Manual de Utilização
UCP Série Nexto
NX3020

MU214102 Rev. F

27 de fevereiro de 2025

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

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

4.14.1. Programa MainPrg . . . 51
4.14.2. Programa StartPrg . . . 52
4.14.3. Programa UserPrg . . . 52
4.14.4. GVL System_Diagnostics . . . 52
4.14.5. GVL Disables . . . 53
4.14.6. GVL IOQualities . . . 54
4.14.7. GVL Module_Diagnostics . . . 55
4.14.8. GVL Qualities . . . 55
4.14.9. GVL ReqDiagnostics . . . 57
4.14.10. Função Prepare_Start . . . 59
4.14.11. Função Prepare_Stop . . . 59
4.14.12. Função Start_Done . . . 59
4.14.13. Função Stop_Done . . . 59
5. Configuração . . . 60
5.1. Device . . . 60
5.1.1. Gerenciamento de Usuários e Direitos de Acesso . . . 60
5.1.2. Configurações do CP . . . 60
5.2. Configuração da UCP . . . 62
5.2.1. Parâmetros Gerais . . . 62
5.2.1.1. Troca a Quente . . . 64
5.2.1.1.1. Troca a Quente Desabilitada, Apenas para Módulos Declarados . . . 64
5.2.1.1.2. Troca a Quente Desabilitada . . . 65
5.2.1.1.3. Troca a Quente Desabilitada, sem Consistência na Partida . . . 65
5.2.1.1.4. Troca a Quente Habilitada, com Consistência na Partida Apenas para Módulos Declarados . . . 65
5.2.1.1.5. Troca a Quente Habilitada, com Consistência na Partida . . . 65
5.2.1.1.6. Troca a Quente Habilitada, sem Consistência na Partida . . . 66
5.2.1.1.7. Como realizar a Troca a Quente . . . 66
5.2.1.2. Áreas de Memória Retentiva e Persistente . . . 68
5.2.1.3. Parâmetros do Projeto . . . 69
5.2.2. Configuração de Evento Externo . . . 70
5.2.3. Configuração do SOE . . . 71
5.2.4. Sincronização de Tempo . . . 74
5.2.4.1. IEC 60870-5-104 . . . 75
5.2.4.2. SNTP . . . 75
5.2.4.3. Horário de Verão . . . 76
5.2.5. Pontos Internos . . . 76
5.2.5.1. Conversões de Qualidade . . . 78
5.2.5.1.1. Qualidade Interna . . . 79
5.2.5.1.2. Conversão IEC 60870-5-104 . . . 80
5.2.5.1.3. Qualidade Interna MODBUS . . . 81
5.2.5.1.4. Qualidade dos Módulos de E/S do Barramento Local . . . 82
5.2.5.1.5. Qualidade dos Módulos de E/S do Barramento PROFIBUS . . . 82
5.2.5.1.6. Qualidade da Entrada Digital PROFIBUS . . . 83
5.2.5.1.7. Qualidade da Saída Digital PROFIBUS . . . 84
5.2.5.1.8. Qualidade da Entrada Analógica PROFIBUS . . . 85
5.2.5.1.9. Qualidade da Saída Analógica PROFIBUS . . . 86
5.3. Configuração das Interfaces Seriais . . . 88

5.3.1. COM 1 . . . 88
5.3.1.1. Configurações Avançadas . . . 90
5.3.2. COM 2 . . . 91
5.3.2.1. Configurações Avançadas . . . 92
5.4. Configuração das Interfaces Ethernet . . . 92
5.4.1. Interfaces Ethernet Integradas . . . 92
5.4.1.1. NET 1 . . . 93
5.4.1.2. NET 2 . . . 93
5.4.1.3. Modos de Operação das Interfaces Ethernet Integradas . . . 93
5.4.1.3.1. Modo Simples . . . 93
5.4.1.3.2. Modo Redundante . . . 94
5.4.2. Interface Ethernet Remota NX5000 . . . 95
5.4.2.1. NET 1 . . . 95
5.4.2.2. Modos de Operação da Interface Ethernet Remota NX5000 . . . 95
5.4.2.2.1. Modo Redundante . . . 95
5.4.3. Portas TCP/UDP Reservadas . . . 96
5.5. Configuração de Protocolos . . . 96
5.5.1. Comportamento dos Protocolos x Estados da UCP . . . 99
5.5.2. Pontos Duplos . . . 100
5.5.3. Fila de Eventos da UCP . . . 100
5.5.3.1. Consumidores . . . 101
5.5.3.2. Princípios de Funcionamento da Fila . . . 101
5.5.3.2.1. Sinalização de Estouro . . . 102
5.5.3.3. Produtores . . . 102
5.5.4. Interceptação de Comandos Oriundos do Centro de Controle . . . 102
5.5.5. MODBUS RTU Mestre . . . 107
5.5.5.1. Configuração do Protocolo MODBUS Mestre por Mapeamento Simbólico . . . 107
5.5.5.1.1. Parâmetros Gerais do Protocolo MODBUS Mestre – Configuração por Mapeamento Simbólico . . . 107
5.5.5.1.2. Configuração dos Dispositivos – Configuração por Mapeamento Simbólico 109
5.5.5.1.3. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico 110
5.5.5.1.4. Configuração das Requisições – Configuração por Mapeamento Simbólico . 111
5.5.5.2. Configuração do Protocolo MODBUS Mestre por Representação Direta (%Q) . . . 115
5.5.5.2.1. Parâmetros Gerais do Protocolo MODBUS Mestre – Configuração por Representação Direta (%Q) . . . 116
5.5.5.2.2. Configuração dos Dispositivos – Configuração por Representação Direta (%Q) . . . 117
5.5.5.2.3. Configuração dos Mapeamentos – Configuração por Representação Direta (%Q) . . . 118
5.5.6. MODBUS RTU Escravo . . . 120
5.5.6.1. Configuração do Protocolo MODBUS Escravo por Mapeamento Simbólico . . . 120
5.5.6.1.1. Parâmetros Gerais do Protocolo MODBUS Escravo – Configuração por Mapeamento Simbólico . . . 120
5.5.6.1.2. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico 123
5.5.6.2. Configuração do Protocolo MODBUS Escravo por Representação Direta (%Q) . . 124
5.5.6.2.1. Parâmetros Gerais do Protocolo MODBUS Escravo – Configuração por Representação Direta (%Q) . . . 124
5.5.6.2.2. Configuração dos Mapeamentos – Configuração por Representação Direta (%Q) . . . 126
5.5.7. MODBUS Ethernet . . . 127

5.5.8. MODBUS Ethernet Cliente . . . 129
5.5.8.1. Configuração do Protocolo MODBUS Ethernet Cliente por Mapeamento Simbólico 129
5.5.8.1.1. Parâmetros Gerais do Protocolo MODBUS Cliente – Configuração por Mapeamento Simbólico . . . 130
5.5.8.1.2. Configuração dos Dispositivos – Configuração por Mapeamento Simbólico 131
5.5.8.1.3. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico 132
5.5.8.1.4. Configuração das Requisições – Configuração por Mapeamento Simbólico . 134
5.5.8.2. Configuração do Protocolo MODBUS Ethernet Cliente por Representação Direta (%Q) . . . 137
5.5.8.2.1. Parâmetros Gerais do Protocolo MODBUS Cliente – Configuração por Representação Direta (%Q) . . . 138
5.5.8.2.2. Configuração dos Dispositivos – configuração por Representação Direta (%Q) . . . 139
5.5.8.2.3. Configuração dos Mapeamentos – Configuração por Representação Direta (%Q) . . . 139
5.5.8.3. Disparo de Relações MODBUS Cliente de Forma Acíclica . . . 142
5.5.9. MODBUS Ethernet Servidor . . . 142
5.5.9.1. Configuração do Protocolo MODBUS Ethernet Servidor por Mapeamento Simbólico 142
5.5.9.1.1. Parâmetros Gerais do Protocolo MODBUS Servidor – Configuração por Mapeamento Simbólico . . . 142
5.5.9.1.2. Diagnósticos MODBUS Servidor – Configuração por Mapeamento Simbólico . . . 144
5.5.9.1.3. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico 145
5.5.9.2. Configuração do Protocolo MODBUS Ethernet Servidor por Representação Direta (%Q) . . . 146
5.5.9.2.1. Parâmetros Gerais do Protocolo MODBUS Servidor – Configuração por Representação Direta (%Q) . . . 146
5.5.9.2.2. Configuração dos Mapeamentos – Configuração por Representação Direta (%Q) . . . 148
5.5.10. OPC DA Servidor . . . 150
5.5.10.1. Criando um Projeto para Comunicação OPC DA . . . 152
5.5.10.2. Configurando um CP no Servidor OPC DA . . . 155
5.5.10.2.1. Importando uma Configuração do Projeto . . . 157
5.5.10.3. Variáveis de Status e Qualidade da Comunicação OPC DA . . . 157
5.5.10.4. Limites da Comunicação com Servidor OPC DA . . . 159
5.5.10.5. Acessando Dados Através de um Cliente OPC DA . . . 159
5.5.11. OPC UA Servidor . . . 161
5.5.11.1. Criando um Projeto para Comunicação OPC UA . . . 162
5.5.11.2. Tipos de Variáveis Suportadas . . . 164
5.5.11.3. Limite de Clientes Conectados no Servidor OPC UA . . . 164
5.5.11.4. Limite de Variáveis de Comunicação no Servidor OPC UA . . . 164
5.5.11.5. Configurações de Criptografia . . . 164
5.5.11.6. Principais Parâmetros de Comunicação Ajustados em um Cliente OPC UA . . . 165
5.5.11.6.1. Endpoint URL . . . 165
5.5.11.6.2. Publishing Interval (ms) e Sampling Interval (ms) . . . 165
5.5.11.6.3. Lifetime Count e Keep-Alive Count . . . 166
5.5.11.6.4. Queue Size e Discard Oldest . . . 166
5.5.11.6.5. Filter Type e Deadband Type . . . 166
5.5.11.6.6. PublishingEnabled, MaxNotificationsPerPublish e Priority . . . 167
5.5.11.7. Acessando Dados Através de um Cliente OPC UA . . . 167
5.5.12. Mestre EtherCAT . . . 168

5.5.12.1. Instalando e Inserindo Dispositivos EtherCAT . . . 168
5.5.12.1.1. EtherCAT - Scan Devices . . . 169
5.5.12.2. Configuração do Mestre EtherCAT . . . 170
5.5.12.2.1. EtherCAT Master - General . . . 170
5.5.12.2.2. EtherCAT Master - Sync Unit Assignment . . . 172
5.5.12.2.3. EtherCAT Master - Overview . . . 172
5.5.12.2.4. EtherCAT Master - I/O Mapping . . . 172
5.5.12.2.5. EtherCAT Master - IEC Objects . . . 172
5.5.12.2.6. EtherCAT Master - Status / Informations . . . 172
5.5.12.3. Configuração do Escravo EtherCAT . . . 172
5.5.12.3.1. EtherCAT Slave - General . . . 172
5.5.12.3.2. EtherCAT Slave - Process Data . . . 176
5.5.12.3.3. EtherCAT Slave - Edit PDO List . . . 178
5.5.12.3.4. EtherCAT Slave - Startup Parameters . . . 178
5.5.12.3.5. EtherCAT Slave - Module I/O Mapping . . . 178
5.5.12.3.6. EtherCAT Slave - Status and Information . . . 179
5.5.13. EtherNet/IP . . . 179
5.5.13.1. Interface EtherNet/IP . . . 180
5.5.13.2. Configuração do Scanner EtherNet/IP . . . 182
5.5.13.2.1. Geral . . . 182
5.5.13.2.2. Conexões . . . 183
5.5.13.2.3. Assemblies . . . 185
5.5.13.2.4. EtherNet/IP: Mapeamento de E/S . . . 186
5.5.13.3. Configuração do Adapter EtherNet/IP . . . 186
5.5.13.3.1. Geral . . . 186
5.5.13.3.2. EtherNet/IP Adapter: Mapeamento de E/S . . . 187
5.5.13.4. Configuração do Módulo EtherNet/IP . . . 187
5.5.13.4.1. Assemblies . . . 188
5.5.13.4.2. EtherNet/IP Module: Mapeamento de E/S . . . 188
5.5.14. IEC 60870-5-104 Servidor . . . 188
5.5.14.1. Tipos de Dados . . . 188
5.5.14.2. Pontos Duplos . . . 190
5.5.14.2.1. Pontos Duplos de Entrada Digital . . . 190
5.5.14.2.2. Pontos Duplos de Saída Digital . . . 192
5.5.14.3. Parâmetros Gerais . . . 197
5.5.14.4. Mapeamento dos Dados . . . 197
5.5.14.5. Camada de Enlace . . . 200
5.5.14.6. Camada de Aplicação . . . 201
5.5.14.7. Diagnósticos do Servidor . . . 203
5.5.14.8. Qualificador dos Comandos . . . 204
5.5.15. PROFINET Controller . . . 205
5.6. Desempenho de Comunicação . . . 205
5.6.1. Servidor MODBUS . . . 205
5.6.1.1. Interfaces Locais da UCP . . . 206
5.6.1.2. Interfaces Remotas . . . 207
5.6.2. Servidor OPC DA . . . 207
5.6.3. Servidor OPC UA . . . 207
5.6.4. Servidor IEC 60870-5-104 . . . 207

5.7. Desempenho do Sistema . . . 208
5.7.1. Escaneamento de E/S . . . 208
5.7.2. Cartão de Memória . . . 209
5.8. Relógio RTC . . . 209
5.8.1. Blocos Funcionais e Funções para Leitura e Escrita do RTC . . . 209
5.8.1.1. Funções de Leitura do RTC . . . 210
5.8.1.1.1. GetDateAndTime . . . 210
5.8.1.1.2. GetTimeZone . . . 210
5.8.1.1.3. GetDayOfWeek . . . 211
5.8.1.2. Funções de Escrita do RTC . . . 212
5.8.1.2.1. SetDateAndTime . . . 212
5.8.1.2.2. SetTimeZone . . . 214
5.8.2. Estruturas de Dados do RTC . . . 215
5.8.2.1. EXTENDED_DATE_AND_TIME . . . 215
5.8.2.2. DAYS_OF_WEEK . . . 215
5.8.2.3. RTC_STATUS . . . 216
5.8.2.4. TIMEZONESETTINGS . . . 216
5.9. Memória de Arquivos de Usuário . . . 216
5.10. Cartão de Memória . . . 219
5.10.1. Preparação do Projeto . . . 219
5.10.2. Envio do Projeto . . . 220
5.10.3. Acesso no MasterTool . . . 221
5.11. Menu Informativo e de Configuração da UCP . . . 222
5.12. Blocos Funcionais e Funções . . . 224
5.12.1. Blocos Funcionais Especiais para Comunicação Serial . . . 224
5.12.1.1. SERIAL_CFG . . . 228
5.12.1.2. SERIAL_GET_CFG . . . 230
5.12.1.3. SERIAL_GET_CTRL . . . 231
5.12.1.4. SERIAL_GET_RX_QUEUE_STATUS . . . 233
5.12.1.5. SERIAL_PURGE_RX_QUEUE . . . 235
5.12.1.6. SERIAL_RX . . . 236
5.12.1.7. SERIAL_RX_EXTENDED . . . 238
5.12.1.8. SERIAL_SET_CTRL . . . 241
5.12.1.9. SERIAL_TX . . . 242
5.12.2. Atualização de Entradas e Saídas . . . 245
5.12.2.1. REFRESH_INPUT . . . 245
5.12.2.2. REFRESH_OUTPUT . . . 246
5.12.3. Bloco Funcional PID . . . 248
5.12.4. Temporizador Retentivo . . . 248
5.12.4.1. TOF_RET . . . 248
5.12.4.2. TON_RET . . . 250
5.12.4.3. TP_RET . . . 251
5.12.5. Log de Usuário . . . 252
5.12.5.1. UserLogAdd . . . 253
5.12.5.2. UserLogDeleteAll . . . 255
5.12.6. ClearRtuDiagnostic . . . 256
5.12.7. ClearEventQueue . . . 256
5.13. SNMP . . . 256

5.13.1. Introdução . . . 256
5.13.2. SNMP nas UCPs Nexto . . . 257
5.13.3. Private MIB . . . 258
5.13.4. Configuração SNMP . . . 258
5.13.5. Usuário e Comunidades SNMP . . . 260
6. Manutenção . . . 261
6.1. Diagnósticos do Módulo . . . 261
6.1.1. One Touch Diag . . . 261
6.1.2. Diagnósticos via LED . . . 264
6.1.2.1. DG (Diagnóstico) . . . 264
6.1.2.2. WD (Cão-de-guarda) . . . 264
6.1.2.3. LEDs Conector RJ45 . . . 265
6.1.3. Diagnósticos via Página Web de Sistema . . . 265
6.1.4. Diagnósticos via Variáveis . . . 267
6.1.4.1. Diagnósticos Resumidos . . . 268
6.1.4.2. Diagnósticos Detalhados . . . 270
6.1.5. Diagnósticos via Blocos Funcionais . . . 277
6.1.5.1. GetTaskInfo . . . 277
6.2. Visor Gráfico . . . 279
6.3. Log de Sistema . . . 281
6.4. Não Carregar a Aplicação na Inicialização . . . 281
6.5. Falha na Alimentação . . . 281
6.6. Problemas mais Comuns . . . 282
6.7. Solução de Problemas . . . 283
6.8. Manutenção Preventiva . . . 283
7. Anexo - Interoperabilidade DNP3 . . . 284
7.1. DNP3 Device Profile . . . 284
7.2. DNP V3.0 Implementation Table . . . 285

1. Introdução

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

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

Figura 1: NX3020

1.1. Série Nexto

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

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

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

Figura 2: Série Nexto – Visão Geral

1.2. Características Inovadoras

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

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

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

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

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

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

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

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

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

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

1.3. Documentos Relacionados a este Manual

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

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

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

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

Código	Descrição	Idioma
CE114000	Nexto Series – Technical Characteristics	Inglês
CT114000	Série Nexto – Características Técnicas	Português
CE114101	NX3020 Technical Characteristics	Inglês
CT114101	Características Técnicas NX3020	Português
CE114200	NX8000 Power Supply Module Technical Characteristics	Inglês
CT114200	Características Técnicas Fonte de Alimentação NX8000	Português
CE114700	Nexto Series Backplane Racks Technical Characteristic	Inglês
CT114700	Características Técnicas dos Bastidores da Série Nexto	Português
CE114810	Nexto Series Accessories for Backplane Rack Technical Characteristics	Inglês
CT114810	Características Técnicas Acessórios para Bastidor Série Nexto	Português
CE114902	Nexto Series PROFIBUS-DP Master Technical Characteristics	Inglês
CT114902	Características Técnicas do Mestre PROFIBUS-DP da Série Nexto	Português
CE114903	Nexto Series Ethernet Module Technical Characteristics	Inglês
CT114903	Características Técnicas Módulo Ethernet Série Nexto	Português
CE114908	NX5110 and NX5210 PROFIBUS-DP Heads Technical Characteristics	Inglês
CT114908	Características Técnicas Interfaces Cabeça PROFIBUSDP NX5110 e NX5210	Português
MU214600	Nexto Series User Manual	Inglês
MU214000	Manual de Utilização Série Nexto	Português
MU214614	NX3020 CPU User Manual	Inglês
MU214102	Manual de Utilização UCP NX3020	Português
MU299609	MasterTool IEC XE User Manual	Inglês
MU299048	Manual de Utilização MasterTool IEC XE	Português
MP399609	MasterTool IEC XE Programming Manual	Inglês
MP399048	Manual de Programação MasterTool IEC XE	Português

Código	Descrição	Idioma
MU214601	NX5001 PROFIBUS DP Master User Manual	Inglês
MU214001	Manual de Utilização Mestre PROFIBUS-DP NX5001	Português
MU214608	Nexto PROFIBUS-DP Head Utilization Manual	Inglês
MU214108	Manual de Utilização da Cabeça PROFIBUS-DP Nexto	Português
MU223603	IEC 60870-5-104 Server Device Profile Document	Inglês
MU214603	Nexto Series HART Manual	Inglês
MU214606	MQTT User Manual	Inglês
MU214609	OPC UA Server for Altus Controllers User Manual	Inglês
MU214610	PID - Advanced Control Functions User Manual	Inglês
MU214621	Nexto Series PROFINET Manual	Inglês
NAP151	Utilização do Tunneller OPC	Português

Tabela 1: Documentos Relacionados

1.4. Inspeção Visual

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

CUIDADO

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

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

1.5. Suporte Técnico

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

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

1.6. Mensagens de Advertência Utilizadas neste Manual

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

PERIGO

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

CUIDADO

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

ATENÇÃO

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

2. Descrição Técnica

Este capítulo apresenta todas as características técnicas da UCP NX3020 da Série Nexto.

2.1. Painéis e Conexões

A figura a seguir mostra o painel frontal da UCP.

Figura 3: NX3020

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

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

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

Tabela 2: Descrição dos LEDs

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

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

Tabela 3: Descrição das Teclas

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

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

Tabela 4: Interfaces de Conexão

2.2. Características do Produto

2.2.1. Características Gerais

	NX3020
Ocupação do bastidor	2 posições sequenciais
Fonte de alimentação integrada	Não
Ethernet TCP/IP interface local	2
Interface Serial	2
Interface CAN	Não
Porta USB Host	Não
Interface Cartão de Memória	Sim
Relógio de tempo real (RTC)	Sim Resolução de 1 ms, máx. variação de 2 segundos por dia.
Cão de guarda	Sim
Indicação de estado e diagnóstico	Visor gráfico LEDs Páginas Web de Sistema Memória interna da UCP
Linguagens de programação	Texto Estruturado (ST) Diagrama Ladder (LD) Sequenciamento Gráfico de Funções (SFC) Diagrama de Blocos Funcionais (FBD) Gráfico Contínuo de Funções (CFC)
Tipos de tarefas	Cíclica (periódica) Disparada por evento (interrupção de software) Disparada por evento externo (interrupção de hardware) Contínua (execução livre) Disparada por estado (interrupção de software)
Alterações online	Sim
Número máximo de tarefas	24
Número máximo de barramentos de expansão	24
Suporte a redundância de expansão de barramento	Sim
Número máximo total de módulos de E/S nos barramentos	128
Número máximo de módulos adicionais de interface Ethernet TCP/IP	2
Suporte a redundância de interfaces Ethernet TCP/IP	Sim
Número máximo de redes PROFIBUS-DP (usando módulos mestres PROFIBUS-DP)	4
Suporte a redundância de rede PROFIBUS-DP	Sim
Suporte a redundância (half-clusters)	Não
Suporte a troca a quente	Sim
Registro de eventos (SOE) **Protocolo** **Tamanho máximo da fila de eventos**	Sim DNP3 1000
Desenvolvimento de páginas Web (acessíveis através de protocolo HTTP)	Não

	NX3020
One Touch Diag (OTD)	Sim
Electronic Tag on Display (ETD)	Sim

Tabela 5: Características Gerais

Notas:

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

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

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

2.2.2. Normas e Certificações

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

Tabela 6: Normas e Certificações

2.2.3. Memória

	NX3020
Memória de variáveis de entrada de representação direta (%I)	64 Kbytes
Memória de variáveis de saída de representação direta (%Q)	64 Kbytes
Memória de variáveis de representação direta (%M)	32 Kbytes
Memória de variáveis simbólicas	5 Mbytes
Memória de variáveis retentivas ou persistentes	112 Kbytes
Memória de dados redundantes total	-
Memória de variáveis de entrada de representação direta (%I)	-
Memória de variáveis de saída de representação direta (%Q)	-
Memória de variáveis de representação direta (%M)	-
Memória de variáveis simbólicas	-
Memória de programa	6 Mbytes
Memória de código fonte (backup)	80 Mbytes
Memória de arquivos de usuário	32 Mbytes

Tabela 7: Memória

Notas:

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

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

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 IEC XE.

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

ATENÇÃO

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

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

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

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

Tabela 8: Comportamento das Variáveis após Evento

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

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

	NX3020
Memória de variáveis de saída de representação direta retentivas (%Q)	16 Kbytes
Memória de variáveis de saída de representação direta persistentes (%Q)	16 Kbytes
Memória de variáveis simbólicas retentivas	48 Kbytes
Memória de variáveis simbólicas persistentes	32 Kbytes

Tabela 9: Memórias Retentivas e Persistentes em Versões Antigas

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

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

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

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

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

2.2.4. Protocolos

	NX3020	Interface
Protocolo aberto	Sim	COM1 / COM2
MODBUS RTU Mestre	Sim	COM1 / COM2
MODBUS RTU Escravo	Sim	COM1 / COM2
MODBUS TCP Cliente	Sim	NET1 / NET2
MODBUS TCP Servidor	Sim	NET1 / NET2
MODBUS RTU via TCP Cliente	Sim	NET1 / NET2
MODBUS RTU via TCP Servidor	Sim	NET1 / NET2
CANopen Mestre	Não	-
CANopen Escravo	Não	-
CAN low level	Não	-
SAE J-1939	Não	-
OPC DA Servidor	Sim	NET1 / NET2
OPC UA Servidor	Sim	NET1 / NET2
EtherCAT Mestre	Sim	NET1 / NET2
SNMP Agente	Sim	NET1 / NET2
SOE (dados orientados ao evento)	Sim	NET1 / NET2
IEC 60870-5-104 Servidor	Sim	NET1 / NET2
EtherNet/IP Scanner	Sim	NET1 / NET2
EtherNet/IP Adapter	Sim	NET1 / NET2
MQTT Cliente	Sim	NET1 / NET2
SNTP Cliente (para sincronismo do relógio)	Sim	NET1 / NET2
PROFINET Controller	Sim	NET1 / NET2
PROFINET Device	Não	-

Tabela 10: Protocolos

Nota:

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

2.2.5. Interfaces Seriais

2.2.5.1. COM 1

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

Tabela 11: Características da Interface Serial COM 1

2.2.5.2. COM 2

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

Tabela 12: Características da Interface Serial COM 2

Nota:

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

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

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

2.2.6. Interfaces Ethernet

2.2.6.1. NET 1

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

Tabela 13: Características da Interface Ethernet NET 1

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

2.2.6.2. NET 2

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

Tabela 14: Características da Interface Ethernet NET 2

2.2.7. Interface do Cartão de Memória

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

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

Tabela 15: Características da Interface com Cartão de Memória

Notas:

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

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

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

2.2.8. Características Ambientais

	NX3020
Consumo de corrente no barramento da fonte de alimentação	1000 mA
Dissipação	5 W
Temperatura de operação	0 a 60 °C
Temperatura de armazenamento	-25 a 75 °C
Umidade relativa de operação e armazenamento	5% a 96%, sem condensação
Revestimento isolante de circuitos eletrônicos	Sim
Índice de proteção	IP 20
Dimensões do produto (L x A x P)	36,00 x 114,63 x 115,30 mm
Dimensões da embalagem (L x A x P)	44,00 x 122,00 x 147,00 mm
Peso	350 g
Peso com embalagem	400 g

Tabela 16: Características Ambientais

Nota:

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

2.3. Compatibilidade com Outros Produtos

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

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

Tabela 17: Compatibilidade com Outros Produtos

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

2.4. Desempenho

O desempenho das UCPs da Série Nexto depende:

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

2.4.1. Tempo de Intervalo da MainTask

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

2.4.2. Tempos de Aplicação

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

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

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

2.4.3. Tempos para Execução de Instruções

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

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

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

Tabela 18: Tempos de Instrução

2.4.4. Tempos de Inicialização

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

2.5. Dimensões Físicas

Dimensões em mm.

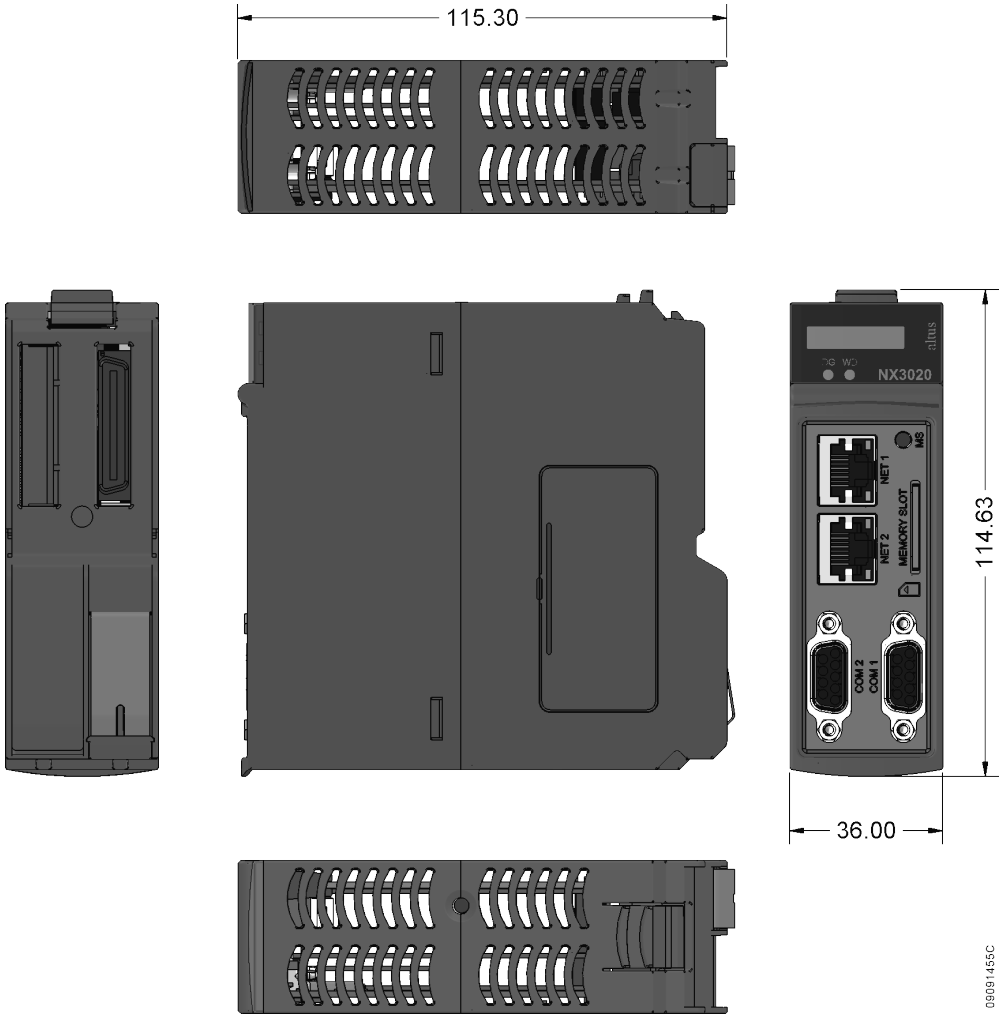


Figura 4: Dimensões Físicas

2.6. Dados para Compra

2.6.1. Itens Integrantes

A embalagem do produto contém os seguintes itens:

- Módulo NX3020

2.6.2. Código do Produto

Os seguintes códigos devem ser usados para compra do produto:

Código	Descrição
NX3020	UCP de alta velocidade, 2 portas Ethernet, 2 canais seriais, interface para cartão de memória e suporte à expansão de barramento

Tabela 19: Código do Produto

2.7. Produtos Relacionados

Os seguintes produtos devem ser adquiridos separadamente quando necessário:

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

Tabela 20: Produtos Relacionados

Notas:

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

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

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

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

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

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

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

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

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

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

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

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

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

3. Instalação

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

CUIDADO

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

3.1. Instalação Mecânica

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

3.2. Instalação Elétrica

PERIGO

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

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

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

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

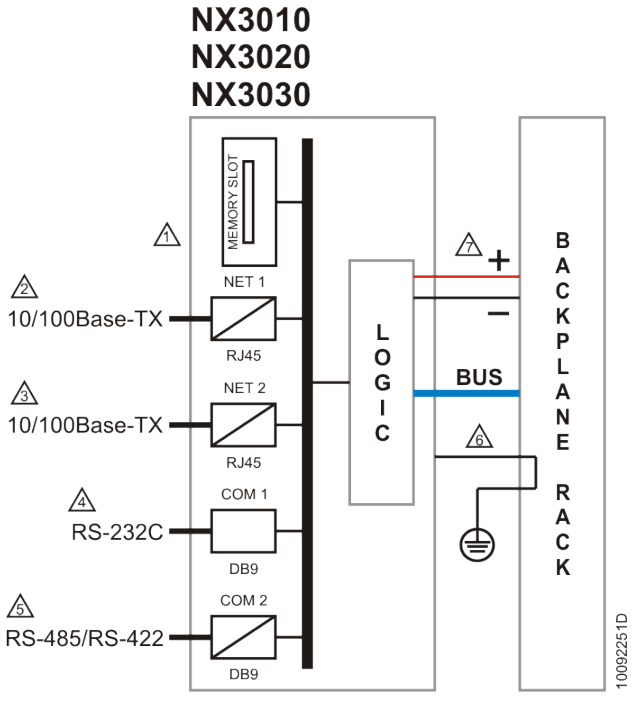


Figura 5: Diagrama Elétrico das UCPs NX3010, NX3020 e NX3030 da Série Nexto

Notas do Diagrama:

1. Interface para cartão de memória.

2. Interface Ethernet padrão 10/100Base-TX para programação, depuração e conexão à rede MODBUS TCP ou outros protocolos.

3. Interface Ethernet padrão 10/100Base-TX para conexão à rede MODBUS TCP ou outros protocolos (somente para NX3020 e NX3030).

4. Interface serial padrão RS-232C para conexão à rede MODBUS RTU ou outros protocolos.

5. Interface serial padrão RS-485/RS-422 para conexão à rede MODBUS RTU ou outros protocolos. A escolha do tipo de interface física depende do cabo utilizado.

6. O módulo é aterrado através dos bastidores da Série Nexto.

7. A alimentação do módulo é proveniente da conexão ao bastidor, não necessitando de conexões externas.

⏚ Terminal terra de proteção.

3.3. Conexão com a Rede Ethernet

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

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

3.3.1. Endereço IP

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

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

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

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

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

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

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

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

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

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

3.3.2. ARP Gratuito

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

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

3.3.3. Instalação do Cabo de Rede

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

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

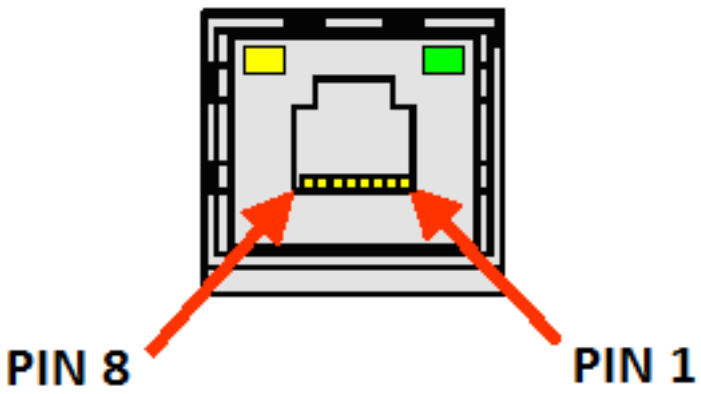


Figura 6: Conector Fêmea RJ45

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

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

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

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

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

3.4. Conexão com a Rede Serial RS-232C

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

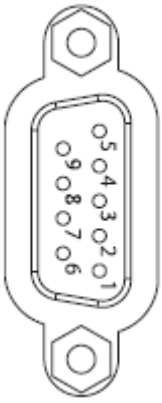


Figura 7: Conector DB9 Fêmea

Pino	Sinal	Descrição
1	DCD	Data Carrier Detect
2	TXD	Transmissão de dados
3	RXD	Recepção de dados
4	-	Não utilizado
5	GND	Ground

Pino	Sinal	Descrição
6	-	Não utilizado
7	CTS	Clear to Send
8	RTS	Request to Send
9	-	Não utilizado

Tabela 24: Pinagem do Conector DB9 Fêmea COM 1

3.4.1. Comunicação RS-232C

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

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

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

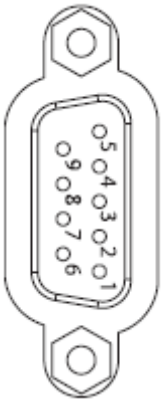


Figura 8: Conector DB9 Fêmea

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

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

3.5.1. Comunicação RS-485 sem Terminação

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

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

Tabela 26: Conexão da Interface RS-485 sem Terminação

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

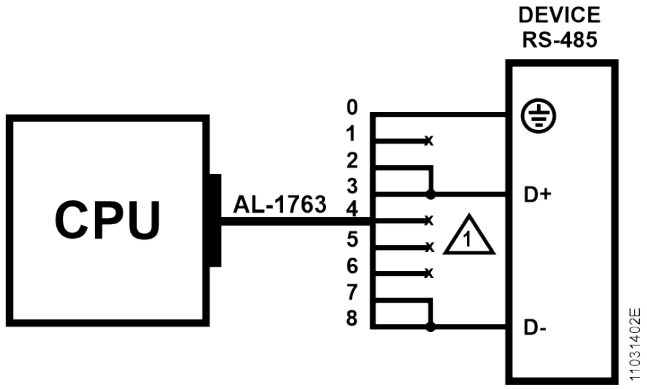


Figura 9: Diagrama de Conexão da Interface RS-485 sem Terminação

Nota do Diagrama:

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

3.5.2. Comunicação RS-485 com Terminação Interna

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

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

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

Obs.: As terminações internas disponíveis são do tipo estado seguro em modo aberto.
O diagrama da figura abaixo indica como deve ser realizada a conexão dos terminais do AL-1763 na borneira do dispositivo.

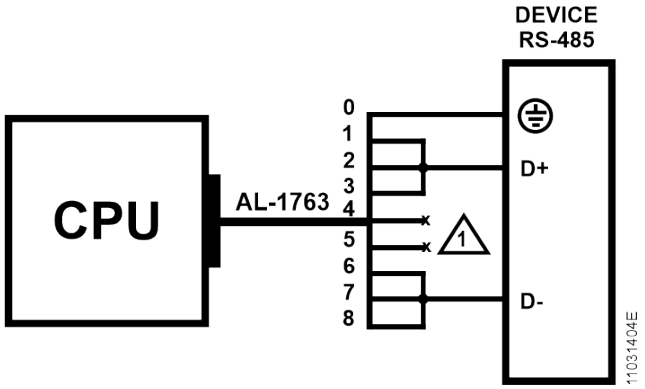


Figura 10: Diagrama de Conexão da Interface RS-485 com Terminação Interna

Nota do Diagrama:

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

3.5.3. Comunicação RS-485 com Terminação Externa

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

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

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

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

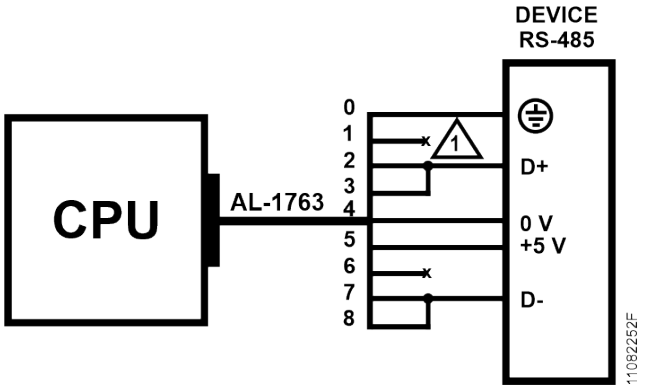


Figura 11: Diagrama de Conexão da Interface RS-485 com Terminação Externa

Nota do Diagrama:

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

3.5.4. Comunicação RS-422 sem Terminação

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

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

Tabela 29: Conexão da Interface RS-422 sem Terminação

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

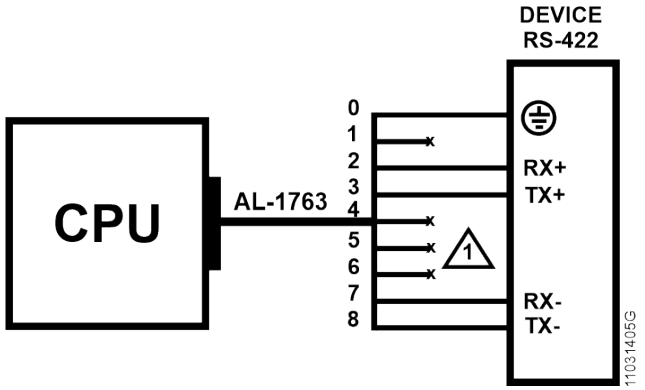


Figura 12: Diagrama de Conexão da Interface RS-422 sem Terminação

Nota do Diagrama:

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

3.5.5. Comunicação RS-422 com Terminação Interna

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

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

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

Obs.: As terminações internas disponíveis são do tipo estado seguro em modo aberto.
O diagrama da figura abaixo indica como deve ser realizada a conexão dos terminais do AL-1763 na borneira do dispositivo.

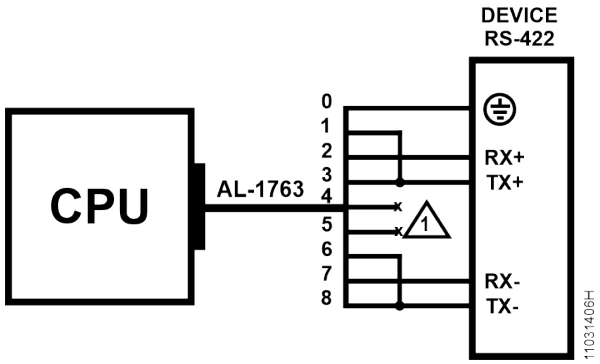


Figura 13: Diagrama de Conexão da Interface RS-422 com Terminação Interna

Nota do Diagrama:

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

3.5.6. Comunicação RS-422 com Terminação Externa

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

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

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

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

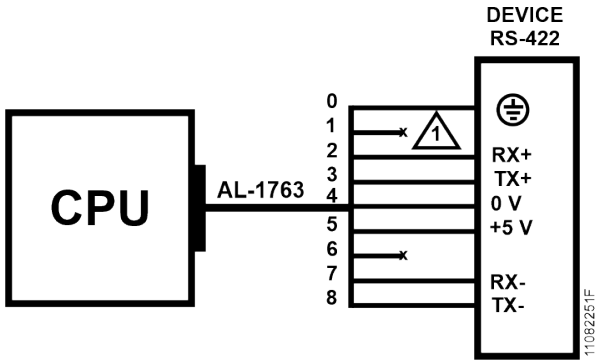


Figura 14: Diagrama de Conexão da Interface RS-422 com Terminação Externa

Nota do Diagrama:

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

3.5.7. Exemplo de Rede RS-422

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

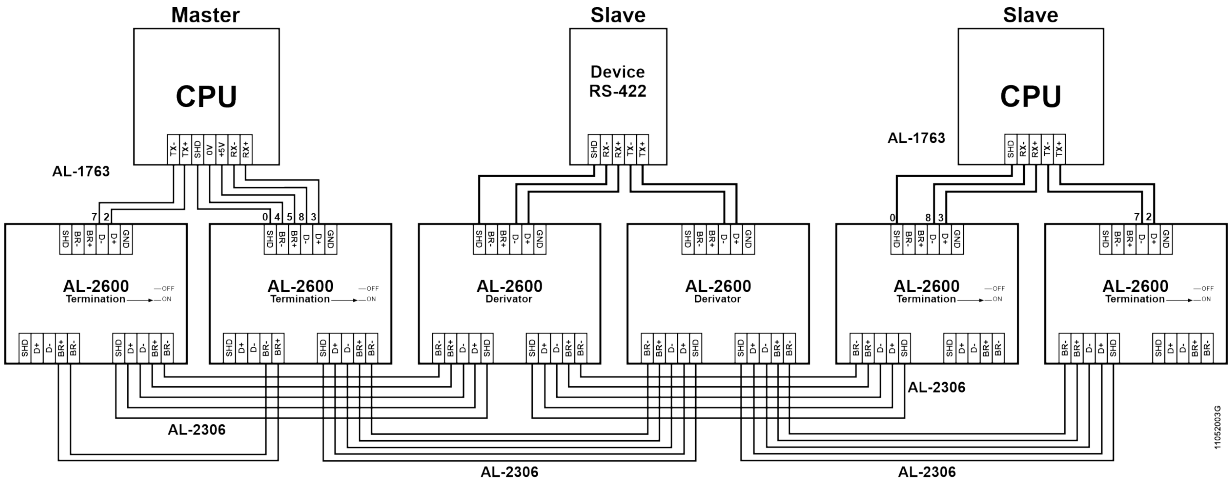


Figura 15: Exemplo de Rede RS-422

Nota do Diagrama:

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

3.6. Instalação do Cartão de Memória

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

Inicialmente, deve-se ter atenção quanto à posição correta que o cartão de memória deve ser inserido. Um dos cantos se diferencia dos demais e o mesmo deverá ser utilizado como referência para a inserção correta do cartão. Sendo assim, o cartão

de memória deverá ser colocado de acordo com o desenho localizado na parte frontal da UCP ou também como mostra a figura abaixo.

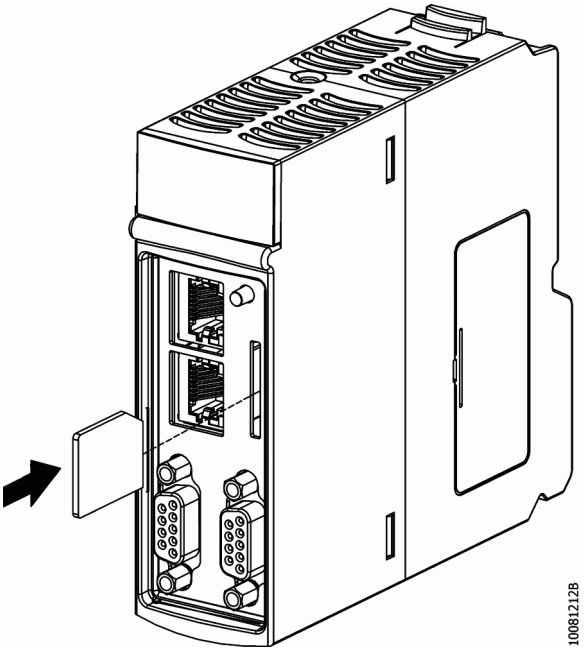


Figura 16: Inserção do Cartão de Memória na UCP

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

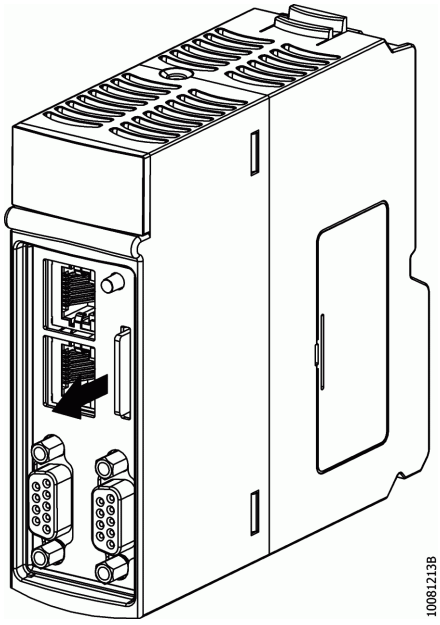


Figura 17: Retirada do Cartão de Memória

3.7. Instalação da Arquitetura

3.7.1. Instalação de Módulos no Bastidor Principal

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

3.8. Instalação do Programador

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

4. Programação Inicial

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

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

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

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

4.1. Organização e Acesso à Memória

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

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

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

Tabela 32: Exemplo Organização e Acesso à Memória

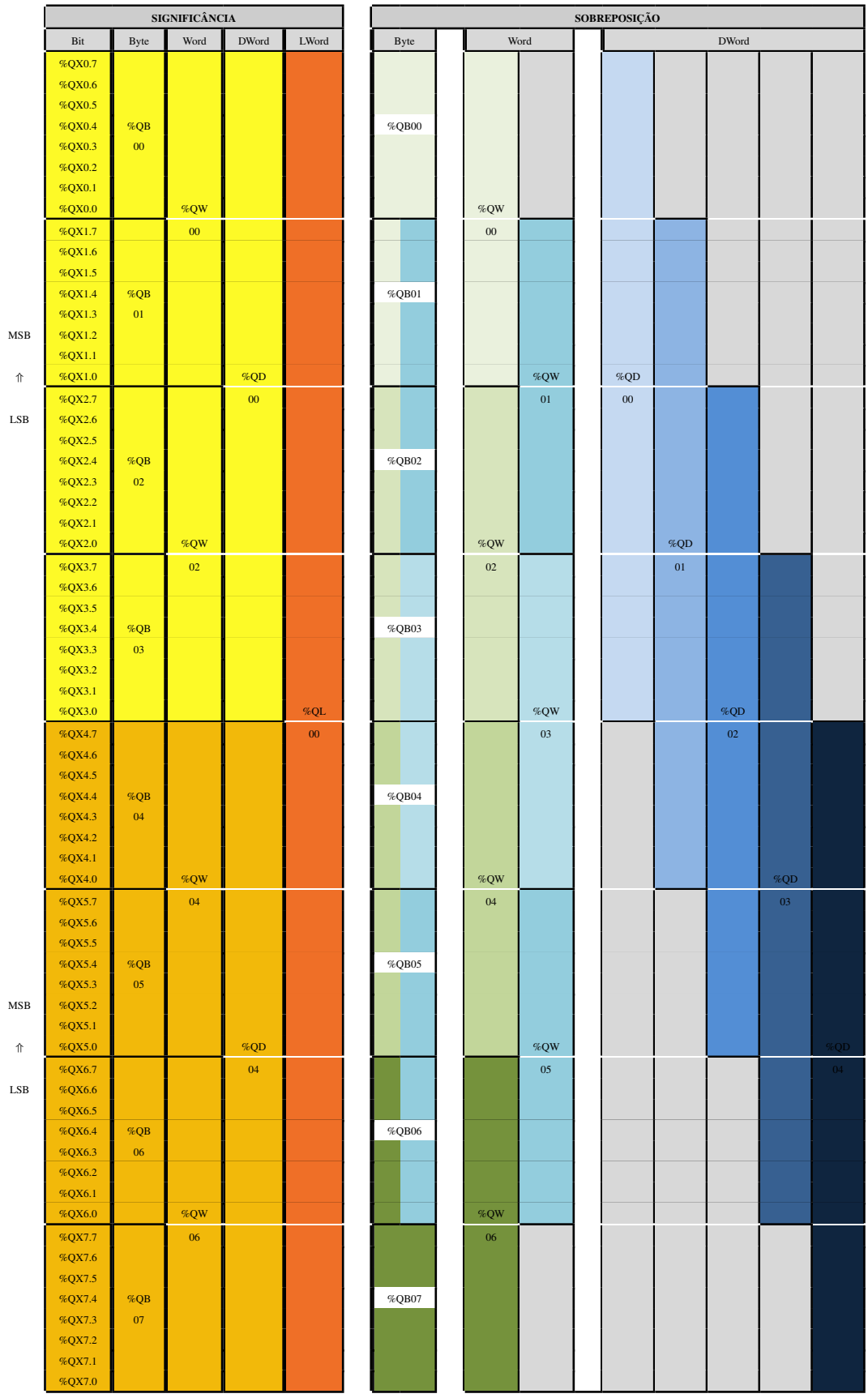


Tabela 33: Organização e Acesso à Memória

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

4.2. Perfis de Projeto

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

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

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

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

ATENÇÃO

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

4.2.1. Simples

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

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

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

Tarefa	POU	Prioridade	Tipo	Intervalo	Evento
MainTask	MainPrg	13	Cíclica	100 ms	-

Tabela 34: Tarefa no Perfil Simples

4.2.2. Básico

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

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

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

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

Tabela 35: Tarefas no Perfil Básico

4.2.3. Normal

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

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

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

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

Tarefa	POU	Prioridade	Tipo	Intervalo	Evento
MainTask	MainPrg	13	Cíclica	100 ms	-
CyclicTask00	CyclicPrg00	13	Cíclica	200 ms	-
CyclicTask01	CyclicPrg01	13	Cíclica	500 ms	-
ExternInterruptTask00	ExternInterruptPrg00	02	Externa	-	IO_EVT_0
TimeInterruptTask00	TimeInterruptPrg00	01	Cíclica	20 ms	-

Tabela 36: Tarefas no Perfil Normal

4.2.4. Experiente

O perfil de projeto Experiente inclui as mesmas tarefas básicas, MainTask, CyclicTask<nn>, ExternInterruptTask00 e TimeInterruptTask00, com as mesmas prioridades (13, 02 e 01 respectivamente), mas é uma expansão dos anteriores, pois admite múltiplas tarefas de evento. Ou seja, a aplicação pode incluir várias tarefas ExternInterruptTask<nn> ou TimeInterrupt-

Task<nn> executando os programas ExternInterruptPrg<nn> e TimeInterruptPrg<nn>. As prioridades das tarefas de evento adicionais podem ser livremente selecionadas na faixa de 08 a 12. Neste perfil, além dos programas padrões, cada tarefa pode executar programas adicionais.

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

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

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

Tarefa	POU	Prioridade	Tipo	Intervalo	Evento
MainTask	MainPrg	13	Cíclica	100 ms	-
CyclicTask00	CyclicPrg00	13	Cíclica	200 ms	-
CyclicTask01	CyclicPrg01	13	Cíclica	500 ms	-
ExternInterruptTask00	ExternInterruptPrg00	02	Externa	-	IO_EVT_0
TimeInterruptTask00	TimeInterruptPrg00	01	Cíclica	20 ms	-
ExternInterruptTask01	ExternInterruptPrg01	11	Externa	-	IO_EVT_1
TimeInterruptTask01	TimeInterruptPrg01	09	Cíclica	30 ms	-
FreeTask	FreePrg	31	Contínua	-	-

Tabela 37: Tarefas no Perfil Experiente

4.2.5. Personalizado

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

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

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

Tarefa	POU	Prioridade	Tipo	Intervalo	Evento
MainTask	MainPrg	13	Cíclica	100 ms	-
CyclicTask00	CyclicPrg00	13	Cíclica	200 ms	-
CyclicTask01	CyclicPrg01	13	Cíclica	500 ms	-
ExternInterruptTask00	ExternInterruptPrg00	02	Externa	-	IO_EVT_0
TimeInterruptTask00	TimeInterruptPrg00	01	Cíclica	20 ms	-
ExternInterruptTask01	ExternInterruptPrg01	11	Externa	-	IO_EVT_1
TimeInterruptTask01	TimeInterruptPrg01	09	Cíclica	30 ms	-
FreeTask	FreePrg	31	Contínua	-	-

Tabela 38: Tarefas no Perfil Personalizado

4.2.6. Perfil de Máquina

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

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

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

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

Tabela 39: Tarefas no Perfil de Máquina

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

4.2.7. Tabela Geral

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

Tabela 40: Tabela Geral de Perfis x Tarefas

ATENÇÃO

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

4.2.8. Número Máximo de Tarefas

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

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

	Tipo de Tarefa	NX3020					
		S	B	N	E	P	M
Configuration Task (Tarefa WHSB)	Cíclica	1	1	1	1	1	0
Tarefas de Usuário	Cíclica	1	1	23	23	23	2
	Disparada por Evento	0	0	0	0	23	0
	Disp. Evento Externo	0	1	1	22	23	0
	Contínua	0	1	0	1	23	0
	Disparada por Estado	0	0	0	0	23	0
NETs – Instâncias Cliente ou Servidor	Cíclica	8					
COM (n) – Instâncias Mestre ou Escravo	Cíclica	1					
TOTAL		24					

Tabela 41: Número Máximo de Tarefas IEC NX3020

Notas:

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

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

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

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

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

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

4.3. Configurando a UCP

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

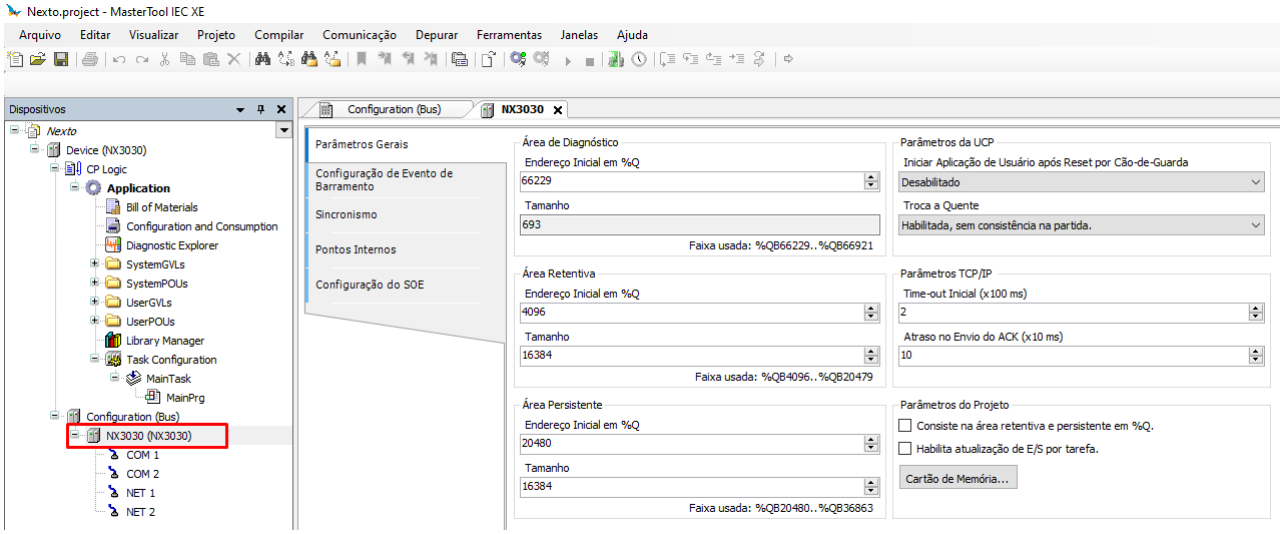


Figura 18: Configuração da UCP

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

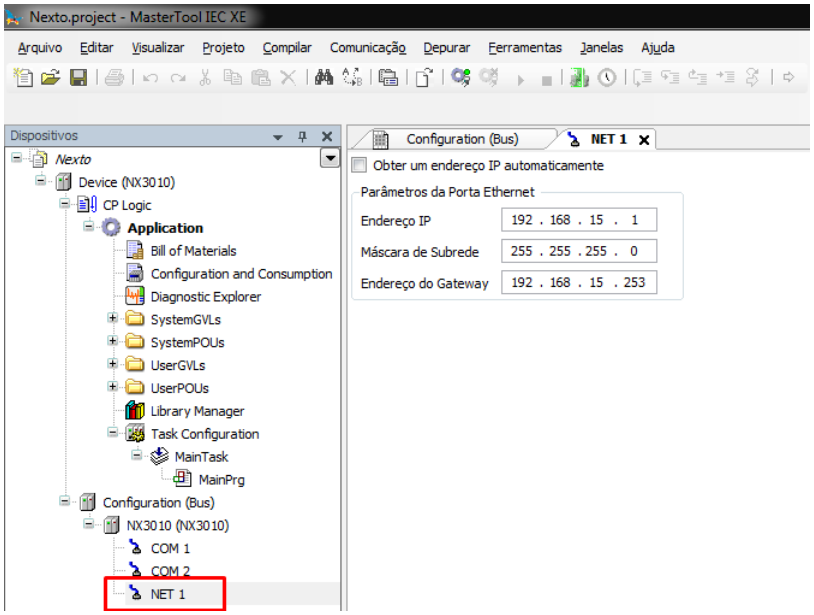


Figura 19: Configurando a Porta de Comunicação da UCP

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

4.4. Bibliotecas

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

4.5. Inserindo uma Instância de Protocolo

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

4.5.1. MODBUS Ethernet

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

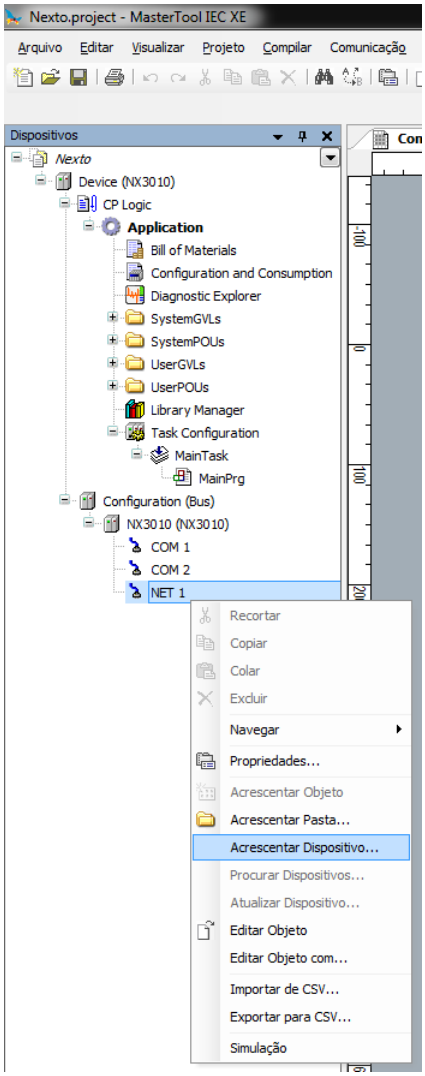


Figura 20: Adicionando a Instância

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

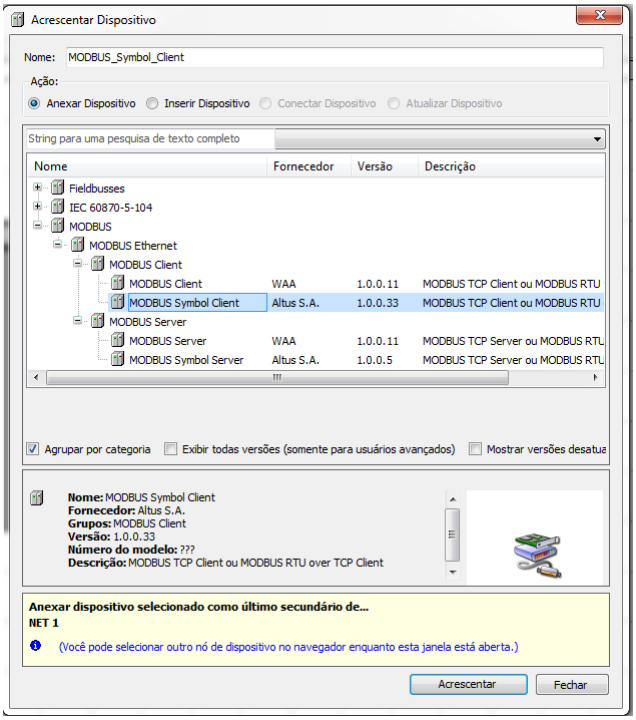


Figura 21: Selecionando o Protocolo

4.6. Localizando o Dispositivo

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

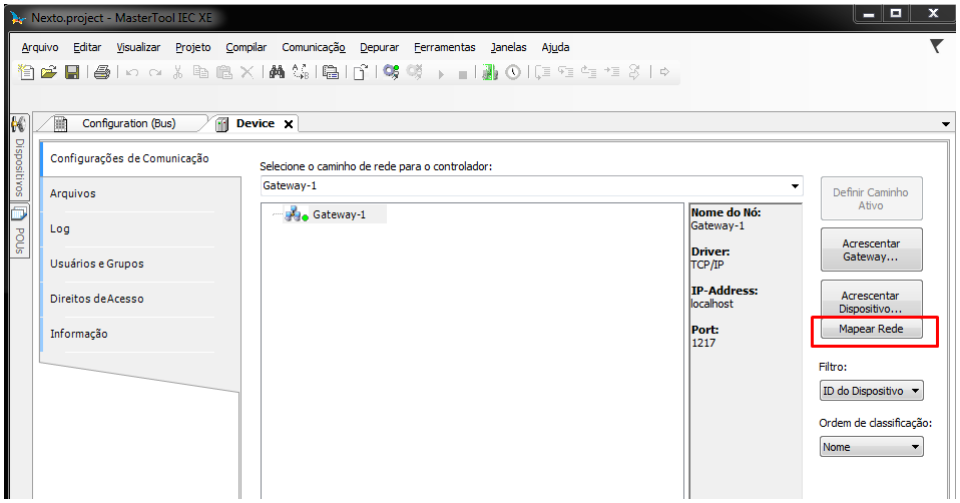


Figura 22: Localizando a UCP

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

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

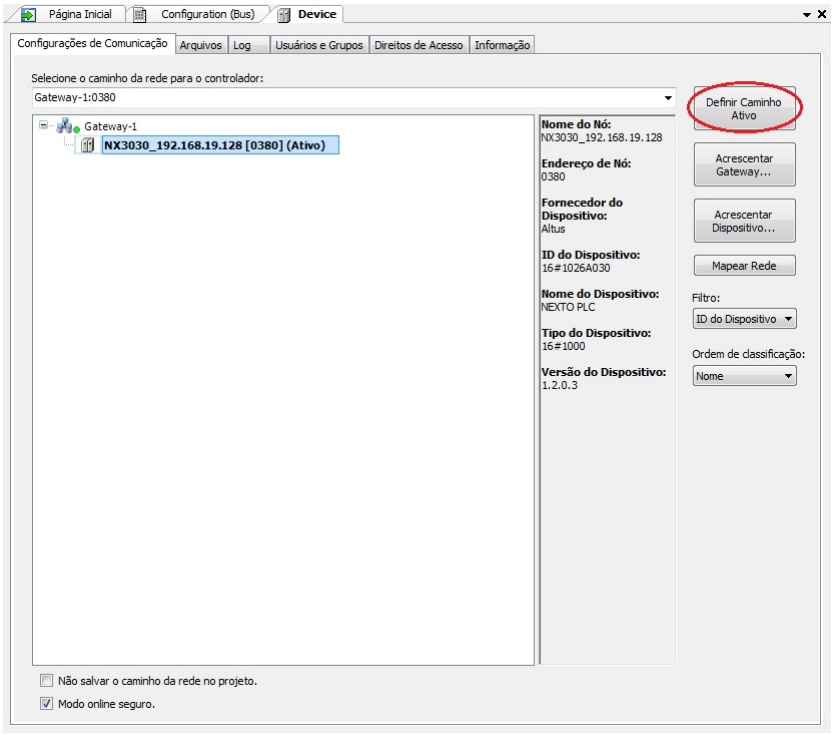


Figura 23: Selecionando a UCP

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

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

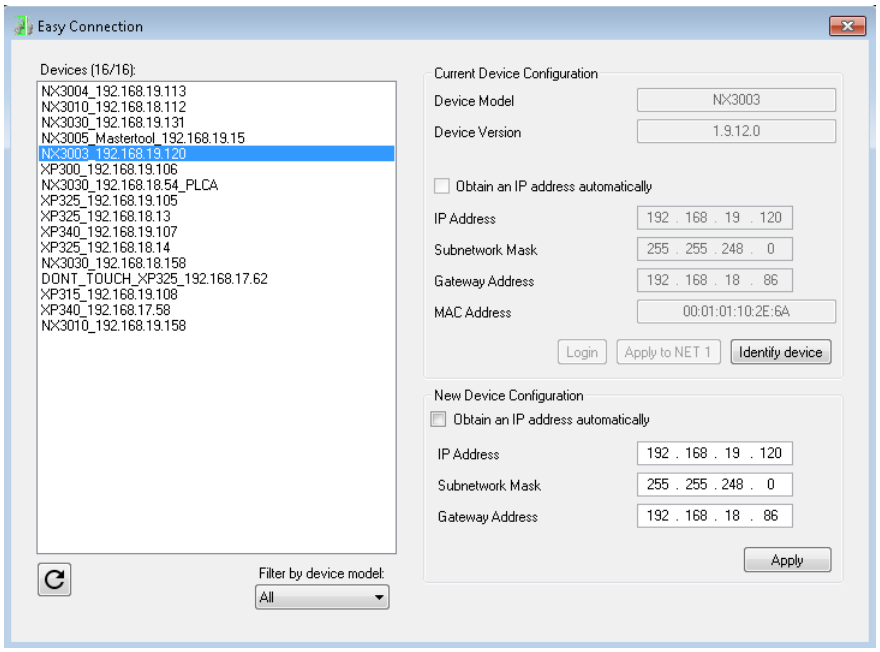


Figura 24: Easy Connection

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

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

4.7. Login

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

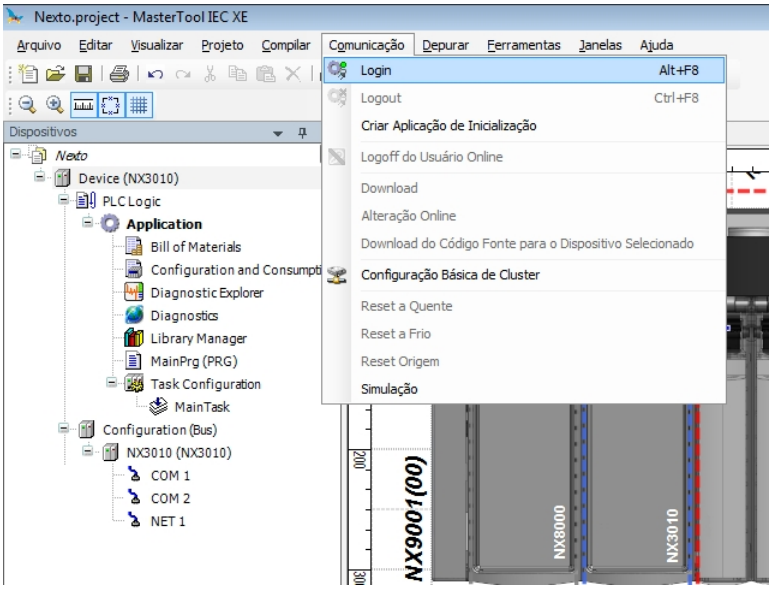


Figura 25: Enviando o Projeto para a UCP

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.

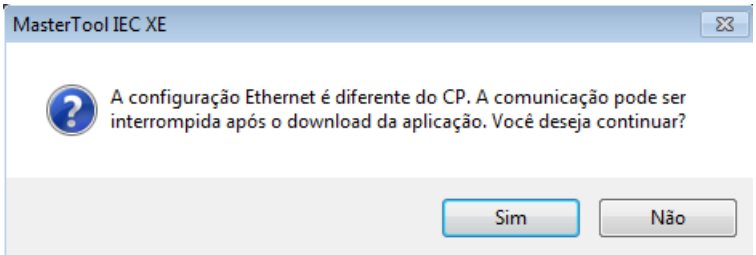


Figura 26: Aviso de configuração IP

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

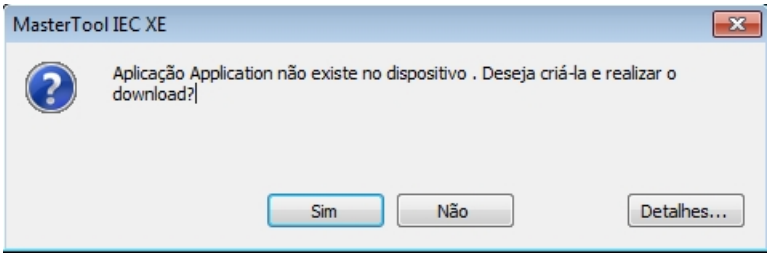


Figura 27: Sem aplicação no dispositivo

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

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

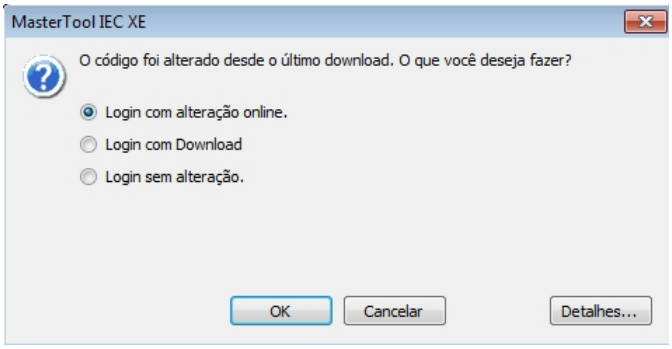


Figura 28: Atualização do Projeto na UCP

ATENÇÃO

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

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

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

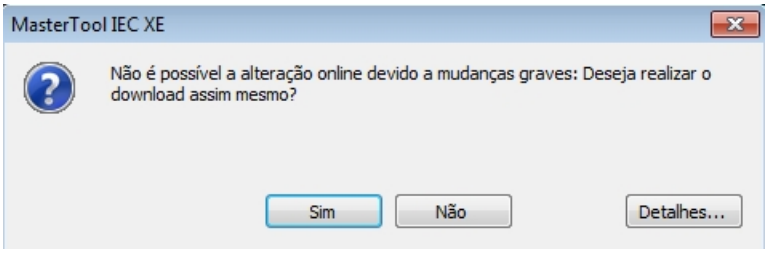


Figura 29: Alteração de Configuração

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

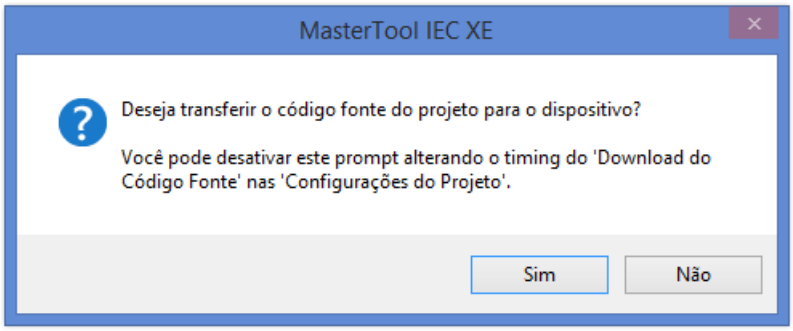


Figura 30: Download de código fonte

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

4.8. Modo Run

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

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

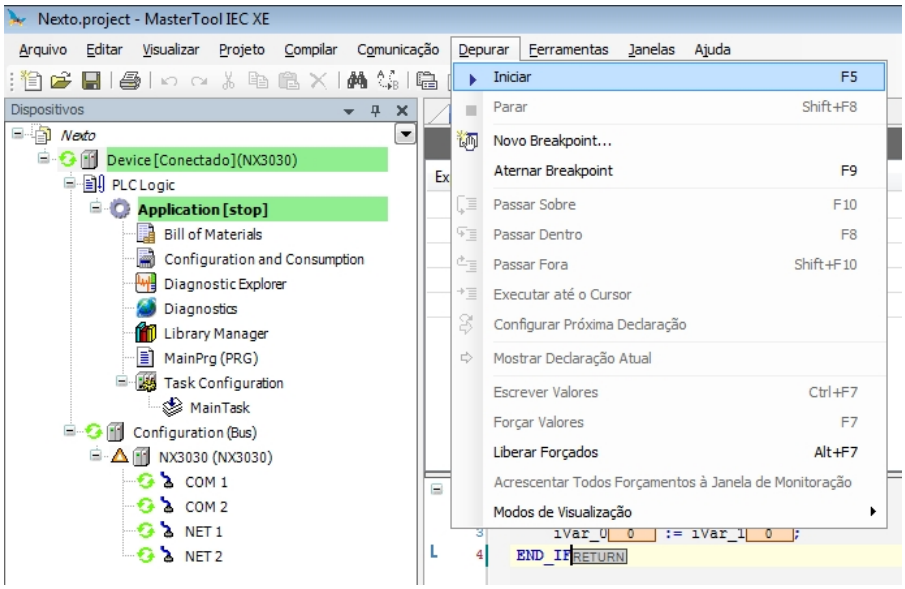


Figura 31: Iniciando a Aplicação

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.

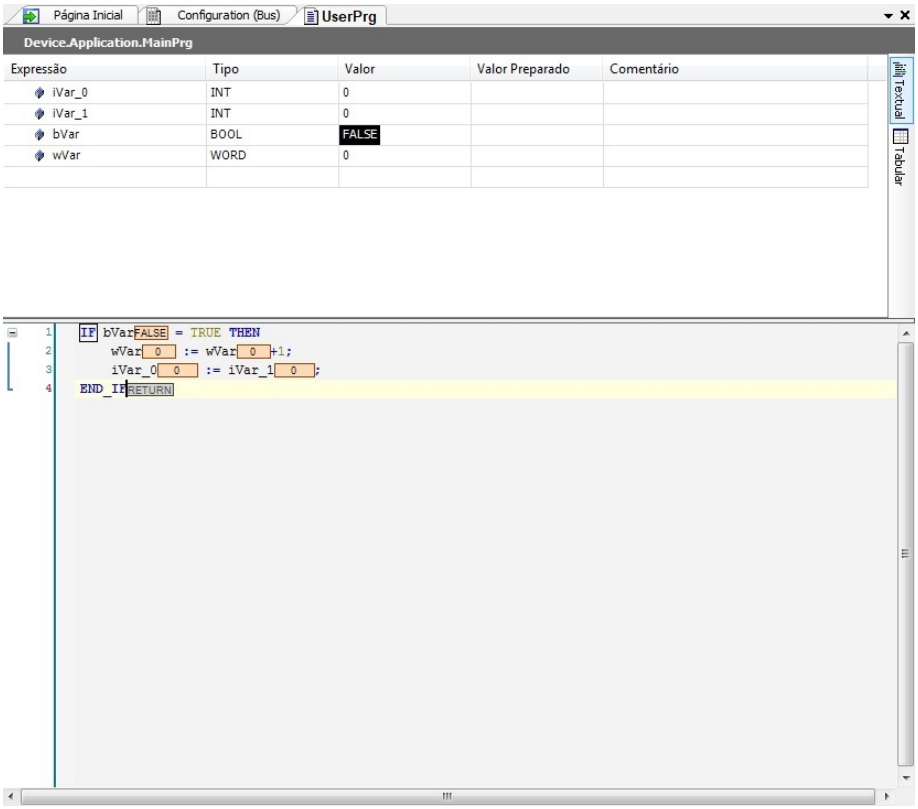


Figura 32: Programa em Execução

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

4.9. Modo Stop

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

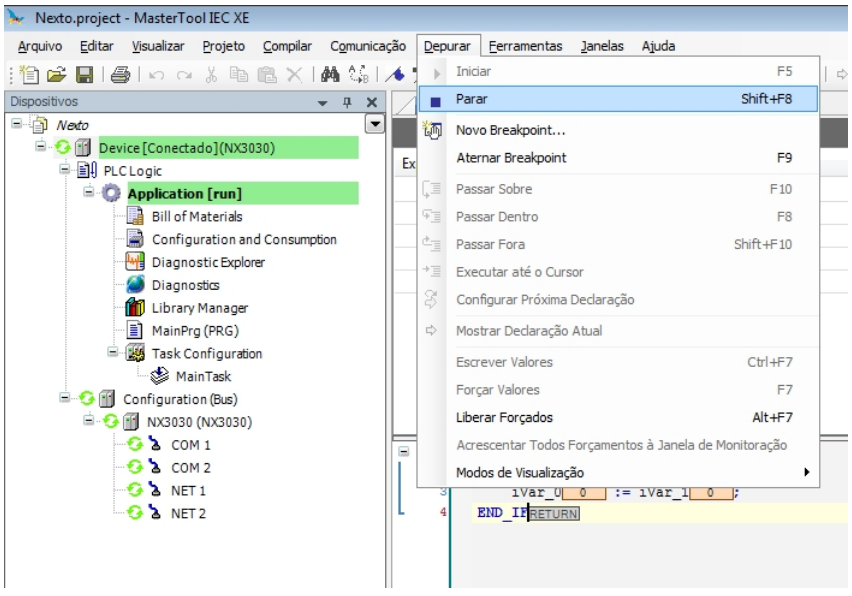


Figura 33: Parando a Aplicação

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

4.10. Escrita e Forçamento de Variáveis

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

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

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

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

ATENÇÃO

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

ATENÇÃO

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

4.11. Logout

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

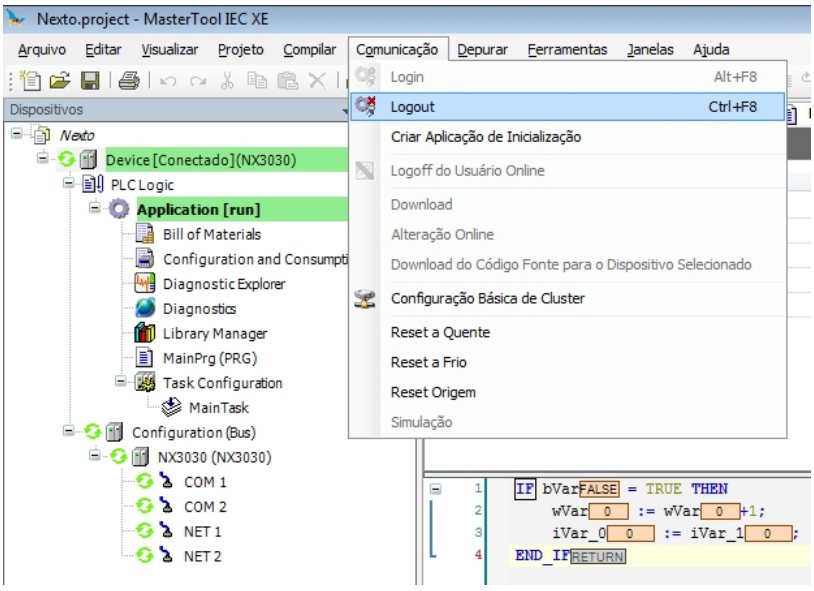


Figura 34: Encerrando a comunicação online com a UCP

4.12. Upload do Projeto

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

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

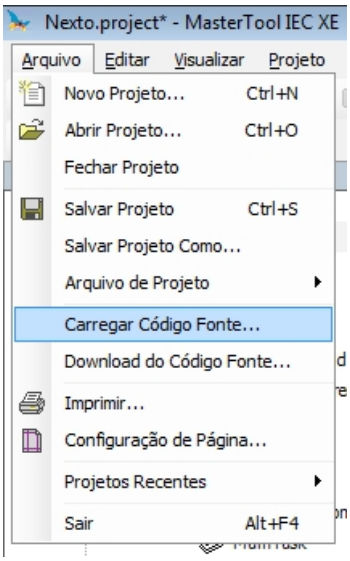


Figura 35: Opção de Upload de Projeto

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

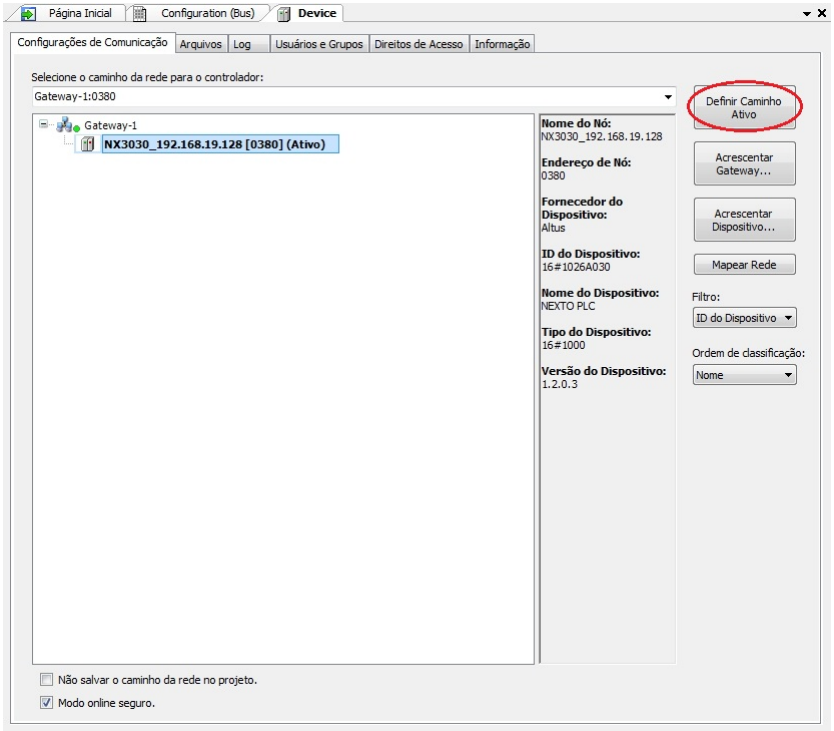


Figura 36: Selecionando a UCP

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

ATENÇÃO

O tamanho da área de memória para armazenar um projeto nas UCPs Nexto está definido na seção Memória.

ATENÇÃO

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

4.13. Estados de Operação da UCP

4.13.1. Run

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

4.13.2. Stop

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

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

4.13.3. Breakpoint

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

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

4.13.4. Exception

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

4.13.5. Reset a Quente

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

4.13.6. Reset a Frio

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

4.13.7. Reset Origem

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

Notas:

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

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

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

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

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

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

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

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

4.14.1. Programa MainPrg

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

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

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

PROGRAM MainPrg
VAR
   isFirstCycle : BOOL := TRUE;
END_VAR

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

MainPrg chama outras duas 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 System_Diagnostics

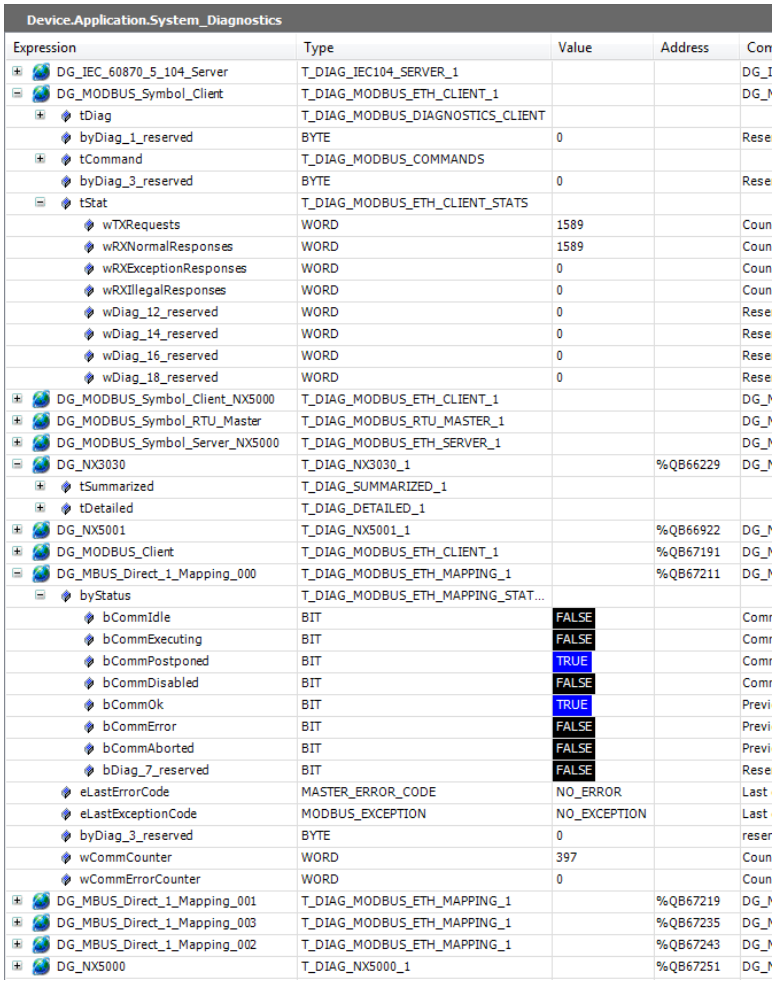
Na GVL *System_Diagnostics* estão presentes as variáveis de diagnóstico da UCP, das interfaces de comunicação (Ethernet e PROFIBUS) e de todos os drivers de comunicação. Essa GVL não é editável e as variáveis são declaradas automaticamente com tipo especificado pelo dispositivo ao qual pertence quando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

ATENÇÃO

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

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

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



Device.Application.System_Diagnostics

Expression	Type	Value	Address	Con
DG_IEC_60870_5_104_Server	T_DIAG_IEC104_SERVER_1			DG_I
DG_MODBUS_Symbol_Client	T_DIAG_MODBUS_ETH_CLIENT_1			DG_M
tDiag	T_DIAG_MODBUS_DIAGNOSTICS_CLIENT			
byDiag_1_reserved	BYTE	0		Rese
tCommand	T_DIAG_MODBUS_COMMANDS			
byDiag_3_reserved	BYTE	0		Rese
tStat	T_DIAG_MODBUS_ETH_CLIENT_STATS			
wTXRequests	WORD	1589		Coun
wRXNormalResponses	WORD	1589		Coun
wRXExceptionResponses	WORD	0		Coun
wRXIllegalResponses	WORD	0		Coun
wDiag_12_reserved	WORD	0		Rese
wDiag_14_reserved	WORD	0		Rese
wDiag_16_reserved	WORD	0		Rese
wDiag_18_reserved	WORD	0		Rese
DG_MODBUS_Symbol_Client_NX5000	T_DIAG_MODBUS_ETH_CLIENT_1			DG_M
DG_MODBUS_Symbol_RTU_Master	T_DIAG_MODBUS_RTU_MASTER_1			DG_M
DG_MODBUS_Symbol_Server_NX5000	T_DIAG_MODBUS_ETH_SERVER_1			DG_M
DG_NX3030	T_DIAG_NX3030_1		%QB66229	DG_M
tSummarized	T_DIAG_SUMMARIZED_1			
tDetailed	T_DIAG_DETAILED_1			
DG_NX5001	T_DIAG_NX5001_1		%QB66922	DG_M
DG_MODBUS_Client	T_DIAG_MODBUS_ETH_CLIENT_1		%QB67191	DG_M
DG_MBUS_Direct_1_Mapping_000	T_DIAG_MODBUS_ETH_MAPPING_1		%QB67211	DG_M
byStatus	T_DIAG_MODBUS_ETH_MAPPING_STAT...			
bCommIdle	BIT	FALSE		Comm
bCommExecuting	BIT	FALSE		Comm
bCommPostponed	BIT	TRUE		Comm
bCommDisabled	BIT	FALSE		Comm
bCommOk	BIT	TRUE		Previ
bCommError	BIT	FALSE		Previ
bCommAborted	BIT	FALSE		Previ
bDiag_7_reserved	BIT	FALSE		Rese
eLastErrorCode	MASTER_ERROR_CODE	NO_ERROR		Last
eLastExceptionCode	MODBUS_EXCEPTION	NO_EXCEPTION		Last
byDiag_3_reserved	BYTE	0		reser
wCommCounter	WORD	397		Coun
wCommErrorCounter	WORD	0		Coun
DG_MBUS_Direct_1_Mapping_001	T_DIAG_MODBUS_ETH_MAPPING_1		%QB67219	DG_M
DG_MBUS_Direct_1_Mapping_003	T_DIAG_MODBUS_ETH_MAPPING_1		%QB67235	DG_M
DG_MBUS_Direct_1_Mapping_002	T_DIAG_MODBUS_ETH_MAPPING_1		%QB67243	DG_M
DG_NX5000	T_DIAG_NX5000_1		%QB67251	DG_M

Figura 37: GVL System_Diagnostics em Modo Online

4.14.5. GVL Disables

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

Declaração de variável de desabilitação de requisição:

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

Onde:

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

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

Exemplo:

Device.Application.Disables

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

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

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

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

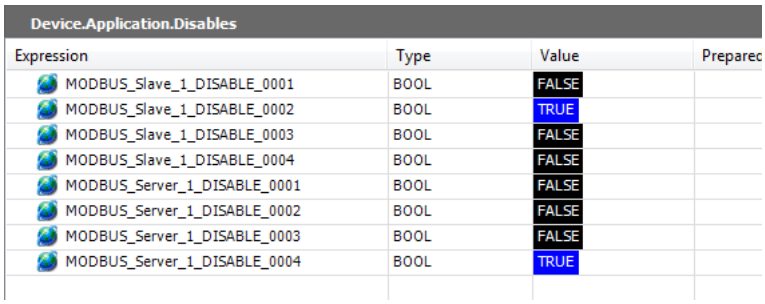


Figura 38: GVL Disables em Modo Online

4.14.6. GVL IOQualities

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

Exemplo: Device.Application.IOQualities

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

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

4.14.7. GVL Module_Diagnostics

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

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

Device.Application.Module_Diagnostics

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

Figura 39: GVL Module_Diagnostics em Modo Online

4.14.8. GVL Qualities

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

Declaração de variável de qualidade de mapeamento:

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

Onde:

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

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

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

Exemplo: Device.Application.Qualities

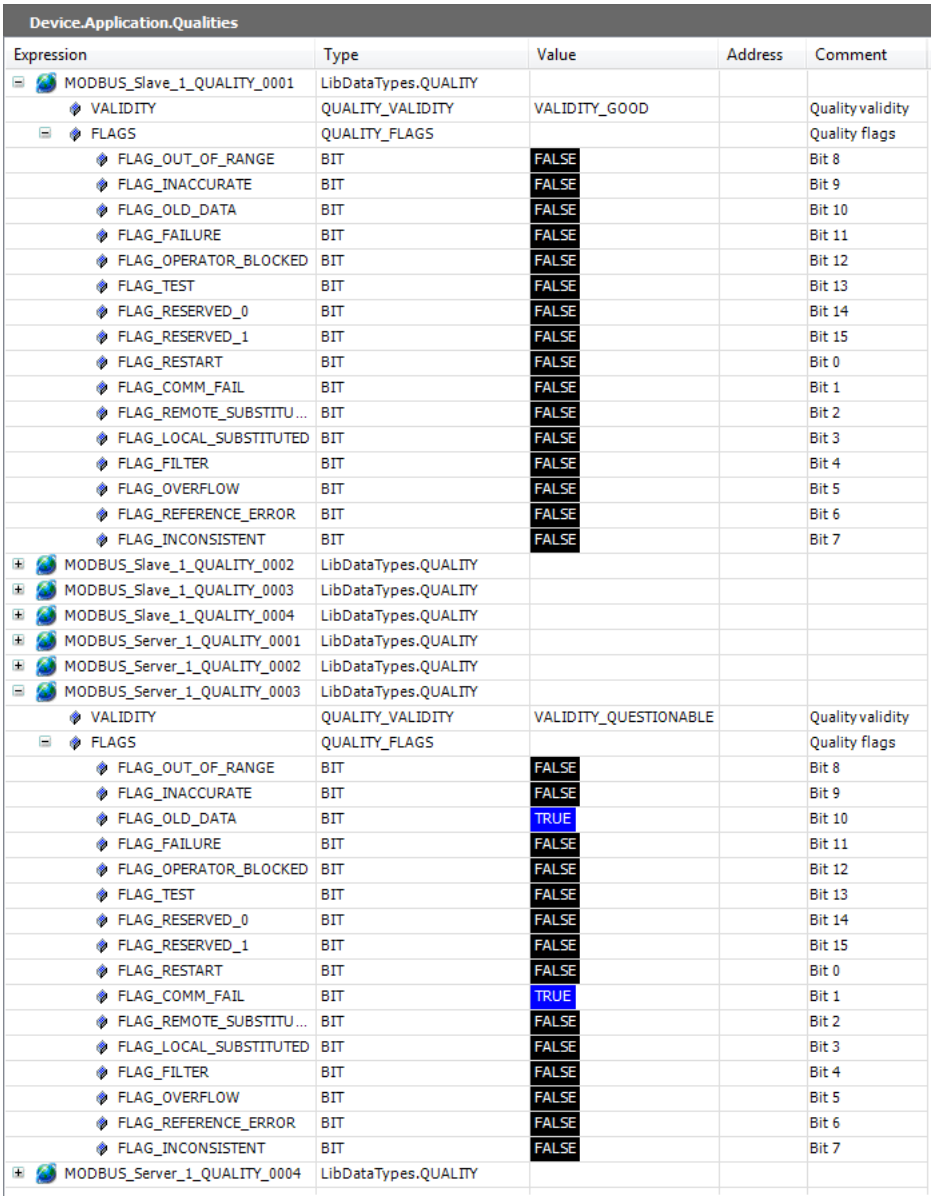
```
VAR_GLOBAL
  MODBUS_Device_QUALITY_0001: LibDataTypes.QUALITY;
  MODBUS_Device_QUALITY_0002: LibDataTypes.QUALITY;
  MODBUS_Device_QUALITY_0003: LibDataTypes.QUALITY;
END_VAR
```

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

Para os dispositivos de comunicação MODBUS, as variáveis de qualidade se comportam da maneira indicada na Tabela 55.

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

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



Device.Application.Qualities

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

Figura 40: GVL Qualities em Modo Online

4.14.9. GVL ReqDiagnostics

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

Declaração de variável de diagnóstico de requisição:

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

Onde:

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

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

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

ATENÇÃO

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

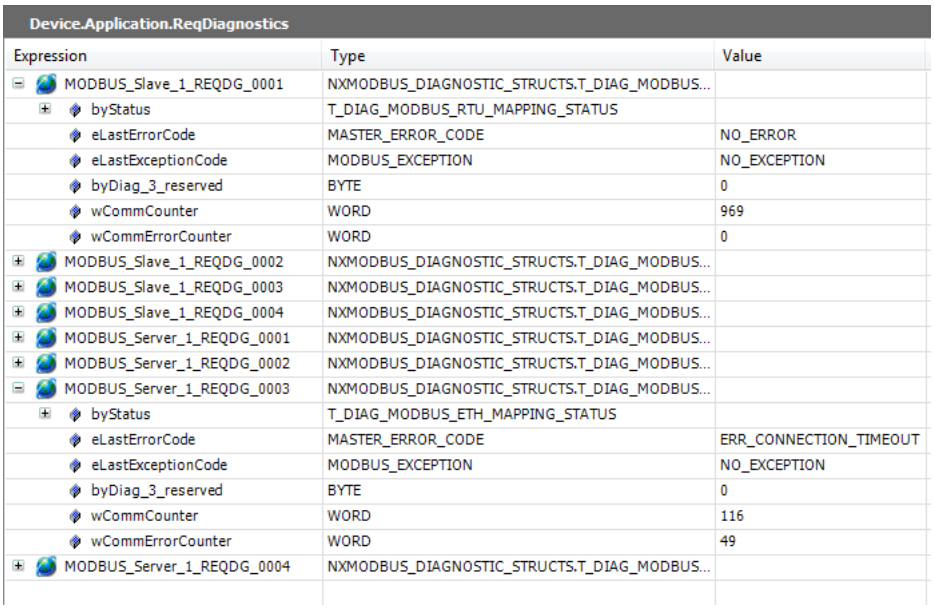
Exemplo:

Device.Application.ReqDiagnostics

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

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

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



Device.Application.ReqDiagnostics

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

Figura 41: GVL ReqDiagnostics em Modo Online

4.14.10. Função Prepare_Start

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

4.14.11. Função Prepare_Stop

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

4.14.12. Função Start_Done

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

4.14.13. Função Stop_Done

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

5. Configuração

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

5.1. Device

5.1.1. Gerenciamento de Usuários e Direitos de Acesso

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

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

5.1.2. Configurações do CP

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

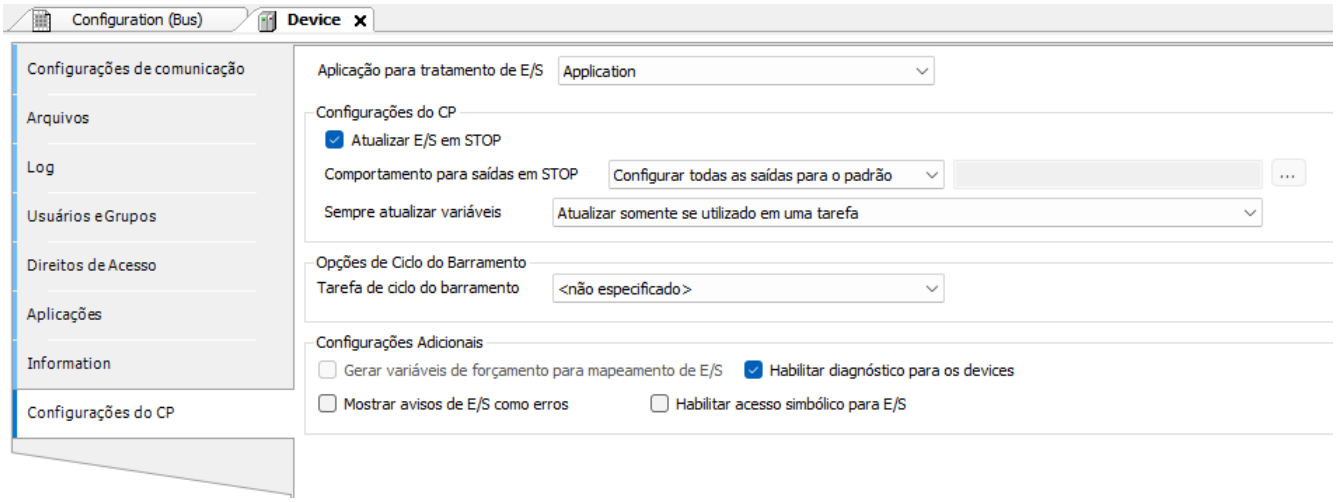


Figura 42: Configurações do CP

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

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

Tabela 42: Configurações do CP

ATENÇÃO

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

5.2. Configuração da UCP

5.2.1. Parâmetros Gerais

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

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

Configuração	Descrição	Padrão	Opções
Endereço Inicial em %Q	**Área de Diagnósticos (%Q)**		
	Endereço inicial dos diagnósticos da UCP (%Q)	Alocado automaticamente na criação do projeto.	0 a 64843
Tamanho	Tamanho da área de diagnósticos em bytes	693	Não é possível alterar o tamanho da área de diagnósticos da UCP
Endereço Inicial em %Q	**Área Retentiva (%Q)**		
	Endereço inicial da memória de dados retentivos (%Q)	4096	0 a 65535
Tamanho	Tamanho da memória de dados retentivos em bytes	65536	0 a 65536
Endereço Inicial em %Q	**Área Persistente (%Q)**		
	Endereço inicial da memória de dados persistentes (%Q)	20480	0 a 65534
Tamanho	Tamanho da memória de dados persistentes em bytes	65536	0 a 65536
Iniciar Aplicação de Usuário após Reset por Cão-de-Guarda	**Parâmetros da UCP**		
	Quando habilitado, inicia a aplicação do usuário após o reset do cão-de-guarda de hardware ou pela reinicialização do Runtime, porém mantêm a indicação do diagnóstico via LED WD e via variáveis.	Desabilitado	Habilitado Desabilitado

Configuração	Descrição	Padrão	Opções
Troca a Quente	Modo troca a quente dos módulos	Habilitada, sem consistência na partida. (pode variar de acordo com o modelo de UCP)	- Desabilitada, apenas para módulos declarados - Desabilitada (com consistência na partida) - Desabilitada, sem consistência na partida - Habilitada, com consistência na partida apenas para módulos declarados - Habilitada, com consistência na partida - Habilitada, sem consistência na partida
	Parâmetros do Projeto		
Habilita atualização de E/S por tarefa	Configuração para atualizar as entradas e saídas nas tarefas em que estas são utilizadas.	Desmarcado	- Marcado: As entradas e saídas são atualizadas pelas tarefas em que forem utilizadas. - Desmarcado: As entradas e saídas são atualizadas apenas pela MainTask
Habilita variáveis retentivas e persistentes em Blocos Funcionais	Configuração que permite o uso de variáveis retentivas e persistentes em Blocos Funcionais	Desmarcado	- Marcado: permite o uso de variáveis retentivas e persistentes em Blocos Funcionais. - Desmarcado: pode ocorrer erro de exceção na partida.

Tabela 43: Configurações da UCP

Notas:

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

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

ATENÇÃO

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

ATENÇÃO

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

ATENÇÃO

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

ATENÇÃO

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

5.2.1.1. Troca a Quente

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

CUIDADO

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

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

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

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

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

Tabela 44: Situações Anormais de Barramento

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

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

5.2.1.1.1. Troca a Quente Desabilitada, Apenas para Módulos Declarados

Nesta configuração, quando ocorre uma situação anormal de barramento (conforme a Tabela 44) a UCP entra em Modo *Stop* em um tempo de até 2 segundos, quando o LED DG começa a piscar 4x (conforme a Tabela 45). Para que a UCP volte

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

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

5.2.1.1.2. Troca a Quente Desabilitada

Esta configuração não permite qualquer situação anormal de barramento (conforme a Tabela 44) inclusive módulos não declarados e presentes no barramento. A UCP entra em Modo *Stop* , sendo que o LED DG começa a piscar 4x (conforme a Tabela 45). Para que, nesses casos, a UCP volte ao estado normal *Run*, além de desfazer o que causou a situação anormal é necessário executar um *Reset a Quente* ou um *Reset a Frio*. Caso seja realizado um *Reset Origem*, será necessário realizar o download para que a UCP possa voltar ao estado normal *Run*. Os comandos de *Reset a Quente*, *Reset a Frio* e *Reset Origem* podem ser feitos pelo MasterTool IEC XE no menu *Comunicação*.

5.2.1.1.3. Troca a Quente Desabilitada, sem Consistência na Partida

Permite que o sistema entre em operação mesmo quando algum módulo estiver em uma situação anormal de barramento (conforme Tabela 44). As situações anormais são relatadas via diagnóstico.

Qualquer modificação no barramento vai fazer a UCP entrar em Modo *Stop*, sendo que o LED DG começa a piscar 2x (conforme a Tabela 45). Para que, nesses casos, a UCP volte ao estado normal *Run*, é necessário executar um *Reset a Quente* ou um *Reset a Frio*. Caso seja realizado um *Reset Origem*, será necessário realizar o download para que a UCP possa voltar ao estado normal *Run*. Os comandos de *Reset a Quente*, *Reset a Frio* e *Reset Origem* podem ser feitos pelo MasterTool IEC XE no menu *Comunicação*.

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

É considerada "partida" o intervalo entre a energização da UCP (ou comando de reset ou download de aplicação) até a primeira vez em que a mesma entra em modo *Run*. Esta configuração verifica se ocorreu alguma situação anormal de barramento (conforme a Tabela 44) durante a partida; em caso positivo, a UCP entra em Modo *Stop* e o LED DG começa a piscar 4x (conforme a Tabela 45). Posteriormente, para que a UCP possa ser colocada em modo *Run*, além de corrigir o que ocasionou a situação anormal, é necessário executar um comando de *Reset a Quente* ou um *Reset a Frio*. Caso seja realizado um *Reset Origem*, será necessário realizar o download para que a UCP possa voltar ao estado normal *Run*. Os comandos de *Reset a Quente*, *Reset a Frio* e *Reset Origem* podem ser feitos pelo MasterTool IEC XE no menu *Comunicação*.

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

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

ATENÇÃO

Nesta configuração, quando ocorrer falta de alimentação (mesmo que temporária), comando *Reset a Quente*, comando *Reset a Frio* ou ter sido realizado o *Download* de uma nova aplicação, e algum módulo estiver em uma situação anormal de barramento; a UCP entrará em Modo *Stop* e o LED DG começa a piscar 4x (conforme a Tabela 45), pois estas são consideradas situações de partida.
Esta é a opção mais recomendada, pois garante a integridade do sistema na sua inicialização e permite a troca de módulos com o sistema funcionando.

5.2.1.1.5. Troca a Quente Habilitada, com Consistência na Partida

Esta configuração verifica se ocorreu alguma situação anormal de barramento (conforme a Tabela 44) durante a partida, inclusive se há módulos não declarados e presentes no barramento; em caso positivo, a UCP entra em Modo *Stop* e o LED DG começa a piscar 4x (conforme a Tabela 45). Posteriormente, para que a UCP possa ser colocada em modo *Run*, além de corrigir o que ocasionou a situação anormal, é necessário executar um comando de *Reset a Quente* ou um *Reset a Frio*. Caso seja realizado um *Reset Origem*, será necessário realizar o download para que a UCP possa voltar ao estado normal *Run*. Os comandos de *Reset a Quente*, *Reset a Frio* e *Reset Origem* podem ser feitos pelo MasterTool IEC XE no menu *Comunicação*.

5.2.1.1.6. *Troca a Quente Habilitada, sem Consistência na Partida*

Permite que o sistema entre em operação mesmo quando algum módulo estiver em uma situação anormal de barramento (conforme Tabela 44). As situações anormais são relatadas via diagnóstico, tanto durante, como após a partida.

ATENÇÃO

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

5.2.1.1.7. *Como realizar a Troca a Quente*

CUIDADO

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

ATENÇÃO

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

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

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

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

É importante salientar que, nos casos em que a UCP entra em Modo *Stop* e o LED DG começa a piscar 4x, conforme a Tabela 45, devido a alguma situação anormal de barramento, conforme a Tabela 44; os módulos de saída têm o comportamento dos seus pontos de acordo com o que foi configurado nos *Parâmetros do Módulo* quando a UCP sai do Modo *Run* e entra em Modo *Stop*. Em caso de inicialização da aplicação, quando a UCP entra em Modo *Stop* sem ter passado para o Modo *Run*, os módulos de saída têm o comportamento de seus pontos em modo seguro de falha, ou seja, o ponto permanece desligado (0 Vdc).

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

ATENÇÃO

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

Abaixo, a Tabela 45 relaciona as condições de barramento e o estado de operação do LED DG da UCP Nexto. Para maiores informações sobre os estados dos LEDs de diagnóstico, consultar a seção Diagnósticos via LED.

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

Tabela 45: Relação entre Condições e Troca a Quente

Nota:

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

5.2.1.2. Áreas de Memória Retentiva e Persistente

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

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

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

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

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

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

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

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

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

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

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

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

```
VAR_GLOBAL PERSISTENT RETAIN
  wVariavelSimbolicaPersistenteGlobal_01 : WORD;
END_VAR

VAR_GLOBAL RETAIN
  wVariavelSimbolicaRetentivaGlobal_01 : WORD;
END_VAR
```

ATENÇÃO

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

5.2.1.3. Parâmetros do Projeto

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

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

Tabela 46: Parâmetros do Projeto da UCP

ATENÇÃO

Após configurar as possibilidades de cópia do projeto e de ter criado a aplicação de inicialização, deve-se localizar o arquivo "*Application.crc*" para que as configurações referentes ao cartão de memória tenham efeito. A localização pode ser realizada em *Selecionar a Application.crc* através do botão *Localizar Arquivo...*, como pode ser visto na figura 131.

5.2.2. Configuração de Evento Externo

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

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

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

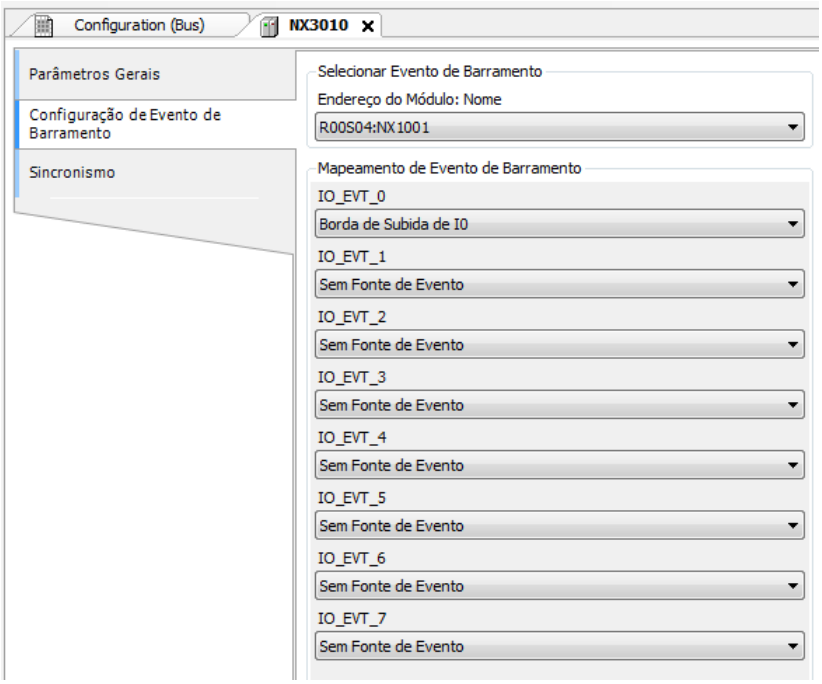


Figura 43: Tela de Configuração de Evento Externo na UCP

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

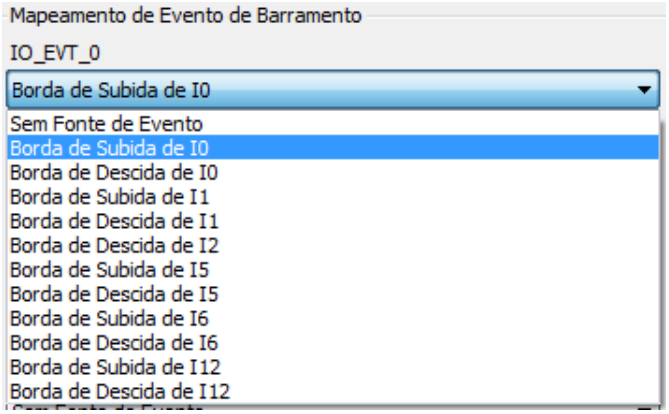


Figura 44: Opções de Origem do Evento Externo no Módulo NX1001

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

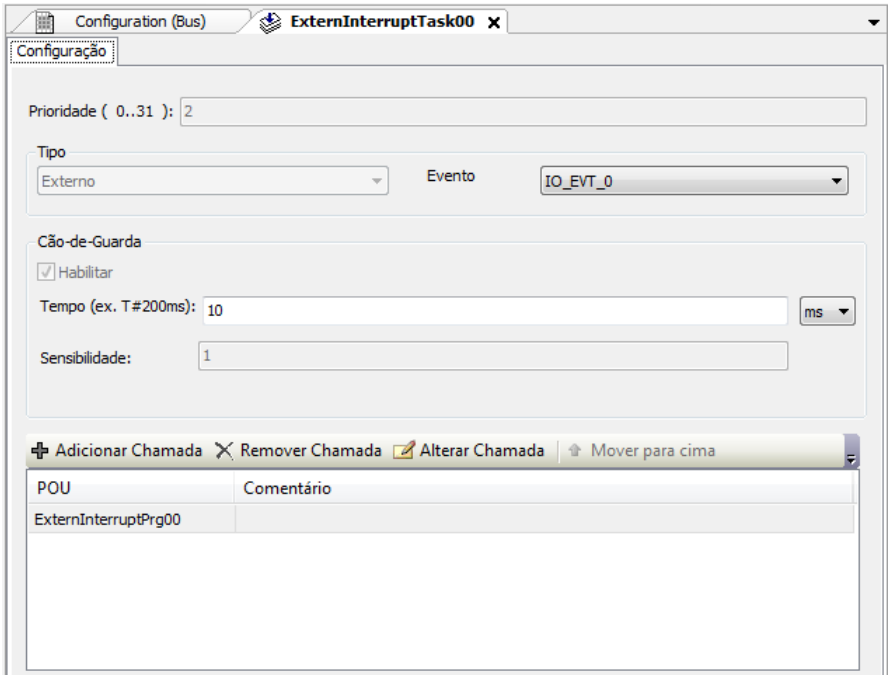


Figura 45: Tela de Configuração da Tarefa ExternInterruptTask00

5.2.3. Configuração do SOE

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

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

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

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

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

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

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

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

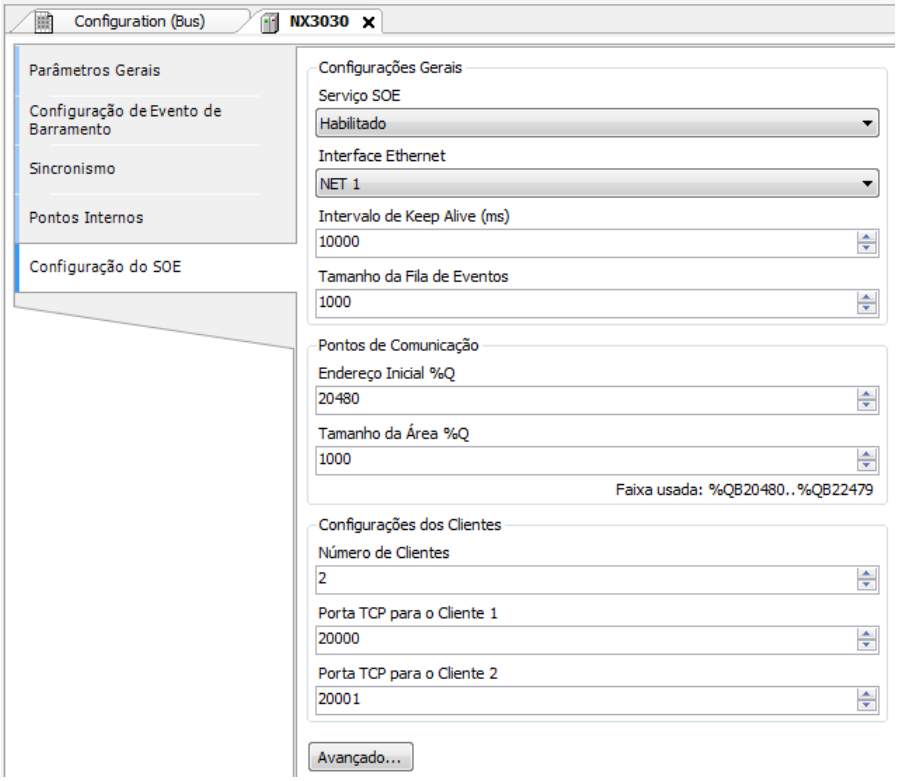


Figura 46: Configuração da Sequência de Eventos

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

Tabela 47: Configuração do SOE

Notas:

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

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

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

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

Tabela 48: Configurações Avançadas do SOE

Nota:

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

ATENÇÃO

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

5.2.4. Sincronização de Tempo

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

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

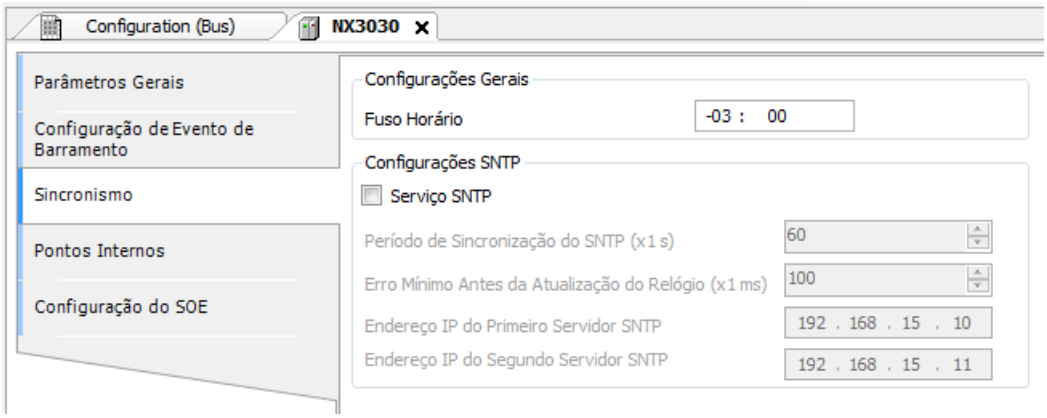


Figura 47: Configuração de SNTP

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

Tabela 49: Configurações SNTP

Notas:

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

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

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

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

5.2.4.1. IEC 60870-5-104

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

ATENÇÃO

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

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

5.2.4.2. SNTP

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

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

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

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

1. Tentativa de sincronização através do primeiro servidor. Caso a sincronização ocorra com sucesso, a UCP aguarda o tempo para a nova sincronização (*Período de Sincronização*) e tentará sincronizar-se novamente com este servidor, utilizando, então, este como servidor primário. Em caso de falha (o servidor não responde em menos de 5 s) o passo 2 é executado.
2. Tentativa de sincronização através do segundo servidor. Caso a sincronização ocorra com sucesso, a UCP aguarda o tempo para a nova sincronização (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 a UCP estiver nos modos *STOP* ou *BREAKPOINT*, desde que exista uma aplicação na UCP com o cliente SNTP habilitado e corretamente configurado.

5.2.4.3. Horário de Verão

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

Para maiores informações, consultar a seção Relógio RTC.

5.2.5. Pontos Internos

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

ATENÇÃO

Diferente do que acontece com os módulos de E/S declarados no barramento local, que têm suas variáveis de qualidade criadas pelo MasterTool (GVL IOQualities) e automaticamente atualizadas pela UCP, os módulos de E/S declarados em remotas PROFIBUS não as têm.

É de responsabilidade do usuário a declaração das variáveis de qualidade dos pontos PROFIBUS, a associação destas variáveis de qualidade com as variáveis de valor na aba de Pontos Internos, bem como a geração e atualização do valor das variáveis de qualidade a partir dos diagnósticos PROFIBUS existentes: módulos de E/S PROFIBUS, cabeça PROFIBUS e mestre PROFIBUS.

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

ATENÇÃO

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

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

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

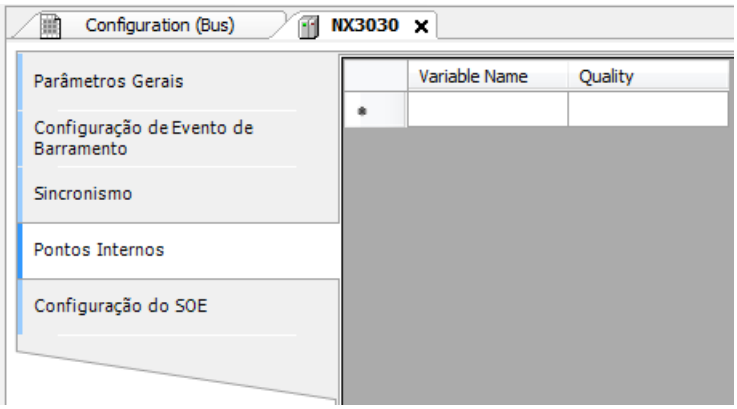


Figura 48: Tela de Configuração dos Pontos Internos

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

Tabela 50: Configuração dos Pontos Internos

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

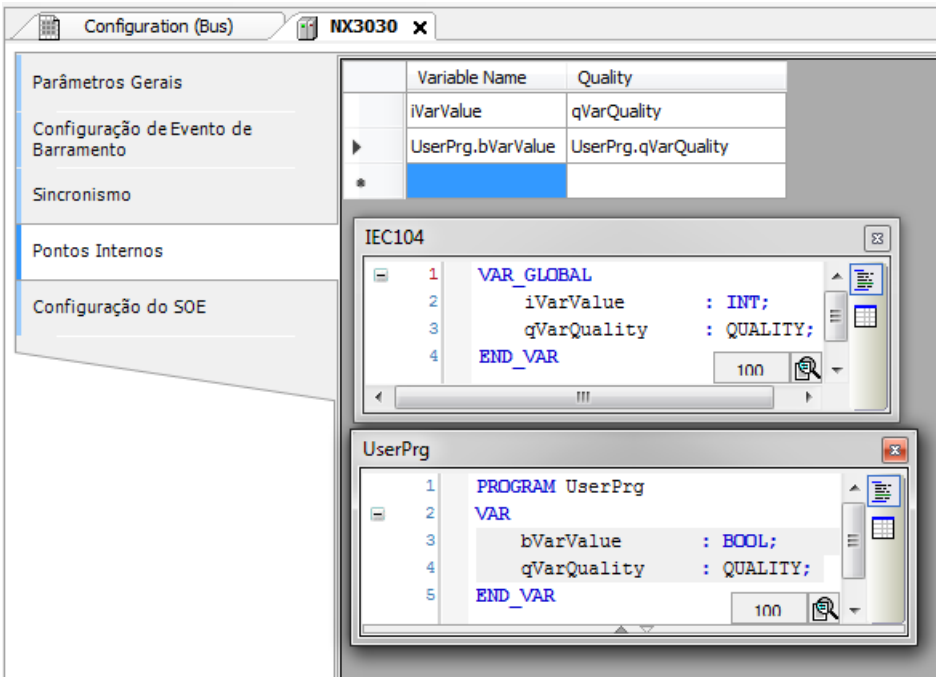


Figura 49: Exemplo de Configuração dos Pontos Internos

5.2.5.1. Conversões de Qualidade

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

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

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

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

ATENÇÃO

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

5.2.5.1.1. Qualidade Interna

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

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

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

Tabela 51: Estrutura QUALITY

5.2.5.1.2. Conversão IEC 60870-5-104

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

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

Tabela 52: Conversão Interno para IEC 60870-5-104 de Pontos Digitais

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

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

Tabela 53: Conversão Interno para IEC 60870-5-104 de Pontos Analógicos, Step Position e Bitstring

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

Tabela 54: Conversão Interno para IEC 60870-5-104 de Contadores

5.2.5.1.3. Qualidade Interna MODBUS

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

Qualidade Resultante	VALIDITY Resultante	Descrição
FLAG_RESTART	VALIDITY_INVALID	Valor de inicialização. O ponto nunca foi atualizado.
-	VALIDITY_GOOD	Comunicação OK. O ponto está atualizado.
FLAG_COMM_FAIL E FLAG_RESTART	VALIDITY_INVALID	Erro de comunicação. O ponto nunca foi atualizado.
FLAG_COMM_FAIL E FLAG_OLD_DATA	VALIDITY_QUESTIONABLE	Um erro ocorreu mas o ponto foi atualizado e agora possui um valor antigo.

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

Tabela 55: Qualidade MODBUS

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

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

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

Para mais informações consulte a seção GVL IOQualities.

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

Tabela 56: Qualidade dos Módulos de E/S

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

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

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

ATENÇÃO

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

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

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

5.2.5.1.6. *Qualidade da Entrada Digital PROFIBUS*

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

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

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

5.2.5.1.7. *Qualidade da Saída Digital PROFIBUS*

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

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

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

5.2.5.1.8. *Qualidade da Entrada Analógica PROFIBUS*

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

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

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

5.2.5.1.9. *Qualidade da Saída Analógica PROFIBUS*

```
// Atualização da qualidade de uma saída analógica PROFIBUS, módulo NX6100

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

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

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

5.3. Configuração das Interfaces Seriais

5.3.1. COM 1

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

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

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

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

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

Tabela 57: Configurações da Interface Serial Padrão RS-232C

Notas:

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

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

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

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

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

Tabela 58: Configurações Específicas

5.3.1.1. Configurações Avançadas

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

Configuração	Descrição	Padrão	Opções
Parâmetros de Porta Avançados			
Sinais de modem	Realiza o controle de requisição para transmissão de um comando, através da interface RS-232C.	RTS Sempre Desligado	- RTS: Habilitado no início da transmissão e reiniciado, o mais rápido possível, após o final da transmissão. Ex: Controle de um conversor RS-232C/RS-485 externo. - RTS Sempre Desligado: Sempre desabilitado. - RTS Sempre Ligado: Sempre habilitado - RTS/CTS: Caso o CTS esteja desabilitado, o RTS é habilitado. Então, aguarda-se o CTS ser habilitado para a transmissão começar e o RTS é reiniciado, o mais rápido possível, no final da transmissão. Ex: Controle de rádio modems com o mesmo sinal de modem. - RTS Manual: o usuário é responsável por controlar todos os sinais de controle
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

Configuração	Descrição	Padrão	Opções
Eventos Seriais			
RX em TX	Quando verdadeiro, todos os bytes recebidos durante a transmissão serão descartados em vez de ir para a fila de RX. Utilizado para desabilitar a operação full-duplex na interface RS-232C.	Desabilitado	- Habilitado: Configuração habilitada - Desabilitado: Configuração desabilitada
Evento RX DCD	Quando verdadeiro, gera um evento externo devido à mudança do sinal DCD	Habilitado	- Habilitado: Configuração habilitada - Desabilitado: Configuração desabilitada
Evento RX CTS	Quando verdadeiro, gera um evento externo devido à mudança do sinal CTS.	Habilitado	- Habilitado: Configuração habilitada - Desabilitado: Configuração desabilitada

Tabela 59: Configurações Avançadas da Interface Serial Padrão RS-232C

Notas:

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

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

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

5.3.2. COM 2

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

A tabela abaixo mostra as configurações permitidas para esta interface.

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

Tabela 60: Configurações Específicas

5.3.2.1. Configurações Avançadas

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

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

Tabela 61: Configurações Avançadas da Interface Serial Padrões RS-422C e RS-485

5.4. Configuração das Interfaces Ethernet

A UCP NX3020 possui NET 1 e NET 2. Além das interfaces Ethernet integradas, a Série Nexto disponibiliza também interfaces Ethernet remotas através da inclusão do módulo NX5000. Os módulos NX5000 possuem apenas a interface NET 1.

5.4.1. Interfaces Ethernet Integradas

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

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

5.4.1.1. NET 1

Configuração	Descrição	Padrão	Opções
Obter um endereço IP automaticamente	Habilita a funcionalidade do Cliente DHCP no dispositivo para atribuição automática de IP	Desmarcado	Marcado ou Desmarcado
Endereço IP	Endereço IP 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 62: Configurações NET 1

5.4.1.2. NET 2

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

Tabela 63: Configurações NET 2

ATENÇÃO

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

5.4.1.3. Modos de Operação das Interfaces Ethernet Integradas

Os canais Ethernet da UCP podem ser configurados em dois modos distintos de funcionamento: Modo Simples e Modo Redundante. Quando a NET 1 é configurada como Modo Redundante, as interfaces NET 1 e NET 2 passam a operar como um par redundante. Nesta arquitetura, a NET 2 tem todos os seus campos de configuração desabilitados automaticamente.

5.4.1.3.1. Modo Simples

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

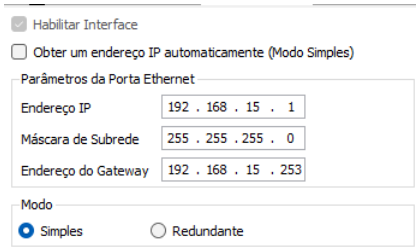


Figura 50: Configurações Avançadas das Interfaces Ethernet - Modo Simples

5.4.1.3.2. Modo Redundante

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.

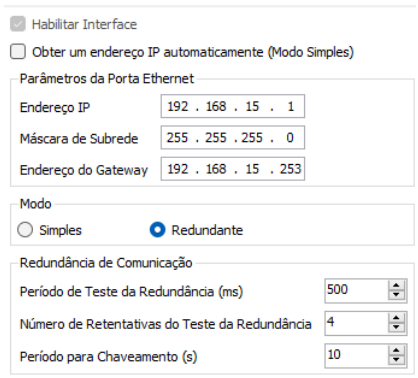


Figura 51: Configurações Avançadas das Interfaces Ethernet - Modo NIC Teaming

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.

5.4.2. Interface Ethernet Remota NX5000

5.4.2.1. NET 1

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

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

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

Tabela 64: Configurações NET 1 Remota NX5000

5.4.2.2. Modos de Operação da Interface Ethernet Remota NX5000

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

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

5.4.2.2.1. Modo Redundante

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

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

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

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

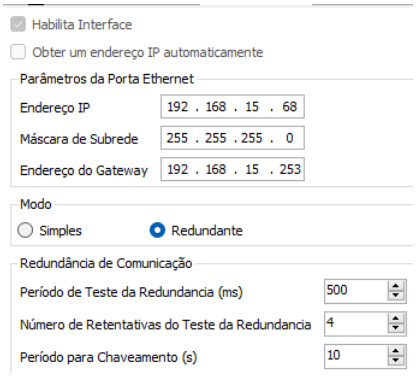


Figura 52: Advanced Configuration of Remote Ethernet Interface - NX5000

5.4.3. Portas TCP/UDP Reservadas

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

Serviço	TCP	UDP
Página Web de Sistema	80	-
SNTP	-	123
SNMP	-	161
MODBUS TCP	502*	-
Mastertool	1217*	1740:1743
SQL Server	1433	-
MQTT	1883* / 8883*	-
EtherNet/IP	44818	2222
IEC 60870-5-104	2404*	-
DNP3	20000* / 20005*	-
OPC UA	4840	-
WEBVISU	8080	-
CODESYS ARTI	11740	-
PROFINET	-	34964
Portainer Docker	9000	-

Tabela 65: Portas TCP/UDP reservadas

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

5.5. Configuração de Protocolos

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

	NX3020
Pontos Mapeados	20480
Mapeamentos (Por Instância / Total)	5120 / 20480
Requisições	512
NETs – Instâncias Cliente ou Servidor (Por NET / Total)	4 / 8
COM (n) – Instâncias Mestre ou Escravo	1
Centros de Controle	3

Tabela 66: Limites dos Protocolos por UCP

Notas:

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

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

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

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

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

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

As limitações do protocolo MODBUS por Representação Direta e por mapeamentos simbólicos para as UCPs podem ser visualizadas na Tabela 67 e na Tabela 68, respectivamente.

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

Tabela 67: Limites do Protocolo MODBUS por Representação Direta

Notas:

Dispositivos por instância:

- Protocolos Mestre ou Cliente: quantidade de dispositivos escravos ou servidores suportados por cada instância de protocolo Mestre ou Cliente.
- Protocolo MODBUS RTU Escravo: o limite (1) informado diz respeito às interfaces seriais, que não permitem que um Escravo possa estabelecer comunicação, simultaneamente, através da mesma interface serial, com mais de um dispositivo

Mestre. Não é necessário, e nem é possível, declarar nem configurar o dispositivo Mestre sob a instância do protocolo MODBUS RTU Escravo. O dispositivo Mestre terá acesso a todos os mapeamentos feitos diretamente na instância do protocolo MODBUS RTU Escravo.

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

Mapeamentos por dispositivo: O número máximo de mapeamentos por dispositivo apesar de estar relacionado acima, também é limitado pelo número máximo de mapeamentos do protocolo. Também deve ser considerado o limite máximo de mapeamentos da UCP conforme na Tabela 66.

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

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

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

Tabela 68: Limites do Protocolo MODBUS por Mapeamentos Simbólicos

Notas:

Dispositivos por instância:

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

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

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

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

ATENÇÃO

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

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

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

Tabela 69: Limites do Protocolo IEC 60870-5-104 Servidor

Notas:

Dispositivos por instância: Quantidade de dispositivos clientes, do tipo centro de controle, suportados por cada instância de protocolo Servidor IEC 60870-5-104. O limite informado pode ser menor em função dos limites totais da UCP (consultar a Tabela 66).

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

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

5.5.1. Comportamento dos Protocolos x Estados da UCP

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

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

Tabela 70: Comportamento dos Protocolos x Estados da UCP

Notas:

Símbolo ✓: O protocolo permanece ativo e operando normalmente.

Símbolo ✗: O protocolo está desabilitado.

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

5.5.2. Pontos Duplos

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

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

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

5.5.3. Fila de Eventos da UCP

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

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

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

ATENÇÃO

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

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

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

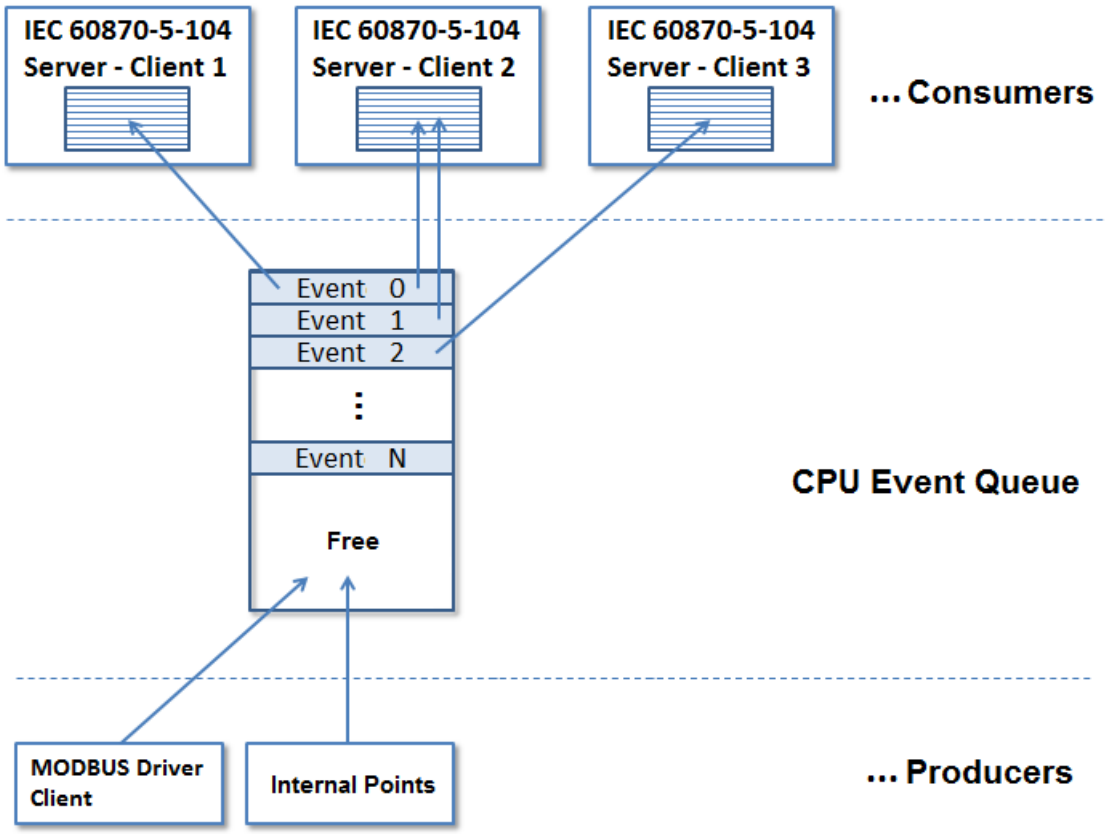


Figura 53: Fila de Eventos da UCP

5.5.3.1. Consumidores

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

5.5.3.2. Princípios de Funcionamento da Fila

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

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

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

5.5.3.2.1. Sinalização de Estouro

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

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

5.5.3.3. Produtores

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

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

5.5.4. Interceptação de Comandos Oriundos do Centro de Controle

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

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

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

Tabela 71: Parâmetros de Entrada do Bloco de Função CommandReceiver

Notas:

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

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

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

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

ATENÇÃO

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

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

Tabela 72: Parâmetros de saída do Bloco de Função CommandReceiver

Nota:

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

Os comandos suportados são descritos na tabela abaixo:

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

Tabela 73: Comandos Suportados do Bloco de Função CommandReceiver

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

Parâmetro	Tipo	Descrição
sSelectConfig	STRUCT	Configuração do comando de seleção recebido. Os parâmetros desta estrutura estão descritos na Tabela 75
sValue	STRUCT	Valor recebido em um select, quando recebido um comando de seleção com valor. Os parâmetros desta estrutura estão descritos na Tabela 78

Tabela 74: Parâmetros sSelectParameters

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

Tabela 75: Parâmetros sSelectConfig

Parâmetro	Tipo	Descrição
sOperateConfig	STRUCT	Configuração do comando de seleção recebido. Os parâmetros desta estrutura estão descritos na Tabela 77
sValue	STRUCT	Campo de valor referente ao comando de operação recebido. Os parâmetros desta estrutura estão descritos na Tabela 78

Tabela 76: Parâmetros sOperateParameters

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

Tabela 77: Parâmetros sOperateConfig

Parâmetro	Tipo	Descrição
eParamType	ENUM	Informa o tipo de comando que foi recebido: NO_COMMAND(0) SINGLE_POINT_COMMAND(1) DOUBLE_POINT_COMMAND(2) INTEGER_STATUS_COMMAND(3) ENUMERATED_STATUS_COMMAND(4) ANALOGUE_VALUE_COMMAND(5)
sSinglePoint	STRUCT	Quando recebido um comando, em função do tipo do comando recebido, definido por eParamType, é preenchida a estrutura de dados correspondente. Os parâmetros destas estruturas são apresentados na Tabela 79 a Tabela 83
sDoublePoint	STRUCT	
sIntegerStatus	STRUCT	
sEnumeratedStatus	STRUCT	
sAnalogueValue	STRUCT	

Tabela 78: Parâmetros sValue

Parâmetro	Tipo	Descrição
bValue	BOOL	Valor de operação do ponto.
sPulseConfig	STRUCT	Os parâmetros de configuração de um comando pulsado são armazenados nesta estrutura. Os parâmetros desta estrutura são apresentados na Tabela 84.

Tabela 79: Parâmetros sSinglePoint

Parâmetro	Tipo	Descrição
bValue	BOOL	Valor de operação do ponto.
sPulseConfig	STRUCT	Os parâmetros de configuração de um comando pulsado são armazenados nesta estrutura. Os parâmetros desta estrutura são apresentados na Tabela 84.

Tabela 80: Parâmetros sDoublePoint

Parâmetro	Tipo	Descrição
diValue	DINT	Valor de operação do ponto.

Tabela 81: Parâmetros sIntegerStatus

Parâmetro	Tipo	Descrição
dwValue	DWORD	Valor de operação do ponto.

Tabela 82: Parâmetros sEnumeratedStatus

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

Tabela 83: Parâmetros sAnalogueValue

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

Tabela 84: Parâmetros sPulseConfig

Para efetuar a interceptação de comandos para um dado ponto, primeiramente deve-se carregar no parâmetro *dwVaria-*

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

5.5.5. MODBUS RTU Mestre

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

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

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

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

5.5.5.1. Configuração do Protocolo MODBUS Mestre por Mapeamento Simbólico

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

- Configurar os parâmetros gerais do protocolo MODBUS Mestre, como: tempos de atraso de envio e interframe mínimo como na Figura 54.
- 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 55.
- Adicionar e configurar os mapeamentos MODBUS na aba Mapeamentos conforme Figura 56, especificando o nome da variável, tipo de dados e endereço inicial do dado, o tamanho do dado e a faixa são preenchidos automaticamente.
- Adicionar e configurar as requisições MODBUS como apresentado na Figura 57, especificando a função, o tempo de varredura da requisição, o endereço inicial (leitura/escrita), o tamanho dos dados (Leitura/Escrita) e gerar as variáveis de diagnóstico e desabilitação das requisições através dos botões na parte inferior da janela.

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

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

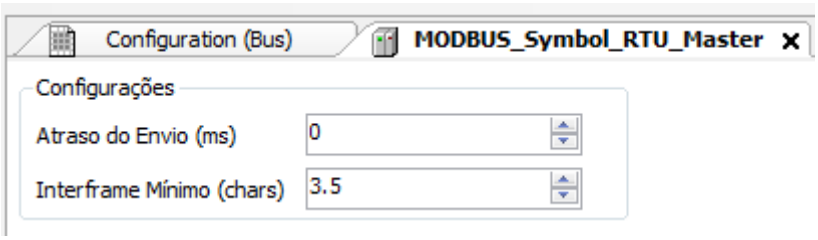


Figura 54: Tela de Configuração Parâmetros Gerais MODBUS RTU Mestre

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

Tabela 85: Configurações Gerais MODBUS RTU Mestre

Notas:

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

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

Os diagnósticos e comandos do protocolo MODBUS Mestre configurado, seja por mapeamento simbólico ou por representação direta, são armazenados em variáveis do tipo *T_DIAG_MODBUS_RTU_MASTER_1* e ainda para o mapeamento por representação direta estão em 4 bytes e 8 words, os quais estão descritos na tabela abaixo:

T_DIAG_MODBUS_RTU_MASTER_1.*	Tamanho	Descrição
Bits de diagnóstico:		
tDiag.bRunning	BIT	O mestre está em execução.
tDiag.bNotRunning	BIT	O mestre não está em execução (ver bit: bInterruptedByCommand).
tDiag.bInterruptedByCommand	BIT	O bit bNotRunning foi habilitado, pois o mestre foi interrompido pelo usuário através de bits de comando.
tDiag.bConfigFailure	BIT	Falha na configuração.
tDiag.bModuleFailure	BIT	Não implementado.
Códigos de erro:		
eErrorCode	SERIAL_STATUS (BYTE)	0: não existem erros 1: porta serial inválida 2: modo da porta serial inválido 3: taxa de transmissão inválida 4: bits de dado inválidos 5: bits de paridade inválidos 6: bits de parada inválidos 7: parâmetro de sinal de modem inválido 8: parâmetro de Threshold de RX da UART inválido 9: parâmetro de time-out inválido 10: porta serial ocupada 11: erro de hardware na UART 12: erro de hardware remoto 20: tamanho do buffer de transmissão inválido 21: método de sinal de modem inválido 22: time-out de CTS = verdadeiro 23: time-out de CTS = falso 24: erro de time-out na transmissão 30: tamanho do buffer de recepção inválido 31: erro de time-out na recepção 32: controle de fluxo configurado diferente de manual 33: controle de fluxo inválido para a porta serial configurada 34: recepção de dados não permitida no modo normal 35: recepção de dados não permitida no modo estendido 36: interrupção DCD não permitida 37: interrupção CTS não permitida 38: interrupção DSR não permitida 39: porta serial não configurada 50: erro interno na porta serial
Bits de comando, reiniciados automaticamente:		

T_DIAG_MODBUS_RTU_MASTER_1.*	Tamanho	Descrição
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).
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 86: Diagnósticos MODBUS RTU Mestre

Nota:

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

5.5.5.1.2. Configuração dos Dispositivos – Configuração por Mapeamento Simbólico

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

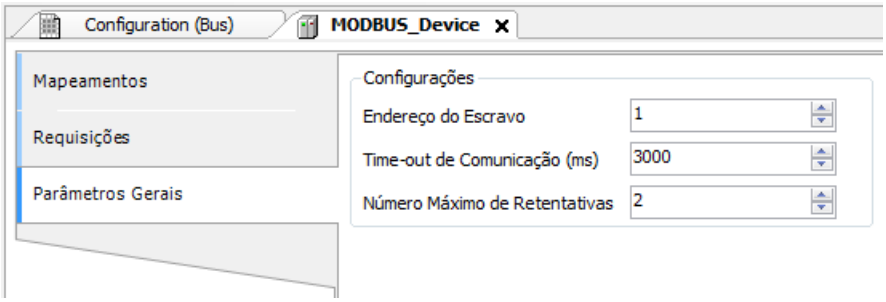


Figura 55: Tela de Configurações dos Parâmetros Gerais do Dispositivo

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

Tabela 87: Configurações do Dispositivo

Notas:

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

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

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

5.5.5.1.3. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico

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

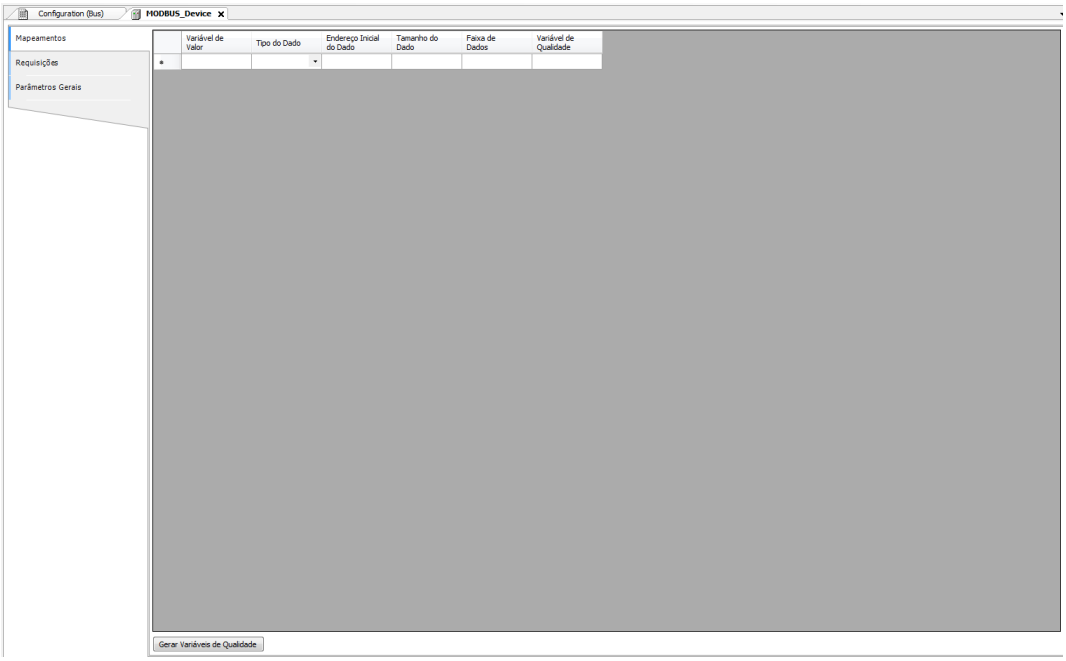


Figura 56: Tela de Mapeamentos de Dados MODBUS

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

Configuração	Descrição	Padrão	Opções
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.

5.5.5.1.4. Configuração das Requisições – Configuração por Mapeamento Simbólico

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

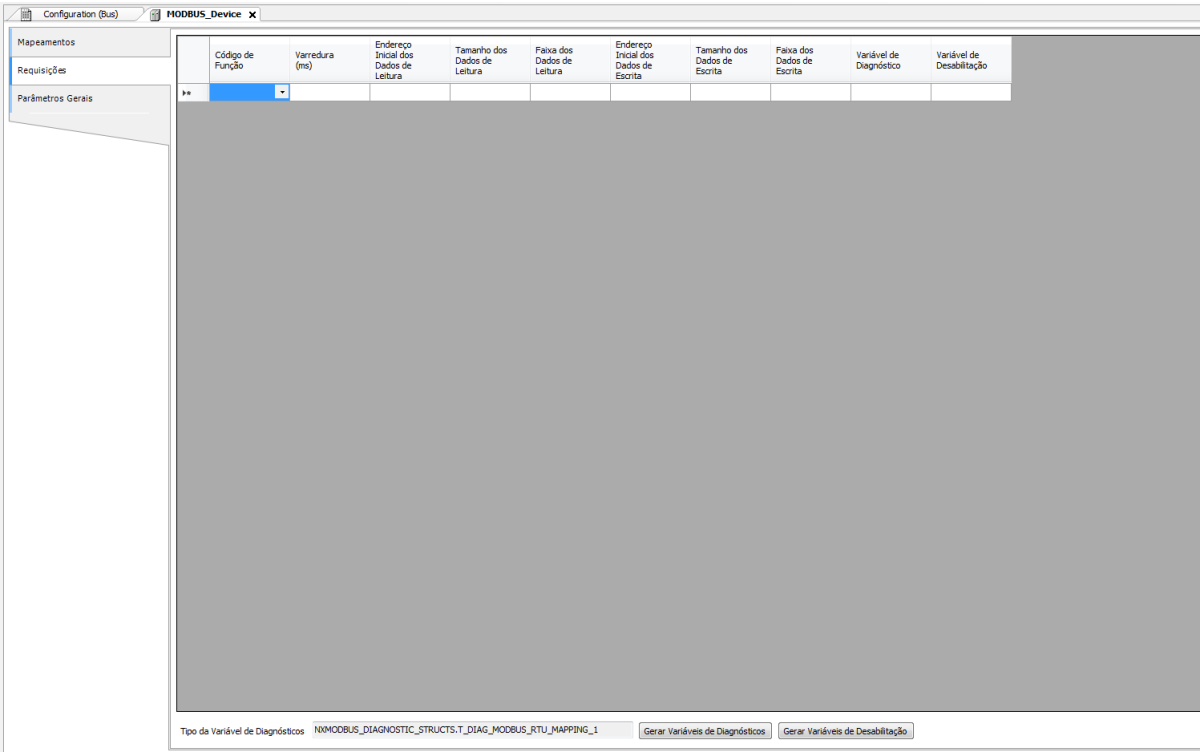


Figura 57: Tela de Requisições de dados MODBUS Mestre

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

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

Tabela 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 pelas UCPs Nexto

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

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

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

- Leitura de Coils (FC 1): 2000
- Leitura de Input Status (FC 2): 2000
- Leitura de Holding Registers (FC 3): 125
- Leitura de Input Registers (FC 4): 125
- Leitura/Escrita de Holding Registers (FC 23): 121

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

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

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

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

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

Variável de Diagnóstico: Os diagnósticos da requisição MODBUS configurada, seja por mapeamento simbólico ou por representação direta , são armazenados em variáveis do tipo *T_DIAG_MODBUS_RTU_MAPPING_1* para dispositivos Mestre e *T_DIAG_MODBUS_ETH_CLIENT_1* para dispositivos Cliente e para o mapeamento por representação direta estão em 4 bytes e 2 words, os quais estão descritos na Tabela 92.

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 115 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 92: 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 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 RTU Mestre, e qualquer outra tarefa MODBUS, irá parar de ser executada no momento em que tentar efetuar uma escrita em uma área de memória. Isto ocorre para manter a consistência dos dados das áreas de memória enquanto a MainTask não estiver em execução.

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

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

- Configurar os parâmetros gerais do protocolo MODBUS, como: tempos de comunicação e variáveis de representação direta (%Q) para receber os diagnósticos.
- Adicionar e configurar dispositivos, definindo endereço, variáveis de representação direta (%Q) para desabilitar as relações, time-outs de comunicação, etc.

- Adicionar e configurar relações MODBUS, especificando o tipo de dado e função MODBUS, time-outs, variáveis de representação direta (%Q) para receber os diagnósticos da relação e outras para receber/escrever os dados, quantidade de dados a comunicar e varredura da relação.

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

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

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

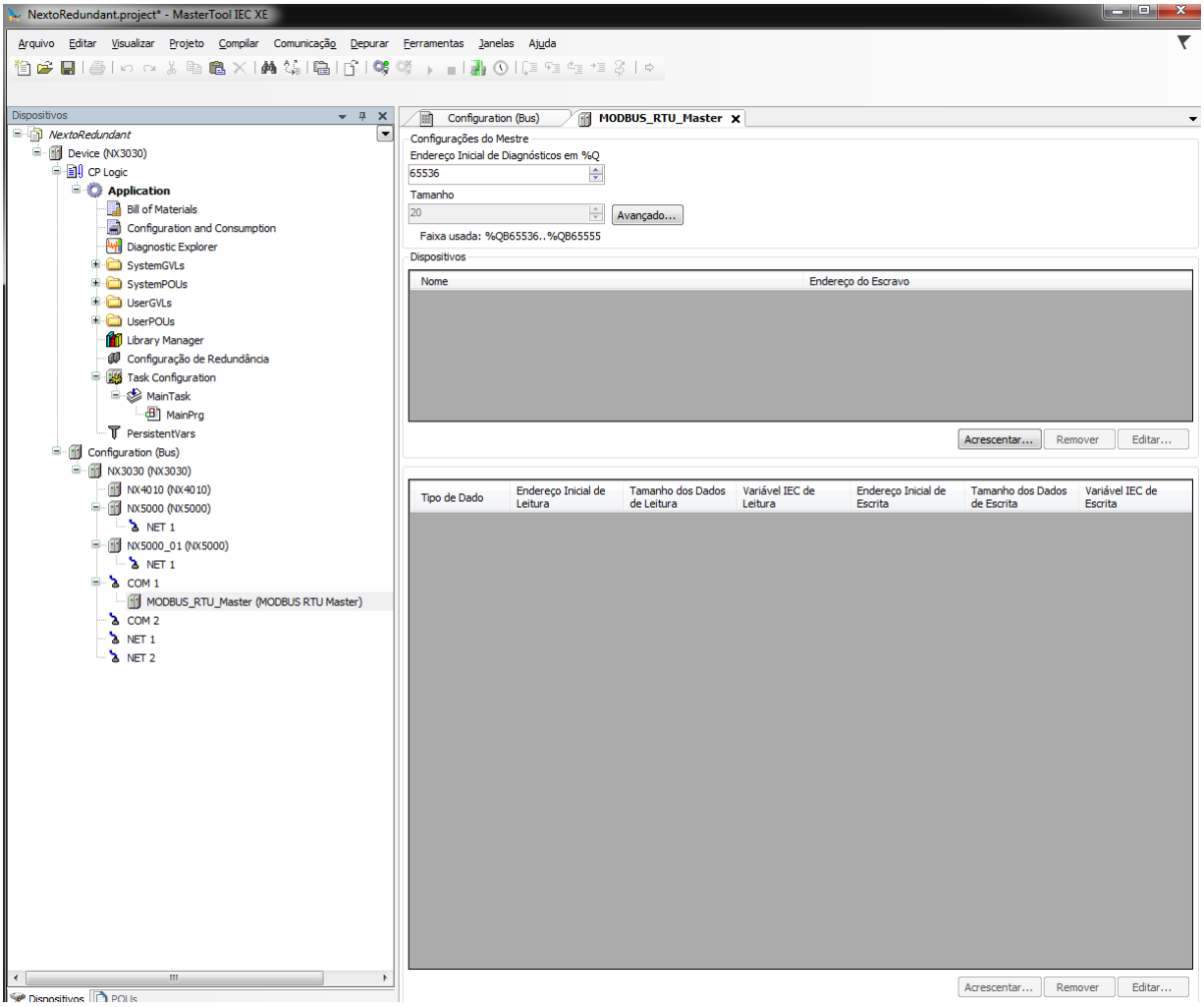


Figura 58: Tela de Configuração MODBUS RTU Mestre

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

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

Tabela 94: Configurações do Mestre – MODBUS Mestre

Notas:

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

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

Os diagnósticos e comandos do protocolo MODBUS estão descritos na Tabela 86.

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

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

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

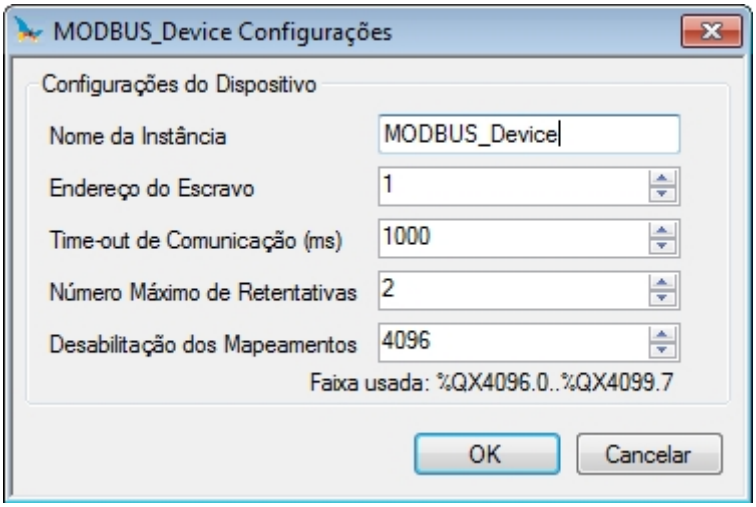


Figura 59: Configurações do Dispositivo

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

Tabela 95: Configurações do Dispositivo – MODBUS Mestre

Notas:

Nome da Instância: Esse campo é o identificador do dispositivo, o qual é verificado segundo a IEC 61131-3, ou seja, não

permite espaços, caracteres especiais e iniciar com caractere numeral. É limitado em 24 caracteres.

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

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

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

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

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

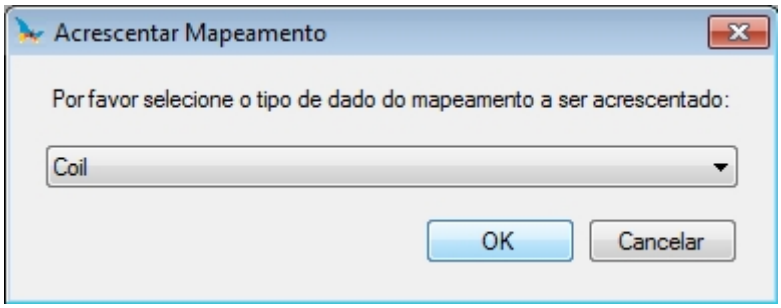


Figura 60: Tipo de Dado MODBUS

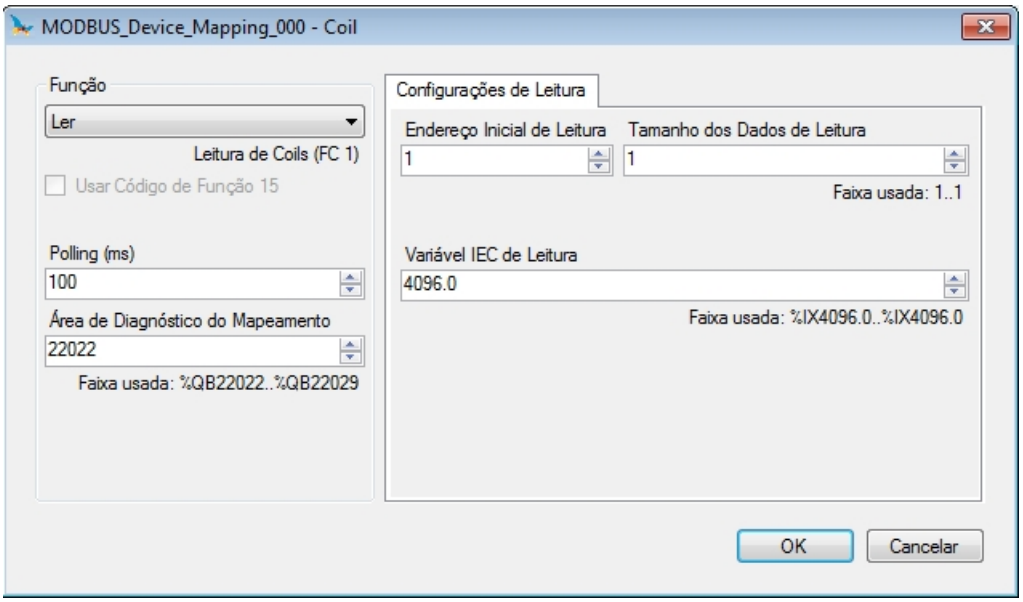


Figura 61: Função MODBUS

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

Configuração	Descrição	Padrão	Opções
Função	Tipo de função MODBUS	Ler	Ler Escrever Ler/Escrever Máscara de Escrita
Polling (ms)	Período de comunicação (ms)	100	0 a 3600000
Área de Diagnóstico do Mapeamento	Endereço inicial dos diagnósticos da relação MODBUS (%Q)	-	0 a 2147483640
Endereço Inicial de Leitura	Endereço inicial dos dados de leitura MODBUS	1	1 a 65536
Tamanho dos Dados de Leitura	Número de dados de leitura MODBUS	-	Depende da função utilizada
Variável IEC de Leitura	Endereço inicial das variáveis de leitura (%I)	-	0 a 2147483646
Endereço Inicial de Escrita	Endereço inicial dos dados de escrita MODBUS	1	1 a 65536
Tamanho dos Dados de Escrita	Número de dados de escrita MODBUS	-	Depende da função utilizada
Variável IEC de Escrita	Endereço inicial das variáveis de escrita (%Q)	-	0 a 2147483647
Máscara de Escrita das Variáveis IEC	Endereço inicial das variáveis para a máscara de escrita (%Q)	-	0 a 2147483644

Tabela 96: Mapeamentos do Dispositivo

Notas:

Função: Os tipos de dados disponíveis estão detalhados na Tabela 115 e as funções MODBUS (FC) disponíveis na Tabela 113.

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

Área de Diagnósticos do Mapeamento: Esse campo é limitado pelo tamanho da memória de variáveis de saídas endereçáveis (%Q) de cada UCP, a qual pode ser consultada na seção Memória. Os diagnósticos da relação MODBUS configurada estão descritos na Tabela 92.

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

ATENÇÃO

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

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

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

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

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

ATENÇÃO

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

5.5.6. MODBUS RTU Escravo

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

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

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

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

5.5.6.1. Configuração do Protocolo MODBUS Escravo por Mapeamento Simbólico

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

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

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

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

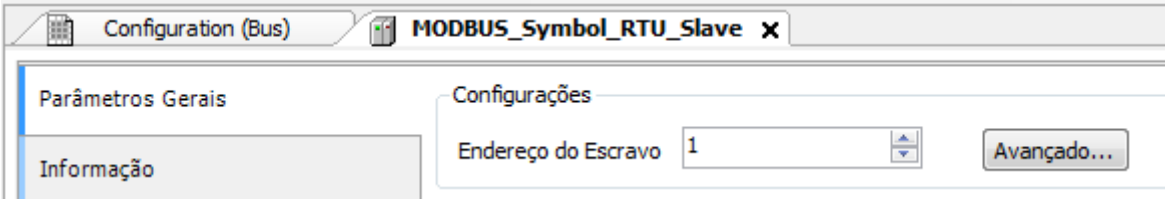


Figura 62: Configuração do Escravo

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

Tabela 97: Configurações do Escravo

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

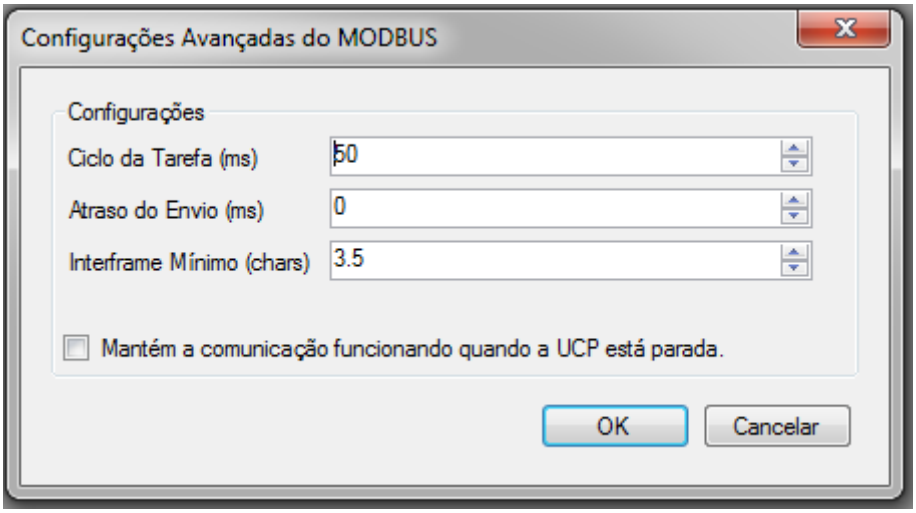


Figura 63: Configurações Avançadas MODBUS do Escravo

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

Tabela 98: Configurações Avançadas MODBUS Escravo

Notas:

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

Atraso do Envio: A resposta à uma requisição MODBUS pode causar problemas em certos momentos, como, por exemplo, na interface RS-485 ou outra half-duplex. Às vezes existe um atraso entre o tempo da requisição do mestre e o silêncio na linha física (atraso no mestre para colocar RTS em zero e colocar o transmissor RS-485 em alta impedância). Para resolver o

problema, o escravo pode esperar o tempo determinado nesse campo antes de enviar a resposta. Caso contrário, os primeiros bytes transmitidos pelo escravo, durante a resposta, podem ser perdidos.

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

Os diagnósticos e comandos do protocolo MODBUS Escravo configurado, seja por mapeamento simbólico ou por representação direta, são armazenados em variáveis do tipo *T_DIAG_MODBUS_RTU_SLAVE_1* e ainda para o mapeamento por representação direta estão em 4 bytes e 8 words, os quais estão descritos na tabela abaixo:

T_DIAG_MODBUS_RTU_SLAVE_1.*	Tamanho	Descrição
Bits de diagnóstico:		
tDiag.bRunning	BIT	O escravo está em execução.
tDiag.bNotRunning	BIT	O escravo não está em execução (ver bit: bInterruptedByCommand).
tDiag.bInterruptedByCommand	BIT	O bit bNotRunning foi habilitado, pois o escravo foi interrompido pelo usuário através de bits de comando.
tDiag.bConfigFailure	BIT	Falha na configuração.
tDiag.bModuleFailure	BIT	Não implementado.
Códigos de erro:		
eErrorCode	SERIAL_STATUS (BYTE)	0: não existem erros 1: porta serial inválida 2: modo da porta serial inválido 3: taxa de transmissão inválida 4: bits de dado inválidos 5: bits de paridade inválidos 6: bits de parada inválidos 7: parâmetro de sinal de modem inválido 8: parâmetro de Threshold de RX da UART inválido 9: parâmetro de time-out inválido 10: porta serial ocupada 11: erro de hardware na UART 12: erro de hardware remoto 20: tamanho do buffer de transmissão inválido 21: método de sinal de modem inválido 22: time-out de CTS = verdadeiro 23: time-out de CTS = falso 24: erro de time-out na transmissão 30: tamanho do buffer de recepção inválido 31: erro de time-out na recepção 32: controle de fluxo configurado diferente de manual 33: controle de fluxo inválido para a porta serial configurada 34: recepção de dados não permitida no modo normal 35: recepção de dados não permitida no modo estendido 36: interrupção DCD não permitida 37: interrupção CTS não permitida 38: interrupção DSR não permitida 39: porta serial não configurada 50: erro interno na porta serial
Bits de comando, reiniciados automaticamente:		
tCommand.bStop	BIT	Parar o escravo.
tCommand.bRestart	BIT	Reiniciar o escravo.
tCommand.bResetCounter	BIT	Reiniciar as estatísticas dos diagnósticos (contadores).
Estatísticas de comunicação:		
tStat.wRXRequests	WORD	Contador de requisições normais recebidas pelo escravo e respondidas normalmente. No caso de um comando broadcast, esse contador é incrementado, porém não é transmitida a resposta (0 a 65535).
tStat.wTXExceptionResponses	WORD	Contador de requisições normais recebidas pelo escravo e respondidas com códigos de exceção. No caso de um comando broadcast, esse contador é incrementado, porém não é transmitida a resposta (0 a 65535).

T_DIAG_MODBUS_RTU_SLAVE_1.*	Tamanho	Descrição
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 99: Diagnósticos MODBUS RTU Escravo

Nota:

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

5.5.6.1.2. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico

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

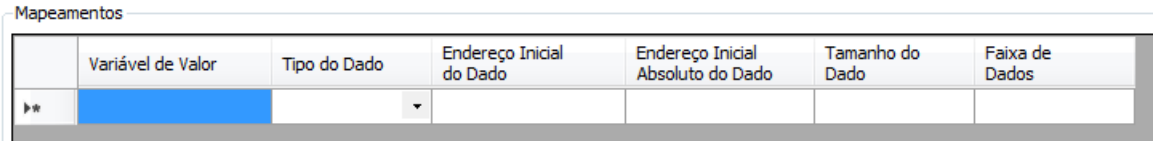


Figura 64: Tela de Mapeamentos de dados MODBUS

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

Tabela 100: Configuração dos Mapeamentos MODBUS

Notas:

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

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

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

Tabela 101: Tipos de Dados MODBUS suportados pelas UCPs Nexto

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

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

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

ATENÇÃO

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

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

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

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

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

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

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

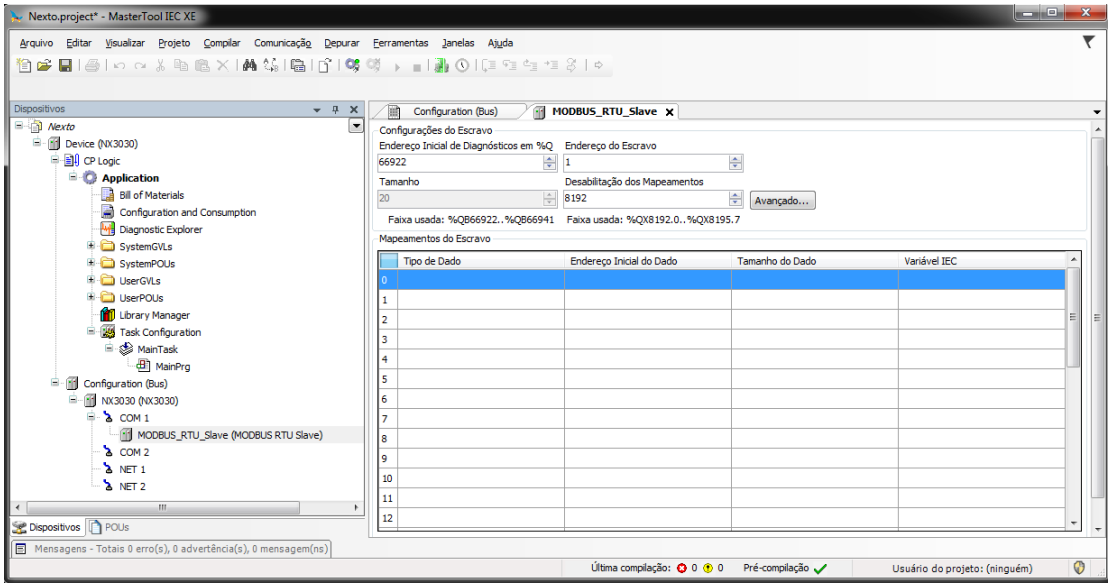


Figura 65: Tela de Configuração MODBUS RTU Escravo por Representação Direta

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

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

Tabela 102: Configurações das Variáveis de Representação Direta e Endereçamento

Notas:

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

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

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

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

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

Os diagnósticos e comandos do protocolo MODBUS estão descritas na Tabela 99.

Os tempos de comunicação do protocolo MODBUS Escravo, encontrados no botão *Avançado...* da tela de configuração, estão descritos na Tabela 98.

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

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

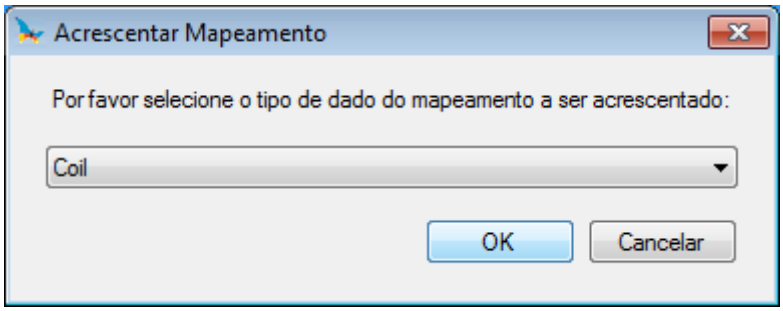


Figura 66: Adicionando Relações MODBUS

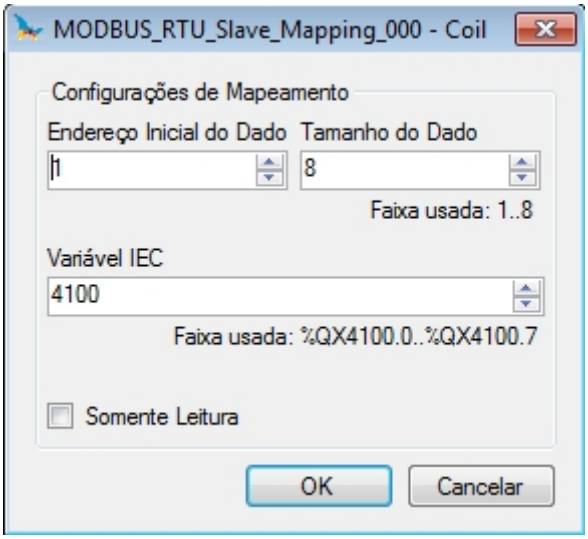


Figura 67: Configurando a Relação MODBUS

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

Tabela 103: Mapeamentos do Escravo

Notas:

Opções: Os valores descritos na coluna *Opções* podem variar de acordo com o tipo de dado MODBUS configurado.

Tamanho do Dado: O valor de *Tamanho do Dado* define a quantidade máxima de dados que uma relação MODBUS poderá acessar, a partir do endereço inicial. Sendo assim, para ler uma faixa de endereços contínua, é necessário que todos os endereços estejam declarados em uma única relação. Este campo varia de acordo com o tipo de dado MODBUS configurado, ou seja, quando selecionado tipo *Coil* ou *Input Status*, o campo Tamanho do Dado deve ser um número múltiplo de oito. Também deve-se dar atenção para que o valor máximo não seja superior ao tamanho da memória de saídas endereçáveis e não sejam atribuídos os mesmos valores já utilizados durante a aplicação.

ATENÇÃO

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

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

Somente Leitura: Quando habilitada, somente permite que o mestre da comunicação leia os dados das variáveis, não permitindo a escrita. Opção válida somente para as funções de escrita.

Padrão: O padrão de fábrica não pode ser definido para o campo *Variável IEC*, pois a criação de uma instância do protocolo pode ser realizada em qualquer momento no desenvolvimento da aplicação, fazendo com que o próprio software MasterTool IEC XE aloque um valor, da faixa de variáveis de saída de representação direta (%Q), ainda não utilizado. O padrão de fábrica não pode ser definido para o campo *Tamanho do Dado*, pois ele vai variar de acordo com o tipo de dado MODBUS selecionado.

Nas relações definidas anteriormente, o tamanho máximo de dados MODBUS pode ser 65535 (máximo valor configurado no campo *Tamanho do Dado*). Porém, a pergunta que chega no MODBUS RTU Escravo deverá endereçar um subconjunto desse mapeamento e esse grupo deve ter, no máximo, o tamanho de dados que depende do código da função, os quais estão definidos abaixo:

- Leitura de Coils (FC 1): 2000
- Leitura de Input Status (FC 2): 2000
- Leitura de Holding Registers (FC 3): 125
- Leitura de Input Registers (FC 4): 125
- Escrita de Um Coil (FC 5): 1
- Escrita de Um Holding Register (FC 6): 1
- Escrita de Múltiplos Coils (FC 15): 1968
- Escrita de Holding Registers (FC 16): 123
- Máscara de Escrita do Register (FC 22): 1
- Leitura/Escrita de Holding Registers (FC 23):
  - Leitura: 121
  - Escrita: 121

ATENÇÃO

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

5.5.7. MODBUS Ethernet

A rede de comunicação multimestre permite que as UCPs Nexto leiam ou escrevam variáveis MODBUS em outros controladores ou IHMs compatíveis com os protocolos MODBUS TCP ou MODBUS RTU via TCP. A UCP Nexto pode, simultaneamente, ser cliente e servidor em uma mesma rede de comunicação, ou até mesmo ter mais instâncias associadas à interface Ethernet, indiferente se as mesmas são MODBUS TCP ou MODBUS RTU via TCP, conforme descreve a Tabela 66.

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

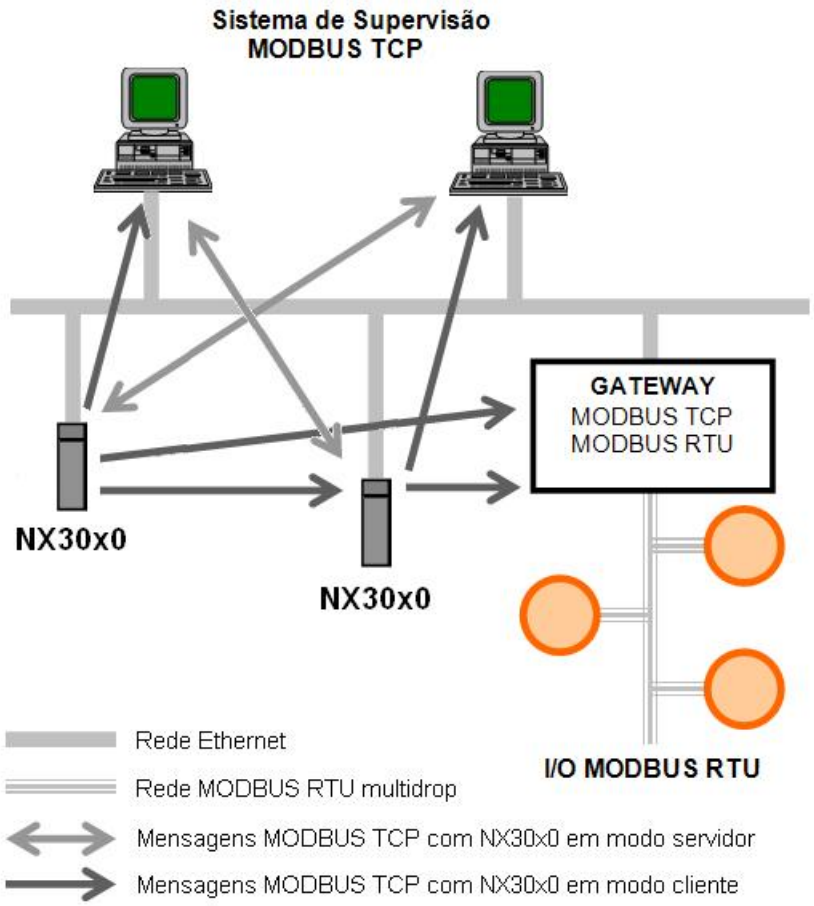


Figura 68: Rede de Comunicação MODBUS TCP

A associação de variáveis MODBUS com variáveis simbólicas da UCP é realizada pelo usuário através da definição de relações via configurador MasterTool IEC XE. Podem ser definidas até 32 relações para o modo servidor e até 128 relações para o modo cliente. Uma relação, em modo servidor, pode definir uma grande área de dados MODBUS e torná-la disponível para vários clientes. As relações em modo cliente, por outro lado, devem respeitar o tamanho máximo de dados de uma função MODBUS: 125 registradores (input registers ou holding registers) ou 2000 bits (coils ou input status). Essas informações são detalhadas na descrição de cada protocolo.

Todas as relações, em modo cliente ou servidor, podem ser desabilitadas através de variáveis de representação direta (%Q) identificadas como Desabilitação dos Mapeamentos pelo MasterTool IEC XE. A desabilitação pode ocorrer através de bits gerais, os quais afetam todas as relações de um modo de operação, ou através de bits específicos, afetando relações específicas.

Para as relações em modo servidor, podem ser definidos conjuntos de endereços IPs com permissão de escrita e leitura, chamados de filtros. Isto é feito através da definição de um endereço de rede IP e de uma máscara de subrede, resultando em um grupo de IPs clientes que podem escrever e ler nas variáveis da relação. Funções de leitura/escrita são filtradas da mesma forma que as funções exclusivas de leitura ou escrita. Essas informações são detalhadas na descrição do protocolo MODBUS Ethernet Servidor.

Quando o protocolo MODBUS TCP é utilizado no modo cliente, pode-se usufruir da característica de múltiplas requisições, utilizando a mesma conexão TCP para acelerar a comunicação com os servidores. Quando esta característica não for desejada ou não for suportada pelo servidor, ela pode ser desabilitada (ação em nível de relação). É importante destacar que o número máximo de conexões TCP entre cliente e servidor é 63, sendo que se alguns parâmetros forem alterados, comunicações inativas podem ser fechadas, possibilitando a abertura de novas conexões.

As tabelas abaixo trazem, respectivamente, a lista completa dos tipos de dados e funções MODBUS suportadas pelas UCPs Nexto.

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

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

Tabela 104: Tipos de Dados MODBUS suportados pelas UCPs Nexto

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

Tabela 105: Funções MODBUS Suportadas pelas UCPs Nexto

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

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

5.5.8. MODBUS Ethernet Cliente

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

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

O procedimento para inserir uma instância de protocolo é encontrado com detalhes no Manual de Utilização do MasterTool IEC XE – MU299048 ou na seção Inserindo uma Instância de Protocolo.

5.5.8.1. Configuração do Protocolo MODBUS Ethernet Cliente por Mapeamento Simbólico

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

- Configurar os parâmetros gerais do protocolo MODBUS Cliente, com o protocolo TCP ou RTU via TCP.
- Adicionar e configurar dispositivos, definindo endereço IP, porta, endereço do escravo e time-out de comunicação (disponível no botão de configurações avançadas do Dispositivo).
- Adicionar e configurar os mapeamentos MODBUS, especificando o nome da variável, tipo de dados, endereço inicial do dado, tamanho do dado e variável que receberá os dados de qualidade.
- Adicionar e configurar as requisições MODBUS, especificando a função desejada, o tempo de varredura da requisição, o endereço inicial (leitura/escrita), o tamanho dos dados (Leitura/Escrita), a variável que receberá os dados de qualidade, e a variável responsável por desabilitar a requisição.

5.5.8.1.1. *Parâmetros Gerais do Protocolo MODBUS Cliente – Configuração por Mapeamento Simbólico*

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

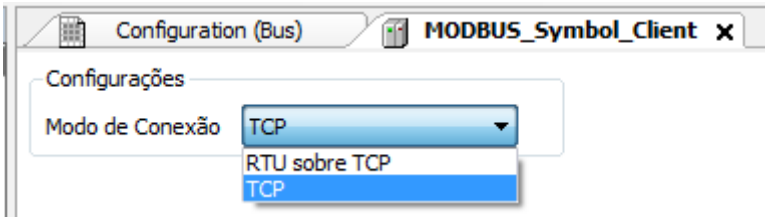


Figura 69: Tela de Configuração Parâmetros Gerais MODBUS Cliente

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

Tabela 106: Configurações Gerais MODBUS Cliente

Os diagnósticos e comandos do protocolo MODBUS Cliente configurado, seja por mapeamento simbólico ou por representação direta, são armazenados em variáveis do tipo *T_DIAG_MODBUS_ETH_CLIENT_1* e ainda para o mapeamento por representação direta estão em 4 bytes e 8 words, os quais estão descritos na tabela abaixo:

T_DIAG_MODBUS_ETH_CLIENT_1.*	Tamanho	Descrição
Bits de diagnóstico:		
tDiag.bRunning	BIT	O cliente está em execução.
tDiag.bNotRunning	BIT	O cliente não está em execução (ver bit 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 107: Diagnósticos MODBUS Cliente

Nota:

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

5.5.8.1.2. *Configuração dos Dispositivos – Configuração por Mapeamento Simbólico*

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

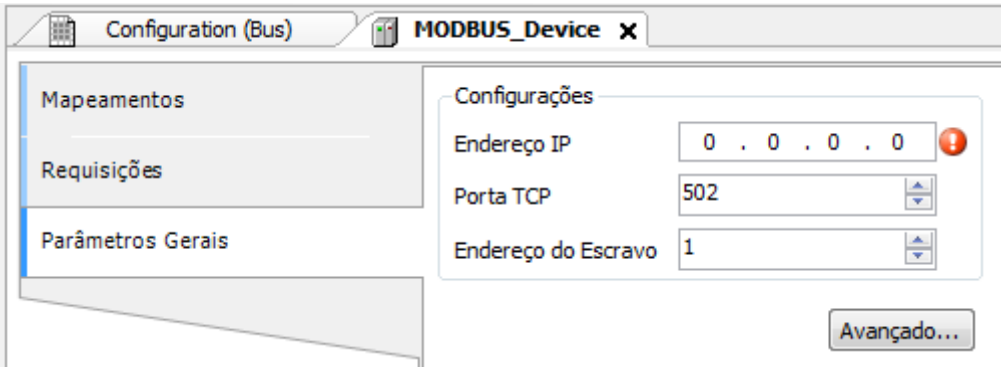


Figura 70: Tela de Configurações dos Parâmetros Gerais do Dispositivo

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

Tabela 108: Parâmetros Gerais MODBUS Cliente

Notas:

Endereço IP: Endereço IP do dispositivo servidor MODBUS.

Porta TCP: Caso sejam adicionadas várias instâncias do protocolo em uma única interface Ethernet, diferentes portas TCP devem ser selecionadas para cada instância. Algumas portas TCP, entre as possibilidades mencionadas acima, são reservadas e, portanto, não podem ser utilizadas. Ver tabela Portas TCP/UDP reservadas.

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

Os parâmetros nas configurações avançadas do dispositivo MODBUS Cliente, encontrados no botão *Avançado...* na aba de *Parâmetros Gerais*, estão divididos em: *Número Máximo de Requisições Simultâneas*, *Time-out de Comunicação*, *Modo* de time-out da conexão e *Tempo de Inatividade*.

Configuração	Descrição	Padrão	Opções
Número Máximo de Requisições Simultâneas	Número de requisições simultâneas que o cliente pode fazer ao servidor	1	1 a 8
Time-out de Comunicação	Time-out do nível da aplicação em ms	3000	10 a 65535
Modo	Define quando a conexão com o servidor é finalizada pelo cliente	Conexão é fechada após um tempo de inatividade de (s): 10 a 3600.	Conexão é fechada depois de um time-out. Conexão é fechada ao final de cada comunicação. Conexão é fechada após um tempo de inatividade de (s): 10 a 3600.

Configuração	Descrição	Padrão	Opções
Tempo de Inatividade	Tempo de inatividade	10	3600

Tabela 109: Configurações Avançadas MODBUS Cliente

Notas:

Número Máximo de Requisições Simultâneas: É utilizado em servidores com um alto ciclo de varredura. Esse parâmetro é fixado em 1 (não editável), quando o protocolo configurado é MODBUS RTU via TCP.

Time-out de Comunicação: O time-out da comunicação é o tempo que o cliente aguardará por uma resposta do servidor à requisição. Para um dispositivo MODBUS Cliente, duas variáveis do sistema devem ser consideradas: o tempo que o servidor leva para processar a requisição e o atraso de envio da resposta caso seja configurado no servidor. É recomendado que o time-out seja igual ou maior que duas vezes a soma destes parâmetros. Para mais informações, ver seção Desempenho de Comunicação.

Modo: Define quando a conexão com o servidor é finalizada pelo cliente. Seguem as opções:

- Conexão é fechada depois de um time-out ou Conexão nunca é fechada em situações normais: Estas opções apresentam o mesmo comportamento do Cliente fechar a conexão devido ao fato do Servidor não ter respondido a uma requisição antes do Time-out de Comunicação ter se esgotado.
- Conexão é fechada ao final de cada comunicação: A conexão é fechada pelo Cliente após concluir cada requisição.
- Conexão é fechada após um tempo de inatividade: A conexão será fechada pelo Cliente caso ele fique por um tempo igual ao Tempo de Inatividade sem realizar requisição para o Servidor.

Tempo de Inatividade: Tempo de inatividade da conexão.

5.5.8.1.3. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico

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

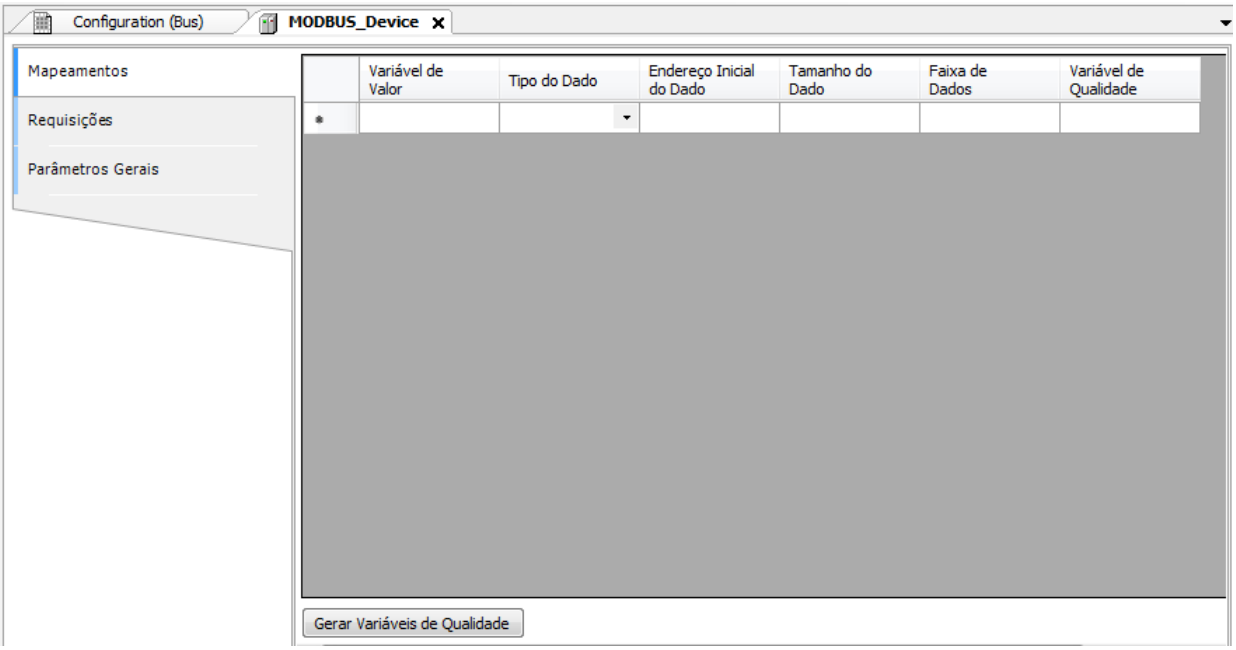


Figura 71: Tela de Mapeamentos de dados MODBUS Cliente

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

Tabela 110: 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 111: Tipos de Dados suportados no MODBUS

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

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

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

5.5.8.1.4. *Configuração das Requisições – Configuração por Mapeamento Simbólico*

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

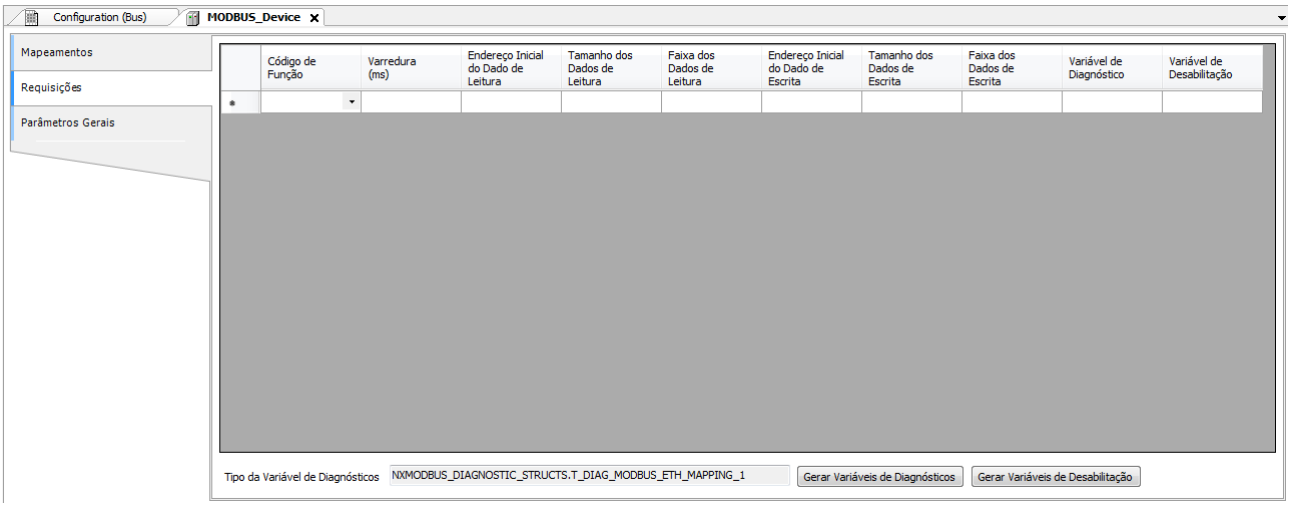


Figura 72: Tela de Requisições de dados MODBUS

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

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

Tabela 112: 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 113: Funções MODBUS Suportadas pelas UCPs Nexto

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

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

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

- Leitura de Coils (FC 1): 2000
- Leitura de Input Status (FC 2): 2000
- Leitura de Holding Registers (FC 3): 125
- Leitura de Input Registers (FC 4): 125
- Leitura/Escrita de Holding Registers (FC 23): 121

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

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

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

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

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

Variável de Diagnóstico: Os diagnósticos da requisição MODBUS configurada, seja por mapeamento simbólico ou por representação direta , são armazenados em variáveis do tipo *T_DIAG_MODBUS_RTU_MAPPING_1* para dispositivos Mestre e *T_DIAG_MODBUS_ETH_CLIENT_1* para dispositivos Cliente e para o mapeamento por representação direta estão em 4 bytes e 2 words, os quais estão descritos na Tabela 92.

T_DIAG_MODBUS_ETH_MAPPING_1.*	Tamanho	Descrição
Bits de estado da comunicação:		
byStatus.bCommIdle	BIT	Comunicação inativa (aguardando ser constada).
byStatus.bCommExecuting	BIT	Comunicação ativa.
byStatus.bCommPostponed	BIT	Comunicação adiada, pois o número máximo de requisições simultâneas foi atingido. As comunicações adiadas serão executadas na mesma sequência em que elas foram requisitadas, evitando a indeterminação. O tempo gasto nesse estado não é contabilizado para efeitos de time-out. Os bits bCommIdle e bCommExecuting são falsos quando o bit bCommPostponed é verdadeiro.
byStatus.bCommDisabled	BIT	Comunicação desabilitada. O bit bCommIdle é reiniciado nessa condição.
byStatus.bCommOk	BIT	Comunicação finalizada anteriormente foi realizada com sucesso.
byStatus.bCommError	BIT	Comunicação finalizada anteriormente teve um erro. Verificar código de erro.
byStatus.bCommAborted	BIT	Comunicação finalizada anteriormente foi interrompida devido à falha de conexão.
Último código de erro (habilitado quando o bCommError = verdadeiro):		
eLastErrorCode	MASTER_ERROR_CODE (BYTE)	Informa a possível causa do último erro ocorrido na relação MODBUS. Consulte a Tabela 115 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 114: Diagnósticos das relações MODBUS Cliente

Notas:

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

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

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

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

Tabela 115: Códigos de Erro das relações MODBUS

ATENÇÃO

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

5.5.8.2. Configuração do Protocolo MODBUS Ethernet Cliente por Representação Direta (%Q)

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

- Configurar os parâmetros gerais do protocolo MODBUS, como: tempos de comunicação e variáveis de representação direta (%Q) para receber os diagnósticos.
- Adicionar e configurar dispositivos, definindo endereço, variáveis de representação direta (%Q) para desabilitar as relações, time-outs de comunicação, etc.

- Adicionar e configurar relações MODBUS, especificando o tipo de dado e função MODBUS, time-outs, variáveis de representação direta (%Q) para receber os diagnósticos da relação e outras para receber/escrever os dados, quantidade de dados a comunicar e varredura da relação.

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

5.5.8.2.1. Parâmetros Gerais do Protocolo MODBUS Cliente – Configuração por Representação Direta (%Q)

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

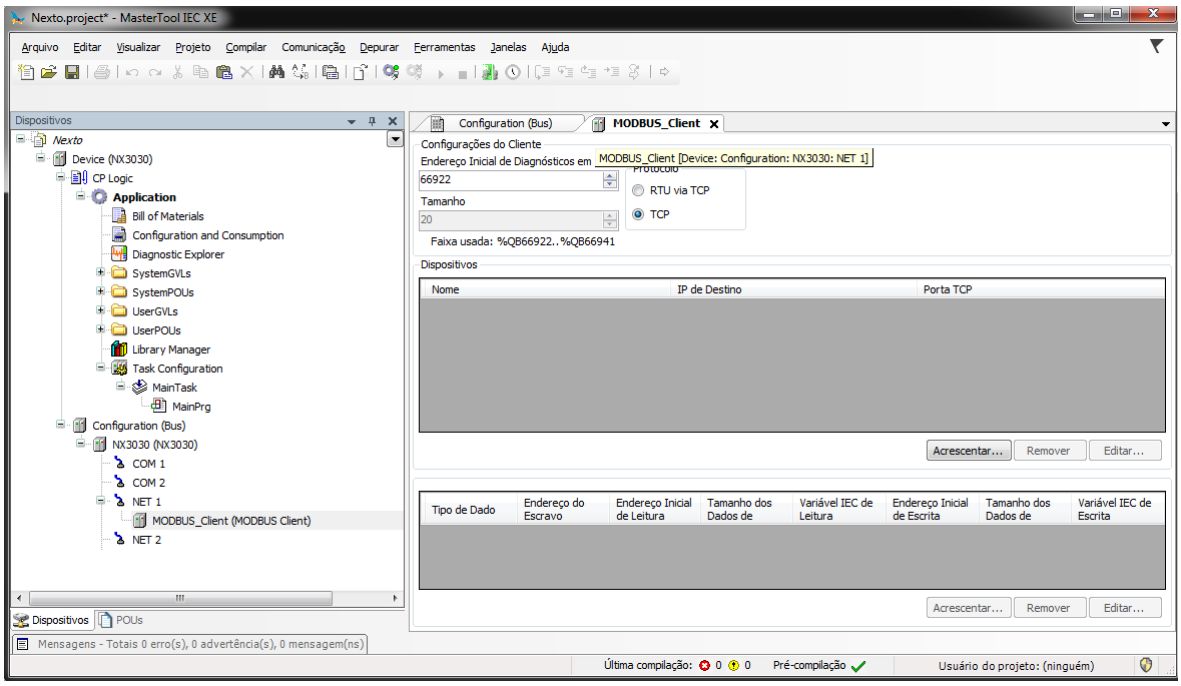


Figura 73: Tela de Configuração – MODBUS Cliente

Seleção do protocolo e variáveis de representação direta (%Q) para os diagnósticos:

Configuração	Descrição	Padrão	Opções
Endereço Inicial de Diagnósticos em %Q	Endereço inicial das variáveis de diagnóstico	-	0 a 2147483628
Tamanho	Tamanho da área de diagnósticos	20	Desabilitado para edição
Protocolo	Seleção do protocolo	TCP	RTU via TCP TCP

Tabela 116: Configurações do MODBUS Cliente

Notas:

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

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

Os diagnósticos e comandos do protocolo MODBUS estão descritos na Tabela 107.

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

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

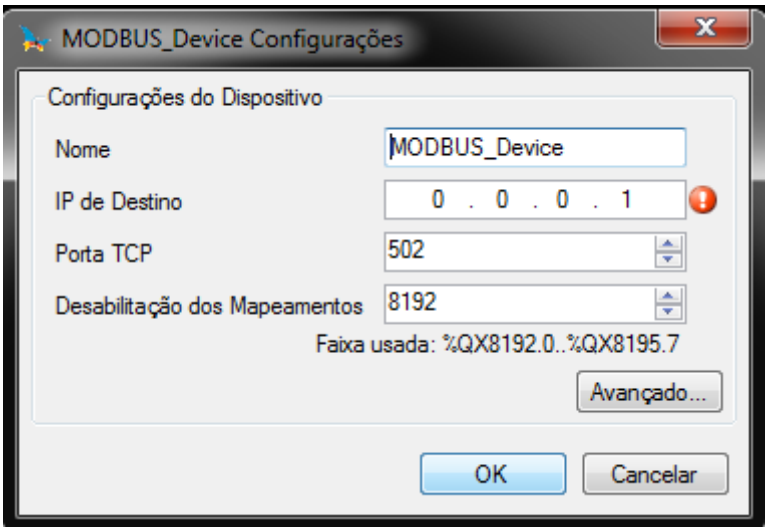


Figura 74: Configurando Cliente MODBUS

Configuração	Descrição	Padrão	Opções
Nome da Instância	Nome da instância	MODBUS_Device	Identificador, segundo a IEC 61131-3
IP de Destino	Endereço IP do servidor	0. 0. 0.1	1.0.0.1 a 223.255.255.255
Porta TCP	Porta TCP	502	2 a 65534
Desabilitação dos Mapeamentos	Endereço inicial utilizado para desabilitar as relações MODBUS	-	Qualquer endereço da área de %Q, limitado conforme UCP utilizada

Tabela 117: Configurações do Dispositivo – MODBUS Cliente

Notas:

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

Porta TCP: Caso sejam adicionadas várias instâncias do protocolo em uma única interface Ethernet, diferentes portas TCP devem ser selecionadas para cada instância. Algumas portas TCP, entre as possibilidades mencionadas acima, são reservadas e, portanto, não podem ser utilizadas. Ver tabela Portas TCP/UDP reservadas.

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

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

Time-out de comunicação: As configurações presentes no botão *Avançado...*, relativas à conexão TCP, estão descritas nas notas da seção Configuração dos Dispositivos – Configuração por Mapeamento Simbólico.

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

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

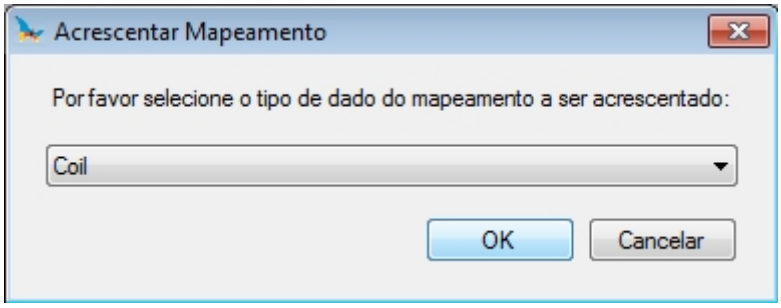


Figura 75: Tipo de Dado MODBUS

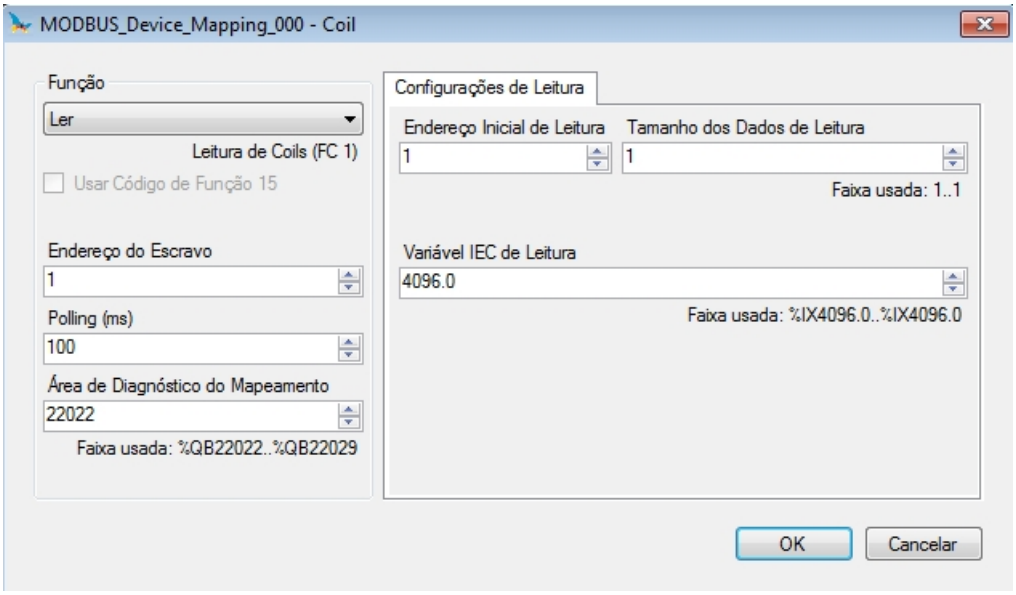


Figura 76: Função MODBUS

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

Configuração	Descrição	Padrão	Opções
Função	Tipo de função MODBUS	Ler	Ler Escrever Ler/Escrever Máscara de Escrita
Endereço do Escravo	Endereço do escravo MODBUS.	1	0 a 255
Polling (ms)	Período de comunicação (ms)	100	0 a 3600000
Área de Diagnóstico do Mapeamento	Endereço inicial dos diagnósticos da relação MODBUS (%Q)	-	0 a 2147483640
Endereço Inicial de Leitura	Endereço inicial dos dados de leitura MODBUS	1	1 a 65536
Tamanho dos Dados de Leitura	Número de dados de leitura MODBUS	-	Depende da função utilizada

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

Tabela 118: Mapeamentos do Dispositivo

Notas:

Tabela de Mapeamentos do Dispositivo: O número de configurações e os valores descritos na coluna Opções podem variar de acordo com o tipo de dado e função MODBUS.

Endereço do Escravo: Normalmente, o endereço 0 é utilizado quando o servidor é um Gateway MODBUS TCP ou MODBUS RTU via TCP, e o mesmo transmite a requisição para todos os dispositivos da rede. Quando o endereço 0 é utilizado, o cliente não espera por uma resposta, seu uso serve somente para comandos de escrita. Além disso, de acordo com a Norma MODBUS, a faixa de endereços válidos para escravos é de 0 a 247, sendo os endereços 248 a 255 reservados.

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

Área de Diagnósticos do Mapeamento: Esse campo é limitado pelo tamanho da memória de variáveis de saídas endereçáveis (%Q) de cada UCP, a qual pode ser consultada na seção Memória. Os diagnósticos da relação MODBUS configurada estão descritos na Tabela 92.

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

ATENÇÃO

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

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

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

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

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

ATENÇÃO

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

5.5.8.3. Disparo de Relações MODBUS Cliente de Forma Acíclica

Para disparar relações MODBUS Cliente de forma acíclica, sugere-se o seguinte método, que pode ser implementado de maneira simples no programa da aplicação do usuário:

- Definir tempo máximo de polling para as relações;
- Manter a relação normalmente desabilitada;
- Habilitar a relação no momento em que se deseja executá-la;
- Esperar pela confirmação de término da execução da relação, e neste momento desabilitá-la novamente.

5.5.9. MODBUS Ethernet Servidor

Este protocolo está disponível para as UCPs da Série Nexto nos seus canais Ethernet. Ao selecionar esta opção no MasterTool IEC XE, a UCP passa a ser servidor da comunicação MODBUS, permitindo a conexão com dispositivos cliente MODBUS. Este protocolo somente está disponível quando a UCP estiver em modo de execução (Modo *Run*).

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

O procedimento para inserir uma instância de protocolo é encontrado com detalhes no Manual de Utilização do MasterTool IEC XE – MU299048.

5.5.9.1. Configuração do Protocolo MODBUS Ethernet Servidor por Mapeamento Simbólico

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

- Configurar os parâmetros gerais do protocolo MODBUS servidor, como: porta TCP, seleção de protocolo, filtros de IP para Escrita e para Leitura (disponível no botão de configuração de filtros) e tempos de comunicação (disponível no botão de configurações avançadas do Servidor).
- Adicionar e configurar os mapeamentos MODBUS, especificando o nome da variável, tipo de dados, endereço inicial do dado e tamanho do dado.

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

5.5.9.1.1. Parâmetros Gerais do Protocolo MODBUS Servidor – Configuração por Mapeamento Simbólico

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

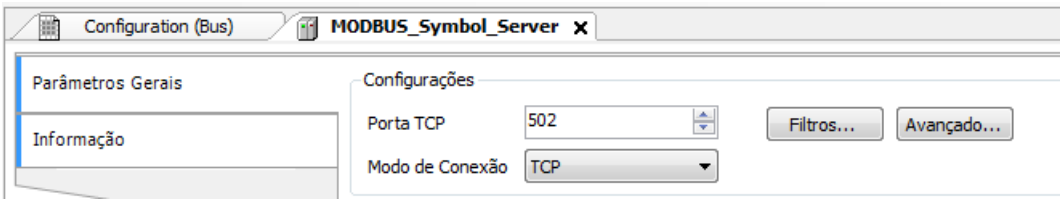


Figura 77: Tela de Configuração Parâmetros Gerais MODBUS Servidor

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

Tabela 119: 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 120: Filtros de IP

Nota:

Filtros: Os filtros são utilizados para estabelecer um intervalo de endereços IP que têm acesso de escrita ou leitura nas relações MODBUS, sendo individualmente configurados. O critério de permissão é realizado através de uma operação lógica AND entre o Filtro de Máscara para Escrita e o endereço IP do cliente. Caso o resultado seja igual ao Filtro de Endereço IP para Escrita, o cliente tem direito de escrita. Por exemplo, se o Filtro de Endereço IP para Escrita = 192.168.15.0 e o Filtro de Máscara para Escrita = 255.255.255.0, então somente clientes com endereço IP = 192.168.15.x terão direito de escrita. O mesmo procedimento é aplicado nos parâmetros de Filtro de Leitura para definir os direitos de leitura.

Os tempos de comunicação do protocolo MODBUS Servidor, encontrados no botão *Avançado...* da tela de configuração, estão divididos em: *Ciclo da Tarefa* e *Time-out da Inatividade da Conexão*.

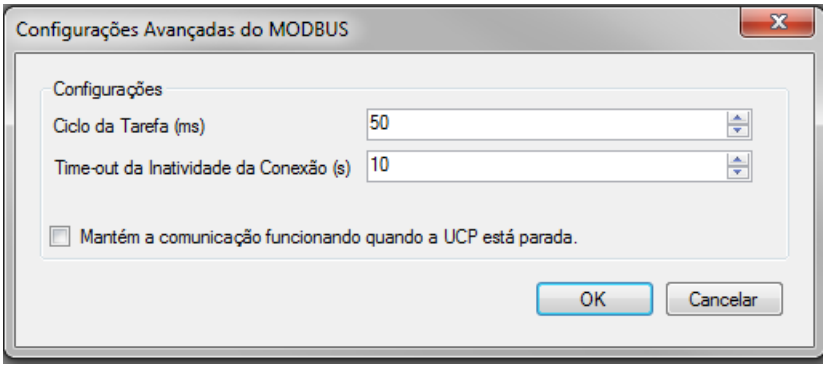


Figura 78: Janela de Configurações Avançadas do MODBUS Servidor

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

Tabela 121: Configurações Avançadas MODBUS Servidor

Notas:

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

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

5.5.9.1.2. Diagnósticos MODBUS Servidor – Configuração por Mapeamento Simbólico

Os diagnósticos e comandos do protocolo MODBUS Servidor configurado, seja por mapeamento simbólico ou por representação direta, são armazenados em variáveis do tipo *T_DIAG_MODBUS_ETH_SERVER_1* e ainda para o mapeamento por representação direta estão em 4 bytes e 8 words, as quais estão descritas na tabela abaixo:

T_DIAG_MODBUS_ETH_SERVER_1.*	Tamanho	Descrição
Bits de diagnóstico:		
tDiag.bRunning	BIT	O servidor está em execução.
tDiag.bNotRunning	BIT	O servidor não está em execução (ver bit bInterruptedByCommand).
tDiag.bInterruptedByCommand	BIT	O bit bNotRunning foi habilitado, pois o servidor foi interrompido pelo usuário através de bits de comando.
tDiag.bConfigFailure	BIT	Falha na configuração.
tDiag.bModuleFailure	BIT	Indica se há falha no módulo ou o módulo não está presente.

T_DIAG_MODBUS_ETH_SERVER_1.*	Tamanho	Descrição
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 122: 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: Diagnótico implementado apenas para MODBUS simbólico.

5.5.9.1.3. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico

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

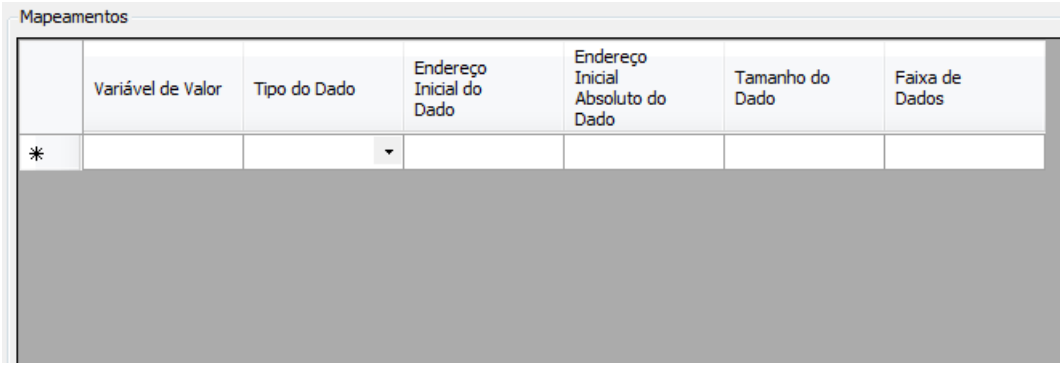


Figura 79: Tela de Mapeamentos de dados MODBUS Servidor

Configuração	Descrição	Padrão	Opções
Variável de Valor	Nome da variável simbólica	-	Nome de uma variável declarada em um programa ou GVL
Tipo do Dado	Tipo de dado MODBUS	-	Coil Input Status Holding Register Input Register

Configuração	Descrição	Padrão	Opções
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 123: Configuração dos Mapeamentos MODBUS

Notas:

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

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

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

Endereço Inicial Absoluto do Dado: Endereço inicial absoluto dos dados MODBUS conforme o seu tipo. Por exemplo, o Holding Register com endereço 5 possui endereço absoluto 400005. Este campo é apenas de leitura e está disponível para auxiliar na configuração do Cliente/Mestre MODBUS que irá comunicar-se com este dispositivo. Os valores dependem do endereço base (offset) de cada tipo de dado MODBUS e do endereço permitido para cada tipo de dado.

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

Faixa de Dados: É um campo somente de leitura e informa a faixa de endereços que está sendo usada por esse mapeamento. Ele é formado pela soma dos campos *Endereço Inicial* e *Tamanho do Dado*. Não podem haver sobreposições de faixa com outros mapeamentos do mesmo *Tipo de Dado*.

ATENÇÃO

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

5.5.9.2. Configuração do Protocolo MODBUS Ethernet Servidor por Representação Direta (%Q)

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

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

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

5.5.9.2.1. Parâmetros Gerais do Protocolo MODBUS Servidor – Configuração por Representação Direta (%Q)

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

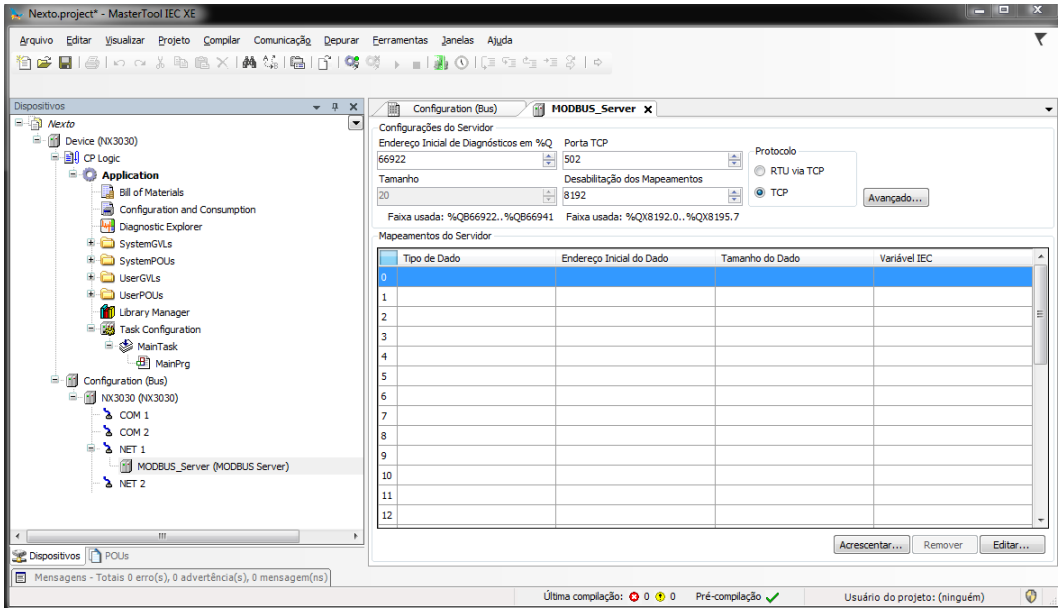


Figura 80: Tela de Configuração MODBUS Servidor

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

Configuração	Descrição	Padrão	Opções
Endereço Inicial de Diagnósticos em %Q	Endereço inicial das variáveis de diagnóstico	-	0 a 2147483628
Tamanho	Tamanho da área de diagnósticos	20	Desabilitado para edição
Porta TCP	Porta TCP	502	2 a 65534
Desabilitação dos Mapeamentos	Endereço inicial utilizado para desabilitar as relações MODBUS	-	0 a 2147483644
Protocolo	Seleção do protocolo	TCP	RTU via TCP TCP

Tabela 124: Configurações das Variáveis de Representação Direta e Endereçamento

Notas:

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

Porta TCP: Caso sejam adicionadas várias instâncias do protocolo em uma única interface Ethernet, diferentes portas TCP devem ser selecionadas para cada instância. Algumas portas TCP, entre as possibilidades mencionadas acima, são reservadas e, portanto, não podem ser utilizadas. Ver tabela Portas TCP/UDP reservadas.

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

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

Os tempos de comunicação do protocolo MODBUS Servidor, encontrados no botão *Avançado...* da tela de configuração, estão divididos em *Ciclo da Tarefa (ms)* e *Time-out da Inatividade da Conexão (s)*, maiores detalhes estão descritos na seção Parâmetros Gerais do Protocolo MODBUS Servidor – Configuração por Mapeamento Simbólico.

Os diagnósticos e comandos do protocolo MODBUS estão descritos na Tabela 122.

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

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

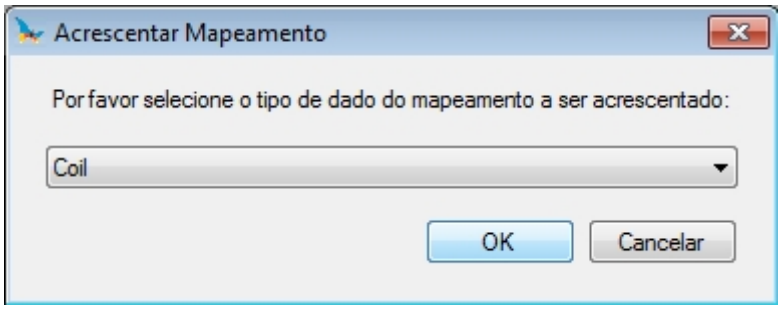


Figura 81: Tipo de Dado MODBUS

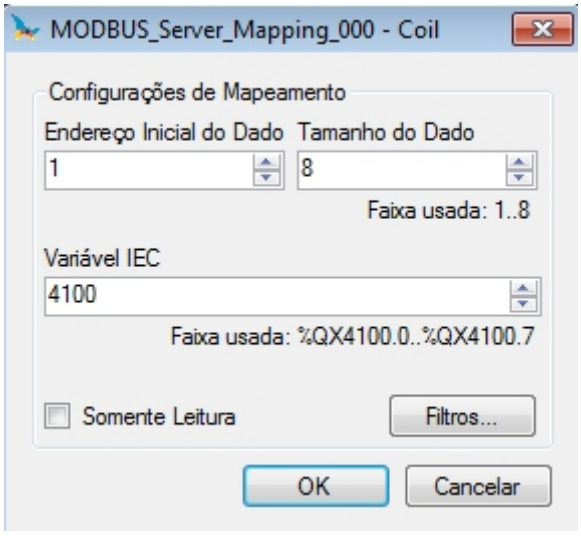


Figura 82: Função MODBUS Servidor

Configuração	Descrição	Padrão	Opções
Tipo de Dado	Tipo de dado MODBUS	Coil	Coil (1 bit) Holding Register (16 bits) Input Status (1 bit) Input Register (16 bits)
Endereço Inicial do Dado	Endereço inicial dos dados MODBUS	1	1 a 65536
Tamanho do Dado	Tamanho do dado MODBUS	8	1 a 65536 (Holding Register e Input Register) 8 a 65536 (Coil e Input Status)
Variável IEC	Endereço inicial das variáveis (%Q)	-	0 a 2147483647
Somente Leitura	Somente permite a leitura	Desabilitada	Habilitada ou Desabilitada

Tabela 125: Mapeamentos do Servidor

Notas:

Opções: Os valores descritos na coluna *Opções* podem variar de acordo com o tipo de dado MODBUS configurado.

Tamanho do Dado: O valor de *Tamanho do Dado* define a quantidade máxima de dados que uma relação MODBUS poderá acessar, a partir do endereço inicial. Sendo assim, para ler uma faixa de endereços contínua, é necessário que todos os endereços estejam declarados em uma única relação. Este campo varia de acordo com o tipo de dado MODBUS configurado, ou seja, quando selecionado tipo *Coil* ou *Input Status*, o campo *Tamanho do Dado* deve ser um número múltiplo de oito. Também deve-se dar atenção para que o valor máximo não seja superior ao tamanho da memória de saídas endereçáveis e não sejam atribuídos os mesmos valores já utilizados durante a aplicação.

ATENÇÃO

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

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

Somente Leitura: Quando habilitada, somente permite que o mestre da comunicação leia os dados das variáveis, não permitindo a escrita. Opção válida somente para as funções de escrita.

Padrão: O padrão de fábrica não pode ser definido para o campo *Variável IEC*, pois a criação de uma instância do protocolo pode ser realizada em qualquer momento no desenvolvimento da aplicação, fazendo com que o próprio software MasterTool IEC XE aloque um valor, da faixa de variáveis de saída de representação direta (%Q), ainda não utilizado. O padrão de fábrica não pode ser definido para o campo *Tamanho do Dado*, pois ele vai variar de acordo com o tipo de dado MODBUS selecionado.

As configurações presentes no botão *Filtros...*, descritas na tabela abaixo, são relativas aos filtros de comunicação TCP:

Configuração	Descrição	Padrão	Opções
Filtro de Endereço IP para Escrita	Especifica um intervalo de IPs com acesso de escrita nas variáveis declaradas na relação MODBUS.	0.0.0.0	0.0.0.0 a 255.255.255.255
Filtro de Máscara para Escrita	Especifica a máscara de subrede em conjunto com o parâmetro Filtro de Endereço IP para Escrita.	0.0.0.0	0.0.0.0 a 255.255.255.255
Filtro de Endereço IP para Leitura	Especifica um intervalo de IPs com acesso de leitura nas variáveis declaradas na relação MODBUS.	0.0.0.0	0.0.0.0 a 255.255.255.255
Filtro de Máscara para Leitura	Especifica a máscara de subrede em conjunto com o parâmetro Filtro de Endereço IP para Leitura.	0.0.0.0	0.0.0.0 a 255.255.255.255

Tabela 126: Filtros de IP

Nota:

Filtros: Os filtros são utilizados para estabelecer um intervalo de endereços IP que têm acesso de escrita ou leitura nas relações MODBUS, sendo individualmente configurados. O critério de permissão é realizado através de uma operação lógica AND entre o Filtro de Máscara para Escrita e o endereço IP do cliente. Caso o resultado seja igual ao Filtro de Endereço IP para Escrita, o cliente tem direito de escrita. Por exemplo, se o Filtro de Endereço IP para Escrita = 192.168.15.0 e o Filtro de Máscara para Escrita = 255.255.255.0, então somente clientes com endereço IP = 192.168.15.x terão direito de escrita. O mesmo procedimento é aplicado nos parâmetros de Filtro de Leitura para definir os direitos de leitura.

Nas relações definidas anteriormente, o tamanho máximo de dados MODBUS pode ser 65536 (máximo valor configurado no campo *Tamanho do Dado*). Porém, a requisição que chega ao MODBUS Ethernet Servidor deverá endereçar um subconjunto desse mapeamento e esse grupo deve ter, no máximo, o tamanho de dados que depende do código da função, os quais

estão definidos abaixo:

- Leitura de Coils (FC 1): 2000
- Leitura de Input Status (FC 2): 2000
- Leitura de Holding Registers (FC 3): 125
- Leitura de Input Registers (FC 4): 125
- Escrita de Um Coil (FC 5): 1
- Escrita de Um Holding Register (FC 6): 1
- Forçamento de Múltiplos Coils (FC 15): 1968
- Escrita de Holding Registers (FC 16): 123
- Máscara de Escrita do Register (FC 22): 1
- Leitura/Escrita de Holding Registers (FC 23):
  - Leitura: 121
  - Escrita: 121

ATENÇÃO

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

5.5.10. OPC DA Servidor

Para comunicar com as UCPs da Série Nexto é possível utilizar a tecnologia OPC DA (*Open Platform Communications Data Access*). Esta plataforma de comunicação aberta foi desenvolvida para ser o padrão utilizado nas comunicações industriais. Baseado na arquitetura cliente/servidor, oferece inúmeras vantagens no desenvolvimento de projeto e facilidades na comunicação com os sistemas de automação.

Uma analogia muito comum, utilizada para descrever a tecnologia OPC DA, é a de uma impressora. Quando corretamente conectada, o computador precisa de um driver para ter a interface com o equipamento. Muito similar, o OPC DA auxilia na interface entre o sistema de supervisão com os dados de campo no CP.

Quando se trata do desenvolvimento de projetos, configurar a comunicação e trocar informações entre os sistemas é extremamente simples utilizando tecnologia OPC DA. Utilizando outros drivers, baseados em endereços, é necessário criar tabelas para relacionar as tags do sistema de supervisão e as variáveis do controlador programável. Quando as áreas de dados são alteradas, no decorrer do desenvolvimento do projeto, é necessário refazer os mapeamentos e novas tabelas com as relações entre as informações do CP com o sistema de Controle Supervisório e Aquisição de Dados (SCADA).

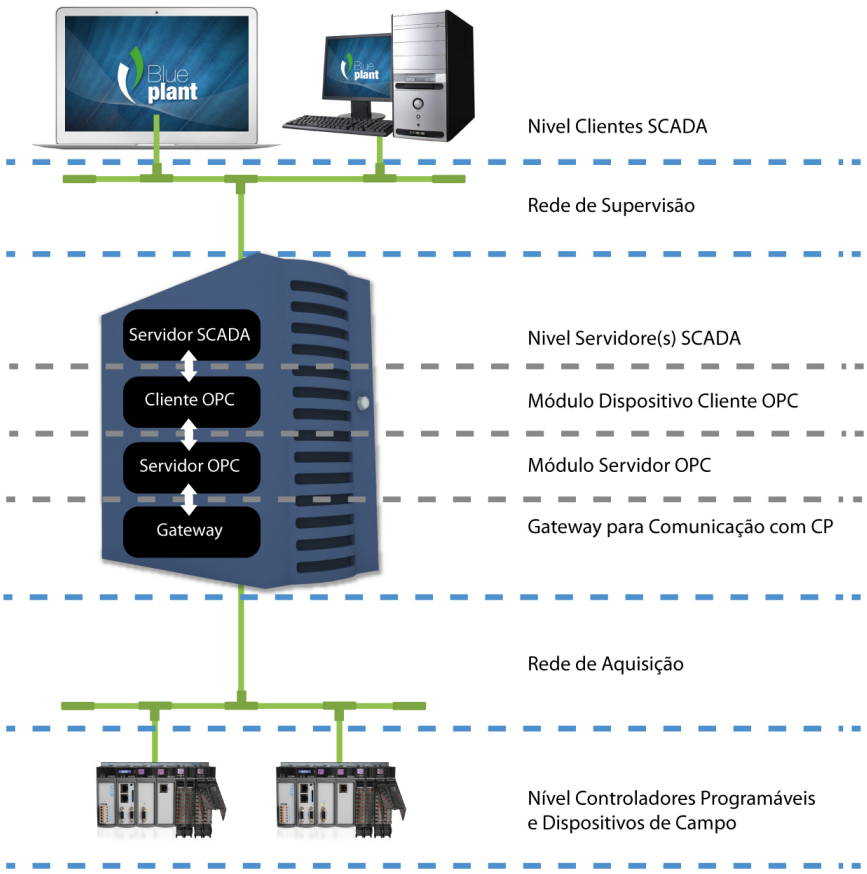


Figura 83: Arquitetura OPC DA

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

Papel	Descrição
Nível Controladores Programáveis e Dispositivos de Campo	Os dispositivos de campo e os CPs são os dispositivos nos quais as informações do estado de operações e controle da planta são armazenados. Os sistemas SCADA acessam as informações nestes dispositivos e armazenam nos servidores SCADA para consulta pelos Clientes SCADA durante a operação da planta.
Rede de Aquisição	A rede de aquisição é a rede na qual trafegam as mensagens para solicitar os dados que são coletados dos dispositivos de campo.
Gateway para Comunicação com CP	Para a comunicação entre o Servidor OPC DA e os CPs da Série Nexto é utilizado um gateway que permite esta comunicação. Sempre é necessário existir um gateway na mesma subrede do CP como descrito no capítulo Configurações de Comunicação, no Manual de Utilização MasterTool IEC XE – MU299048.
Módulo Servidor OPC	O Servidor OPC DA é um Módulo responsável por receber as requisições OPC DA e traduzi-las para a comunicação com os dispositivos de campo.

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 127: Descrição dos Papéis em uma Arquitetura com Servidor OPC DA

A relação entre as tags dos sistemas de supervisão e os dados do processo nas variáveis do controlador é totalmente transparente. Isso significa que se as áreas de dados são alteradas no decorrer do desenvolvimento do projeto, não há a necessidade de refazer relações entre as informações do CP com o SCADA. Basta utilizar a nova variável disponibilizada pelo CP nos sistemas que requisitam esse dado.

O uso do OPC DA oferece maior produtividade e conectividade com os sistemas SCADA. Contribui na redução do tempo de desenvolvimento de aplicações e nos custos com manutenção. Possibilita, ainda, inserção de novos dados na comunicação de forma simplificada com maior flexibilidade e interoperabilidade entre os sistemas de automação por ser um padrão aberto.

O Servidor OPC DA é instalado juntamente com a instalação do MasterTool IEC XE e sua configuração é realizada dentro da ferramenta. Vale salientar que o OPC DA está disponível somente nas interfaces Ethernet locais das UCPs Nexto. Os módulos de expansão Ethernet não suportam essa funcionalidade.

5.5.10.1. Criando um Projeto para Comunicação OPC DA

Diferente das comunicações com drivers como MODBUS e PROFIBUS DP, para configurar a comunicação OPC DA basta configurar o nó corretamente e indicar quais as variáveis que serão utilizadas na comunicação. Existem duas formas de indicar quais as variáveis de projeto estarão disponíveis no Servidor OPC DA. Em ambos os casos é necessário adicionar o objeto *Symbol Configuration* à aplicação, caso este não esteja presente. Para adicioná-lo basta clicar com o botão direito do mouse sobre o objeto *Application* e selecionar a opção.

ATENÇÃO

As variáveis exibidas no objeto *IoConfig_Globals*, *IoConfig_Application_Mappings* e *IoConfig_Global_Mappings* são utilizadas internamente para controle de E/S e não devem ser utilizadas pelo usuário.

ATENÇÃO

Além das variáveis declaradas nas 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.

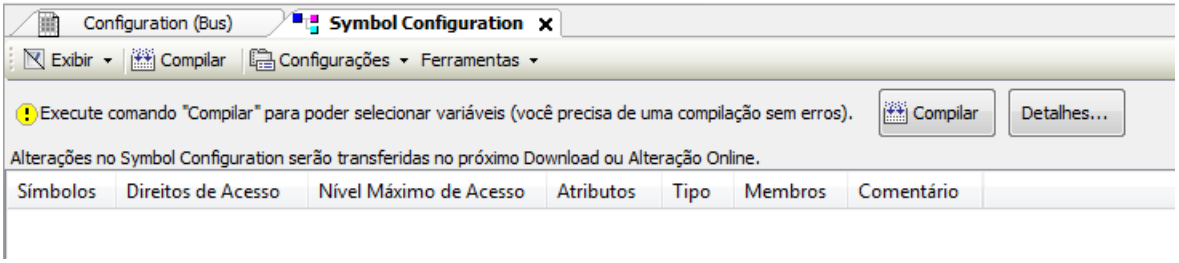


Figura 84: Objeto Symbol Configuration

A tabela abaixo apresenta a descrição dos campos da tela de configurações dos símbolos no objeto *Symbol Configuration*.

Campo	Descrição
Símbolos	Identificador da variável que será disponibilizada para o Servidor OPC DA.
Direitos de Acesso	Indica qual o nível de acesso possível no símbolo declarado. Quando não se utiliza esta coluna, a mesma fica vazia e o nível de acesso é máximo. Caso contrário o nível de acesso pode ser modificado clicando sobre o campo. As opções possíveis são as seguintes: Somente leitura Somente escrita Leitura e escrita
Nível Máximo de Acesso	Indica o máximo nível de acesso que é possível atribuir à variável. Os símbolos que representam têm o mesmo significado do campo Direitos de Acesso. Não é possível alterar e é indicado pela presença ou não do *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 128: Descrição dos Campos da Tela do Objeto Symbol Configuration

Ao executar uma alteração nas configurações do projeto, como adicionar ou remover variáveis, se faz necessário executar o comando *Compilar* para atualizar a lista de variáveis. Este comando deve ser executado até que a mensagem presente na Figura 84 desapareça. Após isso todas as variáveis disponíveis no projeto, sejam declaradas em POUs, GVLs e diagnósticos, serão exibidas e podem ser selecionadas. As variáveis selecionadas estarão disponíveis no Servidor OPC DA para acesso pelos Clientes.

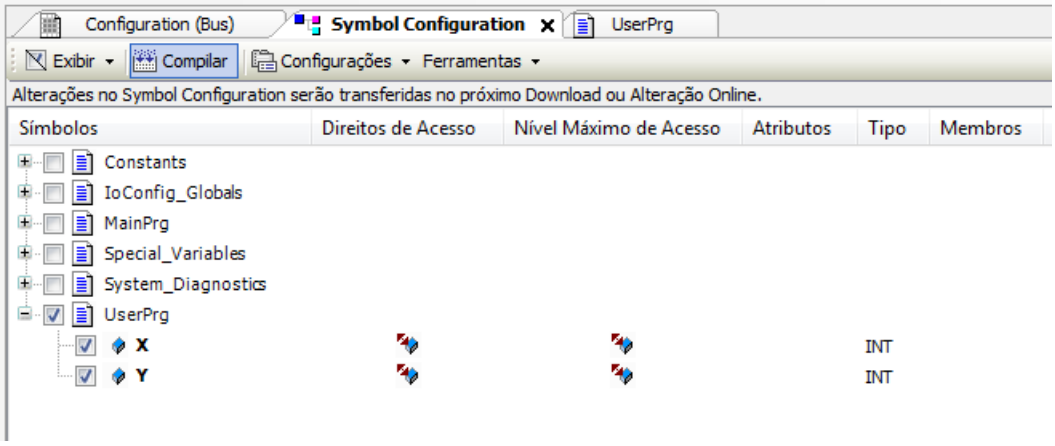


Figura 85: Selecionando Variáveis na Symbol Configuration

Após este procedimento o projeto pode ser carregado em um CP e as variáveis selecionadas estarão disponíveis para comunicação com o Servidor OPC DA. Se a tela do Objeto Symbol Configuration estiver aberta e alguma das variáveis, 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.

5.5.10.2. Configurando um CP no Servidor OPC DA

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

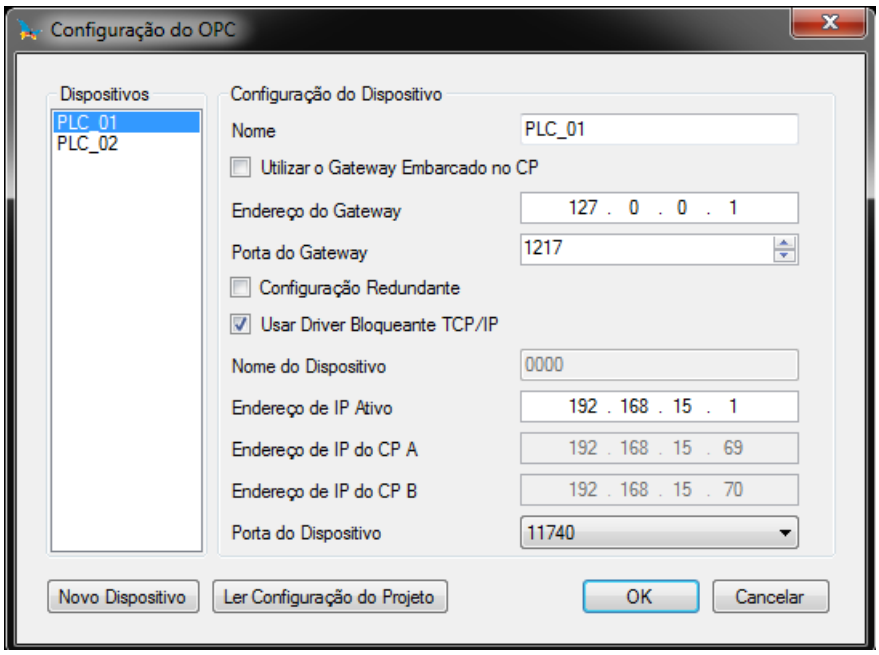


Figura 86: Configuração do OPC DA Servidor

A *Configuração do Gateway* é a mesma configurada no Gateway utilizado para comunicação entre o MasterTool IEC XE e o CP e descrita em Configurações de Comunicação, presente no Manual de Utilização MasterTool IEC XE – MU299048. Se a configuração utilizada for *localhost* o *Endereço do Gateway* deve ser preenchido com 127.0.0.1. Esta configuração é necessária, pois o Servidor OPC utiliza o mesmo gateway de comunicação e o mesmo protocolo utilizados na comunicação entre CP e MasterTool IEC XE.

Existe a opção *Utilizar o Gateway Embarcado no CP* que pode ser selecionada quando se deseja utilizar o Gateway que fica no próprio CP. Esta opção pode ser empregada para otimizar a comunicação, pois ela evita o excesso de tráfego através de uma determinada estação, quando mais de uma estação, com Cliente OPC DA, esteja conectada ao mesmo CP.

Para a configuração do CP, são possíveis dois tipos de configuração, conforme a seleção do checkbox *Usar Driver Bloqueante TCP/IP*. Quando a opção não está selecionada o nome do CP deve ser colocado no campo *Nome do Dispositivo*. Este é o nome exibido pelo CP selecionado como ativo na tela de *Configurações de Comunicação*.

A outra opção é usar o *Endereço de IP* das Interfaces Ethernet. O mesmo endereço configurado nas telas de configuração deve ser colocado neste campo. Além disso, quando for utilizado este método deve ser colocado o número da porta 11740. A confirmação irá salvar as configurações do Servidor OPC DA.

Configuração	Descrição	Padrão	Opções
Nome	Descrição do CP dentro do arquivo de configuração do Servidor OPC DA. Este campo pode ter qualquer nome, mas para organização recomenda-se utilizar o nome do projeto carregado no CP.	'PLC01'	O campo é uma STRING e podem ser colocados caracteres alfanuméricos (letras e números) e o caractere "_". Não é permitido iniciar a STRING com números ou "_". Permite até 49 caracteres.
Endereço do Gateway	Endereço IP do computador onde está instalado o Servidor OPC DA para os casos onde todos os CPs estejam na mesma subrede. Caso exista algum CP em outra subrede deve ser especificado o Gateway utilizado nesta subrede.	127.0.0.1	0.0.0.0 a 255.255.255.255
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

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

Tabela 129: Parâmetros de Configuração de Cada CP para o Servidor OPC DA

Quando um novo CP precisar ser configurado no Servidor OPC DA basta pressionar o botão *Novo Dispositivo* que a configuração será criada. Sempre que a tela de configuração for acessada será exibida uma lista com todos os CPs já configurados no Servidor OPC DA. As configurações existentes podem ser editadas selecionando o CP na lista *Dispositivos* e editando os parâmetros. As configurações de CPs que não são mais utilizadas podem ser excluídas. O número máximo de CPs configurados em um Servidor OPC DA é 16.

Caso a arquitetura de automação utilizada preveja que o servidor OPC DA deve ser executado em um computador onde não é executada a comunicação com o CP via MasterTool IEC XE, a ferramenta deve ser instalada neste computador para permitir a configuração do Servidor OPC DA da mesma maneira como é feito nas outras situações.

ATENÇÃO

Para armazenar a configuração do Servidor OPC DA, o MasterTool IEC XE precisa ser executado com direitos de administrador no Sistema Operacional. Dependendo da versão do Sistema Operacional este direito deve ser autorizado ao executar o programa. Para essa operação clique com o botão direito sobre o executável do MasterTool IEC XE e escolha a opção *Executar como administrador*.

ATENÇÃO

As configurações de um CP no Servidor OPC DA não são armazenadas no projeto criado no MasterTool IEC XE. Por esta razão podem ser realizadas com um projeto aberto ou fechado. As configurações são armazenadas em um arquivo de configuração onde o Servidor OPC DA está instalado. Quando alterar as configurações não é necessária carga de aplicação no CP, mas dependendo do Cliente OPC DA é possível que seja necessário conectar novamente ao Servidor ou carregar as configurações para que os dados sejam atualizados corretamente.

5.5.10.2.1. Importando uma Configuração do Projeto

Utilizando o botão *Ler Configuração do Projeto*, conforme a Figura 86, é possível atribuir a configuração do projeto aberto à configuração do CP que está em edição. Para que esta opção funcione corretamente deve existir um projeto aberto e deve ser definido um *Caminho Ativo* conforme descrito em *Configurações de Comunicação*, presente no Manual de Utilização MasterTool IEC XE – MU299048. Caso alguma destas condições não seja atendida será exibida uma mensagem de erro e nenhum dado será modificado.

Quando as condições anteriores são válidas, as configurações do CP recebem os parâmetros do projeto aberto. As informações de *Endereço de IP* e *Porta do Gateway* são configuradas conforme descrito em *Configurações de Comunicação* de acordo com o *Caminho Ativo*. Entretanto, as configurações de *Endereço de IP* são lidas das configurações da interface Ethernet NET1. A porta para conexão com o CP é sempre atribuída neste caso como 11740.

5.5.10.3. Variáveis de Status e Qualidade da Comunicação OPC DA

Para cada um dos CPs criados no Servidor OPC DA são geradas variáveis de status chamadas de *_CommState* e *_CommStateOK*. A variável *_CommState* indica o estado da comunicação do Servidor OPC DA com CP. Este estado pode ser interpretado pelo Cliente OPC DA conforme a tabela abaixo.

Estado	Valor	Descrição
STATE_TERMINATE	-1	Se a comunicação do Servidor OPC DA com o Cliente OPC DA for encerrada este valor será retornado. Quando houver mais de um Cliente OPC DA conectado ao mesmo tempo o retorno ocorrerá na desconexão do último Cliente conectado. Este status apesar de estar na variável não pode ser visualizado, pois só é alterado quando não existe mais conexão com um cliente.
STATE_PLC_NOT_CONNECTED	0	O CP configurado no Servidor OPC DA não está conectado. Pode acontecer caso a configuração esteja incorreta (Endereço de IP do CP e/ou Gateway errados) ou o CP não esteja disponível naquele momento.
STATE_PLC_CONNECTED	1	O CP configurado no Servidor OPC DA está conectado. Este é um estado transitório durante conexão.
STATE_NO_SYMBOLS	2	Não existem símbolos (variáveis) disponíveis no CP configurado no Servidor OPC DA. Pode acontecer na situação de não existirem símbolos ou ainda quando não houver um projeto carregado no CP.
STATE_SYMBOLS_LOADED	3	Finalizado o processo de leitura dos símbolos (variáveis) do CP configurado no Servidor OPC DA. Este é um estado transitório durante conexão.
STATE_RUNNING	4	Após leitura dos símbolos (variáveis) o Servidor OPC DA está executando a atualização periódica dos valores dos símbolos disponíveis em cada CP configurado.
STATE_DISCONNECT	5	Ocorreu desconexão com o CP configurado no servidor OPC DA.
STATE_NO_CONFIGURATION	6	Quando a configuração do OPC DA (armazenada em um arquivo OPCServer.ini) possuir uma sintaxe errada este será o status da variável. De maneira geral, este comportamento não é observado pois o MasterTool IEC XE mantém esta configuração íntegra.

Tabela 130: 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.

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

Tabela 131: Descrição do valor da Qualidade OPC DA

A Tabela 131 apresenta os valores possíveis de qualidade. O Servidor OPC DA retorna apenas valor com Qualidade *Good* e *Bad*. Um Cliente OPC DA pode manter a qualidade como *Uncertain* em caso de alguma falha na qual ele não consiga uma conexão com o Servidor. No caso de monitoração dos 8 bits de qualidade diretamente do Servidor OPC DA os campos *Substatus* e *Limite* serão nulos e a Qualidade *Good* será representada com o valor 192 e a qualidade *Bad* com o valor 0.

5.5.10.4. 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	-
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 132: Limites da Comunicação com o Servidor OPC DA

Nota:

Número máximo de variáveis comunicando com um único CP: Não há limite de configuração. A quantidade máxima possível de variáveis depende da capacidade de processamento do dispositivo.

ATENÇÃO

O número máximo de conexões simultâneas de Servidor OPC DA em um mesmo CP é compartilhado com as conexões realizadas com o MasterTool IEC XE. Ou seja, a soma de conexões com Servidor OPC DA e MasterTool IEC XE não deve ultrapassar o número máximo definido na Tabela 132.

A comunicação entre o Servidor OPC DA e o CP utiliza o mesmo protocolo utilizado para comunicação do MasterTool IEC XE com o CP. Este protocolo só está disponível para as interfaces Ethernet das UCPs da Série Nexto, não sendo possível estabelecer este tipo de comunicação com módulos de expansão Ethernet.

Quando uma comunicação é estabelecida entre o Servidor OPC DA e o CP estes dois elementos iniciam uma série de transações que visam resolver o endereço de cada variável declarada, otimizando a comunicação em regime de leitura de dados. Além disso, nesta fase também são resolvidas as classificações dos grupos de comunicação usados por alguns Clientes com o intuito de otimizar a comunicação. Este processo inicial demanda algum tempo e depende da quantidade de variáveis mapeadas e da capacidade de processamento do dispositivo.

5.5.10.5. 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 IEC XE algumas ferramentas precisam ser executadas com direitos de administrador no Sistema Operacional para o correto funcionamento do Cliente OPC DA. Dependendo da versão do Sistema Operacional este direito deve ser autorizado ao executar o programa. Para essa operação clique com o botão direito sobre o executável da ferramenta e escolha a opção *Executar como administrador*.

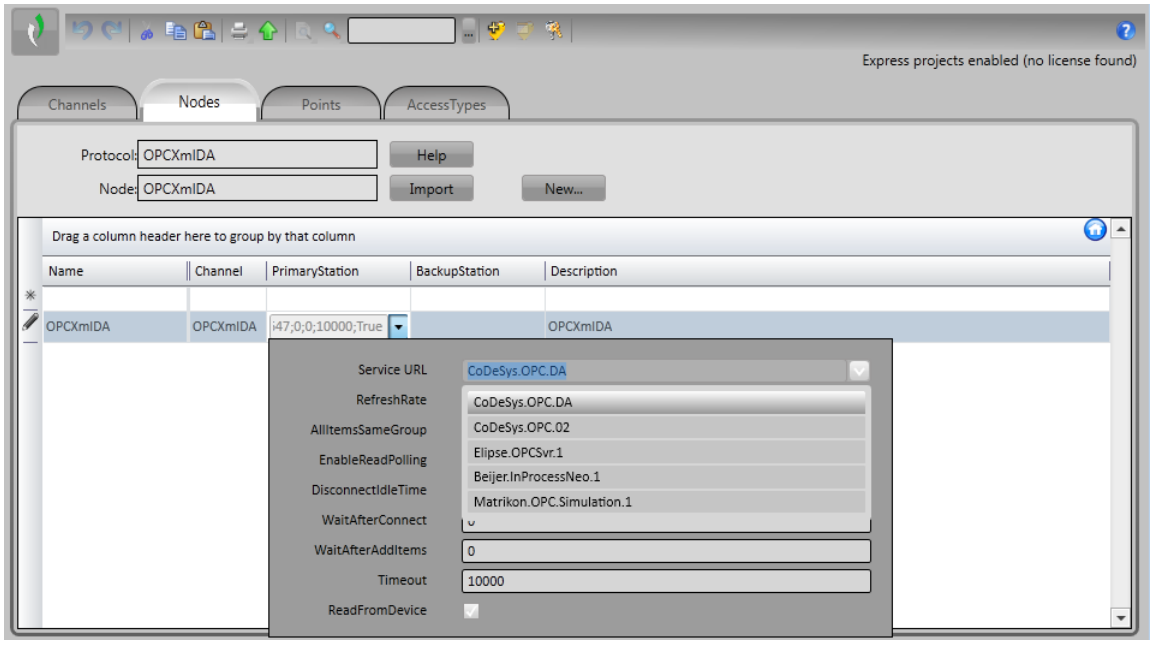


Figura 87: Selecionando Servidor OPC DA na Configuração do Cliente

Nos casos em que o servidor se encontra remotamente pode ser necessário adicionar o caminho da rede ou o endereço de IP do computador onde se encontra o servidor instalado. Nestes casos existem duas opções de configuração. A primeira delas é configurar diretamente para isso sendo necessário liberar os Serviços de COM/DCOM do Windows. Contudo, uma forma mais simples é utilizar uma ferramenta de tunneller que abstrai as configurações de COM/DCOM, além de possibilitar uma comunicação mais segura entre o Cliente e o Servidor. Para mais informações sobre este tipo de ferramenta consultar a *NAP151 - Utilização do Tunneller OPC*.

Uma vez que o Cliente se conecta no Servidor podem ser usados comandos de importação de TAGs. Estes comandos consultam informações declaradas no CP, retornando uma lista com todos os símbolos disponibilizados por este.

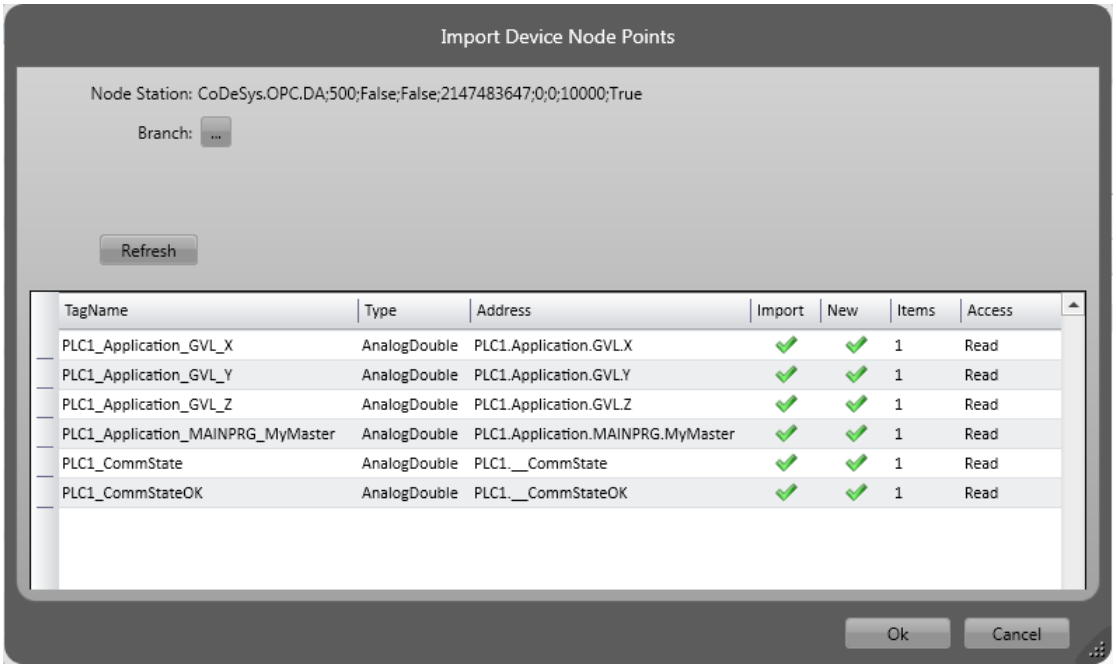


Figura 88: Lista de Símbolos Consultados pelo Cliente OPC DA

A lista de variáveis selecionadas será incluída na lista de comunicações do Cliente e podem ser utilizadas, por exemplo, em telas de um sistema SCADA.

ATENÇÃO

O modo de simulação do software MasterTool IEC XE pode ser utilizado para testes da comunicação OPC DA. As informações sobre como configurá-lo estão presentes na seção *Testando a Comunicação OPC com o Uso do Simulado*r, do Manual de Utilização MasterTool IEC XE – MU299048.

5.5.11. OPC UA Servidor

O protocolo OPC UA é uma evolução da família OPC. Independente de plataforma, foi concebido para ser o novo padrão utilizado nas comunicações industriais.

Baseado na arquitetura cliente/servidor, o protocolo OPC UA oferece inúmeras vantagens no desenvolvimento de projeto e facilidades na comunicação com os sistemas de automação.

Quando se trata do desenvolvimento de projetos, configurar a comunicação e trocar informações entre os sistemas é extremamente simples utilizando tecnologia OPC UA. Utilizando outros drivers, baseados em endereços, é necessário criar tabelas para relacionar as tags do sistema de supervisão e as variáveis do controlador programável. Quando as áreas de dados são alteradas, no decorrer do desenvolvimento do projeto, é necessário refazer os mapeamentos e novas tabelas com as relações entre as informações do CP com o sistema SCADA.

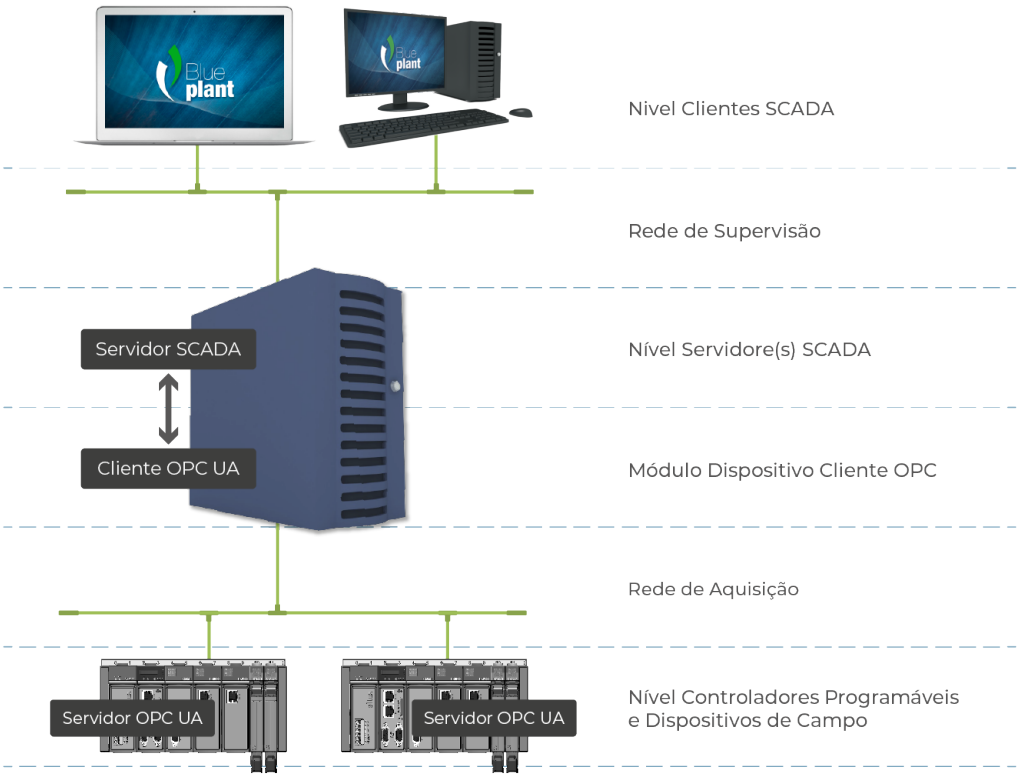


Figura 89: Arquitetura Típica OPC UA

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

Papel	Descrição
Nível Controladores Programáveis e Dispositivos de Campo	Os dispositivos de campo e os CPs são os dispositivos nos quais as informações do estado de operação e controle da planta são armazenadas. Os sistemas SCADA acessam as informações nestes dispositivos e armazenam nos Servidores SCADA para consulta pelos Clientes SCADA durante a operação da planta.
Módulo Servidor OPC UA	O Servidor OPC UA é um módulo interno dos CPs responsável por receber as requisições OPC UA e traduzi-las para a comunicação com os dispositivos de campo.
Rede de Aquisição	A rede de aquisição é a rede na qual trafegam as mensagens OPC UA para solicitar os dados que são coletados dos CPs e dispositivos de campo.
Módulo Cliente OPC UA	O módulo Cliente OPC UA, que faz parte do Servidor SCADA, é responsável por fazer requisições aos Servidores OPC UA utilizando o protocolo OPC UA. Os dados coletados por ele são armazenados na base de dados do Servidor SCADA.
Nível Servidor SCADA	O Servidor SCADA é responsável por se conectar aos diversos dispositivos de comunicação e armazenar os dados coletados destes dispositivos em uma base de dados para que possam ser consultados pelos Clientes SCADA.
Rede de Supervisão	A rede de supervisão é a rede pela qual os Clientes SCADA estão conectados aos Servidores SCADA, muitas vezes utilizando um protocolo proprietário do sistema SCADA específico. Em uma topologia na qual não se usa diversos Clientes ou que o Servidor e o Cliente estejam instalados em um mesmo equipamento, não existe este tipo de rede, e neste caso este equipamento deve utilizar diretamente o protocolo OPC UA para comunicação com o CP.
Nível Clientes SCADA	Os clientes SCADA são responsáveis por solicitar aos servidores SCADA os dados necessários para exibir em uma tela onde é executada a operação de uma planta. Através deles é possível executar leituras e escritas em dados armazenados na base de dados do Servidor SCADA.

Tabela 133: Descrição dos Papéis em uma Arquitetura com Servidor OPC UA

Ao utilizar o protocolo OPC UA, a relação entre as tags dos sistemas de supervisão e os dados do processo nas variáveis do controlador é totalmente transparente. Isso significa que se as áreas de dados são alteradas no decorrer do desenvolvimento do projeto, não há a necessidade de refazer relações entre as informações do CP com o SCADA. Basta utilizar a nova variável disponibilizada pelo CP nos sistemas que requisitam esse dado.

O uso do OPC UA oferece maior produtividade e conectividade com os sistemas SCADA. Contribui na redução do tempo de desenvolvimento de aplicações e nos custos com manutenção. Possibilita, ainda, inserção de novos dados na comunicação de forma simplificada com maior flexibilidade e interoperabilidade entre os sistemas de automação por ser um padrão aberto.

Vale salientar que o OPC UA está disponível somente nas interfaces Ethernet locais das UCPs Nexto. Os módulos de expansão Ethernet não suportam essa funcionalidade.

5.5.11.1. Criando um Projeto para Comunicação OPC UA

Os passos para criação de um projeto com OPC UA são muito similares aos passos descritos na seção Criando um Projeto para Comunicação OPC DA. Assim como no protocolo OPC DA, a configuração do protocolo OPC UA está baseada na configuração da *Symbol Configuration*. Para habilitar o OPC UA basta habilitar a opção *Suporte a característica OPC UA* na configuração, conforme ilustrado na figura abaixo.

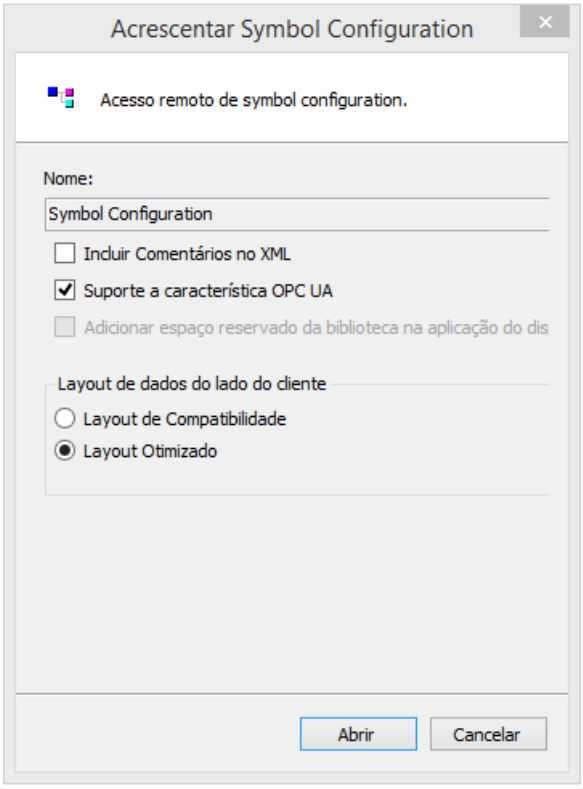


Figura 90: Objeto Symbol Configuration

ATENÇÃO

Ao ativar o suporte ao protocolo OPC UA, o suporte ao protocolo OPC DA continua habilitado. É possível habilitar as comunicações OPC UA e OPC DA ao mesmo tempo, para reportar as variáveis configuradas no objeto *Symbol Configuration* ou via atributos.

Outro caminho para acessar esta configuração, após já criado um projeto com o objeto *Symbol Configuration*, se dá acessando o menu *Configurações* da aba de configuração da *Symbol Configuration*. Basta selecionar a opção *Suporte a características OPC UA* para habilitar o suporte ao protocolo OPC UA, conforme ilustrado na figura abaixo.

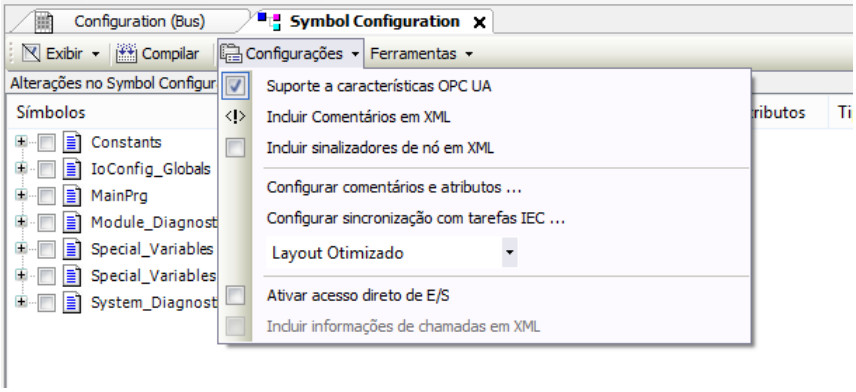


Figura 91: Habilitando OPC UA no Objeto Symbol Configuration

Após este procedimento o projeto pode ser carregado em um CP e as variáveis selecionadas estarão disponíveis para comunicação com o Servidor OPC UA.

5.5.11.2. Tipos de Variáveis Suportadas

Esta seção define os tipos de variáveis que suportam comunicação via protocolo OPC UA, quando declaradas dentro de GVLs ou POUs e selecionadas no objeto *Symbol Configuration* (ver seção anterior).

Os seguintes tipos de variáveis simples são suportados:

- BOOL
- SINT
- USINT / BYTE
- INT
- UINT / WORD
- DINT
- UDINT / DWORD
- LINT
- ULINT / LWORD
- REAL
- LREAL
- STRING
- TIME
- LTIME

É possível também utilizar tipos estruturados (STRUCTs ou Blocos Funcionais) criados a partir dos tipos simples anteriores.

Finalmente, também é possível criar arrays de tipos simples ou de tipos estruturados.

5.5.11.3. Limite de Clientes Conectados no Servidor OPC UA

O número máximo de clientes OPC UA conectados simultaneamente num CP é 8 (oito).

5.5.11.4. Limite de Variáveis de Comunicação no Servidor OPC UA

Não há limite de configuração. A quantidade máxima possível de variáveis depende da capacidade de processamento do dispositivo.

Quando uma comunicação é estabelecida entre o Servidor OPC UA e o CP, estes dois elementos iniciam uma série de transações que visam resolver o endereço de cada variável declarada, otimizando a comunicação em regime de leitura de dados. Além disso, nesta fase também são resolvidas as classificações dos grupos de comunicação usados por alguns Clientes com o intuito de otimizar a comunicação. Este processo inicial demanda algum tempo e depende da quantidade de variáveis mapeadas e da capacidade de processamento do dispositivo.

5.5.11.5. Configurações de Criptografia

Se desejado, o usuário pode configurar criptografia para a comunicação OPC UA usando o perfil *Basic256SHA256*, para obter uma conexão segura (segurança cibernética).

Para configurar a criptografia num servidor OPC UA deve-se criar um certificado para o mesmo, executando os seguintes passos no programador Mastertool:

1. Definir um caminho ativo para comunicação com o controlador (não é necessário fazer login);
2. No menu *Visualizar*, selecionar *Tela de Segurança*;
3. Clicar na aba *Devices* no lado esquerdo desta tela;
4. Clicar no ícone para executar um refresh;
5. Clicar no ícone *Device*, abaixo do qual se abrirão diversas pastas de certificados (*Own Certificates, Trusted Certificates, Untrusted Certificates, Quarantined Certificates*);
6. Clicar no ícone para gerar um certificado e selecione os seguintes parâmetros:
   - *Key length* (bit): 3072
   - *Validity period* (days): 365 (pode ser modificado se desejado)
7. Aguarde enquanto o certificado é calculado e transferido para o controlador (isso pode levar alguns minutos);
8. Reinicialize (desligue e religue) o controlador.

9. No cliente OPC UA, execute os procedimentos necessários para se conectar ao servidor OPC UA e gerar um certificado com o perfil *Basic256Sha256* (ver manual do cliente OPC UA específico para detalhes);
10. De volta ao Mastertool, clique no ícone da *Tela de Segurança* para executar um refresh;
11. Na *Tela de Segurança*, selecione a pasta "*Quarantined Certificates*"abaixo do *Device*. No painel direito deve-se observar um certificado solicitado pelo cliente OPC UA;
12. Arraste este certificado para a pasta "*Trusted Certificates*";
13. Prossiga as configurações no cliente OPC UA (ver manual do cliente OPC UA específico para detalhes).

Para remover a criptografia previamente configurada num controlador, deve-se seguir o seguinte procedimento:

1. Definir um caminho ativo para comunicação com o controlador (não é necessário fazer login);
2. No menu *Visualizar*, selecionar *Tela de Segurança*;
3. Clicar na aba *Devices* no lado esquerdo desta tela;
4. Clicar no ícone para executar um refresh;
5. Clicar no *Device*, abaixo do qual se abrem diversas pastas de certificados (*Own Certificates, Trusted Certificates, Untruted Certificates, Quarantined Certificates*);
6. Clicar na pasta "*Own Certificates*", e no painel direito selecionar o certificado (OPC UA Server);
7. Clicar no ícone para remover este certificado do projeto e do controlador;
8. Reinicialize (desligue e religue) o controlador.

5.5.11.6. Principais Parâmetros de Comunicação Ajustados em um Cliente OPC UA

Alguns parâmetros de comunicação OPC UA são configurados no cliente OPC UA, e negociados com o servidor OPC UA no momento em que a conexão entre ambos é estabelecida. As próximas subseções descrevem os principais parâmetros de comunicação OPC UA, seu significado, e cuidados para selecionar valores adequados para os mesmos.

Num cliente OPC UA é possível agrupar as variáveis de um servidor em diferentes *subscriptions*. Cada *subscription* é um conjunto de variáveis que são reportadas num único pacote de comunicação (*PublishResponse*) enviado do servidor para o cliente. A seleção das variáveis que comporão cada *subscription* é feita no cliente OPC UA.

ATENÇÃO

O agrupamento de variáveis em múltiplas *subscriptions* é interessante para otimizar a capacidade de processamento e consumo de banda de comunicação Ethernet. Tais aspectos de otimização são analisados com maior profundidade no manual do OPC UA Server MU214609, onde algumas regras para a composição de *subscriptions* são sugeridas. Esta manual também discute com maior profundidade diversos conceitos sobre o protocolo OPC UA.

Alguns dos parâmetros de comunicação descritos a seguir devem ser definidos para o servidor como um todo, outros para cada *subscription*, e outros para cada variável que compõe uma *subscription*.

5.5.11.6.1. Endpoint URL

Este parâmetro define o endereço IP e porta TCP do servidor, por exemplo:

opc.tcp://192.168.17.2:4840

Neste exemplo, o endereço IP do controlador é 192.168.17.2.

A porta TCP sempre deve ser 4840.

5.5.11.6.2. Publishing Interval (ms) e Sampling Interval (ms)

O parâmetro *Publishing Interval* (unidade: milissegundos) deve ser definido para cada *subscription*.

O parâmetro *Sampling Interval* deve ser definido para cada variável (unidade: milissegundos). Entretanto, em muitos clientes OPC UA o parâmetro *Sampling Interval* pode ser definido para uma *subscription*, sendo igual para todas as variáveis agrupadas na *subscription*.

Somente as variáveis de uma *subscription* cujos valores se modificaram são reportadas para o cliente através de um pacote de comunicação *PublishResponse*. O parâmetro *Publishing Interval* define o intervalo mínimo entre pacotes *PublishResponse* consecutivos da mesma *subscription*, com o objetivo de limitar o consumo de processamento e de banda de comunicação Ethernet.

Para descobrir quais variáveis da subscription se modificaram e devem ser reportadas para o cliente no próximo pacote *PublishResponse*, o servidor deve executar comparações, e tais comparações (*samplings*) são executadas pelo mesmo com o intervalo *Sampling Interva*l. Recomenda-se que o valor do *Sampling Interval* varie entre 50% e 100% do valor do *Publishing Interval*, pois existe um consumo de processamento relativamente alto associado ao processo de comparação executado em cada *Sampling Interval*.

Pode-se dizer que a soma entre o *Publishing Interval* e o *Sampling Interval* é o retardo máximo entre a mudança de um valor no servidor e a transmissão do pacote *PublishResponse* que reporta esta mudança. A metade desta soma é o retardo médio entre a mudança de um valor no servidor e a transmissão do pacote *PublishResponse* que reporta esta mudança.

5.5.11.6.3. Lifetime Count e Keep-Alive Count

Estes dois parâmetros devem ser configurados para cada *subscription*.

O objetivo destes dois parâmetros é criar um mecanismo de desativação de uma *subscription* por iniciativa do servidor, caso não receba pacotes de comunicação *PublishRequest* do cliente por muito tempo para esta *subscription*. Pacotes *PublishRequest* devem ser recebidos pelo servidor para que o mesmo possa transmitir pacotes *PublishResponse* contendo as variáveis da *subscription* que mudaram seus valores.

Caso o servidor não receba pacotes *PublishRequest* por um tempo superior a *Lifetime Count* multiplicado por *Publishing Interval*, o servidor desativa a *subscription*, que deverá ser criada novamente pelo cliente no futuro se assim o desejar.

Em situações onde as variáveis de uma *subscription* não se modificam, poderia passar um longo tempo sem a transmissão de *PublishResponses* e consequentemente de *PublishRequests* que o sucedem, provocando uma desativação indesejada da *subscription*. Para evitar que isso aconteça, foi criado o parâmetro *Keep-Alive Count*. Caso não haja modificações de dados na *subscription* por um tempo igual a *Keep-Alive Count* multiplicado por *Publishing Interval*, o servidor enviará um pacote *PublishResponse* pequeno e vazio indicando que nenhuma variável mudou. Este *PublishResponse* vazio autorizará que o cliente mande imediatamente o próximo *PublishRequest*.

O valor de *Keep-Alive Count* deve ser menor do que o valor de *Lifetime Count* para evitar uma desativação indesejada da *subscription*. Sugere-se que *LifeTime Count* seja no mínimo 3 vezes maior que *Keep-Alive Count*.

5.5.11.6.4. Queue Size e Discard Oldest

Estes parâmetros devem ser mantidos com os seguintes valores fixos, que normalmente são os valores padrão nos clientes:

- Queue Size: 1
- Discard Oldest: enable

De acordo com a norma OPC UA, é possível definir estes parâmetros para cada variável. No entanto, muitos clientes permitem definir valores comuns para todas as variáveis configuradas numa *subscription*.

Queue Size deve ser mantido com o valor 1 pois não existe suporte a eventos nesta implementação do servidor OPC UA, e portanto é desnecessário definir uma fila. Aumentar o valor de *Queue Size* pode implicar em aumento na banda de comunicação e processamento da CPU, e isso deve ser evitado.

Discard Oldest deve ser mantido com o valor *enable*, para que o pacote *PublishResponse* sempre reporte a mudança de valor mais recente detectada para cada variável.

5.5.11.6.5. Filter Type e Deadband Type

Estes parâmetros devem ser mantidos com os seguintes valores fixos, que normalmente são os valores padrão nos clientes:

- *Filter Type: DataChangeFilter*
- *Deadband Type: none*

De acordo com a norma OPC UA, é possível definir estes parâmetros para cada variável. No entanto, muitos clientes permitem definir valores comuns para todas as variáveis configuradas numa *subscription*.

O parâmetro *Filter Type* deve valer *DataChangeFilter*, indicando que mudanças de valores nas variáveis devem provocar sua transmissão num pacote *PublishResponse*.

Deadband Type deve ser mantido em *"none"* porque não existe implementação de *deadbands* para variáveis analógicas. Desta forma, qualquer alteração da variável analógica, por mínima que seja, provoca sua transmissão num pacote *PublishResponse*.

Para reduzir consumo de processamento e banda de comunicação Ethernet, o usuário poderá implantar *deadbands* por sua conta da seguinte forma:

- Não incluir a variável analógica numa *subscription*;
- Ao invés disso, incluir numa *subscription* uma variável auxiliar vinculada à variável analógica;
- Copiar a variável analógica para a variável auxiliar somente quando o *deadband* gerenciado pelo usuário for extrapolado.

5.5.11.6.6. *PublishingEnabled, MaxNotificationsPerPublish e Priority*

Sugere-se que os seguintes parâmetros sejam mantidos com os seguintes valores, que normalmente são os valores padrão nos clientes:

- *PublishingEnabled: true*
- *MaxNotificationsPerPublish: 0*
- *Priority: 0*

Estes parâmetros devem ser configurados para cada *subscription*.

PublishingEnable deve valer *"true"* para que as variáveis da *subscription* sejam reportadas em caso de mudança de valor.

MaxNotificationsPerPublish indica quantas das variáveis que mudaram de valor podem ser incluídas num mesmo pacote *PublishResponse*. O valor especial *"0"* indica que não existe um limite para isso, e recomenda-se utilizar este valor para que todas as variáveis que mudaram sejam reportadas num mesmo pacote *PublishResponse*.

Priority indica a prioridade relativa desta *subscription* em relação a outras. Caso em determinado momento o servidor deva enviar múltiplos pacotes *PublishResponse* de *subscriptions* diferentes, priorizará aquele com o maior valor de priority. Se todas as *subscriptions* tiverem a mesma prioridade, os pacotes *PublishResponse* serão transmitidos numa sequência fixa.

5.5.11.7. Acessando Dados Através de um Cliente OPC UA

Após a configuração do Servidor OPC UA os dados disponíveis em todos os CPs podem ser acessados via um Cliente OPC UA. Na configuração do Cliente OPC UA deve ser selecionado o endereço do Servidor OPC UA correto. Neste caso o endereço *opc.tcp://endereço-ip-do-dispositivo:4840*. A figura abaixo exibe a seleção do servidor no driver cliente do software SCADA BluePlant.

ATENÇÃO

Da mesma forma que o MasterTool IEC XE, algumas ferramentas precisam ser executadas com direitos de administrador no Sistema Operacional para o correto funcionamento do Cliente OPC UA. Dependendo da versão do Sistema Operacional este direito deve ser autorizado ao executar o programa. Para essa operação clique com o botão direito sobre o executável da ferramenta e escolha a opção *Executar como administrador*.

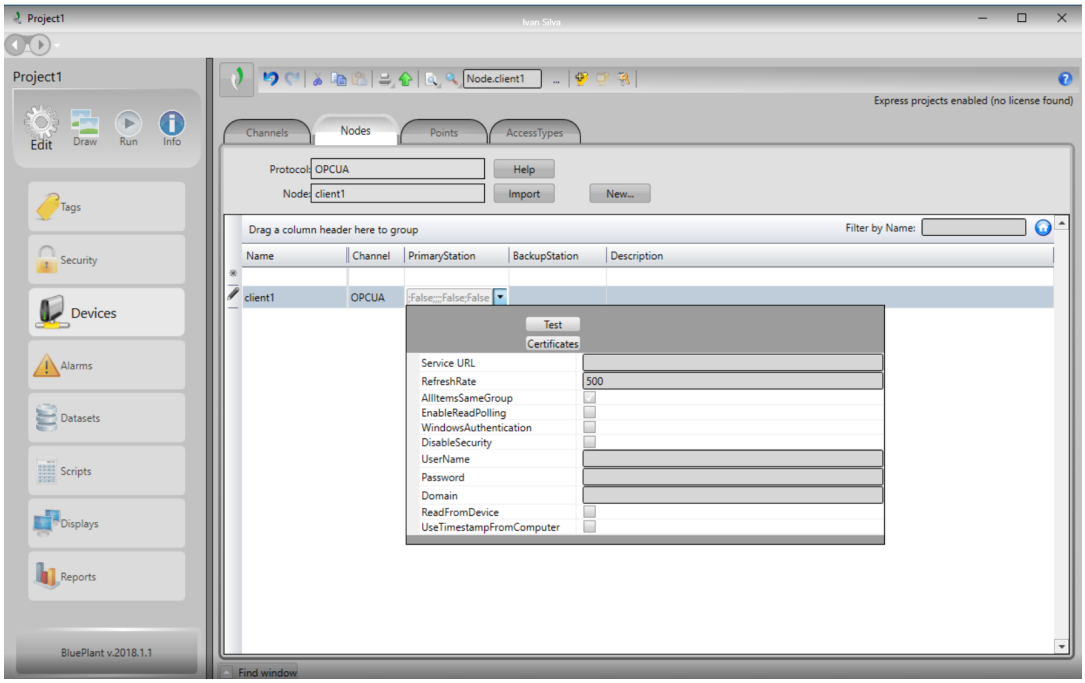


Figura 92: Selecionando Servidor OPC UA na Configuração do Cliente

Uma vez que o Cliente se conecta no Servidor podem ser usados comandos de importação de TAGs. Estes comandos consultam informações declaradas no CP, retornando uma lista com todos os símbolos disponibilizados por este.

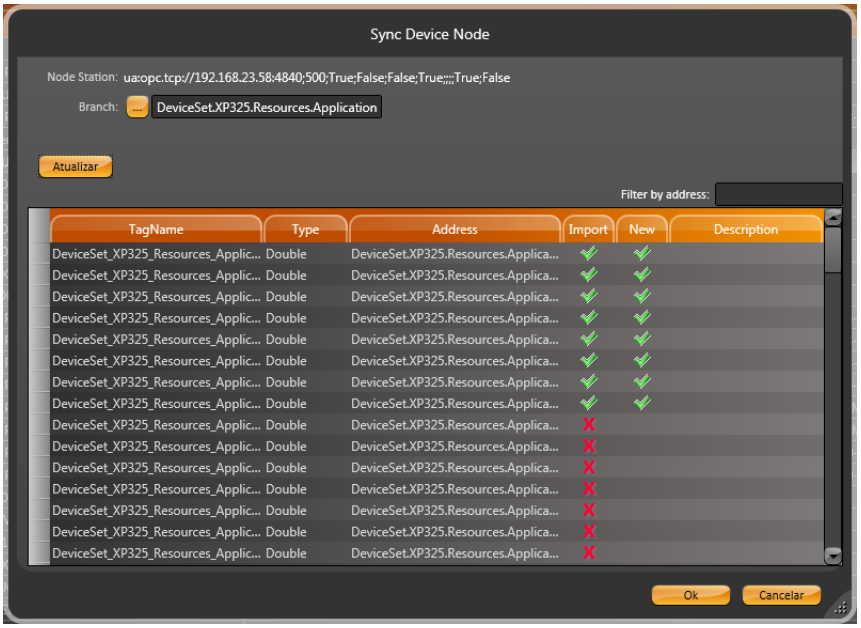


Figura 93: Lista de Símbolos Consultados pelo Cliente OPC UA

A lista de variáveis selecionadas será incluída na lista de comunicações do Cliente e podem ser utilizadas, por exemplo, em telas de um sistema SCADA.

5.5.12. Mestre EtherCAT

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

5.5.12.1. Instalando e Inserindo Dispositivos EtherCAT

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

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

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

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

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

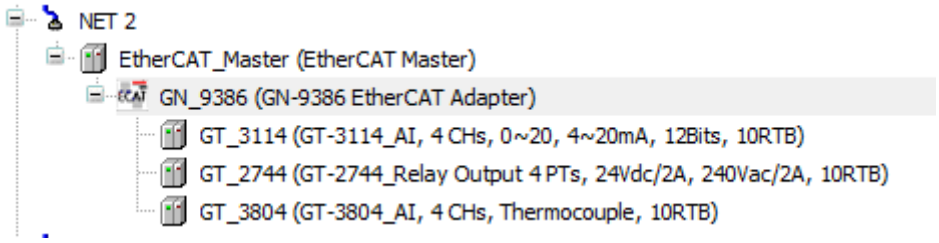


Figura 94: Exemplo de Configuração EtherCAT

ATENÇÃO

- Permitido apenas em portas NET integradas das UCPs.
- Não pode ser usado quando as NETs estiverem configuradas como redundantes/switch.
- Não pode ser usado quando o projeto tiver redundância de cluster.
- Não podem ser instanciados outros drivers na mesma porta NET que o Mestre EtherCAT.

5.5.12.1.1. EtherCAT - Scan Devices

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

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

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

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

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

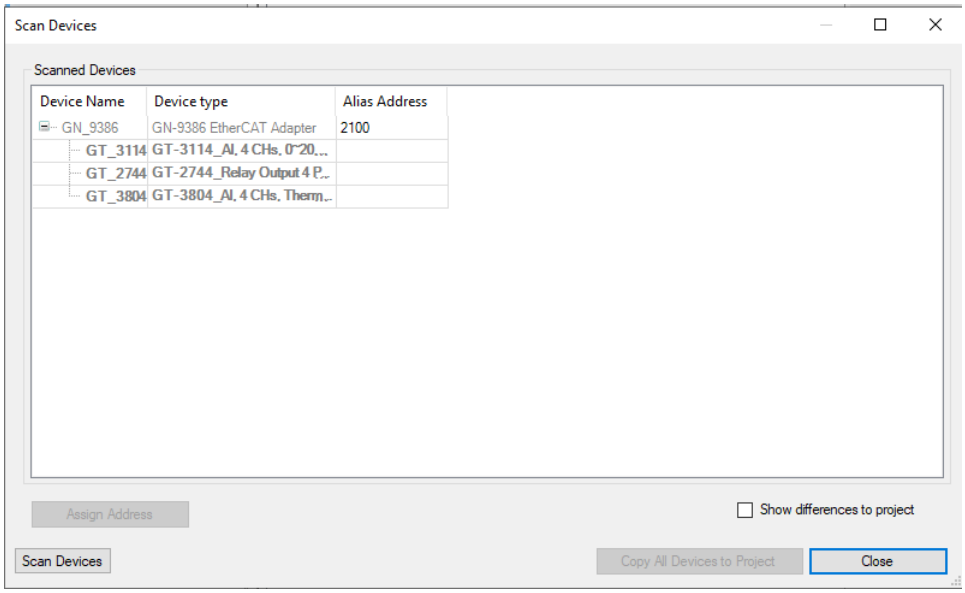


Figura 95: Diálogo Procurar Dispositivos para EtherCAT

5.5.12.2. Configuração do Mestre EtherCAT

A seguir são listadas as opções para a execução da configuração de um Mestre EtherCAT, tal como definido no *Arquivo de Descrição do Dispositivo*.

5.5.12.2.1. EtherCAT Master - General

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

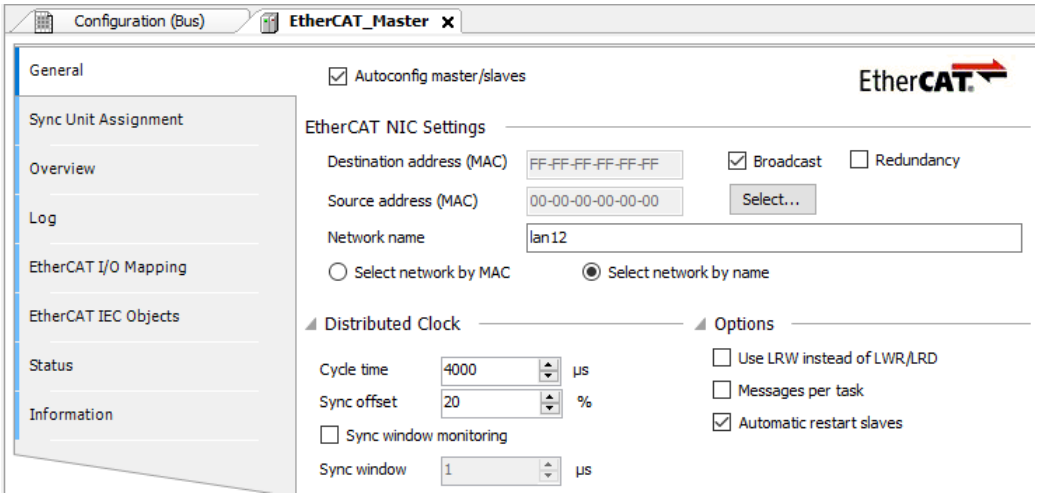


Figura 96: Diálogo de Configuração do Mestre EtherCAT

Configuração	Descrição	Padrão	Opções
Autoconfig master/slaves	Habilita a configuração automática do Mestre e dos Escravos.	Marcado	Marcado Desmarcado
Cycle time [μs]	Configura o período de tempo em que um novo telegrama de dados deve ser enviado ao barramento.	4000	2000 a 1000000
Sync offset [%]	Ajusta o deslocamento da interrupção de sincronização do Escravo EtherCAT para o ciclo do CP.	20	-50 a 50
Sync window monitoring	Se ativada esta opção permite monitorar a sincronização dos Escravos.	Desmarcado	Marcado Desmarcado
Sync window [μs]	Tempo para a Janela de Monitoração de Sincronização.	1	1 a 32768
Use LRW instead of LWR/LRD	Habilitação dos comandos combinados de leitura e escrita.	Desmarcado	Marcado Desmarcado
Messages per task	Se ativada, os comandos leitura e escrita, que estão lidando com mensagens de entrada e saída poderão ser feitos em diferentes tarefas.	Desmarcado	Marcado Desmarcado
Automatic restart slaves	Reinicia os escravos quando a comunicação for abortada.	Marcado	Marcado Desmarcado

Tabela 134: Configurações do Mestre EtherCAT

Notas:

Autoconfig master/slaves: Se esta opção estiver ativada, a maior parte da configuração do Mestre e Escravo será feita automaticamente, baseando-se nos arquivos de descrição e cálculos implícitos. Neste caso, o diálogo de configurações FMMU / Sync não estará disponível. Se esta estiver desmarcada as opções *Endereço Imagem In* e *Endereço Imagem Out* estarão disponíveis para o usuário.

ATENÇÃO

O modo *Autoconfig* é ativado por padrão e, geralmente, suficiente e fortemente recomendado para aplicações padrão. Se a opção estiver desativada, todas as definições de configuração para o Mestre e o(s) Escravo(s) terão de ser feitas manualmente e um conhecimento especializado é necessário. Para a configuração de uma comunicação Escravo-para-Escravo a opção *Autoconfiguração* tem de ser desativada.

Cycle time: Período de tempo após o qual, um novo telegrama de dados deve ser enviado ao barramento. Se a funcionalidade *Relógio Distribuído* estiver ativada, o valor deste parâmetro será transferido para os clocks dos Escravos. Assim, uma sincronização precisa de troca de dados pode ser alcançada, o que particularmente é importante nos casos em que os processos distribuídos especialmente exigem ações simultâneas. Então, uma base de tempo muito precisa para toda a rede, com um jitter significativamente menor do que um microssegundo, pode ser alcançada.

Sync offset: Este valor permite ajustar o deslocamento da interrupção de sincronização do Escravo EtherCAT para o ciclo do CP. Normalmente, o ciclo de tarefa do CP começa 20% mais tarde do que a sincronização de interrupção dos Escravos. Isto significa que a tarefa do CP pode ser atrasada por 80% do tempo de ciclo e nenhuma mensagem será perdida.

Sync window monitoring: Se a sincronização de todos os Escravos está dentro desta janela de tempo o diagnóstico do Mestre EtherCAT *bDistributedClockInSync* será definido como TRUE, caso contrário será FALSE. Quando o Relógio Distribuído é utilizado, é altamente recomendado usar uma tarefa dedicada com alta prioridade como *Tarefa Cíclica de Barramento* do mestre EtherCAT. Desta maneira, é necessário utilizar Perfis de Projeto que possibilitem a criação de novas tarefas, criar

uma tarefa cíclica com prioridade 0 (tarefa de tempo real), e atribuir a *Tarefa Cíclica de Barramento* do mestre para esta nova tarefa na aba EtherCAT Master - I/O Mapping do Mestre EtherCAT. O usuário também poderá alterar o valor da variável *wDCInSyncWindow*, configurando qual o jitter máximo permitido na sincronização entre mestre e escravos.

Use LRW instead of LWR/LRD: A ativação desta opção habilita a comunicação Escravo-para-Escravo, pois, ao invés de comandos separados de leitura (LRD) e escrita (LWR), comandos combinados leitura/escrita (LRW) serão utilizados.

Automatic restart slaves: Habilitando esta opção o Mestre reiniciará os Escravos assim que a comunicação for abortada.

5.5.12.2.2. EtherCAT Master - Sync Unit Assignment

Esta guia do editor de configuração do Mestre EtherCAT mostra todos os escravos que são inseridos abaixo de um mestre específico com uma atribuição para as unidades de sincronização.

Com o EtherCAT sync units, múltiplos escravos são configurados em grupos e subdivididos em unidades menores. Para cada grupo, o contador de trabalho pode ser monitorado para uma melhor e mais exata detecção de erros. Assim que um escravo está em falta num grupo de unidades de sincronização, os outros escravos do grupo também são mostrados como desaparecidos. A detecção ocorre imediatamente no próximo ciclo de barramento, porque o contador de trabalho é verificado continuamente. Com os diagnósticos do dispositivo, o grupo em falta pode ser remediado o mais rapidamente possível.

Os grupos não afectados permanecem operáveis sem qualquer interferência.

5.5.12.2.3. EtherCAT Master - Overview

Esta guia do editor de configuração do Mestre EtherCAT fornece uma visão geral dos estados de todos os escravos, que são inseridos abaixo deste mestre e têm um endereço. Os módulos não são exibidos.

5.5.12.2.4. EtherCAT Master - I/O Mapping

Esta guia do editor de configuração do Mestre EtherCAT oferece a possibilidade de mudar a tarefa que será utilizada para atualização do barramento.

5.5.12.2.5. EtherCAT Master - IEC Objects

Esta guia do editor de configuração do Mestre EtherCAT lista *objetos* que permitem o acesso ao dispositivo a partir da aplicação IEC. Em modo online, este é utilizado para monitoração.

5.5.12.2.6. EtherCAT Master - Status / Informations

A guia Status do editor de configuração do Mestre EtherCAT fornece informações de estado (por exemplo, '*Executando*', '*Parado*') e mensagens de diagnóstico específicas do dispositivo e do sistema de barramento interno.

A guia Informação, presente no editor de configuração do Mestre EtherCAT, exibe, caso disponíveis, as seguintes informações gerais para o módulo: *Nome, Fornecedor, Tipo, Número de Versão, Categorias, Número de Ordem, Descrição, Imagem*.

5.5.12.3. Configuração do Escravo EtherCAT

A seguir são listadas as principais opções de configuração para um Escravo EtherCAT, tal como definido no *Arquivo de Descrição do Dispositivo*.

5.5.12.3.1. EtherCAT Slave - General

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

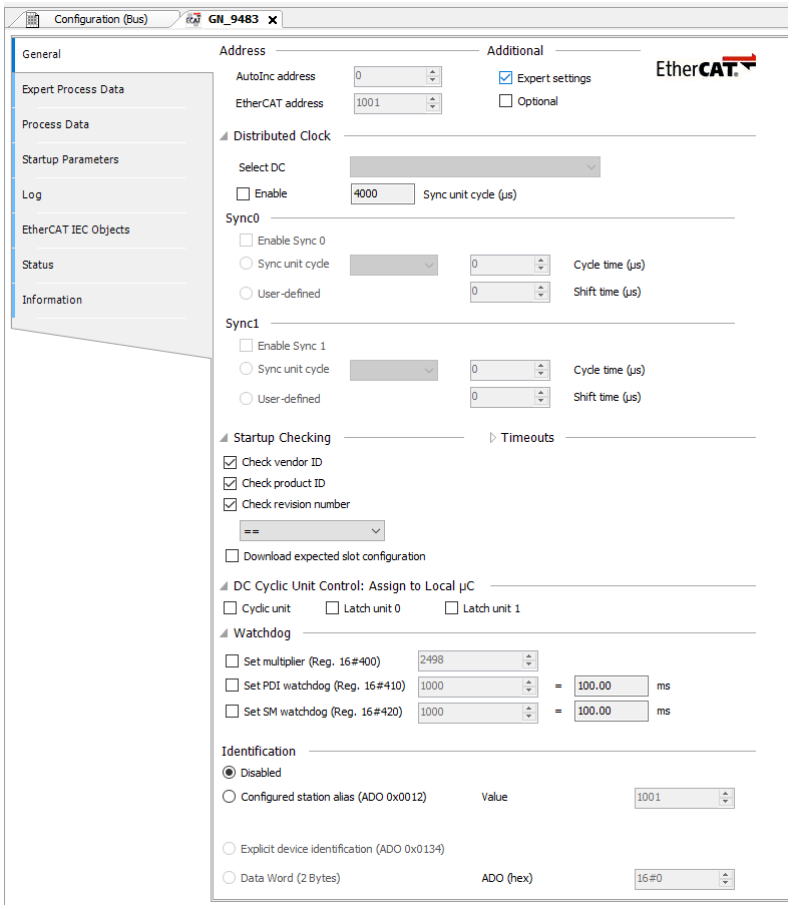


Figura 97: Diálogo de Configuração do Escravo EtherCAT

Configuração	Descrição	Padrão	Opções
AutoInc address	Endereço Autoincremental (16-bit), definido pela posição do Escravo na rede.	-	-65535 a 0
EtherCAT address	Endereço final do Escravo, atribuído pelo Mestre durante a inicialização. Este endereço é independente da posição na rede.	-	1 a 65535
Expert settings	Habilita as opções de configuração avançadas do Escravo.	Desmarcado	Marcado Desmarcado
Optional	Declara o Escravo como Opcional.	Desmarcado	Marcado Desmarcado
Select DC	Apresenta todas as configurações, para Relógios Distribuídos, fornecidas pelo arquivo de descrição do dispositivo.	-	-
Enable Distributed Clock	Habilita as opções de configuração do Relógio Distribuído.	Desmarcado	Marcado Desmarcado

Configuração	Descrição	Padrão	Opções
Sync unit cycle [μs]	Apresenta o Tempo de Ciclo configurado no Mestre.	100000	2000 a 1000000
Enable (Sync 0)	Habilita as configurações da unidade de sincronização Sync 0.	Desmarcado	Marcado Desmarcado
Sync unit cycle Sync 0	Selecionando esta opção o fator escolhido será multiplicado pelo valor em Sinc. Unidade de Ciclo.	Desmarcado	Marcado Desmarcado
User-defined Sync 0	Se marcado, o tempo desejado em microssegundos pode ser inscrito no campo Tempo de Ciclo (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 135: Configurações do Escravo EtherCAT

Notas:

AutoInc address: Este endereço é usado apenas durante a inicialização, quando o Mestre está atribuindo os endereços EtherCAT para os Escravos. Quando para este propósito, o primeiro telegrama percorre os Escravos, cada Escravo de leitura rápida aumenta seu *Endereço AutoInc* por 1. O Escravo com o endereço 0, finalmente, vai receber os dados.

Optional: Se um Escravo é declarado como *Opcional* nenhuma mensagem de erro será criada caso o dispositivo não exista no sistema de barramento. Se essa opção for ativada, um endereço de estação deve ser armazenado no dispositivo Escravo. Assim, um endereço *Pseudônimo de Estação* deve ser definido e gravado na EEPROM. Esta opção só está disponível se a opção *Autoconfiguração Mestre/Escravos* estiver ativada nas configurações do Mestre EtherCAT, e se esta função for suportada pelo Escravo EtherCAT.

Enable Distributed Clock: Se a funcionalidade *Relógio Distribuído* for ativada, o tempo de ciclo de troca de dados, exibido no campo *Unidade de Ciclo Sync(μs)* será determinado pelo *Tempo de Ciclo* do Mestre. Assim, o clock Mestre pode sincronizar a troca de dados dentro da rede. As definições para manusear a(s) unidade(s) de sincronização dependem do Escravo.

Enable Sync 0: Se essa opção for ativada, a unidade de sincronização *Sync0* é utilizada. Uma unidade de sincronização descreve um conjunto de dados do processo que são trocados de forma síncrona.

Sync unit cycle Sync 0: Se esta opção estiver ativada, o *Tempo de Ciclo* do Mestre, multiplicado pelo fator escolhido será utilizado como tempo de ciclo de sincronização para o Escravo. O campo *Tempo de Ciclo(μs)* mostra o tempo de ciclo atualmente definido.

Shift time: O Tempo de deslocamento descreve o tempo entre os eventos de sincronização (*Sync0, Sync1*) e as *Outputs Valid* ou *Input Latch*. Valor gravável, se o escravo suporta transferência de *Outputs Valid* ou *Input Latch*.

Enable Sync 1: Se essa opção for ativada, a unidade de sincronização *Sync1* é utilizada. Uma unidade de sincronização descreve um conjunto de dados do processo que são trocados de forma síncrona.

Sync unit cycle Sync1: Se essa opção for ativada, o *Tempo de Ciclo* do Mestre, multiplicado pelo fator escolhido será utilizado como tempo de ciclo de sincronização para o Escravo. O campo *Tempo de Ciclo(μs)* mostra o tempo de ciclo definido atualmente.

Check vendor ID and Check product ID: Por padrão, na inicialização do sistema o *ID do Fornecedor* e/ou o *ID do Produto* serão verificados contra as configurações atuais. Se for detectada alguma diferença, o barramento será interrompido e nenhuma ação adicional será executada. Isso serve para evitar o download de uma configuração errada. Estas opções têm a finalidade de desativar esta verificação.

SDO access: Por padrão não há tempo limite definido para a ação de envio de uma lista SDO ao iniciar o sistema. Porém, se houver a necessidade de verificar se esta ação excede certo tempo, este deve ser especificado neste campo.

I -> P: Por padrão não há tempo limite definido para a ação de troca do modo *Inicial* para o modo *Pré-Operacional*. Porém, se houver a necessidade de verificar se esta ação excede certo tempo, este deve ser especificado neste campo.

P -> S / S-> O: Por padrão não há tempo limite definido para a ação de troca dos modos *Pré-Operacional* para *Segurança-Operacional* e respectivamente de *Segurança-Operacional* para *Operacional*. Porém, se houver a necessidade de verificar se esta ação excede certo tempo, este deve ser especificado neste campo.

DC Cyclic Unit Control: Escolha a(s) opção(ões) desejada(s), relativa às funções de *Relógio Distribuído*, a fim de definir o que deve ser atribuído ao microprocessador local. O controle é feito no registro 0x980 do Escravo EtherCAT. As configurações possíveis são: *Unidade Cíclica, Unidade Latch 0, Unidade Latch 1*.

Enable: Se a definição de Opcional não estiver ativada, essa configuração pode ser ativada se for suportada pela descrição do dispositivo Escravo. Ele permite a atribuição direta de um endereço de pseudônimo, a fim de obter o endereço dos Escravos independente de sua posição dentro do barramento. Se a opção *Opcional* estiver ativada, esta caixa de seleção estará desativada.

5.5.12.3.2. EtherCAT Slave - Process Data

A guia *Process Data* do editor de configuração do Escravo EtherCAT mostra os dados do processo de entrada e de saída do Escravo, cada um definido por nome, tipo e índice do arquivo de descrição do dispositivo, como pode ser visto na figura abaixo.

As entradas selecionadas (a serem lidas) e saídas (a serem escritas) do dispositivo estarão disponíveis no diálogo de EtherCAT Slave - Module I/O Mapping como saídas e entradas para o CP, no qual as variáveis do projeto podem ser mapeadas.

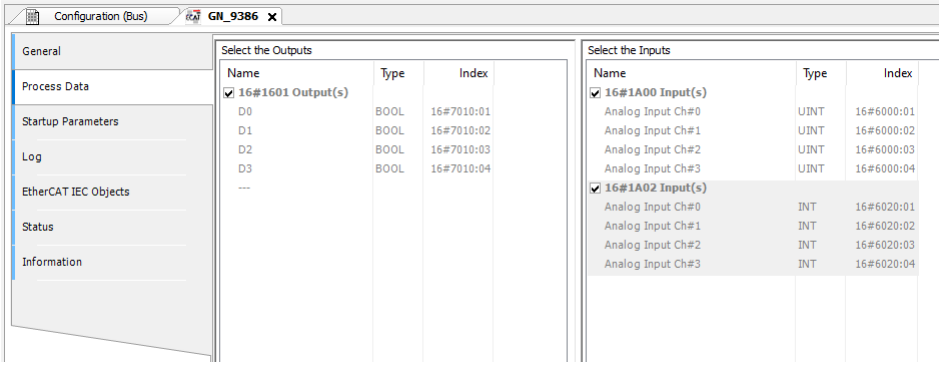


Figura 98: Diálogo Dados do Processo

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.

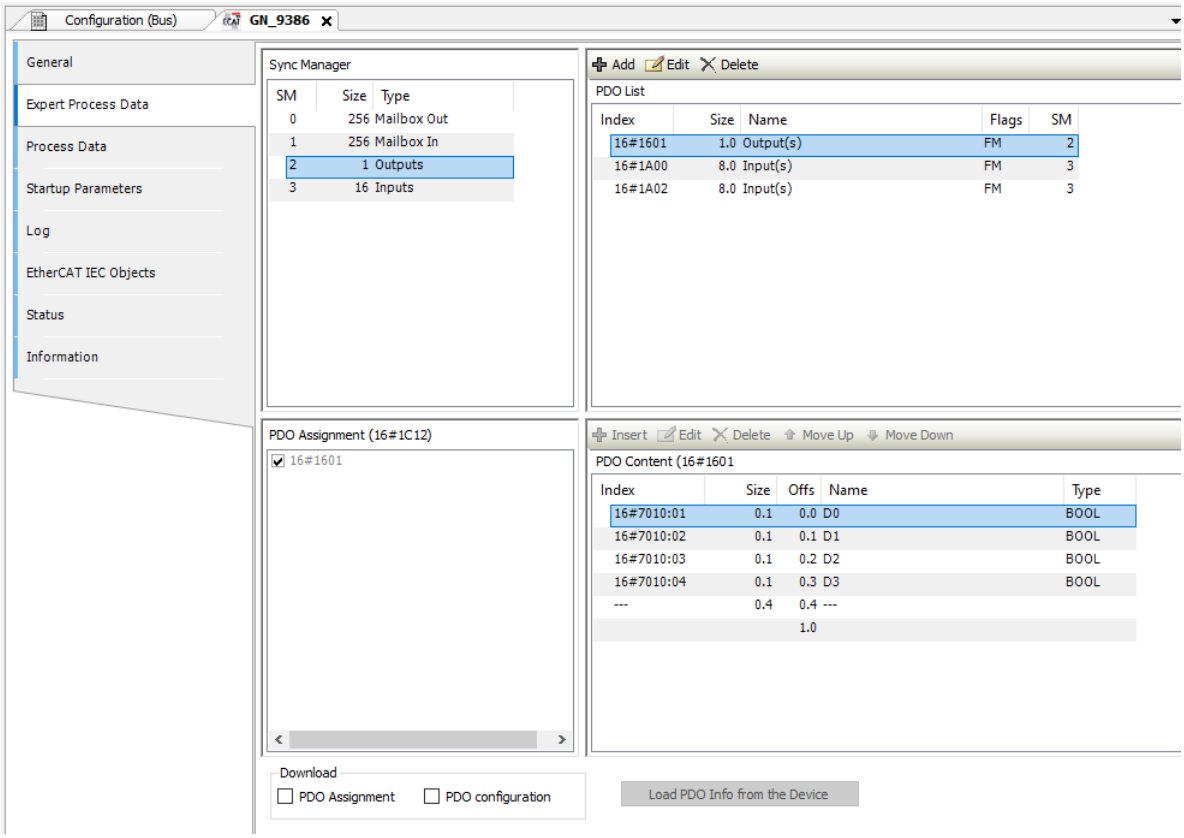


Figura 99: Diálogo Dados de Processo Avançados

Esta guia está dividida em quatro seções e duas opções:

- *Sync Manager*: Lista de SM *Sync Manager* com o tamanho dos dados e tipo de PDO.
- *PDO Assignment*: Lista de PDOs atribuídos ao *Sync Manager* selecionado. A caixa de seleção ativa o PDO e canais de E/S são criados. É parecido com as janelas de configuração de PDO simples. Aqui apenas PDOs podem ser ativados ou desativados.
- *PDO List*: Lista de todos os PDOs definidos no arquivo de descrição do dispositivo. PDOs individuais podem ser apagados, editados ou adicionados por meio da execução do respectivo comando no menu de contexto.
- *PDO Content*: Exibe o conteúdo do PDO selecionado na seção acima. As inscrições podem ser apagadas, editadas ou adicionadas por meio da execução do respectivo comando a partir do menu de contexto.
- *PDO Assignment*: Se ativado, um comando de escrita CoE será adicionado ao índice 0x1CXX para escrever a configuração PDO 0x16XX ou 0x1A00.
- *PDO configuration*: Se ativado, vários comandos da gravação CoE serão adicionados ao escrever o mapeamento PDO no Escravo.

ATENÇÃO

Se o Escravo não suportar a configuração PDO, o download desta pode resultar em um erro de Escravo. Esta função deve ser usada apenas por especialistas.

5.5.12.3.3. *EtherCAT Slave - Edit PDO List*

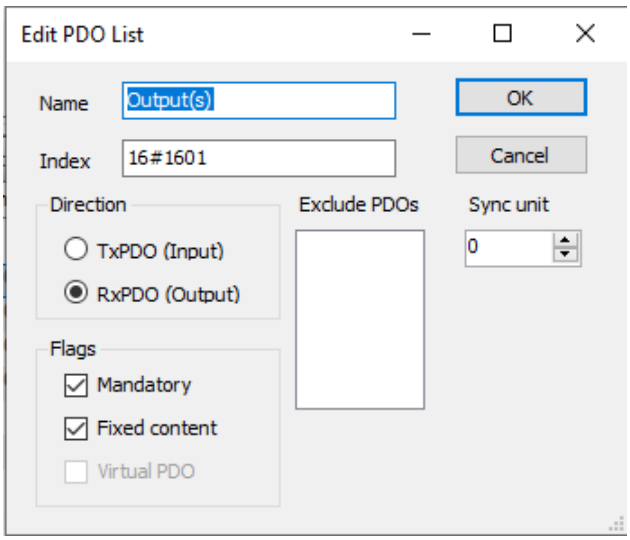


Figura 100: Diálogo Editar Lista PDO

Este diálogo é aberto através do menu de contexto da área Lista PDO, apresentada na Figura 99. 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.

5.5.12.3.4. *EtherCAT Slave - Startup Parameters*

Na guia *Startup Parameters* podem ser definidos parâmetros para o dispositivo, que serão repassados pelos SDOs (*Service Data Objects*) ou IDN na inicialização do sistema. As opções presentes neste ambiente, além da possibilidade de acesso, variam de acordo com o Escravo EtherCAT utilizado e estão presentes no seu *Arquivo de Descrição do Dispositivo*.

5.5.12.3.5. *EtherCAT Slave - Module I/O Mapping*

Esta guia do editor de configuração do Escravo EtherCAT oferece a possibilidade de atribuir as variáveis de projeto para respectivas entradas e saídas do EtherCAT. Assim, as variáveis do Escravo EtherCAT podem ser controladas pela *Aplicação de Usuário*.

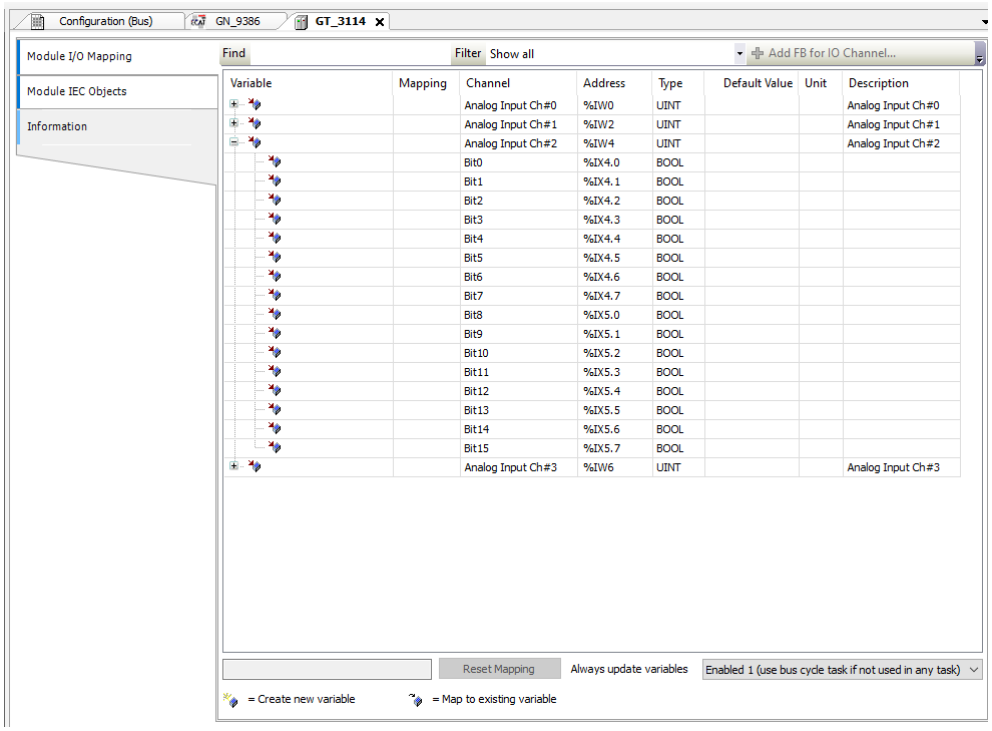


Figura 101: Diálogo Mapeamento de E/S Escravo

5.5.12.3.6. EtherCAT Slave - Status and Information

A guia *Status* do editor de configuração do Escravo EtherCAT fornece informações de estado (por exemplo, '*Executando*', '*Parado*') e mensagens de diagnóstico específicas do dispositivo e do sistema de barramento interno.

A guia *Information*, presente no editor de configuração do Escravo EtherCAT, exibe, caso disponíveis, as seguintes informações gerais para o módulo: *Nome, Fornecedor, Tipo, Número de Versão, Categorias, Número de Ordem, Descrição, Imagem*.

5.5.13. EtherNet/IP

O EtherNet/IP é um protocolo de arquitetura mestre-escravo, o qual consiste de um EtherNet/IP Scanner (o mestre) e um, ou mais, EtherNet/IP Adapters (o escravo).

O EtherNet/IP é um protocolo baseado na CIP (*Common Industrial Protocol*), o qual tem dois propósitos primários: O transporte de dados de controle-orientado associados com dispositivos de E/S e o transporte de outras informações relacionadas ao sistema sendo controlado, tais como parâmetros de configuração e diagnósticos. O primeiro é realizado por mensagens implícitas, enquanto o segundo é realizado através de mensagens explicitas.

O sistema em execução, das UCPs, pode atuar tanto como Scanner como Adapter. Cada interface NET das UCPs suporta apenas uma instância EtherNet/IP e ele não pode ser instanciado em um módulo de expansão Ethernet.

Uma instância EtherNet/IP Adapter suporta um número ilimitado de módulos ou de bytes de Entrada/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 usado em conjunto com Modo Redundante da interface ethernet nem com redundância de Half-Cluster.

ATENÇÃO

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

5.5.13.1. Interface EtherNet/IP

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

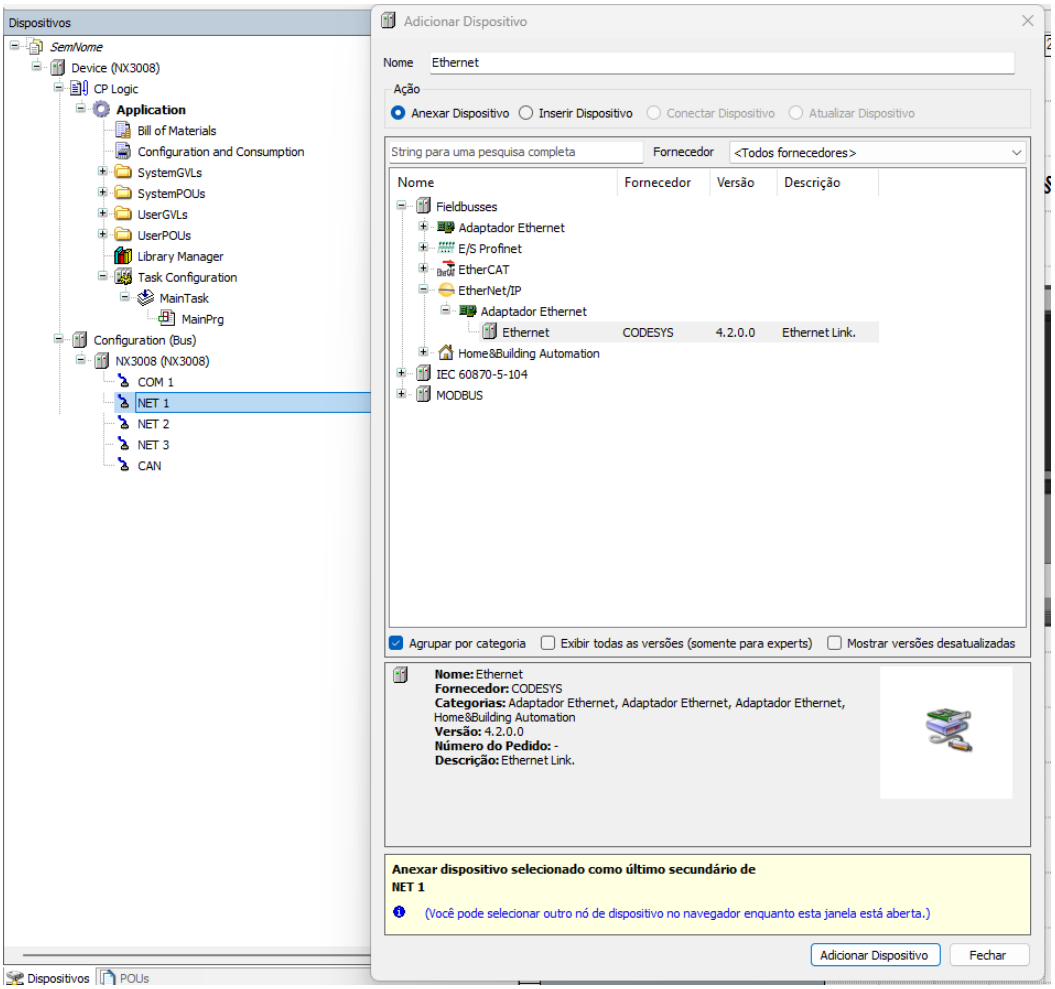


Figura 102: Adicionando um Adaptador Ethernet

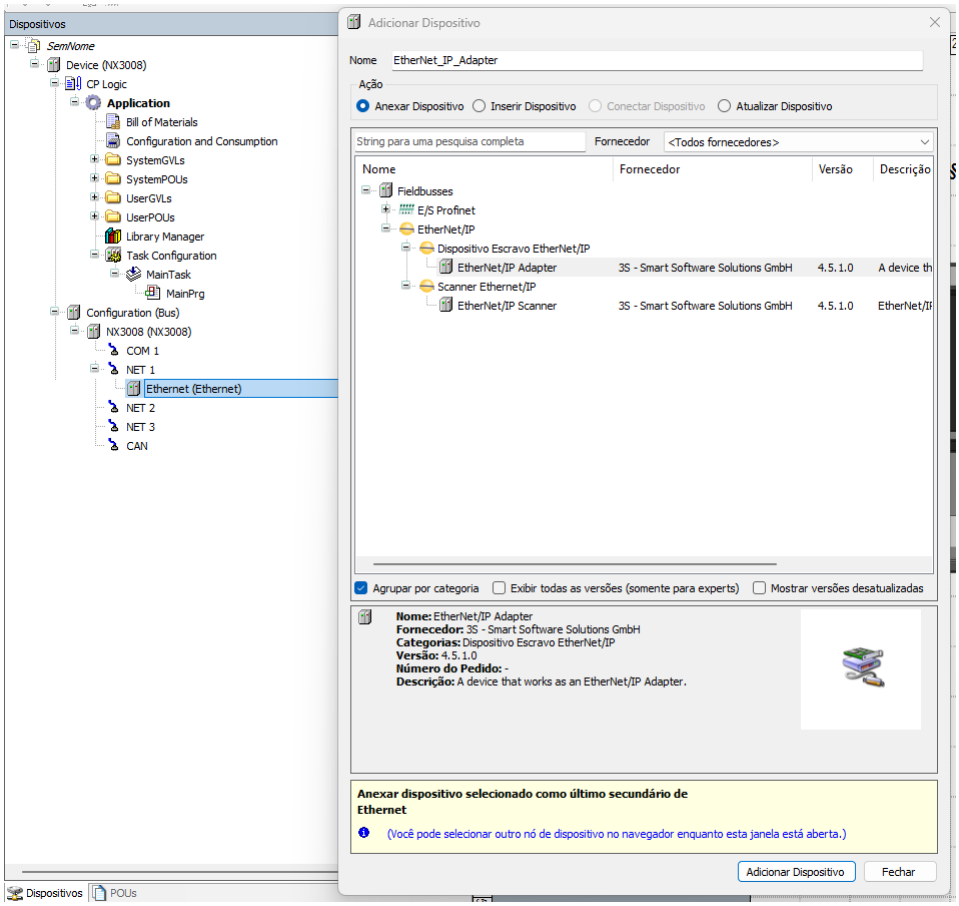


Figura 103: Adicionando um Adapter ou Scanner EtherNet/IP

5.5.13.2. Configuração do Scanner EtherNet/IP

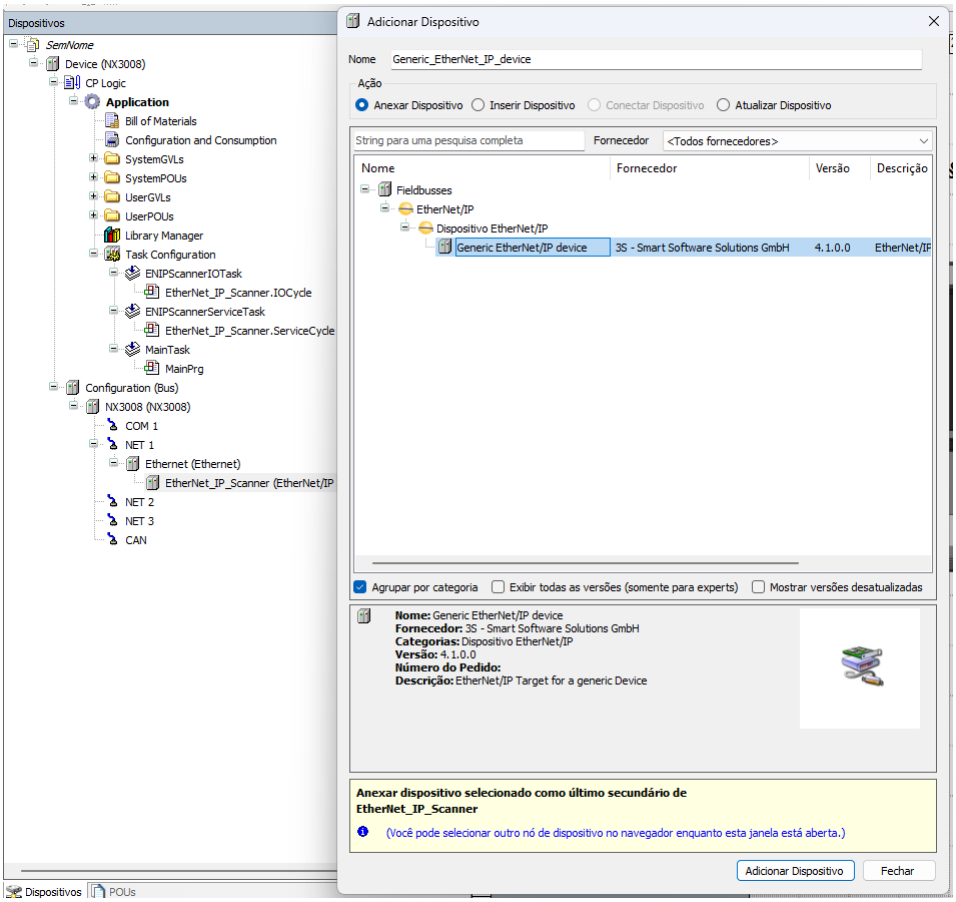


Figura 104: Adicionando um Adapter EtherNet/IP Sob o Scanner

5.5.13.2.1. Geral

Depois de abrir o Dispositivo declarado sob o Scanner é possível configurá-lo como for necessário. A primeira aba é *Geral*, nela é possível configurar o *Endereço IP* e o parâmetro *Chave Eletrônica*. Esses parâmetros devem ser marcados ou desmarcados se um Adapter sendo usado está instalado no MasterTool. De outra forma, se o Adapter usado é do tipo *Generic*, então os campos *ID do Fornecedor*, *Tipo do Dispositivo*, *Código do Produto*, *Revisão grande* e *Revisão pequena* devem ser preenchidos com as informações corretas do fabricante e as caixas marcadas tanto quanto necessário. A Altus por sua vez, tem um ID próprio, que é o ID *"1454"*.

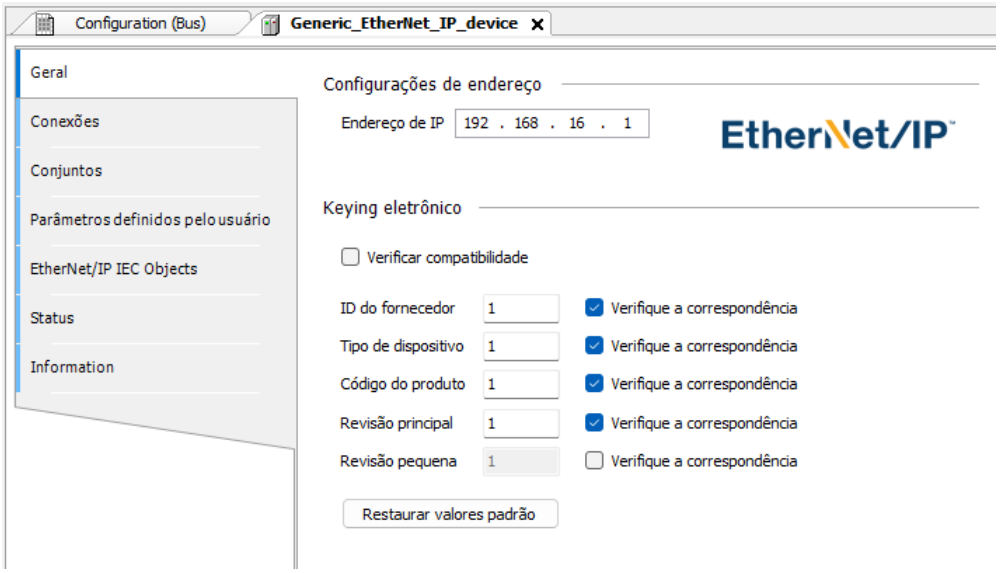


Figura 105: Aba Geral - EtherNet/IP

5.5.13.2.2. *Conexões*

A área superior da aba *Conexões* mostra uma lista de todas as conexões configuradas. Quando há uma conexão *Exclusive Owner* no arquivo EDS, ela é inserida automaticamente quando o Adapter é adicionado. Os dados de configuração para estas conexões podem ser mudados mais a baixo nesta janela.

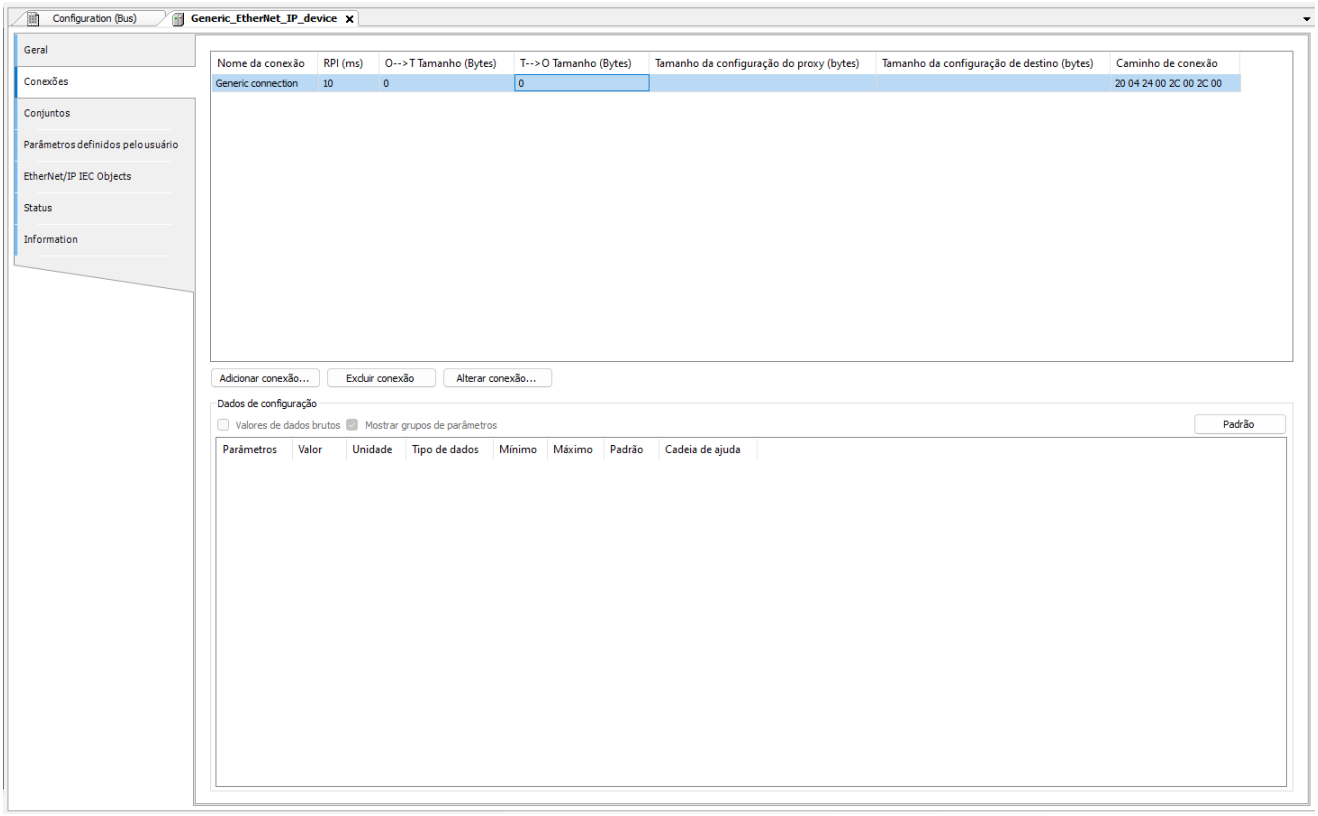


Figura 106: Aba Conexões - EtherNet/IP

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 136: 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.

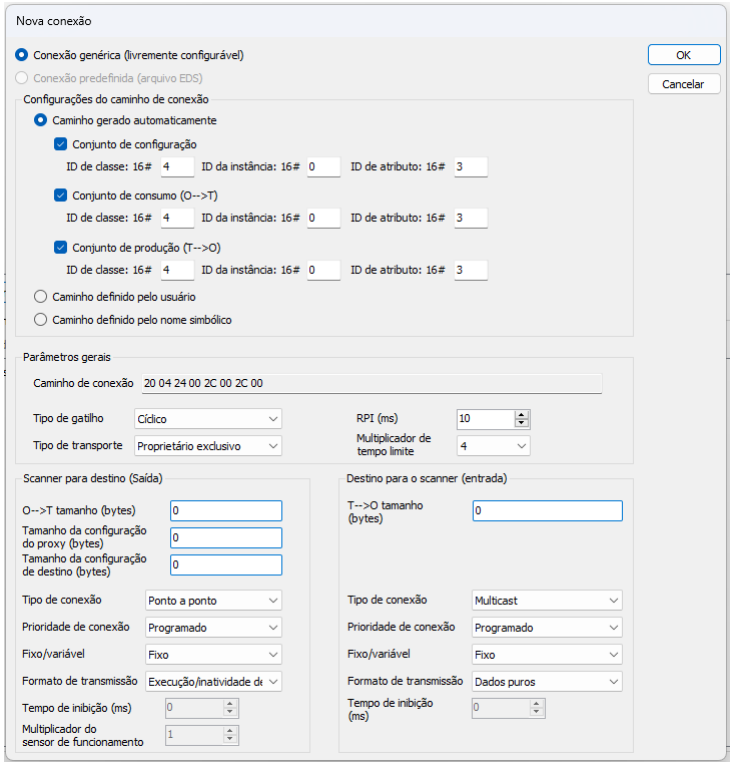


Figura 107: Janela de Novas Conexões EtherNet/IP

5.5.13.2.3. *Assemblies*

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

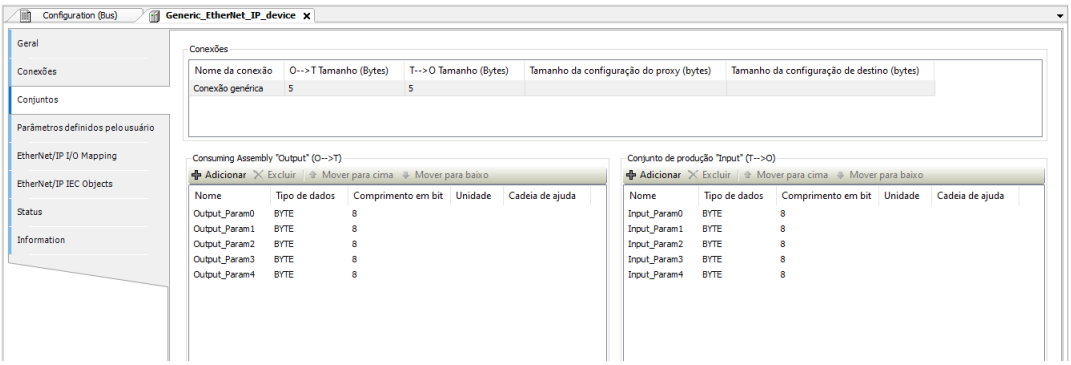


Figura 108: Aba Assemblies - EtherNet/IP

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 137: 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 138: Janela "Adicionar Entrada/Saída" – EtherNet/IP

5.5.13.2.4. EtherNet/IP: Mapeamento de E/S

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

5.5.13.3. Configuração do Adapter EtherNet/IP

O Adapter EtherNet/IP requer módulos EtherNet/IP. Os módulos vão prover mapeamentos de E/S que podem ser manipulados pela aplicação do usuário pelos endereços %I ou %Q de acordo com sua configuração.

Novos Adapters podem ser instalados no MasterTool com os arquivos EDS. As opções de configuração podem divergir dependendo do arquivo de descrição do dispositivo do Adapter adicionado.

5.5.13.3.1. Geral

A primeira aba do EtherNet/IP Adapter é a aba *Geral*. Nela é possível configurar os parâmetros da *Chave Eletrônica*, utilizada no Scanner para verificar a compatibilidade. Nesta aba também é possível instalar o EDS do dispositivo diretamente no repositório de dispositivos do MasterTool ou exporta-lo.

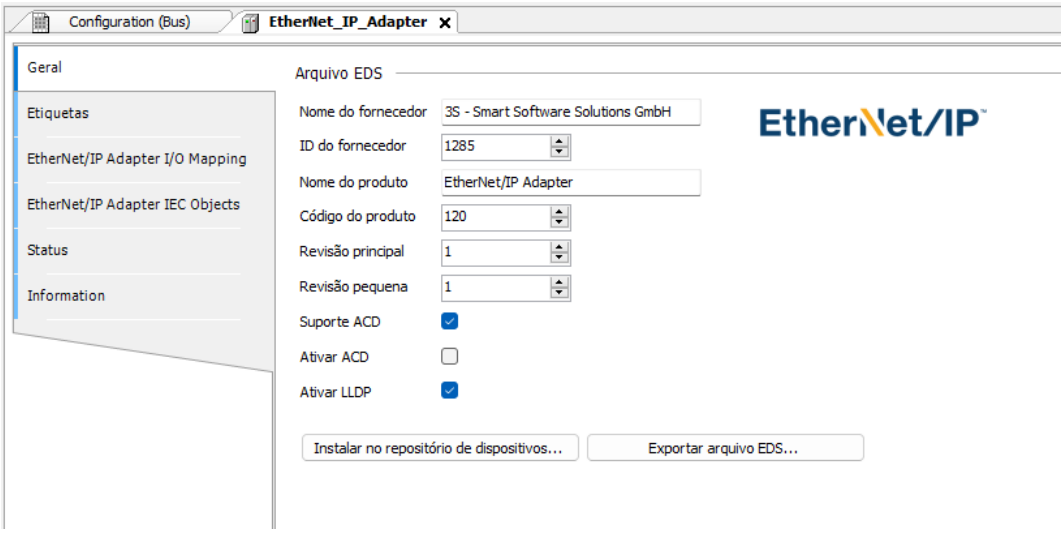


Figura 109: Aba Geral - EtherNet/IP_Adapter

5.5.13.3.2. EtherNet/IP Adapter: Mapeamento de E/S

Na aba *Mapeamento de E/S*, é possível configurar em qual tarefa de ciclo do barramento o Adapter irá executar.

5.5.13.4. Configuração do Módulo EtherNet/IP

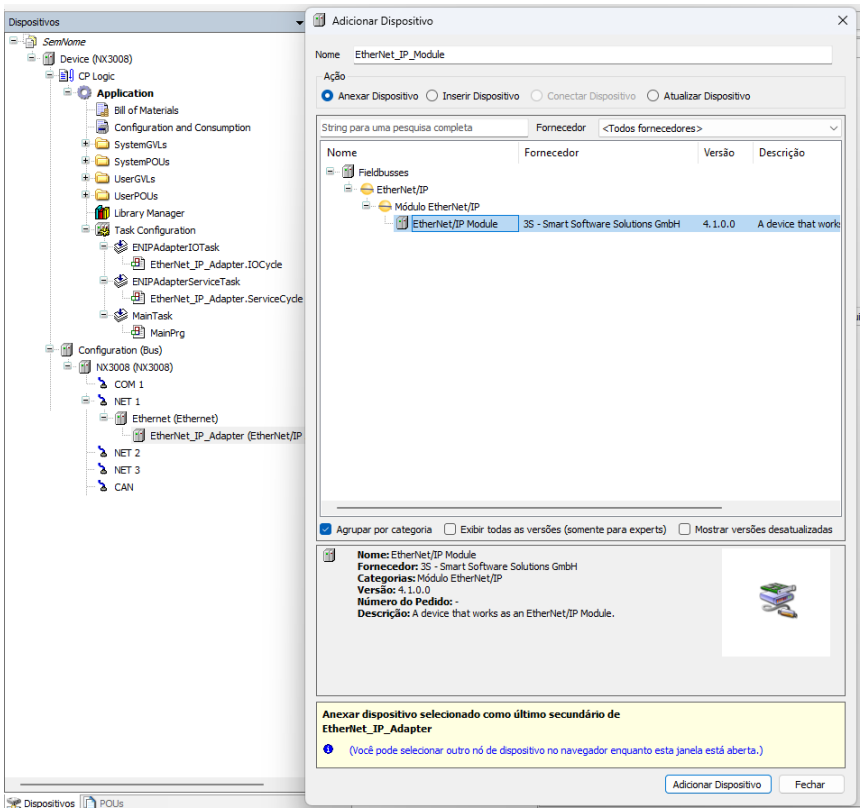


Figura 110: Adicionando um Módulo EtherNet/IP Abaixo do Adapter

5.5.13.4.1. *Assemblies*

Os parâmetros da aba *Geral* do módulo seguem as mesmas regras descritas nas tabelas 137 e 138.

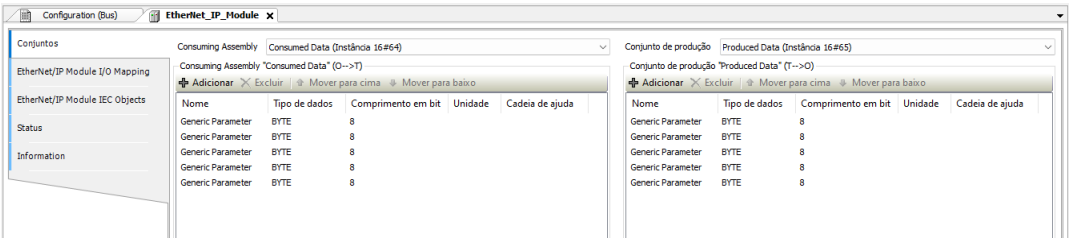


Figura 111: Aba Assemblies Módulo - EtherNet/IP

5.5.13.4.2. *EtherNet/IP Module: Mapeamento de E/S*

Aba *Mapeamento de E/S* mostra, na coluna *Variável*, o nome das instâncias de Adapter, automaticamente geradas. Desta forma, a instância pode ser acessada pela aplicação do usuário.

5.5.14. IEC 60870-5-104 Servidor

Ao selecionar esta opção no MasterTool, a UCP passa a ser servidor da comunicação IEC 60870-5-104, permitindo a conexão com até três dispositivos clientes IEC 60870-5-104. Para cada cliente, o driver possui uma fila de eventos exclusiva com as seguintes características:

- Tamanho: 1000 eventos
- Retentividade: não é retentiva
- Política de estouro: mantém o mais recente

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

- Adicionar a instância do protocolo Servidor IEC 60870-5-104 a um canal Ethernet disponível. Para realizar esse procedimento, consultar a seção Inserindo uma Instância de Protocolo
- Configurar a interface Ethernet. Para realizar esse procedimento, consultar a seção Configuração das Interfaces Ethernet
- Configurar os parâmetros gerais do protocolo Servidor IEC 60870-5-104, com o modo de conexão Porta ou IP, e o número da porta TCP quando o modo de conexão selecionado for o modo IP
- Adicionar e configurar dispositivos, definindo os devidos parâmetros
- Adicionar e configurar os mapeamentos IEC 60870-5-104, especificando o nome da variável, tipo do objeto, endereço do objeto, tamanho, faixa, banda morta, e tipo da banda morta
- Configurar os parâmetros da camada de enlace, especificando os endereçamentos, os time-outs de comunicação e os parâmetros de comunicação
- Configurar os parâmetros da camada de aplicação, configuração do sincronismo, comandos, bem como o modo de transmissão dos objetos *Integrated Totals*

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

5.5.14.1. Tipos de Dados

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

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

Tabela 139: Declaração de Variáveis para IEC 60870-5-104

Notas:

Regulating Step Command: Os estados *Lower* e *Higher* do objeto C_RC_NA estão associados respectivamente aos estados *OFF* e *ON* do tipo interno *DBP*.

Step Position Information: Conforme definido no item 7.3.1.5 da norma IEC 60870-5-101 essa variável de 8 bits é composta por dois campos: valor (definido pelos 7 bits menos significativos da variável) e transiente (definido como o bit mais significativo, o qual indica quando o dispositivo medido está transicionando).

Abaixo, exemplo de código para manipulação dos campos em uma variável do tipo USINT. Atenção, pois este código não consiste se o valor de entrada está dentro da faixa, portanto esta consistência fica a cargo do usuário.

```
PROGRAM UserPrg
VAR
usiVTI: USINT;  // Valor com indicação de estado transiente, mapeado para o
   Cliente
siValue: SINT;  // Valor a ser convertido para VTI. Deve estar entre -64 e +63
bTransient: BOOL; // Transiente a ser convertido para VTI
END_VAR

usiVTI := SINT_TO_USINT(siValue) AND 16#3F;
IF siValue < 0 THEN
usiVTI := usiVTI OR 16#40;
END_IF
IF bTransient THEN
usiVTI := usiVTI OR 16#80;
END_IF
```

PROFIBUS: Com exceção dos objetos digitais, os tipos de dados de objetos analógicos e contadores do protocolo IEC 60870-5-104 são diferentes dos tipos de dados dos módulos analógicos e contadores PROFIBUS, não sendo possível mapear tais tipos de variáveis PROFIBUS diretamente para os clientes IEC 60870-5-104.

Nestes casos é necessário criar uma variável intermediária, a ser mapeada no cliente IEC 60870-5-104, e converter os tipos adequadamente, como pode ser observado no código exemplo a seguir.

```
PROGRAM UserPrg
VAR
  iAnalogIn:    INT;
  iAnalogOut:   INT;
  diCounter:    DINT;
END_VAR

// Conversão da entrada analógica de WORD (PROFIBUS) para INT (IEC104)
iAnalogIn:= WORD_TO_INT(wNX6000in00);

// Conversão da saída analógica de INT (IEC104) para WORD (PROFIBUS)
wNX6100out00:= INT_TO_WORD(iAnalogOut);

// Conversão do contador de WORDs high+low (PROFIBUS) para DINT (IEC104)
diCounter:= DWORD_TO_DINT(ROL(WORD_TO_DWORD(wNX1005cnt00H), 16) OR wNX1005cnt00L
   );
```

5.5.14.2. Pontos Duplos

Os pontos digitais duplos são utilizados para indicar a posição de equipamentos como válvulas, disjuntores e seccionadoras, onde a transição entre os estados aberto e fechado demandam um determinado tempo, podendo assim indicar um estado intermediário de transição entre os dois estados finais.

Pontos digitais duplos também são utilizados como saídas e, de uma forma análoga, é necessário manter uma das saídas acionada por determinado tempo para a transição ser completada. Tal acionamento é realizado através de pulsos, também conhecido por comandos trip/close, com determinada duração (suficiente para o chaveamento do dispositivo sob controle).

Consultar a seção Pontos Duplos do manual de utilização, para informações sobre a representação de pontos digitais duplos através dos tipos de dados DBP.

Uma vez que os módulos de entrada e saída digital da Série Nexto não suportam o mapeamento de pontos DBP, são necessárias algumas artimanhas de aplicação para tornar isto possível. Lembrando que também não é possível a utilização da função *PulsedCommand*, definido na biblioteca *LibRtuStandard*, para acionamento dos pontos digitais duplos na Série Nexto.

5.5.14.2.1. Pontos Duplos de Entrada Digital

Para módulos de entrada digital será necessária a declaração de duas variáveis auxiliares, a serem mapeadas no módulo de entrada digital, além do ponto duplo que se deseja mapear no servidor:

- a variável de valor do ponto duplo: tipo DBP
- a variável de valor do ponto simples OFF/TRIP: tipo BOOL
- a variável de valor do ponto simples ON/CLOSE: tipo BOOL

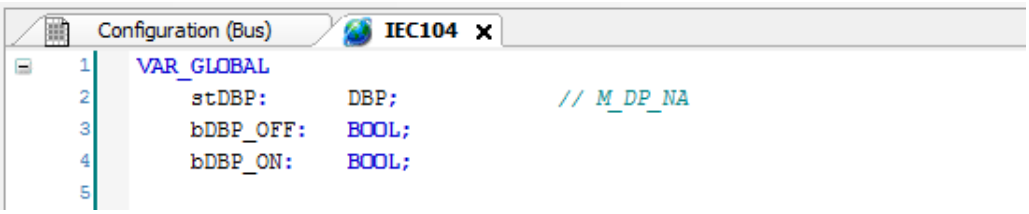


Figura 112: Exemplo de Declaração das Variáveis de Ponto Duplo

Feita a declaração das variáveis, é necessário criar um vínculo entre a variável de valor e a qualidade do módulo de entrada digital, através da aba de *Pontos Internos* da UCP:

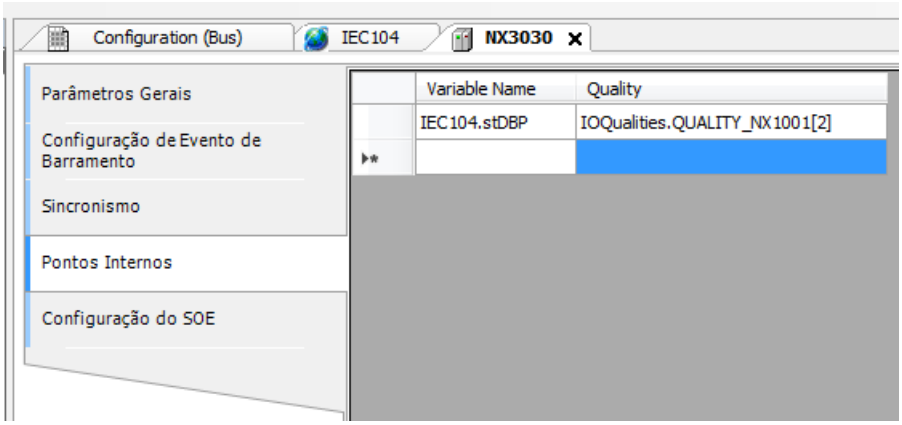


Figura 113: Atribuição das Variáveis de Ponto Duplo aos Pontos Internos

A variável de valor do ponto duplo deve ser mapeada no driver servidor IEC 60870-5-104, e as duas variáveis simples no módulo de entrada digital da Série Nexto (neste exemplo, um NX1001). Tipicamente, o estado OFF (TRIP) é mapeado na entrada par e o estado ON (CLOSE) na entrada ímpar.

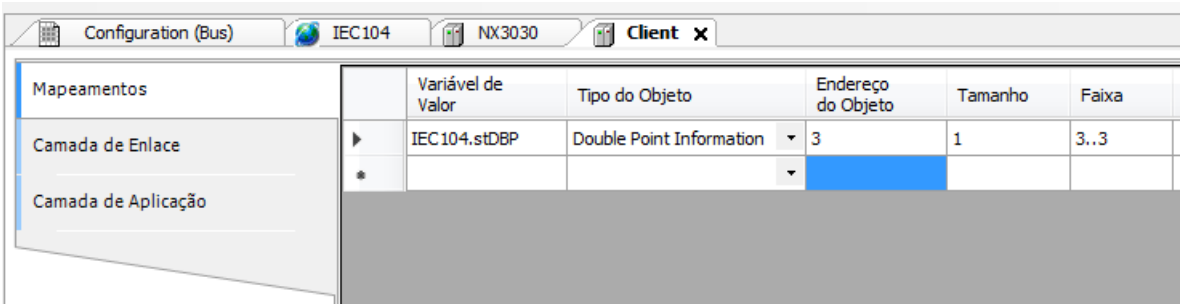


Figura 114: Mapeamento das variáveis de Ponto Duplo no Cliente IEC 60870-5-104

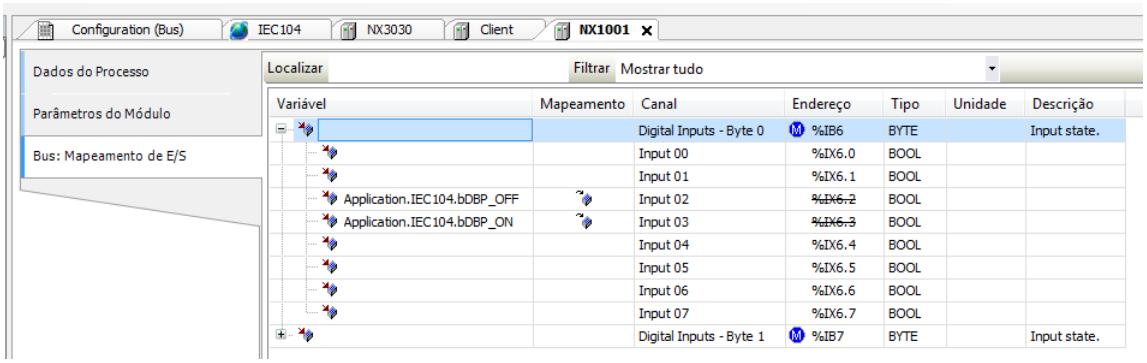


Figura 115: Mapeamento das Variáveis às Entradas do Módulo

Por último, o usuário deve inserir duas linhas de código em sua aplicação, a serem executadas ciclicamente, para atribuição dos valores das variáveis simples ao ponto duplo:

- variável de valor DBP, índice ON, recebe o valor do ponto simples ON

- variável de valor DBP, índice OFF, recebe o valor do ponto simples OFF

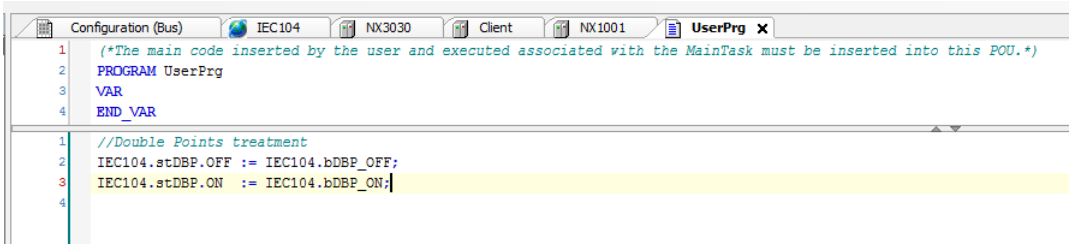


Figura 116: Atribuição dos Valores das Variáveis ao Ponto Duplo

5.5.14.2.2. *Pontos Duplos de Saída Digital*

Para módulos de saída digital deverá ser utilizado o bloco função *CommandReceiver* para a interceptação dos comandos de acionamento de pontos duplos oriundos dos clientes IEC 60870-5-104. Consultar a seção Interceptação de Comandos Oriundos do Centro de Controle para maiores informações.

O código exemplo abaixo, POU *CmdRcv*, trata comandos pulsados recebidos de clientes para um ponto digital duplo, mapeado num módulo NX2020. Além do código ST a seguir, é necessário mapear o ponto DBP no servidor IEC 60870-5-104 do Nexto.

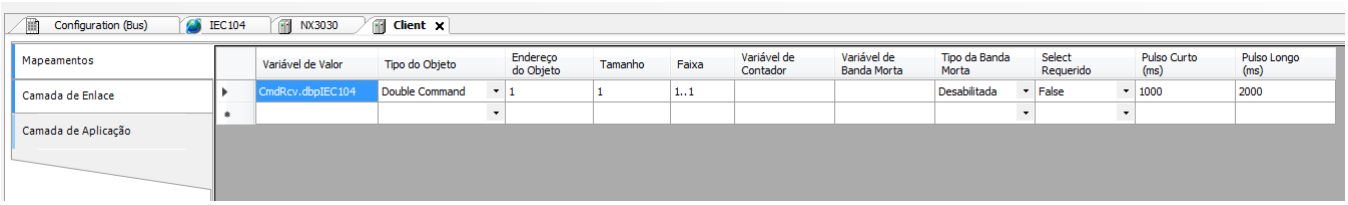


Figura 117: Mapeamento das variáveis de Pontos Duplos de Saída Digital no Cliente IEC 60870-5-104

```
PROGRAM CmdRcv
VAR
CmdReceive:    CommandReceiver; // Instância do Interceptador
fbPulsedCmd:  PulsedCommandNexto; // Instância do comando pulsado
byResult:    BYTE; // Resultado do comando pulsado
dbpIEC104:     DBP; // Variável mapeada no servidor IEC 104
bSetup:    BOOL:= TRUE; // Configuração inicial do interceptador
END_VAR

// Executa a configuração da função no primeiro ciclo
IF bSetup THEN
CmdReceive.dwVariableAddr:= ADR(dbpIEC104);
CmdReceive.bExec:= TRUE;
CmdReceive.eCommandResult:= COMMAND_RESULT.NONE;
CmdReceive.dwTimeout:= 256 * 10;
bSetup:= FALSE;
END_IF

// Caso um comando seja capturado:
IF CmdReceive.bCommandAvailable THEN

// Trata cada um dos comandos possíveis de serem interceptados
CASE CmdReceive.sCommand.eCommand OF
```

```
COMMAND_TYPE.NO_COMMAND:

  // Informar que existe um comando inválido.
  // Não faz nada. Deve sair por timeout.

COMMAND_TYPE.SELECT:

  // Trata apenas comandos para pontos duplos
  IF CmdReceive.sCommand.sSelectParameters.sValue.eParamType =
   DOUBLE_POINT_COMMAND THEN
    // Retorna comando finalizado com sucesso
    // (controlado pelo protocolo IEC104)
    byResult:= 7;
  ELSE
    // Retorna comando não suportado
    byResult:= 1;
  END_IF

COMMAND_TYPE.OPERATE:

  // Trata apenas comandos para pontos duplos
  IF CmdReceive.sCommand.sOperateParameters.sValue.eParamType =
   DOUBLE_POINT_COMMAND THEN
    // Geração do pulso nas saídas
    IF CmdReceive.sCommand.sOperateParameters.sValue.sDoublePoint.bValue THEN
      // Executa a função de TRIP
      fbPulsedCmd(
        byCmdType:= 101,
        byPulseTime:= DWORD_TO_BYTE(CmdReceive.sCommand.sOperateParameters.
    sValue.sDoublePoint.sPulseConfig.dwOnDuration/10),
        ptDbpVarAdr:= ADR(dbpIEC104),
        stQuality:= IOQualities.QUALITY_NX2020[4],
        byStatus=> byResult);
    ELSE
      // Executa a função de CLOSE
      fbPulsedCmd(
        byCmdType:= 102,
        byPulseTime:= DWORD_TO_BYTE(CmdReceive.sCommand.sOperateParameters.
    sValue.sDoublePoint.sPulseConfig.dwOffDuration/10),
        ptDbpVarAdr:= ADR(dbpIEC104),
        stQuality:= IOQualities.QUALITY_NX2020[5],
        byStatus=> byResult);
    END_IF
  ELSE
    // Retorna comando não suportado
    byResult:= 1;
  END_IF

COMMAND_TYPE.CANCEL:

  // Retorna comando finalizado com sucesso
  // (controlado pelo protocolo IEC104)
  byResult:= 7;

END_CASE
```

```
// Trata o resultado da função de comando pulsado
// e gera a resposta para o comando interceptado
CASE byResult OF
1:  // Tipo inválido de comando
  CmdReceive.eCommandResult:= COMMAND_RESULT.NOT_SUPPORTED;
  CmdReceive.bDone:= TRUE;
2:  // Parâmetros de entrada inválidos
  CmdReceive.eCommandResult:= COMMAND_RESULT.INCONSISTENT_PARAMETERS;
  CmdReceive.bDone:= TRUE;
3:  // Mudança de parâmetro em execução
  CmdReceive.eCommandResult:= COMMAND_RESULT.PARAMETER_CHANGE_IN_EXECUTION;
  CmdReceive.bDone:= TRUE;
4:  // Módulo não respondeu ao comando (ausente)
  CmdReceive.eCommandResult:= COMMAND_RESULT.HARDWARE_ERROR;
  CmdReceive.bDone:= TRUE;
5:  // Comando iniciado e em execução (não retorna nada)
6:  // Outro comando já foi enviado para este ponto e está em execução
  CmdReceive.eCommandResult:= COMMAND_RESULT.LOCKED_BY_OTHER_CLIENT;
  CmdReceive.bDone:= TRUE;
7:  // Comando finalizado com sucesso
  CmdReceive.eCommandResult:= COMMAND_RESULT.SUCCESS;
  CmdReceive.bDone:= TRUE;
END_CASE

END_IF

CmdReceive();

IF CmdReceive.bDone THEN
CmdReceive.bDone:= FALSE;
END_IF
```

Como pode ser observado no código anterior, para auxiliar na geração de pulsos em saídas digitais duplas no Nexto, foi criado e utilizado um bloco função equivalente à função *PulsedCommand* da biblioteca *LibRtuStandard*. O bloco função *PulsedCommandNexto()* aparece codificado em linguagem ST.

```
FUNCTION_BLOCK PulsedCommandNexto
VAR_INPUT
byCmdType:    BYTE;     // Tipo comando:
            // 100 = status
            // 101 = close/on
            // 102 = trip/off
byPulseTime:  BYTE;     // Duração do pulso (em centésimos de segundo)
ptDbpVarAdr:  POINTER TO DBP; // Endereço variável DBP (pode ser a mapeada)
stQuality:    QUALITY;    // Qualidade do ponto DBP (do módulo digital)
END_VAR
VAR_OUTPUT
bON:      BOOL;   // Saída ímpar mapeada no módulo DO Nexto
bOFF:     BOOL;   // Saída par mapeada no módulo DO Nexto
byStatus:   BYTE:= 7; // Retorno da função:
          // 1 = comando inválido
          // 2 = tempo fora da faixa válida (2..255)
          // 3 = comando alterado em tempo de execução
```

```
            // 4 = módulo não respondeu ao comando (ausente)
            // 5 = comando iniciado ou em execução
            // 6 = já existe um comando ativo neste ponto
            // 7 = comando de pulso finalizado com sucesso
END_VAR
VAR
byState:    BYTE;   // Estado do bloco função
udiPulseEnd:  UDINT;  // Instante do fim do pulso
END_VAR

// Máquina de estados do PulsedCommandNexto
CASE byState OF

0: // Estado inicial, pronto para receber comandos:
CASE byCmdType OF

  100:// Apenas retorna o último status

  101:  // Executar pulse ON:
    // Valida a duração do pulso
    IF byPulseTime > 1 THEN
      // Verifica se já existe um comando ativo neste ponto
      IF ptDbpVarAdr^.ON OR ptDbpVarAdr^.OFF THEN
        // Retorna que já existe um comando ativo
        byStatus:= 6;
      ELSE
        // Liga saída CLOSE
        ptDbpVarAdr^.ON:= TRUE;
        ptDbpVarAdr^.OFF:= FALSE;
        // Próximo estado: executar pulse ON
        byState:= byCmdType;
        // Retorna comando iniciado
        byStatus:= 5;
      END_IF
    ELSE
      // Retorna pulso fora da faixa
      byStatus:= 2;
    END_IF

  102:  // Executar pulse OFF
    // Valida a duração do pulso
    IF byPulseTime > 1 THEN
      // Verifica se já existe um comando ativo neste ponto
      IF ptDbpVarAdr^.ON OR ptDbpVarAdr^.OFF THEN
        // Retorna que já existe um comando ativo
        byStatus:= 6;
      ELSE
        // Liga saída TRIP
        ptDbpVarAdr^.ON:= FALSE;
        ptDbpVarAdr^.OFF:= TRUE;
        // Próximo estado: executar pulse OFF
        byState:= byCmdType;
        // Retorna comando iniciado
        byStatus:= 5;
      END_IF
    ELSE
```

```
        // Retorna pulso fora da faixa
        byStatus:= 2;
      END_IF
  ELSE
    // Retorna comando inválido
    byStatus:= 1;
  END_CASE

  // Memoriza instante de término do pulso
  udiPulseEnd:= SysTimeGetMs() + BYTE_TO_UDINT(byPulseTime) * 10;

  101, 102:// Continua execução do pulso ON/OFF
  // A princípio retorna que comando está em execução
  byStatus:= 5;
  // Verifica mudança de parâmetro em execução
  IF byCmdType <> 100 AND byCmdType <> byState THEN
    // Retorna mudança de parâmetro em execução
    byStatus:= 3;
  END_IF
  // Verifica término do pulso
  IF SysTimeGetMs() >= udiPulseEnd THEN
    // Desliga saídas TRIP e CLOSE
    ptDbpVarAdr^.ON:= FALSE;
    ptDbpVarAdr^.OFF:= FALSE;
    // Retorna comando finalizado, apenas se comando não mudou
    IF byCmdType = 100 OR byCmdType = byState THEN
      byStatus:= 7;
    END_IF
    // Próximo estado: inicial
    byState:= 0;
  END_IF

  END_CASE

  // Verifica qualidade do módulo digital (do ponto DBP)
  IF stQuality.VALIDITY <> QUALITY_VALIDITY.VALIDITY_GOOD THEN
  // Desliga saídas TRIP e CLOSE
  ptDbpVarAdr^.ON:= FALSE;
  ptDbpVarAdr^.OFF:= FALSE;
  // Retorna que módulo está ausente
  byStatus:= 4;
  // Próximo estado: inicial
  byState:= 0;
  END_IF

  // Copia estados da saída DBP para as saídas simples
  bON:= ptDbpVarAdr^.ON;
  bOFF:= ptDbpVarAdr^.OFF;
```

5.5.14.3. Parâmetros Gerais

Para as configurações dos *Parâmetros Gerais* de um Servidor 60870-5-104 conforme a figura abaixo seguem os parâmetros da tabela abaixo apresentadas a seguir:

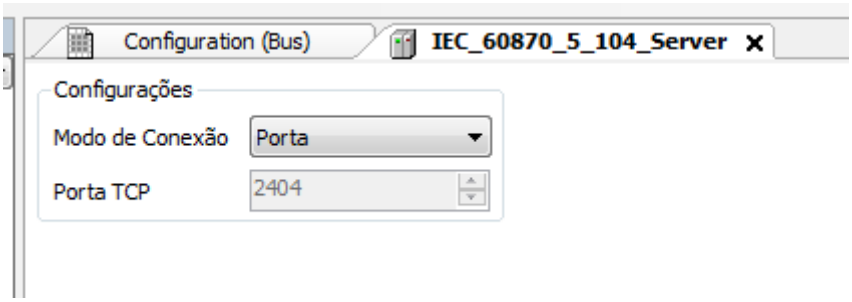


Figura 118: Tela de Parâmetros Gerais do Servidor IEC 60870-5-104

Configuração	Descrição	Padrão	Opções
Modo de Conexão	Configura o modo de conexão com os módulos Cliente Conectados.	Porta	Porta IP
Porta TCP	Define qual o número da porta TCP do CP será utilizada para a comunicação com os módulos Cliente Conectados, caso o campo "Modo de Conexão"esteja selecionado "IP".	2404	1 a 65535

Tabela 140: Configuração dos Parâmetros Gerais do Servidor IEC 60870-5-104

5.5.14.4. Mapeamento dos Dados

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

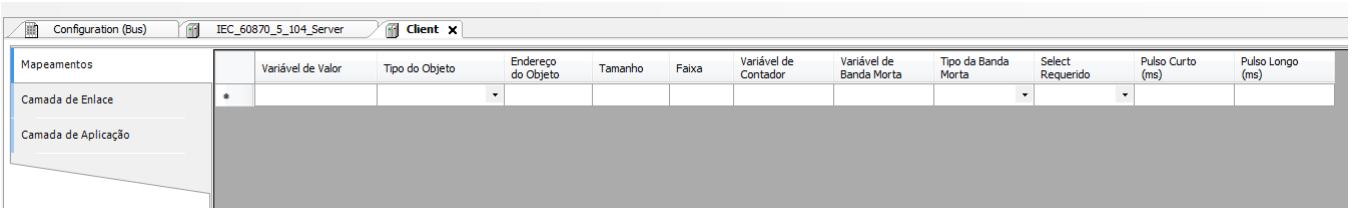


Figura 119: Tela de Mapeamento dos Dados do Servidor IEC 60870-5-104

Configuração	Descrição	Padrão	Opções
Variável de Valor	Nome da variável simbólica	-	Nome de uma variável declarada em um programa ou GVL
Tipo do Objeto	Configuração do tipo do objeto IEC 60870-5-104	-	Single Point Information Double Point Information Step Position Information Measured Value (Normalized) Measured Value (Scaled) Measured Value (Short Floating Point) Integrated Totals Bitstring Information (32 Bits) Single Command Double Command Regulating Step Command Setting Point Command (Normalized) Setting Point Command (Scaled) Setting Point Command (Short Floating Point) Bitstring Command (32 Bits)
Endereço do Objeto	Índice do primeiro ponto do mapeamento IEC 60870-5-104	-	1 a 65535
Tamanho	Especifica a quantidade máxima de dados que um mapeamento IEC 60870-5-104 poderá acessar	-	1 a 86400000
Faixa	Faixa de endereços do dado configurado	-	-
Variável de Contador	Nome da variável simbólica que vai conter o dado da variável contadora	-	Nome de uma variável declarada em um programa, GVL ou módulo contador
Variável de Banda Morta	Nome da variável simbólica que vai conter o dado da banda morta	-	Nome de uma variável declarada em um programa ou GVL
Tipo da Banda Morta	Define o tipo de Banda Morta a ser utilizada no mapeamento	Desabilitada	Absoluta Desabilitada Integrada
Select Requerido	Define se para a execução de um comando é requerido ou não um select prévio	False	True False
Pulso Curto (ms)	Define o tempo do pulso curto de um comando digital IEC 60870-5-104	1000	1 a 86400000

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

Tabela 141: Configuração dos Mapeamentos do Servidor IEC 60870-5-104

Notas:

Variável de Valor: Nome da variável simbólica a ser mapeada. Quando um comando de leitura é enviado, o retorno enviado na resposta está armazenado nesta variável. Quando for um comando de escrita, o valor escrito será armazenado nesta variável. A variável pode ser simples, array ou elemento de array e pode estar em estruturas.

Variável de Contador: Este campo se aplica somente no mapeamento de objetos do tipo *Integrated Totals*, sendo esta a variável contadora, a ser manipulada no processo. Ela deve ter o mesmo tipo e tamanho da variável mapeada na coluna *Variável de Valor*, cujo valor será lido pelo cliente ou reportado ao cliente em caso de eventos.

ATENÇÃO

Quando *Variável de Contador* tem associada uma variável de qualidade, para que esta qualidade seja transferida para a variável congelada no comando de freeze, deve ser associada uma variável de qualidade à variável congelada. Este procedimento pode ser feito através da aba *Pontos Internos*.

Variável de Banda Morta: Este campo se aplica somente à mapeamento de variáveis analógicas de entrada (objetos tipo *Measured Value*). Ela deve ter o mesmo tipo e tamanho da variável mapeada na coluna *Variável de Valor*. Novos valores da variável de banda morta serão considerados somente quando a variável analógica de entrada mudar de valor.

Tipo da Banda Morta: Os tipos de configuração da Banda Morta disponíveis são os seguintes:

Tipo de Função	Configurações	Descrição
Tipo da Banda Morta	Desabilitada	Nesta opção, qualquer modificação de valor num ponto do grupo, por menor que seja, gera evento para este ponto.
	Absoluta	Nesta opção, se o módulo da variação do valor de um ponto do grupo for maior do que o valor configurado pela variável no campo “Banda Morta”, um evento para este ponto é gerado.
	Integrada	Nesta opção, se o módulo da integral da variação do valor de um ponto do grupo for maior do que o valor configurado pela variável no campo “Banda Morta”, um evento para este ponto é gerado. O intervalo de integração é de um segundo.

Tabela 142: Tipos de Banda Morta dos Mapeamentos do Servidor IEC 60870-5-104

Pulso Curto e Pulso Longo: Ao definir o tempo de duração dos pulsos curtos e longos deve se levar em consideração os limites suportados pelo dispositivo que irá tratar o comando. Por exemplo, caso o destino seja um cartão de saída, o que não é suportado de forma nativa pela Série Nexto, deve se verificar na CT do módulo quais os tempos mínimos e máximos, bem como a resolução, para a execução de comandos pulsados.

5.5.14.5. Camada de Enlace

Para a configuração dos parâmetros da camada de enlace do Servidor IEC 60870-5-104, visualizada na figura abaixo, seguem os parâmetros descritos na tabela abaixo:

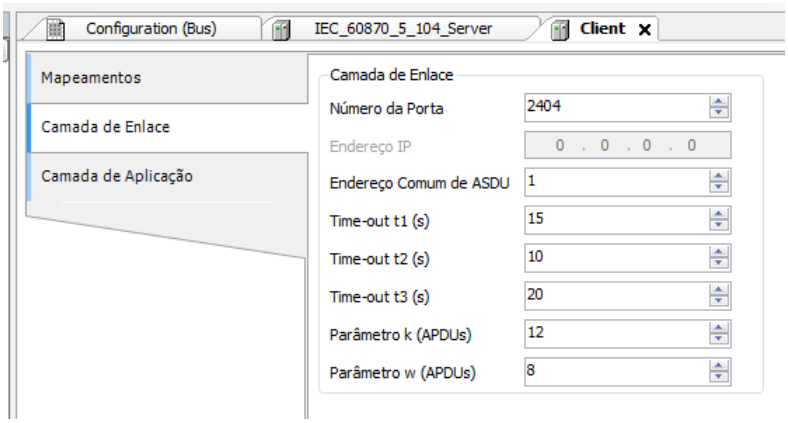


Figura 120: Tela de Configuração da Camada de Enlace do Servidor IEC 60870-5-104

Parâmetro	Descrição	Padrão	Opções
Número da Porta	Endereço da porta listen para conexão do cliente. Utilizado quando a conexão do cliente não é por IP	2404	1 a 65535
Endereço IP	IP do cliente conectado, utilizado quando a conexão do cliente é por IP	0.0.0.0	1.0.0.1 a 223.255.255.254
Endereço Comum de ASDU	Endereço IEC 60870-5-104 do cliente conectado	1	1 a 65534
Time-out t1 (s)	Período de tempo (em segundos) que o dispositivo aguarda o recebimento de uma mensagem de confirmação após o envio de uma mensagem APDU do tipo I ou U (dados) antes de fechar a conexão	15	1 a 180
Time-out t2 (s)	Período de tempo (em segundos) que o dispositivo aguarda para enviar uma mensagem supervisória (S-Frame) confirmando o recebimento de frames de dados	10	1 a 180
Time-out t3 (s)	Período de tempo (em segundos) em que será enviada uma mensagem para teste do link caso não haja nenhuma transmissão por ambos os lados	20	1 a 180

Parâmetro	Descrição	Padrão	Opções
Parâmetro k (APDUs)	Número máximo de mensagens de dados (I-Frame) transmitidas e não confirmadas	12	1 a 12
Parâmetro w (APDUs)	Número máximo de mensagens de dados (I-Frame) recebidas e não confirmadas	8	1 a 8

Tabela 143: Configuração da Camada de Enlace do Servidor IEC 60870-5-104

Nota:

Os campos *Time-out t1 (s)*, *Time-out t2 (s)* e *Time-out t3 (s)* são dependentes entre si e devem ser configurados de tal forma que *Time-out t1 (s)* seja maior do que *Time-out t2 (s)* e *Time-out t3 (s)* seja maior do que *Time-out t1 (s)*. Se alguma destas regras não for respeitada, mensagens de erro serão exibidas na compilação do projeto.

ATENÇÃO

Para links de comunicação lentos (exemplo: comunicação via satélite), deve se ajustar adequadamente os parâmetros *Time-out t1 (s)*, *Time-out t2 (s)* e *Time-out t3 (s)* como por exemplo, dobrando os valores padrões desses campos.

5.5.14.6. Camada de Aplicação

Para a configuração da camada de aplicação do Servidor IEC 60870-5-104, visualizada na figura abaixo, seguem os parâmetros descritos na tabela abaixo:

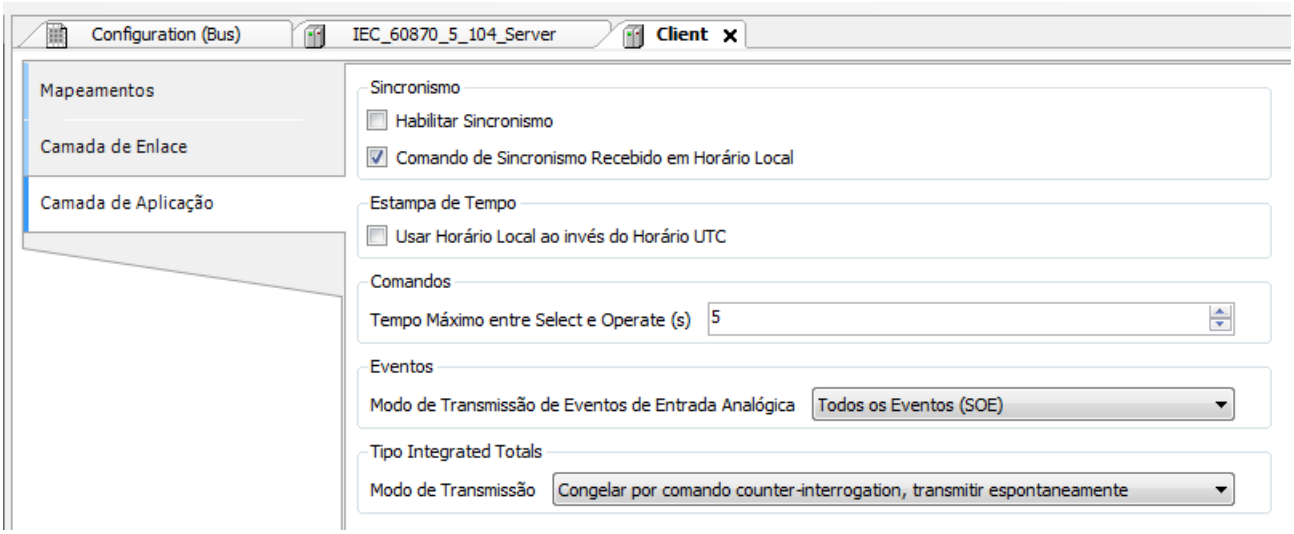


Figura 121: Tela de Configuração da Camada de Aplicação do Servidor IEC 60870-5-104

Parâmetro	Descrição	Padrão	Opções
Habilitar Sincronismo	Opção para Habilitar/Desabilitar a solicitação de sincronismo de tempo	Desabilitado	Desabilitado Habilitado
Comando de Sincronismo Recebido em Horário Local	Opção para Habilitar/Desabilitar o tratamento do comando de sincronização em horário local	Habilitado	Desabilitado Habilitado
Usar Horário Local ao invés do Horário UTC	Opção para Habilitar/Desabilitar a estampa de tempo em horário local para eventos	Desabilitado	Desabilitado Habilitado
Tempo Máximo entre Select e Operate (s)	Período de tempo no qual o comando de seleção permanecerá ativo (inicia a contagem a partir da confirmação da recepção do comando de seleção) aguardando o comando de Operate	5	1 a 180
Modo de Transmissão de Eventos de Entrada Analógica	Modo de transmissão dos eventos de entradas analógicas	Todos os Eventos (SOE)	Todos os Eventos (SOE) Evento Mais Recente
Modo de Transmissão	Modo de transmissão dos contadores congelados (integrated totals).	Congelar por comando counter-interrogation, transmitir espontaneamente	Congelar por comando counter-interrogation, transmitir espontaneamente Congelar e transmitir por comando counter-interrogation

Tabela 144: Configuração da Camada de Aplicação do Servidor IEC 60870-5-104

Notas:

Habilitar Sincronismo: Uma vez habilitado, permite o Servidor IEC 60870-5-104 ajustar o relógio da UCP quando recebido um comando de sincronização.

Comando de Sincronismo Recebido em Horário Local: Quando habilitado, o IEC 60870-5-104 Servidor ajusta o relógio da UCP tratando o horário recebido no comando de sincronização como horário local. Caso contrário esse horário é considerado UTC.

Usar Horário Local ao invés do Horário UTC: Uma vez habilitado, a estampa de tempo dos eventos gerados pelo IEC 60870-5-104 Servidor será enviada conforme o horário local da UCP.

ATENÇÃO

Quando a opção de sincronismo de tempo for marcada em mais de um servidor, os tempos recebidos nos diferentes servidores serão sobrescritos no relógio do sistema em um curto espaço de tempo, podendo causar comportamentos indesejáveis devido a atrasos no tempo de propagação das mensagens e carga do sistema.

Modo de Transmissão de Eventos de Entrada Analógica: Os modos de transmissão dos eventos de entradas analógicas disponíveis são os seguintes:

Tipo de Função	Configurações	Descrição
Modo de Transmissão de Eventos de Entrada Analógica	Todos os Eventos (SOE)	São enviados todos os eventos analógicos gerados.
	Evento Mais Recente	É enviado somente o evento mais recente de cada ponto analógico.

Tabela 145: Modos de Transmissão dos Eventos de Entradas Analógicas do Servidor IEC 60870-5-104

Modo de Transmissão: Os modos de transmissão dos contadores congelados (*integrated totals*) disponíveis são os seguintes:

Tipo de Função	Configurações	Descrição
Modo de Transmissão	Congelar por comando counter-interrogation, transmitir espontaneamente	Equivale ao Modo D de aquisição de contadores (integrated totals) definido pela norma IEC 60870-5-101. Neste modo, comandos de interrogação de contadores (freeze) da estação de controle realizam o congelamento dos contadores. Caso os valores congelados tenham se modificado, eles são reportados através de eventos.
	Congelar e transmitir por comando counter-interrogation	Equivale ao Modo C de aquisição de contadores (integrated totals) definido pela norma IEC 60870-5-101. Neste modo, comandos de interrogação de contadores (freeze) da estação de controle realizam o congelamento dos contadores. Comandos subsequentes de interrogação dos contadores (read) são enviados pela estação de controle para obter os valores congelados.

Tabela 146: Modos de Transmissão dos Contadores Congelados do Servidor IEC 60870-5-104

ATENÇÃO

A norma IEC 60870-5-104, seção *Transmission control using Start/Stop*, prevê a utilização de comandos *STARTDT* e *STOPDT* para controle do tráfego de dados entre o cliente e o servidor, utilizando conexões simples ou múltiplas conexões. Apesar do Nexto suportar tais comandos, a utilização dos mesmos não é recomendada para controlar a transmissão dos dados, principalmente com CPs redundantes, pois tais comandos não são sincronizados entre os dois CPs. Em vez de utilizar múltiplas conexões entre o cliente e o servidor Nexto, sugere-se a utilização do recurso de NIC Teaming para prover canais Ethernet (fisicamente) redundantes e preservar os recursos da UCP (centros de controle por UCP).

5.5.14.7. Diagnósticos do Servidor

Os diagnósticos do protocolo Servidor IEC 60870-5-104, são armazenados em variáveis do tipo *T_DIAG_IEC104_SERVER_1* as quais estão descritas na tabela abaixo:

Variável de diagnóstico do tipo T_DIAG_IEC104_SERVER_1.*	Tamanho	Descrição
Bits de comando, reiniciados automaticamente:		
tCommand.bStop	BOOL	Desabilitar Driver
tCommand.bStart	BOOL	Habilitar o Driver
tCommand.bDiag_01_Reserved	BOOL	Reservado
tCommand.bDiag_02_Reserved	BOOL	Reservado
tCommand.bDiag_03_Reserved	BOOL	Reservado
tCommand.bDiag_04_Reserved	BOOL	Reservado
tCommand.bDiag_05_Reserved	BOOL	Reservado
tCommand.bDiag_06_Reserved	BOOL	Reservado
Diagnósticos:		
tClient_X.bRunning	BOOL	IEC 60870-5-104 Server está sendo executado
tClient_X.eConnectionStatus. CLOSED	ENUM(BYTE)	Canal de comunicação fechado. Servidor não irá aceitar solicitação de conexão. Valor do ENUM (0)
tClient_X.eConnectionStatus. LISTENING		Servidor está escutando a porta configurada, e não existem clientes conectados. Valor do ENUM (1)
tClient_X.eConnectionStatus. CONNECTED		Cliente conectado. Valor do ENUM (2)
tClient_X.tQueueDiags. bOverflow	BOOL	Fila do cliente está com estouro
tClient_X.tQueueDiags. wSize	WORD	Tamanho da fila configurada
tClient_X.tQueueDiags. wUsage	WORD	Número de eventos na fila
tClient_X.tQueueDiags. dwReserved_0	DWORD	Reservado
tClient_X.tQueueDiags. dwReserved_1	DWORD	Reservado
tClient_X.tStats.wRXFrames	WORD	Número de quadros (frames) recebidos
tClient_X.tStats.wTXFrames	WORD	Número de quadros (frames) enviados
tClient_X.tStats.wCommErrors	WORD	Contador de erros de comunicação incluindo erros na Camada Física, Camada de Enlace e Camada de Transporte.
tClient_X.tStats.dwReserved_0	DWORD	Reservado
tClient_X.tStats.dwReserved_1	DWORD	Reservado

Tabela 147: Diagnósticos do Servidor IEC 60870-5-104

5.5.14.8. Qualificador dos Comandos

A norma IEC 60870-5-104 prevê quatro diferentes qualificadores de comandos para os objetos *Single Command*, *Double Command* e *Regulating Step Command*, todos suportados pelo Servidor do Nexto.

Cada tipo de objeto tem um comportamento específico para cada qualificador de comando, como pode ser observado na tabela apresentada a seguir.

Qualificador	Tipo do objeto do protocolo IEC 60870-5-104		
	Single Command	**Double Command**	**Regulating Step Command**
No additional definition (padrão)	Mesmo comportamento do qualificador persistente.	Mesmo comportamento do qualificador de pulso curto.	Mesmo comportamento do qualificador de pulso curto.
Short pulse duration (pulso curto)	Requer interceptação do comando para tratamento pela aplicação, caso contrário irá retornar uma mensagem de confirmação negativa (falha).	Requer interceptação do comando para tratamento pela aplicação, caso contrário irá retornar uma mensagem de confirmação negativa (falha).	Requer interceptação do comando para tratamento pela aplicação, caso contrário irá retornar uma mensagem de confirmação negativa (falha).
Long pulse duration (pulso longo)			
Persistent output (persistente)	A saída será ligada ou desligada, e assim permanecerá até novo comando, conforme valor (ON ou OFF) comandado pelo cliente.		

Tabela 148: Qualificador dos Comandos do Servidor IEC 60870-5-104

Nota:

Interceptação do comando: Para maiores informações sobre a interceptação de comandos de clientes IEC 60870-5-104, consultar a seção Interceptação de Comandos Oriundos do Centro de Controle, implementada através do bloco de função *CommandReceiver*.

5.5.15. PROFINET Controller

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

5.6. Desempenho de Comunicação

5.6.1. Servidor MODBUS

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

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

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

Por exemplo, para uma tarefa MODBUS Ethernet Servidor com um ciclo de 50 ms, em uma aplicação executada por 60 ms a cada 100 ms, o servidor conseguirá executar apenas um ciclo entre cada ciclo da aplicação. Por outro lado, com a mesma aplicação, sendo executada por 60 ms, mas com um intervalo de 500 ms, o MODBUS terá um desempenho melhor, pois enquanto a aplicação não estiver em execução, o mesmo estará sendo executado a cada 50 ms e apenas a cada ciclo da MainTask ele irá demorar mais para ser executado. Para estes casos, o pior desempenho será a soma do tempo de execução da aplicação do usuário com o tempo do ciclo da tarefa MODBUS.

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

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

5.6.1.1. Interfaces Locais da UCP

Para um dispositivo MODBUS Ethernet Servidor, podemos afirmar que ele é capaz de responder a um número x de requisições por segundo, ou seja, será capaz de transferir n bytes por segundo, dependendo do tamanho de cada requisição. Quanto menor for o ciclo da tarefa do Servidor MODBUS, maior será o impacto do número de conexões em sua taxa de resposta. Porém, para tempos de ciclo menores que 20 ms este impacto não é linear, devendo ser consultada a tabela abaixo para informações.

A tabela abaixo exemplifica o número de requisições respondidas por um Servidor MODBUS inserido em uma interface local da UCP, em função do tempo de ciclo configurado para a tarefa e do número de conexões ativas efetuando requisições:

Número de Conexões	Requisições respondidas por segundo com tempo de ciclo da tarefa MODBUS em 5 ms	Requisições respondidas por segundo com tempo de ciclo da tarefa MODBUS em 10 ms	Requisições respondidas por segundo com tempo de ciclo da tarefa MODBUS em 20 ms
1 Conexão	185	99	50
2 Conexões	367	197	100
4 Conexões	760	395	200
7 Conexões	1354	695	350
10 Conexões	1933	976	500

Tabela 149: Taxa de Comunicação de um Servidor MODBUS em uma Interface Local

ATENÇÃO

Os desempenhos de comunicação mencionados neste capítulo são exemplos utilizando uma UCP apenas com um dispositivo MODBUS TCP Servidor, sem nenhuma lógica na aplicação que possa atrasar a comunicação. Portanto, estes desempenhos devem ser tomados como máximos.

Já para tempos de ciclo iguais ou maiores do que 20 ms, o crescimento da taxa de respostas é linear, podendo ser calculada através da fórmula:

N = C 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.

5.6.1.2. Interfaces Remotas

O desempenho de um dispositivo MODBUS Servidor em uma interface Ethernet remota é parecido com o desempenho nas interfaces locais da UCP.

Porém, devido ao tempo de comunicação entre a UCP e os módulos, o máximo desempenho será limitado. Para apenas uma conexão aberta, o número de respostas é limitado em, no máximo, 18 respostas por segundo. Com mais conexões abertas, o número de respostas aumentará linearmente, exatamente conforme para as interfaces locais, porém sendo limitado a, no máximo, 90 respostas por segundo. Portanto, para uma interface Ethernet remota teremos as seguintes formas para calcular seu desempenho:

Para T $\leq$ 55 ms usa-se:

N = C x (18,18 – (18,18 / (0,055 x 1000)))

E para T $\geq$ 55 ms usa-se:

N = C x (Z – (Z / (T x 1000)))

Onde N é o número médio de respostas por segundo, C é o número de conexões e T é igual ao tempo de ciclo da tarefa MODBUS (em segundos).

Deve-se prestar atenção ao fato de que o desempenho máximo de um dispositivo MODBUS Servidor em uma interface Ethernet remota será de 90 respostas de requisições por segundo.

5.6.2. Servidor OPC DA

Para os testes de desempenho da comunicação com Servidor OPC DA foram criados projetos para o CP declarando variáveis do tipo INT e separando-as em POUs com 1.000 variáveis cada. Em todos os cenários foram utilizados projetos com perfil *Simples* e o Intervalo da MainTask configurado em 100 ms. Para realizar a comunicação, foi utilizado o atributo *attribute 'symbol' := 'readwrite'*, a fim de disponibilizar os dados no Servidor OPC DA. Também é importante ressaltar que as medições foram feitas com o software MasterTool desconectado da UCP e com a duração da MainTask ajustada para consumir 5%, 50% e 80% do Intervalo configurado, como apresentado na tabela abaixo.

No ponto de vista do OPC DA Cliente, foi utilizado um driver num sistema SCADA. O tempo de atualização configurado foi de 50 ms. Os resultados do desempenho nestas condições estão descritos na tabela abaixo.

Número Total de variáveis no projeto do CP	**Tempo de atualização das variáveis no OPC DA Cliente**		
	5% da UCP Ocupada	**50% da UCP Ocupada**	**80% da UCP Ocupada**
1.000	600 ms	800 ms	1400 ms
2.000	800 ms	900 ms	2800 ms
5.000	1000 ms	2000 ms	6500 ms
10.000	2000 ms	4000 ms	13700 ms
15.000	3200 ms	6400 ms	20000 ms
20.000	4000 ms	8100 ms	25000 ms

Tabela 150: Taxa de Comunicação de um Servidor OPC DA

5.6.3. Servidor OPC UA

O manual do OPC UA Server MU214609 analisa o desempenho da comunicação OPC UA com maiores detalhes, inclusive abordando o consumo de banda de comunicação Ethernet. Este manual também aborda conceitos sobre o funcionamento do protocolo OPC UA.

5.6.4. Servidor IEC 60870-5-104

O driver Servidor IEC 60870-5-104 é executado pela UCP da mesma forma como os outros drivers de comunicação Servidores, isto é, em segundo plano, com uma prioridade abaixo da aplicação de usuário e de forma cíclica. A tarefa deste driver especificamente executa a cada 50 ms, e basta 1 ciclo de execução do driver para processar e responder requisições. Contudo, por se tratar de uma tarefa de baixa prioridade, não é garantido que consiga executar com esta periodicidade pois depende do percentual de UCP livre (diferença entre o intervalo da MainTask e tempo que a aplicação do usuário leva para ser executada) e também da concorrência com tarefas de outros protocolos configurados na UCP.

Para auxiliar na compreensão do desempenho do driver IEC 60870-5-104 Servidor, a seguir é apresentado o resultado de testes realizados com um simulador Cliente IEC 60870-5-104 conectado a uma NX3030 com um driver IEC 60870-5-104 Servidor. A base de dados configurada era composta de 900 pontos digitais e 100 pontos analógicos (todos com qualidade e estampa de tempo), e o tempo de execução da MainTask estava em 70 ms (intervalo de 100 ms).

- Tempo para completar um comando de interrogação geral: menos de um segundo
- Tempo para transferir 900 eventos digitais + 100 eventos analógicos: 6 segundos

5.7. Desempenho do Sistema

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

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

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

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

ATENÇÃO

Para tempos de ciclo muito altos (tipicamente tempos maiores que 300 ms), mesmo que o percentual de 80% seja respeitado, pode haver uma diminuição no tempo de resposta do visor e da tecla de diagnóstico. Caso o percentual de 80% não seja respeitado e o tempo de execução da tarefa de usuário se aproxime ou exceda o intervalo configurado da MainTask, o visor e a tecla de diagnóstico podem não responder, uma vez que sua prioridade na execução do sistema é menor do que as tarefas de usuário. Caso uma aplicação com erros seja carregada na UCP, pode ser necessário reinicializá-la sem carregar esta aplicação, como descrito na seção Log de Sistema.

ATENÇÃO

Os logs de sistema das UCPs da série Nexto, a partir da versão de firmware 1.4.0.33 são recarregados no caso de uma reinicialização da UCP ou por uma reinicialização do *Runtime System*, isto é, será possível visualizar os logs mais antigos quando uma dessas condições ocorrer.

5.7.1. Escaneamento de E/S

Para um projeto que utilize de módulos de E/S digitais, estando estes inseridos no barramento e declarados no projeto, o tempo da MainTask irá aumentar de acordo com o número de módulos. A tabela abaixo exemplifica o tempo médio que é acrescido à MainTask:

Módulos declarados no barramento	Acréscimo de tempo no ciclo da MainTask (μs)
5	300
10	700
20	1000

Tabela 151: Tempo de Escaneamento de E/S

Em projetos que utilizem E/S remotas, como, por exemplo, utilizando o módulo Mestre PROFIBUS DP NX5001, o manual do respectivo módulo deve ser consultado para informações sobre desempenho e influências do módulo na execução das tarefas de usuário.

5.7.2. Cartão de Memória

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

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

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

5.8. Relógio RTC

As UCPs possuem um relógio interno que pode ser utilizado através da biblioteca *NextoStandard.lib*. Essa biblioteca é carregada automaticamente durante a criação de um novo projeto (para realizar o procedimento de inserção de uma biblioteca, consultar a seção Bibliotecas). A figura abaixo mostra os blocos e funções disponibilizados:

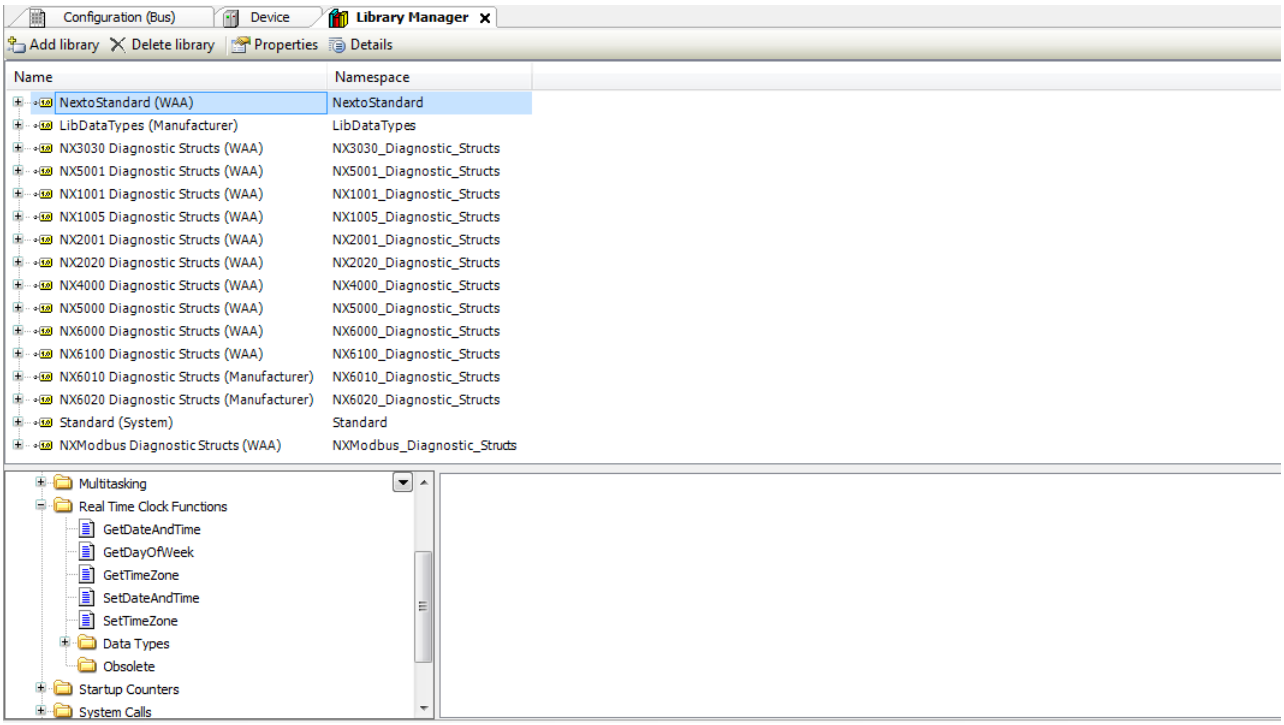


Figura 122: Blocos e Funções de Escrita e Leitura do Relógio

ATENÇÃO

Blocos funcionais para Leitura e Escrita do RTC, disponíveis em versões anteriores a 2.00 do MasterTool IEC XE tornaram-se obsoletos da versão 2.00 em diante, os blocos que ficaram obsoletos são:
NextoGetDateAndTime, *NextoGetDateAndTimeMs*, *NextoGetTimeZone*, *NextoSetDateAndTime*, *NextoSetDateAndTimeMs* e *NextoSetTimeZone*.

5.8.1. Blocos Funcionais e Funções para Leitura e Escrita do RTC

Entre outros blocos funcionais e funções, existem alguns muito importantes utilizados para a leitura do relógio (*GetDateAndTime*, *GetDayOfWeek* e *GetTimeZone*) e para configurar novos valores de data e hora (*SetDateAndTime* e *SetTimeZone*). Estas funções utilizam sempre o horário local, isto é, levam em consideração o valor definido para o *Fuso Horário*.

Abaixo, descrevem-se os procedimentos utilizados para configurar as funções.

5.8.1.1. Funções de Leitura do RTC

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

5.8.1.1.1. GetDateAndTime

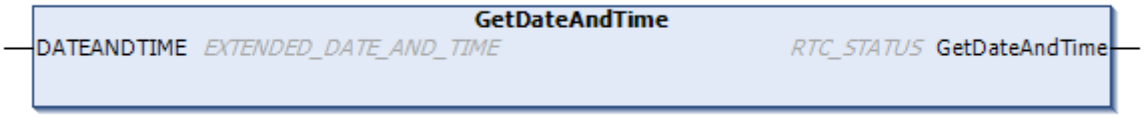


Figura 123: Leitura da Data e Hora

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 161.

Tabela 152: Parâmetros de Entrada GetDateAndTime

Parâmetros de saída	Tipo	Descrição
GETDATEANDTIME	RTC_STATUS	Retorna o estado de erro da função, ver Tabela 163.

Tabela 153: 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
```

5.8.1.1.2. GetTimeZone

A função a seguir faz a leitura das configurações de fuso horário, esta função está diretamente relacionada com o tempo de fuso horário configurado no serviço de sincronismo do SNTP:

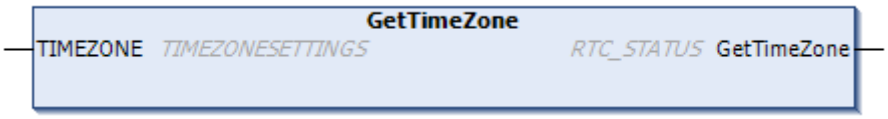


Figura 124: Leitura das Configurações de Fuso Horário

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

Tabela 154: Parâmetros de Entrada GetTimeZone

Parâmetros de saída	Tipo	Descrição
GetTimeZone	RTC_STATUS	Retorna o estado de erro da função, ver Tabela 163.

Tabela 155: 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
```

5.8.1.1.3. GetDayOfWeek

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

Figura 125: Leitura do Dia da Semana

Parâmetros de saída	Tipo	Descrição
GetDayOfWeek	DAYS_OF_WEEK	Retorna o dia da semana. Ver Tabela 162.

Tabela 156: Parâmetros de Saída GetDayOfWeek

Quando chamada, a função lerá o dia da semana e preencherá a estrutura *DAYS_OF_WEEK*.
Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
DayOfWeek : DAYS_OF_WEEK;
END_VAR
------------------------------------------------------------------------------
DayOfWeek := GetDayOfWeek();
```

5.8.1.2. Funções de Escrita do RTC

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

5.8.1.2.1. SetDateAndTime

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

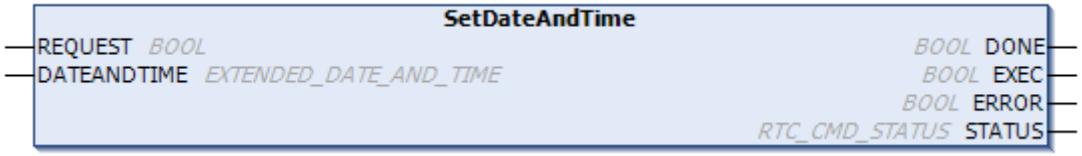


Figura 126: Ajuste de Data e Hora em Milissegundos

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 161.

Tabela 157: Parâmetros de Entrada SetDateAndTime

Parâmetros de saída	Tipo	Descrição
DONE	BOOL	Essa variável, quando verdadeira, indica que a ação foi completada.
EXEC	BOOL	Essa variável, quando verdadeira, indica que a função está processando os valores.
ERROR	BOOL	Essa variável, quando verdadeira, indica que ocorreu algum erro durante a Escrita.
STATUS	RTC_STATUS	Retorna o erro ocorrido durante a configuração, ver Tabela 163.

Tabela 158: Parâmetros de Saída SetDateAndTime

Quando ocorrer uma borda de subida na entrada *REQUEST*, o bloco funcional irá escrever o novo valor *DATEANDTIME* no relógio. Caso a escrita seja realizada com sucesso, a saída *DONE* será igual a *TRUE*. Caso contrário, a saída *ERROR* será igual a *TRUE* e o erro será apresentado na variável *STATUS*.

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
SetDateAndTime : SetDateAndTime;
xRequest : BOOL;
DateAndTime : EXTENDED_DATE_AND_TIME;
xDone : BOOL;
xExec : BOOL;
xError : BOOL;
xStatus : RTC_STATUS;
END_VAR
-------------------------------------------------------------------------------
IF xRequest THEN
  SetDateAndTime.REQUEST:=TRUE;
  SetDateAndTime.DATEANDTIME:=DateAndTime;
  xRequest:= FALSE;
END_IF
SetDateAndTime();
SetDateAndTime.REQUEST:=FALSE;
IF SetDateAndTime.DONE THEN
  xExec:=SetDateAndTime.EXEC;
  xError:=SetDateAndTime.ERROR;
  xStatus:=SetDateAndTime.STATUS;
END_IF
```

ATENÇÃO

Se o usuário tentar escrever valores de hora fora do intervalo do RTC, os valores serão convertidos para valores válidos, desde que não ultrapasse a faixa válida de 01/01/2000 até 31/12/2035. Por exemplo, se o usuário tentar escrever o valor 2000 ms, o mesmo será convertido para 2 segundos, se escrever o valor 100 segundos, o mesmo será convertido para 1 min e 40 segundos. Se escrever o valor de 30 horas, o mesmo será convertido para 1 dia e 6 horas, e assim por diante.

5.8.1.2.2. *SetTimeZone*

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

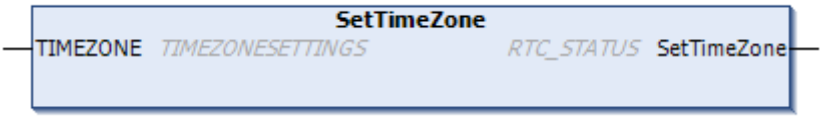


Figura 127: Ajuste de Fuso Horário

Parâmetros de entrada	Tipo	Descrição
TIMEZONE	TIMEZONESETTINGS	Estrutura com o valor de fuso horário a ser configurado. Ver Tabela 164.

Tabela 159: Parâmetros de Entrada SetTimeZone

Parâmetros de saída	Tipo	Descrição
SetTimeZone	RTC_STATUS	Retorna o erro ocorrido durante a leitura/configuração. Ver Tabela 163.

Tabela 160: Parâmetros de Saída SetTimeZone

Quando chamada, a função, irá configurar o valor de *TIMEZONE* como a nova configuração de fuso horário do sistema. O resultado da configuração é retornado pela função.

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
Status : RTC_STATUS;
TimeZone : TIMEZONESETTINGS;
xWrite : BOOL;
END_VAR
------------------------------------------------------------------------
//FB SetTimeZone
IF (xWrite = TRUE) THEN
Status := SetTimeZone(TimeZone);
  IF Status = RTC_STATUS.NO_ERROR THEN
    xWrite := FALSE;
  END_IF
END_IF
```

ATENÇÃO

Para realizar o acerto do relógio, devem-se utilizar valores de hora e datas dentro da seguinte faixa válida: 00:00:00 horas de 01/01/2000 até 23:59:59 horas de 31/12/2035, caso contrário, será reportado um erro através do parâmetro de saída *STATUS*. Para maiores detalhes do parâmetro de saída *STATUS*, consultar a seção RTC_STATUS.

5.8.2. Estruturas de Dados do RTC

Os blocos funcionais de leitura e configuração do RTC das UCPs da série Nexto utilizam as seguintes estruturas de dados na sua configuração:

5.8.2.1. EXTENDED_DATE_AND_TIME

Esta estrutura é utilizada para armazenar a data do RTC quando utilizados os blocos funcionais para leitura/configuração da data com precisão de milissegundos e é descrita na tabela abaixo:

Estrutura	Tipo	Variável	Descrição
EXTENDED_DATE_AND_TIME	BYTE	byDayOfMonth	Armazena o dia do mês da data configurada.
	BYTE	ByMonth	Armazena o mês da data configurada.
	WORD	wYear	Armazena o ano da data configurada.
	BYTE	byHours	Armazena a hora da data configurada.
	BYTE	byMinutes	Armazena os minutos da data configurada.
	BYTE	bySeconds	Armazena os segundos da data configurada.
	WORD	wMilliseconds	Armazena os milissegundos da data configurada.

Tabela 161: Estrutura EXTENDED_DATE_AND_TIME

5.8.2.2. DAYS_OF_WEEK

Esta estrutura é utilizada para armazenar o dia da semana quando utilizada a função para leitura do dia da semana:

Enumerador	Valor	Descrição
DAYS_OF_WEEK	0	INVALID_DAY
	1	SUNDAY
	2	MONDAY
	3	TUESDAY
	4	WEDNESDAY
	5	THURSDAY
	6	FRIDAY
	7	SATURDAY

Tabela 162: Estrutura DAYS_OF_WEEK

5.8.2.3. RTC_STATUS

Este enumerador é utilizado para retornar o tipo de erro na configuração ou leitura do RTC e é descrito na tabela abaixo:

Enumerador	Valor	Descrição
RTC_STATUS	NO_ERROR (0)	Não há erro.
	UNKNOWN_COMMAND (1)	Comando desconhecido.
	DEVICE_BUSY (2)	Dispositivo está ocupado.
	DEVICE_ERROR (3)	Dispositivo com erro.
	ERROR_READING_OSF (4)	Erro na leitura do sinalizador de data e hora válidas.
	ERROR_READING_RTC (5)	Erro na leitura da data e hora.
	ERROR_WRITING_RTC (6)	Erro na escrita da data e hora.
	ERROR_UPDATING_SYSTEM _TIME (7)	Erro na atualização de data e hora do sistema.
	INTERNAL_ERROR (8)	Erro interno.
	INVALID_TIME (9)	Data e hora inválidas.
	INPUT_OUT_OF_RANGE (10)	Fora do limite de Data e hora válidas para o sistema.
	SNTP_NOT_ENABLE (11)	Erro gerado quando o serviço SNTP não está habilitado e é feita uma tentativa de ler ou modificar o fuso horário.

Tabela 163: RTC_STATUS

5.8.2.4. TIMEZONESETTINGS

Esta estrutura é utilizada para armazenar o valor do fuso horário nas requisições de leitura/configuração dos blocos funcionais do RTC e é descrita na tabela abaixo:

Estrutura	Tipo	Variável	Descrição
TIMEZONESETTINGS	INT	iHour	Hora do fuso horário configurado.
	INT	iMinutes	Minuto do fuso horário configurado.

Tabela 164: TIMEZONESETTINGS

Nota:

Blocos funcionais de escrita e leitura de data e hora: Bibliotecas diferentes da *NextoStandard*, que tenham blocos funcionais ou funções que possam fazer acesso de leitura e escrita da data e hora no sistema, não são indicadas. A biblioteca *NextoStandard* possui as interfaces adequadas para escrever e ler a data e hora do sistema adequadamente e informar os diagnósticos corretos.

5.9. Memória de Arquivos de Usuário

As UCPs da Série Nexto possuem uma área de memória destinada ao armazenamento de dados de uso geral, ou seja, o usuário poderá gravar diversos arquivos de qualquer formato na memória da UCP. Esta área de memória pode variar de acordo com o modelo de UCP utilizado (consultar Memória).

Para usar esta área, o usuário deverá acessar um projeto no software MasterTool IEC XE e clicar na *Árvore de Dispositivos*, localizada à esquerda do programa. Deverá dar dois cliques sobre o item *Device* e, após selecionar a UCP na aba *Configurações de Comunicação* que será aberta, selecionar a aba *Arquivos* e clicar em *Atualizar*, tanto na coluna de arquivos do computador (esquerda), como na coluna de arquivos da UCP selecionada (direita), conforme mostram as indicações da figura abaixo.

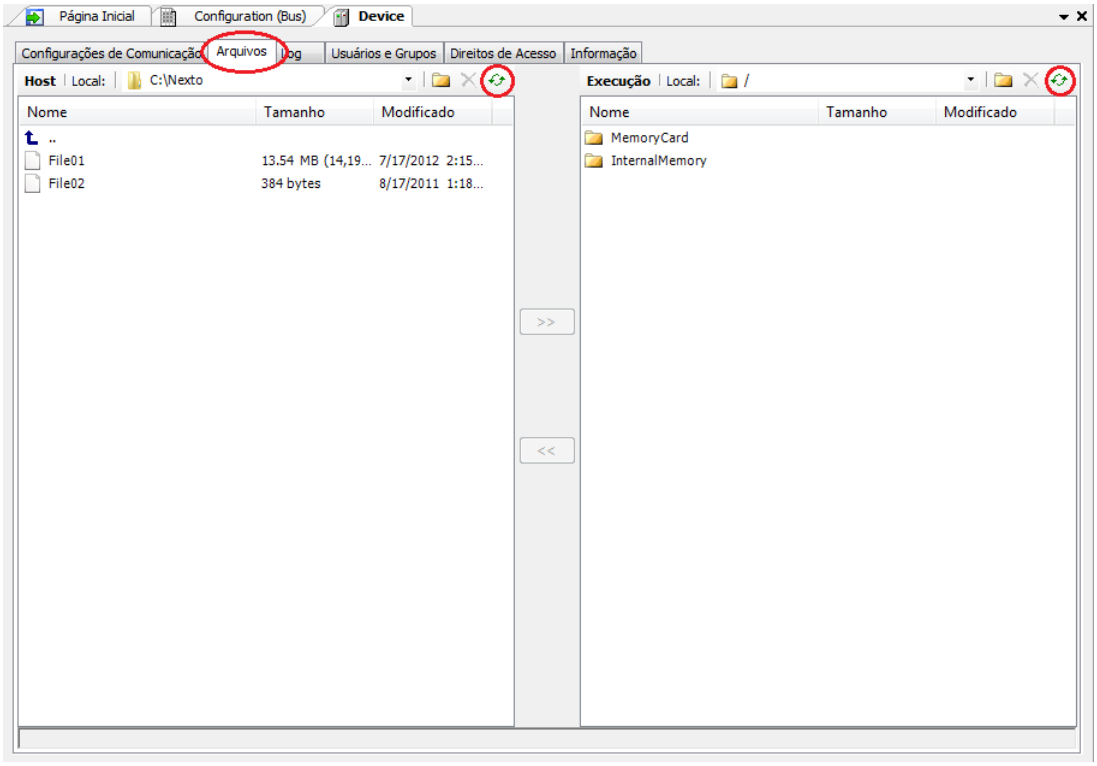


Figura 128: Acesso aos Arquivos de Usuário

Após atualizar a coluna de arquivos da UCP, será exibido o diretório raiz de arquivos armazenados na UCP e poderá ser selecionada a pasta para onde os arquivos serão transferidos. A pasta *"InternalMemory"* é uma pasta padrão, a ser utilizada para armazenar arquivos na memória interna da UCP, uma vez que não é possível transferir arquivos para o diretório raiz. Caso seja necessário, podem ser criadas outras pastas no diretório raiz ou subpastas dentro da pasta *"InternalMemory"*.

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

ATENÇÃO

No caso em que o cartão de memória seja inserido após a inicialização da UCP, será requisitado um usuário e senha para realizar o acesso e/ou transferências de arquivos do MasterTool IEC XE para o cartão de memória ou vice-versa. O usuário padrão com privilégios para acesso à UCP é *"Owner"* e a senha padrão desse usuário é *"Owner"*.

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

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

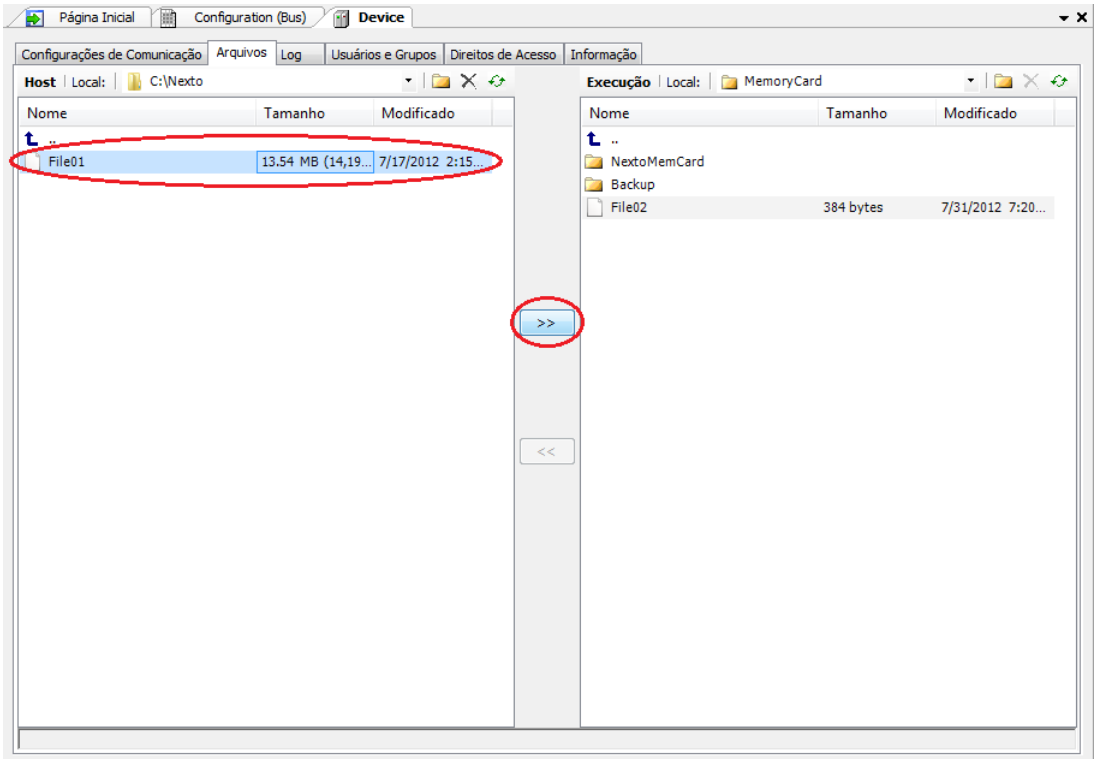


Figura 129: Transferindo Arquivos

ATENÇÃO

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

Caso seja necessário transferir documentos da UCP para o microcomputador em que está instalado o software MasterTool IEC XE o usuário deve realizar um procedimento muito semelhante ao anterior, ou seja, selecionando o arquivo na coluna da direita e pressionar o botão "«", localizado no centro da tela.

Além disso, o usuário possui algumas opções de operação da área de armazenamento de arquivos, são elas:

- Novo diretório : permite a criação de uma nova pasta na área de memória de usuário.
- Excluir item : permite a exclusão de arquivos nos diretórios da área de memória de usuário.
- Atualizar : permite atualizar, na tela do MasterTool IEC XE, os arquivos presentes na memória de usuário e no microcomputador.

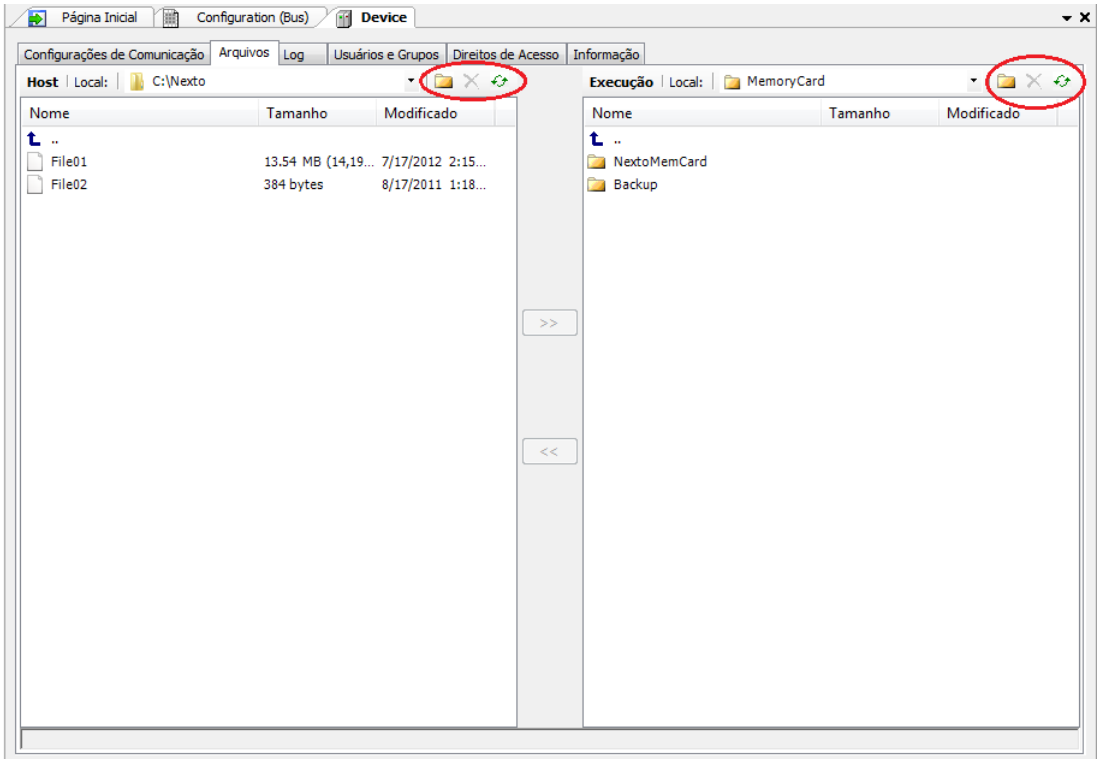


Figura 130: Opções de Utilização

ATENÇÃO

Para uma UCP em Modo Stop ou sem nenhuma aplicação, a taxa de transferência para a memória interna é de aproximadamente 150 Kbytes/s.

5.10. Cartão de Memória

Entre outras funcionalidades,este modelo de UCP da Série Nexto possibilita ao usuário a utilização de um cartão de memória, definido conforme as características da seção Interface do Cartão de Memória , o qual serve para armazenar, entre outros arquivos, o projeto e a aplicação que está na memória interna da UCP.

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

ATENÇÃO

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

5.10.1. Preparação do Projeto

Para utilização da funcionalidade, durante a configuração do projeto, dentro do software MasterTool IEC XE, o usuário deve habilitar a opção de cópia do projeto da UCP para o cartão de memória e cartão de memória para a UCP e configurar

senhas de acesso. Estas senhas serão solicitadas pela UCP no momento da execução da respectiva transferência . Informações sobre a tabela, consultar a seção Parâmetros do Projeto.

ATENÇÃO

Caso a UCP não tenha aplicação carregada, o Menu "*Cartão de Memória*" estará disponível para permitir a realização da transferência do projeto do cartão para a UCP sem que seja necessário realizar qualquer tipo de preparação prévia da UCP.

Para utilização da funcionalidade é necessário executar os passos a seguir.

Deve-se navegar até o menu *Comunicação* e executar o comando *Criar Aplicação de Inicialização*, lembrando que não se pode estar logado na UCP para realizar esse procedimento. Após executar esse comando, dois arquivos são criados na pasta onde o projeto está salvo. Um com a extensão "*app*" e outro com a extensão "*crc*".

Após gerados os arquivos no item anterior deve-se navegar até a configuração dos *Parâmetros Gerais* da UCP e clicar no botão *Cartão de Memória*. Uma nova tela será aberta conforme a figura abaixo. Nesta tela deve-se habilitar a(s) operação(ões) de transferência desejada(s) e, caso necessário, configurar a(s) senha(s), apenas com caracteres numéricos. A utilização de senha não é obrigatória.

Para concluir a operação de configuração é necessário clicar no botão *Localizar Arquivo...* e então localizar o arquivo com extensão "*.crc*" gerado no passo anterior.

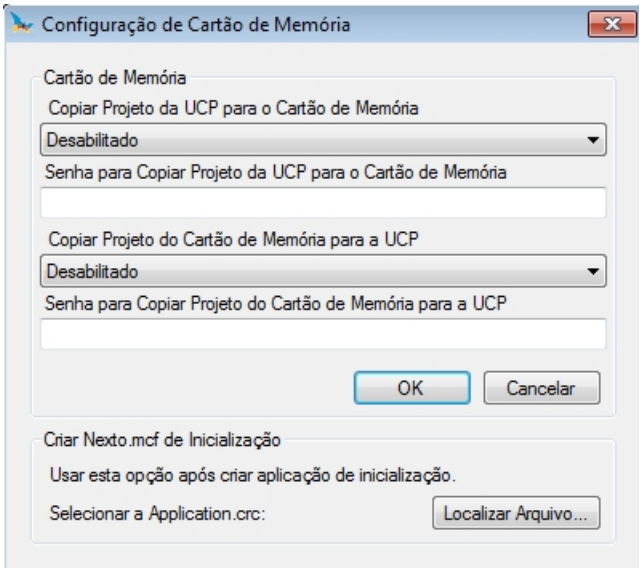


Figura 131: Configurações do Cartão de Memória

Executados esses passos, o MasterTool IEC XE irá enviar todos os arquivos necessários para realizar as operações de envio e recebimento de projetos via cartão de memória. Caso o cartão esteja montado, a senha será gravada no mesmo. Caso contrário, a senha configurada no MasterTool será solicitada se o usuário tentar transferir o projeto da UCP para o cartão.

5.10.2. Envio do Projeto

Para realizar o envio do projeto da UCP para o cartão de memória ou vice-versa, o usuário, além de habilitar no software MasterTool IEC XE o uso da funcionalidade , terá que acessar o Menu *Cartão de Memória* na UCP, utilizando a tecla de diagnósticos, e selecionar a opção de transferência desejada.

ATENÇÃO

O envio do projeto para o cartão de memória, deve ser realizado apenas usando a tecla de diagnósticos da UCP.

Após, será solicitada a senha, caso o usuário tenha configurado durante a configuração da aplicação. Então, com um pressionamento curto na tecla de diagnósticos, os dígitos são incrementados e com um pressionamento longo são confirmados. No sexto dígito confirmado, a UCP irá consistir a senha e iniciará o processo.

Após a transferência do cartão de memória para a UCP, caso haja uma aplicação em RUN ela será mantida em STOP por motivos de segurança. Para colocar a UCP em RUN, ela deve ser reinicializada.

Quando as senhas, tanto da aplicação que está na UCP quanto da aplicação que está no cartão de memória, forem iguais, não é requisitado a inserção das senhas no menu da UCP para realizar as transferências das aplicações. Maiores informações sobre a utilização da tecla de diagnósticos, consultar a seção One Touch Diag.

Para remover o cartão de memória, basta pressionar a tecla *MS* com um pressionamento longo, e aguardar até que o ícone do cartão desapareça da tela de status do visor gráfico.

ATENÇÃO

Caso o cartão de memória seja removido sem ser desmontado via menu da UCP, durante a transferência de arquivos, este processo pode acarretar na perda de dados no cartão, bem como corromper os arquivos nele contidos. Este processo pode implicar na necessidade de uma nova formatação do cartão quando inserido novamente à UCP.

ATENÇÃO

Caso exista algum arquivo na raiz do cartão de memória nomeado "*NextoMemCard*" ou "*Backup*", o mesmo será excluído para a criação das pastas de mesmo nome, utilizadas pela UCP para armazenamento do projeto de aplicação e do *Project archive*. Pastas com estes nomes não serão sobrescritas.

5.10.3. Acesso no MasterTool

O acesso à memória do cartão está vinculado à mesma tela de memória de usuário no software MasterTool IEC XE, sendo ele montado na pasta *MemoryCard*. Os diretórios *NextoMemCard* e *Backup* são criados dentro do cartão de memória toda a vez que ele for inserido na UCP. Caso eles já existam, o sistema irá reconhecer e não sobrescreverá as pastas.

Na pasta *NextoMemCard* no cartão de memória, encontram-se os arquivos referentes à aplicação, nesta janela você ainda tem a opção de salvar em um diretório de preferência o seu projeto(caso tenha enviado o código fonte). No MasterTool, na opção "*Arquivo/ Arquivo de Projeto/ Extrair Arquivo...*"pode-se abrir no MasterTool a aplicação salva, que está localizada no diretório escolhido anteriormente.

A pasta *Backup* não é utilizada pelo usuário.

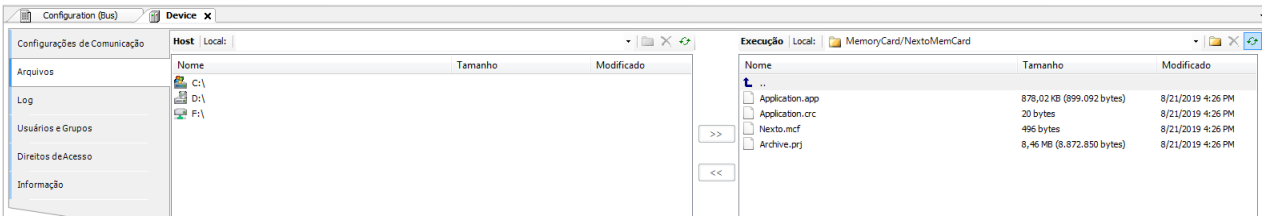


Figura 132: Diretório com Cartão de Memória Inserido com Projeto

ATENÇÃO

O tempo de transferência de arquivos depende da diferença do tempo de intervalo menos o tempo de execução médio da(s) tarefa(s) em execução (tempo disponível até o próximo ciclo da tarefa), isto é, quanto maior for essa diferença para cada tarefa em uma aplicação, mais rápida deverá ser a transferência de um dado a partir do cartão de memória para a UCP/MasterTool IEC XE ou vice-versa.
A transferência de arquivos para o cartão de memória será mais lenta que a transferência para a memória interna da UCP. Para uma UCP em Modo Stop ou sem nenhuma aplicação, a taxa de transferência se aproxima de 100 Kbytes/s.

5.11. Menu Informativo e de Configuração da UCP

O acesso ao *Menu Informativo* e de *Configuração* das UCPs Nexto através do *Visor Gráfico*, assim como o acesso detalhado aos diagnósticos, estão disponíveis através de níveis, sendo que para acessar as informações do menu, trocar de nível e modificar alguma configuração, basta dar um pressionamento longo na tecla de diagnóstico e, para navegar pelos itens de mesmo nível, basta dar um pressionamento curto na tecla de diagnóstico. Consultar a seção One Touch Diag para verificar o funcionamento e a diferença entre tipos de pressionamento na tecla de diagnósticos.

A tabela abaixo mostra os níveis do menu e o tipo de cada tela disponível nas UCPs, ou seja, se ela é de caráter informativo, configurável ou se retorna um nível.

Nível 1	Nível 2	Nível 3	Tipo
HARDWARE	TEMPERATURA	-	Informativo
	CONTRASTE	NIVEL CONTRASTE	Configurável
	DATA E HORA	-	Informativo
	VOLTAR	-	Retorna Nível
IDIOMAS	ENGLISH	>ENGLISH	Configurável
	PORTUGUES	>PORTUGUES	Configurável
	ESPANOL	>ESPANOL	Configurável
	VOLTAR	-	Retorna Nível
REDE	END. IP NET 1	-	Informativo
	MASCARA NET 1		Informativo
	END. IP NET 2		Informativo
	MASCARA NET 2		Informativo
	VOLTAR		Retorna Nível
SOFTWARE	FIRMWARE	-	Informativo
	BOOTLOADER		Informativo
	PROC. AUX.		Informativo
	VOLTAR		Retorna Nível
CARTAO DE MEM.	CARTAO > UCP	SENHA UCP	Configurável
	UCP > CARTAO	SENHA CM	Configurável
	FORMATAR	CONFIRMA ?	Configurável
	VOLTAR	-	Retorna Nível
VOLTAR	-	-	Retorna Nível

Tabela 165: Níveis do Menu da UCP

Notas:

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

Senha: A senha para acesso aos dados do cartão de memória somente será necessária caso ela seja configurada no software MasterTool IEC XE. Não é possível editar a senha via menu.

Conforme já mostrou a Tabela 165, entre as opções disponíveis para visualização e alteração, encontram-se os principais dados necessários ao usuário, como:

- Informações sobre os recursos de hardware:
  - TEMPERATURA – Temperatura interna da UCP (Ex.: 36 C 97 F)
  - CONTRASTE – Ajuste do contraste do visor frontal da UCP
  - DATA E HORA – Data e hora configuradas na UCP (Ex.: 2001.01.31 00:00)
- Alteração do idioma do menu da UCP:
  - PORTUGUES – Altera o idioma para Português
  - ENGLISH – Altera o idioma para Inglês

  - ESPANOL – Altera o idioma para Espanhol
- Visualização de informações sobre a rede configurada no dispositivo:
  - END. IP NET 1 – Endereço IP (Ex.: 192.168.0.1)
  - MASCARA NET 1 – Máscara de subrede (Ex.: 255.255.255.0)
  - END. IP NET 2 – Endereço IP (Ex.: 192.168.0.2)
  - MASCARA NET 2 – Máscara de subrede (Ex.: 255.255.255.0)
- Informações sobre as versões de software:
  - FIRMWARE – Versão de software da UCP (Ex.: 1.0.0.0)
  - BOOTLOADER – Versão do bootloader da UCP (Ex.: 1.0.0.0)
  - PROC. AUX. – Versão do processador auxiliar da UCP (Ex.: 1.0.0.0)
- Acesso aos dados do Cartão de Memória:
  - CARTAO>UCP – Transferência do projeto do cartão de memória para a UCP
  - UCP>CARTAO – Transferência do projeto da UCP para o cartão de memória
  - FORMATAR – Formata o cartão para o sistema de arquivos FAT32

A figura abaixo descreve um exemplo de como operar o menu das UCPs Nexto, através do procedimento de ajuste do contraste a partir da tela de *Status*. Além de facilitar a configuração, é possível identificar todos os níveis de tela e o tipo de pressionamento para navegar entre as mesmas, sendo que para modificar os outros parâmetros, como *Idioma* e inserir a(s) senha(s) no *Cartão de Memória*, basta seguir a mesma lógica de acesso. O pressionamento curto mostra que o contraste está sendo incrementado (mais claro), sendo que no próximo pressionamento após o seu valor máximo, ele retorna ao valor mínimo (menos claro). O pressionamento longo mostra a confirmação do contraste desejado e o retorno ao nível anterior.

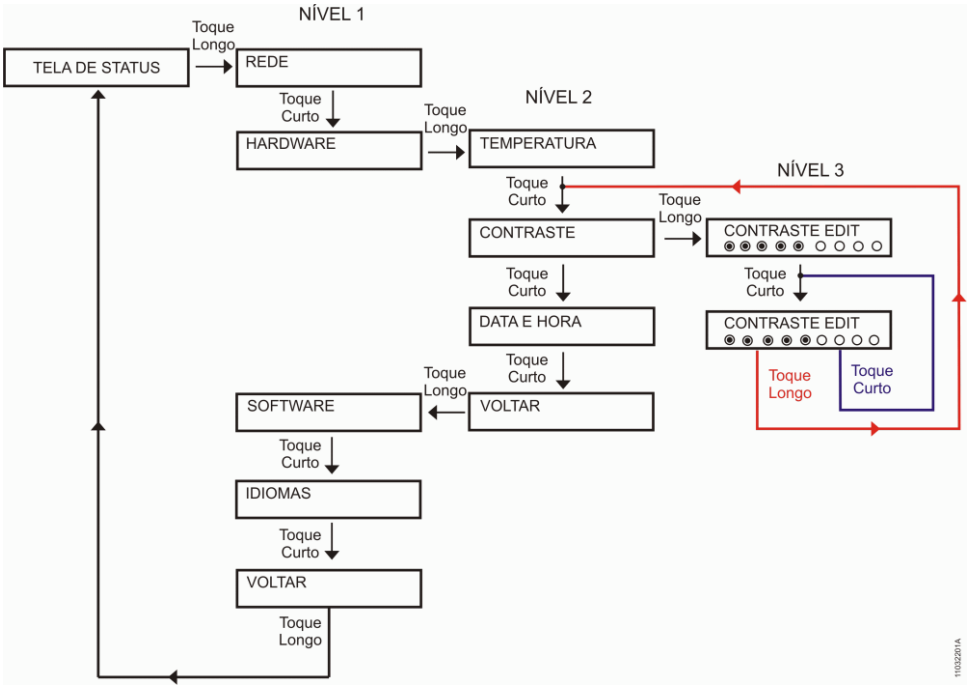


Figura 133: Ajuste do Contraste

Além do menu das UCPs Nexto ser encerrado através de um pressionamento longo na tecla de diagnósticos na tela *VOLTAR* do nível 1, também existem outras condições de saída, as quais estão descritas abaixo:

- Toque curto, em qualquer momento, em um outro módulo existente no barramento, faz com que o menu passe a descrever o diagnóstico dele.
- Tempo de inatividade, em qualquer nível, superior a 5 s.

5.12. Blocos Funcionais e Funções

5.12.1. Blocos Funcionais Especiais para Comunicação Serial

Os blocos funcionais especiais para comunicação serial possibilitam o acesso local (COM) e também a portas seriais remotas (módulos de expansão). Dessa forma, o usuário poderá criar seus próprios protocolos e manipular as portas seriais como quiser, seguindo as linguagens da IEC 61131-3 disponíveis no software MasterTool IEC XE. Os blocos estão disponíveis dentro da biblioteca *NextoSerial*, a qual deve ser adicionada ao projeto para que seja possível utilizá-los (para realizar o procedimento de inserção de uma biblioteca, consultar o Manual de Programação IEC 61131 – MP399048, capítulo Bibliotecas).

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

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

Tipo de dado	Opção	Descrição
SERIAL_BAUDRATE	BAUD200	Lista todas as possibilidades da taxa de transmissão (bits por segundo)
	BAUD300	
	BAUD600	
	BAUD1200	
	BAUD1800	
	BAUD2400	
	BAUD4800	
	BAUD9600	
	BAUD19200	
	BAUD38400	
	BAUD57600	
	BAUD115200	
SERIAL_DATABITS	DATABITS_5	Lista todas as possibilidades de bits de dado.
	DATABITS_6	
	DATABITS_7	
	DATABITS_8	
SERIAL_HANDSHAKE	Define todas as possibilidades dos sinais de modem para as configurações:	
	RS232_RTS	Controla a porta RS-232C da UCP Nexto. O RTS é habilitado no início da transmissão e reiniciado assim que possível após o final da transmissão. Por exemplo, pode ser utilizado para controlar um conversor RS-232/RS-485 externo.
	RS232_RTS_OFF	Controla a porta RS-232C da UCP Nexto. O sinal RTS está sempre desligado.
	RS232_RTS_ON	Controla a porta RS-232C da UCP Nexto. O sinal RTS está sempre ligado.

Tipo de dado	Opção	Descrição
	RS232_RTS_CTS	Controla a porta RS-232C da UCP Nexto. Caso o CTS esteja desabilitado, o RTS é habilitado. Então, aguarda-se o CTS ser habilitado para a transmissão começar e o RTS é reiniciado, o mais rápido possível, no final da transmissão. Ex: Controle de rádio modems com o mesmo sinal de modem.
	RS232_MANUAL	Controla a porta RS-232C da UCP Nexto. O usuário é responsável por controlar todos os sinais (RTS, DTR, CTS, DSR, DCD).
SERIAL_MODE	NORMAL_MODE	Modo normal de operação da comunicação serial.
	EXTENDED_MODE	Modo estendido de operação da comunicação serial, no qual são fornecidas informações sobre o frame de dados recebido.
SERIAL_PARAMETERS	Define todos os parâmetros de configuração da porta serial:	
	BAUDRATE	Definido em SERIAL_BAUDRATE.
	DATABITS	Definido em SERIAL_DATABITS.
	STOPBITS	Definido em SERIAL_STOPBITS.
	PARITY	Definido em SERIAL_PARITY.
	HANDSHAKE	Definido em SERIAL_HANDSHAKE.
	UART_RX_THRESHOLD	Quantidade de bytes que devem ser recebidos para gerar uma nova interrupção na UART. Valores baixos fazem o TIMESTAMP mais preciso quando o MODO ESTENDIDO é utilizado e minimiza os erros de overrun. No entanto, valores baixos podem causar muitas interrupções que podem retardar a UCP.
	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	

Tipo de dado	Opção	Descrição
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).
SERIAL_RX_QUEUE_ STATUS	Contém alguns campos que disponibilizam informações de status/erro sobre a fila RX, utilizados quando é utilizado o formato normal (sem erro e informações de estampa de tempo):	
	RX_FRAMING_ERRORS	Contador de erros de frame, ou seja, formação incorreta do caractere – falta de bit de parada, taxa de transmissão incorreta, entre outros – desde a configuração da porta serial. Retorna para zero caso atinja o valor máximo (65535).
	RX_PARITY_ERRORS	Contador de erros de paridade desde a configuração da porta serial. Retorna para zero caso atinja o valor máximo (65535).
	RX_BREAK_ERRORS	Contador de erros de interrupção desde a configuração da porta serial, ou seja, linha ativa maior do que o tempo de um caractere. Retorna para zero caso atinja o valor máximo (65535).
	RX_FIFO_OVERRUN_ ERRORS	Contador de erros de overrun na FIFO RX desde a configuração da porta serial, ou seja, erro no threshold configurado para a FIFO RX. Retorna para zero caso atinja o valor máximo (65535).
	RX_QUEUE_OVERRUN_ ERRORS	Contador de erros de overrun na fila RX desde a configuração da porta serial, ou seja, o valor máximo de caracteres (1024) foi ultrapassado e os dados estão sendo sobrescritos. Retorna zero caso atinja o valor máximo (65535).
	RX_ANY_ERRORS	Soma dos 5 últimos contadores de erros (frame, paridade, interrupção, overrun RX FIFO, overrun fila RX).
	RX_REMAINING	Número de caracteres remanescentes na fila RX.

Tipo de dado	Opção	Descrição
SERIAL_STATUS	Lista os códigos de erro críticos que podem ser retornados pelos blocos funcionais da serial. Cada bloco retorna erros específicos, os quais serão mencionados na descrição dos mesmos:	
	NO_ERROR	Não existem erros.
	ILLEGAL_*	Retorna os parâmetros com valores inválidos ou fora da faixa: - SERIAL_PORT - SERIAL_MODE - BAUDRATE - DATA_BITS - PARITY - STOP_BITS - HANDSHAKE - UART_RX_THRESHOLD - TIMEOUT - TX_BUFF_LENGTH - HANDSHAKE_METHOD - RX_BUFF_LENGTH
	PORT_BUSY	Indica que a porta serial está sendo utilizada por outra instância
	HW_ERROR_UART	Erro de hardware detectado na UART.
	HW_ERROR_REMOTE	Erro de hardware ao comunicar com a porta serial remota.
	CTS_TIMEOUT_ON	Time-out na espera do CTS ser habilitado, nos sinais de modem RS-232C RTS/CTS, no bloco SERIAL_TX.
	CTS_TIMEOUT_OFF	Time-out na espera do CTS ser desabilitado, no handshake RS-232C RTS/CTS, no bloco SERIAL_TX.
	TX_TIMEOUT_ERROR	Time-out na espera pelo final da transmissão no bloco SERIAL_TX.
	RX_TIMEOUT_ERROR	Time-out na espera de todos os caracteres no bloco SERIAL_RX ou SERIAL_RX_EXTENDED.
	FB_SET_CTRL_ NOT_ALLOWED	O bloco SET_CTRL não pode ser utilizado caso o sinal de modem seja diferente de RS232_MANUAL.
	FB_GET_CTRL_ NOT_ALLOWED	O bloco GET_CTRL não pode ser utilizado caso o sinal de modem seja diferente de RS232_MANUAL.
	FB_SERIAL_RX_ NOT_ALLOWED	O bloco SERIAL_RX não está disponível para a fila RX, modo estendido.
	FB_SERIAL_RX_ EXTEN- DED_NOT_ALLOWED	O bloco SERIAL_RX_EXTENDED não está disponível para a fila RX, modo normal.

Tipo de dado	Opção	Descrição
	DCD_INTERRUPT_ NOT_ALLOWED	A interrupção pelo sinal DCD não pode ser habilitada caso a porta serial não possua o respectivo pino.
	CTS_INTERRUPT_ NOT_ALLOWED	A interrupção pelo sinal CTS não pode ser habilitada caso o sinal de modem seja diferente de RS232_MANUAL ou caso a porta serial não possua o respectivo pino.
	DSR_INTERRUPT_ NOT_ALLOWED	A interrupção pelo sinal DSR não pode ser habilitada caso a porta serial não possua o respectivo pino. (As UCPs Nexto não possuem esse sinal nas portas locais)
	NOT_CONFIGURED	O bloco funcional não pode ser utilizado antes da porta serial ser configurada.
	INTERNAL_ERROR	Indica que algum problema interno ocorreu na porta serial.
SERIAL_STOPBITS	STOPBITS_1	Lista todas as possibilidades de bits de parada.
	STOPBITS_2	
	STOPBITS_1_5	

Tabela 166: Tipos de dados Blocos Funcionais Serial

5.12.1.1. SERIAL_CFG

Esse bloco funcional é utilizado para configurar e inicializar a porta serial desejada. Após a chamada do bloco, todas as filas RX e TX associadas à porta serial e os FIFOs RX e TX, são reiniciados.

Figura 134: Bloco de Configuração da Serial

Parâmetros de entrada	Tipo	Descrição
REQUEST	BOOL	Essa variável, quando verdadeira, habilita o uso do bloco funcional.
PORT	SERIAL_PORT	Seleciona a porta serial, conforme descrito no tipo de dado SERIAL_PORT.
PARAMETERS	SERIAL_PARAMETERS	Essa estrutura define os parâmetros de configuração da porta serial, conforme descrito no tipo de dado SERIAL_PARAMETERS.

Tabela 167: 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 168: Parâmetros de Saída SERIAL_CFG

Exemplo de utilização em Linguagem ST, após a biblioteca Nexto Serial ter sido inserida no projeto:

```
PROGRAM UserPrg
VAR
Config: SERIAL_CFG;
Port: SERIAL_PORT := COM1;
Parameters: SERIAL_PARAMETERS := (BAUDRATE := BAUD9600,
DATABITS := DATABITS_8,
STOPBITS := STOPBITS_1,
PARITY := PARITY_NONE,
HANDSHAKE := RS232_RTS,
UART_RX_THRESHOLD := 8,
MODE :=NORMAL_MODE,
ENABLE_RX_ON_TX := FALSE,
ENABLE_DCD_EVENT := FALSE,
```

```
    ENABLE_CTS_EVENT := FALSE);
    Status: SERIAL_STATUS;
    END_VAR
    //ENTRADAS:
    Config.REQUEST := TRUE;
    Config.PORT := Port;
    Config.PARAMETERS := Parameters;
    //FUNÇÃO:
    Config();
    //SAÍDAS:
    Config.DONE;
    Config.EXEC;
    Config.ERROR;
    Status := Config.STATUS;    //Caso seja necessário tratar o erro.
```

5.12.1.2. SERIAL_GET_CFG

Esse bloco funcional é utilizado para capturar as configurações da porta serial desejada.

Figura 135: Bloco para Capturar a Configuração da Serial

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

Tabela 169: 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 170: 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 136: 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 171: 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 172: 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 137: 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 173: 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 174: 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 138: 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 175: 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 176: 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 139: 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 177: 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 178: Parâmetros de Saída SERIAL_RX

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

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

5.12.1.7. SERIAL_RX_EXTENDED

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

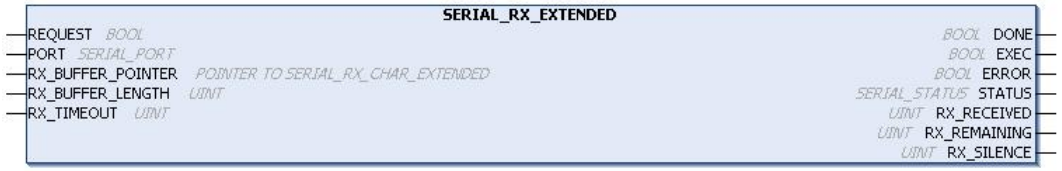


Figura 140: Bloco para Leitura do Buffer de Recepção

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 179: 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.

Parâmetros de saída	Tipo	Descrição
RX_REMAINING	UINT	Retorna o número de caracteres que ainda estão na fila RX depois que o bloco funcional foi executado.
RX_SILENCE	UDINT	Retorna o tempo de silêncio na linha RX, medido desde o fim do último caractere recebido. A unidade de tempo é 1 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 180: 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.

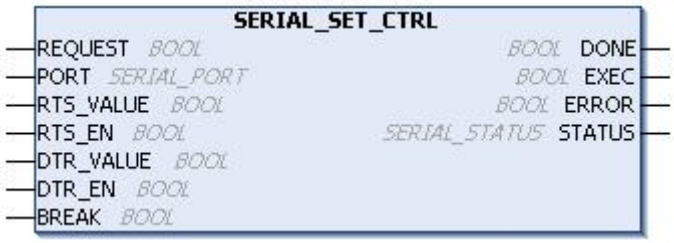


Figura 141: Bloco para Escrever nos Sinais de Controle

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 181: Parâmetros de Entrada SERIAL_SET_CTRL

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

Parâmetros de saída	Tipo	Descrição
STATUS	SERIAL_STATUS	Caso a variável ERROR seja verdadeira, a estrutura STATUS exibirá o erro encontrado na execução do bloco. Os estados, já descritos no tipo de dado SERIAL_STATUS, possíveis são: - NO_ERROR - ILLEGAL_SERIAL_PORT - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - FB_SET_CTRL_NOT_ALLOWED - NOT_CONFIGURED

Tabela 182: Parâmetros de Saída SERIAL_SET_CTRL

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

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

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

5.12.1.9. SERIAL_TX

Esse bloco funcional é utilizado para transmitir um buffer de dados pela porta serial, sendo que o mesmo somente é finalizado depois de todos os bytes serem transmitidos ou após o time-out (gera alguns erros).

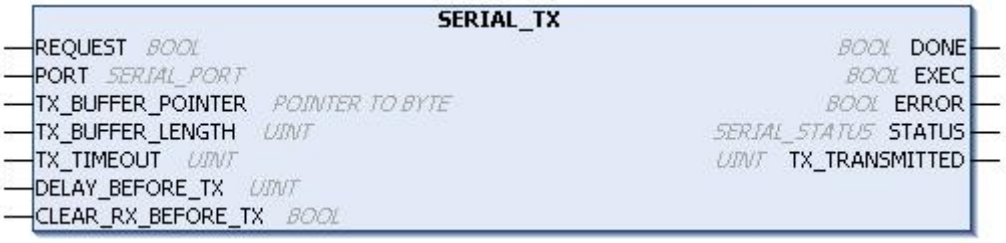


Figura 142: Bloco para Transmitir Valores pela SERIAL_TX

Parâmetros de entrada	Tipo	Descrição
REQUEST	BOOL	Essa variável, quando verdadeira, habilita o uso do bloco funcional.
PORT	SERIAL_PORT	Seleciona a porta serial, conforme descrito no tipo de dado SERIAL_PORT.
TX_BUFFER_ POINTER	POINTER TO BYTE	Ponteiro de um array de bytes para transmitir os valores do buffer.
TX_BUFFER_ LENGTH	UINT	Especifica o número de caracteres a serem transmitidos pelo array de bytes (tamanho máximo da fila TX é igual a 1024 caracteres).
TX_TIMEOUT	UINT	Especifica o time-out [ms] para completar a transmissão, incluindo a fase de handshake. O valor especificado, deve ser positivo e diferente de zero.
DELAY_BEFORE_ TX	UINT	Especifica o atraso [ms] entre a chamada do bloco funcional e o início da transmissão. Essa variável pode ser utilizada em comunicações com alguns 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 183: Parâmetros de Entrada SERIAL_TX

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

Parâmetros de saída	Tipo	Descrição
STATUS	SERIAL_STATUS	Caso a variável ERROR seja verdadeira, a estrutura STATUS exibirá o erro encontrado na execução do bloco. Os estados, já descritos no tipo de dado SERIAL_STATUS, possíveis são: - NO_ERROR - ILLEGAL_SERIAL_PORT - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - ILLEGAL_TX_BUFF_LENGTH - ILLEGAL_TIMEOUT - CTS_TIMEOUT_ON - CTS_TIMEOUT_OFF - TX_TIMEOUT_ERROR - NOT_CONFIGURED
TX_TRANSMITTED	UINT	Retorna o número de bytes transmitidos, o qual deve ser igual ao TX_BUFFER_LENGTH, mas pode ser menor caso ocorra algum erro durante a transmissão.

Tabela 184: Parâmetros de Saída SERIAL_TX

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

```
PROGRAM UserPrg
VAR
Transmit: SERIAL_TX;
Port: SERIAL_PORT := COM1;
Buffer_Pointer: ARRAY [0..9] OF BYTE := [0,1,2,3,4,5,6,7,8,9];
Status: SERIAL_STATUS;
END_VAR

//ENTRADAS:
Transmit.REQUEST := TRUE;
Transmit.PORT := Port;
Transmit.TX_BUFFER_POINTER := ADR(Buffer_Pointer);
Transmit.TX_BUFFER_LENGTH := 10;
Transmit.TX_TIMEOUT := 10000;
Transmit.DELAY_BEFORE_TX := 1000;
Transmit.CLEAR_RX_BEFORE_TX := TRUE;
//FUNÇÃO:
Transmit();
//SAÍDAS:
Transmit.DONE;
Transmit.EXEC;
Transmit.ERROR;
Status := Transmit.STATUS;    //Caso seja necessário tratar o erro.
Transmit.TX_TRANSMITTED;
```

5.12.2. Atualização de Entradas e Saídas

Funcionalidade utilizada para atualizar entradas e saídas no decorrer da aplicação, não sendo necessário aguardar até que seja completado um ciclo.

Quando os blocos funcionais para atualizar as entradas e saídas não são utilizados, a atualização é realizada a cada ciclo da MainTask.

ATENÇÃO

Na inicialização de uma UCP desta série, as entradas e saídas somente estarão atualizadas para leitura e preparadas para escrita quando a MainTask for executada. Todas as demais tarefas do sistema que executarem antes da MainTask estarão com as entradas e as saídas inválidas.

5.12.2.1. REFRESH_INPUT

Essa função é utilizada para atualizar as entradas do módulo especificado, sem aguardar o ciclo ser completado. É importante ressaltar que os filtros, configurados no MasterTool IEC XE, e o tempo de atualização das entradas do módulo, deverão ser considerados no tempo efetivo de atualização das entradas na aplicação desenvolvida pelo usuário.

ATENÇÃO

A função de *REFRESH_INPUT* deve ser utilizada somente na tarefa MainTask.
Para executar a atualização de entradas em outras tarefas, a opção *Habilita atualização de E/S por tarefa* deve ser selecionada, mais informações a respeito desta opção podem ser obtidas na Tabela 43.

ATENÇÃO

A função de *REFRESH_INPUT* não suporta a atualização de entradas que tenham sido mapeadas para variáveis simbólicas. Para o correto funcionamento é necessário que a entrada esteja mapeada para uma variável dentro da memória de variáveis de entrada de representação direta (%I).

ATENÇÃO

A função de *REFRESH_INPUT* atualiza apenas as variáveis diretas %I que estiverem declaradas na aba de "Bus: Mapeamento de E/S"do módulo endereçado no respectivo rack/slot da função. Em se tratando de módulos/interfaces de comunicação (MODBUS, Profibus, etc), a atualização não inclui as variáveis diretas dos mapeamentos dos dispositivos.

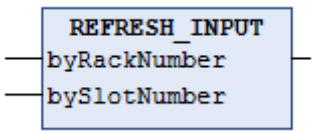


Figura 143: Bloco para Atualizar as Entradas

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

Tabela 185: Parâmetros de Entrada REFRESH_INPUT

Possíveis *TYPE_RESULT*:

- **OK_SUCCESS:** Execução com sucesso.
- **ERROR_FAILED:** Esse erro é retornado caso a função seja chamada para um módulo que possua somente saídas, ou também caso a opção Sempre atualizar variáveis (localizada na tela de configuração do módulo, aba Mapeamento de E/S) não esteja marcada.
- **ERROR_NOTSUPPORTED:** A função não é suportada pelo produto.
- **ERROR_PARAMETER:** Parâmetro inválido / não suportado.
- **ERROR_MODULE_ABSENT:** O módulo está ausente.
- **ERROR_MODULE_NOTCONFIGURED:** O módulo não está configurado na aplicação.
- **ERROR_MODULE_NOTRUNNING:** O módulo não está em funcionamento (não está no estado operacional).
- **ERROR_MODULE_COMMFAIL:** Falha de comunicação com o módulo.
- **ERROR_MODULE_NOTFOUND:** O módulo não foi encontrado na aplicação ou não é suportado.

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
Info: TYPE_RESULT;
byRackNumber: BYTE;
bySlotNumber: BYTE;
END_VAR
//ENTRADAS:
byRackNumber := 0;
bySlotNumber := 10;
//FUNÇÃO:
Info := REFRESH_INPUT (byRackNumber, bySlotNumber); //Chamada da função.
//Variável "Info" recebe possíveis erros da função.
```

5.12.2.2. REFRESH_OUTPUT

Essa função é utilizada para atualizar as saídas do módulo especificado, sem aguardar o ciclo ser completado. É importante ressaltar que o tempo de atualização das saídas do módulo deverá ser considerado no tempo efetivo de atualização das saídas na aplicação desenvolvida pelo usuário.

ATENÇÃO

A função de *REFRESH_OUTPUT* deve ser utilizada somente na tarefa MainTask.
Para executar a atualização de saídas em outras tarefas, a opção *Habilita atualização de E/S por tarefa* deve ser selecionada, mais informações a respeito desta opção podem ser obtidas na Tabela 43.

ATENÇÃO

A função de *REFRESH_OUTPUT* não suporta a atualização de saídas que tiverem sido mapeadas para variáveis simbólicas. Para o correto funcionamento é necessário que a saída esteja mapeada para uma variável dentro da memória de variáveis de saída de representação direta (%Q).

ATENÇÃO

A função de *REFRESH_OUTPUT* atualiza apenas as variáveis diretas %Q que estiverem declaradas na aba de "Bus: Mapeamento de E/S"do módulo endereçado no respectivo rack/slot da função. Em se tratando de módulos/interfaces de comunicação (MODBUS, Profibus, etc), a atualização não inclui as variáveis diretas dos mapeamentos dos dispositivos.

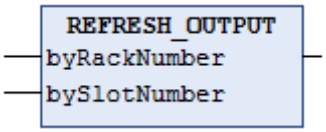


Figura 144: Bloco para Atualizar as Saídas

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

Tabela 186: Parâmetros de Entrada REFRESH_OUTPUT

Possíveis *TYPE_RESULT*:

- **OK_SUCCESS:** Execução com sucesso.
- **ERROR_FAILED:** Esse erro é retornado caso a função seja chamada para um módulo que possua somente entradas, ou também caso a opção Sempre atualizar variáveis (localizada na tela de configuração do módulo, aba Mapeamento de E/S) não esteja marcada.
- **ERROR_NOTSUPPORTED:** A função não é suportada pelo produto.
- **ERROR_PARAMETER:** Parâmetro inválido / não suportado.
- **ERROR_MODULE_ABSENT:** O módulo está ausente.
- **ERROR_MODULE_NOTCONFIGURED:** O módulo não está configurado na aplicação.
- **ERROR_MODULE_NOTRUNNING:** O módulo não está em funcionamento (não está no estado operacional).
- **ERROR_MODULE_COMMFAIL:** Falha de comunicação com o módulo.
- **ERROR_MODULE_NOTFOUND:** O módulo não foi encontrado na aplicação ou não é suportado.

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
Info: TYPE_RESULT;
byRackNumber: BYTE;
bySlotNumber: BYTE;
END_VAR
//ENTRADAS:
byRackNumber := 0;
bySlotNumber := 10;
//FUNÇÃO:
//Chamada da função.
Info := REFRESH_OUTPUT (byRackNumber, bySlotNumber);
//Variável "Info" recebe possíveis erros da função.
```

5.12.3. Bloco Funcional PID

ATENÇÃO

O bloco funcional PID descrito até a revisão anterior O deste manual tornou-se obsoleto e foi removido deste manual.
Os blocos funcionais PID, PID_INT e PID_REAL descritos até a revisão D do MP399048, também tornaram-se obsoletos e foram também removidos a partir das versões mais recentes desse manual. Usuários que precisam da descrição desses blocos funcionais obsoletos devido a motivos de manutenção devem utilizar a revisão D do MP399048.
Os blocos funcionais PID, PID_INT e PID_REAL não devem ser utilizados em novos projetos. Esses blocos funcionais foram substituídos por novos blocos funcionais com recursos adicionais, como ajuste automático e suporte à controles em cascata, sobreposição e feed-forward. Esses novos blocos funcionais são descritos no MU214610, e estão disponíveis após a versão 1.1.0.0 da biblioteca *NextoPID*.

5.12.4. Temporizador Retentivo

O Temporizador Retentivo é um bloco funcional desenvolvido para aplicações como relógios de linha de produção, as quais necessitam armazenar o seu valor e reiniciar a contagem do mesmo ponto em caso de falha na alimentação. Os valores, guardados pelo bloco funcional somente serão zerados em caso de um *Reset a Frio*, *Reset Origem* ou o *Download* de uma nova aplicação (ver Manual de Utilização do MasterTool IEC XE - MU299048), sendo que os contadores continuam em funcionamento mesmo que a aplicação esteja parada (Modo Stop).

ATENÇÃO

É importante destacar que, para o correto funcionamento dos blocos funcionais do Temporizador Retentivo, as variáveis de controle devem ser declaradas como retentivas (*VAR RETAIN*). Também é importante ressaltar que em modo simulação os blocos funcionais do Temporizador Retentivo não são executados adequadamente em virtude de necessitarem da UCP Nexto para o correto comportamento.

Abaixo, são descritos os três tipos de blocos disponíveis na biblioteca *NextoStandard* do software MasterTool IEC XE (para realizar o procedimento de inserção de uma biblioteca, consultar o Manual de Programação IEC 61131 – MP399048, capítulo Bibliotecas).

5.12.4.1. TOF_RET

O bloco funcional *TOF_RET* implementa um tempo de atraso para desabilitar uma saída. Quando a entrada *IN* tem seu estado alterado de verdadeiro (TRUE) para falso (FALSE), ou seja, uma borda de descida, o tempo especificado *PT* irá transcorrer até que a saída *Q* também seja falsa (FALSE). Quando a entrada *IN* tem nível lógico 1 (TRUE), a saída *Q* também permanecerá no mesmo estado (TRUE), mesmo que isso aconteça no meio de uma contagem. O tempo *PT* pode ser alterado durante a contagem, pois o bloco funcional assumirá o novo valor, desde que a contagem não tenha chegado ao final. A Figura 145 representa o bloco *TOF_RET* e a Figura 146 mostra o comportamento gráfico do mesmo.

Figura 145: 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 187: 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 188: Parâmetros de Saída TOF_RET

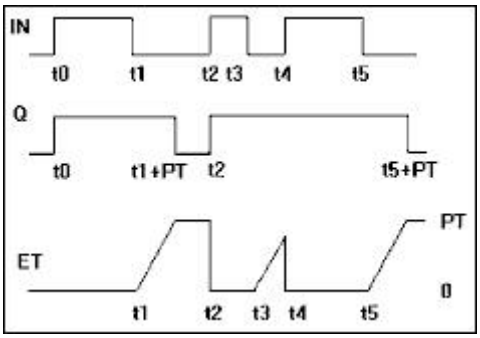


Figura 146: Comportamento Gráfico do Bloco Funcional TOF_RET

Exemplo de utilização em linguagem ST:

```
PROGRAM UserPrg
VAR RETAIN
bStart : BOOL := TRUE;
TOF_RET : TOF_RET;
END_VAR

// Quando bStart=FALSE inicia contagem
TOF_RET( IN := bStart,
PT := T#20S);

// Executa ações ao final da contagem
IF (TOF_RET.Q = FALSE) THEN
bStart := TRUE;
END_IF
```

5.12.4.2. TON_RET

O bloco funcional *TON_RET* implementa um tempo de atraso para habilitar uma saída. Quando a entrada *IN* tem seu estado alterado de falso (FALSE) para verdadeiro (TRUE), ou seja, uma borda de subida, o tempo especificado *PT* irá transcorrer até que a saída *Q* também seja verdadeira (TRUE). Quando a entrada *IN* tem nível lógico 0 (FALSE), a saída *Q* também permanecerá no mesmo estado (FALSE), mesmo que isso aconteça no meio de uma contagem. O tempo *PT* pode ser alterado durante a contagem, pois o bloco funcional assumirá o novo valor, desde que a contagem não tenha chegado ao final. A Figura 147 representa o bloco *TON_RET* e a Figura 148 mostra o comportamento gráfico do mesmo.

Figura 147: 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 189: 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 190: Parâmetros de Saída TON_RET

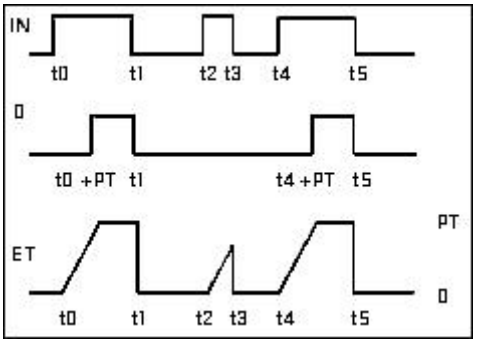


Figura 148: Comportamento Gráfico do Bloco Funcional TON_RET

Exemplo de utilização em linguagem ST:

```
PROGRAM UserPrg
VAR RETAIN
bStart : BOOL;
TON_RET : TON_RET;
END_VAR

// Quando bStart=TRUE inicia contagem
TON_RET( IN := bStart,
PT := T#20S);

// Executa ações ao final da contagem
IF (TON_RET.Q = TRUE) THEN
bStart := FALSE;
END_IF
```

5.12.4.3. TP_RET

O bloco funcional *TP_RET* trabalha como um trigger. O temporizador, que inicia quando a entrada *IN* tem seu estado alterado de falso (FALSE) para verdadeiro (TRUE), ou seja, uma borda de subida, é incrementado até que o limite de tempo *PT* seja atingido. Durante a contagem, a saída *Q* é verdadeira (TRUE), caso contrário ela é falsa (FALSE). O tempo *PT* pode ser alterado durante a contagem, pois o bloco assumirá o novo valor, desde que a contagem não tenha chegado ao final. A Figura 149 representa o bloco *TP_RET* e a Figura 150 mostra o comportamento gráfico do mesmo.

Figura 149: 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 191: 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 192: Parâmetros de Saída TP_RET

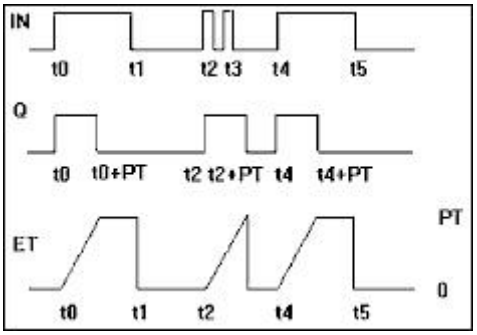


Figura 150: Comportamento Gráfico do Bloco Funcional TP_RET

Exemplo de utilização em linguagem ST:

```
PROGRAM UserPrg
VAR RETAIN
bStart : BOOL;
TP_RET : TP_RET;
END_VAR

// Configura TP_RET
TP_RET( IN := bStart,
PT := T#20S);

bStart := FALSE;

// Ações durante a contagem
IF (TP_RET.Q = TRUE) THEN
// Executa enquanto o contador estiver ativado
ELSE
// Executa somente quando o contador estiver desativado
END_IF
```

5.12.5. Log de Usuário

Recurso que permite ao usuário criar seus próprios registros e gravar em arquivos de log no cartão de memória presente na UCP. Os arquivos são gerados em um diretório específico do cartão de memória no formato CSV, permitindo a visualização em editores de texto e planilhas. O separador adotado foi o caractere ponto e vírgula. Para mais informações sobre a utilização do cartão de memória, ver seção Cartão de Memória.

Há duas funções disponíveis, uma para registrar informações e outra para remover todos os registros. A seguir uma descrição dos tipos de dados utilizados pelas funções:

Tipo de dado	Opção	Descrição
USER_LOG_EVENT_TYPES	USER_LOG_EVENT_ERROR	O usuário é livre para usar a melhor indicação conforme a severidade de sua mensagem de log.
	USER_LOG_EVENT_DEBUG	
	USER_LOG_EVENT_INFO	
	USER_LOG_EVENT_WARN	
USER_LOG_MESSAGE		Mensagem de log com o máximo de 150 caracteres.

Tipo de dado	Opção	Descrição
USER_LOG_ERROR_CODES	USER_LOG_OK	A operação foi executada com sucesso.
	USER_LOG_FAILED	A operação não foi executada com sucesso. O motivo da falha pode ser verificado nos logs de sistema – ver seção Log de Sistema.
	USER_LOG_BUFFER_FULL	Mensagens estão sendo adicionadas além da capacidade de processamento.
	USER_LOG_NO_MEMORY	No momento, não houve recursos para executar a operação.
	USER_LOG_FILE_SYSTEM_ERROR	Houve algum erro no acesso ao cartão de memória ou não há espaço disponível. Informações do erro podem ser verificadas nos logs de sistema – ver seção Log de Sistema.
	USER_LOG_NO_MEMORY_CARD	Não há um cartão de memória presente na UCP.
	USER_LOG_MEMORY_CARD_FULL	Não há espaço livre disponível no cartão de memória.
	USER_LOG_PROCESSING	O recurso está ocupado executando a última operação, por exemplo, deletando todos os arquivos de log.

Tabela 193: Tipos de dados para Log de Usuário

A seguir, são descritas as duas funções disponíveis na biblioteca *LibLogs* no MasterTool IEC XE. Para realizar o procedimento de inserção de uma biblioteca, consultar o Manual de Programação IEC 61131 – MP399048, capítulo Bibliotecas.

ATENÇÃO

Os Logs de Usuário estão disponíveis apenas a partir da versão 1.3.0.20 das UCPs da Série Nexto. Da mesma forma, para utilizar esta funcionalidade é necessária a versão 1.40 ou superior do MasterTool IEC XE.

5.12.5.1. UserLogAdd

Essa função é utilizada para adicionar uma nova mensagem de log do usuário, acrescentando em uma nova linha ao arquivo de log no cartão de memória. A mensagem deve ter um tamanho máximo de 150 caracteres e o tipo de evento da mensagem. Variáveis da aplicação podem ser registradas utilizando conversão para string e concatenação com a mensagem principal. É adicionada automaticamente na mensagem a informação de data e hora em UTC (estampa de tempo) com resolução de milissegundos em que o evento foi registrado. A informação de data e hora também é utilizada na formação dos nomes dos arquivos de log.

A função *UserLogAdd* pode ser utilizada para inserir várias mensagens dentro de uma mesma tarefa e também em tarefas diferentes da aplicação. Porém independente de cada execução da função na aplicação, sendo ela na mesma tarefa ou em tarefas diferentes, todas utilizam o mesmo recurso para gravar as mensagens desejadas. Por esse motivo é recomendado que a adição de mensagens utilizando a função *UserLogAdd* na aplicação seja realizada a cada 50 ms para evitar retorno de buffer cheio. Caso a função seja executada em períodos menores que o indicado, mas respeite o tempo médio de 50 ms entre cada adição de mensagem ao término do intervalo da tarefa, também evita o retorno de buffer cheio. Para que os logs sejam adicionados corretamente, é importante respeitar os limites de tempo quando o cartão é inserido e na inicialização da UCP, mencionados na seção Cartão de Memória. Após a operação a função retorna as opções para o tipo de dado *USER_LOG_ERROR_CODES*, conforme a Tabela 193. Quando o retorno da função for diferente de *USER_LOG_OK*, a mensagem não foi registrada, devendo ser executada novamente a função *UserLogAdd* com a mensagem desejada. Em caso de retorno de falhas consecutivas de escrita, o cartão de memória pode estar danificado. A substituição por um cartão de memória íntegro garante que as últimas mensagens registradas serão gravadas no cartão que não está danificado, desde que a UCP não seja reiniciada.

A figura abaixo representa a função *UserLogAdd* e a tabela abaixo os parâmetros de entrada:

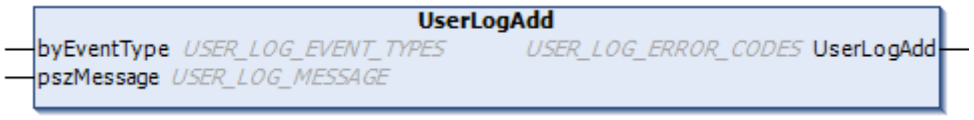


Figura 151: Função UserLogAdd

Parâmetros de entrada	Tipo	Descrição
byEventType	BYTE	Essa variável especifica o tipo de evento do log que está sendo adicionado conforme as opções para o tipo de dado USER_LOG_EVENT_TYPES.
pszMessage	USER_LOG_MESSAGE	Essa variável deve conter o conjunto de caracteres que compõe a mensagem a ser adicionada no arquivo de log. A mensagem deve conter no máximo 150 caracteres.

Tabela 194: Parâmetros de Entrada UserLogAdd

Os arquivos de log são gerados e organizados no cartão de memória em um caminho de diretórios específico que depende do número de série da UCP e da versão de firmware instalada. Por exemplo, para uma UCP com número de série 445627 e versão de firmware 1.4.0.4, o local onde os arquivos de log devem ser gravados no cartão de memória é *MemoryCard/UserLog/445627/1.4.0.4/*.

Os nomes dos arquivos de log são formados pela data e hora (estampa de tempo) da primeira mensagem. Exceto quando há algum problema para utilizar esse nome, como por exemplo, outro arquivo existente com o mesmo nome, nessa situação são utilizadas a data e hora instantâneas. O nome do arquivo segue o seguinte padrão: *ano/mês/dia/hora/minuto/segundo/milissegundo.CSV*. Caso um arquivo apresente problema de acesso por setor defeituoso e não seja possível continuar com escrita, será adicionado ao nome desse arquivo a extensão *"corrupted"* e um novo arquivo será criado. A quantidade de logs por arquivo não é fixa, variando em função do tamanho das mensagens. A quantidade de arquivos criados é limitada em 1024 com tamanho máximo de 1 MB cada, portanto o cartão de memória necessita de 1 GB de espaço livre. Quando atingir o limite de 1024 arquivos criados no cartão de memória, durante a operação da UCP, os arquivos mais antigos são removidos para que os arquivos com logs mais recentes sejam preservados, mesmo nos casos de remoção manual parcial dos arquivos no diretório onde os arquivos estão sendo gravados.

A visualização dos arquivos de log pode ser realizada através de planilhas ou editores de texto convencionais. As informações concatenadas, para melhor visualização, podem utilizar ponto e vírgula entre as strings da mensagem para separá-las. Deve-se ter o cuidado na formatação de células com valores em ponto flutuante.

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
  eLogError : USER_LOG_ERROR_CODES;
  sMessage :USER_LOG_MESSAGE;
END_VAR

IF (m_rTemperature > MAX_TEMPERATURE_ACCEPT) THEN
  sMessage := 'Temperatura acima do esperado: ';
  sMessage := concat(sMessage, REAL_TO_STRING(m_rTemperature));
  sMessage := concat(sMessage, '°');
  eLogError := UserLogAdd(USER_LOG_EVENT_WARN, sMessage);
//Variável eLogError recebe possíveis erros da função.
END_IF
```

Exemplo de conteúdo do arquivo de log: (*UserLog-201308271506245738.csv*)

```
Model; NX3030
Serial number; 445627
Firmware version; 1.4.0.4

27/08/2013 15:06:24.5738; WARN; Temperatura acima do esperado: 25◦
27/08/2013 16:37:45.3476; WARN; Temperatura acima do esperado: 25◦
28/08/2013 09:10:55.4201; WARN; Temperatura acima do esperado: 26◦
```

5.12.5.2. UserLogDeleteAll

A função *UserLogDeleteAll* realiza a exclusão dos arquivos de log presentes no diretório criado especificamente para a UCP em que está inserido o cartão de memória, ou seja, são excluídos apenas os logs contidos no diretório nomeado com a versão de firmware da UCP que existe dentro do diretório com a versão de série da UCP. Os arquivos de log excluídos são apenas os arquivos existentes no momento da montagem do cartão de memória e os gerados pela função *UserLogAdd*. Registros de outras UCPs e arquivos adicionados manualmente pelo usuário durante a execução não são apagados. A figura abaixo representa a função *UserLogDeleteAll*.

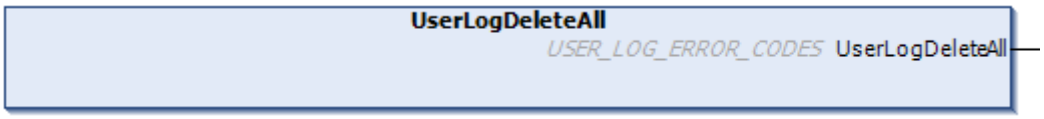


Figura 152: Função UserLogDeleteAll

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
eLogError : USER_LOG_ERROR_CODES;
END_VAR

IF (m_DeleteLogs = TRUE) THEN
//Variável eLogError recebe possíveis erros da função.
eLogError := UserLogDeleteAll();
m_DeleteLogs := FALSE;
END_IF
```

ATENÇÃO

O retorno da função *UserLogDeleteAll* não indica operação concluída, apenas confirmação de execução que pode levar um período de tempo grande caso existam centenas de arquivos de log no diretório. A função para registro de novos logs de usuário estará indisponível nesse momento, retornando a opção *USER_LOG_PROCESSING* para qualquer operação. O resultado da operação também pode ser verificado no log de sistema.

5.12.6. ClearRtuDiagnostic

Esta função não é suportada por esta série de UCPs.

5.12.7. ClearEventQueue

A função *ClearEventQueue* disponibilizada pela biblioteca *LibRtuStandard* pode ser utilizada quando for necessário limpar a fila de eventos da UCP e de todos os drivers instanciados.

Conforme a tabela abaixo, o resultado de execução da função será mostrado na sua variável de saída.

ATENÇÃO

A função *ClearEventQueue* não se aplica à limpeza da fila de serviço de Registro de Eventos (SOE), descrito na seção Configuração do SOE. A função limpa apenas as filas de eventos dos drivers configurados sob as interfaces de comunicação (COMs e NETs) da UCP.

Nome	ENUM (UDINT)	Descrição
OK_SUCCESS	0	Sucesso
ERROR_FAILED	1	Erro
ERROR_NOTSUPPORTED	2	A rotina chamada não é suportada pelo produto
ERROR_PARAMETER	3	Parâmetro inválido/não suportado
ERROR_MODULE_ABSENT	16	O módulo está ausente no barramento
ERROR_MODULE_NOTCONFIGURED	17	O módulo não está configurado na aplicação
ERROR_MODULE_NOTRUNNING	18	O módulo não está rodando (não está em estado operacional)
ERROR_MODULE_COMMFAIL	19	Falha na comunicação com o módulo
ERROR_MODULE_NOTFOUND	20	O módulo não foi encontrado na aplicação ou não é suportado

Tabela 195: Resultados da função ClearEventQueue

Exemplo de utilização em linguagem ST, onde a chamada da função irá limpar a fila de eventos e, consequentemente, zerar os diagnósticos de uso das filas de eventos dos drivers de comunicação *T_DIAG_DNP_SERVER_1.tClient_*.tQueueDiags.wUsage:*

```
PROGRAM UserPrg
VAR
  ClearEventQueueStatus : TYPE_RESULT;
END_VAR

ClearEventQueueStatus := ClearEventQueue();
```

5.13. SNMP

5.13.1. Introdução

SNMP (*Simple Network Management Protocol*) é um protocolo muito utilizado por administradores de redes por fornecer importantes informações e diagnósticos de equipamentos presentes em determinada rede Ethernet.

Este protocolo usa o conceito de agente e gerente, no qual o gerente envia requisições de leitura ou escrita de determinados objetos para o agente. Através de uma MIB (*Management Information Base*) o gerente tem conhecimento dos objetos existentes no agente, e dessa forma poderá fazer requisições desses objetos, respeitando as permissões de leitura ou escrita dos mesmos. MIB é uma coleção de informações organizadas hierarquicamente, onde cada objeto desta árvore é chamado de OID (*Object Identifier*).

Para todos os equipamentos com SNMP, é obrigatório o suporte a MIB-II. Nesta MIB estão descritas as principais informações para gerenciamento de redes Ethernet.

5.13.2. SNMP nas UCPs Nexto

As UCPs da Série Nexto se comportam como agentes na comunicação SNMP. As informações disponibilizadas através do SNMP não podem ser manipuladas ou acessadas através da aplicação do usuário, sendo necessário um gerente SNMP externo para efetuar o acesso. A tabela abaixo contém os objetos disponíveis nas UCPs Nexto. Esta funcionalidade está disponível a partir da versão de firmware 1.4.0.33, das UCPs da Série Nexto, possuindo suporte aos protocolos SNMPv1, SNMPv2c e SNMPv3, além de suporte a MIB-II, onde objetos são descritos na RFC-1213.

OID	Nome	Descrição
1.3.6.1.2.1.1	System	Contém nome, descrição, localização e outras informações de identificação do equipamento.
1.3.6.1.2.1.2	Interfaces	Contém informações das interfaces de rede do equipamento. A tabela ifTable (OID 1.3.6.1.2.1.2.2) possui os índices 6 e 7 disponíveis, pelos quais podem ser visualizadas as estatísticas das interfaces de rede NET 1 e NET 2, respectivamente, das UCPs da série Nexto.
1.3.6.1.2.1.3	At	Contém informações das últimas conexões requisitadas ao agente.
1.3.6.1.2.1.4	Ip	Contém estatísticas de conexões utilizando protocolo IP.
1.3.6.1.2.1.5	Icmp	Contém estatísticas para o protocolo ICMP.
1.3.6.1.2.1.6	Tcp	Contém estatísticas para o protocolo TCP.
1.3.6.1.2.1.7	Udp	Contém estatísticas para o protocolo UDP.
1.3.6.1.2.1.11	Snmp	Contém estatísticas para o protocolo SNMP.

Tabela 196: Objetos da MIB-II - Agente SNMP da Série Nexto

Por padrão, o agente SNMP está ativado, ou seja, o serviço é inicializado no momento em que a UCP é iniciada. O acesso às informações do agente se dá através das interfaces Ethernet das UCPs da Série Nexto na porta UDP 161. Portanto quando o serviço estiver ativo, as informações do agente poderão ser acessadas através de qualquer uma das interfaces Ethernet, conforme a disponibilidade da UCP em uso. Não é possível disponibilizar informações do agente através de interfaces Ethernet dos módulos NX5000. Na figura abaixo é mostrado um exemplo de gerente SNMP, no qual são lidos alguns valores.

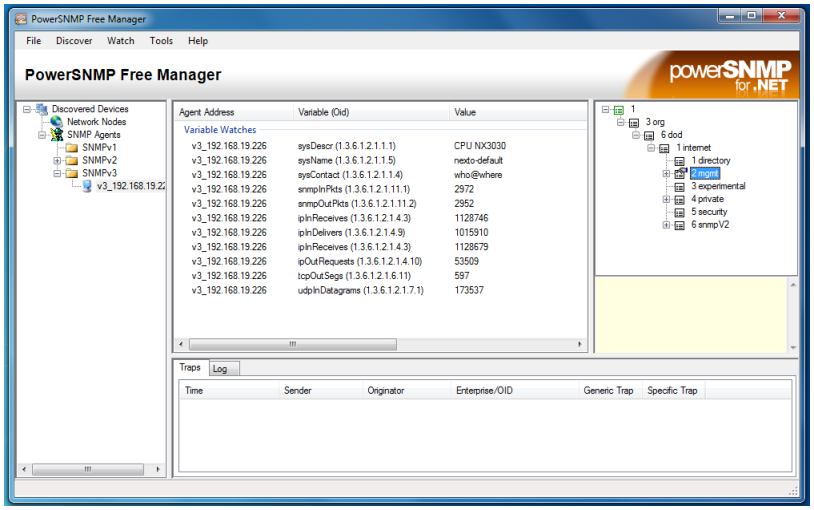


Figura 153: Exemplo de Gerente SNMP

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

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

5.13.3. Private MIB

O Private MIB foi descontinuado desde Junho de 2019.

5.13.4. Configuração SNMP

As configurações do SNMP podem ser alteradas através da Página Web de Sistema da UCP, na aba *Gerenciamento*, na seção *SNMP*. Efetuado o login com sucesso, poderá ser visualizado o estado atual do serviço (ativado ou desativado), assim como as informações de usuário SNMPv3 e comunidades para SNMPv1/v2c.

O usuário poderá ativar ou desativar o serviço através de um checkbox na parte superior da tela.

É possível também alterar as informações de SNMPv3, clicando no botão *Alterar* logo abaixo das informações de usuário. Abrirá um formulário onde é necessário preencher o usuário e senha antigos, e o novo usuário e senha. As demais informações de usuário SNMPv3 não podem ser alteradas.

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

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

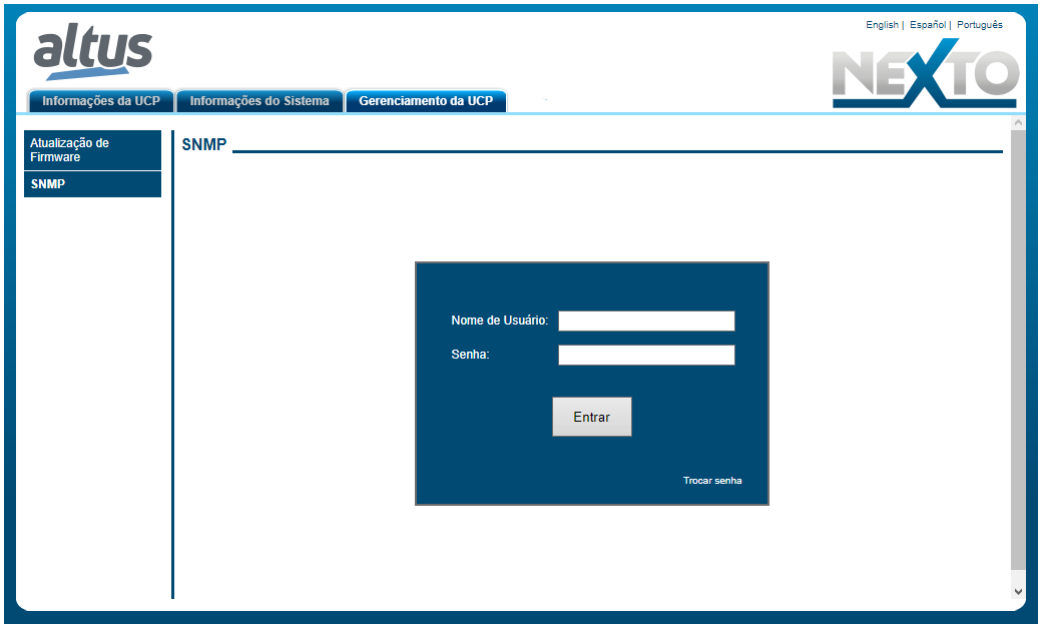


Figura 154: Tela de Login SNMP

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.

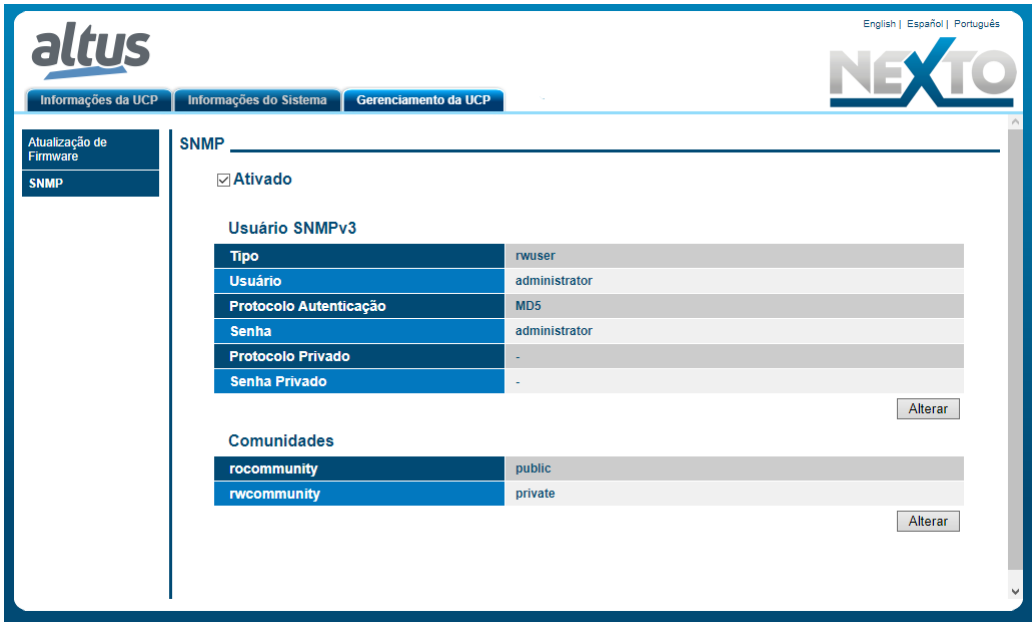


Figura 155: Tela de Configuração e Status SNMP

ATENÇÃO

O *Usuário* e *Senha* para acesso ao agente via protocolo SNMP são os mesmos utilizados para login na seção SNMP.

5.13.5. Usuário e Comunidades SNMP

Para acesso ao SNMPv1/v2c das UCPs da Série Nexto, existem duas comunidades, conforme tabela abaixo.

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

Tabela 197: 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 198: Informações de Usuário Padrão SNMPv3

Para todas as configurações de comunidades, usuário e senha existem limites que devem ser observados na tabela abaixo.

Item configurável	Tamanho mínimo	Tamanho máximo	Caracteres permitidos
rocommunity	-	30	[0-9][a-z][A-Z]@$*_.
rwcommunity	-	30	[0-9][a-z][A-Z]@$*_.
Usuário v3	-	30	[0-9][a-z][A-Z]@$*_.
Senha v3	8	30	[0-9][a-z][A-Z]@$*_.

Tabela 199: Limites para Configurações SNMP

6. Manutenção

Uma das características da Série Nexto é a geração de diagnósticos de comportamentos atípicos, sejam eles falhas, erros ou modos de operação, possibilitando ao operador identificar e solucionar problemas que venham a ocorrer com o sistema com grande facilidade.

As UCPs Nexto proporcionam diversas maneiras de se visualizar os diagnósticos gerados pelo sistema, são elas:

- One Touch Diag
- Diagnósticos via LED
- Diagnósticos via Página Web de Sistema
- Diagnósticos via Variáveis
- Diagnósticos via Blocos Funcionais

O primeiro deles é uma característica inovadora da Série Nexto, a qual possibilita um rápido acesso às condições anormais da aplicação. O segundo é puramente visual, gerado através de dois LEDs presentes no painel (DG e WD) e também dos LEDs presentes no conector RJ45 (exclusivos para a conexão Ethernet). A próxima característica é a visualização gráfica do bastidor e dos respectivos módulos configurados na aba *Informações do Barramento* da Página Web de Sistema, sendo permitido o acesso individual do estado de operação e os diagnósticos ativos. Os diagnósticos também são armazenados diretamente em variáveis da UCP, tanto de representação direta (%Q), como as simbólicas mapeadas através da diretiva AT, e podem ser utilizados pela aplicação do usuário, por exemplo, sendo apresentados em um sistema de supervisão. Os últimos apresentam condições específicas de funcionamento do sistema.

A função destes diagnósticos é apontar possíveis problemas de instalação ou configuração do sistema, e de problemas ou deficiências das redes de comunicação. O capítulo de Manutenção deve ser consultado pelo usuário sempre que necessário.

6.1. Diagnósticos do Módulo

6.1.1. One Touch Diag

One Touch Diag (OTD), ou seja, diagnóstico com um único pressionamento, é uma característica exclusiva que a Série Nexto traz para os controladores programáveis. Com este novo conceito, o usuário pode verificar os diagnósticos de qualquer módulo presente no sistema diretamente no visor gráfico da UCP, com um único pressionamento na tecla de diagnóstico do respectivo módulo. Essa é uma poderosa ferramenta de diagnóstico que pode ser usada offline (sem sistema de supervisão ou software programador), tornando mais fácil encontrar e resolver rapidamente possíveis problemas.

A tecla de diagnóstico está localizada na parte superior da UCP, em local de fácil acesso, e, além de fornecer os diagnósticos ativos, permite o acesso ao menu de navegação, descrito no capítulo Configuração – Menu Informativo e de Configuração da UCP.

A figura abaixo mostra a localização da tecla na UCP:

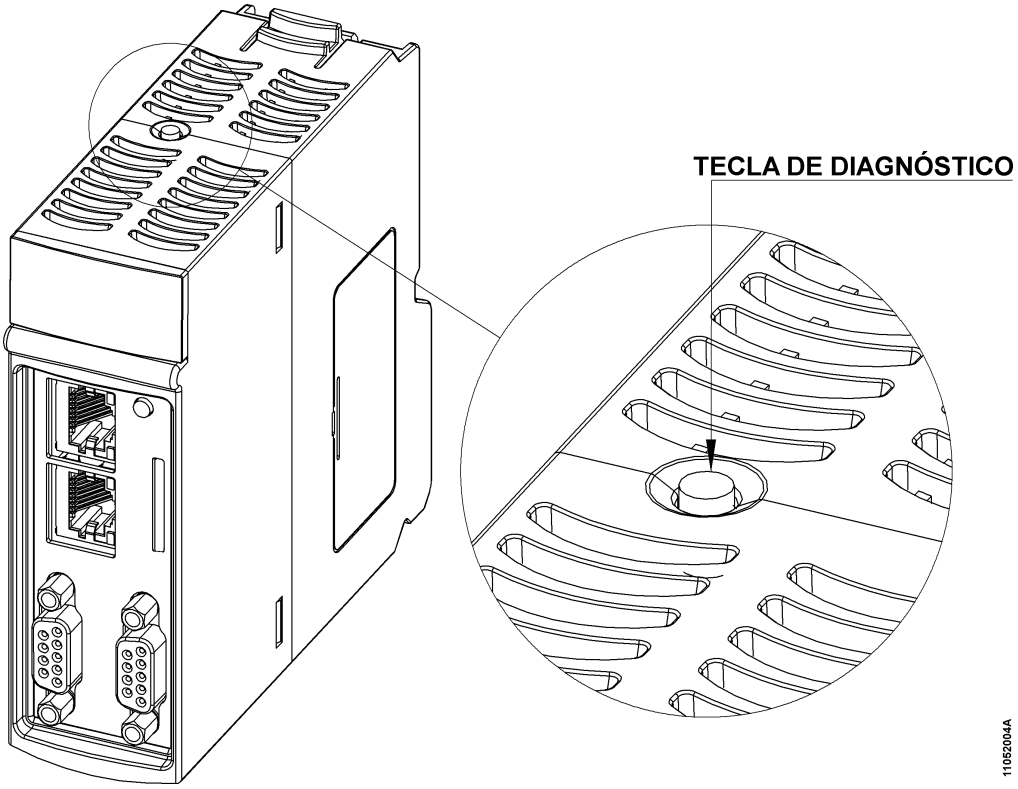


Figura 156: Tecla de Diagnóstico

Com apenas um pressionamento curto, a UCP começa a mostrar os diagnósticos do barramento (quando ativos, caso contrário exibe a mensagem "SEM DIAG"). Inicialmente, será visualizada a Tag (configurada nas propriedades do módulo no software Mastertool, seguindo a IEC 61131-3), ou seja, o nome atribuído à UCP, em seguida, serão mostrados todos os diagnósticos, através de mensagens no visor da UCP. Esse processo será executado por duas vezes no visor. Tudo ocorre de forma automática, sendo que o usuário somente deverá executar o pressionamento curto inicial e a UCP será responsável por exibir os diagnósticos. Os diagnósticos de outros módulos presentes no barramento também serão exibidos no visor gráfico da UCP, através de um pressionamento curto na tecla de diagnóstico dos mesmos, no mesmo modelo da apresentação dos diagnósticos da UCP.

A figura abaixo mostra todo o processo a partir do pressionamento curto, sendo a condição e os tempos da UCP representados nos retângulos menores. É importante salientar que os diagnósticos poderão ter mais de uma tela, ou seja, o tempo especificado no fluxograma abaixo é válido para cada uma delas.

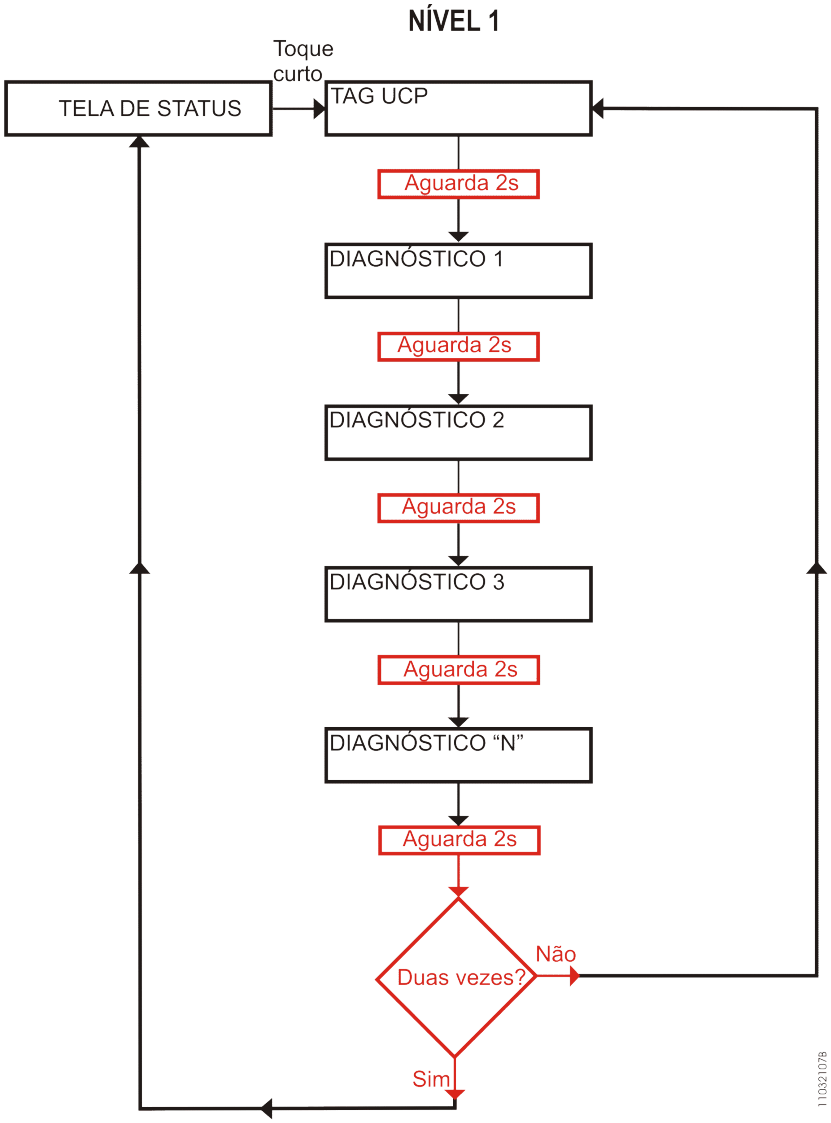


Figura 157: Visualização dos Diagnósticos da UCP

Para finalizar, antes de todo o processo de visualização ser efetuado, basta dar um pressionamento curto na tecla de diagnóstico, em qualquer momento, ou pressionar a tecla diagnóstico de algum módulo de E/S presente no barramento. Também, é importante notar que o One Touch Diag só estará disponível quando o módulo estiver em modo operacional.

Caso seja efetuado um pressionamento longo, a UCP entrará no menu de navegação, o qual está descrito no capítulo Configuração – Menu Informativo e de Configuração da UCP.

A tabela abaixo mostra a diferença entre os tempos do pressionamento curto, pressionamento longo e tecla presa.

Tipo de Pressionamento	Tempo Mínimo	Tempo Máximo	Condição para Indicação
Sem pressionamento	-	59,99 ms	-
Pressionamento curto	60 ms	0,99 s	Liberação
Pressionamento longo	1 s	20 s	Mais do que 1 s até 20 s
Tecla Presa	20,01 s	(∞)	Indicação em diagnóstico, consultar a Tabela 205

Tabela 200: Tempo de um Pressionamento

As mensagens exibidas no visor gráfico das UCPs Nexto, correspondentes aos diagnósticos, estão descritas na seção Diagnósticos via Variáveis, na Tabela 205.

Caso ocorra uma situação de tecla presa de algum módulo de E/S presente no barramento, a tecla de diagnóstico deste deixará de indicar diagnósticos no visor gráfico da UCP quando for pressionada, neste caso a UCP irá indicar que há módulos com diagnóstico ativo. Para que seja possível eliminar este diagnóstico da UCP, será necessário realizar uma troca a quente do módulo com o diagnóstico ativo.

Para mais detalhes sobre o procedimento de visualização dos diagnósticos da UCP ou de outros módulos do barramento, ver descrição no Manual de Utilização Série Nexto – MU214000.

6.1.2. Diagnósticos via LED

Este produto possui um LED para indicação de diagnóstico (LED DG) e um LED para indicar ocorrência de cão-de-guarda (LED WD). A Tabela 201 e Tabela 202 mostram o significado de cada estado e suas respectivas descrições.

6.1.2.1. DG (Diagnóstico)

Verde	Vermelho	Descrição	Causas	Prioridade
Desligado	**Desligado**	Não utilizado	Sem fonte de alimentação. Problema de Hardware	-
Ligado	**Desligado**	Aplicação em Execução (Modo Run)	-	3 (Baixa)
Desligado	**Ligado**	Aplicação Parada (Modo Stop)	-	3 (Baixa)
Piscando 2x	**Desligado**	Módulos do barramento com diagnóstico	No mínimo um módulo do barramento, incluindo a UCP, está com algum diagnóstico ativo	1
Piscando 3x	**Desligado**	Forçamento de dados	Alguma área de memória está sendo forçada pelo usuário via MasterTool IEC XE	2
Desligado	**Piscando 4x**	Erro de configuração ou de hardware no barramento.	O barramento está danificado ou não está configurado corretamente.	0 (Alta)

Tabela 201: Descrição dos Estados do LED de Diagnóstico

6.1.2.2. WD (Cão-de-guarda)

LED Vermelho	Descrição	Causas	Prioridade
Desligado	Sem indicação de cão-de-guarda	Operação normal	3 (Baixa)
Piscando 1x	Cão-de-guarda de Software	Cão-de-guarda da aplicação de usuário	2
Ligado	Cão-de-guarda de Hardware	Módulo danificado e/ou sistema operacional corrompido	1 (Alta)

Tabela 202: Descrição dos Estados do LED de Cão-de-guarda

Notas:

Cão-de-guarda de software: Para remover a indicação de cão-de-guarda, deve-se efetuar um reset da aplicação ou desligar e ligar novamente a UCP. Esse cão-de-guarda ocorre quando o tempo de execução da aplicação de usuário for maior que o tempo de cão-de-guarda configurado.

O diagnóstico pode ser verificado no operando Exception.wExceptionCode, ver Tabela 209.

Cão-de-guarda de hardware: Para limpar qualquer indicação de cão-de-guarda, como no LED WD ou no operando Reset.bWatchdogReset, o módulo deve ser desconectado da fonte de alimentação.

Para verificar as condições da aplicação na reinicialização do módulo, ver configurações na Tabela 43.

6.1.2.3. LEDs Conector RJ45

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

Amarelo	Verde	Descrição
○	○	Ausência de LINK de rede
●	○	LINK de rede de 10 Mbits/s
●	●	LINK de rede de 100 Mbits/s
X	-	Ocorrência de transmissão ou recepção na rede Ethernet, pelo ou para este endereço IP. Pisca sob demanda da UCP Nexto, e não a cada transmissão ou recepção, ou seja, pode piscar com uma frequência menor que a frequência real de transmissão ou recepção.

Tabela 203: Significado dos LEDs Ethernet

6.1.3. Diagnósticos via Página Web de Sistema

Além das características apresentadas anteriormente, a Série Nexto traz para o usuário uma ferramenta inovadora de acesso aos diagnósticos e estados de operação do sistema, através da aba *Informações do Barramento* da Página Web de Sistema.

A utilização, além de dinâmica, é bastante intuitiva e facilita as operações do usuário. Em outras palavras, pode substituir o uso de um sistema de supervisão quando o uso for restrito a verificação de status do sistema.

Para acessar a Página Web de Sistema da UCP desejada, basta utilizar um navegador para desktop e digitar, na barra de endereço, o endereço IP correspondente à UCP (Ex.: http://192.168.1.1). Inicialmente, serão apresentadas as informações da UCP, conforme mostra a Figura 158:

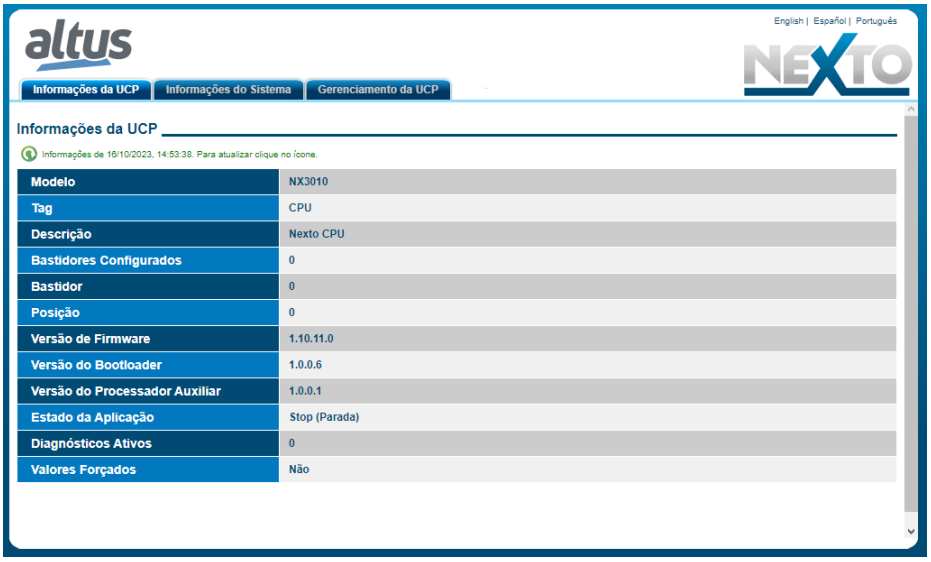


Figura 158: Tela Inicial

Também existe a aba "*Informações do Barramento*", a qual pode ser visualizada através do Bastidor ou da lista dos módulos presentes (opção do lado direito da tela). Quando não houver nenhuma aplicação na UCP, será exibida nesta página uma configuração com o maior Bastidor disponível e uma fonte de alimentação padrão, juntamente com a UCP conectada. Quando a visualização pelo Bastidor é utilizada, os módulos que têm diagnóstico ficam piscando e assumem a cor vermelha, conforme mostra a Figura 159. Caso contrário será exibida uma lista com os módulos presentes no sistema, Tags correspondentes e número de diagnósticos ativos:

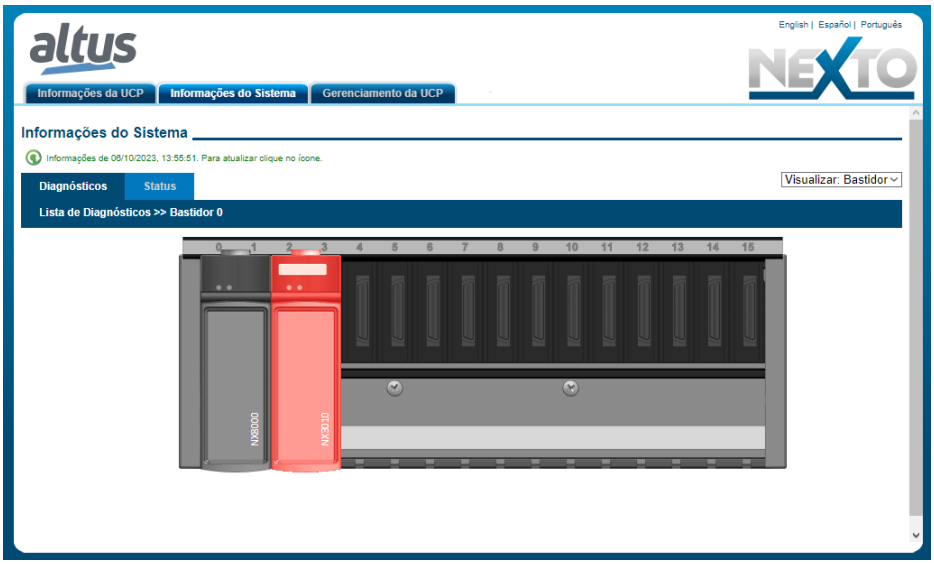


Figura 159: Informações do Barramento

Ao clicar no módulo com diagnóstico, no mesmo instante são mostrados os diagnósticos ativos do módulo, conforme mostra a Figura 160:

ATENÇÃO

Quando uma UCP for reiniciada e a aplicação entrar em exceção na partida do sistema, os diagnósticos não estarão válidos. É necessário corrigir o problema que gera a exceção da aplicação para que os diagnósticos sejam atualizados.

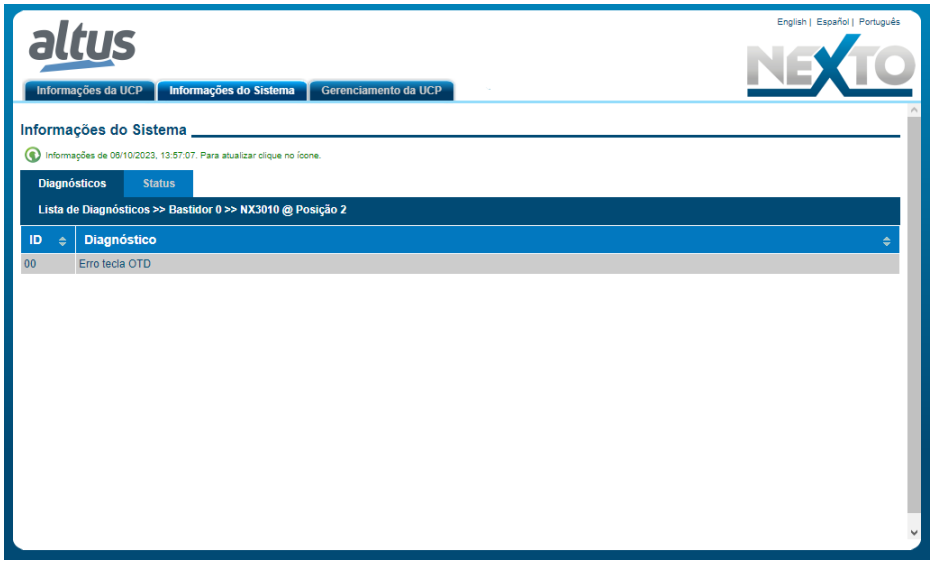


Figura 160: Diagnósticos do Sistema

Caso a seção Status seja selecionada, o estado de todos os diagnósticos detalhados é exibido na tela, conforme mostra a Figura 161:

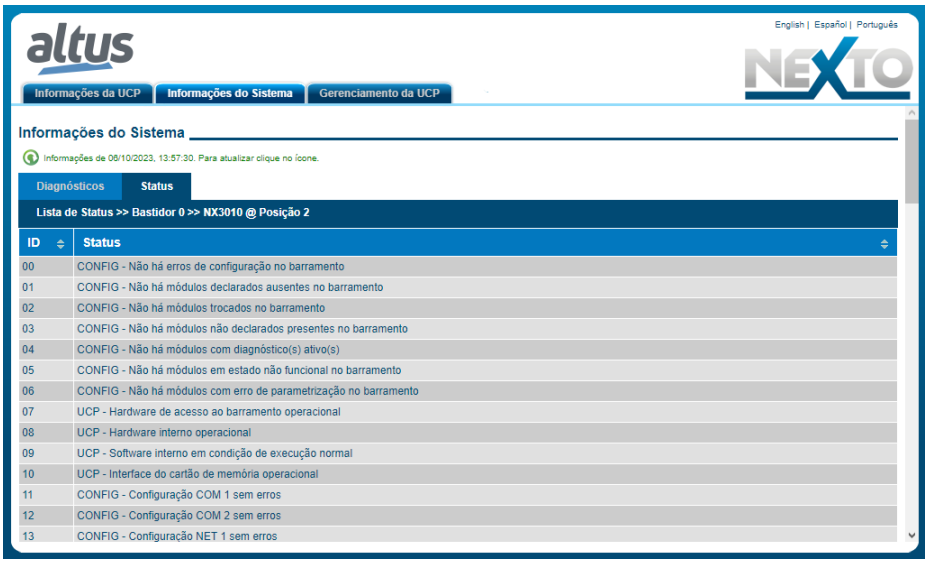


Figura 161: Estado do Sistema

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

6.1.4. Diagnósticos via Variáveis

As UCPs da Série Nexto possuem variáveis para indicação de diagnóstico. Existem estruturas de dados com os diagnósticos de todos os módulos declarados no barramento, mapeadas sobre variáveis de representação direta %Q, e definidas simbolicamente através da diretiva AT, na GVL System_Diagnostics criada automaticamente pelo MasterTool IEC XE.

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

Byte	Descrição
0 a 3	Diagnósticos resumidos da UCP.
4 a 1615	Diagnósticos detalhados da UCP.

Tabela 204: Divisão dos Diagnósticos da UCP

6.1.4.1. Diagnósticos Resumidos

A tabela abaixo mostra o significado de cada bit dos diagnósticos resumidos da UCP:

Mensagem de Diagnóstico	Variável DG_Modulo.tSummarized.*	Descrição
SEM DIAG	-	Não existe diagnóstico ativo.
CONFIG. INCOMPATIVEL	bConfigMismatch	TRUE – Existe algum problema de configuração no barramento, como módulo em posição incorreta.
		FALSE – O barramento está configurado corretamente.
MODULOS AUSENTES	bAbsentModules	TRUE – Um ou mais módulos declarados estão ausentes.
		FALSE – Todos os módulos estão presentes no barramento.
MODULOS TROCADOS	bSwappedModules	TRUE – Dois módulos estão trocados entre si no barramento.
		FALSE – Não há módulos trocados no barramento.
MODULOS NAO DECLARADOS	bNonDeclaredModules	TRUE – Um ou mais módulos presentes no barramento não estão declarados.
		FALSE – Todos os módulos presentes no barramento estão declarados.
MODULOS C/ DIAGNOSTICO	bModulesWithDiagnostic	TRUE – Um ou mais módulos do barramento estão com diagnóstico ativo.
		FALSE – Não existem diagnósticos ativos nos módulos do barramento.
MODULOS C/ ERRO FATAL	bModuleFatalError	TRUE – Um ou mais módulos presentes no barramento estão em estado não funcional.
		FALSE – Todos os módulos presentes no barramento estão em estado funcional.
MODULOS C/ ERRO PARAM.	bModuleParameterError	TRUE – Um ou mais módulos do barramento estão com erro de parametrização.
		FALSE – Todos os módulos estão parametrizados.
ERRO NO BARRAMENTO	bWHSBBusError	TRUE – Indicação do mestre que existe falha no barramento WHSB.
		FALSE – O barramento WHSB está em correto funcionamento.
FALHA DE HARDWARE	bHardwareFailure	TRUE – Falha no hardware da UCP.
		FALSE – O hardware está em correto funcionamento.
EXCECAO NO SOFTWARE	bSoftwareException	TRUE – Uma ou mais exceções geradas pelo software.
		FALSE – Não foram geradas exceções no software.
ERRO CARTAO DE MEMORIA	bMemoryCardError	TRUE – O cartão de memória está inserido na UCP, porém não está funcionando corretamente.
		FALSE – O cartão de memória está funcionando corretamente.
ERRO CONFIG. COM 1	bCOM1ConfigError	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface serial COM 1.
		FALSE – A configuração da interface serial COM 1 está correta.

Mensagem de Diagnóstico	Variável DG_Modulo.tSummarized.*	Descrição
ERRO CONFIG. COM 2	bCOM2ConfigError	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface serial COM 2.
		FALSE – A configuração da interface serial COM 2 está correta.
ERRO CONFIG. NET 1	bNET1ConfigError	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface Ethernet NET 1.
		FALSE – A configuração da interface Ethernet NET 1 está correta.
ERRO CONFIG. NET 2	bNET2ConfigError	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface Ethernet NET 2.
		FALSE – A configuração da interface Ethernet NET 2 está correta.
DATA/HORA INVALIDA	bInvalidDateTime	TRUE – A data ou a hora são inválidas.
		FALSE – A data e a hora estão corretas.
RUNTIME RESET	bRuntimeReset	TRUE – O RTS (Runtime System) foi reiniciado pelo menos uma vez. Esse diagnóstico somente é limpo na reinicialização do sistema.
		FALSE – O RTS (Runtime System) está operando normalmente.
PERDA DE RETENTIVIDADE	bRetentivityLost	TRUE – Ocorreu um erro ao salvar os dados retentivos.
		FALSE – Dados válidos na memória retentiva durante a inicialização.
ERRO TECLA OTD	bOTDSwitchError	TRUE –A tecla ficou travada por mais de 20 s pelo menos uma vez enquanto a UCP esteve energizada. Esse diagnóstico somente é limpo na reinicialização do sistema.
		FALSE – A tecla não está ou ficou travada enquanto a UCP esteve energizada.
BASTIDOR AUSENTE	bAbsentRacks	TRUE – Um ou mais bastidores declarados estão ausentes.
		FALSE – Todos os bastidores declarados estão presentes.
BASTIDOR DUPLICADO	bDuplicatedRacks	TRUE – Há algum bastidor com o número de identificação duplicado.
		FALSE – Não existem bastidores com o número de identificação duplicado.
BASTIDOR INVALIDO	bInvalidRacks	TRUE – Há algum bastidor com o número de identificação inválido.
		FALSE – Não existem bastidores com o número de identificação inválido.
BASTIDOR NAO DECLARADO	bNonDeclaredRacks	TRUE – Há algum bastidor com o número de identificação não declarado.
		FALSE – Não existem bastidores com o número de identificação não declarado.
SLOT DUPLICADO	bDuplicatedSlots	TRUE – Há algum endereço de slot duplicado.
		FALSE – Não existem endereços de slot duplicados.

Tabela 205: Diagnósticos Resumidos da UCP

Notas:

Configuração incompatível: O diagnóstico de configuração incompatível é gerado caso um ou mais módulos do barramento não conferirem com o que foi declarado, ou seja, nas condições de módulos ausentes ou diferentes. Os módulos inseridos no barramento que não foram declarados no projeto não são considerados.

Módulos trocados: Se somente dois módulos estiverem trocados entre si no respectivo barramento, então o diagnóstico de troca pode ser identificado. Caso contrário, o problema será tratado como "Configuração Incompatível".

Módulos com erro fatal: Caso o diagnóstico de Módulos com Erro Fatal seja verdadeiro, verificar qual é o módulo que está com problema no barramento e encaminhar para a Assistência Técnica da Altus, pois o mesmo está apresentando falha do hardware.

Módulos com erro de parametrização: Caso o diagnóstico de Módulo com Erro de Parametrização seja verdadeiro, verificar se os módulos do barramento estão configurados corretamente e se as versões de firmware e do software MasterTool IEC XE estão adequadas. Se o problema ocorrer ao inserir um módulo no barramento, verifique se esse módulo suporta troca à quente.

Erro no barramento: Considerado um erro fatal, interrompendo o acesso aos módulos do barramento. Caso o diagnóstico de erro no barramento seja verdadeiro, talvez haja um problema de hardware nas linhas de comunicação do barramento, sendo assim, deve-se entrar em contato com a Assistência Técnica da Altus.

Falha de hardware: Caso o diagnóstico de Falha de Hardware seja verdadeiro, encaminhar a UCP para Assistência Técnica da Altus, pois a mesma apresenta problemas no RTC, processador auxiliar, ou outros recursos de hardware.

Exceção no software: Caso o diagnóstico de exceção no software seja verdadeiro, o usuário deverá verificar a sua aplicação para garantir que a mesma não esteja acessando indevidamente a memória. Se o problema persistir, o setor de Suporte da Altus deverá ser consultado. Os códigos de exceção no software estão descritos após a tabela de diagnósticos detalhados da UCP.

Mensagem de Diagnóstico: As mensagens de diagnóstico podem ser visualizadas através do visor gráfico da UCP através da tecla OTD ou através da aba de diagnósticos da Página Web de Sistema da UCP.

6.1.4.2. Diagnósticos Detalhados

As tabelas a seguir mostram os diagnósticos detalhados das UCPs da série Nexto, para sua consulta é importante verificar as observações abaixo:

- **Visualização das Estruturas de Diagnóstico:** As Estruturas de Diagnóstico adicionadas ao projeto podem ser visualizadas acessando o item "*Library Manager*" na árvore de dispositivos da janela do Mastertool. Com isso, é possível visualizar todos os tipos de dados definidos na estrutura.
- **Contadores:** Todos os contadores dos diagnósticos da UCP retornam à zero quando o seu valor limite é ultrapassado.
- *DG_Modulo.*, sendo a palavra *Modulo* substituída pelo produto utilizado.

DG_Modulo.tDetailed.Target.*	Tamanho	Descrição
dwCPUModel	DWORD	Modelo da UCP.
abyCPUVersion	BYTE ARRAY(4)	Versão do firmware.
abyBootloaderVersion	BYTE ARRAY(4)	Versão do bootloader.
abyAuxprocVersion	BYTE ARRAY(4)	Versão do processador auxiliar.

Tabela 206: Diagnósticos Detalhados Grupo Target

DG_Modulo.tDetailed.Hardware.*	Tamanho	Descrição
bAuxprocFailure	BIT	Falha na comunicação entre o processador auxiliar e o processador principal.
bRTCFailure	BIT	O processador principal não está habilitado para comunicar com o RTC (relógio da UCP).
bThermometerFailure	BIT	Falha na comunicação entre o termômetro e o processador principal.
bLCDFailure	BIT	Falha na comunicação entre o visor LCD e o processador principal.

Tabela 207: Diagnósticos Detalhados Grupo Hardware

DG_Modulo.tDetailed.Exception.*	Tamanho	Descrição
wExceptionCode	WORD	Código de Exceção gerado pelo RTS, o significado dos códigos pode ser conferido na Tabela 209.
byProcessorLoad	BYTE	Nível, em percentagem (%), de carga no processador.

Tabela 208: Diagnósticos Detalhados Grupo Exception

Nota:

Código de exceção: O código de exceção gerado pelo RTS (Runtime System) pode ser consultado abaixo:

Código	Descrição	Código	Descrição
0x0000	Não há código de exceção.	**0x0051**	Violação de acesso.
0x0010	Tempo de Cão-de-Guarda da tarefa IEC expirado (Cão-de-Guarda de Software).	**0x0052**	Instrução privilegiada.
0x0012	Erro de configuração de E / S.	**0x0053**	Falha na página.
0x0013	Erros de check-up após o download do programa.	**0x0054**	Estouro de pilha.
0x0014	Erro de rede de campo.	**0x0055**	Disposição inválida.
0x0015	Erro de atualização de E / S.	**0x0056**	Manobra inválida.
0x0016	Tempo de ciclo (execução) excedido.	**0x0057**	Página protegida.
0x0017	Atualização online do programa por muito tempo.	**0x0058**	Falha dupla.
0x0018	Referências externas não resolvidas.	**0x0059**	OpCode inválido.
0x0019	Download rejeitado.	**0x0100**	Desalinhamento do tipo de dado.
0x001A	Projeto não-carregado, pois as variáveis retentivas não podem ser realocadas.	**0x0101**	Limite de arrays excedido.
0x001B	Projeto não-carregado e excluído.	**0x0102**	Divisão por zero.
0x001C	Fora da pilha de memória.	**0x0103**	Estouro.
0x001D	Memória retentiva corrompida; não pode ser mapeada.	**0x0104**	Não continuável.
0x001E	O projeto pode ser carregado, mas causa uma ruptura mais tarde.	**0x0105**	Cão-de-guarda na carga do processador de todas as tarefas IEC detectadas.
0x0021	O objetivo da aplicação de inicialização não corresponde ao destino atual.	**0x0150**	FPU: Erro não especificado.
0x0022	Erro de tarefas agendadas... Falha na configuração da tarefa IEC. Aplicação trabalhando com o alvo errado. Instrução ilegal.	**0x0151**	FPU: Operando não normal.
		0x0152	FPU: Divisão por zero.
0x0023	Erro de check-up do arquivo baixado.	**0x0153**	FPU: Resultado inexato.
0x0024	Incompatibilidade entre a identidade retentiva e a identidade atual do programa de projeto de inicialização	**0x0154**	FPU: Operação inválida.
0x0025	Falha na configuração da tarefa IEC.	**0x0155**	FPU: Estouro.
0x0026	A aplicação está sendo executada com o alvo errado.	**0x0156**	FPU: Verificação da pilha.
0x0050	Instrução ilegal.	**0x0157**	FPU: Underflow.

Tabela 209: Códigos de Exceção RTS

DG_Modulo.tDetailed.RetainInfo.*	Tamanho	Descrição
byCPUInitStatus	BYTE	Status da Inicialização da UCP: 01: Hot start 02: Warm Start 03: Cold Start Obs.: Essas variáveis são reinicializadas em todas as energizações.
wCPUColdStartCounter	WORD	Incrementa caso a UCP perca retentividade durante a inicialização. (0 a 65535)
wCPUWarmStartCounter	WORD	Incrementa caso a UCP inicialize com dados retentivos válidos. (0 a 65535)
wCPUHotStartCounter	WORD	Contador de distúrbios menores do que o tempo de suporte à falhas na alimentação da UCP. (0 a 65535).
wRTSResetCounter	WORD	Contador de resets efetuados pelo RTS (Runtime System). (0 a 65535).

Tabela 210: Diagnósticos Detalhados Grupo RetainInfo

DG_Modulo.tDetailed.Reset.*	Tamanho	Descrição
bBrownOutReset	BIT	A UCP foi reiniciada devido a uma falha na alimentação na última inicialização.

DG_Modulo.tDetailed.Reset.*	Tamanho	Descrição
bWatchdogReset	BIT	A UCP foi reiniciada devido ao cão-de-guarda ativo na última inicialização.

Tabela 211: Diagnósticos Detalhados Grupo Reset

Nota:

Reset por Brownout: O diagnóstico de reset por brownout somente será verdadeiro quando a alimentação da fonte exceder o limite mínimo exigido nas características técnicas da mesma, mantendo-se com tensão baixa, ou seja, sem sofrer uma interrupção. A UCP irá identificar a queda da alimentação e indicará o diagnóstico de falha na alimentação. Quando a tensão for restabelecida, a UCP será reinicializada automaticamente e indicará o diagnóstico de reset por brownout.

DG_Modulo.tDetailed.Thermometer.*	Tamanho	Descrição
bOverTemperatureAlarm	BIT	Alarme gerado devido à temperatura interna estar em 85 °C ou acima de 85 °C.
bUnderTemperatureAlarm	BIT	Alarme gerado devido à temperatura interna estar em 0 °C ou abaixo de 0 °C.
diTemperature	DINT	Temperatura lida no sensor interno da UCP.

Tabela 212: Diagnósticos Detalhados Grupo Thermometer

Nota:

Temperatura: Para a visualização da temperatura diretamente no endereço de memória, deve-se realizar uma conversão, pois o tamanho do dado é DINT e a monitoração é realizada em 4 bytes. Por isso, indica-se a utilização da variável simbólica associada, pois a mesma já fornece o valor final da temperatura.

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 1024).
wTXPendingBytes	WORD	Número de caracteres pendentes no buffer de transmissão na COM 1 (0 a 1024).
wBreakErrorCounter	WORD	O transmissor está mantendo a linha de dados em zero por muito tempo, de acordo com o bit de dados configurado.
wParityErrorCounter	WORD	O frame recebido tem o bit de paridade incompatível.
wFrameErrorCounter	WORD	O frame recebido tem o ponto de partida errado, geralmente causado por uma incompatibilidade de ruído ou baud rate.
wRXOverrunCounter	WORD	Quando o buffer de anel recebido está cheio e começa a perder os frames antigos (muitos frames não tratados pelo dispositivo).

Tabela 213: Diagnósticos Detalhados Grupo Serial COM 1

Nota:

Contador de erro de paridade: Quando a paridade da interface serial COM 1 estiver configurada como Sem Paridade, este contador de erros não será incrementado ao receber uma mensagem com paridade diferente. Nesse caso, será indicado erro de frame.

DG_Modulo.tDetailed.Serial.COM2.*	Tamanho	Descrição
byProtocol	BYTE	Protocolo selecionado na COM 2: 00: Sem protocolo 01: MODBUS RTU Master 02: MODBUS RTU Slave 03: Outro protocolo
dwRXBytes	DWORD	Contador de caracteres recebidos através da COM 2 (0 a 4294967295).
dwTXBytes	DWORD	Contador de caracteres transmitidos através da COM 2 (0 a 4294967295).
wRXPendingBytes	WORD	Número de caracteres pendentes no buffer de leitura na COM 2 (0 a 1024).
wTXPendingBytes	WORD	Número de caracteres pendentes no buffer de transmissão na COM 2 (0 a 1024).
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 214: Diagnósticos Detalhados Grupo Serial COM 2

Nota:

Contador de erro de paridade: Quando a paridade da interface serial COM 2 estiver configurada como Sem Paridade, este contador de erros não será incrementado ao receber uma mensagem com paridade diferente. Nesse caso, será indicado erro de frame.

DG_Modulo.tDetailed.Ethernet.*	Tamanho	Descrição
NET[x].bLinkDown	BIT	Indica o estado do link na NET[x].
NET[x].wProtocol	WORD	Protocolo selecionado na NET[x]: 00: Sem protocolo
NET[x].szIP	STRING (15)	Endereço IP NET[x].
NET[x].szMask	STRING (15)	Máscara de Subrede NET[x].
NET[x].szGateway	STRING (15)	Endereço Gateway NET[x].
NET[x].szMAC	STRING (17)	Endereço MAC NET[x].
NET[x].abyIP	BYTE ARRAY(4)	Endereço IP NET[x].
NET[x].abyMask	BYTE ARRAY(4)	Máscara de Subrede NET[x].
NET[x].abyGateway	BYTE ARRAY(4)	Endereço Gateway NET[x].
NET[x].abyMAC	BYTE ARRAY(6)	Endereço MAC NET[x].
NET[x].dwPacketsSent	DWORD	Contador de pacotes enviados através da porta NET[x] (0 a 4294967295).
NET[x].dwPacketsReceived	DWORD	Contador de pacotes recebidos através da porta NET[x] (0 a 4294967295).
NET[x].dwBytesSent	DWORD	Contador de bytes enviados através da porta NET[x] (0 a 4294967295).
NET[x].dwBytesReceived	DWORD	Contador de bytes recebidos através da porta NET[x] (0 a 4294967295).
NET[x].wTXErrors	WORD	Contador de erros de transmissão através da porta NET[x] (0 a 65535).
NET[x].wTXFIFOErrors	WORD	Contador de erros no buffer de transmissão através da porta NET[x] (0 a 65535).
NET[x].wTXDropErrors	WORD	Contador de perdas de conexão na transmissão através da porta NET[x] (0 a 65535).
NET[x].wTXCollisionErrors	WORD	Contador de erros de colisão na transmissão através da porta NET[x] (0 a 65535).
NET[x].wTXCarrierErrors	WORD	Contador de erros de transporte na transmissão através da porta NET[x] (0 a 65535).
NET[x].wRXErrors	WORD	Contador de erros de recepção através da porta NET[x] (0 a 65535).

DG_Modulo.tDetailed.Ethernet.*	Tamanho	Descrição
NET[x].wRXFIFOErrors	WORD	Contador de erros no buffer de recepção através da porta NET[x] (0 a 65535).
NET[x].wRXDropErrors	WORD	Contador de perdas de conexão na recepção através da porta NET[x] (0 a 65535).
NET[x].wRXFrameErrors	WORD	Contador de erros de frame na recepção através da porta NET[x] (0 a 65535).
NET[x].wMulticast	WORD	Contador de pacotes multicast através da porta NET[x] (0 a 65535).

Tabela 215: Diagnósticos Detalhados Grupo Ethernet NET

Nota:
[x] é o número da interface ethernet da UCP, onde por exemplo, NET 1 representa o valor 1.

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 216: Diagnósticos Detalhados Grupo UserFiles

Nota:
UserFiles: A partição de usuário é uma área da memória reservada para o armazenamento de dados na UCP. Por exemplo: arquivos com extensão PDF, arquivos com extensão DOC e demais dados.

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 217: Diagnósticos Detalhados Grupo UserLogs

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
bMemcardtoCPUEnabled	BIT	Nível de Proteção do Cartão de Memória: Leitura de dados do cartão de memória pela UCP autorizada.
bCPUtoMemcardEnabled	BIT	Escrita de dados no cartão de memória pela UCP autorizada.
dwFreeSpacekB	DWORD	Espaço livre no Cartão de Memória em Kbytes.
dwTotalSizekB	DWORD	Capacidade de armazenamento do Cartão de Memória em Kbytes.

Tabela 218: Diagnósticos Detalhados Grupo MemoryCard

DG_Modulo.tDetailed.WHSB.*	Tamanho	Descrição
byHotSwapAndStartupStatus	BYTE	Informa a situação anormal no barramento que ocasionou a parada da aplicação, para cada Modo de troca a quente. Consulte a Tabela 220 para detalhes das possibilidades.

DG_Modulo.tDetailed.WHSB.*	Tamanho	Descrição
adwRackIOErrorStatus	DWORD ARRAY (32)	Identificação de erros em módulos de E/S, individualmente. Para mais informações sobre este diagnóstico consultar a nota abaixo.
adwModulePresenceStatus	DWORD ARRAY (32)	Status de presença de módulos declarados de E/S em barramentos, individualmente. Para mais informações sobre este diagnóstico consultar a nota abaixo.
byWHSBBusErrors	BYTE	Contador de falhas no barramento WHSB. Esse contador é reinicializado na energização (0 a 255).

Tabela 219: Diagnósticos Detalhados Grupo WHSB

Notas:

Diagnóstico de erro dos módulos do barramento: Cada DWORD do Array deste diagnóstico representa um bastidor, cujas posições são representadas pelos bits destas DWORDs. Logo, o Bit-0 da DWORD-0 equivale a posição zero do bastidor com endereço zero. Cada um desses Bits é o resultado de uma operação lógica OU entre os diagnósticos de configuração incompatível (bConfigMismatch), módulos ausentes (bAbsentModules), módulos trocados (bSwappedModules), módulos com erro fatal (bModuleFatalError) e o estado operacional do módulo de uma determinada posição.

Status de presença de módulos: Cada DWORD do Array deste diagnóstico representa um bastidor, cujas posições são representadas pelos bits destas DWORDs. Logo, o Bit-0 da DWORD-0 equivale a posição zero do bastidor com endereço zero. Então, se o módulo estiver presente, este bit será verdadeiro. É importante ressaltar que esse diagnóstico é válido para todos os módulos, exceto fontes de alimentação e UCPs, ou seja, não apresentam a presença no barramento em suas respectivas posições (bit permanece em falso).

Situações que ocasionam parada da aplicação: Os códigos das possíveis situações que ocasionam parada da aplicação podem ser consultados abaixo:

Código	Enumerável	Descrição
00	INITIALIZING	Esse estado é apresentado enquanto os demais estados não estiverem prontos.
01	RESET_WATCHDOG	Aplicação em Modo Stop devido ao reset por cão-de-guarda de hardware ou por uma reinicialização do Runtime, quando desabilitada a configuração *"Iniciar Aplicação de Usuário após Reset por Cão-de-Guarda"*.
02	ABSENT_MODULES_HOT_SWAP_ DISABLED	Aplicação em Modo Stop devido à ativação do diagnóstico Módulos Ausentes, quando configurada troca a quente *"Desabilitada"* ou troca a quente *"Desabilitada, apenas para módulos declarados"*.
03	CFG_MISMATCH_HOT_SWAP_ DISABLED	Aplicação em Modo Stop devido à ativação do diagnóstico Configuração Incompatível, quando configurada troca a quente *"Desabilitada"* ou troca a quente *"Desabilitada, apenas para módulos declarados"*.
04	ABSENT_MODULES_HOT_SWAP_ STARTUP_CONSISTENCY	Aplicação em Modo Stop devido à ativação do diagnóstico Módulos Ausentes, quando configurada troca a quente *"Habilitada, com consistência na partida"* ou troca a quente *"Habilitada, com consistência na partida somente para os módulos declarados"*.
05	CFG_MISMATCH_HOT_SWAP_ STARTUP_CONSISTENCY	Aplicação em Modo Stop devido à ativação do diagnóstico Configuração Incompatível, quando configurada troca a quente *"Habilitada, com consistência na partida"* ou troca a quente *"Habilitada, com consistência na partida somente para os módulos declarados"*.
06	APPL_STOP_ALLOWED_TO_RUN	Aplicação em Modo Stop e todas as consistências realizadas com sucesso. Aplicação pode ser colocada no Modo Run.
07	APPL_STOP_MODULES_NOT_READY	Aplicação em Modo Stop e todas as consistências realizadas com sucesso, mas os módulos de E/S não estão aptos para a partida do sistema. Não é possível colocar a Aplicação em Modo Run.
08	APPL_STOP_MODULES_GETTING_ READY_TO_RUN	Aplicação em Modo Stop e todas as consistências realizadas com sucesso. Os módulos de E/S estão sendo preparados para a partida do sistema. Não é possível colocar a Aplicação em Modo Run.
09	NORMAL_OPERATING_STATE	Aplicação em Modo Run.
10	MODULE_CONSISTENCY_OK	Uso interno.
11	APPL_STOP_DUE_TO_EXCEPTION	Aplicação em modo Stop pois uma exceção ocorreu na UCP.

Código	Enumerável	Descrição
12	DUPLICATED_SLOT_HOT_SWAP_ DISABLED	Aplicação em Modo Stop devido à ativação do diagnóstico Slots Duplicados, quando configurada troca a quente *"Desabilitada"* ou troca a quente *"Desabilitada, apenas para módulos declarados"*.
13	DUPLICATED_SLOT_HOT_SWAP_ STARTUP_CONSISTENCY	Aplicação em Modo Stop devido à ativação do diagnóstico Slots Duplicados, quando configurada troca a quente *"Habilitada, com consistência na partida"* ou troca a quente *"Habilitada, com consistência na partida somente para os módulos declarados"*.
14	DUPLICATED_SLOT_HOT_SWAP_ ENABLED	Aplicação em Modo Stop devido à ativação do diagnóstico Slots Duplicados, quando configurada troca a quente *"Habilitada, sem consistência na partida"*.
15	NON_DECLARED_MODULE_HOT_ SWAP_STARTUP_CONSISTENCY	Aplicação em Modo Stop devido à ativação do diagnóstico Módulos Não Declarados, quando configurada troca a quente *"Habilitada, com consistência na partida"*.
16	NON_DECLARED_MODULE_HOT_ SWAP_DISABLED	Aplicação em Modo Stop devido à ativação do diagnóstico Módulos Não Declarados, quando configurada troca a quente *"Desabilitada"*.

Tabela 220: Códigos de Situações que Ocasionam Parada da Aplicação

DG_Modulo.tDetailed.Application.*	Tamanho	Descrição
byCPUState	BYTE	Informa o estado de operação da UCP: 01: Aplicação em execução (Modo Run) 03: Aplicação parada (Modo Stop)
bForcedIOs	BIT	Existem um ou mais pontos de E/S forçados.
bNetDefinedByWeb	BIT	O endereço IP está definido a partir da Página Web de Sistema

Tabela 221: Diagnósticos Detalhados Grupo Application

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.
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 222: Diagnósticos Detalhados Grupo SNTP

DG_Modulo.tDetailed.SOE.*	Tamanho	Descrição
SOE[0].bConnectionStatus	BIT	Status de conexão do cliente 01
SOE[0].bOverflowStatus	BIT	Status da fila de eventos do cliente 01: FALSE - Não houve estouro TRUE - Limite da fila excedido
SOE[0].wEventsCounter	WORD	Contador de eventos na fila do cliente 01
SOE[1].bConnectionStatus	BIT	Status de conexão do cliente 02
SOE[1].bOverflowStatus	BIT	Status da fila de eventos do cliente 02: FALSE - Não houve estouro TRUE - Limite da fila excedido
SOE[1].wEventsCounter	WORD	Contador de eventos na fila do cliente 02.

Tabela 223: Diagnósticos Detalhados Grupo SOE

Notas:

Sincronismo dos diagnósticos do grupo SOE em sistema operando com redundância de Half-Cluster: Quando um projeto é configurado com redundância de Half-Cluster os diagnósticos do grupo SOE não são sincronizados entre os dois Half-Clusters.

Atualização dos diagnósticos do grupo SOE na transição para o estado ativo: Quando um Half-Cluster passa do estado Reserva para o estado Ativo os diagnósticos do grupo SOE passam a ser atualizados a partir do terceiro ciclo.

DG_Modulo.tDetailed.Rack.*	Tamanho	Descrição
dwAbsentRacks	DWORD	Cada bit representa um número de identificação de um bastidor, se algum bit for TRUE, isso significa que o bastidor, com o referido número de identificação, está ausente.
dwDuplicatedRacks	DWORD	Cada bit representa um número de identificação de um bastidor, se algum bit for TRUE, isso significa que mais de um bastidor está configurado com o mesmo número de identificação.
dwNonDeclaredRacks	DWORD	Cada bit representa um número de identificação de um bastidor, se algum bit for TRUE, significa que há algum bastidor configurado com um número de identificação que não está declarado no projeto.

Tabela 224: Diagnósticos Detalhados Grupo Rack

DG_Modulo.tDetailed.ApplicationInfo.*	Tamanho	Descrição
dwApplicationCRC	DWORD	CRC de 32 bits da aplicação. Quando a aplicação é modificada e enviada para a UCP, um novo CRC é calculado.

Tabela 225: Diagnósticos Detalhados Grupo ApplicationInfo

6.1.5. Diagnósticos via Blocos Funcionais

Os blocos funcionais proporcionam a visualização de alguns parâmetros que não podem ser acessados de outra maneira. A função sobre diagnósticos avançados está localizada na biblioteca *NextoStandard* e está descrita abaixo.

6.1.5.1. GetTaskInfo

Essa função retorna informações sobre uma tarefa de uma determinada aplicação.

Figura 162: 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 226: 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.
dwLimitMaxScan	DWORD	Tempo máximo de ciclo da tarefa antes de ocorrer o cão-de-guarda.
dwIECCycleCount	DWORD	Contador de ciclos IEC decorridos.

Tabela 227: Parâmetros de Saída GetTaskInfo

Possíveis ERRORCODE:

- NoError: execução com sucesso;
- TaskNotPresent: a tarefa desejada não existe.

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
sAppName : STRING;
psAppName : POINTER TO STRING;
sTaskName : STRING;
psTaskName : POINTER TO STRING;
pstTaskInfo : POINTER TO NextoStandard.stTaskInfo;
TaskInfo : NextoStandard.stTaskInfo;
Info : NextoStandard.ERRORCODE;
END_VAR
//ENTRADAS:
sAppName := 'Application';    //Variável recebe o nome da aplicação.
psAppName := ADR(sAppName); //Ponteiro com o nome da aplicação.
```

```
sTaskName := 'MainTask';    //Variável recebe o nome da tarefa.
psTaskName := ADR(sTaskName); //Ponteiro com o nome da tarefa.
pstTaskInfo := ADR(TaskInfo); //Ponteiro que irá receber as informações da
    tarefa.
//FUNÇÃO:
//Chamada da função.
Info := GetTaskInfo (psAppName, psTaskName, pstTaskInfo);
//Variável Info recebe possíveis erros da função.
```

6.2. Visor Gráfico

O visor gráfico disponível neste produto é uma importante ferramenta para o controle de processo, pois através dele podem ser reconhecidas as possíveis condições de erro, presença de componentes ou de diagnósticos ativos. Além disso, é através do visor gráfico que todos os diagnósticos, inclusive dos módulos de E/S, são exibidos ao usuário. Para maiores detalhes sobre a utilização da tecla de diagnósticos e da visualização dos mesmos, consultar a seção One Touch Diag.

Na figura abaixo, é possível visualizar todos os caracteres disponíveis no visor gráfico deste produto e, a seguir, os seus respectivos significados.

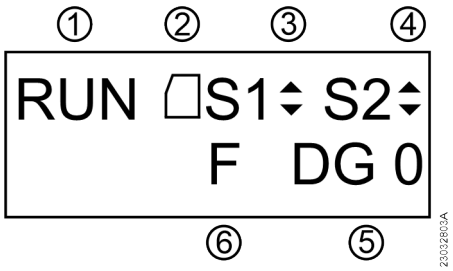


Figura 163: Tela de Status da UCP

Legenda:

- 1. Indicação do estado de operação da UCP. Caso a aplicação da UCP esteja em execução, o estado será RUN. Caso a aplicação da UCP esteja parada, o estado será STOP e quando estiver parada em marcas de depuração da aplicação o estado será BRKP. Para maiores detalhes, consultar a seção Estados de Operação da UCP.
- 2. Indicação da presença do Cartão de Memória. Para maiores detalhes sobre a instalação do cartão, consultar a seção Instalação do Cartão de Memória.
- 3. Indicação de Tráfego na COM 1. A seta para cima (▲) indica transmissão de dados e a seta para baixo (▼) indica recepção de dados. Para maiores informações sobre a interface COM 1, consultar seção Interfaces Seriais.
- 4. Indicação de Tráfego na COM 2. A seta para cima (▲) indica transmissão de dados e a seta para baixo (▼) indica recepção de dados. Para maiores informações sobre a interface COM 2, consultar seção Interfaces Seriais.
- 5. Indicação da quantidade de diagnósticos ativos na UCP. Caso o número mostrado seja diferente de 0 (zero), existem diagnósticos ativos na UCP. Para maiores detalhes sobre a visualização dos mesmos no visor gráfico da UCP, através da tecla de diagnósticos, consultar seção One Touch Diag.
- 6. Indicação de variáveis forçadas na UCP. Caso o caractere F esteja exibido no visor gráfico, alguma variável está sendo forçada pelo usuário, seja ela variável simbólica ou variável de representação direta mapeada em uma variável simbólica. Para maiores detalhes sobre forçamento de variáveis, consultar seção Escrita e Forçamento de Variáveis.

Além dos caracteres descritos acima, as UCPs Nexto poderão apresentar algumas mensagens no visor gráfico, correspondentes a algum processo que está sendo executado no momento.

A tabela abaixo mostra as mensagens e suas respectivas descrições:

Mensagem	Descrição
FORMATANDO...	Indica que a UCP está formatando o cartão de memória.
ERRO NA FORMATACAO	Indica que ocorreu um erro durante a formatação do cartão de memória pela UCP.
FORMATO INCORR.	Indica que o formato do cartão de memória está incorreto.
SENHA INCORRETA	Indica que a senha digitada não confere com a senha configurada.
TRANSFERINDO...	Indica que o projeto está sendo transferido.
ERRO NA TRANSFER.	Indica que ocorreu um erro na transferência do projeto, ocasionado por algum problema no cartão de memória ou a remoção do mesmo durante a transferência.
TRANSFER. COMPLETA	Indica que a transferência foi completada com êxito.
TIMEOUT TRANSFER.	Indica que ocorreu um time-out (expirou tempo de comunicação) durante a transferência do projeto.
MODELO UCP INVALIDO	Indica que o modelo da UCP é diferente do configurado no projeto que está no cartão de memória.
VERSAO UCP INVALIDA	Indica que a versão da UCP é diferente do configurado no projeto que está no cartão de memória.
APLICACAO CORROMPIDA	Indica que a aplicação que está no cartão de memória está corrompida.
APLICACAO INEXISTENTE	Indica que não existe aplicação dentro do cartão de memória para ser transferida à UCP.
CRC INEXISTENTE	Indica que o CRC da aplicação não existe.
MCF INEXISTENTE	Indica que não existe o arquivo MCF dentro do cartão de memória.
SEM TAG	Não existe Tag configurada para a UCP no MasterTool IEC XE.
SEM DESC	Não existe Descrição configurada para a UCP no MasterTool IEC XE.
ERRO MSG.	Indica que há erro(s) na(s) mensagem(s) de diagnóstico(s) do(s) módulo(s) requisitado(s).
SEM ASSINATURA	Indica que o produto apresentou um problema inesperado. Entre em contato com o setor de Suporte da Altus.
ERRO NA APL. REINICIANDO	Indica que ocorreu um erro na aplicação e o Runtime está reiniciando a aplicação.
APL. NAO CARREGADA	Indica que o Runtime não irá carregar a aplicação.
CARREGANDO APL.	Indica que o Runtime irá carregar a aplicação.
POSICAO INCORRETA	Indica que a UCP está em uma posição incorreta no bastidor.
ERRO FATAL	Indica que há problemas graves na inicialização da UCP como por exemplo, as partições da UCP não foram devidamente montadas. Entre em contado com o setor de suporte da Altus.
HW-SW INCOMPATIVEL	Indica que o software e hardware da UCP não são compatíveis pois o produto apresentou algum problema inesperado. Entre em contato com o setor de suporte da Altus.
ATUALIZANDO FIRMWARE	Indica que o firmware está sendo atualizado na UCP.
RECEBENDO FIRMWARE	Indica que o arquivo de atualização está sendo transferido para a UCP.
ATUALIZADO	Exibe a versão de firmware atualizada na UCP.
ERRO NA ATUALIZACAO	Indica que houve algum erro durante a atualização de firmware da UCP, ocasionado por falha na comunicação ou problemas de configuração.

Mensagem	Descrição
REINICIANDO SISTEMA...	Indica que a UCP está sendo reiniciada para que as atualizações tenham efeito.

Tabela 228: Outras Mensagens do Visor Gráfico

6.3. Log de Sistema

O *Log de Sistema* é um recurso disponível no programador MasterTool IEC XE. É uma importante ferramenta de registro de processo, pois através dele é possível localizar eventos na UCP que podem indicar condições de erro, presença de componentes ou de diagnósticos ativos. Tais eventos podem ser visualizados em ordem cronológica com uma resolução de segundos, com uma capacidade de armazenamento de até mil entradas de log armazenadas na memória interna do CP e não podem ser removidos.

Para acessar esses Logs, basta ir na *Árvore de Dispositivos* e dar um duplo clique em *Device*, e em seguida ir na aba *Log*, onde podem ser visualizadas as centenas de operações como: ciclo máximo das tarefas, acesso de usuários, alteração online, download e upload da aplicação, forçamento das variáveis, sincronização da aplicação em CPs redundantes, atualização de firmware entre outros eventos e ações.

Para que os *Logs* possam ser visualizados, basta estar conectado a uma UCP (Caminho Ativo selecionado) e clicar em . Quando este botão é pressionado os Logs serão exibidos e a atualização deles é feita automaticamente. Quando o botão não está pressionado os Logs serão mantidos na tela, mas não serão atualizados. Ou seja, este botão tem dois estados, sendo que um estado mantém os logs sendo atualizados e no outro estado a atualização está desabilitada. Para deixar de exibir os Logs, basta clicar em .

É possível filtrar os Logs em 4 tipos diferentes: Advertência(s), Erro(s), Exceção (s),Informação (s).

Outra maneira de filtrar as mensagens exibidas para o usuário é selecionar qual componente queremos visualizar.

A *Marca de Tempo* da aba de Log é apresentada pela ferramenta MasterTool a partir de informações fornecidas pelo dispositivo (UCP). A ferramenta MasterTool pode apresentar a Marca de Tempo no horário local (do computador) ou no horário UTC, se marcada a opção *Tempo UTC*.

ATENÇÃO

Se o horário ou o parâmetro de fuso horário do dispositivo estiverem incorretos, a Marca de Tempo apresentada pela ferramenta MasterTool não corresponderá ao tempo esperado.

Para maiores informações sobre os Logs de Sistema, favor consultar o Manual de Utilização do MasterTool IEC XE - MU299048 e os subcapítulos Relógio RTC e Sincronização de Tempo deste manual.

ATENÇÃO

Os logs de sistema das UCPs, a partir da versão de firmware 1.4.0.33 (Nexto) e 1.14.36.0 (Xtorm), são recarregados no caso de uma reinicialização da UCP ou por uma reinicialização do Runtime System, isto é, será possível visualizar os logs mais antigos quando uma dessas condições ocorrer.

6.4. Não Carregar a Aplicação na Inicialização

Caso seja necessário, o usuário pode optar por não carregar uma aplicação já existente na UCP durante sua inicialização. Para isto, basta energizar a UCP com a tecla de diagnóstico pressionada e mantê-la assim até ser exibida a mensagem na tela "APL. NÃO CARREGADA". Após a inicialização da UCP e a exibição no visor da mensagem avisando que a aplicação não será carregada a UCP estará também em Modo Stop. Caso seja feito um login, o software MasterTool IEC XE indicará que não existe nenhuma aplicação na UCP. Para voltar a carregar a aplicação, a UCP deve ser reinicializada, ou um novo download da aplicação deve ser feito.

6.5. Falha na Alimentação

A fonte de alimentação da Série Nexto (NX8000) possui um sistema de detecção de falha, de acordo com os níveis definidos em suas características técnicas (consultar Características Técnicas do Módulo de Alimentação 30W 24Vdc – CT114200).

Existem duas formas de diagnosticar a falha:

1. Caso a fonte NX8000 seja energizada com tensão inferior ao limite mínimo exigido, será gerado um diagnóstico de falha de alimentação, o qual será reconhecido pela UCP e a mensagem "FALHA NA ALIMENTAÇÃO" será exibida em seu visor. Quando a alimentação estiver dentro dos limites estabelecidos, a UCP irá reconhecer e, automaticamente, será reiniciada com a aplicação do usuário. O diagnóstico ainda estará ativo para mostrar ao usuário que na última inicialização ocorreu falha na alimentação.
2. Caso a fonte NX8000 tenha uma queda de tensão para um valor inferior ao limite mínimo exigido e retornar para um valor acima em até 10 ms, a falha de alimentação não será reconhecida pela UCP e não será gerado o diagnóstico, pois o sistema se mantém intacto durante esse tempo. Porém, se a queda de tensão for superior a 10 ms, a mensagem "FALHA NA ALIMENTAÇÃO" será exibida no visor da UCP e o diagnóstico será ativado.

Figura 164: Mensagem de Falha na Alimentação

O usuário pode alterar o valor da variável atribuída à falha na alimentação para FALSE durante a execução da aplicação, facilitando a verificação e tratamento deste diagnóstico.

O diagnóstico de FALHA NA ALIMENTACAO já está previamente mapeado em uma região específica de memória, definida como Diagnósticos Detalhados da UCP. Dessa forma, basta utilizá-lo como uma variável global. O nome da variável está descrita na lista de diagnósticos detalhados da UCP na seção Diagnósticos via Variáveis.

6.6. Problemas mais Comuns

Se, ao energizar a UCP, a mesma não entra em funcionamento, os seguintes itens devem ser verificados:

- A temperatura ambiente está dentro da faixa suportada pelos equipamentos?
- A fonte de alimentação do bastidor está sendo alimentada com a tensão correta?
- A fonte de alimentação é o módulo mais à esquerda no bastidor (bastidor sendo visto de frente), e é seguido pela UCP da Série Nexto à sua direita?
- Os equipamentos da rede, como hubs, switches ou roteadores, estão alimentados, interligados, configurados e funcionando corretamente?
- O cabo de rede Ethernet está devidamente conectado à porta NET da UCP Nexto e ao equipamento de rede?
- A UCP da Série Nexto está ligada, em modo de execução (Run) e sem diagnósticos relacionados à hardware?

Se a UCP Nexto indica o estado execução (Run), mas não responde às comunicações solicitadas, sejam elas pelo MasterTool IEC XE ou através de protocolos, os seguintes itens devem ser verificados:

- A configuração dos parâmetros Ethernet da UCP estão corretos?
- O respectivo protocolo de comunicação está configurado corretamente na UCP?
- As variáveis que habilitam as relações MODBUS estão devidamente habilitadas?

Se nenhum problema for identificado, consulte o Suporte à Clientes Altus.

6.7. Solução de Problemas

A tabela abaixo mostra os sintomas de alguns problemas com suas possíveis causas e soluções. Se o problema persistir, entre em contato com o Suporte Técnico da Altus.

Sintoma	Possível Causa	Solução
Não liga	Falta de alimentação ou alimentado incorretamente.	Verificar se a UCP está conectada corretamente no bastidor. Desenergizar e retirar todos os módulos do barramento, menos a fonte de alimentação e a UCP. Energizar o barramento e verificar o funcionamento da fonte de alimentação, tanto a externa como a fonte conectada ao barramento. Verificar se a tensão de alimentação chega ao borne da fonte de alimentação Nexto, e também se a polarização está correta.
Visor Gráfico da UCP apresenta a mensagem WRONG SLOT	UCP na Posição Incorreta.	A UCP deve ocupar os slots 2 e 3 do bastidor 0. Colocá-la na posição correta.
Não comunica	Mau contato ou mal configurado.	Verificar todas as conexões dos cabos de comunicação. Verificar as configurações das interfaces seriais e Ethernet no software MasterTool IEC XE.
Não reconhece o cartão de memória	Mal conectado ou desmontado.	Verificar se o cartão de memória está conectado corretamente no compartimento. Verificar se o cartão de memória foi colocado do lado correto, conforme a indicação no painel frontal da UCP. Verificar se o cartão de memória não foi desmontado através da tecla MS, localizada no painel frontal, visualizando a indicação no visor gráfico da UCP.

Tabela 229: Solução de Problemas

6.8. Manutenção Preventiva

- Deve-se verificar, a cada ano, se os cabos de interligação estão com as conexões firmes, sem depósitos de poeira, principalmente os dispositivos de proteção.
- Em ambientes sujeitos a contaminação excessiva, deve-se limpar periodicamente o equipamento, retirando resíduos, poeira, etc.
- Os diodos TVS utilizados para a proteção contra transientes causados por descargas atmosféricas devem ser inspecionados periodicamente, pois podem estar danificados ou destruídos caso a energia absorvida esteja acima do limite. Em muitos casos, a falha pode não ser evidente ou facilmente visível. Em aplicações críticas, é recomendável a substituição periódica dos diodos TVS, mesmo os que não apresentarem sinais visíveis de falha.
- Aperto e limpeza do barramento a cada 6 meses.
- Para maiores informações, consultar o Manual de Utilização Série Nexto – MU214000.

7. Anexo - Interoperabilidade DNP3

7.1. DNP3 Device Profile

DNP3 **DEVICE PROFILE DOCUMENT**	
Device Identification	
Vendor Name	Altus S/A
Device Name	NX3030
Device Function	Slave
DNP Levels Supported for	Requests: None Responses: None
Connections Supported	IP Networking
Methods to set Configurable Parameters	Software: Master Tool IEC XE
IP Networking	
Type of End Point:	TCP Listening (Outstation Only)
Accepts TCP Connections from	Allows all
IP Address(es) from which TCP Connections are accepted:	*.*.*.*
TCP Listen Port Number	Configurable, range 1 to 65535
TCP Keep-alive timer	Configurable, range 0 to 4294967295
Multiple master connections	Supports up to two masters Based on TCP port number
Time synchronization support	SNTP
Link Layer	
Data Link Address	Configurable, range 0 to 65519
Self Address Support using address 0xFFFC	No
Requires Data Link Layer Confirmation	Never
Maximum number of octets Transmitted in a Data Link Frame	Fixed at 292
Maximum number of octets that can be Received in a Data Link Frame	Fixed at 292
Application Layer	
Maximum number of octets Transmitted in an Application Layer Fragment	Fixed at 2048
Maximum number of octets that can be received in an Application Layer Fragment	Fixed at 2048
Timeout waiting for Application Confirm of solicited response message	Fixed at 10000 ms
Device Trouble Bit IIN1.6	This bit will be set if PLC is not in RUN mode
Event Buffer Overflow Behavior	Discard the oldest event
Sends Multi-Fragment Responses	Yes
Outstation Unsolicited Response Support	
Supports Unsolicited Reporting	No

Tabela 230: DNP3 Device Profile

7.2. DNP V3.0 Implementation Table

DNP OBJECT GROUP & VARIATION			REQUEST Master may issue Outstation must parse		RESPONSE Master must parse Outstation may issue	
Group Num	**Var Num**	**Description**	**Function Codes (dec)**	**Qualifier Codes (hex)**	**Function Codes (dec)**	**Qualifier Codes (hex)**
1	0	Binary Input – Any Variation	1 (read)	00, 01 (start-stop) 06 (no range, or all)		
1	1	Binary Input – Packed format	1 (read)	00, 01 (start-stop) 06 (no range, or all)	129 (response)	00, 01 (start-stop)
2	0	Binary Input Event – Any Variation	1 (read)	06 (no range, or all) 07, 08 (limited qty)		
2	2	Binary Input Event – With absolute time	1 (read)	06 (no range, or all) 07, 08 (limited qty)	129 (response)	17, 28 (index)
60	1	Class Objects – Class 0 data	1 (read)	06 (no range, or all)		
60	2	Class Objects – Class 1 data	1 (read)	06 (no range, or all) 07, 08 (limited qty)		
80	1	Internal Indications - Packed format	1 (read)	00, 01 (start-stop)	129 (response)	00 (start-stop)
			2 (write)	00 (start-stop) index=7		

Tabela 231: DNP V3.0 Implementation Table