Manual de Utilização
Nexto XF
XF3xx

MU218000 Rev. D

26 de outubro de 2026

www.altus.com.br

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 à este manual . . . 1
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 . . . 10
2.2.5. Interface Serial RS-485 . . . 11
2.2.6. Interface CAN . . . 11
2.2.7. Interface USB . . . 12
2.2.8. Ethernet . . . 13
2.2.9. Interface do Cartão de Memória . . . 13
2.2.10. Alimentação . . . 14
2.2.11. Entradas Digitais . . . 14
2.2.12. Entradas Digitais Rápidas . . . 15
2.2.13. Saídas Digitais . . . 16
2.2.14. Saídas Digitais Rápidas . . . 17
2.2.15. Entradas Analógicas . . . 18
2.2.16. Saídas Analógicas . . . 19
2.3. Compatibilidade Com Outros Produtos . . . 20
2.4. Desempenho . . . 20
2.4.1. Tempo de Intervalo . . . 20
2.4.2. Tempos da Aplicação . . . 20
2.4.3. Tempo Para a Execução de Instruções . . . 20
2.4.4. Tempos de Inicialização . . . 20
2.5. Dimensões Físicas . . . 21
2.6. Dados de Compra . . . 22
2.6.1. Itens Integrantes . . . 22
2.6.2. Código de Produto . . . 22
2.7. Produtos Relacionados . . . 23
3. Instalação . . . 24
3.1. Instalação Mecânica . . . 25
3.1.1. Instalando o Controlador . . . 25

3.1.2. Removendo o Controlador . . . 26
3.1.3. Instalando e removendo o conector para o I/O integrado . . . 26
3.1.4. Instalando e removendo a terminação do I/O . . . 27
3.1.5. Instalando e removendo a tampa do I/O integrado . . . 28
3.2. Espaçamento entre a CPU e outros equipamentos no painel . . . 29
3.3. Montagem Horizontal/Vertical . . . 29
3.4. Instalação Elétrica . . . 30
3.5. Conexão à rede Ethernet . . . 33
3.5.1. Endereço IP . . . 33
3.5.2. ARP Gratuito . . . 33
3.5.3. Instalação do Cabo de Rede . . . 33
3.6. Instalação de rede Serial RS-485 e CAN . . . 34
3.7. Instalação do Cartão de Memória . . . 35
4. Programação Inicial . . . 36
4.1. Organização e Acesso à Memória . . . 36
4.2. Perfis de Projeto . . . 38
4.2.1. Perfil Fast . . . 38
4.2.2. Número Máximo de Tarefas . . . 38
4.3. Configuração da UCP . . . 39
4.4. Bibliotecas . . . 40
4.5. Inserindo uma instância de Protocolo . . . 40
4.5.1. MODBUS RTU . . . 40
4.5.2. MODBUS Ethernet . . . 42
4.6. Localizando o Dispositivo . . . 42
4.7. Login . . . 45
4.8. Modo Run . . . 47
4.9. Modo Stop . . . 48
4.10. Escrita e Forçamento de Variáveis . . . 48
4.11. Logout . . . 49
4.12. Upload de Projeto . . . 49
4.13. Estados de Operação da UCP . . . 51
4.13.1. Run . . . 51
4.13.2. Stop . . . 51
4.13.3. Breakpoint . . . 51
4.13.4. Exception . . . 51
4.13.5. Reset a Quente . . . 51
4.13.6. Reset a frio . . . 51
4.13.7. Reset de Origem . . . 51
4.14. Programas (POUs) e Listas de Variáveis Globais (GVLs) . . . 52
4.14.1. Programa MainPrg . . . 52
4.14.2. Programa StartPrg . . . 52
4.14.3. Programa UserPrg . . . 52
4.14.4. GVL IntegratedIO . . . 52
4.14.5. GVL System_Diagnostics . . . 53
4.14.6. GVL Disables . . . 53
4.14.7. GVL Qualities . . . 55
4.14.8. GVL ReqDiagnostics . . . 56
5. Configuração . . . 58

5.1. Device . . . 58
5.1.1. Gerenciamento de Usuários e Direitos de Acesso . . . 58
5.1.2. Configurações do CP . . . 58
5.2. Configuração da UCP . . . 59
5.2.1. Parâmetros Gerais . . . 59
5.2.2. Sincronização de Tempo . . . 60
5.2.2.1. SNTP . . . 61
5.2.2.2. Horário de Verão . . . 62
5.3. Configuração da Interface Serial . . . 62
5.3.1. COM 1 . . . 62
5.3.2. Configurações Avançadas . . . 63
5.4. Configuração das Interfaces Ethernet . . . 64
5.4.1. NET 1 . . . 64
5.4.2. NET 2 . . . 64
5.4.3. Modos de Operação das Interfaces Ethernet Integradas . . . 64
5.4.3.1. Modo Simples . . . 64
5.4.3.2. Modo NIC Teaming . . . 65
5.4.3.3. Modo Switch . . . 66
5.4.3.3.1. Modo Switch - Desabilitado . . . 66
5.4.3.3.2. Modo Switch - RSTP . . . 66
5.4.3.3.3. Modo Switch - MRP . . . 68
5.4.4. Portas TCP/UDP Reservadas . . . 70
5.4.5. Rotamento de Rede . . . 70
5.5. Configuração da Interface CAN (Controller Area Network) . . . 71
5.5.1. CAN . . . 71
5.6. Configuração das E/S Integradas . . . 71
5.6.1. Entradas Digitais . . . 72
5.6.2. Entradas Rápidas . . . 73
5.6.2.1. Contadores Rápidos . . . 75
5.6.2.1.1. Interrupções de Contador . . . 79
5.6.2.2. Interrupções Externas . . . 81
5.6.3. Saídas Rápidas . . . 83
5.6.3.1. VFO/PWM . . . 85
5.6.3.2. PTO . . . 87
5.6.4. Entradas Analógicas . . . 92
5.6.5. Saídas Analógicas . . . 93
5.6.6. Mapeamento de E/S . . . 94
5.7. Protocolos de Comunicação . . . 94
5.7.1. Comportamento dos Protocolos x Estados da UCP . . . 96
5.7.2. MODBUS Mestre RTU . . . 96
5.7.2.1. Configuração do protocolo MODBUS Mestre por Mapeamento Simbólico . . . 97
5.7.2.1.1. Parâmentros gerais do Protocolo MODBUS Mestre – Configuração por Mapeamento Simbólico . . . 97
5.7.2.1.2. Configuração dos Dispositivos – Configuração por Mapeamento Simbólico 99
5.7.2.1.3. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico 100
5.7.2.1.4. Configuração das Requisições – Configuração por Mapeamento Simbólico . 101
5.7.3. MODBUS Escravo RTU . . . 104
5.7.3.1. Configuração do Protocolo MODBUS Escravo por Mapeamento Simbólico . . . 104

5.7.3.1.1. MODBUS Slave Protocol General Parameters – Configuração por Mapeamento Simbólico . . . 105
5.7.3.1.2. Configuração das Requisições – Configuração por Mapeamento Simbólico . 107
5.7.4. MODBUS Ethernet . . . 108
5.7.5. MODBUS Ethernet Cliente . . . 110
5.7.5.1. Configuração do Protocolo MODBUS Ethernet Cliente por Mapeamento Simbólico 110
5.7.5.1.1. Parâmetros Gerais do protocolo MODBUS Cliente – Configuração por Mapeamento Simbólico . . . 111
5.7.5.1.2. Configuração dos Dispositivos – Configuração por Mapeamento Simbólico 112
5.7.5.1.3. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico 113
5.7.5.1.4. Configuração das Requisições – Configuração por Mapeamento Simbólico . 114
5.7.5.2. Disparo de Relações MODBUS Cliente de forma Acíclica . . . 118
5.7.6. MODBUS Ethernet Servidor . . . 118
5.7.6.1. Configuração do Protocolo MODBUS Ethernet Servidor por Mapeamento Simbólico 118
5.7.6.1.1. Parâmetros Gerais do Protocolo MODBUS Servidor – Configuração por Mapeamento Simbólico . . . 119
5.7.6.1.2. Diagnósticos MODBUS Servidor – Configuração por Mapeamento Simbólico . . . 120
5.7.6.1.3. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico 121
5.7.7. OPC DA Servidor . . . 122
5.7.7.1. Criando um Projeto para comunicação OPC DA . . . 124
5.7.7.2. Configurando um CP no Servidor OPC DA . . . 127
5.7.7.2.1. Importando uma Configuração de Projeto . . . 129
5.7.7.3. Variáveis de Status e Qualidade da comunicação OPC DA . . . 129
5.7.7.4. Limites da Comunicação com Servidor OPC DA . . . 130
5.7.7.5. Acessando dados através de um Cliente OPC DA . . . 131
5.7.8. OPC UA Servidor . . . 132
5.7.9. Mestre EtherCAT . . . 134
5.7.9.1. Instalando e inserindo dispositivos EtherCAT . . . 134
5.7.9.1.1. EtherCAT - Scan Devices . . . 135
5.7.9.2. Configuração do Mestre EtherCAT . . . 136
5.7.9.2.1. EtherCAT Master - General . . . 136
5.7.9.2.2. EtherCAT Master - Sync Unit Assignment . . . 138
5.7.9.2.3. EtherCAT Master - Overview . . . 138
5.7.9.2.4. EtherCAT Master - I/O Mapping . . . 138
5.7.9.2.5. EtherCAT Master - IEC Objects . . . 138
5.7.9.2.6. EtherCAT Master - Status / Information Tabs . . . 138
5.7.9.3. Configuração do escravo EtherCAT . . . 139
5.7.9.3.1. EtherCAT Slave - General . . . 139
5.7.9.3.2. EtherCAT Slave - Process Data . . . 142
5.7.9.3.3. EtherCAT Slave - Edit PDO List . . . 144
5.7.9.3.4. EtherCAT Slave - Startup Parameters . . . 144
5.7.9.3.5. EtherCAT Slave - I/O Mapping . . . 144
5.7.9.3.6. EtherCAT Slave - Status and Information . . . 145
5.7.10. EtherNet/IP . . . 145
5.7.10.1. Adição de Scanners e Adapters EtherNet/IP . . . 146
5.7.10.2. Configuração do Scanner EtherNet/IP . . . 148
5.7.10.2.1. Geral . . . 148
5.7.10.2.2. Conexões . . . 149

5.7.10.2.3. Assemblies . . . 151
5.7.10.2.4. EtherNet/IP: Mapeamento de E/S . . . 152
5.7.10.3. Configuração do Adapter EtherNet/IP . . . 152
5.7.10.3.1. Geral . . . 152
5.7.10.3.2. EtherNet/IP Adapter: Mapeamento de E/S . . . 153
5.7.10.4. Configuração do módulo EtherNet/IP . . . 153
5.7.10.4.1. Assemblies . . . 154
5.7.10.4.2. EtherNet/IP Module: Mapeamento de E/S . . . 154
5.7.11. CANopen Manager . . . 154
5.7.11.1. Instalando e inserindo dispositivos CANopen . . . 154
5.7.11.2. Configuração do CANOpen Manager . . . 155
5.7.11.3. Configuração do escravo CANopen . . . 156
5.7.12. PROFINET Controller . . . 157
5.8. Desempenho de Comunicação . . . 157
5.8.1. Servidor MODBUS . . . 157
5.8.1.1. Interfaces Integradas da UCP . . . 158
5.8.2. Servidor OPC UA . . . 158
5.9. Desempenho do Sistema . . . 158
5.9.1. Cartão de Memória . . . 159
5.10. Relógio RTC . . . 159
5.10.1. Blocos funcionais e funções para leitura e escrita do RTC . . . 159
5.10.1.1. Funções de leitura do RTC . . . 160
5.10.1.1.1. GetDateAndTime . . . 160
5.10.1.1.2. GetTimeZone . . . 160
5.10.1.1.3. GetDayOfWeek . . . 161
5.10.1.2. Funções de escrita do RTC . . . 162
5.10.1.2.1. SetDateAndTime . . . 162
5.10.1.2.2. SetTimeZone . . . 163
5.10.2. Estruturas de dados do RTC . . . 164
5.10.2.1. EXTENDED_DATE_AND_TIME . . . 165
5.10.2.2. DAYS_OF_WEEK . . . 165
5.10.2.3. RTC_STATUS . . . 165
5.10.2.4. TIMEZONESETTINGS . . . 166
5.10.3. Limites de operação do RTC . . . 166
5.11. Memória de Arquivos de Usuário . . . 167
5.12. Blocos Funcionais e Funções . . . 168
5.12.1. Blocos Funcionais especiais para Comunicação Serial . . . 168
5.12.1.1. SERIAL_CFG . . . 172
5.12.1.2. SERIAL_GET_CFG . . . 174
5.12.1.3. SERIAL_GET_CTRL . . . 176
5.12.1.4. SERIAL_GET_RX_QUEUE_STATUS . . . 177
5.12.1.5. SERIAL_PURGE_RX_QUEUE . . . 178
5.12.1.6. SERIAL_RX . . . 180
5.12.1.7. SERIAL_RX_EXTENDED . . . 182
5.12.1.8. SERIAL_SET_CTRL . . . 184
5.12.1.9. SERIAL_TX . . . 186
5.12.2. Timer Retain . . . 188
5.12.2.1. TOF_RET . . . 188

5.12.2.2. TON_RET . . . 189
5.12.2.3. TP_RET . . . 191
5.13. Página Web de Usuário . . . 192
5.14. Página Web de Sistema . . . 192
5.14.1. Informações Gerais . . . 192
5.14.2. Gerenciamento . . . 193
5.14.2.1. Sistema . . . 193
5.14.2.1.1. Atualização de Firmware . . . 193
5.14.2.1.2. Ajuste do Relógio . . . 194
5.14.2.1.3. Controle do Sistema . . . 194
5.14.2.2. Rede . . . 195
5.14.2.2.1. Configuração . . . 195
5.14.2.2.2. Sniffer . . . 197
5.14.2.3. SNMP . . . 198
5.14.2.3.1. SNMP nas UCPs Nexto . . . 198
5.14.2.3.2. Configuração SNMP . . . 199
5.14.2.3.3. Usuário e Comunidades SNMP . . . 200
5.14.2.4. Dispositivos USB . . . 201
5.14.2.4.1. Dispositivos de Armazenamento em Massa . . . 202
5.14.2.4.2. Conversor USB para RS-232 . . . 204
5.14.2.4.3. Dispositivos Modem . . . 205
5.14.2.4.4. Adaptadores Wi-Fi . . . 207
5.14.2.4.5. Adaptadores Ethernet . . . 210
5.14.2.4.6. Função de Usuário para Controle da Interface USB . . . 212
5.14.2.5. Firewall . . . 213
5.14.2.5.1. Configurações Gerais . . . 213
5.14.2.5.2. Regras de Usuário . . . 215
5.14.2.5.3. Modo Switch ou NIC Teaming . . . 217
5.14.2.6. OpenVPN . . . 217
5.14.2.6.1. Configuração do OpenVPN . . . 218
5.14.2.6.2. Arquivos de Segurança . . . 221
5.14.2.6.3. Status . . . 222
5.14.2.6.4. Arquivos para Baixar . . . 224
5.14.2.6.5. Configuração de Arquiteturas . . . 224
5.14.2.7. Servidor FTP . . . 226
5.14.2.7.1. Configurações Gerais . . . 227
5.14.2.7.2. Configurações de Usuário . . . 227
5.14.2.7.3. Status . . . 228
5.14.2.8. Cartão de Memória . . . 228
5.14.2.8.1. Configuração . . . 229
5.15. Docker . . . 233
5.15.1. Como acessar um Container Docker . . . 234
5.15.1.1. Como Acessar o Portainer . . . 234
5.15.2. Quick Setup . . . 235
5.15.2.1. Informações do ambiente Local . . . 235
5.15.3. Dashboard de Ambiente . . . 235
5.15.3.1. Menu de Containers . . . 236
5.15.3.2. Menu de Images . . . 237

5.15.4. Criando e Configurando um Container . . . 237
5.15.4.1. Configurando um Container . . . 238
5.15.4.1.1. Menu Command & Logging . . . 238
5.15.4.1.2. Menu Network . . . 239
5.15.4.1.3. Menu Runtime & Resources . . . 239
5.15.4.1.4. Menu Capabilities . . . 240
5.15.4.2. Utilizando um Container . . . 241
5.15.4.3. Como criar uma imagem a partir de um container existente salvando os seus dados 242
5.15.4.4. Monitorando um Container . . . 242
5.15.4.5. Características Técnicas . . . 243
5.15.4.5.1. Memória Flash . . . 243
5.15.4.5.2. Memória RAM . . . 243
5.15.4.5.3. UCP . . . 243
5.15.4.6. Como acessasr as pastas do Docker . . . 244
5.15.4.6.1. Mastertool . . . 244
5.15.4.7. Usando memória externa como volume Docker . . . 244
5.15.4.8. Solução de Problemas . . . 245
5.15.4.8.1. Falha ao instalar Container devido a certificado expirado ou inválido . . . 245
5.15.4.8.2. Portainer inacessível depois de uma reinicialização da UCP . . . 245
6. Manutenção . . . 246
6.1. Diagnósticos . . . 246
6.1.1. Diagnósticos via LEDs . . . 246
6.1.1.1. SYS (Sistema) . . . 246
6.1.1.2. RUN (Aplicação) . . . 246
6.1.1.3. SD (Cartão de Memória) . . . 246
6.1.1.4. DG (Diagnóstico) . . . 247
6.1.1.5. WD (Watchdog) . . . 247
6.1.1.6. PWR (Alimentação) . . . 248
6.1.1.7. CAN . . . 248
6.1.1.8. RS485 . . . 248
6.1.1.9. USB . . . 248
6.1.1.10. LEDs Conector RJ45 . . . 248
6.1.1.11. DO (Saídas Digitais) . . . 249
6.1.1.12. DI (Entradas Digitais) . . . 249
6.1.1.13. AO (Saídas Analógicas) . . . 249
6.1.1.14. AI (Entradas Analógicas) . . . 249
6.1.2. Diagnósticos via Variáveis . . . 250
6.1.2.1. Diagnósticos Resumidos . . . 250
6.1.2.2. Diagnósticos Detalhados . . . 251
6.1.2.2.1. Grupo de diagnósticos detalhados: Target . . . 251
6.1.2.2.2. Grupo de diagnósticos detalhados: Hardware . . . 252
6.1.2.2.3. Grupo de diagnósticos detalhados: Exception . . . 252
6.1.2.2.4. Grupo de diagnósticos detalhados: WebVisualization . . . 254
6.1.2.2.5. Grupo de diagnósticos detalhados: RetainInfo . . . 254
6.1.2.2.6. Grupo de diagnósticos detalhados: Reset . . . 254
6.1.2.2.7. Grupo de diagnósticos detalhados: Thermometer . . . 255
6.1.2.2.8. Grupo de diagnósticos detalhados: Serial . . . 255
6.1.2.2.9. Grupo de diagnósticos detalhados: CAN . . . 256

6.1.2.2.10. Grupo de diagnósticos detalhados: USB . . . 256
6.1.2.2.11. Grupo de diagnósticos detalhados: Ethernet . . . 257
6.1.2.2.12. Grupo de diagnósticos detalhados: UserFiles . . . 260
6.1.2.2.13. Grupo de diagnósticos detalhados: UserLogs . . . 260
6.1.2.2.14. Grupo de diagnósticos detalhados: MemoryCard . . . 260
6.1.2.2.15. Grupo de diagnósticos detalhados: Application . . . 261
6.1.2.2.16. Grupo de diagnósticos detalhados: ApplicationInfo . . . 261
6.1.2.2.17. Grupo de diagnósticos detalhados: SNTP . . . 261
6.1.2.2.18. Grupo de diagnósticos detalhados: Integrated IO . . . 262
6.1.2.2.19. Grupo de diagnósticos detalhados: OpenVPN . . . 262
6.1.2.2.20. Grupo de diagnósticos detalhados: Firewall . . . 263
6.1.2.2.21. Grupo de diagnósticos detalhados: FTP . . . 263
6.2. Botão de usuário da UCP . . . 264
6.2.1. Diagnósticos através de Blocos Funcionais . . . 265
6.2.1.1. GetTaskInfo . . . 265
6.3. Manutenção Preventiva . . . 266
7. Apendices . . . 267
7.1. Gerenciamento de Certificados e Chaves TLS . . . 267
7.1.1. Geração de Certificados por Easy-RSA . . . 267
7.1.2. Geração de Certificados por OpenSSL . . . 272
7.1.3. Geração de Chave TA pelo OpenVPN . . . 274

1. Introdução

O Nexto XF é um poderoso Controlador Lógico Programável (CLP) pertencente à família de controladores e módulos de E/S da Linha XF. O Nexto XF oferece alta velocidade de processamento em um design compacto com E/S integradas, facilmente expansível com módulos de E/S externos. Diversas opções estão disponíveis, permitindo que você escolha a melhor solução, independentemente da complexidade da aplicação.

Este portfólio de produtos é voltado para sistemas de controle distribuído de pequeno, médio e grande porte, oferecendo modelos com uma variedade de opções de pontos de E/S integrados, incluindo entradas e saídas digitais e analógicas concentradas em um único controlador. Para necessidades adicionais de E/S, o sistema pode ser facilmente expandido por meio de módulos de expansão conectados ao barramento de alta velocidade (consulte a seção Produtos Relacionados). Além disso, o número de pontos de E/S pode ser ainda mais aumentado por meio de dispositivos de E/S remotos (distribuídos) que se comunicam via protocolos como CANopen, EtherNet/IP, PROFINET, EtherCAT e MODBUS.

O Nexto XF pode ser utilizado em aplicações como infraestrutura, marítimo, automação predial e industrial (alimentos, têxtil, embalagens, máquinas em geral), tratamento de água e esgoto e inúmeras soluções OEM. As soluções integradas de firewall e VPN adicionam segurança para garantir a integridade dos dados e reduzir os riscos associados a ameaças cibernéticas. Além disso, o controlador é uma solução ideal para complementar grandes aplicações juntamente com o portfólio do Sistema Nexto, ampliando a gama de aplicações utilizando a mesma tecnologia e ambiente de engenharia. Esta é uma vantagem significativa para OEMs e integradores de sistemas com diferentes necessidades e escalas de aplicações.



Figura 1: Nexto XF325

1.1. Documentos relacionados à este manual

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

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

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

Código	Descrição	Idioma
CE118100	Nexto XF Technical Characteristics	Inglês
CT118100	Características Técnicas Nexto XF	Português
MU214622	LibSparkPlug User Manual	Inglês
MU218600	Nexto XF XF3xx User Manual	Inglês
MU218000	Manual de Utilização Nexto XF XF3xx	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
NAP169	RSTP in Nexto CPUs	Inglês

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 ou visite-nos no endereço Av. Theodomiro Porto da Fonseca, 3101, lote 01 – São Leopoldo/RS – Brasil, CEP 93022-715. 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 do controlador, 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 todas as características técnicas dos controladores Nexto XF.

2.1. Painéis e Conexões

A figura a seguir mostra o painel frontal do XF325:

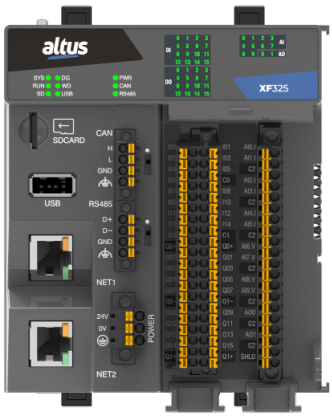


Figura 2: Painel frontal XF325

O painel frontal contém a identificação do estado geral do sistema, das interfaces de E/S e de comunicação disponíveis nos controladores Nexto XF. As interfaces de E/S possuem um LED para cada ponto, indicando o estado lógico, enquanto as interfaces de comunicação possuem um LED para indicar a atividade. A disponibilidade dessas interfaces em cada modelo é descrita na próxima seção. A tabela a seguir mostra a descrição dos LEDs. Para obter mais informações sobre o estado e o significado dos LEDs, consulte o capítulo de Manutenção.

LED	Descrição
SYS	Status do CLP
DG	Status de Diagnóstico
RUN	Status de execução da aplicação
WD	Status do Watchdog
SD	Status da interface do cartão microSD
USB	Status da interface USB
PWR	Status da fonte de alimentação interna
CAN	Status da interface CAN
RS485	Status da interface serial RS-485
DI 0-6	Estado lógico das entradas digitais padrão
DI 7-15	Estado lógico das entradas digitais padrão/rápidas
DO 0-7	Estado lógico das saídas digitais padrão
DO 8-15	Estado lógico das saídas digitais padrão/rápidas
AI 0-9	Estado lógico das entradas analógicas (O LED respectivo acende quando o IO está configurado)
AO 0-1	Estado lógico das saídas analógicas (O LED respectivo acende quando o IO está configurado)
ETH	Status de link e atividade da interface Ethernet (Montado no conector RJ45)

Tabela 2: Descrição dos LEDs

2.2. Características do Produto

2.2.1. Características Gerais

	XF300-B	XF300	XF315	XF325	XF325-W
Entradas digitais	16 (sendo 9 rápidas)				
Saídas digitais	16 (sendo 8 rápidas)				
Número máx. de contadores rápidos	3				
Número máx. de interrupções externas	9				
Número máx. de saídas PTO	3				
Número máx. de saídas VFO/PWM	8				
Entradas analógicas de Tensão	-		4	4	4
Entradas analógicas de Corrente	-		6	6	6
Saídas analógicas V/I	-		-	2	2
Interfaces Ethernet TCP / IP	2				
Suporte a redundância de interfaces Ethernet TCP/IP	Sim				
Interface serial RS-485	1				
Interface CAN	1				
Porta USB Host	1				
Número máximo de módulos de expansão de E/S suportados	10				
Páginas web de usuário (Webvisu)	No				Sim, 2048 tags
Docker Engine	Não	Sim			
Canais de E/S	128	512			
Tags de comunicação	512	2048			4096
Pontos mapeados	20480				
Protocolos de comunicação	Ver tabela Protocolos				
FTP	Sim				
Firewall	Sim				
VPN	Sim				
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				
Número máximo de tarefas	16				
Cão de guarda	Sim				
Indicação de estado e diagnóstico	LEDs, página da web e memória interna da UCP				
Relógio de tempo real (RTC)	Sim Resolução de 1 ms, máx. variação de 3 segundos por dia, tempo de retentividade de 14 dias.				
Isolações					
Terra de proteção ⏚ para todos	1000 Vac / 1 minuto (1414 Vdc / 1 minuto)				
Lógica/USB para todos	1000 Vac / 1 minuto (1414 Vdc / 1 minuto)				
Fonte de alimentação para todos	1000 Vac / 1 minuto (1414 Vdc / 1 minuto)				
NET 1 para todos	1000 Vac / 1 minute (1414 Vdc / 1 minute)				
NET 2 para todos	1000 Vac / 1 minute (1414 Vdc / 1 minute)				
CAN para todos	1000 Vac / 1 minute (1414 Vdc / 1 minute)				
RS-485 para todos	1000 Vac / 1 minute (1414 Vdc / 1 minute)				
E/S analógicas para todos	1000 Vac / 1 minuto (1414 Vdc / 1 minuto)				
Entradas digitais para todos	1000 Vac / 1 minuto (1414 Vdc / 1 minuto)				

Grupo de entradas digitais I0x para I1x **Saídas digitais para todos**	1000 Vac / 1 minuto (1414 Vdc / 1 minuto) 1000 Vac / 1 minuto (1414 Vdc / 1 minuto)
Dissipação máxima de potência	10 W
Área máxima do cabeamento	0,5 mm^2 (20 AWG) com terminal tubular 1,5 mm^2 (16 AWG) sem terminal tubular
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 armazenamento	-40 a 70 °C
Umidade relativa de operação e armazenamento	10% a 95%, sem condensação
Resistência à vibração (IEC 60068-2-6, senoidal)	3,5 mm de 5 a 8,4 Hz 1 G de 8,4 a 500 Hz 10 varreduras em cada eixo, 1 oitava por minuto
Resistência ao choque (IEC 60068-2-27, semi-senoidal)	15 G por 11 ms, 6 choques em cada um dos 3 eixos
Dimensões do produto (L x A x P)	112,2 x 122,8 x 123,0 mm (102,8 x 122,8 x 110,8 mm sem terminação do barramento e sem a tampa do I/O)
Dimensões da embalagem (L x A x P)	128 x 138 x 138 mm
Peso	513,5g
Peso com embalagem	675,1g

Tabela 3: Características Gerais

Notas:

Canais de E/S: Número máximo de canais individuais de entrada e saída utilizados pela aplicação, listados na aba *Mapeamento de E/S* de cada janela do editor de dispositivo. Um canal pode conter múltiplos pontos de E/S (especialmente no caso de E/S digitais), portanto a contagem de canais é menor que o número de pontos de E/S. O número de Canais de E/S utilizados pela aplicação é informado pelo Mastertool na janela *Mensagens*, dentro da categoria *Software Metrics*.

Tags de Comunicação: Número máximo de variáveis explicitamente designadas para troca de dados entre o controlador e sistemas externos utilizando OPC UA, OPC DA e Protocolo CODESYS - ARTI, configuradas no *Gerenciador de Comunicação* ou na *Configuração de Símbolos*. O número de *Tags de Comunicação* utilizadas pela aplicação é informado pelo Mastertool na janela *Mensagens*, dentro da categoria *Software Metrics*.

Tags de Webvisu: Número máximo de variáveis de aplicação utilizadas no painel de visualização web. O número de *Webvisu Tags* utilizadas é informado pelo Mastertool na janela *Mensagens* dentro da categoria *Software Metrics*.

Pontos Mapeados: Número máximo de pontos de comunicação mapeados especificamente nos protocolos MODBUS, IEC 104, DNP3 e IEC 61850 (quando suportado). Cada mapeamento pode conter um ou mais pontos mapeados, dependendo do tamanho dos dados. Isso varia conforme sejam utilizadas variáveis simples ou variáveis do tipo ARRAY. Cada variável simples, bem como cada índice de um ARRAY, é contado 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, mesmo ocupando dois endereços/registros consecutivos no driver.

Isolação: O termo *Lógica* refere-se ao circuito de processamento que é isolado eletricamente das interfaces, com exceção da interface USB.

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.

Número de módulos de E/S: O número máximo de módulos de E/S suportados pela CPU depende da potência consumida no barramento. Utilize o Mastertool para configurar o número de módulos e dimensionar o consumo de potência no barramento.

2.2.2. Normas e Certificações

Normas e Certificações	
IEC	61131-2: Industrial-process measurement and control - Programmable controllers - Part 2: Equipment requirements and tests 61131-3: Programmable controllers - Part 3: Programming languages
	DNV Type Approval – DNV-CG-0339 (TAA000013D)
	2014/30/EU (EMC) 2011/65/EU and 2015/863/EU (ROHS)
	S.I. 2016 No. 1091 (EMC) S.I. 2012 No. 3032 (ROHS)
	Ordinary Locations: cULus LISTED, E473496 Hazardous Locations: cULus LISTED, E536282 Class I, Division 2, Groups A, B, C, D, T4A

Tabela 4: Normas e Certificações

2.2.3. Memória

	XF300-B	XF300	XF315	XF325	XF325-W
Memória de variáveis de entrada endereçáveis (%I)	64 kB				
Memória de variáveis de saída endereçáveis (%Q)	64 kB				
Memória de variáveis de representação direta (%M)	32 kB				
Memória de variáveis simbólicas	10 MB				32 MB
Memória de programa	32 MB				
Memória total **Memória de programa +** **Memória de código fonte (backup) +** **Memória de arquivos de Webvisu**	256 MB				
Memória retentiva/persistente	128 kB				
Memória de arquivos de usuário **Memória da UCP +** **Memória do Docker**	2,5 GB				

Tabela 5: Memória

Notas:

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

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

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

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

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

ATENÇÃO

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

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

Comando/Operação	Variável Simbólica	Variável Retentiva	Variável Persistente
Ciclo de energização	-	X	X
Reset a quente	-	X	X
Reset a frio	-	-	X
Reset de Origem	-	-	-
Download	-	-	X
Alteração Online	X	X	X
Limpar tudo	-	-	X
Runtime/Watchdog Reset	-	-	-

Tabela 6: Comportamento de variável após evento

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

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

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

Memória de arquivos de usuário: Essa área da memória é destinada ao armazenamento de arquivos, como: doc, pdf, imagens, entre outros; ou seja, permite a gravação de dados como se fosse um cartão de memória. Maiores informações na seção Memória de Arquivos de Usuário.

2.2.4. Protocolos

A tabela a seguir descreve os limites de tipos e quantidades (número de instâncias) de cada protocolo. Observe que existem outros limites operacionais relacionados às capacidades de comunicação (número máximo de *Canais de E/S*, *Tags de Comunicação* e *Pontos Mapeados*), os quais são informados na tabela Características Gerais.

	Interface	XF300-B	XF300	XF315	XF325	XF325-W
EtherCAT Mestre **PROFINET Controller** **EtherNet/IP Scanner**	NET 1 / NET 2	✗	Até 1 instância			
EtherNet/IP Adapter	NET 1 / NET 2	Até 1 instância	Até 2 instâncias			
PROFINET Device	-	✗	✗			
CANopen Mestre **SAE J1939**	CAN	Até 1 instância	Até 1 instância			
CAN low level	CAN	✓	✓			
CANopen Escravo	-	✗	✗			
Protocolo Aberto	COM 1 / USB	✓	✓			
MODBUS RTU Mestre Simbólico **MODBUS RTU Escravo Simbólico**	COM 1	Até 1 instância	Até 1 instância			
MODBUS TCP Cliente Simbólico **MODBUS TCP Servidor Simbólico** **MODBUS RTU over TCP Cliente Simbólico** **MODBUS RTU over TCP Servidor Simbólico**	NET 1 / NET 2	Até 8 instâncias	Até 8 instâncias			
IEC 60870-5-104 Cliente	-	✗	✗			
IEC 60870-5-104 Servidor	-	✗	✗			
DNP3 Cliente	-	✗	✗			
DNP3 Servidor	-	✗	✗			
OPC DA Servidor	NET 1 / NET 2 / USB	✓	✓			
OPC UA Servidor	NET 1 / NET 2 / USB	✓	✓			
SNMP Agente	NET 1 / NET 2 / USB	✓	✓			
MQTT Cliente	NET 1 / NET 2 / USB	✓	✓			
Sparkplug B	NET 1 / NET 2 / USB	✓	✓			
SNTP Cliente (para sincronismo de relógio)	NET 1 / NET 2 / USB	✓	✓			
OpenVPN Cliente	NET 1 / NET 2 / USB	✓	✓			
OpenVPN Servidor	NET 1 / NET 2 / USB	✓	✓			
FTP Servidor	NET 1 / NET 2 / USB	✓	✓			
RSTP	NET 1 / NET 2	✓	✓			
MRP	NET 1 / NET 2	✗	✓			

Tabela 7: Protocolos

Notas:
USB: É necessário o uso de um conversor serial, adaptador Wi-Fi ou modem 3G/4G.

2.2.5. Interface Serial RS-485

	RS-485
Conector	Bloco de terminação de 4 pinos (D+, D-, GND e shield)
Interface física	Barramento RS-485
Máx. Transmissores RS-485	32
Terminação	Sim (com chave mecânica)
Taxa de transmissão	2400, 4800, 9600, 19200, 38400, 57600, 115200 bps
Isolação **Lógica para porta serial** **Porta Serial para terra de proteção** ⏚	 1000 Vac / 1 minuto 1000 Vac / 1 minuto

Tabela 8: Característica da Interface Serial RS-485

Nota:

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

2.2.6. Interface CAN

	CAN
Conector	Bloco de terminação de 4 pinos (H, L, GND e shield)
Interface física	Barramento CAN
Normas suportadas	CAN 2.0A 2.0B (identificadores de 11-bit e 29-bit)
Max. número de nós	64
Terminação	Sim (com chave mecânica)
Taxa de transmissão	10, 20, 50, 100, 125, 250, 500, 800, 1000 kbit/s
Isolação **Lógica para CAN** **CAN terra de proteção** ⏚	 1000 Vac / 1 minuto 1000 Vac / 1 minuto

Tabela 9: Característica da Interface CAN

2.2.7. Interface USB

	USB
Conector	USB A fêmea
Interface física	USB V2.0
Taxa de transmissão	1.5 Mbps (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
Isolação **Lógica para USB** **USB para terra de proteção** ⏚	 Não isolada 1000 Vac / 1 minuto

Tabela 10: Características da Interface USB

Notas:
USB RS-232 Conversor Serial: Veja a lista de dispositivos suportados na seção Conversor USB para RS-232.
USB 3G/4G Modem: Veja a lista de dispositivos suportados na seção Dispositivos Modem.
USB Adaptador Wi-Fi: Veja a lista de dispositivos suportados na seção Adaptadores Wi-Fi.
USB Adaptador Ethernet: Veja a lista de dispositivos suportados na seção Adaptadores Ethernet.

ATENÇÃO:

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

2.2.8. Ethernet

	Ethernet
Interfaces	NET 1 e NET 2
Conector	RJ45 fêmea blindado
Auto crossover	Sim
Máximo comprimento de cabo	100 m
Tipo de cabo	UTP ou ScTP, categoria 5
Taxa de transmissão	10/100 Mbps
Camada física	10/100 BASE-TX
Camada de enlace	LLC
Camada de rede	IP
Camada de transporte	TCP (protocolo de controle de transmissão) UDP (protocolo de datagrama de usuário)
Modo de operação	Single / NIC Teaming / Switch
Diagnósticos	LED (Link/Atividade)
Isolação	
Interface Ethernet para lógica	1000 Vac / 1 minuto
Interface Ethernet para Interface Ethernet	1000 Vac / 1 minuto
Interface Ethernet para terra de proteção ⏚	1000 Vac / 1 minuto

Tabela 11: Caracteristicas da Interface Ethernet

2.2.9. Interface do Cartão de Memória

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

	Cartão de memória
Capacidade máxima	32 GB
Capacidade mínima	2 GB
Tipo	microSD / microSDHC
Sistema de arquivos	FAT32
Remoção do cartão de forma segura	Sim (via botão e página web)

Tabela 12: Característica da Interface do Cartão de Memória

Notas:

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

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

Sistema de arquivos: É recomendado formatar o cartão de memória utilizando a própria página web.

Atenção:

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

2.2.10. Alimentação

	Fonte de alimentação
Tensão de entrada nominal	24 Vdc
Tensão de entrada	18 a 30 Vdc
Máxima potência de entrada	25,2 W
Máxima corrente de entrada	1,4 A a 18 Vdc
Máxima corrente de surto (in-rush)	15 A por 1 ms
Tempo máximo de interrupção	10 ms a 18 Vdc
Número de módulos de E/S suportados pela fonte integrada	10
Máxima corrente no barramento XF	2,2 A
Proteções de entrada	Inversão de polaridade Sobre tensão Curto-circuito (fusível) Sobrecarga e curto-circuito no barramento
Isolações **Entrada para lógica** **Entrada para terra de proteção** ⏚	 1000 Vac / 1 minuto 1000 Vac / 1 minuto

Tabela 13: Característica da Alimentação

2.2.11. Entradas Digitais

	Entradas digitais
Número de entradas	7
Tipo das entradas	Optoisolada, sink/source
Identificação no conector	I00 a I06
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 kΩ
Máxima corrente de entrada	6,2 mA @ 30 Vdc
Indicação do estado da entrada	Sim
Tempo de resposta	2 ms
Filtro de entrada	Desabilitado ou 2 ms a 255 ms por software
Isolação **Entrada para lógica** **Entrada para terra de proteção** ⏚	 1000 Vac / 1 minute 1000 Vac / 1 minute

Tabela 14: Característica das Entradas Digitais Padrão

Nota:

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

2.2.12. Entradas Digitais Rápidas

	Entradas digitais rápidas
Número de entradas rápidas	9
Tipo das entradas rápidas	Optoisolada, sink/source
Número máximo de contadores rápidos	3 (Incremento/Decremento, A-pulso, B-direção e Z-zero)
Número máximo de Encoders	3 (Quadratura A, B e Z)
Número máximo de interrupções externas	9 (Subida, descida ou ambos)
Identificação no conector	I07 a I15
Tensão de entrada	24 Vdc 15 a 30 Vdc para nível lógico 1 0 a 5 Vdc para nível lógico 0
Impedância de entrada	3.9 kΩ
Máxima corrente de entrada	7.3 mA a 30 Vdc
Modo de configuração	**Modos para 1 entrada:** Entrada digital padrão Interrupção externa **Modos para 3 entradas:** Contador 0: Quadratura ou Up/Down (I07, I08, e I09) Contador 1: Quadratura ou Up/Down (I10, I11, e I12) Contador 2: Quadratura ou Up/Down (I13, I14, e I15)
Controle do sentido de contagem	Apenas por hardware
Borda de detecção da entrada de contagem	Subida, ativa em nível lógico 1 - padrão (permite outras configurações)
Formato dos dados	Inteiros de 32 bit com sinal
Limite de operação	De -2.147.483.648 até 2.147.483.647
Frequência máxima de entrada	250 kHz
Largura de pulso mínima @24 Vdc	2 μs
Isolação **Entrada para lógica** **Entrada para terra de proteção** ⏚	 1000 Vac / 1 minuto 1000 Vac / 1 minuto

Tabela 15: Característica das Entradas Rápidas

2.2.13. Saídas Digitais

	Saídas digitais
Número de saídas	8
Tipo das saídas	Chave a transistor, optoisolada, source
Identificação no conector	Q00 to Q07
Corrente máxima	1,5 A por saída (individual - não protegida) 6 A total
Corrente de fuga	35 μA
Resistência de saída	100 mΩ
Fonte de alimentação externa	19,2 a 30 Vdc
Tempo de comutação	3 μs - transição desligado para ligado 50 μs - transição ligado para desligado (carga de 400 Ω a 24 Vdc)
Frequência máxima de comutação	250 Hz
Parâmetros configuráveis	Sim
Indicação do estado de saída	Sim (LED)
Proteções de saída	Sim (contra surtos de tensão)
Isolações	
Saída para lógica	1000 Vac / 1 minuto
Saída para terra de proteção ⏚	1000 Vac / 1 minuto

Tabela 16: Característica das Saídas Digitais

Notas:

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

CUIDADO

As saídas não possuem proteção contra sobrecorrente ou curto-circuito.

2.2.14. Saídas Digitais Rápidas

	Saídas digitais rápidas	
Número de saídas	8	
Tipo das saídas	Chave a transistor, optoisolada, source	
Identificação no conector	Q08 a Q15	
Número máx. de saídas PTO	3	
Número máximo de saídas VFO/PWM	8 (se não usar PTO) 7 (ao usar 1 PTOs) 6 (ao usar 2 PTOs) 5 (ao usar 3 PTOs)	
Corrente máxima	3 a 500 Hz: 1,5A por saída (6,0A total) 500 a 250 kHz: 0,5A por saída	
Frequência mínima de geração de pulsos	3 Hz @ 60 mA	
Frequência máxima de geração de pulsos	250 kHz @ 60 mA	
Mínima largura de pulso @ 24 Vdc	Carga mínina	Pulso positivo
	400 Ω	240 ns
Tempo de comutação	0,2 μs - transição desligado para ligado 0,8 μs - transição ligado para desligado (400 Ω carga a 24 Vdc)	
Indicação do estado da saída	Sim (LED)	
Proteções	Surto de tensão (TVS)	
Tensão de operação	19,2 a 30 Vdc	
Impedância de saída	200 mΩ	
Modos de saída	Saída padrão VFO/PWM PTO (somente Q08, Q09 e Q14)	
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 (3 Hz a 250 kHz, +/-1%); Escrita do duty cycle das saídas (1% a 100%); Início / fim de operação das saídas; Diagnósticos de saídas rápidas;
Isolações		
Saída para lógica	1000 Vac / 1 minuto	
Saída para terra de proteção ⏚	1000 Vac / 1 minuto	

Tabela 17: Característica das Saídas Rápidas

2.2.15. Entradas Analógicas

	Entradas analógicas
Número de entradas	10 (sendo 6 corrente e 4 tensão)
Tipo de entrada	Dedicada como tensão ou corrente, não diferencial (*single-ended*), com configuração individual por canal
Formato dos dados	16 bits em complemento de dois, justificado à esquerda
Resolução do conversor	Monotonicidade garantida de 12 bits, sem códigos perdidos
Tempo de conversão	400 μs (todos os canais habilitados)
Indicação do status do canal	Sim (LED)
Proteções de módulo	Sim, proteção contra surtos de tensão e inversão de polaridade
Isolação **Canal para lógica** **Canal para terra de proteção** ⏚	 1000 Vac / 1 minuto 1000 Vac / 1 minuto

Tabela 18: Característica das Entradas Analógicas

	Entrada de tensão		
Faixas de entrada	Faixa	Escala de Engenharia	Resolução
	0 a 10 Vdc	0 a 30.000	2,52 mV
Precisão	±0,3 % do fundo de escala @ 25 °C ± 0,010 % do fundo de escala / °C		
Sobre escala	3 % do fundo de escala		
Tensão máxima de entrada	14 Vdc		
Impedância de entrada	21 kΩ		
Parâmetros configuráveis	Filtros		
Constante de tempo do filtro passa baixa	100 ms, 1 s, 10 s ou desabilitado		

Tabela 19: Característica das Entradas Analógicas de Tensão

	Entrada Modo Corrente		
Faixas de entrada	Faixa	Escala de engenharia	Resolução
	0 a 20 mA	0 a 30.000	5,03 μA
	4 a 20 mA	0 a 30.000	5,03 μA
Precisão	±0.3 % do fundo de escala @ 25 °C ± 0.015 % do fundo de escala / °C		
Sobre escala	3 % do fundo de escala		
Corrente máxima de entrada	30 mA		
Impedância de carga	119 Ω		
Parâmetros configuráveis	Filtros, valor de loop aberto		
Constante de tempo do filtro passa baixa	100 ms, 1 s, 10 s ou desabilitado		

Tabela 20: Característica das Entradas Analógicas de Corrente

2.2.16. Saídas Analógicas

	Saídas analógicas
Número de saídas	2
Tipo de saída	Saída de tensão ou corrente, configurável individualmente
Formato dos dados	16 bits em complemento de dois, justificado à esquerda
Resolução do conversor	Monotonicidade garantida de 12 bits, sem códigos perdidos
Tempo de conversão	400 μs (todas as saídas habilitadas)
Indicação status do canal	Sim (LED)
Proteções	Sim, proteção contra surtos de tensão e inversão de polaridade
Isolações	
Canal para lógica	1000 Vac / 1 minuto
Canal para terra de proteção ⏚	1000 Vac / 1 minuto

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	2,52 mV
Precisão	±0,3 % do fundo de escala @ 25 °C ± 0,010 % do fundo de escala / °C		
Tempo de estabilização	4 ms		
Sobre escala	3% do fundo de escala		
Impedância de carga	$>$ 1 kΩ		
Parâmetros configuráveis	Tipo de sinal (tensão ou corrente)		

Tabela 22: Características das Saídas Analógicas Modo Tensão

	Saída modo corrente		
Faixas de saída	Faixa	Escala de Engenharia	Resolução
	0 a 20 mA	0 a 30.000	5,03 μA
	4 a 20 mA	0 a 30.000	5,03 μA
Precisão	±0,3 % do fundo de escala @ 25 °C ± 0,015 % do fundo de escala / °C		
Tempo de estabilização	4 ms		
Sobre escala	3% do fundo de escala		
Impedância de carga	$<$ 600 Ω		
Parâmetros configuráveis	Tipo de sinal (tensão ou corrente)		

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 do Sistema Nexto, é necessário verificar a versão do Mastertool. A tabela a seguir mostra a versão mínima necessária (onde os controladores foram introduzidos) e a respectiva versão de firmware naquele momento:

Modelo do controlador	Mastertool X	Versão de Firmware
XF300-B, XF300, XF315, XF315 e XF325-W	4.0.0	1.14.88.2

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 XF 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.

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

2.4.3. Tempo Para a Execução de Instruções

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

Instrução	Linguagem	Tipo da Variável	Tempo (μs)
1000 Contatos	LD	BOOL	2,1
1000 Divisões	LD, ST	INT	9,2
		REAL	17,0
1000 Multiplicações	LD, ST	INT	6,4
		REAL	8,2
1000 Somas	LD, ST	INT	4,4
		REAL	8,2

Tabela 25: Tempos de Instrução

2.4.4. Tempos de Inicialização

O tempo de inicialização dos controladores Nexto XF é de aproximadamente 30 s.

2.5. Dimensões Físicas

Dimensões em mm.

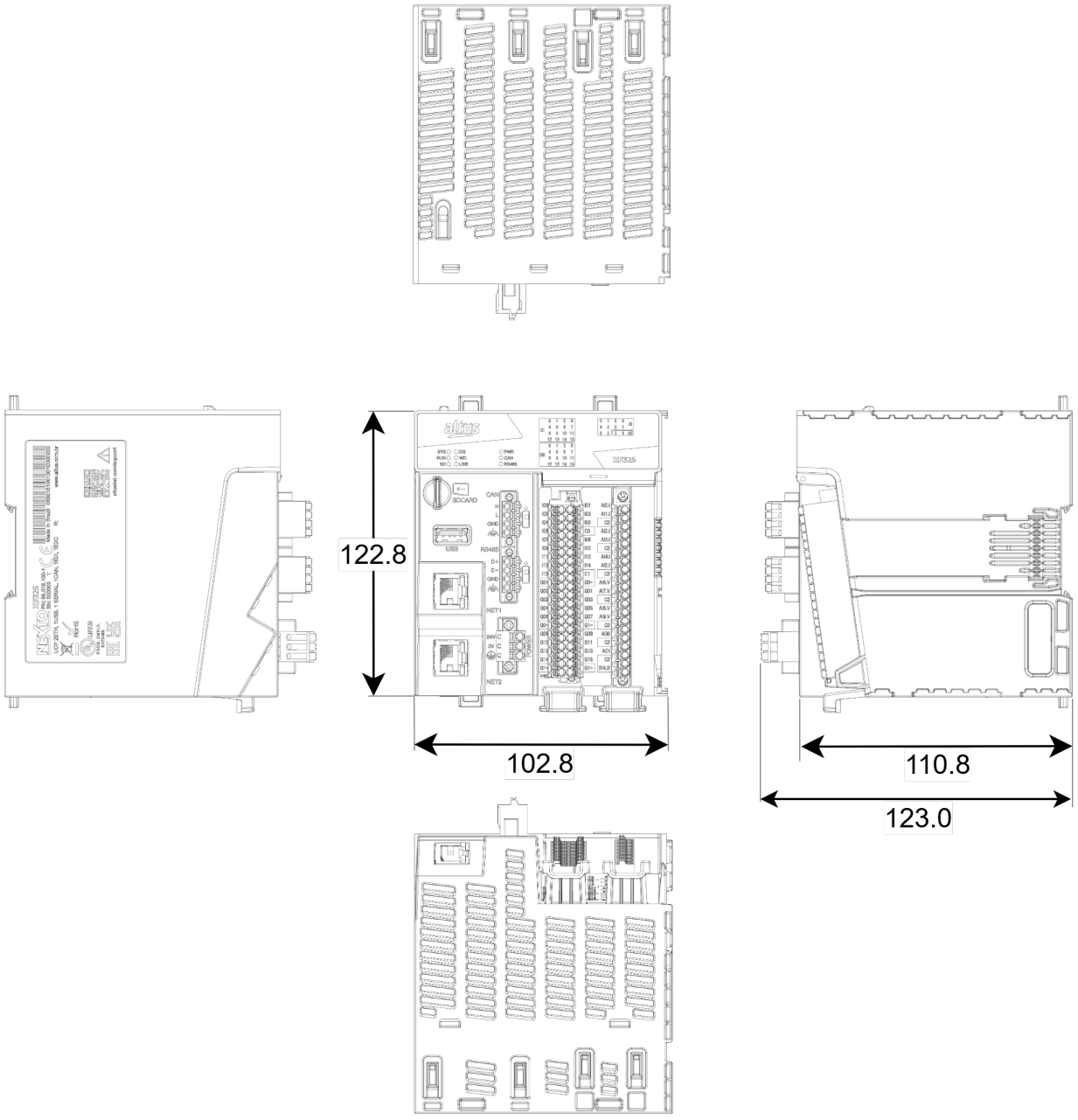


Figura 3: Dimensões Físicas do XF3xx

2.6. Dados de Compra

2.6.1. Itens Integrantes

A embalagem do produto possui os seguintes itens:

- Módulo CP com E/S Integrados XF3xx
- Conector de 40 pinos
- Conector de 20 pinos (exceto para XF300 e XF300-B)
- Conector de 3 pinos
- 2 x Conector de 4 pinos
- Terminador do barramento
- Tampa do cabo de E/S integrado à CPU

2.6.2. Código de Produto

Os seguintes códigos devem ser usados para compra do produto:

Código	Descrição
XF300-B	CLP compacto de alta velocidade com 16 entradas digitais, 16 saídas digitais de transistor, 2 portas Ethernet, 1 porta RS-485, 1 porta CAN, host USB e cartão microSD. Suporte apenas para protocolos básicos.
XF300	CLP compacto de alta velocidade com 16 entradas digitais, 16 saídas digitais de transistor, 2 portas Ethernet, 1 porta RS-485, 1 porta CAN, host USB e cartão microSD.
XF315	CLP compacto de alta velocidade com 16 entradas digitais, 16 saídas digitais a transistor, 6 entradas analógicas de corrente, 4 entradas analógicas de tensão, 2 portas Ethernet, 1 porta RS-485, 1 porta CAN, host USB e cartão microSD.
XF325	CLP compacto de alta velocidade com 16 entradas digitais, 16 saídas digitais a transistor, 6 entradas analógicas de corrente, 4 entradas analógicas de tensão, 2 saídas analógicas (V/I), 2 portas Ethernet, 1 porta RS-485, 1 porta CAN, host USB e cartão microSD.
XF325-W	CLP compacto de alta velocidade com 16 entradas digitais, 16 saídas digitais a transistor, 6 entradas analógicas de corrente, 4 entradas analógicas de tensão, 2 saídas analógicas (V/I), 2 portas Ethernet, 1 porta RS-485, 1 porta CAN, host USB, cartão microSD e suporte a página web do usuário.

Tabela 26: Modelos de Controladores Nexto XF

2.7. Produtos Relacionados

Os seguintes produtos devem ser adquiridos separadamente quando necessário:

Código	Descrição
MT9000	Mastertool X
AL-2600	Derivador e terminador de rede RS-485
AL-2306	Cabo RS-485 p/ rede MODBUS ou CAN
NX9101	Cartão de 32 GB microSD com adaptador para miniSD e SD
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
XF101	Módulo de 16 entradas digitais
XF201	Módulo de 16 saídas digitais a transistor
XF600	Módulo de 8 entradas analógica de Tensão/Corrente
XF610	Módulo de 8 entradas analógicas para termopar
XF620	Módulo de 6 entradas analógicas RTD (resistência)
XF700	Módulo de 4 saídas analógicas de Tensão/Corrente
XF900	Terminação de barramento Nexto XF
XF901	Tampa do I/O integrado da CPU
XF902	Tampa do módulo de I/O

Tabela 27: Produtos Relacionados

Notas:

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.

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

AMJG0808: Cabo para a programação das UCPs.

3. Instalação

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

CUIDADO

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

ATENÇÃO

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

- O produto deverá ser instalado em um gabinete metálico.

ATENÇÃO

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

- Esses dispositivos são Open-type e devem ser instalados em um invólucro adequado ao ambiente, acessível somente com o uso de uma ferramenta ou chave.
- Este equipamento é adequado para uso apenas em locais de Classe I, Divisão 2, Grupos A, B, C e D ou locais não perigosos.

PERIGO

- Atenção – Risco de explosão. Não desconecte enquanto o circuito estiver energizado ou a menos que a área esteja livre de concentrações inflamáveis.
- Atenção – Risco de explosão. Não remova ou substitua os módulos plugáveis a menos que a energia tenha sido desligada ou a área esteja livre de concentrações inflamáveis.

DANGER

- Avertissement – Risque d'explosion. Ne pas débrancher pendant que le circuit est sous tension ou à moins que l'emplacement ne soit exempt de concentrations inflammables.
- Avertissement – Risque d'explosion. Ne pas retirer ni remplacer les modules enfichables à moins que l'alimentation n'ait été coupée ou que l'emplacement ne soit exempt de concentrations inflammables.

3.1. Instalação Mecânica

Os controladores Nexto XF foram projetados para serem instalados em um trilho DIN padrão. Além disso, o usuário deverá providenciar 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.

3.1.1. Instalando o Controlador

Para instalar o controlador no trilho DIN, primeiro mova as quatro travas para a posição aberta, conforme indicado na figura abaixo:

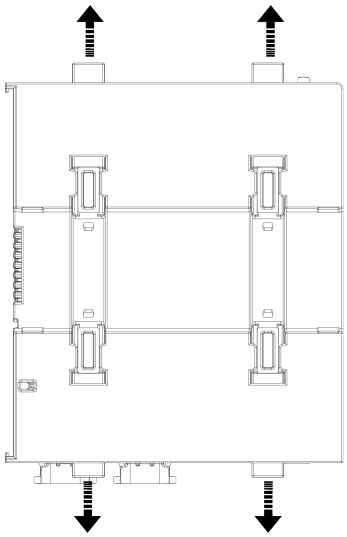


Figura 4: Mover as quatro travas para a posição aberta.

Em seguida, coloque o controlador no encaixe do trilho DIN e mova as quatro travas para a posição fechada para fixar o controlador no trilho DIN, conforme mostrado na figura abaixo:

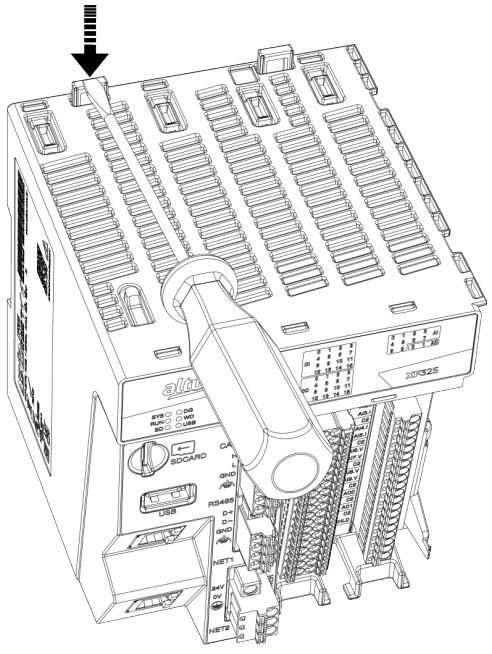


Figura 5: Travando o controlador no trilho DIN

CUIDADO

Instale sempre o produto fixando-o perpendicularmente no trilho DIN. Não o insira e deslize a partir da extremidade do trilho DIN, pois isso pode entortar ou danificar a mola de aterramento traseira.

3.1.2. Removendo o Controlador

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

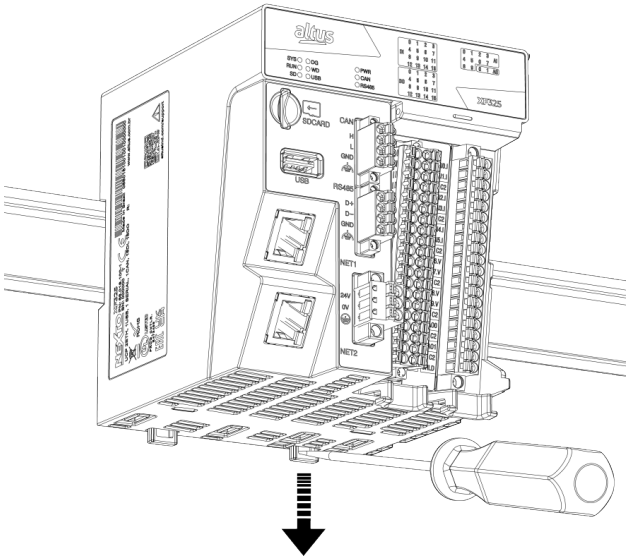


Figura 6: Destravando o controlador do trilho DIN

3.1.3. Instalando e removendo o conector para o I/O integrado

Para instalar o conector no soquete, alinhe-o com o soquete, garantindo a orientação correta, e pressione-o firmemente até que esteja totalmente encaixado. Em seguida, aperte o parafuso dos flanges em ambos os lados do conector para fixá-lo no lugar e evitar desconexões acidentais.

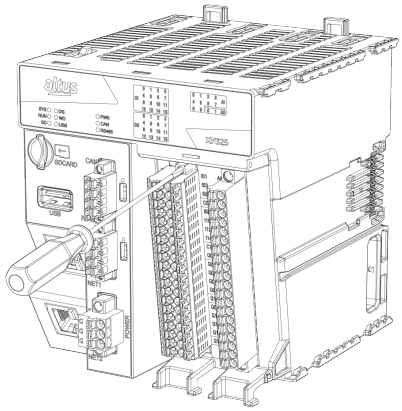


Figura 7: Instalando o conector da E/S integrada

Para remover o conector, afrouxe o parafuso dos flanges em ambos os lados e puxe o conector para fora do soquete. Para remover o conector de 2x20, você deve posicionar duas chaves de fenda finas nas laterais do conector e cuidadosamente fazer uma alavanca para removê-lo.

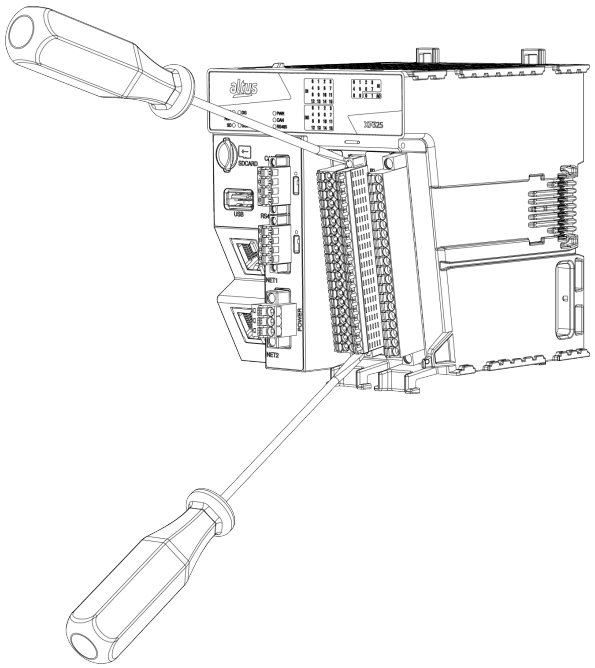


Figura 8: Removendo o conector de 2x20

Para conectar os fios ao conector, insira-os diretamente na abertura do terminal correspondente até ouvir um clique. Não são necessárias ferramentas. Para fios flexíveis, recomenda-se o uso de um terminal de compressão antes da inserção. Para soltar um fio, pressione o botão de liberação no terminal correspondente e puxe o fio para fora.

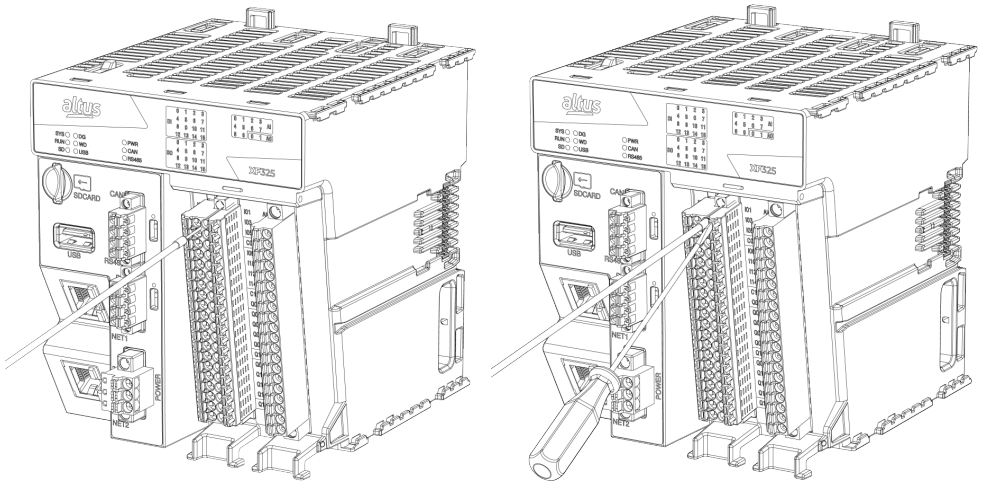


Figura 9: Instalação dos cabos no conector.

3.1.4. Instalando e removendo a terminação do I/O

Para instalar a terminação de E/S, posicione-a no ponto de inserção e empurre-a horizontalmente em direção ao trilho DIN até que o movimento pare. Para remover a terminação de E/S, basta inverter o movimento, deslizando-a horizontalmente até o ponto de inserção, até que as abas na parte plástica estejam desobstruídas.

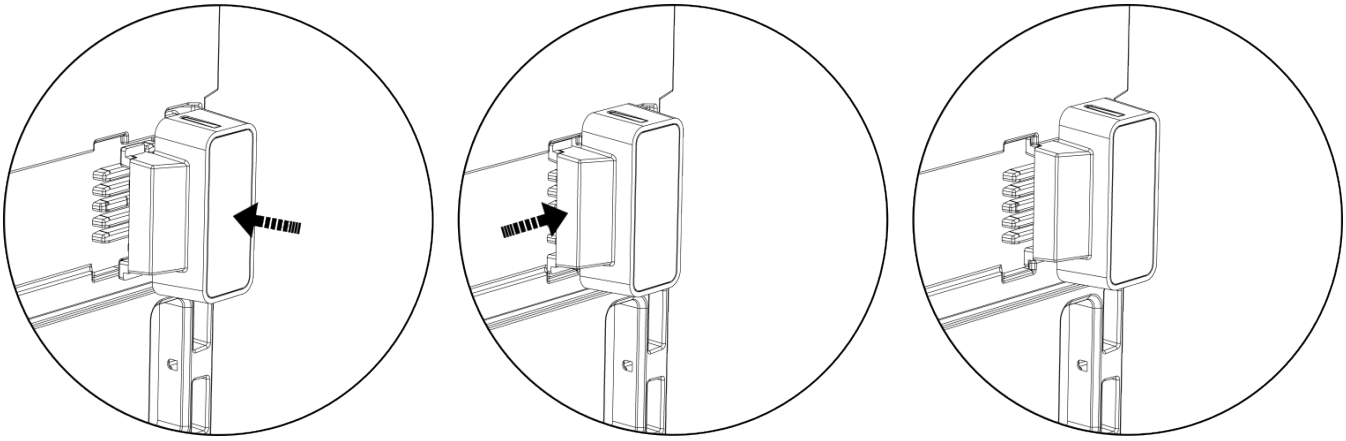


Figura 10: Instalando a terminação da E/S

3.1.5. Instalando e removendo a tampa do I/O integrado

Para instalar a tampa da E/S integrada, coloque-a na horizontal e encaixe-a na cavidade esquerda, conforme mostrado na figura abaixo. Em seguida, flexione a aba da tampa para encaixar o pino na cavidade direita.

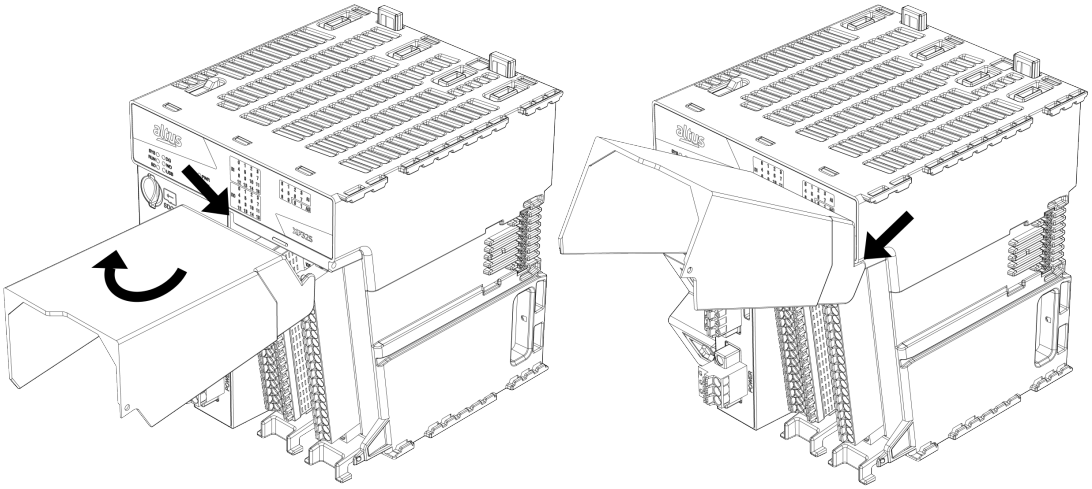


Figura 11: Instalando a tampa da E/S integrada

Para remover a tampa de E/S integrada, coloque a tampa na horizontal e flexione a aba da tampa para remover o pino direito da cavidade. Em seguida, remova o pino esquerdo da cavidade esquerda, conforme mostrado na figura abaixo.

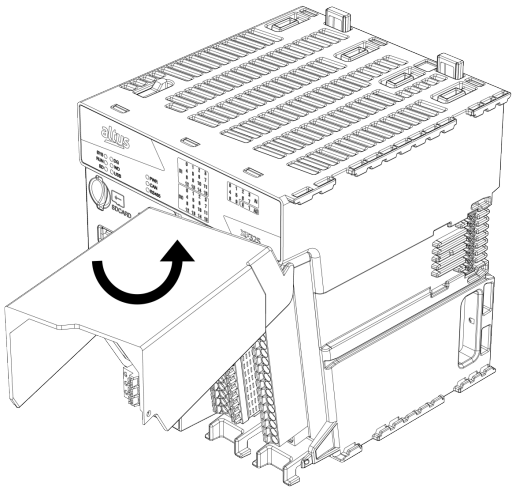


Figura 12: Removendo a tampa do I/O integrado

3.2. Espaçamento entre a CPU e outros equipamentos no painel

A instalação correta do CLP requer espaço livre suficiente em todos os lados. Esse espaço livre tem duas funções: permite o manuseio adequado do dispositivo e garante fluxo de ar suficiente para o resfriamento por convecção, mantendo o dispositivo dentro de sua faixa de temperatura operacional. Os espaços livres mínimos necessários são apresentados a seguir:

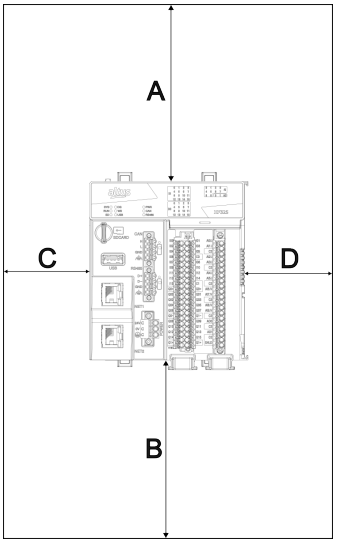


Figura 13: Espaçamento em volta do controlador

Dimensão A	Dimensão B	Dimensão C	Dimensão D
10 cm	10 cm	4 cm	4 cm

Tabela 28: Dimensão do espaço livre ao redor do controlador

3.3. Montagem Horizontal/Vertical

O Sistema Nexto suporta apenas montagem horizontal. A instalação vertical no rack não é permitida.

3.4. Instalação Elétrica

PERIGO

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

CUIDADO

Para as fontes de 24 Vdc das entradas de alimentação e sinais de E/S utilize somente fontes isoladas que forneçam Tensão Extra Baixa Segura (SELV/PELV) com LPS ou Classe 2 ou Limitação de Energia, de acordo com as normas IEC 61131-2, IEC 61010-2-201 ou equivalente.

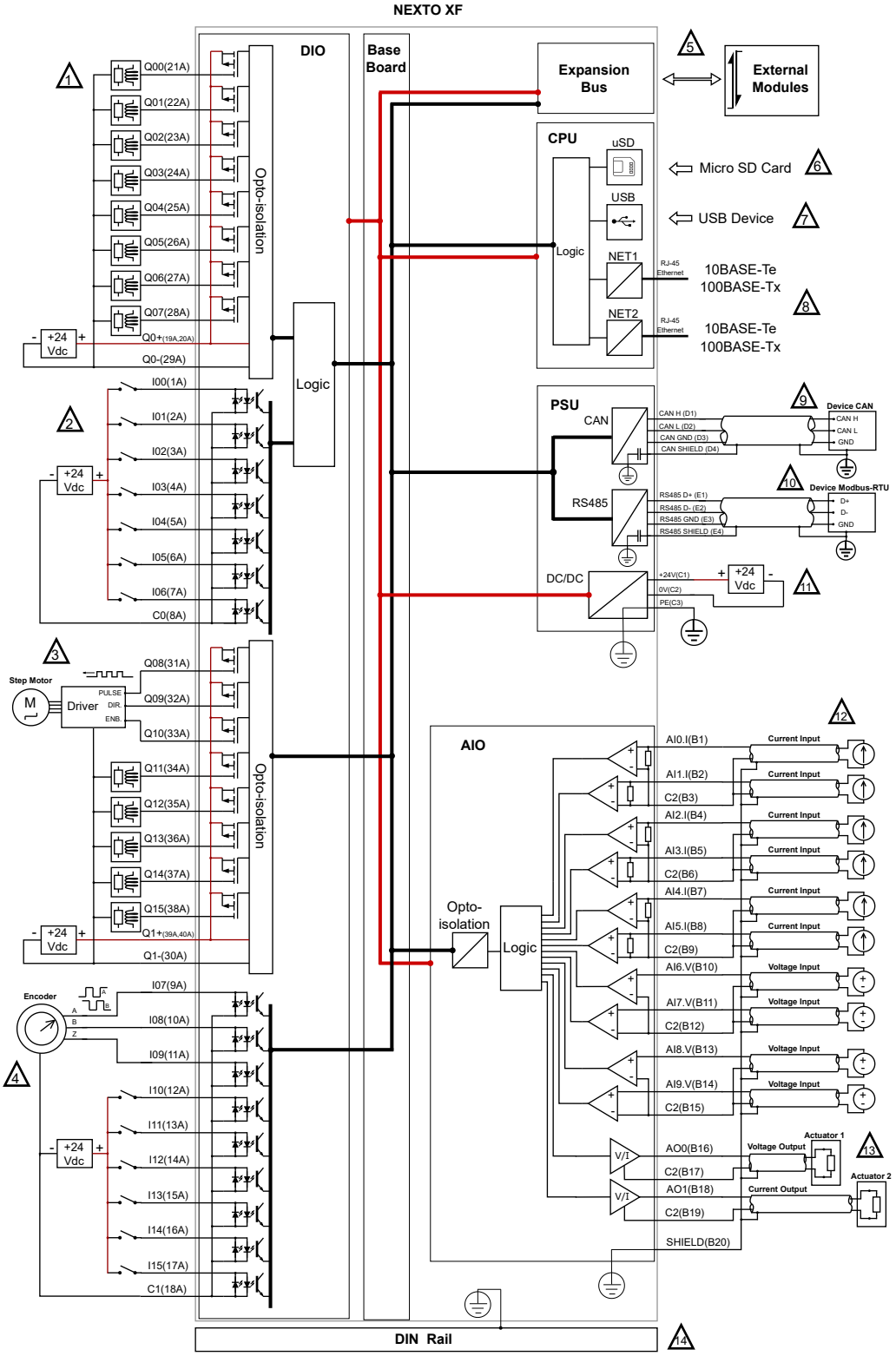


Figura 14: Diagrama de Instalação Elétrica do XF3xx

Notas do Diagrama:

1. Conexão típica das saídas digitais isoladas de Q00 a Q07 (tipo source). Alimentação externa independente de 24 Vdc ligada nos bornes Q0+ e Q0-.

2. Conexão típica das entradas digitais isoladas de I00 a I06 (tipo sink). Alimentação externa independente de 24 Vdc, ligada ao borne comum C0 e contato das chaves.

3. Conexão típica das saídas digitais rápidas isoladas de Q08 a Q15, para controle de Motor de Passo e acionamento de cargas discretas. Alimentação externa independente de 24 Vdc, ligada aos bornes Q1+ e Q1-.

4. Conexão típica das entradas digitais rápidas isoladas de I07 a I15, com dispositivo de Encoder de 3 sinais (A, B e Z) e chaves discretas. Alimentação externa independente de 24 Vdc, ligado ao borne comum C1 e contato das chaves.

5. Conexão do barramento dos Módulos Expansão de E/S ou Tampa com Terminação.

6. Entrada para cartão microSD.

7. Verifique a tabela de características técnicas da porta USB para obter a lista de dispositivos suportados.

8. Utilize os cabos Ethernet especificados na seção Produtos Relacionados. Blindagem do conector conectado diretamente ao aterramento interno.

9. Conexão típica da interface serial CAN. Blindagem do cabo acoplado internamente ao aterramento via capacitor, a blindagem do cabo deve ser conectado ao aterramento externamente em apenas uma das extremidades.

10. Conexão típica da interface serial RS-485. Blindagem do cabo acoplado internamente ao aterramento via capacitor, a blindagem do cabo deve ser conectado ao aterramento externamente em apenas uma das extremidades.

11. Conexão da fonte de alimentação do sistema, nominalmente 24 Vcc. Ponto de aterramento do sistema se o trilho DIN não estiver aterrado.

12. Conexão típica para entradas analógicas de tensão e corrente. O aterramento da blindagem dos cabos ocorre em um único ponto no conector. A conexão da blindagem dos cabos deve ocorrer em apenas uma das extremidades. Essa nota não é aplicável para XF300 e XF300-B.

13. Conexão típica das saídas analógicas para tensão e corrente configurável. O aterramento da blindagem dos cabos ocorre em um único ponto no conector. A conexão da blindagem dos cabos deve ocorrer em apenas uma das extremidades. Essa nota não é aplicável para XF315, XF300 e XF300-B.

14. O aterramento é realizado através do contato direto do produto com o trilho DIN, que deve ser metálico e conectado a um aterramento externo adequado, veja a nota 11.

Terminal terra de proteção.

3.5. Conexão à rede Ethernet

A interface de comunicação ETH, identificada como NET 1 e NET 2 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 XF. Este cabo deve ter uma de suas extremidades conectada à interface que será utilizada e a outra ao HUB, switch, microcomputador ou outro ponto de acesso da rede Ethernet.

3.5.1. Endereço IP

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

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

Tabela 29: 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 subrede 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 das Interfaces Ethernet.

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

	NET 2
Endereço IP	192.168.16.1
Máscara de subrede	255.255.255.0
Endereço do Gateway	0.0.0.0

Tabela 30: Configuração de Parâmetros Padrão para a Interface Ethernet NET 2

3.5.2. ARP Gratuito

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

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

3.5.3. Instalação do Cabo de Rede

As portas Ethernet, identificadas no painel por NET, possuem pinagem padrão, sendo a mesma utilizada, por exemplo, em computadores pessoais. O tipo de conector, tipo de cabo, nível físico, entre outros detalhes para interligar o controlador ao dispositivo de acesso à rede Ethernet, são definidos na seção Configuração das Interfaces Ethernet.

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

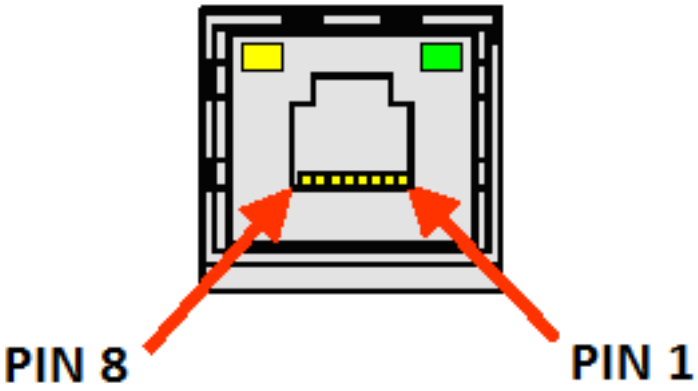


Figura 15: Conector Fêmea RJ45

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

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

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

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

3.6. Instalação de rede Serial RS-485 e CAN

Tanto a interface RS-485 quanto a CAN utilizam dois sinais de comunicação e um terra. O cabo recomendado é o AL-2306, utilizando um dos dois pares e a blindagem. Se o controlador for colocado em uma das extremidades da rede, a terminação externa deverá ser habilitada manualmente através da chave DIP localizada no painel frontal da CPU, conforme ilustrado abaixo e no diagrama Instalação Elétrica. A posição da chave DIP para cima significa terminação habilitada.

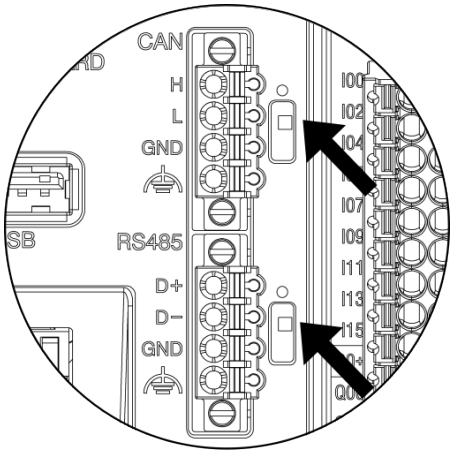


Figura 16: Instalando a terminação de rede RS-485 e CAN

3.7. Instalação do Cartão de Memória

Nesta seção está descrita a maneira como o cartão de memória deve ser inserido na UCP do Sistema Nexto, sendo que, para maiores informações sobre a utilização do mesmo, consultar a seção Cartão de Memória.

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

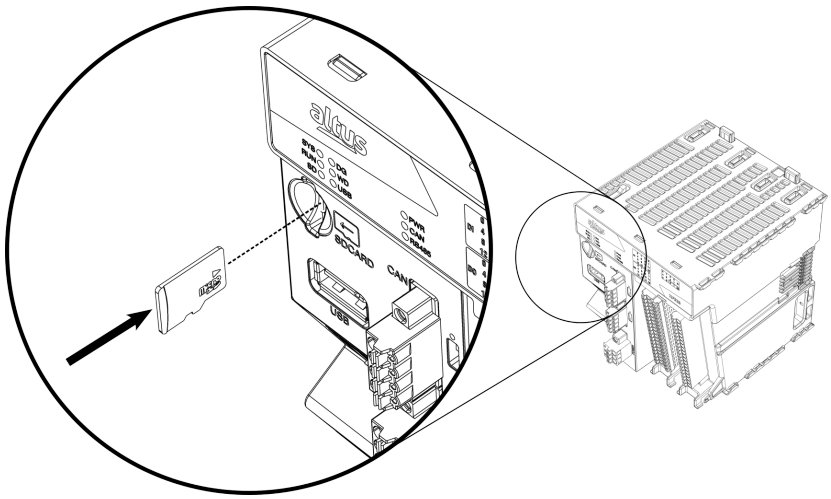


Figura 17: Inserindo o Cartão de Memória

Atenção:

A serigrafia do painel frontal traz a identificação SDCARD e o desenho de um cartão de tamanho padrão, mas a interface aceita exclusivamente cartão microSD. Não insira cartão de tamanho padrão nem adaptador no slot: não há encaixe e a tentativa pode danificar o conector.

Quando o cartão estiver instalado corretamente, o LED correspondente acenderá. Para remover o cartão com segurança, o usuário deve acessar a página da web e clicar no botão “Desmontar”. Logo em seguida, o LED correspondente se apagará e o status será atualizado na página de informações de status. O cartão estará então pronto para ser removido. Para isso, pressione o cartão contra a CPU até ouvir um clique, solte-o e retire-o do compartimento. Nesse momento, o cartão estará solto.

4. Programação Inicial

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

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

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

Para mais informações sobre Programação, consultar Manual de Utilização Mastertool, Manual de Programação IEC 61131 ou o padrão IEC 61131-3.

4.1. Organização e Acesso à Memória

Diferentemente de outros dispositivos do Sistema Nexto (baseados em UCP big-endian), este modelo de UCP é baseada em uma UCP ARM, que utiliza a tradicional organização de memória little-endian (a mesma encontrada nos processadores x86 e Intel). Nesse tipo de organização de memória, o byte menos significativo é armazenado primeiro e sempre será o menor endereço (por exemplo, %QB0 sempre será menos significativo que %QB1, conforme mostrado na tabela abaixo, onde, para CPUNEXTO, a letra O é o byte 0 e a letra C é o byte 7).

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

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

Tabela 32: Exemplo Organização e Acesso à Memória

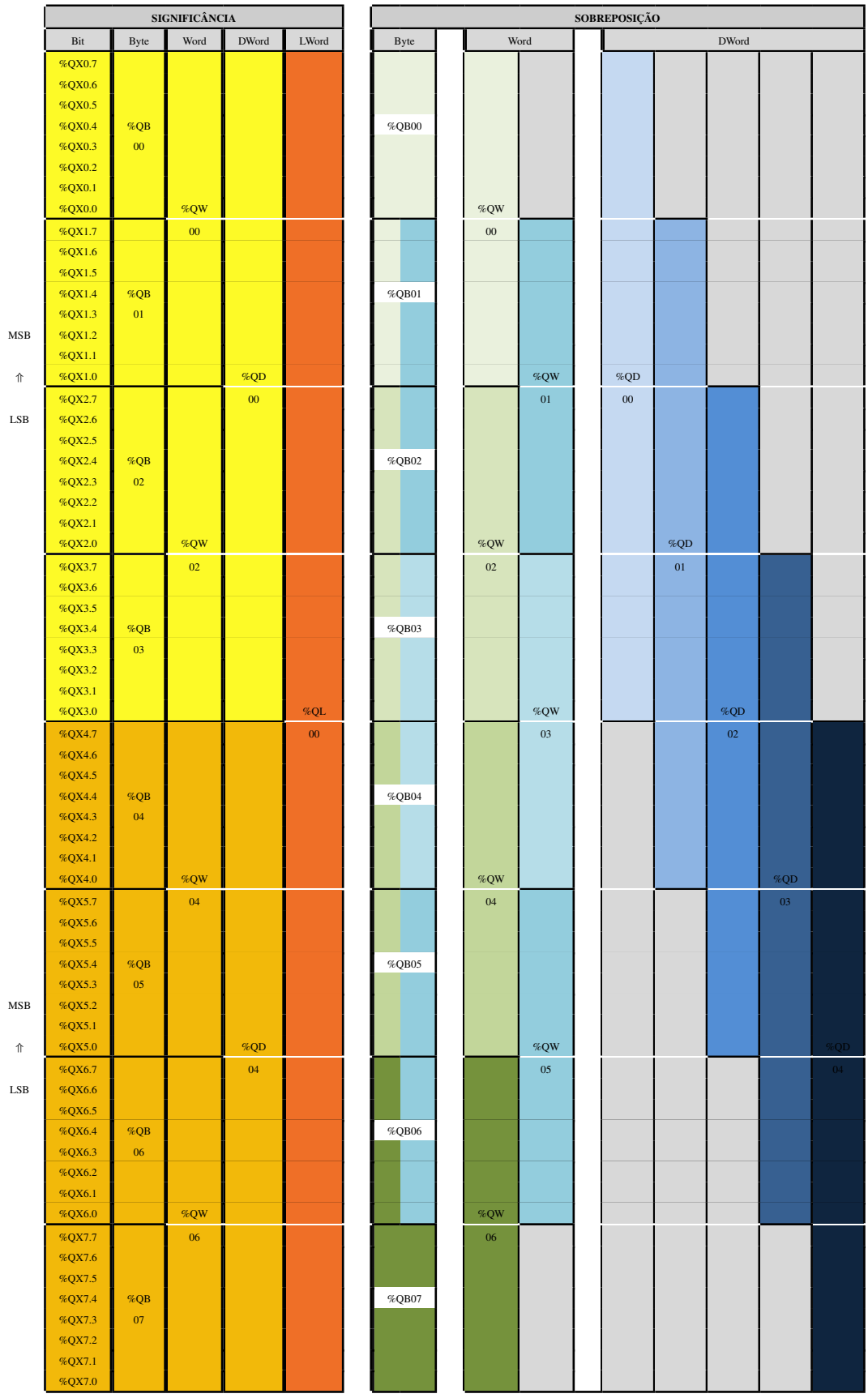


Tabela 33: Organização e Acesso à Memória

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

4.2. Perfis de Projeto

Um perfil de projeto consiste em um modelo de aplicação combinado com um grupo de regras de verificação que orienta o desenvolvimento da aplicação, reduzindo a complexidade da programação.

O perfil do projeto é selecionado no assistente de criação do projeto. Cada perfil de projeto define um modelo de nomes padrão para as tarefas e programas, que são pré-criados de acordo com o perfil do projeto selecionado. Além disso, durante a compilação do projeto (gerar código), o Mastertool verifica todas as regras definidas pelo perfil selecionado.

As seções a seguir detalham as características de cada perfil. É importante notar que o Mastertool permite a mudança de perfil de um projeto existente (consulte a seção de atualização do projeto no manual do Mastertool), mas cabe ao desenvolvedor fazer os ajustes necessários para que o projeto se torne compatível com as regras do novo perfil selecionado.

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

4.2.1. Perfil Fast

No perfil Fast, por padrão, a aplicação possui uma tarefa de usuário do tipo Cíclica chamada MainTask. Essa tarefa é responsável por implementar uma única Unidade de Otimização de Programa (POU) do tipo Programa, chamada MainPrg. Esse programa pode chamar outras unidades de programação dos tipos Programa, Função ou Bloco Funcional, mas qualquer código de usuário será executado exclusivamente pela tarefa MainTask.

Esse perfil se caracteriza por permitir intervalos mais curtos na MainTask, possibilitando uma execução mais rápida do código de usuário. Além disso, esse perfil pode incluir uma tarefa de interrupção, com prioridade maior que a da MainTask, podendo, portanto, interromper sua execução a qualquer momento.

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

Tabela 34: Tarefas no Perfil Fast

4.2.2. Número Máximo de Tarefas

O usuário pode criar vários tipos de tarefas, desde que os números estabelecidos para cada tipo e o valor total não sejam ultrapassados. Na tabela de Características Gerais é informado o número máximo de tarefas suportado por este controlador. O Mastertool exibe os limites detalhados dos tipos de tarefas na guia “Propriedades” da Configuração de Tarefas.

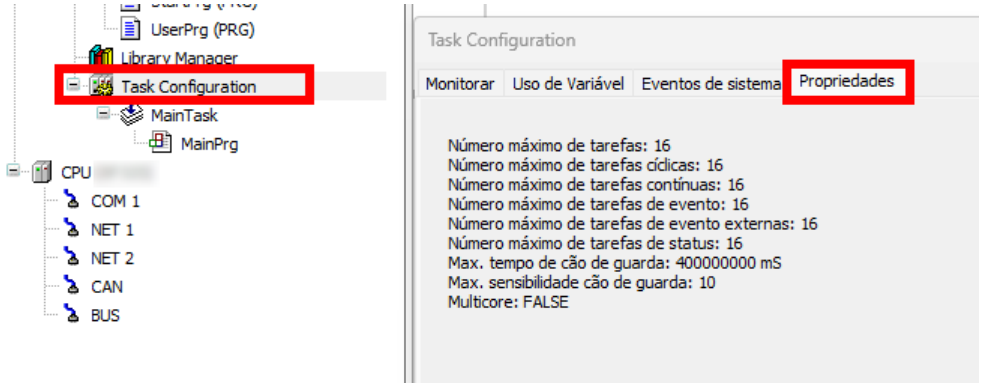


Figura 18: Exemplo do número máximo de tarefas informado pelo Mastertool

4.3. Configuração da UCP

A configuração da UCP do Nexto XF está localizada na árvore de dispositivos, como mostrado na figura abaixo, e pode ser acessada clicando duas vezes no objeto correspondente. Mais informações podem ser encontradas em Configuração da UCP.

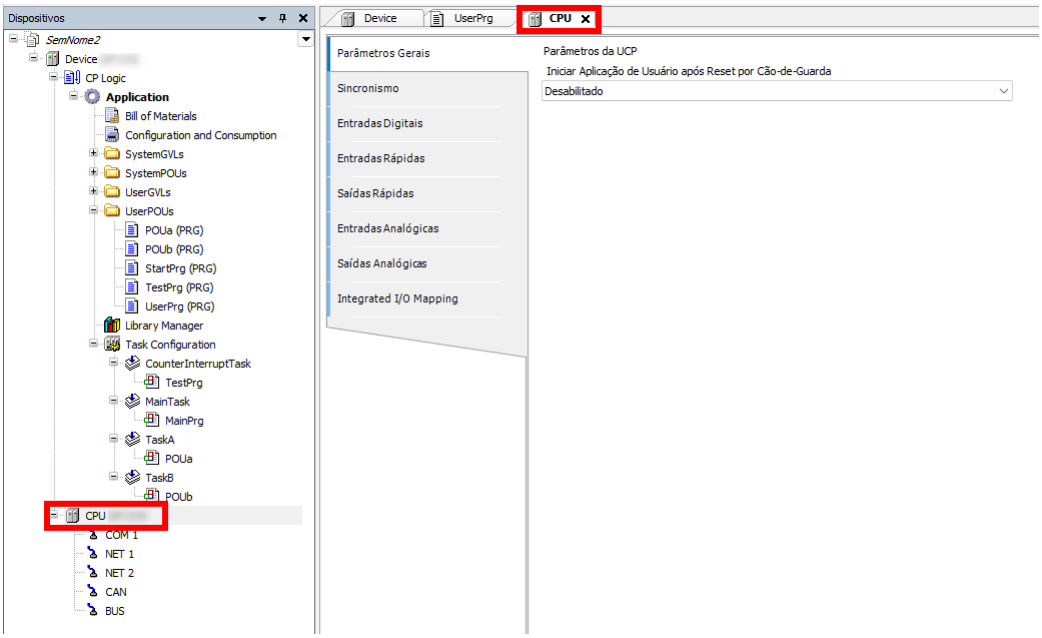


Figura 19: Configuração da UCP

Além disso, clicando duas vezes sobre o ícone NET 1 do controlador é possível realizar a configuração da interface Ethernet através da qual será realizada a comunicação entre o controlador e o Mastertool.

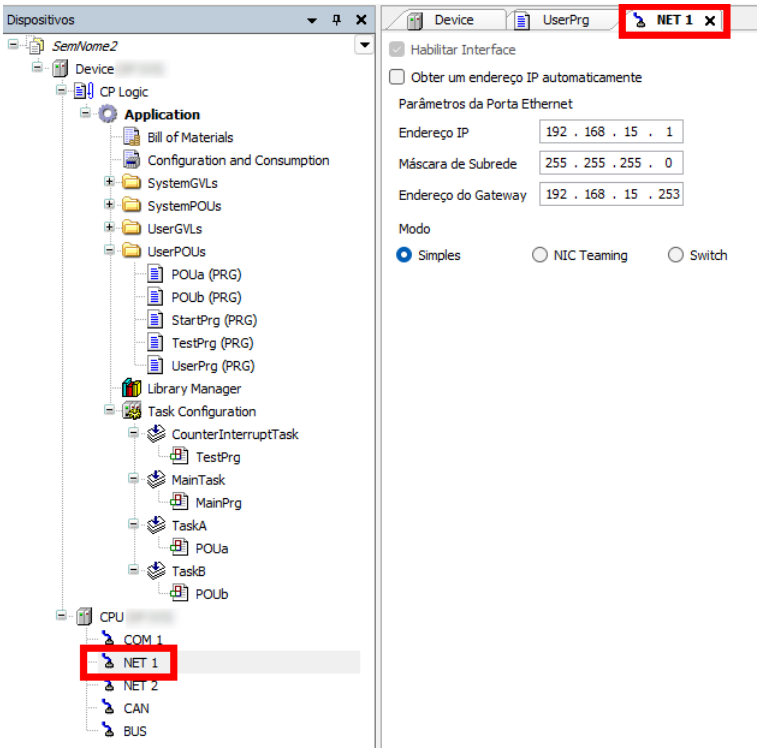


Figura 20: Configurando a Porta de Comunicação

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

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

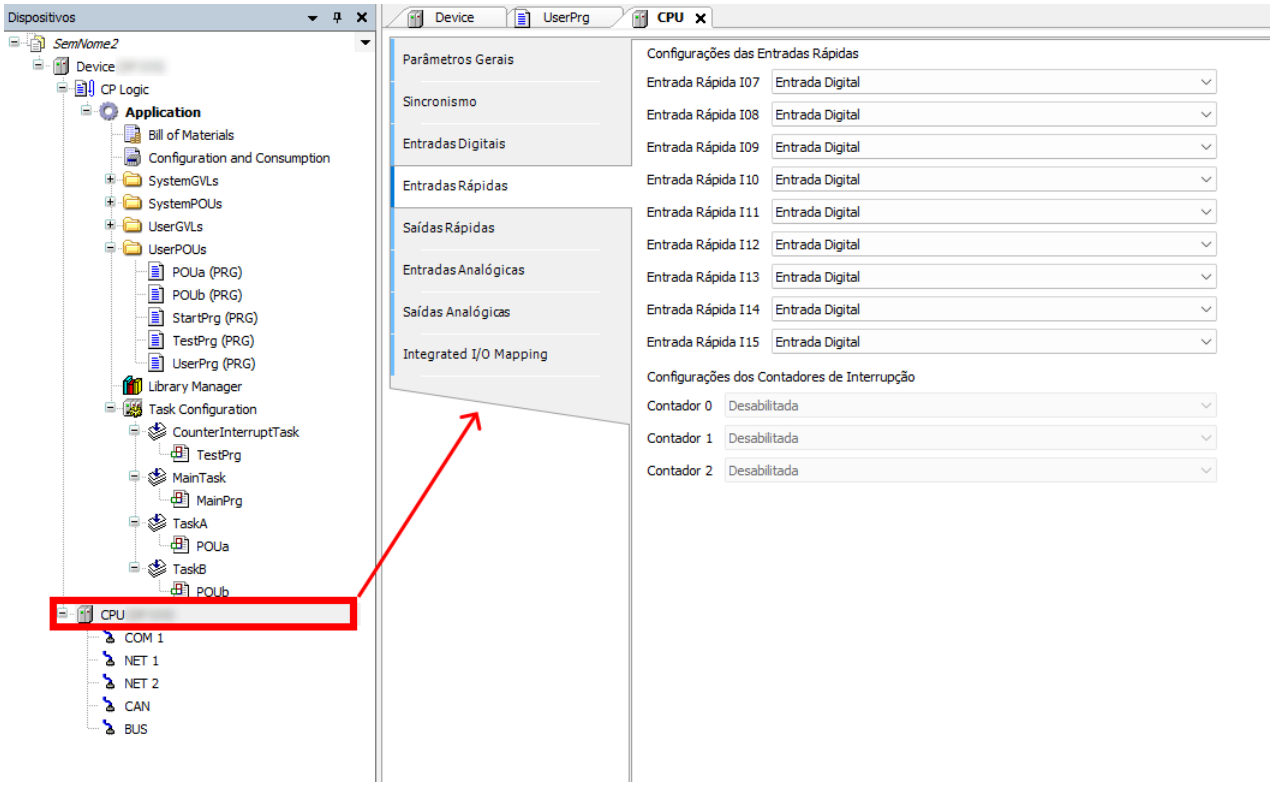


Figura 21: Configurando a E/S integrada

4.4. Bibliotecas

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

4.5. Inserindo uma instância de Protocolo

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

4.5.1. MODBUS RTU

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

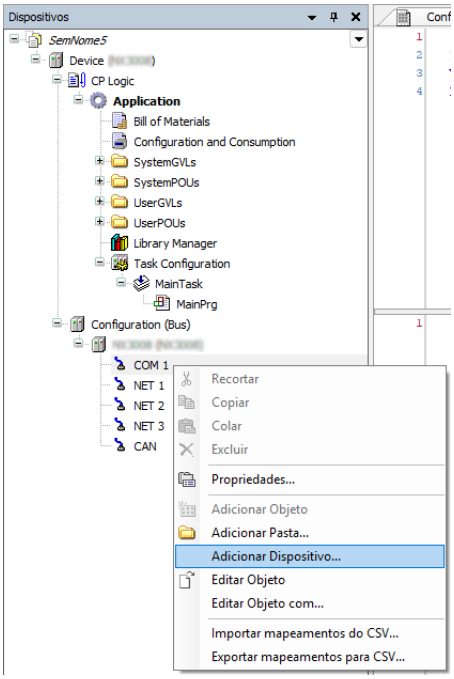


Figura 22: Adicionando a Instância

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

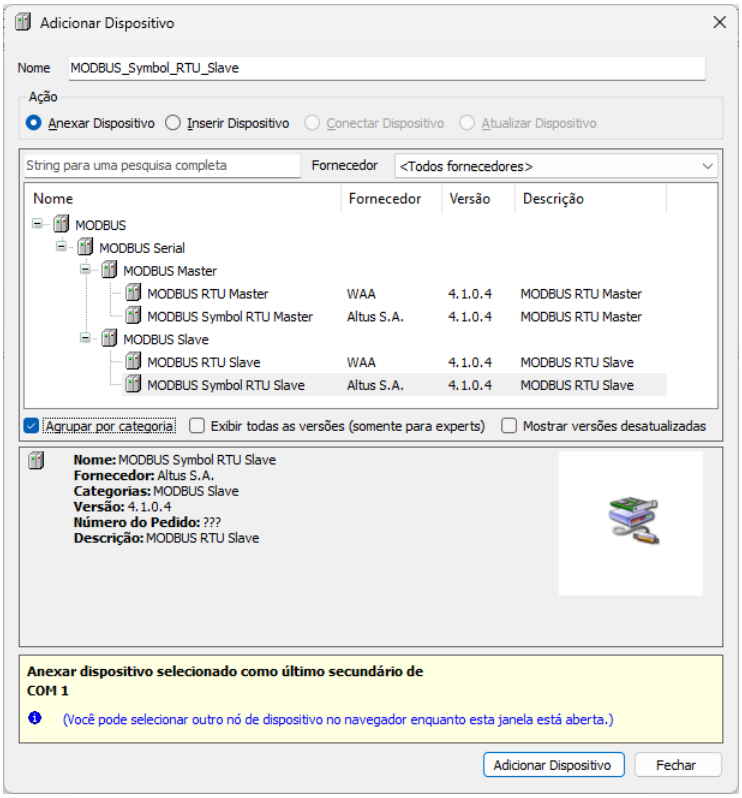


Figura 23: Selecionando o Protocolo

4.5.2. MODBUS Ethernet

O primeiro passo para configurar o MODBUS Ethernet (Servidor neste exemplo) é incluir a instância na rede desejada. Clique com o botão direito do mouse na rede e selecione Adicionar Dispositivo..., como mostrado na Figura 24:

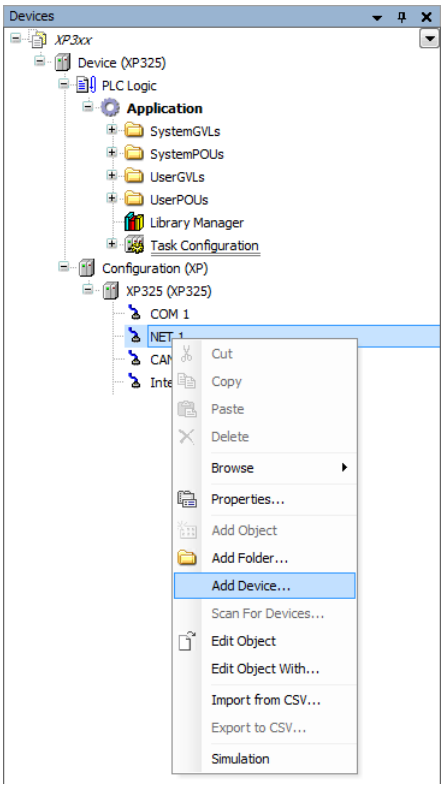


Figura 24: Adicionando a Instância

Em seguida, a lista de protocolos aparecerá na tela. Basta selecionar MODBUS Symbol Server, conforme descrito na figura abaixo:

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.

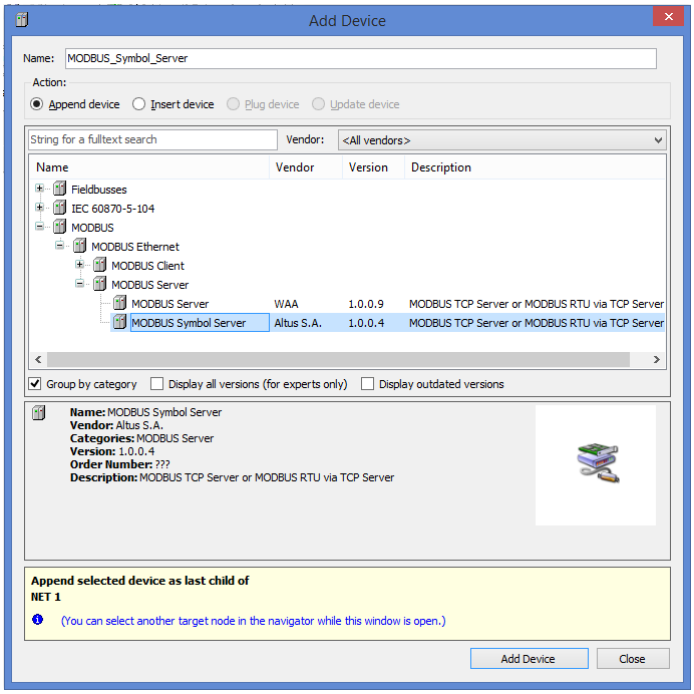


Figura 25: Selecionando o protocolo

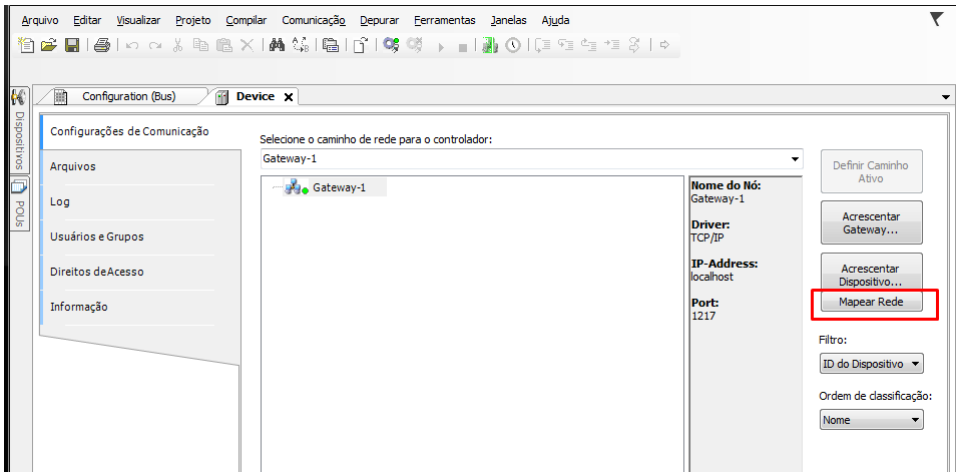


Figura 26: Localizando a UCP

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

Em seguida, basta selecionar 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.

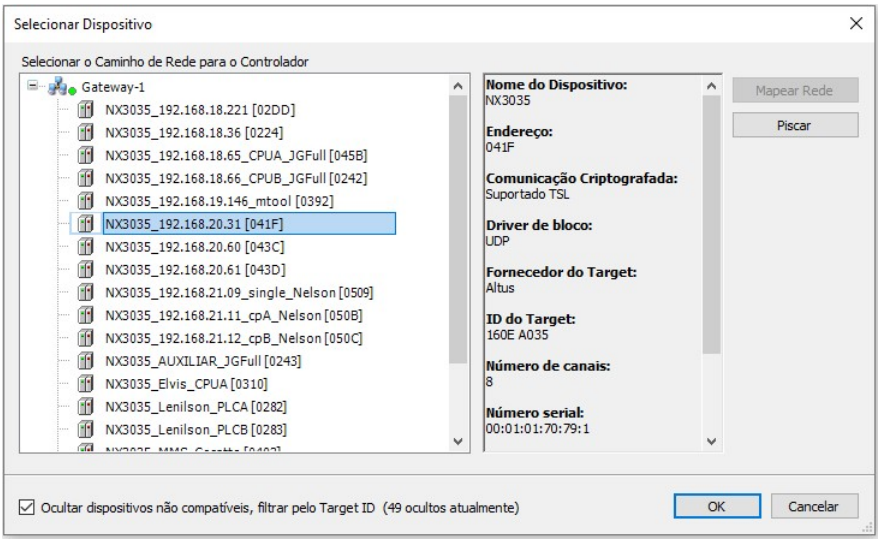


Figura 27: Selecionando a UCP

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

Caso a configuração Ethernet do controlador ao qual 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.

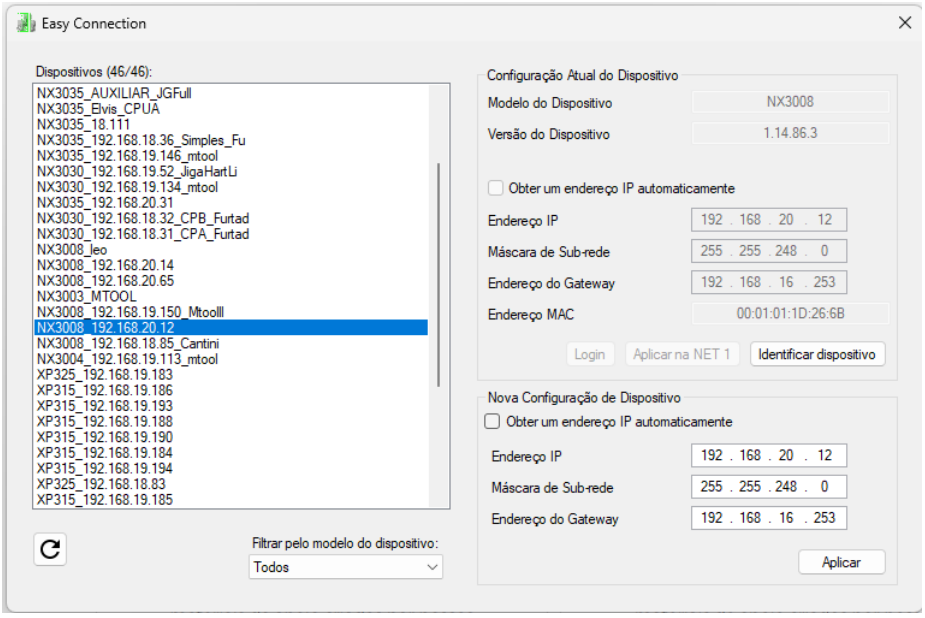


Figura 28: Easy Connection

Este comando realiza uma comunicação em nível MAC com a interface NET 1 do dispositivo, permitindo alterar permanentemente a configuração da interface Ethernet 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.

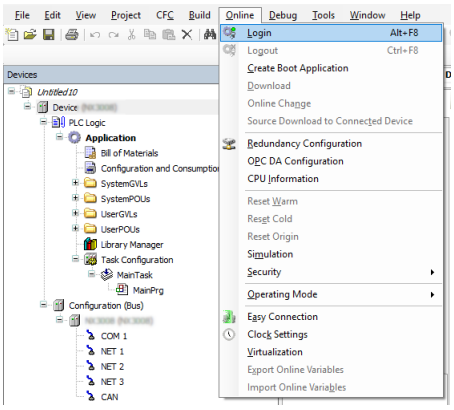


Figura 29: Enviando o Projeto para a UCP

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

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

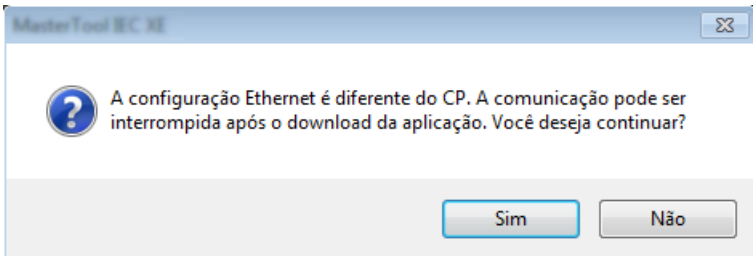


Figura 30: Aviso de configuração IP

Caso não exista aplicação no controlador, a seguinte mensagem será apresentada.

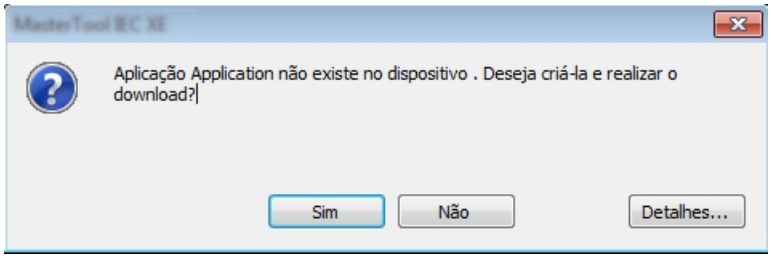


Figura 31: Sem aplicação no dispositivo

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.

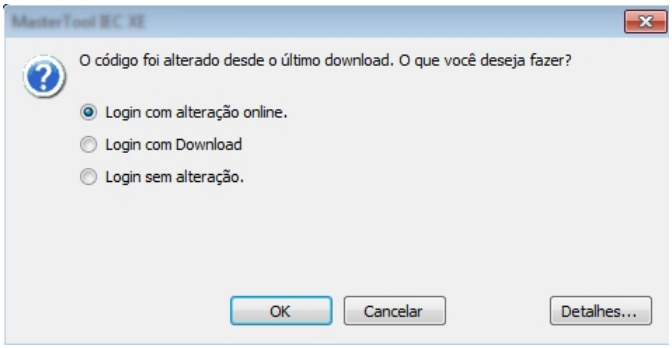


Figura 32: Atualização do Projeto na UCP

ATENÇÃO

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

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

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

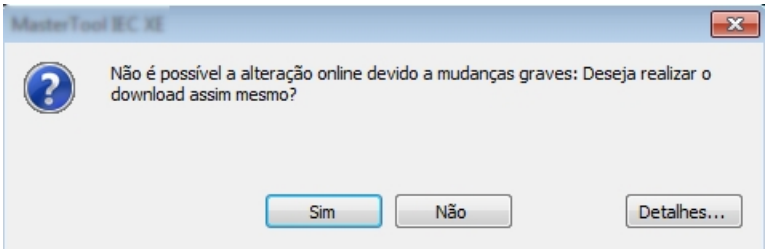


Figura 33: Alteração de Configuração

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.

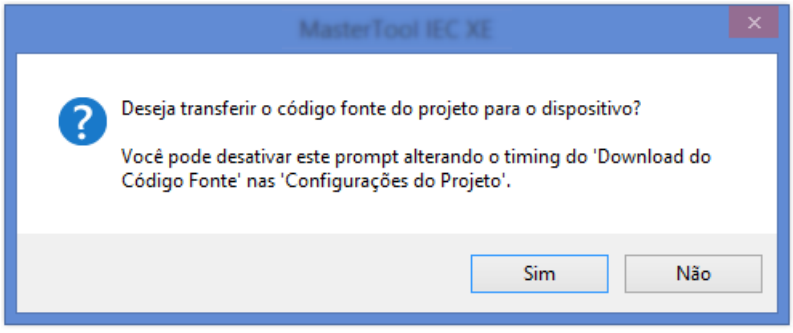


Figura 34: Download de código fonte

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 primeiro ciclo.

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

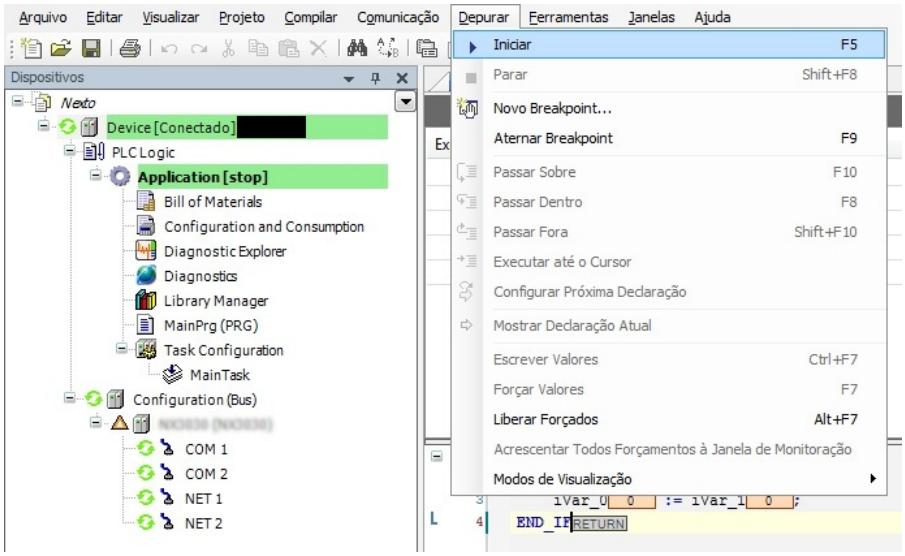


Figura 35: Iniciando a Aplicação

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

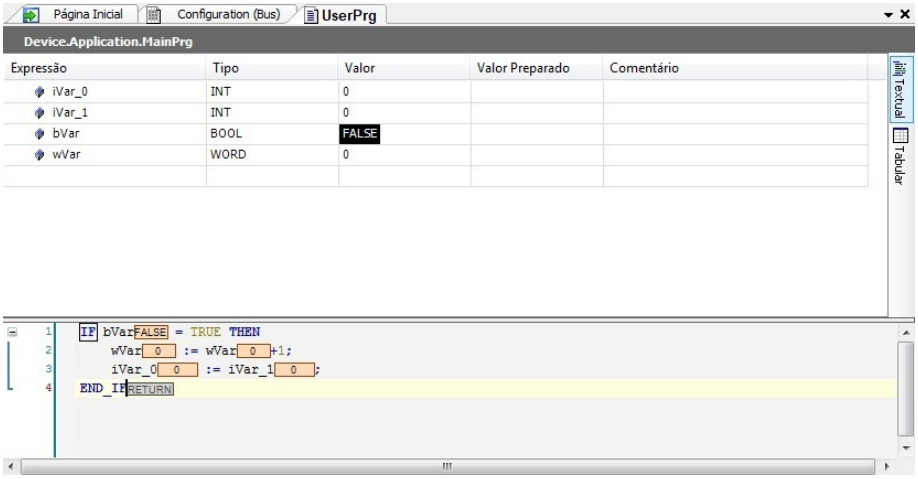


Figura 36: Programa em Execução

Caso o controlador já tenha uma aplicação de inicialização gravada internamente, ele entra automaticamente em Modo Run quando o sistema é energizado, sem a necessidade de realizar o comando online através do Mastertool.

4.9. Modo Stop

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

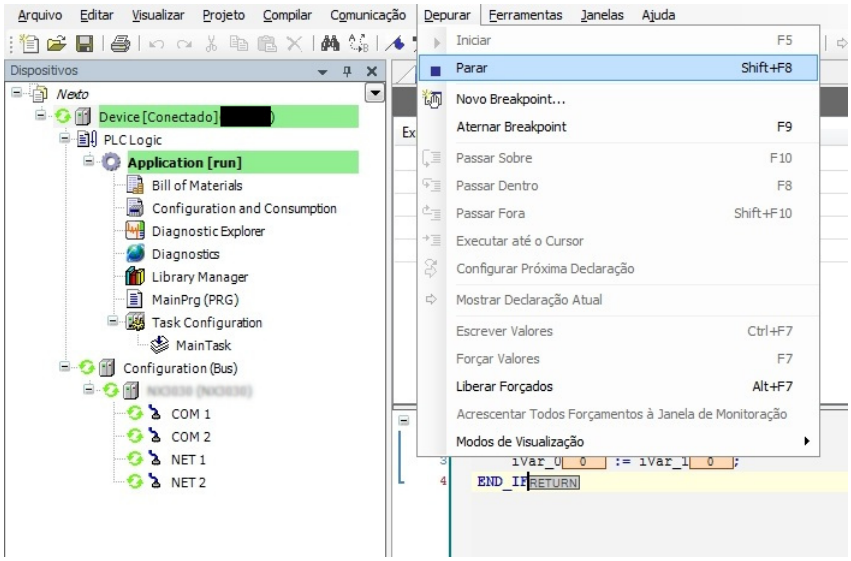


Figura 37: Parando a Aplicação

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 é de 128 variáveis.

4.11. Logout

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

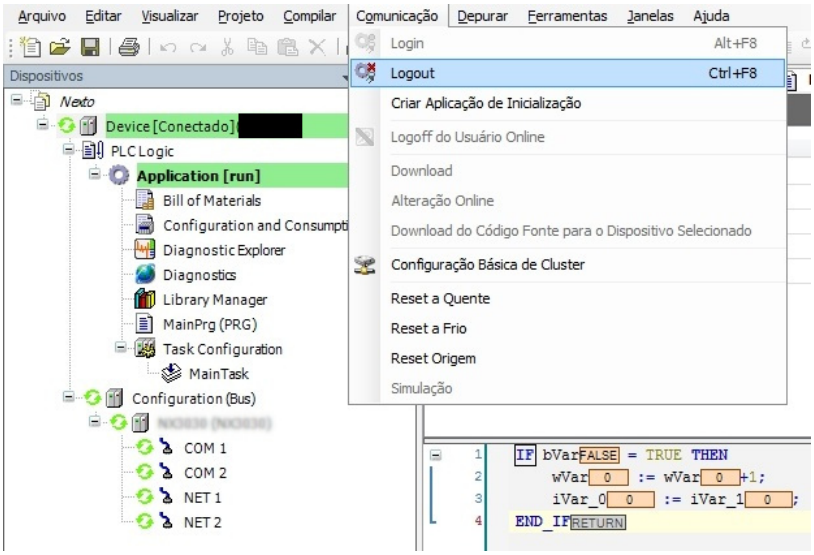


Figura 38: Encerrando a comunicação online com a UCP

4.12. Upload de Projeto

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

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

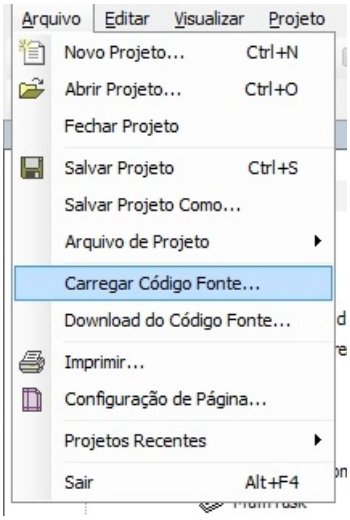


Figura 39: Opção de Upload de Projeto

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

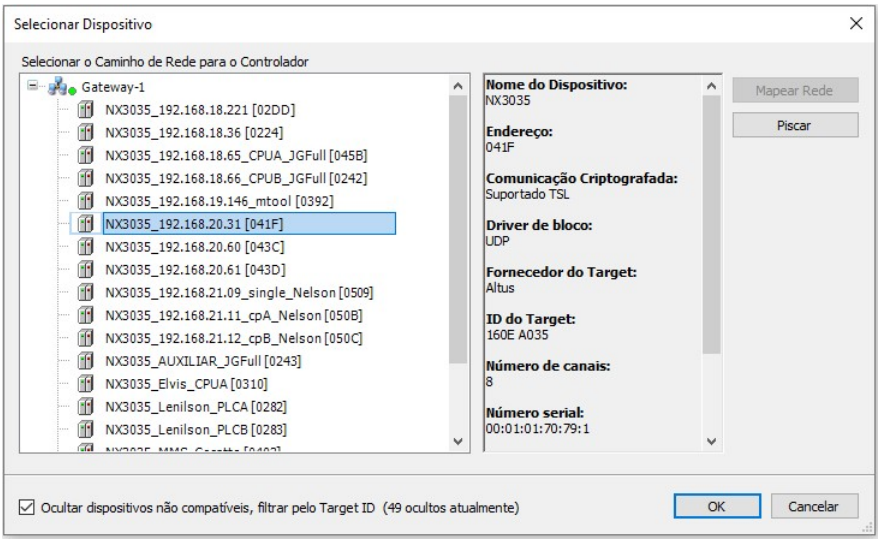


Figura 40: Selecionando a UCP

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 está definido na seção Memória.

ATENÇÃO

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

4.13. Estados de Operação da UCP

4.13.1. Run

Quando o controlador está em modo *Run*, todas as tarefas da aplicação são executadas.

4.13.2. Stop

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

Quando o controlador 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 o controlador 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 do controlador. Portanto somente com um *Reset a Quente*, *Reset a Frio*, *Reset Origem* ou uma reinicialização do controlador coloca novamente a aplicação em modo Run.

4.13.5. Reset a Quente

Este comando coloca o controlador 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 o controlador 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 de Origem

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

Notas:

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

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

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

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

4.14.1. Programa MainPrg

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

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

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

PROGRAM MainPrg
VAR
   isFirstCycle : BOOL := TRUE;
END_VAR

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

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

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

4.14.2. Programa StartPrg

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

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

4.14.3. Programa UserPrg

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

O usuário pode também criar 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*.

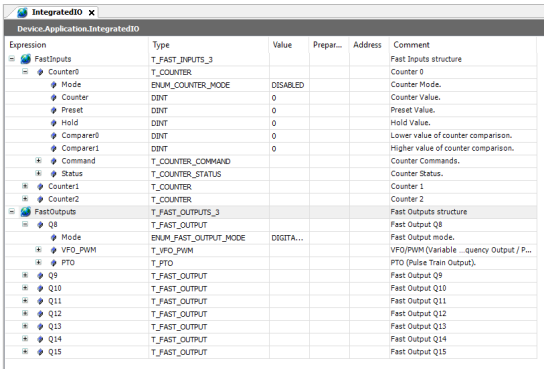


Figura 41: IntegratedIO GVL in Online Mode

4.14.5. GVL System_Diagnostics

Na GVL *System_Diagnostics* estão presentes as variáveis de diagnóstico da UCP, da comunicação e dos demais dispositivos declarados no projeto. Essa GVL não é editável e as variáveis são declaradas automaticamente, com o tipo especificado pelo dispositivo ao qual pertencem, quando este é adicionado ao projeto.

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

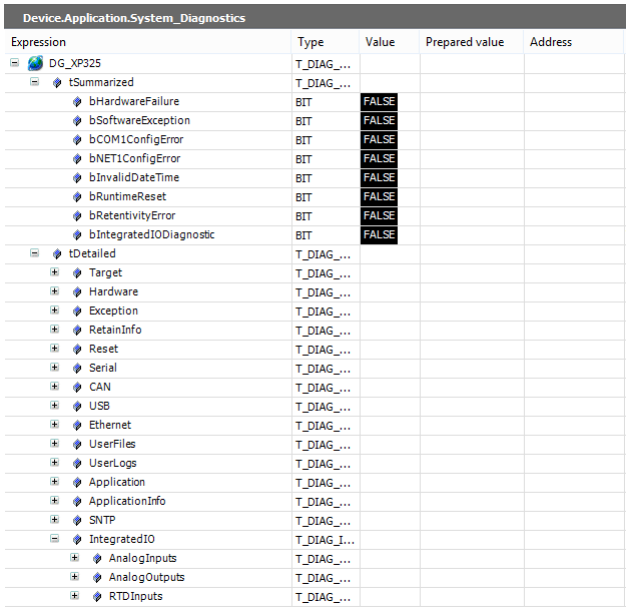


Figura 42: GVL System_Diagnostics em Modo Online

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.

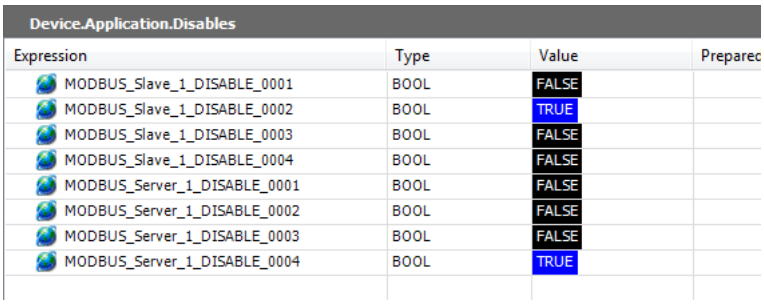


Figura 43: GVL Disables em Modo Online

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 visualização em árvore para o dispositivo.

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

Exemplo:

Device.Application.Qualities

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

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

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

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 44: 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 Diagnósticos* na aba de requisições do dispositivo. A declaração destas variáveis segue a seguinte estrutura:

Declaração de variável de diagnóstico de requisição:

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

Onde:

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

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

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

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 45: GVL ReqDiagnostics em Modo Online

5. Configuração

Os controladores do Sistema Nexto são configurados e programados através do software Mastertool. A configuração realizada define o comportamento e modos de utilização dos periféricos e características especiais dos controladores. 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 ao controlador. Utilizando o software Mastertool é possível criar e gerenciar usuários e grupos, configurando diferentes níveis de direitos de acesso ao projeto para estes.

Simultaneamente, os controladores possuem um sistema de gerenciamento de permissões de usuário, que bloqueia ou permite certas ações para cada grupo de usuários no controlador. 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.

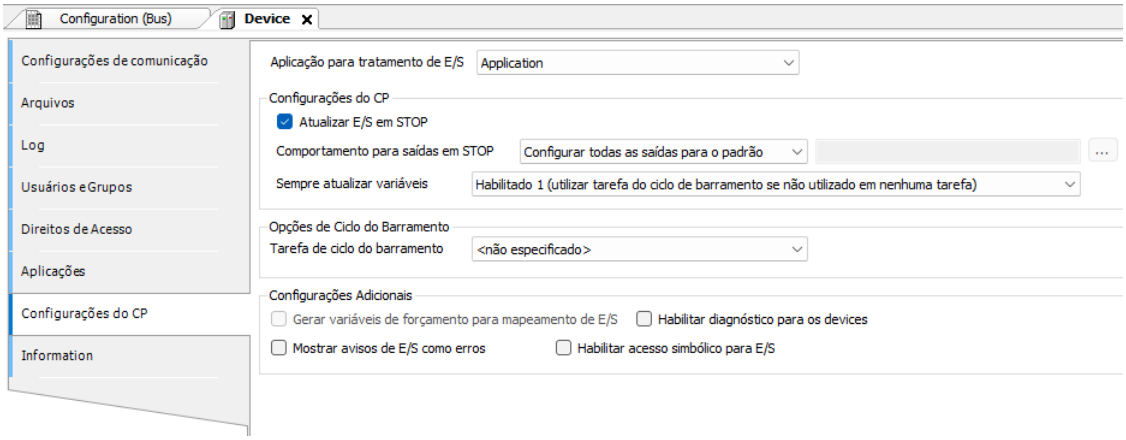


Figura 46: Configurações do CP

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.
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.

Parâmetro	Descrição
Sempre atualizar variáveis	Define globalmente se as variáveis de E/S são ou não atualizadas na tarefa de ciclo do barramento. Esta configuração é efetiva para as variáveis de E/S dos escravos e módulos somente se "desativado" for definido em suas configurações de atualização. **Desativado (atualiza somente se usado em uma tarefa):** As variáveis de E/S são atualizadas somente se forem usadas em uma tarefa. **Habilitado 1 (usar tarefa de ciclo de barramento se não for usada em nenhuma tarefa):** As variáveis de E/S na tarefa de ciclo de barramento são atualizadas se não forem usadas em nenhuma outra tarefa. **Habilitado 2 (sempre na tarefa de ciclo de barramento):** Todas as variáveis em cada ciclo da tarefa de ciclo de barramento são atualizadas, independentemente de serem usadas e mapeadas para um canal de entrada ou saída.
Tarefa de ciclo de barramento	Tarefa que controla o ciclo do barramento. Por padrão, a tarefa definida pela descrição do dispositivo é inserida. Por padrão, a configuração do ciclo de barramento do dispositivo de barramento superior se aplica (use as configurações de ciclo do barramento superior). Isso significa que a árvore de dispositivos é pesquisada para cima para a próxima definição válida da tarefa de ciclo de barramento.
Gerar variáveis de forçamento para mapeamento de E/S	**TRUE:** Ao compilar o aplicativo, duas variáveis globais são criadas para cada canal de E/S que é mapeado para uma variável na caixa de diálogo Mapeamento de E/S.
Habilitar diagnóstico para os devices	**TRUE:** A biblioteca CAA Device Diagnosis é integrada ao 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 *Channel* na aba de *Bus I/O Mapping* de cada dispositivo.

Tabela 35: Configurações do CP

ATENÇÃO

Este controlador não possui suporte ao parâmetro *Habilitar acesso simbólico para as E/S*.

5.2. Configuração da UCP

5.2.1. Parâmetros Gerais

Os parâmetros relacionados à CPU do controlador estão localizados na árvore de projetos, no item XF3xx, logo abaixo de *Configuração*. Cada item deve ser verificado corretamente para a execução adequada do projeto. Esses parâmetros são descritos abaixo:

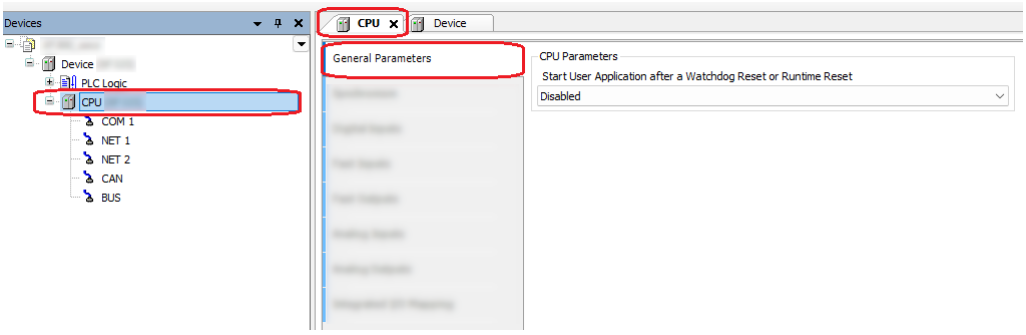


Figura 47: Configuração da UCP

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 watchdog do hardware ou através da reinicialização do Runtime, mas mantém a indicação do diagnóstico via LED DG e via variáveis	Desabilitado	Habilitado Desabilitado

Tabela 36: Configuração da UCP

5.2.2. Sincronização de Tempo

Para a sincronização de tempo, os controladores usam o protocolo SNTP (*Simple Network Time Protocol*).

Para utilizar os protocolos de sincronização de tempo, o usuário deve configurar os seguintes parâmetros na aba *Sincronismo*, acessível através do controlador, na árvore de dispositivos:

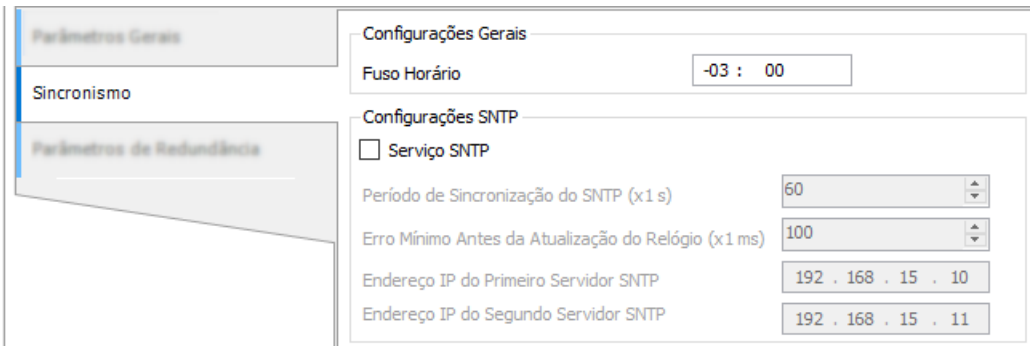


Figura 48: Configuração de SNTP

Configuração	Descrição	Padrão	Opções
Fuso Horário (hh:mm)	Fuso horário de sua localidade. Pode ser inserida a hora e o minuto.	-3:00	-12:59 a +13:59
Serviço SNTP	Ativa o serviço SNTP.	Desabilitado	Desabilitado Habilitado
Período de Sincronização do SNTP (x1 s)	Intervalo de tempo das requisições de sincronização (segundos).	60	1 a 255

Configuração	Descrição	Padrão	Opções
Erro Mínimo Antes da Atualização do Relógio (x1 ms)	Valor de offset aceitável entre a hora do servidor e do cliente (milissegundos).	100	1 a 65519
Endereço IP do Primeiro Servidor SNTP	Endereço IP do servidor SNTP primário.	192.168.15.10	1.0.0.1 a 223.255.255.254
Endereço IP do Segundo Servidor SNTP	Endereço IP do servidor SNTP secundário.	192.168.15.11	1.0.0.1 a 223.255.255.254

Tabela 37: Configurações SNTP

Notas:

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

Fuso Horário: A configuração de fuso horário é utilizada para converter o horário local em horário UTC e vice-versa. Enquanto algumas fontes de sincronismo utilizam o horário local (exemplos: protocolo IEC 60870-5-104, função SetDateAndTime), outras utilizam o horário UTC (SNTP). O horário UTC normalmente é utilizado para estampar os eventos (exemplos: DNP3, IEC 60870-5-104, Mastertool Device LOG), enquanto que o horário local é utilizado por outras funcionalidades do controlador (exemplos: função GetDateAndTime, informação de data e hora do OTD).

É permitido habilitar mais de uma fonte de sincronismo no projeto, no entanto o equipamento não suporta durante a operação a sincronização do tempo por duas fontes distintas de sincronismo, para tanto existe implicitamente definido um mecanismo de prioridade. A sincronização do protocolo SNTP é a mais prioritária. Assim, quando houver mais fontes habilitadas e o servidor SNTP estiver presente, este será responsável pela sincronização do relógio do controlador e qualquer comando de sincronismo de outras fontes será negado.

5.2.2.1. 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, porém com uma precisão de 100 ms. A precisão do sincronismo de tempo por SNTP depende das configurações do protocolo SNTP (erro mínimo para atualização do relógio) e das características da rede Ethernet onde está sendo aplicado, se cliente e servidor SNTP estão na mesma rede (local) ou em redes diferentes (remota). Tipicamente a precisão é da ordem de dezenas de milissegundos.

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 Primeiro Servidor. Caso este não responda, as requisições são direcionadas para o segundo servidor configurado em Endereço de IP do Segundo Servidor, fornecendo uma redundância de servidores SNTP. Caso o segundo servidor também não responda, o mesmo processo de tentativa de sincronização é executado novamente, mas apenas após o *Período de Sincronização* ter passado. Ou seja, a cada período de sincronização, 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 (P*eríodo de Sincronização*) e tentará sincronizar-se novamente com este servidor, utilizando, então, este como servidor primário. Em caso de falha (o servidor não responde em menos de 5 s) é aguardado o tempo referente ao Período de Sincronização e executado novamente o passo 1.

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

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

ATENÇÃO

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

5.2.2.2. 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 diminui-lo em uma hora.

Para maiores informações, consultar a seção Relógio RTC.

5.3. Configuração da Interface Serial

5.3.1. COM 1

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

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

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

Configuração	Descrição	Padrão	Opções
Tipo Serial	Configuração do tipo do canal serial.	RS-485	RS-485
Taxa de Transmissão	Velocidade da porta de comunicação serial.	115200	2400, 4800, 9600, 19200, 38400, 57600, 115200 bps
Paridade	Configura a paridade da porta serial.	Sem Paridade	Ímpar Par Sem Paridade
Bits de Dado	Configura o número de bits de dados em cada caractere da comunicação serial.	8	7 e 8
Bits de Parada	Configura os bits de parada da porta serial.	1	1 e 2
Modo Serial	Configura o modo de operação da porta serial.	Modo Normal	- Modo Estendido: Modo estendido de operação da comunicação serial, no qual são fornecidas informações sobre o frame de dados recebido (ver notas). - Modo Normal: Modo normal de operação da comunicação serial

Tabela 38: Configuração da Interface Serial Padrão RS-485

Notas:

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

- Um byte para o dado recebido (RX_CHAR : BYTE): Armazena os sete ou oito bits dos dados recebidos, dependendo da configuração da serial.
- Um byte para os erros no sinal (RX_ERROR : BYTE): Tem o seguinte formato:
  - Bit 0: 0 - o caractere nos bits 0 a 7 é válido. 1 - o caractere nos bits 0 a 7 não é válido (ou pode não ser válido), devido aos problemas indicados nos bits 10 a 15.
  - Bit 1: Não utilizado.
  - Bit 2: Não utilizado.
  - Bit 3: Erro de BREAK na UART. A entrada serial permaneceu na lógica 0 (paridade sempre zero) por um tempo maior do que um caractere (bit de partida + bit de dados + bit de paridade + bit de parada).
  - Bit 4: Erro no frame UART. A lógica 0 (paridade sempre zero) foi lida quando o primeiro bit de parada era esperado, sendo que deveria ser lógica 1 (paridade sempre um).
  - Bit 5: Erro de paridade UART. O bit de paridade lido não está de acordo com o bit de paridade calculado.
  - Bit 6: Erro de overrun UART. Dados foram perdidos durante a leitura do FIFO UART, porque novos caracteres foram recebidos antes de os antigos serem removidos. Esse erro somente será indicado no primeiro caractere lido após a indicação do erro de overrun. Isso significa que alguns dados antigos foram perdidos.
  - Bit 7: Erro de overrun na fila RX: Esse caractere foi escrito quando a fila RX foi completa, sobrescrevendo caracteres não lidos.
- Dois bytes para o sinal de estampa de tempo (RX_TIMESTAMP : DWORD): Indica o tempo de silêncio, no intervalo de 0 a 4.294.967.295, usando como base de tempo 1 us. Satura em 4.294.967.295 us, caso o tempo de silêncio seja maior do que 4.294.967.295 unidades. O RX_TIMESTAMP de um caractere mede o tempo desde uma referência que pode ser uma das três opções abaixo:
  - Na maioria dos casos, o final do caractere anterior.
  - Configuração da porta serial.
  - O fim da transmissão serial usando o SERIAL_TX FB, ou seja, quando o último caractere foi enviado na linha.

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

5.3.2. Configurações Avançadas

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

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

Tabela 39: Configurações Avançadas da Interface Serial RS-485

5.4. Configuração das Interfaces Ethernet

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

Abaixo, seguem os parâmetros de endereçamento IP que devem ser configurados para o bom funcionamento da aplicação.

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

Tabela 40: Configurações NET 1

5.4.2. NET 2

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

Tabela 41: Configurações NET 2

5.4.3. Modos de Operação das Interfaces Ethernet Integradas

5.4.3.1. Modo Simples

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

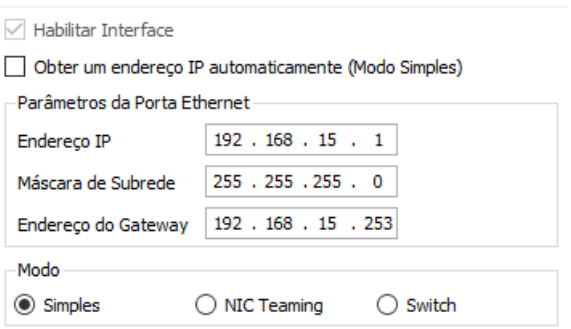


Figura 49: Configurações Avançadas das Interfaces Ethernet - Modo Simples

5.4.3.2. Modo NIC Teaming

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

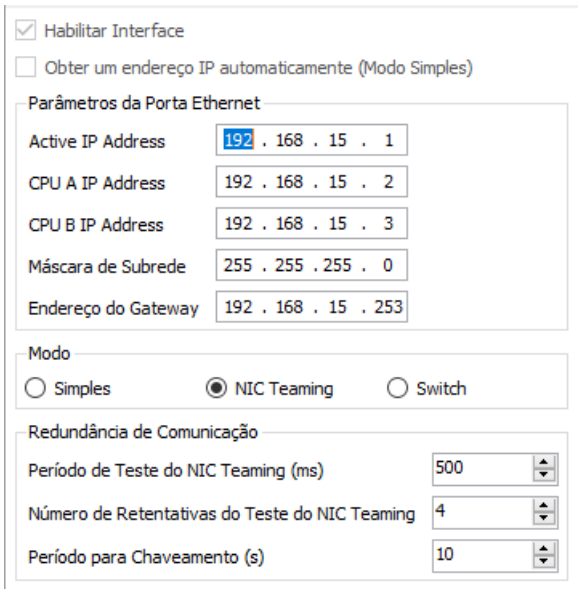


Figura 50: Configurações Avançadas das Interfaces Ethernet - Modo NIC Teaming

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

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

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

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

5.4.3.3. Modo Switch

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

5.4.3.3.1. Modo Switch - Desabilitado

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

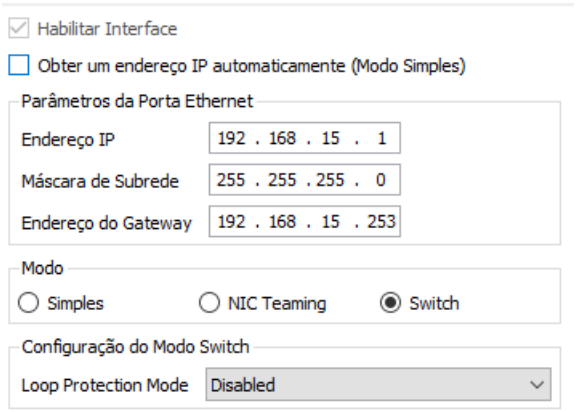


Figura 51: Configurações Avançadas das Interfaces Ethernet - Modo Switch - Loop Protection Disabled

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

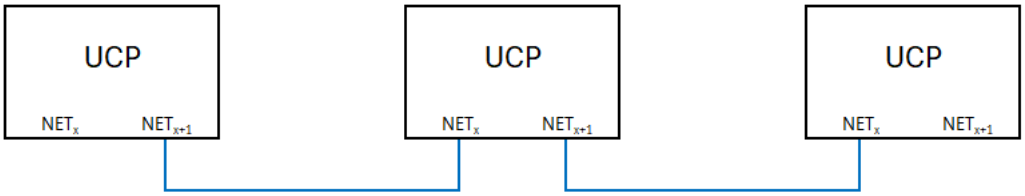


Figura 52: Exemplo de Topologia com o Modo Switch - Loop Protection Disabled

5.4.3.3.2. Modo Switch - RSTP

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

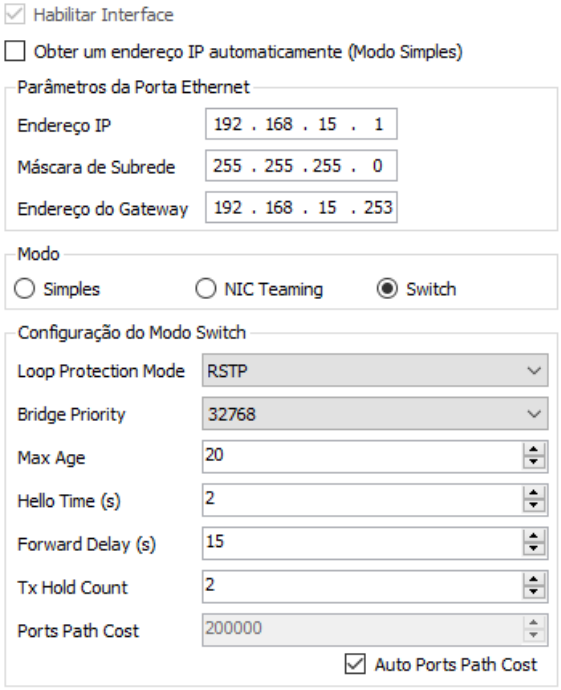


Figura 53: Configurações Avançadas das Interfaces Ethernet - Modo Switch - Loop Protection RSTP

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

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

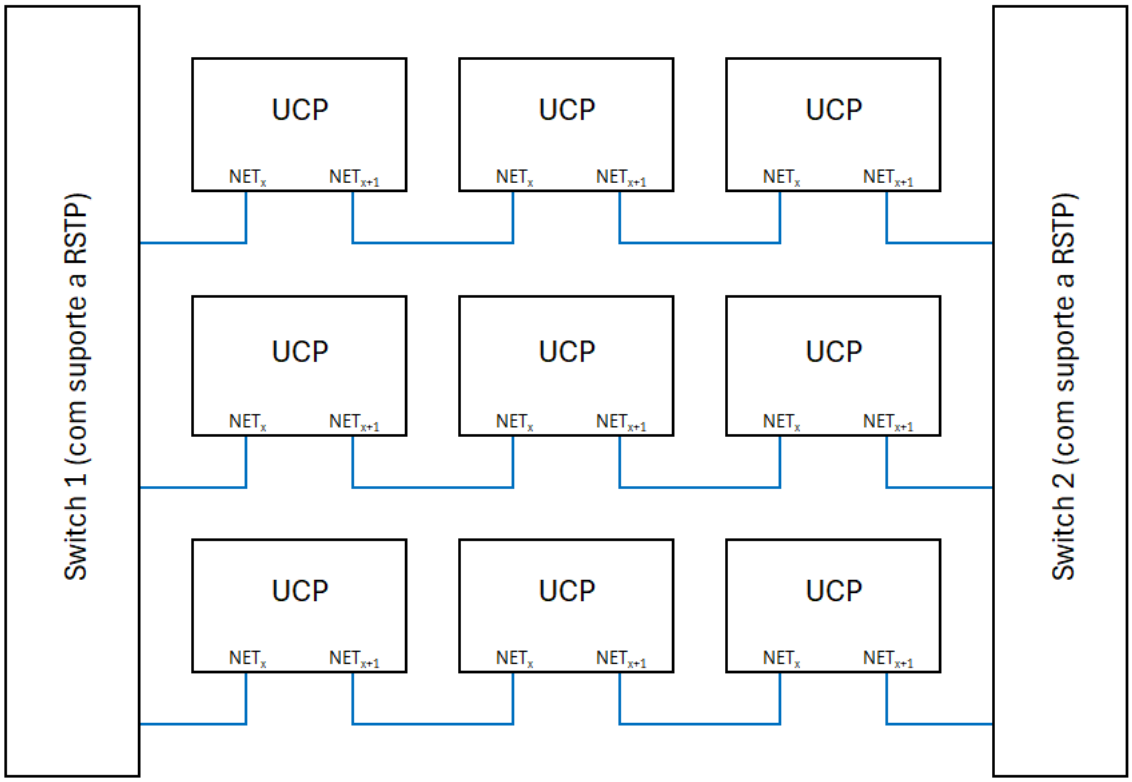


Figura 54: Exemplo de Topologia Mesh com o Modo Switch - RSTP

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

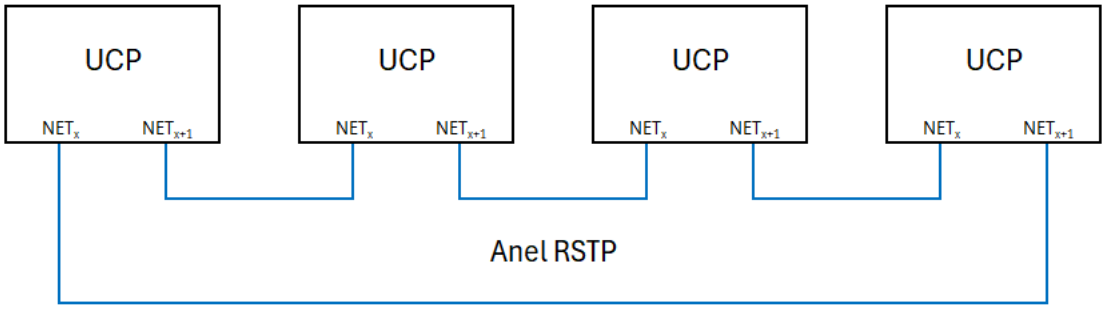


Figura 55: Exemplo de Topologia em Anel com o Modo Switch - RSTP

5.4.3.3.3. Modo Switch - MRP

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

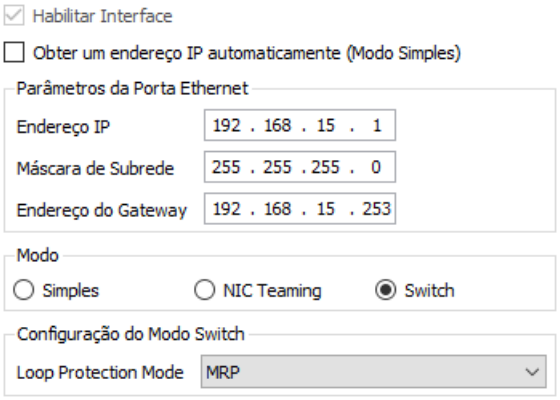


Figura 56: Configurações Avançadas das Interfaces Ethernet - Modo Switch - MRP

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

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

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

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

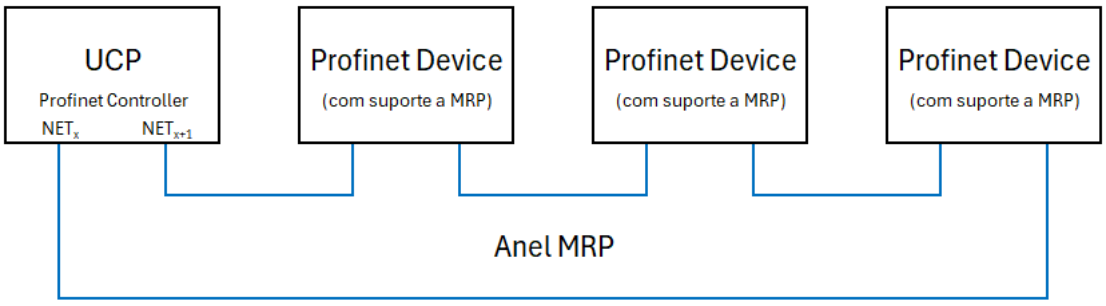


Figura 57: Exemplo de Topologia em Anel com o Modo Switch - MRP

5.4.4. Portas TCP/UDP Reservadas

As seguintes portas TCP/UDP das interfaces Ethernet, tanto integradas quanto remotas, são utilizadas por serviços do controlador (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 42: Portas TCP/UDP reservadas

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

5.4.5. Rotamento 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)
   - Outra interface NET integrada configurada para a mesma rede do endereço de destino (neste caso, a comunicação será roteada internamente)

   Se nenhuma dessas condições ocorrer, o sistema operacional encaminhará a comunicação para o Gateway Padrão configurado nas interfaces NET, iniciando pelo configurado na NET 1 e passando pelos seguintes se a comunicação falhar.

2. Protocolos Servidor aceitarão conexões apenas na interface onde estão instanciados, exceto para o MODBUS, onde um parâmetro 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 43: CAN Configuration

Os parâmetros relacionados ao protocolo CANOpen estão descritos na seção CANopen Manager.

5.6. Configuração das E/S Integradas

Os controladores Nexto XF 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 árvore de projetos relacionados a E/S integradas, conforme mostrado na figura abaixo:

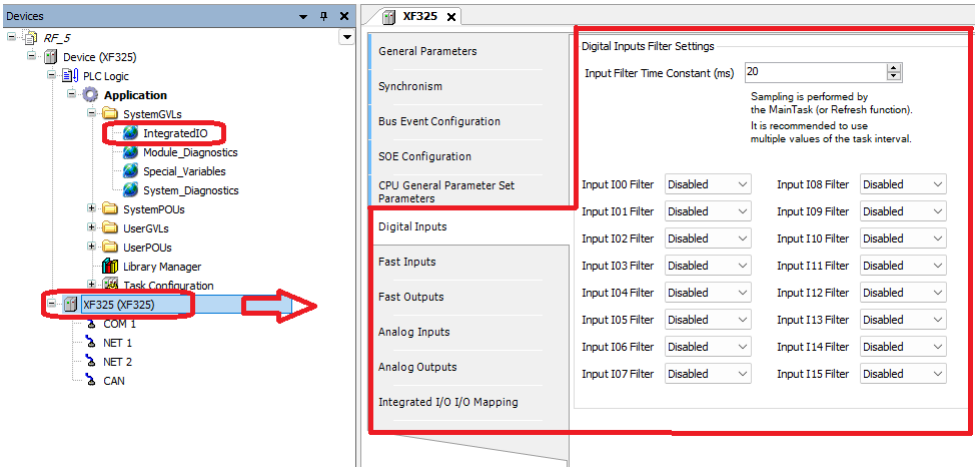


Figura 58: Objetos de E/S integrados na visualização em árvore do projeto

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

5.6.1. Entradas Digitais

Os parâmetros relacionados às *Entradas Digitais* estão localizados na tela abaixo (exemplo do XF325), tanto para E/S padrão quanto para E/S rápida (quando configuradas como entradas digitais padrão):

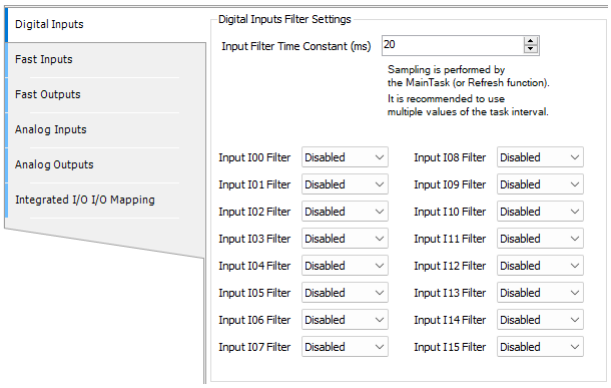


Figura 59: Parâmetros de Entradas Digitais

A tabela abaixo mostra os possíveis valores de configuração:

Configuração	Descrição	Desabilitado	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	Ativar/Desativar filtro para cada entrada	Desabilitado	Habilitado ou Desabilitado

Tabela 44: Parâmetros de Entradas Digitais

Nota:

Constante de tempo do filtro de entrada: A amostragem do filtro é realizada na (MainTask), portanto, recomenda-se usar vários valores para o intervalo da tarefa.

ATENÇÃO

As entradas digitais padrão de I00 a I06 têm um tempo de resposta de atualização de 2 ms. Portanto, o usuário pode observar um atraso de um ciclo da Main Task nessas entradas.

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								
		I07	I08	I09	I10	I11	I12	I13	I14	I15
Contador 0	Up/Down (A conta, B direção) com zeramento	A	B	Z	-	-	-	-	-	-
	Quadratura 2X	A	B	-	-	-	-	-	-	-
	Quadratura 2X com zeramento	A	B	Z	-	-	-	-	-	-
	Quadratura 4X	A	B	-	-	-	-	-	-	-
	Quadratura 4X com zeramento	A	B	Z	-	-	-	-	-	-
Contador 1	Up/Down (A conta, B direção) com zeramento	-	-	-	A	B	Z	-	-	-
	Quadratura 2X	-	-	-	A	B	-	-	-	-
	Quadratura 2X com zeramento	-	-	-	A	B	Z	-	-	-
	Quadratura 4X	-	-	-	A	B	-	-	-	-
	Quadratura 4X com zeramento	-	-	-	A	B	Z	-	-	-
Contador 2	Up/Down (A conta, B direção) com zeramento	-	-	-	-	-	-	A	B	Z
	Quadratura 2X	-	-	-	-	-	-	A	B	-
	Quadratura 2X com zeramento	-	-	-	-	-	-	A	B	Z
	Quadratura 4X	-	-	-	-	-	-	A	B	-
	Quadratura 4X com zeramento	-	-	-	-	-	-	A	B	Z

Tabela 45: 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:

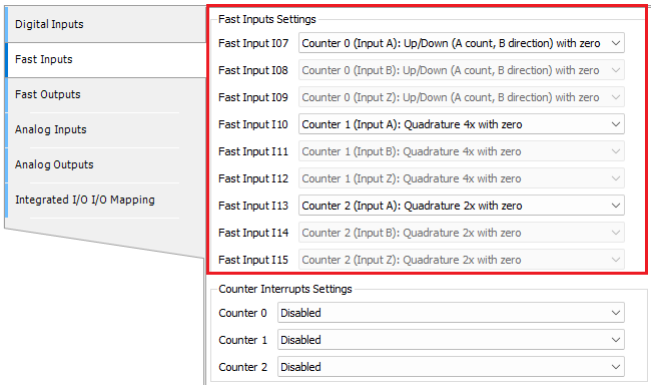


Figura 60: Configuração de Entradas Rapidas

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:

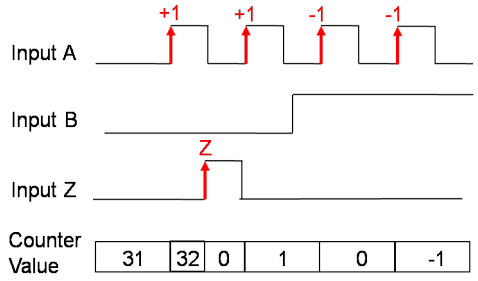
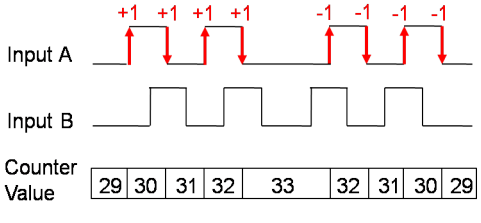
Configuração	Descrição	Padrão	Opções
Rápida Entrada I07	Entrada Rápida I07 configuração	Entrada Digital	Entrada Digital Contador 0 (Entrada A): Incremento/Decremento (A contagem, B direção) com zero Contador 0 (Entrada A): Quadratura 2X Contador 0 (Entrada A): Quadratura 2X com zero Contador 0 (Entrada A): Quadratura 4X Contador 0 (Entrada A): Quadratura 4X com zero Interrupção (Borda de Subida e Borda de Descida) Interrupção (Borda de Descida) Interrupção (Borda de Subida)
Rápida Entrada I08	Entrada Rápida I08 configuração	Entrada Digital	Entrada Digital Interrupção (Borda de Subida e Borda de Descida) Interrupção (Borda de Descida) Interrupção (Borda de Subida) Obs: Este campo será configurado automaticamente quando a Entrada Rápida I07 for configurada como Contador Incremento/Decremento ou Contador de Quadratura.
Rápida Entrada I09	Entrada Rápida I09 configuração	Entrada Digital	Entrada Digital Interrupção (Borda de Subida e Borda de Descida) Interrupção (Borda de Descida) Interrupção (Borda de Subida) Obs: Este campo será configurado automaticamente quando a Entrada Rápida I07 for configurada como Contador Incremento/Decremento ou Contador de Quadratura com zero.
Rápida Entrada I10	Entrada Rápida I10 configuração	Entrada Digital	Entrada Digital Contador 1 (Entrada A): Incremento/Decremento (A contagem, B direção) com zero Contador 1 (Entrada A): Quadratura 2X Contador 1 (Entrada A): Quadratura 2X com zero Contador 1 (Entrada A): Quadratura 4X Contador 1 (Entrada A): Quadratura 4X com zero Interrupção (Borda de Subida e Borda de Descida) Interrupção (Borda de Descida) Interrupção (Borda de Subida)
Rápida Entrada I11	Entrada Rápida I11 configuração	Entrada Digital	Entrada Digital Interrupção (Borda de Subida e Borda de Descida) Interrupção (Borda de Descida) Interrupção (Borda de Subida) Obs: Este campo será configurado automaticamente quando a Entrada Rápida I10 for configurada como Contador Incremento/Decremento ou Contador de Quadratura.
Rápida Entrada I12	Entrada Rápida I12 configuração	Entrada Digital	Entrada Digital Interrupção (Borda de Subida e Borda de Descida) Interrupção (Borda de Descida) Interrupção (Borda de Subida) Obs: Este campo será configurado automaticamente quando a Entrada Rápida I10 for configurada como Contador Incremento/Decremento ou Contador de Quadratura com zero.
			Entrada Digital Contador 2 (Entrada A): Incremento/Decremento (A contagem, B direção) com zero

Configuração	Descrição	Padrão	Opções
Rápida Entrada I13	Entrada Rápida I13 configuração	Entrada Digital	Contador 2 (Entrada A): Quadratura 2X Contador 2 (Entrada A): Quadratura 2X com zero Contador 2 (Entrada A): Quadratura 4X Contador 2 (Entrada A): Quadratura 4X com zero Interrupção (Borda de Subida e Borda de Descida) Interrupção (Borda de Descida) Interrupção (Borda de Subida)
Rápida Entrada I14	Entrada Rápida I14 configuração	Entrada Digital	Entrada Digital Interrupção (Borda de Subida e Borda de Descida) Interrupção (Borda de Descida) Interrupção (Borda de Subida) Obs: Este campo será configurado automaticamente quando a Entrada Rápida I13 for configurada como Contador Incremento/Decremento ou Contador de Quadratura.
Rápida Entrada I15	Entrada Rápida I15 configuração	Entrada Digital	Entrada Digital Interrupção (Borda de Subida e Borda de Descida) Interrupção (Borda de Descida) Interrupção (Borda de Subida) Obs: Este campo será configurado automaticamente quando a Entrada Rápida I13 for configurada como Contador Incremento/Decremento ou Contador de Quadratura com zero.

Tabela 46: Parâmetros das Entradas Rápidas

5.6.2.1. Contadores Rápidos

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

Modo do Contador	Formas de onda de contagem
Incremento/Decremento (A contagem, B direção) com zero	
Quadratura 2X	

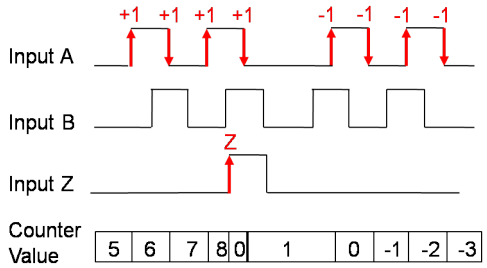
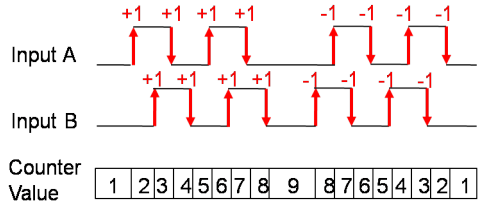
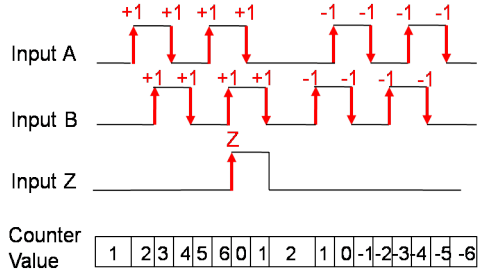
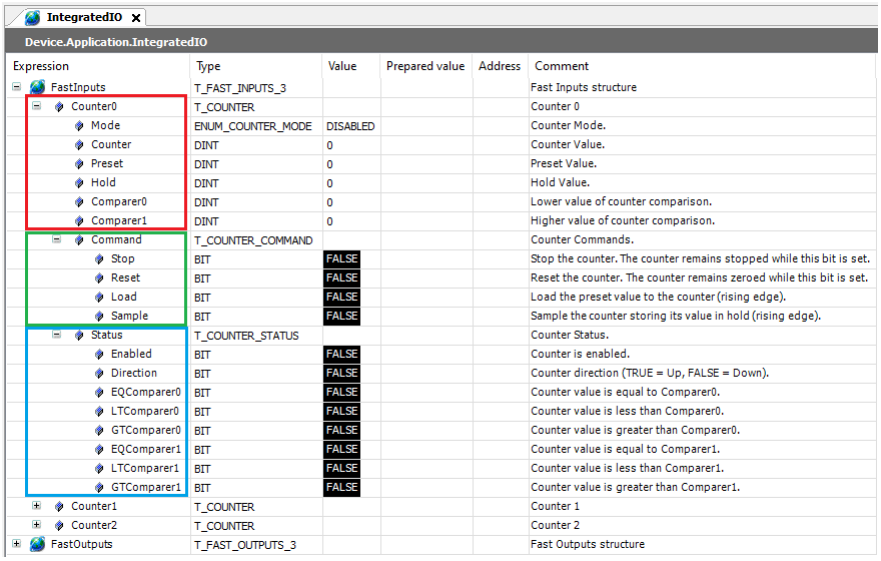
Modo do Contador	Formas de onda de contagem
Quadratura 2X com zero	
Quadratura 4X	
Quadratura 4X com zero	

Tabela 47: Modos de contadores rápidos

O comportamento geral é o mesmo para todos os contadores: quando se incrementa o contador e o valor positivo máximo é atingido, o próximo valor será o valor negativo mínimo. O mesmo ocorre na direção oposta, ou seja, decrementa o contador e o valor negativo mínimo for atingido, o próximo valor será o valor positivo máximo.

O programa do usuário pode acessar os contadores de alta velocidade por meio da estrutura simbólica 'FastInputs', que é criada automaticamente na GVL 'IntegratedIo'. Para cada unidade de contador rápido, existem 3 áreas principais, conforme mostrado na figura a seguir:



Expression	Type	Value	Prepared value	Address	Comment
FastInputs	T_FAST_INPUTS_3				Fast Inputs structure
Counter0	T_COUNTER				Counter 0
Mode	ENUM_COUNTER_MODE	DISABLED			Counter Mode.
Counter	DINT	0			Counter Value.
Preset	DINT	0			Preset Value.
Hold	DINT	0			Hold Value.
Comparer0	DINT	0			Lower value of counter comparison.
Comparer1	DINT	0			Higher value of counter comparison.
Command	T_COUNTER_COMMAND				Counter Commands.
Stop	BIT	FALSE			Stop the counter. The counter remains stopped while this bit is set.
Reset	BIT	FALSE			Reset the counter. The counter remains zeroed while this bit is set.
Load	BIT	FALSE			Load the preset value to the counter (rising edge).
Sample	BIT	FALSE			Sample the counter storing its value in hold (rising edge).
Status	T_COUNTER_STATUS				Counter Status.
Enabled	BIT	FALSE			Counter is enabled.
Direction	BIT	FALSE			Counter direction (TRUE = Up, FALSE = Down).
EQComparer0	BIT	FALSE			Counter value is equal to Comparer0.
LTComparer0	BIT	FALSE			Counter value is less than Comparer0.
GTComparer0	BIT	FALSE			Counter value is greater than Comparer0.
EQComparer1	BIT	FALSE			Counter value is equal to Comparer1.
LTComparer1	BIT	FALSE			Counter value is less than Comparer1.
GTComparer1	BIT	FALSE			Counter value is greater than Comparer1.
Counter1	T_COUNTER				Counter 1
Counter2	T_COUNTER				Counter 2
FastOutputs	T_FAST_OUTPUTS_3				Fast Outputs structure

Figura 61: Estrutura dos contadores

A tabela abaixo descreve a estrutura das variáveis do contador:

Variável	Descrição	Tipo	Valores Permitidos
Mode	Modo do contador configurado (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 48: Estruturas 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. O contador permanece parado enquanto este bit estiver ativado	BIT	FALSE ou TRUE
Reset	Zera o contador. O contador permanece zerado enquanto este bit estiver ativado	BIT	FALSE ou TRUE

Variável	Descrição	Tipo	Valores Permitidos
Load	Carrega o valor de preset para o valor do contador. Esta operação é executada na borda de subida deste bit	BIT	FALSE ou TRUE
Sample	Amostra o contador armazenando seu valor em hold. Esta operação é executada na borda de subida deste bit	BIT	FALSE ou TRUE

Tabela 49: Estrutura de comandos do contador

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

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

Tabela 50: Estrutura de status do contador

Além das variáveis globais do *IntegratedIo*, existe um bloco de função da biblioteca *LibIntegratedIo* que permite instanciar um contador rápido em POUs escritas em linguagens gráficas (por exemplo, diagramas de lógica Ladder). Este bloco de função é, na verdade, um encapsulamento das variáveis estruturadas descritas anteriormente. A figura abaixo mostra o bloco de função instanciado em um programa Ladder.

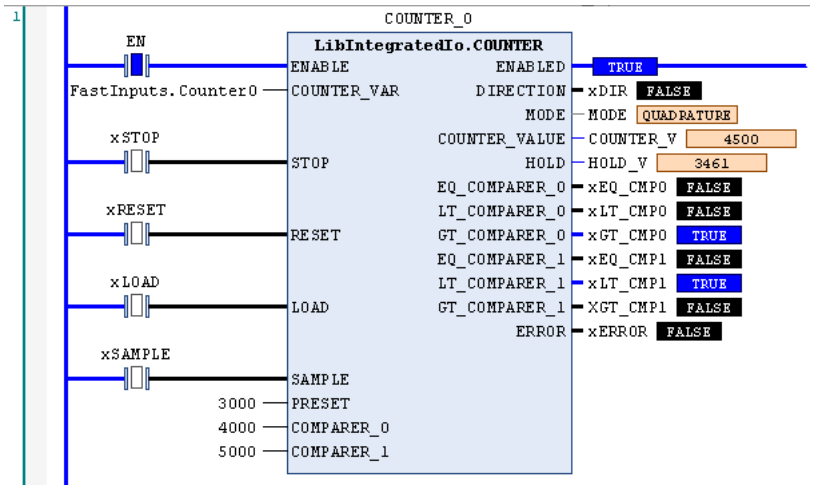


Figura 62: Bloco Funcional LibIntegratedIo.COUNTER

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

Variável	Descrição	Tipo	Valores Permitidos
ENABLE	Habilita a execução do bloco funcional	BOOL	FALSE ou TRUE
COUNTER_VAR	Variável do contador	REFERENCE TO T_COUNTER	FastInputs.Counter0
STOP	Para o contador	BOOL	FALSE ou TRUE
RESET	Zera o contador	BOOL	FALSE ou TRUE
LOAD	Carrega o valor de preset para o valor do contador	BOOL	FALSE ou TRUE
SAMPLE	Amostra o contador armazenando seu valor em hold	BOOL	FALSE ou TRUE
PRESET	Valor de preset	DINT	-2147483648 a 2147483647
COMPARER_0	Valor inferior da comparação do contador	DINT	-2147483648 a 2147483647
COMPARER_1	Valor superior da comparação do contador	DINT	-2147483648 a 2147483647
ENABLED	Contador habilitado	BOOL	FALSE ou TRUE
DIRECTION	Sentido 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 retenção	DINT	-2147483648 a 2147483647
EQ_COMPARER_0	Valor do contador é igual ao Comparer0	BOOL	FALSE ou TRUE
LT_COMPARER_0	Valor do contador é menor que o Comparer0	BOOL	FALSE ou TRUE
GT_COMPARER_0	Valor do contador é maior que o Comparer0	BOOL	FALSE ou TRUE
EQ_COMPARER_1	Valor do contador é igual ao Comparer1	BOOL	FALSE ou TRUE
LT_COMPARER_1	Valor do contador é menor que o Comparer1	BOOL	FALSE ou TRUE
GT_COMPARER_1	Valor do contador é maior que o Comparer1	BOOL	FALSE ou TRUE
ERROR	Erro ocorrido na execução do bloco funcional. Pode ser causado por COUNTER_VAR inválido ou contador desabilitado.	BOOL	FALSE ou TRUE

Tabela 51: Descrição do bloco funcional LibIntegratedIo.COUNTER

5.6.2.1.1. Interrupções de Contador

As unidades de contador rápido têm a capacidade de gerar interrupções por comparação, ou seja, 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 simbólica global correspondente ou FunctionBlock, conforme descrito nas seções anteriores. A configuração da interrupção do contador para cada unidade de contador rápido está localizada na tela a seguir:

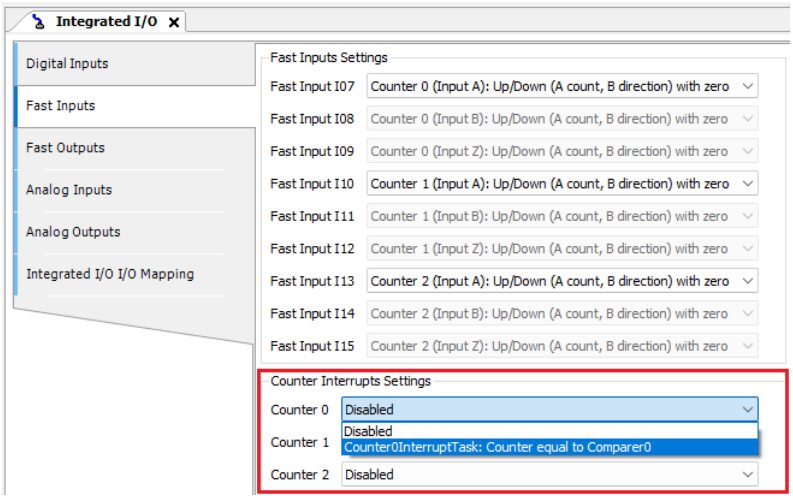


Figura 63: Configurações da interrupção do contador

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

Configuração	Descrição	Padrão	Opções
Counter 0	Configuração da interrupção por comparador do Counter0	Desabilitado	Desabilitado Counter0InterruptTask: Contador igual ao Comparer0 Obs: Esta configuração está disponível quando o Counter0 está associado a alguma Entrada Rápida.
Counter 1	Configuração da interrupção por comparador do Counter1	Desabilitado	Desabilitado Counter1InterruptTask: Contador igual ao Comparer0 Obs: Esta configuração está disponível quando o Counter1 está associado a alguma Entrada Rápida.
Counter 2	Configuração da interrupção por comparador do Counter2	Desabilitado	Desabilitado Counter2InterruptTask: Contador igual ao Comparer0 Obs: Esta configuração está disponível quando o Counter2 está associado a alguma Entrada Rápida.

Tabela 52: Parâmetros da interrupção do contador

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

A figura abaixo ilustra este cenário de configuração no Mastertool.

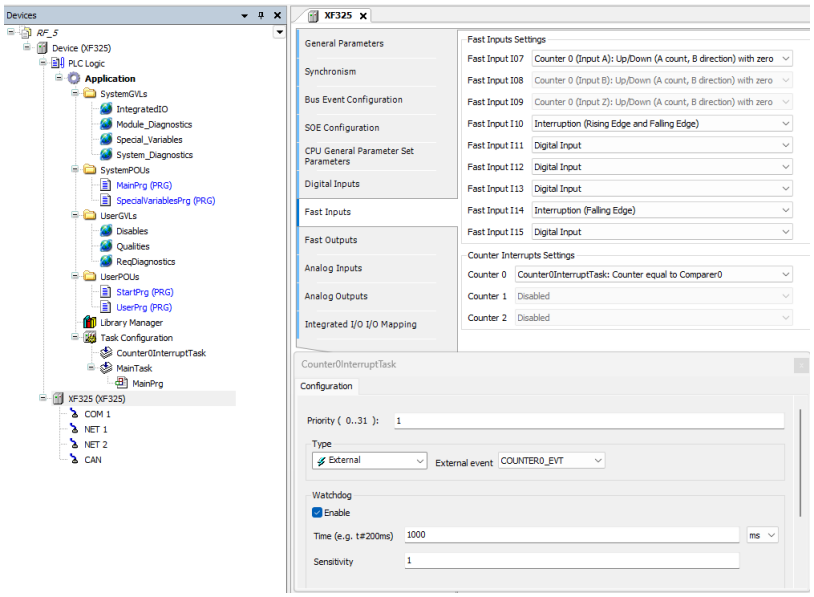


Figura 64: Configurações da Interrupção do Contador

5.6.2.2. Interrupções Externas

As entradas rápidas podem ser configuradas em três modos de interrupção diferentes: modo *Interrupção (Borda de Subida)*, modo *Interrupção (Borda de Descida)* ou modo *Interrupção (Borda de Subida e Borda de Descida)*. Isso significa que, quando a borda escolhida (transição 0V <–> 24V) ocorre na entrada, uma tarefa específica será executada e interromperá a execução do programa principal.

Cada interrupção externa gera um evento específico. Esse evento deve acionar a execução de uma tarefa de evento externo, que, por sua vez, deve chamar uma POU específica. Por exemplo, o evento de interrupção externa gerado para a entrada rápida I12 é chamado *FIN12_EVT*. Portanto, uma tarefa de evento externo chamada *FastInputI12InterruptTask* deve ser configurada para ser acionada por esse evento e deve chamar uma POU chamada *FastInputI12InterruptPrg*, que conterá o programa do usuário a ser executado.

A figura abaixo ilustra esse cenário de configuração no MasterTool.

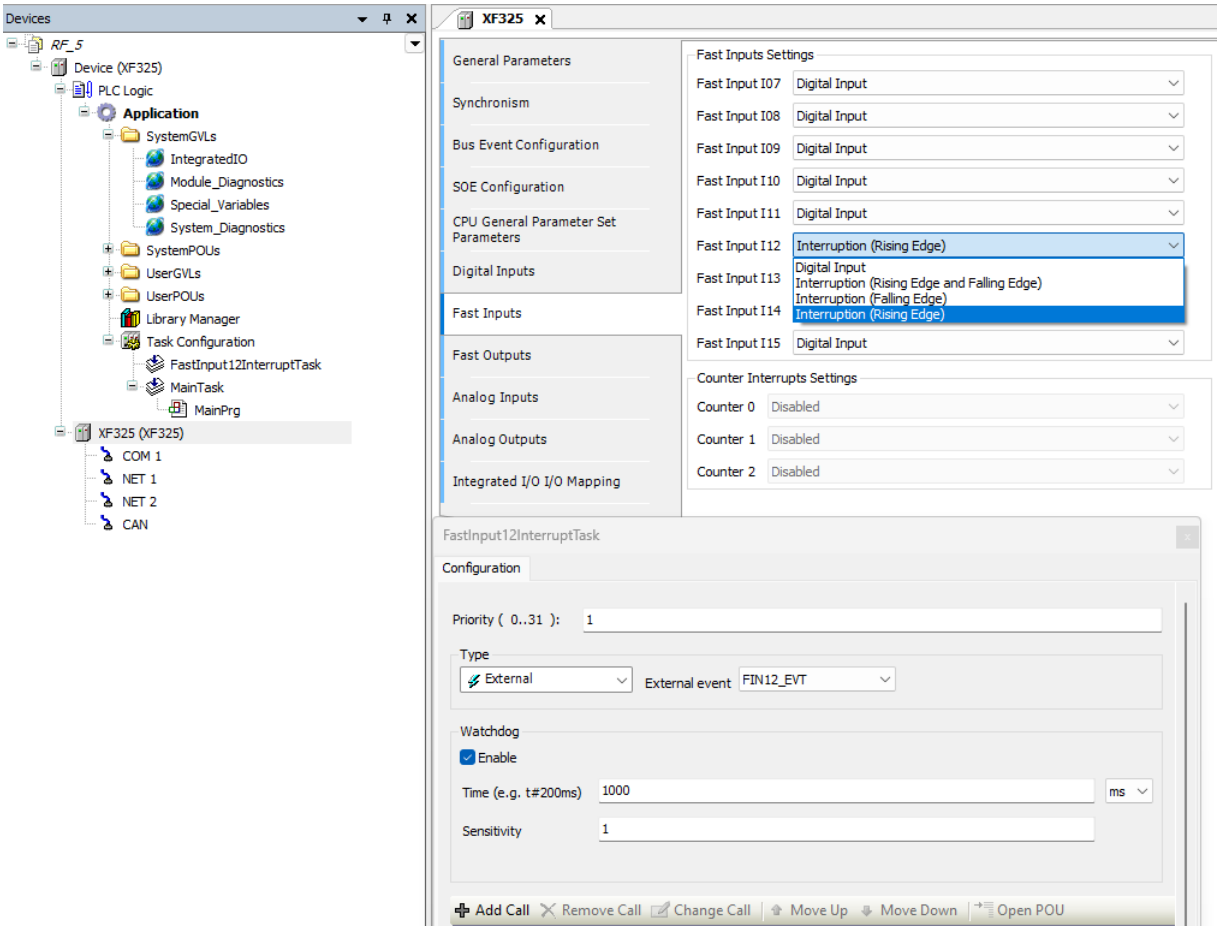


Figura 65: Configurações das Interrupções de Entradas Rápidas

ATENÇÃO

A entrada de interrupção externa possui um filtro de janela de tempo de 10 ms para proteger o controlador contra transições espúrias no sinal de entrada. Essa janela inicia imediatamente após a ocorrência da interrupção e, durante esse período, qualquer outro evento de interrupção externa será descartado.

ATENÇÃO

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

5.6.3. Saídas Rápidas

As saídas rápidas são sinais de saída especiais que podem ser usados para saídas de geradores de pulsos. Essas saídas físicas especiais podem ser atribuídas a dois tipos de elementos lógicos: VFO/PWM (frequência variável/largura de pulso) e PTO (saída de trem de pulsos). Cada um desses elementos lógicos consome um sinal de saída rápida. O número de saídas rápidas físicas, bem como o número máximo de elementos lógicos VFO/PWM e PTO atribuíveis a essas saídas, é descrito na seção Descrição Técnica.

A configuração das saídas rápidas está localizada na tela seguinte:

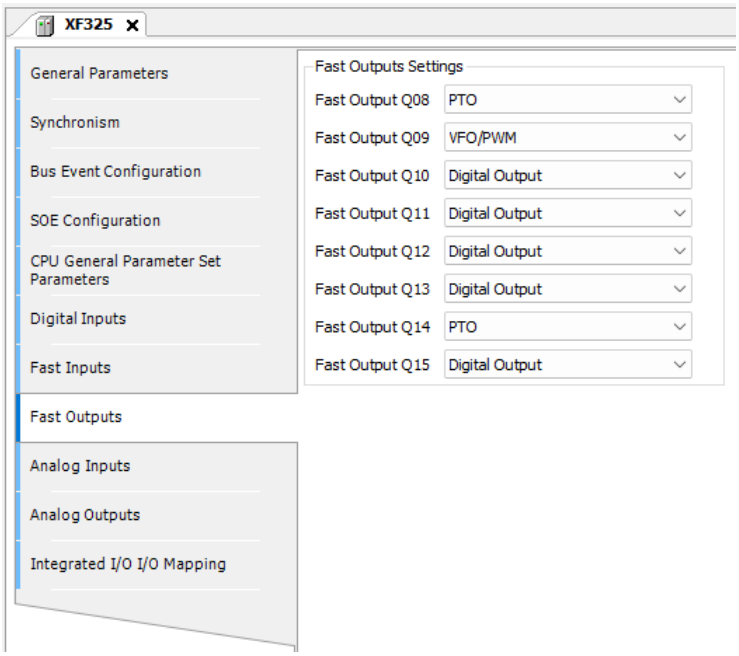


Figura 66: Parâmetro das Saídas Rápidas

A tabela abaixo mostra os possíveis valores de configuração:

Configuração	Descrição	Padrão	Opções
Fast Output Q08	Configuração da Saída Rápida Q08.	Saída Digital	Saída Digital VFO/PWM PTO
Fast Output Q09	Configuração da Saída Rápida Q09.	Saída Digital	Saída Digital VFO/PWM PTO
Fast Output Q10	Configuração da Saída Rápida Q10.	Saída Digital	Saída Digital VFO/PWM
Fast Output Q11	Configuração da Saída Rápida Q11.	Saída Digital	Saída Digital VFO/PWM
Fast Output Q12	Configuração da Saída Rápida Q12.	Saída Digital	Saída Digital VFO/PWM
Fast Output Q13	Configuração da Saída Rápida Q13.	Saída Digital	Saída Digital VFO/PWM
Fast Output Q14	Configuração da Saída Rápida Q14.	Saída Digital	Saída Digital VFO/PWM PTO
Fast Output Q15	Configuração da Saída Rápida Q15.	Saída Digital	Saída Digital VFO/PWM

Tabela 53: Parâmetros das Saídas Rápidas

Conforme mostrado na tabela anterior, as saídas rápidas podem ser configuradas como saídas digitais padrão. Nesse caso, seu valor digital pode ser definido usando a variável padrão mapeada em *IntegratedIo*.

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

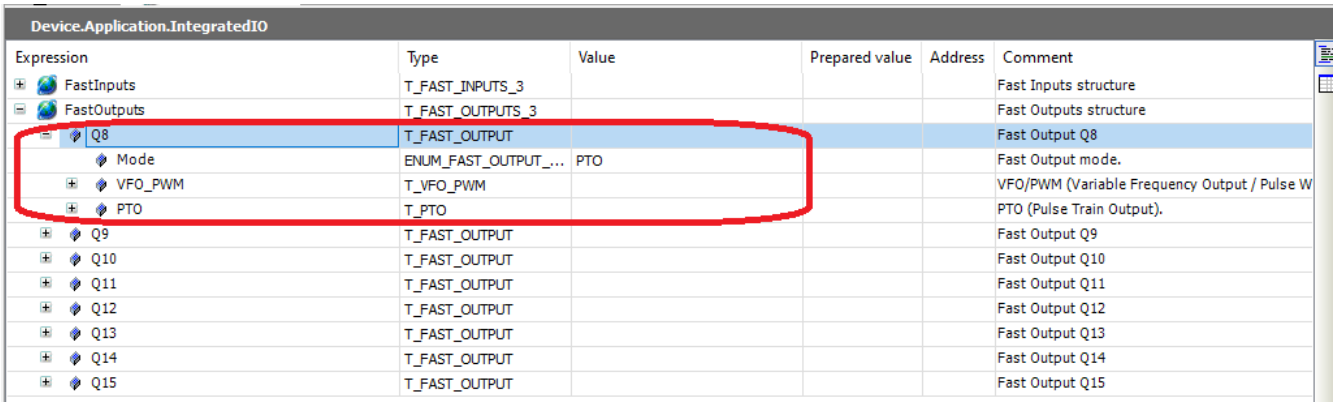


Figura 67: Estrutura das Saídas Rápidas

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

Variável	Descrição	Tipo	Valores Permitidos
Mode	Modo configurado da saída rápida (somente leitura)	ENUM_FAST_OUTPUT_MODE	DIGITAL_OUTPUT PWM PTO

Variável	Descrição	Tipo	Valores Permitidos
VFO_PWM	Estrutura VFO/PWM. Contém uma estrutura para controlar a saída rápida quando esta estiver configurada como VFO/PWM.	T_VFO_PWM	-
PTO	Estrutura PTO. Contém uma estrutura para controlar a saída rápida quando esta estiver configurada como PTO.	T_PTO	-

Tabela 54: Variáveis da estrutura de 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 (Variable Frequency Output / Pulse Width Modulator) é um modo de saída de gerador de pulsos onde a frequência e o ciclo de trabalho podem ser controlados pelo programa do usuário. É aplicável, por exemplo, para controlar a potência transferida para uma carga elétrica ou para controlar o ângulo de um servomotor. O princípio de funcionamento da saída VFO/PWM é muito simples, veja a forma de onda pulsada mostrada na figura abaixo:

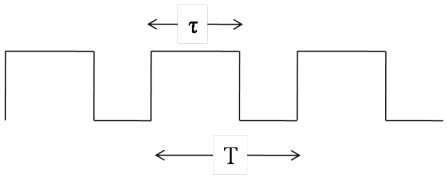


Figura 68: Forma de onda VFO/PWM

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 do 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 é a razão entre a largura do pulso e o período, então:

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

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

Variável	Descrição	Tipo	Valores permitidos
Frequency	Frequência em Hertz	UDINT	3 a 250000
DutyCycle	Ciclo de trabalho em porcentagem	USINT	0 a 100
Command	Estrutura de comandos VFO/PWM	T_VFO_PWM_COMMAND	-
Status	Estrutura de status VFO/PWM	T_VFO_PWM_STATUS	-

Tabela 55: Estrutura de variáveis VFO_PWM

A tabela abaixo mostra a estrutura dos comandos VFO_PWM.

Variável	Descrição	Tipo	Valores Permitidos
Habilitar	Habilitar saída VFO/PWM	BIT	FALSE ou TRUE

Tabela 56: Estrutura de comando VFO/PWM

A tabela abaixo mostra a estrutura de estado do VFO_PWM.

Variável	Descrição	Tipo	Valores Permitidos
InvalidFrequency	Valor de frequência inválido (fora do intervalo)	BIT	FALSE ou TRUE
InvalidDutyCycle	Valor do ciclo de trabalho inválido (fora do intervalo)	BIT	FALSE ou TRUE

Tabela 57: Estrutura de status VFO/PWM

Uma vez que o comando Enable seja TRUE, os parâmetros de entrada serão verificados continuamente e as variáveis de status serão atualizadas de acordo.

Além das variáveis globais do *IntegratedIo*, existe um bloco de função da biblioteca *LibIntegratedIo* que permite instanciar VFO/PWM em POUs escritas em linguagens gráficas (por exemplo, diagramas de lógica Ladder). Este bloco de função é, na verdade, um wrapper para as variáveis estruturadas descritas anteriormente. A figura abaixo mostra o bloco de função instanciado em um programa Ladder.

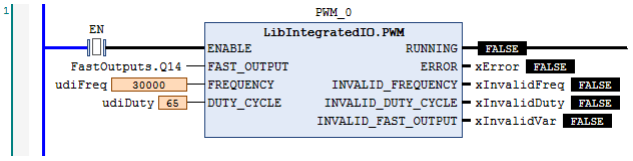


Figura 69: Bloco Funcional LibIntegratedIoPWM

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

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.Q8 FastOutputs.Q9 FastOutputs.Q10 FastOutputs.Q11 FastOutputs.Q12 FastOutputs.Q13 FastOutputs.Q14 FastOutputs.Q15
FREQUENCY	Frequência em Hertz.	UDINT	3 a 250000
DUTY_CYCLE	Duty Cycle em porcentagem.	USINT	0 a 100
RUNNING	VFO/PWM está sendo executado.	BOOL	FALSE ou TRUE
ERROR	Ocorreu um erro na execução do bloco funcional. As variáveis a seguir fornecem informações detalhadas.	BOOL	FALSE ou TRUE

Variável	Descrição	Tipo	Valores Permitidos
INVALID_FREQUENCY	O valor de frequência é inválido (fora do intervalo).	BOOL	FALSE or 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 atribuído ao bloco ou não está configurado como VFO/PWM.	BOOL	FALSE ou TRUE

Tabela 58: Descrição do bloco funcional LibIntegratedIo.PWM

5.6.3.2. PTO

O PTO (Pulse Train Output) é um modo de geração de pulsos. É usado, por exemplo, para controlar motores de passo responsáveis pelo posicionamento de mecanismos com inércia considerável. Nesses casos, a velocidade de rotação deve aumentar lentamente (aceleração) no início do movimento e diminuir lentamente (desaceleração) na parada. Essa aceleração e desaceleração são realizadas no trem de pulsos aumentando e diminuindo a frequência dos pulsos, mantendo um ciclo de trabalho de 50

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 e 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 pulsos mostrado é chamado de perfil trapezoidal, porque as rampas de aceleração e desaceleração produzem um formato trapezoidal.

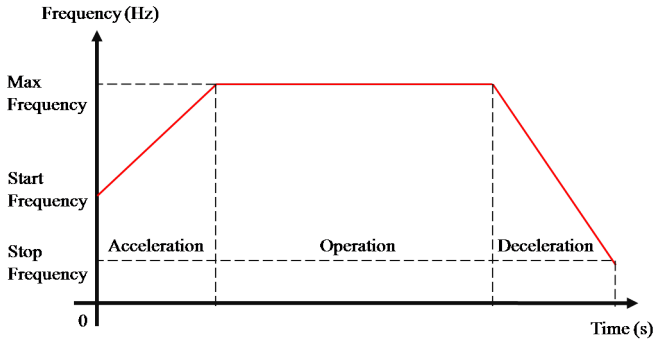


Figura 70: PTO com perfil trapezoidal

Para algumas aplicações, é mais recomendável usar o perfil “S”, cujas curvas de aceleração e desaceleração têm formato semelhante a um “S”. A figura abaixo ilustra esse perfil.

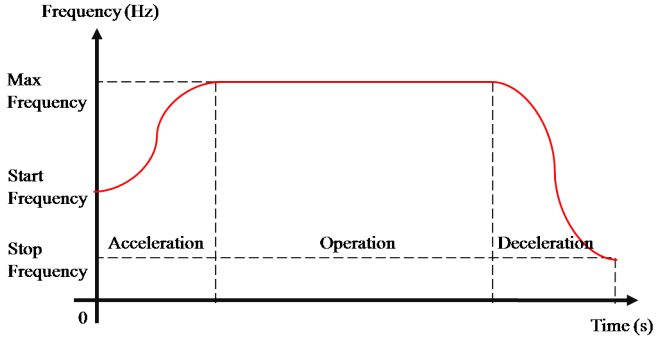


Figura 71: PTO com perfil “S”

Além dos parâmetros do PTO, existem informações de status e comandos que o programa do usuário pode utilizar para controlar a saída. Algumas informações de status importantes são o contador de pulsos (proporcional a uma posição), a etapa do trem de pulsos (aceleração, operação, desaceleração) e até mesmo se a saída está funcionando corretamente. Os comandos necessários para controlar a PTO são iniciar o trem de pulsos, parar o trem de pulsos e parar o trem de pulsos suavemente (parada suave). O comando de parada suave é muito importante, pois pode ser usado em situações de emergência onde o sistema não pode parar abruptamente. As figuras abaixo mostram como o comando de parada suave altera o trem de pulsos quando 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 esteja parando.

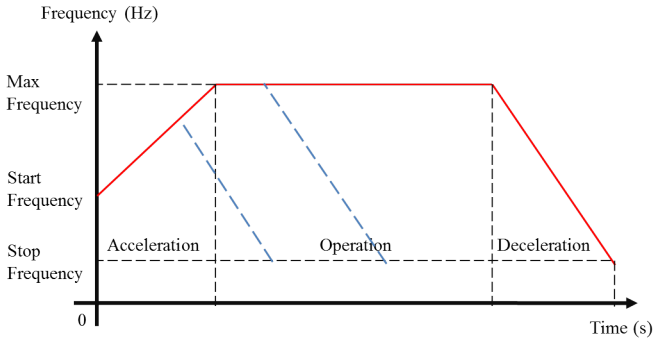


Figura 72: Parada suave da PTO no perfil trapezoidal

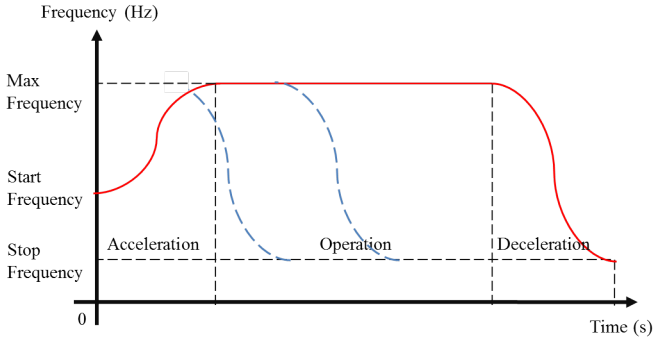


Figura 73: Parada suave da PTO no perfil “S”

Para controlar o 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	3 a 250000
StopFrequency	Frequência de parada em Hertz	UDINT	3 a 250000
MaxFrequency	Frequência máxima em Hertz	UDINT	3 a 250000
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 16777215
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 59: Estrutura de variáveis da PTO

A tabela abaixo mostra a estrutura de comandos do 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 60: Estrutura de comandos da PTO

A tabela abaixo mostra a estrutura de status do 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 (MaxFrequency)	BIT	FALSE ou TRUE
Done	Trem de pulsos já foi executado	BIT	FALSE ou TRUE
InvalidFrequency	A frequency (inicial, de parada ou máxima) é inválida	BIT	FALSE ou TRUE
InvalidPulses	Número de pulsos (TotalPulses, Acceleration or Deceleration) é inválido	BIT	FALSE ou TRUE

Tabela 61: Estrutura de estados da PTO

Assim que o comando Iniciar for TRUE, os parâmetros de entrada serão verificados continuamente e as variáveis de status serão atualizadas de acordo.

Além das variáveis globais do IntegratedIo, existe um bloco de função da biblioteca *LibIntegratedIo* que permite instanciar PTOs em POUs escritas em linguagens gráficas (por exemplo, Diagrama de Lógica Ladder). Este bloco de função é, na verdade, um encapsulamento das variáveis estruturadas descritas anteriormente. A figura abaixo mostra o bloco de função instanciado em um programa Ladder.

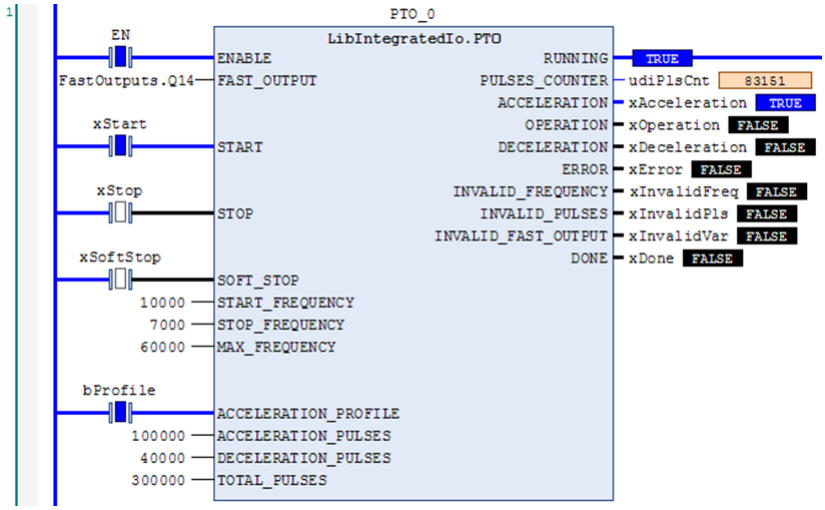


Figura 74: Bloco Funcional LibIntegratedIo.PTO

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

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.Q8 FastOutputs.Q9 FastOutputs.Q14
START	Inicia o trem de pulso quando este bit é setado (borda de subida)	BOOL	FALSE ou TRUE
STOP	Para o trem de pulso quando este bit é setado (borda de subida)	BOOL	FALSE ou TRUE
SOFT_STOP	Para suavemente o trem de pulso quando este bit é setado (borda de subida)	BOOL	FALSE ou TRUE
START_FREQUENCY	Frequência Inicial em Hertz	UDINT	3 a 250000
STOP_FREQUENCY	Frequência de parada em Hertz	UDINT	3 a 250000
MAX_FREQUENCY	Frequência Máxima em Hertz	UDINT	3 a 250000
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 16777215
RUNNING	Trem de pulsos está sendo executado	BOOL	FALSE ou TRUE

Variável	Descrição	Tipo	Valores Permitidos
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
DECELERATION	Etapa de desaceleração (da frequência máxima à frequência 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 inválida	BOOL	FALSE ou TRUE
INVALID_FAST_OUTPUT	FAST_OUTPUT não foi atribuída ao boco ou não está configurada como PTO	BOOL	FALSE ou TRUE
DONE	Trem de pulso já foi executado	BOOL	FALSE ou TRUE

Tabela 62: Descrição do bloco funcional LibIntegratedIo.PTO

5.6.4. Entradas Analógicas

Os parâmetros relacionados às entradas analógicas são mostrados abaixo:

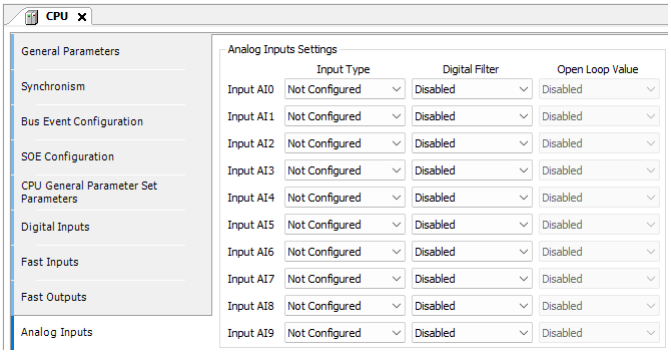


Figura 75: Parâmetros das Entradas Analógicas

A tabela abaixo mostra os possíveis valores de configuração:

Configuração	Descrição	Padrão	Opções
Tipo da Entrada	Seleciona o tipo de entrada (AI0 a AI5 somente corrente) (AI6 a AI9 somente tensão)	Não configurada	Não configurada Tensão 0 - 10 Vdc (AI6 a AI9) Corrente 0 - 20 mA (AI0 a AI5) Corrente 4 - 20 mA (AI0 a AI5)
Filtro Digital	Habilita/Desabilita (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 - 20 mA)	Desabilitado	Desabilitado 0 30000

Tabela 63: Parâmetros das Entradas Analógicas

Notas:

Tipo de entrada: Certifique-se de usar o pino correto no conector correspondente ao tipo selecionado (tensão ou corrente).

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

5.6.5. Saídas Analógicas

Os parâmetros relacionados às saídas analógicas são mostrados abaixo:

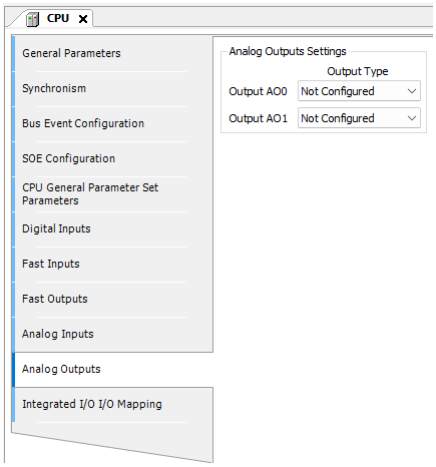


Figura 76: Parâmetro das Saídas Analógicas

A tabela abaixo mostra os possíveis valores de configuração:

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

Tabela 64: Parâmetros das Saídas Analógicas

5.6.6. Mapeamento de E/S

Na aba *Mapeamento de E/S*, é possível configurar o nome e a descrição de cada variável de entrada e saída.

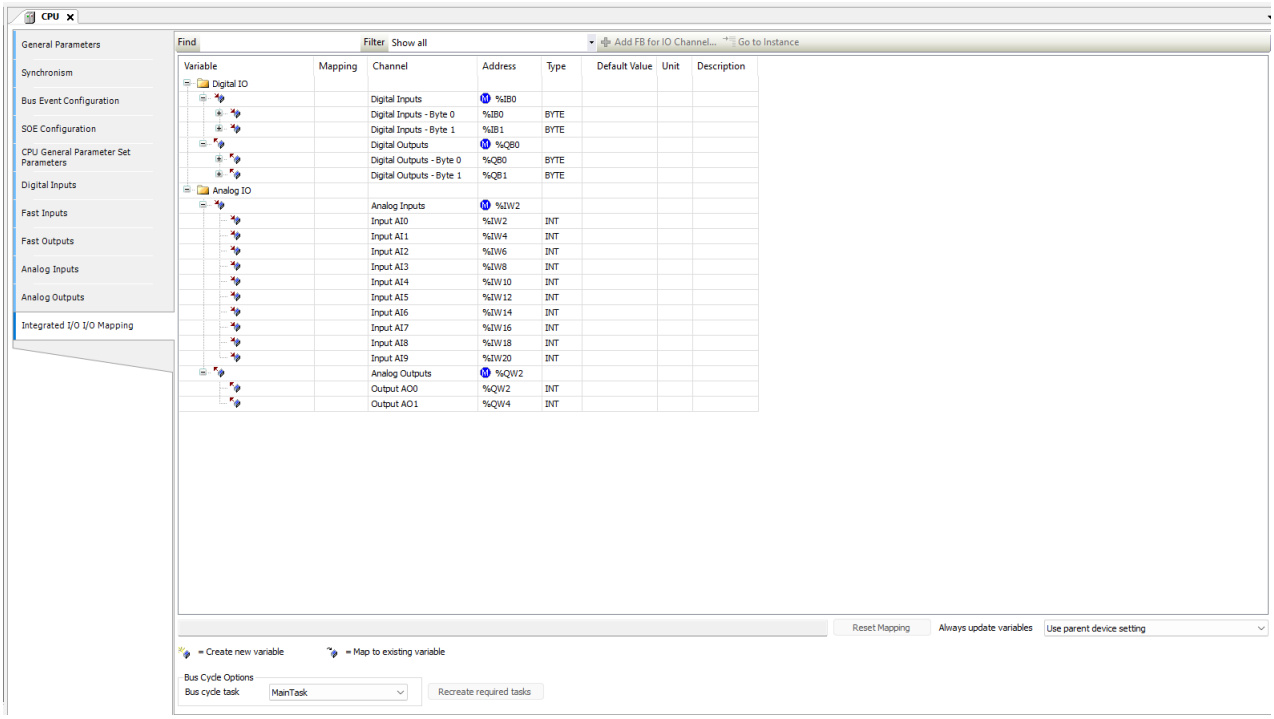


Figura 77: Mapeamento de tags de E/S

5.7. Protocolos de Comunicação

Independente dos protocolos utilizados em cada aplicação, os controladores do Sistema Nexto possuem alguns limites máximos de utilização para cada modelo. 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:

	XF300	XF300-B	XF315	XF325	XF325-W
Pontos Mapeados	20480				
Mapeamentos (Por Instância / Total)	5120 / 20480				
Requisições	512				
NETs – Instâncias Servidor ou Cliente (Total)	8				
COM (n) – Instâncias Mestre ou Escravo	1				
Control Centers	-				

Tabela 65: Protocol Limits per CPU

Notas:

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

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

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

NETs – Instâncias de Clientes ou Servidores: Este campo define o número máximo de instâncias de protocolos total distribuído entre todas as interfaces Ethernet do sistema.

COM (n) – Instâncias de Mestre ou Escravo: Devido às suas características, cada interface serial suporta apenas uma instância de protocolo de comunicação. Exemplos de instâncias compatíveis com interfaces seriais: MODBUS RTU Mestre e MODBUS RTU Escravo.

As limitações do protocolo MODBUS por mapeamentos simbólicos podem ser visualizadas na tabela abaixo. The limitations of the MODBUS protocol for symbolic mappings can be seen in the table below.

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 66: 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.

5.7.1. Comportamento dos Protocolos x Estados da UCP

A tabela abaixo mostra detalhadamente cada comportamento assumido para cada protocolo de comunicação configurável nos controladores do Sistema Nexto, em todos seus estados de funcionamento.

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

Tabela 67: Comportamento dos Protocolos x Estados da UCP

Notas:

Símbolo ✓: O protocolo permanece ativo e operando normalmente.

Símbolo ✗: O protocolo está desabilitado.

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

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

5.7.2. MODBUS Mestre RTU

Este protocolo está disponível para as UCPs do Sistema 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*).

A configuração disponível utiliza Mapeamento Simbólico, onde suas variáveis são definidas por seus nomes.

O procedimento para inserir uma instância de protocolo encontra-se detalhado no Manual do Usuário do MasterTool - MU299609 ou na seção Inserindo uma instância de Protocolo. As etapas de configuração restantes são descritas abaixo para cada modo.

- Adicionar a instância do protocolo MODBUS RTU Mestre ao canal serial. Para realizar esse procedimento, consultar a seção Inserindo uma instância de Protocolo.
- Configurar a interface serial, escolhendo a taxa de transmissão, o comportamento dos sinais de modem RTS/CTS, a paridade, bits de parada do canal e demais configurações através de um duplo clique sobre o canal serial COM 1 (ou COM 2 - se disponível). Consultar a seção Configuração da Interface Serial.

5.7.2.1. 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 78.
- Adicionar e configurar dispositivos através da aba Parâmetros Gerais, definindo endereço do escravo, time-out de comunicação e número de retentativas de comunicação como pode ser visto na Figura 79.
- Adicionar e configurar os mapeamentos MODBUS na aba Mapeamentos conforme Figura 80, 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 81, 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.

5.7.2.1.1. Parâmentros 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:

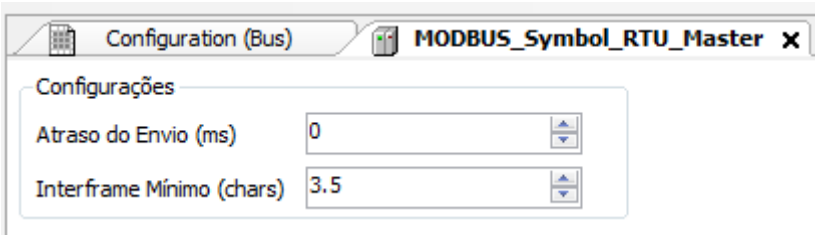


Figura 78: Tela de Configuração Parâmetros Gerais MODBUS RTU Mestre

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 68: 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 configurados são armazenados nas variáveis *T_DIAG_MODBUS_RTU_MASTER* que são descritas 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.
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).

T_DIAG_MODBUS_RTU_MASTER_1.*	Tamanho	Descrição
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 69: 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.

5.7.2.1.2. Configuração dos Dispositivos – Configuração por Mapeamento Simbólico

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

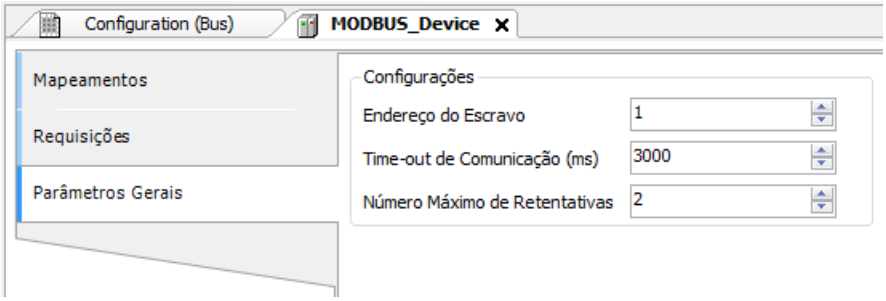


Figura 79: Tela de Configurações dos Parâmetros Gerais do Dispositivo

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 70: 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.

5.7.2.1.3. *Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico*

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

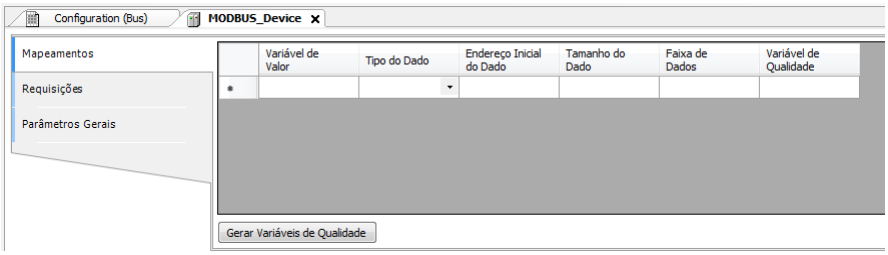


Figura 80: Tela de Mapeamentos de Dados MODBUS

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

Tabela 71: 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 72: 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.

5.7.2.1.4. Configuração das Requisições – Configuração por Mapeamento Simbólico

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

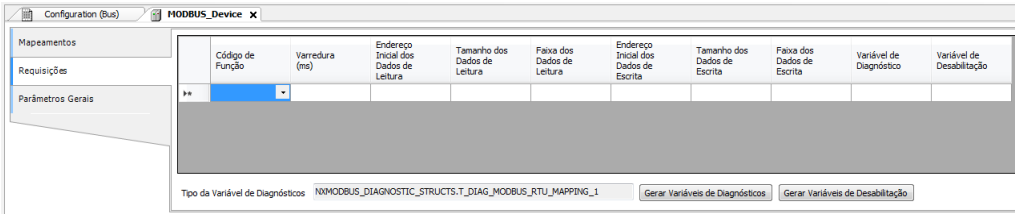


Figura 81: Tela de Requisições de dados MODBUS Mestre

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

Configuração	Descrição	Padrão	Opções
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 73: 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 74: Funções MODBUS Suportadas

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 por mapeamento simbólico 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.

T_DIAG_MODBUS_RTU_MAPPING_1.*	Tamanho	Descrição
Bits de estado da comunicação:		
byStatus.bCommIdle	BIT	Comunicação inativa (aguardando ser constada).
byStatus.bCommExecuting	BIT	Comunicação ativa.
byStatus.bCommPostponed	BIT	Comunicação adiada, pois o número máximo de requisições simultâneas foi atingido. As comunicações adiadas serão executadas na mesma sequência em que elas foram requisitadas, evitando a indeterminação. O tempo gasto nesse estado não é contabilizado para efeitos de time-out. Os bits bCommIdle e bCommExecuting são falsos quando o bit bCommPostponed é verdadeiro.
byStatus.bCommDisabled	BIT	Comunicação desabilitada. O bit bCommIdle é reiniciado nessa condição.
byStatus.bCommOk	BIT	Comunicação finalizada anteriormente foi realizada com sucesso.
byStatus.bCommError	BIT	Comunicação finalizada anteriormente teve um erro. Verificar código de erro.
Último código de erro (habilitado quando o bCommError = verdadeiro):		
eLastErrorCode	MASTER_ERROR_CODE (BYTE)	Informa a possível causa do último erro ocorrido na relação MODBUS. Consulte a Tabela 76 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 75: Diagnósticos das relações MODBUS RTU

Notas:

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

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

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

Código	Enumerável	Descrição
1	ERR_EXCEPTION	Resposta reportada em um código de exceção (ver eLastExceptionCode = Código de Exceção).
2	ERR_CRC	Resposta com CRC inválido.
3	ERR_ADDRESS	Endereço MODBUS não encontrado. O endereço que respondeu à requisição foi diferente do esperado.
4	ERR_FUNCTION	Código inválido da função. A função recebida na resposta foi diferente da esperada pela requisição.
5	ERR_FRAME_DATA_COUNT	A quantidade de dados da resposta foi diferente da esperada.
7	ERR_NOT_ECHO	Resposta não é eco da pergunta (FC 5 e 6).
8	ERR_REFERENCE_NUMBER	Número de referência inválido (FC 15 e 16).
9	ERR_INVALID_FRAME_SIZE	Resposta menor do que a esperada.
20	ERR_CONNECTION	Erro durante a fase de conexão.
21	ERR_SEND	Erro durante a fase de transmissão.
22	ERR_RECEIVE	Erro durante a fase de recepção.
40	ERR_CONNECTION_TIMEOUT	Time-out no nível de aplicação durante a conexão.
41	ERR_SEND_TIMEOUT	Time-out no nível de aplicação durante a transmissão.
42	ERR_RECEIVE_TIMEOUT	Time-out no nível de aplicação enquanto aguarda resposta.
43	ERR_CTS_OFF_TIMEOUT	Time-out enquanto aguarda CTS = falso na transmissão.
44	ERR_CTS_ON_TIMEOUT	Time-out enquanto aguarda CTS = verdadeiro na transmissão.
128	NO_ERROR	Sem erro desde a inicialização.

Tabela 76: 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.

5.7.3. MODBUS Escravo RTU

Este protocolo está disponível para as UCPs do Sistema 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 está disponível apenas no modo de execução (*Run* Mode).

O procedimento para inserir uma instância do protocolo encontra-se detalhado no manual do Mastertool. As etapas de configuração restantes são descritas abaixo para cada modo:

- Adicionar a instância do protocolo MODBUS RTU Escravo ao canal serial. Para realizar esse procedimento, consultar a seção Inserindo uma instância de Protocolo.
- Configurar a interface serial, escolhendo a velocidade de comunicação, o comportamento dos sinais RTS/CTS, a paridade, os bits de parada do canal, entre outros. Consultar a seção Configuração da Interface Serial.

5.7.3.1. 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.

5.7.3.1.1. MODBUS Slave Protocol General Parameters – Configuração por Mapeamento Simbólico

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

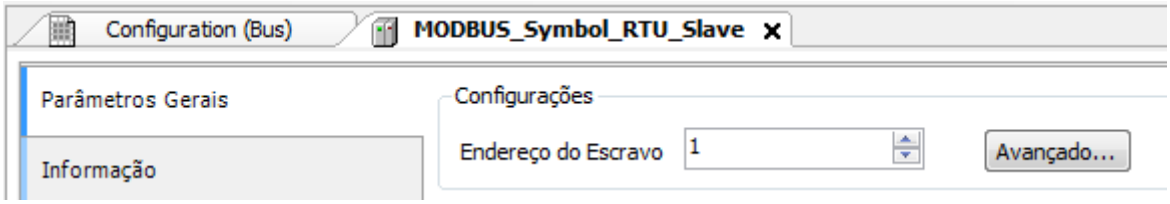


Figura 82: Configuração do Escravo

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

Tabela 77: 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.

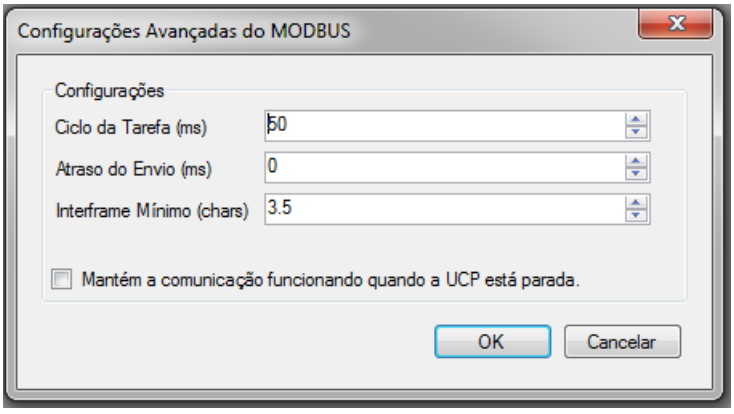


Figura 83: Configurações Avançadas MODBUS do Escravo

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

Configuração	Descrição	Padrão	Opções
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 o controlador estiver em STOP ou parada em um breakpoint	desmarcado	marcado ou desmarcado

Tabela 78: 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 do controlador 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 configurados por mapeamento simbólico são armazenados em variáveis do tipo *T_DIAG_MODBUS_RTU_SLAVE_1*, descritas 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

T_DIAG_MODBUS_RTU_SLAVE_1.*	Tamanho	Descrição
		31: erro de time-out na recepção 32: controle de fluxo configurado diferente de manual 33: controle de fluxo inválido para a porta serial configurada 34: recepção de dados não permitida no modo normal 35: recepção de dados não permitida no modo estendido 36: interrupção DCD não permitida 37: interrupção CTS não permitida 38: interrupção DSR não permitida 39: porta serial não configurada 50: erro interno na porta serial
Bits de comando, reiniciados automaticamente:		
tCommand.bStop	BIT	Parar o escravo.
tCommand.bRestart	BIT	Reiniciar o escravo.
tCommand.bResetCounter	BIT	Reiniciar as estatísticas dos diagnósticos (contadores).
Estatísticas de comunicação:		
tStat.wRXRequests	WORD	Contador de requisições normais recebidas pelo escravo e respondidas normalmente. No caso de um comando broadcast, esse contador é incrementado, porém não é transmitida a resposta (0 a 65535).
tStat.wTXExceptionResponses	WORD	Contador de requisições normais recebidas pelo escravo e respondidas com códigos de exceção. No caso de um comando broadcast, esse contador é incrementado, porém não é transmitida a resposta (0 a 65535).
tStat.wRXFrames	WORD	Contador de frames recebidos pelo escravo. Considera-se frame, algo que é processado e seguido de um período mínimo de silêncio interframes (Interframe Mínimo), ou seja, uma mensagem ilegal também é computada (0 a 65535).
tStat.wRXIllegalRequests	WORD	Contador de requisições ilegais. Estes são frames que iniciam com o endereço 0 (broadcast) ou com o endereço MODBUS do escravo, mas não são requisições legais – sintaxe inválida, frames menores, CRC inválido – (0 a 65535).
tStat.wRXOverrunErrors	WORD	Contador de frames com erros de overrun durante a recepção – UART FIFO ou fila RX – (0 a 65535).
tStat.wRXIncompleteFrames	WORD	Contador de frames com erro de construção, paridade ou falha durante a recepção (0 a 65535).
tStat.wCTSTimeOutErrors	WORD	Contador de erro de time-out no CTS, utilizando o handshake RTS/CTS, durante a transmissão (0 a 65535).

Tabela 79: 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.

5.7.3.1.2. Configuração das Requisições – Configuração por Mapeamento Simbólico

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

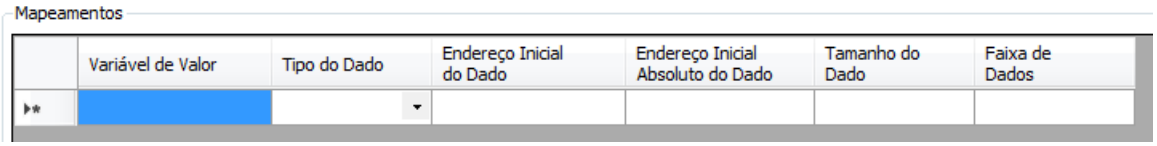


Figura 84: Tela de Mapeamentos de dados MODBUS

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 80: Configuração dos Mapeamentos MODBUS

Notas:

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

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

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

Tabela 81: Tipos de Dados MODBUS Suportados

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.

5.7.4. MODBUS Ethernet

A rede de comunicação multimestre permite que os controladores do Sistema Nexto leiam ou escrevam variáveis MODBUS em outros controladores ou IHMs compatíveis com os protocolos MODBUS TCP ou MODBUS RTU via TCP. O controlador

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 65.

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

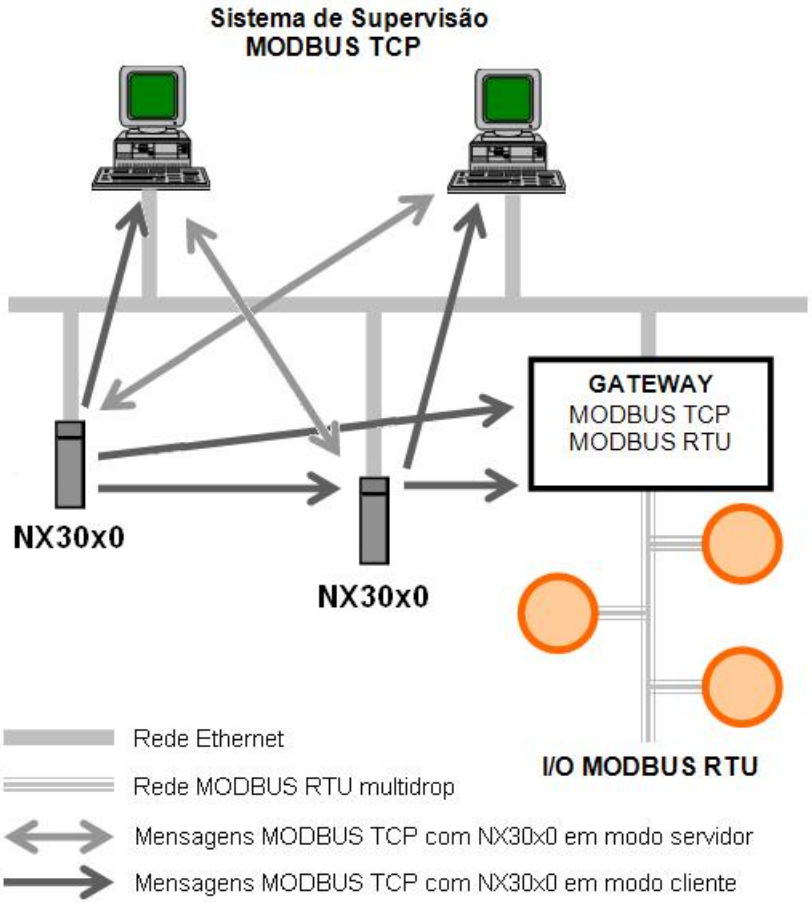


Figura 85: Rede de Comunicação MODBUS TCP

A associação de variáveis MODBUS com variáveis simbólicas da aplicação 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.

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

Tabela 82: Tipos de Dados MODBUS Suportados

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 83: Funções MODBUS Suportadas

Independente do modo de configuração, os passos para inserir uma instância do protocolo e configurar a interface ethernet são iguais. As demais etapas de configuração serão descritas a seguir para cada modalidade.

- Adicionar uma ou mais instâncias do protocolo MODBUS Ethernet Cliente ou Servidor ao canal Ethernet. Para realizar esse procedimento, consultar a seção Inserindo uma instância de Protocolo.
- Configurar a interface Ethernet. Para realizar esse procedimento, consultar a seção Configuração das Interfaces Ethernet.

5.7.5. MODBUS Ethernet Cliente

Este protocolo está disponível para as UCPs do Sistema 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.

5.7.5.1. 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.

5.7.5.1.1. *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:

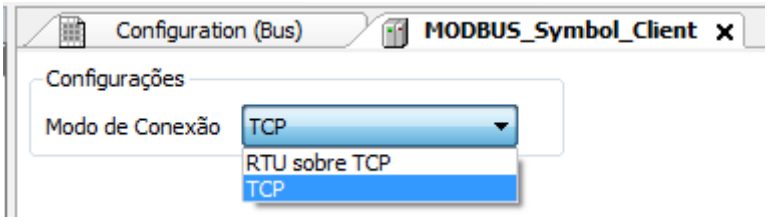


Figura 86: Tela de Configuração Parâmetros Gerais MODBUS Cliente

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

Tabela 84: Configurações Gerais MODBUS Cliente

Os diagnósticos e comandos do protocolo MODBUS Cliente configurado por mapeamento simbólico são armazenados em variáveis do tipo *T_DIAG_MODBUS_ETH_CLIENT_1*, 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 bInterruptedBy-Command).
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 85: Diagnósticos MODBUS Cliente

Nota:

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

5.7.5.1.2. Configuração dos Dispositivos – Configuração por Mapeamento Simbólico

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

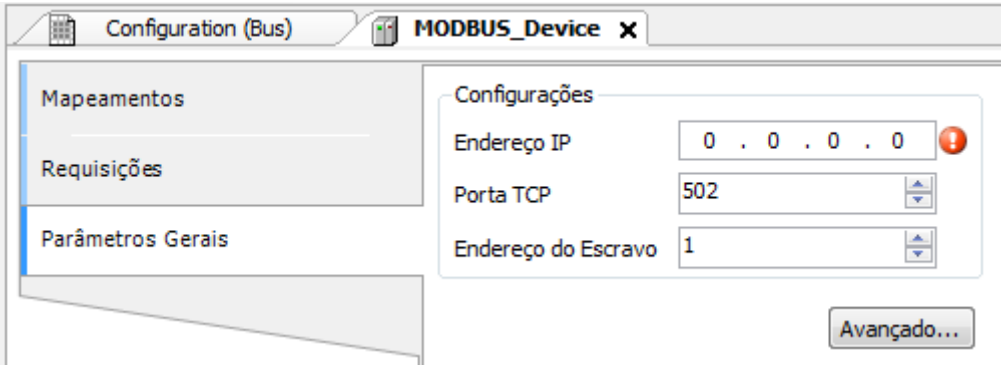


Figura 87: Tela de Configurações dos Parâmetros Gerais do Dispositivo

Configuração	Descrição	Padrão	Opções
Endereço IP	Endereço IP do servidor	0.0.0.0	1.0.0.1 a 223.255.255.255
Porta TCP	Porta TCP	502	2 a 65534
Endereço do Escravo	Endereço do Escravo MODBUS	1	0 a 255

Tabela 86: 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

Configuração	Descrição	Padrão	Opções
Modo	Define quando a conexão com o servidor é finalizada pelo cliente	Conexão permanece aberta enquanto não houver um time-out.	Conexão permanece aberta enquanto não houver um time-out. Conexão é fechada ao final de cada comunicação. Conexão é fechada após um tempo de inatividade de (s): 10 a 3600.
Tempo de Inatividade	Tempo de inatividade	10	10 a 3600

Tabela 87: 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 time-out:** 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.

5.7.5.1.3. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico

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

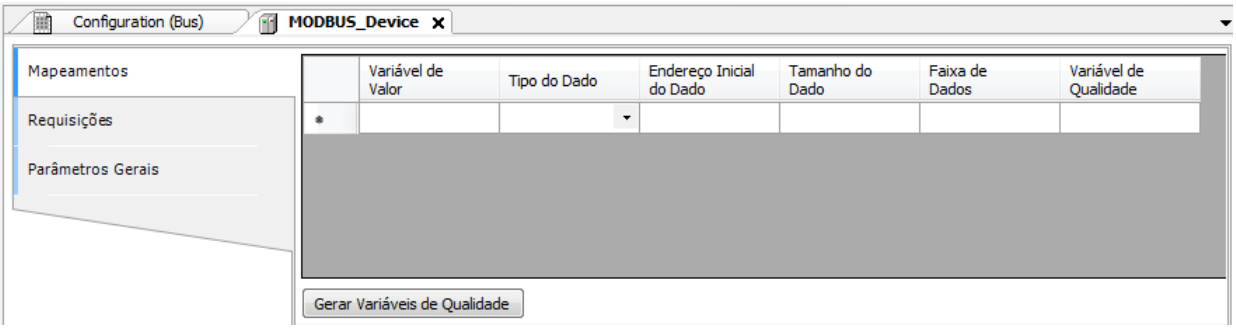


Figura 88: Tela de Mapeamentos de dados MODBUS Cliente

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 88: 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 89: 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:

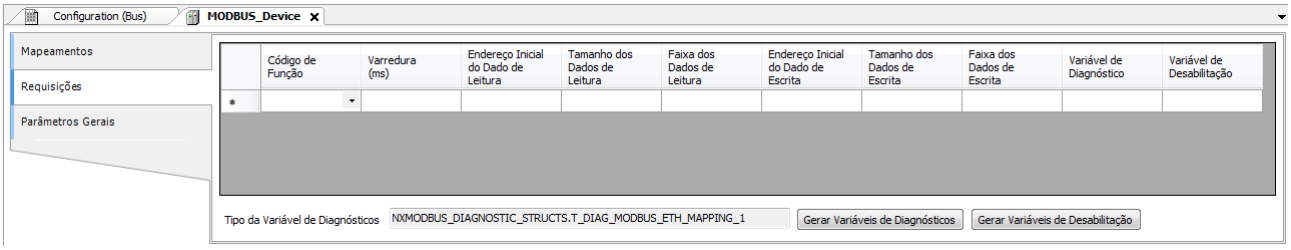


Figura 89: Tela de Requisições de dados MODBUS

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 90: 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 91: Funções MODBUS Suportadas

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 por mapeamento simbólico 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.

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 93 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 92: 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.

Código	Enumerável	Descrição
3	ERR_ADDRESS	Endereço MODBUS não encontrado. O endereço que respondeu à requisição foi diferente do esperado.
4	ERR_FUNCTION	Código inválido da função. A função recebida na resposta foi diferente da esperada pela requisição.
5	ERR_FRAME_DATA_COUNT	A quantidade de dados da resposta foi diferente da esperada.
7	ERR_NOT_ECHO	Resposta não é eco da pergunta (FC 5 e 6).
8	ERR_REFERENCE_NUMBER	Número de referência inválido (FC 15 e 16).
9	ERR_INVALID_FRAME_SIZE	Resposta menor do que a esperada.
20	ERR_CONNECTION	Erro durante a fase de conexão.
21	ERR_SEND	Erro durante a fase de transmissão.
22	ERR_RECEIVE	Erro durante a fase de recepção.
40	ERR_CONNECTION_TIMEOUT	Time-out no nível de aplicação durante a conexão.
41	ERR_SEND_TIMEOUT	Time-out no nível de aplicação durante a transmissão.
42	ERR_RECEIVE_TIMEOUT	Time-out no nível de aplicação enquanto aguarda resposta.
43	ERR_CTS_OFF_TIMEOUT	Time-out enquanto aguarda CTS = falso na transmissão.
44	ERR_CTS_ON_TIMEOUT	Time-out enquanto aguarda CTS = verdadeiro na transmissão.
128	NO_ERROR	Sem erro desde a inicialização.

Tabela 93: 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.

5.7.5.2. 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.

5.7.6. MODBUS Ethernet Servidor

Este protocolo está disponível para as UCPs do Sistema 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.

5.7.6.1. 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.

5.7.6.1.1. *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.

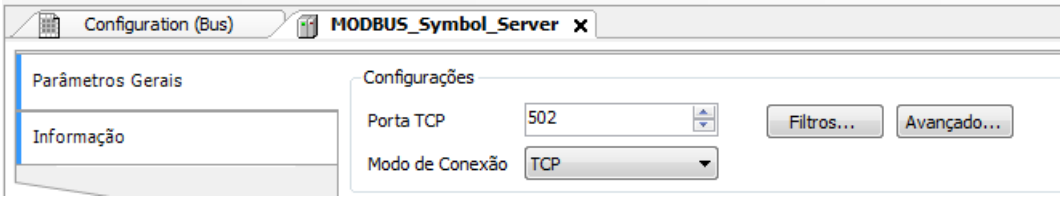


Figura 90: Tela de Configuração Parâmetros Gerais MODBUS Servidor

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

Tabela 94: 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 95: 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*.

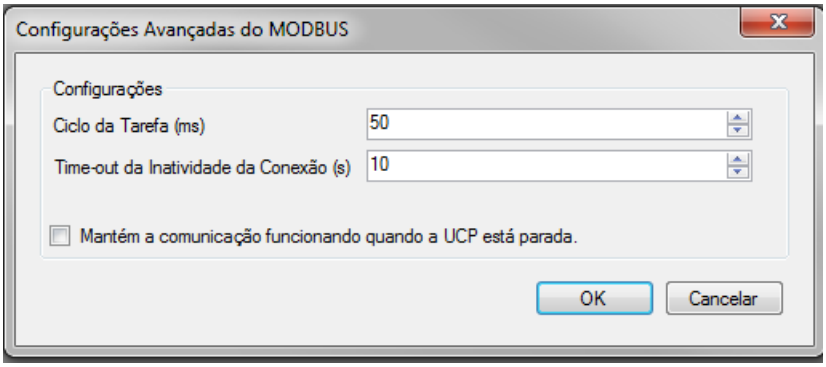


Figura 91: Janela de Configurações Avançadas do MODBUS Servidor

Configuração	Descrição	Padrão	Opções
Ciclo da Tarefa (ms)	Tempo para execução da instância dentro do ciclo, sem considerar o tempo de execução da mesma	50	5 a 100
Time-out da Inatividade da Conexão (s)	Tempo máximo de inatividade entre cliente e servidor antes da conexão ser fechada pelo servidor	10	1 a 3600
Mantém a comunicação funcionando quando a UCP está parada.	Habilita o MODBUS Symbol Server para rodar enquanto a UCP estiver em STOP ou após um breakpoint	desmarcado	marcado ou desmarcado

Tabela 96: Configurações Avançadas MODBUS Servidor

Notas:

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

Time-out da Inatividade da Conexão: Esse parâmetro foi criado para evitar que a quantidade máxima de conexões TCP seja atingida, imaginando que conexões inativas permanecessem abertas pelos mais diversos problemas. Enfim, indica por quanto tempo uma conexão (cliente ou servidora) pode permanecer aberta sem ser utilizada, ou seja, sem trocar mensagens de comunicação. Se o tempo especificado for atingido, a conexão simplesmente é fechada, liberando uma entrada na tabela de conexões.

5.7.6.1.2. Diagnósticos MODBUS Servidor – Configuração por Mapeamento Simbólico

Os diagnósticos e comandos do protocolo MODBUS Servidor configurado por mapeamento simbólico são armazenados em variáveis do tipo *T_DIAG_MODBUS_ETH_SERVER_1*, 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).

T_DIAG_MODBUS_ETH_SERVER_1.*	Tamanho	Descrição
tDiag.bInterruptedByCommand	BIT	O bit bNotRunning foi habilitado, pois o servidor foi interrompido pelo usuário através de bits de comando.
tDiag.bConfigFailure	BIT	Falha na configuração.
tDiag.bModuleFailure	BIT	Indica se há falha no módulo ou o módulo não está presente.
Bits de comando, reiniciados automaticamente:		
tCommand.bStop	BIT	Parar o servidor.
tCommand.bRestart	BIT	Reiniciar o servidor.
tCommand.bResetCounter	BIT	Reiniciar as estatísticas dos diagnósticos (contadores).
Estatísticas de comunicação:		
tStat.wActiveConnections	WORD	Número de conexões estabelecidas entre cliente e servidor (0 a 64).
tStat.wTimeoutClosedConnections	WORD	Contador de conexões, entre cliente e servidor, interrompidas depois de um período de inatividade – 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 97: 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.

5.7.6.1.3. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico

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

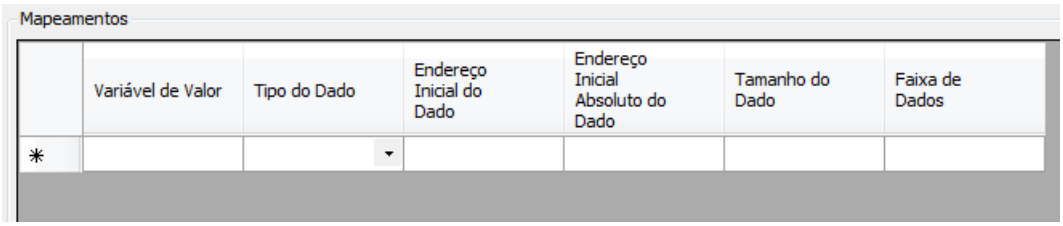


Figura 92: Tela de Mapeamentos de dados MODBUS Servidor

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

Tabela 98: 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.

5.7.7. OPC DA Servidor

Para comunicar com os controladores do Sistema 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).

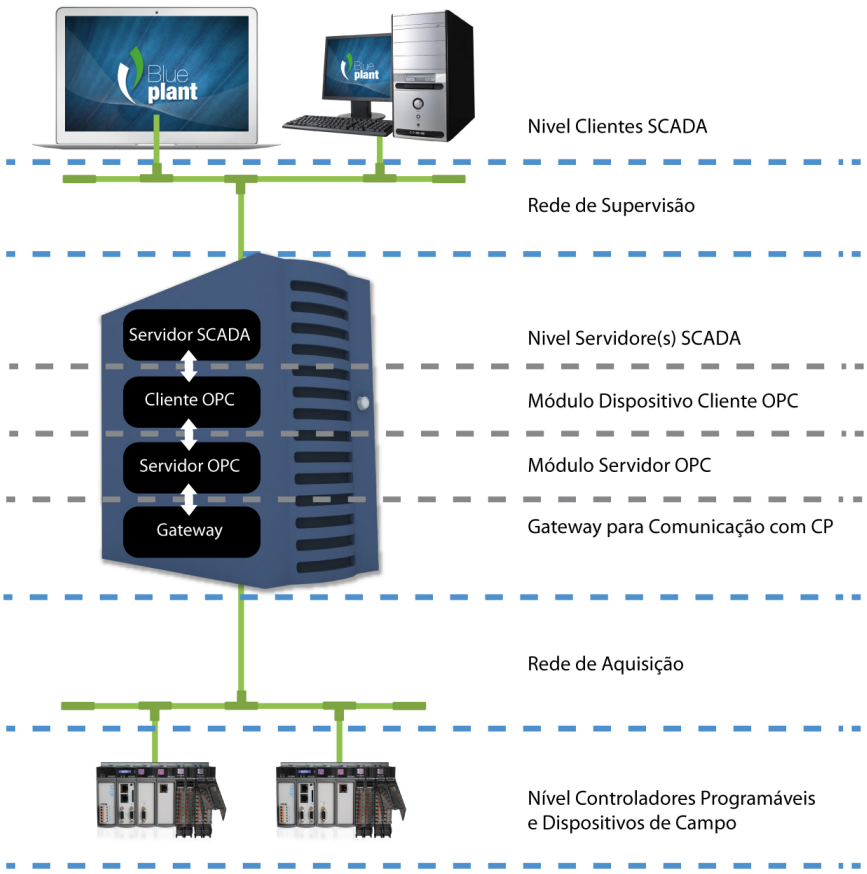


Figura 93: Arquitetura OPC DA

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 do Sistema 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 99: 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. Os módulos de expansão Ethernet não suportam essa funcionalidade.

5.7.7.1. Criando um Projeto para comunicação OPC DA

Diferente das comunicações com drivers como MODBUS, 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 POUs 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.

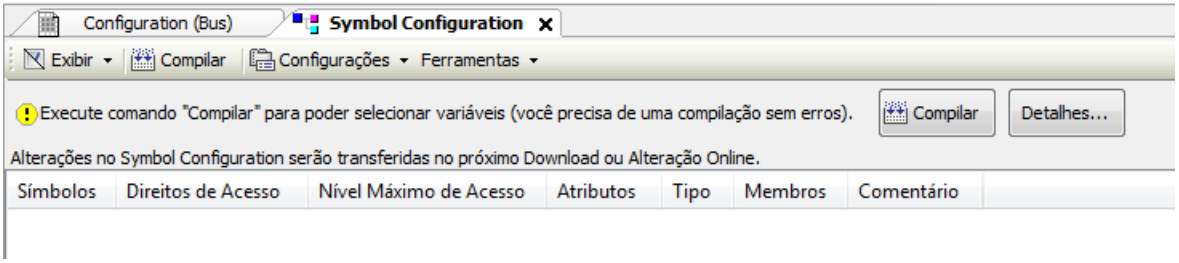


Figura 94: Objeto Symbol Configuration

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 *attribute 'symbol'*
Atributos	Indica se está sendo utilizado *attribute 'symbol'* 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: attribute 'symbol' := 'read' a coluna exibe attribute 'symbol' := 'write' a coluna exibe attribute 'symbol' := 'readwrite' 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 100: 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 94 desapareça. Após isso todas as variáveis disponíveis no projeto, sejam declaradas em POUs, GVLs e diagnósticos, serão exibidas e podem ser selecionadas. As variáveis selecionadas estarão disponíveis no Servidor OPC DA para acesso pelos Clientes.

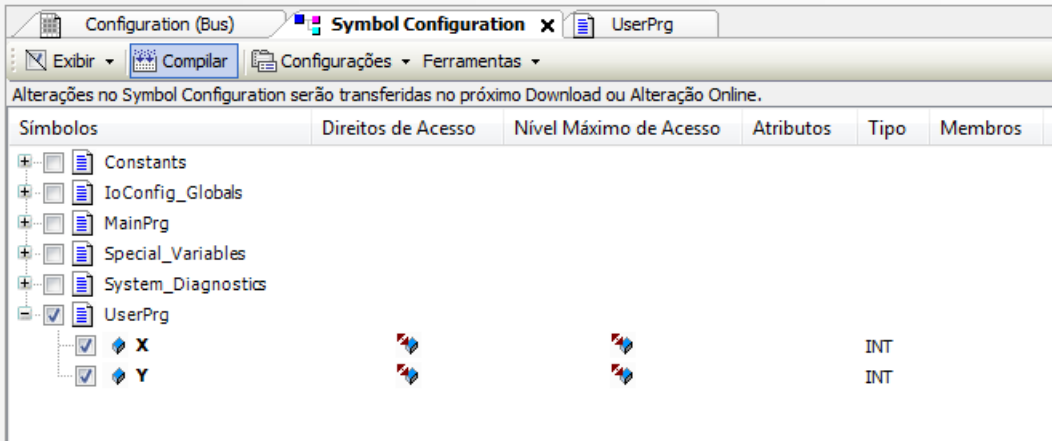


Figura 95: Selecionando Variáveis na Symbol Configuration

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, POUs 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 POUs 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.

5.7.7.2. 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.

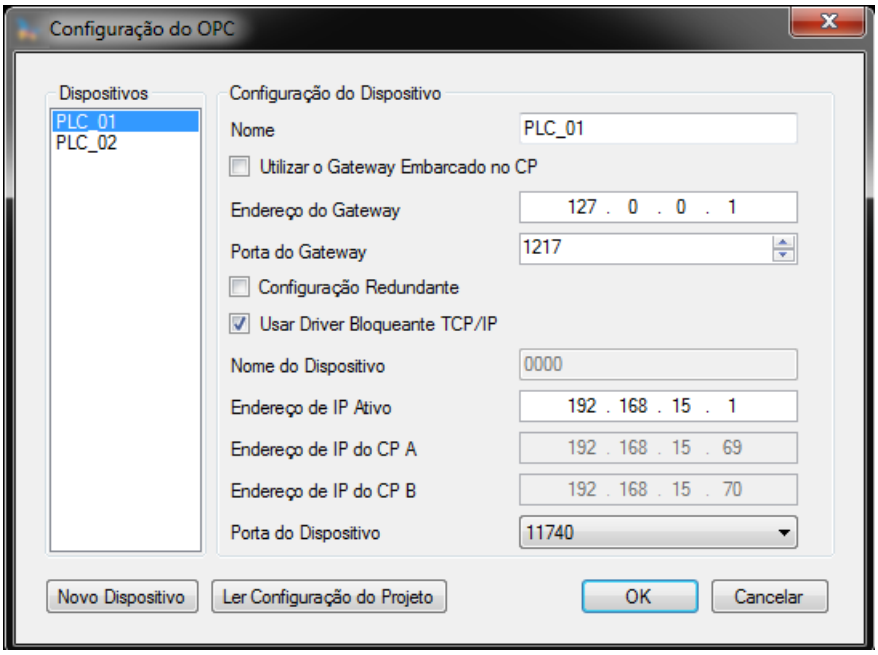


Figura 96: Configuração do OPC DA Servidor

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 *Configurações de Comunicação* do *Device*. O nome é a STRING antes do valor em hexadecimal que esta entre *[ ]*. Somente habilitado quando o checkbox *Usar Driver Bloqueante TCP/IP* 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 *Configurações de Comunicação* do Device. Permite até 49 caracteres.
Endereço de IP Ativo	Endereço IP do CP. Somente habilitado quando o checkbox *Usar Driver Bloqueante TCP/IP* está selecionado. É utilizado somente quando a configuração não for redundante.	192.168.15.1	0.0.0.0 a 255.255.255.255
Endereço de IP do CP A	Endereço IP do CPA. Somente habilitado quando a configuração for redundante. É o endereço do CP primário com o qual o servidor irá comunicar caso não exista falha.	192.168.15.69	0.0.0.0 a 255.255.255.255
Endereço de IP do CP B	Endereço IP do CPB. Somente habilitado quando a configuração for redundante. É o endereço do CP secundário com o qual o servidor irá comunicar caso ocorra uma falha.	192.168.15.70	0.0.0.0 a 255.255.255.255
Porta do Dispositivo	Porta TCP. Somente habilitado quando o checkbox *Usar Driver Bloqueante TCP/IP* está selecionado.	11740	11740 ou 11739

Tabela 101: 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.

5.7.7.2.1. Importando uma Configuração de Projeto

Utilizando o botão *Ler Configuração do Projeto*, conforme a Figura 96, 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 NET 1. A porta para conexão com o CP é sempre atribuída neste caso como 11740.

5.7.7.3. 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.

Estado	Valor	Descrição
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 102: 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 *QQSSSSLL*, 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	00SSSSLL	Bad	O valor lido não pode ser usado porque existe algum problema na conexão. É possível monitorar o valor de *_CommState* e diagnosticar o problema.
1	01SSSSLL	Uncertain	A qualidade não pode ser definida e pode ser apresentada no campo Substatus.
2	10SSSSLL	NA	Este valor é reservado e não é utilizado pelo padrão OPC DA.
3	11SSSSLL	Good	A qualidade é boa e o valor lido pode ser usado.

Tabela 103: Descrição do valor da Qualidade OPC DA

A Tabela 103 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.

5.7.7.4. 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 104: Limites da Comunicação com o Servidor OPC DA

Nota:

Número máximo de variáveis comunicando com um único CP: Esse limite é definido pelas "Tags de Comunicação" informadas na tabela Características Gerais.

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 104.

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 integradas, 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.

5.7.7.5. 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*.

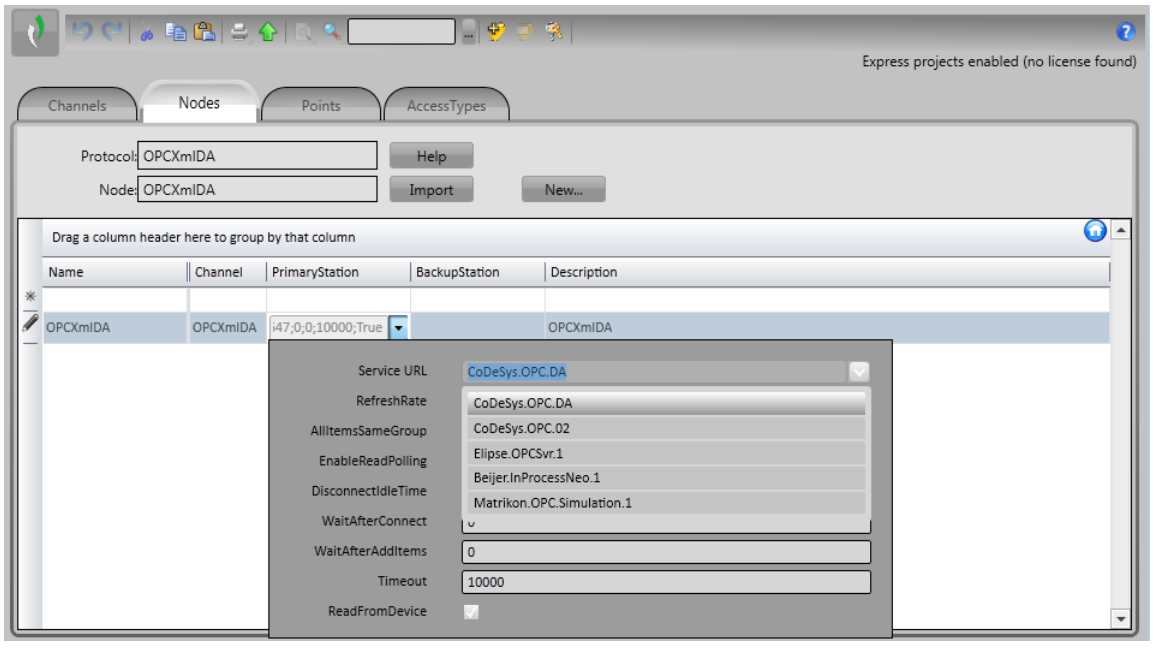


Figura 97: Selecionando Servidor OPC DA na Configuração do Cliente

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.

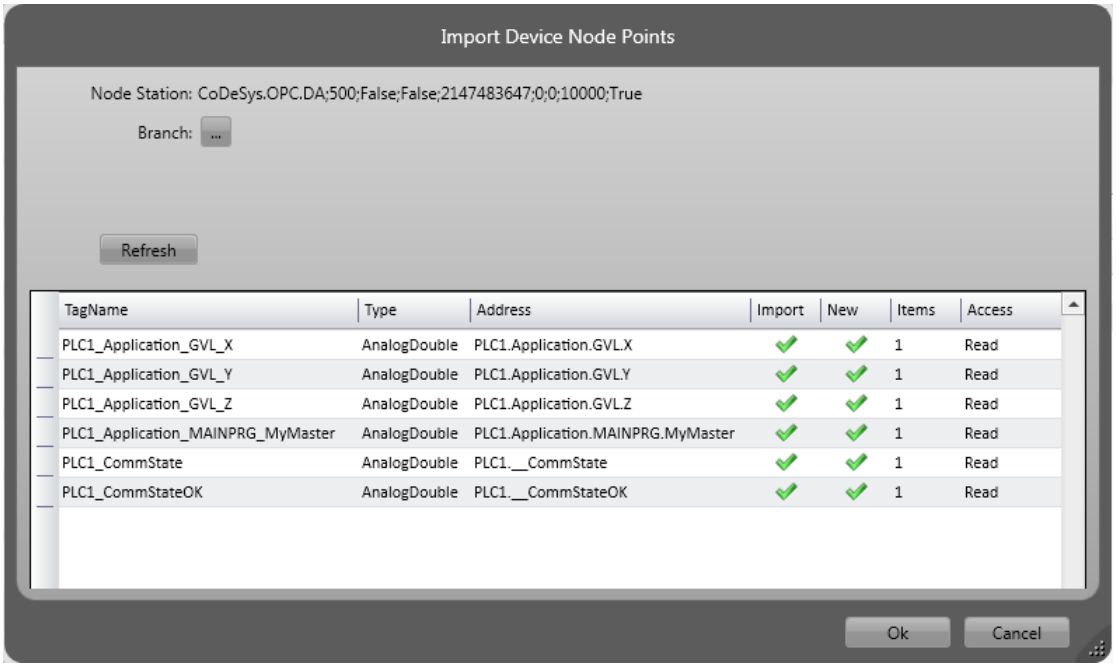


Figura 98: Lista de Símbolos Consultados pelo Cliente OPC DA

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 Simulado*r, do Manual de Utilização Mastertool.

5.7.8. 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.

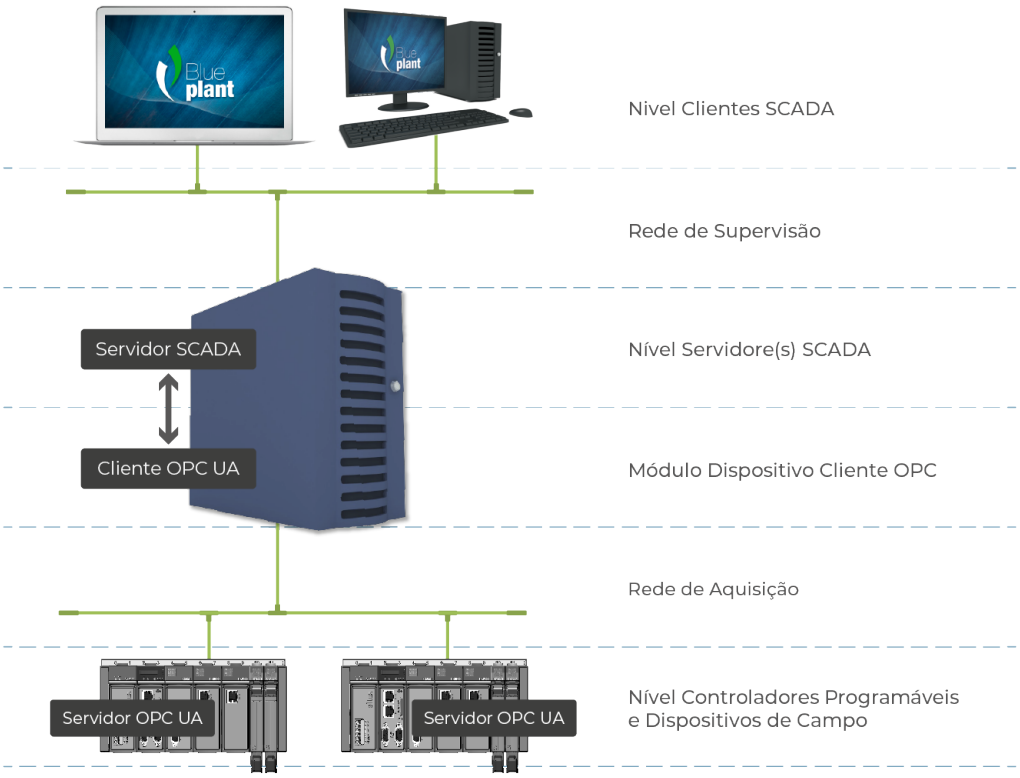


Figura 99: Arquitetura Típica OPC UA

A figura acima apresenta uma arquitetura típica para comunicação de sistema SCADA e CPs em projeto de automação. Todos os papéis presentes na comunicação estão explícitos nesta figura independente do local onde estejam executando, eles podem estar em um mesmo equipamento ou em equipamentos diferentes. Cada um dos papéis desta arquitetura é descrito na tabela abaixo.

Papel	Descrição
Nível Controladores Programáveis e Dispositivos de Campo	Os dispositivos de campo e os CPs são os dispositivos nos quais as informações do estado de operação e controle da planta são armazenadas. Os sistemas SCADA acessam as informações nestes dispositivos e armazenam nos Servidores SCADA para consulta pelos Clientes SCADA durante a operação da planta.
Módulo Servidor OPC UA	O Servidor OPC UA é um módulo interno dos CPs responsável por receber as requisições OPC UA e traduzi-las para a comunicação com os dispositivos de campo.
Rede de Aquisição	A rede de aquisição é a rede na qual trafegam as mensagens OPC UA para solicitar os dados que são coletados dos CPs e dispositivos de campo.
Módulo Cliente OPC UA	O módulo Cliente OPC UA, que faz parte do Servidor SCADA, é responsável por fazer requisições aos Servidores OPC UA utilizando o protocolo OPC UA. Os dados coletados por ele são armazenados na base de dados do Servidor SCADA.

Papel	Descrição
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 105: 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. 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.

5.7.9. Mestre EtherCAT

EtherCAT (*Ethernet Control Automation Technology*) é um protocolo com arquitetura mestre-escravo de alto desempenho, para Ethernet determinística, que permite desempenho em tempo real pois atualiza 1000 E/S distribuídas em 30 μS ou 100 eixos de servomotores a cada 100 μS usando par trançado ou cabos de fibra óptica. Além 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 nanosegundos. 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.

5.7.9.1. Instalando e inserindo dispositivos EtherCAT

A fim de ser possível inserir e configurar dispositivos EtherCAT como objetos na árvore de dispositivos, os dispositivos Escravos devem ser instalados.

O dispositivo Mestre é instalado automaticamente pela instalação padrão do Mastertool. O Mestre EtherCAT define quais Escravos podem ser inseridos.

Para instalar os dispositivos Escravos deve ser aberto o diálogo *Repositório de Dispositivos*, utilizar o filtro *Arquivo de Configuração da Descrição do Dispositivo EtherCAT XML (*.xml)* e selecionar os arquivos de descrição de dispositivo (*EtherCAT XML Device Description / ESI EtherCAT Slave Information*), fornecido com o hardware. As descrições para os Escravos estão disponíveis como arquivos XML (tipo de arquivo: *.xml).

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

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

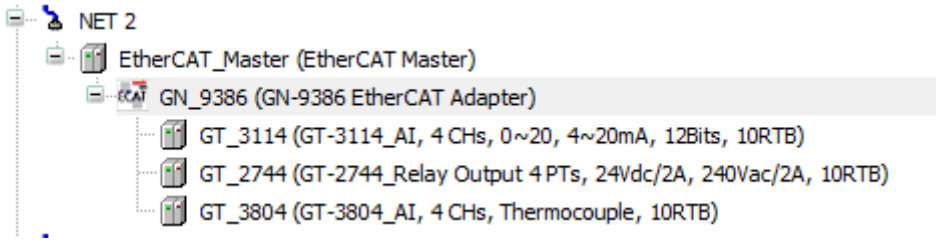


Figura 100: Exemplo de Configuração EtherCAT

ATENÇÃO

- É permitida apenas uma instância do Mestre EtherCAT por projeto.
- Permitido apenas em portas NET integradas.
- 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.

5.7.9.1.1. EtherCAT - Scan Devices

O comando *Scan for Devices...*, disponível no menu de contexto do Mestre EtherCAT, executa uma busca pelos dispositivos Escravos instalados fisicamente na rede EtherCAT do CP atualmente conectado. Isso significa que com este comando é possível detectar e visualizar os componentes de hardware na janela apresentada na figura abaixo, permitindo que o usuário possa mapeá-los diretamente na *Árvore de Dispositivos* do projeto.

É importante salientar que, quando o comando *Scan for Devices...* é selecionado, uma conexão com o CP será estabelecida automaticamente antes da busca iniciar e será encerrada quando a busca terminar. Assim, para que este comando seja executado pela primeira vez, a conexão do Gateway deve ser configurada e deve-se fazer um download do projeto com um Mestre EtherCAT configurado no CP.

Quando o comando for executado, o campo *Scanned Devices* vai conter uma lista de todos os dispositivos e módulos encontrados durante a última verificação. Para adicioná-los ao projeto, basta clicar no botão *Copy All Devices to Project*. É possível, também, executar uma comparação entre os dispositivos encontrados na busca com os presentes no projeto, selecionando a caixa *Show differences to project*.

Se você adicionar um módulo Mestre EtherCAT ao projeto e utilizar o comando *Scan for Devices...* você terá uma lista de todos os Escravos EtherCAT disponíveis. Aparecerão entradas escritas em negrito, caso exista mais de um dispositivo com a mesma descrição. Com um duplo clique sobre esta entrada uma lista será aberta e o dispositivo desejado pode ser selecionado.

Após concluir as modificações na configuração da rede EtherCAT, é necessário executar um novo download do projeto, para que as modificações sejam aplicadas.

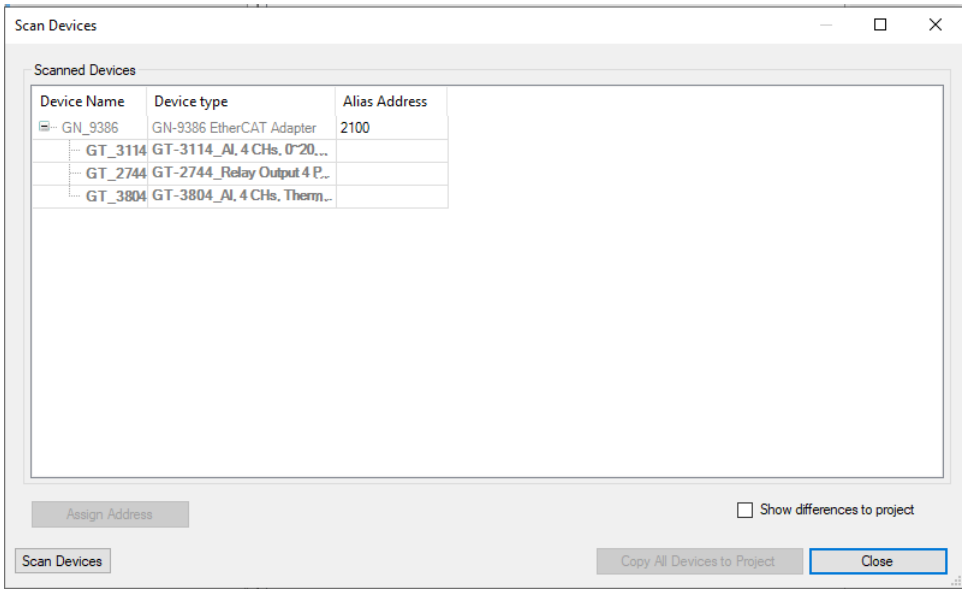


Figura 101: Diálogo Procurar Dispositivos para EtherCAT

5.7.9.2. 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*.

5.7.9.2.1. EtherCAT Master - General

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

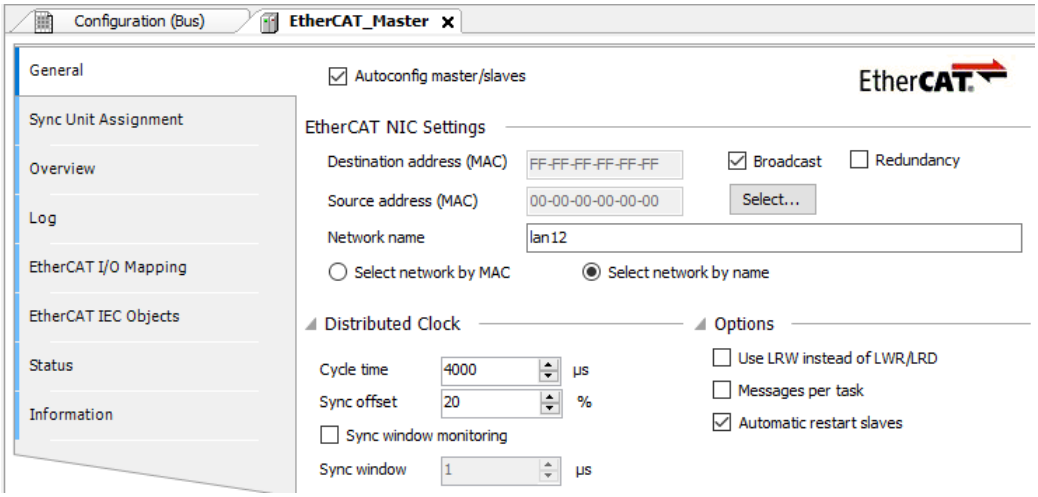


Figura 102: Diálogo de Configuração do Mestre EtherCAT

Configuração	Descrição	Padrão	Opções
Autoconfig master/slaves	Habilita a configuração automática do Mestre e dos Escravos.	Marcado	Marcado Desmarcado
EtherCAT NIC Settings			

Configuração	Descrição	Padrão	Opções
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 106: 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.

5.7.9.2.2. 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.

5.7.9.2.3. 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.

5.7.9.2.4. 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.

5.7.9.2.5. 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.

5.7.9.2.6. EtherCAT Master - Status / Information Tabs

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*.

5.7.9.3. 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*.

5.7.9.3.1. EtherCAT Slave - General

A seguir são apresentados os parâmetros gerais encontrados na tela inicial de configuração do Escravo EtherCAT. Estes campos estarão disponíveis somente se o modo *Autoconfiguração* (Mestre) não estiver ativado.

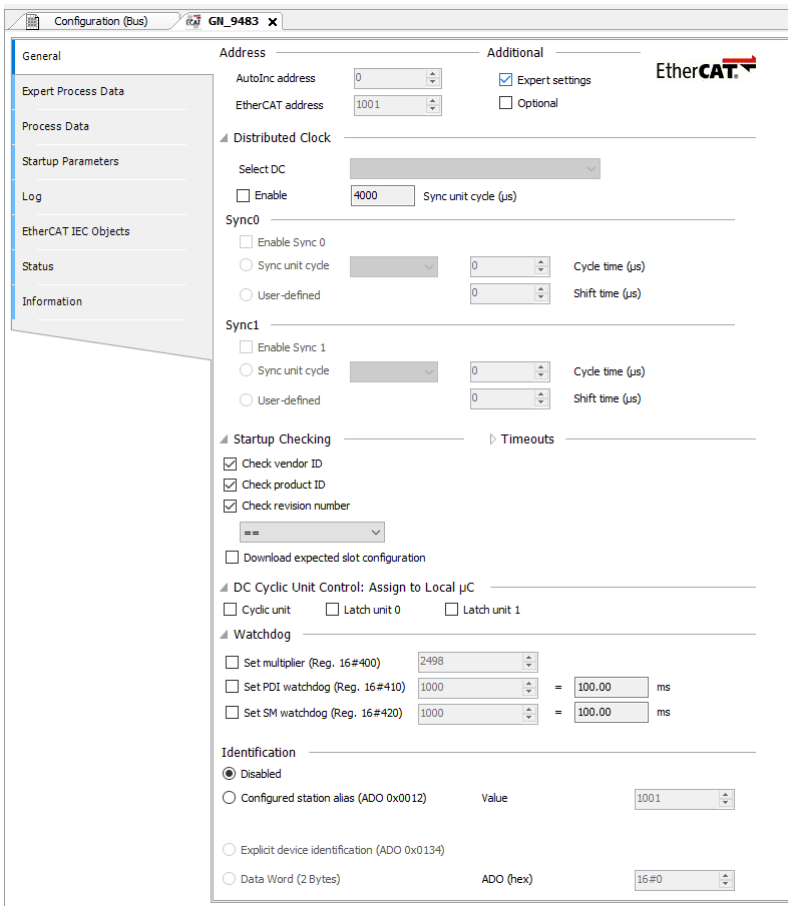


Figura 103: Diálogo de Configuração do Escravo EtherCAT

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	Selecionando esta opção o fator escolhido será multiplicado pelo valor em Sinc. Unidade de Ciclo.	Desmarcado	Marcado Desmarcado
User-defined Sync 0	Se marcado, o tempo desejado em microssegundos pode ser inscrito no campo Tempo de Ciclo (μs).	Desmarcado	Marcado Desmarcado
Cycle time [μs] Sync 0	Mostra o tempo de ciclo atualmente definido.	100000	1 a 2147483647
Shift Time [μs] Sync 0	Tempo entre os eventos de sincronização e as Outputs Valid ou Input Latch.	0	-2147483648 a 2147483647
Enable (Sync 1)	Habilita as configurações da unidade de sincronização Sync 1.	Desmarcado	Marcado Desmarcado
Sync unit cycle Sync 1	Selecionando esta opção o fator escolhido será multiplicado pelo valor em Sinc. Unidade de Ciclo.	Desmarcado	Marcado Desmarcado
User-defined Sync 1	Se marcado, o tempo desejado em microssegundos pode ser inscrito no campo Tempo de Ciclo (μs).	Desmarcado	Marcado Desmarcado
Cycle time [μs] Sync 1	Mostra o tempo de ciclo atualmente definido.	100000	1 a 2147483647
Shift Time [μs] Sync 1	Tempo entre os eventos de sincronização e as Outputs Valid ou Input Latch.	0	-2147483648 a 2147483647
Check vendor ID	Se desmarcado, desabilita a Verificação do ID do Fornecedor.	Marcado	Marcado Desmarcado
Check product ID	Se desmarcado, desabilita a Verificação do ID do Produto.	Marcado	Marcado Desmarcado
SDO access	Configura uma referência para a verificação do tempo de um Acesso SDO.	-	0 a 100000
I -> P	Configura uma referência para a verificação do tempo de troca do estado Inicial para Pré-Operacional.	-	0 a 100000

Configuração	Descrição	Padrão	Opções
P -> S/S -> O	Configura uma referência para a verificação do tempo de troca dos estados Pré-Operacional para Segurança-Operacional e de Segurança-Operacional para Operacional.	-	0 a 100000
Cyclic unit	Atribui a Unidade de Ciclo ao microprocessador local.	Desmarcado	Marcado Desmarcado
Latch Unit 0	Atribui a Latch Unit 0 ao microprocessador local.	Desmarcado	Marcado Desmarcado
Latch Unit 1	Atribui a Latch Unit 1 ao microprocessador local.	Desmarcado	Marcado Desmarcado

Tabela 107: 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.

5.7.9.3.2. *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 - I/O Mapping como saídas e entradas para o CP, no qual as variáveis do projeto podem ser mapeadas.

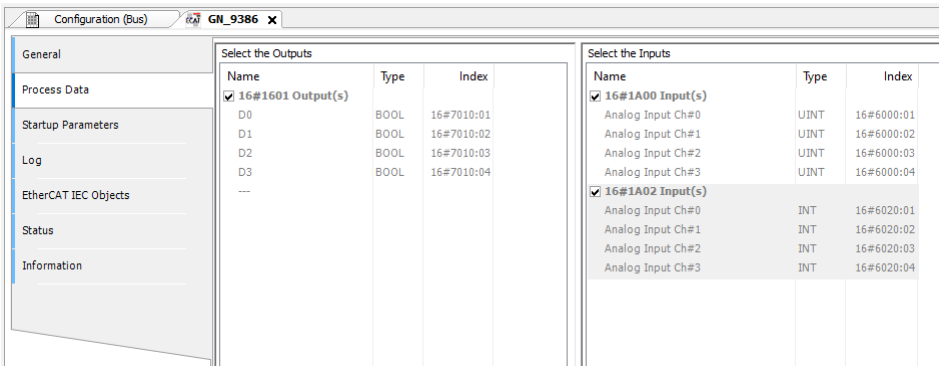


Figura 104: Diálogo Dados do Processo

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 confguration*.

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.

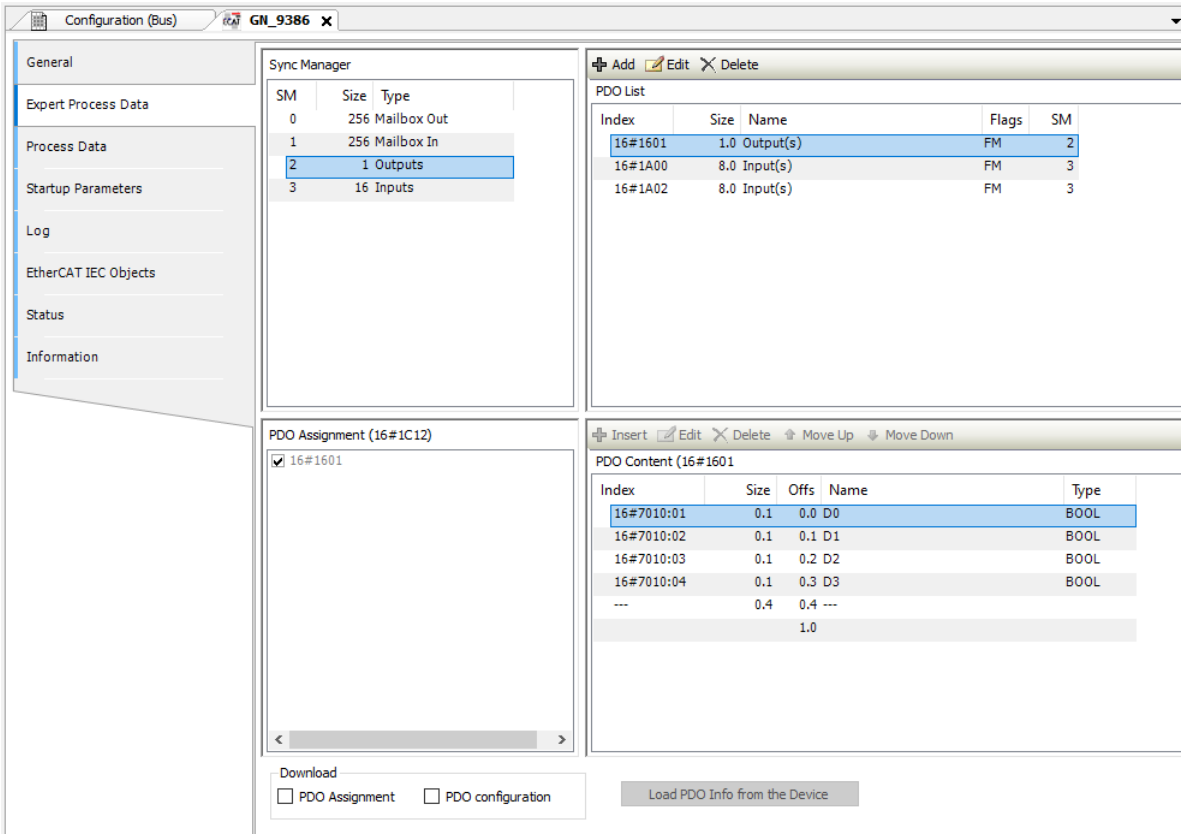


Figura 105: Diálogo Dados de Processo Avançados

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.

5.7.9.3.3. *EtherCAT Slave - Edit PDO List*

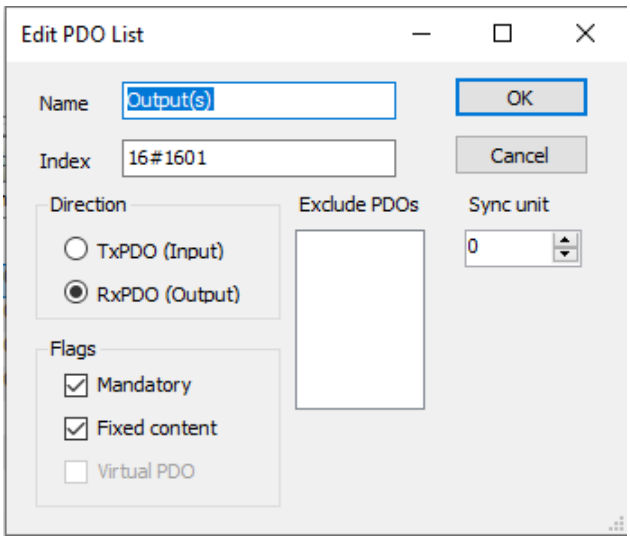


Figura 106: Diálogo Editar Lista PDO

Este diálogo é aberto através do menu de contexto da área Lista PDO, apresentada na Figura 105. 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.
- *Exlude 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.

5.7.9.3.4. *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*.

5.7.9.3.5. *EtherCAT Slave - 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*.

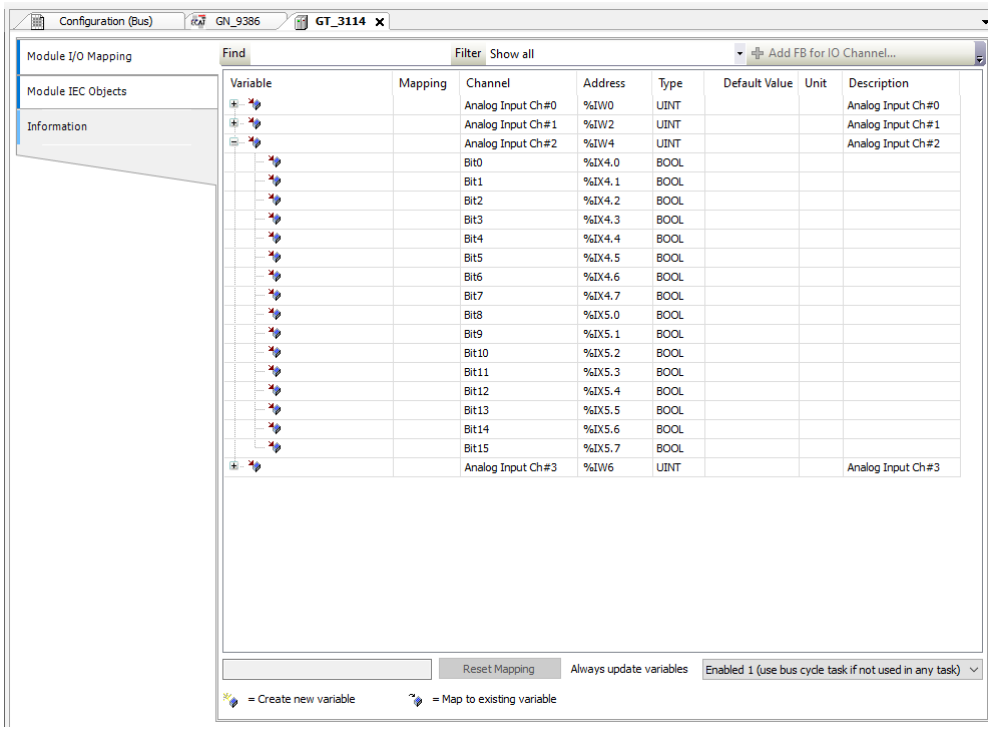


Figura 107: Diálogo Mapeamento de E/S Escravo

5.7.9.3.6. 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*.

5.7.10. 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 explicitas.

O sistema em execução pode atuar tanto como Scanner como Adapter. Cada interface NET integrada 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/Saida, nestes módulos podem ser adicionadas variaveis 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 subrede.

5.7.10.1. Adição de Scanners e Adapters EtherNet/IP

Para adicionar um EtherNet/IP Scanner ou um Adapter é necessário adicionar um *Adaptador Ethernet* abaixo da NET desejada. Isto pode ser feito através do comando *Adicionar Dispositivo*. Abaixo deste *Adaptador Ethernet*, é possível adicionar um *Scanner* ou um *Adapter*.

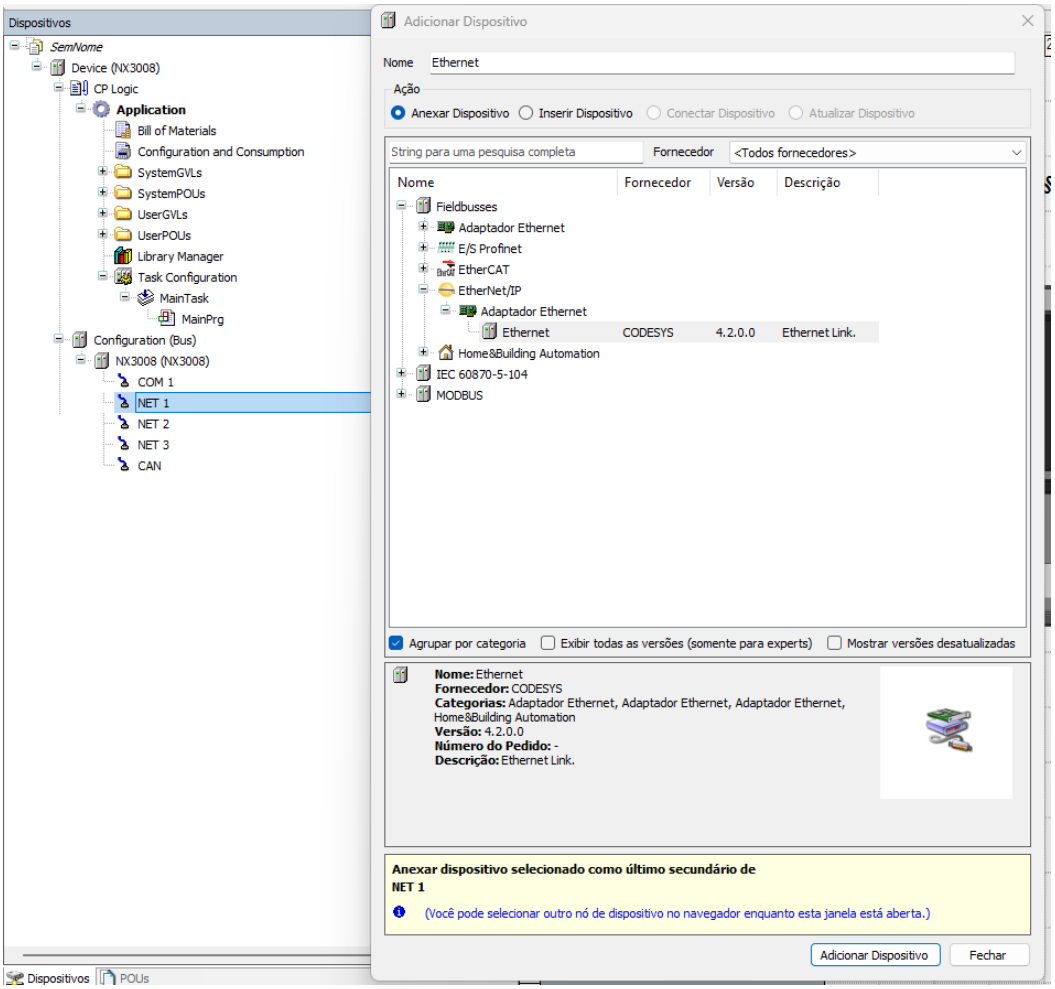


Figura 108: Adicionando um Adaptador Ethernet

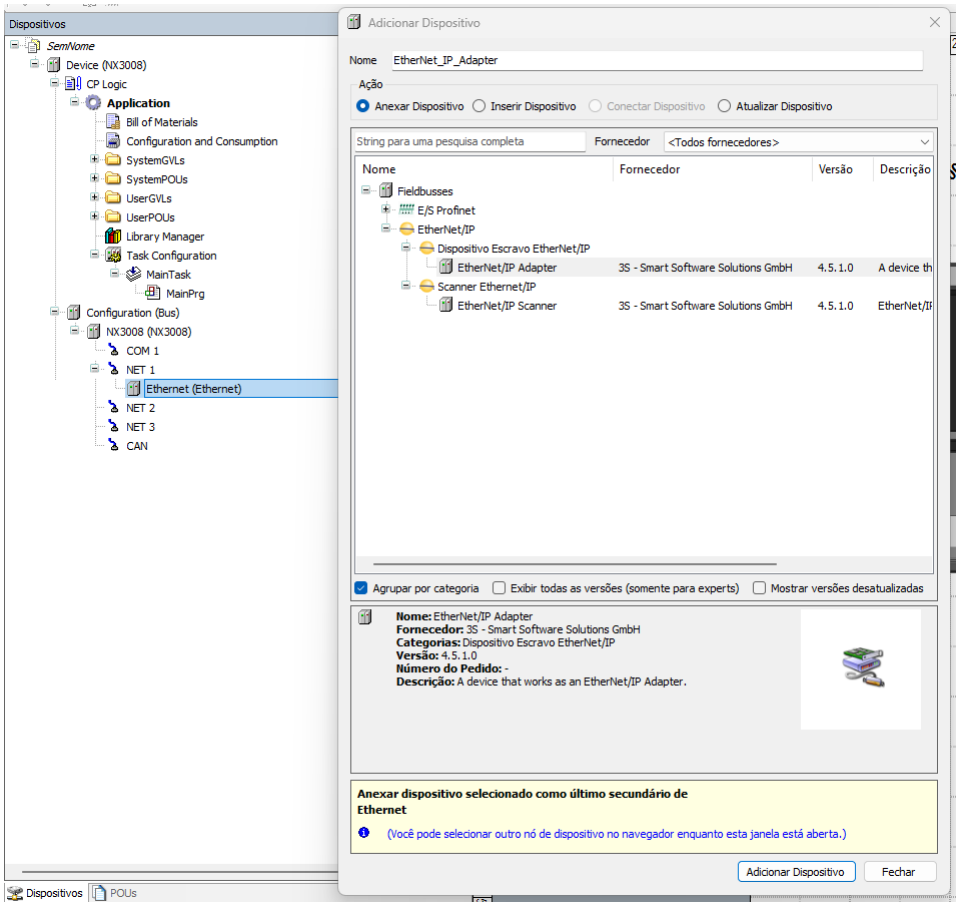


Figura 109: Adicionando um Adapter ou Scanner EtherNet/IP

5.7.10.2. Configuração do Scanner EtherNet/IP

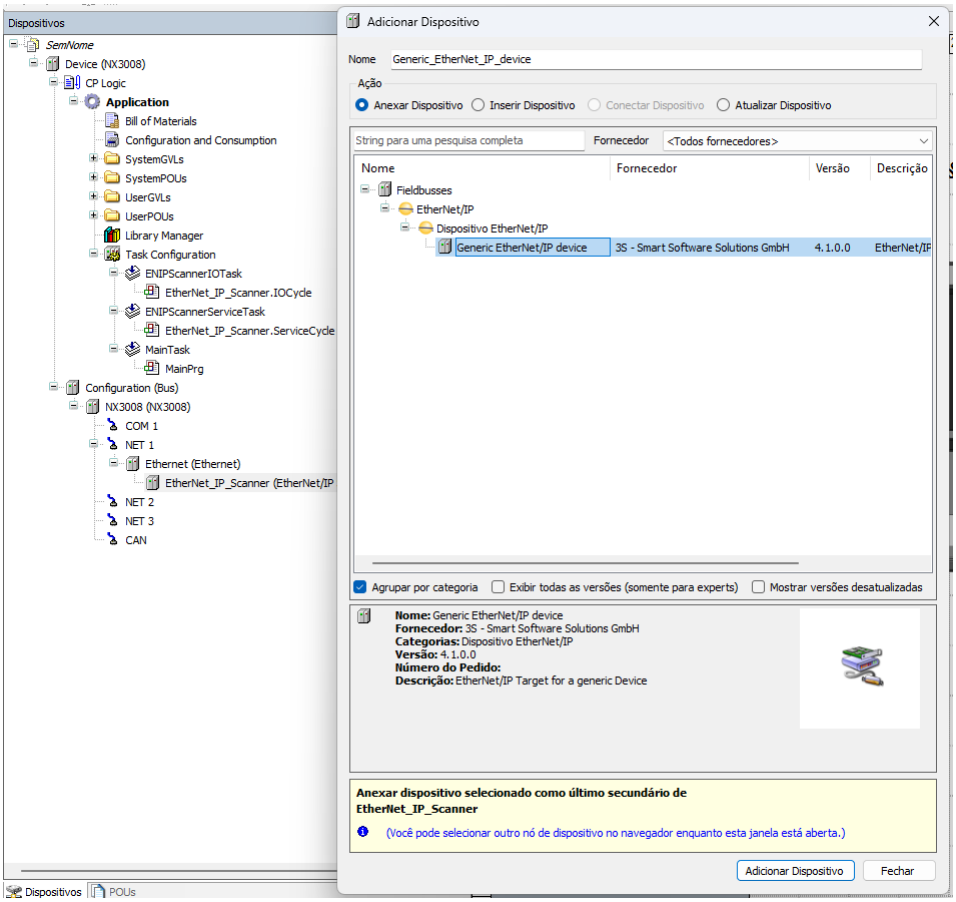


Figura 110: Adicionando um Adapter EtherNet/IP Sob o Scanner

5.7.10.2.1. 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"*.

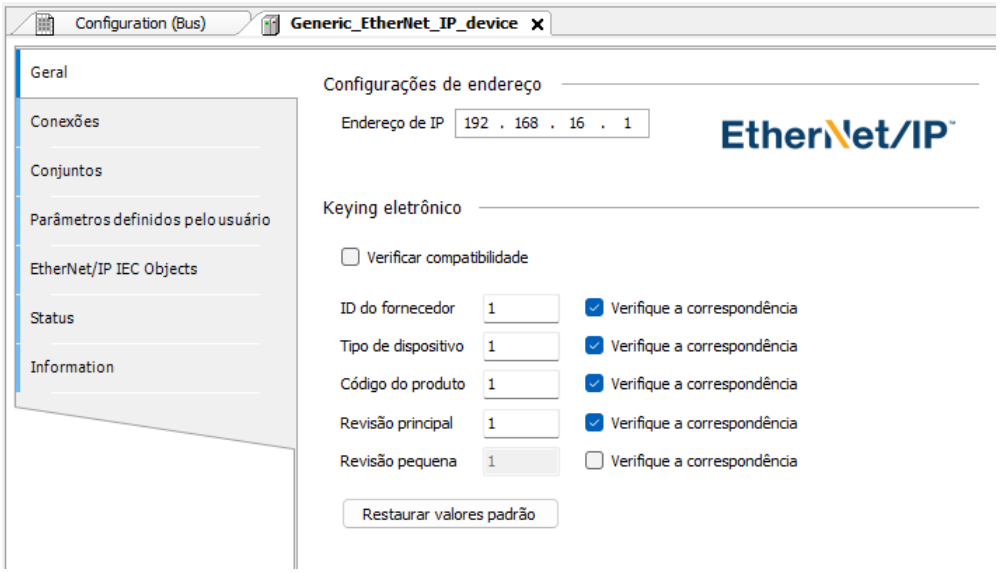


Figura 111: Aba Geral - EtherNet/IP

5.7.10.2.2. *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.

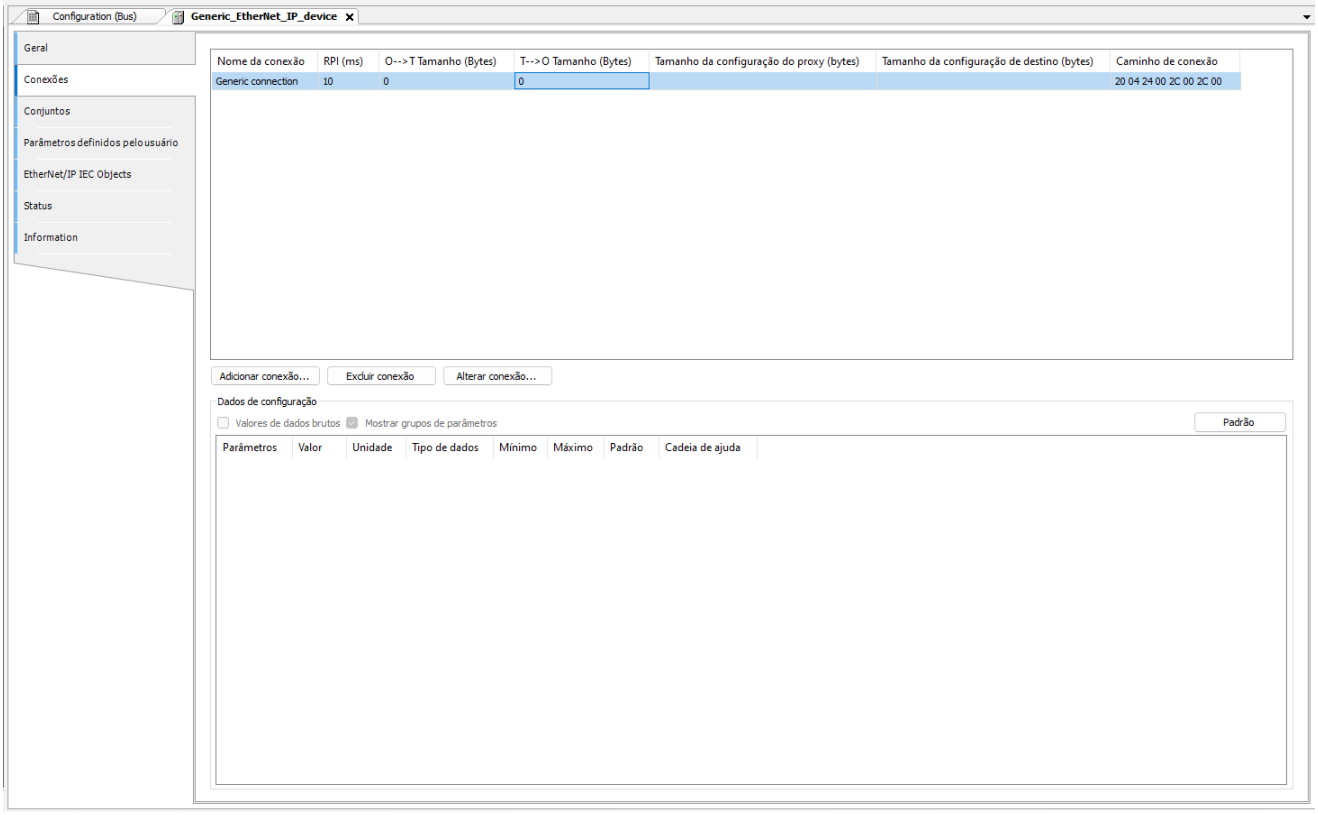


Figura 112: Aba Conexões - EtherNet/IP

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 108: Parâmetros de Conexão do EtherNet/IP

Para *Adicionar* novas conexões, há o botão *Adicionar Conexão...* o qual vai abrir a janela *Nova Conexão*. Nesta janela é possível configurar um novo tipo de conexão daquelas predefinidas no EDS do Adapter ou uma conexão do zero quando utilizado um dispositivo Genérico.

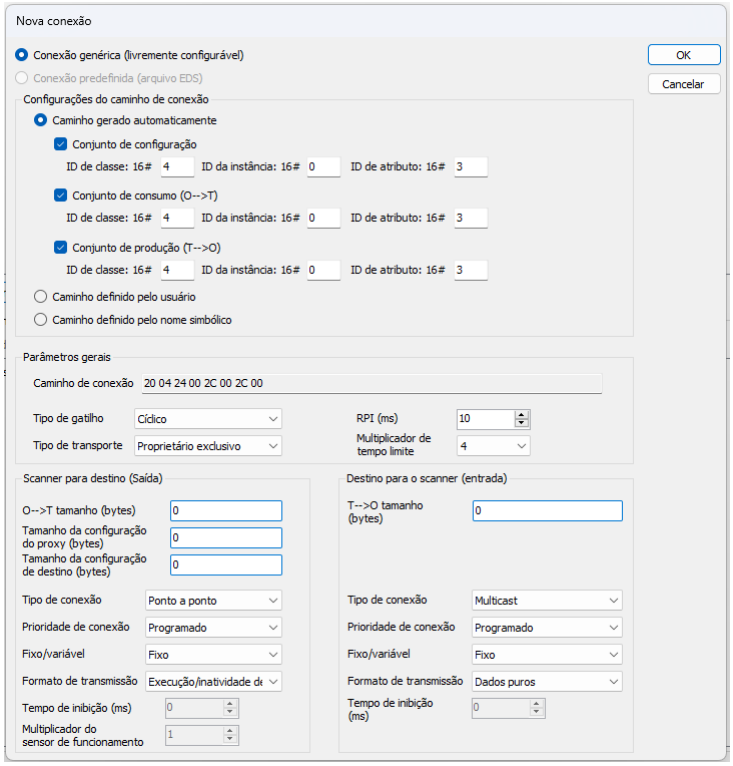


Figura 113: Janela de Novas Conexões EtherNet/IP

5.7.10.2.3. *Assemblies*

A área superior da aba *Assemblies* lista todas as conexões configuradas. Quando uma conexão está selecionada, as entradas e saídas associadas são mostradas na área inferior da aba.

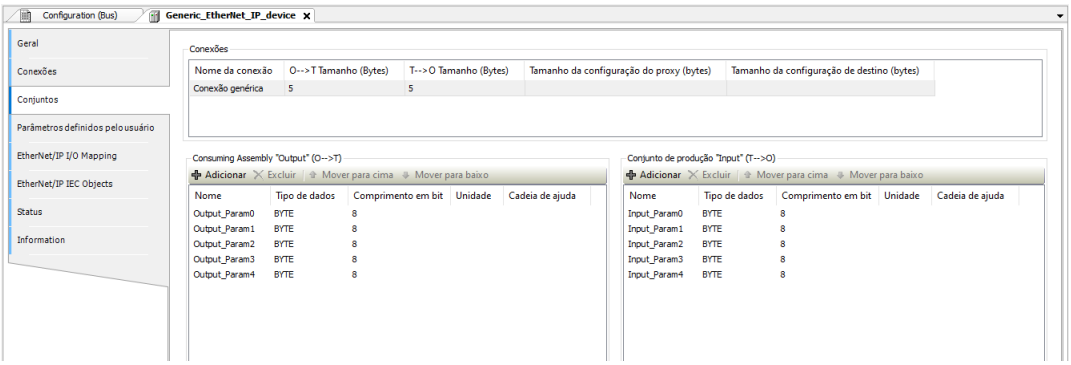


Figura 114: Aba Assemblies - EtherNet/IP

Conjunto de Saídas e *Conjunto de Entradas*:

Configuração	Descrição
Adicionar	Abre a caixa de diálogo "Adicionar Entrada/Saída"
Excluir	Deleta todas as Entradas/Saídas Selecionadas.
Mover Para Cima	Movimenta pela lista a Entrada/Saída selecionada.
Mover Para Baixo	A ordem na lista determina a ordem no Mapeamento de E/S.

Tabela 109: 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 110: Janela "Adicionar Entrada/Saída" – EtherNet/IP

5.7.10.2.4. 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.

5.7.10.3. 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.

5.7.10.3.1. 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 exporta-lo.

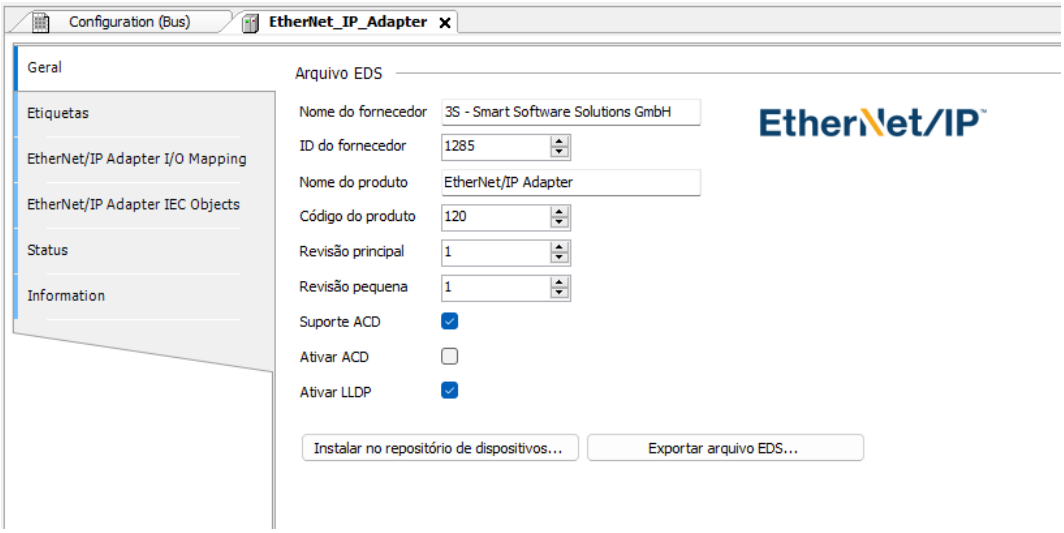


Figura 115: Aba Geral - EtherNet/IP_Adapter

5.7.10.3.2. 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.

5.7.10.4. Configuração do módulo EtherNet/IP

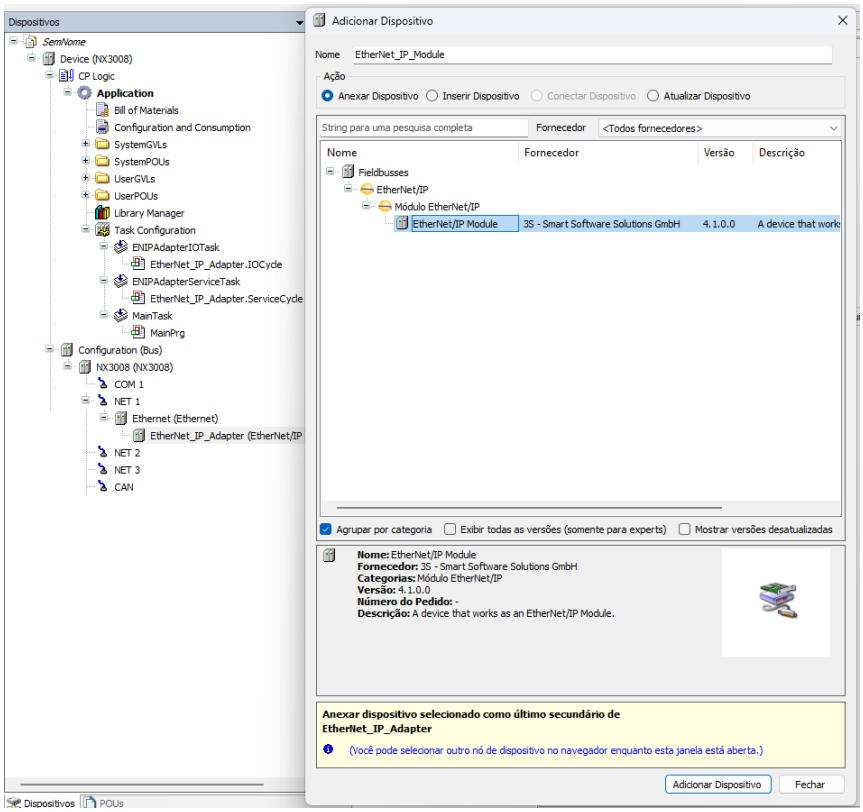


Figura 116: Adicionando um Módulo EtherNet/IP Abaixo do Adapter

5.7.10.4.1. Assemblies

Os parâmetros da aba *Geral* do módulo seguem as mesmas regras descritas nas tabelas 109 e 110.

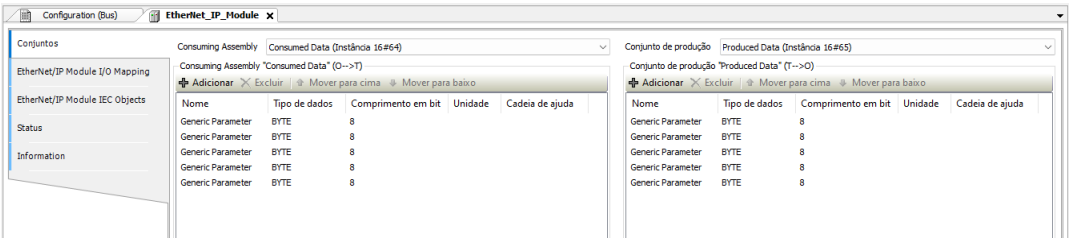


Figura 117: Aba Assemblies Módulo - EtherNet/IP

5.7.10.4.2. 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.

5.7.11. CANopen Manager

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

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

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

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

ATENÇÃO

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

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

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

Tabela 111: Baudrate X Comprimento do Barramento

5.7.11.1. Instalando e inserindo dispositivos CANopen

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

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

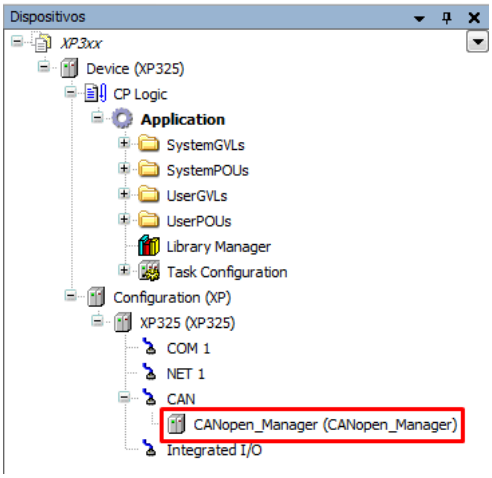


Figura 118: Adicionando o CANopen Manager

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

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

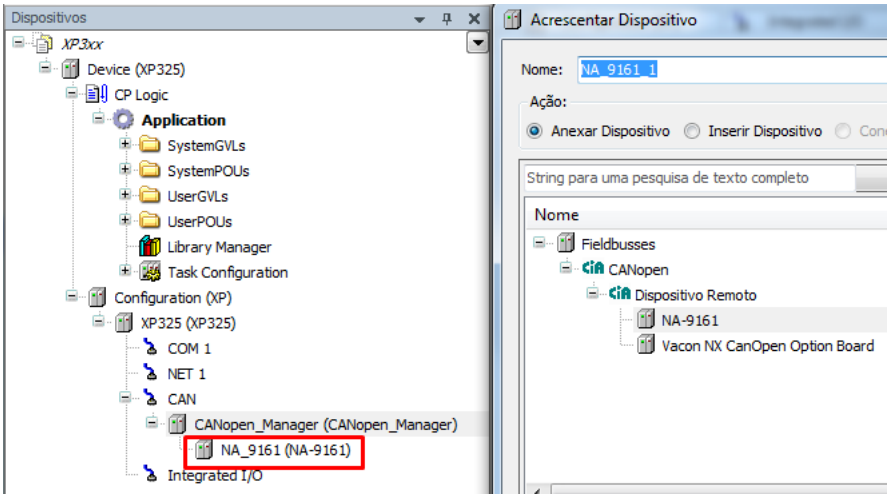


Figura 119: Adicionando um Dispositivo CANopen Escravo

5.7.11.2. 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*:

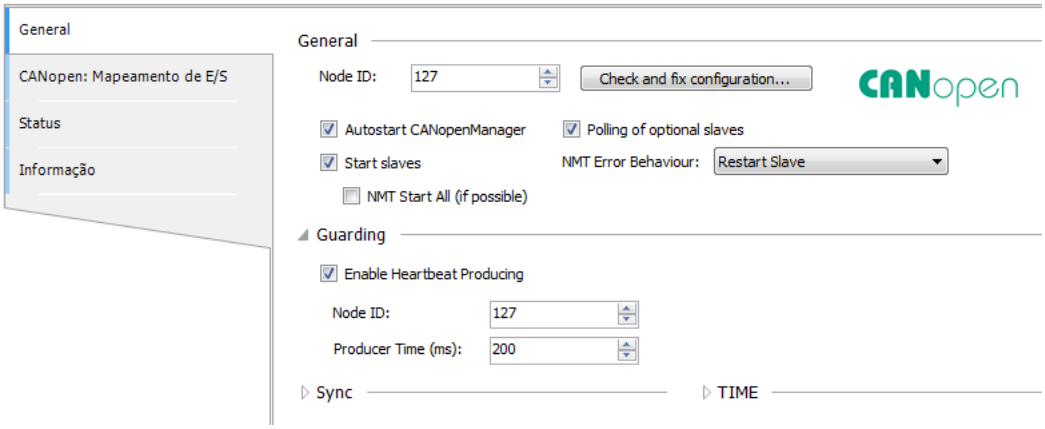


Figura 120: Parâmetros gerais do CANopen Manager

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:

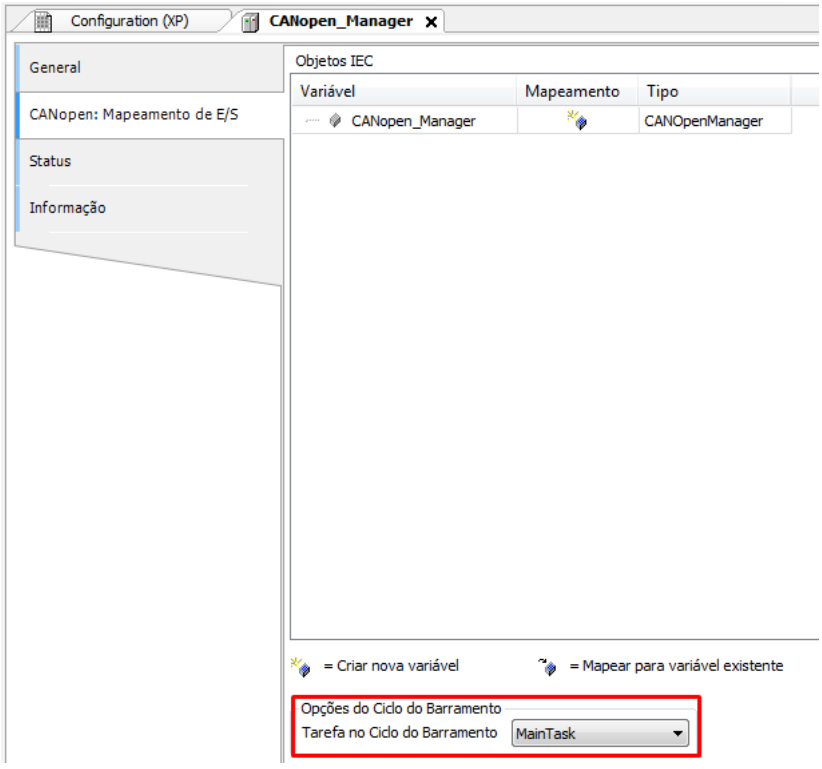


Figura 121: Configuração da tarefa cíclica do barramento CANopen

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 (5 ms, digamos) que não pode ser executado na MainTask devido ao código pesado da aplicação.

5.7.11.3. Configuração do escravo CANopen

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

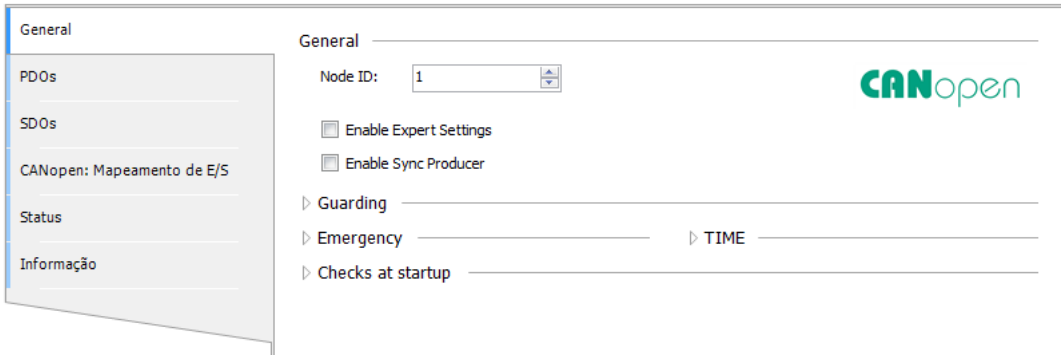


Figura 122: Parâmetros de Escravo CANopen

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).

5.7.12. PROFINET Controller

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

5.8. Desempenho de Comunicação

5.8.1. Servidor MODBUS

Os dispositivos MODBUS configuráveis no controlador 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, o controlador 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 o controlador executar todas as tarefas de menor prioridade. Portanto, um controlador com apenas um MODBUS Ethernet Servidor terá um desempenho maior do que um controlador que utilize quatro destes dispositivos.

5.8.1.1. 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 do controlador, 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 112: Taxa de Comunicação de um Servidor MODBUS em uma Interface Integrada

ATENÇÃO

Os desempenhos de comunicação mencionados neste capítulo são exemplos utilizando um controlador 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 x (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 s;

N = 1 x (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.

5.8.2. 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.

5.9. Desempenho do Sistema

Em casos onde a aplicação possui apenas uma tarefa de usuário MainTask, responsável pela execução de uma única unidade de programação do tipo Programa denominada MainPrg (Como no perfil Simples), o CP consome um determinado tempo para que a tarefa seja processada. A esse tempo damos o nome de *Tempo de Execução*.

Em uma aplicação, podemos conhecer o *Tempo de Execução* médio da aplicação usando o Mastertool, na *Árvore de Dispositivos*, no item *Device*, no seguinte caminho:

CP Logic-> Application-> Task Configuration na aba *Monitorar*, na coluna *Tempo de Ciclo Médio*.

Deve-se atentar ao *Tempo de Execução* para que ele não ultrapasse 80% do intervalo configurado na tarefa de usuário MainTask. Por exemplo, em uma aplicação onde o intervalo é de 100 ms, um *Tempo de Execução* adequado é de até 80 ms. Isso se deve ao fato de que a UCP necessita de um tempo para a execução de outras tarefas como o processamento da comunicação, tratamento do visor e cartão de memória, e essas tarefas também acontecem dentro do intervalo (os 20% restantes do *Tempo de Execução*).

5.9.1. Cartão de Memória

A transferência de dados envolvendo o cartão de memória é realizada em segundo plano pela UCP, pois esta dá prioridade para a execução da aplicação de usuário e processamento da comunicação. Desta forma, a transferência de arquivos para o cartão poderá sofrer um acréscimo de tempo significativo, dependendo do tempo de execução da aplicação do usuário.

O tempo necessário para ler/escrever arquivos no cartão será afetado diretamente pelo tempo de execução da aplicação de usuário, uma vez que essa aplicação tem prioridade na execução.

Para mais informações sobre a utilização do cartão de memória, ver seção Cartão de Memória.

5.10. Relógio RTC

O controlador possui 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:

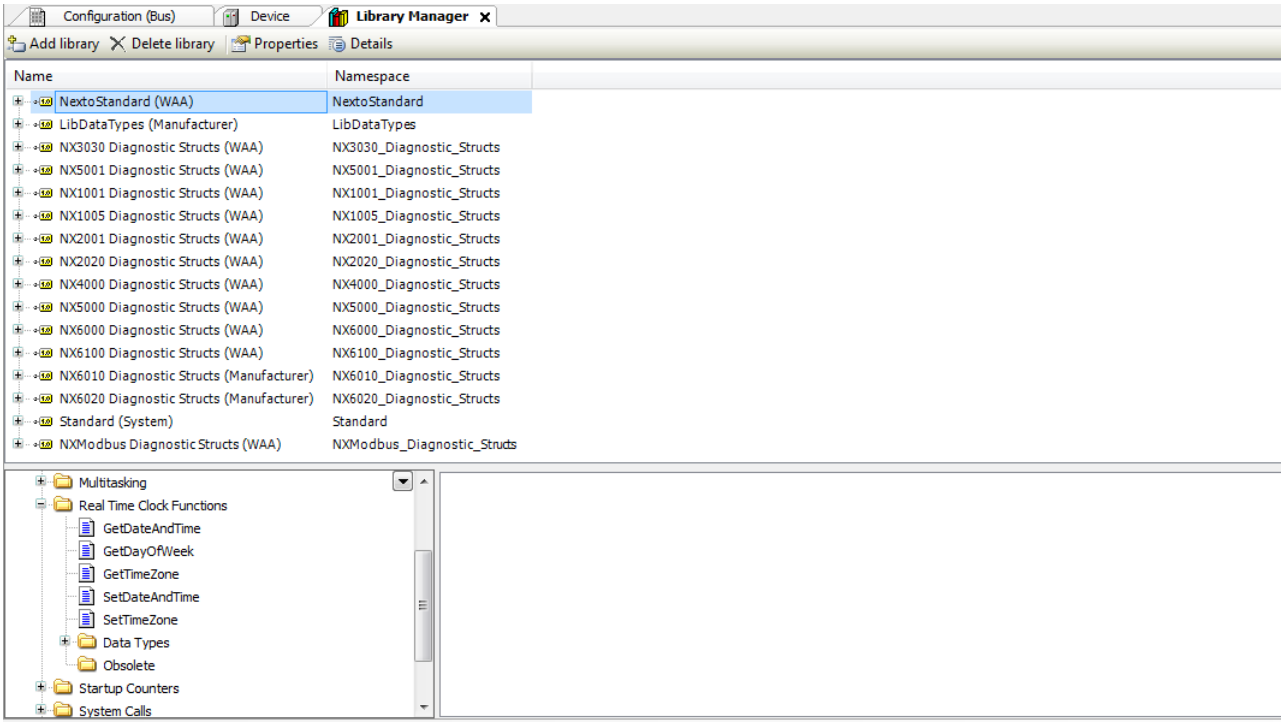


Figura 123: Blocos e Funções de Escrita e Leitura do Relógio

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

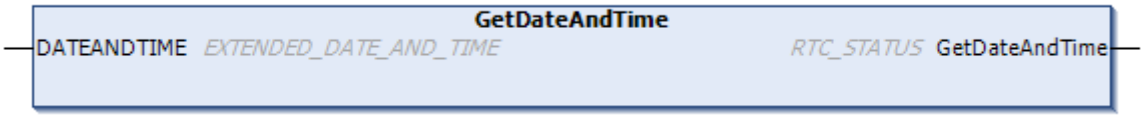


Figura 124: Leitura da Data e Hora

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 122.

Tabela 113: Parâmetros de Entrada GetDateAndTime

Parâmetros de saída	Tipo	Descrição
GETDATEANDTIME	RTC_STATUS	Retorna o estado de erro da função, ver Tabela 124.

Tabela 114: 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:

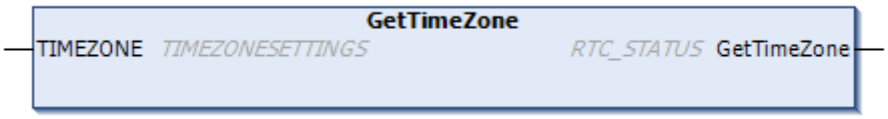


Figura 125: Leitura das Configurações de Fuso Horário

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

Tabela 115: Parâmetros de Entrada GetTimeZone

Parâmetros de saída	Tipo	Descrição
GetTimeZone	RTC_STATUS	Retorna o estado de erro da função, ver Tabela 124.

Tabela 116: 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 126: Leitura do Dia da Semana

Parâmetros de saída	Tipo	Descrição
GetDayOfWeek	DAYS_OF_WEEK	Retorna o dia da semana. Ver Tabela 123.

Tabela 117: 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.

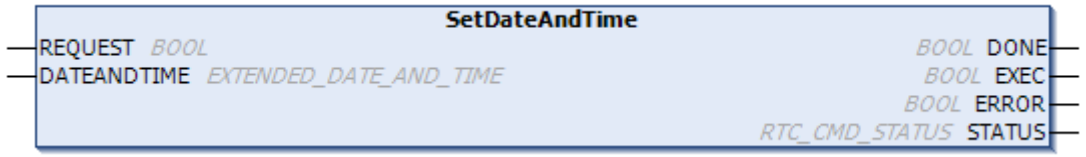


Figura 127: Ajuste de Data e Hora em Milissegundos

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 122.

Tabela 118: 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.

Parâmetros de saída	Tipo	Descrição
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 124.

Tabela 119: 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:

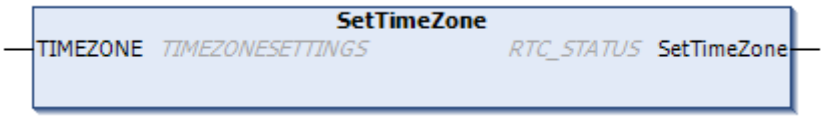


Figura 128: Ajuste de Fuso Horário

Parâmetros de entrada	Tipo	Descrição
TIMEZONE	TIMEZONESETTINGS	Estrutura com o valor de fuso horário a ser configurado. Ver Tabela 125.

Tabela 120: Parâmetros de Entrada SetTimeZone

Parâmetros de saída	Tipo	Descrição
SetTimeZone	RTC_STATUS	Retorna o erro ocorrido durante a leitura/configuração. Ver Tabela 124.

Tabela 121: 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 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 122: 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 123: 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 124: 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 125: 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 126 indicam o menor e o maior instante de tempo que o relógio RTC consegue representar. O valor mínimo corresponde à data e hora mais antiga que pode ser configurada, enquanto o valor máximo representa a data e hora mais futura válida. Esses limites devem ser respeitados ao definir ou ler o relógio do produto, garantindo que todas as funções de data e hora operem corretamente.

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

Tabela 126: Faixa de valores válidos para o Relógio RTC

5.11. Memória de Arquivos de Usuário

Os controladores do Sistema 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 do controlador. Esta área de memória pode variar de acordo com o modelo 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 o controlador 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 do controlador selecionado (direita), conforme mostram as indicações da figura abaixo.

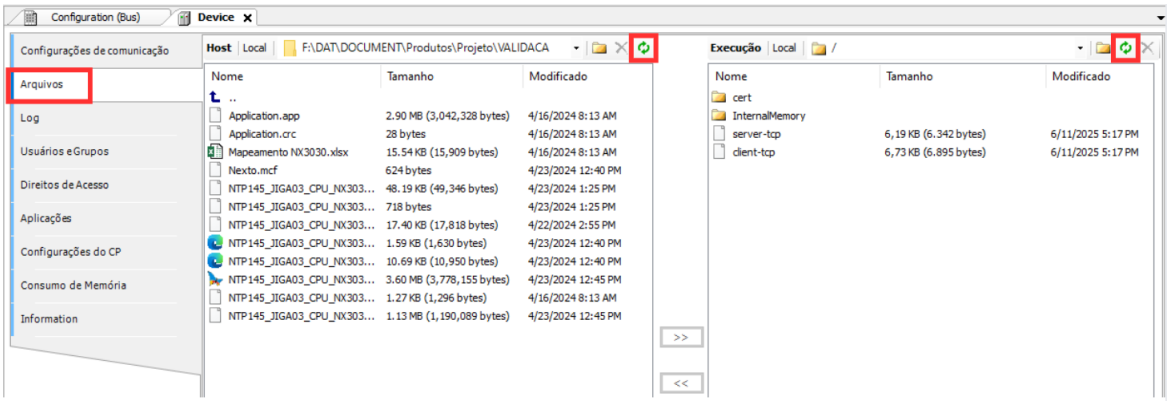


Figura 129: Acesso aos Arquivos de Usuário

Após atualizar a coluna de arquivos do controlador, será exibido o diretório raiz de arquivos armazenados no controlador 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 do controlador, uma vez que não é possível transferir arquivos para o diretório raiz. Caso seja necessário, podem ser criadas outras pastas no diretório raiz ou subpastas dentro da pasta *"InternalMemory"*.

Já a pasta *"MemoryCard"* é o diretório onde o cartão de memória estará montado, caso o mesmo esteja inserido na UCP. Arquivos transferidos para a pasta *"MemoryCard"* estarão sendo transferidos diretamente para dentro do cartão de memória. Conforme novas funcionalidades forem sendo adicionadas ao produto, algumas pastas podem aparecer e que devem ser ignoradas pelo usuário.

Para realizar a transferência de algum arquivo do microcomputador para a UCP, basta selecionar o arquivo desejado na coluna da esquerda e pressionar o botão "»", localizado no centro da tela, conforme figura abaixo. O tempo de transferência irá variar de acordo com o tamanho do arquivo e com o tempo de ciclo (execução) da aplicação atual da UCP, podendo levar vários minutos.

O usuário não precisa estar em Modo *Run* ou conectado à UCP para realizar as transferências, pois ela possui a capacidade de se conectar automaticamente quando o usuário realizar a transferência.

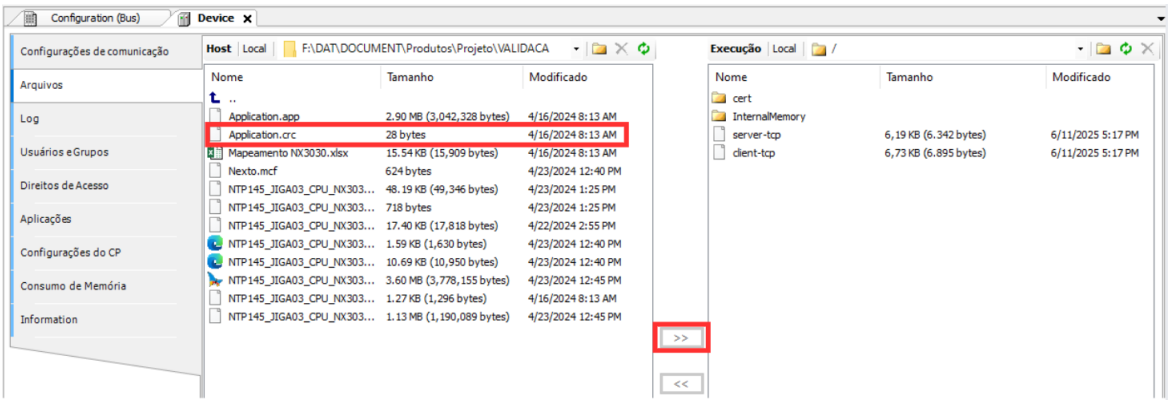


Figura 130: Transferindo Arquivos

ATENÇÃO

Os arquivos contidos dentro da pasta de um projeto criado pela ferramenta Mastertool possuem nomes especiais reservados pelo sistema, desta forma não podem ser transferidos através da aba *Arquivos*. Caso o usuário deseje transferir um projeto para a memória de usuário, será necessário compactar a pasta e então transferir o arquivo compactado (*.zip por exemplo).

Caso seja necessário transferir documentos do controlador 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.

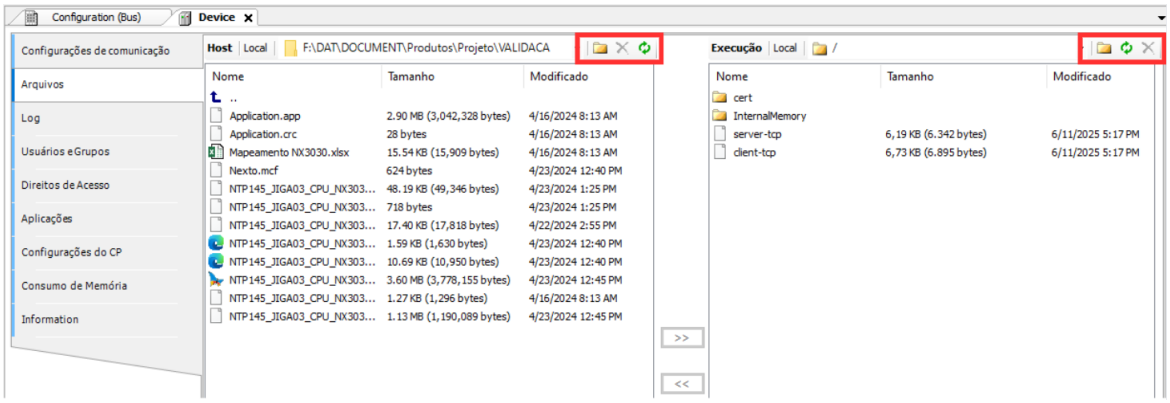


Figura 131: Opções de Utilização

ATENÇÃO

Para um controlador 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 do controlador. 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. 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. O sinal RTS está sempre desligado.
	RS232_RTS_ON	Controla a porta RS-232C. O sinal RTS está sempre ligado.
	RS232_RTS_CTS	Controla a porta RS-232C. 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. 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.

Tipo de dado	Opção	Descrição
	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 o controlador. Valores válidos são: 1, 4, 8 ou 14.
	MODE	Definido em SERIAL_MODE.
	ENABLE_RX_ON_TX	Quando verdadeiro, todos os bytes recebidos durante a transmissão serão descartados em vez de ir para a fila de RX. Utilizado para desabilitar a operação full-duplex na interface RS-422.
	ENABLE_DCD_EVENT	Quando verdadeiro, gera um evento externo quando o DCD é alterado.
	ENABLE_CTS_EVENT	Quando verdadeiro, gera um evento externo quando o CTS é alterado.
SERIAL_PARITY	PARITY_NONE	Lista todas as possibilidades de paridade.
	PARITY_ODD	
	PARITY_EVEN	
	PARITY_MARK	
	PARITY_SPACE	
SERIAL_PORT	COM 1	Lista todas as portas seriais disponíveis (COM 10, COM 11, COM 12, COM 13, COM 14, COM 15, COM 16, COM 17, COM 18, COM 19 – módulos de expansão).
	COM 2	
SERIAL_RX_CHAR_ EXTENDED	Define um caractere da fila RX modo estendido.	
	RX_CHAR	Byte de dados.
	RX_ERROR	Código de erro.
	RX_TIMESTAMP	Silêncio devido ao caractere anterior ou devido a outro evento que aconteceu antes desse caractere (configuração da porta serial, final da transmissão).
	Contém alguns campos que disponibilizam informações de status/erro sobre a fila RX, utilizados quando é utilizado o formato normal (sem erro e informações de estampa de tempo):	
	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).

Tipo de dado	Opção	Descrição
SERIAL_RX_QUEUE_ STATUS	RX_FIFO_OVERRUN_ ERRORS	Contador de erros de overrun na FIFO RX desde a configuração da porta serial, ou seja, erro no threshold configurado para a FIFO RX. Retorna para zero caso atinja o valor máximo (65535).
	RX_QUEUE_OVERRUN_ ERRORS	Contador de erros de overrun na fila RX desde a configuração da porta serial, ou seja, o valor máximo de caracteres (1024) foi ultrapassado e os dados estão sendo sobrescritos. Retorna zero caso atinja o valor máximo (65535).
	RX_ANY_ERRORS	Soma dos 5 últimos contadores de erros (frame, paridade, interrupção, overrun RX FIFO, overrun fila RX).
	RX_REMAINING	Número de caracteres remanescentes na fila RX.
SERIAL_STATUS	Lista os códigos de erro críticos que podem ser retornados pelos blocos funcionais da serial. Cada bloco retorna erros específicos, os quais serão mencionados na descrição dos mesmos:	
	NO_ERROR	Não existem erros.
	ILLEGAL_*	Retorna os parâmetros com valores inválidos ou fora da faixa: - SERIAL_PORT - SERIAL_MODE - BAUDRATE - DATA_BITS - PARITY - STOP_BITS - HANDSHAKE - UART_RX_THRESHOLD - TIMEOUT - TX_BUFF_LENGTH - HANDSHAKE_METHOD - RX_BUFF_LENGTH
	PORT_BUSY	Indica que a porta serial está sendo utilizada por outra instância
	HW_ERROR_UART	Erro de hardware detectado na UART.
	HW_ERROR_REMOTE	Erro de hardware ao comunicar com a porta serial remota.
	CTS_TIMEOUT_ON	Time-out na espera do CTS ser habilitado, nos sinais de modem RS-232C RTS/CTS, no bloco SERIAL_TX.
	CTS_TIMEOUT_OFF	Time-out na espera do CTS ser desabilitado, no handshake RS-232C RTS/CTS, no bloco SERIAL_TX.
	TX_TIMEOUT_ERROR	Time-out na espera pelo final da transmissão no bloco SERIAL_TX.
	RX_TIMEOUT_ERROR	Time-out na espera de todos os caracteres no bloco SERIAL_RX ou SERIAL_RX_EXTENDED.
	FB_SET_CTRL_ NOT_ALLOWED	O bloco SET_CTRL não pode ser utilizado caso o sinal de modem seja diferente de RS232_MANUAL.
	FB_GET_CTRL_ NOT_ALLOWED	O bloco GET_CTRL não pode ser utilizado caso o sinal de modem seja diferente de RS232_MANUAL.
	FB_SERIAL_RX_ NOT_ALLOWED	O bloco SERIAL_RX não está disponível para a fila RX, modo estendido.

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

Tabela 127: 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 132: 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 128: 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 129: 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 133: 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 130: Parâmetros de Entrada SERIAL_GET_CFG

Parâmetros de saída	Tipo	Descrição
DONE	BOOL	Essa variável é verdadeira quando o bloco funcional for executado por completo, caso contrário, ela é falsa.
EXEC	BOOL	Essa variável é verdadeira enquanto o bloco funcional estiver sendo executado, caso contrário, ela é falsa.
ERROR	BOOL	Essa variável é verdadeira quando o bloco funcional concluir a sua execução com algum erro, caso contrário, ela é falsa. Está vinculada à variável DONE, pois seu estado é exibido após a conclusão do bloco funcional.

Parâmetros de saída	Tipo	Descrição
STATUS	SERIAL_STATUS	Caso a variável ERROR seja verdadeira, a estrutura STATUS exibirá o erro encontrado na execução do bloco. Os estados, já descritos no tipo de dado SERIAL_STATUS, possíveis são: - NO_ERROR - ILLEGAL_SERIAL_PORT - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - NOT_CONFIGURED
PARAMETERS	SERIAL_PARAMETERS	Essa estrutura recebe os parâmetros de configuração da porta serial desejada, conforme descrito no tipo de dado SERIAL_PARAMETERS.

Tabela 131: 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 134: 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 132: Parâmetros de Entrada SERIAL_GET_CTRL

Parâmetros de saída	Tipo	Descrição
DONE	BOOL	Essa variável é verdadeira quando o bloco funcional for executado por completo, caso contrário, ela é falsa.
EXEC	BOOL	Essa variável é verdadeira enquanto o bloco funcional estiver sendo executado, caso contrário, ela é falsa.
ERROR	BOOL	Essa variável é verdadeira quando o bloco funcional concluir a sua execução com algum erro, caso contrário, ela é falsa. Está vinculada à variável DONE, pois seu estado é exibido após a conclusão do bloco funcional.
STATUS	SERIAL_STATUS	Caso a variável ERROR seja verdadeira, a estrutura STATUS exibirá o erro encontrado na execução do bloco. Os estados, já descritos no tipo de dado SERIAL_STATUS, possíveis são: - NO_ERROR - ILLEGAL_SERIAL_PORT - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - FB_GET_CTRL_NOT_ALLOWED - NOT_CONFIGURED
CTS_VALUE	BOOL	Valor lido no sinal de controle CTS.
DSR_VALUE	BOOL	Valor lido no sinal de controle DSR.
DCD_VALUE	BOOL	Valor lido no sinal de controle DCD.

Tabela 133: 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 135: 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 134: 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 135: 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 136: 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 136: 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 137: 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 137: Bloco para Ler Valores do Buffer de Recepção

Parâmetros de entrada	Tipo	Descrição
REQUEST	BOOL	Essa variável, quando verdadeira, habilita o uso do bloco funcional.
PORT	SERIAL_PORT	Seleciona a porta serial, conforme descrito no tipo de dado SERIAL_PORT.
RX_BUFFER_ POINTER	POINTER TO BYTE	Ponteiro de um array de bytes para receber os valores do buffer.
RX_BUFFER_ LENGTH	UINT	Especifica o número de caracteres esperados no array de bytes. Caso estejam disponíveis mais bytes do que o esperado, somente a quantidade esperada vai ser lida no array de bytes, sendo que os demais serão deixados na fila RX (tamanho máximo igual a 1024 caracteres).

Parâmetros de entrada	Tipo	Descrição
RX_TIMEOUT	UINT	Especifica o time-out para receber a quantidade de caracteres esperados. Caso ele seja menor do que a quantidade de bytes a receber, será indicado RX_TIMEOUT_ERROR no parâmetro de saída STATUS. Quando o valor especificado, em ms, for igual a zero, a função irá retornar os dados presentes no buffer.

Tabela 138: 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 139: Parâmetros de Saída SERIAL_RX

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

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

5.12.1.7. SERIAL_RX_EXTENDED

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

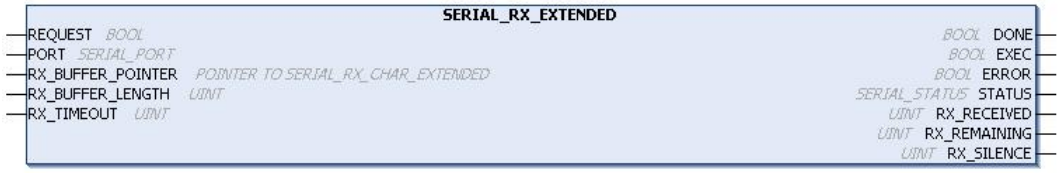


Figura 138: Bloco para Leitura do Buffer de Recepção

Parâmetros de entrada	Tipo	Descrição
REQUEST	BOOL	Essa variável, quando verdadeira, habilita o uso do bloco funcional.
PORT	SERIAL_PORT	Seleciona a porta serial, conforme descrito no tipo de dado SERIAL_PORT.
RX_BUFFER_ POINTER	POINTER TO SERIAL_RX_CHAR _EXTENDED	Ponteiro de um array de SERIAL_RX_CHAR_EXTENDED para receber os valores do buffer.

Parâmetros de entrada	Tipo	Descrição
RX_BUFFER_ LENGTH	UINT	Especifica o número de caracteres esperados no array de SERIAL_RX_CHAR_EXTENDED. Caso estejam disponíveis mais bytes do que o esperado, somente a quantidade esperada vai ser lida no array, sendo que os demais serão deixados na fila RX (tamanho máximo igual a 1024 caracteres).
RX_TIMEOUT	UINT	Especifica o time-out para receber a quantidade de caracteres esperados. Caso ele seja menor do que a quantidade de bytes a receber, será indicado RX_TIMEOUT_ERROR no parâmetro de saída STATUS. Quando o valor especificado, em ms, for igual a zero, a função irá retornar os dados presentes no buffer.

Tabela 140: 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.

Parâmetros de saída	Tipo	Descrição
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 μs. 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 141: 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 139: 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 142: 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 143: 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 time-out (quando erros são gerados).



Figura 140: 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 modens.
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 144: 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 145: 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. Timer Retain

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 do controlador físico 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).

5.12.2.1. 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 141 representa o bloco *TOF_RET* e a Figura 142 mostra o comportamento gráfico do mesmo.



Figura 141: 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 146: 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 147: Parâmetros de Saída TOF_RET

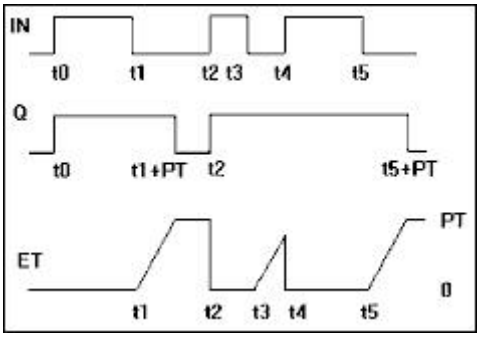


Figura 142: Comportamento Gráfico do Bloco Funcional TOF_RET

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
```

5.12.2.2. 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 143 representa o bloco *TON_RET* e a Figura 144 mostra o comportamento gráfico do mesmo.



Figura 143: Bloco Funcional TON_RET

Parâmetros de entrada	Tipo	Descrição
IN	BOOL	Essa variável, quando recebe uma borda de subida, habilita a contagem do bloco funcional.
PT	TIME	Essa variável especifica o limite de contagem do bloco funcional (tempo de atraso).

Tabela 148: 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 149: Parâmetros de Saída TON_RET

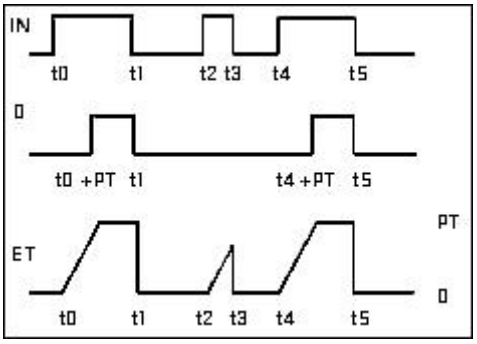


Figura 144: Comportamento Gráfico do Bloco Funcional TON_RET

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
```

5.12.2.3. 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 145 representa o bloco *TP_RET* e a Figura 146 mostra o comportamento gráfico do mesmo.



Figura 145: 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 150: 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 151: Parâmetros de Saída TP_RET

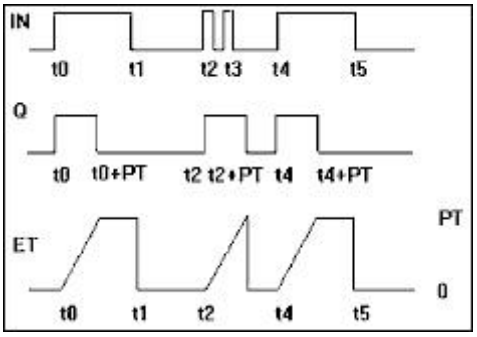


Figura 146: Comportamento Gráfico do Bloco Funcional TP_RET

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. 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.

5.14. Página Web de Sistema

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

5.14.1. Informações Gerais

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



Figura 147: Tela Inicial

5.14.2. 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.

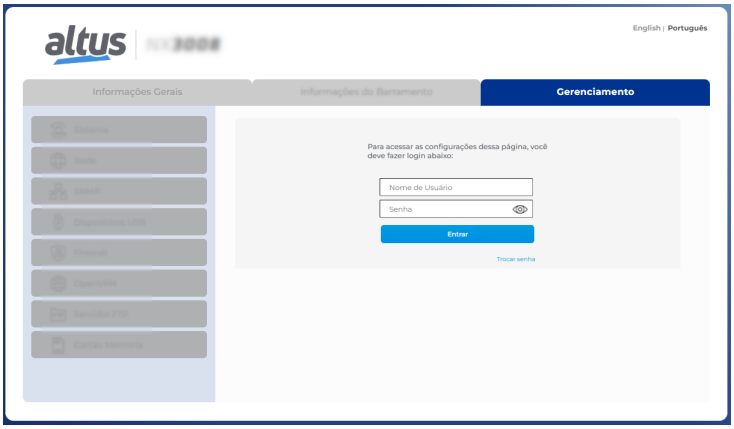


Figura 148: Acesso a aba de Gerenciamento

5.14.2.1. Sistema

Na seção de *Sistema* da aba *Gerenciamento* é possível realizar a atualização de firmware do controlador. 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 128 kbps.

5.14.2.1.1. 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.

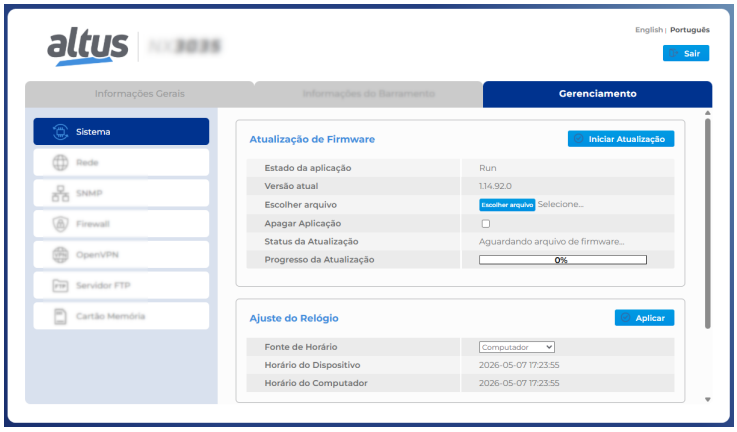


Figura 149: Atualização de Firmware

5.14.2.1.2. 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:

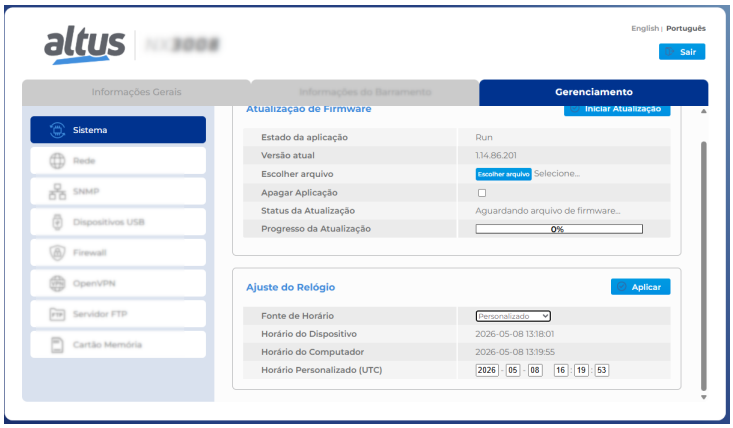


Figura 150: Ajuste do Relógio

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

1. **Horário do Computador (UTC)**
   No modo de horário do computador, o usuário pode aplicar o horário configurado em seu computador em UTC para o seu dispositivo. Para isso, deve-se selecionar no item “Fonte de Horário” a opção “Computador”. Após clicar no botão “Aplicar”, é necessário à validação das credenciais do dispositivo, então o controlador 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)”.

5.14.2.1.3. 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.

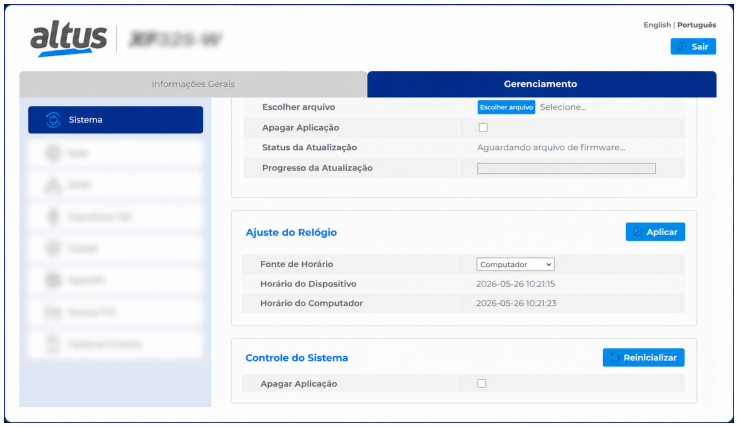


Figura 151: Controle do Sistema

5.14.2.2. 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.

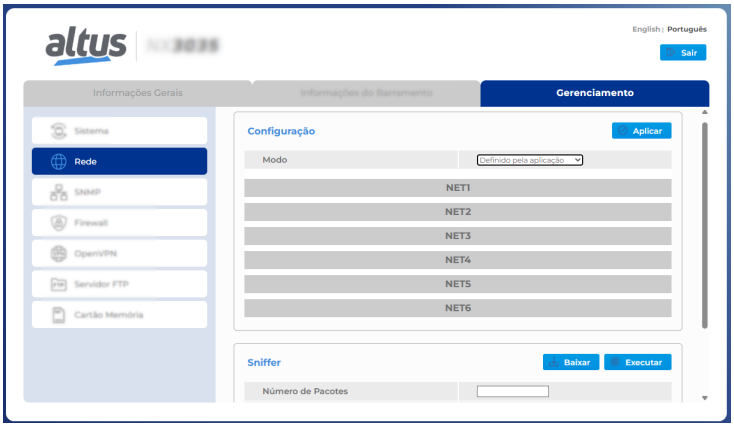


Figura 152: Página Web de Rede

5.14.2.2.1. 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.

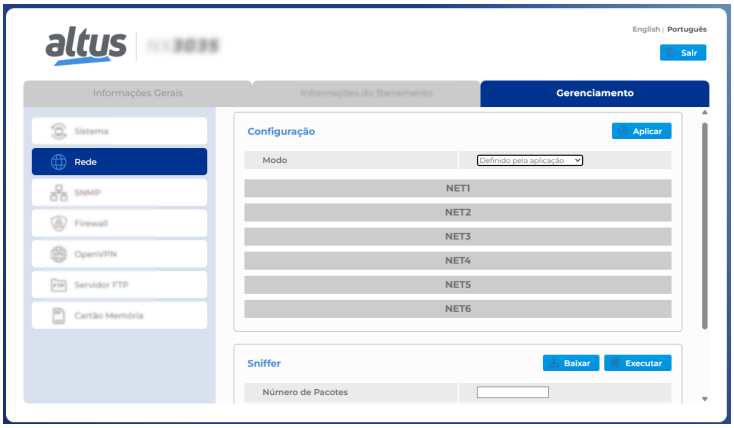


Figura 153: Tabela de Interfaces - Modo Aplicação

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.

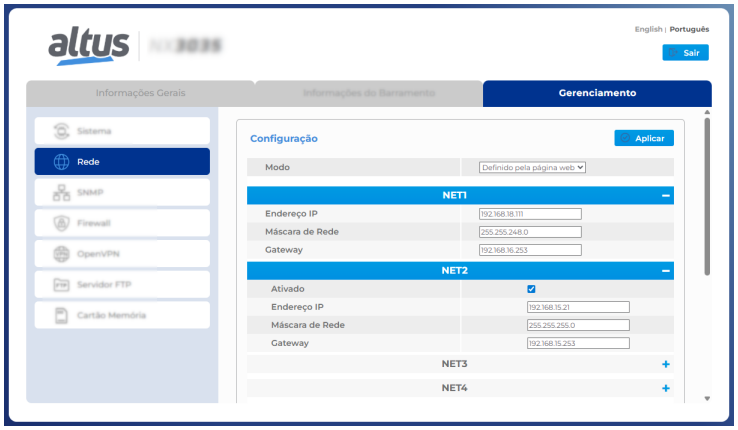


Figura 154: Tabela de Interfaces - Modo Web

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

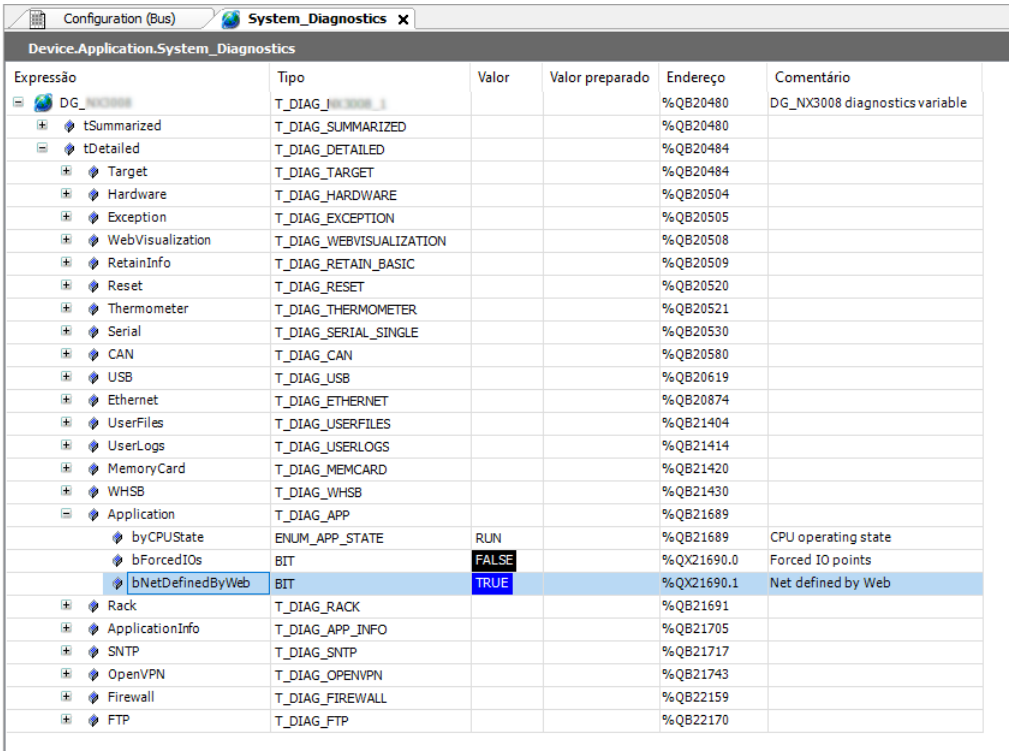
Ao realizar alterações de rede no controlador as interfaces serão reiniciadas, o que pode causar queda na comunicação. Especialmente ao ser alterado o valor do endereço IP.

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_	T_DIAG_			%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 155: Diagnóstico - IP definido pela Página Web

5.14.2.2.2. 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 modens 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.

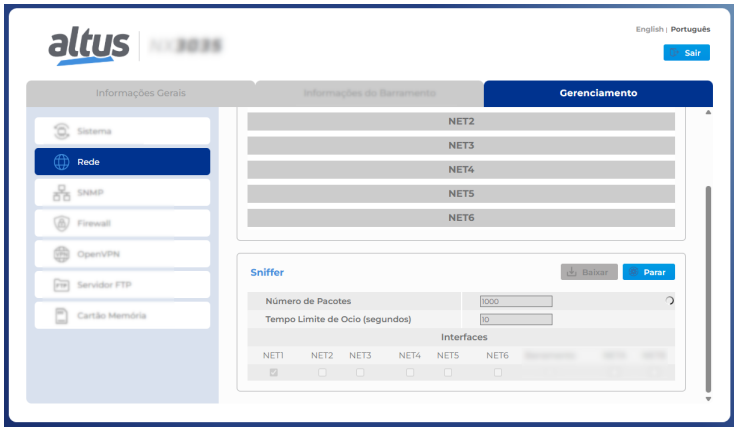


Figura 156: Sniffer de Rede Executando

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

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

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

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

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

5.14.2.3. 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.

5.14.2.3.1. SNMP nas UCPs Nexto

Os controladores do Sistema Nexto se comportam como agentes na comunicação SNMP. As informações disponibilizadas através do SNMP não podem ser manipuladas ou acessadas através da aplicação do usuário, sendo necessário um gerente SNMP externo para efetuar o acesso, com suporte aos protocolos SNMPv1, SNMPv2c e SNMPv3, além de suporte a MIB-II, onde objetos são descritos na RFC-1213. A tabela abaixo contém os objetos disponíveis.

OID	Nome	Descrição
1.3.6.1.2.1.1	System	Contém nome, descrição, localização e outras informações de identificação do equipamento.
1.3.6.1.2.1.2	Interfaces	Contém informações das interfaces de rede do equipamento.
1.3.6.1.2.1.3	At	Contém informações das últimas conexões requisitadas ao agente.
1.3.6.1.2.1.4	IP	Contém estatísticas de conexões utilizando protocolo IP.
1.3.6.1.2.1.5	ICMP	Contém estatísticas para o protocolo ICMP.
1.3.6.1.2.1.6	TCP	Contém estatísticas para o protocolo TCP.
1.3.6.1.2.1.7	UDP	Contém estatísticas para o protocolo UDP.
1.3.6.1.2.1.11	SNMP	Contém estatísticas para o protocolo SNMP.

Tabela 152: Objetos da MIB-II - Agente SNMP do Sistema 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.

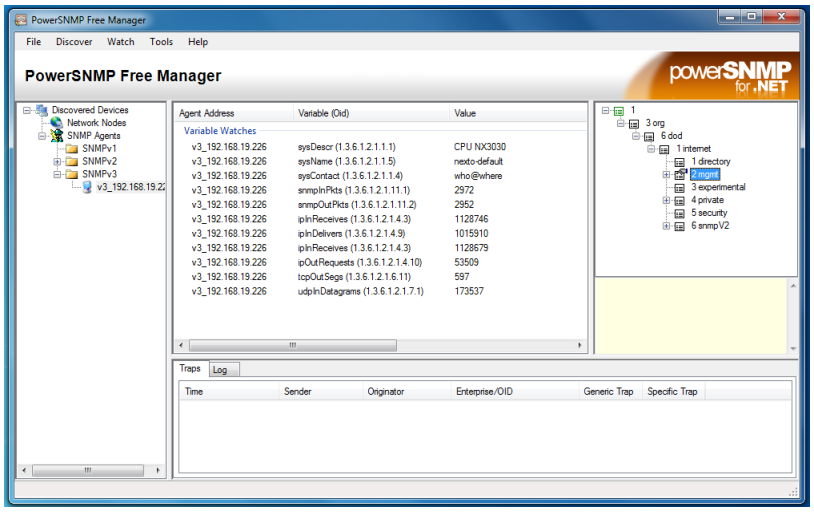


Figura 157: Exemplo de Gerente SNMP

Para SNMPv3, na qual existe autenticação de usuário e senha para requisições via protocolo SNMP, é fornecido um usuário padrão descrito na seção Usuário e Comunidades SNMP.

Caso o usuário deseje desativar o serviço, alterar o usuário SNMPv3 ou as comunidades para SNMPv1/v2c pré-definidos, será necessário acessar a Página Web de Sistema do controlador. Para mais detalhes consulte a seção Configuração SNMP.

5.14.2.3.2. Configuração SNMP

As configurações do SNMP podem ser alteradas através da Página Web de Sistema, 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.

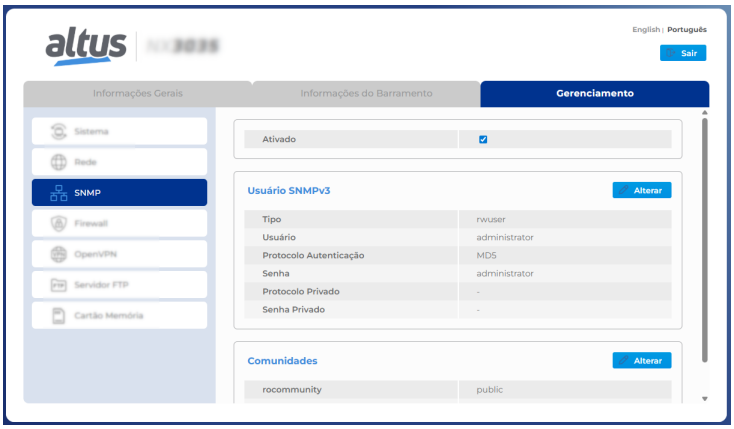


Figura 158: Tela de Configuração e Status SNMP

5.14.2.3.3. Usuário e Comunidades SNMP

Para acesso ao SNMPv1/v2c, existem duas comunidades, conforme tabela abaixo.

Comunidades	String Padrão	Tipo
rocommunity	Public	Somente leitura
rwcommunity	Private	Leitura e escrita

Tabela 153: 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 154: 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 155: Limites para Configurações SNMP

5.14.2.4. Dispositivos USB

A porta Host USB presente nos controladores do Sistema 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:

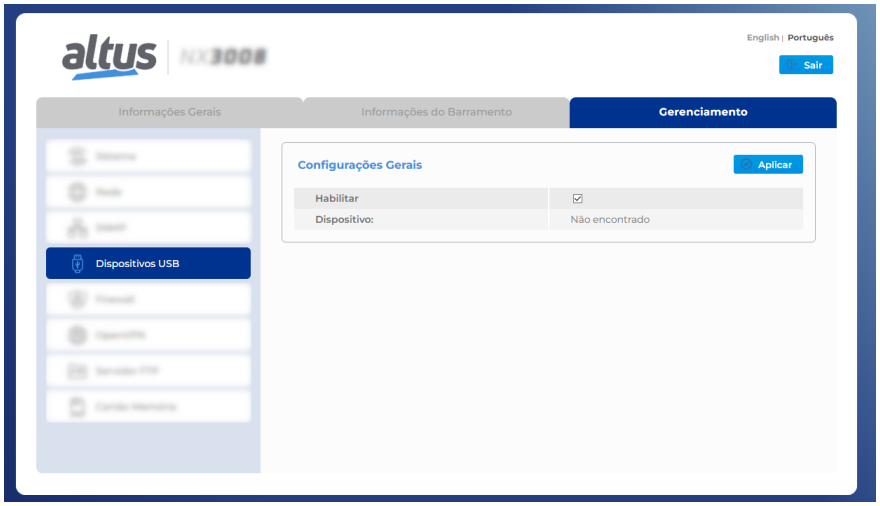


Figura 159: Seção de dispositivos USB

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 160: Dispositivos USB - Desconhecido

ATENÇÃO

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

5.14.2.4.1. 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:

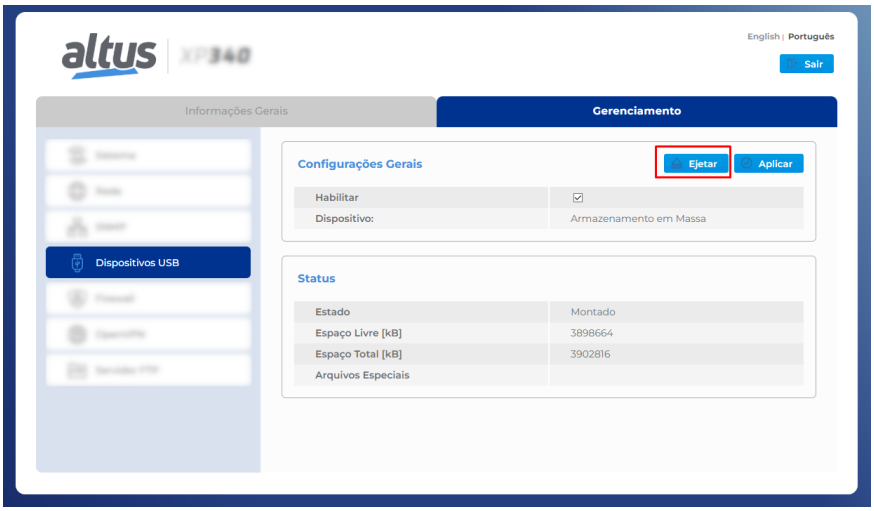


Figura 161: Informações sobre o dispositivo de armazenamento em massa

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:

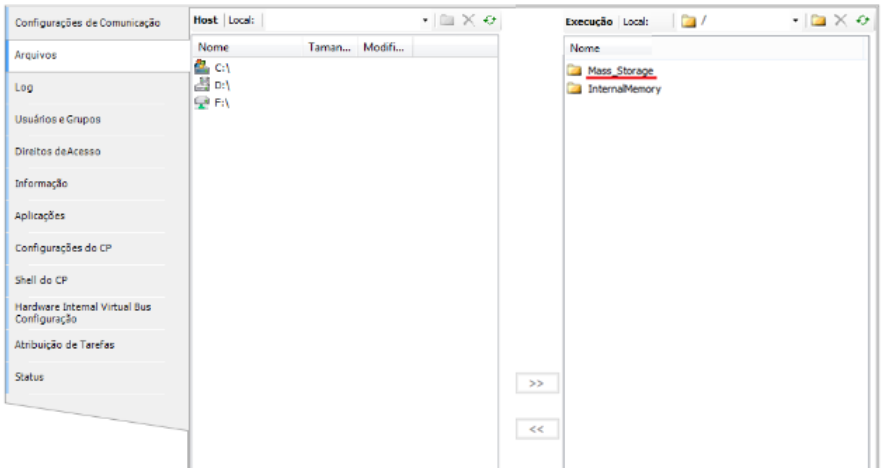


Figura 162: Pasta de armazenamento em massa USB

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.

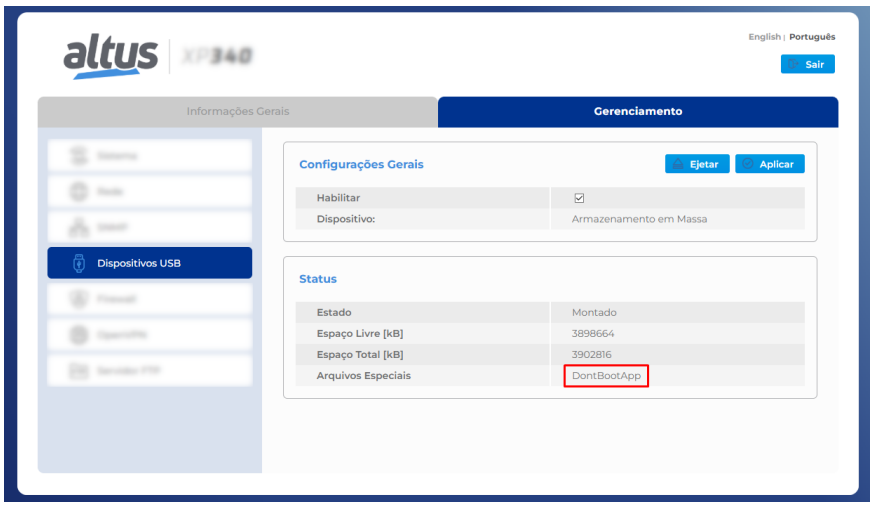


Figura 163: DontBootApp no dispositivo de armazenamento em massa

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:

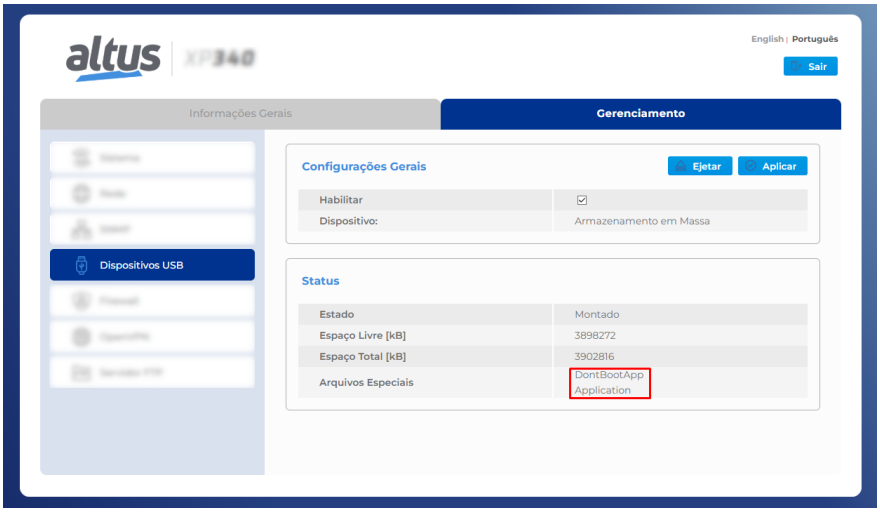


Figura 164: Aplicação no dispositivo de armazenamento em massa

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

5.14.2.4.2. Conversor USB para RS-232

O Sistema 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 156: 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:

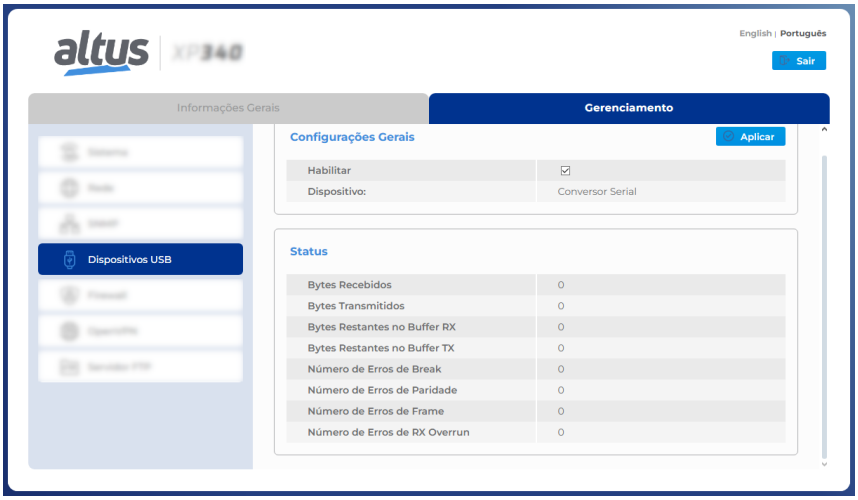


Figura 165: Dispositivos USB - Conversor Serial

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 *COM 10* e não terá uma representação na visualização da árvore do projeto. A partir deste ponto, essa porta pode ser usada para comunicação usando as funções *NextoSerial* semelhantes às portas nativas. Para esse tipo de porta, a configuração do handshake é limitada apenas a *RS232_MANUAL* (deve ser considerada ao configurar a porta com a função *SERIAL_CFG*).

5.14.2.4.3. 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 157: 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:

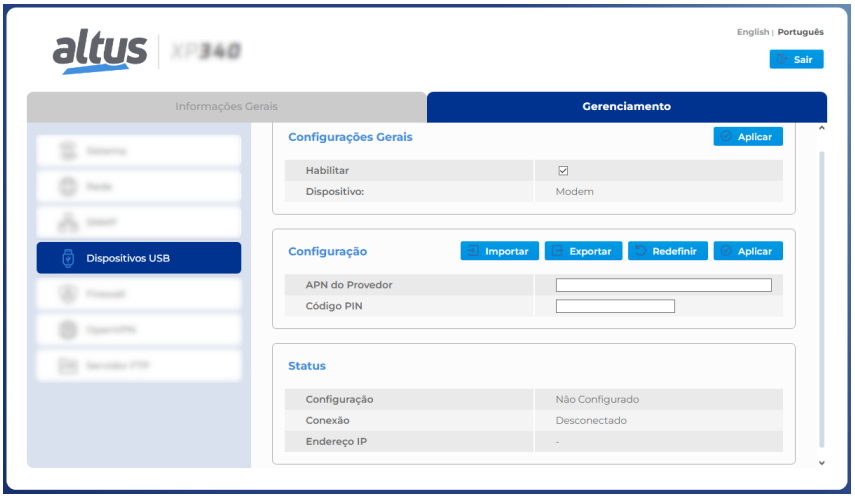


Figura 166: Configuração da Página do Modem USB

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

ATENÇÃO

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

Esta seção de configuração contém os seguintes botões:

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

ATENÇÃO

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

O princípio de funcionamento da funcionalidade do modem USB depende do tipo de modem. Em casos específicos, uma operadora pode fornecer um endereço IP acessível publicamente para o cartão SIM. Nessas situações, o acesso externo ao controlador pode se tornar possível. No entanto, essa configuração é incomum e pode expor o controlador diretamente à internet, o que pode representar um risco de segurança cibernética.
Para modems bridge, assim que o dispositivo é devidamente detectado e os parâmetros do cartão SIM são configurados, o controlador inicia um processo em segundo plano que gerencia continuamente a conexão do modem para mantê-lo conectado à internet. Se a conexão for interrompida por qualquer motivo (como baixa qualidade de sinal ou queda da operadora), o controlador tenta se reconectar automaticamente. O status desse processo pode ser monitorado através do campo *Status* de Conexão.
Um aspecto importante a ser considerado é que, quando um modem USB é configurado, o sistema do controlador o define como o gateway padrão para a comunicação Ethernet. Isso significa que, se o controlador estiver conectado simultaneamente a outra interface de rede local que também possua acesso à internet, as comunicações destinadas a endereços IP externos serão roteadas através do modem USB em vez da interface Ethernet. Quando o modem USB é removido, a interface Ethernet configurada anteriormente torna-se o gateway padrão novamente. A sequência de configuração e operação pode ser resumida nas seguintes etapas:

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

5.14.2.4.4. 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 158: 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:

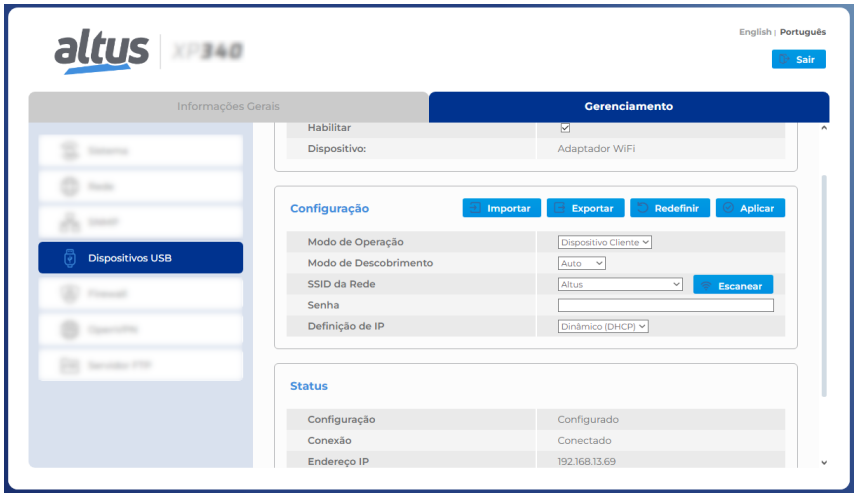


Figura 167: Página do adaptador USB Wi-Fi

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

ATENÇÃO

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

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

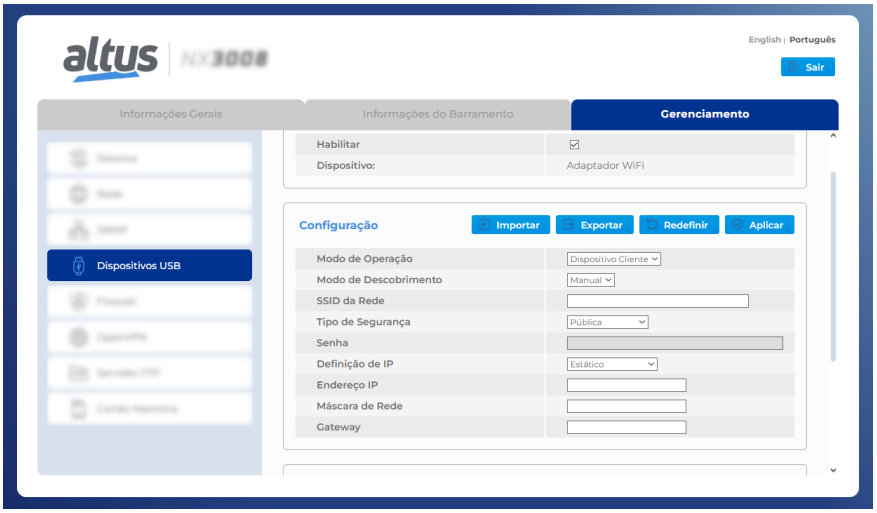


Figura 168: Configuração manual do adaptador USB Wi-Fi

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:

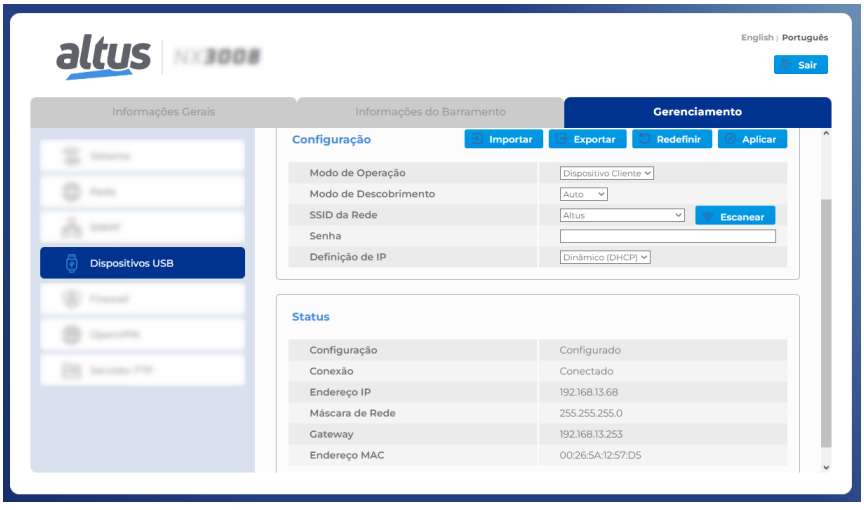


Figura 169: Adaptador USB Wi-Fi conectado a uma rede

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

5.14.2.4.5. Adaptadores Ethernet

Este controlador permite implementar uma outra 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 159: 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:

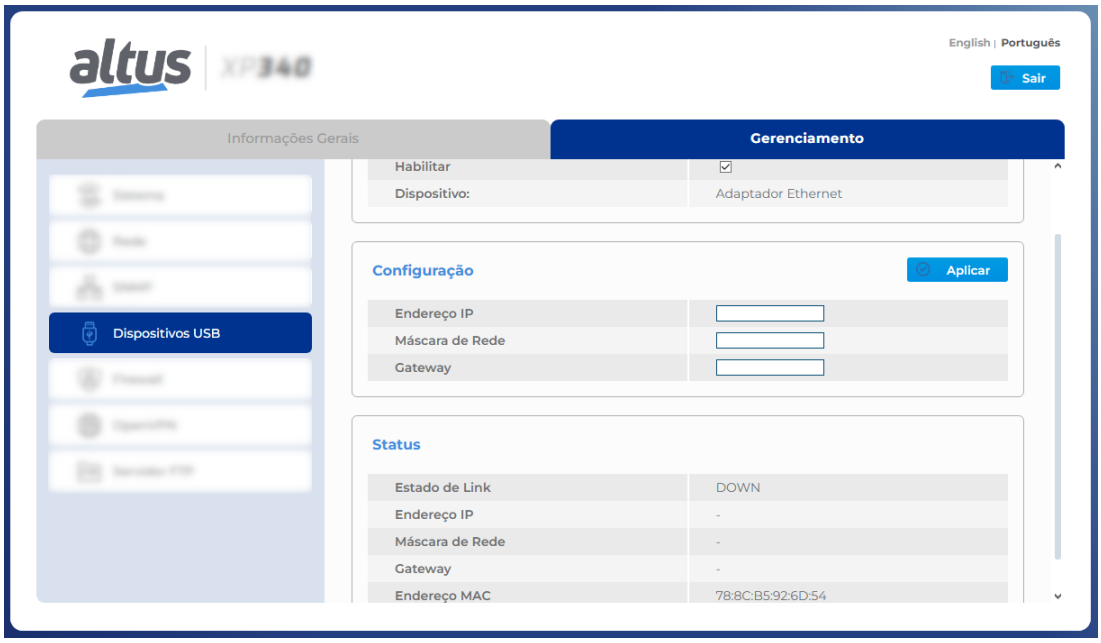


Figura 170: Página do adaptador USB Ethernet sem configuração aplicada

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 NET 1 e NET 2.

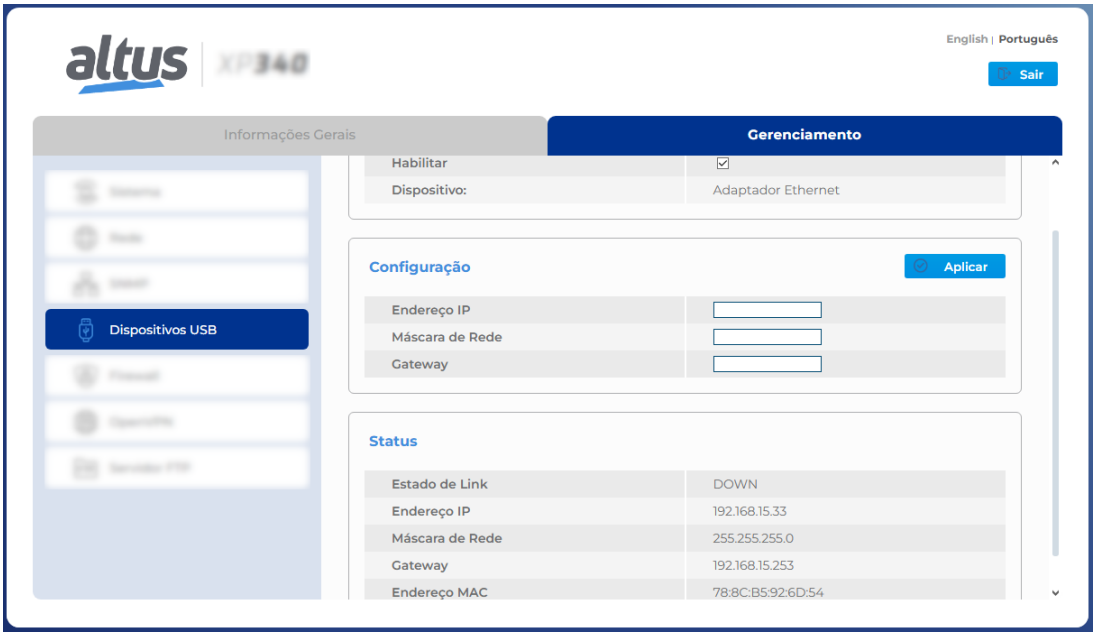


Figura 171: Página do adaptador USB Ethernet com configuração aplicada

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:

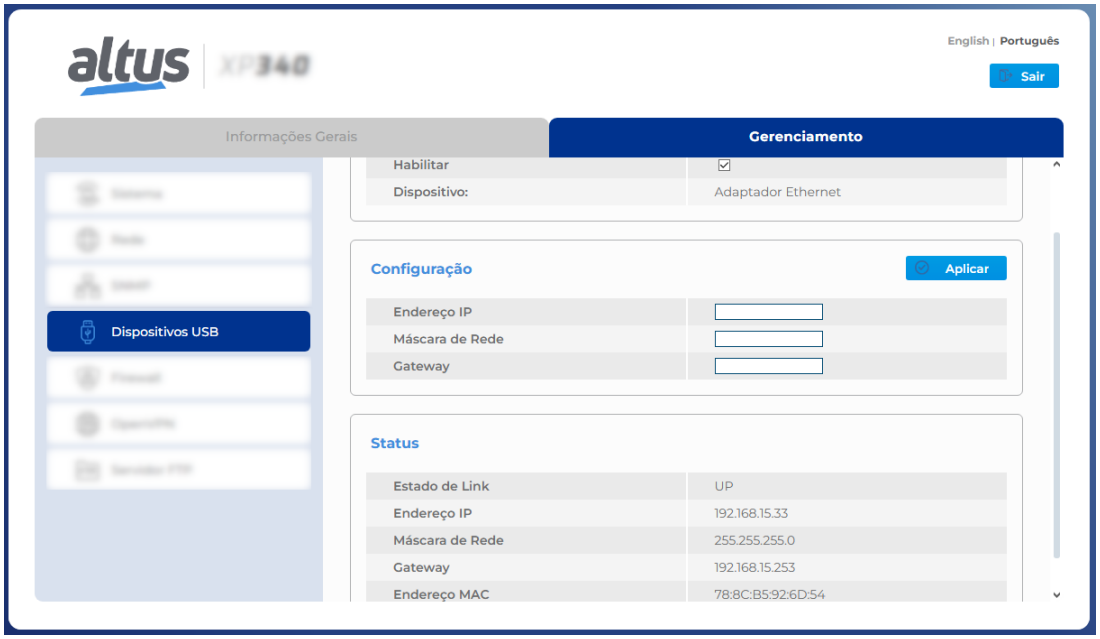


Figura 172: Página do adaptador USB Ethernet com configuração aplicada e link UP

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.

5.14.2.4.6. 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*.

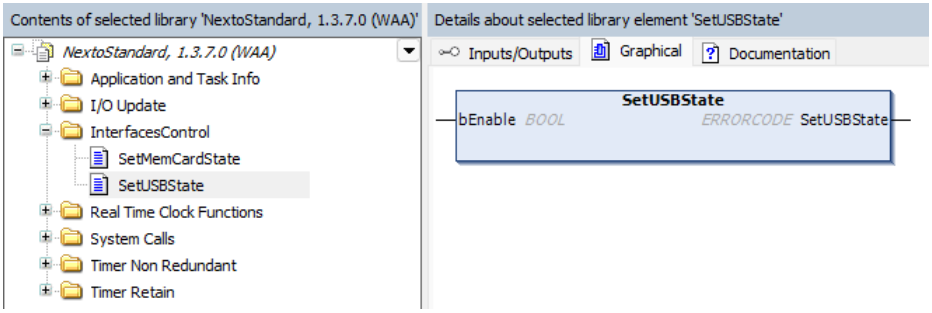


Figura 173: Informações de SetUSBState no Library Manager

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 *ImportFunctionNotFound* caso a função não seja suportada. A imagem abaixo mostra um exemplo básico de declaração de variáveis e chamada da função.

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

```
stErrorCode := NextoStandard.SetUSBState(bEnable := bSetUSBInterfaceState);
```

ATENÇÃO

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

5.14.2.5. 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 botão pressionado, o Firewall será desabilitado e as regras não serão aplicadas.

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

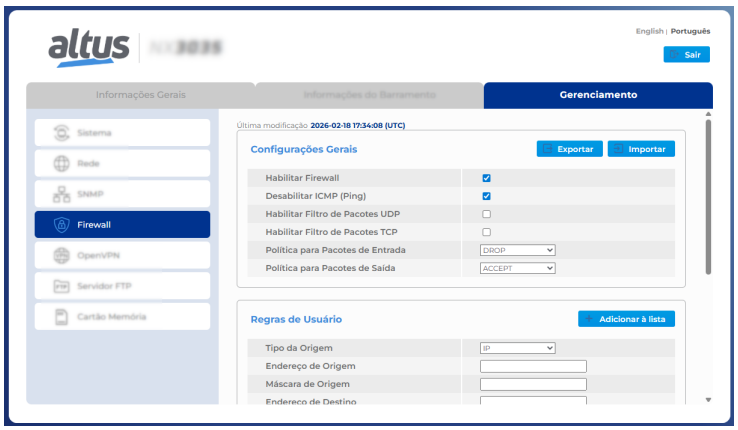


Figura 174: Tela de Configuração do Firewall

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

5.14.2.5.1. Configurações Gerais

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

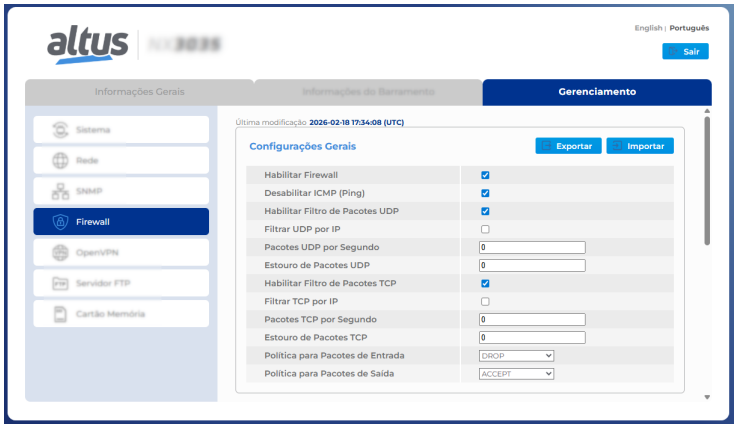


Figura 175: Tabela de Configurações Gerais do Firewall

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

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

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

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

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

ATENÇÃO

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

As configurações desta tabela são aplicadas com o botão de *Aplicar* que aparece na figura 177.

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 politica configurada para estes pacotes, seja ela de *Accept* ou *Drop*. Assim, se for configurado uma politica 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 disponiveis os botões *Exportar* e *Importar*. Ao utilizar o botão *Exportar* é possivel 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 desse ou de outro controlador 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 ao controlador, 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* do mesmo controlador, ou de outro do mesmo modelo, ao realizar a importação de configurações de Firewall.

5.14.2.5.2. 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.

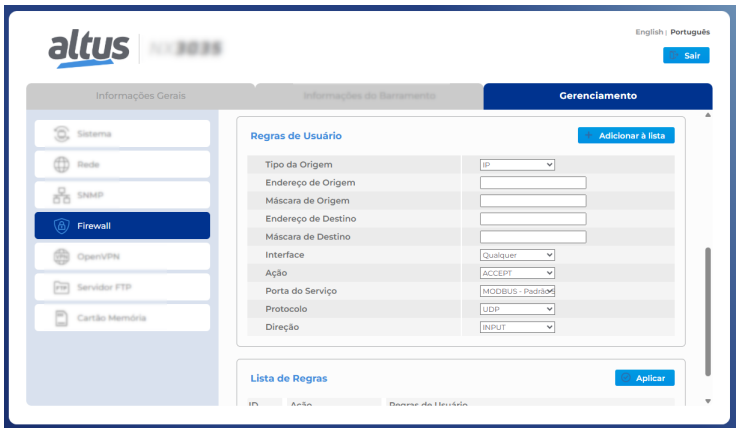


Figura 176: Tabela de Configuração de Regras de Usuário do Firewall

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 NET 1. Sendo assim, caso o protocolo HTTP seja escolhido, os campos *NET 1* 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:

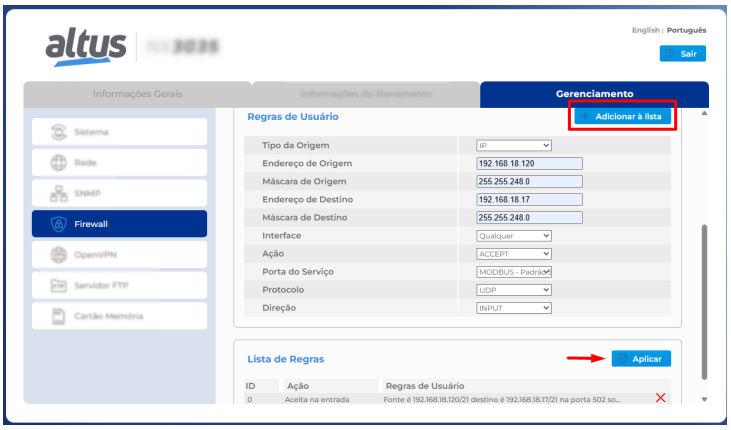


Figura 177: Tabela de Aplicação de Regras de Usuário do Firewall

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

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

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

ATENÇÃO

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

5.14.2.5.3. 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.

5.14.2.6. 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. Conforme é mostrado na figura abaixo.

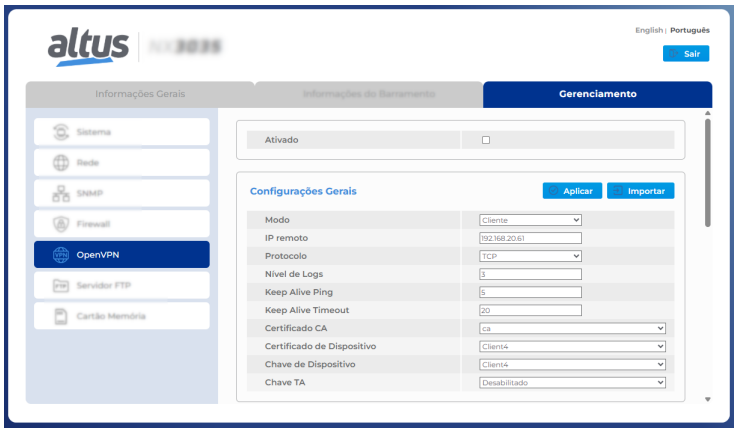


Figura 178: Tela de Configuração de OpenVPN

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 178 no canto superior direito da página. Ao clicar sobre este botão, é aberta uma janela, do explorador de arquivos, para que seja feita a seleção de um arquivo de configuração. Devem ser selecionados os arquivos com extensão *conf* ou *ovpn*. Ao selecionar um arquivo, o seu conteúdo será lido e os parâmetros de configuração que estiverem presentes preencherão os seus respectivos campos de configuração na página web.
  Para que os parâmetros, do arquivo, sejam interpretados corretamente, eles devem seguir o padrão de sintaxe de arquivos de configuração do OpenVPN.
  Caso existam arquivos de segurança, certificados ou chaves, escritos no arquivo de configuração, junto aos demais parâmetros, eles serão lidos e separados em arquivos distintos dentro do controlador para serem utilizados.

  ATENÇÃO

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

5.14.2.6.1. Configuração do OpenVPN

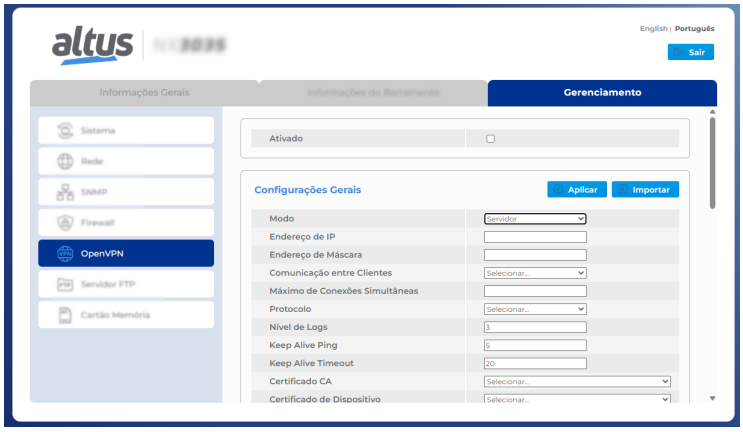


Figura 179: Tabela de Configurações de Servidor OpenVPN

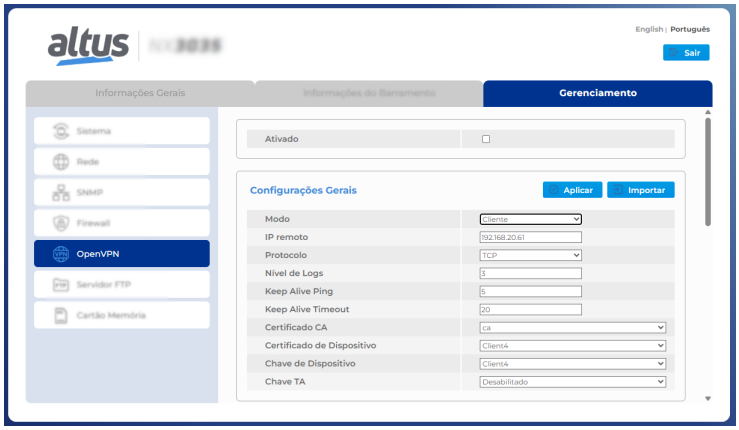


Figura 180: Tabela de Configurações de Cliente OpenVPN

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 180, e a de servidor, figura 179, 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 time-out 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 179, 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 OpenPVN como servidor, será exibido uma tabela, normalmente oculta, que permite a configuração de redes privadas que podem estar abaixo do servidor e de cada cliente.



Figura 181: 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 subrede *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 subrede *70*, este pacote será roteado e encaminhado pelo túnel de VPN. O mesmo comportamento ocorre com o cliente *client2*, porém, com as subredes trocadas, pois, abaixo deste cliente está a subrede *70* e ele encaminhará para o túnel VPN os pacotes com a subrede *80*.

Veja na figura a seguir um exemplo de arquitetura:

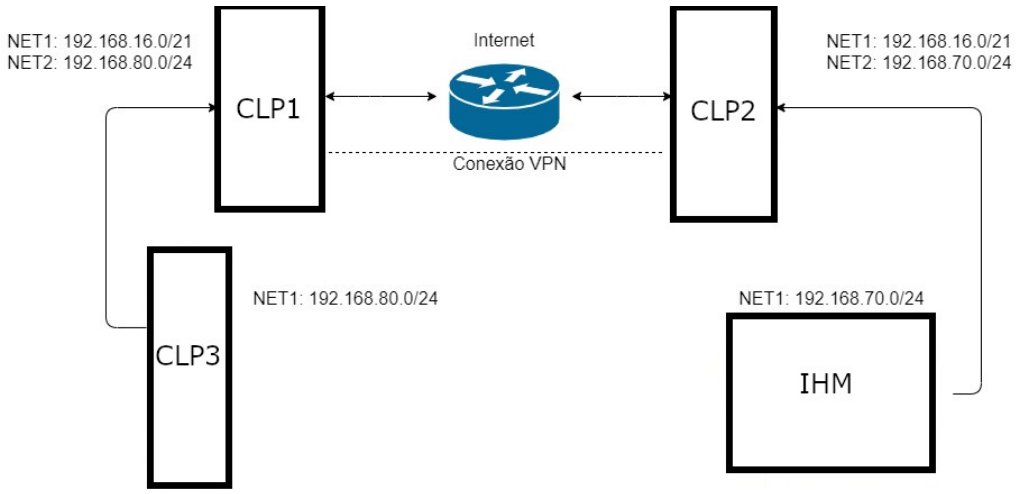


Figura 182: Exemplo de Arquitetura com Redes Privadas

Na Figura 182, 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 180. 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 180. 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 180. Quando as configurações são aplicadas e a VPN está habilitada, a seção *Firewall* irá realizar um *scroll* automático até a tabela de *status* do OpenVPN, exibida na seção Status.

5.14.2.6.2. 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:

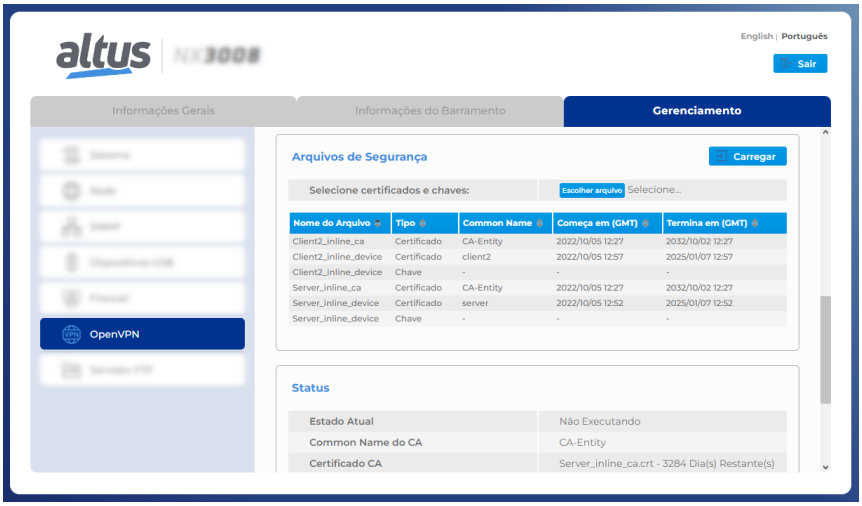


Figura 183: Tabela de Arquivos de Segurança OpenVPN

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

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

ATENÇÃO

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

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

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

5.14.2.6.3. 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.

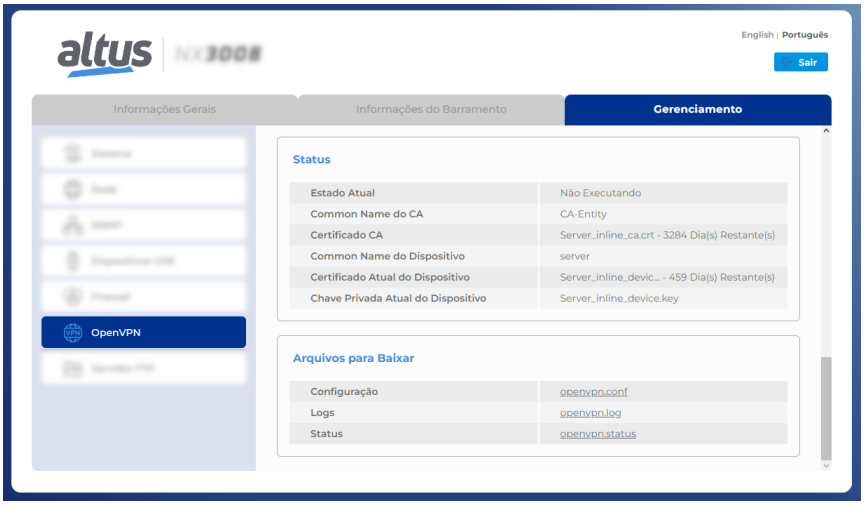


Figura 184: Tabela de Status OpenVPN com Funcionalidade Desativada

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.



Figura 185: Tabela de Status OpenVPN com Funcionalidade Ativada

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

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

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

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

5.14.2.6.4. *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.

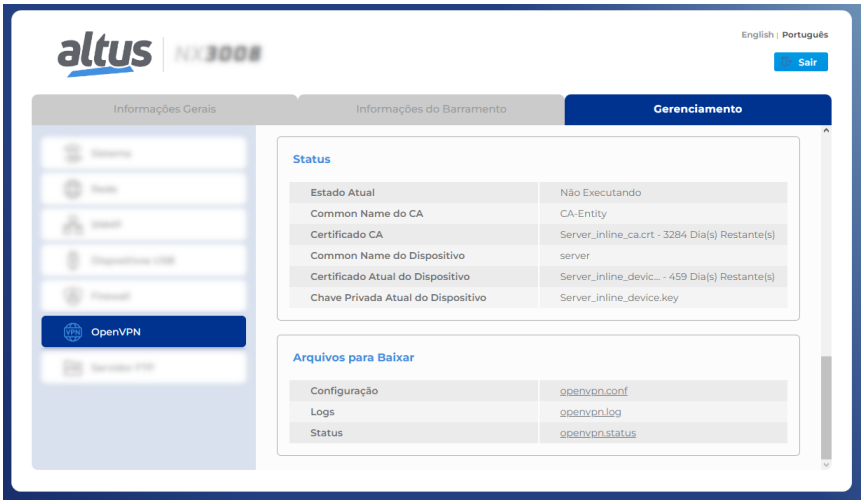


Figura 186: Seção de Downloads OpenVPN

5.14.2.6.5. *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 187: 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**

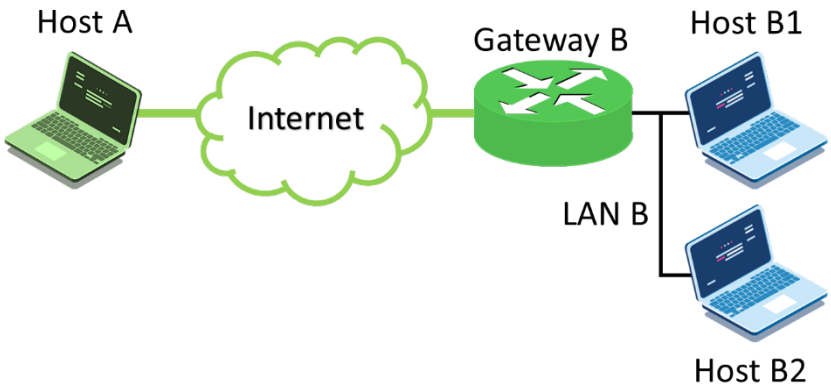


Figura 188: Exemplo de topologia Host-to-Site

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 Redes Privadas 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**

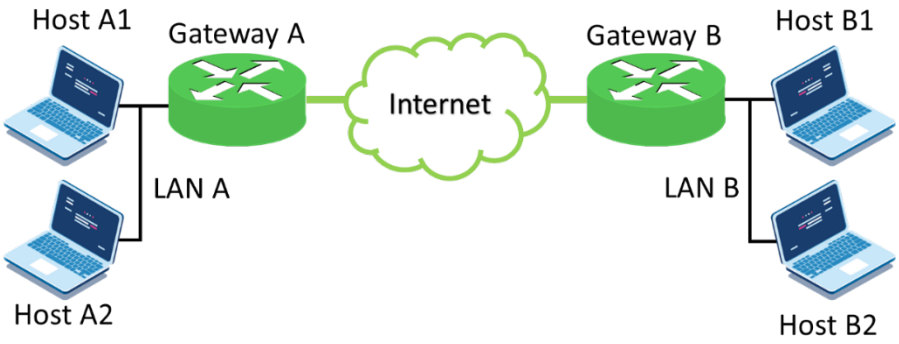


Figura 189: Exemplo de topologia Site-to-Site

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 Redes Privadas para verificar como criar regras de redes privadas.

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

5.14.2.7. Servidor FTP

O FTP (File Transfer Protocol) é um protocolo que permite a transferência de arquivos entre dispositivos. Opera no modelo cliente-servidor, onde o controlador 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 do controlador, 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 do controlador.

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.

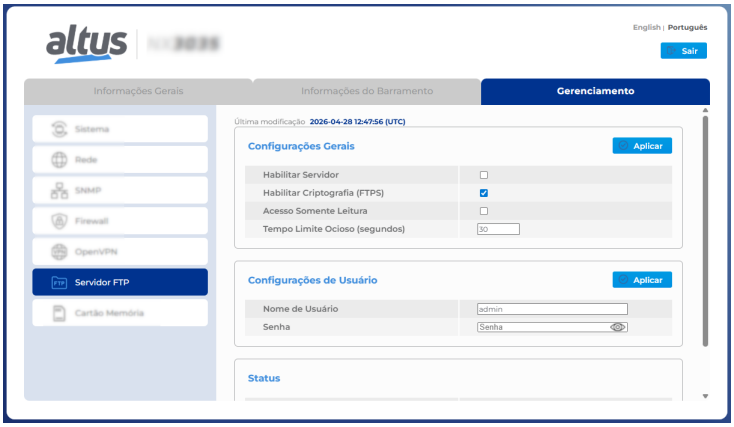


Figura 190: Tela de Configuração do Servidor FTP

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

ATENÇÃO

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

5.14.2.7.1. 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, o controlador gera um certificado *self-signed* para garantir a comunicação utilizando o protocolo SSL (Secure Sockets Layer).
   Quando o Cliente FTP realizar a conexão autenticada, o certificado e suas respectivas informações serão exibidos, permitindo estabelecer a comunicação FTPS Explícita (FTP sobre SSL), através da mesma porta 21.

   ATENÇÃO

   O certificado FTPES tem uma validade de um ano a partir da data de criação. Para gerar um novo certificado, basta aplicar novamente a configuração.
3. **Acesso Somente Leitura**
   Permite habilitar e desabilitar a limitação do acesso de escrita e leitura no Servidor. Por padrão, o parâmetro é habilitado.
   Quando o parâmetro está habilitado, o Cliente FTP tem acesso somente a leitura, sem a possibilidade de adicionar ou remover qualquer arquivo do Servidor FTP. Neste caso, a única operação permitida é o upload de arquivos, ou seja, transferência do Servidor para o Cliente. Já quando o parâmetro está desabilitado, é possível realizar todas as operações.
4. **Tempo Limite Ocioso (Segundos)**
   Define o tempo máximo em que a conexão será mantida antes do Servidor FTP fechá-la por inatividade. Isto é, após aberta a conexão com um Cliente FTP, se não houver nenhuma atividade, após passado o Tempo Limite Ocioso, a conexão com o Cliente é fechada pelo Servidor.
   O parâmetro pode ser configurado com valores entre 10 e 60 segundos sendo, por padrão, 60 segundos.

   ATENÇÃO

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

5.14.2.7.2. 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 multiplas conexões.
2. **Senha**
   Gerencia a chave de autenticação para que o Cliente FTP possa se conectar com o Servidor FTP.
   Não há recuperação de senha. Caso esta tenha sido perdida, é necessário adicionar uma nova configuração de usuário.

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

Tabela 160: Configurações de Usuário FTP

5.14.2.7.3. 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 simultâneamente. Alguns Clientes FTP, como por exemplo o *Filezilla*, utilizam um recurso chamado Transferência de Arquivos Multithread, para melhorar a eficiência e a velocidade da transferência. Este recurso, permite que o Cliente FTP abra mais de uma conexão para a transferência de um arquivo. Consequentemente, quando utilizado um Cliente FTP deste tipo, apenas um Cliente já soma duas conexões ativas no Servidor.
   Já outros Clientes FTP como por exemplo através de terminal de comandos, ocupam apenas uma conexão ativa, assim possibilitando a conexão simultânea de dois Clientes FTP com o Servidor.

5.14.2.8. Cartão de Memória

Entre outras funcionalidades, este modelo de UCP do Sistema Nexto possibilita ao usuário a utilização de um cartão de memória, definido conforme as características da seção Interface do Cartão de Memória, o qual serve para armazenar arquivos.

Quando o cartão for inserido na UCP e estiver com sistema de arquivos diferente de FAT32, ela automaticamente identifica e pergunta ao usuário se ele deseja formatar. Em caso negativo, ele não poderá utilizar o cartão (o cartão não será montado e o visor não indicará a presença do cartão). Caso seja selecionada a opção de formatação, a UCP irá levar alguns minutos, dependendo do tempo de ciclo (execução) da aplicação rodando na UCP, para executar a operação. Assim que o cartão de memória for montado, a UCP irá ler informações gerais do mesmo, deixando o acesso ao cartão de memória mais lento nos primeiros minutos. Esse procedimento é feito quando o cartão é inserido, após a reinicialização da UCP ou através da página web do dispositvo.

ATENÇÃO

Recomenda-se realizar o procedimento de formatação do cartão de memória diretamente na UCP para evitar possíveis problemas de utilização, aumento do tempo de montagem ou até mesmo funcionamento incorreto.
Não é recomendado remover o cartão de memória ou desenergizar a UCP durante a formatação ou durante a transferência de arquivos, pois pode causar a perda de dados bem como danos irreversíveis ao cartão.

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

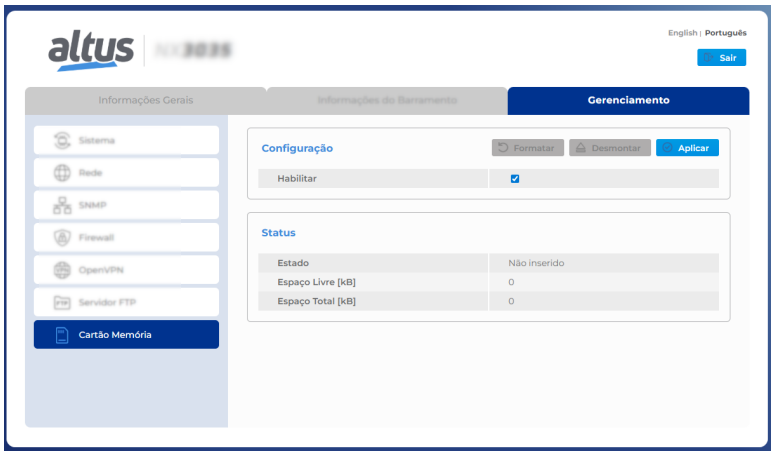


Figura 191: Cartão de Memória Página Inicial

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

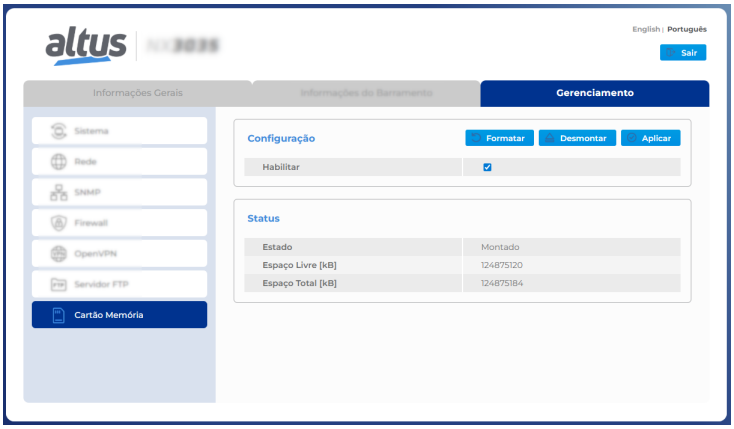


Figura 192: Cartão de Memória com Dispositivo Montado

ATENÇÃO

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

5.14.2.8.1. Configuração

1. **Formato Não Suportado**
   Se o dispositivo tiver um sistema de arquivos não suportado pela UCP, o estado *"Formato não suportado"* aparecerá quando o dispositivo for inserido, conforme mostrado na figura abaixo. Nesse caso, o dispositivo não pode ser usado até que seja formatado com um sistema de arquivos suportado.

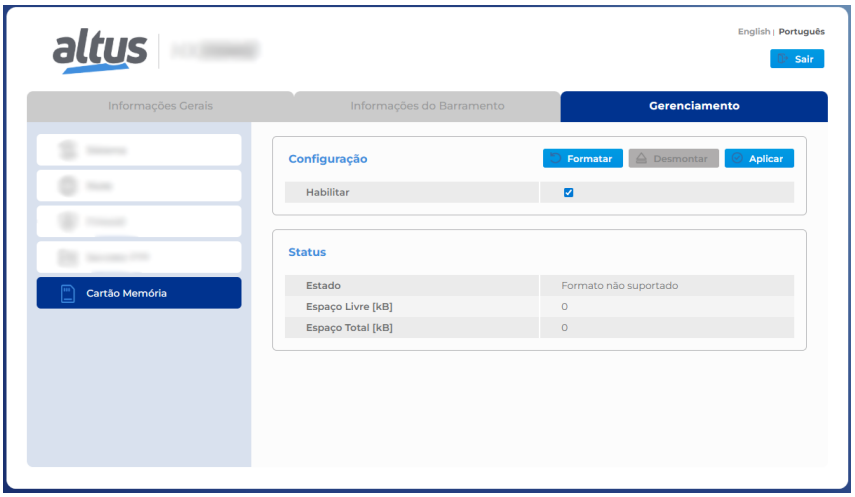


Figura 193: Mensagem de formato não suportado

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

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

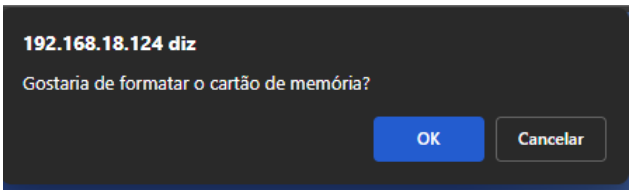


Figura 194: Mensagem Confirmação de Formatação

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

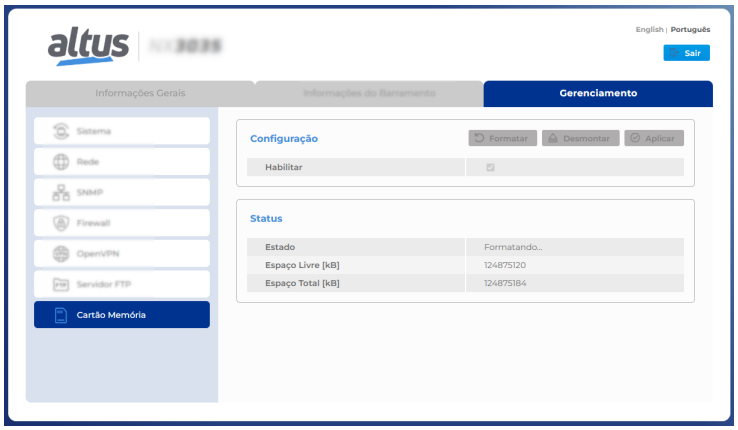


Figura 195: Cartão de Memória Formatando

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

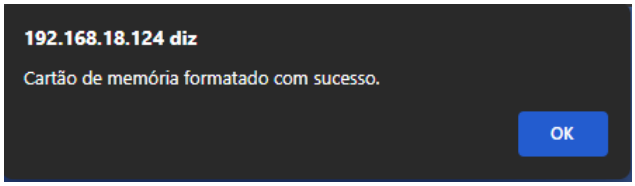


Figura 196: Mensagem de Formatação Concluída

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

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

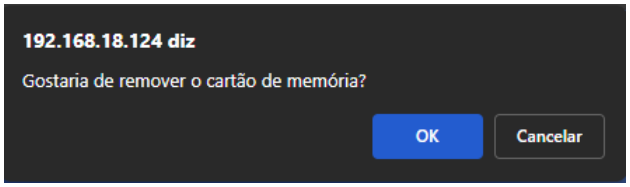


Figura 197: Mensagem Confirmação para Desmontar

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

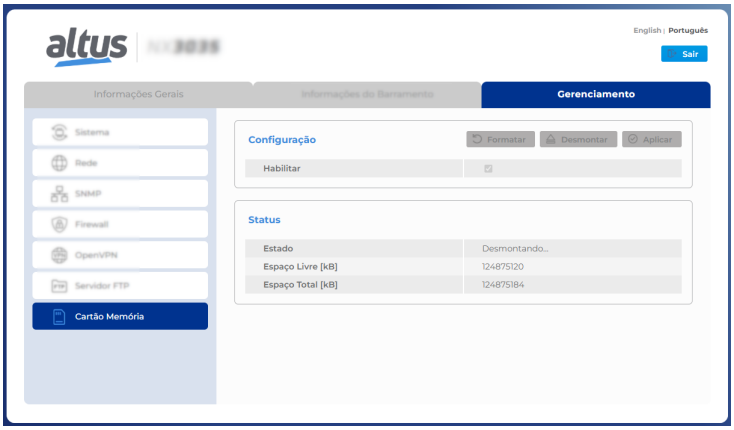


Figura 198: Cartão de Memória Desmontando

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

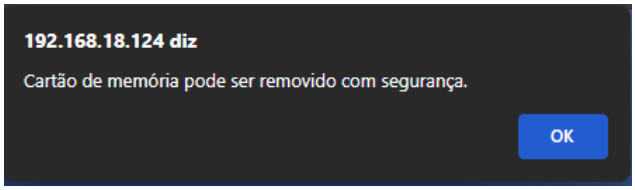


Figura 199: Mensagem de Desmontagem Concluída

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

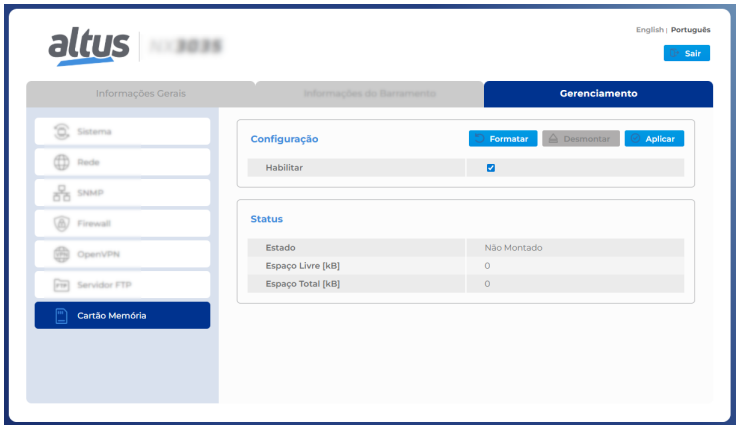


Figura 200: Cartão de Memória Não Montado

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

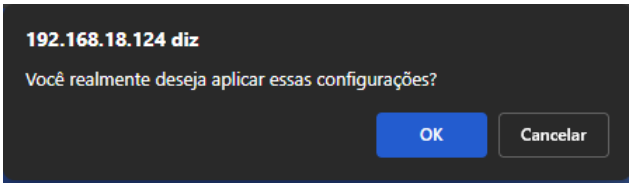


Figura 201: Mensagem Confirmação para Aplicar Configuração

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

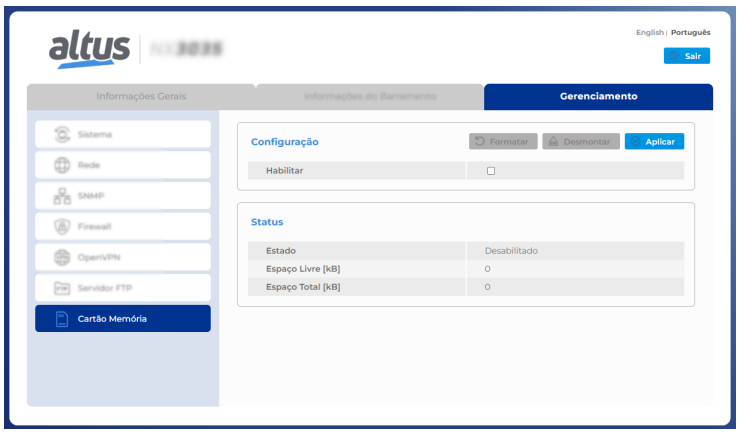


Figura 202: Cartão de Memória Interface Desabilitada

ATENÇÃO

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

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

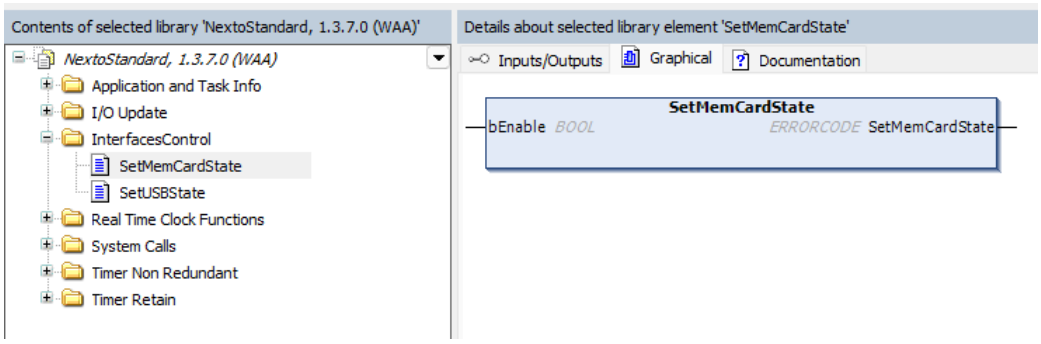


Figura 203: Informações de SetMemCardState no Library Manager

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

```
PROGRAM UserPrg
VAR
  bSetMemoryCardInterfaceState : BOOL;
  stErrorCode : NextoStandard.ERRORCODE;
END_VAR
-------------------------------------------------------------------------
// Exemplo de chamada de função para configurar interface do cartão de memó
   ria
stErrorCode:= NextoStandard.SetMemCardState(bEnable :=
   bSetMemoryCardInterfaceState);
```

ATENÇÃO

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

5.15. Docker

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

ATENÇÃO

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

5.15.1. Como acessar um Container Docker

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

5.15.1.1. Como Acessar o Portainer

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

CUIDADO

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

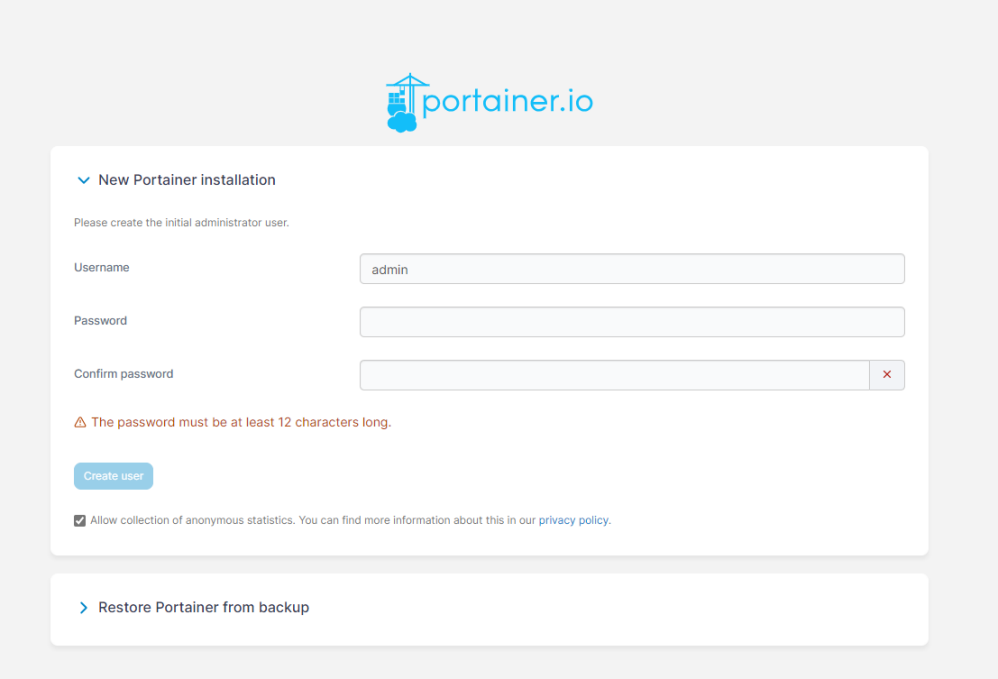


Figura 204: Tela de Login do Portainer

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

ATENÇÃO

Recomenda-se a atualização periódica do arquivo de backup.

5.15.2. Quick Setup

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

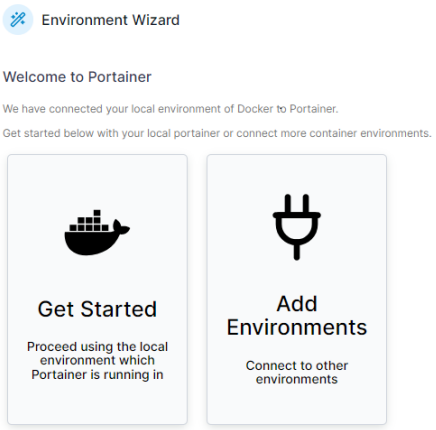


Figura 205: Tela inicial do Portainer

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

5.15.2.1. Informações do ambiente Local

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



Figura 206: Tela das Informações do Ambiente

ATENÇÃO

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

5.15.3. Dashboard de Ambiente

Após acessar a *Dashboard* clicando sob o ambiente que se deseja utilizar, você irá se deparar com algumas das informações que estavam presentes na tela anterior dentro de uma tabela de informações básicas nomeada de *"Environment info"*,

dentre elas temos as informações de hardware disponíveis e a versão do Docker. É possível acessar os menus listados abaixo interagindo com eles. De todos os menu listados abaixo, os de maior relevância são os menus de *Containers* e o de *Images*.

ATENÇÃO

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

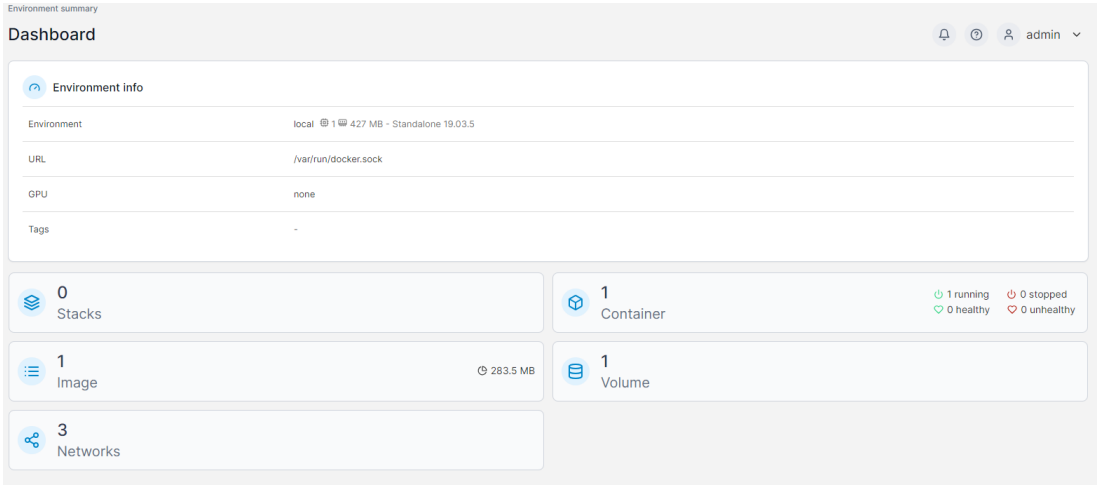


Figura 207: Dashboard

ATENÇÃO

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

5.15.3.1. Menu de Containers

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

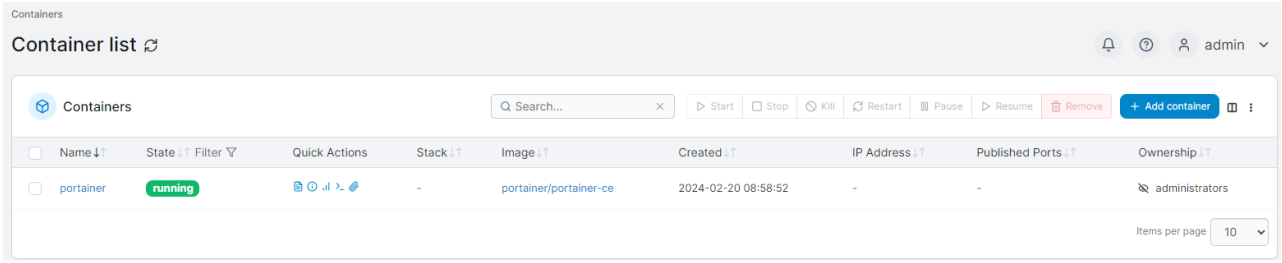


Figura 208: Menu de Containers

CUIDADO

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

5.15.3.2. Menu de Images

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

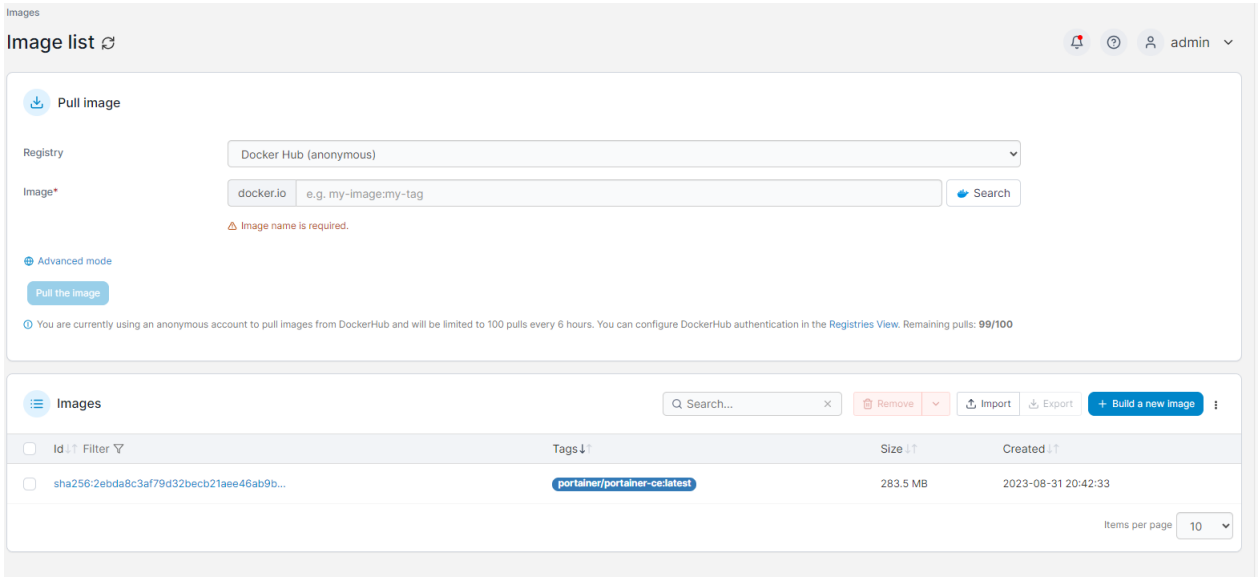


Figura 209: Menu de Imagens

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

5.15.4. Criando e Configurando um Container

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

ATENÇÃO

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

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

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

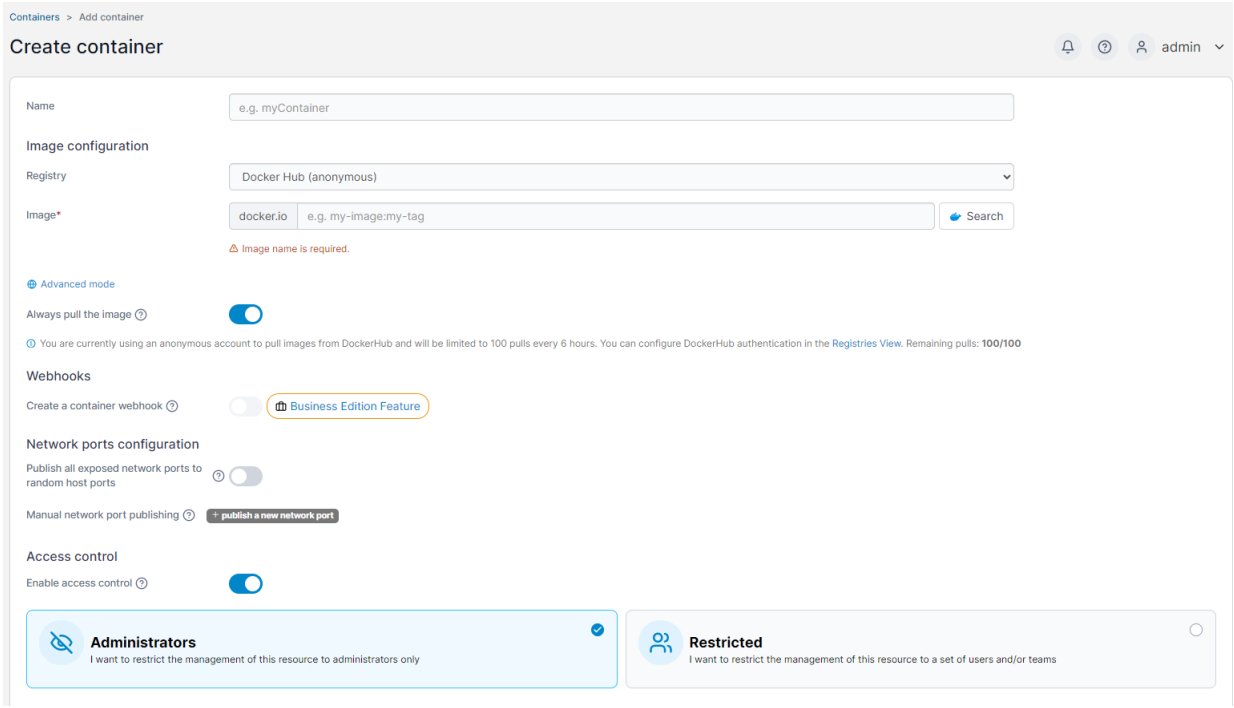


Figura 210: Tela de Criação do Container

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

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

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

CUIDADO

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

5.15.4.1. Configurando um Container

5.15.4.1.1. Menu Command & Logging

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

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

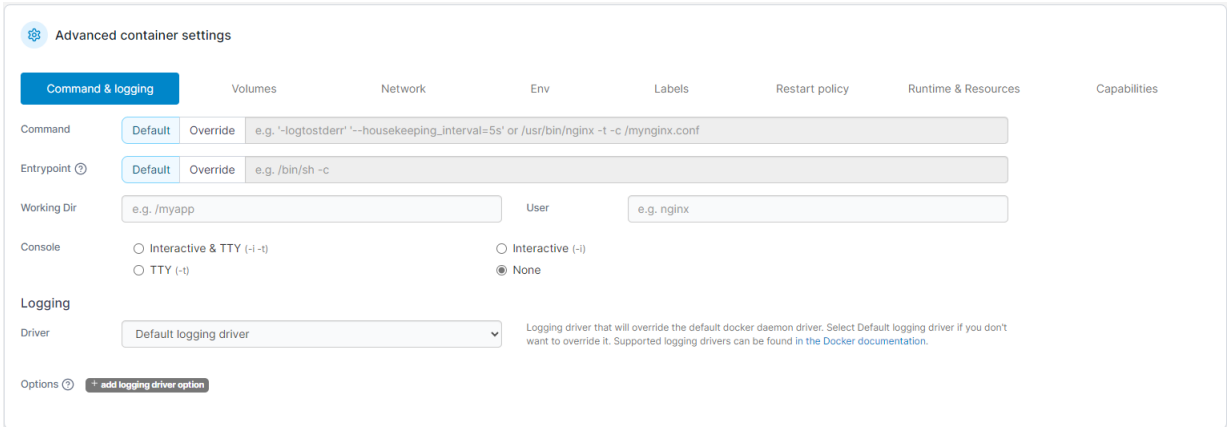


Figura 211: Menu de Configuração de Command & logging

5.15.4.1.2. *Menu Network*

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

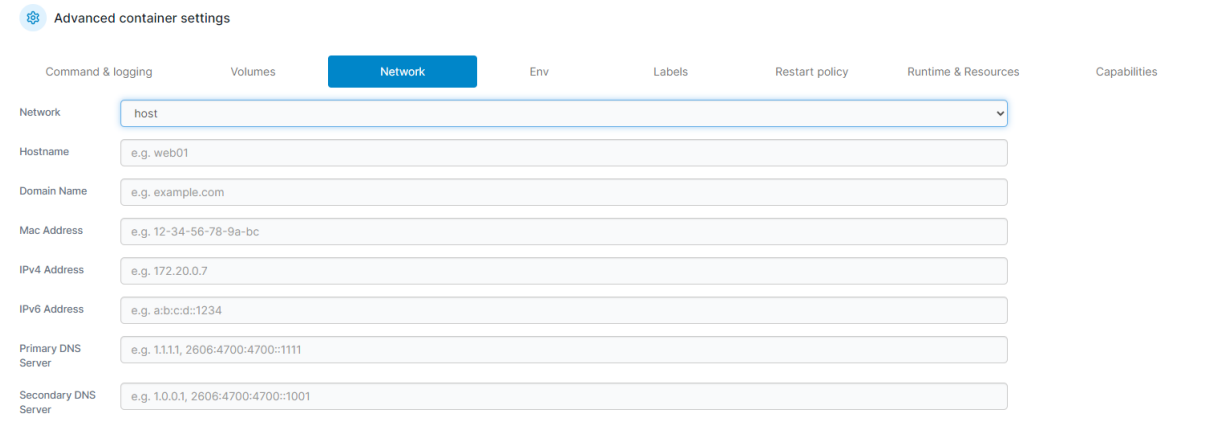


Figura 212: Menu de Configuração de Network

5.15.4.1.3. *Menu Runtime & Resources*

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

CUIDADO

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

ATENÇÃO

Este produto não suporta reserva de memória RAM. Embora a opção possa aparecer disponível na interface, o valor padrão não deve ser alterado.

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

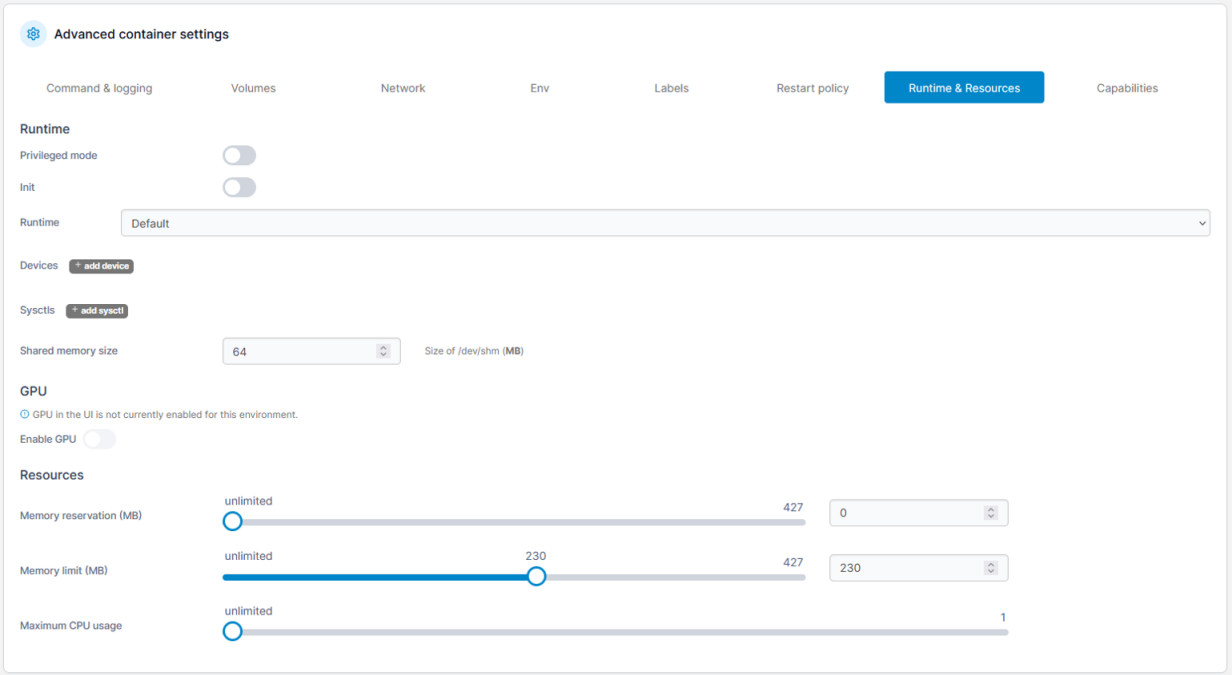


Figura 213: Menu de Configuração de Runtime & Resources

5.15.4.1.4. *Menu Capabilities*

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

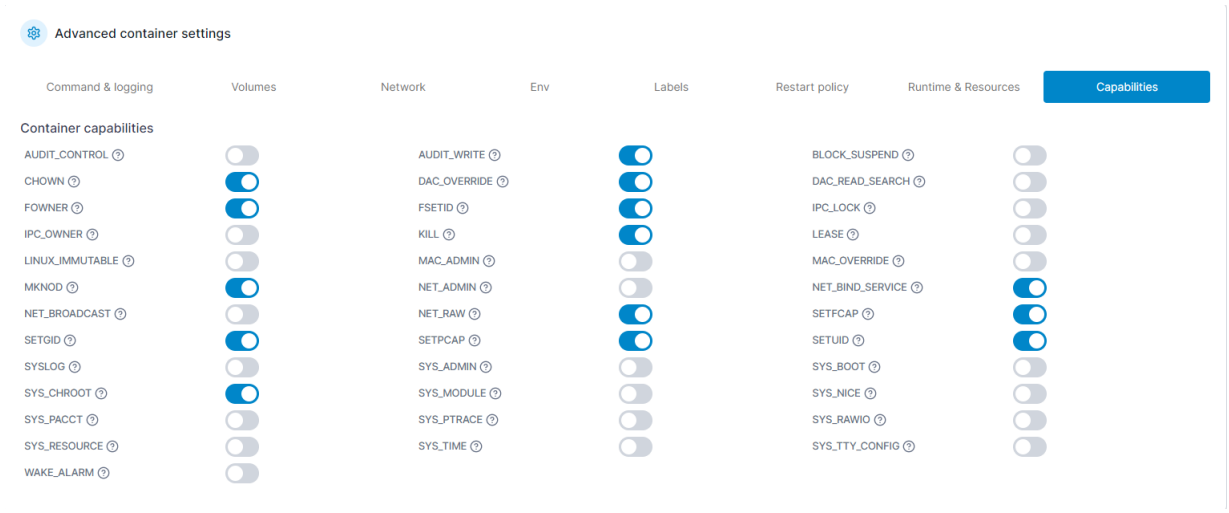


Figura 214: Menu de Configuração de Capabilities

CUIDADO

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

5.15.4.2. Utilizando um Container

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

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

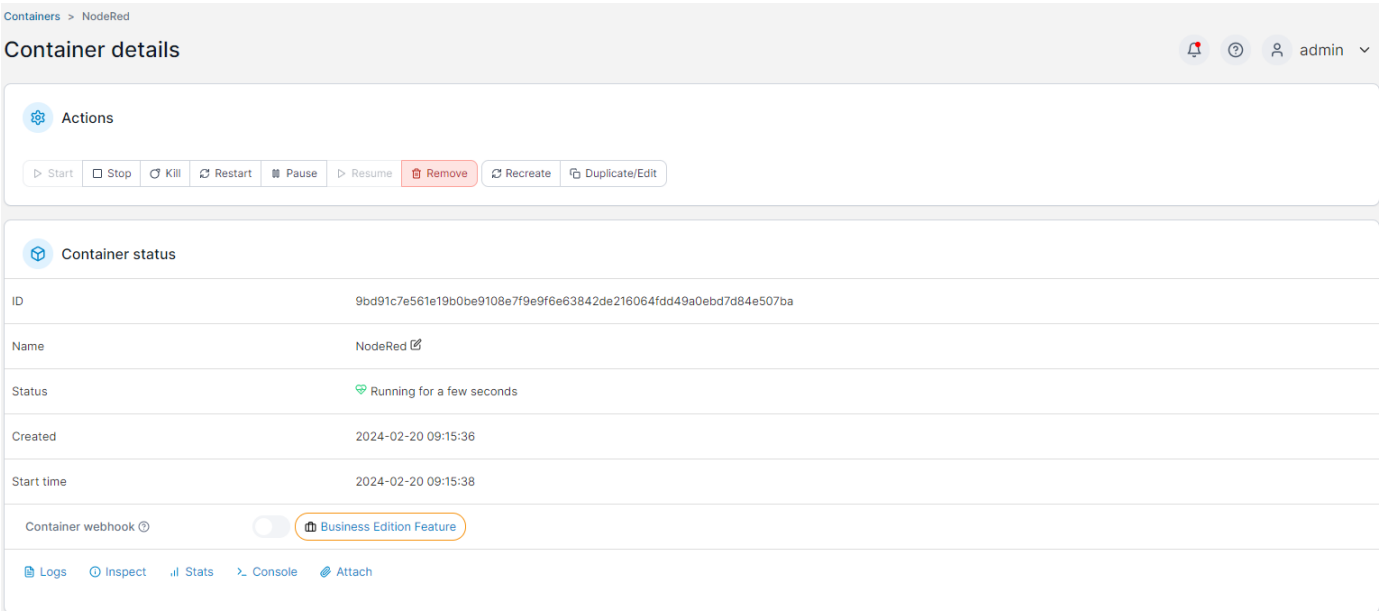


Figura 215: Tabela de Detalhes do Container

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

ATENÇÃO

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

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

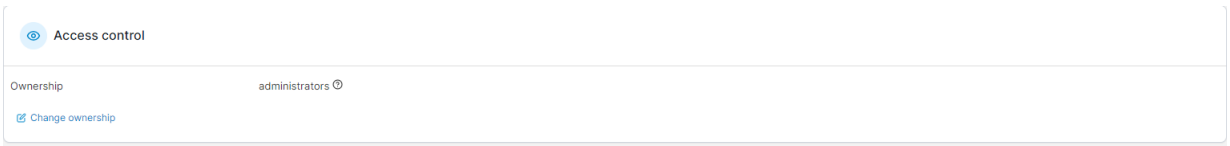


Figura 216: Access Control box

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

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

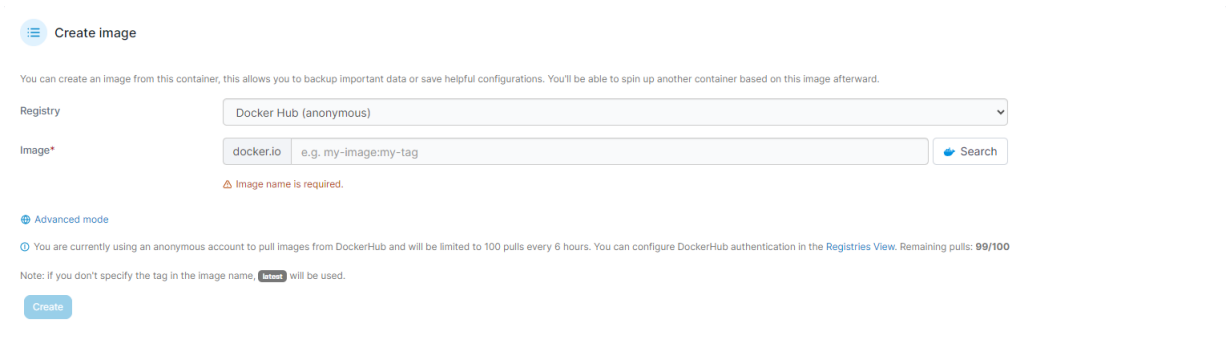


Figura 217: Tabela de Create Image

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

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

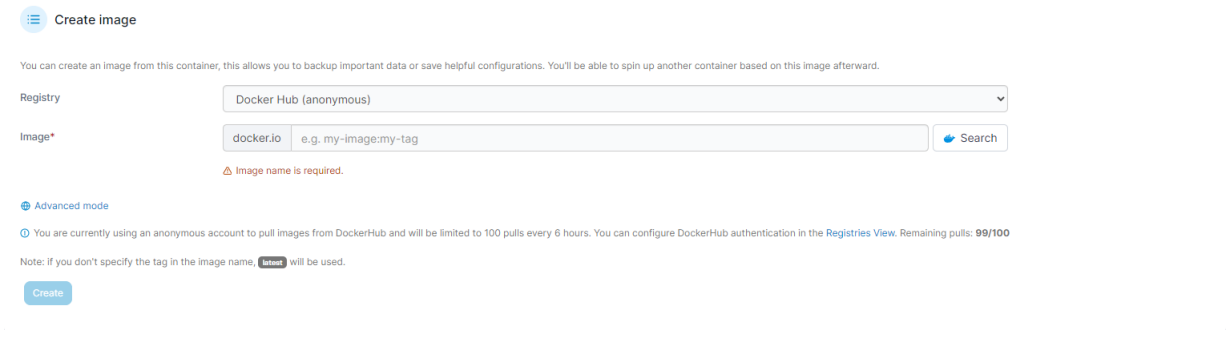


Figura 218: Criando uma imagem de um Container existente

5.15.4.4. Monitorando um Container

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

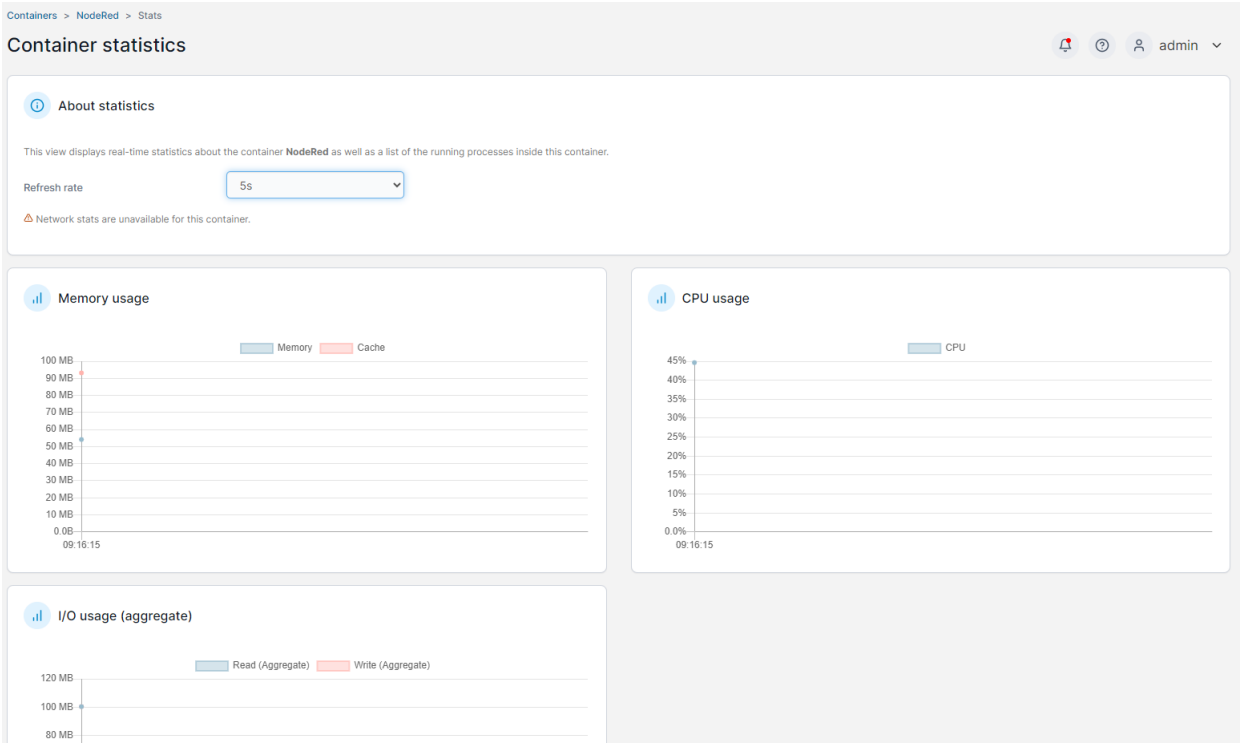


Figura 219: Tela de Monitoramento

5.15.4.5. Características Técnicas

5.15.4.5.1. Memória Flash

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

5.15.4.5.2. Memória RAM

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

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

PERIGO

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

5.15.4.5.3. UCP

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

5.15.4.6. Como acessasr as pastas do Docker

5.15.4.6.1. Mastertool

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

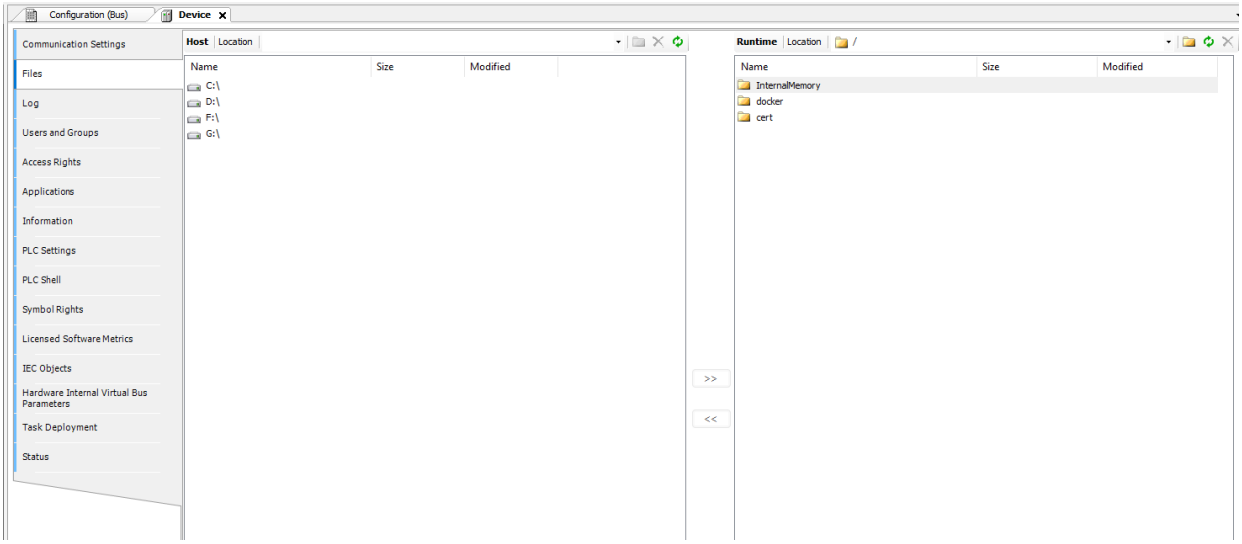


Figura 220: Pasta do Docker no Mastertool

5.15.4.7. Usando memória externa como volume Docker

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

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

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

Tabela 161: Caminhos do host para dispositivos de memória externa

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

Para implantar uma nova stack, navegue até a seção *Stacks* no Portainer e clique em *Add Stack*. A Figura 221 mostra um exemplo de um arquivo de stack que monta tanto um cartão SD quanto um dispositivo de armazenamento em massa USB em um serviço:

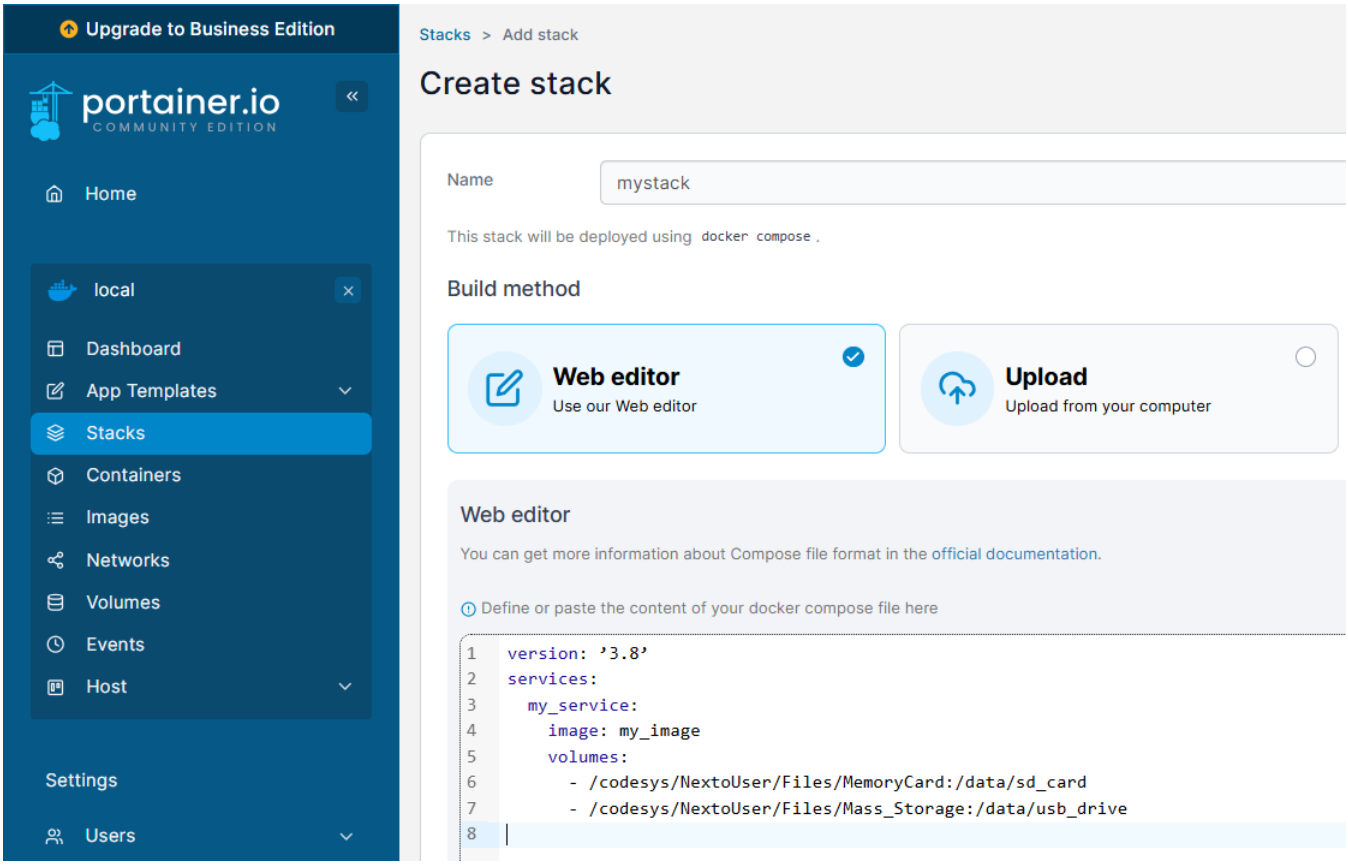


Figura 221: Exemplo de Arquivo de Stack com Memória Externa

ATENÇÃO

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

5.15.4.8. Solução de Problemas

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

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

Causa: horário do PLC é inválido.
Solução: ajustar o horário do PLC.

5.15.4.8.2. Portainer inacessível depois de uma reinicialização da UCP

Causa: não foram aplicados os limites de memória para os Containers conforme seção Menu Runtime & Resources
Solução: aplicar os limites de memória conforme descrito neste manual.

6. Manutenção

6.1. Diagnósticos

6.1.1. Diagnósticos via LEDs

Os controladores da Linha XF possuem cinco LEDs para indicar estados gerais e diagnósticos: SYS, RUN, DG, WD e PWR, além de quatro LEDs específicos para cada interface: SD, USB, CAN e RS485, bem como os LEDs do conector Ethernet. As E/S integradas possuem sua própria indicação por LED, de acordo com sua função: digital ou analógica. As tabelas a seguir mostram o significado de cada estado e sua respectiva descrição.

6.1.1.1. SYS (Sistema)

Cor	Estado	Descrição
Verde	Desligado	Sistema desligado/sistema corrompido
	Piscando	Sistema inicializando
	Ligado	Sistema pronto

Tabela 162: Descrição do status do LED do sistema

Nota:

Piscando indefinidamente: Se passar muito tempo e o LED não parar de piscar, a CPU pode ter perdido sua assinatura. Consulte a tabela Descrição dos estados do LED de Watchdog.

6.1.1.2. RUN (Aplicação)

Cor	Estado	Descrição
Vermelho/ Verde	Desligado	Sem aplicação
Vermelho	Ligado	Aplicação em Modo Stop
	Piscando 5x	Aplicação em Breakpoint
Verde	On	Aplicação em Modo Run

Tabela 163: Descrição do status do LED RUN

6.1.1.3. SD (Cartão de Memória)

Cor	Estado	Descrição
Verde	Desligado	Cartão não montado / Cartão não montado
	Ligado	Cartão montado
	Piscando	Sistema de arquivos desconhecido / Não consegue montar

Tabela 164: Descrição do status do LED do cartão microSD

6.1.1.4. DG (Diagnóstico)

Cor	Estado	Descrição	Causas	Prioridade
Verde	Desligado	Sem diagnóstico ativo	Operação normal	-
	Piscando 2x	Módulos de barramento com diagnóstico	Ao menos um módulo, incluindo a UCP, tem um diagnóstico	1
	Piscando 3x	Forçamento de dados	Alguma área de memória está sendo forçada pelo usuário através do Mastertool	2
	Piscando 4x	Configuração do barramento ou erro de hardware	Configuração do barramento ou erro de hardware	0 (mais alta)
	Piscando	Identificação por piscada do dispositivo	Permanece piscando por 5 segundos	-

Tabela 165: Descrição dos estados dos LEDs de diagnóstico

6.1.1.5. WD (Watchdog)

Cor	Estado	Descrição	Causas	Prioridade
Vermelho	Desligado	Sem indicação de watchdog	Operação normal	-
	Ligado	Hardware Watchdog	Módulo danificado e/ou SO corrompido	0 (mais alta)
	Piscando 1x	Software Watchdog	Watchdog da aplicação do usuário	1
Verde	Ligado	Assinatura ausente	Assinatura da fábrica ausente	-
	Desligado	Assinatura OK	Operação normal	-

Tabela 166: Descrição dos estados do LED de Watchdog

Notas:

Software Watchdog: Para remover a indicação de watchdog, reinicie o aplicativo ou desligue e ligue a CPU novamente. Esse watchdog ocorre quando o tempo de execução do aplicativo do usuário é maior que o tempo de watchdog configurado.

O diagnóstico pode ser verificado na variável Exception.wExceptionCode, consulte a Tabela Códigos de Exceção RTS.

Hardware Watchdog: Para redefinir qualquer indicação de watchdog, como no LED WD ou no operando Reset.bWatchdogReset, o módulo deve ser desconectado da fonte de alimentação.

Para verificar as condições do aplicativo na reinicialização do módulo, consulte as configurações na Tabela Configuração da UCP.

6.1.1.6. PWR (Alimentação)

Cor	Estado	Descrição
Verde	Desligado	Sem fonte de alimentação
	On	Fonte de alimentação conectada

Tabela 167: Descrição do status do LED PWR

6.1.1.7. CAN

Cor	Estado	Descrição
Verde	Desligado	CAN não está sendo usado ou não há troca de dados.
	Piscando	CAN transmitindo ou recebendo dados

Tabela 168: Descrição do status do LED CAN

6.1.1.8. RS485

Cor	Estado	Descrição
Green	Desligado	RS-485 não está sendo usado ou não há troca de dados.
	Piscando	RS-485 transmitindo ou recebendo dados

Tabela 169: Descrição do status do LED RS-485

6.1.1.9. USB

Cor	Estado	Descrição
Verde	Desligado	USB não montado/USB não conectado
	Ligado	USB montado

Tabela 170: Descrição do status do LED USB

6.1.1.10. LEDs Conector RJ45

Os dois LEDs localizados nos conectores RJ45 auxiliam o usuário na detecção de problemas na rede física instalada, indicando a velocidade da conexão e a existência de tráfego de comunicação na interface. O significado dos LEDs é apresentado na tabela abaixo.

Amarelo	Verde	Descrição
○	○	Link de rede ausente
●	○	Link de rede 10 Mbytes/s
●	●	Link de rede 100 Mbytes/s
X	-	Ocorrência de transmissão ou recepção na rede Ethernet, para ou com este endereço IP. Pisca conforme a demanda da CPU Nexto e não a cada transmissão ou recepção; em outras palavras, pode piscar em uma frequência inferior à frequência real de transmissão ou recepção.

Tabela 171: Significado dos LEDs Ethernet

6.1.1.11. DO (Saídas Digitais)

Cor	Estado	Descrição
Verde	Desligado	A saída digital correspondente está DESLIGADA
	Ligado	A saída digital correspondente está LIGADA

Tabela 172: Descrição do status do LED Saída Digital

6.1.1.12. DI (Entradas Digitais)

Cor	Estado	Descrição
Verde	Desligado	A entrada digital correspondente está DESLIGADA.
	Ligado	A entrada digital correspondente está LIGADA.

Tabela 173: Descrição do status do LED Entrada Digital

6.1.1.13. AO (Saídas Analógicas)

Cor	Estado	Descrição
Verde	Desligado	A saída analógica correspondente NÃO está configurada
	Ligado	A saída analógica correspondente está configurada

Tabela 174: Descrição do status do LED Saída Analógica

6.1.1.14. AI (Entradas Analógicas)

Cor	Estado	Descrição
Verde	Desligado	A entrada analógica correspondente NÃO está configurada
	Ligado	A entrada analógica correspondente está configurada

Tabela 175: Descrição do status do LED Entrada Analógica

6.1.2. Diagnósticos via Variáveis

Os controladores Nexto XF 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.

6.1.2.1. Diagnósticos Resumidos

A tabela a seguir mostra o significado de diagnósticos resumidos:

DG_CPU.tSummarized.*	Type	Description
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.
bMemoryCardError	BOOL	TRUE – O cartão de memória está inserido na UCP, porém não está funcionando corretamente.
		FALSE – O cartão de memória está funcionando corretamente.
bCOM1ConfigError	BOOL	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface serial COM 1.
		FALSE – A configuração da interface serial COM 1 está correta.
bNET1ConfigError	BOOL	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface Ethernet NET 1.
		FALSE – A configuração da interface Ethernet NET 1 está correta.
bNET2ConfigError	BOOL	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface Ethernet NET 2.
		FALSE – A configuração da interface Ethernet NET 2 está correta.
bInvalidDateTime	BOOL	TRUE – A data ou a hora são inválidas.
		FALSE – A data e a hora estão corretas.
bRuntimeReset	BOOL	TRUE – O RTS (Runtime System) foi reiniciado pelo menos uma vez. Esse diagnóstico somente é limpo na reinicialização do sistema.
		FALSE – O RTS (Runtime System) está operando normalmente.
bRetentivityLost	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.

DG_CPU.tSummarized.*	Type	Description
bOTDSwitchError	BOOL	TRUE – A tecla ficou travada por mais de 20 s pelo menos uma vez enquanto a UCP esteve energizada. Esse diagnóstico somente é limpo na reinicialização do sistema.
		FALSE – A tecla não está ou ficou travada enquanto a UCP esteve energizada.

Tabela 176: Diagnósticos Resumidos

Notas:

Falha de Hardware: Caso o diagnóstico de Falha de Hardware seja verdadeiro, encaminhar a UCP para Assistência Técnica da Altus, pois a mesma apresenta problemas no RTC, processador auxiliar, ou outros recursos de hardware.

Exceção de software: Caso o diagnóstico de exceção no software seja verdadeiro, o usuário deverá verificar a sua aplicação para garantir que a mesma não esteja acessando indevidamente a memória. Se o problema persistir, o setor de Suporte da Altus deverá ser consultado. Os códigos de exceção no software estão descritos após a tabela de diagnósticos detalhados da UCP.

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.

6.1.2.2. Diagnósticos Detalhados

As tabelas a seguir mostram os diagnósticos detalhados; para sua consulta é importante verificar as observações abaixo:

- **Visualização das Estruturas de Diagnóstico:** As Estruturas de Diagnóstico adicionadas ao projeto podem ser visualizadas acessando o item "*Library Manager*" na árvore de dispositivos da janela do Mastertool. Com isso, é possível visualizar todos os tipos de dados definidos na estrutura.
- **Contadores:** Todos os contadores dos diagnósticos retornam à zero quando o seu valor limite é ultrapassado.
- *DG_Modulo.*, sendo a palavra *Modulo* substituída pelo produto utilizado.

6.1.2.2.1. Grupo de diagnósticos detalhados: Target

DG_Modulo.tDetailed.Target.*	Tamanho	Descrição
dwCPUModel	DWORD	Modelo da CPU
abyCPUVersion	BYTE ARRAY(4)	Versão de firmware
abyBootloaderVersion	BYTE ARRAY(4)	Versão de bootloader
abyAnalogprocVersion	BYTE ARRAY(4)	Versão de coprocessador E/S Analógico
abyBusControllerVersion	BYTE ARRAY(4)	Versão de controlador do barramento
abyFastOutputVersion	BYTE ARRAY(4)	Versão da saída rápida

Tabela 177: Diagnósticos Detalhados Grupo Target

6.1.2.2.2. Grupo de diagnósticos detalhados: Hardware

DG_Modulo.tDetailed.Hardware.*	Tamanho	Descrição
bRTCFailure	BIT	O processador principal não está habilitado para comunicar com o RTC (relógio da UCP)
bIntegratedIoFailure	BIT	Falha na comunicação com o E/S integrado

Tabela 178: Diagnósticos Detalhados Grupo Hardware

6.1.2.2.3. Grupo de diagnósticos detalhados: Exception

DG_Modulo.tDetailed.Exception.*	Tamanho	Descrição
wExceptionCode	WORD	Código de Exceção gerado pelo RTS, o significado dos códigos pode ser conferido na Tabela 180.
byProcessorLoad	BYTE	Nível, em percentagem (%), de carga no processador.

Tabela 179: Diagnósticos Detalhados Grupo Exception

Nota:
Código de exceção: O código de exceção gerado pelo RTS (Runtime System) pode ser consultado abaixo:

Código	Descrição	Código	Descrição
0x0000	Não há código de exceção.	**0x0051**	Violação de acesso.
0x0010	Tempo de Cão-de-Guarda da tarefa IEC expirado (Cão-de-Guarda de Software).	**0x0052**	Instrução privilegiada.
0x0012	Erro de configuração de E / S.	**0x0053**	Falha na página.
0x0013	Erros de check-up após o download do programa.	**0x0054**	Estouro de pilha.
0x0014	Erro de rede de campo.	**0x0055**	Disposição inválida.
0x0015	Erro de atualização de E / S.	**0x0056**	Manobra inválida.
0x0016	Tempo de ciclo (execução) excedido.	**0x0057**	Página protegida.
0x0017	Atualização online do programa por muito tempo.	**0x0058**	Falha dupla.
0x0018	Referências externas não resolvidas.	**0x0059**	OpCode inválido.
0x0019	Download rejeitado.	**0x0100**	Desalinhamento do tipo de dado.
0x001A	Projeto não-carregado, pois as variáveis retentivas não podem ser realocadas.	**0x0101**	Limite de arrays excedido.
0x001B	Projeto não-carregado e excluído.	**0x0102**	Divisão por zero.
0x001C	Fora da pilha de memória.	**0x0103**	Estouro.
0x001D	Memória retentiva corrompida; não pode ser mapeada.	**0x0104**	Não continuável.

Código	Descrição	Código	Descrição
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 180: Códigos de Exceção RTS

6.1.2.2.4. Grupo de diagnósticos detalhados: WebVisualization

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

Tabela 181: Diagnósticos Detalhados Grupo WebVisualization

6.1.2.2.5. Grupo de diagnósticos detalhados: RetainInfo

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

Tabela 182: Diagnósticos Detalhados Grupo RetainInfo

6.1.2.2.6. Grupo de diagnósticos detalhados: Reset

DG_Modulo.tDetailed.Reset.*	Tamanho	Descrição
bWatchdogReset	BIT	A CPU foi reiniciada devido ao watchdog ativo na última inicialização.

Tabela 183: Diagnósticos Detalhados Grupo Reset

6.1.2.2.7. Grupo de diagnósticos detalhados: Thermometer

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

Tabela 184: Diagnósticos Detalhados Grupo Thermometer

Nota:

Temperatura: Para a visualização da temperatura diretamente no endereço de memória, deve-se realizar uma conversão, pois o tamanho do dado é DINT e a monitoração é realizada em 4 bytes. Por isso, indica-se a utilização da variável simbólica associada, pois a mesma já fornece o valor final da temperatura.

6.1.2.2.8. Grupo de diagnósticos detalhados: Serial

DG_Modulo.tDetailed.Serial.COM1.*	Tamanho	Descrição
byProtocol	BYTE	Protocolo selecionado na COM 1: 00: Sem protocolo 01: MODBUS RTU Master 02: MODBUS RTU Slave 03: Outro protocolo
dwRXBytes	DWORD	Contador de caracteres recebidos através da COM 1 (0 a 4294967295).
dwTXBytes	DWORD	Contador de caracteres transmitidos através da COM 1 (0 a 4294967295).
wRXPendingBytes	WORD	Número de caracteres pendentes no buffer de leitura na COM 1 (0 a 4096).
wTXPendingBytes	WORD	Número de caracteres pendentes no buffer de transmissão na COM 1 (0 a 1023).
wBreakErrorCounter	WORD	O transmissor está mantendo a linha de dados em zero por muito tempo, de acordo com o bit de dados configurado.
wParityErrorCounter	WORD	O frame recebido tem o bit de paridade incompatível.
wFrameErrorCounter	WORD	O frame recebido tem o ponto de partida errado, geralmente causado por uma incompatibilidade de ruído ou baud rate.
wRXOverrunCounter	WORD	Quando o buffer de anel recebido está cheio e começa a perder os frames antigos (muitos frames não tratados pelo dispositivo).

Tabela 185: Diagnósticos Detalhados Grupo Serial COM 1

Nota:

Contador de erro de paridade: Quando a paridade da interface serial COM 1 estiver configurada como Sem Paridade, este contador de erros não será incrementado ao receber uma mensagem com paridade diferente. Nesse caso, será indicado erro de frame.

6.1.2.2.9. Grupo de diagnósticos detalhados: CAN

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

Tabela 186: Diagnósticos Detalhados Grupo CAN

6.1.2.2.10. Grupo de diagnósticos detalhados: USB

DG_Modulo.tDetailed.USB.*	Tamanho	Descrição
byUSBDevice	BYTE	Tipo do dispositivo conectado à porta USB.
bOvercurrent	BIT	O dispositivo conectado na porta USB está drenando mais corrente do que o suportado.
bEnable	BIT	A porta USB está habilitada.
tMassStorage.byMountState	BYTE	Informa o estado do dispositivo.
tMassStorage.dwFreeSpaceKb	DWORD	Informa o espaço livre no dispositivo de armazenamento em massa.
tMassStorage.dwTotalSizeKb	DWORD	Informa o tamanho total do dispositivo de armazenamento em massa.
tSerialConverter.byProtocol	BYTE	Protocolo selecionado na COM 1: 00: Sem protocolo 01: MODBUS RTU Master 02: MODBUS RTU Slave 03: Outro 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).

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

Tabela 187: Diagnósticos Detalhados Grupo USB

6.1.2.2.11. Grupo de diagnósticos detalhados: Ethernet

DG_Modulo.tDetailed.Ethernet.*	Tamanho	Descrição
NET[x].bLinkDown	BIT	Indica o estado do link na NET[x].
NET[x].byOperatingMode	BYTE	Indica o modo de operação da interface NET[x]
NET[x].byOperatingState	BYTE	Indica o estado de operação da interface NET[x].
NET[x].szIP	STRING (15)	Endereço IP NET[x].
NET[x].szMask	STRING (15)	Máscara de Subrede NET[x].

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

Tabela 188: Diagnósticos Detalhados Grupo Ethernet NET[x]

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

DG_Modulo.tDetailed.Ethernet.RSTP.*	Tamanho	Descrição
RSTP[x].bProtocolEnabled	BIT	Indica se o protocolo RSTP está habilitado.
RSTP[x].bBridgeIsRoot	BIT	Indica se a Bridge é a root.
RSTP[x].eProtocol	BYTE	Indica a versão do protocolo.
RSTP[x].sBridgeId	STRING (22)	ID da bridge local (prioridade e MAC).
RSTP[x].sRootId	STRING (22)	ID da bridge root (prioridade e MAC).
RSTP[x].eRootPort	BYTE	Indica qual é a porta root.
RSTP[x].dwRootPathCost	DWORD	Path Cost até a bridge root.

DG_Modulo.tDetailed.Ethernet.RSTP.*	Tamanho	Descrição
RSTP[x].byBridgeHelloTime	BYTE	Hello Time configurado.
RSTP[x].byRootHelloTime	BYTE	Hello Time aprendido.
RSTP[x].byBridgeMaxAge	BYTE	Max Age configurado.
RSTP[x].byRootMaxAge	BYTE	Max Age aprendido.
RSTP[x].byBridgeForwDelay	BYTE	Forward Delay configurado.
RSTP[x].byRootForwDelay	BYTE	Forward Delay aprendido.
RSTP[x].dwTxHoldCount	DWORD	Limite de BPDUs transmitidos por ciclo.
RSTP[x].dwTimeSinceTC	DWORD	Tempo, em segundos, desde a última alteração de topologia.
RSTP[x].dwTCCount	DWORD	Número de vezes que a topologia foi alterada.
RSTP[x].aPort[y].eName	BYTE	Nome da porta.
RSTP[x].aPort[y].eRole	BYTE	Função da porta.
RSTP[x].aPort[y].eState	BYTE	Estado da porta.
RSTP[x].aPort[y].wIdentity	WORD	ID da porta.
RSTP[x].aPort[y].dwPathCost	DWORD	Custo da porta.
RSTP[x].aPort[y].bOperEdge	BIT	Indica se a porta está trabalhando como Edge.
RSTP[x].aPort[y].bOperP2P	BIT	Indica se a porta está trabalhando como Point-to-Point.
RSTP[x].aPort[y].dwNumTxBPDU	DWORD	Número de frames BPDUs transmitidos na porta.
RSTP[x].aPort[y].dwNumRxBPDU	DWORD	Número de frames BPDUs recebidos na porta.

Tabela 189: Diagnósticos Detalhados Grupo Ethernet RSTP

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

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

Tabela 190: Diagnósticos Detalhados Grupo Ethernet MRP

Nota:
[x]: é o número do MRP.

6.1.2.2.12. Grupo de diagnósticos detalhados: UserFiles

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

Tabela 191: Diagnósticos Detalhados Grupo UserFiles

Nota:

Partição do Usuário: A partição de usuário é uma área da memória reservada para o armazenamento de dados no controlador. Por exemplo: arquivos com extensão PDF, arquivos com extensão DOC e demais dados.

6.1.2.2.13. Grupo de diagnósticos detalhados: UserLogs

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

Tabela 192: Diagnósticos Detalhados Grupo UserLogs

6.1.2.2.14. Grupo de diagnósticos detalhados: MemoryCard

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

Tabela 193: Diagnósticos Detalhados Grupo MemoryCard

6.1.2.2.15. Grupo de diagnósticos detalhados: Application

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

Tabela 194: Diagnósticos Detalhados Grupo Application

6.1.2.2.16. Grupo de diagnósticos detalhados: ApplicationInfo

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

Tabela 195: Diagnósticos Detalhados Grupo ApplicationInfo

6.1.2.2.17. Grupo de diagnósticos detalhados: SNTP

DG_Modulo.tDetailed.SNTP.*	Tamanho	Descrição
bServiceEnabled	BIT	Serviço SNTP habilitado.
byActiveTimeServer	BYTE	Indica qual servidor está ativo: 00: Nenhum servidor ativo. 01: Servidor primário ativo. 02: Servidor secundário ativo.
wPrimaryServerDownCount	WORD	Contador de vezes que o servidor primário esteve indisponível (0 a 65535).
wSecondaryServerDownCount	WORD	Contador de vezes que o servidor secundário esteve indisponível (0 a 65535).
dwRTCTimeUpdatedCount	DWORD	Contador de vezes que o RTC foi atualizado pelo serviço SNTP (0 a 4294967295).
byLastUpdateSuccessful	BYTE	Indica status da última atualização: 00: Não foi atualizado. 01: Última atualização falhou. 02: Última atualização teve sucesso.

DG_Modulo.tDetailed.SNTP.*	Tamanho	Descrição
byLastUpdateTimeServer	BYTE	Indica qual servidor foi utilizado na última atualização: 00: Nenhuma atualização. 01: Servidor primário. 02: Servidor secundário.
sLastUpdateTime.byDayOfMonth	BYTE	Dia da última atualização do RTC.
sLastUpdateTime.byMonth	BYTE	Mês da última atualização do RTC.
sLastUpdateTime.wYear	WORD	Ano da última atualização do RTC.
sLastUpdateTime.byHours	BYTE	Hora da última atualização do RTC.
sLastUpdateTime.byMinutes	BYTE	Minuto da última atualização do RTC.
sLastUpdateTime.bySeconds	BYTE	Segundo da última atualização do RTC.
sLastUpdateTime.wMilliseconds	WORD	Milissegundo da última atualização do RTC.

Tabela 196: Diagnósticos Detalhados Grupo SNTP

6.1.2.2.18. Grupo de diagnósticos detalhados: Integrated IO

DG_Modulo.tDetailed.IntegratedIO.*	Tamanho	Descrição
AnalogInputs.tAnalogInput_xx.bInputNotEnabled	BIT	O canal de entrada não está habilitado na configuração
AnalogInputs.tAnalogInput_xx.bOverRange	BIT	O nível do sinal de entrada está acima do valor máximo definido para o tipo de entrada selecionado.
AnalogInputs.tAnalogInput_xx.bOpenLoop	BIT	A impedância da carga conectada ao canal de entrada está acima do máximo aceito (somente para entradas de corrente [AI0 a AI5]).
AnalogOutputs.tAnalogOutput_xx.bOutputNotEnabled	BIT	O canal de saída não está habilitado na configuração.
AnalogOututs.tAnalogOutput_xx.bOpenLoop	BIT	A impedância da carga conectada ao canal de saída está acima do máximo aceito (somente para o modo de saída de corrente).
AnalogOututs.tAnalogOutput_xx.bShortCircuit	BIT	A impedância da carga conectada ao canal de saída está abaixo do mínimo aceito (somente para o modo de saída de tensão).

Tabela 197: Diagnóstico detalhado do grupo de E/S integrado

6.1.2.2.19. Grupo de diagnósticos detalhados: OpenVPN

DG_Modulo.tDetailed.OpenVPN.*	Tamanho	Descrição
byOperationMode	BYTE	Modo de operação da VPN: SERVER(0): Dispositivo operando como servidor. CLIENT(1): Dispositivo operando como cliente.
bServiceEnabled	BIT	Serviço de VPN habilitado.
bServiceRunning	BIT	Serviço de VPN em execução.

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

Tabela 198: Diagnósticos Detalhados Grupo OpenVPN

6.1.2.2.20. Grupo de diagnósticos detalhados: Firewall

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

Tabela 199: Diagnósticos Detalhados Grupo Firewall

6.1.2.2.21. Grupo de diagnósticos detalhados: FTP

DG_Modulo.tDetailed.FTP.*	Tamanho	Descrição
bServiceEnabled	BIT	Serviço de FTP habilitado.
dwConnectedClients	DWORD	Quantidade de clientes conectados.

DG_Modulo.tDetailed.FTP.*	Tamanho	Descrição
sLastUpdateTime.byDayOfMonth	BYTE	Dia da última atualização do FTP.
sLastUpdateTime.byMonth	BYTE	Mês da última atualização do FTP.
sLastUpdateTime.wYear	WORD	Ano da última atualização do FTP.
sLastUpdateTime.byHours	BYTE	Hora da última atualização do FTP.
sLastUpdateTime.byMinutes	BYTE	Minuto da última atualização do FTP.
sLastUpdateTime.bySeconds	BYTE	Segundo da última atualização do FTP.
sLastUpdateTime.wMilliseconds	WORD	Milissegundo da última atualização do FTP.

Tabela 200: Diagnósticos Detalhados Grupo FTP

6.2. Botão de usuário da UCP

O botão está localizado na parte superior do controlador, em um local de fácil acesso. A figura abaixo mostra sua localização:

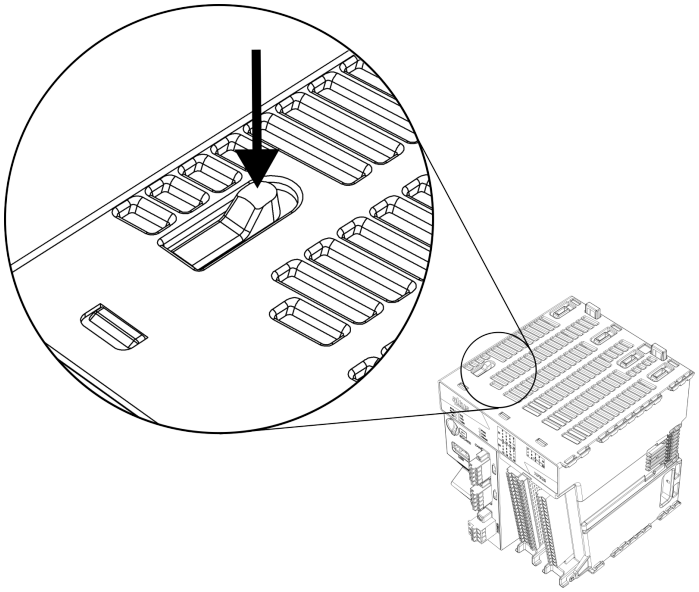


Figura 222: Botão do XF3xx

Função	Ativação
Não carregar a aplicação / Não aplique o firewall	Mantenha o botão pressionado enquanto liga o dispositivo até que o LED SYS permaneça ligado
Desmontar o cartão de memória (ejeção segura)	Pressione e segure até o LED do cartão microSD desligar.

Tabela 201: Descrição das funções do botão

Tipo de Pressionamento	Tempo Mínimo	Tempo Máximo	Condição para Indicação
Sem pressionamento	-	59,99 ms	-
Pressionamento curto	60 ms	0,99 s	Liberação
Pressionamento longo	1 s	20 s	Mais do que 1 s até 20 s
Tecla Presa	20,01 s	(∞)	Indicação em diagnóstico, consultar a Tabela 176

Tabela 202: Tempo de um Pressionamento

6.2.1. Diagnósticos através de 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.

6.2.1.1. GetTaskInfo

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



Figura 223: 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 203: 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 μs de resolução.
dwMinScanTime	DWORD	Tempo mínimo de ciclo da tarefa com 1 μs de resolução.
dwMaxScanTime	DWORD	Tempo máximo de ciclo da tarefa com 1 μs de resolução.
dwAvgScanTime	DWORD	Tempo médio de ciclo da tarefa com 1 μs de resolução.

Parâmetros de saída	Tipo	Descrição
dwLimitMaxScan	DWORD	Tempo máximo de ciclo da tarefa antes de ocorrer o cão-de-guarda.
dwIECCycleCount	DWORD	Contador de ciclos IEC decorridos.

Tabela 204: 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.
```

6.3. Manutenção Preventiva

- Deve ser verificado, anualmente, se os cabos de interconexão estão firmemente conectados, sem acúmulo de poeira, principalmente os dispositivos de proteção.
- Em ambientes sujeitos a contaminação excessiva, o equipamento deve ser limpo periodicamente de poeira, detritos, etc.
- Verificação de estanqueidade e limpeza do barramento a cada seis meses.

7. Apendices

7.1. Gerenciamento de Certificados e Chaves TLS

Esta secção 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.

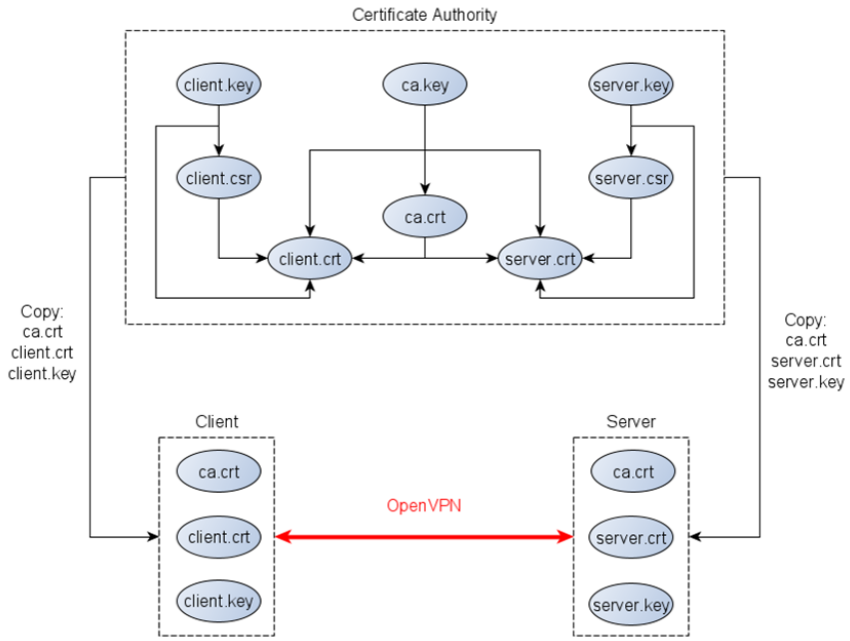


Figura 224: Fluxo de Geração de Certificados TLS

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 um 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 './easyrsa' to call the program. Without commands, help is displayed.

EasyRSA Shell
# _
```

Figura 225: Geração de Certificado utilizando o Easy-RSA (passo 1)

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

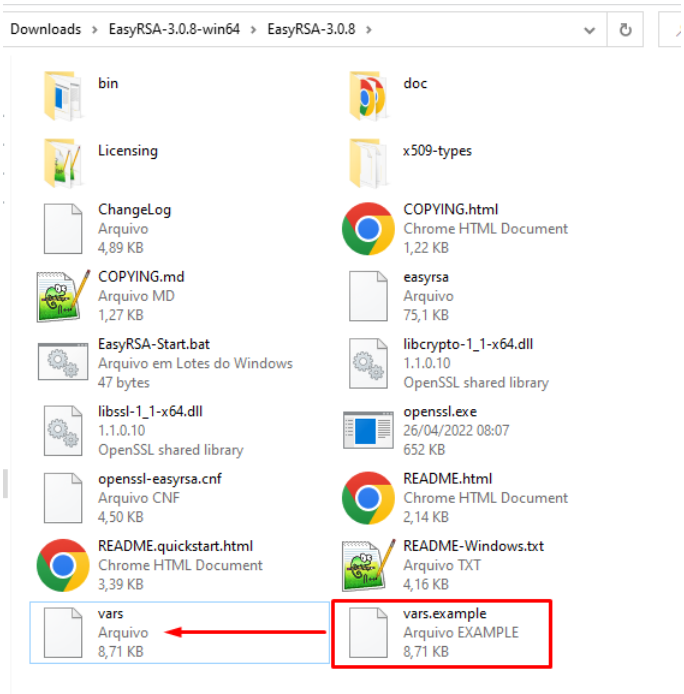
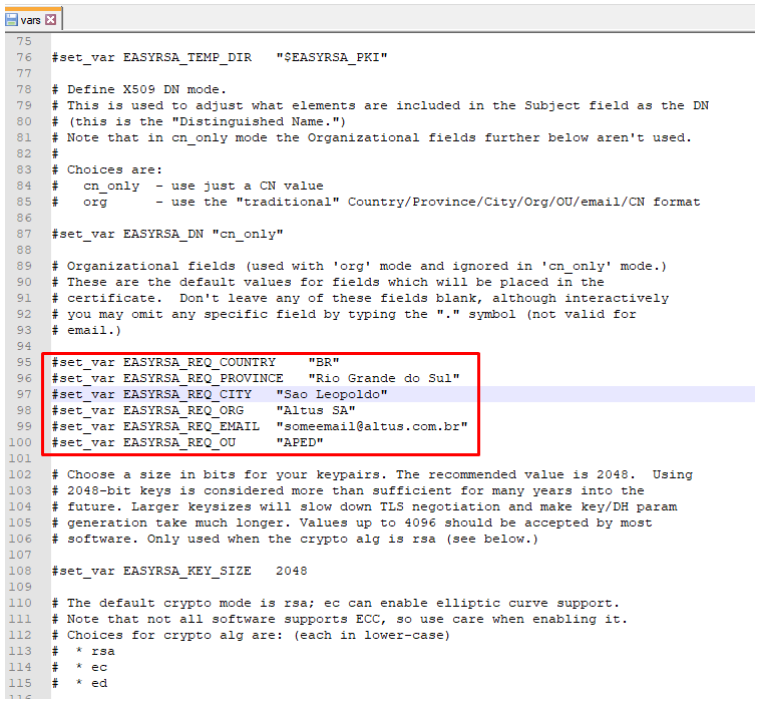


Figura 226: Geração de Certificado utilizando o Easy-RSA (passo 2)

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


vars

```
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 keysizes 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 227: Geração de Certificado utilizando o Easy-RSA (passo 3)

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

```
./easyrsa init-pki
```



Figura 228: Geração de Certificado utilizando o Easy-RSA (passo 4)

5- Em seguida, digite o seguinte para gerar o certificado CA.

```
./easyrsa build-ca nopass
```

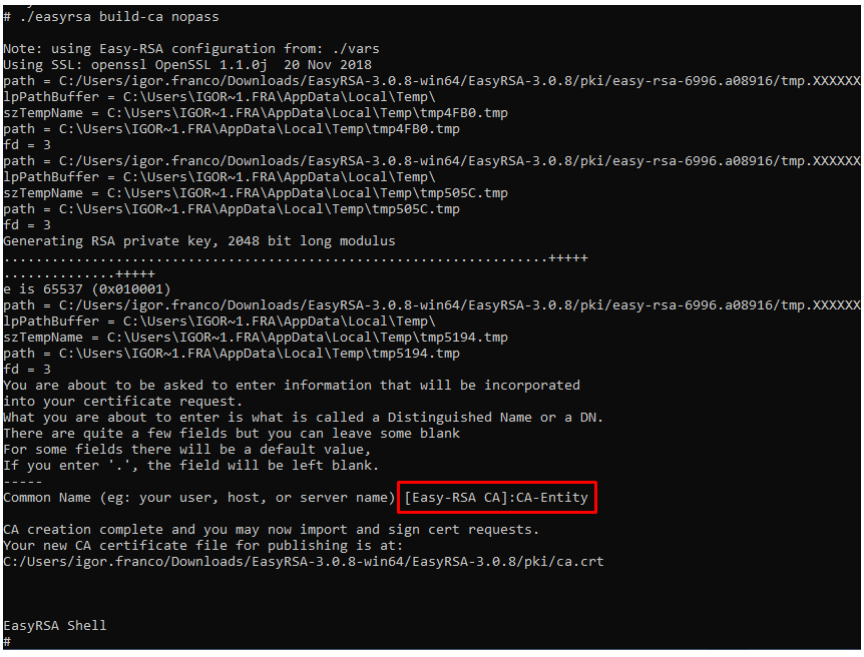


Figura 229: Geração de Certificado utilizando o Easy-RSA (passo 5)

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).

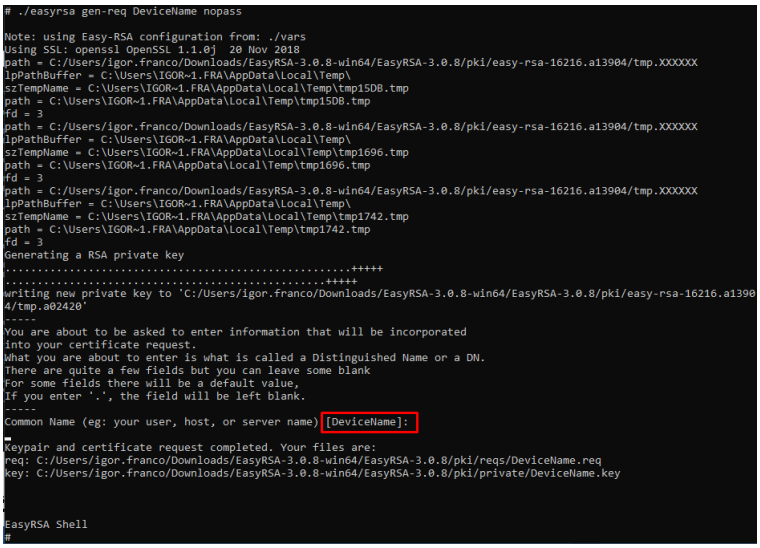


Figura 230: Geração de Certificado utilizando o Easy-RSA (passo 6)

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
```

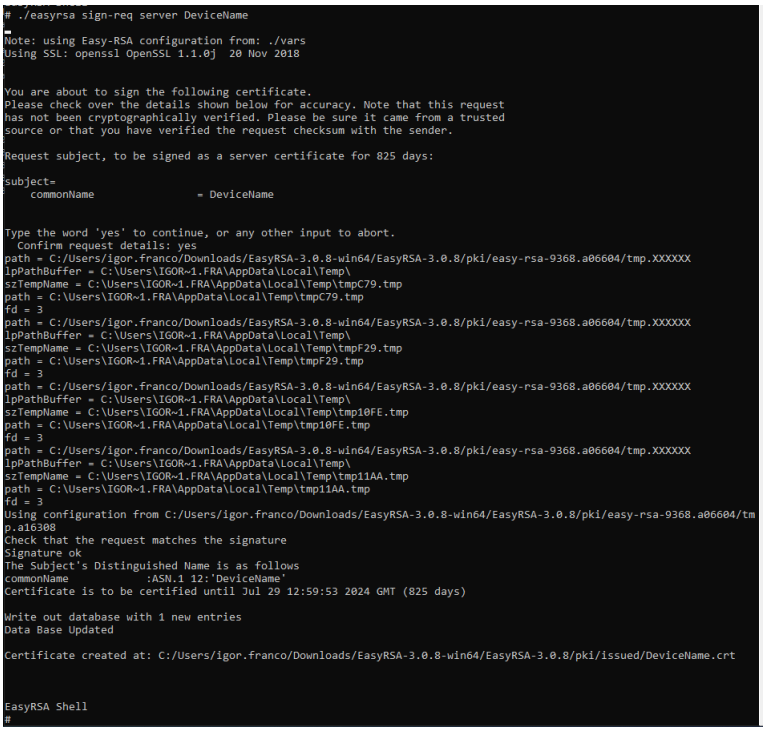


Figura 231: Geração de Certificado utilizando o Easy-RSA (passo 7)

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*.

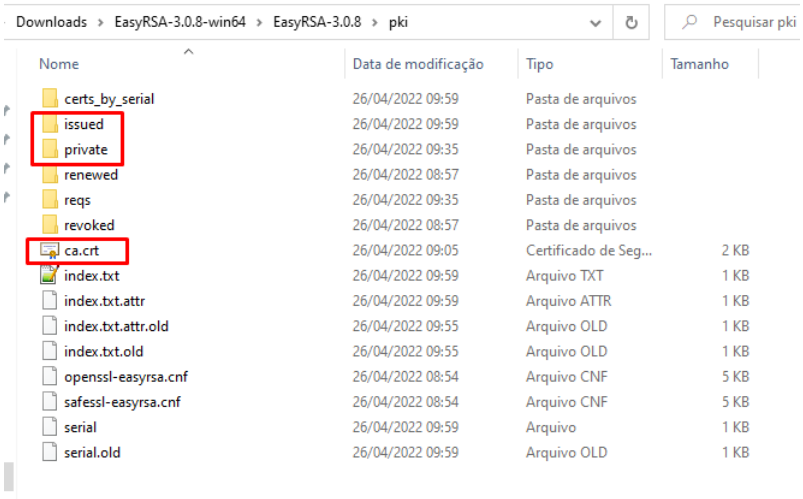


Figura 232: Geração de Certificado utilizando o Easy-RSA (passo 9)

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
```

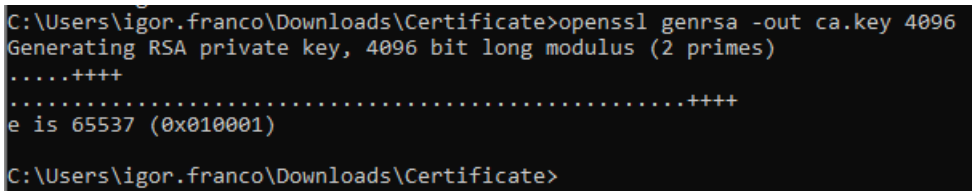


Figura 233: Geração de Certificado utilizando o OpenSSL (passo 2)

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.

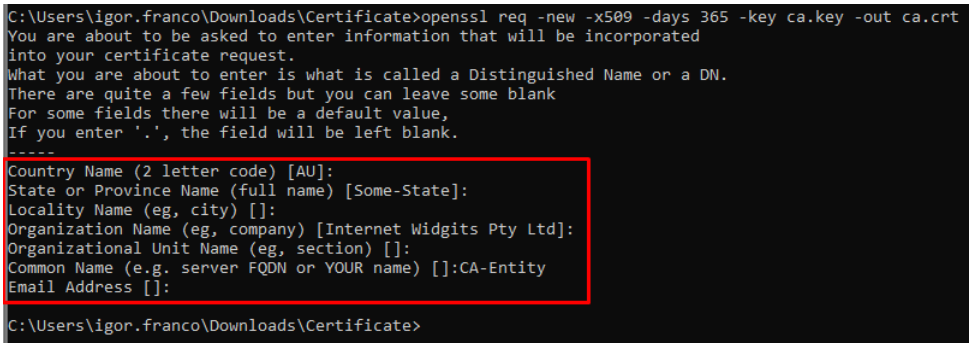


Figura 234: Geração de Certificado utilizando o OpenSSL (passo 3)

4- Agora, gere a chave privada do dispositivo, semelhante à etapa 2, utilizando o seguinte comando:

```
openssl genrsa -out DeviceName.key 2048
```

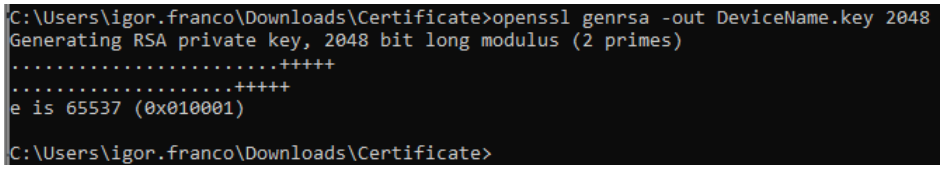


Figura 235: Geração de Certificado utilizando o OpenSSL (passo 4)

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.

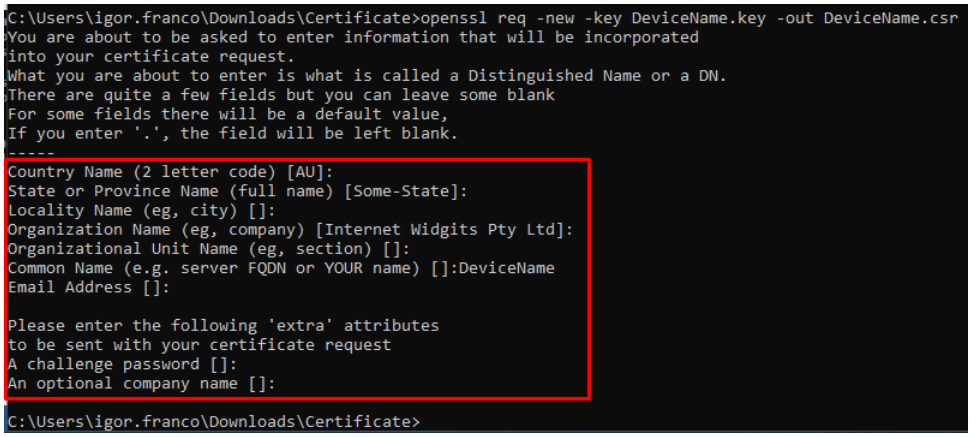


Figura 236: Geração de Certificado utilizando o OpenSSL (passo 5)

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*.

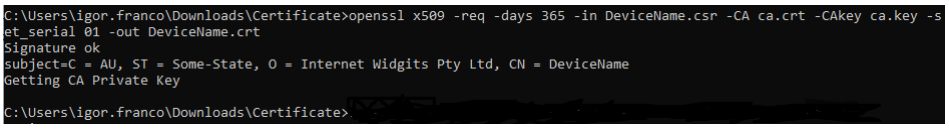


Figura 237: Geração de Certificado utilizando o OpenSSL (passo 6)

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
```

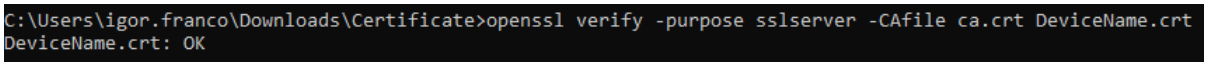


Figura 238: Geração de Certificado utilizando o OpenSSL (passo 8)

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 utilziação 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
```

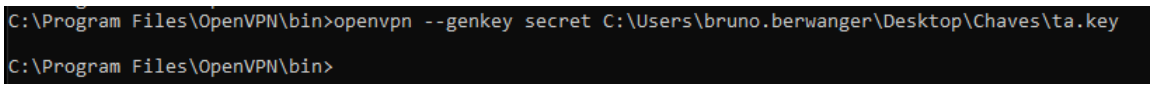


Figura 239: Exemplo de Geração de Chave TA no Windows

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
```

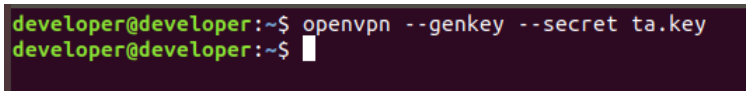


Figura 240: Exemplo de Geração de Chave TA no Linux

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.