



Manual de Utilização
UCP Série Nexto
NX3030

MU214103 Rev. F

25 de setembro de 2024

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

Conforme o Código de Defesa do Consumidor vigente no Brasil, informamos, a seguir, aos clientes que utilizam nossos produtos, aspectos relacionados com a segurança de pessoas e instalações.

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 em caso de defeito em seus componentes e/ou de 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, sirvam para preservar a segurança do sistema, principalmente nos casos da instalação inicial e de testes.

Os equipamentos fabricados pela Altus não trazem riscos ambientais diretos, não emitindo nenhum tipo de poluente durante sua utilização. No entanto, no que se refere ao descarte dos equipamentos, é importante salientar que quaisquer componentes eletrônicos incorporados em produtos contêm materiais nocivos à natureza quando descartados de forma inadequada. Recomenda-se, portanto, que quando da inutilização deste tipo de produto, o mesmo seja encaminhado para usinas de reciclagem que deem o devido tratamento para os resíduos.

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

Os exemplos e figuras deste documento são apresentados apenas para fins ilustrativos. Devido às possíveis atualizações e melhorias que os produtos possam incorrer, a Altus não assume a responsabilidade pelo uso destes exemplos e figuras em aplicações reais. Os mesmos devem ser utilizados apenas para auxiliar na familiarização e treinamento do usuário com os produtos e suas características.

A Altus garante os seus equipamentos conforme descrito nas Condições Gerais de Fornecimento, anexada às propostas comerciais.

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.

Os 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.

Alguns produtos utilizam tecnologia EtherCAT (www.ethercat.org).

DIREITOS AUTORAIS

Nexto, MasterTool, Grano e WebPLC são marcas registradas da Altus Sistemas de Automação S.A.

Windows, Windows NT e Windows Vista são marcas registradas da Microsoft Corporation.

NOTIFICAÇÃO DE USO DE SOFTWARE ABERTO

Para obter o código fonte de componentes de software contidos neste produto que estejam sob licença GPL, LGPL, MPL, entre outras, favor entrar em contato através do e-mail opensource@altus.com.br. Adicionalmente ao código fonte, todos os termos da licença, condições de garantia e informações sobre direitos autorais podem ser disponibilizadas sob requisição.

Sumário

1.	Introdução	1
1.1.	Série Nexto	1
1.2.	Características Inovadoras	2
1.3.	Documentos Relacionados a este Manual	3
1.4.	Inspeção Visual	4
1.5.	Suporte Técnico	5
1.6.	Mensagens de Advertência Utilizadas neste Manual	5
2.	Descrição Técnica	6
2.1.	Painéis e Conexões	6
2.2.	Características do Produto	8
2.2.1.	Características Gerais	8
2.2.2.	Normas e Certificações	9
2.2.3.	Memória	10
2.2.4.	Protocolos	12
2.2.5.	Interfaces Seriais	13
2.2.5.1.	COM 1	13
2.2.5.2.	COM 2	13
2.2.6.	Interfaces Ethernet	14
2.2.6.1.	NET 1	14
2.2.6.2.	NET 2	14
2.2.7.	Interface do Cartão de Memória	15
2.2.8.	Características Ambientais	15
2.3.	Compatibilidade com Outros Produtos	15
2.4.	Desempenho	16
2.4.1.	Tempo de Intervalo da MainTask	16
2.4.2.	Tempos de Aplicação	16
2.4.3.	Tempos para Execução de Instruções	16
2.4.4.	Tempos de Inicialização	17
2.5.	Dimensões Físicas	17
2.6.	Dados para Compra	18
2.6.1.	Itens Integrantes	18
2.6.2.	Código do Produto	18
2.7.	Produtos Relacionados	18
3.	Instalação	20
3.1.	Instalação Mecânica	20
3.2.	Instalação Elétrica	20
3.3.	Conexão com a Rede Ethernet	21
3.3.1.	Endereço IP	21

3.3.2.	ARP Gratuito	22
3.3.3.	Instalação do Cabo de Rede	22
3.4.	Conexão com a Rede Serial RS-232C	23
3.4.1.	Comunicação RS-232C	24
3.5.	Conexão com a Rede Serial RS-485/422	24
3.5.1.	Comunicação RS-485 sem Terminação	24
3.5.2.	Comunicação RS-485 com Terminação Interna	25
3.5.3.	Comunicação RS-485 com Terminação Externa	26
3.5.4.	Exemplo de Ligação de Rede RS-485 com Terminação Externa e Redundância de Mestre	27
3.5.5.	Comunicação RS-422 sem Terminação	27
3.5.6.	Comunicação RS-422 com Terminação Interna	28
3.5.7.	Comunicação RS-422 com Terminação Externa	29
3.5.8.	Exemplo de Rede RS-422	30
3.6.	Instalação do Cartão de Memória	30
3.7.	Instalação da Arquitetura	32
3.7.1.	Instalação de Módulos no Bastidor Principal	32
3.8.	Instalação do Programador	32
4.	Programação Inicial	33
4.1.	Organização e Acesso à Memória	33
4.2.	Perfis de Projeto	35
4.2.1.	Simple	35
4.2.2.	Básico	36
4.2.3.	Normal	36
4.2.4.	Experiente	37
4.2.5.	Personalizado	37
4.2.6.	Perfil de Máquina	38
4.2.7.	Tabela Geral	38
4.2.8.	Número Máximo de Tarefas	39
4.3.	Configurando a UCP	39
4.4.	Bibliotecas	40
4.5.	Inserindo uma Instância de Protocolo	41
4.5.1.	MODBUS Ethernet	41
4.6.	Localizando o Dispositivo	42
4.7.	Login	44
4.8.	Modo Run	46
4.9.	Modo Stop	47
4.10.	Escrita e Forçamento de Variáveis	48
4.11.	Logout	48
4.12.	Upload do Projeto	49
4.13.	Estados de Operação da UCP	50
4.13.1.	Run	50
4.13.2.	Stop	50
4.13.3.	Breakpoint	51
4.13.4.	Exception	51
4.13.5.	Reset a Quente	51
4.13.6.	Reset a Frio	51
4.13.7.	Reset Origem	51
4.13.8.	Reset Process Command (IEC 60870-5-104)	51

4.14.	Programas (POUs) e Listas de Variáveis Globais (GVLs)	51
4.14.1.	Programa MainPrg	51
4.14.2.	Programa StartPrg	52
4.14.3.	Programa UserPrg	52
4.14.4.	GVL System_Diagnostics	52
4.14.5.	GVL Disables	53
4.14.6.	GVL IOQualities	54
4.14.7.	GVL Module_Diagnostics	55
4.14.8.	GVL Qualities	55
4.14.9.	GVL ReqDiagnostics	57
4.14.10.	Função Prepare_Start	59
4.14.11.	Função Prepare_Stop	59
4.14.12.	Função Start_Done	59
4.14.13.	Função Stop_Done	59
5.	Configuração	60
5.1.	Device	60
5.1.1.	Gerenciamento de Usuários e Direitos de Acesso	60
5.1.2.	Configurações do CP	60
5.2.	Configuração da UCP	62
5.2.1.	Parâmetros Gerais	62
5.2.1.1.	Troca a Quente	64
5.2.1.1.1.	Troca a Quente Desabilitada, Apenas para Módulos Declarados	65
5.2.1.1.2.	Troca a Quente Desabilitada	65
5.2.1.1.3.	Troca a Quente Desabilitada, sem Consistência na Partida	65
5.2.1.1.4.	Troca a Quente Habilitada, com Consistência na Partida Apenas para Módulos Declarados	65
5.2.1.1.5.	Troca a Quente Habilitada, com Consistência na Partida	65
5.2.1.1.6.	Troca a Quente Habilitada, sem Consistência na Partida	66
5.2.1.1.7.	Como realizar a Troca a Quente	66
5.2.1.2.	Áreas de Memória Retentiva e Persistente	68
5.2.1.3.	Parâmetros do Projeto	69
5.2.2.	Configuração de Evento Externo	70
5.2.3.	Configuração do SOE	71
5.2.4.	Sincronização de Tempo	74
5.2.4.1.	IEC 60870-5-104	75
5.2.4.2.	SNTP	75
5.2.4.3.	Horário de Verão	76
5.2.5.	Pontos Internos	76
5.2.5.1.	Conversões de Qualidade	78
5.2.5.1.1.	Qualidade Interna	79
5.2.5.1.2.	Conversão IEC 60870-5-104	80
5.2.5.1.3.	Qualidade Interna MODBUS	81
5.2.5.1.4.	Qualidade dos Módulos de E/S do Barramento Local	82
5.2.5.1.5.	Qualidade dos Módulos de E/S do Barramento PROFIBUS	82
5.2.5.1.6.	Qualidade da Entrada Digital PROFIBUS	83
5.2.5.1.7.	Qualidade da Saída Digital PROFIBUS	84
5.2.5.1.8.	Qualidade da Entrada Analógica PROFIBUS	85
5.2.5.1.9.	Qualidade da Saída Analógica PROFIBUS	86

5.3.	Configuração das Interfaces Seriais	88
5.3.1.	COM 1	88
5.3.1.1.	Configurações Avançadas	90
5.3.2.	COM 2	91
5.3.2.1.	Configurações Avançadas	92
5.4.	Configuração das Interfaces Ethernet	92
5.4.1.	Interfaces Ethernet Integradas	92
5.4.1.1.	NET 1	93
5.4.1.2.	NET 2	93
5.4.2.	Interface Ethernet Remota NX5000	93
5.4.2.1.	NET 1	93
5.4.2.2.	Modos de Operação da Interface Ethernet Remota NX5000	94
5.4.2.2.1.	Modo Redundante	94
5.4.3.	Portas TCP/UDP Reservadas	95
5.5.	Configuração de Protocolos	95
5.5.1.	Comportamento dos Protocolos x Estados da UCP	98
5.5.2.	Pontos Duplos	99
5.5.3.	Fila de Eventos da UCP	99
5.5.3.1.	Consumidores	100
5.5.3.2.	Princípios de Funcionamento da Fila	100
5.5.3.2.1.	Sinalização de Estouro	100
5.5.3.3.	Produtores	101
5.5.4.	Interceptação de Comandos Oriundos do Centro de Controle	101
5.5.5.	MODBUS RTU Mestre	106
5.5.5.1.	Configuração do Protocolo MODBUS Mestre por Mapeamento Simbólico	106
5.5.5.1.1.	Parâmetros Gerais do Protocolo MODBUS Mestre – Configuração por Mapeamento Simbólico	106
5.5.5.1.2.	Configuração dos Dispositivos – Configuração por Mapeamento Simbólico	109
5.5.5.1.3.	Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico	110
5.5.5.1.4.	Configuração das Requisições – Configuração por Mapeamento Simbólico	111
5.5.5.2.	Configuração do Protocolo MODBUS Mestre por Representação Direta (%Q)	116
5.5.5.2.1.	Parâmetros Gerais do Protocolo MODBUS Mestre – Configuração por Representação Direta (%Q)	116
5.5.5.2.2.	Configuração dos Dispositivos – Configuração por Representação Direta (%Q)	118
5.5.5.2.3.	Configuração dos Mapeamentos – Configuração por Representação Direta (%Q)	119
5.5.6.	MODBUS RTU Escravo	121
5.5.6.1.	Configuração do Protocolo MODBUS Escravo por Mapeamento Simbólico	121
5.5.6.1.1.	Parâmetros Gerais do Protocolo MODBUS Escravo – Configuração por Mapeamento Simbólico	121
5.5.6.1.2.	Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico	125
5.5.6.2.	Configuração do Protocolo MODBUS Escravo por Representação Direta (%Q)	126
5.5.6.2.1.	Parâmetros Gerais do Protocolo MODBUS Escravo – Configuração por Representação Direta (%Q)	127
5.5.6.2.2.	Configuração dos Mapeamentos – Configuração por Representação Direta (%Q)	128
5.5.7.	MODBUS Ethernet	130
5.5.8.	MODBUS Ethernet Cliente	131
5.5.8.1.	Configuração do Protocolo MODBUS Ethernet Cliente por Mapeamento Simbólico	132

5.5.8.1.1.	Parâmetros Gerais do Protocolo MODBUS Cliente – Configuração por Mapeamento Simbólico	132
5.5.8.1.2.	Configuração dos Dispositivos – Configuração por Mapeamento Simbólico	134
5.5.8.1.3.	Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico	135
5.5.8.1.4.	Configuração das Requisições – Configuração por Mapeamento Simbólico .	137
5.5.8.2.	Configuração do Protocolo MODBUS Ethernet Cliente por Representação Direta (%Q)	141
5.5.8.2.1.	Parâmetros Gerais do Protocolo MODBUS Cliente – Configuração por Representação Direta (%Q)	142
5.5.8.2.2.	Configuração dos Dispositivos – configuração por Representação Direta (%Q)	143
5.5.8.2.3.	Configuração dos Mapeamentos – Configuração por Representação Direta (%Q)	143
5.5.8.3.	Disparo de Relações MODBUS Cliente de Forma Acíclica	146
5.5.9.	MODBUS Ethernet Servidor	146
5.5.9.1.	Configuração do Protocolo MODBUS Ethernet Servidor por Mapeamento Simbólico	146
5.5.9.1.1.	Parâmetros Gerais do Protocolo MODBUS Servidor – Configuração por Mapeamento Simbólico	146
5.5.9.1.2.	Diagnósticos MODBUS Servidor – Configuração por Mapeamento Simbólico	148
5.5.9.1.3.	Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico	150
5.5.9.2.	Configuração do Protocolo MODBUS Ethernet Servidor por Representação Direta (%Q)	151
5.5.9.2.1.	Parâmetros Gerais do Protocolo MODBUS Servidor – Configuração por Representação Direta (%Q)	152
5.5.9.2.2.	Configuração dos Mapeamentos – Configuração por Representação Direta (%Q)	153
5.5.10.	OPC DA Servidor	156
5.5.10.1.	Criando um Projeto para Comunicação OPC DA	157
5.5.10.2.	Configurando um CP no Servidor OPC DA	160
5.5.10.2.1.	Importando uma Configuração do Projeto	163
5.5.10.3.	Configuração com CP no Servidor OPC DA com Redundância de Conexão	163
5.5.10.4.	Variáveis de Status e Qualidade da Comunicação OPC DA	164
5.5.10.5.	Limites da Comunicação com Servidor OPC DA	166
5.5.10.6.	Acessando Dados Através de um Cliente OPC DA	166
5.5.11.	OPC UA Servidor	168
5.5.11.1.	Criando um Projeto para Comunicação OPC UA	169
5.5.11.2.	Tipos de Variáveis Suportadas	171
5.5.11.3.	Limite de Clientes Conectados no Servidor OPC UA	171
5.5.11.4.	Limite de Variáveis de Comunicação no Servidor OPC UA	171
5.5.11.5.	Configurações de Criptografia	171
5.5.11.6.	Principais Parâmetros de Comunicação Ajustados em um Cliente OPC UA	172
5.5.11.6.1.	Endpoint URL	172
5.5.11.6.2.	Publishing Interval (ms) e Sampling Interval (ms)	172
5.5.11.6.3.	Lifetime Count e Keep-Alive Count	173
5.5.11.6.4.	Queue Size e Discard Oldest	173
5.5.11.6.5.	Filter Type e Deadband Type	173
5.5.11.6.6.	PublishingEnabled, MaxNotificationsPerPublish e Priority	174
5.5.11.7.	Acessando Dados Através de um Cliente OPC UA	174
5.5.12.	Mestre EtherCAT	175
5.5.12.1.	Instalando e Inserindo Dispositivos EtherCAT	175

5.5.12.1.1.	EtherCAT - Scan Devices	176
5.5.12.2.	Configuração do Mestre EtherCAT	177
5.5.12.2.1.	EtherCAT Master - General	177
5.5.12.2.2.	EtherCAT Master - Sync Unit Assignment	179
5.5.12.2.3.	EtherCAT Master - Overview	179
5.5.12.2.4.	EtherCAT Master - I/O Mapping	179
5.5.12.2.5.	EtherCAT Master - IEC Objects	179
5.5.12.2.6.	EtherCAT Master - Status / Informations	179
5.5.12.3.	Configuração do Escravo EtherCAT	179
5.5.12.3.1.	EtherCAT Slave - General	179
5.5.12.3.2.	EtherCAT Slave - Process Data	183
5.5.12.3.3.	EtherCAT Slave - Edit PDO List	185
5.5.12.3.4.	EtherCAT Slave - Startup Parameters	185
5.5.12.3.5.	EtherCAT Slave - Module I/O Mapping	185
5.5.12.3.6.	EtherCAT Slave - Status and Information	186
5.5.13.	EtherNet/IP	186
5.5.13.1.	Interface EtherNet/IP	187
5.5.13.2.	Configuração do Scanner EtherNet/IP	189
5.5.13.2.1.	Geral	189
5.5.13.2.2.	Conexões	190
5.5.13.2.3.	Assemblies	192
5.5.13.2.4.	EtherNet/IP: Mapeamento de E/S	193
5.5.13.3.	Configuração do Adapter EtherNet/IP	193
5.5.13.3.1.	Geral	193
5.5.13.3.2.	EtherNet/IP Adapter: Mapeamento de E/S	194
5.5.13.4.	Configuração do Módulo EtherNet/IP	194
5.5.13.4.1.	Assemblies	195
5.5.13.4.2.	EtherNet/IP Module: Mapeamento de E/S	195
5.5.14.	IEC 60870-5-104 Servidor	195
5.5.14.1.	Tipos de Dados	195
5.5.14.2.	Pontos Duplos	197
5.5.14.2.1.	Pontos Duplos de Entrada Digital	197
5.5.14.2.2.	Pontos Duplos de Saída Digital	199
5.5.14.3.	Parâmetros Gerais	204
5.5.14.4.	Mapeamento dos Dados	204
5.5.14.5.	Camada de Enlace	206
5.5.14.6.	Camada de Aplicação	208
5.5.14.7.	Diagnósticos do Servidor	210
5.5.14.8.	Qualificador dos Comandos	211
5.5.15.	PROFINET Controller	212
5.6.	Desempenho de Comunicação	212
5.6.1.	Servidor MODBUS	212
5.6.1.1.	Interfaces Locais da UCP	213
5.6.1.2.	Interfaces Remotas	214
5.6.2.	Servidor OPC DA	214
5.6.3.	Servidor OPC UA	214
5.6.4.	Servidor IEC 60870-5-104	214
5.7.	Desempenho do Sistema	215

5.7.1.	Escaneamento de E/S	215
5.7.2.	Cartão de Memória	216
5.8.	Relógio RTC	216
5.8.1.	Blocos Funcionais e Funções para Leitura e Escrita do RTC	217
5.8.1.1.	Funções de Leitura do RTC	217
5.8.1.1.1.	GetDateAndTime	217
5.8.1.1.2.	GetTimeZone	218
5.8.1.1.3.	GetDayOfWeek	219
5.8.1.2.	Funções de Escrita do RTC	219
5.8.1.2.1.	SetDateAndTime	220
5.8.1.2.2.	SetTimeZone	221
5.8.2.	Estruturas de Dados do RTC	222
5.8.2.1.	EXTENDED_DATE_AND_TIME	222
5.8.2.2.	DAYS_OF_WEEK	223
5.8.2.3.	RTC_STATUS	223
5.8.2.4.	TIMEZONESETTINGS	224
5.9.	Memória de Arquivos de Usuário	224
5.10.	Cartão de Memória	226
5.10.1.	Preparação do Projeto	227
5.10.2.	Envio do Projeto	228
5.10.3.	Acesso no MasterTool	228
5.11.	Menu Informativo e de Configuração da UCP	229
5.12.	Blocos Funcionais e Funções	231
5.12.1.	Blocos Funcionais Especiais para Comunicação Serial	231
5.12.1.1.	SERIAL_CFG	236
5.12.1.2.	SERIAL_GET_CFG	238
5.12.1.3.	SERIAL_GET_CTRL	239
5.12.1.4.	SERIAL_GET_RX_QUEUE_STATUS	241
5.12.1.5.	SERIAL_PURGE_RX_QUEUE	243
5.12.1.6.	SERIAL_RX	244
5.12.1.7.	SERIAL_RX_EXTENDED	246
5.12.1.8.	SERIAL_SET_CTRL	249
5.12.1.9.	SERIAL_TX	250
5.12.2.	Atualização de Entradas e Saídas	253
5.12.2.1.	REFRESH_INPUT	253
5.12.2.2.	REFRESH_OUTPUT	254
5.12.3.	Bloco Funcional PID	256
5.12.4.	Temporizador Retentivo	256
5.12.4.1.	TOF_RET	256
5.12.4.2.	TON_RET	258
5.12.4.3.	TP_RET	259
5.12.5.	Temporizador Não-Redundante	260
5.12.5.1.	TOF_NR	260
5.12.5.2.	TON_NR	261
5.12.5.3.	TP_NR	262
5.12.6.	Log de Usuário	262
5.12.6.1.	UserLogAdd	263
5.12.6.2.	UserLogDeleteAll	265

5.12.7.	ClearRtuDiagnostic	266
5.12.8.	ClearEventQueue	266
5.13.	SNMP	267
5.13.1.	Introdução	267
5.13.2.	SNMP nas UCPs Nexto	267
5.13.3.	Private MIB	268
5.13.4.	Configuração SNMP	268
5.13.5.	Usuário e Comunidades SNMP	270
6.	Redundância com UCP NX3030	271
6.1.	Introdução	271
6.2.	Descrição Técnica e Configuração	273
6.2.1.	Configuração Mínima de um CP Redundante (Sem utilização do Painel PX2612)	273
6.2.2.	Configurações Típicas de um CP Redundante	273
6.2.2.1.	Adição de Módulos NX5001 para Redes PROFIBUS	274
6.2.2.2.	Adição de Módulos NX5000 para Redes Ethernet	274
6.2.3.	Módulo NX4010	274
6.2.3.1.	Características NX4010	275
6.2.4.	Painel de Controle de Redundância PX2612	275
6.2.4.1.	Características PX2612	277
6.2.5.	Interconexões entre Half-Clusters e Painel de Controle de Redundância PX2612	277
6.2.6.	Características Gerais de um CP Redundante	278
6.2.7.	Dados para Compra	281
6.3.	Princípios de Funcionamento	282
6.3.1.	Identificação de uma UCP NX3030	282
6.3.2.	Projeto Redundante Único	282
6.3.3.	Estrutura do Projeto Redundante	283
6.3.3.1.	Redundancy Template	283
6.3.3.2.	Tarefa Única e Cíclica MainTask	283
6.3.3.3.	Programa MainPrg	283
6.3.3.4.	Programa ActivePrg	284
6.3.3.5.	Programa NonSkippedPrg	284
6.3.3.6.	Variáveis Redundantes e Não-Redundantes	284
6.3.3.7.	Variáveis %I Redundantes e Não-Redundantes	284
6.3.3.8.	Variáveis %Q Redundantes e Não-Redundantes	285
6.3.3.9.	Variáveis %M Redundantes e Não-Redundantes	286
6.3.3.10.	Variáveis Simbólicas Redundantes e Não-Redundantes	286
6.3.4.	Mapeamentos Múltiplos	287
6.3.5.	Estruturas de Dados de Diagnósticos, Comandos e Usuário	288
6.3.6.	Serviços de Sincronização Cíclicos através de NETA e NETB	288
6.3.6.1.	Troca de Diagnósticos e Comandos	289
6.3.6.2.	Sincronização de Dados Redundantes	289
6.3.6.3.	Sincronização da Lista de Forçamentos Redundantes	289
6.3.7.	Serviços de Sincronização Esporádicos através de NETA e NETB	289
6.3.7.1.	Sincronização de Projetos	290
6.3.8.	Desabilitação da Sincronização de Projetos	291
6.3.9.	Configurações de Redes PROFIBUS	291
6.3.9.1.	Redundância PROFIBUS	292

6.3.9.2.	Modos de Falha PROFIBUS Vital e Não-Vital	292
6.3.10.	Redes Ethernet Redundantes com NIC Teaming	292
6.3.11.	Métodos de Troca de IP	293
6.3.11.1.	IP Fixo	293
6.3.11.2.	Troca Automática de IP	293
6.3.11.3.	IP Ativo	294
6.3.11.4.	Múltiplos IPs	295
6.3.12.	Uso Combinado de NIC Teaming e IP Ativo	296
6.3.13.	Uso de Interfaces Ethernet com Indicação de Falha Vital	296
6.3.13.1.	Falha na Interface Ethernet	297
6.3.13.2.	Falha em MODBUS Servidor conectado	297
6.3.14.	Uso de Comunicação OPC DA com Projetos Redundantes	297
6.3.15.	Estados de um CP Redundante	297
6.3.15.1.	Estado Não-Configurado	297
6.3.15.2.	Estado Inicializando	298
6.3.15.3.	Estado Ativo	298
6.3.15.4.	Estado Reserva	299
6.3.15.5.	Estado Inativo	299
6.3.16.	Funções do Painel de Comando de Redundância PX2612	299
6.3.16.1.	Botões do PX2612	299
6.3.16.2.	LEDs do PX2612	300
6.3.16.3.	Relés do PX2612	301
6.3.17.	Transições entre Estados de Redundância	301
6.3.17.1.	Transição 1 - Não-Configurado para Inicializando	302
6.3.17.2.	Transição 2 - Inicializando para Não-Configurado	302
6.3.17.3.	Transição 3 - Inicializando para Inativo	302
6.3.17.4.	Transição 4 - Inicializando para Ativo	302
6.3.17.5.	Transição 5 - Inicializando para Reserva	302
6.3.17.6.	Transição 6 - Inativo para Não-Configurado	303
6.3.17.7.	Transição 7 - Ativo para Não-Configurado	303
6.3.17.8.	Transição 8 - Ativo para Inativo	303
6.3.17.9.	Transição 9 - Ativo para Reserva	303
6.3.17.10.	Transição 10 – Reserva para Não-Configurado	303
6.3.17.11.	Transição 11 – Reserva para Inativo	303
6.3.17.12.	Transição 12 – Reserva para Ativo	304
6.3.18.	Primeiros Instantes em Estado Ativo	304
6.3.19.	Falhas mais Comuns Causadoras de Switch-overs Automáticos entre Half-Clusters	304
6.3.20.	Falhas Associadas a Switch-overs entre Half-Clusters Gerenciados pelo Usuário	305
6.3.21.	Tolerância a Falhas	305
6.3.21.1.	Falhas Simples com Indisponibilidade	306
6.3.21.2.	Falhas Simples sem Indisponibilidade Causando um Switch-over	307
6.3.21.3.	Falhas Duplas sem Indisponibilidade Causando um Switch-over	307
6.3.22.	Overhead da Redundância	307
6.4.	Programação de um CP Redundante	308
6.4.1.	Assistente para Criação de um Novo Projeto Redundante	308
6.4.2.	Configuração dos Half-Clusters	312
6.4.2.1.	Configuração Fixa nas Posições 0 a 5 do Bastidor	312
6.4.3.	Configurações das Portas Ethernet da UCP NX3030 (NET 1 e NET 2)	312

6.4.3.1.	Configuração do Endereço de IP	312
6.4.3.2.	NIC Teaming entre NET 1 e NET 2	313
6.4.3.3.	Configuração de falha vital na NET 1 e NET 2	314
6.4.4.	Configurações dos Módulos NX5001	314
6.4.4.1.	Inserção ou Remoção de Módulos NX5001	314
6.4.4.2.	Ajuste de Parâmetros dos Módulos NX5001	314
6.4.4.3.	Configurações de Remotas PROFIBUS	316
6.4.5.	Configurações dos Módulos NX5000	316
6.4.5.1.	Inserção ou Remoção de Módulos NX5000	316
6.4.5.2.	Configuração dos Módulos NX5000	316
6.4.5.3.	Agrupamento de Módulos NX5000 com Redundância NIC Teaming	317
6.4.5.3.1.	Configuração de Falha Vital	317
6.4.6.	Configurações do Módulo NX4010	317
6.4.7.	Configurações de Drivers de E/S	318
6.4.8.	Configurações da MainTask	318
6.4.8.1.	Programa ActivePrg	319
6.4.8.2.	Programa NonSkippedPrg	319
6.4.9.	Objeto Configuração de Redundância	320
6.4.10.	GVL Module_Diagnostics	320
6.4.11.	GVLs com Variáveis Simbólicas Redundantes	320
6.4.12.	POUs do Tipo Programa com Variáveis Simbólicas Redundantes	320
6.4.13.	Utilização de Breakpoints em Sistemas Redundantes	321
6.4.14.	Gerenciamento de Instâncias MODBUS em Sistemas Redundantes	321
6.4.15.	Limitações na Programação de um CP Redundante	321
6.4.15.1.	Limitações em GVLs e POU's Redundantes	321
6.4.15.2.	Limitações no Programa Não-Redundante (NonSkippedPrg)	322
6.4.16.	Obtendo o Estado da Redundância de um Half-Cluster	322
6.4.17.	Leitura de Diagnósticos Não-Redundantes	322
6.5.	Carga de Programas em um CP Redundante	323
6.5.1.	Carga Inicial de um Projeto Redundante	323
6.5.1.1.	Primeiro Passo - Descoberta do Endereço IP para Conexão do MasterTool	323
6.5.1.2.	Segundo Passo – Verificar Conflito de Endereços IP	323
6.5.1.3.	Terceiro Passo – Preparar Conexão do MasterTool (Definir Caminho Ativo)	323
6.5.1.4.	Quarto Passo – Identificar a UCP NX3030 e Conferir no Visor da UCP	324
6.5.1.5.	Quinto Passo – Carga do Projeto Redundante	325
6.5.2.	Conexão do MasterTool com uma UCP NX3030 de um CP Redundante	325
6.5.3.	Carga de Modificações em um Projeto Redundante	326
6.5.4.	Carga de Modificações Offline e Online	326
6.5.4.1.	Modificações que Demandam Carga Offline com Interrupção do Controle do Processo	326
6.5.4.2.	Modificações que Demandam Carga Offline	327
6.5.4.3.	Modificações que Permitem Carga Online	327
6.5.5.	Carga Online de Modificações	327
6.5.6.	Carga Offline de Modificações com Interrupção do Controle do Processo	328
6.5.7.	Planejamento Prévio para Modificações Offline sem Interrupção do Controle do Processo	329
6.5.7.1.	Planejamento Prévio para Modificações a Quente em Redes PROFIBUS Redundantes	329
6.5.7.1.1.	Etapa 1 – Planejar Expansão Futura das Remotas Inseridas na Versão Inicial da Rede PROFIBUS	329

6.5.7.1.2.	Etapa 2 – Inserir a Versão Inicial da Rede PROFIBUS Redundante no Projeto330	
6.5.7.1.3.	Etapa 3 – Alocar Áreas de Variáveis %I e %Q para a Rede PROFIBUS considerando Expansão para Remotas Futuras	330
6.5.7.2.	Planejamento Prévio para Outras Modificações	331
6.5.7.3.	Incompatibilidade de Aplicações	332
6.5.7.4.	Atualizações de Projeto Devido a Atualizações do MasterTool IEC XE	332
6.5.7.4.1.	Atualização de Projetos de Versões Inferiores a 2.00 para versão 2.00 ou Superior	332
6.5.8.	Explorando a Redundância para Carga Offline de Modificações sem Interrupção do Controle do Processo	333
6.5.8.1.	Etapa 1 – Verificar Atendimento de Requisitos Básicos	333
6.5.8.2.	Etapa 2 – Não Carregar em Conjunto Modificações que podem ser Carregadas Online334	
6.5.8.3.	Etapa 3 – Backup do Projeto Anterior	334
6.5.8.4.	Etapa 4 – Cuidados ao Editar as Modificações Carregadas Offline	334
6.5.8.5.	Etapa 5 – Desabilitar Sincronismo de Projetos no CP Não-Ativo	334
6.5.8.6.	Etapa 6 – Executar as Modificações Físicas	335
6.5.8.7.	Etapa 7 – Carregar as Modificações Offline no CP Não-Ativo	335
6.5.8.8.	Etapa 8 – Voltar o CP Não-Ativo ao Modo Run para que volte ao Estado Reserva .	335
6.5.8.9.	Etapa 9 – Executar Switch-over entre CPs Ativo e Reserva	335
6.5.8.10.	Etapa 10 – Habilitar Sincronismo de Projetos no CP Ativo	335
6.5.8.11.	Etapa 11 – Reorganização Opcional de CP e Redes PROFIBUS em Estado Ativo .	336
6.6.	Manutenção da Redundância	336
6.6.1.	Troca a Quente de Módulos em um CP Redundante	336
6.6.2.	Mensagens de Advertência do MasterTool	336
6.6.2.1.	Bloqueio de Carga de um Projeto Redundante ou Não-Redundante	336
6.6.2.2.	Alertas antes de Comandos que Podem Parar o CP Ativo	336
6.6.2.3.	Alerta antes de Logar-se ao CP Não-Ativo	337
6.6.3.	Diagnósticos da Redundância no Visor Gráfico da UCP NX3030	337
6.6.3.1.	Estado de Redundância do CP	337
6.6.3.2.	Telas abaixo do Menu REDUNDANCIA	337
6.6.4.	Estrutura de Diagnósticos da Redundância	337
6.6.4.1.	Diagnósticos da Redundância	338
6.6.4.2.	Comandos da Redundância	349
6.6.4.3.	Informações do Usuário Trocadas entre CPA e CPB	352
6.6.4.4.	Diagnósticos MODBUS utilizados na Redundância	353
6.6.4.5.	Log de Eventos da Redundância	353
6.6.5.	Teste do Painel PX2612	353
6.6.5.1.	Entrada no Modo Teste	354
6.6.5.2.	Saídas Manual e Automática do Modo Teste	354
6.6.5.3.	Teste dos LEDs	354
6.6.5.4.	Teste dos Botões	354
6.6.5.5.	Teste dos Relés	354
6.6.5.6.	Sequência Sugerida para Executar os Testes do PX2612	355
7.	Manutenção	356
7.1.	Diagnósticos do Módulo	356
7.1.1.	One Touch Diag	356
7.1.2.	Diagnósticos via LED	359
7.1.2.1.	DG (Diagnóstico)	359

7.1.2.2.	WD (Cão-de-guarda)	359
7.1.2.3.	LEDs Conector RJ45	360
7.1.3.	Diagnósticos via Página Web de Sistema	360
7.1.4.	Diagnósticos via Variáveis	362
7.1.4.1.	Diagnósticos Resumidos	363
7.1.4.2.	Diagnósticos Detalhados	365
7.1.5.	Diagnósticos via Blocos Funcionais	376
7.1.5.1.	GetTaskInfo	377
7.2.	Visor Gráfico	378
7.3.	Log de Sistema	380
7.4.	Não Carregar a Aplicação na Inicialização	380
7.5.	Falha na Alimentação	381
7.6.	Problemas mais Comuns	381
7.7.	Solução de Problemas	382
7.8.	Manutenção Preventiva	382
8.	Anexo - Interoperabilidade DNP3	383
8.1.	DNP3 Device Profile	383
8.2.	DNP V3.0 Implementation Table	384

1. Introdução

Os controladores programáveis da Série Nexto são a solução definitiva para automação industrial e controle de sistemas. Com alta tecnologia embarcada, os produtos da família são capazes de controlar, de forma distribuída e redundante, complexos sistemas industriais, máquinas, linhas de produção de alto desempenho e os mais avançados processos da Indústria 4.0. Moderna e de alta velocidade, a série Nexto utiliza tecnologia de ponta para proporcionar confiabilidade e conectividade, contribuindo para o aumento de produtividade de diferentes negócios.

Compactos, robustos e com alta disponibilidade, os produtos da série possuem excelente desempenho de processamento e possibilidade de expansão de bastidores. Sua arquitetura permite fácil integração com redes de supervisão, controle e de campo, além de redundância de CLPs. Os equipamentos da família também oferecem diagnósticos avançados e troca a quente, minimizando ou eliminando o tempo de parada de manutenção e garantindo um processo de produção contínuo.



Figura 1: NX3030

1.1. Série Nexto

A Série Nexto é uma poderosa e completa série de Controladores Programáveis (CP) com características exclusivas e inovadoras. Devido a sua flexibilidade, design funcional, recursos de diagnóstico avançado e arquitetura modular, o CP Nexto pode ser usado para controle de sistemas em aplicações de pequeno, médio ou grande porte.

A arquitetura da Série Nexto possui uma extensa variedade de módulos de entradas e saídas. Estes módulos combinados com um poderoso processador e um barramento de alta velocidade baseado em Ethernet se adequam a inúmeros tipos de aplicações como controle de alta velocidade para máquinas de pequeno porte, complexos processos distribuídos, aplicações redundantes e sistemas com grande número de E/S. Além disto, a Série Nexto possui módulos de comunicações com as mais populares redes de campo, entre outras características.

A Série Nexto possui uma avançada tecnologia em seu barramento que utiliza uma interface Ethernet determinística de alta velocidade, possibilitando que informações de entradas, saídas e dados possam ser compartilhadas entre os módulos do sistema com máxima eficiência. O sistema pode ser facilmente distribuído em campo, possibilitando o uso de expansões de bastidores com o mesmo desempenho de um módulo local permitindo que todos os tipos de módulos sejam utilizados tanto no bastidor local quanto nas expansões de bastidores sem restrições. Para a interligação entre as expansões de bastidores é utilizado um simples cabo padrão Ethernet.



Figura 2: Série Nexto – Visão Geral

1.2. Características Inovadoras

A Série Nexto traz aos usuários diversas inovações na utilização, supervisão e manutenção do sistema. Estas características foram desenvolvidas focando um novo conceito em automação industrial.



Battery Free Operation: A Série Nexto não requer nenhum tipo de bateria para manutenção de memória e operação de relógio de tempo real. Esta funcionalidade é extremamente importante porque reduz a necessidade de manutenção do sistema e permite o uso em locais remotos de difícil manutenção. Além disso, esta característica é ambientalmente correta.



Easy Plug System: A Série Nexto conta com um exclusivo método para conectar e desconectar bornes de E/S. Estes bornes são facilmente removíveis com um simples movimento e sem ferramentas especiais. Para conectar o borne novamente ao módulo, a tampa frontal auxilia o procedimento de inserção, encaixando o borne ao módulo.



Multiple Block Storage: Diversos tipos de memória estão disponíveis nas UCPs da Série Nexto, oferecendo a melhor opção para cada necessidade. Estas memórias são divididas em memórias voláteis e memórias não voláteis. Para uso de memórias voláteis, as UCPs da Série Nexto oferecem variáveis de entrada de representação direta (%I), variáveis de saída de representação direta (%Q), variáveis de memória de representação direta (%M), memória de dados e memória de dados redundantes. Para aplicações que necessitam funcionalidades de memória não volátil, a Série Nexto possibilita a utilização de variáveis de representação direta de memória retentiva (%Q), memória retentiva de dados, variáveis de representação direta de memória persistente (%Q), memória persistente de dados, memória de programa, memória de código fonte, sistema de arquivo na UCP (Doc, pdf, dados) e interface para cartão de memória.



One Touch Diag: Esta é uma característica exclusiva dos CPs da Série Nexto. Através deste novo conceito, o usuário pode checar as informações de diagnóstico de qualquer módulo do sistema diretamente no visor gráfico da UCP, mediante apenas um pressionamento no botão de diagnóstico do respectivo módulo. A OTD é uma poderosa ferramenta de diagnóstico que pode ser usada offline (sem supervisor ou programador) e reduz os tempos de manutenção e comissionamento.

OFD – On Board Full Documentation: As UCPs da Série Nexto têm a capacidade de armazenar a documentação completa do projeto na sua memória. Este é um recurso interessante para fins de backup e manutenção, já que a informação completa fica armazenada em um único e seguro local.

ETD – Electronic Tag on Display: Outra característica exclusiva apresentada pela Série Nexto é o ETD. Esta nova funcionalidade possibilita a verificação da tag de qualquer ponto ou módulo de E/S usado no sistema, diretamente no visor gráfico das UCPs. Juntamente com esta informação, o usuário pode também verificar a descrição. Este é um recurso extremamente útil durante a manutenção e resolução de problemas.

DHW – Double Hardware Width: Os módulos da Série Nexto foram projetados para economizar espaço em painéis e nas máquinas. Por esta razão, a Série Nexto oferece duas diferentes larguras de módulos: largura dupla (com ocupação de 2 posições do bastidor) e largura simples (com ocupação de 1 posição do bastidor). Este conceito permite o uso de módulos de E/S compactos, com alta densidade de pontos de E/S, juntamente com módulos complexos, como UCPs, mestres de rede de campo e módulos de fonte de alimentação.

UCP de Alta Velocidade: Todas as UCPs desta Série Nexto foram concebidas para fornecer ao usuário um excelente desempenho e atender a uma ampla gama de exigências nas aplicações.



iF Product Design Award 2012: A Série Nexto foi vencedora do iF Product Design Award 2012 no grupo industry + skilled trades. Este prêmio é reconhecido internacionalmente como um selo de excelência e qualidade, considerado o Oscar do design na Europa.

1.3. Documentos Relacionados a este Manual

Para obter informações adicionais sobre a Série Nexto podem ser consultados outros documentos (manuais e características técnicas) além deste. Estes documentos encontram-se disponíveis em www.altus.com.br.

Cada produto possui um documento denominado Características Técnicas (CT), e neste documento encontram-se as características do produto em questão. Caso o produto possua mais informações, ele pode ter também um manual de utilização.

Por exemplo, o módulo NX2020 tem todas as informações de características, utilização e de compra, na sua CT. Por outro lado, o NX5001 possui, além da CT, um manual de utilização.

Aconselham-se os seguintes documentos como fonte de informação adicional:

Código	Descrição	Idioma
CE114000 CT114000	Nexto Series – Technical Characteristics Série Nexto – Características Técnicas	Inglês Português
CE114102 CT114102	NX3030 Technical Characteristics Características Técnicas NX3030	Inglês Português
CE114200 CT114200	NX8000 Power Supply Module Technical Characteristics Características Técnicas Fonte de Alimentação NX8000	Inglês Português
CE114700 CT114700	Nexto Series Backplane Racks Technical Characteristic Características Técnicas dos Bastidores da Série Nexto	Inglês Português
CE114810 CT114810	Nexto Series Accessories for Backplane Rack Technical Characteristics Características Técnicas Acessórios para Bastidor Série Nexto	Inglês Português
CE114900 CT114900	NX4010 Redundancy Link Module Technical Characteristics Características Técnicas do Módulo de Redundância NX4010	Inglês Português
CE114902 CT114902	Nexto Series PROFIBUS-DP Master Technical Characteristics Características Técnicas do Mestre PROFIBUS-DP da Série Nexto	Inglês Português
CE114903 CT114903	Nexto Series Ethernet Module Technical Characteristics Características Técnicas Módulo Ethernet Série Nexto	Inglês Português
CE114908 CT114908	NX5110 and NX5210 PROFIBUS-DP Heads Technical Characteristics Características Técnicas Interfaces Cabeça PROFIBUSDP NX5110 e NX5210	Inglês Português
CT112500	Características Técnicas do Painel de Controle de Redundância PX2612	Português
MU214600 MU214000	Nexto Series User Manual Manual de Utilização Série Nexto	Inglês Português
MU214615 MU214103	NX3030 CPU User Manual Manual de Utilização UCP NX3030	Inglês Português
MU299609 MU299048	MasterTool IEC XE User Manual Manual de Utilização MasterTool IEC XE	Inglês Português

Código	Descrição	Idioma
MP399609	MasterTool IEC XE Programming Manual	Inglês
MP399048	Manual de Programação MasterTool IEC XE	Português
MU214601	NX5001 PROFIBUS DP Master User Manual	Inglês
MU214001	Manual de Utilização Mestre PROFIBUS-DP NX5001	Português
MU214608	Nexto PROFIBUS-DP Head Utilization Manual	Inglês
MU214108	Manual de Utilização da Cabeça PROFIBUS-DP Nexto	Português
MU219000	Ponto Series Utilization Manual	Inglês
MU209000	Manual de Utilização da Série Ponto	Português
MU209508	Manual de Utilização Cabeça PROFIBUS PO5063V1 e Cabeça Redundante PROFIBUS PO5063V5	Português
MU219511	PO5064 PROFIBUS Head and PO5065 Redundant PROFIBUS Head Utilization Manual	Inglês
MU209511	Manual de Utilização Cabeça PROFIBUS PO5064 e Cabeça Redundante PROFIBUS PO5065	Português
MU209020	Manual de Utilização Rede HART sobre PROFIBUS	Português
MU223603	IEC 60870-5-104 Server Device Profile Document	Inglês
MU214603	Nexto Series HART Manual	Inglês
MU214606	MQTT User Manual	Inglês
MU214609	OPC UA Server for Altus Controllers User Manual	Inglês
MU214610	Advanced Control Functions User Manual	Inglês
MU214621	Nexto Series PROFINET Manual	Inglês
NAP151	Utilização do Tunneller OPC	Português

Tabela 1: Documentos Relacionados

1.4. Inspeção Visual

Antes de proceder à instalação, é recomendável fazer uma inspeção visual cuidadosa dos equipamentos, verificando se não há danos causados pelo transporte. Verifique se todos os componentes de seu pedido estão em perfeito estado. Em caso de defeitos, informe a companhia transportadora ou o distribuidor Altus mais próximo.

CUIDADO

Antes de retirar os módulos da embalagem, é importante descarregar eventuais potenciais estáticos acumulados no corpo. Para isso, toque (com as mãos nuas) em uma superfície metálica aterrada qualquer antes de manipular os módulos. Tal procedimento garante que os níveis de eletricidade estática suportados pelo módulo não serão ultrapassados.

É importante registrar o número de série de cada equipamento recebido, bem como as revisões de software, caso existentes. Essas informações serão necessárias caso se necessite contatar o Suporte Técnico da Altus.

1.5. Suporte Técnico

Para entrar em contato com o Suporte Técnico da Altus em São Leopoldo, RS, ligue para +55 51 3589-9500. Para conhecer os centros de Suporte Técnico da Altus existentes em outras localidades, consulte nosso site www.altus.com.br ou envie um e-mail para altus@altus.com.br. Se o equipamento já estiver instalado, tenha em mãos as seguintes informações ao solicitar assistência:

- Os modelos dos equipamentos utilizados e a configuração do sistema instalado
- O número de série do produto
- A revisão do equipamento e a versão do software executivo, constantes na etiqueta afixada na lateral do produto
- Informações sobre o modo de operação da UCP, obtidas através do programador MasterTool
- O conteúdo do programa da aplicação, obtido através do programador MasterTool
- A versão do programador utilizado

1.6. Mensagens de Advertência Utilizadas neste Manual

Neste manual, as mensagens de advertência apresentarão os seguintes formatos e significados:

PERIGO

Relatam causas potenciais que, se não observadas, levam a danos à integridade física e saúde, patrimônio, meio ambiente e perda da produção.

CUIDADO

Relatam detalhes de configuração, aplicação ou instalação que devem ser seguidos para evitar condições que possam levar a falha do sistema e suas consequências relacionadas.

ATENÇÃO

Indicam detalhes importantes de configuração, aplicação e instalação para obtenção do máximo desempenho operacional do sistema.

2. Descrição Técnica

Este capítulo apresenta todas as características técnicas da UCP NX3030 da Série Nexto.

2.1. Painéis e Conexões

A figura a seguir mostra o painel frontal da UCP.



Figura 3: NX3030

Como se pode observar na figura, na parte superior do painel frontal se encontra o visor gráfico utilizado para mostrar o estado e diagnósticos de todo o sistema, incluindo os diagnósticos específicos de cada módulo. O visor gráfico também oferece um menu fácil de usar que traz ao usuário um modo rápido para ler ou definir alguns parâmetros como: temperatura interna (somente leitura), contraste do visor gráfico, endereço IP para cada interface de rede (somente leitura) e hora local (somente leitura).

Logo abaixo do visor gráfico, encontram-se os 2 LEDs que indicam a ocorrência de diagnóstico e do circuito cão-de-guarda. A Tabela abaixo mostra a descrição dos LEDs. Para maiores informações sobre os estados e significados dos LEDs, consultar a seção [Diagnósticos via LED](#).

LED	Descrição
DG	LED de Diagnóstico
WD	LED de Cão-de-Guarda

Tabela 2: Descrição dos LEDs

As UCPs da Série Nexto contam com duas teclas disponíveis ao usuário. A Tabela abaixo mostra a descrição das teclas. Para maiores informações sobre a tecla de diagnóstico, consulte as seções [One Touch Diag](#) e [Menu Informativo e de Configuração da UCP](#). Para obter mais informações sobre a tecla MS, consulte a seção [Cartão de Memória](#).

Teclas	Descrição
Tecla de Diagnóstico	Tecla situada na parte superior do módulo. Utilizada para visualização dos diagnósticos no visor gráfico ou para navegação no menu informativo e de configurações da UCP.
MS	Tecla localizada no painel frontal. Utilizada para remover o cartão de memória com segurança.

Tabela 3: Descrição das Teclas

No painel frontal estão disponíveis as interfaces de conexão das UCPs da Série Nexto. A Tabela abaixo apresenta uma breve descrição dessas interfaces.

Interfaces	Descrição
NET 1	Conector tipo RJ45 de comunicação no padrão 10/100Base-TX. Permite a comunicação ponto a ponto ou em rede. Para obter mais informações sobre utilização, consulte a seção Configuração das Interfaces Ethernet .
NET 2	Conector tipo RJ45 de comunicação no padrão 10/100Base-TX. Permite a comunicação ponto a ponto ou em rede. Para obter mais informações sobre utilização, consulte a seção Configuração das Interfaces Ethernet .
COM 1	Conector tipo DB9 fêmea para comunicação no padrão RS-232C. Permite a comunicação ponto a ponto. Para obter mais informações sobre utilização, consulte a seção Configuração das Interfaces Seriais .
COM 2	Para obter mais informações sobre utilização, consulte a seção Configuração das Interfaces Seriais .
MEMORY SLOT	Conector para cartão de memória. Permite utilizar um cartão de memória para diferentes tipos de armazenamento de dados como: logs de usuário, a documentação do projeto e arquivos de origem. Para obter mais informações sobre utilização, consulte seção Cartão de Memória .

Tabela 4: Interfaces de Conexão

2.2. Características do Produto

2.2.1. Características Gerais

	NX3030
Ocupação do bastidor	2 posições sequenciais
Fonte de alimentação integrada	Não
Ethernet TCP/IP interface local	2
Interface Serial	2
Interface CAN	Não
Porta USB Host	Não
Interface Cartão de Memória	Sim
Relógio de tempo real (RTC)	Sim Resolução de 1 ms, máx. variação de 2 segundos por dia.
Cão de guarda	Sim
Indicação de estado e diagnóstico	Visor gráfico LEDs Páginas Web de Sistema Memória interna da UCP
Linguagens de programação	Texto Estruturado (ST) Diagrama Ladder (LD) Sequenciamento Gráfico de Funções (SFC) Diagrama de Blocos Funcionais (FBD) Gráfico Contínuo de Funções (CFC)
Tipos de tarefas	Cíclica (periódica) Disparada por evento (interrupção de software) Disparada por evento externo (interrupção de hardware) Contínua (execução livre) Disparada por estado (interrupção de software)
Alterações online	Sim
Número máximo de tarefas	32
Número máximo de barramentos de expansão	24
Suporte a redundância de expansão de barramento	Sim
Número máximo total de módulos de E/S nos barramentos	128
Número máximo de módulos adicionais de interface Ethernet TCP/IP	6
Suporte a redundância de interfaces Ethernet TCP/IP	Sim
Número máximo de redes PROFIBUS-DP (usando módulos mestres PROFIBUS-DP)	4
Suporte a redundância de rede PROFIBUS-DP	Sim
Suporte a redundância (half-clusters)	Sim
Suporte a troca a quente	Sim
Registro de eventos (SOE)	Sim
Protocolo	DNP3
Tamanho máximo da fila de eventos	1000
Desenvolvimento de páginas Web (acessíveis através de protocolo HTTP)	Não

2. DESCRIÇÃO TÉCNICA

	NX3030
One Touch Diag (OTD)	Sim
Electronic Tag on Display (ETD)	Sim

Tabela 5: Características Gerais

Notas:

Relógio de tempo real (RTC): O tempo de retenção, tempo em que o relógio de tempo real continuará a atualizar a data e hora após a desenergização da UCP, é 15 dias para operação a 25 °C. Na temperatura máxima do produto o tempo de retenção é reduzido para 10 dias.

Número máximo de módulos de E/S no barramento: O número máximo de módulos de E/S refere-se a soma de todos os módulos do barramento local e das expansões.

Registro de eventos (SOE): Os tipos de dados encontram-se no DNP3 Device Profile.

2.2.2. Normas e Certificações

Normas e Certificações	
IEC	61131-2: Industrial-process measurement and control - Programmable controllers - Part 2: Equipment requirements and tests 61131-3: Programmable controllers - Part 3: Programming languages
	DNV Type Approval – DNV-CG-0339 (TAA000013D)
CE	2014/30/EU (EMC) 2014/35/EU (LVD) 2011/65/EU and 2015/863/EU (ROHS)
UK CA	S.I. 2016 No. 1091 (EMC) S.I. 2016 No. 1101 (Safety) S.I. 2012 No. 3032 (ROHS)
	UL/cUL Listed – UL 61010-1 UL 61010-2-201 (file E473496)
EAC	TR 004/2011 (LVD) CU TR 020/2011 (EMC)

Tabela 6: Normas e Certificações

2.2.3. Memória

	NX3030
Memória de variáveis de entrada de representação direta (%I)	96 Kbytes
Memória de variáveis de saída de representação direta (%Q)	96 Kbytes
Memória de variáveis de representação direta (%M)	64 Kbytes
Memória de variáveis simbólicas	6 Mbytes
Memória de variáveis retentivas ou persistentes	112 Kbytes
Memória de dados redundantes total	736 Kbytes
Memória de variáveis de entrada de representação direta (%I)	80 Kbytes
Memória de variáveis de saída de representação direta (%Q)	80 Kbytes
Memória de variáveis de representação direta (%M)	64 Kbytes
Memória de variáveis simbólicas	512 Kbytes
Memória de programa	8 Mbytes
Memória de código fonte (backup)	120 Mbytes
Memória de arquivos de usuário	32 Mbytes

Tabela 7: Memória

Notas:

Memória de variáveis de entrada de representação direta (%I): Área onde são alocadas as variáveis de representação direta para o tipo entrada. Variável de representação direta significa que a variável pode ser acessada diretamente na memória utilizando o endereço desejado. Por exemplo: %IB0, %IW100. Variável de entrada de representação direta pode ser utilizada para mapear pontos de entrada analógicos ou digitais. Como referência, 8 pontos de entrada digital podem ser representados por um byte e um ponto de entrada analógica pode ser representado por dois bytes.

Memória de variáveis de saída de representação direta (%Q): Área onde são alocadas todas as variáveis de representação direta para o tipo saída. Variável de representação direta significa que a variável pode ser acessada diretamente na memória utilizando o endereço desejado. Por exemplo: %QB0, %QW100. Variável de saída de representação direta pode ser utilizada para mapear pontos de saída analógicos ou digitais. Como referência, 8 pontos de saída digital podem ser representados por um byte e um ponto de saída analógica pode ser representado por dois bytes. As variáveis de representação direta de saída ainda podem ser configuradas como retentivas, persistentes e/ou redundantes, mas o tamanho total não é alterado em função da configuração.

A UCP NX3030 da Série Nexto permite a definição de uma área de variáveis redundantes inseridas dentro da área de memória de variáveis de saída de representação direta %Q. O subconjunto de tipos de memória de variáveis de representação direta de saída fazem parte do total da memória disponibilizada.

Memória de variáveis de representação direta (%M): Área onde são alocadas as variáveis de representação direta para o tipo marcador. Variável de representação direta significa que a variável pode ser acessada diretamente na memória utilizando o endereço desejado. Por exemplo: %MB0, %MW100.

Memória de variáveis simbólicas: Área onde são alocadas as variáveis simbólicas. As variáveis simbólicas são variáveis IEC criadas em POU's e GVL's durante o desenvolvimento da aplicação, as quais não são endereçadas diretamente na memória. Variáveis simbólicas podem ser definidas como retentivas ou persistentes, neste caso serão utilizadas as áreas de memória de variáveis simbólicas retentiva ou memória de variáveis simbólicas persistente respectivamente. O sistema do CP aloca variáveis nesta área, desta forma o espaço disponível para a alocação de variáveis criadas pelo usuário é inferior ao informado na tabela. A ocupação pelas variáveis de sistema depende das características do projeto (quantidade de módulos, de drivers, etc...), desta forma recomenda-se observar o espaço disponível nas mensagens de compilação da ferramenta MasterTool IEC XE.

Memória de variáveis retentivas ou persistentes: Área onde são alocadas as variáveis retentivas ou persistentes. Os dados retentivos mantêm seus respectivos valores mesmo após um ciclo de desenergização e energização da UCP. Já os dados persistentes mantêm seus respectivos valores mesmo após o download de uma nova aplicação na UCP.

ATENÇÃO

A declaração e utilização de variáveis simbólicas persistentes devem ser realizadas única e exclusivamente através do objeto *Persistent Vars*, o qual pode ser incluído no projeto através da árvore de dispositivos em *Application -> Add Object -> Persistent Variables*. Não deve ser utilizada a expressão *VAR PERSISTENT* no campo de declaração de variáveis das POU's.

A lista completa de quando as variáveis retentivas e persistentes mantêm seus valores e quando o valor é perdido pode ser encontrada na tabela abaixo. Além do tamanho de área persistente declarado nas tabelas acima, estão reservados 44 bytes a mais para armazenar informações sobre as variáveis persistentes (não disponível para uso).

A tabela abaixo mostra o comportamento das variáveis retentivas e persistentes para diferentes situações, em que “-” significa que o valor é perdido e “X” significa que o valor é mantido.

Comando/Operação	VAR	VAR RETAIN	VAR PERSISTENT
Ciclo de energização	-	X	X
Reset a quente	-	X	X
Reset a frio	-	-	X
Reset origem	-	-	-
Remover UCP com fonte integrada do bastidor enquanto energizado	-	X	X
Remover a fonte de alimentação ou uma UCP sem fonte integrada do bastidor enquanto energizado	-	-	-
Download	-	-	X
Alteração online	X	X	X
Limpar tudo	-	-	X
Reset Process (IEC 60870-5-104)	-	X	X

Tabela 8: Comportamento das Variáveis após Evento

Nas versões inferiores ou iguais a 1.5.1.0 para NX3010, NX3020 e NX3030 as memórias retentivas e persistentes simbólicas e de saída de representação direta (%Q) possuíam tamanho máximo fixo. Na tabela abaixo é possível consultar os tamanhos máximos permitidos nas versões antigas das UCPs.

Em versões superiores, as UCPs contam com a funcionalidade de tamanho de memórias retentivas e persistentes flexíveis. Para mais informações do funcionamento consulte a seção [Áreas de Memória Retentiva e Persistente](#).

	NX3030
Memória de variáveis de saída de representação direta retentivas (%Q)	16 Kbytes
Memória de variáveis de saída de representação direta persistentes (%Q)	48 Kbytes
Memória de variáveis simbólicas retentivas	32 Kbytes
Memória de variáveis simbólicas persistentes	16 Kbytes

Tabela 9: Memórias Retentivas e Persistentes em Versões Antigas

No caso do comando de Limpar Tudo, caso a aplicação tenha sido modificada de tal forma que variáveis persistentes tenham sido removidas, inseridas no início da lista ou então tenham tido o seu tipo modificado, o valor destas variáveis será perdido (alertado pela ferramenta MasterTool ao realizar o download). Desta forma recomenda-se que alterações na GVL de variáveis persistentes envolvam somente a inclusão de novas variáveis no final da lista.

Memória de dados redundantes total: Memória de dados redundantes é a quantidade máxima de memória que pode ser utilizada como memória redundante entre duas UCPs que formam o par redundante. Esse valor não se trata de uma memória diferente. Note que a soma de todas as variáveis redundantes (Variáveis de entrada de representação direta, Variáveis de saída de representação direta, Variáveis de representação direta, Variáveis simbólicas, Variáveis simbólicas retentivas, Variáveis simbólicas persistentes) deve ser igual ou menor que a memória de dados redundantes.

Memória de programa: Área da memória que corresponde ao tamanho máximo permitido para a aplicação de usuário. Essa área é compartilhada com a memória de código fonte e com os dados de compilação, sendo a área total a soma de (Memória de programa + Memória de código fonte + Memória de compilação).

Memória de código fonte (backup): Área da memória utilizada como backup do projeto, ou seja, caso o usuário deseje importar o seu projeto, o software MasterTool IEC XE irá buscar as informações necessárias nessa área. É importante salientar que o usuário deve estar atento para não esquecer de atualizar o projeto que está salvo como backup toda vez que enviar a aplicação, evitando que informações sejam perdidas. Essa área é compartilhada com a memória de programa sendo a área total a soma de Memória de programa + Memória de código fonte.

Memória de arquivos de usuário: Essa área da memória é destinada ao armazenamento de arquivos, como: doc, pdf, imagens, entre outros; ou seja, permite a gravação de dados como se fosse um cartão de memória. Maiores informações no capítulo Configuração - Memória de Arquivos de Usuário.

2.2.4. Protocolos

	NX3030	Interface
Protocolo aberto	Sim	COM1 / COM2
MODBUS RTU Mestre	Sim	COM1 / COM2
MODBUS RTU Escravo	Sim	COM1 / COM2
MODBUS TCP Cliente	Sim	NET1 / NET2
MODBUS TCP Servidor	Sim	NET1 / NET2
MODBUS RTU via TCP Cliente	Sim	NET1 / NET2
MODBUS RTU via TCP Servidor	Sim	NET1 / NET2
CANopen Mestre	Não	-
CANopen Escravo	Não	-
CAN low level	Não	-
SAE J-1939	Não	-
OPC DA Servidor	Sim	NET1 / NET2
OPC UA Servidor	Sim	NET1 / NET2
EtherCAT Mestre	Sim	NET1 / NET2
SNMP Agente	Sim	NET1 / NET2
SOE (dados orientados ao evento)	Sim	NET1 / NET2
IEC 60870-5-104 Servidor	Sim	NET1 / NET2
EtherNet/IP Scanner	Sim	NET1 / NET2
EtherNet/IP Adapter	Sim	NET1 / NET2
MQTT Cliente	Sim	NET1 / NET2
SNTP Cliente (para sincronismo do relógio)	Sim	NET1 / NET2
PROFINET Controller	Sim	NET1 / NET2
PROFINET Device	Não	-

Tabela 10: Protocolos

Nota:

PROFINET Controller: Habilitado para uso sem redundância de UCP e em rede simples (sem anel) com até 8 dispositivos. Para aplicações maiores, consultar o suporte técnico.

2.2.5. Interfaces Seriais

2.2.5.1. COM 1

COM 1	
Conector	DB9 fêmea blindado
Interface Física	RS-232C
Sinais de modem	RTS, CTS, DCD
Taxa de Transmissão	200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200 bps
Isolação Lógica para porta serial Porta serial para terra de proteção ⊕	Não isolado 1000 Vac / 1 minuto

Tabela 11: Características da Interface Serial COM 1

2.2.5.2. COM 2

COM 2	
Conector	DB9 fêmea blindado
Interface Física	RS-422 ou RS-485 (dependendo do cabo selecionado)
Direção de Comunicação	RS-422: full duplex RS-485: half duplex
Máx. Transmissores RS-422	11 (1 transmissor e 10 receptores)
Máx. Transmissores RS-485	32
Terminação	Sim (opcional via seleção de cabo)
Taxa de Transmissão	200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200 bps
Isolação Lógica para porta serial Porta serial para terra de proteção ⊕	1000 Vac / 1 minuto 1000 Vac / 1 minuto

Tabela 12: Características da Interface Serial COM 2

Nota:

Meio físico: Dependendo da configuração do cabo usado, é possível escolher o tipo de interface física: RS-422 ou RS-485. A lista dos cabos pode ser encontrada na seção [Produtos Relacionados](#).

Máx. Transmissores RS-422: Refere-se ao número máximo de interfaces RS-422 que podem ser usadas no mesmo barramento.

Máx. Transmissores RS-485: Refere-se ao número máximo de interfaces RS-485 que podem ser usadas no mesmo barramento.

2.2.6. Interfaces Ethernet

2.2.6.1. NET 1

	NET 1
Conector	RJ45 fêmea blindado
Auto crossover	Sim
Máximo Comprimento de Cabo	100 m
Tipo de Cabo	UTP ou ScTP, categoria 5
Taxa de Transmissão	10/100 Mbps
Camada Física	10/100 BASE-TX (Full Duplex)
Camada de Enlace	LLC (Controle de Enlace Lógico)
Camada de Rede	IP (Protocolo de Internet)
Camada de Transporte	TCP (Protocolo de Controle de Transmissão) UDP (Protocolo de Datagrama de Usuário)
Diagnósticos	LEDs - verde (velocidade), amarelo (link/atividade)
Isolação Interfaces Ethernet para lógica e terra	1500 Vac / 1 minuto

Tabela 13: Características da Interface Ethernet NET 1

A Interface NET 1 é a interface utilizada para programação usando a ferramenta MasterTool IEC XE.

2.2.6.2. NET 2

	NET 2
Conector	RJ45 fêmea blindado
Auto crossover	Sim
Máximo Comprimento de Cabo	100 m
Tipo de Cabo	UTP ou ScTP, categoria 5
Taxa de Transmissão	10/100 Mbps
Camada Física	10/100 BASE-TX (Full Duplex)
Camada de Enlace	LLC (Controle de Enlace Lógico)
Camada de Rede	IP (Protocolo de Internet)
Camada de Transporte	TCP (Protocolo de Controle de Transmissão) UDP (Protocolo de Datagrama de Usuário)
Diagnósticos	LEDs - verde (velocidade), amarelo (link/atividade)
Isolação Interfaces Ethernet para lógica e terra	1500 Vac / 1 minuto
Interface Ethernet para Interface Ethernet	1500 Vac / 1 minuto

Tabela 14: Características da Interface Ethernet NET 2

2.2.7. Interface do Cartão de Memória

Os cartões de memória podem ser usados para diferentes tipos de armazenamento de dados como: logs de usuários, documentação de projeto e arquivos fontes.

	Cartão de Memória
Capacidade máxima	32 Gbytes
Capacidade mínima	2 Gbytes
Tipo	MiniSD
Sistema de arquivos	FAT32
Remover cartão de forma segura	Sim, pressionando a tecla MS

Tabela 15: Características da Interface com Cartão de Memória

Notas:

Capacidade máxima: A capacidade do cartão de memória deve ser igual ou inferior a este limite para o correto funcionamento na UCP Nexto, podendo a UCP não reconhecer o cartão ou ocorrer perdas de dados durante transferências.

Capacidade mínima: A capacidade do cartão de memória deve ser igual ou superior a este limite para o seu correto funcionamento na UCP Nexto, podendo a UCP não reconhecer o cartão ou ocorrer perdas de dados durante transferências.

Sistema de arquivos: É recomendado formatar a memória utilizando a própria UCP Nexto, caso contrário poderá ocorrer perda de desempenho no acesso a interface do cartão de memória.

2.2.8. Características Ambientais

	NX3030
Consumo de corrente no barramento da fonte de alimentação	1000 mA
Dissipação	5 W
Temperatura de operação	0 a 60 °C
Temperatura de armazenamento	-25 a 75 °C
Umidade relativa de operação e armazenamento	5% a 96%, sem condensação
Revestimento isolante de circuitos eletrônicos	Sim
Índice de proteção	IP 20
Dimensões do produto (L x A x P)	36,00 x 114,63 x 115,30 mm
Dimensões da embalagem (L x A x P)	44,00 x 122,00 x 147,00 mm
Peso	350 g
Peso com embalagem	400 g

Tabela 16: Características Ambientais

Nota:

Revestimento de circuitos eletrônicos: O revestimento de circuitos eletrônicos protege as partes internas do produto contra umidade, poeira e outros elementos agressivos a circuitos eletrônicos.

2.3. Compatibilidade com Outros Produtos

Para desenvolver uma aplicação para UCPs da Série Nexto, é necessário verificar a versão do MasterTool IEC XE. A tabela a seguir mostra a versão mínima necessária (onde os controladores foram introduzidos) e a respectiva versão de firmware naquele momento:

UCPs da Série Nexto	MasterTool IEC XE	Versão de Firmware
NX3010, NX3020, NX3030	1.00 à 2.09	1.2.1.0 à 1.7.0.14
NX3010, NX3020, NX3030	3.00 ou superior	1.8.3.0 ou superior

Tabela 17: Compatibilidade com Outros Produtos

Além disso, ao longo do roteiro de desenvolvimento do MasterTool IEC XE, alguns recursos podem ser incluídos (como Blocos Funcionais especiais, etc ...), que podem introduzir um requisito da versão mínima do firmware. Durante o download da aplicação, o MasterTool IEC XE verifica a versão do firmware instalada no controlador e, se não atender ao requisito mínimo, exibirá uma mensagem solicitando atualização. A versão mais recente do firmware pode ser baixada no site da Altus e é totalmente compatível com aplicações anteriores.

2.4. Desempenho

O desempenho das UCPs da Série Nexto depende:

- Tempo da Aplicação do Usuário
- Intervalo da Aplicação
- Tempo do Sistema Operacional
- Quantidade de módulos (dados de processo, entradas/saídas, entre outros)

2.4.1. Tempo de Intervalo da MainTask

A configuração de tempo de intervalo da MainTask depende do perfil de projeto selecionado. Para os perfis Simples, Normal, Experiente e Personalizado o intervalo pode ser configurado com valores de 1 ms a 750 ms. Já para o Perfil de Máquina o intervalo pode ser configurado com valores de 1 ms a 100 ms.

2.4.2. Tempos de Aplicação

O tempo de execução de uma aplicação das UCPs da Série Nexto depende das seguintes variáveis:

- Tempo de leitura das entradas (locais e remotas)
- Tempo de execução das tarefas
- Tempo de escrita das saídas (locais e remotas)

É importante ressaltar que o tempo de execução da tarefa “MainTask” será diretamente influenciado pela tarefa de sistema “Configuration”, uma tarefa de alta prioridade, executada periodicamente pelo sistema. A tarefa “Configuration” poderá interromper a “MainTask” e, quando utilizados módulos de comunicação, como, por exemplo, o módulo Ethernet NX5000, o acréscimo de tempo à “MainTask” poderá ser de até 25% do seu tempo médio de execução.

2.4.3. Tempos para Execução de Instruções

A tabela abaixo apresenta o tempo necessário para a execução de diferentes instruções.

Instrução	Linguagem	Variáveis	Tempos da Instrução (μ s)
1000 Contatos	LD	BOOL	6
		INT	43
1000 Divisões	ST	REAL	81
		INT	43
	LD	REAL	81
		INT	43
1000 Multiplicações	ST	REAL	23
		INT	15
	LD	REAL	23
		INT	15

Instrução	Linguagem	Variáveis	Tempos da Instrução (μ S)
1000 Somas	ST	INT	15
		REAL	23
	LD	INT	15
		REAL	23
1000 Laços PID	ST	REAL	< 5000

Tabela 18: Tempos de Instrução

2.4.4. Tempos de Inicialização

As UCPs da Série Nexto possuem tempo de inicialização de 50 segundos, sendo que a tela inicial com o logotipo é apresentada depois de 20 segundos da energização.

2.5. Dimensões Físicas

Dimensões em mm.

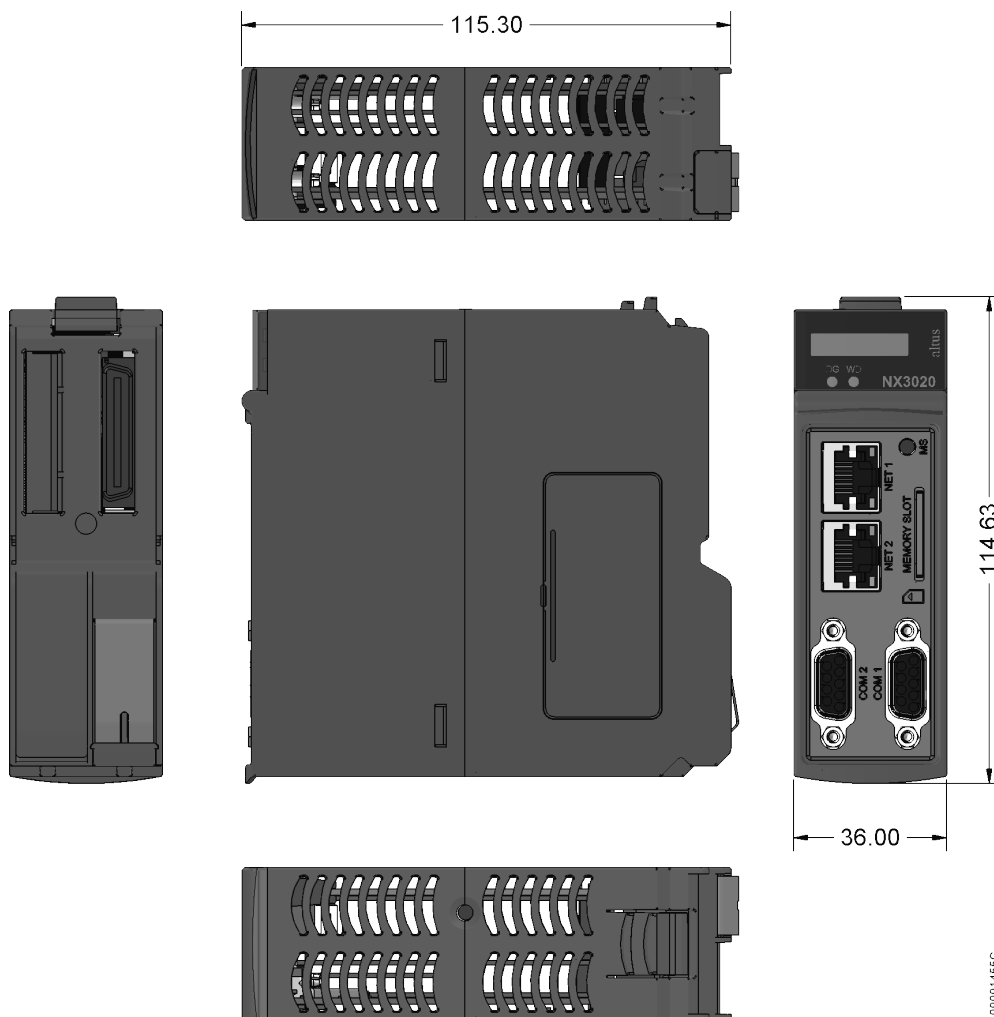


Figura 4: Dimensões Físicas

2.6. Dados para Compra

2.6.1. Itens Integrantes

A embalagem do produto contém os seguintes itens:

- Módulo NX3030

2.6.2. Código do Produto

Os seguintes códigos devem ser usados para compra do produto:

Código	Descrição
NX3030	UCP de alta velocidade, 2 portas Ethernet, 2 canais seriais, interface para cartão de memória, suporte à expansão de barramento e suporte à redundância

Tabela 19: Código do Produto

2.7. Produtos Relacionados

Os seguintes produtos devem ser adquiridos separadamente quando necessário:

Código	Descrição
MT8500	MasterTool IEC XE
AL-2600	Derivador e terminador de rede RS-485
AL-2306	Cabo RS-485 p/ rede MODBUS ou CAN
AL-2319	Cabo RJ45-RJ45
AL-1729	Cabo RJ45-CMDB9
AL-1748	Cabo CMDB9-CFDB9
AL-1752	Cabo CMDB9-CMDB9
AL-1753	Cabo CMDB9-CMDB25
AL-1754	Cabo CMDB9-CFDB9
AL-1761	Cabo CMDB9-CMDB9
AL-1762	Cabo CMDB9-CMDB9
AL-1763	Cabo CMDB9-borneira
AL-1766	Cabo CFDB9-borneira
NX9101	Cartão de 32 GB microSD com adaptador para miniSD e SD
NX9202	Cabo RJ45-RJ45 2 m
NX9205	Cabo RJ45-RJ45 5 m
NX9210	Cabo RJ45-RJ45 10 m
NX9000	Bastidor de 8 Posições
NX9001	Bastidor de 12 Posições
NX9002	Bastidor de 16 Posições
NX9003	Bastidor de 24 Posições
NX8000	Fonte de Alimentação 30 W 24 Vdc

Tabela 20: Produtos Relacionados

Notas:

MT8500: MasterTool IEC XE está disponível em quatro diferentes versões: LITE, BASIC, PROFESSIONAL e ADVANCED. Para maiores informações, favor consultar o Manual de Utilização do MasterTool IEC XE - MU299048.

AL-2600: Este módulo é utilizado para derivação e terminação de uma rede RS-422/485. Para cada nó da rede, deve existir um AL-2600. Os módulos AL-2600 que estiverem nas extremidades da rede devem ser configurados como terminação, exceto quando há um dispositivo com terminação interna ativa, o restante deve ser configurado como derivação.

AL-2306: Cabo blindado de dois pares trançados, sem conectores, para ser utilizado em redes RS-485 ou CAN.

AL-2319: Cabo com dois conectores RJ45 para programação das UCPs da Série Nexto e para comunicação Ethernet ponto-a-ponto com outro dispositivo com interface Ethernet.

AL-1729: Cabo padrão RS-232C com um conector RJ45 e um conector DB9 macho para comunicação entre as UCPs da Série Nexto e outros produtos Altus das Séries DUO, Piccolo e Ponto.

AL-1748: Cabo padrão RS-232C com um conector DB9 macho e um conector DB9 fêmea para comunicação entre UCPs da Série Nexto e outros produtos Altus da Série Cimrex.

AL-1752: Cabo padrão RS-232C com dois conectores DB9 macho para comunicação entre as UCPs da Série Nexto e outros produtos Altus da Série H e Série iX.

AL-1753: Cabo padrão RS-232C com um conector DB9 macho e um conector DB25 macho para comunicação entre as UCPs da Série Nexto e outros produtos Altus da Série H.

AL-1754: Cabo padrão RS-232C com um conector DB9 macho e um conector DB9 fêmea para comunicação entre as UCPs da Série Nexto e outros produtos Altus da Série Exter ou porta Serial padrão RS-232C de um microcomputador.

AL-1761: Cabo padrão RS-232C com dois conectores DB9 macho para comunicação entre as UCPs da Série Nexto e outros produtos Altus da Série AL.

AL-1762: Cabo padrão RS-232C com dois conectores DB9 macho para comunicação entre as UCPs da Série Nexto.

AL-1763: Cabo com um conector DB9 macho e terminais para comunicação entre as UCPs da Série Nexto e produtos com bornes padrão RS-485/RS-422.

AL-1766: Cabo com um conector DB9 fêmea e terminais para comunicação entre as IHM P2 e controladores Nexto Xpress/NX3003.

NX9202/NX9205/NX9210: Cabos utilizados para comunicação Ethernet e para interligar módulos expansores de barramento.

3. Instalação

Este capítulo apresenta os procedimentos necessários para a instalação física das UCPs da Série Nexto, bem como os cuidados que se deve ter com outras instalações existentes no painel elétrico ocupado pelo CP.

CUIDADO

Se o equipamento for utilizado de maneira não especificada neste manual, a proteção fornecida pelo equipamento poderá ser prejudicada.

3.1. Instalação Mecânica

Esse modelo de UCP deve ser inserida na posição 2 do bastidor, logo após o Módulo Fonte de Alimentação. Todas as informações sobre a montagem mecânica e inserção do módulo podem ser encontradas no Manual de Utilização Série Nexto - MU214000.

3.2. Instalação Elétrica

PERIGO

Ao executar qualquer instalação em um painel elétrico, certifique-se de que a fonte de energia esteja DESLIGADA.

A alimentação da UCP é proveniente do Módulo Fonte de Alimentação, o qual fornece tensão às UCPs através da conexão ao bastidor, não necessitando de conexões externas. O aterramento do módulo é realizado através do contato entre a mola de aterramento do módulo e o bastidor.

A figura abaixo ilustra o diagrama elétrico das UCPs da Série Nexto instaladas em um bastidor da Série Nexto.

A disposição dos conectores e bornes na figura é meramente ilustrativa.

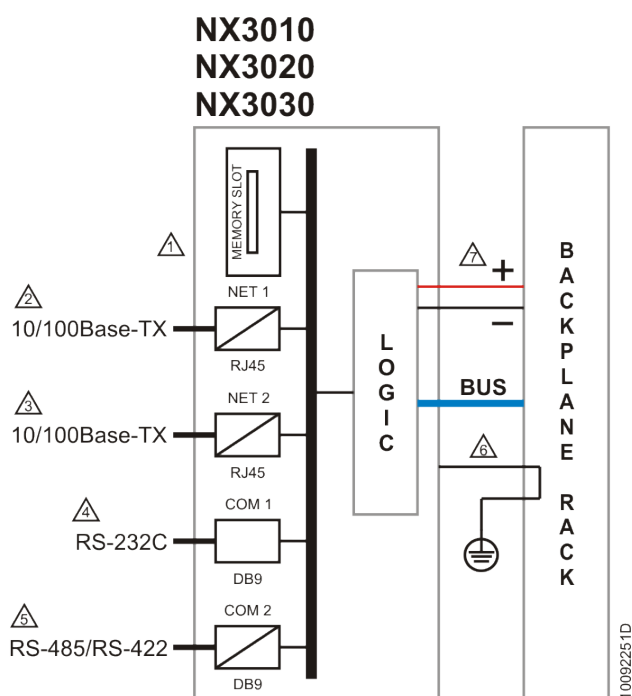










Figura 5: Diagrama Elétrico das UCPs NX3010, NX3020 e NX3030 da Série Nexto

Notas do Diagrama:

-  Interface para cartão de memória.
-  Interface Ethernet padrão 10/100Base-TX para programação, depuração e conexão à rede MODBUS TCP ou outros protocolos.
-  Interface Ethernet padrão 10/100Base-TX para conexão à rede MODBUS TCP ou outros protocolos (somente para NX3020 e NX3030).
-  Interface serial padrão RS-232C para conexão à rede MODBUS RTU ou outros protocolos.
-  Interface serial padrão RS-485/RS-422 para conexão à rede MODBUS RTU ou outros protocolos. A escolha do tipo de interface física depende do cabo utilizado.
-  O módulo é aterrado através dos bastidores da Série Nexto.
-  A alimentação do módulo é proveniente da conexão ao bastidor, não necessitando de conexões externas.
-  Terminal terra de proteção.

3.3. Conexão com a Rede Ethernet

As interfaces isoladas de comunicação NET 1 e NET 2 possibilitam a conexão com uma rede Ethernet, onde a interface NET 1 é mais indicada para comunicação com o MasterTool IEC XE.

A conexão com a rede Ethernet utiliza cabos tipo par trançado (10/100Base-TX), sendo que a detecção da velocidade é realizada automaticamente pela UCP Nexto. Este cabo deve ter uma de suas extremidades ligadas à interface que se pretende utilizar e a outra ao HUB, switch, microcomputador ou outro ponto de rede Ethernet.

3.3.1. Endereço IP

A interface de Ethernet NET 1 é utilizada para comunicação Ethernet e para configurar a UCP, para que isso seja possível, esta vem configurada de fábrica com os seguintes parâmetros:

	NET 1
Endereço IP	192.168.15.1
Máscara de subrede	255.255.255.0
Endereço do Gateway	192.168.15.253

Tabela 21: Configuração de Parâmetros Padrão para a Interface Ethernet NET 1

Os parâmetros Endereço IP e Máscara de Subrede podem ser visualizados no visor gráfico da UCP através do menu de parâmetros, conforme descrito na seção [Menu Informativo e de Configuração da UCP](#).

Inicialmente, deve-se conectar a interface NET 1 da UCP a uma rede ou microcomputador com a mesma subrede para comunicação com o MasterTool IEC XE, onde os parâmetros de rede podem ser alterados. Para maiores detalhes sobre configuração e alteração de parâmetros de rede, verifique a seção [Configuração das Interfaces Ethernet](#).

A interface de Ethernet NET 2 é utilizada para comunicação Ethernet e vem configurada de fábrica com os seguintes parâmetros:

NET 2	
Endereço IP	192.168.16.1
Máscara de subrede	255.255.255.0
Endereço do Gateway	192.168.16.253

Tabela 22: Configuração de Parâmetros Padrão para a Interface Ethernet NET 2

Os parâmetros Endereço IP e Máscara de Subrede podem ser visualizados no visor gráfico da UCP através do menu de parâmetros, conforme descrito na seção [Menu Informativo e de Configuração da UCP](#).

Os parâmetros de rede da interface NET 2 podem ser alterados através do MasterTool IEC XE. Para maiores detalhes sobre configuração e alteração de parâmetros de rede, verifique a seção [Configuração das Interfaces Ethernet](#).

3.3.2. ARP Gratuito

A interface de Ethernet NETx envia espontaneamente pacotes do tipo ARP, em broadcast, informando seu endereço de IP e MAC para todos os dispositivos interligados à rede. Estes pacotes são enviados durante o download de uma nova aplicação pelo software MasterTool IEC XE e na inicialização da UCP, quando a aplicação entra em modo Run.

São disparados 5 comandos ARP com um intervalo inicial de 200 ms, dobrando o intervalo entre cada novo comando disparado, totalizando 3 s. Ex.: Primeiro disparo ocorre no tempo 0, o segundo disparo no tempo 200 ms, o terceiro disparo no tempo 600 ms e assim até o quinto disparo no tempo 3 s.

3.3.3. Instalação do Cabo de Rede

As portas Ethernet das UCPs da Série Nexto, identificadas no painel por NET, possuem pinagem padrão, sendo a mesma utilizada, por exemplo, em computadores pessoais. O tipo de conector, tipo de cabo, nível físico, entre outros detalhes para interligar a UCP ao dispositivo de acesso à rede Ethernet, são definidos na seção [Interfaces Ethernet](#).

A tabela abaixo apresenta o conector RJ45 fêmea das UCPs Nexto, com a identificação e a descrição da pinagem válida para os níveis físicos tipo 10BASE-TE e 100BASE-TX.

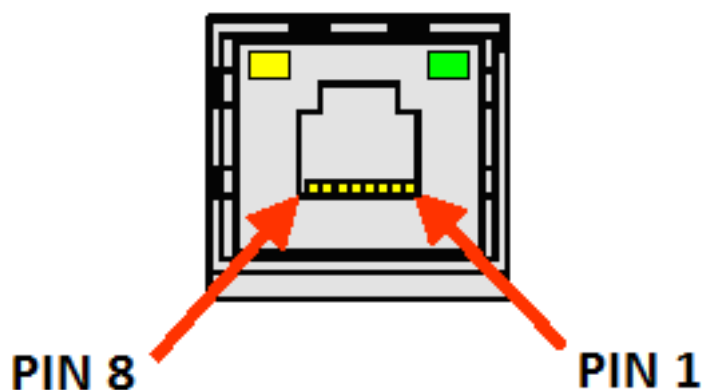


Figura 6: Conector Fêmea RJ45

Pino	Sinal	Descrição
1	TXD +	Transmissão de dados, positivo
2	TXD -	Transmissão de dados, negativo
3	RXD +	Recepção de dados, positivo
4	NU	Não usado
5	NU	Não usado
6	RXD -	Recepção de dados, negativo

Pino	Sinal	Descrição
7	NU	Não usado
8	NU	Não usado

Tabela 23: Pinagem do Conector Fêmea RJ45 - 10BASE-TE e 100BASE-TX

A interface pode ser conectada em uma rede de comunicação através de um hub ou switch, ou então diretamente ao equipamento com o qual irá se comunicar. Neste último caso, devido as UCPs Nexto possuírem a característica de Auto Crossover, não se faz necessária a utilização de um cabo de rede denominado cross-over, o mesmo utilizado para conectar dois computadores pessoais, ponto a ponto, através da porta Ethernet.

É importante ressaltar que entende-se por cabo de rede, um par de conectores RJ45 machos interligados entre si por um cabo UTP ou ScTP, de categoria 5, sob a configuração direta ou cross-over. O mesmo serve para interligar dois dispositivos com porta Ethernet.

Normalmente estes cabos possuem uma trava de conexão que garante uma perfeita conexão entre o conector fêmea da interface e o conector macho do cabo. No momento da instalação, o conector macho do cabo deve ser inserido no fêmea do módulo até que se ouça um som característico ("click"), garantindo a atuação da trava. Para desconectar os mesmos deve-se utilizar a alavanca presente no conector macho.

3.4. Conexão com a Rede Serial RS-232C

A interface não isolada de comunicação COM 1 possibilita a conexão com uma rede RS-232C. A seguir é apresentado o conector DB9 fêmea da UCP Nexto, com a identificação e a descrição dos sinais.

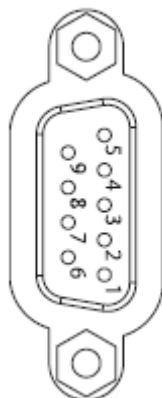


Figura 7: Conector DB9 Fêmea

Pino	Sinal	Descrição
1	DCD	Data Carrier Detect
2	TXD	Transmissão de dados
3	RXD	Recepção de dados
4	-	Não utilizado
5	GND	Ground
6	-	Não utilizado
7	CTS	Clear to Send
8	RTS	Request to Send
9	-	Não utilizado

Tabela 24: Pinagem do Conector DB9 Fêmea COM 1

3.4.1. Comunicação RS-232C

Para conexão com um dispositivo RS-232C, utilizar o cabo apropriado conforme a seção [Produtos Relacionados](#).

3.5. Conexão com a Rede Serial RS-485/422

A interface isolada de comunicação COM 2, possibilita a conexão com uma rede RS-485/422 e utiliza o conector DB9 fêmea para a conexão serial, cuja identificação e a descrição dos sinais é mostrada a seguir.

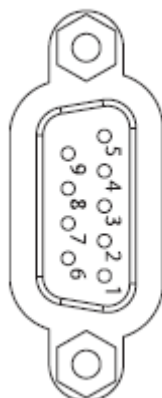


Figura 8: Conector DB9 Fêmea

Pino	Sinal	Descrição
1	-	Não utilizado
2	Term+	Terminação interna, positivo
3	TXD+	Transmissão de dados, positivo
4	RXD+	Recepção de dados, positivo
5	GND	Referência negativa para terminação externa
6	+5V	Referência positiva para terminação externa
7	Term-	Terminação interna, negativo
8	TXD-	Transmissão de dados, negativo
9	RXD-	Recepção de dados, negativo

Tabela 25: COM 1 (NX3004 e NX3005) e COM 2 (NX3010, NX3020 e NX3030)

3.5.1. Comunicação RS-485 sem Terminação

Para conexão a uma rede RS-485 sem terminação, deve-se conectar os terminais identificados do cabo AL-1763 nos respectivos bornes do dispositivo, conforme a tabela abaixo.

Terminais do AL-1763	Sinais na Borneira da UCP
0	Blindagem
1	Não conectado
2	D+
3	D+
4	Não conectado
5	Não conectado
6	Não conectado
7	D-
8	D-

Tabela 26: Conexão da Interface RS-485 sem Terminação

O diagrama da figura abaixo indica como deve ser realizada a conexão dos terminais do AL-1763 na borneira do dispositivo.

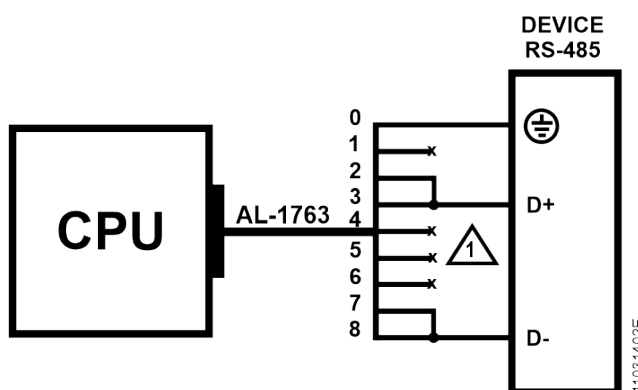


Figura 9: Diagrama de Conexão da Interface RS-485 sem Terminação

Nota do Diagrama:

- Os terminais não conectados devem ser isolados para que não haja contato entre os mesmos.

3.5.2. Comunicação RS-485 com Terminação Interna

Para conexão a uma rede RS-485, utilizando a terminação interna, deve-se conectar os terminais identificados do cabo AL-1763 nos respectivos bornes do dispositivo, conforme a tabela abaixo.

Terminais do AL-1763	Sinais na Borneira da UCP
0	Blindagem
1	D+
2	D+
3	D+
4	Não conectado
5	Não conectado
6	D-
7	D-
8	D-

Tabela 27: Conexão da Interface RS-485 com Terminação Interna

Obs.: As terminações internas disponíveis são do tipo estado seguro em modo aberto.

3. INSTALAÇÃO

O diagrama da figura abaixo indica como deve ser realizada a conexão dos terminais do AL-1763 na borneira do dispositivo.

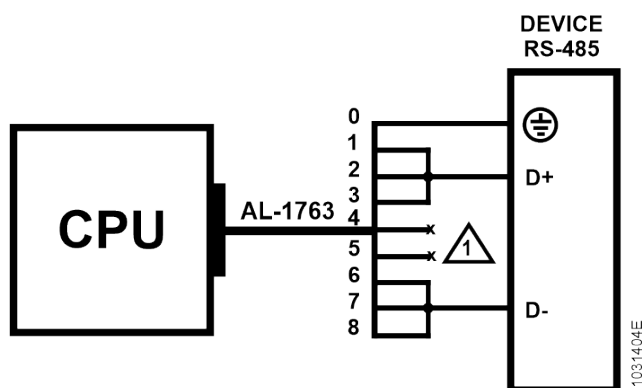


Figura 10: Diagrama de Conexão da Interface RS-485 com Terminação Interna

Nota do Diagrama:

1. Os terminais não conectados devem ser isolados para que não haja contato entre os mesmos.

3.5.3. Comunicação RS-485 com Terminação Externa

Para conexão a uma rede RS-485, utilizando a terminação externa, deve-se conectar os terminais identificados do cabo AL-1763 nos respectivos bornes do dispositivo, conforme a tabela abaixo.

Terminais do AL-1763	Sinais na Borneira da UCP
0	Blindagem
1	Não conectado
2	D+
3	D+
4	0 V
5	+5 V
6	Não conectado
7	D-
8	D-

Tabela 28: Conexão da Interface RS-485 com Terminação Externa

O diagrama da figura abaixo indica como deve ser realizada a conexão dos terminais do AL-1763 na borneira do dispositivo.

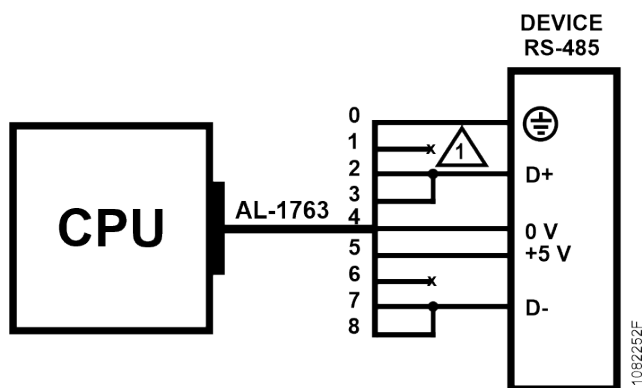


Figura 11: Diagrama de Conexão da Interface RS-485 com Terminação Externa

Nota do Diagrama:

1. Os terminais não conectados devem ser isolados para que não haja contato entre os mesmos.

3.5.4. Exemplo de Ligação de Rede RS-485 com Terminação Externa e Redundância de Mestre

A figura abaixo mostra um exemplo de ligação da rede RS-485 com terminação externa, utilizando duas UCPs Nexto com redundância de half-cluster, como mestre.

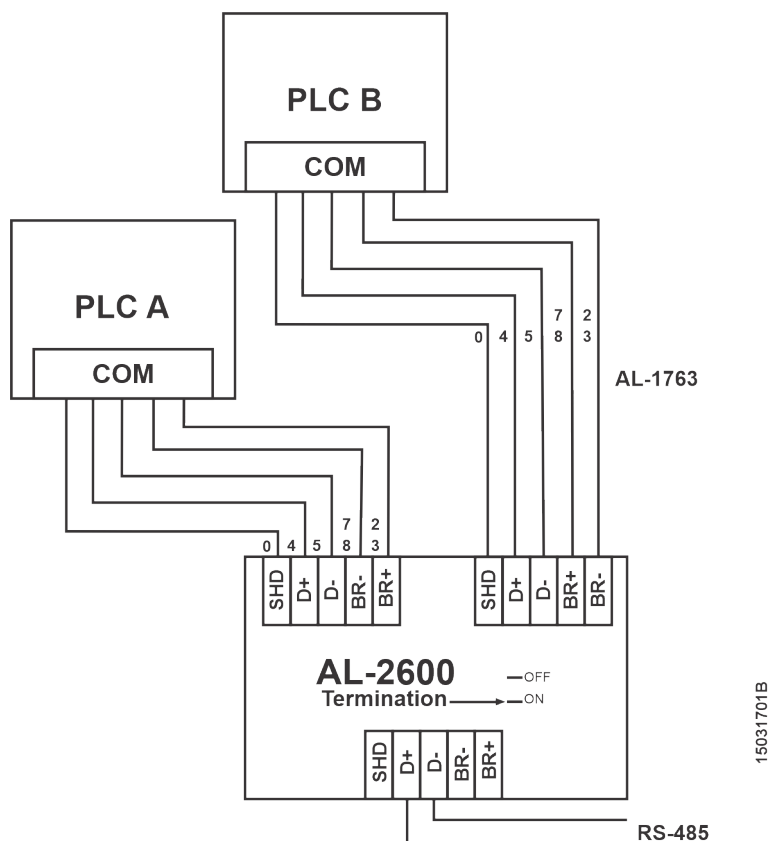


Figura 12: Diagrama de Conexão de Rede RS-485 com Terminação Externa e Redundância de Mestre

3.5.5. Comunicação RS-422 sem Terminação

Para conexão a uma rede RS-422 sem terminação, deve-se conectar os terminais identificados do cabo AL-1763 nos respectivos bornes do dispositivo, conforme a tabela abaixo.

Terminais do AL-1763	Sinais na Borneira da UCP
0	Blindagem
1	Não conectado
2	TX+
3	RX+
4	Não conectado
5	Não conectado

Terminais do AL-1763	Sinais na Borneira da UCP
6	Não conectado
7	TX-
8	RX-

Tabela 29: Conexão da Interface RS-422 sem Terminação

O diagrama da figura abaixo indica como deve ser realizada a conexão dos terminais do AL-1763 na borneira do dispositivo.

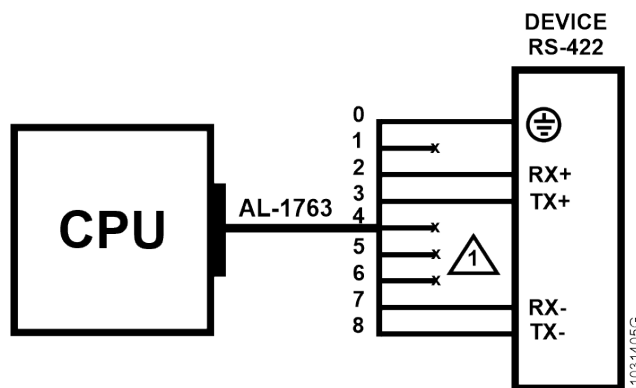


Figura 13: Diagrama de Conexão da Interface RS-422 sem Terminação

Nota do Diagrama:

1. Os terminais não conectados devem ser isolados para que não haja contato entre os mesmos.

3.5.6. Comunicação RS-422 com Terminação Interna

Para conexão a uma rede RS-422, utilizando a terminação interna, deve-se conectar os terminais identificados do cabo AL-1763 nos respectivos bornes do dispositivo, conforme a tabela abaixo.

Terminais do AL-1763	Sinais na Borneira da UCP
0	Blindagem
1	TERM+
2	TX+
3	RX+
4	Não conectado
5	Não conectado
6	TERM-
7	TX-
8	RX-

Tabela 30: Conexão da Interface RS-422 com Terminação Interna

Obs.: As terminações internas disponíveis são do tipo estado seguro em modo aberto.

O diagrama da figura abaixo indica como deve ser realizada a conexão dos terminais do AL-1763 na borneira do dispositivo.

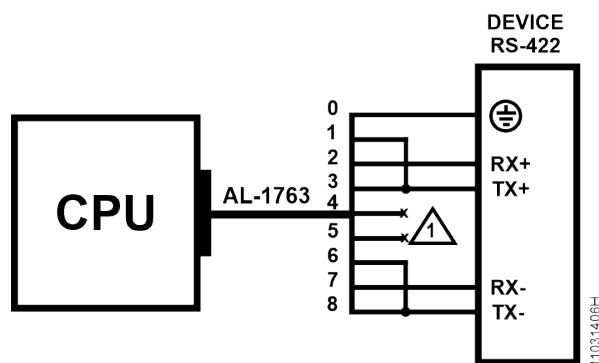


Figura 14: Diagrama de Conexão da Interface RS-422 com Terminação Interna

Nota do Diagrama:

1. Os terminais não conectados devem ser isolados para que não haja contato entre os mesmos.

3.5.7. Comunicação RS-422 com Terminação Externa

Para conexão a uma rede RS-422, utilizando a terminação externa, deve-se conectar os terminais identificados do cabo AL-1763 nos respectivos bornes do dispositivo, conforme a tabela abaixo.

Terminais do AL-1763	Sinais na Borneira da UCP
0	Blindagem
1	Não conectado
2	TX+
3	RX+
4	0 V
5	+5 V
6	Não conectado
7	TX-
8	RX-

Tabela 31: Conexão da Interface RS-422 com Terminação Externa

O diagrama da figura abaixo indica como deve ser realizada a conexão dos terminais do AL-1763 na borneira do dispositivo.

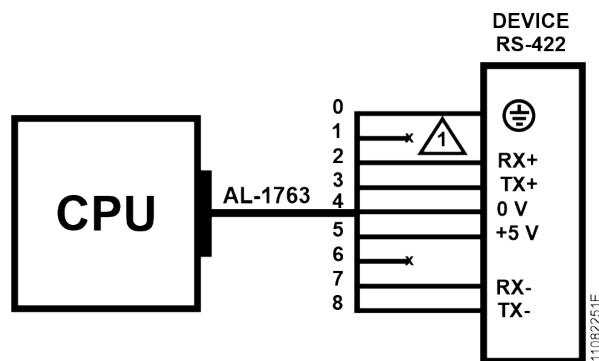


Figura 15: Diagrama de Conexão da Interface RS-422 com Terminação Externa

Nota do Diagrama:

1. Os terminais não conectados devem ser isolados para que não haja contato entre os mesmos.

3.5.8. Exemplo de Rede RS-422

A figura abaixo, mostra um exemplo de utilização da rede RS-422, utilizando a UCP Nexto como mestre, dispositivos escravos com interface RS-422, e solução Altus para derivadores e terminadores.

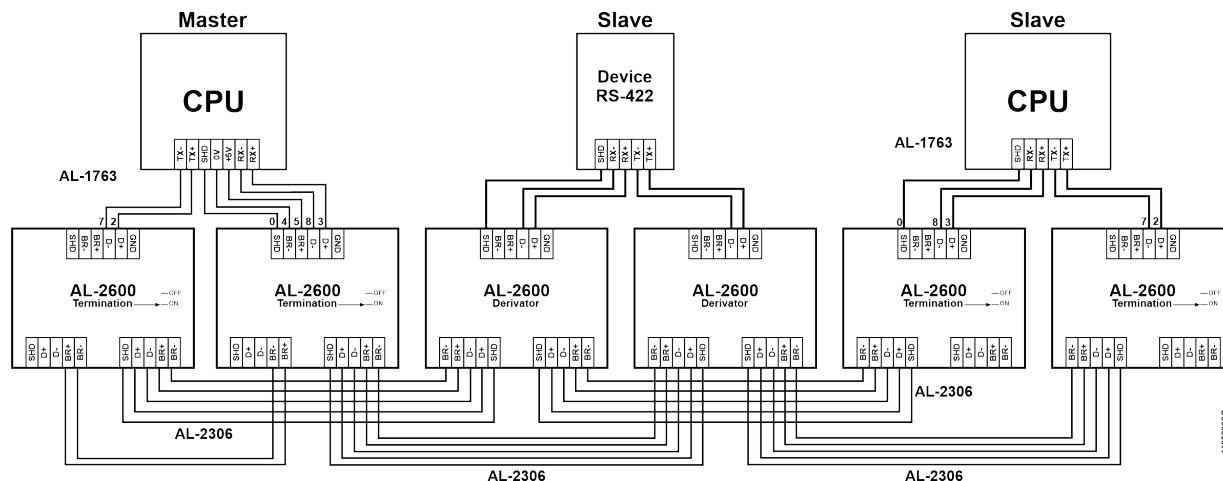


Figura 16: Exemplo de Rede RS-422

Nota do Diagrama:

Os módulos AL-2600 que estão nas extremidades da rede fazem a função de terminadores. Neste caso deve-se configurar as chaves do AL-2600 em Terminação PROFIBUS.

3.6. Instalação do Cartão de Memória

Nesta seção está descrita a maneira como o cartão de memória deve ser inserido na UCP da Série Nexto, sendo que, para maiores informações sobre a utilização do mesmo, consultar a seção [Cartão de Memória](#).

Inicialmente, deve-se ter atenção quanto à posição correta que o cartão de memória deve ser inserido. Um dos cantos se diferencia dos demais e o mesmo deverá ser utilizado como referência para a inserção correta do cartão. Sendo assim, o cartão de memória deverá ser colocado de acordo com o desenho localizado na parte frontal da UCP ou também como mostra a figura abaixo.

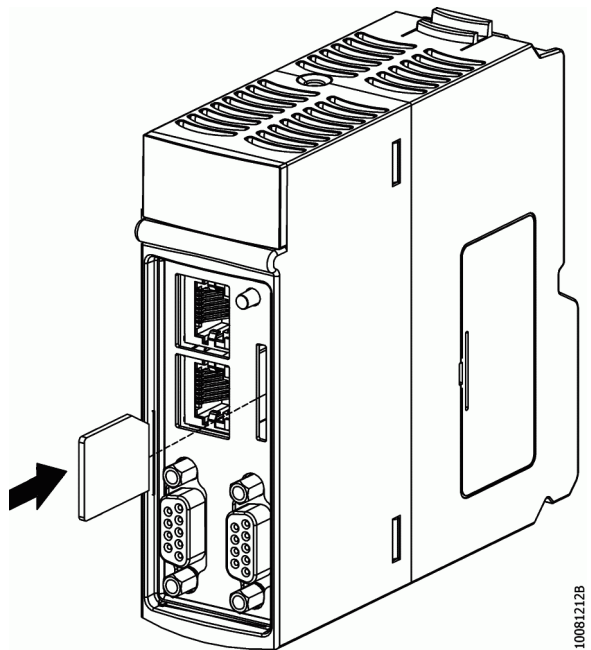


Figura 17: Inserção do Cartão de Memória na UCP

Quando o cartão estiver instalado corretamente aparecerá um símbolo no visor gráfico da UCP. Para remover o cartão com segurança, deve-se pressionar a tecla MS, esperar para que o símbolo do cartão desapareça do visor gráfico e então retirar o cartão. Para que isso seja possível, deve-se pressionar o cartão contra a UCP até que seja gerado um clique. Então, basta soltá-lo e retirá-lo do compartimento, conforme mostra a figura abaixo, pois o mesmo estará solto.

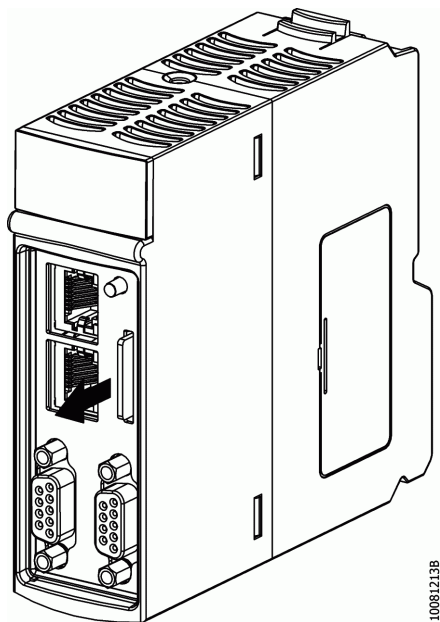


Figura 18: Retirada do Cartão de Memória

3.7. Instalação da Arquitetura

3.7.1. Instalação de Módulos no Bastidor Principal

A Série Nexto possui um exclusivo método para conectar e desconectar módulos do barramento, o qual não exige muito esforço do operador e garante a integridade da conexão. Para maiores informações sobre fixação dos produtos da Série Nexto, favor consultar o Manual de Utilização Série Nexto – MU214000.

3.8. Instalação do Programador

Para realizar a instalação do software de desenvolvimento MasterTool IEC XE é necessário ter em mãos o CD-ROM de distribuição ou efetuar o download do arquivo de instalação no site <http://www.altus.com.br>. Para mais informações e o passo a passo a ser seguido, consulte o Manual de Utilização MasterTool IEC XE – MU299048, no capítulo Instalação.

4. Programação Inicial

O objetivo deste capítulo é auxiliar na programação e configuração das UCPs da Série Nexto, permitindo ao usuário que realize os primeiros passos antes de iniciar a programação de um CP.

As UCPs da Série Nexto utilizam a norma padrão de linguagens IEC 61131-3, que são IL, ST, LD, SFC e FBD, e, além disso, uma linguagem extra, CFC. Pode-se separar essas linguagens em textuais e gráficas. O IL e ST são linguagens textuais e são semelhantes ao Assembly e a linguagem C. O LD, SFC, FBD e CFC são linguagens gráficas. LD usa a representação de blocos de relés e é similar ao diagrama de relés. SFC usa a representação de diagrama de sequência, possibilitando uma maneira fácil de enxergar as sequências de eventos. FBD e o CFC usam um conjunto de blocos funcionais, permitindo uma visão clara das funções exercidas por cada ação.

A programação é feita através da interface de desenvolvimento MasterTool IEC XE (IDE). O MasterTool IEC XE possibilita o uso das seis linguagens no mesmo projeto, assim provendo as melhores características que cada linguagem pode oferecer ao usuário, resultando no desenvolvimento de aplicações mais eficientes, permitindo fácil documentação e manutenção futura.

Para mais informações sobre Programação, consultar Manual de Utilização MasterTool IEC XE - MU299048, Manual de Programação MasterTool IEC XE - MP399048 ou o padrão IEC 61131-3.

4.1. Organização e Acesso à Memória

A Série Nexto utiliza uma inovadora característica de organização e acesso à memória denominada big-endian, ou seja, o byte mais significativo é armazenado primeiramente e sempre será o de menor endereço (Ex.: %QB0 sempre será mais significativo que o %QB1, como na tabela abaixo, que utiliza a palavra CPUNEXTO, onde a letra C é o byte 0 e a letra O é o byte 7).

O acesso à memória deve ser realizado com cuidado pois, variáveis com maior número de bits (WORD, DWORD, LONG), utilizam como índice o byte mais significativo, ou seja, a %QD4 terá, como byte mais significativo, o byte %QB4. Não sendo necessário realizar cálculos para saber qual é a DWORD correspondente a determinados bytes. Organizações little e big endian estão na tabela abaixo.

MSB ← Little-endian → LSB								
BYTE	%QB7	%QB6	%QB5	%QB4	%QB3	%QB2	%QB1	%QB0
	C	P	U	N	E	X	T	O
WORD	%QW6		%QW4		%QW2		%QW0	
	CP		UN		EX		TO	
DWORD	%QD4				%QD0			
	CPUN				EXTO			
LWORD	%QL0							
	CPUNEXTO							
MSB ← Big-endian → LSB								
BYTE	%QB0	%QB1	%QB2	%QB3	%QB4	%QB5	%QB6	%QB7
	C	P	U	N	E	X	T	O
WORD	%QW0		%QW2		%QW4		%QW6	
	CP		UN		EX		TO	
DWORD	%QD0				%QD4			
	CPUN				EXTO			
LWORD	%QL0							
	CPUNEXTO							

Tabela 32: Exemplo Organização e Acesso à Memória

4. PROGRAMAÇÃO INICIAL

SIGNIFICÂNCIA					SOBREPOSIÇÃO					
Bit	Byte	Word	DWord	LWord	Byte	Word	DWord			
%QX0.7	%QB 00	%QW			%QB00	%QW				
%QX0.6										
%QX0.5										
%QX0.4										
%QX0.3										
%QX0.2										
%QX0.1										
%QX0.0										
%QX1.7	%QB 01	00			%QB01	%QW				
%QX1.6										
%QX1.5										
%QX1.4										
%QX1.3										
%QX1.2										
%QX1.1										
%QX1.0	%QD									
%QX2.7	%QB 02	%QW	00		%QB02	%QW	01		00	
%QX2.6										
%QX2.5										
%QX2.4										
%QX2.3										
%QX2.2										
%QX2.1										
%QX2.0	%QD									
%QX3.7	%QB 03	02			%QB03	%QW			01	
%QX3.6										
%QX3.5										
%QX3.4										
%QX3.3										
%QX3.2										
%QX3.1										
%QX3.0	%QL	%QD								
%QX4.7	%QB 04	%QW	00		%QB04	%QW	03			02
%QX4.6										
%QX4.5										
%QX4.4										
%QX4.3										
%QX4.2										
%QX4.1										
%QX4.0	%QD									
%QX5.7	%QB 05	04			%QB05	%QW	04			03
%QX5.6										
%QX5.5										
%QX5.4										
%QX5.3										
%QX5.2										
%QX5.1										
%QX5.0	%QD	%QD								
%QX6.7	%QB 06	%QW	04		%QB06	%QW	05			04
%QX6.6										
%QX6.5										
%QX6.4										
%QX6.3										
%QX6.2										
%QX6.1										
%QX6.0	%QD									
%QX7.7	%QB 07	06			%QB07		06			
%QX7.6										
%QX7.5										
%QX7.4										
%QX7.3										
%QX7.2										
%QX7.1										
%QX7.0										

Tabela 33: Organização e Acesso à Memória

A tabela acima mostra a organização e acesso à memória, exemplificando a significância dos bytes e a disposição dos demais tipos de variável, inclusive a sobreposição.

4.2. Perfis de Projeto

Um Perfil de Projeto no MasterTool IEC XE consiste em um modelo de aplicação em conjunto com um grupo de regras de verificação que orientam o desenvolvimento da aplicação, reduzindo a complexidade da programação. As aplicações podem ser criadas conforme um dos seguintes perfis:

- Simples
- Básico
- Normal
- Experiente
- Personalizado
- Perfil de Máquina

A seleção do perfil é realizada no assistente de criação do projeto. Cada perfil de projeto define um modelo de nomes padronizados para as tarefas e programas, os quais são pré-criados conforme o Perfil de Projeto selecionado. Além disso, na etapa de compilação de projeto (geração de código), o MasterTool IEC XE realiza a verificação de todas as regras definidas pelo perfil selecionado.

As próximas seções detalham as características ou padrões de cada perfil de projeto que seguem uma escala gradual de complexidade. Com base nestas definições, recomenda-se que o usuário sempre procure utilizar o perfil mais simples que atenda as necessidades da sua aplicação, migrando para outro mais sofisticado apenas quando as regras correspondentes estiverem sendo mais entraves ao desenvolvimento do que simplificações didáticas. Cabe ressaltar que a ferramenta de programação permite a alteração do perfil de um projeto existente (consulte a seção Atualização de projeto, no Manual de Utilização MasterTool IEC XE – MU299048), mas caberá ao desenvolvedor realizar qualquer ajuste necessário para que o projeto se torne compatível com as regras do novo perfil selecionado.

ATENÇÃO

No decorrer dos perfis de projeto são nomeados alguns tipos de tarefas, as quais estão descritas na seção Configuração das Tarefas, no Manual de Utilização MasterTool IEC XE – MU299048.

ATENÇÃO

Quando utilizada mais de uma tarefa, o acesso de E/S só deve ser executado no contexto da tarefa principal, MainTask. Caso não possa ser utilizada a opção Habilita atualização de E/S por tarefa, presente a partir da versão 2.01 do MasterTool IEC XE.

4.2.1. Simples

No perfil de projeto Simples, a aplicação possui apenas a tarefa de usuário MainTask. Esta tarefa é responsável pela execução de uma única unidade de programação do tipo Programa denominada MainPrg. Este único programa pode chamar outras unidades de programação do tipo Programa, Função ou Bloco Funcional, mas todo código de usuário será executado exclusivamente pela tarefa MainTask.

Neste perfil, a tarefa MainTask será do tipo cíclica (Cyclic) com prioridade fixada como 13 (treze) e executa exclusivamente o programa MainPrg em um laço contínuo. A tarefa MainTask já está completamente definida e o desenvolvedor precisa criar o programa MainPrg optando por qualquer uma das linguagens da norma IEC 61131-3. Nem sempre é possível converter um programa para outra linguagem, mas sempre é possível criar um novo programa com o mesmo nome em substituição que seja construída em linguagem diversa. A opção padrão do MasterTool IEC XE é utilizar o Projeto MasterTool Padrão associado ao perfil Simples, o qual também inclui o programa MainPrg criado na linguagem escolhida na criação do projeto.

Uma aplicação deste tipo nunca precisa levar em consideração questões como consistência de dados, compartilhamento de recursos ou mecanismos de exclusão mútua.

Tarefa	POU	Prioridade	Tipo	Intervalo	Evento
MainTask	MainPrg	13	Cíclica	20 ms	-

Tabela 34: Tarefa no Perfil Simples

4.2.2. Básico

No perfil de projeto Básico, a aplicação possui uma tarefa de usuário do tipo Contínua denominada MainTask, que executa em um laço contínuo (sem definição de intervalo) com prioridade fixada como 13 (treze). Esta tarefa é responsável pela execução de uma única unidade de programação POU denominada MainPrg. É importante ressaltar que o intervalo pode variar em função da quantidade de tarefas de comunicação utilizadas, pois nesse modo, a tarefa principal é interrompida por tarefas de comunicação.

Este perfil também permite a inclusão de duas tarefas de evento com maior prioridade que podem interromper (preemptar) a MainTask a qualquer momento: a tarefa chamada ExternInterruptTask00 é uma tarefa de evento do tipo Externa com prioridade fixada em 02 (dois); a tarefa chamada TimeInterruptTask00 é uma tarefa de evento do tipo Cíclica com prioridade fixada como 01 (um).

O modelo de projeto Básico, inclui estas três tarefas já completamente definidas conforme apresentado na tabela abaixo. O desenvolvedor precisa apenas criar os programas associados.

Tarefa	POU	Prioridade	Tipo	Intervalo	Evento
MainTask	MainPrg	13	Contínua	-	-
ExternInterruptTask00	ExternInterruptPrg00	02	Externa	-	IO_EVT_0
TimeInterruptTask00	TimeInterruptPrg00	01	Cíclica	20 ms	-

Tabela 35: Tarefas no Perfil Básico

4.2.3. Normal

No perfil de projeto Normal, a aplicação possui uma tarefa de usuário do tipo Cíclica denominada MainTask. Essa tarefa é responsável pela execução de uma única unidade de programação do tipo Programa denominada MainPrg. Este programa e esta tarefa são equivalentes a única tarefa e único programa do perfil Simples, mas aqui a aplicação pode integrar tarefas adicionais de usuário. Estas outras tarefas são denominadas CyclicTask00 e CyclicTask01, cada qual responsável pela execução exclusiva do respectivo programa CyclicPrg<nn>. As tarefas CyclicTask<nn> são sempre do tipo cíclica (Cyclic) com prioridade fixada como 13 (treze), prioridade idêntica a tarefa MainTask. Estes dois tipos formam um conjunto denominado de tarefas básicas, cujos programas associados podem chamar outras POUs do tipo Programa, Função ou Bloco Funcional.

Este perfil pode adicionalmente incluir tarefas de evento com maior prioridade que as tarefas básicas e que consequentemente podem interromper (preemptar) a execução daquelas a qualquer momento.

A tarefa chamada ExternInterruptTask00 é uma tarefa de evento do tipo Externa (Extern) cuja execução é disparada por algum evento externo, tais como variação de um sinal de controle em uma porta serial ou variação de uma entrada digital no barramento do NEXTO. A prioridade desta tarefa é fixada em 02 (dois), sendo responsável pela execução exclusiva do programa ExternInterruptPrg00. A tarefa chamada TimeInterruptTask00 é uma tarefa de evento do tipo Cíclica (Cyclic) com prioridade fixada como 01 (um), sendo responsável pela execução exclusiva do programa TimeInterruptPrg00.

No modelo de projeto Normal, existem cinco tarefas, e suas POUs, já completamente definidas conforme apresentado na tabela abaixo. O desenvolvedor precisa apenas implementar os conteúdos dos programas optando, no assistente, por qualquer uma das linguagens da norma IEC 61131-3. Os intervalos e eventos de disparo das tarefas podem ser configurados pelo desenvolvedor e as tarefas desnecessárias deverão ser eliminadas.

Tarefa	POU	Prioridade	Tipo	Intervalo	Evento
MainTask	MainPrg	13	Cíclica	20 ms	-
CyclicTask00	CyclicPrg00	13	Cíclica	200 ms	-
CyclicTask01	CyclicPrg01	13	Cíclica	500 ms	-
ExternInterruptTask00	ExternInterruptPrg00	02	Externa	-	IO_EVT_0
TimeInterruptTask00	TimeInterruptPrg00	01	Cíclica	20 ms	-

Tabela 36: Tarefas no Perfil Normal

4.2.4. Experiente

O perfil de projeto Experiente inclui as mesmas tarefas básicas, MainTask, CyclicTask<nn>, ExternInterruptTask00 e TimeInterruptTask00, com as mesmas prioridades (13, 02 e 01 respectivamente), mas é uma expansão dos anteriores, pois admite múltiplas tarefas de evento. Ou seja, a aplicação pode incluir várias tarefas ExternInterruptTask<nn> ou TimeInterruptTask<nn> executando os programas ExternInterruptPrg<nn> e TimeInterruptPrg<nn>. As prioridades das tarefas de evento adicionais podem ser livremente selecionadas na faixa de 08 a 12. Neste perfil, além dos programas padrões, cada tarefa pode executar programas adicionais.

Neste perfil de projeto, a aplicação ainda pode incluir a tarefa de usuário FreeTask do tipo Contínua (Freewheeling) com prioridade 31, responsável pela execução do programa FreePrg. Como esta tarefa é de baixa prioridade pode ser interrompida por todas as demais, logo a mesma é capaz de executar códigos que podem ficar bloqueados.

Existem oito tarefas já completamente definidas conforme apresentado na tabela abaixo, bem como os respectivos programas associados criados na linguagem que o usuário selecionar. Os intervalos e eventos de disparo de qualquer tarefa e as prioridades das tarefas de evento podem ser configurados também pelo usuário.

Ao desenvolver a aplicação usando o perfil de projeto Experiente, é necessário um cuidado especial com o escalonamento das tarefas de evento. Caso exista compartilhamento de informações e recursos entre estas tarefas ou entre estas e as tarefas básicas é fortemente recomendável adotar estratégias para garantir a consistência de dados.

Tarefa	POU	Prioridade	Tipo	Intervalo	Evento
MainTask	MainPrg	13	Cíclica	20 ms	-
CyclicTask00	CyclicPrg00	13	Cíclica	200 ms	-
CyclicTask01	CyclicPrg01	13	Cíclica	500 ms	-
ExternInterruptTask00	ExternInterruptPrg00	02	Externa	-	IO_EVT_0
TimeInterruptTask00	TimeInterruptPrg00	01	Cíclica	20 ms	-
ExternInterruptTask01	ExternInterruptPrg01	11	Externa	-	IO_EVT_1
TimeInterruptTask01	TimeInterruptPrg01	09	Cíclica	30 ms	-
FreeTask	FreePrg	31	Contínua	-	-

Tabela 37: Tarefas no Perfil Experiente

4.2.5. Personalizado

O perfil de projeto Personalizado permite ao desenvolvedor explorar todas as potencialidades do Runtime System implantado nas UCPs. Nenhuma das funcionalidades é desabilitada, nenhuma prioridade, associação entre tarefas e programas ou nomenclatura é imposta. A única exceção se faz para a MainTask que deve sempre existir com este nome neste Perfil.

Além das tarefas em tempo real com prioridades 00 a 15, as quais são escalonadas por prioridade, neste perfil também é possível definir tarefas com prioridades menores na faixa 16 a 31. Nesta faixa, é usado o Completely Fair Scheduler (compartilhamento de tempo), o que é necessário para execução de códigos que podem ficar bloqueados (por exemplo, uso de sockets).

O desenvolvedor tem a liberdade para seguir parcialmente ou não a organização definida nos demais perfis de projeto, conforme as particularidades de sua aplicação. Por outro lado, o modelo personalizado, associado a este perfil não necessita elementos pré-definindo como tarefa, programa ou parâmetro, cabendo ao desenvolvedor a criação de todos os elementos que compõe a aplicação.

Tarefa	POU	Prioridade	Tipo	Intervalo	Evento
MainTask	MainPrg	13	Cíclica	20 ms	-
CyclicTask00	CyclicPrg00	13	Cíclica	200 ms	-
CyclicTask01	CyclicPrg01	13	Cíclica	500 ms	-
ExternInterruptTask00	ExternInterruptPrg00	02	Externa	-	IO_EVT_0
TimeInterruptTask00	TimeInterruptPrg00	01	Cíclica	20 ms	-
ExternInterruptTask01	ExternInterruptPrg01	11	Externa	-	IO_EVT_1
TimeInterruptTask01	TimeInterruptPrg01	09	Cíclica	30 ms	-
FreeTask	FreePrg	31	Contínua	-	-

Tabela 38: Tarefas no Perfil Personalizado

4.2.6. Perfil de Máquina

No perfil de Máquina, por padrão, a aplicação possui uma tarefa de usuário do tipo Cíclica denominada MainTask. Esta tarefa é responsável pela execução de uma única unidade de programação do tipo Programa denominada MainPrg. Este programa pode chamar outras unidades de programação do tipo Programa, Função ou Bloco Funcional, mas todo o código do usuário será executado exclusivamente pela tarefa MainTask.

Este perfil se caracteriza por permitir intervalos menores na tarefa MainTask, permitindo a execução mais rápida do código do usuário. Esta otimização é possível pois a MainTask executa também o processamento do barramento. Desta forma diferente de outros perfis, o perfil de máquina não necessita de chaveamento de contexto para o tratamento do barramento, o que reduz o tempo de processamento geral.

Este perfil pode adicionalmente incluir uma tarefa de interrupção de tempo, denominada TimeInterruptTask00, com maior prioridade que a tarefa MainTask e que conseqüentemente pode interromper a execução dela a qualquer momento.

Tarefa	POU	Prioridade	Tipo	Intervalo	Evento
MainTask	MainPrg	13	Cíclica	20 ms	-
TimeInterruptTask00	TimeInterruptPrg00	01	Cíclica	4 ms	-

Tabela 39: Tarefas no Perfil de Máquina

Além disso, este perfil suporta a inclusão de tarefas adicionais associadas à interrupções externas.

4.2.7. Tabela Geral

		Perfis de Projeto					
		Simple	Máquina	Básico	Normal	Experiente	Personalizado
Total de tarefas		01	04	[01..03]	[01..32]	[01..32]	[01..32]
Tarefas por programa		01		01	01	<n>	<n>
Main Task	Tipo	Cíclica	Cíclica	Contínua	Cíclica	Cíclica	Cíclica
	Prioridade	13	13	13	13	13	13
	Quantidade	01	01	01	01	01	01
Time Interrupt Task	Tipo		Cíclica	Cíclica	Cíclica	Cíclica	Cíclica
	Prioridade		01	01	01	01 ou [08..12]	01 ou [08..12]
	Quantidade		[00..01]	[00..01]	[00..01]	[00..31]	[00..31]
Extern Interrupt Task	Tipo		Externa	Externa	Externa	Externa	Externa
	Prioridade		02	02	02	02 ou [08..12]	02 ou [08..12]
	Quantidade		[00..01]	[00..01]	[00..01]	[00..31]	[00..31]
Ciclic Task	Tipo				Cíclica	Cíclica	Cíclica
	Prioridade				13	13	13
	Quantidade				[00..31]	[00..31]	[00..31]
Free Task	Tipo					Contínua	Contínua
	Prioridade					31	31
	Quantidade					[00..01]	[00..01]
Event Task	Tipo						Evento
	Prioridade						<n>
	Quantidade						[00..31]

Tabela 40: Tabela Geral de Perfis x Tarefas

ATENÇÃO

Os nomes sugeridos para as POU's associadas às tarefas não são consistidos. Os mesmos podem ser substituídos desde que sejam substituídos também nas configurações das tarefas.

4.2.8. Número Máximo de Tarefas

O número máximo de tarefas que o usuário poderá criar somente está definido para o perfil Personalizado, ou seja, o único que tem essa permissão. Os demais já têm as suas tarefas criadas e configuradas. No entanto, as tarefas que forem criadas precisam utilizar os prefixos, de acordo com o tipo de cada uma das tarefas, a seguir: CyclicTaskxx, TimeInterruptTaskxx, ExternInterruptTaskxx, onde xx representa o número da tarefa que está sendo criada.

A tabela abaixo descreve a quantidade máxima de tarefas IEC por UCP e perfil de projeto, sendo que as instâncias de protocolo também são consideradas tarefas de comunicação pela UCP.

	Tipo de Tarefa	NX3030					
		S	B	N	E	P	M
Configuration Task (Tarefa WHSB)	Cíclica	1	1	1	1	1	0
Tarefas de Usuário	Cíclica	1	1	31	31	31	2
	Disparada por Evento	0	0	0	0	31	0
	Disp. Evento Externo	0	1	0	30	31	0
	Contínua	0	1	0	1	31	0
	Disparada por Estado	0	0	0	0	31	0
NETs – Instâncias Cliente ou Servidor	Cíclica	16					
COM (n) – Instâncias Mestre ou Escravo	Cíclica	1					
TOTAL		32					

Tabela 41: Número Máximo de Tarefas IEC NX3030

Notas:

Legenda dos Perfis: As letras S, B, N, E, P e M corresponde respectivamente aos Perfis Simples, Básico, Normal, Experiência, Personalizado e Máquina.

Valores: Os números definidos para cada tipo de tarefa representam os valores máximos permitidos.

Tarefa WHSB: A tarefa WHSB que é uma tarefa de sistema deve ser considerada para que não seja ultrapassado o valor total.

NETs - Instâncias Cliente ou Servidor: O valor máximo definido considera todas as interfaces Ethernet do sistema, ou seja, inclui os módulos de expansão, quando estes são aplicados. Exemplos para esse tipo de tarefa são as instâncias do Protocolo MODBUS.

COM (n) - Instâncias Mestre ou Escravo: O "n" representa o número da interface serial, ou seja, mesmo com módulos de expansão, o valor da tabela será o máximo por interface. Exemplos para esse tipo de tarefa são as instâncias do Protocolo MODBUS.

Total: O valor total não representa a soma de todas as tarefas por perfil, mas o valor máximo permitido por UCP. Então, o usuário poderá criar vários tipos de tarefas, desde que o número estabelecido para cada uma e o valor total, não sejam ultrapassados.

4.3. Configurando a UCP

A configuração da UCP Nexto está localizada na árvore de dispositivos, conforme mostra a figura abaixo, e pode ser acessada através de um duplo clique no objeto correspondente. Nesta tela é possível configurar a área de diagnósticos, a área de memória retentiva e persistente, modo de troca a quente, entre outros parâmetros, conforme descrito na seção [Configuração da UCP](#).

4. PROGRAMAÇÃO INICIAL

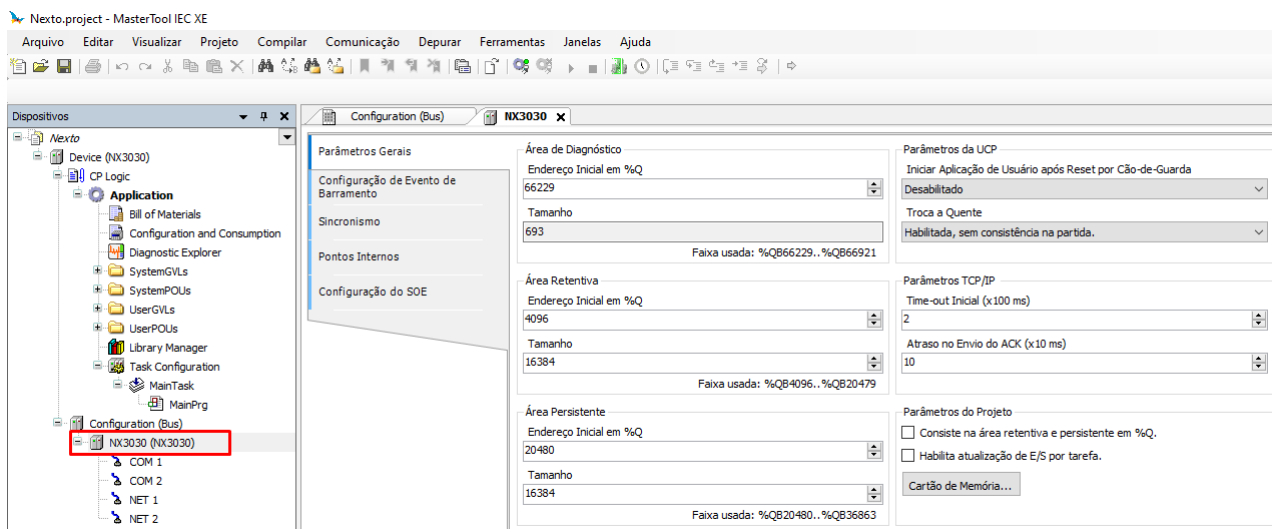


Figura 19: Configuração da UCP

Além disso, clicando duas vezes sobre o ícone NET 1 da UCP é possível realizar a configuração da interface Ethernet através da qual será realizada a comunicação entre a UCP e o MasterTool IEC XE.

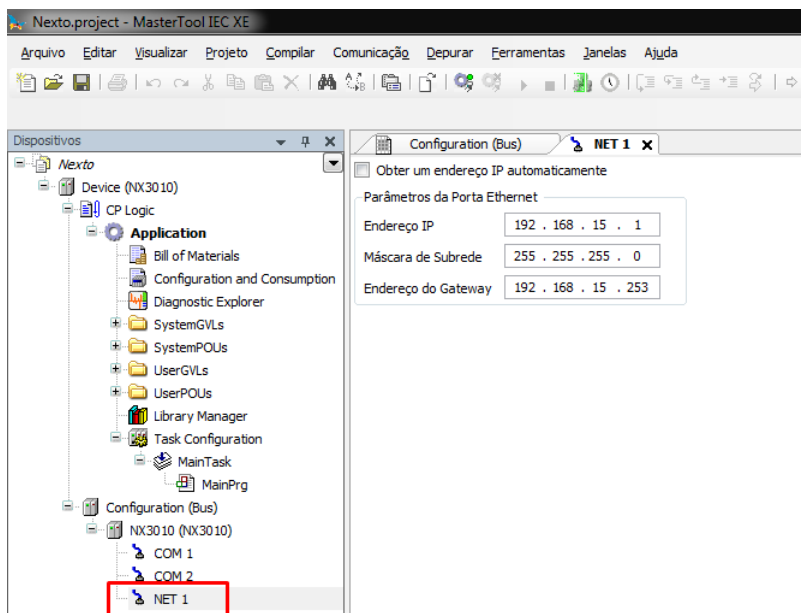


Figura 20: Configurando a Porta de Comunicação da UCP

As configurações de rede definidas nesta tela serão aplicadas no dispositivo somente quando for realizado o envio da aplicação (download), cujo procedimento é descrito a seguir nas seções [Localizando o Dispositivo](#) e [Login](#).

4.4. Bibliotecas

Existem diversos recursos da ferramenta de programação que estão disponíveis através de bibliotecas. Sendo assim, os mesmos devem ser inseridos no projeto para que a sua utilização seja possível. O procedimento de inserção, bem como mais informações sobre as bibliotecas disponíveis podem ser encontrados no Manual de Programação IEC 61131 – MP399048, capítulo Bibliotecas.

4.5. Inserindo uma Instância de Protocolo

As UCPs da Série Nexto, conforme a seção [Protocolos](#), oferecem diversos protocolos de comunicação. Exceto as comunicações OPC DA e OPC UA, que possuem um procedimento de configuração diferente, para inserir um protocolo basta clicar com o botão direito sobre a interface de comunicação desejada, adicionar o dispositivo e finalmente configurar a instância do protocolo conforme descrito na seção [Configuração de Protocolos](#). A seguir apresenta-se um exemplo.

4.5.1. MODBUS Ethernet

O primeiro passo para configurar o MODBUS Ethernet (Cliente neste exemplo), é incluir a instância na NET desejada (nesse caso selecionamos a NET 1, pois a UCP NX3010 possui uma interface Ethernet). Clicar com o botão direito sobre a NET e selecionar *Acrescentar Dispositivo...*, conforme mostra a figura abaixo.

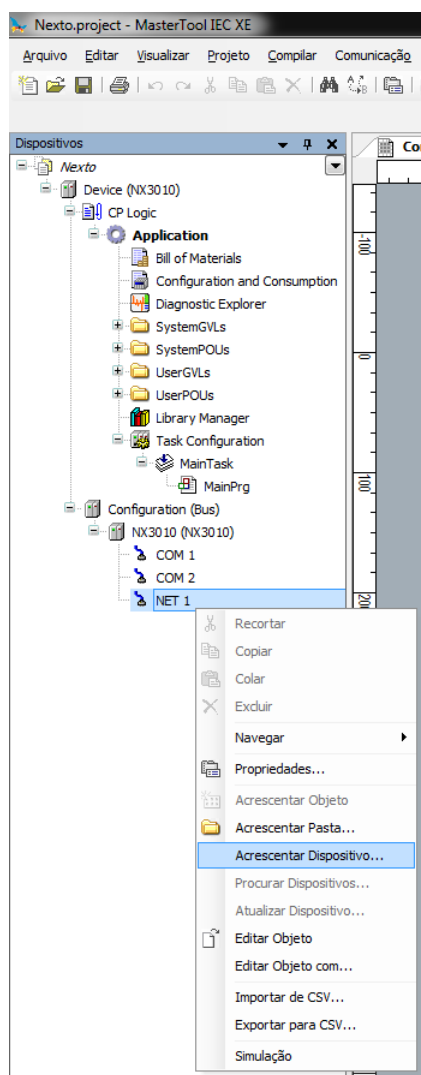


Figura 21: Adicionando a Instância

Após irá surgir na tela os protocolos disponíveis ao usuário. Neste menu define-se o modo de configuração do protocolo. Selecionando a opção *MODBUS Symbol Client*, para configuração por Mapeamento Simbólico, ou *MODBUS Client*, para configuração por Representação Direta (%Q). A seguir, clicar em *Acrescentar*, conforme mostra a figura abaixo.

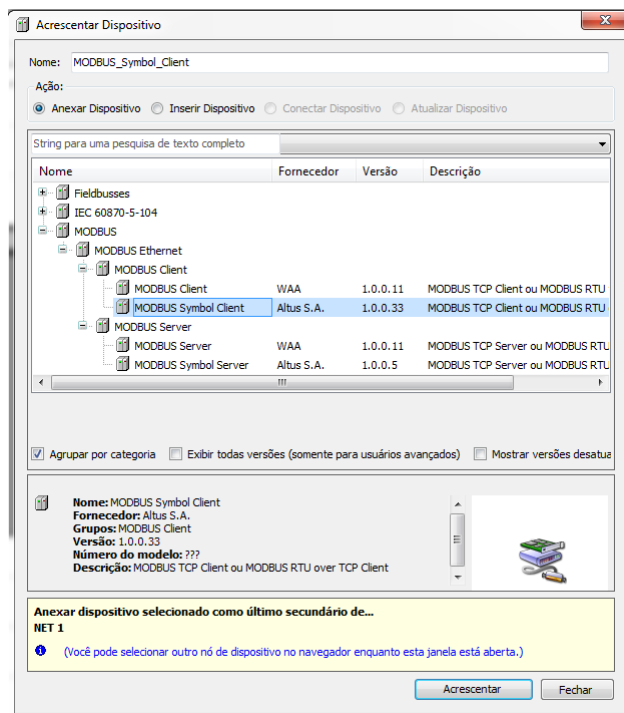


Figura 22: Selecionando o Protocolo

4.6. Localizando o Dispositivo

Para estabelecer a comunicação da UCP com a ferramenta MasterTool IEC XE, é necessário em primeiro lugar localizar e selecionar o dispositivo desejado. A configuração desta comunicação está localizada no objeto *Device* na árvore de dispositivos, na aba *Configurações de Comunicação*. Nesta tela, ao selecionar o *Gateway* e clicar no botão *Mapear Rede*, o software MasterTool IEC XE realiza a busca por dispositivos e apresenta as UCPs encontradas na rede da interface Ethernet da estação onde está sendo executada a ferramenta.

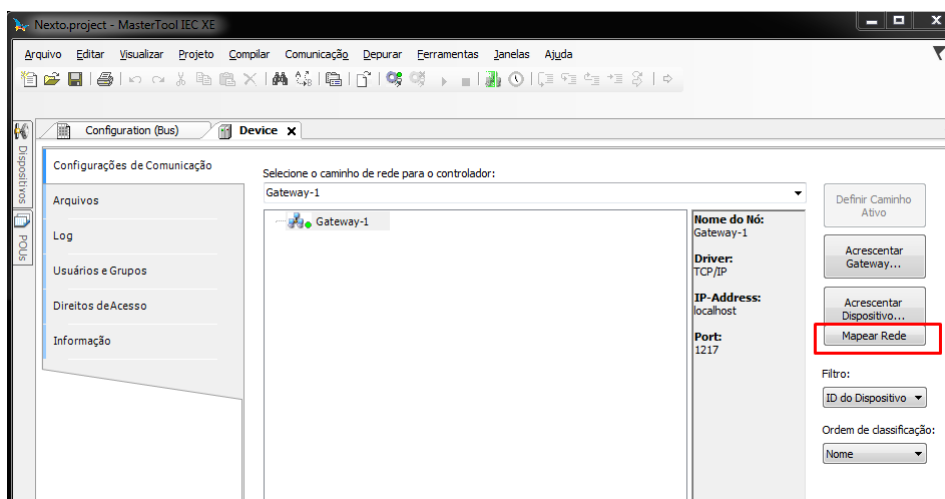


Figura 23: Localizando a UCP

Caso não haja um gateway previamente configurado, o mesmo pode ser adicionado através do botão *Acrescentar Gateway*, mantendo endereço IP padrão *localhost* para utilizar o gateway da estação ou alterando o endereço IP para utilizar o gateway interno do dispositivo.

Em seguida, basta selecionar a UCP desejada e clicar na opção *Definir Caminho Ativo*. Esta ação seleciona o dispositivo e informa ao configurador qual UCP deve ser utilizada para comunicar e enviar o projeto.

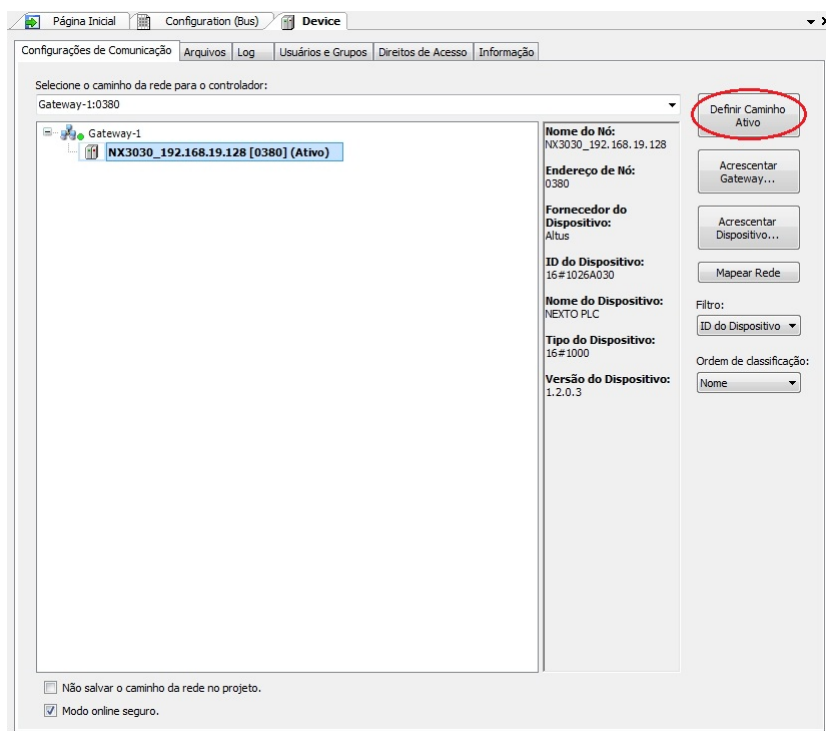


Figura 24: Selecionando a UCP

Adicionalmente, o usuário pode alterar o nome padrão do dispositivo que é exibido. Para isso, deve clicar com o botão direito do mouse sobre o dispositivo desejado e selecionar a opção *Alterar Nome do Dispositivo*. Após uma mudança de nome, o dispositivo não voltará ao nome padrão em nenhuma circunstância.

Caso a configuração Ethernet da UCP que deseja-se conectar esteja em uma rede diferente da interface Ethernet da estação, o MasterTool IEC XE não conseguirá localizar o dispositivo. Neste caso, recomenda-se a utilização do comando *Easy Connection* disponível no menu Comunicação da ferramenta.

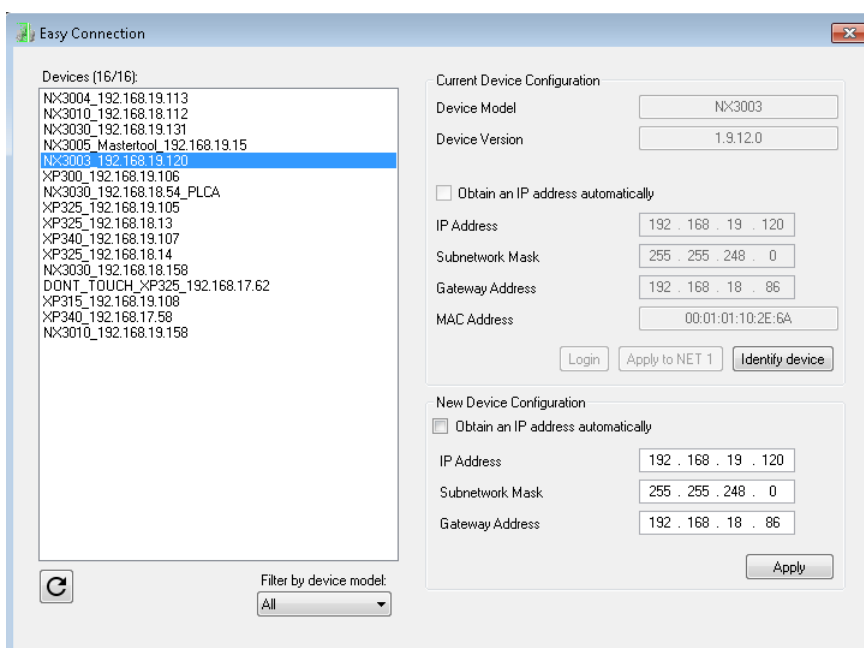


Figura 25: Easy Connection

Este comando realiza uma comunicação em nível MAC com a interface NET 1 do dispositivo, permitindo alterar permanen-

temente a configuração da interface Ethernet da UCP, independentemente da configuração IP da estação e daquela previamente existente no dispositivo. Com isso, é possível configurar o dispositivo para que fique na mesma rede da interface Ethernet da estação onde está sendo executada a ferramenta MasterTool IEC XE, permitindo localizar e selecionar o dispositivo para comunicação. A descrição completa do comando *Easy Connection* pode ser encontrada no Manual de Utilização do MasterTool IEC XE código MU299048.

4.7. Login

Após compilar a aplicação e corrigir todos os erros encontrados, é o momento de enviar o projeto para a UCP. Para isto, basta executar o comando de *Login* localizado no menu *Comunicação* no software MasterTool IEC XE conforme mostra o exemplo da figura a seguir. Essa operação pode levar alguns segundos, dependendo do tamanho do arquivo gerado.

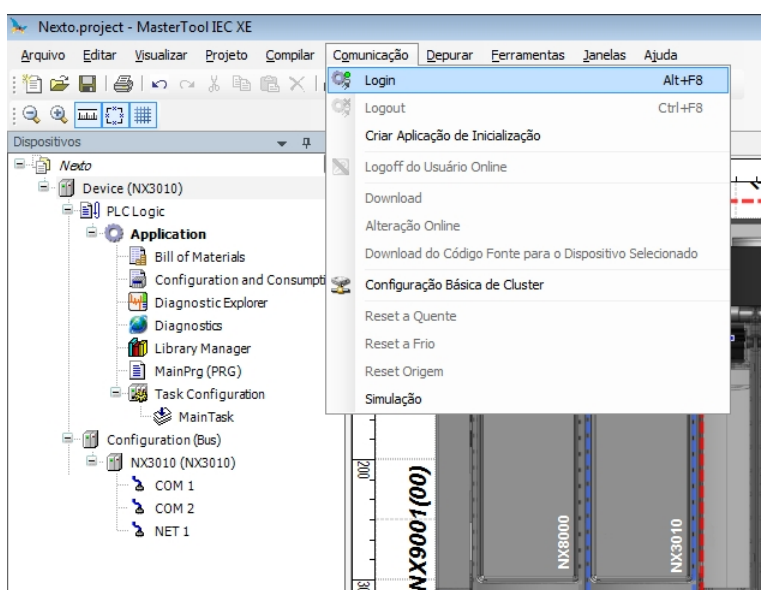


Figura 26: Enviando o Projeto para a UCP

Após a execução do comando, poderão surgir algumas mensagens de interface com o usuário, as quais são apresentadas devido a diferença entre um projeto antigo e o que está sendo enviado ou, simplesmente, alteração no valor de alguma variável.

Caso a configuração Ethernet do projeto seja diferente daquela contida no dispositivo, poderá haver a interrupção da comunicação no final do processo de download quando a nova configuração é aplicada. Portanto, a seguinte mensagem de alerta será apresentada, perguntando ao usuário se deve proceder ou não com esta operação.

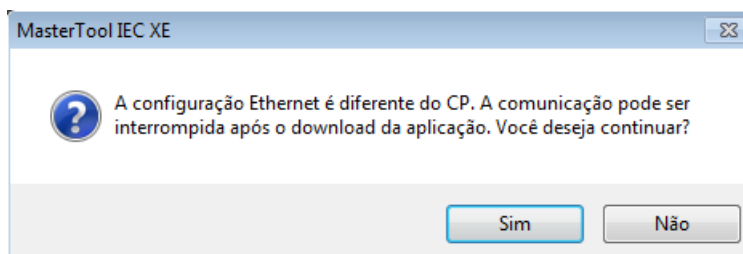


Figura 27: Aviso de configuração IP

Caso não exista aplicação na UCP, a seguinte mensagem será apresentada.

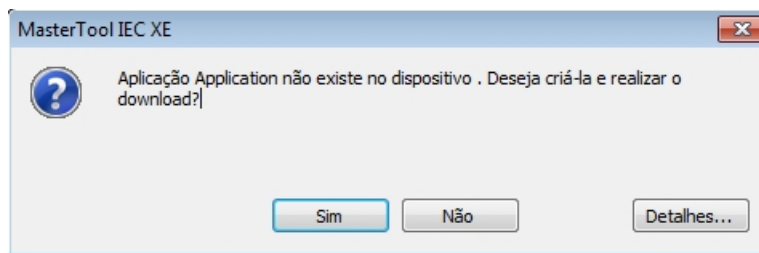


Figura 28: Sem aplicação no dispositivo

Caso já exista uma aplicação na UCP, dependendo das diferenças entre os projetos, serão disponibilizadas as seguintes opções:

- **Login com alteração online:** executar o login e enviar o novo projeto sem que a aplicação atual da UCP seja parada (ver item [Modo Run](#)), atualizando as alterações quando um novo ciclo é executado.
- **Login com Download:** executar o login e enviar o novo projeto com a UCP parada (ver item [Modo Stop](#)). Quando a aplicação for iniciada, as atualizações já terão sido realizadas.
- **Login sem alteração:** executa o login sem enviar o novo projeto.

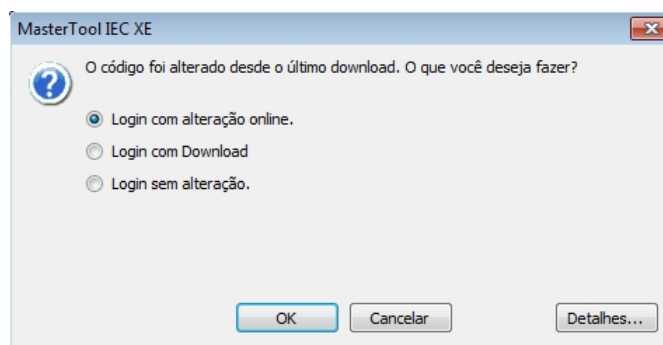


Figura 29: Atualização do Projeto na UCP

ATENÇÃO

Nas alterações online não é permitido associar mapeamentos de variáveis simbólicas de uma lista de variáveis globais (GVL) e utilizar essas variáveis em outra lista de variáveis globais (GVL).

Caso a nova aplicação tenha sofrido alterações de configuração, a alteração online não será possível. Neste caso, o MasterTool IEC XE solicita ao usuário se o login deve ser executado como download (parando a execução da aplicação) ou se a operação deve ser cancelada, conforme mostra a figura a seguir.

Obs.: O botão *Detalhes...* apresenta as modificações realizadas na aplicação.

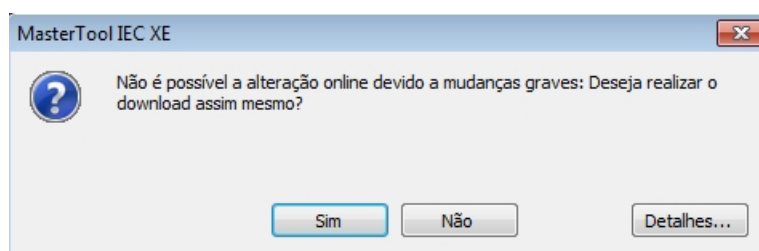


Figura 30: Alteração de Configuração

Por fim, ao final deste processo o MasterTool IEC XE oferece a opção de realizar a transferência (download) do código fonte para a memória interna do dispositivo, conforme mostra a figura a seguir.

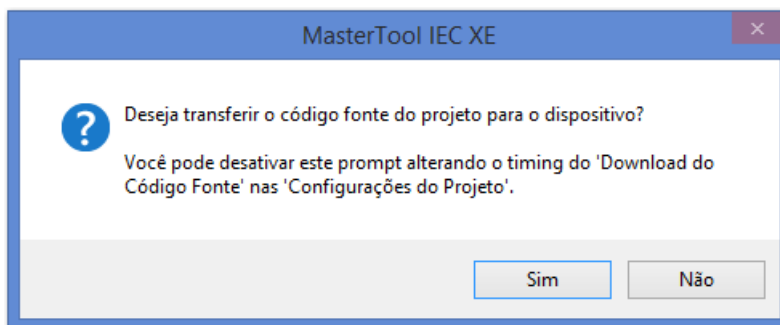


Figura 31: Download de código fonte

A transferência do código fonte é fundamental para permitir a futura recuperação do projeto e a realização de modificações sobre a aplicação que está carregada no dispositivo.

4.8. Modo Run

Logo após enviar o projeto para a UCP, a aplicação não será executada imediatamente (exceto no caso de uma alteração online). Para que isto aconteça, o comando Iniciar deve ser executado. Desta forma, o usuário pode controlar a execução da aplicação enviada para a UCP, permitindo que valores iniciais sejam pré-configurados para que sejam utilizados na UCP no primeiro ciclo.

Para executar este comando, basta acessar o menu *Depurar* e clicar em *Iniciar*, conforme mostra a figura abaixo.

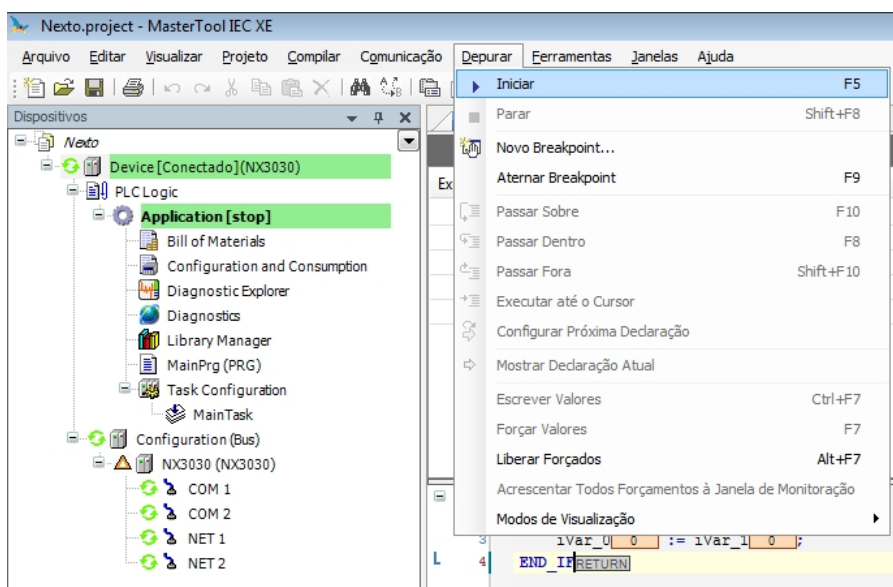


Figura 32: Iniciando a Aplicação

A figura abaixo mostra a aplicação em execução. Caso seja selecionada a aba de uma POU, as variáveis criadas serão listadas em uma janela de monitoração, na qual valores podem ser visualizados e forçados pelo usuário.

4. PROGRAMAÇÃO INICIAL

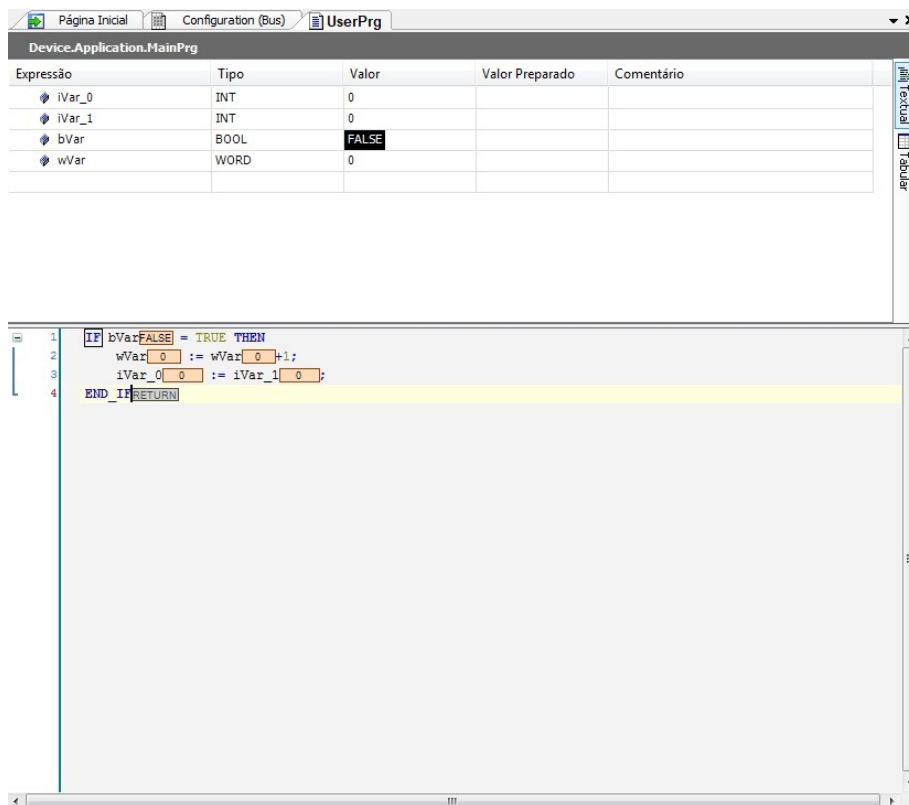


Figura 33: Programa em Execução

Caso a UCP já tenha uma aplicação de inicialização gravada internamente, ela entra automaticamente em Modo Run quando o sistema é energizado, sem a necessidade de realizar o comando online através do MasterTool IEC XE.

4.9. Modo Stop

Para interromper a execução da aplicação, o usuário deve executar o comando *Parar*, disponível no menu *Depurar*, conforme mostra a figura abaixo.

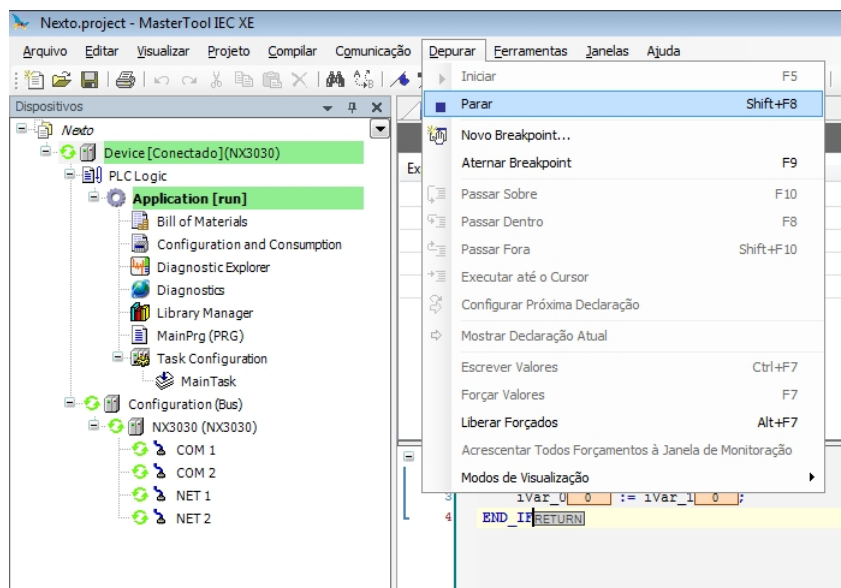


Figura 34: Parando a Aplicação

Caso a UCP seja inicializada sem aplicação gravada, ela automaticamente entra em Modo Stop, assim como quando ocorre uma exceção de software.

4.10. Escrita e Forçamento de Variáveis

Após efetuar um Login em um CP, o usuário poderá escrever ou forçar valores em variáveis do projeto.

O comando de escrita (*CTRL + F7*) escreve um valor em uma variável, podendo este valor ser sobrescrito pelas instruções executadas na aplicação.

Já um comando de escrita forçada (*F7*) irá escrever um valor na variável, sem permitir que este valor seja alterado até que sejam liberadas as variáveis forçadas.

É importante ressaltar que, quando utilizados os protocolos MODBUS RTU Escravo e MODBUS Ethernet Servidor e a opção *Somente Leitura* das relações configuradas não estiver selecionada, o comando de escrita forçada (*F7*) deve ser realizado sobre as variáveis disponíveis na janela de monitoração, pois o comando de escrita (*CTRL + F7*) deixa as variáveis serem sobrescritas quando realizadas novas leituras.

ATENÇÃO

O forçamento de variáveis pode ser realizado somente em modo Online. Variáveis de diagnóstico não podem ser forçadas, apenas escritas, pois diagnósticos são providos pelo controlador e sobrescritos pelo mesmo.

Quando for efetuada uma escrita forçada em uma variável redundante do CP Ativo, a MainTask da aplicação sofrerá um impacto em seu tempo de execução, tanto no CP Ativo, quanto no CP Reserva. Isto porque os dois Half-Clusters irão trocar a cada ciclo informações sobre as variáveis forçadas. Portanto, quando for forçar variáveis em um sistema redundante, deve-se considerar o acréscimo que pode ter a execução da tarefa. A tabela abaixo exemplifica em quanto será acrescida, em média, a execução da MainTask quando isto ocorrer:

Tempo de Execução	CP Ativo			CP Reserva		
	50 ms	100 ms	200 ms	50 ms	100 ms	200 ms
Acréscimo com 10 forçamentos	2,4 %	2,2 %	1,7 %	4,0 %	3,4 %	2,0 %
Acréscimo com 50 forçamentos	12,0 %	9,2 %	6,0 %	18,0 %	12,0 %	8,0 %
Acréscimo com 128 forçamentos	26,0 %	21,0 %	16,0 %	56,0 %	34,0 %	22,5 %

Tabela 42: Influência do Forçamento de Variáveis em um CP Redundante

ATENÇÃO

Quando um controlador está com variáveis forçadas e é desenergizado, na próxima inicialização as variáveis perderão o forçamento. O limite de forçamentos para todos os modelos de controladores Nexto é de 128 variáveis.

4.11. Logout

Para encerrar a comunicação online com a UCP, deve ser utilizado comando *Logout*, localizado no menu *Comunicação*, conforme mostra a figura abaixo.

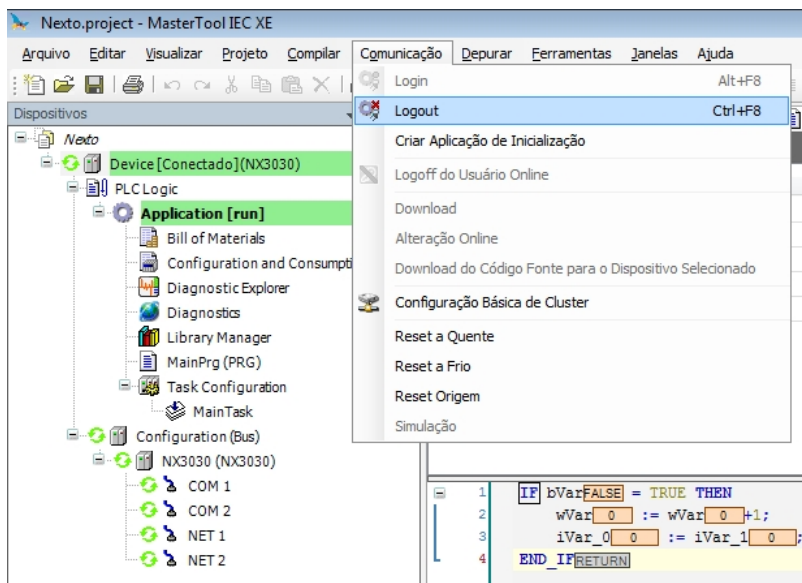


Figura 35: Encerrando a comunicação online com a UCP

4.12. Upload do Projeto

As UCPs da Série Nexto possibilitam o armazenamento do código fonte do projeto na memória interna do dispositivo, permitindo a futura recuperação completa do projeto e a realização de modificações na aplicação.

Para realizar a recuperação de um projeto previamente armazenado na memória interna da UCP, deve ser executado o comando localizado no menu *Arquivo* conforme mostra a figura abaixo.

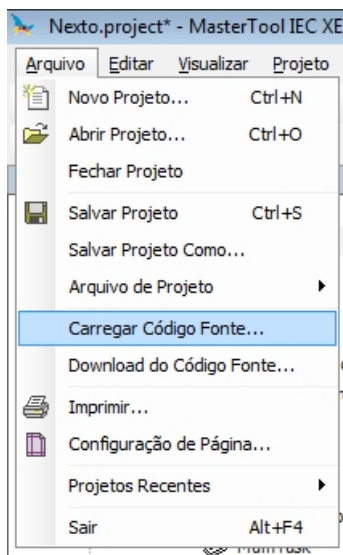


Figura 36: Opção de Upload de Projeto

Após, basta selecionar a UCP desejada e clicar em *OK*, conforme a figura abaixo.

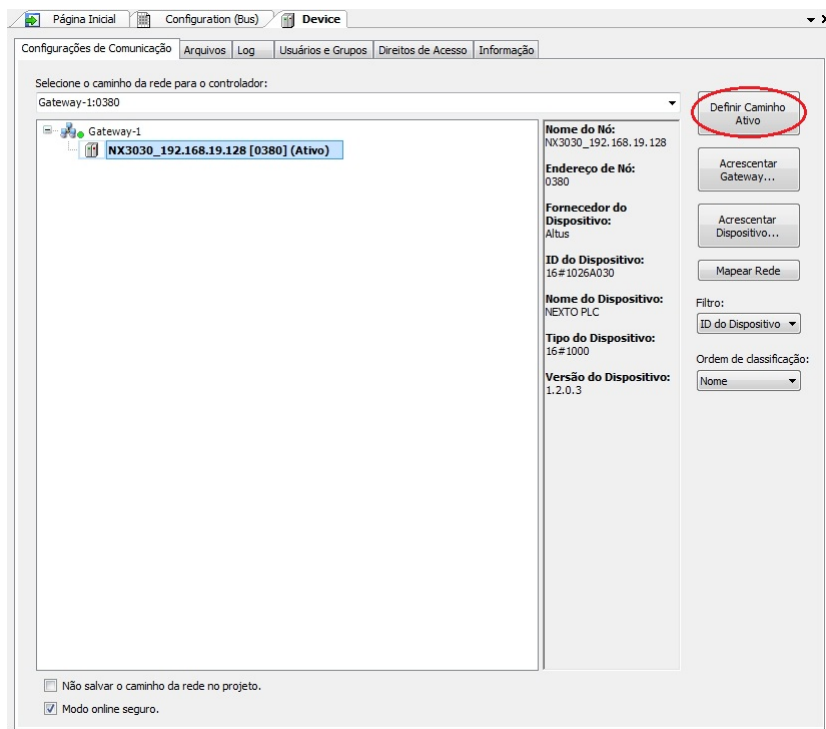


Figura 37: Selecionando a UCP

Para garantir que o projeto carregado da UCP seja completamente igual e possa ser carregado sem problemas a partir de outras estações, consulte o capítulo *Método de Envio/Login de Projetos Sem Diferença de Projetos* do Manual de Utilização MasterTool IEC XE MT8500 – MU299048.

ATENÇÃO

O tamanho da área de memória para armazenar um projeto nas UCPs Nexto está definido na seção [Memória](#).

ATENÇÃO

O Upload recupera o último projeto armazenado no controlador conforme descrito nos parágrafos anteriores. Caso tenha sido realizado apenas o download da aplicação, sem a transferência do seu código fonte, a mesma não poderá ser recuperada pelo procedimento de Upload.

4.13. Estados de Operação da UCP

4.13.1. Run

Quando uma UCP está em modo *Run*, todas as tarefas da aplicação são executadas.

4.13.2. Stop

Quando uma UCP está em modo *Stop*, todas as tarefas da aplicação estão paradas. O valor das variáveis nas tarefas são mantidos com o valor atual e os pontos de saída assumem o seu estado seguro.

Quando uma UCP passa para modo *Stop* devido ao envio de uma aplicação, as variáveis nas tarefas da aplicação serão perdidas exceto as variáveis do tipo persistente.

4.13.3. Breakpoint

Quando uma marca de depuração é atingida em uma tarefa, essa tarefa é interrompida. Todas as demais tarefas ativas na aplicação não serão interrompidas, continuando a sua execução. Com este recurso, é possível percorrer e investigar a execução de um programa passo a passo no modo *Online* conforme as posições de interrupção incluídas através do editor.

Para maiores informações sobre a utilização de marcas de depuração, favor consultar o Manual de Utilização do MasterTool IEC XE – MU299048.

4.13.4. Exception

Quando uma UCP está em *Exception* indica que alguma operação indevida ocorreu em uma das tarefas ativas da aplicação. A tarefa que causou o *Exception* será suspensa e as demais tarefas irão para o modo *Stop*. Somente é possível tirar as tarefas desse estado e colocá-las em execução novamente após uma nova condição de partida da UCP. Portanto somente com um *Reset a Quente*, *Reset a Frio*, *Reset Origem* ou uma reinicialização da UCP coloca novamente a aplicação em modo *Run*.

4.13.5. Reset a Quente

Este comando coloca a UCP em modo *Stop* e inicializa todas as variáveis das tarefas da aplicação, exceto as variáveis dos tipos retentiva e persistente. As variáveis inicializadas com um valor específico assumirão exatamente este valor, as demais variáveis assumirão o valor padrão de inicialização (zero).

4.13.6. Reset a Frio

Este comando coloca a UCP em modo *Stop* e inicializa todas as variáveis das tarefas da aplicação, exceto as variáveis do tipo persistente. As variáveis inicializadas com um valor específico assumirão exatamente este valor, as demais variáveis assumirão o valor padrão de inicialização (zero).

4.13.7. Reset Origem

Este comando remove todas as variáveis das tarefas da aplicação, inclusive as variáveis do tipo persistente, apaga a aplicação da UCP e coloca a UCP em modo *Stop*.

Notas:

Reset: Quando um *Reset* é executado, os breakpoints definidos na aplicação são desabilitados.

Comando: Para executar um comando de *Reset a Quente*, *a Frio* ou *Origem*, é necessário estar com o MasterTool em modo *Online* na UCP.

4.13.8. Reset Process Command (IEC 60870-5-104)

Este comando de reset do processo pode ser solicitado pelos clientes IEC 60870-5-104. Depois de responder ao cliente, a UCP entra num procedimento de reinicialização, como se estivesse sendo executado um ciclo de energização.

No caso de CPs redundantes, o comando de reset do processo é sincronizado com o CP não ativo, resultando na reinicialização dos dois CPs.

A norma IEC 60870-5-104 prevê a passagem de um valor de qualificação (0..255) com o comando de reset do processo, mas este “parâmetro” não é considerado pela UCP.

4.14. Programas (POUs) e Listas de Variáveis Globais (GVLs)

O projeto criado pela ferramenta MasterTool IEC XE contém um conjunto de módulos de programa (POUs) e listas de variáveis globais que visam facilitar a programação e a utilização do controlador. A seguir são descritos os principais elementos que fazem parte desta estrutura padrão do projeto.

4.14.1. Programa MainPrg

A tarefa *MainTask* está associada a uma única POU do tipo programa, denominada *MainPrg*. O programa *MainPrg* é criado automaticamente e não pode ser editado pelo usuário.

O código do programa *MainPrg* é o seguinte, em linguagem ST:

```
(*Main POU associated with MainTask that calls StartPrg,
  UserPrg/ActivePrg and NonSkippedPrg.
  This POU is blocked to edit.*)

PROGRAM MainPrg
VAR
  isFirstCycle : BOOL := TRUE;
END_VAR

SpecialVariablesPrg();
IF isFirstCycle THEN
  StartPrg();
  isFirstCycle := FALSE;
ELSE
  UserPrg();
END_IF;
```

MainPrg chama outras duas POU's do tipo programa, denominadas *StartPrg* e *UserPrg*. Enquanto a *UserPrg* sempre é chamada, a *StartPrg* só é chamada uma única vez na partida da aplicação do CP.

Ao contrário do programa *MainPrg*, que não deve ser modificado, o usuário pode modificar os programas *StartPrg* e *UserPrg*. Inicialmente, quando o projeto é criado a partir do Assistente, estes dois programas são criados vazios, mas o usuário poderá inserir código nos mesmos.

4.14.2. Programa StartPrg

Nesta POU o usuário pode criar lógicas, laços, iniciar variáveis, etc. que serão executados somente uma única vez no primeiro ciclo do CP, antes de executar a POU *UserPrg* pela primeira vez, não sendo chamado novamente durante a execução do projeto.

Caso o usuário carregue uma nova aplicação, ou se o CP for desenergizado, bem como em condições de *Reset Origem*, *Reset a Frio* e *Reset a Quente*, esta POU será executada novamente.

4.14.3. Programa UserPrg

Nesta POU o usuário deve criar a aplicação principal, responsável pelo controle de seu processo. Esta POU é chamada pela POU principal (MainPrg).

O usuário pode também criar POU's adicionais (programa, funções ou bloco funcional), e chamá-las ou instanciá-las dentro da POU *UserPrg*, para fins de estruturação de seu programa. Também é possível chamar funções e instanciar blocos funcionais definidos em bibliotecas.

4.14.4. GVL System_Diagnostics

Na GVL *System_Diagnostics* estão presentes as variáveis de diagnóstico da UCP, das interfaces de comunicação (Ethernet e PROFIBUS) e de todos os drivers de comunicação. Essa GVL não é editável e as variáveis são declaradas automaticamente com tipo especificado pelo dispositivo ao qual pertence quando este é adicionado ao projeto.

ATENÇÃO

Na GVL *System_Diagnostics* também são declaradas as variáveis de diagnóstico das requisições MODBUS Mestre/Cliente por representação direta.

Alguns dispositivos, como o driver de comunicação MODBUS de mapeamento simbólico, não têm seus diagnósticos alocados em variáveis %Q com a diretiva AT. O mesmo ocorre com drivers de comunicação mais recentes, como é o caso do IEC 60870-5-104 servidor.

A figura a seguir mostra um exemplo da apresentação desta GVL quando em modo *Online*.

Device.Application.System_Diagnostics				
Expression	Type	Value	Address	Con
DG_IEC_60870_5_104_Server	T_DIAG_IEC104_SERVER_1			DG_I
DG_MODBUS_Symbol_Client	T_DIAG_MODBUS_ETH_CLIENT_1			DG_I
tDiag	T_DIAG_MODBUS_DIAGNOSTICS_CLIENT			
byDiag_1_reserved	BYTE	0		Rese
tCommand	T_DIAG_MODBUS_COMMANDS			
byDiag_3_reserved	BYTE	0		Rese
tStat	T_DIAG_MODBUS_ETH_CLIENT_STATS			
wTXRequests	WORD	1589		Coun
wRXNormalResponses	WORD	1589		Coun
wRXExceptionResponses	WORD	0		Coun
wRXIllegalResponses	WORD	0		Coun
wDiag_12_reserved	WORD	0		Rese
wDiag_14_reserved	WORD	0		Rese
wDiag_16_reserved	WORD	0		Rese
wDiag_18_reserved	WORD	0		Rese
DG_MODBUS_Symbol_Client_NX5000	T_DIAG_MODBUS_ETH_CLIENT_1			DG_I
DG_MODBUS_Symbol_RTU_Master	T_DIAG_MODBUS_RTU_MASTER_1			DG_I
DG_MODBUS_Symbol_Server_NX5000	T_DIAG_MODBUS_ETH_SERVER_1			DG_I
DG_NX030	T_DIAG_NX030_1		%QB66229	DG_I
tSummarized	T_DIAG_SUMMARIZED_1			
tDetailed	T_DIAG_DETAILED_1			
DG_NX001	T_DIAG_NX001_1		%QB66922	DG_I
DG_MODBUS_Client	T_DIAG_MODBUS_ETH_CLIENT_1		%QB67191	DG_I
DG_MBUS_Direct_1_Mapping_000	T_DIAG_MODBUS_ETH_MAPPING_1		%QB67211	DG_I
byStatus	T_DIAG_MODBUS_ETH_MAPPING_STAT...			
bCommIdle	BIT	FALSE		Comr
bCommExecuting	BIT	FALSE		Comr
bCommPostponed	BIT	TRUE		Comr
bCommDisabled	BIT	FALSE		Comr
bCommOk	BIT	TRUE		Previ
bCommError	BIT	FALSE		Previ
bCommAborted	BIT	FALSE		Previ
bDiag_7_reserved	BIT	FALSE		Rese
eLastErrorCode	MASTER_ERROR_CODE	NO_ERROR		Last
eLastExceptionCode	MODBUS_EXCEPTION	NO_EXCEPTION		Last
byDiag_3_reserved	BYTE	0		rese
wCommCounter	WORD	397		Coun
wCommErrorCounter	WORD	0		Coun
DG_MBUS_Direct_1_Mapping_001	T_DIAG_MODBUS_ETH_MAPPING_1		%QB67219	DG_I
DG_MBUS_Direct_1_Mapping_003	T_DIAG_MODBUS_ETH_MAPPING_1		%QB67235	DG_I
DG_MBUS_Direct_1_Mapping_002	T_DIAG_MODBUS_ETH_MAPPING_1		%QB67243	DG_I
DG_NX5000	T_DIAG_NX5000_1		%QB67251	DG_I

Figura 38: GVL System_Diagnostics em Modo Online

4.14.5. GVL Disables

A GVL *Disables* contém as variáveis de desabilitação das requisições MODBUS Mestre/Cliente por mapeamento simbólico. Não é obrigatória, mas é recomendada a utilização da geração automática destas variáveis, o que é feito clicando no botão *Gerar Variáveis de Desabilitação* na aba de requisições do dispositivo. Essas variáveis são declaradas com o tipo BOOL e seguem a seguinte estrutura:

Declaração de variável de desabilitação de requisição:

```
[Nome do Dispositivo]_DISABLE_[Número da Requisição] : BOOL;
```

Onde:

Nome do dispositivo: Nome que aparece na visualização em árvore para o dispositivo MODBUS.

Número da Requisição: Número da requisição que foi declarada na tabela de requisições do dispositivo MODBUS seguindo a sequência de cima para baixo, começando em 0001.

Exemplo:

Device.Application.Disables

```

VAR_GLOBAL
MODBUS_Device_DISABLE_0001 : BOOL;
MODBUS_Device_DISABLE_0002 : BOOL;
MODBUS_Device_DISABLE_0003 : BOOL;
MODBUS_Device_1_DISABLE_0001 : BOOL;
MODBUS_Device_1_DISABLE_0002 : BOOL;
END_VAR
    
```

A geração automática através do botão *Gerar Variáveis de Desabilitação* apenas cria variáveis, e não remove automaticamente. Desta forma, caso alguma relação seja removida, a sua respectiva variável de desabilitação deve ser removida manualmente.

A GVL *Disables* é editável, portanto as variáveis de desabilitação das requisições podem ser criadas manualmente não precisando seguir o modelo criado pela declaração automática e podem ser usadas as duas maneiras ao mesmo tempo, mas devem sempre ser do tipo **BOOL** e deve-se tomar o cuidado para não excluir ou alterar as variáveis declaradas automaticamente, pois elas podem estar sendo usadas por algum dispositivo MODBUS. Se a variável for excluída ou alterada será gerado um erro ao compilar o projeto. Para corrigir o nome de uma variável declarada automaticamente, deve-se seguir o modelo exemplificado acima de acordo com o dispositivo e a requisição aos quais pertence.

A figura a seguir mostra um exemplo da apresentação desta GVL quando em modo *Online*. Se o valor da variável for **TRUE** significa que a requisição, à qual a variável pertence, está desabilitada e o inverso é válido para quando o valor da variável for **FALSE**.









Device.Application.Disables			
Expression	Type	Value	Prepared
 MODBUS_Slave_1_DISABLE_0001	BOOL	FALSE	
 MODBUS_Slave_1_DISABLE_0002	BOOL	TRUE	
 MODBUS_Slave_1_DISABLE_0003	BOOL	FALSE	
 MODBUS_Slave_1_DISABLE_0004	BOOL	FALSE	
 MODBUS_Server_1_DISABLE_0001	BOOL	FALSE	
 MODBUS_Server_1_DISABLE_0002	BOOL	FALSE	
 MODBUS_Server_1_DISABLE_0003	BOOL	FALSE	
 MODBUS_Server_1_DISABLE_0004	BOOL	TRUE	

Figura 39: GVL Disables em Modo Online

4.14.6. GVL IOQualities

A GVL *IOQualities* contém as variáveis de qualidade dos módulos de E/S declarados no barramento da UCP. Essa GVL não é editável e as variáveis são declaradas automaticamente como arrays do tipo *LibDataTypes.QUALITY* e dimensões de acordo com a quantidade de E/S do módulo ao qual pertence quando este é adicionado ao projeto.

Exemplo: Device.Application.IOQualities

```

VAR_GLOBAL
QUALITY_NX1001: ARRAY [0..15] OF LibDataTypes.QUALITY;
QUALITY_NX2020: ARRAY [0..15] OF LibDataTypes.QUALITY;
QUALITY_NX6000: ARRAY [0..7] OF LibDataTypes.QUALITY;
QUALITY_NX6100: ARRAY [0..3] OF LibDataTypes.QUALITY;
END_VAR
    
```

Uma vez a aplicação estando em *RUN* é possível monitorar os valores das variáveis de qualidade dos módulos de E/S que foram adicionados ao projeto através da GVL *IOQualities*.

4.14.7. GVL Module_Diagnostics

A *GVL Module_Diagnostics* contém as variáveis de diagnóstico dos módulos de E/S utilizados no projeto, exceto UCP e drivers de comunicação. Essa GVL não é editável e as variáveis são declaradas automaticamente com tipo especificado pelo módulo ao qual pertence quando este é adicionado ao projeto.

A figura a seguir mostra um exemplo da apresentação desta GVL quando em modo *Online*.

Device.Application.Module_Diagnostics				
Expression	Type	Value	Address	Comment
[-] DG_NX1001	T_DIAG_NX1001_1		%QB67008	DG_NX1001 diagnostics variable
[-] tGeneral	T_DIAG_GENERAL_NX1001_1			
[-] bReserved_8	BIT	FALSE		Reserved
[-] bReserved_9	BIT	FALSE		Reserved
[-] bReserved_10	BIT	FALSE		Reserved
[-] bReserved_11	BIT	FALSE		Reserved
[-] bReserved_12	BIT	FALSE		Reserved
[-] bReserved_13	BIT	FALSE		Reserved
[-] bReserved_14	BIT	FALSE		Reserved
[-] bReserved_15	BIT	FALSE		Reserved
[-] bActiveDiagnostics	BIT	FALSE		Module has active diagnostics
[-] bFatalError	BIT	FALSE		Module has fatal error
[-] bConfigMismatch	BIT	FALSE		Module has parameterization error
[-] bWatchdogError	BIT	FALSE		Module has watchdog expired
[-] bOTDSwitchError	BIT	FALSE		Module one touch diag switch error
[-] bReserved_5	BIT	FALSE		Reserved
[-] bReserved_6	BIT	FALSE		Reserved
[-] bReserved_7	BIT	FALSE		Reserved
[+] DG_NX1005	T_DIAG_NX1005_1		%QB67010	DG_NX1005 diagnostics variable
[+] DG_NX2001	T_DIAG_NX2001_1		%QB67014	DG_NX2001 diagnostics variable
[+] DG_NX2020	T_DIAG_NX2020_1		%QB67018	DG_NX2020 diagnostics variable
[+] DG_NX6000	T_DIAG_NX6000_1		%QB67022	DG_NX6000 diagnostics variable
[+] DG_NX6100	T_DIAG_NX6100_1		%QB67040	DG_NX6100 diagnostics variable
[-] tGeneral	T_DIAG_GENERAL_NX6100_1			
[-] bActiveDiagnosticsOutput00	BIT	FALSE		Output 00 with diagnostics
[-] bActiveDiagnosticsOutput01	BIT	FALSE		Output 01 with diagnostics
[-] bActiveDiagnosticsOutput02	BIT	FALSE		Output 02 with diagnostics
[-] bActiveDiagnosticsOutput03	BIT	FALSE		Output 03 with diagnostics
[-] bReserved_12	BIT	FALSE		Reserved
[-] bReserved_13	BIT	FALSE		Reserved
[-] bReserved_14	BIT	FALSE		Reserved
[-] bReserved_15	BIT	FALSE		Reserved
[-] bActiveDiagnostics	BIT	FALSE		Module has active diagnostics
[-] bFatalError	BIT	FALSE		Module has fatal error
[-] bConfigMismatch	BIT	FALSE		Module has parameterization error
[-] bWatchdogError	BIT	FALSE		Module has watchdog expired
[-] bOTDSwitchError	BIT	FALSE		Module one touch diag switch error
[-] bCalibrationError	BIT	FALSE		Module has calibration error
[-] bNoExternalSupply	BIT	FALSE		External power s...y is below the ...
[-] bReserved_07	BIT	FALSE		Reserved
[-] tDetailed	T_DIAG_DETAILED_NX6100_1			
[-] tAnalogOutput_00	T_DIAG_ANALOG_OUTPUT			
[-] tAnalogOutput_01	T_DIAG_ANALOG_OUTPUT			
[-] tAnalogOutput_02	T_DIAG_ANALOG_OUTPUT			

Figura 40: GVL Module_Diagnostics em Modo Online

4.14.8. GVL Qualities

A *GVL Qualities* contém as variáveis de qualidade dos mapeamentos de variáveis internas MODBUS Mestre/Cliente de mapeamento simbólico. Não é obrigatória, mas é recomendada a utilização da geração automática destas variáveis, o que é feito clicando no botão *Gerar Variáveis de Qualidade* na aba de mapeamentos do dispositivo. Essas variáveis são declaradas com o tipo *LibDataTypes.QUALITY* e seguem a seguinte estrutura:

Declaração de variável de qualidade de mapeamento:

4. PROGRAMAÇÃO INICIAL

```
[Nome do Dispositivo]_QUALITY_[Número do Mapeamento]: LibDataTypes.QUALITY;
```

Onde:

Nome do dispositivo: Nome que aparece na visualização em árvore para o dispositivo.

Número do Mapeamento: Número do mapeamento que foi declarado na tabela de mapeamentos do dispositivo seguindo a sequência de cima para baixo, começando em 0001.

ATENÇÃO

Não é possível associar variáveis de qualidade para os mapeamentos dos drivers MODBUS Mestre/Cliente por representação direta, portanto é recomendada a utilização de drivers MODBUS de mapeamento simbólico.

Exemplo: Device.Application.Qualities

```
VAR_GLOBAL
MODBUS_Device_QUALITY_0001: LibDataTypes.QUALITY;
MODBUS_Device_QUALITY_0002: LibDataTypes.QUALITY;
MODBUS_Device_QUALITY_0003: LibDataTypes.QUALITY;
END_VAR
```

A GVL *Qualities* é editável, portanto as variáveis de qualidade dos mapeamentos podem ser criadas manualmente não precisando seguir o modelo criado pela declaração automática e podem ser usadas as duas maneiras ao mesmo tempo, mas devem sempre ser do tipo *LibDataTypes.QUALITY* e deve-se tomar o cuidado para não excluir ou alterar as variáveis declaradas automaticamente, pois elas podem estar sendo usadas por algum dispositivo. Se a variável for excluída ou alterada será gerado um erro ao compilar o projeto. Para corrigir o nome de uma variável declarada automaticamente, deve-se seguir o modelo exemplificado acima de acordo com o dispositivo e o mapeamento aos quais pertence.

Para os dispositivos de comunicação MODBUS, as variáveis de qualidade se comportam da maneira indicada na Tabela 56.

A figura a seguir mostra um exemplo da apresentação desta GVL quando em modo *Online*.

ATENÇÃO

Se uma variável do driver MODBUS Mestre/Cliente de mapeamento simbólico for mapeada no driver IEC 60870-5-104 Servidor, é necessário que as variáveis de qualidade dos mapeamentos MODBUS tenham sido criadas para que sejam gerados eventos de qualidade válidos para tais pontos dos servidores IEC 60870-5-104. Caso contrário, não serão gerados eventos de qualidade “ruim” para os clientes do servidor IEC 60870-5-104 nas situações que o MODBUS Mestre/Cliente não consiga comunicar com os seus escravos/servidores, por exemplo.

Device.Application.Qualities				
Expression	Type	Value	Address	Comment
MODBUS_Slave_1_QUALITY_0001	LibDataTypes.QUALITY			
VALIDITY	QUALITY_VALIDITY	VALIDITY_GOOD		Quality validity
FLAGS	QUALITY_FLAGS			Quality flags
FLAG_OUT_OF_RANGE	BIT	FALSE		Bit 8
FLAG_INACCURATE	BIT	FALSE		Bit 9
FLAG_OLD_DATA	BIT	FALSE		Bit 10
FLAG_FAILURE	BIT	FALSE		Bit 11
FLAG_OPERATOR_BLOCKED	BIT	FALSE		Bit 12
FLAG_TEST	BIT	FALSE		Bit 13
FLAG_RESERVED_0	BIT	FALSE		Bit 14
FLAG_RESERVED_1	BIT	FALSE		Bit 15
FLAG_RESTART	BIT	FALSE		Bit 0
FLAG_COMM_FAIL	BIT	FALSE		Bit 1
FLAG_REMOTE_SUBSTITU...	BIT	FALSE		Bit 2
FLAG_LOCAL_SUBSTITUTED	BIT	FALSE		Bit 3
FLAG_FILTER	BIT	FALSE		Bit 4
FLAG_OVERFLOW	BIT	FALSE		Bit 5
FLAG_REFERENCE_ERROR	BIT	FALSE		Bit 6
FLAG_INCONSISTENT	BIT	FALSE		Bit 7
MODBUS_Slave_1_QUALITY_0002	LibDataTypes.QUALITY			
MODBUS_Slave_1_QUALITY_0003	LibDataTypes.QUALITY			
MODBUS_Slave_1_QUALITY_0004	LibDataTypes.QUALITY			
MODBUS_Server_1_QUALITY_0001	LibDataTypes.QUALITY			
MODBUS_Server_1_QUALITY_0002	LibDataTypes.QUALITY			
MODBUS_Server_1_QUALITY_0003	LibDataTypes.QUALITY			
VALIDITY	QUALITY_VALIDITY	VALIDITY_QUESTIONABLE		Quality validity
FLAGS	QUALITY_FLAGS			Quality flags
FLAG_OUT_OF_RANGE	BIT	FALSE		Bit 8
FLAG_INACCURATE	BIT	FALSE		Bit 9
FLAG_OLD_DATA	BIT	TRUE		Bit 10
FLAG_FAILURE	BIT	FALSE		Bit 11
FLAG_OPERATOR_BLOCKED	BIT	FALSE		Bit 12
FLAG_TEST	BIT	FALSE		Bit 13
FLAG_RESERVED_0	BIT	FALSE		Bit 14
FLAG_RESERVED_1	BIT	FALSE		Bit 15
FLAG_RESTART	BIT	FALSE		Bit 0
FLAG_COMM_FAIL	BIT	TRUE		Bit 1
FLAG_REMOTE_SUBSTITU...	BIT	FALSE		Bit 2
FLAG_LOCAL_SUBSTITUTED	BIT	FALSE		Bit 3
FLAG_FILTER	BIT	FALSE		Bit 4
FLAG_OVERFLOW	BIT	FALSE		Bit 5
FLAG_REFERENCE_ERROR	BIT	FALSE		Bit 6
FLAG_INCONSISTENT	BIT	FALSE		Bit 7
MODBUS_Server_1_QUALITY_0004	LibDataTypes.QUALITY			

Figura 41: GVL Qualities em Modo Online

4.14.9. GVL ReqDiagnostics

A GVL *ReqDiagnostics* contém as variáveis de diagnóstico das requisições MODBUS Mestre/Cliente de mapeamento simbólico. Não é obrigatória, mas é recomendada a utilização da geração automática destas variáveis, o que é feito clicando no botão *Gerar Variáveis de Diagnósticos* na aba de requisições do dispositivo. A declaração destas variáveis segue a seguinte estrutura:

Declaração de variável de diagnóstico de requisição:

```
[Nome do Dispositivo]_REQDG_[Número da requisição]: [Tipo da Variável];
```

Onde:

Nome do dispositivo: Nome que aparece na visualização em árvore para o dispositivo.

Número da Requisição: Número da requisição que foi declarada na tabela de requisições do dispositivo seguindo a sequência de cima para baixo, começando em 0001.

Tipo da Variável: NXMODBUS_DIAGNOSTIC_STRUCTS.T_DIAG_MODBUS_RTU_MAPPING_1 para MODBUS Mestre e NXMODBUS_DIAGNOSTIC_STRUCTS.T_DIAG_MODBUS_ETH_MAPPING_1 para MODBUS Cliente.

ATENÇÃO

As variáveis de diagnóstico das requisições MODBUS Mestre/Cliente por representação direta são declaradas na GVL *System_Diagnostics*.

Exemplo:

Device.Application.ReqDiagnostics

VAR_GLOBAL

```

MODBUS_Device_REQDG_0001 : NXMODBUS_DIAGNOSTIC_STRUCTS.
                             T_DIAG_MODBUS_RTU_MAPPING_1;
MODBUS_Device_REQDG_0002 : NXMODBUS_DIAGNOSTIC_STRUCTS.
                             T_DIAG_MODBUS_RTU_MAPPING_1;
MODBUS_Device_REQDG_0003 : NXMODBUS_DIAGNOSTIC_STRUCTS.
                             T_DIAG_MODBUS_RTU_MAPPING_1;
MODBUS_Device_1_REQDG_0001 : NXMODBUS_DIAGNOSTIC_STRUCTS.
                              T_DIAG_MODBUS_ETH_MAPPING_1;
MODBUS_Device_1_REQDG_0002 : NXMODBUS_DIAGNOSTIC_STRUCTS.
                              T_DIAG_MODBUS_ETH_MAPPING_1;
    
```

END_VAR

A GVL *ReqDiagnostics* é editável, portanto as variáveis de diagnóstico das requisições podem ser criadas manualmente não precisando seguir o modelo criado pela declaração automática e podem ser usadas as duas maneiras ao mesmo tempo, mas as variáveis devem ser sempre do tipo referente ao dispositivo, como exemplificado acima, e deve-se tomar o cuidado para não excluir ou alterar as variáveis declaradas automaticamente, pois elas podem estar sendo utilizadas por um dispositivo. Se a variável for excluída ou alterada será gerado um erro ao compilar o projeto. Para corrigir o nome de uma variável declarada automaticamente, deve-se seguir o modelo exemplificado acima de acordo com o dispositivo e a requisição aos quais pertence.

A figura a seguir mostra um exemplo da apresentação desta GVL quando em modo *Online*.

Device.Application.ReqDiagnostics		
Expression	Type	Value
MODBUS_Slave_1_REQDG_0001	NXMODBUS_DIAGNOSTIC_STRUCTS.T_DIAG_MODBUS...	
byStatus	T_DIAG_MODBUS_RTU_MAPPING_STATUS	
eLastErrorCode	MASTER_ERROR_CODE	NO_ERROR
eLastExceptionCode	MODBUS_EXCEPTION	NO_EXCEPTION
byDiag_3_reserved	BYTE	0
wCommCounter	WORD	969
wCommErrorCounter	WORD	0
MODBUS_Slave_1_REQDG_0002	NXMODBUS_DIAGNOSTIC_STRUCTS.T_DIAG_MODBUS...	
MODBUS_Slave_1_REQDG_0003	NXMODBUS_DIAGNOSTIC_STRUCTS.T_DIAG_MODBUS...	
MODBUS_Slave_1_REQDG_0004	NXMODBUS_DIAGNOSTIC_STRUCTS.T_DIAG_MODBUS...	
MODBUS_Server_1_REQDG_0001	NXMODBUS_DIAGNOSTIC_STRUCTS.T_DIAG_MODBUS...	
MODBUS_Server_1_REQDG_0002	NXMODBUS_DIAGNOSTIC_STRUCTS.T_DIAG_MODBUS...	
MODBUS_Server_1_REQDG_0003	NXMODBUS_DIAGNOSTIC_STRUCTS.T_DIAG_MODBUS...	
byStatus	T_DIAG_MODBUS_ETH_MAPPING_STATUS	
eLastErrorCode	MASTER_ERROR_CODE	ERR_CONNECTION_TIMEOUT
eLastExceptionCode	MODBUS_EXCEPTION	NO_EXCEPTION
byDiag_3_reserved	BYTE	0
wCommCounter	WORD	116
wCommErrorCounter	WORD	49
MODBUS_Server_1_REQDG_0004	NXMODBUS_DIAGNOSTIC_STRUCTS.T_DIAG_MODBUS...	

Figura 42: GVL ReqDiagnostics em Modo Online

4.14.10. Função Prepare_Start

Nesta POU é definida a função do evento de sistema *PrepareStart*. Ela pertence a tarefa de comunicação e é chamada antes de dar início à aplicação. Quando houver uma comunicação ativa com o CLP, é possível observar o status do evento e a contagem de chamadas na aba *Eventos de Sistema* no objeto *Configuração da Tarefa*. Toda vez que o usuário dar início na aplicação, a contagem é incrementada.

4.14.11. Função Prepare_Stop

Nesta POU é definida a função do evento de sistema *PrepareStop*. Ela pertence a tarefa de comunicação e é chamada antes de parar à aplicação. Quando houver uma comunicação ativa com o CLP, é possível observar o status do evento e a contagem de chamadas na aba *Eventos de Sistema* no objeto *Configuração da Tarefa*. Toda vez que o usuário parar à aplicação, a contagem é incrementada.

4.14.12. Função Start_Done

Nesta POU é definida a função do evento de sistema *StartDone*. Ela pertence a tarefa de comunicação e é chamada quando a aplicação é iniciada com sucesso. Quando houver uma comunicação ativa com o CLP, é possível observar o status do evento e a contagem de chamadas na aba *Eventos de Sistema* no objeto *Configuração da Tarefa*. Toda vez que o usuário iniciar à aplicação com sucesso, a contagem é incrementada.

4.14.13. Função Stop_Done

Nesta POU é definida a função do evento de sistema *StopDone*. Ela pertence a tarefa de comunicação e é chamada quando a aplicação é parada com sucesso. Quando houver uma comunicação ativa com o CLP, é possível observar o status do evento e a contagem de chamadas na aba *Eventos de Sistema* no objeto *Configuração da Tarefa*. Toda vez que o usuário parar à aplicação com sucesso, a contagem é incrementada.

5. Configuração

As UCPs da Série Nexto são configuradas e programadas através do software MasterTool IEC XE. A configuração realizada define o comportamento e modos de utilização dos periféricos e características especiais das UCPs. A programação representa a aplicação desenvolvida pelo usuário, também conhecida como Application.

5.1. Device

5.1.1. Gerenciamento de Usuários e Direitos de Acesso

Fornecer funções para definir contas dos usuários e configurar os direitos de acesso ao projeto e a UCP. Utilizando o software MasterTool IEC XE é possível criar e gerenciar usuários e grupos, configurando diferentes níveis de direitos de acesso ao projeto para estes.

Simultaneamente, as UCPs Nexto possuem um sistema de gerenciamento de permissões de usuário, que bloqueia ou permite certas ações para cada grupo de usuários na UCP. Para mais informações é necessário consultar o Manual de Utilização MasterTool IEC XE MT8500 – MU299048, capítulo Gerenciamento de Usuários e Direitos de Acesso.

5.1.2. Configurações do CP

Nesta aba do editor de dispositivo genérico, você faz as configurações básicas para a configuração do CP, por exemplo, o manuseio de entradas e saídas e a tarefa de ciclo de barramento.

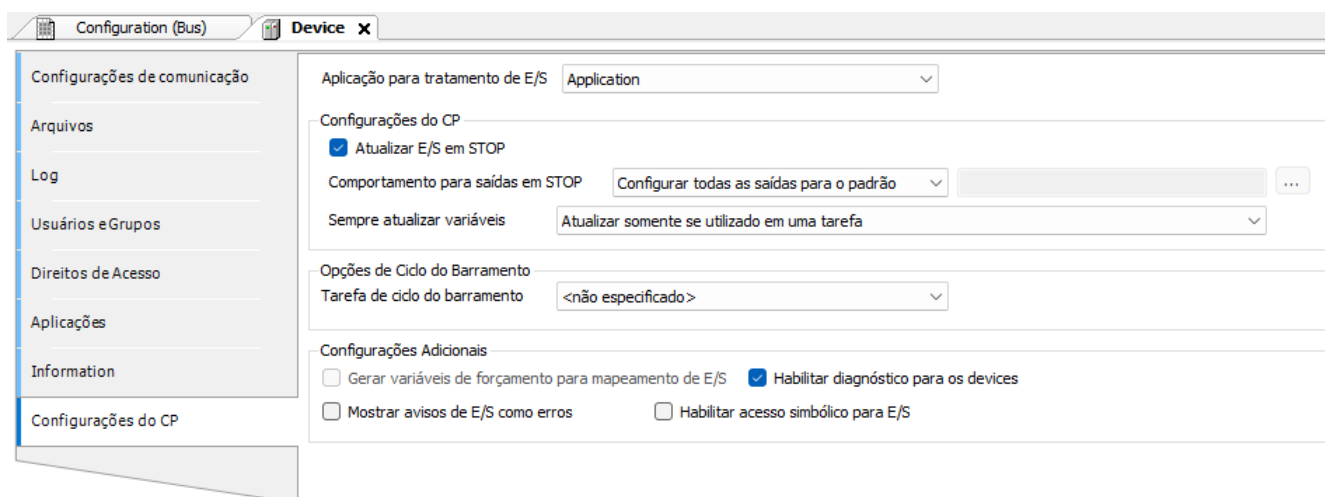


Figura 43: Configurações do CP

Parâmetro	Descrição
Aplicação para tratamento de E/S	Aplicação que é responsável pelo tratamento das entradas e saídas.
Atualizar E/S em STOP	<p>TRUE: Os valores dos canais de entrada e saída também são atualizados quando o CP está no modo STOP. Se o watchdog detectar um mau funcionamento, as saídas são definidas para os valores padrão predefinidos.</p> <p>FALSE: Os valores dos canais de entrada e saída no modo STOP não são atualizados.</p>

Parâmetro	Descrição
Comportamento para as saídas em STOP	<p>Manuseio dos canais de saída quando o controlador entra no modo STOP:</p> <p>Reter valores: Os valores atuais são retidos.</p> <p>Todas as saídas para o valor padrão: Os valores padrão resultantes do mapeamento de E/S são atribuídos.</p> <p>Executar programa: A manipulação dos valores de saída é controlada por um programa contido no projeto que é executado no modo STOP. Digite o nome do programa no campo à direita.</p>
Sempre atualizar variáveis	<p>Define globalmente se as variáveis de I/O são ou não atualizadas na tarefa de ciclo do barramento.</p> <p>Esta configuração é efetiva para as variáveis de E/S dos escravos e módulos somente se "desativado" for definido em suas configurações de atualização.</p> <p>Desativado (atualiza somente se usado em uma tarefa): As variáveis de E/S são atualizadas somente se forem usadas em uma tarefa.</p> <p>Habilitado 1 (usar tarefa de ciclo de barramento se não for usada em nenhuma tarefa): As variáveis de E/S na tarefa de ciclo de barramento são atualizadas se não forem usadas em nenhuma outra tarefa.</p> <p>Habilitado 2 (sempre na tarefa de ciclo de barramento): Todas as variáveis em cada ciclo da tarefa de ciclo de barramento são atualizadas, independentemente de serem usadas e mapeadas para um canal de entrada ou saída.</p>
Tarefa de ciclo de barramento	<p>Tarefa que controla o ciclo do barramento. Por padrão, a tarefa definida pela descrição do dispositivo é inserida.</p> <p>Por padrão, a configuração do ciclo de barramento do dispositivo de barramento superior se aplica (use as configurações de ciclo do barramento superior). Isso significa que a árvore de dispositivos é pesquisada para cima para a próxima definição válida da tarefa de ciclo de barramento.</p>
Gerar variáveis de forçamento para mapeamento de E/S	<p>TRUE: Ao compilar o aplicativo, duas variáveis globais são criadas para cada canal de E/S que é mapeado para uma variável na caixa de diálogo Mapeamento de E/S.</p>
Habilitar diagnóstico para os devices	<p>TRUE: A biblioteca CAA Device Diagnosis está integrada no projeto. Um bloco de função implícito é gerado para cada dispositivo. Se já existe um bloco de função para o dispositivo, então um bloco de função estendido é gerado (exemplo: EtherCAT) ou outra instância de bloco de função é adicionada. Isso contém uma implementação geral do diagnóstico do dispositivo.</p>
Mostrar avisos de E/S como erros	<p>Os avisos relativos à configuração de E/S são exibidos como erros.</p>
Habilitar acesso simbólico para as E/S	<p>TRUE: Permite o acesso aos pontos de E/S a partir do nome simbólico interno gerado na declaração do dispositivo. O nome simbólico pode ser consultado na coluna <i>Channel</i> na aba de <i>Bus I/O Mapping</i> de cada dispositivo.</p>

Tabela 43: Configurações do CP

ATENÇÃO

Os produtos Nexto (NX), Nexto Jet (NJ) e Xtorm (HX) não possuem suporte ao parâmetro *Habilitar acesso simbólico para as E/S*.

5.2. Configuração da UCP**5.2.1. Parâmetros Gerais**

Os parâmetros relacionados abaixo fazem parte da configuração da UCP inserida na aplicação. Cada item deve ser devidamente revisado para o correto funcionamento do projeto.

Além destes parâmetros, é possível alterar o nome de cada módulo inserido na aplicação, para isto, clique com o botão direito sobre o módulo, no item *Propriedades*, na guia *Comum*, altere o nome, sendo limitado a 24 caracteres.

Configuração	Descrição	Padrão	Opções
Área de Diagnósticos (%Q)			
Endereço Inicial em %Q	Endereço inicial dos diagnósticos da UCP (%Q)	Alocado automaticamente na criação do projeto.	0 a 97611
Tamanho	Tamanho da área de diagnósticos em bytes	693	Não é possível alterar o tamanho da área de diagnósticos da UCP
Área Retentiva (%Q)			
Endereço Inicial em %Q	Endereço inicial da memória de dados retentivos (%Q)	4096	0 a 98303
Tamanho	Tamanho da memória de dados retentivos em bytes	98304	0 a 98304
Área Persistente (%Q)			
Endereço Inicial em %Q	Endereço inicial da memória de dados persistentes (%Q)	20480	0 a 98302
Tamanho	Tamanho da memória de dados persistentes em bytes	98304	0 a 98304
Parâmetros da UCP			
Iniciar Aplicação de Usuário após Reset por Cão-de-Guarda	Quando habilitado, inicia a aplicação do usuário após o reset do cão-de-guarda de hardware ou pela reinicialização do Runtime, porém mantém a indicação do diagnóstico via LED WD e via variáveis.	Desabilitado	Habilitado Desabilitado

Configuração	Descrição	Padrão	Opções
Troca a Quente	Modo troca a quente dos módulos	Habilitada, sem consistência na partida. (pode variar de acordo com o modelo de UCP)	<ul style="list-style-type: none"> - Desabilitada, apenas para módulos declarados - Desabilitada (com consistência na partida) - Desabilitada, sem consistência na partida - Habilitada, com consistência na partida apenas para módulos declarados - Habilitada, com consistência na partida - Habilitada, sem consistência na partida
Parâmetros do Projeto			
Habilita atualização de E/S por tarefa	Configuração para atualizar as entradas e saídas nas tarefas em que estas são utilizadas.	Desmarcado	<ul style="list-style-type: none"> - Marcado: As entradas e saídas são atualizadas pelas tarefas em que forem utilizadas. - Desmarcado: As entradas e saídas são atualizadas apenas pela MainTask
Habilita variáveis retentivas e persistentes em Blocos Funcionais	Configuração que permite o uso de variáveis retentivas e persistentes em Blocos Funcionais	Desmarcado	<ul style="list-style-type: none"> - Marcado: permite o uso de variáveis retentivas e persistentes em Blocos Funcionais. - Desmarcado: pode ocorrer erro de exceção na partida.

Tabela 44: Configurações da UCP

Notas:

Gerar erro na consistência do cão-de-guarda das tarefas: Este parâmetro foi descontinuado a partir da versão 1.32 do software MasterTool IEC XE.

Habilita atualização de E/S por tarefa: Este parâmetro foi adicionado a partir da versão 2.01 do software MasterTool IEC XE.

ATENÇÃO

Quando o endereço inicial ou o tamanho da memória de dados retentivos ou persistentes são alterados na aplicação do usuário, a memória é totalmente realocada, fazendo com que a área de variáveis retentivas e persistentes seja limpa. Então, o usuário deverá ter precaução para não perder os dados salvos na memória.

ATENÇÃO

Em situações em que a área de memória simbólica persistente é modificada, é apresentada uma mensagem pelo programador MasterTool IEC XE para que seja escolhido um comportamento para esta área após a carga do programa modificado. A escolha deste comportamento não afeta a área persistente de representação direta, que é sempre limpa.

ATENÇÃO

A opção *Habilita atualização de E/S por tarefa* não é suportada para módulos mestre de rede de campo como o NX5001 por exemplo. Esta funcionalidade é aplicável somente para módulos de entrada e saída do barramento local do controlador (bastidor principal e bastidores de expansão).

ATENÇÃO

Mesmo que um ponto de E/S seja utilizado e atualizado em outras tarefas, com a opção *Habilita atualização de E/S por tarefa* marcada, ele continuará sendo atualizado também na MainTask; exceto quando todos os pontos de E/S do módulo forem utilizados e atualizados em outra tarefa, neste caso não serão mais atualizados pela MainTask.

5.2.1.1. Troca a Quente

As UCPs da série Nexto apresentam a possibilidade de troca dos módulos de E/S do barramento sem a necessidade de desligamento do sistema e sem perda de informações. Esta característica é conhecida como troca a quente.

CUIDADO

As UCPs da Série Nexto não garantem a retentividade das variáveis persistentes e retentivas caso a fonte de alimentação, ou a própria UCP, seja removida do bastidor energizado.

Na troca a quente, o comportamento do sistema relacionado se modifica conforme a configuração definida pelo usuário, que apresenta as seguintes opções, conforme descrito abaixo:

- Desabilitada, apenas para módulos declarados
- Desabilitada (com consistência na partida)
- Desabilitada, sem consistência na partida
- Habilitada, com consistência na partida apenas para módulos declarados
- Habilitada, com consistência na partida
- Habilitada, sem consistência na partida

Assim, o usuário pode escolher o comportamento que o sistema deverá apresentar em situações anormais de barramento e quando a UCP estiver em Modo *Run*. A tabela abaixo apresenta as possíveis situações anormais de barramento.

Situação	Possíveis causas
Configuração incompatível	- Algum módulo presente no barramento é diferente do modelo que está declarado na configuração
Módulo ausente	- O módulo foi retirado do barramento - Algum módulo não está respondendo à UCP por estar com defeito - Alguma posição do bastidor está com defeito

Tabela 45: Situações Anormais de Barramento

Para maiores informações sobre os diagnósticos correspondentes às situações descritas acima, consultar a seção *Diagnósticos via Variáveis*.

Se um módulo está presente em uma posição do bastidor na qual não deveria existir módulo de acordo com a configuração, este módulo é considerado como não declarado. As opções de troca a quente *Desabilitada, apenas para módulos declarados* e *Habilitada, com consistência na partida apenas para módulos declarados* não levam em consideração os módulos que se encontram nesta condição.

5.2.1.1.1. Troca a Quente Desabilitada, Apenas para Módulos Declarados

Nesta configuração, quando ocorre uma situação anormal de barramento (conforme a Tabela 45) a UCP entra em Modo *Stop* em um tempo de até 2 segundos, quando o LED DG começa a piscar 4x (conforme a Tabela 46). Para que a UCP volte ao estado normal *Run*, além de desfazer o que causou a situação anormal é necessário executar um *Reset a Quente* ou um *Reset a Frio*. Caso seja realizado um *Reset Origem*, será necessário realizar o download para que a UCP possa voltar ao estado normal *Run*. Os comandos de *Reset a Quente*, *Reset a Frio* e *Reset Origem* podem ser feitos pelo MasterTool IEC XE no menu *Comunicação*.

A UCP irá permanecer no estado normal *Run* mesmo que encontre um módulo não declarado no barramento.

5.2.1.1.2. Troca a Quente Desabilitada

Esta configuração não permite qualquer situação anormal de barramento (conforme a Tabela 45) inclusive módulos não declarados e presentes no barramento. A UCP entra em Modo *Stop*, sendo que o LED DG começa a piscar 4x (conforme a Tabela 46). Para que, nesses casos, a UCP volte ao estado normal *Run*, além de desfazer o que causou a situação anormal é necessário executar um *Reset a Quente* ou um *Reset a Frio*. Caso seja realizado um *Reset Origem*, será necessário realizar o download para que a UCP possa voltar ao estado normal *Run*. Os comandos de *Reset a Quente*, *Reset a Frio* e *Reset Origem* podem ser feitos pelo MasterTool IEC XE no menu *Comunicação*.

5.2.1.1.3. Troca a Quente Desabilitada, sem Consistência na Partida

Permite que o sistema entre em operação mesmo quando algum módulo estiver em uma situação anormal de barramento (conforme Tabela 45). As situações anormais são relatadas via diagnóstico.

Qualquer modificação no barramento vai fazer a UCP entrar em Modo *Stop*, sendo que o LED DG começa a piscar 2x (conforme a Tabela 46). Para que, nesses casos, a UCP volte ao estado normal *Run*, é necessário executar um *Reset a Quente* ou um *Reset a Frio*. Caso seja realizado um *Reset Origem*, será necessário realizar o download para que a UCP possa voltar ao estado normal *Run*. Os comandos de *Reset a Quente*, *Reset a Frio* e *Reset Origem* podem ser feitos pelo MasterTool IEC XE no menu *Comunicação*.

5.2.1.1.4. Troca a Quente Habilitada, com Consistência na Partida Apenas para Módulos Declarados

É considerada “partida” o intervalo entre a energização da UCP (ou comando de reset ou download de aplicação) até a primeira vez em que a mesma entra em modo *Run*. Esta configuração verifica se ocorreu alguma situação anormal de barramento (conforme a Tabela 45) durante a partida; em caso positivo, a UCP entra em Modo *Stop* e o LED DG começa a piscar 4x (conforme a Tabela 46). Posteriormente, para que a UCP possa ser colocada em modo *Run*, além de corrigir o que ocasionou a situação anormal, é necessário executar um comando de *Reset a Quente* ou um *Reset a Frio*. Caso seja realizado um *Reset Origem*, será necessário realizar o download para que a UCP possa voltar ao estado normal *Run*. Os comandos de *Reset a Quente*, *Reset a Frio* e *Reset Origem* podem ser feitos pelo MasterTool IEC XE no menu *Comunicação*.

Após a partida, se algum módulo apresentar alguma das situações citadas na tabela anterior, o sistema continuará trabalhando normalmente e sinalizará o problema via diagnóstico.

Se não existir outra situação anormal para os módulos declarados, a UCP irá para o estado normal *Run* mesmo que encontre um módulo não declarado no barramento.

ATENÇÃO

Nesta configuração, quando ocorrer falta de alimentação (mesmo que temporária), comando *Reset a Quente*, comando *Reset a Frio* ou ter sido realizado o *Download* de uma nova aplicação, e algum módulo estiver em uma situação anormal de barramento; a UCP entrará em Modo *Stop* e o LED DG começa a piscar 4x (conforme a Tabela 46), pois estas são consideradas situações de partida.

Esta é a opção mais recomendada, pois garante a integridade do sistema na sua inicialização e permite a troca de módulos com o sistema funcionando.

5.2.1.1.5. Troca a Quente Habilitada, com Consistência na Partida

Esta configuração verifica se ocorreu alguma situação anormal de barramento (conforme a Tabela 45) durante a partida, inclusive se há módulos não declarados e presentes no barramento; em caso positivo, a UCP entra em Modo *Stop* e o LED DG começa a piscar 4x (conforme a Tabela 46). Posteriormente, para que a UCP possa ser colocada em modo *Run*, além de corrigir o que ocasionou a situação anormal, é necessário executar um comando de *Reset a Quente* ou um *Reset a Frio*. Caso

seja realizado um *Reset Origem*, será necessário realizar o download para que a UCP possa voltar ao estado normal *Run*. Os comandos de *Reset a Quente*, *Reset a Frio* e *Reset Origem* podem ser feitos pelo MasterTool IEC XE no menu *Comunicação*.

5.2.1.1.6. Troca a Quente Habilitada, sem Consistência na Partida

Permite que o sistema entre em operação mesmo quando algum módulo estiver em uma situação anormal de barramento (conforme Tabela 45). As situações anormais são relatadas via diagnóstico, tanto durante, como após a partida.

ATENÇÃO

Esta opção é recomendada para a fase de implantação do sistema, pois permite que cargas de novas aplicações e o desligamento da alimentação sejam feitos sem a presença de todos os módulos configurados.

5.2.1.1.7. Como realizar a Troca a Quente

CUIDADO

Antes de proceder à troca a quente, é importante descarregar eventuais potenciais estáticos acumulados no corpo. Para isso, toque (com as mãos nuas) em uma superfície metálica aterrada antes de manipular os módulos. Tal procedimento garante que os níveis de eletricidade estática suportados pelo módulo não serão ultrapassados.

ATENÇÃO

Recomenda-se o monitoramento dos diagnósticos de troca a quente na aplicação de controle desenvolvida pelo usuário, a fim de garantir que o valor retornado pelo módulo seja validado antes de ser utilizado.

O procedimento para a troca de módulos a quente é descrito a seguir:

- Destrave o módulo junto ao bastidor, através da trava de segurança.
- Retire o módulo, puxando-o firmemente.
- Insira o novo módulo no bastidor.
- Certifique-se de que a trava que prende o módulo ao bastidor está totalmente conectada; caso necessário, empurre o módulo em direção ao bastidor com mais força.

No caso de módulos de saída, é conveniente que os pontos estejam desligados por ocasião da troca, a fim de reduzir a geração de arcos no conector do módulo. Isso pode ser feito pelo desligamento da fonte de campo ou pelo forçamento dos pontos via ferramentas de software. Se a carga for pequena, não há a necessidade de desligar os pontos.

É importante salientar que, nos casos em que a UCP entra em Modo *Stop* e o LED DG começa a piscar 4x, conforme a Tabela 46, devido a alguma situação anormal de barramento, conforme a Tabela 45; os módulos de saída têm o comportamento dos seus pontos de acordo com o que foi configurado nos *Parâmetros do Módulo* quando a UCP sai do Modo *Run* e entra em Modo *Stop*. Em caso de inicialização da aplicação, quando a UCP entra em Modo *Stop* sem ter passado para o Modo *Run*, os módulos de saída têm o comportamento de seus pontos em modo seguro de falha, ou seja, o ponto permanece desligado (0 Vdc).

No caso dos módulos de entrada, caso o mesmo seja removido do barramento energizado, o estado lógico dos pontos permanecerá no último valor. Caso os conectores sejam removidos, o estado lógico dos pontos será colocado em estado seguro, ou seja, zero ou alta impedância.

ATENÇÃO

Proceda sempre à substituição de um módulo por vez, para que a UCP atualize os estados dos módulos.

Abaixo, a Tabela 46 relaciona as condições de barramento e o estado de operação do LED DG da UCP Nexto. Para maiores informações sobre os estados dos LEDs de diagnóstico, consultar a seção [Diagnósticos via LED](#).

Condição	Habilitada, com consistência na partida	Habilitada, com consistência na partida apenas para módulos declarados	Habilitada, sem consistência na partida	Desabilitada	Desabilitada, apenas para módulos declarados	Desabilitada, sem consistência na partida
Módulo não declarado	LED DG: Pisca 2x Aplicação: Run	LED DG: Pisca 2x Aplicação: Run	LED DG: Pisca 2x Aplicação: Run	LED DG: Pisca 4x Aplicação: Stop	LED DG: Pisca 2x Aplicação: Run	LED DG: Pisca 2x Aplicação: Stop
Módulo não declarado (condição partida)	LED DG: Pisca 4x Aplicação: Stop	LED DG: Pisca 2x Aplicação: Run	LED DG: Pisca 2x Aplicação: Run	LED DG: Pisca 4x Aplicação: Stop	LED DG: Pisca 2x Aplicação: Run	LED DG: Pisca 2x Aplicação: Run
Módulo ausente	LED DG: Pisca 2x Aplicação: Run	LED DG: Pisca 2x Aplicação: Run	LED DG: Pisca 2x Aplicação: Run	LED DG: Pisca 4x Aplicação: Stop	LED DG: Pisca 4x Aplicação: Stop	LED DG: Pisca 2x Aplicação: Stop
Módulo ausente (condição partida)	LED DG: Pisca 4x Aplicação: Stop	LED DG: Pisca 4x Aplicação: Stop	LED DG: Pisca 2x Aplicação: Run	LED DG: Pisca 4x Aplicação: Stop	LED DG: Pisca 4x Aplicação: Stop	LED DG: Pisca 2x Aplicação: Run
Configuração incompatível	LED DG: Pisca 2x Aplicação: Run	LED DG: Pisca 2x Aplicação: Run	LED DG: Pisca 2x Aplicação: Run	LED DG: Pisca 4x Aplicação: Stop	LED DG: Pisca 4x Aplicação: Stop	LED DG: Pisca 2x Aplicação: Stop
Configuração incompatível (condição partida)	LED DG: Pisca 4x Aplicação: Stop	LED DG: Pisca 4x Aplicação: Stop	LED DG: Pisca 2x Aplicação: Run ou LED DG: Pisca 4x Aplicação: Stop	LED DG: Pisca 4x Aplicação: Stop	LED DG: Pisca 4x Aplicação: Stop	LED DG: Pisca 2x Aplicação: Run
Endereço de slot duplicado	LED DG: Pisca 4x Aplicação: Stop	LED DG: Pisca 4x Aplicação: Stop	LED DG: Pisca 4x Aplicação: Stop	LED DG: Pisca 4x Aplicação: Stop	LED DG: Pisca 4x Aplicação: Stop	LED DG: Pisca 2x Aplicação: Stop
Módulo não operacional	LED DG: Pisca 4x Aplicação: Stop	LED DG: Pisca 4x Aplicação: Stop	LED DG: Pisca 4x Aplicação: Stop	LED DG: Pisca 4x Aplicação: Stop	LED DG: Pisca 4x Aplicação: Stop	LED DG: Pisca 2x Aplicação: Stop

Tabela 46: Relação entre Condições e Troca a Quente

Nota:

Habilitada, sem consistência na partida: Quando este modo de troca a quente está configurado, em situações normais quando existir um módulo incompatível na partida do sistema a aplicação passará de Stop para Run. Contudo, se o módulo configurado for um NX5000 ou NX5001 e existir outro módulo na posição, a aplicação permanecerá em Stop.

5.2.1.2. Áreas de Memória Retentiva e Persistente

A UCP Nexto permite a utilização de variáveis simbólicas e de variáveis de saída de representação direta como variáveis retentivas ou persistentes.

As variáveis de saída de representação direta que serão retentivas ou persistentes devem ser declaradas nos *Parâmetros Gerais da UCP*, conforme descrito na seção [Configuração da UCP](#). Também podem ser atribuídos nomes simbólicos a estas variáveis de saída de representação direta através da diretiva AT, e usando a palavra-chave RETAIN ou PERSISTENT no sua declaração. Por exemplo, estando %QB4096 e %QB20480, dentro das áreas de memória retentiva e persistente, respectivamente:

```
PROGRAM UserPrg
VAR RETAIN
byVariavelRetentiva_01 AT %QB4096 : BYTE;
END_VAR
VAR PERSISTENT
byVariavelPersistente_01 AT %QB20480 : BYTE;
END_VAR
```

Caso as variáveis simbólicas que utilizam a diretiva AT não sejam declaradas dentro das áreas respectivas da memória retentiva e/ou persistente, erros durante a geração de código no MasterTool podem ser apresentados. Informando que existem variáveis não retentivas ou não persistentes definidas em espaços de memória retentiva ou persistente.

Quanto às variáveis simbólicas que serão retentivas ou persistentes, apenas as variáveis retentivas poderão ser locais ou globais, sendo que variáveis simbólicas persistentes deverão sempre ser globais. Para a declaração de variáveis simbólicas retentivas, deve ser utilizada a palavra-chave *RETAIN*. Por exemplo, para variáveis locais:

```
PROGRAM UserPrg
VAR RETAIN
wVariavelSimbolicaRetentivaLocal_01 : WORD;
END_VAR
```

Ou, para variáveis globais, declaradas dentro de uma lista de variáveis globais:

```
VAR_GLOBAL RETAIN
wVariavelSimbolicaRetentivaGlobal_01 : WORD;
END_VAR
```

Já as variáveis simbólicas persistentes deverão ser declaradas dentro de um objeto Variáveis Persistentes, adicionado à aplicação. Estas variáveis serão globais e estarão declaradas da seguinte forma dentro do objeto:

```
VAR_GLOBAL PERSISTENT RETAIN
wVariavelSimbolicaPersistenteGlobal_01 : WORD;
END_VAR
```

A partir da versão 1.5.1.1 esta UCP permite flexibilidade no uso das memórias retentivas e persistentes. Isto significa que o usuário poderá optar o tamanho que irá utilizar em cada tipo de memória, desde que o somatório do tamanho das memórias retentivas e persistentes não ultrapasse o limite total disponível para cada modelo de UCP. O total de memória retentiva e persistente disponível está descrito na Tabela 7 em [Memória](#).

Caso o somatório de memória simbólica retentiva, simbólica persistente, retentiva em %Q e persistente em %Q ultrapassar o total disponível, o MasterTool apresentará erro durante a geração de código.

```

VAR_GLOBAL PERSISTENT RETAIN
wVariavelSimbolicaPersistenteGlobal_01 : WORD;
END_VAR

VAR_GLOBAL RETAIN
wVariavelSimbolicaRetentivaGlobal_01 : WORD;
END_VAR
    
```

ATENÇÃO

Para utilizar as memórias retentivas e persistentes de forma flexível é necessário utilizar o MasterTool IEC XE 2.03 ou superior.

5.2.1.3. Parâmetros do Projeto

Os parâmetros do projeto da UCP são relacionados à configurações para atualização das entradas e saídas nas tarefas em que estas são utilizadas e opções de gravação e leitura do cartão de memória.

Configuração	Descrição	Padrão	Opções
Habilita atualização de E/S por tarefa	Atualiza as entradas e saídas nas tarefas em que estas são utilizadas	Desmarcado	- Marcado - Desmarcado
Habilita variáveis retentivas e persistentes em Blocos Funcionais	Configuração para permitir o uso de variáveis retentivas e persistentes em Blocos Funcionais	Desmarcado	- Marcado - Desmarcado
Cartão de Memória			
Copiar Projeto da UCP para o Cartão de Memória	Copia o projeto da memória interna da UCP para o cartão de memória	Desabilitado	- Habilitado: Configuração habilitada - Desabilitado: Configuração desabilitada
Senha para Copiar Projeto da UCP para o Cartão de Memória	Senha para copiar o projeto da memória interna da UCP para o cartão de memória	-	Senha de 6 dígitos (0 a 999999)
Copiar Projeto do Cartão de Memória para a UCP	Copia o projeto do cartão de memória para a memória interna da UCP	Desabilitado	- Habilitado: Configuração habilitada - Desabilitado: Configuração desabilitada
Senha para Copiar Projeto do Cartão de Memória para a UCP	Senha para copiar o projeto do cartão de memória para a memória interna da UCP	-	Senha de 6 dígitos (0 a 999999)

Tabela 47: Parâmetros do Projeto da UCP

ATENÇÃO

Após configurar as possibilidades de cópia do projeto e de ter criado a aplicação de inicialização, deve-se localizar o arquivo “*Application.crc*” para que as configurações referentes ao cartão de memória tenham efeito. A localização pode ser realizada em *Selecionar a Application.crc* através do botão *Localizar Arquivo...*, como pode ser visto na figura 130.

5.2.2. Configuração de Evento Externo

O evento externo é uma característica disponível na UCP a qual permite que uma entrada digital, configurada pelo usuário, quando acionada, dispare a execução de uma tarefa específica com código definido pelo usuário. Desta forma, é possível que através desta entrada, quando acionada, interrompa a execução da aplicação principal e execute a aplicação definida na tarefa *ExternInterruptTask00*, que possui maior prioridade que as demais tarefas de aplicação. Devido as entradas e saídas serem atualizadas no contexto da tarefa *MainTask*, a tarefa de Evento Externo não possui os dados das entradas e saídas atualizados no momento da sua chamada. Caso necessário, usar as funções de atualização de E/S.

É importante também observar que, para evitar a geração de vários eventos em um espaço muito curto de tempo foi limitado o tratamento deste tipo de evento a cada 10 ms, ou seja, caso ocorra dois ou mais eventos durante 10 ms após o primeiro evento, o segundo e demais eventos serão descartados. Esta limitação foi imposta para evitar que um evento externo, que seja gerado de forma descontrolada, não bloqueie a UCP, visto que a sua tarefa possui maior prioridade frente às demais.

Para configurar um evento externo é necessário inserir um módulo de entrada digital e realizar as configurações descritas a seguir, na UCP, através do software de programação MT8500.

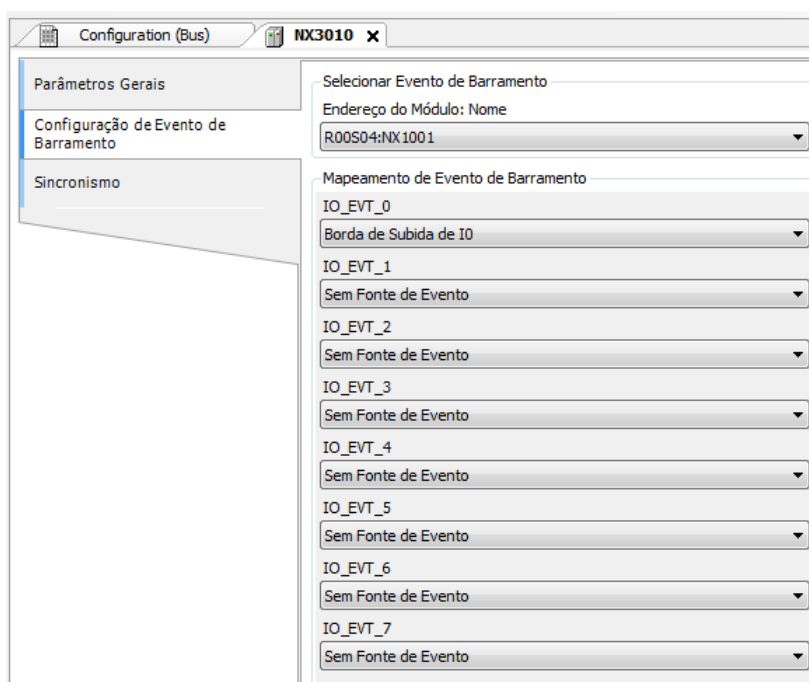


Figura 44: Tela de Configuração de Evento Externo na UCP

Na aba de configuração de evento externo, dentro das configurações da UCP, é preciso selecionar qual módulo será a fonte de interrupção no campo *Endereço do Módulo: Nome*, a seguir deve ser selecionado qual entrada deste módulo será responsável por gerar o evento (*IO_EVT_0*), nesta seleção é possível optar pelas opções descritas na figura abaixo.

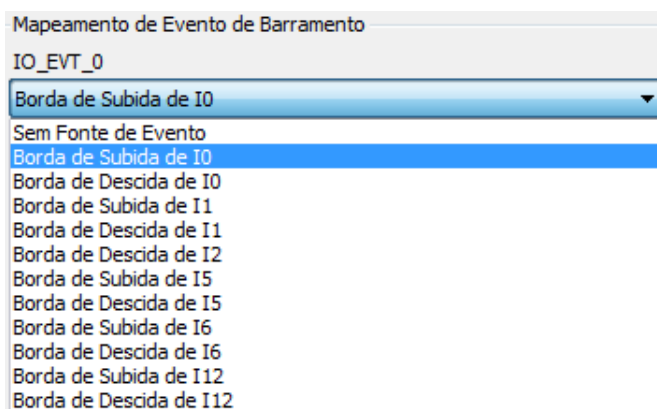


Figura 45: Opções de Origem do Evento Externo no Módulo NX1001

Além da configuração da UCP é necessário configurar a tarefa responsável por executar as ações definidas pelo usuário, neste caso o usuário deve utilizar um perfil de projeto que suporte eventos externos, mais informações consulte a seção [Perfis de Projeto](#). Na tela de configuração da tarefa *ExternInterruptTask00*, na figura abaixo, é necessário selecionar a origem do evento, no campo *Evento*, neste caso deve ser selecionado *IO_EVT_0*, pois as demais fontes de origem *IO_EVT_1* a *IO_EVT_7* não estão disponíveis. Na sequência deve ser verificado se a POU selecionada no campo *POUs* está correta, pois a mesma será utilizada pelo usuário para definir as ações que serão executadas quando um evento externo ocorrer.

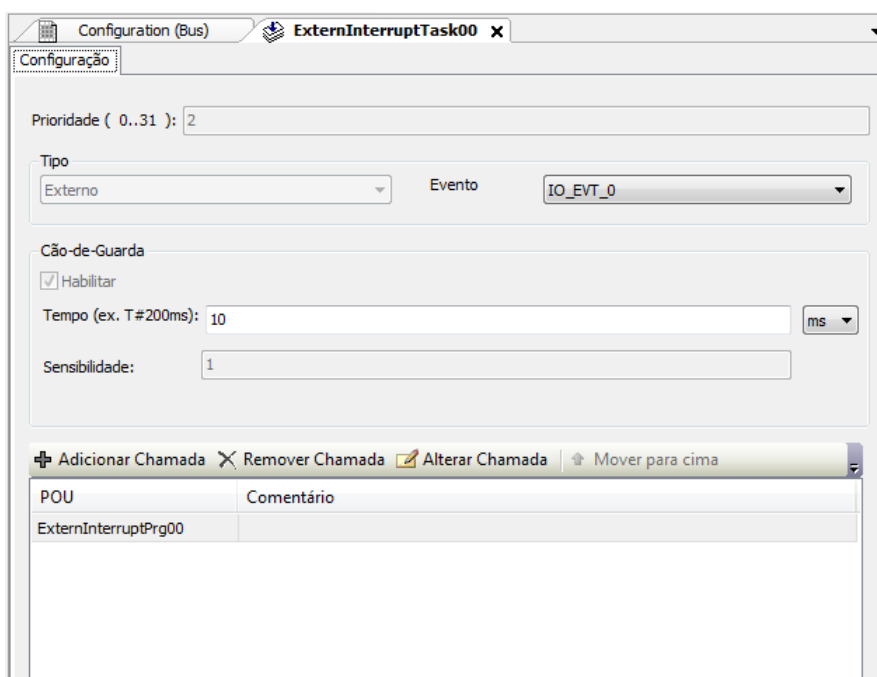


Figura 46: Tela de Configuração da Tarefa ExternInterruptTask00

5.2.3. Configuração do SOE

O SOE (*Sequence of Events*) é responsável pela geração de uma sequência de eventos digitais. Através do mesmo, é possível analisar o histórico comportamental das variáveis do sistema mapeadas na sua área de monitoramento. O SOE é um serviço exclusivo disponível nos modelos NX3020 e NX3030.

Uma vez habilitado o serviço SOE, a UCP passa a se comportar como um servidor DNP3, sendo assim, necessário suporte ao protocolo DNP3 por parte do cliente para utilização deste recurso. Os tipos de objetos suportados bem como os códigos de função e os qualificadores podem ser encontrados no [Anexo - Interoperabilidade DNP3](#).

O serviço SOE faz a utilização de endereços %Q para constituir sua base de dados estáticos. Para isso, deve ser configurada uma área contínua de memória %Q onde o usuário irá informar seu início e tamanho. Para projetos redundantes a área de %Q também deve ser redundante, para que, no momento do switchover, a base de dados do servidor DNP3 seja mantida.

O endereço do primeiro objeto DNP3 sempre será 0, correspondendo ao bit 0 do %QBxxxx, onde xxxx é o endereço inicial do %Q.

Dessa maneira, definida a base de dados estática, o usuário deve copiar cada ponto digital que deve gerar eventos para dentro da área contínua de %Q. O número máximo de pontos que podem ser copiados é 8000.

Para a configuração dos eventos, é necessário informar apenas o tamanho da fila de eventos. O SOE usa uma fila de eventos especial e dedicada (não a que está descrita na seção [Configuração de Protocolos](#)), a qual é persistente e redundante, desse modo, não serão perdidos eventos no momento do switchover, ou no momento de uma falha na alimentação. Caso ocorra um estouro na fila de eventos, os eventos mais antigos serão sobrescritos. Caso em um único ciclo sejam gerados mais eventos do que o suportado pela fila, sua geração é interrompida e o diagnóstico de estouro é ligado (SOE[x].bOverflowStatus). Por exemplo, se 100+n bits variam em uma configuração de 100 eventos, ocorre um descarte de n eventos.

O SOE irá rodar no contexto da tarefa MainTask, já a partir do primeiro ciclo. O SOE irá ser executado ao final de cada ciclo da tarefa e comparará os bits mapeados, a fim de detectar transições ocorridas durante o ciclo. Sendo assim, a cada ciclo em que eventos sejam gerados, irá ocorrer um acréscimo de tempo neste ciclo da MainTask. Para o pior caso (1000 eventos, sendo que serão gerados apenas 1000 e o restante descartado), esta influência será de aproximadamente 5 ms. Portanto, para uma aplicação com o SOE habilitado, o usuário deverá levar em consideração este tempo ao configurar os parâmetros de tempo de cão-de-guarda e intervalo da tarefa MainTask.

Para a utilização do mesmo o usuário deverá configurar os seguintes parâmetros na aba Configuração do SOE:

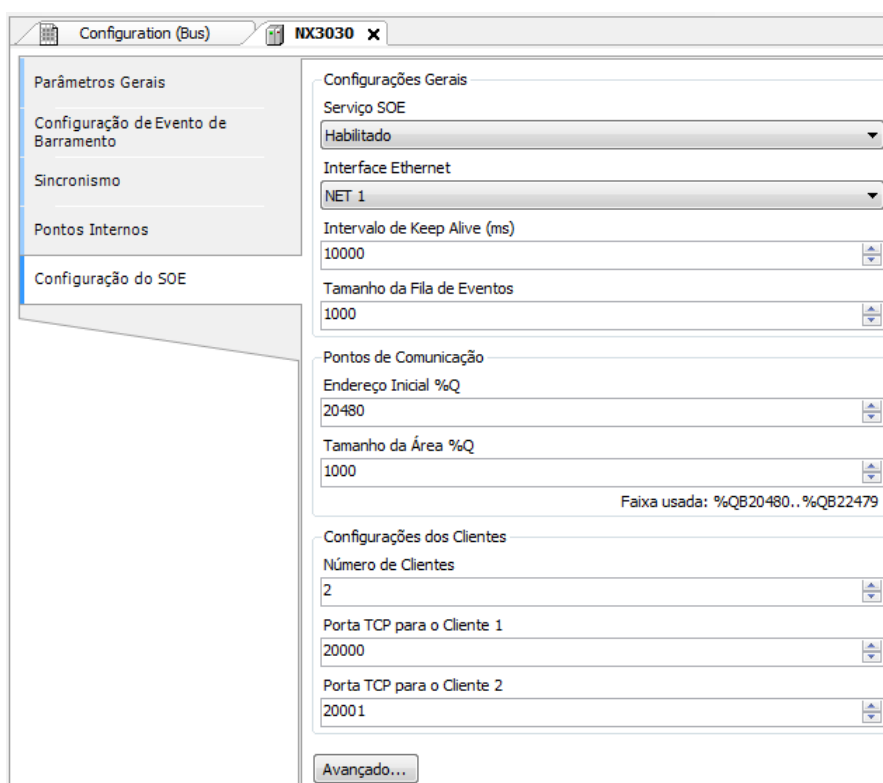


Figura 47: Configuração da Sequência de Eventos

Configuração	Descrição	Padrão	Opções
	Configurações Gerais		
Serviço SOE	Ativa o SOE	Desabilitado	Habilitado Desabilitado
Interface Ethernet	Seleciona a interface utilizada	NET 1	NET 1 NET 2
Intervalo de Keep Alive	Intervalo das mensagens de keep alive (ms)	10000	0 a 4294967295
Tamanho da Fila de Eventos	Tamanho da fila de eventos	1000	100 a 1000
	Pontos de Comunicação		
Endereço Inicial %Q	Endereço inicial para dados estáticos	20480	Pode ser utilizado qualquer endereço da área de %Q
Tamanho da Área %Q	Tamanho da memória a ser utilizada pelos dados estáticos (%Q)	1000	1 a 1000
	Configurações dos Clientes		
Número de Clientes	Define o número de clientes	2	1, 2
Porta TCP para o Cliente 1	Seleciona a porta de comunicação para o primeiro cliente	20000	1 a 65535
Porta TCP para o Cliente 2	Seleciona a porta de comunicação para o segundo cliente	20001	1 a 65535

Tabela 48: Configuração do SOE

Notas:

Tamanho da Memória de Dados: O tamanho da área de memória reservada para ser utilizada pelos dados estáticos será sempre o dobro do valor configurado, pois a segunda metade da área de memória é utilizada para armazenar os valores anteriores das variáveis da primeira metade.

Keep Alive: Enquanto estiver conectado com um cliente, serão enviadas mensagens de keep alive em intervalos, conforme o configurado. Caso o cliente não responda a estas mensagens, a conexão é fechada. Ou seja, uma conexão entre cliente e servidor poderá levar um tempo igual ao intervalo configurado para ser fechada em caso de erro.

Nas opções avançadas (botão *Avançado...*) é possível configurar os endereços de comunicação referentes ao protocolo DNP3.

Configuração	Descrição	Padrão	Opções
Endereço DNP3 da Origem	Endereço de Origem (CP)	4	0 a 65519
Endereço DNP3 do Destino do Cliente 1	Endereço do primeiro cliente	3	0 a 65519
Endereço DNP3 do Destino do Cliente 2	Endereço do segundo cliente	3	0 a 65519

Tabela 49: Configurações Avançadas do SOE

Nota:

Endereço DNP3: Os endereços DNP3 da faixa de 65520 a 65535 não podem ser configurados na origem ou em um destino, pois os mesmos são utilizados para mensagens em broadcast.

ATENÇÃO

As mensagens de *Data Link* do DNP3 não são utilizadas pelas UCPs da série Nexto, pois a norma não recomenda o seu uso ao comunicar por TCP/IP.

5.2.4. Sincronização de Tempo

Para a sincronização de tempo, as UCPs da série Nexto usam o protocolo SNTP (*Simple Network Time Protocol*) ou sincronismo pelo protocolo IEC 60870-5-104.

Para a utilização de protocolos de sincronismo de tempo, o usuário deve configurar os seguintes parâmetros na aba de *Sincronismo*, acessado através da UCP, na árvore de dispositivos:

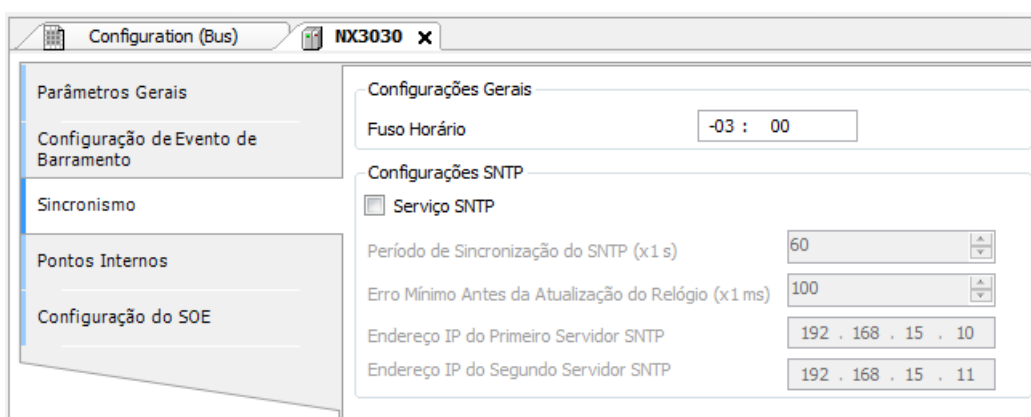


Figura 48: Configuração de SNTP

Configuração	Descrição	Padrão	Opções
Fuso Horário (hh:mm)	Fuso horário de sua localidade. Pode ser inserida a hora e o minuto.	-3:00	12:59 a +13:59
Serviço SNTP	Ativa o serviço SNTP.	Desabilitado	Desabilitado Habilitado
Período de Sincronização do SNTP (x1 s)	Intervalo de tempo das requisições de sincronização (segundos).	60	1 a 255
Erro Mínimo Antes da Atualização do Relógio (x1 ms)	Valor de offset aceitável entre a hora do servidor e do cliente (milissegundos).	100	1 a 65519
Endereço IP do Primeiro Servidor SNTP	Endereço IP do servidor SNTP primário.	192.168.15.10	1.0.0.1 a 223.255.255.254
Endereço IP do Segundo Servidor SNTP	Endereço IP do servidor SNTP secundário.	192.168.15.11	1.0.0.1 a 223.255.255.254

Tabela 50: Configurações SNTP

Notas:

Servidor SNTP: É possível definir um endereço preferencial e outro secundário para acessar um servidor SNTP e, assim, obter sincronismo de tempo.

Fuso Horário: A configuração de fuso horário é utilizada para converter o horário local em horário UTC e vice-versa. Enquanto algumas fontes de sincronismo utilizam o horário local (protocolo IEC 60870-5-104, função *SetDateAndTime*), outras utilizam o horário UTC (SNTP). O horário UTC normalmente é utilizado para estampar os eventos (DNP3, IEC 60870-5-104,

MasterTool Device LOG), enquanto que o horário local é utilizado por outras funcionalidades da UCP (função GetDateAndTime, informação de data e hora do OTD).

É permitido habilitar mais de uma fonte de sincronismo no projeto, no entanto o equipamento não suporta durante a operação a sincronização do tempo por duas fontes distintas de sincronismo, para tanto existe implicitamente definido um mecanismo de prioridade. A sincronização do protocolo SNTP é mais prioritária que a sincronização através do protocolo IEC 60870-5-104. Assim, quando habilitadas ambas as fontes e o servidor SNTP estiver presente este será responsável pela sincronização do relógio da UCP e qualquer comando de sincronismo pelo protocolo IEC 60870-5-104 será negado.

5.2.4.1. IEC 60870-5-104

No caso do sincronismo através do protocolo IEC 60870-5-104, o usuário deve habilitar o sincronismo de tempo na tela de configuração do protocolo para receber a sincronização de relógio. Para configurar o dispositivo com essa opção, consultar o parâmetro *Habilitar Sincronismo* disponível na seção [Camada de Aplicação](#).

ATENÇÃO

Se o CP receber um comando de sincronismo de tempo do centro de controle, e esta opção estiver desabilitada, será retornada uma resposta de erro para esse comando. Se a opção estiver habilitada, o CP irá retornar sucesso ao centro de controle, mesmo que o comando de sincronismo seja descartado por existir um método de sincronismo ativo com maior prioridade.

Este método de sincronismo deve ser utilizado apenas como um método auxiliar de sincronismo, uma vez que a precisão do processo de sincronização do relógio do CP depende muito do atraso e tráfego de rede, bem como da carga de processamento da UCP, uma vez que este mecanismo é tratado por uma tarefa de baixa prioridade.

ATENÇÃO

Em arquiteturas com redundância de CP, o driver IEC 60870-5-104 Servidor é desabilitado no CP não ativo. Desta forma, não é recomendada a utilização deste método de sincronismo em sistemas redundantes, pois o CP não ativo pode levar vários segundos após um switchover até que o seu relógio seja sincronizado. Em sistemas redundantes, recomenda-se a utilização de SNTP.

5.2.4.2. SNTP

Para o sincronismo via SNTP a UCP irá se comportar como um cliente SNTP, ou seja, enviará requisições de sincronização de tempo para um servidor SNTP/NTP, que pode estar na rede local ou na internet. O cliente SNTP trabalha com uma resolução de 1 ms, porém com uma precisão de 100 ms. A precisão do sincronismo de tempo por SNTP depende das configurações do protocolo SNTP (erro mínimo para atualização do relógio) e das características da rede Ethernet onde está sendo aplicado, se cliente e servidor SNTP estão na mesma rede (local) ou em redes diferentes (remota). Tipicamente a precisão é da ordem de dezenas de milissegundos.

A UCP envia as requisições de sincronização cíclicas, de acordo com o tempo configurado no campo Período de Sincronização do SNTP. Na primeira tentativa de sincronização, logo após a inicialização do serviço, a requisição é para o primeiro servidor, configurado em Endereço de IP do Primeiro Servidor. Caso este não responda, as requisições são direcionadas para o segundo servidor configurado em Endereço de IP do Segundo Servidor, fornecendo uma redundância de servidores SNTP. Caso o segundo servidor também não responda, o mesmo processo de tentativa de sincronização é executado novamente, mas apenas após o *Período de Sincronização* ter passado. Ou seja, a cada período de sincronização, a UCP tenta se conectar uma vez em cada servidor, ela tenta o segundo caso o primeiro não responda. O tempo de espera por uma resposta do servidor SNTP é definido por padrão em 5 segundos e não pode ser modificado.

Caso, após uma requisição de sincronização, a diferença entre o horário atual da UCP e o recebido pelo servidor for maior que o valor configurado no parâmetro *Erro Mínimo Antes da Atualização do Relógio*, o horário da UCP é atualizado. O SNTP usa o horário no formato UTC (Universal Time Coordinated), logo o parâmetro de *Fuso Horário* deve ser configurado corretamente, para que o horário lido pelo SNTP seja convertido corretamente para a hora local.

O Processo de execução do cliente SNTP pode ser exemplificado com os seguintes passos:

1. Tentativa de sincronização através do primeiro servidor. Caso a sincronização ocorra com sucesso, a UCP aguarda o tempo para a nova sincronização (*Período de Sincronização*) e tentará sincronizar-se novamente com este servidor, utilizando, então, este como servidor primário. Em caso de falha (o servidor não responde em menos de 5 s) o passo 2 é executado.

2. Tentativa de sincronização através do segundo servidor. Caso a sincronização ocorra com sucesso, a UCP aguarda o tempo para a nova sincronização (*Período de Sincronização*) e tentará sincronizar-se novamente com este servidor, utilizando, então, este como servidor primário. Em caso de falha (o servidor não responde em menos de 5 s) é aguardado o tempo referente ao Período de Sincronização e executado novamente o passo 1.

Como o tempo de espera pela resposta do servidor SNTP é de 5 s, deve-se prestar atenção ao configurar valores menores do que 10 s para o Período de Sincronização. Caso o servidor primário não responda, o tempo para a sincronização irá ser de, no mínimo, 5 s (aguardo da resposta do servidor primário e tentativa de sincronização com o servidor secundário). Caso nem o servidor primário nem o secundário respondam, o tempo para a sincronização irá ser de, no mínimo, 10 s (aguardo da resposta dos dois servidores e nova tentativa de conexão com o 1º servidor).

Dependendo da subrede do servidor SNTP, o cliente irá utilizar a interface Ethernet que esteja na subrede correspondente para fazer as requisições de sincronismo. Caso não haja uma interface configurada na mesma subrede do servidor, a requisição poderá ser feita por qualquer interface que possa achar uma rota para o servidor.

ATENÇÃO

O Serviço SNTP depende da aplicação do usuário apenas para a sua configuração. Portanto, este serviço vai ser executado mesmo quando a UCP estiver nos modos *STOP* ou *BREAKPOINT*, desde que exista uma aplicação na UCP com o cliente SNTP habilitado e corretamente configurado.

CUIDADO

É vital a configuração de, no mínimo, um servidor SNTP. O recomendável é configurar dois servidores SNTP (primário e secundário). O sincronismo SNTP é necessário para gerar eventos com estampa de tempo coerente entre o CPA e CPB e com a hora mundial. Outra utilidade é evitar descontinuidades durante um switchover em aplicações que referenciam data e hora, tendo em vista que não existe sincronismo de data e hora entre os CPs através dos canais de sincronismo NETA e NETB.

5.2.4.3. Horário de Verão

A configuração do horário de verão deve ser feita indiretamente, através da função *SetTimeZone*, que altera o fuso horário aplicado ao RTC. No início do horário de verão, deve-se usar a função para aumentar em uma hora o fuso horário. Ao final do horário de verão, ela é usada novamente para diminuir-lo em uma hora.

Para maiores informações, consultar a seção [Relógio RTC](#).

5.2.5. Pontos Internos

Um ponto de comunicação é armazenado na memória do CP sob a forma de duas variáveis distintas. Uma delas representa o valor do ponto (tipo BOOL, BYTE, WORD, etc...), enquanto a outra representa a sua qualidade (tipo QUALITY). Pontos Internos são aqueles cujo valor e qualidade são calculados internamente pela aplicação do usuário, isto é, não possuem uma origem externa como ocorre para pontos associados a IEDs (drivers de comunicação do tipo Mestre/Cliente) ou a módulos de E/S do barramento local.

ATENÇÃO

Diferente do que acontece com os módulos de E/S declarados no barramento local, que têm suas variáveis de qualidade criadas pelo MasterTool (GVL IOQualities) e automaticamente atualizadas pela UCP, os módulos de E/S declarados em remotas PROFIBUS não as têm. É de responsabilidade do usuário a declaração das variáveis de qualidade dos pontos PROFIBUS, a associação destas variáveis de qualidade com as variáveis de valor na aba de Pontos Internos, bem como a geração e atualização do valor das variáveis de qualidade a partir dos diagnósticos PROFIBUS existentes: módulos de E/S PROFIBUS, cabeça PROFIBUS e mestre PROFIBUS.

A função desta aba de configuração de *Pontos Internos* é relacionar a variável que representa o valor de um ponto com a variável que representa a sua qualidade. Deve ser utilizada para relacionar variáveis de valor e qualidade criadas internamente no programa do CP (como em uma GVL por exemplo), as quais tipicamente serão mapeadas posteriormente para um driver de comunicação do tipo Servidor para comunicação com o centro de controle.

ATENÇÃO

Se uma variável de valor não possuir uma variável de qualidade relacionada, será reportada uma qualidade padrão constante boa (nenhuma indicação significativa) quando a variável de valor for reportada para um cliente ou centro de controle.

Desta forma, o objetivo desta aba não é criar ou declarar pontos internos. Para realizar isto, basta declarar as variáveis de valor e/ou qualidade em uma GVL e mapeá-la no driver de comunicação.

A configuração dos pontos internos, visualizada na figura abaixo, segue os parâmetros descritos na tabela abaixo. É possível a configuração de até 5120 entradas na tabela de *Pontos Internos*.

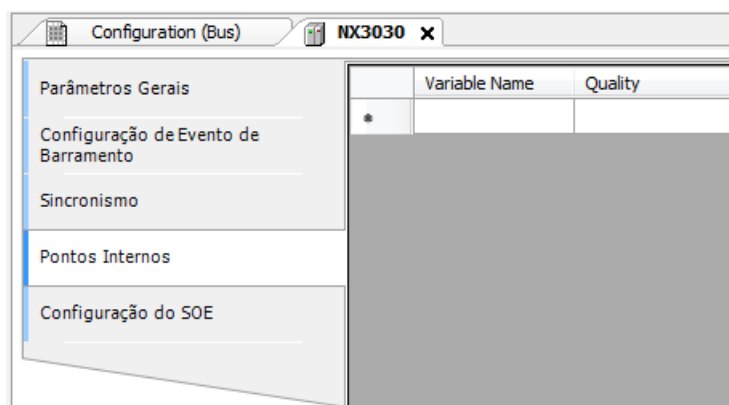


Figura 49: Tela de Configuração dos Pontos Internos

Configuração	Descrição	Padrão	Opções
Variable Name	Variável simbólica que armazena o valor do ponto interno.	-	Aceita variáveis do tipo BOOL, WORD, DWORD, LWORD, INT, DINT, LINT, UINT, UDINT, ULINT, REAL, LREAL ou DBP. A variável pode ser simples, array ou elemento de array e pode estar em estruturas.
Quality	Variável simbólica que armazena a qualidade do ponto interno representado em "Variable Name".	-	Variáveis do tipo QUALITY (LibRtuStandard), que podem ser simples, array ou elemento de array e podem estar em estruturas.

Tabela 51: Configuração dos Pontos Internos

A figura abaixo mostra um exemplo de configuração de dois pontos internos:

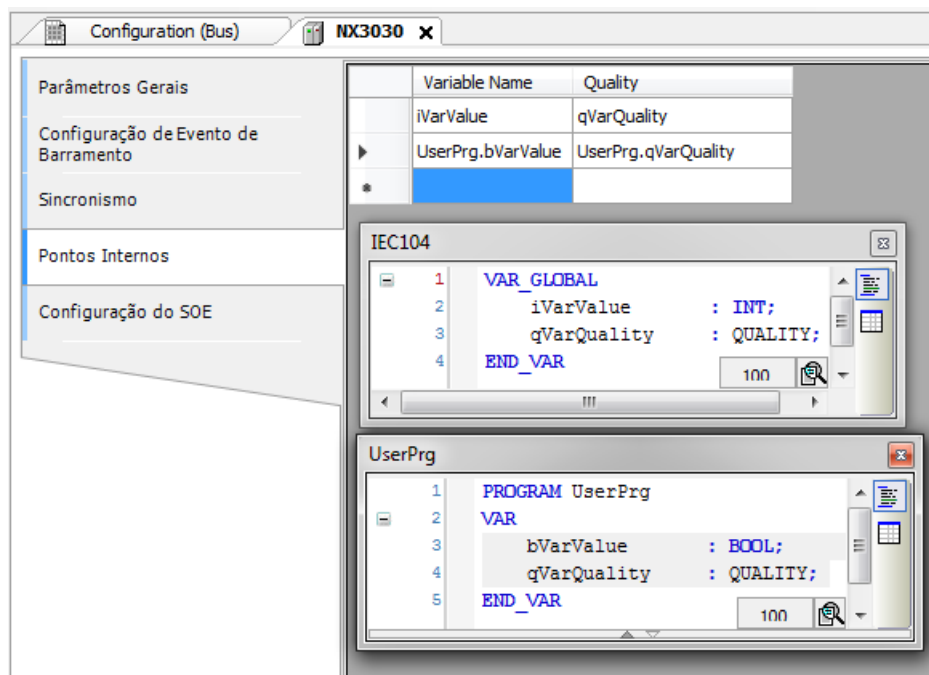


Figura 50: Exemplo de Configuração dos Pontos Internos

5.2.5.1. Conversões de Qualidade

A qualidade de um ponto interno é uma informação que indica o nível de confiança que se pode ter no valor que está armazenado naquele ponto. A qualidade pode informar, por exemplo, que o valor armazenado está fora de escala, ou ainda que seja válido, mas pouco confiável.

As normas IEC 61850, DNP3 e IEC104 possuem os seus próprios formatos para representação da informação de qualidade de um ponto. A Série Nexto, por sua vez, possui um formato de qualidade próprio (mas muito similar ao IEC 61850) chamado de *Qualidade Interna*. Este formato é definido pelo tipo QUALITY (biblioteca LibRtuStandard) e é utilizado internamente para armazenamento da qualidade, permitindo que sejam realizadas conversões entre protocolos sem que haja perda de informação.

Quando se realiza o mapeamento de um mesmo ponto de comunicação entre dois drivers, a conversão de qualidade é realizada automaticamente em dois estágios. Por exemplo: caso um ponto de comunicação seja mapeado de um driver DNP3 Cliente para um driver IEC104 Servidor, primeiro a qualidade será convertida do formato DNP3 para o formato interno (e armazenada internamente na UCP), e depois será convertida do formato interno para o formato IEC104.

As tabelas a seguir definem as conversões dos formatos proprietários dos protocolos para o formato interno. Caso seja necessário consultar a conversão entre os protocolos, é necessário realizar a análise em dois estágios consultando cada uma das tabelas para o formato interno e depois realizando a correlação entre elas.

ATENÇÃO

No caso de pontos internos mapeados para drivers de comunicação, não é recomendado modificar o valor das flags de qualidade que não possuem correspondência no protocolo em questão (isto é, que não estão nas tabelas a seguir). Isto resultará na geração de eventos iguais ao anterior (com estampa de tempo mais recente) e, desta forma, dependendo da configuração selecionada para o modo de transmissão dos eventos de entradas analógicas, poderá sobrescrever o evento anterior caso este ainda não tenha sido entregue ao centro de controle.

5. CONFIGURAÇÃO

5.2.5.1.1. Qualidade Interna

Esta é a estrutura QUALITY, a tabela abaixo mostra detalhadamente cada um de seus componentes.

Bit	Nome	Tipo	Descrição
0	FLAG_RESTART	BOOL	O identificador de RESTART indica que o dado não foi atualizado pelo campo desde o Reset do dispositivo.
1	FLAG_COMM_FAIL	BOOL	Indica que existe uma falha de comunicação no caminho entre o dispositivo onde o dado é originado e o dispositivo de relatórios.
2	FLAG_REMOTE_SUBSTITUTED	BOOL	Se TRUE, o valor dos dados é substituído em um dispositivo de comunicação remota.
3	FLAG_LOCAL_SUBSTITUTED	BOOL	Se TRUE, o valor do dado é substituído pelo dispositivo que gerou esse identificador. Este comportamento pode ter ocorrido devido a um funcionamento em diagnóstico ou temporário devido à intervenção humana.
4	FLAG_FILTER	BOOL	Identificador utilizado para sinalizar e prevenir a sobrecarga dos canais de comunicação de eventos, como oscilações (mudanças rápidas) nas entradas binárias.
5	FLAG_OVERFLOW	BOOL	Este identificador deve indicar um problema de qualidade, de que o valor, do atributo ao qual a qualidade for associada, está além da capacidade de representação.
6	FLAG_REFERENCE_ERROR	BOOL	Este identificador deve indicar que o valor pode não ser um valor correto devido à referência estar fora de calibração.
7	FLAG_INCONSISTENT	BOOL	Este identificador deve indicar que uma função de avaliação detectou uma inconsistência.
8	FLAG_OUT_OF_RANGE	BOOL	Este identificador deve indicar um erro de qualidade que o atributo ao qual a qualidade foi associada está além da capacidade de valores pré-definidos.
9	FLAG_INACCURATE	BOOL	Este identificador deve indicar que o valor não atende a precisão declarada da fonte.
10	FLAG_OLD_DATA	BOOL	Um valor parece estar desatualizado, caso uma atualização não ocorra em um período de tempo específico.
11	FLAG_FAILURE	BOOL	Este identificador deve indicar que uma função de supervisão detectou uma falha interna ou externa.
12	FLAG_OPERATOR_BLOCKED	BOOL	Atualização foi bloqueada pelo operador.
13	FLAG_TEST	BOOL	Test deve ser um identificador adicional que pode ser usado para classificar um valor sendo um valor teste e que não será usado para fins operacionais.
14-15	RESERVED	-	Reservado

Bit	Nome	Tipo	Descrição
16-17	VALIDITY	QUALITY_VALIDITY	0 – Good (Valor confiável, significa que não há condição anormal) 1 – Invalid (Valor não corresponde ao valor no IED) 2 – Reserved (Reservado) 3 – Questionable (Valor presente pode não ser o mesmo do IED)

Tabela 52: Estrutura QUALITY

5.2.5.1.2. Conversão IEC 60870-5-104

As tabelas abaixo apresentam a conversão de pontos internos digitais, analógicos, Step Position, Bitstring e contadores para IEC 60870-5-104 da Série Nexto disponíveis para o MT8500.

Pontos Internos -> IEC 60870-5-104 Digital		
Qualidade Interna		
Identificadores	VALIDITY	Qualidade IEC 60870-5-104
FLAG_RESTART	ANY	NOT TOPICAL
FLAG_COMM_FAIL	ANY	NOT TOPICAL
FLAG_REMOTE_SUBSTITUTED	ANY	SUBSTITUTED
FLAG_LOCAL_SUBSTITUTED	ANY	SUBSTITUTED
FLAG_FILTER	ANY	-
FLAG_OVERFLOW	ANY	-
FLAG_REFERENCE_ERROR	ANY	-
FLAG_INCONSISTENT	ANY	-
FLAG_OUT_OF_RANGE	ANY	-
FLAG_INACCURATE	ANY	-
FLAG_OLD_DATA	ANY	NOT TOPICAL
FLAG_FAILURE	ANY	INVALID
FLAG_OPERATOR_BLOCKED	ANY	BLOCKED
FLAG_TEST	ANY	-
ANY	VALIDITY_INVALID	INVALID

Tabela 53: Conversão Interno para IEC 60870-5-104 de Pontos Digitais

Pontos Internos -> IEC 60870-5-104 Analógicos, Step Position e Bitstring		
Qualidade Interna		
Identificadores	VALIDITY	Qualidade IEC 60870-5-104
FLAG_RESTART	ANY	NOT TOPICAL
FLAG_COMM_FAIL	ANY	NOT TOPICAL
FLAG_REMOTE_SUBSTITUTED	ANY	SUBSTITUTED
FLAG_LOCAL_SUBSTITUTED	ANY	SUBSTITUTED
FLAG_FILTER	ANY	-
FLAG_OVERFLOW	ANY	OVERFLOW
FLAG_REFERENCE_ERROR	ANY	INVALID
FLAG_INCONSISTENT	ANY	INVALID

Pontos Internos -> IEC 60870-5-104 Analógicos, Step Position e Bitstring		
Qualidade Interna		
Identificadores	VALIDITY	Qualidade IEC 60870-5-104
FLAG_OUT_OF_RANGE	ANY	OVERFLOW
FLAG_INACCURATE	ANY	INVALID
FLAG_OLD_DATA	ANY	NOT TOPICAL
FLAG_FAILURE	ANY	INVALID
FLAG_OPERATOR_BLOCKED	ANY	BLOCKED
FLAG_TEST	ANY	-
ANY	VALIDITY_INVALID	INVALID

Tabela 54: Conversão Interno para IEC 60870-5-104 de Pontos Analógicos, Step Position e Bitstring

Pontos Internos -> IEC 60870-5-104 Contadores		
Qualidade Interna		
Identificadores	VALIDITY	Qualidade IEC 60870-5-104
FLAG_RESTART	ANY	-
FLAG_COMM_FAIL	ANY	-
FLAG_REMOTE_SUBSTITUTED	ANY	-
FLAG_LOCAL_SUBSTITUTED	ANY	-
FLAG_FILTER	ANY	-
FLAG_OVERFLOW	ANY	OVERFLOW
FLAG_REFERENCE_ERROR	ANY	-
FLAG_INCONSISTENT	ANY	-
FLAG_OUT_OF_RANGE	ANY	-
FLAG_INACCURATE	ANY	-
FLAG_OLD_DATA	ANY	-
FLAG_FAILURE	ANY	INVALID
FLAG_OPERATOR_BLOCKED	ANY	-
FLAG_TEST	ANY	-
ANY	VALIDITY_INVALID	INVALID

Tabela 55: Conversão Interno para IEC 60870-5-104 de Contadores

5.2.5.1.3. Qualidade Interna MODBUS

Como a norma MODBUS não especifica tipos de qualidade para cada ponto, mas para auxílio no uso dos diagnósticos de comunicação de cada ponto, o MasterTool, através de uma estrutura interna própria, permite o mapeamento de variáveis de qualidade para cada ponto MODBUS. A tabela abaixo descreve os tipos de qualidade que cada ponto MODBUS pode assumir.

Qualidade Resultante	VALIDITY Resultante	Descrição
FLAG_RESTART	VALIDITY_INVALID	Valor de inicialização. O ponto nunca foi atualizado.
-	VALIDITY_GOOD	Comunicação OK. O ponto está atualizado.
FLAG_COMM_FAIL E FLAG_RESTART	VALIDITY_INVALID	Erro de comunicação. O ponto nunca foi atualizado.

Qualidade Resultante		VALIDITY Resultante	Descrição
FLAG_COMM_FAIL FLAG_OLD_DATA	E	VALIDITY_QUESTIONABLE	Um erro ocorreu mas o ponto foi atualizado e agora possui um valor antigo.
FLAG_FAILURE FLAG_RESTART	E	VALIDITY_INVALID	Recebeu uma resposta de exceção, e o ponto ainda está com o valor inicial.
FLAG_FAILURE FLAG_OLD_DATA	E	VALIDITY_QUESTIONABLE	Recebeu uma resposta de exceção, mas o ponto possui um valor antigo válido.
FLAG_RESTART FLAG_OLD_DATA	E	VALIDITY_QUESTIONABLE	Dispositivo está parado. O ponto contém um valor antigo.

Tabela 56: Qualidade MODBUS

5.2.5.1.4. Qualidade dos Módulos de E/S do Barramento Local

Para auxiliar no uso dos diagnósticos de cada ponto de E/S, o MasterTool cria automaticamente uma estrutura de qualidade para cada módulo do barramento local utilizado no projeto do CP, através de uma estrutura interna própria acessível via estrutura QUALITY, disponível na GVL *IOQualities*.

A tabela abaixo descreve os tipos de qualidade para cada ponto de entrada e saída.

Para mais informações consulte a seção [GVL IOQualities](#).

Diagnóstico	Flags Resultantes	VALIDITY Resultante	Descrição
Não importa	FLAG_RESTART	VALIDITY_INVALID	A qualidade tem este valor antes de ser lida ou escrita pela primeira vez.
Nenhum	-	VALIDITY_GOOD	Comunicação OK. O ponto está atualizado.
Nenhum	FLAG_OLD_DATA FLAG_FAILURE	E VALIDITY_QUESTIONABLE	Módulo não operacional, entretanto dados foram lidos/escritos pelo menos uma vez.
bOverRange OU bUnder-Range	FLAG_OUT_OF_RANGE	VALIDITY_INVALID	Valor está acima ou abaixo da faixa permitida na entrada do módulo.
bInputNotEnable OU bOutputNotEnable	FLAG_OPERATOR_BLOCKED	VALIDITY_INVALID	Entrada/Saída não está habilitada.
bOpenLoop	FLAG_FAILURE	VALIDITY_INVALID	Laço aberto em módulo de entrada.
bFatalError	FLAG_FAILURE	VALIDITY_INVALID	Falha fatal de hardware.
bNoExternalSupply	FLAG_FAILURE	VALIDITY_INVALID	Alimentação externa está abaixo do limite mínimo de funcionamento.
bShortCircuit OU bOutputShortCircuit	FLAG_FAILURE	VALIDITY_INVALID	Curto-circuito na saída.
bCalibrationError	FLAG_INACCURATE	VALIDITY_INVALID	Erro de Calibração.
bColdJunctionSensorError	FLAG_INACCURATE	VALIDITY_INVALID	Erro no Sensor de Junta Fria.

Tabela 57: Qualidade dos Módulos de E/S

5.2.5.1.5. Qualidade dos Módulos de E/S do Barramento PROFIBUS

Diferente do barramento local, o MasterTool não cria automaticamente estruturas de qualidade para os módulos do barramento PROFIBUS, e o CP também não atualiza tais estruturas. Portanto, a criação e atualização cíclica da qualidade dos módulos PROFIBUS é de responsabilidade do usuário que desenvolve a aplicação de controle.

Para auxiliar o desenvolvimento de tais aplicações, seguem exemplos práticos, em linguagem ST, para os principais tipos de módulos PROFIBUS (ED, SD, EA e SA), baseados em escravos PROFIBUS da própria Série Nexto (NX5110). O usuário deve sentir-se encorajado a fazer adaptações e alterações que julgar necessário à sua aplicação.

ATENÇÃO

Para o correto funcionamento das rotinas apresentadas a seguir, é necessário a habilitação do *Status in Diagnose* dos escravos PROFIBUS.

O desenvolvimento das rotinas de atualização da qualidade dos pontos dos módulos de E/S PROFIBUS deve iniciar pela declaração e inicialização das variáveis de qualidade a partir de uma GVL:

```
VAR_GLOBAL
QUALITY_PB_NX1005_I: LibDataTypes.QUALITY:= (VALIDITY:= VALIDITY_INVALID,
        FLAGS:= (FLAG_RESTART:= TRUE));
QUALITY_PB_NX1005_O: LibDataTypes.QUALITY:= (VALIDITY:= VALIDITY_INVALID,
        FLAGS:= (FLAG_RESTART:= TRUE));
QUALITY_PB_NX6000: LibDataTypes.QUALITY:= (VALIDITY:= VALIDITY_INVALID,
        FLAGS:= (FLAG_RESTART:= TRUE));
QUALITY_PB_NX6100: LibDataTypes.QUALITY:= (VALIDITY:= VALIDITY_INVALID,
        FLAGS:= (FLAG_RESTART:= TRUE));
END_VAR
```

5.2.5.1.6. Qualidade da Entrada Digital PROFIBUS

```
// Atualização da qualidade de uma entrada digital PROFIBUS, módulo NX1005

// Em caso de sucesso de comunicação com o escravo PROFIBUS (endereço = 99) ...
IF DG_NX5001.tMstStatus.abvSlv_State.bSlave_99 = TRUE THEN
    // Aguarda o escravo PROFIBUS ficar apto a trocar dados e diagnósticos
    // (é necessário aguardar, evitando a geração de qualidades inválidas)
    IF DG_NX5110.tPbusHeadA.tStatus1.bStation_Non_Existent = FALSE AND
        DG_NX5110.tPbusHeadA.tStatus1.bStation_Not_Ready = FALSE AND
        DG_NX5110.tPbusHeadA.wIdentNumber > 0 THEN
        QUALITY_PB_NX1005_I.FLAGS.FLAG_COMM_FAIL:= FALSE;
        // Se existe um módulo presente no barramento (slot = 2) e
        // se não existe nenhum problema de config dos módulos (geral) e
        // se não existe nenhum problema de config do módulo (específico) e
        // se não existe indicação de erro fatal pelo módulo ...
        IF (DG_NX5110.tPbusHeadA.dwModuleNotPresent AND SHL(1, 2)) = 0 AND
            DG_NX5110.tPbusHeadA.tSummarized.bConfigMismatch = FALSE AND
            DG_NX1005_24_Vdc_8_DO_Trans_8_DI.tGeneral.bConfigMismatch = FALSE AND
            DG_NX1005_24_Vdc_8_DO_Trans_8_DI.tGeneral.bFatalError = FALSE THEN
            QUALITY_PB_NX1005_I.VALIDITY:= VALIDITY_GOOD;
            QUALITY_PB_NX1005_I.FLAGS.FLAG_RESTART:= FALSE;
            QUALITY_PB_NX1005_I.FLAGS.FLAG_FAILURE:= FALSE;
            QUALITY_PB_NX1005_I.FLAGS.FLAG_OLD_DATA:= FALSE;
        ELSE
            QUALITY_PB_NX1005_I.VALIDITY:= VALIDITY_INVALID;
            QUALITY_PB_NX1005_I.FLAGS.FLAG_FAILURE:= TRUE;
            // Se o ponto já havia sido atualizado uma vez ...
            IF NOT QUALITY_PB_NX1005_I.FLAGS.FLAG_RESTART THEN
                QUALITY_PB_NX1005_I.FLAGS.FLAG_OLD_DATA:= TRUE;
            END_IF
        END_IF
    END_IF
// Em caso de falha de comunicação com o escravo PROFIBUS ...
ELSE
```

5. CONFIGURAÇÃO

```
QUALITY_PB_NX1005_I.VALIDITY:= VALIDITY_INVALID;  
QUALITY_PB_NX1005_I.FLAGS.FLAG_COMM_FAIL:= TRUE;  
QUALITY_PB_NX1005_I.FLAGS.FLAG_FAILURE:= FALSE;  
// Se o ponto já havia sido atualizado uma vez ...  
IF NOT QUALITY_PB_NX1005_I.FLAGS.FLAG_RESTART THEN  
    QUALITY_PB_NX1005_I.FLAGS.FLAG_OLD_DATA:= TRUE;  
END_IF  
END_IF
```

5.2.5.1.7. Qualidade da Saída Digital PROFIBUS

```
// Atualização da qualidade de uma saída digital PROFIBUS, módulo NX1005  
  
// Em caso de sucesso de comunicação com o escravo PROFIBUS (endereço = 99) ...  
IF DG_NX5001.tMstStatus.abvSlv_State.bSlave_99 = TRUE THEN  
// Aguarda o escravo PROFIBUS ficar apto a trocar dados e diagnósticos  
// (é necessário aguardar, evitando a geração de qualidades inválidas)  
IF DG_NX5110.tPbusHeadA.tStatus1.bStation_Non_Existent = FALSE AND  
    DG_NX5110.tPbusHeadA.tStatus1.bStation_Not_Ready = FALSE AND  
    DG_NX5110.tPbusHeadA.wIdentNumber > 0 THEN  
QUALITY_PB_NX1005_O.FLAGS.FLAG_COMM_FAIL:= FALSE;  
// Se existe um módulo presente no barramento (slot = 2) e  
// se não existe nenhum problema de config dos módulos (geral) e  
// se não existe nenhum problema de config do módulo (específico) e  
// se não existe indicação de erro fatal pelo módulo e  
// se não existe indicação de curto circuito das saídas e  
// se não existe indicação de falta de alimentação externa ...  
IF (DG_NX5110.tPbusHeadA.dwModuleNotPresent AND SHL(1, 2)) = 0 AND  
    DG_NX5110.tPbusHeadA.tSummarized.bConfigMismatch = FALSE AND  
    DG_NX1005_24_Vdc_8_DO_Trans_8_DI.tGeneral.bConfigMismatch = FALSE AND  
    DG_NX1005_24_Vdc_8_DO_Trans_8_DI.tGeneral.bFatalError = FALSE AND  
    DG_NX1005_24_Vdc_8_DO_Trans_8_DI.tDetailed.bOutputShortCircuit = FALSE AND  
    DG_NX1005_24_Vdc_8_DO_Trans_8_DI.tDetailed.bNoExternalSupply = FALSE THEN  
    QUALITY_PB_NX1005_O.VALIDITY:= VALIDITY_GOOD;  
    QUALITY_PB_NX1005_O.FLAGS.FLAG_RESTART:= FALSE;  
    QUALITY_PB_NX1005_O.FLAGS.FLAG_FAILURE:= FALSE;  
    QUALITY_PB_NX1005_O.FLAGS.FLAG_OLD_DATA:= FALSE;  
ELSE  
    QUALITY_PB_NX1005_O.VALIDITY:= VALIDITY_INVALID;  
    QUALITY_PB_NX1005_O.FLAGS.FLAG_FAILURE:= TRUE;  
    // Se o ponto já havia sido atualizado uma vez ...  
    IF NOT QUALITY_PB_NX1005_O.FLAGS.FLAG_RESTART THEN  
        QUALITY_PB_NX1005_O.FLAGS.FLAG_OLD_DATA:= TRUE;  
    END_IF  
END_IF  
END_IF  
// Em caso de falha de comunicação com o escravo PROFIBUS ...  
ELSE  
QUALITY_PB_NX1005_O.VALIDITY:= VALIDITY_INVALID;  
QUALITY_PB_NX1005_O.FLAGS.FLAG_COMM_FAIL:= TRUE;  
QUALITY_PB_NX1005_O.FLAGS.FLAG_FAILURE:= FALSE;  
// Se o ponto já havia sido atualizado uma vez ...
```

5. CONFIGURAÇÃO

```
IF NOT QUALITY_PB_NX1005_O.FLAGS.FLAG_RESTART THEN
  QUALITY_PB_NX1005_O.FLAGS.FLAG_OLD_DATA:= TRUE;
END_IF
END_IF
```

5.2.5.1.8. Qualidade da Entrada Analógica PROFIBUS

```
// Atualização da qualidade de uma entrada analógica PROFIBUS, módulo NX6000

// Em caso de sucesso de comunicação com o escravo PROFIBUS (endereço = 99) ...
IF DG_NX5001.tMstStatus.abvSlv_State.bSlave_99 = TRUE THEN
// Aguarda o escravo PROFIBUS ficar apto a trocar dados e diagnósticos
// (é necessário aguardar, evitando a geração de qualidades inválidas)
IF DG_NX5110.tPbusHeadA.tStatus1.bStation_Non_Existent = FALSE AND
  DG_NX5110.tPbusHeadA.tStatus1.bStation_Not_Ready = FALSE AND
  DG_NX5110.tPbusHeadA.wIdentNumber > 0 THEN
QUALITY_PB_NX6000.FLAGS.FLAG_COMM_FAIL:= FALSE;
// Se existe um módulo presente no barramento (slot = 3) e
// se não existe nenhum problema de config dos módulos (geral) e
// se não existe nenhum problema de config do módulo (específico) e
// se não existe indicação de erro fatal pelo módulo e
// se não existe indicação de erro de calibração e
// se não existe indicação de erro de over/under range e
// se não existe indicação de erro de entrada em laço aberto ...
IF (DG_NX5110.tPbusHeadA.dwModuleNotPresent AND SHL(1, 3)) = 0 AND
  DG_NX5110.tPbusHeadA.tSummarized.bConfigMismatch = FALSE AND
  DG_NX6000_8_AI_Voltage_Current.tGeneral.bConfigMismatch = FALSE AND
  DG_NX6000_8_AI_Voltage_Current.tGeneral.bFatalError = FALSE AND
  DG_NX6000_8_AI_Voltage_Current.tGeneral.bCalibrationError = FALSE AND
  DG_NX6000_8_AI_Voltage_Current.tDetailed.tAnalogInput_00.bOverRange = FALSE
  AND
  DG_NX6000_8_AI_Voltage_Current.tDetailed.tAnalogInput_00.bUnderRange = FALSE
  AND
  DG_NX6000_8_AI_Voltage_Current.tDetailed.tAnalogInput_00.bOpenLoop = FALSE
  THEN
  QUALITY_PB_NX6000.VALIDITY:= VALIDITY_GOOD;
  QUALITY_PB_NX6000.FLAGS.FLAG_RESTART:= FALSE;
  QUALITY_PB_NX6000.FLAGS.FLAG_FAILURE:= FALSE;
  QUALITY_PB_NX6000.FLAGS.FLAG_OLD_DATA:= FALSE;
  QUALITY_PB_NX6000.FLAGS.FLAG_INACCURATE:= FALSE;
  QUALITY_PB_NX6000.FLAGS.FLAG_OUT_OF_RANGE:= FALSE;
ELSE
  // Condição poara ligar a indicação de imprecisão
  // (avaliar primeiro, pois validade inválida deve prevalecer)
  IF DG_NX6000_8_AI_Voltage_Current.tGeneral.bCalibrationError = TRUE THEN
    QUALITY_PB_NX6000.VALIDITY:= VALIDITY_QUESTIONABLE;
    QUALITY_PB_NX6000.FLAGS.FLAG_INACCURATE:= TRUE;
  ELSE
    QUALITY_PB_NX6000.FLAGS.FLAG_INACCURATE:= FALSE;
  END_IF
  // Condição para ligar a indicação de fora de faixa
  // (avaliar primeiro, pois validade inválida deve prevalecer)
```

5. CONFIGURAÇÃO

```
IF DG_NX6000_8_AI_Voltage_Current.tDetailed.tAnalogInput_00.bOverRange =
TRUE OR
  DG_NX6000_8_AI_Voltage_Current.tDetailed.tAnalogInput_00.bUnderRange =
TRUE THEN
  QUALITY_PB_NX6000.VALIDITY:= VALIDITY_QUESTIONABLE;
  QUALITY_PB_NX6000.FLAGS.FLAG_OUT_OF_RANGE:= TRUE;
ELSE
  QUALITY_PB_NX6000.FLAGS.FLAG_OUT_OF_RANGE:= FALSE;
END_IF
// Condições para ligar a indicação de falha geral (prioritária)
IF (DG_NX5110.tPbusHeadA.dwModuleNotPresent AND SHL(1, 3)) > 0 OR
  DG_NX5110.tPbusHeadA.tSummarized.bConfigMismatch = TRUE OR
  DG_NX6000_8_AI_Voltage_Current.tGeneral.bConfigMismatch = TRUE OR
  DG_NX6000_8_AI_Voltage_Current.tGeneral.bFatalError = TRUE OR
  DG_NX6000_8_AI_Voltage_Current.tDetailed.tAnalogInput_00.bOpenLoop = TRUE
THEN
  QUALITY_PB_NX6000.VALIDITY:= VALIDITY_INVALID;
  QUALITY_PB_NX6000.FLAGS.FLAG_FAILURE:= TRUE;
  // Se o ponto já havia sido atualizado uma vez ...
  IF NOT QUALITY_PB_NX6000.FLAGS.FLAG_RESTART AND
  NOT DG_NX6000_8_AI_Voltage_Current.tDetailed.tAnalogInput_00.bOpenLoop
THEN
    QUALITY_PB_NX6000.FLAGS.FLAG_OLD_DATA:= TRUE;
  END_IF
ELSE
  QUALITY_PB_NX6000.FLAGS.FLAG_RESTART:= FALSE;
  QUALITY_PB_NX6000.FLAGS.FLAG_FAILURE:= FALSE;
  QUALITY_PB_NX6000.FLAGS.FLAG_OLD_DATA:= FALSE;
END_IF
END_IF
// Em caso de falha de comunicação com o escravo PROFIBUS ...
ELSE
  QUALITY_PB_NX6000.VALIDITY:= VALIDITY_INVALID;
  QUALITY_PB_NX6000.FLAGS.FLAG_COMM_FAIL:= TRUE;
  QUALITY_PB_NX6000.FLAGS.FLAG_FAILURE:= FALSE;
  // Se o ponto já havia sido atualizado uma vez ...
  IF NOT QUALITY_PB_NX6000.FLAGS.FLAG_RESTART AND
  NOT DG_NX6000_8_AI_Voltage_Current.tDetailed.tAnalogInput_00.bOpenLoop THEN
    QUALITY_PB_NX6000.FLAGS.FLAG_OLD_DATA:= TRUE;
  END_IF
END_IF
```

5.2.5.1.9. Qualidade da Saída Analógica PROFIBUS

```
// Atualização da qualidade de uma saída analógica PROFIBUS, módulo NX6100
// Em caso de sucesso de comunicação com o escravo PROFIBUS (endereço = 99) ...
IF DG_NX5001.tMstStatus.abvSlv_State.bSlave_99 = TRUE THEN
  // Aguarda o escravo PROFIBUS ficar apto a trocar dados e diagnósticos
  // (é necessário aguardar, evitando a geração de qualidades inválidas)
  IF DG_NX5110.tPbusHeadA.tStatus1.bStation_Non_Existent = FALSE AND
```


5. CONFIGURAÇÃO

```
DG_NX5110.tPbusHeadA.tStatus1.bStation_Not_Ready = FALSE AND
DG_NX5110.tPbusHeadA.wIdentNumber > 0 THEN
QUALITY_PB_NX6100.FLAGS.FLAG_COMM_FAIL:= FALSE;
// Se existe um módulo presente no barramento (slot = 4) e
// se não existe nenhum problema de config dos módulos (geral) e
// se não existe nenhum problema de config do módulo (específico) e
// se não existe indicação de erro fatal pelo módulo e
// se não existe indicação de erro de calibração e
// se não existe indicação de falta de alimentação externa e
// se não existe indicação de saída em laço aberto e
// se não existe indicação de curto circuito das saídas ...
IF (DG_NX5110.tPbusHeadA.dwModuleNotPresent AND SHL(1, 4)) = 0 AND
DG_NX5110.tPbusHeadA.tSummarized.bConfigMismatch = FALSE AND
DG_NX6100_4_AO_Voltage_Current.tGeneral.bConfigMismatch = FALSE AND
DG_NX6100_4_AO_Voltage_Current.tGeneral.bFatalError = FALSE AND
DG_NX6100_4_AO_Voltage_Current.tGeneral.bCalibrationError = FALSE AND
DG_NX6100_4_AO_Voltage_Current.tGeneral.bNoExternalSupply = FALSE AND
DG_NX6100_4_AO_Voltage_Current.tDetailed.tAnalogOutput_00.bOpenLoop = FALSE
AND
DG_NX6100_4_AO_Voltage_Current.tDetailed.tAnalogOutput_00.bShortCircuit =
FALSE THEN
QUALITY_PB_NX6100.VALIDITY:= VALIDITY_GOOD;
QUALITY_PB_NX6100.FLAGS.FLAG_RESTART:= FALSE;
QUALITY_PB_NX6100.FLAGS.FLAG_FAILURE:= FALSE;
QUALITY_PB_NX6100.FLAGS.FLAG_INACCURATE:= FALSE;
QUALITY_PB_NX6100.FLAGS.FLAG_OLD_DATA:= FALSE;
ELSE
// Condição para ligar a indicação de imprecisão
// (avaliar primeiro, pois validade inválida deve prevalecer)
IF DG_NX6100_4_AO_Voltage_Current.tGeneral.bCalibrationError = TRUE THEN
QUALITY_PB_NX6100.VALIDITY:= VALIDITY_QUESTIONABLE;
QUALITY_PB_NX6100.FLAGS.FLAG_INACCURATE:= TRUE;
ELSE
QUALITY_PB_NX6100.FLAGS.FLAG_INACCURATE:= FALSE;
END_IF
// Condições para ligar a indicação de falha geral (prioritária)
IF (DG_NX5110.tPbusHeadA.dwModuleNotPresent AND SHL(1, 4)) > 0 OR
DG_NX5110.tPbusHeadA.tSummarized.bConfigMismatch = TRUE OR
DG_NX6100_4_AO_Voltage_Current.tGeneral.bConfigMismatch = TRUE OR
DG_NX6100_4_AO_Voltage_Current.tGeneral.bFatalError = TRUE OR
DG_NX6100_4_AO_Voltage_Current.tGeneral.bNoExternalSupply = TRUE OR
DG_NX6100_4_AO_Voltage_Current.tDetailed.tAnalogOutput_00.bOpenLoop = TRUE
OR
DG_NX6100_4_AO_Voltage_Current.tDetailed.tAnalogOutput_00.bShortCircuit =
TRUE THEN
QUALITY_PB_NX6100.VALIDITY:= VALIDITY_INVALID;
QUALITY_PB_NX6100.FLAGS.FLAG_FAILURE:= TRUE;
// Se o ponto já havia sido atualizado uma vez ...
IF NOT QUALITY_PB_NX6100.FLAGS.FLAG_RESTART AND NOT
DG_NX6100_4_AO_Voltage_Current.tDetailed.tAnalogOutput_00.bOpenLoop
THEN
QUALITY_PB_NX6100.FLAGS.FLAG_OLD_DATA:= TRUE;
END_IF
ELSE
QUALITY_PB_NX6100.FLAGS.FLAG_RESTART:= FALSE;
QUALITY_PB_NX6100.FLAGS.FLAG_FAILURE:= FALSE;
```

```

        QUALITY_PB_NX6100.FLAGS.FLAG_OLD_DATA:= FALSE;
    END_IF
END_IF
END_IF
// Em caso de falha de comunicação com o escravo PROFIBUS ...
ELSE
QUALITY_PB_NX6100.VALIDITY:= VALIDITY_INVALID;
QUALITY_PB_NX6100.FLAGS.FLAG_COMM_FAIL:= TRUE;
QUALITY_PB_NX6100.FLAGS.FLAG_FAILURE:= FALSE;
// Se o ponto já havia sido atualizado uma vez ...
IF NOT QUALITY_PB_NX6100.FLAGS.FLAG_RESTART AND
    NOT DG_NX6100_4_AO_Voltage_Current.tDetailed.tAnalogOutput_00.bOpenLoop THEN
QUALITY_PB_NX6100.FLAGS.FLAG_OLD_DATA:= TRUE;
END_IF
END_IF

```

5.3. Configuração das Interfaces Seriais

5.3.1. COM1

A interface de comunicação COM 1, é composta por um conector tipo DB9 fêmea para o padrão RS-232C. Permite a comunicação ponto a ponto (ou em rede utilizando conversor) nos protocolos abertos MODBUS RTU escravo ou MODBUS RTU mestre.

Abaixo, segue os parâmetros que devem ser configurados para o bom funcionamento da aplicação.

Quando utilizado o protocolo MODBUS Mestre/Escravo, alguns destes parâmetros (como Modo Serial, Bits de Dado, Threshold de RX e Eventos Seriais) são ajustados automaticamente pela ferramenta MasterTool para o correto funcionamento deste protocolo.

Configuração	Descrição	Padrão	Opções
Tipo Serial	Configuração do tipo do canal serial.	RS-232C	RS-232C
Taxa de Transmissão	Velocidade da porta de comunicação serial.	115200	200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200 bps
Paridade	Configura a paridade da porta serial.	Sem Paridade	Ímpar Par Paridade Sempre Um Paridade Sempre Zero Sem Paridade
Bits de Dado	Configura o número de bits de dados em cada caractere da comunicação serial.	8	5, 6, 7 e 8
Bits de Parada	Configura os bits de parada da porta serial.	1	1, 1.5 e 2

Configuração	Descrição	Padrão	Opções
Modo Serial	Configura o modo de operação da porta serial.	Modo Normal	<ul style="list-style-type: none"> - Modo Estendido: Modo estendido de operação da comunicação serial, no qual são fornecidas informações sobre o frame de dados recebido. - Modo Normal: Modo normal de operação da comunicação serial

Tabela 58: Configurações da Interface Serial Padrão RS-232C

Notas:

Modo Estendido: Neste modo de operação da comunicação serial são fornecidas informações sobre o frame de dados recebido. As informações disponibilizadas são as seguintes:

- Um byte para o dado recebido (RX_CHAR : BYTE): Armazena os cinco, seis, sete ou oito bits dos dados recebidos, dependendo da configuração da serial.
- Um byte para os erros no sinal (RX_ERROR : BYTE): Tem o seguinte formato:
 - Bit 0: 0 - o caractere nos bits 0 a 7 é válido. 1 - o caractere nos bits 0 a 7 não é válido (ou pode não ser válido), devido aos problemas indicados nos bits 10 a 15.
 - Bit 1: Não utilizado.
 - Bit 2: Não utilizado.
 - Bit 3: Erro de interrupção na UART. A entrada serial permaneceu na lógica 0 (paridade sempre zero) por um tempo maior do que um caractere (bit de partida + bit de dados + bit de paridade + bit de parada).
 - Bit 4: Erro no frame UART. A lógica 0 (paridade sempre zero) foi lida quando o primeiro bit de parada era esperado, sendo que deveria ser lógica 1 (paridade sempre um).
 - Bit 5: Erro de paridade UART. O bit de paridade lido não está de acordo com o bit de paridade calculado.
 - Bit 6: Erro de overrun UART. Dados foram perdidos durante a leitura do FIFO UART, porque novos caracteres foram recebidos antes de os antigos serem removidos. Esse erro somente será indicado no primeiro caractere lido após a indicação do erro de overrun. Isso significa que alguns dados antigos foram perdidos.
 - Bit 7: Erro de overrun na fila RX: Esse caractere foi escrito quando a fila RX foi completa, sobrescrevendo caracteres não lidos.
- Dois bytes para o sinal de estampa de tempo (RX_TIMESTAMP : WORD): Indica o tempo de silêncio, no intervalo de 0 a 65535, usando como base de tempo 10 us. Satura em 655,35 ms, caso o tempo de silêncio seja maior do que 65535 unidades. O RX_TIMESTAMP de um caractere mede o tempo desde uma referência que pode ser uma das três opções abaixo:
 - Na maioria dos casos, o final do caractere anterior.
 - Configuração da porta serial.
 - O fim da transmissão serial usando o SERIAL_TX FB, ou seja, quando o último caractere foi enviado na linha.

Além de medir o tempo de silêncio antes de cada caractere, o RX_TIMESTAMP também é importante, pois mede o tempo de silêncio do último caractere da fila RX. A medição do silêncio é importante para implementar corretamente alguns protocolos, como MODBUS RTU. Esse protocolo especifica um interframe maior do que 3,5 caracteres e um interbyte menor do que 1,5 caracteres.

Bits de Dado: A configuração *Bits de Dado* das interfaces seriais limita os campos *Bits de Parada* e *Paridade* da comunicação, ou seja, o número de bits de parada e o método de paridade irão variar de acordo com o número de bits de dado.

Bits de Dados	Bits de Parada	Paridade
5	1, 1.5	SEM PARIDADE, ÍMPAR, PAR, PARIDADE SEMPRE UM, PARIDADE SEMPRE ZERO
6	1, 2	SEM PARIDADE, ÍMPAR, PAR, PARIDADE SEMPRE UM, PARIDADE SEMPRE ZERO
7	1, 2	SEM PARIDADE, ÍMPAR, PAR, PARIDADE SEMPRE UM, PARIDADE SEMPRE ZERO
8	1, 2	SEM PARIDADE, ÍMPAR, PAR, PARIDADE SEMPRE UM, PARIDADE SEMPRE ZERO

Tabela 59: Configurações Específicas

5.3.1.1. Configurações Avançadas

As configurações avançadas são relacionadas aos sinais de controle da comunicação serial, ou seja, quando se faz necessária a utilização de um controle mais apurado da transmissão e recepção dos dados.

Configuração	Descrição	Padrão	Opções
Parâmetros de Porta Avançados			
Sinais de modem	Realiza o controle de requisição para transmissão de um comando, através da interface RS-232C.	RTS Sempre Desligado	<ul style="list-style-type: none"> - RTS: Habilitado no início da transmissão e reiniciado, o mais rápido possível, após o final da transmissão. Ex: Controle de um conversor RS-232C/RS-485 externo. - RTS Sempre Desligado: Sempre desabilitado. - RTS Sempre Ligado: Sempre habilitado - RTS/CTS: Caso o CTS esteja desabilitado, o RTS é habilitado. Então, aguarda-se o CTS ser habilitado para a transmissão começar e o RTS é reiniciado, o mais rápido possível, no final da transmissão. Ex: Controle de rádio modems com o mesmo sinal de modem. - RTS Manual: o usuário é responsável por controlar todos os sinais de controle

Configuração	Descrição	Padrão	Opções
Threshold de RX da UART	Quantidade de bytes que devem ser recebidos para gerar uma nova interrupção na UART. Valores baixos fazem o Timestamp mais preciso quando o modo Estendido é utilizado e minimiza os erros de overrun. No entanto, valores baixos podem causar muitas interrupções que podem retardar a UCP.	8	1, 4, 8 e 14
Eventos Seriais			
RX em TX	Quando verdadeiro, todos os bytes recebidos durante a transmissão serão descartados em vez de ir para a fila de RX. Utilizado para desabilitar a operação full-duplex na interface RS-232C.	Desabilitado	- Habilitado: Configuração habilitada - Desabilitado: Configuração desabilitada
Evento RX DCD	Quando verdadeiro, gera um evento externo devido à mudança do sinal DCD	Habilitado	- Habilitado: Configuração habilitada - Desabilitado: Configuração desabilitada
Evento RX CTS	Quando verdadeiro, gera um evento externo devido à mudança do sinal CTS.	Habilitado	- Habilitado: Configuração habilitada - Desabilitado: Configuração desabilitada

Tabela 60: Configurações Avançadas da Interface Serial Padrão RS-232C

Notas:

RX em TX: Este parâmetro avançado é válido para configurações RS-232C e para RS-422.

Evento RX DCD: Os eventos externos, como o sinal de DCD da COM 1 das UCPs NX3010, NX3020 e NX3030, só podem ser associados às tarefas do perfil de projeto personalizado, descrito com detalhes no Manual de Utilização do MasterTool IEC XE – MU299048.

Evento RX CTS: Os eventos externos, como o sinal de CTS da COM 1 das UCPs NX3010, NX3020 e NX3030, só podem ser associados às tarefas do perfil de projeto personalizado, descrito com detalhes no Manual de Utilização do MasterTool IEC XE – MU299048.

5.3.2. COM 2

A configuração *Bits de Dado* das interfaces seriais limita os campos *Bits de Parada* e *Paridade* da comunicação, ou seja, o número de bits de parada e o método de paridade irão variar de acordo com o número de bits de dado.

A tabela abaixo mostra as configurações permitidas para esta interface.

Bits de Dados	Bits de Parada	Paridade
5	1, 1.5	SEM PARIDADE, ÍMPAR, PAR, PARIDADE SEMPRE UM, PARIDADE SEMPRE ZERO
6	1, 2	SEM PARIDADE, ÍMPAR, PAR, PARIDADE SEMPRE UM, PARIDADE SEMPRE ZERO
7	1, 2	SEM PARIDADE, ÍMPAR, PAR, PARIDADE SEMPRE UM, PARIDADE SEMPRE ZERO
8	1, 2	SEM PARIDADE, ÍMPAR, PAR, PARIDADE SEMPRE UM, PARIDADE SEMPRE ZERO

Tabela 61: Configurações Específicas

5.3.2.1. Configurações Avançadas

As configurações avançadas são relacionadas aos sinais de controle da comunicação serial, ou seja, quando se faz necessária a utilização de um controle mais apurado da transmissão e recepção dos dados.

Configuração	Descrição	Padrão	Opções
Threshold de RX da UART	Quantidade de bytes que devem ser recebidos para gerar uma nova interrupção na UART. Valores baixos fazem o TIMESTAMP mais preciso quando o MODO ESTENDIDO é utilizado e minimiza os erros de overrun. No entanto, valores baixos podem causar muitas interrupções que podem retardar a UCP.	8	1, 4, 8 e 14

Tabela 62: Configurações Avançadas da Interface Serial Padrões RS-422C e RS-485

5.4. Configuração das Interfaces Ethernet

A UCP NX3030 possui NET 1 e NET 2. Além das interfaces Ethernet integradas, a Série Nexto disponibiliza também interfaces Ethernet remotas através da inclusão do módulo NX5000. Os módulos NX5000 possuem apenas a interface NET 1.

5.4.1. Interfaces Ethernet Integradas

As interfaces são compostas por um conector tipo RJ45 de comunicação no padrão 10/100Base-TX. Permite a comunicação ponto a ponto ou em rede nos protocolos abertos, como por exemplo, MODBUS TCP Cliente, MODBUS RTU via TCP Cliente, MODBUS TCP Servidor e MODBUS RTU via TCP Servidor.

Abaixo, segue os parâmetros que devem ser configurados para o bom funcionamento da aplicação.

5.4.1.1. NET 1

Configuração	Descrição	Padrão	Opções
Obter um endereço IP automaticamente	Habilita a funcionalidade do Cliente DHCP no dispositivo para atribuição automática de IP	Desmarcado	Marcado ou Desmarcado
Endereço IP	Endereço IP do controlador no barramento Ethernet	192.168.15.1	1.0.0.1 a 223.255.255.254
Máscara de Subrede	Máscara de subrede do controlador no barramento Ethernet	255.255.255.0	128.0.0.0 a 255.255.255.252
Endereço do Gateway	Endereço do Gateway do controlador no barramento Ethernet	192.168.15.253	0.0.0.0 a 223.255.255.254

Tabela 63: Configurações NET 1

5.4.1.2. NET 2

Configuração	Descrição	Padrão	Opções
Endereço IP	Endereço IP do controlador no barramento Ethernet	192.168.16.1	1.0.0.1 a 223.255.255.254
Máscara de Subrede	Máscara de subrede do controlador no barramento Ethernet	255.255.255.0	128.0.0.0 a 255.255.255.252
Endereço do Gateway	Endereço do Gateway do controlador no barramento Ethernet	192.168.16.253	0.0.0.0 a 223.255.255.254

Tabela 64: Configurações NET 2

ATENÇÃO

Não é possível configurar mais de uma interface Ethernet de uma UCP na mesma subrede, sendo este tipo de configuração bloqueada pela ferramenta MasterTool. Desta forma, cada interface Ethernet deve ser configurada em uma subrede distinta.

5.4.2. Interface Ethernet Remota NX5000

5.4.2.1. NET 1

A interface é composta por um conector tipo RJ45 de comunicação no padrão 10/100Base-TX. Permite a comunicação ponto a ponto ou em rede nos protocolos abertos, como por exemplo, MODBUS TCP Cliente, MODBUS RTU via TCP Cliente, MODBUS TCP Servidor e MODBUS RTU via TCP Servidor.

Abaixo, segue os parâmetros que devem ser configurados para o bom funcionamento da aplicação.

Configuração	Descrição	Padrão	Opções
Endereço IP	Endereço IP do controlador no barramento Ethernet	192.168.xx.68	1.0.0.1 a 223.255.255.254
Máscara de Subrede	Máscara de subrede do controlador no barramento Ethernet	255.255.255.0	128.0.0.0 a 255.255.255.252
Endereço do Gateway	Endereço do Gateway do controlador no barramento Ethernet	192.168.xx.253	0.0.0.0 a 223.255.255.254

Tabela 65: Configurações NET 1 Remota NX5000

5.4.2.2. Modos de Operação da Interface Ethernet Remota NX5000

Os módulos NX5000 podem ser inseridos no projeto para aumentar a quantidade de interfaces Ethernet caso as interfaces locais da UCP não sejam suficientes.

Os canais Ethernet dos módulos NX5000 podem ser usados de forma individual, ou organizados em pares redundantes.

5.4.2.2.1. Modo Redundante

Um conjunto de duas portas Ethernet formando um par redundante possui um único endereço de IP, vinculado ao par de portas. Desta forma, um cliente como um SCADA ou MasterTool, conectado a um servidor no CP, não precisa se preocupar em trocar o endereço IP caso haja falha em algumas das portas do par redundante.

Para agrupar dois módulos NX5000 como um par redundante, estes dois módulos devem necessariamente ocupar posições vizinhas no bastidor, e o checkbox *Redundante* do módulo à esquerda deve ser marcado. Ao fazer isto, o módulo à direita tem a edição de seus parâmetros bloqueada. Os parâmetros editados no módulo à esquerda passam a ser comuns para os dois módulos.

Por outro lado, desmarcando o checkbox *Redundante* do módulo à esquerda, ocorre a separação dos módulos, que voltam a se comportar como módulos individuais sem redundância.

Quando o Modo *Redundante* for selecionado, outros parâmetros serão habilitados automaticamente na mesma tela e deverão ser configurados:

- **Período de Teste da Redundância (ms):** Período de envio do frame de teste de comunicação entre as duas NETs. Pode ser configurado com valores entre 100 e 9900, padrão 500
- **Número de Retentativas do Teste da Redundância:** Número máximo de vezes que a NET que enviou o frame irá esperar pela resposta. Pode ser configurado com valores entre 1 e 100, padrão 4
- **Período para Chaveamento (s):** Tempo máximo que a NET Ativa irá esperar por um pacote qualquer. Pode ser configurado com valores entre 1 e 25, padrão 10

Caso o tempo de resposta do *Teste da Redundância* atinja *Período de Teste* vezes o *Número de Retentativas* e a interface ativa permaneça por um tempo maior que o *Período para Chaveamento* sem receber nenhum pacote, ocorrerá um switchover, tornando ativa a interface que até então estava inativa. É importante ressaltar que há um delay entre a detecção da falha e ativação da interface inativa devido ao tempo necessário para sua configuração. Tal delay pode chegar a algumas dezenas de milissegundos

Quando uma das NETs estiver ativa, esta assumirá o endereço de IP configurado, permanecendo a NET inativa com seus parâmetros de Endereço de IP, Máscara de Subrede e Endereço do Gateway em branco nos diagnósticos da UCP.

Figura 51: Advanced Configuration of Remote Ethernet Interface - NX5000

5.4.3. Portas TCP/UDP Reservadas

As seguintes portas TCP/UDP das interfaces Ethernet, tanto locais quanto remotas, são utilizadas por serviços da UCP (dependem da disponibilidade conforme tabela [Protocolos](#)) e, portanto, são reservadas e não devem ser utilizadas pelo usuário.

Serviço	TCP	UDP
Página Web de Sistema	80	-
SNTP	-	123
SNMP	-	161
MODBUS TCP	502*	-
MasterTool MT8500	1217*	1740:1743
SQL Server	1433	-
MQTT	1883* / 8883*	-
EtherNet/IP	44818	2222
IEC 60870-5-104	2404*	-
OPC UA	4840	-
WEBVISU	8080	-
CODESYS ARTI	11740	-
PROFINET	-	34964
Portainer Docker	9000	-

Tabela 66: Portas TCP/UDP reservadas

* Porta padrão, mas que pode ser alterada pelo usuário.

5.5. Configuração de Protocolos

Independente dos protocolos utilizados em cada aplicação, as UCPs da Série Nexto possuem alguns limites máximos de utilização para cada modelo de UCP. Existem basicamente dois diferentes tipos de protocolos de comunicação, os que utilizam mapeamentos simbólicos e os que utilizam mapeamentos por representação direta. O limite máximo de mapeamentos assim como a quantidade máxima de protocolos (instâncias) estão definidos na tabela abaixo:

	NX3030
Pontos Mapeados	20480
Mapeamentos (Por Instância / Total)	5120 / 20480
NETs – Instâncias Cliente ou Servidor (Por NET / Total)	4 / 16
COM (n) – Instâncias Mestre ou Escravo	1
Requisições	512
Centros de Controle	3

Tabela 67: Limites dos Protocolos por UCP

Notas:

Pontos Mapeados: É o número máximo de pontos mapeados que a UCP suporta. Lembrando que cada mapeamento suporta 1 ou mais pontos mapeados, dependendo do tamanho do dado, quando utilizado com variáveis do tipo ARRAY.

Mapeamentos: Um “mapeamento” é uma relação entre uma variável interna de aplicação e um objeto do protocolo de aplicação. Este campo informa a quantidade máxima de mapeamentos suportada pela UCP. Corresponde a soma de todos os mapeamentos realizados nas instâncias de protocolos de comunicação e seus respectivos dispositivos.

Requisições: A soma das requisições dos protocolos de comunicação, declaradas nos dispositivos, não pode ultrapassar a quantidade máxima de requisições suportadas pela UCP.

NETs: Instâncias Cliente ou Servidor: Este campo define o número máximo de instâncias de protocolos para cada interface Ethernet, assim como o número máximo total distribuído entre todas as interfaces Ethernet do sistema.

COM (n): Instâncias Mestre ou Escravo: Devido as suas características, cada interface serial suporta apenas uma instância de protocolo de comunicação. São exemplos de instâncias compatíveis com interfaces Seriais: MODBUS RTU Mestre e MODBUS RTU Escravo.

Centros de Controle: “Centro de Controle” é todo dispositivo cliente conectado à UCP através do protocolo IEC 60870-5-104. Este campo informa a quantidade máxima de dispositivos clientes, do tipo centro de controle, suportada pela UCP. Corresponde a soma de todos os dispositivos clientes das instâncias de protocolos de comunicação Servidor IEC 60870-5-104 (não inclui os mestres e clientes dos protocolos MODBUS RTU Escravo, MODBUS Servidor e DNP3 Servidor).

As limitações do protocolo MODBUS por Representação Direta e por mapeamentos simbólicos para as UCPs podem ser visualizadas na Tabela 68 e na Tabela 69, respectivamente.

Limitações	MODBUS RTU Mestre	MODBUS RTU Escravo	MODBUS Ethernet Cliente	MODBUS Ethernet Servidor
Mapeamentos por instância	128	32	128	32
Dispositivos por instância	64	1 ⁽¹⁾	64	64 ⁽²⁾
Mapeamentos por dispositivo	32	32	32	32
Requisições simultâneas por instância	-	-	128	64
Requisições simultâneas por dispositivo	-	-	8	64

Tabela 68: Limites do Protocolo MODBUS por Representação Direta

Notas:**Dispositivos por instância:**

- Protocolos Mestre ou Cliente: quantidade de dispositivos escravos ou servidores suportados por cada instância de protocolo Mestre ou Cliente.
- Protocolo MODBUS RTU Escravo: o limite ⁽¹⁾ informado diz respeito às interfaces seriais, que não permitem que um Escravo possa estabelecer comunicação, simultaneamente, através da mesma interface serial, com mais de um dispositivo Mestre. Não é necessário, e nem é possível, declarar nem configurar o dispositivo Mestre sob a instância do protocolo MODBUS RTU Escravo. O dispositivo Mestre terá acesso a todos os mapeamentos feitos diretamente na instância do protocolo MODBUS RTU Escravo.

- Protocolo MODBUS Servidor: o limite ⁽²⁾ informado diz respeito às interfaces Ethernet, que limitam a quantidade de conexões que podem ser estabelecidas com outros dispositivos através de uma mesma interface Ethernet. Não é necessário, e nem é possível, declarar nem configurar os dispositivos Clientes sob a instância do protocolo MODBUS Servidor. Todos os dispositivos Clientes terão acesso a todos os mapeamentos feitos diretamente na instância do protocolo MODBUS Servidor.

Mapeamentos por dispositivo: O número máximo de mapeamentos por dispositivo apesar de estar relacionado acima, também é limitado pelo número máximo de mapeamentos do protocolo. Também deve ser considerado o limite máximo de mapeamentos da UCP conforme na Tabela 67.

Requisições simultâneas por instância: Quantidade de requisições que podem ser transmitidas simultaneamente por cada instância de protocolo Cliente, ou que podem ser recebidas simultaneamente por cada instância de protocolo Servidor. Instâncias de protocolo MODBUS RTU, Mestre ou Escravo, não suportam requisições simultâneas.

Requisições simultâneas por dispositivo: Quantidade de requisições que podem ser transmitidas simultaneamente para cada dispositivo MODBUS Servidor, ou que podem ser recebidas simultaneamente de cada dispositivo MODBUS Cliente. Dispositivos MODBUS RTU, Mestre ou Escravo não suportam requisições simultâneas.

Limitações	MODBUS RTU Mestre	MODBUS RTU Escravo	MODBUS Ethernet Cliente	MODBUS Ethernet Servidor
Dispositivos por instância	64	1 ⁽¹⁾	64	64 ⁽²⁾
Requisições por dispositivo	32	-	32	-
Requisições simultâneas por instância	-	-	128	64
Requisições simultâneas por dispositivo	-	-	8	64

Tabela 69: Limites do Protocolo MODBUS por Mapeamentos Simbólicos

Notas:

Dispositivos por instância:

- Protocolos Mestre ou Cliente: quantidade de dispositivos escravos ou servidores suportados por cada instância de protocolo Mestre ou Cliente.
- Protocolo MODBUS RTU Escravo: o limite ⁽¹⁾ informado diz respeito às interfaces seriais, que não permitem que um Escravo possa estabelecer comunicação, simultaneamente, através da mesma interface serial, com mais de um dispositivo Mestre. Não é necessário, e nem é possível, declarar nem configurar o dispositivo Mestre sob a instância do protocolo MODBUS RTU Escravo. O dispositivo Mestre terá acesso a todos os mapeamentos feitos diretamente na instância do protocolo MODBUS RTU Escravo.
- Protocolo MODBUS Servidor: o limite ⁽²⁾ informado diz respeito às interfaces Ethernet, que limitam a quantidade de conexões que podem ser estabelecidas com outros dispositivos através de uma mesma interface Ethernet. Não é necessário, e nem é possível, declarar nem configurar os dispositivos Clientes sob a instância do protocolo MODBUS Servidor. Todos os dispositivos Clientes terão acesso a todos os mapeamentos feitos diretamente na instância do protocolo MODBUS Servidor.

Requisições por dispositivo: Quantidade de requisições, como por exemplo de leitura ou escrita de holding registers, que podem ser configuradas para cada um dos dispositivos (escravos ou servidores) de instâncias de protocolos Mestre ou Cliente. Este parâmetro não é aplicável às instâncias de protocolos Escravo ou Servidor.

Requisições simultâneas por instância: Quantidade de requisições que podem ser transmitidas simultaneamente por cada instância de protocolo Cliente, ou que podem ser recebidas simultaneamente por cada instância de protocolo Servidor. Instâncias de protocolo MODBUS RTU, Mestre ou Escravo, não suportam requisições simultâneas.

Requisições simultâneas por dispositivo: Quantidade de requisições que podem ser transmitidas simultaneamente para cada dispositivo MODBUS Servidor, ou que podem ser recebidas simultaneamente de cada dispositivo MODBUS Cliente. Dispositivos MODBUS RTU, Mestre ou Escravo não suportam requisições simultâneas.

ATENÇÃO

Não são suportadas requisições simultâneas para uma mesma variável associada à pontos de comunicação, os quais suportem o modo de operação SBO (Select Before Operate), mesmo sendo recebidas por dispositivos diferentes. Uma vez iniciada a seleção e/ou operação de um ponto por um dado dispositivo, esta deve ser finalizada antes que este ponto possa ser comandado por outro dispositivo.

As limitações do protocolo IEC 60870-5-104 Servidor podem ser visualizadas na tabela abaixo.

Limitações	IEC 60870-5-104 Servidor
Dispositivos por instância	3
Requisições simultâneas por instância	3
Requisições simultâneas por dispositivo	1

Tabela 70: Limites do Protocolo IEC 60870-5-104 Servidor

Notas:

Dispositivos por instância: Quantidade de dispositivos clientes, do tipo centro de controle, suportados por cada instância de protocolo Servidor IEC 60870-5-104. O limite informado pode ser menor em função dos limites totais da UCP (consultar a Tabela 67).

Requisições simultâneas por instância: Quantidade de requisições que podem ser recebidas simultaneamente por cada instância de protocolo Servidor.

Requisições simultâneas por dispositivo: Quantidade de requisições que podem ser recebidas simultaneamente de cada dispositivo IEC 60870-5-104 Cliente.

5.5.1. Comportamento dos Protocolos x Estados da UCP

A tabela abaixo mostra detalhadamente cada comportamento assumido para cada protocolo de comunicação configurável nas UCPs da série Nexto, em todos seus estados de funcionamento.

Protocolo	Tipo	Estado operacional da UCP					
		STOP			RUN		
		Depois do download antes da aplicação iniciar	Depois de a aplicação ir para STOP (PAUSE)	Depois de uma exceção	Não redundante ou Ativo	Redundante em Reserva	Após um break-point na MainPrg
MODBUS Symbol	Slave/Server	✓	✓	✓	✓	✓	✓
	Master/Client	✗	✗	✗	✓	✓	✓
MODBUS	Slave/Server	✗	✗	✗	✓	✓	✗
	Master/Client	✗	✗	✗	✓	✓	✓
SOE (DNP3)	Outstation	✓	✓	✓	✓	✗	✓
IEC 60870-5-104	Server	✗	✗	✗	✓	✗	✓
EtherCAT	Master	✓	✓	✗	✓	NA	✓
OPC DA	Server	✓	✓	✓	✓	✗	✓
OPC UA	Server	✓	✓	✓	✓	✓	✓
SNTP	Client	✓	✓	✓	✓	✓	✓
HTTP	Server	✓	✓	✓	✓	✓	✓
SNMP	Agent	✓	✓	✓	✓	✓	✓
EtherNet/IP	Scanner	✓	✓	✗	✓	NA	✗
	Adapter	✗	✓	✗	✓	NA	✗

Tabela 71: Comportamento dos Protocolos x Estados da UCP

Notas:

Símbolo ✓: O protocolo permanece ativo e operando normalmente.

Símbolo ✗: O protocolo está desabilitado.

MODBUS Symbol Slave/Server: Para que o protocolo comunique nas condições em que a UCP não está em RUN ou após um breakpoint é necessário que seja marcada a opção “*Mantém a comunicação funcionando quando a UCP está parada*”.

5.5.2. Pontos Duplos

A representação de pontos digitais duplos de entrada e saída é realizada através de um tipo de dados especial chamado DBP (definido na biblioteca *LibDataTypes*). Este tipo consiste basicamente em uma estrutura de dois elementos do tipo BOOL chamados OFF e ON (equivalentes a TRIP e CLOSE respectivamente).

No Nexto, variáveis deste tipo não podem ser associadas à módulos digitais de entrada e saída, sendo necessário o mapeamento de pontos digitais individuais, tipo BOOL, e o tratamento via aplicação para conversão em pontos duplos.

Maiores informações sobre o mapeamento de pontos duplos nos módulos de entrada e saída digital podem ser obtidas na seção [IEC 60870-5-104 Servidor](#).

5.5.3. Fila de Eventos da UCP

A UCP possui uma fila de eventos do tipo FIFO (First In, First Out) utilizada para armazenar temporariamente os eventos relacionados aos pontos de comunicação até que eles sejam transferidos ao seu destino final.

Todos os eventos de pontos de comunicação gerados no CP são direcionados e armazenados na fila da UCP. Esta fila possui as seguintes características:

- Tamanho: 1000 eventos
- Retentividade: não é retentiva
- Política de estouro: mantém os mais recentes

ATENÇÃO

No CP Nexto, a fila de eventos é armazenada em uma área de memória não retentiva (volátil). Desta forma, os eventos presentes na fila da UCP, que ainda não foram transmitidos ao centro de controle, serão perdidos em um eventual desligamento do CP.

A fila de eventos da UCP é redundante, isto é, é sincronizada ciclo a ciclo entre as duas UCPs quando se utiliza redundância de UCP. Maiores informações podem ser encontradas na seção específica sobre redundância de UCP.

A entrada e saída de eventos nesta fila seguem um conceito de produtor/consumidor. Produtores são aqueles elementos do sistema capazes de gerar eventos, adicionando eventos na fila da UCP, enquanto os consumidores são os elementos do sistema que recebem e utilizam estes eventos, retirando eventos da fila da UCP. A figura abaixo descreve este funcionamento, incluindo exemplo de alguns consumidores e produtores de eventos:

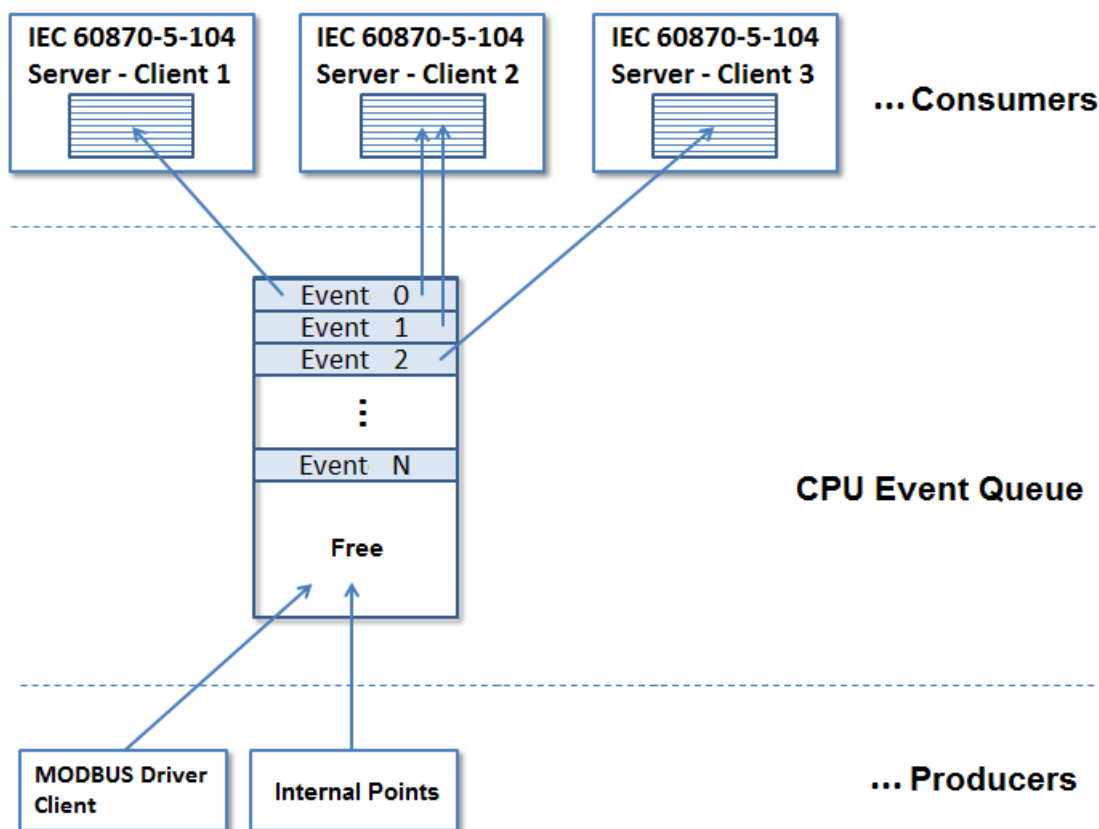


Figura 52: Fila de Eventos da UCP

5.5.3.1. Consumidores

Os consumidores são tipicamente drivers de comunicação que irão se comunicar com um SCADA ou IHM. Após serem armazenados na fila da UCP, os consumidores recebem os eventos relativos a pontos de comunicação mapeados na sua configuração. Estes eventos são então armazenados em uma fila de eventos própria do consumidor, cujo tamanho e funcionamento é descrito na seção específica do driver de comunicação.

5.5.3.2. Princípios de Funcionamento da Fila

Uma vez armazenado na fila da UCP, cada evento é transmitido para o consumidor que possui este ponto de comunicação na sua base de dados. Na figura acima, o Evento 0 é referente a um ponto de comunicação mapeado para 2 centros de controle IEC 60870-5-104 (Cliente 1 e 2). Já o Evento 1 é referente a um ponto de comunicação mapeado apenas para um centro de controle IEC 60870-5-104 (Cliente 2). Por sua vez, o Evento 2 é referente a um ponto de comunicação mapeado para um outro centro de controle IEC 60870-5-104 (Cliente 3).

Os eventos permanecem armazenados na fila da UCP até que todos os seus consumidores confirmem a sua recepção. O critério utilizado para confirmar a recepção é específico de cada consumidor. No caso do IEC 60870-5-104 Servidor, a confirmação ocorre quando o evento foi transmitido para o cliente IEC 60870-5-104.

No caso da Série Nexto, não são disponibilizados diagnósticos para monitorar a ocupação da fila de eventos da UCP, nem mesmo informações sobre estouro da fila. Já os consumidores disponibilizam um conjunto de diagnósticos referente a sua fila de eventos. Maiores informações podem ser encontradas na seção específica do driver de comunicação.

5.5.3.2.1. Sinalização de Estouro

A sinalização de estouro para a fila de eventos dos consumidores ocorre em duas situações:

- Quando a fila de eventos do consumidor não tiver mais espaço para armazenar novos eventos
- Se a UCP abortou a geração de eventos (porque ocorreram mais eventos em um único ciclo de execução do que o tamanho total da sua fila de eventos)

5.5.3.3. Produtores

Os produtores são tipicamente drivers de comunicação ou elementos internos do CP que são capazes de gerar eventos. A figura anterior mostra alguns exemplos:

- **Pontos Internos:** este é um elemento interno no firmware do CP, o qual realiza a detecção de eventos a cada ciclo de execução (MainTask) para aqueles pontos de comunicação que não possuem uma origem definida e então insere os eventos na fila da UCP. O número máximo de eventos que podem ser detectados a cada ciclo da MainTask é igual ao tamanho da fila de eventos da UCP. Caso seja gerado um número de eventos maior que este em um mesmo ciclo da MainTask, os eventos excedentes serão perdidos
- **Driver MODBUS (Cliente/Servidor/Mestre/Escravo):** as alterações de valor em variáveis causadas por leituras/escritas MODBUS são detectadas a cada ciclo da MainTask e então os eventos são inseridos na fila da UCP. No caso dos drivers Cliente/Mestre, são gerados também eventos de qualidade quando houver falha de comunicação com o dispositivo escravo

5.5.4. Intercepção de Comandos Oriundos do Centro de Controle

O CP Nexto possui um bloco de função que permite que comandos de seleção e operação para pontos de saída recebidos por drivers servidores (IEC 60870-5-104 Servidor) sejam tratados pela lógica de usuário. Este recurso permite a implementação de intertravamentos, bem como a manipulação dos dados do comando recebido na lógica do usuário ou ainda o redirecionamento do comando para diferentes IEDs.

A intercepção de comandos é implementada pelo bloco de função *CommandReceiver*, definido na biblioteca *LibRtuStandard*. Os parâmetros de entrada e saída são descritos nas tabelas a seguir:

Parâmetro	Tipo	Descrição
bExec	BOOL	Quando TRUE, executa a interceptação do comando
bDone	BOOL	Indica que os dados de saída do comando já foram processados, liberado o bloco de função para recepção de um novo comando
dwVariableAddr	DWORD	Endereço da variável, mapeado no driver servidor, que receberá o comando do cliente
eCommandResult	ENUM	Ação de entrada definida pelo usuário a partir da seguinte lista: SUCCESS(0) NOT_SUPPORTED(1) BLOCKED_BY_SWITCHING_HIERARCHY(2) SELECT_FAILED(3) INVALID_POSITION(4) POSITION_REACHED(5) PARAMETER_CHANGE_IN_EXECUTION(6) STEP_LIMIT(7) BLOCKED_BY_MODE(8) BLOCKED_BY_PROCESS(9) BLOCKED_BY_INTERLOCKING(10) BLOCKED_BY_SYNCHROCHECK(11) COMMAND_ALREADY_IN_EXECUTION(12) BLOCKED_BY_HEALTH (13) ONE_OF_N_CONTROL(14) ABORTION_BY_CANCEL(15) TIME_LIMIT_OVER(16) ABORTION_BY_TRIP(17) OBJECT_NOT_SELECTED(18) OBJECT_ALREADY_SELECTED(19) NO_ACCESS_AUTHORITY(20) ENDED_WITH_OVERSHOOT(21) ABORTION_DUE_TO_DEVIATION(22) ABORTION_BY_COMMUNICATION_LOSS(23) BLOCKED_BY_COMAND(24) NONE(25) INCONSISTENT_PARAMETERS(26) LOCKED_BY_OTHER_CLIENT(27) HARDWARE_ERROR(28) UNKNOWN(29)
dwTimeout	DWORD	Time-out [ms] para o tratamento pela lógica do usuário

Tabela 72: Parâmetros de Entrada do Bloco de Função CommandReceiver

Notas:

bExec: quando FALSE, o comando apenas deixa de ser interceptado para a aplicação do usuário, mas o comando continua sendo tratado normalmente pelo servidor.

bDone: após a interceptação do comando, o usuário será responsável por tratá-lo. Ao fim do tratamento, esta entrada deve ser habilitada para que um novo comando possa ser recebido. Caso esta entrada não seja habilitada, o bloco aguardará o tempo definido em *dwTimeout*, para então ficar apto a interceptar novos comandos.

eCommandResult: resultado do tratamento do comando interceptado pelo usuário. O resultado retornado ao cliente que enviou o comando, o qual deve ser atribuído juntamente com a entrada *bDone*, é convertido para o formato do protocolo do qual foi recebido o comando. Na série Nexto somente é suportada a interceptação de comandos oriundos do protocolo

IEC 60870-5-104. Na interceptação do protocolo IEC 60870-5-104, qualquer retorno diferente de *SUCCESS* resulta em uma confirmação negativa no protocolo.

ATENÇÃO

Não é recomendada a interceptação simultânea de comandos para uma mesma variável por dois ou mais blocos de Função *CommandReceiver*. Apenas um dos blocos de função irá interceptar corretamente o comando, podendo no entanto sofrer interferências indesejadas dos demais blocos de função se endereçados à mesma variável.

Parâmetro	Tipo	Descrição
bCommandAvailable	BOOL	Indica que foi interceptado um comando, e que os dados estão disponíveis para serem processados
sCommand	STRUCT	Esta estrutura armazena os dados do comando recebido, a qual é composta pelo seguintes campos: eCommand sSelectParameters sOperateParameters A descrição de cada campo segue nesta seção.
eStatus	ENUM (TYPE_RESULT)	Ação de saída da função a partir do resultado obtido, conforme lista: OK_SUCCESS(0) ERROR_FAILED(1)

Tabela 73: Parâmetros de saída do Bloco de Função *CommandReceiver***Nota:**

eStatus: retorno do processo de registro da interceptação de um comando para um ponto de comunicação. Quando a interceptação é registrada com sucesso é retornado *OK_SUCCESS*, caso contrário *ERROR_FAILED*. Em caso de falha no registro do interceptador, comandos para o determinado ponto não são interceptados pelo respectivo bloco de função. *TYPE_RESULT* está definido na biblioteca *LibDataTypes*.

Os comandos suportados são descritos na tabela abaixo:

Parâmetro	Tipo	Descrição
eCommand	ENUM	NO_COMMAND(0) SELECT(1) OPERATE(2)

Tabela 74: Comandos Suportados do Bloco de Função *CommandReceiver*

Os parâmetros que compõem as estruturas *sSelectParameters*, *sOperateParameters* e *sCancelParameters* são descritos nas tabelas a seguir:

Parâmetro	Tipo	Descrição
sSelectConfig	STRUCT	Configuração do comando de seleção recebido. Os parâmetros desta estrutura estão descritos na Tabela 76
sValue	STRUCT	Valor recebido em um select, quando recebido um comando de seleção com valor. Os parâmetros desta estrutura estão descritos na Tabela 79

Tabela 75: Parâmetros *sSelectParameters*

Parâmetro	Tipo	Descrição
bSelectWith Value	BOOL	Quando true indica a recepção de um comando de seleção com valor.

Tabela 76: Parâmetros sSelectConfig

Parâmetro	Tipo	Descrição
sOperateConfig	STRUCT	Configuração do comando de seleção recebido. Os parâmetros desta estrutura estão descritos na Tabela 78
sValue	STRUCT	Campo de valor referente ao comando de operação recebido. Os parâmetros desta estrutura estão descritos na Tabela 79

Tabela 77: Parâmetros sOperateParameters

Parâmetro	Tipo	Descrição
bDirectOperate	BOOL	Quando true indica que foi recebido um comando de operação sem select.
bNoAcknowledgement	BOOL	Quando true indica que foi recebido um comando que não requer a confirmação de recebimento.
bTimedOperate	BOOL	Quando true indica que foi recebido um comando de operação ativado por tempo.
liOperateTime	LINT	Programação do instante em que deve ser executado o comando. Este campo é válido somente quando <i>bTimedOperate</i> for true.
bTest	BOOL	Quando true indica que o comando recebido foi enviado somente para fins de teste, desta forma o mesmo não deve ser executado.

Tabela 78: Parâmetros sOperateConfig

Parâmetro	Tipo	Descrição
eParamType	ENUM	Informa o tipo de comando que foi recebido: NO_COMMAND(0) SINGLE_POINT_COMMAND(1) DOUBLE_POINT_COMMAND(2) INTEGER_STATUS_COMMAND(3) ENUMERATED_STATUS_COMMAND(4) ANALOGUE_VALUE_COMMAND(5)
sSinglePoint	STRUCT	Quando recebido um comando, em função do tipo do comando recebido, definido por eParamType, é preenchida a estrutura de dados correspondente. Os parâmetros destas estruturas são apresentados na Tabela 80 a Tabela 84
sDoublePoint	STRUCT	
sIntegerStatus	STRUCT	
sEnumeratedStatus	STRUCT	
sAnalogueValue	STRUCT	

Tabela 79: Parâmetros sValue

Parâmetro	Tipo	Descrição
bValue	BOOL	Valor de operação do ponto.
sPulseConfig	STRUCT	Os parâmetros de configuração de um comando pulsado são armazenados nesta estrutura. Os parâmetros desta estrutura são apresentados na Tabela 85.

Tabela 80: Parâmetros sSinglePoint

Parâmetro	Tipo	Descrição
bValue	BOOL	Valor de operação do ponto.
sPulseConfig	STRUCT	Os parâmetros de configuração de um comando pulsado são armazenados nesta estrutura. Os parâmetros desta estrutura são apresentados na Tabela 85.

Tabela 81: Parâmetros sDoublePoint

Parâmetro	Tipo	Descrição
diValue	DINT	Valor de operação do ponto.

Tabela 82: Parâmetros sIntegerStatus

Parâmetro	Tipo	Descrição
dwValue	DWORD	Valor de operação do ponto.

Tabela 83: Parâmetros sEnumeratedStatus

Parâmetro	Tipo	Descrição
eType	ENUM	Informa o tipo de dado do valor analógico recebido: INTEGER (0) FLOAT (1)
diValue	DINT	Valor de operação do ponto, no formato inteiro.
fValue	REAL	Valor de operação do ponto, no formato float.

Tabela 84: Parâmetros sAnalogueValue

Parâmetro	Tipo	Descrição
bPulseCommand	BOOL	Quando true indica que o comando recebido é pulsado.
dwOnDuration	DWORD	Esta é a duração, expressa em milissegundos, que a saída digital deve permanecer ligada.
dwOffDuration	DWORD	Esta é a duração, expressa em milissegundos, que a saída digital deve permanecer desligada.
dwPulseCount	DWORD	Número de vezes que o comando deve ser executado.

Tabela 85: Parâmetros sPulseConfig

Para efetuar a interceptação de comandos para um dado ponto, primeiramente deve-se carregar no parâmetro *dwVariableAddr* o endereço da variável correspondente ao ponto que busca-se interceptar comandos e então executar um pulso no

parâmetro *bExec*. Uma vez recebido o comando, o bloco funcional informa que um comando foi interceptado através do parâmetro *bCommandAvailable*. As informações do comando interceptado são então preenchidas nos parâmetros de saída *sCommand* e *eStatus* conforme o tipo de comando que foi recebido. Esta operação depende somente do tipo de comando recebido, não importando o tipo de dado da variável para qual está se interceptando o comando. A interceptação é finalizada e então o bloco de função pode ser liberado para interceptar um novo comando quando sinalizado *true* no parâmetro *bDone*. Ainda deve ser indicado o resultado do processamento do comando em *eCommandResult*.

5.5.5. MODBUS RTU Mestre

Este protocolo está disponível para as UCPs da Série Nexto nos seus canais seriais. Ao selecionar esta opção no MasterTool IEC XE, a UCP passa a ser mestre da comunicação MODBUS, possibilitando o acesso a outros dispositivos com o mesmo protocolo, quando esta estiver em modo de execução (Modo *Run*).

Há dois modos de configuração para este protocolo. Um deles faz uso de Representação Direta (%Q), no qual as variáveis são definidas pelo seu endereço. O outro, chamado Mapeamento Simbólico, tem as variáveis definidas pelo seu nome.

Independente do modo de configuração, os passos para inserir uma instância do protocolo e configurar a interface serial são iguais. O procedimento para inserir uma instância de protocolo é encontrado com detalhes no Manual de Utilização do MasterTool IEC XE – MU299048 ou na seção [Inserindo uma Instância de Protocolo](#). As demais etapas de configuração serão descritas a seguir para cada modo.

- Adicionar a instância do protocolo MODBUS RTU Mestre ao canal serial COM 1 ou COM 2 (ou ambos, em caso de duas redes de comunicação). Para realizar esse procedimento, consultar a seção [Inserindo uma Instância de Protocolo](#).
- Configurar a interface serial, escolhendo a taxa de transmissão, o comportamento dos sinais de modem RTS/CTS, a paridade, bits de parada do canal e demais configurações através de um duplo clique sobre o canal serial COM 1 ou COM 2. Consultar a seção [Configuração das Interfaces Seriais](#).

5.5.5.1. Configuração do Protocolo MODBUS Mestre por Mapeamento Simbólico

Para configurar este protocolo usando Mapeamento Simbólico, é necessário executar os seguintes passos:

- Configurar os parâmetros gerais do protocolo MODBUS Mestre, como: tempos de atraso de envio e interframe mínimo como na Figura 53.
- Adicionar e configurar dispositivos através da aba Parâmetros Gerais, definindo endereço do escravo, time-out de comunicação e número de tentativas de comunicação como pode ser visto na Figura 54.
- Adicionar e configurar os mapeamentos MODBUS na aba Mapeamentos conforme Figura 55, especificando o nome da variável, tipo de dados e endereço inicial do dado, o tamanho do dado e a faixa são preenchidos automaticamente.
- Adicionar e configurar as requisições MODBUS como apresentado na Figura 56, especificando a função, o tempo de varredura da requisição, o endereço inicial (leitura/escrita), o tamanho dos dados (Leitura/Escrita) e gerar as variáveis de diagnóstico e desabilitação das requisições através dos botões na parte inferior da janela.

5.5.5.1.1. Parâmetros Gerais do Protocolo MODBUS Mestre – Configuração por Mapeamento Simbólico

Os parâmetros gerais, encontrados na tela inicial de configuração do protocolo MODBUS (figura abaixo), são definidos como:

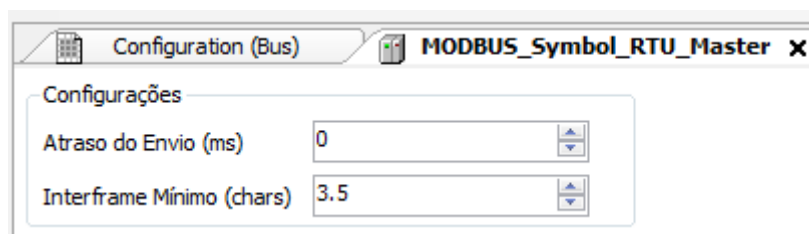


Figura 53: Tela de Configuração Parâmetros Gerais MODBUS RTU Mestre

Configuração	Descrição	Padrão	Opções
Atraso do Envio (ms)	Tempo de atraso para envio da resposta.	0	0 a 65535
Interframe Mínimo (chars)	Tempo mínimo de silêncio entre diferentes frames.	3.5	3.5 a 100.0

Tabela 86: Configurações Gerais MODBUS RTU Mestre

Notas:

Atraso do envio: A resposta à uma requisição MODBUS pode causar problemas em certos momentos, como, por exemplo, na interface RS-485 ou outra half-duplex. Às vezes existe um atraso entre o tempo de resposta do escravo e o silêncio na linha física (atraso no escravo para colocar RTS em zero e colocar o transmissor RS-485 em alta impedância). Para resolver o problema, o mestre pode esperar o tempo determinado nesse campo antes de enviar a nova requisição. Caso contrário, os primeiros bytes transmitidos pelo mestre, durante a requisição, podem ser perdidos.

Interframe mínimo: A norma MODBUS define esse tempo como 3,5 caracteres, porém esse parâmetro é configurável para atender aos dispositivos que não estão de acordo com o padrão.

Os diagnósticos e comandos do protocolo MODBUS Mestre configurado, seja por mapeamento simbólico ou por representação direta, são armazenados em variáveis do tipo *T_DIAG_MODBUS_RTU_MASTER_1* e ainda para o mapeamento por representação direta estão em 4 bytes e 8 words, os quais estão descritos na tabela abaixo (n é o valor configurado no campo *Endereço Inicial de Diagnósticos em %Q*):

Variável de Representação Direta	Variável de diagnóstico do tipo T_DIAG_MODBUS_RTU_MASTER_1.*	Tamanho	Descrição
Bits de diagnóstico:			
%QX(n).0	tDiag. bRunning	BIT	O mestre está em execução.
%QX(n).1	tDiag. bNotRunning	BIT	O mestre não está em execução (ver bit: bInterruptedByCommand).
%QX(n).2	tDiag. bInterruptedByCommand	BIT	O bit bNotRunning foi habilitado, pois o mestre foi interrompido pelo usuário através de bits de comando.
%QX(n).3	tDiag. bConfigFailure	BIT	Diagnóstico descontinuado.
%QX(n).4	tDiag. bRXFailure	BIT	Diagnóstico descontinuado.
%QX(n).5	tDiag. bTXFailure	BIT	Diagnóstico descontinuado.
%QX(n).6	tDiag. bModuleFailure	BIT	Indica se há falha no módulo ou o módulo não está presente.
%QX(n).7	tDiag. bDiag_7_reserved	BIT	Reservado
Códigos de erro:			
			0: não existem erros 1: porta serial inválida 2: modo da porta serial inválido 3: taxa de transmissão inválida 4: bits de dado inválidos 5: bits de paridade inválidos

Variável de Representação Direta	Variável de diagnóstico do tipo T_DIAG_MODBUS_RTU_MASTER_1.*	Tamanho	Descrição
%QB(n+1)	eErrorCode	SERIAL_STATUS (BYTE)	6: bits de parada inválidos 7: parâmetro de sinal de modem inválido 8: parâmetro de Threshold de RX da UART inválido 9: parâmetro de time-out inválido 10: porta serial ocupada 11: erro de hardware na UART 12: erro de hardware remoto 20: tamanho do buffer de transmissão inválido 21: método de sinal de modem inválido 22: time-out de CTS = verdadeiro 23: time-out de CTS = falso 24: erro de time-out na transmissão 30: tamanho do buffer de recepção inválido 31: erro de time-out na recepção 32: controle de fluxo configurado diferente de manual 33: controle de fluxo inválido para a porta serial configurada 34: recepção de dados não permitida no modo normal 35: recepção de dados não permitida no modo estendido 36: interrupção DCD não permitida 37: interrupção CTS não permitida 38: interrupção DSR não permitida 39: porta serial não configurada 50: erro interno na porta serial
Bits de comando, reiniciados automaticamente:			
%QX(n+2).0	tCommand. bStop	BIT	Parar o mestre.
%QX(n+2).1	tCommand. bRestart	BIT	Reiniciar o mestre.
%QX(n+2).2	tCommand. bResetCounter	BIT	Reiniciar as estatísticas dos diagnósticos (contadores).
%QX(n+2).3	tCommand. bDiag_19_reserved	BIT	Reservado
%QX(n+2).4	tCommand. bDiag_20_reserved	BIT	Reservado
%QX(n+2).5	tCommand. bDiag_21_reserved	BIT	Reservado
%QX(n+2).6	tCommand. bDiag_22_reserved	BIT	Reservado
%QX(n+2).7	tCommand. bDiag_23_reserved	BIT	Reservado
%QB(n+3)	byDiag_3_reserved	BYTE	Reservado

Variável de Representação Direta	Variável de diagnóstico do tipo T_DIAG_MODBUS_RTU_MASTER_1.*	Tamanho	Descrição
Estatísticas de comunicação:			
%QW(n+4)	tStat. wTXRequests	WORD	Contador de requisições transmitidas pelo mestre (0 a 65535).
%QW(n+6)	tStat. wRXNormalResponses	WORD	Contador de respostas normais recebidas pelo mestre (0 a 65535).
%QW(n+8)	tStat. wRXExceptionResponses	WORD	Contador de respostas com códigos de exceção recebidas pelo mestre (0 a 65535).
%QW(n+10)	tStat. wRXIllegalResponses	WORD	Contador de respostas ilegais recebidas pelo mestre – sintaxe inválida, número insuficiente de bytes recebidos, CRC inválido – (0 a 65535).
%QW(n+12)	tStat. wRXOverrunErrors	WORD	Contador de erros de overrun durante a recepção – UART FIFO ou fila RX – (0 a 65535).
%QW(n+14)	tStat. wRXIncompleteFrames	WORD	Contador de respostas com erro de construção, paridade ou falha durante a recepção (0 a 65535).
%QW(n+16)	tStat. wCTSTimeOutErrors	WORD	Contador de erro de time-out no CTS, utilizando o handshake RTS/CTS, durante a transmissão (0 a 65535).
%QW(n+18)	tStat. wDiag_18_Reserved	WORD	Reservado

Tabela 87: Diagnósticos MODBUS RTU Mestre

Nota:

Contadores: Todos os contadores dos diagnósticos do MODBUS RTU Mestre retornam à zero quando o valor limite 65535 é ultrapassado.

5.5.5.1.2. Configuração dos Dispositivos – Configuração por Mapeamento Simbólico

A configuração dos dispositivos, visualizada na figura abaixo, segue os seguintes parâmetros:

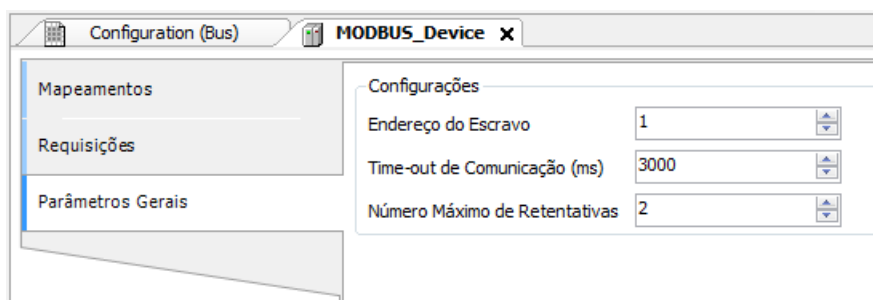


Figura 54: Tela de Configurações dos Parâmetros Gerais do Dispositivo

Configuração	Descrição	Padrão	Opções
Endereço do escravo	Endereço do escravo MOD-BUS	1	0 a 255
Time-out de comunicação (ms)	Define o time-out do nível de aplicação	3000	10 a 65535
Número Máximo de Retentativas	Define o número de tentativas antes de reportar um erro de comunicação	2	0 a 9

Tabela 88: Configurações do Dispositivo

Notas:

Endereço do escravo: De acordo com a Norma MODBUS, a faixa de endereços válidos para escravos é de 0 a 247, sendo os endereços 248 a 255 reservados. Quando o mestre envia um comando de escrita com o endereço configurado como zero, ele está realizando requisições broadcast na rede.

Time-out de comunicação: O time-out da comunicação é o tempo que o mestre aguardará por uma resposta do escravo à requisição. Para um dispositivo MODBUS RTU Mestre devem ser levadas em consideração ao menos as seguintes variáveis do sistema: o tempo que o escravo leva para transmitir o frame (de acordo com a taxa de transmissão), o tempo que o escravo leva para processar a requisição e o atraso de envio da resposta caso seja configurado no escravo. É recomendado que o time-out seja igual ou maior que o tempo para transmitir o frame somado ao atraso de envio da resposta e a duas vezes o tempo de processamento da requisição. Para mais informações, ver seção [Desempenho de Comunicação](#).

Número máximo de retentativas: Define o número de retentativas antes de reportar um erro de comunicação. Por exemplo, se o escravo não responder a um pedido e o mestre estiver configurado para enviar três retentativas, o número do contador de erros será incrementado por 1 unidade ao final da execução destas três retentativas. Após o incremento do erro o processo de tentativa de comunicação é reiniciado e caso o número de retentativas seja atingido novamente, novo erro será incrementado no contador.

5.5.5.1.3. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico

A configuração dos mapeamentos MODBUS, visualizada na figura abaixo, segue os parâmetros descritos na tabela abaixo:

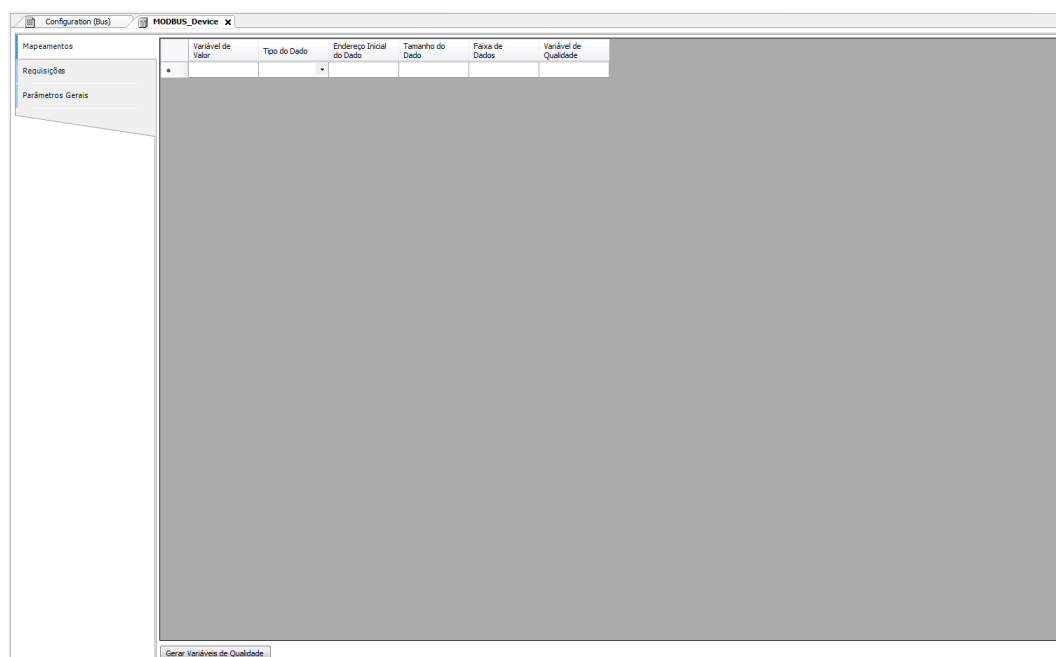


Figura 55: Tela de Mapeamentos de Dados MODBUS

Configuração	Descrição	Padrão	Opções
Variável de Valor	Nome da variável simbólica	-	Nome de uma variável declarada em um programa ou GVL
Tipo do Dado	Tipo de dado MODBUS	-	Coil - Escrita (1 bit) Coil - Leitura (1 bit) Holding Register - Escrita (16 bits) Holding Register - Leitura (16 bits) Holding Register – Máscara AND (16 bits) Holding Register – Máscara OR (16 bits) Input Register (16 bits) Input Status (1 bit)
Endereço Inicial do Dado	Endereço inicial dos dados MODBUS	-	1 a 65536
Tamanho do Dado	Tamanho do dado MODBUS	-	1 a 65536
Faixa de Dados	Faixa de endereços do dado configurado	-	-

Tabela 89: Configuração dos Mapeamentos MODBUS

Notas:

Variável de Valor: Esse campo é utilizado para especificar uma variável simbólica na relação MODBUS.

Tipo de Dado: Esse campo é utilizado para especificar o tipo de dado utilizado na relação MODBUS.

Tipo de Dado	Tamanho [bits]	Descrição
Coil - Escrita	1	Saída digital de escrita.
Coil - Leitura	1	Saída digital de leitura.
Holding Register - Escrita	16	Saída analógica de escrita.
Holding Register - Leitura	16	Saída analógica de leitura.
Holding Register - Máscara AND	16	Saída analógica que pode ser lida ou escrita com máscara AND.
Holding Register - Máscara OR	16	Saída analógica que pode ser lida ou escrita com máscara OR.
Input Register	16	Entrada analógica que pode ser apenas lida.
Input Status	1	Entrada digital que pode ser apenas lida.

Tabela 90: Tipos de Dados suportados no MODBUS

Endereço Inicial do Dado: Endereço inicial do dado de um mapeamento MODBUS.

Tamanho do Dado: O valor de Tamanho especifica a quantidade máxima de dados que uma relação MODBUS poderá acessar, a partir do endereço inicial. Sendo assim, para ler uma faixa de endereços contínua, é necessário que todos os endereços estejam declarados em uma única relação. Este campo varia de acordo com o tipo de dado MODBUS configurado.

Faixa de Dados: Este campo mostra ao usuário a faixa de endereços de memória utilizada pela relação MODBUS.

5.5.5.1.4. Configuração das Requisições – Configuração por Mapeamento Simbólico

A configuração das requisições MODBUS, visualizada na figura abaixo, segue os parâmetros descritos na tabela abaixo:

5. CONFIGURAÇÃO

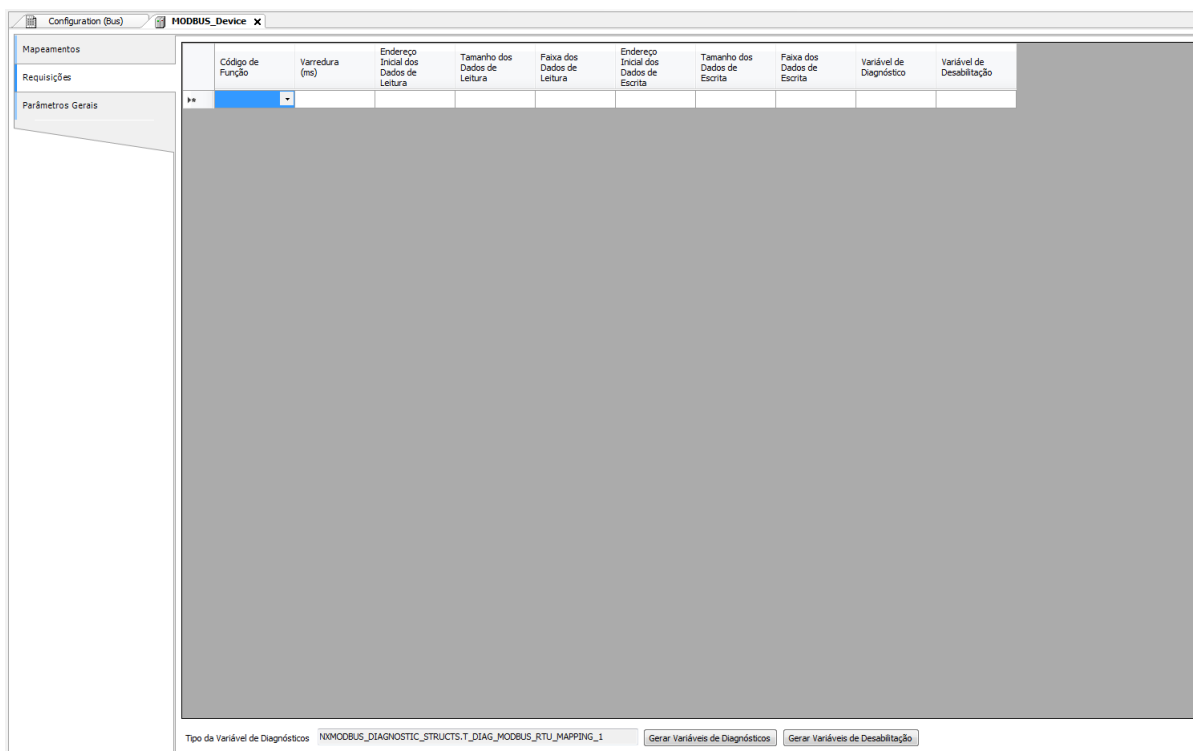


Figura 56: Tela de Requisições de dados MODBUS Mestre

Configuração	Descrição	Padrão	Opções
Código de Função	Tipo de função MODBUS	-	01 – Leitura de Coils 02 – Leitura de Input Status 03 – Leitura de Holding Registers 04 – Leitura de Input Registers 05 – Escrita de Um Coil 06 – Escrita de Um Register 15 – Escrita de Múltiplos Coils 16 – Escrita de Múltiplos Registers 22 – Escrita Mascarada de Register 23 – Leitura/Escrita de Múltiplos Registers
Varredura (ms)	Período de comunicação (ms)	100	0 a 3600000
Endereço Inicial do Dado de Leitura	Endereço inicial dos dados de leitura MODBUS	-	1 a 65536
Tamanho dos Dados de Leitura	Tamanho dos dados de leitura MODBUS	-	Depende da função utilizada
Faixa dos Dados de Leitura	Faixa de endereço dos dados de leitura MODBUS	-	0 a 2147483646
Endereço Inicial do Dado de Escrita	Endereço inicial dos dados de escrita MODBUS	-	1 a 65536

Configuração	Descrição	Padrão	Opções
Tamanho dos Dados de Escrita	Tamanho dos dados de escrita MODBUS	-	Depende da função utilizada
Faixa dos Dados de Escrita	Faixa de endereço dos dados de escrita MODBUS	-	0 a 2147483647
Variável de Diagnóstico	Nome da variável de diagnóstico	-	Nome de uma variável declarada em um programa ou GVL
Variável de Desabilitação	Variável utilizada para desabilitar a relação MODBUS	-	Campo destinado a variável simbólica utilizada para desabilitar, individualmente, as requisições MODBUS configuradas. Esta variável deve ser do tipo BOOL. A variável pode ser simples ou elemento de array e pode estar em estruturas.

Tabela 91: Configuração das Relações MODBUS

Notas:

Configuração: O número de configurações, padrão de fábrica e os valores da coluna opções, podem variar de acordo com o tipo de dado e função MODBUS (FC).

Código de Função: As funções MODBUS (FC) disponíveis são as seguintes:

Código		Descrição
DEC	HEX	
1	0x01	Leitura de coils (FC 01)
2	0x02	Leitura de input status (FC 02)
3	0x03	Leitura de holding registers (FC 03)
4	0x04	Leitura de input registers (FC 04)
5	0x05	Escrita de um coil (FC 05)
6	0x06	Escrita de um holding register (FC 06)
15	0x0F	Escrita de múltiplos coils (FC 15)
16	0x10	Escrita de múltiplos holding registers (FC 16)
22	0x16	Escrita mascarada de um holding register (FC 22)
23	0x17	Leitura/escrita de múltiplos holding registers (FC 23)

Tabela 92: Funções MODBUS Suportadas pelas UCPs Nexto

Varredura: Este parâmetro indica com que frequência a comunicação definida por esta requisição deve ser executada. Ao ser finalizada uma comunicação será aguardado um tempo igual ao configurado no campo varredura e, após, será executada uma nova comunicação.

Endereço Inicial dos Dados de Leitura: Campo destinado ao endereço inicial dos dados de leitura MODBUS.

Tamanho dos Dados de Leitura: O valor mínimo para o tamanho dos dados de leitura é 1 e o valor máximo depende da função MODBUS (FC) utilizada, conforme abaixo:

- Leitura de Coils (FC 1): 2000
- Leitura de Input Status (FC 2): 2000
- Leitura de Holding Registers (FC 3): 125
- Leitura de Input Registers (FC 4): 125

- Leitura/Escrita de Holding Registers (FC 23): 121

Faixa dos Dados de Leitura: Este campo mostra a faixa de dados de leitura MODBUS configurada para cada requisição. O endereço inicial de leitura, somado ao tamanho do dado de leitura resultará na faixa de dados de leitura de cada uma das requisições.

Endereço Inicial dos Dados de Escrita: Campo destinado ao endereço inicial dos dados de escrita MODBUS.

Tamanho dos Dados de Escrita: O valor mínimo para o tamanho dos dados de escrita é 1 e o valor máximo depende da função MODBUS (FC) utilizada, conforme abaixo:

- Escrita de Um Coil (FC 5): 1
- Escrita de Um Holding Register (FC 6): 1
- Escrita de Múltiplos Coils (FC 15): 1968
- Escrita de Holding Registers (FC 16): 123
- Máscara de Escrita do Register (FC 22): 1
- Leitura/Escrita de Holding Registers (FC 23): 121

Faixa dos Dados de Escrita: Este campo mostra a faixa de dados de escrita MODBUS configurada para cada requisição. O endereço inicial de escrita, somado ao tamanho do dado de escrita resultará na faixa de dados de escrita de cada uma das requisições.

Variável de Diagnóstico: Os diagnósticos da requisição MODBUS configurada, seja por mapeamento simbólico ou por representação direta, são armazenados em variáveis do tipo *T_DIAG_MODBUS_RTU_MAPPING_1* para dispositivos Mestre e *T_DIAG_MODBUS_ETH_CLIENT_1* para dispositivos Cliente e para o mapeamento por representação direta estão em 4 bytes e 2 words, os quais estão descritos na Tabela 93 ("n" é o valor configurado no campo *Endereço Inicial de Diagnósticos em %Q*).

Variável de Representação Direta	Variável de diagnóstico do tipo T_DIAG_MODBUS_RTU_MAPPING_1.*	Tamanho	Descrição
Bits de estado da comunicação:			
%QX(n).0	byStatus. bCommIdle	BIT	Comunicação inativa (aguardando ser constatada).
%QX(n).1	byStatus. bCommExecuting	BIT	Comunicação ativa.
%QX(n).2	byStatus. bCommPostponed	BIT	Comunicação adiada, pois o número máximo de requisições simultâneas foi atingido. As comunicações adiadas serão executadas na mesma sequência em que elas foram requisitadas, evitando a indeterminação. O tempo gasto nesse estado não é contabilizado para efeitos de time-out. Os bits bCommIdle e bCommExecuting são falsos quando o bit bCommPostponed é verdadeiro.
%QX(n).3	byStatus. bCommDisabled	BIT	Comunicação desabilitada. O bit bCommIdle é reiniciado nessa condição.
%QX(n).4	byStatus. bCommOk	BIT	Comunicação finalizada anteriormente foi realizada com sucesso.
%QX(n).5	byStatus. bCommError	BIT	Comunicação finalizada anteriormente teve um erro. Verificar código de erro.
%QX(n).6	byStatus. bCommAborted	BIT	Não utilizado no MODBUS Mestre RTU.

Variável de Representação Direta	Variável de diagnóstico do tipo T_DIAG_MODBUS_RTU_MAPPING_1.*	Tamanho	Descrição
%QX(n).7	byStatus. bDiag_7_reserved	BIT	Reservado
Último código de erro (habilitado quando o bCommError = verdadeiro):			
%QB(n+1)	eLastErrorCode	MASTER_ERROR_CODE (BYTE)	Informa a possível causa do último erro ocorrido na relação MODBUS. Consulte a Tabela 116 para detalhes das possibilidades.
Último código de exceção recebido pelo mestre:			
%QB(n+2)	eLastExceptionCode	MODBUS_EXCEPTION (BYTE)	NO_EXCEPTION (0) FUNCTION_NOT_SUPPORTED (1) MAPPING_NOT_FOUND (2) ILLEGAL_VALUE (3) ACCESS_DENIED (128)* MAPPING_DISABLED (129)* IGNORE_FRAME (255)*
Estatísticas de comunicação:			
%QB(n+3)	byDiag_3_reserved	BYTE	Reservado.
%QW(n+4)	wCommCounter	WORD	Contador de comunicações finalizadas, com ou sem erros. O usuário pode testar quando a comunicação foi finalizada testando a variação desse contador. Quando o valor 65535 é atingido, o contador retorna à zero.
%QW(n+6)	wCommErrorCounter	WORD	Contador de comunicações finalizadas com erros. Quando o valor 65535 é atingido, o contador retorna a zero.

Tabela 93: Diagnósticos das relações MODBUS

Notas:

Códigos de exceção: Os códigos de exceção apresentados neste campo são os valores retornados pelo escravo. As definições dos códigos de exceção 128, 129 e 255, apresentadas nessa tabela, são válidas apenas na utilização de escravos Altus. Para escravos de outros fabricantes esses códigos de exceção podem ter significados diferentes.

Variável de Desabilitação: Campo destinado a variável do tipo booleana utilizada para desabilitar, individualmente, as requisições MODBUS configuradas na aba Requisições através do botão na parte inferior da janela. A requisição é desabilitada quando a variável, correspondente a requisição, for igual a 1, caso contrário, a requisição está habilitada.

Último código de Erro: Os códigos das possíveis situações que ocasionam erro na comunicação MODBUS podem ser consultados abaixo:

Código	Enumerável	Descrição
1	ERR_EXCEPTION	Resposta reportada em um código de exceção (ver eLastExceptionCode = Código de Exceção).
2	ERR_CRC	Resposta com CRC inválido.
3	ERR_ADDRESS	Endereço MODBUS não encontrado. O endereço que respondeu à requisição foi diferente do esperado.
4	ERR_FUNCTION	Código inválido da função. A função recebida na resposta foi diferente da esperada pela requisição.
5	ERR_FRAME_DATA_COUNT	A quantidade de dados da resposta foi diferente da esperada.
7	ERR_NOT_ECHO	Resposta não é eco da pergunta (FC 5 e 6).

Código	Enumerável	Descrição
8	ERR_REFERENCE_NUMBER	Número de referência inválido (FC 15 e 16).
9	ERR_INVALID_FRAME_SIZE	Resposta menor do que a esperada.
20	ERR_CONNECTION	Erro durante a fase de conexão.
21	ERR_SEND	Erro durante a fase de transmissão.
22	ERR_RECEIVE	Erro durante a fase de recepção.
40	ERR_CONNECTION_TIMEOUT	Time-out no nível de aplicação durante a conexão.
41	ERR_SEND_TIMEOUT	Time-out no nível de aplicação durante a transmissão.
42	ERR_RECEIVE_TIMEOUT	Time-out no nível de aplicação enquanto aguarda resposta.
43	ERR_CTS_OFF_TIMEOUT	Time-out enquanto aguarda CTS = falso na transmissão.
44	ERR_CTS_ON_TIMEOUT	Time-out enquanto aguarda CTS = verdadeiro na transmissão.
128	NO_ERROR	Sem erro desde a inicialização.

Tabela 94: Códigos de Erro das relações MODBUS

ATENÇÃO

Diferentemente de outras tarefas de uma aplicação, quando for atingida uma marca de depuração na MainTask, a tarefa de uma instância MODBUS RTU Mestre, e qualquer outra tarefa MODBUS, irá parar de ser executada no momento em que tentar efetuar uma escrita em uma área de memória. Isto ocorre para manter a consistência dos dados das áreas de memória enquanto a MainTask não estiver em execução.

5.5.5.2. Configuração do Protocolo MODBUS Mestre por Representação Direta (%Q)

Para configurar este protocolo usando representação direta (%Q), é necessário executar os seguintes passos:

- Configurar os parâmetros gerais do protocolo MODBUS, como: tempos de comunicação e variáveis de representação direta (%Q) para receber os diagnósticos.
- Adicionar e configurar dispositivos, definindo endereço, variáveis de representação direta (%Q) para desabilitar as relações, time-outs de comunicação, etc.
- Adicionar e configurar relações MODBUS, especificando o tipo de dado e função MODBUS, time-outs, variáveis de representação direta (%Q) para receber os diagnósticos da relação e outras para receber/escrever os dados, quantidade de dados a comunicar e varredura da relação.

As descrições de cada configuração estão relacionadas a seguir, neste capítulo.

5.5.5.2.1. Parâmetros Gerais do Protocolo MODBUS Mestre – Configuração por Representação Direta (%Q)

Os parâmetros gerais, encontrados na tela inicial de configuração do protocolo MODBUS (figura abaixo), são definidos como:

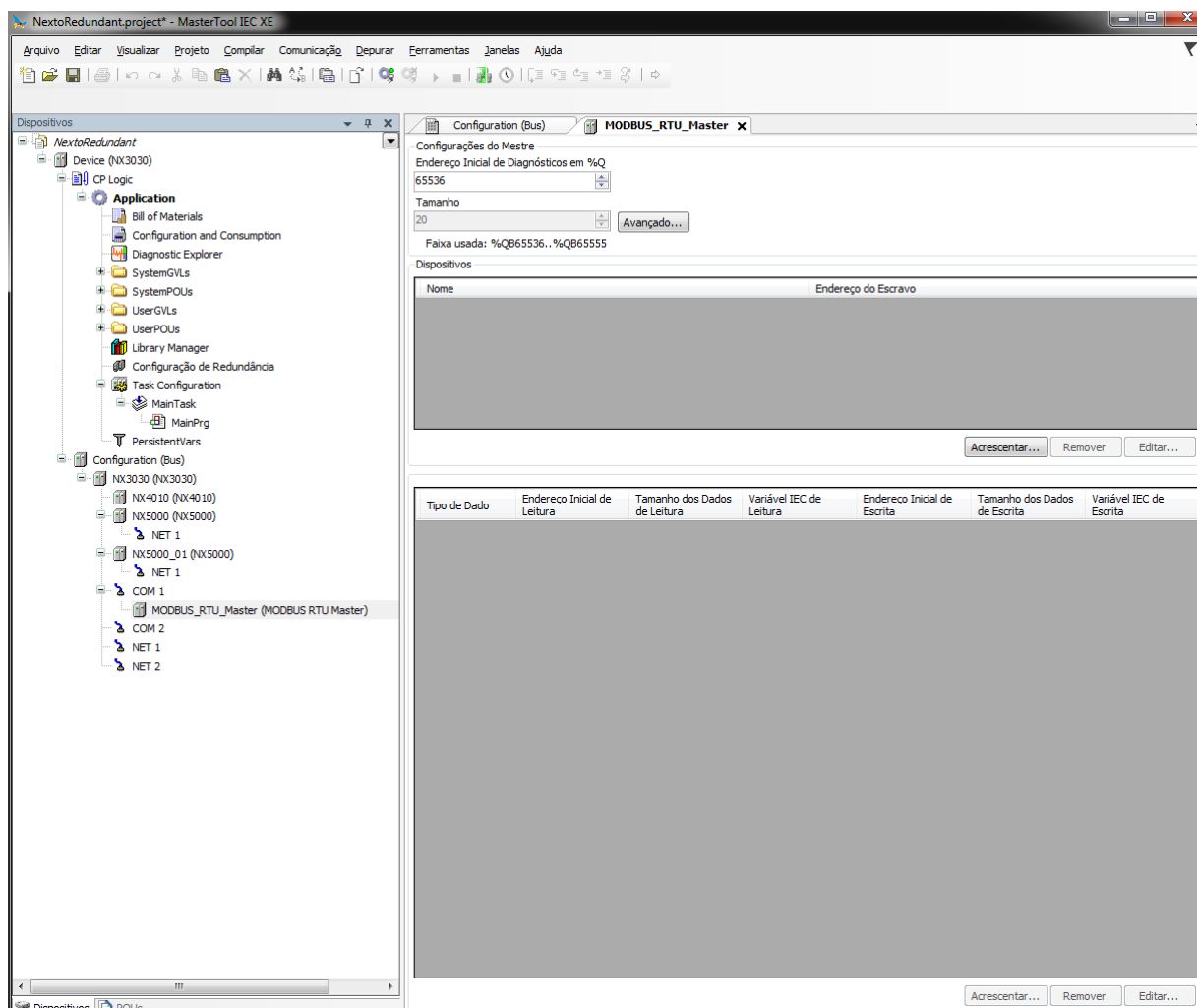


Figura 57: Tela de Configuração MODBUS RTU Mestre

Variáveis de representação direta (%Q) para os diagnósticos do protocolo:

Configuração	Descrição	Padrão	Opções
Endereço Inicial de Diagnósticos em %Q	Endereço inicial das variáveis de diagnóstico	-	0 a 2147483628
Tamanho	Tamanho da área de diagnósticos	20	Desabilitado para edição

Tabela 95: Configurações do Mestre – MODBUS Mestre

Notas:

Endereço Inicial de Diagnósticos em %Q: Esse campo é limitado pelo tamanho da memória de variáveis de saídas endereçáveis (%Q) de cada UCP, a qual pode ser consultada na seção [Memória](#).

Padrão: O padrão de fábrica não pode ser definido para o campo *Endereço Inicial de Diagnósticos em %Q*, pois a criação de uma instância do protocolo pode ser realizada em qualquer momento no desenvolvimento da aplicação, fazendo com que o próprio software MasterTool IEC XE aloque um valor, da faixa de variáveis de saída de representação direta (%Q), ainda não utilizado.

Os diagnósticos e comandos do protocolo MODBUS estão descritos na Tabela [87](#).

Os tempos de comunicação do protocolo MODBUS Mestre, encontrados no botão *Avançado...* da tela de configuração, estão divididos em *Atraso do Envio* e *Interframe Mínimo*, maiores detalhes estão descritos na seção [Parâmetros Gerais do Protocolo MODBUS Mestre – Configuração por Mapeamento Simbólico](#).

5.5.5.2.2. Configuração dos Dispositivos – Configuração por Representação Direta (%Q)

A configuração dos dispositivos, visualizada na figura abaixo, segue os seguintes parâmetros:

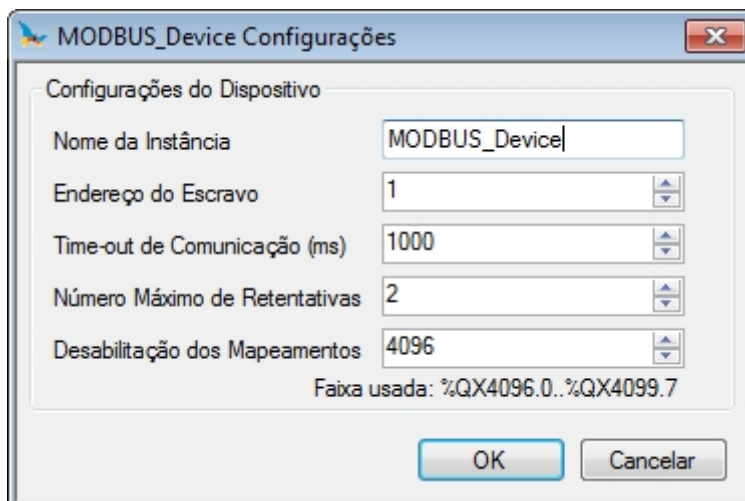


Figura 58: Configurações do Dispositivo

Configuração	Descrição	Padrão	Opções
Nome da Instância	Nome da instância	MODBUS_Device	Identificador, segundo a IEC 61131-3
Endereço do Escravo	Endereço do escravo MODBUS	1	0 a 255
Time-out de Comunicação (ms)	Define o time-out do nível de aplicação	1000	10 a 65535
Número Máximo de Retentativas	Define o número de tentativas antes de reportar um erro de comunicação	2	0 a 9
Desabilitação dos Mapeamentos	Endereço inicial utilizado para desabilitar as relações MODBUS	-	0 a 2147483644

Tabela 96: Configurações do Dispositivo – MODBUS Mestre

Notas:

Nome da Instância: Esse campo é o identificador do dispositivo, o qual é verificado segundo a IEC 61131-3, ou seja, não permite espaços, caracteres especiais e iniciar com caractere numeral. É limitado em 24 caracteres.

Desabilitação dos Mapeamentos: Composta por 32 bits, utilizados para desabilitar, individualmente, as 32 relações MODBUS configuradas no espaço *Mapeamentos do Dispositivo*. A relação é desabilitada quando o bit, correspondente à relação, for igual a 1, caso contrário, o mapeamento está habilitado. Esse campo é limitado pelo tamanho da memória de variáveis de saídas endereçáveis (%Q) de cada UCP, a qual pode ser consultada na seção [Memória](#).

Padrão: O padrão de fábrica não pode ser definido para o campo *Desabilitação dos Mapeamentos*, pois a criação de uma instância do protocolo pode ser realizada em qualquer momento no desenvolvimento da aplicação, fazendo com que o próprio software MasterTool IEC XE aloque um valor, da faixa de variáveis de saída de representação direta (%Q), ainda não utilizado.

Para maiores detalhes dos parâmetros *Endereço do Escravo*, *Time-out de Comunicação* e *Número máximo de retentativas* ver notas na seção [Configuração dos Dispositivos – Configuração por Mapeamento Simbólico](#).

5.5.5.2.3. Configuração dos Mapeamentos – Configuração por Representação Direta (%Q)

A configuração das relações MODBUS, visualizada nas figuras abaixo, segue os parâmetros descritos na tabela abaixo:

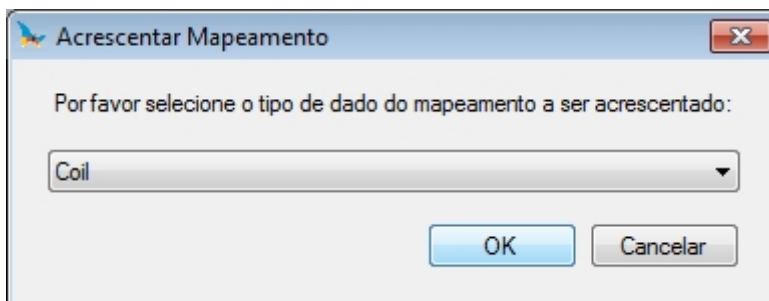


Figura 59: Tipo de Dado MODBUS

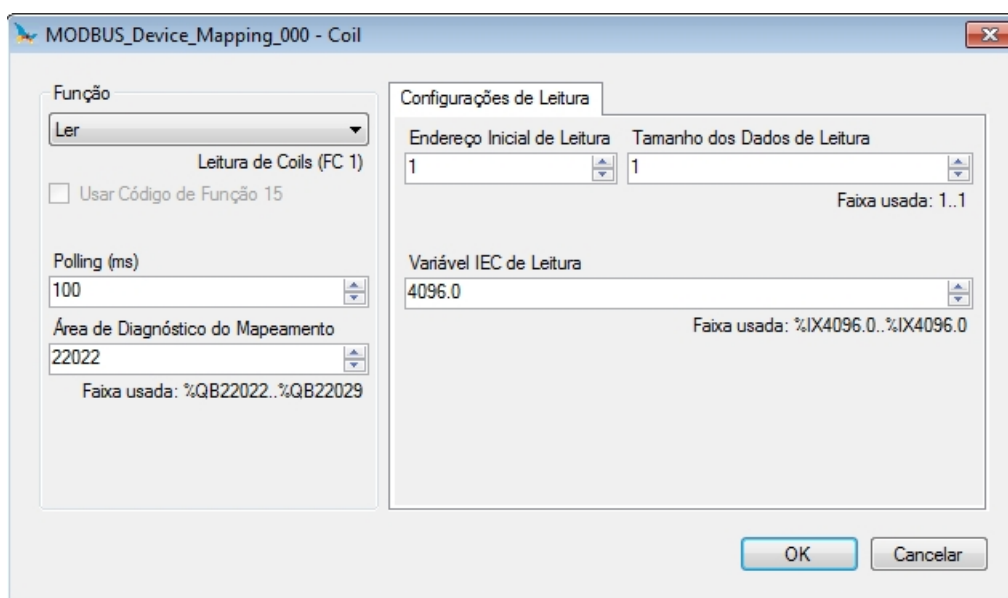


Figura 60: Função MODBUS

Na tabela abaixo, o número de configurações, padrão de fábrica e os valores da coluna opções, podem variar de acordo com o tipo de dado e função MODBUS (FC).

Configuração	Descrição	Padrão	Opções
Função	Tipo de função MODBUS	Ler	Ler Escrever Ler/Escrever Máscara de Escrita
Polling (ms)	Período de comunicação (ms)	100	0 a 3600000
Área de Diagnóstico do Mapeamento	Endereço inicial dos diagnósticos da relação MODBUS (%Q)	-	0 a 2147483640

Configuração	Descrição	Padrão	Opções
Endereço Inicial de Leitura	Endereço inicial dos dados de leitura MODBUS	1	1 a 65536
Tamanho dos Dados de Leitura	Número de dados de leitura MODBUS	-	Depende da função utilizada
Variável IEC de Leitura	Endereço inicial das variáveis de leitura (%I)	-	0 a 2147483646
Endereço Inicial de Escrita	Endereço inicial dos dados de escrita MODBUS	1	1 a 65536
Tamanho dos Dados de Escrita	Número de dados de escrita MODBUS	-	Depende da função utilizada
Variável IEC de Escrita	Endereço inicial das variáveis de escrita (%Q)	-	0 a 2147483647
Máscara de Escrita das Variáveis IEC	Endereço inicial das variáveis para a máscara de escrita (%Q)	-	0 a 2147483644

Tabela 97: Mapeamentos do Dispositivo

Notas:

Função: Os tipos de dados disponíveis estão detalhados na Tabela 116 e as funções MODBUS (FC) disponíveis na Tabela 114.

Polling: Este parâmetro indica com que frequência a comunicação definida por esta relação deve ser executada. Ao ser finalizada uma comunicação será aguardado um tempo igual ao polling configurado e, após, será executada uma nova comunicação o mais rápido possível.

Área de Diagnósticos do Mapeamento: Esse campo é limitado pelo tamanho da memória de variáveis de saídas endereçáveis (%Q) de cada UCP, a qual pode ser consultada na seção **Memória**. Os diagnósticos da relação MODBUS configurada estão descritos na Tabela 93.

Tamanho dos Dados de Leitura e de Escrita: Detalhes do tamanho dos dados suportados por cada função está descrito nas notas da seção **Configuração das Requisições – Configuração por Mapeamento Simbólico**.

ATENÇÃO

Quando o acesso à memória de dados da comunicação for entre equipamentos com endianness diferentes (Little-Endian e Big-Endian), pode ocorrer a inversão dos dados lidos/escritos. Neste caso o usuário deve fazer o ajuste dos dados na aplicação.

Variável IEC de Leitura: Caso o tipo de dado MODBUS seja *Coil* ou *Input Status* (bit), o endereço inicial das variáveis IEC de leitura terá o formato por exemplo %IX10.1. Porém, se o tipo de dado MODBUS for *Holding Register* ou *Input Register* (16 bits), o endereço inicial das variáveis IEC de leitura terá o formato %IW. Esse campo é limitado pelo tamanho da memória de variáveis de entrada endereçáveis (%I) de cada UCP, a qual pode ser consultada na seção **Memória**.

Variável IEC de Escrita: Caso o tipo de dado MODBUS seja *Coil*, o endereço inicial das variáveis IEC de escrita terá o formato por exemplo %QX10.1. Porém, se o tipo de dado MODBUS for *Holding Register* (16 bits), o endereço inicial das variáveis IEC de escrita terá o formato %QW. Esse campo é limitado pelo tamanho da memória de variáveis de saídas endereçáveis (%Q) de cada UCP, a qual pode ser consultada na seção **Memória**.

Máscara de Escrita do Register: A função *Máscara de Escrita* (FC 22), através de uma lógica entre o valor já escrito e as duas words configuradas neste campo, sendo a %QW(0) para a máscara AND e a %QW(2) para a máscara OR; permite ao usuário manipular a word. Esse campo é limitado pelo tamanho da memória de variáveis de saídas endereçáveis (%Q) de cada UCP, a qual pode ser consultada na seção **Memória**.

Padrão: O padrão de fábrica não pode ser definido para os campos *Área de Diagnóstico do Mapeamento*, *Variável IEC de Leitura*, *Variável IEC de Escrita* e *Máscara de Escrita das Variáveis IEC*, pois a criação de uma relação pode ser realizada em qualquer momento no desenvolvimento da aplicação, fazendo com que o próprio software MasterTool IEC XE aloque um valor, da faixa de variáveis de saída de representação direta (%Q), ainda não utilizado. O padrão de fábrica não pode ser definido para os campos *Tamanho dos Dados de Leitura* e *Tamanho dos Dados de Escrita*, pois eles vão variar de acordo com o tipo de dado MODBUS selecionado.

ATENÇÃO

Diferentemente de outras tarefas de uma aplicação, quando for atingida uma marca de depuração na MainTask, a tarefa de uma instância MODBUS RTU Mestre, e qualquer outra tarefa MODBUS, irá parar de ser executada no momento em que tentar efetuar uma escrita em uma área de memória. Isto ocorre para manter a consistência dos dados das áreas de memória enquanto a MainTask não estiver em execução.

5.5.6. MODBUS RTU Escravo

Este protocolo está disponível para as UCPs da Série Nexto nos seus canais seriais. Ao selecionar esta opção no MasterTool IEC XE, a UCP passa a ser escravo da comunicação MODBUS, permitindo a conexão com dispositivos mestre MODBUS RTU.

Há dois modos de configuração para este protocolo. Um faz uso de representação direta (%Q), no qual as variáveis são definidas pelo seu endereço. O outro, através de mapeamento simbólico, tem as variáveis definidas pelo seu nome.

Independente do modo de configuração, os passos para inserir uma instância do protocolo e configurar a interface serial são iguais. O procedimento para inserir uma instância de protocolo é encontrado com detalhes no Manual de Utilização do MasterTool IEC XE – MU299048. As demais etapas de configuração serão descritas a seguir para cada modalidade.

- Adicionar a instância do protocolo MODBUS RTU Escravo ao canal serial COM 1 ou COM 2 (ou ambos, em caso de duas redes de comunicação). Para realizar esse procedimento, consultar a seção [Inserindo uma Instância de Protocolo](#).
- Configurar a interface serial, escolhendo a velocidade de comunicação, o comportamento dos sinais RTS/CTS, a paridade, os bits de parada do canal, entre outros. Consultar a seção [Configuração das Interfaces Seriais](#).

5.5.6.1. Configuração do Protocolo MODBUS Escravo por Mapeamento Simbólico

Para configurar este protocolo usando Mapeamento Simbólico, é necessário executar os seguintes passos:

- Configurar os parâmetros gerais do protocolo MODBUS escravo, como: endereço do escravo e tempos de comunicação (disponível no botão de configurações avançadas do Escravo).
- Adicionar e configurar as relações MODBUS, especificando o nome da variável, tipo de dado MODBUS, o endereço inicial do dado e automaticamente são preenchidos o tamanho do dado e a faixa de acordo com o tipo da variável declarada.

5.5.6.1.1. Parâmetros Gerais do Protocolo MODBUS Escravo – Configuração por Mapeamento Simbólico

Os parâmetros gerais, encontrados na tela inicial de configuração do protocolo MODBUS como apresentado na figura abaixo.

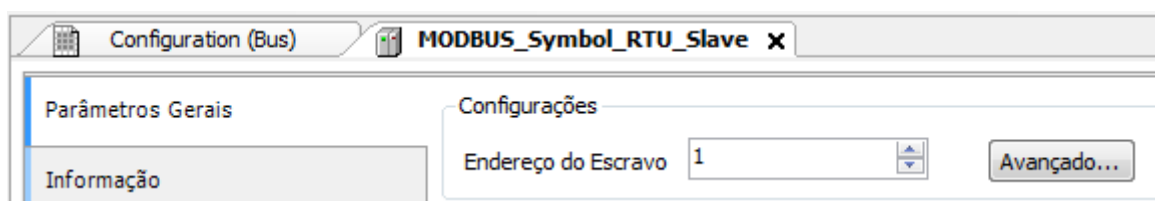


Figura 61: Configuração do Escravo

Configuração	Descrição	Padrão	Opções
Endereço do Escravo	Endereço do escravo MOD-BUS	1	1 a 255

Tabela 98: Configurações do Escravo

Os tempos de comunicação do protocolo MODBUS escravo, encontrados no botão *Avançado...* da tela de configuração, estão divididos em: *Ciclo da Tarefa*, *Atraso do Envio* e *Interframe Mínimo* como pode ser visto na figura abaixo e na tabela abaixo.

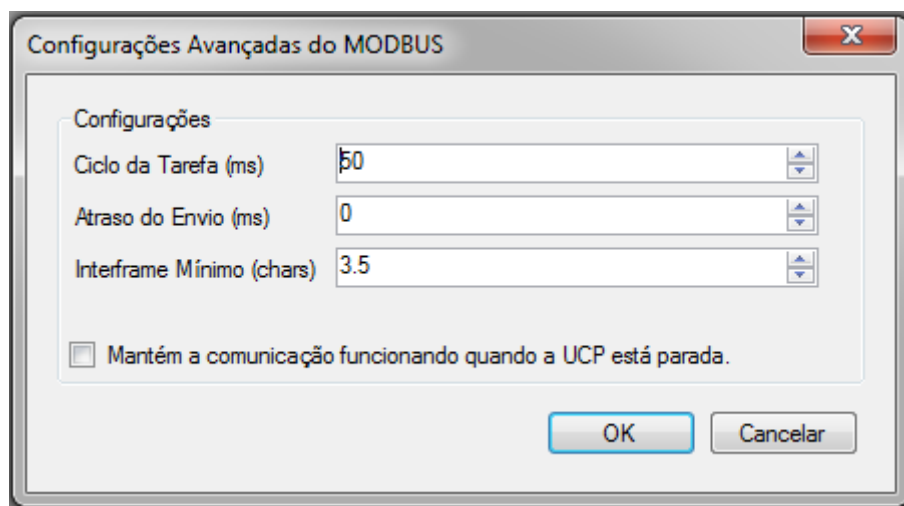


Figura 62: Configurações Avançadas MODBUS do Escravo

Configuração	Descrição	Padrão	Opções
Ciclo da Tarefa (ms)	Tempo para execução da instância dentro do ciclo, sem considerar o tempo de execução da mesma	50	20 a 100
Atraso do Envio (ms)	Tempo de atraso para envio da resposta	0	0 a 65535
Interframe Mínimo (chars)	Tempo mínimo de silêncio entre diferentes frames	3.5	3.5 a 100.0
Mantém a comunicação funcionando quando a UCP está parada.	Habilita o Modbus Symbol Slave para comunicar enquanto a UCP estiver em STOP ou parada em um breakpoint	desmarcado	marcado ou desmarcado

Tabela 99: Configurações Avançadas MODBUS Escravo

Notas:

Ciclo da Tarefa: O usuário deverá ter cuidado ao alterar esse parâmetro, pois o mesmo interfere diretamente no tempo de resposta, volume de dados por varredura e, principalmente, no balanceamento dos recursos da UCP entre comunicações e outras tarefas.

Atraso do Envio: A resposta à uma requisição MODBUS pode causar problemas em certos momentos, como, por exemplo, na interface RS-485 ou outra half-duplex. Às vezes existe um atraso entre o tempo da requisição do mestre e o silêncio na linha física (atraso no mestre para colocar RTS em zero e colocar o transmissor RS-485 em alta impedância). Para resolver o problema, o escravo pode esperar o tempo determinado nesse campo antes de enviar a resposta. Caso contrário, os primeiros bytes transmitidos pelo escravo, durante a resposta, podem ser perdidos.

Interframe Mínimo: A norma MODBUS define esse tempo como 3.5 caracteres, porém esse parâmetro é configurável para atender aos dispositivos que não estão de acordo com o padrão.

Os diagnósticos e comandos do protocolo MODBUS Escravo configurado, seja por mapeamento simbólico ou por representação direta, são armazenados em variáveis do tipo *T_DIAG_MODBUS_RTU_SLAVE_I* e ainda para o mapeamento por representação direta estão em 4 bytes e 8 words, os quais estão descritos na tabela abaixo (n é o valor configurado no campo *Endereço Inicial de Diagnósticos em %Q*):

Variável de Representação Direta	Variável de diagnóstico do tipo T_DIAG_MODBUS_RTU_SLAVE_1.*	Tamanho	Descrição
Bits de diagnóstico:			
%QX(n).0	tDiag. bRunning	BIT	O escravo está em execução.
%QX(n).1	tDiag. bNotRunning	BIT	O escravo não está em execução (ver bit: bInterruptedByCommand).
%QX(n).2	tDiag. bInterruptedByCommand	BIT	O bit bNotRunning foi habilitado, pois o escravo foi interrompido pelo usuário através de bits de comando.
%QX(n).3	tDiag. bConfigFailure	BIT	Diagnóstico descontinuado.
%QX(n).4	tDiag. bRXFailure	BIT	Diagnóstico descontinuado.
%QX(n).5	tDiag. bTXFailure	BIT	Diagnóstico descontinuado.
%QX(n).6	tDiag. bModuleFailure	BIT	Diagnóstico descontinuado.
%QX(n).7	tDiag. bDiag_7_reserved	BIT	Reservado
Códigos de erro:			
%QB(n+1)	eErrorCode	SERIAL_STATUS (BYTE)	<p>0: não existem erros</p> <p>1: porta serial inválida</p> <p>2: modo da porta serial inválido</p> <p>3: taxa de transmissão inválida</p> <p>4: bits de dado inválidos</p> <p>5: bits de paridade inválidos</p> <p>6: bits de parada inválidos</p> <p>7: parâmetro de sinal de modem inválido</p> <p>8: parâmetro de Threshold de RX da UART inválido</p> <p>9: parâmetro de time-out inválido</p> <p>10: porta serial ocupada</p> <p>11: erro de hardware na UART</p> <p>12: erro de hardware remoto</p> <p>20: tamanho do buffer de transmissão inválido</p> <p>21: método de sinal de modem inválido</p> <p>22: time-out de CTS = verdadeiro</p> <p>23: time-out de CTS = falso</p> <p>24: erro de time-out na transmissão</p> <p>30: tamanho do buffer de recepção inválido</p> <p>31: erro de time-out na recepção</p> <p>32: controle de fluxo configurado diferente de manual</p> <p>33: controle de fluxo inválido para a porta serial configurada</p> <p>34: recepção de dados não permitida no modo normal</p>

Variável de Representação Direta	Variável de diagnóstico do tipo T_DIAG_MODBUS_RTU_SLAVE_1.*	Tamanho	Descrição
			35: recepção de dados não permitida no modo estendido 36: interrupção DCD não permitida 37: interrupção CTS não permitida 38: interrupção DSR não permitida 39: porta serial não configurada 50: erro interno na porta serial
Bits de comando, reiniciados automaticamente:			
%QX(n+2).0	tCommand. bStop	BIT	Parar o escravo.
%QX(n+2).1	tCommand. bRestart	BIT	Reiniciar o escravo.
%QX(n+2).2	tCommand. bResetCounter	BIT	Reiniciar as estatísticas dos diagnósticos (contadores).
%QX(n+2).3	tCommand. bDiag_19_reserved	BIT	Reservado
%QX(n+2).4	tCommand. bDiag_20_reserved	BIT	Reservado
%QX(n+2).5	tCommand. bDiag_21_reserved	BIT	Reservado
%QX(n+2).6	tCommand. bDiag_22_reserved	BIT	Reservado
%QX(n+2).7	tCommand. bDiag_23_reserved	BIT	Reservado
%QB(n+3)	byDiag_3_reserved	BYTE	Reservado
Estatísticas de comunicação:			
%QW(n+4)	tStat. wRXRequests	WORD	Contador de requisições normais recebidas pelo escravo e respondidas normalmente. No caso de um comando broadcast, esse contador é incrementado, porém não é transmitida a resposta (0 a 65535).
%QW(n+6)	tStat. wTXExceptionResponses	WORD	Contador de requisições normais recebidas pelo escravo e respondidas com códigos de exceção. No caso de um comando broadcast, esse contador é incrementado, porém não é transmitida a resposta (0 a 65535). Códigos de exceção: 1: o código da função (FC) é legal, porém não suportado. 2: relação não encontrada nestes dados MODBUS. 3: valor ilegal para o endereço. 128: o mestre/cliente não tem direito de escrita ou leitura. 129: a relação MODBUS está desabilitada.

Variável de Representação Direta	Variável de diagnóstico do tipo T_DIAG_MODBUS_RTU_SLAVE_1.*	Tamanho	Descrição
%QW(n+8)	tStat. wRXFrames	WORD	Contador de frames recebidos pelo escravo. Considera-se frame, algo que é processado e seguido de um período mínimo de silêncio interframes (Interframe Mínimo), ou seja, uma mensagem ilegal também é computada (0 a 65535).
%QW(n+10)	tStat. wRXIllegalRequests	WORD	Contador de requisições ilegais. Estes são frames que iniciam com o endereço 0 (broadcast) ou com o endereço MODBUS do escravo, mas não são requisições legais – sintaxe inválida, frames menores, CRC inválido – (0 a 65535).
%QW(n+12)	tStat. wRXOverrunErrors	WORD	Contador de frames com erros de overrun durante a recepção – UART FIFO ou fila RX – (0 a 65535).
%QW(n+14)	tStat. wRXIncompleteFrames	WORD	Contador de frames com erro de construção, paridade ou falha durante a recepção (0 a 65535).
%QW(n+16)	tStat. wCTSTimeOutErrors	WORD	Contador de erro de time-out no CTS, utilizando o handshake RTS/CTS, durante a transmissão (0 a 65535).
%QW(n+18)	tStat. wDiag_18_Reserved	WORD	Reservado

Tabela 100: Diagnósticos MODBUS RTU Escravo

Nota:

Contadores: Todos os contadores dos diagnósticos do MODBUS RTU Escravo retornam à zero quando o valor limite 65535 é ultrapassado.

5.5.6.1.2. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico

A configuração dos mapeamentos MODBUS, visualizada na figura abaixo, segue os parâmetros descritos na tabela abaixo:

Mapeamentos						
	Variável de Valor	Tipo do Dado	Endereço Inicial do Dado	Endereço Inicial Absoluto do Dado	Tamanho do Dado	Faixa de Dados
▶*						

Figura 63: Tela de Mapeamentos de dados MODBUS

Configuração	Descrição	Padrão	Opções
Variável de Valor	Nome da variável simbólica	-	Nome de uma variável declarada em um programa ou GVL
Tipo de Dado	Tipo de dado MODBUS	-	Coil Input Status Holding Register Input Register
Endereço Inicial do Dado	Endereço inicial dos dados MODBUS	-	1 a 65536
Endereço Inicial Absoluto do Dado	Endereço inicial absoluto dos dados MODBUS conforme o seu tipo.	-	-
Tamanho do Dado	Tamanho do dado MODBUS	-	1 a 65536
Faixa de Dados	Faixa de endereços do dado configurado	-	-

Tabela 101: Configuração dos Mapeamentos MODBUS

Notas:

Variável de Valor: Esse campo é utilizado para especificar uma variável simbólica na relação MODBUS.

Tipo de Dado: Esse campo é utilizado para especificar o tipo de dado utilizado na relação MODBUS.

Tipo de Dado	Tamanho [bits]	Descrição
Coil	1	Saída digital que pode ser lida ou escrita.
Input Status	1	Entrada digital que pode ser apenas lida.
Holding Register	16	Saída analógica que pode ser lida ou escrita.
Input Register	16	Entrada analógica que pode ser apenas lida.

Tabela 102: Tipos de Dados MODBUS suportados pelas UCPs Nexto

Endereço Inicial do Dado: Endereço inicial do dado de um mapeamento MODBUS.

Tamanho do Dado: O valor de tamanho especifica a quantidade máxima de dados que uma relação MODBUS poderá acessar, a partir do endereço inicial. Sendo assim, para ler uma faixa de endereços contínua, é necessário que todos os endereços estejam declarados em uma única relação. Este campo varia de acordo com o tipo de dado MODBUS configurado.

Faixa de Dados: Este campo mostra ao usuário a faixa de endereços de memória utilizada pela relação MODBUS.

ATENÇÃO

Diferentemente de outras tarefas de uma aplicação, quando for atingida uma marca de depuração na MainTask, a tarefa de uma instância MODBUS RTU Escravo, e qualquer outra tarefa MODBUS, irá parar de ser executada no momento em que tentar efetuar uma escrita em uma área de memória. Isto ocorre para manter a consistência dos dados das áreas de memória enquanto a MainTask não estiver em execução.

5.5.6.2. Configuração do Protocolo MODBUS Escravo por Representação Direta (%Q)

Para configurar este protocolo usando Representação Direta (%Q), é necessário executar os seguintes passos:

- Configurar os parâmetros gerais do protocolo MODBUS escravo, como: tempos de comunicação, endereço e variáveis de representação direta (%Q) para receber os diagnósticos e controlar as relações.
- Adicionar e configurar relações MODBUS, especificando o tipo de dado MODBUS, variáveis de representação direta (%Q) para receber/escrever os dados e quantidade de dados a comunicar.

As descrições de cada configuração estão relacionadas a seguir, neste capítulo.

5.5.6.2.1. Parâmetros Gerais do Protocolo MODBUS Escravo – Configuração por Representação Direta (%Q)

Os parâmetros gerais, encontrados na tela inicial de configuração do protocolo MODBUS (figura abaixo), são definidos como:

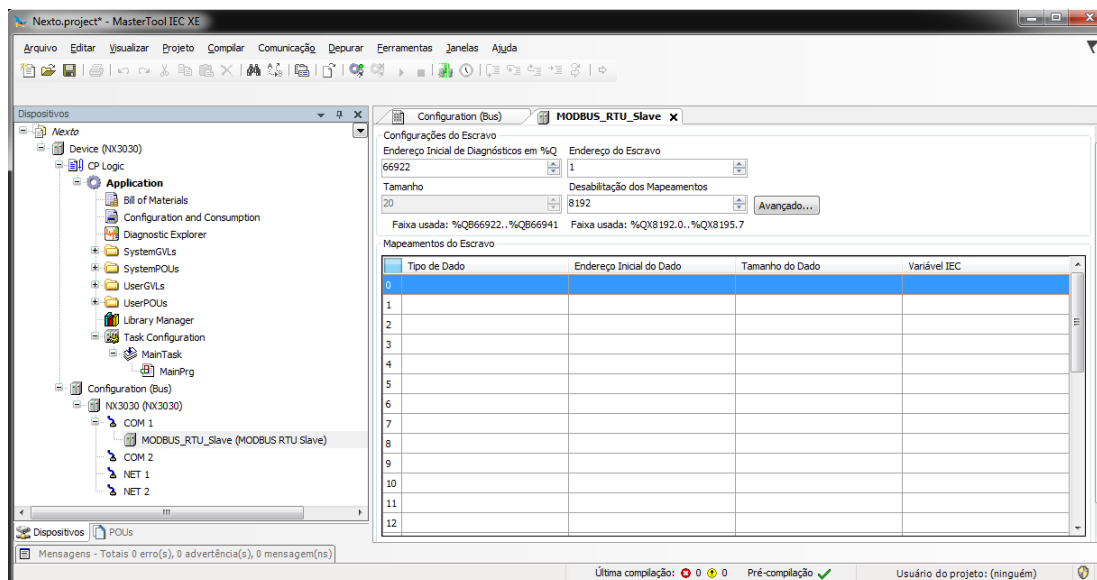


Figura 64: Tela de Configuração MODBUS RTU Escravo por Representação Direta

Endereço, Variáveis de representação direta (%Q) para controlar as relações e os diagnósticos:

Configuração	Descrição	Padrão	Opções
Endereço Inicial de Diagnósticos em %Q	Endereço inicial das variáveis de diagnóstico	-	0 a 2147483628
Tamanho	Tamanho da área de diagnósticos	-	Desabilitado para edição
Endereço do Escravo	Endereço do escravo MODBUS	1	1 a 255
Desabilitação dos Mapeamentos	Endereço inicial utilizado para desabilitar as relações MODBUS	-	0 a 2147483644

Tabela 103: Configurações das Variáveis de Representação Direta e Endereçamento

Notas:

Endereço Inicial de Diagnósticos em %Q: Esse campo é limitado pelo tamanho da memória de variáveis de saídas endereçáveis (%Q) de cada UCP, a qual pode ser consultada na seção [Memória](#).

Endereço do Escravo: É importante salientar que o escravo aceita requisições broadcast, quando o mestre envia um comando com o endereço configurado como zero. Além disso, de acordo com a Norma MODBUS, a faixa de endereços válidos para escravos é de 1 a 247, sendo os endereços 248 a 255 reservados.

Desabilitação dos Mapeamentos: Composta por 32 bits, utilizados para desabilitar, individualmente, as 32 relações MODBUS configuradas no espaço *Mapeamentos do Escravo*. A relação é desabilitada quando o bit, correspondente à relação, for igual a 1, caso contrário, o mapeamento está habilitado. Esse campo é limitado pelo tamanho da memória de variáveis de saídas endereçáveis (%Q) de cada UCP, a qual pode ser consultada na seção [Memória](#).

Padrão: O padrão de fábrica não pode ser definido para os campos *Endereço Inicial de Diagnósticos em %Q* e *Desabilitação dos Mapeamentos*, pois a criação de uma instância do protocolo pode ser realizada em qualquer momento no desenvolvimento da aplicação, fazendo com que o próprio software MasterTool IEC XE aloque um valor, da faixa de variáveis de saída de representação direta (%Q), ainda não utilizado.

O protocolo MODBUS Escravo por Representação Direta para de comunicar enquanto a UCP estiver em STOP ou parada em um breakpoint.

Os diagnósticos e comandos do protocolo MODBUS estão descritas na Tabela 100.

Os tempos de comunicação do protocolo MODBUS Escravo, encontrados no botão *Avançado...* da tela de configuração, estão descritos na Tabela 99.

5.5.6.2.2. Configuração dos Mapeamentos – Configuração por Representação Direta (%Q)

A configuração das relações MODBUS, visualizada nas figuras abaixo, segue os parâmetros descritos na tabela abaixo:

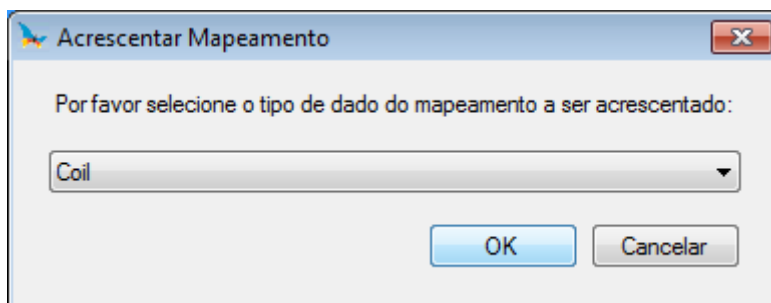


Figura 65: Adicionando Relações MODBUS

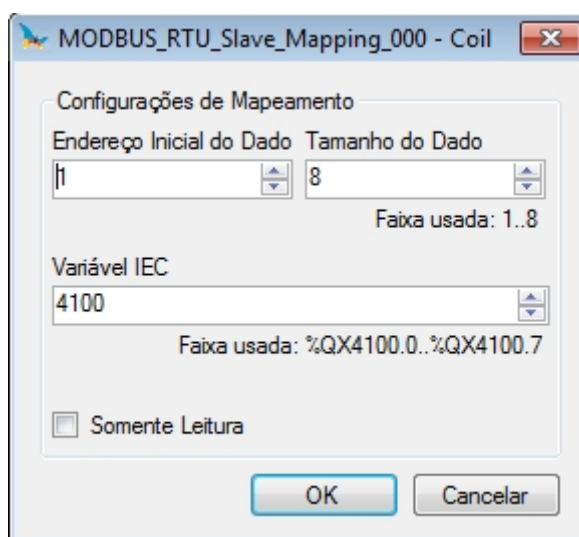


Figura 66: Configurando a Relação MODBUS

Configuração	Descrição	Padrão	Opções
Tipo de Dado	Tipo de dado MODBUS	Coil	Coil (1 bit) Holding Register (16 bits) Input Register (16 bits) Input Status (1 bit)
Endereço Inicial do Dado	Endereço inicial dos dados MODBUS	1	1 a 65536
Tamanho do Dado	Tamanho do dado MODBUS	-	1 a 65536
Variável IEC	Endereço inicial das variáveis (%Q)	-	0 a 2147483647
Somente Leitura	Somente permite a leitura	Desabilitada	Habilitada ou Desabilitada

Tabela 104: Mapeamentos do Escravo

Notas:

Opções: Os valores descritos na coluna *Opções* podem variar de acordo com o tipo de dado MODBUS configurado.

Tamanho do Dado: O valor de *Tamanho do Dado* define a quantidade máxima de dados que uma relação MODBUS poderá acessar, a partir do endereço inicial. Sendo assim, para ler uma faixa de endereços contínua, é necessário que todos os endereços estejam declarados em uma única relação. Este campo varia de acordo com o tipo de dado MODBUS configurado, ou seja, quando selecionado tipo *Coil* ou *Input Status*, o campo Tamanho do Dado deve ser um número múltiplo de oito. Também deve-se dar atenção para que o valor máximo não seja superior ao tamanho da memória de saídas endereçáveis e não sejam atribuídos os mesmos valores já utilizados durante a aplicação.

ATENÇÃO

Quando o acesso à memória de dados da comunicação for entre equipamentos com endianness diferentes (Little-Endian e Big-Endian), pode ocorrer a inversão dos dados lidos/escritos. Neste caso o usuário deve fazer o ajuste dos dados na aplicação.

Variável IEC: Caso o tipo de dado MODBUS seja *Coil* ou *Input Status* (bit), o endereço inicial das variáveis IEC terá o formato por exemplo *%QX10.1*. Porém, se o tipo de dado MODBUS for *Holding Register* ou *Input Register* (16 bits), o endereço inicial das variáveis IEC terá o formato *%QW*. Esse campo é limitado pelo tamanho da memória de variáveis de saídas endereçáveis (%Q) de cada UCP, a qual pode ser consultada na seção [Memória](#).

Somente Leitura: Quando habilitada, somente permite que o mestre da comunicação leia os dados das variáveis, não permitindo a escrita. Opção válida somente para as funções de escrita.

Padrão: O padrão de fábrica não pode ser definido para o campo *Variável IEC*, pois a criação de uma instância do protocolo pode ser realizada em qualquer momento no desenvolvimento da aplicação, fazendo com que o próprio software MasterTool IEC XE aloque um valor, da faixa de variáveis de saída de representação direta (%Q), ainda não utilizado. O padrão de fábrica não pode ser definido para o campo *Tamanho do Dado*, pois ele vai variar de acordo com o tipo de dado MODBUS selecionado.

Nas relações definidas anteriormente, o tamanho máximo de dados MODBUS pode ser 65535 (máximo valor configurado no campo *Tamanho do Dado*). Porém, a pergunta que chega no MODBUS RTU Escravo deverá endereçar um subconjunto desse mapeamento e esse grupo deve ter, no máximo, o tamanho de dados que depende do código da função, os quais estão definidos abaixo:

- Leitura de Coils (FC 1): 2000
- Leitura de Input Status (FC 2): 2000
- Leitura de Holding Registers (FC 3): 125
- Leitura de Input Registers (FC 4): 125
- Escrita de Um Coil (FC 5): 1
- Escrita de Um Holding Register (FC 6): 1
- Escrita de Múltiplos Coils (FC 15): 1968
- Escrita de Holding Registers (FC 16): 123
- Máscara de Escrita do Register (FC 22): 1
- Leitura/Escrita de Holding Registers (FC 23):
 - Leitura: 121
 - Escrita: 121

ATENÇÃO

Diferentemente de outras tarefas de uma aplicação, quando for atingida uma marca de depuração na MainTask, a tarefa de uma instância MODBUS RTU Escravo, e qualquer outra tarefa MODBUS, irá parar de ser executada no momento em que tentar efetuar uma escrita em uma área de memória. Isto ocorre para manter a consistência dos dados das áreas de memória enquanto a MainTask não estiver em execução.

5.5.7. MODBUS Ethernet

A rede de comunicação multimestre permite que as UCPs Nexto leiam ou escrevam variáveis MODBUS em outros controladores ou IHMs compatíveis com os protocolos MODBUS TCP ou MODBUS RTU via TCP. A UCP Nexto pode, simultaneamente, ser cliente e servidor em uma mesma rede de comunicação, ou até mesmo ter mais instâncias associadas à interface Ethernet, indiferente se as mesmas são MODBUS TCP ou MODBUS RTU via TCP, conforme descreve a Tabela 67.

A figura abaixo representa algumas das possibilidades de comunicação utilizando-se o protocolo MODBUS TCP simultaneamente com o protocolo MODBUS RTU via TCP.

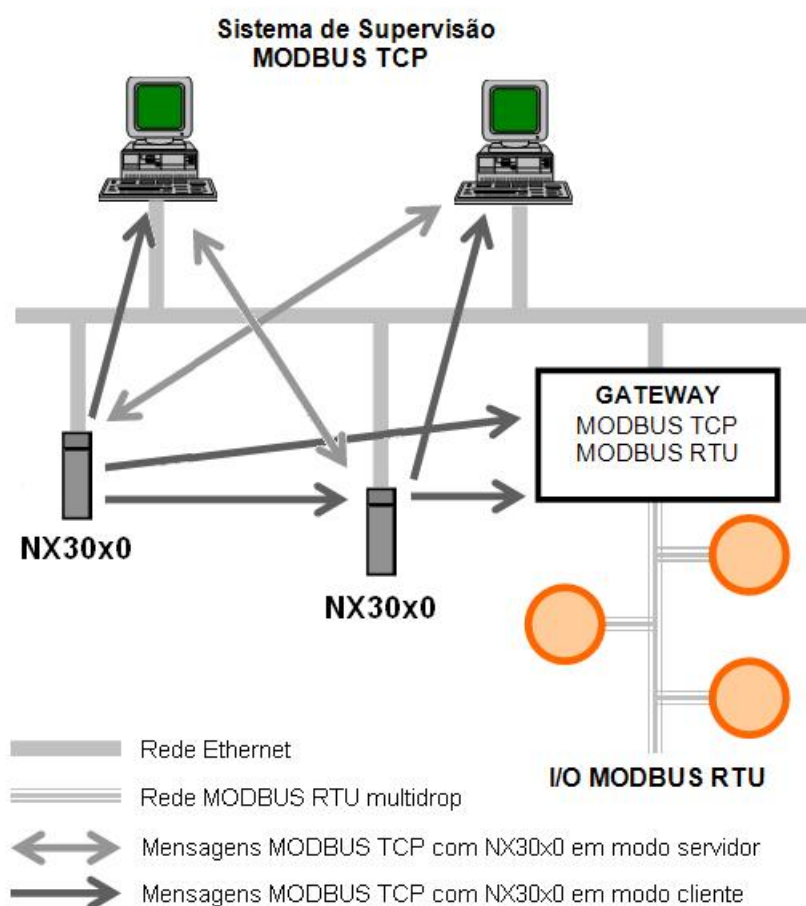


Figura 67: Rede de Comunicação MODBUS TCP

A associação de variáveis MODBUS com variáveis simbólicas da UCP é realizada pelo usuário através da definição de relações via configurador MasterTool IEC XE. Podem ser definidas até 32 relações para o modo servidor e até 128 relações para o modo cliente. Uma relação, em modo servidor, pode definir uma grande área de dados MODBUS e torná-la disponível para vários clientes. As relações em modo cliente, por outro lado, devem respeitar o tamanho máximo de dados de uma função MODBUS: 125 registradores (input registers ou holding registers) ou 2000 bits (coils ou input status). Essas informações são detalhadas na descrição de cada protocolo.

Todas as relações, em modo cliente ou servidor, podem ser desabilitadas através de variáveis de representação direta (%Q) identificadas como Desabilitação dos Mapeamentos pelo MasterTool IEC XE. A desabilitação pode ocorrer através de bits

gerais, os quais afetam todas as relações de um modo de operação, ou através de bits específicos, afetando relações específicas.

Para as relações em modo servidor, podem ser definidos conjuntos de endereços IPs com permissão de escrita e leitura, chamados de filtros. Isto é feito através da definição de um endereço de rede IP e de uma máscara de subrede, resultando em um grupo de IPs clientes que podem escrever e ler nas variáveis da relação. Funções de leitura/escrita são filtradas da mesma forma que as funções exclusivas de leitura ou escrita. Essas informações são detalhadas na descrição do protocolo MODBUS Ethernet Servidor.

Quando o protocolo MODBUS TCP é utilizado no modo cliente, pode-se usufruir da característica de múltiplas requisições, utilizando a mesma conexão TCP para acelerar a comunicação com os servidores. Quando esta característica não for desejada ou não for suportada pelo servidor, ela pode ser desabilitada (ação em nível de relação). É importante destacar que o número máximo de conexões TCP entre cliente e servidor é 63, sendo que se alguns parâmetros forem alterados, comunicações inativas podem ser fechadas, possibilitando a abertura de novas conexões.

As tabelas abaixo trazem, respectivamente, a lista completa dos tipos de dados e funções MODBUS suportadas pelas UCPs Nexto.

Tipo de Dado	Tamanho [bits]	Descrição
Coil	1	Saída digital que pode ser lida ou escrita.
Input Status	1	Entrada digital que pode ser apenas lida.
Holding Register	16	Saída analógica que pode ser lida ou escrita.
Input Register	16	Entrada analógica que pode ser apenas lida.

Tabela 105: Tipos de Dados MODBUS suportados pelas UCPs Nexto

Código		Descrição
DEC	HEX	
1	0x01	Leitura de coils (FC 01)
2	0x02	Leitura de input status (FC 02)
3	0x03	Leitura de holding registers (FC 03)
4	0x04	Leitura de input registers (FC 04)
5	0x05	Escrita de um coil (FC 05)
6	0x06	Escrita de um holding register (FC 06)
15	0x0F	Escrita de múltiplos coils (FC 15)
16	0x10	Escrita de múltiplos holding registers (FC 16)
22	0x16	Escrita mascarada de um holding register (FC 22)
23	0x17	Leitura/escrita de múltiplos holding registers (FC 23)

Tabela 106: Funções MODBUS Suportadas pelas UCPs Nexto

Independente do modo de configuração, os passos para inserir uma instância do protocolo e configurar a interface ethernet são iguais. As demais etapas de configuração serão descritas a seguir para cada modalidade.

- Adicionar uma ou mais instâncias do protocolo MODBUS Ethernet Cliente ou Servidor ao canal Ethernet. Para realizar esse procedimento, consultar a seção [Inserindo uma Instância de Protocolo](#).
- Configurar a interface Ethernet. Para realizar esse procedimento, consultar a seção [Configuração das Interfaces Ethernet](#).

5.5.8. MODBUS Ethernet Cliente

Este protocolo está disponível para as UCPs da Série Nexto nos seus canais Ethernet. Ao selecionar esta opção no MasterTool IEC XE, a UCP passa a ser cliente da comunicação MODBUS, possibilitando o acesso a outros dispositivos com o mesmo protocolo, quando esta estiver em modo de execução (Modo *Run*).

Há dois modos de configuração para este protocolo. Um deles faz uso de *Representação Direta (%Q)*, no qual as variáveis são definidas pelo seu endereço. O outro, chamado *Mapeamento Simbólico*, tem as variáveis definidas pelo seu nome.

O procedimento para inserir uma instância de protocolo é encontrado com detalhes no Manual de Utilização do MasterTool IEC XE – MU299048 ou na seção [Inserindo uma Instância de Protocolo](#).

5.5.8.1. Configuração do Protocolo MODBUS Ethernet Cliente por Mapeamento Simbólico

Para configurar este protocolo usando *Mapeamento Simbólico*, é necessário executar os seguintes passos:

- Configurar os parâmetros gerais do protocolo MODBUS Cliente, com o protocolo TCP ou RTU via TCP.
- Adicionar e configurar dispositivos, definindo endereço IP, porta, endereço do escravo e time-out de comunicação (disponível no botão de configurações avançadas do Dispositivo).
- Adicionar e configurar os mapeamentos MODBUS, especificando o nome da variável, tipo de dados, endereço inicial do dado, tamanho do dado e variável que receberá os dados de qualidade.
- Adicionar e configurar as requisições MODBUS, especificando a função desejada, o tempo de varredura da requisição, o endereço inicial (leitura/escrita), o tamanho dos dados (Leitura/Escrita), a variável que receberá os dados de qualidade, e a variável responsável por desabilitar a requisição.

5.5.8.1.1. Parâmetros Gerais do Protocolo MODBUS Cliente – Configuração por Mapeamento Simbólico

Os parâmetros gerais, encontrados na tela inicial de configuração do protocolo MODBUS (figura abaixo), são definidos como:

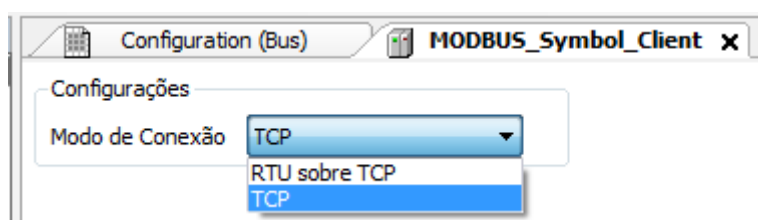


Figura 68: Tela de Configuração Parâmetros Gerais MODBUS Cliente

Configuração	Descrição	Padrão	Opções
Modo de Conexão	Seleção do protocolo	TCP	RTU via TCP TCP

Tabela 107: Configurações Gerais MODBUS Cliente

Os diagnósticos e comandos do protocolo MODBUS Cliente configurado, seja por mapeamento simbólico ou por representação direta, são armazenados em variáveis do tipo *T_DIAG_MODBUS_ETH_CLIENT_1* e ainda para o mapeamento por representação direta estão em 4 bytes e 8 words, os quais estão descritos na tabela abaixo (n é o valor configurado no campo *Endereço Inicial de Diagnósticos em %Q*):

Variável de Representação Direta	Variável de diagnóstico do tipo T_DIAG_MODBUS_ETH_CLIENT_1.*	Tamanho	Descrição
Bits de diagnóstico:			
%QX(n).0	tDiag. bRunning	BIT	O cliente está em execução.
%QX(n).1	tDiag. bNotRunning	BIT	O cliente não está em execução (ver bit bInterruptedByCommand).

Variável de Representação Direta	Variável de diagnóstico do tipo T_DIAG_MODBUS_ETH_CLIENT_1.*	Tamanho	Descrição
%QX(n).2	tDiag. bInterruptedByCommand	BIT	O bit bNotRunning foi habilitado pois o cliente foi interrompido pelo usuário através de bits de comando.
%QX(n).3	tDiag. bConfigFailure	BIT	Diagnóstico descontinuado.
%QX(n).4	tDiag. bRXFailure	BIT	Diagnóstico descontinuado.
%QX(n).5	tDiag. bTXFailure	BIT	Diagnóstico descontinuado.
%QX(n).6	tDiag. bModuleFailure	BIT	Indica se há falha no módulo ou o módulo não está presente.
%QX(n).7	tDiag. bAllDevicesCommFailure	BIT	Indica que todos os dispositivos configurados no cliente estão com falha.
%QB(n+1)	byDiag_1_reserved	BYTE	Reservado
Bits de comando, reiniciados automaticamente:			
%QX(n+2).0	tCommand. bStop	BIT	Parar o cliente.
%QX(n+2).1	tCommand. bRestart	BIT	Reiniciar o cliente.
%QX(n+2).2	tCommand. bResetCounter	BIT	Reiniciar as estatísticas dos diagnósticos (contadores).
%QX(n+2).3	tCommand. bDiag_19_reserved	BIT	Reservado
%QX(n+2).4	tCommand. bDiag_20_reserved	BIT	Reservado
%QX(n+2).5	tCommand. bDiag_21_reserved	BIT	Reservado
%QX(n+2).6	tCommand. bDiag_22_reserved	BIT	Reservado
%QX(n+2).7	tCommand. bDiag_23_reserved	BIT	Reservado
%QB(n+3)	byDiag_3_reserved	BYTE	Reservado
Estatísticas de comunicação:			
%QW(n+4)	tStat. wTXRequests	WORD	Contador de requisições transmitidas pelo cliente (0 a 65535).
%QW(n+6)	tStat. wRXNormalResponses	WORD	Contador de respostas normais recebidas pelo cliente (0 a 65535).
%QW(n+8)	tStat. wRXExceptionResponses	WORD	Contador de respostas com códigos de exceção (0 a 65535).
%QW(n+10)	tStat. wRXIllegalResponses	WORD	Contador de respostas ilegais recebidas pelo cliente – sintaxe inválida, CRC inválido ou número insuficiente de bytes recebidos (0 a 65535).
%QW(n+12)	tStat. wDiag_12_reserved	WORD	Reservado

Variável de Representação Direta	Variável de diagnóstico do tipo T_DIAG_MODBUS_ETH_CLIENT_1.*	Tamanho	Descrição
%QW(n+14)	tStat. wDiag_14_reserved	WORD	Reservado
%QW(n+16)	tStat. wDiag_16_reserved	WORD	Reservado
%QW(n+18)	tStat. wDiag_18_Reserved	WORD	Reservado

Tabela 108: Diagnósticos MODBUS Cliente

Nota:

Contadores: Todos os contadores dos diagnósticos do MODBUS TCP Cliente retornam à zero quando o valor limite 65535 é ultrapassado.

5.5.8.1.2. Configuração dos Dispositivos – Configuração por Mapeamento Simbólico

A configuração dos dispositivos, visualizada na figura abaixo, segue os seguintes parâmetros:

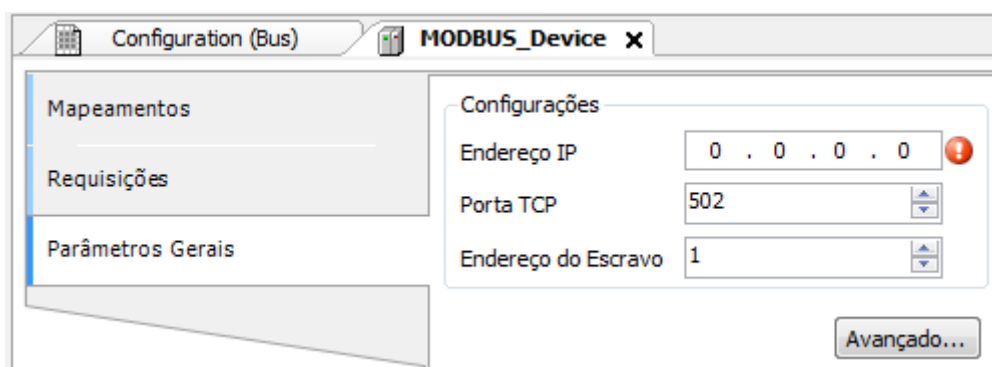


Figura 69: Tela de Configurações dos Parâmetros Gerais do Dispositivo

Configuração	Descrição	Padrão	Opções
Endereço IP	Endereço IP do servidor	0.0.0.0	1.0.0.1 a 223.255.255.255
Porta TCP	Porta TCP	502	2 a 65534
Endereço do Escravo	Endereço do Escravo MODBUS	1	0 a 255

Tabela 109: Parâmetros Gerais MODBUS Cliente

Notas:

Endereço IP: Endereço IP do dispositivo servidor MODBUS.

Porta TCP: Caso sejam adicionadas várias instâncias do protocolo em uma única interface Ethernet, diferentes portas TCP devem ser selecionadas para cada instância. Algumas portas TCP, entre as possibilidades mencionadas acima, são reservadas e, portanto, não podem ser utilizadas. Ver tabela [Portas TCP/UDP reservadas](#).

Endereço do Escravo: De acordo com a norma MODBUS, a faixa de endereços válidos para escravos é de 0 a 247, sendo os endereços 248 a 255 reservados. Quando o mestre envia um comando de escrita com o endereço configurado como zero, ele está realizando requisições broadcast na rede.

Os parâmetros nas configurações avançadas do dispositivo MODBUS Cliente, encontrados no botão *Avançado...* na aba de *Parâmetros Gerais*, estão divididos em: *Número Máximo de Requisições Simultâneas*, *Time-out de Comunicação*, *Modo de time-out da conexão* e *Tempo de Inatividade*.

Configuração	Descrição	Padrão	Opções
Número Máximo de Requisições Simultâneas	Número de requisições simultâneas que o cliente pode fazer ao servidor	1	1 a 8
Time-out de Comunicação	Time-out do nível da aplicação em ms	3000	10 a 65535
Modo	Define quando a conexão com o servidor é finalizada pelo cliente	Conexão é fechada após um tempo de inatividade de (s): 10 a 3600.	Conexão é fechada depois de um time-out. Conexão é fechada ao final de cada comunicação. Conexão é fechada após um tempo de inatividade de (s): 10 a 3600.
Tempo de Inatividade	Tempo de inatividade	10	3600

Tabela 110: Configurações Avançadas MODBUS Cliente

Notas:

Número Máximo de Requisições Simultâneas: É utilizado em servidores com um alto ciclo de varredura. Esse parâmetro é fixado em 1 (não editável), quando o protocolo configurado é MODBUS RTU via TCP.

Time-out de Comunicação: O time-out da comunicação é o tempo que o cliente aguardará por uma resposta do servidor à requisição. Para um dispositivo MODBUS Cliente, duas variáveis do sistema devem ser consideradas: o tempo que o servidor leva para processar a requisição e o atraso de envio da resposta caso seja configurado no servidor. É recomendado que o time-out seja igual ou maior que duas vezes a soma destes parâmetros. Para mais informações, ver seção [Desempenho de Comunicação](#).

Modo: Define quando a conexão com o servidor é finalizada pelo cliente. Seguem as opções:

- Conexão é fechada depois de um time-out ou Conexão nunca é fechada em situações normais: Estas opções apresentam o mesmo comportamento do Cliente fechar a conexão devido ao fato do Servidor não ter respondido a uma requisição antes do Time-out de Comunicação ter se esgotado.
- Conexão é fechada ao final de cada comunicação: A conexão é fechada pelo Cliente após concluir cada requisição.
- Conexão é fechada após um tempo de inatividade: A conexão será fechada pelo Cliente caso ele fique por um tempo igual ao Tempo de Inatividade sem realizar requisição para o Servidor.

Tempo de Inatividade: Tempo de inatividade da conexão.

5.5.8.1.3. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico

A configuração dos mapeamentos MODBUS, visualizada na figura abaixo, segue os parâmetros descritos na tabela abaixo:

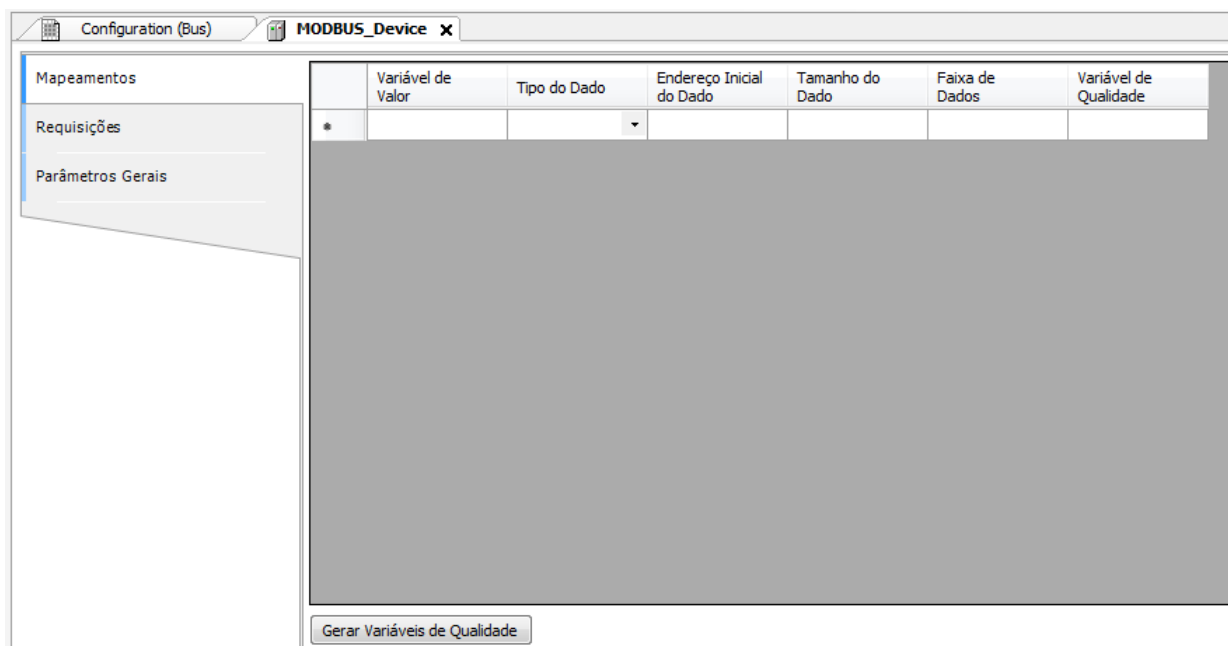


Figura 70: Tela de Mapeamentos de dados MODBUS Cliente

Configuração	Descrição	Padrão	Opções
Variável de Valor	Nome da variável simbólica	-	Nome de uma variável declarada em um programa ou GVL
Tipo do Dado	Tipo de dado MODBUS	-	Coil - Escrita (1 bit) Coil - Leitura (1 bit) Holding Register - Escrita (16 bits) Holding Register - Leitura (16 bits) Holding Register – Máscara AND (16 bits) Holding Register – Máscara OR (16 bits) Input Register (16 bits) Input Status (1 bit)
Endereço Inicial do Dado	Endereço inicial dos dados MODBUS	-	1 a 65536
Tamanho do Dado	Tamanho do dado MODBUS	-	1 a 65536
Faixa de Dados	Faixa de endereços do dado configurado	-	-

Tabela 111: Configuração dos Mapeamentos MODBUS

Notas:

Variável de Valor: Esse campo é utilizado para especificar uma variável simbólica na relação MODBUS.

Tipo de Dado: Esse campo é utilizado para especificar o tipo de dado utilizado na relação MODBUS.

Tipo de Dado	Tamanho [bits]	Descrição
Coil - Escrita	1	Saída digital de escrita.
Coil - Leitura	1	Saída digital de leitura.
Holding Register - Escrita	16	Saída analógica de escrita.
Holding Register - Leitura	16	Saída analógica de leitura.
Holding Register - Máscara AND	16	Saída analógica que pode ser lida ou escrita com máscara AND.
Holding Register - Máscara OR	16	Saída analógica que pode ser lida ou escrita com máscara OR.
Input Register	16	Entrada analógica que pode ser apenas lida.
Input Status	1	Entrada digital que pode ser apenas lida.

Tabela 112: Tipos de Dados suportados no MODBUS

Endereço Inicial do Dado: Endereço inicial do dado de um mapeamento MODBUS.

Tamanho do Dado: O valor de Tamanho especifica a quantidade máxima de dados que uma relação MODBUS poderá acessar, a partir do endereço inicial. Sendo assim, para ler uma faixa de endereços contínua, é necessário que todos os endereços estejam declarados em uma única relação. Este campo varia de acordo com o tipo de dado MODBUS configurado.

Faixa de Dados: Este campo mostra ao usuário a faixa de endereços de memória utilizada pela relação MODBUS.

5.5.8.1.4. Configuração das Requisições – Configuração por Mapeamento Simbólico

A configuração das requisições MODBUS, visualizada na figura abaixo, segue os parâmetros descritos na tabela abaixo:

The screenshot shows the 'MODBUS_Device' configuration window. On the left, there is a sidebar with 'Mapeamentos', 'Requisições', and 'Parâmetros Gerais'. The main area displays a table for configuring requests. The table has 10 columns: 'Código de Função', 'Varredura (ms)', 'Endereço Inicial do Dado de Leitura', 'Tamanho dos Dados de Leitura', 'Faixa dos Dados de Leitura', 'Endereço Inicial do Dado de Escrita', 'Tamanho dos Dados de Escrita', 'Faixa dos Dados de Escrita', 'Variável de Diagnóstico', and 'Variável de Desabilitação'. Below the table, there are three buttons: 'Gerar Variáveis de Diagnósticos', 'Gerar Variáveis de Desabilitação', and a text field for 'Tipo da Variável de Diagnósticos' containing the value 'NXMODBUS_DIAGNOSTIC_STRUCTS.T_DIAG_MODBUS_ETH_MAPPING_1'.

Figura 71: Tela de Requisições de dados MODBUS

Configuração	Descrição	Padrão	Opções
Código de Função	Tipo de função MODBUS	-	01 – Leitura de Coils 02 – Leitura de Input Status 03 – Leitura de Holding Registers 04 – Leitura de Input Registers 05 – Escrita de Um Coil 06 – Escrita de Um Register 15 – Escrita de Múltiplos Coils 16 – Escrita de Múltiplos Registers 22 – Escrita Mascarada de Register 23 – Leitura/Escrita de Múltiplos Registers
Varredura (ms)	Período de comunicação (ms)	100	0 a 3600000
Endereço Inicial do Dado de Leitura	Endereço inicial dos dados de leitura MODBUS	-	1 a 65536
Tamanho dos Dados de Leitura	Tamanho dos dados de leitura MODBUS	-	Depende da função utilizada
Faixa dos Dados de Leitura	Faixa de endereço dos dados de leitura MODBUS	-	0 a 2147483646
Endereço Inicial do Dado de Escrita	Endereço inicial dos dados de escrita MODBUS	-	1 a 65536
Tamanho dos Dados de Escrita	Tamanho dos dados de escrita MODBUS	-	Depende da função utilizada
Faixa dos Dados de Escrita	Faixa de endereço dos dados de escrita MODBUS	-	0 a 2147483647
Variável de Diagnóstico	Nome da variável de diagnóstico	-	Nome de uma variável declarada em um programa ou GVL
Variável de Desabilitação	Variável utilizada para desabilitar a relação MODBUS	-	Campo destinado a variável simbólica utilizada para desabilitar, individualmente, as requisições MODBUS configuradas. Esta variável deve ser do tipo BOOL. A variável pode ser simples ou elemento de array e pode estar em estruturas.

Tabela 113: Configuração das Relações MODBUS

Notas:

Configuração: O número de configurações, padrão de fábrica e os valores da coluna opções, podem variar de acordo com o tipo de dado e função MODBUS (FC).

Código de Função: As funções MODBUS (FC) disponíveis são as seguintes:

Código		Descrição
DEC	HEX	
1	0x01	Leitura de coils (FC 01)
2	0x02	Leitura de input status (FC 02)
3	0x03	Leitura de holding registers (FC 03)
4	0x04	Leitura de input registers (FC 04)
5	0x05	Escrita de um coil (FC 05)
6	0x06	Escrita de um holding register (FC 06)
15	0x0F	Escrita de múltiplos coils (FC 15)
16	0x10	Escrita de múltiplos holding registers (FC 16)
22	0x16	Escrita mascarada de um holding register (FC 22)
23	0x17	Leitura/escrita de múltiplos holding registers (FC 23)

Tabela 114: Funções MODBUS Suportadas pelas UCPS Nexto

Varredura: Este parâmetro indica com que frequência a comunicação definida por esta requisição deve ser executada. Ao ser finalizada uma comunicação será aguardado um tempo igual ao configurado no campo varredura e, após, será executada uma nova comunicação.

Endereço Inicial dos Dados de Leitura: Campo destinado ao endereço inicial dos dados de leitura MODBUS.

Tamanho dos Dados de Leitura: O valor mínimo para o tamanho dos dados de leitura é 1 e o valor máximo depende da função MODBUS (FC) utilizada, conforme abaixo:

- Leitura de Coils (FC 1): 2000
- Leitura de Input Status (FC 2): 2000
- Leitura de Holding Registers (FC 3): 125
- Leitura de Input Registers (FC 4): 125
- Leitura/Escrita de Holding Registers (FC 23): 121

Faixa dos Dados de Leitura: Este campo mostra a faixa de dados de leitura MODBUS configurada para cada requisição. O endereço inicial de leitura, somado ao tamanho do dado de leitura resultará na faixa de dados de leitura de cada uma das requisições.

Endereço Inicial dos Dados de Escrita: Campo destinado ao endereço inicial dos dados de escrita MODBUS.

Tamanho dos Dados de Escrita: O valor mínimo para o tamanho dos dados de escrita é 1 e o valor máximo depende da função MODBUS (FC) utilizada, conforme abaixo:

- Escrita de Um Coil (FC 5): 1
- Escrita de Um Holding Register (FC 6): 1
- Escrita de Múltiplos Coils (FC 15): 1968
- Escrita de Holding Registers (FC 16): 123
- Máscara de Escrita do Register (FC 22): 1
- Leitura/Escrita de Holding Registers (FC 23): 121

Faixa dos Dados de Escrita: Este campo mostra a faixa de dados de escrita MODBUS configurada para cada requisição. O endereço inicial de escrita, somado ao tamanho do dado de escrita resultará na faixa de dados de escrita de cada uma das requisições.

Variável de Diagnóstico: Os diagnósticos da requisição MODBUS configurada, seja por mapeamento simbólico ou por representação direta, são armazenados em variáveis do tipo *T_DIAG_MODBUS_RTU_MAPPING_1* para dispositivos Mestre e *T_DIAG_MODBUS_ETH_CLIENT_1* para dispositivos Cliente e para o mapeamento por representação direta estão em 4 bytes e 2 words, os quais estão descritos na Tabela 93 ("n" é o valor configurado no campo *Endereço Inicial de Diagnósticos em %Q*).

Variável de Representação Direta	Variável de diagnóstico do tipo T_DIAG_MODBUS_ETH_MAPPING_1.*	Tamanho	Descrição
Bits de estado da comunicação:			
%QX(n).0	byStatus. bCommIdle	BIT	Comunicação inativa (aguardando ser constada).
%QX(n).1	byStatus. bCommExecuting	BIT	Comunicação ativa.
%QX(n).2	byStatus. bCommPostponed	BIT	Comunicação adiada, pois o número máximo de requisições simultâneas foi atingido. As comunicações adiadas serão executadas na mesma sequência em que elas foram requisitadas, evitando a indeterminação. O tempo gasto nesse estado não é contabilizado para efeitos de time-out. Os bits bCommIdle e bCommExecuting são falsos quando o bit bCommPostponed é verdadeiro.
%QX(n).3	byStatus. bCommDisabled	BIT	Comunicação desabilitada. O bit bCommIdle é reiniciado nessa condição.
%QX(n).4	byStatus. bCommOk	BIT	Comunicação finalizada anteriormente foi realizada com sucesso.
%QX(n).5	byStatus. bCommError	BIT	Comunicação finalizada anteriormente teve um erro. Verificar código de erro.
%QX(n).6	byStatus. bCommAborted	BIT	Comunicação finalizada anteriormente foi interrompida devido à falha de conexão.
%QX(n).7	byStatus. bDiag_7_reserved	BIT	Reservado
Último código de erro (habilitado quando o bCommError = verdadeiro):			
%QB(n+1)	eLastErrorCode	MASTER_ERROR_CODE (BYTE)	Informa a possível causa do último erro ocorrido na relação MODBUS. Consulte a Tabela 116 para detalhes das possibilidades.
Último código de exceção recebido pelo cliente:			
%QB(n+2)	eLastExceptionCode	MODBUS_EXCEPTION (BYTE)	NO_EXCEPTION (0) FUNCTION_NOT_SUPPORTED (1) MAPPING_NOT_FOUND (2) ILLEGAL_VALUE (3) ACCESS_DENIED (128)* MAPPING_DISABLED (129)* IGNORE_FRAME (255)*
Estatísticas de comunicação:			
%QB(n+3)	byDiag_3_reserved	BYTE	Reservado.
%QW(n+4)	wCommCounter	WORD	Contador de comunicações finalizadas, com ou sem erros. O usuário pode testar quando a comunicação foi finalizada testando a variação desse contador. Quando o valor 65535 é atingido, o contador retorna à zero.
%QW(n+6)	wCommErrorCounter	WORD	Contador de comunicações finalizadas com erros. Quando o valor 65535 é atingido, o contador retorna a zero.

Tabela 115: Diagnósticos das relações MODBUS Cliente

Notas:

Códigos de exceção: Os códigos de exceção apresentados neste campo são os valores retornados pelo servidor. As definições dos códigos de exceção 128, 129 e 255, apresentadas nessa tabela, são válidas apenas na utilização de escravos Altus. Para escravos de outros fabricantes esses códigos de exceção podem ter significados diferentes.

Variável de Desabilitação: Campo destinado à variável do tipo booleana utilizada para desabilitar, individualmente, as requisições MODBUS configuradas na aba Requisições através do botão na parte inferior da janela. A requisição é desabilitada quando a variável, correspondente a requisição, for igual a 1, caso contrário, a requisição está habilitada.

Último código de Erro: Os códigos das possíveis situações que ocasionam erro na comunicação MODBUS podem ser consultados abaixo:

Código	Enumerável	Descrição
1	ERR_EXCEPTION	Resposta reportada em um código de exceção (ver eLastExceptionCode = Código de Exceção).
2	ERR_CRC	Resposta com CRC inválido.
3	ERR_ADDRESS	Endereço MODBUS não encontrado. O endereço que respondeu à requisição foi diferente do esperado.
4	ERR_FUNCTION	Código inválido da função. A função recebida na resposta foi diferente da esperada pela requisição.
5	ERR_FRAME_DATA_COUNT	A quantidade de dados da resposta foi diferente da esperada.
7	ERR_NOT_ECHO	Resposta não é eco da pergunta (FC 5 e 6).
8	ERR_REFERENCE_NUMBER	Número de referência inválido (FC 15 e 16).
9	ERR_INVALID_FRAME_SIZE	Resposta menor do que a esperada.
20	ERR_CONNECTION	Erro durante a fase de conexão.
21	ERR_SEND	Erro durante a fase de transmissão.
22	ERR_RECEIVE	Erro durante a fase de recepção.
40	ERR_CONNECTION_TIMEOUT	Time-out no nível de aplicação durante a conexão.
41	ERR_SEND_TIMEOUT	Time-out no nível de aplicação durante a transmissão.
42	ERR_RECEIVE_TIMEOUT	Time-out no nível de aplicação enquanto aguarda resposta.
43	ERR_CTS_OFF_TIMEOUT	Time-out enquanto aguarda CTS = falso na transmissão.
44	ERR_CTS_ON_TIMEOUT	Time-out enquanto aguarda CTS = verdadeiro na transmissão.
128	NO_ERROR	Sem erro desde a inicialização.

Tabela 116: Códigos de Erro das relações MODBUS

ATENÇÃO

Diferentemente de outras tarefas de uma aplicação, quando for atingida uma marca de depuração na MainTask, a tarefa de uma instância MODBUS Ethernet Cliente, e qualquer outra tarefa MODBUS, irá parar de ser executada no momento em que tentar efetuar uma escrita em uma área de memória. Isto ocorre para manter a consistência dos dados das áreas de memória enquanto a MainTask não estiver em execução.

5.5.8.2. Configuração do Protocolo MODBUS Ethernet Cliente por Representação Direta (%Q)

Para configurar este protocolo usando representação direta (%Q), é necessário executar os seguintes passos:

- Configurar os parâmetros gerais do protocolo MODBUS, como: tempos de comunicação e variáveis de representação direta (%Q) para receber os diagnósticos.
- Adicionar e configurar dispositivos, definindo endereço, variáveis de representação direta (%Q) para desabilitar as relações, time-outs de comunicação, etc.

5. CONFIGURAÇÃO

- Adicionar e configurar relações MODBUS, especificando o tipo de dado e função MODBUS, time-outs, variáveis de representação direta (%Q) para receber os diagnósticos da relação e outras para receber/escrever os dados, quantidade de dados a comunicar e varredura da relação.

As descrições de cada configuração estão relacionadas a seguir, neste capítulo.

5.5.8.2.1. Parâmetros Gerais do Protocolo MODBUS Cliente – Configuração por Representação Direta (%Q)

Os parâmetros gerais, encontrados na tela inicial de configuração do protocolo MODBUS (figura abaixo), são definidos como:

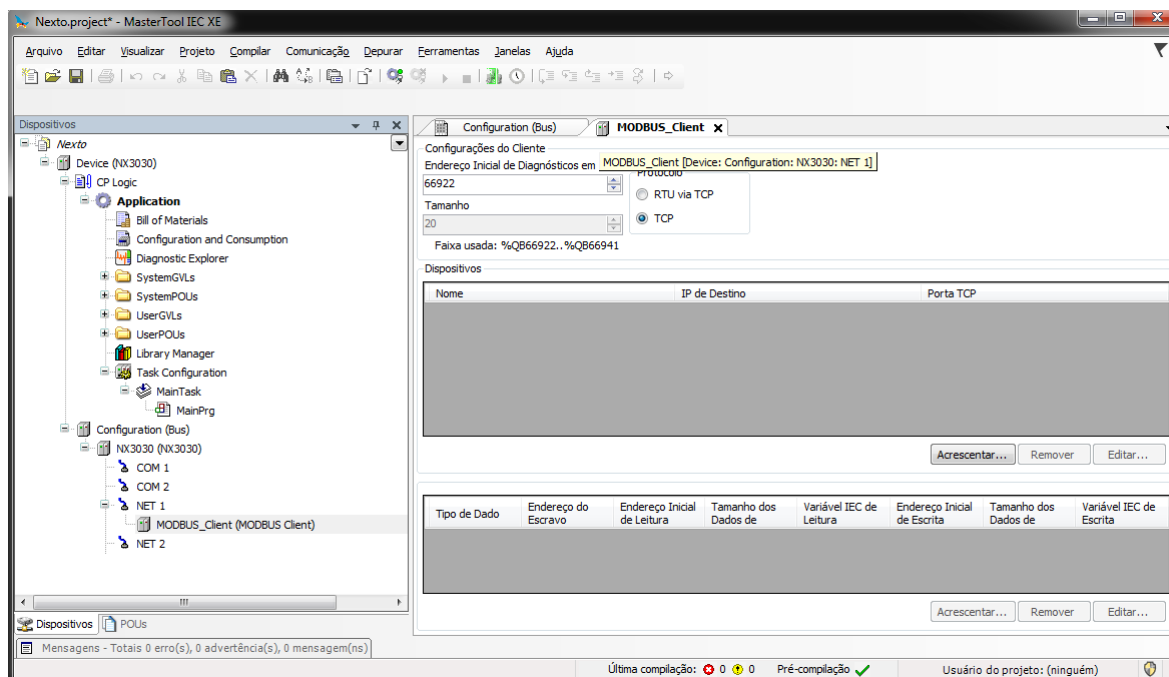


Figura 72: Tela de Configuração – MODBUS Cliente

Seleção do protocolo e variáveis de representação direta (%Q) para os diagnósticos:

Configuração	Descrição	Padrão	Opções
Endereço Inicial de Diagnósticos em %Q	Endereço inicial das variáveis de diagnóstico	-	0 a 2147483628
Tamanho	Tamanho da área de diagnósticos	20	Desabilitado para edição
Protocolo	Seleção do protocolo	TCP	RTU via TCP TCP

Tabela 117: Configurações do MODBUS Cliente

Notas:

Endereço Inicial de Diagnósticos em %Q: Esse campo é limitado pelo tamanho da memória de variáveis de saídas endereçáveis (%Q) de cada UCP, a qual pode ser consultada na seção [Memória](#).

Padrão: O padrão de fábrica não pode ser definido para o campo *Endereço Inicial de Diagnósticos em %Q*, pois a criação de uma instância do protocolo pode ser realizada em qualquer momento no desenvolvimento da aplicação, fazendo com que o próprio software MasterTool IEC XE aloque um valor, da faixa de variáveis de saída de representação direta (%Q), ainda não utilizado.

Os diagnósticos e comandos do protocolo MODBUS estão descritos na Tabela [108](#).

5.5.8.2.2. Configuração dos Dispositivos – configuração por Representação Direta (%Q)

A configuração dos dispositivos, visualizada na figura abaixo, segue os seguintes parâmetros:

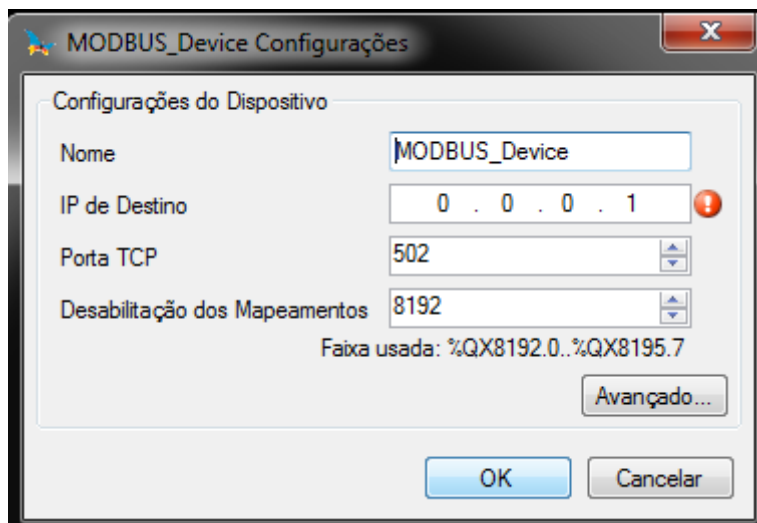


Figura 73: Configurando Cliente MODBUS

Configuração	Descrição	Padrão	Opções
Nome da Instância	Nome da instância	MODBUS_Device	Identificador, segundo a IEC 61131-3
IP de Destino	Endereço IP do servidor	0. 0. 0. 1	1.0.0.1 a 223.255.255.255
Porta TCP	Porta TCP	502	2 a 65534
Desabilitação dos Mapeamentos	Endereço inicial utilizado para desabilitar as relações MODBUS	-	Qualquer endereço da área de %Q, limitado conforme UCP utilizada

Tabela 118: Configurações do Dispositivo – MODBUS Cliente

Notas:

Nome da Instância: Esse campo é o identificador do dispositivo, o qual é verificado segundo a IEC 61131-3, ou seja, não permite espaços, caracteres especiais e iniciar com caractere numeral. É limitado em 24 caracteres.

Porta TCP: Caso sejam adicionadas várias instâncias do protocolo em uma única interface Ethernet, diferentes portas TCP devem ser selecionadas para cada instância. Algumas portas TCP, entre as possibilidades mencionadas acima, são reservadas e, portanto, não podem ser utilizadas. Ver tabela [Portas TCP/UDP reservadas](#).

Desabilitação dos Mapeamentos: Composta por 32 bits, utilizados para desabilitar, individualmente, as 32 relações MODBUS configuradas no espaço *Mapeamentos do Dispositivo*. A relação é desabilitada quando o bit, correspondente à relação, for igual a 1, caso contrário, o mapeamento está habilitado. Esse campo é limitado pelo tamanho da memória de variáveis de saídas endereçáveis (%Q) de cada UCP, a qual pode ser consultada na seção [Memória](#).

Padrão: O padrão de fábrica não pode ser definido para o campo *Desabilitação dos Mapeamentos*, pois a criação de uma instância do protocolo pode ser realizada em qualquer momento no desenvolvimento da aplicação, fazendo com que o próprio software MasterTool IEC XE aloque um valor, da faixa de variáveis de saída de representação direta (%Q), ainda não utilizado.

Time-out de comunicação: As configurações presentes no botão *Avançado...*, relativas à conexão TCP, estão descritas nas notas da seção [Configuração dos Dispositivos – Configuração por Mapeamento Simbólico](#).

5.5.8.2.3. Configuração dos Mapeamentos – Configuração por Representação Direta (%Q)

A configuração das relações MODBUS, visualizada nas figuras abaixo, segue os parâmetros descritos na tabela abaixo:

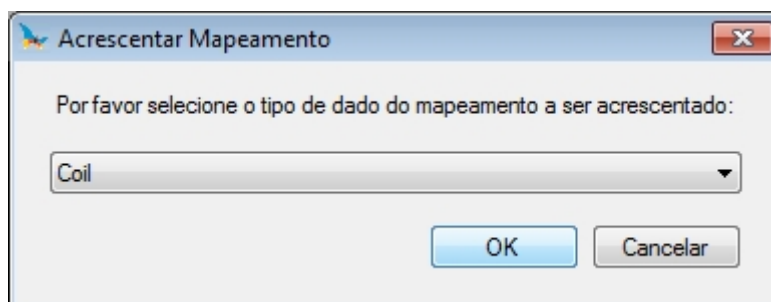


Figura 74: Tipo de Dado MODBUS

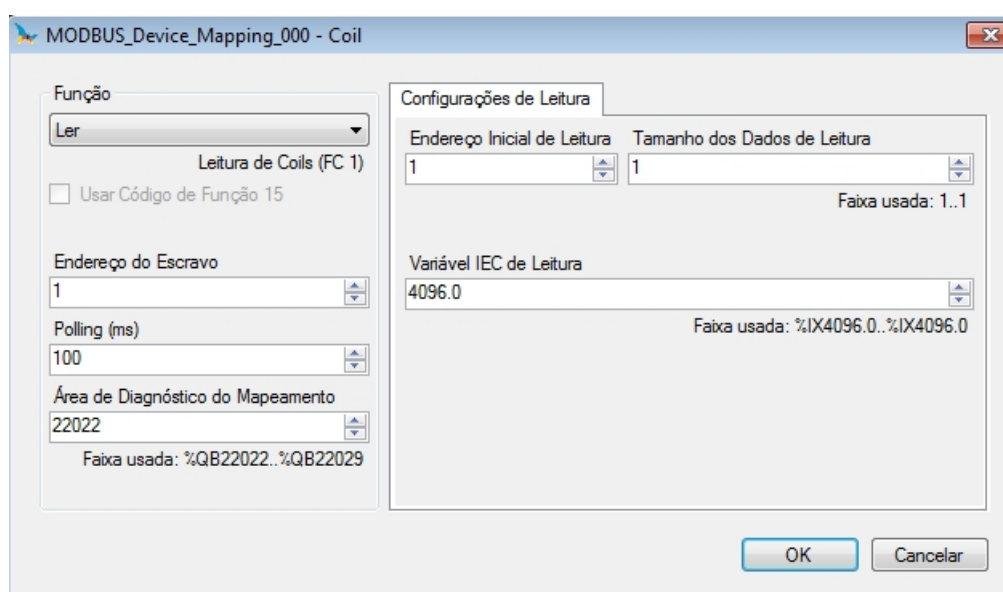


Figura 75: Função MODBUS

Na tabela abaixo, o número de configurações, padrão de fábrica e os valores da coluna opções, podem variar de acordo com o tipo de dado e função MODBUS (FC).

Configuração	Descrição	Padrão	Opções
Função	Tipo de função MODBUS	Ler	Ler Escrever Ler/Escrever Máscara de Escrita
Endereço do Escravo	Endereço do escravo MODBUS.	1	0 a 255
Polling (ms)	Período de comunicação (ms)	100	0 a 3600000
Área de Diagnóstico do Mapeamento	Endereço inicial dos diagnósticos da relação MODBUS (%Q)	-	0 a 2147483640
Endereço Inicial de Leitura	Endereço inicial dos dados de leitura MODBUS	1	1 a 65536
Tamanho dos Dados de Leitura	Número de dados de leitura MODBUS	-	Depende da função utilizada

Configuração	Descrição	Padrão	Opções
Variável IEC de Leitura	Endereço inicial das variáveis de leitura (%I)	-	0 a 2147483647
Endereço Inicial de Escrita	Endereço inicial dos dados de escrita MODBUS	1	1 a 65536
Tamanho dos Dados de Escrita	Número de dados de escrita MODBUS	-	Depende da função utilizada
Variável IEC de Escrita	Endereço inicial das variáveis de escrita (%Q)	-	0 a 2147483647
Máscara de Escrita das Variáveis IEC	Endereço inicial das variáveis para a máscara de escrita (%Q)	-	0 a 2147483644

Tabela 119: Mapeamentos do Dispositivo

Notas:

Tabela de Mapeamentos do Dispositivo: O número de configurações e os valores descritos na coluna Opções podem variar de acordo com o tipo de dado e função MODBUS.

Endereço do Escravo: Normalmente, o endereço 0 é utilizado quando o servidor é um Gateway MODBUS TCP ou MODBUS RTU via TCP, e o mesmo transmite a requisição para todos os dispositivos da rede. Quando o endereço 0 é utilizado, o cliente não espera por uma resposta, seu uso serve somente para comandos de escrita. Além disso, de acordo com a Norma MODBUS, a faixa de endereços válidos para escravos é de 0 a 247, sendo os endereços 248 a 255 reservados.

Polling: Este parâmetro indica com que frequência a comunicação definida por esta relação deve ser executada. Ao ser finalizada uma comunicação será aguardado um tempo igual ao polling configurado e, após, será executada uma nova comunicação o mais rápido possível.

Área de Diagnósticos do Mapeamento: Esse campo é limitado pelo tamanho da memória de variáveis de saídas endereçáveis (%Q) de cada UCP, a qual pode ser consultada na seção [Memória](#). Os diagnósticos da relação MODBUS configurada estão descritos na Tabela 93.

Tamanho dos Dados de Leitura e de Escrita: Detalhes do tamanho dos dados suportados por cada função estão descritos nas notas da seção [Configuração das Requisições – Configuração por Mapeamento Simbólico](#).

ATENÇÃO

Quando o acesso à memória de dados da comunicação for entre equipamentos com endianness diferentes (Little-Endian e Big-Endian), pode ocorrer a inversão dos dados lidos/escritos. Neste caso o usuário deve fazer o ajuste dos dados na aplicação.

Variável IEC de Leitura: Caso o tipo de dado MODBUS seja *Coil* ou *Input Status* (bit), o endereço inicial das variáveis IEC de leitura terá o formato por exemplo *%IX10.1*. Porém, se o tipo de dado MODBUS for *Holding Register* ou *Input Register* (16 bits), o endereço inicial das variáveis IEC de leitura terá o formato *%IW*. Esse campo é limitado pelo tamanho da memória de variáveis de entrada endereçáveis (%I) de cada UCP, a qual pode ser consultada na seção [Memória](#).

Variável IEC de Escrita: Caso o tipo de dado MODBUS seja *Coil*, o endereço inicial das variáveis IEC de escrita terá o formato por exemplo *%QX10.1*. Porém, se o tipo de dado MODBUS for *Holding Register* (16 bits), o endereço inicial das variáveis IEC de escrita terá o formato *%QW*. Esse campo é limitado pelo tamanho da memória de variáveis de saídas endereçáveis (%Q) de cada UCP, a qual pode ser consultada na seção [Memória](#).

Máscara de Escrita do Register: A função *Máscara de Escrita* (FC 22), através de uma lógica entre o valor já escrito e as duas words configuradas neste campo, sendo a *%QW(0)* para a máscara AND e a *%QW(2)* para a máscara OR; permite ao usuário manipular a word. Esse campo é limitado pelo tamanho da memória de variáveis de saídas endereçáveis (%Q) de cada UCP, a qual pode ser consultada na seção [Memória](#).

Padrão: O padrão de fábrica não pode ser definido para os campos *Área de Diagnóstico do Mapeamento*, *Variável IEC de Leitura*, *Variável IEC de Escrita* e *Máscara de Escrita das Variáveis IEC*, pois a criação de uma relação pode ser realizada em qualquer momento no desenvolvimento da aplicação, fazendo com que o próprio software MasterTool IEC XE alocue um valor, da faixa de variáveis de saída de representação direta (%Q), ainda não utilizado. O padrão de fábrica não pode ser definido para os campos *Tamanho dos Dados de Leitura* e *Tamanho dos Dados de Escrita*, pois eles vão variar de acordo com o tipo de dado MODBUS selecionado.

ATENÇÃO

Diferentemente de outras tarefas de uma aplicação, quando for atingida uma marca de depuração na MainTask, a tarefa de uma instância MODBUS Ethernet Cliente, e qualquer outra tarefa MODBUS, irá parar de ser executada no momento em que tentar efetuar uma escrita em uma área de memória. Isto ocorre para manter a consistência dos dados das áreas de memória enquanto a MainTask não estiver em execução.

5.5.8.3. Disparo de Relações MODBUS Cliente de Forma Acíclica

Para disparar relações MODBUS Cliente de forma acíclica, sugere-se o seguinte método, que pode ser implementado de maneira simples no programa da aplicação do usuário:

- Definir tempo máximo de polling para as relações;
- Manter a relação normalmente desabilitada;
- Habilitar a relação no momento em que se deseja executá-la;
- Esperar pela confirmação de término da execução da relação, e neste momento desabilitá-la novamente.

5.5.9. MODBUS Ethernet Servidor

Este protocolo está disponível para as UCPs da Série Nexto nos seus canais Ethernet. Ao selecionar esta opção no MasterTool IEC XE, a UCP passa a ser servidor da comunicação MODBUS, permitindo a conexão com dispositivos cliente MODBUS. Este protocolo somente está disponível quando a UCP estiver em modo de execução (Modo *Run*).

Há dois modos de configuração para este protocolo. Um deles faz uso de *Representação Direta (%Q)*, no qual as variáveis são definidas pelo seu endereço. O outro, chamado *Mapeamento Simbólico*, tem as variáveis definidas pelo seu nome.

O procedimento para inserir uma instância de protocolo é encontrado com detalhes no Manual de Utilização do MasterTool IEC XE – MU299048.

5.5.9.1. Configuração do Protocolo MODBUS Ethernet Servidor por Mapeamento Simbólico

Para configurar este protocolo usando *Mapeamento Simbólico*, é necessário executar os seguintes passos:

- Configurar os parâmetros gerais do protocolo MODBUS servidor, como: porta TCP, seleção de protocolo, filtros de IP para Escrita e para Leitura (disponível no botão de configuração de filtros) e tempos de comunicação (disponível no botão de configurações avançadas do Servidor).
- Adicionar e configurar os mapeamentos MODBUS, especificando o nome da variável, tipo de dados, endereço inicial do dado e tamanho do dado.

As descrições de cada configuração estão relacionadas a seguir, neste capítulo.

5.5.9.1.1. Parâmetros Gerais do Protocolo MODBUS Servidor – Configuração por Mapeamento Simbólico

Os parâmetros gerais, encontrados na tela inicial de configuração do protocolo MODBUS como apresentado na figura abaixo.

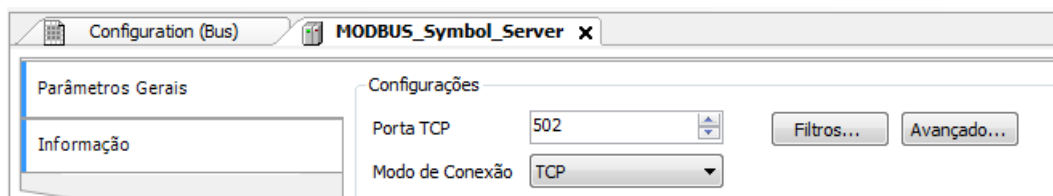


Figura 76: Tela de Configuração Parâmetros Gerais MODBUS Servidor

Configuração	Descrição	Padrão	Opções
Porta TCP	Porta TCP	502	2 a 65534
Modo de Conexão	Seleção do protocolo	TCP	RTU via TCP TCP

Tabela 120: Configurações Gerais MODBUS Servidor

Nota:

Porta TCP: Caso sejam adicionadas várias instâncias do protocolo em uma única interface Ethernet, diferentes portas TCP devem ser selecionadas para cada instância. Algumas portas TCP, entre as possibilidades mencionadas acima, são reservadas e, portanto, não podem ser utilizadas. Ver tabela [Portas TCP/UDP reservadas](#).

As configurações presentes no botão *Filtros...*, descritas na tabela abaixo, são relativas aos filtros de comunicação TCP:

Configuração	Descrição	Padrão	Opções
Filtro de Endereço IP para Escrita	Especifica um intervalo de IPs com acesso de escrita nas variáveis declaradas na relação MODBUS.	0.0.0.0	0.0.0.0 a 255.255.255.255
Filtro de Máscara para Escrita	Especifica a máscara de subrede em conjunto com o parâmetro Filtro de Endereço IP para Escrita.	0.0.0.0	0.0.0.0 a 255.255.255.255
Filtro de Endereço IP para Leitura	Especifica um intervalo de IPs com acesso de leitura nas variáveis declaradas na relação MODBUS.	0.0.0.0	0.0.0.0 a 255.255.255.255
Filtro de Máscara para Leitura	Especifica a máscara de subrede em conjunto com o parâmetro Filtro de Endereço IP para Leitura.	0.0.0.0	0.0.0.0 a 255.255.255.255

Tabela 121: Filtros de IP

Nota:

Filtros: Os filtros são utilizados para estabelecer um intervalo de endereços IP que têm acesso de escrita ou leitura nas relações MODBUS, sendo individualmente configurados. O critério de permissão é realizado através de uma operação lógica AND entre o Filtro de Máscara para Escrita e o endereço IP do cliente. Caso o resultado seja igual ao Filtro de Endereço IP para Escrita, o cliente tem direito de escrita. Por exemplo, se o Filtro de Endereço IP para Escrita = 192.168.15.0 e o Filtro de Máscara para Escrita = 255.255.255.0, então somente clientes com endereço IP = 192.168.15.x terão direito de escrita. O mesmo procedimento é aplicado nos parâmetros de Filtro de Leitura para definir os direitos de leitura.

Os tempos de comunicação do protocolo MODBUS Servidor, encontrados no botão *Avançado...* da tela de configuração, estão divididos em: *Ciclo da Tarefa* e *Time-out da Inatividade da Conexão*.

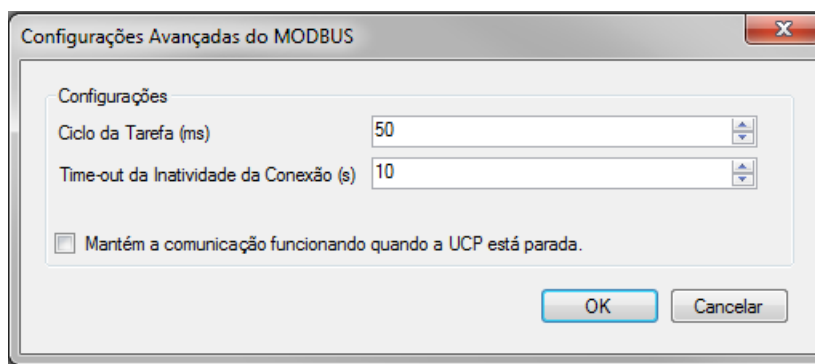


Figura 77: Janela de Configurações Avançadas do MODBUS Servidor

Configuração	Descrição	Padrão	Opções
Ciclo da Tarefa (ms)	Tempo para execução da instância dentro do ciclo, sem considerar o tempo de execução da mesma	50	5 a 100
Time-out da Inatividade da Conexão (s)	Tempo máximo de inatividade entre cliente e servidor antes da conexão ser fechada pelo servidor	10	1 a 3600
Mantém a comunicação funcionando quando a UCP está parada.	Habilita o Modbus Symbol Server para rodar enquanto a UCP estiver em STOP ou após um breakpoint	desmarcado	marcado ou desmarcado

Tabela 122: Configurações Avançadas MODBUS Servidor

Notas:

Ciclo da Tarefa: O usuário deverá ter cuidado ao alterar esse parâmetro, pois o mesmo interfere diretamente no tempo de resposta, volume de dados por varredura e, principalmente, no balanceamento dos recursos da UCP entre comunicações e outras tarefas.

Time-out da Inatividade da Conexão: Esse parâmetro foi criado para evitar que a quantidade máxima de conexões TCP seja atingida, imaginando que conexões inativas permanecessem abertas pelos mais diversos problemas. Enfim, indica por quanto tempo uma conexão (cliente ou servidora) pode permanecer aberta sem ser utilizada, ou seja, sem trocar mensagens de comunicação. Se o tempo especificado for atingido, a conexão simplesmente é fechada, liberando uma entrada na tabela de conexões.

5.5.9.1.2. Diagnósticos MODBUS Servidor – Configuração por Mapeamento Simbólico

Os diagnósticos e comandos do protocolo MODBUS Servidor configurado, seja por mapeamento simbólico ou por representação direta, são armazenados em variáveis do tipo *T_DIAG_MODBUS_ETH_SERVER_1* e ainda para o mapeamento por representação direta estão em 4 bytes e 8 words, as quais estão descritas na tabela abaixo (n é o valor configurado no campo *Endereço Inicial de Diagnósticos em %Q*):

5. CONFIGURAÇÃO

Variável de Representação Direta	Variável de diagnóstico do tipo T_DIAG_MODBUS_ETH_SERVER_1.*	Tamanho	Descrição
Bits de diagnóstico:			
%QX(n).0	tDiag. bRunning	BIT	O servidor está em execução.
%QX(n).1	tDiag. bNotRunning	BIT	O servidor não está em execução (ver bit bInterruptedByCommand).
%QX(n).2	tDiag. bInterruptedByCommand	BIT	O bit bNotRunning foi habilitado, pois o servidor foi interrompido pelo usuário através de bits de comando.
%QX(n).3	tDiag. bConfigFailure	BIT	Diagnóstico descontinuado.
%QX(n).4	tDiag. bRXFailure	BIT	Diagnóstico descontinuado.
%QX(n).5	tDiag. bTXFailure	BIT	Diagnóstico descontinuado.
%QX(n).6	tDiag. bModuleFailure	BIT	Diagnóstico descontinuado.
%QX(n).7	tDiag. bDiag_7_reserved	BIT	Reservado
%QB(n+1)	byDiag_1_reserved	BYTE	Reservado
Bits de comando, reiniciados automaticamente:			
%QX(n+2).0	tCommand. bStop	BIT	Parar o servidor.
%QX(n+2).1	tCommand. bRestart	BIT	Reiniciar o servidor.
%QX(n+2).2	tCommand. bResetCounter	BIT	Reiniciar as estatísticas dos diagnósticos (contadores).
%QX(n+2).3	tCommand. bDiag_19_reserved	BIT	Reservado
%QX(n+2).4	tCommand. bDiag_20_reserved	BIT	Reservado
%QX(n+2).5	tCommand. bDiag_21_reserved	BIT	Reservado
%QX(n+2).6	tCommand. bDiag_22_reserved	BIT	Reservado
%QX(n+2).7	tCommand. bDiag_23_reserved	BIT	Reservado
%QB(n+3)	byDiag_3_reserved	BYTE	Reservado
Estatísticas de comunicação:			
%QW(n+4)	tStat. wActiveConnections	WORD	Número de conexões estabelecidas entre cliente e servidor (0 a 64).
%QW(n+6)	tStat. wTimeoutClosedConnections	WORD	Contador de conexões, entre cliente e servidor, interrompidas depois de um período de inatividade – time-out (0 a 65535).
%QW(n+8)	tStat. wClientClosedConnections	WORD	Contador de conexões interrompidas devido à solicitação do cliente (0 a 65535).

Variável de Representação Direta	Variável de diagnóstico do tipo T_DIAG_MODBUS_ETH_SERVER_1.*	Tamanho	Descrição
%QW(n+10)	tStat. wRXFrames	WORD	Contador de frames Ethernet recebidos pelo servidor, sendo que um frame Ethernet pode conter mais de uma requisição (0 a 65535).
%QW(n+12)	tStat. wRXRequests	WORD	Contador de requisições recebidas pelo servidor e respondidas normalmente (0 a 65535).
%QW(n+14)	tStat. wTXExceptionResponses	WORD	Contador de requisições recebidas pelo servidor e respondidas com códigos de exceção (0 a 65535). Os códigos de exceção estão relacionados abaixo: 1: o código da função (FC) é legal, porém não suportado. 2: relação não encontrada nestes dados MODBUS. 3: valor ilegal para o endereço. 128: o mestre/cliente não tem direito de escrita ou leitura. 129: a relação MODBUS está desabilitada.
%QW(n+16)	tStat. wRXIllegalRequests	WORD	Contador de requisições ilegais. (0 a 65535).
%QW(n+18)	tStat. wDiag_18_Reserved	WORD	Reservado

Tabela 123: Diagnósticos MODBUS Servidor

Nota:

Contadores: Todos os contadores dos diagnósticos do MODBUS Ethernet Servidor retornam à zero quando o valor limite 65535 é ultrapassado.

5.5.9.1.3. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico

A configuração dos mapeamentos MODBUS, visualizada na figura abaixo, segue os parâmetros descritos na tabela abaixo:

	Variável de Valor	Tipo do Dado	Endereço Inicial do Dado	Endereço Inicial Absoluto do Dado	Tamanho do Dado	Faixa de Dados
*		▼				

Figura 78: Tela de Mapeamentos de dados MODBUS Servidor

Configuração	Descrição	Padrão	Opções
Variável de Valor	Nome da variável simbólica	-	Nome de uma variável declarada em um programa ou GVL
Tipo do Dado	Tipo de dado MODBUS	-	Coil Input Status Holding Register Input Register
Endereço Inicial do Dado	Endereço inicial dos dados MODBUS	-	1 a 65536
Endereço Inicial Absoluto do Dado	Endereço inicial absoluto dos dados MODBUS conforme o seu tipo.	-	-
Tamanho do Dado	Tamanho do dado MODBUS	-	1 a 65536
Faixa de Dados	Faixa de endereços do dado configurado	-	-

Tabela 124: Configuração dos Mapeamentos MODBUS

Notas:

Variável de Valor: Esse campo é utilizado para especificar uma variável simbólica na relação MODBUS.

Tipo do Dado: Esse campo é utilizado para especificar o tipo de dado utilizado na relação MODBUS.

Endereço Inicial do Dado: Endereço inicial do dado de um mapeamento MODBUS.

Endereço Inicial Absoluto do Dado: Endereço inicial absoluto dos dados MODBUS conforme o seu tipo. Por exemplo, o Holding Register com endereço 5 possui endereço absoluto 400005. Este campo é apenas de leitura e está disponível para auxiliar na configuração do Cliente/Mestre MODBUS que irá comunicar-se com este dispositivo. Os valores dependem do endereço base (offset) de cada tipo de dado MODBUS e do endereço permitido para cada tipo de dado.

Tamanho do Dado: O valor de Tamanho do Dado especifica a quantidade máxima de dados que uma relação MODBUS poderá acessar, a partir do endereço inicial. Sendo assim, para ler uma faixa de endereços contínua, é necessário que todos os endereços estejam declarados em uma única relação. Este campo varia de acordo com o tipo de dado MODBUS configurado.

Faixa de Dados: É um campo somente de leitura e informa a faixa de endereços que está sendo usada por esse mapeamento. Ele é formado pela soma dos campos *Endereço Inicial* e *Tamanho do Dado*. Não podem haver sobreposições de faixa com outros mapeamentos do mesmo *Tipo de Dado*.

ATENÇÃO

Diferentemente de outras tarefas de uma aplicação, quando for atingida uma marca de depuração na MainTask, a tarefa de uma instância MODBUS Ethernet Servidor, e qualquer outra tarefa MODBUS, irá parar de ser executada no momento em que tentar efetuar uma escrita em uma área de memória. Isto ocorre para manter a consistência dos dados das áreas de memória enquanto a MainTask não estiver em execução.

5.5.9.2. Configuração do Protocolo MODBUS Ethernet Servidor por Representação Direta (%Q)

Para configurar este protocolo usando *Representação Direta (%Q)*, é necessário executar os seguintes passos:

- Configurar os parâmetros gerais do protocolo MODBUS servidor, como: tempos de comunicação, endereço e variáveis de representação direta (%Q) para receber os diagnósticos e controlar as relações.
- Adicionar e configurar relações MODBUS, especificando o tipo de dado MODBUS, variáveis de representação direta (%Q) para receber/escrever os dados e quantidade de dados a comunicar.

As descrições de cada configuração estão relacionadas a seguir, neste capítulo.

5.5.9.2.1. Parâmetros Gerais do Protocolo MODBUS Servidor – Configuração por Representação Direta (%Q)

Os parâmetros gerais, encontrados na tela inicial de configuração do protocolo MODBUS (figura abaixo), são definidos como:

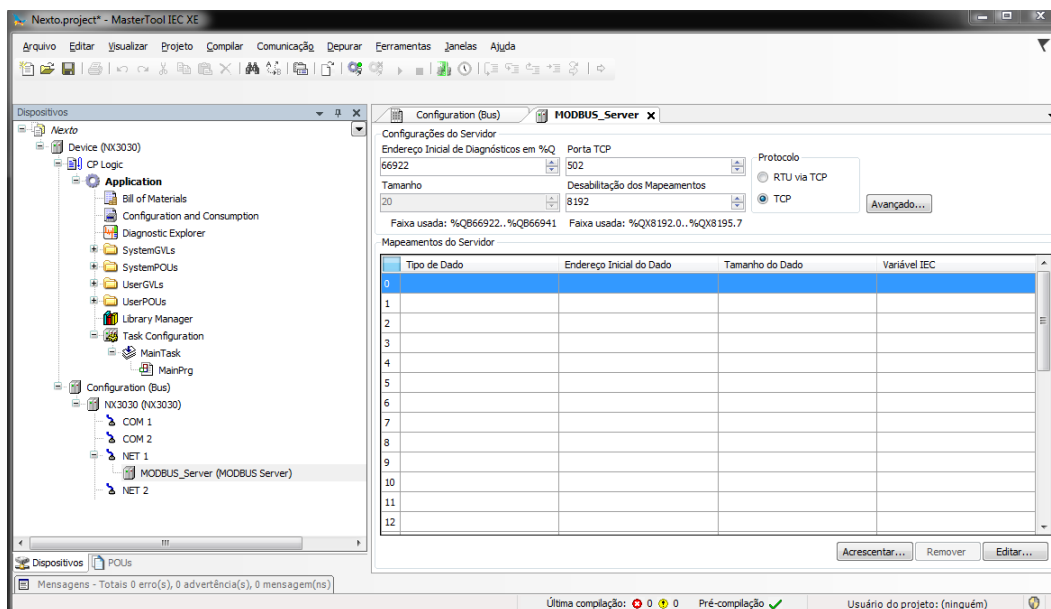


Figura 79: Tela de Configuração MODBUS Servidor

Porta TCP, Protocolo, Variáveis de representação direta (%Q) para controlar as relações e os diagnósticos:

Configuração	Descrição	Padrão	Opções
Endereço Inicial de Diagnósticos em %Q	Endereço inicial das variáveis de diagnóstico	-	0 a 2147483628
Tamanho	Tamanho da área de diagnósticos	20	Desabilitado para edição
Porta TCP	Porta TCP	502	2 a 65534
Desabilitação dos Mapeamentos	Endereço inicial utilizado para desabilitar as relações MODBUS	-	0 a 2147483644
Protocolo	Seleção do protocolo	TCP	RTU via TCP TCP

Tabela 125: Configurações das Variáveis de Representação Direta e Endereçamento

Notas:

Endereço Inicial de Diagnósticos em %Q: Esse campo é limitado pelo tamanho da memória de variáveis de saídas endereçáveis (%Q) de cada UCP, a qual pode ser consultada na seção [Memória](#).

Porta TCP: Caso sejam adicionadas várias instâncias do protocolo em uma única interface Ethernet, diferentes portas TCP devem ser selecionadas para cada instância. Algumas portas TCP, entre as possibilidades mencionadas acima, são reservadas e, portanto, não podem ser utilizadas. Ver tabela [Portas TCP/UDP reservadas](#).

Desabilitação dos Mapeamentos: Composta por 32 bits, utilizados para desabilitar, individualmente, as 32 relações MODBUS configuradas no espaço *Mapeamentos do Servidor*. A relação é desabilitada quando o bit, correspondente à relação, for igual a 1, caso contrário, o mapeamento está habilitado. Esse campo é limitado pelo tamanho da memória de variáveis de saídas endereçáveis (%Q) de cada UCP, a qual pode ser consultada na seção [Memória](#).

Padrão: O padrão de fábrica não pode ser definido para os campos *Endereço Inicial de Diagnósticos em %Q* e *Desabilitação dos Mapeamentos*, pois a criação de uma instância do protocolo pode ser realizada em qualquer momento no

desenvolvimento da aplicação, fazendo com que o próprio software MasterTool IEC XE alocue um valor, da faixa de variáveis de saída de representação direta (%Q), ainda não utilizado.

Os tempos de comunicação do protocolo MODBUS Servidor, encontrados no botão *Avançado...* da tela de configuração, estão divididos em *Ciclo da Tarefa (ms)* e *Time-out da Inatividade da Conexão (s)*, maiores detalhes estão descritos na seção [Parâmetros Gerais do Protocolo MODBUS Servidor – Configuração por Mapeamento Simbólico](#).

Os diagnósticos e comandos do protocolo MODBUS estão descritos na Tabela 123.

5.5.9.2.2. Configuração dos Mapeamentos – Configuração por Representação Direta (%Q)

A configuração das relações MODBUS, visualizada nas figuras abaixo, segue os parâmetros descritos na tabela abaixo:

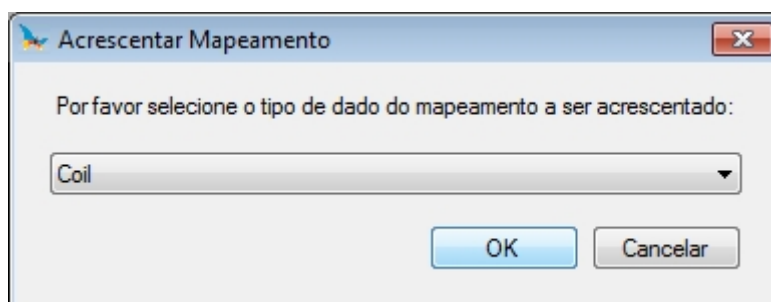


Figura 80: Tipo de Dado MODBUS

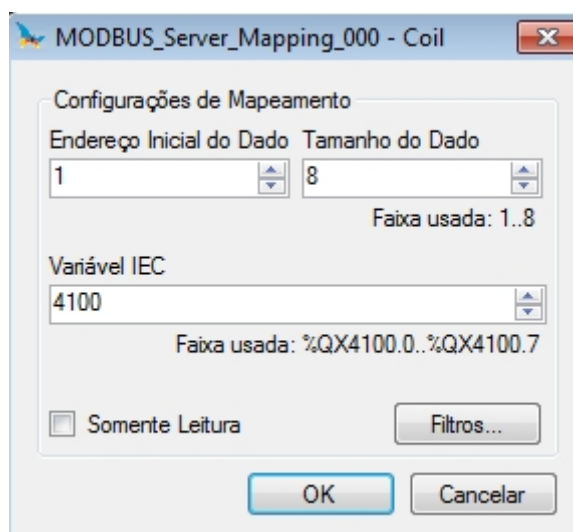


Figura 81: Função MODBUS Servidor

Configuração	Descrição	Padrão	Opções
Tipo de Dado	Tipo de dado MODBUS	Coil	Coil (1 bit) Holding Register (16 bits) Input Status (1 bit) Input Register (16 bits)
Endereço Inicial do Dado	Endereço inicial dos dados MODBUS	1	1 a 65536
Tamanho do Dado	Tamanho do dado MODBUS	8	1 a 65536 (Holding Register e Input Register) 8 a 65536 (Coil e Input Status)
Variável IEC	Endereço inicial das variáveis (%Q)	-	0 a 2147483647
Somente Leitura	Somente permite a leitura	Desabilitada	Habilitada ou Desabilitada

Tabela 126: Mapeamentos do Servidor

Notas:

Opções: Os valores descritos na coluna *Opções* podem variar de acordo com o tipo de dado MODBUS configurado.

Tamanho do Dado: O valor de *Tamanho do Dado* define a quantidade máxima de dados que uma relação MODBUS poderá acessar, a partir do endereço inicial. Sendo assim, para ler uma faixa de endereços contínua, é necessário que todos os endereços estejam declarados em uma única relação. Este campo varia de acordo com o tipo de dado MODBUS configurado, ou seja, quando selecionado tipo *Coil* ou *Input Status*, o campo *Tamanho do Dado* deve ser um número múltiplo de oito. Também deve-se dar atenção para que o valor máximo não seja superior ao tamanho da memória de saídas endereçáveis e não sejam atribuídos os mesmos valores já utilizados durante a aplicação.

ATENÇÃO

Quando o acesso à memória de dados da comunicação for entre equipamentos com endianness diferentes (Little-Endian e Big-Endian), pode ocorrer a inversão dos dados lidos/escritos. Neste caso o usuário deve fazer o ajuste dos dados na aplicação.

Variável IEC: Caso o tipo de dado MODBUS seja *Coil* ou *Input Status* (bit), o endereço inicial das variáveis IEC terá o formato por exemplo *%QX10.1*. Porém, se o tipo de dado MODBUS for *Holding Register* ou *Input Register* (16 bits), o endereço inicial das variáveis IEC terá o formato *%QW*. Esse campo é limitado pelo tamanho da memória de variáveis de saídas endereçáveis (%Q) de cada UCP, a qual pode ser consultada na seção [Memória](#).

Somente Leitura: Quando habilitada, somente permite que o mestre da comunicação leia os dados das variáveis, não permitindo a escrita. Opção válida somente para as funções de escrita.

Padrão: O padrão de fábrica não pode ser definido para o campo *Variável IEC*, pois a criação de uma instância do protocolo pode ser realizada em qualquer momento no desenvolvimento da aplicação, fazendo com que o próprio software MasterTool IEC XE aloque um valor, da faixa de variáveis de saída de representação direta (%Q), ainda não utilizado. O padrão de fábrica não pode ser definido para o campo *Tamanho do Dado*, pois ele vai variar de acordo com o tipo de dado MODBUS selecionado.

As configurações presentes no botão *Filtros...*, descritas na tabela abaixo, são relativas aos filtros de comunicação TCP:

Configuração	Descrição	Padrão	Opções
Filtro de Endereço IP para Escrita	Especifica um intervalo de IPs com acesso de escrita nas variáveis declaradas na relação MODBUS.	0.0.0.0	0.0.0.0 a 255.255.255.255
Filtro de Máscara para Escrita	Especifica a máscara de subrede em conjunto com o parâmetro Filtro de Endereço IP para Escrita.	0.0.0.0	0.0.0.0 a 255.255.255.255
Filtro de Endereço IP para Leitura	Especifica um intervalo de IPs com acesso de leitura nas variáveis declaradas na relação MODBUS.	0.0.0.0	0.0.0.0 a 255.255.255.255
Filtro de Máscara para Leitura	Especifica a máscara de subrede em conjunto com o parâmetro Filtro de Endereço IP para Leitura.	0.0.0.0	0.0.0.0 a 255.255.255.255

Tabela 127: Filtros de IP

Nota:

Filtros: Os filtros são utilizados para estabelecer um intervalo de endereços IP que têm acesso de escrita ou leitura nas relações MODBUS, sendo individualmente configurados. O critério de permissão é realizado através de uma operação lógica AND entre o Filtro de Máscara para Escrita e o endereço IP do cliente. Caso o resultado seja igual ao Filtro de Endereço IP para Escrita, o cliente tem direito de escrita. Por exemplo, se o Filtro de Endereço IP para Escrita = 192.168.15.0 e o Filtro de Máscara para Escrita = 255.255.255.0, então somente clientes com endereço IP = 192.168.15.x terão direito de escrita. O mesmo procedimento é aplicado nos parâmetros de Filtro de Leitura para definir os direitos de leitura.

Nas relações definidas anteriormente, o tamanho máximo de dados MODBUS pode ser 65536 (máximo valor configurado no campo *Tamanho do Dado*). Porém, a requisição que chega ao MODBUS Ethernet Servidor deverá endereçar um subconjunto desse mapeamento e esse grupo deve ter, no máximo, o tamanho de dados que depende do código da função, os quais estão definidos abaixo:

- Leitura de Coils (FC 1): 2000
- Leitura de Input Status (FC 2): 2000
- Leitura de Holding Registers (FC 3): 125
- Leitura de Input Registers (FC 4): 125
- Escrita de Um Coil (FC 5): 1
- Escrita de Um Holding Register (FC 6): 1
- Forçamento de Múltiplos Coils (FC 15): 1968
- Escrita de Holding Registers (FC 16): 123
- Máscara de Escrita do Register (FC 22): 1
- Leitura/Escrita de Holding Registers (FC 23):
 - Leitura: 121
 - Escrita: 121

ATENÇÃO

Diferentemente de outras tarefas de uma aplicação, quando for atingida uma marca de depuração na MainTask, a tarefa de uma instância MODBUS Ethernet Servidor, e qualquer outra tarefa MODBUS, irá parar de ser executada no momento em que tentar efetuar uma escrita em uma área de memória. Isto ocorre para manter a consistência dos dados das áreas de memória enquanto a MainTask não estiver em execução.

5.5.10. OPC DA Servidor

Para comunicar com as UCPs da Série Nexto é possível utilizar a tecnologia OPC DA (*Open Platform Communications Data Access*). Esta plataforma de comunicação aberta foi desenvolvida para ser o padrão utilizado nas comunicações industriais. Baseado na arquitetura cliente/servidor, oferece inúmeras vantagens no desenvolvimento de projeto e facilidades na comunicação com os sistemas de automação.

Uma analogia muito comum, utilizada para descrever a tecnologia OPC DA, é a de uma impressora. Quando corretamente conectada, o computador precisa de um driver para ter a interface com o equipamento. Muito similar, o OPC DA auxilia na interface entre o sistema de supervisão com os dados de campo no CP.

Quando se trata do desenvolvimento de projetos, configurar a comunicação e trocar informações entre os sistemas é extremamente simples utilizando tecnologia OPC DA. Utilizando outros drivers, baseados em endereços, é necessário criar tabelas para relacionar as tags do sistema de supervisão e as variáveis do controlador programável. Quando as áreas de dados são alteradas, no decorrer do desenvolvimento do projeto, é necessário refazer os mapeamentos e novas tabelas com as relações entre as informações do CP com o sistema de Controle Supervisório e Aquisição de Dados (SCADA).

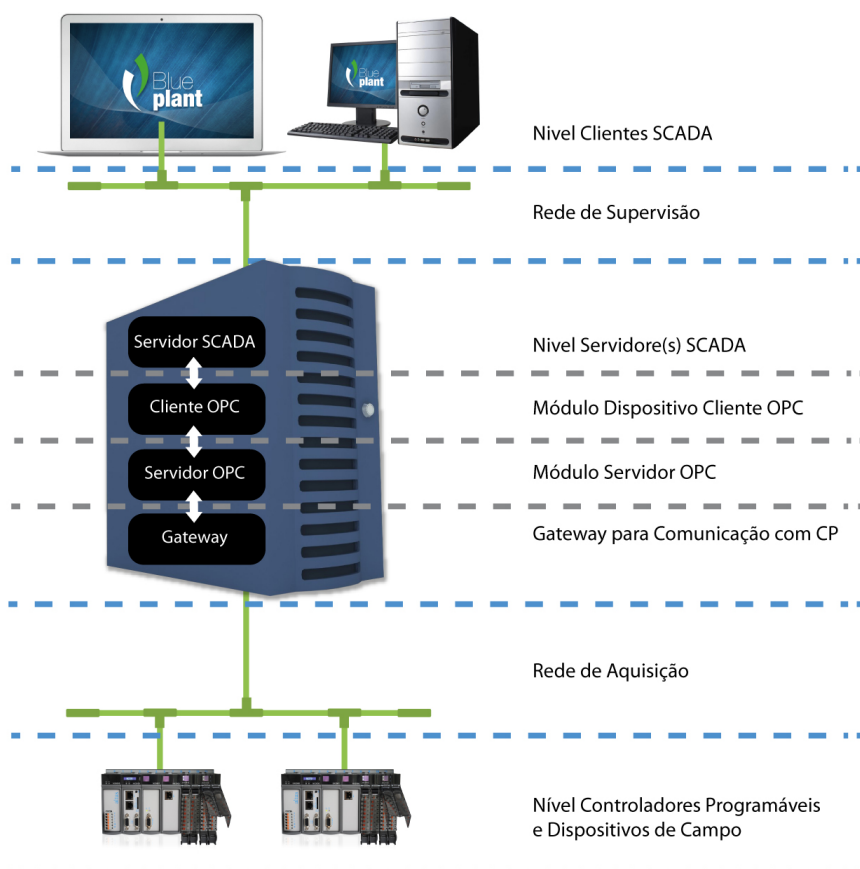


Figura 82: Arquitetura OPC DA

A figura acima apresenta uma arquitetura para comunicação de sistema SCADA e CPs em projeto de automação. Todos os papéis presentes na comunicação estão explícitos nesta figura independente do local onde estejam executando, eles podem estar em um mesmo equipamento ou em equipamentos diferentes. Cada um dos papéis desta arquitetura é descrito na tabela abaixo.

Papel	Descrição
Nível Controladores Programáveis e Dispositivos de Campo	Os dispositivos de campo e os CPs são os dispositivos nos quais as informações do estado de operações e controle da planta são armazenados. Os sistemas SCADA acessam as informações nestes dispositivos e armazenam nos servidores SCADA para consulta pelos Clientes SCADA durante a operação da planta.
Rede de Aquisição	A rede de aquisição é a rede na qual trafegam as mensagens para solicitar os dados que são coletados dos dispositivos de campo.
Gateway para Comunicação com CP	Para a comunicação entre o Servidor OPC DA e os CPs da Série Nexto é utilizado um gateway que permite esta comunicação. Sempre é necessário existir um gateway na mesma subrede do CP como descrito no capítulo Configurações de Comunicação, no Manual de Utilização MasterTool IEC XE – MU299048.
Módulo Servidor OPC	O Servidor OPC DA é um Módulo responsável por receber as requisições OPC DA e traduzi-las para a comunicação com os dispositivos de campo.
Módulo Dispositivo Cliente OPC	O módulo Dispositivo do Cliente OPC DA é responsável por fazer requisições aos Servidores OPC DA utilizando o protocolo OPC DA. Os dados coletados por ele são armazenados na base de dados do Servidor SCADA.
Nível Servidor SCADA	O Servidor SCADA é responsável por se conectar aos diversos dispositivos de comunicação e armazenar os dados coletados destes dispositivos em uma base de dados para que possam ser consultados pelos Clientes SCADA.
Rede de Supervisão	A rede de supervisão é a rede pela qual os Clientes SCADA estão conectados aos Servidores SCADA. Em uma topologia na qual não se usa diversos Clientes ou que o servidor e o Cliente estejam instalados em um mesmo equipamento, não existe este tipo de rede.
Nível Clientes SCADA	Os clientes SCADA são responsáveis por solicitar aos servidores SCADA os dados necessários para exibir em uma tela onde é executada a operação de uma planta. Através deles é possível executar leituras e escritas em dados armazenados na base de dados do Servidor SCADA.

Tabela 128: Descrição dos Papéis em uma Arquitetura com Servidor OPC DA

A relação entre as tags dos sistemas de supervisão e os dados do processo nas variáveis do controlador é totalmente transparente. Isso significa que se as áreas de dados são alteradas no decorrer do desenvolvimento do projeto, não há a necessidade de refazer relações entre as informações do CP com o SCADA. Basta utilizar a nova variável disponibilizada pelo CP nos sistemas que requisitam esse dado.

O uso do OPC DA oferece maior produtividade e conectividade com os sistemas SCADA. Contribui na redução do tempo de desenvolvimento de aplicações e nos custos com manutenção. Possibilita, ainda, inserção de novos dados na comunicação de forma simplificada com maior flexibilidade e interoperabilidade entre os sistemas de automação por ser um padrão aberto.

O Servidor OPC DA é instalado juntamente com a instalação do MasterTool IEC XE e sua configuração é realizada dentro da ferramenta. Vale salientar que o OPC DA está disponível somente nas interfaces Ethernet locais das UCPs Nexto. Os módulos de expansão Ethernet não suportam essa funcionalidade.

5.5.10.1. Criando um Projeto para Comunicação OPC DA

Diferente das comunicações com drivers como MODBUS e PROFIBUS DP, para configurar a comunicação OPC DA basta configurar o nó corretamente e indicar quais as variáveis que serão utilizadas na comunicação. Existem duas formas de indicar quais as variáveis de projeto estarão disponíveis no Servidor OPC DA. Em ambos os casos é necessário adicionar o objeto *Symbol Configuration* à aplicação, caso este não esteja presente. Para adicioná-lo basta clicar com o botão direito do mouse sobre o objeto *Application* e selecionar a opção.

ATENÇÃO

As variáveis exibidas no objeto *IoConfig_Globals*, *IoConfig_Application_Mappings* e *IoConfig_Global_Mappings* são utilizadas internamente para controle de E/S e não devem ser utilizadas pelo usuário.

ATENÇÃO

Além das variáveis declaradas nas POU's em linguagem SFC são exibidas algumas variáveis criadas implicitamente. Para cada passo criado é criada uma variável do tipo *IecSfc.SFCStepType* onde podem ser monitorados os estados do passo, ou seja, se o mesmo é ativo ou não e o tempo que é ativo conforme define a norma IEC61131-3. Para cada transição é criada uma variável do tipo BOOL que define se a transição é verdadeira ou falsa. Essas variáveis são exibidas no objeto *Symbol Configuration* podendo ser disponibilizadas para acesso pelo Cliente OPC DA.

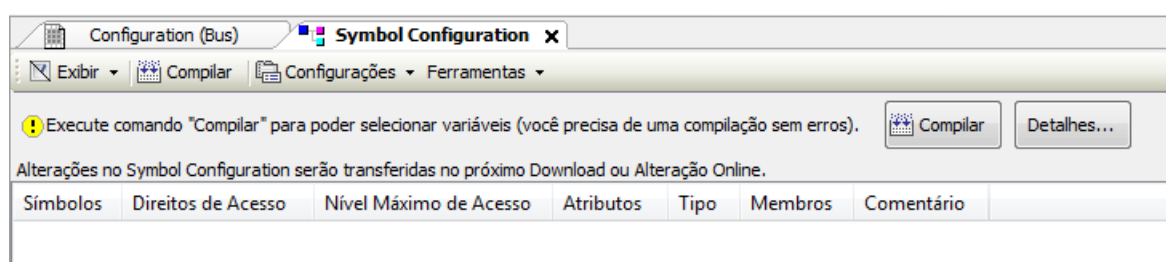








Figura 83: Objeto Symbol Configuration

A tabela abaixo apresenta a descrição dos campos da tela de configurações dos símbolos no objeto *Symbol Configuration*.

Campo	Descrição
Símbolos	Identificador da variável que será disponibilizada para o Servidor OPC DA.
Direitos de Acesso	Indica qual o nível de acesso possível no símbolo declarado. Quando não se utiliza esta coluna, a mesma fica vazia e o nível de acesso é máximo. Caso contrário o nível de acesso pode ser modificado clicando sobre o campo. As opções possíveis são as seguintes: Somente leitura  Somente escrita  Leitura e escrita 
Nível Máximo de Acesso	Indica o máximo nível de acesso que é possível atribuir à variável. Os símbolos que representam têm o mesmo significado do campo Direitos de Acesso. Não é possível alterar e é indicado pela presença ou não do <i>attribute 'symbol'</i>
Atributos	Indica se está sendo utilizado <i>attribute 'symbol'</i> quando declarada a variável. Quando não é utilizado esta coluna fica vazia. Para os casos nos quais se usa o atributo o comportamento é o seguinte: <i>attribute 'symbol' := 'read'</i> a coluna exibe  <i>attribute 'symbol' := 'write'</i> a coluna exibe  <i>attribute 'symbol' := 'readwrite'</i> a coluna exibe 

Campo	Descrição
Tipo	Tipo de dado da variável declarada.
Membros	Quando o tipo de dado for uma Struct é habilitado um botão nesta coluna. Ao clicar no botão é possível selecionar quais elementos da estrutura serão disponibilizados para o Servidor OPC.
Comentário	Comentário da variável inserido na POU ou GVL onde a mesma é declarada. Para aparecer como comentário da variável, o comentário deve ser inserido uma linha antes da declaração da variável, no editor quando em modo texto ou na coluna comentário, quando em modo tabular.

Tabela 129: Descrição dos Campos da Tela do Objeto Symbol Configuration

Ao executar uma alteração nas configurações do projeto, como adicionar ou remover variáveis, se faz necessário executar o comando *Compilar* para atualizar a lista de variáveis. Este comando deve ser executado até que a mensagem presente na Figura 83 desapareça. Após isso todas as variáveis disponíveis no projeto, sejam declaradas em POU, GVLs e diagnósticos, serão exibidas e podem ser selecionadas. As variáveis selecionadas estarão disponíveis no Servidor OPC DA para acesso pelos Clientes.

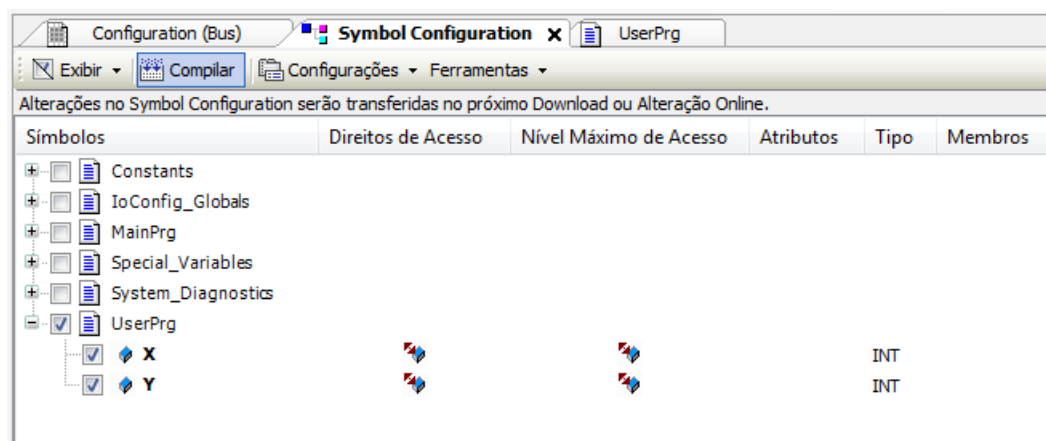


Figura 84: Selecionando Variáveis na Symbol Configuration

Após este procedimento o projeto pode ser carregado em um CP e as variáveis selecionadas estarão disponíveis para comunicação com o Servidor OPC DA. Se a tela do Objeto Symbol Configuration estiver aberta e alguma das variáveis, POU ou GVLs selecionadas for alterada, os nomes destes objetos aparecerão na cor vermelha. As situações nas quais isso acontece é caso a variável seja deletada ou o valor do atributo tenha sido modificado.

Também é possível configurar quais variáveis estarão disponíveis no Servidor OPC DA através de um atributo inserido diretamente nas POU ou GVLs onde as variáveis são declaradas. Quando o atributo *attribute 'symbol'* está presente na declaração das variáveis, podendo estar antes da definição do nome da POU ou GVL, ou para cada variável individualmente, estas são enviadas diretamente para o objeto *Symbol Configuration*, as quais são apresentadas com um símbolo na coluna *Atributos*. Antes de carregar o projeto neste caso é necessário executar o comando *Compilar* dentro do objeto *Symbol Configuration*.

As sintaxes válidas para uso do atributo são:

- *attribute 'symbol' := 'none'* – quando o valor do atributo for igual a *'none'* as variáveis não serão disponibilizadas para o Servidor OPC DA e não serão exibidas na tela do objeto *Symbol Configuration*.
- *attribute 'symbol' := 'read'* - quando o valor do atributo for igual a *'read'* as variáveis serão disponibilizadas para o Servidor OPC DA com direito de acesso somente de leitura.
- *attribute 'symbol' := 'write'* - quando o valor do atributo for igual a *'write'* as variáveis serão disponibilizadas para o Servidor OPC DA com direito de acesso somente de escrita.
- *attribute 'symbol' := 'readwrite'* – quando o valor do atributo for igual a *'readwrite'* as variáveis serão disponibilizadas para o Servidor OPC DA com direito de acesso de leitura e escrita.

No exemplo a seguir de declaração de variáveis, a configuração das variáveis A e B permite que um Servidor OPC DA acesse as mesmas com direito de acesso para leitura e escrita. Em contraponto a variável C não pode ser acessada, enquanto a variável D é acessada com direito de acesso apenas para leitura.

```
{attribute 'symbol' := 'readwrite'}  
PROGRAM UserPrg  
VAR  
A: INT;  
B: INT;  
{attribute 'symbol' := 'none'}  
C: INT;  
{attribute 'symbol' := 'read'}  
D :INT;  
END_VAR
```

Quando uma variável diferente dos tipos básicos é definida, o uso do atributo deve ser feito dentro da declaração desta DUT e não somente no contexto onde a variável é declarada. Por exemplo, no caso de uma instância DUT declarada dentro de uma POU ou GVL que possuem um atributo, este não irá impactar no comportamento dos elementos da instância desta DUT. Será necessário aplicar o mesmo nível de acesso na declaração da DUT.

ATENÇÃO

As configurações dos símbolos que serão disponibilizados ao Servidor OPC DA são armazenadas dentro do projeto do CP. Ao modificar estas configurações é necessário carregar a aplicação no CP para que seja possível acessar estas variáveis.

ATENÇÃO

Quando uma variável é excluída do projeto e carregada no CP desmarcando a mesma do objeto *Symbol Configuration*, a variável não pode mais ser lida com o Cliente OPC DA. Se a variável for adicionada ao projeto novamente, com o mesmo nome e o mesmo contexto, e inserida no objeto *Symbol Configuration*, será necessário reinicializar o Cliente OPC DA para atualizar a referência do endereço da variável, que passa a ser criada em outra área de memória no CP.

5.5.10.2. Configurando um CP no Servidor OPC DA

A configuração de um CP é executada dentro do MasterTool IEC XE através da opção disponível no menu *Comunicação*. É necessário que o MasterTool IEC XE seja executado como administrador.

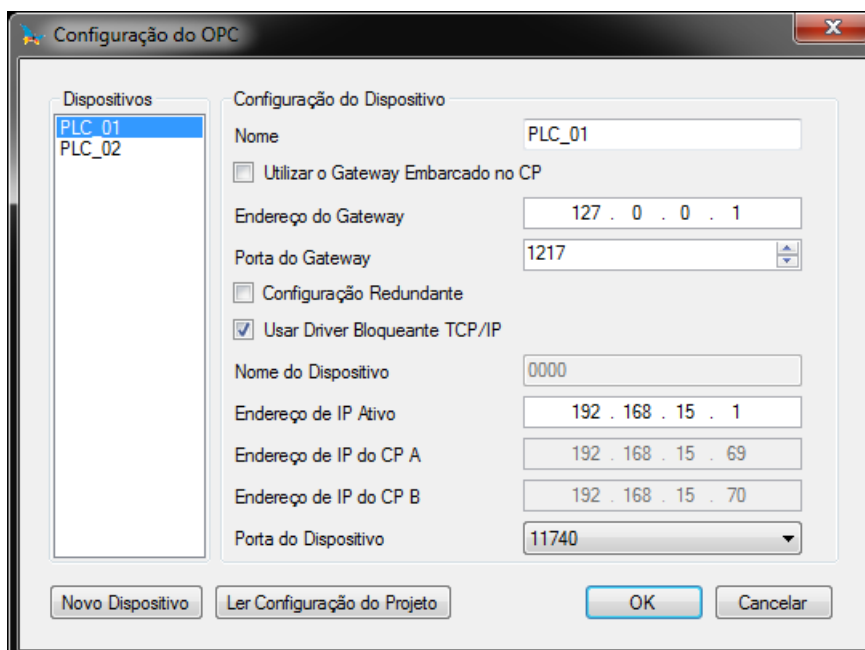


Figura 85: Configuração do OPC DA Servidor

A *Configuração do Gateway* é a mesma configurada no Gateway utilizado para comunicação entre o MasterTool IEC XE e o CP e descrita em Configurações de Comunicação, presente no Manual de Utilização MasterTool IEC XE – MU299048. Se a configuração utilizada for *localhost* o *Endereço do Gateway* deve ser preenchido com 127.0.0.1. Esta configuração é necessária, pois o Servidor OPC utiliza o mesmo gateway de comunicação e o mesmo protocolo utilizados na comunicação entre CP e MasterTool IEC XE.

Existe a opção *Utilizar o Gateway Embarcado no CP* que pode ser selecionada quando se deseja utilizar o Gateway que fica no próprio CP. Esta opção pode ser empregada para otimizar a comunicação, pois ela evita o excesso de tráfego através de uma determinada estação, quando mais de uma estação, com Cliente OPC DA, esteja conectada ao mesmo CP.

Para a configuração do CP, são possíveis dois tipos de configuração, conforme a seleção do checkbox *Usar Driver Bloqueante TCP/IP*. Quando a opção não está selecionada o nome do CP deve ser colocado no campo *Nome do Dispositivo*. Este é o nome exibido pelo CP selecionado como ativo na tela de *Configurações de Comunicação*.

A outra opção é usar o *Endereço de IP* das Interfaces Ethernet. O mesmo endereço configurado nas telas de configuração deve ser colocado neste campo. Além disso, quando for utilizado este método deve ser colocado o número da porta 11740. A confirmação irá salvar as configurações do Servidor OPC DA.

Configuração	Descrição	Padrão	Opções
Nome	Descrição do CP dentro do arquivo de configuração do Servidor OPC DA. Este campo pode ter qualquer nome, mas para organização recomenda-se utilizar o nome do projeto carregado no CP.	'PLC01'	O campo é uma STRING e podem ser colocados caracteres alfanuméricos (letras e números) e o caractere “_”. Não é permitido iniciar a STRING com números ou “_”. Permite até 49 caracteres.
Endereço do Gateway	Endereço IP do computador onde está instalado o Servidor OPC DA para os casos onde todos os CPs estejam na mesma subrede. Caso exista algum CP em outra subrede deve ser especificado o Gateway utilizado nesta subrede.	127.0.0.1	0.0.0.0 a 255.255.255.255

Configuração	Descrição	Padrão	Opções
Porta do Gateway	Porta TCP para a conexão com o Gateway.	1217	2 a 65534
Nome do Dispositivo	É o nome do CP exibido na aba <i>Configurações de Comunicação</i> do <i>Device</i> . O nome é a STRING antes do valor em hexadecimal que esta entre []. Somente habilitado quando o checkbox <i>Usar Driver Bloqueante TCP/IP</i> não está selecionado.	'0000'	O campo é uma STRING e podem ser colocados quaisquer caracteres assim como é feito na configuração do nome do CP na aba <i>Configurações de Comunicação</i> do <i>Device</i> . Permite até 49 caracteres.
Endereço de IP Ativo	Endereço IP do CP. Somente habilitado quando o checkbox <i>Usar Driver Bloqueante TCP/IP</i> está selecionado. É utilizado somente quando a configuração não for redundante.	192.168.15.1	0.0.0.0 a 255.255.255.255
Endereço de IP do CP A	Endereço IP do CPA. Somente habilitado quando a configuração for redundante. É o endereço do CP primário com o qual o servidor irá comunicar caso não exista falha.	192.168.15.69	0.0.0.0 a 255.255.255.255
Endereço de IP do CP B	Endereço IP do CPB. Somente habilitado quando a configuração for redundante. É o endereço do CP secundário com o qual o servidor irá comunicar caso ocorra uma falha.	192.168.15.70	0.0.0.0 a 255.255.255.255
Porta do Dispositivo	Porta TCP. Somente habilitado quando o checkbox <i>Usar Driver Bloqueante TCP/IP</i> está selecionado.	11740	11740 ou 11739

Tabela 130: Parâmetros de Configuração de Cada CP para o Servidor OPC DA

Quando um novo CP precisar ser configurado no Servidor OPC DA basta pressionar o botão *Novo Dispositivo* que a configuração será criada. Sempre que a tela de configuração for acessada será exibida uma lista com todos os CPs já configurados no Servidor OPC DA. As configurações existentes podem ser editadas selecionando o CP na lista *Dispositivos* e editando os parâmetros. As configurações de CPs que não são mais utilizadas podem ser excluídas. O número máximo de CPs configurados em um Servidor OPC DA é 16.

Caso a arquitetura de automação utilizada preveja que o servidor OPC DA deve ser executado em um computador onde não é executada a comunicação com o CP via MasterTool IEC XE, a ferramenta deve ser instalada neste computador para permitir a configuração do Servidor OPC DA da mesma maneira como é feito nas outras situações.

ATENÇÃO

Para armazenar a configuração do Servidor OPC DA, o MasterTool IEC XE precisa ser executado com direitos de administrador no Sistema Operacional. Dependendo da versão do Sistema Operacional este direito deve ser autorizado ao executar o programa. Para essa operação clique com o botão direito sobre o executável do MasterTool IEC XE e escolha a opção *Executar como administrador*.

ATENÇÃO

As configurações de um CP no Servidor OPC DA não são armazenadas no projeto criado no MasterTool IEC XE. Por esta razão podem ser realizadas com um projeto aberto ou fechado. As configurações são armazenadas em um arquivo de configuração onde o Servidor OPC DA está instalado. Quando alterar as configurações não é necessária carga de aplicação no CP, mas dependendo do Cliente OPC DA é possível que seja necessário conectar novamente ao Servidor ou carregar as configurações para que os dados sejam atualizados corretamente.

5.5.10.2.1. Importando uma Configuração do Projeto

Utilizando o botão *Ler Configuração do Projeto*, conforme a Figura 85, é possível atribuir a configuração do projeto aberto à configuração do CP que está em edição. Para que esta opção funcione corretamente deve existir um projeto aberto e deve ser definido um *Caminho Ativo* conforme descrito em *Configurações de Comunicação*, presente no Manual de Utilização MasterTool IEC XE – MU299048. Caso alguma destas condições não seja atendida será exibida uma mensagem de erro e nenhum dado será modificado.

Quando as condições anteriores são válidas, as configurações do CP recebem os parâmetros do projeto aberto. As informações de *Endereço de IP* e *Porta do Gateway* são configuradas conforme descrito em *Configurações de Comunicação* de acordo com o *Caminho Ativo*. Entretanto, as configurações de *Endereço de IP* são lidas das configurações da interface Ethernet NET1. A porta para conexão com o CP é sempre atribuída neste caso como 11740.

5.5.10.3. Configuração com CP no Servidor OPC DA com Redundância de Conexão

É possível configurar o Servidor OPC DA para que este opere com redundância de conexão. Desta forma o Servidor OPC DA irá se comunicar preferencialmente com um CP, mas quando, por alguma razão, não conseguir estabelecer uma comunicação com este CP um segundo CP também configurado será acessado. Esta configuração é especialmente importante para a comunicação de sistemas SCADA com os CPs da Série Nexto que utilizam redundância de Half-Cluster, onde existe um CP em estado ativo executando o processo e outro CP em estado reserva apto a assumir o controle do processo quando ocorrer algum tipo de falha.

A configuração do projeto nestes casos é semelhante a descrita em [Criando um Projeto para Comunicação OPC DA](#). Contudo, quando um Projeto é criado com Redundância de Half-Cluster e a comunicação com o sistema de supervisão se dará através do Servidor OPC DA, é necessário selecionar a opção da *Configuração de comunicação OPC DA* como habilitada durante o Assistente de criação de projeto do MasterTool IEC XE. Ao habilitar esta opção o projeto criado terá o código necessário para funcionamento da comunicação com redundância de conexão OPC DA.

No caso redundante, uma variável é declarada dentro da POU, chamada *NonSkippedPrg*. Esta POU é executada em ambos os CPs, independente do estado de redundância. Dentro desta POU é declarada uma variável do tipo BOOL, utilizada para o controle da conexão com o Servidor OPC DA chamada *OPCRedundancyActive*. Esta variável pode ser acessada de qualquer ponto da aplicação, através de todo o contexto, ou seja, *Application.NonSkippedPrg.OPCRedundancyActive*. Ela é declarada dentro do objeto *Symbol Configuration* com direito apenas de leitura por parte do SCADA. Quando o valor da variável for igual a TRUE os dados são lidos através da conexão com este CP. Desta forma toda vez que ocorre uma troca de estado entre os CPs a variável tem seu estado alterado, permanecendo no estado TRUE no CP que está no estado ativo de redundância.

O código do programa *NonSkippedPrg* é o seguinte, em linguagem ST:

```

PROGRAM NonSkippedPrg
VAR
  {attribute 'symbol' := 'read'}
  OPCRedundancyActive : BOOL;
END_VAR

IF fbRedundancyManagement.m_fbDiagnosticsLocal.eRedState = REDUNDANCY_STATE.
  ACTIVE THEN
  OPCRedundancyActive := TRUE;
ELSE
  OPCRedundancyActive := FALSE;
END_IF

```

O código do programa *NonSkippedPrg* pode ser editado tomando-se o cuidado de manter o código acima sem alterações. Este código testa o estado da redundância e preenche uma variável do tipo `BOOL` chamada *OPCRedundancyActive*, em função deste estado. Caso o CP seja o Ativo, o valor da variável será `TRUE`, caso contrário será `FALSE`. Esta variável recebe o *attribute* `'symbol' := 'read'` para permitir que o Servidor OPC DA acesse o seu conteúdo e defina de onde a informação deve ser lida.

Caso se decida adicionar comunicação OPC DA após um projeto ter sido criado, é possível configurar o OPC DA adicionando o código anterior no programa *NonSkippedPrg* e adicionando o objeto *Symbol Configuration* ao projeto.

Para a configuração do CP redundante no Servidor OPC DA é necessário selecionar a opção *Configuração Redundante* na tela de configuração conforme exibido na Figura 85. Quando esta opção é selecionada, será sempre utilizada a opção *Usar Driver Bloqueante TCP/IP*. Além disso, serão habilitados os campos *Endereço de IP do CPA* e *Endereço de IP do CPB* conforme descrito na Tabela 130. Estes *Endereços de IP* são os mesmos configurados nas interfaces Ethernet dentro do projeto do CP com redundância de Half-Cluster. Para facilitar a configuração quando um projeto redundante estiver aberto, o botão *Le Configuração do Projeto* pode ser utilizado.

ATENÇÃO

A redundância de conexão do Servidor OPC DA é realizada por apenas um Servidor. Nos casos em que se deseje uma maior disponibilidade dos dados para os sistemas de supervisão deve ser usada uma arquitetura com Servidores SCADA redundantes. Nestes casos não é necessária nenhuma configuração no Servidor OPC DA. Consulte as documentações do sistema SCADA para verificar quais as configurações necessárias para o funcionamento da arquitetura redundante.

5.5.10.4. Variáveis de Status e Qualidade da Comunicação OPC DA

Para cada um dos CPs criados no Servidor OPC DA são geradas variáveis de status chamadas de *_CommState* e *_CommStateOK*. A variável *_CommState* indica o estado da comunicação do Servidor OPC DA com CP. Este estado pode ser interpretado pelo Cliente OPC DA conforme a tabela abaixo.

Estado	Valor	Descrição
STATE_TERMINATE	-1	Se a comunicação do Servidor OPC DA com o Cliente OPC DA for encerrada este valor será retornado. Quando houver mais de um Cliente OPC DA conectado ao mesmo tempo o retorno ocorrerá na desconexão do último Cliente conectado. Este status apesar de estar na variável não pode ser visualizado, pois só é alterado quando não existe mais conexão com um cliente.
STATE_PLC_NOT_CONNECTED	0	O CP configurado no Servidor OPC DA não está conectado. Pode acontecer caso a configuração esteja incorreta (Endereço de IP do CP e/ou Gateway errados) ou o CP não esteja disponível naquele momento.

Estado	Valor	Descrição
STATE_PLC_CONNECTED	1	O CP configurado no Servidor OPC DA está conectado. Este é um estado transitório durante conexão.
STATE_NO_SYMBOLS	2	Não existem símbolos (variáveis) disponíveis no CP configurado no Servidor OPC DA. Pode acontecer na situação de não existirem símbolos ou ainda quando não houver um projeto carregado no CP.
STATE_SYMBOLS_LOADED	3	Finalizado o processo de leitura dos símbolos (variáveis) do CP configurado no Servidor OPC DA. Este é um estado transitório durante conexão.
STATE_RUNNING	4	Após leitura dos símbolos (variáveis) o Servidor OPC DA está executando a atualização periódica dos valores dos símbolos disponíveis em cada CP configurado.
STATE_DISCONNECT	5	Ocorreu desconexão com o CP configurado no servidor OPC DA.
STATE_NO_CONFIGURATION	6	Quando a configuração do OPC DA (armazenada em um arquivo OPCServer.ini) possuir uma sintaxe errada este será o status da variável. De maneira geral, este comportamento não é observado pois o MasterTool IEC XE mantém esta configuração íntegra.

Tabela 131: Descrição dos estados da comunicação do Servidor OPC DA com o CP

A variável *_CommStateOK* é uma variável do tipo BOOL que indica se a comunicação está funcionando entre o Servidor OPC DA e o CP. Quando o valor é TRUE, indica que a comunicação está funcionando corretamente. Se o valor for FALSE não é possível comunicar por alguma razão com o CP.

Além de monitorar o estado da comunicação, o Cliente OPC DA pode acessar informações da qualidade de comunicação. Os bits de qualidade formam um byte. Eles estão divididos em três grupos de bits: *Qualidade*, *Substatus* e *Limite*. Os bits estão distribuídos da seguinte forma *QQSSSLL*, onde *QQ* representa os bits de *Qualidade*, *SSSS* os bits de *Substatus* e *LL* os bits de *Limite*. Neste caso os bits *QQ* são os mais significativos no byte, enquanto os bits *LL* são os menos significativos.

QQ	Valor dos Bits	Definição	Descrição
0	00SSSLL	Bad	O valor lido não pode ser usado porque existe algum problema na conexão. É possível monitorar o valor de <i>_CommState</i> e diagnosticar o problema.
1	01SSSLL	Uncertain	A qualidade não pode ser definida e pode ser apresentada no campo <i>Substatus</i> .
2	10SSSLL	NA	Este valor é reservado e não é utilizado pelo padrão OPC DA.
3	11SSSLL	Good	A qualidade é boa e o valor lido pode ser usado.

Tabela 132: Descrição do valor da Qualidade OPC DA

A Tabela 132 apresenta os valores possíveis de qualidade. O Servidor OPC DA retorna apenas valor com Qualidade *Good* e *Bad*. Um Cliente OPC DA pode manter a qualidade como *Uncertain* em caso de alguma falha na qual ele não consiga uma conexão com o Servidor. No caso de monitoração dos 8 bits de qualidade diretamente do Servidor OPC DA os campos *Substatus* e *Limite* serão nulos e a Qualidade *Good* será representada com o valor 192 e a qualidade *Bad* com o valor 0.

5.5.10.5. Limites da Comunicação com Servidor OPC DA

A tabela abaixo apresenta os limites de configuração do Servidor OPC DA.

Número máximo de variáveis comunicando com um único CP	-
Número máximo de CPs em um Servidor OPC DA	16
Número máximo de conexões simultâneas de Servidor OPC DA em um mesmo CP	8

Tabela 133: Limites da Comunicação com o Servidor OPC DA

Nota:

Número máximo de variáveis comunicando com um único CP: Não há limite de configuração. A quantidade máxima possível de variáveis depende da capacidade de processamento do dispositivo.

ATENÇÃO

O número máximo de conexões simultâneas de Servidor OPC DA em um mesmo CP é compartilhado com as conexões realizadas com o MasterTool IEC XE. Ou seja, a soma de conexões com Servidor OPC DA e MasterTool IEC XE não deve ultrapassar o número máximo definido na Tabela 133.

A comunicação entre o Servidor OPC DA e o CP utiliza o mesmo protocolo utilizado para comunicação do MasterTool IEC XE com o CP. Este protocolo só está disponível para as interfaces Ethernet das UCPs da Série Nexto, não sendo possível estabelecer este tipo de comunicação com módulos de expansão Ethernet.

Quando uma comunicação é estabelecida entre o Servidor OPC DA e o CP estes dois elementos iniciam uma série de transações que visam resolver o endereço de cada variável declarada, otimizando a comunicação em regime de leitura de dados. Além disso, nesta fase também são resolvidas as classificações dos grupos de comunicação usados por alguns Clientes com o intuito de otimizar a comunicação. Este processo inicial demanda algum tempo e depende da quantidade de variáveis mapeadas e da capacidade de processamento do dispositivo.

5.5.10.6. Acessando Dados Através de um Cliente OPC DA

Após a configuração do Servidor OPC DA os dados disponíveis em todos os CPs podem ser acessados via um Cliente OPC DA. Na configuração do Cliente OPC DA deve ser selecionado o nome do Servidor OPC DA correto. Neste caso o nome é *CoDeSys.OPC.DA*. A figura abaixo exibe a seleção do servidor no driver cliente do software SCADA BluePlant.

ATENÇÃO

Da mesma forma que o MasterTool IEC XE algumas ferramentas precisam ser executadas com direitos de administrador no Sistema Operacional para o correto funcionamento do Cliente OPC DA. Dependendo da versão do Sistema Operacional este direito deve ser autorizado ao executar o programa. Para essa operação clique com o botão direito sobre o executável da ferramenta e escolha a opção *Executar como administrador*.

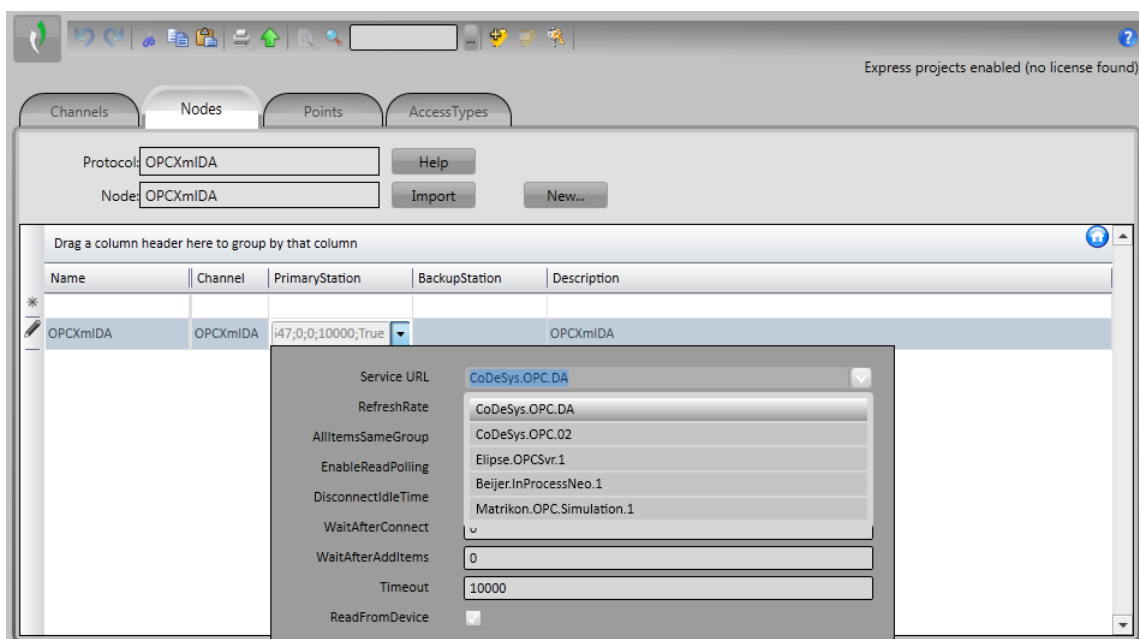


Figura 86: Selecionando Servidor OPC DA na Configuração do Cliente

Nos casos em que o servidor se encontra remotamente pode ser necessário adicionar o caminho da rede ou o endereço de IP do computador onde se encontra o servidor instalado. Nestes casos existem duas opções de configuração. A primeira delas é configurar diretamente para isso sendo necessário liberar os Serviços de COM/DCOM do Windows. Contudo, uma forma mais simples é utilizar uma ferramenta de tunneller que abstrai as configurações de COM/DCOM, além de possibilitar uma comunicação mais segura entre o Cliente e o Servidor. Para mais informações sobre este tipo de ferramenta consultar a *NAP151 - Utilização do Tunneller OPC*.

Uma vez que o Cliente se conecta no Servidor podem ser usados comandos de importação de TAGs. Estes comandos consultam informações declaradas no CP, retornando uma lista com todos os símbolos disponibilizados por este.

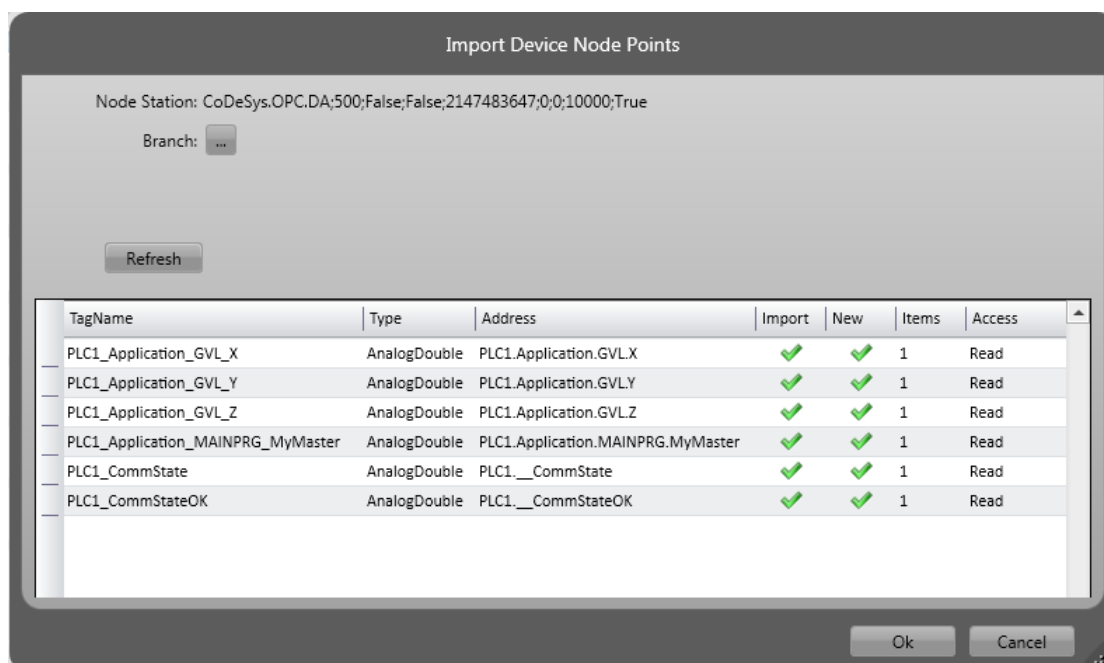


Figura 87: Lista de Símbolos Consultados pelo Cliente OPC DA

A lista de variáveis selecionadas será incluída na lista de comunicações do Cliente e podem ser utilizadas, por exemplo, em telas de um sistema SCADA.

ATENÇÃO

O modo de simulação do software MasterTool IEC XE pode ser utilizado para testes da comunicação OPC DA. As informações sobre como configurá-lo estão presentes na seção *Testando a Comunicação OPC com o Uso do Simulador*, do Manual de Utilização MasterTool IEC XE – MU299048.

5.5.11. OPC UA Servidor

O protocolo OPC UA é uma evolução da família OPC. Independente de plataforma, foi concebido para ser o novo padrão utilizado nas comunicações industriais.

Baseado na arquitetura cliente/servidor, o protocolo OPC UA oferece inúmeras vantagens no desenvolvimento de projeto e facilidades na comunicação com os sistemas de automação.

Quando se trata do desenvolvimento de projetos, configurar a comunicação e trocar informações entre os sistemas é extremamente simples utilizando tecnologia OPC UA. Utilizando outros drivers, baseados em endereços, é necessário criar tabelas para relacionar as tags do sistema de supervisão e as variáveis do controlador programável. Quando as áreas de dados são alteradas, no decorrer do desenvolvimento do projeto, é necessário refazer os mapeamentos e novas tabelas com as relações entre as informações do CP com o sistema SCADA.

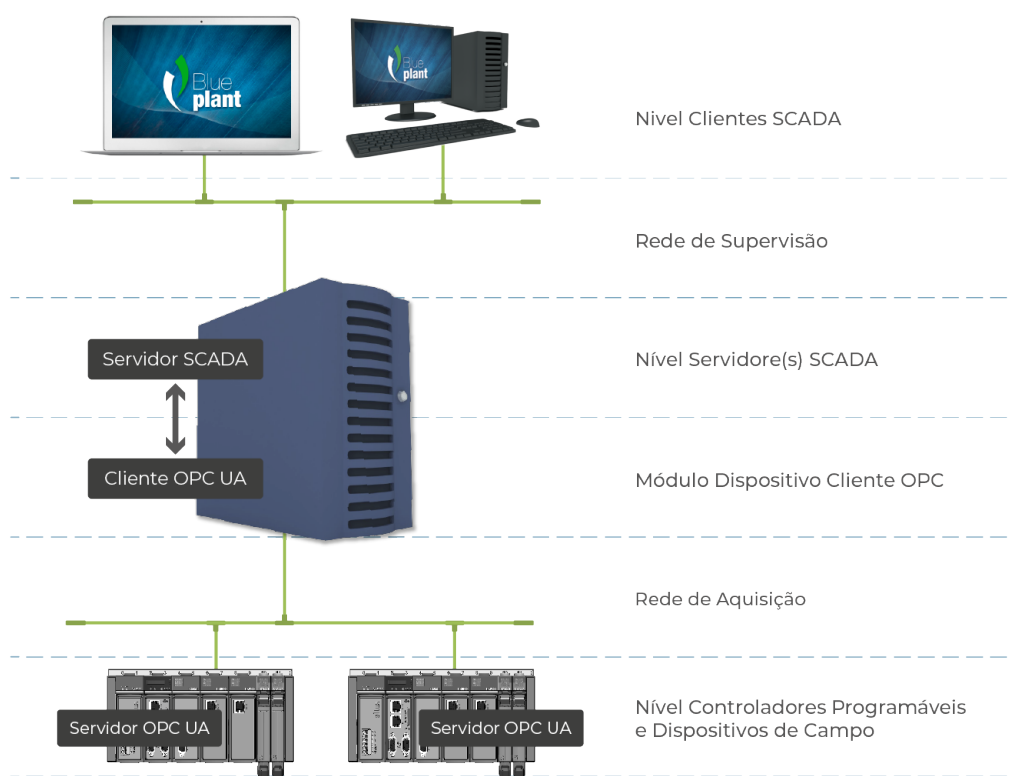


Figura 88: Arquitetura Típica OPC UA

A figura acima apresenta uma arquitetura típica para comunicação de sistema SCADA e CPs em projeto de automação. Todos os papéis presentes na comunicação estão explícitos nesta figura independente do local onde estejam executando, eles podem estar em um mesmo equipamento ou em equipamentos diferentes. Cada um dos papéis desta arquitetura é descrito na tabela abaixo.

Papel	Descrição
Nível Controladores Programáveis e Dispositivos de Campo	Os dispositivos de campo e os CPs são os dispositivos nos quais as informações do estado de operação e controle da planta são armazenadas. Os sistemas SCADA acessam as informações nestes dispositivos e armazenam nos Servidores SCADA para consulta pelos Clientes SCADA durante a operação da planta.
Módulo Servidor OPC UA	O Servidor OPC UA é um módulo interno dos CPs responsável por receber as requisições OPC UA e traduzi-las para a comunicação com os dispositivos de campo.
Rede de Aquisição	A rede de aquisição é a rede na qual trafegam as mensagens OPC UA para solicitar os dados que são coletados dos CPs e dispositivos de campo.
Módulo Cliente OPC UA	O módulo Cliente OPC UA, que faz parte do Servidor SCADA, é responsável por fazer requisições aos Servidores OPC UA utilizando o protocolo OPC UA. Os dados coletados por ele são armazenados na base de dados do Servidor SCADA.
Nível Servidor SCADA	O Servidor SCADA é responsável por se conectar aos diversos dispositivos de comunicação e armazenar os dados coletados destes dispositivos em uma base de dados para que possam ser consultados pelos Clientes SCADA.
Rede de Supervisão	A rede de supervisão é a rede pela qual os Clientes SCADA estão conectados aos Servidores SCADA, muitas vezes utilizando um protocolo proprietário do sistema SCADA específico. Em uma topologia na qual não se usa diversos Clientes ou que o Servidor e o Cliente estejam instalados em um mesmo equipamento, não existe este tipo de rede, e neste caso este equipamento deve utilizar diretamente o protocolo OPC UA para comunicação com o CP.
Nível Clientes SCADA	Os clientes SCADA são responsáveis por solicitar aos servidores SCADA os dados necessários para exibir em uma tela onde é executada a operação de uma planta. Através deles é possível executar leituras e escritas em dados armazenados na base de dados do Servidor SCADA.

Tabela 134: Descrição dos Papéis em uma Arquitetura com Servidor OPC UA

Ao utilizar o protocolo OPC UA, a relação entre as tags dos sistemas de supervisão e os dados do processo nas variáveis do controlador é totalmente transparente. Isso significa que se as áreas de dados são alteradas no decorrer do desenvolvimento do projeto, não há a necessidade de refazer relações entre as informações do CP com o SCADA. Basta utilizar a nova variável disponibilizada pelo CP nos sistemas que requisitam esse dado.

O uso do OPC UA oferece maior produtividade e conectividade com os sistemas SCADA. Contribui na redução do tempo de desenvolvimento de aplicações e nos custos com manutenção. Possibilita, ainda, inserção de novos dados na comunicação de forma simplificada com maior flexibilidade e interoperabilidade entre os sistemas de automação por ser um padrão aberto.

Vale salientar que o OPC UA está disponível somente nas interfaces Ethernet locais das UCPs Nexto. Os módulos de expansão Ethernet não suportam essa funcionalidade.

5.5.11.1. Criando um Projeto para Comunicação OPC UA

Os passos para criação de um projeto com OPC UA são muito similares aos passos descritos na seção [Criando um Projeto para Comunicação OPC DA](#). Assim como no protocolo OPC DA, a configuração do protocolo OPC UA está baseada na configuração da *Symbol Configuration*. Para habilitar o OPC UA basta habilitar a opção *Suporte a característica OPC UA* na configuração, conforme ilustrado na figura abaixo.

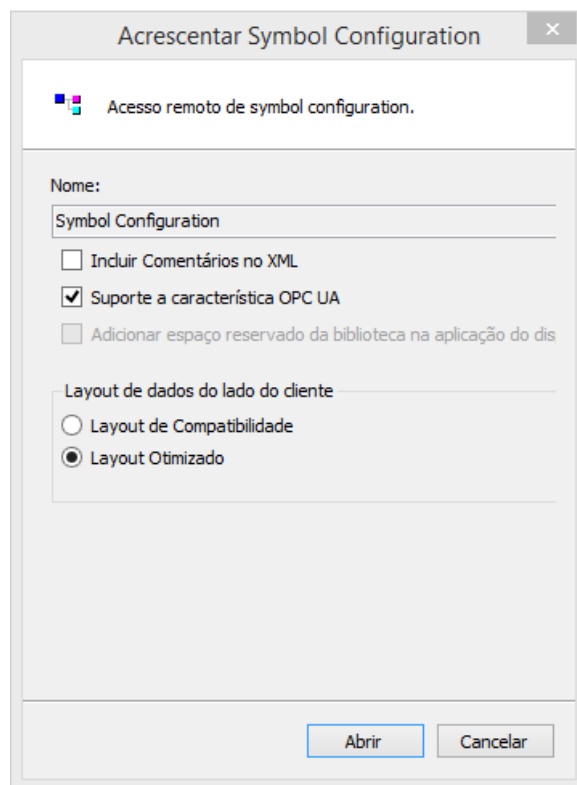


Figura 89: Objeto Symbol Configuration

ATENÇÃO

Ao ativar o suporte ao protocolo OPC UA, o suporte ao protocolo OPC DA continua habilitado. É possível habilitar as comunicações OPC UA e OPC DA ao mesmo tempo, para reportar as variáveis configuradas no objeto *Symbol Configuration* ou via atributos.

Outro caminho para acessar esta configuração, após já criado um projeto com o objeto *Symbol Configuration*, se dá acessando o menu *Configurações* da aba de configuração da *Symbol Configuration*. Basta selecionar a opção *Suporte a características OPC UA* para habilitar o suporte ao protocolo OPC UA, conforme ilustrado na figura abaixo.

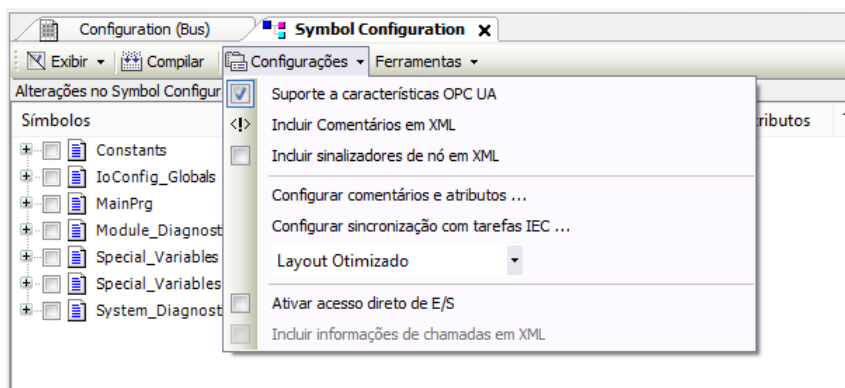


Figura 90: Habilitando OPC UA no Objeto Symbol Configuration

Após este procedimento o projeto pode ser carregado em um CP e as variáveis selecionadas estarão disponíveis para comunicação com o Servidor OPC UA.

5.5.11.2. Tipos de Variáveis Suportadas

Esta seção define os tipos de variáveis que suportam comunicação via protocolo OPC UA, quando declaradas dentro de GVLs ou POU's e selecionadas no objeto *Symbol Configuration* (ver seção anterior).

Os seguintes tipos de variáveis simples são suportados:

- BOOL
- SINT
- USINT / BYTE
- INT
- UINT / WORD
- DINT
- UDINT / DWORD
- LINT
- ULINT / LWORD
- REAL
- LREAL
- STRING
- TIME
- LTIME

É possível também utilizar tipos estruturados (STRUCTs ou Blocos Funcionais) criados a partir dos tipos simples anteriores.

Finalmente, também é possível criar arrays de tipos simples ou de tipos estruturados.

5.5.11.3. Limite de Clientes Conectados no Servidor OPC UA

O número máximo de clientes OPC UA conectados simultaneamente num CP é 8 (oito).

5.5.11.4. Limite de Variáveis de Comunicação no Servidor OPC UA



Não há limite de configuração. A quantidade máxima possível de variáveis depende da capacidade de processamento do dispositivo.


Quando uma comunicação é estabelecida entre o Servidor OPC UA e o CP, estes dois elementos iniciam uma série de transações que visam resolver o endereço de cada variável declarada, otimizando a comunicação em regime de leitura de dados. Além disso, nesta fase também são resolvidas as classificações dos grupos de comunicação usados por alguns Clientes com o intuito de otimizar a comunicação. Este processo inicial demanda algum tempo e depende da quantidade de variáveis mapeadas e da capacidade de processamento do dispositivo.

5.5.11.5. Configurações de Criptografia



Se desejado, o usuário pode configurar criptografia para a comunicação OPC UA usando o perfil *Basic256SHA256*, para obter uma conexão segura (segurança cibernética).

Para configurar a criptografia num servidor OPC UA deve-se criar um certificado para o mesmo, executando os seguintes passos no programador Mastertool:

1. Definir um caminho ativo para comunicação com o controlador (não é necessário fazer login);
2. No menu *Visualizar*, selecionar *Tela de Segurança*;
3. Clicar na aba *Devices* no lado esquerdo desta tela;
4. Clicar no ícone  para executar um refresh;
5. Clicar no ícone *Device*, abaixo do qual se abrirão diversas pastas de certificados (*Own Certificates*, *Trusted Certificates*, *Untrusted Certificates*, *Quarantined Certificates*);
6. Clicar no ícone  para gerar um certificado e selecione os seguintes parâmetros:
 - *Key length* (bit): 3072
 - *Validity period* (days): 365 (pode ser modificado se desejado)
7. Aguarde enquanto o certificado é calculado e transferido para o controlador (isso pode levar alguns minutos);
8. Reinicialize (desligue e religue) o controlador.

9. No cliente OPC UA, execute os procedimentos necessários para se conectar ao servidor OPC UA e gerar um certificado com o perfil *Basic256Sha256* (ver manual do cliente OPC UA específico para detalhes);
10. De volta ao Mastertool, clique no ícone  da *Tela de Segurança* para executar um refresh;
11. Na *Tela de Segurança*, selecione a pasta "*Quarantined Certificates*" abaixo do *Device*. No painel direito deve-se observar um certificado solicitado pelo cliente OPC UA;
12. Arraste este certificado para a pasta "*Trusted Certificates*";
13. Prossiga as configurações no cliente OPC UA (ver manual do cliente OPC UA específico para detalhes).

Para remover a criptografia previamente configurada num controlador, deve-se seguir o seguinte procedimento:

1. Definir um caminho ativo para comunicação com o controlador (não é necessário fazer login);
2. No menu *Visualizar*, selecionar *Tela de Segurança*;
3. Clicar na aba *Devices* no lado esquerdo desta tela;
4. Clicar no ícone  para executar um refresh;
5. Clicar no *Device*, abaixo do qual se abrem diversas pastas de certificados (*Own Certificates*, *Trusted Certificates*, *Untrusted Certificates*, *Quarantined Certificates*);
6. Clicar na pasta "*Own Certificates*", e no painel direito selecionar o certificado (OPC UA Server);
7. Clicar no ícone  para remover este certificado do projeto e do controlador;
8. Reinicialize (desligue e religue) o controlador.

5.5.11.6. Principais Parâmetros de Comunicação Ajustados em um Cliente OPC UA

Alguns parâmetros de comunicação OPC UA são configurados no cliente OPC UA, e negociados com o servidor OPC UA no momento em que a conexão entre ambos é estabelecida. As próximas subseções descrevem os principais parâmetros de comunicação OPC UA, seu significado, e cuidados para selecionar valores adequados para os mesmos.

Num cliente OPC UA é possível agrupar as variáveis de um servidor em diferentes *subscriptions*. Cada *subscription* é um conjunto de variáveis que são reportadas num único pacote de comunicação (*PublishResponse*) enviado do servidor para o cliente. A seleção das variáveis que compõem cada *subscription* é feita no cliente OPC UA.

ATENÇÃO

O agrupamento de variáveis em múltiplas *subscriptions* é interessante para otimizar a capacidade de processamento e consumo de banda de comunicação Ethernet. Tais aspectos de otimização são analisados com maior profundidade no manual do OPC UA Server MU214609, onde algumas regras para a composição de *subscriptions* são sugeridas. Esta manual também discute com maior profundidade diversos conceitos sobre o protocolo OPC UA.

Alguns dos parâmetros de comunicação descritos a seguir devem ser definidos para o servidor como um todo, outros para cada *subscription*, e outros para cada variável que compõe uma *subscription*.

5.5.11.6.1. Endpoint URL

Este parâmetro define o endereço IP e porta TCP do servidor, por exemplo:

opc.tcp://192.168.17.2:4840

Neste exemplo, o endereço IP do controlador é 192.168.17.2.

A porta TCP sempre deve ser 4840.

5.5.11.6.2. Publishing Interval (ms) e Sampling Interval (ms)

O parâmetro *Publishing Interval* (unidade: milissegundos) deve ser definido para cada *subscription*.

O parâmetro *Sampling Interval* deve ser definido para cada variável (unidade: milissegundos). Entretanto, em muitos clientes OPC UA o parâmetro *Sampling Interval* pode ser definido para uma *subscription*, sendo igual para todas as variáveis agrupadas na *subscription*.

Somente as variáveis de uma *subscription* cujos valores se modificaram são reportadas para o cliente através de um pacote de comunicação *PublishResponse*. O parâmetro *Publishing Interval* define o intervalo mínimo entre pacotes *PublishResponse* consecutivos da mesma *subscription*, com o objetivo de limitar o consumo de processamento e de banda de comunicação Ethernet.

Para descobrir quais variáveis da *subscription* se modificaram e devem ser reportadas para o cliente no próximo pacote *PublishResponse*, o servidor deve executar comparações, e tais comparações (*samplings*) são executadas pelo mesmo com o intervalo *Sampling Interval*. Recomenda-se que o valor do *Sampling Interval* varie entre 50% e 100% do valor do *Publishing Interval*, pois existe um consumo de processamento relativamente alto associado ao processo de comparação executado em cada *Sampling Interval*.

Pode-se dizer que a soma entre o *Publishing Interval* e o *Sampling Interval* é o retardo máximo entre a mudança de um valor no servidor e a transmissão do pacote *PublishResponse* que reporta esta mudança. A metade desta soma é o retardo médio entre a mudança de um valor no servidor e a transmissão do pacote *PublishResponse* que reporta esta mudança.

5.5.11.6.3. *Lifetime Count e Keep-Alive Count*

Estes dois parâmetros devem ser configurados para cada *subscription*.

O objetivo destes dois parâmetros é criar um mecanismo de desativação de uma *subscription* por iniciativa do servidor, caso não receba pacotes de comunicação *PublishRequest* do cliente por muito tempo para esta *subscription*. Pacotes *PublishRequest* devem ser recebidos pelo servidor para que o mesmo possa transmitir pacotes *PublishResponse* contendo as variáveis da *subscription* que mudaram seus valores.

Caso o servidor não receba pacotes *PublishRequest* por um tempo superior a *Lifetime Count* multiplicado por *Publishing Interval*, o servidor desativa a *subscription*, que deverá ser criada novamente pelo cliente no futuro se assim o desejar.

Em situações onde as variáveis de uma *subscription* não se modificam, poderia passar um longo tempo sem a transmissão de *PublishResponses* e consequentemente de *PublishRequests* que o sucedem, provocando uma desativação indesejada da *subscription*. Para evitar que isso aconteça, foi criado o parâmetro *Keep-Alive Count*. Caso não haja modificações de dados na *subscription* por um tempo igual a *Keep-Alive Count* multiplicado por *Publishing Interval*, o servidor enviará um pacote *PublishResponse* pequeno e vazio indicando que nenhuma variável mudou. Este *PublishResponse* vazio autorizará que o cliente mande imediatamente o próximo *PublishRequest*.

O valor de *Keep-Alive Count* deve ser menor do que o valor de *Lifetime Count* para evitar uma desativação indesejada da *subscription*. Sugere-se que *LifeTime Count* seja no mínimo 3 vezes maior que *Keep-Alive Count*.

5.5.11.6.4. *Queue Size e Discard Oldest*

Estes parâmetros devem ser mantidos com os seguintes valores fixos, que normalmente são os valores padrão nos clientes:

- *Queue Size*: 1
- *Discard Oldest*: enable

De acordo com a norma OPC UA, é possível definir estes parâmetros para cada variável. No entanto, muitos clientes permitem definir valores comuns para todas as variáveis configuradas numa *subscription*.

Queue Size deve ser mantido com o valor 1 pois não existe suporte a eventos nesta implementação do servidor OPC UA, e portanto é desnecessário definir uma fila. Aumentar o valor de *Queue Size* pode implicar em aumento na banda de comunicação e processamento da CPU, e isso deve ser evitado.

Discard Oldest deve ser mantido com o valor *enable*, para que o pacote *PublishResponse* sempre reporte a mudança de valor mais recente detectada para cada variável.

5.5.11.6.5. *Filter Type e Deadband Type*

Estes parâmetros devem ser mantidos com os seguintes valores fixos, que normalmente são os valores padrão nos clientes:

- *Filter Type*: *DataChangeFilter*
- *Deadband Type*: *none*

De acordo com a norma OPC UA, é possível definir estes parâmetros para cada variável. No entanto, muitos clientes permitem definir valores comuns para todas as variáveis configuradas numa *subscription*.

O parâmetro *Filter Type* deve valer *DataChangeFilter*, indicando que mudanças de valores nas variáveis devem provocar sua transmissão num pacote *PublishResponse*.

Deadband Type deve ser mantido em “*none*” porque não existe implementação de *deadbands* para variáveis analógicas. Desta forma, qualquer alteração da variável analógica, por mínima que seja, provoca sua transmissão num pacote *PublishResponse*.

Para reduzir consumo de processamento e banda de comunicação Ethernet, o usuário poderá implantar *deadbands* por sua conta da seguinte forma:

- Não incluir a variável analógica numa *subscription*;
- Ao invés disso, incluir numa *subscription* uma variável auxiliar vinculada à variável analógica;
- Copiar a variável analógica para a variável auxiliar somente quando o *deadband* gerenciado pelo usuário for extrapolado.

5.5.11.6.6. PublishingEnabled, MaxNotificationsPerPublish e Priority

Sugere-se que os seguintes parâmetros sejam mantidos com os seguintes valores, que normalmente são os valores padrão nos clientes:

- *PublishingEnabled*: true
- *MaxNotificationsPerPublish*: 0
- *Priority*: 0

Estes parâmetros devem ser configurados para cada *subscription*.

PublishingEnable deve valer “true” para que as variáveis da *subscription* sejam reportadas em caso de mudança de valor.

MaxNotificationsPerPublish indica quantas das variáveis que mudaram de valor podem ser incluídas num mesmo pacote *PublishResponse*. O valor especial “0” indica que não existe um limite para isso, e recomenda-se utilizar este valor para que todas as variáveis que mudaram sejam reportadas num mesmo pacote *PublishResponse*.

Priority indica a prioridade relativa desta *subscription* em relação a outras. Caso em determinado momento o servidor deva enviar múltiplos pacotes *PublishResponse* de *subscriptions* diferentes, priorizará aquele com o maior valor de priority. Se todas as *subscriptions* tiverem a mesma prioridade, os pacotes *PublishResponse* serão transmitidos numa sequência fixa.

5.5.11.7. Acessando Dados Através de um Cliente OPC UA

Após a configuração do Servidor OPC UA os dados disponíveis em todos os CPs podem ser acessados via um Cliente OPC UA. Na configuração do Cliente OPC UA deve ser selecionado o endereço do Servidor OPC UA correto. Neste caso o endereço *opc.tcp://endereço-ip-do-dispositivo:4840*. A figura abaixo exhibe a seleção do servidor no driver cliente do software SCADA BluePlant.

ATENÇÃO

Da mesma forma que o MasterTool IEC XE, algumas ferramentas precisam ser executadas com direitos de administrador no Sistema Operacional para o correto funcionamento do Cliente OPC UA. Dependendo da versão do Sistema Operacional este direito deve ser autorizado ao executar o programa. Para essa operação clique com o botão direito sobre o executável da ferramenta e escolha a opção *Executar como administrador*.

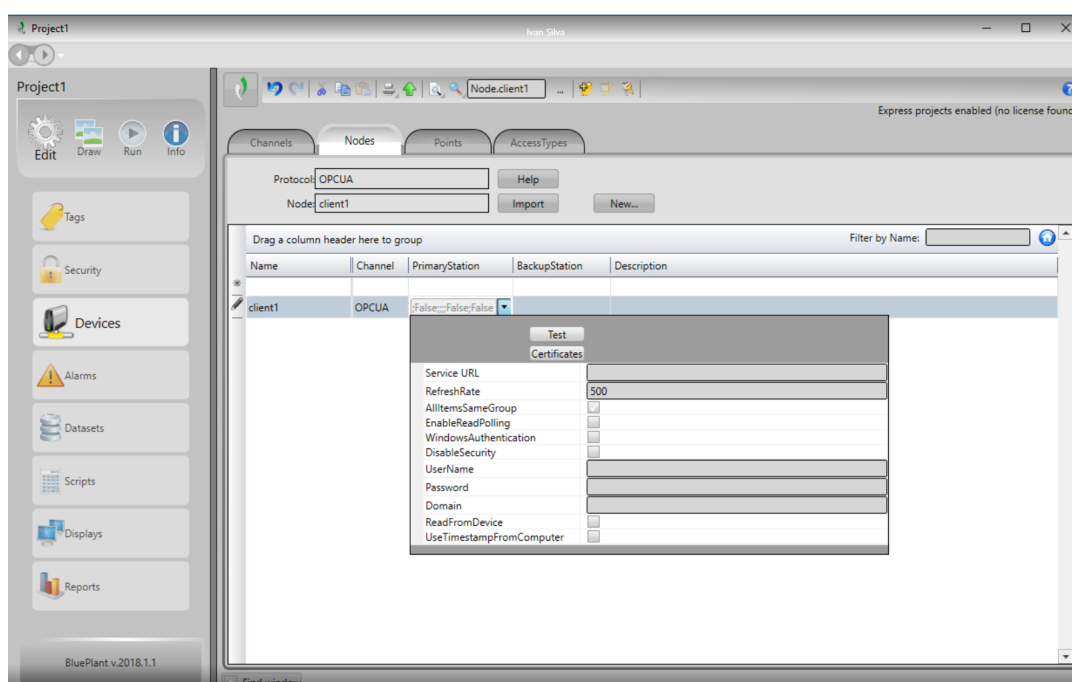


Figura 91: Selecionando Servidor OPC UA na Configuração do Cliente

Uma vez que o Cliente se conecta no Servidor podem ser usados comandos de importação de TAGs. Estes comandos consultam informações declaradas no CP, retornando uma lista com todos os símbolos disponibilizados por este.

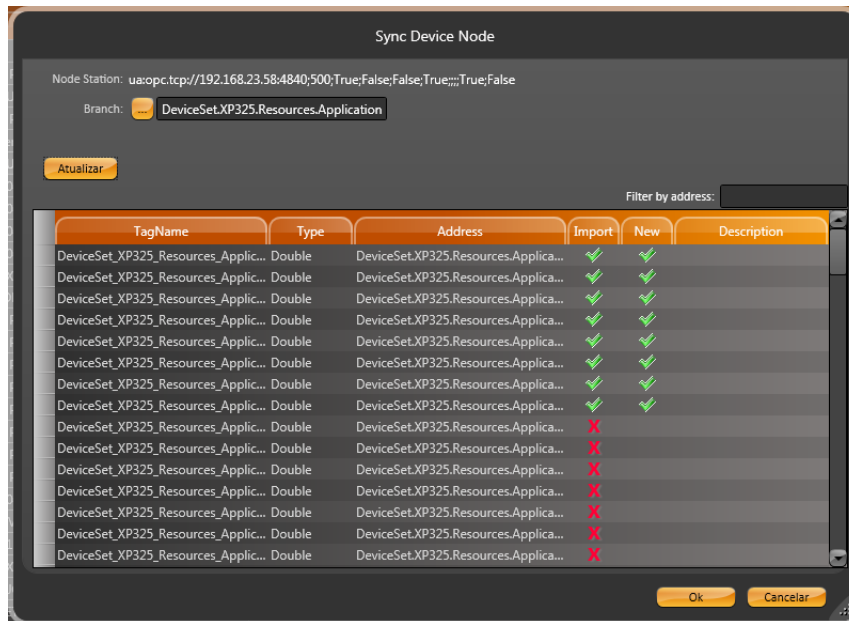


Figura 92: Lista de Símbolos Consultados pelo Cliente OPC UA

A lista de variáveis selecionadas será incluída na lista de comunicações do Cliente e podem ser utilizadas, por exemplo, em telas de um sistema SCADA.

5.5.12. Mestre EtherCAT

EtherCAT (*Ethernet Control Automation Technology*) é um protocolo com arquitetura mestre-escravo de alto desempenho, para Ethernet determinística, que permite desempenho em tempo real pois atualiza 1000 E/S distribuídas em 30 µS ou 100 eixos de servomotores a cada 100 µS usando par trançado ou cabos de fibra óptica. Além disso, este possui topologia flexível, permitindo ligações em linha, árvore e/ou estrela.

Um frame Ethernet pode ser processado em tempo real em vez de ser recebido, interpretado e copiado como dados do processo em cada conexão. O FMMU (*Fieldbus Memory Management Unit*) em cada nó Escravo lê os dados que lhe são endereçados ao mesmo tempo em que o telegrama é encaminhado para o próximo dispositivo. De uma maneira similar, os dados de entrada são inseridos enquanto o telegrama é passado. Devido a isso, os frames são atrasados apenas por alguns nanossegundos. Acessos nos terminais de Ethernet podem ser feitos em qualquer ordem, porque a sequência de dados é independente da ordem física. É possível executar comunicação Broadcast, Multicast e comunicação entre os Escravos.

O protocolo EtherCAT permite uma sincronização com precisão, que é necessária, por exemplo, em aplicações onde vários eixos realizam movimentos coordenados simultaneamente, esta pode ser realizada através de um ajuste exato do *Distributed Clock*. Há também a possibilidade de configuração de dispositivos que, em contraste com a comunicação síncrona, têm um elevado grau de tolerância dentro do sistema de comunicação.

A configuração de módulos EtherCAT inicialmente é determinada pelos *Arquivos de Descrição de Dispositivo* dos dispositivos Mestre e Escravo utilizados e pode ser modificada pelo usuário nas caixas de diálogo do *Editor de Configuração*. No entanto, para aplicações convencionais e com o objetivo de ter uma manipulação tão fácil quanto possível, a configuração em larga escala pode ser automatizada, escolhendo o modo *Autoconfig master/slaves* em [EtherCAT Master - General](#).

Observe a possibilidade de modificação dos parâmetros de configuração do Mestre e Escravo também no modo operacional, através das instâncias de Mestre e Escravo, de acordo com a disponibilidade do dispositivo em questão.

5.5.12.1. Instalando e Inserindo Dispositivos EtherCAT

A fim de ser possível inserir e configurar dispositivos EtherCAT como objetos na árvore de dispositivos, os dispositivos Escravos devem ser instalados.

O dispositivo Mestre é instalado automaticamente pela instalação padrão do MasterTool IEC XE. O Mestre EtherCAT define quais Escravos podem ser inseridos.

Para instalar os dispositivos Escravos deve ser aberto o diálogo *Repositório de Dispositivos*, utilizar o filtro *Arquivo de Configuração da Descrição do Dispositivo EtherCAT XML (*.xml)* e selecionar os arquivos de descrição de dispositivo (*EtherCAT XML Device Description / ESI EtherCAT Slave Information*), fornecido com o hardware. As descrições para os Escravos estão disponíveis como arquivos XML (tipo de arquivo: *.xml).

Um Mestre EtherCAT pode ser adicionado à *Árvore de Dispositivos* através do comando *Acréscetar Dispositivo*, através do menu de contexto dos conectores NET da UCP.

Abaixo de um mestre, um ou mais escravos podem ser inseridos, selecionando um Mestre EtherCAT e executando o comando *Acréscetar Dispositivo* (menu de contexto do Mestre EtherCAT) ou executando o comando *Procurar Dispositivos*.

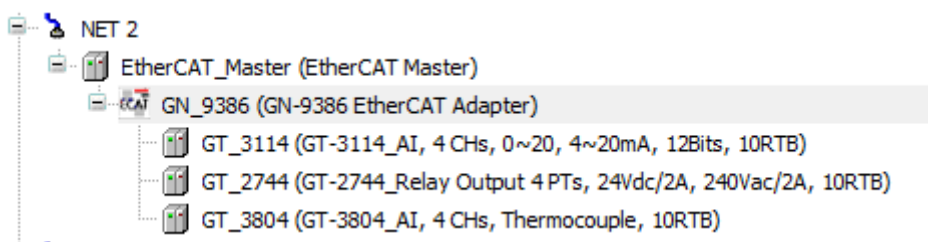


Figura 93: Exemplo de Configuração EtherCAT

ATENÇÃO

- Permitido apenas em portas NET integradas das UCPs.
- Não pode ser usado quando as NETs estiverem configuradas como redundantes/switch.
- Não pode ser usado quando o projeto tiver redundância de cluster.
- Não podem ser instanciados outros drivers na mesma porta NET que o Mestre EtherCAT.

5.5.12.1.1. EtherCAT - Scan Devices

O comando *Scan for Devices...*, disponível no menu de contexto do Mestre EtherCAT, executa uma busca pelos dispositivos Escravos instalados fisicamente na rede EtherCAT do CP atualmente conectado. Isso significa que com este comando é possível detectar e visualizar os componentes de hardware na janela apresentada na figura abaixo, permitindo que o usuário possa mapeá-los diretamente na *Árvore de Dispositivos* do projeto.

É importante salientar que, quando o comando *Scan for Devices...* é selecionado, uma conexão com o CP será estabelecida automaticamente antes da busca iniciar e será encerrada quando a busca terminar. Assim, para que este comando seja executado pela primeira vez, a conexão do Gateway deve ser configurada e deve-se fazer um download do projeto com um Mestre EtherCAT configurado no CP.

Quando o comando for executado, o campo *Scanned Devices* vai conter uma lista de todos os dispositivos e módulos encontrados durante a última verificação. Para adicioná-los ao projeto, basta clicar no botão *Copy All Devices to Project*. É possível, também, executar uma comparação entre os dispositivos encontrados na busca com os presentes no projeto, selecionando a caixa *Show differences to project*.

Se você adicionar um módulo Mestre EtherCAT ao projeto e utilizar o comando *Scan for Devices...* você terá uma lista de todos os Escravos EtherCAT disponíveis. Aparecerão entradas escritas em negrito, caso exista mais de um dispositivo com a mesma descrição. Com um duplo clique sobre esta entrada uma lista será aberta e o dispositivo desejado pode ser selecionado.

Após concluir as modificações na configuração da rede EtherCAT, é necessário executar um novo download do projeto, para que as modificações sejam aplicadas.

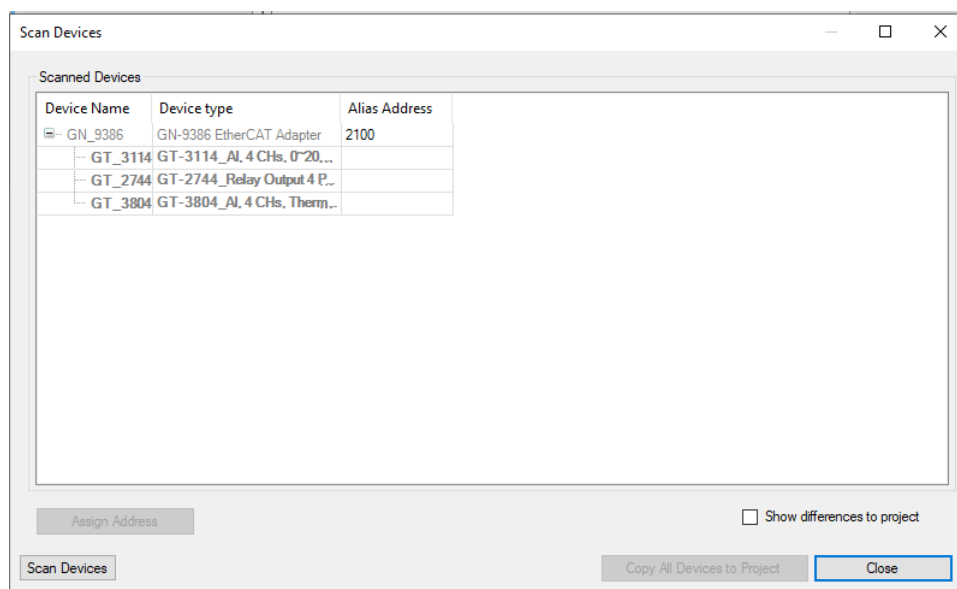


Figura 94: Diálogo Procurar Dispositivos para EtherCAT

5.5.12.2. Configuração do Mestre EtherCAT

A seguir são listadas as opções para a execução da configuração de um Mestre EtherCAT, tal como definido no *Arquivo de Descrição do Dispositivo*.

5.5.12.2.1. EtherCAT Master - General

A seguir são apresentados os parâmetros gerais encontrados na tela inicial de configuração do Mestre EtherCAT, conforme figura abaixo.

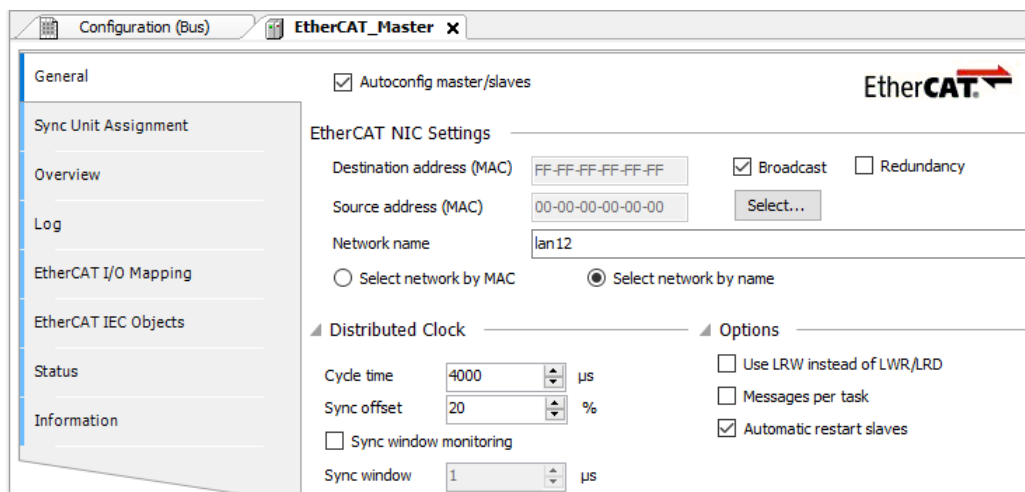


Figura 95: Diálogo de Configuração do Mestre EtherCAT

Configuração	Descrição	Padrão	Opções
Autoconfig master/slaves	Habilita a configuração automática do Mestre e dos Escravos.	Marcado	Marcado Desmarcado
Cycle time [μs]	Configura o período de tempo em que um novo telegrama de dados deve ser enviado ao barramento.	4000	2000 a 1000000
Sync offset [%]	Ajusta o deslocamento da interrupção de sincronização do Escravo EtherCAT para o ciclo do CP.	20	-50 a 50
Sync window monitoring	Se ativada esta opção permite monitorar a sincronização dos Escravos.	Desmarcado	Marcado Desmarcado
Sync window [μs]	Tempo para a Janela de Monitoração de Sincronização.	1	1 a 32768
Use LRW instead of LWR/LRD	Habilitação dos comandos combinados de leitura e escrita.	Desmarcado	Marcado Desmarcado
Messages per task	Se ativada, os comandos leitura e escrita, que estão lidando com mensagens de entrada e saída poderão ser feitos em diferentes tarefas.	Desmarcado	Marcado Desmarcado
Automatic restart slaves	Reinicia os escravos quando a comunicação for abortada.	Marcado	Marcado Desmarcado

Tabela 135: Configurações do Mestre EtherCAT

Notas:

Autoconfig master/slaves: Se esta opção estiver ativada, a maior parte da configuração do Mestre e Escravo será feita automaticamente, baseando-se nos arquivos de descrição e cálculos implícitos. Neste caso, o diálogo de configurações FMMU / Sync não estará disponível. Se esta estiver desmarcada as opções *Endereço Imagem In* e *Endereço Imagem Out* estarão disponíveis para o usuário.

ATENÇÃO

O modo *Autoconfig* é ativado por padrão e, geralmente, suficiente e fortemente recomendado para aplicações padrão. Se a opção estiver desativada, todas as definições de configuração para o Mestre e o(s) Escravo(s) terão de ser feitas manualmente e um conhecimento especializado é necessário. Para a configuração de uma comunicação Escravo-para-Escravo a opção *Autoconfiguração* tem de ser desativada.

Cycle time: Período de tempo após o qual, um novo telegrama de dados deve ser enviado ao barramento. Se a funcionalidade *Relógio Distribuído* estiver ativada, o valor deste parâmetro será transferido para os clocks dos Escravos. Assim, uma sincronização precisa de troca de dados pode ser alcançada, o que particularmente é importante nos casos em que os processos distribuídos especialmente exigem ações simultâneas. Então, uma base de tempo muito precisa para toda a rede, com um jitter significativamente menor do que um microssegundo, pode ser alcançada.

Sync offset: Este valor permite ajustar o deslocamento da interrupção de sincronização do Escravo EtherCAT para o ciclo do CP. Normalmente, o ciclo de tarefa do CP começa 20% mais tarde do que a sincronização de interrupção dos Escravos. Isto significa que a tarefa do CP pode ser atrasada por 80% do tempo de ciclo e nenhuma mensagem será perdida.

Sync window monitoring: Se a sincronização de todos os Escravos está dentro desta janela de tempo o diagnóstico do Mestre EtherCAT *bDistributedClockInSync* será definido como TRUE, caso contrário será FALSE. Quando o Relógio Distribuído é utilizado, é altamente recomendado usar uma tarefa dedicada com alta prioridade como *Tarefa Cíclica de Barramento* do mestre EtherCAT. Desta maneira, é necessário utilizar [Perfis de Projeto](#) que possibilitem a criação de novas tarefas, criar

uma tarefa cíclica com prioridade 0 (tarefa de tempo real), e atribuir a *Tarefa Cíclica de Barramento* do mestre para esta nova tarefa na aba [EtherCAT Master - I/O Mapping](#) do Mestre EtherCAT. O usuário também poderá alterar o valor da variável *wDCInSyncWindow*, configurando qual o jitter máximo permitido na sincronização entre mestre e escravos.

Use LRW instead of LWR/LRD: A ativação desta opção habilita a comunicação Escravo-para-Escravo, pois, ao invés de comandos separados de leitura (LRD) e escrita (LWR), comandos combinados leitura/escrita (LRW) serão utilizados.

Automatic restart slaves: Habilitando esta opção o Mestre reiniciará os Escravos assim que a comunicação for abortada.

5.5.12.2.2. EtherCAT Master - Sync Unit Assignment

Esta guia do editor de configuração do Mestre EtherCAT mostra todos os escravos que são inseridos abaixo de um mestre específico com uma atribuição para as unidades de sincronização.

Com o EtherCAT sync units, múltiplos escravos são configurados em grupos e subdivididos em unidades menores. Para cada grupo, o contador de trabalho pode ser monitorado para uma melhor e mais exata detecção de erros. Assim que um escravo está em falta num grupo de unidades de sincronização, os outros escravos do grupo também são mostrados como desaparecidos. A detecção ocorre imediatamente no próximo ciclo de barramento, porque o contador de trabalho é verificado continuamente. Com os diagnósticos do dispositivo, o grupo em falta pode ser remediado o mais rapidamente possível.

Os grupos não afectados permanecem operáveis sem qualquer interferência.

5.5.12.2.3. EtherCAT Master - Overview

Esta guia do editor de configuração do Mestre EtherCAT fornece uma visão geral dos estados de todos os escravos, que são inseridos abaixo deste mestre e têm um endereço. Os módulos não são exibidos.

5.5.12.2.4. EtherCAT Master - I/O Mapping

Esta guia do editor de configuração do Mestre EtherCAT oferece a possibilidade de mudar a tarefa que será utilizada para atualização do barramento.

5.5.12.2.5. EtherCAT Master - IEC Objects

Esta guia do editor de configuração do Mestre EtherCAT lista *objetos* que permitem o acesso ao dispositivo a partir da aplicação IEC. Em modo online, este é utilizado para monitoração.

5.5.12.2.6. EtherCAT Master - Status / Informations

A guia Status do editor de configuração do Mestre EtherCAT fornece informações de estado (por exemplo, *'Executando'*, *'Parado'*) e mensagens de diagnóstico específicas do dispositivo e do sistema de barramento interno.

A guia Informação, presente no editor de configuração do Mestre EtherCAT, exhibe, caso disponíveis, as seguintes informações gerais para o módulo: *Nome, Fornecedor, Tipo, Número de Versão, Categorias, Número de Ordem, Descrição, Imagem.*

5.5.12.3. Configuração do Escravo EtherCAT

A seguir são listadas as principais opções de configuração para um Escravo EtherCAT, tal como definido no *Arquivo de Descrição do Dispositivo*.

5.5.12.3.1. EtherCAT Slave - General

A seguir são apresentados os parâmetros gerais encontrados na tela inicial de configuração do Escravo EtherCAT. Estes campos estarão disponíveis somente se o modo *Autoconfiguração* (Mestre) não estiver ativado.

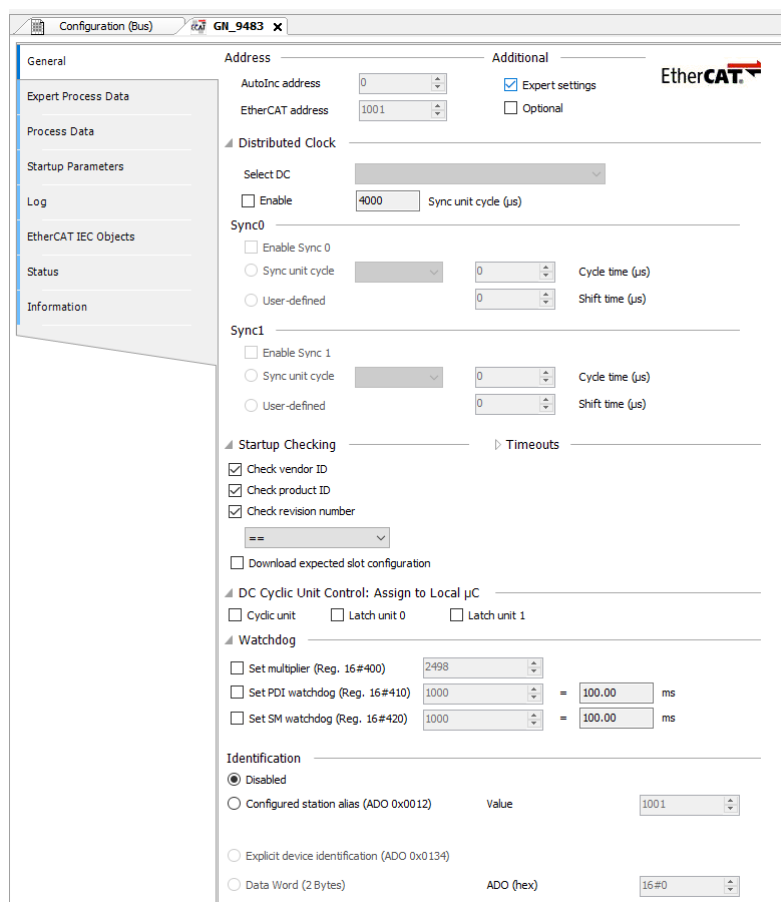


Figura 96: Diálogo de Configuração do Escravo EtherCAT

Configuração	Descrição	Padrão	Opções
AutoInc address	Endereço Autoincremental (16-bit), definido pela posição do Escravo na rede.	-	-65535 a 0
EtherCAT address	Endereço final do Escravo, atribuído pelo Mestre durante a inicialização. Este endereço é independente da posição na rede.	-	1 a 65535
Expert settings	Habilita as opções de configuração avançadas do Escravo.	Desmarcado	Marcado Desmarcado
Optional	Declara o Escravo como Opcional.	Desmarcado	Marcado Desmarcado
Select DC	Apresenta todas as configurações, para Relógios Distribuídos, fornecidas pelo arquivo de descrição do dispositivo.	-	-
Enable Distributed Clock	Habilita as opções de configuração do Relógio Distribuído.	Desmarcado	Marcado Desmarcado

Configuração	Descrição	Padrão	Opções
Sync unit cycle [μs]	Apresenta o Tempo de Ciclo configurado no Mestre.	100000	2000 a 1000000
Enable (Sync 0)	Habilita as configurações da unidade de sincronização Sync 0.	Desmarcado	Marcado Desmarcado
Sync unit cycle Sync 0	Selecionando esta opção o fator escolhido será multiplicado pelo valor em Sinc. Unidade de Ciclo.	Desmarcado	Marcado Desmarcado
User-defined Sync 0	Se marcado, o tempo desejado em microssegundos pode ser inscrito no campo Tempo de Ciclo (μ s).	Desmarcado	Marcado Desmarcado
Cycle time [μs] Sync 0	Mostra o tempo de ciclo atualmente definido.	100000	1 a 2147483647
Shift Time [μs] Sync 0	Tempo entre os eventos de sincronização e as Outputs Valid ou Input Latch.	0	-2147483648 a 2147483647
Enable (Sync 1)	Habilita as configurações da unidade de sincronização Sync 1.	Desmarcado	Marcado Desmarcado
Sync unit cycle Sync 1	Selecionando esta opção o fator escolhido será multiplicado pelo valor em Sinc. Unidade de Ciclo.	Desmarcado	Marcado Desmarcado
User-defined Sync 1	Se marcado, o tempo desejado em microssegundos pode ser inscrito no campo Tempo de Ciclo (μ s).	Desmarcado	Marcado Desmarcado
Cycle time [μs] Sync 1	Mostra o tempo de ciclo atualmente definido.	100000	1 a 2147483647
Shift Time [μs] Sync 1	Tempo entre os eventos de sincronização e as Outputs Valid ou Input Latch.	0	-2147483648 a 2147483647
Check vendor ID	Se desmarcado, desabilita a Verificação do ID do Fornecedor.	Marcado	Marcado Desmarcado
Check product ID	Se desmarcado, desabilita a Verificação do ID do Produto.	Marcado	Marcado Desmarcado
SDO access	Configura uma referência para a verificação do tempo de um Acesso SDO.	-	0 a 100000
I -> P	Configura uma referência para a verificação do tempo de troca do estado Inicial para Pré-Operacional.	-	0 a 100000

Configuração	Descrição	Padrão	Opções
P -> S/S -> O	Configura uma referência para a verificação do tempo de troca dos estados Pré-Operacional para Segurança-Operacional e de Segurança-Operacional para Operacional.	-	0 a 100000
Cyclic unit	Atribui a Unidade de Ciclo ao microprocessador local.	Desmarcado	Marcado Desmarcado
Latch Unit 0	Atribui a Latch Unit 0 ao microprocessador local.	Desmarcado	Marcado Desmarcado
Latch Unit 1	Atribui a Latch Unit 1 ao microprocessador local.	Desmarcado	Marcado Desmarcado

Tabela 136: Configurações do Escravo EtherCAT

Notas:

AutoInc address: Este endereço é usado apenas durante a inicialização, quando o Mestre está atribuindo os endereços EtherCAT para os Escravos. Quando para este propósito, o primeiro telegrama percorre os Escravos, cada Escravo de leitura rápida aumenta seu *Endereço AutoInc* por 1. O Escravo com o endereço 0, finalmente, vai receber os dados.

Optional: Se um Escravo é declarado como *Opcional* nenhuma mensagem de erro será criada caso o dispositivo não exista no sistema de barramento. Se essa opção for ativada, um endereço de estação deve ser armazenado no dispositivo Escravo. Assim, um endereço *Pseudônimo de Estação* deve ser definido e gravado na EEPROM. Esta opção só está disponível se a opção *Autoconfiguração Mestre/Escravos* estiver ativada nas configurações do Mestre EtherCAT, e se esta função for suportada pelo Escravo EtherCAT.

Enable Distributed Clock: Se a funcionalidade *Relógio Distribuído* for ativada, o tempo de ciclo de troca de dados, exibido no campo *Unidade de Ciclo Sync(μs)* será determinado pelo *Tempo de Ciclo* do Mestre. Assim, o clock Mestre pode sincronizar a troca de dados dentro da rede. As definições para manusear a(s) unidade(s) de sincronização dependem do Escravo.

Enable Sync 0: Se essa opção for ativada, a unidade de sincronização *Sync0* é utilizada. Uma unidade de sincronização descreve um conjunto de dados do processo que são trocados de forma síncrona.

Sync unit cycle Sync 0: Se esta opção estiver ativada, o *Tempo de Ciclo* do Mestre, multiplicado pelo fator escolhido será utilizado como tempo de ciclo de sincronização para o Escravo. O campo *Tempo de Ciclo(μs)* mostra o tempo de ciclo atualmente definido.

Shift time: O Tempo de deslocamento descreve o tempo entre os eventos de sincronização (*Sync0*, *Sync1*) e as *Outputs Valid* ou *Input Latch*. Valor gravável, se o escravo suporta transferência de *Outputs Valid* ou *Input Latch*.

Enable Sync 1: Se essa opção for ativada, a unidade de sincronização *Sync1* é utilizada. Uma unidade de sincronização descreve um conjunto de dados do processo que são trocados de forma síncrona.

Sync unit cycle Sync1: Se essa opção for ativada, o *Tempo de Ciclo* do Mestre, multiplicado pelo fator escolhido será utilizado como tempo de ciclo de sincronização para o Escravo. O campo *Tempo de Ciclo(μs)* mostra o tempo de ciclo definido atualmente.

Check vendor ID and Check product ID: Por padrão, na inicialização do sistema o *ID do Fornecedor* e/ou o *ID do Produto* serão verificados contra as configurações atuais. Se for detectada alguma diferença, o barramento será interrompido e nenhuma ação adicional será executada. Isso serve para evitar o download de uma configuração errada. Estas opções têm a finalidade de desativar esta verificação.

SDO access: Por padrão não há tempo limite definido para a ação de envio de uma lista SDO ao iniciar o sistema. Porém, se houver a necessidade de verificar se esta ação excede certo tempo, este deve ser especificado neste campo.

I -> P: Por padrão não há tempo limite definido para a ação de troca do modo *Inicial* para o modo *Pré-Operacional*. Porém, se houver a necessidade de verificar se esta ação excede certo tempo, este deve ser especificado neste campo.

P -> S / S -> O: Por padrão não há tempo limite definido para a ação de troca dos modos *Pré-Operacional* para *Segurança-Operacional* e respectivamente de *Segurança-Operacional* para *Operacional*. Porém, se houver a necessidade de verificar se esta ação excede certo tempo, este deve ser especificado neste campo.

DC Cyclic Unit Control: Escolha a(s) opção(ões) desejada(s), relativa às funções de *Relógio Distribuído*, a fim de definir o que deve ser atribuído ao microprocessador local. O controle é feito no registro 0x980 do Escravo EtherCAT. As configurações possíveis são: *Unidade Cíclica*, *Unidade Latch 0*, *Unidade Latch 1*.

Enable: Se a definição de Opcional não estiver ativada, essa configuração pode ser ativada se for suportada pela descrição do dispositivo Escravo. Ele permite a atribuição direta de um endereço de pseudônimo, a fim de obter o endereço dos Escravos independente de sua posição dentro do barramento. Se a opção *Opcional* estiver ativada, esta caixa de seleção estará desativada.

5.5.12.3.2. EtherCAT Slave - Process Data

A guia *Process Data* do editor de configuração do Escravo EtherCAT mostra os dados do processo de entrada e de saída do Escravo, cada um definido por nome, tipo e índice do arquivo de descrição do dispositivo, como pode ser visto na figura abaixo.

As entradas selecionadas (a serem lidas) e saídas (a serem escritas) do dispositivo estarão disponíveis no diálogo de [EtherCAT Slave - Module I/O Mapping](#) como saídas e entradas para o CP, no qual as variáveis do projeto podem ser mapeadas.

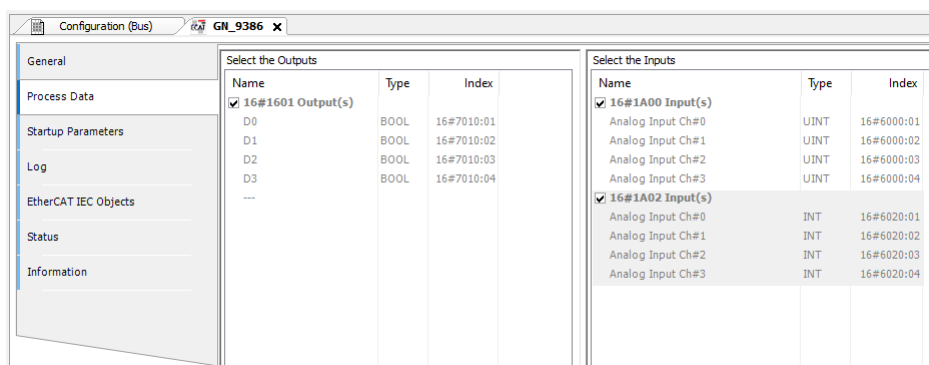


Figura 97: Diálogo Dados do Processo

O diálogo *Expert Process Data* só será visível no editor de configuração do Escravo EtherCAT se a opção *Expert settings* estiver ativada. Ele fornece uma visão mais detalhada dos dados do processo, além do que é apresentado na aba *Process data*. Além disso, nesta aba é possível habilitar o download da *PDO Assignment* e a *PDO configuration*.

ATENÇÃO

Se o Escravo não aceitar a configuração do PDO, ele ficará em modo pré-operacional e nenhuma troca de dados em tempo real será possível.

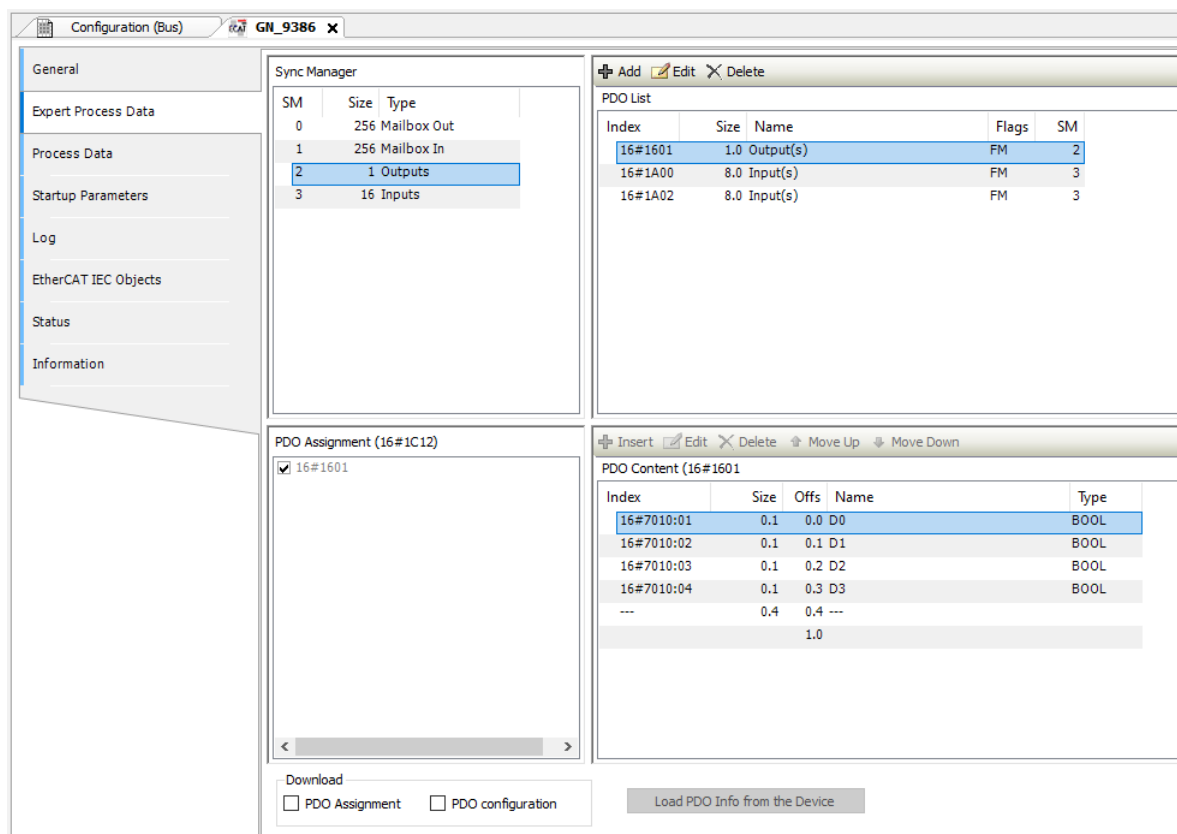


Figura 98: Diálogo Dados de Processo Avançados

Esta guia está dividida em quatro seções e duas opções:

- *Sync Manager*: Lista de SM *Sync Manager* com o tamanho dos dados e tipo de PDO.
- *PDO Assignment*: Lista de PDOs atribuídos ao *Sync Manager* selecionado. A caixa de seleção ativa o PDO e canais de E/S são criados. É parecido com as janelas de configuração de PDO simples. Aqui apenas PDOs podem ser ativados ou desativados.
- *PDO List*: Lista de todos os PDOs definidos no arquivo de descrição do dispositivo. PDOs individuais podem ser apagados, editados ou adicionados por meio da execução do respectivo comando no menu de contexto.
- *PDO Content*: Exibe o conteúdo do PDO selecionado na seção acima. As inscrições podem ser apagadas, editadas ou adicionadas por meio da execução do respectivo comando a partir do menu de contexto.
- *PDO Assignment*: Se ativado, um comando de escrita CoE será adicionado ao índice 0x1CXX para escrever a configuração PDO 0x16XX ou 0x1A00.
- *PDO configuration*: Se ativado, vários comandos da gravação CoE serão adicionados ao escrever o mapeamento PDO no Escravo.

ATENÇÃO

Se o Escravo não suportar a configuração PDO, o download desta pode resultar em um erro de Escravo. Esta função deve ser usada apenas por especialistas.

5.5.12.3.3. EtherCAT Slave - Edit PDO List

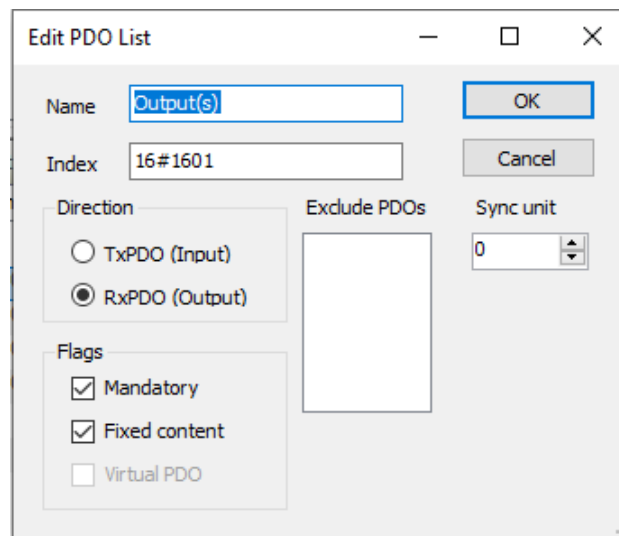


Figura 99: Diálogo Editar Lista PDO

Este diálogo é aberto através do menu de contexto da área Lista PDO, apresentada na Figura 98. Seguem algumas explicações das opções de configuração presentes neste diálogo.

- *Name*: Nome da entrada PDO.
- *Index*: Índice do PDO em edição.
- *TxPDO (Input)*: Se ativado, o PDO será transferido do Mestre para o Escravo.
- *RxPDO (Output)*: Se ativado, o PDO será transferido do Escravo para o Mestre.
- *Mandatory*: O PDO é necessário e não pode ser desmarcado na área PDO *Assignment*.
- *Fixed content*: O conteúdo do PDO é fixo e não pode ser alterado. Não é possível adicionar entradas no painel PDO *Fixed content*.
- *Virtual PDO*: Reservado para uso futuro.
- *Exclude PDOs*: É possível definir PDOs que podem, ou não, serem selecionados juntos ao PDO em edição na área PDO *Assignment*, ou na guia *Process data*. Se um PDO é marcado nesta lista este não poderá ser selecionado, ficando cinza, no ambiente PDO *Assignment* quando o PDO em edição for selecionado.
- *Sync unit*: ID do *Sync Manager* do PDO para o qual deve ser atribuído.

5.5.12.3.4. EtherCAT Slave - Startup Parameters

Na guia *Startup Parameters* podem ser definidos parâmetros para o dispositivo, que serão repassados pelos SDOs (*Service Data Objects*) ou IDN na inicialização do sistema. As opções presentes neste ambiente, além da possibilidade de acesso, variam de acordo com o Escravo EtherCAT utilizado e estão presentes no seu *Arquivo de Descrição do Dispositivo*.

5.5.12.3.5. EtherCAT Slave - Module I/O Mapping

Esta guia do editor de configuração do Escravo EtherCAT oferece a possibilidade de atribuir as variáveis de projeto para respectivas entradas e saídas do EtherCAT. Assim, as variáveis do Escravo EtherCAT podem ser controladas pela *Aplicação de Usuário*.

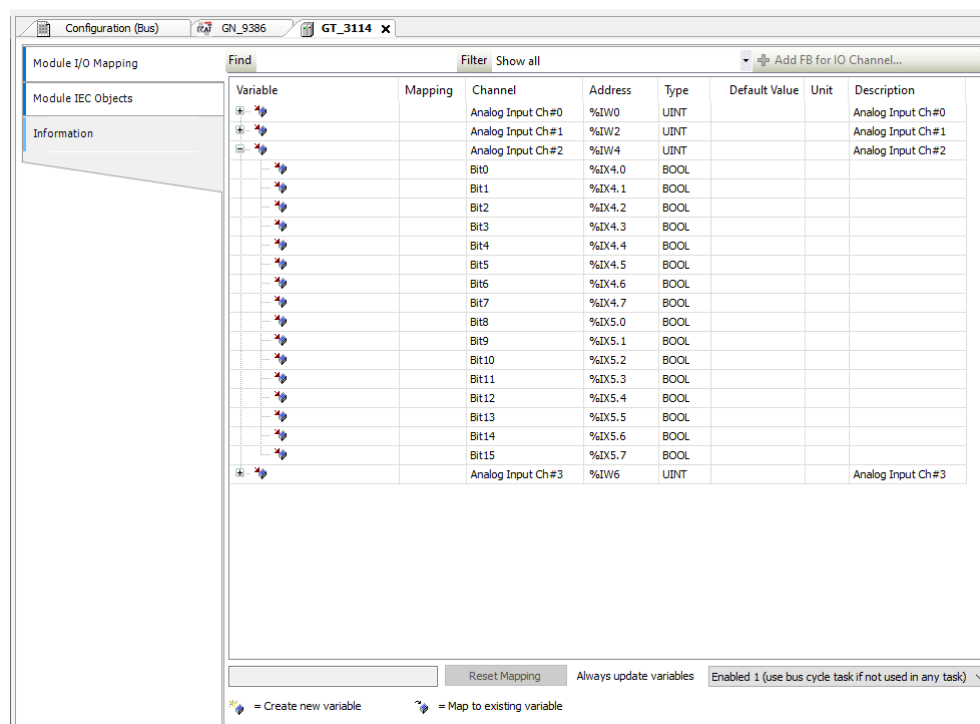


Figura 100: Diálogo Mapeamento de E/S Escravo

5.5.12.3.6. EtherCAT Slave - Status and Information

A guia *Status* do editor de configuração do Escravo EtherCAT fornece informações de estado (por exemplo, 'Executando', 'Parado') e mensagens de diagnóstico específicas do dispositivo e do sistema de barramento interno.

A guia *Information*, presente no editor de configuração do Escravo EtherCAT, exibe, caso disponíveis, as seguintes informações gerais para o módulo: *Nome*, *Fornecedor*, *Tipo*, *Número de Versão*, *Categorias*, *Número de Ordem*, *Descrição*, *Imagem*.

5.5.13. EtherNet/IP

O EtherNet/IP é um protocolo de arquitetura mestre-escravo, o qual consiste de um EtherNet/IP Scanner (o mestre) e um, ou mais, EtherNet/IP Adapters (o escravo).

O EtherNet/IP é um protocolo baseado na CIP (*Common Industrial Protocol*), o qual tem dois propósitos primários: O transporte de dados de controle-orientado associados com dispositivos de E/S e o transporte de outras informações relacionadas ao sistema sendo controlado, tais como parâmetros de configuração e diagnósticos. O primeiro é realizado por mensagens implícitas, enquanto o segundo é realizado através de mensagens explícitas.

O sistema em execução, das UCPs, pode atuar tanto como Scanner como Adapter. Cada interface NET das UCPs suporta apenas uma instância EtherNet/IP e ele não pode ser instanciado em um módulo de expansão Ethernet.

Uma instância EtherNet/IP Adapter suporta um número ilimitado de módulos ou de bytes de Entrada/Saída, nestes módulos podem ser adicionadas variáveis dos tipos: BYTE, BOOL, WORD, DWORD, LWORD, USINT, UINT, UDINT, ULINT, SINT, INT, DINT, LINT, REAL E LREAL.

ATENÇÃO

EtherNet/IP não pode ser usado em conjunto com Modo Redundante da interface ethernet nem com redundância de Half-Cluster.

ATENÇÃO

Para evitar problemas de comunicação, o Ethernet/IP Scanner só pode ter dispositivos Adapters que estejam configurados na mesma sub-rede.

5.5.13.1. Interface EtherNet/IP

Para adicionar um EtherNet/IP Scanner ou um Adapter é necessário adicionar um *Adaptador Ethernet* abaixo da NET desejada. Isto pode ser feito através do comando *Adicionar Dispositivo*. Abaixo deste *Adaptador Ethernet*, é possível adicionar um *Scanner* ou um *Adapter*.

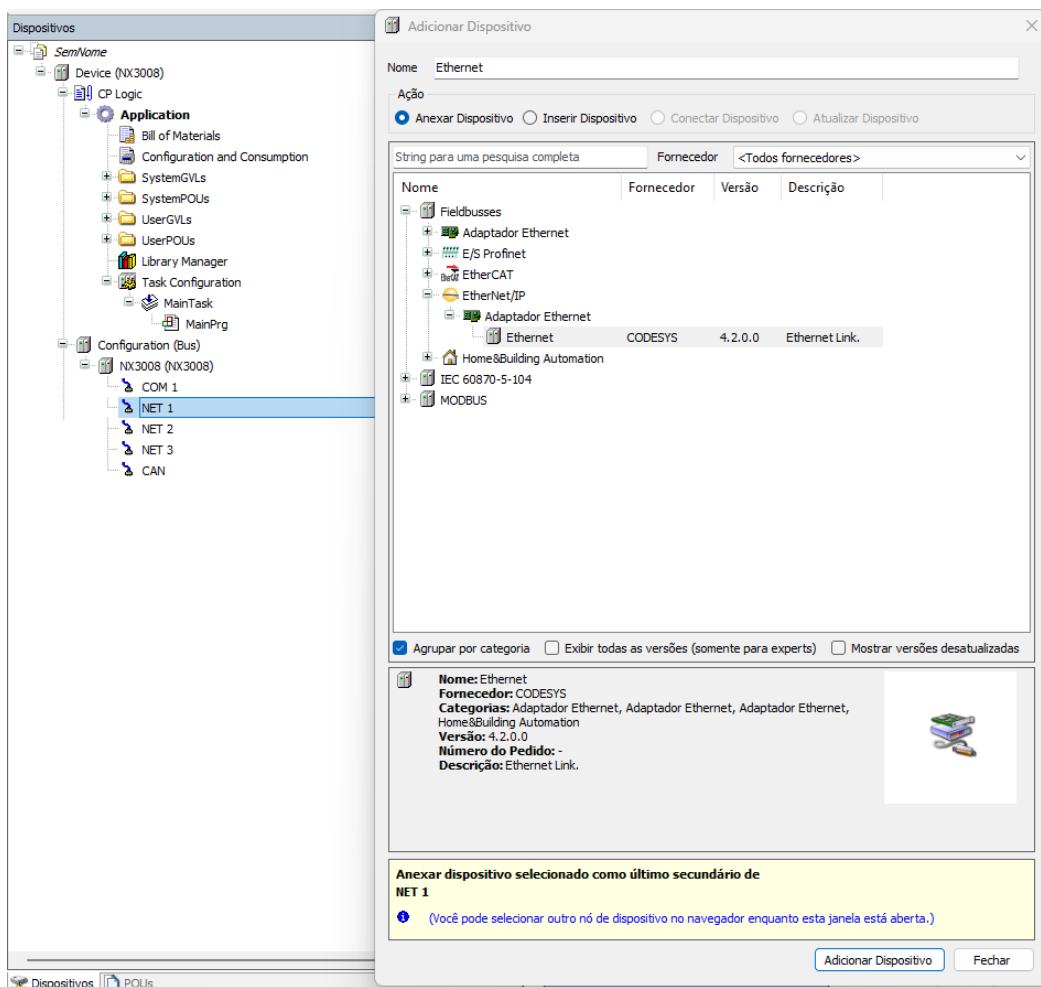


Figura 101: Adicionando um Adaptador Ethernet

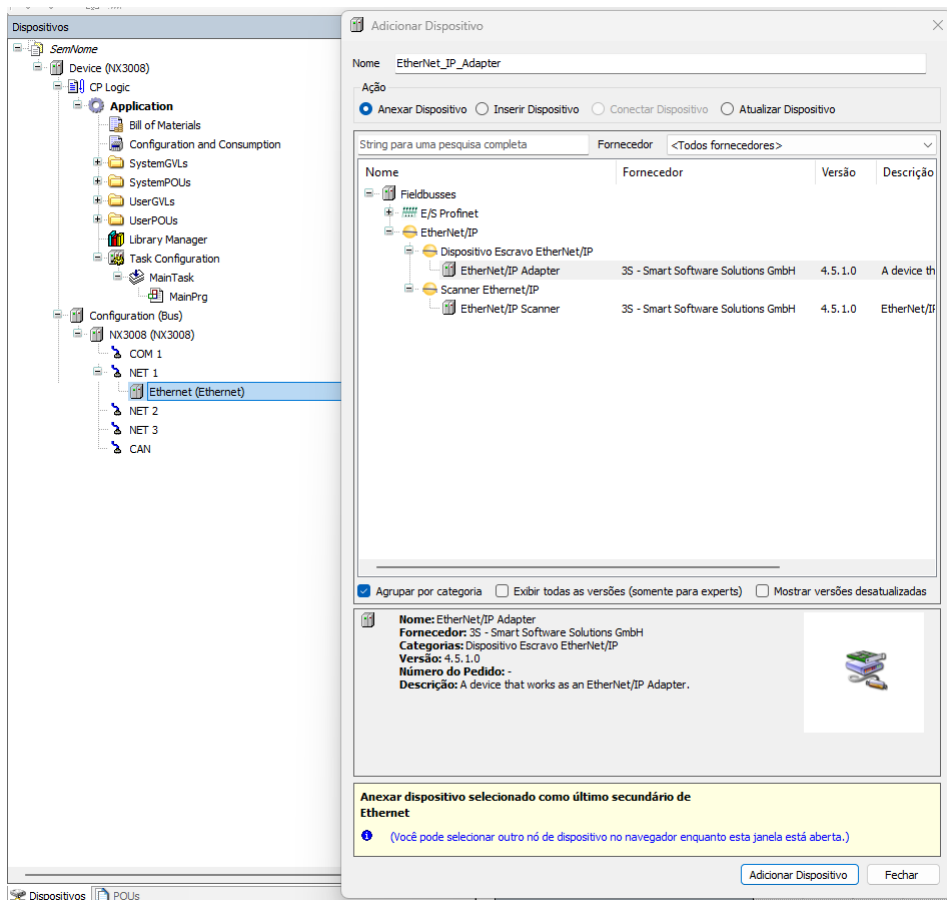


Figura 102: Adicionando um Adapter ou Scanner EtherNet/IP

5.5.13.2. Configuração do Scanner EtherNet/IP

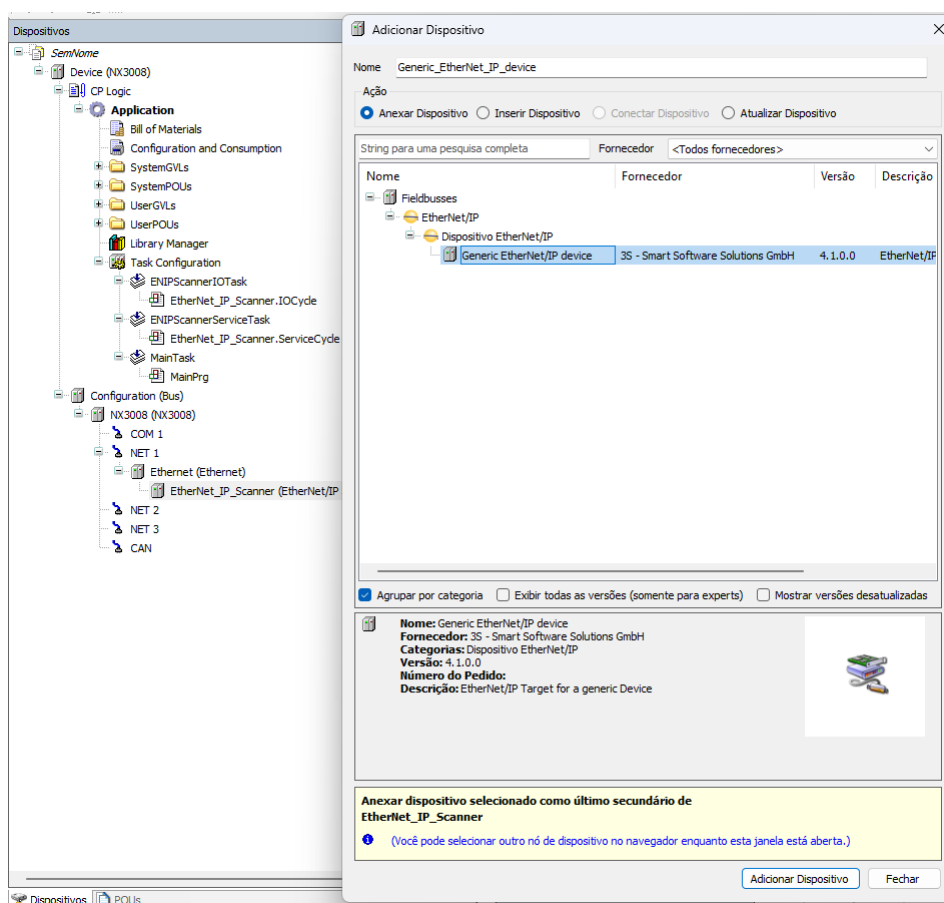


Figura 103: Adicionando um Adapter EtherNet/IP Sob o Scanner

5.5.13.2.1. Geral

Depois de abrir o Dispositivo declarado sob o Scanner é possível configurá-lo como for necessário. A primeira aba é *Geral*, nela é possível configurar o *Endereço IP* e o parâmetro *Chave Eletrônica*. Esses parâmetros devem ser marcados ou desmarcados se um Adapter sendo usado está instalado no MasterTool. De outra forma, se o Adapter usado é do tipo *Generic*, então os campos *ID do Fornecedor*, *Tipo do Dispositivo*, *Código do Produto*, *Revisão grande* e *Revisão pequena* devem ser preenchidos com as informações corretas do fabricante e as caixas marcadas tanto quanto necessário. A Altus por sua vez, tem um ID próprio, que é o ID "1454".

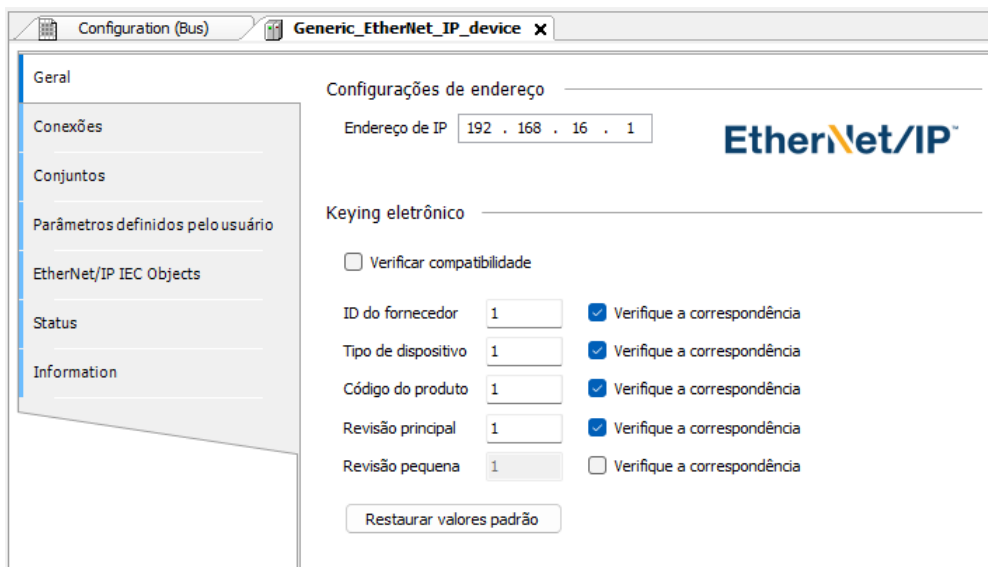


Figura 104: Aba Gerais - EtherNet/IP

5.5.13.2.2. Conexões

A área superior da aba *Conexões* mostra uma lista de todas as conexões configuradas. Quando há uma conexão *Exclusive Owner* no arquivo EDS, ela é inserida automaticamente quando o Adapter é adicionado. Os dados de configuração para estas conexões podem ser mudados mais a baixo nesta janela.

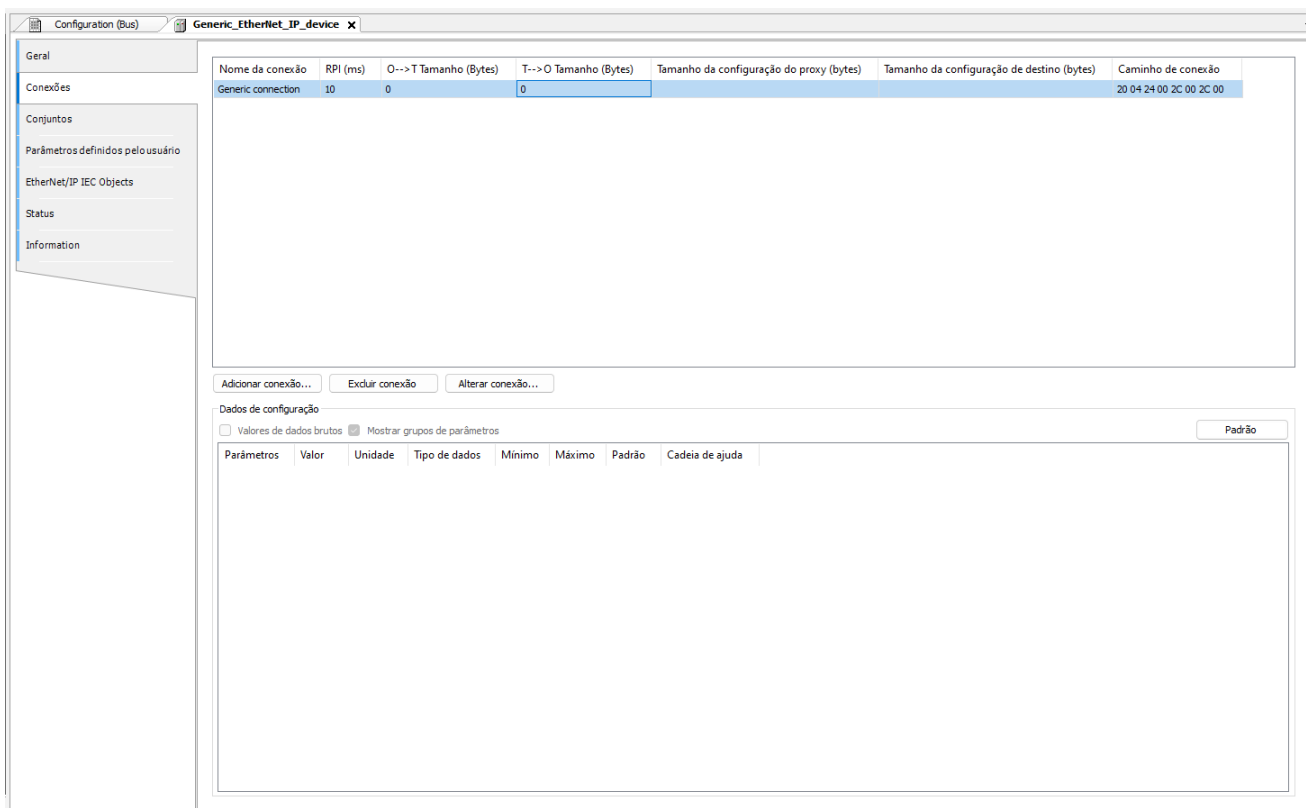


Figura 105: Aba Conexões - EtherNet/IP

5. CONFIGURAÇÃO

Notas:

Para dois ou mais EtherNet/IP Scanners conectarem no mesmo Adapter remoto:

1. Apenas um dos Scanners pode estabelecer uma conexão *Exclusive Owner*.
2. O mesmo valor de *RPI(ms)* deve estar configurado para ambos os Scanners.

Os dados de configuração estão definidos no arquivo EDS. Os dados são transmitidos para o Adapter remoto quando a conexão é aberta.

Configuração	Descrição	Padrão	Opções
RPI (ms)	Intervalo de Requisição de Pacote: Intervalo de troca de dados de entrada e saída.	10 ms	Múltiplo do intervalo da Bus Cycle Task a qual está associada
O -> T Tamanho (Bytes)	Tamanho dos dados produtores do Scanner para o Adapter (Originator -> Target)	0	0 - 65527
T -> O Tamanho (Bytes)	Tamanho dos dados consumidores do Adapter para o Scanner (Target -> Originator)	0	0 - 65531
Tamanho da Configuração do Proxy (Bytes)	Tamanho dos dados de configuração do Proxy.	-	-
Tamanho da Configuração de destino (Bytes)	Tamanho dos dados de configuração do Destino.	-	-
Caminho de Conexão	Endereço dos objetos de configuração – objetos de entrada – objetos de saída.	Caminho gerado automaticamente	Caminho gerado automaticamente, Caminho Definido pelo Usuário e Caminho definido pelo nome Simbólico

Tabela 137: Parâmetros de Conexão do EtherNet/IP

Para *Adicionar* novas conexões, há o botão *Adicionar Conexão...* o qual vai abrir a janela *Nova Conexão*. Nesta janela é possível configurar um novo tipo de conexão daquelas predefinidas no EDS do Adapter ou uma conexão do zero quando utilizado um dispositivo Genérico.

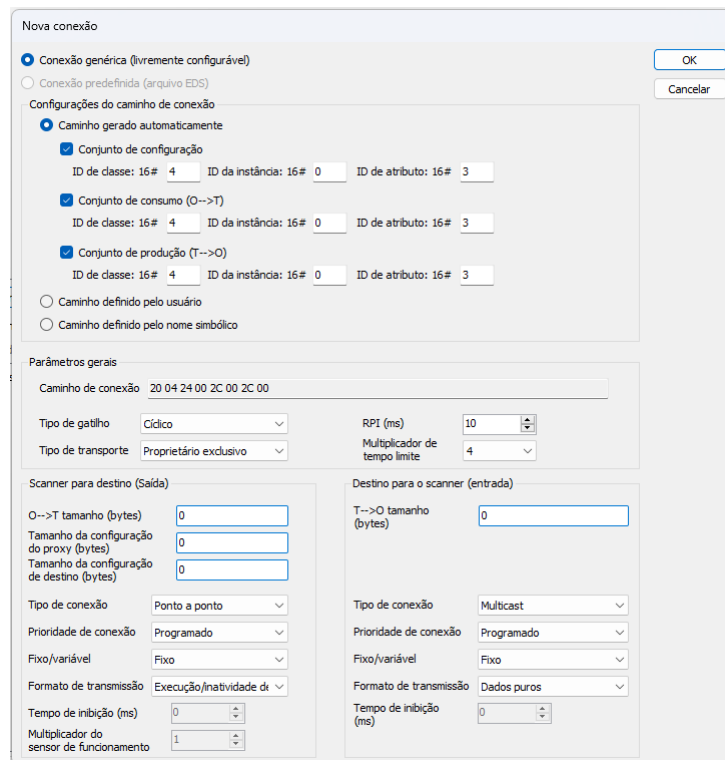


Figura 106: Janela de Novas Conexões EtherNet/IP

5.5.13.2.3. Assemblies

A área superior da aba *Assemblies* lista todas as conexões configuradas. Quando uma conexão está selecionada, as entradas e saídas associadas são mostradas na área inferior da aba.

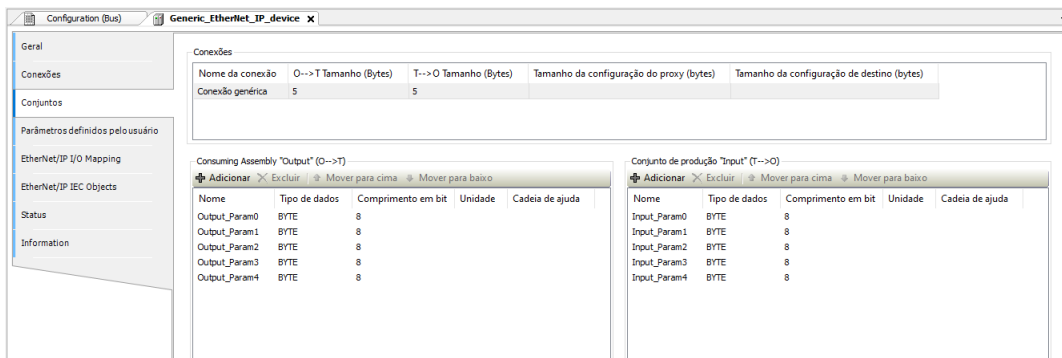


Figura 107: Aba Assemblies - EtherNet/IP

Conjunto de Saídas e Conjunto de Entradas:

Configuração	Descrição
Adicionar	Abre a caixa de diálogo “Adicionar Entrada/Saída”
Excluir	Deleta todas as Entradas/-Saídas Seleccionadas.
Mover Para Cima	Movimenta pela lista a Entrada/Saída seleccionada.
Mover Para Baixo	A ordem na lista determina a ordem no Mapeamento de E/S.

Tabela 138: Aba Assemblies - EtherNet/IP

Caixa de diálogo *Adicionar Entrada/Saída*:

Configuração	Descrição
Nome	Nome da Entrada/Saída a ser inserida
String de Ajuda	
Tipo de Dado	Tipo de Entrada/Saída a ser inserida. Este tipo também define seu tamanho em bits (Comprimento do Bit).
Comprimento do Bit	Este valor NÃO deve ser alterado.

Tabela 139: Janela “Adicionar Entrada/Saída” – EtherNet/IP

5.5.13.2.4. EtherNet/IP: Mapeamento de E/S

Aba *Mapeamento de E/S* mostra, na coluna *Variável*, o nome das instâncias de *Adapter*, automaticamente geradas, abaixo de *Objetos IEC*. Desta forma, a instância pode ser acessada pela aplicação. Aqui as variáveis do projeto são mapeadas para as entradas e saídas do adapter.

5.5.13.3. Configuração do Adapter EtherNet/IP

O Adapter EtherNet/IP requer módulos EtherNet/IP. Os módulos vão prover mapeamentos de E/S que podem ser manipulados pela aplicação do usuário pelos endereços %I ou %Q de acordo com sua configuração.

Novos Adapters podem ser instalados no MasterTool com os arquivos EDS. As opções de configuração podem divergir dependendo do arquivo de descrição do dispositivo do Adapter adicionado.

5.5.13.3.1. Geral

A primeira aba do EtherNet/IP Adapter é a aba *Geral*. Nela é possível configurar os parâmetros da *Chave Eletrônica*, utilizada no Scanner para verificar a compatibilidade. Nesta aba também é possível instalar o EDS do dispositivo diretamente no repositório de dispositivos do MasterTool ou exportá-lo.

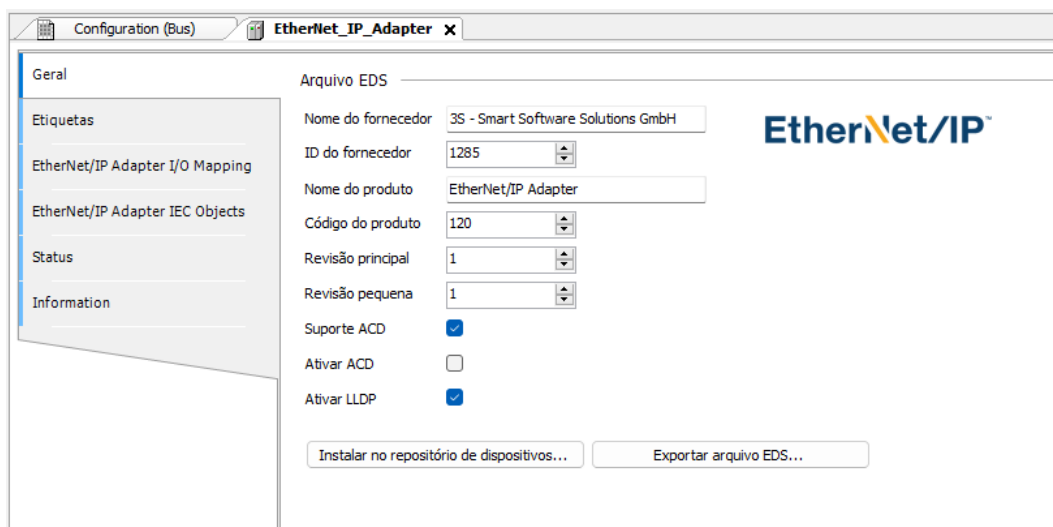


Figura 108: Aba Geral - EtherNet/IP_Adapter

5.5.13.3.2. EtherNet/IP Adapter: Mapeamento de E/S

Na aba *Mapeamento de E/S*, é possível configurar em qual tarefa de ciclo do barramento o Adapter irá executar.

5.5.13.4. Configuração do Módulo EtherNet/IP

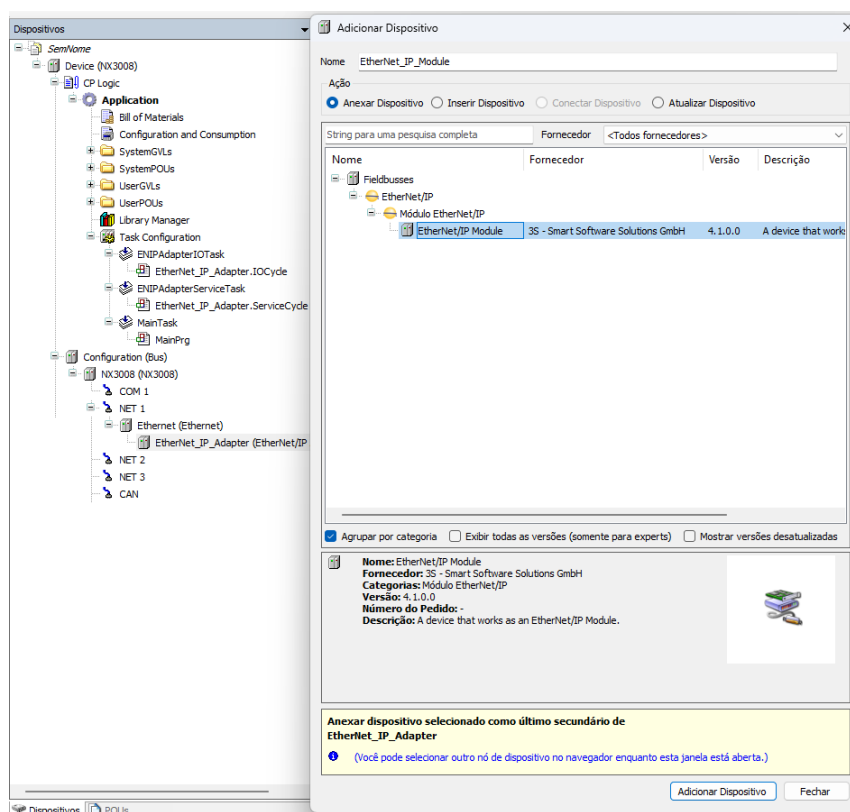


Figura 109: Adicionando um Módulo EtherNet/IP Abaixo do Adapter

5.5.13.4.1. Assemblies

Os parâmetros da aba *Geral* do módulo seguem as mesmas regras descritas nas tabelas 138 e 139.

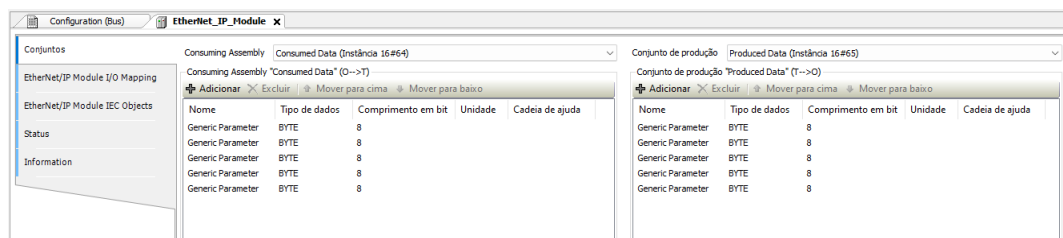


Figura 110: Aba Assemblies Módulo - EtherNet/IP

5.5.13.4.2. EtherNet/IP Module: Mapeamento de E/S

Aba *Mapeamento de E/S* mostra, na coluna *Variável*, o nome das instâncias de Adapter, automaticamente geradas. Desta forma, a instância pode ser acessada pela aplicação do usuário.

5.5.14. IEC 60870-5-104 Servidor

Ao selecionar esta opção no MasterTool, a UCP passa a ser servidor da comunicação IEC 60870-5-104, permitindo a conexão com até três dispositivos clientes IEC 60870-5-104. Para cada cliente, o driver possui uma fila de eventos exclusiva com as seguintes características:

- Tamanho: 1000 eventos
- Retentividade: não é retentiva
- Política de estouro: mantém o mais recente

Para configurar este protocolo, é necessário executar os seguintes passos:

- Adicionar a instância do protocolo Servidor IEC 60870-5-104 a um canal Ethernet disponível. Para realizar esse procedimento, consultar a seção [Inserindo uma Instância de Protocolo](#)
- Configurar a interface Ethernet. Para realizar esse procedimento, consultar a seção [Configuração das Interfaces Ethernet](#)
- Configurar os parâmetros gerais do protocolo Servidor IEC 60870-5-104, com o modo de conexão Porta ou IP, e o número da porta TCP quando o modo de conexão selecionado for o modo IP
- Adicionar e configurar dispositivos, definindo os devidos parâmetros
- Adicionar e configurar os mapeamentos IEC 60870-5-104, especificando o nome da variável, tipo do objeto, endereço do objeto, tamanho, faixa, banda morta, e tipo da banda morta
- Configurar os parâmetros da camada de enlace, especificando os endereçamentos, os time-outs de comunicação e os parâmetros de comunicação
- Configurar os parâmetros da camada de aplicação, configuração do sincronismo, comandos, bem como o modo de transmissão dos objetos *Integrated Totals*

As descrições de cada configuração estão relacionadas a seguir, neste capítulo.

5.5.14.1. Tipos de Dados

A tabela abaixo mostra o tipo de variável suportada pelas UCPs da Série Nexto para cada um dos tipos de dados do protocolo IEC 60870-5-104.

Tipo do Objeto	Tipos de Variáveis IEC
Single Point Information (M_SP_NA)	BOOL
	BIT
Double Point Information (M_DP_NA)	DBP
Step Position Information (M_ST_NA)	USINT
Measured Value, normalized value (M_ME_NA)	INT
Measured Value, scaled value (M_ME_NB)	INT
Measured Value, short floating point value (M_ME_NC)	INT
	UINT
	DINT
	UDINT
	REAL
Integrated Totals (M_IT_NA)	INT
	DINT
Bitstring Information (M_BO_NA)	DWORD
Single Command (C_SC_NA)	BOOL
	BIT
Double Command (C_DC_NA)	DBP
Regulating Step Command (C_RC_NA)	DBP
Setting Point Command, normalized Value (C_SE_NA)	INT
Setting Point Command, scaled Value (C_SE_NB)	INT
Setting Point Command, short floating point Value (C_SE_NC)	REAL
Bitstring Command (C_BO_NA)	DWORD

Tabela 140: Declaração de Variáveis para IEC 60870-5-104

Notas:

Regulating Step Command: Os estados *Lower* e *Higher* do objeto C_RC_NA estão associados respectivamente aos estados *OFF* e *ON* do tipo interno *DBP*.

Step Position Information: Conforme definido no item 7.3.1.5 da norma IEC 60870-5-101 essa variável de 8 bits é composta por dois campos: valor (definido pelos 7 bits menos significativos da variável) e transiente (definido como o bit mais significativo, o qual indica quando o dispositivo medido está transicionando).

Abaixo, exemplo de código para manipulação dos campos em uma variável do tipo USINT. Atenção, pois este código não consiste se o valor de entrada está dentro da faixa, portanto esta consistência fica a cargo do usuário.

```

PROGRAM UserPrg
VAR
usiVTI: USINT; // Valor com indicação de estado transiente, mapeado para o
  Cliente
siValue: SINT; // Valor a ser convertido para VTI. Deve estar entre -64 e +63
bTransient: BOOL; // Transiente a ser convertido para VTI
END_VAR

usiVTI := SINT_TO_USINT(siValue) AND 16#3F;
IF siValue < 0 THEN
usiVTI := usiVTI OR 16#40;
END_IF
IF bTransient THEN
usiVTI := usiVTI OR 16#80;
END_IF

```

PROFIBUS: Com exceção dos objetos digitais, os tipos de dados de objetos analógicos e contadores do protocolo IEC 60870-5-104 são diferentes dos tipos de dados dos módulos analógicos e contadores PROFIBUS, não sendo possível mapear tais tipos de variáveis PROFIBUS diretamente para os clientes IEC 60870-5-104.

Nestes casos é necessário criar uma variável intermediária, a ser mapeada no cliente IEC 60870-5-104, e converter os tipos adequadamente, como pode ser observado no código exemplo a seguir.

```
PROGRAM UserPrg
VAR
  iAnalogIn:    INT;
  iAnalogOut:   INT;
  diCounter:    DINT;
END_VAR

// Conversão da entrada analógica de WORD (PROFIBUS) para INT (IEC104)
iAnalogIn:= WORD_TO_INT(wNX6000in00);

// Conversão da saída analógica de INT (IEC104) para WORD (PROFIBUS)
wNX6100out00:= INT_TO_WORD(iAnalogOut);

// Conversão do contador de WORDs high+low (PROFIBUS) para DINT (IEC104)
diCounter:= DWORD_TO_DINT(ROL(WORD_TO_DWORD(wNX1005cnt00H), 16) OR wNX1005cnt00L
);
```

5.5.14.2. Pontos Duplos

Os pontos digitais duplos são utilizados para indicar a posição de equipamentos como válvulas, disjuntores e seccionadoras, onde a transição entre os estados aberto e fechado demandam um determinado tempo, podendo assim indicar um estado intermediário de transição entre os dois estados finais.

Pontos digitais duplos também são utilizados como saídas e, de uma forma análoga, é necessário manter uma das saídas acionada por determinado tempo para a transição ser completada. Tal acionamento é realizado através de pulsos, também conhecido por comandos trip/close, com determinada duração (suficiente para o chaveamento do dispositivo sob controle).

Consultar a seção Pontos Duplos do manual de utilização, para informações sobre a representação de pontos digitais duplos através dos tipos de dados DBP.

Uma vez que os módulos de entrada e saída digital da Série Nexto não suportam o mapeamento de pontos DBP, são necessárias algumas artimanhas de aplicação para tornar isto possível. Lembrando que também não é possível a utilização da função *PulsedCommand*, definido na biblioteca *LibRtuStandard*, para acionamento dos pontos digitais duplos na Série Nexto.

5.5.14.2.1. Pontos Duplos de Entrada Digital

Para módulos de entrada digital será necessária a declaração de duas variáveis auxiliares, a serem mapeadas no módulo de entrada digital, além do ponto duplo que se deseja mapear no servidor:

- a variável de valor do ponto duplo: tipo DBP
- a variável de valor do ponto simples OFF/TRIP: tipo BOOL
- a variável de valor do ponto simples ON/CLOSE: tipo BOOL

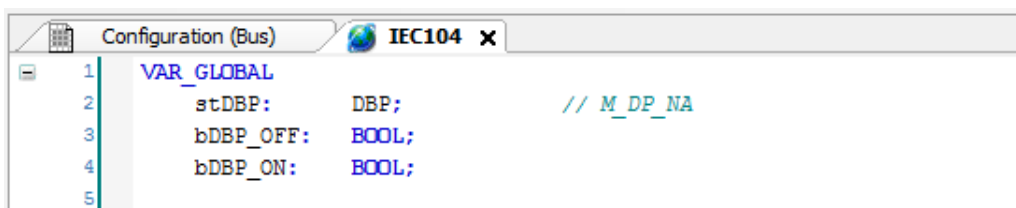


Figura 111: Exemplo de Declaração das Variáveis de Ponto Duplo

5. CONFIGURAÇÃO

Feita a declaração das variáveis, é necessário criar um vínculo entre a variável de valor e a qualidade do módulo de entrada digital, através da aba de *Pontos Internos* da UCP:

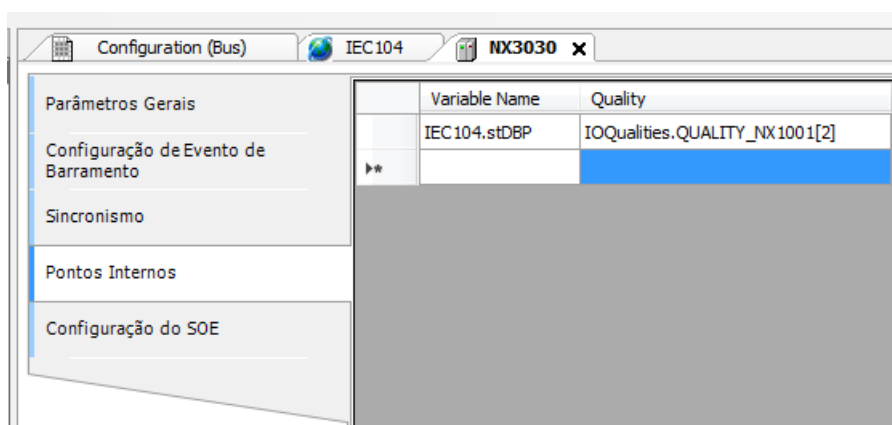


Figura 112: Atribuição das Variáveis de Ponto Duplo aos Pontos Internos

A variável de valor do ponto duplo deve ser mapeada no driver servidor IEC 60870-5-104, e as duas variáveis simples no módulo de entrada digital da Série Nexto (neste exemplo, um NX1001). Tipicamente, o estado OFF (TRIP) é mapeado na entrada par e o estado ON (CLOSE) na entrada ímpar.

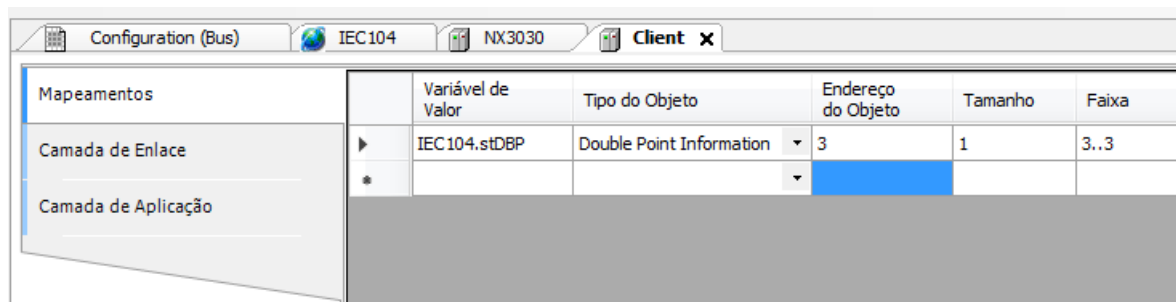


Figura 113: Mapeamento das variáveis de Ponto Duplo no Cliente IEC 60870-5-104

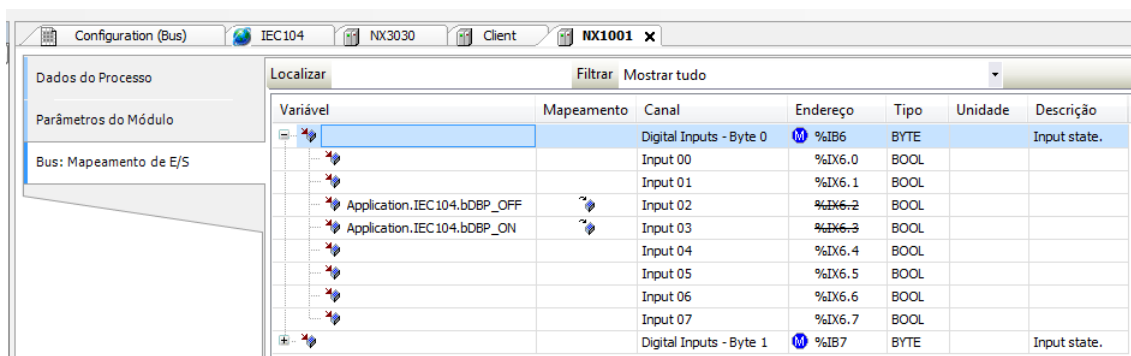


Figura 114: Mapeamento das Variáveis às Entradas do Módulo

Por último, o usuário deve inserir duas linhas de código em sua aplicação, a serem executadas ciclicamente, para atribuição dos valores das variáveis simples ao ponto duplo:

- variável de valor DBP, índice ON, recebe o valor do ponto simples ON

- variável de valor DBP, índice OFF, recebe o valor do ponto simples OFF

```

1  (*The main code inserted by the user and executed associated with the MainTask must be inserted into this POU.*)
2  PROGRAM UserPrg
3  VAR
4  END_VAR

1  //Double Points treatment
2  IEC104.stDBP.OFF := IEC104.bDBP_OFF;
3  IEC104.stDBP.ON := IEC104.bDBP_ON;
4

```

Figura 115: Atribuição dos Valores das Variáveis ao Ponto Duplo

5.5.14.2.2. Pontos Duplos de Saída Digital

Para módulos de saída digital deverá ser utilizado o bloco função *CommandReceiver* para a interceptação dos comandos de acionamento de pontos duplos oriundos dos clientes IEC 60870-5-104. Consultar a seção [Interceptação de Comandos Oriundos do Centro de Controle](#) para maiores informações.

O código exemplo abaixo, POU *CmdRcv*, trata comandos pulsados recebidos de clientes para um ponto digital duplo, mapeado num módulo NX2020. Além do código ST a seguir, é necessário mapear o ponto DBP no servidor IEC 60870-5-104 do Nexto.

Mapeamentos	Variável de Valor	Tipo do Objeto	Endereço do Objeto	Tamanho	Faixa	Variável de Contador	Variável de Banda Morta	Tipo da Banda Morta	Select Requerido	Pulso Curto (ms)	Pulso Longo (ms)
Camada de Enlace	CmdRcv.dbpIEC104	Double Command	1	1	1..1			Desabilitada	False	1000	2000
Camada de Aplicação											

Figura 116: Mapeamento das variáveis de Pontos Duplos de Saída Digital no Cliente IEC 60870-5-104

```

PROGRAM CmdRcv
VAR
CmdReceive:  CommandReceiver; // Instância do Interceptador
fbPulsedCmd: PulsedCommandNexto; // Instância do comando pulsado
byResult:   BYTE; // Resultado do comando pulsado
dbpIEC104:  DBP; // Variável mapeada no servidor IEC 104
bSetup:    BOOL:= TRUE; // Configuração inicial do interceptador
END_VAR

// Executa a configuração da função no primeiro ciclo
IF bSetup THEN
CmdReceive.dwVariableAddr:= ADR(dbpIEC104);
CmdReceive.bExec:= TRUE;
CmdReceive.eCommandResult:= COMMAND_RESULT.NONE;
CmdReceive.dwTimeout:= 256 * 10;
bSetup:= FALSE;
END_IF

// Caso um comando seja capturado:
IF CmdReceive.bCommandAvailable THEN

// Trata cada um dos comandos possíveis de serem interceptados
CASE CmdReceive.sCommand.eCommand OF

```

```
COMMAND_TYPE.NO_COMMAND:

    // Informar que existe um comando inválido.
    // Não faz nada. Deve sair por timeout.

COMMAND_TYPE.SELECT:

    // Trata apenas comandos para pontos duplos
    IF CmdReceive.sCommand.sSelectParameters.sValue.eParamType =
        DOUBLE_POINT_COMMAND THEN
        // Retorna comando finalizado com sucesso
        // (controlado pelo protocolo IEC104)
        byResult:= 7;
    ELSE
        // Retorna comando não suportado
        byResult:= 1;
    END_IF

COMMAND_TYPE.OPERATE:

    // Trata apenas comandos para pontos duplos
    IF CmdReceive.sCommand.sOperateParameters.sValue.eParamType =
        DOUBLE_POINT_COMMAND THEN
        // Geração do pulso nas saídas
        IF CmdReceive.sCommand.sOperateParameters.sValue.sDoublePoint.bValue THEN
            // Executa a função de TRIP
            fbPulsedCmd(
                byCmdType:= 101,
                byPulseTime:= DWORD_TO_BYTE(CmdReceive.sCommand.sOperateParameters.
                    sValue.sDoublePoint.sPulseConfig.dwOnDuration/10),
                ptDbpVarAdr:= ADR(dbpIEC104),
                stQuality:= IOQualities.QUALITY_NX2020[4],
                byStatus=> byResult);
        ELSE
            // Executa a função de CLOSE
            fbPulsedCmd(
                byCmdType:= 102,
                byPulseTime:= DWORD_TO_BYTE(CmdReceive.sCommand.sOperateParameters.
                    sValue.sDoublePoint.sPulseConfig.dwOffDuration/10),
                ptDbpVarAdr:= ADR(dbpIEC104),
                stQuality:= IOQualities.QUALITY_NX2020[5],
                byStatus=> byResult);
        END_IF
    ELSE
        // Retorna comando não suportado
        byResult:= 1;
    END_IF

COMMAND_TYPE.CANCEL:

    // Retorna comando finalizado com sucesso
    // (controlado pelo protocolo IEC104)
    byResult:= 7;

END_CASE
```

```

// Trata o resultado da função de comando pulsado
// e gera a resposta para o comando interceptado
CASE byResult OF
1: // Tipo inválido de comando
  CmdReceive.eCommandResult:= COMMAND_RESULT.NOT_SUPPORTED;
  CmdReceive.bDone:= TRUE;
2: // Parâmetros de entrada inválidos
  CmdReceive.eCommandResult:= COMMAND_RESULT.INCONSISTENT_PARAMETERS;
  CmdReceive.bDone:= TRUE;
3: // Mudança de parâmetro em execução
  CmdReceive.eCommandResult:= COMMAND_RESULT.PARAMETER_CHANGE_IN_EXECUTION;
  CmdReceive.bDone:= TRUE;
4: // Módulo não respondeu ao comando (ausente)
  CmdReceive.eCommandResult:= COMMAND_RESULT.HARDWARE_ERROR;
  CmdReceive.bDone:= TRUE;
5: // Comando iniciado e em execução (não retorna nada)
6: // Outro comando já foi enviado para este ponto e está em execução
  CmdReceive.eCommandResult:= COMMAND_RESULT.LOCKED_BY_OTHER_CLIENT;
  CmdReceive.bDone:= TRUE;
7: // Comando finalizado com sucesso
  CmdReceive.eCommandResult:= COMMAND_RESULT.SUCCESS;
  CmdReceive.bDone:= TRUE;
END_CASE

END_IF

CmdReceive();

IF CmdReceive.bDone THEN
CmdReceive.bDone:= FALSE;
END_IF

```

Como pode ser observado no código anterior, para auxiliar na geração de pulsos em saídas digitais duplas no Nexto, foi criado e utilizado um bloco função equivalente à função *PulsedCommand* da biblioteca *LibRtuStandard*. O bloco função *PulsedCommandNexto()* aparece codificado em linguagem ST.

```

FUNCTION_BLOCK PulsedCommandNexto
VAR_INPUT
byCmdType:   BYTE;      // Tipo comando:
                // 100 = status
                // 101 = close/on
                // 102 = trip/off
byPulseTime: BYTE;      // Duração do pulso (em centésimos de segundo)
ptDbpVarAdr: POINTER TO DBP; // Endereço variável DBP (pode ser a mapeada)
stQuality:   QUALITY;   // Qualidade do ponto DBP (do módulo digital)
END_VAR
VAR_OUTPUT
bON:        BOOL;      // Saída ímpar mapeada no módulo DO Nexto
bOFF:       BOOL;      // Saída par mapeada no módulo DO Nexto
byStatus:   BYTE:= 7; // Retorno da função:
                // 1 = comando inválido
                // 2 = tempo fora da faixa válida (2..255)
                // 3 = comando alterado em tempo de execução

```

```

        // 4 = módulo não respondeu ao comando (ausente)
        // 5 = comando iniciado ou em execução
        // 6 = já existe um comando ativo neste ponto
        // 7 = comando de pulso finalizado com sucesso
END_VAR
VAR
byState:    BYTE;    // Estado do bloco função
udiPulseEnd: UDINT; // Instante do fim do pulso
END_VAR

// Máquina de estados do PulsedCommandNexto
CASE byState OF

0: // Estado inicial, pronto para receber comandos:
CASE byCmdType OF

100:// Apenas retorna o último status

101: // Executar pulso ON:
// Valida a duração do pulso
IF byPulseTime > 1 THEN
// Verifica se já existe um comando ativo neste ponto
IF ptDbpVarAdr^.ON OR ptDbpVarAdr^.OFF THEN
// Retorna que já existe um comando ativo
byStatus:= 6;
ELSE
// Liga saída CLOSE
ptDbpVarAdr^.ON:= TRUE;
ptDbpVarAdr^.OFF:= FALSE;
// Próximo estado: executar pulso ON
byState:= byCmdType;
// Retorna comando iniciado
byStatus:= 5;
END_IF
ELSE
// Retorna pulso fora da faixa
byStatus:= 2;
END_IF

102: // Executar pulso OFF
// Valida a duração do pulso
IF byPulseTime > 1 THEN
// Verifica se já existe um comando ativo neste ponto
IF ptDbpVarAdr^.ON OR ptDbpVarAdr^.OFF THEN
// Retorna que já existe um comando ativo
byStatus:= 6;
ELSE
// Liga saída TRIP
ptDbpVarAdr^.ON:= FALSE;
ptDbpVarAdr^.OFF:= TRUE;
// Próximo estado: executar pulso OFF
byState:= byCmdType;
// Retorna comando iniciado
byStatus:= 5;
END_IF
ELSE

```

```

        // Retorna pulso fora da faixa
        byStatus:= 2;
    END_IF
ELSE
    // Retorna comando inválido
    byStatus:= 1;
END_CASE

// Memoriza instante de término do pulso
udiPulseEnd:= SysTimeGetMs() + BYTE_TO_UDINT(byPulseTime) * 10;

101, 102:// Continua execução do pulso ON/OFF
// A princípio retorna que comando está em execução
byStatus:= 5;
// Verifica mudança de parâmetro em execução
IF byCmdType <> 100 AND byCmdType <> byState THEN
    // Retorna mudança de parâmetro em execução
    byStatus:= 3;
END_IF
// Verifica término do pulso
IF SysTimeGetMs() >= udiPulseEnd THEN
    // Desliga saídas TRIP e CLOSE
    ptDbpVarAdr^.ON:= FALSE;
    ptDbpVarAdr^.OFF:= FALSE;
    // Retorna comando finalizado, apenas se comando não mudou
    IF byCmdType = 100 OR byCmdType = byState THEN
        byStatus:= 7;
    END_IF
    // Próximo estado: inicial
    byState:= 0;
END_IF

END_CASE

// Verifica qualidade do módulo digital (do ponto DBP)
IF stQuality.VALIDITY <> QUALITY_VALIDITY.VALIDITY_GOOD THEN
    // Desliga saídas TRIP e CLOSE
    ptDbpVarAdr^.ON:= FALSE;
    ptDbpVarAdr^.OFF:= FALSE;
    // Retorna que módulo está ausente
    byStatus:= 4;
    // Próximo estado: inicial
    byState:= 0;
END_IF

// Copia estados da saída DBP para as saídas simples
bON:= ptDbpVarAdr^.ON;
bOFF:= ptDbpVarAdr^.OFF;

```

5.5.14.3. Parâmetros Gerais

Para as configurações dos *Parâmetros Gerais* de um Servidor 60870-5-104 conforme a figura abaixo seguem os parâmetros da tabela abaixo apresentadas a seguir:

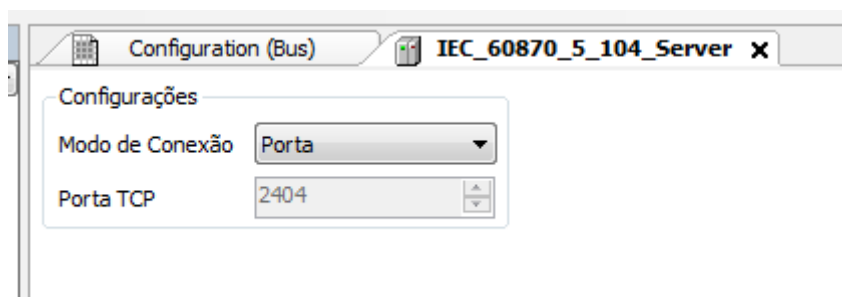


Figura 117: Tela de Parâmetros Gerais do Servidor IEC 60870-5-104

Configuração	Descrição	Padrão	Opções
Modo de Conexão	Configura o modo de conexão com os módulos Cliente Conectados.	Porta	Porta IP
Porta TCP	Define qual o número da porta TCP do CP será utilizada para a comunicação com os módulos Cliente Conectados, caso o campo "Modo de Conexão" esteja selecionado "IP".	2404	1 a 65535

Tabela 141: Configuração dos Parâmetros Gerais do Servidor IEC 60870-5-104

5.5.14.4. Mapeamento dos Dados

Para a configuração das relações de dados do Servidor IEC60870-5-104, visualizadas na figura abaixo, seguem os parâmetros descritos na tabela abaixo:

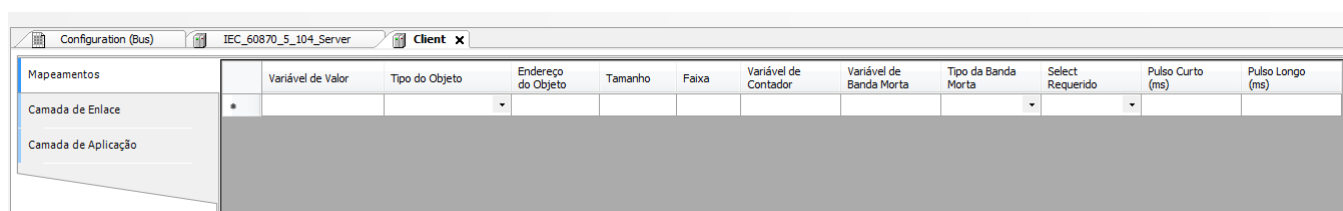


Figura 118: Tela de Mapeamento dos Dados do Servidor IEC 60870-5-104

Configuração	Descrição	Padrão	Opções
Variável de Valor	Nome da variável simbólica	-	Nome de uma variável declarada em um programa ou GVL
Tipo do Objeto	Configuração do tipo do objeto IEC 60870-5-104	-	Single Point Information Double Point Information Step Position Information Measured Value (Normalized) Measured Value (Scaled) Measured Value (Short Floating Point) Integrated Totals Bitstring Information (32 Bits) Single Command Double Command Regulating Step Command Setting Point Command (Normalized) Setting Point Command (Scaled) Setting Point Command (Short Floating Point) Bitstring Command (32 Bits)
Endereço do Objeto	Índice do primeiro ponto do mapeamento IEC 60870-5-104	-	1 a 65535
Tamanho	Especifica a quantidade máxima de dados que um mapeamento IEC 60870-5-104 poderá acessar	-	1 a 86400000
Faixa	Faixa de endereços do dado configurado	-	-
Variável de Contador	Nome da variável simbólica que vai conter o dado da variável contadora	-	Nome de uma variável declarada em um programa, GVL ou módulo contador
Variável de Banda Morta	Nome da variável simbólica que vai conter o dado da banda morta	-	Nome de uma variável declarada em um programa ou GVL
Tipo da Banda Morta	Define o tipo de Banda Morta a ser utilizada no mapeamento	Desabilitada	Absoluta Desabilitada Integrada
Select Requerido	Define se para a execução de um comando é requerido ou não um select prévio	False	True False
Pulso Curto (ms)	Define o tempo do pulso curto de um comando digital IEC 60870-5-104	1000	1 a 86400000

Configuração	Descrição	Padrão	Opções
Pulso Longo (ms)	Define o tempo do pulso longo de um comando digital IEC 60870-5-104	2000	1 a 86400000

Tabela 142: Configuração dos Mapeamentos do Servidor IEC 60870-5-104

Notas:

Variável de Valor: Nome da variável simbólica a ser mapeada. Quando um comando de leitura é enviado, o retorno enviado na resposta está armazenado nesta variável. Quando for um comando de escrita, o valor escrito será armazenado nesta variável. A variável pode ser simples, array ou elemento de array e pode estar em estruturas.

Variável de Contador: Este campo se aplica somente no mapeamento de objetos do tipo *Integrated Totals*, sendo esta a variável contadora, a ser manipulada no processo. Ela deve ter o mesmo tipo e tamanho da variável mapeada na coluna *Variável de Valor*, cujo valor será lido pelo cliente ou reportado ao cliente em caso de eventos.

ATENÇÃO

Quando *Variável de Contador* tem associada uma variável de qualidade, para que esta qualidade seja transferida para a variável congelada no comando de freeze, deve ser associada uma variável de qualidade à variável congelada. Este procedimento pode ser feito através da aba *Pontos Internos*.

Variável de Banda Morta: Este campo se aplica somente à mapeamento de variáveis analógicas de entrada (objetos tipo *Measured Value*). Ela deve ter o mesmo tipo e tamanho da variável mapeada na coluna *Variável de Valor*. Novos valores da variável de banda morta serão considerados somente quando a variável analógica de entrada mudar de valor.

Tipo da Banda Morta: Os tipos de configuração da Banda Morta disponíveis são os seguintes:

Tipo de Função	Configurações	Descrição
Tipo da Banda Morta	Desabilitada	Nesta opção, qualquer modificação de valor num ponto do grupo, por menor que seja, gera evento para este ponto.
	Absoluta	Nesta opção, se o módulo da variação do valor de um ponto do grupo for maior do que o valor configurado pela variável no campo “Banda Morta”, um evento para este ponto é gerado.
	Integrada	Nesta opção, se o módulo da integral da variação do valor de um ponto do grupo for maior do que o valor configurado pela variável no campo “Banda Morta”, um evento para este ponto é gerado. O intervalo de integração é de um segundo.

Tabela 143: Tipos de Banda Morta dos Mapeamentos do Servidor IEC 60870-5-104

Pulso Curto e Pulso Longo: Ao definir o tempo de duração dos pulsos curtos e longos deve se levar em consideração os limites suportados pelo dispositivo que irá tratar o comando. Por exemplo, caso o destino seja um cartão de saída, o que não é suportado de forma nativa pela Série Nexto, deve se verificar na CT do módulo quais os tempos mínimos e máximos, bem como a resolução, para a execução de comandos pulsados.

5.5.14.5. Camada de Enlace

Para a configuração dos parâmetros da camada de enlace do Servidor IEC 60870-5-104, visualizada na figura abaixo, seguem os parâmetros descritos na tabela abaixo:

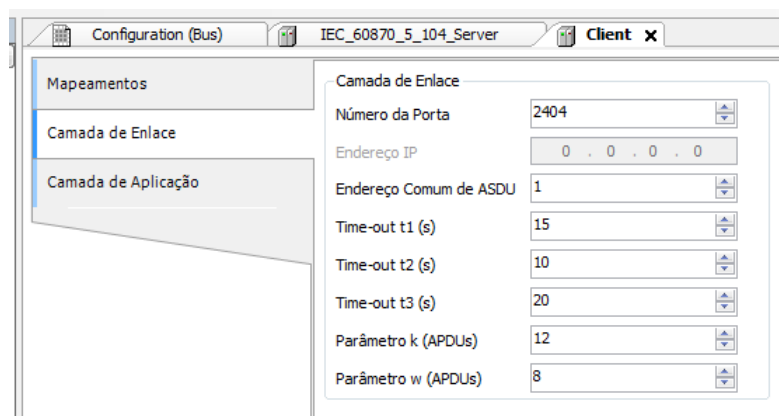


Figura 119: Tela de Configuração da Camada de Enlace do Servidor IEC 60870-5-104

Parâmetro	Descrição	Padrão	Opções
Número da Porta	Endereço da porta listen para conexão do cliente. Utilizado quando a conexão do cliente não é por IP	2404	1 a 65535
Endereço IP	IP do cliente conectado, utilizado quando a conexão do cliente é por IP	0.0.0.0	1.0.0.1 a 223.255.255.254
Endereço Comum de ASDU	Endereço IEC 60870-5-104 do cliente conectado	1	1 a 65534
Time-out t1 (s)	Período de tempo (em segundos) que o dispositivo aguarda o recebimento de uma mensagem de confirmação após o envio de uma mensagem APDU do tipo I ou U (dados) antes de fechar a conexão	15	1 a 180
Time-out t2 (s)	Período de tempo (em segundos) que o dispositivo aguarda para enviar uma mensagem supervisória (S-Frame) confirmando o recebimento de frames de dados	10	1 a 180
Time-out t3 (s)	Período de tempo (em segundos) em que será enviada uma mensagem para teste do link caso não haja nenhuma transmissão por ambos os lados	20	1 a 180
Parâmetro k (APDUs)	Número máximo de mensagens de dados (I-Frame) transmitidas e não confirmadas	12	1 a 12
Parâmetro w (APDUs)	Número máximo de mensagens de dados (I-Frame) recebidas e não confirmadas	8	1 a 8

Tabela 144: Configuração da Camada de Enlace do Servidor IEC 60870-5-104

Nota:

Os campos *Time-out t1 (s)*, *Time-out t2 (s)* e *Time-out t3 (s)* são dependentes entre si e devem ser configurados de tal forma que *Time-out t1 (s)* seja maior do que *Time-out t2 (s)* e *Time-out t3 (s)* seja maior do que *Time-out t1 (s)*. Se alguma destas regras não for respeitada, mensagens de erro serão exibidas na compilação do projeto.

ATENÇÃO

Para links de comunicação lentos (exemplo: comunicação via satélite), deve se ajustar adequadamente os parâmetros *Time-out t1 (s)*, *Time-out t2 (s)* e *Time-out t3 (s)* como por exemplo, dobrando os valores padrões desses campos.

5.5.14.6. Camada de Aplicação

Para a configuração da camada de aplicação do Servidor IEC 60870-5-104, visualizada na figura abaixo, seguem os parâmetros descritos na tabela abaixo:

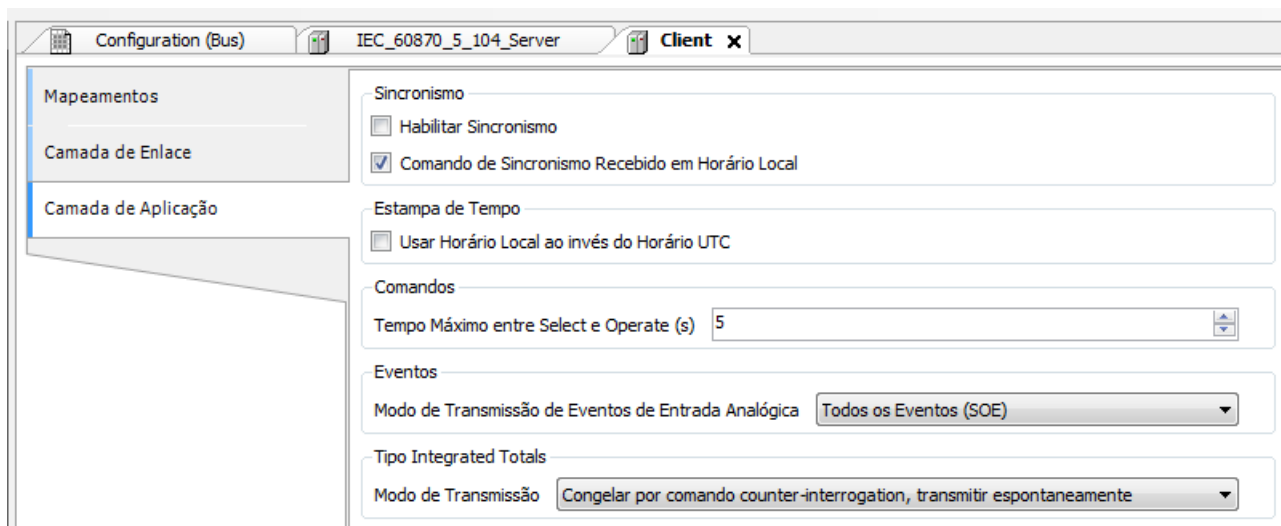


Figura 120: Tela de Configuração da Camada de Aplicação do Servidor IEC 60870-5-104

Parâmetro	Descrição	Padrão	Opções
Habilitar Sincronismo	Opção para Habilitar/Desabilitar a solicitação de sincronismo de tempo	Desabilitado	Desabilitado Habilitado
Comando de Sincronismo Recebido em Horário Local	Opção para Habilitar/Desabilitar o tratamento do comando de sincronização em horário local	Habilitado	Desabilitado Habilitado
Usar Horário Local ao invés do Horário UTC	Opção para Habilitar/Desabilitar a estampa de tempo em horário local para eventos	Desabilitado	Desabilitado Habilitado

Parâmetro	Descrição	Padrão	Opções
Tempo Máximo entre Select e Operate (s)	Período de tempo no qual o comando de seleção permanecerá ativo (inicia a contagem a partir da confirmação da recepção do comando de seleção) aguardando o comando de Operate	5	1 a 180
Modo de Transmissão de Eventos de Entrada Analógica	Modo de transmissão dos eventos de entradas analógicas	Todos os Eventos (SOE)	Todos os Eventos (SOE) Evento Mais Recente
Modo de Transmissão	Modo de transmissão dos contadores congelados (integrated totals).	Congelar por comando counter-interrogation, transmitir espontaneamente	Congelar por comando counter-interrogation, transmitir espontaneamente Congelar e transmitir por comando counter-interrogation

Tabela 145: Configuração da Camada de Aplicação do Servidor IEC 60870-5-104

Notas:

Habilitar Sincronismo: Uma vez habilitado, permite o Servidor IEC 60870-5-104 ajustar o relógio da UCP quando recebido um comando de sincronização.

Comando de Sincronismo Recebido em Horário Local: Quando habilitado, o IEC 60870-5-104 Servidor ajusta o relógio da UCP tratando o horário recebido no comando de sincronização como horário local. Caso contrário esse horário é considerado UTC.

Usar Horário Local ao invés do Horário UTC: Uma vez habilitado, a estampa de tempo dos eventos gerados pelo IEC 60870-5-104 Servidor será enviada conforme o horário local da UCP.

ATENÇÃO

Quando a opção de sincronismo de tempo for marcada em mais de um servidor, os tempos recebidos nos diferentes servidores serão sobrescritos no relógio do sistema em um curto espaço de tempo, podendo causar comportamentos indesejáveis devido a atrasos no tempo de propagação das mensagens e carga do sistema.

Modo de Transmissão de Eventos de Entrada Analógica: Os modos de transmissão dos eventos de entradas analógicas disponíveis são os seguintes:

Tipo de Função	Configurações	Descrição
Modo de Transmissão de Eventos de Entrada Analógica	Todos os Eventos (SOE)	São enviados todos os eventos analógicos gerados.
	Evento Mais Recente	É enviado somente o evento mais recente de cada ponto analógico.

Tabela 146: Modos de Transmissão dos Eventos de Entradas Analógicas do Servidor IEC 60870-5-104

Modo de Transmissão: Os modos de transmissão dos contadores congelados (*integrated totals*) disponíveis são os seguintes:

Tipo de Função	Configurações	Descrição
Modo de Transmissão	Congelar por comando counter-interrogation, transmitir espontaneamente	Equivale ao Modo D de aquisição de contadores (integrated totals) definido pela norma IEC 60870-5-101. Neste modo, comandos de interrogação de contadores (freeze) da estação de controle realizam o congelamento dos contadores. Caso os valores congelados tenham se modificado, eles são reportados através de eventos.
	Congelar e transmitir por comando counter-interrogation	Equivale ao Modo C de aquisição de contadores (integrated totals) definido pela norma IEC 60870-5-101. Neste modo, comandos de interrogação de contadores (freeze) da estação de controle realizam o congelamento dos contadores. Comandos subsequentes de interrogação dos contadores (read) são enviados pela estação de controle para obter os valores congelados.

Tabela 147: Modos de Transmissão dos Contadores Congelados do Servidor IEC 60870-5-104

ATENÇÃO

A norma IEC 60870-5-104, seção *Transmission control using Start/Stop*, prevê a utilização de comandos *STARTDT* e *STOPDT* para controle do tráfego de dados entre o cliente e o servidor, utilizando conexões simples ou múltiplas conexões. Apesar do Nexto suportar tais comandos, a utilização dos mesmos não é recomendada para controlar a transmissão dos dados, principalmente com CPs redundantes, pois tais comandos não são sincronizados entre os dois CPs. Em vez de utilizar múltiplas conexões entre o cliente e o servidor Nexto, sugere-se a utilização do recurso de NIC Teaming para prover canais Ethernet (fisicamente) redundantes e preservar os recursos da UCP (centros de controle por UCP).

5.5.14.7. Diagnósticos do Servidor

Os diagnósticos do protocolo Servidor IEC 60870-5-104, são armazenados em variáveis do tipo *T_DIAG_IEC104_SERVER_1* as quais estão descritas na tabela abaixo:

Variável de diagnóstico do tipo <i>T_DIAG_IEC104_SERVER_1.*</i>	Tamanho	Descrição
Bits de comando, reiniciados automaticamente:		
tCommand.bStop	BOOL	Desabilitar Driver
tCommand.bStart	BOOL	Habilitar o Driver
tCommand.bDiag_01_Reserved	BOOL	Reservado
tCommand.bDiag_02_Reserved	BOOL	Reservado
tCommand.bDiag_03_Reserved	BOOL	Reservado
tCommand.bDiag_04_Reserved	BOOL	Reservado
tCommand.bDiag_05_Reserved	BOOL	Reservado
tCommand.bDiag_06_Reserved	BOOL	Reservado
Diagnósticos:		
tClient_X.bRunning	BOOL	IEC 60870-5-104 Server está sendo executado

Variável de diagnóstico do tipo T_DIAG_IEC104_SERVER_1.*	Tamanho	Descrição
tClient_X.eConnectionStatus. CLOSED	ENUM(BYTE)	Canal de comunicação fechado. Servidor não irá aceitar solicitação de conexão. Valor do ENUM (0)
tClient_X.eConnectionStatus. LISTENING		Servidor está escutando a porta configurada, e não existem clientes conectados. Valor do ENUM (1)
tClient_X.eConnectionStatus. CONNECTED		Cliente conectado. Valor do ENUM (2)
tClient_X.tQueueDiags. bOverflow	BOOL	Fila do cliente está com estouro
tClient_X.tQueueDiags. wSize	WORD	Tamanho da fila configurada
tClient_X.tQueueDiags. wUsage	WORD	Número de eventos na fila
tClient_X.tQueueDiags. dwReserved_0	DWORD	Reservado
tClient_X.tQueueDiags. dwReserved_1	DWORD	Reservado
tClient_X.tStats.wRXFrames	WORD	Número de quadros (frames) recebidos
tClient_X.tStats.wTXFrames	WORD	Número de quadros (frames) enviados
tClient_X.tStats.wCommErrors	WORD	Contador de erros de comunicação incluindo erros na Camada Física, Camada de Enlace e Camada de Transporte.
tClient_X.tStats.dwReserved_0	DWORD	Reservado
tClient_X.tStats.dwReserved_1	DWORD	Reservado

Tabela 148: Diagnósticos do Servidor IEC 60870-5-104

5.5.14.8. Qualificador dos Comandos

A norma IEC 60870-5-104 prevê quatro diferentes qualificadores de comandos para os objetos *Single Command*, *Double Command* e *Regulating Step Command*, todos suportados pelo Servidor do Nexto.

Cada tipo de objeto tem um comportamento específico para cada qualificador de comando, como pode ser observado na tabela apresentada a seguir.

Qualificador	Tipo do objeto do protocolo IEC 60870-5-104		
	Single Command	Double Command	Regulating Step Command
No additional definition (padrão)	Mesmo comportamento do qualificador persistente.	Mesmo comportamento do qualificador de pulso curto.	Mesmo comportamento do qualificador de pulso curto.
Short pulse duration (pulso curto)	Requer interceptação do comando para tratamento pela aplicação, caso contrário irá retornar uma mensagem de confirmação negativa (falha).	Requer interceptação do comando para tratamento pela aplicação, caso contrário irá retornar uma mensagem de confirmação negativa (falha).	Requer interceptação do comando para tratamento pela aplicação, caso contrário irá retornar uma mensagem de confirmação negativa (falha).
Long pulse duration (pulso longo)			
Persistent output (persistente)	A saída será ligada ou desligada, e assim permanecerá até novo comando, conforme valor (ON ou OFF) comandado pelo cliente.		

Tabela 149: Qualificador dos Comandos do Servidor IEC 60870-5-104

Nota:

Interceptação do comando: Para maiores informações sobre a interceptação de comandos de clientes IEC 60870-5-104, consultar a seção [Interceptação de Comandos Oriundos do Centro de Controle](#), implementada através do bloco de função *CommandReceiver*.

5.5.15. PROFINET Controller

Para a correta utilização do protocolo PROFINET Controller é necessário consultar o manual MU214621 - Nexto Series PROFINET Manual.

5.6. Desempenho de Comunicação**5.6.1. Servidor MODBUS**

Os dispositivos MODBUS configuráveis na UCP Nexto executam em segundo plano, com uma prioridade abaixo da aplicação de usuário e de forma cíclica. Sendo assim, seu desempenho irá variar de acordo com o tempo restante, levando em consideração a diferença entre o intervalo e tempo que a aplicação leva para ser executada. Por exemplo, um dispositivo MODBUS em uma aplicação que é executada a cada 100 ms, com um tempo de execução de 50 ms, irá ter um desempenho menor do que com uma aplicação de 50 ms executando a cada 200 ms de intervalo. Isto acontece porque, no segundo caso, a UCP terá um tempo maior entre cada ciclo da MainTask para executar as tarefas com prioridade mais baixa.

Também deve-se levar em conta o número de ciclos que o dispositivo, escravo ou servidor, leva para responder a uma requisição. Para processar e transmitir uma resposta, um MODBUS RTU Escravo irá levar dois ciclos (tempo do ciclo da tarefa MODBUS), enquanto que um MODBUS Ethernet Servidor levará apenas um ciclo. Porém, esse é o tempo mínimo entre a recepção de uma requisição e o envio da resposta. Caso a requisição seja enviada logo após a execução de um ciclo da tarefa MODBUS, o tempo poderá ser equivalente a 2 ou 3 vezes o tempo de ciclo para o MODBUS Escravo, e de 1 a 2 vezes o tempo de ciclo para o MODBUS Servidor.

Neste caso: Tempo Máximo de Resposta = 3* (tempo de ciclo) + (tempo de execução das tarefas) + (tempo interframe chars) + (tempo de atraso do envio).

Por exemplo, para uma tarefa MODBUS Ethernet Servidor com um ciclo de 50 ms, em uma aplicação executada por 60 ms a cada 100 ms, o servidor conseguirá executar apenas um ciclo entre cada ciclo da aplicação. Por outro lado, com a mesma aplicação, sendo executada por 60 ms, mas com um intervalo de 500 ms, o MODBUS terá um desempenho melhor, pois enquanto a aplicação não estiver em execução, o mesmo estará sendo executado a cada 50 ms e apenas a cada ciclo da MainTask ele irá demorar mais para ser executado. Para estes casos, o pior desempenho será a soma do tempo de execução da aplicação do usuário com o tempo do ciclo da tarefa MODBUS.

Para os dispositivos mestre e cliente o princípio de funcionamento é exatamente igual, mas levando em consideração o tempo de varredura da relação MODBUS e não o tempo do ciclo da tarefa MODBUS. Para estes casos, o pior desempenho de uma relação será executado após o seu tempo de varredura, somado ao tempo de execução da aplicação de usuário.

É importante ressaltar que o número de dispositivos MODBUS em execução também irá alterar o seu desempenho. Em uma aplicação de usuário com tempo de execução de 60 ms e intervalo de 100 ms, irá restar 40 ms para a UCP executar todas as tarefas de menor prioridade. Portanto, uma UCP com apenas um MODBUS Ethernet Servidor terá um desempenho maior do que uma UCP que utilize quatro destes dispositivos.

5.6.1.1. Interfaces Locais da UCP

Para um dispositivo MODBUS Ethernet Servidor, podemos afirmar que ele é capaz de responder a um número x de requisições por segundo, ou seja, será capaz de transferir n bytes por segundo, dependendo do tamanho de cada requisição. Quanto menor for o ciclo da tarefa do Servidor MODBUS, maior será o impacto do número de conexões em sua taxa de resposta. Porém, para tempos de ciclo menores que 20 ms este impacto não é linear, devendo ser consultada a tabela abaixo para informações.

A tabela abaixo exemplifica o número de requisições respondidas por um Servidor MODBUS inserido em uma interface local da UCP, em função do tempo de ciclo configurado para a tarefa e do número de conexões ativas efetuando requisições:

Número de Conexões	Requisições respondidas por segundo com tempo de ciclo da tarefa MODBUS em 5 ms	Requisições respondidas por segundo com tempo de ciclo da tarefa MODBUS em 10 ms	Requisições respondidas por segundo com tempo de ciclo da tarefa MODBUS em 20 ms
1 Conexão	185	99	50
2 Conexões	367	197	100
4 Conexões	760	395	200
7 Conexões	1354	695	350
10 Conexões	1933	976	500

Tabela 150: Taxa de Comunicação de um Servidor MODBUS em uma Interface Local

ATENÇÃO

Os desempenhos de comunicação mencionados neste capítulo são exemplos utilizando uma UCP apenas com um dispositivo MODBUS TCP Servidor, sem nenhuma lógica na aplicação que possa atrasar a comunicação. Portanto, estes desempenhos devem ser tomados como máximos.

Já para tempos de ciclo iguais ou maiores do que 20 ms, o crescimento da taxa de respostas é linear, podendo ser calculada através da fórmula:

$$N = C \times (1 / T)$$

Onde:

N é o número médio de respostas por segundo;

C é o número de conexões estabelecidas;

T é o intervalo da tarefa MODBUS em segundos.

Tomando como exemplo um Servidor MODBUS com uma conexão e um tempo de ciclo de 50 ms temos:

$$C = 1; T = 0,05 \text{ s};$$

$$N = 1 \times (1 / (0,05))$$

$$N = 20$$

Ou seja, nesta configuração o Servidor MODBUS será capaz de responder, em média, 20 requisições por segundo.

Caso este valor obtido seja multiplicado pelo número de bytes em cada requisição, será obtida uma taxa de transferência de n bytes por segundo.

5.6.1.2. Interfaces Remotas

O desempenho de um dispositivo MODBUS Servidor em uma interface Ethernet remota é parecido com o desempenho nas interfaces locais da UCP.

Porém, devido ao tempo de comunicação entre a UCP e os módulos, o máximo desempenho será limitado. Para apenas uma conexão aberta, o número de respostas é limitado em, no máximo, 18 respostas por segundo. Com mais conexões abertas, o número de respostas aumentará linearmente, exatamente conforme para as interfaces locais, porém sendo limitado a, no máximo, 90 respostas por segundo. Portanto, para uma interface Ethernet remota teremos as seguintes formas para calcular seu desempenho:

Para $T \leq 55$ ms usa-se:

$$N = C \times (18,18 - (18,18 / (0,055 \times 1000)))$$

E para $T \geq 55$ ms usa-se:

$$N = C \times (Z - (Z / (T \times 1000)))$$

Onde N é o número médio de respostas por segundo, C é o número de conexões e T é igual ao tempo de ciclo da tarefa MODBUS (em segundos).

Deve-se prestar atenção ao fato de que o desempenho máximo de um dispositivo MODBUS Servidor em uma interface Ethernet remota será de 90 respostas de requisições por segundo.

5.6.2. Servidor OPC DA

Para os testes de desempenho da comunicação com Servidor OPC DA foram criados projetos para o CP declarando variáveis do tipo INT e separando-as em POU's com 1.000 variáveis cada. Em todos os cenários foram utilizados projetos com perfil *Simple* e o Intervalo da MainTask configurado em 100 ms. Para realizar a comunicação, foi utilizado o atributo *attribute 'symbol' := 'readwrite'*, a fim de disponibilizar os dados no Servidor OPC DA. Também é importante ressaltar que as medições foram feitas com o software MasterTool desconectado da UCP e com a duração da MainTask ajustada para consumir 5%, 50% e 80% do Intervalo configurado, como apresentado na tabela abaixo.

No ponto de vista do OPC DA Cliente, foi utilizado um driver num sistema SCADA. O tempo de atualização configurado foi de 50 ms. Os resultados do desempenho nestas condições estão descritos na tabela abaixo.

Número Total de variáveis no projeto do CP	Tempo de atualização das variáveis no OPC DA Cliente		
	5% da UCP Ocupada	50% da UCP Ocupada	80% da UCP Ocupada
1.000	600 ms	800 ms	1400 ms
2.000	800 ms	900 ms	2800 ms
5.000	1000 ms	2000 ms	6500 ms
10.000	2000 ms	4000 ms	13700 ms
15.000	3200 ms	6400 ms	20000 ms
20.000	4000 ms	8100 ms	25000 ms

Tabela 151: Taxa de Comunicação de um Servidor OPC DA

5.6.3. Servidor OPC UA

O manual do OPC UA Server MU214609 analisa o desempenho da comunicação OPC UA com maiores detalhes, inclusive abordando o consumo de banda de comunicação Ethernet. Este manual também aborda conceitos sobre o funcionamento do protocolo OPC UA.

5.6.4. Servidor IEC 60870-5-104

O driver Servidor IEC 60870-5-104 é executado pela UCP da mesma forma como os outros drivers de comunicação Servidores, isto é, em segundo plano, com uma prioridade abaixo da aplicação de usuário e de forma cíclica. A tarefa deste driver especificamente executa a cada 50 ms, e basta 1 ciclo de execução do driver para processar e responder requisições. Contudo, por se tratar de uma tarefa de baixa prioridade, não é garantido que consiga executar com esta periodicidade pois depende do percentual de UCP livre (diferença entre o intervalo da MainTask e tempo que a aplicação do usuário leva para ser executada) e também da concorrência com tarefas de outros protocolos configurados na UCP.

Para auxiliar na compreensão do desempenho do driver IEC 60870-5-104 Servidor, a seguir é apresentado o resultado de testes realizados com um simulador Cliente IEC 60870-5-104 conectado a uma NX3030 com um driver IEC 60870-5-104 Servidor. A base de dados configurada era composta de 900 pontos digitais e 100 pontos analógicos (todos com qualidade e estampa de tempo), e o tempo de execução da MainTask estava em 70 ms (intervalo de 100 ms).

- Tempo para completar um comando de interrogação geral: menos de um segundo
- Tempo para transferir 900 eventos digitais + 100 eventos analógicos: 6 segundos

5.7. Desempenho do Sistema

Em casos onde a aplicação possui apenas uma tarefa de usuário MainTask, responsável pela execução de uma única unidade de programação do tipo Programa denominada MainPrg (Como no perfil Simples), o CP consome um determinado tempo para que a tarefa seja processada. A esse tempo damos o nome de *Tempo de Execução*.

Em uma aplicação, podemos conhecer o *Tempo de Execução* médio da aplicação usando o MasterTool IEC XE, na *Árvore de Dispositivos*, no item *Device*, no seguinte caminho:

CP Logic-> Application-> Task Configuration na aba *Monitorar*, na coluna *Tempo de Ciclo Médio*.

Deve-se atentar ao *Tempo de Execução* para que ele não ultrapasse 80% do intervalo configurado na tarefa de usuário MainTask. Por exemplo, em uma aplicação onde o intervalo é de 100 ms, um *Tempo de Execução* adequado é de até 80 ms. Isso se deve ao fato de que a UCP necessita de um tempo para a execução de outras tarefas como o processamento da comunicação, tratamento do visor e cartão de memória, e essas tarefas também acontecem dentro do intervalo (os 20% restantes do *Tempo de Execução*).

ATENÇÃO

Para tempos de ciclo muito altos (tipicamente tempos maiores que 300 ms), mesmo que o percentual de 80% seja respeitado, pode haver uma diminuição no tempo de resposta do visor e da tecla de diagnóstico. Caso o percentual de 80% não seja respeitado e o tempo de execução da tarefa de usuário se aproxime ou exceda o intervalo configurado da MainTask, o visor e a tecla de diagnóstico podem não responder, uma vez que sua prioridade na execução do sistema é menor do que as tarefas de usuário. Caso uma aplicação com erros seja carregada na UCP, pode ser necessário reinicializá-la sem carregar esta aplicação, como descrito na seção Log de Sistema.

ATENÇÃO

Os logs de sistema das UCPs da série Nexto, a partir da versão de firmware 1.4.0.33 são recarregados no caso de uma reinicialização da UCP ou por uma reinicialização do *Runtime System*, isto é, será possível visualizar os logs mais antigos quando uma dessas condições ocorrer.

5.7.1. Escaneamento de E/S

Para um projeto que utilize de módulos de E/S digitais, estando estes inseridos no barramento e declarados no projeto, o tempo da MainTask irá aumentar de acordo com o número de módulos. A tabela abaixo exemplifica o tempo médio que é acrescido à MainTask:

Módulos declarados no barramento	Acréscimo de tempo no ciclo da MainTask (μ s)
5	300
10	700
20	1000

Tabela 152: Tempo de Escaneamento de E/S

Em projetos que utilizem E/S remotas, como, por exemplo, utilizando o módulo Mestre PROFIBUS DP NX5001, o manual do respectivo módulo deve ser consultado para informações sobre desempenho e influências do módulo na execução das tarefas de usuário.

5.7.2. Cartão de Memória

A transferência de dados envolvendo o cartão de memória é realizada em segundo plano pela UCP, pois esta dá prioridade para a execução da aplicação de usuário e processamento da comunicação. Desta forma, a transferência de arquivos para o cartão poderá sofrer um acréscimo de tempo significativo, dependendo do tempo de execução da aplicação do usuário.

O tempo necessário para ler/escrever arquivos no cartão será afetado diretamente pelo tempo de execução da aplicação de usuário, uma vez que essa aplicação tem prioridade na execução.

Para mais informações sobre a utilização do cartão de memória, ver seção [Cartão de Memória](#).

5.8. Relógio RTC

As UCPs possuem um relógio interno que pode ser utilizado através da biblioteca *NextoStandard.lib*. Essa biblioteca é carregada automaticamente durante a criação de um novo projeto (para realizar o procedimento de inserção de uma biblioteca, consultar a seção [Bibliotecas](#)). A figura abaixo mostra os blocos e funções disponibilizados:

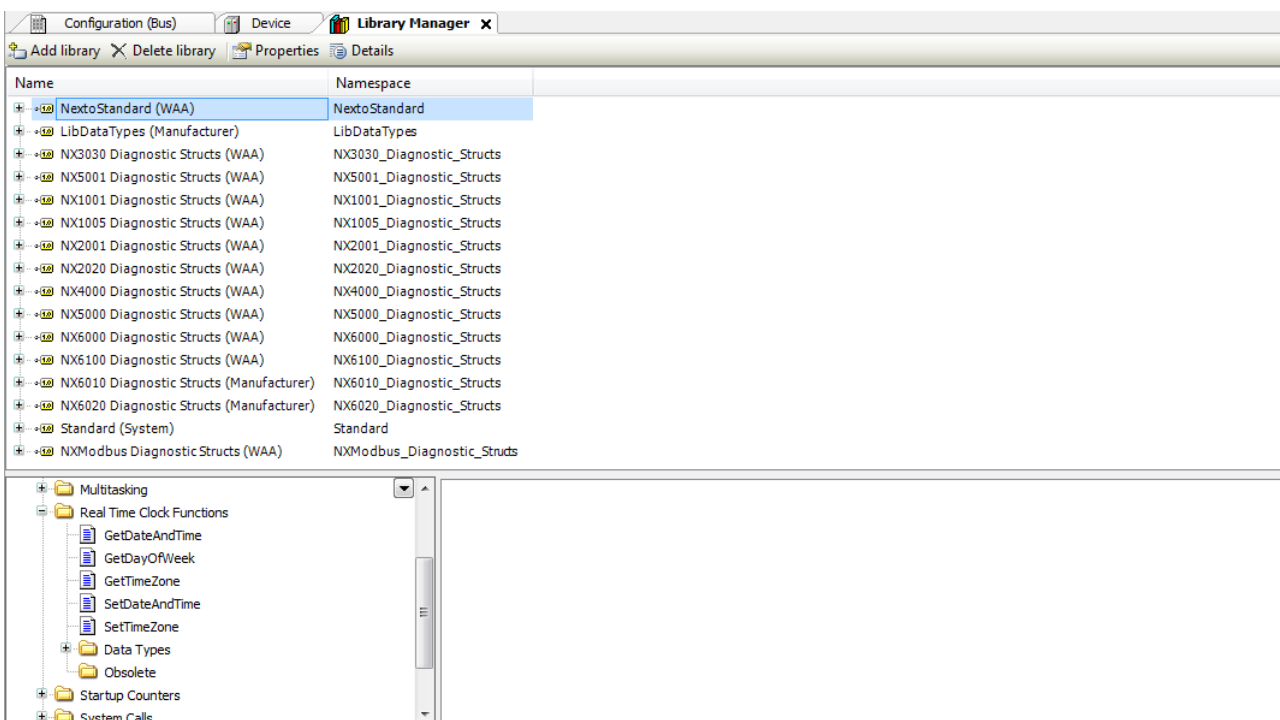


Figura 121: Blocos e Funções de Escrita e Leitura do Relógio

ATENÇÃO

Blocos funcionais para Leitura e Escrita do RTC, disponíveis em versões anteriores a 2.00 do MasterTool IEC XE tornaram-se obsoletos da versão 2.00 em diante, os blocos que ficaram obsoletos são:

NextoGetDateAndTime, *NextoGetDateAndTimeMs*, *NextoGetTimeZone*, *NextoSetDateAndTime*, *NextoSetDateAndTimeMs* e *NextoSetTimeZone*.

+

5.8.1. Blocos Funcionais e Funções para Leitura e Escrita do RTC

Entre outros blocos funcionais e funções, existem alguns muito importantes utilizados para a leitura do relógio (*GetDateAndTime*, *GetDayOfWeek* e *GetTimeZone*) e para configurar novos valores de data e hora (*SetDateAndTime* e *SetTimeZone*). Estas funções utilizam sempre o horário local, isto é, levam em consideração o valor definido para o *Fuso Horário*.

Abaixo, descrevem-se os procedimentos utilizados para configurar as funções.

ATENÇÃO

Os blocos funcionais obsoletos para leitura e escrita do RTC (*NextoGetDateAndTime*, *NextoGetDateAndTimeMs*, *NextoSetDateAndTime* e *NextoSetDateAndTimeMs*) não podem ser utilizados na área de dados redundantes em projetos redundantes. Os mesmos devem ser utilizados em POU's não redundantes, como a POU *NonSkippedPrg*. Mais detalhes sobre o funcionamento da POU *NonSkippedPrg* podem ser encontrados em [Programa NonSkippedPrg](#).

5.8.1.1. Funções de Leitura do RTC

A leitura do relógio pode ser feita através das funções a seguir:

5.8.1.1.1. *GetDateAndTime*

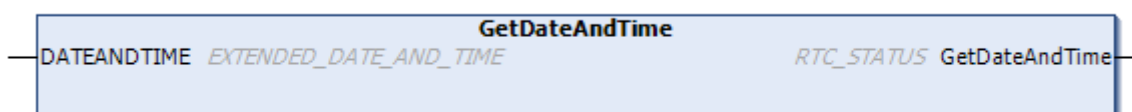


Figura 122: Leitura da Data e Hora

Parâmetros de entrada	Tipo	Descrição
DATEANDTIME	EXTENDED_DATE_AND_TIME	Esta variável retorna o valor de data e hora do RTC no formato apresentado na Tabela 162.

Tabela 153: Parâmetros de Entrada GetDateAndTime

Parâmetros de saída	Tipo	Descrição
GETDATEANDTIME	RTC_STATUS	Retorna o estado de erro da função, ver Tabela 164.

Tabela 154: Parâmetros de Saída GetDateAndTime

5. CONFIGURAÇÃO

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
Result : RTC_STATUS;
DATEANDTIME : EXTENDED_DATE_AND_TIME;
xEnable : BOOL;
END_VAR
-----
IF xEnable = TRUE THEN
Result := GetDateAndTime (DATEANDTIME);
xEnable := FALSE;
END_IF
```

5.8.1.1.2. GetTimeZone

A função a seguir faz a leitura das configurações de fuso horário, esta função está diretamente relacionada com o tempo de fuso horário configurado no serviço de sincronismo do SNTP:

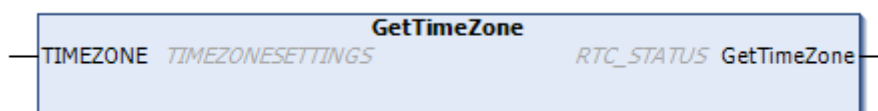


Figura 123: Leitura das Configurações de Fuso Horário

Parâmetros de entrada	Tipo	Descrição
TIMEZONE	TIMEZONESETTINGS	Essa variável apresenta a leitura das configurações de fuso horário.

Tabela 155: Parâmetros de Entrada GetTimeZone

Parâmetros de saída	Tipo	Descrição
GetTimeZone	RTC_STATUS	Retorna o estado de erro da função, ver Tabela 164.

Tabela 156: Parâmetros de Saída GetTimeZone

5. CONFIGURAÇÃO

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
GetTimeZone_Status : RTC_STATUS;
TimeZone           : TIMEZONESETTINGS;
xEnable : BOOL;
END_VAR
-----
IF xEnable = TRUE THEN
GetTimeZone_Status := GetTimeZone(TimeZone);
xEnable := FALSE;
END_IF
```

5.8.1.1.3. *GetDayOfWeek*

A função *GetDayOfWeek* é utilizada para fazer a leitura do dia da semana.

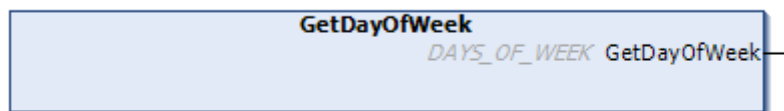


Figura 124: Leitura do Dia da Semana

Parâmetros de saída	Tipo	Descrição
GetDayOfWeek	DAYS_OF_WEEK	Retorna o dia da semana. Ver Tabela 163.

Tabela 157: Parâmetros de Saída GetDayOfWeek

Quando chamada, a função lerá o dia da semana e preencherá a estrutura *DAYS_OF_WEEK*.

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
DayOfWeek : DAYS_OF_WEEK;
END_VAR
-----
DayOfWeek := GetDayOfWeek();
```

5.8.1.2. Funções de Escrita do RTC

As configurações de relógio são feitas através das funções e blocos funcionais a seguir:

5.8.1.2.1. *SetDateAndTime*

O Bloco Funcional *SetDateAndTime* é utilizado para realizar o ajuste do relógio. Tipicamente a precisão é da ordem de centenas de milissegundos.

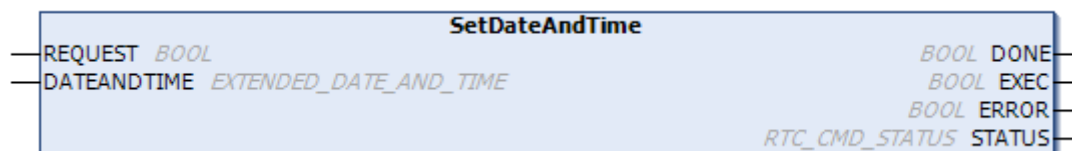


Figura 125: Ajuste de Data e Hora em Milissegundos

Parâmetros de entrada	Tipo	Descrição
REQUEST	BOOL	Essa variável, quando recebe uma borda de subida, habilita a escrita do relógio.
DATEANDTIME	EXTENDED_DATE_AND_TIME	Recebe os valores de data e hora em milissegundos, conforme na Tabela 162.

Tabela 158: Parâmetros de Entrada SetDateAndTime

Parâmetros de saída	Tipo	Descrição
DONE	BOOL	Essa variável, quando verdadeira, indica que a ação foi completada.
EXEC	BOOL	Essa variável, quando verdadeira, indica que a função está processando os valores.
ERROR	BOOL	Essa variável, quando verdadeira, indica que ocorreu algum erro durante a Escrita.
STATUS	RTC_CMD_STATUS	Retorna o erro ocorrido durante a configuração, ver Tabela 164.

Tabela 159: Parâmetros de Saída SetDateAndTime

Quando ocorrer uma borda de subida na entrada *REQUEST*, o bloco funcional irá escrever o novo valor *DATEANDTIME* no relógio. Caso a escrita seja realizada com sucesso, a saída *DONE* será igual a *TRUE*. Caso contrário, a saída *ERROR* será igual a *TRUE* e o erro será apresentado na variável *STATUS*.

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
SetDateAndTime : SetDateAndTime;
xRequest : BOOL;
DateAndTime : EXTENDED_DATE_AND_TIME;
xDone : BOOL;
xExec : BOOL;
xError : BOOL;
xStatus : RTC_STATUS;
```

```

END_VAR
-----
IF xRequest THEN
  SetDateAndTime.REQUEST:=TRUE;
  SetDateAndTime.DATEANDTIME:=DateAndTime;
  xRequest:= FALSE;
END_IF
SetDateAndTime ();
SetDateAndTime.REQUEST:=FALSE;
IF SetDateAndTime.DONE THEN
  xExec:=SetDateAndTime.EXEC;
  xError:=SetDateAndTime.ERROR;
  xStatus:=SetDateAndTime.STATUS;
END_IF
    
```

ATENÇÃO

Se o usuário tentar escrever valores de hora fora do intervalo do RTC, os valores serão convertidos para valores válidos, desde que não ultrapasse a faixa válida de 01/01/2000 até 31/12/2035. Por exemplo, se o usuário tentar escrever o valor 2000 ms, o mesmo será convertido para 2 segundos, se escrever o valor 100 segundos, o mesmo será convertido para 1 min e 40 segundos. Se escrever o valor de 30 horas, o mesmo será convertido para 1 dia e 6 horas, e assim por diante.

5.8.1.2.2. SetTimeZone

A função *SetTimeZone* realiza a escrita do ajuste de fuso horário:

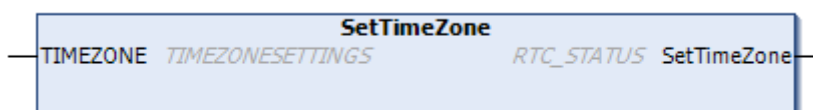


Figura 126: Ajuste de Fuso Horário

Parâmetros de entrada	Tipo	Descrição
TIMEZONE	TIMEZONESETTINGS	Estrutura com o valor de fuso horário a ser configurado. Ver Tabela 165.

Tabela 160: Parâmetros de Entrada SetTimeZone

Parâmetros de saída	Tipo	Descrição
SetTimeZone	RTC_STATUS	Retorna o erro ocorrido durante a leitura/configuração. Ver Tabela 164.

Tabela 161: Parâmetros de Saída SetTimeZone

Quando chamada, a função, irá configurar o valor de *TIMEZONE* como a nova configuração de fuso horário do sistema. O resultado da configuração é retornado pela função.

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
Status : RTC_STATUS;
TimeZone : TIMEZONESETTINGS;
xWrite : BOOL;
END_VAR
-----
//FB SetTimeZone
IF (xWrite = TRUE) THEN
Status := SetTimeZone(TimeZone);
  IF Status = RTC_STATUS.NO_ERROR THEN
    xWrite := FALSE;
  END_IF
END_IF
```

ATENÇÃO

Para realizar o acerto do relógio, devem-se utilizar valores de hora e datas dentro da seguinte faixa válida: 00:00:00 horas de 01/01/2000 até 23:59:59 horas de 31/12/2035, caso contrário, será reportado um erro através do parâmetro de saída *STATUS*. Para maiores detalhes do parâmetro de saída *STATUS*, consultar a seção [RTC_STATUS](#).

5.8.2. Estruturas de Dados do RTC

Os blocos funcionais de leitura e configuração do RTC das UCPs da série Nexto utilizam as seguintes estruturas de dados na sua configuração:

5.8.2.1. EXTENDED_DATE_AND_TIME

Esta estrutura é utilizada para armazenar a data do RTC quando utilizados os blocos funcionais para leitura/configuração da data com precisão de milissegundos e é descrita na tabela abaixo:

Estrutura	Tipo	Variável	Descrição
EXTENDED_DATE_AND_TIME	BYTE	byDayOfMonth	Armazena o dia do mês da data configurada.
	BYTE	ByMonth	Armazena o mês da data configurada.
	WORD	wYear	Armazena o ano da data configurada.
	BYTE	byHours	Armazena a hora da data configurada.
	BYTE	byMinutes	Armazena os minutos da data configurada.
	BYTE	bySeconds	Armazena os segundos da data configurada.
	WORD	wMilliseconds	Armazena os milissegundos da data configurada.

Tabela 162: Estrutura EXTENDED_DATE_AND_TIME

5.8.2.2. DAYS_OF_WEEK

Esta estrutura é utilizada para armazenar o dia da semana quando utilizada a função para leitura do dia da semana:

Enumerador	Valor	Descrição
DAYS_OF_WEEK	0	INVALID_DAY
	1	SUNDAY
	2	MONDAY
	3	TUESDAY
	4	WEDNESDAY
	5	THURSDAY
	6	FRIDAY
	7	SATURDAY

Tabela 163: Estrutura DAYS_OF_WEEK

5.8.2.3. RTC_STATUS

Este enumerador é utilizado para retornar o tipo de erro na configuração ou leitura do RTC e é descrito na tabela abaixo:

Enumerador	Valor	Descrição
RTC_STATUS	NO_ERROR (0)	Não há erro.
	UNKNOWN_COMMAND (1)	Comando desconhecido.
	DEVICE_BUSY (2)	Dispositivo está ocupado.
	DEVICE_ERROR (3)	Dispositivo com erro.
	ERROR_READING_OSF (4)	Erro na leitura do sinalizador de data e hora válidas.
	ERROR_READING_RTC (5)	Erro na leitura da data e hora.
	ERROR_WRITING_RTC (6)	Erro na escrita da data e hora.
	ERROR_UPDATING_SYSTEM_TIME (7)	Erro na atualização de data e hora do sistema.
	INTERNAL_ERROR (8)	Erro interno.
	INVALID_TIME (9)	Data e hora inválidas.
	INPUT_OUT_OF_RANGE (10)	Fora do limite de Data e hora válidas para o sistema.
	SNTP_NOT_ENABLE (11)	Erro gerado quando o serviço SNTP não está habilitado e é feita uma tentativa de ler ou modificar o fuso horário.

Tabela 164: RTC_STATUS

5.8.2.4. TIMEZONESETTINGS

Esta estrutura é utilizada para armazenar o valor do fuso horário nas requisições de leitura/configuração dos blocos funcionais do RTC e é descrita na tabela abaixo:

Estrutura	Tipo	Variável	Descrição
TIMEZONESETTINGS	INT	iHour	Hora do fuso horário configurado.
	INT	iMinutes	Minuto do fuso horário configurado.

Tabela 165: TIMEZONESETTINGS

Nota:

Blocos funcionais de escrita e leitura de data e hora: Bibliotecas diferentes da *NextoStandard*, que tenham blocos funcionais ou funções que possam fazer acesso de leitura e escrita da data e hora no sistema, não são indicadas. A biblioteca *NextoStandard* possui as interfaces adequadas para escrever e ler a data e hora do sistema adequadamente e informar os diagnósticos corretos.

5.9. Memória de Arquivos de Usuário

As UCPs da Série Nexto possuem uma área de memória destinada ao armazenamento de dados de uso geral, ou seja, o usuário poderá gravar diversos arquivos de qualquer formato na memória da UCP. Esta área de memória pode variar de acordo com o modelo de UCP utilizado (consultar [Memória](#)).

Para usar esta área, o usuário deverá acessar um projeto no software MasterTool IEC XE e clicar na *Árvore de Dispositivos*, localizada à esquerda do programa. Deverá dar dois cliques sobre o item *Device* e, após selecionar a UCP na aba *Configurações de Comunicação* que será aberta, selecionar a aba *Arquivos* e clicar em *Atualizar*, tanto na coluna de arquivos do computador (esquerda), como na coluna de arquivos da UCP selecionada (direita), conforme mostram as indicações da figura abaixo.

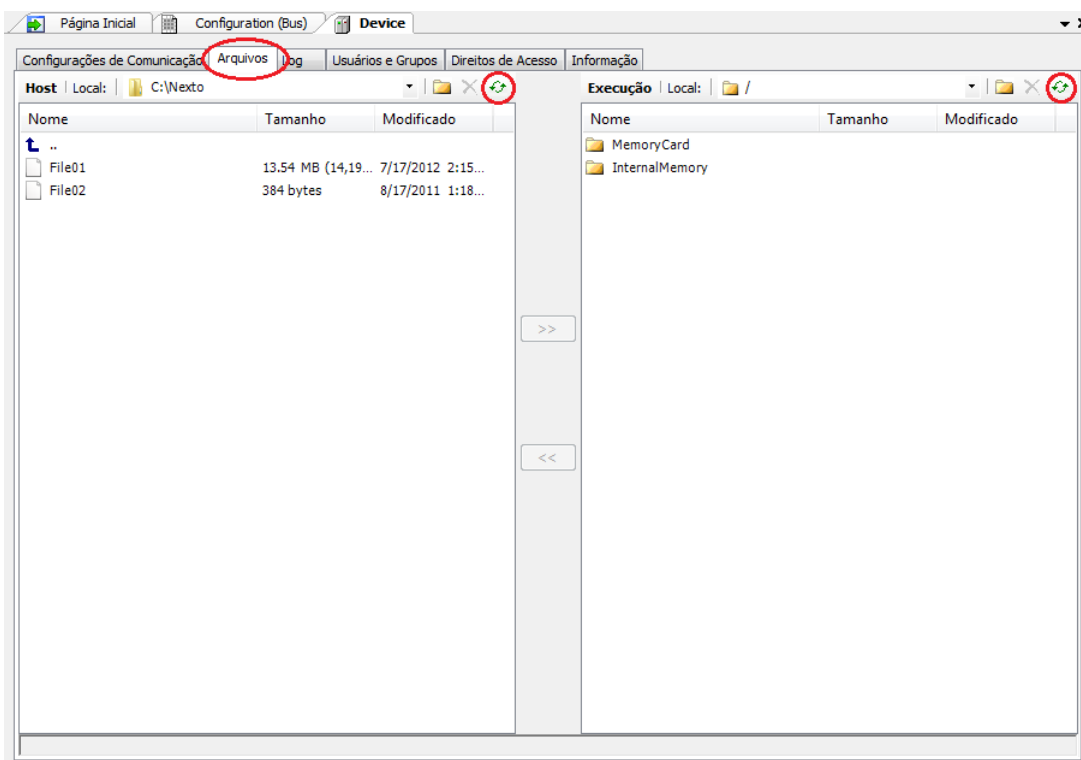


Figura 127: Acesso aos Arquivos de Usuário

Após atualizar a coluna de arquivos da UCP, será exibido o diretório raiz de arquivos armazenados na UCP e poderá ser selecionada a pasta para onde os arquivos serão transferidos. A pasta “*InternalMemory*” é uma pasta padrão, a ser utilizada para armazenar arquivos na memória interna da UCP, uma vez que não é possível transferir arquivos para o diretório raiz. Caso seja necessário, podem ser criadas outras pastas no diretório raiz ou subpastas dentro da pasta “*InternalMemory*”.

Já a pasta “*MemoryCard*” é o diretório onde o cartão de memória estará montado, caso o mesmo esteja inserido na UCP. Arquivos transferidos para a pasta “*MemoryCard*” estarão sendo transferidos diretamente para dentro do cartão de memória. Conforme novas funcionalidades forem sendo adicionadas ao produto, algumas pastas podem aparecer e que devem ser ignoradas pelo usuário.

ATENÇÃO

No caso em que o cartão de memória seja inserido após a inicialização da UCP, será requisitado um usuário e senha para realizar o acesso e/ou transferências de arquivos do MasterTool IEC XE para o cartão de memória ou vice-versa. O usuário padrão com privilégios para acesso à UCP é “*Owner*” e a senha padrão desse usuário é “*Owner*”.

Para realizar a transferência de algum arquivo do microcomputador para a UCP, basta selecionar o arquivo desejado na coluna da esquerda e pressionar o botão “>>”, localizado no centro da tela, conforme figura abaixo. O tempo de transferência irá variar de acordo com o tamanho do arquivo e com o tempo de ciclo (execução) da aplicação atual da UCP, podendo levar vários minutos.

O usuário não precisa estar em Modo *Run* ou conectado à UCP para realizar as transferências, pois ela possui a capacidade de se conectar automaticamente quando o usuário realizar a transferência.

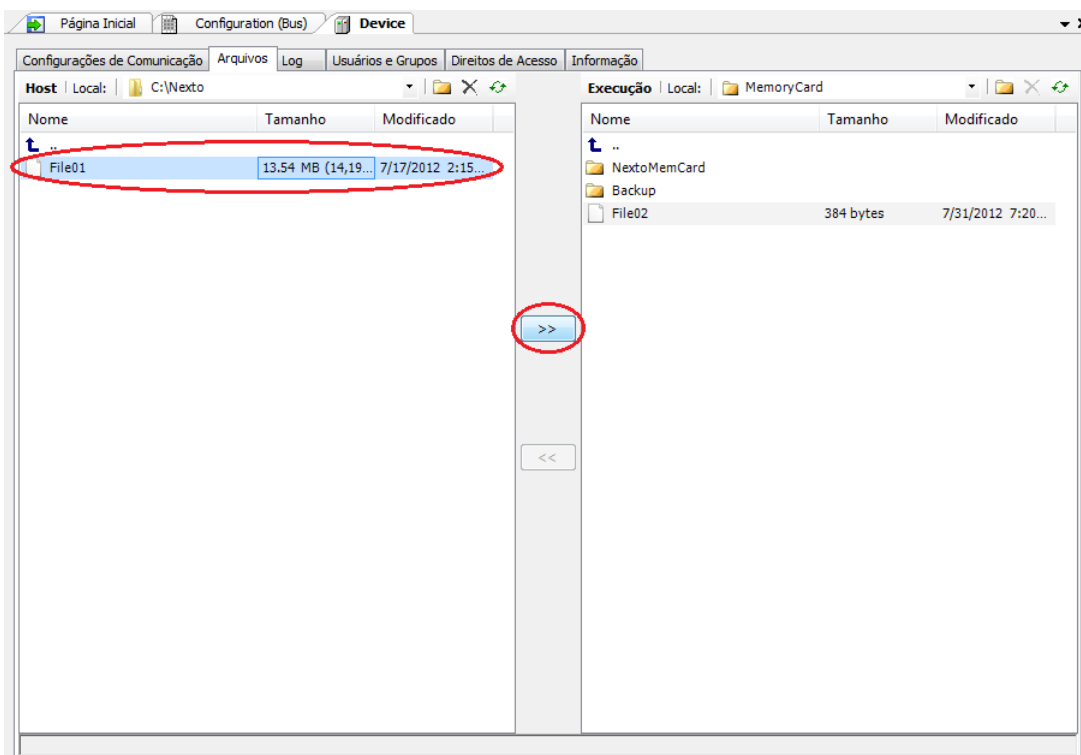





Figura 128: Transferindo Arquivos

ATENÇÃO

Os arquivos contidos dentro da pasta de um projeto criado pela ferramenta MasterTool IEC XE possuem nomes especiais reservados pelo sistema, desta forma não podem ser transferidos através da aba *Arquivos*. Caso o usuário deseje transferir um projeto para a memória de usuário, será necessário compactar a pasta e então transferir o arquivo compactado (*.zip por exemplo).

Caso seja necessário transferir documentos da UCP para o microcomputador em que está instalado o software MasterTool IEC XE o usuário deve realizar um procedimento muito semelhante ao anterior, ou seja, selecionando o arquivo na coluna da direita e pressionar o botão “<”, localizado no centro da tela.

Além disso, o usuário possui algumas opções de operação da área de armazenamento de arquivos, são elas:

- Novo diretório : permite a criação de uma nova pasta na área de memória de usuário.
- Excluir item : permite a exclusão de arquivos nos diretórios da área de memória de usuário.
- Atualizar : permite atualizar, na tela do MasterTool IEC XE, os arquivos presentes na memória de usuário e no microcomputador.

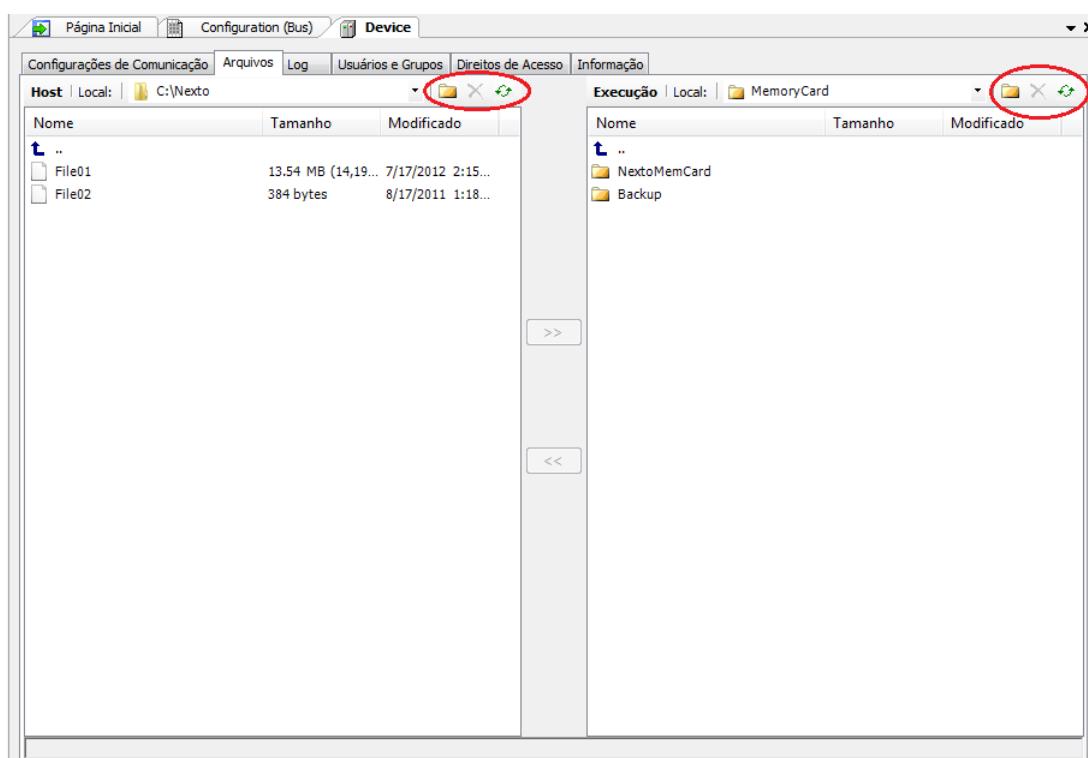


Figura 129: Opções de Utilização

ATENÇÃO

Para uma UCP em Modo Stop ou sem nenhuma aplicação, a taxa de transferência para a memória interna é de aproximadamente 150 Kbytes/s.

5.10. Cartão de Memória

Entre outras funcionalidades, este modelo de UCP da Série Nexto possibilita ao usuário a utilização de um cartão de memória, definido conforme as características da seção [Interface do Cartão de Memória](#), o qual serve para armazenar, entre outros arquivos, o projeto e a aplicação que está na memória interna da UCP.

Quando o cartão for inserido na UCP e estiver com sistema de arquivos diferente de FAT32, ela automaticamente identifica e pergunta ao usuário se ele deseja formatar. Em caso negativo, ele não poderá utilizar o cartão (o cartão não será montado, aparecerá uma mensagem dizendo que o formato não foi reconhecido, e o visor não indicará a presença do cartão). Caso seja selecionada a opção de formatação, a UCP irá levar alguns minutos, dependendo do tempo de ciclo (execução) da aplicação rodando na UCP, para executar a operação. Assim que o cartão de memória for montado, a UCP irá ler informações gerais do mesmo, deixando o acesso ao cartão de memória mais lento nos primeiros minutos. Esse procedimento é feito apenas quando o cartão é inserido ou a UCP reinicializada.

ATENÇÃO

Recomenda-se realizar o procedimento de formatação do cartão de memória diretamente na UCP Nexto para evitar possíveis problemas de utilização, aumento do tempo de montagem ou até mesmo funcionamento incorreto.

Não é recomendado remover o cartão de memória ou desenergizar a UCP durante a formatação ou durante a transferência de arquivos, pois pode causar a perda de dados bem como danos irreversíveis ao cartão.

5.10.1. Preparação do Projeto

Para utilização da funcionalidade, durante a configuração do projeto, dentro do software MasterTool IEC XE, o usuário deve habilitar a opção de cópia do projeto da UCP para o cartão de memória e cartão de memória para a UCP e configurar senhas de acesso. Estas senhas serão solicitadas pela UCP no momento da execução da respectiva transferência. Informações sobre a tabela, consultar a seção [Parâmetros do Projeto](#).

ATENÇÃO

Caso a UCP não tenha aplicação carregada, o Menu “*Cartão de Memória*” estará disponível para permitir a realização da transferência do projeto do cartão para a UCP sem que seja necessário realizar qualquer tipo de preparação prévia da UCP.

Para utilização da funcionalidade é necessário executar os passos a seguir.

Deve-se navegar até o menu *Comunicação* e executar o comando *Criar Aplicação de Inicialização*, lembrando que não se pode estar logado na UCP para realizar esse procedimento. Após executar esse comando, dois arquivos são criados na pasta onde o projeto está salvo. Um com a extensão “*app*” e outro com a extensão “*crc*”.

Após gerados os arquivos no item anterior deve-se navegar até a configuração dos *Parâmetros Gerais* da UCP e clicar no botão *Cartão de Memória*. Uma nova tela será aberta conforme a figura abaixo. Nesta tela deve-se habilitar a(s) operação(ões) de transferência desejada(s) e, caso necessário, configurar a(s) senha(s), apenas com caracteres numéricos. A utilização de senha não é obrigatória.

Para concluir a operação de configuração é necessário clicar no botão *Localizar Arquivo...* e então localizar o arquivo com extensão “*crc*” gerado no passo anterior.

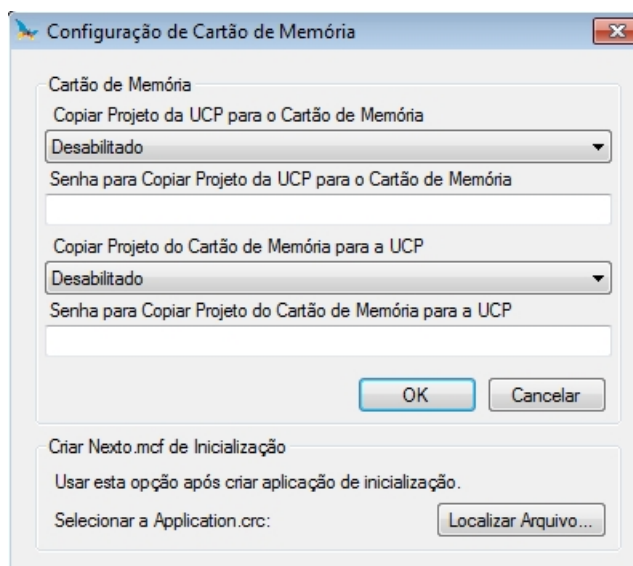


Figura 130: Configurações do Cartão de Memória

Executados esses passos, o MasterTool IEC XE irá enviar todos os arquivos necessários para realizar as operações de envio e recebimento de projetos via cartão de memória. Caso o cartão esteja montado, a senha será gravada no mesmo. Caso contrário, a senha configurada no MasterTool será solicitada se o usuário tentar transferir o projeto da UCP para o cartão.

5.10.2. Envio do Projeto

Para realizar o envio do projeto da UCP para o cartão de memória ou vice-versa, o usuário, além de habilitar no software MasterTool IEC XE o uso da funcionalidade , terá que acessar o Menu *Cartão de Memória* na UCP, utilizando a tecla de diagnósticos, e selecionar a opção de transferência desejada.

ATENÇÃO

O envio do projeto para o cartão de memória, deve ser realizado apenas usando a tecla de diagnósticos da UCP.

Após, será solicitada a senha, caso o usuário tenha configurado durante a configuração da aplicação. Então, com um pressionamento curto na tecla de diagnósticos, os dígitos são incrementados e com um pressionamento longo são confirmados. No sexto dígito confirmado, a UCP irá consistir a senha e iniciará o processo.

Após a transferência do cartão de memória para a UCP, caso haja uma aplicação em RUN ela será mantida em STOP por motivos de segurança. Para colocar a UCP em RUN, ela deve ser reinicializada.

Quando as senhas, tanto da aplicação que está na UCP quanto da aplicação que está no cartão de memória, forem iguais, não é requisitado a inserção das senhas no menu da UCP para realizar as transferências das aplicações. Maiores informações sobre a utilização da tecla de diagnósticos, consultar a seção [One Touch Diag](#).

Para remover o cartão de memória, basta pressionar a tecla *MS* com um pressionamento longo, e aguardar até que o ícone do cartão desapareça da tela de status do visor gráfico.

ATENÇÃO

Caso o cartão de memória seja removido sem ser desmontado via menu da UCP, durante a transferência de arquivos, este processo pode acarretar na perda de dados no cartão, bem como corromper os arquivos nele contidos. Este processo pode implicar na necessidade de uma nova formatação do cartão quando inserido novamente à UCP.

ATENÇÃO

Caso exista algum arquivo na raiz do cartão de memória nomeado “*NextoMemCard*” ou “*Backup*”, o mesmo será excluído para a criação das pastas de mesmo nome, utilizadas pela UCP para armazenamento do projeto de aplicação e do *Project archive*. Pastas com estes nomes não serão sobrescritas.

5.10.3. Acesso no MasterTool

O acesso à memória do cartão está vinculado à mesma tela de memória de usuário no software MasterTool IEC XE, sendo ele montado na pasta *MemoryCard*. Os diretórios *NextoMemCard* e *Backup* são criados dentro do cartão de memória toda a vez que ele for inserido na UCP. Caso eles já existam, o sistema irá reconhecer e não sobrescreverá as pastas.

Na pasta *NextoMemCard* no cartão de memória, encontram-se os arquivos referentes à aplicação, nesta janela você ainda tem a opção de salvar em um diretório de preferência o seu projeto(caso tenha enviado o código fonte). No MasterTool, na opção “*Arquivo/ Arquivo de Projeto/ Extrair Arquivo...*”pode-se abrir no MasterTool a aplicação salva, que está localizada no diretório escolhido anteriormente.

A pasta *Backup* não é utilizada pelo usuário.

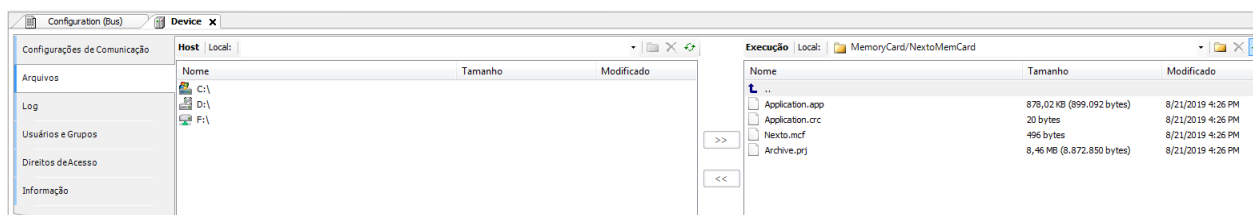


Figura 131: Diretório com Cartão de Memória Inserido com Projeto

ATENÇÃO

O tempo de transferência de arquivos depende da diferença do tempo de intervalo menos o tempo de execução médio da(s) tarefa(s) em execução (tempo disponível até o próximo ciclo da tarefa), isto é, quanto maior for essa diferença para cada tarefa em uma aplicação, mais rápida deverá ser a transferência de um dado a partir do cartão de memória para a UCP/MasterTool IEC XE ou vice-versa.

A transferência de arquivos para o cartão de memória será mais lenta que a transferência para a memória interna da UCP. Para uma UCP em Modo Stop ou sem nenhuma aplicação, a taxa de transferência se aproxima de 100 Kbytes/s.

5.11. Menu Informativo e de Configuração da UCP

O acesso ao *Menu Informativo* e de *Configuração* das UCPs Nexto através do *Visor Gráfico*, assim como o acesso detalhado aos diagnósticos, estão disponíveis através de níveis, sendo que para acessar as informações do menu, trocar de nível e modificar alguma configuração, basta dar um pressionamento longo na tecla de diagnóstico e, para navegar pelos itens de mesmo nível, basta dar um pressionamento curto na tecla de diagnóstico. Consultar a seção [One Touch Diag](#) para verificar o funcionamento e a diferença entre tipos de pressionamento na tecla de diagnósticos.

A tabela abaixo mostra os níveis do menu e o tipo de cada tela disponível nas UCPs, ou seja, se ela é de caráter informativo, configurável ou se retorna um nível.

Nível 1	Nível 2	Nível 3	Tipo
HARDWARE	TEMPERATURA	-	Informativo
	CONTRASTE	NIVEL CONTRASTE	Configurável
	DATA E HORA	-	Informativo
	VOLTAR	-	Retorna Nível
IDIOMAS	ENGLISH	>ENGLISH	Configurável
	PORTUGUES	>PORTUGUES	Configurável
	ESPAÑOL	>ESPAÑOL	Configurável
	VOLTAR	-	Retorna Nível
REDE	END. IP NET 1		Informativo
	MASCARA NET 1		Informativo
	END. IP NET 2	-	Informativo
	MASCARA NET 2		Informativo
	VOLTAR		Retorna Nível
REDUNDANCIA	IDENT. CP		Informativo
	ESTADO REM.	-	Informativo
	SINCR. PROJ.		Informativo
	VOLTAR		Retorna Nível
SOFTWARE	FIRMWARE		Informativo
	BOOTLOADER	-	Informativo
	PROC. AUX.		Informativo
	VOLTAR		Retorna Nível
CARTAO DE MEM.	CARTAO > UCP	SENHA UCP	Configurável
	UCP > CARTAO	SENHA CM	Configurável
	FORMATAR	CONFIRMA ?	Configurável
	VOLTAR	-	Retorna Nível
VOLTAR	-	-	Retorna Nível

Tabela 166: Níveis do Menu da UCP

Notas:

Cartão de Memória: O cartão de memória somente estará disponível no menu caso o mesmo esteja inserido na UCP Nexto.

Redundância: O menu “*REDUNDANCIA*” somente estará disponível caso a UCP NX3030 esteja identificada como Redundante.

Senha: A senha para acesso aos dados do cartão de memória somente será necessária caso ela seja configurada no software MasterTool IEC XE. Não é possível editar a senha via menu.

Conforme já mostrou a Tabela 166, entre as opções disponíveis para visualização e alteração, encontram-se os principais dados necessários ao usuário, como:

- Informações sobre os recursos de hardware:
 - TEMPERATURA – Temperatura interna da UCP (Ex.: 36 C 97 F)
 - CONTRASTE – Ajuste do contraste do visor frontal da UCP
 - DATA E HORA – Data e hora configuradas na UCP (Ex.: 2001.01.31 00:00)
- Alteração do idioma do menu da UCP:
 - PORTUGUES – Altera o idioma para Português
 - ENGLISH – Altera o idioma para Inglês
 - ESPANOL – Altera o idioma para Espanhol
- Visualização de informações sobre a rede configurada no dispositivo:
 - END. IP NET 1 – Endereço IP (Ex.: 192.168.0.1)
 - MASCARA NET 1 – Máscara de subrede (Ex.: 255.255.255.0)
 - END. IP NET 2 – Endereço IP (Ex.: 192.168.0.2)
 - MASCARA NET 2 – Máscara de subrede (Ex.: 255.255.255.0)
- Acesso às informações de redundância do CP:
 - IDENT. CP – Informa a identificação do CP na redundância. Possíveis informações:
 - CPA
 - CPB
 - ESTADO REM. – Informa o estado do CP redundante remoto. Possíveis estados:
 - ATIVO
 - RESERVA
 - INATIVO
 - NAO CONFIG.
 - INICIANDO
 - INDISPON.
 - SINCR. PROJ. – Informa se a sincronização de projetos está habilitada
 - CONECTADO
 - NÃO CONEC.
 - DESABILITADO
 - SINC. INI.
 - SINCRONIZADO
- Informações sobre as versões de software:
 - FIRMWARE – Versão de software da UCP (Ex.: 1.0.0.0)
 - BOOTLOADER – Versão do bootloader da UCP (Ex.: 1.0.0.0)
 - PROC. AUX. – Versão do processador auxiliar da UCP (Ex.: 1.0.0.0)
- Acesso aos dados do Cartão de Memória:
 - CARTAO>UCP – Transferência do projeto do cartão de memória para a UCP
 - UCP>CARTAO – Transferência do projeto da UCP para o cartão de memória
 - FORMATAR – Formata o cartão para o sistema de arquivos FAT32

A figura abaixo descreve um exemplo de como operar o menu das UCPs Nexto, através do procedimento de ajuste do contraste a partir da tela de *Status*. Além de facilitar a configuração, é possível identificar todos os níveis de tela e o tipo de pressionamento para navegar entre as mesmas, sendo que para modificar os outros parâmetros, como *Idioma* e inserir a(s) senha(s) no *Cartão de Memória*, basta seguir a mesma lógica de acesso. O pressionamento curto mostra que o contraste está sendo incrementado (mais claro), sendo que no próximo pressionamento após o seu valor máximo, ele retorna ao valor mínimo (menos claro). O pressionamento longo mostra a confirmação do contraste desejado e o retorno ao nível anterior.

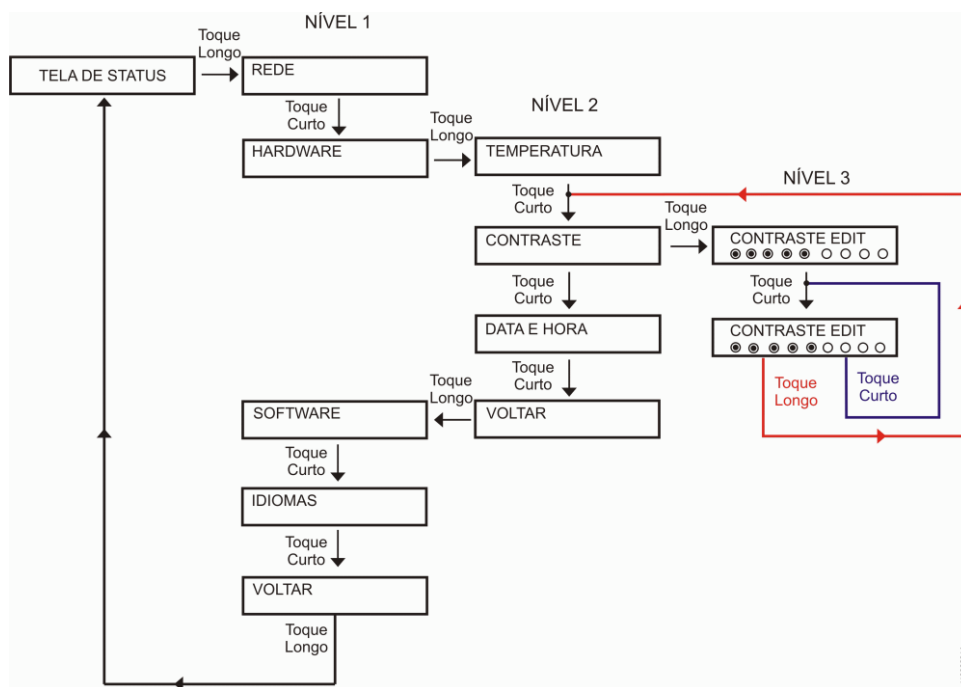


Figura 132: Ajuste do Contraste

Além do menu das UCPs Nexto ser encerrado através de um pressionamento longo na tecla de diagnósticos na tela *VOLTAR* do nível 1, também existem outras condições de saída, as quais estão descritas abaixo:

- Toque curto, em qualquer momento, em um outro módulo existente no barramento, faz com que o menu passe a descrever o diagnóstico dele.
- Tempo de inatividade, em qualquer nível, superior a 5 s.

5.12. Blocos Funcionais e Funções

5.12.1. Blocos Funcionais Especiais para Comunicação Serial

Os blocos funcionais especiais para comunicação serial possibilitam o acesso local (COM) e também a portas seriais remotas (módulos de expansão). Dessa forma, o usuário poderá criar seus próprios protocolos e manipular as portas seriais como quiser, seguindo as linguagens da IEC 61131-3 disponíveis no software MasterTool IEC XE. Os blocos estão disponíveis dentro da biblioteca *NextoSerial*, a qual deve ser adicionada ao projeto para que seja possível utilizá-los (para realizar o procedimento de inserção de uma biblioteca, consultar o Manual de Programação IEC 61131 – MP399048, capítulo Bibliotecas).

Os blocos funcionais especiais para comunicação serial podem levar vários ciclos (consecutivas chamadas) para completar a execução da tarefa. Às vezes, um bloco pode completar em um único ciclo, mas, em geral, necessita de vários ciclos. A execução da tarefa associada a um bloco pode compreender vários passos, sendo que alguns dependem de eventos externos, os quais podem ter atrasos significantes para o sistema. O bloco funcional não pode implementar rotinas para ocupar o tempo, enquanto aguarda por esses eventos, pois assim iria utilizar todo recurso da UCP. A solução poderia ser a criação de blocos funcionais bloqueadores, mas isso não é aconselhável pois iria complicar a aplicação do usuário, pois normalmente não se tem disponível a programação multitarefa. Então, quando um evento externo é esperado, os blocos funcionais da serial são finalizados e o controle é retornado para o programa de chamada. O tratamento da tarefa continua no próximo ciclo, ou seja, na próxima vez que o bloco for chamado.

Antes de descrever os blocos funcionais especiais para controle das interfaces seriais, é importante conhecer os *Datatypes*, ou seja, os tipos de dados utilizados pelos blocos:

Tipo de dado	Opção	Descrição
SERIAL_BAUDRATE	BAUD200	Lista todas as possibilidades da taxa de transmissão (bits por segundo)
	BAUD300	
	BAUD600	
	BAUD1200	
	BAUD1800	
	BAUD2400	
	BAUD4800	
	BAUD9600	
	BAUD19200	
	BAUD38400	
	BAUD57600	
	BAUD115200	
	SERIAL_DATABITS	
DATABITS_6		
DATABITS_7		
DATABITS_8		
SERIAL_HANDSHAKE	Define todas as possibilidades dos sinais de modem para as configurações:	
	RS232_RTS	Controla a porta RS-232C da UCP Nexto. O RTS é habilitado no início da transmissão e reiniciado assim que possível após o final da transmissão. Por exemplo, pode ser utilizado para controlar um conversor RS-232/RS-485 externo.
	RS232_RTS_OFF	Controla a porta RS-232C da UCP Nexto. O sinal RTS está sempre desligado.
	RS232_RTS_ON	Controla a porta RS-232C da UCP Nexto. O sinal RTS está sempre ligado.
	RS232_RTS_CTS	Controla a porta RS-232C da UCP Nexto. Caso o CTS esteja desabilitado, o RTS é habilitado. Então, aguarda-se o CTS ser habilitado para a transmissão começar e o RTS é reiniciado, o mais rápido possível, no final da transmissão. Ex: Controle de rádio modems com o mesmo sinal de modem.
	RS232_MANUAL	Controla a porta RS-232C da UCP Nexto. O usuário é responsável por controlar todos os sinais (RTS, DTR, CTS, DSR, DCD).
SERIAL_MODE	NORMAL_MODE	Modo normal de operação da comunicação serial.
	EXTENDED_MODE	Modo estendido de operação da comunicação serial, no qual são fornecidas informações sobre o frame de dados recebido.

Tipo de dado	Opção	Descrição
SERIAL_PARAMETERS	Define todos os parâmetros de configuração da porta serial:	
	BAUDRATE	Definido em SERIAL_BAUDRATE.
	DATABITS	Definido em SERIAL_DATABITS.
	STOPBITS	Definido em SERIAL_STOPBITS.
	PARITY	Definido em SERIAL_PARITY.
	HANDSHAKE	Definido em SERIAL_HANDSHAKE.
	UART_RX_THRESHOLD	Quantidade de bytes que devem ser recebidos para gerar uma nova interrupção na UART. Valores baixos fazem o TIMESTAMP mais preciso quando o MODO ESTENDIDO é utilizado e minimiza os erros de overrun. No entanto, valores baixos podem causar muitas interrupções que podem retardar a UCP.
	MODE	Definido em SERIAL_MODE.
	ENABLE_RX_ON_TX	Quando verdadeiro, todos os bytes recebidos durante a transmissão serão descartados em vez de ir para a fila de RX. Utilizado para desabilitar a operação full-duplex na interface RS-422.
	ENABLE_DCD_EVENT	Quando verdadeiro, gera um evento externo quando o DCD é alterado.
ENABLE_CTS_EVENT	Quando verdadeiro, gera um evento externo quando o CTS é alterado.	
SERIAL_PARITY	PARITY_NONE	Lista todas as possibilidades de paridade.
	PARITY_ODD	
	PARITY_EVEN	
	PARITY_MARK	
	PARITY_SPACE	
SERIAL_PORT	COM 1	Lista todas as portas seriais disponíveis (COM 10, COM 11, COM 12, COM 13, COM 14, COM 15, COM 16, COM 17, COM 18, COM 19 – módulos de expansão).
	COM 2	
SERIAL_RX_CHAR_EXTENDED	Define um caractere da fila RX modo estendido.	
	RX_CHAR	Byte de dados.
	RX_ERROR	Código de erro.
	RX_TIMESTAMP	Silêncio devido ao caractere anterior ou devido a outro evento que aconteceu antes desse caractere (configuração da porta serial, final da transmissão).
Contém alguns campos que disponibilizam informações de status/erro sobre a fila RX, utilizados quando é utilizado o formato normal (sem erro e informações de estampa de tempo):		

Tipo de dado	Opção	Descrição
SERIAL_RX_QUEUE_STATUS	RX_FRAMING_ERRORS	Contador de erros de frame, ou seja, formação incorreta do caractere – falta de bit de parada, taxa de transmissão incorreta, entre outros – desde a configuração da porta serial. Retorna para zero caso atinja o valor máximo (65535).
	RX_PARITY_ERRORS	Contador de erros de paridade desde a configuração da porta serial. Retorna para zero caso atinja o valor máximo (65535).
	RX_BREAK_ERRORS	Contador de erros de interrupção desde a configuração da porta serial, ou seja, linha ativa maior do que o tempo de um caractere. Retorna para zero caso atinja o valor máximo (65535).
	RX_FIFO_OVERRUN_ERRORS	Contador de erros de overrun na FIFO RX desde a configuração da porta serial, ou seja, erro no threshold configurado para a FIFO RX. Retorna para zero caso atinja o valor máximo (65535).
	RX_QUEUE_OVERRUN_ERRORS	Contador de erros de overrun na fila RX desde a configuração da porta serial, ou seja, o valor máximo de caracteres (1024) foi ultrapassado e os dados estão sendo sobrescritos. Retorna zero caso atinja o valor máximo (65535).
	RX_ANY_ERRORS	Soma dos 5 últimos contadores de erros (frame, paridade, interrupção, overrun RX FIFO, overrun fila RX).
	RX_REMAINING	Número de caracteres remanescentes na fila RX.
	Lista os códigos de erro críticos que podem ser retornados pelos blocos funcionais da serial. Cada bloco retorna erros específicos, os quais serão mencionados na descrição dos mesmos:	
	NO_ERROR	Não existem erros.
	ILLEGAL_*	Retorna os parâmetros com valores inválidos ou fora da faixa: - SERIAL_PORT - SERIAL_MODE - BAUDRATE - DATA_BITS - PARITY - STOP_BITS - HANDSHAKE - UART_RX_THRESHOLD - TIMEOUT - TX_BUFF_LENGTH - HANDSHAKE_METHOD - RX_BUFF_LENGTH

Tipo de dado	Opção	Descrição
SERIAL_STATUS	PORT_BUSY	Indica que a porta serial está sendo utilizada por outra instância
	HW_ERROR_UART	Erro de hardware detectado na UART.
	HW_ERROR_REMOTE	Erro de hardware ao comunicar com a porta serial remota.
	CTS_TIMEOUT_ON	Time-out na espera do CTS ser habilitado, nos sinais de modem RS-232C RTS/CTS, no bloco SERIAL_TX.
	CTS_TIMEOUT_OFF	Time-out na espera do CTS ser desabilitado, no handshake RS-232C RTS/CTS, no bloco SERIAL_TX.
	TX_TIMEOUT_ERROR	Time-out na espera pelo final da transmissão no bloco SERIAL_TX.
	RX_TIMEOUT_ERROR	Time-out na espera de todos os caracteres no bloco SERIAL_RX ou SERIAL_RX_EXTENDED.
	FB_SET_CTRL_NOT_ALLOWED	O bloco SET_CTRL não pode ser utilizado caso o sinal de modem seja diferente de RS232_MANUAL.
	FB_GET_CTRL_NOT_ALLOWED	O bloco GET_CTRL não pode ser utilizado caso o sinal de modem seja diferente de RS232_MANUAL.
	FB_SERIAL_RX_NOT_ALLOWED	O bloco SERIAL_RX não está disponível para a fila RX, modo estendido.
	FB_SERIAL_RX_EXTENDED_NOT_ALLOWED	O bloco SERIAL_RX_EXTENDED não está disponível para a fila RX, modo normal.
	DCD_INTERRUPT_NOT_ALLOWED	A interrupção pelo sinal DCD não pode ser habilitada caso a porta serial não possua o respectivo pino.
	CTS_INTERRUPT_NOT_ALLOWED	A interrupção pelo sinal CTS não pode ser habilitada caso o sinal de modem seja diferente de RS232_MANUAL ou caso a porta serial não possua o respectivo pino.
	DSR_INTERRUPT_NOT_ALLOWED	A interrupção pelo sinal DSR não pode ser habilitada caso a porta serial não possua o respectivo pino. (As UCPs Nexto não possuem esse sinal nas portas locais)
	NOT_CONFIGURED	O bloco funcional não pode ser utilizado antes da porta serial ser configurada.
INTERNAL_ERROR	Indica que algum problema interno ocorreu na porta serial.	
	STOPBITS_1	Lista todas as possibilidades de bits de parada.

Tipo de dado	Opção	Descrição
SERIAL_STOPBITS	STOPBITS_2	
	STOPBITS_1_5	

Tabela 167: Tipos de dados Blocos Funcionais Serial

5.12.1.1. SERIAL_CFG

Esse bloco funcional é utilizado para configurar e inicializar a porta serial desejada. Após a chamada do bloco, todas as filas RX e TX associadas à porta serial e os FIFOs RX e TX, são reiniciados.

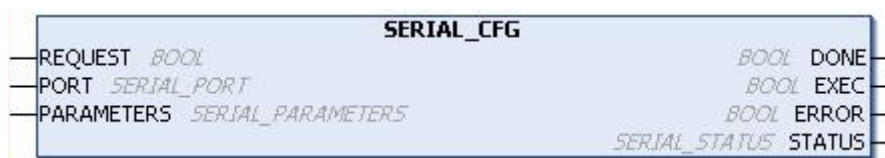


Figura 133: Bloco de Configuração da Serial

Parâmetros de entrada	Tipo	Descrição
REQUEST	BOOL	Essa variável, quando verdadeira, habilita o uso do bloco funcional.
PORT	SERIAL_PORT	Seleciona a porta serial, conforme descrito no tipo de dado SERIAL_PORT.
PARAMETERS	SERIAL_PARAMETERS	Essa estrutura define os parâmetros de configuração da porta serial, conforme descrito no tipo de dado SERIAL_PARAMETERS.

Tabela 168: Parâmetros de Entrada SERIAL_CFG

Parâmetros de saída	Tipo	Descrição
DONE	BOOL	Essa variável é verdadeira quando o bloco funcional for executado por completo, caso contrário, ela é falsa.
EXEC	BOOL	Essa variável é verdadeira enquanto o bloco funcional estiver sendo executado, caso contrário, ela é falsa.
ERROR	BOOL	Essa variável é verdadeira quando o bloco funcional concluir a sua execução com algum erro, caso contrário, ela é falsa. Está vinculada à variável DONE, pois seu estado é exibido após a conclusão do bloco funcional.
STATUS	SERIAL_STATUS	Caso a variável ERROR seja verdadeira, a estrutura STATUS exibirá o erro encontrado na execução do bloco funcional. Os estados, já descritos no tipo de dado SERIAL_STATUS, possíveis são: - NO_ERROR - ILLEGAL_SERIAL_PORT - ILLEGAL_SERIAL_MODE - ILLEGAL_BAUDRATE - ILLEGAL_DATA_BITS - ILLEGAL_PARITY - ILLEGAL_STOP_BITS - ILLEGAL_HANDSHAKE - ILLEGAL_UART_RX_THRESHOLD - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - DCD_INTERRUPT_NOT_ALLOWED - CTS_INTERRUPT_NOT_ALLOWED - DSR_INTERRUPT_NOT_ALLOWED

Tabela 169: Parâmetros de Saída SERIAL_CFG

Exemplo de utilização em Linguagem ST, após a biblioteca Nexto Serial ter sido inserida no projeto:

```

PROGRAM UserPrg
VAR
Config: SERIAL_CFG;
Port: SERIAL_PORT := COM1;
Parameters: SERIAL_PARAMETERS := (BAUDRATE := BAUD9600,
DATABITS := DATABITS_8,
STOPBITS := STOPBITS_1,
PARITY := PARITY_NONE,
HANDSHAKE := RS232_RTS,
UART_RX_THRESHOLD := 8,
MODE :=NORMAL_MODE,
ENABLE_RX_ON_TX := FALSE,
ENABLE_DCD_EVENT := FALSE,
ENABLE_CTS_EVENT := FALSE);
Status: SERIAL_STATUS;
END_VAR

```

5. CONFIGURAÇÃO

```
//ENTRADAS:
Config.REQUEST := TRUE;
Config.PORT := Port;
Config.PARAMETERS := Parameters;
//FUNÇÃO:
Config();
//SAÍDAS:
Config.DONE;
Config.EXEC;
Config.ERROR;
Status := Config.STATUS; //Caso seja necessário tratar o erro.
```

5.12.1.2. SERIAL_GET_CFG

Esse bloco funcional é utilizado para capturar as configurações da porta serial desejada.



Figura 134: Bloco para Capturar a Configuração da Serial

Parâmetros de entrada	Tipo	Descrição
REQUEST	BOOL	Essa variável, quando verdadeira, habilita o uso do bloco funcional.
PORT	SERIAL_PORT	Seleciona a porta serial, conforme descrito no tipo de dado SERIAL_PORT.

Tabela 170: Parâmetros de Entrada SERIAL_GET_CFG

Parâmetros de saída	Tipo	Descrição
DONE	BOOL	Essa variável é verdadeira quando o bloco funcional for executado por completo, caso contrário, ela é falsa.
EXEC	BOOL	Essa variável é verdadeira enquanto o bloco funcional estiver sendo executado, caso contrário, ela é falsa.
ERROR	BOOL	Essa variável é verdadeira quando o bloco funcional concluir a sua execução com algum erro, caso contrário, ela é falsa. Está vinculada à variável DONE, pois seu estado é exibido após a conclusão do bloco funcional.

Parâmetros de saída	Tipo	Descrição
STATUS	SERIAL_STATUS	Caso a variável ERROR seja verdadeira, a estrutura STATUS exibirá o erro encontrado na execução do bloco. Os estados, já descritos no tipo de dado SERIAL_STATUS, possíveis são: - NO_ERROR - ILLEGAL_SERIAL_PORT - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - NOT_CONFIGURED
PARAMETERS	SERIAL_PARAMETERS	Essa estrutura recebe os parâmetros de configuração da porta serial desejada, conforme descrito no tipo de dado SERIAL_PARAMETERS.

Tabela 171: Parâmetros de Saída SERIAL_GET_CFG

Exemplo de utilização em Linguagem ST, após a biblioteca ser inserida no projeto:

```

PROGRAM UserPrg
VAR
  GetConfig: SERIAL_GET_CFG;
  Port: SERIAL_PORT := COM1;
  Parameters: SERIAL_PARAMETERS;
  Status: SERIAL_STATUS;
END_VAR
//ENTRADAS:
GetConfig.REQUEST := TRUE;
GetConfig.PORT := Port;
//FUNÇÃO:
GetConfig();
//SAÍDAS:
GetConfig.DONE;
GetConfig.EXEC;
GetConfig.ERROR;
Status := GetConfig.STATUS; //Caso seja necessário tratar o erro.
Parameters := GetConfig.PARAMETERS; //Recebe os parâmetros da porta serial
desejada.

```

5.12.1.3. SERIAL_GET_CTRL

Esse bloco funcional é utilizado para ler os sinais de controle CTS, DSR e DCD, caso eles estejam disponíveis na porta serial. Será retornado um valor falso quando os sinais de controle não existirem.



Figura 135: Bloco para Visualizar os Sinais de Controle

Parâmetros de entrada	Tipo	Descrição
REQUEST	BOOL	Essa variável, quando verdadeira, habilita o uso do bloco funcional.
PORT	SERIAL_PORT	Seleciona a porta serial, conforme descrito no tipo de dado SERIAL_PORT.

Tabela 172: Parâmetros de Entrada SERIAL_GET_CTRL

Parâmetros de saída	Tipo	Descrição
DONE	BOOL	Essa variável é verdadeira quando o bloco funcional for executado por completo, caso contrário, ela é falsa.
EXEC	BOOL	Essa variável é verdadeira enquanto o bloco funcional estiver sendo executado, caso contrário, ela é falsa.
ERROR	BOOL	Essa variável é verdadeira quando o bloco funcional concluir a sua execução com algum erro, caso contrário, ela é falsa. Está vinculada à variável DONE, pois seu estado é exibido após a conclusão do bloco funcional.
STATUS	SERIAL_STATUS	Caso a variável ERROR seja verdadeira, a estrutura STATUS exibirá o erro encontrado na execução do bloco. Os estados, já descritos no tipo de dado SERIAL_STATUS, possíveis são: - NO_ERROR - ILLEGAL_SERIAL_PORT - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - FB_GET_CTRL_NOT_ALLOWED - NOT_CONFIGURED
CTS_VALUE	BOOL	Valor lido no sinal de controle CTS.
DSR_VALUE	BOOL	Valor lido no sinal de controle DSR.
DCD_VALUE	BOOL	Valor lido no sinal de controle DCD.

Tabela 173: Parâmetros de Saída SERIAL_GET_CTRL

Exemplo de utilização em Linguagem ST, após a biblioteca ser inserida no projeto e a porta serial ser configurada:

```

PROGRAM UserPrg
VAR
Get_Control: SERIAL_GET_CTRL;
Port: SERIAL_PORT := COM1;
Status: SERIAL_STATUS;
END_VAR
//ENTRADAS:
Get_Control.REQUEST := TRUE;
Get_Control.PORT := Port;
//FUNÇÃO:
Get_Control();
//SAÍDAS:
Get_Control.DONE;
Get_Control.EXEC;
Get_Control.ERROR;
Status := Get_Control.STATUS; //Caso seja necessário tratar o erro.
Get_Control.CTS_VALUE;
Get_Control.DSR_VALUE;
Get_Control.DCD_VALUE;
    
```

5.12.1.4. SERIAL_GET_RX_QUEUE_STATUS

Esse bloco funcional é utilizado para ler algumas informações de status sobre a fila RX, sendo especialmente desenvolvido para o modo normal, mas pode também ser utilizado no modo estendido.



Figura 136: Bloco para Visualizar o Status da Fila RX

Parâmetros de entrada	Tipo	Descrição
REQUEST	BOOL	Essa variável, quando verdadeira, habilita o uso do bloco funcional.
PORT	SERIAL_PORT	Seleciona a porta serial, conforme descrito no tipo de dado SERIAL_PORT.

Tabela 174: Parâmetros de Entrada SERIAL_GET_RX_QUEUE_STATUS

Parâmetros de saída	Tipo	Descrição
DONE	BOOL	Essa variável é verdadeira quando o bloco funcional for executado por completo, caso contrário, ela é falsa.
EXEC	BOOL	Essa variável é verdadeira enquanto o bloco funcional estiver sendo executado, caso contrário, ela é falsa.
ERROR	BOOL	Essa variável é verdadeira quando o bloco funcional concluir a sua execução com algum erro, caso contrário, ela é falsa. Está vinculada à variável DONE, pois seu estado é exibido após a conclusão do bloco funcional.
STATUS	SERIAL_STATUS	Caso a variável ERROR seja verdadeira, a estrutura STATUS exibirá o erro encontrado na execução do bloco. Os estados, já descritos no tipo de dado SERIAL_STATUS, possíveis são: - NO_ERROR - ILLEGAL_SERIAL_PORT - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - NOT_CONFIGURED
RXQ_STATUS	SERIAL_RX_QUEUE_STATUS	Retorna status/erros da fila RX, conforme descrito no tipo de dado SERIAL_RX_QUEUE_STATUS.

Tabela 175: Parâmetros de Saída SERIAL_GET_RX_QUEUE_STATUS

Exemplo de utilização em Linguagem ST, após a biblioteca ser inserida no projeto e a porta serial ser configurada:

```

PROGRAM UserPrg
VAR
Get_Status: SERIAL_GET_RX_QUEUE_STATUS;
Port: SERIAL_PORT := COM1;
Status: SERIAL_STATUS;
Status_RX: SERIAL_RX_QUEUE_STATUS;
END_VAR
//ENTRADAS:
Get_Status.REQUEST := TRUE;
Get_Status.PORT := Port;
//FUNÇÃO:
Get_Status();
//SAÍDAS:
Get_Status.DONE;
Get_Status.EXEC;
Get_Status.ERROR;
Status := Get_Status.STATUS; //Caso seja necessário tratar o erro.
Status_RX := Get_Status.RXQ_STATUS; //Caso seja necessário tratar o erro da fila
RX.

```

5.12.1.5. SERIAL_PURGE_RX_QUEUE

Esse bloco funcional é utilizado para limpar a fila RX, local e remota, da porta serial. A UART RX FIFO também é reiniciada.

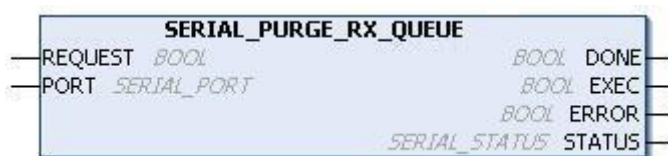


Figura 137: Bloco para Limpar a Fila RX

Parâmetros de entrada	Tipo	Descrição
REQUEST	BOOL	Essa variável, quando verdadeira, habilita o uso do bloco funcional.
PORT	SERIAL_PORT	Seleciona a porta serial, conforme descrito no tipo de dado SERIAL_PORT.

Tabela 176: Parâmetros de Entrada SERIAL_PURGE_RX_QUEUE

Parâmetros de saída	Tipo	Descrição
DONE	BOOL	Essa variável é verdadeira quando o bloco funcional for executado por completo, caso contrário, ela é falsa.
EXEC	BOOL	Essa variável é verdadeira enquanto o bloco funcional estiver sendo executado, caso contrário, ela é falsa.
ERROR	BOOL	Essa variável é verdadeira quando o bloco funcional concluir a sua execução com algum erro, caso contrário, ela é falsa. Está vinculada à variável DONE, pois seu estado é exibido após a conclusão do bloco funcional.
STATUS	SERIAL_STATUS	Caso a variável ERROR seja verdadeira, a estrutura STATUS exibirá o erro encontrado na execução do bloco. Os estados, já descritos no tipo de dado SERIAL_STATUS, possíveis são: - NO_ERROR - ILLEGAL_SERIAL_PORT - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - NOT_CONFIGURED

Tabela 177: Parâmetros de Saída SERIAL_PURGE_RX_QUEUE

Exemplo de utilização em Linguagem ST, após a biblioteca ser inserida no projeto e a porta serial ser configurada:

```

PROGRAM UserPrg
VAR
Purge_Queue: SERIAL_PURGE_RX_QUEUE;
Port: SERIAL_PORT := COM1;
Status: SERIAL_STATUS;
END_VAR
//ENTRADAS:
Purge_Queue.REQUEST := TRUE;
Purge_Queue.PORT := Port;
//FUNÇÃO:
Purge_Queue();
//SAÍDAS:
Purge_Queue.DONE;
Purge_Queue.EXEC;
Purge_Queue.ERROR;
Status := Purge_Queue.STATUS; //Caso seja necessário tratar o erro.
    
```

5.12.1.6. SERIAL_RX

Esse bloco funcional é utilizado para receber um buffer da porta serial utilizando o modo normal da fila RX. Neste modo, cada caractere na fila RX ocupa um único byte que contém o dado recebido, ou seja, armazena 5, 6, 7 ou 8 bits, de acordo com a configuração da interface serial.



Figura 138: Bloco para Ler Valores do Buffer de Recepção

Parâmetros de entrada	Tipo	Descrição
REQUEST	BOOL	Essa variável, quando verdadeira, habilita o uso do bloco funcional.
PORT	SERIAL_PORT	Seleciona a porta serial, conforme descrito no tipo de dado SERIAL_PORT.
RX_BUFFER_POINTER	POINTER TO BYTE	Ponteiro de um array de bytes para receber os valores do buffer.
RX_BUFFER_LENGTH	UINT	Especifica o número de caracteres esperados no array de bytes. Caso estejam disponíveis mais bytes do que o esperado, somente a quantidade esperada vai ser lida no array de bytes, sendo que os demais serão deixados na fila RX (tamanho máximo igual a 1024 caracteres).

Parâmetros de entrada	Tipo	Descrição
RX_TIMEOUT	UINT	Especifica o time-out para receber a quantidade de caracteres esperados. Caso ele seja menor do que a quantidade de bytes a receber, será indicado RX_TIMEOUT_ERROR no parâmetro de saída STATUS . Quando o valor especificado, em ms, for igual a zero, a função irá retornar os dados presentes no buffer.

Tabela 178: Parâmetros de Entrada SERIAL_RX

Parâmetros de saída	Tipo	Descrição
DONE	BOOL	Essa variável é verdadeira quando o bloco funcional for executado por completo, caso contrário, ela é falsa.
EXEC	BOOL	Essa variável é verdadeira enquanto o bloco funcional estiver sendo executado, caso contrário, ela é falsa.
ERROR	BOOL	Essa variável é verdadeira quando o bloco funcional concluir a sua execução com algum erro, caso contrário, ela é falsa. Está vinculada à variável DONE , pois seu estado é exibido após a conclusão do bloco funcional.
STATUS	SERIAL_STATUS	Caso a variável ERROR seja verdadeira, a estrutura STATUS exibirá o erro encontrado na execução do bloco. Os estados, já descritos no tipo de dado SERIAL_STATUS , possíveis são: - NO_ERROR - ILLEGAL_SERIAL_PORT - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - ILLEGAL_RX_BUFF_LENGTH - RX_TIMEOUT_ERROR - FB_SERIAL_RX_NOT_ALLOWED - NOT_CONFIGURED
RX_RECEIVED	UINT	Retorna o número de caracteres recebidos. Esse número pode estar entre zero e o valor configurado em RX_BUFFER_LENGTH . Caso seja menor, um erro será indicado pelo bloco funcional.
RX_REMAINING	UINT	Retorna o número de caracteres que ainda estão na fila RX depois que o bloco funcional foi executado.

Tabela 179: Parâmetros de Saída SERIAL_RX

Exemplo de utilização em Linguagem ST, após a biblioteca ser inserida no projeto e a porta serial ser configurada:

```

PROGRAM UserPrg
VAR
Receive: SERIAL_RX;
Port: SERIAL_PORT := COM1;
Buffer_Pointer: ARRAY [0..1023] OF BYTE; //Tamanho máximo.
Status: SERIAL_STATUS;
END_VAR
//ENTRADAS:
Receive.REQUEST := TRUE;
Receive.PORT := Port;
Receive.RX_BUFFER_POINTER := ADR(Buffer_Pointer);
Receive.RX_BUFFER_LENGTH := 1024; //Tamanho máximo.
Receive.RX_TIMEOUT := 10000;
//FUNÇÃO:
Receive();
//SAÍDAS:
Receive.DONE;
Receive.EXEC;
Receive.ERROR;
Status := Receive.STATUS; //Caso seja necessário tratar o erro.
Receive.RX_RECEIVED;
Receive.RX_REMAINING;

```

5.12.1.7. SERIAL_RX_EXTENDED

Esse bloco funcional é utilizado para receber um buffer da porta serial utilizando o modo estendido da fila RX, conforme detalhado na seção [Configuração das Interfaces Seriais](#).

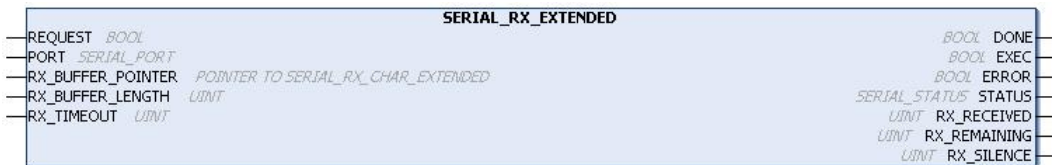


Figura 139: Bloco para Leitura do Buffer de Recepção

Parâmetros de entrada	Tipo	Descrição
REQUEST	BOOL	Essa variável, quando verdadeira, habilita o uso do bloco funcional.
PORT	SERIAL_PORT	Seleciona a porta serial, conforme descrito no tipo de dado SERIAL_PORT.
RX_BUFFER_POINTER	POINTER TO SERIAL_RX_CHAR_EXTENDED	Ponteiro de um array de SERIAL_RX_CHAR_EXTENDED para receber os valores do buffer.

Parâmetros de entrada	Tipo	Descrição
RX_BUFFER_LENGTH	UINT	Especifica o número de caracteres esperados no array de SERIAL_RX_CHAR_EXTENDED. Caso estejam disponíveis mais bytes do que o esperado, somente a quantidade esperada vai ser lida no array, sendo que os demais serão deixados na fila RX (tamanho máximo igual a 1024 caracteres).
RX_TIMEOUT	UINT	Especifica o time-out para receber a quantidade de caracteres esperados. Caso ele seja menor do que a quantidade de bytes a receber, será indicado RX_TIMEOUT_ERROR no parâmetro de saída STATUS. Quando o valor especificado, em ms, for igual a zero, a função irá retornar os dados presentes no buffer.

Tabela 180: Parâmetros de Entrada SERIAL_RX_EXTENDED

Parâmetros de saída	Tipo	Descrição
DONE	BOOL	Essa variável é verdadeira quando o bloco funcional for executado por completo, caso contrário, ela é falsa.
EXEC	BOOL	Essa variável é verdadeira enquanto o bloco funcional estiver sendo executado, caso contrário, ela é falsa.
ERROR	BOOL	Essa variável é verdadeira quando o bloco funcional concluir a sua execução com algum erro, caso contrário, ela é falsa. Está vinculada à variável DONE, pois seu estado é exibido após a conclusão do bloco funcional.
STATUS	SERIAL_STATUS	Caso a variável ERROR seja verdadeira, a estrutura STATUS exibirá o erro encontrado na execução do bloco. Os estados, já descritos no tipo de dado SERIAL_STATUS, possíveis são: - NO_ERROR - ILLEGAL_SERIAL_PORT - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - ILLEGAL_RX_BUFF_LENGTH - RX_TIMEOUT_ERROR - FB_SERIAL_RX_EXTENDED_NOT_ALLOWED - NOT_CONFIGURED

Parâmetros de saída	Tipo	Descrição
RX_RECEIVED	UINT	Retorna o número de caracteres recebidos. Esse número pode estar entre zero e o valor configurado em RX_BUFFER_LENGTH. Caso seja menor, um erro será indicado pelo bloco funcional.
RX_REMAINING	UINT	Retorna o número de caracteres que ainda estão na fila RX depois que o bloco funcional foi executado.
RX_SILENCE	UDINT	Retorna o tempo de silêncio na linha RX, medido desde o fim do último caractere recebido. A unidade de tempo é 1 μ s. Esse tipo de parâmetro de saída é importante para detectar o tempo de silêncio em protocolos como MODBUS RTU. Pode não ser o tempo de silêncio depois do último caractere recebido por esse bloco funcional, pois somente é verdade se RX_REMAINING = 0.

Tabela 181: Parâmetros de Saída SERIAL_RX_EXTENDED

Exemplo de utilização em Linguagem ST, após a biblioteca ser inserida no projeto e a porta serial ser configurada:

```

PROGRAM UserPrg
VAR
Receive_Ex: SERIAL_RX_EXTENDED;
Port: SERIAL_PORT := COM1;
Buffer_Pointer: ARRAY [0..1023] OF SERIAL_RX_CHAR_EXTENDED;
Status: SERIAL_STATUS;
END_VAR
//ENTRADAS:
Receive_Ex.REQUEST := TRUE;
Receive_Ex.PORT := Port;
Receive_Ex.RX_BUFFER_POINTER := ADR(Buffer_Pointer);
Receive_Ex.RX_BUFFER_LENGTH := 1024; //Tamanho máximo.
Receive_Ex.RX_TIMEOUT := 10000;
//FUNÇÃO:
Receive_Ex();
//SAÍDAS:
Receive_Ex.DONE;
Receive_Ex.EXEC;
Receive_Ex.ERROR;
Status := Receive_Ex.STATUS; //Caso seja necessário tratar o erro.
Receive_Ex.RX_RECEIVED;
Receive_Ex.RX_REMAINING;
Receive_Ex.RX_SILENCE;

```

5.12.1.8. SERIAL_SET_CTRL

Esse bloco funcional é utilizado para escrever nos sinais de controle (RTS e DTR), quando estes estiverem disponíveis na porta serial. Também pode determinar uma condição de ocupado para a transmissão, através do parâmetro BREAK, sendo que somente pode ser utilizado se o sinal de modem estiver configurado para RS232_MANUAL.



Figura 140: Bloco para Escrever nos Sinais de Controle

Parâmetros de entrada	Tipo	Descrição
REQUEST	BOOL	Essa variável, quando verdadeira, habilita o uso do bloco funcional.
PORT	SERIAL_PORT	Seleciona a porta serial, conforme descrito no tipo de dado SERIAL_PORT.
RTS_VALUE	BOOL	Valor a ser escrito no sinal RTS.
RTS_EN	BOOL	Habilita a escrita do parâmetro RTS_VALUE.
DTR_VALUE	BOOL	Valor a ser escrito no sinal DTR.
DTR_EN	BOOL	Habilita a escrita do parâmetro DTR_VALUE.
BREAK	BOOL	Caso seja verdadeiro, habilita lógica 0 (ocupado) na linha de transmissão.

Tabela 182: Parâmetros de Entrada SERIAL_SET_CTRL

Parâmetros de saída	Tipo	Descrição
DONE	BOOL	Essa variável é verdadeira quando o bloco funcional for executado por completo, caso contrário, ela é falsa.
EXEC	BOOL	Essa variável é verdadeira enquanto o bloco funcional estiver sendo executado, caso contrário, ela é falsa.
ERROR	BOOL	Essa variável é verdadeira quando o bloco funcional concluir a sua execução com algum erro, caso contrário, ela é falsa. Está vinculada à variável DONE, pois seu estado é exibido após a conclusão do bloco funcional.

Parâmetros de saída	Tipo	Descrição
STATUS	SERIAL_STATUS	Caso a variável ERROR seja verdadeira, a estrutura STATUS exibirá o erro encontrado na execução do bloco. Os estados, já descritos no tipo de dado SERIAL_STATUS, possíveis são: - NO_ERROR - ILLEGAL_SERIAL_PORT - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - FB_SET_CTRL_NOT_ALLOWED - NOT_CONFIGURED

Tabela 183: Parâmetros de Saída SERIAL_SET_CTRL

Exemplo de utilização em Linguagem ST, após a biblioteca ser inserida no projeto e a porta serial ser configurada:

```

PROGRAM UserPrg
VAR
Set_Control: SERIAL_SET_CTRL;
Port: SERIAL_PORT := COM1;
Status: SERIAL_STATUS;
END_VAR

//ENTRADAS:
Set_Control.REQUEST := TRUE;
Set_Control.PORT := Port;
Set_Control.RTS_VALUE := FALSE;
Set_Control.RTS_EN := FALSE;
Set_Control.DTR_VALUE := FALSE;
Set_Control.DTR_EN := FALSE;
Set_Control.BREAK := FALSE;
//FUNÇÃO:
Set_Control();
//SAÍDAS:
Set_Control.DONE;
Set_Control.EXEC;
Set_Control.ERROR;
Status := Set_Control.STATUS; //Caso seja necessário tratar o erro.

```

5.12.1.9. SERIAL_TX

Esse bloco funcional é utilizado para transmitir um buffer de dados pela porta serial, sendo que o mesmo somente é finalizado depois de todos os bytes serem transmitidos ou após o time-out (gera alguns erros).



Figura 141: Bloco para Transmitir Valores pela SERIAL_TX

Parâmetros de entrada	Tipo	Descrição
REQUEST	BOOL	Essa variável, quando verdadeira, habilita o uso do bloco funcional.
PORT	SERIAL_PORT	Seleciona a porta serial, conforme descrito no tipo de dado SERIAL_PORT.
TX_BUFFER_POINTER	POINTER TO BYTE	Ponteiro de um array de bytes para transmitir os valores do buffer.
TX_BUFFER_LENGTH	UINT	Especifica o número de caracteres a serem transmitidos pelo array de bytes (tamanho máximo da fila TX é igual a 1024 caracteres).
TX_TIMEOUT	UINT	Especifica o time-out [ms] para completar a transmissão, incluindo a fase de handshake. O valor especificado, deve ser positivo e diferente de zero.
DELAY_BEFORE_TX	UINT	Especifica o atraso [ms] entre a chamada do bloco funcional e o início da transmissão. Essa variável pode ser utilizada em comunicações com alguns modems.
CLEAR_RX_BEFORE_TX	BOOL	Quando verdadeiro, a fila RX e a UART FIFO RX são limpas antes de iniciar a transmissão. Esse comportamento é típico de protocolos mestre/escravo half-duplex.

Tabela 184: Parâmetros de Entrada SERIAL_TX

Parâmetros de saída	Tipo	Descrição
DONE	BOOL	Essa variável é verdadeira quando o bloco funcional for executado por completo, caso contrário, ela é falsa.
EXEC	BOOL	Essa variável é verdadeira enquanto o bloco funcional estiver sendo executado, caso contrário, ela é falsa.
ERROR	BOOL	Essa variável é verdadeira quando o bloco funcional concluir a sua execução com algum erro, caso contrário, ela é falsa. Está vinculada à variável DONE, pois seu estado é exibido após a conclusão do bloco funcional.

Parâmetros de saída	Tipo	Descrição
STATUS	SERIAL_STATUS	Caso a variável ERROR seja verdadeira, a estrutura STATUS exibirá o erro encontrado na execução do bloco. Os estados, já descritos no tipo de dado SERIAL_STATUS, possíveis são: - NO_ERROR - ILLEGAL_SERIAL_PORT - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - ILLEGAL_TX_BUFF_LENGTH - ILLEGAL_TIMEOUT - CTS_TIMEOUT_ON - CTS_TIMEOUT_OFF - TX_TIMEOUT_ERROR - NOT_CONFIGURED
TX_TRANSMITTED	UINT	Retorna o número de bytes transmitidos, o qual deve ser igual ao TX_BUFFER_LENGTH, mas pode ser menor caso ocorra algum erro durante a transmissão.

Tabela 185: Parâmetros de Saída SERIAL_TX

Exemplo de utilização em Linguagem ST, após a biblioteca ser inserida no projeto e a porta serial ser configurada:

```

PROGRAM UserPrg
VAR
Transmit: SERIAL_TX;
Port: SERIAL_PORT := COM1;
Buffer_Pointer: ARRAY [0..9] OF BYTE := [0,1,2,3,4,5,6,7,8,9];
Status: SERIAL_STATUS;
END_VAR

//ENTRADAS:
Transmit.REQUEST := TRUE;
Transmit.PORT := Port;
Transmit.TX_BUFFER_POINTER := ADR(Buffer_Pointer);
Transmit.TX_BUFFER_LENGTH := 10;
Transmit.TX_TIMEOUT := 10000;
Transmit.DELAY_BEFORE_TX := 1000;
Transmit.CLEAR_RX_BEFORE_TX := TRUE;
//FUNÇÃO:
Transmit();
//SAÍDAS:
Transmit.DONE;
Transmit.EXEC;
Transmit.ERROR;
Status := Transmit.STATUS; //Caso seja necessário tratar o erro.
Transmit.TX_TRANSMITTED;

```

5.12.2. Atualização de Entradas e Saídas

Funcionalidade utilizada para atualizar entradas e saídas no decorrer da aplicação, não sendo necessário aguardar até que seja completado um ciclo.

Quando os blocos funcionais para atualizar as entradas e saídas não são utilizados, a atualização é realizada a cada ciclo da MainTask.

ATENÇÃO

Na inicialização de uma UCP desta série, as entradas e saídas somente estarão atualizadas para leitura e preparadas para escrita quando a MainTask for executada. Todas as demais tarefas do sistema que executarem antes da MainTask estarão com as entradas e as saídas inválidas.

5.12.2.1. REFRESH_INPUT

Essa função é utilizada para atualizar as entradas do módulo especificado, sem aguardar o ciclo ser completado. É importante ressaltar que os filtros, configurados no MasterTool IEC XE, e o tempo de atualização das entradas do módulo, deverão ser considerados no tempo efetivo de atualização das entradas na aplicação desenvolvida pelo usuário.

ATENÇÃO

A função de *REFRESH_INPUT* deve ser utilizada somente na tarefa MainTask. Para executar a atualização de entradas em outras tarefas, a opção *Habilita atualização de E/S por tarefa* deve ser selecionada, mais informações a respeito desta opção podem ser obtidas na Tabela 44.

ATENÇÃO

A função de *REFRESH_INPUT* não suporta a atualização de entradas que tenham sido mapeadas para variáveis simbólicas. Para o correto funcionamento é necessário que a entrada esteja mapeada para uma variável dentro da memória de variáveis de entrada de representação direta (%I).

ATENÇÃO

A função de *REFRESH_INPUT* atualiza apenas as variáveis diretas %I que estiverem declaradas na aba de "Bus: Mapeamento de E/S" do módulo endereçado no respectivo rack/slot da função. Em se tratando de módulos/interfaces de comunicação (MODBUS, Profibus, etc), a atualização não inclui as variáveis diretas dos mapeamentos dos dispositivos.

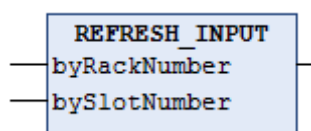


Figura 142: Bloco para Atualizar as Entradas

Parâmetros de entrada	Tipo	Descrição
byRackNumber	BYTE	Número do bastidor.
bySlotNumber	BYTE	Número da posição, na qual se encontra o módulo.

Tabela 186: Parâmetros de Entrada REFRESH_INPUT

Possíveis *TYPE_RESULT*:

- **OK_SUCCESS:** Execução com sucesso.
- **ERROR_FAILED:** Esse erro é retornado caso a função seja chamada para um módulo que possua somente saídas, ou também caso a opção Sempre atualizar variáveis (localizada na tela de configuração do módulo, aba Mapeamento de E/S) não esteja marcada.
- **ERROR_NOTSUPPORTED:** A função não é suportada pelo produto.
- **ERROR_PARAMETER:** Parâmetro inválido / não suportado.
- **ERROR_MODULE_ABSENT:** O módulo está ausente.
- **ERROR_MODULE_NOTCONFIGURED:** O módulo não está configurado na aplicação.
- **ERROR_MODULE_NOTRUNNING:** O módulo não está em funcionamento (não está no estado operacional).
- **ERROR_MODULE_COMMFAIL:** Falha de comunicação com o módulo.
- **ERROR_MODULE_NOTFOUND:** O módulo não foi encontrado na aplicação ou não é suportado.

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
Info: TYPE_RESULT;
byRackNumber: BYTE;
bySlotNumber: BYTE;
END_VAR
//ENTRADAS:
byRackNumber := 0;
bySlotNumber := 10;
//FUNÇÃO:
Info := REFRESH_INPUT (byRackNumber, bySlotNumber); //Chamada da função.
//Variável "Info" recebe possíveis erros da função.
```

5.12.2.2. REFRESH_OUTPUT

Essa função é utilizada para atualizar as saídas do módulo especificado, sem aguardar o ciclo ser completado. É importante ressaltar que o tempo de atualização das saídas do módulo deverá ser considerado no tempo efetivo de atualização das saídas na aplicação desenvolvida pelo usuário.

ATENÇÃO

A função de *REFRESH_OUTPUT* deve ser utilizada somente na tarefa *MainTask*. Para executar a atualização de saídas em outras tarefas, a opção *Habilita atualização de E/S por tarefa* deve ser selecionada, mais informações a respeito desta opção podem ser obtidas na Tabela 44.

ATENÇÃO

A função de *REFRESH_OUTPUT* não suporta a atualização de saídas que tiverem sido mapeadas para variáveis simbólicas. Para o correto funcionamento é necessário que a saída esteja mapeada para uma variável dentro da memória de variáveis de saída de representação direta (%Q).

ATENÇÃO

A função de *REFRESH_OUTPUT* atualiza apenas as variáveis diretas %Q que estiverem declaradas na aba de "Bus: Mapeamento de E/S" do módulo endereçado no respectivo rack/slot da função. Em se tratando de módulos/interfaces de comunicação (MODBUS, Profibus, etc), a atualização não inclui as variáveis diretas dos mapeamentos dos dispositivos.

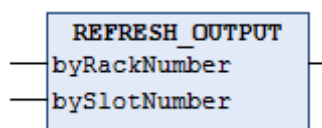


Figura 143: Bloco para Atualizar as Saídas

Parâmetros de entrada	Tipo	Descrição
byRackNumber	BYTE	Número do bastidor.
bySlotNumber	BYTE	Número da posição, na qual se encontra o módulo.

Tabela 187: Parâmetros de Entrada REFRESH_OUTPUT

Possíveis *TYPE_RESULT*:

- **OK_SUCCESS:** Execução com sucesso.
- **ERROR_FAILED:** Esse erro é retornado caso a função seja chamada para um módulo que possua somente entradas, ou também caso a opção Sempre atualizar variáveis (localizada na tela de configuração do módulo, aba Mapeamento de E/S) não esteja marcada.
- **ERROR_NOTSUPPORTED:** A função não é suportada pelo produto.
- **ERROR_PARAMETER:** Parâmetro inválido / não suportado.
- **ERROR_MODULE_ABSENT:** O módulo está ausente.
- **ERROR_MODULE_NOTCONFIGURED:** O módulo não está configurado na aplicação.
- **ERROR_MODULE_NOTRUNNING:** O módulo não está em funcionamento (não está no estado operacional).
- **ERROR_MODULE_COMMFAIL:** Falha de comunicação com o módulo.
- **ERROR_MODULE_NOTFOUND:** O módulo não foi encontrado na aplicação ou não é suportado.

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
Info: TYPE_RESULT;
byRackNumber: BYTE;
bySlotNumber: BYTE;
END_VAR
//ENTRADAS:
byRackNumber := 0;
bySlotNumber := 10;
//FUNÇÃO:
//Chamada da função.
Info := REFRESH_OUTPUT (byRackNumber, bySlotNumber);
//Variável "Info" recebe possíveis erros da função.
```

5.12.3. Bloco Funcional PID

ATENÇÃO

O bloco funcional PID descrito até a revisão anterior O deste manual tornou-se obsoleto e foi removido deste manual.

Os blocos funcionais PID, PID_INT e PID_REAL descritos até a revisão D do MP399048, também tornaram-se obsoletos e foram também removidos a partir das versões mais recentes desse manual. Usuários que precisam da descrição desses blocos funcionais obsoletos devido a motivos de manutenção devem utilizar a revisão D do MP399048.

Os blocos funcionais PID, PID_INT e PID_REAL não devem ser utilizados em novos projetos. Esses blocos funcionais foram substituídos por novos blocos funcionais com recursos adicionais, como ajuste automático e suporte à controles em cascata, sobreposição e feed-forward. Esses novos blocos funcionais são descritos no MU214610, e estão disponíveis após a versão 1.1.0.0 da biblioteca *NextoPID*.

5.12.4. Temporizador Retentivo

O Temporizador Retentivo é um bloco funcional desenvolvido para aplicações como relógios de linha de produção, as quais necessitam armazenar o seu valor e reiniciar a contagem do mesmo ponto em caso de falha na alimentação. Os valores, guardados pelo bloco funcional somente serão zerados em caso de um *Reset a Frio*, *Reset Origem* ou o *Download* de uma nova aplicação (ver Manual de Utilização do MasterTool IEC XE - MU299048), sendo que os contadores continuam em funcionamento mesmo que a aplicação esteja parada (Modo Stop).

ATENÇÃO

É importante destacar que, para o correto funcionamento dos blocos funcionais do Temporizador Retentivo, as variáveis de controle devem ser declaradas como retentivas (*VAR RETAIN*). Também é importante ressaltar que em modo simulação os blocos funcionais do Temporizador Retentivo não são executados adequadamente em virtude de necessitarem da UCP Nexto para o correto comportamento.

Abaixo, são descritos os três tipos de blocos disponíveis na biblioteca *NextoStandard* do software MasterTool IEC XE (para realizar o procedimento de inserção de uma biblioteca, consultar o Manual de Programação IEC 61131 – MP399048, capítulo Bibliotecas).

5.12.4.1. TOF_RET

O bloco funcional *TOF_RET* implementa um tempo de atraso para desabilitar uma saída. Quando a entrada *IN* tem seu estado alterado de verdadeiro (TRUE) para falso (FALSE), ou seja, uma borda de descida, o tempo especificado *PT* irá transcorrer até que a saída *Q* também seja falsa (FALSE). Quando a entrada *IN* tem nível lógico 1 (TRUE), a saída *Q* também permanecerá no mesmo estado (TRUE), mesmo que isso aconteça no meio de uma contagem. O tempo *PT* pode ser alterado durante a contagem, pois o bloco funcional assumirá o novo valor, desde que a contagem não tenha chegado ao final. A Figura 144 representa o bloco *TOF_RET* e a Figura 145 mostra o comportamento gráfico do mesmo.



Figura 144: Bloco Funcional TOF_RET

Parâmetros de entrada	Tipo	Descrição
IN	BOOL	Essa variável, quando recebe uma borda de descida, habilita a contagem do bloco funcional.
PT	TIME	Essa variável especifica o limite de contagem do bloco funcional (tempo de atraso).

Tabela 188: Parâmetros de Entrada TOF_RET

Parâmetros de saída	Tipo	Descrição
Q	BOOL	Essa variável vai para FALSE assim que a variável PT (tempo de atraso) atinge o seu valor máximo.
ET	TIME	Essa variável exibe o valor atual do tempo de atraso.

Tabela 189: Parâmetros de Saída TOF_RET

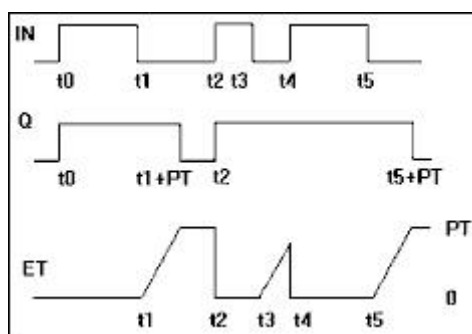


Figura 145: Comportamento Gráfico do Bloco Funcional TOF_RET

Exemplo de utilização em linguagem ST:

```

PROGRAM UserPrg
VAR RETAIN
bStart : BOOL := TRUE;
TOF_RET : TOF_RET;
END_VAR

// Quando bStart=FALSE inicia contagem
TOF_RET( IN := bStart,
PT := T#20S);

// Executa ações ao final da contagem
IF (TOF_RET.Q = FALSE) THEN
bStart := TRUE;
END_IF

```

5.12.4.2. TON_RET

O bloco funcional *TON_RET* implementa um tempo de atraso para habilitar uma saída. Quando a entrada *IN* tem seu estado alterado de falso (FALSE) para verdadeiro (TRUE), ou seja, uma borda de subida, o tempo especificado *PT* irá transcorrer até que a saída *Q* também seja verdadeira (TRUE). Quando a entrada *IN* tem nível lógico 0 (FALSE), a saída *Q* também permanecerá no mesmo estado (FALSE), mesmo que isso aconteça no meio de uma contagem. O tempo *PT* pode ser alterado durante a contagem, pois o bloco funcional assumirá o novo valor, desde que a contagem não tenha chegado ao final. A Figura 146 representa o bloco *TON_RET* e a Figura 147 mostra o comportamento gráfico do mesmo.



Figura 146: Bloco Funcional TON_RET

Parâmetros de entrada	Tipo	Descrição
IN	BOOL	Essa variável, quando recebe uma borda de subida, habilita a contagem do bloco funcional.
PT	TIME	Essa variável especifica o limite de contagem do bloco funcional (tempo de atraso).

Tabela 190: Parâmetros de Entrada TON_RET

Parâmetros de saída	Tipo	Descrição
Q	BOOL	Essa variável vai para TRUE assim que a variável PT (tempo de atraso) atinge o seu valor máximo.
ET	TIME	Essa variável exibe o valor atual do tempo de atraso.

Tabela 191: Parâmetros de Saída TON_RET

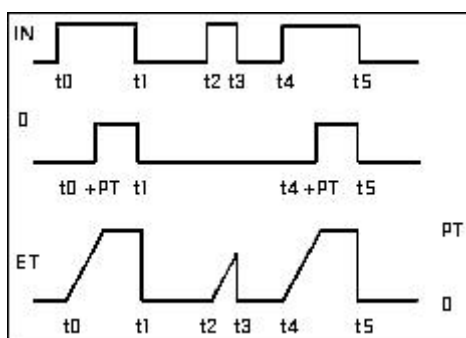


Figura 147: Comportamento Gráfico do Bloco Funcional TON_RET

Exemplo de utilização em linguagem ST:

```

PROGRAM UserPrg
VAR RETAIN
bStart : BOOL;
TON_RET : TON_RET;
END_VAR

// Quando bStart=TRUE inicia contagem
TON_RET( IN := bStart,
PT := T#20S);

// Executa ações ao final da contagem
IF (TON_RET.Q = TRUE) THEN
bStart := FALSE;
END_IF

```

5.12.4.3. TP_RET

O bloco funcional *TP_RET* trabalha como um trigger. O temporizador, que inicia quando a entrada *IN* tem seu estado alterado de falso (FALSE) para verdadeiro (TRUE), ou seja, uma borda de subida, é incrementado até que o limite de tempo *PT* seja atingido. Durante a contagem, a saída *Q* é verdadeira (TRUE), caso contrário ela é falsa (FALSE). O tempo *PT* pode ser alterado durante a contagem, pois o bloco assumirá o novo valor, desde que a contagem não tenha chegado ao final. A Figura 148 representa o bloco *TP_RET* e a Figura 149 mostra o comportamento gráfico do mesmo.



Figura 148: Bloco Funcional TP_RET

Parâmetros de entrada	Tipo	Descrição
IN	BOOL	Essa variável, quando recebe uma borda de subida, habilita a contagem do bloco funcional.
PT	TIME	Essa variável especifica o limite de contagem do bloco funcional (tempo de atraso).

Tabela 192: Parâmetros de Entrada TP_RET

Parâmetros de saída	Tipo	Descrição
Q	BOOL	Essa variável é verdadeira durante a contagem. Caso contrário é falsa.
ET	TIME	Essa variável exibe o valor atual do tempo de atraso.

Tabela 193: Parâmetros de Saída TP_RET

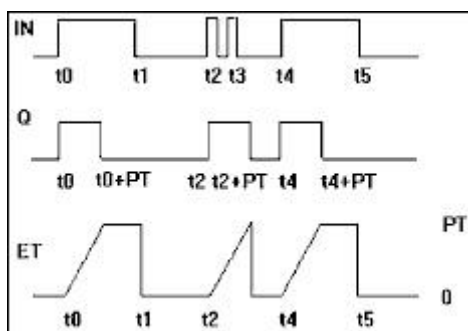


Figura 149: Comportamento Gráfico do Bloco Funcional TP_RET

Exemplo de utilização em linguagem ST:

```
PROGRAM UserPrg
VAR RETAIN
bStart : BOOL;
TP_RET : TP_RET;
END_VAR

// Configura TP_RET
TP_RET( IN := bStart,
PT := T#20S);

bStart := FALSE;

// Ações durante a contagem
IF (TP_RET.Q = TRUE) THEN
// Executa enquanto o contador estiver ativado
ELSE
// Executa somente quando o contador estiver desativado
END_IF
```

5.12.5. Temporizador Não-Redundante

O temporizador não redundante é utilizado em aplicações para as UCPs redundantes que necessitam de um temporizador no programa não redundante de um half-cluster. Este temporizador não utiliza o temporizador IEC, portanto, não irá ser sincronizado no caso de o half-cluster reserva assumir o estado ativo e o ativo passar para reserva.

Abaixo, são descritos os três tipos de blocos já disponíveis na biblioteca *NextoStandard* do software MasterTool IEC XE (para realizar o procedimento de inserção de uma biblioteca, consultar o Manual de Programação IEC 61131 – MP399048, capítulo Bibliotecas).

5.12.5.1. TOF_NR

O bloco funcional *TOF_NR* implementa um tempo de atraso para desabilitar uma saída e tem o funcionamento e configuração parecidos com o bloco funcional *TOF_RET*, se diferenciando apenas por não ser redundante e nem retentivo.

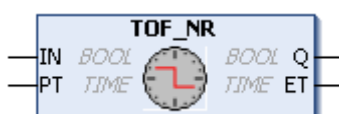


Figura 150: Bloco Funcional TOF_NR

Exemplo de utilização em linguagem ST:

```
PROGRAM NonSkippedPrg
VAR
bStart : BOOL := TRUE;
TOF_NR : TOF_NR;
END_VAR

// Quando bStart=FALSE inicia contagem
TOF_NR( IN := bStart,
PT := T#20S);

// Executa ações ao final da contagem
IF (TOF_NR.Q = FALSE) THEN
bStart := TRUE;
END_IF
```

5.12.5.2. TON_NR

O bloco funcional *TON_NR* implementa um tempo de atraso para habilitar uma saída e tem o funcionamento e configuração parecidos com o bloco funcional *TON_RET*, se diferenciando apenas por não ser redundante e nem retentivo.



Figura 151: Bloco Funcional TON_NR

Exemplo de utilização em linguagem ST:

```
PROGRAM NonSkippedPrg
VAR
bStart : BOOL;
TON_NR : TON_NR;
END_VAR

// Quando bStart=TRUE inicia contagem
TON_NR( IN := bStart,
PT := T#20S);

// Executa ações ao final da contagem
IF (TON_NR.Q = TRUE) THEN
bStart := FALSE;
END_IF
```

5.12.5.3. TP_NR

O bloco funcional *TP_NR* trabalha como um trigger e tem o funcionamento e configuração parecidos com o bloco funcional *TP_RET*, se diferenciando apenas por não ser redundante e nem retentivo.

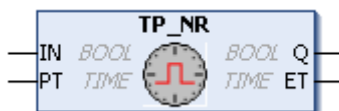


Figura 152: Bloco Funcional TP_NR

Exemplo de utilização em linguagem ST:

```
PROGRAM NonSkippedPrg
VAR
bStart : BOOL;
TP_NR : TP_NR;
END_VAR

// Configura TP_NR
TP_NR( IN := bStart,
PT := T#20S);

bStart := FALSE;

// Ações durante a contagem
IF (TP_NR.Q = TRUE) THEN
// Executa enquanto o contador estiver ativado
ELSE
// Executa somente quando o contador estiver desativado
END_IF
```

5.12.6. Log de Usuário

Recurso que permite ao usuário criar seus próprios registros e gravar em arquivos de log no cartão de memória presente na UCP. Os arquivos são gerados em um diretório específico do cartão de memória no formato CSV, permitindo a visualização em editores de texto e planilhas. O separador adotado foi o caractere ponto e vírgula. Para mais informações sobre a utilização do cartão de memória, ver seção [Cartão de Memória](#).

Há duas funções disponíveis, uma para registrar informações e outra para remover todos os registros. A seguir uma descrição dos tipos de dados utilizados pelas funções:

Tipo de dado	Opção	Descrição
USER_LOG_EVENT_TYPES	USER_LOG_EVENT_ERROR	O usuário é livre para usar a melhor indicação conforme a severidade de sua mensagem de log.
	USER_LOG_EVENT_DEBUG	
	USER_LOG_EVENT_INFO	
	USER_LOG_EVENT_WARN	
USER_LOG_MESSAGE		Mensagem de log com o máximo de 150 caracteres.
USER_LOG_ERROR_CODES	USER_LOG_OK	A operação foi executada com sucesso.
	USER_LOG_FAILED	A operação não foi executada com sucesso. O motivo da falha pode ser verificado nos logs de sistema – ver seção Log de Sistema .
	USER_LOG_BUFFER_FULL	Mensagens estão sendo adicionadas além da capacidade de processamento.
	USER_LOG_NO_MEMORY	No momento, não houve recursos para executar a operação.
	USER_LOG_FILE_SYSTEM_ERROR	Houve algum erro no acesso ao cartão de memória ou não há espaço disponível. Informações do erro podem ser verificadas nos logs de sistema – ver seção Log de Sistema .
	USER_LOG_NO_MEMORY_CARD	Não há um cartão de memória presente na UCP.
	USER_LOG_MEMORY_CARD_FULL	Não há espaço livre disponível no cartão de memória.
	USER_LOG_PROCESSING	O recurso está ocupado executando a última operação, por exemplo, deletando todos os arquivos de log.

Tabela 194: Tipos de dados para Log de Usuário

A seguir, são descritas as duas funções disponíveis na biblioteca *LibLogs* no MasterTool IEC XE. Para realizar o procedimento de inserção de uma biblioteca, consultar o Manual de Programação IEC 61131 – MP399048, capítulo Bibliotecas.

ATENÇÃO

Os Logs de Usuário estão disponíveis apenas a partir da versão 1.3.0.20 das UCPs da Série Nexto. Da mesma forma, para utilizar esta funcionalidade é necessária a versão 1.40 ou superior do MasterTool IEC XE.

5.12.6.1. UserLogAdd

Essa função é utilizada para adicionar uma nova mensagem de log do usuário, acrescentando em uma nova linha ao arquivo de log no cartão de memória. A mensagem deve ter um tamanho máximo de 150 caracteres e o tipo de evento da mensagem. Variáveis da aplicação podem ser registradas utilizando conversão para string e concatenação com a mensagem principal. É adicionada automaticamente na mensagem a informação de data e hora em UTC (estampa de tempo) com resolução de milissegundos em que o evento foi registrado. A informação de data e hora também é utilizada na formação dos nomes dos arquivos de log.

A função *UserLogAdd* pode ser utilizada para inserir várias mensagens dentro de uma mesma tarefa e também em tarefas diferentes da aplicação. Porém independente de cada execução da função na aplicação, sendo ela na mesma tarefa ou em tarefas diferentes, todas utilizam o mesmo recurso para gravar as mensagens desejadas. Por esse motivo é recomendado que a adição de mensagens utilizando a função *UserLogAdd* na aplicação seja realizada a cada 50 ms para evitar retorno de buffer cheio. Caso a função seja executada em períodos menores que o indicado, mas respeite o tempo médio de 50 ms entre cada adição

de mensagem ao término do intervalo da tarefa, também evita o retorno de buffer cheio. Para que os logs sejam adicionados corretamente, é importante respeitar os limites de tempo quando o cartão é inserido e na inicialização da UCP, mencionados na seção [Cartão de Memória](#). Após a operação a função retorna as opções para o tipo de dado `USER_LOG_ERROR_CODES`, conforme a Tabela 194. Quando o retorno da função for diferente de `USER_LOG_OK`, a mensagem não foi registrada, devendo ser executada novamente a função `UserLogAdd` com a mensagem desejada. Em caso de retorno de falhas consecutivas de escrita, o cartão de memória pode estar danificado. A substituição por um cartão de memória íntegro garante que as últimas mensagens registradas serão gravadas no cartão que não está danificado, desde que a UCP não seja reiniciada.

A figura abaixo representa a função `UserLogAdd` e a tabela abaixo os parâmetros de entrada:

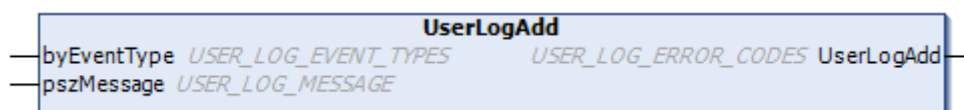


Figura 153: Função UserLogAdd

Parâmetros de entrada	Tipo	Descrição
byEventType	BYTE	Essa variável especifica o tipo de evento do log que está sendo adicionado conforme as opções para o tipo de dado <code>USER_LOG_EVENT_TYPES</code> .
pszMessage	<code>USER_LOG_MESSAGE</code>	Essa variável deve conter o conjunto de caracteres que compõe a mensagem a ser adicionada no arquivo de log. A mensagem deve conter no máximo 150 caracteres.

Tabela 195: Parâmetros de Entrada UserLogAdd

Os arquivos de log são gerados e organizados no cartão de memória em um caminho de diretórios específico que depende do número de série da UCP e da versão de firmware instalada. Por exemplo, para uma UCP com número de série 445627 e versão de firmware 1.4.0.4, o local onde os arquivos de log devem ser gravados no cartão de memória é `MemoryCard/UserLog/445627/1.4.0.4/`.

Os nomes dos arquivos de log são formados pela data e hora (estampa de tempo) da primeira mensagem. Exceto quando há algum problema para utilizar esse nome, como por exemplo, outro arquivo existente com o mesmo nome, nessa situação são utilizadas a data e hora instantâneas. O nome do arquivo segue o seguinte padrão: `ano/mês/dia/hora/minuto/segundo/milissegundo.CSV`. Caso um arquivo apresente problema de acesso por setor defeituoso e não seja possível continuar com escrita, será adicionado ao nome desse arquivo a extensão `"corrupted"` e um novo arquivo será criado. A quantidade de logs por arquivo não é fixa, variando em função do tamanho das mensagens. A quantidade de arquivos criados é limitada em 1024 com tamanho máximo de 1 MB cada, portanto o cartão de memória necessita de 1 GB de espaço livre. Quando atingir o limite de 1024 arquivos criados no cartão de memória, durante a operação da UCP, os arquivos mais antigos são removidos para que os arquivos com logs mais recentes sejam preservados, mesmo nos casos de remoção manual parcial dos arquivos no diretório onde os arquivos estão sendo gravados.

A visualização dos arquivos de log pode ser realizada através de planilhas ou editores de texto convencionais. As informações concatenadas, para melhor visualização, podem utilizar ponto e vírgula entre as strings da mensagem para separá-las. Deve-se ter o cuidado na formatação de células com valores em ponto flutuante.

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
  eLogError : USER_LOG_ERROR_CODES;
  sMessage : USER_LOG_MESSAGE;
END_VAR

IF (m_rTemperature > MAX_TEMPERATURE_ACCEPT) THEN
  sMessage := 'Temperatura acima do esperado: ';
  sMessage := concat (sMessage, REAL_TO_STRING (m_rTemperature));
  sMessage := concat (sMessage, '°');
```

5. CONFIGURAÇÃO

```
eLogError := UserLogAdd(USER_LOG_EVENT_WARN, sMessage);  
//Variável eLogError recebe possíveis erros da função.  
END_IF
```

Exemplo de conteúdo do arquivo de log: (*UserLog-201308271506245738.csv*)

```
Model; NX3030  
Serial number; 445627  
Firmware version; 1.4.0.4  
  
27/08/2013 15:06:24.5738; WARN; Temperatura acima do esperado: 25°  
27/08/2013 16:37:45.3476; WARN; Temperatura acima do esperado: 25°  
28/08/2013 09:10:55.4201; WARN; Temperatura acima do esperado: 26°
```

5.12.6.2. UserLogDeleteAll

A função *UserLogDeleteAll* realiza a exclusão dos arquivos de log presentes no diretório criado especificamente para a UCP em que está inserido o cartão de memória, ou seja, são excluídos apenas os logs contidos no diretório nomeado com a versão de firmware da UCP que existe dentro do diretório com a versão de série da UCP. Os arquivos de log excluídos são apenas os arquivos existentes no momento da montagem do cartão de memória e os gerados pela função *UserLogAdd*. Registros de outras UCPs e arquivos adicionados manualmente pelo usuário durante a execução não são apagados. A figura abaixo representa a função *UserLogDeleteAll*.

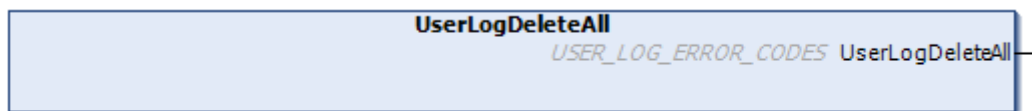


Figura 154: Função UserLogDeleteAll

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg  
VAR  
eLogError : USER_LOG_ERROR_CODES;  
END_VAR  
  
IF (m_DeleteLogs = TRUE) THEN  
//Variável eLogError recebe possíveis erros da função.  
eLogError := UserLogDeleteAll();  
m_DeleteLogs := FALSE;  
END_IF
```

ATENÇÃO

O retorno da função *UserLogDeleteAll* não indica operação concluída, apenas confirmação de execução que pode levar um período de tempo grande caso existam centenas de arquivos de log no diretório. A função para registro de novos logs de usuário estará indisponível nesse momento, retornando a opção *USER_LOG_PROCESSING* para qualquer operação. O resultado da operação também pode ser verificado no log de sistema.

5.12.7. ClearRtuDiagnostic

Esta função não é suportada por esta série de UCPs.

5.12.8. ClearEventQueue

A função *ClearEventQueue* disponibilizada pela biblioteca *LibRtuStandard* pode ser utilizada quando for necessário limpar a fila de eventos da UCP e de todos os drivers instanciados.

Conforme a tabela abaixo, o resultado de execução da função será mostrado na sua variável de saída.

ATENÇÃO

A função *ClearEventQueue* não se aplica à limpeza da fila de serviço de Registro de Eventos (SOE), descrito na seção [Configuração do SOE](#). A função limpa apenas as filas de eventos dos drivers configurados sob as interfaces de comunicação (COMs e NETs) da UCP.

Nome	ENUM (UDINT)	Descrição
OK_SUCCESS	0	Sucesso
ERROR_FAILED	1	Erro
ERROR_NOTSUPPORTED	2	A rotina chamada não é suportada pelo produto
ERROR_PARAMETER	3	Parâmetro inválido/não suportado
ERROR_MODULE_ABSENT	16	O módulo está ausente no barramento
ERROR_MODULE_NOTCONFIGURED	17	O módulo não está configurado na aplicação
ERROR_MODULE_NOTRUNNING	18	O módulo não está rodando (não está em estado operacional)
ERROR_MODULE_COMMFAIL	19	Falha na comunicação com o módulo
ERROR_MODULE_NOTFOUND	20	O módulo não foi encontrado na aplicação ou não é suportado

Tabela 196: Resultados da função ClearEventQueue

Exemplo de utilização em linguagem ST, onde a chamada da função irá limpar a fila de eventos e, conseqüentemente, zerar os diagnósticos de uso das filas de eventos dos drivers de comunicação *T_DIAG_DNP_SERVER_1.tClient_*.tQueueDiags.wUsage*:

```
PROGRAM UserPrg
VAR
  ClearEventQueueStatus : TYPE_RESULT;
END_VAR

ClearEventQueueStatus := ClearEventQueue();
```

5.13. SNMP

5.13.1. Introdução

SNMP (*Simple Network Management Protocol*) é um protocolo muito utilizado por administradores de redes por fornecer importantes informações e diagnósticos de equipamentos presentes em determinada rede Ethernet.

Este protocolo usa o conceito de agente e gerente, no qual o gerente envia requisições de leitura ou escrita de determinados objetos para o agente. Através de uma MIB (*Management Information Base*) o gerente tem conhecimento dos objetos existentes no agente, e dessa forma poderá fazer requisições desses objetos, respeitando as permissões de leitura ou escrita dos mesmos. MIB é uma coleção de informações organizadas hierarquicamente, onde cada objeto desta árvore é chamado de OID (*Object Identifier*).

Para todos os equipamentos com SNMP, é obrigatório o suporte a MIB-II. Nesta MIB estão descritas as principais informações para gerenciamento de redes Ethernet.

5.13.2. SNMP nas UCPs Nexto

As UCPs da Série Nexto se comportam como agentes na comunicação SNMP. As informações disponibilizadas através do SNMP não podem ser manipuladas ou acessadas através da aplicação do usuário, sendo necessário um gerente SNMP externo para efetuar o acesso. A tabela abaixo contém os objetos disponíveis nas UCPs Nexto. Esta funcionalidade está disponível a partir da versão de firmware 1.4.0.33, das UCPs da Série Nexto, possuindo suporte aos protocolos SNMPv1, SNMPv2c e SNMPv3, além de suporte a MIB-II, onde objetos são descritos na RFC-1213.

OID	Nome	Descrição
1.3.6.1.2.1.1	System	Contém nome, descrição, localização e outras informações de identificação do equipamento.
1.3.6.1.2.1.2	Interfaces	Contém informações das interfaces de rede do equipamento. A tabela ifTable (OID 1.3.6.1.2.1.2.2) possui os índices 6 e 7 disponíveis, pelos quais podem ser visualizadas as estatísticas das interfaces de rede NET 1 e NET 2, respectivamente, das UCPs da série Nexto.
1.3.6.1.2.1.3	At	Contém informações das últimas conexões requisitadas ao agente.
1.3.6.1.2.1.4	Ip	Contém estatísticas de conexões utilizando protocolo IP.
1.3.6.1.2.1.5	Icmp	Contém estatísticas para o protocolo ICMP.
1.3.6.1.2.1.6	Tcp	Contém estatísticas para o protocolo TCP.
1.3.6.1.2.1.7	Udp	Contém estatísticas para o protocolo UDP.
1.3.6.1.2.1.11	Snmp	Contém estatísticas para o protocolo SNMP.

Tabela 197: Objetos da MIB-II - Agente SNMP da Série Nexto

Por padrão, o agente SNMP está ativado, ou seja, o serviço é inicializado no momento em que a UCP é iniciada. O acesso às informações do agente se dá através das interfaces Ethernet das UCPs da Série Nexto na porta UDP 161. Portanto quando o serviço estiver ativo, as informações do agente poderão ser acessadas através de qualquer uma das interfaces Ethernet, conforme a disponibilidade da UCP em uso. Não é possível disponibilizar informações do agente através de interfaces Ethernet dos módulos NX5000. Na figura abaixo é mostrado um exemplo de gerente SNMP, no qual são lidos alguns valores.

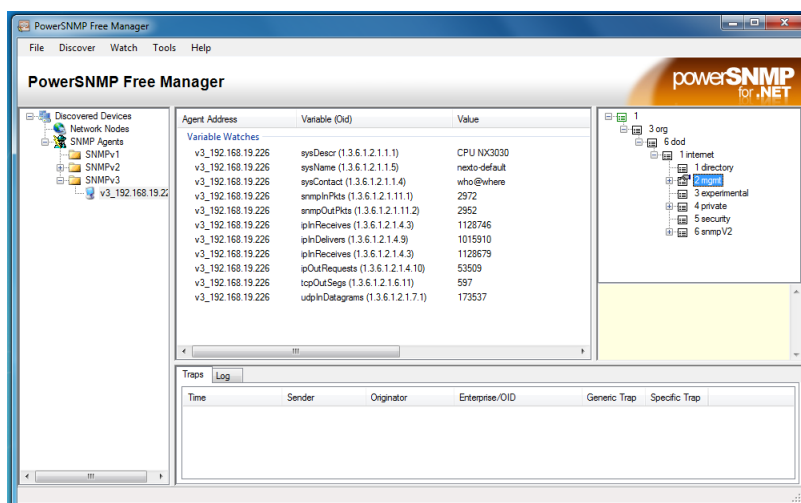


Figura 155: Exemplo de Gerente SNMP

Para SNMPv3, na qual existe autenticação de usuário e senha para requisições via protocolo SNMP, é fornecido um usuário padrão descrito na seção [Usuário e Comunidades SNMP](#).

Caso o usuário deseje desativar o serviço, alterar o usuário SNMPv3 ou as comunidades para SNMPv1/v2c pré-definidos, será necessário acessar a Página Web de Sistema da UCP. Para mais detalhes consulte a seção [Configuração SNMP](#).

5.13.3. Private MIB

O Private MIB foi descontinuado desde Junho de 2019.

5.13.4. Configuração SNMP

As configurações do SNMP podem ser alteradas através da Página Web de Sistema da UCP, na aba *Gerenciamento*, na seção *SNMP*. Efetuado o login com sucesso, poderá ser visualizado o estado atual do serviço (ativado ou desativado), assim como as informações de usuário SNMPv3 e comunidades para SNMPv1/v2c.

O usuário poderá ativar ou desativar o serviço através de um checkbox na parte superior da tela.

É possível também alterar as informações de SNMPv3, clicando no botão *Alterar* logo abaixo das informações de usuário. Abrirá um formulário onde é necessário preencher o usuário e senha antigos, e o novo usuário e senha. As demais informações de usuário SNMPv3 não podem ser alteradas.

Para alterar os dados das comunidades SNMPv1/v2c, o processo é parecido, basta clicar no botão *Alterar* abaixo das informações das comunidades. Uma nova tela será aberta onde serão inseridos os novos dados para os campos *rocommunity* e *rwcommunity*. Caso o usuário deixe qualquer um dos campos em branco, a respectiva comunidade será desativada. Dessa forma, se o usuário deixar os 2 campos em branco, o acesso ao agente SNMP será possível somente através do SNMPv3.

Caso o usuário deseje retornar para as configurações padrão, será necessário reconfigurar manualmente as mesmas de acordo com a seção [Usuário e Comunidades SNMP](#). Portanto, todas as configurações SNMP atuais serão mantidas no processo de atualização de firmware. Estas opções podem ser visualizadas na figura abaixo.

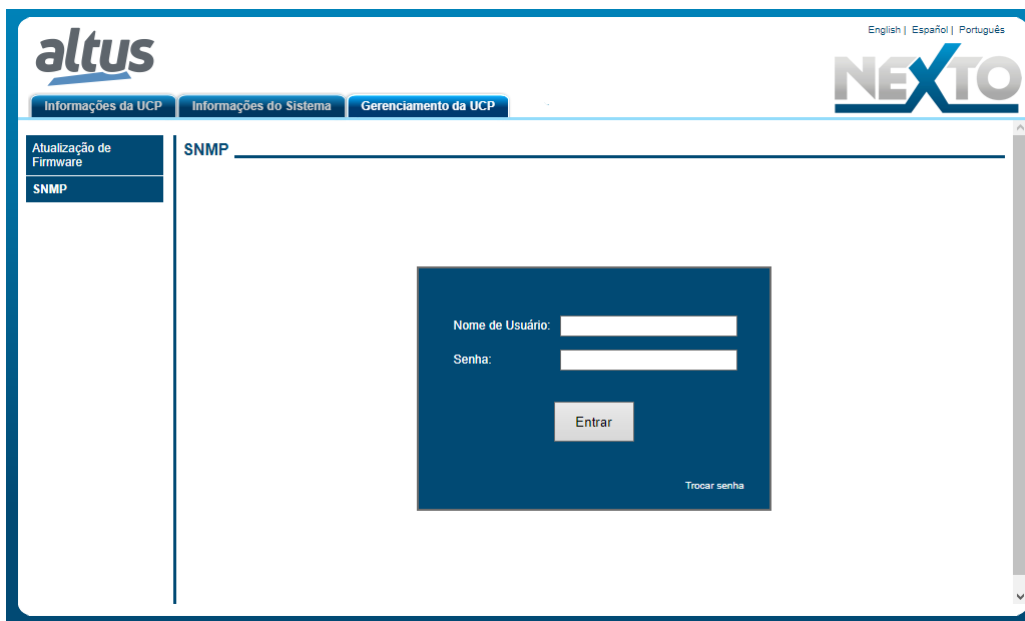


Figura 156: Tela de Login SNMP

Efetuada o login com sucesso, poderá ser visualizado o estado atual do serviço (ativado ou desativado), assim como as informações de usuário SNMPv3 e comunidades para SNMPv1/v2c.

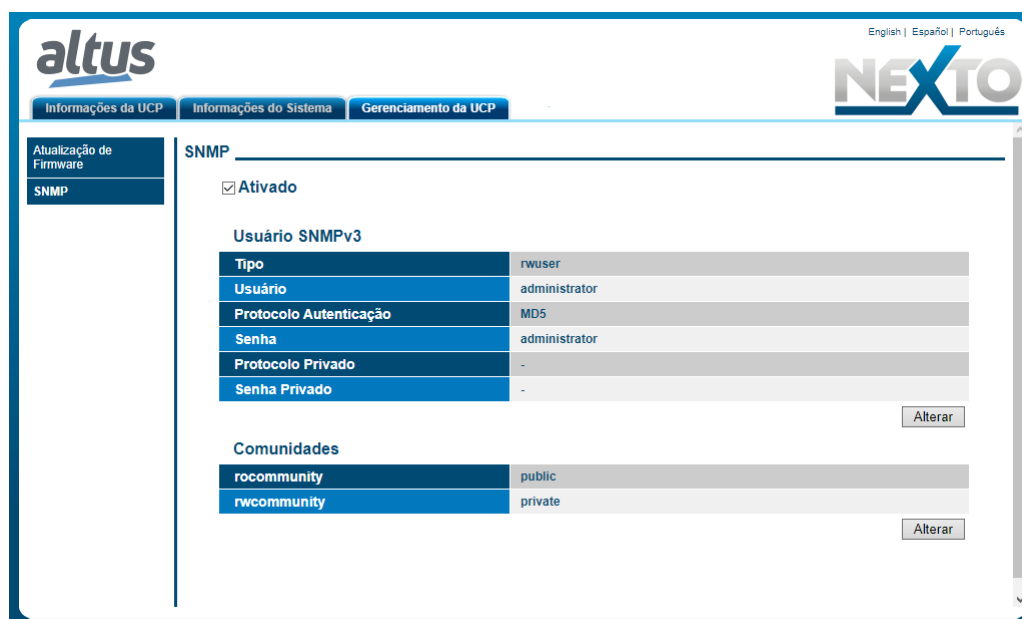


Figura 157: Tela de Configuração e Status SNMP

ATENÇÃO

O *Usuário* e *Senha* para acesso ao agente via protocolo SNMP são os mesmos utilizados para login na seção SNMP.

5.13.5. Usuário e Comunidades SNMP

Para acesso ao SNMPv1/v2c das UCPs da Série Nexto, existem duas comunidades, conforme tabela abaixo.

Comunidades	String Padrão	Tipo
rocommunity	Public	Somente leitura
rwcommunity	Private	Leitura e escrita

Tabela 198: Informações de Comunidades Padrão SNMPv1/v2c

É possível efetuar acesso ao SNMPv3 através do usuário padrão conforme tabela abaixo.

Usuário	Tipo	Protocolo Autenticação	Senha	Protocolo Privado	Senha Privado
administrator	rwuser	MD5	administrator	-	-

Tabela 199: Informações de Usuário Padrão SNMPv3

Para todas as configurações de comunidades, usuário e senha existem limites que devem ser observados na tabela abaixo.

Item configurável	Tamanho mínimo	Tamanho máximo	Caracteres permitidos
rocommunity	-	30	[0-9][a-z][A-Z]@\$*_.
rwcommunity	-	30	[0-9][a-z][A-Z]@\$*_.
Usuário v3	-	30	[0-9][a-z][A-Z]@\$*_.
Senha v3	8	30	[0-9][a-z][A-Z]@\$*_.

Tabela 200: Limites para Configurações SNMP

6. Redundância com UCP NX3030

6.1. Introdução

Este capítulo aborda a redundância de CPs da Série Nexto, que pode ser utilizada somente com a UCP NX3030.

Trata-se de uma redundância do tipo hot-standby, onde os controladores são duplicados. Um dos controladores normalmente está em estado Ativo e controlando o processo, enquanto o outro controlador normalmente está em estado Reserva, mantendo-se sincronizado com o controlador Ativo. Caso ocorra uma falha no controlador Ativo, que o impeça de continuar controlando o processo, o controlador Reserva chaveia automaticamente para Ativo, em um tempo suficientemente baixo para não perturbar o processo, sem causar descontinuidades nas saídas que controlam o processo.

A redundância hot-standby é um método utilizado para aumentar a tolerância a falhas e, conseqüentemente, aumentar a disponibilidade do sistema de automação. A ideia básica é que nenhuma falha simples em componentes duplicados cause a interrupção do controle do processo.

A redundância hot-standby é muito aplicada em:

- plataformas de exploração de petróleo
- sistemas de geração e distribuição de energia
- intertravamentos de segurança (Sistemas Instrumentados de Segurança)
- processos contínuos, tais como plantas químicas, refinarias de petróleo, produção de celulose, etc.

Além dos controladores, podem ser também duplicadas, opcionalmente, as redes de campo (PROFIBUS DP e Ethernet), as redes de supervisão Ethernet, e as redes de controle Ethernet HSDN (*High Speed Deterministic Network*). Optando-se pela duplicação destas redes, obtém-se uma disponibilidade ainda maior.

A redundância hot-standby de CPs da Série Nexto não prevê duplicação de módulos de E/S. Caso a redundância de módulos de E/S seja desejável, ela pode ser tratada em nível de aplicação, pelo usuário final. Por exemplo, o usuário pode duplicar ou até mesmo triplicar um módulo de entradas analógicas, e criar um algoritmo de votação para determinar qual das entradas irá ser considerada em determinado momento em sua aplicação.

A figura abaixo mostra um exemplo típico de arquitetura redundante com a UCP NX3030.

A parte central de um CP redundante é formada por dois bastidores idênticos, denominados CPA e CPB, além de um painel de controle de redundância (PX2612), quando esta opção for configurada. No contexto da redundância, cada bastidor (CPA e CPB) é denominado half-cluster, enquanto o conjunto formado por estes dois bastidores é denominado cluster.

Neste exemplo, observam-se também duplicações em uma rede de campo PROFIBUS, na rede Ethernet de supervisão, e na rede de controle Ethernet HSDN.

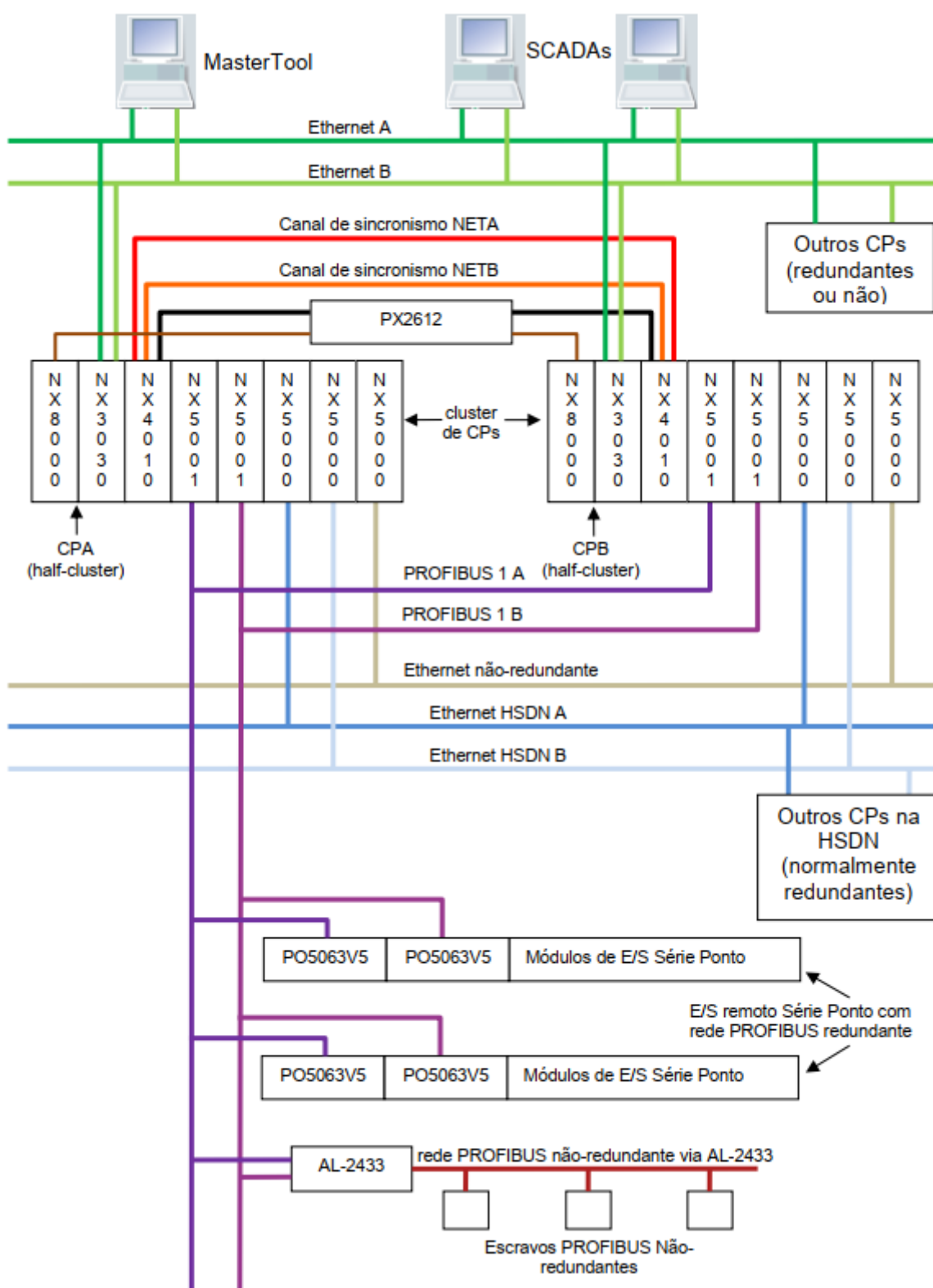


Figura 158: Exemplo de Arquitetura Redundante com UCP NX3030

6.2. Descrição Técnica e Configuração

6.2.1. Configuração Mínima de um CP Redundante (Sem utilização do Painel PX2612)

Um CP redundante é composto, no mínimo, de:

- dois half-clusters idênticos

Cada half-cluster é constituído, no mínimo, dos seguintes módulos:

- o próprio bastidor onde os módulos são inseridos, que pode ser um dos seguintes:
 - NX9000, com 8 posições
 - NX9001, com 12 posições
 - NX9002, com 16 posições
 - NX9003, com 24 posições
- a fonte de alimentação NX8000, nas posições 0 e 1 do bastidor
- a UCP NX3030, nas posições 2 e 3 do bastidor
- o módulo NX4010, nas posições 4 e 5 do bastidor

A figura abaixo mostra um exemplo de configuração mínima de um CP redundante, utilizando o bastidor NX9001 (12 posições). Neste caso, observa-se que os 3 módulos inseridos no bastidor têm largura dupla (ocupam duas posições do bastidor).

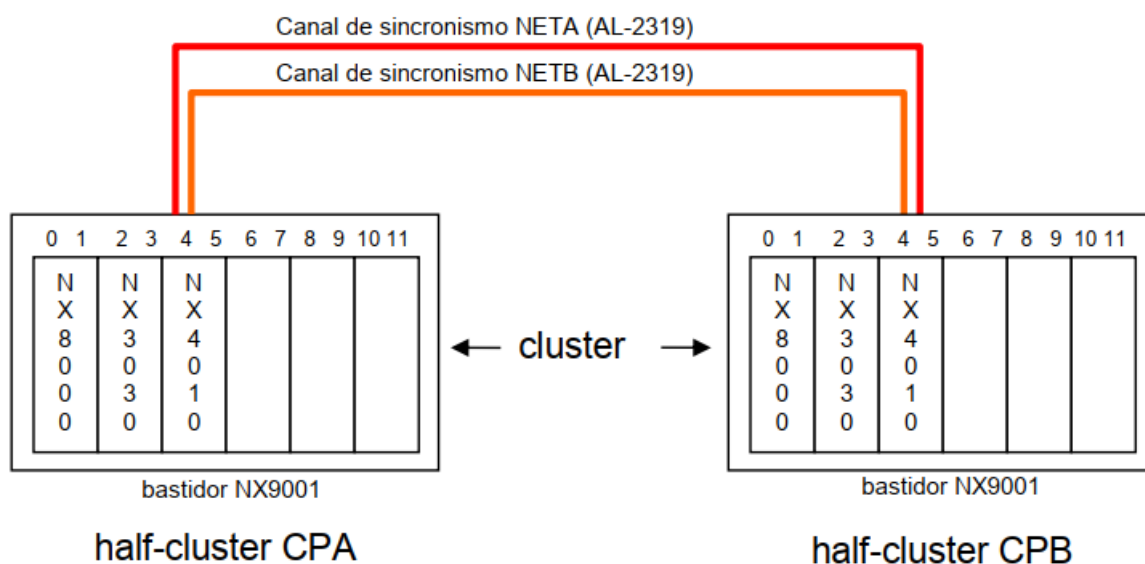


Figura 159: Configuração Mínima de um CP Redundante em Bastidor NX9001

6.2.2. Configurações Típicas de um CP Redundante

Uma configuração mínima, como esta mostrada na Figura 159, já pode ser útil como uma “unidade de processamento redundante”. Seria possível utilizar os canais de comunicação seriais e Ethernet da UCP NX3030, por exemplo, para comunicação MODBUS TCP com um sistema SCADA, e comunicação MODBUS RTU e/ou MODBUS TCP com dispositivos de campo inteligentes ou sistemas de E/S remotas MODBUS.

Em configurações mais típicas, no entanto, são inseridos módulos adicionais nos half-clusters CPA e CPB, por exemplo, para prover um sistema de E/S remotas PROFIBUS e canais Ethernet adicionais.

Entre os módulos adicionais que, opcionalmente, podem ser inseridos nos half-clusters, pode-se citar:

- Mestres PROFIBUS NX5001
- Interfaces Ethernet NX5000

Caso seja necessário, pode-se utilizar bastidores maiores, como NX9002 (16 posições) e NX9003 (24 posições). Deve-se observar que todos os módulos citados até o momento neste capítulo têm largura dupla (ocupam duas posições). Adicionalmente, também pode ser utilizado o painel PX2612, que permite executar algumas transições na máquina de estados redundante que, caso contrário, não seriam possíveis, além do desligamento automático de half-clusters em algumas situações de falha.

6.2.2.1. Adição de Módulos NX5001 para Redes PROFIBUS

É possível inserir até quatro módulos NX5001 para a utilização de redes PROFIBUS em um CP redundante. Cada uma destas redes pode ser simples ou redundante. Caso a rede PROFIBUS “n” (sendo “n” um número de 1 a 4) seja redundante, as duas redes que a compõem são nomeadas PROFIBUS “n” A e PROFIBUS “n” B. Caso a rede PROFIBUS “n” seja simples, a rede que a compõe é nomeada PROFIBUS “n” A.

Para criar uma rede PROFIBUS redundante, é necessário inserir dois módulos NX5001 em cada half-cluster. Para criar uma rede PROFIBUS simples, basta inserir um módulo NX5001 em cada half-cluster. Sendo assim, é possível configurar até quatro redes simples, ou duas redes redundantes, ou uma redundante e duas simples. Nos demais casos, menos do que quatro módulos NX5001 serão necessários em cada half-cluster. Mais informações sobre redes PROFIBUS são fornecidas na seção [Configurações de Redes PROFIBUS](#).

No exemplo da Figura 158, existe apenas uma rede PROFIBUS (PROFIBUS 1), e a mesma é redundante (PROFIBUS 1 A e PROFIBUS 1 B). Neste exemplo, portanto, foram inseridos dois módulos NX5001 em cada half-cluster.

6.2.2.2. Adição de Módulos NX5000 para Redes Ethernet

É possível inserir até seis módulos NX5000 em cada half-cluster, provendo seis canais Ethernet adicionais, além dos dois canais Ethernet já existentes na UCP NX3030.

Os canais Ethernet podem ser usados de forma individual, ou organizados em pares NIC Teaming. Pares NIC Teaming são utilizados para prover canais Ethernet redundantes e serão descritos com maiores detalhes na seção [Redes Ethernet Redundantes com NIC Teaming](#).

Uma aplicação típica para o módulo NX5000 pode ser construir uma HSDN (High Speed Deterministic Network), para comunicação entre diversos CPs redundantes. Através desta rede, diversos CPs redundantes podem trocar mensagens em uma rede totalmente segregada, para garantir determinismo e uma comunicação rápida. Além disso, configurando esta rede como redundante com pares NIC Teaming, pode-se obter excelente disponibilidade. Para construir uma HSDN redundante, deve-se inserir dois módulos NX5000 em cada half-cluster. A Figura 158 mostra um exemplo de HSDN redundante, utilizando dois módulos NX5000 em cada half-cluster.

Aplicações nas quais os módulos de entradas e saídas estejam conectados a redes Ethernet podem necessitar interfaces extras de módulos NX5000 para se conectar a estas redes. Nestes casos, a rede que conecta os módulos de entradas e saídas pode ser uma rede simples ou redundante. Além disso, as interfaces podem ser configuradas com a opção de gerar falha vital. Neste caso, uma falha na rede provocará um switch-over.

A Figura 158 também mostra um exemplo com um módulo NX5000 usado de forma isolada (sem redundância NIC Teaming), inserido à direita dos demais módulos em cada bastidor.

6.2.3. Módulo NX4010

O módulo NX4010, como mostra a figura abaixo, foi concebido para prover a interconexão entre os dois half-clusters CPA e CPB, e também para conectar estes half-clusters ao painel de controle de redundância PX2612. Maiores detalhes sobre as conexões destes módulos serão fornecidos na seção [Interconexões entre Half-Clusters e Painel de Controle de Redundância PX2612](#).



Figura 160: NX4010

6.2.3.1. Características NX4010

Suas principais características são:

- Sincronismo de dados e aplicação entre dois half-clusters
- Interface de comunicação redundante entre dois half-clusters
- Switch-over automático (troca do half-cluster ativo) em caso de time-out de comunicação entre NX4010 e a sua respectiva UCP
- Possibilidade de desligar o outro half-cluster
- One Touch Diag
- Electronic Tag on Display
- Visor e LEDs para indicação de diagnóstico

Demais características (gerais, elétricas, mecânicas e de ambiente) estão presentes nas Características Técnicas do Módulo de Redundância NX4010 – CT114900.

6.2.4. Painel de Controle de Redundância PX2612

O painel de controle PX2612 é um item opcional em um sistema redundante. Ele deve ser utilizado quando a opção de redundância com painel for selecionada na criação de projeto usando o assistente. A Figura 161 mostra uma fotografia do painel de controle de redundância PX2612, enquanto a Figura 162 mostra com mais detalhes seu painel frontal.

Através do conector DB9 denominado CONTROL PLC A, é feita a conexão ao conector CONTROL do NX4010 do CPA, utilizando o cabo AL-2317/A.

Através do conector DB9 denominado CONTROL PLC B, é feita a conexão ao conector CONTROL do NX4010 do CPB, utilizando o cabo AL-2317/B.

Além disso, existe um conector com cinco bornes machos:

- GND: borne para conexão de aterramento.
- RL A: 2 bornes que correspondem aos terminais de um relé NA (normalmente aberto), que pode ser comandado pelo CPB para desligar o CPA. Este relé deve ser fechado pelo CPB para desligar o CPA.
- RL B: 2 bornes que correspondem aos terminais de um relé NA (normalmente aberto), que pode ser comandado pelo CPA para desligar o CPB. Este relé deve ser fechado pelo CPA para desligar o CPB.

Um CP (CPA ou CPB) poderá desligar o outro CP (CPB ou CPA) em algumas situações excepcionais, utilizando, então, os relés NA nos conectores RL A e RL B. Tais situações são descritas na seção [Transições entre Estados de Redundância](#).

O PX2612 também dispõe de seis botões para comandos de redundância, e seis LEDs utilizados para indicações de estado de redundância. Cada CP lê três destes botões e controla três destes LEDs.

Maiores informações sobre as funções destes botões e LEDs serão fornecidas na seção [Funções do Painel de Comando de Redundância PX2612](#).



Figura 161: Painel de Controle de Redundância PX2612

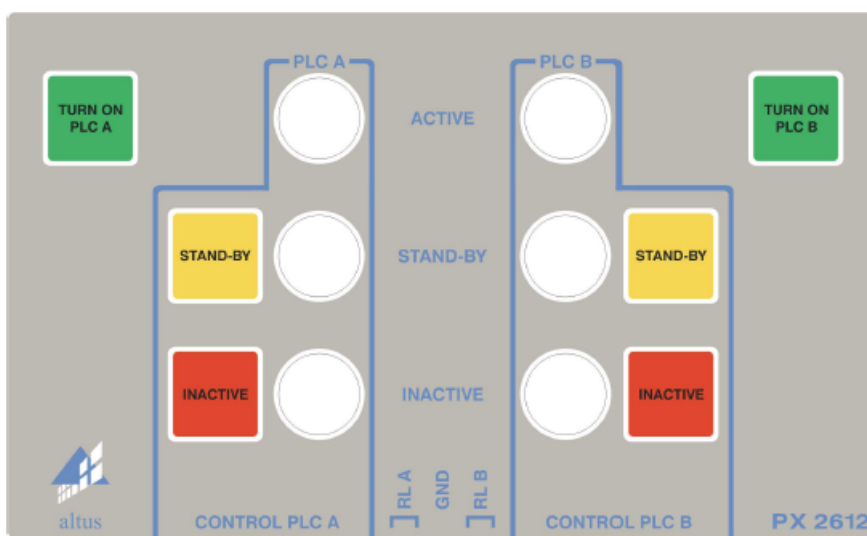


Figura 162: Frontal do Painel de Controle de Redundância PX2612

6.2.4.1. Características PX2612

O painel de controle de redundância PX2612 possui as seguintes características:

- CONTROL PLC A: conexão ao módulo NX4010 do CPA
- CONTROL PLC B: conexão ao módulo NX4010 do CPB
- RL A: terminais de um relé NA utilizado para desligar o CPA
- RL B: terminais de um relé NA utilizado para desligar o CPB
- GND: aterramento

Demais características (gerais, elétricas, mecânicas e de ambiente) estão presentes nas Características Técnicas do Painel de Controle de Redundância PX2612 – CT112500.

6.2.5. Interconexões entre Half-Clusters e Painel de Controle de Redundância PX2612

A figura abaixo mostra como devem ser feitas as conexões entre CPA, CPB e PX2612, inclusive para possibilitar que um CP desligue o outro, o que é necessário em situações excepcionais.

Dois cabos AL-2319 devem ser utilizados para os canais de sincronismo de redundância NETA e NETB. Um destes cabos interconecta os conectores NET 1 dos NX4010 de cada CP (canal NETA). O outro cabo interconecta os conectores NET 2 dos NX4010 de cada CP (canal NETB).

Um cabo AL-2317/A interconecta o conector CONTROL do NX4010 do CPA ao conector CONTROL PLC A do PX2612.

Um cabo AL-2317/B interconecta o conector CONTROL do NX4010 do CPB ao conector CONTROL PLC B do PX2612.

Além disso, é necessário fazer um circuito especial de alimentação, para que um CP seja capaz de desligar o outro CP em casos extremos.

Para maior confiabilidade, deve-se utilizar duas fontes separadas de 24V, uma para energizar o CPA, outra para energizar o CPB.

Observa-se que é necessário utilizar dois relés externos do tipo normalmente fechado (NF), com capacidade de corrente suficiente para energizar o NX8000. Estes relés devem ser dimensionados para uma corrente nominal da ordem de 2 A. Diodos em paralelo com as bobinas dos relés NF devem ser utilizados para proteger os contatos dos relés NA do PX2612.

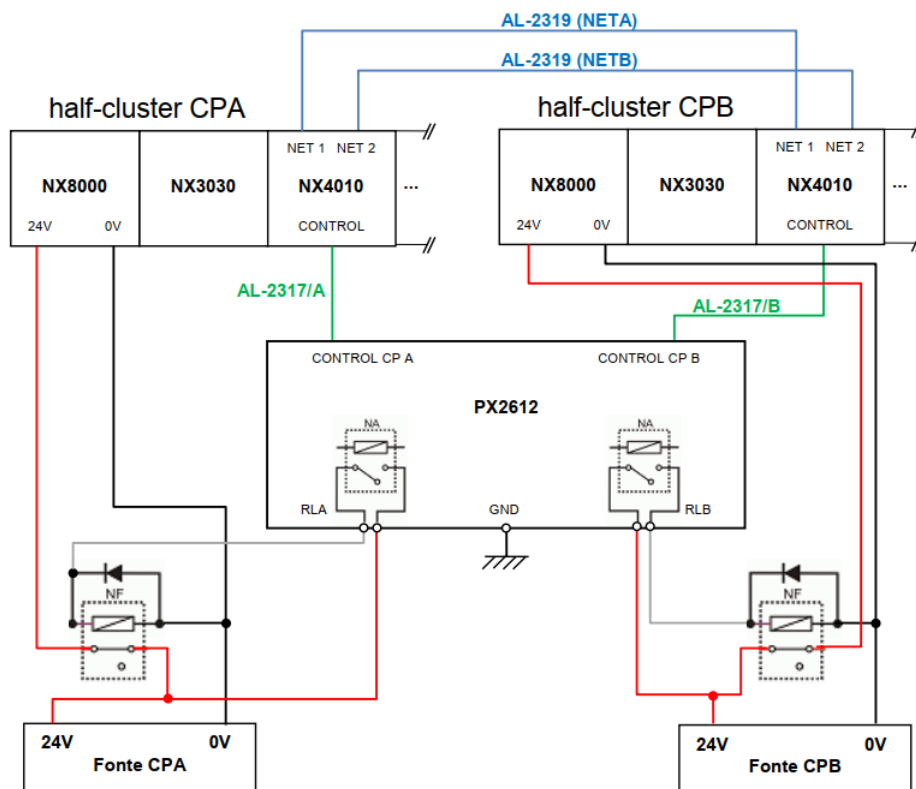


Figura 163: Interconexões entre CPA, CPB e PX2612

6.2.6. Características Gerais de um CP Redundante

	Características Gerais de um CP redundante
UCPs permitidas	NX3030
Tipo de redundância	Hot-standby
Tolerância a falhas	Tolera, no mínimo, falha simples em equipamentos duplicados nos half-clusters. Em casos específicos, pode tolerar múltiplas falhas.
5 estados de redundância de um half-cluster	<ul style="list-style-type: none"> - Não-Configurado: estado inicial, também considerado quando CP está desligado ou não está executando a tarefa principal (MainTask). - Inicializando: estado temporário assumido depois de Não-Configurado, onde alguns testes determinarão o próximo estado (Inativo, Ativo, Reserva – ou de volta para Não-Configurado). - Inativo: estado atingido depois de determinados tipos de falhas ou para manutenção programada. - Ativo: controlando o processo do usuário. - Reserva: apto para chavear para Ativo e controlar o processo do usuário, caso haja uma demanda (ex: falha do CP Ativo).
Principais falhas que causam switch-over entre CP Ativo e CP Reserva. CP Reserva chaveia para Ativo e o Ativo pode ir para Inativo ou Não-Configurado.	<ul style="list-style-type: none"> - Falta de alimentação. - Fonte. - UCP (parada de execução da tarefa principal - MainTask). - NX4010. - Falha de ambos os canais de sincronismo (NETA e NETB) sem que a causa seja interna ao CP Reserva. Neste caso, o CP Reserva, além de assumir como Ativo, desliga o outro CP. - Falha de ambos os canais de sincronismo (NETA e NETB) sendo a causa interna ao CP Ativo. - Falha em alguma rede PROFIBUS configurada como vital. - Falha em alguma rede Ethernet configurada como vital.
Comandos que causam switch-over entre CP Ativo e CP Reserva	<ul style="list-style-type: none"> - Comandos via painel de controle de redundância (PX2612). - Comandos recebidos do MasterTool ou de um sistema SCADA, através deste CP (local) ou do outro CP (remoto). - Comandos gerados pela aplicação do usuário, por exemplo, em função de outros diagnósticos (ex: falha de comunicação Ethernet), através deste CP (local) ou do outro CP (remoto).

	Características Gerais de um CP redundante
Principais falhas que impedem um CP de ficar no estado Reserva, ou assumir o estado Reserva. Tais falhas fazem com que este CP assumira os estados Não-Configurado ou Inativo.	<ul style="list-style-type: none"> - Falta de alimentação. - Fonte. - UCP (parada de execução da tarefa principal - MainTask). - NX4010. - Falha de ambos os canais de sincronismo (NETA e NETB) onde a causa é interna ao CP Reserva. - Falha no serviço de sincronismo de dados redundantes. - Falha no serviço de sincronismo da lista de forçamentos redundantes. - Falha total em alguma rede PROFIBUS configurada como vital. - Falha total em alguma rede Ethernet configurada como vital. - Projeto diferente daquele do CP Ativo, com sincronização automática de projetos habilitada. - Versão de firmware incompatível com CP Ativo. - Falha de Interface Ethernet configurada como falha vital.
Comandos que tiram o CP do estado Reserva	<ul style="list-style-type: none"> - Comandos via painel de controle de redundância (PX2612). - Comandos recebidos do MasterTool ou de um sistema SCADA, através deste CP (local) ou do outro CP (remoto). - Comandos gerados pela aplicação do usuário, por exemplo, em função de outros diagnósticos (ex: falha de comunicação Ethernet), através deste CP (local) ou do outro CP (remoto).
Tempo de switch-over	<ul style="list-style-type: none"> - Tipicamente de um a três ciclos da MainTask, dependendo do estímulo para troca de estado (comando ou falha). - Em caso de falha na rede PROFIBUS simples, dois ciclos da MainTask + 500 ms.
Switch-over sem descontinuidades (bump-less)	<ul style="list-style-type: none"> - Um switch-over não provoca descontinuidades nas saídas do controlador, nem em variáveis internas.
Overhead da redundância (consumo de UCP a cada ciclo da MainTask adicionado pela redundância)	<ul style="list-style-type: none"> - Valor máximo calculado automaticamente pelo MasterTool e informado ao usuário, considerando uma lista de forçamentos redundantes vazia. - Valor médio típico de 60 ms para 224 Kbytes de dados redundantes, em um sistema com uma rede PROFIBUS redundante e duas redes Ethernet HSDN redundantes
Visor da UCP	<p>Entre outros diagnósticos, mostra o estado da redundância (Ativo, Reserva, Inativo, Não-Configurado, Inicializando). Para a identificação do CP, é necessário verificar no menu Redundancia, onde há a identificação do CPA ou CPB.</p>
Painel de controle de redundância PX2612	<ul style="list-style-type: none"> - Este item é opcional. - Através de botões, permite comandos de switch-over ou transições entre estados de redundância para manutenção. - LEDs sinalizam o estado da redundância em cada half-cluster. - Relé NA permite que um half-cluster desligue o outro em casos extremos. Botão permite o religamento do outro half-cluster. - Recursos embutidos para teste dos botões, LEDs e relés. - Um projeto criado com painel não pode ser convertido em um projeto sem painel, e vice-versa.

	Características Gerais de um CP redundante
Diagnósticos da redundância	<ul style="list-style-type: none"> - Indicam falhas tanto no CPA como no CPB, independente dos seus estados (Ativo ou Não-Ativo). - Evitam “falhas ocultas”. - Permitem manutenção rápida, o que é essencial para obter alta disponibilidade.
Comandos da redundância	<ul style="list-style-type: none"> - Permitem executar as mesmas ações do painel de controle PX2612, além de outros comandos (ex: comandar switch-overs). - Podem ser executados no CP local, ou repassados para o outro CP (remoto) via canais de sincronismo NETA / NETB. - Podem ser recebidos a partir do MasterTool ou de um sistema SCADA. - Podem ser executados a partir da aplicação do usuário.
Eventos da redundância	Registram modificações nos diagnósticos e comandos de redundância, com estampa de tempo, permitindo assim uma investigação das causas de um switch-over.
SNTP (Simple Network Time Protocol)	Permite que os eventos tenham estampa de tempo precisa e ajustada à hora mundial. Também sincroniza o relógio de tempo real do CP para outras aplicações.
Sincronização de comandos e diagnósticos	A cada ciclo da MainTask, CPA e CPB trocam diagnósticos e comandos através dos canais de sincronismo NETA ou NETB. Desta maneira, um CP conhece os diagnósticos e comandos do outro.
Sincronização de dados redundantes	A cada ciclo da MainTask, o CP Ativo copia dados redundantes para o CP Não-Ativo através dos canais de sincronismo NETA e NETB. Dados não-redundantes não são sincronizados.
Sincronização da lista de forçamentos redundantes	A cada ciclo da MainTask, o CP Ativo copia a lista de forçamentos redundantes para o CP Não-Ativo através dos canais de sincronismo NETA e NETB. Esta lista inclui apenas variáveis redundantes forçadas. Desta maneira, CPA e CPB poderão ter diferentes conjuntos de dados não-redundantes forçados, pois estes forçamentos não são sincronizados.
Projeto único para CPA e CPB	Existe um projeto único comum para o CPA e CPB, gerado pelo MasterTool. O projeto é composto do projeto de aplicação (código executável) e do <i>project archive</i> (código fonte).
Identificação de um CP	Através do MasterTool, identifica-se uma UCP NX3030 como CPA, CPB ou CP não-redundante. Esta identificação não faz parte do projeto de aplicação gerado pelo MasterTool, embora seja escrita em um CP usando o MasterTool. A identificação de cada CP viabiliza a característica de ter um projeto único para CPA e CPB.
Sincronização automática do projeto	Se o projeto do CP Ativo se tornar diferente do projeto do CP Não-Ativo, o mesmo é copiado do CP Ativo para o CP Não-Ativo. Esta sincronização pode durar diversos ciclos da MainTask. Deve-se lembrar que o projeto é composto do projeto de aplicação (código executável) e do código fonte (<i>project archive</i>), sendo ambos sincronizados. Esta sincronização pode ser desabilitada em circunstâncias especiais, para viabilizar modificações de projeto que só poderiam ser carregadas offline em CPs não-redundantes.

	Características Gerais de um CP redundante
Expansão a quente de módulos e remotas PROFIBUS	Existem modificações de projeto que não podem ser feitas à quente (online) em um CP não-redundante, tais como a inclusão de novos módulos ou remotas PROFIBUS. No entanto, valendo-se da redundância do CP e das redes PROFIBUS, foi definido um procedimento para atingir este objetivo, muito importante para sistemas que necessitam de alta disponibilidade.
Endereços IP privados para CPA e CPB	É possível conectar-se a um CP específico (CPA ou CPB) utilizando um endereço IP privado para o mesmo, para, por exemplo, obter diagnósticos específicos desse half-cluster. O Endereço de IP do CPA estará sempre associado à interface NET(i) do CPA, enquanto o Endereço de IP do CPB estará sempre associado à interface NET(i) do CPB.
IP Ativo	Nome de uma estratégia que permite que clientes Ethernet se conectem a um servidor no CP redundante, utilizando sempre o mesmo endereço de IP. Isto evita a necessidade de complexos scripts para mudar o endereço IP quando ocorrem switch-overs devido à redundância. O Endereço de IP Ativo estará sempre associado à interface NET(i) do CP Ativo.
NIC Teaming	Nome de uma estratégia que permite que duas interfaces Ethernet de um half-cluster formem um par redundante compartilhando um mesmo endereço IP. Desta forma, pode-se construir redes Ethernet redundantes com maior facilidade, sem que os clientes conectados a um par NIC Teaming tenham de implementar complexos scripts para chavear endereços IP.
Redes PROFIBUS e Configuração de Falhas Vitais	O CP suporta até quatro redes PROFIBUS simples ou até duas redes PROFIBUS redundantes. Também é possível configurar se a falha de cada rede PROFIBUS é considerada vital (causa switch-over) ou não.
Redes Ethernet e Configuração de Falhas Vitais	O CP suporta até 8 redes Ethernet simples ou até quatro redes Ethernet redundantes (se consideradas as interfaces frontais da NX3030). Também é possível configurar se a falha de cada rede Ethernet é considerada vital (causa switch-over) ou não.
Tarefa de usuário única e cíclica	Somente uma tarefa de usuário é permitida, sendo denominada MainTask. Esta tarefa é cíclica.
POUs principais: NonSkippedPrg e ActivePrg	Na criação de um projeto redundante, o MasterTool gera automaticamente duas POU's do tipo programa vazias que devem ser preenchidas pelo usuário: - NonSkippedPrg: esta POU é executada nos dois CPs (CPA e CPB), independente do estado de redundância (Ativo ou Não-Ativo). Destina-se ao gerenciamento de diagnósticos e comandos especiais. - ActivePrg: esta POU é executada somente no CP Ativo, e destina-se ao controle do processo do usuário final.

Tabela 201: Características Gerais de um CP Redundante

6.2.7. Dados para Compra

A configuração mínima de um CP redundante implica na aquisição dos seguintes módulos:

- Dois bastidores, que devem ser escolhidos entre quatro modelos disponíveis de acordo com os módulos que devem ser

instalados:

- NX9000: oito posições (quatro módulos duplos)
 - NX9001: doze posições (seis módulos duplos)
 - NX9002: dezesseis posições (oito módulos duplos)
 - NX9003: vinte e quatro posições (doze módulos duplos)
- Dois NX8000
 - Dois NX3030
 - Dois NX4010
 - Dois AL-2319

Além disso, pode ser necessário adquirir os seguintes módulos opcionais:

- Um PX2612
- Um AL-2317/A
- Um AL-2317/B
- Dois módulos NX5001 para cada rede PROFIBUS simples adicional
- Quatro módulos NX5001 para cada rede PROFIBUS redundante adicional
- Dois módulos NX5000 para cada rede Ethernet simples adicional
- Quatro módulos NX5000 para cada rede Ethernet redundante adicional (NIC Teaming)

ATENÇÃO

Pode-se instalar até quatro módulos PROFIBUS em cada CP redundante. Isto significa que podemos configurar até quatro redes PROFIBUS simples ou duas redes PROFIBUS redundantes.

6.3. Princípios de Funcionamento

Nesta seção, descrevem-se as funções de um CP redundante com UCP NX3030, seu comportamento e estados. Também são apresentados conceitos e restrições de programação e configuração que serão utilizados nas próximas seções.

6.3.1. Identificação de uma UCP NX3030

Uma UCP NX3030 possui um registro de identificação, não volátil, onde é possível identificá-la como:

- Não-Redundante: não pode ser utilizada em um CP redundante (configuração padrão de fábrica)
- CPA: utilizada no CPA de um CP redundante
- CPB: utilizada no CPB de um CP redundante

O registro de identificação pode ser ajustado com o programador MasterTool. A primeira coisa a ser feita em uma UCP NX3030, antes de carregar o projeto do CP redundante na mesma, é identificá-la como CPA ou CPB. Caso a identificação não seja realizada, muitas características da redundância não vão funcionar corretamente, como, por exemplo, a sincronização da aplicação entre os CPs.

ATENÇÃO

O registro de Identificação do CP não faz parte do projeto do CP redundante, e portanto não é salvo como parte deste projeto no computador onde o MasterTool IEC XE é executado. O registro é salvo apenas na memória não volátil da UCP.

6.3.2. Projeto Redundante Único

Graças ao registro de identificação descrito anteriormente, existe um único projeto para o CP redundante, idêntico para os CPA e CPB.

Parâmetros de configuração que devem ser diferentes entre CPA e CPB (ex: endereços IP de interfaces Ethernet) aparecem duplicados no projeto do CP redundante (um para CPA, outro para CPB). Cada CP irá considerar aquele que lhe corresponde, após analisar seu registro de identificação.

6.3.3. Estrutura do Projeto Redundante

6.3.3.1. Redundancy Template

O projeto de um CP redundante é criado automaticamente a partir de um modelo, denominado Redundancy Template.

O modelo parte da configuração mínima de um CP redundante, conforme definido na seção [Configuração Mínima de um CP Redundante \(Sem utilização do Pannel PX2612\)](#). Além disso, alguns diálogos com o usuário são realizados para inserção de módulos adicionais nos half-clusters, tais como mestres PROFIBUS (NX5001) e módulos Ethernet (NX5000).

Remotas PROFIBUS devem ser inseridas pelo usuário, abaixo dos mestres PROFIBUS NX5001 já inseridos.

Além disso, são criadas tarefas e POU's básicas do tipo programa, conforme descrito nas próximas seções.

6.3.3.2. Tarefa Única e Cíclica MainTask

O projeto de um CP redundante possui uma única tarefa, denominada MainTask, que é cíclica. O usuário pode ajustar o intervalo da tarefa.

6.3.3.3. Programa MainPrg

A tarefa MainTask está associada a uma única POU do tipo programa, denominada MainPrg. O programa MainPrg é criado automaticamente.

O código do programa MainPrg é o seguinte, em linguagem ST:

```
SpecialVariablesPrg();
IF isFirstCycle THEN
  StartPrg();
  isFirstCycle := FALSE;
ELSE
  fbRedundancyManagement();
  IF fbRedundancyManagement.m_fbDiagnosticsLocal.eRedState = REDUNDANCY_STATE.
  ACTIVE THEN
    SpecialVariablesRedundantPrg();
  END_IF;
  NonSkippedPrg();

  IF fbRedundancyManagement.m_fbDiagnosticsLocal.eRedState = REDUNDANCY_STATE.
  ACTIVE THEN
    ActivePrg();
  END_IF;
END_IF;
```

MainPrg chama outras duas POU's do tipo programa, denominadas NonSkippedPrg e ActivePrg. NonSkippedPrg sempre é chamada, pois é executada tanto no CP Ativo quanto no Não-Ativo. Já a POU ActivePrg só é chamada quando a condição "RedDgnLoc.sGeneral.Diag.eRedState = Active" é verdadeira, ou seja, quando o CP se encontra em estado Ativo.

Portanto, o programa NonSkippedPrg será sempre executado em ambos os CPs (CPA e CPB), independente do estado de redundância deste CP. Por outro lado, o programa ActivePrg será executado somente no CP que se encontra em estado Ativo.

Ao contrário do programa MainPrg, que não deve ser modificado, o usuário pode modificar os programas NonSkippedPrg e ActivePrg. Inicialmente, quando o projeto redundante é criado a partir do Redundancy Template, estes dois programas são criados "vazios", mas o usuário poderá inserir código nos mesmos.

ATENÇÃO

Quando a opção de comunicação OPC DA for habilitada durante a criação do projeto, o programa NonSkippedPrg não será criado vazio. Para maiores informações, consultar a seção [Uso de Comunicação OPC DA com Projetos Redundantes](#).

6.3.3.4. Programa ActivePrg

O principal objetivo deste programa, que é executado somente no CP Ativo, é controlar o processo do usuário final.

Este programa normalmente atua sobre as variáveis redundantes, entre as quais encontram-se as variáveis de representação direta %I e %Q associadas ao sistema de E/S remotas. Para maiores informações, consultar a seção [Programação de um CP Redundante - Configurações da MainTask - Programa ActivePrg](#).

ATENÇÃO

Havendo sucesso ou não na compilação, o MasterTool informa se a folga é respeitada e o cálculo do overhead de redundância na janela de mensagens.

6.3.3.5. Programa NonSkippedPrg

Este programa, que é executado em ambos os CPs (CPA e CPB) independente do estado de redundância, é tipicamente utilizado para funções como:

- Organizar diagnósticos não-redundantes para reportar a um sistema SCADA.
- Receber e executar comandos não-redundantes a partir de um sistema SCADA.
- Gerenciar condições de switch-over normalmente não contempladas automaticamente pelo CP redundante, que podem variar de usuário para usuário. Por exemplo, determinado usuário poderá executar um switch-over para o CP Reserva se o CP Ativo não estiver se comunicando com o sistema SCADA, enquanto outro usuário pode não desejar um switch-over nesta situação.
- Habilitar ou desabilitar drivers de E/S em função do estado de redundância, por exemplo, desabilitar um mestre MODBUS RS-485 no CP Não-Ativo.
- Detectar falhas em drivers de E/S em um CP Não-Ativo para evitar falhas ocultas. Alguns drivers de E/S não contemplam detecção automática de tais falhas. Outros drivers de E/S, como o PROFIBUS, o fazem automaticamente.
- Outras atividades que, por algum motivo, precisam ser executadas tanto no CP Ativo como no CP Reserva.

Para maiores informações, consultar a seção [Programação de um CP Redundante - Configurações da MainTask - Programa NonSkippedPrg](#).

6.3.3.6. Variáveis Redundantes e Não-Redundantes

As variáveis de um CP redundante podem ser classificadas entre redundantes e não-redundantes. Variáveis redundantes são copiadas do CP Ativo para o CP Não-Ativo, no início de cada ciclo da MainTask, através dos canais de sincronismo NETA e NETB. Por outro lado, variáveis não-redundantes não são copiadas entre half-clusters e, portanto, podem ter valores diferentes nos CPs A e B.

As variáveis não-redundantes são utilizadas para armazenar informações privativas de cada half-cluster (CPA ou CPB), tais como diagnósticos de um módulo dentro deste half-cluster, incluindo os diagnósticos da redundância (estado da redundância deste half-cluster, etc...).

As variáveis redundantes dizem respeito às informações compartilhadas e relativas ao controle do processo. As variáveis associadas aos módulos de E/S são exemplos típicos de variáveis redundantes.

6.3.3.7. Variáveis %I Redundantes e Não-Redundantes

A UCP NX3030 aloca 96 Kbytes de variáveis %I (%IB0 ... %IB98303).

Os primeiros 80 Kbytes podem ser redundantes (%IB0 ... %IB81919). Os últimos 16 kbytes sempre são não-redundantes (%IB81920 ... %IB98303).

A área de 80 Kbytes que pode ser redundante tipicamente é alocada para entradas, que podem ser lidas a partir de um sistema de E/S remotas (PROFIBUS, MODBUS, etc...).

A área de 16 Kbytes não-redundante é alocada para diagnósticos privados rápidos de um half-cluster e, também, para os botões do painel de comando de redundância PX2612. Diagnósticos rápidos são aqueles que, necessariamente, devem ser atualizados a cada ciclo da MainTask.

O usuário pode configurar a quantidade de variáveis %I redundantes, entre 0 bytes e 81920 bytes, em múltiplos de 1 byte (o valor padrão é 16384 bytes - %IB0 ... %IB16383). A configuração adequada da quantidade de %I redundantes a partir de %IB0 é importante para reduzir o tempo necessário para sincronizar variáveis redundantes (reduzir o overhead da redundância). Por exemplo, se a aplicação real aloca apenas %IB0 ... %IB1499 para entradas redundantes, pode-se definir o tamanho da área redundante de %I em 1500 bytes.

A figura seguinte ilustra a alocação de variáveis de representação direta %I redundantes e não-redundantes, onde RI é a quantidade de %I configuradas como redundantes de fato.

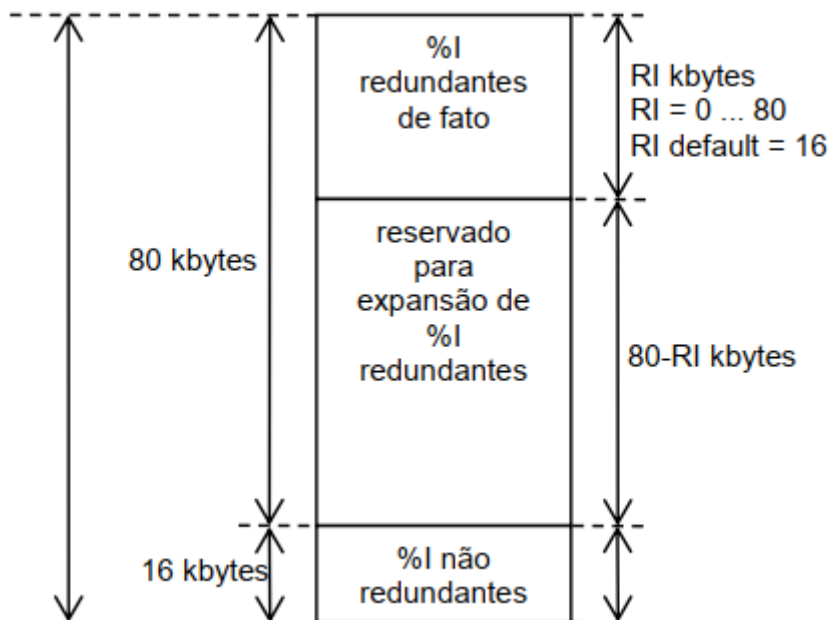


Figura 164: Alocação de %I Redundantes e Não-Redundantes

6.3.3.8. Variáveis %Q Redundantes e Não-Redundantes

A UCP NX3030 aloca 96 Kbytes de variáveis %Q (%QB0 ... %QB98303). Os primeiros 80 Kbytes podem ser redundantes (%QB0 ... %QB81919). Os últimos 16 Kbytes sempre são não-redundantes (%QB81920 ... %QB98303).

A área de 80 Kbytes que pode ser redundante é dividida em duas seções:

- Os primeiros bytes são reservados para saídas que podem ser redundantes, e tipicamente alocadas para um sistema de E/S remotas (PROFIBUS, MODBUS, etc...). O tamanho dessa seção é configurável, e seu valor padrão é 16384. Ela engloba %QB0 ... %QB16383, podendo ser configurada para até 64 Kbytes (%QB0 ... %QB65535).
- Os próximos bytes são reservados para diagnósticos que podem ser redundantes, por exemplo, do sistema de E/S (diagnósticos de módulos de E/S, de interfaces de comunicação escravos PROFIBUS, etc...). Ao contrário dos diagnósticos rápidos (alocados em %I), tais diagnósticos alocados em %Q podem levar mais do que um ciclo da MainTask para serem atualizados. Por padrão, esta seção engloba 16 Kbytes (%QB65536 ... %QB81919).

A área não-redundante (%QB81920 ... %QB98303) tipicamente é alocada para diagnósticos e comandos privados de um half-cluster, e também para os LEDs e relé do painel de comando de redundância PX2612.

O usuário pode reduzir a quantidade de variáveis %Q redundantes em cada uma das duas seções que podem ser redundantes:

- Na primeira seção, o tamanho da área redundante pode ser configurado entre 0 bytes e 65535 bytes, em múltiplos de 1 byte (o valor padrão é 16384 bytes). A configuração adequada da quantidade de %Q redundantes nesta seção é importante para reduzir o tempo necessário para sincronizar variáveis redundantes (reduzir o overhead da redundância). Por exemplo, se a aplicação real aloca apenas %Q0 ... %Q1499 para saídas redundantes, pode-se definir o tamanho desta seção redundante de %Q em 1500 bytes.
- Na segunda seção, o tamanho da área redundante pode ser configurado entre 0 bytes e 81919 bytes, em múltiplos de 1 byte (o valor padrão é 16384 bytes). A configuração adequada da quantidade de %Q redundantes nesta seção é importante para reduzir o tempo necessário para sincronizar variáveis redundantes (reduzir o overhead da redundância). Por exemplo, se a aplicação real aloca apenas %QB65536 ... %QB66999 para diagnósticos redundantes, pode-se definir o tamanho desta seção redundante de %Q em 1464 bytes.

A figura seguinte ilustra a alocação de variáveis de representação direta %Q redundantes e não-redundantes, onde RQS é a quantidade de saídas %Q configuradas como redundantes na primeira seção, e RQD é a quantidade de diagnósticos %Q configurados como redundantes na segunda seção.

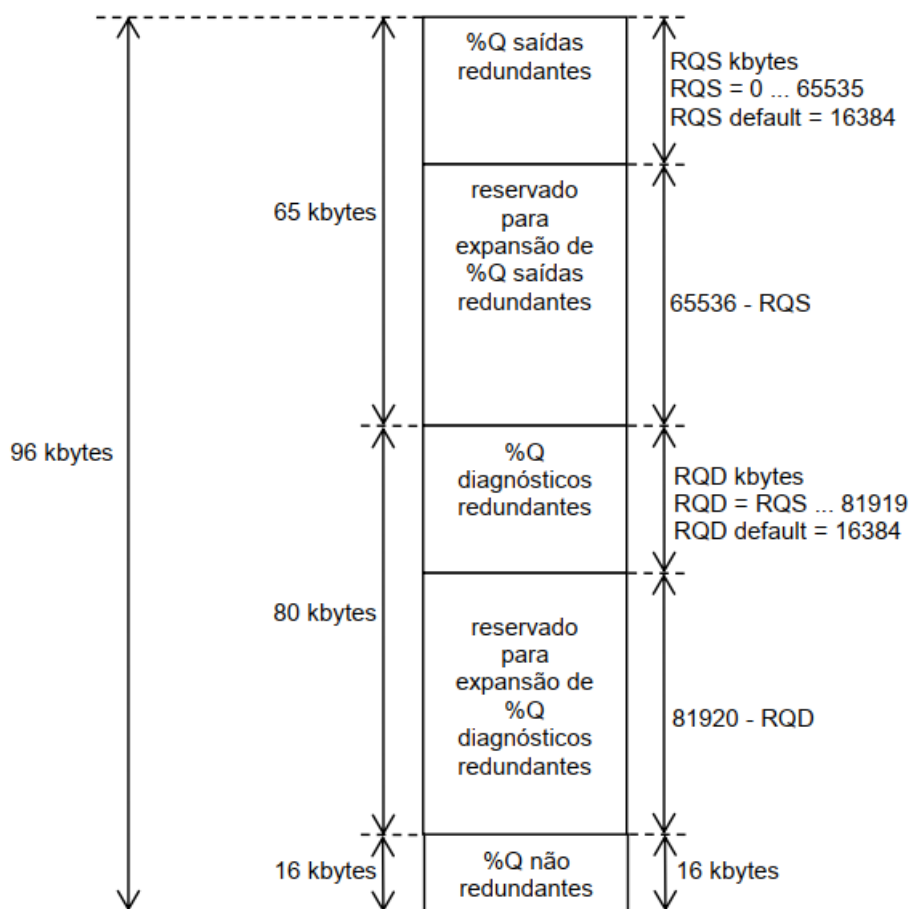


Figura 165: Alocação de %Q Redundantes e Não-Redundantes

6.3.3.9. Variáveis %M Redundantes e Não-Redundantes

A UCP NX3030 aloca 64 Kbytes de variáveis %M (%MB0 ... %MB65535).

Todos os 65536 bytes podem ser redundantes (%MB0000 ... %MB65535). Por padrão, a quantidade de variáveis %M redundantes é 0.

O uso de variáveis %M redundantes deve ser evitado, preferindo-se a utilização de variáveis simbólicas (ver seção [Variáveis Simbólicas Redundantes e Não-Redundantes](#)).

6.3.3.10. Variáveis Simbólicas Redundantes e Não-Redundantes

Além das variáveis de representação direta (%I, %Q e %M) que são alocadas automaticamente, o usuário pode declarar explicitamente variáveis simbólicas, dentro de POU's ou GVL's. O tamanho máximo permitido para alocação de variáveis simbólicas redundantes é de 512 Kbytes.

ATENÇÃO

Não se deve confundir variáveis simbólicas com variáveis simbólicas endereçadas através da diretiva AT. Variáveis simbólicas que utilizam a diretiva AT são nomes simbólicos atribuídos a variáveis de representação direta (%I, %Q e %M). Portanto, variáveis que utilizam a diretiva AT não alocam nenhuma memória de variáveis simbólicas.

Variáveis simbólicas são redundantes nos seguintes casos:

- Quando declaradas em POU's do tipo "programa" criadas na aplicação do usuário, com exceção do programa NonSkipPrg
- Quando declaradas em GVL's criadas na aplicação do usuário e estas GVL's marcadas como redundantes

Variáveis simbólicas são não-redundantes nos seguintes casos:

- Quando declaradas no programa NonSkippedPrg, já descrito anteriormente na seção [Programa NonSkippedPrg](#)
- Quando declaradas em POU's do tipo “função”. Observar que tais tipos de POU's normalmente deveriam alocar variáveis apenas na pilha (não estáticas), que conseqüentemente não precisam ser redundantes. Mesmo sabendo que o usuário pode declarar variáveis estáticas (VAR STATIC) dentro de POU's do tipo “função”, isto é considerado uma má prática de programação. Tais variáveis estáticas, caso sejam criadas, serão consideradas não-redundantes
- Quando declaradas em POU's do tipo “bloco funcional”. Observar que a mera declaração de um “bloco funcional” não aloca memória (o que aloca memória é instanciar um Bloco Funcional)

Deve-se observar que instâncias de blocos funcionais, declaradas dentro de POU's do tipo programa ou dentro de GVL's, comportam-se como variáveis simbólicas, ou seja, alocam memória redundante. Da mesma maneira que variáveis simbólicas, quando instâncias de bloco funcional são declaradas no programa NonSkippedPrg ou quando a GVL não é marcada como redundante, tais instâncias são não-redundantes.

6.3.4. Mapeamentos Múltiplos

Se o usuário desejar mapear as variáveis de comandos da redundância em mais de uma porta de comunicação (COMx ou NETx) será necessária a implementação de um controle pelo usuário na própria aplicação.

A lógica de controle a ser implementada deverá escrever nas variáveis de controle da redundância baseado nos valores das variáveis (comandos) provenientes de cada uma das portas de comunicação (COMx ou NETx). Além disso, a lógica de controle deve reinicializar as variáveis de comandos das portas de comunicação, uma vez que o controle da redundância só reinicializa suas próprias variáveis de comandos.

Segue um exemplo dessa implementação:

```

VAR
var_comando_StandBy_relacao_Ethernet : BOOL;
var_comando_StandBy_relacao_Serial : BOOL;
var_comando_Inactive_relacao_Ethernet : BOOL;
var_comando_Inactive_relacao_Serial : BOOL;
var_comando_TurnOn_relacao_Ethernet : BOOL;
var_comando_Turn_relacao_Serial : BOOL;
END_VAR

//Lógica para colocar o CP Local em Reserva
IF var_comando_StandBy_relacao_Ethernet = TRUE THEN
DG_NX4010.tRedundancy.RedCmdLoc.bStandbyLocal:=TRUE;
var_comando_StandBy_relacao_Ethernet:=FALSE;
END_IF
IF var_comando_StandBy_relacao_Serial = TRUE THEN
DG_NX4010.tRedundancy.RedCmdLoc.bStandbyLocal:=TRUE;
var_comando_StandBy_relacao_Serial:=FALSE;
END_IF
//Lógica para colocar o CP Local em Inativo
IF var_comando_Inactive_relacao_Ethernet = TRUE THEN
DG_NX4010.tRedundancy.RedCmdLoc.bInactiveLocal:=TRUE;
var_comando_Inactive_relacao_Ethernet:=FALSE;
END_IF
IF var_comando_Inactive_relacao_Serial = TRUE THEN
DG_NX4010.tRedundancy.RedCmdLoc.bInactiveLocal:=TRUE;
var_comando_Inactive_relacao_Serial:=FALSE;
END_IF
//Lógica para religar o CP Local desligado pelo PX2612
IF var_comando_TurnOn_relacao_Ethernet = TRUE THEN
DG_NX4010.tRedundancy.RedCmdLoc.bTurnOnLocal:=TRUE;
var_comando_TurnOn_relacao_Ethernet:=FALSE;
END_IF
IF var_comando_Turn_relacao_Serial = TRUE THEN

```

```
DG_NX4010.tRedundancy.RedCmdLoc.bTurnOnLocal:=TRUE;  
var_comando_Turn_relacao_Serial:=FALSE;  
END_IF
```

Acima temos um exemplo de uma lógica em linguagem ST, onde o comando da redundância pode ser feito por duas variáveis, que provêm de portas de comunicação diferentes. No exemplo, realizamos três comandos diferentes (StandBy, Inactive e TurnOn).

Onde:

var_comando_StandBy_relacao_Ethernet: variável do tipo Bool atribuída a um Coil da comunicação Ethernet que realizará o comando para colocar o Half-Cluster Local em Reserva.

var_comando_StandBy_relacao_Serial: variável do tipo Bool atribuída a um Coil da comunicação Serial que realizará o comando para colocar o Half-Cluster Local em Reserva.

DG_NX4010.tRedundancy.RedCmdLoc.bStandbyLocal: este comando produz uma ação equivalente ao botão STAND-BY do PX2612, no CP local.

var_comando_Inactive_relacao_Ethernet: variável do tipo Bool atribuída a um Coil da comunicação Ethernet que realizará o comando para colocar o Half-Cluster Local em Inativo.

var_comando_Inactive_relacao_Serial: variável do tipo Bool atribuída a um Coil da comunicação Serial que realizará o comando para colocar o Half-Cluster Local em Inativo.

DG_NX4010.tRedundancy.RedCmdLoc.bInactiveLocal: este comando produz uma ação equivalente ao botão INACTIVE do PX2612, no CP local.

var_comando_TurnOn_relacao_Ethernet: variável do tipo Bool atribuída a um Coil da comunicação Ethernet que realizará o comando para religar o Half-Cluster Local depois que o relé do PX2612 o desligou.

var_comando_Turn_relacao_Serial: variável do tipo Bool atribuída a um Coil da comunicação Serial que realizará o comando para religar o Half-Cluster Local depois que o relé do PX2612 o desligou.

DG_NX4010.tRedundancy.RedCmdLoc.bTurnOnLocal: este comando produz uma ação equivalente ao botão TURN ON PLC do PX2612, no CP local.

6.3.5. Estruturas de Dados de Diagnósticos, Comandos e Usuário

Cada CP dispõe de diversas estruturas de dados relacionadas com a redundância. As seguintes estruturas são variáveis simbólicas que utilizam a diretiva AT para mapear variáveis do tipo %Q:

- RedDgnLoc: contém diagnósticos deste CP (local) relacionados com a redundância, como por exemplo, o estado da redundância deste CP
- RedDgnRem: é uma cópia de RedDgnLoc do outro CP, recebida via canais de sincronismo NETA / NETB. Desta forma, este CP (local) tem acesso aos diagnósticos do outro CP (remoto)
- RedCmdLoc: contém comandos que devem ser aplicados neste CP (quando têm sufixo Local) ou no outro CP (quando têm sufixo Remote). Por exemplo o campo StandbyLocal desta estrutura de dados corresponde a um comando que deve ser executado neste CP (local), enquanto o campo StandbyRemote corresponde a um comando que deve ser executado no outro CP (remoto)
- RedCmdRem: é uma cópia de RedCmdLoc do outro CP, recebida via canais de sincronismo NETA / NETB. Desta forma, este CP pode executar comandos recebidos do outro CP que tenham o sufixo Remote
- RedUsrLoc: contém 128 bytes de dados preenchidos livremente pelo usuário (ex: diagnóstico de comunicação com um sistema SCADA). Estes 128 bytes de dados podem ser intercambiados com o outro CP (remoto)
- RedUsrRem: é uma cópia de RedUsrLoc do outro CP, recebida via canais de sincronismo NETA / NETB

Na seção [Manutenção da Redundância](#), as seguintes subseções fornecerão maiores detalhes sobre estas estruturas de dados:

- [Estrutura de Diagnósticos da Redundância](#)
- [Comandos da Redundância](#)
- [Informações do Usuário Trocadas entre CPA e CPB](#)

6.3.6. Serviços de Sincronização Cíclicos através de NETA e NETB

Esta seção descreve os três serviços de sincronização que ocorrem ciclicamente em um CP redundante, entre CPA e CPB, através dos canais de sincronismo NETA e NETB.

Estes serviços são executados no início de cada ciclo da MainTask, e na sequência em que aparecem abaixo, ou seja:

- Primeiro, executa-se o serviço Troca de Diagnósticos e Comandos
- Segundo, executa-se o serviço Sincronização de Dados Redundantes
- Terceiro, executa-se o serviço Sincronização da Lista de Forçamentos Redundantes

6.3.6.1. Troca de Diagnósticos e Comandos

Este serviço é responsável pelo intercâmbio das seguintes estruturas de dados, em cada ciclo da MainTask:

- Copiar RedDgnLoc do CPA para RedDgnRem do CPB
- Copiar RedCmdLoc do CPA para RedCmdRem do CPB
- Copiar RedUsrLoc do CPA para RedUsrRem do CPB
- Copiar RedDgnLoc do CPB para RedDgnRem do CPA
- Copiar RedCmdLoc do CPB para RedCmdRem do CPA
- Copiar RedUsrLoc do CPB para RedUsrRem do CPA

O serviço será executado utilizando apenas um dos canais de sincronismo (NETA ou NETB). Desta maneira, o serviço pode ser completado mesmo que um dos canais esteja com problemas.

6.3.6.2. Sincronização de Dados Redundantes

Este serviço é responsável pela transferência de variáveis redundantes, do CP Ativo para o CP Não-Ativo. Conforme visto anteriormente, existem variáveis redundantes simbólicas e também variáveis redundantes de representação direta (%I, %M e %Q).

Para que este serviço seja executado, diversas condições devem ser satisfeitas:

- O serviço de sincronização anterior deste ciclo da MainTask (Troca de Diagnósticos e Comando) deve ter completado com sucesso.
- Caso este CP esteja em estado Ativo, o outro deve estar em estado Não-Ativo. Por outro lado, caso este CP esteja em estado Não-Ativo, o outro deve estar em estado Ativo.
- Os projetos dos 2 CPs devem estar idênticos, exceto quando a sincronização automática de projetos estiver desabilitada (ver seção [Desabilitação da Sincronização de Projetos](#)).
- Necessário pelo menos um canal de sincronismo (NETA e/ou NETB) em estado operacional. Se ambos os canais estiverem em operação, a comunicação é distribuída entre ambos (balanceamento de carga) para reduzir o tempo da sincronização. Caso apenas um canal estiver em operação, o sincronismo continuará sendo realizado somente por este canal, garantindo assim a sincronização dos dados redundantes.

6.3.6.3. Sincronização da Lista de Forçamentos Redundantes

Este serviço é responsável pela transferência da lista de forçamentos redundantes do CP Ativo para o CP Não-Ativo.

Para que este serviço seja executado, diversas condições devem ser satisfeitas:

- Os dois serviços de sincronização anteriores deste ciclo (Troca de Diagnósticos e Comando, Sincronização de Dados Redundantes) devem ter completado com sucesso
- Caso este CP esteja em estado Ativo, o outro deve estar em estado Não-Ativo. Por outro lado, caso este CP esteja em estado Não-Ativo, o outro deve estar em estado Ativo
- Os projetos dos dois CPs devem estar idênticos, exceto quando a sincronização automática de projetos estiver desabilitada (ver seção [Desabilitação da Sincronização de Projetos](#))
- Necessário pelo menos um canal de sincronismo (NETA e/ou NETB) em estado operacional. Se ambos os canais estiverem em operação, a comunicação é distribuída entre ambos (balanceamento de carga) para reduzir o tempo da sincronização. Caso apenas um canal estiver em operação, o sincronismo continuará sendo realizado somente por este canal, garantindo assim a sincronização dos dados redundantes

ATENÇÃO

A lista de forçamentos redundantes contém apenas forçamentos sobre variáveis redundantes. Em cada um dos CPs (CPA e CPB), pode existir uma lista diferente de forçamentos sobre variáveis não-redundantes.

6.3.7. Serviços de Sincronização Esporádicos através de NETA e NETB

Os seguintes serviços de sincronização são executados de forma esporádica. Ou seja, não são executados a cada ciclo da tarefa MainTask e outra tarefa do sistema executa estes serviços esporádicos em segundo plano.

6.3.7.1. Sincronização de Projetos

Este serviço é responsável por manter sincronizados os projetos dos CPs Ativo e Não-Ativo. Isto ocorre apenas quando os projetos estão diferentes e a sincronização automática de projetos está habilitada em ambos os CPs.

A sincronização é sempre no sentido do CP Ativo para o Não-Ativo e basta que um dos canais de sincronismo (NETA ou NETB) esteja operacional para que este serviço seja executado.

Quando a sincronização estiver habilitada, serão sincronizados os seguintes arquivos e serviços:

- Projeto de aplicação (código executável)
- *Project archive* (código fonte)
- Usuários e grupos
- Direitos de acesso
- Trace

O serviço de sincronização irá iniciar em até trinta segundos, após um dos CPs entrar no modo Ativo, e, após seu início, irá checar o CRC dos projetos a cada cinco segundos.

Quando uma sincronização for iniciada, o CP Não-Ativo irá para o modo Stop, no estado de Não Configurado. Após a transferência de todos os arquivos necessários, o CP Não-Ativo irá para Run, no estado de Inicializando. Caso a transferência falhe, o CP voltará para o estado Não-Configurado.

O tempo que a sincronização levará para ser efetuada dependerá do tamanho do projeto. Em média, a taxa de transferência entre os canais de sincronismo é de aproximadamente 500 Kbytes/s.

Caso a sincronização seja interrompida (perda da comunicação entre os canais de sincronismo) durante a transferência dos arquivos do CP Ativo para o Não-Ativo, o procedimento será abortado e reinicializado quando a comunicação for restaurada. Somente após a conclusão de todo o procedimento o CP Não-Ativo irá para o modo Run.

Além de manter sincronizados os projetos, a Sincronização do Projeto também irá impedir que o CP Não-Ativo assuma estados superiores a Inicializando caso o CRC esteja diferente ou alguma Alteração Online esteja pendente no CP Ativo.

ATENÇÃO

Uma sincronização de projeto terá os mesmos efeitos de um download no CP Não-Ativo. Este serviço não é executado se a sincronização automática de projetos estiver desabilitada, conforme descreve-se na seção [Desabilitação da Sincronização de Projetos](#). Nenhum serviço de sincronização entre o CPA e o CPB irá funcionar caso sejam invertidos os cabos dos canais de sincronismo. Por exemplo, conectar o canal NETA do CPA no NETB do CPB e o canal NETB do CPA no NETA do CPB.

ATENÇÃO

Na atualização da versão 1.20 para versões superiores do software MasterTool IEC XE houve uma modificação no protocolo de comunicação dos canais de sincronismo. Portanto, não será possível efetuar a sincronização de dados entre dois CPs quando uma das aplicações tiver sido criada em uma versão anterior à 1.21 e outra em uma versão igual ou superior. Para voltar a efetuar o sincronismo, deve-se [Não Carregar a Aplicação na Inicialização](#) do CP com um projeto mais antigo e deixar que a aplicação seja sincronizada automaticamente, quando este CP for para o estado Não-Configurado, durante a sua inicialização.

ATENÇÃO

Antes da versão 2.01 do MasterTool IEC XE, ao enviar o código fonte para o CP Ativo, o CP Reserva passava para o estado Não-Configurado para sincronização do mesmo. Contudo, ao concluir a operação de sincronização, o CP permanecia no estado Não-Configurado, sendo necessário passar o CP para o estado Reserva através do botão STAND-BY do PX2612 ou de comando equivalente. A partir da versão 2.01 o CP que está em Reserva irá alterar seu estado para Não-Configurado durante o processo de sincronização, mas irá voltar automaticamente quando os códigos fonte forem iguais entre os dois Half-Clusters.

6.3.8. Desabilitação da Sincronização de Projetos

Na seção [Serviços de Sincronização Esporádicos através de NETA e NETB](#) foram descritos serviços para a sincronização do projeto de aplicação e do *project archive*. Estes serviços normalmente devem estar habilitados e são úteis quando as modificações de projeto podem ser carregadas online no CP Ativo, sendo, a seguir, automaticamente retransmitidas para o CP Reserva via canais de sincronismo NETA / NETB.

Entretanto, existem modificações de projeto que não podem ser carregadas online em nenhum CP, como por exemplo a inclusão de módulos em uma remota PROFIBUS, ou a inclusão de uma nova remota PROFIBUS. Nestes casos, valendo-se da redundância do CP e da rede PROFIBUS, pode-se fazer tais modificações sem interromper o controle de processo. Um procedimento para atingir este objetivo é descrito na seção [Explorando a Redundância para Carga Offline de Modificações sem Interrupção do Controle do Processo](#).

Neste procedimento, é necessário desabilitar temporariamente as sincronizações de projeto, permitindo que, por algum tempo, um dos CPs opere com a nova versão do projeto, enquanto o outro CP ainda opera com a versão antiga do projeto.

Uma UCP NX3030 possui um registro de Desabilitação da Sincronização de Projetos, não volátil, que permite desabilitar os serviços para sincronização do projeto de aplicação e do *project archive*. Este registro pode ser ajustado com o programador MasterTool. Basta desabilitar a sincronização de projetos em um dos dois CPs para que ela não mais ocorra.

Para desabilitar a sincronização de projetos, deve-se, primeiramente, conectar-se no CP desejado com o software MasterTool (ver seção [Conexão do MasterTool com uma UCP NX3030 de um CP Redundante](#)).

A seguir, no menu Comunicação / Configuração de Redundância, abre-se a combo-box “Sincronização do Projeto”, que permite selecionar uma das duas seguintes opções:

- Habilitar
- Desabilitar

Deve-se selecionar a opção “*Desabilitar*” e depois pressionar o botão “*Escrever*”, ao lado desta combo-box. Uma mensagem informará se a operação teve sucesso ou não.

A configuração da desabilitação de sincronismo de projetos não faz parte do projeto redundante desenvolvido com o MasterTool. Tal configuração reside apenas em uma área de memória não-volátil da UCP, que pode ser lida ou gravada usando o MasterTool. O MasterTool não salva esta configuração em nenhum arquivo.

Esta configuração é copiada a cada ciclo da MainTask, da memória não volátil para o diagnóstico DG_NX4010.tRedundancy.RedDgnLoc. O usuário pode verificar este diagnóstico no CP para conferir se o comando teve sucesso, desde que o mesmo esteja em modo Run (DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag. bProjectSyncDisable deve valer 1). Caso o CP não esteja em modo Run, é possível verificar esta configuração diretamente no visor da UCP NX3030 (ver seção [Diagnósticos da Redundância no Visor Gráfico da UCP NX3030](#)).

O diagnóstico DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.bProjectSyncDisable do CP pode ser observado também no outro half-cluster, através do diagnóstico DG_NX4010.tRedundancy.RedDgnRem.sGeneral_Diag.bProjectSyncDisable (desde que o CP Não-Ativo esteja em modo Run). Determinado CP (Ativo ou Não-Ativo) suspenderá o serviço de sincronização de projetos sempre que qualquer um dos seguintes bits estiver ligado:

- DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.bProjectSyncDisable
 - Bit local, deste CP. Este CP está com a sincronização de projetos desabilitada
- DG_NX4010.tRedundancy.RedDgnRem.sGeneral_Diag.bProjectSyncDisable
 - Bit remoto, do outro CP. O outro CP está com a sincronização de projetos desabilitada

ATENÇÃO

O registro de Desabilitação da Sincronização de Projetos não faz parte do projeto do CP redundante, e portanto não é salvo como parte deste projeto no computador onde o MasterTool executa. O registro é salvo apenas na memória não volátil da UCP.

6.3.9. Configurações de Redes PROFIBUS

É possível instalar até quatro módulos Mestre PROFIBUS NX5001 em cada half-cluster. Portanto, pode-se configurar até duas redes PROFIBUS redundantes, denominadas PROFIBUS 1 e PROFIBUS 2, ou até quatro redes simples, denominadas PROFIBUS 1, PROFIBUS 2, PROFIBUS 3 e PROFIBUS 4, ou ainda uma rede redundante e duas simples, denominadas PROFIBUS 1, PROFIBUS 2 e PROFIBUS 3.

6.3.9.1. Redundância PROFIBUS

Cada uma das redes PROFIBUS pode ser redundante ou não-redundante. Por exemplo, se a rede PROFIBUS 1 for redundante, ela se dividirá em redes PROFIBUS 1 A e PROFIBUS 1 B. Caso não seja redundante, existirá apenas a rede PROFIBUS 1 A. O mesmo vale para a PROFIBUS 2.

A Figura 158 mostra um exemplo com uma única rede PROFIBUS (PROFIBUS 1), que é redundante (PROFIBUS 1 A e PROFIBUS 1 B).

Somente alguns tipos de remotas podem ser conectadas diretamente a este tipo de rede PROFIBUS redundante:

- PO5063V5: escravo PROFIBUS DP-V0 para remotas da Série Ponto
- PO5065: escravo PROFIBUS DP-V1 com Hart, para remotas da Série Ponto
- AL-3416: escravo PROFIBUS DP-V0 para CP AL-2004
- NX5210: escravo PROFIBUS DP-V0 para remotas da Série Nexto

A Figura 158 também mostra a possibilidade de conectar remotas não-redundantes a este tipo de rede PROFIBUS redundante, através do módulo AL-2433 (ProfiSwitch). Tais remotas PROFIBUS não-redundantes podem ser de qualquer marca ou modelo.

6.3.9.2. Modos de Falha PROFIBUS Vital e Não-Vital

Cada uma das redes PROFIBUS pode ser configurada em dois modos diferentes:

- Falha vital: caso esta rede falhe completamente, esta falha poderá determinar uma transição de estado de redundância no CP redundante (switch-over). No caso de uma rede PROFIBUS redundante, uma falha completa consiste na falha das duas redes que a compõem.
- Falha não vital: mesmo que esta rede falhe completamente, esta falha não determinará uma transição de estado de redundância no CP redundante (switch-over).

6.3.10. Redes Ethernet Redundantes com NIC Teaming

A Figura 158 mostra dois exemplos de rede Ethernet redundantes, com NIC Teaming.

No primeiro caso, a UCP NX3030 conecta-se à rede de supervisão (SCADAs), também utilizada para configuração através do MasterTool. As duas portas Ethernet da UCP NX3030 (NET 1 e NET 2) formam um par redundante NIC Teaming, interligadas em dois switches diferentes (Ethernet A e Ethernet B). Em algum ponto, estes dois switches devem ser interligados, para que exista conexão entre as duas portas NIC Teaming e disponibilidade ainda maior (contra falhas duplas).

No segundo caso, dois módulos NX5000 também formam um par redundante NIC Teaming, interligados em dois switches diferentes (Ethernet HSDN A e Ethernet HSDN B). Em algum ponto, estes dois switches devem ser interligados, para que exista conexão entre as duas portas NIC Teaming e disponibilidade ainda maior (contra falhas duplas).

Tais arquiteturas Ethernet possibilitam excelente disponibilidade, contra falhas nas portas Ethernet, em cabos e em switches.

ATENÇÃO

Caso dois módulos consecutivos, ou portas Ethernet, formem um par redundante NIC Teaming, a configuração dos parâmetros básicos e adição de dispositivos MODBUS apenas será possível na primeira interface, a seguinte, terá seus parâmetros bloqueados.

Um conjunto de duas portas Ethernet formando um par NIC Teaming possui um único endereço IP, vinculado ao par de portas. Desta forma, um cliente como um SCADA ou MasterTool, conectado a um servidor no CP, não precisa se preocupar em trocar o endereço IP caso haja falha em alguma das portas do par NIC Teaming.

Além disso, cada uma das interfaces que formam um par NIC Teaming possui diagnósticos separados, com o intuito de facilitar a depuração de falhas que eventualmente possam surgir.

Maiores detalhes sobre configuração e diagnósticos de portas NIC Teaming são fornecidos nas seções:

- [Configurações das Portas Ethernet da UCP NX3030 \(NET 1 e NET 2\)](#)
- [Configuração dos Módulos NX5000](#)

6.3.11. Métodos de Troca de IP

Um cluster redundante da Série Nexto dispõe de quatro métodos para a troca de IP das portas Ethernet dos módulos NX5000 de cada half-cluster e um método para a troca de IP das portas NET1 e NET2 da UCP NX3030. Esses métodos definem o comportamento das portas, no que se refere ao IP da mesma, de acordo com o estado atual do half-cluster (Ativo ou Não-Ativo) e com o half-cluster em questão (CPA ou CPB).

Os métodos são: IP Fixo, Troca Automática de IP, IP Ativo e Múltiplos IPs.

Ao todo, podemos relacionar até quatro IPs, de acordo com o método de troca de IP.

6.3.11.1. IP Fixo

É o método mais simples de endereçamento IP e pode ser configurado nas interfaces Ethernet dos módulos Ethernet NX5000. Nesse método, somente relacionamos os endereços de IP do CPA e do CPB. Independente do estado da redundância, CP Ativo ou Não-Ativo, o CPA sempre responderá pelo IP configurado, assim como o CPB.

Parâmetros da Porta Ethernet

Método de Endereçamento de IP do Cluster

IP Fixo

Endereçamento de IP do Cluster

Endereço de IP do CPA 192 . 168 . 15 . 68

Endereço de IP do CPB 192 . 168 . 15 . 69

Máscara de Subrede 255 . 255 . 255 . 0

Endereço do Gateway 192 . 168 . 15 . 253

Avançado...

Figura 166: Método IP Fixo

Parâmetros que devem ser configurados no método de IP Fixo:

- Endereço de IP do CPA: Endereço para comunicação com o CPA
- Endereço de IP do CPB: Endereço para comunicação com o CPB
- Máscara de Subrede
- Endereço do Gateway

6.3.11.2. Troca Automática de IP

A Troca Automática de IP pode ser configurada nas interfaces Ethernet dos módulos Ethernet NX5000. Nesse método, o IP do half-cluster irá depender do estado do CP (Ativo ou Não-Ativo). A cada switch-over ocorre a troca de IPs entre os half-clusters para que eles assumam o endereço IP do novo estado da redundância.

Observação: para este método de endereçamento, as portas Ethernet de ambos os CPs (CPA e CPB) irão assumir o mesmo endereço IP enquanto os dois CPs estiverem no estado não-ativo, provocando conflito de endereçamento na rede. Considerando que essa é uma situação incomum, onde nenhum CP está controlando o sistema, isso não chega a ser um problema grave, mas deve ser considerado.

Figura 167: Método de Troca Automática de IP

Parâmetros que devem ser configurados no método de Troca Automática de IP:

- Endereço de IP Ativo: Endereço de IP para comunicação com o CP Ativo
- Endereço de IP Não Ativo: Endereço de IP para comunicação com o CP Não-Ativo
- Máscara de Subrede
- Endereço do Gateway

6.3.11.3. IP Ativo

Esse método é o utilizado nas NETs da UCP NX3030 redundante e também é possível ser configurado nos módulos NX5000. Nesse método há um IP para o half-cluster Ativo e mais dois IPs, um para o CPA e outro para o CPB. Nas NETs da UCP NX3030 redundante, o Endereço de IP Ativo será adicionado à interface do CP Ativo, podendo ser utilizados tanto o Endereço de IP Ativo, quanto o Endereço de IP do CPX para comunicar com o CP. Já para os módulos Ethernet NX5000, o Endereço de IP Ativo irá substituir o Endereço de IP do CPX Não Ativo, quando o CP estiver em modo Ativo.

Figura 168: Método IP Ativo – NX3030 Redundante

Parâmetros que devem ser configurados no método de IP Ativo para as NETs de uma UCP NX3030 Redundante:

- Endereço de IP Ativo: Endereço de IP adicionado à interface quando o CP estiver em estado Ativo
- Endereço de IP do CPA: Endereço para comunicação com o CPA, independente do seu estado atual
- Endereço de IP do CPB: Endereço para comunicação com o CPB, independente do seu estado atual

- Máscara de Subrede
- Endereço do Gateway

Parâmetros da Porta Ethernet

Método de Endereçamento de IP do Cluster

IP Ativo

Endereçamento de IP do Cluster

Endereço de IP Ativo	192 . 168 . 15 . 68
Endereço de IP do CPA Não Ativo	192 . 168 . 15 . 69
Endereço de IP do CPB Não Ativo	192 . 168 . 15 . 70
Máscara de Subrede	255 . 255 . 255 . 0
Endereço do Gateway	192 . 168 . 15 . 253

Avançado...

Figura 169: Método IP Ativo – NX5000

Parâmetros que devem ser configurados no método de IP Ativo para módulos Ethernet NX5000:

- Endereço de IP Ativo: Endereço para comunicação com o CP Ativo. Substitui o Endereço de IP do CPX Não Ativo
- Endereço de IP do CPA Não Ativo: Endereço para comunicação com o CPA, quando em estado Não-Ativo
- Endereço de IP do CPB Não Ativo: Endereço para comunicação com o CPB, quando em estado Não-Ativo
- Máscara de Subrede
- Endereço do Gateway

6.3.11.4. Múltiplos IPs

O método de Múltiplos IPs pode ser configurado nas interfaces Ethernet dos módulos Ethernet NX5000. Nesse método, há um IP para cada half-cluster e para cada estado do CP. O CPA irá assumir um endereço IP quando estiver em Ativo e outro endereço IP quando estiver em Não-Ativo. O mesmo ocorre para o CPB em função do seu estado (Ativo e Não-Ativo).

Parâmetros da Porta Ethernet

Método de Endereçamento de IP do Cluster

Múltiplos IPs

Endereçamento de IP do Cluster

Endereço de IP do CPA Ativo: 192 . 168 . 15 . 68

Endereço de IP do CPA Não Ativo: 192 . 168 . 15 . 69

Endereço de IP do CPB Ativo: 192 . 168 . 15 . 70

Endereço de IP do CPB Não Ativo: 192 . 168 . 15 . 71

Máscara de Subrede: 255 . 255 . 255 . 0

Endereço do Gateway: 192 . 168 . 15 . 253

Avançado...

Figura 170: Método Múltiplos IPs

Parâmetros que devem ser configurados no método de Múltiplos IPs:

- Endereço de IP do CPA Ativo: Endereço para comunicação com o CPA, quando o mesmo estiver em estado Ativo
- Endereço de IP do CPA Não Ativo: Endereço para comunicação com o CPB, quando o mesmo estiver em estado Não-Ativo
- Endereço de IP do CPB Ativo: Endereço para comunicação com o CPB, quando o mesmo estiver em estado Ativo
- Endereço de IP do CPB Não Ativo: Endereço para comunicação com o CPB, quando o mesmo estiver em estado Não-Ativo
- Máscara de Subrede
- Endereço do Gateway

6.3.12. Uso Combinado de NIC Teaming e IP Ativo

Caso determinado par de portas forme um par NIC Teaming em um CP redundante, estas portas podem implementar, ao mesmo tempo, as estratégias NIC Teaming e IP Ativo.

Por exemplo, se as portas NET 1 e NET 2 da UCP NX3030 formam um par NIC Teaming, então:

- Endereço de IP do CPA: endereço IP das portas NET 1 + NET 2 da UCP NX3030 do CPA
- Endereço de IP do CPB: endereço IP das portas NET 1 + NET 2 da UCP NX3030 do CPB
- Endereço de IP Ativo: endereço IP das portas NET 1 + NET 2 daquela UCP NX3030 que se encontra no CP Ativo

Desta maneira, associa-se a excelente disponibilidade da estratégia NIC Teaming com a praticidade da estratégia de IP Ativo, que dispensa scripts em sistemas SCADA ou em outros clientes conectados a servidores no CP Ativo.

6.3.13. Uso de Interfaces Ethernet com Indicação de Falha Vital

As portas Ethernet da UCP NX3030 e dos módulos NX5000 podem ser configuradas para gerar falhas vitais. Esta opção é importante para as aplicações nas quais os módulos de entradas e saídas estão distribuídos através de rede Ethernet. Neste caso, se ocorrer uma falha na porta Ethernet, isso irá gerar um switch-over. Este comportamento é aplicável apenas em portas Ethernet nas quais exista pelo menos um driver de comunicação que gere falha.

Os drivers de comunicação que geram falha vital são MODBUS Client e o MODBUS Symbol Client (todas as referências a MODBUS Cliente nas próximas seções se aplicam aos dois casos). Já os drivers MODBUS Server, MODBUS Symbol Server e EtherCAT Master não geram falha vital. Desta forma, se uma porta Ethernet tem um driver MODBUS Cliente configurado e ocorre uma falha na porta Ethernet, será gerado um switch-over se a opção de falha vital estiver habilitada. Se o driver configurado na porta Ethernet for um MODBUS Servidor, mesmo que ocorra falha na porta, isso não irá gerar uma falha vital que causa um switch-over.

Para que uma falha seja considerada falha vital em uma porta Ethernet de um MODBUS Cliente, todos os servidores configurados no driver devem estar em falha. Ou seja, caso exista mais de um driver MODBUS Cliente configurado na mesma porta Ethernet, será considerada falha vital quando todos os servidores de ambos os Clientes estiverem em falha.

Quando a porta Ethernet estiver configurada para operar com NIC Teaming, a falha vital será considerada apenas quando as duas portas do par falharem.

6.3.13.1. Falha na Interface Ethernet

Um switch-over pode ser gerado devido a falha na interface Ethernet, por exemplo uma perda de link. A perda de link pode ser ocasionada, por exemplo, pelo rompimento de um cabo ou falha em um switch na rede Ethernet. Nestas condições, é necessário que, além de estar configurado para gerar falha vital, exista uma instância MODBUS Cliente configurada na interface Ethernet.

Quando o intervalo da MainTask for maior ou igual a 100 ms, após a falha ser detectada o switch-over irá acontecer em até dois ciclos da MainTask. Quando o intervalo da MainTask for menor que 100 ms o switch-over ocorrerá em até 100 ms mais o tempo da MainTask após a detecção da falha.

6.3.13.2. Falha em MODBUS Servidor conectado

O tempo para detectar a falha em uma remota MODBUS Servidor depende das configurações de time-out configuradas em cada MODBUS Cliente. Quando é detectada a falha em todos os Servidores, o diagnóstico `bAllDevicesCommFailure` (ver seção [Diagnósticos MODBUS utilizados na Redundância](#)) muda seu estado para TRUE. Quando isso acontece, o switch-over irá acontecer 3 segundos após esta transição.

6.3.14. Uso de Comunicação OPC DA com Projetos Redundantes

O protocolo OPC DA pode ser configurado para comunicação de clusters redundantes com sistemas SCADA. Quando esta opção é selecionada na criação de um projeto redundante, o objeto *Symbol Configuration* é adicionado ao projeto. Neste objeto são configuradas as variáveis do sistema que serão enviadas para o sistema SCADA. Esta opção de comunicação é habilitada nas portas Ethernet da UCP NX3030. Para mais informações a respeito da configuração da comunicação OPC DA com projetos redundantes consulte a seção [Configuração com CP no Servidor OPC DA com Redundância de Conexão](#) deste Manual.

6.3.15. Estados de um CP Redundante

Em um sistema redundante, um CP (CPA ou CPB) pode assumir os seguintes estados:

- Ativo
- Reserva
- Inativo
- Não-Configurado
- Inicializando

ATENÇÃO

Frequentemente este manual utilizará a designação “Não-Ativo” para qualquer estado diferente de Ativo, ou seja, para designar qualquer um dos outros quatro estados (Reserva, Inativo, Não-Configurado e Inicializando).

A seguir, estes cinco estados são descritos brevemente. Maiores detalhes sobre os estados de um CP redundante serão abordados na seção *Transições entre Estados de Redundância*, ao descrever a máquina de estados e as causas das transições entre os mesmos.

6.3.15.1. Estado Não-Configurado

Este é o estado de redundância inicial. O CP encontra-se neste estado de redundância:

- Por convenção, enquanto o CP estiver desligado
- Antes de iniciar a MainTask
- Antes de chavear para o estado Inicializando
- Caso ocorra uma reinicialização através de um comando como Reset a Quente, Reset a Frio ou Reset Origem

Caso a MainTask esteja executando no estado Não-Configurado, as seguintes tarefas são executadas:

- Os mestres PROFIBUS estão desabilitados
- Os serviços de sincronização cíclicos são executados (ver seção [Serviços de Sincronização Cíclicos através de NETA e NETB](#)), desde que as condições para sua execução estejam presentes
- Os serviços de sincronização esporádicos também podem ser executados (ver seção [Serviços de Sincronização Esporádicos através de NETA e NETB](#))

O CP ficará bloqueado no estado Não-Configurado se o outro CP estiver em estado Ativo, e o projeto deste CP for diferente do projeto do CP Ativo (exceto se a sincronização automática de projetos estiver desabilitada – ver seção [Desabilitação da Sincronização de Projetos](#)). Caso esta situação não ocorra, ocorre uma transição do estado Não-Configurado para o estado Inicializando assim que chegar uma requisição de configuração.

Algumas vezes, o CP entra no estado Não-Configurado já tendo recebido uma requisição de configuração automática, não sendo necessária uma nova requisição para mudar para o estado Inicializando. Isto ocorre, por exemplo, na energização do CP.

Outras vezes, o usuário deverá requisitar manualmente esta configuração, por exemplo, pressionando um botão do painel de comando de redundância PX2612 ou utilizando um comando na estrutura de diagnósticos. Requisições de configuração manuais tipicamente são necessárias quando alguma manutenção por parte do usuário é necessária antes de sair do estado Não-Configurado, por exemplo, se o CP atingiu o estado Não-Configurado devido a alguma falha.

Depois de sair do estado Não-Configurado, o CP pode voltar a este estado, devido a eventos tais como:

- Reinicialização (Reset a Quente, Reset a Frio ou Reset Origem)
- Desligamento do CP
- Diferença de projeto entre este CP e o CP Ativo

6.3.15.2. Estado Inicializando

Diferente de todos os outros quatro estados que podem ter duração indeterminada, o estado Inicializando é temporário, durando poucos segundos. Este estado é sempre alcançado a partir do estado Não-Configurado, através de uma requisição de configuração.

Ao entrar no estado Inicializando, diversas ações, testes e verificações são executados, para decidir qual será o próximo estado:

- Mestres PROFIBUS são habilitados em estado passivo. O modo passivo serve para testar os circuitos de transmissão e recepção PROFIBUS e o meio físico, para evitar a ocorrência de uma falha oculta
- Verificar se a identificação do CP está correta (deve ser CPA ou CPB)
- Verificar se há problemas nos parâmetros de configuração extraídos do projeto MasterTool
- Verificar a integridade do módulo NX4010
- Os serviços de sincronização cíclicos são executados (ver seção [Serviços de Sincronização Cíclicos através de NETA e NETB](#)), desde que as condições para sua execução estejam presentes
- Verificar compatibilidade de versões de firmware entre os dois CPs
- Verificar igualdade de projetos entre os dois CPs, se a sincronização automática de projetos estiver habilitada (ver seção [Desabilitação da Sincronização de Projetos](#))
- Caso o outro CP esteja em estado Ativo, verificar a possibilidade de estabelecer comunicação PROFIBUS passiva com o mesmo. O modo passivo serve para testar os circuitos de transmissão e recepção PROFIBUS e o meio físico, para evitar a ocorrência de uma falha oculta
- Caso o outro CP esteja em estado desconhecido devido a falhas em NETA e NETB, verificar a possibilidade de estabelecer comunicação PROFIBUS passiva com o mesmo; caso não exista rede PROFIBUS no projeto e o mesmo não faça uso do Painel PX2612, verificar se a NET1/NET2 da UCP estão recebendo pacotes de Keep Alive da UCP do outro half-cluster.

Dependendo do resultado destas verificações e testes, o CP pode ir do estado Inicializando para qualquer um dos outros quatro estados.

6.3.15.3. Estado Ativo

Neste estado, o CP controla o processo automatizado, usando o programa ActivePrg, executado somente neste estado. O CP Ativo também atualiza o sistema de E/S remotas PROFIBUS, colocando seus mestres PROFIBUS em modo ativo. O modo ativo serve para estabelecer comunicação com as remotas (escravos) PROFIBUS.

O CP Ativo também verifica seus diagnósticos internos e requisições de switch-over do usuário para determinar se um switch-over é necessário. O CP normalmente só sairá do estado Ativo se souber que o outro CP está em estado Reserva, e apto para assumir como Ativo.

No entanto, existem algumas situações em que o CP Ativo poderá sair do estado Ativo mesmo sem ter certeza de que o outro CP está em estado Reserva (por exemplo, se este CP for desligado).

6.3.15.4. Estado Reserva

Neste estado, o CP está pronto para chavear para o estado Ativo, caso haja uma demanda para isso, tal como uma falha no CP Ativo.

O CP Reserva também verifica seus próprios diagnósticos, e poderá chavear para o estado Não-Configurado ou Inativo em função de algumas falhas.

Mestres PROFIBUS são habilitados em estado passivo. O modo passivo serve para testar os circuitos de transmissão e recepção PROFIBUS e o meio físico, para evitar a ocorrência de uma falha oculta. Falhas totais em redes PROFIBUS configuradas como vitais causam um chaveamento para o estado Inativo. Uma falha total em uma rede PROFIBUS atinge as duas redes que a compõem (rede PROFIBUS redundante) ou a única rede que a compõe (rede PROFIBUS não-redundante).

Se estiverem sendo usadas as interfaces ethernet com a opção de falha vital habilitada, os clientes são habilitados em estado passivo. Falhas totais em redes ethernet configuradas como vitais causam um chaveamento para o estado Inativo. Uma falha total em uma rede Ethernet atinge as duas redes que a compõem (opção de Redundância de Comunicação habilitada) ou a única rede que a compõe (opção de Redundância de Comunicação desabilitada).

6.3.15.5. Estado Inativo

Este estado normalmente é alcançado depois de alguns tipos de falhas, ou devido a alguma solicitação manual antes de uma manutenção programada.

Mestres PROFIBUS são habilitados em estado passivo. O modo passivo serve para testar os circuitos de transmissão e recepção PROFIBUS e o meio físico, para evitar a ocorrência de uma falha oculta.

Antes de deixar este estado, primeiro deve-se corrigir as falhas diagnosticadas ou executar as manutenções programadas, que causaram a transição para o estado Inativo. Depois, deve-se causar uma transição para o estado Não-Configurado já solicitando uma configuração, para logo em seguida chavear para o estado Inicializando. Depois do estado Inicializando, o CP pode:

- Retornar ao estado Inativo se determinados tipos de falhas persistem
- Retornar ao estado Não-Configurado para outros tipos de falhas
- Ir para o estado Reserva se o outro CP está no estado Ativo
- Ir para o estado Ativo se o outro CP não está no estado Ativo

6.3.16. Funções do Painel de Comando de Redundância PX2612

O painel de comando de redundância PX2612 é mostrado na Figura 161. A Figura 162 mostra sua serigrafia frontal com maiores detalhes e a Figura 163 mostra como esse painel deve ser interligado aos half-clusters CPA e CPB.

O PX2612 divide-se em duas seções, uma delas controlada pelo CPA, e outra pelo CPB. Estes controles ocorrem através dos cabos AL-2317/A para o CPA, e AL-2317/B para o CPB, e permitem que cada CP leia três botões, e escreva sobre três LEDs e sobre um relé NA.

Observando a frontal na Figura 162:

- CPA executa a leitura dos botões STAND-BY e INACTIVE dentro do setor PLC A
- CPA executa a leitura do botão TURN ON PLC B
- CPA executa a escrita nos três LEDs (ACTIVE, STAND-BY e INACTIVE) do setor PLC A
- CPA executa a escrita no relé RL B, utilizado para desligar o CPB
- CPB executa a leitura dos botões STAND-BY e INACTIVE dentro do setor PLC B
- CPB executa a leitura do botão TURN ON PLC A
- CPB executa a escrita nos três LEDs (ACTIVE, STAND-BY e INACTIVE) do setor PLC B
- CPB executa a escrita no relé RL A, utilizado para desligar o CPA

6.3.16.1. Botões do PX2612

Esta seção descreve as funções dos botões do PX2612.

O botão STAND-BY tem as seguintes funções:

- Solicitar um chaveamento do estado Ativo para o estado Reserva, o que pode ser útil para executar alguma manutenção programada no CP Ativo. Depois que o CP Ativo chaveia para Reserva (e consequentemente o CP Reserva chaveia para Ativo), é possível chaveá-lo de Reserva para Inativo usando o botão INACTIVE, e então executar a manutenção programada no estado Inativo

- Solicitar uma configuração que provoca um chaveamento do estado Não-Configurado para o estado Inicializando, tipicamente depois de reparar falhas que provocaram a transição para o estado Não-Configurado. Depois do estado Inicializando, normalmente espera-se que o CP vá para o estado Reserva (ou Ativo, se o outro CP não estiver no estado Ativo)
- Solicitar um chaveamento do estado Inativo para o estado Não-Configurado já solicitando uma configuração. Isto ocorre tipicamente depois de corrigir falhas que provocaram a transição para o estado Inativo. Depois do estado Não-Configurado, a configuração já deve levar ao estado Inicializando. Depois do estado Inicializando, normalmente espera-se que o CP vá para o estado Reserva (ou Ativo, se o outro CP não estiver no estado Ativo)

O botão INACTIVE solicita um chaveamento do estado Reserva para o estado Inativo, o que pode ser útil para executar alguma manutenção programada no CP Reserva. Depois desta manutenção, pode-se utilizar o botão STAND-BY para tentar voltar ao estado Reserva, passando pelos estados Não-Configurado e Inicializando (ver descrição anterior das funções do botão STAND-BY).

O botão TURN ON PLCx (x = B para CPA, ou x = A para CPB) serve para provocar um religamento do outro CP, caso este CP o tenha desligado anteriormente. Conforme será visto na seção *Transições entre Estados de Redundância*.

Existem situações excepcionais em que um CP desliga o outro ao assumir como Ativo, para evitar que haja a possibilidade de ambos os CPs assumirem simultaneamente o estado Ativo.

ATENÇÃO

Para que um pressionamento de botão seja considerado, deve-se mantê-lo pressionado no mínimo por 1 segundo. Durante esse segundo, somente esse botão deve estar pressionado (os outros 2 botões devem estar liberados).

ATENÇÃO

Existem maneiras alternativas de gerar os mesmos efeitos dos botões STAND-BY, INACTIVE e TURN ON PLCx. Pode-se utilizar comandos gerados por este CP ou pelo CP remoto, conforme descrito preliminarmente na seção *Estruturas de Dados de Diagnósticos, Comandos e Usuário*. Uma descrição mais detalhada destes comandos pode ser encontrada na seção [Comandos da Redundância](#).

6.3.16.2. LEDs do PX2612

Os LEDs do PX2612 servem para informar o estado de redundância, conforme mostra a tabela abaixo:

Estado de redundância	LED ACTIVE	LED STAND-BY	LED INACTIVE
Não-Configurado	apagado	apagado	apagado
Inicializando	aceso	aceso	aceso
Ativo	aceso	apagado	apagado
Ativo (recente)	piscando	apagado	apagado
Ativo (desligando outro CP)	aceso	piscando	apagado
Ativo (recente, e desligando outro CP)	piscando	piscando	apagado
Reserva	apagado	aceso	apagado
Inativo	apagado	apagado	aceso

Tabela 202: LEDs do PX2612

Cada LED pode estar apagado, aceso ou piscando. Caso esteja piscando, fica aceso por 0,5 segundos, e apagado por 0,5 segundos.

Observa-se que existem quatro animações diferentes para o estado Ativo, devido às duas seguintes características:

- Nos dois primeiros segundos em estado Ativo o LED ACTIVE pisca, e depois permanece aceso. Esta animação foi criada porque, nestes primeiros instantes em estado Ativo, normalmente o CP não aceitará comandos que o façam sair do estado Ativo. Maiores detalhes sobre este comportamento do CP Ativo serão fornecidos na seção *Transições entre Estados de Redundância* e na seção [Primeiros Instantes em Estado Ativo](#)
- Caso este CP esteja desligando o outro CP através do seu relé do PX2612, o LED STAND-BY pisca, e do contrário permanece apagado

6.3.16.3. Relés do PX2612

O PX2612 possui dois relés NA. O CPA pode controlar RL B, para comandar o desligamento do CPB. O CPB pode controlar RL A, para comandar o desligamento do CPA.

Tais situações de desligamento ocorrem em situações excepcionais, descritas na seção *Transições entre Estados de Redundância*.

6.3.17. Transições entre Estados de Redundância

A figura seguinte mostra a máquina de estados da redundância, ilustrando todas as possíveis transições entre estados de redundância.

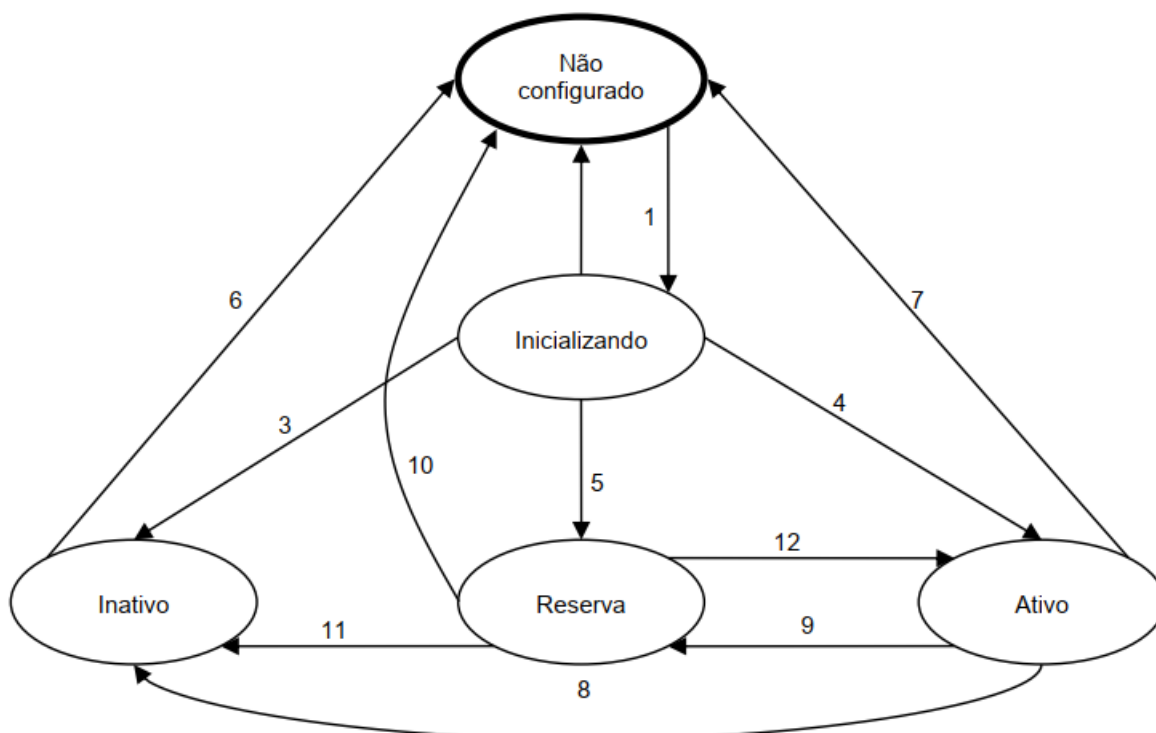


Figura 171: Máquina de Estados da Redundância

As seguintes subseções descreverão todas estas transições, e as causas que podem dispará-las. Para interpretar corretamente o funcionamento desta máquina de estados, é preciso estabelecer algumas regras e sequências:

- Transições que se originam do mesmo estado devem ser avaliadas na sequência dada pelo número da transição. Por exemplo, as transições 2, 3, 4 e 5 se originam do estado Inicializando. Neste exemplo, avalia-se primeiro a transição 2, depois a 3, depois a 4 e finalmente a 5. Caso a transição 2 seja disparada, as transições 3, 4 e 5 nem serão avaliadas
- Dentro de uma subseção específica que descreve uma transição, diversas condições podem disparar esta transição. Estas condições devem ser avaliadas na ordem que aparecem na subseção. Qualquer uma destas condições que se tornar verdadeira pode causar a transição. Se uma condição causar a transição, as próximas condições não serão avaliadas
- Transições só podem ser disparadas se o CP estiver energizado e a MainTask estiver executando. Do contrário, assume-se que o CP esteja no estado Não-Configurado
- Em diversos casos, mencionam-se transições causadas por botões do painel PX2612. Deve-se lembrar que existem alternativas para estes botões, que são comandos internos provenientes deste CP ou do outro CP (via NETA / NETB). Tais comandos foram mencionados preliminarmente na seção *Estruturas de Dados de Diagnósticos, Comandos e Usuário*, e serão melhor detalhados na seção [Comandos da Redundância](#). Nas subseções seguintes, por simplicidade, estes comandos alternativos não serão citados, mas deve-se ter consciência que eles podem causar as mesmas transições do botão equivalente no PX2612

6.3.17.1. Transição 1 - Não-Configurado para Inicializando

ATENÇÃO

As condições desta subseção não devem ser avaliadas caso o outro CP esteja em estado Ativo e os projetos estejam diferentes. Este CP deve permanecer no estado Não-Configurado enquanto seu projeto estiver diferente do projeto do outro CP, se o outro CP estiver em estado Ativo. Esta nota não é válida se a sincronização automática de projetos estiver desabilitada (ver seção [Desabilitação da Sincronização de Projetos](#)), pois neste caso admite-se diferenças de projetos entre os CPs.

- Um pedido de configuração já existia ao entrar no estado Não-Configurado. Isto ocorre na energização do CP, e também em outras situações, descritas nas próximas subseções
- O botão STAND-BY foi pressionado ou o comando de ir para Reserva foi executado nos diagnósticos da redundância durante o estado Não-Configurado. Isto causa um pedido de configuração manual. O usuário tipicamente aperta STAND-BY ou executa o comando para ir para Reserva depois de reparar falhas que anteriormente levaram este CP ao estado Não-Configurado

6.3.17.2. Transição 2 - Inicializando para Não-Configurado

- Este CP foi desligado ou reinicializado (Reset a Quente, Reset a Frio ou Reset Origem) ou sua UCP entrou em Modo Stop
- O registro de identificação deste CP é inválido (diferente de CPA ou CPB)
- Existem erros lógicos de configuração no projeto recebido do MasterTool IEC XE
- O outro CP está no estado Ativo, e a versão de firmware deste CP é incompatível com a versão de firmware do CP Ativo
- O outro CP está em estado Ativo, e o projeto deste CP é diferente do projeto do CP Ativo. Além de ir para o estado Não-Configurado, uma solicitação de configuração é feita. Desta maneira, depois que os projetos forem sincronizados, o CP sai automaticamente do estado Não-Configurado para o estado Inicializando. Esta condição não é avaliada se a sincronização automática de projetos estiver desabilitada (ver seção [Desabilitação da Sincronização de Projetos](#))

6.3.17.3. Transição 3 - Inicializando para Inativo

- Módulo NX4010 não detectado no barramento, ou falha no seu microprocessador
- Algum dos canais de sincronismo (NETA ou NETB) está em falha, e este CP sabe que esta falha foi causada por componentes de hardware ou software internos (falhas internas de NETA ou NETB)
- O outro CP está em estado Ativo. No entanto, não é possível sincronizar os dados redundantes ou a lista de forçamentos redundantes
- O estado do outro CP não pode ser descoberto via NETA / NETB, mas este CP consegue monitorar tráfego em alguma das redes PROFIBUS configurada em modo de falha vital. Desta maneira, parece que o outro CP está controlando o processo, embora NETA / NETB não estejam funcionando para ter certeza disso
- Na redundância sem painel PX-2612 e sem rede PROFIBUS, caso o estado do outro CP não pode ser descoberto via NETA / NETB, mas este CP está recebendo informação que o estado do outro CP é Ativo, através de pacotes de Keep Alive recebidos via NET1 ou NET2 da UCP NX3030
- Houve perda de link em uma interface Ethernet configurada como falha vital.

6.3.17.4. Transição 4 - Inicializando para Ativo

- O outro CP encontra-se em estado Não-Ativo. Antes de fazer esta transição, esta condição deve permanecer verdadeira durante algum tempo. Quando o CPA e CPB são energizados simultaneamente, o CP que terminar a inicialização do sistema antes assume como Ativo
- O estado do outro CP não pode ser descoberto via NETA / NETB, e além disso este CP não consegue monitorar tráfego em nenhuma das redes PROFIBUS configuradas em modo de falha vital, ou então não possui nenhuma rede PROFIBUS configurada em modo de falha vital. Portanto, parece realmente que o outro CP está desligado ou fora de execução. Quando utilizando projeto com painel PX2612, além de chavear para Ativo, este CP desligará o outro CP utilizando seu relé do PX2612. Esta condição deve se manter durante algum tempo antes de fazer esta transição

6.3.17.5. Transição 5 - Inicializando para Reserva

- O outro CP encontra-se em estado Ativo e os serviços de sincronização de dados redundantes e de sincronização da lista de forçamentos redundantes estão funcionando corretamente

6.3.17.6. Transição 6 - Inativo para Não-Configurado

- Este CP foi desligado ou reinicializado (Reset a Quente, Reset a Frio ou Reset Origem) ou sua UCP entrou em Modo Stop
- O botão STAND-BY foi pressionado no PX2612 ou o comando de ir para Reserva foi executado nos diagnósticos da redundância. Além de ir para o estado Não-Configurado, uma solicitação de configuração é feita. Desta maneira, o CP sai automaticamente do estado Não-Configurado para o estado Inicializando. O usuário tipicamente aperta este botão depois de reparar uma falha que levou o CP anteriormente ao estado Inativo
- Este CP está com a sincronização desabilitada e o projeto está diferente do CP Ativo, ao pressionar o botão STAND-BY, o CP vai de inativo para Não-Configurado

6.3.17.7. Transição 7 - Ativo para Não-Configurado

- Este CP foi desligado ou reinicializado (Reset a Quente, Reset a Frio ou Reset Origem) ou sua UCP entrou em Modo Stop

6.3.17.8. Transição 8 - Ativo para Inativo

- Módulo NX4010 não detectado no barramento, ou falha no seu microprocessador. Além disso, este CP sabe que o outro CP estava em estado Reserva antes desta falha ocorrer. Esta condição não é avaliada nos primeiros 2 segundos em estado Ativo
- Este CP perdeu comunicação com o outro CP via NETA e NETB devido a uma falha interna, mas sabe que o outro CP estava em estado Reserva pouco antes desta falha ocorrer. Esta condição não é avaliada nos primeiros 2 segundos em estado Ativo
- Este CP não consegue controlar todas as redes PROFIBUS configuradas em modo de falha vital, e sabe que o outro CP está em estado Reserva. Esta condição não é avaliada nos primeiros 2 segundos em estado Ativo
- Este CP detectou falha total nas redes Ethernet configuradas em modo de falha vital, e sabe que o outro CP está em estado Reserva

6.3.17.9. Transição 9 - Ativo para Reserva

- Os dois CPs, por algum motivo, estão no estado Ativo, e este conflito deve ser resolvido. O CPA chaveará para o estado Reserva caso este conflito dure certo tempo, e o CPB fará o mesmo após um tempo diferente, que é menor do que o tempo do CPA. Desta maneira, em caso de conflito, o CPA tem prioridade para continuar em estado Ativo. Caso o estado do outro CP seja desconhecido o CP Ativo não irá para o estado Reserva. Isso acontece no caso do projeto redundante não utilizar painel PX2612
- O botão STAND-BY foi pressionado, ou o comando de ir para Reserva foi executado nos diagnósticos da redundância, e este CP sabe que o outro CP encontra-se no estado Reserva. Esta condição não é avaliada nos primeiros 2 segundos em estado Ativo

6.3.17.10. Transição 10 – Reserva para Não-Configurado

- Este CP foi desligado ou reinicializado (Reset a Quente, Reset a Frio ou Reset Origem) ou sua UCP entrou em Modo Stop
- O outro CP encontra-se em estado Ativo e o projeto deste CP é diferente do projeto do CP Ativo. Além de ir para o estado Não-Configurado, uma solicitação de configuração é feita. Desta maneira, depois que os projetos forem sincronizados, o CP sai automaticamente do estado Não-Configurado para o estado Inicializando. Esta condição não é avaliada se a sincronização automática de projetos estiver desabilitada (ver seção [Desabilitação da Sincronização de Projetos](#))
- O outro CP está no estado Ativo, e a versão de firmware deste CP é incompatível com a versão de firmware do CP Ativo

6.3.17.11. Transição 11 – Reserva para Inativo

- Módulo NX4010 não detectado no barramento, ou falha no seu microprocessador
- O botão INACTIVE foi pressionado no PX2612 ou o comando de ir para Inativo foi executado nos diagnósticos da redundância. Isto é feito tipicamente para executar uma manutenção programada no CP Não-Ativo. Deve-se evitar fazer manutenções programadas no CP Reserva, por isso, antes é aconselhável chaveá-lo para Inativo
- O outro CP encontra-se em estado Ativo. No entanto, o serviço de sincronização de dados redundantes não funcionou corretamente nos últimos quatro ciclos da MainTask, ou o serviço de sincronização de diagnósticos redundantes não funcionou corretamente nos últimos dois ciclos da mesma
- O outro CP encontra-se em estado Ativo. No entanto, este CP não consegue monitorar tráfego PROFIBUS em todas as redes configuradas em modo de falha vital
- O outro CP encontra-se em estado Ativo. No entanto, este CP detectou falha nas portas Ethernet configuradas em modo de falha vital

6.3.17.12. Transição 12 – Reserva para Ativo

- O estado do outro CP é desconhecido devido a falhas em NETA e NETB. Neste caso, além de ir para o estado Ativo, por segurança este CP desliga o outro CP usando o relé do PX2612. Quando a Redundância não usa painel PX2612 e não existe rede PROFIBUS DP, o CP utiliza um mecanismo de Keep Alive pelas portas NET1 / NET2 da NX3030, para intercomunicar o estado entre os CPs e detectar que o CP Ativo não está mais controlando o processo.
- O estado do outro CP é conhecido e diferente de Ativo

6.3.18. Primeiros Instantes em Estado Ativo

Caso a aplicação utilize o Painel PX2612, nos primeiros 2 segundos em estado Ativo, conforme já foi descrito na seção [Funções do Painel de Comando de Redundância PX2612](#), o LED ACTIVE pisca, e só depois deste tempo fica aceso. Nos casos em que não é utilizado painel PX2612 este comportamento pode ser observado nos diagnósticos da redundância, na variável que reflete o estado do LED.

Nesta condição, diversas transições que normalmente poderiam tirar o CP do estado Ativo não são avaliadas (ver subseções anteriores que definem transições a partir do estado Ativo). Por exemplo, durante este tempo, não adianta apertar o botão STAND-BY ou o comando de ir para Reserva nos diagnósticos da redundância para tentar fazer o CP ir para o estado Reserva.

Apenas duas condições permitem que o CP saia do estado Ativo deste estado transitório. Estas condições são as seguintes:

- Este CP foi desligado ou reinicializado (Reset a Quente, Reset a Frio ou Reset Origem) ou sua UCP entrou em Modo Stop, causando transição para Não-Configurado
- Os dois CPs, por algum motivo, estão no estado Ativo, e este conflito deve ser resolvido. O CPA chaveará para o estado Reserva caso este conflito dure certo tempo, e o CPB fará o mesmo após um tempo diferente, que é menor do que o tempo do CPA. Desta maneira, em caso de conflito, o CPA tem prioridade para continuar em estado Ativo

Além disso, nos primeiros instantes que um CP assume o estado Ativo, alguns diagnósticos que não são redundantes podem não ser válidos, como, por exemplo, os diagnósticos dos módulos NX5000 e NX5001. O método para não considerar estes diagnósticos possivelmente “inválidos” é descrito na seção [Leitura de Diagnósticos Não-Redundantes](#).

6.3.19. Falhas mais Comuns Causadoras de Switch-overs Automáticos entre Half-Clusters

Nesta seção, são listadas as falhas mais comuns que, de forma automática, causam um switch-over do CP Ativo para Não-Ativo, e do CP Reserva para Ativo. Estas falhas disparam um sub-conjunto daquelas transições examinadas na seção *Transições entre Estados de Redundância*.

- Falta de alimentação no CP Ativo. É importante que os dois CPs tenham sistemas de alimentação redundantes, para que uma falha comum na alimentação não afete também o CP Reserva
- Falha na fonte de alimentação NX8000 do CP Ativo
- Falha no barramento do bastidor (NX9000, NX9001, NX9002 ou NX9003) do CP Ativo
- Falhas na UCP NX3030 do CP Ativo, tais como:
 - Cão-de-guarda
 - Reinicialização (Reset a Quente, Reset a Frio ou Reset Origem)
 - Parada (Stop)
 - Falha nas interfaces de barramento em um ou ambos os canais de sincronismo NETA e NETB
- Falhas no NX4010 do CP Ativo, tais como:
 - Módulo não reconhecido no barramento pela UCP NX3030
 - Falha no microprocessador do NX4010, que impede atualização dos diagnósticos internos de NETA / NETB e do painel de controle PX2612 (botões, LEDs e relé)
 - Falha internas que afetam um ou ambos os canais de sincronismo NETA e NETB
- Falha total de uma rede PROFIBUS no CP Ativo, caso esta rede esteja configurada em modo vital. Caso a rede PROFIBUS seja redundante, ambas as redes que a compõem devem estar em falha (falha dupla)
- Falha total de uma rede Ethernet no CP Ativo, caso esta rede esteja configurada com vital. Caso a rede Ethernet seja redundante, ambas as redes que a compõem devem estar em falha (falha dupla)

6.3.20. Falhas Associadas a Switch-overs entre Half-Clusters Gerenciados pelo Usuário

Entre as transições examinadas na seção [Transições entre Estados de Redundância](#), algumas possibilitam que o usuário gerencie switch-overs entre half-clusters, devido a falhas que normalmente não geram switch-overs de forma automática.

Existem casos muito particulares, que dependem da filosofia de cada cliente. Considere um exemplo em que o sistema SCADA perde comunicação com o CP Ativo, mas mantém-se comunicando com o CP Reserva.

Alguns clientes poderão preferir um switch-over manual, onde o operador pressiona o botão STAND-BY do PX2612, para o CP Ativo. O switch-over provoca a retomada da comunicação com o novo CP Ativo.

Uma solução alternativa seria provocar o switch-over enviando um comando do SCADA para o CP Reserva, que o repassaria para o CP Ativo via NETA / NETB, usando as estruturas de dados RedCmdLocal (CP Reserva) e RedCmdRem (CP Ativo) para transportar um comando equivalente ao botão STAND-BY do PX2612.

Também seria possível o próprio CP Ativo detectar sua perda de comunicação com o SCADA, e ativar um comando em RedCmdLocal, equivalente ao botão STAND-BY do PX2612. Esta seria uma solução totalmente automática e sem intervenção do operador, que tipicamente seria implementada na POU ActivePrg.

Através de estruturas de dados como aquelas citadas na seção [Estruturas de Dados de Diagnósticos, Comandos e Usuário](#), é possível trocar diagnósticos e comandos entre os half-clusters via NETA e NETB e, desta maneira, o usuário pode executar gerenciamentos especiais de redundância, para falhas que normalmente não causariam switch-overs. Maiores detalhes sobre estas estruturas de dados serão fornecidos nas seções:

- [Estrutura de Diagnósticos da Redundância](#)
- [Comandos da Redundância](#)
- [Informações do Usuário Trocadas entre CPA e CPB](#)

Abaixo é exemplificado como o usuário pode gerenciar falhas e executar um switch-over devido a um erro nas interfaces Ethernet do CP Ativo (o código deve ser utilizado na POU ActivePrg):

```
//Verifica se NIC Teaming está habilitado ou não.
IF ((DG_NX3030.tDetailed.Ethernet.NET1.szIP = '0.0.0.0') OR (DG_NX3030.tDetailed
.Ethernet.NET2.szIP = '0.0.0.0')) THEN
//NIC Teaming habilitado: erro nas duas NETs para executar switch-over.
IF (DG_NX3030.tDetailed.Ethernet.NET1.bLinkDown AND DG_NX3030.tDetailed.Ethernet
.NET2.bLinkDown) THEN
//Coloca o CP Local em Reserva.
DG_NX4010.tRedundancy.RedCmdLoc.bStandbyLocal := TRUE;
END_IF
ELSE
//NIC Teaming desabilitado: erro em uma NET para executar switch-over.
IF (DG_NX3030.tDetailed.Ethernet.NET1.bLinkDown OR DG_NX3030.tDetailed.Ethernet.
NET2.bLinkDown) THEN
//Coloca o CP Local em Reserva.
DG_NX4010.tRedundancy.RedCmdLoc.bStandbyLocal := TRUE;
END_IF
END_IF
```

ATENÇÃO

Quando duas interfaces formarem um par NIC Teaming, a interface inativa sempre terá o endereço de IP 0.0.0.0. Este não é um endereço de IP válido e não é possível configurar manualmente uma interface com este endereço.

6.3.21. Tolerância a Falhas

O objetivo principal de um CP redundante é o aumento da disponibilidade do sistema. A disponibilidade é a razão entre o tempo em que o sistema está funcionando corretamente e o tempo total desde a implantação do sistema. Por exemplo, se um sistema foi implantado há 10 anos, e durante este período esteve parado no tempo total de um ano devido a falhas, então sua

disponibilidade foi de apenas 90%. Disponibilidades desta ordem são geralmente inaceitáveis para sistemas críticos, sendo que valores da ordem de 99,99% ou ainda superiores podem ser solicitados nestes sistemas.

Para atingir disponibilidades desta ordem, são necessárias diversas estratégias:

- Utilização de componentes mais confiáveis (com alto MTBF, ou tempo médio entre falhas), o que contribuirá para o aumento do MTBF do sistema como um todo
- Utilização de redundância pelo menos para os componentes mais críticos ou componentes com menor MTBF, de tal maneira que a falha de um componente possa ser tolerada sem parar o sistema. Se a redundância for implementada através da duplicação de componentes, será necessário que os dois falhem para que o sistema como um todo fique indisponível
- Alta cobertura de diagnósticos, em especial de componentes redundantes. A redundância de componentes é pouco útil para o aumento da disponibilidade quando não se pode descobrir que um componente redundante falhou. Neste caso, a primeira falha em um dos componentes ainda não compromete o sistema, mas por permanecer oculta, algum dia acontecerá a segunda falha que comprometerá o sistema, já que a primeira falha ainda não foi reparada. As falhas podem ser classificadas entre diagnosticáveis e ocultas, sendo altamente desejável que todas as falhas de componentes redundantes sejam diagnosticáveis
- Também é importante que componentes não-redundantes tenham ampla cobertura de diagnósticos, pois muitas vezes o sistema pode continuar funcionando mesmo com a falha de um componente não-redundante. O componente pode não estar sendo solicitado. Por exemplo, um relé com contato aberto, que raramente tem sua bobina acionada, não tem sua falha detectada até o momento em que o sistema solicitar seu fechamento
- Baixo tempo de reparo de componentes não-redundantes. A falha de um componente não-redundante pode comprometer o sistema, e durante o reparo, o sistema estará indisponível
- Possibilidade de reparar ou substituir um componente redundante sem parar o sistema. Se esta possibilidade existe, obtém-se um grande aumento de disponibilidade. Do contrário, deve-se programar uma parada do sistema para substituir o componente, e o tempo de reparo conta como tempo indisponível
- Baixo tempo de reparo de componentes redundantes. A falha de um componente redundante não compromete o sistema, mas durante seu reparo, eventualmente pode ocorrer uma falha em seu par redundante. Por este motivo, é importante que a falha seja reparada brevemente, após diagnosticada. Quanto maior o tempo de reparo, maior a probabilidade de acontecer uma segunda falha no componente redundante durante o reparo da primeira falha, o que comprometeria o sistema. Portanto, quanto maior o tempo de reparo, menor a disponibilidade do sistema
- Programar testes offline periódicos em componentes para detectar falhas não diagnosticáveis automaticamente pelo sistema. O objetivo é detectar falhas ocultas, especialmente em componentes redundantes ou componentes simples que não estejam sendo solicitados (por exemplo, um relé de segurança). Testes offline às vezes implicam em paradas no sistema, o que diminui a disponibilidade. Normalmente aproveita-se ocasiões especiais, tais como paradas programadas de manutenção da planta. Quanto maior o período entre testes offline, maior o tempo em que uma falha pode permanecer oculta, e portanto maior a probabilidade de uma falha comprometer o sistema, ou seja, menor a disponibilidade do sistema

Estes princípios foram considerados no projeto de CPs redundantes com UCP NX3030.

As próximas subseções analisam diversos tipos de falhas e como são toleradas ou não toleradas, e se existem switch-overs associados às falhas toleradas.

6.3.21.1. Falhas Simples com Indisponibilidade

Alguns componentes, por não serem duplicados, não toleram nem sequer falha simples sem causar algum tipo de indisponibilidade. Em um CP redundante com UCP NX3030, trata-se dos seguintes componentes:

- Remotas (escravos) PROFIBUS em uma rede PROFIBUS não-redundante
- Remotas (escravos) Ethernet em uma rede Ethernet não redundante
- Módulos de E/S

A intolerância a falhas de uma rede PROFIBUS não-redundante pode ser resolvida optando-se por uma rede PROFIBUS redundante, que é aconselhada em sistemas que demandam alta tolerância a falhas. A Figura 158 mostra um exemplo de arquitetura com uma rede PROFIBUS redundante. Da mesma forma a intolerância a falhas de uma rede Ethernet não redundante pode ser resolvida usando-se uma rede Ethernet redundante com a configuração de NIC Teaming.

Quanto à indisponibilidade de um módulo de E/S, deve-se observar que a mesma não se constitui em uma indisponibilidade total do sistema. Ela se constitui em uma indisponibilidade parcial, somente das malhas de controle que utilizam este módulo de E/S.

Embora não haja previsão de redundância de módulos de E/S, a aplicação do usuário pode gerenciá-la em casos especiais. Por exemplo, o usuário pode inserir três módulos de entradas analógicas em três remotas PROFIBUS diferentes, e implementar um esquema de votação entre triplas entradas analógicas, para algum sistema crítico. No entanto, tais soluções, como foi enfatizado, devem ser gerenciadas pelo usuário. Não existe nenhum suporte automático para as mesmas. Tais soluções, em geral, também implicam na redundância de transdutores e atuadores no campo.

6.3.21.2. Falhas Simples sem Indisponibilidade Causando um Switch-over

Alguns componentes redundantes toleram falhas simples sem causar indisponibilidade, mas causam switch-over:

- Bastidores (NX9000, NX9001, NX9002 ou NX9003)
- Fontes de alimentação (NX8000)
- UCPs (NX3030)
- Módulo NX4010
- Módulos NX5001 (mestres PROFIBUS) em configuração de rede PROFIBUS não-redundante
- Módulos NX5000 (Ethernet) em configurações sem NIC Teaming
- Interface escravo PROFIBUS de uma remota redundante (PO5063V5, PO5065, NX5210 ou AL-3416). Neste caso, diferente dos anteriores, o switch-over acontece dentro da remota, entre as redes PROFIBUS A e B.

ATENÇÃO

No caso de falha da UCP NX3030 ou do módulo NX4010 em arquiteturas onde não é utilizado painel PX2612 nem rede PROFIBUS, o CP irá permanecer no estado atual. Neste caso, se a falha acontece no half-cluster Ativo, ocorre indisponibilidade do sistema.

6.3.21.3. Falhas Duplas sem Indisponibilidade Causando um Switch-over

Alguns componentes são duplicados em cada half-cluster, desta maneira, antes de causar um switch-over, é necessário que ambos falhem:

- Módulos NX5001 (mestres PROFIBUS) em configuração redundante, configurados em modo de falha vital.
- Módulos NX5000 (Ethernet) em configurações com NIC Teaming (redundância gerenciada pelo usuário).

6.3.22. Overhead da Redundância

Uma aplicação redundante causa um aumento no tempo de processamento de uma aplicação, quando comparado ao tempo necessário para uma aplicação equivalente não-redundante.

Este tempo adicional deve-se principalmente à execução dos serviços de sincronização cíclicos, descritos na seção [Serviços de Sincronização Cíclicos através de NETA e NETB](#), além de um tempo menor para o próprio gerenciamento da redundância (máquina de estados, etc...). O tempo adicional total, devido à redundância (overhead da redundância), é estimado pelo MasterTool e exibido na janela de Mensagens, após compilar o projeto do CP redundante.

ATENÇÃO

O overhead calculado pelo MasterTool considera uma lista de forçamentos de variáveis redundantes vazia.

Além disso, cabe ao usuário definir um intervalo para a MainTask que acomode:

- O tempo necessário para executar as POU's principais (NonSkippedPrg e ActivePrg). Este tempo normalmente é medido após o desenvolvimento do projeto (descontando o tempo adicional para redundância)
- Tempo necessário para detecção e geração de eventos de pontos internos (por exemplo, a ocorrência de 1000 eventos de pontos analógicos com banda morta no mesmo ciclo pode levar até 30 ms)
- Alguma folga do ciclo de MainTask, para execução de outras tarefas da UCP (sistema operacional, drivers de E/S PROFIBUS, MODBUS, etc...). O percentual desta folga pode variar de acordo com o desempenho solicitado destas outras tarefas. Por exemplo, se a comunicação MODBUS com o sistema SCADA precisa alocar muito processamento para atingir um desempenho satisfatório, esta folga deverá ser aumentada

ATENÇÃO

Dependendo do alinhamento de memória, o número de bytes utilizados no cálculo do overhead de redundância poderá ser maior do que o total de bytes declarados nas variáveis.

6.4. Programação de um CP Redundante

6.4.1. Assistente para Criação de um Novo Projeto Redundante

Para criar um novo projeto redundante, deve-se utilizar o comando Arquivo / Novo Projeto, e selecionar o Projeto Master-Tool Padrão.

Inicialmente, o usuário deve informar o nome que deseja dar ao projeto e o diretório onde deseja armazená-lo, conforme mostra a figura abaixo.

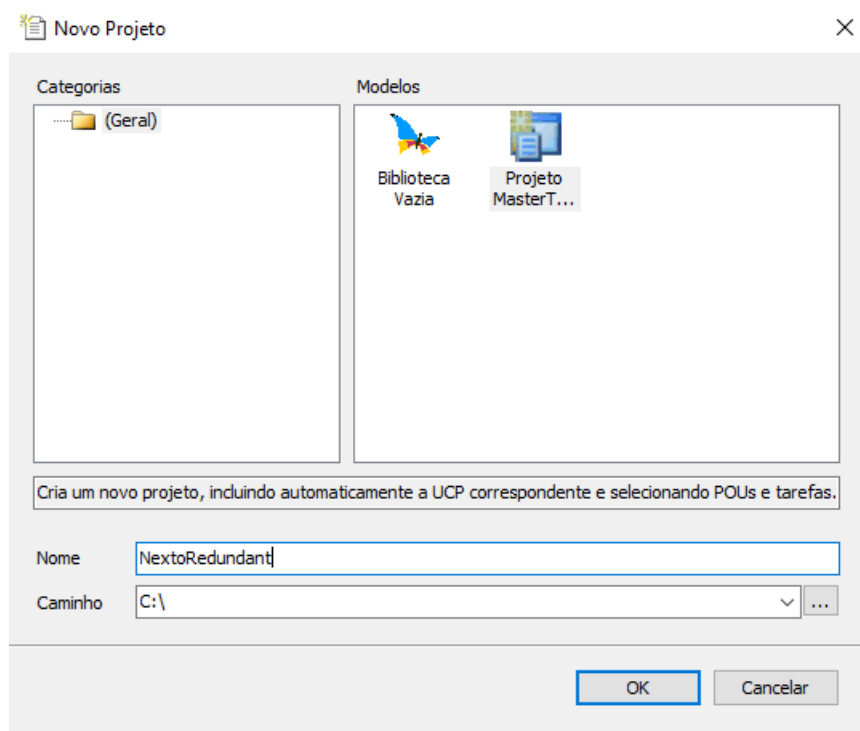


Figura 172: Novo Projeto

A seguir, o Assistente que gera o projeto de redundância realiza alguns questionamentos ao usuário quanto à configuração desejada, os quais devem ser respondidos sucessivamente.

O primeiro ponto a ser definido é a configuração inicial de hardware do half-cluster:

- Selecione a categoria do dispositivo: NX3030 pode ser selecionada em duas categorias Todos os Dispositivos ou em Controladores Modulares
- Selecionar o modelo da UCP: Como a redundância está implementada somente na NX3030, a mesma deve ser selecionada pelo usuário
- Selecionar o modelo do bastidor: Existem quatro opções de bastidores disponíveis e a escolha do mesmo dependerá da quantidade de módulos utilizados na redundância. O MasterTool consiste o tamanho do bastidor de acordo com a quantidade de redes configuradas (próximo item do Assistente)
- Selecionar o modelo da fonte de alimentação
- Selecionar a configuração da redundância. Para um projeto redundante é necessário escolher a opção Com Redundância
- Selecionar o modo de operação da redundância. Neste caso as opções são operação com painel de redundância (PX2612) ou sem ele
- Selecionar se a opção de comunicação OPC DA estará habilitada ou não
- Selecionar se será usada redundância na expansão de barramento

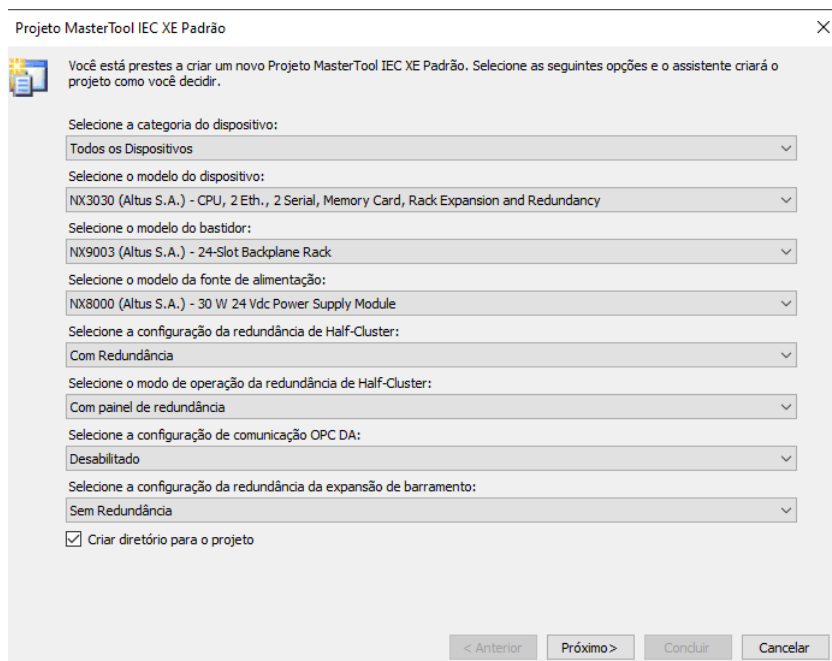


Figura 173: Configuração Inicial de Hardware

Após, o usuário deve definir as redes de comunicação utilizadas na aplicação redundante:

- Selecionar o número de redes PROFIBUS: Pelo Assistente, é possível a criação de até quatro redes PROFIBUS, sendo elas simples ou redundantes. É importante salientar que esta configuração é apenas a indicada pelo Assistente, que somente propõe arquiteturas típicas. Após a conclusão do Assistente, o MasterTool permite a criação de mais redes PROFIBUS, redundantes ou únicas, desde que seja respeitado o limite de quatro módulos em cada half-cluster
- Selecionar o tipo das redes PROFIBUS:
 - Não existe (nenhum módulo NX5001 alocado)
 - Rede Simples (aloca um módulo NX5001)
 - Rede Redundante (aloca dois módulos NX5001)
- Selecione o tipo de rede Ethernet das interfaces da UCP
 - Rede Simples com Modo de Falha Desabilitado (não gera switch-over em caso de falha)
 - Rede Simples com Modo de Falha Habilitado (gera switch-over em caso de falha)
 - Rede Redundante com Modo de Falha Desabilitado (opera em conjunto com a outra interface e não gera switch-over em caso de falha)
 - Rede Redundante com Modo de Falha Habilitado (opera em conjunto com a outra interface e gera switch-over em caso de falha)
- Selecionar o número de redes Ethernet: Nesse caso o Assistente possibilita ao usuário criar até quatro redes simples, ou até duas redes redundantes, ou nenhuma. É importante salientar que esta configuração é apenas a indicada pelo Assistente, que somente propõe arquiteturas típicas. Após a conclusão do Assistente, o MasterTool permite a criação de mais redes Ethernet, desde que seja respeitado o limite de seis módulos em cada half-cluster.
- Selecionar o tipo de rede Ethernet:
 - Não existe (nenhum módulo NX5000 alocado)
 - Rede Simples com Modo de Falha Desabilitado (aloca um módulo NX5000 e não gera switch-over em caso de falha)
 - Rede Simples com Modo de Falha Habilitado (aloca um módulo NX5000 e gera switch-over em caso de falha)
 - Rede Redundante com Modo de Falha Desabilitado (aloca dois módulos NX5000 e não gera switch-over em caso de falha)
 - Rede Redundante com Modo de Falha Habilitado (aloca dois módulos NX5000 e gera switch-over em caso de falha)

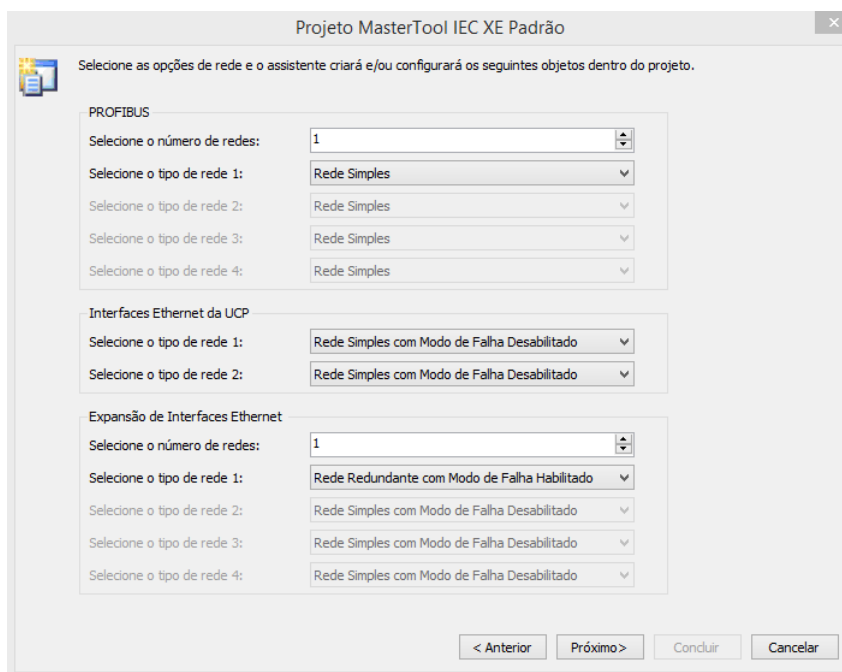


Figura 174: Configuração das Redes de Comunicação

Então, deve ser selecionado o perfil de projeto e a linguagem padrão para a criação dos programas:

- Selecionar o perfil do projeto: Somente é possível utilizar o perfil de projeto simples para a redundância, logo a opção de seleção está desabilitada
- Selecionar a linguagem padrão para todos os programas: A linguagem selecionada pelo usuário será a padrão para todas os programas, porém pode optar por utilizar qualquer outra para uma POU específica

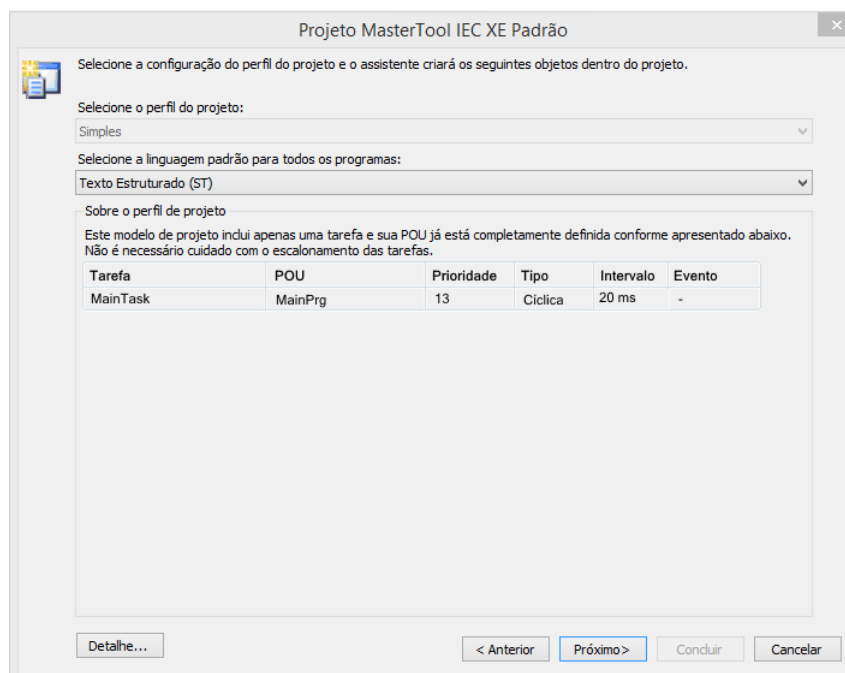


Figura 175: Perfil de Projeto e Linguagem Padrão

Para finalizar, o usuário deve selecionar a linguagem de programas comuns e associados à redundância:

- Programa associado à MainTask (MainPrg): Deverá ser, obrigatoriamente, em linguagem ST, sendo que o MasterTool desabilita as outras opções
- Programas associados às tarefas principais da redundância

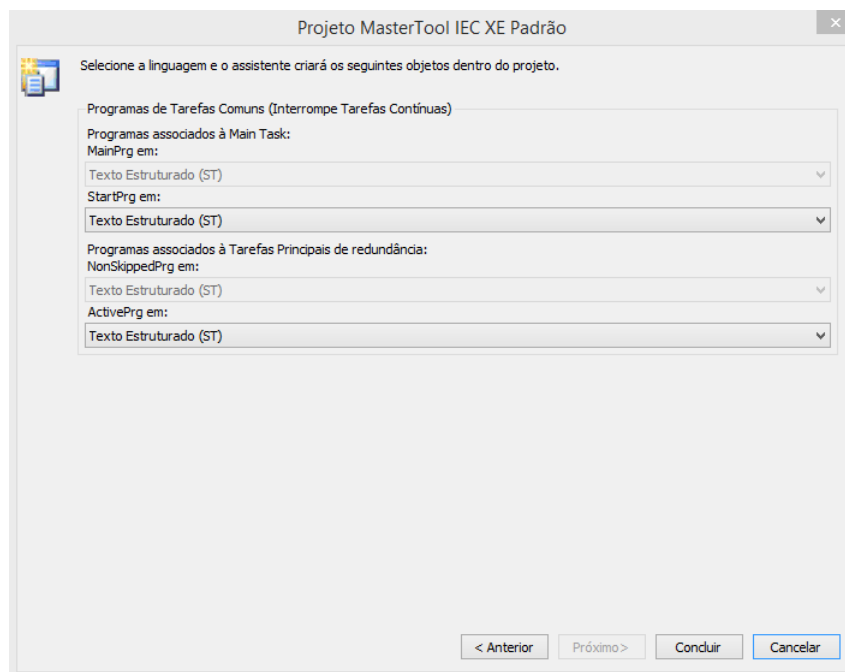


Figura 176: Linguagem dos Programas Específicos

ATENÇÃO

As POU's ActivePrg e NonSkippedPrg são criadas automaticamente, vazias, na linguagem selecionada nas perguntas anteriores. Em POU's criadas manualmente pelo usuário, poderá ser utilizada qualquer uma das linguagens disponíveis, exceto em POU's redundantes, que não podem ser escritas na linguagem SFC, pois esta utiliza o temporizador IEC em segundo plano. Para mais informações, ver a seção [Limitações na Programação de um CP Redundante](#).

ATENÇÃO

A POU MainPrg será sempre gerada automaticamente em linguagem ST, e não pode ser alterada pelo usuário. Esta POU chama as POU's ActivePrg (somente no CP Ativo) e NonSkippedPrg (em ambos os CPs).

Depois de obter respostas para as perguntas anteriores, o Assistente gera o projeto inicial, definindo um half-cluster com a seguinte configuração inicial de hardware:

- Bastidor selecionado
- Fonte de alimentação (posições 0 e 1)
- UCP NX3030 (posições 2 e 3)
- Módulo NX4010 (posições 4 e 5) e Painel PX2612, caso tenha sido selecionado.
- Após o módulo NX4010, são inseridos os módulos NX5001 necessários para implementar a rede PROFIBUS com as características definidas anteriormente pelo usuário
- Após os módulos NX5001, são inseridos os módulos NX5000 necessários para implementar a rede Ethernet com as características definidas anteriormente pelo usuário

6.4.2. Configuração dos Half-Clusters

O Assistente sempre é utilizado para gerar a primeira versão de um projeto redundante. Isso garante que a versão inicial do projeto será gerada rápida e corretamente.

No entanto, é possível que algumas modificações sejam necessárias em um half-cluster, tal como a inserção de novos módulos NX5001 e NX5000, que podem ser feitas alterando a tela de configuração do half-cluster. Os capítulos a seguir mostram como adicionar e/ou configurar os módulos NX5000, NX5001 e NX4010.

Algumas regras e precauções devem ser seguidas para um projeto redundante, conforme descreve a seção seguinte.

6.4.2.1. Configuração Fixa nas Posições 0 a 5 do Bastidor

Nas posições 0 a 5 do bastidor selecionado sempre devem estar instalados os seguintes módulos:

- Fonte de alimentação (posição 0)
- UCP NX3030 (posição 2)
- Módulo NX4010 (posição 4)

Estes módulos não devem ser removidos do projeto original gerado pelo Assistente.

Qualquer configuração diferente nestas posições resultará em um erro notificado pelo MasterTool na compilação do projeto.

6.4.3. Configurações das Portas Ethernet da UCP NX3030 (NET 1 e NET 2)

6.4.3.1. Configuração do Endereço de IP

A figura abaixo mostra as configurações da porta NET 1 da UCP NX3030 (a tela para configuração da porta NET 2 contém um sub-conjunto destes parâmetros). Para abrir esta tela, deve-se dar um duplo clique sobre NET 1 ou NET 2, abaixo da UCP NX3030 na árvore de dispositivos.

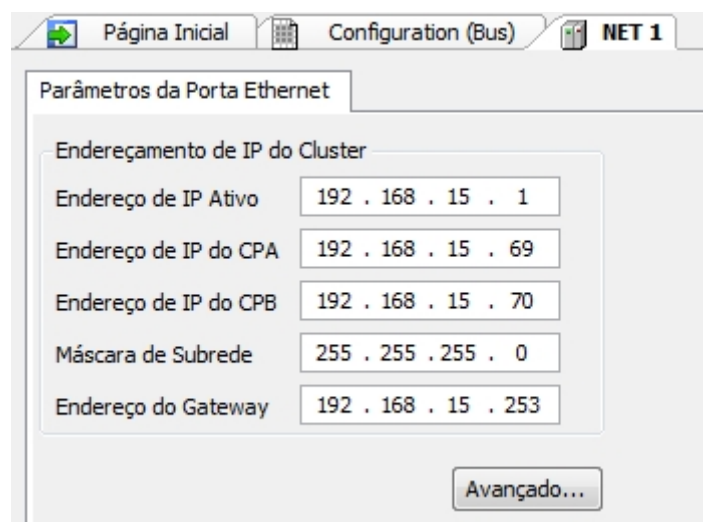


Figura 177: Parâmetros da Porta Ethernet NET 1

A seguir devem ser editados os parâmetros básicos para as interfaces NET 1 e NET 2. O endereçamento será de acordo com o método de troca IP Ativo, conforme está descrito em [Princípios de Funcionamento - Métodos de Troca de IP - IP Ativo](#).

ATENÇÃO

Os dois endereços IP das interfaces NET 1 e NET 2, bem como o Endereço do Gateway, devem pertencer à mesma subrede.

ATENÇÃO

A tela de configuração de NET 2 possui a mesma estrutura da tela de configuração da NET 1, porém não contém a opção de configurações avançadas, ou seja, os parâmetros do NIC Teaming e Falha Ethernet.

6.4.3.2. NIC Teaming entre NET 1 e NET 2

A opção Avançado, na tela de configuração da interface NET 1, abre uma nova tela de configuração, a qual define se NET 1 será redundante. Caso o checkbox de Redundância de Comunicação seja marcado, as interfaces NET 1 e NET 2 formarão um par redundante com NIC Teaming, conforme descrito na seção [Princípios de Funcionamento - Redes Ethernet Redundantes com NIC Teaming](#). Automaticamente, outros parâmetros serão habilitados e deverão ser configurados:

- Período de Teste da Redundância (ms): Período de envio do frame de teste de comunicação entre as duas NETs. Pode ser configurado com valores entre 100 e 9900
- Número de Retentativas do Teste da Redundância: Número máximo de vezes que a NET que enviou o frame irá esperar pela resposta. Pode ser configurado com valores entre 1 e 100
- Período para Chaveamento (s): Tempo máximo que a NET Ativa irá esperar por um pacote qualquer. Pode ser configurado com valores entre 1 e 25

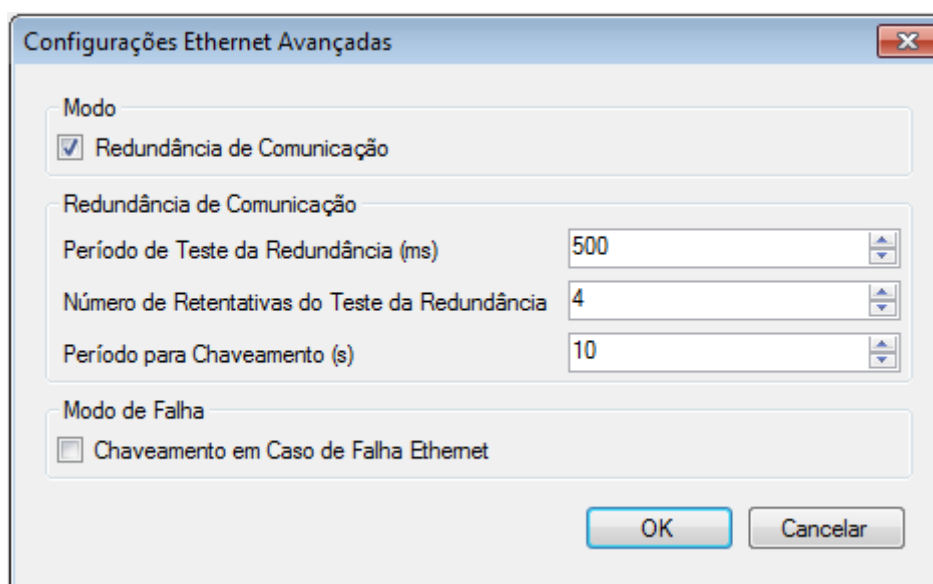


Figura 178: Configurações Avançadas da Ethernet

Caso o tempo de resposta do Teste da Redundância atinja Período de Teste vezes o Número de Retentativas e a interface ativa permaneça por um tempo maior que o Período para Chaveamento sem receber nenhum pacote, ocorrerá um switch-over, tornando ativa a interface que até então estava inativa. É importante ressaltar que há um atraso entre a detecção da falha e ativação da interface inativa devido ao tempo necessário para sua configuração. Tal atraso pode chegar a algumas dezenas de milissegundos.

Quando uma das NETs estiver ativa, esta assumirá o endereço de IP configurado, permanecendo a NET inativa com seus parâmetros de Endereço de IP, Máscara de Subrede e Endereço do Gateway em branco nos diagnósticos da UCP.

ATENÇÃO

Quando um comando de Reset Origem for executado em uma UCP configurada com NIC Teaming nas interfaces frontais NET1 e NET2, apenas a interface que estava ativa antes do comando continuará acessível. Após o comando, a interface acessível poderá ser visualizada no [Menu Informativo e de Configuração da UCP](#).

6.4.3.3. Configuração de falha vital na NET 1 e NET 2

A opção Avançado, na tela de configuração das interfaces NET 1 e NET2, abre uma tela de configuração onde além de habilitar a redundância de comunicação também é possível configurar se a interface irá gerar um switch-over em caso de falha na interface, conforme está descrito em [Princípios de Funcionamento - Uso de Interfaces Ethernet com Indicação de Falha Vital](#).

Quando configurado em conjunto com a redundância NIC Teaming, a falha vital será considerada quando ocorrer falha nas interfaces NET1 e NET2.

6.4.4. Configurações dos Módulos NX5001

6.4.4.1. Inserção ou Remoção de Módulos NX5001

Pode-se inserir ou remover módulos NX5001 no bastidor do half-cluster. Para executar esta operação de forma correta, deve-se ter consciência das seguintes regras:

- O número de módulos NX5001 em cada half-cluster pode variar entre zero e quatro
- Pode-se definir até quatro redes PROFIBUS simples ou duas redes redundantes, respeitando o limite de módulos em cada half-cluster
- Quando uma rede PROFIBUS é simples, ela necessita de um único módulo NX5001 em cada half-cluster. Quando é redundante, ela necessita de 2 módulos NX5001 em cada half-cluster
- Dois módulos NX5001 utilizados para compor uma rede PROFIBUS redundante devem ocupar posições vizinhas no bastidor
- A quantidade de módulos NX5001 no bastidor deve ser compatível com o número de redes PROFIBUS existentes, e com o atributo de redundância de cada rede, ou seja:
 - 0 x NX5001: Nenhuma rede PROFIBUS
 - 1 x NX5001: Uma rede PROFIBUS simples
 - 2 x NX5001: Neste caso há duas opções:
 - Duas redes PROFIBUS simples
 - Uma rede PROFIBUS redundante
 - 3 x NX5001: Neste caso há duas opções:
 - Três redes PROFIBUS simples
 - Uma rede PROFIBUS simples e uma rede PROFIBUS redundante
 - 4 x NX5001: Neste caso há três opções:
 - Quatro redes PROFIBUS simples
 - Duas redes PROFIBUS simples e uma rede PROFIBUS redundante
 - Duas redes PROFIBUS redundantes

Depois de inserir ou remover módulos NX5001, deve-se revisar a configuração dos módulos NX5001 que restaram no bastidor.

6.4.4.2. Ajuste de Parâmetros dos Módulos NX5001

Cada módulo NX5001 utilizado em uma rede PROFIBUS simples, ou cada par redundante de NX5001 utilizado em uma rede PROFIBUS redundante, possui os seguintes parâmetros que devem ser ajustados.

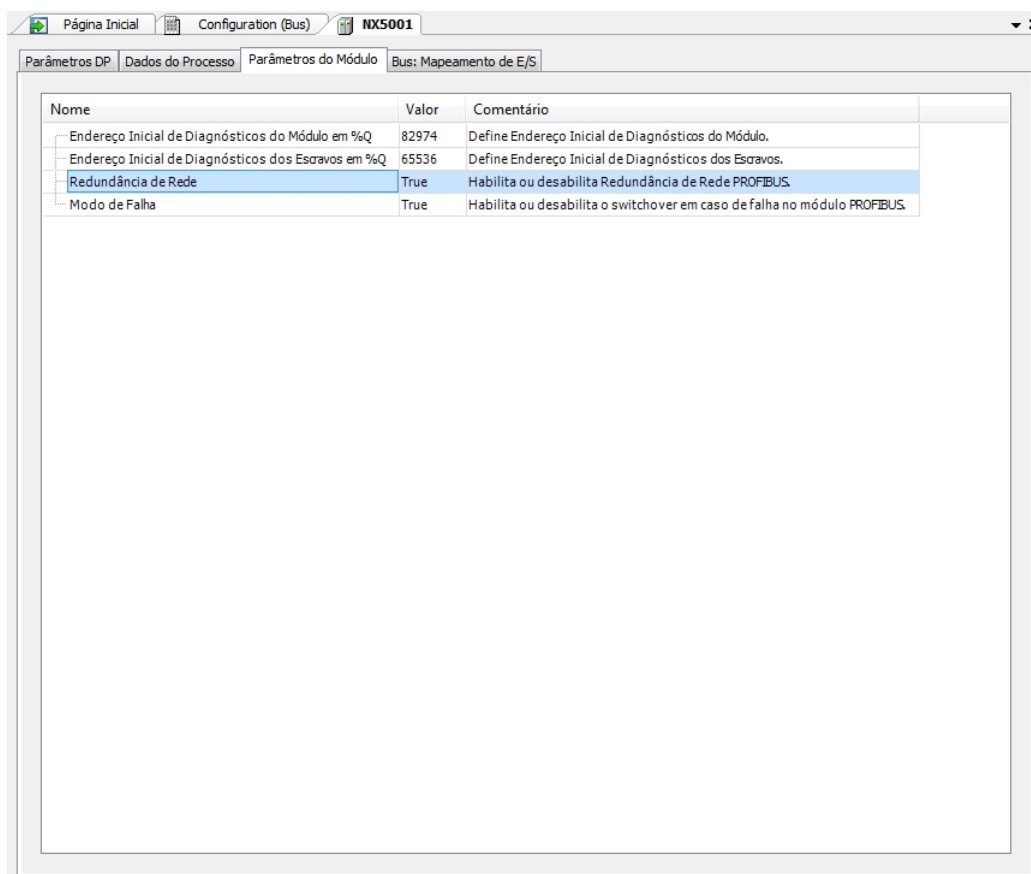


Figura 179: Parâmetro de Redundância NX5001

Para agrupar dois módulos NX5001 em uma rede PROFIBUS redundante, deve-se dar um duplo clique em um módulo NX5001 desagrupado que tenha outro módulo NX5001 também desagrupado à sua direita no bastidor. A seguir deve-se marcar como verdadeiro (TRUE) o parâmetro “Redundância de Rede” disponível na aba “Parâmetros do Módulo”, conforme mostra Figura 179. Para desagrupá-los, basta seguir o mesmo procedimento, porém marcando o parâmetro como falso (FALSE). Se este parâmetro estiver marcado como verdadeiro (TRUE), os parâmetros DP e parâmetros do módulo NX5001 à sua direita estarão bloqueados para edição.

ATENÇÃO

No caso de redes redundantes, deve-se ajustar apenas os parâmetros do NX5001 presente mais a esquerda no barramento, enquanto os parâmetros do NX5001 da direita permanecem bloqueados para edição. Alguns parâmetros de uma rede são idênticos aos da outra, enquanto outros são calculados automaticamente a partir dos parâmetros da rede do NX5001 da esquerda.

É aconselhável que o endereço configurado para um mestre NX5001 em um CP redundante seja 2, pois o endereço do mestre NX5001 no CP Não-Ativo é decrementado em um, portanto o endereço do mestre NX5001 ficaria 1.

Além disso, lembra-se que:

- Os endereços de 3 a 125 são geralmente utilizados para escravos PROFIBUS
- O endereço 0 é frequentemente utilizado para configuração e diagnóstico dos dispositivos
- O endereço 1, em especial, é reservado para ser assumido, dinamicamente, pelo mestre PROFIBUS no CP Não-Ativo (mestre PROFIBUS em modo passivo)
- O endereço 126 é muito utilizado por dispositivos escravos quando saem de fábrica
- O endereço 127 é utilizado para enviar frames em broadcast

Na próxima compilação do projeto, o MasterTool verificará possíveis erros que o usuário possa ter cometido ao inserir ou remover módulos NX5001 manualmente.

Importante salientar que, durante a execução de um projeto redundante configurado com módulos NX5001, o bit 0 de Comandos (Canal Enable Interface %QXn.0 na aba Bus: Mapeamento de E/S) é manipulado pela aplicação redundante. As

interfaces devem permanecer habilitadas durante toda a execução do programa. Desta forma, um comando executado pelo usuário para desabilitar uma interface não será executado da forma que pode se esperar. Por exemplo: se uma interface tiver o estado deste bit alterado de TRUE para FALSE em um CP Ativo, isso não será interpretado como uma falha que levaria o CP Ativo para o estado Inativo. Neste caso, o CP Ativo permanecerá Ativo e o outro CP que irá para o estado Inativo. Por estas razões, este bit de comando não deve ser manipulado pelo usuário em uma aplicação redundante.

Maiores detalhes sobre a configuração de redes PROFIBUS são obtidos no Manual de utilização Mestre PROFIBUS DP NX5001.

6.4.4.3. Configurações de Remotas PROFIBUS

Para configurar remotas PROFIBUS abaixo de um mestre NX5001, deve-se consultar, além do Manual de Utilização Mestre PROFIBUS DP NX5001, os seguintes manuais:

- Manual de Utilização da Série Ponto
- Manual de Utilização Cabeça PROFIBUS PO5063V1 e Cabeça Redundante PROFIBUS PO5063V5
- Manual de Utilização Cabeça PROFIBUS PO5064 e Cabeça Redundante PROFIBUS PO5065
- Manual de Utilização Rede HART sobre PROFIBUS

Para um sistema redundante deve-se atentar à configuração do parâmetro de cão-de-guarda da remota. Caso, na tela de configuração de remota, esteja habilitado o checkbox “Controle de Cão-de-Guarda”, o campo “Tempo” deverá ser configurado corretamente. Existem duas opções de configuração e o tempo utilizado deve ser o maior tempo entre:

- $WT \geq I \times 2 + 500\text{ms}$; e
- $WT \geq I \times 3$;

Onde WT é o tempo de cão-de-guarda e I é o intervalo configurado da MainTask.

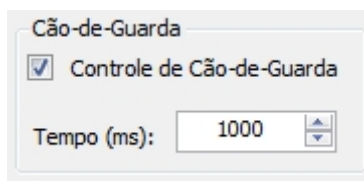


Figura 180: Configuração de Cão-de-Guarda de uma Remota PROFIBUS

6.4.5. Configurações dos Módulos NX5000

6.4.5.1. Inserção ou Remoção de Módulos NX5000

Pode-se inserir ou remover módulos Ethernet NX5000 no bastidor do half-cluster. Para executar esta operação de forma correta, deve-se ter consciência de que o número de módulos NX5000 em cada half-cluster pode variar entre zero e seis. Deve-se atentar também ao fato de que módulos que formarão um par redundante NIC Teaming devem ser inseridos em posições vizinhas no bastidor.

Na próxima compilação do projeto, o MasterTool verificará possíveis erros que o usuário possa ter cometido ao inserir ou remover módulos NX5000 manualmente. Por exemplo, se o usuário inseriu mais do que 6 módulos NX5000, haverá um erro.

A interface de cada módulo será identificada como NET 1, assim como são identificadas na mecânica do produto. Caso o usuário acrescente manualmente módulos NX5000 no barramento, a identificação ocorre da mesma forma que no Assistente.

Depois de inserir ou remover módulos NX5000, deve-se revisar a configuração dos módulos NX5000 que restaram no bastidor.

6.4.5.2. Configuração dos Módulos NX5000

Para cada módulo NX5000 em um CP redundante, devem ser ajustados os parâmetros de endereço, conforme descrito na seção [Princípios de Funcionamento - Métodos de Troca de IP](#), os quais podem ser acessados através de um duplo clique sobre a interface NET 1, abaixo de cada módulo NX5000 localizado na árvore de dispositivos.

ATENÇÃO

Caso dois módulos consecutivos formem um par redundante NIC Teaming, configura-se os parâmetros básicos somente do NX5000 da esquerda, e a edição dos parâmetros do NX5000 da direita estará bloqueada.

6.4.5.3. Agrupamento de Módulos NX5000 com Redundância NIC Teaming

Os módulos NX5000, assim como a interface NET 1 das UCPs NX3020 e NX3030, apresentam uma tela de configurações avançadas, que definirá se o módulo formará um par NIC Teaming Redundante com o módulo a sua direita. A configuração é feita conforme já descrito na seção [NIC Teaming entre NET 1 e NET 2](#).

Para agrupar dois módulos NX5000 como um par redundante, a seguinte condição deve existir:

- Estes dois módulos NX5000 devem ocupar posições vizinhas no bastidor.

Ao fazer isto, o módulo à direita tem a edição de seus parâmetros bloqueada e os parâmetros editados no módulo à esquerda passam a ser comuns para os dois módulos. Desmarcando o checkbox “Redundância de Comunicação” do módulo à esquerda, provoca-se a separação dos módulos, que voltam a se comportar como módulos individuais sem redundância NIC Teaming.

6.4.5.3.1. Configuração de Falha Vital

Os módulos NX5000, assim como as interfaces NET 1 e NET 2 permitem configurar se a interface irá gerar um switch-over em caso de falha na interface, conforme está descrito em [Princípios de Funcionamento - Uso de Interfaces Ethernet com Indicação de Falha Vital](#).

Quando configurado em conjunto com a redundância NIC Teaming a falha vital será considerada quando ocorrer falha nas dois módulos do par redundante.

6.4.6. Configurações do Módulo NX4010

As configurações relacionadas às variáveis %I, %Q e %M redundantes podem ser acessadas através de um duplo clique sobre o módulo NX4010, seguido da seleção da aba “Parâmetros de Redundância”.

Para entender estes parâmetros, deve-se ler as seções [Variáveis %I Redundantes e Não-Redundantes](#), [Variáveis %Q Redundantes e Não-Redundantes](#) e [Variáveis %M Redundantes e Não-Redundantes](#).

Os seguintes parâmetros devem ser configurados:

Configuração	Descrição	Padrão	Opções
	Memória (%M)		
Offset da Memória %M Redundante	Endereço inicial da memória %M redundante	0	0 (desabilitado)
Tamanho da Memória %M Redundante	Tamanho da memória %M redundante	0	0 a 65536
	Memória (%I)		
Offset da Memória %I Redundante	Endereço inicial da memória %I redundante	0	0 (desabilitado)
Tamanho da Memória %I Redundante	Tamanho da memória %I redundante	16384	0 a 81920
	Memória (%Q)		
Offset da Memória %Q Redundante Reservado Para Drivers de E/S	Endereço inicial da memória %Q redundante reservado para drivers de entrada e saída	0	0 (desabilitado)
Tamanho da Memória %Q Redundante Reservado para Drivers de E/S	Tamanho da memória %Q redundante reservado para drivers de entrada e saída	16384	0 a 81920
Offset da Memória %Q Redundante Reservada para Diagnósticos	Endereço inicial da memória %Q redundante reservada para a área de diagnósticos	65536	0 a 81919

Configuração	Descrição	Padrão	Opções
Tamanho da Memória %Q Redundante Reservada para Diagnósticos	Tamanho da memória %Q redundante reservada para a área de diagnósticos	16384	0 a 81920

Tabela 203: Parâmetros NX4010

6.4.7. Configurações de Drivers de E/S

A configuração de drivers de E/S, em princípio, não apresenta diferenças em relação a um CP não-redundante.

O que se pode perceber é que alguns drivers de E/S possuem comandos que permitem seu uso em um CP redundante, mas isto não implica em diferenças de configuração dos mesmos. Estes comandos normalmente devem ser executados no programa NonSkippedPrg. Por exemplo, um driver mestre MODBUS RTU em uma rede serial RS-485 deve ser desabilitado em um CP em estado Não-Ativo, usando código inserido pelo usuário em NonSkippedPrg. Mais informações sobre como administrar um driver MODBUS em um sistema redundante podem ser encontradas na seção [Gerenciamento de Instâncias MODBUS em Sistemas Redundantes](#).

No caso da rede PROFIBUS, também existem comandos especiais diferenciados para os CPs em estado Ativo e Não-Ativo. Neste caso, no entanto, o gerenciamento da redundância executa tais comandos de forma automática, sem que seja necessário qualquer gerenciamento por parte do usuário.

Para configuração de E/S remotas PROFIBUS, incluindo remotas e módulos de E/S, deve-se consultar o Manual de utilização Mestre PROFIBUS DP NX5001, além da seção [Configurações dos Módulos NX5001](#) deste manual.

6.4.8. Configurações da MainTask

A tela de configurações associadas à única tarefa de um CP redundante, denominada MainTask, a qual é cíclica, pode ser acessada clicando sobre MainTask na árvore de dispositivos.

Dois parâmetros devem ser ajustados nesta tela:

- Intervalo da MainTask
- Tempo de Cão-de-Guarda

Além disso, uma estimativa do tempo necessário para gerenciamento da redundância é calculada pelo MasterTool. Tal estimativa só é confiável depois que o projeto estiver completo, com todas as POU's desenvolvidas, e áreas de memória redundantes definidas. Diversas considerações devem ser feitas para ajustar adequadamente o intervalo da MainTask:

- O intervalo deve ser suficientemente baixo para controlar o processo efetivamente, observando os tempos de resposta de todos os laços de controle
- O intervalo deve ser suficientemente alto para acomodar, no mínimo, a soma dos dois seguintes tempos:
 - O tempo máximo de execução das POU's NonSkippedPrg e ActivePrg, em conjunto
 - O tempo necessário para gerenciar a redundância (overhead da redundância)
- Além disso, o intervalo deve ter uma folga adicional, necessária para que outros processos tenham tempo de ser executados (comunicação PROFIBUS, comunicação Ethernet com sistemas SCADA, etc...)

O MasterTool tem condições de calcular o tempo necessário para gerenciar a redundância (overhead de redundância), depois que o projeto estiver terminado (todas as POU's desenvolvidas, e áreas de memória redundantes definidas).

Quanto ao tempo máximo de execução das POU's NonSkippedPrg e ActivePrg, é possível medir os tempos depois que estas POU's forem desenvolvidas. Inicialmente, o MasterTool estima 10 ms para o tempo máximo destas duas POU's em conjunto, mas o usuário deverá revisar este tempo posteriormente, em função de medidas executadas usando o projeto final.

Após cada compilação do projeto, o MasterTool soma o overhead de redundância calculado com o parâmetro que informa os tempos das POU's (NonSkippedPrg e ActivePrg), e verifica se a folga mínima parametrizada está sendo obedecida.

Por exemplo:

- Parâmetros configurados na tela da MainTask
 - intervalo da MainTask: 100 ms
 - Tempo estimado para POU's NonSkippedPrg + ActivePrg: 10 ms
 - Folga Mínima: 30%
- Overhead calculado para redundância: 50 ms

Neste caso, o tempo total utilizado é de 60 ms (10 ms + 50 ms), o que consiste em 60% do ciclo da MainTask (100 ms). Desta forma, a folga é de 40%, e portanto a folga mínima de 30% está sendo respeitada.

ATENÇÃO

Um erro de compilação pode ser gerado caso a folga mínima não seja respeitada. Desde que esteja configurado nos Parâmetros do Projeto da UCP.

ATENÇÃO

Havendo sucesso ou não na compilação, o MasterTool informa se a folga é respeitada e o cálculo do overhead de redundância na janela de mensagens.

6.4.8.1. Programa ActivePrg

Nesta POU o usuário deve criar a aplicação principal, responsável pelo controle de seu processo. Esta POU é chamada pela POU principal (MainPrg), sendo executada apenas no CP Ativo.

O usuário pode também criar POUs adicionais (programa, funções ou bloco funcional), e chamá-las ou instanciá-las dentro da POU ActivePrg, para fins de estruturação de seu programa. Também é possível chamar funções e instanciar blocos funcionais definidos em bibliotecas.

Deve-se lembrar que todas as variáveis simbólicas definidas na POU ActivePrg, assim como instâncias de blocos funcionais definidas, serão variáveis redundantes. Variáveis simbólicas definidas em POUs adicionais do tipo programa que forem chamadas dentro de ActivePrg também serão variáveis redundantes.

ATENÇÃO

Variáveis do tipo VAR_TEMP não devem ser utilizadas no programa redundante.

6.4.8.2. Programa NonSkippedPrg

Esta POU é destinada para controles que devem ser executados em ambos os CPs (CPA e CPB), independente do seu estado de redundância. Esta POU também é chamada pela POU principal (MainPrg).

Deve-se lembrar que todas as variáveis simbólicas definidas na POU NonSkippedPrg, assim como instâncias de blocos funcionais definidas, serão variáveis não-redundantes. O usuário pode também criar POUs adicionais (programa, funções ou bloco funcional), e chamá-las ou instanciá-las dentro da POU NonSkippedPrg, para fins de estruturação de seu programa. Também é possível chamar funções e instanciar blocos funcionais definidos em bibliotecas.

ATENÇÃO

Quando chamar POUs adicionais do tipo programa dentro de NonSkippedPrg, desmarque as mesmas na janela Configuração de Redundância do MasterTool. Por padrão as variáveis simbólicas declaradas dentro destas POUs serão redundantes e, dentro da NonSkippedPrg, normalmente deseja-se variáveis não-redundantes. Normalmente o código de NonSkippedPrg é pequeno, dispensando a chamada de POUs adicionais do tipo programa para sua estruturação, mas, caso a sua estruturação seja necessária o mais recomendável é utilizar bloco funcional ou funções.

Exemplos típicos de controles executados na NonSkippedPrg são os seguintes:

- Criar uma estrutura compacta de diagnósticos (%Q) a serem reportados para um sistema SCADA, a partir de uma estrutura completa de diagnósticos, onde diversos diagnósticos não são de interesse para o sistema SCADA. Estes diagnósticos podem ser extraídos de estruturas de dados como RedDgnLoc, RedDgnRem, RedUsrLoc, RedUsrRem, etc.
- Copiar comandos recebidos de um SCADA para os respectivos campos da estrutura de dados RedCmdLoc, e intertravar estes comandos se necessário
- Gerenciar switch-overs controlados pelo usuário, em função de falhas não-vitais tais como a comunicação com um sistema SCADA ou com dispositivos MODBUS
- Habilitar ou desabilitar alguns drivers de E/S específicos, dependendo do estado da redundância (Ativo ou Não-Ativo). Por exemplo, um driver mestre MODBUS RTU em um barramento RS-485 deve ser desabilitado no CP Não-Ativo. Para mais informações, consultar a seção [Gerenciamento de Instâncias MODBUS em Sistemas Redundantes](#)

ATENÇÃO

Não recomenda-se utilizar os blocos funcionais TOF_RET, TON_RET, TOF e TON no programa NonSkippedPrg. Ver seção [Limitações na Programação de um CP Redundante](#).

6.4.9. Objeto Configuração de Redundância

Este objeto, localizado na árvore de dispositivos do projeto, é criado automaticamente pelo Assistente.

Ele é utilizado para determinar quais POU e GVLs são redundantes, e portanto sincronizadas entre os CPs. Por padrão, POU e GVLs criadas pelo usuário são marcadas como redundantes, cabendo a ele reverter a marcação quando necessário.

ATENÇÃO

Os objetos PV, PIDControl e PidRetainGVL não podem ser marcados individualmente. Caso seja necessário que estes sejam alterados, deve ser marcada a opção *Selecionar Todos*.

6.4.10. GVL Module_Diagnostics

Esta GVL especial é criada e preenchida automaticamente pelo Assistente, e não pode ser modificada pelo usuário.

Diagnósticos e comandos do sistema, inclusive das estruturas de dados de redundância (RedDgnLoc, RedDgnRem, RedCmdLoc, RedCmdRem), residem em variáveis de representação direta %Q ou %I especiais.

A GVL *Module_Diagnostics* contém diversas sentenças com a diretiva AT para definir nomes simbólicos para estes diagnósticos e comandos. Desta maneira, quando o usuário precisar referenciar estas variáveis, poderá usar um nome simbólico ao invés de uma referência numérica.

6.4.11. GVLs com Variáveis Simbólicas Redundantes

O usuário pode criar outras GVLs, diferentes das citadas anteriormente, para declarar variáveis simbólicas redundantes. Para isso, a GVL deve estar marcada na configuração do objeto *Configuração de Redundância*, na árvore de dispositivos do projeto, significando que essa GVL é redundante. Por padrão, todas as GVLs criadas pelo usuário serão, inicialmente, redundantes.

ATENÇÃO

Por boa prática, é recomendado evitar a utilização da diretiva AT em GVLs que contenham declarações de variáveis simbólicas redundantes, para evitar o mapeamento de variáveis em áreas não-redundantes.

6.4.12. POU do Tipo Programa com Variáveis Simbólicas Redundantes

O usuário pode declarar variáveis simbólicas redundantes em POU do tipo programa, com exceção da POU NonSkippedPrg onde as variáveis simbólicas declaradas são consideradas não-redundantes.

Para definir uma nova POU como redundante deve-se, após a criação da POU, marcá-la na configuração do objeto *Configuração de Redundância*, na árvore de dispositivos do projeto. Por padrão, todas as POUs criadas pelo usuário serão, inicialmente, redundantes.

ATENÇÃO

Por boa prática, é recomendado evitar a utilização da diretiva AT em POU que sejam redundantes, a fim de evitar o mapeamento de variáveis em áreas não-redundantes.

6.4.13. Utilização de Breakpoints em Sistemas Redundantes

Para sistemas redundantes, o recomendado é utilizar breakpoints apenas no half-cluster Ativo, com o outro half-cluster desligado. Caso contrário, quando a execução da aplicação atingir um breakpoint, o half-cluster Reserva irá assumir o estado Ativo, desligando o CP ativo.

6.4.14. Gerenciamento de Instâncias MODBUS em Sistemas Redundantes

Para o caso de utilização da falha vital das interfaces Ethernet estar desabilitada ou no caso de instâncias MODBUS Servidor, as instâncias MODBUS são independentes da redundância e, portanto, deverão ser gerenciadas na aplicação, ficando a critério do usuário quais instâncias deverão ser habilitadas/desabilitadas quando um CP entrar em um estado Não-Ativo. Quando for habilitada a falha vital para portas Ethernet com MODBUS Cliente não é necessário implementar código adicional para controlar o switch-over.

O exemplo abaixo, inserido no programa NonSkippedPrg, faz a verificação do estado atual do CP e, caso esteja em um estado Não-Ativo, desabilita as instâncias MODBUS RTU Mestre e Escravo e a instância MODBUS Ethernet Servidor:

```

VAR
eRedStateLocal : REDUNDANCY_STATE;
eRedStateLocal_old : REDUNDANCY_STATE;
END_VAR

// Leitura do estado atual do CP local
eRedStateLocal := DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.eRedState;

// O estado do CP local mudou?
IF eRedStateLocal <> eRedStateLocal_old THEN
IF eRedStateLocal = REDUNDANCY_STATE.ACTIVE THEN
// CP local entrou em estado Ativo
Diagnostics.DG_MODBUS_RTU_Slave.tCommand.bRestart := TRUE;
Diagnostics.DG_MODBUS_RTU_Master.tCommand.bRestart := TRUE;
Diagnostics.DG_MODBUS_Server.tCommand.bRestart := TRUE;
ELSE
// CP local entrou em um estado Não Ativo
Diagnostics.DG_MODBUS_RTU_Slave.tCommand.bStop := TRUE;
Diagnostics.DG_MODBUS_RTU_Master.tCommand.bStop := TRUE;
Diagnostics.DG_MODBUS_Server.tCommand.bStop := TRUE;
END_IF
// Salva o último estado do CP local
eRedStateLocal_old:= eRedStateLocal;
END_IF

```

6.4.15. Limitações na Programação de um CP Redundante

Em um CP redundante, irão existir algumas limitações quanto à programação dos half-clusters. Estas limitações são tratadas nas subseções abaixo.

6.4.15.1. Limitações em GVLs e POU's Redundantes

Em uma GVL ou em uma POU do tipo programa que sejam redundantes, as seguintes limitações devem ser respeitadas para um correto funcionamento dos half-clusters:

- Não utilizar variáveis do tipo VAR_TEMP
- Não misturar tipos de variáveis (VAR, VAR RETAIN, VAR PERSISTENT, VAR CONSTANT, etc...), devendo ser utilizado somente um dos tipos em cada GVL ou POU
- Não misturar declaração de variáveis simbólicas com ATs nas GVLs. Devem ser criadas GVLs separadas, declarando em uma as variáveis do tipo AT e em outra as variáveis simbólicas

- Não armazenar o endereço de uma variável em uma variável redundante (fazer de uma variável redundante um ponteiro para um endereço), pois os endereços das variáveis podem ser diferentes no CPA e no CPB
- Não utilizar blocos funcionais de escrita e leitura ao relógio RTC em POU's redundantes. Mais detalhes podem ser encontrados na seção [Relógio RTC](#)

6.4.15.2. Limitações no Programa Não-Redundante (NonSkippedPrg)

Em uma POU do tipo programa que não seja redundante, no caso, a POU NonSkippedPrg, as seguintes limitações devem ser respeitadas para um correto funcionamento dos half-clusters:

- Não podem ser utilizados os blocos funcionais TON e TOF tradicionais, pois os mesmos utilizam o temporizador IEC. Quando o CP Reserva entrar em estado ativo (com o outro half-cluster saindo do estado Ativo), o temporizador IEC será sincronizado, causando uma descontinuidade no valor do temporizador. Deve-se optar pelos blocos funcionais TON_NR e TOF_NR disponibilizados na biblioteca *NextoStandard*. Ver seção [Temporizador Não-Redundante](#)
- Não podem ser utilizadas POU's do tipo programa escritas na linguagem SFC (Sequenciamento Gráfico de Funções), pois estas utilizam o temporizador IEC para temporizar as transições
- Não misturar declaração de variáveis simbólicas com AT's nas GVL's. Devem ser criadas GVL's separadas, declarando em uma as variáveis do tipo AT e em outra as variáveis simbólicas

6.4.16. Obtendo o Estado da Redundância de um Half-Cluster

É possível verificar o estado da redundância de um half-cluster na [Estrutura de Diagnósticos da Redundância](#):

```
VAR
eRedStateLocal : REDUNDANCY_STATE;
END_VAR

eRedStateLocal := DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.eRedState;
```

Deste modo, o usuário poderá fazer o controle de lógicas que dependam do estado redundante do CP.

6.4.17. Leitura de Diagnósticos Não-Redundantes

Um projeto redundante, além de apresentar diagnósticos redundantes ([Estrutura de Diagnósticos da Redundância](#) ou os diagnósticos de uma remota PROFIBUS), apresenta também diagnósticos que não são redundantes (diagnósticos específicos dos módulos NX5000, NX5001, NX3030, etc.). Estes diagnósticos não-redundantes poderão não ser válidos nos primeiros instantes após o CP assumir o estado Ativo, pois os mesmos não são sincronizados com o outro CP (não se sabe o estado em que estava o diagnóstico quando o CP remoto estava Ativo). Portanto, estes diagnósticos devem ser desconsiderados nos primeiros instantes em estado Ativo, até que tenham valores válidos. Tipicamente o tempo durante o qual os diagnósticos devem ser desconsiderados é de 5 s.

O exemplo abaixo mostra como desconsiderar os diagnósticos *bSlaveNotPresent* e *bPbusCommFail* do módulo Mestre PROFIBUS NX5001 durante os primeiros 5 s em que o CP está em estado Ativo.

Lógica em *NonSkippedPrg*:

```
PROGRAM NonSkippedPrg
VAR
TON_DiagEnable : TON_NR;
bDiagEnable : BOOL;
bIsActiveState : BOOL;
bIsActiveState_old : BOOL;
END_VAR

bIsActiveState := (DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.eRedState =
REDUNDANCY_STATE.ACTIVE);
TON_DiagEnable(IN:= (bIsActiveState = bIsActiveState_old), PT:= T#5S, Q=>
bDiagEnable);
bIsActiveState_old := bIsActiveState;
```

```
Lógica em ActivePrg:  
IF NonSkippedPrg.bDiagEnable THEN  
IF DG_NX5001.tGeneral.bSlaveNotPresent OR DG_NX5001.tGeneral.bPbusCommFail THEN  
//Executa ações de diagnóstico ativo  
END_IF  
END_IF
```

6.5. Carga de Programas em um CP Redundante

A seção [Programação de um CP Redundante](#) tratou de aspectos relativos ao desenvolvimento de um projeto para um CP redundante com UCP NX3030.

Nesta seção, discutem-se métodos e etapas para carregar este projeto em um CP redundante, considerando diversas situações, tais como:

- Carga do projeto em uma UCP NX3030 nova retirada da caixa, ou em uma UCP contendo um projeto desconhecido
- Carga online de modificações
- Carga offline de modificações com interrupção do controle do processo, durante uma parada programada do processo
- Carga offline de modificações sem interrupção do controle do processo, valendo-se de características da redundância

6.5.1. Carga Inicial de um Projeto Redundante

Esta seção descreve os passos necessários para fazer a primeira carga de projeto redundante em uma UCP NX3030. Isto é necessário, por exemplo, para uma UCP nova de fábrica, ou para uma UCP que contenha um projeto desconhecido.

ATENÇÃO

Os passos seguintes devem ser executados para os dois half-clusters (CPA e CPB) que compõem um CP redundante. Primeiro deve-se executar todos os passos para um dos CPs, e depois para o outro.

6.5.1.1. Primeiro Passo - Descoberta do Endereço IP para Conexão do MasterTool

O primeiro é descobrir o endereço IP do canal NET 1 desta UCP, para conexão ao MasterTool.

Isto deve ser feito através do visor e a tecla da UCP NX3030, conforme descrito em [Menu Informativo e de Configuração da UCP](#). O menu REDE informa os endereços IPs que podem ser utilizados para conexão via MasterTool.

6.5.1.2. Segundo Passo – Verificar Conflito de Endereços IP

Antes de executar o próximo passo, deve-se ter certeza de que não existe, na rede, outro equipamento com o mesmo endereço IP descoberto no primeiro passo. Isso pode ser descoberto, por exemplo, desconectando o CP da rede e executando um comando “ping” no seu endereço IP. Como o CP está desconectado, espera-se que este “ping” falhe. Se o “ping” funcionar, existe outro equipamento com o mesmo endereço IP.

Caso o endereço IP já esteja em uso por outro equipamento na rede, deve-se executar o terceiro passo, e também alguns passos seguintes, utilizando um cabo de crossover para conectar diretamente o PC com o software MasterTool IEC XE ao CP, evitando assim conflitos de endereços IP. Em um dos passos seguintes, ao carregar o projeto no CP, serão atualizados os endereços IP definitivos do CP (ver seção [Configurações das Portas Ethernet da UCP NX3030 \(NET 1 e NET 2\)](#)).

6.5.1.3. Terceiro Passo – Preparar Conexão do MasterTool (Definir Caminho Ativo)

O terceiro passo consiste em dar um duplo clique sobre o Device (NX3030) na árvore de dispositivos, entrar na aba “*Configurações de Comunicação*”, clicar sobre o Gateway, e pressionar o botão “*Mapear Rede*” para listar todos os CPs detectados pelo MasterTool na rede.

Neste momento, espera-se encontrar uma lista de CPs cuja identificação contém o endereço IP descoberto no primeiro passo e o tipo da UCP (NX3030). Caso o usuário tenha trocado anteriormente o nome da UCP na rede, este nome será o

visualizado neste momento. A seção [Conexão do MasterTool com uma UCP NX3030 de um CP Redundante](#) descreve com maiores detalhes as possíveis identificações que podem ser observadas nesta lista. De qualquer maneira, todas as possíveis identificações possuem um campo que mostra o endereço IP ou parte dele. Por exemplo, os bytes que estão entre colchetes formam o endereço da UCP. O byte que está a direita, entre os bytes que estão entre colchetes, indica o final do endereço IP em hexadecimal. Se os bytes formarem o seguinte endereço [0010], isto significa que o byte com o valor 10 indica que o final do endereço IP dessa UCP é xxx.xxx.xxx.16. A seguir, deve-se clicar sobre este CP na lista e pressionar o botão “*Definir Caminho Ativo*”. Feito isso, o CP selecionado deve aparecer em negrito na lista, o que indica que o MasterTool está preparado para conectar-se a este CP.

6.5.1.4. Quarto Passo – Identificar a UCP NX3030 e Conferir no Visor da UCP

O quarto passo consiste em identificar o half-cluster, como CPA ou CPB. Isto é feito através do menu *Comunicação / Configuração de Redundância*:

A seguir, a combo-box “*Identificação do CP*” permite selecionar uma das três seguintes opções:

- CPA
- CPB
- Não-Redundante

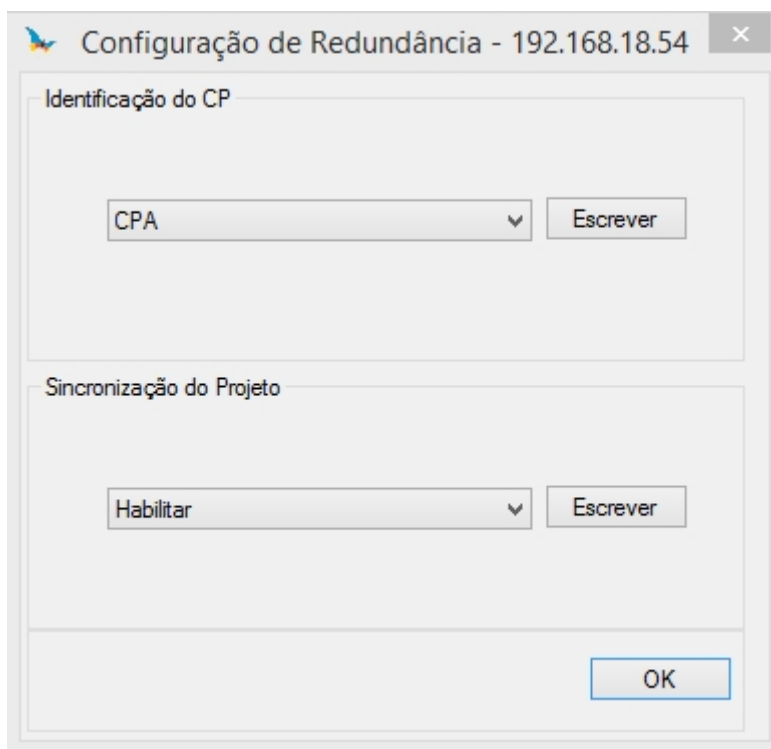


Figura 181: Identificação do CP

No caso de um CP redundante, o usuário deve selecionar CPA ou CPB. Depois de selecionar a opção desejada, o botão “*Escrever*” ao lado desta combo-box deve ser pressionado. O MasterTool retornará um aviso de que a UCP será reinicializada e se o usuário deseja confirmar a ação. Após, irá aparecer uma mensagem indicando sucesso ou falha do comando e, em caso de sucesso, a UCP será reinicializada.

ATENÇÃO

A UCP NX3030 não pode estar em modo Run quando este comando é executado. Antes de executar o comando, o usuário deve colocar a UCP em Modo Stop. Caso a UCP esteja em modo Run, o comando não é executado e o MasterTool avisa que o comando falhou.

Logo depois de executar o comando de identificação com sucesso, observa-se que a identificação selecionada aparece nos [Diagnósticos da Redundância no Visor Gráfico da UCP NX3030](#).

A identificação do CP também estará disponível em um diagnóstico interno (DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.ePLC_ID). Este diagnóstico é atualizado a partir da memória não-volátil a cada ciclo da MainTask, portanto é necessário que o CP volte ao modo Run para que o mesmo seja atualizado. Abaixo, estão relacionados os códigos retornados pelo diagnóstico e suas respectivas descrições:

- Não-Redundante: 0
- CPA: 2
- CPB: 3

A identificação de um CP não faz parte do projeto redundante desenvolvido com o MasterTool. Tal identificação reside apenas em uma área de memória não-volátil da UCP, que pode ser modificada usando o MasterTool.

CUIDADO

A redundância não funcionará adequadamente caso uma das UCPs não seja identificada como CPA e a outra UCP não seja identificada como CPB, podendo ocasionar interrupção do controle do processo. Caso uma UCP NX3030 deva ser substituída (ex: após um dano), a nova UCP deve ser previamente identificada com a mesma identificação daquela UCP que está sendo substituída. Deve-se usar o visor da UCP para verificar se os dois CPs foram corretamente identificados.

6.5.1.5. Quinto Passo – Carga do Projeto Redundante

Este passo descreve a carga do projeto redundante no CP. Este projeto deve ser preparado conforme descrito na seção [Programação de um CP Redundante](#).

Um projeto simples (básico) pode ser preparado seguindo, no mínimo, as seguintes subseções desta seção:

- [Assistente para Criação de um Novo Projeto Redundante](#)
- [Configurações das Portas Ethernet da UCP NX3030 \(NET 1 e NET 2\)](#)

Obviamente, também é possível fazer um projeto completo e só depois carregá-lo no CPA e CPB, por exemplo, caso o hardware destes CPs não esteja disponível durante o desenvolvimento do projeto com o MasterTool.

A primeira carga de um projeto redundante em uma UCP, previamente identificada como CPA ou CPB, ainda deve ser feita utilizando aquele endereço IP descoberto no primeiro passo deste procedimento, e selecionado no terceiro passo deste procedimento. A carga do projeto é feita através do menu *Comunicação / Login*.

ATENÇÃO

Dentro do projeto desenvolvido com o MasterTool IEC XE e carregado no CP neste passo, foram definidos novos endereços IP para a interface NET 1 do CPA e CPB (Endereço de IP do CPA e Endereço de IP do CPB), assim como um endereço IP para a interface NET 1 do CP Ativo (Endereço de IP Ativo) – ver seção [Configurações das Portas Ethernet da UCP NX3030 \(NET 1 e NET 2\)](#).

Portanto, depois desta carga inicial, aquele endereço IP descoberto no primeiro passo deste procedimento normalmente não será mais válido. Esta mudança do endereço IP em NET 1 ou NET2 provocará uma perda da conexão do MasterTool com o CP, que será notificada.

Para maiores detalhes sobre como reconectar o MasterTool, ver seção [Conexão do MasterTool com uma UCP NX3030 de um CP Redundante](#).

6.5.2. Conexão do MasterTool com uma UCP NX3030 de um CP Redundante

Depois de executar o procedimento descrito na seção [Carga Inicial de um Projeto Redundante](#) nos dois CPs (CPA e CPB), a conexão ao MasterTool, através da interface NET 1 da UCP NX3030, poderá ser feita através de um dos seguintes endereços:

- Endereço de IP do CPA: endereço de NET 1 exclusivo para o CPA
- Endereço de IP do CPB: endereço de NET 1 exclusivo para o CPB

Independente do estado do CP, o MasterTool só consegue se conectar ao mesmo utilizando o endereço exclusivo do CP, configurado em Endereço de IP do CPX. Porém, caso o CP se encontre em estado Ativo, todos os outros serviços poderão se conectar ao CP tanto pelo Endereço de IP do CPX quanto pelo Endereço de IP Ativo.

Para conectar-se a determinado CP, em primeiro lugar deve-se dar um duplo clique sobre o *Device* (NX3030) na árvore de dispositivos, entrar na aba “*Configurações de Comunicação*”, clicar sobre o *Gateway*, e pressionar o botão “*Mapear Rede*” para listar todos os CPs detectados pelo MasterTool na rede.

Nesta lista, será possível encontrar as seguintes identificações padrão, caso o nome do CP na rede não tenha sido alterado anteriormente pelo usuário:

- NX3030_<Endereço IP>_PLCA: identificação relacionada ao CPA. Neste caso, o campo <Endereço IP> deve coincidir com o Endereço de IP do CPA configurado no projeto.
- NX3030_<Endereço IP>_PLCB: identificação relacionada ao CPB. Neste caso, o campo <Endereço IP> deve coincidir com o Endereço de IP do CPB configurado no projeto.

A seguir, deve-se selecionar o CP desta lista onde o MasterTool deve se conectar, e pressionar o botão “*Definir Caminho Ativo*”. Posteriormente, ao executar o comando do menu *Comunicação / Login*, o MasterTool se conectará a este CP.

ATENÇÃO

O MasterTool só consegue se conectar a um CP de cada vez. Para conectar-se a diversos CPs, deve-se abrir múltiplas instâncias do MasterTool, cuidando sempre para abrir o projeto correto em cada instância.

6.5.3. Carga de Modificações em um Projeto Redundante

Depois que os dois CPs (CPA e CPB) que formam um CP redundante já receberam uma carga inicial, conforme descrito na seção [Carga Inicial de um Projeto Redundante](#), é possível carregar modificações sucessivas do projeto, à medida que tais modificações forem necessárias.

A conexão do MasterTool aos CPs para executar a carga de modificações deve ser feita conforme descreve-se na seção [Conexão do MasterTool com uma UCP NX3030 de um CP Redundante](#). Nesta seção explica-se como é possível conectar-se a um CP específico (CPA ou CPB), ou ao CP Ativo, ou ao CP Não-Ativo.

Normalmente as modificações devem ser carregadas no CP Ativo, e a seguir serão automaticamente sincronizadas para o CP Não-Ativo, através dos canais de sincronismo NETA/NETB. Portanto, o MasterTool normalmente deve usar o endereço de IP exclusivo do CP Ativo (Endereço de IP do CPX), para se conectar ao canal NET 1 da UCP NX3030 do CP Ativo. Para verificar qual dos CPs está em estado Ativo, poderá ser seguido o mesmo passo descrito em [Carga Inicial de um Projeto Redundante - Quarto Passo – Identificar a UCP NX3030 e Conferir no Visor da UCP](#).

ATENÇÃO

Carregar um projeto no CP Não-Ativo normalmente é inútil, pois a sincronização automática de projetos (do CP Ativo para o CP Não-Ativo) cancelaria o efeito desta carga. Entretanto, há situações especiais em que a sincronização de projetos pode ser desabilitada temporariamente, sendo possível e útil carregar um projeto diferente no CP Não-Ativo. Estas situações especiais são discutidas na seção [Explorando a Redundância para Carga Offline de Modificações sem Interrupção do Controle do Processo](#).

6.5.4. Carga de Modificações Offline e Online

Modificações de projeto podem ser carregadas offline ou online.

Cargas offline requerem a parada do CP onde a modificação deve ser carregada. Por outro lado, cargas online permitem que o CP continue executando sua aplicação enquanto a modificação é carregada.

Alguns tipos de modificações exigem carga offline, ou seja, não podem ser carregadas online no CP onde o MasterTool está conectado. Neste caso, existem duas opções:

- Interromper o controle do processo, executando o procedimento descrito na seção [Carga Offline de Modificações com Interrupção do Controle do Processo](#)
- Valer-se da redundância do CP e das redes PROFIBUS para não interromper o controle do processo, mesmo com a necessidade de fazer cargas offline em cada um dos half-clusters (CPA ou CPB). Um procedimento para atingir este objetivo é descrito na seção [Explorando a Redundância para Carga Offline de Modificações sem Interrupção do Controle do Processo](#)

6.5.4.1. Modificações que Demandam Carga Offline com Interrupção do Controle do Processo

As seguintes modificações em um projeto farão com o que o mesmo se torne impossível de ser carregado num sistema redundante sem a interrupção do controle do processo:

- Modificações nas áreas de memória redundantes (alteração dos *Parâmetros de Redundância* do módulo NX4010)

ATENÇÃO

Não será possível alterar o tamanho das áreas de memória redundantes sem interromper o controle do processo. Portanto estas áreas devem ser cuidadosamente planejadas e configuradas previamente.

6.5.4.2. Modificações que Demandam Carga Offline

As seguintes modificações demandam carga offline no CP:

- Adicionar ou remover dispositivos na árvore de dispositivos, por exemplo:
 - Módulos no bastidor principal (mestres PROFIBUS NX5001, interfaces Ethernet NX5000, etc.)
 - Remotas em redes PROFIBUS
 - Módulos de E/S em remotas em redes PROFIBUS
 - Instâncias MODBUS
- Modificar parâmetros dentro de dispositivos existentes na árvore de dispositivos, por exemplo:
 - Endereços IP e outros parâmetros de interfaces Ethernet
 - Parâmetros de mestres PROFIBUS
 - Parâmetros de remotas PROFIBUS
 - Parâmetros de módulos de E/S dentro de remotas PROFIBUS
- Modificações nas configurações das tarefas.
- Atualização de Projeto devido a Atualização do programador MasterTool IEC XE

6.5.4.3. Modificações que Permitem Carga Online

Em princípio, as modificações não citadas nas seções [Modificações que Demandam Carga Offline com Interrupção do Controle do Processo](#) e [Modificações que Demandam Carga Offline](#), permitem carga online.

Mesmo assim, a seguir serão citadas as principais modificações que permitem carga online no CP onde o MasterTool está conectado. As modificações citadas abaixo valem para variáveis, POU's e GVL's, redundantes ou não.

- Adicionar POU's do tipo programa, desde que estas POU's não precisem ser associadas à alguma tarefa
- Remover POU's do tipo programa, desde que estas POU's não estejam associadas à alguma tarefa.
- Adicionar ou remover POU's do tipo função ou bloco funcional
- Modificar o código de quaisquer tipos de POU (programa, função ou bloco funcional)
- Adicionar ou remover variáveis simbólicas em quaisquer tipos de POU (programa, função ou bloco funcional, sendo elas redundantes ou não)
- Adicionar ou remover instâncias de bloco funcional em POU's do tipo programa ou bloco funcional
- Adicionar ou remover GVL's
- Adicionar ou remover variáveis simbólicas ou instâncias de bloco funcional em GVL's

6.5.5. Carga Online de Modificações

Na seção [Carga de Modificações Offline e Online](#), foram descritas modificações que demandam carga offline e aquelas que permitem carga online.

Uma carga online deve ser feita conectando o MasterTool ao canal NET 1 do CP Ativo, utilizando seu endereço exclusivo de IP. Antes da versão 2.01 do MasterTool IEC XE, era necessário que o usuário selecionasse a opção “*Criar Aplicação de Inicialização*” no menu *Comunicação*, após o envio da aplicação, para que as modificações fossem enviadas para a área de memória não-volátil da UCP e pudessem ser sincronizadas. A partir da versão 2.01 esta operação não é mais necessária. Após o envio da aplicação, a operação de envio para a memória não volátil é realizada de forma automática.

ATENÇÃO

É importante lembrar que modificações online, sem que a opção mencionada anteriormente seja selecionada, serão perdidas no caso de um Reset a Quente ou a UCP ser desligada.

ATENÇÃO

Uma alteração online na declaração de variáveis retentivas da aplicação (inclusão ou remoção de variáveis) seguida de uma queda da alimentação do CP antes de “*Criar Aplicação de Inicialização*”, irá corromper a memória retentiva, pois o valor das variáveis retentivas que foram salvas não corresponde às variáveis da aplicação recuperadas na memória restaurada.

Quando o CP Não-Ativo percebe que possui um projeto diferente do CP Ativo, ele executa as seguintes ações:

- Negocia uma sincronização automática de projeto com o CP Ativo
- Caso esteja no estado Reserva ou Inicializando, chaveia para o estado Não-Configurado, e permanece neste estado até que os projetos estejam novamente sincronizados. Depois disso, volta automaticamente ao estado Reserva
- Caso esteja no estado Não-Configurado ou Inativo, o botão STAND-BY do painel PX2612 deve ser pressionado, ou comando equivalente a este botão deve ser executado. Desta forma, depois da sincronização do projeto, sairá do estado Não-Configurado e poderá ir ao estado Reserva, ou voltar para o estado Inativo se ocorrer alguma falha

6.5.6. Carga Offline de Modificações com Interrupção do Controle do Processo

Nesta seção, define-se o procedimento para executar uma carga offline que interromperá o controle do processo. Tal situação é aceitável em determinados tipos de processos e durante paradas programadas deste processo.

Uma carga offline deste tipo deve ser feita conectando o MasterTool ao canal NET 1 do CP Ativo, utilizando o endereço exclusivo do CP em estado Ativo (Endereço de IP do CPX). Antes de iniciar uma carga offline no CP Ativo, o usuário recebe o seguinte aviso do MasterTool:

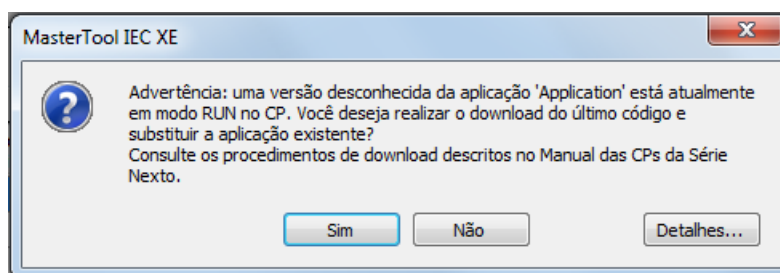


Figura 182: Aviso de Carga Offline

Ao pressionar o botão *Sim* o projeto é enviado. Quando uma carga offline é executada, o controle do processo é interrompido, pois o envio do projeto deve ser feito para o CP Ativo que sairá do modo de execução (Run), e portanto estará no estado Não-Configurado.

Outro ponto importante é que se o outro CP estiver em estado Reserva, deve-se passá-lo para o estado Inativo, por exemplo, apertando o botão INACTIVE do PX2612 para este CP, ou comando equivalente nos diagnósticos da redundância. Desta forma, evita-se que o outro CP desligue este CP e assuma como Ativo.

ATENÇÃO

Quando o CP Ativo sai do modo Run e vai para o estado Não-Configurado, se o outro CP foi esquecido em estado Reserva, o outro CP assumirá como Ativo e desligará o CP que passou de Ativo para Não-Configurado. Neste caso, portanto, a carga offline não poderá ser completada porque o CP conectado ao MasterTool foi desligado.

Quando a carga offline terminar, é possível reiniciar a execução do programa no CP onde a aplicação foi carregada (colocar no modo Run novamente). Depois de alguns segundos, este CP reassume o estado Ativo.

Depois que este CP reassume o estado Ativo, pode-se tirar o outro CP do estado Inativo, por exemplo, apertando o botão STAND-BY do PX2612 para este CP. Isso provocará a transição deste CP para o estado Não-Configurado. Este CP permanecerá no estado Não-Configurado até que o processo automático de sincronização de projetos termine. Depois disso, este CP passa para o estado Inicializando e depois voltará para o estado Reserva.

6.5.7. Planejamento Prévio para Modificações Offline sem Interrupção do Controle do Processo

A subseção seguinte – [Planejamento Prévio para Modificações a Quente em Redes PROFIBUS Redundantes](#) – descreve um procedimento muito importante que permite fazer cargas offline de modificações sem interromper o controle do processo. Embora este procedimento não se aplique à quaisquer modificações que demandem carga offline, ele se aplica àquelas modificações realizadas com maior frequência.

No entanto, para que o procedimento seja aplicável, os projetos devem ser desenvolvidos com algum planejamento prévio, particularmente para modificações que afetam a rede PROFIBUS. As seguintes subseções descrevem tais planejamentos prévios para modificações que afetam a rede PROFIBUS, e também para outras modificações.

6.5.7.1. Planejamento Prévio para Modificações a Quente em Redes PROFIBUS Redundantes

Entre as modificações que afetam uma rede PROFIBUS e demandam carga offline, as seguintes são suportadas pelo procedimento que permite fazer cargas offline de modificações sem interromper o controle do processo, desde que a rede PROFIBUS seja redundante:

- Inserir uma nova rede PROFIBUS
- Inserir uma nova remota da série Ponto
- Inserir um novo módulo de E/S em uma remota da série Ponto
- Modificar parâmetros em remotas da série Ponto ou em módulos de E/S de remotas da Série Ponto

ATENÇÃO

É possível inserir remotas não-redundantes abaixo de uma rede PROFIBUS redundante utilizando o módulo AL-2433 (ProfiSwitch), conforme exemplo mostrado na Figura 158. No entanto, tais remotas não-redundantes sofrerão descontinuidades (desligamento de saídas) quando forem carregadas modificações offline.

A seguir, descreve-se as etapas de um planejamento que deve iniciar no momento da criação de uma nova rede PROFIBUS redundante.

6.5.7.1.1. Etapa 1 – Planejar Expansão Futura das Remotas Inseridas na Versão Inicial da Rede PROFIBUS

Em primeiro lugar, deve-se fazer uma lista dos módulos de E/S que constituirão cada remota PROFIBUS redundante da Série Ponto, na versão inicial da rede PROFIBUS. Na lista, deve constar a posição onde cada módulo de E/S será inserido no bastidor da remota.

A seguir, deve-se planejar como cada uma destas remotas poderá ser expandida no futuro. Para tanto, deve-se fazer uma lista complementar com módulos de E/S que poderão ser inseridos futuramente. Na lista complementar, deve constar a posição onde cada módulo de E/S será inserido futuramente no bastidor de cada remota.

ATENÇÃO

Na construção física destas remotas (painéis elétricos), é altamente recomendável inserir bases compatíveis com os módulos de E/S futuros nas respectivas posições. Desta maneira, quando for necessário inserir o módulo de E/S nesta remota, não será necessário desligar a mesma para inserir a base. Caso este detalhe não seja observado, será necessário desligar esta remota específica, pois não é possível inserir uma base à quente em uma remota. Observa-se que a parada de uma remota específica em alguns casos pode ser tolerável, mas em outros casos pode não ser.

ATENÇÃO

Deve-se colocar os módulos de E/S originais nas primeiras posições do bastidor da remota, e os módulos de E/S futuros nas últimas posições do bastidor da remota.

ATENÇÃO

Considerar as limitações das remotas redundantes da Série Ponto ao elaborar esta lista, conforme o Manual de Utilização Cabeça PROFIBUS PO5063V1 e Cabeça Redundante PROFIBUS PO5063V5, e o Manual de Utilização Cabeça PROFIBUS PO5064 e Cabeça Redundante PROFIBUS PO5065. Existem limites quanto ao número de módulos por remota, número de bytes de E/S por remota, consumo de corrente por fonte, etc. Estes limites podem ser verificados automaticamente utilizando o ProPonto. Para maiores informações, consulte o Manual de Utilização MT6000 MasterTool ProPonto – MU299040.

6.5.7.1.2. *Etapa 2 – Inserir a Versão Inicial da Rede PROFIBUS Redundante no Projeto*

Para inserir a versão inicial de rede PROFIBUS redundante no Projeto, inicialmente deve-se inserir os dois módulos NX5001 redundantes no bastidor, ou utilizar aqueles já inseridos pelo Assistente da redundância.

Em seguida, deve-se inserir cada uma das remotas na árvore de dispositivos abaixo do primeiro NX5001 do par redundante na árvore de dispositivos, bem como os módulos de E/S abaixo de cada remota.

Quanto aos módulos de E/S inseridos, existem duas categorias que devem ser tratadas de maneira diferente:

- Aqueles que fazem parte da versão inicial da rede PROFIBUS, e serão instalados imediatamente
- Aqueles que serão utilizados para expansão futura

No caso daqueles que fazem parte da versão inicial da rede PROFIBUS, deve-se inserir o próprio módulo na árvore de dispositivos, na posição planejada da remota correspondente.

No caso daqueles que serão utilizados para expansão futura, deve-se inserir um módulo virtual correspondente na posição planejada. Um módulo virtual simula um módulo real e precisa alocar o mesmo número de bytes de E/S que o módulo real. A inserção de um módulo virtual no lugar de um módulo real impede que diagnósticos de ausência do módulo real sejam produzidos.

A tabela abaixo exemplifica alguns módulos reais e seus respectivos módulos virtuais correspondentes:

Módulo Real	Módulo Virtual Correspondente
PO1000	PO9999 – 2 bytes Input
PO1001	PO9999 – 2 bytes Input
PO1002	PO9999 – 2 bytes Input
PO1003	PO9999 – 2 bytes Input
PO2020	PO9999 – 2 bytes Output
PO2022	PO9999 – 2 bytes Output

Tabela 204: Módulos Virtuais Correspondentes aos Módulos Reais

6.5.7.1.3. *Etapa 3 – Alocar Áreas de Variáveis %I e %Q para a Rede PROFIBUS considerando Expansão para Remotas Futuras*

Na medida que os NX5001, remotas e módulos de E/S foram sendo inseridos na árvore de dispositivos na etapa 2 anterior, foram alocadas variáveis %I e %Q em três áreas diferentes:

- Área de variáveis %I para entradas
- Área de variáveis %Q para saídas
- Área de variáveis %Q para diagnósticos

O MasterTool executa a alocação de cada uma destas três áreas de variáveis de maneira contígua, sem deixar intervalos nas alocações.

Deve-se planejar o endereço inicial e final de cada uma destas três áreas, considerando as remotas inicialmente instaladas na rede (ver etapas 1 e 2 anteriores), mas também remotas que poderão ser inseridas no futuro nesta mesma rede PROFIBUS.

Ao definir o endereço inicial de cada área, é importante reservar expansão para o dispositivo que aloca endereços imediatamente antes do início desta área. Por outro lado, ao definir o endereço final de cada área, é importante reservar expansão para esta rede PROFIBUS.

A seguir, mostra-se um exemplo de tal planejamento, para a área de variáveis %I de entrada:

- Rede PROFIBUS 1:
 - %IB0 ... %IB499 (endereços alocados para remotas já instaladas)
 - %IB500 ... %IB999 (endereços alocados para remotas futuras)
- Rede PROFIBUS 2:
 - %IB1000 ... %IB1499 (endereços alocados para remotas já instaladas)
 - %IB1500 ... %IB1999 (endereços alocados para remotas futuras)
- Servidor MODBUS TCP:
 - %IB2000 ... %IB2999 (endereços alocados para mapeamentos atuais)
 - %IB3000 ... %IB3999 (endereços alocados para mapeamentos futuros)

Para as outras duas áreas (%Q de saída, %Q de diagnósticos) poderia-se mostrar exemplos similares de planejamentos.

É possível prever o tamanho das áreas inicialmente alocadas e de expansão futura utilizando a tabela abaixo, que indica as quantidades de bytes alocados em cada uma das três áreas, para cada módulo:

Módulo	%I entradas (bytes)	%Q saídas (bytes)	%Q diagnósticos (bytes)
NX5001	4	2	86
PO5063V5	0	0	25
PO5065	0	0	25
PO9100 (um por remota)	2	2	10
PO1000	2	0	10
PO2020	0	2	10
PO9999 – 2 bytes Output	0	2	10
PO9999 – 2 bytes Input	2	0	10

Tabela 205: Alocação de Variáveis %I e %Q para Módulos da Rede PROFIBUS

Nota:

Alocação de Variáveis: Mais informações sobre o tamanho e o tipo de memória alocado para cada módulo podem ser encontradas no Manual de utilização Mestre PROFIBUS DP NX5001.

Depois de executar o planejamento das três áreas (endereços inicial e final de cada área), deve-se introduzir os endereços iniciais no projeto iniciado na etapa 2.

Em primeiro lugar, deve-se modificar o parâmetro “*Endereço Inicial de Diagnósticos do Módulo em %Q*” do primeiro módulo NX5001. Deve-se utilizar o endereço inicial planejado para a área de variáveis %Q para diagnósticos.

Em segundo lugar, deve-se procurar o primeiro módulo de E/S da rede, a partir dos NX5001, que aloque variáveis %I para entradas. Ao encontrá-lo, deve-se alterar o parâmetro “*Endereço*” correspondente.

Em terceiro lugar, deve-se procurar o primeiro módulo de E/S da rede, a partir dos NX5001, que aloque variáveis %Q para saídas. Ao encontrá-lo, deve-se alterar o parâmetro “*Endereço*” correspondente.

ATENÇÃO

Neste momento, é aconselhável verificar a faixa de endereços alocados para as três áreas de variáveis, verificando se os endereços finais de cada área estão dentro da faixa planejada, e se existe uma boa expansão livre para novas remotas a serem inseridas futuramente.

6.5.7.2. Planejamento Prévio para Outras Modificações

Existem outras modificações, que embora não afetem uma rede PROFIBUS, também demandam carga offline. A seguir, são mostrados alguns exemplos de modificações deste tipo suportadas pelo procedimento que permite fazer cargas offline de modificações sem interromper o controle do processo:

- Inserção de módulos NX5000 (Ethernet)
- Inserção de drivers de E/S de comunicação Ethernet ou Serial

- Inserção de novos mapeamentos em drivers de E/S de comunicação Ethernet ou Serial

Por outro lado, as demais modificações exemplificadas anteriormente nesta seção envolvem a alocação de variáveis de representação direta %I e %Q para diagnósticos, entradas e saídas, de maneira similar ao que se discutiu na etapa 3 do planejamento prévio para modificações a quente que afetam a rede PROFIBUS (ver seção [Etapa 3 – Alocar Áreas de Variáveis %I e %Q para a Rede PROFIBUS considerando Expansão para Remotas Futuras](#)).

Desta forma, ao inserir um módulo NX5000, ou ao inserir um driver de E/S Ethernet ou Serial, deve-se planejar a alocação das 3 seguintes áreas para o dispositivo inserido:

- Área de variáveis %I para entradas
- Área de variáveis %Q para saídas
- Área de variáveis %Q para diagnósticos

A seção [Etapa 3 – Alocar Áreas de Variáveis %I e %Q para a Rede PROFIBUS considerando Expansão para Remotas Futuras](#) mostra um exemplo de alocação conjunta destas áreas, envolvendo redes PROFIBUS e um driver de E/S (Servidor MODBUS TCP).

6.5.7.3. Incompatibilidade de Aplicações

Caso as áreas a serem utilizadas futuramente não sejam planejadas corretamente, as áreas de memória redundante podem ter que ser alteradas, gerando, assim, uma incompatibilidade entre as aplicações. Isto irá implicar em apenas um CP manter-se no estado Ativo, com o outro CP permanecendo Inativo, sem a possibilidade de sincronizar dados redundantes ou a aplicação entre os dois CPs.

Esta incompatibilidade é informada pelo diagnóstico da redundância:
DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.bApplicationIncompatible.

Este diagnóstico é ativo quando o download de uma nova aplicação é feito para um dos CPs, geralmente o Não-Ativo, com uma das seguintes alterações:

- Modificação das áreas de dados redundantes, configuradas nos parâmetros do módulo NX4010
- Alteração (criar ou remover) de variáveis simbólicas redundantes, declaradas em POU's ou GVL's redundantes

É importante observar que, para a alteração de variáveis simbólicas redundantes, o problema de incompatibilidade ocorre somente com um novo download para um dos CPs. Caso a modificação das variáveis simbólicas redundantes, e todas as outras alterações feitas no projeto, se enquadrem no grupo de [Modificações que Permitem Carga Online](#), é possível fazer uma [Carga Online de Modificações](#) e não gerar nenhuma incompatibilidade entre as aplicações dos CPs.

6.5.7.4. Atualizações de Projeto Devido a Atualizações do MasterTool IEC XE

A ferramenta de programação MasterTool IEC XE passa por um constante processo de melhoria, aprimorando suas características e adicionando novas. Quando é necessária a atualização da ferramenta em um sistema redundante o projeto utilizado precisa ser atualizado. Esta atualização é realizada através do menu *Projeto/Atualização de Projeto* disponível na ferramenta. Após atualizar o projeto, o procedimento de carga Offline sem Interrupção do Controle do Processo pode ser realizado.

6.5.7.4.1. Atualização de Projetos de Versões Inferiores a 2.00 para versão 2.00 ou Superior

Dentre as trocas entre versões do software MasterTool IEC XE existe um caso especial que deve ser planejado com maior cautela para evitar a parada do processo. A atualização de projeto criado com versão inferior a 2.00 do MasterTool IEC XE para a versão 2.00 ou superior causa uma reconfiguração na área alocada para o objeto Variáveis Persistentes. Esta reconfiguração foi implementada visando uma otimização na alocação desta área. Contudo, se este objeto estiver presente e marcado como redundante em um projeto, esta reorganização não permitirá que os dados sejam sincronizados entre os dois projetos, colocando sempre um dos Half-Clusters em estado Inativo.

Desta forma, se esta situação ocorrer, o software da UCP NX3030 consegue detectar a situação e para a sincronização de dados do objeto *Variáveis Persistentes* até que os projetos nos dois Half-Clusters sejam iguais e por conseguinte estejam utilizando um projeto com a versão atualizada do MasterTool IEC XE. Esta situação não irá parar o controle do processo, porém se não for seguida uma sequência correta de atualização, os dados do objeto Variáveis Persistentes podem ser reinicializados.

Nestes casos a seguinte sequência de Carga Offline deve ser realizada:

- Alterar o projeto do Half-Cluster em estado Ativo desmarcando o objeto *PersistentVars* dentro do objeto *Configuração de Redundância*. Esta carga deve ser realizada como uma carga Online e para que possa ser feita se faz necessária mais uma alteração no projeto, por exemplo, declarando uma variável dentro da POU *NonSkippedPrg*

- Ao final da Carga Online será necessário executar o comando *Criar Aplicação de Inicialização* quando estiver Online no CP que se encontra no estado Ativo. Isto é necessário para que a aplicação seja sincronizada com o Half-Cluster que passou para o estado Não-Configurado após a carga
- Atualizar o projeto da versão inferior à 2.00 para versão 2.00 ou superior através do menu *Projeto/Atualização de Projeto* do MasterTool IEC XE
- Desabilitar o sincronismo de projeto através do menu *Comunicação/Configuração de Redundância*
- Executar a carga de projeto atualizado no Half-Cluster que se encontra no estado Reserva. Aparecerá uma mensagem indicando a reorganização das áreas de memória do objeto *PersistentVars*. O procedimento deve continuar e ao final da carga de projeto o Half-Cluster permanecerá em STOP com estado de redundância como Não Configurado
- Passar a UCP para RUN. O Half-Cluster irá transicionar para estado Inicializando e depois para o estado Reserva. O Half-Cluster irá sincronizar seus dados com o outro que está no estado Ativo
- Os dados do objeto *PersistentVars* devem ser copiados manualmente do Half-Cluster Ativo para o Reserva ou deve se utilizar o recurso de receitas para copiar os dados de um Half-Cluster para o outro
- Colocar o Half-Cluster do estado Ativo no estado Reserva. Com esta ação, o estado do outro Half-Cluster passará para Ativo
- Habilitar o sincronismo de projeto através do menu *Comunicação/Configuração de Redundância*. Após este processo o Half-Cluster que estava no estado Reserva irá para o estado Não-Configurado e receberá o projeto do Half-Cluster em estado Ativo. Ao final deste processo o estado do Half-Cluster mudará para Inicializando e por fim de volta à Reserva
- Alterar o projeto do Half-Cluster em estado Ativo marcando o objeto *PersistentVars* dentro do objeto *Configuração de Redundância*. Esta carga deve ser realizada como uma carga Online e para que possa ser feita se faz necessária mais uma alteração no projeto, por exemplo, removendo uma variável dentro da POU NonSkippedPrg (por exemplo a declarada no início deste procedimento)
- Após este processo o Half-Cluster que estava no estado Reserva irá para o estado Não-Configurado e receberá o projeto do Half-Cluster em estado Ativo. Ao final deste processo o estado do Half-Cluster mudará para Inicializando e por fim de volta à Reserva

6.5.8. Explorando a Redundância para Carga Offline de Modificações sem Interrupção do Controle do Processo

Na seção [Carga de Modificações Offline e Online](#), informou-se que algumas modificações demandam carga offline no CP onde tais modificações devem ser carregadas. Nestes casos, o usuário tem a opção de interromper o controle do processo, conforme procedimento definido na seção [Carga Offline de Modificações com Interrupção do Controle do Processo](#). Para tanto, geralmente é necessário programar previamente uma parada do processo, o que nem sempre é possível no momento em que é necessária uma modificação urgente.

Graças à redundância dos CPs e, em alguns casos, graças à redundância da rede PROFIBUS, é possível realizar cargas offline sem interromper o controle do processo para a maior parte das modificações usualmente necessárias. Para atingir este objetivo, é necessário seguir atentamente um procedimento, cujas etapas são descritas nas subseções seguintes.

6.5.8.1. Etapa 1 – Verificar Atendimento de Requisitos Básicos

Para que sejam possíveis cargas offline sem interromper o controle do processo, os seguintes requisitos básicos devem ser atendidos:

- O projeto original deve ter sido elaborado em conformidade com as recomendações dadas na seção [Planejamento Prévio para Modificações Offline sem Interrupção do Controle do Processo](#)
- O CP deve ser redundante
- Caso a modificação afete uma rede PROFIBUS, é necessário que esta rede PROFIBUS seja redundante. Tais modificações podem ser:
 - Inserção de novas remotas
 - Inserção de módulos de E/S em remotas existentes, em posições previamente reservadas por módulos virtuais correspondentes. Para que a remota não tenha de ser desligada, além disso, deve-se ter uma base compatível com o novo módulo de E/S na posição reservada para o mesmo
 - Modificação de parâmetros em remotas ou módulos de E/S existentes
- Os projetos de ambos os CPs devem estar sincronizados (iguais), e os serviços Sincronização de Dados Redundantes e Sincronização da Lista de Forçamentos Redundantes devem estar funcionando corretamente sem nenhum diagnóstico de falha. Estas condições estão satisfeitas quando existe um CP em estado Ativo e outro em estado Reserva. Caso o CP Não-Ativo não esteja em estado Reserva, pode-se observar seus diagnósticos:
 - DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.bRedDataSync = TRUE, indica sucesso no serviço Sincronização de Dados Redundantes
 - DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.bRedForceSync = TRUE, indica sucesso no serviço Sincronização da Lista de Forçamentos Redundantes
 - DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.dwApplicationCRC = DG_NX4010.tRedundancy.RedDgnRem.dwApplicationCRC, indica que os projetos são iguais nos 2 CPs

6.5.8.2. Etapa 2 – Não Carregar em Conjunto Modificações que podem ser Carregadas Online

Modificações que podem ser carregadas online não devem ser carregadas junto com modificações que devem ser carregadas offline sem interrupção do controle do processo. Sempre que for necessário fazer estes dois tipos de modificações, deve-se executá-las e carregá-las separadamente.

Para que o presente procedimento tenha sucesso, é absolutamente necessário que as modificações realizadas não alterem a estrutura das variáveis redundantes trocadas entre CP Ativo e Não-Ativo, através dos serviços Sincronização de Dados Redundantes e Sincronização da Lista de Forçamentos Redundantes. Estes dois serviços devem continuar funcionando corretamente mesmo enquanto houver diferenças temporárias entre os projetos dos dois CPs. As modificações que devem ser carregadas offline, e suportadas por este procedimento, não afetam a estrutura das variáveis redundantes.

No entanto, algumas modificações que podem ser carregadas online podem alterar a estrutura das variáveis redundantes, por exemplo:

- Inserção de variáveis simbólicas (redundantes ou não), dentro de uma POU ou GVL existente, ou em uma nova POU ou GVL
- Remoção de variáveis simbólicas (redundantes ou não), dentro de uma POU ou GVL existente. A remoção de uma POU ou GVL também pode implicar na remoção de variáveis simbólicas
- Modificação de tamanho/estrutura de variáveis simbólicas (redundantes ou não) em uma POU ou GVL existente

6.5.8.3. Etapa 3 – Backup do Projeto Anterior

Antes de editar as modificações que devem ser carregadas offline sem interromper o processo, por segurança, deve-se realizar um backup da versão anterior do projeto. Poderá ser necessário reinstalar a versão anterior caso algum erro seja cometido durante a execução deste procedimento.

ATENÇÃO

A recomendação de ter um backup de todas as versões que foram carregadas nos CPs não deve ser seguida somente para este procedimento específico. Deve ser uma prática usual.

6.5.8.4. Etapa 4 – Cuidados ao Editar as Modificações Carregadas Offline

As modificações carregadas offline através deste procedimento geralmente são as seguintes:

- Inserção de novos dispositivos na árvore de dispositivos
- Alteração de propriedades ou parâmetros em dispositivos existentes na árvore de dispositivos

Tais dispositivos normalmente são os seguintes:

- Módulos tais como mestres PROFIBUS (NX5001) ou módulos Ethernet (NX5000)
- Remotas PROFIBUS da Série Ponto
- Módulos de E/S dentro de remotas PROFIBUS da Série Ponto
- Drivers de E/S de comunicação MODBUS
- Mapeamentos de drivers de comunicação MODBUS

Os seguintes cuidados devem ser tomados ao editar estas modificações no projeto:

- Se um mesmo dispositivo existia na versão anterior do projeto e continua existindo na versão modificada, as variáveis %I e %Q alocadas para o mesmo devem continuar sendo as mesmas (comandos, diagnósticos, entradas e saídas). Deve-se ter cuidado para que as modificações inseridas no projeto não alterem tais alocações
- Se um dispositivo foi inserido na versão modificada do projeto, as variáveis %I e %Q alocadas para o mesmo devem estar desalocadas na versão anterior do projeto (comandos, diagnósticos, entradas e saídas)

6.5.8.5. Etapa 5 – Desabilitar Sincronismo de Projetos no CP Não-Ativo

Nos procedimentos descritos nas seções [Carga Online de Modificações](#) e [Carga Offline de Modificações com Interrupção do Controle do Processo](#), o projeto é sincronizado automaticamente, do CP Ativo para o CP Não-Ativo.

No entanto, durante o procedimento de carga offline sem interrupção do controle do processo, o sincronismo de projetos deve ser temporariamente desabilitado. Este processo é explicado na seção [Desabilitação da Sincronização de Projetos](#), e a desabilitação deve ser executada no CP Não-Ativo.

6.5.8.6. Etapa 6 – Executar as Modificações Físicas

Neste momento, pode-se executar as modificações físicas, tais como:

- Instalar um novo módulo NX5000. Isto pode ser feito inserindo o módulo a quente no bastidor de cada half-cluster, e depois conectando-o à rede Ethernet
- Instalar uma nova rede PROFIBUS redundante. Os módulos NX5001 podem ser instalados a quente no bastidor de cada half-cluster. Em seguida, a rede PROFIBUS redundante pode ser conectada aos mesmos
- Instalar uma nova remota redundante da Série Ponto. Neste caso, deve-se instalar uma cabeça remota de cada vez, por exemplo, primeiro na rede B, depois na rede A:
 - Para instalar a cabeça na rede B, pode ser necessário abrir o cabo ou a terminação, e desta maneira perturbar a comunicação com as demais cabeças já instaladas na rede B. Antes de fazer isso, deve-se colocar todas as cabeças ativas operando na rede A, e as cabeças reservas operando na rede B
 - Para instalar a cabeça na rede A, pode ser necessário abrir o cabo ou a terminação, e desta maneira perturbar a comunicação com as demais cabeças já instaladas na rede A. Antes de fazer isso, deve-se colocar todas as cabeças ativas operando na rede B, e as cabeças reservas operando na rede A
- Instalar um módulo de E/S em uma base previamente reservada para o mesmo, em uma remota existente

6.5.8.7. Etapa 7 – Carregar as Modificações Offline no CP Não-Ativo

Em primeiro lugar, o MasterTool deve estar conectado ao CP Não-Ativo (ver seção [Conexão do MasterTool com uma UCP NX3030 de um CP Redundante](#)).

A seguir, deve-se carregar as modificações offline. Ao carregá-las, a aplicação do CP Não-Ativo será automaticamente interrompida (sairá do modo Run).

6.5.8.8. Etapa 8 – Voltar o CP Não-Ativo ao Modo Run para que volte ao Estado Reserva

Terminada a carga offline, pode-se voltar o CP Não-Ativo para o modo Run.

Alguns segundos depois, o CP Não-Ativo deve assumir o estado Reserva.

Caso o CP não assuma o estado Reserva, os seguintes problemas podem ter causado este efeito:

- As modificações realizadas acabaram introduzindo modificações na estrutura das variáveis redundantes, o que impede a execução correta do serviço Sincronização de Dados Redundantes. Isto pode ser verificado através do diagnóstico DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.bRedDataSync (0 = falha) no CP Não-Ativo. Neste caso, deve-se desfazer as modificações, recuperando o backup do projeto anterior, e iniciando novamente este procedimento
- Outros problemas podem, eventualmente, impedir a transição para o estado Reserva, embora isto não seja esperado. Neste caso, deve-se observar os diagnósticos e log da redundância

Caso o CP tenha assumido o estado Reserva, é aconselhável verificar se os projetos estão diferentes entre CP Ativo e CP Não-Ativo. Isto pode ser feito comparando os diagnósticos DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.dwApplicationCRC e DG_NX4010.tRedundancy.RedDgnRem.dwApplicationCRC no CP Não-Ativo (os CRCs devem estar diferentes).

Caso os projetos estejam iguais nos 2 CPs, é possível que a desabilitação de sincronismo de projetos (etapa 5) não tenha sido realizada corretamente. Isto pode ser verificado através do diagnóstico DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.bProjectSyncDisable, que deveria valer 1 no CP Não-Ativo. Se isto não ocorre, deve-se voltar à etapa 5.

6.5.8.9. Etapa 9 – Executar Switch-over entre CPs Ativo e Reserva

Deve-se executar switch-over entre os CPs, por exemplo, apertando o botão STAND-BY do CP Ativo, ou executando o comando equivalente. O CP Reserva, que possui o projeto novo com as modificações, assume como Ativo. O CP Ativo, que possui o projeto antigo, assume como Reserva.

6.5.8.10. Etapa 10 – Habilitar Sincronismo de Projetos no CP Ativo

Na etapa 5, o sincronismo de projetos foi desabilitado no CP que estava em estado Não-Ativo. Observa-se que agora este CP está em estado Ativo.

Nesta etapa, deve-se habilitar novamente o sincronismo de projetos neste CP. A tela e metodologia utilizada para isto é a mesma descrita na seção [Desabilitação da Sincronização de Projetos](#), bastando desta vez selecionar a opção “Habilitar” da combo-box. Desta vez, o MasterTool deve estar conectado ao CP Ativo (ver seção [Conexão do MasterTool com uma UCP NX3030 de um CP Redundante](#)).

Depois de habilitar o sincronismo de projetos no CP Ativo, o usuário deve conferir se este comando teve sucesso, verificando se `DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.bProjectSyncDisable = 0`, no CP Ativo.

Assim que o sincronismo de projetos for novamente habilitado, a seguinte sequência de ações é esperada:

- O CP Não-Ativo (em estado Reserva), que já sabia da diferença entre os projetos, sai do estado Reserva para o estado Não-Configurado
- O projeto modificado (novo) é copiado do CP Ativo para o outro CP, temporariamente em estado Não-Configurado
- Assim que os projetos estiverem novamente sincronizados, o CP Não-Configurado passa para o estado Inicializando e, depois, espera-se que volte ao estado Reserva

6.5.8.11. Etapa 11 – Reorganização Opcional de CP e Redes PROFIBUS em Estado Ativo

No final do procedimento, por questões de padronização ou organização, o usuário poderá fazer um switch-over para que o CPA assuma como Ativo, e para que todas as cabeças remotas PROFIBUS em estado ativo estejam na rede A.

6.6. Manutenção da Redundância

6.6.1. Troca a Quente de Módulos em um CP Redundante

Em caso de falha em algum módulo de um dos CPs (CPA ou CPB), pode ser necessária uma troca do módulo, sem interromper o controle do processo. Para isso, os seguintes passos devem ser seguidos:

- Verificar se o half-cluster que não será modificado está em estado Ativo ou em estado Reserva, podendo assumir o controle do processo
- Colocar o half-cluster em que será efetuada a troca do módulo em estado Inativo, através do [Painel de Controle de Redundância PX2612](#) ou dos [Comandos da Redundância](#)
- Efetuar as trocas necessárias no half-cluster Inativo, conforme indicado no capítulo [Configuração da UCP - Parâmetros Gerais - Como realizar a Troca a Quente](#)
- Colocar o half-cluster novamente em estado Reserva ou Ativo, conforme o necessário

6.6.2. Mensagens de Advertência do MasterTool

Quando o MasterTool está com um projeto redundante aberto, ou quando está conectado a uma UCP NX3030 identificada como CPA ou CPB, algumas mensagens de advertência especiais poderão ocorrer, conforme descreve-se nas próximas subseções.

6.6.2.1. Bloqueio de Carga de um Projeto Redundante ou Não-Redundante

O MasterTool não permitirá a carga de um projeto redundante, a menos que a UCP seja NX3030, e que esta UCP esteja identificada como CPA ou CPB (ver seção [Identificação de uma UCP NX3030](#)).

O MasterTool também não permitirá a carga de um projeto não-redundante em uma UCP NX3030 identificada como CPA ou CPB.

Caso ocorra uma tentativa de executar alguma destas ações ilegais, o MasterTool advertirá com uma mensagem apropriada.

6.6.2.2. Alertas antes de Comandos que Podem Parar o CP Ativo

Alguns comandos, como os seguintes, podem parar um CP:

- Carga offline após Comunicação / Login
- Depurar / Parar
- Depurar / Novo Breakpoint
- Comunicação / Reset (a Quente, a Frio, Origem)

Caso o MasterTool esteja logado ao CP ativo e ocorra uma tentativa de execução de um destes comandos, antes de enviá-lo, o MasterTool envia a seguinte mensagem e aguarda autorização para enviar o comando:

"Caso o outro CP estiver no estado Reserva, ele irá assumir como Ativo e desligará este CP. Caso contrário, isto não irá acontecer, mas o processo automatizado não será mais controlado."

6.6.2.3. Alerta antes de Logar-se ao CP Não-Ativo

Em circunstâncias normais, não é usual o MasterTool logar-se ao CP Não-Ativo, portanto, quando houver uma tentativa de execução de um comando deste tipo, o MasterTool emite o seguinte aviso:

"Você está executando um comando de Login em um CP Não-Ativo e isso não é usual. Você tem certeza que deseja executar esse comando?"

Por outro lado, há circunstâncias (não muito usuais) em que é necessário logar-se ao CP Não-Ativo, e nestes casos o usuário deve autorizar o login. Tais circunstâncias podem ocorrer, por exemplo:

- Para configurações iniciais, conforme descrito na seção [Carga Inicial de um Projeto Redundante](#)
- Para carregar offline um projeto diferente no CP Não-Ativo, conforme descrito na seção [Explorando a Redundância para Carga Offline de Modificações sem Interrupção do Controle do Processo](#)
- Para monitorar ou forçar variáveis não-redundantes no CP Não-Ativo

6.6.3. Diagnósticos da Redundância no Visor Gráfico da UCP NX3030

Diversos diagnósticos relativos à redundância podem ser observados no visor da UCP NX3030.

6.6.3.1. Estado de Redundância do CP

O estado de redundância do CP, descrito em [Estados de um CP Redundante](#), é visto nos três caracteres iniciais da segunda linha da tela principal, conforme mostra a seção [Visor Gráfico](#). A tela do visor é apresentada na inicialização, e volta a ser apresentada alguns segundos depois de terminada a navegação (sem apertar a tecla da UCP NX3030).

6.6.3.2. Telas abaixo do Menu REDUNDANCIA

Existe um menu denominado REDUNDANCIA, abaixo do qual existem diversas telas. A descrição e o acesso às telas de redundância estão disponíveis na seção [Configuração – Menu Informativo e de Configuração da UCP](#).

6.6.4. Estrutura de Diagnósticos da Redundância

A área de diagnósticos do módulo NX4010 é mapeada sobre variáveis de representação direta %Q, e definida simbolicamente através da diretiva AT, na GVL Module_Diagnostics.

Essa seção é dividida em duas partes:

- DG_NX4010.tGeneral: Diagnósticos gerais de operação do módulo NX4010. Os mesmos são descritos nas Características Técnicas do Módulo de Link de Redundância – CT114900
- DG_NX4010.tRedundancy: Diagnósticos específicos da redundância, os quais serão descritos no decorrer do capítulo. Esse item é dividido em outras 6 estruturas de dados:
 - RedDgnLoc: Contém diagnósticos de redundância do CP local (conectado), como por exemplo, o estado da redundância do CP. Essa seção é descrita em [Diagnósticos da Redundância](#)
 - RedDgnRem: É uma cópia de RedDgnLoc do outro CP, recebida via canais de sincronismo NETA / NETB. Desta forma, o CP local tem acesso aos diagnósticos do CP remoto. Essa seção é descrita em [Diagnósticos da Redundância](#)
 - RedCmdLoc: Contém comandos de redundância gerados neste CP (local), por exemplo, através de escritas a partir de um sistema SCADA, ou gerados em POU's deste CP (ex: ActivePrg ou NonSkippedPrg). Essa seção é descrita em [Comandos da Redundância](#)
 - RedCmdRem: Trata-se de uma cópia de RedCmdLoc do outro CP (remoto), recebida via canais de sincronismo NETA / NETB. Essa seção é descrita em [Comandos da Redundância](#)
 - RedUsrLoc: Utilizado para que o usuário troque informações entre CPA e CPB. Essa seção é descrita em [Informações do Usuário Trocadas entre CPA e CPB](#)
 - RedUsrRem: Utilizado para que o usuário troque informações entre CPA e CPB. Essa seção é descrita em [Informações do Usuário Trocadas entre CPA e CPB](#)

É importante ressaltar que as estruturas de diagnósticos da redundância do CP remoto são atualizadas somente quando ocorre, com sucesso, uma sincronização de dados. Portanto, enquanto uma nova sincronização não ocorrer, os diagnósticos irão permanecer com o valor lido na última troca de dados.

Além disso, as estruturas do CP remoto são somente para leitura, isto é, valores escritos nestas estruturas serão sobrescritos, sem serem considerados, na próxima sincronização. Sendo assim, não é possível utilizar a estrutura RedCmdRem para executar um comando no CP remoto. A estrutura utilizada para executar comandos deve ser sempre RedCmdLoc.

6.6.4.1. Diagnósticos da Redundância

Os diagnósticos da Redundância podem ter várias utilidades. Entre elas:

- Podem ser consultados para verificar a existência de algum problema que precisa ser sanado
- Toda vez que ocorrem variações nos mesmos, tais variações são inseridas como eventos nos logs de usuário. Consultando a sequência histórica de tais eventos pode-se descobrir, por exemplo, a causa de um switch-over
- Podem ser referenciados na aplicação do usuário (ActivePrg ou NonSkippedPrg). Por exemplo, pode-se testar o estado do CP e, caso não seja Ativo, desabilitar um driver de E/S mestre serial MODBUS RTU, em NonSkippedPrg

ATENÇÃO

O diagnóstico DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.bExchangeSync (definido logo a seguir) deve ser testado para verificar se a estrutura de dados RedDgnRem foi lida com sucesso do CP remoto no último ciclo da MainTask. Caso o valor deste diagnóstico seja FALSE, isso significa que a estrutura de dados RedDgnRem não foi lida com sucesso do CP remoto e, portanto, os valores de RedDgnRem podem ser inválidos ou obsoletos.

Como RedDgnRem é uma cópia de RedDgnLoc do outro CP, as duas estruturas têm o mesmo formato. Estas ainda são divididas em quatro sub-estruturas:

- sGeneral_Diag: Diagnósticos gerais da redundância
- sNETA_Diag: Diagnósticos do canal de sincronismo NETA
- sNETB_Diag: Diagnósticos do canal de sincronismo NETB
- sNET_Stat: Estatísticas comuns para os canais de sincronismo NETA e NETB, para contagem de sucessos e falhas dos serviços de sincronização
- sGeneral_DiagExt: Diagnósticos gerais da redundância, parte estendida (continuação de sGeneral_Diag)

A sub-estrutura “sGeneral_Diag” possui os seguintes campos para diagnósticos gerais da redundância:

Variável Direta		Variável AT	Descrição
Variável	Bit	DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.*	
	0	bConfigDone	TRUE – O processo de configuração, executado no estado Não-Configurado, terminou.
			FALSE – O processo de configuração, executado no estado Não-Configurado, ainda não terminou ou não foi executado.
	1	bConfigError	TRUE – O processo de configuração, executado no estado Não-Configurado, terminou com erros. Trata-se de um erro de sistema, normalmente não esperado. Entre em contato com o suporte da Altus para reportá-lo. Informe também o valor do diagnóstico ConfigErrorCode para o suporte da Altus.
			FALSE – O processo de configuração ocorreu com sucesso ou não foi realizado.
	2	bTooManyRedAreas	TRUE – O número de áreas redundantes excedeu o número máximo permitido. Trata-se de um erro de sistema, normalmente não esperado. Entre em contato com o suporte da Altus para reportá-lo.
			FALSE – O número de áreas redundantes está dentro do esperado.

Variável Direta		Variável AT DG_NX4010.tRedundancy. RedDgnLoc.sGeneral_Diag.*	Descrição	
Variável	Bit			
%QB(n+4)	3	bTemporaryBufferTooSmall	TRUE – Estrutura de dados intermediária com tamanho insuficiente. Trata-se de um erro de sistema, normalmente não esperado. Entre em contato com o suporte da Altus para reportá-lo.	
			FALSE – O tamanho da estrutura de dados está dentro do esperado.	
	4	bExchangeSync	TRUE – O serviço de sincronização Troca de Diagnósticos e Comandos foi executado com sucesso neste ciclo da MainTask.	
			FALSE – A estrutura RedDgnRem tem valores obsoletos ou inválidos, pois não foi lida do outro CP (remota) neste ciclo da MainTask.	
	5	bRedDataSync	TRUE – O serviço Sincronização de Dados Redundantes foi executado com sucesso neste ciclo da MainTask.	
			FALSE – O serviço Sincronização de Dados Redundantes não foi executado com sucesso neste ciclo da MainTask.	
	6	bRedForceSync	TRUE – O serviço Sincronização da Lista de Forçamentos Redundantes foi executado com sucesso neste ciclo da MainTask.	
			FALSE – O serviço Sincronização da Lista de Forçamentos Redundantes não foi executado com sucesso neste ciclo da MainTask.	
	7	bApplicationIncompatible	TRUE – A aplicação não é compatível entre os dois CPs. Foi efetuado o download de uma nova aplicação para um dos CPs com uma das seguintes alterações: Modificação da área de dados redundantes; Modificação de variáveis simbólicas redundantes. Enquanto este diagnóstico estiver ligado, um dos CPs ficará em estado Inativo até que a mesma aplicação esteja presente nos dois CPs. Isto implica em recarregar a aplicação antiga nos dois CPs ou atualizar os dois CPs com a nova aplicação. Para mais informações sobre como proceder, consultar capítulo Carga de Programas em um CP Redundante .	
			FALSE – A aplicação é compatível entre os dois CPs.	
		0	Reservado	Bit reservado.

Variável Direta		Variável AT DG_NX4010.tRedundancy. RedDgnLoc.sGeneral_Diag.*	Descrição
Variável	Bit		
%QB(n+5)	1	bProjectSyncDisable	TRUE – Não será efetuado o sincronismo do projeto de aplicação e do <i>Project archive</i> entre os dois CPs. Trata-se de uma cópia da variável não volátil utilizada para desabilitar a sincronização de projetos, conforme descrito na seção Desabilitação da Sincronização de Projetos . A sincronização de projetos pode ser desabilitada tanto no CP local como no remoto, ou seja, basta executar o comando de desabilitação em um dos CPs para que a sincronização de projetos seja desabilitada. Os comandos de desabilitação do sincronismo de projetos são descritos na seção Desabilitação da Sincronização de Projetos .
			FALSE – Será efetuado o sincronismo do projeto de aplicação e do <i>Project archive</i> entre os dois CPs
	2	bIncompatibleFirmware	TRUE – A versão de firmware é incompatível entre este CP e o CP remoto.
			FALSE – A versão de firmware é compatível entre este CP e o CP remoto.
	3	bApplicationProjectDiff	TRUE – O projeto de aplicação deste CP é diferente do presente no CP remoto.
			FALSE – O projeto de aplicação deste CP é igual ao do CP remoto.
	4	bProjectArchiveDiff	TRUE – O <i>Project archive</i> deste CP é diferente do presente no CP remoto.
			FALSE – O <i>Project archive</i> deste CP é igual ao do CP remoto.
	5	bOnlineChangeApply	TRUE – Foi realizada alguma alteração online na aplicação e esta ainda não foi sincronizada com o CP reserva.
			FALSE – Não foram realizadas alterações online na aplicação ou estas já foram sincronizadas com o CP reserva.
	6	bFailedRED	TRUE – Falha no módulo NX4010. A UCP NX3030 não consegue comunicar-se com este módulo através do barramento, ou existe uma falha no microprocessador do NX4010.
			FALSE – O módulo NX4010 está em correto funcionamento.

Variável Direta		Variável AT DG_NX4010.tRedundancy. RedDgnLoc.sGeneral_Diag.*	Descrição
Variável	Bit		
%QB(n+6)	7	bFailedPBUS1A	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) na rede PROFIBUS 1 A. O modo ativo (comunicando com escravos) é assumido pelo CP Ativo. O modo passivo (comunicando com o mestre em modo ativo) é assumido pelo CP Não-Ativo. Esta falha também pode ser indicada caso o módulo NX5001 tenha falha em seu microprocessador, ou caso não consiga se comunicar com a UCP NX3030 via barramento.
			FALSE – Não existem falhas na rede PROFIBUS 1 A.
	0	bFailedPBUS1B	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) na rede PROFIBUS 1 B. O modo ativo (comunicando com escravos) é assumido pelo CP Ativo. O modo passivo (comunicando com o mestre em modo ativo) é assumido pelo CP Não-Ativo. Esta falha também pode ser indicada caso o módulo NX5001 tenha falha em seu microprocessador, ou caso não consiga se comunicar com a UCP NX3030 via barramento.
			FALSE – Não existem falhas na rede PROFIBUS 1 B.
	1	bFailureProfibus_1	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) na rede PROFIBUS 1. Caso a rede PROFIBUS 1 seja redundante, FailureProfibus_1 resulta de um “E lógico” entre FailedPBUS1A e FailedPBUS1B. Caso a rede PROFIBUS 1 não seja redundante, FailureProfibus_1 é uma cópia de FailedPBUS1A.
			FALSE – Não existem falhas na rede PROFIBUS 1.
2	bFailedPBUS2A	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) na rede PROFIBUS 2 A. O modo ativo (comunicando com escravos) é assumido pelo CP Ativo. O modo passivo (comunicando com o mestre em modo ativo) é assumido pelo CP Não-Ativo. Esta falha também pode ser indicada caso o módulo NX5001 tenha falha em seu microprocessador, ou caso não consiga se comunicar com a UCP NX3030 via barramento.	
		FALSE – Não existem falhas na rede PROFIBUS 2 A.	

Variável Direta		Variável AT DG_NX4010.tRedundancy. RedDgnLoc.sGeneral_Diag.*	Descrição
Variável	Bit		
	3	bFailedPBUS2B	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) na rede PROFIBUS 2 B. O modo ativo (comunicando com escravos) é assumido pelo CP Ativo. O modo passivo (comunicando com o mestre em modo ativo) é assumido pelo CP Não-Ativo. Esta falha também pode ser indicada caso o módulo NX5001 tenha falha em seu microprocessador, ou caso não consiga se comunicar com a UCP NX3030 via barramento.
			FALSE – Não existem falhas na rede PROFIBUS 2 B.
	4	bFailureProfibus_2	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) na rede PROFIBUS 2. Caso a rede PROFIBUS 2 seja redundante, FailureProfibus_2 resulta de um “E lógico” entre FailedPBUS2A e FailedPBUS2B. Caso a rede PROFIBUS 2 não seja redundante, FailureProfibus_2 é uma cópia de FailedPBUS2A.
			FALSE – Não existem falhas na rede PROFIBUS 2.
	5	bProfibusVitalFailureAny	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) com pelo menos uma das redes PROFIBUS configuradas em modo de falha vital.
			FALSE – Não existem falhas nas redes PROFIBUS configuradas em modo de falha vital.
	6	bProfibusVitalFailureAll	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) com todas as redes PROFIBUS configuradas em modo de falha vital.
			FALSE – Não existem falhas nas redes PROFIBUS configuradas em modo de falha vital.
	7	bTurnOffOtherPLC_Normal	TRUE – Este CP está fechando o relé do PX2612 para manter o outro CP desligado em condições normais, e não devido ao modo teste do painel PX2612.
			FALSE – O relé do PX2612 está ligado (bTurnOffOtherPLC_TestMode) ou desligado.
	0	bTurnOffOtherPLC_TestMode	TRUE – Este CP está fechando o relé do PX2612 para manter o outro CP desligado devido ao modo teste do painel PX2612.
			FALSE – O relé do PX2612 está ligado (bTurnOffOtherPLC_Normal) ou desligado.

Variável Direta		Variável AT DG_NX4010.tRedundancy. RedDgnLoc.sGeneral_Diag.*	Descrição
Variável	Bit		
%QB(n+7)	1	bActiveLED	TRUE – O LED ACTIVE do PX2612 está ligado.
			FALSE – O LED ACTIVE do PX2612 está piscando (bBlinkActiveLED) ou desligado.
	2	bBlinkActiveLED	TRUE – O LED ACTIVE do PX2612 está piscando.
			FALSE – O LED ACTIVE do PX2612 está ligado (bActiveLED) ou desligado.
	3	bStandbyLED	TRUE – O LED STAND-BY do PX2612 está ligado.
			FALSE – O LED STAND-BY do PX2612 está piscando (bBlinkStandbyLED) ou desligado.
	4	bBlinkStandbyLED	TRUE – O LED STAND-BY do PX2612 está piscando.
			FALSE – O LED STAND-BY do PX2612 está ligado (bStandbyLED) ou desligado.
	5	bInactiveLED	TRUE – O LED INACTIVE do PX2612 está ligado.
			FALSE – O LED INACTIVE do PX2612 está desligado ou piscando (bBlinkInactiveLED).
	6	bBlinkInactiveLED	TRUE – O LED INACTIVE do PX2612 está piscando.
			FALSE – O LED INACTIVE do PX2612 está ligado (bInactiveLED) ou desligado.
	7	bRedPanelTestMode	TRUE – O painel PX2612 está em modo teste.
			FALSE – O painel PX2612 está em modo normal.
%QB(n+8)	-	ePLC_ID	<p>Este diagnóstico informa a identificação deste CP:</p> <ul style="list-style-type: none"> - 0 = não-redundante - 2 = CPA - 3 = CPB <p>Trata-se de uma cópia da variável não-volátil utilizada para identificar o CP, conforme descrito na seção Identificação de uma UCP NX3030. Na seção Carga Inicial de um Projeto Redundante descreve-se o comando do MasterTool utilizado para escrever sobre esta variável não-volátil.</p>

Variável Direta		Variável AT DG_NX4010.tRedundancy. RedDgnLoc.sGeneral_Diag.*	Descrição
Variável	Bit		
%QB(n+9)	-	eRedState	Informa o estado de redundância deste CP: - Não-Configurado = 0 - Inicializando = 2 - Reserva = 3 - Ativo = 4 - Inativo = 5
%QB(n+10)	-	ePreviousRedState	Valor que o diagnóstico RedState possuía antes da última transição de estados.
%QW(n+11)	-	wRedStateDuration	Mede há quanto tempo (milissegundos) o estado de redundância atual foi assumido. Este tempo para de incrementar quanto atinge 65535 ms.
%QW(n+13)	-	wConfigErrorCode	Código de erro descoberto durante o processo de configuração no estado Não-Configurado. Ver diagnóstico ConfigError descrito anteriormente.
%QD(n+15)	-	dwApplicationCRC	CRC de 32 bits da aplicação, utilizado para detectar diferenças entre as aplicações dos 2 CPs.
%QD(n+19)	-	dwArchiveCRC	CRC de 32 bits do <i>project archive</i> , utilizado para detectar diferenças entre os <i>project archives</i> dos 2 CPs.
%QD(n+23)	-	dwFirmwareVersion	Versão de firmware deste CP, utilizada para verificar compatibilidade entre firmware dos 2 CPs.
%QD(n+27)	-	dwIECTimer	A sincronização do Temporizador IEC é necessária para operação sem colisões de alguns blocos de função como TON e TOF. Através deste diagnóstico o Temporizador IEC do CP Ativo é recebido e atualizado no CP Não-Ativo, desde que o serviço Troca de Diagnósticos e Comandos tenha sido executado com sucesso. Sua contagem inicia em 0 e incrementa até 4294967295. Após estouro de contagem, reinicia com valor 0.
%QW(n+31)	-	wCycleCounter	Contador de 16 bits utilizado como informação auxiliar de sequência nos Logs de Eventos da Redundância. No CP Ativo, é incrementado a cada ciclo da MainTask. No CP Não-Ativo, recebe uma cópia do valor existente no CP Ativo, desde que o serviço Troca de Diagnósticos e Comandos tenha sido executado com sucesso. Sua contagem inicia em 0 e incrementa até 65535. Após estouro de contagem, reinicia com valor 0.

Tabela 206: Diagnósticos Gerais da Redundância

Notas:

Visualização das Estruturas de Diagnóstico: As Estruturas de Diagnóstico adicionadas ao projeto podem ser visualizadas acessando o item “Library Manager” na árvore de dispositivos da janela do MasterTool IEC XE. Com isso, é possível visualizar todos os tipos de dados definidos na estrutura.

Variável de representação direta: O “n” representa o valor configurado no módulo NX4010, através do software MasterTool IEC XE, como Endereço Inicial de Diagnósticos do Módulo em %Q. Esta definição vale para todas as estruturas de diagnóstico.

Diretiva AT: A diretiva AT é uma palavra reservada no software programador, que se relaciona a um endereço de uma variável. No caso do módulo NX4010 a variável DG_NX4010 está relacionada ao endereço inicial de diagnósticos do módulo.

A sub-estrutura “sNETA_Diag” possui os seguintes campos para diagnósticos dos canais de sincronismo NETA:

DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.bExchangeSync: Quando esta variável de diagnóstico estiver com o valor igual a FALSE não é possível definir o estado de alguns outros diagnósticos, como por exemplo bIncompatibleFirmware, bApplicationProjectDiff e bProjectArchiveDiff. Não irão representar o valor correto pois dependem do correto funcionamento da comunicação entre os dois CPs para que a informação possa ser corretamente gerada.

Variável Direta		Variável AT	Descrição
Variável	Bit	DG_NX4010.tRedundancy.RedDgnLoc.sNETA_Diag.*	
%QB(n+33)	0	bGeneralFailure	TRUE – O canal de sincronismo possui algum tipo de falha. Os 3 próximos diagnósticos indicarão a falha específica.
			FALSE – O canal de sincronismo está em correto funcionamento.
	1	bInternalFailure	TRUE – A falha detectada tem sua causa localizada dentro deste CP. Tais falhas são tratadas de forma diferenciada.
			FALSE – O módulo NX4010 está em correto funcionamento.
	2	bLinkDownFailure	TRUE – O cabo AL-2319A está desconectado do módulo NX4010 ou rompido, em um dos 2 CPs.
			FALSE – O cabo AL-2319A está conectado ao módulo NX4010.
	3	bTimeoutFailure	TRUE – Esta falha é reportada caso um serviço de sincronização não tenha terminado com sucesso até um time-out especificado, e não tenham sido encontradas falhas do tipo bInternalFailure ou bLinkDownFailure que justificassem isso.
			FALSE – O módulo NX4010 está em correto funcionamento.
	4 .. 7	bReserved[4..7]	Reservados.

Tabela 207: Diagnósticos Específicos da Interface NETA

A sub-estrutura “sNETB_Diag” possui os seguintes campos para diagnósticos dos canais de sincronismo NETB:

Variável Direta		Variável AT DG_NX4010.tRedundancy. RedDgnLoc.sNETB_Diag.*	Descrição
Variável	Bit		
%QB(n+34)	0	bGeneralFailure	TRUE – O canal de sincronismo possui algum tipo de falha. Os 3 próximos diagnósticos indicarão a falha específica. FALSE – O canal de sincronismo está em correto funcionamento.
	1	bInternalFailure	TRUE – A falha detectada tem sua causa localizada dentro deste CP. Tais falhas são tratadas de forma diferenciada. FALSE – O módulo NX4010 está em correto funcionamento.
	2	bLinkDownFailure	TRUE – O cabo AL-2319B está desconectado do módulo NX4010 ou rompido, em um dos 2 CPs. FALSE – O cabo AL-2319B está conectado ao módulo NX4010.
	3	bTimeoutFailure	TRUE – Esta falha é reportada caso um serviço de sincronização não tenha terminado com sucesso até um time-out especificado, e não tenham sido encontradas falhas do tipo bInternalFailure ou bLinkDownFailure que justificassem isso. FALSE – O módulo NX4010 está em correto funcionamento.
	4 .. 7	bReserved[4..7]	Reservados.

Tabela 208: Diagnósticos Específicos da Interface NETB

A sub-estrutura “sNET_Stat” contém estatísticas de falhas e sucessos dos serviços. As estatísticas dos CPs local e remoto podem ser reiniciadas através dos comandos:

```
//CP Local
DG_NX4010.tRedundancy.RedCmdLoc.bResetNETStatisticsLocal := TRUE;
//CP Remoto
DG_NX4010.tRedundancy.RedCmdLoc.bResetNETStatisticsRemote := TRUE;
```

A sub-estrutura contém os seguintes contadores:

Variável Direta	Variável AT DG_NX4010.tRedundancy. RedDgnLoc.sNET_Stat.*	Descrição
%QW(n+35)	wSuccessExchDgCmdSync	Contagem de sucessos do serviço Troca de Diagnósticos e Comandos. (0 a 65535)
%QW(n+37)	wFailedExchDgCmdSync	Contagem de falhas do serviço Troca de Diagnósticos e Comandos. (0 a 65535)
%QW(n+39)	wSuccessRedDataSync	Contagem de sucessos do serviço Sincronização de Dados Redundantes. (0 a 65535)
%QW(n+41)	wFailedRedDataSync	Contagem de falhas do serviço Sincronização de Dados Redundantes. (0 a 65535)

Variável Direta	Variável AT DG_NX4010.tRedundancy.RedDgnLoc.sNET_Stat.*	Descrição
%QW(n+43)	wSuccessRedForceSync	Contagem de sucessos do serviço Sincronização de Lista de Forçamentos Redundantes. (0 a 65535)
%QW(n+45)	wFailedRedForceSync	Contagem de falhas do serviço Sincronização de Lista de Forçamentos Redundantes. (0 a 65535)

Tabela 209: Diagnósticos Específicos da Interface

Nota:

Contadores: Todos os contadores retornam a zero quando o seu valor limite é ultrapassado.

A sub-estrutura “sGeneral_DiagExt” possui os seguintes campos para diagnósticos gerais da redundância:

Variável Direta		Variável AT DG_NX4010.tRedundancy.Red	Descrição
Variável	Bit		
	0	bFailedPBUS3A	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) na rede PROFIBUS 3 A. O modo ativo (comunicando com escravos) é assumido pelo CP Ativo. O modo passivo (comunicando com o mestre em modo ativo) é assumido pelo CP Não-Ativo. Esta falha também pode ser indicada caso o módulo NX5001 tenha falha em seu microprocessador, ou caso não consiga se comunicar com a UCP NX3030 via barramento.
			FALSE – Não existem falhas na rede PROFIBUS 3 A.
	1	bFailedPBUS3B	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) na rede PROFIBUS 3 B. O modo ativo (comunicando com escravos) é assumido pelo CP Ativo. O modo passivo (comunicando com o mestre em modo ativo) é assumido pelo CP Não-Ativo. Esta falha também pode ser indicada caso o módulo NX5001 tenha falha em seu microprocessador, ou caso não consiga se comunicar com a UCP NX3030 via barramento.
FALSE – Não existem falhas na rede PROFIBUS 3 B.			
	2	bFailureProfibus_3	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) na rede PROFIBUS 3. Caso a rede PROFIBUS 3 seja redundante, FailureProfibus_3 resulta de um “E lógico” entre FailedPBUS3A e FailedPBUS3B. Caso a rede PROFIBUS 3 não seja redundante, FailureProfibus_3 é uma cópia de FailedPBUS3A.
			FALSE – Não existem falhas na rede PROFIBUS 3.

Variável Direta		Variável AT DG_NX4010.tRedundancy.Red	Descrição	
Variável	Bit			
%QB(n+47)	3	bFailedPBUS4A	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) na rede PROFIBUS 4 A. O modo ativo (comunicando com escravos) é assumido pelo CP Ativo. O modo passivo (comunicando com o mestre em modo ativo) é assumido pelo CP Não-Ativo. Esta falha também pode ser indicada caso o módulo NX5001 tenha falha em seu microprocessador, ou caso não consiga se comunicar com a UCP NX3030 via barramento.	
			FALSE – Não existem falhas na rede PROFIBUS 4 A.	
	4	bFailedPBUS4B	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) na rede PROFIBUS 4 B. O modo ativo (comunicando com escravos) é assumido pelo CP Ativo. O modo passivo (comunicando com o mestre em modo ativo) é assumido pelo CP Não-Ativo. Esta falha também pode ser indicada caso o módulo NX5001 tenha falha em seu microprocessador, ou caso não consiga se comunicar com a UCP NX3030 via barramento.	
			FALSE – Não existem falhas na rede PROFIBUS 4 B.	
	5	bFailureProfibus_4	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) na rede PROFIBUS 4. Caso a rede PROFIBUS 4 seja redundante, FailureProfibus_4 resulta de um “E lógico” entre FailedPBUS4A e FailedPBUS4B. Caso a rede PROFIBUS 4 não seja redundante, FailureProfibus_4 é uma cópia de FailedPBUS4A.	
			FALSE – Não existem falhas na rede PROFIBUS 4.	
	6	bFailureCommBusAny	TRUE – Existe falha em alguma rede de campo de comunicação, como por exemplo, uma porta ethernet com protocolo Modbus configurada para falha vital.	
			FALSE – Nenhuma rede de campo apresenta falha.	
	7	bFailureCommBusAll	TRUE – Existe falha em todas as redes de campo de comunicação. Por exemplo, NET1 e NET2 são configuradas para vital e ambas estão em falha.	
			FALSE – há pelo menos uma rede de campo de comunicação funcionando sem falhas.	
	%QB(n+48)	-	byFailedCommBusCount	Quantidade de redes de campo de comunicação reportando falhas.

Variável Direta		Variável AT DG_NX4010.tRedundancy.Red	Descrição
Variável	Bit		
%QB(n+49)	0	bRemCpuKeepAliveNet1	TRUE – A UCP NX3030 está recebendo pacotes de Keep Alive da UCP do outro half-cluster, através da NET1. Os pacotes de Keep Alive só são enviados em projetos que não utilizam painel PX2612 e que não têm nenhuma rede PROFIBUS. FALSE – Pacotes de Keep Alive não estão sendo recebidos.
	1	bRemCpuKeepAliveNet2	TRUE – A UCP NX3030 está recebendo pacotes de Keep Alive da UCP do outro half-cluster, através da NET2. FALSE – Pacotes de Keep Alive não estão sendo recebidos.
	2..7	(bits ocultos reservados)	Bits reservados para uso futuro. Não aparecem na estrutura simbólica (ocultos).
%QB(n+50)	-	abyReservedBytes[1..5]	5 Bytes reservados para uso futuro. Não aparecem na estrutura simbólica (ocultos).

Tabela 210: Diagnósticos Gerais da Redundância, Parte Estendida

6.6.4.2. Comandos da Redundância

Os campos de comandos das estruturas RedCmdLoc e RedCmdRem possuem sempre um sufixo que pode ser Local ou Remote. Por exemplo, existem os campos de comando StandbyLocal e StandbyRemote, que têm efeito equivalente ao botão STAND-BY do painel PX2612.

Um comando com sufixo Local gerado em RedCmdLoc será executado no próprio CP (local). Por outro lado, um comando com sufixo Remote gerado em RedCmdLoc será executado no outro CP (remoto). Isto funciona da seguinte maneira:

- O CP remoto, a cada ciclo da MainTask, recebe uma cópia de RedCmdLoc do CP local via NETA / NETB, e esta cópia é chamada de RedCmdRem no CP remoto
- O CP remoto somente executa comandos de RedCmdRem que tenham o sufixo Remote

Exemplo 1: se o CP local estiver em estado Ativo, e deseja-se chaveá-lo para o estado Reserva, pode-se ligar o bit DG_NX4010.tRedundancy.RedCmdLoc.bStandbyLocal no CP local.

Exemplo 2: se o CP remoto estiver em estado Ativo, e deseja-se chaveá-lo para o estado Reserva, pode-se ligar o bit DG_NX4010.tRedundancy.RedCmdLoc.bStandbyRemote no CP local. Isto pode ser útil, por exemplo, se a comunicação de um sistema SCADA não está disponível temporariamente com o CP remoto. Neste caso, o comando é escrito pelo SCADA no CP local, que o repassa para o CP remoto via NETA / NETB.

ATENÇÃO

Se o diagnóstico DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.bExchangeSync estiver indicando falha no serviço Troca de Diagnósticos e Comandos, um comando com sufixo Remote não poderá ser repassado para o CP remoto, e portanto não será executado.

Para disparar um comando, deve-se sempre ligar o bit correspondente em RedCmdLoc. Isto pode ser feito por um sistema SCADA, fazendo uma escrita via MasterTool, ou até mesmo ligando o bit dentro de uma POU como ActivePrg ou NonSkippedPrg.

O usuário não precisa se preocupar com o desligamento do bit de comando, que será feito automaticamente pelo gerenciador de redundância:

- No caso de comandos executados no CP local (RedCmdLoc + comando com sufixo Local), o bit é desligado assim que o comando for percebido e executado
- No caso de comandos executados no CP remoto (RedCmdRem + comando com sufixo Remote):
 - no CP remoto, o comando é executado quando o gerenciador de redundância percebe uma borda de subida no bit de comando

- no CP local onde o comando foi gerado, o bit é desligado automaticamente no próximo ciclo da MainTask

ATENÇÃO

Existem dois bits de comando que normalmente devem ser desligados pelo usuário: DG_NX4010.tRedundancy.RedCmdLoc.bTestModeLocal e DG_NX4010.tRedundancy.RedCmdLoc.bTestRelayLocal. Ver maiores detalhes sobre estes comandos adiante nesta seção. Caso o usuário esqueça de desligá-los, existem mecanismos automáticos previstos para desligá-los.

É importante salientar que quaisquer ligamentos ou desligamentos de comandos serão registrados nos logs de eventos da redundância, o que é importante para avaliação histórica, por exemplo, para determinar as causas de um switch-over.

A seguir, são definidos os campos das estruturas RedCmdLoc e RedCmdRem:

Variável Direta		Variável AT DG_NX4010.tRedundancy. RedCmdLoc.*	Descrição
Variável	Bit		
%QB(n+55)	0	bButtonTurnOnLocal	TRUE – Trata-se de uma cópia processada do botão TURN ON PLCx lido do painel PX2612. Este bit só é ligado um segundo depois do pressionamento do botão, e desligado imediatamente ao liberar o botão. É importante ressaltar que esse bit será ligado quando o botão TURN ON do CP remoto estiver pressionado, visto que esse tipo de comando sempre é enviado do CP remoto.
			FALSE – O botão TURN ON PLCx não está pressionado.
	1	bButtonStandbyLocal	TRUE – Trata-se de uma cópia processada do botão STAND-BY lido do painel PX2612. Este bit só é ligado um segundo depois do pressionamento do botão, e desligado imediatamente ao liberar o botão.
			FALSE – O botão STAND-BY não está pressionado.
	2	bButtonInactiveLocal	TRUE – Trata-se de uma cópia processada do botão INACTIVE lido do painel PX2612. Este bit só é ligado um segundo depois do pressionamento do botão, e desligado imediatamente ao liberar o botão.
			FALSE – O botão INACTIVE não está pressionado.
	3	bAutoConfigLocal	TRUE – Este diagnóstico informa uma solicitação de configuração automática, necessária para deixar o estado Não-Configurado em algumas situações.
			FALSE – Solicitação de configuração automática desabilitada.
	4	bTurnOnLocal	TRUE – Este comando produz uma ação equivalente ao botão TURN ON PLCX do PX2612, no CP local.
			FALSE – O botão TURN ON PLCx do CP local não está pressionado.

Variável Direta		Variável AT DG_NX4010.tRedundancy. RedCmdLoc.*	Descrição
Variável	Bit		
	5	bStandbyLocal	TRUE – Este comando produz uma ação equivalente ao botão STAND-BY do PX2612, no CP local.
			FALSE – O botão STAND-BY do CP local não está pressionado.
	6	bInactiveLocal	TRUE – Este comando produz uma ação equivalente ao botão INACTIVE do PX2612, no CP local.
			FALSE – O botão INACTIVE do CP local não está pressionado.
	7	bResetNETStatisticsLocal	TRUE – Este comando reseta as estatísticas de NETA / NETB (veja sub-estrutura SNET_Stat em RedDgnLoc e RedDgnRem). Tais estatísticas são contadores de falhas e sucessos em serviços de sincronização.
			FALSE – O comando de reset das estatísticas de NETA / NETB no CP local não foi acionado.
0	bTestModeLocal	TRUE – Este comando coloca o painel PX2612 em modo teste, permitindo que seus componentes sejam testados (LEDs, relés e botões), conforme explica-se na seção Teste do Painel PX2612 . O modo teste do PX2612 só é efetivamente aceito quando este bit é ligado nos dois CPs. Ou seja, para que o PX2612 esteja em modo teste, o CP verifica se RedCmdLoc.bTestModeLocal e se RedCmdRem.bTestModeLocal estão ambos ligados. O diagnóstico RedDgnLoc.bRedPanelTestMode é ligado para informar que o PX2612 está realmente em modo teste. Normalmente, o usuário deve desligar o bit bTestModeLocal nos 2 CPs assim que concluir os testes do PX2612, mas caso esqueça de fazer isto, o bit será desligado automaticamente 15 minutos depois de ser ligado.	
		FALSE – O comando que coloca o painel PX2612 em modo teste está desativado.	

Variável Direta		Variável AT DG_NX4010.tRedundancy. RedCmdLoc.*	Descrição
Variável	Bit		
%QB(n+56)	1	bTestRelayLocal	TRUE – Este comando é usado para testar o relé NA do PX2612, e consequentemente também o relé NF externo, usados para um eventual desligamento do outro CP. Este comando só é aceito enquanto o PX2612 está em modo teste, sendo automaticamente desligado e ignorado caso o PX2612 não esteja em modo teste. Normalmente, o usuário deve desligar o bit bTestRelayLocal assim que concluir o teste do relé, mas caso esqueça de fazer isto, o bit será desligado assim que o modo teste for encerrado. É importante também salientar que este comando só é aceito no CP Ativo, para evitar que o CP Não-Ativo desligue o CP Ativo.
			FALSE – O comando utilizado para testar o relé NA do PX2612 está desativado.
	2	bStandbyRemote	TRUE – Este comando produz uma ação equivalente ao botão STAND-BY do PX2612, no CP remoto.
			FALSE – O botão STAND-BY do CP remoto não está pressionado.
	3	bInactiveRemote	TRUE – Este comando produz uma ação equivalente ao botão INACTIVE do PX2612, no CP remoto.
			FALSE – O botão INACTIVE do CP local não está pressionado.
	4	bResetNETStatisticsRemote	TRUE – Produz uma ação equivalente ao comando ResetNETStatisticsLocal, porém desta vez no CP remoto.
			FALSE – O comando de reset das estatísticas de NETA / NETB no CP remoto não foi acionado.
	5..7	bReserved[5..7]	Reservados.

Tabela 211: Comandos da Redundância

6.6.4.3. Informações do Usuário Trocadas entre CPA e CPB

O serviço de sincronização Troca de Diagnósticos e Comando, troca as seguintes estruturas de dados entre os dois CPs em cada ciclo da MainTask, usando os canais de sincronismo NETA / NETB:

- Diagnósticos de Redundância (RedDgnLoc e RedDgnRem), já discutidos na seção [Estrutura de Diagnósticos da Redundância](#)
- Comandos de Redundância (RedCmdLoc e RedCmdRem), já discutidos na seção [Comandos da Redundância](#)
- Informações do Usuário Trocadas entre CPA e CPB (RedUsrLoc e RedUsrRem), que serão discutidas nesta seção

As estruturas RedUsrLoc e RedUsrRem são simplesmente um array de 128 bytes, cuja utilização pode ser livremente definida pelo usuário. Elas permitem que o usuário transfira, a cada ciclo, 128 bytes de informação do CPA para o CPB, e outros 128 bytes do CPB para o CPA.

RedUsrRem é uma cópia de RedUsrLoc do outro CP, recebida via NETA / NETB. Determinado CP escreve informações em RedUsrLoc, que serão lidas no outro CP em RedUsrRem.

Estas estruturas de dados podem ter diversas utilidades. Por exemplo, supondo que o sistema SCADA se conecte somente ao CP Ativo, e que deseja-se visualizar algumas informações do CP Não-Ativo, o CP Não-Ativo pode colocar estas informações nesta estrutura de dados. Entre tais informações, por exemplo, podem constar alguns diagnósticos que não estejam mapeados em RedDgnLoc e RedDgnRem.

6.6.4.4. Diagnósticos MODBUS utilizados na Redundância

Para verificar se existe falha em todos os MODBUS Servidores configurados em uma instância MODBUS Cliente, existe um diagnóstico específico em cada instância MODBUS Cliente configurada. Na tabela abaixo, é exibido o diagnóstico para este tipo de falha em uma instância MODBUS Cliente chamada MODBUS_Symbol_Client.

Variável DG_MODBUS_Symbol_ Client.tDiag.*	Descrição
bAllDevicesCommFailure	TRUE – Todos os Servidores configurados neste Cliente estão apresentando erro.
	FALSE – Existe pelo menos um Servidor configurado neste Cliente que não está apresentando erro.

Tabela 212: Diagnóstico MODBUS Cliente

Quando configurado com falha vital Ethernet, este diagnóstico é consultado e 3 segundos após a alteração do seu estado de FALSE para TRUE, ocorre um switch-over caso as outras condições para este evento estejam satisfeitas.

6.6.4.5. Log de Eventos da Redundância

O MasterTool permite observar diversos logs para um CP Nexto, entre os quais encontra-se os Log de Eventos da Redundância. Estas mensagens, específicas para a redundância, registram no Log de Sistema modificações relevantes em campos das estruturas de dados de diagnósticos e comandos de redundância..

Cada linha mostrada no log possui as seguintes colunas:

- Marca de Tempo: data e hora do evento, com resolução de milissegundos
- Severidade: informação, advertência, erro ou exceção
- Descrição: texto que descreve o evento
- Componente: componente que gerou o evento, que no caso do Log de Eventos da Redundância, será o “*Redundancy Management*”

O texto na coluna “*Descrição*” contém informações sobre o evento que ocorreu.

Um exemplo da coluna *Descrição* poderia ser o seguinte:

Redundancy new state (local): Starting

Para acessar esta tela, deve-se dar um duplo clique sobre o dispositivo (NX3030) na árvore de dispositivos, e depois selecionar a aba “*Log*”. Existe um filtro que permite selecionar somente o componente “*Redundancy Management*”, para exibir somente os eventos de redundância.

ATENÇÃO

Alguns diagnósticos podem apontar possíveis falhas durante a inicialização do sistema redundante e nos primeiros ciclos das tarefas. Mas, em um correto funcionamento do sistema, estes diagnósticos voltam a indicar a ausência de erros logo após a inicialização do sistema.

6.6.5. Teste do Painel PX2612

O painel PX2612 é constituído de botões, LEDs e relés. Muitos destes recursos são pouco utilizados, e portanto raramente são testados para perceber que possuem algum defeito. É importante que, periodicamente, se façam testes para verificar se estes recursos estão funcionando, para evitar que falhas permaneçam ocultas e impeçam o uso do PX2612 quando o mesmo for necessário.

6.6.5.1. Entrada no Modo Teste

O primeiro passo para testar o PX2612 é colocá-lo em modo teste. Isto é feito ligando o bit de comando DG_NX4010.tRedundancy.RedCmdLoc.bTestModeLocal nos dois CPs.

Determinado CP perceberá que está em modo teste sempre que os dois seguintes bits estiverem ligados:

- DG_NX4010.tRedundancy.RedCmdLoc.bTestModeLocal (RedCmdLoc.bTestModeLocal ligado neste CP)
- DG_NX4010.tRedundancy.RedCmdRem.bTestModeLocal (RedCmdLoc.bTestModeLocal ligado no outro CP)

Quando estes dois bits estão ligados, o CP liga o bit de diagnóstico DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.bRedPanelTestMode, para informar que o PX2612 está em modo teste.

6.6.5.2. Saídas Manual e Automática do Modo Teste

O usuário pode encerrar o modo teste manualmente, desligando o bit DG_NX4010.tRedundancy.RedCmdLoc.bTestModeLocal nos dois CPs. Na verdade, basta desligá-lo em um dos CPs, pois o modo teste exige que este bit esteja ligado nos dois CPs. No entanto, aconselha-se desligá-lo nos dois CPs.

Caso o usuário esqueça de desligar DG_NX4010.tRedundancy.RedCmdLoc.bTestModeLocal, este bit é desligado automaticamente, 15 minutos depois de ser ligado, encerrando automaticamente o modo teste.

6.6.5.3. Teste dos LEDs

Durante o modo teste, os seis LEDs devem piscar, perdendo sua utilidade normal, que é mostrar o estado da redundância.

6.6.5.4. Teste dos Botões

Ao pressionar um botão no modo teste, um LED correspondente irá parar de piscar, e ficará aceso. A tabela abaixo mostra a correspondência entre o botão pressionado e o LED que permanecerá aceso.

Botão Testado	LED Correspondente
TURN ON PLC A	ACTIVE PLC B
STAND-BY PLC A	STAND-BY PLC A
INACTIVE PLC A	INACTIVE PLC A
TURN ON PLC B	ACTIVE PLC A
STAND-BY PLC B	STAND-BY PLC B
INACTIVE PLC B	INACTIVE PLC B

Tabela 213: Correspondência entre Botões e LEDs no Teste de Botões do PX2612

Observa-se que, normalmente, o LED fica ao lado do botão pressionado, exceto para os botões TURN ON PLCx.

Antes de o LED permanecer aceso, é necessário manter o botão pressionado por, no mínimo, um segundo. O LED volta a piscar assim que o botão for liberado.

Durante o modo teste, os botões não permitem executar as funções que executariam fora do modo teste, tais como provocar uma mudança de estado de redundância.

6.6.5.5. Teste dos Relés

Para testar os relés, foi criado o bit de comando DG_NX4010.tRedundancy.RedCmdLoc.bTestRelayLocal. Ao ligar este bit, se este CP estiver em modo teste, e além disso estiver em estado Ativo, ele acionará o relé, o que deverá provocar o desligamento do outro CP (Não-Ativo). Desligando DG_NX4010.tRedundancy.RedCmdLoc.bTestRelayLocal, o relé é liberado, permitindo o religamento do outro CP.

O comando não tem efeito no CP Não-Ativo, para evitar que o mesmo desligue o CP Ativo.

O usuário deverá provocar um switch-over entre CPs (Ativo X Não-Ativo), para testar o relé nos dois CPs.

Quando o CP que foi desligado é religado e reinicializado, ele volta com DG_NX4010.tRedundancy.RedCmdLoc.bTestModeLocal desligado, portanto, o modo teste está cancelado. Deve-se ligar novamente o bit DG_NX4010.tRedundancy.RedCmdLoc.bTestModeLocal neste CP, e passá-lo para o estado Ativo, antes de testar o seu relé.

Quando o modo teste é encerrado, o bit de comando DG_NX4010.tRedundancy.RedCmdLoc.bTestRelayLocal é automaticamente desligado, caso o usuário o tenha esquecido ligado.

6.6.5.6. Sequência Sugerida para Executar os Testes do PX2612

A seguinte sequência é sugerida para executar os testes do PX2612:

- Ligar o bit de comando DG_NX4010.tRedundancy.RedCmdLoc.bTestModeLocal nos dois CPs (CPA e CPB).
- Deve-se observar que os 6 LEDs estão piscando.
- Pressionar, um a um, os 6 botões, e verificar se o LED correspondente para de piscar e permanece aceso enquanto o botão está pressionado. Deve-se lembrar que o botão deve ficar pressionado no mínimo por um segundo antes que o LED pare de piscar e fique aceso, e que o LED voltará a piscar assim que o botão for liberado.
- Ligar o bit de comando DG_NX4010.tRedundancy.RedCmdLoc.bTestRelayLocal no CP Ativo. Deve-se observar o desligamento do CP Não-Ativo.
- Desligar o bit de comando DG_NX4010.tRedundancy.RedCmdLoc.bTestRelayLocal no CP Ativo. Deve-se observar o religamento do CP Não-Ativo.
- Aguardar que o CP Não-Ativo seja reinicializado e assumo o estado Reserva. O modo teste não estará ativo, pois o bit DG_NX4010.tRedundancy.RedCmdLoc.bTestModeLocal foi desligado na inicialização do CP que agora está em Reserva.
- Provocar um switch-over entre CPs, pressionando o botão STAND-BY do CP Ativo. O uso normal do botão STAND-BY é possível porque o modo teste não está ativo.
- Ligar o bit de comando DG_NX4010.tRedundancy.RedCmdLoc.bTestModeLocal no novo CP Ativo, que acabou de sair do modo Reserva. Desta forma, o modo teste volta a estar ativo. Caso o bit do CP Reserva esteja desligado, o mesmo deve ser ligado novamente.
- Ligar o bit de comando DG_NX4010.tRedundancy.RedCmdLoc.bTestRelayLocal no CP Ativo. Deve-se observar o desligamento do CP Não-Ativo.
- Desligar o bit de comando DG_NX4010.tRedundancy.RedCmdLoc.bTestRelayLocal no CP Ativo. Deve-se observar o religamento do CP Não-Ativo.
- Desligar o bit de comando DG_NX4010.tRedundancy.RedCmdLoc.bTestModeLocal no CP Ativo, para encerrar o modo teste. Não é necessário fazer isto no CP Reserva, pois o mesmo acabou de ser inicializado, com o bit DG_NX4010.tRedundancy.RedCmdLoc.bTestModeLocal desligado.

7. Manutenção

Uma das características da Série Nexto é a geração de diagnósticos de comportamentos atípicos, sejam eles falhas, erros ou modos de operação, possibilitando ao operador identificar e solucionar problemas que venham a ocorrer com o sistema com grande facilidade.

As UCPs Nexto proporcionam diversas maneiras de se visualizar os diagnósticos gerados pelo sistema, são elas:

- [One Touch Diag](#)
- [Diagnósticos via LED](#)
- [Diagnósticos via Página Web de Sistema](#)
- [Diagnósticos via Variáveis](#)
- [Diagnósticos via Blocos Funcionais](#)

O primeiro deles é uma característica inovadora da Série Nexto, a qual possibilita um rápido acesso às condições anormais da aplicação. O segundo é puramente visual, gerado através de dois LEDs presentes no painel (DG e WD) e também dos LEDs presentes no conector RJ45 (exclusivos para a conexão Ethernet). A próxima característica é a visualização gráfica do bastidor e dos respectivos módulos configurados na aba *Informações do Barramento* da Página Web de Sistema, sendo permitido o acesso individual do estado de operação e os diagnósticos ativos. Os diagnósticos também são armazenados diretamente em variáveis da UCP, tanto de representação direta (%Q), como as simbólicas mapeadas através da diretiva AT, e podem ser utilizados pela aplicação do usuário, por exemplo, sendo apresentados em um sistema de supervisão. Os últimos apresentam condições específicas de funcionamento do sistema.

A função destes diagnósticos é apontar possíveis problemas de instalação ou configuração do sistema, e de problemas ou deficiências das redes de comunicação. O capítulo de Manutenção deve ser consultado pelo usuário sempre que necessário.

7.1. Diagnósticos do Módulo

7.1.1. One Touch Diag

One Touch Diag (OTD), ou seja, diagnóstico com um único pressionamento, é uma característica exclusiva que a Série Nexto traz para os controladores programáveis. Com este novo conceito, o usuário pode verificar os diagnósticos de qualquer módulo presente no sistema diretamente no visor gráfico da UCP, com um único pressionamento na tecla de diagnóstico do respectivo módulo. Essa é uma poderosa ferramenta de diagnóstico que pode ser usada offline (sem sistema de supervisão ou software programador), tornando mais fácil encontrar e resolver rapidamente possíveis problemas.

A tecla de diagnóstico está localizada na parte superior da UCP, em local de fácil acesso, e, além de fornecer os diagnósticos ativos, permite o acesso ao menu de navegação, descrito no capítulo [Configuração – Menu Informativo e de Configuração da UCP](#).

A figura abaixo mostra a localização da tecla na UCP:

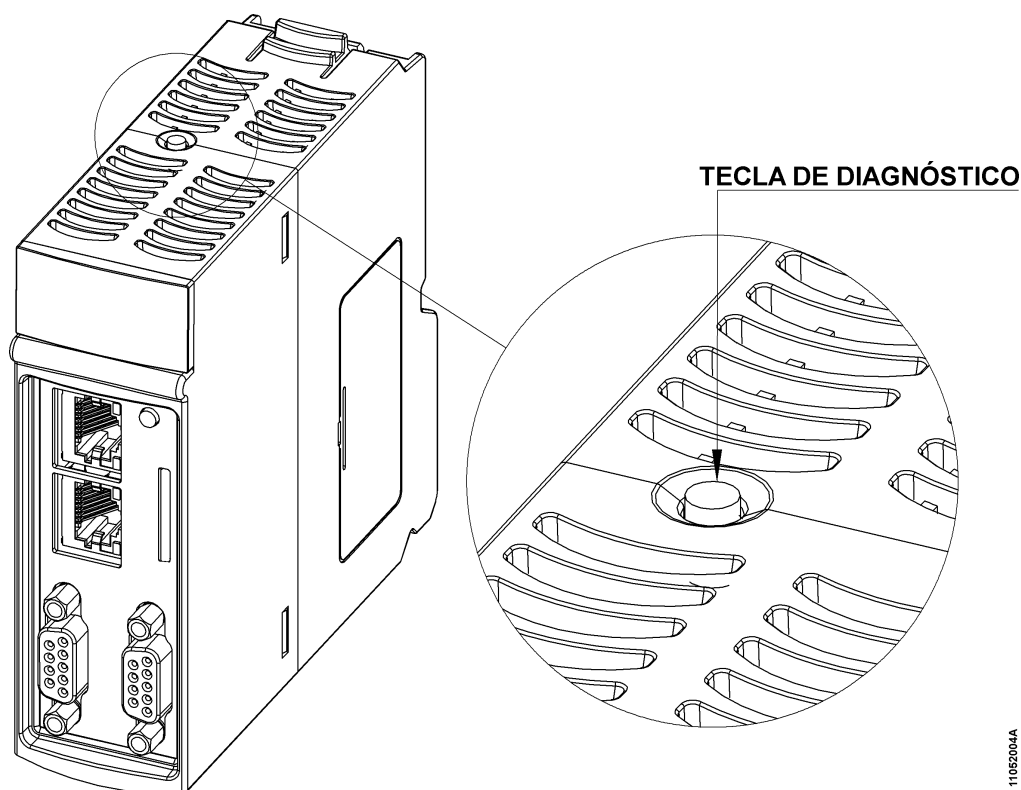


Figura 183: Tecla de Diagnóstico

Com apenas um pressionamento curto, a UCP começa a mostrar os diagnósticos do barramento (quando ativos, caso contrário exibe a mensagem "SEM DIAG"). Inicialmente, será visualizada a Tag (configurada nas propriedades do módulo no software MasterTool IEC XE, seguindo a IEC 61131-3), ou seja, o nome atribuído à UCP, em seguida, serão mostrados todos os diagnósticos, através de mensagens no visor da UCP. Esse processo será executado por duas vezes no visor. Tudo ocorre de forma automática, sendo que o usuário somente deverá executar o pressionamento curto inicial e a UCP será responsável por exibir os diagnósticos. Os diagnósticos de outros módulos presentes no barramento também serão exibidos no visor gráfico da UCP, através de um pressionamento curto na tecla de diagnóstico dos mesmos, no mesmo modelo da apresentação dos diagnósticos da UCP.

A figura abaixo mostra todo o processo a partir do pressionamento curto, sendo a condição e os tempos da UCP representados nos retângulos menores. É importante salientar que os diagnósticos poderão ter mais de uma tela, ou seja, o tempo especificado no fluxograma abaixo é válido para cada uma delas.

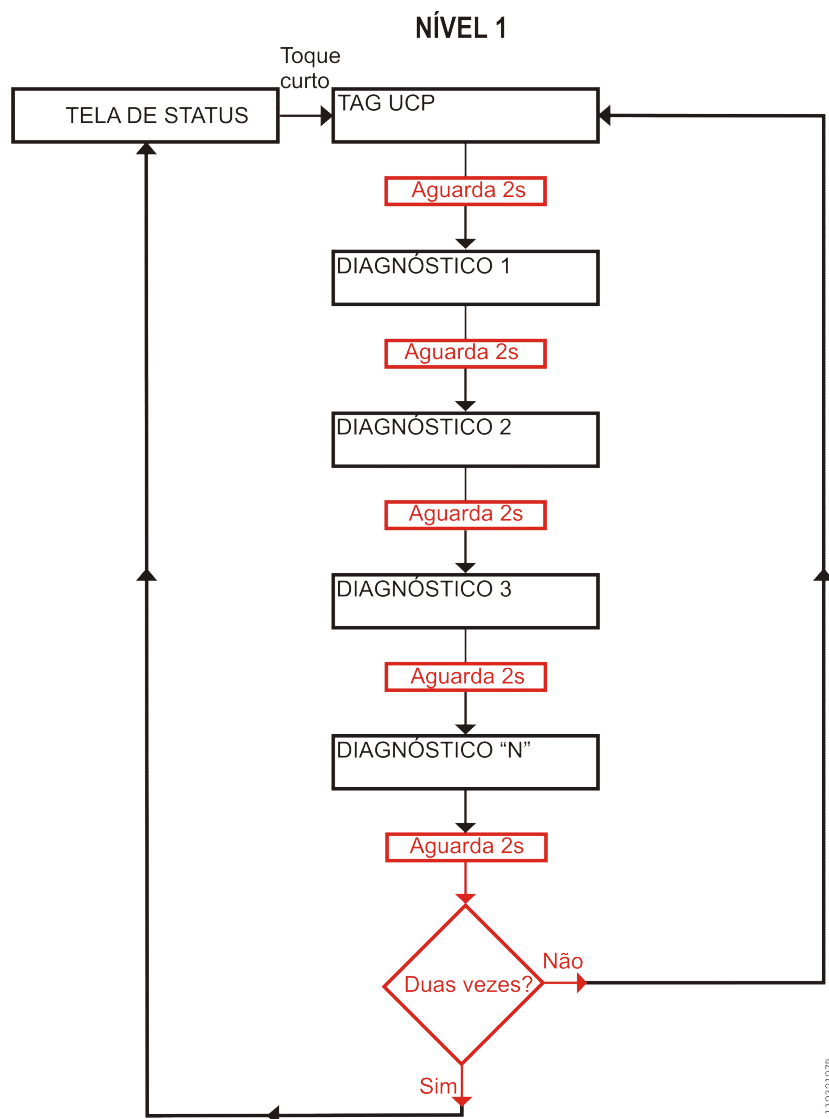


Figura 184: Visualização dos Diagnósticos da UCP

Para finalizar, antes de todo o processo de visualização ser efetuado, basta dar um pressionamento curto na tecla de diagnóstico, em qualquer momento, ou pressionar a tecla diagnóstico de algum módulo de E/S presente no barramento. Também, é importante notar que o One Touch Diag só estará disponível quando o módulo estiver em modo operacional.

Caso seja efetuado um pressionamento longo, a UCP entrará no menu de navegação, o qual está descrito no capítulo [Configuração – Menu Informativo e de Configuração da UCP](#).

A tabela abaixo mostra a diferença entre os tempos do pressionamento curto, pressionamento longo e tecla presa.

Tipo de Pressionamento	Tempo Mínimo	Tempo Máximo	Condição para Indicação
Sem pressionamento	-	59,99 ms	-
Pressionamento curto	60 ms	0,99 s	Liberação
Pressionamento longo	1 s	20 s	Mais do que 1 s até 20 s
Tecla Presa	20,01 s	(∞)	Indicação em diagnóstico, consultar a Tabela 219

Tabela 214: Tempo de um Pressionamento

As mensagens exibidas no visor gráfico das UCPs Nexto, correspondentes aos diagnósticos, estão descritas na seção

Diagnósticos via Variáveis, na Tabela 219.

Caso ocorra uma situação de tecla presa de algum módulo de E/S presente no barramento, a tecla de diagnóstico deste deixará de indicar diagnósticos no visor gráfico da UCP quando for pressionada, neste caso a UCP irá indicar que há módulos com diagnóstico ativo. Para que seja possível eliminar este diagnóstico da UCP, será necessário realizar uma troca a quente do módulo com o diagnóstico ativo.

Para mais detalhes sobre o procedimento de visualização dos diagnósticos da UCP ou de outros módulos do barramento, ver descrição no Manual de Utilização Série Nexto – MU214000.

7.1.2. Diagnósticos via LED

Este produto possui um LED para indicação de diagnóstico (LED DG) e um LED para indicar ocorrência de cão-de-guarda (LED WD). A Tabela 215 e Tabela 216 mostram o significado de cada estado e suas respectivas descrições.

7.1.2.1. DG (Diagnóstico)

Verde	Vermelho	Descrição	Causas	Prioridade
Desligado	Desligado	Não utilizado	Sem fonte de alimentação. Problema de Hardware	-
Ligado	Desligado	Aplicação em Execução (Modo Run)	-	3 (Baixa)
Desligado	Ligado	Aplicação Parada (Modo Stop)	-	3 (Baixa)
Piscando 2x	Desligado	Módulos do barramento com diagnóstico	No mínimo um módulo do barramento, incluindo a UCP, está com algum diagnóstico ativo	1
Piscando 3x	Desligado	Forçamento de dados	Alguma área de memória está sendo forçada pelo usuário via MasterTool IEC XE	2
Desligado	Piscando 4x	Erro de configuração ou de hardware no barramento.	O barramento está danificado ou não está configurado corretamente.	0 (Alta)

Tabela 215: Descrição dos Estados do LED de Diagnóstico

7.1.2.2. WD (Cão-de-guarda)

LED Vermelho	Descrição	Causas	Prioridade
Desligado	Sem indicação de cão-de-guarda	Operação normal	3 (Baixa)
Piscando 1x	Cão-de-guarda de Software	Cão-de-guarda da aplicação de usuário	2
Ligado	Cão-de-guarda de Hardware	Módulo danificado e/ou sistema operacional corrompido	1 (Alta)

Tabela 216: Descrição dos Estados do LED de Cão-de-guarda

Notas:

Cão-de-guarda de software: Para remover a indicação de cão-de-guarda, deve-se efetuar um reset da aplicação ou desligar e ligar novamente a UCP. Esse cão-de-guarda ocorre quando o tempo de execução da aplicação de usuário for maior que o tempo de cão-de-guarda configurado.

O diagnóstico pode ser verificado no operando `Exception.wExceptionCode`, ver Tabela 223.

Cão-de-guarda de hardware: Para limpar qualquer indicação de cão-de-guarda, como no LED WD ou no operando `Reset.bWatchdogReset`, o módulo deve ser desconectado da fonte de alimentação.

Para verificar as condições da aplicação na reinicialização do módulo, ver configurações na Tabela 44.

7.1.2.3. LEDs Conector RJ45

Os dois LEDs presentes nos conectores RJ45, auxiliam o usuário na detecção de problemas na rede física instalada, indicando a velocidade do link de rede e a existência de tráfego de comunicação com a interface. O significado dos LEDs é apresentado na tabela abaixo.

Amarelo	Verde	Descrição
○	○	Ausência de LINK de rede
●	○	LINK de rede de 10 Mbits/s
●	●	LINK de rede de 100 Mbits/s
X	-	Ocorrência de transmissão ou recepção na rede Ethernet, pelo ou para este endereço IP. Pisca sob demanda da UCP Nexto, e não a cada transmissão ou recepção, ou seja, pode piscar com uma frequência menor que a frequência real de transmissão ou recepção.

Tabela 217: Significado dos LEDs Ethernet

7.1.3. Diagnósticos via Página Web de Sistema

Além das características apresentadas anteriormente, a Série Nexto traz para o usuário uma ferramenta inovadora de acesso aos diagnósticos e estados de operação do sistema, através da aba *Informações do Barramento* da Página Web de Sistema.

A utilização, além de dinâmica, é bastante intuitiva e facilita as operações do usuário. Em outras palavras, pode substituir o uso de um sistema de supervisão quando o uso for restrito a verificação de status do sistema.

Para acessar a Página Web de Sistema da UCP desejada, basta utilizar um navegador padrão (Internet Explorer 7 ou superior, Mozilla Firefox 3.0 ou superior e Google Chrome 8 ou superior) e digitar, na barra de endereço, o endereço IP correspondente à UCP (Ex.: <http://192.168.1.1>). Inicialmente, serão apresentadas as informações da UCP, conforme mostra a Figura 185:

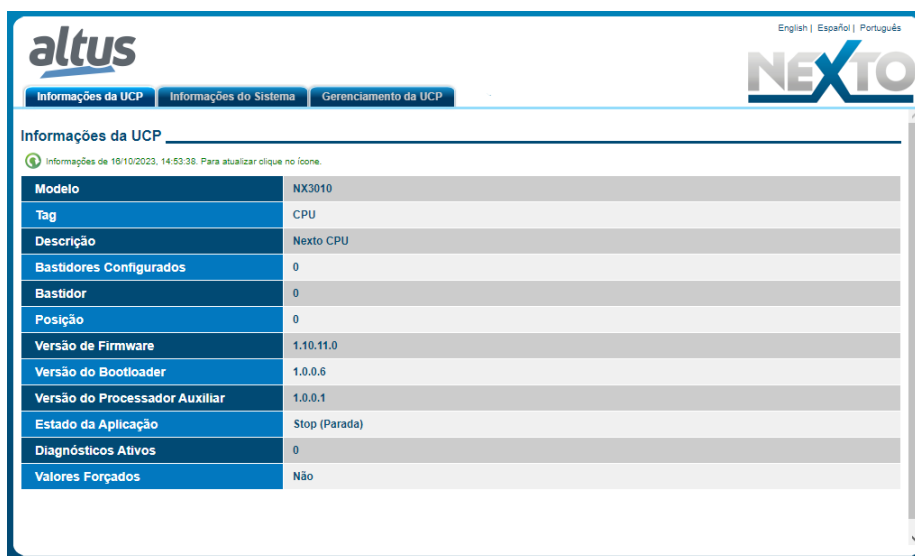


Figura 185: Tela Inicial

Também existe a aba “*Informações do Barramento*”, a qual pode ser visualizada através do Bastidor ou da lista dos módulos presentes (opção do lado direito da tela). Quando não houver nenhuma aplicação na UCP, será exibida nesta página uma configuração com o maior Bastidor disponível e uma fonte de alimentação padrão, juntamente com a UCP conectada. Quando a visualização pelo Bastidor é utilizada, os módulos que têm diagnóstico ficam piscando e assumem a cor vermelha, conforme mostra a Figura 186. Caso contrário será exibida uma lista com os módulos presentes no sistema, Tags correspondentes e número de diagnósticos ativos:

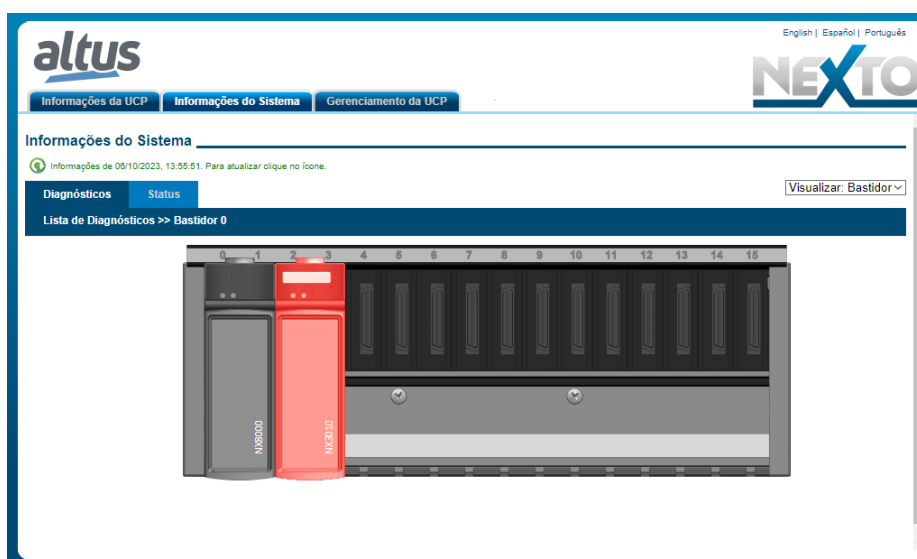


Figura 186: Informações do Barramento

Ao clicar no módulo com diagnóstico, no mesmo instante são mostrados os diagnósticos ativos do módulo, conforme mostra a Figura 187:

ATENÇÃO

Quando uma UCP for reiniciada e a aplicação entrar em exceção na partida do sistema, os diagnósticos não estarão válidos. É necessário corrigir o problema que gera a exceção da aplicação para que os diagnósticos sejam atualizados.

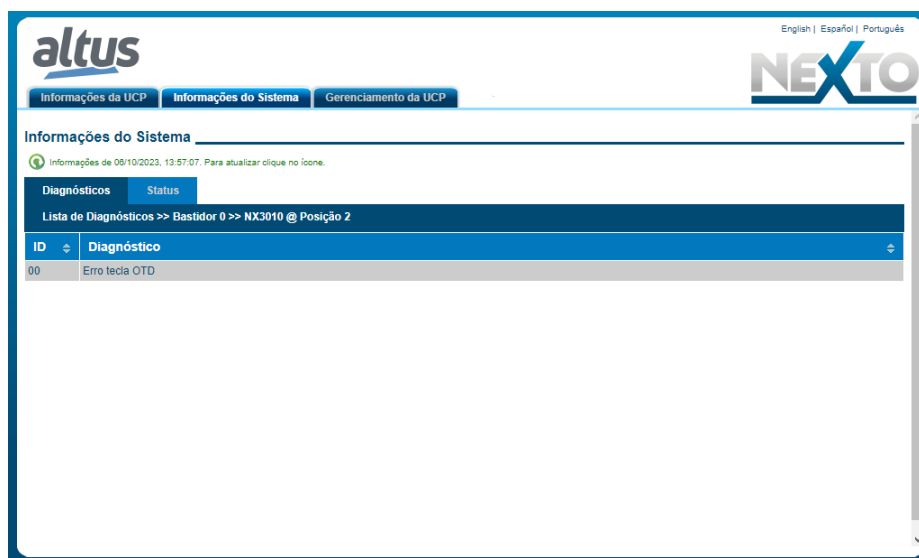


Figura 187: Diagnósticos do Sistema

Caso a seção Status seja selecionada, o estado de todos os diagnósticos detalhados é exibido na tela, conforme mostra a Figura 188:

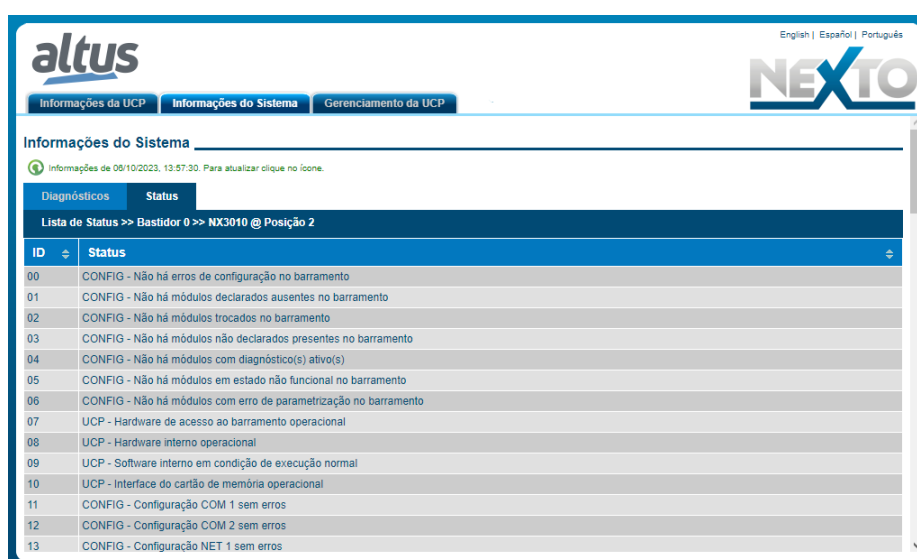


Figura 188: Estado do Sistema

O usuário pode optar por duas opções de idioma: Português e Inglês. Basta alterar na parte superior direita da tela para o idioma desejado.

7.1.4. Diagnósticos via Variáveis

As UCPs da Série Nexto possuem variáveis para indicação de diagnóstico. Existem estruturas de dados com os diagnósticos de todos os módulos declarados no barramento, mapeadas sobre variáveis de representação direta %Q, e definidas simbolicamente através da diretiva AT, na GVL System_Diagnostics criada automaticamente pelo MasterTool IEC XE.

A tabela abaixo faz um resumo da divisão dos bytes/words de diagnósticos:

Byte	Descrição
0 a 3	Diagnósticos resumidos da UCP.
4 a 693	Diagnósticos detalhados da UCP.

Tabela 218: Divisão dos Diagnósticos da UCP

7.1.4.1. Diagnósticos Resumidos

A tabela abaixo mostra o significado de cada bit dos diagnósticos resumidos da UCP:

Variável Direta		Mensagem de Diagnóstico	Variável DG_Modulo.tSummarized.*	Descrição
Variável	Bit			
-	-	SEM DIAG	-	Não existe diagnóstico ativo.
%QB(n)	0	CONFIG. INCOMPATIVEL	bConfigMismatch	TRUE – Existe algum problema de configuração no barramento, como módulo em posição incorreta.
				FALSE – O barramento está configurado corretamente.
	1	MODULOS AUSENTES	bAbsentModules	TRUE – Um ou mais módulos declarados estão ausentes.
				FALSE – Todos os módulos estão presentes no barramento.
	2	MODULOS TROCADOS	bSwappedModules	TRUE – Dois módulos estão trocados entre si no barramento.
				FALSE – Não há módulos trocados no barramento.
	3	MODULOS NAO DECLARADOS	bNonDeclaredModules	TRUE – Um ou mais módulos presentes no barramento não estão declarados.
				FALSE – Todos os módulos presentes no barramento estão declarados.
	4	MODULOS C/ DIAGNOSTICO	bModulesWithDiagnostic	TRUE – Um ou mais módulos do barramento estão com diagnóstico ativo.
				FALSE – Não existem diagnósticos ativos nos módulos do barramento.
	5	MODULOS C/ ERRO FATAL	bModuleFatalError	TRUE – Um ou mais módulos presentes no barramento estão em estado não funcional.
				FALSE – Todos os módulos presentes no barramento estão em estado funcional.
	6	MODULOS C/ ERRO PARAM.	bModuleParameterError	TRUE – Um ou mais módulos do barramento estão com erro de parametrização.
				FALSE – Todos os módulos estão parametrizados.
7	ERRO NO BARRAMENTO	bWHSBBusError	TRUE – Indicação do mestre que existe falha no barramento WHSB.	
			FALSE – O barramento WHSB está em correto funcionamento.	
0	FALHA DE HARDWARE	bHardwareFailure	TRUE – Falha no hardware da UCP.	
			FALSE – O hardware está em correto funcionamento.	

Variável Direta		Mensagem de Diagnóstico	Variável DG_Modulo.tSummarized.*	Descrição
Variável	Bit			
%QB(n+1)	1	EXCECAO NO SOFTWARE	bSoftwareException	TRUE – Uma ou mais exceções geradas pelo software.
				FALSE – Não foram geradas exceções no software.
	3	ERRO CARTAO DE MEMORIA	bMemoryCardError	TRUE – O cartão de memória está inserido na UCP, porém não está funcionando corretamente.
				FALSE – O cartão de memória está funcionando corretamente.
	4	ERRO CONFIG. COM 1	bCOM1ConfigError	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface serial COM 1.
				FALSE – A configuração da interface serial COM 1 está correta.
	5	ERRO CONFIG. COM 2	bCOM2ConfigError	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface serial COM 2.
				FALSE – A configuração da interface serial COM 2 está correta.
	6	ERRO CONFIG. NET 1	bNET1ConfigError	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface Ethernet NET 1.
				FALSE – A configuração da interface Ethernet NET 1 está correta.
	7	ERRO CONFIG. NET 2	bNET2ConfigError	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface Ethernet NET 2.
				FALSE – A configuração da interface Ethernet NET 2 está correta.
%QB(n+2)	0	DATA/HORA INVALIDA	bRTCInvalid	TRUE – A data ou a hora são inválidas.
				FALSE – A data e a hora estão corretas.
	1	RUNTIME RESET	bRTSReset	TRUE – O RTS (Runtime System) foi reiniciado pelo menos uma vez. Esse diagnóstico somente é limpo na reinicialização do sistema.
				FALSE – O RTS (Runtime System) está operando normalmente.
	2	ERRO TECLA OTD	bOTDSwitchError	TRUE – A tecla ficou travada por mais de 20 s pelo menos uma vez enquanto a UCP esteve energizada. Esse diagnóstico somente é limpo na reinicialização do sistema.
				FALSE – A tecla não está ou ficou travada enquanto a UCP esteve energizada.
0	BASTIDOR AUSENTE	bAbsentRacks	TRUE – Um ou mais bastidores declarados estão ausentes.	
			FALSE – Todos os bastidores declarados estão presentes.	
1	BASTIDOR DUPLICADO	bDuplicatedRacks	TRUE – Há algum bastidor com o número de identificação duplicado.	

Variável Direta		Mensagem de Diagnóstico	Variável DG_Modulo.tSummarized.*	Descrição
Variável	Bit			
%QB(n+3)	2	BASTIDOR INVALIDO	bInvalidRacks	FALSE – Não existem bastidores com o número de identificação duplicado.
				TRUE – Há algum bastidor com o número de identificação inválido.
	3	BASTIDOR NAO DECLARADO	bNonDeclaredRacks	FALSE – Não existem bastidores com o número de identificação inválido.
				TRUE – Há algum bastidor com o número de identificação não declarado.
	4	SLOT DUPLICADO	bDuplicatedSlots	FALSE – Não existem bastidores com o número de identificação não declarado.
				TRUE – Há algum endereço de slot duplicado.
			FALSE – Não existem endereços de slot duplicados.	

Tabela 219: Diagnósticos Resumidos da UCP

Notas:

Variável de representação direta: O “n” representa o valor configurado na UCP, através do software MasterTool IEC XE, como endereço inicial de diagnósticos.

Diretiva AT: Na descrição das variáveis simbólicas que utilizam a diretiva AT para fazer o mapeamento em variáveis de endereçamento direto, a sintaxe que deve ser colocada antes do diagnóstico resumido desejado é DG_Modulo.tSummarized., sendo a palavra Modulo substituída pela UCP utilizada. Por exemplo, para o diagnóstico de configuração incompatível basta utilizar a seguinte variável, DG_NX3010.tSummarized.bConfigMismatch. A diretiva AT é uma palavra reservada no software programador, sendo que algumas variáveis simbólicas que utilizam essa diretiva servem para indicar os diagnósticos.

Configuração incompatível: O diagnóstico de configuração incompatível é gerado caso um ou mais módulos do barramento não conferirem com o que foi declarado, ou seja, nas condições de módulos ausentes ou diferentes. Os módulos inseridos no barramento que não foram declarados no projeto não são considerados.

Módulos trocados: Se somente dois módulos estiverem trocados entre si no respectivo barramento, então o diagnóstico de troca pode ser identificado. Caso contrário, o problema será tratado como “Configuração Incompatível”.

Módulos com erro fatal: Caso o diagnóstico de Módulos com Erro Fatal seja verdadeiro, verificar qual é o módulo que está com problema no barramento e encaminhar para a Assistência Técnica da Altus, pois o mesmo está apresentando falha do hardware.

Módulos com erro de parametrização: Caso o diagnóstico de Módulo com Erro de Parametrização seja verdadeiro, verificar se os módulos do barramento estão configurados corretamente e se as versões de firmware e do software MasterTool IEC XE estão adequadas. Se o problema ocorrer ao inserir um módulo no barramento, verifique se esse módulo suporta troca à quente.

Erro no barramento: Considerado um erro fatal, interrompendo o acesso aos módulos do barramento. Caso o diagnóstico de erro no barramento seja verdadeiro, talvez haja um problema de hardware nas linhas de comunicação do barramento, sendo assim, deve-se entrar em contato com a Assistência Técnica da Altus.

Falha de hardware: Caso o diagnóstico de Falha de Hardware seja verdadeiro, encaminhar a UCP para Assistência Técnica da Altus, pois a mesma apresenta problemas no RTC, processador auxiliar, ou outros recursos de hardware.

Exceção no software: Caso o diagnóstico de exceção no software seja verdadeiro, o usuário deverá verificar a sua aplicação para garantir que a mesma não esteja acessando indevidamente a memória. Se o problema persistir, o setor de Suporte da Altus deverá ser consultado. Os códigos de exceção no software estão descritos após a tabela de diagnósticos detalhados da UCP.

Mensagem de Diagnóstico: As mensagens de diagnóstico podem ser visualizadas através do visor gráfico da UCP através da tecla OTD ou através da aba de diagnósticos da Página Web de Sistema da UCP.

7.1.4.2. Diagnósticos Detalhados

As tabelas a seguir mostram os diagnósticos detalhados das UCPs da série Nexto, para sua consulta é importante verificar as observações abaixo:

- **Visualização das Estruturas de Diagnóstico:** As Estruturas de Diagnóstico adicionadas ao projeto podem ser visualizadas acessando o item “*Library Manager*” na árvore de dispositivos da janela do MasterTool IEC XE. Com isso, é possível visualizar todos os tipos de dados definidos na estrutura.
- **Contadores:** Todos os contadores dos diagnósticos da UCP retornam à zero quando o seu valor limite é ultrapassado.
- **Variável de representação direta:** O “n” representa o valor configurado na UCP, através do software MasterTool IEC XE, como endereço inicial de diagnósticos.
- **Diretiva AT:** Na descrição das variáveis simbólicas que utilizam a diretiva AT para fazer o mapeamento em variáveis de endereçamento direto, a sintaxe que deve ser colocada antes do diagnóstico detalhado desejado é *DG_Modulo.tDetailed.*, sendo a palavra Modulo substituída pela UCP utilizada. A diretiva AT é uma palavra reservada no software programador, sendo que algumas variáveis simbólicas que utilizam essa diretiva servem para indicar os diagnósticos.

Representação Direta	Tamanho	Variável DG_Modulo.tDetailed.* ^{AT}	Descrição
%QD(n+4)	DWORD	Target. dwCPUModel	NX3003 = 0x3003 NX3004 = 0x3004 NX3005 = 0x3005 NX3010 = 0x3010 NX3020 = 0x3020 NX3030 = 0x3030 NX3035 = 0x3035
%QB(n+8)	BYTE ARRAY(4)	Target. abyCPUVersion	Versão do firmware.
%QB(n+12)	BYTE ARRAY(4)	Target. abyBootloaderVersion	Versão do bootloader.
%QB(n+16)	BYTE ARRAY(4)	Target. abyAuxprocVersion	Versão do processador auxiliar.

Tabela 220: Diagnósticos Detalhados Grupo Target

Representação Direta	Tamanho	Variável DG_Modulo.tDetailed.* ^{AT}	Descrição
%QX(n+20).0	BIT	Hardware. bAuxprocFailure	Falha na comunicação entre o processador auxiliar e o processador principal.
%QX(n+20).1	BIT	Hardware. bRTCFailure	O processador principal não está habilitado para comunicar com o RTC (relógio da UCP).
%QX(n+20).2	BIT	Hardware. bThermometerFailure	Falha na comunicação entre o termômetro e o processador principal.
%QX(n+20).3	BIT	Hardware. bLCDFailure	Falha na comunicação entre o visor LCD e o processador principal.

Tabela 221: Diagnósticos Detalhados Grupo Hardware

Representação Direta	Tamanho	Variável DG_Modulo.tDetailed.* ^{AT}	Descrição
%QW(n+21)	WORD	Exception. wExceptionCode	Código de Exceção gerado pelo RTS, o significado dos códigos pode ser conferido na Tabela 223.
%QB(n+23)	BYTE	Exception. byProcessorLoad	Nível, em percentagem (%), de carga no processador.

Tabela 222: Diagnósticos Detalhados Grupo Exception

Nota:

Código de exceção: O código de exceção gerado pelo RTS (Runtime System) pode ser consultado abaixo:

Código	Descrição	Código	Descrição
0x0000	Não há código de exceção.	0x0051	Violação de acesso.
0x0010	Tempo de Cão-de-Guarda da tarefa IEC expirado (Cão-de-Guarda de Software).	0x0052	Instrução privilegiada.
0x0012	Erro de configuração de E / S.	0x0053	Falha na página.
0x0013	Erros de check-up após o download do programa.	0x0054	Estouro de pilha.
0x0014	Erro de rede de campo.	0x0055	Disposição inválida.
0x0015	Erro de atualização de E / S.	0x0056	Manobra inválida.
0x0016	Tempo de ciclo (execução) excedido.	0x0057	Página protegida.
0x0017	Atualização online do programa por muito tempo.	0x0058	Falha dupla.
0x0018	Referências externas não resolvidas.	0x0059	OpCode inválido.
0x0019	Download rejeitado.	0x0100	Desalinhamento do tipo de dado.
0x001A	Projeto não-carregado, pois as variáveis retentivas não podem ser realocadas.	0x0101	Limite de arrays excedido.
0x001B	Projeto não-carregado e excluído.	0x0102	Divisão por zero.
0x001C	Fora da pilha de memória.	0x0103	Estouro.
0x001D	Memória retentiva corrompida; não pode ser mapeada.	0x0104	Não continuável.
0x001E	O projeto pode ser carregado, mas causa uma ruptura mais tarde.	0x0105	Cão-de-guarda na carga do processador de todas as tarefas IEC detectadas.
0x0021	O objetivo da aplicação de inicialização não corresponde ao destino atual.	0x0150	FPU: Erro não especificado.
0x0022	Erro de tarefas agendadas... Falha na configuração da tarefa IEC. Aplicação trabalhando com o alvo errado. Instrução ilegal.	0x0151	FPU: Operando não normal.
		0x0152	FPU: Divisão por zero.
0x0023	Erro de check-up do arquivo baixado.	0x0153	FPU: Resultado inexato.
0x0024	Incompatibilidade entre a identidade retentiva e a identidade atual do programa de projeto de inicialização	0x0154	FPU: Operação inválida.
0x0025	Falha na configuração da tarefa IEC.	0x0155	FPU: Estouro.
0x0026	A aplicação está sendo executada com o alvo errado.	0x0156	FPU: Verificação da pilha.
0x0050	Instrução ilegal.	0x0157	FPU: Underflow.

Tabela 223: Códigos de Exceção RTS

Representação Direta	Tamanho	Variável DG_Modulo.tDetailed.* ^{AT}	Descrição
%QB(n+25)	BYTE	RetainInfo. byCPUInitStatus	Status da Inicialização da UCP: 01: Hot start 02: Warm Start 03: Cold Start Obs.: Essas variáveis são reinicializadas em todas as energizações.
%QW(n+26)	WORD	RetainInfo. wCPUColdStartCounter	Incrementa caso a UCP perca retentividade durante a inicialização. (0 a 65535)
%QW(n+28)	WORD	RetainInfo. wCPUWarmStartCounter	Incrementa caso a UCP inicie com dados retentivos válidos. (0 a 65535)
%QW(n+30)	WORD	RetainInfo. wCPUHotStartCounter	Contador de distúrbios menores do que o tempo de suporte à falhas na alimentação da UCP. (0 a 65535).
%QW(n+32)	WORD	RetainInfo. wRTSResetCounter	Contador de resets efetuados pelo RTS (Runtime System). (0 a 65535).

Tabela 224: Diagnósticos Detalhados Grupo RetainInfo

Representação Direta	Tamanho	Variável DG_Modulo.tDetailed.* ^{AT}	Descrição
%QX(n+36).0	BIT	Reset. bBrownOutReset	A UCP foi reiniciada devido a uma falha na alimentação na última inicialização.
%QX(n+36).1	BIT	Reset. bWatchdogReset	A UCP foi reiniciada devido ao cão-de-guarda ativo na última inicialização.

Tabela 225: Diagnósticos Detalhados Grupo Reset

Nota:

Reset por Brownout: O diagnóstico de reset por brownout somente será verdadeiro quando a alimentação da fonte exceder o limite mínimo exigido nas características técnicas da mesma, mantendo-se com tensão baixa, ou seja, sem sofrer uma interrupção. A UCP irá identificar a queda da alimentação e indicará o diagnóstico de falha na alimentação. Quando a tensão for restabelecida, a UCP será reinicializada automaticamente e indicará o diagnóstico de reset por brownout.

Representação Direta	Tamanho	Variável DG_Modulo.tDetailed.* ^{AT}	Descrição
%QX(n+37).0	BIT	Thermometer. bOverTemperatureAlarm	Alarme gerado devido à temperatura interna estar em 85 °C ou acima de 85 °C.
%QX(n+37).1	BIT	Thermometer. bUnderTemperatureAlarm	Alarme gerado devido à temperatura interna estar em 0 °C ou abaixo de 0 °C.
%QD(n+38)	DINT	Thermometer. diTemperature	Temperatura lida no sensor interno da UCP.

Tabela 226: Diagnósticos Detalhados Grupo Thermometer

Nota:

Temperatura: Para a visualização da temperatura diretamente no endereço de memória, deve-se realizar uma conversão, pois o tamanho do dado é DINT e a monitoração é realizada em 4 bytes. Por isso, indica-se a utilização da variável simbólica associada, pois a mesma já fornece o valor final da temperatura.

Representação Direta	Tamanho	Variável DG_Modulo.tDetailed.* AT	Descrição
%QB(n+42)	BYTE	Serial.COM1. byProtocol	Protocolo selecionado na COM 1: 00: Sem protocolo 01: MODBUS RTU Master 02: MODBUS RTU Slave 03: Outro protocolo
%QD(n+43)	DWORD	Serial.COM1. dwRXBytes	Contador de caracteres recebidos através da COM 1 (0 a 4294967295).
%QD(n+47)	DWORD	Serial.COM1. dwTXBytes	Contador de caracteres transmitidos através da COM 1 (0 a 4294967295).
%QW(n+51)	WORD	Serial.COM1. wRXPendingBytes	Número de caracteres pendentes no buffer de leitura na COM 1 (0 a 1024).
%QW(n+53)	WORD	Serial.COM1. wTXPendingBytes	Número de caracteres pendentes no buffer de transmissão na COM 1 (0 a 1024).
%QW(n+55)	WORD	Serial.COM1. wBreakErrorCounter	O transmissor está mantendo a linha de dados em zero por muito tempo, de acordo com o bit de dados configurado.
%QW(n+57)	WORD	Serial.COM1. wParityErrorCounter	O frame recebido tem o bit de paridade incompatível.
%QW(n+59)	WORD	Serial.COM1. wFrameErrorCounter	O frame recebido tem o ponto de partida errado, geralmente causado por uma incompatibilidade de ruído ou baud rate.
%QW(n+61)	WORD	Serial.COM1. wRXOverrunCounter	Quando o buffer de anel recebido está cheio e começa a perder os frames antigos (muitos frames não tratados pelo dispositivo).

Tabela 227: Diagnósticos Detalhados Grupo Serial COM 1

Nota:

Contador de erro de paridade: Quando a paridade da interface serial COM 1 estiver configurada como Sem Paridade, este contador de erros não será incrementado ao receber uma mensagem com paridade diferente. Nesse caso, será indicado erro de frame.

Representação Direta	Tamanho	Variável DG_Modulo.tDetailed.* AT	Descrição
%QB(n+67)	BYTE	Serial.COM2. byProtocol	Protocolo selecionado na COM 2: 00: Sem protocolo 01: MODBUS RTU Master 02: MODBUS RTU Slave 03: Outro protocolo
%QD(n+68)	DWORD	Serial.COM2. dwRXBytes	Contador de caracteres recebidos através da COM 2 (0 a 4294967295).
%QD(n+72)	DWORD	Serial.COM2. dwTXBytes	Contador de caracteres transmitidos através da COM 2 (0 a 4294967295).
%QW(n+76)	WORD	Serial.COM2. wRXPendingBytes	Número de caracteres pendentes no buffer de leitura na COM 2 (0 a 1024).
%QW(n+78)	WORD	Serial.COM2. wTXPendingBytes	Número de caracteres pendentes no buffer de transmissão na COM 2 (0 a 1024).
%QW(n+80)	WORD	Serial.COM2. wBreakErrorCounter	O transmissor está mantendo a linha de dados em zero por muito tempo, de acordo com o bit de dados configurado.
%QW(n+82)	WORD	Serial.COM2. wParityErrorCounter	O frame recebido tem o bit de paridade incompatível.
%QW(n+84)	WORD	Serial.COM2. wFrameErrorCounter	O frame recebido tem o ponto de partida errado, geralmente causado por uma incompatibilidade de ruído ou baud rate.
%QW(n+86)	WORD	Serial.COM2. wRXOverrunCounter	Quando o buffer de anel recebido está cheio e começa a perder os frames antigos (muitos frames não tratados pelo dispositivo).

Tabela 228: Diagnósticos Detalhados Grupo Serial COM 2

Nota:

Contador de erro de paridade: Quando a paridade da interface serial COM 2 estiver configurada como Sem Paridade, este contador de erros não será incrementado ao receber uma mensagem com paridade diferente. Nesse caso, será indicado erro de frame.

Representação Direta	Tamanho	Variável AT DG_Modulo.tDetailed.*	Descrição
%QX(n+92).0	BIT	Ethernet.NET1. bLinkDown	Indica o estado do link na NET 1.
%QW(n+93)	WORD	Ethernet.NET1.wProtocol	Protocolo selecionado na NET 1: 00: Sem protocolo
%QX(n+93).0	BIT	Ethernet.NET1.wProtocol. bMODBUS_RTU_ETH_Client	Cliente MODBUS RTU via TCP.
%QX(n+93).1	BIT	Ethernet.NET1.wProtocol. bMODBUS_ETH_Client	Cliente MODBUS TCP.
%QX(n+93).2	BIT	Ethernet.NET1.wProtocol. bMODBUS_RTU_ETH_Server	Servidor MODBUS RTU via TCP.
%QX(n+93).3	BIT	Ethernet.NET1.wProtocol. bMODBUS_ETH_Server	Servidor MODBUS TCP.
%QB(n+95)	STRING (15)	Ethernet.NET1. szIP	Endereço IP NET 1.
%QB(n+111)	STRING (15)	Ethernet.NET1. szMask	Máscara de Subrede NET 1.
%QB(n+127)	STRING (15)	Ethernet.NET1. szGateway	Endereço Gateway NET 1.
%QB(n+143)	STRING (17)	Ethernet.NET1. szMAC	Endereço MAC NET 1.
%QB(n+161)	BYTE ARRAY(4)	Ethernet.NET1. abyIP	Endereço IP NET 1.
%QB(n+165)	BYTE ARRAY(4)	Ethernet.NET1. abyMask	Máscara de Subrede NET 1.
%QB(n+169)	BYTE ARRAY(4)	Ethernet.NET1. abyGateway	Endereço Gateway NET 1.
%QB(n+173)	BYTE ARRAY(6)	Ethernet.NET1. abyMAC	Endereço MAC NET 1.
%QD(n+179)	DWORD	Ethernet.NET1. dwPacketsSent	Contador de pacotes enviados através da porta NET 1 (0 a 4294967295).
%QD(n+183)	DWORD	Ethernet.NET1. dwPacketsReceived	Contador de pacotes recebidos através da porta NET 1 (0 a 4294967295).
%QD(n+187)	DWORD	Ethernet.NET1. dwBytesSent	Contador de bytes enviados através da porta NET 1 (0 a 4294967295).
%QD(n+191)	DWORD	Ethernet.NET1. dwBytesReceived	Contador de bytes recebidos através da porta NET 1 (0 a 4294967295).
%QW(n+195)	WORD	Ethernet.NET1. wTXErrors	Contador de erros de transmissão através da porta NET 1 (0 a 65535).
%QW(n+197)	WORD	Ethernet.NET1. wTXFIFOErrors	Contador de erros no buffer de transmissão atra- vés da porta NET 1 (0 a 65535).
%QW(n+199)	WORD	Ethernet.NET1. wTXDropErrors	Contador de perdas de conexão na transmissão através da porta NET 1 (0 a 65535).
%QW(n+201)	WORD	Ethernet.NET1. wTXCollisionErrors	Contador de erros de colisão na transmissão através da porta NET 1 (0 a 65535).
%QW(n+203)	WORD	Ethernet.NET1. wTXCarrierErrors	Contador de erros de transporte na transmissão através da porta NET 1 (0 a 65535).
%QW(n+205)	WORD	Ethernet.NET1. wRXErrors	Contador de erros de recepção através da porta NET 1 (0 a 65535).
%QW(n+207)	WORD	Ethernet.NET1. wRXFIFOErrors	Contador de erros no buffer de recepção através da porta NET 1 (0 a 65535).
%QW(n+209)	WORD	Ethernet.NET1. wRXDropErrors	Contador de perdas de conexão na recepção através da porta NET 1 (0 a 65535).

Representação Direta	Tamanho	Variável AT DG_Modulo.tDetailed.*	Descrição
%QW(n+211)	WORD	Ethernet.NET1. wRXFrameErrors	Contador de erros de frame na recepção através da porta NET 1 (0 a 65535).
%QW(n+213)	WORD	Ethernet.NET1. wMulticast	Contador de pacotes multicast através da porta NET 1 (0 a 65535).

Tabela 229: Diagnósticos Detalhados Grupo Ethernet NET1

Representação Direta	Tamanho	Variável AT DG_Modulo.tDetailed.*	Descrição
%QX(n+219).0	BIT	Ethernet.NET2. bLinkDown	Indica o estado do link na NET 2.
%QW(n+220)	WORD	Ethernet.NET2.wProtocol	Protocolo selecionado na NET 2: 00: Sem protocolo
%QX(n+220).0	BIT	Ethernet.NET2.wProtocol. bMODBUS_RTU_ETH_Client	Cliente MODBUS RTU via TCP.
%QX(n+220).1	BIT	Ethernet.NET2.wProtocol. bMODBUS_ETH_Client	Cliente MODBUS TCP.
%QX(n+220).2	BIT	Ethernet.NET2.wProtocol. bMODBUS_RTU_ETH_Server	Servidor MODBUS RTU via TCP.
%QX(n+220).3	BIT	Ethernet.NET2.wProtocol. bMODBUS_ETH_Server	Servidor MODBUS TCP.
%QB(n+222)	STRING (15)	Ethernet.NET2. szIP	Endereço IP NET 2.
%QB(n+238)	STRING (15)	Ethernet.NET2. szMask	Máscara de Subrede NET 2.
%QB(n+254)	STRING (15)	Ethernet.NET2. szGateway	Endereço Gateway NET 2.
%QB(n+270)	STRING (17)	Ethernet.NET2. szMAC	Endereço MAC NET 2.
%QB(n+288)	BYTE ARRAY(4)	Ethernet.NET2. abyIP	Endereço IP NET 2.
%QB(n+292)	BYTE ARRAY(4)	Ethernet.NET2. abyMask	Máscara de Subrede NET 2.
%QB(n+296)	BYTE ARRAY(4)	Ethernet.NET2. abyGateway	Endereço Gateway NET 2.
%QB(n+300)	BYTE ARRAY(6)	Ethernet.NET2. abyMAC	Endereço MAC NET 2.
%QD(n+306)	DWORD	Ethernet.NET2. dwPacketsSent	Contador de pacotes enviados através da porta NET 2 (0 a 4294967295).
%QD(n+310)	DWORD	Ethernet.NET2. dwPacketsReceived	Contador de pacotes recebidos através da porta NET 2 (0 a 4294967295).
%QD(n+314)	DWORD	Ethernet.NET2. dwBytesSent	Contador de bytes enviados através da porta NET 2 (0 a 4294967295).
%QD(n+318)	DWORD	Ethernet.NET2. dwBytesReceived	Contador de bytes recebidos através da porta NET 2 (0 a 4294967295).
%QW(n+322)	WORD	Ethernet.NET2. wTXErrors	Contador de erros de transmissão através da porta NET 2 (0 a 65535).
%QW(n+324)	WORD	Ethernet.NET2. wTXFIFOErrors	Contador de erros no buffer de transmissão através da porta NET 2 (0 a 65535).
%QW(n+326)	WORD	Ethernet.NET2. wTXDropErrors	Contador de perdas de conexão na transmissão através da porta NET 2 (0 a 65535).
%QW(n+328)	WORD	Ethernet.NET2. wTXCollisionErrors	Contador de erros de colisão na transmissão através da porta NET 2 (0 a 65535).
%QW(n+330)	WORD	Ethernet.NET2. wTXCarrierErrors	Contador de erros de transporte na transmissão através da porta NET 2 (0 a 65535).
%QW(n+332)	WORD	Ethernet.NET2. wRXErrors	Contador de erros de recepção através da porta NET 2 (0 a 65535).

Representação Direta	Tamanho	Variável AT DG_Modulo.tDetailed.*	Descrição
%QW(n+334)	WORD	Ethernet.NET2. wRXFIFOErrors	Contador de erros no buffer de recepção através da porta NET 2 (0 a 65535).
%QW(n+336)	WORD	Ethernet.NET2. wRXDropErrors	Contador de perdas de conexão na recepção através da porta NET 2 (0 a 65535).
%QW(n+338)	WORD	Ethernet.NET2. wRXFrameErrors	Contador de erros de frame na recepção através da porta NET 2 (0 a 65535).
%QW(n+340)	WORD	Ethernet.NET2. wMulticast	Contador de pacotes multicast através da porta NET 2 (0 a 65535).

Tabela 230: Diagnósticos Detalhados Grupo Ethernet NET2

Representação Direta	Tamanho	Variável AT DG_Modulo.tDetailed.*	Descrição
%QB(n+346)	BYTE	UserFiles. byMounted	Indica se a memória utilizada para gravar arquivos de usuário está apta para receber os dados.
%QD(n+347)	DWORD	UserFiles. dwFreeSpacekB	Espaço livre da memória de arquivos de usuário em Kbytes.
%QD(n+351)	DWORD	UserFiles. dwTotalSizekB	Capacidade de armazenamento da memória de arquivos de usuário em Kbytes.

Tabela 231: Diagnósticos Detalhados Grupo UserFiles

Nota:

Partição de Usuário: A partição de usuário é uma área da memória reservada para o armazenamento de dados na UCP. Por exemplo: arquivos com extensão PDF, arquivos com extensão DOC e demais dados.

Representação Direta	Tamanho	Variável AT DG_Modulo.tDetailed.*	Descrição
%QB(n+356)	BYTE	UserLogs. byMounted	Status da memória em que são inseridos os logs de usuário.
%QW(n+357)	WORD	UserLogs. wFreeSpacekB	Espaço livre na memória de logs de usuário em Kbytes.
%QW(n+359)	WORD	UserLogs. wTotalSizekB	Capacidade de armazenamento da memória de logs de usuário em Kbytes.

Tabela 232: Diagnósticos Detalhados Grupo UserLogs

Representação Direta	Tamanho	Variável AT DG_Modulo.tDetailed.*	Descrição
%QB(n+362)	BYTE	MemoryCard. byMounted	Status do Cartão de Memória: 00: Cartão de memória não montado 01: Cartão de memória inserido e montado
%QX(n+363).0	BIT	MemoryCard. bMemcardtoCPUEnabled	Nível de Proteção do Cartão de Memória: Leitura de dados do cartão de memória pela UCP autorizada.
%QX(n+363).1	BIT	MemoryCard. bCPUtoMemcardEnabled	Escrita de dados no cartão de memória pela UCP autorizada.
%QD(n+364)	DWORD	MemoryCard. dwFreeSpacekB	Espaço livre no Cartão de Memória em Kbytes.
%QD(n+368)	DWORD	MemoryCard. dwTotalSizekB	Capacidade de armazenamento do Cartão de Memória em Kbytes.

Tabela 233: Diagnósticos Detalhados Grupo MemoryCard

Representação Direta	Tamanho	Variável DG_Modulo.tDetailed.* AT	Descrição
%QB(n+372)	BYTE	WHSB. byHotSwapAndStartupStatus	Informa a situação anormal no barramento que ocasionou a parada da aplicação, para cada Modo de troca a quente. Consulte a Tabela 235 para detalhes das possibilidades.
%QB(n+373)	BYTE	WHSB. byReserved_0	Reservado.
%QD(n+374)	DWORD ARRAY (32)	WHSB. adwRackIOErrorStatus	Identificação de erros em módulos de E/S, individualmente. Para mais informações sobre este diagnóstico consultar a nota abaixo.
%QD(n+502)	DWORD ARRAY (32)	WHSB. adwModulePresenceStatus	Status de presença de módulos declarados de E/S em barramentos, individualmente. Para mais informações sobre este diagnóstico consultar a nota abaixo.
%QB(n+630)	BYTE	WHSB. byWHSBBusErrors	Contador de falhas no barramento WHSB. Esse contador é reinicializado na energização (0 a 255).

Tabela 234: Diagnósticos Detalhados Grupo WHSB

Notas:

Diagnóstico de erro dos módulos do barramento: Cada DWORD do Array deste diagnóstico representa um bastidor, cujas posições são representadas pelos bits destas DWORDs. Logo, o Bit-0 da DWORD-0 equivale a posição zero do bastidor com endereço zero. Cada um desses Bits é o resultado de uma operação lógica OU entre os diagnósticos de configuração incompatível (bConfigMismatch), módulos ausentes (bAbsentModules), módulos trocados (bSwappedModules), módulos com erro fatal (bModuleFatalError) e o estado operacional do módulo de uma determinada posição.

Status de presença de módulos: Cada DWORD do Array deste diagnóstico representa um bastidor, cujas posições são representadas pelos bits destas DWORDs. Logo, o Bit-0 da DWORD-0 equivale a posição zero do bastidor com endereço zero. Então, se o módulo estiver presente, este bit será verdadeiro. É importante ressaltar que esse diagnóstico é válido para todos os módulos, exceto fontes de alimentação e UCPs, ou seja, não apresentam a presença no barramento em suas respectivas posições (bit permanece em falso).

Situações que ocasionam parada da aplicação: Os códigos das possíveis situações que ocasionam parada da aplicação podem ser consultados abaixo:

Código	Enumerável	Descrição
00	INITIALIZING	Esse estado é apresentado enquanto os demais estados não estiverem prontos.
01	RESET_WATCHDOG	Aplicação em Modo Stop devido ao reset por cão-de-guarda de hardware ou por uma reinicialização do Runtime, quando desabilitada a configuração "Iniciar Aplicação de Usuário após Reset por Cão-de-Guarda".
02	ABSENT_MODULES_HOT_SWAP_DISABLED	Aplicação em Modo Stop devido à ativação do diagnóstico Módulos Ausentes, quando configurada troca a quente "Desabilitada" ou troca a quente "Desabilitada, apenas para módulos declarados".
03	CFG_MISMATCH_HOT_SWAP_DISABLED	Aplicação em Modo Stop devido à ativação do diagnóstico Configuração Incompatível, quando configurada troca a quente "Desabilitada" ou troca a quente "Desabilitada, apenas para módulos declarados".
04	ABSENT_MODULES_HOT_SWAP_STARTUP_CONSISTENCY	Aplicação em Modo Stop devido à ativação do diagnóstico Módulos Ausentes, quando configurada troca a quente "Habilitada, com consistência na partida" ou troca a quente "Habilitada, com consistência na partida somente para os módulos declarados".

Código	Enumerável	Descrição
05	CFG_MISMATCH_HOT_SWAP_STARTUP_CONSISTENCY	Aplicação em Modo Stop devido à ativação do diagnóstico Configuração Incompatível, quando configurada troca a quente <i>"Habilitada, com consistência na partida"</i> ou troca a quente <i>"Habilitada, com consistência na partida somente para os módulos declarados"</i> .
06	APPL_STOP_ALLOWED_TO_RUN	Aplicação em Modo Stop e todas as consistências realizadas com sucesso. Aplicação pode ser colocada no Modo Run.
07	APPL_STOP_MODULES_NOT_READY	Aplicação em Modo Stop e todas as consistências realizadas com sucesso, mas os módulos de E/S não estão aptos para a partida do sistema. Não é possível colocar a Aplicação em Modo Run.
08	APPL_STOP_MODULES_GETTING_READY_TO_RUN	Aplicação em Modo Stop e todas as consistências realizadas com sucesso. Os módulos de E/S estão sendo preparados para a partida do sistema. Não é possível colocar a Aplicação em Modo Run.
09	NORMAL_OPERATING_STATE	Aplicação em Modo Run.
10	MODULE_CONSISTENCY_OK	Uso interno.
11	APPL_STOP_DUE_TO_EXCEPTION	Aplicação em modo Stop pois uma exceção ocorreu na UCP.
12	DUPLICATED_SLOT_HOT_SWAP_DISABLED	Aplicação em Modo Stop devido à ativação do diagnóstico Slots Duplicados, quando configurada troca a quente <i>"Desabilitada"</i> ou troca a quente <i>"Desabilitada, apenas para módulos declarados"</i> .
13	DUPLICATED_SLOT_HOT_SWAP_STARTUP_CONSISTENCY	Aplicação em Modo Stop devido à ativação do diagnóstico Slots Duplicados, quando configurada troca a quente <i>"Habilitada, com consistência na partida"</i> ou troca a quente <i>"Habilitada, com consistência na partida somente para os módulos declarados"</i> .
14	DUPLICATED_SLOT_HOT_SWAP_ENABLED	Aplicação em Modo Stop devido à ativação do diagnóstico Slots Duplicados, quando configurada troca a quente <i>"Habilitada, sem consistência na partida"</i> .
15	NON_DECLARED_MODULE_HOT_SWAP_STARTUP_CONSISTENCY	Aplicação em Modo Stop devido à ativação do diagnóstico Módulos Não Declarados, quando configurada troca a quente <i>"Habilitada, com consistência na partida"</i> .
16	NON_DECLARED_MODULE_HOT_SWAP_DISABLED	Aplicação em Modo Stop devido à ativação do diagnóstico Módulos Não Declarados, quando configurada troca a quente <i>"Desabilitada"</i> .

Tabela 235: Códigos de Situações que Ocasionalmente Param a Aplicação

Representação Direta	Tamanho	Variável DG_Modulo.tDetailed.* ^{AT}	Descrição
%QB(n+631)	BYTE	Application. byCPUState	Informa o estado de operação da UCP: 01: Aplicação em execução (Modo Run) 03: Aplicação parada (Modo Stop)
%QX(n+632).0	BIT	Application. bForcedIOs	Existem um ou mais pontos de E/S forçados.
%QX(n+632).1	BIT	Application. bNetDefinedByWeb	A UCP NX30x0 não possui suporte a troca de IP via Página Web de Sistema. Então o diagnóstico permanecerá sempre em FALSE

Tabela 236: Diagnósticos Detalhados Grupo Application

Representação Direta	Tamanho	Variável DG_Modulo.tDetailed.* ^{AT}	Descrição
%QX(n+633).0	BIT	SNTP. bServiceEnabled	Serviço SNTP habilitado.
%QB(n+634)	BYTE	SNTP. byActiveTimeServer	Indica qual servidor está ativo: 00: Nenhum servidor ativo. 01: Servidor primário ativo. 02: Servidor secundário ativo.
%QW(n+635)	WORD	SNTP. wPrimaryServerDownCount	Contador de vezes que o servidor primário esteve indisponível (0 a 65535).
%QW(n+637)	WORD	SNTP. wSecondaryServerDownCount	Contador de vezes que o servidor secundário esteve indisponível (0 a 65535).
%QD(n+639)	DWORD	SNTP. dwRTCTimeUpdatedCount	Contador de vezes que o RTC foi atualizado pelo serviço SNTP (0 a 4294967295).
%QB(n+643)	BYTE	SNTP. byLastUpdateSuccessful	Indica status da última atualização: 00: Não foi atualizado. 01: Última atualização falhou. 02: Última atualização teve sucesso.
%QB(n+644)	BYTE	SNTP. byLastUpdateTimeServer	Indica qual servidor foi utilizado na última atualização: 00: Nenhuma atualização. 01: Servidor primário. 02: Servidor secundário.
%QB(n+645)	BYTE	SNTP.sLastUpdateTime. byDayOfMonth	Dia da última atualização do RTC.
%QB(n+646)	BYTE	SNTP.sLastUpdateTime. byMonth	Mês da última atualização do RTC.
%QW(n+647)	WORD	SNTP.sLastUpdateTime. wYear	Ano da última atualização do RTC.
%QB(n+649)	BYTE	SNTP.sLastUpdateTime. byHours	Hora da última atualização do RTC.
%QB(n+650)	BYTE	SNTP.sLastUpdateTime. byMinutes	Minuto da última atualização do RTC.
%QB(n+651)	BYTE	SNTP.sLastUpdateTime. bReservedAlign	Reservado para alinhamento.
%QB(n+652)	BYTE	SNTP.sLastUpdateTime. bySeconds	Segundo da última atualização do RTC.
%QW(n+653)	WORD	SNTP.sLastUpdateTime. wMilliseconds	Milissegundo da última atualização do RTC.

Tabela 237: Diagnósticos Detalhados Grupo SNTP

Representação Direta	Tamanho	Variável DG_Modulo.tDetailed.* ^{AT}	Descrição
%QX(n+659).0	BIT	SOE[1]. bConnectionStatus	Status de conexão do cliente 01
%QX(n+659).1	BIT	SOE[1]. bOverflowStatus	Status da fila de eventos do cliente 01: FALSE - Não houve estouro TRUE - Limite da fila excedido
%QB(n+660)	BYTE	SOE[1]. byReserved_0	Reservado
%QW(n+661)	WORD	SOE[1]. wEventsCounter	Contador de eventos na fila do cliente 01
%QX(n+663).0	BIT	SOE[2]. bConnectionStatus	Status de conexão do cliente 02
%QX(n+663).1	BIT	SOE[2]. bOverflowStatus	Status da fila de eventos do cliente 02: FALSE - Não houve estouro TRUE - Limite da fila excedido
%QB(n+664)	BYTE	SOE[2]. byReserved_0	Reservado.
%QW(n+665)	WORD	SOE[2]. wEventsCounter	Contador de eventos na fila do cliente 02.

Tabela 238: Diagnósticos Detalhados Grupo SOE

Notas:

Sincronismo dos diagnósticos do grupo SOE em sistema operando com redundância de Half-Cluster: Quando um projeto é configurado com redundância de Half-Cluster os diagnósticos do grupo SOE não são sincronizados entre os dois Half-Clusters.

Atualização dos diagnósticos do grupo SOE na transição para o estado ativo: Quando um Half-Cluster passa do estado Reserva para o estado Ativo os diagnósticos do grupo SOE passam a ser atualizados a partir do terceiro ciclo.

Representação Direta	Tamanho	Variável DG_Modulo.tDetailed.* ^{AT}	Descrição
%QD(n+667)	DWORD	Rack. dwAbsentRacks	Cada bit representa um número de identificação de um bastidor, se algum bit for TRUE, isso significa que o bastidor, com o referido número de identificação, está ausente.
%QD(n+671)	DWORD	Rack. dwDuplicatedRacks	Cada bit representa um número de identificação de um bastidor, se algum bit for TRUE, isso significa que mais de um bastidor está configurado com o mesmo número de identificação.
%QD(n+675)	DWORD	Rack. dwNonDeclaredRacks	Cada bit representa um número de identificação de um bastidor, se algum bit for TRUE, significa que há algum bastidor configurado com um número de identificação que não está declarado no projeto.

Tabela 239: Diagnósticos Detalhados Grupo Rack

Representação Direta	Tamanho	Variável DG_Modulo.tDetailed.* ^{AT}	Descrição
%QD(n+681)	DWORD	ApplicationInfo. dwApplicationCRC	CRC de 32 bits da aplicação. Quando a aplicação é modificada e enviada para a UCP, um novo CRC é calculado.

Tabela 240: Diagnósticos Detalhados Grupo ApplicationInfo

7.1.5. Diagnósticos via Blocos Funcionais

Os blocos funcionais proporcionam a visualização de alguns parâmetros que não podem ser acessados de outra maneira. A função sobre diagnósticos avançados está localizada na biblioteca *NextoStandard* e está descrita abaixo.

7.1.5.1. GetTaskInfo

Essa função retorna informações sobre uma tarefa de uma determinada aplicação.



Figura 189: Função GetTaskInfo

Abaixo, são descritos os parâmetros que devem ser repassados à função para que ela retorne as informações da aplicação.

Parâmetros de entrada	Tipo	Descrição
psAppName	POINTER TO STRING	Nome da aplicação.
psTaskName	POINTER TO STRING	Nome da tarefa.
pstTaskInfo	POINTER TO stTaskInfo	Ponteiro para receber informações da tarefa.

Tabela 241: Parâmetros de Entrada GetTaskInfo

Os dados que a função retorna, através do ponteiro informado nos parâmetros de entrada, são os descritos na tabela abaixo.

Parâmetros de saída	Tipo	Descrição
dwCurScanTime	DWORD	Tempo de ciclo (execução) da tarefa com 1 μ s de resolução.
dwMinScanTime	DWORD	Tempo mínimo de ciclo da tarefa com 1 μ s de resolução.
dwMaxScanTime	DWORD	Tempo máximo de ciclo da tarefa com 1 μ s de resolução.
dwAvgScanTime	DWORD	Tempo médio de ciclo da tarefa com 1 μ s de resolução.
dwLimitMaxScan	DWORD	Tempo máximo de ciclo da tarefa antes de ocorrer o cão-de-guarda.
dwIECCycleCount	DWORD	Contador de ciclos IEC decorridos.

Tabela 242: Parâmetros de Saída GetTaskInfo

Possíveis ERRORCODE:

- NoError: execução com sucesso;
- TaskNotPresent: a tarefa desejada não existe.

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
sAppName : STRING;
psAppName : POINTER TO STRING;
sTaskName : STRING;
psTaskName : POINTER TO STRING;
pstTaskInfo : POINTER TO stTaskInfo;
TaskInfo : stTaskInfo;
```

```

Info : ERRORCODE;
END_VAR
//ENTRADAS:
sAppName := 'Application'; //Variável recebe o nome da aplicação.
psAppName := ADR(sAppName); //Ponteiro com o nome da aplicação.
sTaskName := 'MainTask'; //Variável recebe o nome da tarefa.
psTaskName := ADR(sTaskName); //Ponteiro com o nome da tarefa.
pstTaskInfo := ADR(TaskInfo); //Ponteiro que irá receber as informações da
tarefa.
//FUNÇÃO:
//Chamada da função.
Info := GetTaskInfo (psAppName, psTaskName, pstTaskInfo);
//Variável Info recebe possíveis erros da função.

```

7.2. Visor Gráfico

O visor gráfico disponível neste produto da Série Nexto é uma importante ferramenta para o controle de processo, pois através dele podem ser reconhecidas as possíveis condições de erro, presença de componentes ou de diagnósticos ativos. Além disso, é através do visor gráfico que todos os diagnósticos, inclusive dos módulos de E/S, são exibidos ao usuário. Para maiores detalhes sobre a utilização da tecla de diagnósticos e da visualização dos mesmos, consultar a seção [One Touch Diag](#).

Na figura abaixo, é possível visualizar todos os caracteres disponíveis no visor gráfico deste produto e, a seguir, os seus respectivos significados.

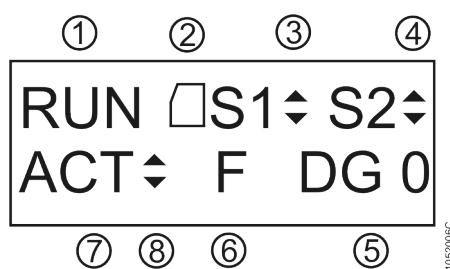


Figura 190: Tela de Status da UCP

Legenda:

- 1. Indicação do estado de operação da UCP. Caso a aplicação da UCP esteja em execução, o estado será RUN. Caso a aplicação da UCP esteja parada, o estado será STOP e quando estiver parada em marcas de depuração da aplicação o estado será BRKP. Para maiores detalhes, consultar a seção [Estados de Operação da UCP](#).
- 2. Indicação da presença do Cartão de Memória. Para maiores detalhes sobre a instalação do cartão, consultar a seção [Instalação do Cartão de Memória](#).
- 3. Indicação de Tráfego na COM 1. A seta para cima (▲) indica transmissão de dados e a seta para baixo (▼) indica recepção de dados. Para maiores informações sobre a interface COM 1, consultar seção [Interfaces Seriais](#).
- 4. Indicação de Tráfego na COM 2. A seta para cima (▲) indica transmissão de dados e a seta para baixo (▼) indica recepção de dados. Para maiores informações sobre a interface COM 2, consultar seção [Interfaces Seriais](#).
- 5. Indicação da quantidade de diagnósticos ativos na UCP. Caso o número mostrado seja diferente de 0 (zero), existem diagnósticos ativos na UCP. Para maiores detalhes sobre a visualização dos mesmos no visor gráfico da UCP, através da tecla de diagnósticos, consultar seção [One Touch Diag](#).
- 6. Indicação de variáveis forçadas na UCP. Caso o caractere F esteja exibido no visor gráfico, alguma variável está sendo forçada pelo usuário, seja ela variável simbólica ou variável de representação direta mapeada em uma variável simbólica. Para maiores detalhes sobre forçamento de variáveis, consultar seção [Escrita e Forçamento de Variáveis](#).
- 7. Identificação do estado da redundância na UCP (mensagem válida somente na NX3030 em modo redundante). Caso a UCP seja o CP Ativo, a informação ACT será apresentada. Os demais estados possíveis são NCF (Não Configurado), STR (Inicializando), INA (Inativo) e SBY (Reserva).

- 8. Indicação de que está sendo executada a sincronização de projeto. A seta para cima (▲) indica transmissão de dados do projeto e a seta para baixo (▼) indica recepção de dados de um projeto.

Além dos caracteres descritos acima, as UCPs Nexto poderão apresentar algumas mensagens no visor gráfico, correspondentes a algum processo que está sendo executado no momento.

A tabela abaixo mostra as mensagens e suas respectivas descrições:

Mensagem	Descrição
FORMATANDO...	Indica que a UCP está formatando o cartão de memória.
ERRO NA FORMATAÇÃO	Indica que ocorreu um erro durante a formatação do cartão de memória pela UCP.
FORMATO INCORR.	Indica que o formato do cartão de memória está incorreto.
SENHA INCORRETA	Indica que a senha digitada não confere com a senha configurada.
TRANSFERINDO...	Indica que o projeto está sendo transferido.
ERRO NA TRANSFER.	Indica que ocorreu um erro na transferência do projeto, ocasionado por algum problema no cartão de memória ou a remoção do mesmo durante a transferência.
TRANSFER. COMPLETA	Indica que a transferência foi completada com êxito.
TIMEOUT TRANSFER.	Indica que ocorreu um time-out (expirou tempo de comunicação) durante a transferência do projeto.
MODELO UCP INVALIDO	Indica que o modelo da UCP é diferente do configurado no projeto que está no cartão de memória.
VERSAO UCP INVALIDA	Indica que a versão da UCP é diferente do configurado no projeto que está no cartão de memória.
APLICACAO CORROMPIDA	Indica que a aplicação que está no cartão de memória está corrompida.
APLICACAO INEXISTENTE	Indica que não existe aplicação dentro do cartão de memória para ser transferida à UCP.
CRC INEXISTENTE	Indica que o CRC da aplicação não existe.
MCF INEXISTENTE	Indica que não existe o arquivo MCF dentro do cartão de memória.
SEM TAG	Não existe Tag configurada para a UCP no MasterTool IEC XE.
SEM DESC	Não existe Descrição configurada para a UCP no MasterTool IEC XE.
ERRO MSG.	Indica que há erro(s) na(s) mensagem(s) de diagnóstico(s) do(s) módulo(s) requisitado(s).
SEM ASSINATURA	Indica que o produto apresentou um problema inesperado. Entre em contato com o setor de Suporte da Altus.
ERRO NA APL. REINICIANDO	Indica que ocorreu um erro na aplicação e o Runtime está reiniciando a aplicação.
APL. NAO CARREGADA	Indica que o Runtime não irá carregar a aplicação.
CARREGANDO APL.	Indica que o Runtime irá carregar a aplicação.
POSICAO INCORRETA	Indica que a UCP está em uma posição incorreta no bastidor.
ERRO FATAL	Indica que há problemas graves na inicialização da UCP como por exemplo, as partições da UCP não foram devidamente montadas. Entre em contato com o setor de suporte da Altus.
HW-SW INCOMPATIVEL	Indica que o software e hardware da UCP não são compatíveis pois o produto apresentou algum problema inesperado. Entre em contato com o setor de suporte da Altus.
ATUALIZANDO FIRMWARE	Indica que o firmware está sendo atualizado na UCP.



Mensagem	Descrição
RECEBENDO FIRMWARE	Indica que o arquivo de atualização está sendo transferido para a UCP.
ATUALIZADO	Exibe a versão de firmware atualizada na UCP.
ERRO NA ATUALIZACAO	Indica que houve algum erro durante a atualização de firmware da UCP, ocasionado por falha na comunicação ou problemas de configuração.
REINICIANDO SISTEMA...	Indica que a UCP está sendo reiniciada para que as atualizações tenham efeito.

Tabela 243: Outras Mensagens do Visor Gráfico

7.3. Log de Sistema

O *Log de Sistema* é um recurso disponível no programador MasterTool IEC XE. É uma importante ferramenta de registro de processo, pois através dele é possível localizar eventos na UCP que podem indicar condições de erro, presença de componentes ou de diagnósticos ativos. Tais eventos podem ser visualizados em ordem cronológica com uma resolução de segundos, com uma capacidade de armazenamento de até mil entradas de log armazenadas na memória interna do CP e não podem ser removidos.

Para acessar esses Logs, basta ir na *Árvore de Dispositivos* e dar um duplo clique em *Device*, e em seguida ir na aba *Log*, onde podem ser visualizadas as centenas de operações como: ciclo máximo das tarefas, acesso de usuários, alteração online, download e upload da aplicação, forçamento das variáveis, sincronização da aplicação em CPs redundantes, atualização de firmware entre outros eventos e ações.

Para que os *Logs* possam ser visualizados, basta estar conectado a uma UCP (Caminho Ativo selecionado) e clicar em . Quando este botão é pressionado os Logs serão exibidos e a atualização deles é feita automaticamente. Quando o botão não está pressionado os Logs serão mantidos na tela, mas não serão atualizados. Ou seja, este botão tem dois estados, sendo que um estado mantém os logs sendo atualizados e no outro estado a atualização está desabilitada. Para deixar de exibir os Logs, basta clicar em .

É possível filtrar os Logs em 4 tipos diferentes: Advertência(s), Erro(s), Exceção (s), Informação (s).

Outra maneira de filtrar as mensagens exibidas para o usuário é selecionar qual componente queremos visualizar.

A *Marca de Tempo* da aba de Log é apresentada pela ferramenta MasterTool a partir de informações fornecidas pelo dispositivo (UCP). A ferramenta MasterTool pode apresentar a Marca de Tempo no horário local (do computador) ou no horário UTC, se marcada a opção *Tempo UTC*.

ATENÇÃO

Se o horário ou o parâmetro de fuso horário do dispositivo estiverem incorretos, a Marca de Tempo apresentada pela ferramenta MasterTool não corresponderá ao tempo esperado.

Para maiores informações sobre os Logs de Sistema, favor consultar o Manual de Utilização do MasterTool IEC XE - MU299048 e os subcapítulos Relógio RTC e Sincronização de Tempo deste manual.

ATENÇÃO

Os logs de sistema das UCPs, a partir da versão de firmware 1.4.0.33 (Nexto) e 1.14.36.0 (Xtorm), são recarregados no caso de uma reinicialização da UCP ou por uma reinicialização do Runtime System, isto é, será possível visualizar os logs mais antigos quando uma dessas condições ocorrer.

7.4. Não Carregar a Aplicação na Inicialização

Caso seja necessário, o usuário pode optar por não carregar uma aplicação já existente na UCP durante sua inicialização. Para isto, basta energizar a UCP com a tecla de diagnóstico pressionada e mantê-la assim até ser exibida a mensagem na tela "APL. NÃO CARREGADA". Após a inicialização da UCP e a exibição no visor da mensagem avisando que a aplicação não será carregada a UCP estará também em Modo Stop. Caso seja feito um login, o software MasterTool IEC XE indicará que

não existe nenhuma aplicação na UCP. Para voltar a carregar a aplicação, a UCP deve ser reinicializada, ou um novo download da aplicação deve ser feito.

7.5. Falha na Alimentação

A fonte de alimentação da Série Nexto (NX8000) possui um sistema de detecção de falha, de acordo com os níveis definidos em suas características técnicas (consultar Características Técnicas do Módulo de Alimentação 30W 24Vdc – CT114200). Existem duas formas de diagnosticar a falha:

1. Caso a fonte NX8000 seja energizada com tensão inferior ao limite mínimo exigido, será gerado um diagnóstico de falha de alimentação, o qual será reconhecido pela UCP e a mensagem “FALHA NA ALIMENTAÇÃO” será exibida em seu visor. Quando a alimentação estiver dentro dos limites estabelecidos, a UCP irá reconhecer e, automaticamente, será reiniciada com a aplicação do usuário. O diagnóstico ainda estará ativo para mostrar ao usuário que na última inicialização ocorreu falha na alimentação.
2. Caso a fonte NX8000 tenha uma queda de tensão para um valor inferior ao limite mínimo exigido e retornar para um valor acima em até 10 ms, a falha de alimentação não será reconhecida pela UCP e não será gerado o diagnóstico, pois o sistema se mantém intacto durante esse tempo. Porém, se a queda de tensão for superior a 10 ms, a mensagem “FALHA NA ALIMENTAÇÃO” será exibida no visor da UCP e o diagnóstico será ativado.



Figura 191: Mensagem de Falha na Alimentação

O usuário pode alterar o valor da variável atribuída à falha na alimentação para FALSE durante a execução da aplicação, facilitando a verificação e tratamento deste diagnóstico.

O diagnóstico de FALHA NA ALIMENTAÇÃO já está previamente mapeado em uma região específica de memória, definida como Diagnósticos Detalhados da UCP. Dessa forma, basta utilizá-lo como uma variável global. O nome da variável está descrita na lista de diagnósticos detalhados da UCP na seção [Diagnósticos via Variáveis](#).

7.6. Problemas mais Comuns

Se, ao energizar a UCP, a mesma não entra em funcionamento, os seguintes itens devem ser verificados:

- A temperatura ambiente está dentro da faixa suportada pelos equipamentos?
- A fonte de alimentação do bastidor está sendo alimentada com a tensão correta?
- A fonte de alimentação é o módulo mais à esquerda no bastidor (bastidor sendo visto de frente), e é seguido pela UCP da Série Nexto à sua direita?
- Os equipamentos da rede, como hubs, switches ou roteadores, estão alimentados, interligados, configurados e funcionando corretamente?
- O cabo de rede Ethernet está devidamente conectado à porta NET da UCP Nexto e ao equipamento de rede?
- A UCP da Série Nexto está ligada, em modo de execução (Run) e sem diagnósticos relacionados à hardware?

Se a UCP Nexto indica o estado execução (Run), mas não responde às comunicações solicitadas, sejam elas pelo Master-Tool IEC XE ou através de protocolos, os seguintes itens devem ser verificados:

- A configuração dos parâmetros Ethernet da UCP estão corretos?
- O respectivo protocolo de comunicação está configurado corretamente na UCP?
- As variáveis que habilitam as relações MODBUS estão devidamente habilitadas?

Se nenhum problema for identificado, consulte o Suporte à Clientes Altus.

7.7. Solução de Problemas

A tabela abaixo mostra os sintomas de alguns problemas com suas possíveis causas e soluções. Se o problema persistir, entre em contato com o Suporte Técnico da Altus.

Sintoma	Possível Causa	Solução
Não liga	Falta de alimentação ou alimentado incorretamente.	Verificar se a UCP está conectada corretamente no bastidor. Desenergizar e retirar todos os módulos do barramento, menos a fonte de alimentação e a UCP. Energizar o barramento e verificar o funcionamento da fonte de alimentação, tanto a externa como a fonte conectada ao barramento. Verificar se a tensão de alimentação chega ao borne da fonte de alimentação Nexto, e também se a polarização está correta.
Visor Gráfico da UCP apresenta a mensagem WRONG SLOT	UCP na Posição Incorreta.	A UCP deve ocupar os slots 2 e 3 do bastidor 0. Colocá-la na posição correta.
Não comunica	Mau contato ou mal configurado.	Verificar todas as conexões dos cabos de comunicação. Verificar as configurações das interfaces seriais e Ethernet no software MasterTool IEC XE.
Não reconhece o cartão de memória	Mal conectado ou desmontado.	Verificar se o cartão de memória está conectado corretamente no compartimento. Verificar se o cartão de memória foi colocado do lado correto, conforme a indicação no painel frontal da UCP. Verificar se o cartão de memória não foi desmontado através da tecla MS, localizada no painel frontal, visualizando a indicação no visor gráfico da UCP.

Tabela 244: Solução de Problemas

7.8. Manutenção Preventiva

- Deve-se verificar, a cada ano, se os cabos de interligação estão com as conexões firmes, sem depósitos de poeira, principalmente os dispositivos de proteção.
- Em ambientes sujeitos a contaminação excessiva, deve-se limpar periodicamente o equipamento, retirando resíduos, poeira, etc.
- Os diodos TVS utilizados para a proteção contra transientes causados por descargas atmosféricas devem ser inspecionados periodicamente, pois podem estar danificados ou destruídos caso a energia absorvida esteja acima do limite. Em muitos casos, a falha pode não ser evidente ou facilmente visível. Em aplicações críticas, é recomendável a substituição periódica dos diodos TVS, mesmo os que não apresentarem sinais visíveis de falha.
- Aperto e limpeza do barramento a cada 6 meses.
- Para maiores informações, consultar o Manual de Utilização Série Nexto – MU214000.

8. Anexo - Interoperabilidade DNP3

8.1. DNP3 Device Profile

DNP3 DEVICE PROFILE DOCUMENT	
Device Identification	
Vendor Name	Altus S/A
Device Name	NX3030
Device Function	Slave
DNP Levels Supported for	Requests: None Responses: None
Connections Supported	IP Networking
Methods to set Configurable Parameters	Software: Master Tool IEC XE
IP Networking	
Type of End Point:	TCP Listening (Outstation Only)
Accepts TCP Connections from	Allows all
IP Address(es) from which TCP Connections are accepted:	*.*.*.*
TCP Listen Port Number	Configurable, range 1 to 65535
TCP Keep-alive timer	Configurable, range 0 to 4294967295
Multiple master connections	Supports up to two masters Based on TCP port number
Time synchronization support	SNTP
Link Layer	
Data Link Address	Configurable, range 0 to 65519
Self Address Support using address 0xFFFC	No
Requires Data Link Layer Confirmation	Never
Maximum number of octets Transmitted in a Data Link Frame	Fixed at 292
Maximum number of octets that can be Received in a Data Link Frame	Fixed at 292
Application Layer	
Maximum number of octets Transmitted in an Application Layer Fragment	Fixed at 2048
Maximum number of octets that can be received in an Application Layer Fragment	Fixed at 2048
Timeout waiting for Application Confirm of solicited response message	Fixed at 10000 ms
Device Trouble Bit IIN1.6	This bit will be set if PLC is not in RUN mode
Event Buffer Overflow Behavior	Discard the oldest event
Sends Multi-Fragment Responses	Yes
Outstation Unsolicited Response Support	
Supports Unsolicited Reporting	No

Tabela 245: DNP3 Device Profile

8.2. DNP V3.0 Implementation Table

DNP OBJECT GROUP & VARIATION			REQUEST Master may issue Outstation must parse		RESPONSE Master must parse Outstation may issue	
Group Num	Var Num	Description	Function Codes (dec)	Qualifier Codes (hex)	Function Codes (dec)	Qualifier Codes (hex)
1	0	Binary Input – Any Variation	1 (read)	00, 01 (start-stop) 06 (no range, or all)		
1	1	Binary Input – Packed format	1 (read)	00, 01 (start-stop) 06 (no range, or all)	129 (response)	00, 01 (start-stop)
2	0	Binary Input Event – Any Variation	1 (read)	06 (no range, or all) 07, 08 (limited qty)		
2	2	Binary Input Event – With absolute time	1 (read)	06 (no range, or all) 07, 08 (limited qty)	129 (response)	17, 28 (index)
60	1	Class Objects – Class 0 data	1 (read)	06 (no range, or all)		
60	2	Class Objects – Class 1 data	1 (read)	06 (no range, or all) 07, 08 (limited qty)		
80	1	Internal Indications - Packed format	1 (read)	00, 01 (start-stop)	129 (response)	00 (start-stop)
			2 (write)	00 (start-stop) index=7		

Tabela 246: DNP V3.0 Implementation Table