



Manual de Utilização UCP Série Nexto NX3008

MU214109 Rev. O

30 de julho de 2026

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	2
1.2.	Características Inovadoras	2
1.3.	Documentos Relacionados a este Manual	4
1.4.	Inspeção Visual	5
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	7
2.2.1.	Características Gerais	7
2.2.2.	Normas e Certificações	9
2.2.3.	Memória	10
2.2.4.	Protocolos	12
2.2.5.	Interface Serial	13
2.2.5.1.	COM 1	13
2.2.6.	Interface CAN	13
2.2.7.	Interface USB	14
2.2.8.	Interfaces Ethernet	15
2.2.8.1.	NET 1	15
2.2.8.2.	NET 2	16
2.2.8.3.	NET 3	16
2.2.9.	Interface do Cartão de Memória	17
2.2.10.	Alimentação	17
2.2.11.	Características Ambientais	18
2.3.	Compatibilidade com Outros Produtos	18
2.4.	Desempenho	18
2.4.1.	Tempo de Intervalo da MainTask	19
2.4.2.	Tempos de Aplicação	19
2.4.3.	Tempos para Execução de Instruções	19
2.4.4.	Tempos de Inicialização	19
2.5.	Dimensões Físicas	20
2.6.	Dados para Compra	20
2.6.1.	Itens Integrantes	20
2.6.2.	Código do Produto	21
2.7.	Produtos Relacionados	21
3.	Instalação	22
3.1.	Instalação Mecânica	22

3.2.	Instalação Elétrica	22
3.2.1.	Utilizando a fonte de alimentação integrada	22
3.2.2.	Utilizando o módulo fonte de alimentação NX8000	23
3.3.	Conexão com a Rede Ethernet	25
3.3.1.	Endereço IP	25
3.3.2.	ARP Gratuito	26
3.3.3.	Instalação do Cabo de Rede	26
3.4.	Conexão de Rede Serial RS-485 e CAN	27
3.5.	Instalação do Cartão de Memória	27
3.6.	Instalação da Arquitetura	29
3.6.1.	Instalação de Módulos no Bastidor Principal	29
3.7.	Instalação do Programador	29
4.	Programação Inicial	30
4.1.	Organização e Acesso à Memória	30
4.2.	Perfis de Projeto	32
4.2.1.	Simples - MT8500	32
4.2.2.	Básico - MT8500	32
4.2.3.	Normal - MT8500	33
4.2.4.	Experiente - MT8500	33
4.2.5.	Personalizado - MT8500	34
4.2.6.	Perfil de Máquina - MT8500	34
4.2.7.	Perfis do MT9000	35
4.2.7.1.	Perfil Fast (MT9000)	35
4.2.7.2.	Perfil Process (MT9000)	35
4.2.8.	Tabela Geral - MT8500	35
4.2.9.	Número Máximo de Tarefas	36
4.3.	Configurando a UCP	36
4.4.	Bibliotecas	37
4.5.	Inserindo uma Instância de Protocolo	38
4.5.1.	MODBUS RTU	38
4.5.2.	MODBUS Ethernet	39
4.6.	Localizando o Dispositivo	40
4.7.	Login	42
4.8.	Modo Run	44
4.9.	Modo Stop	45
4.10.	Escrita e Forçamento de Variáveis	46
4.11.	Logout	46
4.12.	Upload do Projeto	47
4.13.	Estados de Operação da UCP	48
4.13.1.	Run	48
4.13.2.	Stop	48
4.13.3.	Breakpoint	48
4.13.4.	Exception	49
4.13.5.	Reset a Quente	49
4.13.6.	Reset a Frio	49
4.13.7.	Reset Origem	49
4.13.8.	Reset Process Command (IEC 60870-5-104)	49
4.14.	Programas (POUs) e Listas de Variáveis Globais (GVLs)	49

4.14.1.	Programa MainPrg	49
4.14.2.	Programa StartPrg	50
4.14.3.	Programa UserPrg	50
4.14.4.	GVL System_Diagnostics	50
4.14.5.	GVL Disables	51
4.14.6.	GVL IOQualities	52
4.14.7.	GVL Module_Diagnostics	53
4.14.8.	GVL Qualities	54
4.14.9.	GVL ReqDiagnostics	55
5.	Configuração	57
5.1.	Device	57
5.1.1.	Gerenciamento de Usuários e Direitos de Acesso	57
5.1.2.	Configurações do CP	57
5.2.	Configuração da UCP	59
5.2.1.	Parâmetros Gerais	59
5.2.1.1.	Troca a Quente	60
5.2.1.1.1.	Troca a Quente Desabilitada, Apenas para Módulos Declarados	61
5.2.1.1.2.	Troca a Quente Desabilitada	61
5.2.1.1.3.	Troca a Quente Desabilitada, sem Consistência na Partida	61
5.2.1.1.4.	Troca a Quente Habilitada, com Consistência na Partida Apenas para Módulos Declarados	61
5.2.1.1.5.	Troca a Quente Habilitada, com Consistência na Partida	62
5.2.1.1.6.	Troca a Quente Habilitada, sem Consistência na Partida	62
5.2.1.1.7.	Como realizar a Troca a Quente	62
5.2.1.2.	Áreas de Memória Retentiva e Persistente	64
5.2.1.3.	Parâmetros da Porta CAN	65
5.2.2.	Configuração de Evento Externo	65
5.2.3.	Sincronização de Tempo	67
5.2.3.1.	IEC 60870-5-104	68
5.2.3.2.	SNTP	68
5.2.3.3.	Horário de Verão	69
5.2.4.	Pontos Internos	69
5.2.4.1.	Conversões de Qualidade	71
5.2.4.1.1.	Qualidade Interna	72
5.2.4.1.2.	Conversão IEC 60870-5-104	73
5.2.4.1.3.	Qualidade Interna MODBUS	74
5.2.4.1.4.	Qualidade dos Módulos de E/S do Barramento Local	75
5.2.4.1.5.	Qualidade dos Módulos de E/S do Barramento PROFIBUS	76
5.2.4.1.6.	Qualidade da Entrada Digital PROFIBUS	76
5.2.4.1.7.	Qualidade da Saída Digital PROFIBUS	77
5.2.4.1.8.	Qualidade da Entrada Analógica PROFIBUS	78
5.2.4.1.9.	Qualidade da Saída Analógica PROFIBUS	80
5.3.	Configuração da Interface Serial	81
5.3.1.	COM 1	81
5.3.1.1.	Configurações Avançadas	83
5.4.	Configuração da Interface CAN (Controller Area Network)	83
5.4.1.	CAN	83
5.5.	Configuração das Interfaces Ethernet	83

5.5.1.	Interfaces Ethernet Integradas	84
5.5.1.1.	NET 1	84
5.5.1.2.	NET 2	84
5.5.1.3.	NET 3	84
5.5.1.4.	Modos de Operação das Interfaces Ethernet Integradas	85
5.5.1.4.1.	Modo Simples	85
5.5.1.4.2.	Modo NIC Teaming	85
5.5.1.4.3.	Modo Switch	86
5.5.2.	Interface Ethernet Remota NX5000	89
5.5.2.1.	NET 1	89
5.5.2.2.	Modos de Operação da Interface Ethernet Remota NX5000	90
5.5.2.2.1.	Modo Redundante	90
5.5.3.	Portas TCP/UDP Reservadas	90
5.5.4.	Roteamento de Rede	91
5.6.	Configuração de Protocolos	92
5.6.1.	Comportamento dos Protocolos x Estados da UCP	94
5.6.2.	Pontos Duplos	95
5.6.3.	Fila de Eventos da UCP	95
5.6.3.1.	Consumidores	96
5.6.3.2.	Princípios de Funcionamento da Fila	96
5.6.3.2.1.	Sinalização de Estouro	96
5.6.3.3.	Produtores	97
5.6.4.	Interceptação de Comandos Oriundos do Centro de Controle	97
5.6.5.	MODBUS RTU Mestre	102
5.6.5.1.	Configuração do Protocolo MODBUS Mestre por Mapeamento Simbólico	102
5.6.5.1.1.	Parâmetros Gerais do Protocolo MODBUS Mestre – Configuração por Mapeamento Simbólico	102
5.6.5.1.2.	Configuração dos Dispositivos – Configuração por Mapeamento Simbólico	104
5.6.5.1.3.	Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico	105
5.6.5.1.4.	Configuração das Requisições – Configuração por Mapeamento Simbólico	106
5.6.5.2.	Configuração do Protocolo MODBUS Mestre por Representação Direta (%Q)	109
5.6.5.2.1.	Parâmetros Gerais do Protocolo MODBUS Mestre – Configuração por Representação Direta (%Q)	109
5.6.5.2.2.	Configuração dos Dispositivos – Configuração por Representação Direta (%Q)	111
5.6.5.2.3.	Configuração dos Mapeamentos – Configuração por Representação Direta (%Q)	112
5.6.6.	MODBUS RTU Escravo	114
5.6.6.1.	Configuração do Protocolo MODBUS Escravo por Mapeamento Simbólico	114
5.6.6.1.1.	Parâmetros Gerais do Protocolo MODBUS Escravo – Configuração por Mapeamento Simbólico	114
5.6.6.1.2.	Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico	117
5.6.6.2.	Configuração do Protocolo MODBUS Escravo por Representação Direta (%Q)	118
5.6.6.2.1.	Parâmetros Gerais do Protocolo MODBUS Escravo – Configuração por Representação Direta (%Q)	118
5.6.6.2.2.	Configuração dos Mapeamentos – Configuração por Representação Direta (%Q)	119
5.6.7.	MODBUS Ethernet	121
5.6.8.	MODBUS Ethernet Cliente	123
5.6.8.1.	Configuração do Protocolo MODBUS Ethernet Cliente por Mapeamento Simbólico	123

5.6.8.1.1.	Parâmetros Gerais do Protocolo MODBUS Cliente – Configuração por Mapeamento Simbólico	124
5.6.8.1.2.	Configuração dos Dispositivos – Configuração por Mapeamento Simbólico	125
5.6.8.1.3.	Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico	126
5.6.8.1.4.	Configuração das Requisições – Configuração por Mapeamento Simbólico	127
5.6.8.2.	Configuração do Protocolo MODBUS Ethernet Cliente por Representação Direta (%Q)	130
5.6.8.2.1.	Parâmetros Gerais do Protocolo MODBUS Cliente – Configuração por Representação Direta (%Q)	130
5.6.8.2.2.	Configuração dos Dispositivos – configuração por Representação Direta (%Q)	131
5.6.8.2.3.	Configuração dos Mapeamentos – Configuração por Representação Direta (%Q)	132
5.6.8.3.	Disparo de Relações MODBUS Cliente de Forma Acíclica	135
5.6.9.	MODBUS Ethernet Servidor	135
5.6.9.1.	Configuração do Protocolo MODBUS Ethernet Servidor por Mapeamento Simbólico	135
5.6.9.1.1.	Parâmetros Gerais do Protocolo MODBUS Servidor – Configuração por Mapeamento Simbólico	135
5.6.9.1.2.	Diagnósticos MODBUS Servidor – Configuração por Mapeamento Simbólico	137
5.6.9.1.3.	Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico	138
5.6.9.2.	Configuração do Protocolo MODBUS Ethernet Servidor por Representação Direta (%Q)	139
5.6.9.2.1.	Parâmetros Gerais do Protocolo MODBUS Servidor – Configuração por Representação Direta (%Q)	139
5.6.9.2.2.	Configuração dos Mapeamentos – Configuração por Representação Direta (%Q)	140
5.6.10.	OPC DA Servidor	142
5.6.10.1.	Criando um Projeto para Comunicação OPC DA	144
5.6.10.2.	Configurando um CP no Servidor OPC DA	147
5.6.10.2.1.	Importando uma Configuração do Projeto	149
5.6.10.3.	Variáveis de Status e Qualidade da Comunicação OPC DA	149
5.6.10.4.	Limites da Comunicação com Servidor OPC DA	151
5.6.10.5.	Acessando Dados Através de um Cliente OPC DA	151
5.6.11.	OPC UA Servidor	153
5.6.12.	Mestre EtherCAT	154
5.6.12.1.	Instalando e Inserindo Dispositivos EtherCAT	155
5.6.12.1.1.	EtherCAT - Scan Devices	155
5.6.12.2.	Configuração do Mestre EtherCAT	156
5.6.12.2.1.	EtherCAT Master - General	156
5.6.12.2.2.	EtherCAT Master - Sync Unit Assignment	159
5.6.12.2.3.	EtherCAT Master - Overview	159
5.6.12.2.4.	EtherCAT Master - I/O Mapping	159
5.6.12.2.5.	EtherCAT Master - IEC Objects	159
5.6.12.2.6.	EtherCAT Master - Status / Informations	159
5.6.12.3.	Configuração do Escravo EtherCAT	160
5.6.12.3.1.	EtherCAT Slave - General	160
5.6.12.3.2.	EtherCAT Slave - Process Data	162
5.6.12.3.3.	EtherCAT Slave - Edit PDO List	164
5.6.12.3.4.	EtherCAT Slave - Startup Parameters	165
5.6.12.3.5.	EtherCAT Slave - Module I/O Mapping	165

5.6.12.3.6.	EtherCAT Slave - Status and Information	165
5.6.13.	EtherNet/IP	165
5.6.13.1.	Adaptador Ethernet	166
5.6.13.2.	Configuração do Scanner EtherNet/IP	168
5.6.13.2.1.	Geral	168
5.6.13.2.2.	Conexões	169
5.6.13.2.3.	Assemblies	171
5.6.13.2.4.	Parâmetros Definidos pelo Usuário	172
5.6.13.2.5.	EtherNet/IP: Mapeamento de E/S	174
5.6.13.3.	Configuração do Adapter EtherNet/IP	174
5.6.13.3.1.	Geral	174
5.6.13.3.2.	EtherNet/IP Adapter: Mapeamento de E/S	175
5.6.13.4.	Configuração do Módulo EtherNet/IP	175
5.6.13.4.1.	Assemblies	175
5.6.13.4.2.	EtherNet/IP Module: Mapeamento de E/S	176
5.6.14.	IEC 60870-5-104 Servidor	176
5.6.14.1.	Tipos de Dados	176
5.6.14.2.	Pontos Duplos	178
5.6.14.2.1.	Pontos Duplos de Entrada Digital	178
5.6.14.2.2.	Pontos Duplos de Saída Digital	180
5.6.14.3.	Parâmetros Gerais	184
5.6.14.4.	Mapeamento dos Dados	185
5.6.14.5.	Camada de Enlace	187
5.6.14.6.	Camada de Aplicação	189
5.6.14.7.	Diagnósticos do Servidor	191
5.6.14.8.	Qualificador dos Comandos	192
5.6.15.	CANOpen Manager	193
5.6.15.1.	Instalando e inserindo dispositivos CANopen	194
5.6.15.2.	Configuração do CANOpen Manager	195
5.6.15.3.	Configuração de Escravo CANopen	196
5.6.16.	PROFINET Controller	197
5.7.	Desempenho de Comunicação	197
5.7.1.	Servidor MODBUS	197
5.7.1.1.	Interfaces Integradas da UCP	197
5.7.1.2.	Interfaces Remotas	198
5.7.2.	Servidor OPC UA	198
5.7.3.	Servidor IEC 60870-5-104	199
5.8.	Desempenho do Sistema	199
5.8.1.	Escaneamento de E/S	199
5.8.2.	Cartão de Memória	200
5.9.	Relógio RTC	200
5.9.1.	Blocos Funcionais e Funções para Leitura e Escrita do RTC	201
5.9.1.1.	Funções de Leitura do RTC	201
5.9.1.1.1.	GetDateAndTime	201
5.9.1.1.2.	GetTimeZone	202
5.9.1.1.3.	GetDayOfWeek	202
5.9.1.2.	Funções de Escrita do RTC	203
5.9.1.2.1.	SetDateAndTime	203

5.9.1.2.2.	SetTimeZone	204
5.9.2.	Estruturas de Dados do RTC	205
5.9.2.1.	EXTENDED_DATE_AND_TIME	206
5.9.2.2.	DAYS_OF_WEEK	206
5.9.2.3.	RTC_STATUS	207
5.9.2.4.	TIMEZONESETTINGS	207
5.9.3.	Limites de Operação do RTC	207
5.10.	Memória de Arquivos de Usuário	208
5.11.	Menu Informativo e de Configuração da UCP	210
5.12.	Blocos Funcionais e Funções	211
5.12.1.	Blocos Funcionais Especiais para Comunicação Serial	211
5.12.1.1.	SERIAL_CFG	214
5.12.1.2.	SERIAL_GET_CFG	216
5.12.1.3.	SERIAL_GET_RX_QUEUE_STATUS	217
5.12.1.4.	SERIAL_PURGE_RX_QUEUE	218
5.12.1.5.	SERIAL_RX	219
5.12.1.6.	SERIAL_RX_EXTENDED	221
5.12.1.7.	SERIAL_SET_CTRL	223
5.12.1.8.	SERIAL_TX	225
5.12.2.	Atualização de Entradas e Saídas	227
5.12.2.1.	REFRESH_INPUT	227
5.12.2.2.	REFRESH_OUTPUT	228
5.12.3.	Bloco Funcional PID	229
5.12.4.	Temporizador Retentivo	229
5.12.4.1.	TOF_RET	230
5.12.4.2.	TON_RET	231
5.12.4.3.	TP_RET	232
5.12.5.	Log de Usuário	234
5.12.5.1.	UserLogAdd	235
5.12.5.2.	UserLogDeleteAll	236
5.12.6.	ClearRtuDiagnostic	237
5.12.7.	ClearEventQueue	237
5.13.	Páginas Web de Usuário	238
5.14.	Página Web de Sistema	238
5.14.1.	Informações Gerais	238
5.14.2.	Informações do Barramento	238
5.14.3.	Gerenciamento	240
5.14.3.1.	Sistema	240
5.14.3.1.1.	Atualização de Firmware	240
5.14.3.1.2.	Ajuste do Relógio	241
5.14.3.1.3.	Controle do Sistema	242
5.14.3.2.	Rede	242
5.14.3.2.1.	Configuração	243
5.14.3.2.2.	Sniffer	244
5.14.3.3.	SNMP	245
5.14.3.3.1.	SNMP nas UCPs Nexto	246
5.14.3.3.2.	Configuração SNMP	247
5.14.3.3.3.	Usuário e Comunidades SNMP	247

5.14.3.4.	Dispositivos USB	248
5.14.3.4.1.	Dispositivos de Armazenamento em Massa	249
5.14.3.4.2.	Conversor USB para RS-232	251
5.14.3.4.3.	Dispositivos Modem	252
5.14.3.4.4.	Adaptadores Wi-Fi	254
5.14.3.4.5.	Função de Usuário para Controle da Interface USB	257
5.14.3.5.	Firewall	257
5.14.3.5.1.	Configurações Gerais	258
5.14.3.5.2.	Regras de Usuário	259
5.14.3.5.3.	Modo Switch ou NIC Teaming	261
5.14.3.6.	OpenVPN	261
5.14.3.6.1.	Configuração do OpenVPN	262
5.14.3.6.2.	Arquivos de Segurança	266
5.14.3.6.3.	Status	267
5.14.3.6.4.	Arquivos para Baixar	268
5.14.3.6.5.	Configuração de Arquiteturas	269
5.14.3.7.	Servidor FTP	270
5.14.3.7.1.	Configurações Gerais	271
5.14.3.7.2.	Configurações de Usuário	272
5.14.3.7.3.	Status	272
5.14.3.8.	Cartão de Memória	272
5.14.3.8.1.	Configuração	273
5.15.	Docker	278
5.15.1.	Como acessar um Container Docker	278
5.15.1.1.	Como Acessar o Portainer	278
5.15.2.	Quick Setup	279
5.15.2.1.	Informações Do Ambiente Local	280
5.15.3.	Dashboard de Ambiente	280
5.15.3.1.	Menu De Containers	281
5.15.3.2.	Menu De Imagens	281
5.15.4.	Criando e configurando um Container	282
5.15.4.1.	Configurando Um Container	283
5.15.4.1.1.	Menu Command & Logging	283
5.15.4.1.2.	Menu Network	283
5.15.4.1.3.	Menu Runtime & Resources	284
5.15.4.1.4.	Menu Capabilities	285
5.15.4.2.	Utilizando Um Container	285
5.15.4.3.	Como criar uma imagem a partir de um container existente salvando os seus dados	287
5.15.4.4.	Monitorando Um Container	287
5.15.4.5.	Características Técnicas	288
5.15.4.5.1.	Memória Flash	288
5.15.4.5.2.	Memória RAM	288
5.15.4.5.3.	UCP	288
5.15.4.6.	Como acessar a pasta do Docker	289
5.15.4.6.1.	MasterTool	289
5.15.4.7.	Usando Memória Externa como Volume Docker	289
5.15.4.8.	Solução de problemas	290
5.15.4.8.1.	Falha ao instalar Container devido a Certificado expirado ou inválido	290

5.15.4.8.2.	Portainer inacessível, retornando ao reinicializar a UCP	290
6.	Manutenção	291
6.1.	Diagnósticos do Módulo	291
6.1.1.	One Touch Diag	291
6.1.2.	Diagnósticos via LED	294
6.1.2.1.	DG (Diagnóstico)	294
6.1.2.2.	WD (Cão-de-guarda)	294
6.1.2.3.	LEDs Conector RJ45	295
6.1.3.	Diagnósticos via Página Web de Sistema	295
6.1.4.	Diagnósticos via Variáveis	295
6.1.4.1.	Diagnósticos Resumidos	296
6.1.4.2.	Diagnósticos Detalhados	298
6.1.4.2.1.	Diagnósticos Detalhados Grupo Target	298
6.1.4.2.2.	Diagnósticos Detalhados Grupo Hardware	298
6.1.4.2.3.	Diagnósticos Detalhados Grupo Exception	298
6.1.4.2.4.	Diagnósticos Detalhados Grupo WebVisualization	299
6.1.4.2.5.	Diagnósticos Detalhados Grupo RetainInfo	299
6.1.4.2.6.	Diagnósticos Detalhados Grupo Reset	300
6.1.4.2.7.	Diagnósticos Detalhados Grupo Thermometer	300
6.1.4.2.8.	Diagnósticos Detalhados Grupo Serial	300
6.1.4.2.9.	Diagnósticos Detalhados Grupo CAN	301
6.1.4.2.10.	Diagnósticos Detalhados Grupo USB	301
6.1.4.2.11.	Diagnósticos Detalhados Grupo Ethernet	302
6.1.4.2.12.	Diagnósticos Detalhados Grupo UserFiles	304
6.1.4.2.13.	Diagnósticos Detalhados Grupo UserLogs	304
6.1.4.2.14.	Diagnósticos Detalhados Grupo MemoryCard	304
6.1.4.2.15.	Diagnósticos Detalhados Grupo WHSB	304
6.1.4.2.16.	Diagnósticos Detalhados Grupo Application	306
6.1.4.2.17.	Diagnósticos Detalhados Grupo Rack	306
6.1.4.2.18.	Diagnósticos Detalhados Grupo ApplicationInfo	306
6.1.4.2.19.	Diagnósticos Detalhados Grupo SNTP	306
6.1.4.2.20.	Diagnósticos Detalhados Grupo OpenVPN	307
6.1.4.2.21.	Diagnósticos Detalhados Grupo Firewall	308
6.1.4.2.22.	Diagnósticos Detalhados Grupo FTP	308
6.1.5.	Diagnósticos via Blocos Funcionais	308
6.1.5.1.	GetTaskInfo	308
6.2.	Visor Gráfico	309
6.3.	Log de Sistema	311
6.4.	Não Carregar a Aplicação na Inicialização	311
6.5.	Problemas mais Comuns	312
6.6.	Solução de Problemas	312
6.7.	Manutenção Preventiva	312
7.	Apêndices	313
7.1.	Gerenciamento de Certificados e Chaves TLS	313
7.1.1.	Geração de Certificados por Easy-RSA	313
7.1.2.	Geração de Certificados por OpenSSL	318
7.1.3.	Geração de Chave TA pelo OpenVPN	320

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.

Com um poderoso processador ARM de 64-bits e 1 GHz, a UCP NX3008 é ideal para controle de máquinas e processos industriais de pequeno a grande porte. Além de diagnósticos avançados e a diversidade de protocolos de comunicação consolidados, conta com recursos de segurança cibernética, firewall, operação remota e capacidade de customização da aplicação do usuário com a instalação de programas externos, graças a plataforma Docker e o sistema operacional Linux que equipa todos os controladores da Série Nexto.



Figura 1: NX3008

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.



VPN: Os produtos Nexto contam com o serviço de VPN embarcado, o que cria um túnel privado de conexão direta com a UCP. Essa funcionalidade, disponível em alguns modelos da família, permite acessar uma rede de controle de forma remota e completamente segura.



FTP: Com suporte a conexões do tipo FTP, os equipamentos da série estão habilitados a trocar dados com um servidor que utilize este mesmo modelo de tecnologia. Essa funcionalidade permite que os arquivos gerados pelo controlador, como logs coletados através de uma função datalogger, sejam acessados à distância.



Linux: Outra funcionalidade inovadora da série é sua plataforma Linux embarcada. O recurso torna possível a virtualização de softwares desenvolvidos para sistemas operacionais com tecnologia Unix. O recurso dá mais versatilidade e velocidade a operação do sistema, pois permite o processamento de múltiplos dados dentro da própria UCP.



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 disto, 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.

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.

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

Código	Descrição	Idioma
CE114000	Nexto Series – Technical Characteristics	Inglês
CT114000	Série Nexto – Características Técnicas	Português
CE114109	NX3008 Technical Characteristics	Inglês
CT114109	Características Técnicas NX3008	Português
CE114700	Nexto Series Backplane Racks Technical Characteristics	Inglês
CT114700	Características Técnicas dos Bastidores da Série Nexto	Português
CE114810	Nexto Series Accessories for Backplane Rack Technical Characteristics	Inglês
CT114810	Características Técnicas Acessórios para Bastidor Série Nexto	Português
CE114902	Nexto Series PROFIBUS-DP Master Technical Characteristics	Inglês
CT114902	Características Técnicas do Mestre PROFIBUS-DP da Série Nexto	Português
CE114908	NX5110 and NX5210 PROFIBUS-DP Heads Technical Characteristics	Inglês
CT114908	Características Técnicas Interfaces Cabeça PROFIBUSDP NX5110 e NX5210	Português
MU214600	Nexto Series User Manual	Inglês
MU214000	Manual de Utilização Série Nexto	Português
MU214620	NX3008 CPU User Manual	Inglês
MU214109	Manual de Utilização UCP NX3008	Português
MU299609	MasterTool IEC XE User Manual	Inglês
MU299048	Manual de Utilização MasterTool IEC XE	Português
MU299611	Mastertool X User Manual	Inglês
MU299049	Manual de Utilização Mastertool X	Português
MP399609	IEC 61131 Programming Manual	Inglês
MP399048	Manual de Programação IEC 61131	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
MU214603	Nexto Series HART Manual	Inglês
MU214606	MQTT User Manual	Inglês
MU214609	OPC UA Server for Altus Controllers User Manual	Inglês
MU214610	PID - Advanced Control Functions User Manual	Inglês
MU214621	Nexto Series PROFINET Manual	Inglês
MU214622	LibSparkPlug User Manual	Inglês
MU223603	IEC 60870-5-104 Server Device Profile Document	Inglês
NAP151	Utilização do Tunneller OPC	Português
NAP169	RSTP in Nexto CPUs	Inglês

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, ligue para 0800 510 9500, envie e-mail para suporte@altus.com.br ou acesse nosso site www.altus.com.br/suporte. Se o equipamento já estiver instalado, tenha as seguintes informações em mãos ao solicitar assistência:

- Modelos dos equipamentos utilizados e configuração do sistema instalado.
- Número de série do produto.
- Revisão do equipamento e versão do software executivo, indicadas na etiqueta afixada na lateral do produto.
- Informações sobre o modo de operação da UCP, obtidas por meio do Mastertool.
- Conteúdo do programa da aplicação, obtido por meio do Mastertool.
- Versão do Mastertool 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 NX3008 da Série Nexto.

2.1. Painéis e Conexões

A figura a seguir mostra o painel frontal da UCP.



Figura 3: NX3008

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

Esta UCP conta com uma tecla disponível ao usuário. A Tabela abaixo mostra a descrição da tecla. Para maiores informações sobre a tecla de diagnóstico, consulte a seção [One Touch Diag](#).

Tecla	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.

Tabela 3: Descrição da Tecla

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..NET 3	Conector tipo RJ45 de comunicação no padrão 10/100/1000Base-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	Para obter mais informações sobre utilização, consulte a seção Configuração da Interface Serial .
CAN	Para obter mais informações sobre utilização, consulte a seção Configuração da Interface CAN (Controller Area Network) .
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 .
USB Host	Para obter mais informações sobre utilização, consulte seção Dispositivos USB .

Tabela 4: Interfaces de Conexão

2.2. Características do Produto

2.2.1. Características Gerais

	NX3008
Ocupação do bastidor	2 posições sequenciais
Consumo de corrente do bastidor	-
Fonte de alimentação integrada	Sim
Ethernet TCP/IP interface integrada	3
Interface Serial	1
Interface CAN	1
Porta USB Host	1
Interface Cartão de Memória	1
Relógio de tempo real (RTC)	Sim Resolução de 1 ms, máx. variação de 2 segundos por dia.
Cão de guarda	Sim

	NX3008
Indicação de estado e diagnóstico	Visor gráfico LEDs Página 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	24
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	2
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)	Não
Suporte a troca a quente	Sim
Registro de eventos (SOE)	Não
Protocolo	-
Tamanho máximo da fila de eventos	-
Páginas web de usuário (Webvisu)	Sim
Firewall	Sim
Docker	Sim
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.

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

	NX3008
Memória de variáveis de entrada de representação direta (%I)	64 Kbytes
Memória de variáveis de saída de representação direta (%Q)	64 Kbytes
Memória de variáveis de representação direta (%M)	32 Kbytes
Memória de variáveis simbólicas	40 Mbytes
Memória de variáveis retentivas ou persistentes	1 Mbytes
Memória de dados redundantes total	-
Memória de variáveis de entrada de representação direta (%I)	-
Memória de variáveis de saída de representação direta (%Q)	-
Memória de variáveis de representação direta (%M)	-
Memória de variáveis simbólicas	-
Memória total	
Memória de programa (limitado a 32 MBytes) + Memória de código fonte (backup) + Memória de arquivos de Webvisu	256 Mbytes
Memória de arquivos de usuário Memória da UCP + Memória do Docker	4 GBytes

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.

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.

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 persistentes simbólicas mantêm seus valores e quando o valor é perdido pode ser encontrada na tabela abaixo. Além do tamanho da área persistente declarado na tabela acima, esses 44 bytes são reservados para armazenar informações sobre as variáveis persistentes (não disponíveis 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
Runtime/Watchdog Reset	-	-	-

Tabela 8: Comportamento das Variáveis após Evento

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 programa: Área da memória que corresponde ao tamanho máximo permitido para a aplicação de usuário.

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 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.

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 na seção [Memória de Arquivos de Usuário](#).

2.2.4. Protocolos

	NX3008	Interface
Comunicação com o software de programação	Sim	NET 1 / NET 2 / NET 3 / USB
Protocolo aberto	Sim	COM 1 / USB
MODBUS RTU Mestre	Sim	COM 1
MODBUS RTU Escravo	Sim	COM 1
MODBUS TCP Cliente	Sim	NET 1 / NET 2 / NET 3
MODBUS TCP Servidor	Sim	NET 1 / NET 2 / NET 3
MODBUS RTU via TCP Cliente	Sim	NET 1 / NET 2 / NET 3
MODBUS RTU via TCP Servidor	Sim	NET 1 / NET 2 / NET 3
CANopen Mestre	Sim	CAN
CANopen Escravo	Não	-
CAN low level	Sim	CAN
SAE J-1939	Sim	CAN
OPC DA Servidor	Sim	NET 1 / NET 2 / NET 3
OPC UA Servidor	Sim	NET 1 / NET 2 / NET 3
EtherCAT Mestre	Sim	NET 1 / NET 2 / NET 3
SNMP Agente	Sim	NET 1 / NET 2 / NET 3
SOE (dados orientados ao evento)	Não	-
IEC 60870-5-104 Servidor	Sim	NET 1 / NET 2 / NET 3
EtherNet/IP Scanner	Sim	NET 1 / NET 2 / NET 3
EtherNet/IP Adapter	Sim	NET 1 / NET 2 / NET 3
MQTT Cliente	Sim	NET 1 / NET 2 / NET 3 / USB
SparkPlugB	Sim	NET 1 / NET 2 / NET 3 / USB
SNTP Cliente (para sincronismo do relógio)	Sim	NET 1 / NET 2 / NET 3 / USB
PROFINET Controller	Sim	NET 1 / NET 2 / NET 3
PROFINET Device	Não	-
OpenVPN Client	Sim	NET 1 / NET 2 / NET 3
OpenVPN Server	Sim	NET 1 / NET 2 / NET 3
FTP Server	Sim	NET 1 / NET 2 / NET 3 / USB
RSTP	Sim	NET 2 / NET 3
MRP	Sim	NET 2 / NET 3

Tabela 9: Protocolos

Nota:

USB: Necessário o uso de adaptador Serial, Wi-Fi ou Modem.

Comunicação com o software de programação: Para realizar a comunicação com a UCP a partir de uma interface, que não seja a NET 1, é necessário adicionar um gateway com o endereço IP da determinada interface.

2.2.5. Interface Serial

2.2.5.1. COM 1

	COM 1
Conector	Borne, terminal D+ e D- com shield
Interface Física	RS-485
Direção de Comunicação	Half duplex
Máx. Transmissores RS-485	32
Terminação	Sim (Configurável)
Seção do Condutor	0,5 mm ²
Taxa de Transmissão	2400, 4800, 9600, 19200, 38400, 57600, 115200 bps
Isolação	
Lógica para porta serial	1000 Vac / 1 minuto
Porta serial para terra de proteção ⚡	1000 Vac / 1 minuto

Tabela 10: Características da Interface Serial COM 1

Nota:

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

2.2.6. Interface CAN

	CAN
Conector	Borne, terminal H e L com shield
Interface Física	Barramento CAN
Normas Suportadas	CAN 2.0A 2.0B (identificadores de 11-bit e 29-bit)
Max. Número de nós	64
Terminação	Sim (Configurável)
Secção do Condutor	0,5 mm ²
Taxa de Transmissão	10, 20, 50, 100, 125, 250, 500, 1000 kbit/s
Isolação	
CAN para lógica	1000 Vac / 1 minuto
CAN para terra de proteção ⚡	1000 Vac / 1 minuto

Tabela 11: Característica da Interface CAN

2.2.7. Interface USB

	USB
Conector	USB A Fêmea
Interface Física	USB V2.0
Taxa de Transmissão	1.5 Mbps (Low-Speed), 12 Mbps (Full-Speed) e 480 Mbps (Hi-Speed)
Corrente Máxima	500 mA
Dispositivos Suportados	Dispositivo de armazenamento em massa Conversor serial USB RS-232 Modem USB 3G/4G Adaptador USB Wi-Fi
Isolação USB para lógica USB para terra de proteção 	Não isolado 1000 Vac / 1 minuto

Tabela 12: Característica da Interface USB

Notas:

Conversor serial USB RS-232: Consulte a lista de dispositivos suportados na seção [Conversor USB para RS-232](#).

Modem USB 3G/4G: Consulte a lista de dispositivos suportados na seção [Dispositivos Modem](#).

Adaptador USB Wi-Fi: Consulte a lista de dispositivos suportados na seção [Adaptadores Wi-Fi](#).

ATENÇÃO:

A UCP suporta o uso de apenas um dispositivo USB por vez. Dispositivos como HUBs USB, por exemplo, não são suportados.

2.2.8. Interfaces Ethernet

2.2.8.1. 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	10/100/1000 Mbps
Camada Física	10BASE-T/100BASE-TX/1000BASE-T
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	LED - verde 1000 Mbps (link/atividade) LED - amarelo 100 Mbps (link/atividade) LEDs - verde e amarelo 10 Mbps (link/atividade)
Isolação	
Interface Ethernet para lógica	1500 Vac / 1 minuto
Interface Ethernet para Interface Ethernet	1500 Vac / 1 minuto
Interface Ethernet para terra de proteção 	1500 Vac / 1 minuto

Tabela 13: Características da Interface Ethernet NET 1

ATENÇÃO:

A NET1 não suporta cabos de 4 fios, sendo necessário um cabo CAT5 completo.

2.2.8.2. 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	10BASE-Tc/100BASE-TX
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	LED – amarelo 100 Mbps (link/atividade) LEDs – verde e amarelo 10 Mbps (link/atividade)
Isolação	
Interface Ethernet para lógica	1500 Vac / 1 minuto
Interface Ethernet para Interface Ethernet	1500 Vac / 1 minuto
Interface Ethernet para terra de proteção 	1500 Vac / 1 minuto

Tabela 14: Características da Interface Ethernet NET 2

2.2.8.3. NET 3

	NET 3
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	10BASE-Tc/100BASE-TX
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	LED – amarelo 100 Mbps (link/atividade) LEDs – verde e amarelo 10 Mbps (link/atividade)
Isolação	
Interface Ethernet para lógica	1500 Vac / 1 minuto
Interface Ethernet para Interface Ethernet	1500 Vac / 1 minuto
Interface Ethernet para terra de proteção 	1500 Vac / 1 minuto

Tabela 15: Características da Interface Ethernet NET 3

2.2.9. 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	MicroSD
Sistema de arquivos	FAT32
Remover cartão de forma segura	Sim, através de menu específico para essa função.

Tabela 16: 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.

Atenção:

É recomendado usar o Cartão de Memória como principal meio de armazenamento quando comparado com armazenamento USB.

2.2.10. Alimentação

Tensão de Entrada Nominal	24 Vdc
Potência de saída máxima	15W ^{1,2}
Corrente de saída máxima	3 A ¹
Tensão de Entrada	18 a 30 Vdc
Máxima Corrente de Entrada (in-rush)	15 A
Máxima Corrente de Entrada	1,5 A ¹
Tempo máximo de interrupção da tensão de entrada	10 ms @ 24 Vdc
Isolação	
Entrada para lógica	1000 Vac / 1 minuto
Entrada para terra de proteção PE	1000 Vac / 1 minuto
Seção do Condutor	0,5 mm ²
Proteção inversão de polaridade	Sim
Fusível interno	Sim
Proteção contra curto-circuito na saída	Não
Proteção contra sobrecorrente na saída	Não

Tabela 17: Características da Fonte de Alimentação

Notas:

¹ **Potência/Corrente de saída máxima:** Para utilização com potência/corrente de saída máxima estendida de 20W/4A, algumas condições devem ser respeitadas: uso exclusivo de módulos de E/S Nexto Jet; redução da temperatura ambiente máxima de operação para 50°C; não realizar troca à quente de módulos de E/S, sob o risco de afetar o funcionamento do

2. DESCRIÇÃO TÉCNICA

sistema; módulos NJ6000, NJ6010 e NJ6100 devem ter revisão AB ou superior. Neste caso a informação de Máxima Corrente de Entrada passa para 2,0A.

² **Suporte à NX8000:** A operação da NX3008 com suporte à NX8000 alimentando o barramento visa permitir a extensão de potência/corrente de saída máxima para 30W/6A. Para isto, algumas condições devem ser respeitadas: as condições estão detalhadas na seção [Utilizando o módulo fonte de alimentação NX8000](#).

2.2.11. Características Ambientais

Dissipação	9 W
Temperatura de operação	-20 a 60 °C @ 15W -20 a 50 °C @ 20W
Temperatura de armazenamento	-40 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	330 g
Peso com embalagem	380 g

Tabela 18: 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. 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	Versão de Firmware
NX3008	3.40 ou superior	1.10.0.0 ou superior

Tabela 19: Compatibilidade com Outros Produtos

Além disso, ao longo do roteiro de desenvolvimento do Mastertool, 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 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 o perfil Process o intervalo pode ser configurado com valores de 1 ms a 750 ms. Já para o perfil Fast 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	Tipo da Variável	Tempo (μ s)
1000 Contatos	LD	BOOL	2,1
1000 Divisões	LD, ST	INT	9,2
		REAL	17,0
1000 Multiplicações	LD, ST	INT	6,4
		REAL	8,2
1000 Somas	LD, ST	INT	4,4
		REAL	8,2

Tabela 20: Tempos de Instrução

2.4.4. Tempos de Inicialização

Esta UCP da Série Nexto possui tempo de inicialização de 30 segundos, sendo que a tela inicial com o logotipo é apresentada depois de 2 segundos da energização.

2.5. Dimensões Físicas

Dimensões em mm.

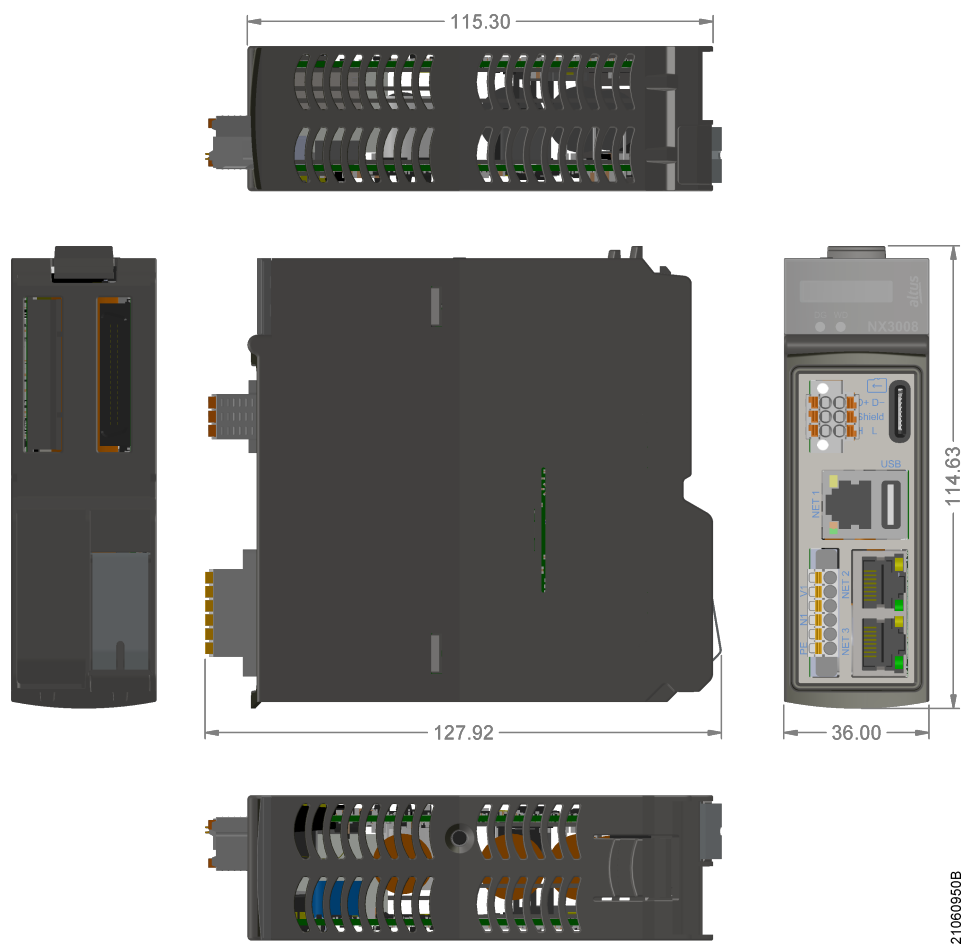


Figura 4: Dimensões Físicas

2.6. Dados para Compra

2.6.1. Itens Integrantes

A embalagem do produto contém os seguintes itens:

- Módulo NX3008
- Conector 6 terminais com fixação
- Conector 2x3 terminais com fixação

2.6.2. Código do Produto

Os seguintes códigos devem ser usados para compra do produto:

Código	Descrição
NX3008	UCP com 3 portas Ethernet, 1 USB, 1 serial, 1 CAN, interface para cartão de memória, suporte a expansão de barramento, fonte de alimentação integrada e suporte a páginas Web de usuário

Tabela 21: Código do Produto

2.7. Produtos Relacionados

Os seguintes produtos devem ser adquiridos separadamente quando necessário:

Código	Descrição
MT8500	MasterTool IEC XE
MT9000	Mastertool X
AL-2600	Derivador e terminador de rede RS-485
AL-2306	Cabo RS-485 p/ rede MODBUS ou CAN
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
NX9010	Bastidor de 8 Posições Sem Troca a Quente
NX9020	Base com 2 posições para montagem em painel

Tabela 22: Produtos Relacionados

Notas:

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.

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

NX9020: Base com 2 slots para montagem em painel.

3. Instalação

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

CUIDADO

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

3.1. Instalação Mecânica

Este produto deve estar posicionado na posição 0 do bastidor da Série Nexto. São necessárias duas posições sequenciais, isto significa que este ocupará as posições 0 e 1 de um dado bastidor.

A montagem mecânica deste módulo é descrita no Manual de Utilização Série Nexto – MU214000.

3.2. Instalação Elétrica

3.2.1. Utilizando a fonte de alimentação integrada

A figura abaixo ilustra o diagrama elétrico da UCP NX3008 instalada em um bastidor da Série Nexto.

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

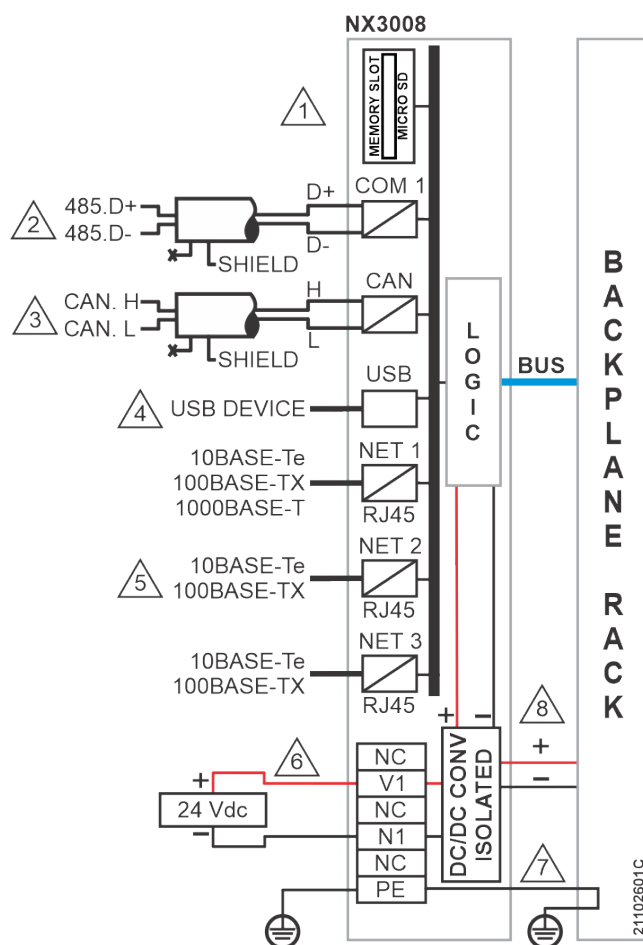


Figura 5: Diagrama Elétrico da UCP NX3008

Notas do Diagrama:

-  Interface para cartão de memória.
-  Interface RS-485.
-  Interface CAN.
-  Interface USB 2.0
-  Interfaces Ethernet padrão NET 1 10BASE-T_e/100BASE-TX/1000BASE-T e NET 2/NET 3 10BASE-T_e/100BASE-TX.
-  A alimentação é conectada aos terminais V1 e N1. Utilizar cabo de 0,5 mm².
-  A UCP é aterrada através dos bastidores da Série Nexto. Recomenda-se reforçar o aterramento conectando ao terminal PE. Utilizar cabo de 0,5 mm².
-  A UCP alimenta os demais módulos através da conexão ao bastidor.
-  Terminal terra de proteção.

3.2.2. Utilizando o módulo fonte de alimentação NX8000

Se a potência requisitada pelo conjunto de módulos de E/S conectados ao barramento for superior à capacidade oferecida pela fonte integrada da NX3008, pode-se adicionar um módulo fonte de alimentação NX8000 para alimentar o barramento. Nesta configuração, a NX3008 ainda deve ser energizada (para alimentação da eletrônica interna), contudo a sua saída de alimentação do barramento permanecerá desligada, isto é, não haverá paralelismo, e toda a corrente demandada pelos módulos de E/S do barramento será entregue pela NX8000.

PERIGO

Para operar nesta configuração, a NX3008 deve ter OBRIGATORIAMENTE a versão de firmware 1.14.35.0 ou superior e a versão de Mastertool 3.63 ou superior. Certifique-se que a versão de firmware está atualizada antes da instalação.

Caso esta condição não seja atendida, NÃO conecte a NX8000 ao barramento, sob risco de dano permanente ao hardware.

PERIGO

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

A figura abaixo ilustra o diagrama elétrico da UCP NX3008 utilizando o módulo fonte de alimentação NX8000 em um bastidor da Série Nexto. A disposição dos conectores e bornes na figura é meramente ilustrativa.

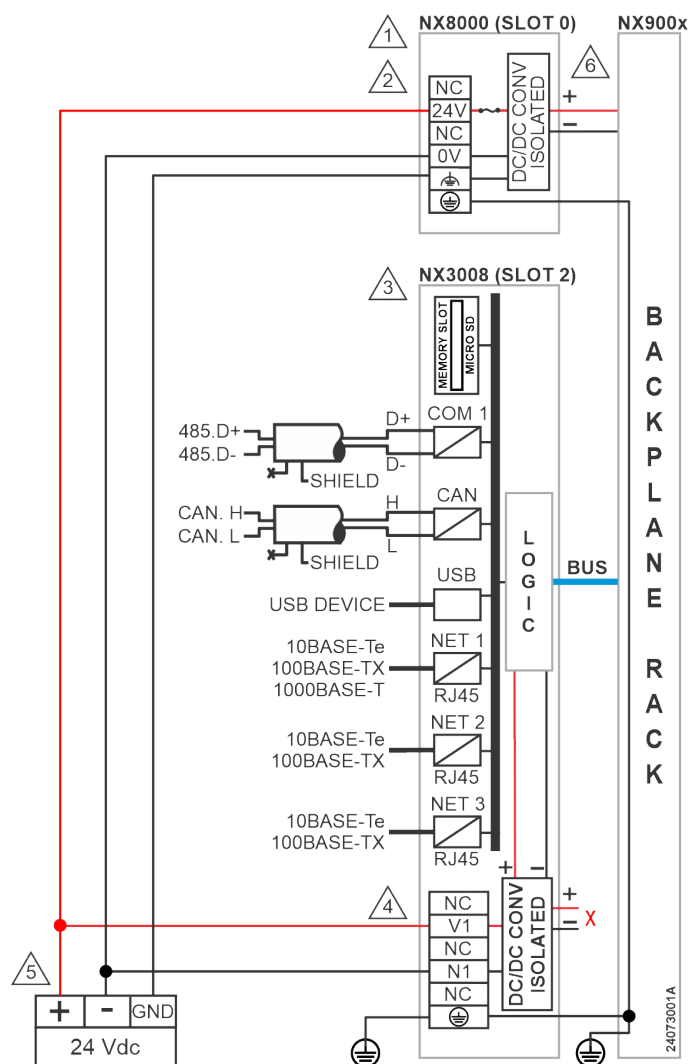


Figura 6: Diagrama Elétrico da UCP NX3008 utilizando o módulo fonte de alimentação NX8000

Notas do Diagrama:

- ① A NX8000 deve ser posicionada obrigatoriamente no Slot 0.
 - ② A NX3008 deve ser posicionada obrigatoriamente no Slot 2.
 - ③ A fonte de alimentação está conectada aos terminais 24V e 0V da NX8000. Utilizar cabos de 2,5 mm². Terminais NC: Não Conectados.
 - ④ A fonte de alimentação está conectada aos terminais V1 e N1 da NX3008. Utilizar cabo de 0,5 mm². Terminais NC: Não Conectados.
 - ⑤ A NX8000 e a NX3008 devem ser alimentadas com a mesma fonte de alimentação 24V.
 - ⑥ O módulo NX8000 alimenta os outros módulos do barramento através da conexão com o bastidor. Nesta condição a NX3008 não fornece alimentação para os módulos do barramento.
- ⊕ Terminal terra de proteção.

3.3. Conexão com a Rede Ethernet

As interfaces isoladas de comunicação NET 1, NET 2 e NET 3 possibilitam a conexão com uma rede Ethernet, onde a interface NET 1 é mais indicada para comunicação com o Mastertool.

A conexão com a rede Ethernet utiliza cabos tipo par trançado (10BASE-T/100BASE-TX/1000BASE-T), 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 23: 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, 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	0.0.0.0

Tabela 24: 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. Para maiores detalhes sobre configuração e alteração de parâmetros de rede, verifique a seção [Configuração das Interfaces Ethernet](#).

	NET 3
Endereço IP	192.168.17.1
Máscara de subrede	255.255.255.0
Endereço do Gateway	0.0.0.0

Tabela 25: Configuração de Parâmetros Padrão para a Interface Ethernet NET 3

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 3 podem ser alterados através do Mastertool. 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 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.

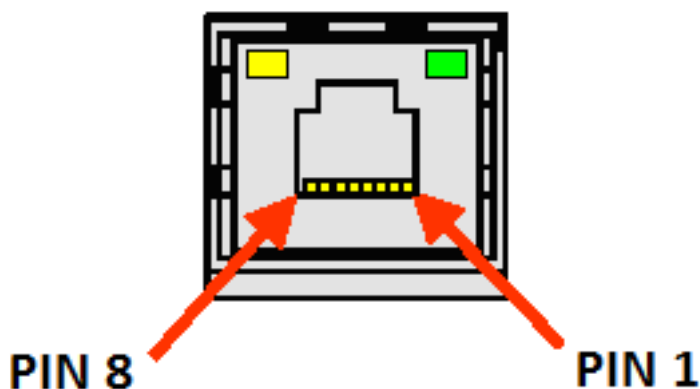


Figura 7: Conector Fêmea RJ45

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
7	NU	Não usado
8	NU	Não usado

Tabela 26: Pinagem do Conector Fêmea RJ45 - 10BASE-TE e 100BASE-TX

A tabela abaixo apresenta o conector RJ45 fêmea das UCPs Nexto, com a identificação e a descrição da pinagem válida para o nível físico tipo 1GBASE-T.

Pino	Sinal	Descrição
1	TxRx A +	Bidirecional, positivo
2	TxRx A -	Bidirecional, negativo
3	TxRx B +	Bidirecional, positivo
4	TxRx C +	Bidirecional, positivo
5	TxRx C -	Bidirecional, negativo
6	TxRx B -	Bidirecional, negativo
7	TxRx D +	Bidirecional, positivo
8	TxRx D -	Bidirecional, negativo

Tabela 27: Pinagem do Conector Fêmea RJ45 - 1GBASE-T

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 de Rede Serial RS-485 e CAN

Conforme ilustrado no diagrama [Instalação Elétrica](#), as interfaces RS-485 e CAN usam dois sinais de comunicação e um terra. O cabo recomendado é o AL-2306, usando um dos dois pares e a blindagem. Se o controlador for colocado em uma das extremidades da rede, a terminação interna deverá ser habilitada (consulte as seções de configuração [Configuração da Interface CAN \(Controller Area Network\)](#) e [Configurações Avançadas](#) para CAN e RS-485, respectivamente).

3.5. 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.

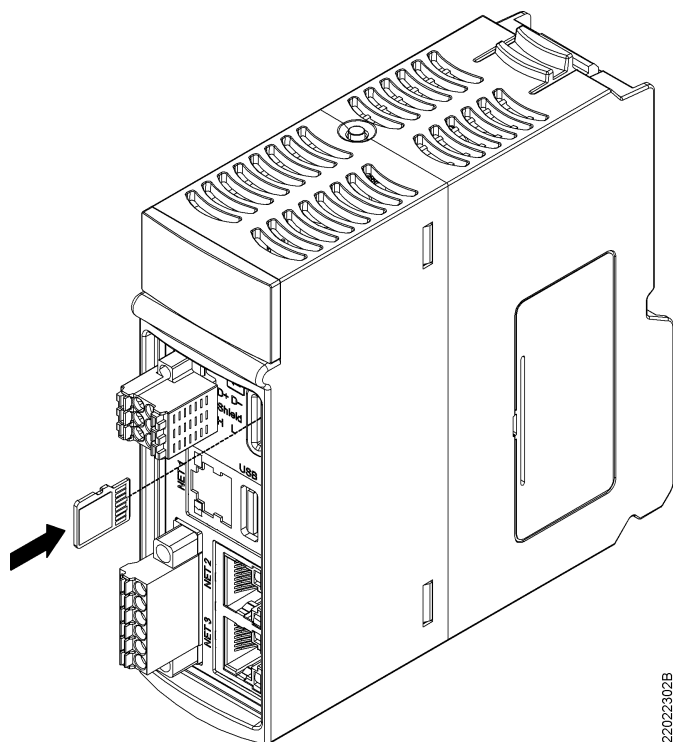


Figura 8: Inserindo o Cartão de Memória

Quando o cartão estiver instalado corretamente aparecerá um símbolo no visor gráfico da UCP. Para remover o cartão com segurança, deve-se usar o botão OTD da UCP, navegar até o MENU do cartão de memória, e selecionar a opção para desmontar o cartão, esperar para que o símbolo do cartão desapareça do visor gráfico e então retirar o cartão. Após isso, basta retirá-lo do compartimento, conforme mostra a figura abaixo.

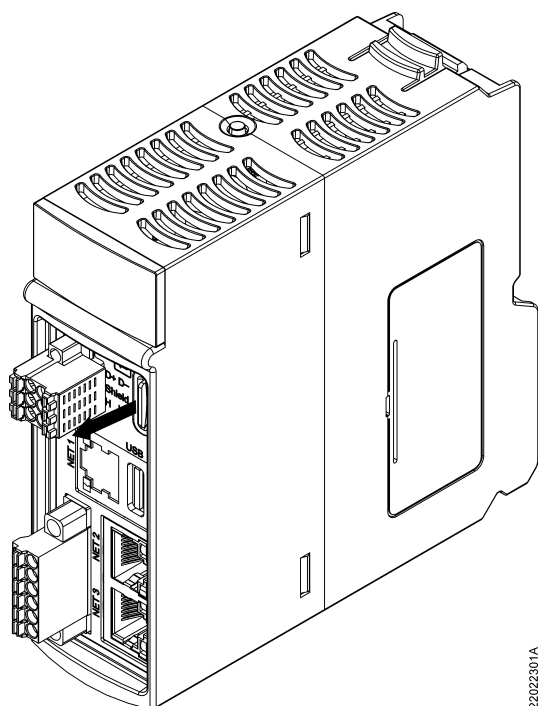


Figura 9: Retirada do Cartão de Memória na UCP

3.6. Instalação da Arquitetura

3.6.1. 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.

3.7. Instalação do Programador

Para realizar a instalação do software de desenvolvimento Mastertool é necessário 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, 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 (IDE). O Mastertool 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, Manual de Programação IEC 61131 ou o padrão IEC 61131-3.

4.1. Organização e Acesso à Memória

Diferentemente de outros dispositivos da Série Nexto (baseados em UCP big-endian), este modelo de UCP é baseada em uma UCP ARM, que utiliza a tradicional organização de memória little-endian (a mesma encontrada nos processadores x86 e Intel). Nesse tipo de organização de memória, o byte menos significativo é armazenado primeiro e sempre será o menor endereço (por exemplo, %QB0 sempre será menos significativo que %QB1, conforme mostrado na tabela abaixo, onde, para CPUNEXTO, a letra O é o byte 0 e a letra C 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 28: 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									
%QX2.7	%QB 02	%QW	00		%QB02	%QW	00	%QD	
%QX2.6									
%QX2.5									
%QX2.4									
%QX2.3									
%QX2.2									
%QX2.1									
%QX2.0									
%QX3.7	%QB 03	02			%QB03	%QW	01	%QD	
%QX3.6									
%QX3.5									
%QX3.4									
%QX3.3									
%QX3.2									
%QX3.1									
%QX3.0									
%QX4.7	%QB 04	%QW		00	%QB04	%QW	02	%QD	
%QX4.6									
%QX4.5									
%QX4.4									
%QX4.3									
%QX4.2									
%QX4.1									
%QX4.0									
%QX5.7	%QB 05	04			%QB05	%QW	03	%QD	
%QX5.6									
%QX5.5									
%QX5.4									
%QX5.3									
%QX5.2									
%QX5.1									
%QX5.0									
%QX6.7	%QB 06	%QW	04		%QB06	%QW	04	%QD	
%QX6.6									
%QX6.5									
%QX6.4									
%QX6.3									
%QX6.2									
%QX6.1									
%QX6.0									
%QX7.7	%QB 07	06			%QB07				
%QX7.6									
%QX7.5									
%QX7.4									
%QX7.3									
%QX7.2									
%QX7.1									
%QX7.0									

Tabela 29: 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 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.

Os perfis de projeto disponíveis variam conforme a versão do programador:

- **MT8500:** disponibiliza os perfis Simples, Básico, Normal, Experiente, Personalizado e Perfil de Máquina, com diferentes níveis de complexidade, regras de nomenclatura e estruturas predefinidas.
- **MT9000:** disponibiliza os perfis Fast e Process, que mantêm o modelo baseado em tarefas, porém sem restrições de nomenclatura e com maior flexibilidade de configuração.

A seleção do perfil é realizada no assistente de criação do projeto. Cada perfil define a estrutura inicial da aplicação e, no caso do MT8500, também impõe padrões de nomenclatura para tarefas e programas.

Durante a compilação, o ambiente realiza a verificação da consistência da aplicação conforme o perfil adotado.

As seções a seguir detalham as características de cada perfil.

4.2.1. Simples - MT8500

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 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	100 ms	-

Tabela 30: Tarefa no Perfil Simples

4.2.2. Básico - MT8500

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	-	-

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

Tabela 31: Tarefas no Perfil Básico

4.2.3. Normal - MT8500

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	100 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 32: Tarefas no Perfil Normal

4.2.4. Experiente - MT8500

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	100 ms	-

Tarefa	POU	Prioridade	Tipo	Intervalo	Evento
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 33: Tarefas no Perfil Experiente

4.2.5. Personalizado - MT8500

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	100 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 34: Tarefas no Perfil Personalizado

4.2.6. Perfil de Máquina - MT8500

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 consequentemente pode interromper a execução dela a qualquer momento.

Tarefa	POU	Prioridade	Tipo	Intervalo	Evento
MainTask	MainPrg	13	Cíclica	100 ms	-
TimeInterruptTask00	TimeInterruptPrg00	01	Cíclica	4 ms	-

Tarefa	POU	Prioridade	Tipo	Intervalo	Evento
--------	-----	------------	------	-----------	--------

Tabela 35: Tarefas no Perfil de Máquina

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

4.2.7. Perfis do MT9000

No MT9000, o modelo de perfis foi simplificado, sendo disponibilizados apenas dois perfis de projeto: *Fast* e *Process*. Diferentemente do MT8500:

- Não há restrições de nomenclatura para tarefas ou POUs
- Não existem estruturas fixas obrigatórias de tarefas
- O limite de tarefas é definido exclusivamente pelas capacidades do controlador

4.2.7.1. Perfil Fast (MT9000)

O perfil *Fast* é o equivalente ao *Perfil de Máquina* do MT8500.

Diferentemente do *Perfil de Máquina*, não há restrições quanto à nomenclatura de tarefas ou POUs, e a configuração pode ser ajustada conforme a necessidade da aplicação.

4.2.7.2. Perfil Process (MT9000)

O perfil *Process* é destinado a aplicações que demandam maior flexibilidade e modularização da lógica de controle.

As prioridades, intervalos e associações entre tarefas e programas são totalmente configuráveis.

Este perfil substitui, no MT9000, os perfis Básico, Normal, Experiente e Personalizado, consolidando suas funcionalidades em um único modelo mais flexível.

4.2.8. Tabela Geral - MT8500

		Perfis de Projeto					
		Simples	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 Prioridade Quantidade	Perfis de Projeto					Evento <n> [00..31]

Tabela 36: Tabela Geral de Perfis x Tarefas

ATENÇÃO

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

4.2.9. Número Máximo de Tarefas

No MT8500, o número máximo de tarefas depende do perfil de projeto e do modelo da UCP, conforme descrito a seguir.

No MT9000, o número de tarefas não é limitado por perfil, sendo definido exclusivamente pelas capacidades do controlador utilizado.

		NX3008					
		Tipo de Tarefa					
		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 37: Número Máximo de Tarefas IEC NX3008

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](#).

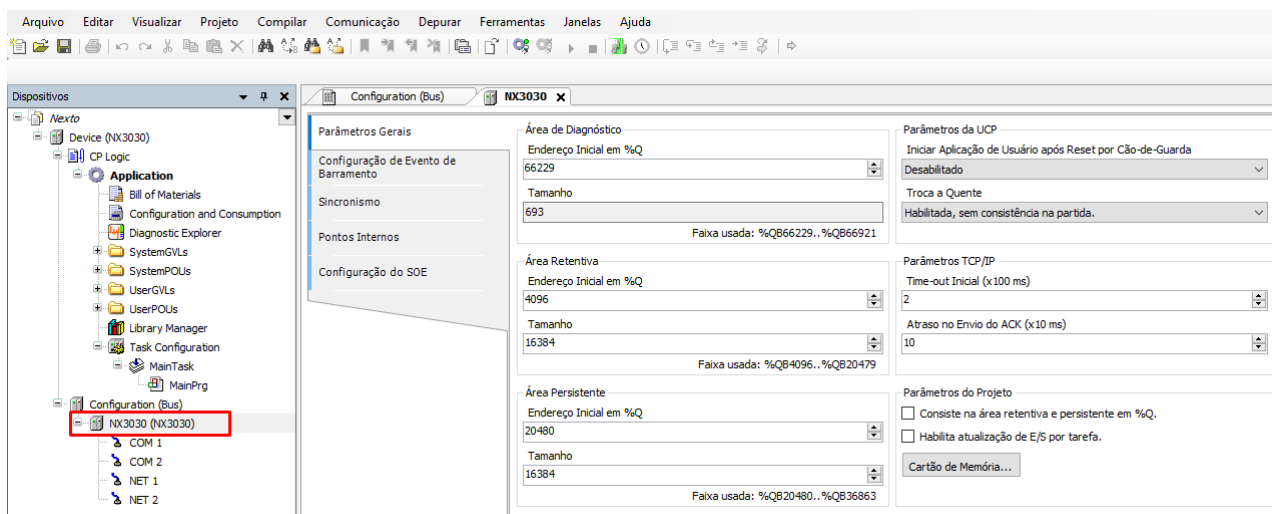


Figura 10: Configuração da UCP

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.

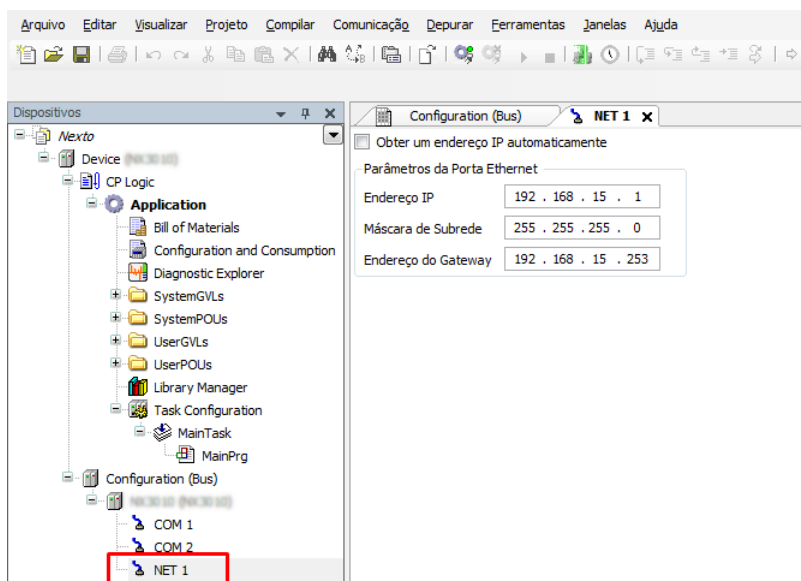


Figura 11: Configurando a Porta de Comunicação da UCP

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 do Mastertool 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 descrito na seção [Protocolos](#), oferecem diversos protocolos de comunicação. Alguns protocolos possuem um procedimento de configuração específico e não são adicionados diretamente às interfaces de comunicação. Para os demais protocolos, a inserção pode ser realizada clicando com o botão direito sobre a interface de comunicação desejada, selecionando a opção de adicionar o dispositivo e, por fim, configurando a instância do protocolo conforme descrito na seção [Configuração de Protocolos](#). A seguir apresenta-se um exemplo.

4.5.1. MODBUS RTU

O primeiro passo para configurar o MODBUS RTU, em modo escravo, é incluir a instância na COM desejada (nesse caso, COM 1). Clicar com o botão direito sobre a *COM* e selecionar *Adicionar Dispositivo...*, conforme mostra a figura abaixo:

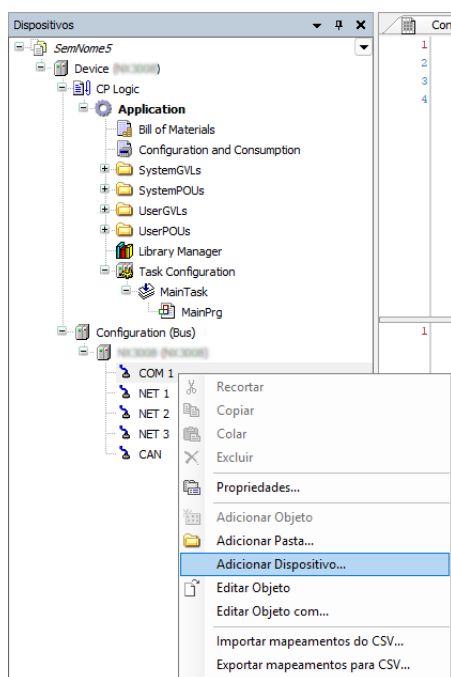


Figura 12: Adicionando a Instância

Após, surgirão na tela os protocolos disponíveis ao usuário. Definir o modo de configuração do protocolo, selecionar a opção *MODBUS Symbol RTU Slave*, para configuração por *Mapeamento Simbólico* ou *MODBUS RTU Slave*, para configuração por *Representação Direta (%Q)*. A seguir, clicar em *Acrescentar*, conforme mostra a figura abaixo.

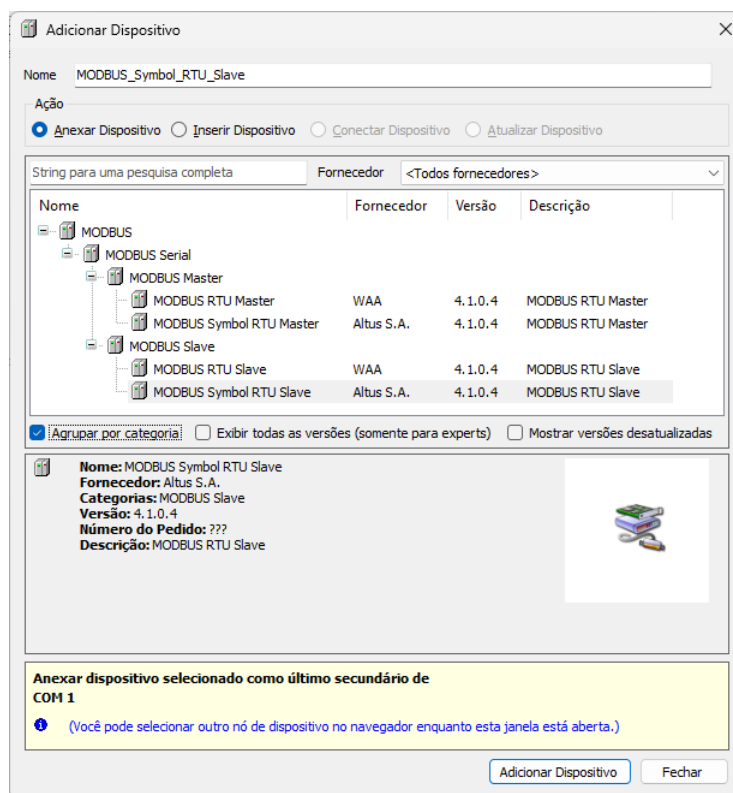


Figura 13: Selecionando o Protocolo

4.5.2. MODBUS Ethernet

O primeiro passo para configurar o MODBUS Ethernet (Cliente neste exemplo), é incluir a instância na NET desejada. Clicar com o botão direito sobre a *NET* e selecionar *Acrescentar Dispositivo...*, conforme mostra a figura abaixo.

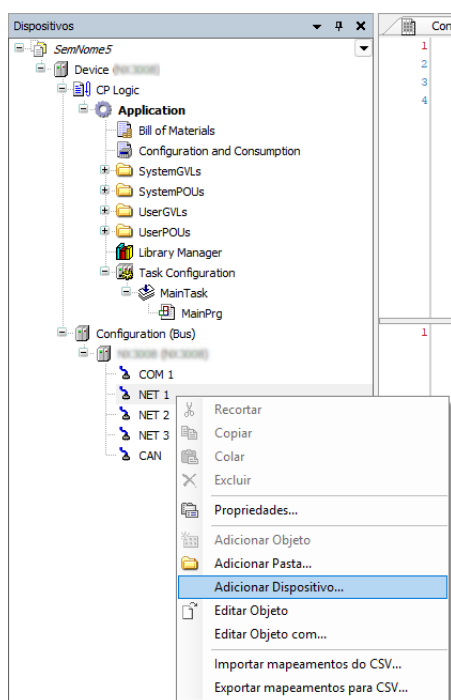


Figura 14: Adicionando a Instância

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 *Acredenciar*, conforme mostra a figura abaixo.

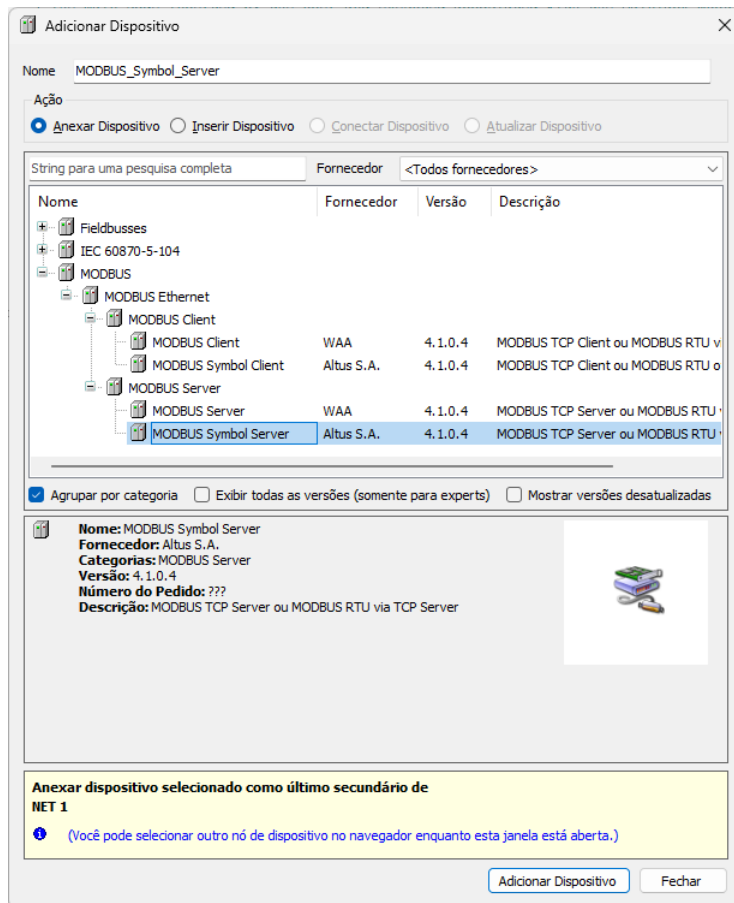


Figura 15: Selecionando o Protocolo

4.6. Localizando o Dispositivo

Para estabelecer a comunicação da UCP com a ferramenta Mastertool, 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 realiza a busca por dispositivos e apresenta as UCPs encontradas na rede da interface Ethernet da estação onde está sendo executada a ferramenta.

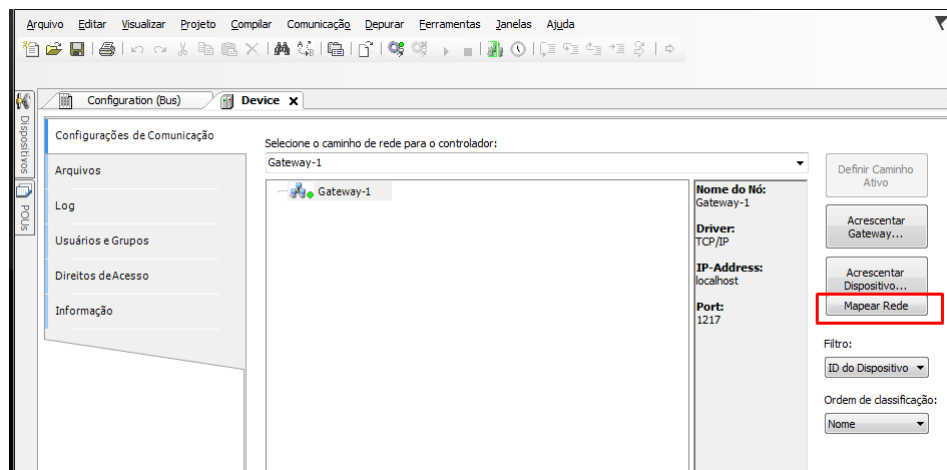


Figura 16: Localizando a UCP

Caso não haja um gateway previamente configurado, o mesmo pode ser adicionado através do botão *Acréscitar 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.

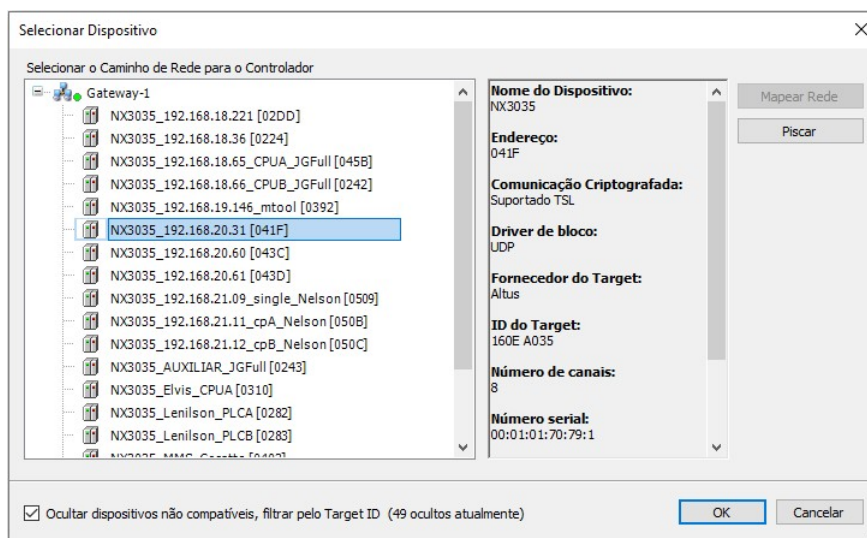


Figura 17: Selecionando a UCP

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 não conseguirá localizar o dispositivo. Neste caso, recomenda-se a utilização do comando *Easy Connection* disponível no menu Comunicação da ferramenta.

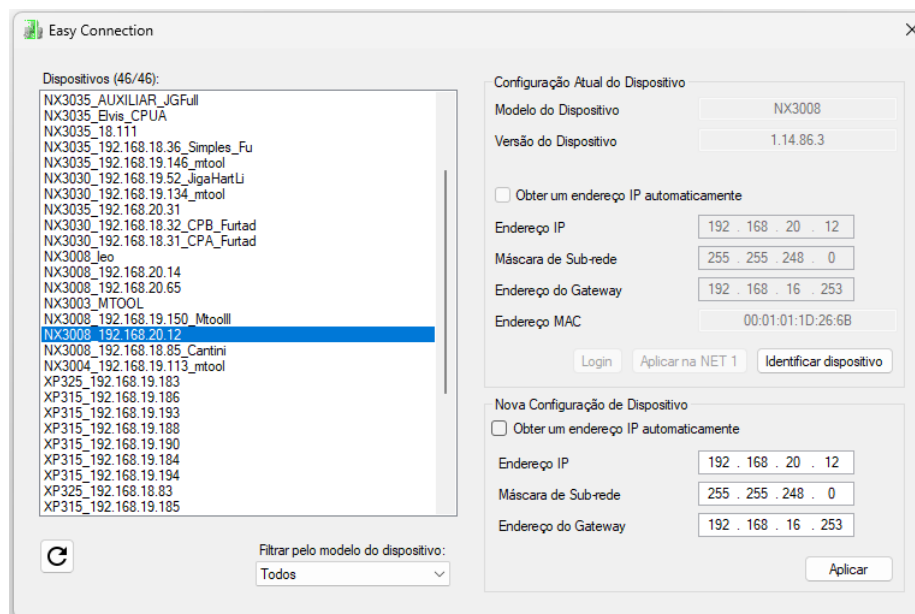


Figura 18: Easy Connection

Este comando realiza uma comunicação em nível MAC com a interface NET 1 do dispositivo, permitindo alterar permanentemente 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, permitindo localizar e selecionar o dispositivo para comunicação. A descrição completa do comando *Easy Connection* pode ser encontrada no Manual de Utilização do Mastertool.

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 conforme mostra o exemplo da figura a seguir. Essa operação pode levar alguns segundos, dependendo do tamanho do arquivo gerado.

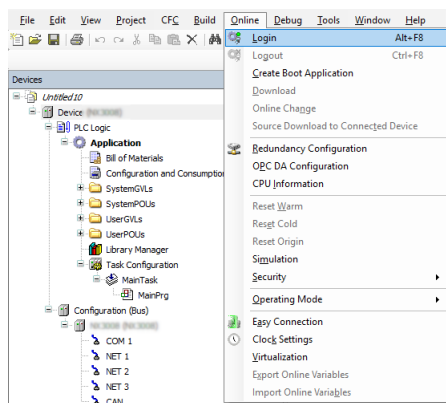


Figura 19: Enviando o Projeto para a UCP

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.

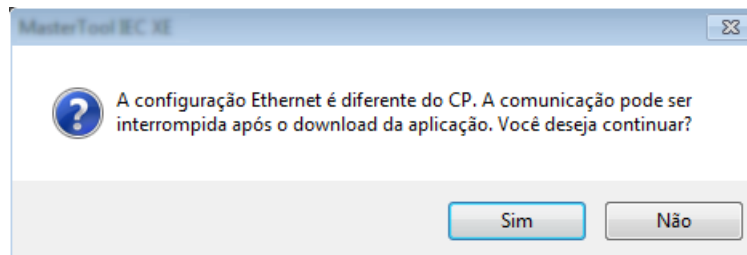


Figura 20: Aviso de configuração IP

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

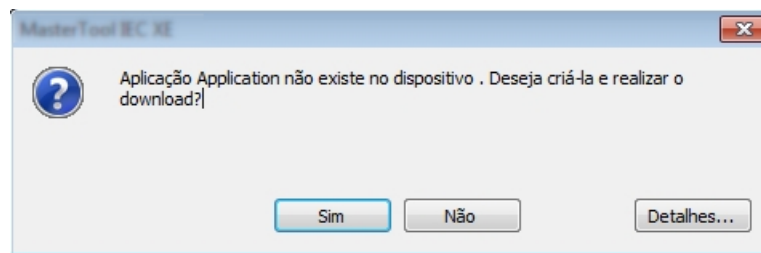


Figura 21: Sem aplicação no dispositivo

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.

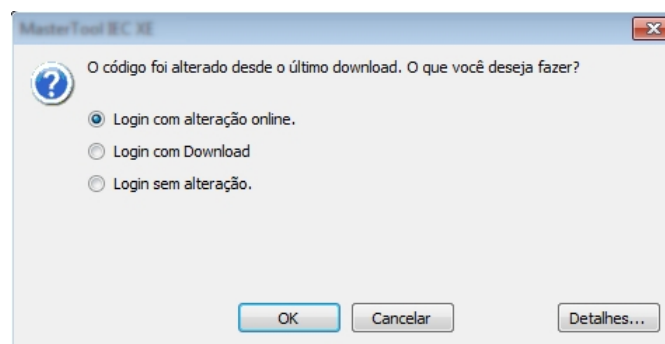


Figura 22: Atualização do Projeto na UCP

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 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.

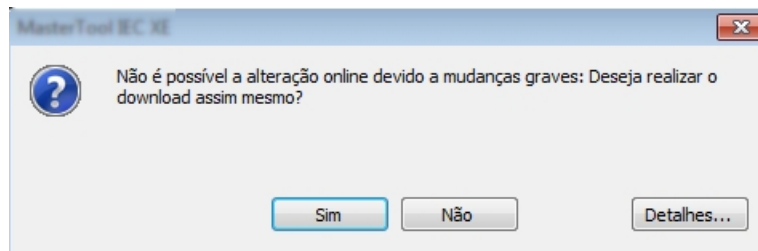


Figura 23: Alteração de Configuração

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

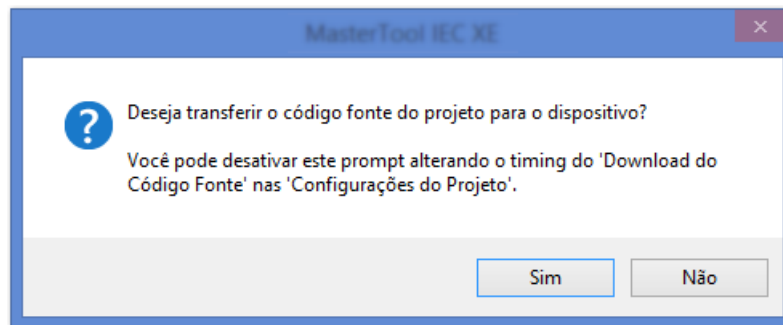


Figura 24: Download de código fonte

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.

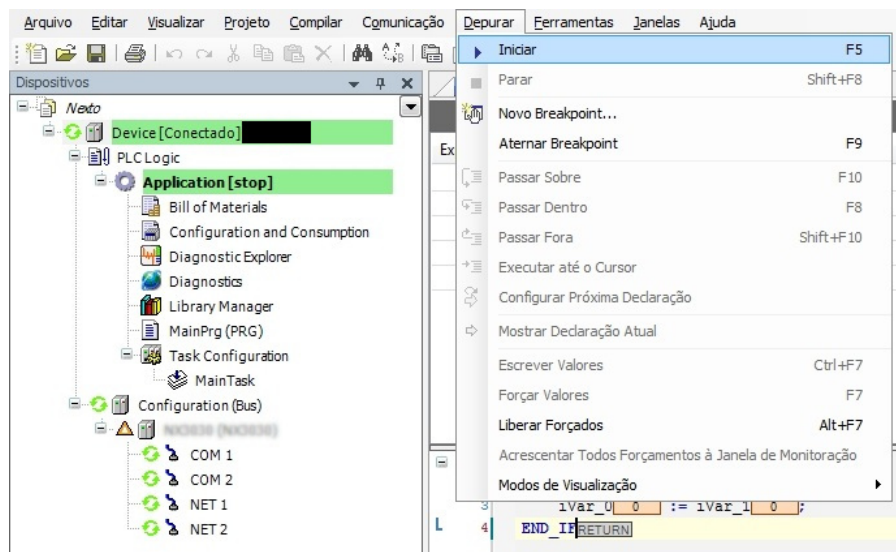


Figura 25: Iniciando a Aplicação

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.

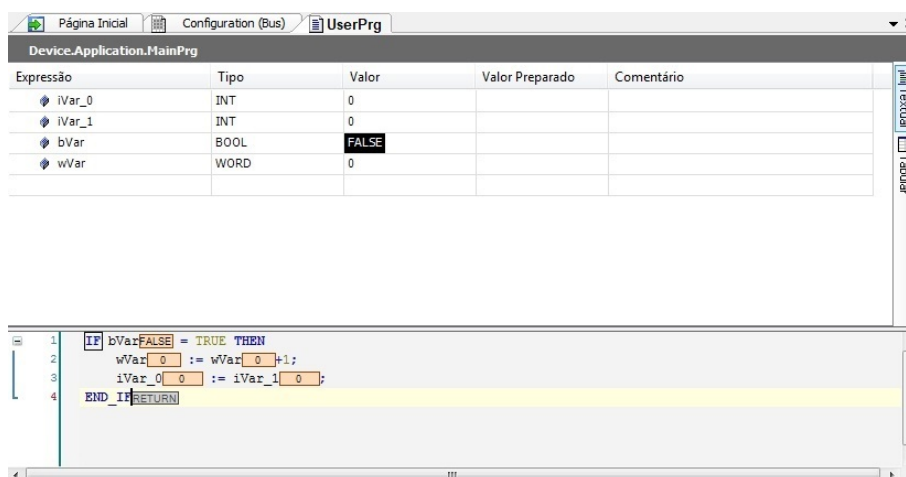


Figura 26: Programa em Execução

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.

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.

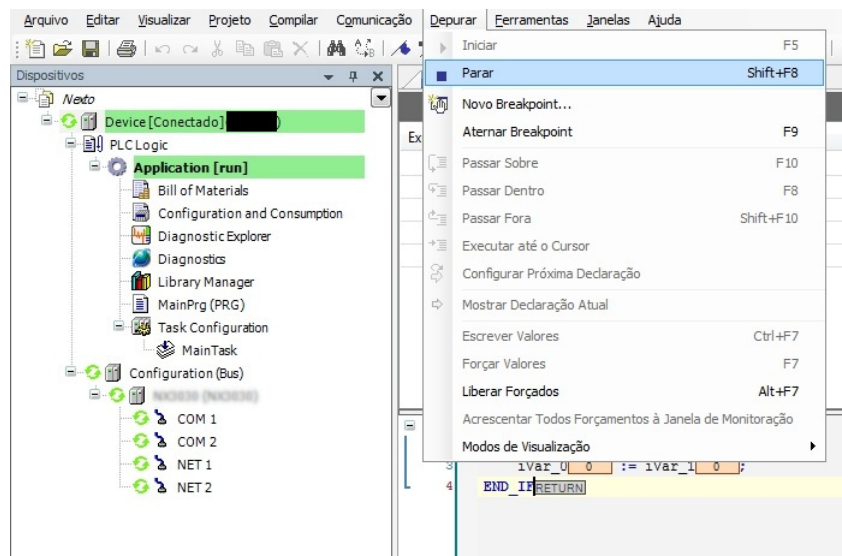


Figura 27: Parando a Aplicação

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.

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.

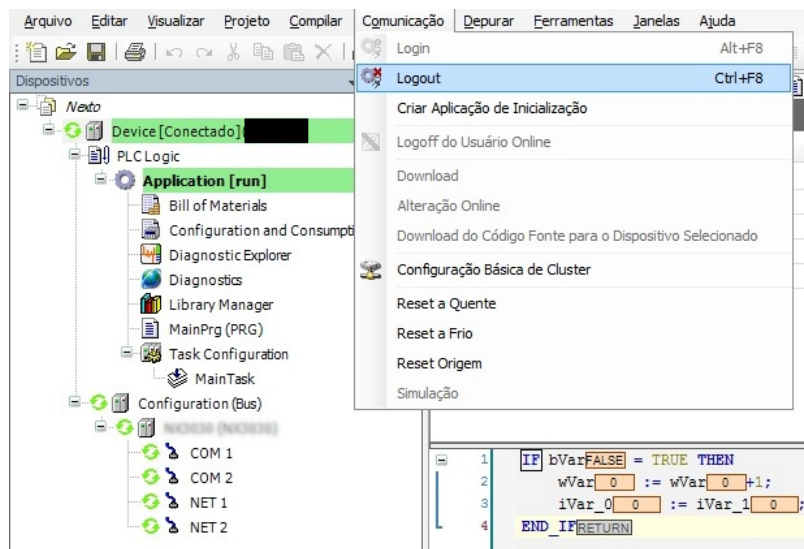


Figura 28: Encerrando a comunicação online com a UCP

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.

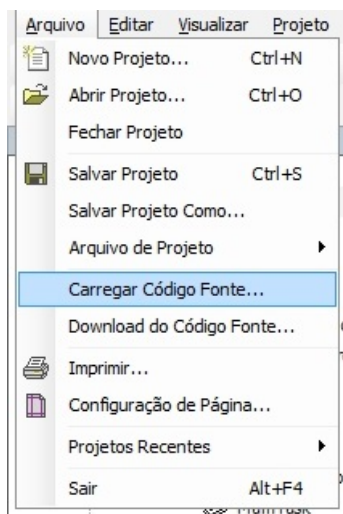


Figura 29: Opção de Upload de Projeto

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

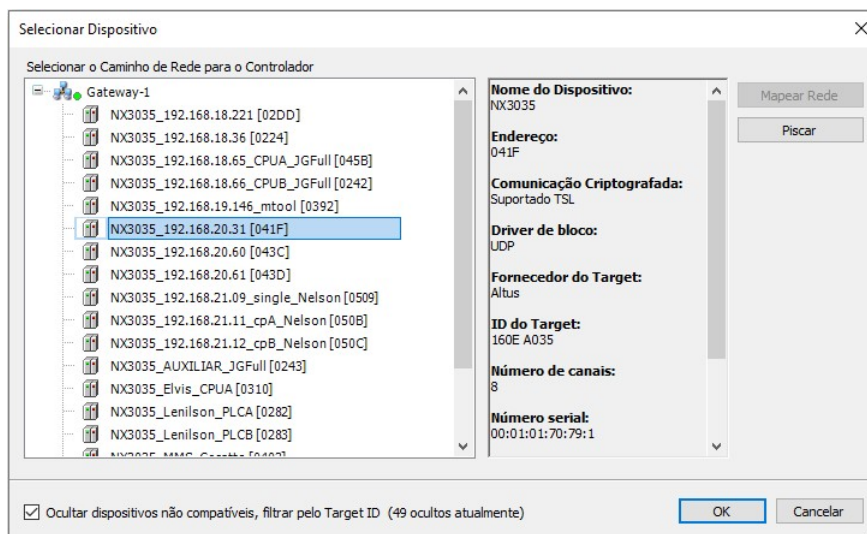


Figura 30: Selecionando a UCP

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.

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.

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 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.

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		reser
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_NX000_1		%QB67251	DG_I

Figura 31: 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 32: 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 supply 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 33: 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:

```
[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.

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 34: 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 35: GVL ReqDiagnostics em Modo Online

5. Configuração

As UCPs da Série Nexto são configuradas e programadas através do software Mastertool. 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

Fornece funções para definir contas dos usuários e configurar os direitos de acesso ao projeto e a UCP. Utilizando o software Mastertool 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, 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.

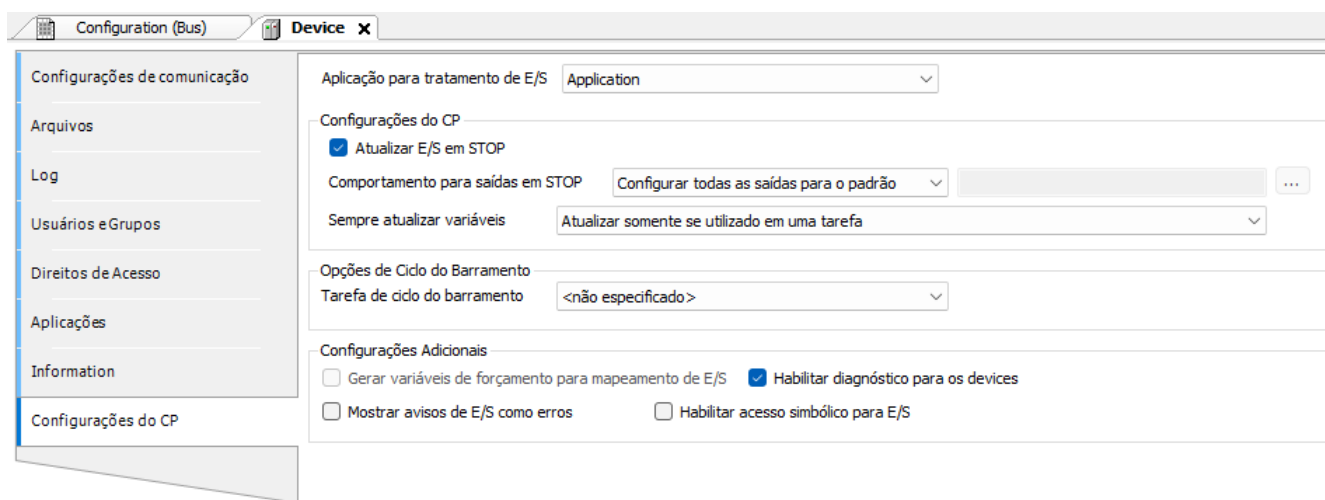


Figura 36: Configurações do CP

Parâmetro	Descrição
Aplicação para tratamento de E/S	Aplicação que é responsável pelo tratamento das entradas e saídas.
Atualizar E/S em STOP	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. FALSE: Os valores dos canais de entrada e saída no modo STOP não são atualizados.

Parâmetro	Descrição
Comportamento para as saídas em STOP	Manuseio dos canais de saída quando o controlador entra no modo STOP: Reter valores: Os valores atuais são retidos. Todas as saídas para o valor padrão: Os valores padrão resultantes do mapeamento de E/S são atribuídos. 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.
Sempre atualizar variáveis	Define globalmente se as variáveis de E/S são ou não atualizadas na tarefa de ciclo do barramento. Esta configuração é efetiva para as variáveis de E/S dos escravos e módulos somente se "desativado" for definido em suas configurações de atualização. Desativado (atualiza somente se usado em uma tarefa): As variáveis de E/S são atualizadas somente se forem usadas em uma tarefa. 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. 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.
Tarefa de ciclo de barramento	Tarefa que controla o ciclo do barramento. Por padrão, a tarefa definida pela descrição do dispositivo é inserida. 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.
Gerar variáveis de forçamento para mapeamento de E/S	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.
Habilitar diagnóstico para os devices	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.
Mostrar avisos de E/S como erros	Os avisos relativos à configuração de E/S são exibidos como erros.
Habilitar acesso simbólico para as E/S	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.

Tabela 38: Configurações do CP

ATENÇÃO

Os produtos Nexto (NX), Nexto Jet (NJ), Nexto XF 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 64682
Tamanho	Tamanho da área de diagnósticos em bytes	1253	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 65535
Tamanho	Tamanho da memória de dados retentivos em bytes	2048	0 a 65535
Área Persistente (%Q)			
Endereço Inicial em %Q	Endereço inicial da memória de dados persistentes (%Q)	12288	0 a 65535
Tamanho	Tamanho da memória de dados persistentes em bytes	2048	0 a 65535
Parâmetros da UCP			
Iniciar Aplicação de Usuário após Reset por Cão-de-Guarda ou Reset de Runtime	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)	- 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
Habilitar terminação interna	Parâmetros da Porta CAN		
	Quando habilitado, utiliza terminação interna na interface CAN	Desmarcado	Marcado: terminação interna na interface CAN habilitada Desmarcado: terminação interna na interface CAN desabilitada

Tabela 39: Configurações da UCP

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 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.

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 40: 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 40) a UCP entra em Modo *Stop* em um tempo de até 2 segundos, quando o LED DG começa a piscar 4x (conforme a Tabela 41). 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 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 40) 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 41). 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 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 40). 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 41). 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 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 40) durante a partida; em caso positivo, a UCP entra em Modo *Stop* e o LED DG começa a piscar 4x (conforme a Tabela 41). 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 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 41), 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 40) 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 41). 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 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 40). 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 41, devido a alguma situação anormal de barramento, conforme a Tabela 40; 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 41 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

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
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 41: 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
```

```
VAR_GLOBAL PERSISTENT RETAIN
    wVariavelSimbolicaPersistenteGlobal_01 : WORD;
END_VAR
VAR_GLOBAL RETAIN
    wVariavelSimbolicaRetentivaGlobal_01 : WORD;
END_VAR
```

5.2.1.3. Parâmetros da Porta CAN

Os parâmetros da porta CAN são relacionados à interface CAN.

Configuração	Descrição	Padrão	Opções
Habilitar terminação interna	Habilita a terminação interna da porta CAN	Desmarcado	- Marcado - Desmarcado

Tabela 42: Parâmetros da Porta CAN

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.

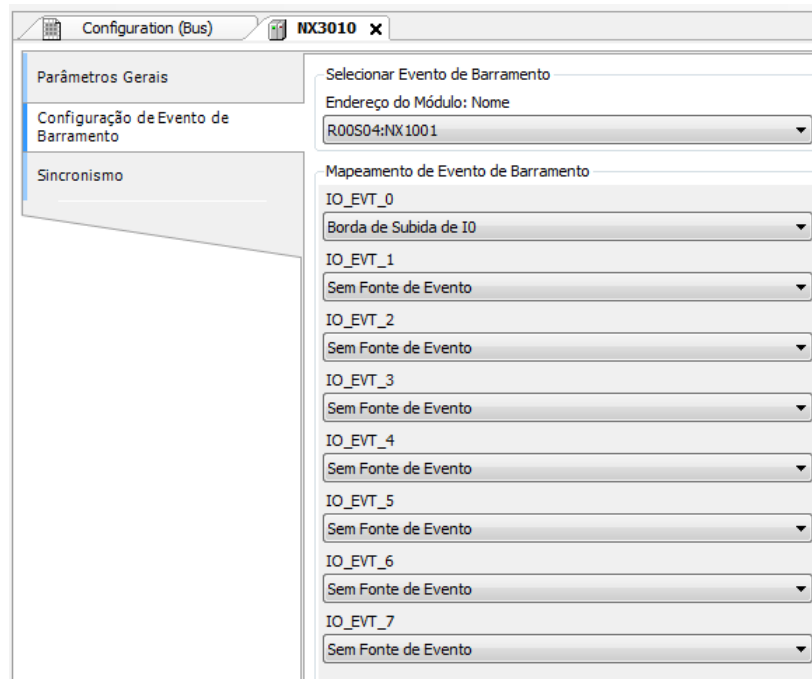


Figura 37: Tela de Configuração de Evento Externo na UCP

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.

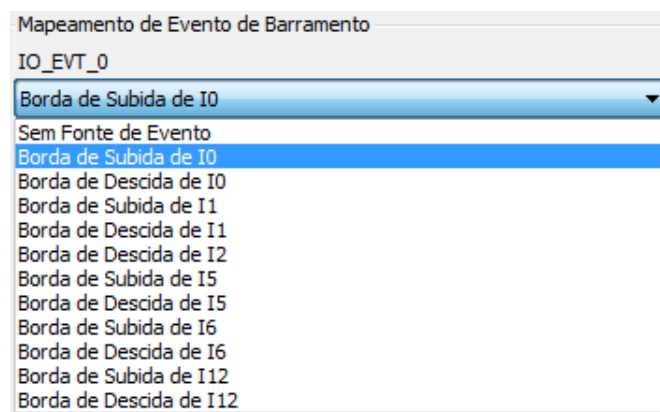


Figura 38: Opções de Origem do Evento Externo no Módulo NX1001

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.

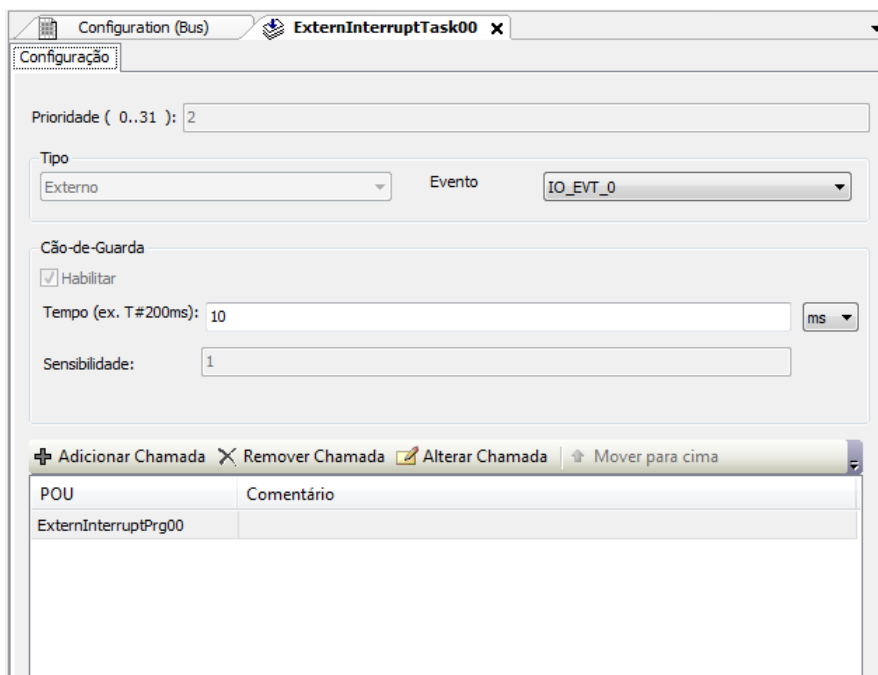


Figura 39: Tela de Configuração da Tarefa ExternInterruptTask00

5.2.3. 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:

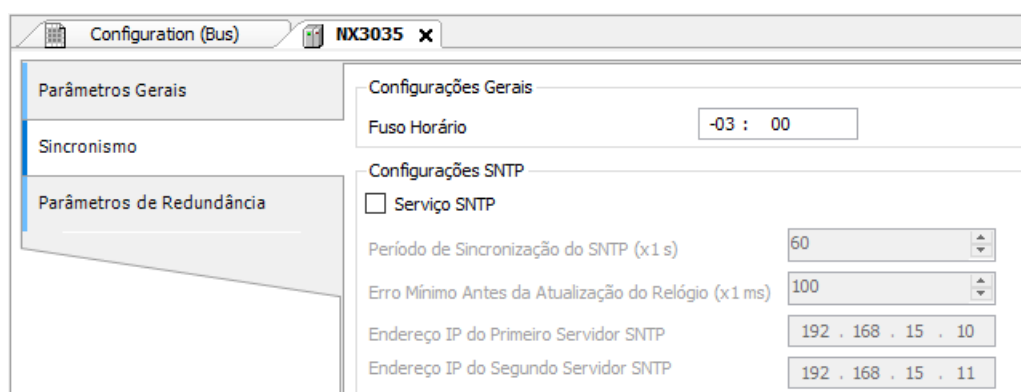


Figura 40: Configuração de SNTP

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 43: 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 (exemplos: protocolo IEC 60870-5-104, função SetDateAndTime), outras utilizam o horário UTC (SNTP). O horário UTC normalmente é utilizado para estampar os eventos (exemplos: DNP3, IEC 60870-5-104, Mastertool Device LOG), enquanto que o horário local é utilizado por outras funcionalidades da UCP (exemplos: 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 a mais prioritária. Assim, quando houver mais fontes habilitadas e o servidor SNTP estiver presente, este será responsável pela sincronização do relógio da UCP e qualquer comando de sincronismo de outras fontes será negado.

5.2.3.1. 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.

5.2.3.2. 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.

5.2.3.3. 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](#).

5.2.4. 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*.

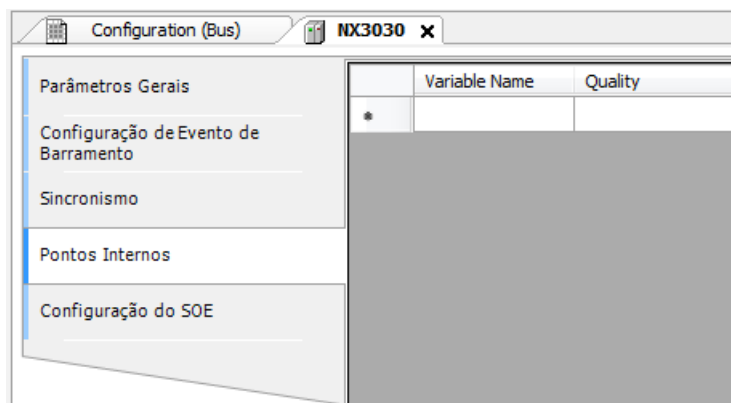


Figura 41: Tela de Configuração dos Pontos Internos

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 44: Configuração dos Pontos Internos

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

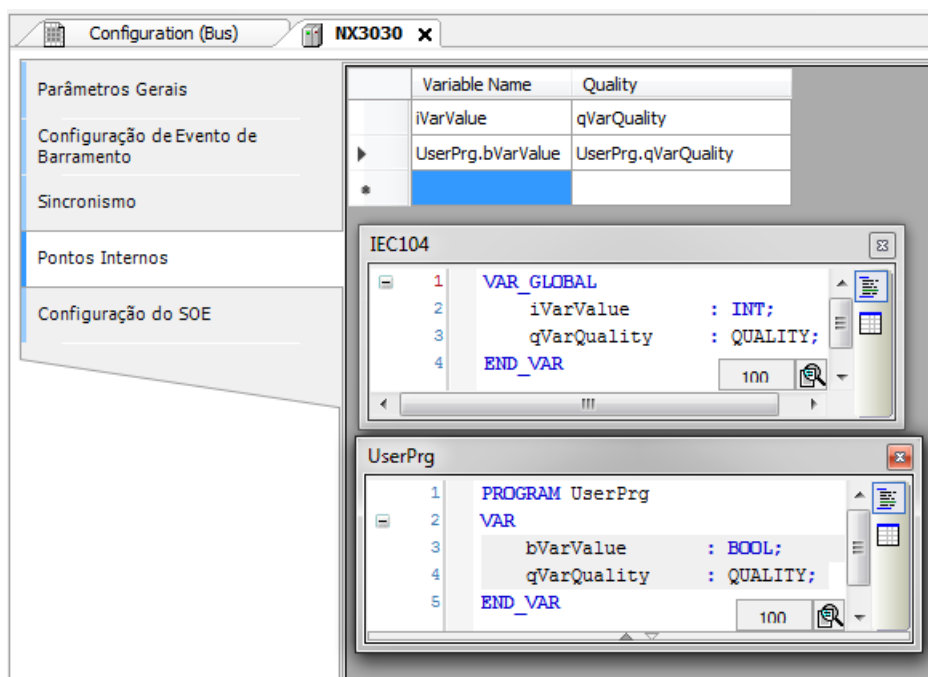


Figura 42: Exemplo de Configuração dos Pontos Internos

5.2.4.1. 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.

5.2.4.1.1. *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.

Bit	Nome	Tipo	Descrição
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
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 45: Estrutura QUALITY

5.2.4.1.2. 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 46: 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
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 47: 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 48: Conversão Interno para IEC 60870-5-104 de Contadores

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

Tabela 49: Qualidade MODBUS

5.2.4.1.4. 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	VALIDITY_QUESTIONABLE	Módulo não operacional, entretanto dados foram lidos/escritos pelo menos uma vez.
bOverRange OU bUnderRange	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 50: Qualidade dos Módulos de E/S

5.2.4.1.5. 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
```

5.2.4.1.6. 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
    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

```

5.2.4.1.7. 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
END_IF

```

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

5.2.4.1.8. 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)
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
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

```

5.2.4.1.9. 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
   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 da Interface Serial

5.3.1. COM 1

A interface de comunicação COM 1 é composta pelos terminais D+ e D-, para o padrão RS-485 half-duplex, permitindo a comunicação ponto a ponto ou em rede nos protocolos abertos MODBUS RTU Escravo ou MODBUS RTU Mestre.

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

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

Configuração	Descrição	Padrão	Opções
Tipo Serial	Configuração do tipo do canal serial.	RS-485	RS-485
Taxa de Transmissão	Velocidade da porta de comunicação serial.	115200	2400, 4800, 9600, 19200, 38400, 57600, 115200 bps
Paridade	Configura a paridade da porta serial.	Sem Paridade	Ímpar Par Sem Paridade
Bits de Dado	Configura o número de bits de dados em cada caractere da comunicação serial.	8	7 e 8
Bits de Parada	Configura os bits de parada da porta serial.	1	1 e 2

Configuração	Descrição	Padrão	Opções
Modo Serial	Configura o modo de operação da porta serial.	Modo Normal	- Modo Estendido: Modo estendido de operação da comunicação serial, no qual são fornecidas informações sobre o frame de dados recebido (ver notas). - Modo Normal: Modo normal de operação da comunicação serial

Tabela 51: Configuração da Interface Serial Padrão RS-485

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 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 BREAK 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 : DWORD): Indica o tempo de silêncio, no intervalo de 0 a 4.294.967.295, usando como base de tempo 1 us. Satura em 4.294.967.295 us, caso o tempo de silêncio seja maior do que 4.294.967.295 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
7	1, 2	SEM PARIDADE, ÍMPAR, PAR
8	1, 2	SEM PARIDADE, ÍMPAR, PAR

Tabela 52: 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
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
Terminação RS-485	Habilita a terminação na interface serial COM1	Habilitado	Habilitado Desabilitado

Tabela 53: Configurações Avançadas da Interface Serial RS-485

5.4. Configuração da Interface CAN (Controller Area Network)

5.4.1. CAN

A interface CAN permite comunicação ponto-a-ponto ou por rede com outros dispositivos que possuem essa interface usando o protocolo de aplicação CANOpen.

Os parâmetros da interface CAN que devem ser configurados para o funcionamento correto da aplicação são descritos abaixo:

Configuração	Descrição	Padrão	Opções
Network	Número de ID da interface CAN	0	0 (fixado)
Taxa de Transmissão	Taxa de transmissão do barramento CAN (kbit/s). Os outros dispositivos devem utilizar a mesma taxa de transmissão.	250	10, 20, 50, 100, 125, 250, 500, 1000

Tabela 54: Configuração CAN

Os parâmetros relacionados ao protocolo CANOpen estão descritos na seção [Configuração de Protocolos](#).

5.5. Configuração das Interfaces Ethernet

A UCP NX3008 possui NET 1, NET 2 e NET 3. 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.

5.5.1. Interfaces Ethernet Integradas

As interfaces são compostas por um conector tipo RJ45 de comunicação no padrão 10BASE-T/100BASE-TX/1000BASE-T. Permitem 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.

5.5.1.1. 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 da porta na rede Ethernet	192.168.15.1	1.0.0.1 a 223.255.255.254
Máscara de Subrede	Máscara de subrede da porta na rede Ethernet	255.255.255.0	128.0.0.0 a 255.255.255.252
Endereço do Gateway	Endereço do Gateway da porta na rede Ethernet	192.168.15.253	0.0.0.0 a 223.255.255.254

Tabela 55: Configurações NET 1

5.5.1.2. NET 2

Configuração	Descrição	Padrão	Opções
Endereço IP	Endereço IP da porta na rede Ethernet	192.168.16.1	1.0.0.1 a 223.255.255.254
Máscara de Subrede	Máscara de subrede da porta na rede Ethernet	255.255.255.0	128.0.0.0 a 255.255.255.252
Endereço do Gateway	Endereço do Gateway da porta na rede Ethernet	192.168.16.253	0.0.0.0 a 223.255.255.254

Tabela 56: Configurações NET 2

5.5.1.3. NET 3

Configuração	Descrição	Padrão	Opções
Endereço IP	Endereço IP da porta na rede Ethernet	192.168.17.1	1.0.0.1 a 223.255.255.254
Máscara de Subrede	Máscara de subrede da porta na rede Ethernet	255.255.255.0	128.0.0.0 a 255.255.255.252
Endereço do Gateway	Endereço do Gateway da porta na rede Ethernet	192.168.17.253	0.0.0.0 a 223.255.255.254

Tabela 57: Configurações NET 3

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.

5.5.1.4. Modos de Operação das Interfaces Ethernet Integradas

Os canais Ethernet da UCP podem ser configurados em três modos distintos de funcionamento: Modo Simples, Modo Redundante ou Modo Switch. Diferente de outras CPUs da Série Nexto, a NX3008 possui três portas Ethernet (NET 1, NET 2 e NET 3).

Por padrão, a interface NET 1 não pode ser configurada em nenhum outro modo além do simples. Já a NET 2 possui as três opções de configuração e, a NET 3, depende das configurações da NET 2 para ser definida.

Quando a NET 2 é configurada como Modo Redundante ou Modo Switch, as interfaces NET 2 e NET 3 passam a operar como um par (par redundante/ switch) e toda a informação passa a ser armazenada na NET par, neste caso na NET 2. E, nesta arquitetura, a NET 3 tem todos os seus campos de configuração desabilitados automaticamente.

Caso a NET 2 seja desabilitada ou configurada como Modo Simples, a interface NET 3, caso habilitada, opera como Modo Simples.

5.5.1.4.1. Modo Simples

Neste modo, a interface opera como uma porta Ethernet independente, não havendo qualquer relação com a interface subsequente.

☒ Habilitar Interface

☐ Obter um endereço IP automaticamente (Modo Simples)

Parâmetros da Porta Ethernet

Endereço IP: 192 . 168 . 15 . 1

Máscara de Subrede: 255 . 255 . 255 . 0

Endereço do Gateway: 192 . 168 . 15 . 253

Modo

☒ Simples ☐ NIC Teaming ☐ Switch

Figura 43: Configurações Avançadas das Interfaces Ethernet - Modo Simples

5.5.1.4.2. Modo NIC Teaming

Neste modo, a interface forma um par redundante com a interface subsequente, operando em um esquema do tipo ativa/-reserva. Um par de *NIC Teaming* possui um único endereço IP, associado à interface que estiver ativa no momento. Desta forma, um cliente como um SCADA ou Mastertool, conectado a um servidor na UCP, não precisa se preocupar em trocar o endereço IP caso haja falha em alguma das portas do par. Além disso, cada uma das interfaces que formam um par redundante de comunicação, possuem diagnósticos separados, com o intuito de facilitar a depuração de falhas que eventualmente possam surgir.

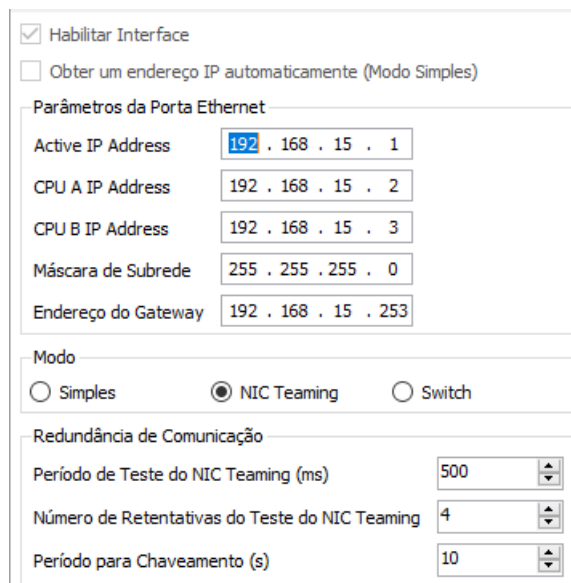


Figura 44: Configurações Avançadas das Interfaces Ethernet - Modo NIC Teaming

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

- **Período de Teste do NIC Teaming (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 do NIC Teaming:** 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 do NIC Teaming* 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. Esse 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.

5.5.1.4.3. Modo Switch

Neste modo, a interface forma um par com a interface subsequente operando como um switch Ethernet de duas portas, permitindo então a comunicação por ambas às portas. As opções de configuração de proteção de loop no modo switch, estão descritas nas próximas subseções.

1. Disabled

Nesse modo, as duas portas subsequentes da UCP atuam como um switch simples não gerenciável, sem qualquer proteção de loop.

☒ Habilitar Interface
☐ Obter um endereço IP automaticamente (Modo Simples)

Parâmetros da Porta Ethernet

Endereço IP: 192 . 168 . 15 . 1

Máscara de Subrede: 255 . 255 . 255 . 0

Endereço do Gateway: 192 . 168 . 15 . 253

Modo

☐ Simples ☐ NIC Teaming ☒ Switch

Configuração do Modo Switch

Loop Protection Mode: Disabled

Figura 45: Configurações Avançadas das Interfaces Ethernet - Modo Switch - Loop Protection Disabled

A figura seguinte mostra um exemplo de topologia envolvendo duas portas com esta configuração. As três UCPs foram conectadas diretamente sem a necessidade de um switch externo.

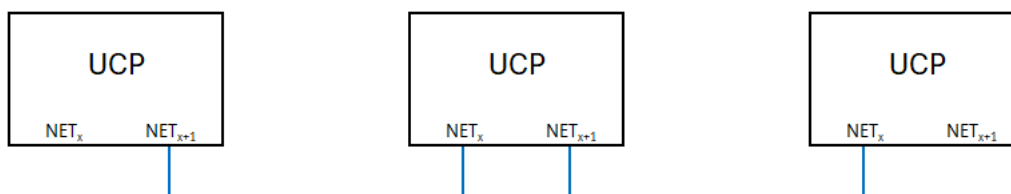


Figura 46: Exemplo de Topologia com o Modo Switch - Loop Protection Disabled

2. MRP

O modo de proteção MRP permite configurar topologias em anel suportando o protocolo MRP (Media Redundancy Protocol).

☒ Habilitar Interface
☐ Obter um endereço IP automaticamente (Modo Simples)

Parâmetros da Porta Ethernet

Endereço IP: 192 . 168 . 15 . 1

Máscara de Subrede: 255 . 255 . 255 . 0

Endereço do Gateway: 192 . 168 . 15 . 253

Modo

☐ Simples ☐ NIC Teaming ☒ Switch

Configuração do Modo Switch

Loop Protection Mode: MRP

Figura 47: Configurações Avançadas das Interfaces Ethernet - Modo Switch - MRP

Esta configuração só é possível caso haja, abaixo das duas portas subsequentes, uma instância do protocolo PROFINET com porta dupla (Ethernet / PN_Controller_Dual_port). Neste anel MRP:

- A UCP deve assumir o papel de *Profinet Controller*, e os demais dispositivos devem assumir o papel de *PROFINET device*.
- A UCP pode ser configurada como *MRP manager* ou *MRP client*.

Para informações detalhadas sobre a configuração e o funcionamento do protocolo MRP consulte o Nexto Series PRO-FINET Manual - MU214621.

A figura seguinte mostra um exemplo de anel MRP envolvendo duas portas subsequentes de uma UCP. Topologias em anel oferecem alta disponibilidade, com continuidade da comunicação no caso de rompimento de uma das conexões entre dispositivos. Além disso, em caso de falha total em um Profinet Device, a UCP pode manter a comunicação com os demais Profinet Devices.

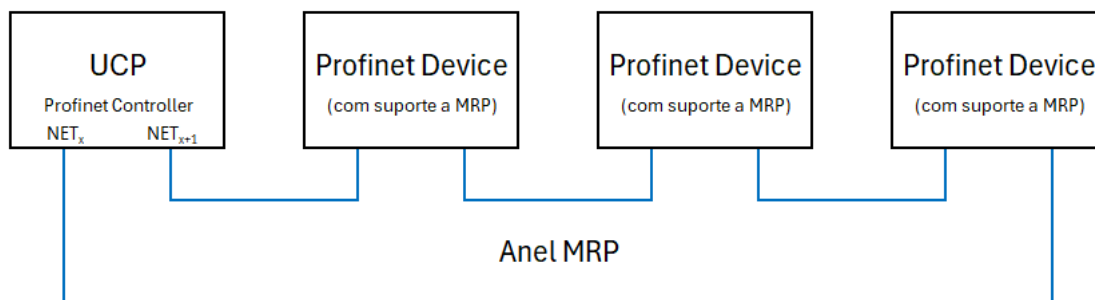


Figura 48: Exemplo de Topologia em Anel com o Modo Switch - MRP

3. RSTP

Neste modo, o par de interfaces de rede utiliza o protocolo *Rapid Spanning-Tree protocol* (RSTP) para evitar loops de rede e permitir redundância de comunicação entre os dispositivos usando topologias como “mesh” ou anel. Contudo, se for recebido um frame STP pelas suas portas, o protocolo RSTP é automaticamente rebaixado para *Spanning-Tree Protocol* (STP), mantendo assim a comunicação com a rede. Deve-se observar que o protocolo STP possui tempos de resposta mais lentos do que o protocolo RSTP.

☒ Habilitar Interface

☐ Obter um endereço IP automaticamente (Modo Simples)

Parâmetros da Porta Ethernet

Endereço IP: 192 . 168 . 15 . 1

Máscara de Subrede: 255 . 255 . 255 . 0

Endereço do Gateway: 192 . 168 . 15 . 253

Modo

☐ Simples ☐ NIC Teaming ☒ Switch

Configuração do Modo Switch

Loop Protection Mode: RSTP

Bridge Priority: 32768

Max Age: 20

Hello Time (s): 2

Forward Delay (s): 15

Tx Hold Count: 2

Ports Path Cost: 200000

☒ Auto Ports Path Cost

Figura 49: Configurações Avançadas das Interfaces Ethernet - Modo Switch - Loop Protection RSTP

Para mais informações sobre o uso e configuração do protocolo RSTP deve-se consultar a nota de aplicação NAP169 - RSTP in Nexto CPUs.

A figura seguinte mostra um exemplo de uso do protocolo RSTP em duas portas subsequentes de algumas UCPs em uma topologia do tipo “mesh”. Os outros switches nesta rede também devem suportar RSTP ou STP. Caso haja algum switch STP na rede, a UCP rebaixa seu protocolo de RSTP para STP, resultando em tempos de recuperação de falha mais altos.

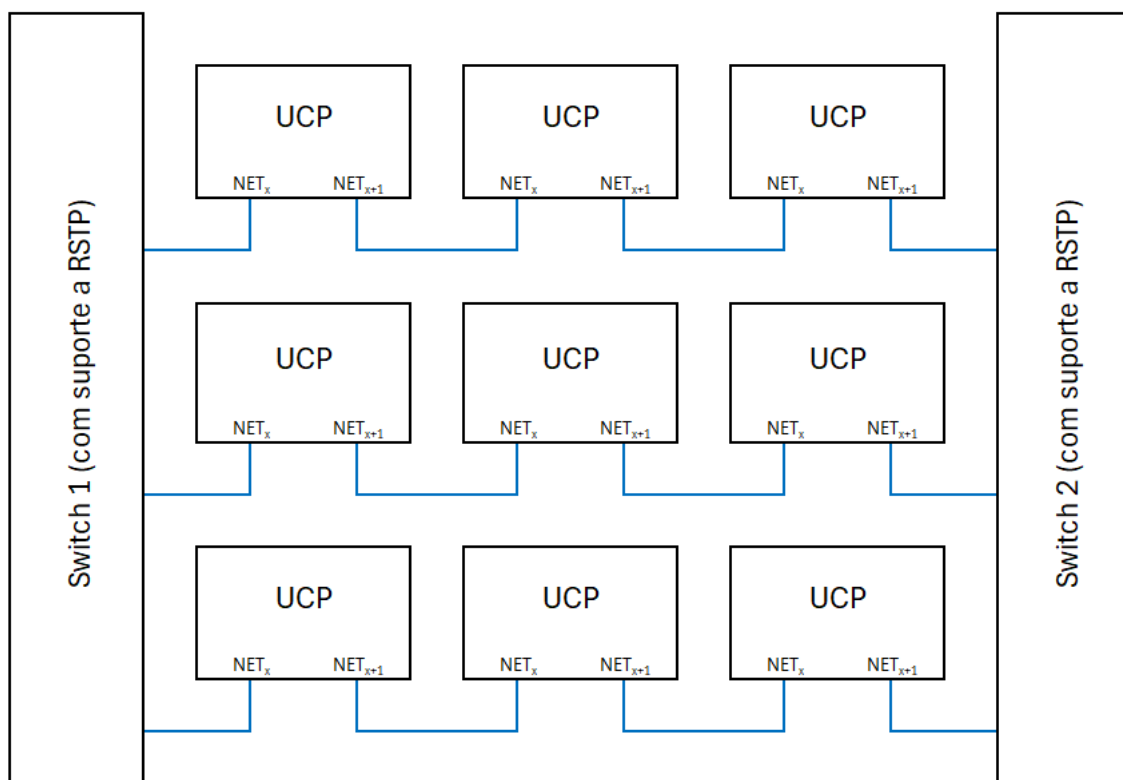


Figura 50: Exemplo de Topologia Mesh com o Modo Switch - RSTP

A figura seguinte mostra um exemplo de uso do protocolo RSTP com duas portas subsequentes de algumas UCPs em uma topologia do tipo anel. Diferente do anel com modo de proteção MRP descrito anteriormente, neste caso não é necessário utilizar um controlador PROFINET nas portas subsequentes.

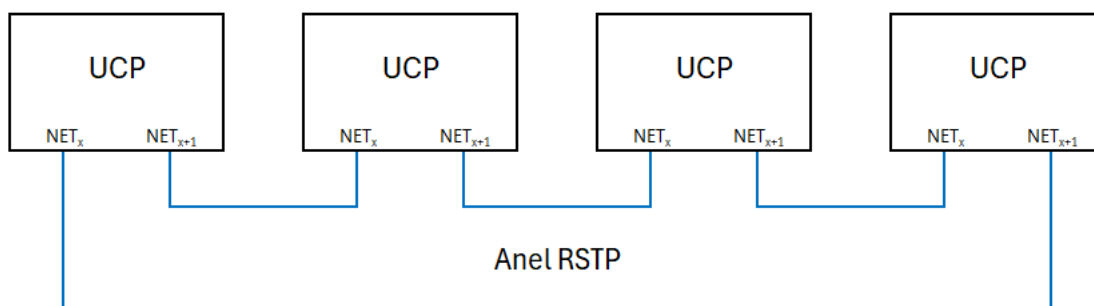


Figura 51: Exemplo de Topologia em Anel com o Modo Switch - RSTP

5.5.2. Interface Ethernet Remota NX5000

5.5.2.1. 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 da porta na rede Ethernet	192.168.xx.68	1.0.0.1 a 223.255.255.254
Máscara de Subrede	Máscara de subrede da porta na rede Ethernet	255.255.255.0	128.0.0.0 a 255.255.255.252
Endereço do Gateway	Endereço do Gateway da porta na rede Ethernet	192.168.xx.253	0.0.0.0 a 223.255.255.254

Tabela 58: Configurações NET 1 Remota NX5000

5.5.2.2. 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 integradas da UCP não sejam suficientes.

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

5.5.2.2.1. 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 *NIC Teaming* 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 *NIC Teaming* do módulo à esquerda, ocorre a separação dos módulos, que voltam a se comportar como módulos individuais sem redundância.

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

Figura 52: Advanced Configuration of Remote Ethernet Interface - NX5000

5.5.3. Portas TCP/UDP Reservadas

As seguintes portas TCP/UDP das interfaces Ethernet, tanto integradas 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	1217*	1740:1743
SQL Server	1433	-
MQTT	1883* / 8883*	-
EtherNet/IP	44818	2222
IEC 60870-5-104	2404*	-
IEC 61850	102*	-
DNP3	20000* / 20005*	-
OPC UA	4840	-
WEBVISU	8080	-
CODESYS ARTI	11740	-
PROFINET	-	34964
Portainer Docker	9000	-
SysLog	-	514
LibHART	1234	-

Tabela 59: Portas TCP/UDP reservadas

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

5.5.4. Roteamento de Rede

Para portas Ethernet integradas, o sistema operacional da UCP possui um mecanismo de roteamento de rede que permite a comunicação entre diferentes redes:

1. Protocolos Cliente, quando configurados para se comunicar com um dispositivo Servidor localizado em uma rede diferente, resolverão o roteamento pela seguinte prioridade:
 - VPN (se este recurso estiver habilitado e a rede da interface NET do protocolo Cliente configurada como *Rede Privada* nas configurações do Servidor VPN)
 - Dispositivos USB (Modem, Ethernet ou Wi-Fi)
 - Outra interface NET integrada configurada para a mesma rede do endereço de destino (neste caso, a comunicação será roteada internamente)

Se nenhuma dessas condições ocorrer, o sistema operacional encaminhará a comunicação para o Gateway Padrão configurado nas interfaces NET, iniciando pelo configurado na NET1 e passando pelos seguintes se a comunicação falhar.

2. Protocolos Servidor aceitarão conexões apenas na interface onde estão instanciados, exceto para o MODBUS, onde um novo parâmetro (introduzido no Mastertool 3.75) chamado *Permitir Conexões de Qualquer Interface* torna capaz receber conexões de entrada de outras interfaces, como um Modem USB ou VPN.

Para portas Ethernet remotas (módulo NX5000), o roteamento é mais restrito e não interage com as portas integradas:

1. Protocolos Cliente, quando configurados para se comunicar com um dispositivo servidor localizado em uma rede diferente, encaminharão a comunicação para o Gateway Padrão configurado em sua interface de rede.
2. Protocolos Servidor aceitarão conexões apenas de sua interface de rede.

5.6. 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:

	NX3008
Pontos Mapeados	20480
Mapeamentos (Por Instância / Total)	5120 / 20480
Requisições	512
NETs – Instâncias Cliente ou Servidor (Por NET / Total)	4 / 16
COM (n) – Instâncias Mestre ou Escravo	1
Centros de Controle	3

Tabela 60: Limites dos Protocolos por UCP

Notas:

Pontos Mapeados: É o número máximo de pontos mapeados que a UCP suporta. Cada mapeamento pode conter um ou mais pontos mapeados, dependendo do tamanho do dado. Isso varia conforme o uso de variáveis simples ou de variáveis do tipo ARRAY. Cada variável simples, assim como cada índice de um ARRAY, é contabilizado como um ponto mapeado, mesmo que ocupe mais de um endereço no driver. Por exemplo, uma variável simples do tipo DWORD mapeada no protocolo MODBUS será contabilizada como um único ponto, ainda que ocupe dois endereços/registros consecutivos no driver.

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 61 e na Tabela 62, 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 61: 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 60.

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 Clientes MODBUS. 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 62: 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 Clientes MODBUS. 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 63: 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 60).

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.

5.6.1. 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	✗	✗	✗	✓	✓	✓
IEC 60870-5-104	Server	✗	✗	✗	✓	✗	✓
EtherCAT	Master	✓	✓	✗	✓	NA	✓
OPC DA	Server	✓	✓	✓	✓	✗	✓
OPC UA	Server	✓	✓	✓	✓	NA	✓
SNTP	Client	✓	✓	✓	✓	✓	✓
HTTP	Server	✓	✓	✓	✓	✓	✓
SNMP	Agent	✓	✓	✓	✓	✓	✓
EtherNet/IP	Scanner	✓	✓	✓	✓	NA	✗
	Adapter	✓	✓	✓	✓	NA	✗

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
PROFINET	Controller	✗	✓	✗	✓	NA	✓
MQTT	Client	✗	✗	✗	✓	✓	✗
SparkPlugB	Client	✗	✗	✗	✓	✓	✗
OpenVPN	Server	✓	✓	✓	✓	✓	✓
	Client	✓	✓	✓	✓	✓	✓
FTP	Server	✓	✓	✓	✓	✓	✓
Syslog	Client	✓	✓	✓	✓	✓	✓

Tabela 64: 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*”.

Testes realizados com o valor padrão definido nas *Configurações do CP* (ver tabela [Configurações do CP](#)), estando marcada a opção *Atualizar E/S em STOP* e configurado a opção *Configurar todas as saídas para o padrão*.

5.6.2. 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](#).

5.6.3. 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 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:

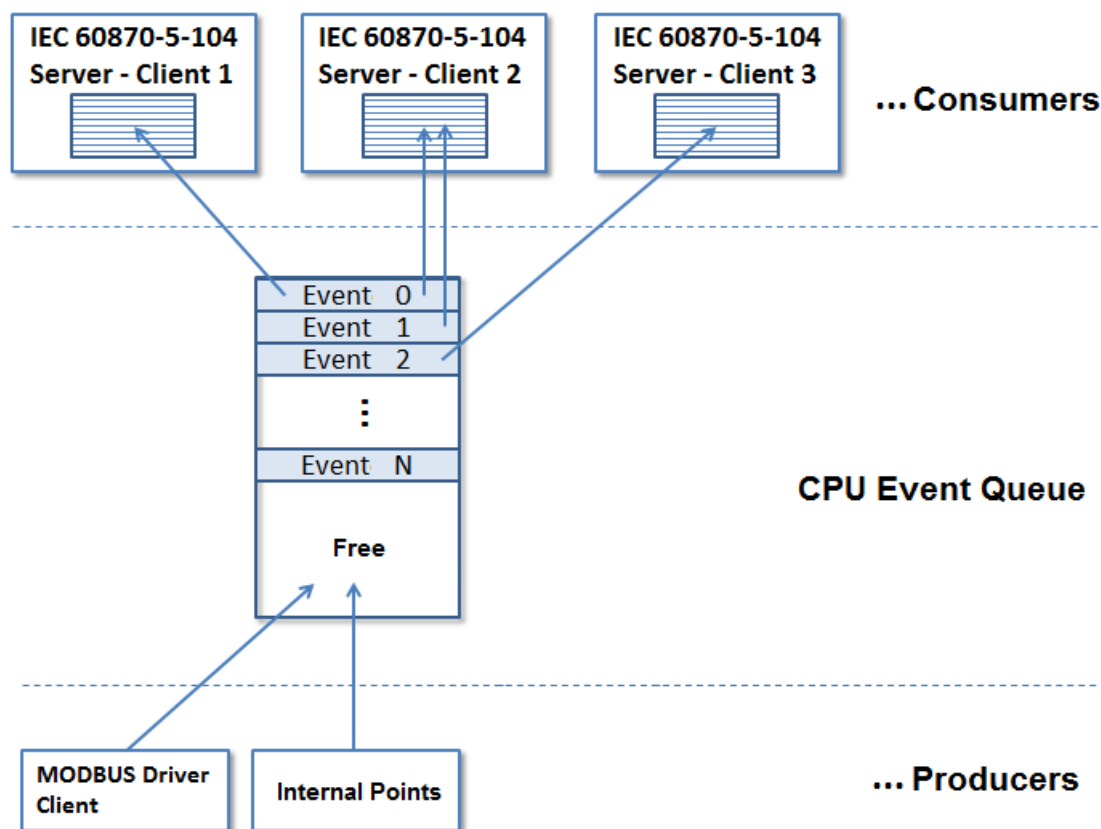


Figura 53: Fila de Eventos da UCP

5.6.3.1. 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.

5.6.3.2. 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.

5.6.3.2.1. 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)

5.6.3.3. 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

5.6.4. 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 65: 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 66: 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 67: 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 69
sValue	STRUCT	Valor recebido em um select, quando recebido um comando de seleção com valor. Os parâmetros desta estrutura estão descritos na Tabela 72

Tabela 68: Parâmetros sSelectParameters

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

Tabela 69: Parâmetros sSelectConfig

Parâmetro	Tipo	Descrição
sOperateConfig	STRUCT	Configuração do comando de seleção recebido. Os parâmetros desta estrutura estão descritos na Tabela 71
sValue	STRUCT	Campo de valor referente ao comando de operação recebido. Os parâmetros desta estrutura estão descritos na Tabela 72

Tabela 70: 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 71: 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 73 a Tabela 77
sDoublePoint	STRUCT	
sIntegerStatus	STRUCT	
sEnumeratedStatus	STRUCT	
sAnalogueValue	STRUCT	

Tabela 72: 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 78.

Tabela 73: 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 78.

Tabela 74: Parâmetros sDoublePoint

Parâmetro	Tipo	Descrição
diValue	DINT	Valor de operação do ponto.

Tabela 75: Parâmetros sIntegerStatus

Parâmetro	Tipo	Descrição
dwValue	DWORD	Valor de operação do ponto.

Tabela 76: 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 77: 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 78: 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*.

5.6.5. MODBUS RTU Mestre

Este protocolo está disponível para as UCPs da Série Nexto nos seus canais seriais. Ao selecionar esta opção no Mastertool, 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 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 da Interface Serial](#).

5.6.5.1. 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 54.
- 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 55.
- Adicionar e configurar os mapeamentos MODBUS na aba Mapeamentos conforme Figura 56, 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 57, 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.

5.6.5.1.1. 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:

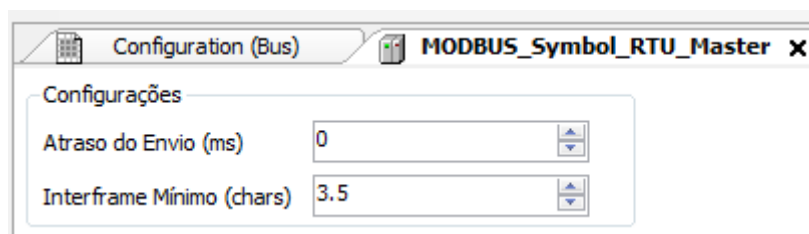


Figura 54: Tela de Configuração Parâmetros Gerais MODBUS RTU Mestre

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 79: 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:

T_DIAG_MODBUS_RTU_MASTER_1.*	Tamanho	Descrição
Bits de diagnóstico:		
tDiag.bRunning	BIT	O mestre está em execução.
tDiag.bNotRunning	BIT	O mestre não está em execução (ver bit: bInterruptedByCommand).
tDiag.bInterruptedByCommand	BIT	O bit bNotRunning foi habilitado, pois o mestre foi interrompido pelo usuário através de bits de comando.
tDiag.bConfigFailure	BIT	Falha na configuração.
tDiag.bModuleFailure	BIT	Não implementado.
Códigos de erro:		
eErrorCode	SERIAL_STATUS (BYTE)	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 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:		
tCommand.bStop	BIT	Parar o mestre.

T_DIAG_MODBUS_RTU_MASTER_1.*	Tamanho	Descrição
tCommand.bRestart	BIT	Reiniciar o mestre.
tCommand.bResetCounter	BIT	Reiniciar as estatísticas dos diagnósticos (contadores).
Estatísticas de comunicação:		
tStat.wTXRequests	WORD	Contador de requisições transmitidas pelo mestre (0 a 65535).
tStat.wRXNormalResponses	WORD	Contador de respostas normais recebidas pelo mestre (0 a 65535).
tStat.wRXExceptionResponses	WORD	Contador de respostas com códigos de exceção recebidas pelo mestre (0 a 65535).
tStat.wRXIllegalResponses	WORD	Contador de respostas ilegais recebidas pelo mestre – sintaxe inválida, número insuficiente de bytes recebidos, CRC inválido – (0 a 65535).
tStat.wRXOverrunErrors	WORD	Contador de erros de overrun durante a recepção – UART FIFO ou fila RX – (0 a 65535).
tStat.wRXIncompleteFrames	WORD	Contador de respostas com erro de construção, paridade ou falha durante a recepção (0 a 65535).
tStat.wCTSTimeOutErrors	WORD	Contador de erro de time-out no CTS, utilizando o handshake RTS/CTS, durante a transmissão (0 a 65535).

Tabela 80: 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.

5.6.5.1.2. Configuração dos Dispositivos – Configuração por Mapeamento Simbólico

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

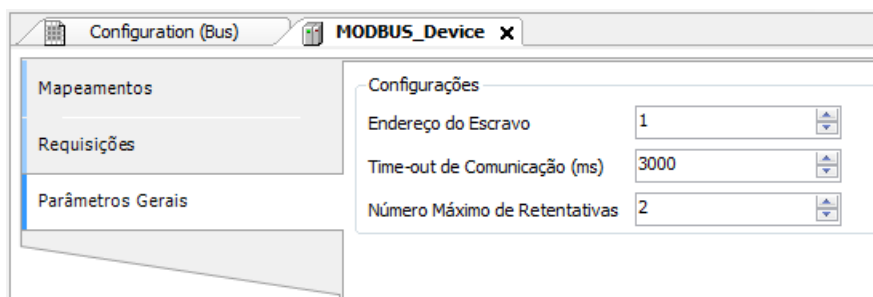


Figura 55: Tela de Configurações dos Parâmetros Gerais do Dispositivo

Configuração	Descrição	Padrão	Opções
Endereço do escravo	Endereço do escravo MODBUS	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 81: 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.

5.6.5.1.3. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico

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

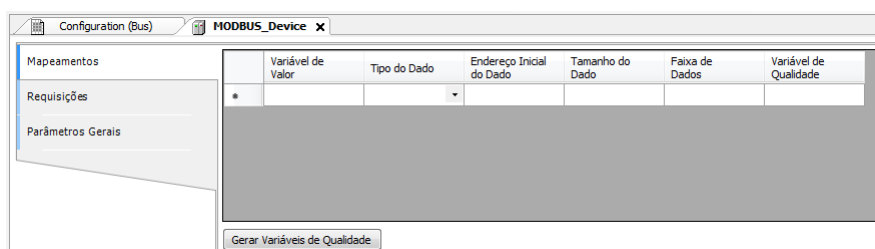


Figura 56: Tela de Mapeamentos de Dados MODBUS

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 82: 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 83: 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.

5.6.5.1.4. Configuração das Requisições – Configuração por Mapeamento Simbólico

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

Figura 57: 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 Mascaramada 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

Configuração	Descrição	Padrão	Opções
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 84: 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 85: 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 86.

T_DIAG_MODBUS_RTU_MAPPING_1.*	Tamanho	Descrição
Bits de estado da comunicação:		
byStatus.bCommIdle	BIT	Comunicação inativa (aguardando ser constada).
byStatus.bCommExecuting	BIT	Comunicação ativa.
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.
byStatus.bCommDisabled	BIT	Comunicação desabilitada. O bit bCommIdle é reiniciado nessa condição.
byStatus.bCommOk	BIT	Comunicação finalizada anteriormente foi realizada com sucesso.
byStatus.bCommError	BIT	Comunicação finalizada anteriormente teve um erro. Verificar código de erro.
Último código de erro (habilitado quando o bCommError = verdadeiro):		
eLastErrorCode	MASTER_ERROR_CODE (BYTE)	Informa a possível causa do último erro ocorrido na relação MODBUS. Consulte a Tabela 109 para detalhes das possibilidades.
Último código de exceção recebido pelo mestre:		
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:		
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.
wCommErrorCounter	WORD	Contador de comunicações finalizadas com erros. Quando o valor 65535 é atingido, o contador retorna a zero.

Tabela 86: Diagnósticos das relações MODBUS RTU

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).
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 87: 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.

5.6.5.2. 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.

5.6.5.2.1. 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:

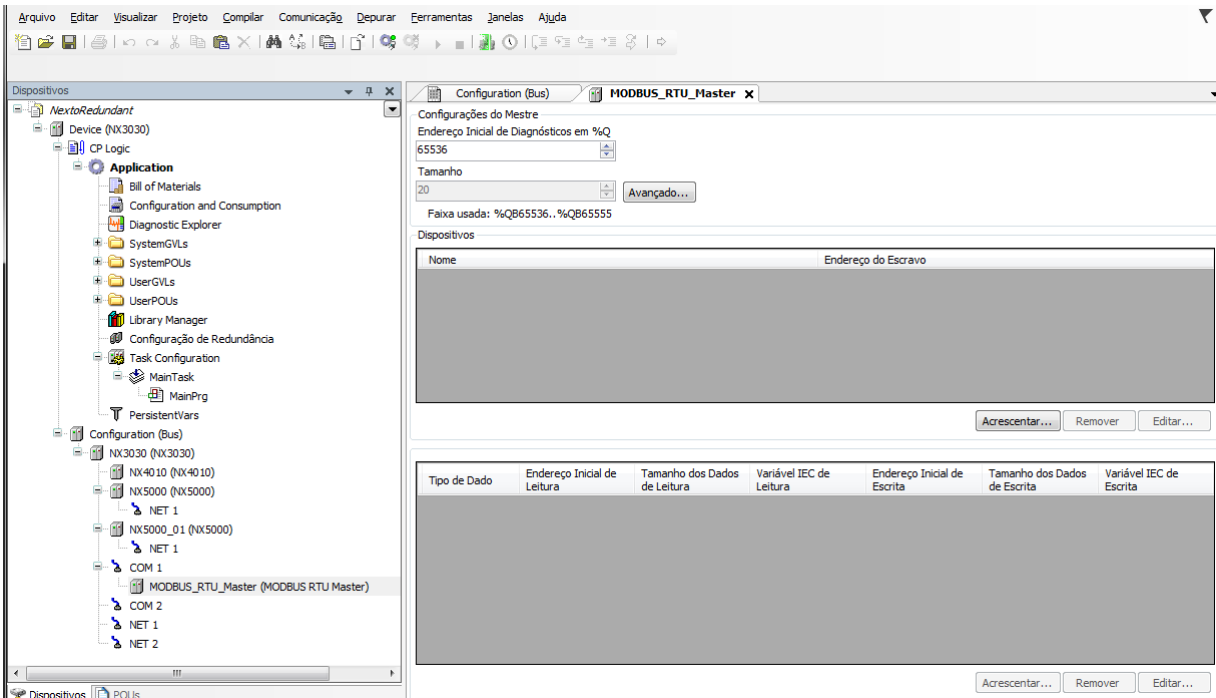


Figura 58: Tela de Configuração MODBUS RTU Mestre

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 88: 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 alocue 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 [80](#).

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](#).

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

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

Figura 59: 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 89: 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 alocue 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](#).

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

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

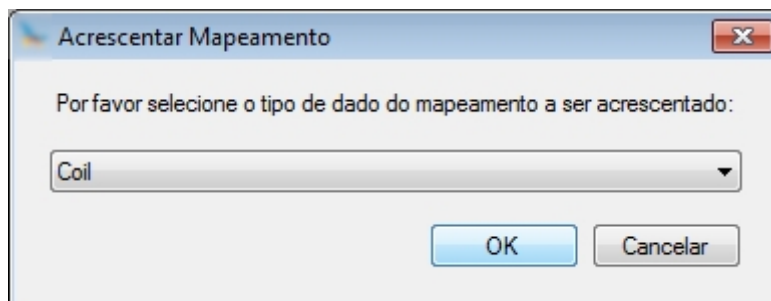


Figura 60: Tipo de Dado MODBUS

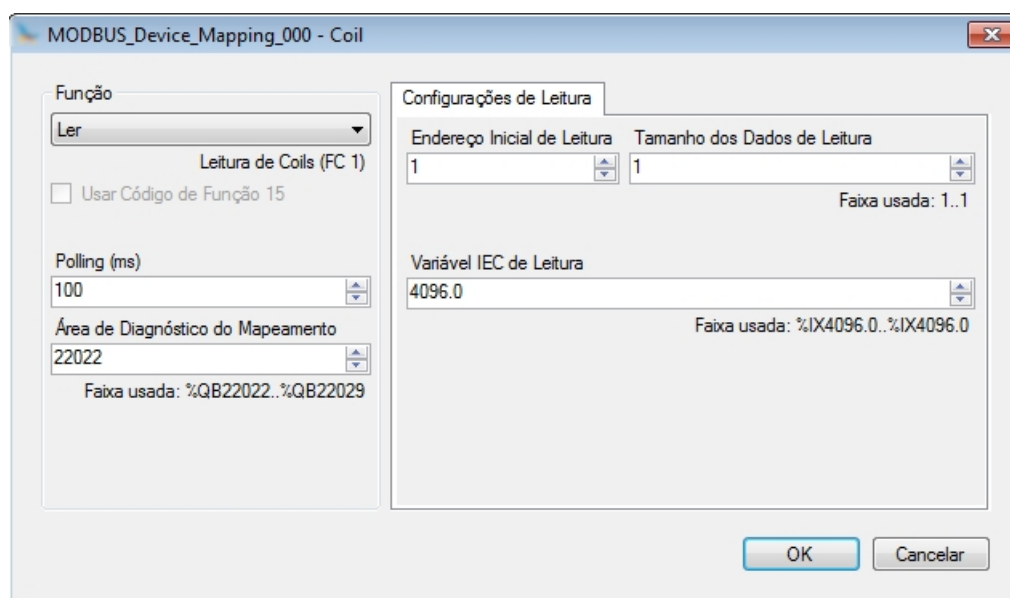


Figura 61: Função MODBUS

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
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

Configuração	Descrição	Padrão	Opções
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 90: Mapeamentos do Dispositivo

Notas:

Função: Os tipos de dados disponíveis estão detalhados na Tabela 109 e as funções MODBUS (FC) disponíveis na Tabela 107.

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 86.

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 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 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.

5.6.6. MODBUS RTU Escravo

Este protocolo está disponível para as UCPs da Série Nexto nos seus canais seriais. Ao selecionar esta opção no Mastertool, 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. 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 da Interface Serial](#).

5.6.6.1. 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.

5.6.6.1.1. 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.

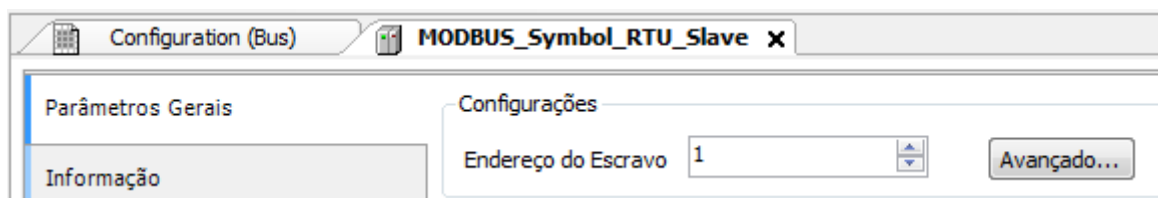


Figura 62: Configuração do Escravo

Configuração	Descrição	Padrão	Opções
Endereço do Escravo	Endereço do escravo MODBUS	1	1 a 255

Tabela 91: Configuração 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.

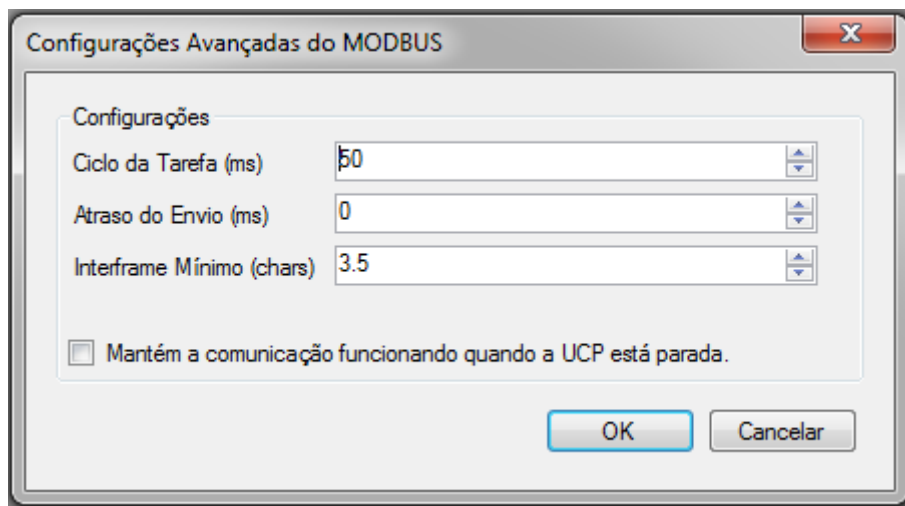


Figura 63: Configurações Avançadas MODBUS do Escravo

Configuração	Descrição	Padrão	Opções
Ciclo da Tarefa (ms)	Tempo para execução da instância dentro do ciclo, sem considerar o tempo de execução da mesma	20	5 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 92: 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.

Mantém a comunicação funcionando quando a UCP está parada: Este parâmetro é válido apenas para CPs não redundantes e não está disponível para CPs redundantes.

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_1* e ainda para o mapeamento por representação direta estão em 4 bytes e 8 words, os quais estão descritos na tabela abaixo:

<i>T_DIAG_MODBUS_RTU_SLAVE_1.*</i>	Tamanho	Descrição
Bits de diagnóstico:		
tDiag.bRunning	BIT	O escravo está em execução.
tDiag.bNotRunning	BIT	O escravo não está em execução (ver bit: bInterruptedByCommand).
tDiag.bInterruptedByCommand	BIT	O bit bNotRunning foi habilitado, pois o escravo foi interrompido pelo usuário através de bits de comando.

T_DIAG_MODBUS_RTU_SLAVE_1.*	Tamanho	Descrição
tDiag.bConfigFailure	BIT	Falha na configuração.
tDiag.bModuleFailure	BIT	Não implementado.
Códigos de erro:		
eErrorCode	SERIAL_STATUS (BYTE)	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 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:		
tCommand.bStop	BIT	Parar o escravo.
tCommand.bRestart	BIT	Reiniciar o escravo.
tCommand.bResetCounter	BIT	Reiniciar as estatísticas dos diagnósticos (contadores).
Estatísticas de comunicação:		
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).
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).
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).
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).
tStat.wRXOverrunErrors	WORD	Contador de frames com erros de overrun durante a recepção – UART FIFO ou fila RX – (0 a 65535).
tStat.wRXIncompleteFrames	WORD	Contador de frames com erro de construção, paridade ou falha durante a recepção (0 a 65535).
tStat.wCTSTimeOutErrors	WORD	Contador de erro de time-out no CTS, utilizando o handshake RTS/CTS, durante a transmissão (0 a 65535).

Tabela 93: 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.

5.6.6.1.2. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico

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

Figura 64: 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 94: 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 95: 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.

5.6.6.2. 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.

5.6.6.2.1. 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:

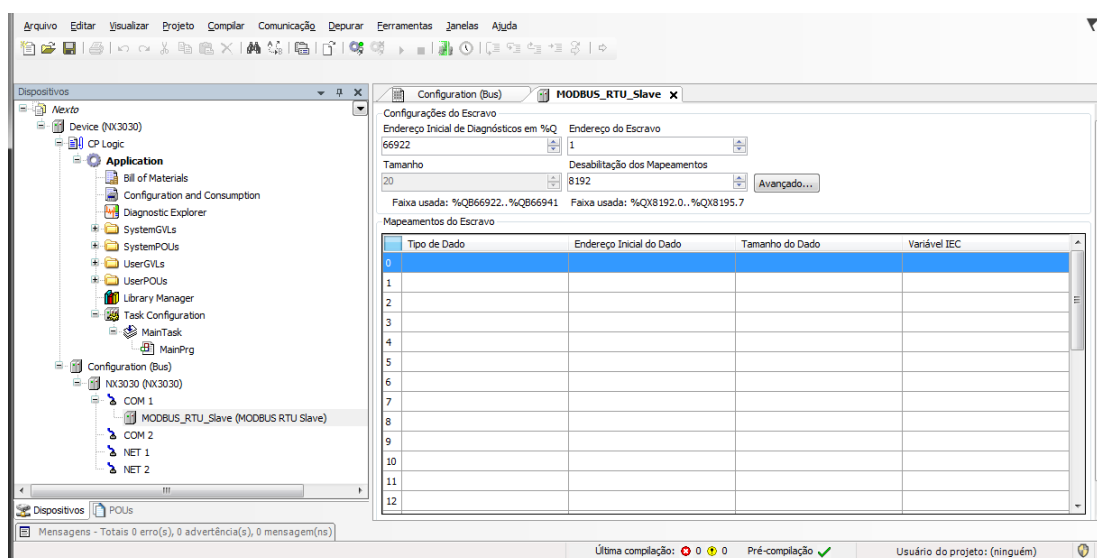


Figura 65: Tela de Configuração MODBUS RTU Escravo por Representação Direta

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

Configuração	Descrição	Padrão	Opções
--------------	-----------	--------	--------

Tabela 96: 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 alocue 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 [93](#).

Os tempos de comunicação do protocolo MODBUS Escravo, encontrados no botão *Avançado...* da tela de configuração, estão descritos na Tabela [92](#).

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

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

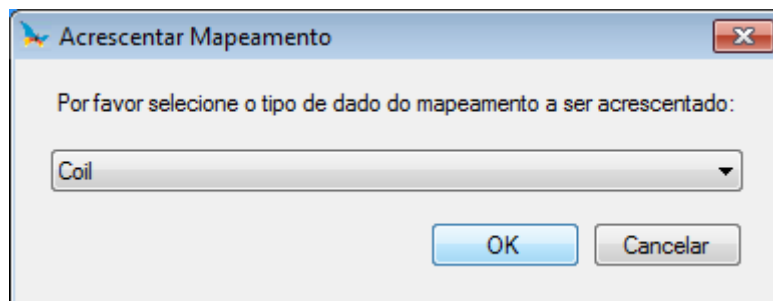


Figura 66: Adicionando Relações MODBUS

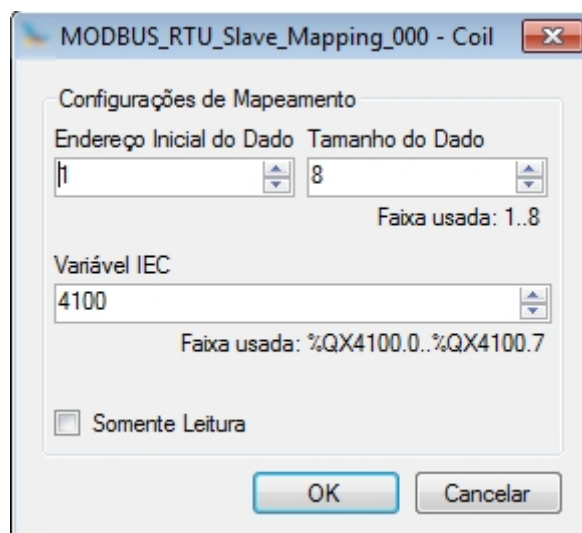


Figura 67: Configurando a Relação MODBUS

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 97: 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 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.

5.6.7. 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 60.

A figura abaixo representa algumas das possibilidades de comunicação utilizando-se o protocolo MODBUS TCP simultaneamente com o protocolo MODBUS RTU via TCP.

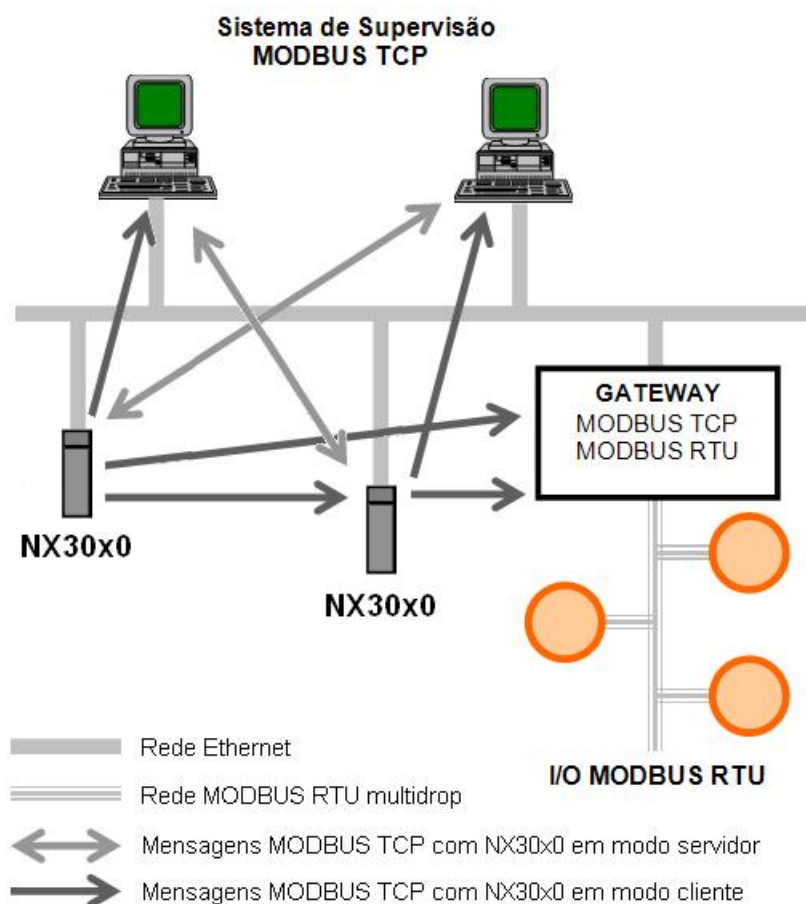


Figura 68: Rede de Comunicação MODBUS TCP

A associação de variáveis MODBUS com variáveis simbólicas da UCP é realizada pelo usuário através da definição de relações via configurador Mastertool. 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. 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 98: 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 99: 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](#).

5.6.8. MODBUS Ethernet Cliente

Este protocolo está disponível para as UCPs da Série Nexto nos seus canais Ethernet. Ao selecionar esta opção no Mastertool, 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 ou na seção [Inserindo uma Instância de Protocolo](#).

5.6.8.1. 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.

5.6.8.1.1. 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:

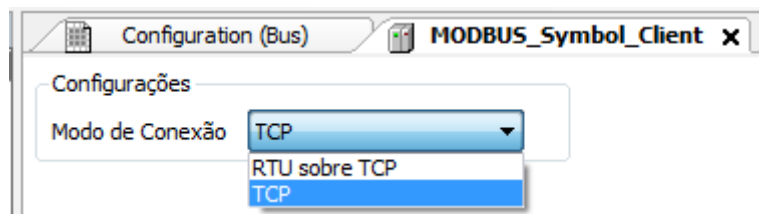


Figura 69: Tela de Configuração Parâmetros Gerais MODBUS Cliente

Configuração	Descrição	Padrão	Opções
Modo de Conexão	Seleção do protocolo	TCP	RTU via TCP TCP

Tabela 100: 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:

T_DIAG_MODBUS_ETH_CLIENT_1.*	Tamanho	Descrição
Bits de diagnóstico:		
tDiag.bRunning	BIT	O cliente está em execução.
tDiag.bNotRunning	BIT	O cliente não está em execução (ver bit bInterruptedByCommand).
tDiag.bInterruptedByCommand	BIT	O bit bNotRunning foi habilitado pois o cliente foi interrompido pelo usuário através de bits de comando.
tDiag.bConfigFailure	BIT	Falha na configuração.
tDiag.bModuleFailure	BIT	Indica se há falha no módulo ou o módulo não está presente.
tDiag.bAllDevicesCommFailure	BIT	Indica que todos os dispositivos configurados no cliente estão com falha.
Bits de comando, reiniciados automaticamente:		
tCommand.bStop	BIT	Parar o cliente.
tCommand.bRestart	BIT	Reiniciar o cliente.
tCommand.bResetCounter	BIT	Reiniciar as estatísticas dos diagnósticos (contadores).
Estatísticas de comunicação:		
tStat.wTXRequests	WORD	Contador de requisições transmitidas pelo cliente (0 a 65535).
tStat.wRXNormalResponses	WORD	Contador de respostas normais recebidas pelo cliente (0 a 65535).
tStat.wRXExceptionResponses	WORD	Contador de respostas com códigos de exceção (0 a 65535).
tStat.wRXIllegalResponses	WORD	Contador de respostas ilegais recebidas pelo cliente – sintaxe inválida, CRC inválido ou número insuficiente de bytes recebidos (0 a 65535).

Tabela 101: Diagnósticos MODBUS Cliente

Nota:

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

5.6.8.1.2. Configuração dos Dispositivos – Configuração por Mapeamento Simbólico

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

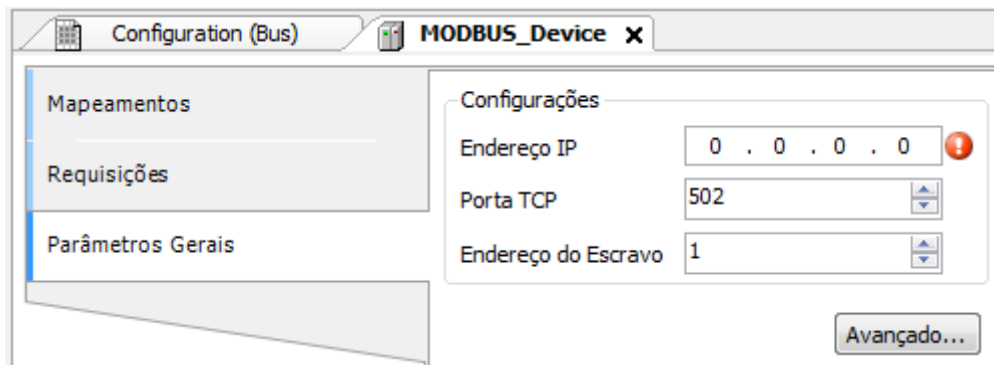


Figura 70: Tela de Configurações dos Parâmetros Gerais do Dispositivo

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 MOD-BUS	1	0 a 255

Tabela 102: Configurações - 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 permanece aberta enquanto houver timeout.	Conexão permanece aberta enquanto não houver um timeout. 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	10 a 3600

Tabela 103: 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 permanece aberta enquanto não houver um timeout:** Esta opção apresenta 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.

5.6.8.1.3. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico

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

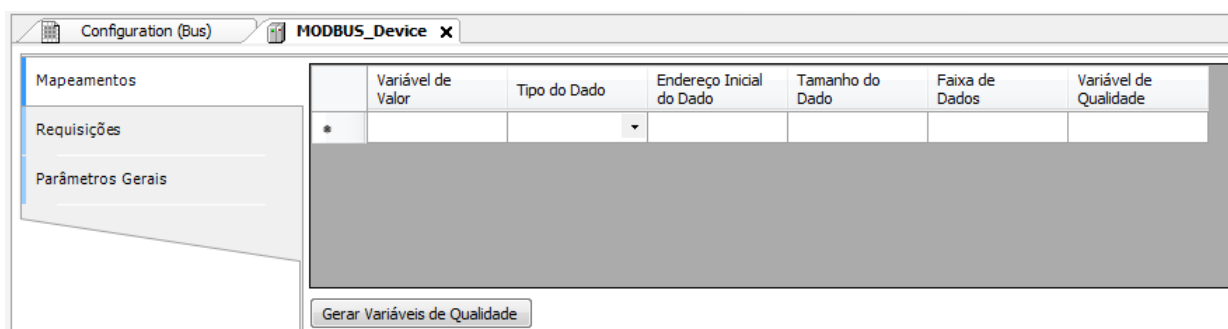


Figura 71: Tela de Mapeamentos de dados MODBUS Cliente

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 104: 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 105: 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.

5.6.8.1.4. Configuração das Requisições – Configuração por Mapeamento Simbólico

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

Figura 72: 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

Configuração	Descrição	Padrão	Opções
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 106: 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 107: 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 86.

T_DIAG_MODBUS_ETH_MAPPING_1.*	Tamanho	Descrição
Bits de estado da comunicação:		
byStatus.bCommIdle	BIT	Comunicação inativa (aguardando ser constada).
byStatus.bCommExecuting	BIT	Comunicação ativa.
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.
byStatus.bCommDisabled	BIT	Comunicação desabilitada. O bit bCommIdle é reiniciado nessa condição.
byStatus.bCommOk	BIT	Comunicação finalizada anteriormente foi realizada com sucesso.
byStatus.bCommError	BIT	Comunicação finalizada anteriormente teve um erro. Verificar código de erro.
byStatus.bCommAborted	BIT	Comunicação finalizada anteriormente foi interrompida devido à falha de conexão.
Último código de erro (habilitado quando o bCommError = verdadeiro):		
eLastErrorCode	MASTER_ERROR_CODE (BYTE)	Informa a possível causa do último erro ocorrido na relação MODBUS. Consulte a Tabela 109 para detalhes das possibilidades.
Último código de exceção recebido pelo cliente:		
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:		
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.
wCommErrorCounter	WORD	Contador de comunicações finalizadas com erros. Quando o valor 65535 é atingido, o contador retorna a zero.

Tabela 108: 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 109: 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.

5.6.8.2. 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.
- 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.

5.6.8.2.1. 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:

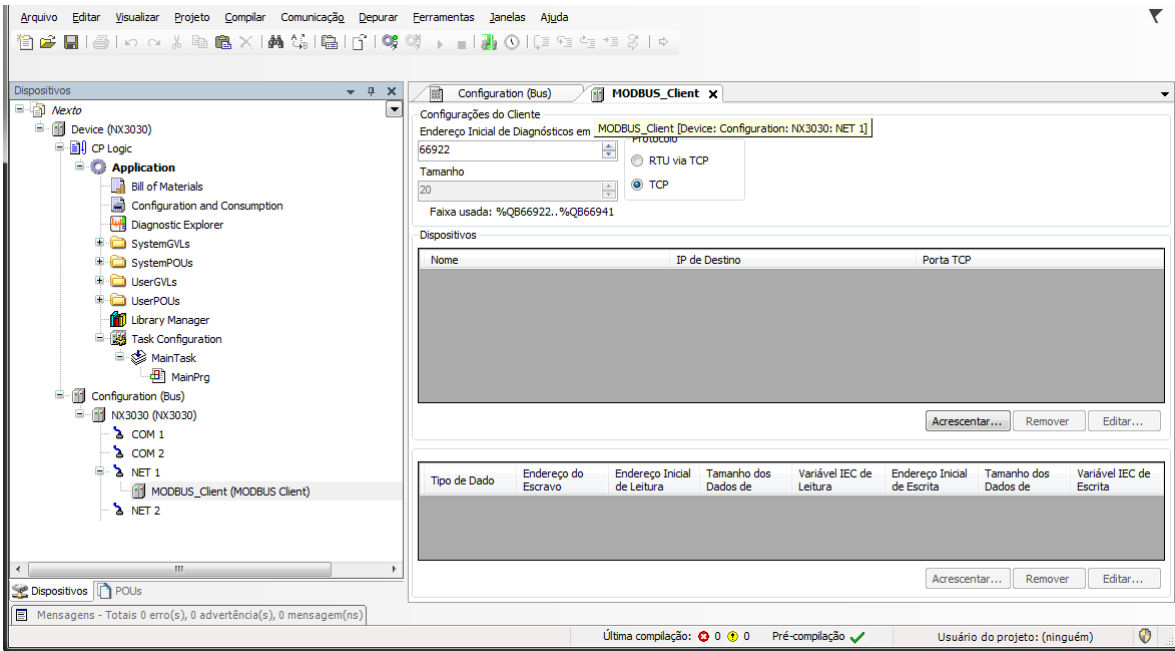


Figura 73: Tela de Configuração – MODBUS Cliente

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 110: 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 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 [101](#).

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

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

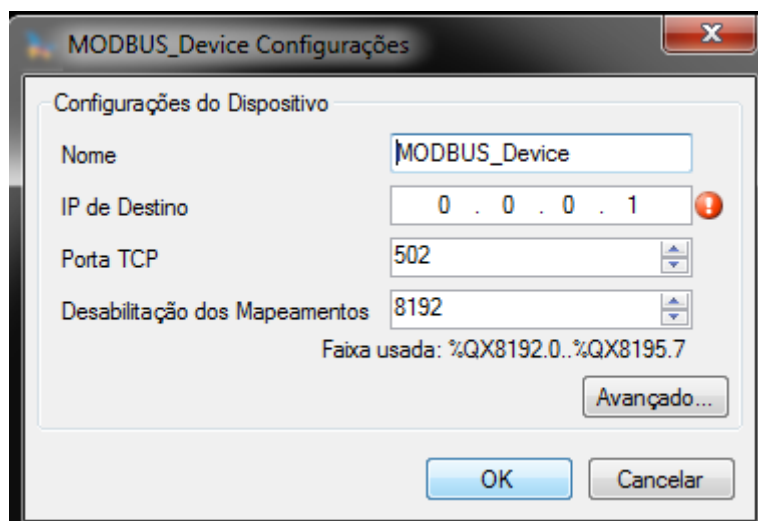


Figura 74: Configurando Cliente MODBUS

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 111: 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 alocue 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](#).

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

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

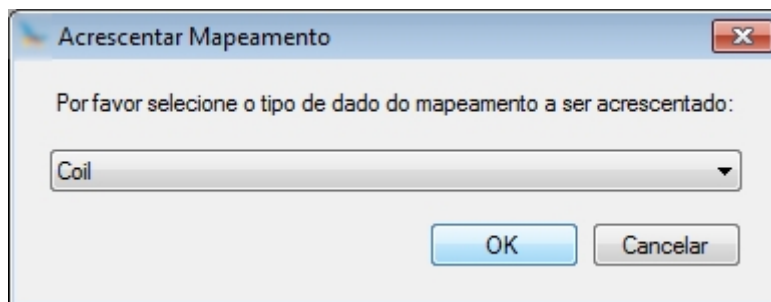


Figura 75: Tipo de Dado MODBUS

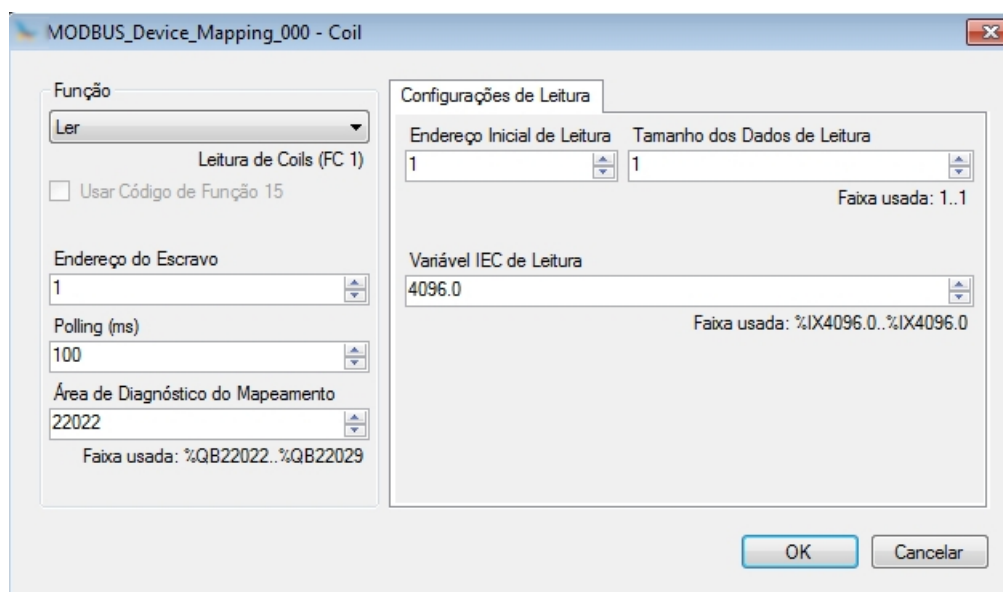


Figura 76: Função MODBUS

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
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

Configuração	Descrição	Padrão	Opções
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 112: 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 86.

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 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 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.

5.6.8.3. 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.

5.6.9. MODBUS Ethernet Servidor

Este protocolo está disponível para as UCPs da Série Nexto nos seus canais Ethernet. Ao selecionar esta opção no Mastertool, 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.

5.6.9.1. 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.

5.6.9.1.1. 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.

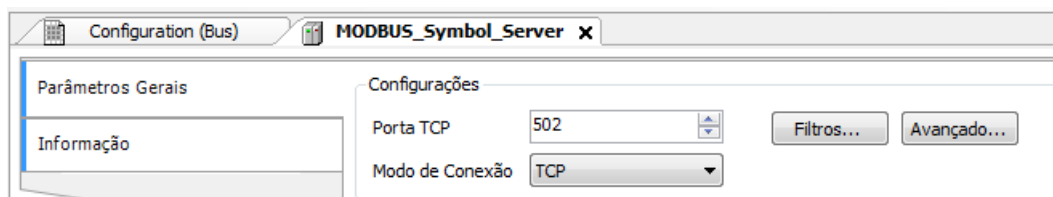


Figura 77: Tela de Configuração Parâmetros Gerais MODBUS Servidor

Configuração	Descrição	Padrão	Opções
Porta TCP	Porta TCP	502	2 a 65534
Modo de Conexão	Seleção do protocolo	TCP	RTU via TCP TCP

Tabela 113: 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 114: 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*.

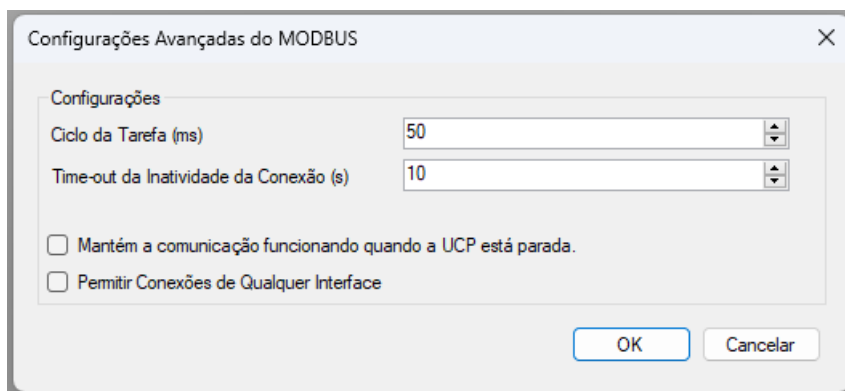


Figura 78: Janela de Configurações Avançadas do MODBUS Servidor

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
Permitir Conexões de Qualquer Interface	Habilita conexões através de qualquer interface de rede, incluindo a VPN	desmarcado	marcado ou desmarcado

Tabela 115: 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.

Mantém a comunicação funcionando quando a UCP está parada: Este parâmetro é válido apenas para CPs não redundantes e não está disponível para CPs redundantes.

Permitir Conexões de Qualquer Interface: Quando habilitado, o servidor MODBUS aceitará conexões de qualquer interface disponível no dispositivo, incluindo interfaces VPN. Quando desabilitado, ele aceitará conexões apenas na interface onde está instanciado. Ao habilitar essa opção, certifique-se de que nenhum outro protocolo no projeto use a mesma porta do servidor MODBUS, para evitar conflitos de comunicação.

5.6.9.1.2. 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:

T_DIAG_MODBUS_ETH_SERVER_1.*	Tamanho	Descrição
Bits de diagnóstico:		
tDiag.bRunning	BIT	O servidor está em execução.
tDiag.bNotRunning	BIT	O servidor não está em execução (ver bit bInterruptedByCommand).
tDiag.bInterruptedByCommand	BIT	O bit bNotRunning foi habilitado, pois o servidor foi interrompido pelo usuário através de bits de comando.
tDiag.bConfigFailure	BIT	Falha na configuração.
tDiag.bModuleFailure	BIT	Indica se há falha no módulo ou o módulo não está presente.
Bits de comando, reiniciados automaticamente:		
tCommand.bStop	BIT	Parar o servidor.
tCommand.bRestart	BIT	Reiniciar o servidor.
tCommand.bResetCounter	BIT	Reiniciar as estatísticas dos diagnósticos (contadores).
Estatísticas de comunicação:		
tStat.wActiveConnections	WORD	Número de conexões estabelecidas entre cliente e servidor (0 a 64).
tStat.wTimeoutClosedConnections	WORD	Contador de conexões, entre cliente e servidor, interrompidas depois de um período de inatividade – time-out (0 a 65535).
tStat.wClientClosedConnections	WORD	Contador de conexões interrompidas devido à solicitação do cliente (0 a 65535).
tStat.wRXFrames	WORD	Contador de frames Ethernet recebidos pelo servidor, sendo que um frame Ethernet pode conter mais de uma requisição (0 a 65535).
tStat.wRXRequests	WORD	Contador de requisições recebidas pelo servidor e respondidas normalmente (0 a 65535).
tStat.wTXExceptionResponses	WORD	Contador de requisições recebidas pelo servidor e respondidas com códigos de exceção (0 a 65535).
tStat.wRXIllegalRequests	WORD	Contador de requisições ilegais. (0 a 65535).

Tabela 116: Diagnósticos MODBUS Servidor

Nota:

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

bModuleFailure: Diagnóstico implementado apenas para MODBUS simbólico.

5.6.9.1.3. 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 79: 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 117: 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.

5.6.9.2. 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.

5.6.9.2.1. 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:

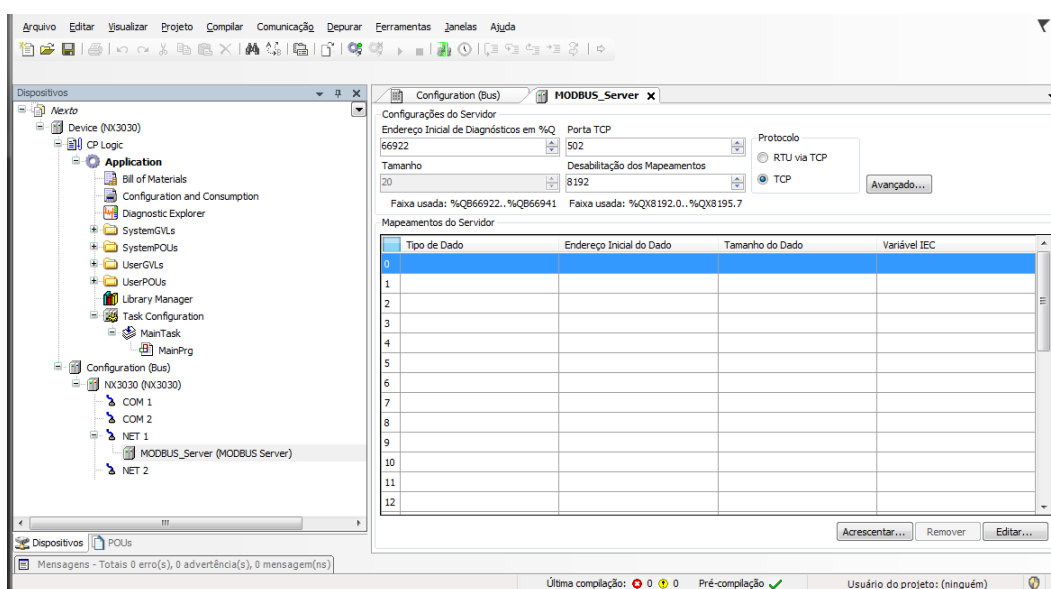


Figura 80: Tela de Configuração MODBUS Servidor

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 118: 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 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 116.

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

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

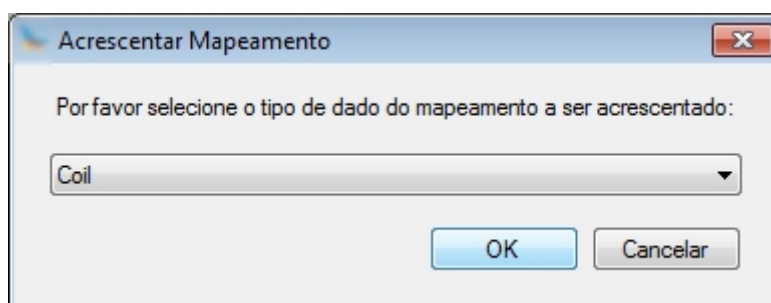


Figura 81: Tipo de Dado MODBUS

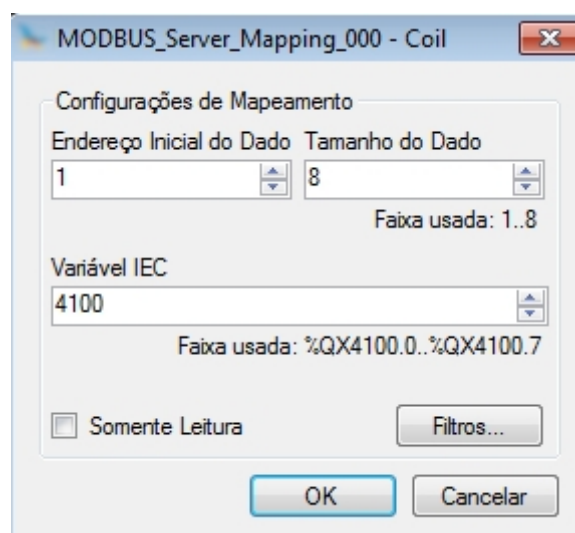


Figura 82: Função MODBUS Servidor

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)

Configuração	Descrição	Padrão	Opções
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 119: 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 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 120: 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.

5.6.10. 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).

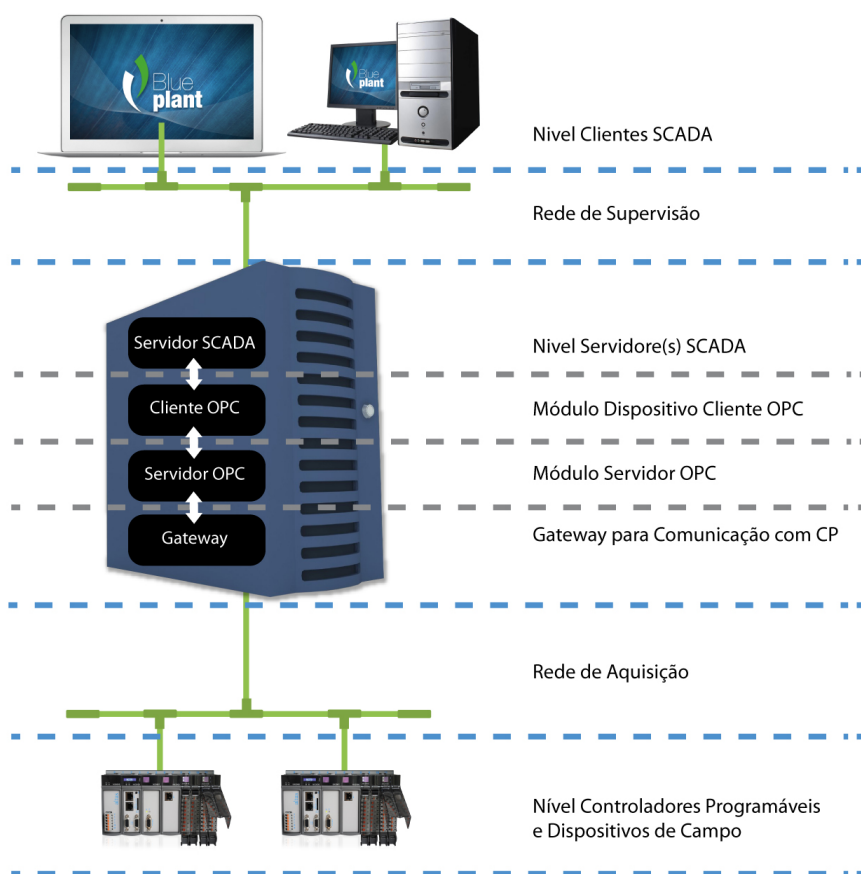


Figura 83: Arquitetura OPC DA

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.
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.

Papel	Descrição
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 121: 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 e sua configuração é realizada dentro da ferramenta. Vale salientar que o OPC DA está disponível somente nas interfaces Ethernet integradas das UCPs Nexto. Os módulos de expansão Ethernet não suportam essa funcionalidade.

5.6.10.1. 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.

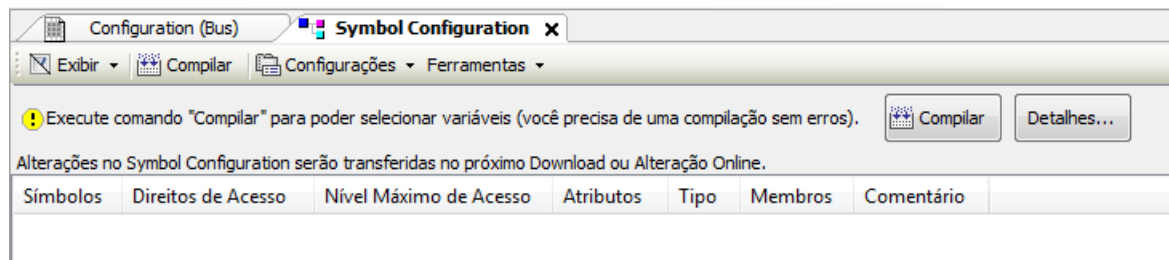


Figura 84: Objeto Symbol Configuration

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 
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 122: 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 84 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.

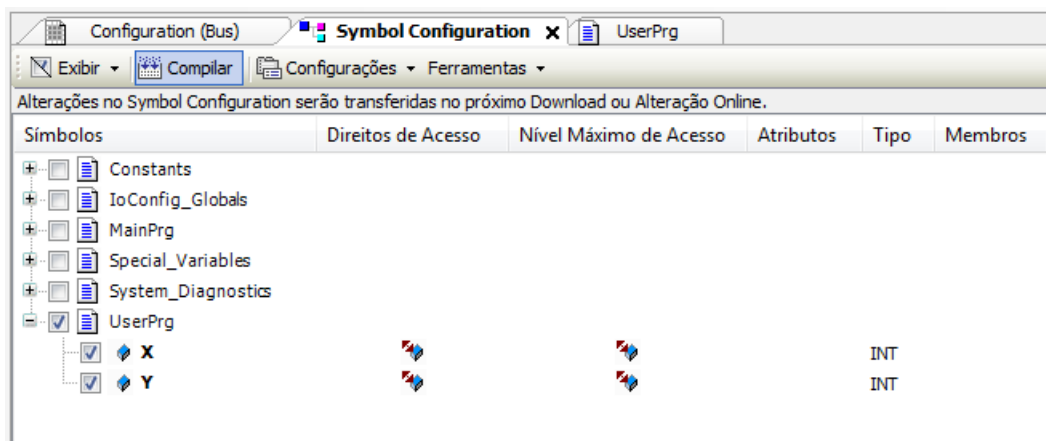


Figura 85: Selecionando Variáveis na Symbol Configuration

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.

5.6.10.2. Configurando um CP no Servidor OPC DA

A configuração de um CP é executada dentro do Mastertool através da opção disponível no menu *Comunicação*. É necessário que o Mastertool seja executado como administrador.

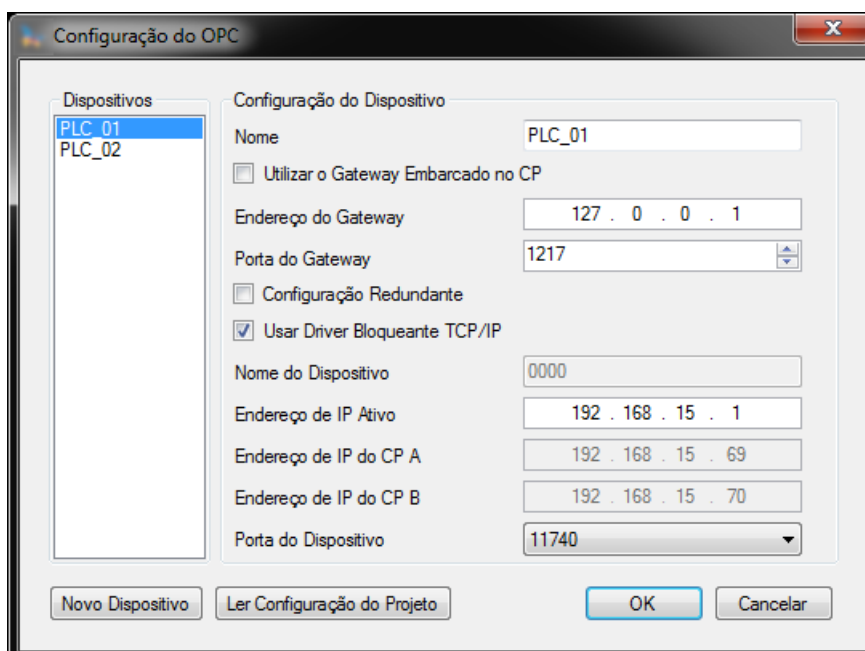


Figura 86: Configuração do OPC DA Servidor

A *Configuração do Gateway* é a mesma configurada no Gateway utilizado para comunicação entre o Mastertool e o CP e descrita em Configurações de Comunicação, presente no Manual de Utilização Mastertool. 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.

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 porta 11739 é apenas para uso em modo simulação). 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
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 Device. 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 Device. 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

Configuração	Descrição	Padrão	Opções
Porta do Dispositivo	Porta TCP. Somente habilitado quando o checkbox <i>Usar Driver Bloqueante TCP/IP</i> está selecionado.	11740	11740 ou 11739

Tabela 123: 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, 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 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 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. 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.

5.6.10.2.1. Importando uma Configuração do Projeto

Utilizando o botão *Ler Configuração do Projeto*, conforme a Figura 86, 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. 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.

5.6.10.3. 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.
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 mantém esta configuração íntegra.

Tabela 124: 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 *QQSSSSL*, 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	00SSSSL	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	01SSSSL	Uncertain	A qualidade não pode ser definida e pode ser apresentada no campo <i>Substatus</i> .
2	10SSSSL	NA	Este valor é reservado e não é utilizado pelo padrão OPC DA.

QQ	Valor dos Bits	Definição	Descrição
3	11SSSSL	Good	A qualidade é boa e o valor lido pode ser usado.

Tabela 125: Descrição do valor da Qualidade OPC DA

A Tabela 125 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.

5.6.10.4. 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	Ver Nota
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 126: 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. Ou seja, a soma de conexões com Servidor OPC DA e Mastertool não deve ultrapassar o número máximo definido na Tabela 126.

A comunicação entre o Servidor OPC DA e o CP utiliza o mesmo protocolo utilizado para comunicação do Mastertool 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.

5.6.10.5. 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 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*.

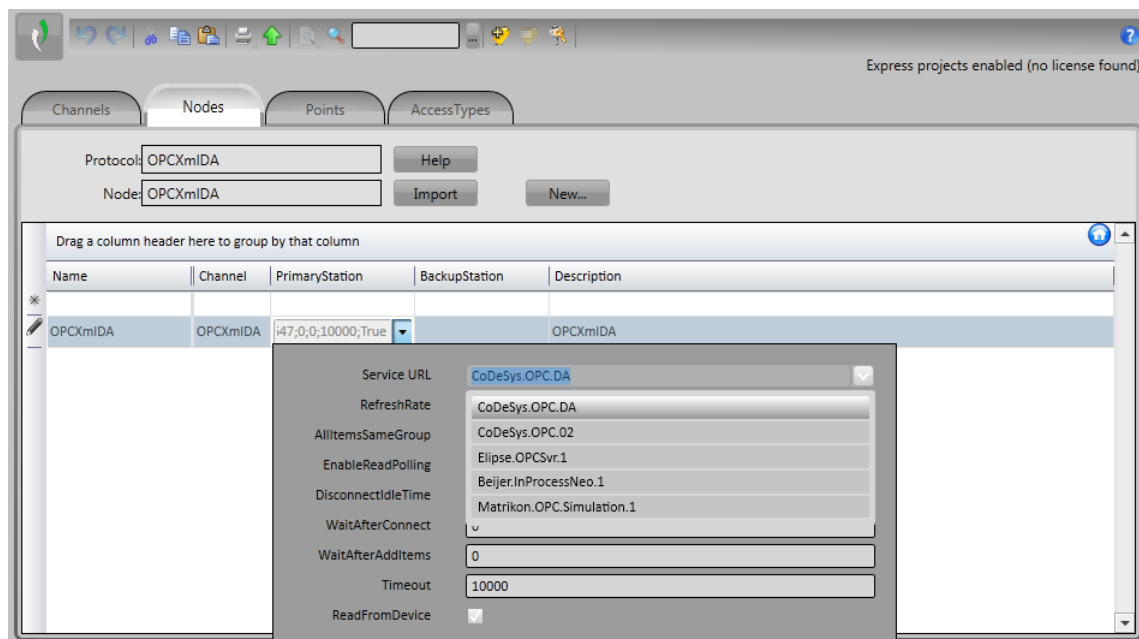


Figura 87: Selecionando Servidor OPC DA na Configuração do Cliente

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.

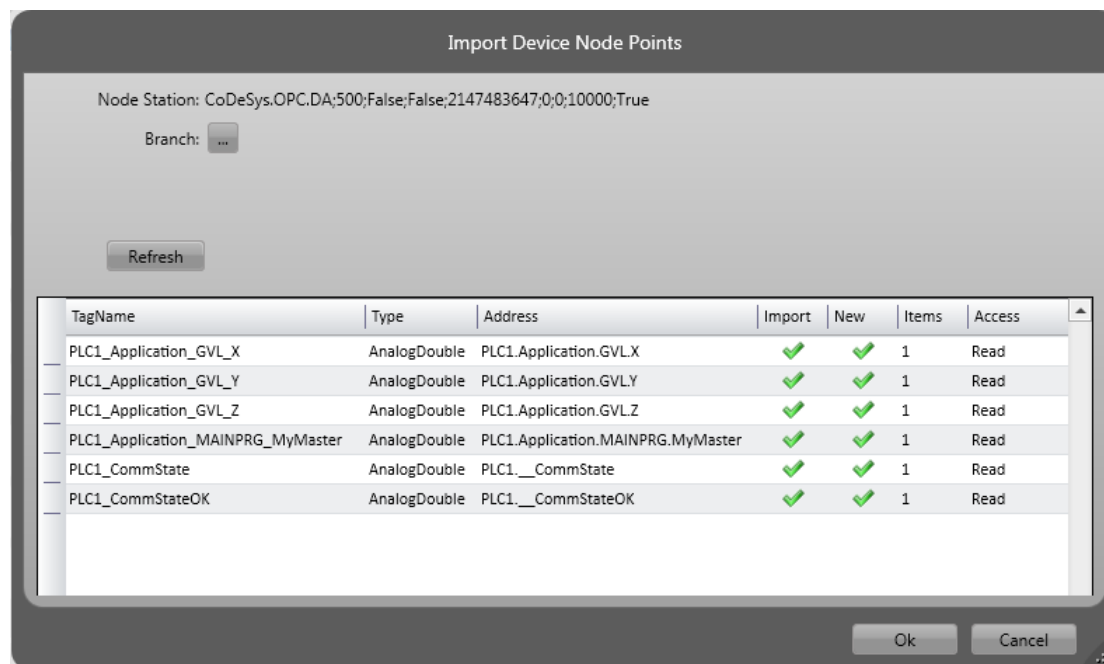


Figura 88: Lista de Símbolos Consultados pelo Cliente OPC DA

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 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.

5.6.11. 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.

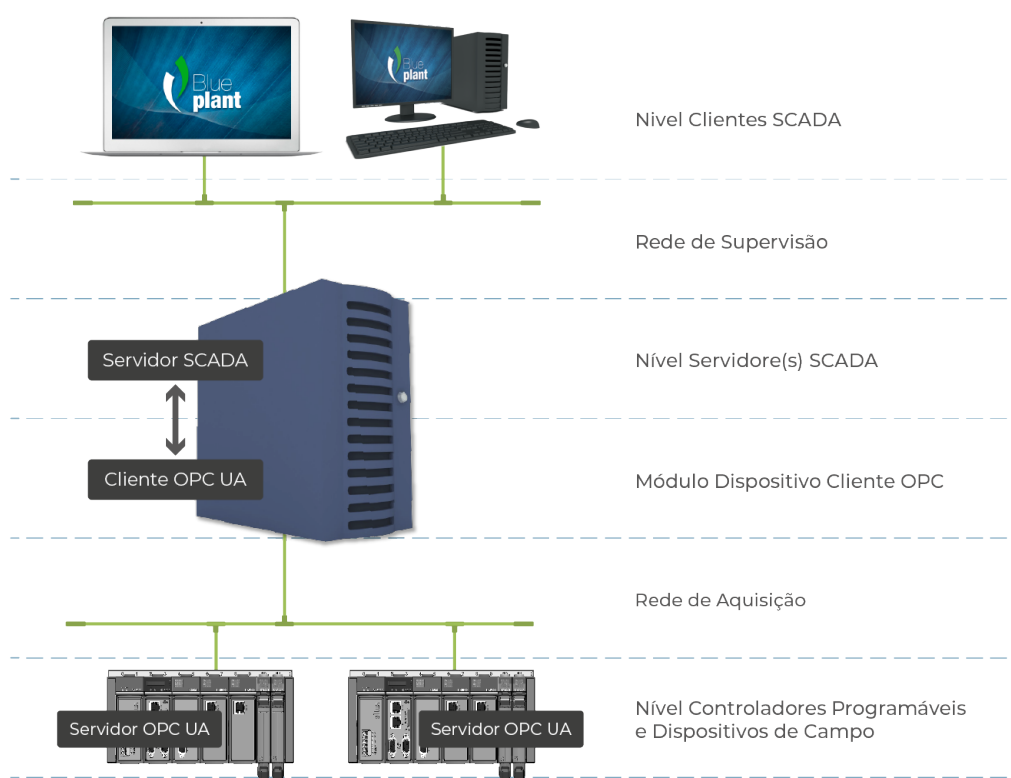


Figura 89: Arquitetura Típica OPC UA

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 127: 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 integradas das UCPs Nexto. Os módulos de expansão Ethernet não suportam essa funcionalidade.

Para informações detalhadas sobre a configuração e o funcionamento do protocolo OPC UA consulte o OPC UA Server for Altus Controllers User Manual - MU214609.

5.6.12. 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 μ S ou 100 eixos de servomotores a cada 100 μ S usando par trançado ou cabos de fibra óptica. Além disto, 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.

5.6.12.1. 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. 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 *Acrescentar 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 *Acrescentar Dispositivo* (menu de contexto do Mestre EtherCAT) ou executando o comando *Procurar Dispositivos*.

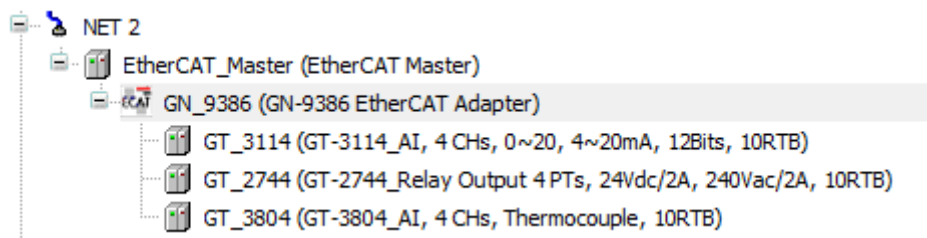


Figura 90: Exemplo de Configuração EtherCAT

ATENÇÃO

- É permitida apenas uma instância do Mestre EtherCAT por projeto.
- 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.

5.6.12.1.1. 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.

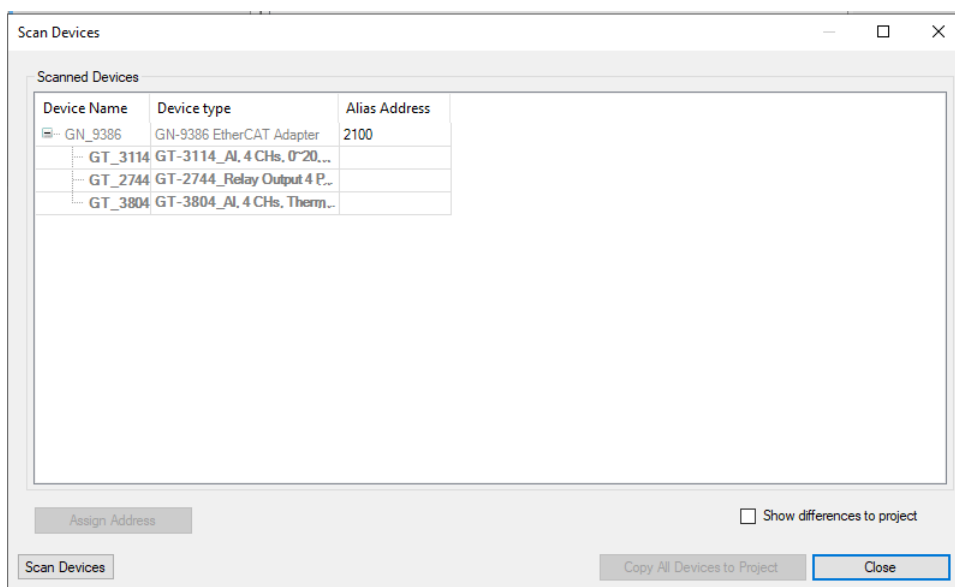


Figura 91: Diálogo Procurar Dispositivos para EtherCAT

5.6.12.2. 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*.

5.6.12.2.1. EtherCAT Master - General

A seguir são apresentados os parâmetros gerais encontrados na tela inicial de configuração do Mestre EtherCAT, conforme figura abaixo.

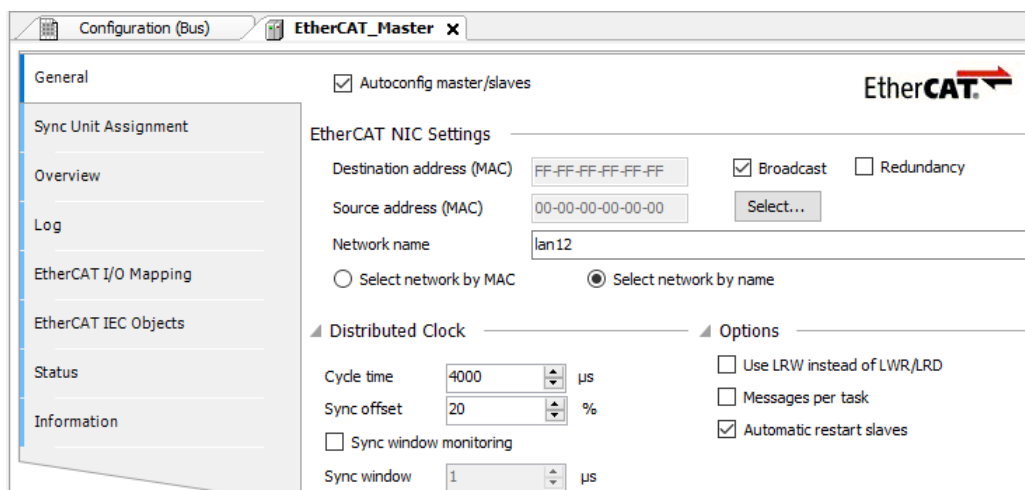


Figura 92: Diálogo de Configuração do Mestre EtherCAT

Configuração	Descrição	Padrão	Opções
Autoconfig master/slaves	Habilita a configuração automática do Mestre e dos Escravos.	Marcado	Marcado Desmarcado
EtherCAT NIC Settings			
Network name	Configura o nome da rede. Não deve ser alterado.	-	-
Select network by name	A rede é identificada pelo nome da rede e o projeto é independente de dispositivo. Não deve ser alterado.	Marcado	-
Redundancy	Habilita a redundância em uma topologia em anel. A rede EtherCAT permanece operacional mesmo em caso de falha de cabo. Os parâmetros de redundância são configurados na seção Configurações da NIC EtherCAT de Redundância.	Desmarcado	Marcado Desmarcado
Distributed Clock			
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
Options			
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
Mailbox Gateway			
Enable	Permite a comunicação de uma ferramenta externa de configuração de dispositivos, por meio do gateway de mailbox, com os dispositivos EtherCAT.	Desmarcado	Marcado Desmarcado

Tabela 128: 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.

5.6.12.2.2. 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.

5.6.12.2.3. 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.

5.6.12.2.4. 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.

5.6.12.2.5. 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.

5.6.12.2.6. 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, exibe, caso disponíveis, as seguintes informações gerais para o módulo: *Nome, Fornecedor, Tipo, Número de Versão, Categorias, Número de Ordem, Descrição, Imagem*.

5.6.12.3. 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*.

5.6.12.3.1. 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.

The screenshot shows the 'Configuration (Bus)' window for a device named 'GN_9483'. The 'General' tab is active. It contains several sections: 'Address' with 'AutoInc address' (0) and 'EtherCAT address' (1001); 'Additional' with 'Expert settings' (checked) and 'Optional' (unchecked); 'Distributed Clock' with 'Select DC' (dropdown) and 'Enable' (checkbox); 'Sync0' and 'Sync1' sections with 'Enable Sync' (checkbox), 'Sync unit cycle' (dropdown), 'Cycle time (µs)' (input), and 'Shift time (µs)' (input); 'Startup Checking' with 'Check vendor ID', 'Check product ID', and 'Check revision number' (all checked); 'DC Cyclic Unit Control: Assign to Local µC' with 'Cyclic unit', 'Latch unit 0', and 'Latch unit 1' (all unchecked); 'Watchdog' with 'Set multiplier (Reg. 16#400)' (2498), 'Set PDI watchdog (Reg. 16#410)' (1000), and 'Set SM watchdog (Reg. 16#420)' (1000); and 'Identification' with 'Disabled' (selected radio button), 'Configured station alias (ADO 0x0012)' (Value: 1001), 'Explicit device identification (ADO 0x0134)' (unchecked), and 'Data Word (2 Bytes)' (ADO (hex): 16#0).

Figura 93: 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

Configuração	Descrição	Padrão	Opções
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
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	Selecione 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	Selecione 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
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 129: 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.

5.6.12.3.2. 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.

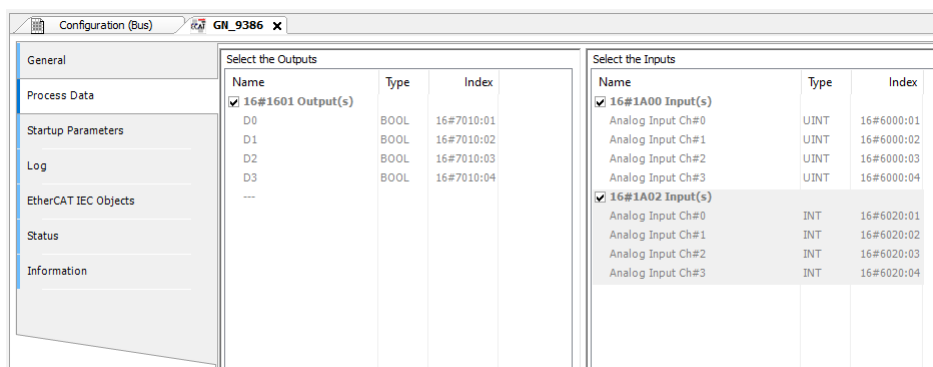


Figura 94: Diálogo Dados do Processo

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.

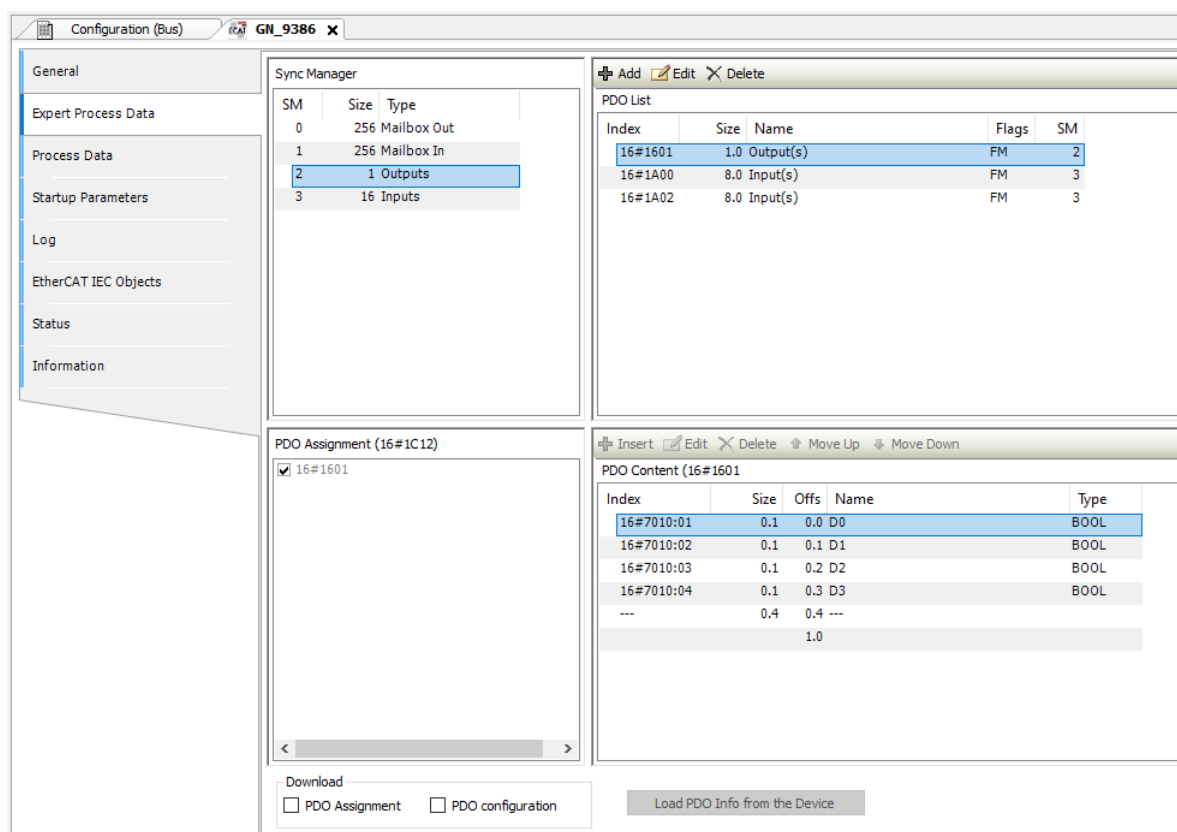


Figura 95: Diálogo Dados de Processo Avançados

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.

5.6.12.3.3. EtherCAT Slave - Edit PDO List

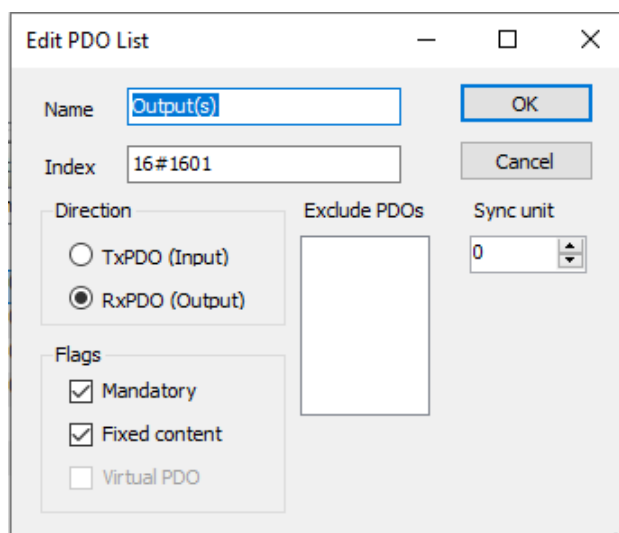


Figura 96: Diálogo Editar Lista PDO

Este diálogo é aberto através do menu de contexto da área Lista PDO, apresentada na Figura 95. 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.

5.6.12.3.4. 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*.

5.6.12.3.5. 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*.

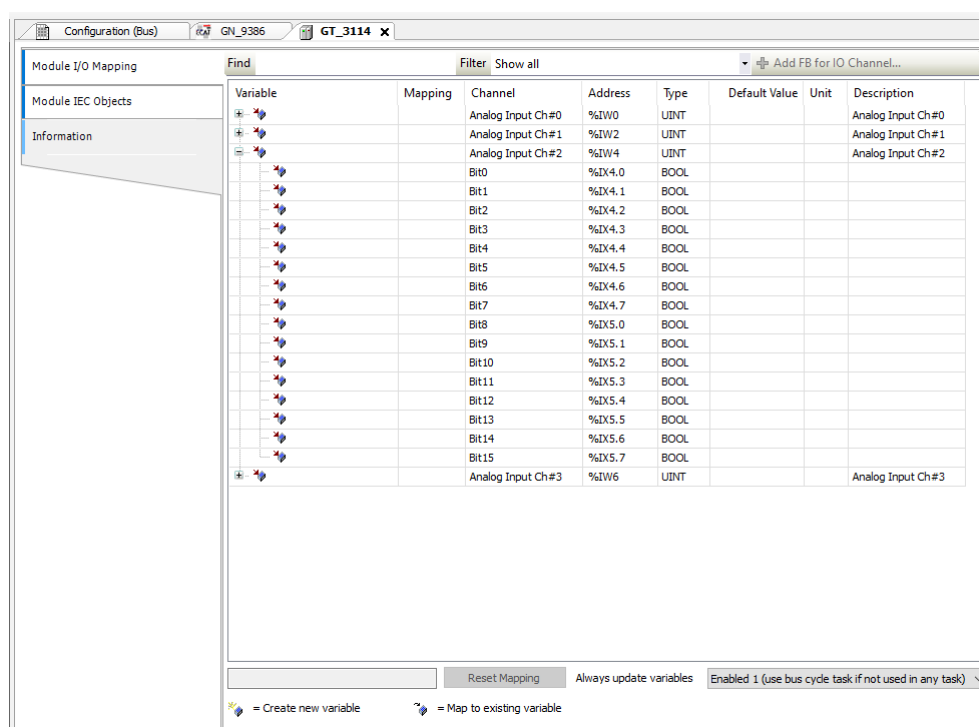


Figura 97: Diálogo Mapeamento de E/S Escravo

5.6.12.3.6. 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*.

5.6.13. 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 utilizado quando as interfaces Ethernet estiverem configuradas em Modo NIC Teaming ou Modo Switch, nem em aplicações com redundância Half-Cluster.

ATENÇÃO

Para evitar problemas de comunicação, o Ethernet/IP Scanner só pode ter dispositivos Adapters que estejam configurados na mesma sub-rede.

5.6.13.1. Adaptador Ethernet

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*.

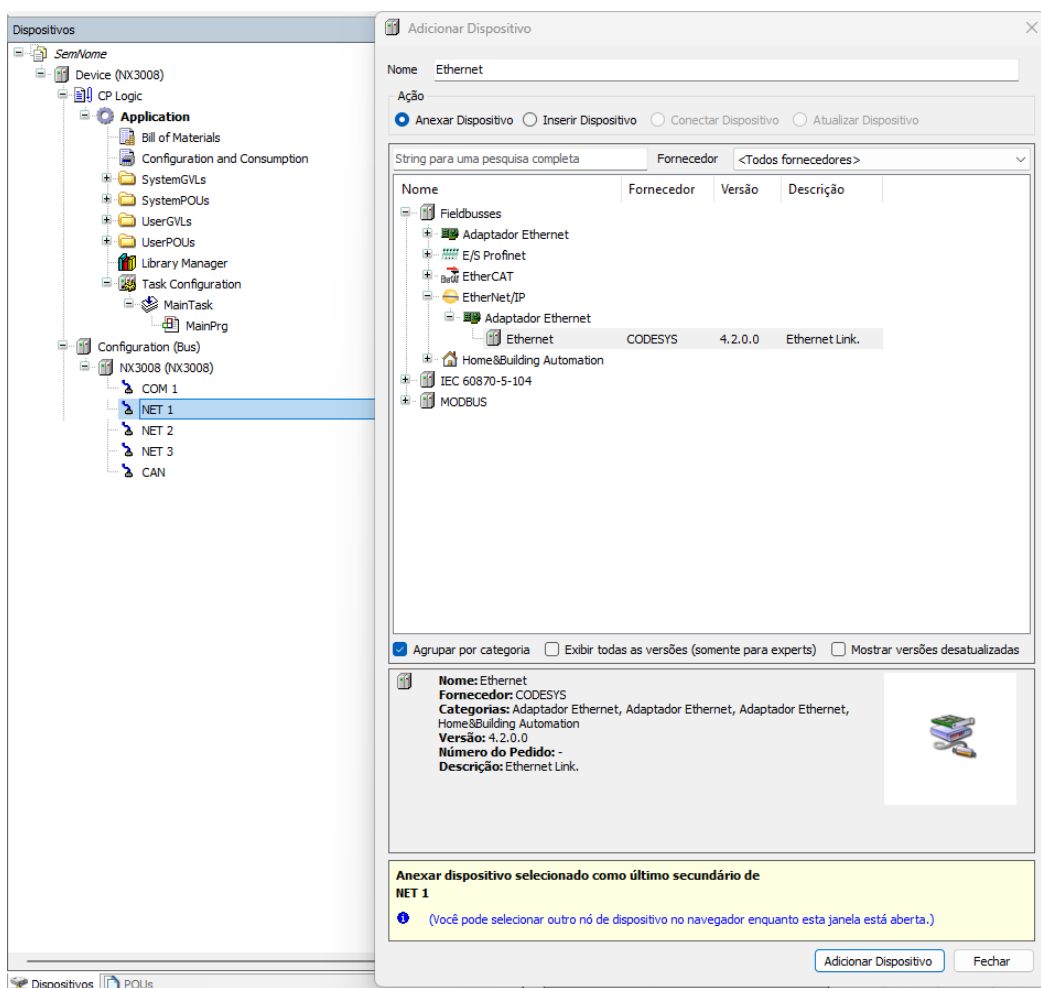


Figura 98: Adicionando um Adaptador Ethernet

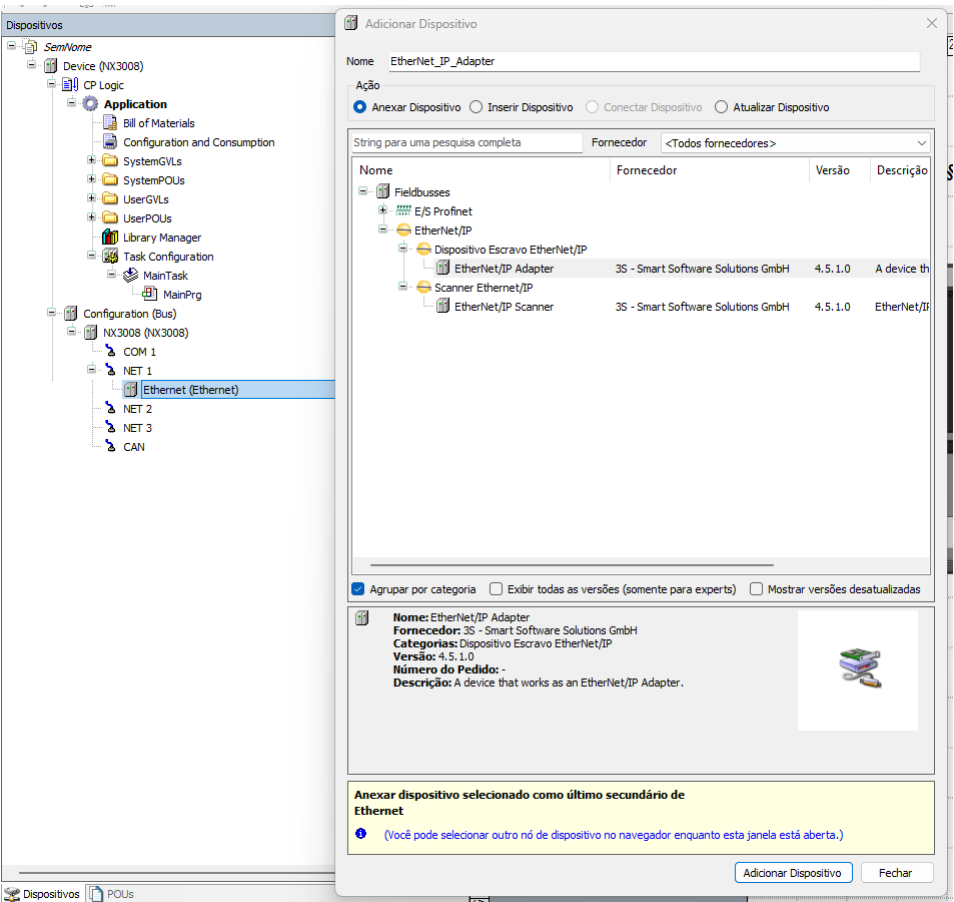


Figura 99: Adicionando um Adapter ou Scanner EtherNet/IP

5.6.13.2. Configuração do Scanner EtherNet/IP

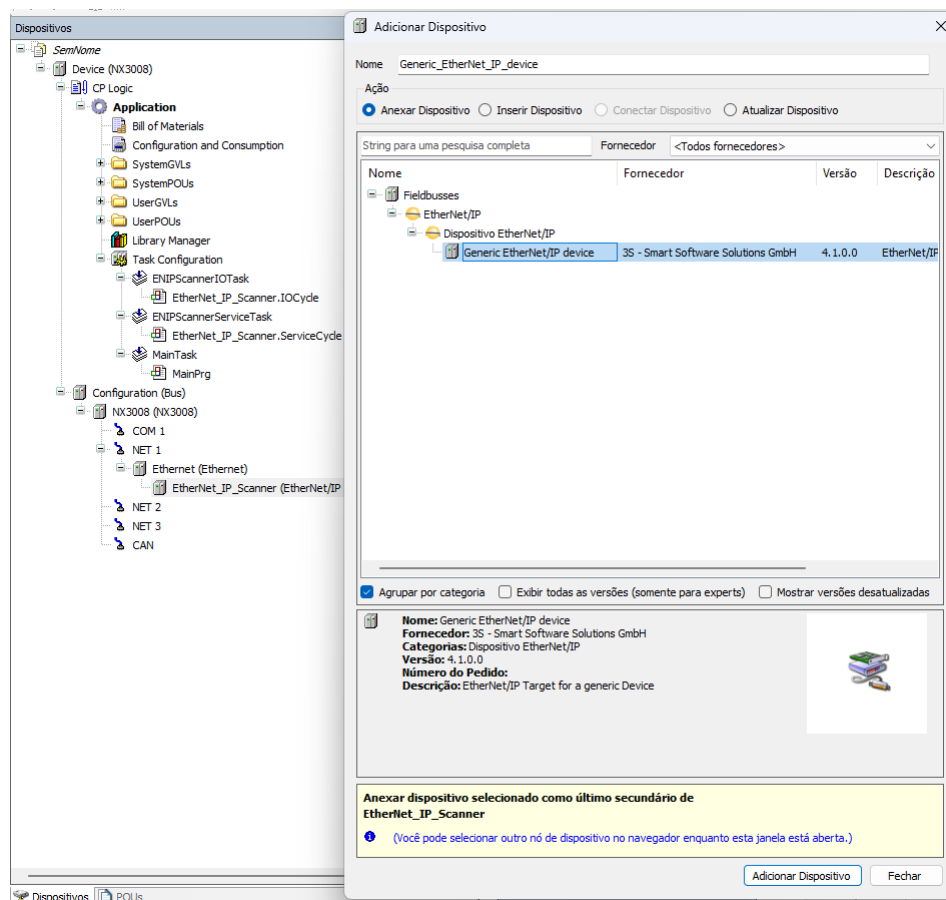


Figura 100: Adicionando um Adapter EtherNet/IP Sob o Scanner

5.6.13.2.1. 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".

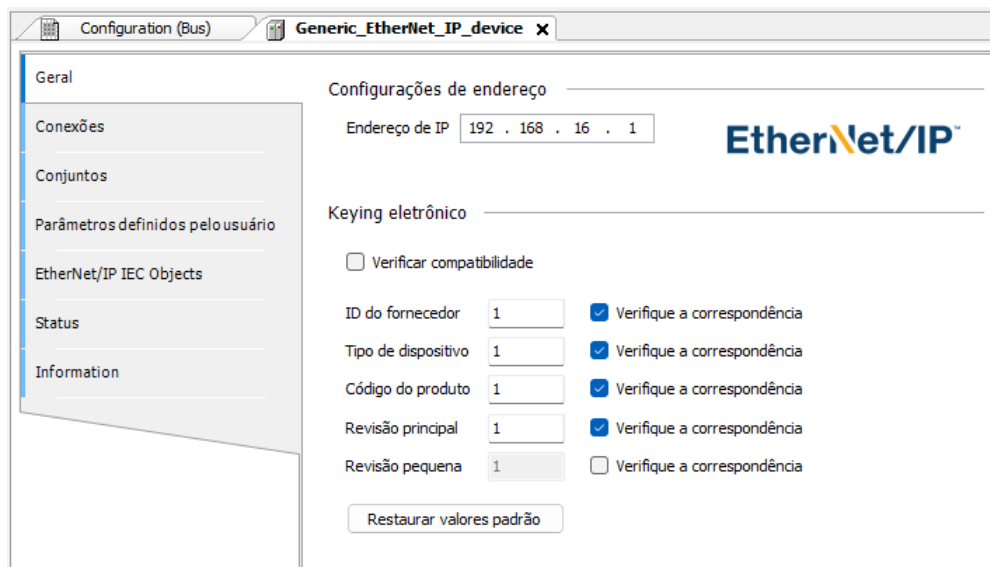


Figura 101: Aba Geral - EtherNet/IP

5.6.13.2.2. 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.

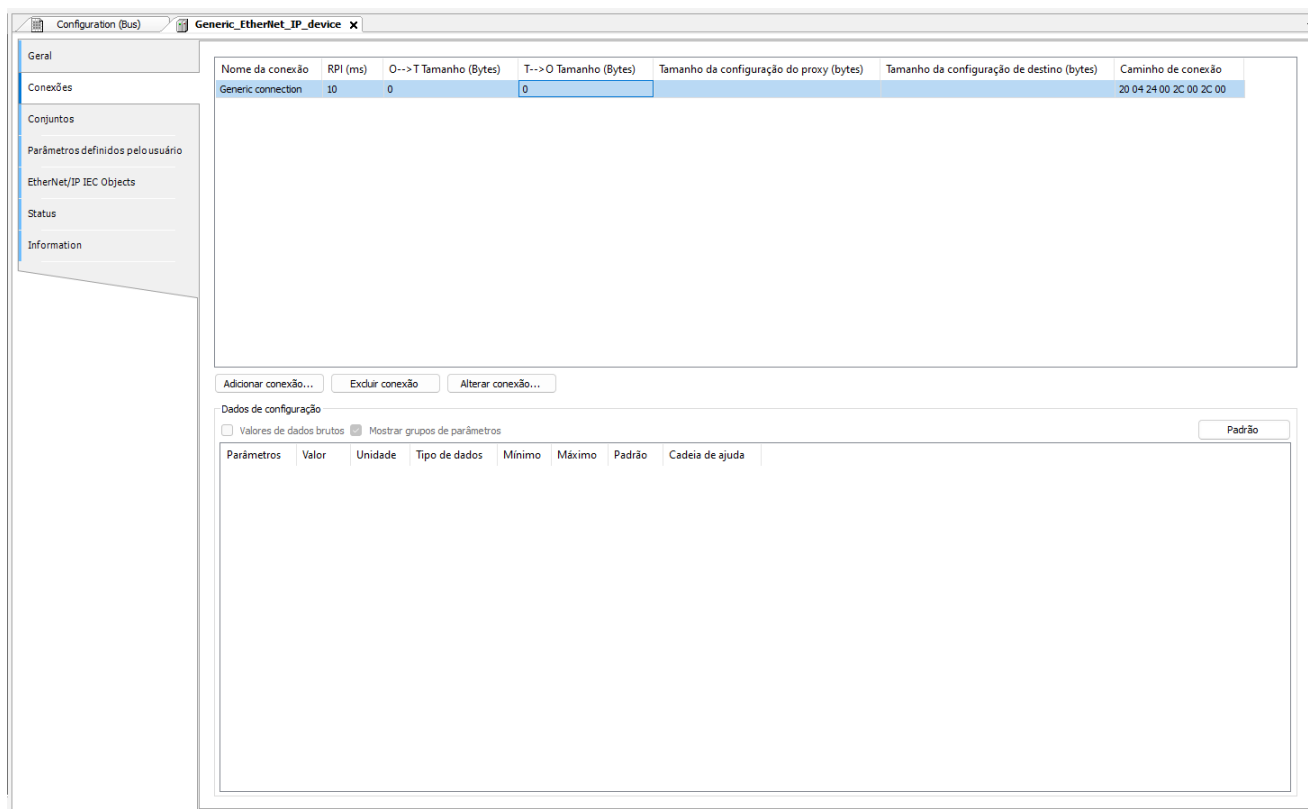


Figura 102: Aba Conexões - EtherNet/IP

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 130: 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.

Nova conexão

☒ Conexão genérica (livremente configurável)
☐ Conexão predefinida (arquivo EDS)

Configurações do caminho de conexão

☒ Caminho gerado automaticamente

☒ Conjunto de configuração
ID de classe: 16# 4 ID da instância: 16# 0 ID de atributo: 16# 3

☒ Conjunto de consumo (O-->T)
ID de classe: 16# 4 ID da instância: 16# 0 ID de atributo: 16# 3

☒ Conjunto de produção (T-->O)
ID de classe: 16# 4 ID da instância: 16# 0 ID de atributo: 16# 3

☐ Caminho definido pelo usuário
☐ Caminho definido pelo nome simbólico

Parâmetros gerais

Caminho de conexão 20 04 24 00 2C 00 2C 00

Tipo de gatilho Cíclico RPI (ms) 10
Tipo de transporte Proprietário exclusivo Multiplicador de tempo limite 4

Scanner para destino (Saída)

O-->T tamanho (bytes) 0
Tamanho da configuração do proxy (bytes) 0
Tamanho da configuração de destino (bytes) 0

Tipo de conexão Ponto a ponto
Prioridade de conexão Programado
Fixo/variável Fixo
Formato de transmissão Execução/inatividade de
Tempo de inibição (ms) 0
Multiplicador do sensor de funcionamento 1

Destino para o scanner (entrada)

T-->O tamanho (bytes) 0

Tipo de conexão Multicast
Prioridade de conexão Programado
Fixo/variável Fixo
Formato de transmissão Dados puros
Tempo de inibição (ms) 0

Figura 103: Janela de Novas Conexões EtherNet/IP

5.6.13.2.3. 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.

Configuration (Bus) Generic_Ethernet_IP_device x

Geral
Conexões
Conjuntos
Parâmetros definidos pelo usuário
EtherNet/IP I/O Mapping
EtherNet/IP IEC Objects
Status
Information

Conexões

Nome da conexão	O-->T Tamanho (Bytes)	T-->O Tamanho (Bytes)	Tamanho da configuração do proxy (bytes)	Tamanho da configuração de destino (bytes)
Conexão genérica	5	5		

- Consuming Assembly "Output" (O-->T)

Nome	Tipo de dados	Comprimento em bit	Unidade	Cadeia de ajuda
Output_Param0	BYTE	8		
Output_Param1	BYTE	8		
Output_Param2	BYTE	8		
Output_Param3	BYTE	8		
Output_Param4	BYTE	8		

Conjunto de produção "Input" (T-->O)

Nome	Tipo de dados	Comprimento em bit	Unidade	Cadeia de ajuda
Input_Param0	BYTE	8		
Input_Param1	BYTE	8		
Input_Param2	BYTE	8		
Input_Param3	BYTE	8		
Input_Param4	BYTE	8		

Figura 104: 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 131: 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 132: Janela “Adicionar Entrada/Saída” – EtherNet/IP

5.6.13.2.4. Parâmetros Definidos pelo Usuário

A aba *Parâmetros Definidos pelo Usuário* exibe todos os parâmetros adicionais que são transmitidos uma única vez para o sistema de barramento durante a fase de inicialização alocada para esse procedimento. Os parâmetros são transmitidos para o adapter remoto por meio de serviços acíclicos.

ATENÇÃO

Os parâmetros do usuário também são transmitidos novamente quando uma conexão é reestabelecida, por exemplo, após a falha de um Adapter remoto.

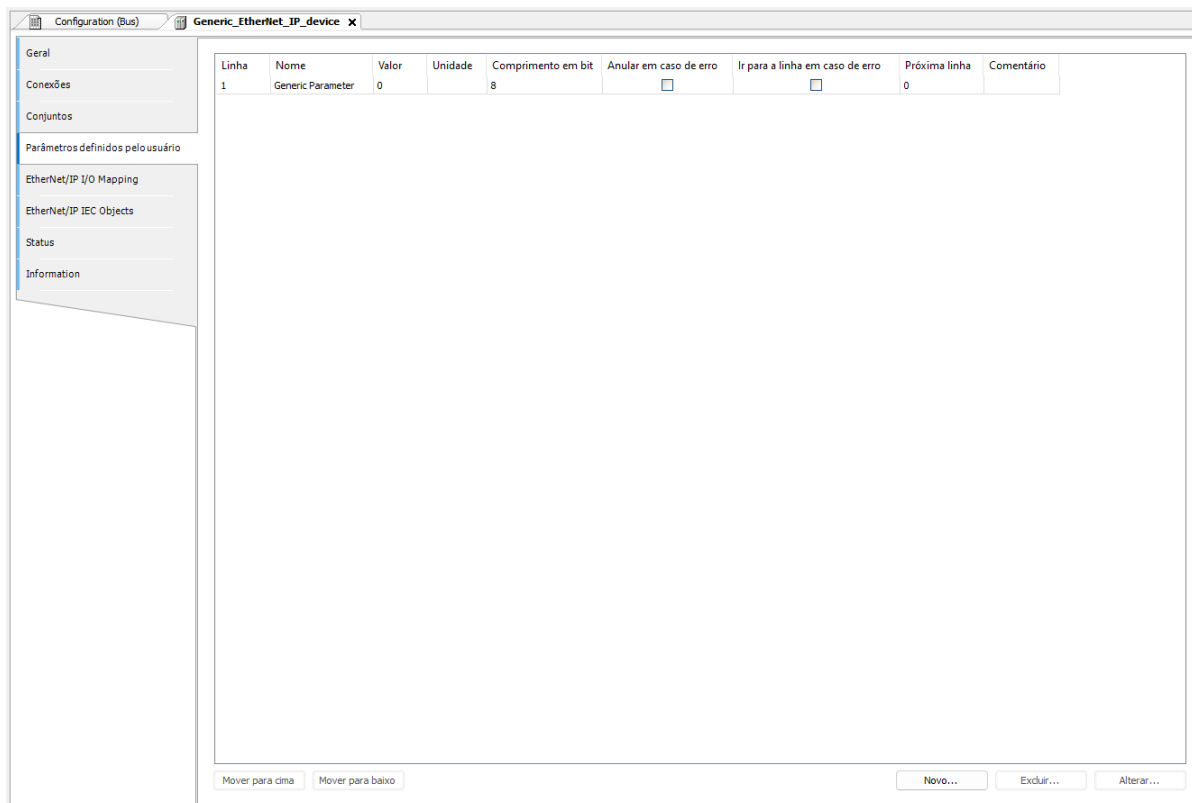


Figura 105: Aba Parâmetros Definidos pelo Usuário - EtherNet/IP

Configuração	Descrição
Novo	Abre a caixa de diálogo Selecionar parâmetros para adicionar um novo parâmetro.
Alterar	Abre a caixa de diálogo Selecionar parâmetros para alterar um parâmetro existente.
Mover Para Cima	Altera a ordem dos parâmetros do usuário.
Mover Para Baixo	A ordem dos parâmetros na lista corresponde à ordem na inicialização.
Valor	O valor do respectivo parâmetro pode ser alterado diretamente com um clique duplo no valor.
Anular em caso de Erro	Em caso de erro, toda a transmissão dos parâmetros é abortada.

Configuração	Descrição
Ir para a linha em caso de Erro	Em caso de erro, o programa é retomado com a linha especificada na coluna Next Line. Dessa forma, um bloco inteiro pode ser ignorado durante a inicialização ou um retorno pode ser definido.

Tabela 133: Aba Parâmetros definidor pelo Usuário - EtherNet/IP

Note:

Ir para a linha em caso de Erro: Um retorno pode levar a um loop infinito se nunca for possível escrever um determinado parâmetro.

5.6.13.2.5. 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.

5.6.13.3. 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.

5.6.13.3.1. 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.

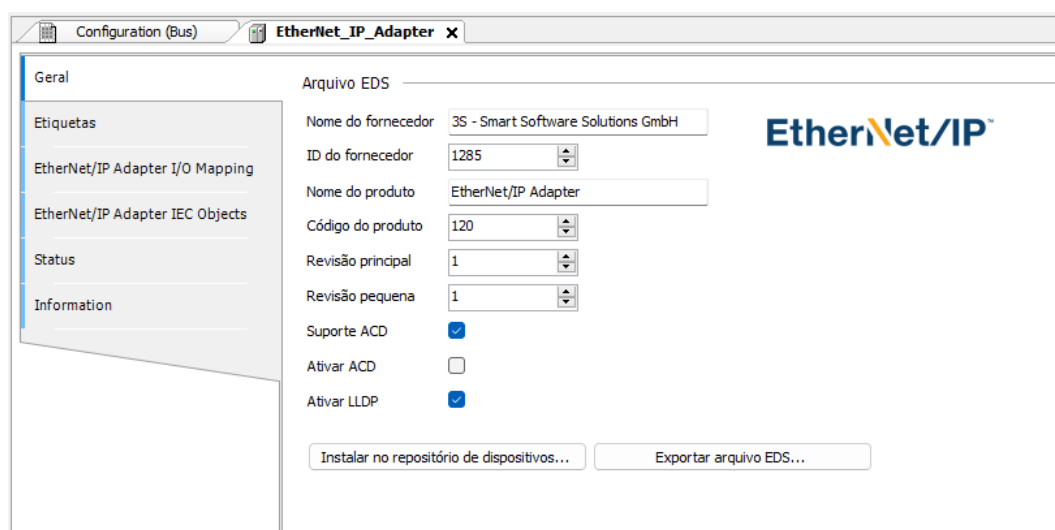


Figura 106: Aba Geral - EtherNet/IP_Adapter

5.6.13.3.2. 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.

5.6.13.4. Configuração do Módulo EtherNet/IP

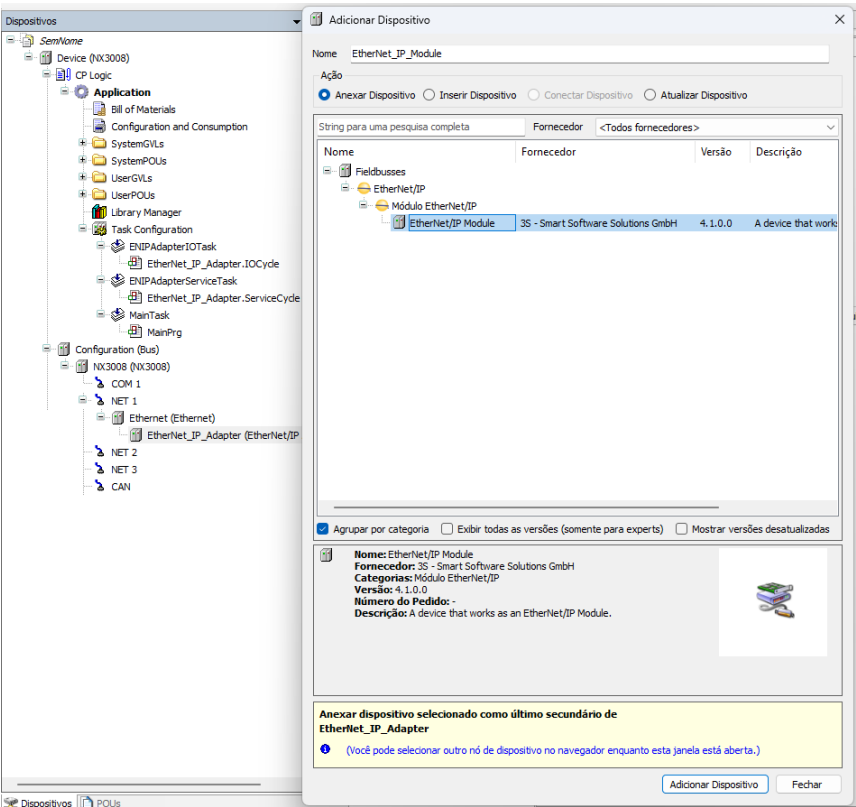


Figura 107: Adicionando um Módulo EtherNet/IP Abaixo do Adapter

5.6.13.4.1. Assemblies

Os parâmetros da aba *Geral* do módulo seguem as mesmas regras descritas nas tabelas 131 e 132.

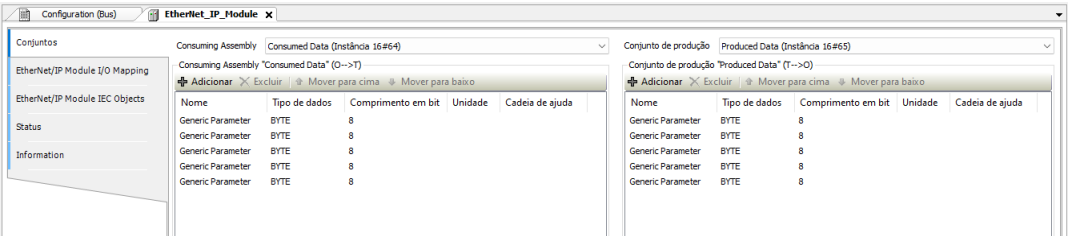


Figura 108: Aba Assemblies Módulo - EtherNet/IP

5.6.13.4.2. 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.

5.6.14. 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.

5.6.14.1. 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

Tipo do Objeto	Tipos de Variáveis IEC
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 134: 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
);
```

5.6.14.2. 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.

5.6.14.2.1. 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

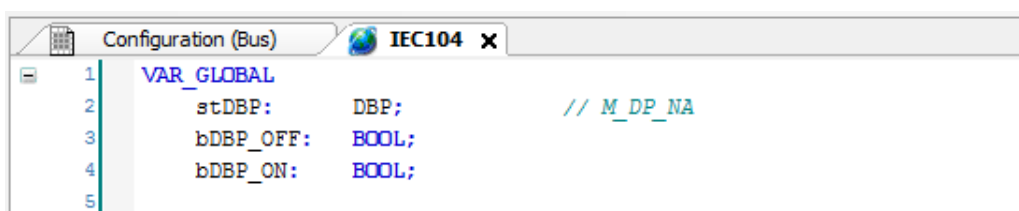


Figura 109: Exemplo de Declaração das Variáveis de Ponto Duplo

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:

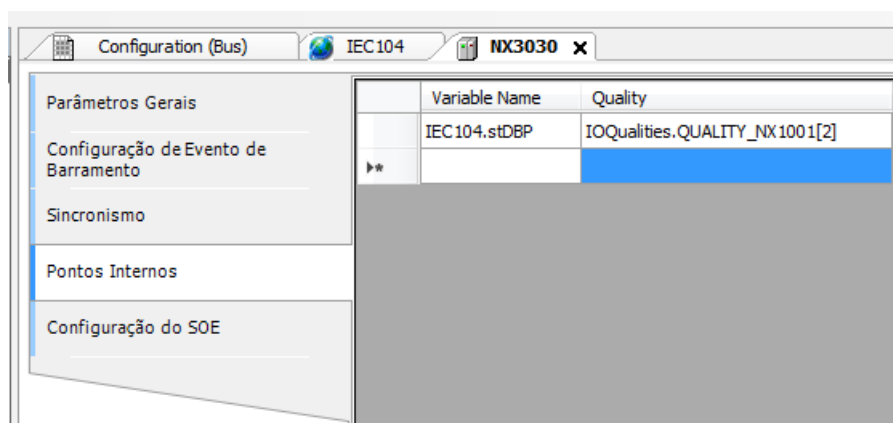


Figura 110: Atribuição das Variáveis de Ponto Duplo aos Pontos Internos

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.

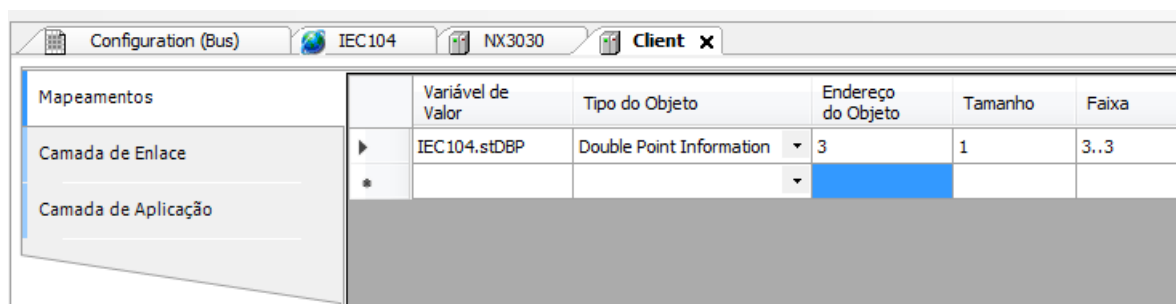


Figura 111: Mapeamento das variáveis de Ponto Duplo no Cliente IEC 60870-5-104

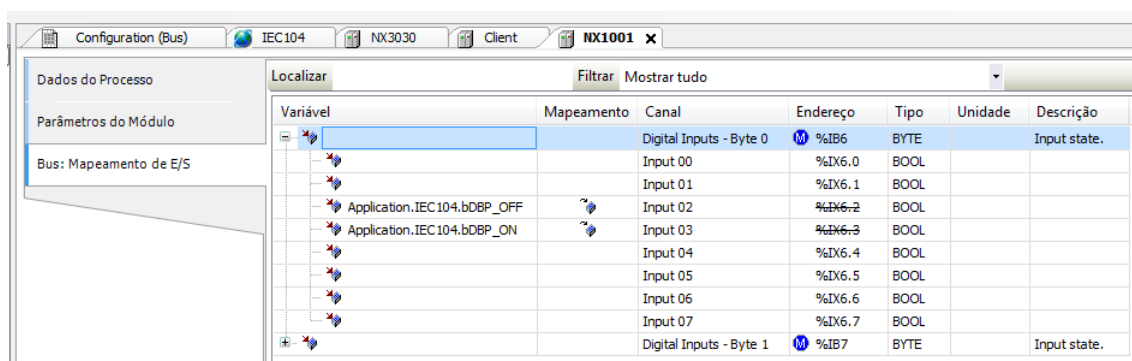


Figura 112: Mapeamento das Variáveis às Entradas do Módulo

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

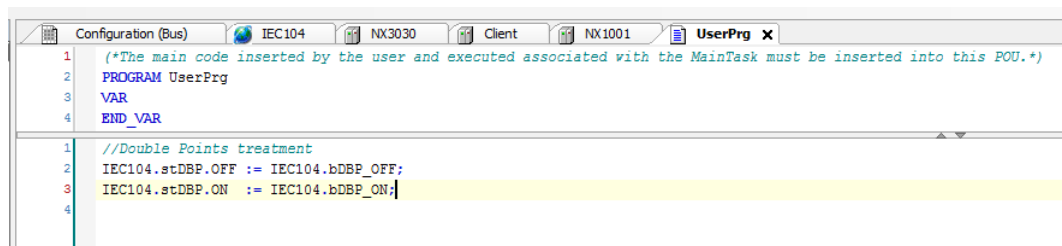


Figura 113: Atribuição dos Valores das Variáveis ao Ponto Duplo

5.6.14.2.2. 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 114: 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 pulse 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 pulse ON
byState:= byCmdType;
// Retorna comando iniciado
byStatus:= 5;
END_IF
ELSE
// Retorna pulso fora da faixa
byStatus:= 2;
END_IF

102: // Executar pulse 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 pulse 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;

```

5.6.14.3. 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:

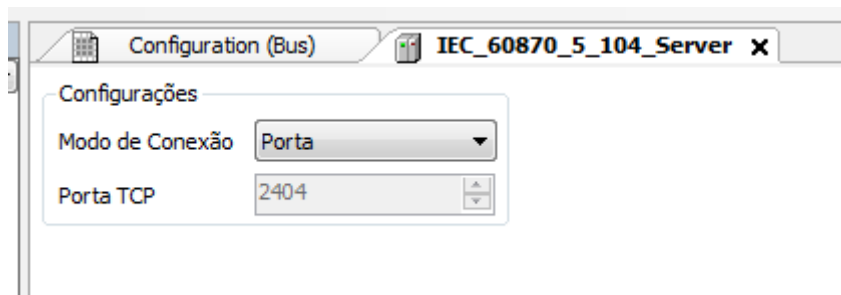


Figura 115: Tela de Parâmetros Gerais do Servidor IEC 60870-5-104

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 135: Configuração dos Parâmetros Gerais do Servidor IEC 60870-5-104

5.6.14.4. 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:

	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	*										
Camada de Aplicação											

Figura 116: 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 136: 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 137: 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.

5.6.14.5. 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:

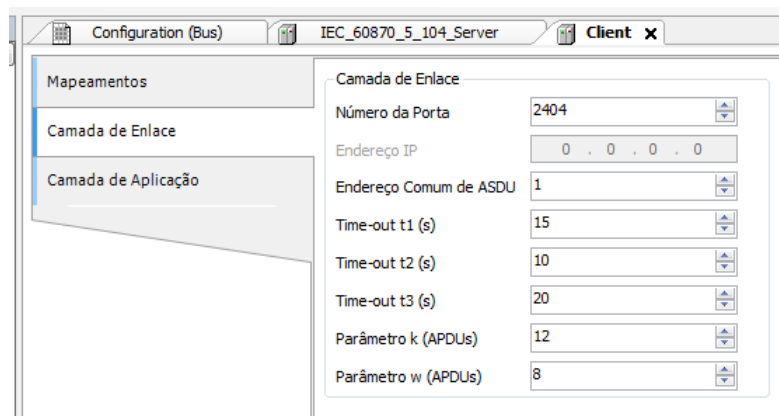


Figura 117: Tela de Configuração da Camada de Enlace do Servidor IEC 60870-5-104

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 138: 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.

5.6.14.6. 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 118: 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 139: 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 140: 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 (freze) 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 (freze) 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 141: 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).

5.6.14.7. 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 142: Diagnósticos do Servidor IEC 60870-5-104

5.6.14.8. 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 143: 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*.

5.6.15. CANOpen Manager

O CANOpen é um protocolo baseado no barramento CAN que fornece rápida atualização de E / S (cerca de 5 ms para uma rede de 1000 kbit/s com alguns escravos) com uma infraestrutura de barramento físico de par trançado simples.

O CANOpen Manager (mestre) é responsável por controlar os dispositivos escravos, gerenciar seu estado de operação e trocar E/S e outros dados de serviço. Por padrão, as atividades do protocolo do CANOpen Manager (ciclo de barramento) são executadas no contexto da MainTask, mantendo-o sincronizado com a execução do código da aplicação.

A configuração da rede CANOpen é realizada com o suporte de arquivos EDS, que descreve os objetos de dados e serviços de E/S (PDO e SDO) do escravo e deve ser fornecida pelo fabricante do dispositivo.

Além disso, uma biblioteca de aplicação chamada CiA405 é fornecida com Blocos Funcionais que permitem executar várias ações específicas, como alterar o estado escravo (NMT), receber objetos de emergência, consultar o estado escravo e executar comandos de leitura/gravação SDO. A descrição completa da biblioteca CiA405 pode ser encontrada na *Ajuda On-line* (F1) do Mastertool.

ATENÇÃO

- Somente uma instância do CANOpen Manager por projeto é permitida
- A especificação CANOpen permita até 127 nós (incluindo o Manager).

Um cuidado especial deve ser tomado considerando o comprimento do barramento físico e a taxa de transmissão selecionada. A tabela a seguir mostra o comprimento máximo de barramento que pode ser usado com segurança com uma determinada taxa de transmissão (Baudrate):

Baudrate	Comprimento Máximo do Barramento
1000 kbit/s	25 m
500 kbit/s	100 m
250 kbit/s	250 m
125 kbit/s	500 m
100 kbit/s	700 m
< 50 kbit/s	1000 m

Tabela 144: Baudrate X Comprimento do Barramento

5.6.15.1. Instalando e inserindo dispositivos CANopen

A configuração de uma rede CANopen usa o mesmo procedimento padrão de outras configurações de redes de campo no Mastertool.

Para adicionar um CANOpen Manager, clique com o botão direito do mouse na interface CAN e selecione *Acrescentar Dispositivo*. Expanda os itens até encontrar o dispositivo *CANopen_Manager* e clique no botão *Acrescentar*. O dispositivo *CANopen_Manager* aparecerá abaixo da interface CAN, conforme mostrado na figura a seguir:

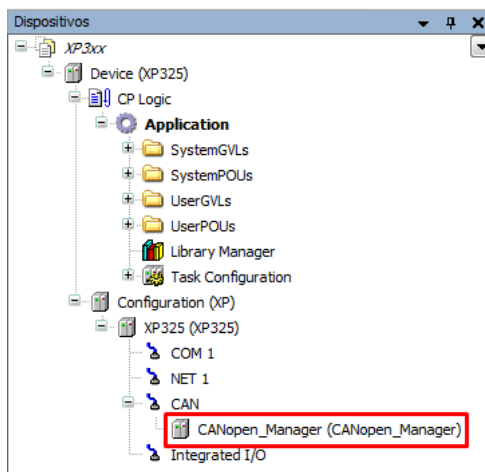


Figura 119: Adicionando o CANopen Manager

Para adicionar um dispositivo escravo CANopen, primeiro você precisa instalá-lo no Repositório de dispositivos. Para fazer isso, vá em *Ferramentas -> Repositório de Dispositivos* e instale o arquivo EDS do dispositivo.

Depois disso, clique com o botão direito do mouse no dispositivo *CANopen_Manager* e clique em *Acrescentar Dispositivo*. Pesquise os dispositivos que você deseja e clique no botão *Acrescentar*, como mostrado na imagem a seguir:

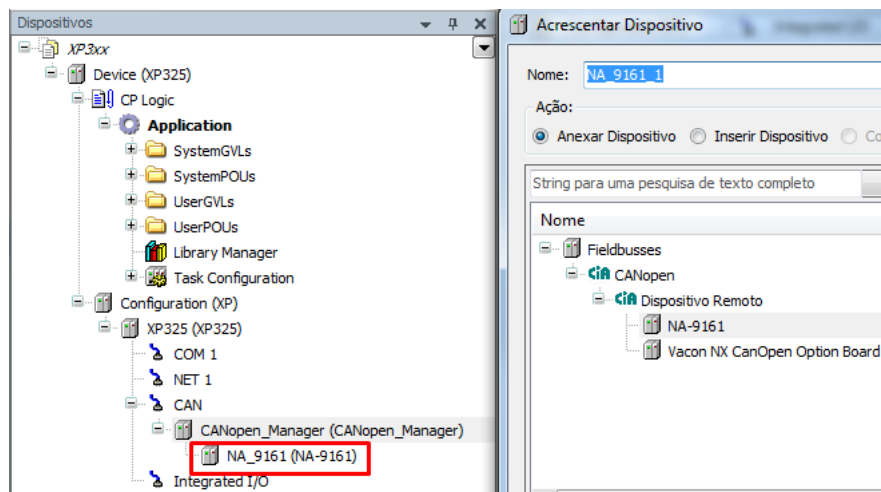


Figura 120: Adicionando um Dispositivo CANOpen Escravo

5.6.15.2. Configuração do CANOpen Manager

O CANOpen Manager vem com uma configuração pronta para uso (valores padrão). Normalmente, é necessário apenas definir a taxa de transmissão correta e o endereço do escravo para ter uma rede em execução.

Os principais parâmetros do CANOpen Manager estão localizados na guia *General*:

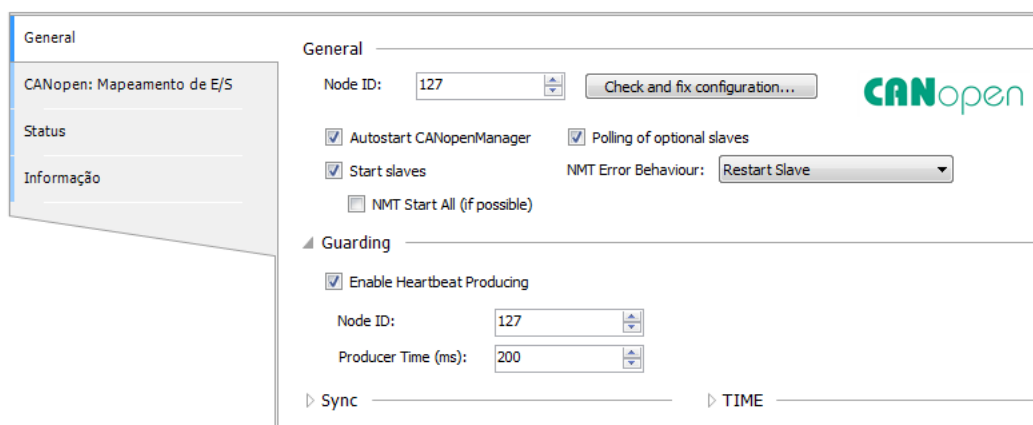


Figura 121: Parâmetros gerais do CANOpen Manager

A descrição detalhada dos parâmetros gerais do CANOpen Manager pode ser encontrada na seção *Editores de Dispositivos* -> *CANopen* da Ajuda Online do Mastertool (FI).

Além disso, a guia *CANopen: Mapeamento de E/S* permite alterar a tarefa do ciclo de barramento:

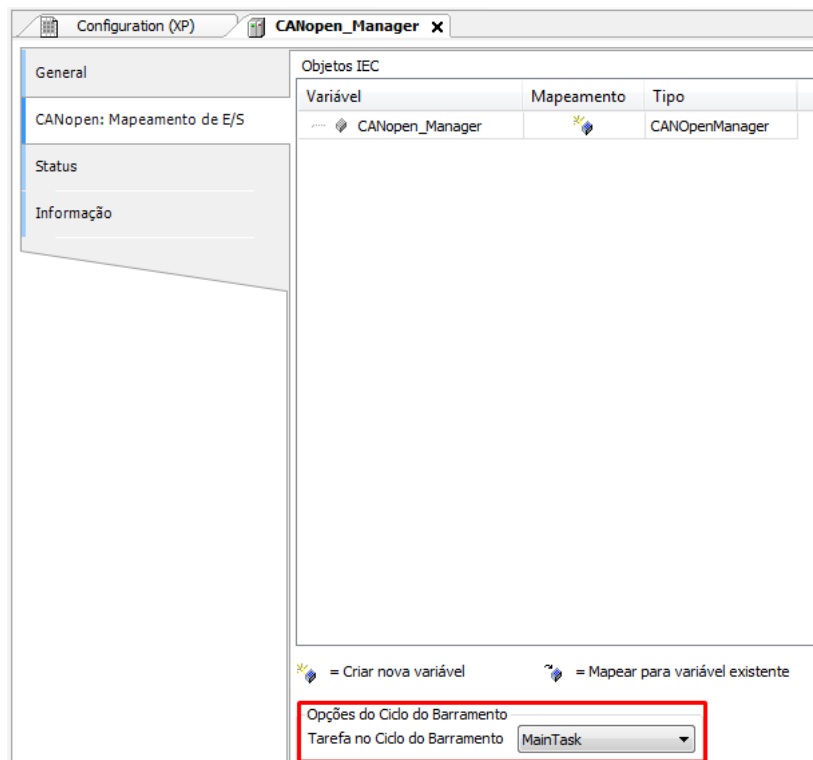


Figura 122: Configuração da tarefa cíclica do barramento CANopen

Por padrão, a tarefa de ciclo de barramento é configurada para usar a MainTask. Essa é a configuração recomendada para a maioria das aplicações. Alterar essa configuração é necessário apenas em um cenário muito específico, que requer a implementação de um loop de controle de tempo crítico usando E/S CANopen (5ms, digamos) que não pode ser executado na MainTask devido ao código pesado da aplicação.

5.6.15.3. Configuração de Escravo CANopen

A configuração dos Dispositivos Remotos CANopen (escravos) é separada nas quatro primeiras guias, como mostrado na figura a seguir:

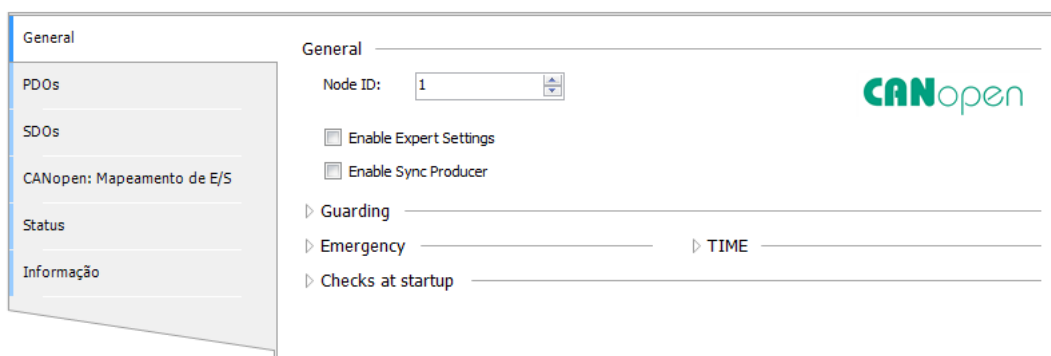


Figura 123: Parâmetros de Escravo CANopen

A guia *General* contém o endereço do escravo (Node ID), Nodeguarding e Emergency.

A guia *PDO* contém a configuração dos dados do processo (dados de E / S) que serão trocados.

A guia *SDO* contém os objetos SDO que podem ser selecionados para serem acessados pelo Bloco Funcional de leitura/-gravação SDO fornecido pela biblioteca CiA405.

A descrição detalhada dos parâmetros do Escravo CANopen pode ser encontrada na seção *Editores de Dispositivos* -> *CANopen* da Ajuda Online do Mastertool (F1).

5.6.16. PROFINET Controller

Para a correta utilização do protocolo PROFINET Controller é necessário consultar o manual MU214621 - Nexto Series PROFINET Manual .

5.7. Desempenho de Comunicação

5.7.1. 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: $\text{Tempo Máximo de Resposta} = 3 * (\text{tempo de ciclo}) + (\text{tempo de execução das tarefas}) + (\text{tempo interframe chars}) + (\text{tempo de atraso do envio})$.

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.

5.7.1.1. Interfaces Integradas 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 integrada da UCP, em função do tempo de ciclo configurado para a tarefa e do número de conexões ativas efetuando requisições:

Número de Conexões	Requisições respondi- das por segundo com tempo de ciclo da tarefa MODBUS em 5 ms	Requisições respondi- das por segundo com tempo de ciclo da tarefa MODBUS em 10 ms	Requisições respondi- das por segundo com tempo de ciclo da tarefa MODBUS em 20 ms
1 Conexão	192	100	50
2 Conexões	390	200	100
4 Conexões	780	400	200
7 Conexões	1354	700	350
10 Conexões	1922	998	500

Tabela 145: Taxa de Comunicação de um Servidor MODBUS em uma Interface Integrada

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.

5.7.1.2. Interfaces Remotas

O desempenho de um dispositivo MODBUS Servidor em uma interface Ethernet remota é parecido com o desempenho nas interfaces integradas 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 integradas. Portanto, para uma interface Ethernet remota teremos as seguintes formas para calcular seu desempenho:

Para $T \leq 55 \text{ ms}$ usa-se:

$$N = C \times (18,18 - (18,18 / (0,055 \times 1000)))$$

E para $T \geq 55 \text{ ms}$ 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).

5.7.2. 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.

5.7.3. 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

5.8. 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, 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, 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.

5.8.1. 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 146: 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.

5.8.2. 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](#).

5.9. 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:

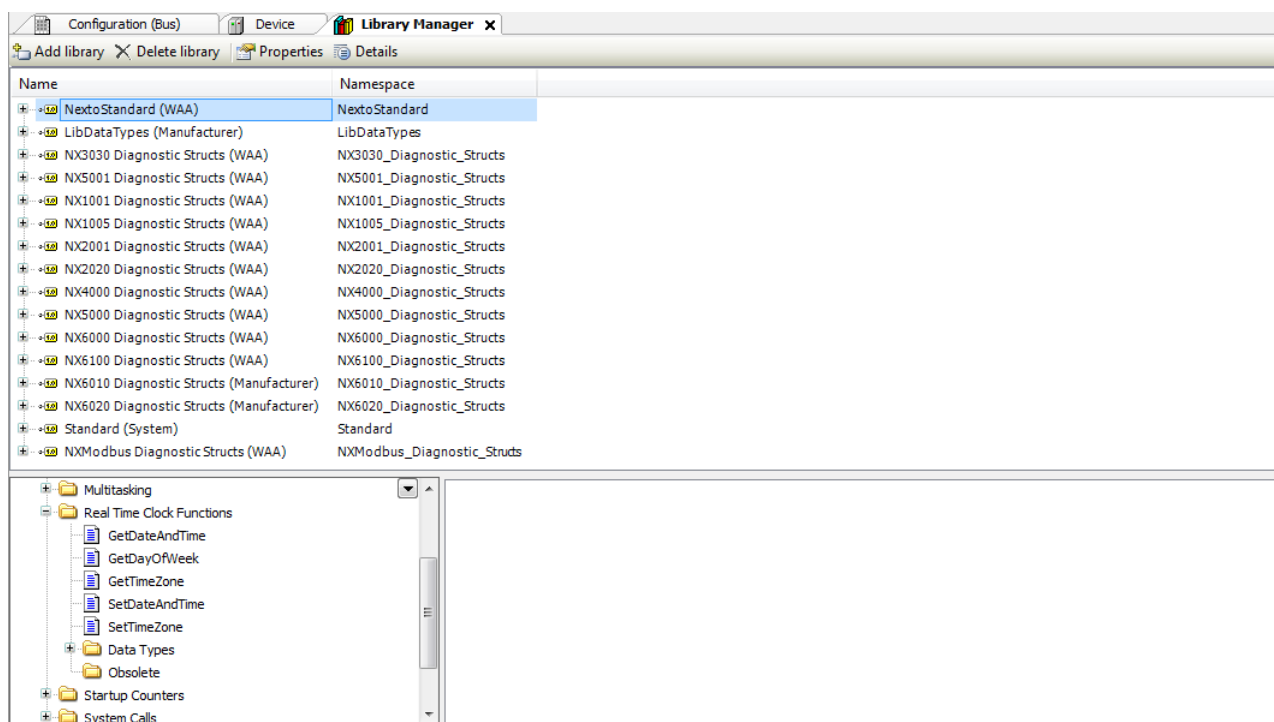


Figura 124: Blocos e Funções de Escrita e Leitura do Relógio

ATENÇÃO

Blocos funcionais para Leitura e Escrita do RTC, disponíveis em versões anteriores a 2.00 do Mastertool tornaram-se obsoletos da versão 2.00 em diante, os blocos que ficaram obsoletos são:

NextoGetDateAndTime, *NextoGetDateAndTimeMs*, *NextoGetTimeZone*, *NextoSetDateAndTime*, *NextoSetDateAndTimeMs* e *NextoSetTimeZone*.

5.9.1. 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.

5.9.1.1. Funções de Leitura do RTC

A leitura do relógio pode ser feita através das funções a seguir:

5.9.1.1.1. *GetDateAndTime*

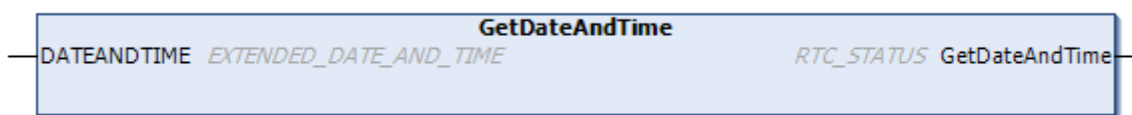


Figura 125: Leitura da Data e Hora

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 156.

Tabela 147: Parâmetros de Entrada GetDateAndTime

Parâmetros de saída	Tipo	Descrição
GETDATEANDTIME	RTC_STATUS	Retorna o estado de erro da função, ver Tabela 158.

Tabela 148: Parâmetros de Saída GetDateAndTime

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

```


5.9.1.1.2. *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:

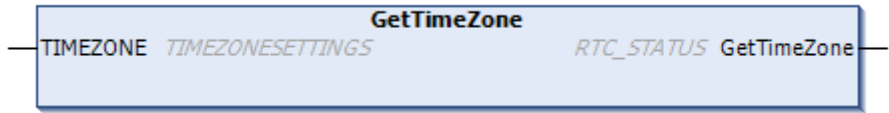


Figura 126: Leitura das Configurações de Fuso Horário

Parâmetros de entrada	Tipo	Descrição
TIMEZONE	TIMEZONESETTINGS	Essa variável apresenta a leitura das configurações de fuso horário.

Tabela 149: Parâmetros de Entrada GetTimeZone

Parâmetros de saída	Tipo	Descrição
GetTimeZone	RTC_STATUS	Retorna o estado de erro da função, ver Tabela 158.

Tabela 150: Parâmetros de Saída GetTimeZone

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
```

5.9.1.1.3. *GetDayOfWeek*

A função *GetDayOfWeek* é utilizada para fazer a leitura do dia da semana.

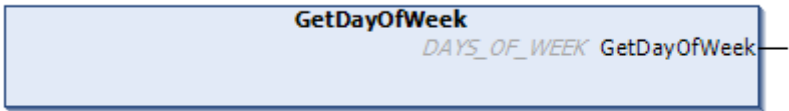


Figura 127: Leitura do Dia da Semana

Parâmetros de saída	Tipo	Descrição
GetDayOfWeek	DAYS_OF_WEEK	Retorna o dia da semana. Ver Tabela 157.

Tabela 151: 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();
```

5.9.1.2. Funções de Escrita do RTC

As configurações de relógio são feitas através das funções e blocos funcionais a seguir:

5.9.1.2.1. SetDateAndTime

O Bloco Funcional *SetDateAndTime* é utilizado para realizar o ajuste do relógio. Tipicamente a precisão é da ordem de centenas de milissegundos.

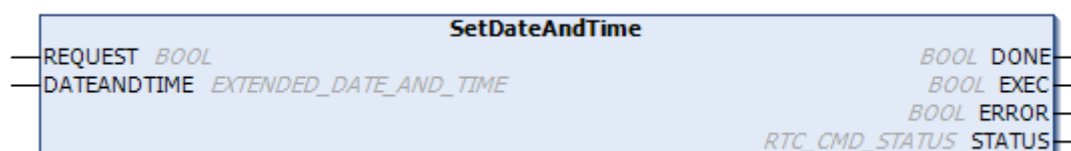


Figura 128: Ajuste de Data e Hora em Milissegundos

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 156.

Tabela 152: Parâmetros de Entrada SetDateAndTime

Parâmetros de saída	Tipo	Descrição
DONE	BOOL	Essa variável, quando verdadeira, indica que a ação foi completada.

Parâmetros de saída	Tipo	Descrição
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_STATUS	Retorna o erro ocorrido durante a configuração, ver Tabela 158.

Tabela 153: 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 descrita na seção [Limites de Operação do RTC](#). 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.

5.9.1.2.2. SetTimeZone

A função *SetTimeZone* realiza a escrita do ajuste de fuso horário:

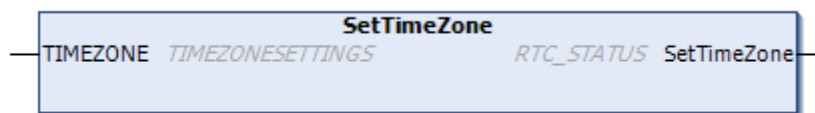


Figura 129: Ajuste de Fuso Horário

Parâmetros de entrada	Tipo	Descrição
TIMEZONE	TIMEZONESETTINGS	Estrutura com o valor de fuso horário a ser configurado. Ver Tabela 159.

Tabela 154: Parâmetros de Entrada SetTimeZone

Parâmetros de saída	Tipo	Descrição
SetTimeZone	RTC_STATUS	Retorna o erro ocorrido durante a leitura/configuração. Ver Tabela 158.

Tabela 155: 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 faixa descrita na seção [Limites de Operação do RTC](#), 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](#).

5.9.2. 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:

5.9.2.1. 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 156: Estrutura EXTENDED_DATE_AND_TIME

5.9.2.2. 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 157: Estrutura DAYS_OF_WEEK

5.9.2.3. 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 158: RTC_STATUS

5.9.2.4. 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 159: 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.

5.9.3. Limites de Operação do RTC

Os valores apresentados na [Tabela 160](#) indicam o menor e o maior instante de tempo que o relógio RTC consegue representar. O valor mínimo corresponde à data e hora mais antiga que pode ser configurada, enquanto o valor máximo representa a data e hora mais futura válida. Esses limites devem ser respeitados ao definir ou ler o relógio do produto, garantindo que todas as funções de data e hora operem corretamente.

Parâmetro	Mínimo	Máximo
Dia	1	31
Mês	1	12
Ano	2000	2099
Hora	0	23
Minuto	0	59
Segundo	0	59
Milissegundo	0	999

Tabela 160: Faixa de valores válidos para o Relógio RTC

5.10. 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 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.

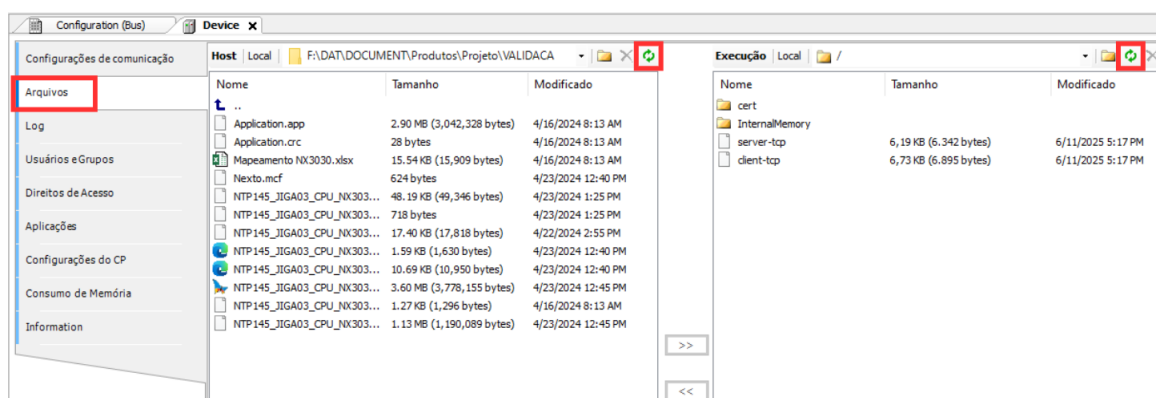


Figura 130: Acesso aos Arquivos de Usuário

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.

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.

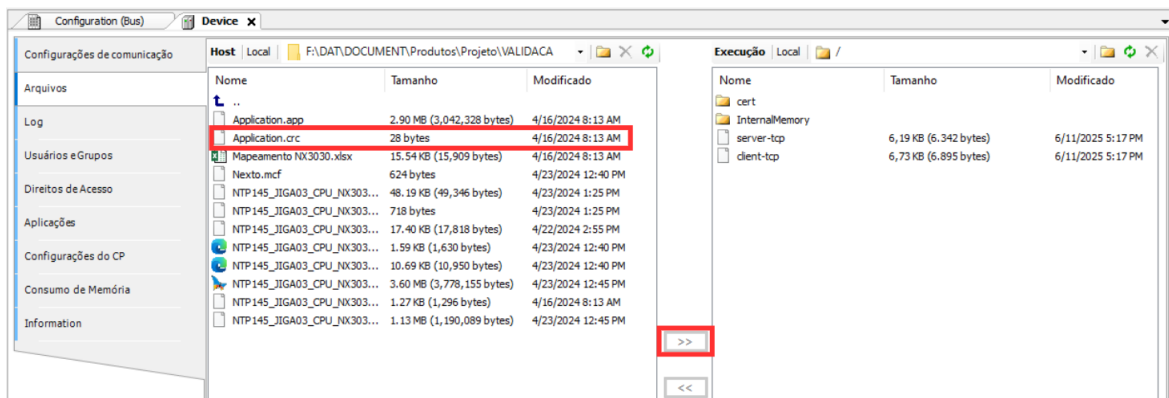


Figura 131: Transferindo Arquivos

ATENÇÃO

Os arquivos contidos dentro da pasta de um projeto criado pela ferramenta Mastertool 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 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, os arquivos presentes na memória de usuário e no microcomputador.

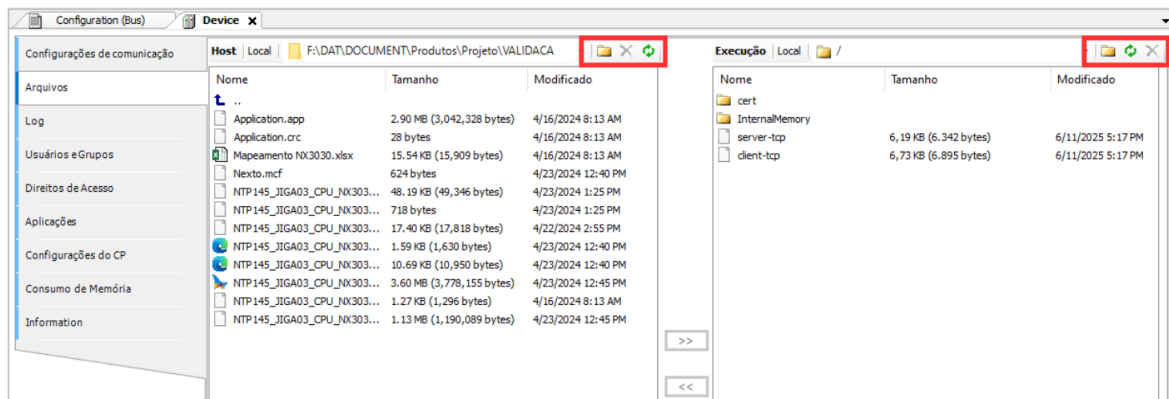


Figura 132: Opções de Utilização

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.

5.11. 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
	ESPANOL	>ESPANOL	Configurável
	VOLTAR	-	Retorna Nível
REDE	END. IP NET 1	-	Informativo
	MASCARA NET 1		Informativo
	VOLTAR		Retorna Nível
SOFTWARE	FIRMWARE	-	Informativo
	BOOTLOADER		Informativo
	PROC. AUX.		Informativo
	VOLTAR		Retorna Nível
CARTAO DE MEM.	DESMONTAR	-	Configurável
	FORMATAR	CONFIRMA?	Configurável
	VOLTAR	-	Retorna Nível
VOLTAR	-	-	Retorna Nível

Tabela 161: Níveis do Menu da UCP

Cartão de Memória: O cartão de memória somente estará disponível no menu caso o mesmo esteja inserido na UCP Nexto.

Conforme já mostrou a Tabela 161, 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
- 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)
- 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)

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 a troca do *Idioma*, 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.

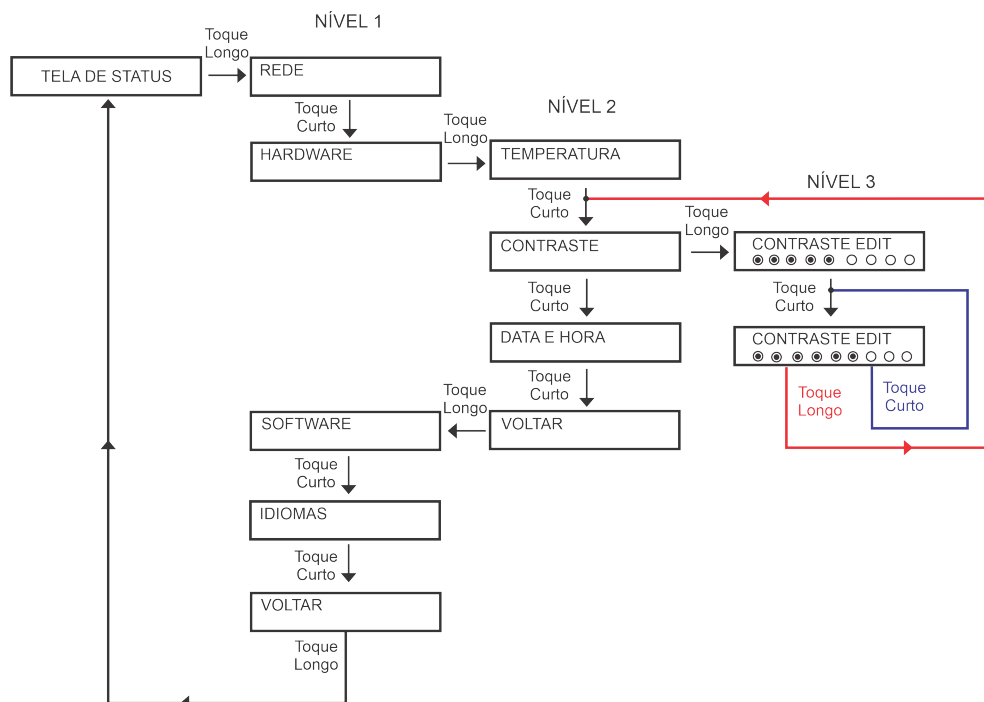


Figura 133: Ajuste do Contraste

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. 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 (consecutivos 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	BAUD2400	Lista todas as possibilidades da taxa de transmissão (bits por segundo)
	BAUD4800	
	BAUD9600	
	BAUD19200	
	BAUD38400	
	BAUD57600	
	BAUD115200	
SERIAL_DATABITS	DATABITS_7	Lista todas as possibilidades de bits de dado.
	DATABITS_8	
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.
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.
	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. Valores válidos são: 1, 4, 8 ou 14.
	MODE	Definido em SERIAL_MODE.
SERIAL_PARITY	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.
	PARITY_NONE	Lista todas as possibilidades de paridade.
	PARITY_ODD	
	PARITY_EVEN	
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, como conversores USB serial).
SERIAL_RX_CHAR_ EX-TENDED	Define um caractere da fila RX modo estendido.	
	RX_CHAR	Byte de dado.
	RX_ERROR	Código de erro.
	RX_TIMESTAMP	Silêncio devido ao caractere anterior ou devido a outro evento que aconteceu antes desse caractere (Para mais informações ver notas da seção COM 1).
	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):	
	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).

Tipo de dado	Opção	Descrição
SERIAL_RX_QUEUE_STATUS	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.
SERIAL_STATUS	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
	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.

Tipo de dado	Opção	Descrição
	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 integradas.)
	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.
SERIAL_STOPBITS	STOPBITS_1	Lista todas as possibilidades de bits de parada.
	STOPBITS_2	
	STOPBITS_1_5	

Tabela 162: 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 134: 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 163: 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.

Parâmetros de saída	Tipo	Descrição
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 164: 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 := RTS,
ENABLE_DCD_EVENT := FALSE,
ENABLE_CTS_EVENT := FALSE,
UART_RX_THRESHOLD := 8,
MODE := NORMAL_MODE,
ENABLE_RX_ON_TX := FALSE);
Status: SERIAL_STATUS;
END_VAR
//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 135: 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 165: 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.
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 166: 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_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.

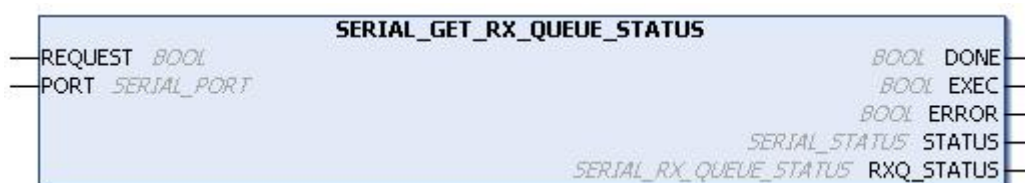


Figura 136: Bloco para Visualizar o Status da Fila RX

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 167: 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.

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
RXQ_STATUS	SERIAL_RX_QUEUE_STATUS	Retorna status/erros da fila RX, conforme descrito no tipo de dado SERIAL_RX_QUEUE_STATUS.

Tabela 168: 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.

```

5.12.1.4. 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 169: 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 170: 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.

```

5.12.1.5. 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).
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 171: 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 172: 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;

```

5.12.1.6. 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 da Interface Serial](#).



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.
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).

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 173: 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
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 174: 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;
```

5.12.1.7. 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.

Parâmetros de entrada	Tipo	Descrição
BREAK	BOOL	Caso seja verdadeiro, habilita lógica 0 (ocupado) na linha de transmissão.

Tabela 175: 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.
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 176: 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.

```

5.12.1.8. SERIAL_TX

Este bloco funcional é utilizado para transmitir um buffer de dados pela porta serial. Sua execução é finalizada após todos os bytes serem copiados para a fila interna de transmissão (TX) ou após a ocorrência de um timeout (quando erros são gerados).

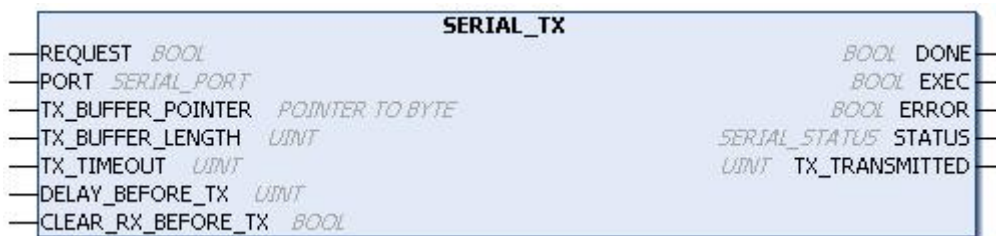


Figura 141: Bloco para Transmitir Valores pela SERIAL_TX

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 177: 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.
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 178: 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, 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.

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.

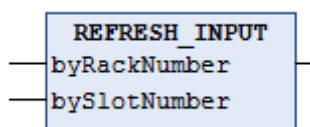


Figura 142: Bloco para Atualizar as Entradas

Parâmetros de entrada	Tipo	Descrição
byRackNumber	BYTE	Número do bastidor.
bySlotNumber	BYTE	Número da posição, na qual se encontra o módulo.

Tabela 179: Parâmetros de Entrada REFRESH_INPUT

Exemplo de utilização em Linguagem ST:

```

PROGRAM UserPrg
VAR
Info: ERRORCODE;
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.

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.

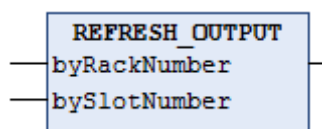


Figura 143: Bloco para Atualizar as Saídas

Parâmetros de entrada	Tipo	Descrição
byRackNumber	BYTE	Número do bastidor.
bySlotNumber	BYTE	Número da posição, na qual se encontra o módulo.

Tabela 180: Parâmetros de Entrada REFRESH_OUTPUT

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
Info: ERRORCODE;
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), 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 (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 181: 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 182: Parâmetros de Saída TOF_RET

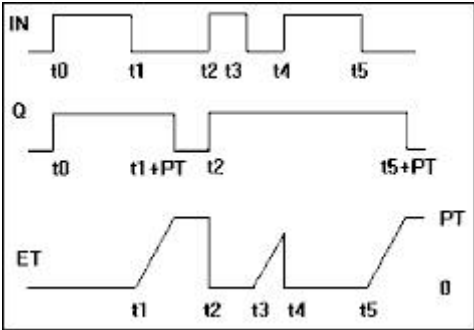


Figura 145: Comportamento Gráfico do Bloco Funcional TOF_RET

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 183: 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 184: Parâmetros de Saída TON_RET

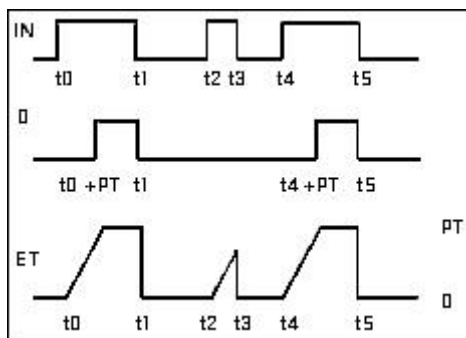


Figura 147: Comportamento Gráfico do Bloco Funcional TON_RET

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 185: 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 186: Parâmetros de Saída TP_RET

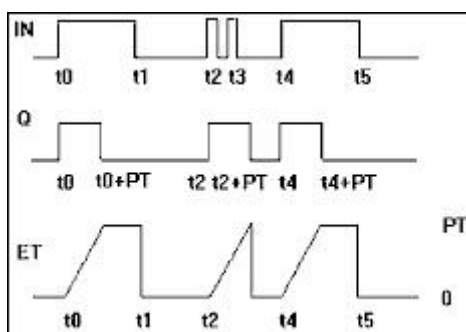


Figura 149: Comportamento Gráfico do Bloco Funcional TP_RET

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. 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 187: Tipos de dados para Log de Usuário

A seguir, são descritas as duas funções disponíveis na biblioteca *LibLogs* no Mastertool. 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.

5.12.5.1. 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 187. 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:

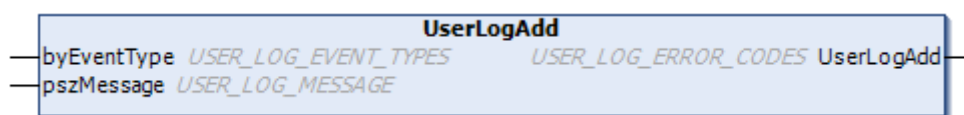


Figura 150: Função UserLogAdd

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 <i>USER_LOG_EVENT_TYPES</i> .
pszMessage	<i>USER_LOG_MESSAGE</i>	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 188: 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 : LibLogs.USER_LOG_ERROR_CODES;
  sMessage : LibLogs.USER_LOG_MESSAGE;
  m_rTemperature : REAL;
  MAX_TEMPERATURE_ACCEPT : REAL;
END_VAR

IF (m_rTemperature > MAX_TEMPERATURE_ACCEPT) THEN
  sMessage := 'Temperatura acima do esperado: ';
  sMessage := Standard.CONCAT(sMessage, REAL_TO_STRING(m_rTemperature));
  sMessage := Standard.CONCAT(sMessage, '°');
  eLogError := LibLogs.UserLogAdd(LibLogs.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°
```

5.12.5.2. 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*.

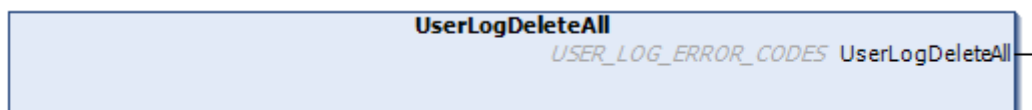


Figura 151: Função UserLogDeleteAll

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
  eLogError : LibLogs.USER_LOG_ERROR_CODES;
  m_DeleteLogs: BOOL;
END_VAR

IF (m_DeleteLogs = TRUE) THEN
  //Variável eLogError recebe possíveis erros da função.
  eLogError := LibLogs.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.

5.12.6. ClearRtuDiagnostic

Esta função não é suportada por esta série de UCPs.

5.12.7. 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.

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 189: Resultados da função ClearEventQueue

Exemplo de utilização em linguagem ST, onde a chamada da função irá limpar a fila de eventos e, consequentemente, 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. Páginas Web de Usuário

Também chamado de *Web Visualization*, ou simplesmente *Webvisu*, esse recurso permite implementar um SCADA simplificado incorporado ao controlador. As telas de visualização são desenvolvidas no mesmo ambiente da aplicação controlador usando o Mastertool. Após o download da aplicação, o controlador inicia um servidor da web que hospeda esta página da web especial. As informações completas sobre essa funcionalidade podem ser encontradas na Ajuda do Mastertool.

5.14. Página Web de Sistema

A *Página Web de Sistema* dos controladores da Série Nexto apresenta informações de status e configurações dos controladores, que serão descritas nas próximas seções.

5.14.1. Informações Gerais

Para acessar a Página Web de Sistema da UCP desejada, basta utilizar um navegador para desktop e digitar, na barra de endereço, o endereço IP correspondente à UCP (Ex.: <http://192.168.15.1>). Inicialmente, serão apresentadas as informações gerais da UCP, conforme mostra a figura abaixo.



A interface web de sistema Altus, versão 3.0.0.0, exibe a aba 'Informações Gerais'. No topo, há uma barra de navegação com o logotipo 'altus' e 'Nexto', e uma seção de informações de status que indica 'Informações de 2026-05-08 16:13:06 (UTC)' com um botão 'Atualizar'. A interface é dividida em três abas principais: 'Informações Gerais' (selecionada), 'Informações do Barramento' e 'Gerenciamento'. A aba 'Informações Gerais' contém uma tabela com os seguintes dados:

Informações Gerais	
Modelo	NX3000
Tag	NX3000
Descrição	
Bastidores Configurados	1
Bastidor	0
Posição	0
Versão de Firmware	114.86.201
Versão do Bootloader	1.0.5.0
Estado do Sistema	Run (Executando)
Versão de MasterTool	4.0.0
Diagnósticos Ativos	0
Valores Forçados	Não
Dispositivo USB	Não encontrado

Figura 152: Tela Inicial

5.14.2. Informações do Barramento

Desenvolvida para apresentar informações e diagnósticos relacionados ao barramento utilizado pelo controlador. A aba Informações do Barramento permite ao usuário visualizar o estado dos módulos, identificar dispositivos conectados e acessar dados de diagnóstico e status de cada módulo. Os recursos disponíveis nesta aba variam de acordo com o tipo de barramento e as funcionalidades suportadas pelo controlador utilizado.

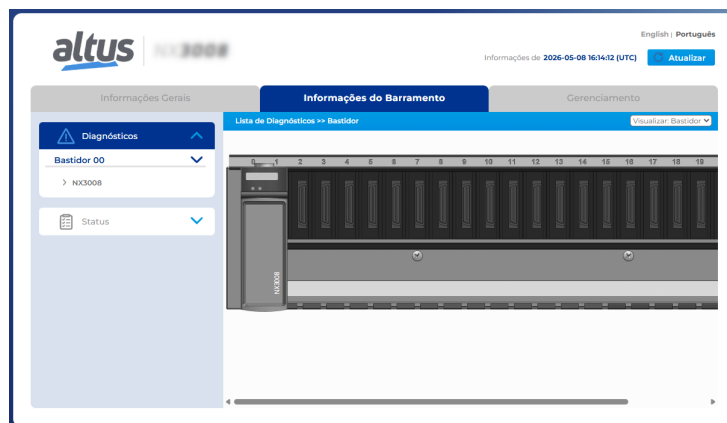


Figura 153: Informações do Barramento

Quando não houver nenhuma aplicação na UCP, não será exibida nenhuma imagem. Quando a visualização pelo Bastidor é utilizada, os módulos que têm diagnóstico ficam piscando e assumem a cor vermelha. Caso contrário será exibida uma lista com os módulos presentes no sistema, Tags correspondentes e número de diagnósticos ativos.

Ao clicar no módulo com diagnóstico, no mesmo instante são mostrados os diagnósticos ativos do módulo, conforme mostra a figura abaixo:

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.

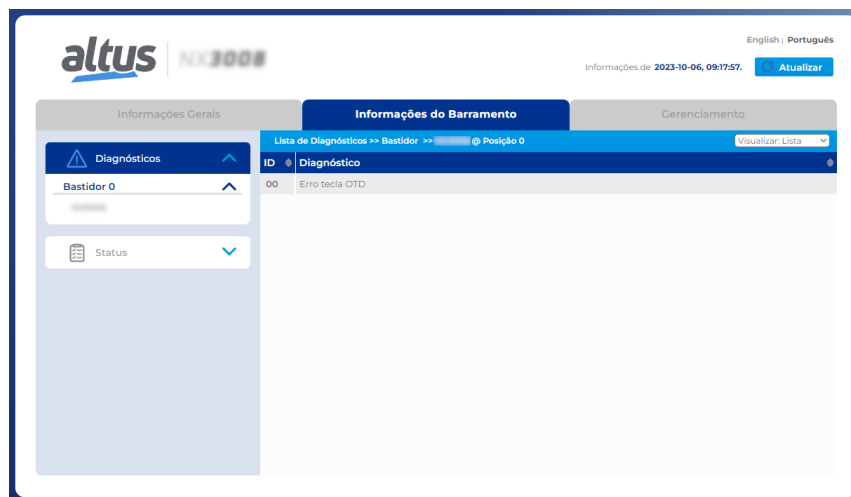


Figura 154: Diagnósticos do Sistema

Caso a seção Status seja selecionada, o estado de todos os diagnósticos detalhados é exibido na tela, conforme mostra a figura abaixo:

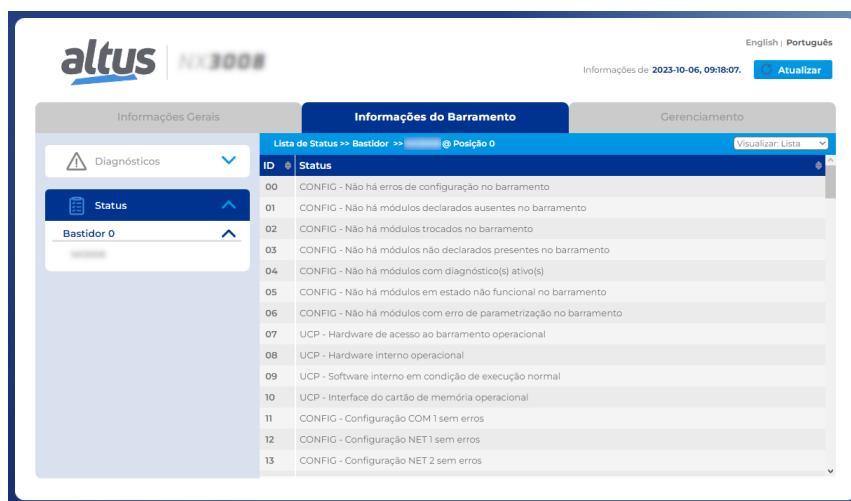


Figura 155: Estado do Sistema

O usuário pode optar por duas opções de idioma: Português e Inglês. Basta alterar na parte superior direita da tela para o idioma desejado.

5.14.3. Gerenciamento

Desenvolvida para realizar a configuração e o acesso aos diagnósticos de algumas funcionalidades. A aba *Gerenciamento* de Página Web de Sistema tem seu acesso protegido por usuário e senha, sendo *admin* o valor padrão para ambos os campos.

Na aba Gerenciamento, existem outros recursos como, por exemplo, *Sistema*, *Rede*, *Dispositivo USB*, *Firewall*, *OpenVPN* e *Servidor FTP*. Os recursos disponíveis nesta aba variam de acordo com as funcionalidades disponíveis para o controlador utilizado e somente podem ser acessados após o usuário realizar o Login, conforme mostra a figura abaixo.

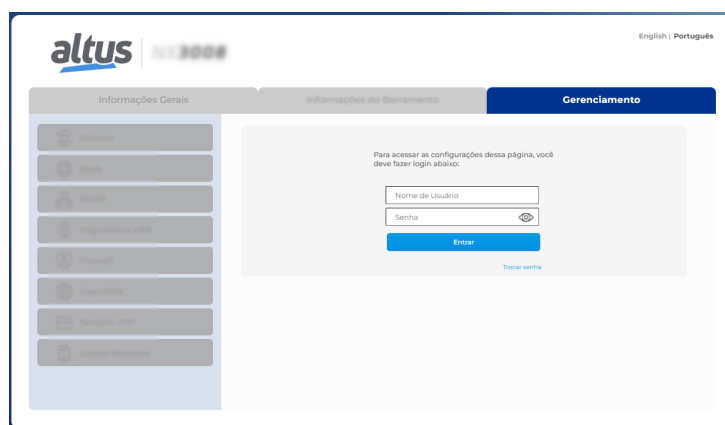


Figura 156: Acesso a aba de Gerenciamento

5.14.3.1. Sistema

Na seção de *Sistema* da aba *Gerenciamento* é possível realizar a atualização de firmware da UCP. Para casos em que a atualização é realizada remotamente (através de uma conexão de rádio ou satélite por exemplo), a velocidade mínima deste link deve ser de 128Kbps.

5.14.3.1.1. Atualização de Firmware

Na página *Sistema*, é possível realizar a atualização do firmware do controlador, permitindo que o usuário instale a versão desejada. A seção *Atualização de Firmware* exibe informações sobre o estado atual da aplicação do controlador, a versão do firmware instalada e o status do processo de atualização.

Para realizar a atualização, o usuário deve selecionar o arquivo de firmware correspondente. Opcionalmente, é possível definir se a aplicação será apagada após a atualização por meio da opção *Apagar Aplicação*. Após configurar as opções desejadas, o usuário deve clicar no botão *Iniciar Atualização*. Em seguida, será solicitada a confirmação da operação por meio da autenticação com o usuário e a senha configurados.

A versão mais recente do firmware está disponível no site da Altus. O acesso ao recurso de atualização requer autenticação prévia do usuário, conforme ilustrado na figura abaixo.

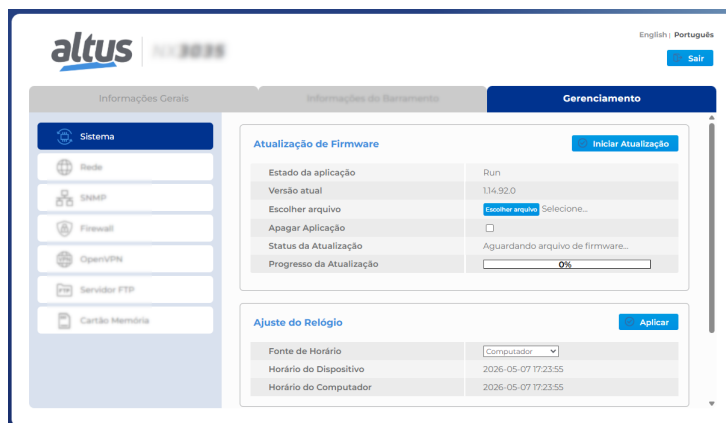


Figura 157: Atualização de Firmware

5.14.3.1.2. Ajuste do Relógio

Na Página Web de Sistema, é possível realizar o ajuste de relógio do controlador, que se encontra na Seção Sistema da aba Gerenciamento da Página. O formato de data e horário seguem a norma ISO 8601 para amostragem de data e horário (AAAA/MM/DD hh:mm:ss), como mostra a imagem abaixo:

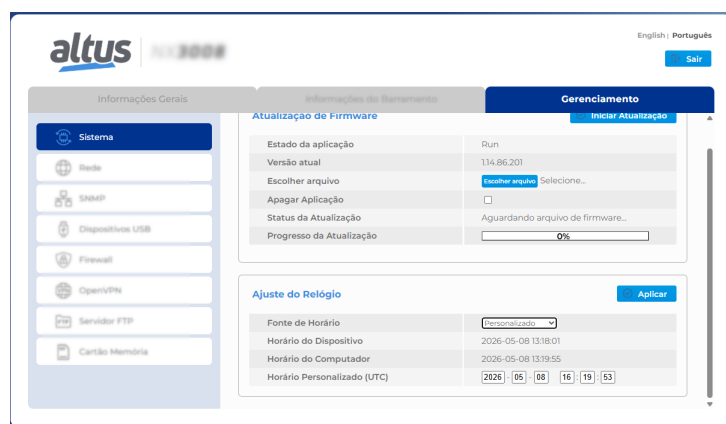


Figura 158: Ajuste do Relógio

Este recurso possui dois modos de ajuste do horário do dispositivo, que podem ser selecionados no item “Fonte de Horário”, disponibilizando ao usuário duas opções para sincronizar o relógio.

1. Horário do Computador (UTC)

No modo de horário do computador, o usuário pode aplicar o horário configurado em seu computador em UTC para o seu dispositivo. Para isso, deve-se selecionar no item “Fonte de Horário” a opção “Computador”. Após clicar no botão “Aplicar”, é necessário a validação das credenciais do dispositivo, então a UCP receberá a data e o horário UTC que estão configurados no computador.

2. Horário Personalizado (UTC)

No modo de horário personalizado, o usuário pode preparar um horário personalizado no padrão UTC para ser aplicado

na data e horário interna de seu dispositivo. Para isso, deve-se selecionar no item “Fonte de Horário” a opção “Personalizado”. Com o modo selecionado, o usuário deverá configurar a data e o horário desejado no item “Horário Personalizado (UTC)”, que será inicializada com o horário local do navegador. Então, após o usuário clicar no botão “Aplicar” e validar as credenciais do dispositivo, o mesmo estará com seu horário interno configurado com o horário configurado no item “Horário Personalizado (UTC)”. Para limites de configuração verificar a seção [Limites de Operação do RTC](#).

5.14.3.1.3. Controle do Sistema

Na página Web de Sistema, é possível realizar operações de controle do sistema diretamente através da interface web. A opção de Controle do Sistema fornece um ambiente dedicado para ações de gerenciamento do controlador, permitindo que o usuário execute o procedimento de reinicialização do dispositivo. Para isso, é necessário clicar no botão *Reinicializar*. Se o operador quiser apagar a aplicação existente, é necessário marcar a opção *Apagar Aplicação* antes de realizar a operação. O recurso requer autenticação do usuário antes que qualquer operação de controle do sistema possa ser executada, como mostrado na figura abaixo.

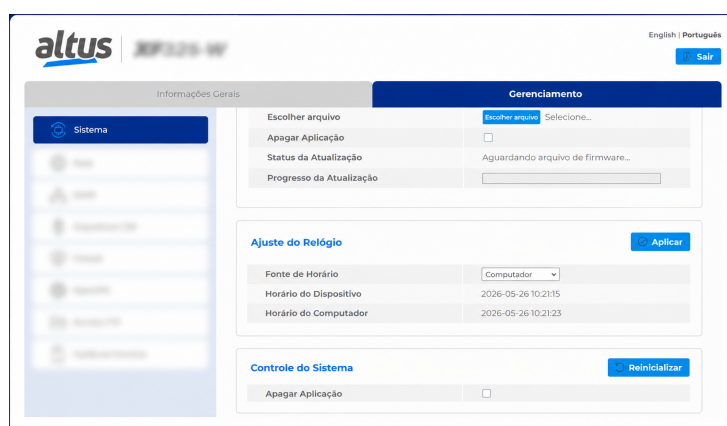


Figura 159: Controle do Sistema

5.14.3.2. Rede

Desenvolvida para auxiliar na usabilidade do controlador, a seção de Rede possibilita a alteração dos endereços de rede e a execução do Sniffer de rede. A quantidade de interfaces Ethernet apresentada é de acordo com a especificação de cada controlador.

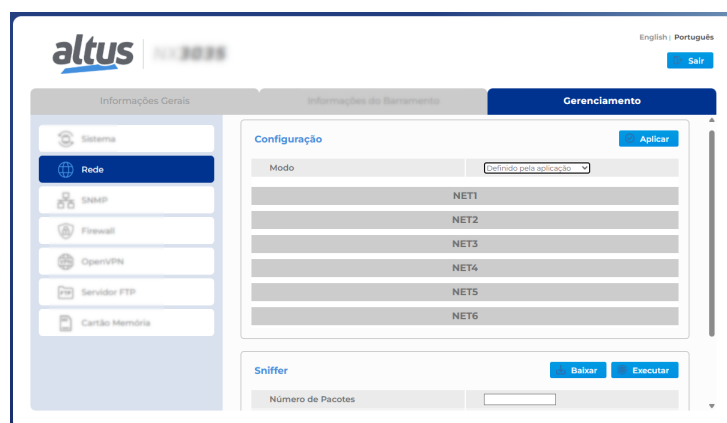


Figura 160: Página Web de Rede

5.14.3.2.1. Configuração

ATENÇÃO

Em um CP redundante, a configuração é definida pela aplicação e o campo do Modo está desabilitado.

1. Modo Definido pela aplicação

O campo Modo define qual configuração o controlador deve carregar para as suas interfaces. Este campo pode ser configurado como *Definido pela página web* ou *Definido pela aplicação*.

Quando configurado como *Definido pela aplicação*, a tabela de interfaces fica desabilitada, não permitindo alterações, conforme mostra a figura abaixo. Neste modo, as configurações aplicadas no controlador são as definidas pela aplicação. Abaixo uma imagem com o modo *Definido pela aplicação* selecionado, mostrando a tabela de interfaces desabilitada.

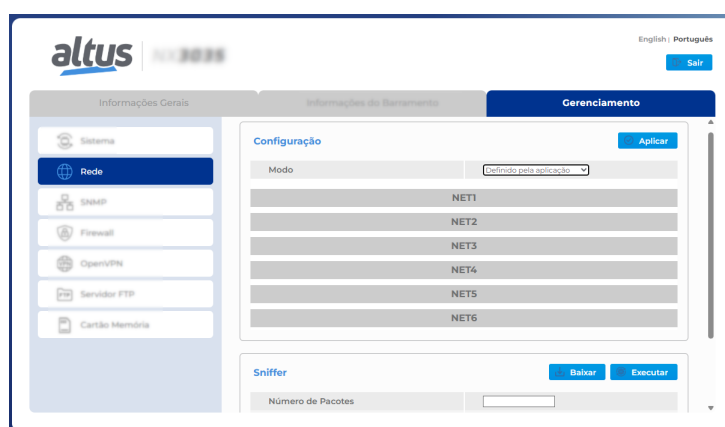


Figura 161: Tabela de Interfaces - Modo Aplicação

2. Modo Definido pela página web

Para o modo *Definido pela Página Web* a tabela de interfaces permanece habilitada, conforme mostra a figura abaixo.

Neste modo, o usuário pode configurar Endereço IP, Máscara de Rede e Gateway de cada uma das interfaces Ethernet disponíveis, além de poder habilitar e desabilitar quando possuir mais de uma interface Ethernet.

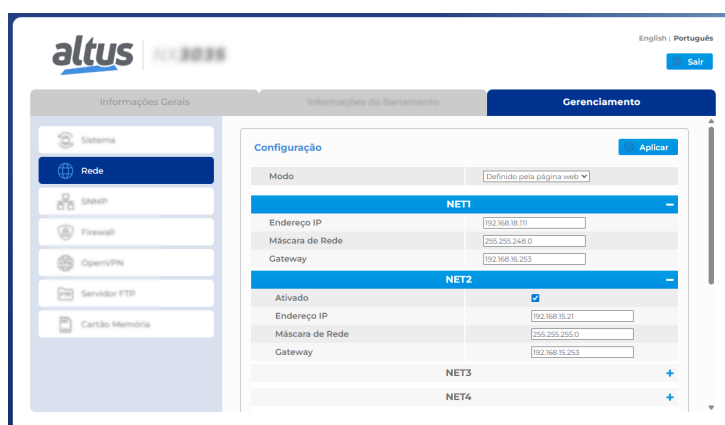


Figura 162: Tabela de Interfaces - Modo Web

O checkbox *Ativado* permite ao usuário habilitar e desabilitar as interfaces Ethernet. Este somente é disponível para ser marcado ou desmarcado quando a configuração for *Definido pela página web*.

Quando este checkbox estiver desmarcado, indica que aquela interface NET está desabilitada, ou seja, não receberá configuração e ficará desativada. Já quando a interface está habilitada (com o checkbox marcado), as configurações ficam disponíveis para edição.

Para que as configurações sejam aplicadas no controlador, basta clicar no botão *Aplicar*. Este processo verifica se houve algum erro na configuração realizada e, se for o caso, exibe uma mensagem na tela do navegador indicando o erro. Se as configurações estiverem corretas, após clicar em *Aplicar*, será exibida uma janela, no navegador, de confirmação para aplicar as novas configurações. Clicando em *OK*, as configurações são enviadas para o controlador e aplicadas.

ATENÇÃO

Ao realizar alterações de rede no controlador as interfaces serão reiniciadas, o que pode causar queda na comunicação. Especialmente ao ser alterado o valor do endereço IP.

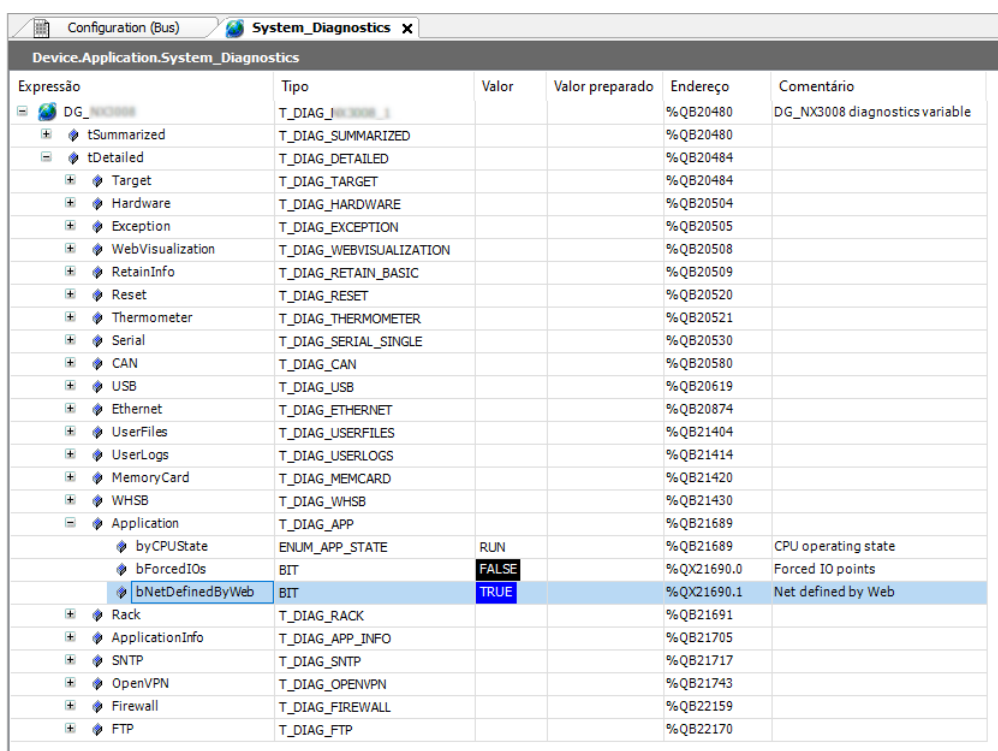
Ao aplicar as configurações utilizando o modo de operação *Definido pela Aplicação*, o controlador vai assumir as configurações que foram definidas pela aplicação já carregada nele anteriormente. Caso não haja nenhuma aplicação no controlador, a configuração atual será mantida, sendo alterado apenas o modo de configuração.

Utilizando o modo *Definido pela Página Web*, os endereços indicados na Seção de Rede serão carregados.

ATENÇÃO

O modo de operação *Definido pela Página Web* configura as interface para operarem em *Modo Simples*.

É possível monitorar pelo Mastertool se o endereço IP está configurado a partir da Página Web ou da aplicação a partir do BIT de diagnóstico *bNetDefinedByWeb* do grupo *Application*, que ira mudar para *TRUE* se o IP está configurado pela Página Web e em *FALSE* se configurado pela aplicação.



Expressão	Tipo	Valor	Valor preparado	Endereço	Comentário
DG_NX3008	T_DIAG_J			%QB20480	DG_NX3008 diagnostics variable
tSummarized	T_DIAG_SUMMARIZED			%QB20480	
tDetailed	T_DIAG_DETAILED			%QB20484	
Target	T_DIAG_TARGET			%QB20484	
Hardware	T_DIAG_HARDWARE			%QB20504	
Exception	T_DIAG_EXCEPTION			%QB20505	
WebVisualization	T_DIAG_WEBVISUALIZATION			%QB20508	
RetainInfo	T_DIAG_RETAIN_BASIC			%QB20509	
Reset	T_DIAG_RESET			%QB20520	
Thermometer	T_DIAG_THERMOMETER			%QB20521	
Serial	T_DIAG_SERIAL_SINGLE			%QB20530	
CAN	T_DIAG_CAN			%QB20580	
USB	T_DIAG_USB			%QB20619	
Ethernet	T_DIAG_ETHERNET			%QB20874	
UserFiles	T_DIAG_USERFILES			%QB21404	
UserLogs	T_DIAG_USERLOGS			%QB21414	
MemoryCard	T_DIAG_MEMCARD			%QB21420	
WHSB	T_DIAG_WHSB			%QB21430	
Application	T_DIAG_APP			%QB21689	
byCPUState	ENUM_APP_STATE	RUN		%QB21689	CPU operating state
bForcedIOs	BIT	FALSE		%QX21690.0	Forced IO points
bNetDefinedByWeb	BIT	TRUE		%QX21690.1	Net defined by Web
Rack	T_DIAG_RACK			%QB21691	
ApplicationInfo	T_DIAG_APP_INFO			%QB21705	
SNTP	T_DIAG_SNTP			%QB21717	
OpenVPN	T_DIAG_OPENVPN			%QB21743	
Firewall	T_DIAG_FIREWALL			%QB22159	
FTP	T_DIAG_FTP			%QB22170	

Figura 163: Diagnóstico - IP definido pela Página Web

5.14.3.2.2. Sniffer

O sniffer de rede, mostrado na figura abaixo, pode ser utilizado para observar o tráfego nas interfaces físicas, exceto para os dispositivos USB como modems e adaptadores Wi-Fi. Ele possui duas configurações básicas:

Número de Pacotes: É o número de pacotes que se deseja capturar. O valor configurado desse parâmetro deve estar dentro da faixa de 100 a 25000 pacotes;

Tempo Limite de Ocio (segundos): Se não houver nenhum tráfego de pacotes na interface após este tempo limite configurado, a execução do Sniffer é encerrada. Pode ser configurado com valores entre 1 e 3600 segundos.

Através da tabela *Interfaces*, o usuário pode selecionar em quais interfaces deseja executar o Sniffer, ou seja, fazer a análise da rede. Esta tela irá apresentar as interfaces de Ethernet específicas de cada controlador. Pode-se selecionar todas as interfaces disponíveis e executa-lo em todas elas, simultaneamente. Para interfaces desabilitadas, não é possível executar o Sniffer, caso a opção selecionada esteja desabilitada, será exibido um erro no navegador.

Somente após alguns instantes, depois da abertura da tela, o botão *Executar*, que inicia a execução do Sniffer, se tornará disponível. O botão *Baixar* será desbloqueado apenas se houver algum arquivo, referente ao Sniffer, disponível para ser baixado. Se o Sniffer nunca foi executado, ou o arquivo for excluído, o botão não estará disponível.

Ao executar o Sniffer de Rede, a página irá desabilitar os campos de edição, o botão *Baixar* será bloqueado e o botão *Executar*, se tornará o botão *Parar*, conforme mostra a figura exemplo abaixo.

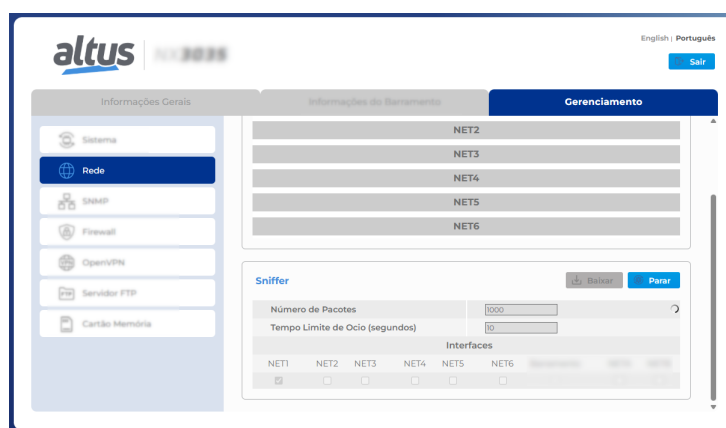


Figura 164: Sniffer de Rede Executando

O botão *Parar* pode ser utilizado para encerrar a execução do sniffer a qualquer momento após ter sido iniciada.

Para cada interface em que o Sniffer é executado, é gerado um arquivo **.pcap**. Esses arquivos são nomeados de acordo com o modelo do controlador e a interface analisada, seguindo o formato **[PLC_MODEL]_[INTERFACE].pcap**. Eles são armazenados dentro de um arquivo **.zip**, também nomeado de acordo com o modelo do controlador, seguindo o formato **[PLC_MODEL]_capture.zip**.

Ao final da execução do sniffer, é exibida uma mensagem questionando se deve ser realizado ou não o download automático dos arquivos gerados. Estes arquivos são armazenados na pasta *InternalMemory* da [Memória de Arquivos de Usuário](#) e podem ser acessado através do software de programação do controlador. O arquivo baixado é sempre o com a extensão **.zip**, que agrupa os demais.

Caso ocorra algum problema em relação a memória insuficiente, ocasionado em função da geração dos arquivos sniffer, será indicado para o usuário. Recomenda-se então executar o analisador outra vez, com uma configuração de *Número de Pacotes* menor.

O sniffer de rede pode encerrar a sua execução por três razões, falta de memória, estouro do tempo limite de ócio das interfaces e também o cancelamento manual.

5.14.3.3. SNMP

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.

5.14.3.3.1. SNMP nas UCPs Nexto

Os controladores 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, com suporte aos protocolos SNMPv1, SNMPv2c e SNMPv3, além de suporte a MIB-II, onde objetos são descritos na RFC-1213. A tabela abaixo contém os objetos disponíveis.

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.
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 190: 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 o controlador é iniciado. O acesso às informações do agente se dá através das interfaces Ethernet integradas na porta UDP 161. Portanto quando o serviço estiver ativo, as informações do agente poderão ser acessadas através de qualquer uma das interfaces Ethernet integradas, conforme a disponibilidade do controlador em uso. Na figura abaixo é mostrado um exemplo de gerente SNMP, no qual são lidos alguns valores.

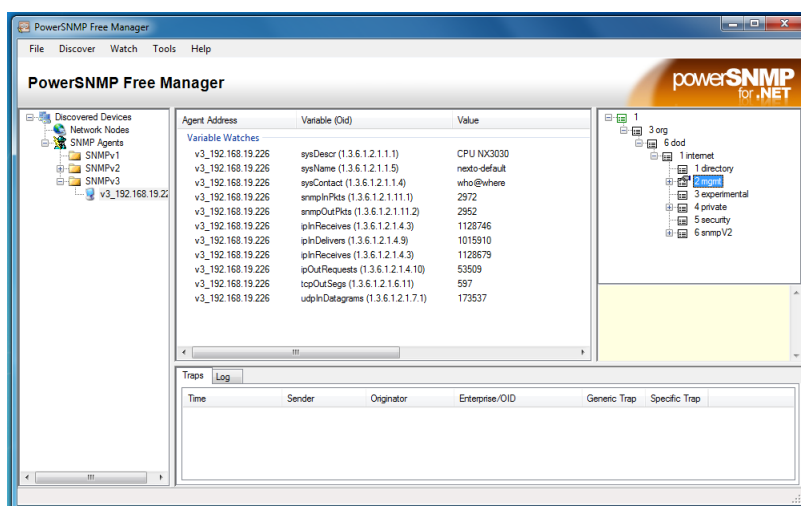


Figura 165: Exemplo de Gerente SNMP

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 do controlador. Para mais detalhes consulte a seção [Configuração SNMP](#).

5.14.3.3.2. 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. Abre-se 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.

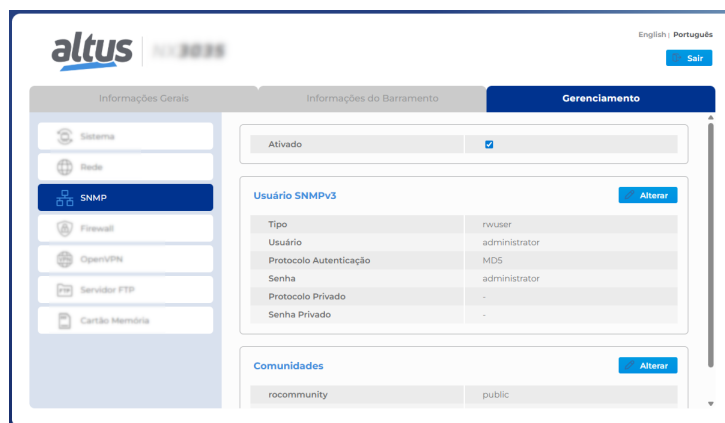


Figura 166: Tela de Configuração e Status SNMP

5.14.3.3.3. 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 191: 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 192: 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	1	30	[0-9][a-z][A-Z]@\$*_.
Senha v3	8	30	[0-9][a-z][A-Z]@\$*_.

Tabela 193: Limites para Configurações SNMP

5.14.3.4. Dispositivos USB

A porta Host USB presente nos controladores da Série Nexto permite ampliar as funcionalidades do controlador utilizando diversos tipos de dongles USB.

O gerenciamento de dispositivos USB é feito através de uma seção dedicada localizada na aba *Gerenciamento* da Página Web de Sistema do controlador, conforme mostrado abaixo:

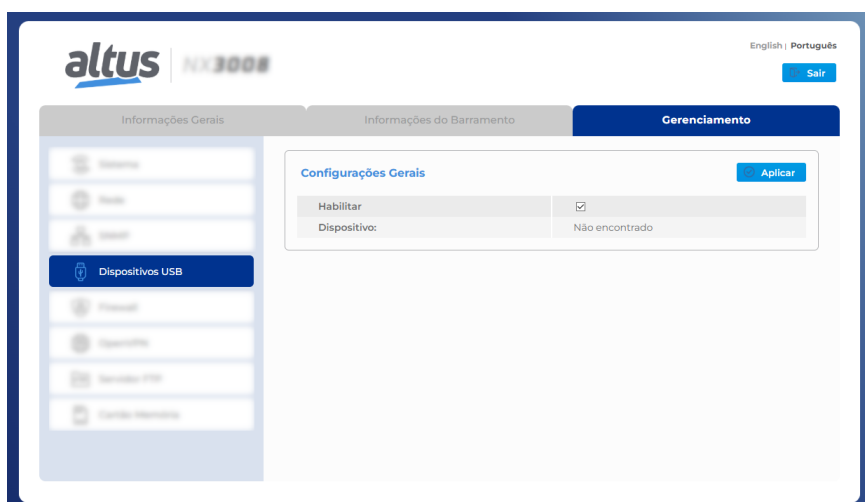


Figura 167: Seção de dispositivos USB

Nessa página, o estado da interface USB pode ser configurado, marcando ou desmarcando a caixa de seleção e aplicando a configuração com o botão Aplicar. Se a interface USB está habilitada, o conteúdo desta página muda dinamicamente de acordo com o tipo de dispositivo USB conectado. No exemplo acima, a interface está habilitada e não há nenhum dispositivo conectado. Caso a porta USB esteja desabilitada, o conteúdo é o apresentado na figura, sempre apresentando Dispositivo como "Não Encontrado", mas com a caixa de seleção desmarcada.

As seções a seguir descrevem todos os tipos de dispositivos USB atualmente suportados. Se um dispositivo não suportado estiver conectado, a página informará que o dispositivo é desconhecido:

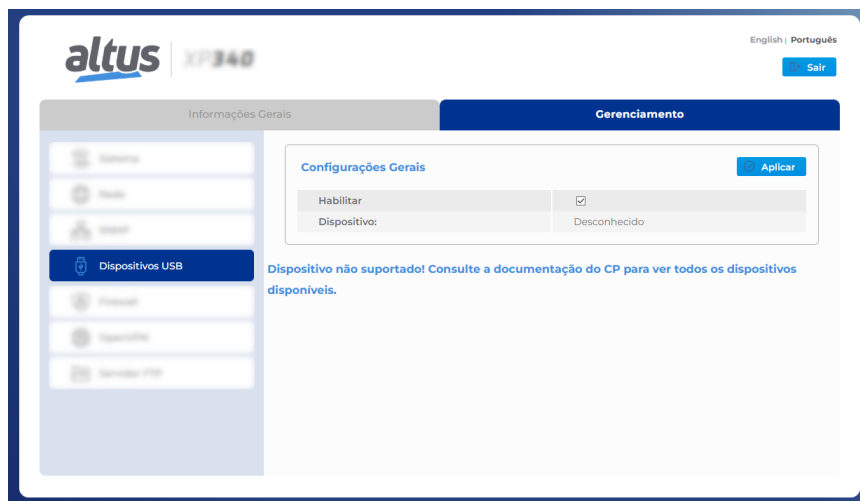


Figura 168: Dispositivos USB - Desconhecido

5.14.3.4.1. Dispositivos de Armazenamento em Massa

1. Armazenamento Geral

Os dispositivos de armazenamento em massa podem ser usados para expandir a memória flash do controlador para armazenar uma grande quantidade de dados, como em aplicações de registrador de dados, por exemplo. Para usar um dispositivo de armazenamento em massa USB, basta conectá-lo à porta USB. Após alguns segundos, quando o dispositivo for detectado e montado corretamente, as informações do dispositivo aparecerão na seção *Dispositivos USB* localizada na aba *Gerenciamento* da Página Web de Sistema do controlador, conforme mostrado abaixo:

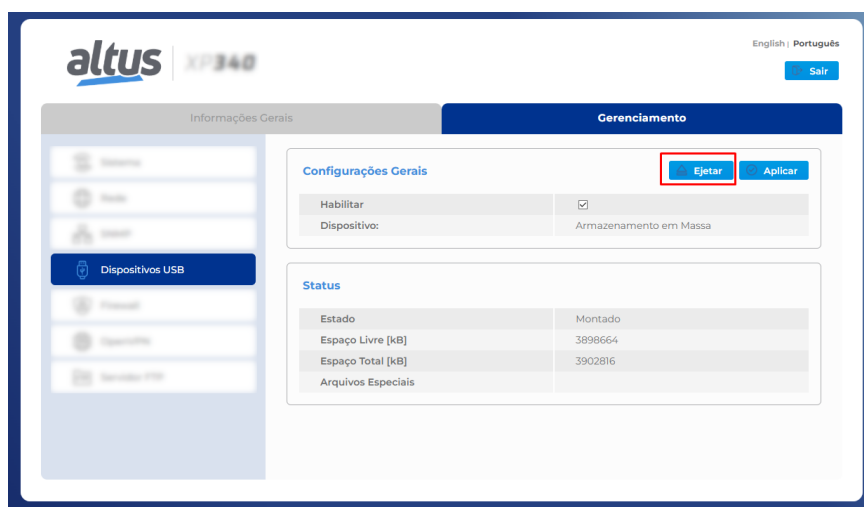


Figura 169: Informações sobre o dispositivo de armazenamento em massa

ATENÇÃO

Para que o dispositivo USB seja reconhecido corretamente pelo controlador, deve haver pelo menos *1Kb* de espaço livre no *CLP*.

As informações mostradas na seção de *Status* desta página também estão disponíveis na estrutura de diagnóstico de variáveis simbólicas (consulte a Seção [Diagnósticos via Variáveis](#)).

ATENÇÃO

O dispositivo de armazenamento em massa USB deve ser formatado como um volume FAT32. Outros formatos de sistema de arquivos não são suportados.

O dispositivo pode ser ejetado usando o comando fornecido na área *Comandos* desta página, conforme indicado na figura acima.

Depois que o dispositivo for detectado e montado corretamente, uma nova pasta chamada *Mass_Storage* aparecerá na memória do controlador, conforme mostrado na figura abaixo:

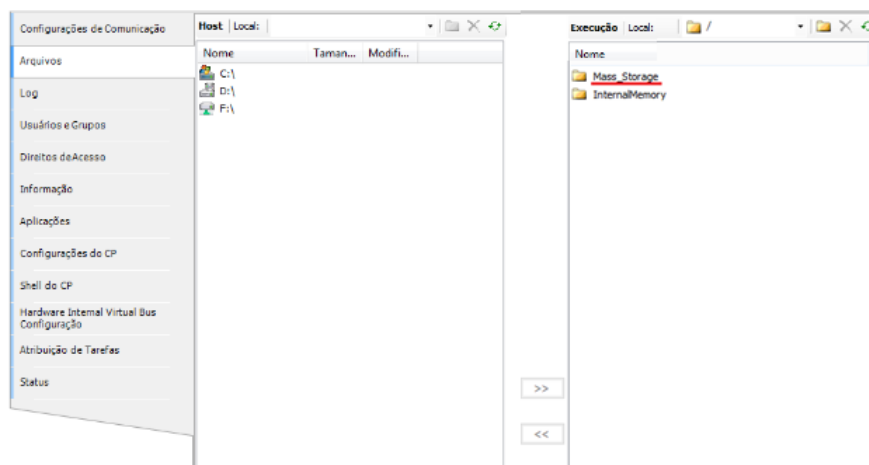


Figura 170: Pasta de armazenamento em massa USB

2. Não Carregando a Aplicação na Inicialização

O dispositivo de armazenamento em massa USB pode ser usado para impedir que o controlador carregue automaticamente a aplicação após a energização. Para fazer isso, basta colocar um arquivo de texto vazio chamado *dontbootapp.txt* na pasta raiz do dispositivo de armazenamento em massa. A presença desse arquivo é informada no campo *Arquivos Especiais* da seção *Dispositivos USB* da Página Web de Sistema, conforme mostrado abaixo.

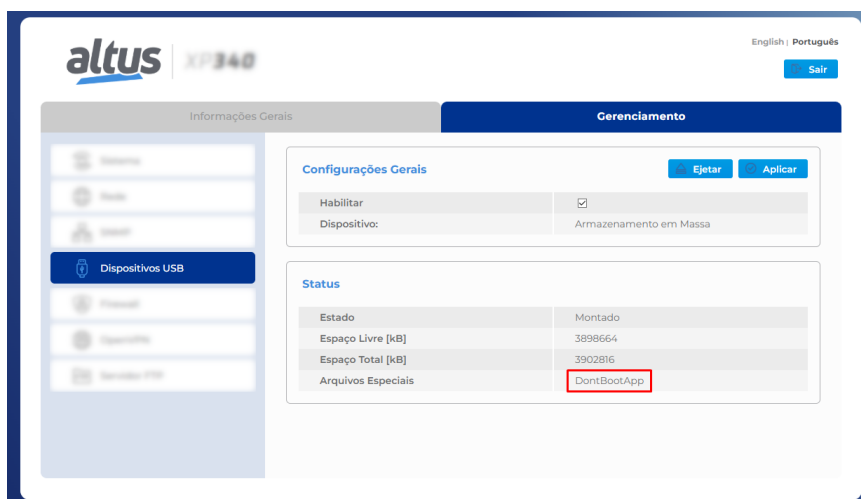


Figura 171: DontBootApplication no dispositivo de armazenamento em massa

3. Transferindo uma Aplicação a partir do dispositivo USB

O dispositivo de armazenamento em massa USB também pode ser usado para transferir uma aplicação para o controlador. Para fazer isso, coloque os dois arquivos *Application.app* e *Application.crc* na pasta raiz do dispositivo de armazenamento em massa, caso exista um WebVisu declarado no projeto, também deve ser copiada a pasta *PLC Folder* (esses arquivos são criados usando o Mastertool executando o comando *Comunicação -> Criar Aplicação de Inicialização* quando off-line). Após a energização, se o controlador detectar a presença desses arquivos no dispositivo de armazenamento em massa USB, a seguinte sequência de ações ocorrerá:

- O controlador iniciará a cópia da aplicação do dispositivo USB para a memória interna
- Depois de terminar o processo de cópia, o dispositivo USB será ejetado
- A nova aplicação será iniciada (RUN) automaticamente (se *dontbootapp.txt* não estiver presente)

A presença da aplicação é informada no campo *Arquivos Especiais* na seção *Dispositivos USB* da Página Web de Sistema, conforme mostrado abaixo:

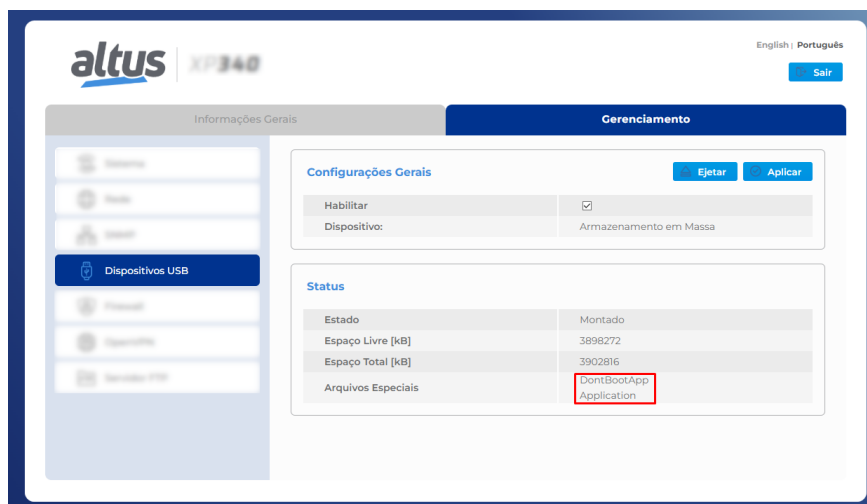


Figura 172: Aplicação no dispositivo de armazenamento em massa

Observe que é possível ter vários arquivos especiais no mesmo armazenamento em massa. No exemplo acima, o CP transferirá a nova aplicação para a memória interna, mas não a carregará na inicialização (portanto, não será executada).

5.14.3.4.2. Conversor USB para RS-232

A Série Nexto permite implementar uma porta RS-232 usando um conversor USB para Serial. Esses conversores são todos baseados em um chip controlador interno. A tabela abaixo mostra a lista de controladores suportados.

Controlador	Fabricante
FT232	FTDI
PL2303	Prolific

Tabela 194: Conversores USB para RS-232 suportados

Essa porta deve ser usada exclusivamente com os blocos funcionais de comunicação serial fornecidos pela biblioteca *NextoSerial*, permitindo implementar uma comunicação ponto a ponto com equipamentos que utilizam protocolos simples (não críticos ao tempo) como *Radio modems*, *leitores de código de barras*, *leitores RFID*, etc ... Além disso, esse tipo de solução possui as seguintes limitações:

- **Baud Rate:** valores inferiores a 4800 bps não são suportados
- **Data Bits:** o valor “5” não é suportado (apenas 6, 7 ou 8)
- **Parity:** os valores “mark” e “space” não são suportados (apenas Odd, Even e None)
- **Stop Bits:** o valor “1.5” não é suportado (apenas 1 ou 2)

Depois de conectar o conversor à porta USB, e o dispositivo for detectado e montado corretamente, as informações do dispositivo aparecerão na seção *Dispositivos USB*, localizada na aba *Gerenciamento* da Página Web de Sistemas, como mostrado abaixo:

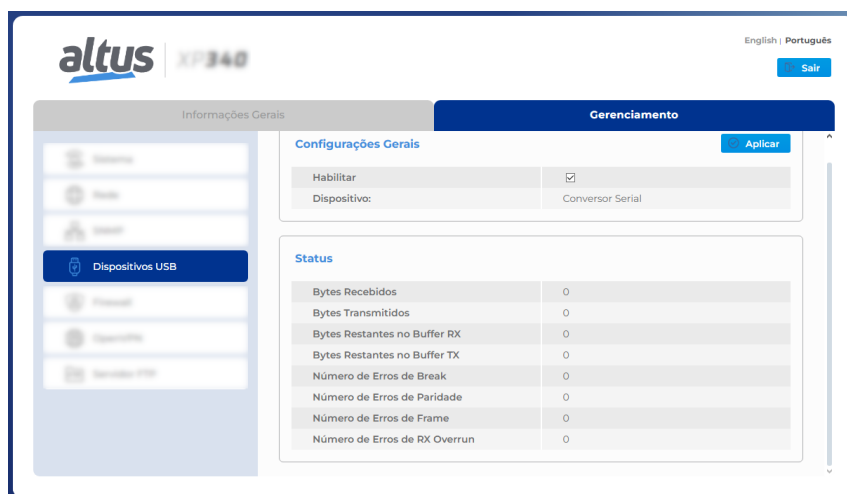


Figura 173: Dispositivos USB - Conversor Serial

As informações mostradas nesta página também estão disponíveis na estrutura de diagnóstico de variáveis simbólicas (consulte a Seção [Diagnósticos via Variáveis](#)).

Essa porta serial adicional será identificada internamente como *COM10* e não terá uma representação na visualização da árvore do projeto. A partir deste ponto, essa porta pode ser usada para comunicação usando as funções *NextoSerial* semelhantes às portas nativas. Para esse tipo de porta, a configuração do handshake é limitada apenas a *RS232_MANUAL* (deve ser considerada ao configurar a porta com a função *SERIAL_CFG*).

5.14.3.4.3. Dispositivos Modem

Um modem USB com cartão SIM pode ser usado para conectar o CP (CLP) à internet utilizando a rede de dados celulares. O controlador suporta dois tipos de modem: Bridge (Ponte) e Router (Roteador).

A tabela a seguir mostra a lista de dispositivos compatíveis:

Modelo	Fabricante	Tipo	Observações
E303	Huawei	Bridge	-
E3272	Huawei	Bridge	-
E3276	Huawei	Bridge	-
E8372	Huawei	Roteador	-

Tabela 195: Modems USB suportados

1. Modems tipo Bridge

Modems bridge estabelecem uma conexão PPP (Point-to-Point Protocol) interna entre o controlador e o modem. Essa conexão é utilizada apenas para comunicações de saída iniciadas pelo próprio controlador, tais como comunicação MQTT em nuvem, sincronização SNTP, contêineres acessando serviços externos e serviços semelhantes baseados na internet.

ATENÇÃO

Os modems bridge não expõem uma rede Ethernet local acessível e, portanto, não podem ser usados para acesso externo direto aos serviços do controlador, como a Página Web do Sistema, WebVisu, Docker ou comunicação Mastertool.

Para modems bridge, a configuração é realizada através da Página Web do Sistema do controlador, conforme descrito na seção [Dispositivos Modem](#).

2. Modems tipo Router

Modems roteadores operam de maneira diferente. Esses dispositivos criam e gerenciam sua própria rede Ethernet local, normalmente atribuindo endereços IP privados aos dispositivos conectados. Qualquer dispositivo conectado à mesma rede local gerada pelo modem pode se comunicar com o controlador utilizando o endereço IP privado atribuído pelo

modem. Nessa configuração, serviços como a Página Web do controlador, WebVisu, acesso ao Docker e comunicação Mastertool são suportados, desde que:

- O dispositivo do usuário esteja conectado à mesma rede local gerada pelo modem;
- A configuração de rede e firewall do modem permita a comunicação entre os dispositivos.

Para identificar o endereço IP privado, é necessário acessar a própria interface web ou aplicativo móvel do modem, onde a lista de dispositivos conectados e seus respectivos endereços IP privados pode ser visualizada.

ATENÇÃO

O endereço IP atualmente exibido na Página Web do controlador e nos diagnósticos do Mastertool para modems roteadores é, normalmente, o endereço IP público atribuído pela operadora. Esse endereço é útil apenas para diagnóstico e verificação de conectividade com a internet.

Modems roteadores executam internamente seu próprio processo de conexão automática, comumente chamado de Auto Connect. Essa funcionalidade geralmente vem habilitada por padrão, mas normalmente pode ser configurada através da própria interface web do modem.

Modems roteadores são dispositivos gerenciados que implementam internamente políticas de firewall e roteamento. Toda a configuração específica do modem é realizada através da própria interface web do modem. Por padrão, a maioria dos modems roteadores bloqueia conexões de entrada vindas de redes externas. Para permitir o acesso remoto aos serviços do controlador, o modem deve ser configurado para abrir as portas TCP associadas ao serviço desejado.

A tabela na Seção [Portas TCP/UDP reservadas](#) apresenta as portas TCP e UDP utilizadas pelos principais serviços do controlador.

3. Gerenciamento de Modem

A gestão da funcionalidade do modem USB é feita através da seção Dispositivos USB na aba Gerenciamento da Página Web do Sistema do controlador. Assim que o dispositivo modem for devidamente detectado e montado, as informações aparecerão conforme mostrado abaixo:

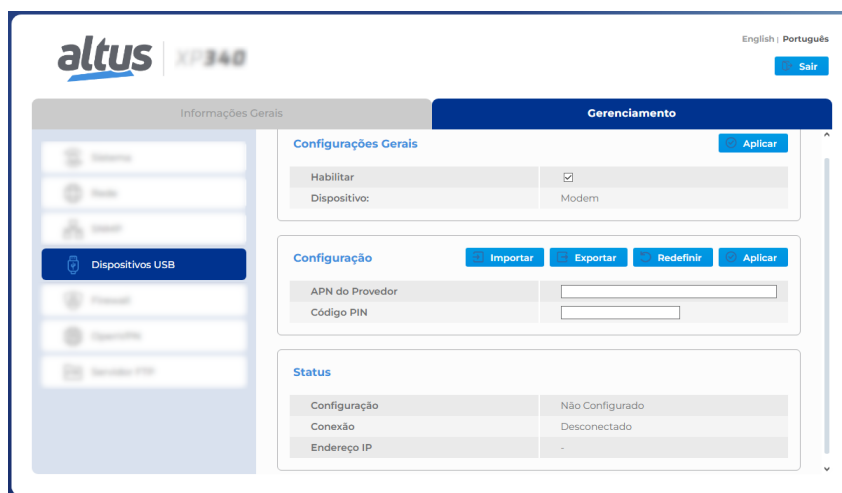


Figura 174: Configuração da Página do Modem USB

Esta área contém duas seções: *Status* e *Configuração*.

A seção *Status* exibe todos os diagnósticos relacionados ao modem USB, incluindo o estado de configuração, o estado da conexão com a internet e o endereço IP do modem. Esses campos são atualizados automaticamente sempre que um valor é alterado e não exigem que a página seja recarregada manualmente. As mesmas informações também estão disponíveis através da estrutura de diagnóstico por variáveis simbólicas (consulte a Seção [Diagnósticos via Variáveis](#)), que adicionalmente contém descrições detalhadas dos valores possíveis para cada campo. A seção *Configuração* é utilizada para configurar o modem. Para modems bridge, esta seção permite ao usuário configurar os parâmetros do cartão SIM.

ATENÇÃO

Os parâmetros de APN da operadora e código PIN podem ser necessários dependendo do provedor do cartão SIM. Se a operadora especificar esses parâmetros, eles devem ser configurados de acordo. Alguns cartões SIM usam valores predefinidos internamente e podem operar corretamente mesmo se esses campos permanecerem com os valores padrão. O parâmetro PIN aceita apenas de 4 a 8 dígitos, enquanto o parâmetro APN não aceita valores vazios.

Esta seção de configuração contém os seguintes botões:

- **Importar:** carrega a configuração a partir de um arquivo externo
- **Exportar:** salva a configuração atual em um arquivo externo
- **Resetar:** apaga a configuração do modem da memória do controlador
- **Aplicar:** grava a configuração na memória do controlador

ATENÇÃO

Para modems roteadores, esta seção atualmente não contém nenhum botão de configuração.

O princípio de funcionamento da funcionalidade do modem USB depende do tipo de modem. Em casos específicos, uma operadora pode fornecer um endereço IP acessível publicamente para o cartão SIM. Nessas situações, o acesso externo ao controlador pode se tornar possível. No entanto, essa configuração é incomum e pode expor o controlador diretamente à internet, o que pode representar um risco de segurança cibernética.

Para modems bridge, assim que o dispositivo é devidamente detectado e os parâmetros do cartão SIM são configurados, o controlador inicia um processo em segundo plano que gerencia continuamente a conexão do modem para mantê-lo conectado à internet. Se a conexão for interrompida por qualquer motivo (como baixa qualidade de sinal ou queda da operadora), o controlador tenta se reconectar automaticamente. O status desse processo pode ser monitorado através do campo *Status* de Conexão.

Um aspecto importante a ser considerado é que, quando um modem USB é configurado, o sistema do controlador o define como o gateway padrão para a comunicação Ethernet. Isso significa que, se o controlador estiver conectado simultaneamente a outra interface de rede local que também possua acesso à internet, as comunicações destinadas a endereços IP externos serão roteadas através do modem USB em vez da interface Ethernet. Quando o modem USB é removido, a interface Ethernet configurada anteriormente torna-se o gateway padrão novamente. A sequência de configuração e operação pode ser resumida nas seguintes etapas:

- Conecte o dispositivo à porta USB. Após alguns segundos, a página *Dispositivos USB* deve identificar o dispositivo como *Modem*. Se isso não ocorrer, o dispositivo pode não ser compatível ou estar com defeito.
- (Para modems bridge) Configure os parâmetros do cartão SIM, incluindo a APN (Access Point Name) da operadora e o código PIN do cartão SIM, de acordo com as informações fornecidas pela operadora. Após clicar no botão *Aplicar*, o controlador inicia o processo de conexão automaticamente. Uma vez configurados, esses parâmetros permanecem armazenados na memória do controlador e não precisam ser inseridos novamente.
- Monitore o estado da conexão através do campo *Status* correspondente. Se a conexão for bem-sucedida, o modem indicará que está conectado à internet e um endereço IP será exibido.

5.14.3.4.4. Adaptadores Wi-Fi

Um adaptador USB Wi-Fi pode ser usado para conectar o CP a uma rede Wi-Fi existente, criando um segundo adaptador de rede que pode ser usado para programação e comunicação. A tabela a seguir mostra a lista de chipsets suportados.

Chipset	Fabricante	Exemplo de produtos comerciais
RTL8188EU	Realtek	TP-LINK modelo TL-WN725N
		LM Technologies modelo LM007
RT28xx	Ralink/Mediatek	D-Link modelo DWA-125

Tabela 196: Chipsets suportados para adaptadores USB Wi-Fi

O gerenciamento da funcionalidade do adaptador Wi-Fi é feito através da página *Dispositivos USB* na guia *Gerenciamento* da página web do controlador. Depois que o dispositivo for detectado e montado corretamente, o LED da USB irá ligar e as informações aparecerão como mostrado abaixo:

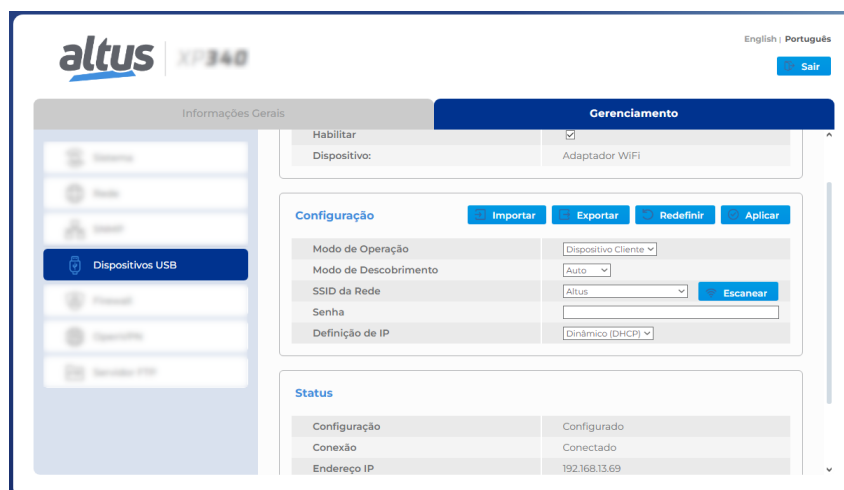


Figura 175: Página do adaptador USB Wi-Fi

A seção *Configuração* é composta pelos seguintes parâmetros:

- **Modo de Operação:** Define como o adaptador Wi-Fi funcionará. Atualmente, apenas o modo *Dispositivo Cliente* está disponível.
- **Modo de Descobrimto:** Método para definir a rede Wi-Fi. Se selecionado como *Auto*, o botão *Escanear* deve ser usado para escolher a rede desejada. Se selecionado como *Manual*, os demais itens listados devem ser preenchidos manualmente.
- **SSID da Rede:** Quando o *Modo de Descobrimto* estiver definido como *Auto*, esse campo mostrará as redes disponíveis encontradas no processo de verificação, classificadas do melhor ao pior nível de sinal (de cima para baixo). Quando estiver definido como *Manual*, este campo deve ser preenchido com o *SSID* da rede desejada.
- **Tipo de Segurança:** Define o tipo de segurança usada na rede Wi-Fi, que pode ser *Pública* ou *WPA2-Personal*.
- **Senha:** Senha da rede Wi-Fi. O campo será bloqueado automaticamente se o *Tipo de Segurança* estiver definido como *Pública* ou se a rede escaneada escolhida não usar um protocolo de segurança.
- **Definição de IP:** Escolhe se o adaptador Wi-Fi definirá o endereço IP dinamicamente, atribuído pelo servidor DHCP da rede, ou estaticamente, no qual o usuário precisará inserir as configurações de IP manualmente.
- **Endereço IP, Máscara de Rede e Gateway:** Disponível apenas quando a *Definição de IP* estar definida como *Estático*. Servem para configurar os parâmetros de rede do adaptador Wi-Fi.

ATENÇÃO

Para uma operação adequada, a rede do adaptador Wi-Fi, definida por IP e Máscara, deve ser diferente da rede configurada nas interfaces nativas do CP.

A seção *Status* mostra todos os diagnósticos relacionados ao adaptador Wi-Fi: estado de configuração, estado da conexão, endereço IP, máscara de rede, gateway e endereço MAC. Esses campos são atualizados assim que os valores mudam. Essas informações também são fornecidas na estrutura de diagnóstico de variáveis simbólicas (consulte a Seção [Diagnósticos via Variáveis](#)).

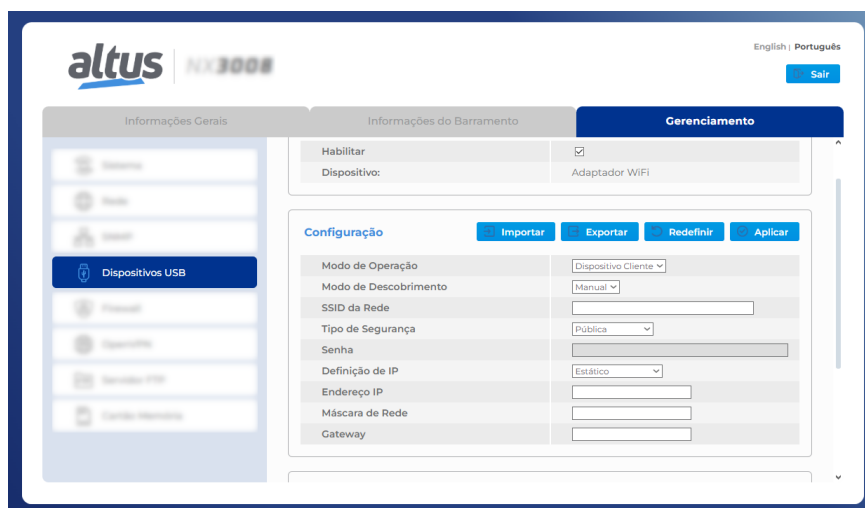


Figura 176: Configuração manual do adaptador USB Wi-Fi

Além disso, a seção *Configuração* contém os seguintes botões:

- **Importar:** Carrega um arquivo de configuração.
- **Exportar:** Baixa um arquivo com a configuração atual.
- **Redefinir:** Retorna à configuração padrão.
- **Aplicar:** Aplica a configuração atual.

Depois que o dispositivo for detectado corretamente e os parâmetros de rede estiverem configurados, o controlador sempre tentará manter-se conectado à rede Wi-Fi. Esse processo pode ser observado no campo *Status*, *Conexão*.

ATENÇÃO

O adaptador USB Wi-Fi torna-se o Gateway Padrão quando configurado.

A sequência de configuração e operação pode ser resumida nas seguintes etapas:

1. Conecte o adaptador na porta USB. Após alguns segundos, a página *Dispositivos USB* exibirá o *Dispositivo* como *Adaptador Wi-Fi*. Caso contrário, o dispositivo pode não ter suporte ou estar com defeito.
2. Defina a configuração de rede. Após clicar no botão *Aplicar*, o processo de conexão será iniciado. Uma vez configurada, não é mais necessário definir essas informações. Elas serão salvas na memória do controlador.
3. Monitore o status da conexão no campo correspondente. Em caso de sucesso, após alguns segundos será informado que o adaptador está conectado à rede Wi-Fi.

A imagem a seguir mostra a página de um controlador conectado a uma rede Wi-Fi:

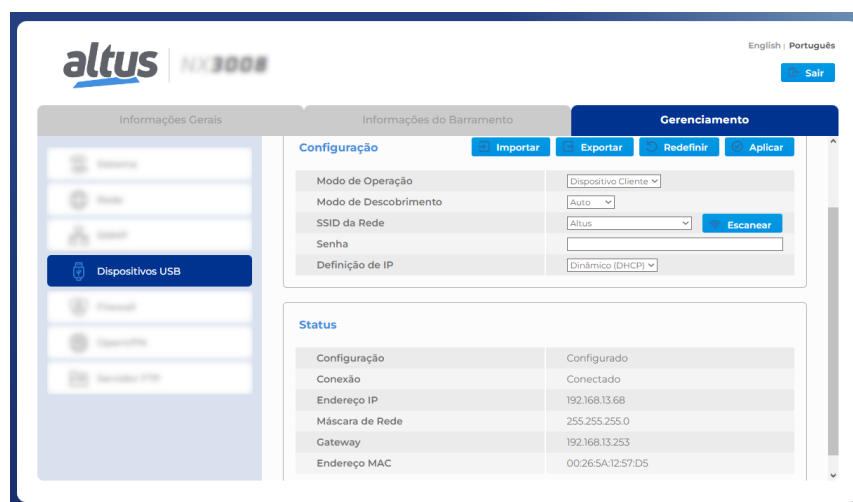


Figura 177: Adaptador USB Wi-Fi conectado a uma rede

Uma vez conectado à rede Wi-Fi, esse canal de comunicação pode ser usado para vários propósitos. Para programar o CP com o Mastertool, o gateway deve ser configurado com o endereço IP atribuído ao adaptador Wi-Fi. Esse endereço IP também pode ser usado para acessar a página web do controlador, no qual é possível executar uma atualização de firmware. Além disso, esse canal de comunicação também pode ser usado com o Bloco de Funções do cliente MQTT para relatar dados para um broker externo na Internet, sendo nesse caso o adaptador Wi-Fi como Gateway Padrão.

5.14.3.4.5. Função de Usuário para Controle da Interface USB

Para facilitar o gerenciamento da interface da USB, foi desenvolvida uma função que pode ser chamada diretamente pelo código de aplicação do usuário. A função **SetUSBState** foi implementada dentro da biblioteca **NextoStandard**. A imagem abaixo mostra as informações da biblioteca, apresentadas no *Library Manager*.

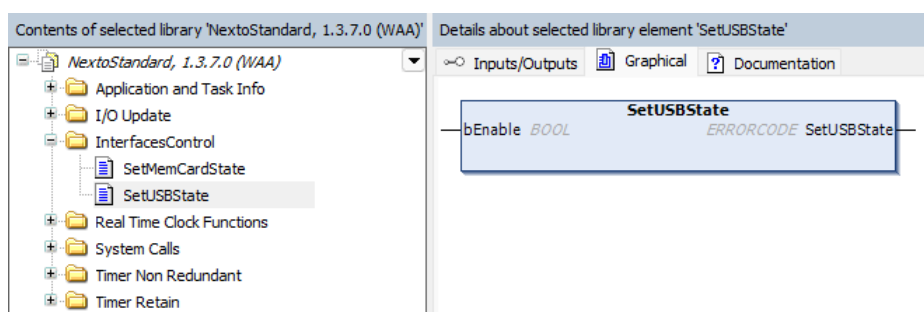


Figura 178: Informações de SetUSBState no Library Manager

A função possui uma variável de entrada do tipo *bool*, **bEnable**, que recebe o valor para habilitar ou desabilitar a interface USB. A função possui três valores de retorno: *NoError* em caso de sucesso, *SetUSBStateFail* em caso de falha ou *Import-FunctionNotFound* caso a função não seja suportada. A imagem abaixo mostra um exemplo básico de declaração de variáveis e chamada da função.

```
PROGRAM UserPrg
VAR
bSetUSBInterfaceState : BOOL;
stErrorCode           : NextoStandard.ERRORCODE;
END_VAR
-----
stErrorCode := NextoStandard.SetUSBState(bEnable := bSetUSBInterfaceState);
```

ATENÇÃO

A função executa o comando para definir o valor desejado para a interface da USB. Não é necessário, nem recomendado, que a função seja chamada ciclicamente.

5.14.3.5. Firewall

O Firewall foi desenvolvido para aumentar a segurança do dispositivo durante a sua utilização. A principal função do Firewall é realizar um filtro sobre os pacotes de dados que chegam e que saem do dispositivo. O filtro implementado utiliza informações de cada pacote de dados para decidir se aquele pacote é permitido ou não. Os principais parâmetros utilizados são as interfaces de entrada/saída, a porta, o protocolo da camada de transporte e os endereços de origem e destino.

ATENÇÃO

Quando o dispositivo é inicializado com o botão pressionado, o Firewall será desabilitado e as regras não serão aplicadas.

A configuração do Firewall é feita através de uma seção dedicada localizada na aba *Gerenciamento* da Página Web de Sistema do controlador, conforme mostrado abaixo:

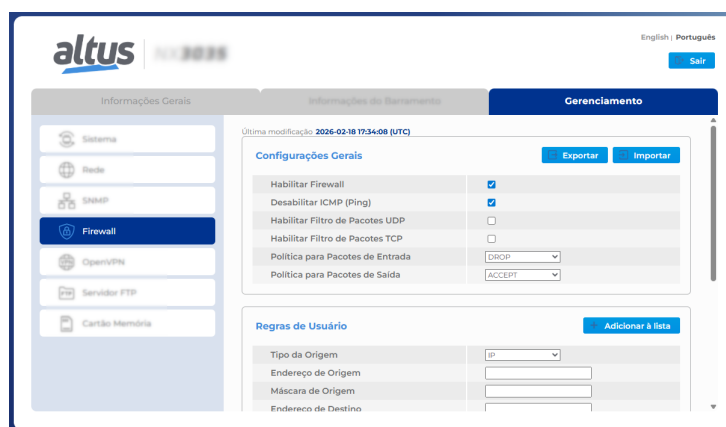


Figura 179: Tela de Configuração do Firewall

O Firewall trata-se de uma funcionalidade à parte ao Mastertool, isto é, não necessita de interação alguma com o Mastertool. As configurações aplicadas na Página Web de Sistema passam a valer quando confirmadas através do botão *Aplicar* e, são salvas automaticamente no controlador. Contanto que a funcionalidade esteja habilitada, voltará a operar mesmo após a reinicialização do dispositivo.

As próximas seções descrevem as possíveis configurações para o Firewall, divididas de acordo com as tabelas da seção *Firewall*.

5.14.3.5.1. Configurações Gerais

A imagem abaixo exhibe todas as configurações da tabela *Configurações Gerais*:

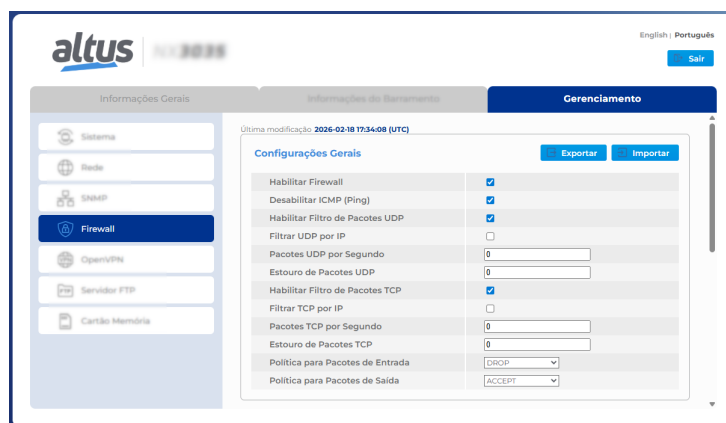


Figura 180: Tabela de Configurações Gerais do Firewall

Esta tabela expande de forma dinâmica ao selecionar as opções de habilitar filtros de pacotes UDP e TCP, revelando todos os itens possíveis de serem configurados. O primeiro item desta tabela, *Habilitar Firewall*, serve para habilitar e desabilitar esta funcionalidade. Quando o Firewall está habilitado, as configurações da seção, ao serem submetidas ao dispositivo, serão aplicadas nos arquivos de configurações e então, o Firewall passará a filtrar o que foi configurado. Caso o Firewall esteja desabilitado, a configuração que foi realizada é armazenada, porém as regras não são aplicadas no controlador.

O campo *Desabilitar ICMP (Ping)* habilita ou desabilita a proteção contra o protocolo ICMP. Quando selecionado, o controlador passa a não responder as requisições de *Ping*, uma vez que a proteção irá realizar o descarte de pacotes que utilizarem o protocolo ICMP. Já quando desativada, o funcionamento do dispositivo em relação a respostas de *Ping* mantém seu comportamento normal.

Os campos que habilitam o filtro de pacotes UDP e TCP, quando habilitado, realizam o filtro destes protocolos dentro dos limites configurados nos seus respectivos campos. A regra que realiza a filtragem dos pacotes possui o seguinte funcionamento: para que um pacote seja aceito, é necessário que existam “créditos” disponíveis, um crédito é utilizado para aceitar um pacote de dado.

A configuração do campo *Estouro de Pacotes XXX* configura o valor inicial de pacotes, ou créditos, que serão aceitos. Dessa forma, é possível configurar um limite de estouro destes pacotes, onde caso haja um fluxo muito grande de pacotes, somente serão aceitos a quantidade configurada. O campo de *Pacotes XXX por Segundo*, configura quantos créditos aquela regra irá ganhar por segundo, ou seja, se for configurado o valor 5, a cada segundo a regra receberá cinco novos créditos, logo poderá aceitar mais cinco pacotes. A limitação para esse incremento no número de créditos é a própria configuração de *Estouro de Pacotes XXX*, o limite estabelecido aqui não é ultrapassado, mesmo com o incremento de pacotes a cada segundo. Estas configurações são aplicadas como um *estoque*, onde ao receber um pacote de dados, primeiro é verificado se há algum crédito disponível no estoque e então é tomada a decisão de aceitar ou não o pacote. Se o pacote é aceito neste filtro de quantidade, ele é direcionado para o filtro das demais regras de Firewall.

A configuração *Habilitar Filtro por IP*, faz com que a regra diferencie os endereços de origem de cada pacote e aplique os filtros de estouro e de pacotes por segundo de forma individual para cada endereço. Assim, retomando o exemplo anterior, pode ser considerado que cada endereço de origem possui o seu *estoque* de créditos e um endereço não pode utilizar os créditos que estão no *estoque* reservados para outro.

ATENÇÃO

Não é permitido a configuração de valores negativos para os campos de *Pacotes XXX por Segundo* e *Estouro de Pacotes XXX*. Caso sejam configurados valores negativos, ao aplicar as configurações será exibido uma mensagem de erro na tela indicando o campo que houve algum conflito. Caso o filtro seja habilitado, porém, os valores nestes campos sejam deixados em 0, o filtro não é aplicado.

As configurações desta tabela são aplicadas com o botão de *Aplicar* que aparece na figura 182.

Os campos para selecionar ambas as políticas, de entrada e de saída, possuem as opções de *Accept*, aceitar, e de *Drop*, descartar. As políticas possuem o seguinte funcionamento: se o Firewall estiver ativo, quando os pacotes de dados chegam, todas as regras que foram configuradas são verificadas, então será aplicada a política configurada para estes pacotes, seja ela de *Accept* ou *Drop*. Assim, se for configurado uma política de aceitação, *Accept*, todos os pacotes que não combinarem com nenhuma regra configurada, serão aceitos pelo Firewall e se configurado uma política de rejeição, *Drop*, os mesmos seriam todos descartados.

Na parte superior da tabela, estão disponíveis os botões *Exportar* e *Importar*. Ao utilizar o botão *Exportar* é possível realizar o download de um arquivo com extensão *.firewall* contendo todas as configurações de Firewall que estão aplicadas. Já o botão *Importar*, permite a importação de um arquivo *.firewall* previamente exportado dessa ou de outra UCP de mesmo modelo, contendo as configurações de Firewall. Ao importar um arquivo, as regras contidas nele serão validadas e configuradas nos campos das *Configurações Gerais* e na tabela de *Regras de Usuário*. Após a importação, as regras serão aplicadas ao clicar no botão *Aplicar*.

ATENÇÃO

O arquivo exportado contém apenas as regras que foram efetivamente aplicadas ao dispositivo. Qualquer configuração ou regra adicionada na lista da página web, mas não aplicada à UCP, não será incluída no arquivo gerado.

ATENÇÃO

Qualquer modificação no arquivo *.firewall*, como edição manual ou alteração da extensão, pode impedir que algumas regras sejam aplicadas corretamente. Sempre utilize arquivos gerados pelo botão *Exportar* da mesma UCP, ou de outra do mesmo modelo, ao realizar a importação de configurações de Firewall.

5.14.3.5.2. Regras de Usuário

A tabela *Regras de Usuário* foi criada para permitir um controle maior sobre as configurações de regras do Firewall. Com ela é possível configurar diferentes regras de forma dinâmica e com filtros mais precisos.

Figura 181: Tabela de Configuração de Regras de Usuário do Firewall

Esta tabela altera o seu formato de acordo com o *Tipo de Origem* selecionado, podendo ser IP ou MAC. Quando o tipo é IP, a tabela possui os itens exibidos na figura acima, porém, quando é selecionado o tipo como MAC os campos de máscaras de origem e destinos desaparecem, assim como o campo de *Endereço do Destino*. O item *Endereço de Origem* passa a aceitar um endereço MAC como entrada em um formato de seis grupos de dois dígitos hexadecimais separados por dois pontos, ex: "1A:2B:3C:4D:5E:6F". Além disso, uma regra com base em endereço MAC somente pode ser configurada como uma regra de entrada, ou seja, o campo *Direção* será forçado com o valor *INPUT*.

Com os campos de *Endereço de Origem* e *Endereço de Destino*, é possível informar os endereços que serão configurados para aquela regra específica e com o uso dos campos *Máscara de Origem* e *Máscara de Destino* é possível configurar uma faixa de rede para esta regra. A definição de um endereço e uma máscara resulta em um grupo de IPs que será atribuído a regra que está sendo configurada. Se for realizada somente a configuração do endereço, este único endereço será atribuído à regra, porém com diferentes configurações de máscara de rede é possível obter grupos de IPs de diversos tamanhos, que serão aplicados a regra.

A configuração de interface possibilita a seleção de cada interface física ou virtual disponível para o controlador, de forma individual. Além disso, existe a opção *Qualquer*. Com base na interface que for selecionada para determinada regra, somente os pacotes de dados que estiverem entrando ou saindo por ela, serão filtrados pelo Firewall. Se utilizado a opção *Qualquer*, a regra não possuirá filtro de interface, logo, a regra passa a valer para todas as disponíveis.

O campo *Ação* possui três opções de configuração: *ACCEPT*, *DROP* e *REJECT*. A ação configura o que deve ser feito com o pacote cujas características conferem com a regra aplicada. Caso a ação escolhida seja *ACCEPT*, o pacote de dados que tiver suas características de acordo com a regra, será aceito. Caso seja *DROP*, o pacote será descartado e nenhuma resposta será enviada a quem enviou o pacote. Por último, caso seja configurado como *REJECT*, o pacote será rejeitado e será encaminhada uma resposta para quem enviou o pacote, informando que o *host* solicitado está inacessível.

O campo *Porta de Serviço* serve para indicar quais as portas serão configuradas nesta regra. Todas as portas de serviços que possuem um determinado protocolo ou comunicação *padrão* para o controlador, como por exemplo o protocolo MODBUS que tem a porta padrão 502, estão disponíveis com o nome do serviço e a porta utilizada ao lado. Assim, caso seja configurada a regra para o protocolo MODBUS, será aplicada a porta 502, caso seja configurada a regra para o serviço WebVisu, será aplicada a porta 8080 e assim segue para os demais protocolos listados no campo de seleção.

Este campo também possui outras duas configurações, que são *Qualquer* e *Outra*. Quando é selecionada a opção *Qualquer*, a regra é aplicada para todas as portas de serviços, exceto a porta 80, então são criadas duas regras utilizando as seguintes faixas de portas: 1:79 e 81:65535. Caso seja selecionada a opção *Outra*, será exibido uma caixa de texto na qual é possível configurar a porta que deseja, exceto a porta 80. Para configurar uma porta, basta escrever o seu número na caixa de texto, se desejar adicionar mais de uma única porta, é necessário utilizar o separador "&" e caso queira inserir uma faixa de portas, basta informar a porta inicial e final utilizando o separador ":".

Exemplo de configuração das portas 120, 144, e da faixa de 1300 a 1450 no mesmo campo: 120 & 144 & 1300:1450.

Este campo não aceita valores fora da faixa 1:65535, a porta 80 ou repetições de portas.

A porta HTTP, 80, somente pode ser configurada através da seleção na lista de protocolos conhecidos e não pode ser aplicada a interface NET1. Sendo assim, caso o protocolo HTTP seja escolhido, os campos *NET1* e *Qualquer* do campo de Interface não serão possíveis de serem selecionados.

No campo de *Protocolo* é possível selecionar entre os protocolos *UDP*, *TCP* e *UDPICTP*. Caso seja selecionado a opção *UDPICTP*, serão criadas duas regras no Firewall, uma para cada protocolo de transporte.

No campo de *Direção* é possível selecionar entre *INPUT*, *OUTPUT* e *INPUTOUTPUT*. Estas opções fazem com que a regra seja aplicada para os pacotes que estão chegando no dispositivo, opção de *INPUT*, ou que estão saindo do dispositivo, opção de *OUTPUT*. Caso seja configurado a opção conjunta, serão criadas duas regras, uma com cada opção de direção.

A figura abaixo demonstra como é feita a aplicação de uma regra:

The screenshot shows the Altus 2020 Firewall configuration interface. The 'Gerenciamento' tab is selected, displaying the 'Regras de Usuário' configuration form. A red box highlights the 'Adicionar à lista' button. Below the form is a table titled 'Lista de Regras' with columns for ID, Ação, and Regras de Usuário. A red arrow points to the 'Aplicar' button.

Figura 182: Tabela de Aplicação de Regras de Usuário do Firewall

Após preencher os campos conforme deseja configurar a regra de Firewall, deve-se clicar no botão *Adicionar à lista*. Ao fazer isso todas as configurações serão analisadas para conferir se há valores inválidos ou se há alguma regra duplicada. Não é possível adicionar duas regras com os mesmos parâmetros de endereços, máscaras, interface, portas e direção. Caso algum conflito seja encontrado, será exibido uma mensagem indicando o campo que houve uma configuração inválida, ou ainda, o ID da regra presente na tabela cujas configurações ocasionaram o conflito com a nova regra configurada.

Após serem verificados todos os parâmetros, a regra será adicionada à lista abaixo da tabela de configuração. Esta lista se expande de forma automática, conforme são adicionadas ou excluídas regras. Caso queira excluir uma regra da lista, basta posicionar o mouse sobre a regra que deseja excluir. Ao fazer isto, será exibido um botão X, da cor vermelha, conforme mostrado na figura anterior. Ao clicar nele, a regra será excluída da tabela.

Ao adicionar novas regras, ou excluir uma existente na tabela das regras, deve-se clicar no botão de *Aplicar*, que aparece mais abaixo, para que a configuração seja aplicada no dispositivo.

ATENÇÃO

Durante a aplicação das regras de firewall pode haver uma momentânea instabilidade na comunicação Ethernet.

5.14.3.5.3. Modo Switch ou NIC Teaming

Dispositivos configurados com interfaces Ethernet em modo Switch ou NIC Teaming devem ter as regras de usuário atribuídas sempre à primeira interface do par. Assim, as regras de usuário serão corretamente aplicadas nas duas interfaces que formam o par Switch ou Nic Teaming.

ATENÇÃO

Regras de usuário atribuídas à segunda interface do par serão ignoradas.

5.14.3.6. OpenVPN

VPN (Virtual Private Network) é uma sigla para Rede Virtual Privada, utilizada para navegar em redes não seguras, trafegando dados importantes ou, simplesmente, realizando o acesso à internet com um nível elevado de privacidade. A rede virtual da VPN pode ser compreendida como um túnel no qual as informações trafegam de forma segura, protegidas por certificados e chaves de segurança. O OpenVPN é um serviço do tipo *open source*, ou seja, gratuito para ser utilizado e distribuído, e com o seu código fonte aberto para que sejam realizadas modificações, caso sejam necessárias.

O principal objetivo da VPN é realizar uma comunicação de forma segura através de uma rede não segura. Para que isso seja possível, é utilizada a encriptação dos dados com base em certificados e chaves gerados utilizando o TLS, Transport Layer Security, um protocolo que realiza encriptações de 256 bits, uma das mais seguras.

Para realizar a configuração de um cliente ou servidor OpenVPN, foi criada a página OpenVPN, na aba *Gerenciamento* da página Web de Sistema da UCP. Conforme é mostrado na figura abaixo.

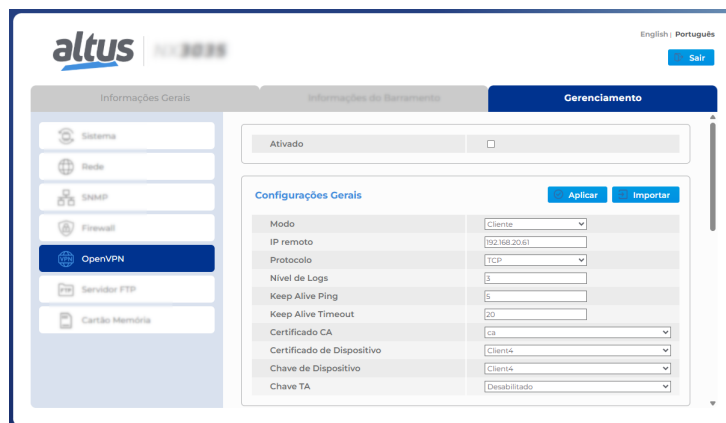


Figura 183: Tela de Configuração de OpenVPN

Por estar localizada dentro da aba *Gerenciamento*, o acesso a esta página é protegido por senha. As próximas seções descrevem as configurações e funcionalidades desta página.

■ Importação de Configurações

Para realizar a configuração da VPN de forma rápida e fácil no seu dispositivo, pode-se utilizar o botão *Importar* que aparece na figura 183 no canto superior direito da página. Ao clicar sobre este botão, é aberta uma janela, do explorador de arquivos, para que seja feita a seleção de um arquivo de configuração. Devem ser selecionados os arquivos com extensão *conf* ou *ovpn*. Ao selecionar um arquivo, o seu conteúdo será lido e os parâmetros de configuração que estiverem presentes preencherão os seus respectivos campos de configuração na página web.

Para que os parâmetros, do arquivo, sejam interpretados corretamente, eles devem seguir o padrão de sintaxe de arquivos de configuração do OpenVPN.

Caso existam arquivos de segurança, certificados ou chaves, escritos no arquivo de configuração, junto aos demais parâmetros, eles serão lidos e separados em arquivos distintos dentro do controlador para serem utilizados.

ATENÇÃO

Não devem ser utilizados espaços para separação das palavras no nome dos arquivos ".conf". Em vez disso, utilize "_" para separá-las.

5.14.3.6.1. Configuração do OpenVPN

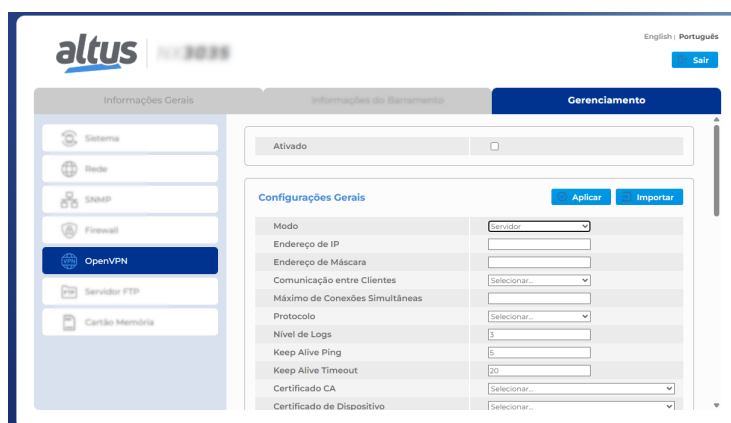


Figura 184: Tabela de Configurações de Servidor OpenVPN

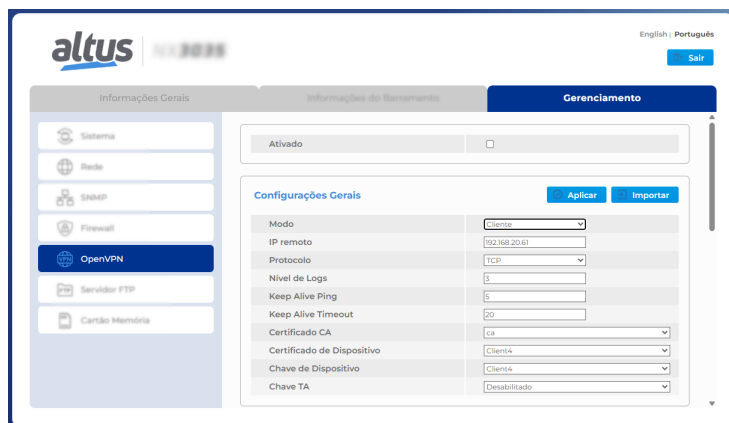


Figura 185: Tabela de Configurações de Cliente OpenVPN

Esta seção mostra como é realizada a configuração do OpenVPN. As configurações serão divididas em três partes: as comuns para ambos os modos de operação, as configurações exclusivas de um servidor e as configurações exclusivas de um cliente.

■ Configurações Comuns

Observando as figuras com as configurações de cliente, figura 185, e a de servidor, figura 184, é possível identificar que diversos parâmetros são os mesmos para ambas as configurações. São eles:

1. Modo

Com a configuração do *Modo* é possível selecionar entre duas opções, cliente ou servidor. Ao realizar a seleção de algum dos dois modos, a tabela de configurações se modifica de forma automática para permitir a configuração dos campos necessário para cada modo de operação.

2. Protocolo

Este campo configura qual será o protocolo de transporte a ser utilizado para a comunicação da VPN. É possível configurar entre UDP e TCP.

ATENÇÃO

A configuração do servidor e de todos os seus clientes deve ser a mesma. Com uma configuração divergente, o OpenVPN não é capaz de realizar a comunicação.

3. Nível de Logs

Este campo configura qual será o nível que o arquivo de logs receberá. A configuração varia de 0 até 5, sendo 0 o nível mais básico e 5 o nível mais avançado.

O nível 0 exibe logs somente sobre alguma falha crítica no OpenVPN e os níveis a partir de 4 são utilizados para depuração, pois há uma quantidade muito grande de informações sendo escritas no arquivo de logs. Para uma operação normal, recomenda-se a utilização do valor 3.

Este campo aceita somente números como entrada. Não é permitido utilizar letras e nem caracteres especiais.

4. Keep Alive Ping

Este campo configura qual o tempo, *em segundos*, em que o será encaminhada uma requisição de *Ping*. Esta requisição serve para verificar a conexão entre o servidor e os clientes.

Este parâmetro pode ser configurado tanto no servidor, quanto nos clientes do OpenVPN, porém, caso este parâmetro seja configurado no servidor, os clientes irão assumir o valor do servidor e não o valor configurado neles. Se o Servidor não possuir tal configuração, cada cliente assume a sua configuração normalmente. Caso deseje desabilitar o ping entre o servidor e os clientes, configure o valor 0.

Este campo aceita somente números como entrada. Não é permitido utilizar letras e nem caracteres especiais.

5. Keep Alive Timeout

Este campo configura o tempo, *em segundos*, em que ocorrerá o timeout da requisição de *Ping*. Após o término deste tempo, sem uma resposta do outro dispositivo VPN, ele será considerado desconectado.

Este parâmetro pode ser configurado tanto no servidor, quanto nos clientes do OpenVPN, porém, caso este parâmetro seja configurado no servidor, os clientes irão assumir metade do valor do servidor e não o valor configurado neles. Os clientes recebem metade do valor para garantir que eles estejam desconectados no caso de o servidor desconectar. Se o Servidor não possuir tal configuração, cada cliente assume a sua configuração normalmente. Caso deseje desabilitar esta funcionalidade, configure o valor 0.

Este campo aceita somente números como entrada. Não é permitido utilizar letras e nem caracteres especiais.

6. Arquivos de Segurança

Nos campos *Certificado CA*, *Certificado de Dispositivo*, *Chave de Dispositivo* e *Chave TA*, deve ser selecionado qual o arquivo de segurança, certificado ou chave, será utilizado para estabelecer a comunicação do OpenVPN. As opções de cada campo, *combobox*, são filtradas de acordo com o tipo de arquivo chave ou certificado, embora não exista diferenciação das chaves entre si e nem dos certificados entre si.

Para que seja possível selecionar algum arquivo é necessário que este tenha sido importado anteriormente.

Todos os arquivos de segurança são obrigatórios para que seja estabelecida a comunicação correta entre os clientes e o servidor VPN, exceto a *Chave TA*. Esta chave é opcional para realizar a comunicação, porém, caso ela seja utilizada no servidor, ela se torna de uso obrigatório para todos os clientes deste mesmo servidor.

Consulte a seção [Gerenciamento de Certificados e Chaves TLS](#), para maiores informações sobre a geração de certificados e chaves de segurança com base no TLS.

7. Chave TA

No campo *Chave TA* é configurada qual será o tipo de criptografia aplicada para a *Chave TA*. Este campo se mantém oculto até que seja selecionado algum arquivo para a chave TLS, pois, ele somente é utilizado em conjunto com esta chave. O valor padrão deste parâmetro é *SHA1*, porém é possível selecionar entre os seguintes valores: *SHA256*, *SHA512* e *MD5*, além do padrão *SHA1*.

ATENÇÃO

É necessário que esta configuração seja igual entre os clientes e o servidor da mesma rede OpenVPN. Caso o valor deste campo seja diferente entre cliente e servidor, a conexão não será estabelecida.

■ Configurações Exclusivas do Servidor

As configurações exclusivas do servidor, observadas na figura 184, são descritas a seguir.

1. Endereço de Rede

A faixa de IP que será utilizada para atribuir os endereços do servidor e dos clientes da rede VPN é configurado pelo servidor através da configuração dos campos *Endereço de IP* e *Endereço de Máscara*. Todos os IPs que serão atribuídos para os clientes e ao servidor serão retirados da faixa especificada.

O endereço IP do servidor é sempre o endereço inicial da faixa configurada, para a atribuição de IP dos clientes, são utilizados os valores ainda disponíveis da faixa. Por exemplo, caso seja configurada uma rede com o endereço IP 10.8.12.4 e máscara 255.255.255.248, o servidor assumirá o IP 10.8.12.5 que é o primeiro endereço disponível da faixa configurada. Porém, caso seja configurada a máscara 255.255.255.0, o servidor assumirá o IP 10.8.12.1, que é o primeiro endereço disponível da faixa.

Os campos de endereço IP e máscara, somente aceitam as configurações que tenham a sintaxe de um endereço de IP e endereço de máscara, respectivamente. Caso seja configurado algo fora do padrão, será exibida uma mensagem de alerta, informando que houve algum erro.

2. Comunicação entre Clientes

Neste campo é possível habilitar ou desabilitar a comunicação entre os clientes da rede VPN. Quando a opção for selecionada como *Desativado*, somente é possível realizar a comunicação cliente-servidor diretamente. Se a opção for selecionada como *Ativado*, será permitida além da comunicação cliente-servidor, a comunicação entre os próprios clientes.

3. Máximos Clientes Conectados

Neste campo é possível configurar qual é o número máximo de clientes que podem se conectar com o servidor OpenVPN simultaneamente. Este campo aceita somente caracteres numéricos e o valor mínimo para ele é 1.

4. Redes Privadas

Ao selecionar o modo de operação do OpenVPN como servidor, será exibido uma tabela, normalmente oculta, que permite a configuração de redes privadas que podem estar abaixo do servidor e de cada cliente.

Figura 186: Tabela de Configurações de Redes Privadas OpenVPN

Para realizar a configuração de uma rede privada que está abaixo do servidor, basta selecionar o tipo da rede como *Servidor* e configurar os endereços de rede e máscara. A configuração de uma rede privada de um cliente necessita, além de configurar o tipo como *Cliente*, que seja informado o *Common Name* do cliente que possui a rede que está sendo configurada.

O Common Name de um cliente é configurado na geração do *Certificado de Dispositivo*. Este parâmetro é informado na criação do certificado e é único para cada cliente e cada servidor. A configuração destas redes privadas cria uma tabela de roteamento que será verificada ao receber ou enviar pacotes pela VPN.

A figura acima, mostra uma configuração de uma sub-rede 80 no servidor OpenVPN, logo, será configurada uma regra de roteamento que encaminhará os pacotes de dados, recebidos pela VPN, para a interface do dispositivo configurada nesta rede. Também é criada uma regra, interna do servidor, que caso um pacote de dado possua a sub-rede 70, este pacote será roteado e encaminhado pelo túnel de VPN. O mesmo comportamento ocorre com o cliente *client2*, porém, com as sub-redes trocadas, pois, abaixo deste cliente está a sub-rede 70 e ele encaminhará para o túnel VPN os pacotes com a sub-rede 80.

Veja na figura a seguir um exemplo de arquitetura:

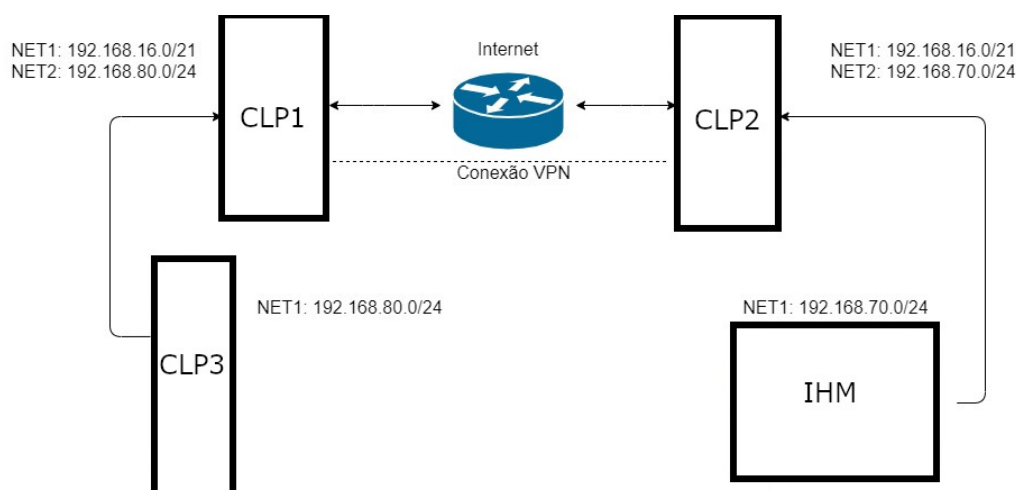


Figura 187: Exemplo de Arquitetura com Redes Privadas

Na Figura 187, o CLP1 da esquerda possui a rede privada 80 configurada na sua NET 2 e conectado a ele existe um CLP3 na mesma rede. O CLP2 da direita possui uma rede privada 70 configurada na NET 2 e conectado a ele existe uma IHM, na mesma rede. A arquitetura exemplo realiza a comunicação entre os dispositivos CLP3 e IHM pela VPN, através da configuração das suas respectivas redes privadas.

Para que as configurações presentes na tabela sejam aplicadas no dispositivo é preciso clicar no botão *Aplicar* e confirmar a operação na janela de confirmação que irá surgir. Ao serem aplicadas as regras, será exibida uma mensagem indicando se houve êxito ou falha na operação.

■ Configurações Exclusivas de Cliente

Existe somente uma única configuração exclusiva de clientes OpenVPN na página, que pode ser visualizada na figura 185. Esta configuração é o *IP Remoto*.

1. IP Remoto

O campo de *IP Remoto*, configura qual é o endereço cujo o servidor VPN está esperando a comunicação dos clientes. Caso seja estabelecido um servidor OpenVPN em um computador, a configuração de IP remoto deve ser realizada conforme o endereço IP deste computador. Este campo também aceita *host names* como endereço remoto, portanto, é possível configurar um IP ou um host name neste parâmetro.

ATENÇÃO

Em função da necessidade de permitir parâmetros tão diferentes, IPs e host names, a única verificação existente neste campo é sobre a existência ou não de dados. Tenha atenção ao realizar a configuração.

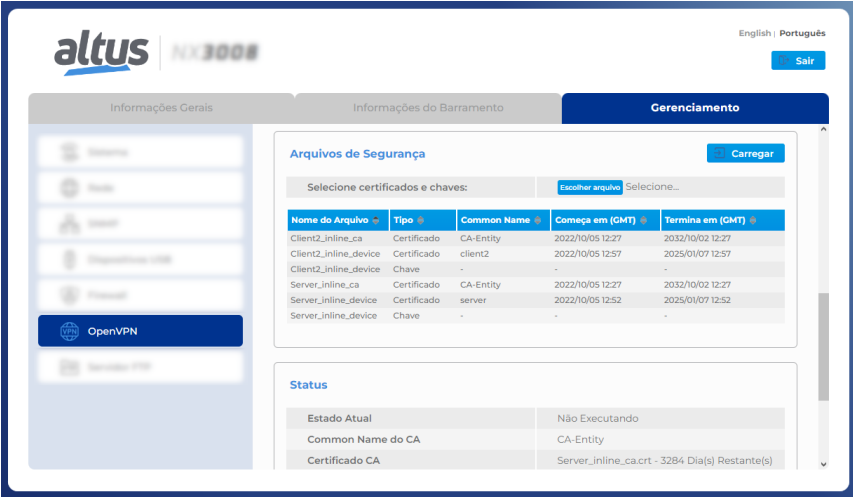
2. Aplicação de Configurações

Para habilitar a funcionalidade, deve-se marcar o checkbox *Ativado*, exibido na figura 185. Caso deseje apenas aplicar as configurações realizadas e não habilitar o OpenVPN, desmarque este checkbox.

Após realizar todas as configurações desejadas, as configurações devem ser aplicadas no dispositivo, para isso utilize o botão *Aplicar*. Este botão é mostrado na figura 185. Quando as configurações são aplicadas e a VPN está habilitada, a seção *Firewall* irá realizar um *scroll* automático até a tabela de *status* do OpenVPN, exibida na seção *Status*.

5.14.3.6.2. Arquivos de Segurança

Os arquivos de segurança são utilizados para estabelecer a comunicação do OpenVPN de forma segura, realizando o papel de criptografar e de descriptografar os pacotes de dados que trafegam pelo túnel da VPN. Na seção [Gerenciamento de Certificados e Chaves TLS](#), é descrito como gerar chaves e certificados TLS. A figura a seguir mostra a seção responsável pelo gerenciamento dos arquivos de segurança:



Nome do Arquivo	Tipo	Common Name	Começa em (GMT)	Termina em (GMT)
Client2_inline_ca	Certificado	CA-Entity	2022/10/05 12:27	2032/10/02 12:27
Client2_inline_device	Certificado	client2	2022/10/05 12:57	2025/01/07 12:57
Client2_inline_device	Chave	-	-	-
Server_inline_ca	Certificado	CA-Entity	2022/10/05 12:27	2032/10/02 12:27
Server_inline_device	Certificado	server	2022/10/05 12:52	2025/01/07 12:52
Server_inline_device	Chave	-	-	-

Status	
Estado Atual	Não Executando
Common Name do CA	CA-Entity
Certificado CA	Server_inline_ca.crt - 3284 Dia(s) Restante(s)

Figura 188: Tabela de Arquivos de Segurança OpenVPN

Nesta seção da Página Web de Sistema é possível realizar o gerenciamento dos arquivos de segurança. É possível realizar a importação de arquivos, monitorar a validade dos certificados, realizar o download dos arquivos carregados no dispositivo e excluir os arquivos que foram carregados.

Ao clicar no botão *Escolher arquivos*, é possível realizar importação de certificados e chaves, estes arquivos devem estar com as respectivas extensões *.crt* e *.key*. Este botão abre uma janela do explorador de arquivos e permite a seleção de um ou mais arquivos.

ATENÇÃO

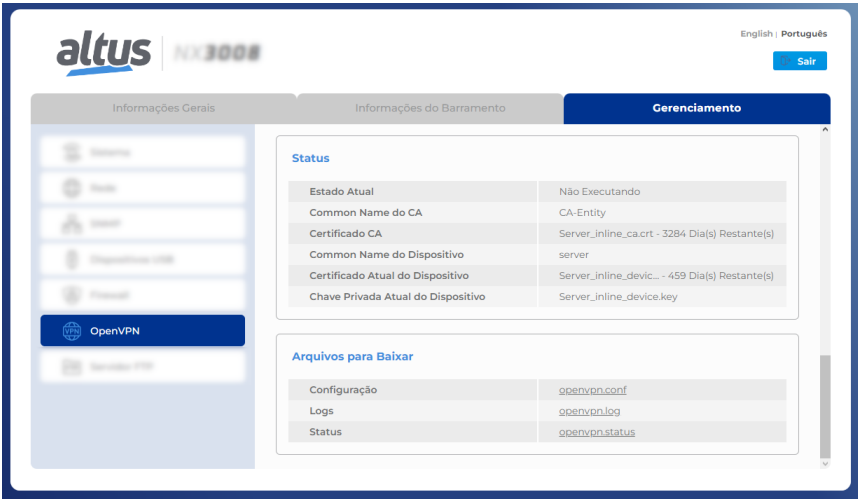
Há um limite de importação de 12 arquivos para o controlador.

O controle dos arquivos é feito na tabela, que é mostrada na figura acima. Esta tabela adiciona novos itens, ou remove, conforme forem ocorrendo as operações de importar, ou de excluir arquivos. É possível identificar se o arquivo é uma chave ou um certificado através do segundo item da lista, o *Tipo*, que indica o que é aquele arquivo. Para os certificados, também são exibidos os seus *commons names* e as suas datas de validade, tanto de início, quanto de expiração.

É possível recuperar um arquivo que foi importado para a peça e, também, excluí-lo. Quando é arrastado o mouse sobre um arquivo da tabela, são exibidos dois botões, um para o download e outro para a exclusão.

5.14.3.6.3. Status

Desenvolvida para permitir um monitoramento de dados, a tabela de status do OpenVPN se expande de forma automática, conforme é alterado alguma configuração e exibe diversos dados sobre a conexão, como o estado da VPN, o IP de VPN atribuído àquele dispositivo, os dados trafegados e os arquivos de segurança que estão sendo utilizados na comunicação.



Status	
Estado Atual	Não Executando
Common Name do CA	CA-Entity
Certificado CA	Server_inline_ca.crt - 3284 Dia(s) Restante(s)
Common Name do Dispositivo	server
Certificado Atual do Dispositivo	Server_inline_devic... - 459 Dia(s) Restante(s)
Chave Privada Atual do Dispositivo	Server_inline_device.key

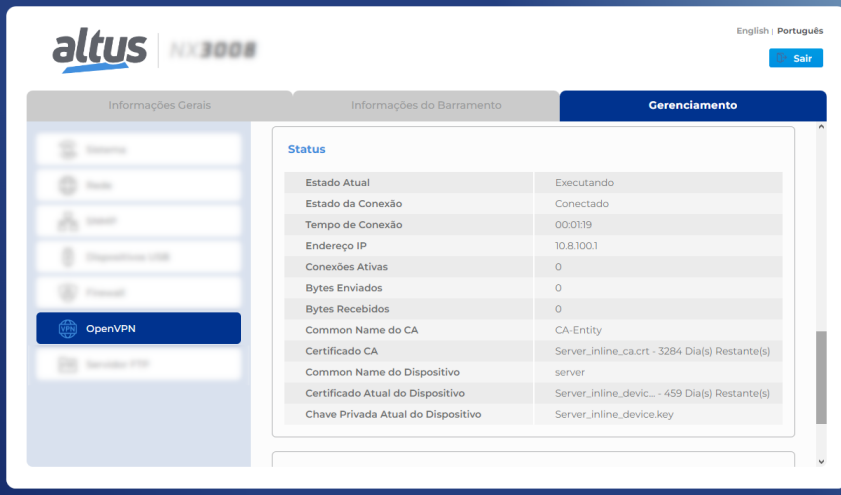
Arquivos para Baixar	
Configuração	openvpn.conf
Logs	openvpn.log
Status	openvpn.status

Figura 189: Tabela de Status OpenVPN com Funcionalidade Desativada

Quando a VPN está desativada, a tabela possui poucos parâmetros. O campo *Estado Atual* indica se a VPN está ativada ou não, e os demais campos exibem qual os certificados e chaves que estão configurados para a comunicação da VPN. Caso não tenha sido selecionado algum dos arquivos de segurança, no lugar do seu nome será exibido o caractere "-", indicando que não há um arquivo configurado.

Os campos de common name, tanto do CA quanto do dispositivo, exibem os commons names dados aos respectivos certificados, de autoridade certificadora e de dispositivo.

Ao lado do nome do arquivo, de cada certificado, é exibido o tempo restante, *em dias*, até a data da sua expiração.



The screenshot shows the Altus NX3008 web interface. The left sidebar contains navigation links: Home, Config, Logs, OpenVPN, and Settings. The main content area is titled 'Gerenciamento' and displays a 'Status' table for OpenVPN.

Status	
Estado Atual	Executando
Estado da Conexão	Conectado
Tempo de Conexão	00:01:19
Endereço IP	10.8.100.1
Conexões Ativas	0
Bytes Enviados	0
Bytes Recebidos	0
Common Name do CA	CA-Entity
Certificado CA	Server_inline_ca.crt - 3284 Dia(s) Restante(s)
Common Name do Dispositivo	server
Certificado Atual do Dispositivo	Server_inline_devic... - 459 Dia(s) Restante(s)
Chave Privada Atual do Dispositivo	Server_inline_device.key

Figura 190: Tabela de Status OpenVPN com Funcionalidade Ativada

Quando a funcionalidade é ativada e as configurações são aplicadas no dispositivo, a tabela tem as suas células dinamicamente modificadas para que as demais informações sejam exibidas. As informações sobre o estado da conexão do OpenVPN, podem ser consultadas nos dois primeiros tópicos da lista.

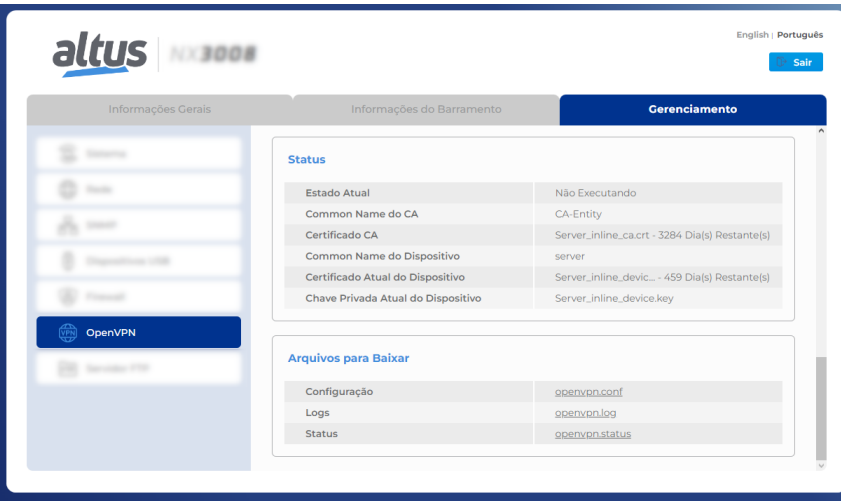
O item *Estado Atual* possui os estados de *Não Executando*, *Iniciando serviço...* e *Executando*, que indicam, respectivamente, que a VPN está desativada, que está iniciando, e que ela está ativada.

O item *Estado da Conexão* possui os estados de *Não conectado*, *Conectando...* e *Conectado*.

As demais informações que podem ser obtidas nesta tabela são, o tempo total de conexão, o endereço IP do dispositivo e as quantidades de dados enviados e recebidos, em bytes. O status de quantos clientes estão atualmente conectados, somente é exibido quando a OpenVPN estiver operando como um servidor.

5.14.3.6.4. Arquivos para Baixar

É possível conferir as informações geradas pelo OpenVPN, através de arquivos de status e de log. A lista de arquivos para download somente é exibida quando há algum arquivo para ser baixado, caso não exista nenhum, a mensagem "Nenhum arquivo encontrado no controlador!" é exibida. Ao clicar sobre qualquer um dos links, será realizado o download do arquivo solicitado, através do navegador.



The screenshot shows the Altus NX3008 web interface. The left sidebar is the same as in Figure 190. The main content area is titled 'Gerenciamento' and displays a 'Status' table and a 'Arquivos para Baixar' table.

Status	
Estado Atual	Não Executando
Common Name do CA	CA-Entity
Certificado CA	Server_inline_ca.crt - 3284 Dia(s) Restante(s)
Common Name do Dispositivo	server
Certificado Atual do Dispositivo	Server_inline_devic... - 459 Dia(s) Restante(s)
Chave Privada Atual do Dispositivo	Server_inline_device.key

Arquivos para Baixar	
Configuração	openvpn.conf
Logs	openvpn.log
Status	openvpn.status

Figura 191: Seção de Downloads OpenVPN

5.14.3.6.5. Configuração de Arquiteturas

Nesta seção serão abordadas algumas possibilidades de configurações para o OpenVPN, como as arquiteturas Host-to-Host, Host-to-Site e Site-to-Site.

1. Host-to-Host



Figura 192: Exemplo de arquitetura Host-to-Host

Esta topologia permite a conexão entre dois hosts VPN. Ambos os hosts podem ser escolhidos para serem configurados como o servidor, logo, o outro deverá ser configurado como cliente, ou ainda, ambos os hosts podem ser configurados como clientes e haver um terceiro host que será o servidor da rede VPN.

A configuração deste tipo de arquitetura não exige nenhuma configuração específica, ou seja, não há restrição quanto às configurações disponíveis na seção *OpenVPN* da Página Web de Sistema.

2. Host-to-Site

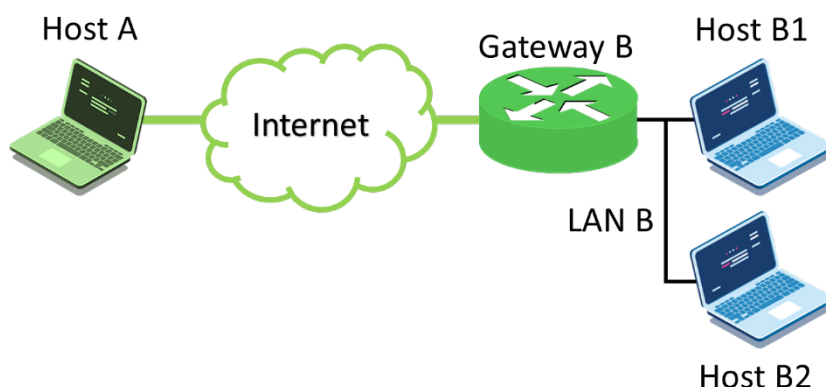


Figura 193: Exemplo de topologia Host-to-Site

Esta topologia permite a conexão entre dois hosts VPN, porém, um destes hosts atua também como um gateway para a rede VPN. Através deste gateway é realizado o roteamento para que seja estabelecida a comunicação entre os hosts A, B1, B2 e Gateway B. Neste cenário, tanto o Host A, quanto o Gateway B podem assumir o papel de servidor, sendo que, quando um é o servidor da rede, o outro será o cliente.

Os hosts, B1 e B2, que estão em uma rede privada Lan B, abaixo do Gateway B, não precisam ter suporte ao OpenVPN para conseguir se comunicar, pois, toda a comunicação é gerida pelo gateway da rede VPN.

Para que a comunicação entre todos os dispositivos da rede seja possível, é necessário criar regras de roteamento para o túnel da VPN. Consultar a seção [Configuração do OpenVPN](#) para verificar como criar regras de redes privadas.

Esta arquitetura de conexão VPN exige algumas configurações específicas. É necessário que o servidor tenha a sua configuração de topologia como subnet, sendo esta a configuração padrão do controlador, para que seja possível configurar as redes privadas abaixo do Gateway B, conforme observado na imagem acima.

Também é preciso informar o endereço da rede privada, Lan B, que estará realizando a comunicação através da VPN. Esta configuração é feita utilizando o comando `push "route IP_da_Lan_B Mascara_da_Lan_B"`, e é obrigatória independente se a rede privada está localizada abaixo do cliente ou do servidor do OpenVPN, porém, caso a rede privada esteja abaixo do cliente VPN, é necessário adicionar, além deste comando, a seguinte configuração: `route IP_da_Lan_B Mascara_da_Lan_B`. O comando `route` informa o servidor sobre qual é a rede privada que está conectada a rede da VPN

e o comando utilizando o *push* faz com que os clientes daquele servidor VPN obtenham a mesma informação. Estas configurações são escritas no arquivo de configuração do servidor VPN.

3. Site-to-Site

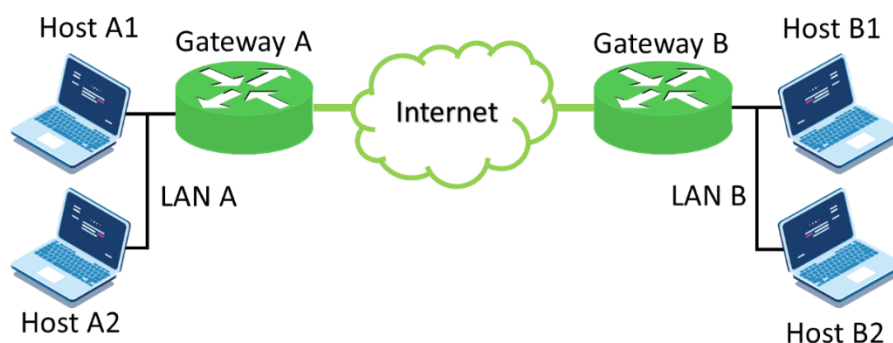


Figura 194: Exemplo de topologia Site-to-Site

Esta topologia permite a conexão entre dois hosts VPN, sendo que ambos atuam como gateways para a rede VPN. Através destes gateways é realizado o acesso para que seja estabelecida a comunicação entre os hosts A1, A2, B1, B2, Gateway A e Gateway B. Neste cenário, qualquer gateway pode assumir o papel de servidor, logo, o outro será o cliente. Nenhum dos hosts que estão em uma rede privada, abaixo de um dos dois gateways, precisam ter suporte ao OpenVPN para conseguir se comunicar, pois, toda a comunicação é gerida pelos gateways da rede VPN.

Para que a comunicação entre todos os dispositivos da rede seja possível, é necessário criar regras de roteamento para o túnel da VPN. Consultar a seção [Configuração do OpenVPN](#) para verificar como criar regras de redes privadas.

As configurações para esta arquitetura precisam das mesmas configurações específicas descritas na seção [Configuração de Arquiteturas](#), com a diferença de que, nesta existem duas redes privadas e as duas devem seguir a configuração que foi demonstrada. Admitindo que o Gateway A é o servidor nesta conexão, deve ser adicionado os seguintes comandos ao arquivo de configuração: *push "route IP_da_Lan_A Mascara_da_Lan_A"*, *route IP_da_Lan_B Mascara_da_Lan_B* e *push "route IP_da_Lan_B Mascara_da_Lan_B"*. Caso o servidor seja o Gateway B, no arquivo de configuração seria adicionado: *push "route IP_da_Lan_B Mascara_da_Lan_B"*, *route IP_da_Lan_A Mascara_da_Lan_A* e *push "route IP_da_Lan_A Mascara_da_Lan_A"*.

5.14.3.7. Servidor FTP

O FTP (File Transfer Protocol) é um protocolo que permite a transferência de arquivos entre dispositivos. Opera no modelo cliente-servidor, onde a UCP torna-se um Servidor FTP, armazenando arquivos para que Clientes FTP possam acessá-los para transferência, isto é, download e upload.

A conexão FTP de controle é uma conexão TCP, estabelecida através da porta 21 do Servidor FTP. Onde o Cliente e o Servidor trocam comandos e repostas para o gerenciamento da sessão de transferência de arquivos.

Através do protocolo FTP, o Cliente FTP consegue ler e escrever arquivos que estão, tanto armazenados na memória interna da UCP, como também nas memórias externas (cartão de memória, por exemplo), caso presentes na arquitetura. O tamanho máximo do arquivo que pode ser transferido varia de acordo com a quantidade de memória disponível, entre memória interna e memórias externas. Ao atingir o limite de memória, a transferência irá parar, e caso o arquivo não tenha sido transferido por completo, o mesmo ficará corrompido.

ATENÇÃO

Download e upload de grandes arquivos via FTP, tanto da memória interna, quanto das memórias externas, podem afetar consideravelmente na performance da UCP.

ATENÇÃO

O FTP Server não possui suporte à comunicação via Explorador de Arquivos do Windows, sendo necessário a utilização de um software Cliente FTP para realizar o acesso.

A configuração do Servidor FTP é feita através de uma seção dedicada, localizada na aba *Gerenciamento* da Página Web de Sistema do controlador, conforme mostra a figura abaixo.

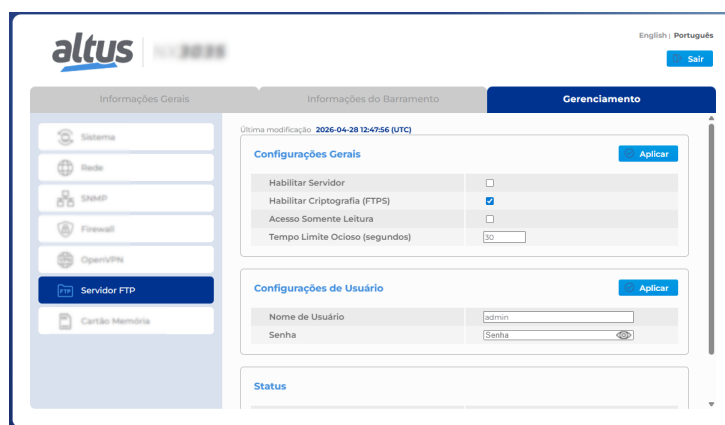


Figura 195: Tela de Configuração do Servidor FTP

O FTP trata-se de uma funcionalidade à parte ao Mastertool, isto é, não necessita de interação alguma com o Mastertool. As configurações aplicadas na Página Web de Sistema passam a valer quando confirmadas através do botão *Aplicar* e, são salvas automaticamente no controlador. Contanto que a funcionalidade esteja habilitada, voltará a operar mesmo após a reinicialização do dispositivo.

ATENÇÃO

Para correto funcionamento do FTP, é necessária a utilização de Clientes FTP no modo Ativo. O modo Passivo não é suportado.

5.14.3.7.1. Configurações Gerais

1. Habilitar Servidor

Permite habilitar e desabilitar a funcionalidade. Quando o Servidor FTP está habilitado, o serviço torna-se disponível e as demais configurações realizadas são aplicadas. Caso o Servidor FTP esteja desabilitado, apesar de o serviço estar indisponível, suas configurações são armazenadas. Por padrão, o Servidor FTP é desabilitado.

2. Habilitar Comunicação Criptografada (FTPS)

Permite habilitar e desabilitar a comunicação criptografada. Essa comunicação é feita através do *FTPS Explícito*, também conhecido como *FTPES*. Trata-se de uma extensão segura do protocolo FTP, que adiciona uma camada de criptografia à transferência.

Neste tipo de comunicação, quando estabelecida uma conexão cliente-servidor, em vez de iniciar imediatamente a transferência de dados, o Cliente FTP envia um comando *AUTH SSL*, para solicitar a conexão segura. Então, ao habilitar a comunicação criptografada, a UCP gera um certificado *self-signed* para garantir a comunicação utilizando o protocolo SSL (Secure Sockets Layer).

Quando o Cliente FTP realizar a conexão autenticada, o certificado e suas respectivas informações serão exibidos, permitindo estabelecer a comunicação FTPS Explícita (FTP sobre SSL), através da mesma porta 21.

ATENÇÃO

O certificado FTPES tem uma validade de um ano a partir da data de criação. Para gerar um novo certificado, basta aplicar novamente a configuração.

3. Acesso Somente Leitura

Permite habilitar e desabilitar a limitação do acesso de escrita e leitura no Servidor. Por padrão, o parâmetro é habilitado. Quando o parâmetro está habilitado, o Cliente FTP tem acesso somente a leitura, sem a possibilidade de adicionar ou remover qualquer arquivo do Servidor FTP. Neste caso, a única operação permitida é o upload de arquivos, ou seja, transferência do Servidor para o Cliente. Já quando o parâmetro está desabilitado, é possível realizar todas as operações.

4. Tempo Limite Ocioso (Segundos)

Define o tempo máximo em que a conexão será mantida antes do Servidor FTP fechá-la por inatividade. Isto é, após

aberta a conexão com um Cliente FTP, se não houver nenhuma atividade, após passado o Tempo Limite Ocioso, a conexão com o Cliente é fechada pelo Servidor.

O parâmetro pode ser configurado com valores entre 10 e 60 segundos sendo, por padrão, 60 segundos.

ATENÇÃO

O tempo limite ocioso configurado no servidor deve ser igual ou menor ao do cliente.

5.14.3.7.2. Configurações de Usuário

1. Nome de Usuário

Usuário ou *Nome de Usuário* solicitado para que o Cliente realize a conexão com o Servidor.

Caso seja realizada uma nova configuração, a anterior é removida. Isto é, há apenas um único usuário FTP, porém, este pode ser utilizado em multiplas conexões.

2. Senha

Gerencia a chave de autenticação para que o Cliente FTP possa se conectar com o Servidor FTP.

Não há recuperação de senha. Caso esta tenha sido perdida, é necessário adicionar uma nova configuração de usuário.

Item Configurável	Tamanho Mínimo	Tamanho Máximo	Caracteres Permitidos	Configuração Padrão
Usuário	4	30	[a-z][A-Z][0-9]@\$*_	admin
Senha	4	30	[a-z][A-Z][0-9]@\$*_	admin

Tabela 197: Configurações de Usuário FTP

5.14.3.7.3. Status

1. Estado Atual

Exibe o estado atual do Servidor FTP. Os possíveis estados são: *"Executando"*, *"Não Executando"* e *"Reiniciando Serviço"*.

Cada configuração aplicada reiniciará o serviço.

2. Conexões Ativas

Mostra ao usuário o número atual de conexões ativas.

O Servidor FTP aceita no máximo duas conexões ativas simultaneamente. Alguns Clientes FTP, como por exemplo o *Filezilla*, utilizam um recurso chamado Transferência de Arquivos Multithread, para melhorar a eficiência e a velocidade da transferência. Este recurso, permite que o Cliente FTP abra mais de uma conexão para a transferência de um arquivo. Consequentemente, quando utilizado um Cliente FTP deste tipo, apenas um Cliente já soma duas conexões ativas no Servidor.

Já outros Clientes FTP como por exemplo através de terminal de comandos, ocupam apenas uma conexão ativa, assim possibilitando a conexão simultânea de dois Clientes FTP com o Servidor.

5.14.3.8. 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 arquivos.

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 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 quando o cartão é inserido, após a reinicialização da UCP ou através da página web do dispositivo.

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.

Foi desenvolvida uma página web para gerenciamento do cartão de memória. Dentre as funcionalidades oferecidas estão a formatação, opção para desmontar e remover o cartão e a possibilidade de habilitar e desabilitar a interface do cartão. Estas configurações foram desenvolvidas na seção "Cartão de Memória", na página web da UCP, dentro da aba de "Gerenciamento". Além das configurações, também são exibidas informações sobre o estado atual do dispositivo, espaço de armazenamento total e livre, ambos medidos em *kB*. A imagem abaixo apresenta a página inicial, com a interface habilitada e sem nenhum dispositivo conectado.

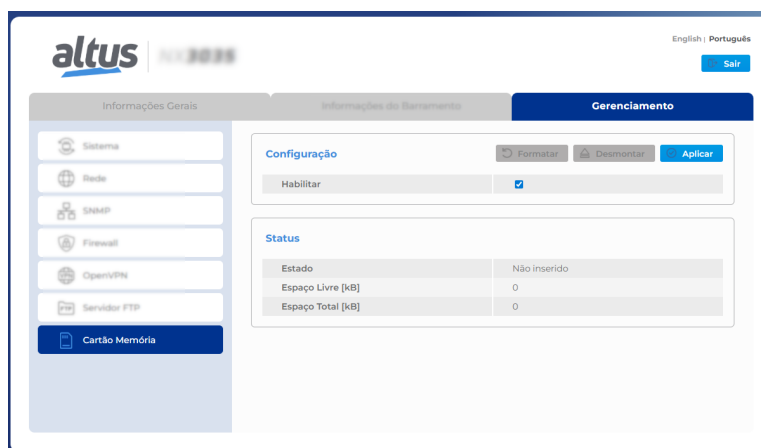


Figura 196: Cartão de Memória Página Inicial

Enquanto não há um cartão de memória inserido na UCP, os botões *Formatar* e *Desmontar* ficam bloqueados para uso. A tabela de *Status* indica o Estado "*Não Inserido*". Ao inserir um cartão de memória na UCP, o botão *Formatar* é habilitado para uso. Depois que o cartão é montado o botão *Desmontar* também fica disponível para ser utilizado. Ao inserir um cartão de memória na UCP, pode levar alguns instantes para que o cartão seja montado e as informações sejam atualizadas na página da web. A imagem abaixo apresenta a página web quando há um cartão conectado e montado na UCP.

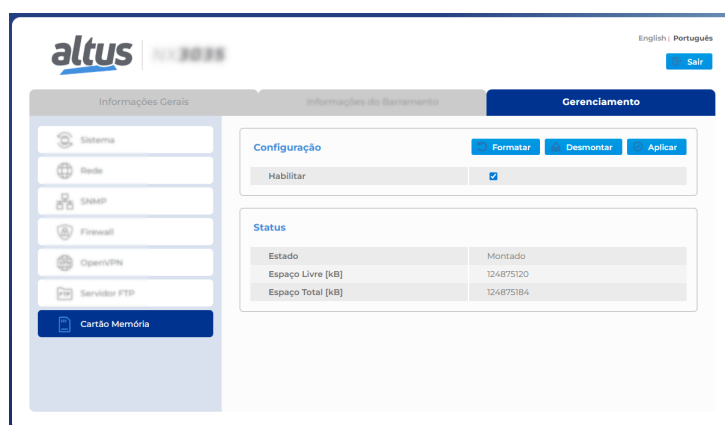


Figura 197: Cartão de Memória com Dispositivo Montado

ATENÇÃO

Para que o cartão de memória seja montado corretamente, deve haver pelo menos *1Kb* de espaço livre disponível na UCP. Se o cartão for inserido e não houver espaço livre suficiente, o cartão não será montado e a página web exibirá a mensagem "*Não Montado*".

5.14.3.8.1. Configuração

1. Formato Não Suportado

Se o dispositivo tiver um sistema de arquivos não suportado pela UCP, o estado "*Formato não suportado*" aparecerá

quando o dispositivo for inserido, conforme mostrado na figura abaixo. Nesse caso, o dispositivo não pode ser usado até que seja formatado com um sistema de arquivos suportado.

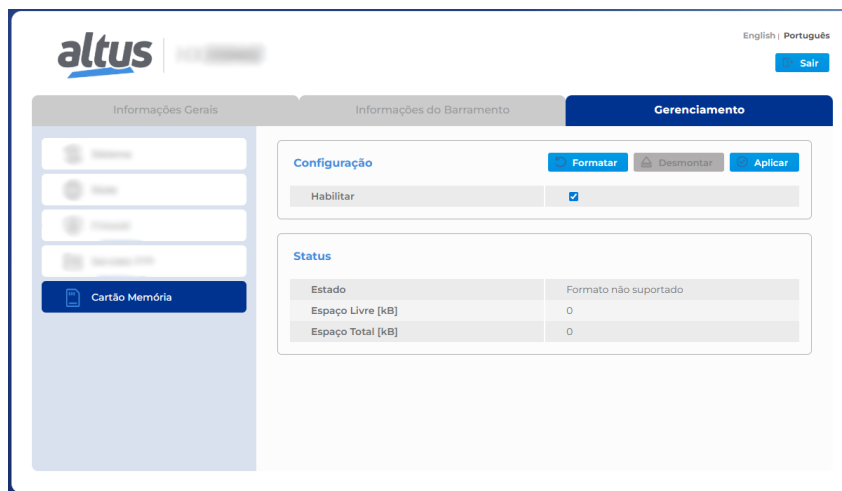


Figura 198: Mensagem de formato não suportado

Para formatar o dispositivo, o botão *Formatar* pode ser usado, conforme mostrado abaixo.

2. Formatar

Para formatar o dispositivo, utilize o botão *Formatar*. Ao clicar, será exibida uma mensagem, estilo *pop-up*, solicitando a confirmação da operação. A imagem abaixo apresenta esta mensagem.

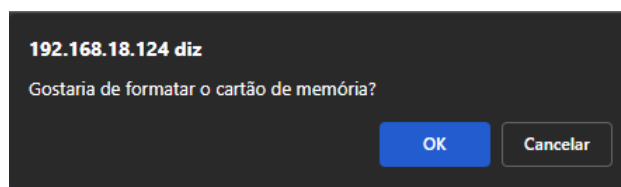


Figura 199: Mensagem Confirmação de Formatação

Ao confirmar no botão *OK*, a operação é iniciada, logo após, todas as configurações são bloqueadas. Os botões *Formatar*, *Desmontar* e *Aplicar* e também a caixa de seleção ficam indisponíveis durante a formatação. O processo de formatação é indicado na tabela de *Status*, com o valor do *Estado* alterado para *"Formatando..."*, como apresentado na figura abaixo.

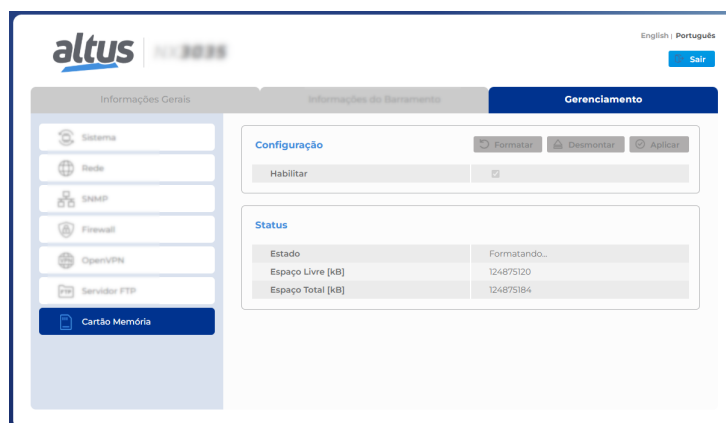


Figura 200: Cartão de Memória Formatando

Ao final do processo de formatação, é exibida uma mensagem indicando a finalização da operação no dispositivo. A figura a seguir apresenta esta mensagem.

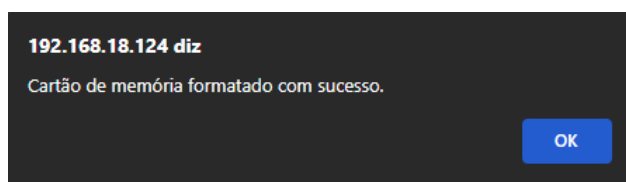


Figura 201: Mensagem de Formatação Concluída

Com a conclusão da operação, a página web retorna para o estado inicial, desbloqueando todos os botões e a caixa de seleção, da mesma forma que é apresentada na figura [Cartão de Memória com Dispositivo Montado](#)

3. Desmontar

Para desmontar e remover o dispositivo, utilize o botão *Desmontar*. Ao clicar, será exibida uma mensagem, estilo *pop-up*, solicitando a confirmação da operação. A imagem abaixo apresenta esta mensagem.

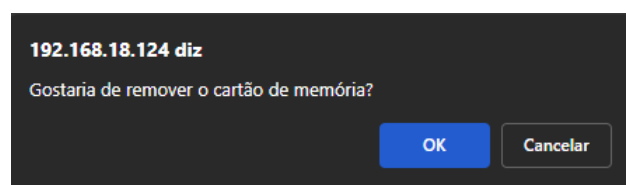


Figura 202: Mensagem Confirmação para Desmontar

Ao confirmar no botão *OK*, a operação é iniciada, logo após, todas as configurações são bloqueadas. Os botões *Formatar*, *Desmontar* e *Aplicar* e também a caixa de seleção ficam indisponíveis durante a desmontagem. O processo de desmontagem é indicado na tabela de *Status*, com o valor do *Estado* alterado para *Desmontando...*, como apresentado na figura abaixo.

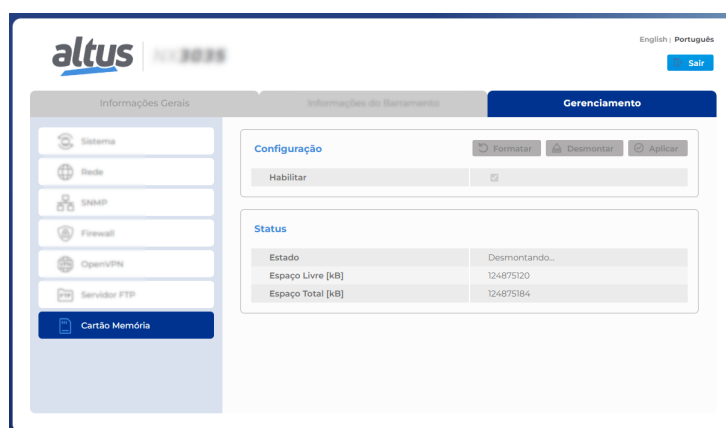


Figura 203: Cartão de Memória Desmontando

Ao final do processo de desmontagem, é exibida uma mensagem indicando a finalização da operação no dispositivo. A figura a seguir apresenta esta mensagem.

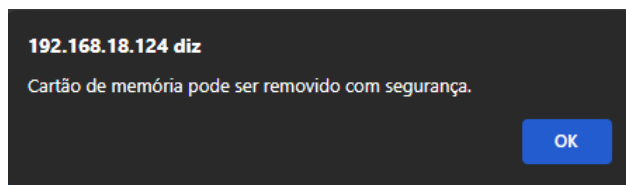


Figura 204: Mensagem de Desmontagem Concluída

Após a operação ser concluída com sucesso, os botões *Formatar* e *Aplicar* e também a caixa de seleção ficam disponíveis para uso. O botão *Desmontar* permanece bloqueado, pois o cartão já foi desmontado. A tabela de *Status* apresenta os dados para um cartão não montado, como pode ser visualizado na imagem abaixo:

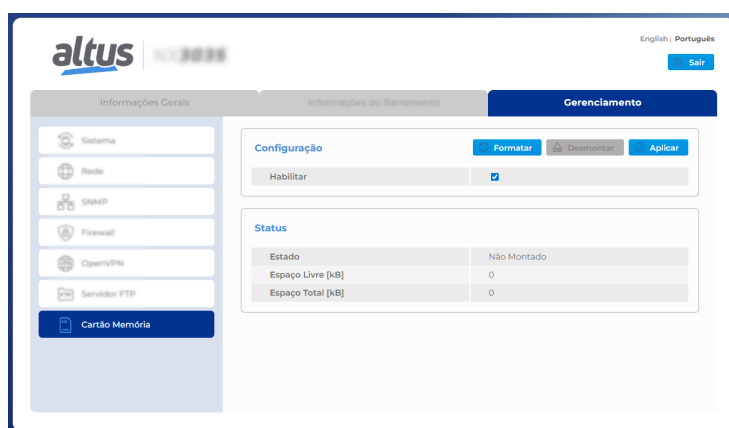


Figura 205: Cartão de Memória Não Montado

4. Habilitar

Foi desenvolvida uma configuração na página web do cartão para habilitar e desabilitar a interface do cartão de memória, esta funcionalidade faz parte dos requisitos de segurança cibernética de nível um, de acordo com a IEC 62443. Para habilitar, marque a caixa de seleção *Habilitar* na tabela de *Configuração*. Então utilize o botão *Aplicar* para enviar a nova configuração. Para desabilitar, desmarque a caixa de seleção *Habilitar* e utilize o mesmo botão para aplicar a configuração. Ao clicar no botão *Aplicar* será exibida uma mensagem, do estilo *pop-up*, solicitando a confirmação da operação. A imagem abaixo mostra a mensagem.

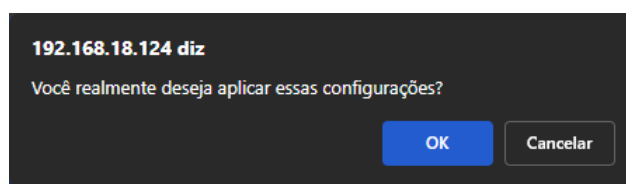


Figura 206: Mensagem Confirmação para Aplicar Configuração

Após confirmar no botão *OK*, a nova configuração é enviada para a UCP. Caso a interface tenha sido habilitada, as informações exibidas na página web dependem se há ou não um cartão de memória inserido para ser montado. Se não houver um dispositivo, a página apresentará as informações mostradas na figura [Cartão de Memória Página Inicial](#). Quando há um dispositivo conectado as informações exibidas serão as apresentadas na figura [Cartão de Memória com Dispositivo Montado](#), após o cartão ser devidamente montado.

Se a nova configuração desabilitar a interface, as informações apresentadas serão as da imagem abaixo.

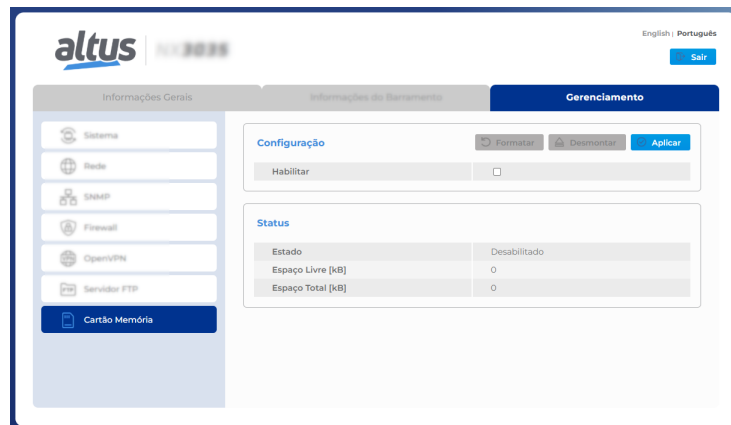


Figura 207: Cartão de Memória Interface Desabilitada

ATENÇÃO

No caso em que a desativação do cartão de memória estiver efetivada, a pasta MemoryCard não será montada.

5. Gerenciamento da Interface do Cartão de Memória via Aplicação

Para facilitar o gerenciamento da interface do cartão de memória, foi desenvolvida uma função que pode ser chamada diretamente pelo código de aplicação do usuário. A função *SetMemCardState* foi implementada dentro da biblioteca *NextoStandard*. A imagem abaixo mostra a informações da biblioteca, apresentadas no *Library Manager*.

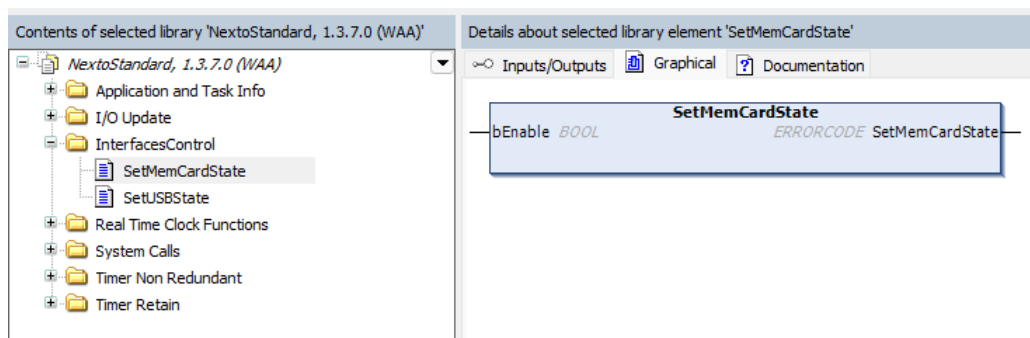


Figura 208: Informações de SetMemCardState no Library Manager

A função possui uma variável de entrada do tipo *bool*, *bEnable*, que recebe o valor para habilitar ou desabilitar a interface do cartão. A função possui três valores de retorno *NoError* em caso de sucesso, *SetMemCardStateFail* em caso de falha ou *ImportFunctionNotFound* caso a função não seja suportada. Segue abaixo um exemplo básico de declaração de variáveis e chamada da função.

```
PROGRAM UserPrg
VAR
  bSetMemoryCardInterfaceState : BOOL;
  stErrorCode : NextoStandard.ERRORCODE;
END_VAR

-----
// Exemplo de chamada de função para configurar interface do cartão de memória
stErrorCode:= NextoStandard.SetMemCardState(bEnable :=
  bSetMemoryCardInterfaceState);
```

ATENÇÃO

A função executa o comando para definir o valor desejado para a interface do cartão de memória. Não é necessário, nem recomendado, que a função seja chamada ciclicamente.

5.15. Docker

A funcionalidade Docker foi desenvolvida para expandir e aprofundar a integração e a operação da UCP. A principal função do Docker é permitir a execução e a manutenção de *Containers* Docker. A instalação, execução e manutenção do *Container* é realizada através da plataforma de gerenciamento *Portainer*.

ATENÇÃO

Existe uma ampla oferta de aplicações disponibilizadas sob a forma de Containers e distribuídas abertamente em plataformas como a *Docker Hub*. A Altus não presta suporte técnico específico relacionado à utilização e operação destas aplicações, devendo ser realizado diretamente com o desenvolvedor ou mantenedor do Container. Além disso, devido às restrições de processamento e memória que existem na UCP do controlador programável, muitos destes Containers podem não executar adequadamente. Por isso, recomendamos consultar os tutoriais disponíveis no site da Altus, onde são utilizados containers previamente testados e compatíveis com os recursos disponíveis nesta plataforma.

5.15.1. Como acessar um Container Docker

A grande maioria dos *Container* Docker são acessíveis através do endereçamento da porta da qual o *Container* está vinculado. A porta padrão para cada *Container* pode ser descoberta através de pesquisas, ou na documentação do próprio *Container*. A tendência é que o *Container* esteja sempre vinculado a porta padrão presente nessas fontes.

5.15.1.1. Como Acessar o Portainer

A nossa plataforma de gerenciamento de containers, *Portainer*, é também um *Container* Docker. Assim como grande parte dos *Containers* a serem utilizados, ela é acessível através do endereço da porta da qual ela está vinculada. No caso do *Portainer*, esta porta está localizada no endereço de IP da sua UCP seguido de “:9000”. Por exemplo: “192.168.15.1:9000”. Após o acesso ao link, será requisitado a criação de uma conta para o acesso a plataforma *Portainer*. Essa conta deve possuir um nome de usuário e uma senha de, ao menos, 12 caracteres.

CUIDADO

Essa senha não é de domínio *Altus*, e **NÃO** é recuperável!! A senha deve ser armazenada com cuidado!

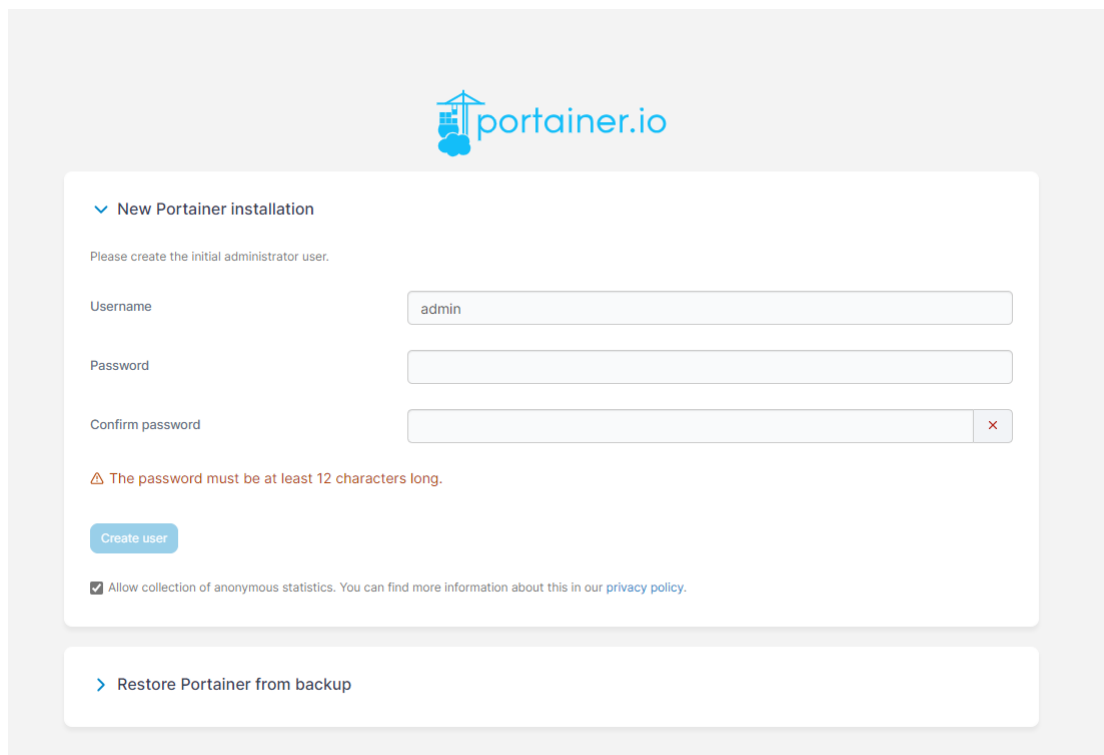


Figura 209: Tela de Login do Portainer

Também é possível utilizar a opção de “*Restore Portainer from backup*” para restaurar todas as configurações através da utilização de um arquivo de restauração.

ATENÇÃO

Recomenda-se a atualização periódica do arquivo de backup.

5.15.2. Quick Setup

Uma vez conectado, a plataforma irá redirecioná-lo para a tela inicial. Esta tela inicial irá conter duas opções, proceder usando o ambiente local do qual o *Portainer* está executando, ou adicionar um ambiente para se conectar a outro ambiente disponível.

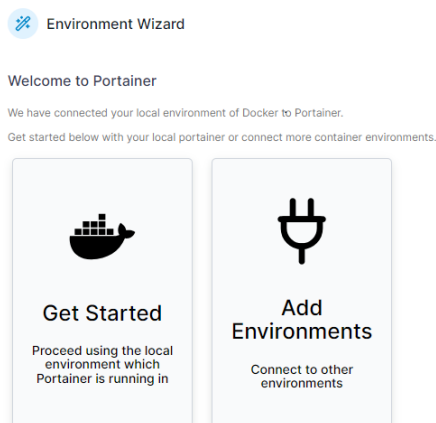


Figura 210: Tela inicial do Portainer

A opção a ser escolhida é a de "Get started". Assim que escolhida, o *Portainer* irá redirecioná-lo a tela que demonstra os ambientes disponíveis em seu dispositivo. Atualmente, haverá um único ambiente disponível que é o ambiente Docker, facilmente identificável pelo logotipo em forma de baleia que foi adotado pela tecnologia.

5.15.2.1. Informações Do Ambiente Local

Ainda na tela inicial, também chamada de "Home", é possível identificar algumas informações referentes ao ambiente Docker sendo executado dentro da UCP. Dentre essas informações temos: Data completa, versão do Docker server; *Stacks*, número de *Containers* seguido pelos seus status. A quantidade de *Volumes* criado, Imagens presentes na UCP, número de UCPs utilizáveis e quantidade máxima de memória disponível para todos os *Containers*.



Figura 211: Tela das Informações do Ambiente

ATENÇÃO

É importante que a data presente no ambiente esteja correta! Ela pode ser configurada através da função de clock setting presente na página *web* da UCP.

5.15.3. Dashboard de Ambiente

Após acessar a *Dashboard* clicando sob o ambiente que se deseja utilizar, você irá se deparar com algumas das informações que estavam presentes na tela anterior dentro de uma tabela de informações básicas nomeada de "Environment info", dentre elas temos as informações de hardware disponíveis e a versão do Docker. É possível acessar os menus listados abaixo interagindo com eles. De todos os menu listados abaixo, os de maior relevância são os menus de *Containers* e o de *Images*.

ATENÇÃO

Containers que dependem de um processador gráfico dedicado não são suportados pela UCP.

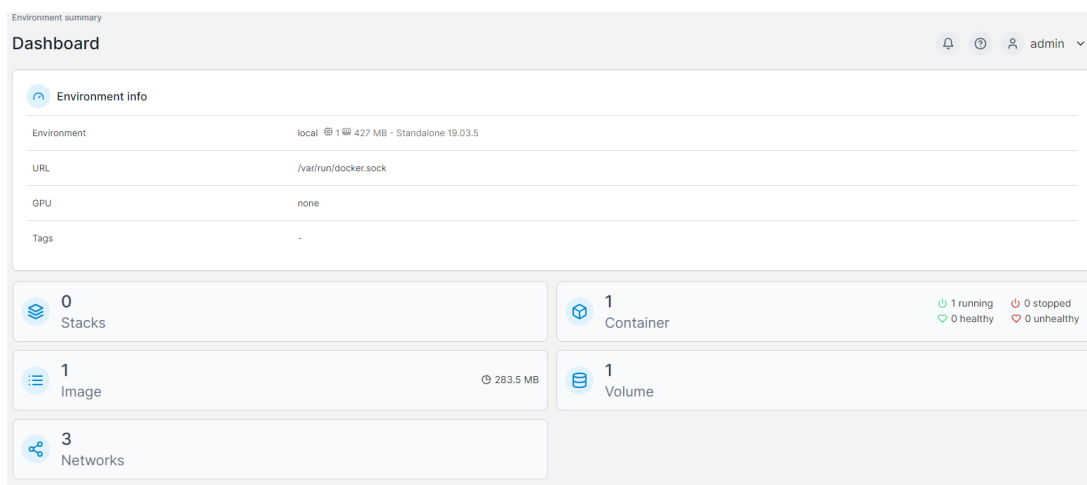


Figura 212: Dashboard

ATENÇÃO

Para mais informações sobre os menus e/ou sobre o próprio Docker, é possível acessar a documentação do *Portainer*, que está disponível ao clicar no símbolo de "(?)" ao lado do usuário no canto superior direito da tela.

5.15.3.1. Menu De Containers

O menu de *Containers* é responsável pela administração completa dos *Containers*. É possível acessar o *Container*, a Imagem, e algumas ações rápidas relevantes através desse menu. É possível ver-se data de criação, o endereço de IP e portas publicadas, quando aplicável, e o perfil proprietário do *Container*.

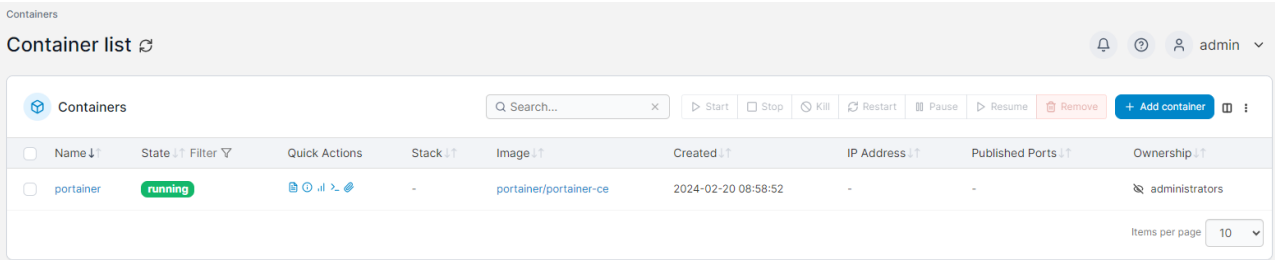


Figura 213: Menu de Containers

CUIDADO

Não deve ser realizado nenhum comando de *Stop*, *Kill*, *Restart* ou *Pause* no *Container* do *Portainer*. Se alguma dessas operações for realizada, é necessário um reboot para o *Container* voltar a funcionar.

5.15.3.2. Menu De Imagens

O menu de *Images* é responsável pela administração das Imagens baixadas, e os status delas. O nome e tag das Imagens pode ser obtida através do botão “*Search*” localizado a direita da caixa de inserção da Imagem. O botão de “*Pull the Image*” é responsável por obter a Imagem. Como é possível ver na figura abaixo, há uma limitação de “*pulls*” de Imagens.

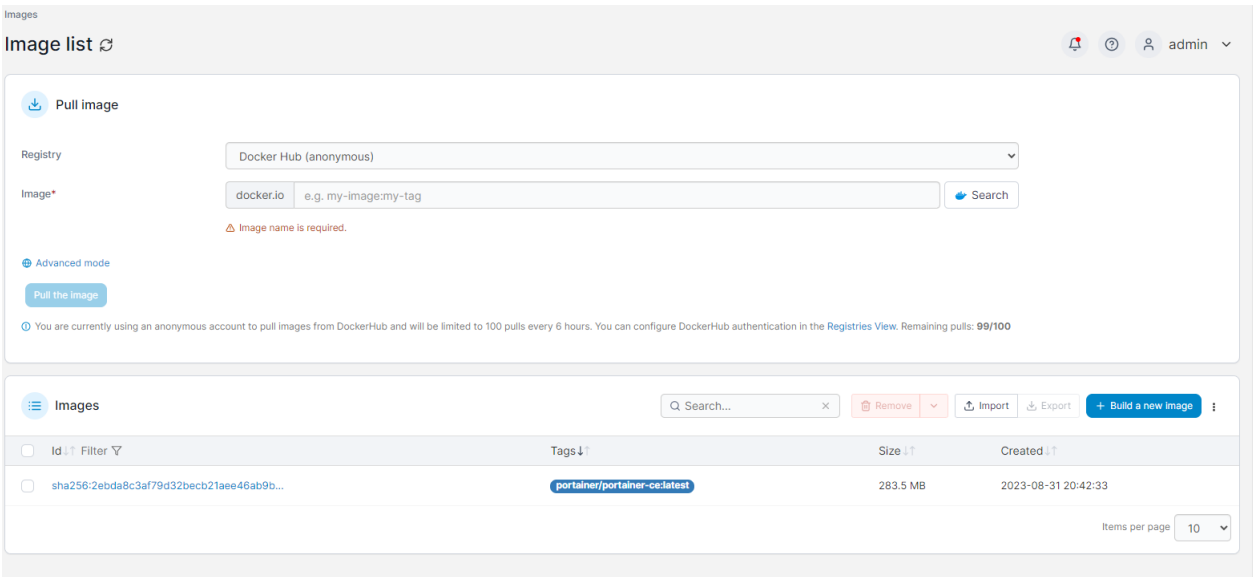


Figura 214: Menu de Imagens

É possível identificar todas as *Images* obtidas, se o status dela é de “Unused”, a sua tag, o tamanho e a data de criação. Assim como na *Stack*, aqui é possível de se adicionar Imagens usando um editor web através dos botões de “Import” e de “Build a new Image”.

5.15.4. Criando e configurando um Container

Os *Containers* podem ser criados através do menu de *Containers*, acessível pela dashboard, ou pelo menu de acesso rápido presente na esquerda da página web do *Portainer*. Há a também a possibilidade de criar *Containers* através de *Stacks* e configurá-los posteriormente.

ATENÇÃO

Durante o processo de instalação e configuração dos *Containers*, é altamente recomendado que o controlador esteja parado, com a aplicação removida através do comando *Reset Origin*. Isto é necessário pois, devido a características intrínsecas da tecnologia Docker, estas operações realizam acessos de baixo nível no sistema operacional do controlador que podem afetar temporariamente a estabilidade da função de controle (degradação temporária de performance e perda de determinismo) e, em casos extremos, até mesmo causar a reinicialização do sistema.

Através do botão de *Add Container*, é possível adicionar *Containers* à partir de *Images* obtidas através do menu *Images* ou *Imagens* que serão obtidas no processo de criação do *Container*.

Ao clicar em *Add Container*, você irá se deparar com a tela *Create container*.

The screenshot shows the 'Create container' page in Portainer. The interface is clean and modern, with a light gray background. The form is organized into sections: 'Name', 'Image configuration', 'Webhooks', 'Network ports configuration', and 'Access control'. The 'Name' field has a placeholder 'e.g. myContainer'. The 'Image configuration' section includes a 'Registry' dropdown set to 'Docker Hub (anonymous)' and an 'Image*' field with 'docker.io' and 'e.g. my-image:my-tag'. A red warning icon indicates 'Image name is required'. The 'Webhooks' section has a 'Create a container webhook' toggle and a 'Business Edition Feature' button. The 'Network ports configuration' section has a 'Publish all exposed network ports to random host ports' toggle and a 'Manual network port publishing' button. The 'Access control' section has an 'Enable access control' toggle. At the bottom, there are two radio button options: 'Administrators' (selected) and 'Restricted'.

Figura 215: Tela de Criação do Container

Nesta tela é possível nomear o *Container* sendo utilizado. e escolher a *Imagem* da qual quer que ele utilize.

Assim como através do menu de *Images*, é possível acessar o Docker hub para ver as “*Images*” disponíveis e seus respectivos “*names*”. Há uma opção de “*Always pull the Image*”, caso ativado não deverá se fazer necessário adicionar a *Image* através do menu de “*Images*” previamente. Porém, isto nem sempre é o caso. O *Container* pode acabar não sendo criado em caso de alguma falha devido a ausência da imagem no dispositivo. Para solucionar isso, basta adicionar uma image. Você pode usar a [Menu De Imagens](#) como um guia. As opções contidas nesta caixa de configurações básicas do *Container* são sensíveis caso a caso e devem ser avaliadas de acordo a necessidade.

Para criar o *Container* em definitivo, basta utilizar o botão de “*deploy the container*”, entretanto, é de suma importância realizar as demais configurações presentes no item abaixo. No caso de criações de *Containers* através do método de *Stacks*, é importante ressaltar que:

CUIDADO

Quando utilizado a opção de *stacks* (docker-compose) ao passar os parâmetros para criação de um volume, deve ser utilizado o seguinte caminho: `"/codesys/NextoUser/Files/docker/var-lib/"`. Caso contrário, o startup do seu *container* será abortado.

5.15.4.1. Configurando Um Container

5.15.4.1.1. Menu Command & Logging

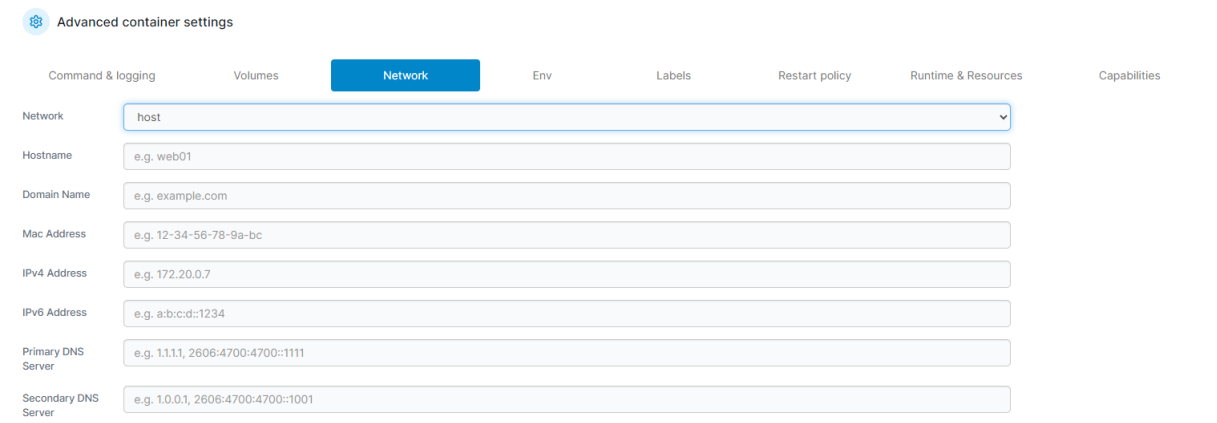
O primeiro menu disponível nas Configurações Avançadas do Container é o de *Command & logging*. Neste primeiro menu não é necessário a alteração de nenhum parâmetro, porém é recomendável a ativação da configuração de console para *Container* que possam tirar proveito disso.

Por padrão, o log driver é configurado como json-file e é configurado para imprimir mensagens em um único arquivo de tamanho máximo de 10MB. Isso pode ser reconfigurado no menu *Command & Logging*.

Figura 216: Menu de Configuração de Command & logging

5.15.4.1.2. Menu Network

O menu de *Network* permite a configuração da rede na qual este *Container* irá atuar. A utilização da *network* no modo “*host*” é mandatória para a operação do *Container*. Isso ocorre pois o nosso Docker **não** suporta nenhuma outra forma de configuração de *network*.



Advanced container settings

Command & logging Volumes **Network** Env Labels Restart policy Runtime & Resources Capabilities

Network: host

Hostname: e.g. web01

Domain Name: e.g. example.com

Mac Address: e.g. 12-34-56-78-9a-bc

IPv4 Address: e.g. 172.20.0.7

IPv6 Address: e.g. abc:d::1234

Primary DNS Server: e.g. 1.1.1.1, 2606:4700:4700::1111

Secondary DNS Server: e.g. 1.0.0.1, 2606:4700:4700::1001

Figura 217: Menu de Configuração de Network

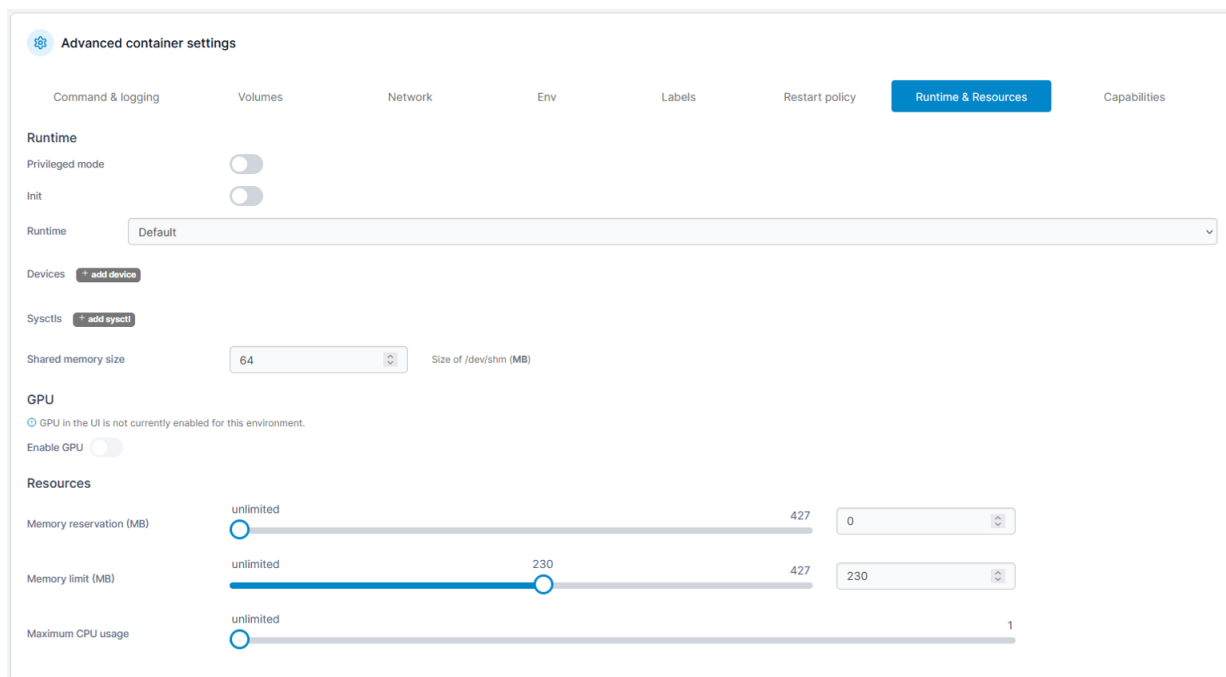
5.15.4.1.3. Menu Runtime & Resources

O menu “*Runtime & Resources*” é o mais importante dos menus presentes em um *Container*. Neste menu o usuário pode configurar o máximo de CPU e memória RAM que um *Container* pode consumir.

CUIDADO

É recomendado que a soma do limite de memória RAM de todos os *Containers* não ultrapasse 230 *MegaBytes* para garantir uma reserva de memória suficiente para a operação do PLC.

A figura abaixo exemplifica as configurações recomendadas em caso de uso de apenas um *Container*:



Advanced container settings

Command & logging Volumes Network Env Labels Restart policy **Runtime & Resources** Capabilities

Runtime

Privileged mode: ☐

Init: ☐

Runtime: Default

Devices: [+ add device](#)

Sysctls: [+ add sysctl](#)

Shared memory size: 64 Size of /dev/shm (MB)

GPU

☐ GPU in the UI is not currently enabled for this environment.

Enable GPU: ☐

Resources

Memory reservation (MB): unlimited (slider) 427 0

Memory limit (MB): unlimited (slider) 230 427 230

Maximum CPU usage: unlimited (slider) 1

Figura 218: Menu de Configuração de Runtime & Resources

5.15.4.1.4. Menu Capabilities

Por último há o menu de *capabilities*, o qual disponibiliza uma grande quantidade de capacidades do *Container*. Cada uma delas com sua própria explicação de funcionamento, acessíveis através do pequeno símbolo de "(?)" ao lado de cada configuração.

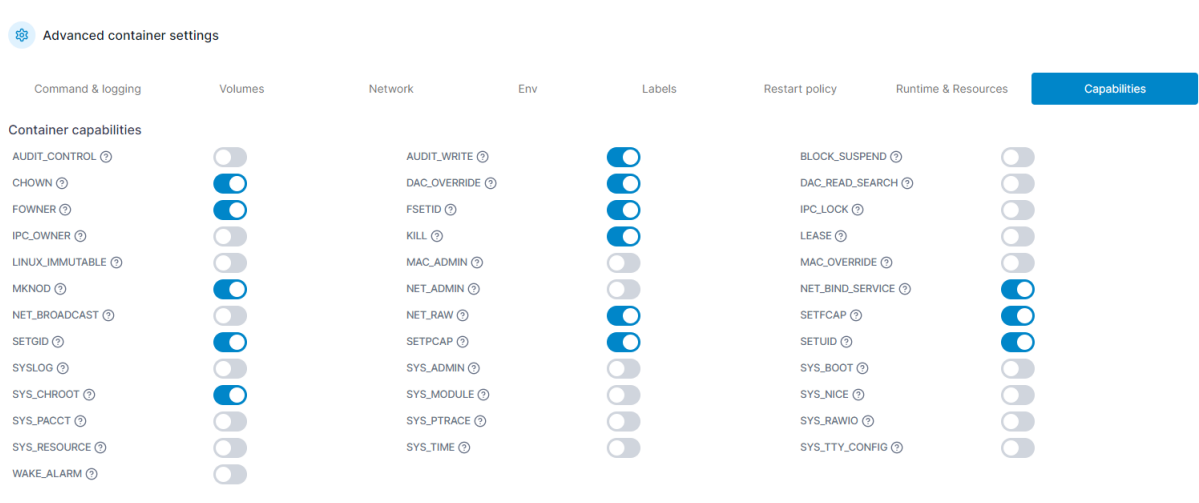


Figura 219: Menu de Configuração de Capabilities

CUIDADO

Não garantimos o funcionamento desejado do sistema no caso de ativação de certos parâmetros. Cuidado deve ser tomado no caso de ativação, ou desativação, de capacidades sensíveis.

5.15.4.2. Utilizando Um Container

Como mencionado na seção [Como Acessar o Portainer](#), a maior parte dos *Containers* podem ser acessados através da utilização do endereço "*IP.DA.SUA.UCP:PORTA DO CONTAINER*", seguido por um endereçamento de porta. Uma forma de acessar um *Container* é através do próprio *Portainer* e nesse caso, o endereço da porta é ":9000", ou seja, para acessá-lo seria utilizado "*IP.DA.SUA.UCP:9000*".

Assim que criado o *Container*, ao acessá-lo, duas tabelas serão visíveis, uma tabela de ações e um menu de status.

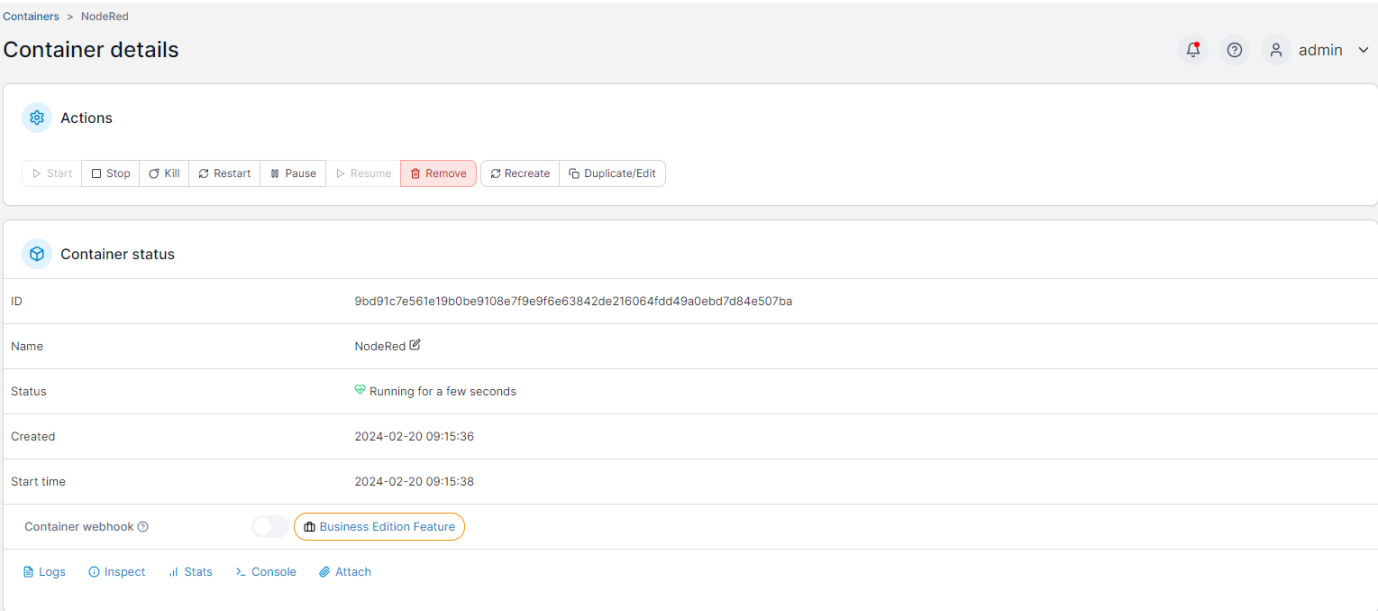


Figura 220: Tabela de Detalhes do Container

Essa tabela de ações possibilita a execução de algumas ações gerais a serem realizadas com o *Container*. A tabela, chamada de “*Container status*”, apresenta algumas informações gerais do *Container*. Ao final desta tabela há algumas funcionalidades que podem ser acessadas, a primeira delas (*Logs*) é a página de logs do *Container* sendo executado, a segunda (*Inspect*) é uma inspeção profunda de parâmetros e configurações do *Container*, a terceira (*Stats*) são estatísticas relevantes do consumo de recursos gerados durante o funcionamento do *Container*. A quarta (*Console*) é um console para a interação com o *Container* em casos pertinentes.

ATENÇÃO

Deve-se saber qual tipo de console é suportado pelo *Container* a ser utilizado, caso a opção selecionada não seja suportada, não será possível interagir com ele.

A próxima tabela é a tabela de controle de acesso, onde o proprietário do *Container* pode ser selecionado.

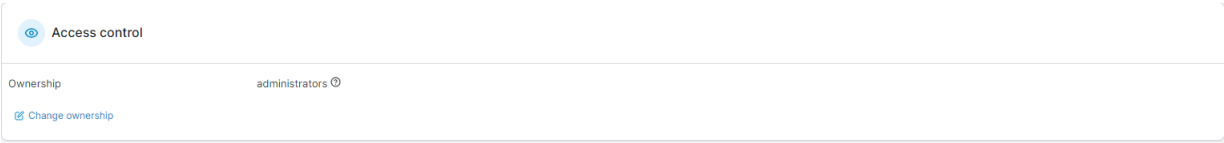


Figura 221: Access Control box

Seguindo há a tabela de "saúde"do *Container*, onde ele irá informar o seu estado, contador de falhas e o último código de erro que fora registrado.

A tabela “*Create Image*” possibilita o usuário a criar uma Imagem daquele Docker.

Create image

You can create an image from this container, this allows you to backup important data or save helpful configurations. You'll be able to spin up another container based on this image afterward.

Registry: Docker Hub (anonymous)

Image*: docker.io e.g. my-image:my-tag Search

Image name is required.

Advanced mode

You are currently using an anonymous account to pull images from DockerHub and will be limited to 100 pulls every 6 hours. You can configure DockerHub authentication in the [Registries View](#). Remaining pulls: 99/100

Note: If you don't specify the tag in the image name, `latest` will be used.

Create

Figura 222: Tabela de Create Image

5.15.4.3. Como criar uma imagem a partir de um container existente salvando os seus dados

Na tela *Container Details*, acessada a partir do menu *Containers* e clicando sobre o nome do *Container*, é possível criar uma imagem deste *Container* contendo todos os dados e configurações já realizadas. Este recurso pode ser utilizado para facilitar a distribuição de exemplos, para realizar backup ou também para migrar a aplicação Docker para um outro controlador.

Figura 223: Criando uma imagem de um Container existente

5.15.4.4. Monitorando Um Container

Para monitorar um *Container*, pode ser utilizado uma ferramenta presente dentro do próprio *Portainer*. Essa ferramenta é acessível tanto através da lista de *Containers*, em um ícone nos "*Quick Actions*" chamado "*Stats*" quanto através da tela de "*Container Statistics*". Assim que acessado, o usuário verá o menu representado na figura abaixo. Será possível ver estatísticas de consumo de memória, UCP e E/S. A taxa de atualização das estatísticas de consumo de memória, UCP e E/S pode ser configurada através do campo "*Refresh rate*". Essa ferramenta monitora exclusivamente um *Container*, e não é possível monitorar a soma de todos os *Containers* através do *Portainer*. Porém, existem diagnósticos da UCP que podem ser monitorados através do *Mastertool* e estão diretamente relacionados a utilização do *Container*, sendo eles o diagnóstico *byProcessorLoad*, onde é possível visualizar a porcentagem de uso do processador da UCP e os diagnósticos *dwFreeSpacekB* e *dwTotalSizekB* que apresentam o valor total e disponível de memória de usuário.

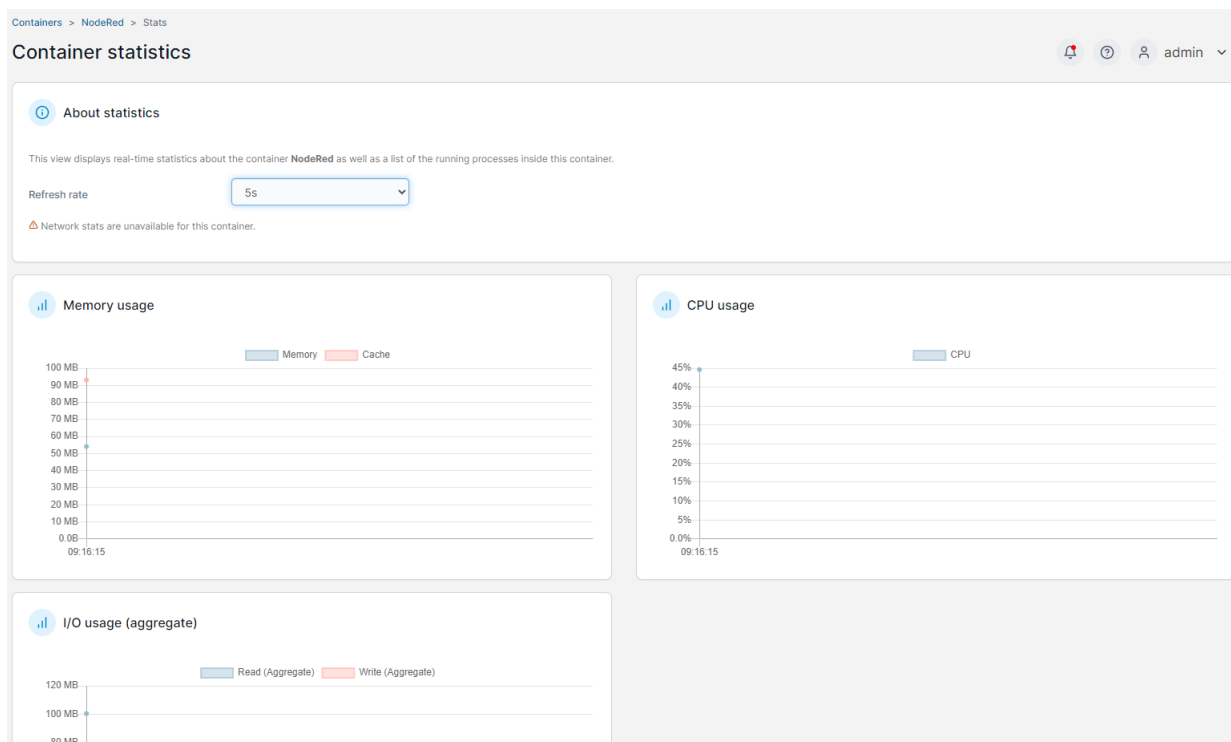


Figura 224: Tela de Monitoramento

5.15.4.5. Características Técnicas

5.15.4.5.1. Memória Flash

Os *Containers* Docker estão alocados na pasta de armazenamento do usuário, acessível através do menu *Files* do *Master-tool*. Dessa forma, o Docker compartilha o mesmo espaço destinado à memória do usuário, de modo que a capacidade máxima disponível para os *Containers* e para a Aplicação corresponde ao limite total desse armazenamento. *Containers* que utilizam uma grande quantia de armazenamento podem, por sua vez, demorar para realizarem o processo de obtenção da Imagem, e de deploy.

5.15.4.5.2. Memória RAM

O uso máximo de memória RAM, somando-se todos os *Containers* Docker, **DEVE** ser inferior a 230 *MegaBytes*.

Caso haja tentativa de estabelecer um valor superior ao disponível o *Portainer* irá apresentar uma mensagem de erro. Pode resultar também no mal funcionamento do dispositivo. A opção de uso ilimitado de recursos **NÃO** deve ser utilizada, podendo resultar em mal funcionamento do dispositivo.

PERIGO

A memória é compartilhada entre **TODOS** os *Containers* Docker e a aplicação devendo-se ser cuidadosamente mensurada para não gerar mal funcionamento no dispositivo. Recomenda-se que a soma de uso de memória RAM de todos os *Containers* Docker seja limitada dentro do valor de 230 *MegaBytes* para o funcionamento do dispositivo.

5.15.4.5.3. UCP

A limitação de UCP utilizada pelo Docker também deve ser configurada através do menu descrito na seção [Menu Runtime & Resources](#). O valor configurado se refere a UCP por completo, ou seja, no caso de haver uma aplicação e/ou *Containers* que estejam consumindo um poder de processamento elevado, este valor deve ser modificado de acordo. A UCP não possui um processador gráfico dedicado, integrado ou drivers gráficos, podendo apresentar instabilidades em *Containers* que necessitem desse recurso.

5.15.4.6. Como acessar a pasta do Docker

5.15.4.6.1. MasterTool

Através do *Mastertool*, há a possibilidade de conferir os arquivos dos *Containers* Docker utilizando funcionalidade de acesso aos arquivos da UCP, sob o menu de dispositivo. A imagem a seguir demonstra o menu e a pasta responsável por permitir o acesso desses arquivos.

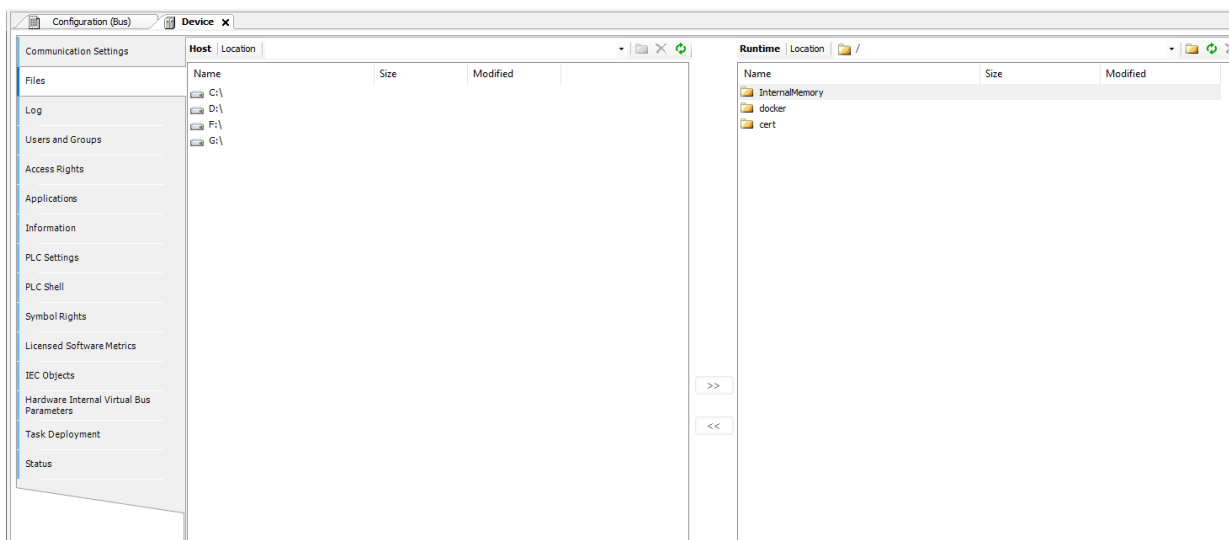


Figura 225: Pasta do Docker no Mastertool

5.15.4.7. Usando Memória Externa como Volume Docker

Para expandir a capacidade de armazenamento do Docker com dispositivos de memória externa, como cartões SD e dispositivos de armazenamento em massa USB, você deve usar volumes para montar os sistemas de arquivos externos nos contêineres. Este processo é suportado apenas através de Portainer Stacks (Docker Compose).

Use os seguintes caminhos do host para configurar volumes no seu arquivo de stack:

Dispositivo Externo	Caminho no Host
Cartão SD	/codesys/NextoUser/Files/MemoryCard
Armazenamento em Massa USB	/codesys/NextoUser/Files/Mass_Storage

Tabela 198: Caminhos do host para dispositivos de memória externa

Para criar uma Stack no Portainer que usa memória externa, você deve configurar volumes para montar os caminhos físicos dos dispositivos em seu contêiner. Isso permite que serviços em contêineres acessem arquivos no armazenamento externo.

Para implantar uma nova stack, navegue até a seção *Stacks* no Portainer e clique em *Add Stack*. A [Figura 226](#) mostra um exemplo de um arquivo de stack que monta tanto um cartão SD quanto um dispositivo de armazenamento em massa USB em um serviço:

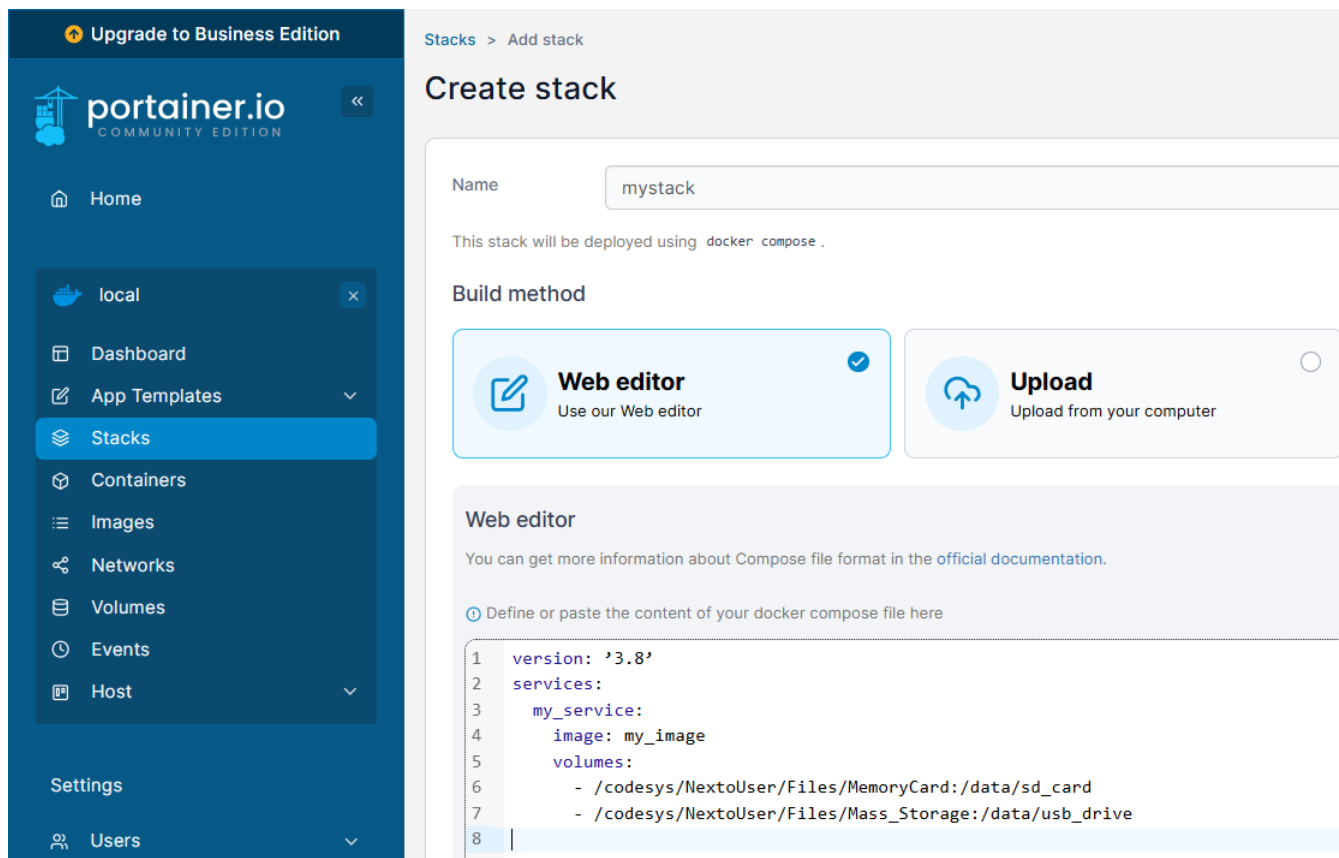


Figura 226: Exemplo de Arquivo de Stack com Memória Externa

ATENÇÃO

Se o dispositivo externo não estiver montado durante a inicialização do contêiner, ele não terá acesso aos arquivos no dispositivo. Certifique-se de que os dispositivos de memória externa estão corretamente montados antes de iniciar os contêineres. Da mesma forma, se o dispositivo externo for removido enquanto o contêiner estiver em execução, o contêiner perderá o acesso aos arquivos e precisará ser reiniciado para recuperar o acesso.

5.15.4.8. Solução de problemas

Esta seção tem como objetivo descrever os problemas mais comuns, as suas causas e soluções.

5.15.4.8.1. Falha ao instalar Container devido a Certificado expirado ou inválido

Causa: horário do PLC é inválido.

Solução: ajustar o horário do PLC.

5.15.4.8.2. Portainer inacessível, retornando ao reinicializar a UCP

Causa: não foram aplicados os limites de memória para os Containers conforme seção [Menu Runtime & Resources](#)

Solução: aplicar os limites de memória conforme descrito neste manual.

6. 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.

6.1. Diagnósticos do Módulo

6.1.1. 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:

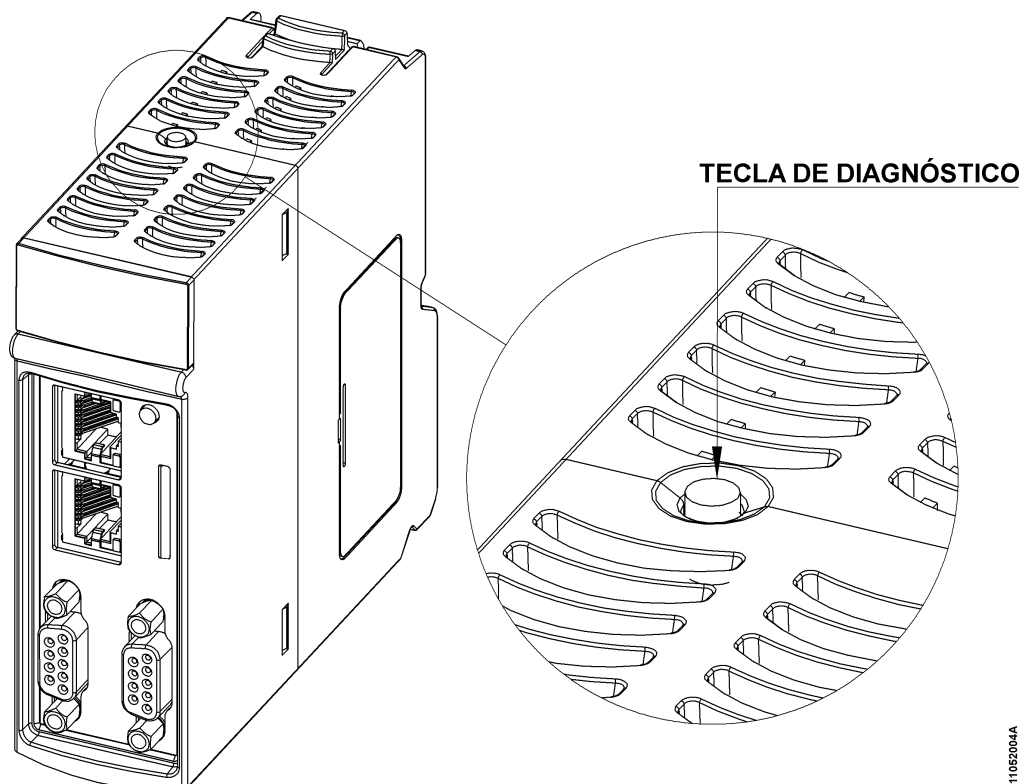


Figura 227: Tecla de Diagnóstico

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, 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.

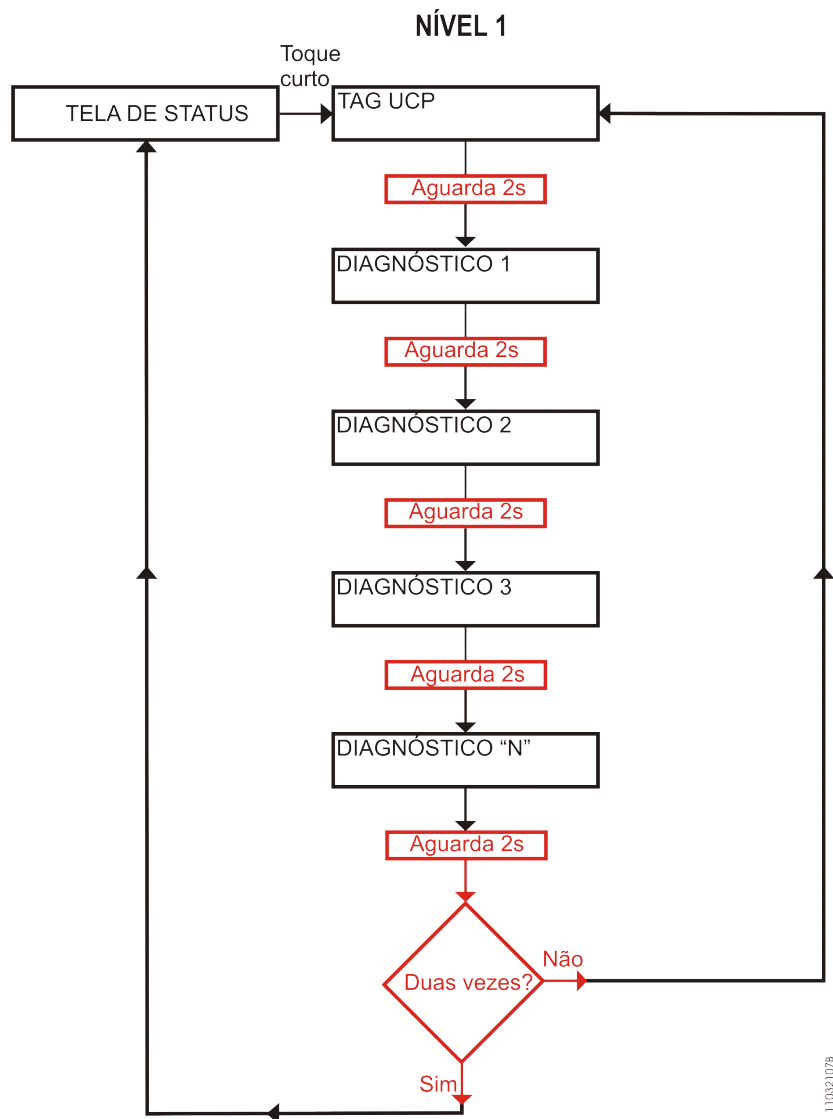


Figura 228: Visualização dos Diagnósticos da UCP

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 204

Tabela 199: 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 204.

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.

6.1.2. 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 200 e Tabela 201 mostram o significado de cada estado e suas respectivas descrições.

6.1.2.1. 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	2
Desligado	Piscando 4x	Erro de configuração ou de hardware no barramento.	O barramento está danificado ou não está configurado corretamente.	0 (Alta)

Tabela 200: Descrição dos Estados do LED de Diagnóstico

6.1.2.2. 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	UCP danificada e/ou sistema operacional corrompido	1 (Alta)

Tabela 201: 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 208.

Cão-de-guarda de hardware: Para limpar qualquer indicação de cão-de-guarda, como no LED WD ou no operando `Reset.bWatchdogReset`, a UCP deve ser desconectada da fonte de alimentação e é preciso aguardar até o LED de indicação de cão-de-guarda (LED WD) desligar.

Para verificar as condições da aplicação na reinicialização da UCP, ver configurações na Tabela 39.

6.1.2.3. 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 1000 Mbits/s
●	○	LINK de rede de 100 Mbits/s
●	●	LINK de rede de 10 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 202: Significado dos LEDs Ethernet

6.1.3. 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 para desktop e digitar, na barra de endereço, o endereço IP correspondente à UCP (Ex.: <http://192.168.1.1>). Para maiores detalhes, consultar a seção [Página Web de Sistema](#).

6.1.4. 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.

A tabela abaixo faz um resumo da divisão dos bytes/words de diagnósticos:

Byte	Descrição
0 a 3	Diagnósticos resumidos da UCP.
4 a 1686	Diagnósticos detalhados da UCP.

Tabela 203: Divisão dos Diagnósticos da UCP

6.1.4.1. Diagnósticos Resumidos

A tabela abaixo mostra o significado de cada bit dos diagnósticos resumidos da UCP:

Mensagem de Diagnóstico	Variável DG_Modulo.tSummarized.*	Descrição
SEM DIAG	-	Não existe diagnóstico ativo.
CONFIG. INCOMPATIVEL	bConfigMismatch	TRUE – Existe algum problema de configuração no barramento, como módulo em posição incorreta.
		FALSE – O barramento está configurado corretamente.
MODULOS AUSENTES	bAbsentModules	TRUE – Um ou mais módulos declarados estão ausentes.
		FALSE – Todos os módulos estão presentes no barramento.
MODULOS TROCADOS	bSwappedModules	TRUE – Dois módulos estão trocados entre si no barramento.
		FALSE – Não há módulos trocados no barramento.
MODULOS NAO DECLARADOS	bNonDeclaredModules	TRUE – Um ou mais módulos presentes no barramento não estão declarados.
		FALSE – Todos os módulos presentes no barramento estão declarados.
MODULOS C/ DIAGNOSTICO	bModulesWithDiagnostic	TRUE – Um ou mais módulos do barramento estão com diagnóstico ativo.
		FALSE – Não existem diagnósticos ativos nos módulos do barramento.
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.
MODULOS C/ ERRO PARAM.	bModuleParameterError	TRUE – Um ou mais módulos do barramento estão com erro de parametrização.
		FALSE – Todos os módulos estão parametrizados.
ERRO NO BARRAMENTO	bWHSBBusError	TRUE – Indicação do mestre que existe falha no barramento WHSB.
		FALSE – O barramento WHSB está em correto funcionamento.
FALHA DE HARDWARE	bHardwareFailure	TRUE – Falha no hardware da UCP.
		FALSE – O hardware está em correto funcionamento.
EXCECAO NO SOFTWARE	bSoftwareException	TRUE – Uma ou mais exceções geradas pelo software.
		FALSE – Não foram geradas exceções no software.
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.
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.
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.
ERRO CONFIG. NET 2	bNET2ConfigError	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface Ethernet NET 2.

Mensagem de Diagnóstico	Variável DG_Modulo.tSummarized.*	Descrição
		FALSE – A configuração da interface Ethernet NET 2 está correta.
ERRO CONFIG. NET 3	bNET3ConfigError	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface Ethernet NET 3.
		FALSE – A configuração da interface Ethernet NET 1 está correta.
DATA/HORA INVALIDA	bInvalidDateTime	TRUE – A data ou a hora são inválidas.
		FALSE – A data e a hora estão corretas.
RUNTIME RESET	bRuntimeReset	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.
PERDA DE RETENTIVIDADE	bRetentivityLost	TRUE – Ocorreu um erro ao salvar os dados retentivos.
		FALSE – Dados válidos na memória retentiva durante a inicialização.
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.
UCP POSICAO INCORRETA	bWrongCPUSlot	TRUE - A UCP está conectada em um slot diferente do declarado na aplicação de usuário.
		FALSE - A UCP está devidamente conectada no slot declarado na aplicação de usuário.
ERRO PORTA USB	bUSBDiagnostic	TRUE - A USB possui algum diagnóstico.
		FALSE - A USB está sem problemas.
BASTIDOR AUSENTE	bAbsentRacks	TRUE – Um ou mais bastidores declarados estão ausentes.
		FALSE – Todos os bastidores declarados estão presentes.
BASTIDOR DUPLICADO	bDuplicatedRacks	TRUE – Há algum bastidor com o número de identificação duplicado.
		FALSE – Não existem bastidores com o número de identificação duplicado.
BASTIDOR INVALIDO	bInvalidRacks	TRUE – Há algum bastidor com o número de identificação inválido.
		FALSE – Não existem bastidores com o número de identificação inválido.
BASTIDOR NAO DECLARADO	bNonDeclaredRacks	TRUE – Há algum bastidor com o número de identificação não declarado.
		FALSE – Não existem bastidores com o número de identificação não declarado.
SLOT DUPLICADO	bDuplicatedSlots	TRUE – Há algum endereço de slot duplicado.
		FALSE – Não existem endereços de slot duplicados.

Tabela 204: Diagnósticos Resumidos da UCP

Notas:

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 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 Página Web de Sistema da UCP.

6.1.4.2. 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. 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.
- *DG_Modulo.*, sendo a palavra *Modulo* substituída pelo produto utilizado.

6.1.4.2.1. Diagnósticos Detalhados Grupo Target

DG_Modulo.tDetailed.Target.*	Tamanho	Descrição
dwCPUModel	DWORD	Modelo da UCP.
abyCPUVersion	BYTE ARRAY(4)	Versão do firmware.
abyBootloaderVersion	BYTE ARRAY(4)	Versão do bootloader.
abyAuxprocVersion	BYTE ARRAY(4)	Versão do processador auxiliar.
abyPowprocVersion	BYTE ARRAY(4)	Versão do processador de alimentação.

Tabela 205: Diagnósticos Detalhados Grupo Target

6.1.4.2.2. Diagnósticos Detalhados Grupo Hardware

DG_Modulo.tDetailed.Hardware.*	Tamanho	Descrição
bRTCFailure	BIT	O processador principal não está habilitado para comunicar com o RTC (relógio da UCP).
bThermometerFailure	BIT	Falha na comunicação entre o termômetro e o processador principal.
bNoBusSupplyStatus	BIT	A fonte de alimentação integrada da UCP não está alimentando os módulos de E/S do barramento. Para isso deve ser utilizado um módulo de alimentação adicional, como o NX8000.

Tabela 206: Diagnósticos Detalhados Grupo Hardware

6.1.4.2.3. Diagnósticos Detalhados Grupo Exception

DG_Modulo.tDetailed.Exception.*	Tamanho	Descrição
wExceptionCode	WORD	Código de Exceção gerado pelo RTS, o significado dos códigos pode ser conferido na Tabela 208.
byProcessorLoad	BYTE	Nível, em percentagem (%), de carga no processador.

DG_Modulo.tDetailed.Exception.*	Tamanho	Descrição
---------------------------------	---------	-----------

Tabela 207: 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 208: Códigos de Exceção RTS

6.1.4.2.4. Diagnósticos Detalhados Grupo WebVisualization

DG_Modulo.tDetailed.WebVisualization.*	Tamanho	Descrição
byConnectedClients	BYTE	Quantidade de clientes conectados ao WebVisualization.

Tabela 209: Diagnósticos Detalhados Grupo WebVisualization

6.1.4.2.5. Diagnósticos Detalhados Grupo RetainInfo

DG_Modulo.tDetailed.RetainInfo.*	Tamanho	Descrição
byCPUInitStatus	BYTE	Status da Inicialização da UCP: 01: Hot start 02: Warm Start 03: Cold Start Obs.: Essas variáveis são reinicializadas em todas as energizações.
wCPUColdStartCounter	WORD	Incrementa caso a UCP perca retentividade durante a inicialização. (0 a 65535)
wCPUWarmStartCounter	WORD	Incrementa caso a UCP inicie com dados retentivos válidos. (0 a 65535)
wRTSResetCounter	WORD	Contador de resets efetuados pelo RTS (Runtime System). (0 a 65535).

Tabela 210: Diagnósticos Detalhados Grupo RetainInfo

6.1.4.2.6. Diagnósticos Detalhados Grupo Reset

DG_Modulo.tDetailed.Reset.*	Tamanho	Descrição
bBrownOutReset	BIT	A UCP foi reiniciada devido a uma falha na alimentação na última inicialização.
bWatchdogReset	BIT	A UCP foi reiniciada devido ao cão-de-guarda ativo na última inicialização.

Tabela 211: 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 reiniciada automaticamente e indicará o diagnóstico de reset por brownout.

6.1.4.2.7. Diagnósticos Detalhados Grupo Thermometer

DG_Modulo.tDetailed.Thermometer.*	Tamanho	Descrição
bOverTemperatureAlarm	BIT	Alarme gerado devido à temperatura interna estar em 85 °C ou acima de 85 °C.
bUnderTemperatureAlarm	BIT	Alarme gerado devido à temperatura interna estar em 0 °C ou abaixo de 0 °C.
diTemperature	DINT	Temperatura lida no sensor interno da UCP.

Tabela 212: 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.

6.1.4.2.8. Diagnósticos Detalhados Grupo Serial

DG_Modulo.tDetailed.Serial.COM1.*	Tamanho	Descrição
byProtocol	BYTE	Protocolo selecionado na COM 1: 00: Sem protocolo 01: MODBUS RTU Master 02: MODBUS RTU Slave 03: Outro protocolo

DG_Modulo.tDetailed.Serial.COM1.*	Tamanho	Descrição
dwRXBytes	DWORD	Contador de caracteres recebidos através da COM 1 (0 a 4294967295).
dwTXBytes	DWORD	Contador de caracteres transmitidos através da COM 1 (0 a 4294967295).
wRXPendingBytes	WORD	Número de caracteres pendentes no buffer de leitura na COM 1 (0 a 4096).
wTXPendingBytes	WORD	Número de caracteres pendentes no buffer de transmissão na COM 1 (0 a 1023).
wBreakErrorCounter	WORD	O transmissor está mantendo a linha de dados em zero por muito tempo, de acordo com o bit de dados configurado.
wParityErrorCounter	WORD	O frame recebido tem o bit de paridade incompatível.
wFrameErrorCounter	WORD	O frame recebido tem o ponto de partida errado, geralmente causado por uma incompatibilidade de ruído ou baud rate.
wRXOverrunCounter	WORD	Quando o buffer de anel recebido está cheio e começa a perder os frames antigos (muitos frames não tratados pelo dispositivo).

Tabela 213: 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.

6.1.4.2.9. Diagnósticos Detalhados Grupo CAN

DG_Modulo.tDetailed.CAN.*	Tamanho	Descrição
bBusAlarm	BIT	O barramento tem um erro crítico e está desligado.
byBusState	BYTE	Informa o estado do dispositivo.
udiTxCounter	UDINT	Número de pacotes (Tx) alterados no barramento CAN da UCP.
udiRxCounter	UDINT	Número de pacotes (Rx) alterados no barramento CAN da UCP.
udiTxErrorCounter	UDINT	Número de pacotes (Tx) com erros no barramento CAN da UCP.
udiRxErrorCounter	UDINT	Número de pacotes (Rx) com erros no barramento CAN da UCP.
udiLostCounter	UDINT	Número de pacotes perdidos no barramento CAN da UCP.

Tabela 214: Diagnósticos Detalhados Grupo CAN

6.1.4.2.10. Diagnósticos Detalhados Grupo USB

DG_Modulo.tDetailed.USB.*	Tamanho	Descrição
byUSBDevice	BYTE	Tipo do dispositivo conectado à porta USB.
bOvercurrent	BIT	O dispositivo conectado na porta USB está drenando mais corrente do que o suportado.
bEnable	BIT	A porta USB está habilitada.
tMassStorage.byMountState	BYTE	Informa o estado do dispositivo.
tMassStorage.dwFreeSpaceKb	DWORD	Informa o espaço livre no dispositivo de armazenamento em massa.
tMassStorage.dwTotalSizeKb	DWORD	Informa o tamanho total do dispositivo de armazenamento em massa.
tSerialConverter.dwRXBytes	DWORD	Contador de caracteres recebidos na COM 10 (0 a 4294967295).

DG_Modulo.tDetailed.USB.*	Tamanho	Descrição
tSerialConverter.dwTXBytes	DWORD	Contador de caracteres transmitidos pela COM 10 (0 a 4294967295).
tSerialConverter.wRXPendingBytes	WORD	Número de caracteres restantes no buffer de leitura na COM 10 (0 a 4095).
tSerialConverter.wTXPendingBytes	WORD	Número de caracteres restantes no buffer de transmissão na COM 10 (0 a 1023).
tSerialConverter.wBreakErrorCounter	WORD	O transmissor está mantendo a linha de dados em zero por muito tempo, de acordo com o bit de dados configurado.
tSerialConverter.wParityErrorCounter	WORD	O frame recebido tem o bit de paridade incompatível.
tSerialConverter.wFrameErrorCounter	WORD	O frame recebido tem o ponto de partida errado, geralmente causado por uma incompatibilidade de ruído ou baud rate.
tSerialConverter.wRXOverrunCounter	WORD	Quando o buffer de anel recebido está cheio e começa a perder os frames antigos (muitos frames não tratados pelo dispositivo).
tModem.bConfigured	BOOL	Indica que o modem foi configurado na página da Web.
tModem.byConnectionState	BYTE	Estado da conexão do modem.
tModem.szIP	STRING (15)	String com endereço IP usado pelo modem.
tWifiAdapter.bConfigured	BOOL	Indica que o adaptador Wi-Fi foi configurado na página da Web.
tWifiAdapter.byConnectionState	BYTE	Estado da conexão do adaptador Wi-Fi.
tWifiAdapter.szIP	STRING (15)	String com endereço IP usado pelo Wi-Fi.
tWifiAdapter.szMask	STRING (15)	String com a máscara usada pelo Wi-Fi.
tWifiAdapter.szGateway	STRING (15)	String com o Gateway usado pelo Wi-Fi.
tWifiAdapter.szMAC	STRING (17)	String com o endereço MAC do adaptador Wi-Fi. (exclusivo para o dispositivo).

Tabela 215: Diagnósticos Detalhados Grupo USB

6.1.4.2.11. Diagnósticos Detalhados Grupo Ethernet

DG_Modulo.tDetailed.Ethernet.*	Tamanho	Descrição
NET[x].bLinkDown	BIT	Indica o estado do link na NET[x].
NET[x].byOperatingMode	BYTE	Indica o modo de operação da interface NET[x].
NET[x].byOperatingState	BYTE	Indica o estado de operação da interface NET[x].
NET[x].szIP	STRING (15)	Endereço IP NET[x].
NET[x].szMask	STRING (15)	Máscara de Subrede NET[x].
NET[x].szGateway	STRING (15)	Endereço Gateway NET[x].
NET[x].szMAC	STRING (17)	Endereço MAC NET[x].
NET[x].abyIP	BYTE ARRAY(4)	Endereço IP NET[x].
NET[x].abyMask	BYTE ARRAY(4)	Máscara de Subrede NET[x].
NET[x].abyGateway	BYTE ARRAY(4)	Endereço Gateway NET[x].
NET[x].abyMAC	BYTE ARRAY(6)	Endereço MAC NET[x].
NET[x].NICTeaming.bStandbyState	BIT	Indica que a interface está em estado reserva.
NET[x].NICTeaming.bActiveState	BIT	Indica que a interface está em estado ativo.
NET[x].NICTeaming.bLinkDown	BIT	Indica que o link de rede está inativo.
NET[x].NICTeaming.InterMsgTimeout	BIT	Indica ausência de comunicação com outra interface.
NET[x].NICTeaming.GeneralRxMsgTimeout	BIT	Indica ausência de atividade de comunicação na rede.
NET[x].dwTransmittedBytes	DWORD	Contador de bytes enviados através da porta NET[x] (0 a 4294967295).
NET[x].dwTransmittedPackets	DWORD	Contador de pacotes enviados através da porta NET[x] (0 a 4294967295).
NET[x].dwTransmittedDropErrors	DWORD	Contador de perdas de conexão na transmissão através da porta NET[x] (0 a 4294967295).
NET[x].dwTransmittedCollisionErrors	DWORD	Contador de erros de colisão na transmissão através da porta NET[x] (0 a 4294967295).
NET[x].dwReceivedBytes	DWORD	Contador de bytes recebidos através da porta NET[x] (0 a 4294967295).
NET[x].dwReceivedPackets	DWORD	Contador de pacotes recebidos através da porta NET[x] (0 a 4294967295).

DG_Modulo.tDetailed.Ethernet.*	Tamanho	Descrição
NET[x].dwReceivedDropErrors	DWORD	Contador de perdas de conexão na recepção através da porta NET[x] (0 a 4294967295).
NET[x].dwReceivedFrameErrors	DWORD	Contador de erros de frame na recepção através da porta NET[x] (0 a 4294967295).

Tabela 216: Diagnósticos Detalhados Grupo Ethernet NET[x]

Nota:

[x] é o número da interface ethernet da UCP, onde por exemplo, NET 1 representa o valor 1.

DG_Modulo.tDetailed.Ethernet.RSTP.*	Tamanho	Descrição
RSTP[x].bProtocolEnabled	BIT	Indica se o protocolo RSTP está habilitado.
RSTP[x].bBridgeIsRoot	BIT	Indica se a Bridge é a root.
RSTP[x].eProtocol	BYTE	Indica a versão do protocolo.
RSTP[x].sBridgeId	STRING (22)	ID da bridge local (prioridade e MAC).
RSTP[x].sRootId	STRING (22)	ID da bridge root (prioridade e MAC).
RSTP[x].eRootPort	BYTE	Indica qual é a porta root.
RSTP[x].dwRootPathCost	DWORD	Path Cost até a bridge root.
RSTP[x].byBridgeHelloTime	BYTE	Hello Time configurado.
RSTP[x].byRootHelloTime	BYTE	Hello Time aprendido.
RSTP[x].byBridgeMaxAge	BYTE	Max Age configurado.
RSTP[x].byRootMaxAge	BYTE	Max Age aprendido.
RSTP[x].byBridgeForwDelay	BYTE	Forward Delay configurado.
RSTP[x].byRootForwDelay	BYTE	Forward Delay aprendido.
RSTP[x].dwTxHoldCount	DWORD	Limite de BPDUs transmitidos por ciclo.
RSTP[x].dwTimeSinceTC	DWORD	Tempo, em segundos, desde a última alteração de topologia.
RSTP[x].dwTCCCount	DWORD	Número de vezes que a topologia foi alterada.
RSTP[x].aPort[y].eName	BYTE	Nome da porta.
RSTP[x].aPort[y].eRole	BYTE	Função da porta.
RSTP[x].aPort[y].eState	BYTE	Estado da porta.
RSTP[x].aPort[y].wIdentity	WORD	ID da porta.
RSTP[x].aPort[y].dwPathCost	DWORD	Custo da porta.
RSTP[x].aPort[y].bOperEdge	BIT	Indica se a porta está trabalhando como Edge.
RSTP[x].aPort[y].bOperP2P	BIT	Indica se a porta está trabalhando como Point-to-Point.
RSTP[x].aPort[y].dwNumTxBPDUs	DWORD	Número de frames BPDUs transmitidos na porta.
RSTP[x].aPort[y].dwNumRxBPDUs	DWORD	Número de frames BPDUs recebidos na porta.

Tabela 217: Diagnósticos Detalhados Grupo Ethernet RSTP

Nota:

[x]: é o número da bridge.

[y]: é o número da porta dentro da bridge.

DG_Modulo.tDetailed.Ethernet.MRP.*	Tamanho	Descrição
MRP[x].eConfiguredRingRole	BYTE	Indica qual a configuração do dispositivo (Manager ou Client)
MRP[x].eCurrentRingRole	BYTE	Indica a configuração atual do dispositivo
MRP[x].eRingState	BYTE	Indica o estado do anel
MRP[x].ePrimaryPort	BYTE	Indica qual a NET principal do MRP
MRP[x].eSecondaryPort	BYTE	Indica qual a NET secundária do MRP
MRP[x].eManagerDiagnostics	BYTE	Indica tipo de diagnostico Manager

Tabela 218: Diagnósticos Detalhados Grupo Ethernet MRP

Nota:

[x]: é o número do MRP.

6.1.4.2.12. Diagnósticos Detalhados Grupo UserFiles

DG_Modulo.tDetailed.UserFiles.*	Tamanho	Descrição
byMounted	BYTE	Indica se a memória utilizada para gravar arquivos de usuário está apta para receber os dados.
dwFreeSpacekB	DWORD	Espaço livre da memória de arquivos de usuário em Kbytes.
dwTotalSizekB	DWORD	Capacidade de armazenamento da memória de arquivos de usuário em Kbytes.

Tabela 219: Diagnósticos Detalhados Grupo UserFiles

Nota:

User Partition: 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.

6.1.4.2.13. Diagnósticos Detalhados Grupo UserLogs

DG_Modulo.tDetailed.UserLogs.*	Tamanho	Descrição
byMounted	BYTE	Status da memória em que são inseridos os logs de usuário.
wFreeSpacekB	WORD	Espaço livre na memória de logs de usuário em Kbytes.
wTotalSizekB	WORD	Capacidade de armazenamento da memória de logs de usuário em Kbytes.

Tabela 220: Diagnósticos Detalhados Grupo UserLogs

6.1.4.2.14. Diagnósticos Detalhados Grupo MemoryCard

DG_Modulo.tDetailed.MemoryCard.*	Tamanho	Descrição
byMounted	BYTE	Status do Cartão de Memória: 00: Cartão de memória não montado 01: Cartão de memória inserido e montado
bEnable	BIT	Cartão de Memória habilitado.
dwFreeSpacekB	DWORD	Espaço livre no Cartão de Memória em Kbytes.
dwTotalSizekB	DWORD	Capacidade de armazenamento do Cartão de Memória em Kbytes.

Tabela 221: Diagnósticos Detalhados Grupo MemoryCard

6.1.4.2.15. Diagnósticos Detalhados Grupo WHSB

DG_Modulo.tDetailed.WHSB.*	Tamanho	Descrição
byHotSwapAndStartupStatus	BYTE	Informa a situação anormal no barramento que ocasionou a parada da aplicação, para cada Modo de troca a quente. Consulte a Tabela 223 para detalhes das possibilidades.
adwRackIOErrorStatus	DWORD ARRAY (32)	Identificação de erros em módulos de E/S, individualmente. Para mais informações sobre este diagnóstico consultar a nota abaixo.
adwModulePresenceStatus	DWORD ARRAY (32)	Status de presença de módulos declarados de E/S em barramentos, individualmente. Para mais informações sobre este diagnóstico consultar a nota abaixo.
byWHSBBusErrors	BYTE	Contador de falhas no barramento WHSB. Esse contador é reinicializado na energização (0 a 255).

Tabela 222: 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".
05	CFG_MISMATCH_HOT_SWAP_STARTUP_CONSISTENCY	Aplicação em Modo Stop devido à ativação do diagnóstico Configuração Incompatível, quando configurada troca a quente "Habilitada, com consistência na partida" ou troca a quente "Habilitada, com consistência na partida somente para os módulos declarados".
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 "Desabilitada" ou troca a quente "Desabilitada, apenas para módulos declarados".
13	DUPLICATED_SLOT_HOT_SWAP_STARTUP_CONSISTENCY	Aplicação em Modo Stop devido à ativação do diagnóstico Slots Duplicados, 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
14	DUPLICATED_SLOT_HOT_SWAP_ENABLED	Aplicação em Modo Stop devido à ativação do diagnóstico Slots Duplicados, quando configurada troca a quente "Habilitada, sem consistência na partida".
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 "Habilitada, com consistência na partida".
16	NON_DECLARED_MODULE_HOT_SWAP_DISABLED	Aplicação em Modo Stop devido à ativação do diagnóstico Módulos Não Declarados, quando configurada troca a quente "Desabilitada".

Tabela 223: Códigos de Situações que Ocasionam Parada da Aplicação

6.1.4.2.16. Diagnósticos Detalhados Grupo Application

DG_Modulo.tDetailed.Application.*	Tamanho	Descrição
byCPUState	BYTE	Informa o estado de operação da UCP: 01: Aplicação em execução (Modo Run) 03: Aplicação parada (Modo Stop)
bForcedIOS	BIT	Existem um ou mais pontos de E/S forçados.
bNetDefinedByWeb	BIT	O endereço IP está definido a partir da Página Web de Sistema

Tabela 224: Diagnósticos Detalhados Grupo Application

6.1.4.2.17. Diagnósticos Detalhados Grupo Rack

DG_Modulo.tDetailed.Rack.*	Tamanho	Descrição
dwAbsentRacks	DWORD	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.
dwDuplicatedRacks	DWORD	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.
dwNonDeclaredRacks	DWORD	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 225: Diagnósticos Detalhados Grupo Rack

6.1.4.2.18. Diagnósticos Detalhados Grupo ApplicationInfo

DG_Modulo.tDetailed.ApplicationInfo.*	Tamanho	Descrição
dwApplicationCRC	DWORD	CRC de 32 bits da aplicação. Quando a aplicação é modificada e enviada para a UCP, um novo CRC é calculado.

Tabela 226: Diagnósticos Detalhados Grupo ApplicationInfo

6.1.4.2.19. Diagnósticos Detalhados Grupo SNTP

DG_Modulo.tDetailed.SNTP.*	Tamanho	Descrição
bServiceEnabled	BIT	Serviço SNTP habilitado.

DG_Modulo.tDetailed.SNTP.*	Tamanho	Descrição
byActiveTimeServer	BYTE	Indica qual servidor está ativo: 00: Nenhum servidor ativo. 01: Servidor primário ativo. 02: Servidor secundário ativo.
wPrimaryServerDownCount	WORD	Contador de vezes que o servidor primário esteve indisponível (0 a 65535).
wSecondaryServerDownCount	WORD	Contador de vezes que o servidor secundário esteve indisponível (0 a 65535).
dwRTCTimeUpdatedCount	DWORD	Contador de vezes que o RTC foi atualizado pelo serviço SNTP (0 a 4294967295).
byLastUpdateSuccessful	BYTE	Indica status da última atualização: 00: Não foi atualizado. 01: Última atualização falhou. 02: Última atualização teve sucesso.
byLastUpdateTimeServer	BYTE	Indica qual servidor foi utilizado na última atualização: 00: Nenhuma atualização. 01: Servidor primário. 02: Servidor secundário.
sLastUpdateTime.byDayOfMonth	BYTE	Dia da última atualização do RTC.
sLastUpdateTime.byMonth	BYTE	Mês da última atualização do RTC.
sLastUpdateTime.wYear	WORD	Ano da última atualização do RTC.
sLastUpdateTime.byHours	BYTE	Hora da última atualização do RTC.
sLastUpdateTime.byMinutes	BYTE	Minuto da última atualização do RTC.
sLastUpdateTime.bySeconds	BYTE	Segundo da última atualização do RTC.
sLastUpdateTime.wMilliseconds	WORD	Milissegundo da última atualização do RTC.

Tabela 227: Diagnósticos Detalhados Grupo SNTP

6.1.4.2.20. Diagnósticos Detalhados Grupo OpenVPN

DG_Modulo.tDetailed.OpenVPN.*	Tamanho	Descrição
byOperationMode	BYTE	Modo de operação da VPN: SERVER(0): Dispositivo operando como servidor. CLIENT(1): Dispositivo operando como cliente.
bServiceEnabled	BIT	Serviço de VPN habilitado.
bServiceRunning	BIT	Serviço de VPN em execução.
byConnectionState	BYTE	Estado de conexão da VPN: DISCONNECTED(0): Dispositivo desconectado. CONNECTING(1): Dispositivo conectando-se. CONNECTED(2): Dispositivo conectado.
uliConnectionTime	ULINT	Tempo de conexão atual em segundos.
sIPAddress	STRING(15)	Endereço IP da VPN.
dwConnectedClients	DWORD	Quantidade de clientes conectados.
dwTransmittedBytes	DWORD	Quantidade de bytes transmitidos através da VPN.
dwReceivedBytes	DWORD	Quantidade de bytes recebidos através da VPN.
CACertificate.CertificateName	STRING(64)	Nome do certificado CA.
CACertificate.CommonName	STRING(64)	Nome comum do certificado CA.
CACertificate.sStartDate	EXTENDED_DATE_AND_TIME	Data e hora de início do certificado CA.
CACertificate.sExpirationDate	EXTENDED_DATE_AND_TIME	Data e hora de expiração do certificado CA.
CACertificate.dwValidDaysLeft	DWORD	Dias até a expiração do certificado CA.
DeviceCertificate.CertificateName	STRING(64)	Nome do certificado do dispositivo.
DeviceCertificate.CommonName	STRING(64)	Nome comum do certificado do dispositivo.
DeviceCertificate.sStartDate	EXTENDED_DATE_AND_TIME	Data e hora de início do certificado do dispositivo.
DeviceCertificate.sExpirationDate	EXTENDED_DATE_AND_TIME	Data e hora de expiração do certificado do dispositivo.
DeviceCertificate.dwValidDaysLeft	DWORD	Dias até a expiração do certificado do dispositivo.
sDeviceKeyName	STRING(64)	Nome da chave privada do dispositivo.

Tabela 228: Diagnósticos Detalhados Grupo OpenVPN

6.1.4.2.21. Diagnósticos Detalhados Grupo Firewall

DG_Modulo.tDetailed.Firewall.*	Tamanho	Descrição
bServiceEnabled	BIT	O serviço de Firewall está habilitado.
sLatsModificationDateUTC	NextoStandard. EXTENDED_DATE_AND_TIME	Data e hora da última alteração da configuração em UTC.

Tabela 229: Diagnósticos Detalhados Grupo Firewall

6.1.4.2.22. Diagnósticos Detalhados Grupo FTP

DG_Modulo.tDetailed.FTP.*	Tamanho	Descrição
bServiceEnabled	BIT	Serviço de FTP habilitado.
dwConnectedClients	DWORD	Quantidade de clientes conectados.
sLastUpdateTime.byDayOfMonth	BYTE	Dia da última atualização do FTP.
sLastUpdateTime.byMonth	BYTE	Mês da última atualização do FTP.
sLastUpdateTime.wYear	WORD	Ano da última atualização do FTP.
sLastUpdateTime.byHours	BYTE	Hora da última atualização do FTP.
sLastUpdateTime.byMinutes	BYTE	Minuto da última atualização do FTP.
sLastUpdateTime.bySeconds	BYTE	Segundo da última atualização do FTP.
sLastUpdateTime.wMilliseconds	WORD	Milissegundo da última atualização do FTP.

Tabela 230: Diagnósticos Detalhados Grupo FTP

6.1.5. 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.

6.1.5.1. GetTaskInfo

Essa função retorna informações sobre uma tarefa de uma determinada aplicação.



Figura 229: 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 231: 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 232: 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 NextoStandard.stTaskInfo;
  TaskInfo : NextoStandard.stTaskInfo;
  Info : NextoStandard.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.

```

6.2. Visor Gráfico

O visor gráfico disponível neste produto 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.

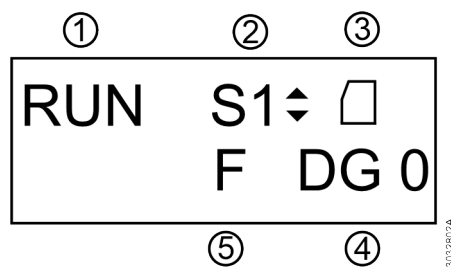


Figura 230: Tela de Status da UCP

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. Caso a UCP esteja em exceção o estado será EXCP. Para maiores detalhes, consultar a seção [Estados de Operação da UCP](#).
- 2. 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 [Interface Serial](#).
- 3. Indicação da presença do Cartão de Memória. Para maiores detalhes sobre a instalação do cartão, consultar a seção [Instalação do Cartão de Memória](#).
- 4. 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](#).
- 5. 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](#).

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 FORMATACAO	Indica que ocorreu um erro durante a formatação do cartão de memória pela UCP.
SEM TAG	Não existe Tag configurada para a UCP no Mastertool.
SEM DESC	Não existe Descrição configurada para a UCP no Mastertool.
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 233: Outras Mensagens do Visor Gráfico

6.3. Log de Sistema

O *Log de Sistema* é um recurso disponível no programador Mastertool. 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 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.

6.4. 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 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.

6.5. 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 Mastertool 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.

6.6. 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.
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.

Tabela 234: Solução de Problemas

6.7. 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.

7. Apêndices

7.1. Gerenciamento de Certificados e Chaves TLS

Esta seção aborda a geração de arquivos de segurança, certificados e chaves, utilizando o TLS. Os certificados comentados a seguir são assinados por CA, este tipo de certificado considera uma entidade, denominada Autoridade Certificadora (CA), para gerar os certificados. Esta entidade pode ser um serviço de autoridade oficial ou um simples computador. Só é necessário restringir o acesso à CA para evitar qualquer quebra de segurança, uma vez que esta entidade pode gerar certificados para qualquer dispositivo. A imagem abaixo mostra como cada dispositivo interage com os arquivos.

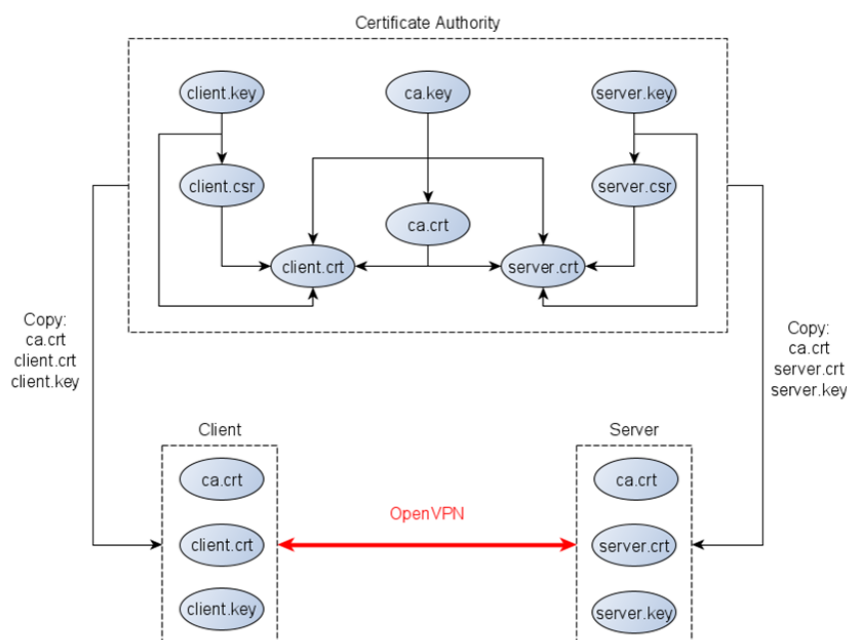


Figura 231: Fluxo de Geração de Certificados TLS

Em primeiro lugar, os arquivos gerados são chaves privadas. Cada dispositivo possui seu arquivo de chave, criado pela entidade CA ou pelo próprio dispositivo. O arquivo mais importante é a chave privada da CA *ca.key*, que não deve sair da entidade. A entidade CA gera seu certificado com base em sua chave privada *ca.crt*. Este certificado é um arquivo público usado pelos dispositivos para validar a conexão VPN. A geração de certificados do dispositivo requer primeiro um arquivo de solicitação (*.csr* ou *.req* dependendo da ferramenta) com base na chave privada do dispositivo. Este documento apresenta duas ferramentas diferentes para gerar os arquivos de certificado: Easy-RSA e OpenSSL.

Certifique-se de estar com a data e hora configuradas corretamente na entidade CA para que a geração dos certificados seja com base em uma configuração atual.

7.1.1. Geração de Certificados por Easy-RSA

O projeto OpenVPN fornece essa ferramenta para ajudar com o certificado e as chaves. O Easy-RSA está disponível para Windows e Linux. Veja abaixo o passo a passo para gerar os arquivos em uma configuração do Windows:

- 1- Abra um prompt do Windows na pasta Easy-RSA e execute o seguinte comando para entrar no terminal da ferramenta:

```
.\EasyRSA-Start.bat
```



```
C:\Users\igor.franco\Downloads\EasyRSA-3.0.8-win64\EasyRSA-3.0.8>.\EasyRSA-Start.bat

Welcome to the EasyRSA 3 Shell for Windows.
Easy-RSA 3 is available under a GNU GPLv2 license.

Invoke './easysrsa' to call the program. Without commands, help is displayed.

EasyRSA Shell
#
```

Figura 232: Geração de Certificado utilizando o Easy-RSA (passo 1)

2- Copie o arquivo *vars.example* e renomeie-o para *vars* na pasta de ferramentas.

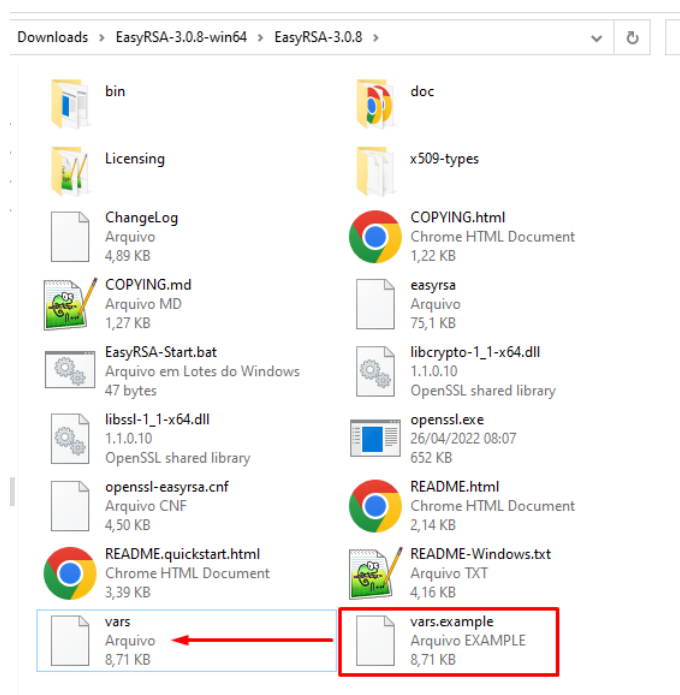


Figura 233: Geração de Certificado utilizando o Easy-RSA (passo 2)

3- Abra o arquivo *vars* com um editor de texto e altere as informações da Autoridade de Certificação.

```

75
76 #set_var EASYRSA_TEMP_DIR "$EASYRSA_PKI"
77
78 # Define X509 DN mode.
79 # This is used to adjust what elements are included in the Subject field as the DN
80 # (this is the "Distinguished Name.")
81 # Note that in cn_only mode the Organizational fields further below aren't used.
82 #
83 # Choices are:
84 #   cn_only - use just a CN value
85 #   org     - use the "traditional" Country/Province/City/Org/OU/email/CN format
86
87 #set_var EASYRSA_DN "cn_only"
88
89 # Organizational fields (used with 'org' mode and ignored in 'cn_only' mode.)
90 # These are the default values for fields which will be placed in the
91 # certificate. Don't leave any of these fields blank, although interactively
92 # you may omit any specific field by typing the "." symbol (not valid for
93 # email.)
94
95 #set_var EASYRSA_REQ_COUNTRY  "BR"
96 #set_var EASYRSA_REQ_PROVINCE "Rio Grande do Sul"
97 #set_var EASYRSA_REQ_CITY     "Sao Leopoldo"
98 #set_var EASYRSA_REQ_ORG      "Altus SA"
99 #set_var EASYRSA_REQ_EMAIL     "someemail@altus.com.br"
100 #set_var EASYRSA_REQ_OU       "APED"
101
102 # Choose a size in bits for your keypairs. The recommended value is 2048. Using
103 # 2048-bit keys is considered more than sufficient for many years into the
104 # future. Larger key sizes will slow down TLS negotiation and make key/DH param
105 # generation take much longer. Values up to 4096 should be accepted by most
106 # software. Only used when the crypto alg is rsa (see below.)
107
108 #set_var EASYRSA_KEY_SIZE 2048
109
110 # The default crypto mode is rsa; ec can enable elliptic curve support.
111 # Note that not all software supports ECC, so use care when enabling it.
112 # Choices for crypto alg are: (each in lower-case)
113 #   * rsa
114 #   * ec
115 #   * ed

```

Figura 234: Geração de Certificado utilizando o Easy-RSA (passo 3)

4- Use o comando a seguir para preparar a configuração:

```
./easysrsa init-pki
```

```

# ./easysrsa init-pki
Note: using Easy-RSA configuration from: ./vars
path = C:/Users/igor.franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/easy-rsa-13020.a14492/tmp.XXXXXX
lpPathBuffer = C:/Users/IGOR~1.FRA/AppData/Local/Temp/
szTempName = C:/Users/IGOR~1.FRA/AppData/Local/Temp/tmpC051.tmp
path = C:/Users/IGOR~1.FRA/AppData/Local/Temp/tmpC051.tmp
fd = 3

init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: C:/Users/igor.franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki

EasyRSA Shell
#

```

Figura 235: Geração de Certificado utilizando o Easy-RSA (passo 4)

5- Em seguida, digite o seguinte para gerar o certificado CA.

```
./easysrsa build-ca nopass
```

```
# ./easyrsa build-ca nopass

Note: using Easy-RSA configuration from: ./vars
Using SSL: openssl OpenSSL 1.1.0j  20 Nov 2018
path = C:/Users/igor.franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/easy-rsa-6996.a08916/tmp.XXXXXX
lpPathBuffer = C:/Users/IGOR~1.FRA/AppData/Local/Temp/
szTempName = C:/Users/IGOR~1.FRA/AppData/Local/Temp/tmp4FB0.tmp
path = C:/Users/IGOR~1.FRA/AppData/Local/Temp/tmp4FB0.tmp
fd = 3
path = C:/Users/igor.franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/easy-rsa-6996.a08916/tmp.XXXXXX
lpPathBuffer = C:/Users/IGOR~1.FRA/AppData/Local/Temp/
szTempName = C:/Users/IGOR~1.FRA/AppData/Local/Temp/tmp505C.tmp
path = C:/Users/IGOR~1.FRA/AppData/Local/Temp/tmp505C.tmp
fd = 3
Generating RSA private key, 2048 bit long modulus
.....+++++
e is 65537 (0x010001)
path = C:/Users/igor.franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/easy-rsa-6996.a08916/tmp.XXXXXX
lpPathBuffer = C:/Users/IGOR~1.FRA/AppData/Local/Temp/
szTempName = C:/Users/IGOR~1.FRA/AppData/Local/Temp/tmp5194.tmp
path = C:/Users/IGOR~1.FRA/AppData/Local/Temp/tmp5194.tmp
fd = 3
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:CA-Entity

CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
C:/Users/igor.franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/ca.crt

EasyRSA Shell
#
```

Figura 236: Geração de Certificado utilizando o Easy-RSA (passo 5)

6- Gere a chave do dispositivo e solicite arquivos usando o seguinte comando (altere o *DeviceName* com o nome comum desejado):

```
./easyrsa gen-req DeviceName nopass
```

Novamente, remova o argumento *nopass* para usar uma senha para o arquivo de certificado. Ao entrar com o Nome Comum como argumento, basta pressionar enter quando solicitado (quadrado vermelho).

```
# ./easyrsa gen-req DeviceName nopass

Note: using Easy-RSA configuration from: ./vars
Using SSL: openssl OpenSSL 1.1.0j  20 Nov 2018
path = C:/Users/igor.franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/easy-rsa-16216.a13904/tmp.XXXXXX
lpPathBuffer = C:/Users/IGOR~1.FRA/AppData/Local/Temp/
szTempName = C:/Users/IGOR~1.FRA/AppData/Local/Temp/tmp150B.tmp
path = C:/Users/IGOR~1.FRA/AppData/Local/Temp/tmp150B.tmp
fd = 3
path = C:/Users/igor.franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/easy-rsa-16216.a13904/tmp.XXXXXX
lpPathBuffer = C:/Users/IGOR~1.FRA/AppData/Local/Temp/
szTempName = C:/Users/IGOR~1.FRA/AppData/Local/Temp/tmp1696.tmp
path = C:/Users/IGOR~1.FRA/AppData/Local/Temp/tmp1696.tmp
fd = 3
path = C:/Users/igor.franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/easy-rsa-16216.a13904/tmp.XXXXXX
lpPathBuffer = C:/Users/IGOR~1.FRA/AppData/Local/Temp/
szTempName = C:/Users/IGOR~1.FRA/AppData/Local/Temp/tmp1742.tmp
path = C:/Users/IGOR~1.FRA/AppData/Local/Temp/tmp1742.tmp
fd = 3
Generating a RSA private key
.....+++++
.....+++++
writing new private key to 'C:/Users/igor.franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/easy-rsa-16216.a13904/tmp.a02420'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [DeviceName]:

Keypair and certificate request completed. Your files are:
req: C:/Users/igor.franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/reqs/DeviceName.req
key: C:/Users/igor.franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/private/DeviceName.key

EasyRSA Shell
#
```

Figura 237: Geração de Certificado utilizando o Easy-RSA (passo 6)

7- Por fim, digite o seguinte comando para gerar o certificado do dispositivo (o *DeviceName* é o nome comum desejado e o *servidor* é o tipo; use *cliente* se estiver gerando para um cliente VPN).

```
./easyrsa sign-req server DeviceName
```

```
# ./easyrsa sign-req server DeviceName
Note: using Easy-RSA configuration from: ./vars
Using SSL: openssl OpenSSL 1.1.0j 20 Nov 2018

You are about to sign the following certificate.
Please check over the details shown below for accuracy. Note that this request
has not been cryptographically verified. Please be sure it came from a trusted
source or that you have verified the request checksum with the sender.

Request subject, to be signed as a server certificate for 825 days:

subject=
  commonName           = DeviceName

Type the word 'yes' to continue, or any other input to abort.
Confirm request details: yes
path = C:/Users/igor.franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/easy-rsa-9368.a06604/tmp.XXXXXX
lpPathBuffer = C:/Users/IGOR~1.FRA/AppData/Local/Temp\
szTempName = C:/Users/IGOR~1.FRA/AppData/Local/Temp/tmpC79.tmp
path = C:/Users/IGOR~1.FRA/AppData/Local/Temp/tmpC79.tmp
fd = 3
path = C:/Users/igor.franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/easy-rsa-9368.a06604/tmp.XXXXXX
lpPathBuffer = C:/Users/IGOR~1.FRA/AppData/Local/Temp\
szTempName = C:/Users/IGOR~1.FRA/AppData/Local/Temp/tmpF29.tmp
path = C:/Users/IGOR~1.FRA/AppData/Local/Temp/tmpF29.tmp
fd = 3
path = C:/Users/igor.franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/easy-rsa-9368.a06604/tmp.XXXXXX
lpPathBuffer = C:/Users/IGOR~1.FRA/AppData/Local/Temp\
szTempName = C:/Users/IGOR~1.FRA/AppData/Local/Temp/tmp18FE.tmp
path = C:/Users/IGOR~1.FRA/AppData/Local/Temp/tmp18FE.tmp
fd = 3
path = C:/Users/igor.franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/easy-rsa-9368.a06604/tmp.XXXXXX
lpPathBuffer = C:/Users/IGOR~1.FRA/AppData/Local/Temp\
szTempName = C:/Users/IGOR~1.FRA/AppData/Local/Temp/tmp11AA.tmp
path = C:/Users/IGOR~1.FRA/AppData/Local/Temp/tmp11AA.tmp
fd = 3
Using configuration from C:/Users/igor.franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/easy-rsa-9368.a06604/tm
p.a16388
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'DeviceName'
Certificate is to be certified until Jul 29 12:59:53 2024 GMT (825 days)

Write out database with 1 new entries
Data Base Updated

Certificate created at: C:/Users/igor.franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/issued/DeviceName.crt

EasyRSA Shell
#
```

Figura 238: Geração de Certificado utilizando o Easy-RSA (passo 7)

8- Repita as etapas 6 e 7 para gerar mais certificados de dispositivo.

9- Encontre o *ca.crt* na pasta *pki*, as chaves privadas do dispositivo no caminho *pki/private* e os certificados do dispositivo no diretório *pki/issued*.

Downloads > EasyRSA-3.0.8-win64 > EasyRSA-3.0.8 > pki				Pesquisar pki
Nome	Data de modificação	Tipo	Tamanho	
certs_by_serial	26/04/2022 09:59	Pasta de arquivos		
issued	26/04/2022 09:59	Pasta de arquivos		
private	26/04/2022 09:35	Pasta de arquivos		
renewed	26/04/2022 08:57	Pasta de arquivos		
reqs	26/04/2022 09:35	Pasta de arquivos		
revoked	26/04/2022 08:57	Pasta de arquivos		
ca.crt	26/04/2022 09:05	Certificado de Seg...	2 KB	
index.txt	26/04/2022 09:59	Arquivo TXT	1 KB	
index.txt.attr	26/04/2022 09:59	Arquivo ATTR	1 KB	
index.txt.attr.old	26/04/2022 09:55	Arquivo OLD	1 KB	
index.txt.old	26/04/2022 09:55	Arquivo OLD	1 KB	
openssl-easyrsa.cnf	26/04/2022 08:54	Arquivo CNF	5 KB	
safessl-easyrsa.cnf	26/04/2022 08:54	Arquivo CNF	5 KB	
serial	26/04/2022 09:59	Arquivo	1 KB	
serial.old	26/04/2022 09:59	Arquivo OLD	1 KB	

Figura 239: Geração de Certificado utilizando o Easy-RSA (passo 9)

7.1.2. Geração de Certificados por OpenSSL

O OpenSSL é um pacote de código aberto com ferramentas que ajudam a gerar muitos arquivos e recursos de segurança. Este pacote é nativo para a maioria das distribuições do Linux e está disponível para Windows. Apenas lembre-se de definir a pasta OpenSSL no PATH (variável de ambiente) para permitir o uso do comando de qualquer lugar através do prompt. Encontre abaixo o passo a passo utilizando este recurso (todos os arquivos podem ter qualquer nome conforme desejado, os passos consideram apenas um exemplo):

- 1- Abra um prompt na pasta do certificado (onde criará os arquivos).
- 2- Gere a chave privada da CA com o seguinte comando:

```
openssl genrsa -out ca.key 4096
```

```
C:\Users\igor.franco\Downloads\Certificate>openssl genrsa -out ca.key 4096
Generating RSA private key, 4096 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
C:\Users\igor.franco\Downloads\Certificate>
```

Figura 240: Geração de Certificado utilizando o OpenSSL (passo 2)

- 3- Em seguida, gere o certificado CA com base na chave privada, utilizando o comando a seguir.

```
openssl req -new -x509 -days 365 -key ca.key -out ca.crt
```

O parâmetro *-days* representa o tempo de expiração do certificado. Configure-o como desejar. Neste exemplo, o certificado é válido por um ano. Preencha os valores solicitados no prompt conforme necessário (pressione enter para usar o padrão, que está entre colchetes []). É obrigatório definir um Nome Comum para o trabalho do certificado.

```
C:\Users\igor.franco\Downloads\Certificate>openssl req -new -x509 -days 365 -key ca.key -out ca.crt
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:CA-Entity
Email Address []:
C:\Users\igor.franco\Downloads\Certificate>
```

Figura 241: Geração de Certificado utilizando o OpenSSL (passo 3)

- 4- Agora, gere a chave privada do dispositivo, semelhante à etapa 2, utilizando o seguinte comando:

```
openssl genrsa -out DeviceName.key 2048
```

```
C:\Users\igor.franco\Downloads\Certificate>openssl genrsa -out DeviceName.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
C:\Users\igor.franco\Downloads\Certificate>
```

Figura 242: Geração de Certificado utilizando o OpenSSL (passo 4)

5- Depois disso, gere o arquivo de solicitação de certificado com base na chave privada, utilizando o comando a seguir:

```
openssl req -new -key DeviceName.key -out DeviceName.csr
```

Insira as informações desejadas e lembre-se de usar um nome comum diferente da CA.

```
C:\Users\igor.franco\Downloads\Certificate>openssl req -new -key DeviceName.key -out DeviceName.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:DeviceName
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
C:\Users\igor.franco\Downloads\Certificate>
```

Figura 243: Geração de Certificado utilizando o OpenSSL (passo 5)

6- Por fim, gere o certificado do dispositivo usando a chave privada da CA, o certificado da CA e o arquivo de solicitação de certificado do dispositivo, utilizando o comando que segue:

```
openssl x509 -req -days 365 -in DeviceName.csr -CA ca.crt -CAkey ca.key -set_serial 01 -out DeviceName.crt
```

Defina a data de expiração conforme desejado com o parâmetro *-days* e o número de série do certificado com o argumento *-set_serial*.

```
C:\Users\igor.franco\Downloads\Certificate>openssl x509 -req -days 365 -in DeviceName.csr -CA ca.crt -CAkey ca.key -set_serial 01 -out DeviceName.crt
Signature ok
subject=C = AU, ST = Some-State, O = Internet Widgits Pty Ltd, CN = DeviceName
Getting CA Private Key
C:\Users\igor.franco\Downloads\Certificate>
```

Figura 244: Geração de Certificado utilizando o OpenSSL (passo 6)

7- Repita as etapas 4 a 6 para qualquer novo dispositivo.

8- (Opcional) O OpenSSL oferece uma ferramenta para verificar se o certificado do dispositivo funciona com CA:

Utilize o seguinte comando:

```
openssl verify -purpose sslserver -CAfile ca.crt DeviceName.crt
```

```
C:\Users\igor.franco\Downloads\Certificate>openssl verify -purpose sslserver -CAfile ca.crt DeviceName.crt
DeviceName.crt: OK
```

Figura 245: Geração de Certificado utilizando o OpenSSL (passo 8)

7.1.3. Geração de Chave TA pelo OpenVPN

O projeto OpenVPN fornece uma ferramenta para geração de uma chave TLS, comumente chamada de ta.key. Esta chave é uma camada de proteção extra nas portas UDP/TCP de comunicação do OpenVPN, sendo assim, a utilização desta chave pode ser interpretada como um Firewall de HMAC para a comunicação VPN, fazendo com que seja necessário a existência do parâmetro dos dois lados da comunicação para que ela seja estabelecida.

A geração desta chave no Windows pode ser feita com o comando a seguir:

```
openvpn --genkey secret ta.key
```

```
C:\Program Files\OpenVPN\bin>openvpn --genkey secret C:\Users\bruno.berwanger\Desktop\Chaves\ta.key
C:\Program Files\OpenVPN\bin>
```

Figura 246: Exemplo de Geração de Chave TA no Windows

Para executar o comando foi utilizado o executável que é instalado juntamente ao pacote do OpenVPN. O diretório utilizado na imagem a cima é um exemplo e é opcional, pode ser utilizado somente o nome do arquivo desejado.

O comando pode ser utilizado para realizar a geração da chave pelo Linux, porém, há uma pequena alteração no comando em relação ao windows. Para gerar a chave no Linux utilize o seguinte comando:

```
openvpn --genkey --secret ta.key
```

```
developer@developer:~$ openvpn --genkey --secret ta.key
developer@developer:~$
```

Figura 247: Exemplo de Geração de Chave TA no Linux

Este parâmetro não é obrigatório para a comunicação VPN, porém, caso o servidor esteja utilizando, todos os seus clientes devem também utilizar, sendo que a chave do servidor e dos clientes deve ser a mesma.