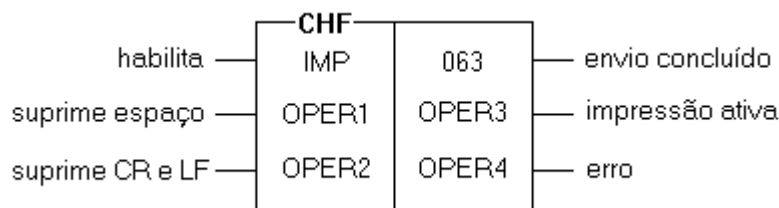
Núm.	Descrição do comando	V1.00	V2.00
005	Passa para modo programação	*	*
007	Passa para modo ciclado	*	*
008	Executa um ciclo	*	*
011	Passa para modo execução	*	*
031	Pedido de carga de módulo de programa		*
033	Remove módulo de programa		*
034	Transfere módulo em EPROM para RAM		*
035	Reabilita módulo em EPROM		*
036	Compacta memória RAM		*
045	Transfere módulo de RAM para Flash EPROM		*
046	Apaga memória Flash EPROM		*
047	Lê status da comunicação		*
193	Carga de módulo de programa		*

Tabela 4-27 Comandos Não Executados pelo Módulo F-ALNET1.062

Utilização

Este módulo pode ser utilizado na UCP QK801.

F-IMP.063 - Função para Impressão de Caracteres ASCII



Introdução

A função F-IMP.063 permite o envio de caracteres ASCII através do canal serial principal dos controladores programáveis para dispositivos como impressoras e terminais de vídeo, permitindo a impressão ou a visualização de textos pré-definidos em conjunto com valores de operandos.

Esta função pode trabalhar de duas formas distintas: enviando texto e operandos memória em conjunto ou enviando uma sequência de valores de até 255 operandos memória, sem textos. Os valores dos operandos memória são codificados para o formato ASCII, para então serem enviados através da interface de comunicação serial.

Programação

Operandos

As células da instrução CHF utilizada para a chamada da função são programadas do seguinte modo:

- **OPER1** - Especifica o número de parâmetros que são passados para a função em OPER3. Este operando deve ser obrigatoriamente uma constante memória com valor 3 (%KM+00003).
- **OPER2** - Deve ser um operando do tipo constante memória com valor 0 (%KM+00000). Determina o número de parâmetros possíveis de serem programados na janela de edição de OPER4. Como esta função não necessita de nenhum parâmetro em OPER4, o valor de OPER2 é zerado.
- **OPER3** - Contém os parâmetros que são passados para a função, declarados através de uma janela visualizada no MasterTool quando a instrução CHF for editada. O número de parâmetros editáveis é especificado em OPER1, sendo fixo em três para este módulo:

—* **%MXXXX ou %TMXXXX** - Quando especificado como %MXXXX, indica o operando inicial a ser enviado através do canal primário do CP. Quando especificado como %TMXXXX indica a tabela que contém o texto e/ou operandos memória a serem enviados. O número máximo de caracteres enviados é 255, permitindo o uso de uma %TM de no máximo 128 posições.

—* **%KM+XXXX** - Indica o número de operandos memória a serem enviados a partir do operando %MXXXX especificado no parâmetro anterior. Quando se utiliza %TMXXXX no parâmetro anterior, o valor deste parâmetro é desprezado pela função. O valor deste operando deve estar entre 1 e 255.

—* **%MXXXX** - Operando de controle da função.

ATENÇÃO:

O operando de controle é de uso exclusivo da função, não devendo ser alterado em nenhuma parte do programa aplicativo, sob pena de prejudicar a sua correta execução.

- **OPER4** - Não utilizado.

Descrição das entradas:

- **habilita** - quando esta entrada está energizada a função é chamada, sendo analisados os parâmetros programados na instrução CHF. Caso os mesmos estejam incorretos, a saída **erro** é energizada. Esta entrada deve permanecer energizada até que a saída **envio concluído** seja pulsada.

ATENÇÃO:

Uma vez iniciada, a execução da função F-IMP.063 deve prosseguir até o seu término (nível da saída **envio concluído** retornado para 0 após ter estado em 1). Desta forma, a instrução CHF que realiza a chamada a F-IMP.063 não deve ser saltada nem desenergizada.

- **suprime espaço** - esta entrada é utilizada apenas quando se especifica o primeiro parâmetro de OPER3 como %MXXXX. Quando energizada, suprime os 5 espaços em branco enviados entre o valor de cada operando %M.
- **suprime CR e LF** - esta entrada é utilizada apenas quando se especifica o primeiro parâmetro de OPER3 como %MXXXX. Quando energizada, suprime o envio dos caracteres CR (retorno de carro) e LF (nova linha) após o término do envio dos operandos %M.

Descrição das saídas:

- **envio concluído** - esta saída é energizada por uma varredura assim que o envio de caracteres esteja concluído.
- **impressão ativa** - esta saída é energizada enquanto os caracteres estiverem sendo enviados para o dispositivo de saída. Tão logo encerre-se o envio, a saída é desenergizada.
- **erro** - é energizada caso ocorra erro na especificação dos operandos, tentativa de acesso a operandos não declarados ou time-out no teste dos sinais de MODEM (CTS e RTS).

Utilização

- Número de chamadas - podem existir no programa aplicativo tantas chamadas da F-IMP.063 quanto necessário. No entanto, somente uma chamada permanece ativa no programa, até o final do envio dos seus caracteres, mesmo que outras estejam habilitadas. Após o término do envio dos caracteres ASCII da função ativa, um pulso de uma varredura na saída **envio concluído** indica que o canal de comunicação está liberado, sendo ativada a próxima chamada da F-IMP.063 que estiver habilitada.
- Impressão de textos - caso seja utilizado um operando %TM como primeiro parâmetro da função, a primeira posição da tabela subsequente ao último caractere a ser enviado deve possuir o valor 0. Este valor é reconhecido pela função como marcador de fim de texto. Caso este valor seja omitido, a função irá enviar todos os caracteres existentes na tabela até o limite de 255.

O texto a ser enviado deve ser armazenado na tabela antes da habilitação da chamada do módulo F-IMP.063, podendo serem utilizadas instruções CAB para esta finalidade. O texto pode ser visualizado e editado diretamente no formato ASCII durante a inserção da instrução CAB.

Os caracteres ASCII utilizados pela função estão na faixa de 0 a 127.

- Utilização dos sinais de MODEM (CTS e RTS) - de forma a permitir a utilização de dispositivos que utilizem ou não sinais de MODEM, a função pode realizar tratamento dos sinais RTS e CTS ou desprezá-los. Esta definição é feita via programador MasterTool. Caso se utilize a comunicação com sinais de MODEM, o tratamento dos sinais é feito apenas no envio do primeiro byte de cada comunicação. Desta forma, deve-se administrar o envio dos caracteres em função do tamanho do buffer de recepção do dispositivo destino.

Estando selecionado para uso dos sinais de MODEM, caso não ocorra acionamento do sinal CTS após 200 ms do acionamento do sinal RTS, a saída **erro** é energizada, devendo ser verificados os seguintes aspectos:

- * conexão do cabo que interliga o CP ao dispositivo de saída;
- * envio de caractere não reconhecido pelo dispositivo de saída;
- * incompatibilidade de configuração entre o CP e o dispositivo de saída;
- * overflow permanente do buffer de recepção do dispositivo de saída em função de seu tamanho x taxa de envio de caracteres.

- Configuração do canal serial - a função opera com a seguinte configuração de canal serial:

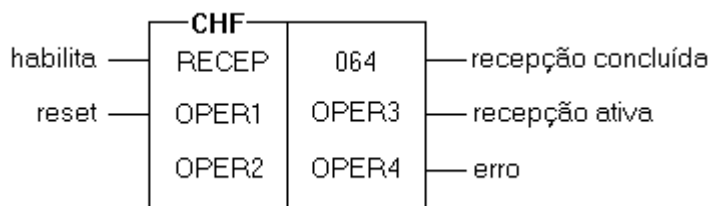
—* 8 bits de dados
—* sem paridade
—* baud rate configurável (via programador MasterTool)

Após o término da execução, a função reconfigura automaticamente o canal serial, permitindo a comunicação via protocolo ALNET I.

- Tempo de execução - quando a entrada habilita for energizada e não houver alguma outra impressão ativa, a função analisa os parâmetros, prepara o conjunto de caracteres a ser enviado e dispara a sua transmissão. Dependendo do conteúdo a ser enviado, esta preparação pode necessitar um tempo bastante longo para ser executada, aumentando significativamente o tempo de execução deste ciclo do programa aplicativo. Nos ciclos subsequentes a função somente testa o final do envio dos caracteres, com o tempo de execução bastante reduzido.

Esta função pode ser utilizada nas UCPs QK800, QK801 e QK2000/MSP.

F-RECEP.064 - Função para Recepção de Caracteres ASCII



Introdução

A função F-RECEP.064 permite a recepção de caracteres ASCII através do canal serial principal do CP.

Através desta função podem ser recebidos caracteres de quaisquer dispositivos com interface serial RS-232 que possuam 1 start bit, 7 bits de dados, 1 bit de paridade (par) e 1 stop bit.

A função pode ser configurada para receber um número máximo de 255 caracteres, que serão armazenados em operandos memória ou em uma tabela.

Programação

Operandos

As células da instrução CHF utilizada para a chamada da função são programadas do seguinte modo:

- **OPER1** - Especifica o número de parâmetros que são passados para a função em OPER4. Este operando deve ser obrigatoriamente uma constante memória com valor 4 (%KM+00004).
- **OPER2** - Deve ser um operando do tipo constante memória com valor 0 (%KM+00000). Determina o número de parâmetros possíveis de serem programados na janela de edição de OPER4. Como esta função não necessita de nenhum parâmetro em OPER4, o valor de OPER2 é zerado.
- **OPER3** - Contém os parâmetros que são passados para a função, declarados através de uma janela visualizada no MasterTool quando a instrução CHF for editada. O número de parâmetros editáveis é especificado em OPER1, sendo fixo em quatro para este módulo:

—* **%MXXXX ou %TMXXXX** - Operando de armazenamento. Quando especificado como %MXXXX indica o primeiro operando memória de recepção, a partir do qual serão armazenados os caracteres recebidos pelo canal serial. Quando especificado como %TMXXXX indica a tabela que deverá receber os caracteres ASCII.

—* **%KM+XXXXX** - Posição de armazenamento. Indica a primeira posição da tabela a ser ocupada pelos dados recebidos pelo canal serial, caso o operando de armazenamento (primeiro parâmetro) seja uma tabela. Se o primeiro parâmetro for uma memória (%MXXXX), o valor desta constante é desconsiderado.

—* **%KM+XXXXX** - Número de caracteres. Indica o número de caracteres a serem recebidos em uma comunicação.

ATENÇÃO:

Uma comunicação pode receber no máximo 255 caracteres. No caso de armazenamento em tabela deve-se atentar para que a tabela possua posições em número suficiente para a recepção de caracteres programada. Ou seja, a posição inicial da tabela mais o número de caracteres programados não deve ultrapassar o número de posições do operando tabela.

—* **%MXXXX** - Operando de controle da função.

ATENÇÃO:

O operando de controle é de uso exclusivo da função, não devendo ser alterado em nenhuma parte do programa aplicativo sob pena de prejudicar a sua correta execução.

- **OPER4** - Não utilizado.

Descrição das entradas:

- **habilita** - quando esta entrada está energizada a função é chamada, sendo analisados os parâmetros programados na instrução CHF. Caso os mesmos estejam incorretos, a saída de erro é energizada. Esta entrada deve permanecer energizada até que a saída envio concluído seja pulsada.

ATENÇÃO:

Para que os caracteres sejam corretamente recebidos e armazenados, enquanto estiver habilitada a chamada da função F-RECEP.064 não deve ser saltada através de alguma instrução bobina de salto.

- **reset** - esta entrada é utilizada para reinicializar a função, voltando a armazenar os caracteres recebidos a partir da primeira posição ou operando programado. Enquanto a mesma estiver energizada a função permanece inativa.

Descrição das saídas:

- **recepção concluída** - esta saída é energizada por uma varredura assim que o número total programado de caracteres for recebido.
- **recepção ativa** - esta saída é energizada enquanto os caracteres estiverem sendo recebidos. Após o término da recepção, com a chegada do último caracter programado, esta saída é desenergizada.
- **erro** - esta saída é energizada caso ocorra erro na especificação dos operandos, tentativa de acesso a operandos não declarados ou erro de paridade nos caracteres recebidos.

Utilização

Podem existir no programa aplicativo tantas chamadas da F-RECEP.064 quanto necessário. No entanto, somente uma chamada permanece ativa no programa, até o final da recepção dos número de caracteres programados, mesmo que outras estejam habilitadas. Após o término da recepção dos caracteres ASCII da função ativa, um pulso de uma varredura na saída **recepção concluída** indica que o canal de comunicação está liberado, sendo ativada a próxima chamada da F-RECEP.064 que estiver habilitada.

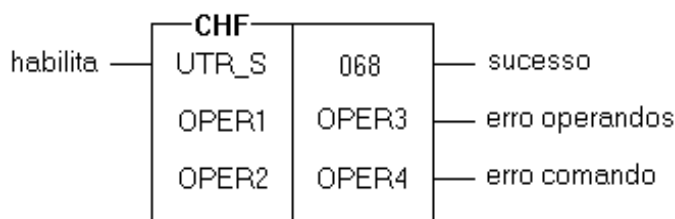
Os caracteres ASCII utilizados pela função estão na faixa de 0 a 127.

A função opera com a seguinte configuração de canal serial:

- 7 bits de dados
- 1 bit de paridade (par)
- baud rate configurável (via programador MasterTool)

Esta função pode ser utilizada nas UCPs QK800, QK801 e QK2000/MSP.

F-UTR_S.068 - Função para Acionamento de Saídas em UTRs



Introdução

A função F-UTR_S.068 realiza o acesso aos módulos de saídas digitais AL-3202, implementando acionamentos especiais para a utilização de CPs em unidades terminais remotas (UTRs).

O módulo AL-3202 possui 32 saídas digitais e funciona segundo o princípio "check before operate". Seus pontos podem ser configurados para operar como saídas comuns - acessadas por operandos %S através da varredura de E/S, saídas biestáveis, saídas "trip/close" ou saídas "rise/lower".

A função F-UTR_S.068 recebe comandos de acionamentos da estação supervisória através de uma tabela, envia-os para os módulos de saída AL-3202, lê os estados das operações e os coloca disponíveis para consultas. Para este tipo de operação, os módulos AL-3202 devem estar configurados no modo biestável, "trip/close", "trip/close SBO" ou "rise/lower".

A função F-UTR_S.068 deve ser utilizada em apenas uma lógica do programa, pois o controle de todos os módulos AL-3202 do CP são feitos por uma única chamada.

A função F-CBO.018 deve ser utilizada para configurar cada módulo AL-3202 presente no barramento.

Programação

Operandos

As células da instrução CHF utilizada para a chamada da função são programadas do seguinte modo:

- **OPER1** - Especifica o número de parâmetros que são passados para a função em OPER3. Este operando deverá ser obrigatoriamente uma constante memória com valor 1 (%KM+00001).
- **OPER2** - Deve ser um operando do tipo constante memória com valor 0 (%KM+00000). Determina o número de parâmetros possíveis de serem programados na janela de edição de OPER4. Como esta função não necessita de nenhum parâmetro em OPER4, o valor de OPER2 é 0.
- **OPER3** - Contém os parâmetros que são passados para a função, declarados através de uma janela visualizada no MasterTool quando a instrução CHF for editada. O número de parâmetros editáveis é especificado em OPER1, sendo fixo em 1 para esta função:

—* **TMXXXX** - Tabela onde o supervisor escreve os comandos para o acionamento das saídas e lê o estado correspondente. A função interpreta os comandos presentes nas suas primeiras 3 posições e armazena o estado resultante da operação nas posições seguintes, conforme o formato mostrado neste capítulo. Caso a tabela possua menos que 7 posições, os comandos não são interpretados pela função e a sua saída "erro operandos" é energizada.

- **OPER4** - Não utilizado.

Entradas e Saídas

Descrição das entradas:

- **habilita** - quando esta entrada está energizada a função é chamada, sendo analisados os parâmetros programados na instrução CHF. Caso os mesmos estejam corretos, somente a saída sucesso é energizada. Se estiverem incorretos, as saídas sucesso e erro operandos são energizadas simultaneamente.

Descrição das saídas:

- **sucesso** - indica que a função foi executada.
- **erro operandos** - é energizada caso ocorra erro na especificação dos operandos ou tentativa de acesso a operandos não declarados.
- **erro comando** - é energizada caso o comando seja rejeitado pelo módulo AL-3202. Neste caso, os campos de "estado" na tabela %TMXXXX devem ser analisados para saber o motivo do erro.

ATENÇÃO:

Esta saída só é energizada por uma varredura (pulso).

Utilização

Esta função pode ser utilizada somente nas UCPs AL-2002, AL-2003 e AL-2004.

Processamento da função

O programa supervisorio deve escrever os comandos de acionamento nas posições 0 a 2 da tabela declarada na função. As posições 3 a 5 só precisam ser lidas caso a saída erro comando tenha sido acionada.

A tabela %TMXXXX deve ser inicializada com zeros por uma instrução CAB na primeira varredura da UCP.

Formato da tabela %TMXXXX

0	Comando	Supervisorio escreve
1	Endereço	
2	Tempo	
3	Estado 0	Supervisorio pode ler
4	Estado 1	
5	Estado 2	
6	<reserv. função>	

Descrição dos campos da tabela:

Posição 0 - comando do supervisorio:

Código	Comando
1	Liga ponto biestável
3	Desliga ponto biestável
5	Rise
7	Lower
9	Trip
11	Close
13	Select trip
15	Select close
17	Operate trip
19	Operate close
21	Cancel

O supervisor aciona o comando através dos códigos mostrados na tabela acima. A função desliga o bit 0 da posição 0, após executar o comando. Ver formato:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	C	C	C	C	A

A = 1 - ativar comando

0 - comando executado

CCCC - código comando

Posição 1 - endereço do ponto a ser acionado:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	R	R	R	R	0	o	o	P	P	P

RRRR - posição do módulo no barramento (0-15)

oo - número octeto no módulo (0-3)

PPP - número do ponto a ser acionado no octeto (0-7 para comandos biestáveis, 0-3 para os demais comandos)

O endereço não é necessário para o comando 21.

Posição 2 - tempo do acionamento "rise/lower" (1 a 255 décimos de segundo).

Este tempo somente é necessário nos comandos 5 e 7.

Posição 3 - estado #0 do comando:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Er	x	x	x	x	Se	x	In	To	Oc	Sa	Co	Sc	Fo	Nc	x

Er - qualquer erro

Se - Operate/Cancel inválido

In - AL-3202 inativo
 To - Time out seleção
 Oc - Módulo ocupado
 Sa - Erro nas saídas
 Co - Comando inválido
 Sc - Comando Operate/Cancel após configuração
 Fo - Fonte 24V desligada
 Nc - Módulo não configurado
 x - Reservado

Posição 4 - estado #1 do comando:

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

<----- octeto 3 ----->								<----- octeto 2 ----->							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

Posição 5 - estado #2 do comando:

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

<----- octeto 1 ----->								<----- octeto 0 ----->							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

As posições 4 e 5 da tabela contém o estado de cada ponto de saída:

- "1" significa ponto de saída com defeito
- "0" significa ponto OK.

Só é preciso consultar essas posições se ocorrer erro nas saídas (bit Sa no estado #0)

Posição 6 - reservada para uso interno da função

Configuração

O módulo AL-3202 deve ser configurado para aceitar os comandos biestáveis, "rise/lower", e "trip/close". Esta configuração é realizada módulo a módulo pela função F-CBO.018.

Códigos de configuração:

0	acesso por operandos S (varredura de E/S)
1	modo biestável
2	modo "rise/lower"
3	modo "trip/close"
4	modo "trip/close" SBO

Situações de exceção

O módulo de saída CBO AL-3202 tem programação complexa, o que requer um correto entendimento da arquitetura para compreensão das situações de exceção.

Uma única chamada da função F-UTR_S acessa todas as placas AL-3202 de um CP, de modo que o estado reportado deve ser relacionado ao endereço da placa (posição 1 da TMXXXX).

Os bits de erro entre 1 e 7 são gerados pelo hardware (AL-3202) sendo específicos para um cartão. Os bits 8 a 10 e 15 são gerados pelo software (UCP), correspondendo a uma situação que envolve todo o CP.

Certos bits são do tipo "latch", que significa que o erro fica armazenado até que um novo comando seja executado.

Abaixo descrição detalhada do comportamento do bits de erro:

- **Er - qualquer erro** - O bit Er indica qualquer situação de erro. Os outros bits detalham que tipo de erro ocorreu. Concomitante com o bit Er, a saída 2 da função também é ligada, podendo ser testada diretamente pela aplicação.
- **Se - Operate/Cancel inválido** - Erro que acontece ao se tentar um "operate" ou "cancel" sem ter sido executado um comando "select" previamente, ou o ponto endereçado por "operate" não é o mesmo que o selecionado (endereço diferente), ou ainda se a seleção foi para "trip" e o "operate" foi para close e inversamente.
- **In - AL-3202 inativo** - Ocorre quando a UCP não consegue acessar o módulo AL-3202: chave de troca em STDBY, módulo AL-3202 retirado do bastidor, módulo AL-3202 em erro, endereçamento errado (campo RRRR na posição 1 de %TMXXXX), ou ainda, módulo não declarado no barramento.
- **To - Time out seleção (latch)** - Ocorre quando um comando "operate" ou "cancel" chega ao módulo AL-3202, depois de expirado o tempo de seleção. Este tempo deve ter sido especificado na configuração do módulo, entre 0,1 a 25,5 segundos (ver descrição da F-CBO.018).
- **Oc - Módulo ocupado** - O bit Oc permanece ligado durante a execução de um comando. O bit Oc é considerado erro se um comando for acionado durante a execução do anterior. Neste caso, o bit Er também é ligado. É normal o bit Oc aparecer ligado após um comando t/c, r/l ou SBO t/c.
- **Sa - Erro nas saídas** - Este bit indica que o módulo AL-3202 tem um defeito de hardware em uma ou mais saídas. Deve-se analisar as posições 4 e 5 da %TMXXXX para saber qual(is) saída(s) está (ão) com defeito.
- **Co - Comando inválido (latch)** - O octeto do módulo AL-3202 endereçado não está configurado de acordo com comando enviado (ver F-CBO.018 - Configuração).
- **Sc - Comando Operate/Cancel após configuração (latch)** - Este erro acontece se o comando operate/cancel for enviado ao AL-3202 após um comando de configuração. Isto pode acontecer se a entrada 1 da função F-CBO for mantida ligada durante a execução da sequência SELECT e OPERATE/CANCEL (ver descrição da F-CBO.018).
- **Fo - Fonte 24V desligada** - Este bit avisa que a fonte 24V que alimenta os módulo AL-3202 não está ativa, ou que o cabo que interliga os dois módulos do AL-3202 não está conectado. Este erro pode ser revertido após corrigir o problema e reconfigurar o módulo AL-3202, ligando as entradas 0, 1, e 2 da F-CBO (ver descrição da F-CBO.018).
- **Nc - Não configurado** - Indica que o módulo não recebeu configuração após ter sido trocado, energizado ou a UCP passado de "programação" para "execução". Pode acontecer após uma troca do módulo, caso a função F-CBO não esteja ativada (ver descrição da F-CBO.018).

Exemplos de aplicação

A seguir apresentam-se alguns exemplos de comandos de acionamento de pontos.

No lado esquerdo está colocada a representação dos valores escritos na tabela pelo supervisor, antes de serem processados pela função (posições 0 a 4). No lado direito, os valores monitorados pelo supervisor após o processamento da função.

- 1) Acionamento imediato do ponto biestável de saída 7 do octeto 1 do módulo alojado no endereço %R0032 do barramento de E/S.

Sup.--->CP			Sup.<---CP		
TMXXX			TMXXX		
0	1	- acionamento biestável fechar contato	0	0	
1	279	- endereço do ponto $(32+1) \times 8 + 7$	1	279	
2	0	- não utilizado	2	0	
3		- Estado da operação	3	0	
4		- Estado saídas oct 3/2	4	0	
5		- Estado saídas oct 1/0	5	0	
6		- uso interno da função	6		

- 2) Acionamento imediato por 1 segundo do ponto "rise/lower" de saída 2 do octeto 3 do módulo alojado no endereço %R0048 do barramento de E/S.

Sup.--->CP			Sup.<---CP		
TMXXX			TMXXX		
0	7	- acionamento "lower"	0	6	
1	410	- endereço do ponto $(48+3) \times 8 + 2$	1	410	
2	100	- tempo de acionamento 1 segundo	2	100	
3		- Estado da operação	3	0	
4		- Estado saídas oct 3/2	4	0	
5		- Estado saídas oct 1/0	5	0	
6		- uso interno da função	6		

- 3) Seleção do ponto "trip/close" de saída 5 do octeto 0 do módulo alojado no endereço %R0024 do barramento de E/S.

Sup.--->CP			Sup.<---CP		
TMXXX			TMXXX		
0	13	- comando seleção "trip"	0	12	
1	197	- endereço do ponto $(24+0) \times 8 + 5$	1	197	
2	0	- não utilizado	2	0	
3		- Estado da operação	3	0	
4		- Estado saídas oct 3/2	4	0	
5		- Estado saídas oct 1/0	5	0	
6		- uso interno da função	6		

4) Operação do ponto "trip/close" selecionado no exemplo anterior.

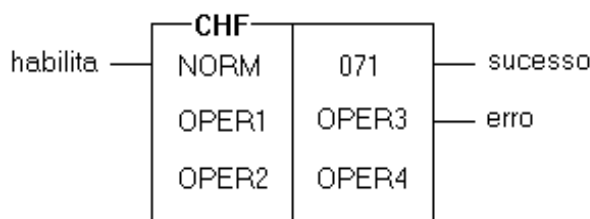
Sup.--->CP

TMXXX		
0	17	- comando operação "trip"
1	197	- endereço do ponto (24+0) X 8 + 5
2	0	- não utilizado
3		- Estado da operação
4		- Estado saídas oct 3/2
5		- Estado saídas oct 1/0
6		- uso interno da função

Sup.<---CP

TMXXX	
0	16
1	197
2	0
3	0
4	0
5	0
6	

F-NORM.071 - Função para Normalização



Introdução

A função F-NORM.071 normaliza operandos inteiros, implementando o seguinte algoritmo de normalização:

$$M(saída) = \frac{(M(entrada) - A) \cdot (C)}{B - A}$$

Onde A, B e C são constantes.

Programação

- OPER1 - Especifica o número de parâmetros que são passados para a função em OPER3. Este operando deverá ser obrigatoriamente uma constante memória com valor 6 (%KM+00006).
- OPER2 - Deve ser um operando do tipo constante memória com valor 0 (%KM+00000). Determina o número de parâmetros possíveis de serem programados na janela de edição de OPER4. Como esta função não necessita de nenhum parâmetro em OPER4, o valor de OPER2 é 0.
- OPER3 - Contém os parâmetros que são passados para a função, declarados através de uma janela visualizada no MasterTool quando a instrução CHF for editada. O número de parâmetros editáveis é especificado em OPER1, sendo 6 para esta chamada:
 - %KMXXXXXX – número de operandos;
 - %MXXXXXX – primeiro operando de entrada;
 - %MXXXXXX – primeiro operando de saída;
 - %KMXXXXXX – início da faixa de entrada (A);
 - %KMXXXXXX – fim da faixa de entrada (B);
 - %KMXXXXXX – fim da faixa de saída (C);

A partir da versão 1.10 do F-NORM.071, disponível para os CPs AL-2003, AL-2004 e Série Ponto, pode-se utilizar um sétimo parâmetro (D), que define o início da faixa de entrada. Quando forem declarados 6 parâmetros na CHF da F-NORM.071 versão 1.10, será admitido que o sétimo parâmetro (D) é igual a zero e executado o mesmo algoritmo de normalização.

O algoritmo de normalização passa a ser escrito como:

$$M(saída) = \frac{(M(entrada) - A) \cdot (C - D)}{B - A} + D$$

Depois de declarar o valor 7 (%KM+00007) para o OPER1, os parâmetros passados através de OPER3 são os seguintes:

- %KMXXXXXX – número de operandos;

- %MXXXXXX – primeiro operando de entrada;
- %MXXXXXX – primeiro operando de saída;
- %KXXXXXX – início da faixa de entrada (A);
- %KXXXXXX – fim da faixa de entrada (B);
- %KXXXXXX – fim da faixa de saída (C);
- %KXXXXXX – início da faixa de saída (D) (somente na versão 1.10).

Operação

A F-NORM.071 implementa o seguinte cálculo:

$$M[\text{saída}] = (M[\text{entrada}] - A) * C / (B - A)$$

sendo:

- M[entrada] - faixa de operandos inteiros de entrada
- M[saída] - faixa de operandos inteiros de saída
- A - offset para a entrada
- B - valor referência da entrada para normalizar
- C - valor normalizado da saída correspondente à B

A saída é a normalização da entrada e de modo que para um dado de entrada com o valor A a saída correspondente é 0, e para um valor de entrada B a saída

correspondente será C. Fora dessa faixa, o valor da saída será proporcional ao da entrada, conforme a fórmula dada.

A função trabalha com faixa de até 127 operandos (1 a 127)

Entradas e Saídas

Descrição das entradas da função:

- **habilita** - quando esta entrada está energizada a função é chamada, sendo analisados os parâmetros programados na instrução CHF. Caso os mesmos estejam incorretos, a saída erro da instrução é energizada, e as demais ficam desenergizadas. Se os parâmetros estiverem corretos, somente a saída sucesso é energizada.

Descrição das saídas da função:

- **sucesso** - indica que os parâmetros da chamada estão corretos e que a função foi corretamente executada. Vide observação.

- **erro** - é ligada caso ocorra erro nos parâmetros da chamada. Vide observação.

OBS: quando as duas saídas (sucesso e erro) estiverem energizadas é porque o intervalo de operandos M de entrada é o mesmo que o intervalo de operandos M de saída (nos parâmetros).

Utilização

Esta função pode ser utilizada nas UCPs AL-600, AL-2000/MSP, AL-2002, AL-2003, AL-2004, PL104, PL105, PL106, QK800, QK801, QK2000 e Série Ponto.

Exemplo

Seja a entrada um valor inteiro proveniente de uma instrução A/D, com variação de **0** a **4095**. Deseja-se normalizar a saída para **0** a **100**, correspondendo à valores de entrada entre **800** a **4000**. Teremos:

$$A = 800$$

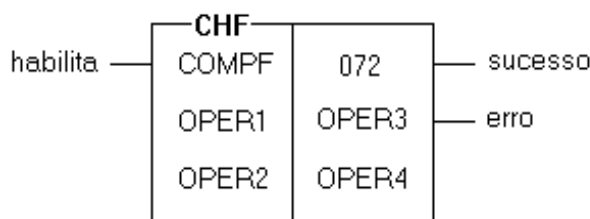
$$B = 4000$$

$$C = 100$$

Resultados:

Entrada	Saída
0	-25
800	0
2400	50
4000	100
4095	102

F-COMPF.072 - Função para Múltiplas Comparações



Introdução

A função F-COMPF.072 divide um operando em faixas especificadas, apresentando saída em forma binária, onde o bit ligado indica que o operando pertence à faixa respectiva.

Programação

As células da instrução CHF utilizada para a chamada são programadas do seguinte modo:

- **OPER1** - Especifica o número de parâmetros que são passados para a função em OPER3. Este operando deverá ser obrigatoriamente uma constante memória com valor 4 (%KM+00004).
- **OPER2** - Deve ser um operando do tipo constante memória com valor 0 (%KM+00000). Determina o número de parâmetros possíveis de serem programados na janela de edição de OPER4. Como esta função não necessita de nenhum parâmetro em OPER4, o valor de OPER2 é 0.
- **OPER3** - Contém os parâmetros que são passados para a função, declarados através de uma janela visualizada no MasterTool quando a instrução CHF for editada. O número de parâmetros editáveis é especificado em OPER1, sendo 4 para esta chamada:

- * **%KM+XXXX** - Número de operandos %MXXXX a examinar
- * **%MXXXX** - Operando de entrada inicial para a comparação
- * **%MXXXX** - Operando de saída inicial para os bits indicadores
- * **%TMXXXX** - Tabela que especifica até 16 faixas de valores para qualificar os operandos de entrada (ver formato a seguir)

Posição Tabela	Conteúdo
0	Reservada
1	Reservada
2	Início faixa 0
3	Fim da faixa 0
4-31	<continuam as definições de faixas>
32	Início faixa 15
33	Fim da faixa 15

Tabela 4-28 - Definição das faixas

A tabela deve ter tamanho mínimo de 4 posições (1 faixa). Para otimizar o tempo de execução da função, recomenda-se que a tabela seja definida com o tamanho exato para conter as definições de faixas necessárias.

- **OPER4** - Não utilizado.

Operação

O início e o fim de cada faixa de comparação são especificados como números inteiros.

O operando é considerado na faixa se cumprir:

$$(\text{início da faixa}) \leq \%MXXXX < (\text{fim da faixa})$$

A cada faixa na tabela %TMXXXX corresponde um bit no operando %MXXXX, sendo que o bit 0 do operando de saída corresponde a faixa 0 e assim sucessivamente. Os bits correspondentes às faixas não definidas são sempre 0. As faixas podem ser sobrepostas.

O número de operandos a processar é dado pelo primeiro parâmetro (%KM+XXXX), podendo ser definido de 1 a 127.

Entradas e Saídas

Descrição das entradas da função:

- **habilita** - quando esta entrada está energizada a função é chamada, sendo analisados os parâmetros programados na instrução CHF. Caso os mesmos estejam incorretos, a saída **erro** da instrução é energizada, e as demais ficam desenergizadas. Se os parâmetros estiverem corretos, somente a saída sucesso é energizada.

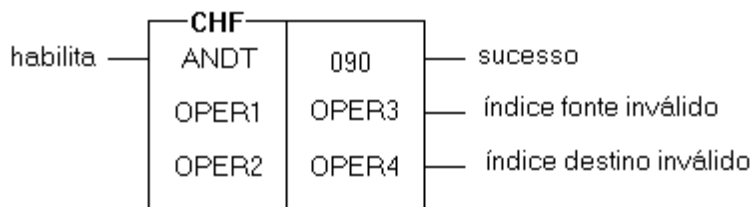
Descrição das saídas da função:

- **sucesso** - indica que os parâmetros da chamada estão corretos e que a função foi corretamente executada.
- **erro** - é ligada caso ocorra erro nos parâmetros da chamada

Utilização

Esta função pode ser utilizada nas UCPs AL-600, AL-2000/MSP, AL-2002, AL-2003, AL-2004, QK800, QK801 e QK2000.

F-ANDT.090, F-ORT.091 e F-XORT.092 - Funções de Operações Lógicas entre Operandos Tabela



Introdução

As funções F-ANDT.090, F-ORT.091 e F-XORT.092 permitem realizar operações lógicas AND (e), OR (ou) ou XOR (ou exclusivo), respectivamente, entre operandos simples (M ou D) e/ou tabelas (TM ou TD). Podem ser realizadas até 255 operações lógicas em uma única chamada da função. É necessário que os três operandos (fonte1, fonte2 e destino) sejam do mesmo tipo (memória ou decimal).

Programação

As células da instrução CHF utilizada para a chamada são programadas do seguinte modo:

- **OPER1** - Especifica o número de parâmetros que são passados para a função em OPER3. Este operando deverá ser obrigatoriamente uma constante memória com valor 4 (KM+00004).
- **OPER2** - Deve ser um operando do tipo constante memória com valor 0 (KM+00000). Determina o número de parâmetros possíveis de serem programados na janela de edição de OPER4. Como estas funções não necessitam de nenhum parâmetro em OPER4, o valor de OPER2 é 0.
- **OPER3** - Contém os parâmetros que são passados para a função, declarados através de uma janela visualizada no AL-3830 quando a instrução CHF for editada. O número de parâmetros editáveis é especificado em OPER1, sendo fixo em 4 para estes módulos:

—* **MXXXX, DXXXX, TMXXXX ou TDXXXX** - Operando simples ou tabela cujo valor será utilizado para realização da lógica (operando fonte 1). |

—* **MXXXX, DXXXX, TMXXXX ou TDXXXX** - Operando simples ou tabela cujo valor será utilizado para realização da lógica (operando fonte 2). |

—* **MXXXX, DXXXX, TMXXXX ou TDXXXX** - Operando simples ou tabela onde o valor resultante da lógica será armazenado (operando destino). |

—* **KMXXXX** - Número de operandos simples ou posições da tabela com as quais se fará a operação lógica. |

- **OPER4** - Não utilizado.

Entradas e Saídas

Descrição das entradas da função:

- **habilita** - quando esta entrada está energizada a função é chamada, sendo analisados os parâmetros programados na instrução CHF. Caso os mesmos estejam incorretos, são acionadas as saídas de índice inválido.

Descrição das saídas da função:

- **sucesso** - indica que a movimentação foi corretamente realizada.
- **índice fonte inválido** - indica que houve erro na especificação do operando fonte:
 - * o operando não está declarado no módulo C
 - * não existem posições suficientes para realizar a lógica
- **índice destino inválido** - indica que houve erro na especificação do operando destino:
 - * o operando não está declarado no módulo C
 - * não existem posições suficientes para realizar a lógica

Caso as duas saídas de índice inválido sejam acionadas simultaneamente, ocorreu algum dos seguintes erros:

- - o número de parâmetros programado em OPER1 é diferente de quatro
- - o tipo de um ou mais parâmetros em OPER4 não é válido
- - os parâmetros em OPER4 são de tipos diferentes (memória e decimal)
- - o número total de posições a serem transferidas é maior que 255

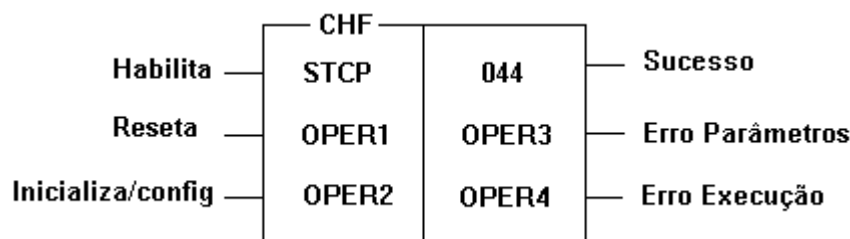
Utilização

Esta função pode ser utilizada nas UCPs AL-600, AL-2000/MSP, AL-2002/MSP, QK800, QK801 e AL-2000/MSP.

ATENÇÃO:

Estas funções permitem executar operações lógicas de um grande número de operandos em uma única varredura. Deve-se utilizá-las com cuidado para que o tempo máximo de ciclo do programa não seja excedido.

F-ETHDG.089 – Função de Diagnóstico de Ethernet



Introdução

O módulo F-ETHDG.089 permite que os controladores programáveis AL-2003 e AL-2004 tenham acesso aos diagnósticos e estatísticas de operação da interface para rede Ethernet AL-3405. O acesso aos diagnósticos da placa AL-3405 permite a avaliação do seu modo de operação, bem como da qualidade da comunicação com a rede Ethernet.

Descrição dos operandos:

- **OPER1** - especifica o número de parâmetros que são passados para a função em OPER3. Este operando deverá ser obrigatoriamente uma constante memória com valor 1 (%KM + 00001).
- **OPER2** - especifica o número de parâmetros que são passados para a função em OPER4. Como esta função não necessita de nenhum parâmetro em OPER4, o valor deste operando deverá ser obrigatoriamente uma constante memória com valor 0 (%KM + 00000).
- **OPER3** - Contém os parâmetros que são passados para a função, declarados quando a instrução CHF for editada. O número de parâmetros editáveis é especificado em OPER1, sendo fixo em 1 (um) para este módulo:

%Mxxxx ou %TMxxx - Operando memória ou tabela memória que recebe os valores dos diagnósticos, estatísticas de erros, parâmetros e configuração da interface para rede Ethernet. Se for utilizado um operando memória %M, devem estar declarados no mínimo 90 operandos a partir do declarado (incluindo o mesmo) para que a função seja executada corretamente. Caso seja um operando tabela memória %TM, este deve possuir pelo menos 90 posições declaradas.

- **OPER4** - Contém os parâmetros que são recebidos da função, declarados quando a instrução CHF for editada. O número de parâmetros editáveis é especificado em OPER2, sendo fixo em 0 (nenhum) para este módulo.

Descrição das entradas:

- **HABILITA** - Quando esta entrada está energizada a função é chamada, sendo analisado o parâmetro programado na instrução CHF. Caso o mesmo esteja incorretos, a saída de erro de parâmetros da instrução é acionada. Se estiver correto, as informações de diagnóstico são copiadas, sendo acionada a saída sucesso.
- **INICIALIZA** - Quando energizada em conjunto com a entrada de habilitação da função, zera os valores das estatísticas.
- **RESETA** - Quando energizada em conjunto com a entrada de habilitação da função, reseta fisicamente a interface AL-3405. Esta entrada tem prioridade sobre a entrada de inicialização e configuração da função.

Descrição das saídas:

- **SUCESSO** - É energizada quando a função foi corretamente executada.

- **ERRO PARÂMETROS** - É energizada caso ocorra erro na especificação dos operandos declarados como primeiro parâmetro de OPER3, ou tentativa de acesso a operandos não declarados na UCP.
- **ERRO EXECUÇÃO** - É energizada caso ocorra erro no pedido de configuração ou de diagnóstico para a interface AL-3405, motivada por um dos seguintes fatores:

A interface AL-3405 não se encontra no barramento: a saída permanecerá acionada para sempre;

A interface AL-3405 está fisicamente desconfigurada: a saída permanecerá acionada até que a UCP seja configurada adequadamente;

A interface AL-3405 está em processo de configuração física: a saída permanecerá acionada até que a interface encerre a sua configuração;

A interface AL-3405 está em processo de power-up ou de reset: a saída permanecerá acionada até que seja encerrada sua configuração;

A interface AL-3405 está temporariamente ocupada: pode vir a ocorrer periodicamente, principalmente quando mensagens Ethernet estão sendo recebidas ou transmitidas, e não deve ser interpretado como um erro. É provável que na próxima chamada esta saída não esteja mais acionada.

Observações importantes:

A função F-ETHER.094 é executada em aproximadamente 2 ms. No caso de acionamento da entrada de reset, a interface AL-3405 levará em torno de 3 segundos para retornar ao modo normal de execução e estar apta a comunicar através da rede Ethernet novamente.

Descrição dos operandos de diagnóstico:

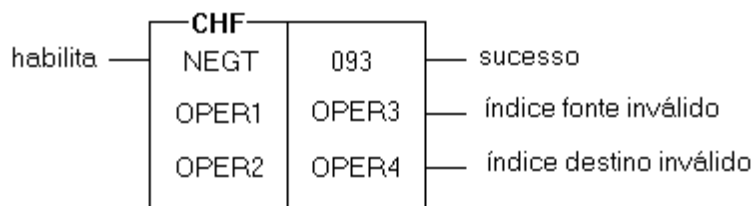
Operando	Identificação	Descrição
00	Estado da interface AL-3405 diagnosticada pela UCP	<p>Bit 0: 0 ou 1 (normal ou ocupado) - atualizado internamente a cada ciclo do CP ou a cada chamada da instrução LAI; pode retornar ocupado esporadicamente quando do uso de instruções consecutivas do tipo LAI; estado ocupado por vários ciclos consecutivos pode significar interface inoperante;</p> <p>Bit 1: quando em 1 indica se a interface AL-3405 já esteve ocupado pelo menos uma vez, desde a sua última inicialização/configuração;</p> <p>Bit 2 ao 15: não utilizados (sempre 0).</p>
01	Não utilizado	Reservado para o futuro
02	Endereço IP da estação	(XXX.xxx.xxx.xxx)
03	Endereço IP da estação	(xxx.XXX.xxx.xxx)
04	Endereço IP da estação	(xxx.xxx.XXX.xxx)
05	Endereço IP da estação	(xxx.xxx.xxx.XXX)
06	Endereço do gateway default	(XXX.xxx.xxx.xxx)
07	Endereço do gateway default	(xxx.XXX.xxx.xxx)
08	Endereço do gateway default	(xxx.xxx.XXX.xxx)
09	Endereço do gateway default	(xxx.xxx.xxx.XXX)
10	Máscara de sub-rede	(XXX.xxx.xxx.xxx)
11	Máscara de sub-rede	(xxx.XXX.xxx.xxx)
12	Máscara de sub-rede	(xxx.xxx.XXX.xxx)
13	Máscara de sub-rede	(xxx.xxx.xxx.XXX)
14 a 17	Não utilizados	Reservado para uso futuro
18	Quantidade de pacotes de controle Ethernet transmitidos sem erro	<p>0 – 32767.</p> <p>Inclui as mensagens Ethernet sem dados ALNET II, como as do tipo ARP, ICMP (PING) e algumas do TCP.</p>
19	Quantidade de pacotes ALNET II transmitidos sem erro	<p>0 – 32767.</p> <p>Inclui as mensagens Ethernet com dados ALNET II (todas são do tipo TCP).</p>
20	Quantidade de erros por colisão	<p>0 – 32767.</p> <p>Colisão do frame em transmissão com outro frame sendo transmitido por outro equipamento. Pode ocorrer em função de altos índices de tráfego na rede Ethernet. É automaticamente resolvido pelo controlador LAN da interface de rede.</p>
21	Quantidade de erros por underrun	<p>0 – 32767.</p> <p>A velocidade de leitura do frame a ser transmitido da memória da UCP é inferior a taxa de transmissão. Não deve ocorrer.</p>
22	Quantidade de erros por perda de portadora	<p>0 – 32767.</p> <p>Perda do sinal de portadora ("carrier sense") durante a transmissão do frame. Não deve ocorrer.</p>
23	Quantidade de erros por perda de CTS	<p>0 – 32767.</p> <p>Perda do sinal de CTS ("clear to send") durante a transmissão do frame. Não deve ocorrer.</p>
24	Não utilizado	Reservado para uso futuro.
25	Identificador do LED de erro	<p>Código que indica o motivo do acendimento do o LED ER do painel da interface AL-3405. O código de erro permanece até a ocorrência de um novo erro, ou até a reinicialização da interface ou dos diagnósticos. Está normalmente com valor 0.</p> <p>Valor 0: nenhum erro;</p> <p>Valor 1: erro de transmissão de pacote;</p> <p>Valor 2: erro de retransmissão do pacote;</p> <p>Valor 3: erro de timeout de pacote;</p> <p>Valores 4 ao 32767: reservado para uso futuro.</p>

26	Quantidade de erros por falta de buffers de transmissão	0 – 32767. Houve a tentativa de alocar um buffer Ethernet para transmissão de um frame, mas não havia nenhum disponível. Pode ocorrer esporadicamente, sem consequências negativas, quando da ocorrência simultânea de várias mensagens BROADCAST.
27 a 33	Não utilizados	reservado para uso futuro.
34	Quantidade de pacotes de controle Ethernet recebidos sem erro	0 – 32767. Inclui as mensagens Ethernet sem dados ALNET II, como as do tipo ARP (protocolo para resolução de endereços), ICMP (protocolo de mensagens de controle da internet) e algumas do TCP (protocolo de controle de transporte).
35	Quantidade de pacotes ALNET II recebidos sem erro	0 – 32767. Inclui as mensagens Ethernet com dados ALNET II (todas são do tipo TCP).
36	Quantidade de erros por colisão	0 – 32767. Colisão de frames durante a recepção. Pode ocorrer em função de altos índices de tráfego na rede Ethernet.
37	Quantidade de erros por overrun	0 – 32767. A velocidade de transferência do frame recebido para a memória da UCP é inferior à taxa de recepção. Deste modo, ocorre overrun quando um frame é recebido antes que o último tenha sido transferido para a memória da UCP. Este erro pode ocorrer com frequência, em função da quantidade de mensagens BROADCAST ou mensagens endereçadas à UCP recebidas. As retentativas de comunicação efetuadas pela camada TCP da rede Ethernet se encarregam de enviar nova requisição, de modo que esta falha não é percebida pelas aplicações.
38	Quantidade de erros por CRC inválido	0 – 32767. Frame recebido com erro na informação de CRC, informação esta que garante a integridade da mensagem. Não deve ocorrer.
39	Quantidade de erros por alinhamento	0 – 32767. Frame recebido com erro de alinhamento (campos inconsistentes do frame). Não deve ocorrer.
40	Quantidade de erros por tamanho de pacote inválido	0 – 32767. Recepção de um frame com tamanho inválido. Não deve ocorrer.
41	Quantidade de erros por timeout de serviço	0 – 32767. Um serviço solicitado pela UCP (LTR/ECR) não pode ser realizado em tempo hábil. Pode ocorrer se o timeout foi mal configurado pelo usuário.
42	Quantidade de erros por falta de buffers de recepção	0 – 32767. Houve a tentativa de alocar um buffer Ethernet para recepção de um novo frame, mas não havia nenhum disponível. Pode ocorrer esporadicamente, sem maiores consequências, quando da ocorrência simultânea de várias mensagens BROADCAST ou endereçadas à UCP.
43 a 49	Não utilizados	Reservado para uso futuro.
50	Quantidade de buffers Ethernet disponíveis (transmissão e recepção)	0 – 15.
51	Quantidade de buffers de transmissão de interface com a aplicação disponíveis	0 – 16.
52	Quantidade de buffers de recepção de interface com a aplicação disponíveis	0 – 16.
53	Quantidade de liberações de buffers de transmissão por existência de outro com o mesmo endereço destino na fila de transmissão	0 – 32767. Não deve ocorrer.

54	Quantidade de liberações de buffers de transmissão por inexistência de conexão ativa (cliente) estabelecida com o endereço destino	0 – 32767. Não deve ocorrer.
55	Quantidade de liberações de buffers de transmissão por inexistência de conexão passiva (servidor) estabelecida com o endereço destino	0 – 32767. Não deve ocorrer.
56	Quantidade de liberações de buffers de transmissão devido a conexão com o endereço destino estar fechada	0 – 32767. Não deve ocorrer.
57	Quantidade de liberações de buffers de transmissão devido a erro na tentativa de abertura de conexão com o endereço destino	0 – 32767. Não deve ocorrer.
58	Quantidade de conexões estabelecidas no modo cliente	0 – 127
59	Quantidade de conexões estabelecidas no modo servidor	0 – 127
60	Quantidade de conexões estabelecidas no nível Ethernet, incluindo as conexões em modo cliente, modo servidor e a conexão LISTEN (de espera):	0 – 128
61	Quantidade de conexões Ethernet fechadas por inatividade	0 – 32767. Conexões abertas mas inativas (sem troca de mensagens) por mais de 10 minutos, são abortadas automaticamente.
62	Quantidade de buffers Ethernet liberados por tempo de existência	0 – 32767. Proteção interna: buffer que permanecem na fila sem serem tratados por mais de 20 segundos, são descartados automaticamente.
63	Bits de diagnósticos gerais	Bit 0: indica a inexistência/existência (0/1) da conexão LISTEN (a perda da conexão LISTEN impede a interface de abrir novas conexões); Bit 1 ao 15: não utilizado.
64 a 80	Não utilizados	Reservado para uso futuro.
81	Versão de software executivo (BIOS) da interface AL-3405.	
82	Configuração do modo de operação da interface para rede Ethernet AL-3405	Bit 0: endereçamento das mensagens controlados pela UCP; Bit 1: tratamento de mensagens com múltiplos pacotes ALNET II; Bit 2: transmissão de mensagens com o tamanho real dos pacotes ALNET II; Bit 3: velocidade dobrada quanto ao tratamento de mensagens Ethernet; Bit 4: modo de operação turbo quanto ao tratamento de mensagens Ethernet; Bit 5 ao 15: não utilizados (sempre zerados). Estes campos, com exceção do bit 0, são habilitados somente com o módulo F AL-2704. O bit 0 é colocado em somente nas seguintes versões de módulos: <ul style="list-style-type: none"> AL-2003 versão 1.42 ou posterior; AL-2004 versão 1.04 ou posterior; AL-3405 versão 1.66 ou posterior.

83	Configuração dos parâmetros da camada TCP (dígitos decimais)	Unidade [0 – 9]: timeout de retransmissão inicial (x 100 ms) utilizado pela camada TCP; Dezena [0 – 5]: retentativas de transmissão da camada TCP; Demais dígitos : não utilizados (sempre zerados). Deve coincidir com a segunda constante de configuração declarada em OPER3 da CHF.
84	Não utilizados	Reservado para uso futuro.

F-NEGT.093 - Função para Negação Lógica de Operandos Tabela



Introdução

A função F-NEGT.093 realiza a negação lógica de operandos simples (M ou D) ou tabelas (TM ou TD). Podem ser negadas até 255 posições em uma única chamada a função. O resultado da alteração pode ser armazenado no próprio operando, substituindo o valor original, ou em um outro operando, desde que ele seja do mesmo tipo do primeiro (memória ou decimal).

Programação

As células da instrução CHF utilizada para a chamada são programadas do seguinte modo:

- **OPER1** - Especifica o número de parâmetros que são passados para a função em OPER3. Este operando deverá ser obrigatoriamente uma constante memória com valor 3 (KM+00003).
- **OPER2** - Deve ser um operando do tipo constante memória com valor 0 (KM+00000). Determina o número de parâmetros possíveis de serem programados na janela de edição de OPER4. Como esta função não necessita de nenhum parâmetro em OPER4, o valor de OPER2 é 0.
- **OPER3** - Contém os parâmetros que são passados para a função, declarados através de uma janela visualizada no AL-3830 quando a instrução CHF for editada. O número de parâmetros editáveis é especificado em OPER1, sendo fixo em 3 para este módulo:

—* **MXXXX, DXXXX, TMXXXX ou TDXXXX** - Operando simples ou tabela cujos valores serão negados (operando fonte).

—* **MXXXX, DXXXX, TMXXXX ou TDXXXX** - Operando simples ou tabela onde os valores negados serão armazenados (operando destino).

—* **KMXXXX** - Número de operandos simples ou posições da tabela a serem negadas. Deve ser menor ou igual a 255

- **OPER4** - Não utilizado.

Entradas e Saídas

Descrição das entradas da função:

- **habilita** - quando esta entrada está energizada a função é chamada, sendo analisados os parâmetros programados na instrução CHF. Caso os mesmos estejam incorretos, são acionadas as saídas de índice inválido.

Descrição das saídas da função:

- **sucesso** - indica que a movimentação foi corretamente realizada.
- **índice fonte inválido** - indica que houve erro na especificação do operando fonte:
 - * o operando não está declarado no módulo C
 - * não existem posições suficientes para realizar a lógica
- **índice destino inválido** - indica que houve erro na especificação do operando destino:

- * o operando não está declarado no módulo C
- * não existem posições suficientes para realizar a lógica

Caso as duas saídas de índice inválido sejam acionadas simultaneamente, ocorreu algum dos seguintes erros:

- o número de parâmetros programado em OPER1 é diferente de três
- o tipo de um dos parâmetros em OPER3 não é válido
- o tipo do operando destino é diferente do operando fonte
- o número total de posições a serem transferidas é maior que 255

Utilização

Esta função pode ser utilizada nas UCPs AL-600, AL-2000/MSP, AL-2002/MSP, QK800, QK801 e AL-2000/MSP.

ATENÇÃO:

Esta função permite a negação de um grande número de operandos em uma única varredura. Deve-se utilizá-la com cuidado para que o tempo máximo de ciclo do programa não seja excedido.

Apêndice A

Tempos de Execução das Instruções

Os tempos de execução apresentado a seguir são válidos para as UCPs AL-600, AL-600/4, AL-600/8, AL-600/16, AL-2000/MSP, AL-2002/MSP, QK600, QK800, QK801, QK2000/MSP, PL101, PL102 e PL103.

As UCPs AL-3003 e AL-3004 possuem velocidade de processamento 25% menor. Para obter o tempo de execução deve-se multiplicar o tempo apresentado por 1,25.

A UCP AL-3003/V2 possui velocidade de processamento 12,5% maior. Para obter o tempo de execução deve-se multiplicar o tempo apresentado por 0.875.

Descrição dos Tempos de Execução

Os tempos de execução das instruções são descritos em uma tabela para cada instrução possuindo os seguintes itens:

- **Cabeçalho** - nome da instrução ou instruções para as quais foram medidos os tempos
- **Situação** - descreve a situação em que foi medido o tempo.
Ex.: habilitada, desabilitada, acesso direto, acesso indireto,
- **Tempo de Execução** - descreve tempo de execução medido para cada situação da instrução.
Ex.: 2,4 μ s

Relés

RNA - Contato Normalmente Aberto

Situação	Tempo de Execução
Tempo Máximo (%E0000.0 a %E0015.7)	2,4 μ s
Tempo Médio (%E0016.0 a %E0063.7) (%A0000.0 a %A0095.7)	4,8 μ s
Tempo Mínimo (%M0000.0 a %M0127.F)	8,0 μ s

RNF - Contato Normalmente Fechado

Situação	Tempo de Execução
Tempo Máximo (%E0000.0 a %E0015.7)	2,4 μ s
Tempo Médio (%E0016.0 a %E0063.7) (%A0000.0 a %A0095.7)	4,8 μ s
Tempo Mínimo (%M0000.0 a %M0127.F)	8,0 μ s

BOB - Bobina Simples

Situação	Tempo de Execução
Tempo mínimo (%S0000.0 a %S0015.7)	2,4 μ s
Tempo médio (%S0001.5 a %S0063.7) (%A0000.0 a %A0095.7)	6,4 μ s
Tempo máximo (%M0000.0 a %M0127.F)	9,6 μ s

BBL - Bobina Liga

Situação	Tempo de Execução
Tempo mínimo (%S0000.0 a %S0015.7)	4,0 μ s
Tempo médio (%S0016.0 a %S0063.7) (%A0000.0 a %A0095.7)	8,0 μ s
Tempo máximo (%M0000.0 a %M0127)	11,2 μ s

BBD - Bobina Desliga

Situação	Tempo de Execução
Tempo mínimo (%S0000.0 a %S0015.7)	4,0 μ s
Tempo médio (%S0016.0 a %S0063.7) (%A0000.0 a %A0095.7)	8,0 μ s
Tempo máximo (%M0000.0 a %M0127)	11,2 μ s

SLT - Bobina de Salto

Situação	Tempo de Execução
Desabilitada	26 μ s
Habilitada	32 μ s

PLS - Relé de Pulso

Situação	Tempo de Execução
Desabilitada, habilitada	51 μ s

RM - Relé Mestre

Situação	Tempo de Execução
Desabilitada, habilitada	26 μ s

FRM - Fim de Relé Mestre

Situação	Tempo de Execução
Desabilitada, habilitada	15 μ s

Movimentadores

MOV - Movimentação de Operandos Simples

Situação	Tempo de Execução
Desabilitada	35 μ s
Tempo médio (acesso direto)	88 μ s
Tempo máximo (acesso indireto)	128 μ s

MOP - Movimentação de Partes de Operandos

Situação	Tempo de Execução
Desabilitada	35 μ s
Habilitada	120 μ s

MOB - Movimentação de Blocos de Operandos

Situação	Tempo de Execução
Desabilitada	42 μ s
Movimentação de 8 operandos %M / posições %TM por varredura	365 μ s
Movimentação de 128 operandos %M / posições %TM por varredura	2400 μ s
Movimentação de 255 operandos %M / posições %TM por varredura	4600 μ s
Movimentação de 8 operandos %D / posições %TD por varredura	480 μ s
Movimentação de 128 operandos %D / posições %TD por varredura	4100 μ s
Movimentação de 255 operandos %D / posições %TD por varredura	7900 μ s

MOT - Movimentação de Tabelas

Situação	Tempo de Execução
Desabilitada	38 μ s
Tempo médio (acesso direto)	160 μ s
Tempo máximo (acesso indireto)	230 μ s

MES - Movimentação de Entradas ou Saídas

Situação	Tempo de Execução
Desabilitada	32 μ s
Movimentação de 1 operando %M acesso direto	150 μ s
Movimentação de 8 operandos %M acesso direto	470 μ s
Movimentação de 8 operandos %M acesso indireto	4990 μ s

AES - Atualização de Entradas ou Saídas

Situação	Tempo de Execução
Desabilitada	32 μ s
Atualização de 1 operandos %E ou %S	150 μ s
Atualização de 8 operandos %E ou %S	242 μ s

CES - Conversão de Entradas ou Saídas

Situação	Tempo de Execução
Desabilitada	32 μ s
Conversão de leitura com acesso direto	260 μ s
Conversão de escrita com acesso direto	245 μ s
Conversão de leitura com acesso indireto	270 μ s
Conversão de escrita com acesso indireto	260 μ s

CAB - Carrega Bloco de Operandos

Situação	Tempo de Execução
Desabilitada	52 μ s
Carga de 8 operandos %A acesso direto	145 μ s
Carga de 8 operandos %M acesso direto	220 μ s
Carga de 8 operandos %M acesso indireto	285 μ s

Aritméticas

SOM - Adição

Situação	Tempo de Execução
Desabilitada	35 μ s
Tempo médio (operandos %M)	90 μ s
Tempo máximo (operandos %D)	128 μ s

SUB - Subtração

Situação	Tempo de Execução
Desabilitada	35 μ s
Tempo médio (operandos %M)	110 μ s
Tempo máximo (operandos %D)	170 μ s

MUL - Multiplicação

Situação	Tempo de Execução
Desabilitada	35 μ s
Habilitada sem estouro	130 μ s
Habilitada com estouro	120 μ s

DIV - Divisão

Situação	Tempo de Execução
Desabilitada	46 μ s
Valor menor que 128 no dividendo	140 μ s
Valor maior que 128 no dividendo e menor que 128 no divisor	258 μ s
Valores maiores que 128 no dividendo e no divisor	460 μ s

AND - E Binário entre Operandos

Situação	Tempo de Execução
Desabilitada	35 μ s
Tempo médio (operandos %M)	92 μ s
Tempo máximo (operandos %D)	110 μ s

OR - Ou Binário entre Operandos

Situação	Tempo de Execução
Desabilitada	35 μ s
Tempo médio (operandos %M)	92 μ s
Tempo máximo (operandos %D)	110 μ s

XOR - Ou Exclusivo Binário entre Operandos

Situação	Tempo de Execução
Desabilitada	35 μ s
Tempo médio (operandos %M)	92 μ s
Tempo máximo (operandos %D)	110 μ s

CAR - Carrega Operandos

Situação	Tempo de Execução
Desabilitada	34 μ s
Tempo médio (acessos diretos)	75 μ s
Tempo máximo (acessos indiretos)	100 μ s

= - Igual

Situação	Tempo de Execução
Desabilitada	34 μ s
Tempo médio (acessos diretos)	80 μ s
Tempo máximo (acessos indiretos)	100 μ s

< - Menor

Situação	Tempo de Execução
Desabilitada	34 μ s
Tempo médio (acessos diretos)	95 μ s
Tempo máximo (acessos indiretos)	130 μ s

> - Maior

Situação	Tempo de Execução
Desabilitada	34 μ s
Tempo médio (acessos diretos)	95 μ s
Tempo máximo (acessos indiretos)	130 μ s

Contadores

CON - Contador Simples

Situação	Tempo de Execução
Desabilitada	115 μ s
Tempo médio (acessos diretos)	120 μ s
Tempo máximo (acessos indiretos)	130 μ s

COB - Contador Bidirecional

Situação	Tempo de Execução
Desabilitada	170 μ s
Tempo médio (acessos diretos)	180 μ s
Tempo máximo (acessos indiretos)	230 μ s

TEE - Temporizador na Energização

Situação	Tempo de Execução
Desabilitada	85 μ s
Tempo médio (acessos diretos)	90 μ s
Tempo máximo (acessos indiretos)	110 μ s

TED - Temporizador na Desenergização

Situação	Tempo de Execução
Desabilitada	85 μ s
Tempo médio (acessos diretos)	90 μ s
Tempo máximo (acessos indiretos)	110 μ s

Conversores

B/D - Conversão Binário-Decimal

Situação	Tempo de Execução
Desabilitada	34 μ s
Tempo médio (acessos diretos)	115 μ s
Tempo máximo (acessos indiretos)	155 μ s

D/B - Conversão Decimal-Binário

Situação	Tempo de Execução
Desabilitada	34 μ s

Tempo médio (acessos diretos)	135 μ s
Tempo máximo (acessos indiretos)	170 μ s

A/D - Conversão Analógico-Digital

Situação	Tempo de Execução
Desabilitada	34 μ s
Conversão de 1 canal AL-1103	315 μ s
Conversão de 8 canais AL-1103	1570 μ s
Conversão de 1 canal AL-1116 ou AL-1119	350 μ s
Conversão de 8 canais AL-1116 ou AL-1119	2140 μ s

D/A - Conversão Digital-Analógico

Situação	Tempo de Execução
Desabilitada	34 μ s
Conversão de 1 canal AL-1203	210 μ s
Conversão de 4 canais AL-1203	550 μ s

Geral

LDI - Liga ou Desliga Indexado

Situação	Tempo de Execução
Desabilitada	40 μ s
Habilitada	65 μ s

TEI - Teste de Estado Indexado

Situação	Tempo de Execução
Desabilitada	40 μ s
Habilitada	85 μ s

SEQ - Seqüenciador

Situação	Tempo de Execução
Desabilitada	40 μ s
Habilitada (modo AL-1000)	95 μ s

CHP - Chama Módulo Procedimento

Situação	Tempo de Execução
----------	-------------------

Desabilitada	35 μ s
Habilitada	138 μ s

CHF - Chama Módulo Função

Situação	Tempo de Execução
Desabilitada	45 μ s
Chamada de módulo em linguagem de máquina	80 μ s
Chamada de módulo em linguagem de relés sem passagem de parâmetros	240 μ s
Chamada de módulo em linguagem de relés com a passagem de 4 parâmetros	550 μ s
Chamada de módulo em linguagem de relés com a passagem de 8 parâmetros	710 μ s

ECR - Escrita de Operandos em outro CP

Situação	Tempo de Execução
Desabilitada	150 μ s
Habilitada com outra ECR ou LTR comunicando (inativa)	150 μ s
Primeiro ciclo do processamento de uma comunicação com bloco de 8 operandos	900 μ s
Primeiro ciclo do processamento de uma comunicação com bloco de 110 operandos	2370 μ s
Habilitada esperando resposta	150 μ s

LTR - Leitura de Operandos de outro CP

Situação	Tempo de Execução
Desabilitada	150 μ s
Habilitada com outra ECR ou LTR comunicando (inativa)	150 μ s
Primeiro ciclo do processamento de uma comunicação com qualquer número de operandos	790 μ s
Habilitada esperando resposta	150 μ s

LAI - Libera Atualização de Imagens de Operandos

Situação	Tempo de Execução
Desabilitada	32 μ s
Habilitada inativa	150 μ s
Tempo máximo de processamento de requisições ou respostas a comandos com dados	2100 μ s
Tempo máximo de processamento de respostas a comandos sem dados, somente confirmações	450 μ s

Apêndice B

Tempos de Execução dos Módulos Função

Os tempos de execução apresentado a seguir são válidos para as UCPs AL-600, AL-600/4, AL-600/8, AL-600/16, AL-2000/MSP, AL-2002/MSP, QK800, QK801 e QK2000/MSP.

As UCPs AL-3003 e AL-3004 possuem velocidade de processamento 25% menor. Para obter o tempo de execução deve-se multiplicar o tempo apresentado por 1,25.

A UCP AL-3003/V2 possui velocidade de processamento 12,5% maior. Para obter o tempo de execução deve-se multiplicar o tempo apresentado por 0.875.

Descrição dos Tempos de Execução

Os tempos de execução das instruções são descritos em uma tabela para cada instrução ou grupo de instruções que possuam os mesmos tempos possuindo os seguintes itens:

- **Cabeçalho** - nome do módulo função para o qual foram medidos os tempos
- **Situação** - descreve a situação em que foi medido o tempo.

Ex.: habilitada, desabilitada, acesso direto, acesso indireto, etc.

- **ENT0, ENT1 e ENT2** - descrevem os valores das entradas.

—* x - valor não é relevante

—* 0 - entrada desenergizada

—* 1 - entrada energizada

- **Tempo de Execução** - descreve tempo de execução medido para cada situação da instrução.

Ex.: 2,4 μ s

F-RELOG.000

Situação	ENT0	ENT1	ENT2	Tempo de Execução
Desabilitado	0	x	x	50 μ s
Leitura de pulso	1	0	0	250 μ s
Leitura de tempo	1	1	0	940 μ s
Acerto de tempo	1	0	1	1050 μ s
Leitura e acerto de tempo	1	1	1	1050 μ s

- **ENT0** = entrada habilita
- **ENT1** = entrada lê relógio
- **ENT2** = entrada acerta relógio

F-LEDS.001

Situação	ENT0	ENT1	ENT2	Tempo de Execução
Desabilitado	0	x	x	50 μ s
Teste de LEDs	1	0	1	2100 μ s
Inicialização	1	1	0	55 μ s
Transferência de 8 octetos	1	0	0	925 μ s
Transferência de 32 octetos	1	0	0	2360 μ s

- **ENT0** = entrada habilita
- **ENT1** = entrada inicializa
- **ENT2** = entrada teste de leds

F-PT100.002

Situação	ENT0	Tempo de Execução
Desabilitado	0	50 μ s
Conversão de 1 canal sem linearização	1	425 μ s
Conversão de 1 canal com linearização	1	430 μ s
Conversão de 4 canais sem linearização	1	405 μ s
Conversão de 4 canais com linearização	1	420 μ s

- **ENT0** = entrada habilita

F-TERMO.003

Situação	ENT0	Tempo de Execução
Desabilitado	0	50 μ s
Conversão de 1 canal sem lin. em graus	1	260 μ s
Conversão de 1 canal com lin. normal.	1	260 μ s
Conversão de 4 canais sem lin. em graus	1	260 μ s
Conversão de 4 canais com lin. normal.	1	260 μ s

- **ENT0** = entrada habilita

F-CONTR.004

Situação	ENT0	Tempo de Execução
Desabilitado	0	50 μ s
Ativa relés de comparação	1	500 μ s
Inibição da contagem	1	756 μ s
Escrita contador	1	785 μ s
Leitura contador	1	940 μ s

- **ENT0** = entrada
- **ENT1** = entrada
- **ENT2** = entrada

F-CONT.005

Situação	ENT0	ENT1	ENT2	Tempo de Execução
Desabilitado	0	x	x	50 μ s
Leitura contador	1	0	0	360 μ s
Zera contador	1	1	0	295 μ s
Carrega contador	1	0	1	285 μ s

- **ENT0** = entrada habilita
- **ENT1** = entrada zera
- **ENT2** = entrada carrega

F-ANLOG.006

Situação	ENT0	Tempo de Execução
Desabilitado	0	50 μ s
Funcionamento como A/D	1	555 μ s
Funcionamento como D/A	1	280 μ s

- **ENT0** = entrada habilita

F-EVENT.017

Situação	ENT0	ENT1	ENT2	Tempo de Execução
Desabilitado	0	x	x	50 μ s
Configuração	1	1	x	1100 μ s
Eventos	1	0	1	1000 μ s + 500 ms por evento

- **ENT0** = entrada habilita
- **ENT1** = entrada lê evento/configura
- **ENT2** = entrada leitura/escrita

F-ALNET2.032

Situação	ENT0	ENT1	Tempo de Execução
Desabilitado	0	x	50 μ s
Leitura de Valores	1	0	640 μ s
Inicialização dos valores	1	1	790 μ s

- **ENT0** = entrada habilita
- **ENT1** = entrada inicializa

F-PID.033

Situação	ENT0	ENT1	ENT2	Tempo de Execução
Desabilitado	0	x	x	50 μ s
Cálculo dos fatores PID com modo automático	1	0	x	1600 μ s

Ação integral inibida com modo automático	1	0	x	1330 μ s
Ação derivativa inibida com modo automático	1	0	x	1240 μ s
Ajustes I e D inibidas com modo automático	1	0	x	950 μ s
Cálculo dos fatores PID com modo manual	1	1	x	1160 μ s
Ação integral inibida com modo manual	1	1	x	470 μ s
Ação derivativa inibida com modo manual	1	1	x	780 μ s
Ajustes I e D inibidas com modo manual	1	1	x	470 μ s

- **ENT0** = entrada habilita
- **ENT1** = entrada modo automático/manual
- **ENT2** = entrada ação direta/reversa

F-RAIZN.034

Situação	ENT0	ENT1	Tempo de Execução
Desabilitado	0	x	50 μ s
Raiz 0 a 127 sem normalização	1	0	330 μ s
Raiz 128 a 32767 sem normalização	1	0	440 μ s
Raiz 0 a 127 com normalização	1	1	380 μ s
Raiz 128 a 32767 com normalização	1	1	490 μ s

- **ENT0** = entrada habilita
- **ENT1** = entrada normaliza

F-ARQ2.035 a F-ARQ31.042

Situação	ENT0	ENT1	Tempo de Execução
Desabilitado	0	x	50 µs
Leitura de valor no arquivo	1	0	520 µs
Escrita de valor no arquivo	1	1	430 µs
Tentativa de acesso inválido	1	x	190 µs

- **ENT0** = entrada habilita
- **ENT1** = entrada lê/escreve

F-MOBT.043

Situação	ENT0	Tempo de Execução
Desabilitado	0	50 µs
Movimentação de 8 operandos %M/posições %TM	1	595 µs
Movimentação de 128 operandos %M/posições %TM	1	2520 µs
Movimentação de 255 operandos %M/posições %TM	1	4620 µs
Movimentação de 8 operandos %D/posições %TD	1	720 µs
Movimentação de 128 operandos %D/posições %TD	1	4410 µs
Movimentação de 255 operandos %D/posições %TD	1	8360 µs

- **ENT0** = entrada habilita

F-STDMOD.045

Situação	ENT0	ENT1	ENT2	Tempo de Execução
Desabilitado	0	x	x	50 µs
Leitura do estado dos octetos de E/S do barramento	1	0	0	650 µs
Leitura do diretório dos módulos	1	0	1	2350 µs
Leitura do estado dos módulos	1	1	0	2350 µs
Leitura do diretório e estado dos módulos	1	1	1	4060 µs

- **ENT0** = entrada habilita
- **ENT1** = entrada estado módulos
- **ENT2** = entrada diretório módulos

F-RELG.048

Situação	ENT0	ENT1	Tempo de Execução
Desabilitado	0	x	50 μ s
Leitura do relógio	1	0	475 μ s
Acerto do relógio	1	1	690 μ s

- **ENT0** = entrada habilita
- **ENT1** = entrada acerto relógio

F-SINC.049

Situação	ENT0	ENT1	ENT2	Tempo de Execução
Desabilitado	0	x	x	50 μ s
Leitura do relógio	1	0	0	475 μ s
Acerto do relógio no próximo pulso	1	0	1	690 μ s
Acerto do relógio no próximo segundo	1	1	0	690 μ s

- **ENT0** = entrada habilita
- **ENT1** = entrada acerta mantendo sincronismo
- **ENT2** = entrada acerto pulso externo

F-ALNET1.062

Situação	ENT0	Tempo de Execução
Desabilitado	0	50 μ s
Configuração do segundo canal serial (primeiro ciclo do programa)	1	360 μ s
Habilitada sem recepção de comando	1	300 μ s
Habilitada processando comando de monitoração de 1 operando %M	1	330 μ s
Habilitada processando comando de monitoração de 48 operandos %D	1	1700 μ s

- **ENT0** = entrada habilita

F-IMP.063

Situação	ENT0	Tempo de Execução
Desabilitado	0	50 μ s
Habilitado com outra transmissão ativa	1	100 μ s
Habilitado esperando final da transmissão	1	105 μ s
Preparando transmissão de texto com 1 caractere (um ciclo de programa)	1	170 μ s
Preparando transmissão de texto com 100 caracteres (um ciclo do programa)	1	1520 μ s
Preparando transmissão de texto com 255 caracteres (um ciclo de programa)	1	3625 μ s
Preparando transmissão de 1 operando %M (um ciclo de programa)	1	279 μ s
Preparando transmissão de 100 operandos %M (um ciclo de programa)	1	14000 μ s
Preparando transmissão de 255 operandos %M (um ciclo de programa)	1	35000 μ s

- **ENT0** = entrada habilita

F-RECEP.064

Situação	ENT0	Tempo de Execução
Desabilitado	0	50 μ s
Habilitado aguardando recepção	1	70 μ s
Recebendo caracteres	1	80 μ s

- **ENT0** = entrada habilita

Glossário

Glossário de Redes

- **Acesso ao meio:** Método utilizado por todos os nós de uma rede de comunicação para sincronizar as transmissões de dados e resolver possíveis conflitos de transmissões simultâneas.
- **Backoff:** Tempo que um nó de uma rede tipo CSMA/CD aguarda antes de voltar a transmitir dados após a ocorrência de colisão no meio físico.
- **Baud rate:** Taxa com que os bits de informação são transmitidos através de uma interface serial ou rede de comunicação. (medido em Bits/segundo)
- **Bridge (ponte) :** Equipamento para conexão de duas redes de comunicação dentro de um mesmo protocolo.
- **Broadcast:** Disseminação simultânea de informação a todos os nós interligados a uma rede de comunicação.
- **Canal serial:** Interface de um equipamento que transfere dados no modo serial.
- **CSMA/CD.** Disciplina de acesso ao meio físico, baseada na colisão de dados, utilizada pelas redes ETEHRNET.
- **EIA RS-485:** Padrão industrial (nível físico) para comunicação de dados.
- **Escravo:** Equipamento ligado a uma rede de comunicação que só transmite dados se for solicitado por outro equipamento denominado mestre.
- **Frame:** Uma unidade de informação transmitida na rede.
- **Gateway:** Equipamento para a conexão de duas redes de comunicação com diferentes protocolos.
- **Mestre:** Equipamento ligado a uma rede de comunicação de onde se originam solicitações de comandos para outros equipamentos da rede.
- **Multicast:** Disseminação simultânea de informação a um determinado grupo de nós interligados a uma rede de comunicação.
- **Nó ou nodo:** Qualquer estação de uma rede com capacidade de comunicação utilizando um protocolo estabelecido.
- **Peer to peer:** é um tipo de comunicação onde dois parceiros trocam dados e/ou avisos sem depender de um mestre.
- **Protocolo:** Regras de procedimentos e formatos convencionais que, mediante sinais de controle, permitem o estabelecimento de uma transmissão de dados e a recuperação de erros entre equipamentos.
- **Rede de comunicação determinística:** Rede de comunicação onde a transmissão e recepção de informações entre os diversos nós é garantida com um tempo máximo conhecido.
- **Rede de comunicação mestre-escravo:** Rede de comunicação onde as transferências de informações são iniciadas somente a partir de um único nó (o mestre da rede) ligado ao barramento de dados. Os demais nós da rede (escravos) apenas respondem quando solicitados.
- **Rede de comunicação multimestre.** Rede de comunicação onde as transferências de informações são iniciadas por qualquer nó ligado ao barramento de dados.
- **Rede de comunicação:** Conjunto de equipamentos (nós) interconectados por canais de comunicação.
- **Sub rede:** Segmento de uma rede de comunicação que interliga um grupo de equipamentos (nós) com o objetivo de isolar o tráfego local ou utilizar diferentes protocolos ou meio físicos.
- **Time-out:** Tempo preestabelecido máximo para que uma comunicação seja completada, que, se for excedido, provoca a ocorrência de um erro de comunicação.
- **Token:** é uma marca que indica quem é o mestre do barramento no momento.

Glossário Redes PROFIBUS

- **Auto-clear:** parâmetro do PROFIBUS que quando ativado muda o estado do mestre para Clear ao ocorrer um erro na rede.
- **EN 50170:** norma que define a rede de campo PROFIBUS
- **Freeze:** estado da rede PROFIBUS quando os dados das entrada são congelados.
- **Mono-master:** rede PROFIBUS com apenas um mestre.
- **Multi-master:** rede PROFIBUS com mais de um mestre.
- **Sync:** modo de operação da rede PROFIBUS que sincroniza as saídas.

Glossário Geral

- **Algoritmo:** Sequência finita de instruções bem definidas objetivando a resolução de problemas.
- **Arrestor:** Dispositivo de proteção contra raios carregado com gás inerte.
- **Barramento:** Conjunto de sinais elétricos agrupados logicamente com a função de transferir informação e controle entre diferentes elementos de um subsistema.
- **Bit:** Unidade básica de informação, podendo estar no estado 0 ou 1.
- **Byte:** Unidade de informação composta por oito bits.
- **Ciclo de varredura:** Uma execução completa do programa aplicativo de um controlador programável.
- **Circuito de cão-de-guarda:** Circuito eletrônico destinado a verificar a integridade no funcionamento de um equipamento.
- **Controlador Programável:** Equipamento que realiza controle sob o comando de um programa aplicativo escrito em linguagem de relés e blocos. Compõe-se de uma UCP, fonte de alimentação e estrutura de entrada/saída.
- **Database:** banco de dados.
- **Default:** valor pré-definido para uma variável, utilizado em caso de não haver definição.
- **Diagnóstico.** Procedimento utilizado para detectar e isolar falhas. É também o conjunto de dados usados para tal determinação, que serve para a análise e correção de problemas.
- **Download:** carga de programa ou configuração nos módulos.
- **Encoder:** transdutor para medidas de posição.
- **Endereço de módulo:** Endereço pelo qual o CP realiza acessos a um determinado módulo de E/S colocado no barramento.
- **EPROM (Erasable Programmable Read Only Memory) :** Memória somente de leitura, apagável e programável. Não perde seu conteúdo quando desenergizada.
- **Estação de supervisão:** Equipamento ligado a uma rede de CPs ou instrumentação com a finalidade de monitorar ou controlar variáveis de um processo.
- **E2PROM:** Memória não volátil, que pode ser apagada eletricamente.
- **E/S (entrada/saída):** Dispositivos de entrada e/ou saída de dados de um sistema. No caso de CPs, correspondem tipicamente a módulos digitais ou analógicos de entrada ou saída, que monitoram ou acionam o dispositivo controlado.
- **Flash EPROM.** Memória não volátil que pode ser apagada eletricamente.
- **Hardkey:** Conector normalmente ligado à interface paralela do microcomputador com a finalidade de impedir a execução de cópias ilegais de um software.
- **Hardware:** Equipamentos físicos usados em processamento de dados, onde normalmente são executados programas (software).

- **IEC Pub. 144 (1963):** norma para proteção contra acesso incidentais ao equipamento e vedação para água, pó ou outros objetos estranhos ao equipamento.
- **IEC 1131:** Norma genérica para operação e utilização de Controladores Programáveis.
- **IEC-536-1976:** Norma para proteção contra choque elétrico
- **IEC-801-4:** norma para testes de imunidade a interferências por trem de pulsos
- **IEEE C37.90.1 (SWC- Surge Withstand Capability):** norma para proteção contra ruídos tipo onda oscilatória.
- **Interface:** Dispositivo que adapta elétrica e/ou logicamente a transferência de sinais entre dois equipamentos.
- **Interrupção:** Evento com atendimento prioritário que temporariamente suspende a execução de um programa.
- **Kbytes:** Unidade representativa de quantidade de memória. Representa 1024 bytes.
- **LED (Light Emitting Diode):** Tipo de diodo semicondutor que emite luz quando estimulado por eletricidade. Utilizado como indicador luminoso.
- **Linguagem Assemble:** Linguagem de programação do microprocessador, também conhecida como linguagem de máquina.
- **Linguagem de programação:** Um conjunto de regras, de convenções e de sintaxe utilizado para a elaboração de um programa.
- **Linguagem de Relés e Blocos ALTUS:** Conjunto de instruções e operandos que permitem a edição de um programa aplicativo para ser utilizado em um CP.
- **Lógica:** Matriz gráfica onde são inseridas as instruções da linguagem de diagrama de relés que compõem um programa aplicativo. Um conjunto de lógicas ordenadas sequencialmente constitui um módulo de programa.
- **Menu:** Conjunto de opções disponíveis e exibidas no vídeo por um programa, a serem selecionadas pelo usuário a fim de ativar ou executar uma determinada tarefa.
- **Módulo de configuração (Módulo C) :** Módulo único em um programa de CP que contém diversos parâmetros necessários ao funcionamento do controlador, tais como a quantidade de operandos e a disposição dos módulos de E/S no barramento.
- **Módulo de E/S:** Módulo pertencente ao subsistema de Entradas e Saídas.
- **Módulo função (Módulo F):** Módulo de um programa de CP que é chamado a partir do módulo principal (módulo E) ou a partir de outro módulo função ou procedimento, com passagem de parâmetros e retorno de valores, servindo como uma sub-rotina.
- **Módulo procedimento (Módulo P):** Módulo de um programa de CP que é chamado a partir do módulo principal (módulo E) ou a partir de outro módulo procedimento ou função, sem a passagem de parâmetros.
- **Módulo (quando se referir a hardware):** Elemento básico de um sistema completo que possui funções bem definidas. Normalmente é ligado ao sistema por conectores podendo ser facilmente substituído.
- **Módulo (quando se referir a software):** Parte de um programa aplicativo capaz de realizar uma função específica. Pode ser executado independentemente ou em conjunto com outros módulos trocando informações através da passagem de parâmetros.
- **Módulos execução (Módulo E):** Módulos que contêm o programa aplicativo, podendo ser de três tipos: E000, E001 e E018. O módulo E000 é executado uma única vez na energização do CP ou na passagem de programação para execução. O módulo E001 contém o trecho principal do programa que é executado ciclicamente, enquanto que o módulo E018 é acionado por interrupção de tempo.
- **Nibble:** Unidade de informação composta por quatro bits.
- **Octeto:** Conjunto de oito bits numerados de 0 a 7.
- **Operandos:** Elementos sobre os quais as instruções atuam. Podem representar constantes, variáveis ou conjunto de variáveis.
- **PC (Programmable Controller):** Abreviatura de Controlador Programável em inglês.

- **Ponte-de-ajuste:** Chave de seleção de endereços ou configuração, composta por pinos presentes na placa do circuito e um pequeno conector removível, utilizado para a seleção.
- **Posta-em-marcha:** Procedimento de depuração final do sistema de controle, quando os programas de todas as estações remotas e UCPs são executados em conjunto, após terem sido desenvolvidos e verificados individualmente.
- **Programa aplicativo:** É o programa carregado em um CP, que determina o funcionamento de uma máquina ou processo.
- **Programa executivo:** Sistema operacional de um controlador programável; controla as funções básicas do controlador e a execução de programas aplicativos.
- **RAM (Random Access Memory):** Memória onde todos os endereços podem ser acessados diretamente de forma aleatória e a mesma velocidade. É volátil, ou seja, seu conteúdo é perdido quando desenergizada, a menos que possua bateria para retenção dos valores.
- **Ripple:** Ondulação presente em tensão de alimentação contínua.
- **Sistema redundante:** Sistema que contém elementos de reserva ou duplicados para executar determinada tarefa, que podem tolerar determinados tipos de falha sem que execução da tarefa seja comprometida.
- **Software:** Programas de computador, procedimentos e regras relacionadas à operação de um sistema de processamento de dados.
- **Soquete:** Dispositivo no qual se encaixam circuitos integrados ou outros componentes, facilitando a substituição dos mesmos e simplificando a manutenção.
- **Subsistema de E/S:** Conjunto de módulos de E/S digitais ou analógicos e interfaces de um Controlador Programável.
- **Tag:** Nome associado a um operando ou a uma lógica que permite uma identificação resumida de seu conteúdo.
- **Toggle.** Elemento que possui dois estados estáveis, trocados alternadamente a cada ativação.
- **Troca a quente:** Procedimento de substituição de módulos de um sistema sem a necessidade de desenergização do mesmo. Normalmente utilizado em trocas de módulos de E/S.
- **UCP ativa:** Em um sistema redundante, é a UCP que realiza o controle do sistema, lendo os valores dos pontos de entrada, executando o programa aplicativo e acionando os valores das saídas.
- **UCP inoperante:** UCP que não está no estado ativo (controlando o sistema) nem no estado reserva (supervisionando a UCP ativa), não podendo assumir o controle do sistema.
- **UCP redundante:** Corresponde à outra UCP do sistema, em relação à que o texto do manual está se referindo. Por exemplo, a UCP redundante da UCP 2 é a UCP 1 e vice versa.
- **UCP reserva:** Em um sistema redundante, é a UCP que supervisiona a UCP ativa, não realizando o controle do sistema, estando pronta para assumir o controle em caso de falha na UCP ativa.
- **UCP:** Unidade central de processamento. Controla o fluxo de informações, interpreta e executa as instruções do programa e monitora os dispositivos do sistema.
- **Upload:** leitura de programa ou configuração dos módulos.
- **Varistor:** Dispositivo de proteção contra surto de tensão.
- **Word:** Unidade de informação composta por dezesseis bits.

Principais Abreviaturas

- BAT: Bateria
- BT: Teste de Bateria, do inglês "Battery Test"
- CT: Características Técnicas
- CP: Controlador Programável

- DP: Abreviatura para Decentralized Periphery
- EEPROM: "Electric Erasable Programmable Read Only Memory"
- EMI: Electromagnetic Interference. Interferência Eletromagnética
- EPROM: "Erasable Programmable Read Only Memory"
- ER: Erro
- ESD: ElectroStatic Discharge. Descarga devida a eletricidade estática.
- EX: Execução
- E2PROM: "Electric Erasable Programmable Read Only Memory"
- E/S: Entradas e Saídas
- FC: Forçamento
- Flash EPROM: "Flash Erase Programmable Read Only Memory"
- FMS: Abreviatura para Fieldbus Message System
- INTERF.: Interface
- ISOL.: Isolado(s), Isolamento
- LED: diodo emissor de luz, do inglês "Light Emitting Diode"
- Máx.: máximo ou máxima
- Mín.: mínimo ou mínima
- Obs.: observação ou observações
- PAs: Pontes de Ajuste
- PA: Abreviatura para Process Automation
- PG: Programação
- PID: controle Proporcional, Integral e Derivativo.
- RAM: "Random Access Memory"
- ref.: referência
- RX: Recepção Serial
- SELEC.: Seleccionável
- TX: Transmissão serial
- UCP: Unidade Central de Processamento
- UTIL.: Utilização
- WD: cão-de-guarda, do inglês "watchdog"