



Manual de Utilização Nexto Xpress

MU216000 Rev. AA

31 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.	Documentos Relacionados a este Manual	2
1.2.	Inspeção Visual	2
1.3.	Suporte Técnico	3
1.4.	Mensagens de Advertência Utilizadas neste Manual	3
2.	Descrição Técnica	4
2.1.	Painéis e Conexões	4
2.2.	Características do Produto	5
2.2.1.	Características Gerais	5
2.2.2.	Normas e Certificações	7
2.2.3.	Memória	8
2.2.4.	Protocolos	9
2.2.5.	RS-485	10
2.2.6.	CAN	10
2.2.7.	USB	10
2.2.8.	Ethernet	11
2.2.9.	Alimentação	11
2.2.10.	Entradas Digitais	12
2.2.11.	Entradas Rápidas	12
2.2.12.	Saídas Digitais	13
2.2.13.	Saídas Rápidas	13
2.2.14.	Entradas Analógicas	14
2.2.15.	Saídas Analógicas	16
2.3.	Compatibilidade com Outros Produtos	17
2.4.	Desempenho	17
2.4.1.	Tempo de Intervalo	17
2.4.2.	Tempos da Aplicação	18
2.4.3.	Tempo para a Execução de Instruções	18
2.4.4.	Tempo de Inicialização	18
2.5.	Dimensões Físicas	19
2.6.	Dados de Compra	20
2.6.1.	Itens Integrantes	20
2.6.2.	Código do Produto	20
2.7.	Produtos Relacionados	21
3.	Instalação	22
3.1.	Instalação Mecânica	22
3.1.1.	Instalando o Controlador	23
3.1.2.	Removendo o Controlador	24

3.2.	Instalação Elétrica	25
3.3.	Conexão de Rede Ethernet	27
3.3.1.	Endereço IP	27
3.3.2.	ARP Gratuito	27
3.3.3.	Instalação do Cabo de Rede	27
3.4.	Conexão de Rede Serial RS-485 e CAN	28
4.	Programação Inicial	29
4.1.	Organização e Acesso à Memória	29
4.2.	Perfis de Projeto	31
4.2.1.	Perfil de Máquina	31
4.2.2.	Perfil Fast	31
4.3.	Configurando a UCP	32
4.4.	Bibliotecas	33
4.5.	Inserindo uma Instância de Protocolo	33
4.5.1.	MODBUS Ethernet	34
4.6.	Localizando o Dispositivo	35
4.7.	Login	37
4.8.	Modo Run	39
4.9.	Modo Stop	41
4.10.	Escrita e Forçamento de Variáveis	41
4.11.	Logout	42
4.12.	Upload do Projeto	42
4.13.	Estados de Operação da UCP	44
4.13.1.	Run	44
4.13.2.	Stop	44
4.13.3.	Breakpoint	44
4.13.4.	Exception	44
4.13.5.	Reset a Quente	44
4.13.6.	Reset a Frio	44
4.13.7.	Reset Origem	44
4.13.8.	Reset Process Command (IEC 60870-5-104)	45
4.14.	Programas (POUs) e Listas de Variáveis Globais (GVLs)	45
4.14.1.	Programa MainPrg	45
4.14.2.	Programa StartPrg	45
4.14.3.	Programa UserPrg	46
4.14.4.	GVL IntegratedIO	46
4.14.5.	GVL System_Diagnostics	46
4.14.6.	GVL Disables	47
4.14.7.	GVL Qualities	48
4.14.8.	GVL ReqDiagnostics	50
5.	Configuração	52
5.1.	Device	52
5.1.1.	Gerenciamento de Usuários e Direitos de Acesso	52
5.1.2.	Configurações do CP	52
5.2.	Configuração da UCP	54
5.2.1.	Parâmetros Gerais	54
5.2.2.	Sincronização de Tempo	55
5.2.2.1.	IEC 60870-5-104	56

5.2.2.2.	SNTP	56
5.2.2.3.	Horário de Verão	57
5.2.3.	Pontos Internos	57
5.2.3.1.	Conversões de Qualidade	59
5.2.3.1.1.	Qualidade Interna	59
5.2.3.1.2.	Conversão IEC 60870-5-104	60
5.2.3.1.3.	Qualidade Interna MODBUS	61
5.3.	Configuração da Interface Serial	62
5.3.1.	COM 1	62
5.3.2.	Configurações Avançadas	63
5.4.	Configuração da Interface Ethernet	63
5.4.1.	NET 1	64
5.4.2.	Portas TCP/UDP Reservadas	64
5.4.3.	Roteamento de Rede	64
5.5.	Configuração da Interface CAN (Controller Area Network)	65
5.5.1.	CAN	65
5.6.	Configuração das E/S Integradas	65
5.6.1.	Entradas Digitais	67
5.6.2.	Entradas Rápidas	68
5.6.2.1.	Contadores Rápidos	69
5.6.2.1.1.	Interrupções do Contador	74
5.6.2.2.	Interrupção Externa	75
5.6.3.	Saídas Rápidas	76
5.6.3.1.	VFO/PWM	78
5.6.3.2.	PTO	80
5.6.4.	Entradas Analógicas	85
5.6.5.	Entradas RTD	86
5.6.6.	Saídas Analógicas	87
5.6.7.	Mapeamento de E/S	88
5.7.	Protocolos de Comunicação	88
5.7.1.	Comportamento dos Protocolos x Estados da UCP	90
5.7.2.	Pontos Duplos	91
5.7.3.	Fila de Eventos da UCP	91
5.7.3.1.	Consumidores	93
5.7.3.2.	Princípios de Funcionamento da Fila	93
5.7.3.2.1.	Sinalização de Estouro	93
5.7.3.3.	Produtores	93
5.7.4.	Interceptação de Comandos Oriundos do Centro de Controle	93
5.7.5.	MODBUS MESTRE RTU	98
5.7.5.1.	Configuração do Protocolo MODBUS Mestre por Mapeamento Simbólico	98
5.7.5.1.1.	Parâmetros Gerais do Protocolo MODBUS Mestre – Configuração por Mapeamento Simbólico	98
5.7.5.1.2.	Configuração dos Dispositivos – Configuração por Mapeamento Simbólico	100
5.7.5.1.3.	Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico	101
5.7.5.1.4.	Configuração das Requisições – Configuração por Mapeamento Simbólico	103
5.7.6.	MODBUS ESCRAVO RTU	106
5.7.6.1.	Configuração do Protocolo MODBUS Escravo por Mapeamento Simbólico	106

5.7.6.1.1.	Parâmetros Gerais do Protocolo MODBUS Escravo – Configuração por Mapeamento Simbólico	106
5.7.6.1.2.	Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico	109
5.7.7.	MODBUS ETHERNET	110
5.7.8.	MODBUS Ethernet Cliente	112
5.7.8.1.	Configuração do Protocolo MODBUS Ethernet Cliente por Mapeamento Simbólico	112
5.7.8.1.1.	Parâmetros Gerais do Protocolo MODBUS Cliente – Configuração por Mapeamento Simbólico	112
5.7.8.1.2.	Configuração dos Dispositivos – Configuração por Mapeamento Simbólico	113
5.7.8.1.3.	Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico	115
5.7.8.1.4.	Configuração das Requisições – Configuração por Mapeamento Simbólico .	117
5.7.8.2.	Disparo de Relações MODBUS Cliente de Forma Acíclica	120
5.7.9.	MODBUS Ethernet Servidor	120
5.7.9.1.	Configuração do Protocolo MODBUS Ethernet Servidor por Mapeamento Simbólico	120
5.7.9.1.1.	Parâmetros Gerais do Protocolo MODBUS Servidor – Configuração por Mapeamento Simbólico	120
5.7.9.1.2.	Diagnósticos MODBUS Servidor – Configuração por Mapeamento Simbólico	122
5.7.9.1.3.	Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico	123
5.7.10.	OPC DA Servidor	124
5.7.10.1.	Criando um Projeto para Comunicação OPC DA	126
5.7.10.2.	Configurando um CP no Servidor OPC DA	129
5.7.10.2.1.	Importando uma Configuração do Projeto	131
5.7.10.3.	Variáveis de Status e Qualidade da Comunicação OPC DA	131
5.7.10.4.	Limites da Comunicação com Servidor OPC DA	133
5.7.10.5.	Acessando Dados Através de um Cliente OPC DA	133
5.7.11.	OPC UA Servidor	136
5.7.12.	Mestre EtherCAT	137
5.7.12.1.	Instalando e Inserindo Dispositivos EtherCAT	138
5.7.12.1.1.	EtherCAT - Scan Devices	138
5.7.12.2.	Configuração do Mestre EtherCAT	139
5.7.12.2.1.	EtherCAT Master - General	139
5.7.12.2.2.	EtherCAT Master - Sync Unit Assignment	142
5.7.12.2.3.	EtherCAT Master - Overview	142
5.7.12.2.4.	EtherCAT Master - I/O Mapping	142
5.7.12.2.5.	EtherCAT Master - IEC Objects	142
5.7.12.2.6.	EtherCAT Master - Status / Informations	142
5.7.12.3.	Configuração do Escravo EtherCAT	143
5.7.12.3.1.	EtherCAT Slave - General	143
5.7.12.3.2.	EtherCAT Slave - Process Data	145
5.7.12.3.3.	EtherCAT Slave - Edit PDO List	147
5.7.12.3.4.	EtherCAT Slave - Startup Parameters	148
5.7.12.3.5.	EtherCAT Slave - Module I/O Mapping	148
5.7.12.3.6.	EtherCAT Slave - Status and Information	148
5.7.13.	EtherNet/IP	148
5.7.13.1.	Interface EtherNet	150
5.7.13.2.	Configuração do Scanner EtherNet/IP	152
5.7.13.2.1.	Geral	152
5.7.13.2.2.	Conexões	153

5.7.13.2.3.	Assemblies	155
5.7.13.2.4.	EtherNet/IP: Mapeamento de E/S	156
5.7.13.3.	Configuração do Adapter EtherNet/IP	156
5.7.13.3.1.	Geral	156
5.7.13.3.2.	EtherNet/IP Adapter: Mapeamento de E/S	157
5.7.13.4.	Configuração do Módulo EtherNet/IP	157
5.7.13.4.1.	Assemblies	158
5.7.13.4.2.	EtherNet/IP Module: Mapeamento de E/S	158
5.7.14.	IEC 60870-5-104 Servidor	158
5.7.14.1.	Tipos de Dados	158
5.7.14.2.	Pontos Duplos	160
5.7.14.2.1.	Pontos Duplos de Entrada Digital	160
5.7.14.2.2.	Pontos Duplos de Saída Digital	162
5.7.14.3.	Parâmetros Gerais	167
5.7.14.4.	Mapeamento dos Dados	167
5.7.14.5.	Camada de Enlace	169
5.7.14.6.	Camada de Aplicação	171
5.7.14.7.	Diagnósticos do Servidor	173
5.7.14.8.	Qualificador dos Comandos	174
5.7.15.	CANOpen Manager	175
5.7.15.1.	Instalando e inserindo dispositivos CANopen	176
5.7.15.2.	Configuração do CANOpen Manager	176
5.7.15.3.	Configuração de Escravo CANopen	177
5.8.	Modo de E/S Remotas	178
5.8.1.	Escravo CANopen	179
5.8.2.	EtherNet/IP Adapter	180
5.8.3.	Configuração das E/S em Modo Remoto	181
5.8.4.	PROFINET Controller	185
5.9.	Desempenho de Comunicação	185
5.9.1.	Servidor MODBUS	185
5.9.1.1.	Interfaces Integradas da UCP	186
5.9.2.	Servidor OPC UA	187
5.10.	Relógio RTC	187
5.10.1.	Blocos Funcionais e Funções para Leitura e Escrita do RTC	187
5.10.1.1.	Funções de Leitura do RTC	187
5.10.1.1.1.	GetDateAndTime	188
5.10.1.1.2.	GetTimeZone	188
5.10.1.1.3.	GetDayOfWeek	189
5.10.1.2.	Funções de Escrita do RTC	190
5.10.1.2.1.	SetDateAndTime	190
5.10.1.2.2.	SetTimeZone	192
5.10.2.	Estruturas de Dados do RTC	193
5.10.2.1.	EXTENDED_DATE_AND_TIME	193
5.10.2.2.	DAYS_OF_WEEK	193
5.10.2.3.	RTC_STATUS	194
5.10.2.4.	TIMEZONESETTINGS	194
5.10.3.	Limites de Operação do RTC	194
5.11.	Memória de Arquivos de Usuário	195

5.12.	Blocos Funcionais e Funções	196
5.12.1.	Blocos Funcionais Especiais para Comunicação Serial	196
5.12.1.1.	SERIAL_CFG	200
5.12.1.2.	SERIAL_GET_CFG	202
5.12.1.3.	SERIAL_GET_CTRL	203
5.12.1.4.	SERIAL_GET_RX_QUEUE_STATUS	204
5.12.1.5.	SERIAL_PURGE_RX_QUEUE	206
5.12.1.6.	SERIAL_RX	208
5.12.1.7.	SERIAL_RX_EXTENDED	210
5.12.1.8.	SERIAL_SET_CTRL	213
5.12.1.9.	SERIAL_TX	214
5.12.2.	Atualização de Entradas e Saídas	216
5.12.2.1.	RefreshIntegratedIoInputs	216
5.12.2.2.	RefreshIntegratedIoOutputs	216
5.12.3.	Temporizador Retentivo	217
5.12.3.1.	TOF_RET	217
5.12.3.2.	TON_RET	218
5.12.3.3.	TP_RET	219
5.13.	Controle de Movimento (Softmotion)	221
5.14.	Página Web de Usuário	221
5.15.	Página Web de Sistema	221
5.15.1.	Informações Gerais	221
5.15.2.	Gerenciamento	221
5.15.2.1.	Sistema	222
5.15.2.1.1.	Atualização de Firmware	222
5.15.2.1.2.	Ajuste do Relógio	223
5.15.2.1.3.	Controle do Sistema	223
5.15.2.2.	Rede	224
5.15.2.2.1.	Configuração	224
5.15.2.2.2.	Sniffer	226
5.15.2.3.	SNMP	227
5.15.2.3.1.	SNMP nas UCPs Nexto	227
5.15.2.3.2.	Configuração SNMP	228
5.15.2.3.3.	Usuário e Comunidades SNMP	229
5.15.2.4.	Dispositivos USB	229
5.15.2.4.1.	Dispositivos de Armazenamento em Massa	231
5.15.2.4.2.	Conversor USB para RS-232	233
5.15.2.4.3.	Dispositivos Modem	234
5.15.2.4.4.	Adaptadores Wi-Fi	236
5.15.2.4.5.	Adaptadores Ethernet	238
5.15.2.4.6.	Função de Usuário para Controle da Interface USB	241
5.15.2.5.	Firewall	242
5.15.2.5.1.	Configurações Gerais	242
5.15.2.5.2.	Regras de Usuário	244
5.15.2.5.3.	Modo Switch ou NIC Teaming	246
5.15.2.6.	OpenVPN	246
5.15.2.6.1.	Configuração do OpenVPN	247
5.15.2.6.2.	Arquivos de Segurança	250

5.15.2.6.3.	Status	251
5.15.2.6.4.	Arquivos para Baixar	253
5.15.2.6.5.	Configuração de Arquiteturas	253
5.15.2.7.	Servidor FTP	255
5.15.2.7.1.	Configurações Gerais	256
5.15.2.7.2.	Configurações de Usuário	256
5.15.2.7.3.	Status	257
6.	Manutenção	258
6.1.	Diagnósticos	258
6.1.1.	Diagnósticos via LED	258
6.1.2.	Diagnósticos via Página Web de Sistema	259
6.1.3.	Diagnósticos via Variáveis	259
6.1.3.1.	Diagnósticos Resumidos	259
6.1.3.2.	Diagnósticos Detalhados	260
6.1.4.	Diagnósticos via Blocos Funcionais	272
6.1.4.1.	GetTaskInfo	272
6.2.	Manutenção Preventiva	273
7.	Apêndices	274
7.1.	Gerenciamento de Certificados e Chaves TLS	274
7.1.1.	Geração de Certificados por Easy-RSA	274
7.1.2.	Geração de Certificados por OpenSSL	279
7.1.3.	Geração de Chave TA pelo OpenVPN	281

1. Introdução

O Nexto Xpress é um poderoso Controlador Programável (CP), pertencente à família de controladores e módulos de E/S da Série Nexto. O Nexto Xpress fornece poder de processamento de alta velocidade em um design compacto com E/S integradas. Existem várias opções para escolher, permitindo a melhor solução para aplicações básicas.

Esse portfólio de produtos visa sistemas de controle pequenos, oferecendo modelos contendo desde algumas entradas e saídas digitais até opções com 43 pontos de E/S concentrados em um único controlador, incluindo entradas e saídas analógicas com suporte de temperatura (sensores RTD). No caso de necessidades adicionais de E/S, o sistema pode ser facilmente expandido através dos módulos de expansão (consultar seção [Produtos Relacionados](#)). Adicionalmente, a quantidade de pontos de E/S pode ser ampliada ainda mais através dispositivos de E/S remotos (distribuído) comunicando-se através de protocolos como CANopen, EtherNet/IP, PROFINET e MODBUS.

O Nexto Xpress é adequado para pequenas aplicações e E/S distribuídas remotamente. Pode ser utilizado em aplicações verticais como infraestrutura, automação predial, água, efluentes, alimentos, têxteis, automação de fábrica, máquinas e várias outras soluções OEM, incluindo aplicações de controle de movimento (motion). A presença do firewall integrado garante proteção e segurança aos sistemas, mantendo a integridade dos dados e evitando possíveis ameaças cibernéticas. Além disso, o controlador é uma ideal solução para complementar grandes aplicações junto com o portfólio da Série Nexto, ampliando a gama de aplicações usando a mesma tecnologia e ambiente de engenharia. Esta é uma grande vantagem para OEMs e integradores de sistemas com necessidades de aplicações pequenas a grandes.

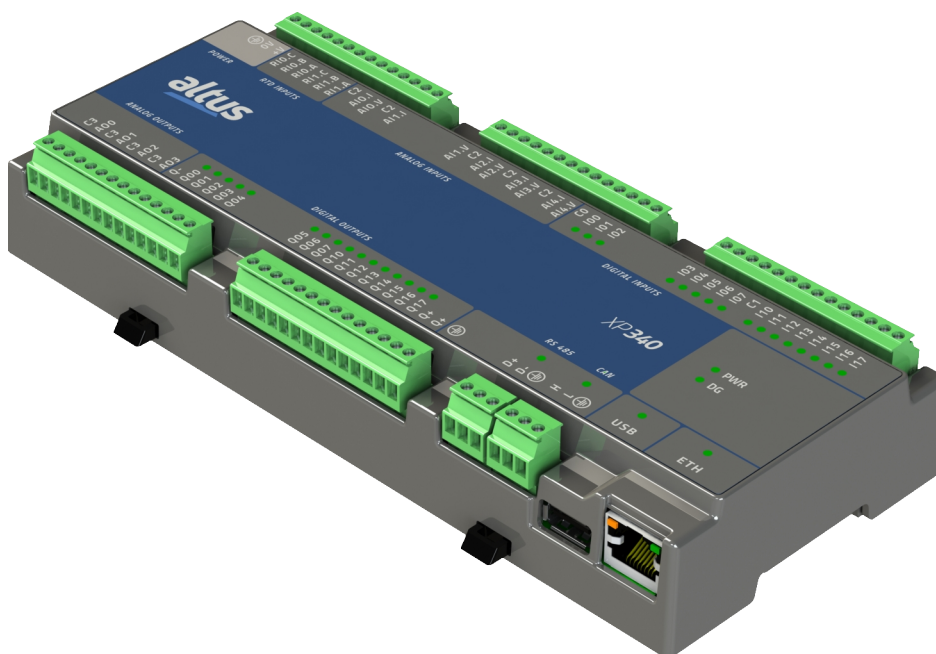


Figura 1: Nexto Xpress

1.1. Documentos Relacionados a este Manual

Este manual irá focar em informações específicas para os controladores da série Nexto Xpress. Para outras funcionalidades idênticas em todos os controladores da Série Nexto, este manual apenas apontará para o manual correspondente da Série Nexto que contém as informações. Esses manuais relacionados estão descritos na tabela a seguir e estão disponíveis em sua última versão no site www.altus.com.br.

Código	Descrição	Idioma
CE114000	Nexto Series – Technical Characteristics	Inglês
CT114000	Série Nexto – Características Técnicas	Português
MU216600	Nexto Xpress User Manual	Inglês
MU216000	Manual de Utilização Nexto Xpress	Português
MU214600	Nexto Series User Manual	Inglês
MU214000	Manual de Utilização Série Nexto	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
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

Tabela 1: Documentos Relacionados

1.2. 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.

1.3. 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.

1.4. 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 as características técnicas dos controladores Nexto Xpress, abordando as partes integrantes do sistema, sua arquitetura, características gerais e elétricas.

2.1. Painéis e Conexões

A figura abaixo mostra o painel frontal do XP325:

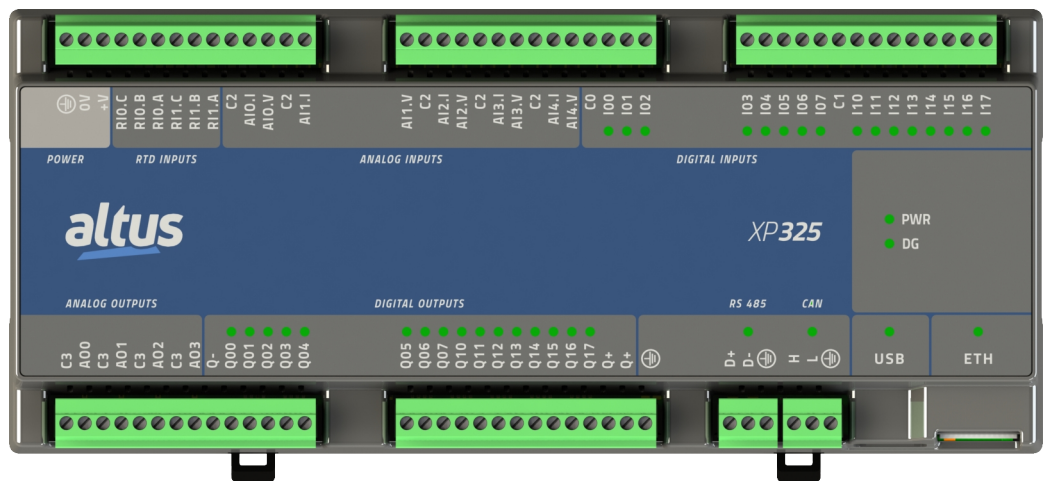


Figura 2: Painel Frontal do XP325

O painel frontal contém a identificação das interfaces de comunicação e E/S disponíveis nos controladores Nexto Xpress. As interfaces de E/S digitais têm um LED para cada ponto para indicar o estado lógico, enquanto as interfaces de comunicação têm um LED cada para indicar atividade. A disponibilidade dessas interfaces em cada modelo é descrita na próxima seção.

Além disso, no lado direito do painel frontal, há 2 LEDs usados para indicar alimentação e diagnósticos. A tabela a seguir mostra a descrição dos LEDs. Para mais informações sobre o estado e significado dos LEDs, consulte o capítulo [Manutenção](#).

LED	Descrição
PWR	Estado da alimentação interna de energia
DG	Indicação de Diagnóstico
Ixx.x	Estado das Entradas Digitais
Qxx.x	Estado das Saídas Digitais
D+/-	Estado da interface RS-485 (pisca quando em atividade)
H/L	Estado da interface CAN (pisca quando em atividade)
USB	Estado da porta USB (liga quando um dispositivo está montado)
ETH	Estado da interface Ethernet (liga quando conectado, pisca quando em atividade)

Tabela 2: Descrição de LEDs

2.2. Características do Produto

2.2.1. Características Gerais

	XP300	XP315	XP325	XP340	XP350	XP351
Entradas Digitais	12					
Entradas Rápidas	4					
Saídas Digitais	12					
Saídas Rápidas	4					
Número máx. de contadores rápidos	1					
Número máx. de interrupções externas	2					
Número máx. de saídas PTO	2					
Número máx. de saídas VFO/PWM	4					
Entradas analógicas V/I (EA)	-	5 até 10 Ver Notas	5 até 10 Ver Notas	5 até 10 Ver Notas	5 até 10 Ver Notas	5 até 10 Ver Notas
Entradas analógicas RTD (EA)	-	2	2	2	2	2
Saídas analógicas V/I (SA)	-	-	4	4	-	-
Interface Ethernet TCP / IP	1					
Interface serial RS-485	1					
Interface CAN	1					
Porta USB Host	1					
Páginas web de usuário (Webvisu)	Não	Não	Não	Sim	Não	Não
Controle de movimento (Softmotion)	Não	Não	Não	Não	Sim, sem CNC	Sim, com CNC
Modo E/S Remota CANopen Slave e EtherNet/IP Adapter	Sim	Sim	Sim	Não	Não	Não
FTP	Sim					
Firewall	Sim					
VPN	Sim					
Número Máximo de Tarefas	16					
Linguagens de programação	Texto Estruturado (ST) Diagrama Ladder (LD) Sequenciamento Gráfico de Funções (SFC) Diagrama de Blocos Funcionais (FBD) Gráfico Contínuo de Funções (CFC)					
Alterações online	Sim					
Cão de guarda	Sim					
Relógio de tempo real (RTC)	Sim Resolução de 1 ms, máx. variação de 3 segundos por dia, tempo de retentividade de 14 dias.					
Indicação de estado e diagnóstico	LEDs, página da web e memória interna da UCP					
Isolação Terra de proteção  para todos Lógica/RS-485/CAN/USB para todos Ethernet para todos Fonte de Alimentação/ E/S Analógi- cas para todos Entradas Digitais para todos	1500 Vdc / 1 minuto (1000 Vac / 1 minuto) 1500 Vdc / 1 minuto (1000 Vac / 1 minuto) 1500 Vdc / 1 minuto (1000 Vac / 1 minuto) 1500 Vdc / 1 minuto (1000 Vac / 1 minuto) 1500 Vdc / 1 minuto (1000 Vac / 1 minuto)					

	XP300	XP315	XP325	XP340	XP350	XP351
Grupo de entradas digitais I0x para I1x	1500 Vdc / 1 minuto (1000 Vac / 1 minuto)					
Saídas digitais para todos	1500 Vdc / 1 minuto (1000 Vac / 1 minuto)					
Dissipação máxima de potência	5 W					
Área máxima do cabeamento	0,5 mm ² (20 AWG) com virola 1,5 mm ² (16 AWG) sem virola					
Classificação mínima da temperatura do fio	75 °C					
Material do fio	Apenas cobre					
Índice de proteção	IP 20					
Revestimento isolante de circuitos eletrônicos	Sim					
Temperatura de operação	-20 a 60 °C					
Temperatura de operação (UL/cUL)	0 a 60 °C					
Temperatura de armazenamento	-25 a 75 °C					
Umidade relativa de operação e armazenamento	5% a 96%, sem condensação					
Resistência à vibração (IEC 60068-2-6, sinus)	7 mm de 5 a 8,4 Hz 2 G de 8,4 a 500 Hz 10 em cada eixo, 1 oitava por minuto					
Resistência ao choque (IEC 60068-2-27, half-sine)	15 G por 11 ms, 6 choques em cada um dos 3 eixos					
Dimensões do produto (L x A x P)	215,5 x 98,8 x 34,0 mm					
Dimensões da embalagem (L x A x P)	270,0 x 102,0 x 40,0 mm					
Peso	370 g					
Peso com embalagem	430 g					

Tabela 3: Características Gerais

Notas:

Entradas analógicas V/I (EA): Por padrão, cada entrada analógica é composta por 2 terminais (AIx.V e AIx.I), e ao selecionar um modo (V, por exemplo), o outro pino (I, por exemplo) fica sem uso. Com a função *AnalogInputProbe*, fornecida pela biblioteca *LibIntegratedIoExt*, é possível utilizar estas entradas livres, permitindo ter até 10 entradas analógicas (5 nos terminais AIx.V e outros 5 nos terminais AIx.I), com as mesmas características técnicas informadas neste documento. Para informações adicionais, consulte o Suporte Técnico.

Controle de movimento (motion): Suporte a blocos de função PLCopen Motion Control Part 1 para comando de eixo único, sincronização de multi eixos, engrenagem eletrônica (CAME), editor especial para planejar movimentos (CAM), entre outros.

Número Máximo de Tarefas: Este valor representa o número máximo de tarefas do usuário e do sistema. A descrição detalhada de possíveis tarefas do usuário pode ser encontrada na seção Perfis do projeto no Manual do usuário. Antes do Mastertool v3.30, esse valor era definido como "5".

Isolação: O termo *Lógica* refere-se às interfaces internas, como processadores, memórias e interfaces de comunicação USB, Serial e CAN.

Revestimento isolante de circuitos eletrônicos: O revestimento isolante protege os componentes eletrônicos no interior do produto contra umidade, poeira e outros elementos agressivos para circuitos eletrônicos.

Temperatura de operação: A temperatura mínima de operação é 0°C para unidades com revisão de produto inferior a AS/AS/AW/AE para XP300/XP315/XP325/XP340, respectivamente.

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 (TAA000034G) (Exceto XP350 e XP351)
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)
	Ordinary Locations: cULus LISTED, E473496 Hazardous Locations: cULus LISTED, E536282 Class I Division II, Groups A, B, C, D
EAC	TR 004/2011 (LVD) CU TR 020/2011 (EMC)

Tabela 4: Normas e Certificações

2.2.3. Memória

	XP300	XP315	XP325	XP340	XP350	XP351
Memória de variáveis de entrada de representação direta (%I)	2 KB					
Memória de variáveis de saída de representação direta (%Q)	2 KB					
Memória de variáveis de representação direta (%M)	1 KB					
Memória de variáveis simbólicas	2 MB	2 MB	2 MB	6 MB	6 MB	6 MB
Memória de programa	3 MB	3 MB	3 MB	8 MB	8 MB	8 MB
Memória total Memória de programa (Máx. definido por modelo) + Memória de código fonte (backup) + Memória de arquivos de Webvisu	64 Mbytes					
Memória retentiva/persistente (configurável pelo usuário)	7,5 KB (Expansível até 64 KB através do uso de Receitas armazenadas na memória de arquivos de usuário (ver artigo na base de conhecimento))					
Memória de arquivos do usuário (backup)	8 MB					

Tabela 5: Memória

Notas:

Memória de programa: A partir da versão 3.40 do Mastertool, a memória foi ampliada de 2MB para 3MB nos modelos XP300, XP315 e XP325 e de 6MB para 8MB no modelo XP340.

Memória de variáveis simbólicas Retentivas e Persistentes: Área de memória onde são alocadas as variáveis simbólicas retentivas / persistentes. O controlador realiza a gravação dos dados retentivos/persistentes de forma cíclica com periodicidade de 5 segundos em tempo de execução.

ATENÇÃO

A declaração e o uso de variáveis persistentes simbólicas devem ser executados exclusivamente por meio do objeto *Persistent Vars*, que pode ser incluído no projeto por meio da visualização em árvore em *Aplicação -> Adicionar Objeto -> Variáveis Persistentes*. Não deve ser utilizada a expressão *VAR PERSISTENT* na declaração de variáveis de campo de POU's.

O comportamento detalhado da memória de variáveis simbólicas retentivas/persistentes pode ser encontrada na tabela a seguir, onde "X" significa manutenção (salvamento) do valor de memória no cenário apresentado, enquanto "-" significa sua perda.

Comando	Variável Simbólica	Variável Retentiva	Variável Persistente
Ciclo de energização	-	X	X
Reset a quente	-	X	X
Reset a frio	-	-	X
Reset origem	-	-	-
Download	-	-	X
Alteração online	X	X	X
Limpar tudo	-	-	X
Reset Process (IEC 60870-5-104)	-	X	X
Runtime/Watchdog Reset	-	-	-

Tabela 6: Comportamento de variável após comando

2.2.4. Protocolos

		Interface
Protocolo aberto	✓	COM 1 / USB
MODBUS RTU Mestre	✓	COM 1
MODBUS RTU Escravo	✓	COM 1
MODBUS TCP Cliente	✓	NET 1
MODBUS TCP Servidor	✓	NET 1
MODBUS RTU via TCP Cliente	✓	NET 1
MODBUS RTU via TCP Servidor	✓	NET 1
CANopen Mestre	✓	CAN
CANopen Escravo	✓ (exceto XP340, XP350 e XP351)	CAN
CAN low level	✓	CAN
SAE J-1939	✓	CAN
OPC DA Servidor	✓	NET 1 / USB
OPC UA Servidor	✓	NET 1 / USB
EtherCAT Mestre	✓	NET 1
SNMP Agente	✓	NET 1 / USB
IEC 60870-5-104 Servidor	✓ (apenas XP340)	NET 1
EtherNet/IP Scanner	✓	NET 1
EtherNet/IP Adapter	✓	NET 1
MQTT Cliente	✓	NET 1 / USB
SparkPlugB	✓	NET 1 / USB
SNTP Cliente (para sincronismo do relógio)	✓	NET 1 / USB
PROFINET Controller	✓	NET 1
PROFINET Device	✗	-
OpenVPN Client	✓	NET 1 / USB
OpenVPN Server	✓	NET 1 / USB
FTP Server	✓	NET 1 / USB

Tabela 7: Protocolos

Notas:

USB: Necessário o uso de Conversor Serial, Adaptador Wi-Fi, Modem 3G/4G ou Adaptador Ethernet.

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

2.2.5. RS-485

	RS-485
Conector	Bloco de terminação de 3 pinos
Interface Física	RS-485
Direção de Comunicação	RS-485: half duplex
Máx. Transmissores RS-485	32
Terminação	Sim (Configurável)
Taxa de Transmissão	2400, 4800, 9600, 19200, 38400, 57600, 115200 bps

Tabela 8: Característica da Interface Serial RS-485

2.2.6. CAN

	CAN
Conector	Bloco de terminação de 3 pinos
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)
Taxa de Transmissão	10, 20, 50, 100, 125, 250, 500, 800, 1000 kbit/s

Tabela 9: Característica da Interface CAN

2.2.7. USB

	USB
Conector	USB A Fêmea
Interface Física	USB V2.0
Taxa de Transmissão	1.5 Mbps (Baixa Velocidade), 12 Mbps (Velocidade Nominal) e 480 Mbps (Velocidade Máxima)
Corrente Máxima	500 mA
Dispositivos Suportados	Dispositivo de armazenamento em massa Conversor serial USB RS-232 Modem USB 3G/4G Adaptador USB Wi-Fi Adaptador USB Ethernet

Tabela 10: 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](#).

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

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. Ethernet

	Ethernet
Conector	RJ45 fêmea blindado
Auto crossover	Sim
Máximo Comprimento de Cabo	100 m
Tipo de Cabo	UTP ou ScTP, categoria 5
Taxa de Transmissão	10/100 Mbps
Camada Física	10/100 BASE-TX
Camada de Enlace	LLC
Camada de Rede	IP
Camada de Transporte	TCP (Protocolo de Controle de Transmissão) UDP (Protocolo de Datagrama de Usuário)
Diagnósticos	LED (Link/Atividade)

Tabela 11: Característica da Interface Ethernet

2.2.9. Alimentação

	Alimentação
Tensão de Entrada Nominal	24 Vdc
Tensão de Entrada	19,2 a 30 Vdc
Máxima Corrente de Entrada (in-rush)	50A / 300 us
Máxima Corrente de Entrada	300 mA

Tabela 12: Característica da Alimentação

2.2.10. Entradas Digitais

	Entradas Digitais
Tipo de Entrada	Ponto optoisolado tipo 1 Dois grupos isolados de 8 entradas cada
Tensão de Entrada	24 Vdc 15 a 30 Vdc para nível lógico 1 0 a 5 Vdc para nível lógico 0
Impedância de Entrada	4,95 k Ω
Máxima Corrente de Entrada	6,2 mA @ 30 Vdc
Indicação do estado da Entrada	Sim
Tempo de Resposta	0,1 ms
Filtro de Entrada	Desabilitado ou 2 ms a 255 ms – por software

Tabela 13: Característica das Entradas Digitais

Nota:

Filtro de Entrada: A amostragem do filtro é realizada na MainTask (ou função de atualização), então é recomendado usar valores múltiplos do intervalo da tarefa.

2.2.11. Entradas Rápidas

	Entradas Rápidas
Número de entradas rápidas	4 (podem ser usadas como contador rápido, interrupção externa ou entrada digital convencional)
Número max. de contadores rápidos	1
Número max. de interrupções externas	2
Configuração dos conectores	I00, I01, I02 e I03
Tensão de entrada	24 Vdc 15 a 30 Vdc para nível lógico 1 0 a 5 Vdc para nível lógico 0
Impedância de entrada	1,85 k Ω
Máxima corrente de entrada	16,2 mA @ 30 Vdc
Modo de configuração	Modos de 1 entrada: Entrada digital convencional Interrupção externa Modos de 2 entradas: Contador Up/Down (A conta, B sentido) com zeramento (usa I00, I01, I02) Quadratura 2x (usa I00, I01) Quadratura 2x com zeramento (usa I00, I01, I02) Quadratura 4x (usa I00, I01) Quadratura 4x com zeramento (usa I00, I01, I02)
Controle do sentido de contagem	Apenas por hardware
Borda de detecção da entrada de contagem	Subida, ativa em nível lógico 1 (exceto para quadratura 4 x, onde conta nas duas bordas)
Formato dos dados	Inteiros de 32 bit com sinal

	Entradas Rápidas
Limite de operação	De - 2.147.483.648 até 2.147.483.647
Frequência máxima de entrada	100 kHz
Largura de pulso mínima @ 24 Vdc	2 μ s

Tabela 14: Característica das Entradas Rápidas

2.2.12. Saídas Digitais

	Saídas Digitais
Tipo de Saída	Ponto transistorizado optoisolado
Corrente Máxima de Saída	1,5 A por saída 12 A total
Corrente de fuga	35 μ A
Resistência de saída	105 m Ω
Fonte de alimentação externa	19,2 a 30 Vdc
Tempo de comutação	20 μ s - transição desligado para ligado @ 24 Vdc 500 μ s - transição ligado para desligado @ 24 Vdc
Frequência máxima de comutação	250 Hz
Parâmetros configuráveis	Sim
Indicação do estado de saída	Sim
Proteções de saída	Sim, proteção contra surtos de tensão

Tabela 15: Característica das Saídas Digitais

Nota:

Tempo de comutação: O tempo necessário para se desligar uma saída depende da carga empregada.

2.2.13. Saídas Rápidas

	Saídas Rápidas	
Número de saídas	4 (podem ser usadas como: VFO/PWM, PTO ou saída digital convencional)	
Número Máx. de Saídas PTO	2	
Número Máx. de Saídas VFO/PWM	4 se não usar PTO 2 ao usar 1 PTO 0 ao usar 2 PTO	
Configuração dos Conectores	Q14, Q15, Q16 e Q17	
Corrente Máxima	0 a 500 Hz: 1,5A por saída / 6,0A total 500 a 200 KHz: 0,5A por saída / 2,0A total	
Tipo de Saída	Saída Transistorizada	
Máxima Frequência de Geração de Pulsos	200 kHz @ 60 mA	
Mínima Largura de Pulso @ 24 Vdc	CARGA MÍNIMA 400 Ω	MÍNIMO TEMPO DE PULSO 320 ns
Indicação de estado	Através de operandos reservados estáticos	
Proteções	Todas saídas rápidas possuem Diodo TVS	

	Saídas Rápidas	
Tensão de Operação	19,2 a 30 Vdc	
Impedância de Saída	700 mΩ	
Modos de Saída	Saída digital convencional VFO/PWM PTO (somente Q14 e Q16. Saídas adjacentes são forçadas como saídas digital convencional.)	
Funções Executadas por Software	PTO	VFO/PWM
	Escrita do número de pulsos a serem gerados Escrita do número de pulsos de aceleração e desaceleração Início / fim de operação das saídas Diagnósticos de saídas rápidas Monitoração do estado atual das saídas rápidas	Escrita do valor de frequência a ser gerado (1 Hz a 200 kHz). Escrita do duty cycle das saídas (1% a 100%) Início / fim de operação das saídas Diagnósticos de saídas rápidas

Tabela 16: Característica das Saídas Rápidas

2.2.14. Entradas Analógicas

	Entradas Analógicas
Tipo de Entrada	Entrada de tensão ou corrente, terminação única, configurada individualmente
Formato dos dados	16 bits em complemento de dois, justificado à esquerda
Resolução do conversor	Monotonia de 12 bits garantida, sem códigos perdidos
Tempo de conversão	400 μs (todos os canais V/I e RTD habilitados)
Indicação do estado da Entrada	Sim
Proteções de módulo	Sim, proteção contra surtos de tensão e inversão de polaridade

Tabela 17: Característica das Entradas Analógicas

Faixas de entrada	Entrada Modo Tensão		
	Faixa	Escala de Engenharia	Resolução
	0 a 10 Vdc	0 a 30.000	2,5 mV
Precisão	$\pm 0,3 \%$ do fundo de escala @ 25 °C $\pm 0,010 \%$ do fundo de escala / °C		
Sobre escala	3 % do fundo de escala		
Tensão máxima de entrada	12 Vdc		
Impedância de entrada	21 k Ω		
Parâmetros configuráveis	Tipo de sinal por entrada Filtros		
Constante de tempo do filtro passa baixa	100 ms, 1 s, 10 s ou desabilitado		

Tabela 18: Característica das Entradas Analógicas Modo Tensão

Faixas de entrada	Entrada Modo Corrente		
	Faixa	Escala de Engenharia	Resolução
	0 a 20 mA	0 a 30.000	5,12 μ A
	4 a 20 mA	0 a 30.000	5,12 μ A
Precisão	$\pm 0,3 \%$ do fundo de escala @ 25 °C $\pm 0,015 \%$ do fundo de escala / °C		
Sobre escala	3 % do fundo de escala		
Corrente máxima de entrada	30 mA		
Impedância de entrada	119 Ω		
Parâmetros configuráveis	Tipo de sinal por entrada Filtros Canal aberto		
Constante de tempo do filtro passa baixa	100 ms, 1 s, 10 s ou desabilitado		

Tabela 19: Característica das Entradas Analógicas Modo Corrente

Nota:

Faixas de entrada: Quando configurado como 4 a 20 mA, os sinais de entrada inferiores a 4 mA resultarão em valores negativos (-7.500 para 0 mA). A partir da versão 3.16 do Mastertool, um novo parâmetro chamado *Valor de Circuito Aberto* foi incluído para selecionar o comportamento nesta situação. O valor padrão é *Desabilitado* (que fornece uma leitura linear como descrito acima), tendo também a opção de fornecer uma leitura fixa igual aos limites inferior e superior (0 ou 30.000).

	Entrada RTD
Precisão	$\pm 0,5$ % do fundo de escala @ 25 °C
Escala suportadas	Pt100, Pt1000, 0 a 400 Ω , 0 a 4000 Ω
Corrente de excitação	1 mA
Faixa de resistência (escala)	0 a 400 Ω (utilizado para PT100) 0 a 4000 Ω (utilizado para PT1000)
Sobre escala	5 % do fundo de escala
Parâmetros configuráveis	Tipo de sinal por entrada Filtros
Constante de tempo do filtro passa baixa	100 ms, 1 s, 10 s ou desabilitado
Impedância máxima do cabo do sensor (por cabo)	5 Ω

Tabela 20: Característica das Entradas Analógicas Modo RTD

2.2.15. Saídas Analógicas

	Saídas Analógicas
Tipo de saída	Saída de tensão ou corrente, configurada individualmente
Formato dos dados	16 bits em complemento de dois, justificado à esquerda
Resolução do conversor	Monotonia de 12 bits garantida, sem códigos perdidos
Tempo de conversão	450 μ s (todas as saídas habilitadas)
Indicação do estado da saída	Sim
Proteções de módulo	Sim, proteção contra surtos de tensão e inversão de polaridade

Tabela 21: Característica das Saídas Analógicas

	Saída Modo Tensão		
Faixas de saída	Faixa	Escala de Engenharia	Resolução
	0 a 10 V	0 a 30.000	2,5 mV
Precisão	$\pm 0,3$ % do fundo de escala @ 25 °C $\pm 0,025$ % do fundo de escala / °C		
Tempo de estabilização	4 ms		
Valor máximo de saída	+ 10,3 Vdc		
Impedância de carga	> 1 k Ω		
Parâmetros configuráveis	Tipo de sinal por saída		

Tabela 22: Características das Saídas Analógicas Modo Tensão

Faixas de saída	Saída Modo Corrente		
	Faixa	Escala de Engenharia	Resolução
	0 a 20 mA 4 a 20 mA	0 a 30.000 0 a 30.000	5,18 μ A 5,18 μ A
Precisão	$\pm 0,3$ % do fundo de escala @ 25 °C $\pm 0,020$ % do fundo de escala / °C		
Tempo de estabilização	4 ms		
Valor máximo de saída	+ 20,6 mA		
Impedância de carga	< 600 Ω		
Parâmetros configuráveis	Tipo de Sinal por Saída		

Tabela 23: Característica das Saídas Analógicas Modo Corrente

Nota:

Faixas de Saída: Quando configurada como 4 a 20 mA, a saída pode ser configurada para valores inferiores a 4 mA, atribuindo valores negativos à variável de saída (-7.500 para 0 mA).

2.3. Compatibilidade com Outros Produtos

Para desenvolver uma aplicação para controladores Nexto Xpress, 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:

Modelo do controlador	Mastertool	Versão de Firmware
XP300, XP315 e XP325	3.10 ou superior	1.7.0.0 ou superior
XP340	3.18 ou superior	1.8.0.0 ou superior
XP350	3.50 ou superior	1.12.5.0 ou superior
XP351	3.52 ou superior	1.12.29.0 ou superior

Tabela 24: 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 do controlador Nexto Xpress depende de:

- Tempo de Intervalo da Aplicação
- Tempo da Aplicação do Usuário
- Tempo do Sistema Operacional
- Número de canais habilitados de E/S integrada

2.4.1. Tempo de Intervalo

A aplicação e a atualização de E/S são executadas em uma tarefa cíclica (periódica) chamada MainTask. O tempo de intervalo desta tarefa pode ser configurado de 1 ms a 100 ms. O tempo gasto para essas operações é chamado de Tempo de Ciclo e sempre deve ser menor que o intervalo, porque o tempo livre é usado para comunicação e outras tarefas de baixa prioridade do controlador.

Além disso, as E/S integradas podem ser atualizadas de forma assíncrona em qualquer ponto do código de aplicação do usuário, usando as funções de atualização disponíveis na seção [Atualização de Entradas e Saídas](#).

2.4.2. Tempos da Aplicação

O tempo de execução da aplicação (tempo de ciclo) depende das seguintes variáveis:

- Tempo de leitura de entradas integradas
- Tempo de execução de tarefa
- Tempo de escrita de saídas integradas

O tempo necessário para ler e escrever na E/S integrada depende do número e do tipo de canais de E/S habilitados. Para E/S digitais, todos os canais estão sempre ativados e o tempo adicionado à MainTask não é relevante. Para E/S analógicas, o tempo adicionado à MainTask é determinado pelo tempo de conversão (para entradas analógicas) e pelo tempo de atualização (para saídas analógicas), ambos descritos na tabela Características Gerais.

2.4.3. Tempo para a Execução de Instruções

A tabela abaixo apresenta o tempo necessário para a execução de diferentes instruções nas UCPs Nexto Xpress.

Instrução	Linguagem	Variáveis	Tempos da Instrução (μ s)
1000 Contatos	LD	BOOL	10
1000 Divisões	ST	INT	180
		REAL	40
	LD	INT	180
		REAL	40
1000 Multiplicações	ST	INT	13
		REAL	13
	LD	INT	13
		REAL	13
1000 Somas	ST	INT	13
		REAL	13
	LD	INT	13
		REAL	13

Tabela 25: Tempos de Instrução

2.4.4. Tempo de Inicialização

O tempo de inicialização do controlador Nexto Xpress é de aproximadamente 40 s.

2.5. Dimensões Físicas

Dimensões em mm.

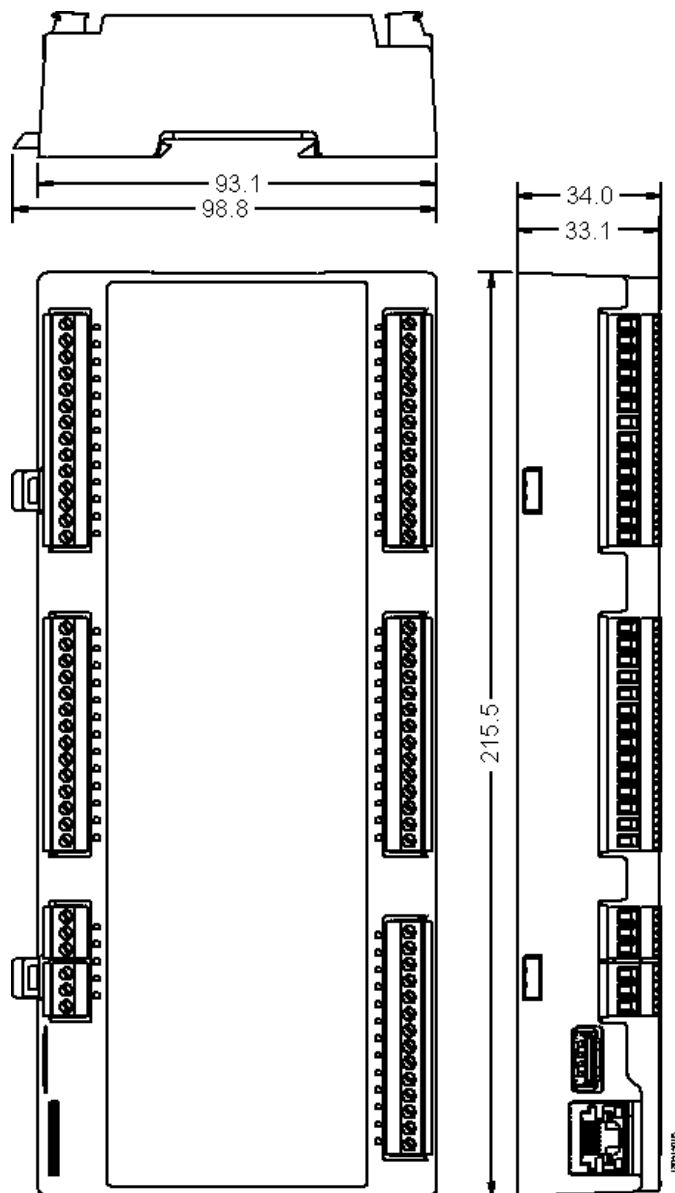


Figura 3: Dimensões Físicas do XP3xx

2.6. Dados de Compra

2.6.1. Itens Integrantes

A embalagem do produto possui os seguintes itens:

- Módulo CP Compacto
- Conectores

2.6.2. Código do Produto

Os seguintes códigos devem ser usados para compra do produto:

Código	Descrição
XP300	CP Compacto de alta-velocidade com 16 ED, 16 SD à Transistor, 1 porta Ethernet, 1 porta serial RS-485 e Mestre CANopen
XP315	CP Compacto de alta-velocidade com 16 ED, 16 SD à Transistor, 5 EA V/I, 2 EA RTD (3 fios), 1 porta Ethernet, 1 porta serial RS-485 e Mestre CANopen
XP325	CP Compacto de alta-velocidade com 16 ED, 16 SD à Transistor, 5 EA V/I, 2 EA RTD (3 fios), 4 SA, 1 porta Ethernet, 1 porta serial RS-485 e Mestre CANopen
XP340	CP compacto de alta velocidade com 16 ED, 16 SD à Transistor, 5 EA V/I, 2 EA RTD (3 fios), 4 SA, 1 porta Ethernet, 1 porta serial RS-485, Mestre CANopen e suporte a páginas web de usuário
XP350	CP compacto de alta velocidade com Softmotion Padrão, 16 ED, 16 SD à transistor, 5 EA V/I, 2 EA RTD (3 fios), 1 porta Ethernet, 1 canal serial RS-485 e Mestre CANopen
XP351	CP compacto de alta velocidade com Softmotion Avançado (CNC), 16 ED, 16 SD à transistor, 5 EA V/I, 2 EA RTD (3 fios), 1 porta Ethernet, 1 canal serial RS-485 e Mestre CANopen

Tabela 26: Modelos de Controladores Nexto Xpress

2.7. Produtos Relacionados

Os seguintes produtos devem ser adquiridos separadamente quando necessário:

Código	Descrição
MT8500	MasterTool IEC XE
MT9000	Mastertool X
NX9202	Cabo RJ45-RJ45 2 m
NX9205	Cabo RJ45-RJ45 5 m
NX9210	Cabo RJ45-RJ45 10 m
AL-2600	Derivador e terminador de rede RS-485
AL-2306	Cabo RS-485 p/ rede MODBUS ou CAN
AL-1766	Cabo CFDB9-borneira
FBS-USB-232M9	Cabo conversor universal USB-Serial / 2m
XP900	Adaptador USB TP-Link nano Wireless 150 Mbps TL-WN725N (disponível apenas no Brasil)
AMJG0808	Cabo simples RJ45-RJ45 2 m
XP101	Expansão Nexto Xpress, 16 ED 24 Vdc
XP106	Expansão Nexto Xpress, 8 ED 24 Vdc e 6 SD Relé
XP201	Expansão Nexto Xpress, 16 SD à Transistor
TLE3-21100	Gateway IoT Industrial

Tabela 27: Produtos Relacionados

Notas:

NX92xx: Cabo para a programação das UCPs da Série Nexto e Ethernet ponto-a-ponto com outro dispositivo com interface Ethernet.

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

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

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

FBS-USB-232M9: Cabo para uso como um conversor USB-serial na interface USB dos controladores Xpress.

AMJG0808: Cabo para a programação das UCPs.

XP101 / XP106 / XP201: Módulos de expansão CANopen.

3. Instalação

Este capítulo apresenta os procedimentos necessários para a instalação física dos controladores Nexto Xpress, bem como os cuidados que devem ser tomados com outra instalação dentro do painel onde o controlador está instalado.

CUIDADO

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

ATENÇÃO

Para aplicações marítimas, além das instruções padrão descritas neste capítulo, os seguintes requisitos de instalação devem ser atendidos:

- O produto deve ser instalado em um painel metálico.
- A porta de alimentação de 24 Vdc deve ser equipada com filtro TDK-Lambda modelo RSMN-2003 ou equivalente.
- Os cabos de todas as portas (alimentação, E/S e comunicação) devem ser equipados com um par de ferrites de baixa/alta frequência Wurth Electronics 74272221/74271221 ou equivalente.

ATENÇÃO

Além das instruções padrão descritas neste capítulo, os seguintes requisitos de instalação devem ser atendidos:

- Esses dispositivos são do tipo *open-type* e devem ser instalados em um compartimento adequado, e acessíveis somente com o uso de uma ferramenta ou chave.
- Este equipamento é adequado para uso na Classe I, Divisão 2, Grupos A, B, C e D ou somente em locais classificados como não perigosos.

PERIGO

PERIGO DE EXPLOÇÃO - Não desconecte o equipamento a menos que a alimentação tenha sido removida ou a área seja classificada como não perigosa.

3.1. Instalação Mecânica

Os controladores Nexto Xpress foram projetados para serem instalados em um trilho DIN padrão. Além disso, o usuário deve fornecer um painel adequado que atenda aos requisitos de proteção e segurança do sistema. As próximas seções mostram os procedimentos para instalar e remover o controlador.

CUIDADO

Para obter a especificação de temperatura do controlador, a instalação deve fornecer um espaço livre ao redor do dispositivo, conforme descrito na seção Projeto do Painel do Manual de Utilização da Série Nexto Código MU214000.

3.1.1. Instalando o Controlador

Para instalar o controlador no trilho DIN, primeiro configure as duas travas na posição aberta conforme indicado na figura abaixo:

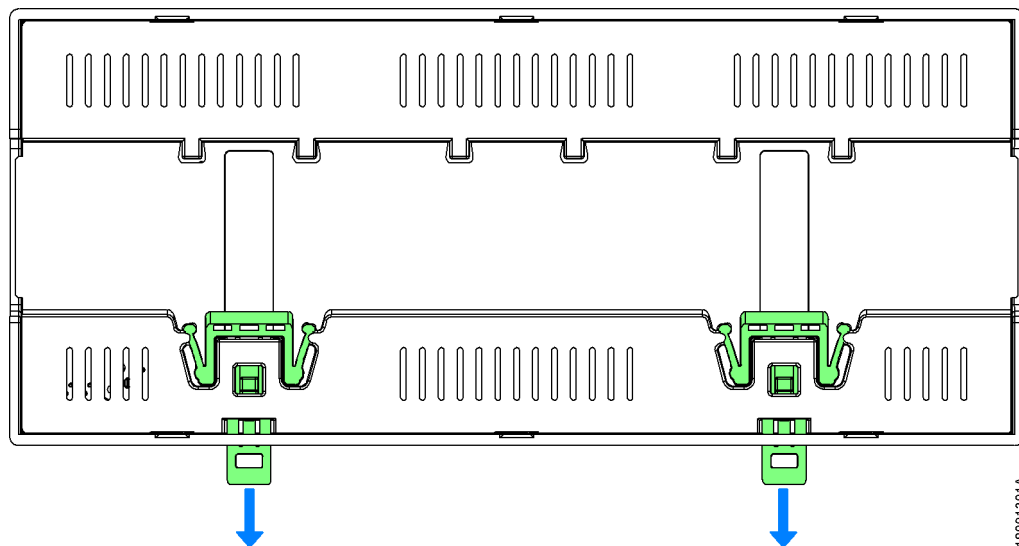


Figura 4: Movendo as travas para a posição aberta

Em seguida, coloque o controlador no trilho DIN, encaixando o lado superior primeiro e depois o lado inferior, conforme indicado nas etapas 1 e 2 da figura abaixo:

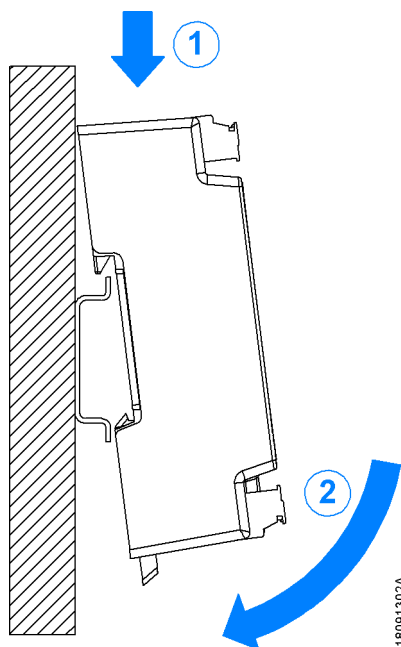


Figura 5: Posicionando o Controlador no Trilho DIN

3. INSTALAÇÃO

Finalmente, mova as duas travas para a posição fechada para travar o controlador no trilho DIN, conforme mostrado na figura abaixo:

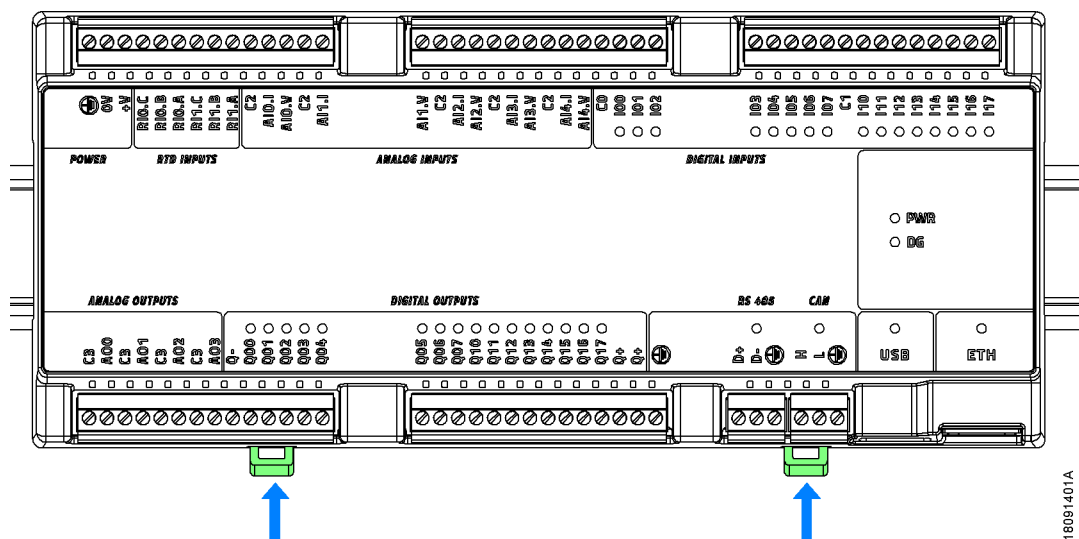


Figura 6: Travando o Controlador no Trilho DIN

3.1.2. Removendo o Controlador

Para remover o controlador do trilho DIN, basta mover as duas travas para a posição aberta, conforme mostrado na figura abaixo:

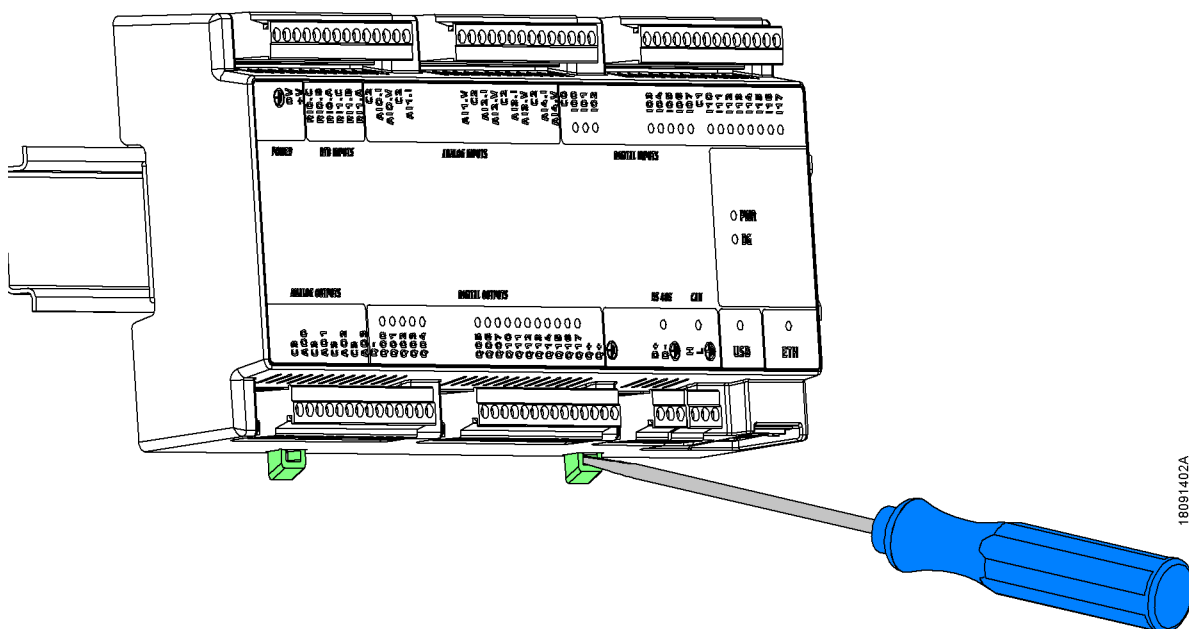


Figura 7: Destravando o Controlador no Trilho DIN

3.2. Instalação Elétrica

PERIGO

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

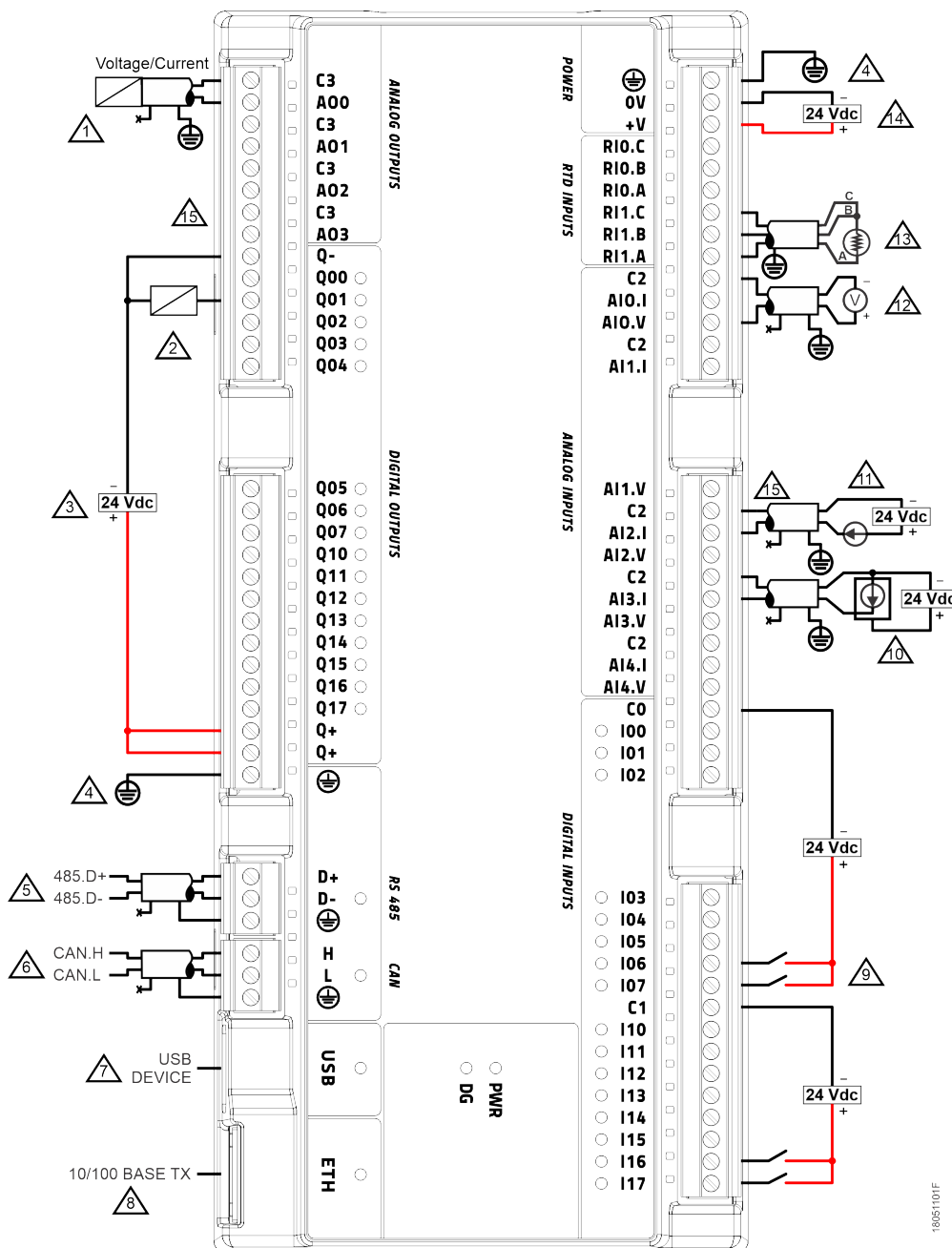


Figura 8: Diagrama de Instalação Elétrica do XP3xx

Notas do Diagrama:

-  Conexão típica da saída analógica no modo de tensão / corrente.
-  Conexão típica de saída digital (tipo fonte).
-  Fonte de alimentação externa para alimentação das saídas Q00 à Q17, os bornes Q+ devem ser conectados ao +24 Vdc, e borne Q- deve ser conectado ao 0 Vdc.
-  Terminais de aterramento de proteção para a fonte de alimentação e portas de comunicação. Ambos devem estar conectados externamente ao terra.
-  Conexão típica da interface serial RS-485.
-  Conexão típica da interface CAN.
-  Verifique a tabela de características técnicas da porta USB para obter a lista de dispositivos suportados.
-  Use cabos Ethernet informados na seção Produtos Relacionados.
-  Conexão típica de entrada digital (tipo sink). C0 e C1 são os pontos comuns para os grupos isolados I0x e I1x respectivamente.
-  Conexão típica da entrada analógica de corrente (dispositivo de campo com alimentação fornecida separadamente do sinal analógico).
-  Conexão típica da entrada analógica de corrente (dispositivo de campo com alimentação fornecida com o sinal analógico, 2 fios).
-  Conexão típica de entrada analógica de tensão.
-  Conexão típica de entrada analógica RTD (3 fios).
-  Conexão da fonte de alimentação externa.
-  Os sinais das entradas e saídas analógicas não são isolados da alimentação principal, logo os sinais C2 e C3 não podem ter diferença de potencial em relação ao 0V da fonte de alimentação principal. É recomendada a conexão do 0V da fonte de alimentação principal com as referências analógicas C2 e C3 antes da conexão com o Xpress.
-  Terminal terra de proteção.

3.3. Conexão de Rede Ethernet

A interface de comunicação ETH, identificada como NET 1 no Mastertool, permite a conexão com uma rede Ethernet e a programação com esta ferramenta.

A conexão de rede Ethernet utiliza cabos de par trançado (10/100Base-TX) e a detecção de velocidade é feita automaticamente pelo controlador Nexto Xpress. Este cabo deve ter uma de suas extremidades conectadas à interface e outra ao HUB, switch, microcomputador ou outro ponto de rede Ethernet.

3.3.1. Endereço IP

A interface Ethernet vem com a seguinte configuração de parâmetros padrão:

	NET 1
Endereço IP	192.168.15.1
Máscara de sub-rede	255.255.255.0
Endereço do Gateway	192.168.15.253

Tabela 28: Configuração de Parâmetros Padrão para a Interface Ethernet NET 1

Primeiro, a interface NET 1 deve estar conectada a uma rede de PC com a mesma máscara de sub-rede para se comunicar com o Mastertool, onde os parâmetros de rede podem ser modificados. Para mais informações sobre configuração e modificações de parâmetros, veja a seção [Configuração da Interface Ethernet](#).

3.3.2. ARP Gratuito

A interface Ethernet NET1 envia prontamente pacotes do tipo ARP na transmissão, informando seu endereço IP e MAC para todos os dispositivos conectados à rede. Esses pacotes são enviados durante o download de uma nova aplicação pelo software Mastertool e na inicialização do controlador quando a aplicação entra no modo Run.

Cinco comandos ARP são acionados com um intervalo inicial de 200 ms, duplicando o intervalo a cada novo comando acionado, totalizando 3 s. Exemplo: primeiro disparo ocorre no tempo 0, o segundo em 200 ms e o terceiro em 600 ms e assim por diante até o quinto disparo no tempo 3 s.

3.3.3. Instalação do Cabo de Rede

A porta Ethernet do Nexto Xpress possui pinagem padrão que é a mesma usada em PCs. O tipo de conector, o tipo de cabo, o nível físico, entre outros detalhes, são definidos na tabela de Características Gerais. Abaixo está a descrição do conector fêmea RJ-45, com a identificação e descrição da pinagem válida para os níveis físicos 10Base-T e 100Base-TX.

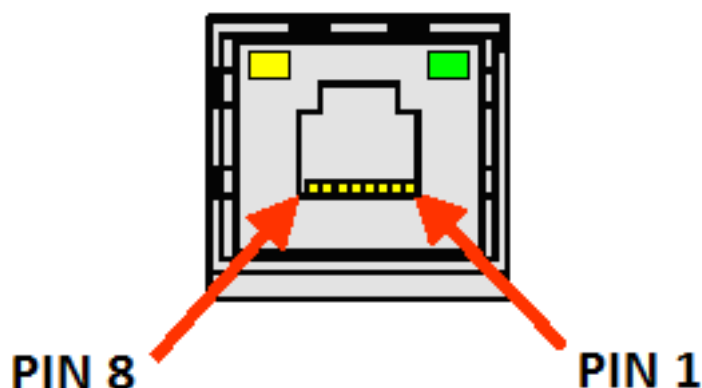


Figura 9: Conector Fêmea RJ45

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 29: Pinagem do Conector Fêmea RJ45

A interface pode ser conectada em uma rede de comunicação através de um hub ou switch, ou diretamente no equipamento de comunicação. Neste último caso, devido ao recurso Auto Crossover, não há necessidade de um cabo de rede cross-over, aquele usado para conectar dois PCs ponto-a-ponto via porta Ethernet.

É importante ressaltar que se entende por cabo de rede um par de conectores macho RJ45 conectados por um cabo UTP ou ScTP, categoria 5, seja em linha reta ou cruzada. É usado para comunicar dois dispositivos através da porta Ethernet.

Esses cabos normalmente possuem uma trava de conexão que garante uma conexão perfeita entre o conector fêmea da interface e o conector macho do cabo. No momento da instalação, o conector macho deve ser inserido no conector fêmea do módulo até ouvir um clique, garantindo a ação da trava. Para desconectar o cabo do módulo, a alavanca de trava deve ser usada para destravar um do outro.

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 UCP](#) e [Configuração da Interface Serial](#) para CAN e RS-485, respectivamente).

4. Programação Inicial

O principal objetivo deste capítulo é ajudar na programação e configuração dos controladores Nexto Xpress, possibilitando ao usuário tomar os primeiros passos antes de começar a programar o dispositivo.

Assim como para os demais dispositivos da Série Nexto, a programação dos controladores Nexto Xpress é feita através da interface de desenvolvimento do Mastertool (IDE), que oferece um completo sistema de programação IEC 61131-3 com todas as linguagens definidas por este padrão (ST, LD, SFC, FBD, etc.) e uma adicional, o CFC. Essas linguagens podem ser usadas simultaneamente no mesmo projeto, permitindo que o usuário use os melhores recursos de cada linguagem, resultando em desenvolvimento de aplicações mais eficiente, para fácil documentação e manutenção futura.

Para mais informações sobre a programação, consulte o Manual de Utilização Mastertool, Manual de Programação do Mastertool 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), os controladores Nexto Xpress são baseados 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).

Além disso, o acesso à memória deve ser feito com cuidado, pois as variáveis com maior número de bits (WORD, DWORD, LONG), usam como índice o byte mais significativo, ou seja, o %QD4 sempre terá como byte mais significativo o %QB4. Portanto, não será necessário fazer cálculos para descobrir quais DWORD correspondem aos bytes definidos. A Tabela 30 mostra as organizações little-endian e big-endian.

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 30: 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		
%QX2.6									
%QX2.5									
%QX2.4									
%QX2.3									
%QX2.2									
%QX2.1									
%QX2.0									
%QX3.7	%QB 03	02			%QB03	%QW	01		
%QX3.6									
%QX3.5									
%QX3.4									
%QX3.3									
%QX3.2									
%QX3.1									
%QX3.0									
%QX4.7	%QB 04	%QW		00	%QB04	%QW	02		
%QX4.6									
%QX4.5									
%QX4.4									
%QX4.3									
%QX4.2									
%QX4.1									
%QX4.0									
%QX5.7	%QB 05	04			%QB05	%QW		03	
%QX5.6									
%QX5.5									
%QX5.4									
%QX5.3									
%QX5.2									
%QX5.1									
%QX5.0									
%QX6.7	%QB 06	%QW	04		%QB06	%QW			04
%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 31: Organização e Acesso à Memória

4.2. Perfis de Projeto

Um perfil de projeto consiste em um modelo de aplicação que define a estrutura de execução do programa do usuário, incluindo a organização de tarefas e regras associadas. Esse modelo tem como objetivo simplificar o desenvolvimento e padronizar a aplicação.

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

- **MT8500:** utiliza o perfil de *Máquina*, com regras predefinidas de nomenclatura e estrutura.
- **MT9000:** utiliza o perfil *Fast*, que mantém o modelo baseado em tarefas, porém sem restrições de nomenclatura, oferecendo maior flexibilidade ao desenvolvedor.

O perfil é selecionado durante a criação do projeto e define a estrutura inicial da aplicação. Durante a compilação, o ambiente valida a consistência da configuração conforme o perfil adotado.

4.2.1. Perfil de Máquina

No perfil de *Máquina*, por padrão, a aplicação possui uma tarefa do tipo cíclica denominada *MainTask*. Esta tarefa é responsável pela execução de uma única unidade de programação do tipo *Program*, denominada *MainPrg*.

Esse programa pode chamar outras unidades de programação do tipo *Program*, *Function* ou *Function Block*, porém todo o código do usuário é executado exclusivamente pela tarefa *MainTask*.

Este perfil se caracteriza por permitir intervalos menores na tarefa *MainTask*, possibilitando uma execução mais rápida do código do usuário.

Adicionalmente, o perfil pode incluir uma tarefa de interrupção de tempo, denominada *TimeInterruptTask00*, com prioridade superior à *MainTask*, podendo interromper sua execução a qualquer momento.

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

Tabela 32: Tarefas no Perfil de Máquina

Além disso, esse perfil suporta a inclusão de tarefas adicionais associadas às interrupções externas e de contador, resultando em um máximo de 5 tarefas para a aplicação do usuário.

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.

4.2.2. Perfil Fast

No perfil *Fast*, a aplicação também é estruturada com base em tarefas responsáveis pela execução do programa do usuário.

Por padrão, é criada uma tarefa do tipo cíclica associada a uma unidade de programação do tipo *Program*. Essa unidade pode chamar outras POUs do tipo *Program*, *Function* ou *Function Block*.

Diferentemente do *Perfil de Máquina*, o perfil *Fast* não impõe restrições de nomenclatura para tarefas ou POUs. O desenvolvedor tem liberdade para definir nomes conforme as necessidades do projeto.

Além disso, o perfil permite maior flexibilidade na configuração da aplicação, incluindo definição de múltiplas tarefas, prioridades e intervalos de execução, conforme os requisitos do sistema.

4.3. Configurando a UCP

A configuração da UCP do controlador está localizada na árvore de dispositivos, conforme mostrado na figura abaixo, e pode ser acessada com um clique duplo no objeto correspondente. Nesta aba é possível configurar o comportamento do cão de guarda, sincronismo do clock, entre outros parâmetros, conforme descrito na seção [Configuração da UCP](#).

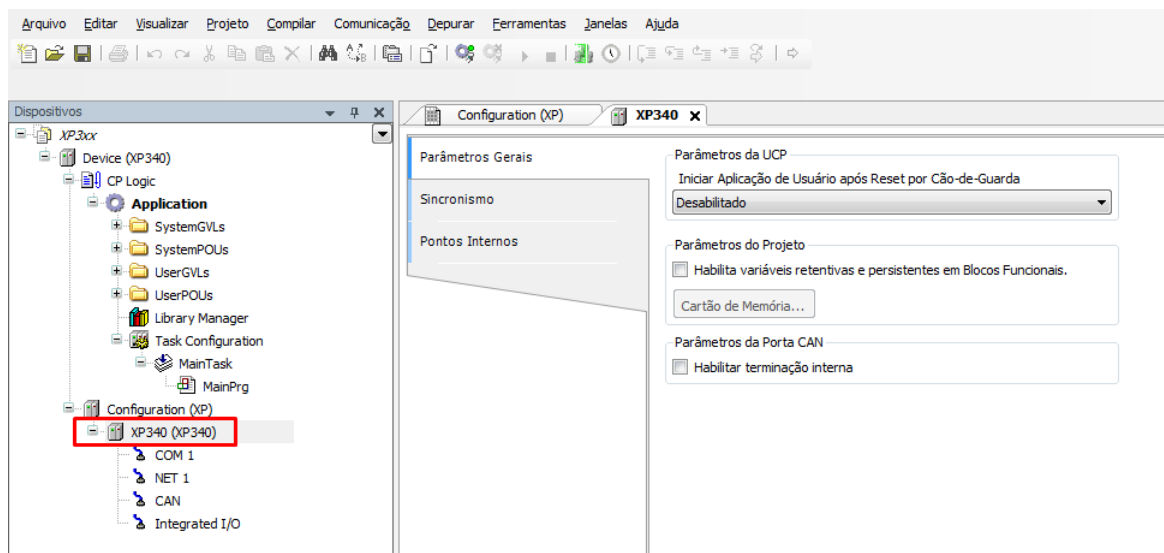


Figura 10: Configuração da UCP

Além disso, ao clicar duas vezes no ícone NET 1 do controlador, é possível configurar a interface Ethernet que será utilizada para comunicação entre o controlador e o software Mastertool.

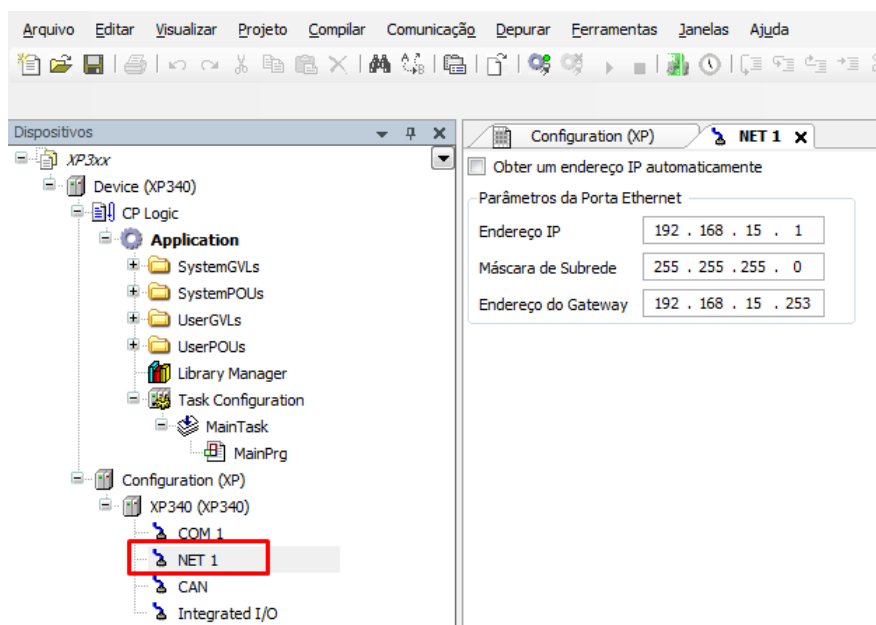


Figura 11: Configurando a Porta de Comunicação

A configuração definida nesta guia será aplicada ao dispositivo somente ao enviar a aplicação para o dispositivo (download), que é descrito mais adiante nas seções [Localizando o Dispositivo](#) e [Login](#).

Além disso, a árvore de dispositivos também oferece a configuração das E/S integradas disponíveis nos controladores Nexto Xpress, conforme mostrado na figura abaixo. Nesta aba é possível configurar filtros de entradas digitais, o modo de cada entrada analógica, entre outros parâmetros.

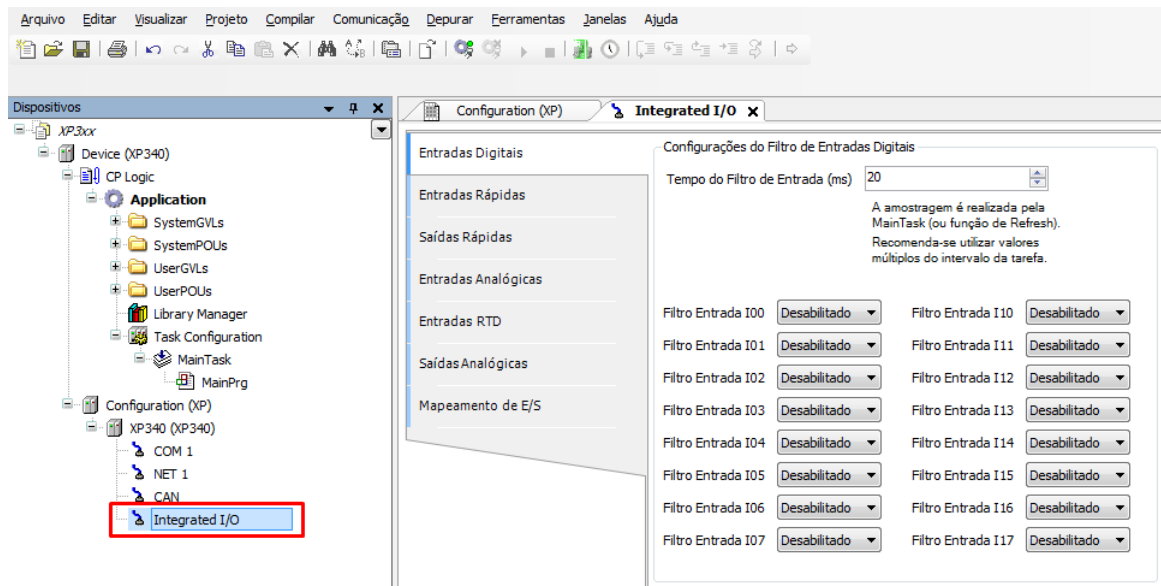


Figura 12: Configurando as E/S Integradas

4.4. Bibliotecas

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

4.5. Inserindo uma Instância de Protocolo

Os controladores Nexto Xpress, conforme o capítulo Características Gerais, oferecem diversos protocolos de comunicação. Exceto as comunicações OPC, que possuem um procedimento de configuração diferente, para inserir um protocolo basta clicar com o botão direito sobre a interface de comunicação desejada, adicionar o dispositivo e finalmente configurar a instância do protocolo conforme descrito na seção [Protocolos de Comunicação](#). A seguir são apresentados alguns exemplos.

4.5.1. MODBUS Ethernet

O primeiro passo para configurar o MODBUS Ethernet (Servidor neste exemplo) é incluir a instância na NET desejada (neste caso, NET 1, já que o XP3xx possui apenas uma interface Ethernet). Clique na NET com o botão direito do mouse e selecione Acrescentar Dispositivo..., como mostrado na Figura 13:

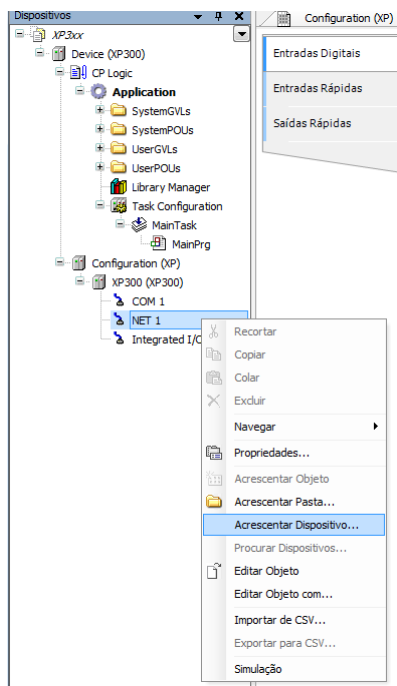


Figura 13: Adicionando a Instância

Depois disso, a lista de protocolos aparecerá na tela. Basta selecionar o MODBUS Symbol Server conforme descrito na figura abaixo:

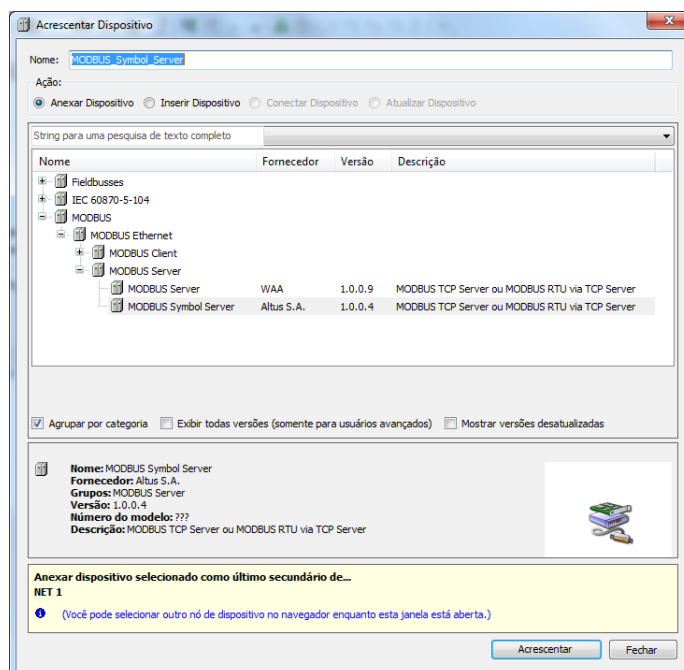


Figura 14: Selecionando o Protocolo

4.6. Localizando o Dispositivo

Para estabelecer a comunicação do controlador 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 os controladores encontrados na rede da interface Ethernet da estação onde está sendo executada a ferramenta.

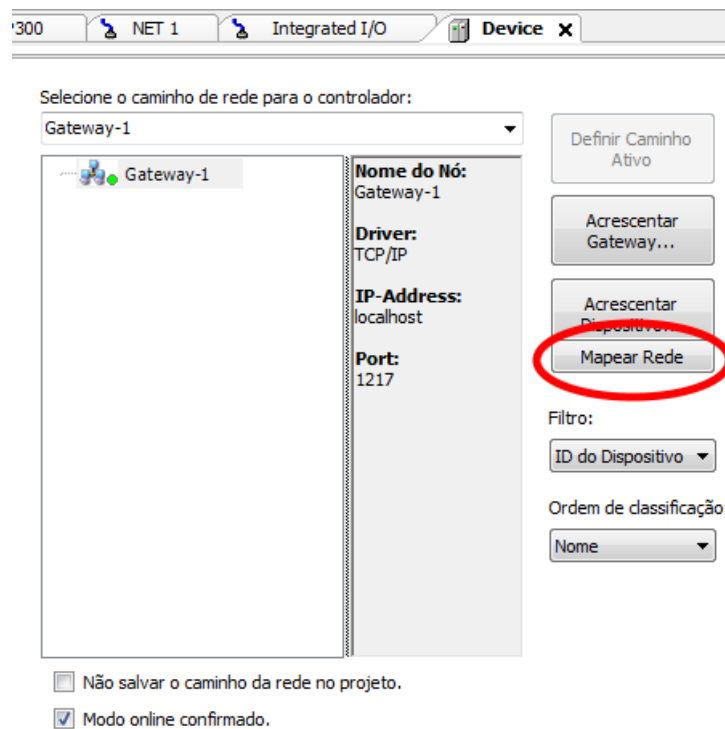


Figura 15: Localizando o Dispositivo

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

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

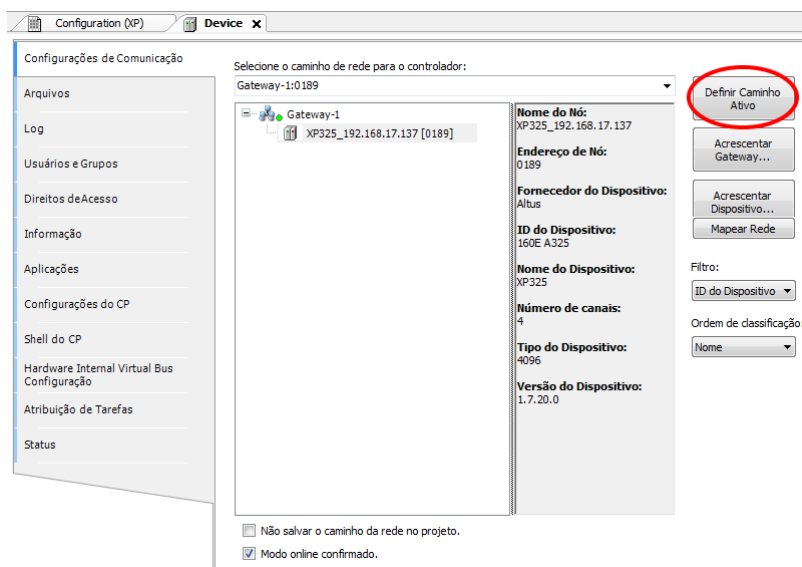


Figura 16: Selecionando o Controlador

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 do controlador que se deseja 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.

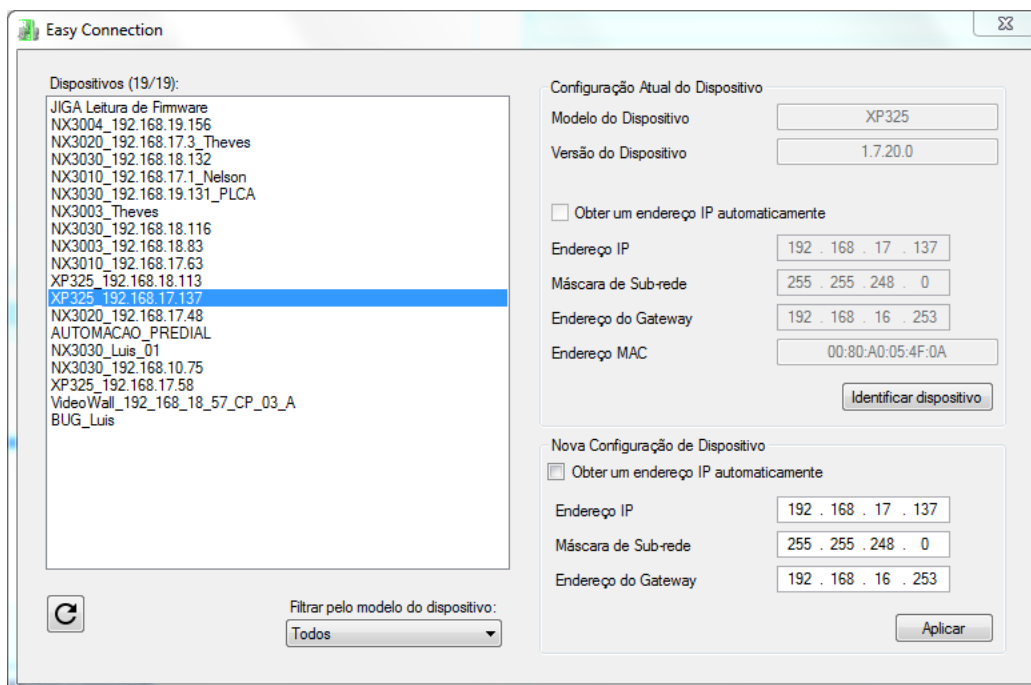


Figura 17: Easy Connection

Este comando realiza uma comunicação em nível MAC com o dispositivo, permitindo alterar permanentemente a configuração da interface Ethernet do controlador, 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 o controlador. 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.

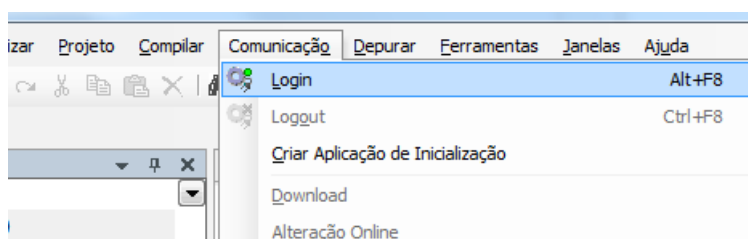


Figura 18: Enviando o Projeto para o controlador

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:

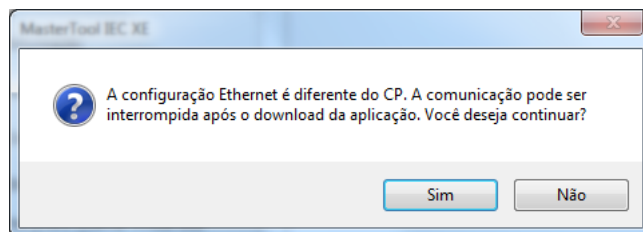


Figura 19: Aviso de configuração IP

Se não houver aplicação rodando no controlador, a seguinte mensagem será apresentada:

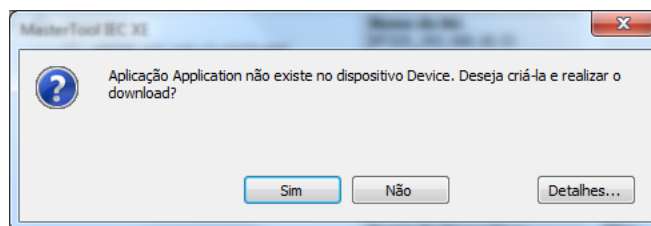


Figura 20: Sem aplicação no dispositivo

Caso já exista uma aplicação no controlador, 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 do controlador 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 o controlador parado (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.

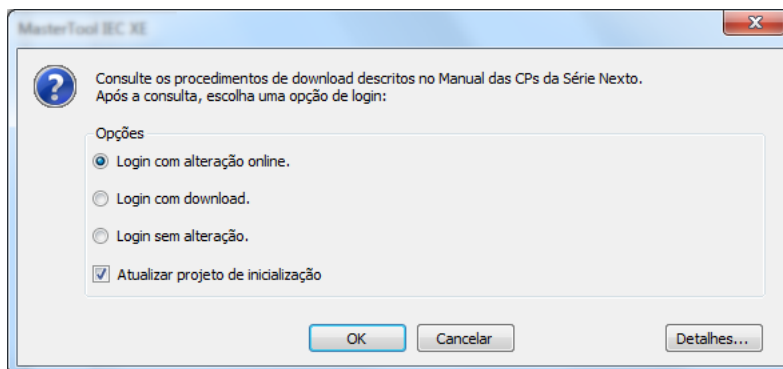


Figura 21: Download de Nova Aplicação

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.

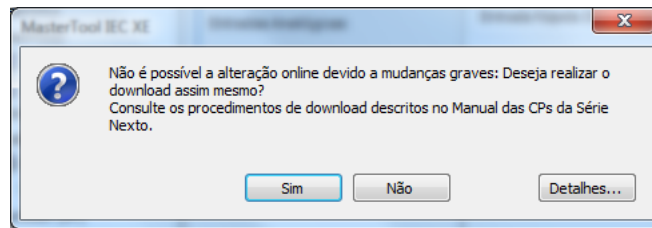


Figura 22: Alteração de Configuração

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:

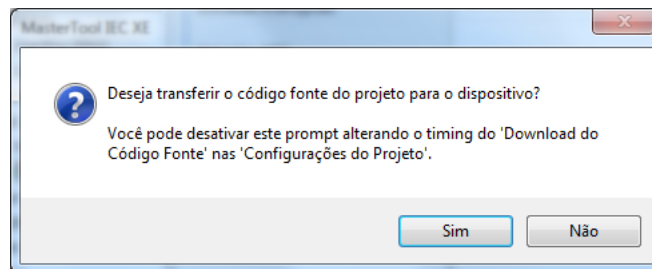


Figura 23: Download de código fonte

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 o controlador, 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 o controlador, permitindo que valores iniciais sejam pré-configurados para que sejam utilizados no controlador no primeiro ciclo.

Para executar este comando, basta acessar o menu Depurar e clicar em *Iniciar*, conforme mostra a Figura 24.

4. PROGRAMAÇÃO INICIAL

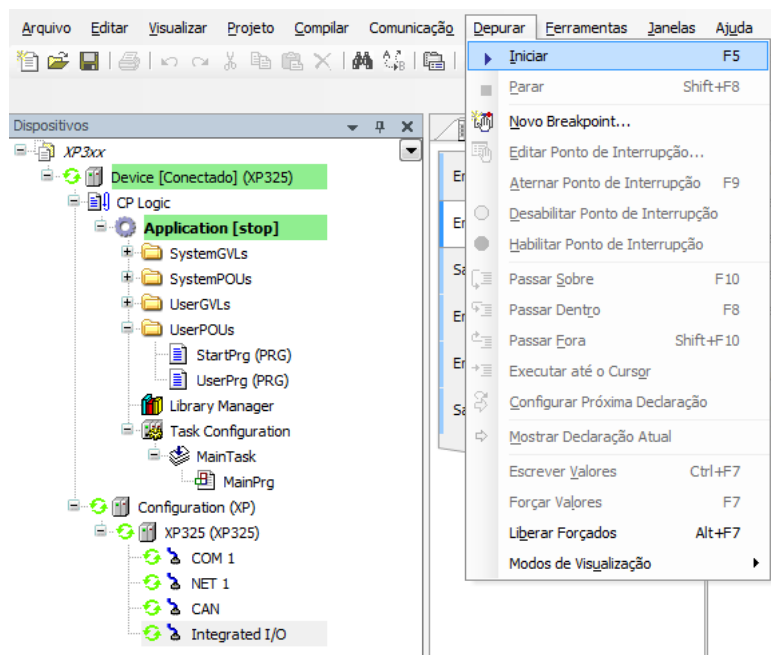


Figura 24: Iniciando a Aplicação

A Figura 25 mostra a aplicação em execução. Caso seja seleccionada 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.

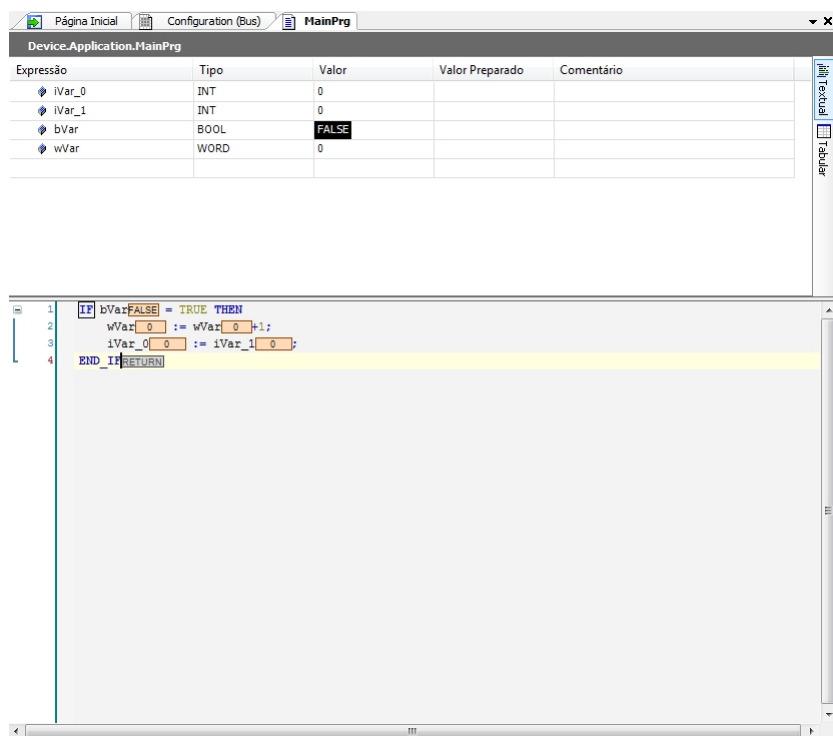


Figura 25: Programa em Execução

Caso o controlador já tenha uma aplicação 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 26.

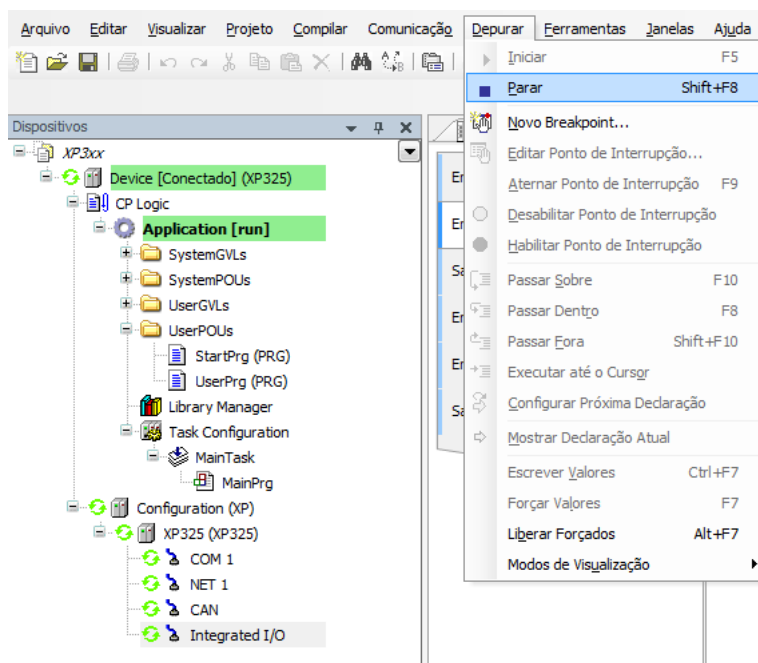


Figura 26: Parando a Aplicação

Caso o controlador seja inicializado sem aplicação gravada, ele 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 o controlador, deve ser utilizado o comando *Logout*, localizado no menu *Comunicação*, conforme mostra a Figura 27.

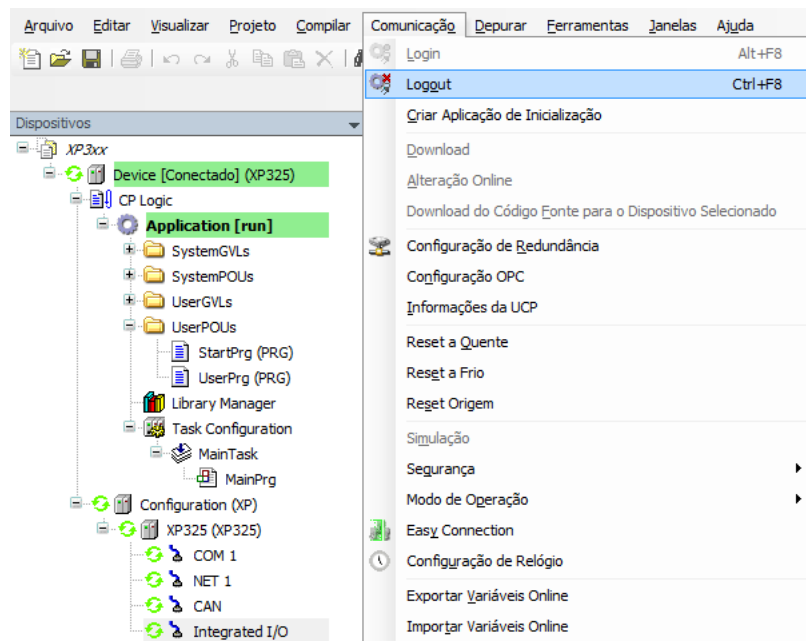


Figura 27: Encerrando a comunicação online com o controlador

4.12. Upload do Projeto

Os controladores Nexto possibilitam o armazenamento do código fonte do projeto na memória interna do dispositivo, permitindo a futura recuperação completa do projeto (*upload*) e a realização de modificações na aplicação.

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

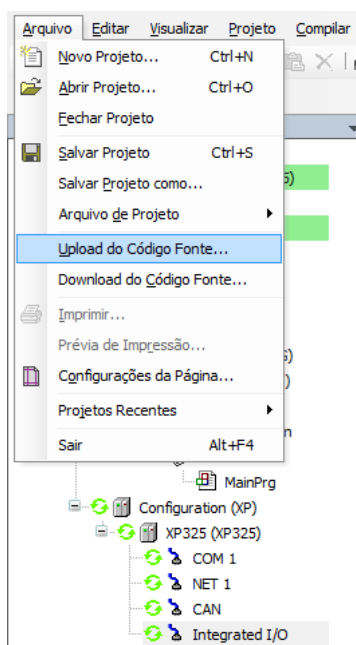


Figura 28: Opção de Upload de Projeto

Após, basta selecionar o controlador desejado e clicar em **OK**, conforme a Figura 29.

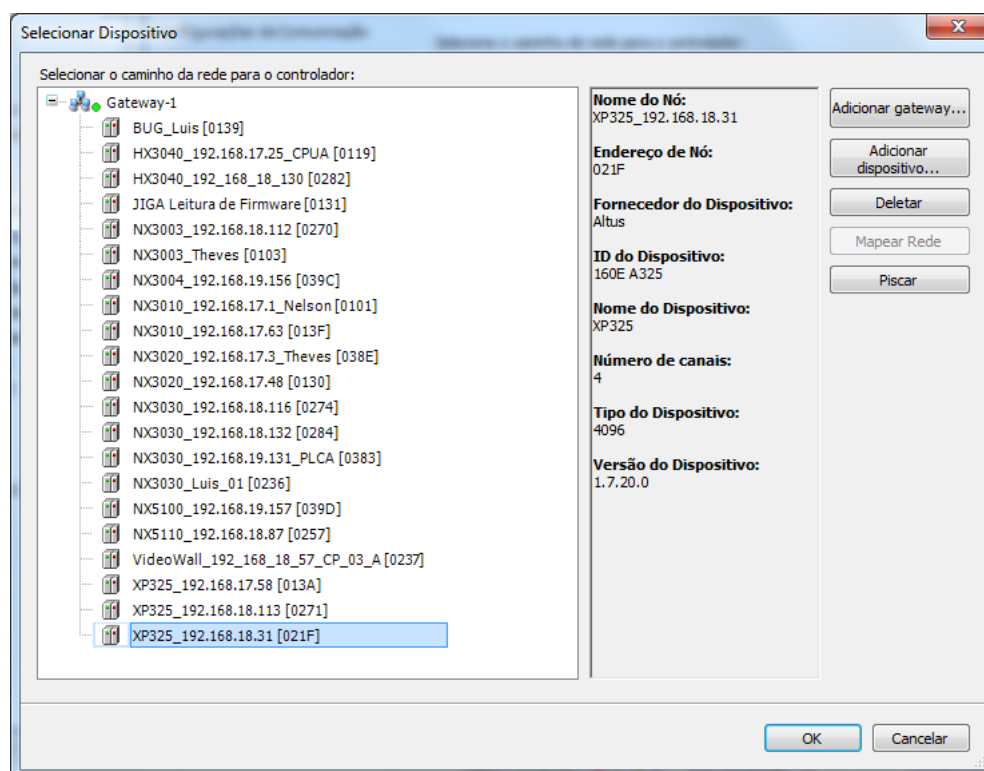


Figura 29: Selecionando o controlador

Para garantir que o projeto carregado do controlador 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 nos controladores Nexto Xpress está definido na tabela de [Características Gerais](#).

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 o controlador 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. Os valores 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, e apaga a aplicação da UCP.

Notas:

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

Comando: Para executar um comando de *Reset a Quente*, *Reset a Frio* ou *Reset Origem*, é necessário estar com o Master-tool em modo *Online* no controlador.

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.

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

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 Wizard, 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 POUs 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 IntegratedIO

A GVL *IntegratedIO* contém as variáveis correspondentes aos canais físicos integrados de entrada e saída.

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

Device.Application.IntegratedIO				
Expression	Type	Value	Prepared value	Address
[-] DigitalInputs	T_DIGITAL_INPUTS_1			
I00	BOOL	FALSE		
I01	BOOL	FALSE		
I02	BOOL	FALSE		
I03	BOOL	FALSE		
I04	BOOL	FALSE		
I05	BOOL	FALSE		
I06	BOOL	FALSE		
I07	BOOL	FALSE		
I10	BOOL	FALSE		
I11	BOOL	FALSE		
I12	BOOL	FALSE		
I13	BOOL	FALSE		
I14	BOOL	FALSE		
I15	BOOL	FALSE		
I16	BOOL	FALSE		
I17	BOOL	FALSE		
[-] DigitalOutputs	T_DIGITAL_OUTPUT...			
Q00	BOOL	FALSE		
Q01	BOOL	FALSE		
Q02	BOOL	FALSE		
Q03	BOOL	FALSE		
Q04	BOOL	FALSE		
Q05	BOOL	FALSE		
Q06	BOOL	FALSE		
Q07	BOOL	FALSE		
Q10	BOOL	FALSE		
Q11	BOOL	FALSE		
Q12	BOOL	FALSE		
Q13	BOOL	FALSE		
Q14	BOOL	FALSE		
Q15	BOOL	FALSE		
Q16	BOOL	FALSE		
Q17	BOOL	FALSE		
[+] FastInputs	T_FAST_INPUTS_1			
[+] FastOutputs	T_FAST_OUTPUTS_1			
[-] AnalogInputs	T_ANALOG_INPUTS_1			
A10	INT	0		
A11	INT	0		
A12	INT	0		
A13	INT	0		
A14	INT	0		
[-] AnalogOutputs	T_ANALOG_OUTPUT...			
A00	INT	0		
A01	INT	0		
A02	INT	0		
A03	INT	0		
[-] RtdInputs	T_RTD_INPUTS_1			
R10	INT	0		
R11	INT	0		

Figura 30: GVL IntegratedIO quando em modo Online

4.14.5. GVL System_Diagnostics

Na GVL *System_Diagnostics* estão presentes as variáveis de diagnóstico da UCP do controlador, das interfaces de comunicação e E/S. 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.

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

Device.Application.System_Diagnostics				
Expression	Type	Value	Prepared value	Address
DIG_XP325	T_DIAG_...			
tSummarized	T_DIAG_...			
bHardwareFailure	BIT	FALSE		
bSoftwareException	BIT	FALSE		
bCOM1ConfigError	BIT	FALSE		
bNET1ConfigError	BIT	FALSE		
bInvalidDateTime	BIT	FALSE		
bRuntimeReset	BIT	FALSE		
bRetentivityError	BIT	FALSE		
bIntegratedIODiagnostic	BIT	FALSE		
tDetailed	T_DIAG_...			
Target	T_DIAG_...			
Hardware	T_DIAG_...			
Exception	T_DIAG_...			
RetainInfo	T_DIAG_...			
Reset	T_DIAG_...			
Serial	T_DIAG_...			
CAN	T_DIAG_...			
USB	T_DIAG_...			
Ethernet	T_DIAG_...			
UserFiles	T_DIAG_...			
UserLogs	T_DIAG_...			
Application	T_DIAG_...			
ApplicationInfo	T_DIAG_...			
SNTP	T_DIAG_...			
IntegratedIO	T_DIAG_I...			
AnalogInputs	T_DIAG_...			
AnalogOutputs	T_DIAG_...			
RTDInputs	T_DIAG_...			

Figura 31: GVL *System_Diagnostics* em Modo Online

4.14.6. 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 Disable em Modo Online

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

Para os dispositivos de comunicação MODBUS, as variáveis de qualidade se comportam da maneira indicada na Tabela 41.

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.

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

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 33: GVL Qualities em Modo Online

4.14.8. 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 Diagnóstico* 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 Master 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 34: GVL ReqDiagnostics em Modo Online

5. Configuração

Os controladores Nexto Xpress são configurados e programados através do software Mastertool. A configuração define o comportamento e o uso de periféricos e recursos especiais do controlador. 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.

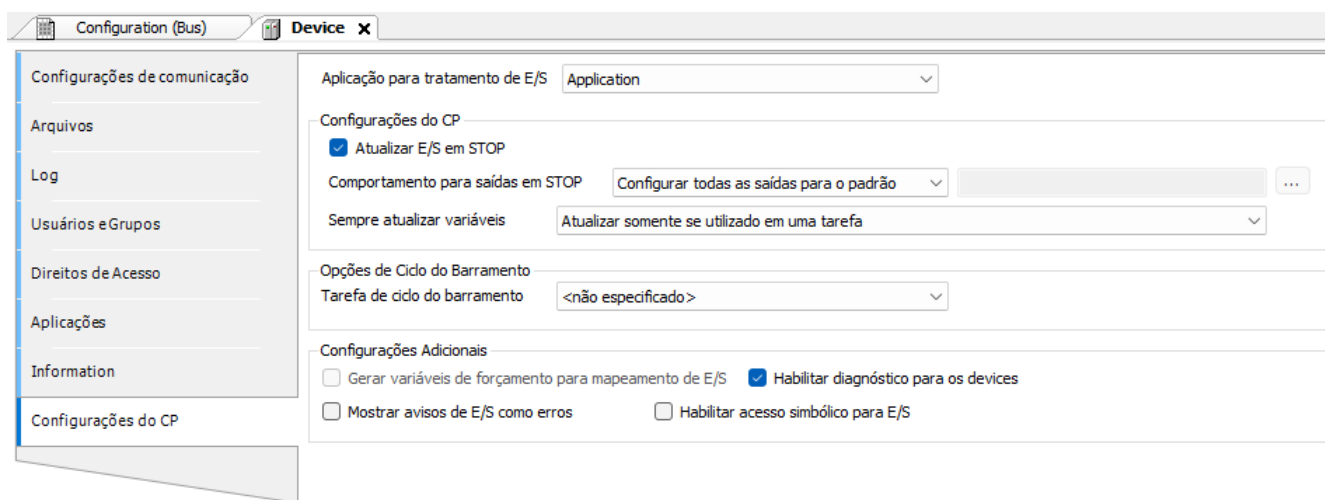


Figura 35: Configurações do CP

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 força-mento 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 33: 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 relativos à UCP do controlador estão localizados na visualização em árvore sob o item XP3xx, logo abaixo de *Configuration*. Cada item deve ser propriamente verificado para a correta execução do projeto. Estes parâmetros estão descritos abaixo:

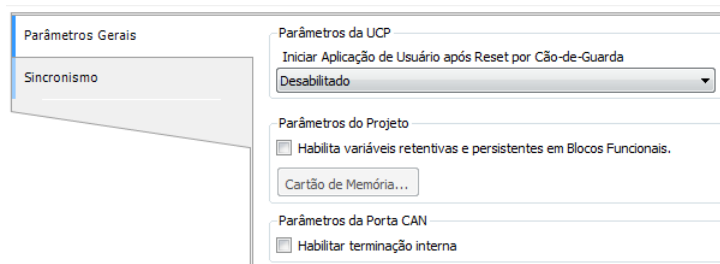


Figura 36: Configuração da UCP

Configuração	Descrição	Padrão	Opções
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 por cão-de-guarda do hardware ou através da reinicialização do Runtime, mas mantém a indicação do diagnóstico via LED DG e via variáveis	Desabilitado	Habilitado Desabilitado
Habilitar terminação interna	Quando habilitado, utiliza terminação interna na interface CAN	Desmarcado	Marcado: terminação interna na interface CAN habilitada Desmarcado: terminação interna na interface CAN desabilitada

Tabela 34: Configuração da UCP

5.2.2. Sincronização de Tempo

Para sincronização temporal, controladores Nexto Xpress usam o protocolo SNTP (Simple Network Time Protocol) ou o sincronismo através do IEC 60870-5-104.

Para usar os protocolos de sincronismo, o usuário deve configurar os seguintes parâmetros na aba *Sincronismo* localizada dentro da Configuração da UCP, em XP3xx, na visualização em árvore.

Parâmetros Gerais

Sincronismo

Configurações Gerais

Fuso Horário: -03 : 00

Configurações SNTP

☒ Serviço SNTP

Período de Sincronização do SNTP (x1 s): 60

Erro Mínimo Antes da Atualização do Relógio (x1 ms): 100

Endereço IP do Primeiro Servidor SNTP: 192 . 168 . 15 . 10

Endereço IP do Segundo Servidor SNTP: 192 . 168 . 15 . 11

Figura 37: Configuração SNTP

Configuração	Descrição	Padrão	Possibilidades
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	Habilitado Desabilitado
Período de Sincronização do SNTP (x1 seg)	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 1º Servidor SNTP	Endereço IP do servidor SNTP primário.	192.168.15.10	1.0.0.1 a 223.255.255.254.
Endereço IP do 2º Servidor SNTP	Endereço IP do servidor SNTP secundário.	192.168.15.11	1.0.0.1 a 223.255.255.254.

Tabela 35: 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. Se ambos os campos estiverem vazios, o serviço SNTP permanecerá desabilitado.

Fuso Horário: A configuração de fuso horário é usada para converter a hora local em UTC e vice-versa. Enquanto algumas fontes de sincronização usam o horário local (Protocolo IEC 60870-5-104, Função *SetDateAndTime*), outras usam o horário UTC (SNTP). A hora UTC é geralmente usada para marcar eventos (Protocolo IEC 60870-5-104 e LOG de Dispositivo do Mastertool), enquanto a hora local é usada por outros recursos da UCP (função *GetDateAndTime*).

É permitido ativar mais de uma fonte de sincronização no projeto, no entanto, o dispositivo não suporta o sincronismo de mais de uma fonte de sincronização durante a operação. Portanto, há implicitamente definido um mecanismo de prioridade. O sincronismo através do SNTP é mais prioritário do que através do protocolo IEC 60870-5-104. Portanto, quando as duas fontes estiverem ativas e o servidor SNTP estiver presente, ele será responsável pela sincronização do relógio da UCP e qualquer comando de sincronização do IEC 60870-5-104 será negado.

5.2.2.1. 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.

5.2.2.2. SNTP

Para o sincronismo via SNTP, o controlador 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. 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.

O controlador 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 1º Servidor. Caso este não responda, as requisições são direcionadas para o segundo servidor configurado em Endereço de IP do 2º 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, o controlador tenta se conectar uma vez em cada servidor, ele 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 do controlador e o recebido pelo servidor for maior que o valor configurado no parâmetro *Erro Mínimo Antes da Atualização do Relógio*, o horário do controlador 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, o controlador 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, o controlador 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 segundos para o *Período de Sincronização*. Caso o servidor primário não responda, o tempo para a sincronização será 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 será de, no mínimo, 10 s (aguardo da resposta dos dois servidores e nova tentativa de conexão com o 1º servidor).

ATENÇÃO

O Serviço SNTP depende da aplicação do usuário apenas para a sua configuração. Portanto, este serviço vai ser executado mesmo quando o controlador estiver nos modos *STOP* ou *BRE-AKPOINT*, desde que exista uma aplicação no controlador com o cliente SNTP habilitado e corretamente configurado.

5.2.2.3. 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 diminuí-lo em uma hora.

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

5.2.3. 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).

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*.

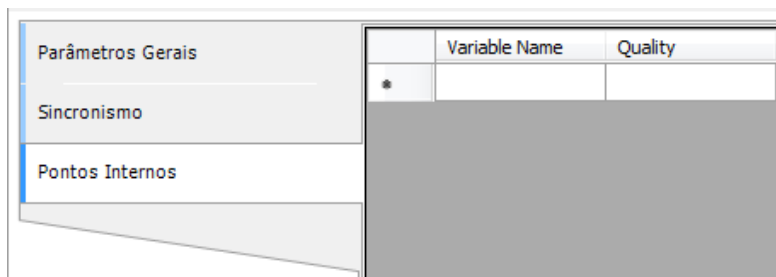


Figura 38: Tela de Configuração dos Pontos Internos

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 36: Configuração dos Pontos Internos

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

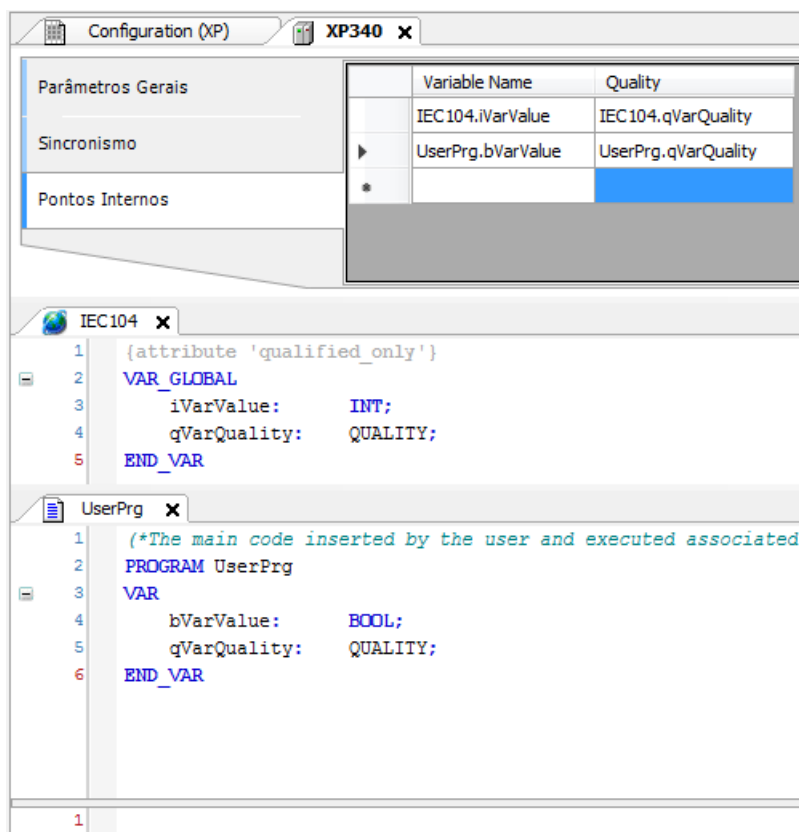


Figura 39: Exemplo de Configuração dos Pontos Internos

5.2.3.1. 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 como 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.

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.

5.2.3.1.1. 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 37: Estrutura QUALITY

5.2.3.1.2. 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 38: 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 39: 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 40: Conversão Interno para IEC 60870-5-104 de Contadores

5.2.3.1.3. 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 41: Qualidade MODBUS

5.3. Configuração da Interface Serial

5.3.1. COM 1

A interface COM 1 é uma interface serial padrão RS-485. Permite comunicação ponto-a-ponto ou por rede nos protocolos escravo MODBUS RTU ou mestre MODBUS RTU.

Ao utilizar o protocolo mestre/escravo MODBUS, alguns desses parâmetros (como Modo Serial, Bits de Dado, RX Threshold e Eventos Seriais) são ajustados automaticamente pela ferramenta Mastertool, visando o correto funcionamento deste protocolo.

Os parâmetros que devem ser configurados para o funcionamento adequado da aplicação são descritos abaixo:

Configuração	Descrição	Padrão	Opções
Tipo serial	Configuração de canal serial	RS-485	RS-485
Taxa de Transmissão	Configuração da velocidade da porta de comunicação serial	115200	2400, 4800, 9600, 19200, 38400, 57600, 115200 bps
Paridade	Configuração da paridade da porta serial	Nenhuma	Impar Par Sem paridade
Bits de dado	Configura a quantidade de bits de dados em cada caractere da comunicação serial	8	6, 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 de operação prolongado que fornece informações relativas ao quadro de dados recebido (veja a nota na seção COM 1) - Modo normal: modo de operação normal de comunicação serial

Tabela 42: Configuração serial padrão RS-485

5.3.2. Configurações Avançadas

A seção de configurações avançadas permite configurar parâmetros adicionais da porta serial conforme descrito abaixo:

Configuração	Descrição	Padrão	Opções
Threshold de RX da UART	Quantidade de bytes que deve ser recebida para uma nova interrupção UART ser gerada. Valores baixos tornam o TIMESTAMP mais preciso quando o MODO ESTENDIDO é usado e minimiza os erros de saturação. No entanto, valores muito baixos podem causar várias interrupções atrasando a UCP.	8	1, 4, 8 e 14
Terminação RS-485	Ativa a terminação do padrão RS-485 interna. Deve ser ativado somente se o controlador estiver fisicamente posicionado em uma das extremidades da rede RS-485.	Habilitado	Desabilitado ou Habilitado

Tabela 43: Configurações Avançadas da Comunicação Serial RS-485

5.4. Configuração da Interface Ethernet

A interface é composta por um conector de comunicação RJ45 10/100Base-TX padrão. Permite comunicação ponto-a-ponto ou por rede com vários protocolos, por exemplo: MODBUS, OPC UA, etc.

Os parâmetros que devem ser configurados para o funcionamento adequado da aplicação são descritos abaixo:

5.4.1. NET 1

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

Tabela 44: Configuração da NET 1

5.4.2. 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 45: Portas TCP/UDP reservadas

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

5.4.3. Roteamento de Rede

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)

Se nenhuma dessas condições ocorrer, o sistema operacional encaminhará a comunicação para o Gateway Padrão configurado na interface NET integrada.

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.

5.5. Configuração da Interface CAN (Controller Area Network)

5.5.1. 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, 800, 1000

Tabela 46: Configuração CAN

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

5.6. Configuração das E/S Integradas

Os controladores Nexto Xpress possuem pontos de E/S integrados, o que permite a interface com dispositivos externos como sensores, atuadores, motores de passo, encoders, etc.

Existem dois objetos na estrutura do projeto relacionados às E/S Integradas, conforme mostrado na figura abaixo:

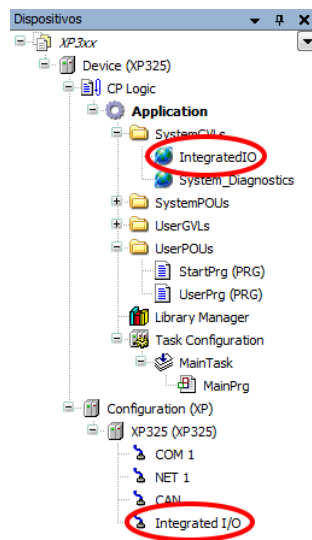


Figura 40: E/S Integradas na árvore de projeto

Um desses objetos é a GVL *IntegratedIO*, que é criada automaticamente pelo Mastertool e contém uma lista de variáveis simbólicas globais que são mapeadas diretamente para as entradas e saídas integradas.

O outro objeto é o conector *Integrated IO*, que contém a configuração para cada tipo de ponto de E/S. Essas configurações serão detalhadas nas próximas seções.

5.6.1. Entradas Digitais

Os parâmetros relacionados às *Entradas Digitais* estão localizados na tela abaixo (exemplo do XP325), tanto para entradas padrão quanto entradas rápidas (quando configuradas como entradas digitais normais):

Figura 41: Parâmetros de Entradas Digitais

A tabela abaixo mostra as configurações possíveis:

Configuração	Descrição	Padrão	Opções
Tempo do Filtro de Entrada	Tempo mínimo que uma entrada deve permanecer em um determinado estado para confirmar a mudança de estado	20 ms	2 a 255 ms
Filtro	Habilita/Desabilita o filtro para cada entrada	Desabilitado	Habilitado Desabilitado

Tabela 47: Parâmetros de Entradas Digitais

Nota:

Tempo do Filtro de Entrada: A amostragem do filtro é executada na MainTask (ou função Refresh), então é recomendado usar valores múltiplos do intervalo da tarefa.

5.6.2. Entradas Rápidas

As entradas rápidas são sinais de entrada especiais que podem ser usados para funções especiais de alta velocidade. Essas entradas físicas especiais podem ser atribuídas a dois tipos de elementos lógicos: contadores rápidos e interrupção externa. Cada um desses elementos lógicos consome certa quantidade de sinais de entrada rápida. Para a unidade de contadores rápidos, depende do modo selecionado (Up / Down, Quadratura, etc...), enquanto cada interrupção externa usa um sinal de entrada rápida. O número de entradas físicas rápidas, bem como o número máximo de elementos lógicos de contadores rápidos e interrupções externas designados para essas entradas é descrito em [Descrição Técnica](#).

A tabela a seguir mostra como cada unidade de contador rápido é atribuída aos sinais de entrada rápida:

Contador Rápido	Modo do Contador	Entradas Rápidas			
		I00	I01	I02	I03
Contador 0	Up/Down (A conta, B direção) com zeramento	B	Z	A	-
	Quadratura 2X	A	B	-	-
	Quadratura 2X com zeramento	A	B	Z	-
	Quadratura 4X	A	B	-	-
	Quadratura 4X com zeramento	A	B	Z	-

Tabela 48: Alocação de Contadores Rápidos e Entradas

Para cada configuração descrita acima, os sinais de entrada rápida restantes (não usados pelas unidades de contador rápido) podem ser usados como interrupção externa, respeitando o número máximo deste tipo de elemento lógico especificado no capítulo [Descrição Técnica](#).

A configuração de contadores rápidos e interrupções está localizada na seguinte tela:

Figura 42: Configuração de Entradas Rápidas

Ao selecionar a função de uma entrada rápida, as seguintes entradas são atribuídas automaticamente (bloqueadas para edição) de acordo com o modo de contador rápido.

A tabela abaixo mostra os possíveis valores de configuração para cada entrada rápida:

Configuração	Descrição	Padrão	Opções
Entrada Rápida I00	Configuração da Entrada Rápida I00	Entrada Digital	Entrada Digital Contador 0 (Entrada B): Up/Down (A contagem, B sentido) com zeramento Contador 0 (Entrada A): Quadratura 2X Contador 0 (Entrada A): Quadratura 2X com zeramento Contador 0 (Entrada A): Quadratura 4X Contador 0 (Entrada A): Quadratura 4X com zeramento
Entrada Rápida I01	Configuração da Entrada Rápida I01	Entrada Digital	Entrada Digital Obs.: Este campo será configurado automaticamente quando a Entrada Rápida I00 estiver configurada como Contador Up/Down ou Quadratura
Entrada Rápida I02	Configuração da Entrada Rápida I02	Entrada Digital	Entrada Digital Interrupção (Borda de Subida) Obs.: Este campo será configurado automaticamente quando a Entrada Rápida I00 estiver configurada como Contador Up/Down ou Quadratura com zeramento.
Entrada Rápida I03	Configuração da Entrada Rápida I03	Entrada Digital	Entrada Digital Interrupção (Borda de Subida)

Tabela 49: Parâmetros das Entradas Rápidas

Mesmo que uma entrada rápida seja configurada como um contador ou interrupção, seu valor digital ainda pode ser lido por meio da variável *IntegratedIo.DigitalInputs*. As próximas subseções fornecem mais detalhes sobre a configuração e programação das *Entradas Rápidas*.

5.6.2.1. Contadores Rápidos

Os contadores rápidos possuem vários modos de operação. A tabela a seguir descreve os detalhes de cada um desses modos:

Modo do Contador	Formas de Onda de Contagem
Up/Down (A conta, B sentido) com zeramento	Input A Input B Input Z Counter Value: 31 32 0 1 0 -1
Quadratura 2X	Input A Input B Counter Value: 29 30 31 32 33 32 31 30 29

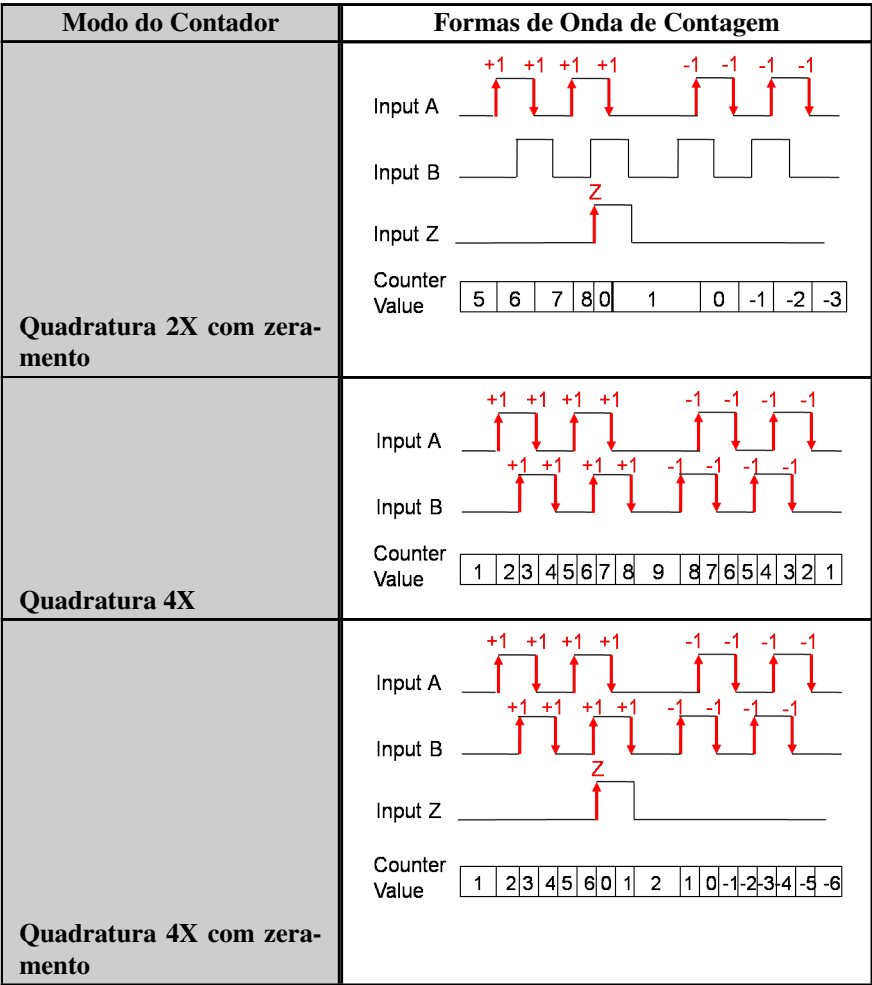


Tabela 50: Modos de Contadores Rápidos

O comportamento geral é o mesmo para todos os contadores: quando se incrementa o contador e o valor positivo máximo foi atingido, o próximo valor será o valor negativo mínimo. A mesma coisa acontece para a direção oposta, então quando se decrementa o contador e o valor negativo mínimo foi atingido, o próximo valor será o valor positivo máximo.

O programa do usuário pode acessar os contadores rápidos através da estrutura simbólica *FastInputs*, que é criada automaticamente na GVL *IntegratedIo*. Para cada unidade do contador rápido, existem 3 áreas principais, como mostrado na figura a seguir:

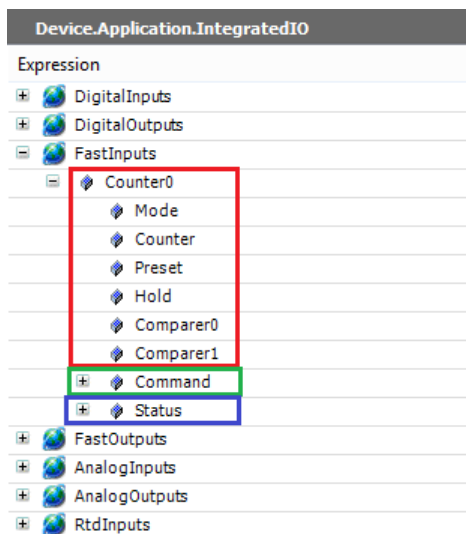


Figura 43: Estrutura dos contadores

A tabela abaixo descreve a estrutura das variáveis do contador:

Variável	Descrição	Tipo	Valores Permitidos
Mode	Modo configurado do contador (somente leitura)	ENUM_COUNTER_MODE	DISABLED UP_DOWN_A_COUNT_B_DIR_WITH_ZERO QUADRATURE_2X QUADRATURE_2X_WITH_ZERO QUADRATURE_4X QUADRATURE_4X_WITH_ZERO
Counter	Valor do contador	DINT	-2147483648 a 2147483647
Preset	Valor de preset	DINT	-2147483648 a 2147483647
Hold	Valor de hold	DINT	-2147483648 a 2147483647
Comparer0	Valor menor de comparação com o contador	DINT	-2147483648 a 2147483647
Comparer1	Valor maior de comparação com o contador	DINT	-2147483648 a 2147483647
Command	Estrutura de comandos do contador	T_COUNTER_COMMAND	-
Status	Estrutura de estados do contador	T_COUNTER_STATUS	-

Tabela 51: Estrutura de variáveis do contador

O comando e o estado são estruturas de bits que permitem ao programa do usuário controlar a operação do contador. A tabela a seguir descreve a estrutura de comando do contador:

Variável	Descrição	Tipo	Valores Permitidos
Stop	Para o contador, que se mantém parado enquanto o bit está setado	BIT	FALSE ou TRUE
Reset	Reset do contador, que se mantém zerado enquanto este bit está setado	BIT	FALSE ou TRUE
Load	Carregamento do valor de preset, que ocorre na borda de subida do comando	BIT	FALSE ou TRUE
Sample	Amostragem do valor do contador, que é salvo no campo hold quando ocorrer uma borda de subida neste comando	BIT	FALSE ou TRUE

Tabela 52: Estrutura de comando do contador

A tabela a seguir descreve a estrutura de estado do contador.

Variável	Descrição	Tipo	Valores Permitidos
Enabled	Contador está habilitado	BIT	FALSE ou TRUE
Direction	Direção do contador (TRUE = Incremento, FALSE = Decremento)	BIT	FALSE ou TRUE
EQComparer0	Valor do contador é igual à Comparer0	BIT	FALSE ou TRUE
LTComparer0	Valor do contador é menor que Comparer0	BIT	FALSE ou TRUE
GTComparer0	Valor do contador é maior que Comparer0	BIT	FALSE ou TRUE
EQComparer1	Valor do contador é igual à Comparer1	BIT	FALSE ou TRUE
LTComparer1	Valor do contador é menor que Comparer1	BIT	FALSE ou TRUE
GTComparer1	Valor do contador é maior que Comparer1	BIT	FALSE ou TRUE

Tabela 53: Estrutura de estado do contador

Além das variáveis globais *IntegratedIo*, existe um bloco de funções da biblioteca *LibIntegratedIo* que permite instanciar o contador rápido em POUs escritas em linguagens gráficas (por exemplo, Diagrama Lógico Ladder). Esse bloco de funções é, na verdade, um invólucro para as estruturas de variáveis descritas anteriormente. A figura abaixo mostra um bloco de funções instanciado em um programa Ladder.

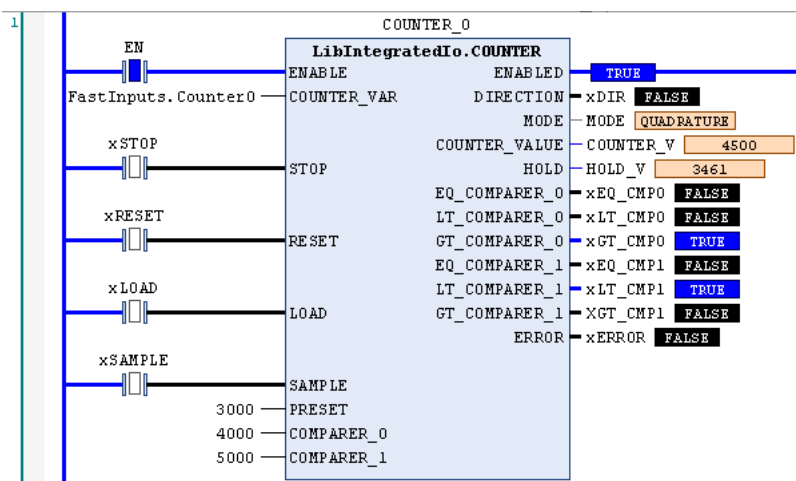


Figura 44: Bloco Funcional LibIntegratedIo.COUNTER

A tabela abaixo descreve as variáveis de entrada e saída do bloco de funções.

Variável	Descrição	Tipo	Valores Permitidos
ENABLE	Habilita a execução do bloco	BOOL	FALSE ou TRUE
COUNTER_VAR	Variável do contador	REFERENCE TO T_COUNTER	FastInputs.Counter0
STOP	Para o contador	BOOL	FALSE ou TRUE
RESET	Reseta o contador	BOOL	FALSE ou TRUE
LOAD	Carrega o valor de preset no contador	BOOL	FALSE ou TRUE
SAMPLE	Amostra o valor do contador em hold	BOOL	FALSE ou TRUE
PRESET	Valor de preset	DINT	-2147483648 a 2147483647
COMPARER_0	Menor valor de comparação	DINT	-2147483648 a 2147483647
COMPARER_1	Maior valor de comparação	DINT	-2147483648 a 2147483647
ENABLED	Contador está habilitado	BOOL	FALSE ou TRUE
DIRECTION	Direção do contador (TRUE = Incremento, FALSE = Decremento)	BOOL	FALSE ou TRUE
Mode	Modo do Contador	ENUM_COUNTER_MODE	DISABLED UP_DOWN_A_COUNT_B_DIR_WITH_ZERO QUADRATURE_2X QUADRATURE_2X_WITH_ZERO QUADRATURE_4X QUADRATURE_4X_WITH_ZERO
COUNTER_VALUE	Valor do contador	DINT	-2147483648 a 2147483647
HOLD	Valor de hold	DINT	-2147483648 a 2147483647

Variável	Descrição	Tipo	Valores Permitidos
EQ_COMPARER_0	Valor do contador é igual à Comparer0	BOOL	FALSE ou TRUE
LT_COMPARER_0	Valor do contador é menor que Comparer0	BOOL	FALSE ou TRUE
GT_COMPARER_0	Valor do contador é maior que Comparer0	BOOL	FALSE ou TRUE
EQ_COMPARER_1	Valor do contador é igual à Comparer1	BOOL	FALSE ou TRUE
LT_COMPARER_1	Valor do contador é menor que Comparer1	BOOL	FALSE ou TRUE
GT_COMPARER_1	Valor do contador é maior que Comparer1	BOOL	FALSE ou TRUE
ERROR	Erro ocorreu na execução do bloco funcional. Pode ser causado por um COUNTER_VAR inválido ou o contador está desabilitado.	BOOL	FALSE ou TRUE

Tabela 54: Descrição do bloco funcional LibIntegratedIo.COUNTER

5.6.2.1.1. Interrupções do Contador

Os contadores rápidos têm a capacidade de gerar interrupções por comparação, isto é, quando o contador atinge um determinado valor de comparação, uma tarefa específica será executada e interromperá a execução do programa principal. Cada unidade de contador rápido possui dois valores de comparação, chamados *Comparer0* e *Comparer1*, que estão presentes na estrutura de dados simbólicos global correspondente ou Bloco Funcional, conforme descrito nas seções anteriores. A configuração da interrupção do contador para cada unidade de contador rápido está localizada na seguinte tela:

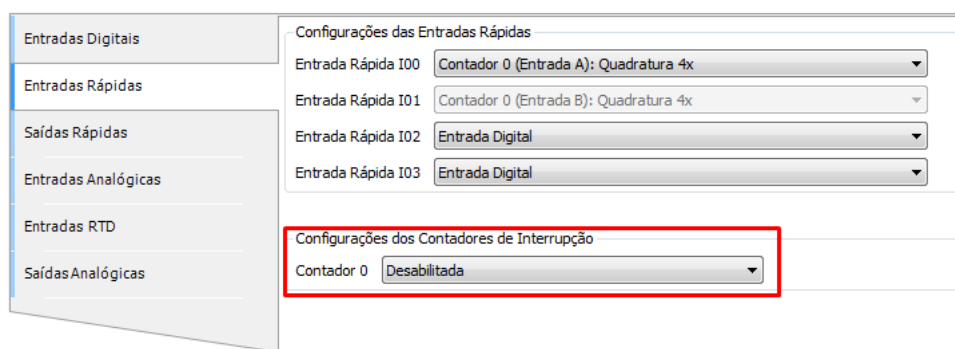


Figura 45: Configurações da interrupção do contador

A tabela abaixo mostra os possíveis valores de configuração para a interrupção do contador:

Configuração	Descrição	Padrão	Opções
Contador 0	Configuração do comparador de interrupção do Contador 0	Desabilitada	Desabilitada Counter0InterruptTask: Contador igual ao Comparer0 Obs.: Esta configuração está disponível quando o Contador 0 está associado a alguma entrada rápida.

Tabela 55: Parâmetros da interrupção do contador

A interrupção por comparação irá gerar um evento específico. Esse evento deve acionar a execução de uma tarefa de evento externo, que deve chamar uma POU específica. Por exemplo, o evento de comparação gerado para o contador 0 é chamado *COUNTER0_EVT*. Portanto, uma tarefa de evento externo chamada *Counter0InterruptTask* deve ser configurada para ser acionada por esse evento e deve chamar uma POU chamada *Counter0InterruptPrg* que conterá o programa do usuário a ser executado.

A figura abaixo mostra este cenário de configuração no Mastertool.

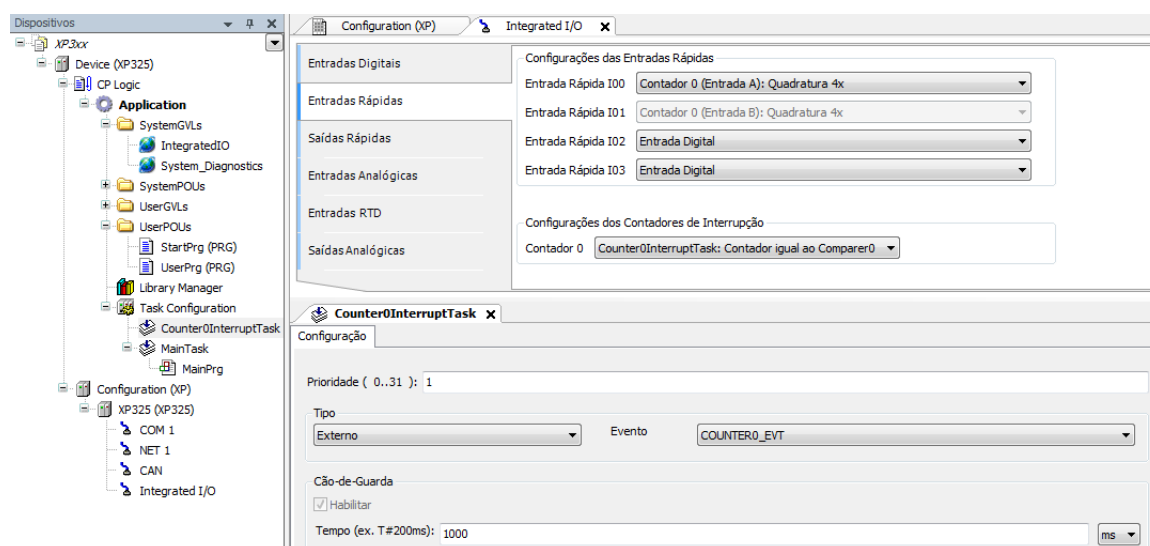


Figura 46: Configurações da Interrupção do Contador

5.6.2.2. Interrupção Externa

As entradas rápidas podem ser configuradas no modo de *Interrupção (Borda de Subida)*, o que significa que quando uma borda de subida (transição de 0V para 24V) é executada na entrada, uma tarefa específica será executada e interromperá a execução do programa principal.

Cada interrupção externa irá gerar um evento específico. Esse evento deve acionar a execução da tarefa de evento externo, que deve chamar uma POU específica. Por exemplo, o evento de interrupção externa gerado para a entrada rápida 102 é chamado *FIN2_EVT*. Portanto, uma tarefa de evento externo chamada *FastInput102InterruptTask* deve ser configurada para ser acionada por esse evento e deve chamar uma POU chamada *FastInput102InterruptPrg* que conterá o programa do usuário a ser executado.

A figura abaixo mostra este cenário de configuração no Mastertool.

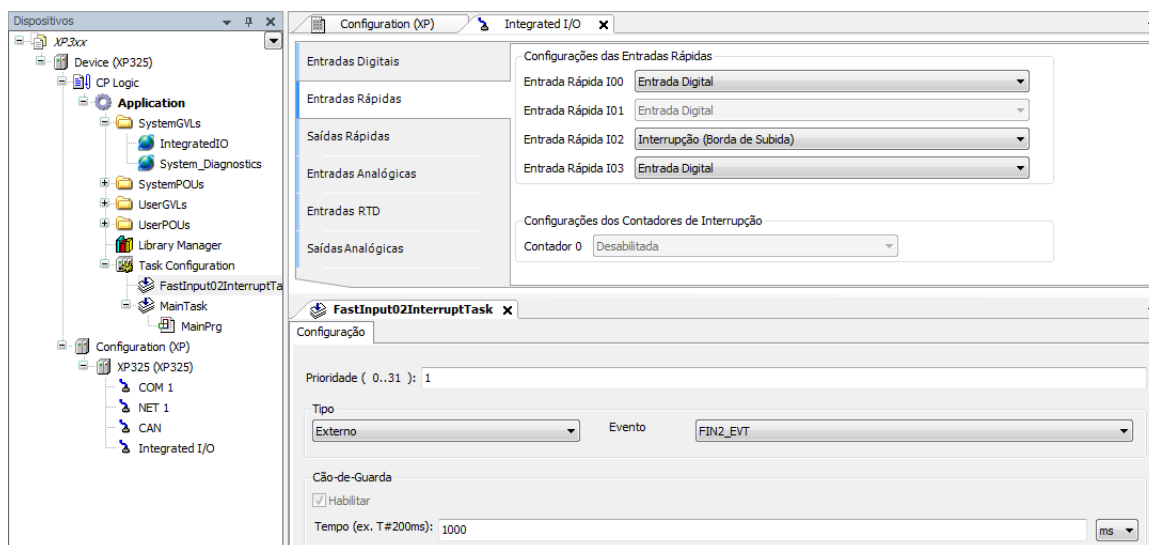


Figura 47: Configurações das Interrupções de Entradas Rápidas

ATENÇÃO

A entrada de interrupção externa possui uma janela de tempo de filtragem de 10ms para proteger o controlador contra transições espúrias no sinal de entrada. Esta janela inicia logo após a ocorrência da interrupção e, durante este tempo, qualquer outro evento de interrupção externa será descartado.

ATENÇÃO

A interrupção externa não suporta reentrância. Se outra interrupção ocorrer (após o tempo de filtro) e a execução do seu programa ainda não tiver sido concluída, esta interrupção será descartada.

5.6.3. Saídas Rápidas

As saídas rápidas são sinais de saída especiais que podem ser usados para geração de pulsos. Essas saídas físicas especiais podem ser associadas a dois tipos de elementos lógicos: VFO / PWM (frequência variável / largura de pulso) e PTO (saída de trem de pulso). Cada um desses elementos lógicos consome um sinal de saída rápida. O número de saídas físicas rápidas, bem como o número máximo de elementos lógicos VFO / PWM e PTO atribuíveis a essas saídas são descritos no capítulo [Descrição Técnica](#).

A configuração de saídas rápidas está localizada na tela seguinte:

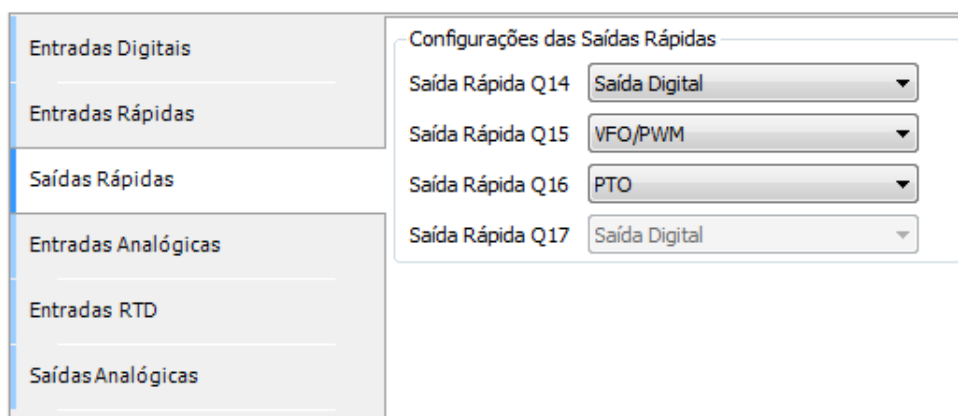


Figura 48: Parâmetros das Saídas Rápidas

5. CONFIGURAÇÃO

A tabela abaixo mostra os possíveis valores de configuração:

Configuração	Descrição	Padrão	Opções
Saída Rápida Q14	Configuração da Saída Rápida Q14.	Saída Digital	Saída Digital VFO/PWM PTO
Saída Rápida Q15	Configuração da Saída Rápida Q15.	Saída Digital	Saída Digital VFO/PWM
Saída Rápida Q16	Configuração da Saída Rápida Q16.	Saída Digital	Saída Digital VFO/PWM PTO
Saída Rápida Q17	Configuração da Saída Rápida Q17.	Saída Digital	Saída Digital VFO/PWM

Tabela 56: Parâmetros das Saídas Rápidas

A função PTO pode ser atribuída apenas para Q14 e Q16. Quando a saída é configurada neste modo, a saída adjacente (Q15 ou Q17) será forçada ao modo de saída digital normal.

Como mostrado na tabela anterior, as saídas rápidas podem ser configuradas como saída digital normal. Nesse caso, seu valor digital pode ser definido usando a variável global padrão *IntegratedIo.DigitalOutputs*.

Quando configurado como VFO/PWM ou PTO, o programa do usuário pode controlar as saídas rápidas através da estrutura simbólica *FastOutputs*, que é criada automaticamente na GVL *IntegratedIo*, conforme mostrado na figura a seguir:

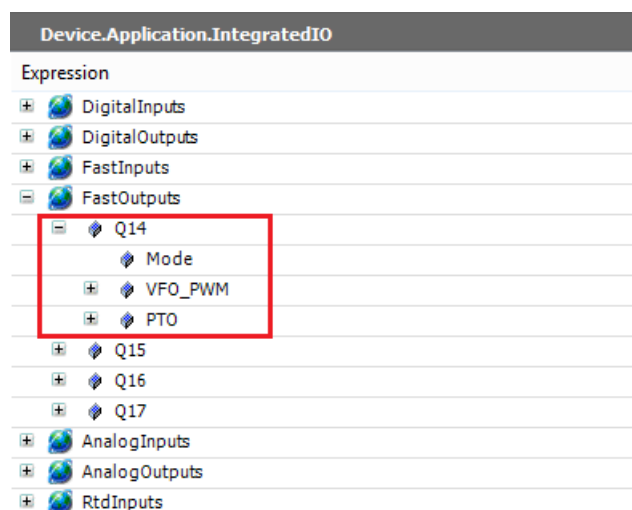


Figura 49: Estrutura das Saídas Rápidas

A tabela abaixo descreve a estrutura das variáveis de saída rápida:

Variável	Descrição	Tipo	Valores Permitidos
Mode	Modo configurado à saída rápida (somente leitura)	ENUM_FAST_OUTPUT_MODE	DIGITAL_OUTPUT PWM PTO
VFO_PWM	Estrutura VFO/PWM. Contém uma estrutura para controlar a saída rápida quando ela é configurada como VFO / PWM.	T_VFO_PWM	-
PTO	Estrutura do PTO. Contém uma estrutura para controlar a saída rápida quando ela é configurada como PTO.	T_PTO	-

Tabela 57: Estrutura de Variáveis das Saídas Rápidas

As próximas subseções fornecem mais detalhes sobre como usar essas funções do gerador de pulsos, descrevendo essas estruturas para cada modo.

5.6.3.1. VFO/PWM

O VFO / PWM (saída de frequência variável / modulador de largura de pulso) é um modo de saída geradora de pulsos onde a frequência e o ciclo de trabalho podem ser controlados pelo programa do usuário. É aplicável, por exemplo, para controlar a energia transferida para uma carga elétrica ou controlar o ângulo de um servo motor. O princípio de operação da saída do VFO / PWM é muito simples, veja a forma de onda pulsada que é mostrada na figura abaixo:

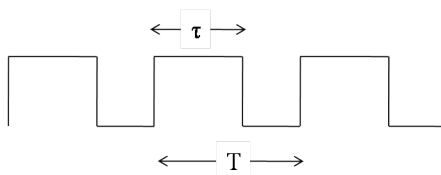


Figura 50: Forma de Onda VFO/PWM

A figura mostra uma forma de onda pulsada, onde T é o período dos pulsos e τ é a largura do pulso. Esses são os parâmetros de pulso que podem ser alterados no modo VFO / PWM. A frequência é definida como o inverso do período, então:

$$f = \frac{1}{T}$$

O ciclo de trabalho (duty cycle) é a razão entre a largura do pulso e o período, então:

$$D = \frac{\tau}{T} 100\%$$

Para controlar a saída do VFO/PWM, o programa do usuário deve acessar a variável VFO_PWM da estrutura de saída rápida. A estrutura do VFO_PWM é mostrada na tabela abaixo:

Variável	Descrição	Tipo	Valores Permitidos
Frequency	Frequência em Hertz	UDINT	1 a 200000
DutyCycle	Duty Cycle em porcentagem	USINT	0 a 100
Command	Estrutura de comando do VFO/PWM	T_VFO_PWM_COMMAND	-
Status	Estrutura de estado do VFO/PWM	T_VFO_PWM_STATUS	-

Tabela 58: Estrutura de variáveis do VFO_PWM

A tabela abaixo mostra a estrutura de comandos do VFO_PWM.

Variável	Descrição	Tipo	Valores Permitidos
Enable	Habilita a saída VFO/PWM	BIT	FALSE ou TRUE

Tabela 59: Estrutura de Comando do VFO/PWM

A tabela abaixo mostra a estrutura de estados do VFO_PWM.

Variável	Descrição	Tipo	Valores Permitidos
InvalidFrequency	Valor de frequência é inválido (fora dos limites)	BIT	FALSE ou TRUE
InvalidDutyCycle	Valor de Duty Cycle é inválido (fora dos limites)	BIT	FALSE ou TRUE

Tabela 60: Estrutura de Estados do VFO/PWM

Quando o comando Enable for TRUE, os parâmetros de entrada serão verificados continuamente e as variáveis de estado serão atualizadas de acordo.

Além das variáveis globais *IntegratedIo*, existe um bloco de funções da biblioteca *LibIntegratedIo* que permite instanciar o VFO/PWM em POU's escritas em linguagens gráficas (por exemplo, Diagrama Lógico Ladder). Esse bloco de funções é, na verdade, um invólucro para as variáveis estruturadas descritas anteriormente. A figura abaixo mostra o bloco de funções instanciado em um programa Ladder.

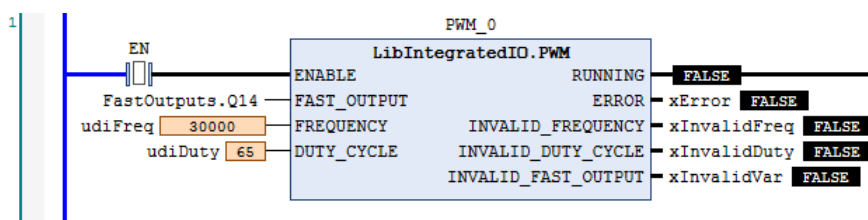


Figura 51: Bloco Funcional LibIntegratedIo.PWM

A tabela abaixo descreve as variáveis de entrada e saída do bloco de funções.

Variável	Descrição	Tipo	Valores Permitidos
ENABLE	Habilita a execução do bloco funcional	BOOL	FALSE ou TRUE
FAST_OUTPUT	Variável de Saída Rápida	REFERENCE TO T_FAST_OUTPUT	FastOutputs.Q14 FastOutputs.Q15 FastOutputs.Q16 FastOutputs.Q17
FREQUENCY	Frequência em Hertz	UDINT	1 a 200000
DUTY_CYCLE	Duty Cycle em porcentagem	USINT	0 a 100
RUNNING	VFO/PWM está sendo executado	BOOL	FALSE ou TRUE
ERROR	Erro ocorrido na execução do bloco funcional. As variáveis a seguir fornecem informações detalhadas.	BOOL	FALSE ou TRUE
INVALID_FREQUENCY	O valor de frequência é inválido (fora do intervalo)	BOOL	FALSE ou TRUE
INVALID_DUTY_CYCLE	O valor de Duty Cycle é inválido (fora do intervalo)	BOOL	FALSE ou TRUE
INVALID_FAST_OUTPUT	FAST_OUTPUT não foi atribuída ao bloco ou não foi configurada como VFO/PWM.	BOOL	FALSE ou TRUE

Tabela 61: Descrição do bloco funcional LibIntegratedIo.PWM

5.6.3.2. PTO

A PTO (saída de trem de pulso) é um modo de gerador de pulsos. É usado, por exemplo, para controlar motores de passo responsáveis pelo posicionamento de mecanismos com inércia considerável. Para estes casos, a velocidade de rotação deve aumentar lentamente (aceleração) quando o movimento estiver começando e diminuir lentamente (desaceleração) quando o movimento estiver parando. Estas acelerações e desacelerações são feitas no trem de pulsos aumentando e diminuindo a frequência dos pulsos, mantendo os 50% do ciclo de trabalho.

Há um conjunto de parâmetros que devem ser definidos para um trem de pulsos: frequência de partida, frequência de operação, frequência de parada, perfil de aceleração, número total de pulsos, número de pulsos na etapa de aceleração, número de pulsos na etapa de desaceleração. A figura abaixo mostra, no plano cartesiano, a relação entre a frequência dos pulsos e o tempo. O trem de pulso mostrado é chamado de perfil trapezoidal, porque as rampas de aceleração e desaceleração produzem uma forma de trapézio.

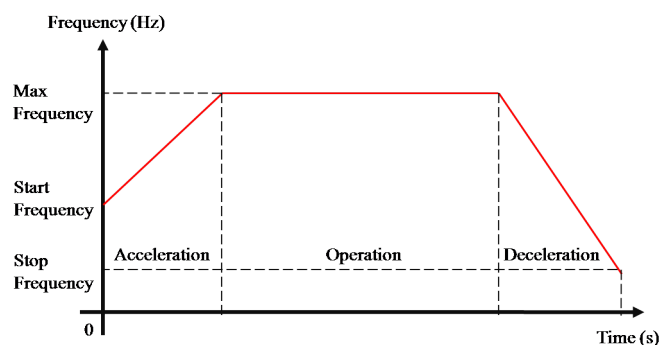


Figura 52: PTO com perfil trapezoidal

Para algumas aplicações, é mais recomendado usar o perfil "S", cujas curvas de aceleração e desaceleração são semelhantes à forma "S". A figura abaixo mostra este perfil.

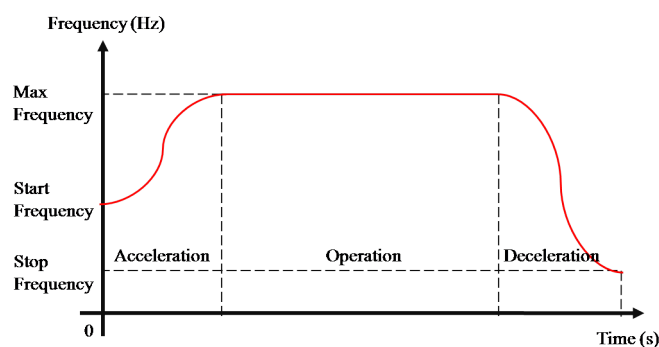


Figura 53: PTO com perfil "S"

Além dos parâmetros da PTO, há informações de estados e comandos que o programa do usuário pode usar para controlar a saída. Algumas informações de estados importantes são o contador de pulsos (proporcional a uma posição), a etapa (step) do trem de pulso (aceleração, operação, desaceleração) e, mesmo, se a saída está funcionando bem. Os comandos necessários para controlar a PTO são para iniciar o trem de pulsos (start), parar o trem de pulsos (stop) e parar o trem de pulsos suavemente (softstop). O comando de parada suave é muito importante, uma vez que pode ser usado para situações de emergência em que o sistema não pode parar abruptamente. As figuras abaixo mostram como o comando de parada suave altera o trem de pulso quando ele é executado. As linhas azuis tracejadas representam a PTO se o comando de parada suave for executado nas etapas de aceleração e operação. O comando de parada suave na etapa de desaceleração não tem efeito, uma vez que o sistema já está parando.

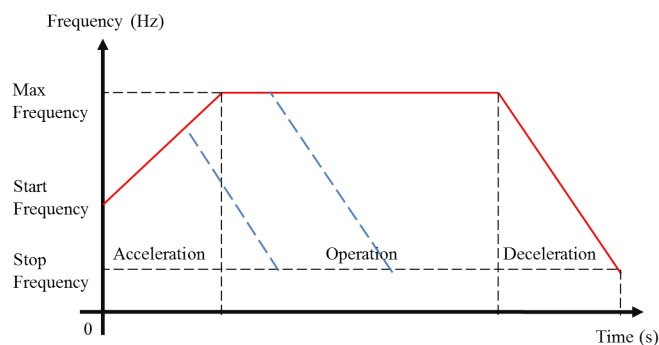


Figura 54: Parada suave da PTO no perfil trapezoidal

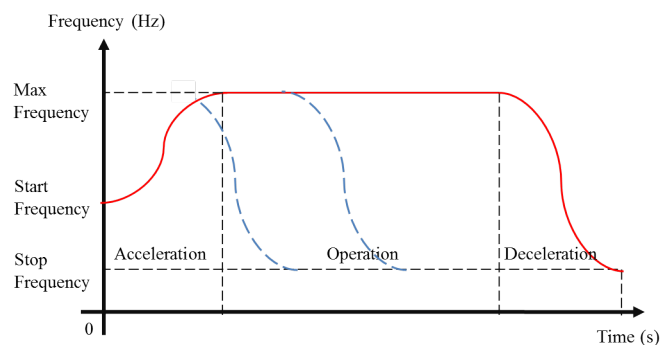


Figura 55: Parada suave da PTO no perfil S

Para controlar a PTO, o programa do usuário deve acessar a variável PTO da estrutura de saída rápida. A estrutura da PTO é mostrada na tabela abaixo:

Variável	Descrição	Tipo	Valores Permitidos
StartFrequency	Frequência inicial em Hertz	UDINT	0 a 200000
StopFrequency	Frequência de parada em Hertz	UDINT	0 a 200000
MaxFrequency	Frequência máxima em Hertz	UDINT	1 a 200000
AccelerationProfile	Perfil de aceleração (FALSE = Trapezoidal, TRUE = perfil S)	BOOL	FALSE ou TRUE
AccelerationPulses	Pulsos na aceleração	UDINT	0 a (TotalPulses-DecelerationPulses-1)
DecelerationPulses	Pulsos na desaceleração	UDINT	0 a (TotalPulses-AccelerationPulses-1)
TotalPulses	Número total de pulsos	UDINT	1 a 4294967295
PulsesCounter	Número de pulsos gerados para o trem de pulsos atual	UDINT	0 a 4294967295
Command	Estrutura de comando da PTO	T_PTO_COMMAND	-
Status	Estrutura de estados da PTO	T_PTO_STATUS	-

Tabela 62: Estrutura de variáveis da PTO

A tabela abaixo mostra a estrutura de comandos da PTO.

Variável	Descrição	Tipo	Valores Permitidos
Start	Inicia o trem de pulsos quando este bit é setado (borda de subida)	BIT	FALSE ou TRUE
Stop	Para o trem de pulsos quando este bit é setado (borda de subida)	BIT	FALSE ou TRUE
Softstop	Para suavemente o trem de pulsos quando este bit é setado (borda de subida)	BIT	FALSE ou TRUE

Tabela 63: Estrutura de comandos da PTO

A tabela abaixo mostra a estrutura de estados da PTO.

Variável	Descrição	Tipo	Valores Permitidos
Running	Trem de pulsos está sendo executado	BIT	FALSE ou TRUE
Acceleration	Etapa de aceleração (da StartFrequency a MaxFrequency)	BIT	FALSE ou TRUE
Deceleration	Etapa de desaceleração (da MaxFrequency a StopFrequency)	BIT	FALSE ou TRUE
Operation	Etapa de Operação (Max-Frequency)	BIT	FALSE ou TRUE
Done	Trem de pulsos já foi executado	BIT	FALSE ou TRUE
InvalidFrequency	A frequência (inicial, de parada ou máxima) é inválida	BIT	FALSE ou TRUE
InvalidPulses	Número de pulsos (TotalPulses, Acceleration ou Deceleration) é inválido	BIT	FALSE ou TRUE

Tabela 64: Estrutura de estados da PTO

Quando o comando Start for TRUE, os parâmetros de entrada serão verificados continuamente e as variáveis de estado serão atualizadas de acordo.

Além das variáveis globais *IntegratedIo*, existe um bloco funcional na biblioteca LibIntegratedIo que permite instanciar PTO em POUs escritas em linguagens gráficas (por exemplo, Diagrama Lógico Ladder). Esse bloco de funções é, na verdade, um invólucro para as estruturas de variáveis descritas anteriormente. A figura abaixo mostra o bloco funcional instanciado em um programa Ladder.

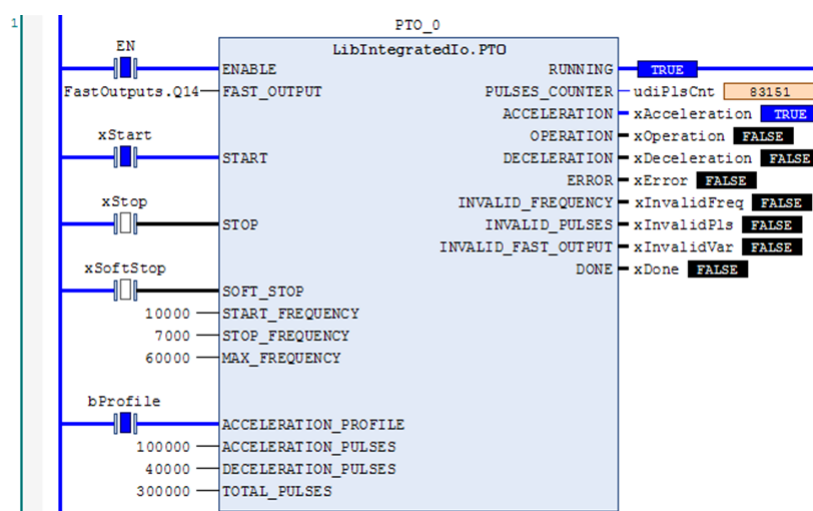


Figura 56: Bloco Funcional LibIntegratedIo.PTO

A tabela abaixo descreve as variáveis de entrada e saída do bloco funcional.

Variável	Descrição	Tipo	Valores Permitidos
ENABLE	Habilita a execução do bloco funcional	BOOL	FALSE ou TRUE
FAST_OUTPUT	Variável de Saída Rápida	REFERENCE TO T_FAST_OUTPUT	FastOutputs.Q14 FastOutputs.Q15 FastOutputs.Q16 FastOutputs.Q17
START	Inicia o trem de pulsos quando este bit é setado (borda de subida)	BOOL	FALSE ou TRUE
STOP	Para o trem de pulsos quando este bit é setado (borda de subida)	BOOL	FALSE ou TRUE
SOFT_STOP	Para suavemente o trem de pulsos quando este bit é setado (borda de subida)	BOOL	FALSE ou TRUE
START_FREQUENCY	Frequência inicial em Hertz	UDINT	1 a 200000
STOP_FREQUENCY	Frequência de parada em Hertz	UDINT	1 a 200000
MAX_FREQUENCY	Frequência máxima em Hertz	UDINT	1 a 200000
ACCELERATION_PROFILE	Perfil de aceleração (FALSE = Trapezoidal, TRUE = perfil S)	BOOL	FALSE ou TRUE
ACCELERATION_PULSES	Pulsos durante a aceleração	UDINT	0 a (TotalPulses-DecelerationPulses-1)
DECELERATION_PULSES	Pulsos durante a desaceleração	UDINT	0 a (TotalPulses-AccelerationPulses-1)
TOTAL_PULSES	Número total de pulsos	UDINT	1 a 4294967295
RUNNING	Trem de pulsos está sendo executado	BOOL	FALSE ou TRUE
PULSES_COUNTER	Número de pulsos gerados para o atual trem de pulsos	UDINT	0 a 4294967295
ACCELERATION	Etapa de aceleração (da frequência inicial à máxima)	BOOL	FALSE ou TRUE
OPERATION	Etapa de operação (frequência máxima)	BOOL	FALSE ou TRUE

Variável	Descrição	Tipo	Valores Permitidos
DECELERATION	Etapa de desaceleração (da frequência máxima à de parada)	BOOL	FALSE ou TRUE
ERROR	Erro ocorrido na execução do bloco funcional. As variáveis a seguir detalham o erro.	BOOL	FALSE ou TRUE
INVALID_FREQUENCY	Frequência (inicial, de parada ou máxima) é inválida	BOOL	FALSE ou TRUE
INVALID_PULSES	Número de pulsos (aceleração ou desaceleração) é inválido	BOOL	FALSE ou TRUE
INVALID_FAST_OUTPUT	FAST_OUTPUT não foi atribuída ao bloco ou não está configurada como PTO.	BOOL	FALSE ou TRUE
DONE	Trem de pulsos já foi executado	BOOL	FALSE ou TRUE

Tabela 65: Descrição do bloco funcional LibIntegratedIo.PTO

5.6.4. Entradas Analógicas

Os parâmetros relacionados às Entradas Analógicas são mostrados abaixo:

Configurações das Entradas Analógicas			
	Tipo de Entrada	Filtro Digital	Valor de Circuito Aberto
Entrada AI0	Não Configurado	Desabilitado	Desabilitado
Entrada AI1	Não Configurado	Desabilitado	Desabilitado
Entrada AI2	Não Configurado	Desabilitado	Desabilitado
Entrada AI3	Não Configurado	Desabilitado	Desabilitado
Entrada AI4	Não Configurado	Desabilitado	Desabilitado

Figura 57: Parâmetros das Entradas Analógicas

A tabela abaixo mostra os possíveis valores de configuração:

Configuração	Descrição	Padrão	Opções
Tipo de Entrada	Seleciona o tipo de entrada	Não Configurado	Não Configurado Tensão 0 - 10 Vdc Corrente 0 - 20 mA Corrente 4 - 20 mA
Filtro Digital	Habilita/desabilita um filtro digital passa-baixa de primeira ordem para cada entrada	Desabilitado	Desabilitado 100 ms 1 s 10 s
Valor de Circuito Aberto	Define o valor quando em condição de circuito aberto (válido apenas para escala de 4 a 20 mA).	Desabilitado	Desabilitado 0 30000

Tabela 66: Parâmetros das Entradas Analógicas

Notas:

Tipo de Entrada: Certifique-se de usar o pino adequado no bloco terminal correspondente ao tipo selecionado (tensão ou corrente).

Valor de Circuito Aberto: Determina o comportamento da variável de entrada quando configurada para escala 4 – 20 mA e corrente inferior a 3 mA.

5.6.5. Entradas RTD

Os parâmetros relacionados às entradas do RTD são mostrados abaixo:

Figura 58: Parâmetros das entradas RTD

A tabela abaixo mostra os possíveis valores de configuração:

Configuração	Descrição	Padrão	Opções
Unidade de Temperatura	Seleciona a unidade de temperatura	Graus Celsius	Graus Celsius Graus Fahrenheit
Tipo de Entrada	Seleciona o tipo de entrada	Não Configurado	Não Configurado 400 Ω 4000 Ω Pt100A Pt100E Pt1000A Pt1000E
Filtro Digital	Habilita / desabilita um filtro digital passa-baixa para cada entrada	Desabilitado	Desabilitado 100 ms 1 s 10 s

Tabela 67: Parâmetros das entradas RTD

A tabela a seguir descreve detalhes adicionais sobre cada tipo de entrada:

Tipo de Entrada	Coefficiente de Temperatura (α)	Banda de Medida	Contagem	Resolução
400 Ω	-	0 a 400 Ω	0 a 4000	0,1 Ω
4000 Ω	-	0 a 4000 Ω	0 a 4000	1 Ω
Pt100E, Pt1000E	0,00385	-200 a 850 °C -328 a 1562 °F	-2000 a 8500 -3280 a 15620	0,3 °C 0,6 °F
Pt100A, Pt1000A	0,003916	-200 a 630 °C -328 a 1166 °F	-2000 a 6300 -3280 a 11660	0,3 °C 0,6 °F

Tabela 68: Tipos de entrada RTD

5.6.6. Saídas Analógicas

Os parâmetros relacionados às saídas analógicas são mostrados abaixo:

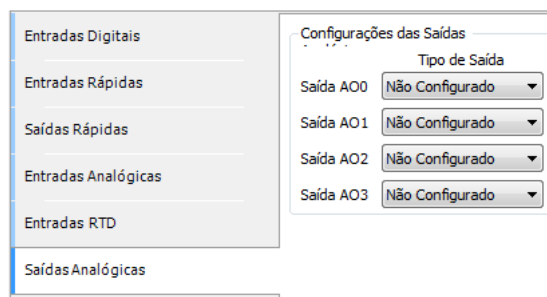


Figura 59: Parâmetros das Saídas Analógicas

A tabela abaixo mostra os possíveis valores de configuração:

Configuração	Descrição	Padrão	Opções
Tipo de Saída	Seleciona o tipo de saída	Não Configurado	Não Configurado Tensão 0 - 10 Vdc Corrente 0 - 20 mA Corrente 4 - 20 mA

Tabela 69: Parâmetros das Saídas Analógicas

5.6.7. Mapeamento de E/S

Na aba *Mapeamento de E/S* é possível configurar nome e descrição para cada variável de entrada e saída do controlador.

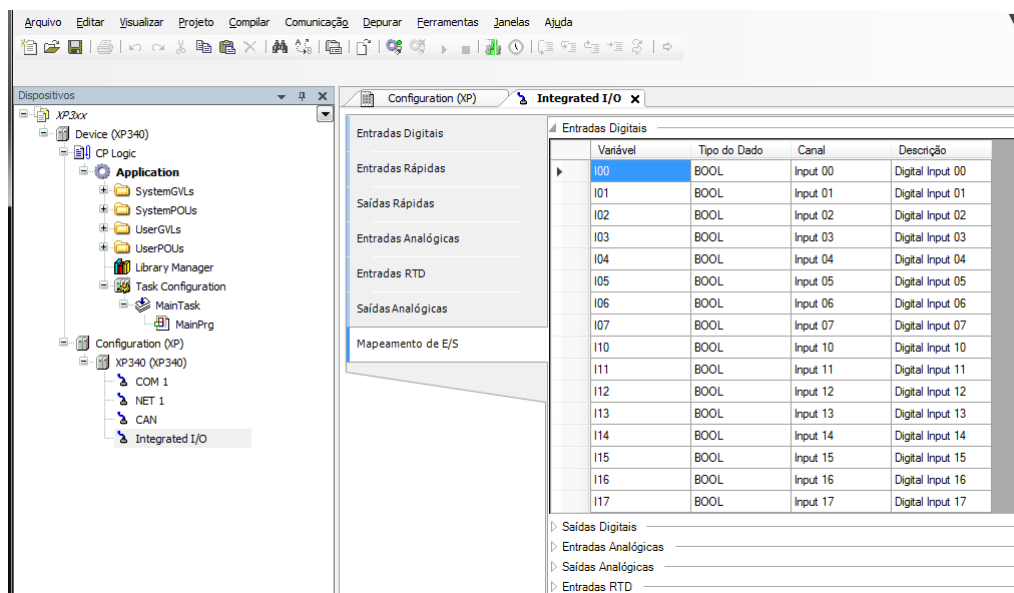


Figura 60: Mapeamento de tags para E/S

5.7. Protocolos de Comunicação

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:

	XP300	XP315	XP325	XP340	XP350	XP351
Pontos Mapeados	20480					
Mapeamentos (Por Instância / Total)	5120 / 20480					
Requisições	512					
NETs – Instâncias Servidor ou Cliente	4					
COM (n) – Instâncias Mestre ou Escravo	1					
Centros de Controle	-	-	-	3	-	-

Tabela 70: 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.

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 ou MODBUS Servidor).

As limitações do protocolo MODBUS por mapeamentos simbólicos podem ser visualizadas na tabela abaixo.

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 71: 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.

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 72: 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 70).

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.

5.7.1. 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	Execução	Após um breakpoint na MainPrg
MODBUS Symbol	Slave/Server	✓	✓	✓	✓	✓
	Master/Client	✗	✗	✗	✓	✓
MODBUS	Slave/Server	✗	✗	✗	✓	✗
	Master/Client	✗	✗	✗	✓	✓
SOE (DNP3)	Outstation	✓	✓	✓	✓	✓
IEC 60870-5-104	Server	✗	✗	✗	✓	✓
EtherCAT	Master	✓	✓	✗	✓	✓
OPC DA	Server	✓	✓	✓	✓	✓
OPC UA	Server	✓	✓	✓	✓	✓
SNTP	Client	✓	✓	✓	✓	✓
HTTP	Server	✓	✓	✓	✓	✓
SNMP	Agent	✓	✓	✓	✓	✓
EtherNet/IP	Scanner	✓	✓	✗	✓	✗
	Adapter	✗	✓	✗	✓	✗

Tabela 73: Comportamento dos Protocolos x Estados da UCP

Notas:

Símbolo ✓: O protocolo permanece ativo e operando normalmente.

Símbolo ✗: O protocolo está desabilitado.

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

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

5.7.2. 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](#).

5.7.3. Fila de Eventos da UCP

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

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

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

ATENÇÃO

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

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

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

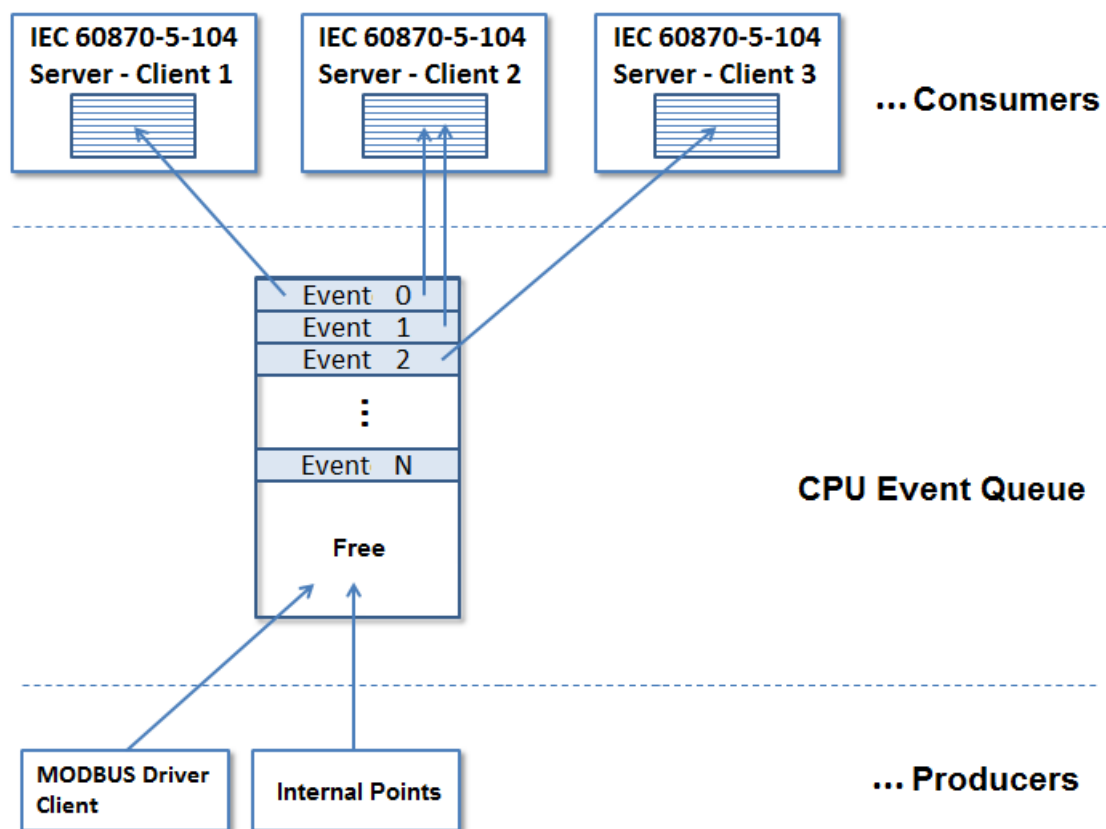


Figura 61: Fila de Eventos da UCP

5.7.3.1. 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.

5.7.3.2. Princípios de Funcionamento da Fila

Uma vez armazenado na fila da UCP, cada evento é transmitido para o consumidor que possuir 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.

5.7.3.2.1. 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)

5.7.3.3. 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

5.7.4. 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 74: 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 75: 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 76: 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 78
sValue	STRUCT	Valor recebido em um select, quando recebido um comando de seleção com valor. Os parâmetros desta estrutura estão descritos na Tabela 81

Tabela 77: Parâmetros *sSelectParameters*

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

Tabela 78: Parâmetros sSelectConfig

Parâmetro	Tipo	Descrição
sOperateConfig	STRUCT	Configuração do comando de seleção recebido. Os parâmetros desta estrutura estão descritos na Tabela 80
sValue	STRUCT	Campo de valor referente ao comando de operação recebido. Os parâmetros desta estrutura estão descritos na Tabela 81

Tabela 79: 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 80: 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 82 a Tabela 86
sDoublePoint	STRUCT	
sIntegerStatus	STRUCT	
sEnumeratedStatus	STRUCT	
sAnalogueValue	STRUCT	

Tabela 81: 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 87.

Tabela 82: 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 87.

Tabela 83: Parâmetros sDoublePoint

Parâmetro	Tipo	Descrição
diValue	DINT	Valor de operação do ponto.

Tabela 84: Parâmetros sIntegerStatus

Parâmetro	Tipo	Descrição
dwValue	DWORD	Valor de operação do ponto.

Tabela 85: 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 86: 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 87: 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*.

5.7.5. MODBUS MESTRE RTU

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*).

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. Para realizar esse procedimento, consultar a seção [Inserindo uma Instância de Protocolo](#).
- Configurar a interface serial, escolhendo a taxa de transmissão, a paridade, bits de parada do canal e demais configurações através de um duplo clique sobre o canal serial COM 1. Consultar a seção [Configuração da Interface Serial](#).

5.7.5.1. 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 62.
- 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 63.
- Adicionar e configurar os mapeamentos MODBUS na aba Mapeamentos conforme Figura 64, 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 65, 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.

5.7.5.1.1. 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:

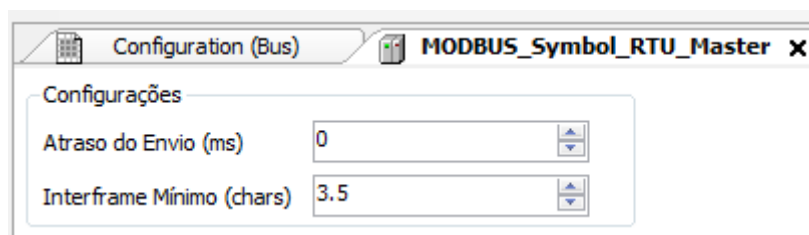


Figura 62: Tela de Configuração Parâmetros Gerais MODBUS RTU Mestre

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 88: 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 89: 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.

5.7.5.1.2. Configuração dos Dispositivos – Configuração por Mapeamento Simbólico

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

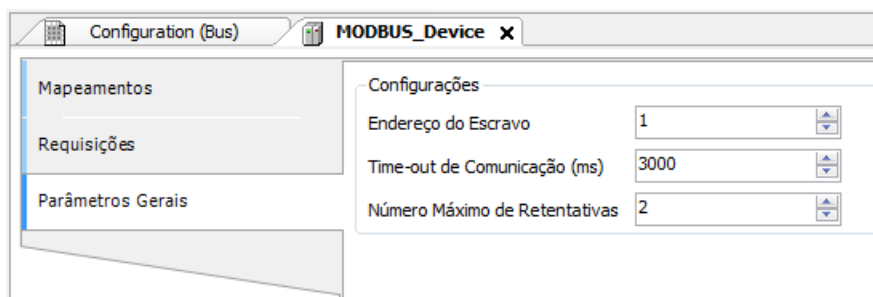


Figura 63: Tela de Configurações dos Parâmetros Gerais do Dispositivo

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 90: 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.

5.7.5.1.3. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico

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

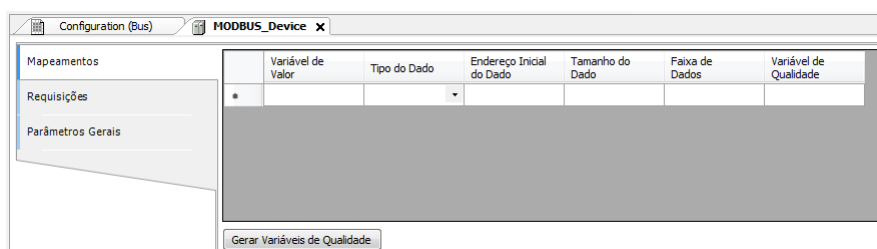


Figura 64: Tela de Mapeamentos de Dados MODBUS

Configuração	Descrição	Padrão	Opções
Variável de Valor	Nome da variável simbólica	-	Nome de uma variável declarada em um programa ou GVL
Tipo 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

Configuração	Descrição	Padrão	Opções
Faixa de Dados	Faixa de endereços do dado configurado	-	-

Tabela 91: 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 92: 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.

5.7.5.1.4. Configuração das Requisições – Configuração por Mapeamento Simbólico

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

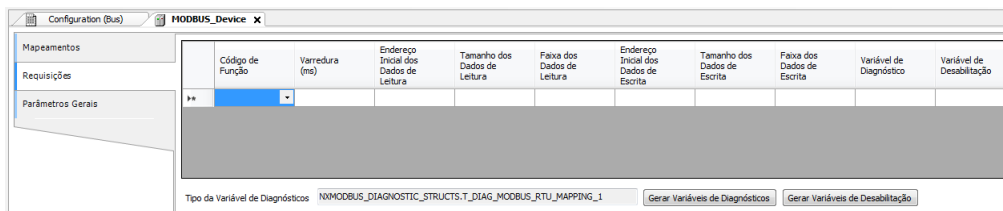


Figura 65: Tela de Requisições de dados MODBUS Mestre

Configuração	Descrição	Padrão	Opções
Código de Função	Tipo de função MODBUS	-	01 – Leitura de Coils 02 – Leitura de Input Status 03 – Leitura de Holding Registers 04 – Leitura de Input Registers 05 – Escrita de Um Coil 06 – Escrita de Um Register 15 – Escrita de Múltiplos Coils 16 – Escrita de Múltiplos Registers 22 – Escrita Mascarada de Register 23 – Leitura/Escrita de Múltiplos Registers
Varredura (ms)	Período de comunicação (ms)	100	0 a 3600000
Endereço Inicial do Dado de Leitura	Endereço inicial dos dados de leitura MODBUS	-	1 a 65536
Tamanho dos Dados de Leitura	Tamanho dos dados de leitura MODBUS	-	Depende da função utilizada
Faixa dos Dados de Leitura	Faixa de endereço dos dados de leitura MODBUS	-	0 a 2147483646
Endereço Inicial do Dado de Escrita	Endereço inicial dos dados de escrita MODBUS	-	1 a 65536
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 93: 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 94: 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 95.

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.

T_DIAG_MODBUS_RTU_MAPPING_1.*	Tamanho	Descrição
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 113 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 95: 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.

Código	Enumerável	Descrição
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 96: 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.

5.7.6. MODBUS ESCRAVO RTU

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. Este protocolo somente está disponível quando a UCP estiver em modo de execução (Modo *Run*).

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. Para realizar esse procedimento, consultar a seção [Inserindo uma Instância de Protocolo](#).
- Configurar a interface serial, escolhendo a velocidade de comunicação, a paridade, os bits de parada do canal, entre outros. Consultar a seção [Configuração da Interface Serial](#).

5.7.6.1. 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.

5.7.6.1.1. 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.

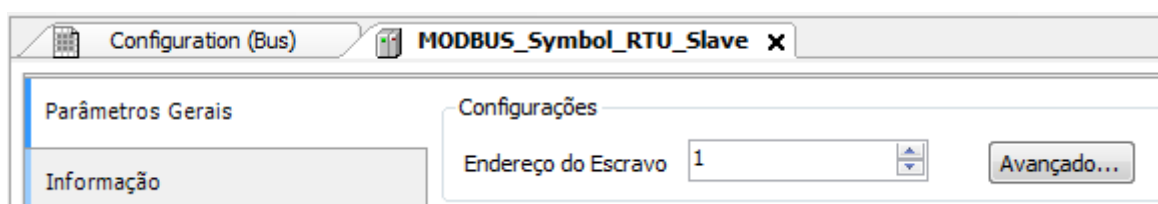


Figura 66: Configuração do Escravo

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

Tabela 97: 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.

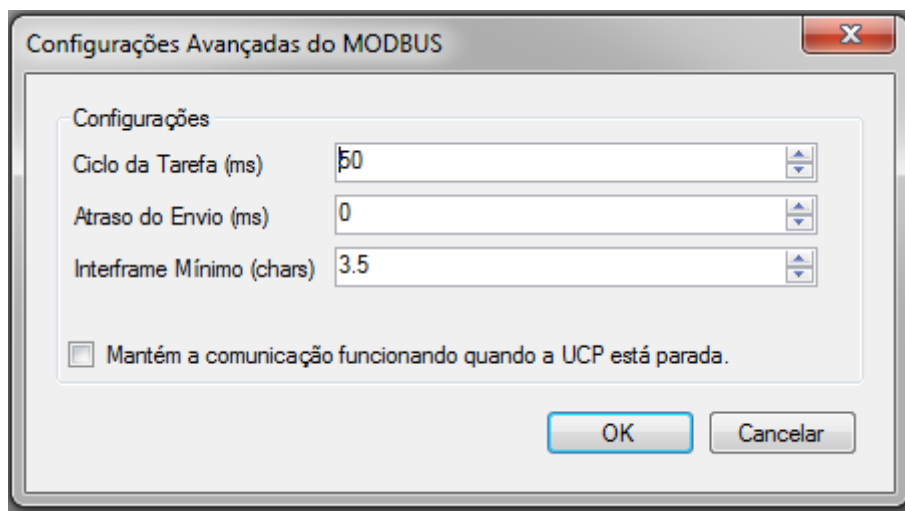


Figura 67: Configurações Avançadas MODBUS do Escravo

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 98: 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:

T_DIAG_MODBUS_RTU_SLAVE_1.*	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.
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).

T_DIAG_MODBUS_RTU_SLAVE_1.*	Tamanho	Descrição
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 99: 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.

5.7.6.1.2. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico

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

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

Figura 68: 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 100: 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.

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

Tabela 101: 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.

5.7.7. 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 70.

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

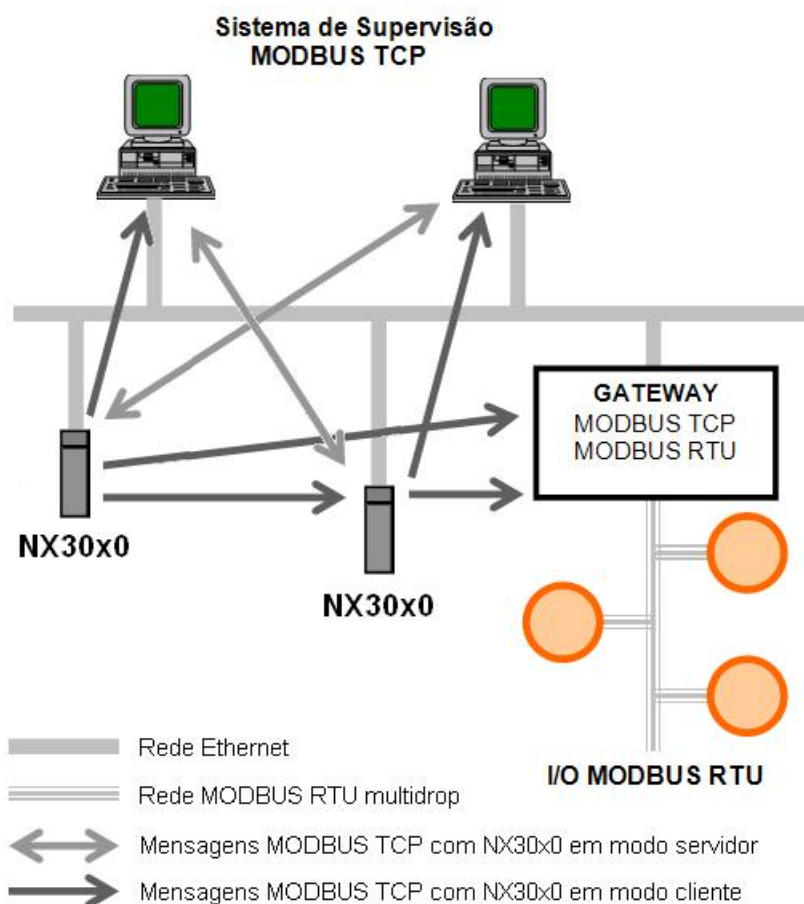


Figura 69: Rede de Comunicação MODBUS TCP

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.

Tipo de Dado	Tamanho [bits]	Descrição
Input Register	16	Entrada analógica que pode ser apenas lida.

Tabela 102: 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 103: 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 da Interface Ethernet](#).

5.7.8. 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*).

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](#).

5.7.8.1. 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.

5.7.8.1.1. 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:

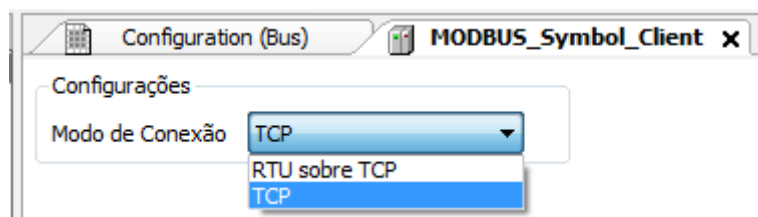


Figura 70: Tela de Configuração Parâmetros Gerais MODBUS Cliente

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

Tabela 104: 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 105: Diagnósticos MODBUS Cliente

Nota:

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

5.7.8.1.2. Configuração dos Dispositivos – Configuração por Mapeamento Simbólico

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

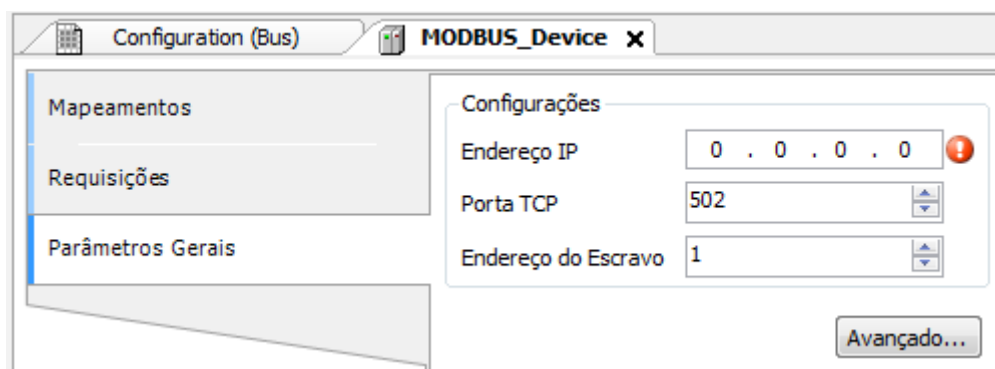


Figura 71: Tela de Configurações dos Parâmetros Gerais do Dispositivo

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 106: 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 um 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 107: 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.

5.7.8.1.3. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico

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

Variável de Valor	Tipo do Dado	Endereço Inicial do Dado	Tamanho do Dado	Faixa de Dados	Variável de Qualidade
*					

Gerar Variáveis de Qualidade

Figura 72: 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 108: 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 109: 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.

5.7.8.1.4. Configuração das Requisições – Configuração por Mapeamento Simbólico

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

Figura 73: Tela de Requisições de dados MODBUS

Configuração	Descrição	Padrão	Opções
Código de Função	Tipo de função MODBUS	-	01 – Leitura de Coils 02 – Leitura de Input Status 03 – Leitura de Holding Registers 04 – Leitura de Input Registers 05 – Escrita de Um Coil 06 – Escrita de Um Register 15 – Escrita de Múltiplos Coils 16 – Escrita de Múltiplos Registers 22 – Escrita Mascarada de Register 23 – Leitura/Escrita de Múltiplos Registers
Varredura (ms)	Período de comunicação (ms)	100	0 a 3600000
Endereço Inicial do Dado de Leitura	Endereço inicial dos dados de leitura MODBUS	-	1 a 65536
Tamanho dos Dados de Leitura	Tamanho dos dados de leitura MODBUS	-	Depende da função utilizada
Faixa dos Dados de Leitura	Faixa de endereço dos dados de leitura MODBUS	-	0 a 2147483646
Endereço Inicial do Dado de Escrita	Endereço inicial dos dados de escrita MODBUS	-	1 a 65536
Tamanho dos Dados de Escrita	Tamanho dos dados de escrita MODBUS	-	Depende da função utilizada
Faixa dos Dados de Escrita	Faixa de endereço dos dados de escrita MODBUS	-	0 a 2147483647
Variável de Diagnóstico	Nome da variável de diagnóstico	-	Nome de uma variável declarada em um programa ou GVL
Variável de Desabilitação	Variável utilizada para desabilitar a relação MODBUS	-	Campo destinado a variável simbólica utilizada para desabilitar, individualmente, as requisições MODBUS configuradas. Esta variável deve ser do tipo BOOL. A variável pode ser simples ou elemento de array e pode estar em estruturas.

Tabela 110: 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 111: 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 95.

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.

T_DIAG_MODBUS_ETH_MAPPING_1.*	Tamanho	Descrição
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 113 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 112: 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.

Código	Enumerável	Descrição
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 113: 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.

5.7.8.2. 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.

5.7.9. 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*).

O procedimento para inserir uma instância de protocolo é encontrado com detalhes no Manual de Utilização do Mastertool.

5.7.9.1. 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.

5.7.9.1.1. 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.

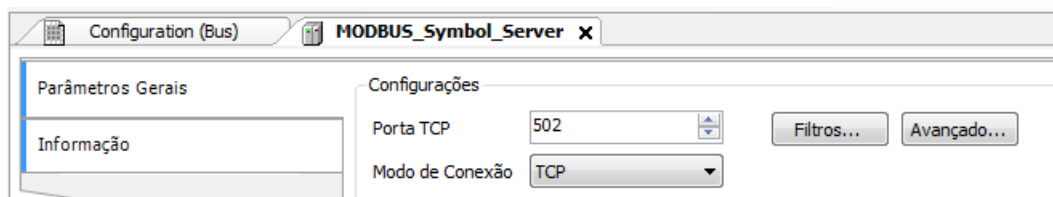


Figura 74: Tela de Configuração Parâmetros Gerais MODBUS Servidor

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

Tabela 114: 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 115: 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*.

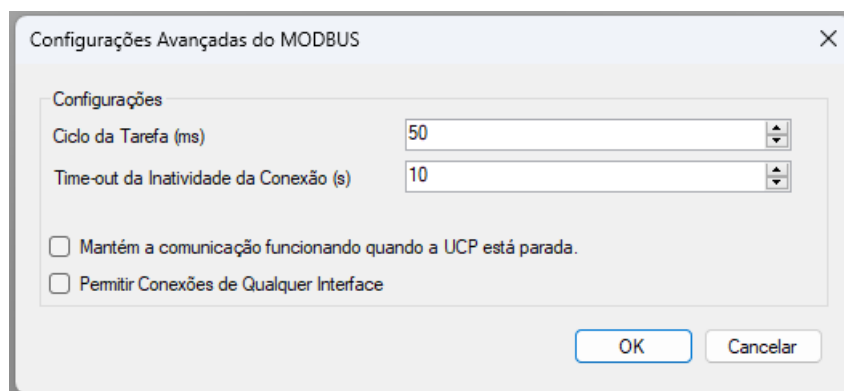


Figura 75: Janela de Configurações Avançadas do MODBUS Servidor

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 116: 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.

5.7.9.1.2. 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.

T_DIAG_MODBUS_ETH_SERVER_1.*	Tamanho	Descrição
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 – timeout (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 117: 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.

5.7.9.1.3. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico

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

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

Figura 76: 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

Configuração	Descrição	Padrão	Opções
Faixa de Dados	Faixa de endereços do dado configurado	-	-

Tabela 118: 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.

5.7.10. 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).

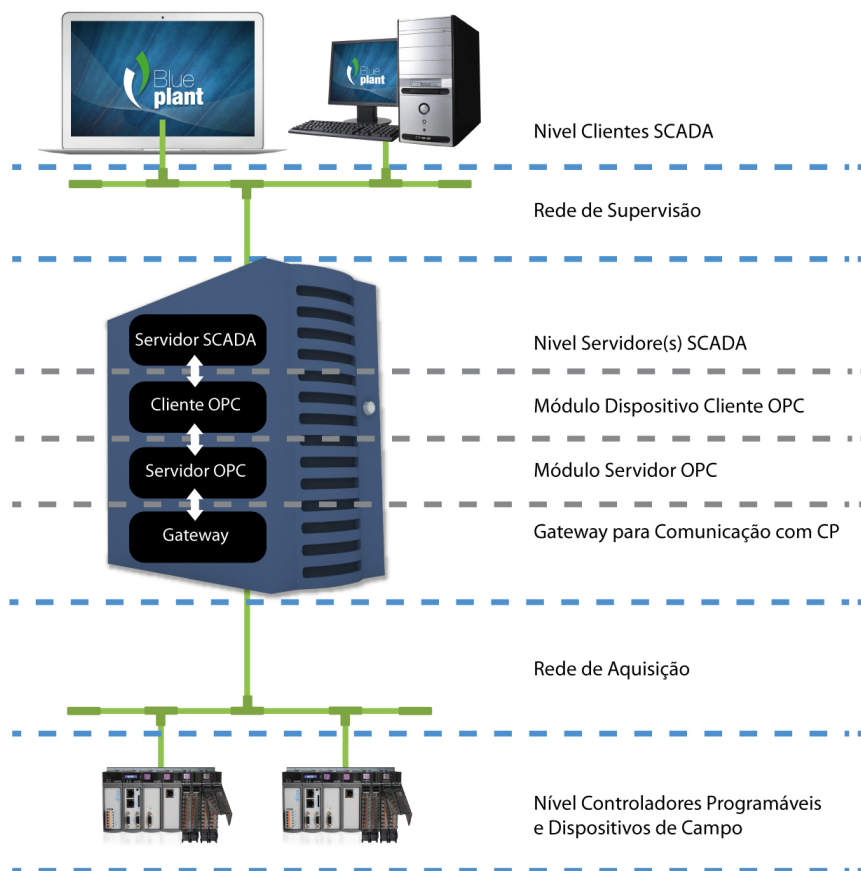


Figura 77: Arquitetura OPC DA

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 119: 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.

5.7.10.1. 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.

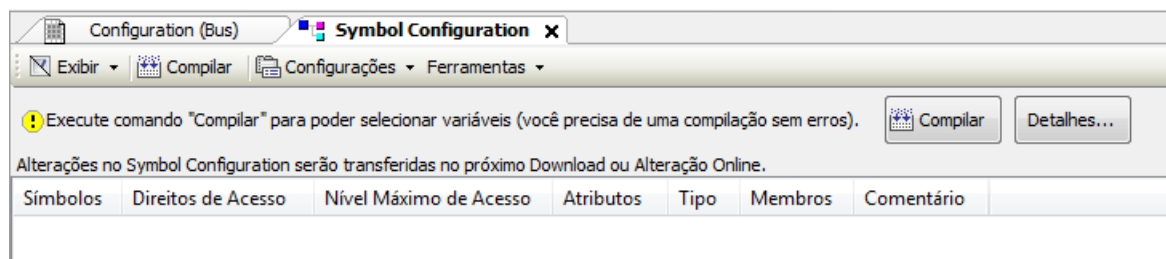


Figura 78: Objeto Symbol Configuration

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 120: 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 78 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.

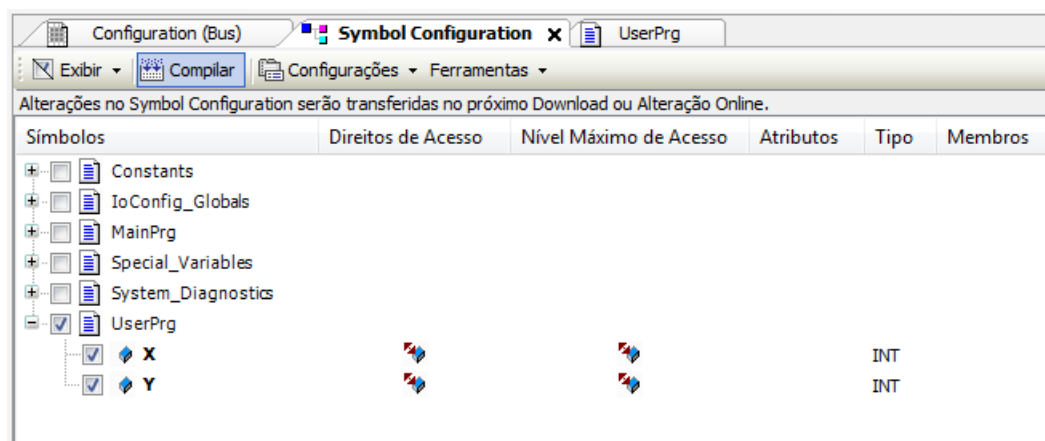


Figura 79: Selecionando Variáveis na Symbol Configuration

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.

5.7.10.2. 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.

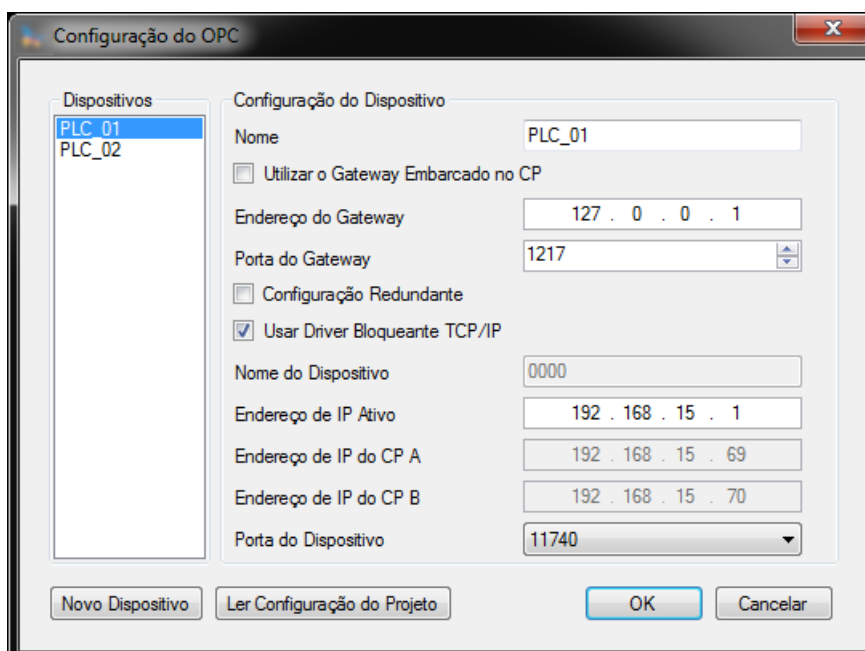


Figura 80: Configuração do OPC DA Servidor

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 121: 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.

5.7.10.2.1. Importando uma Configuração do Projeto

Utilizando o botão *Ler Configuração do Projeto*, conforme a Figura 80, 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.

5.7.10.3. 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 122: 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 123: Descrição do valor da Qualidade OPC DA

A Tabela 123 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.

5.7.10.4. 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 124: 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 124.

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.

5.7.10.5. 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*.

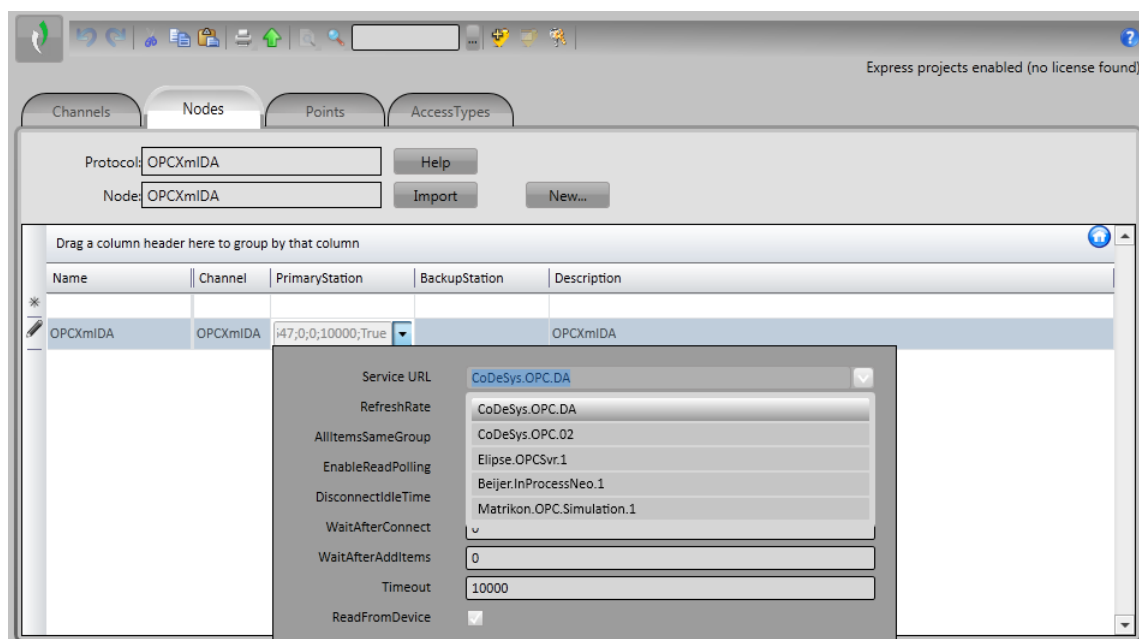


Figura 81: Selecionando Servidor OPC DA na Configuração do Cliente

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.

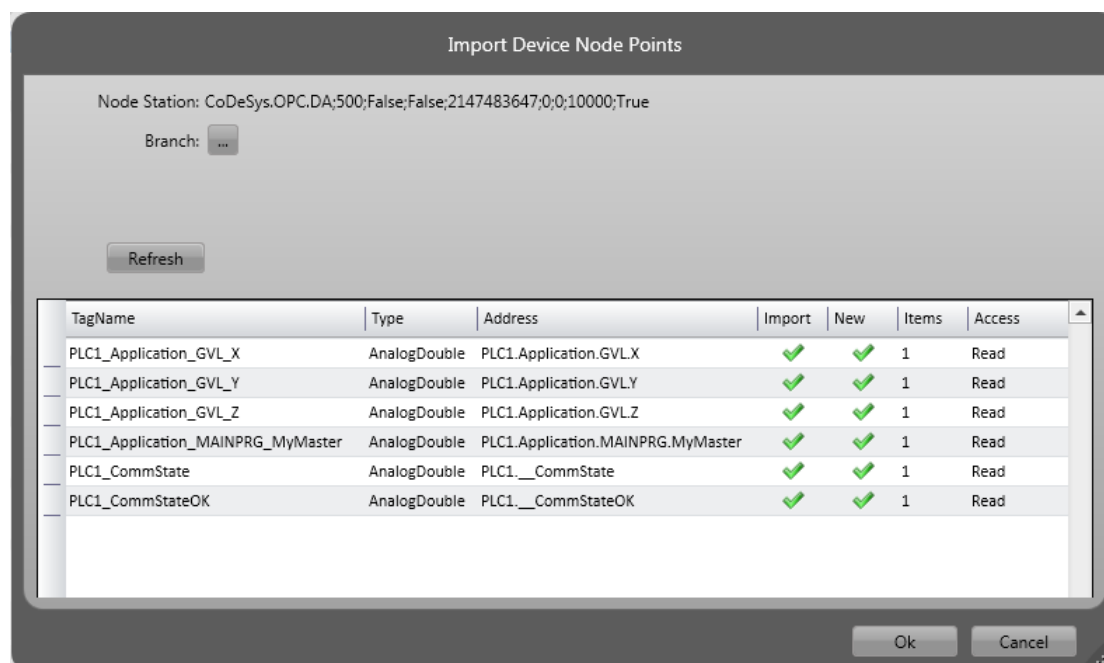


Figura 82: Lista de Símbolos Consultados pelo Cliente OPC DA

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.

5.7.11. 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.

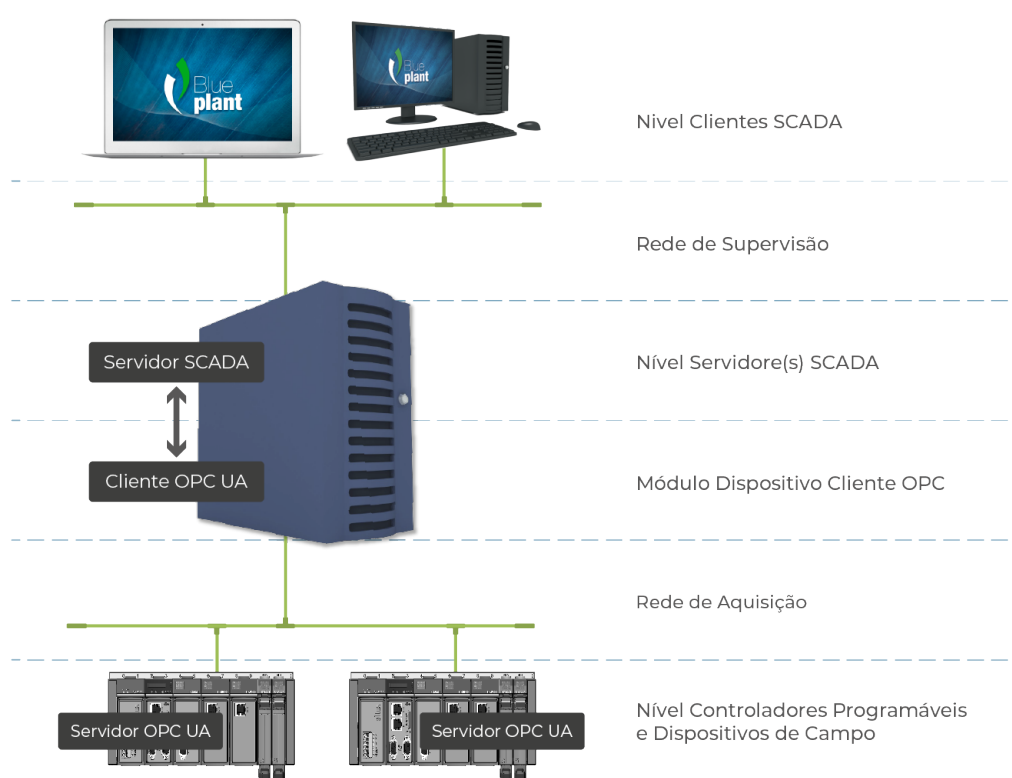


Figura 83: Arquitetura Típica OPC UA

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.

Papel	Descrição
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 125: 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.

5.7.12. 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.

5.7.12.1. 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 seleccionar os arquivos de descrição de dispositivo (*EtherCAT XML Device Description / ESI EtherCAT Slave Information*), fornecido com o hardware. As descrições para os Escravos estão disponíveis como arquivos XML (tipo de arquivo: *.xml).

Um Mestre EtherCAT pode ser adicionado à *Árvore de Dispositivos* através do comando *Acréscetar Dispositivo*, através do menu de contexto dos conectores NET da UCP.

Abaixo de um mestre, um ou mais escravos podem ser inseridos, seleccionando um Mestre EtherCAT e executando o comando *Acréscetar Dispositivo* (menu de contexto do Mestre EtherCAT) ou executando o comando *Procurar Dispositivos*.

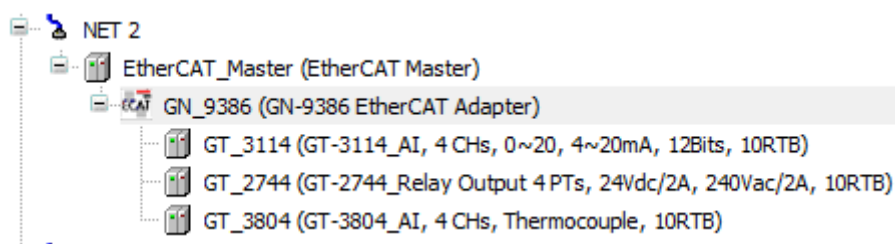


Figura 84: Exemplo de Configuração EtherCAT

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.

5.7.12.1.1. 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 seleccionado, 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.

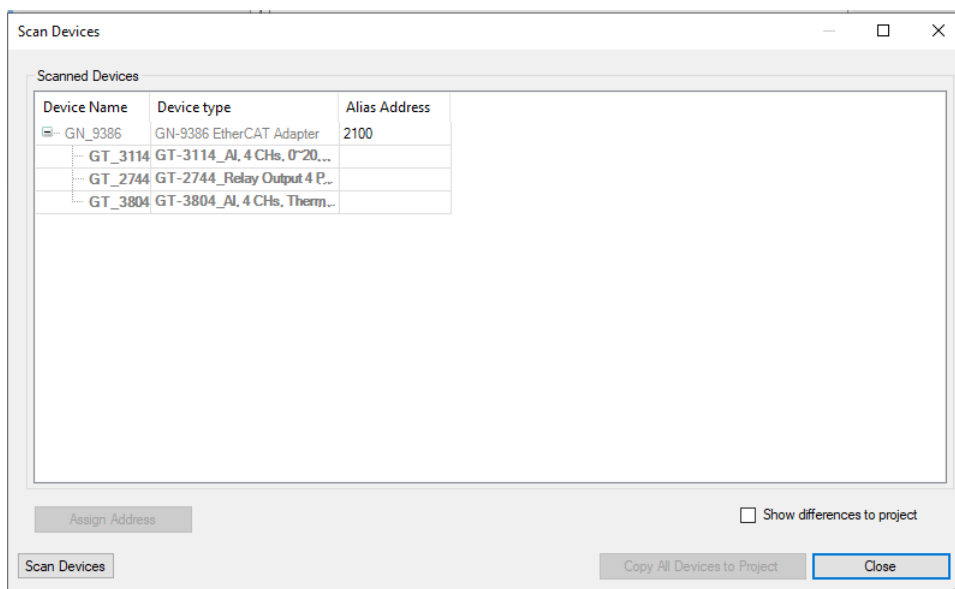


Figura 85: Diálogo Procurar Dispositivos para EtherCAT

5.7.12.2. 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*.

5.7.12.2.1. EtherCAT Master - General

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

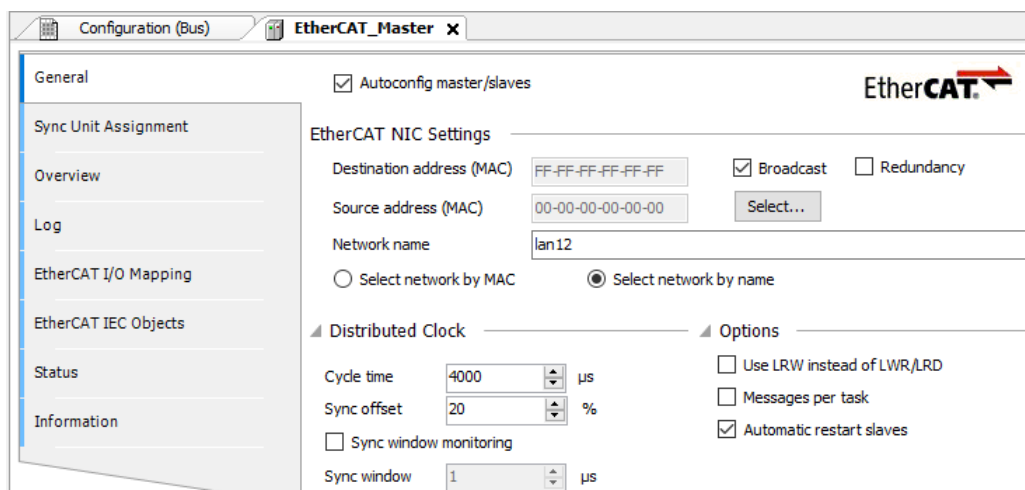


Figura 86: Diálogo de Configuração do Mestre EtherCAT

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 126: 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.

5.7.12.2.2. 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.

5.7.12.2.3. 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.

5.7.12.2.4. 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.

5.7.12.2.5. 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.

5.7.12.2.6. 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*.

5.7.12.3. 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*.

5.7.12.3.1. 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 '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' (checkboxes), 'Sync unit cycle' (dropdowns), and 'Cycle time'/'Shift time' (spinners); 'Startup Checking' with 'Check vendor ID', 'Check product ID', and 'Check revision number' (checked checkboxes); 'DC Cyclic Unit Control' with 'Assign to Local µC' (checkbox) and 'Cyclic unit', 'Latch unit 0', 'Latch unit 1' (checkboxes); 'Watchdog' with 'Set multiplier', 'Set PDI watchdog', and 'Set SM watchdog' (checkboxes); and 'Identification' with 'Disabled' (selected radio button), 'Configured station alias', 'Explicit device identification', and 'Data Word' (radio buttons).

Figura 87: 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 127: 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.

5.7.12.3.2. 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.

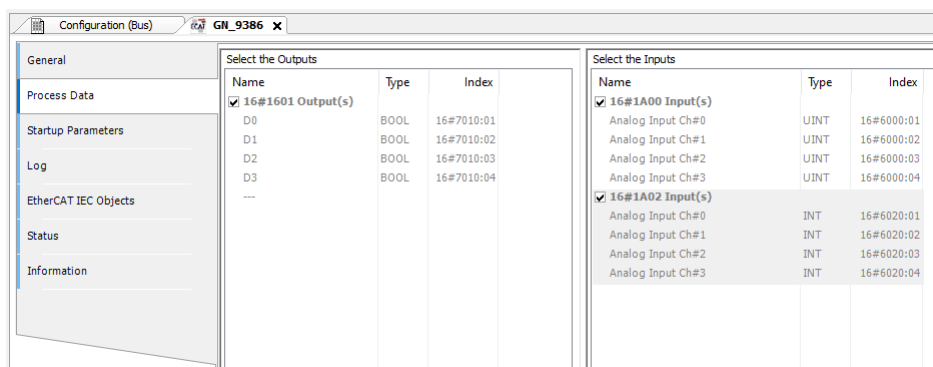


Figura 88: Diálogo Dados do Processo

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.

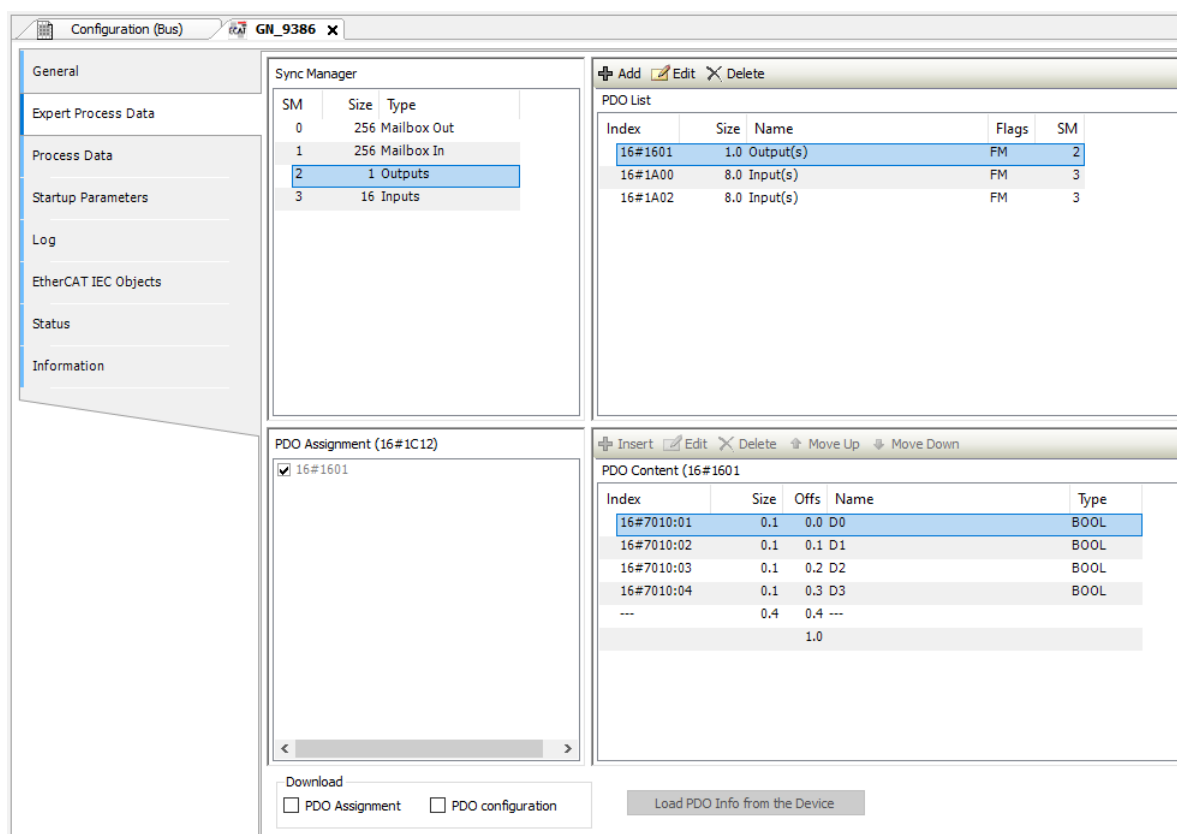


Figura 89: Diálogo Dados de Processo Avançados

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.

5.7.12.3.3. EtherCAT Slave - Edit PDO List

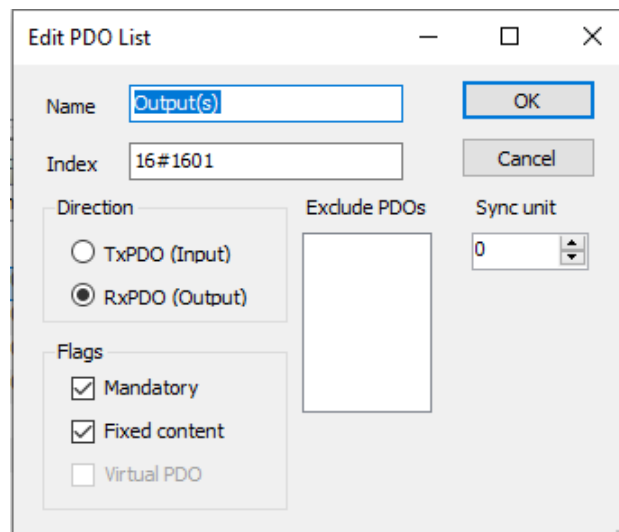


Figura 90: Diálogo Editar Lista PDO

Este diálogo é aberto através do menu de contexto da área Lista PDO, apresentada na Figura 89. 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.

5.7.12.3.4. 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*.

5.7.12.3.5. 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*.

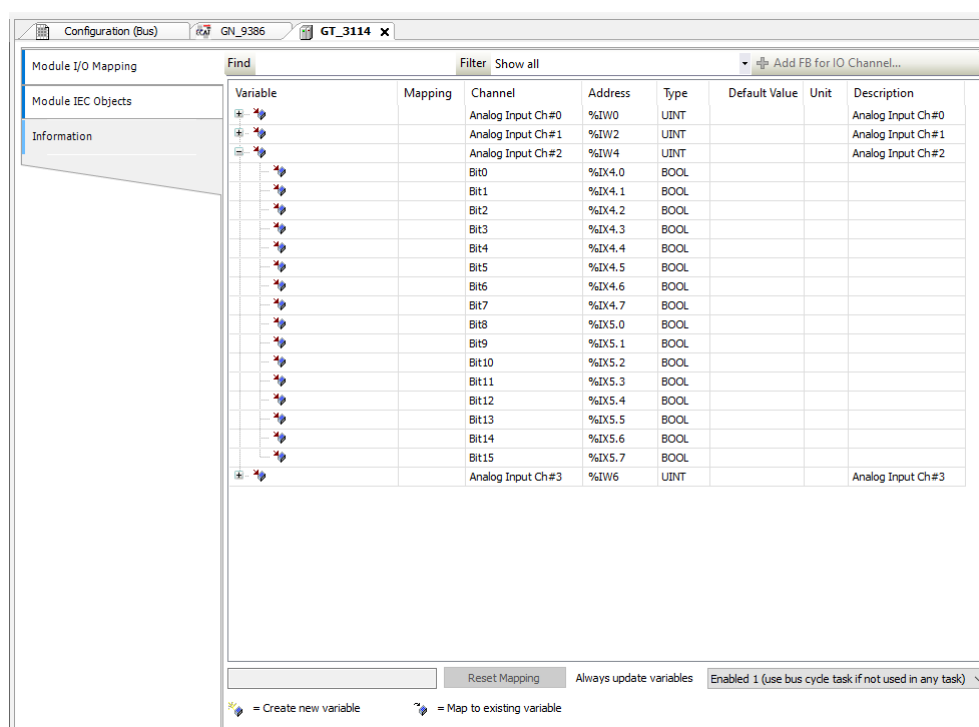


Figura 91: Diálogo Mapeamento de E/S Escravo

5.7.12.3.6. 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*.

5.7.13. 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.

5.7.13.1. Interface 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*.

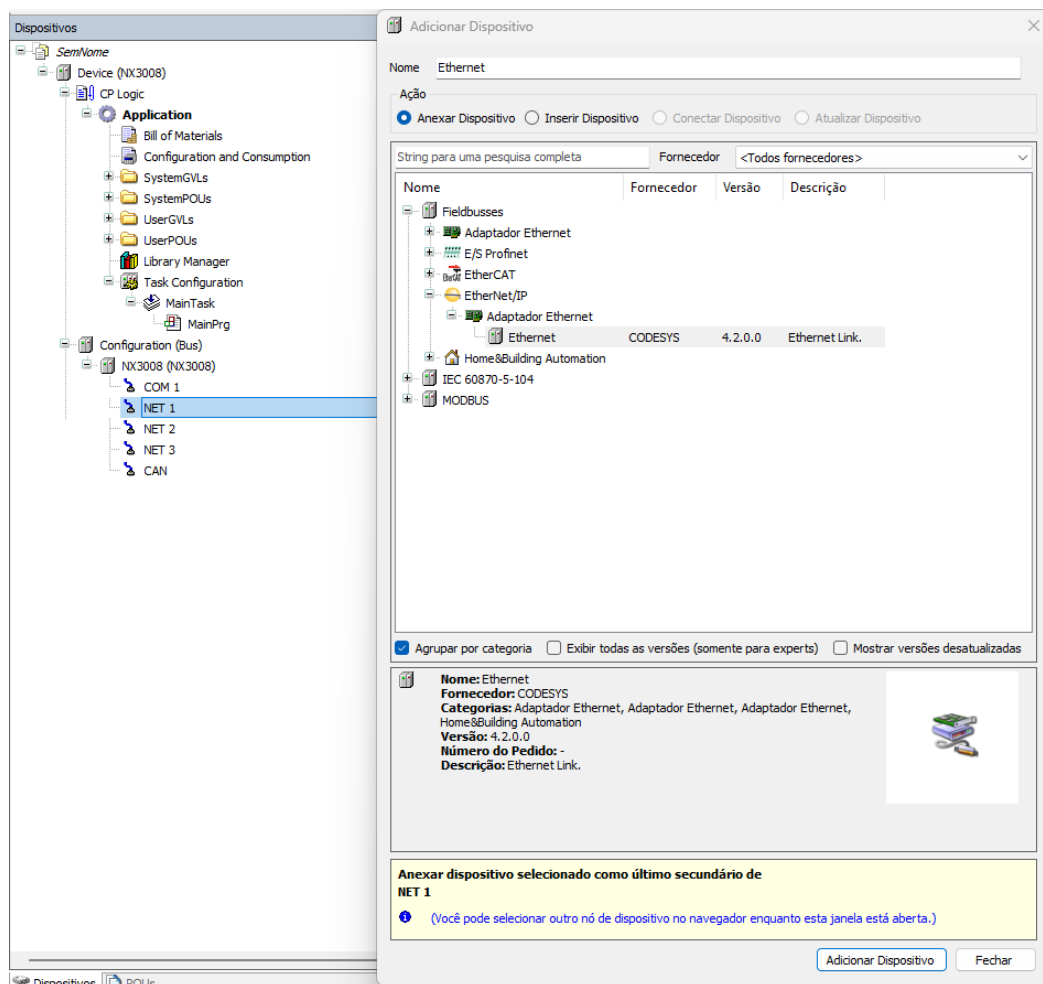


Figura 92: Adicionando um Adaptador Ethernet

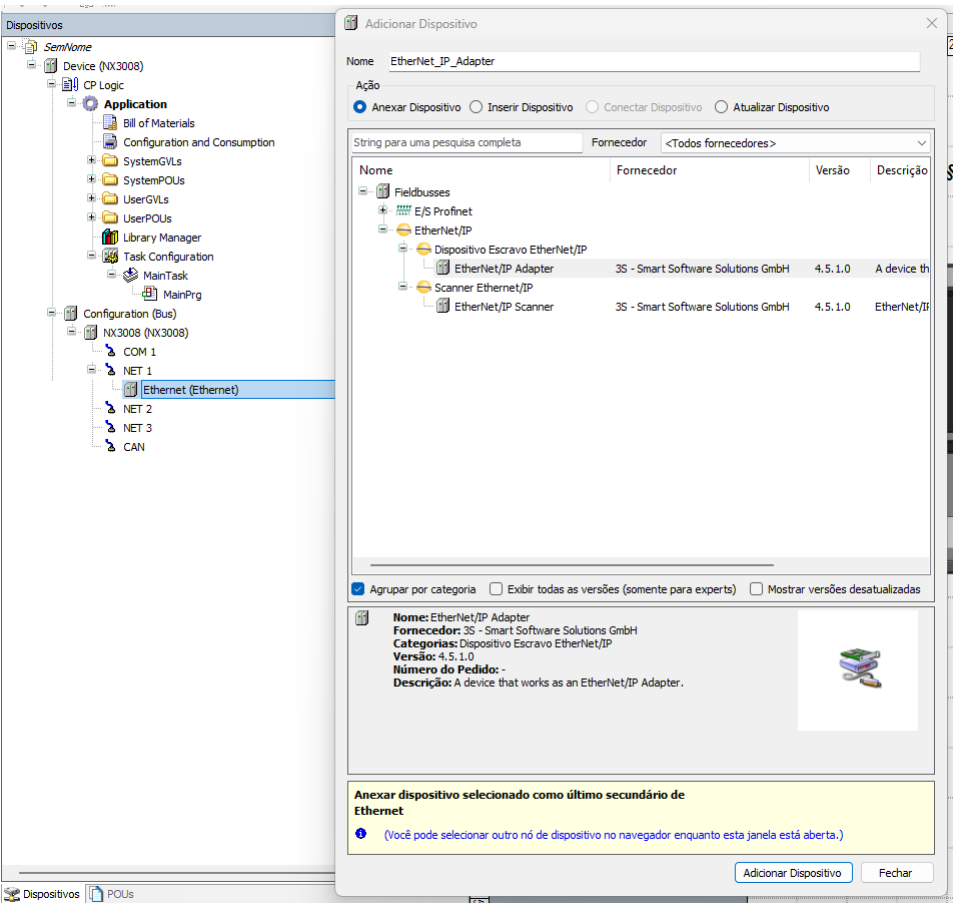


Figura 93: Adicionando um Adapter ou Scanner EtherNet/IP

5.7.13.2. Configuração do Scanner EtherNet/IP

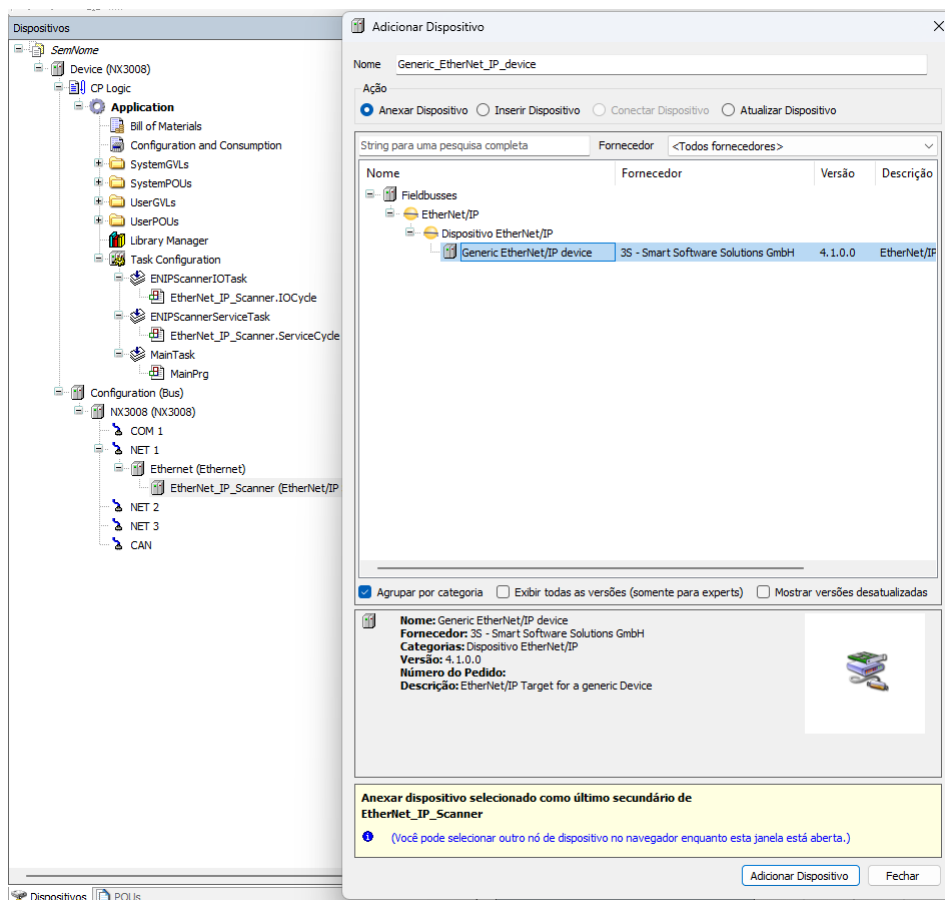


Figura 94: Adicionando um Adapter EtherNet/IP Sob o Scanner

5.7.13.2.1. 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".

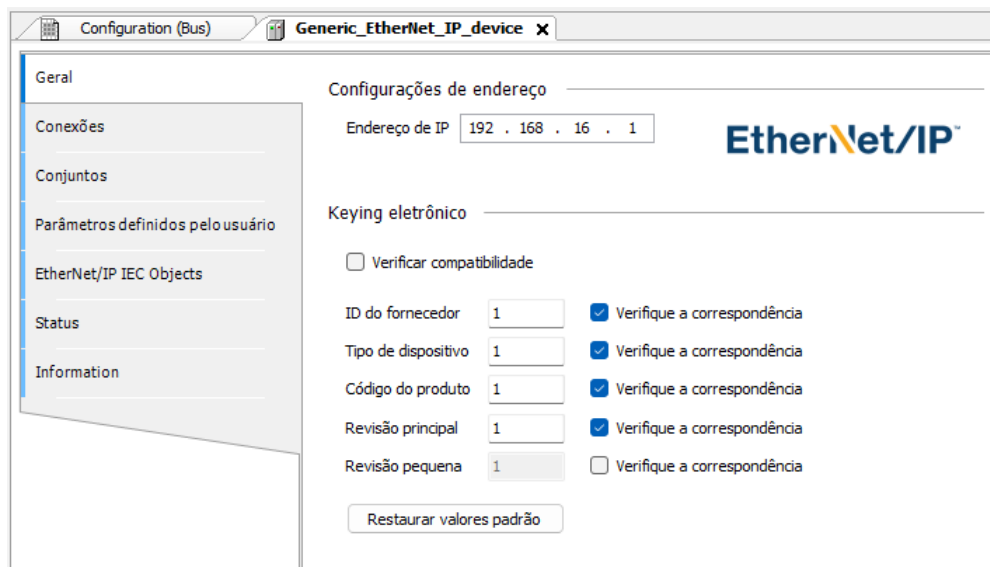


Figura 95: Aba Geral - EtherNet/IP

5.7.13.2.2. 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.

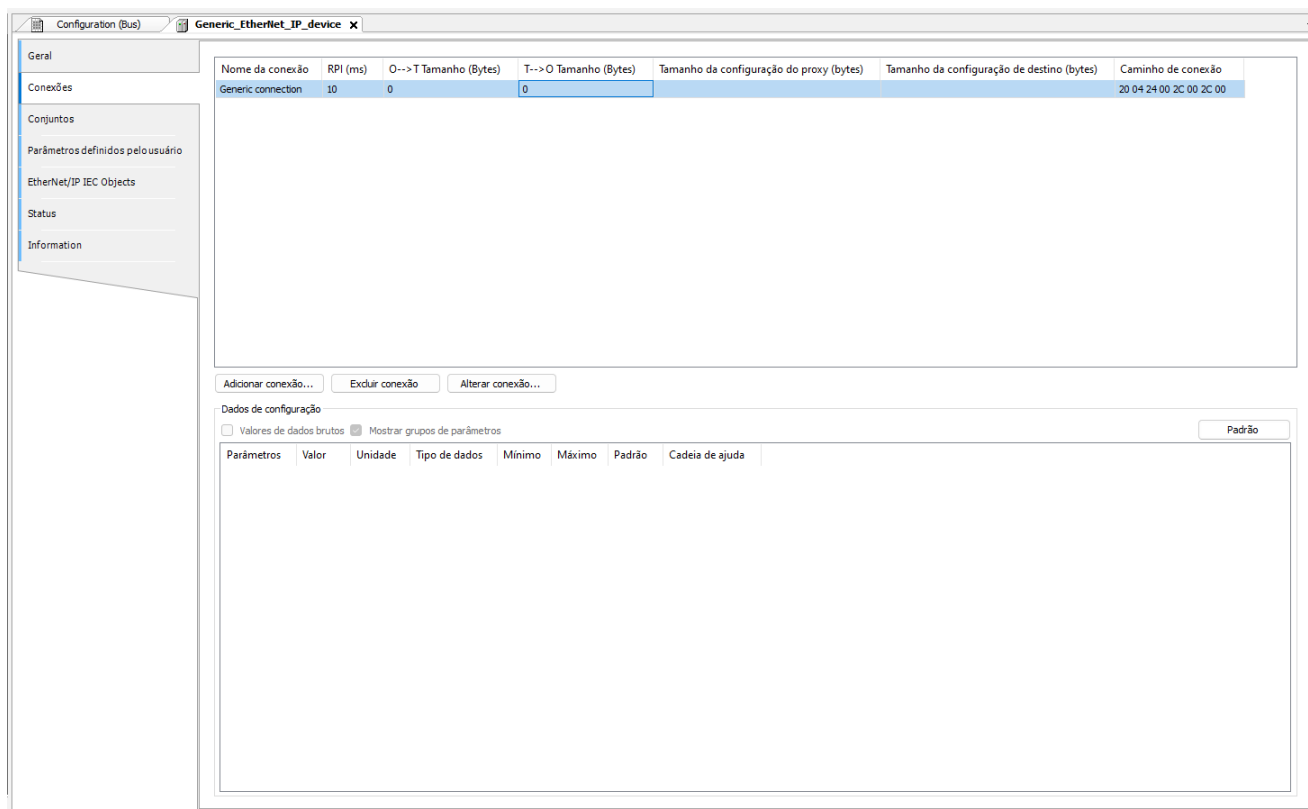


Figura 96: Aba Conexões - EtherNet/IP

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 128: 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 97: Janela de Novas Conexões EtherNet/IP

5.7.13.2.3. 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 98: 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 129: 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 130: Janela “Adicionar Entrada/Saída” – EtherNet/IP

5.7.13.2.4. EtherNet/IP: Mapeamento de E/S

Aba *Mapeamento de E/S* mostra, na coluna *Variável*, o nome das instâncias de *Adapter*, automaticamente geradas, abaixo de *Objetos IEC*. Desta forma, a instância pode ser acessada pela aplicação. Aqui as variáveis do projeto são mapeadas para as entradas e saídas do adapter.

5.7.13.3. 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.

5.7.13.3.1. 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.

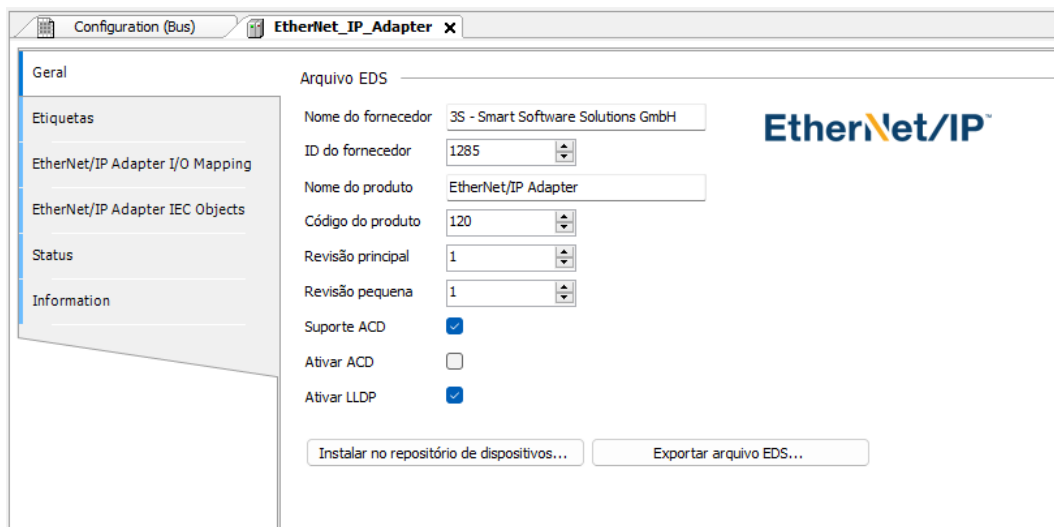


Figura 99: Aba Geral - EtherNet/IP_Adapter

5.7.13.3.2. 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.

5.7.13.4. Configuração do Módulo EtherNet/IP

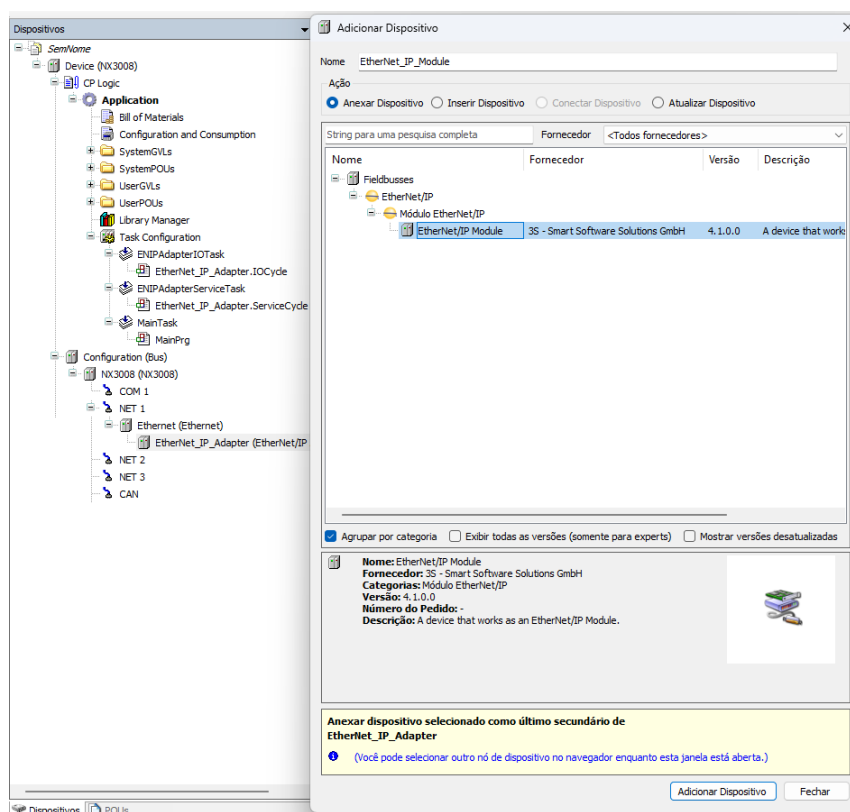


Figura 100: Adicionando um Módulo EtherNet/IP Abaixo do Adapter

5.7.13.4.1. Assemblies

Os parâmetros da aba *Geral* do módulo seguem as mesmas regras descritas nas tabelas 129 e 130.

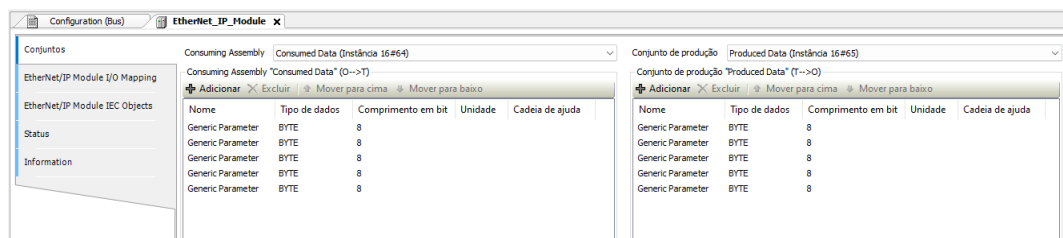


Figura 101: Aba Assemblies Módulo - EtherNet/IP

5.7.13.4.2. 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.

5.7.14. 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 da Interface 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.

5.7.14.1. Tipos de Dados

A tabela abaixo mostra o tipo de variável suportada pelas UCPs da Série Nexto para cada um dos tipos de dados do protocolo IEC 60870-5-104.

Tipo do Objeto	Tipos de Variáveis IEC
Single Point Information (M_SP_NA)	BOOL
	BIT
Double Point Information (M_DP_NA)	DBP
Step Position Information (M_ST_NA)	USINT
Measured Value, normalized value (M_ME_NA)	INT
Measured Value, scaled value (M_ME_NB)	INT
Measured Value, short floating point value (M_ME_NC)	INT
	UINT
	DINT
	UDINT
	REAL
Integrated Totals (M_IT_NA)	INT
	DINT
Bitstring Information (M_BO_NA)	DWORD
Single Command (C_SC_NA)	BOOL
	BIT
Double Command (C_DC_NA)	DBP
Regulating Step Command (C_RC_NA)	DBP
Setting Point Command, normalized Value (C_SE_NA)	INT
Setting Point Command, scaled Value (C_SE_NB)	INT
Setting Point Command, short floating point Value (C_SE_NC)	REAL
Bitstring Command (C_BO_NA)	DWORD

Tabela 131: 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

```



```

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
);

```

5.7.14.2. 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.

5.7.14.2.1. 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

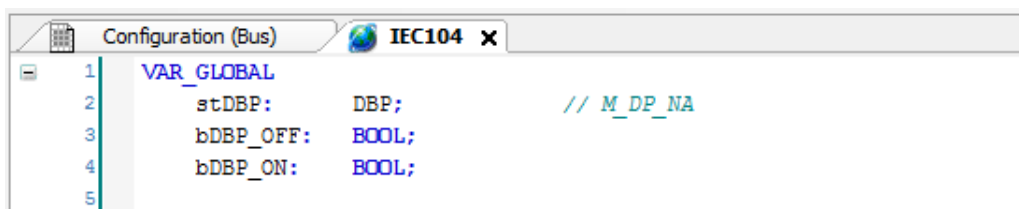


Figura 102: Exemplo de Declaração das Variáveis de Ponto Duplo

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:

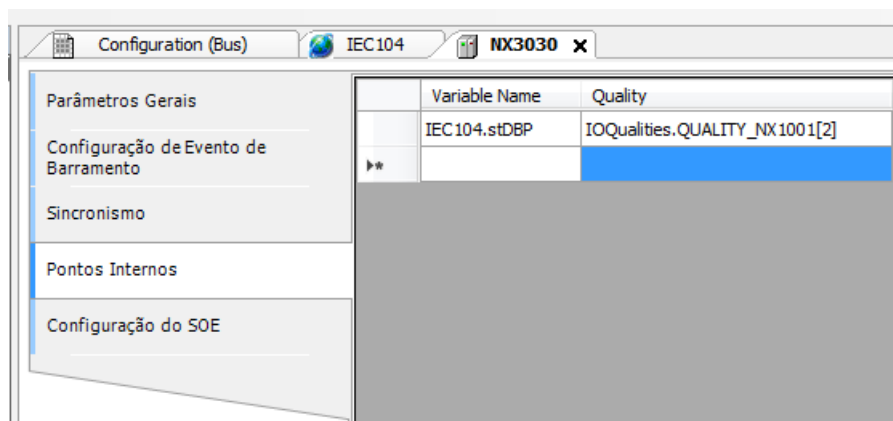


Figura 103: Atribuição das Variáveis de Ponto Duplo aos Pontos Internos

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.

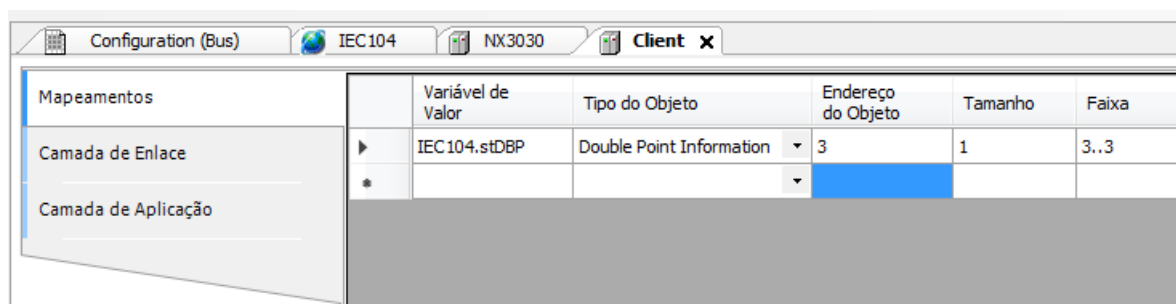


Figura 104: Mapeamento das variáveis de Ponto Duplo no Cliente IEC 60870-5-104

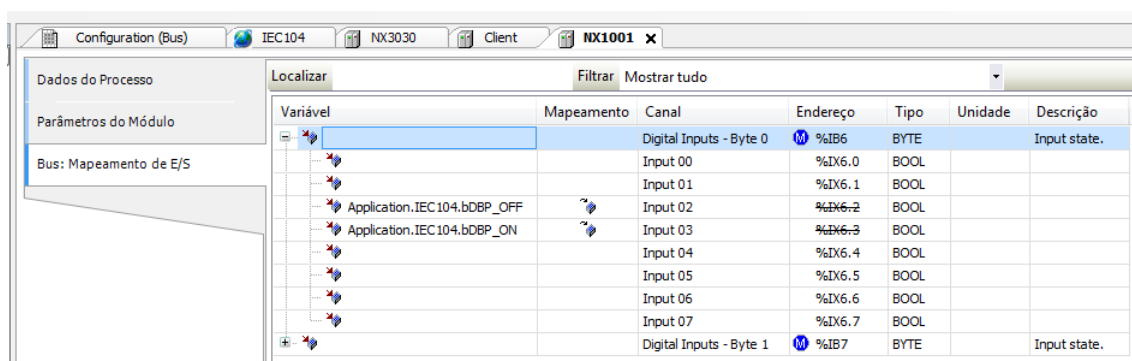


Figura 105: Mapeamento das Variáveis às Entradas do Módulo

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

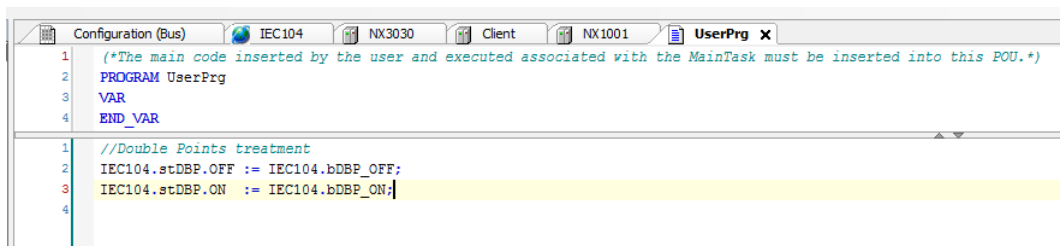


Figura 106: Atribuição dos Valores das Variáveis ao Ponto Duplo

5.7.14.2.2. 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 107: 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;

```

5.7.14.3. 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:

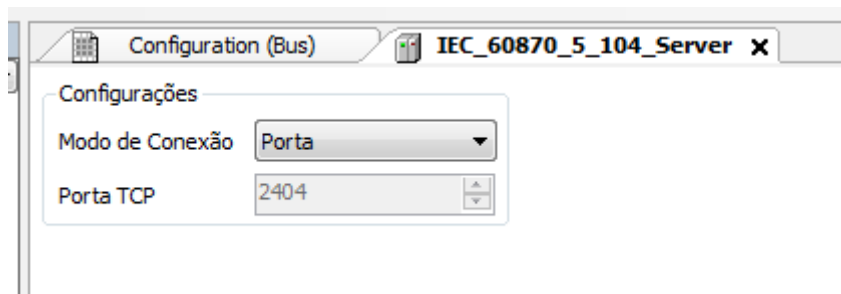


Figura 108: Tela de Parâmetros Gerais do Servidor IEC 60870-5-104

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

5.7.14.4. Mapeamento dos Dados

Para a configuração das relações de dados do Servidor IEC60870-5-104, visualizadas na figura abaixo, seguem os parâmetros descritos na tabela abaixo:



Figura 109: 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

Configuração	Descrição	Padrão	Opções
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
Pulso Longo (ms)	Define o tempo do pulso longo de um comando digital IEC 60870-5-104	2000	1 a 86400000

Tabela 133: 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 134: 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.

5.7.14.5. 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:

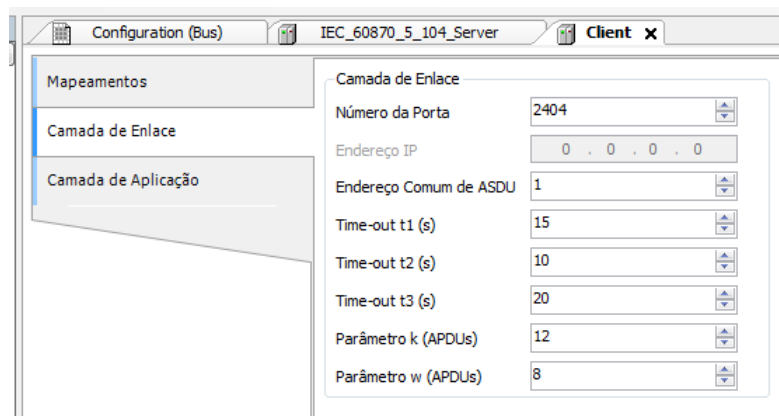


Figura 110: Tela de Configuração da Camada de Enlace do Servidor IEC 60870-5-104

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 135: 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.

5.7.14.6. 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 111: 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 136: 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 137: 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 138: 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).

5.7.14.7. 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 139: Diagnósticos do Servidor IEC 60870-5-104

5.7.14.8. 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)			

Qualificador	Tipo do objeto do protocolo IEC 60870-5-104		
	Single Command	Double Command	Regulating Step Command
Persistent output (persistente)	A saída será ligada ou desligada, e assim permanecerá até novo comando, conforme valor (ON ou OFF) comandado pelo cliente.		

Tabela 140: 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*.

5.7.15. 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
800 kbit/s	50 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 141: Baudrate X Comprimento do Barramento

5.7.15.1. 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 *Acréscitar Dispositivo*. Expanda os itens até encontrar o dispositivo *CANopen_Manager* e clique no botão *Acréscitar*. O dispositivo *CANopen_Manager* aparecerá abaixo da interface CAN, conforme mostrado na figura a seguir:

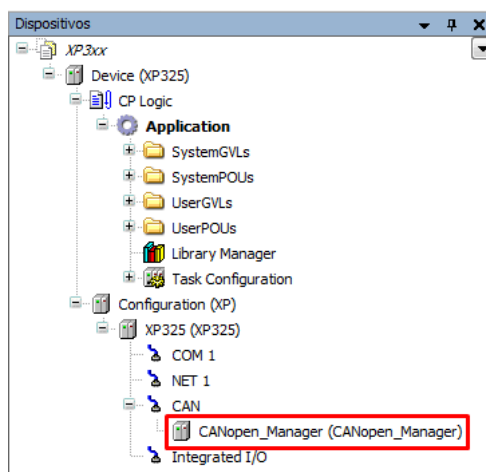


Figura 112: Adicionando o CANopen Manager

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 *Acréscitar Dispositivo*. Pesquise os dispositivos que você deseja e clique no botão *Acréscitar*, como mostrado na imagem a seguir:

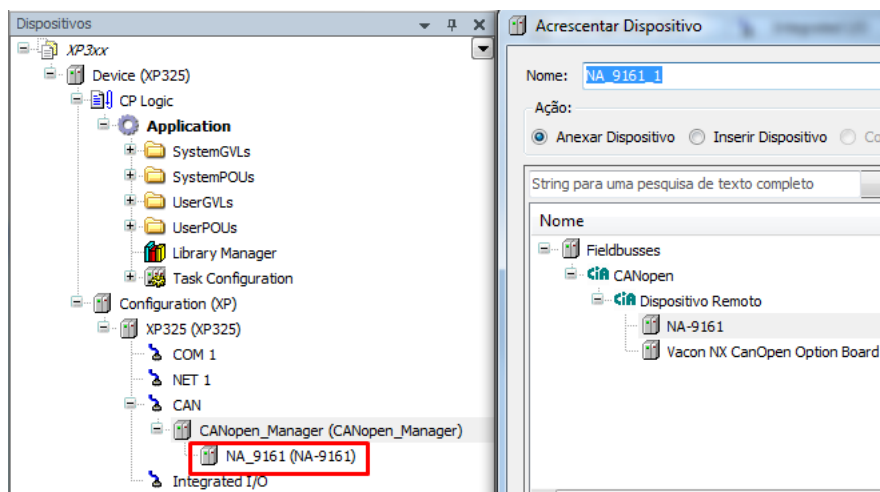


Figura 113: Adicionando um Dispositivo CANopen Escravo

5.7.15.2. 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*:

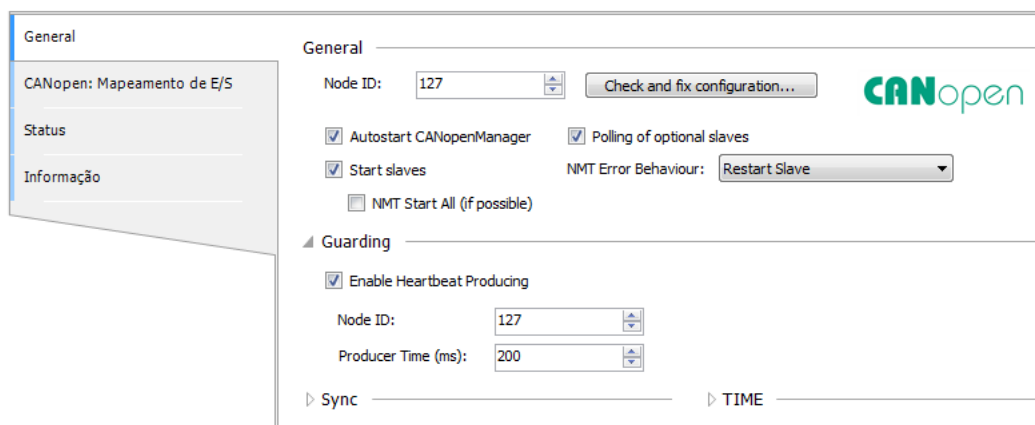


Figura 114: Parâmetros gerais do CANopen Manager

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

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

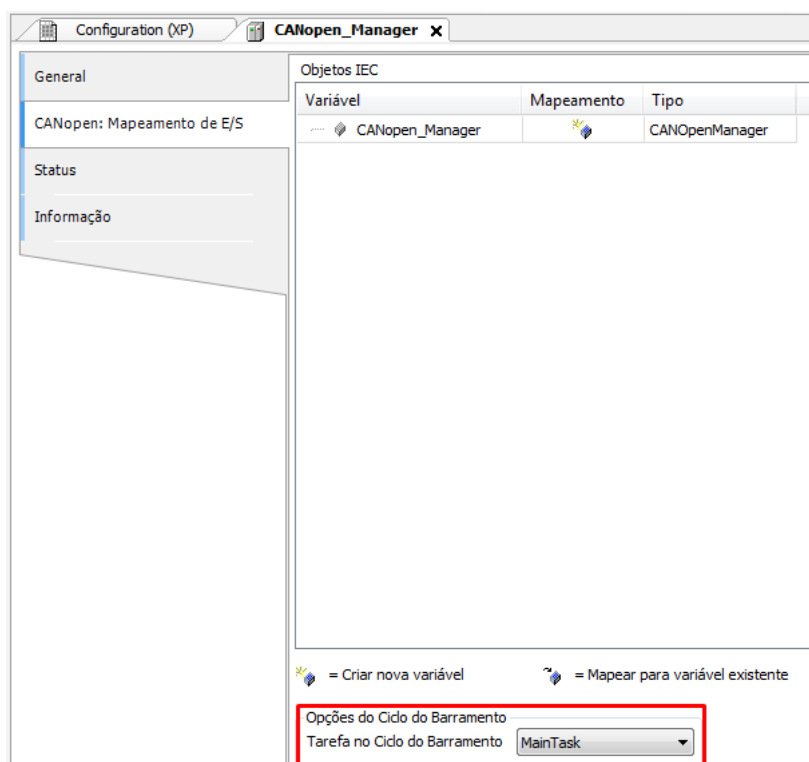


Figura 115: Configuração da tarefa cíclica do barramento CANopen

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.

5.7.15.3. Configuração de Escravo CANopen

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

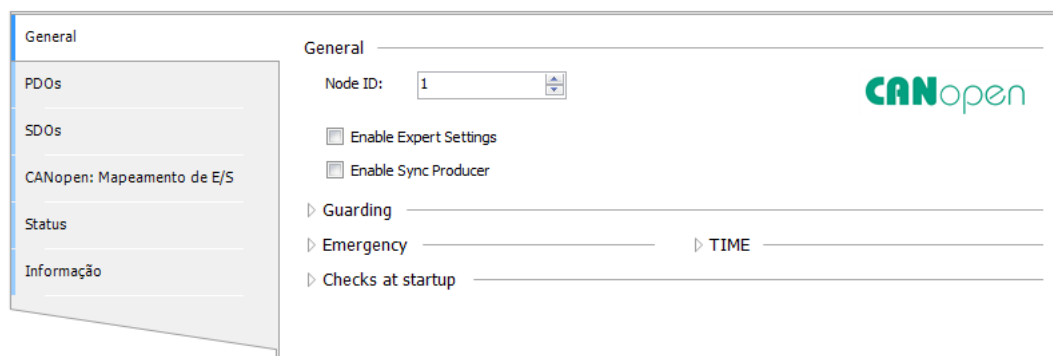


Figura 116: Parâmetros de Escravo CANOpen

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).

5.8. Modo de E/S Remotas

Os controladores Nexto Xpress dispõem de um modo de operação, utilizado para expansão do número de pontos de E/S. Esta expansão pode ser realizada com base nos protocolos CANopen e Ethernet/IP. Quando em modo remoto, o controlador deixa de operar como um CP comum e atua como um escravo remoto, no modo CANopen ou Adapter quando selecionado o EtherNet/IP. Para configurar o seu Xpress como uma expansão de E/S remotas, utilize a seção *Modo de Operação*, na aba *Gerenciamento*, da Página Web de Sistema.



Figura 117: Tela de Configuração Modo de Operação

Nesta página, é possível selecionar o modo de operação do controlador por meio do parâmetro *Modo de Operação*, enquanto o CP encontrar-se em STOP. Utilize o botão *Aplicar* para realizar a troca para o modo desejado. O Xpress será reiniciado, a fim de configurá-lo no novo modo de operação. As opções disponíveis para configuração são:

- **Controlador Programável:** função padrão do controlador, podendo ser programado conforme as necessidades do usuário.
- **Escravo CANopen:** função de expansão de E/S remotas, onde o controlador torna-se um escravo CANopen, podendo ser comandado por outros controladores com CANopen Manager.
- **EtherNet/IP Adapter:** função de expansão de E/S remotas, em que o controlador torna-se um EtherNet/IP Adapter, podendo ser comandado por outros controladores com EtherNet/IP.

ATENÇÃO

O modo de operação remoto utiliza uma aplicação desenvolvida apenas para fins de expansão de E/S. Para o correto funcionamento do CP no modo escravo CANopen, o ciclo de barramento da rede CAN deve ser maior do que 10 ms. Para o caso do CP no modo EtherNet/IP Adapter o RPI mínimo deve ser 20 ms. Essas configurações não podem ser alteradas, nem mesmo é possível realizar a carga de uma nova aplicação durante sua execução.

Quando o modo de operação for remoto, algumas características do controlador serão alteradas. O controlador não poderá mais ser encontrado pelo Mastertool. Contudo, é possível utilizar o *Easy Connection* para localizar o controlador, e até mesmo alterar seu IP. Além disto, na tela de *Atualização de Firmware*, a opção *Apagar Aplicação* encontra-se indisponível.

5.8.1. Escravo CANopen

Para utilizar o modo de expansão como escravo CANopen, primeiramente, troque o *Modo de Operação* para *Escravo CANopen*, na aba de *Modo de Operação* na página Web. Em seguida, realize as configurações para o modo de *Escravo CANopen*: configure os dados de rede (*Endereço IP*, *Máscara de Rede* e *Gateway*), os dados de operação do CANopen (*Node ID*, *Baudrate* e *Terminação*), e os parâmetros de *Configuração de E/S* (conforme a disponibilidade do controlador). Estas configurações são semelhantes às realizadas em uma aplicação tradicional. Enquanto os parâmetros do Escravo CANopen são semelhantes aos apresentados na seção [CANopen Manager](#).

Figura 118: Tela de Configuração do Escravo CANopen

A configuração das entradas e saídas no modo Escravo CANopen é realizada diretamente na página web, da seção Configuração de Entradas e Saídas. As configurações são apresentadas na seção [Configuração das E/S em Modo Remoto](#).

ATENÇÃO

Os PDOs não podem ser editados ou excluídos do Escravo CANopen, não sendo possível ao usuário criar seu próprio escravo CANopen.

Como o Escravo CANopen não é acessível ao usuário pelo Mastertool, o estado RUN e STOP da aplicação é controlado pelo estado de operação do escravo CANopen. Para colocar o Escravo CANopen em RUN, é necessário que o mesmo entre em estado *Operational* (marcado pelo símbolo verde ao lado do dispositivo). Para colocá-lo em STOP, utilize o Bloco Funcional NMT da biblioteca CiA405 - consulte a Ajuda On-line (F1), se necessário - para trocar o estado de operação do escravo CANopen (recomendado), ou remova o conector CAN do controlador em modo remoto. O LED do CAN pode permanecer piscando, pois ele indica a transmissão e recepção de mensagens CAN, não o estado de operação do protocolo CANopen.

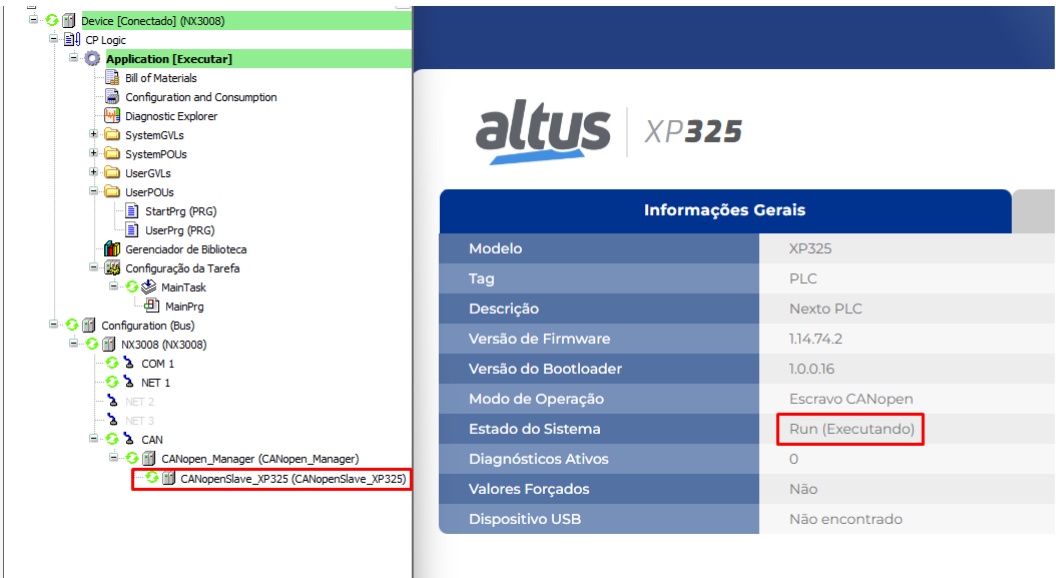


Figura 119: Escravo CANopen em Operational - RUN

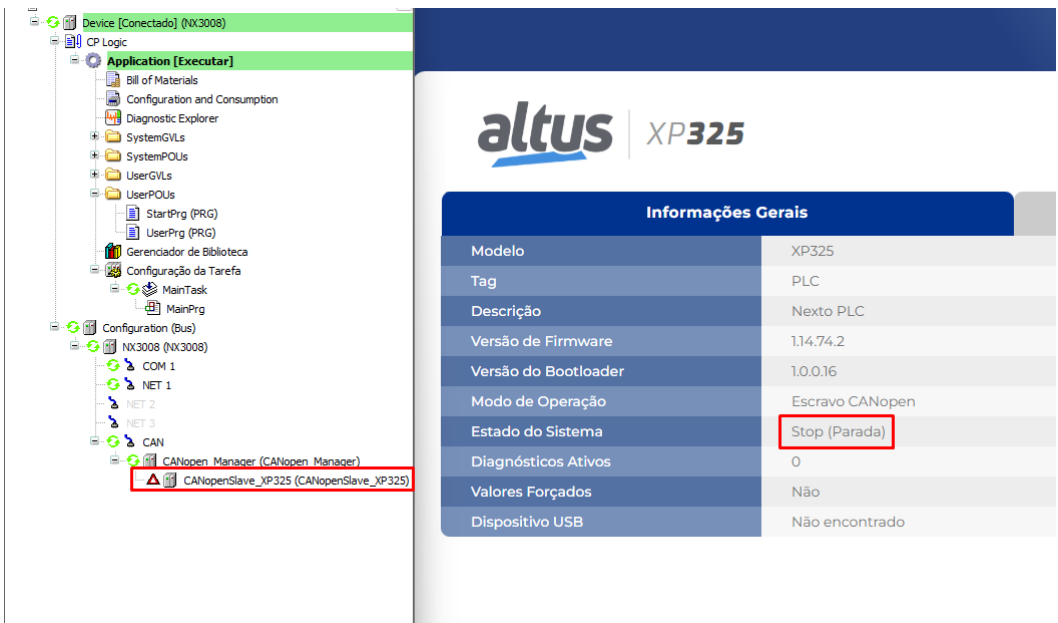


Figura 120: Escravo CANopen em Pre-Operational - STOP

5.8.2. EtherNet/IP Adapter

Para utilizar o modo de expansão como EtherNet/IP Adapter, primeiramente, troque o *Modo de Operação* para *EtherNet/IP Adapter*, na aba de *Modo de Operação* na página Web. Em seguida, realize as devidas configurações para o modo de *EtherNet/IP Adapter*: configure os dados de rede (*Endereço IP*, *Máscara de Rede* e *Gateway*) e os parâmetros de *Configuração*

de E/S (conforme a disponibilidade do controlador). Estas configurações são semelhantes às realizadas em uma aplicação tradicional.

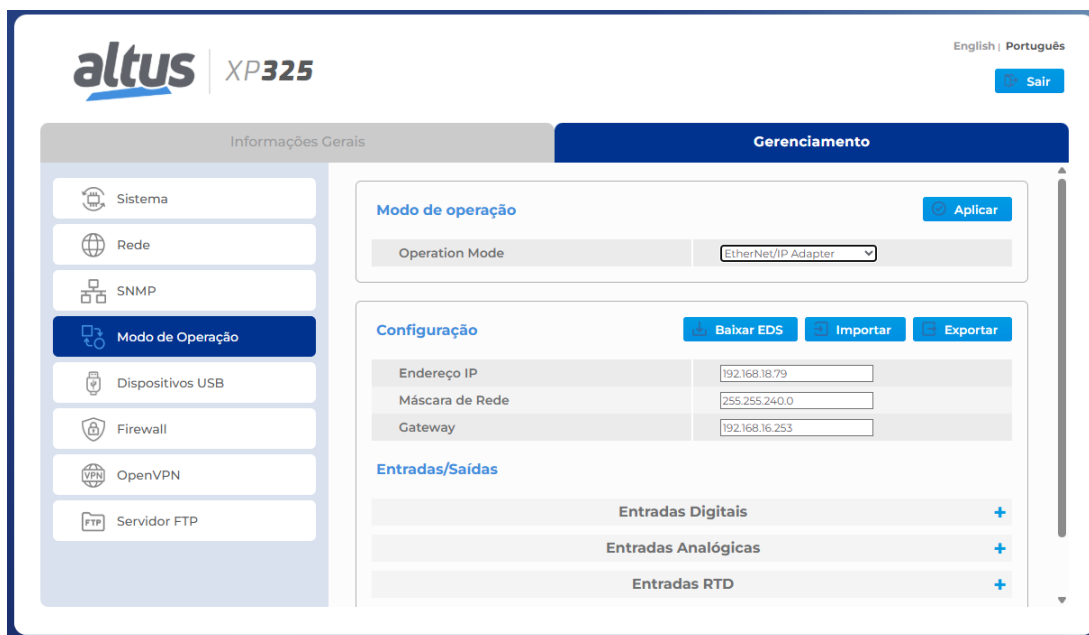


Figura 121: Tela de Configuração do EtherNet/IP Adapter

A configuração das entradas e saídas no modo EtherNet/IP Adapter é realizada diretamente na página web, da seção Configuração de Entradas e Saídas. As configurações são apresentadas na seção [Configuração das E/S em Modo Remoto](#).

ATENÇÃO

Os dados e informações não podem ser editados ou excluídos da Remota EtherNet/IP, não sendo possível ao usuário criar seu próprio EtherNet/IP Adapter.

Ao selecionar o modo de operação EtherNet/IP Adapter, não é possível acessar o CP via Mastertool. Portanto, os estados de operação RUN e STOP da aplicação são controlados pelo estado de operação do EtherNet/IP Adapter. Para assumir o estado RUN, é necessário que um EtherNet/IP Scanner esteja conectado ao controlador. Se nenhum EtherNet/IP Scanner estiver conectado, o controlador permanece no estado STOP.

5.8.3. Configuração das E/S em Modo Remoto

As configurações de entrada e saída são as mesmas para qualquer modo de configuração. Sendo aplicados para o modo escravo CANopen e para o modo EtherNet/IP Adapter. Para isso, clique nos itens com um + à direita para expandir a janela de configuração. Os parâmetros apresentados na *Configuração de E/S* são os mesmos citados na seção [Configuração das E/S Integradas](#). A seguir são apresentadas as telas de configuração para cada tipo de entrada e saída possível.

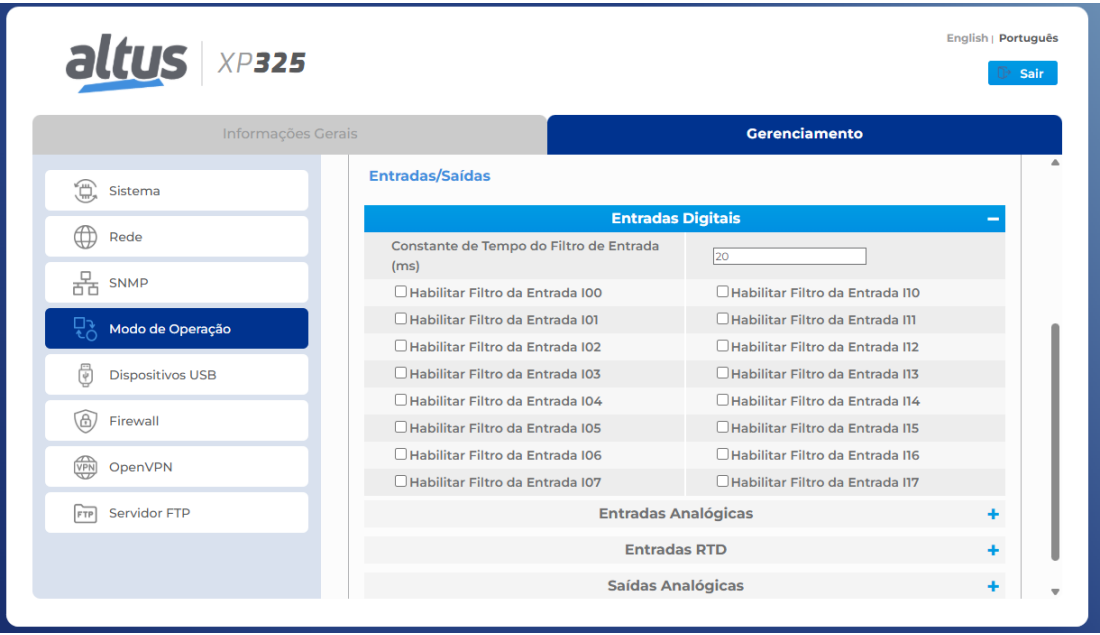


Figura 122: Tela de Configuração das Entradas Digitais



Figura 123: Tela de Configuração das Entradas Analógicas

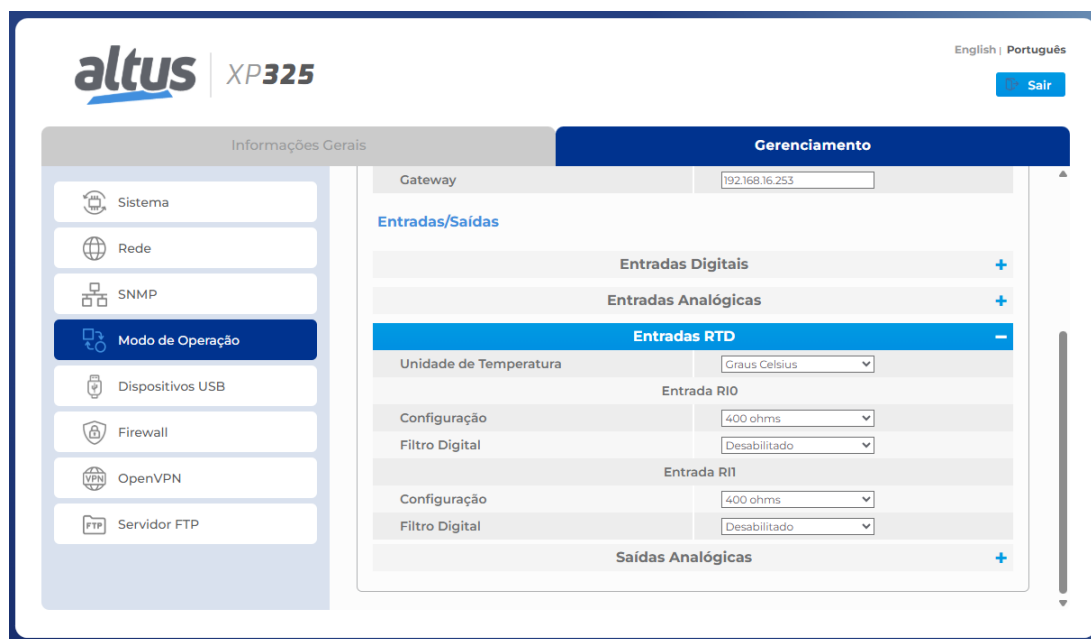


Figura 124: Tela de Configuração das Entradas RTD



Figura 125: Tela de Configuração das Saídas Analógicas

Após as configurações realizadas, é possível utilizar o botão *Exportar Configuração* para realizar o download de um arquivo de configuração, chamado *WebRemoteConfigurationCAN.config*, para o modo Escravo CANOpen e *WebRemoteConfigurationENIP.config*, para o modo EtherNet/IP Adapter, contendo os parâmetros configurados na tela. Posteriormente, este arquivo pode ser carregado utilizando-se o botão *Importar Configuração*. Além disto, é possível fazer o download do arquivo Electronic Data Sheet (EDS) do respectivo modo de operação remoto diretamente pela Web, utilizando o botão *Baixar EDS*.

Concluída a etapa de configuração, clique em *Aplicar Configuração*, o que fará o controlador se reiniciar. A página Web será automaticamente atualizada para o IP configurado. Na guia de *Informações do CP*, é possível verificar o modo de operação do CP, confirmando a troca do modo de operação selecionado.



Figura 126: Modo de Operação na Tela de Informações do CP

Com isto, é possível utilizar um controlador com a funcionalidade CANopen Manager ou EtherNet/IP Scanner (por exemplo, o XP340), a fim de acessar os pontos de E/S do CP no modo Escravo CANopen ou EtherNet/IP Adapter. Para isto, realize os mesmos passos descritos na seção [CANopen Manager](#) deste documento quando selecionado o modo Escravo CANopen ou então os passos escritos na seção [Configuração do Scanner EtherNet/IP](#) para o modo EtherNet/IP Adapter. A estrutura dos dados contendo as informações de entrada e saída disponíveis no CP são organizados conforme a tabela abaixo:

Nome da Variável	Representação	Tipo de Variável
Digital_Outputs_1	Grupo Q0	USINT - 8 bits
Digital_Outputs_2	Grupo Q1	USINT - 8 bits
Analog_Outputs_1	AO0	INT - 16 bits
Analog_Outputs_2	AO1	INT - 16 bits
Analog_Outputs_3	AO2	INT - 16 bits
Analog_Outputs_4	AO3	INT - 16 bits
Digital_Inputs_1	Grupo I0	USINT - 8 bits
Digital_Inputs_2	Grupo I1	USINT - 8 bits
Analog_Inputs_1	AI0	INT - 16 bits
Analog_Inputs_2	AI1	INT - 16 bits
Analog_Inputs_3	AI2	INT - 16 bits
Analog_Inputs_4	AI3	INT - 16 bits
Analog_Inputs_5	AI4	INT - 16 bits
RTD_Inputs_1	RI0	INT - 16 bits
RTD_Inputs_2	RI1	INT - 16 bits
Diagnostics_Analog_Inputs_1	Diagnósticos do AI0	USINT - 8 bits
Diagnostics_Analog_Inputs_2	Diagnósticos do AI1	USINT - 8 bits
Diagnostics_Analog_Inputs_3	Diagnósticos do AI2	USINT - 8 bits
Diagnostics_Analog_Inputs_4	Diagnósticos do AI3	USINT - 8 bits
Diagnostics_Analog_Inputs_5	Diagnósticos do AI4	USINT - 8 bits
Diagnostics_RTD_Inputs_1	Diagnósticos do RI0	USINT - 8 bits
Diagnostics_RTD_Inputs_2	Diagnósticos do RI1	USINT - 8 bits
Diagnostics_Analog_Outputs_1	Diagnósticos do AO0	USINT - 8 bits

Nome da Variável	Representação	Tipo de Variável
Diagnostics_Analog_Outputs_2	Diagnósticos do AO1	USINT - 8 bits
Diagnostics_Analog_Outputs_3	Diagnósticos do AO2	USINT - 8 bits
Diagnostics_Analog_Outputs_4	Diagnósticos do AO3	USINT - 8 bits

Tabela 142: Organização dos dados no modo remoto

As E/S digitais são acessadas por grupos, por meio de uma variável de um byte, onde cada bit representa uma entrada ou saída digital, sendo, por exemplo, o I00 o bit menos significativo e I07, o mais significativo. As E/S analógicas são transmitidas/recebidas diretamente por meio de um inteiro. E os diagnósticos de cada E/S analógica são recebidos em um byte, conforme as tabelas abaixo.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	-	-	bOpenLoop	bOverRange	bInputNotEnable

Tabela 143: Diagnósticos da AIx

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	-	-	bUnderRange	bOverRange	bInputNotEnable

Tabela 144: Diagnósticos da RIx

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	-	-	bShortCircuit	bOpenLoop	bOutputNotEnable

Tabela 145: Diagnósticos da AOx

5.8.4. PROFINET Controller

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

5.9. Desempenho de Comunicação

5.9.1. Servidor MODBUS

Os dispositivos MODBUS configuráveis na UCP Nexto executam em segundo plano, com uma prioridade abaixo da aplicação de usuário e de forma cíclica. Sendo assim, seu desempenho irá variar de acordo com o tempo restante, levando em consideração a diferença entre o intervalo e tempo que a aplicação leva para ser executada. Por exemplo, um dispositivo MODBUS em uma aplicação que é executada a cada 100 ms, com um tempo de execução de 50 ms, irá ter um desempenho menor do que com uma aplicação de 50 ms executando a cada 200 ms de intervalo. Isto acontece porque, no segundo caso, a UCP terá um tempo maior entre cada ciclo da MainTask para executar as tarefas com prioridade mais baixa.

Também deve-se levar em conta o número de ciclos que o dispositivo, escravo ou servidor, leva para responder a uma requisição. Para processar e transmitir uma resposta, um MODBUS RTU Escravo irá levar dois ciclos (tempo do ciclo da tarefa MODBUS), enquanto que um MODBUS Ethernet Servidor levará apenas um ciclo. Porém, esse é o tempo mínimo entre a recepção de uma requisição e o envio da resposta. Caso a requisição seja enviada logo após a execução de um ciclo da tarefa MODBUS, o tempo poderá ser equivalente a 2 ou 3 vezes o tempo de ciclo para o MODBUS Escravo, e de 1 a 2 vezes o tempo de ciclo para o MODBUS Servidor.

Neste caso: Tempo Máximo de Resposta = 3* (tempo de ciclo) + (tempo de execução das tarefas) + (tempo interframe chars) + (tempo de atraso do envio).

Por exemplo, para uma tarefa MODBUS Ethernet Servidor com um ciclo de 50 ms, em uma aplicação executada por 60 ms a cada 100 ms, o servidor conseguirá executar apenas um ciclo entre cada ciclo da aplicação. Por outro lado, com a

mesma aplicação, sendo executada por 60 ms, mas com um intervalo de 500 ms, o MODBUS terá um desempenho melhor, pois enquanto a aplicação não estiver em execução, o mesmo estará sendo executado a cada 50 ms e apenas a cada ciclo da MainTask ele irá demorar mais para ser executado. Para estes casos, o pior desempenho será a soma do tempo de execução da aplicação do usuário com o tempo do ciclo da tarefa MODBUS.

Para os dispositivos mestre e cliente o princípio de funcionamento é exatamente igual, mas levando em consideração o tempo de varredura da relação MODBUS e não o tempo do ciclo da tarefa MODBUS. Para estes casos, o pior desempenho de uma relação será executado após o seu tempo de varredura, somado ao tempo de execução da aplicação de usuário.

É importante ressaltar que o número de dispositivos MODBUS em execução também irá alterar o seu desempenho. Em uma aplicação de usuário com tempo de execução de 60 ms e intervalo de 100 ms, irá restar 40 ms para a UCP executar todas as tarefas de menor prioridade. Portanto, uma UCP com apenas um MODBUS Ethernet Servidor terá um desempenho maior do que uma UCP que utilize quatro destes dispositivos.

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

Tabela 146: 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.

5.9.2. 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.

5.10. 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:

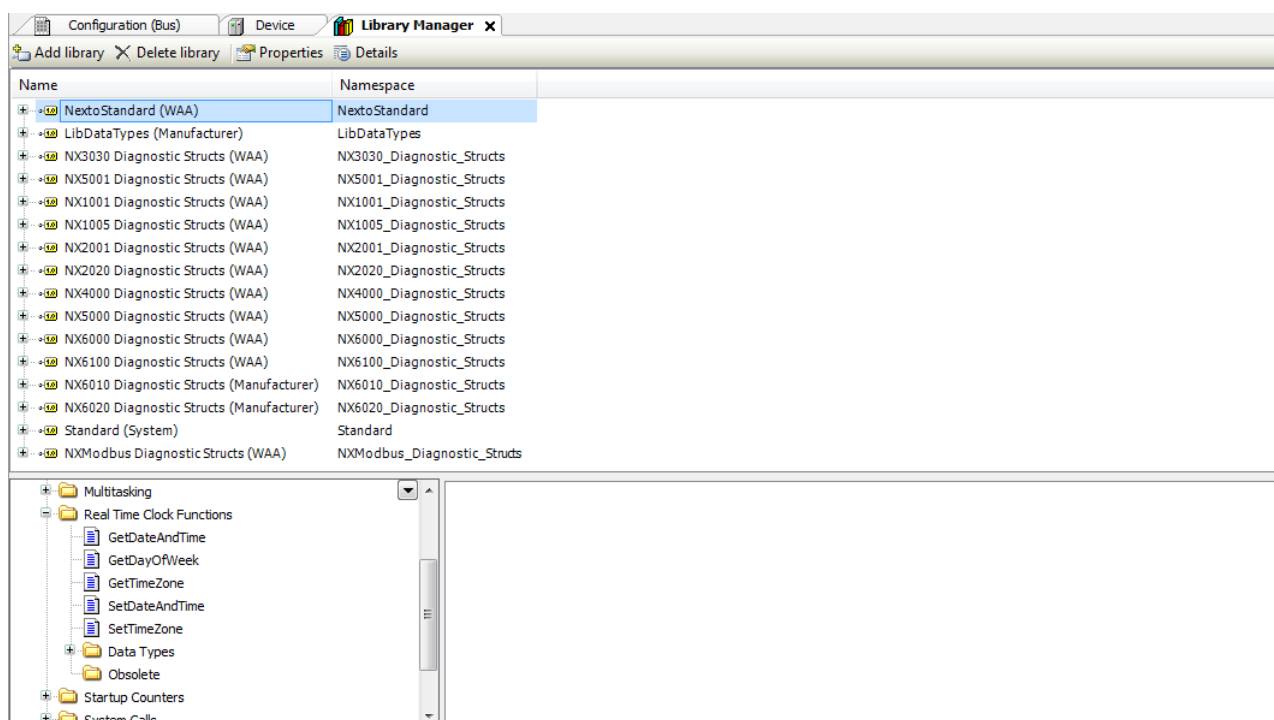


Figura 127: Blocos e Funções de Escrita e Leitura do Relógio

5.10.1. 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.

5.10.1.1. Funções de Leitura do RTC

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

5.10.1.1.1. *GetDateAndTime*

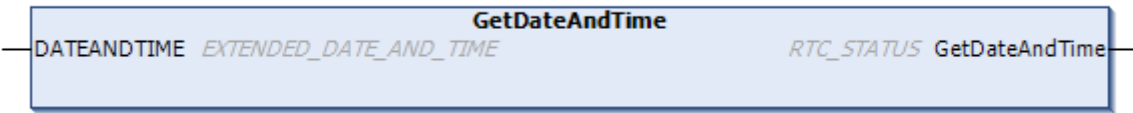


Figura 128: Leitura da Data e Hora

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
```

5.10.1.1.2. *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:

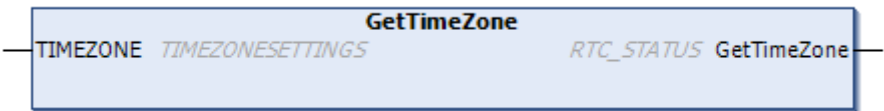


Figura 129: Leitura das Configurações de Fuso Horário

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

```

5.10.1.1.3. GetDayOfWeek

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

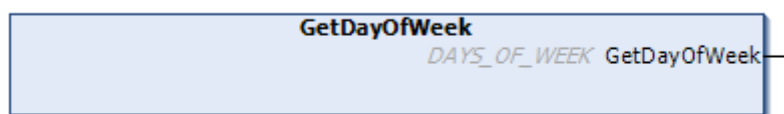


Figura 130: Leitura do Dia da Semana

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();

```

5.10.1.2. Funções de Escrita do RTC

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

5.10.1.2.1. SetDateAndTime

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

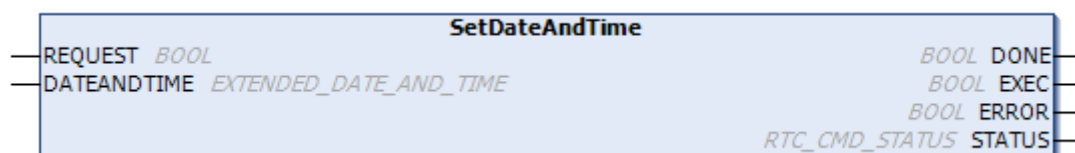


Figura 131: Ajuste de Data e Hora em Milissegundos

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.
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.

5.10.1.2.2. SetTimeZone

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

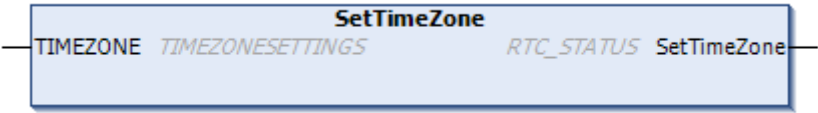


Figura 132: Ajuste de Fuso Horário

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](#).

5.10.2. 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:

5.10.2.1. 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

5.10.2.2. 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

5.10.2.3. 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

5.10.2.4. 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.

5.10.3. 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	2035
Hora	0	23
Minuto	0	59
Segundo	0	59
Milissegundo	0	999

Tabela 160: Faixa de valores válidos para o Relógio RTC

5.11. 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.

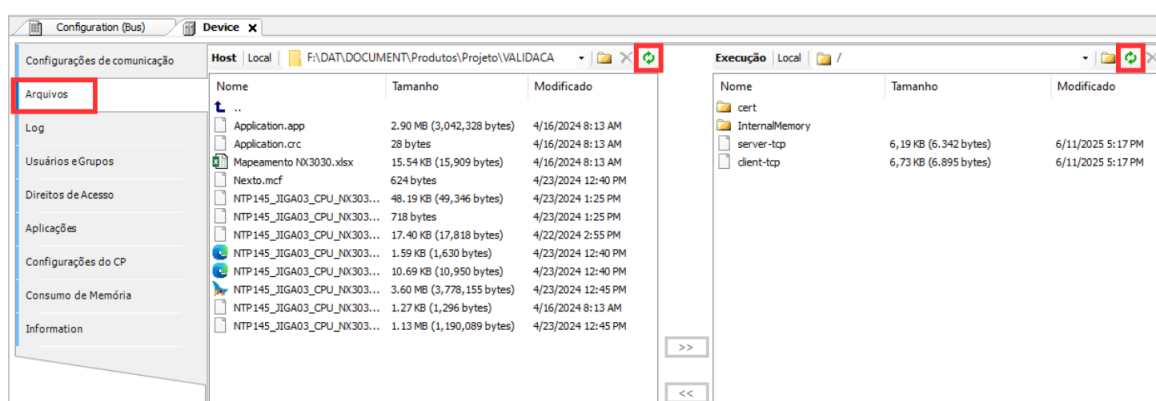


Figura 133: Acesso aos Arquivos de Usuário

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”*.

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.

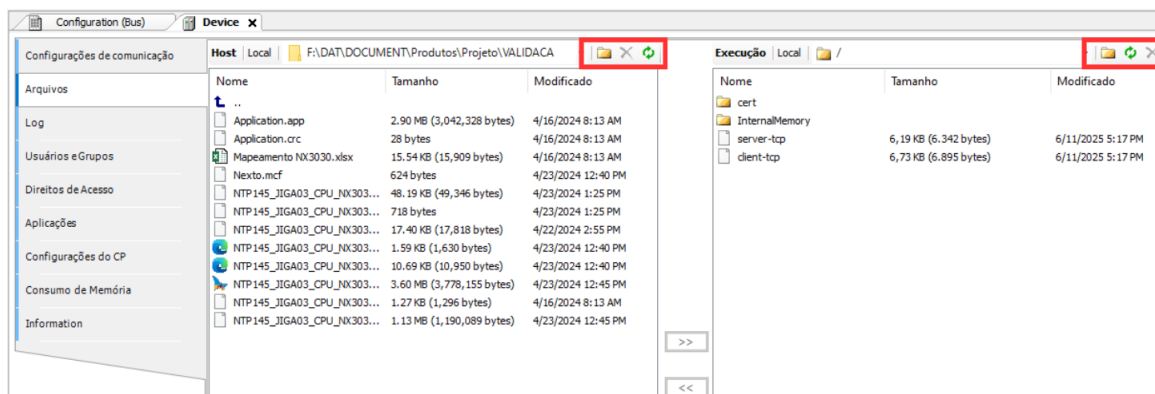


Figura 134: Opções de Utilização

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.

5.12. 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 (consecutivas chamadas) para completar a execução da tarefa. Às vezes, um bloco pode completar em um único ciclo, mas, em geral, necessita de vários ciclos. A execução da tarefa associada a um bloco pode compreender vários passos, sendo que alguns dependem de eventos externos, os quais podem ter atrasos significantes para o sistema. O bloco funcional não pode implementar rotinas para ocupar o tempo, enquanto aguarda por esses eventos, pois assim iria utilizar todo recurso da UCP. A solução poderia ser a criação de blocos funcionais bloqueadores, mas isso não é aconselhável pois iria complicar a aplicação do usuário, pois normalmente não se tem disponível a programação multitarefa. Então, quando um evento externo é esperado, os blocos funcionais da serial são finalizados e o controle é retornado para o programa de chamada. O tratamento da tarefa continua no próximo ciclo, ou seja, na próxima vez que o bloco for chamado.

Antes de descrever os blocos funcionais especiais para controle das interfaces seriais, é importante conhecer os *Datatypes*, ou seja, os tipos de dados utilizados pelos blocos:

Tipo de dado	Opção	Descrição
SERIAL_BAUDRATE	BAUD200	Lista todas as possibilidades da taxa de transmissão (bits por segundo)
	BAUD300	
	BAUD600	
	BAUD1200	
	BAUD1800	
	BAUD2400	
	BAUD4800	
	BAUD9600	
	BAUD19200	
	BAUD38400	
	BAUD57600	
	BAUD115200	
SERIAL_DATABITS	DATABITS_5	Lista todas as possibilidades de bits de dado.
	DATABITS_6	
	DATABITS_7	
	DATABITS_8	
SERIAL_HANDSHAKE	Define todas as possibilidades dos sinais de modem para as configurações:	
	RS232_RTS	Controla a porta RS-232C da UCP Nexto. O RTS é habilitado no início da transmissão e reiniciado assim que possível após o final da transmissão. Por exemplo, pode ser utilizado para controlar um conversor RS-232/RS-485 externo.
	RS232_RTS_OFF	Controla a porta RS-232C da UCP Nexto. O sinal RTS está sempre desligado.
	RS232_RTS_ON	Controla a porta RS-232C da UCP Nexto. O sinal RTS está sempre ligado.
	RS232_RTS_CTS	Controla a porta RS-232C da UCP Nexto. Caso o CTS esteja desabilitado, o RTS é habilitado. Então, aguarda-se o CTS ser habilitado para a transmissão começar e o RTS é reiniciado, o mais rápido possível, no final da transmissão. Ex: Controle de rádio modems com o mesmo sinal de modem.
	RS232_MANUAL	Controla a porta RS-232C da UCP Nexto. O usuário é responsável por controlar todos os sinais (RTS, DTR, CTS, DSR, DCD).
SERIAL_MODE	NORMAL_MODE	Modo normal de operação da comunicação serial.
	EXTENDED_MODE	Modo estendido de operação da comunicação serial, no qual são fornecidas informações sobre o frame de dados recebido.
SERIAL_PARAMETERS	Define todos os parâmetros de configuração da porta serial:	
	BAUDRATE	Definido em SERIAL_BAUDRATE.
	DATABITS	Definido em SERIAL_DATABITS.
	STOPBITS	Definido em SERIAL_STOPBITS.
	PARITY	Definido em SERIAL_PARITY.
	HANDSHAKE	Definido em SERIAL_HANDSHAKE.
	UART_RX_THRESHOLD	Quantidade de bytes que devem ser recebidos para gerar uma nova interrupção na UART. Valores baixos fazem o TIMESTAMP mais preciso quando o MODO ESTENDIDO é utilizado e minimiza os erros de overrun. No entanto, valores baixos podem causar muitas interrupções que podem retardar a UCP. Valores válidos são: 1, 4, 8 ou 14.
	MODE	Definido em SERIAL_MODE.

Tipo de dado	Opção	Descrição
	ENABLE_RX_ON_TX	Quando verdadeiro, todos os bytes recebidos durante a transmissão serão descartados em vez de ir para a fila de RX. Utilizado para desabilitar a operação full-duplex na interface RS-422.
	ENABLE_DCD_EVENT	Quando verdadeiro, gera um evento externo quando o DCD é alterado.
	ENABLE_CTS_EVENT	Quando verdadeiro, gera um evento externo quando o CTS é alterado.
SERIAL_PARITY	PARITY_NONE	Lista todas as possibilidades de paridade.
	PARITY_ODD	
	PARITY_EVEN	
	PARITY_MARK	
	PARITY_SPACE	
SERIAL_PORT	COM 1	Lista todas as portas seriais disponíveis (COM 10, COM 11, COM 12, COM 13, COM 14, COM 15, COM 16, COM 17, COM 18, COM 19 – módulos de expansão).
	COM 2	
SERIAL_RX_CHAR_EX-TENDED	Define um caractere da fila RX modo estendido.	
	RX_CHAR	Byte de dados.
	RX_ERROR	Código de erro.
	RX_TIMESTAMP	Silêncio devido ao caractere anterior ou devido a outro evento que aconteceu antes desse caractere (configuração da porta serial, final da transmissão).
SERIAL_RX_QUEUE_STATUS	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).
	RX_PARITY_ERRORS	Contador de erros de paridade desde a configuração da porta serial. Retorna para zero caso atinja o valor máximo (65535).
	RX_BREAK_ERRORS	Contador de erros de interrupção desde a configuração da porta serial, ou seja, linha ativa maior do que o tempo de um caractere. Retorna para zero caso atinja o valor máximo (65535).
	RX_FIFO_OVERRUN_ERRORS	Contador de erros de overrun na FIFO RX desde a configuração da porta serial, ou seja, erro no threshold configurado para a FIFO RX. Retorna para zero caso atinja o valor máximo (65535).
	RX_QUEUE_OVERRUN_ERRORS	Contador de erros de overrun na fila RX desde a configuração da porta serial, ou seja, o valor máximo de caracteres (1024) foi ultrapassado e os dados estão sendo sobrescritos. Retorna zero caso atinja o valor máximo (65535).
	RX_ANY_ERRORS	Soma dos 5 últimos contadores de erros (frame, paridade, interrupção, overrun RX FIFO, overrun fila RX).
	RX_REMAINING	Número de caracteres remanescentes na fila RX.
	Lista os códigos de erro críticos que podem ser retornados pelos blocos funcionais da serial. Cada bloco retorna erros específicos, os quais serão mencionados na descrição dos mesmos:	
	NO_ERROR	Não existem erros.

Tipo de dado	Opção	Descrição
SERIAL_STATUS	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.
	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)
SERIAL_STOPBITS	NOT_CONFIGURED	O bloco funcional não pode ser utilizado antes da porta serial ser configurada.
	INTERNAL_ERROR	Indica que algum problema interno ocorreu na porta serial.
	STOPBITS_1	Lista todas as possibilidades de bits de parada.
	STOPBITS_2	
	STOPBITS_1_5	

Tabela 161: 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 135: 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 162: Parâmetros de Entrada SERIAL_CFG

Parâmetros de saída	Tipo	Descrição
DONE	BOOL	Essa variável é verdadeira quando o bloco funcional for executado por completo, caso contrário, ela é falsa.
EXEC	BOOL	Essa variável é verdadeira enquanto o bloco funcional estiver sendo executado, caso contrário, ela é falsa.
ERROR	BOOL	Essa variável é verdadeira quando o bloco funcional concluir a sua execução com algum erro, caso contrário, ela é falsa. Está vinculada à variável DONE, pois seu estado é exibido após a conclusão do bloco funcional.
STATUS	SERIAL_STATUS	Caso a variável ERROR seja verdadeira, a estrutura STATUS exibirá o erro encontrado na execução do bloco funcional. Os estados, já descritos no tipo de dado SERIAL_STATUS, possíveis são: - NO_ERROR - ILLEGAL_SERIAL_PORT - ILLEGAL_SERIAL_MODE - ILLEGAL_BAUDRATE - ILLEGAL_DATA_BITS - ILLEGAL_PARITY - ILLEGAL_STOP_BITS - ILLEGAL_HANDSHAKE - ILLEGAL_UART_RX_THRESHOLD - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - DCD_INTERRUPT_NOT_ALLOWED - CTS_INTERRUPT_NOT_ALLOWED - DSR_INTERRUPT_NOT_ALLOWED

Tabela 163: 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 136: 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 164: 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 165: Parâmetros de Saída SERIAL_GET_CFG

Exemplo de utilização em Linguagem ST, após a biblioteca ser inserida no projeto:

```
PROGRAM UserPrg
VAR
  GetConfig: SERIAL_GET_CFG;
  Port: SERIAL_PORT := COM1;
  Parameters: SERIAL_PARAMETERS;
  Status: SERIAL_STATUS;
END_VAR
//ENTRADAS:
GetConfig.REQUEST := TRUE;
GetConfig.PORT := Port;
//FUNÇÃO:
GetConfig();
//SAÍDAS:
GetConfig.DONE;
GetConfig.EXEC;
GetConfig.ERROR;
Status := GetConfig.STATUS; //Caso seja necessário tratar o erro.
Parameters := GetConfig.PARAMETERS; //Recebe os parâmetros da porta serial
desejada.
```

5.12.1.3. SERIAL_GET_CTRL

Esse bloco funcional é utilizado para ler os sinais de controle CTS, DSR e DCD, caso eles estejam disponíveis na porta serial. Será retornado um valor falso quando os sinais de controle não existirem.



Figura 137: Bloco para Visualizar os Sinais de Controle

Parâmetros de entrada	Tipo	Descrição
REQUEST	BOOL	Essa variável, quando verdadeira, habilita o uso do bloco funcional.
PORT	SERIAL_PORT	Seleciona a porta serial, conforme descrito no tipo de dado SERIAL_PORT.

Tabela 166: Parâmetros de Entrada SERIAL_GET_CTRL

Parâmetros de saída	Tipo	Descrição
DONE	BOOL	Essa variável é verdadeira quando o bloco funcional for executado por completo, caso contrário, ela é falsa.
EXEC	BOOL	Essa variável é verdadeira enquanto o bloco funcional estiver sendo executado, caso contrário, ela é falsa.

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. Os estados, já descritos no tipo de dado SERIAL_STATUS, possíveis são: - NO_ERROR - ILLEGAL_SERIAL_PORT - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - FB_GET_CTRL_NOT_ALLOWED - NOT_CONFIGURED
CTS_VALUE	BOOL	Valor lido no sinal de controle CTS.
DSR_VALUE	BOOL	Valor lido no sinal de controle DSR.
DCD_VALUE	BOOL	Valor lido no sinal de controle DCD.

Tabela 167: Parâmetros de Saída SERIAL_GET_CTRL

Exemplo de utilização em Linguagem ST, após a biblioteca ser inserida no projeto e a porta serial ser configurada:

```

PROGRAM UserPrg
VAR
Get_Control: SERIAL_GET_CTRL;
Port: SERIAL_PORT := COM1;
Status: SERIAL_STATUS;
END_VAR
//ENTRADAS:
Get_Control.REQUEST := TRUE;
Get_Control.PORT := Port;
//FUNÇÃO:
Get_Control();
//SAÍDAS:
Get_Control.DONE;
Get_Control.EXEC;
Get_Control.ERROR;
Status := Get_Control.STATUS; //Caso seja necessário tratar o erro.
Get_Control.CTS_VALUE;
Get_Control.DSR_VALUE;
Get_Control.DCD_VALUE;

```

5.12.1.4. SERIAL_GET_RX_QUEUE_STATUS

Esse bloco funcional é utilizado para ler algumas informações de status sobre a fila RX, sendo especialmente desenvolvido para o modo normal, mas pode também ser utilizado no modo estendido.



Figura 138: 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 168: Parâmetros de Entrada SERIAL_GET_RX_QUEUE_STATUS

Parâmetros de saída	Tipo	Descrição
DONE	BOOL	Essa variável é verdadeira quando o bloco funcional for executado por completo, caso contrário, ela é falsa.
EXEC	BOOL	Essa variável é verdadeira enquanto o bloco funcional estiver sendo executado, caso contrário, ela é falsa.
ERROR	BOOL	Essa variável é verdadeira quando o bloco funcional concluir a sua execução com algum erro, caso contrário, ela é falsa. Está vinculada à variável DONE, pois seu estado é exibido após a conclusão do bloco funcional.
STATUS	SERIAL_STATUS	Caso a variável ERROR seja verdadeira, a estrutura STATUS exibirá o erro encontrado na execução do bloco. Os estados, já descritos no tipo de dado SERIAL_STATUS, possíveis são: - NO_ERROR - ILLEGAL_SERIAL_PORT - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - NOT_CONFIGURED
RXQ_STATUS	SERIAL_RX_QUEUE_STATUS	Retorna status/erros da fila RX, conforme descrito no tipo de dado SERIAL_RX_QUEUE_STATUS.

Tabela 169: Parâmetros de Saída SERIAL_GET_RX_QUEUE_STATUS

Exemplo de utilização em Linguagem ST, após a biblioteca ser inserida no projeto e a porta serial ser configurada:

```

PROGRAM UserPrg
VAR
Get_Status: SERIAL_GET_RX_QUEUE_STATUS;
Port: SERIAL_PORT := COM1;
Status: SERIAL_STATUS;
Status_RX: SERIAL_RX_QUEUE_STATUS;
END_VAR
//ENTRADAS:
Get_Status.REQUEST := TRUE;
Get_Status.PORT := Port;
//FUNÇÃO:
Get_Status();
//SAÍDAS:
Get_Status.DONE;
Get_Status.EXEC;
Get_Status.ERROR;
Status := Get_Status.STATUS;    //Caso seja necessário tratar o erro.
Status_RX := Get_Status.RXQ_STATUS; //Caso seja necessário tratar o erro da fila
RX.

```

5.12.1.5. SERIAL_PURGE_RX_QUEUE

Esse bloco funcional é utilizado para limpar a fila RX, local e remota, da porta serial. A UART RX FIFO também é reiniciada.



Figura 139: 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 170: 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 171: Parâmetros de Saída SERIAL_PURGE_RX_QUEUE

Exemplo de utilização em Linguagem ST, após a biblioteca ser inserida no projeto e a porta serial ser configurada:

```
PROGRAM UserPrg
VAR
Purge_Queue: SERIAL_PURGE_RX_QUEUE;
Port: SERIAL_PORT := COM1;
Status: SERIAL_STATUS;
END_VAR
//ENTRADAS:
Purge_Queue.REQUEST := TRUE;
Purge_Queue.PORT := Port;
//FUNÇÃO:
Purge_Queue();
//SAÍDAS:
Purge_Queue.DONE;
Purge_Queue.EXEC;
Purge_Queue.ERROR;
Status := Purge_Queue.STATUS; //Caso seja necessário tratar o erro.
```

5.12.1.6. SERIAL_RX

Esse bloco funcional é utilizado para receber um buffer da porta serial utilizando o modo normal da fila RX. Neste modo, cada caractere na fila RX ocupa um único byte que contém o dado recebido, ou seja, armazena 5, 6, 7 ou 8 bits, de acordo com a configuração da interface serial.



Figura 140: 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.

Parâmetros de entrada	Tipo	Descrição
-----------------------	------	-----------

Tabela 172: 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 173: Parâmetros de Saída SERIAL_RX

5. CONFIGURAÇÃO

Exemplo de utilização em Linguagem ST, após a biblioteca ser inserida no projeto e a porta serial ser configurada:

```
PROGRAM UserPrg
VAR
Receive: SERIAL_RX;
Port: SERIAL_PORT := COM1;
Buffer_Pointer: ARRAY [0..1023] OF BYTE;    //Tamanho máximo.
Status: SERIAL_STATUS;
END_VAR
//ENTRADAS:
Receive.REQUEST := TRUE;
Receive.PORT := Port;
Receive.RX_BUFFER_POINTER := ADR(Buffer_Pointer);
Receive.RX_BUFFER_LENGTH := 1024;    //Tamanho máximo.
Receive.RX_TIMEOUT := 10000;
//FUNÇÃO:
Receive();
//SAÍDAS:
Receive.DONE;
Receive.EXEC;
Receive.ERROR;
Status := Receive.STATUS;    //Caso seja necessário tratar o erro.
Receive.RX_RECEIVED;
Receive.RX_REMAINING;
```

5.12.1.7. SERIAL_RX_EXTENDED

Esse bloco funcional é utilizado para receber um buffer da porta serial utilizando o modo estendido da fila RX, conforme detalhado na seção [Configuração da Interface Serial](#).

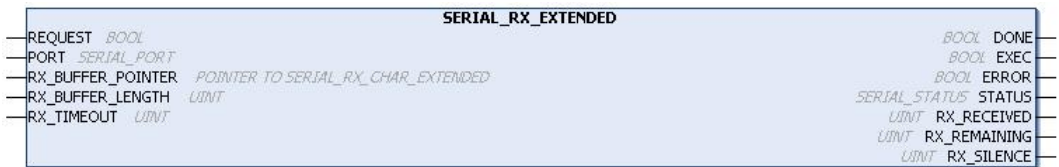


Figura 141: Bloco para Leitura do Buffer de Recepção

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 174: 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 175: Parâmetros de Saída SERIAL_RX_EXTENDED

Exemplo de utilização em Linguagem ST, após a biblioteca ser inserida no projeto e a porta serial ser configurada:

```

PROGRAM UserPrg
VAR
Receive_Ex: SERIAL_RX_EXTENDED;
Port: SERIAL_PORT := COM1;
Buffer_Pointer: ARRAY [0..1023] OF SERIAL_RX_CHAR_EXTENDED;
Status: SERIAL_STATUS;

```

```
END_VAR
//ENTRADAS:
Receive_Ex.REQUEST := TRUE;
Receive_Ex.PORT := Port;
Receive_Ex.RX_BUFFER_POINTER := ADR(Buffer_Pointer);
Receive_Ex.RX_BUFFER_LENGTH := 1024;      //Tamanho máximo.
Receive_Ex.RX_TIMEOUT := 10000;
//FUNÇÃO:
Receive_Ex();
//SAÍDAS:
Receive_Ex.DONE;
Receive_Ex.EXEC;
Receive_Ex.ERROR;
Status := Receive_Ex.STATUS; //Caso seja necessário tratar o erro.
Receive_Ex.RX_RECEIVED;
Receive_Ex.RX_REMAINING;
Receive_Ex.RX_SILENCE;
```

5.12.1.8. SERIAL_SET_CTRL

Esse bloco funcional é utilizado para escrever nos sinais de controle (RTS e DTR), quando estes estiverem disponíveis na porta serial. Também pode determinar uma condição de ocupado para a transmissão, através do parâmetro BREAK, sendo que somente pode ser utilizado se o sinal de modem estiver configurado para RS232_MANUAL.



Figura 142: Bloco para Escrever nos Sinais de Controle

Parâmetros de entrada	Tipo	Descrição
REQUEST	BOOL	Essa variável, quando verdadeira, habilita o uso do bloco funcional.
PORT	SERIAL_PORT	Seleciona a porta serial, conforme descrito no tipo de dado SERIAL_PORT.
RTS_VALUE	BOOL	Valor a ser escrito no sinal RTS.
RTS_EN	BOOL	Habilita a escrita do parâmetro RTS_VALUE.
DTR_VALUE	BOOL	Valor a ser escrito no sinal DTR.
DTR_EN	BOOL	Habilita a escrita do parâmetro DTR_VALUE.
BREAK	BOOL	Caso seja verdadeiro, habilita lógica 0 (ocupado) na linha de transmissão.

Tabela 176: 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 177: Parâmetros de Saída SERIAL_SET_CTRL

Exemplo de utilização em Linguagem ST, após a biblioteca ser inserida no projeto e a porta serial ser configurada:

```

PROGRAM UserPrg
VAR
Set_Control: SERIAL_SET_CTRL;
Port: SERIAL_PORT := COM1;
Status: SERIAL_STATUS;
END_VAR

//ENTRADAS:
Set_Control.REQUEST := TRUE;
Set_Control.PORT := Port;
Set_Control.RTS_VALUE := FALSE;
Set_Control.RTS_EN := FALSE;
Set_Control.DTR_VALUE := FALSE;
Set_Control.DTR_EN := FALSE;
Set_Control.BREAK := FALSE;
//FUNÇÃO:
Set_Control();
//SAÍDAS:
Set_Control.DONE;
Set_Control.EXEC;
Set_Control.ERROR;
Status := Set_Control.STATUS; //Caso seja necessário tratar o erro.

```

5.12.1.9. SERIAL_TX

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).

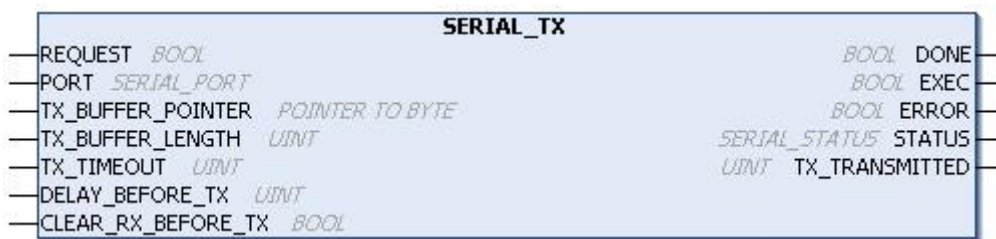


Figura 143: Bloco para Transmitir Valores pela SERIAL_TX

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.

Parâmetros de entrada	Tipo	Descrição
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 178: 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 179: 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. RefreshIntegratedIoInputs

Esta função permite atualizar instantaneamente todas as entradas integradas à UCP do controlador. A função não possui parâmetros de entrada e somente finaliza a execução após atualizar todas as entradas integradas.

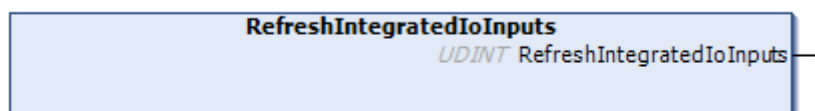


Figura 144: Função para atualizar as entradas integradas

5.12.2.2. RefreshIntegratedIoOutputs

Esta função permite atualizar instantaneamente todas as saídas integradas à UCP do controlador. A função não possui parâmetros de entrada e somente finaliza a execução após atualizar todas as saídas integradas.

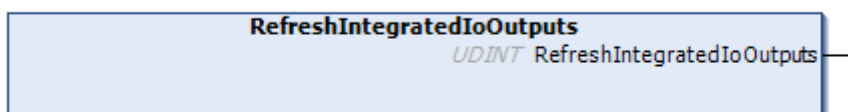


Figura 145: Função para atualizar as saídas integradas

5.12.3. 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).

5.12.3.1. 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 146 representa o bloco *TOF_RET* e a Figura 147 mostra o comportamento gráfico do mesmo.



Figura 146: 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 180: 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 181: Parâmetros de Saída TOF_RET

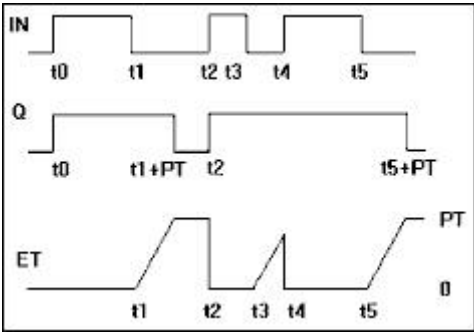


Figura 147: Comportamento Gráfico do Bloco Funcional TOF_RET

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
```

5.12.3.2. 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 148 representa o bloco *TON_RET* e a Figura 149 mostra o comportamento gráfico do mesmo.



Figura 148: 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).

Parâmetros de entrada	Tipo	Descrição
-----------------------	------	-----------

Tabela 182: 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 183: Parâmetros de Saída TON_RET

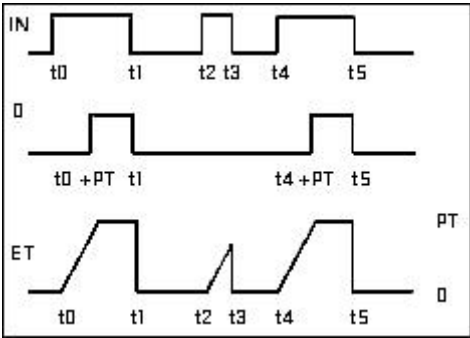


Figura 149: Comportamento Gráfico do Bloco Funcional TON_RET

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
```

5.12.3.3. 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 150 representa o bloco *TP_RET* e a Figura 151 mostra o comportamento gráfico do mesmo.



Figura 150: 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 184: 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 185: Parâmetros de Saída TP_RET

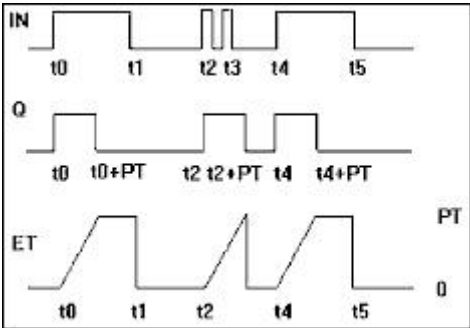


Figura 151: Comportamento Gráfico do Bloco Funcional TP_RET

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
```

5.13. Controle de Movimento (Softmotion)

Os PLCs XP350 e XP351 da Série Nexto possuem suporte a funcionalidade de Controle de Movimento (Softmotion), permitindo aplicações de até 3 eixos a 8 ms. Para informações mais detalhadas ou exemplos de uso, consulte o Help da CODESYS presente em: <https://www.helpme-codesys.com/codesys-softmotion.html>.

Devido a limitações de memória, os XP350 e XP351 não possuem suporte ao Objeto AxisGroup.

5.14. Página 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.

5.15. 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.

5.15.1. 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.



altus XP350		English Portuguese
Informações Gerais		Informações do Sistema
Modelo	XP35008	
Tag	XP35008	
Descrição		
Bastidores Configurados	1	
Bastidor	0	
Posição	0	
Versão de Firmware	134.86.201	
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

5.15.2. 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.

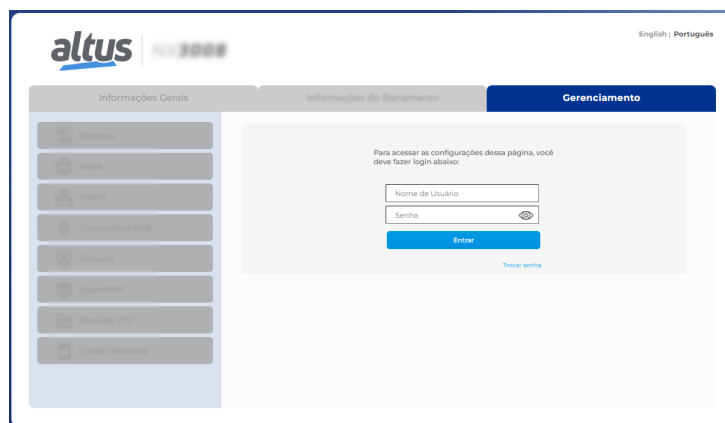


Figura 153: Acesso a aba de Gerenciamento

5.15.2.1. 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.

5.15.2.1.1. 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.

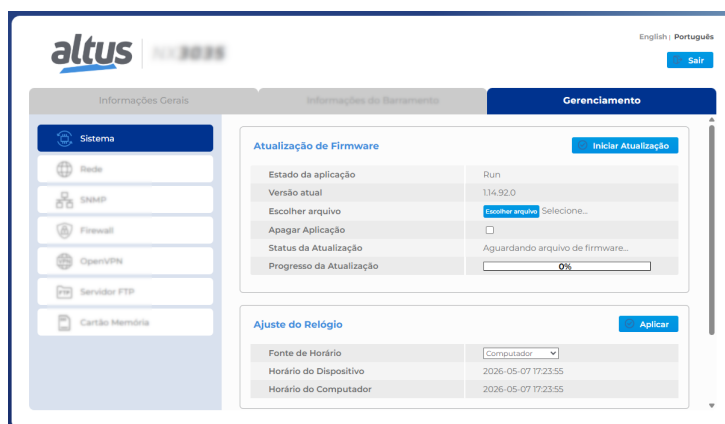


Figura 154: Atualização de Firmware

5.15.2.1.2. 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:

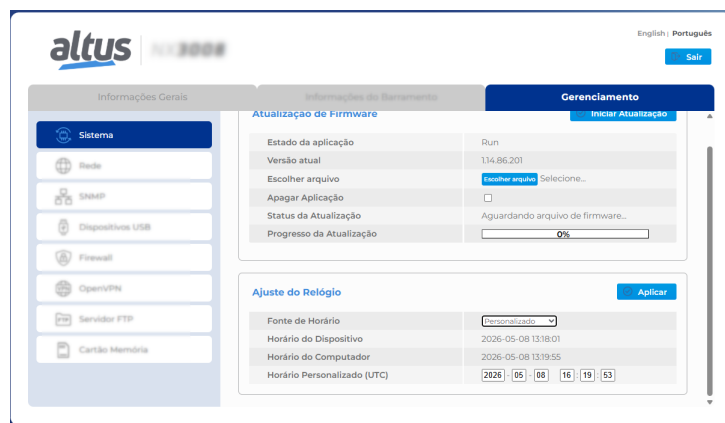


Figura 155: Ajuste do Relógio

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 necessária 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](#).

5.15.2.1.3. 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.

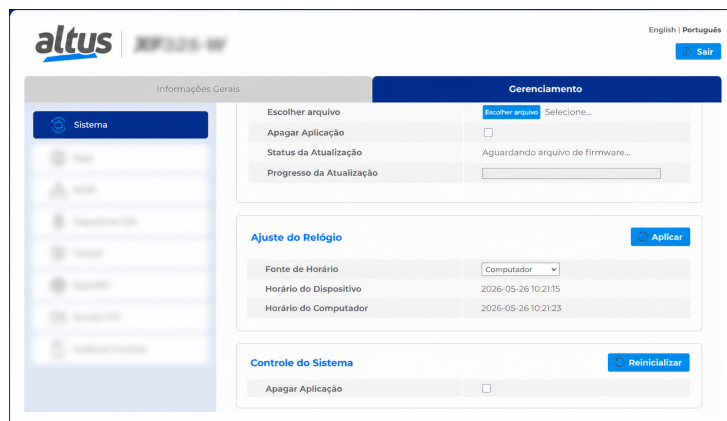


Figura 156: Controle do Sistema

5.15.2.2. 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.

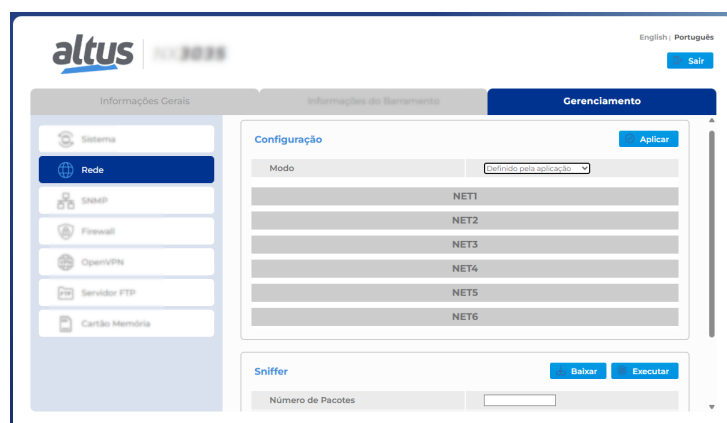


Figura 157: Página Web de Rede

5.15.2.2.1. 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.

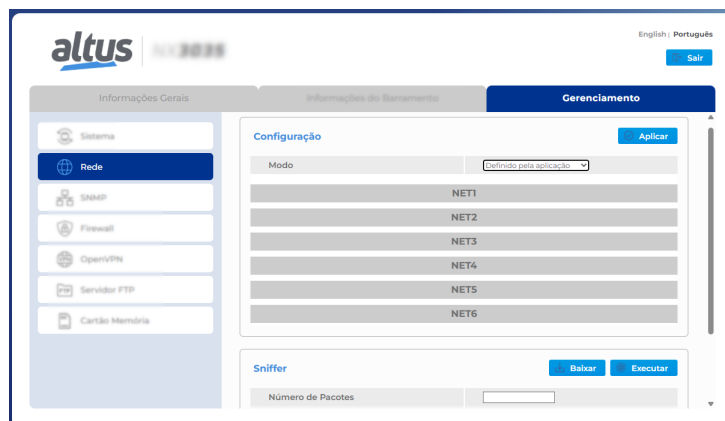


Figura 158: Tabela de Interfaces - Modo Aplicação

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.

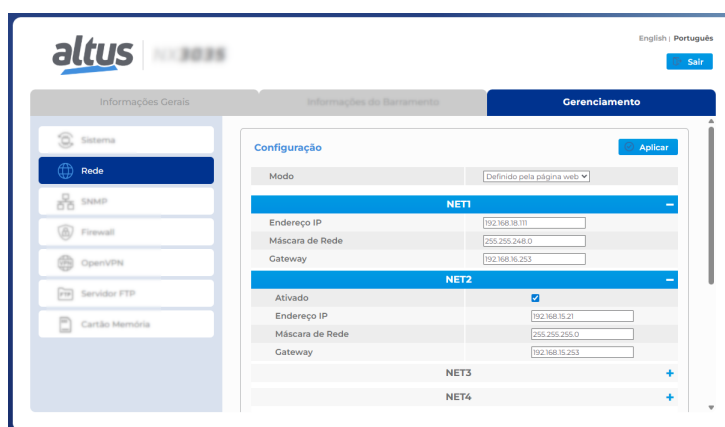


Figura 159: Tabela de Interfaces - Modo Web

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.

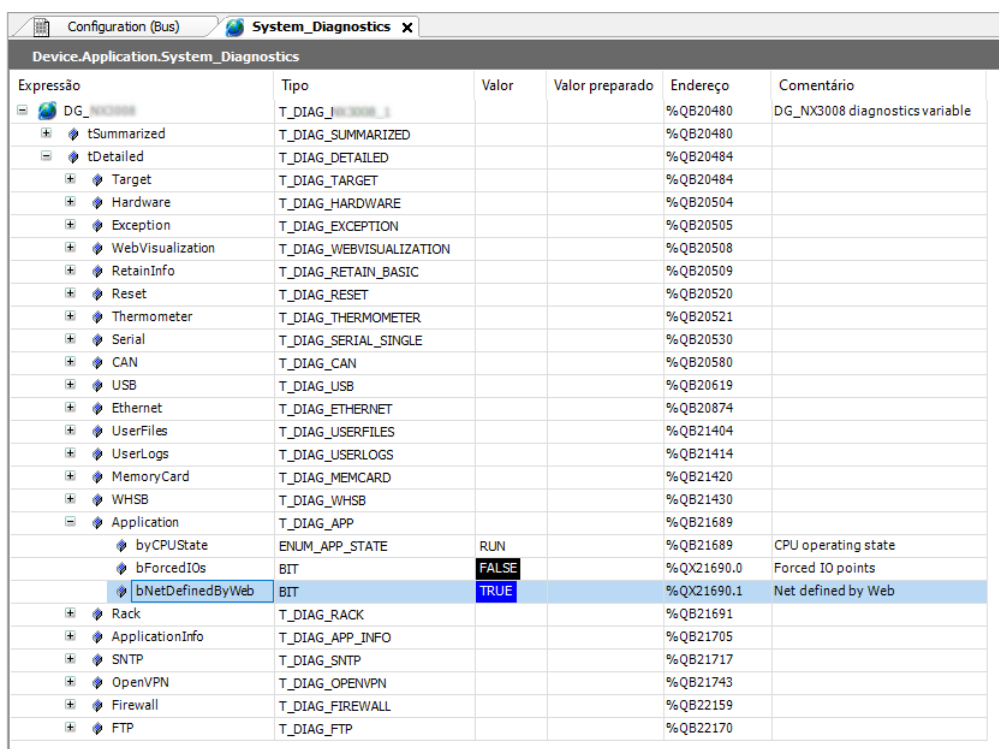
Ao aplicar as configurações utilizando o modo de operação *Definido pela Aplicação*, o controlador vai assumir as configurações que foram definidas pela aplicação já carregada nele anteriormente. Caso não haja nenhuma aplicação no controlador, a configuração atual será mantida, sendo alterado apenas 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 160: Diagnóstico - IP definido pela Página Web

5.15.2.2.2. 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.

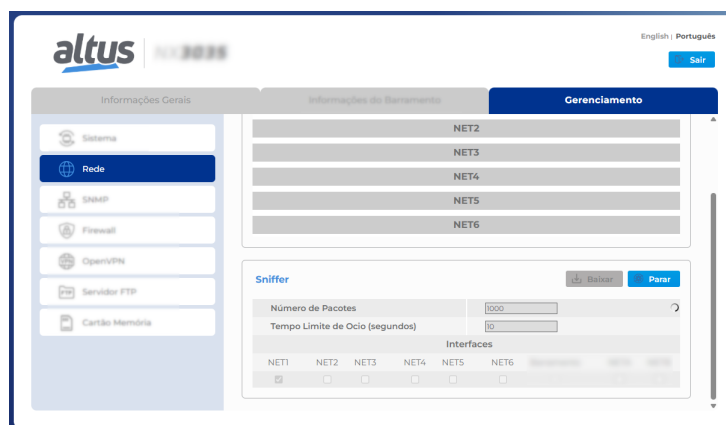


Figura 161: Sniffer de Rede Executando

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.

5.15.2.3. 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.

5.15.2.3.1. 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.

OID	Nome	Descrição
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 186: 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.

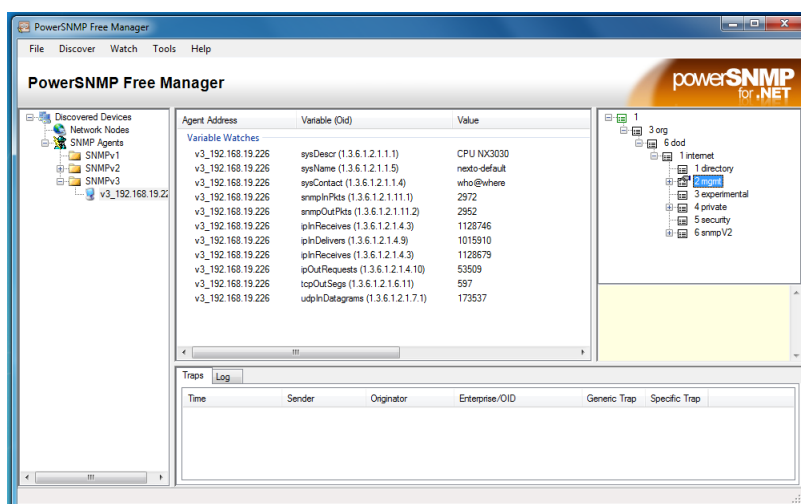


Figura 162: Exemplo de Gerente SNMP

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](#).

5.15.2.3.2. 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. Abrirá um formulário onde é necessário preencher o usuário e senha antigos, e o novo usuário e senha. As demais informações de usuário SNMPv3 não podem ser alteradas.

Para alterar os dados das comunidades SNMPv1/v2c, o processo é parecido, basta clicar no botão *Alterar* abaixo das informações das comunidades. Uma nova tela será aberta onde serão inseridos os novos dados para os campos *rocommunity* e *rwcommunity*. Caso o usuário deixe qualquer um dos campos em branco, a respectiva comunidade será desativada. Dessa forma, se o usuário deixar os 2 campos em branco, o acesso ao agente SNMP será possível somente através do SNMPv3.

Caso o usuário deseje retornar para as configurações padrão, será necessário reconfigurar manualmente as mesmas de acordo com a seção [Usuário e Comunidades SNMP](#). Portanto, todas as configurações SNMP atuais serão mantidas no processo de atualização de firmware. Estas opções podem ser visualizadas na figura abaixo.

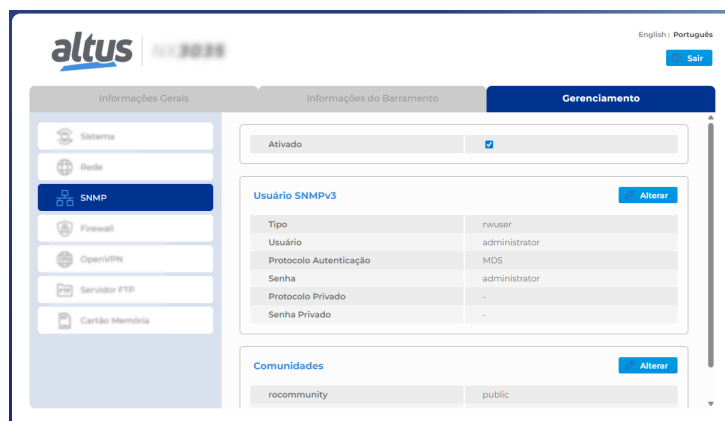


Figura 163: Tela de Configuração e Status SNMP

5.15.2.3.3. 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 187: 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 188: 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 189: Limites para Configurações SNMP

5.15.2.4. 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:

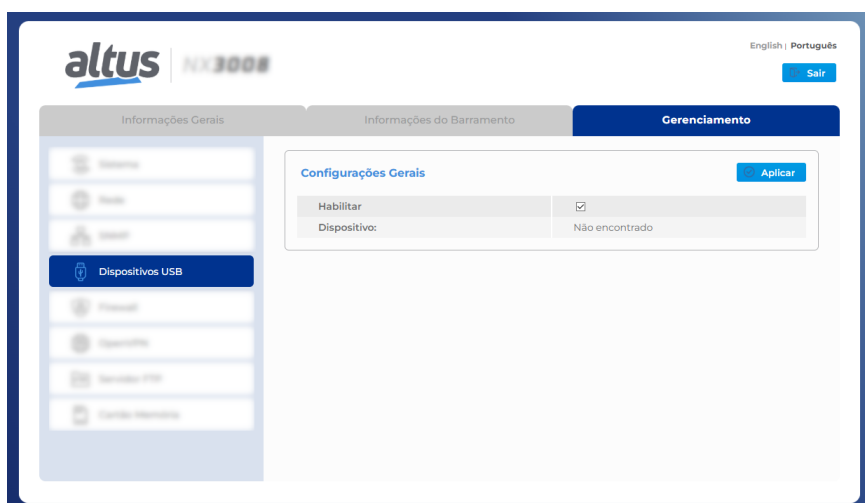


Figura 164: Seção de dispositivos USB

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:

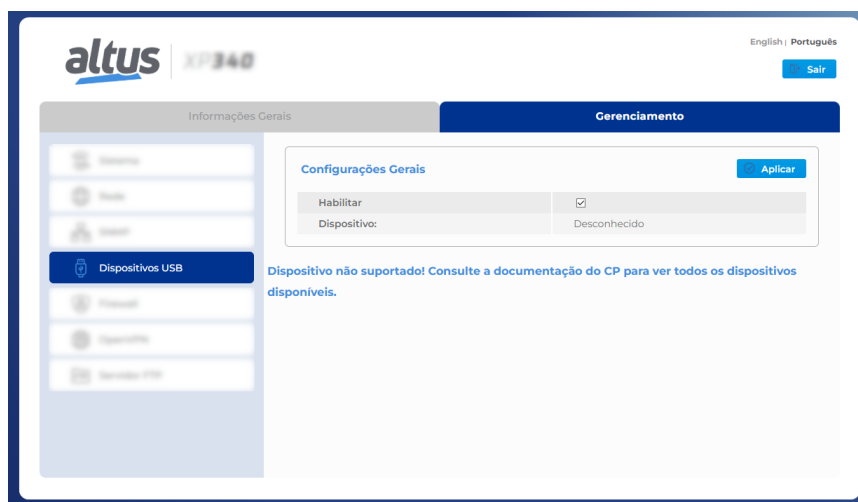


Figura 165: Dispositivos USB - Desconhecido

ATENÇÃO

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

5.15.2.4.1. 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:

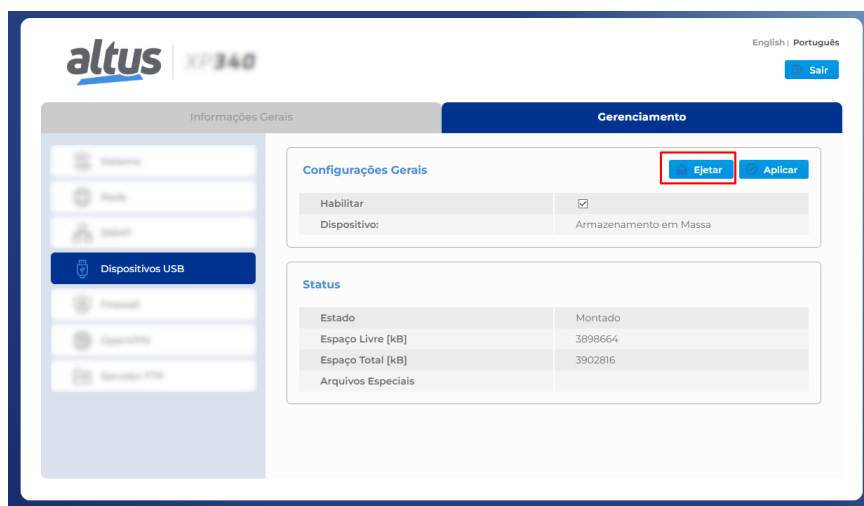


Figura 166: Informações sobre o dispositivo de armazenamento em massa

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:

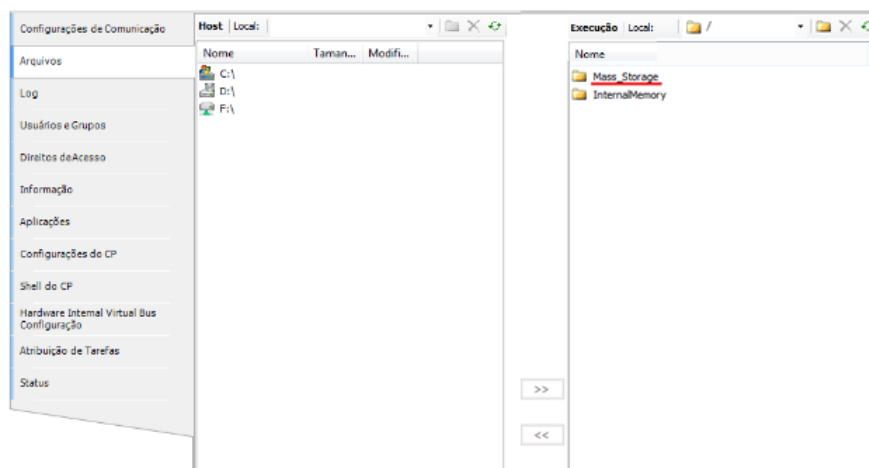


Figura 167: Pasta de armazenamento em massa USB

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.

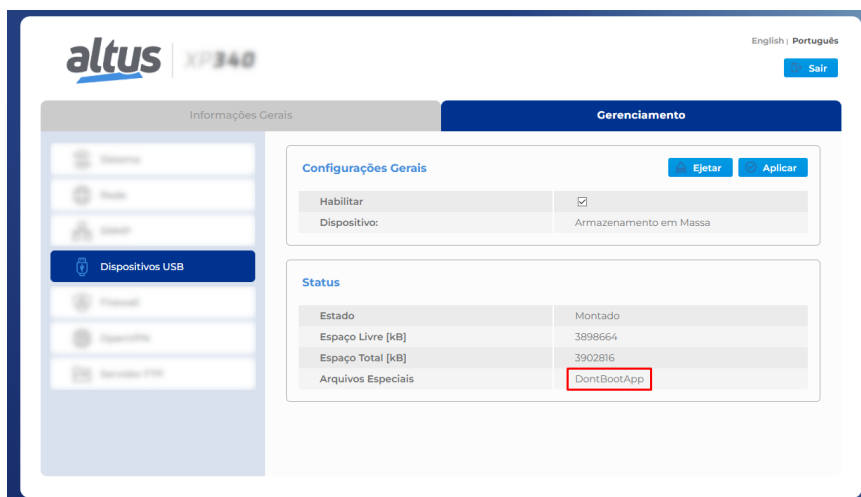


Figura 168: DontBootApp no dispositivo de armazenamento em massa

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:

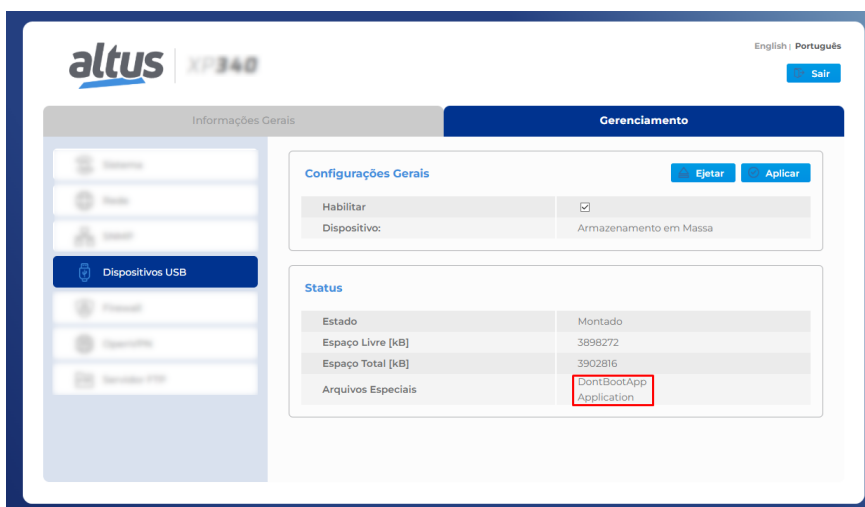


Figura 169: Aplicação no dispositivo de armazenamento em massa

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).

5.15.2.4.2. 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 190: 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:

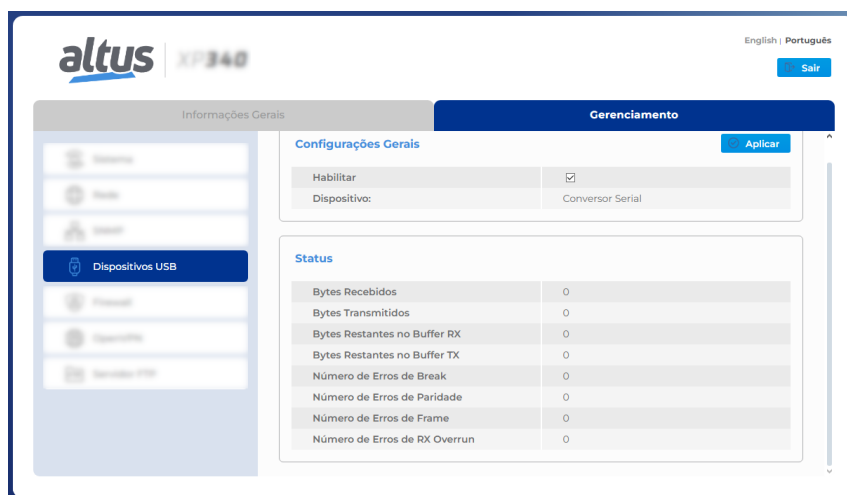


Figura 170: Dispositivos USB - Conversor Serial

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*).

5.15.2.4.3. 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 191: 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:

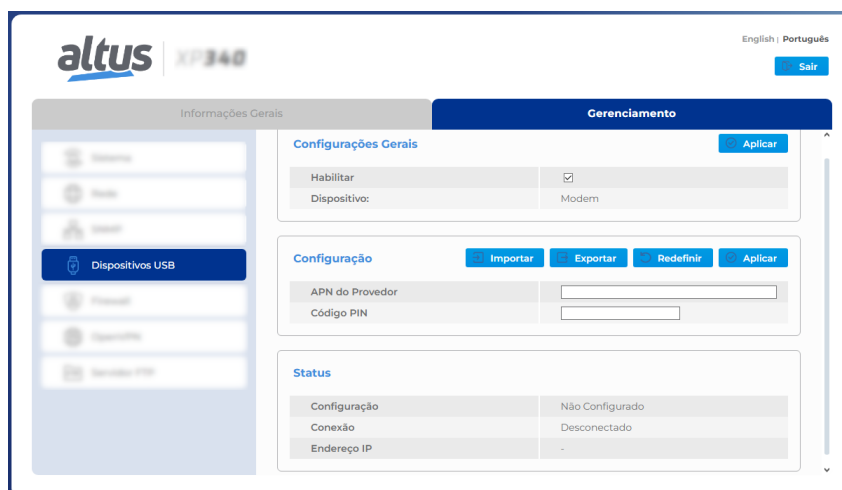


Figura 171: Configuração da Página do Modem USB

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.

5.15.2.4.4. 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 192: 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:

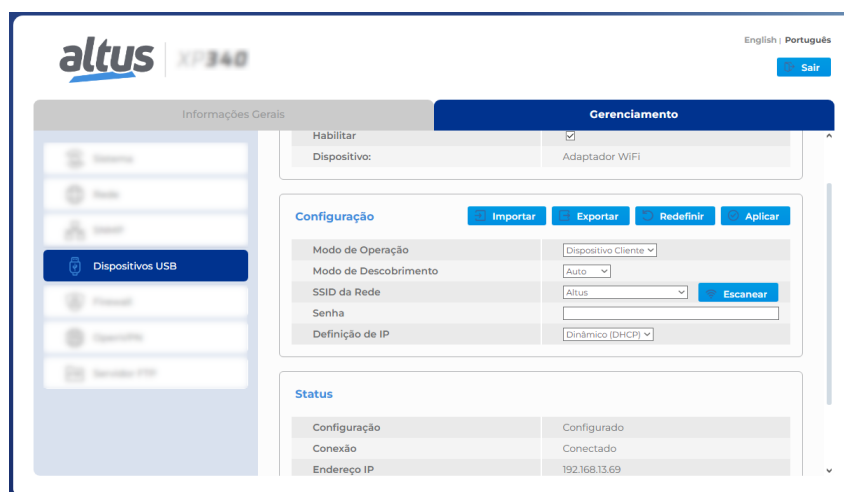


Figura 172: Página do adaptador USB Wi-Fi

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* estiver 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](#)).

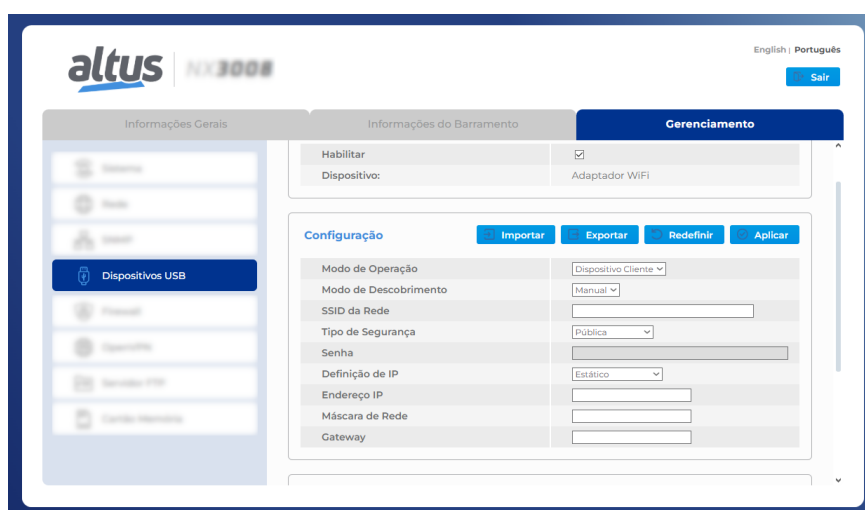


Figura 173: Configuração manual do adaptador USB Wi-Fi

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:

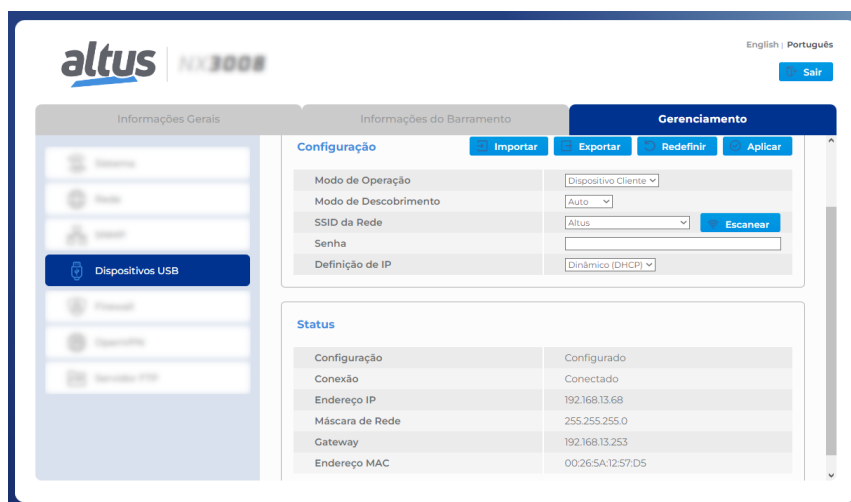


Figura 174: Adaptador USB Wi-Fi conectado a uma rede

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.

5.15.2.4.5. Adaptadores Ethernet

A série de controladores Nexto XF permite implementar uma terceira porta Ethernet usando um adaptador USB para Ethernet. A tabela abaixo mostra a lista de dispositivos suportados:

Adaptador	Fabricante
Adaptador de Rede TP-Link Ethernet Gigabit USB 3.0 UE300	TP-LINK
USB31000S USB 3.0 para Gigabit Ethernet	STAR TECH

Tabela 193: Adaptadores USB para Ethernet suportados

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

Informações Gerais		Gerenciamento										
Habilitar Dispositivo: Adaptador Ethernet		Configuração Endereço IP Máscara de Rede Gateway Aplicar										
Status <table border="1"> <tbody> <tr> <td>Estado de Link</td> <td>DOWN</td> </tr> <tr> <td>Endereço IP</td> <td>-</td> </tr> <tr> <td>Máscara de Rede</td> <td>-</td> </tr> <tr> <td>Gateway</td> <td>-</td> </tr> <tr> <td>Endereço MAC</td> <td>78:8C:B5:92:6D:54</td> </tr> </tbody> </table>		Estado de Link	DOWN	Endereço IP	-	Máscara de Rede	-	Gateway	-	Endereço MAC	78:8C:B5:92:6D:54	
Estado de Link	DOWN											
Endereço IP	-											
Máscara de Rede	-											
Gateway	-											
Endereço MAC	78:8C:B5:92:6D:54											

Figura 175: Página do adaptador USB Ethernet sem configuração aplicada

A página do adaptador USB Ethernet contém basicamente duas seções: *Status* e *Configuração*.

A seção *Status* mostra todos os diagnósticos relacionados ao adaptador Ethernet: estado de Link, endereço IP, máscara de rede, gateway e endereço MAC. Estes campos são atualizados automaticamente conforme o estado atual do dispositivo. Essas informações também são fornecidas na estrutura de diagnóstico de variáveis simbólicas (consulte a Seção [Diagnósticos via Variáveis](#)).

A seção *Configuração* é composta pelos seguintes parâmetros: endereço de IP, máscara de rede e gateway. Esses campos serão utilizados para configurar os parâmetros de rede do adaptador Ethernet.

ATENÇÃO

Para uma operação adequada, a rede do adaptador Ethernet (definida por IP e Máscara) deve ser diferente da rede configurada para NET1 e NET2.

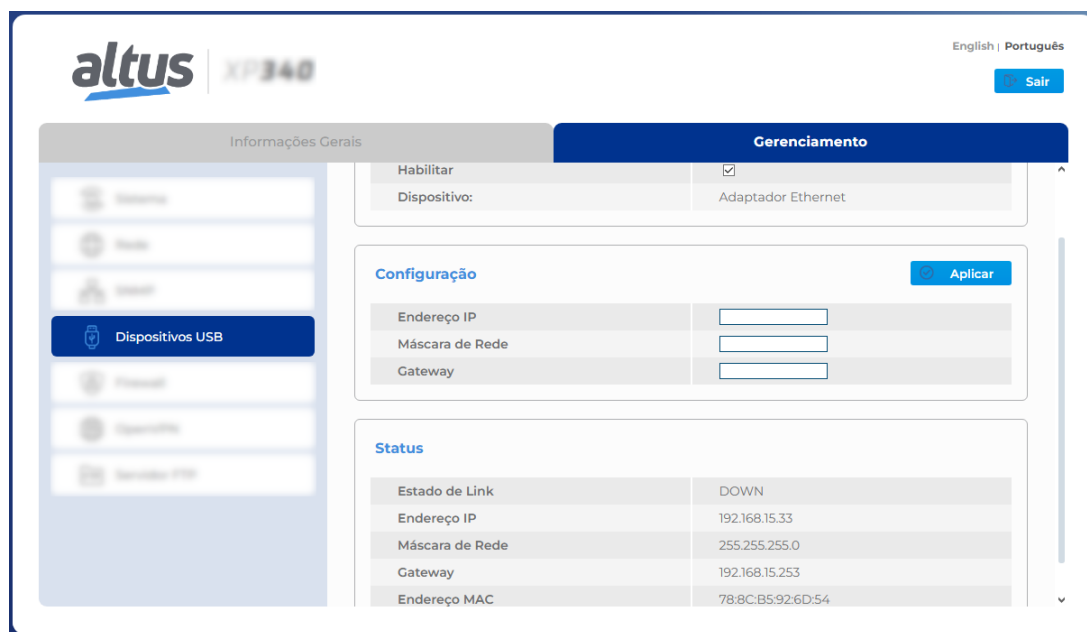


Figura 176: Página do adaptador USB Ethernet com configuração aplicada

Além disso, a seção *Configuração* contém o botão *Aplicar* para que as configurações de IP, Máscara e Gateway sejam definitivamente aplicadas. Depois que o dispositivo for configurado corretamente com parâmetros de rede adequados, sempre ao conectar o adaptador, a última configuração realizada será aplicada, independentemente se for um adaptador diferente do último configurado.

ATENÇÃO

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

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

- Conecte o dispositivo na porta USB. Após alguns segundos, a página *Dispositivos USB* exibirá o *Dispositivo* como "Adaptador Ethernet". Caso contrário, o dispositivo pode não ter suporte ou estar com defeito.
- Defina a configuração de rede. Após preencher os campos, utilize o botão *Aplicar* para realizar as configurações. Uma vez configurado, não é mais necessário definir essas informações. Elas serão salvas na memória do controlador.
- Monitore o status das configurações de rede nos campos correspondentes. Se tudo der certo, após alguns segundos serão exibidos os parâmetros aplicados no item anterior. Após isso, basta conectar um cabo Ethernet para conectar-se a rede (ao conectar o cabo ao adaptador, o estado do Link deverá ir para *UP*).

A imagem a seguir mostra a página de um controlador com um adaptador USB Ethernet configurado e conectado a uma rede:

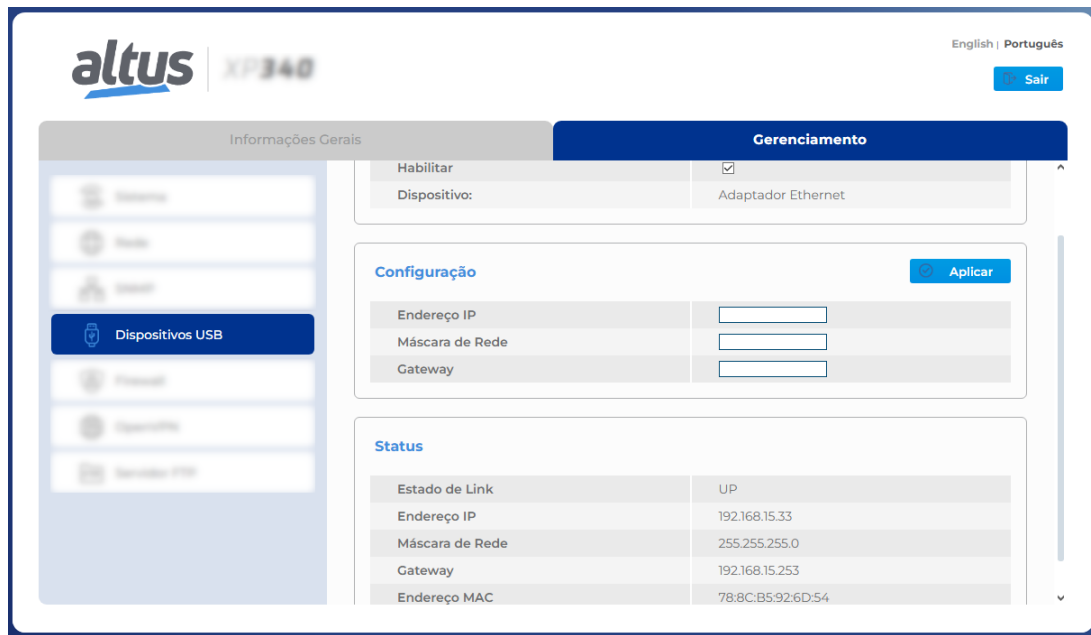


Figura 177: Página do adaptador USB Ethernet com configuração aplicada e link UP

O adaptador USB Ethernet permite que o controlador expanda suas interfaces de rede, criando a possibilidade de dedicar as interfaces nativas do CP para rodar, por exemplo, um protocolo de comunicação. Com isso, o adaptador pode ser utilizado para outros propósitos, tais como programar o CP com o Mastertool ou utilizar o endereço de IP para acessar a página web do sistema do controlador, onde também é possível executar uma atualização de firmware.

5.15.2.4.6. 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*.

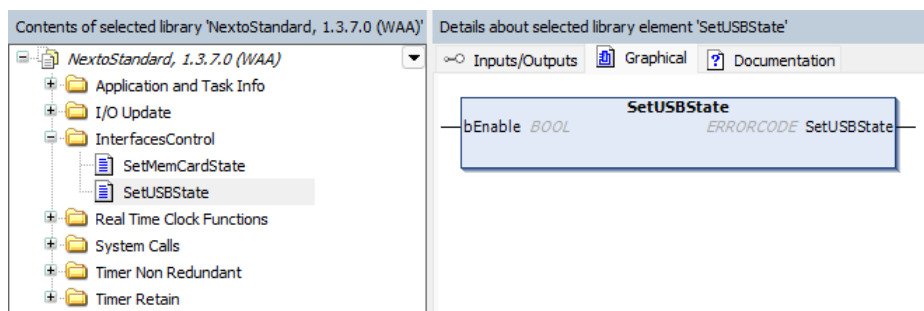


Figura 178: Informações de SetUSBState no Library Manager

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.

5.15.2.5. 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 arquivo *dontbootapp* via USB (veja a Seção [Dispositivos de Armazenamento em Massa](#)), 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:

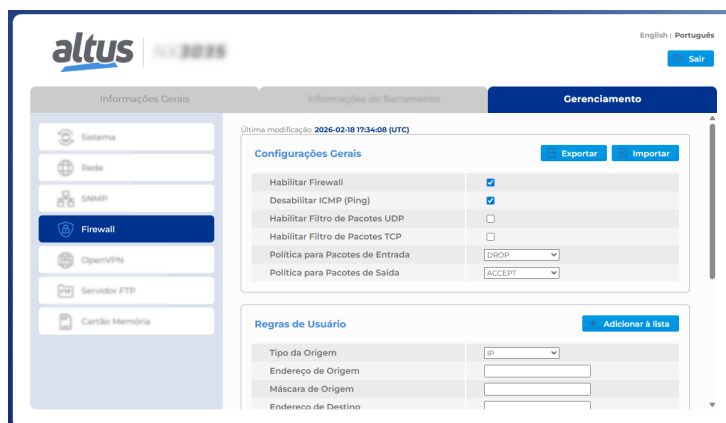


Figura 179: Tela de Configuração do Firewall

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*.

5.15.2.5.1. Configurações Gerais

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

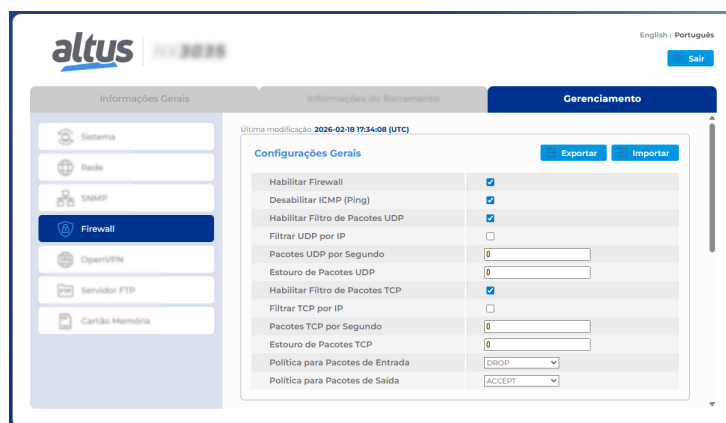


Figura 180: Tabela de Configurações Gerais do Firewall

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 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.

5.15.2.5.2. 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.

A interface web mostra a aba 'Gerenciamento' selecionada. No menu lateral, 'Firewall' está destacado. A seção 'Regras de Usuário' contém um formulário com os seguintes campos: 'Tipo da Origem' (menu suspenso com 'IP' selecionado), 'Endereço de Origem' (campo de texto), 'Máscara de Origem' (campo de texto), 'Endereço de Destino' (campo de texto), 'Máscara de Destino' (campo de texto), 'Interface' (menu suspenso com 'Qualquer' selecionado), 'Ação' (menu suspenso com 'ACCEPT' selecionado), 'Porta do Serviço' (menu suspenso com 'MODBUS - Padrão' selecionado), 'Protocolo' (menu suspenso com 'UDP' selecionado) e 'Direção' (menu suspenso com 'INPUT' selecionado). Abaixo do formulário, há uma seção 'Lista de Regras' com uma tabela vazia e botões 'Adicionar à lista' e 'Aplicar'.

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 *UDP/TCP*. Caso seja selecionado a opção *UDP/TCP*, serão criadas duas regras no Firewall, uma para cada protocolo de transporte.

No campo de *Direção* é possível selecionar entre *INPUT*, *OUTPUT* e *INPUT/OUTPUT*. 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:

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.

5.15.2.5.3. 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.

5.15.2.6. 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.

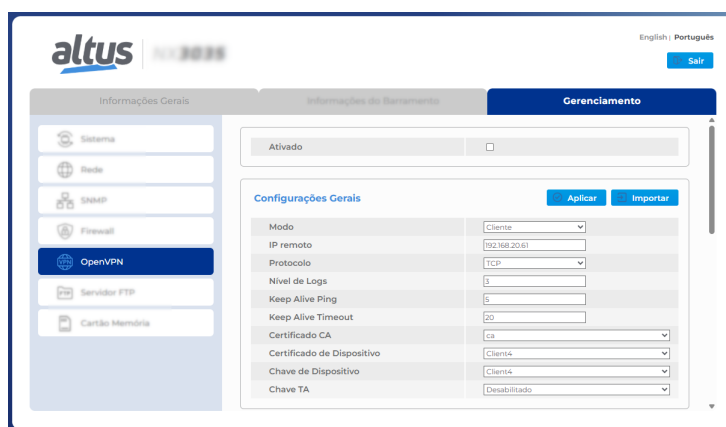


Figura 183: Tela de Configuração de OpenVPN

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.

5.15.2.6.1. Configuração do OpenVPN

Configurações Gerais	
Modo	Servidor
Endereço de IP	
Endereço de Máscara	
Comunicação entre Clientes	Selecionar...
Máximo de Conexões Simultâneas	
Protocolo	Selecionar...
Nível de Logs	5
Keep Alive Ping	5
Keep Alive Timeout	20
Certificado CA	Selecionar...
Certificado de Dispositivo	Selecionar...

Figura 184: Tabela de Configurações de Servidor OpenVPN

Configurações Gerais	
Modo	Cliente
IP remoto	192.168.20.61
Protocolo	TCP
Nível de Logs	5
Keep Alive Ping	5
Keep Alive Timeout	20
Certificado CA	ca
Certificado de Dispositivo	Cliente
Chave de Dispositivo	Cliente
Chave TA	Desabilitado

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.

Tipo	Common Name	Endereço	Máscara
Servidor	server	192.168.80.0	255.255.255.0
Cliente	client2	192.168.70.0	255.255.255.0

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:

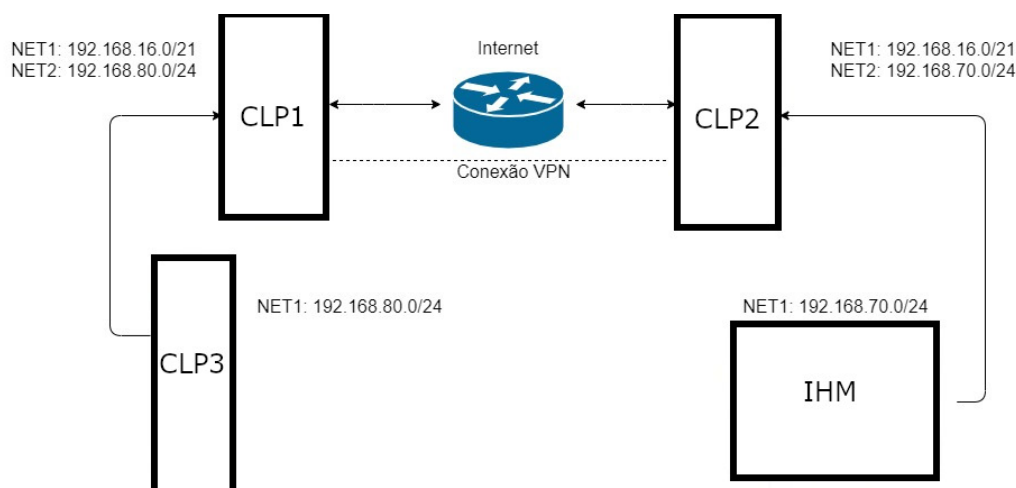


Figura 187: Exemplo de Arquitetura com Redes Privadas

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](#).

5.15.2.6.2. 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:

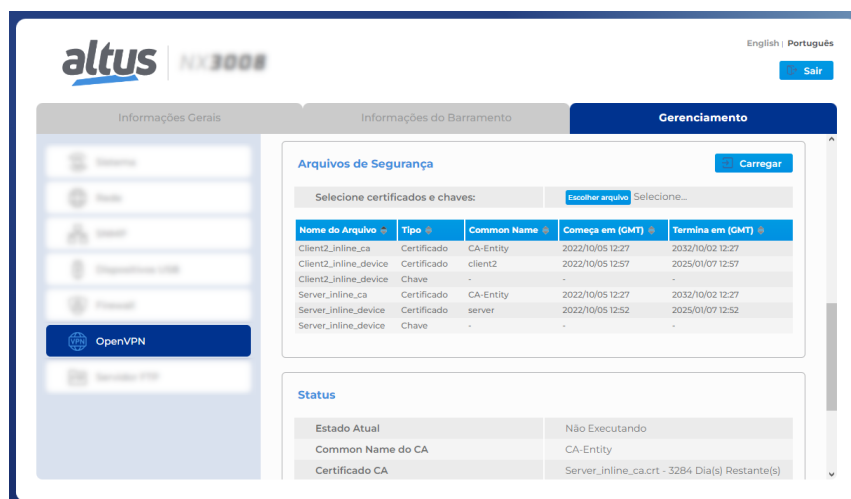


Figura 188: Tabela de Arquivos de Segurança OpenVPN

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 *.cert* e *.key*. Este botão abre uma janela do explorador de arquivos e permite a seleção de um ou mais arquivos.

ATENÇÃO

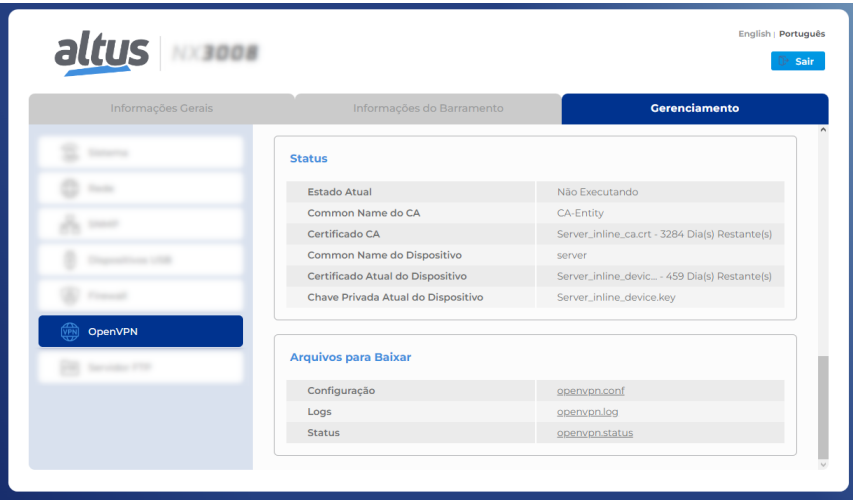
Há um limite de importação de 12 arquivos para o controlador.

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 *common 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.

5.15.2.6.3. 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.



The screenshot shows the Altus NX3008 web interface. The 'Gerenciamento' tab is selected. The 'Status' table displays the following information:

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_device... - 459 Dia(s) Restante(s)
Chave Privada Atual do Dispositivo	Server_inline_device.key

Below the status table, there is a section for 'Arquivos para Baixar' (Files to Download):

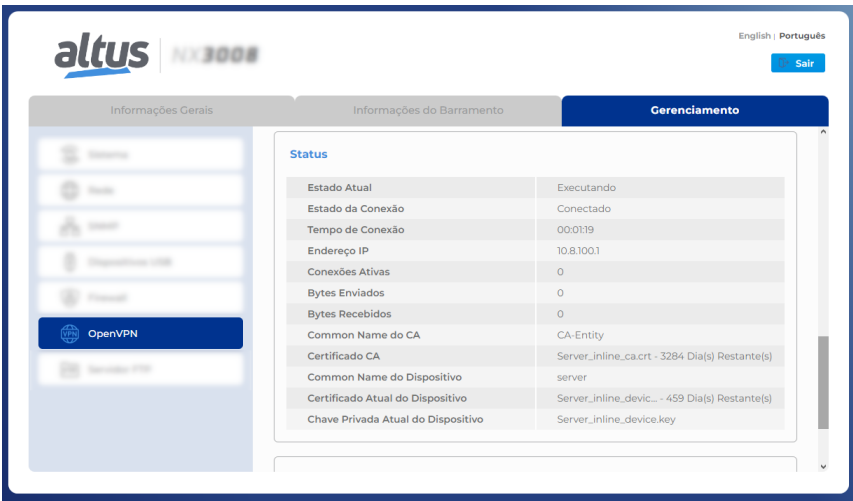
Arquivos para Baixar	
Configuração	openvpn.conf
Logs	openvpn.log
Status	openvpn.status

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 'Gerenciamento' tab is selected. The 'Status' table displays the following information:

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_device... - 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.

5.15.2.6.4. 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.

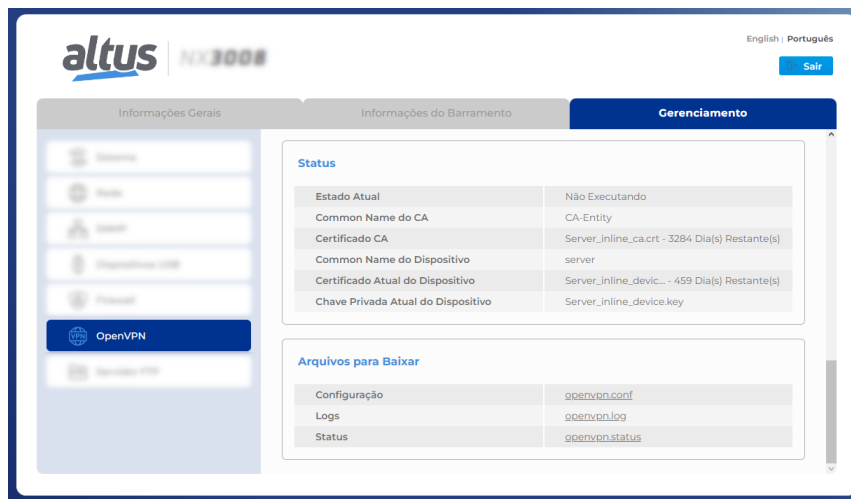


Figura 191: Seção de Downloads OpenVPN

5.15.2.6.5. 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

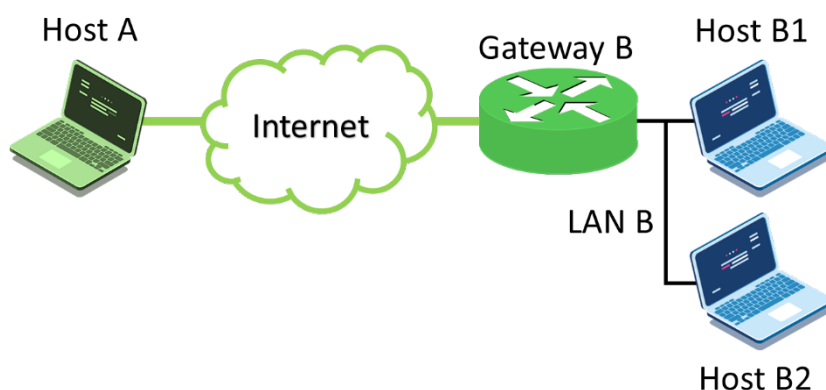


Figura 193: Exemplo de topologia Host-to-Site

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

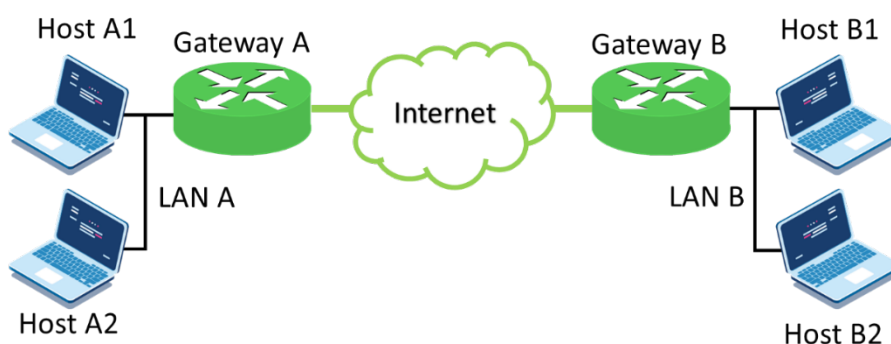


Figura 194: Exemplo de topologia Site-to-Site

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”*.

5.15.2.7. 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.

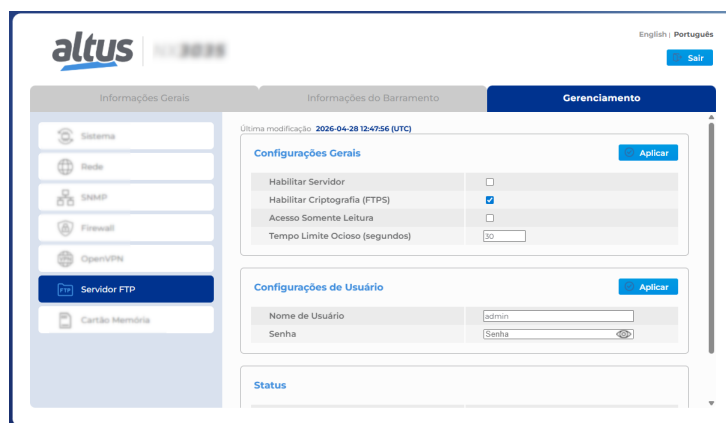


Figura 195: Tela de Configuração do Servidor FTP

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.

5.15.2.7.1. 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.

5.15.2.7.2. 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 múltiplas 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 194: Configurações de Usuário FTP

5.15.2.7.3. 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.

6. Manutenção

6.1. Diagnósticos

Os controladores Nexto Xpress permitem muitas formas de visualizar os diagnósticos gerados pelo sistema, que são:

- Diagnósticos via LED
- Diagnósticos via Página Web de Sistema
- Diagnósticos via Variáveis
- Diagnósticos via Blocos Funcionais

O primeiro é puramente visual, gerado através de dois LEDs colocados no painel frontal (PWR e DG). O próximo recurso é a visualização gráfica em uma página da WEB. Os diagnósticos também são apresentados como variáveis simbólicas globais a serem utilizadas na aplicação do usuário, por exemplo, sendo apresentadas em um sistema de supervisão. Os últimos apresentam condições específicas do funcionamento do sistema.

Essa função de diagnóstico é apontar possíveis problemas de instalação ou configuração do sistema e problemas ou deficiências na rede de comunicação.

6.1.1. Diagnósticos via LED

Os controladores Nexto Xpress possuem um LED de energia (PWR) e um LED de indicação de diagnóstico (DG). A tabela a seguir mostra o significado de cada estado e suas respectivas descrições:

PWR	DG	Descrição	Causas	Prioridade
Desligado	Desligado	Não utilizado	Sem alimentação ou Problema de Hardware	-
Ligado	Desligado	Controlador está iniciando	-	-
Ligado	Ligado	UCP está em RUN e não há diagnósticos ativos	-	5(baixa)
Ligado	Piscando 1x	UCP está em STOP ou não há aplicação carregada	-	2
Ligado	Piscando 2x	Existem diagnósticos ativos	-	3
Ligado	Piscando 3x	Forçamento de dados	Alguma área de memória está sendo forçada pelo usuário através do Mastertool	4
Ligado	Piscando 4x	Erro de hardware	Erro interno de hardware	1
Ligado	Piscando 5x	Falha de Energia	A tensão da fonte de alimentação externa é menor do que o threshold aceitável	0 (alta)

Tabela 195: Descrição dos estados dos LEDs de diagnóstico

6.1.2. Diagnósticos via Página Web de Sistema

Como já é conhecido na Série Nexto, o Nexto Xpress fornece acesso aos estados de operação através de uma Página Web de Sistema.

A utilização, além de dinâmica, é muito intuitiva e facilita as operações do usuário. O uso de um sistema de supervisão pode ser substituído quando estiver restrito à verificação do estado do sistema.

Para acessar a Página Web de Sistema do controlador, basta usar um navegador padrão (Internet Explorer 7 ou superior, Mozilla Firefox 3.0 ou superior e Google Chrome 8 ou superior) e digitar, na barra de endereço, o endereço IP do controlador (e.g. Ex.: <http://192.168.15.1>). Primeiro, as informações do controlador são apresentadas, conforme a figura abaixo:



The screenshot shows the Altus XP340 web interface. At the top, there's a header with the Altus logo, the model 'XP340', and language options 'English | Português'. Below this, it says 'Informações de 2023-10-16, 10:02:36.' and has an 'Atualizar' button. The main content area has two tabs: 'Informações Gerais' (selected) and 'Gerenciamento'. Under 'Informações Gerais', there's a table with the following data:

Informações Gerais	
Modelo	XP340
Tag	PLC
Descrição	Nexto PLC
Versão de Firmware	1.12.29.0
Versão do Bootloader	1.0.0.16
Modo de Operação	Controlador Programável
Estado do Sistema	Stop (Parada)
Diagnósticos Ativos	0
Valores Forçados	Não
Dispositivo USB	Não encontrado

Figura 196: Tela Inicial

O usuário pode optar por visualizar em três diferentes opções de idioma: português, inglês e espanhol. O idioma desejado é selecionado pelo menu superior direito. Além disso, a guia de gerenciamento possui outros recursos, como *Atualização de Firmware* e *SNMP*.

A guia *Atualização de Firmware* é restrita ao usuário, ou seja, apenas para uso interno da Altus. Nos casos em que a atualização é realizada remotamente (por meio de uma conexão de rádio ou satélite, por exemplo), a velocidade mínima do link deve ser de 128 Kbps.

6.1.3. Diagnósticos via Variáveis

Os controladores Nexto Xpress oferecem um conjunto de variáveis simbólicas globais, que fornecem diversas informações de diagnóstico relacionadas ao hardware e ao software. Essas estruturas de dados simbólicas são criadas automaticamente pelo Mastertool.

6.1.3.1. Diagnósticos Resumidos

A tabela a seguir mostra o significado dos diagnósticos resumidos:

DG_XP3xx.tSummarized.*	Tipo	Descrição
bHardwareFailure	BOOL	TRUE – O controlador tem falha interna de hardware.
		FALSE – O hardware está funcionando corretamente.
bSoftwareException	BOOL	TRUE – Uma ou mais exceções geradas pelo software.
		FALSE – Nenhuma exceção gerada no software.
bCOM1ConfigError	BOOL	TRUE – Erro durante / após a configuração da interface serial COM 1.
		FALSE – Correta configuração da interface serial COM 1.
bNET1ConfigError	BOOL	TRUE – Erro durante / após a configuração da interface Ethernet da NET 1.
		FALSE – Correta configuração da interface Ethernet da NET 1.
bInvalidDateTime	BOOL	TRUE – Hora/Data inválida.
		FALSE – Hora/Data correta.
bRuntimeReset	BOOL	TRUE – O RTS (Runtime System) foi reiniciado pelo menos uma vez. Esse diagnóstico só é limpo durante a reinicialização do sistema.
		FALSE – O RTS (Runtime System) está operando normalmente.
bRetentivityError	BOOL	TRUE – Ocorreu um erro ao salvar os dados retentivos.
		FALSE – Dados válidos na memória retentiva durante a inicialização.
bIntegratedIODiagnostic	BOOL	TRUE - Existe algum diagnóstico nas E/S Integradas (ver detalhado)
		FALSE – Sem diagnóstico nas E/S Integradas
bUSBDiagnostic	BOOL	TRUE - Existe algum diagnóstico na USB (ver detalhado)
		FALSE – Sem diagnóstico na USB

Tabela 196: Diagnósticos Resumidos

Notas:

Falha de Hardware: Caso o diagnóstico de falha de hardware seja verdadeiro, o controlador deve ser enviado para a Assistência Técnica da Altus, pois há problemas no RTC ou em outros recursos de hardware.

Exceção de Software: Caso o diagnóstico de exceção de software seja verdadeiro, o usuário deve verificar sua aplicação para garantir que não está acessando a memória incorretamente. Se o problema persistir, o setor de Suporte Técnico da Altus deve ser consultado. Os códigos de exceção do software são descritos a seguir na tabela de diagnósticos detalhados do controlador.

Erro de Retentividade: Caso o diagnóstico de erro de retentividade apresente o valor verdadeiro, o setor de Suporte Técnico da Altus deve ser consultado. Os comandos *Reset a Frio* e *Reset Origem* acionados pelo Mastertool não causam a indicação deste diagnóstico.

6.1.3.2. Diagnósticos Detalhados

As tabelas abaixo contêm os diagnósticos detalhados dos controladores Nexto Xpress. É importante ter em mente as observações abaixo antes de consultá-las:

- **Visualização das Estruturas de Diagnóstico:** As Estruturas de Diagnósticos adicionadas ao Projeto podem ser vistas no item *Gerenciador de bibliotecas* da visualização em árvore do Mastertool. Lá, é possível ver todos os tipos de dados definidos na estrutura

- Contadores: Todos os contadores de diagnósticos do controlador retornam a zero quando seu valor limite é excedido

DG_XP3xx.tDetailed.*		Tipo	Descrição
Target.*	dwCPUModel	ENUM (BYTE)	Modelo do controlador, ex: MODEL_XP325.
	abyCPUVersion	BYTE ARRAY(4)	Versão de firmware.
	abyBootloaderVersion	BYTE ARRAY(4)	Versão de bootloader.
Hardware.*	bRTCFailure	BOOL	O processador principal não consegue se comunicar com o hardware RTC.
	bIntegratedIoFailure	BOOL	O processador principal não consegue se comunicar com o hardware de E/S integradas.
Exception.*	wExceptionCode	WORD	Código de exceção gerado pelo RTS.
	byProcessorLoad	BYTE	Nível, em porcentagem (%), de carga no processador.
WebVisualization.*	byConnectedClients	BYTE	Quantidade de clientes conectados ao WebVisualization.
RetainInfo.*	byCPUInitStatus	BYTE	Estado da Inicialização do controlador: 01: Hot start 02: Warm Start 03: Cold Start Obs.: Essas variáveis são reinicializadas em todas as energizações.
	wCPUColdStartCounter	WORD	Contador de Inicializações a frio: Incrementa caso o CP perca retentividade durante a inicialização. (0 a 65535)
	wCPUWarmStartCounter	WORD	Contador de Inicializações a quente: Incrementa caso o CP inicie com dados retentivos válidos. (0 a 65535)
	wRTSResetCounter	WORD	Contador de reset efetuado pelo RTS (Runtime System) (0 a 65535).
	wWritesCounter	WORD	Contador de gravações na memória retentiva.
Reset.*	bBrownOutReset	BOOL	Última Reinicialização ocorreu devido a uma falha na alimentação.
	bWatchdogReset	BOOL	Última reinicialização ocorreu devido ao cão-de-guarda interno.
Serial. COM1.*	byProtocol	ENUM (BYTE)	Protocolo selecionado na COM 1: NO_PROTOCOL (0): Sem protocolo MODBUS_RTU_MASTER (1): Mestre RTU MODBUS MODBUS_RTU_SLAVE (2): Escravo RTU MODBUS OTHER_PROTOCOL (3): Outro protocolo
	dwRXBytes	DWORD	Contador de caracteres recebidos na COM 1 (0 a 4294967295).
	dwTXBytes	DWORD	Contador de caracteres transmitidos pela COM 1 (0 a 4294967295).
	wRXPendingBytes	WORD	Número de caracteres restantes no buffer de leitura na COM 1 (0 a 4096).
	wTXPendingBytes	WORD	Número de caracteres restantes no buffer de transmissão na COM 1 (0 a 1023).

DG_XP3xx.tDetailed.*		Tipo	Descrição
	wBreakErrorCounter	WORD	Esses contadores são reiniciados nas seguintes condições: - Energização - Configuração da porta serial COM 1 - Remoção de filas de RX e TX Obs .: Quando o controlador está configurado sem paridade, o contador de erros de paridade não é incrementado caso receba uma paridade diferente. Nesse caso, um erro de quadro é indicado. O valor máximo de cada contador é 65535.
	wParityErrorCounter	WORD	
	wFrameErrorCounter	WORD	
	wRXOverrunCounter	WORD	
CAN.*	bBusAlarm	BOOL	O barramento tem um erro crítico e está desligado.
	byBusState	ENUM (BYTE)	Informa o estado do dispositivo: UNKNOWN: impossível obter o estado da rede. ERR_FREE: nenhuma ocorrência de erros de barramento CAN. ACTIVE: somente alguns erros de barramento CAN (abaixo do nível de aviso). WARNING: Ocorrência de alguns erros do barramento CAN (acima do nível de aviso). PASSIVE: muitos erros de barramento CAN (acima do nível de erro). BUSOFF: o nó está desligado (os erros excederam o máximo admissível).
	udiTxCounter	UDINT	Número de pacotes (Tx) alterados no barramento CAN do CP.
	udiRxCounter	UDINT	Número de pacotes (Rx) alterados no barramento CAN do CP.
	udiTxErrorCounter	UDINT	Número de pacotes (Tx) com erros no barramento CAN do CP.
	udiRxErrorCounter	UDINT	Número de pacotes (Rx) com erros no barramento CAN do CP.
	udiLostCounter	UDINT	Número de pacotes perdidos no barramento CAN do CP.
	byUSBDevice	ENUM (BYTE)	Tipo do dispositivo conectado à porta USB: NO_DEVICE UNKNOWN_DEVICE MASS_STORAGE_DEVICE SERIAL_CONVERTER_DEVICE MODEM_DEVICE WIFI_ADAPTER_DEVICE ETHERNET_CONVERTER_DEVICE
	bOvercurrent	BOOL	O dispositivo conectado na porta USB está drenando mais corrente do que o suportado
	bEnable	BOOL	A porta USB está habilitada
	tMassStorage. byMountState	ENUM (BYTE)	Informa o estado do dispositivo: MOUNTED UNMOUNTED

DG_XP3xx.tDetailed.*		Tipo	Descrição
USB.*	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. byProtocol	ENUM (BYTE)	Protocolo selecionado na COM 10: NO_PROTOCOL (0): Sem protocolo
	tSerialConverter. dwRXBytes	DWORD	Contador de caracteres recebidos na COM 10 (0 a 4294967295).
	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	Esses contadores são reiniciados nas seguintes condições: - Energização - Configuração da porta serial COM 10 - Remoção de filas de RX e TX Obs .: Quando o controlador está configurado sem paridade, o contador de erros de paridade não é incrementado caso receba uma paridade diferente. Nesse caso, um erro de quadro é indicado. O valor máximo de cada contador é 65535.
	tSerialConverter. wParityErrorCounter	WORD	
	tSerialConverter. wFrameErrorCounter	WORD	
	tSerialConverter. wRXOverrunCounter	WORD	
	tModem. bConfigured	BOOL	Indica que o modem foi configurado na página da Web.
	tModem. byConnectionState	ENUM (BYTE)	Estado da conexão do modem: DISCONNECTED (1): o modem não está conectado ou configurado. CONNECTING (2): dispositivo configurado, tentando se conectar à Internet. FAILED_RETRYING (3): conexão falhou, o modem tentará se conectar novamente. CONNECTED (4): modem conectado e ativo, o IP está disponível no diagnóstico szIP.
	tModem. szIP	STRING	String com endereço IP usado pelo modem.
tWifiAdapter. bConfigured	BOOL	Indica que o adaptador Wi-Fi foi configurado na página da Web.	
tWifiAdapter. byConnectionState	ENUM (BYTE)	Estado da conexão do adaptador Wi-Fi: DISCONNECTED (1): adaptador Wi-Fi não está conectado ou configurado. CONNECTING (2): dispositivo configurado, tentando se conectar à rede Wi-Fi.	

DG_XP3xx.tDetailed.*		Tipo	Descrição
			FAILED_RETRYING (3): conexão falhou, o adaptador Wi-Fi tentará se conectar novamente. Isso pode ocorrer devido a senha incorreta ou a rede não está disponível. CONNECTED (4): adaptador Wi-Fi conectado à rede.
	tWifiAdapter.szIP	STRING	String com endereço IP usado pelo Wi-Fi.
	tWifiAdapter.szMask	STRING	String com a máscara usada pelo Wi-Fi.
	tWifiAdapter.szGateway	STRING	String com o Gateway usado pelo Wi-Fi.
	tWifiAdapter.szMAC	STRING	String com o endereço MAC do adaptador Wi-Fi. (exclusivo para o dispositivo).
	tEthernetAdapter.bLinkDown	BOOL	Indica o estado do link da interface Ethernet do adaptador.
	tEthernetAdapter.szIP	STRING	String com o endereço IP usado pelo adaptador Ethernet.
	tEthernetAdapter.szMask	STRING	String com a máscara usada pelo adaptador Ethernet.
	tEthernetAdapter.szGateway	STRING	String com o Gateway usado pelo adaptador Ethernet.
	tEthernetAdapter.szMAC	STRING	String com o endereço MAC do adaptador Ethernet. (exclusivo para o dispositivo).
Ethernet. NET1.*	bLinkDown	BOOL	Indica o estado do link na interface.
	wProtocol	WORD	Protocolo selecionado na NET 1: 00: Sem protocolo
	wProtocol. bMODBUS_RTU_ETH_Client	BOOL	Cliente MODBUS RTU via TCP.
	wProtocol. bMODBUS_ETH_Client	BOOL	Cliente TCP MODBUS.
	wProtocol. bMODBUS_RTU_ETH_Server	BOOL	Servidor MODBUS RTU via TCP.
	wProtocol. bMODBUS_ETH_Server	BOOL	Servidor TCP MODBUS.
	szIP	STRING(15)	Endereço IP da porta.
	szMask	STRING(15)	Máscara de sub-rede da porta
	szGateway	STRING(15)	Endereço do Gateway da Porta
	szMAC	STRING(15)	Endereço MAC da porta
	abyIP	BYTE ARRAY(4)	Endereço IP da porta
	abyMask	BYTE ARRAY(4)	Máscara de sub-rede da porta
	abyGateway	BYTE ARRAY(4)	Endereço do Gateway da Porta
	abyMAC	BYTE ARRAY(6)	Endereço MAC da porta
	dwPacketsSent	DWORD	Contador de pacotes enviados pela interface (0 a 4294967295).
	dwPacketsReceived	DWORD	Contador de pacotes recebidos através da interface (0 a 4294967295).

DG_XP3xx.tDetailed.*		Tipo	Descrição
	dwBytesSent	DWORD	Contador de bytes enviados pela interface (0 a 4294967295).
	dwBytesReceived	DWORD	Contador de bytes recebidos pela porta (0 a 4294967295).
	dwTXDropErrors	DWORD	Contador de perdas de conexão na transmissão através da interface (0 a 4294967295).
	dwTXCollisionErrors	DWORD	Contador de erros de colisão na transmissão através da interface (0 a 4294967295).
	dwRXDropErrors	DWORD	Contador de perdas de conexão na recepção através da interface (0 a 4294967295).
	dwRXFrameErrors	DWORD	Contador de erros de quadro na recepção através da interface (0 a 4294967295).
UserFiles.*	byMounted	BYTE	Indica se a memória usada para gravar os arquivos do usuário pode receber dados.
	dwFreeSpacekB	DWORD	Espaço de memória livre para arquivos do usuário (Kbytes).
	dwTotalSizekB	DWORD	Capacidade de armazenamento da memória de arquivos do usuário (Kbytes).
UserLogs.*	byMounted	BYTE	Estado da memória em que os logs do usuário são inseridos.
	wFreeSpacekB	WORD	Espaço de memória livre de logs do usuário (Kbytes).
	wTotalSizekB	WORD	Capacidade de armazenamento da memória dos logs do usuário (Kbytes).
Application.*	byCPUState	ENUM (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.
Application Info.*	dwApplicationCRC	DWORD	CRC de 32 bits de aplicação. Quando a aplicação é modificada e enviada ao controlador, um novo CRC é gerado.
SNTP.*	bServiceEnabled	BOOL	Serviço SNTP ativado.
	byActiveTimeServer	ENUM (BYTE)	Indica qual servidor está ativo: NO_TIME_SERVER (0): Nenhum servidor ativo. PRIMARY_TIME_SERVER (1): servidor principal ativo. SECONDARY_TIME_SERVER (2): Servidor secundário ativo.
	wPrimaryServerDownCount	WORD	Contador de vezes em que o servidor principal está indisponível (0 a 65535).
	wSecondaryServerDownCount	WORD	Contador de vezes em que o servidor secundário está indisponível (0 a 65535).
	dwRTCTimeUpdatedCount	DWORD	Contador de vezes que o RTC foi atualizado pelo serviço SNTP (0 a 4294967295).
	byLastUpdateSuccessful	ENUM (BYTE)	Última atualização: NOT_UPDATED (0): não atualizado. UPDATE_FAILED (1): falha.

DG_XP3xx.tDetailed.*	Tipo	Descrição
		UPDATE_SUCCESSFUL (2): bem sucedido.
	byLastUpdateTimeServer	ENUM (BYTE) Servidor usado na última atualização: NO_TIME_SERVER (0): Nenhuma atualização. PRIMARY_TIME_SERVER (1): servidor principal. SECONDARY_TIME_SERVER (2): servidor secundário.
	sLastUpdateTime. byDayOfMonth	BYTE
	sLastUpdateTime. byMonth	BYTE
	sLastUpdateTime. wYear	WORD
	sLastUpdateTime. byHours	BYTE
	sLastUpdateTime. byMinutes	BYTE
	sLastUpdateTime. bySeconds	BYTE
	sLastUpdateTime. wMilliseconds	WORD
IntegratedIO.*	AnalogInputs. tAnalogInput_xx. bInputNotEnable	BOOL O canal de entrada não está habilitado na configuração.
	AnalogInputs. tAnalogInput_xx. bOverRange	BOOL O nível do sinal de entrada está acima do valor máximo definido para o tipo de entrada selecionado.
	AnalogInputs. tAnalogInput_xx. bOpenLoop	BOOL O nível do sinal de entrada está abaixo do mínimo (somente para o modo 4-20mA).
	AnalogOutputs. tAnalogOutput_xx. bOutputNotEnable	BOOL O canal de saída não está habilitado na configuração.
	AnalogOutputs. tAnalogOutput_xx. bOpenLoop	BOOL A impedância da carga conectada ao canal de saída está acima do máximo aceito (somente para o modo de saída de corrente).
	AnalogOutputs. tAnalogOutput_xx. bShortCircuit	BOOL A impedância da carga conectada ao canal de saída está abaixo do mínimo aceito (somente para o modo de saída de tensão).
	RTDInputs. tRtdInput_xx. bInputNotEnable	BOOL O canal de entrada não está habilitado na configuração.
	RTDInputs. tRtdInput_xx. bOverRange	BOOL A resistência está acima do valor máximo definido para o tipo selecionado.
	RTDInputs. tRtdInput_xx. bUnderRange	BOOL A resistência está abaixo do valor mínimo definido para o tipo selecionado (somente para sensores de temperatura).
		Modo de operação da VPN:

DG_XP3xx.tDetailed.*	Tipo	Descrição
		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	ENUM (BYTE)	Estado de conexão da VPN: DISCONNECTED (0): Dispositivo des- contado da VPN. CONNECTING (1): Dispositivo configu- rado, tentando conectar-se à VPN. CONNECTED (2): Dispositivo conectado a VPN.
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 comunicação VPN.
dwReceivedBytes	DWORD	Quantidade de bytes recebidos através da comunicação VPN.
CACertificate. CertificateName	STRING(64)	Nome do certificado CA.
CACertificate. CommonName	STRING(64)	Nome comum do certificado CA.
CACertificate. sStartDate. byDayOfMonth	BYTE	Data e hora de início do certificado CA.
CACertificate. sStartDate. byMonth	BYTE	
CACertificate. sStartDate. wYear	WORD	
CACertificate. sStartDate. byHours	BYTE	
CACertificate. sStartDate. byMinutes	BYTE	
CACertificate. sStartDate. bySeconds	BYTE	
CACertificate. sStartDate. wMilliseconds	WORD	
CACertificate. sExpirationDate. byDayOfMonth	BYTE	
CACertificate. sExpirationDate. byMonth	BYTE	

DG_XP3xx.tDetailed.*	Tipo	Descrição
CACertificate. sExpirationDate. wYear	WORD	Data e hora de expiração do certificado CA.
CACertificate. sExpirationDate. byHours	BYTE	
CACertificate. sExpirationDate. byMinutes	BYTE	
CACertificate. sExpirationDate. bySeconds	BYTE	
CACertificate. sExpirationDate. wMilliseconds	WORD	
CACertificate. dwValidDaysLeft	DWORD	Quantidade de dias restantes 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. byDayOfMonth	BYTE	Data e hora de início do certificado do dispositivo.
DeviceCertificate. sStartDate. byMonth	BYTE	
DeviceCertificate. sStartDate. wYear	WORD	
DeviceCertificate. sStartDate. byHours	BYTE	
DeviceCertificate. sStartDate. byMinutes	BYTE	
DeviceCertificate. sStartDate. bySeconds	BYTE	
DeviceCertificate. sStartDate. wMilliseconds	WORD	
DeviceCertificate. sExpirationDate. byDayOfMonth	BYTE	
DeviceCertificate. sExpirationDate. byMonth	BYTE	
DeviceCertificate. sExpirationDate. wYear	WORD	

DG_XP3xx.tDetailed.*		Tipo	Descrição
	DeviceCertificate. sExpirationDate. byHours	BYTE	Data e hora de expiração do certificado do dispositivo.
	DeviceCertificate. sExpirationDate. byMinutes	BYTE	
	DeviceCertificate. sExpirationDate. bySeconds	BYTE	
	DeviceCertificate. sExpirationDate. wMilliseconds	WORD	
	DeviceCertificate. dwValidDaysLeft	DWORD	Quantidade de dias restantes até a expiração do certificado do dispositivo.
	sDeviceKeyName	STRING(64)	Nome da chave privada do dispositivo.
Firewall.*	bServiceEnabled	BIT	Serviço de firewall habilitado.
	sLatsModificationDateUTC. byDayOfMonth	BYTE	Data e hora da última modificação das regras de firewall, em UTC.
	sLatsModificationDateUTC. byMonth	BYTE	
	sLatsModificationDateUTC. wYear	WORD	
	sLatsModificationDateUTC. byHours	BYTE	
	sLatsModificationDateUTC. byMinutes	BYTE	
	sLatsModificationDateUTC. bySeconds	BYTE	
	sLatsModificationDateUTC. wMilliseconds	WORD	
FTP.*	bServiceEnabled	BIT	Servidor FTP habilitado.
	dwConnectedClients	DWORD	Número de clientes FTP conectados simultaneamente.
	sLatsModificationDateUTC. byDayOfMonth	BYTE	Data e hora da última modificação das configurações do Servidor FTP, em UTC.
	sLatsModificationDateUTC. byMonth	BYTE	
	sLatsModificationDateUTC. wYear	WORD	
	sLatsModificationDateUTC. byHours	BYTE	
	sLatsModificationDateUTC. byMinutes	BYTE	
	sLatsModificationDateUTC. bySeconds	BYTE	
	sLatsModificationDateUTC. wMilliseconds	WORD	

Tabela 197: Descrição dos Diagnósticos Detalhados

Notas:

Códigos de Exceção: Os códigos de exceção gerados pelo RTS (Runtime System) são apresentados 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 (Watchdog 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 on-line 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 descarregado, pois as variáveis retentivas não podem ser realocadas.	0x0101	Limite de arrays excedido.
0x001B	Projeto descarregado 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 198: Códigos de Exceção RTS

Reset por Brownout: O diagnóstico de reset de queda de energia (brownout) só é verdadeiro quando a fonte de alimentação excede o limite mínimo requerido em suas características técnicas, permanecendo em baixa tensão, sem sofrer qualquer interrupção. O controlador identifica a quebra de tensão e indica o diagnóstico de falha da fonte de alimentação. Quando a

tensão é restabelecida, o controlador é reiniciado automaticamente e indica o diagnóstico de reinicialização.

Contador de Erro de Paridade: Quando a serial COM 1 é configurada sem paridade, esse contador de erros não será incrementado quando receber uma mensagem com uma paridade diferente. Neste caso, um erro de frame será indicado.

Partição do Usuário: A partição do usuário é uma área de memória reservada para o armazenamento de dados na UCP. Por exemplo: arquivos com extensão PDF, arquivos com extensão DOC e outros dados.

Entradas RTD: A tabela abaixo descreve o comportamento dos diagnósticos acima e abaixo da faixa de acordo com o tipo de entrada selecionado:

Diagnósticos	Escala 0 a 400 Ω		Escala 0 a 4000 Ω		Sensores de Platina tipo (Pt) $\alpha = 0.00385$		Sensores de Platina tipo (Pt) $\alpha = 0.003916$	
	Resist.	Cont.	Resist.	Cont.	Temp.	Cont.	Temp.	Cont.
Over range	>420 Ω (420 a 404,1 Ω)	4200 (4200 a 4041)	>4200 Ω (4200 a 4041 Ω)	>4200 (4200 a 4041)	>850 $^{\circ}\text{C}$	8500	>630 $^{\circ}\text{C}$	6300
No diagnostics	0 a 404 Ω	0 a 4040	0 a 4040 Ω	0 a 4040	-200 a 850 $^{\circ}\text{C}$	-2000 a 8500	-200 a 630 $^{\circ}\text{C}$	-2000 a 6300
Under range	-	-	-	-	<-200 $^{\circ}\text{C}$	-2000	<-200 $^{\circ}\text{C}$	-2000

Tabela 199: Diagnósticos de Entrada RTD

6.1.4. 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.

6.1.4.1. GetTaskInfo

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



Figura 197: 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 200: 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 201: 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. Manutenção Preventiva

- Deve ser verificado, a cada ano, se os cabos de interconexão estão conectados firmemente, sem acumulação de poeira, principalmente os dispositivos de proteção
- Em ambientes sujeitos a contaminação excessiva, o equipamento deve ser periodicamente limpo de poeira, detritos, etc.
- Aperto e limpeza do bloco conector a cada seis meses

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.

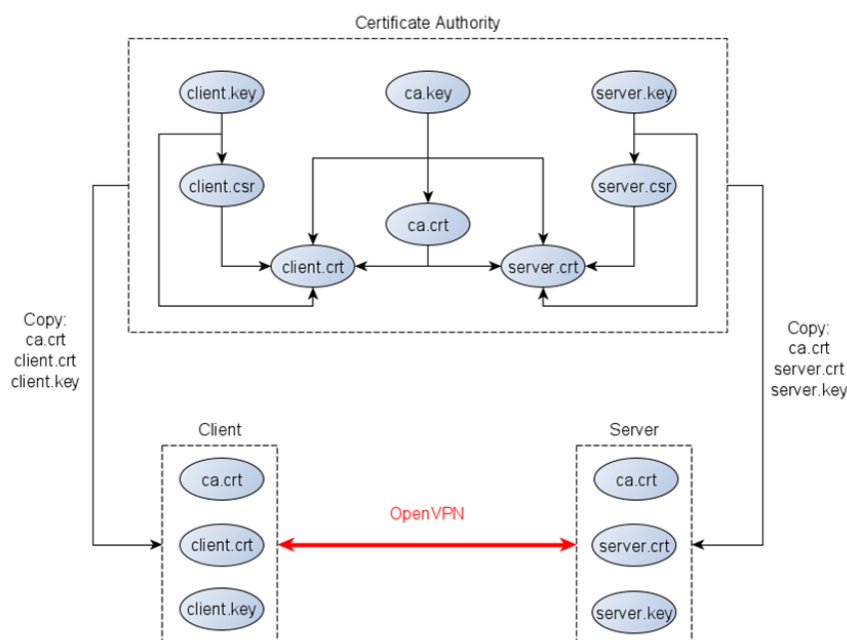


Figura 198: Fluxo de Geração de Certificados TLS

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

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

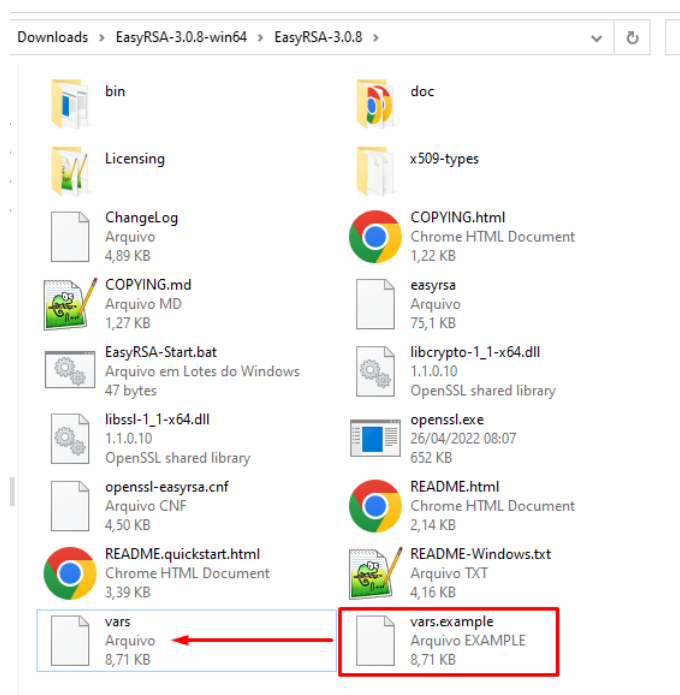


Figura 200: Geração de Certificado utilizando o Easy-RSA (passo 2)

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 201: 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 202: 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 203: 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 204: 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/tmp
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 205: 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 206: 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 207: 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 208: 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 209: 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 210: 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 211: 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 212: 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 213: 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 214: 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.