Manual de Utilização
UCP Série Nexto
NX3035

MU214107 Rev. B

23 de abril de 2026

www.altus.com.br

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

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

Os equipamentos de automação industrial fabricados pela Altus são robustos e confiáveis devido ao rígido controle de qualidade a que são submetidos. No entanto, equipamentos eletrônicos de controle industrial (controladores programáveis, comandos numéricos, etc.) podem causar danos às máquinas ou processos por eles controlados em caso de defeito em seus componentes e/ou de erros de programação ou instalação, podendo inclusive colocar em risco vidas humanas.

O usuário deve analisar as possíveis consequências destes defeitos e providenciar instalações adicionais externas de segurança que, em caso de necessidade, sirvam para preservar a segurança do sistema, principalmente nos casos da instalação inicial e de testes.

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

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

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

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

A Altus garante que seus equipamentos funcionam de acordo com as descrições contidas explicitamente em seus manuais e/ou características técnicas, não garantindo a satisfação de algum tipo particular de aplicação dos equipamentos.

A Altus desconsiderará qualquer outra garantia, direta ou implícita, principalmente quando se tratar de fornecimento de terceiros.

Os pedidos de informações adicionais sobre o fornecimento e/ou características dos equipamentos e serviços Altus devem ser feitos por escrito. A Altus não se responsabiliza por informações fornecidas sobre seus equipamentos sem registro formal.

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

DIREITOS AUTORAIS

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

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

NOTIFICAÇÃO DE USO DE SOFTWARE ABERTO

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

Sumário

1. Introdução . . . 1
   - 1.1. 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. Interface Serial . . . 13
       - 2.2.5.1. COM 1 . . . 13
     - 2.2.6. Interfaces Ethernet . . . 13
       - 2.2.6.1. NET 1 ... NET 6 . . . 13
     - 2.2.7. Link de Redundância . . . 14
       - 2.2.7.1. NET A / NET B . . . 14
     - 2.2.8. Interface do Cartão de Memória . . . 14
     - 2.2.9. Características Ambientais . . . 14
   - 2.3. Compatibilidade com Outros Produtos . . . 15
   - 2.4. Desempenho . . . 15
     - 2.4.1. Tempo de Intervalo da MainTask . . . 15
     - 2.4.2. Tempos de Aplicação . . . 15
     - 2.4.3. Tempos para Execução de Instruções . . . 16
     - 2.4.4. Tempos de Inicialização . . . 16
   - 2.5. Dimensões Físicas . . . 17
   - 2.6. Dados para Compra . . . 17
     - 2.6.1. Itens Integrantes . . . 17
     - 2.6.2. Código do Produto . . . 18
   - 2.7. Produtos Relacionados . . . 18
3. Instalação . . . 19
   - 3.1. Instalação Mecânica . . . 19
   - 3.2. Instalação Elétrica . . . 19
   - 3.3. Conexão com a Rede Ethernet . . . 20
     - 3.3.1. Endereço IP . . . 20

3.3.2. ARP Gratuito . . . 22
3.3.3. Instalação do Cabo de Rede . . . 22
3.4. Conexão com a Rede Serial RS-485 . . . 24
3.4.1. Comunicação RS-485 sem Terminação . . . 24
3.4.2. Comunicação RS-485 com Terminação Interna . . . 25
3.4.3. Comunicação RS-485 com Terminação Externa . . . 26
3.4.4. Exemplo de Ligação de Rede RS-485 com Terminação Externa e Redundância de Mestre . . . 27
3.5. Instalação do Cartão de Memória . . . 27
3.6. Instalação da Arquitetura . . . 29
3.6.1. Instalação de Módulos no Bastidor Principal . . . 29
3.7. Instalação do Programador . . . 29
4. Programação Inicial . . . 30
4.1. Organização e Acesso à Memória . . . 30
4.2. Perfis de Projeto . . . 32
4.2.1. Simples . . . 32
4.2.2. Básico . . . 33
4.2.3. Normal . . . 33
4.2.4. Experiente . . . 33
4.2.5. Personalizado . . . 34
4.2.6. Perfil de Máquina . . . 34
4.2.7. Tabela Geral . . . 35
4.2.8. Número Máximo de Tarefas . . . 36
4.3. Configurando a UCP . . . 36
4.4. Bibliotecas . . . 37
4.5. Inserindo uma Instância de Protocolo . . . 37
4.5.1. MODBUS RTU . . . 38
4.5.2. MODBUS Ethernet . . . 39
4.6. Localizando o Dispositivo . . . 41
4.7. Login . . . 43
4.8. Modo Run . . . 45
4.9. Modo Stop . . . 46
4.10. Escrita e Forçamento de Variáveis . . . 47
4.11. Logout . . . 47
4.12. Upload do Projeto . . . 48
4.13. Estados de Operação da UCP . . . 49
4.13.1. Run . . . 49
4.13.2. Stop . . . 49
4.13.3. Breakpoint . . . 49
4.13.4. Exception . . . 49
4.13.5. Reset a Quente . . . 49
4.13.6. Reset a Frio . . . 49
4.13.7. Reset Origem . . . 49
4.14. Programas (POUs) e Listas de Variáveis Globais (GVLs) . . . 50
4.14.1. Programa MainPrg . . . 50
4.14.2. Programa StartPrg . . . 50
4.14.3. Programa UserPrg . . . 50
4.14.4. GVL System_Diagnostics . . . 51
4.14.5. GVL Disables . . . 51

4.14.6. GVL IOQualities . . . 52
4.14.7. GVL Module_Diagnostics . . . 53
4.14.8. GVL ReqDiagnostics . . . 54
5. Configuração . . . 56
5.1. Device . . . 56
5.1.1. Gerênciamento de Usuários e Direitos de Acesso . . . 56
5.1.2. Configurações do CP . . . 56
5.2. Configuração da UCP . . . 58
5.2.1. Parâmetros Gerais . . . 58
5.2.1.1. Troca a Quente . . . 59
5.2.1.1.1. Troca a Quente Desabilitada, Apenas para Módulos Declarados . . . 60
5.2.1.1.2. Troca a Quente Desabilitada . . . 60
5.2.1.1.3. Troca a Quente Desabilitada, sem Consistência na Partida . . . 60
5.2.1.1.4. Troca a Quente Habilitada, com Consistência na Partida Apenas para Módulos Declarados . . . 60
5.2.1.1.5. Troca a Quente Habilitada, com Consistência na Partida . . . 60
5.2.1.1.6. Troca a Quente Habilitada, sem Consistência na Partida . . . 61
5.2.1.1.7. Como realizar a Troca a Quente . . . 61
5.2.1.2. Áreas de Memória Retentiva e Persistente . . . 63
5.2.2. Sincronização de Tempo . . . 64
5.2.2.1. SNTP . . . 65
5.2.2.2. Horário de Verão . . . 65
5.3. Configuração da Interface Serial . . . 66
5.3.1. COM 1 . . . 66
5.3.1.1. Configurações Avançadas . . . 67
5.4. Configuração das Interfaces Ethernet Integradas . . . 67
5.4.1. Configurações Básicas das Portas Ethernet . . . 68
5.4.2. Configuração de Modo das Portas Ethernet . . . 69
5.4.2.1. Modo Simples . . . 69
5.4.2.2. Modo NIC Teaming na UCP NX3035 . . . 69
5.4.2.3. Modo Switch . . . 71
5.4.2.3.1. Modo Switch - Disabled . . . 71
5.4.2.3.2. Modo Switch - MRP . . . 71
5.4.2.3.3. Modo Switch - RSTP . . . 72
5.4.3. Configuração de Modo de Falha das Portas Ethernet . . . 73
5.4.4. Portas TCP/UDP Reservadas . . . 73
5.4.5. Roteamento de Rede . . . 74
5.5. Configuração de Interfaces Ethernet com Módulos NX5000 . . . 74
5.5.1. Configurações Básicas da Porta Ethernet do NX5000 . . . 74
5.5.1.1. Endereçamento IP de Portas Ethernet Remotas do NX5000 (NET1) . . . 74
5.5.1.1.1. NET 1 . . . 74
5.5.2. Configuração de Modo da Porta Ethernet do NX5000 . . . 75
5.5.2.1. Modo NIC Teaming no NX5000 . . . 75
5.5.3. Configuração de Modo de Falha das Portas Ethernet . . . 76
5.5.4. Outras Configurações do Módulo NX5000 . . . 76
5.5.4.1. Aba “Parâmetros do Módulo” . . . 76
5.5.4.2. Aba “Bus I/O Mapping” . . . 77
5.6. Configuração de Redes PROFIBUS com Módulos NX5001 . . . 77

5.6.1. Inserção ou Remoção de Módulos NX5001 . . . 78
5.6.2. Ajuste de Parâmetros dos Módulos NX5001 . . . 78
5.6.2.1. Aba “Geral” . . . 79
5.6.2.2. Aba “Parâmetros do Módulo” . . . 80
5.6.2.3. Aba “Bus I/O Mapping” . . . 81
5.6.3. Configurações de Remotas PROFIBUS . . . 81
5.6.3.1. Remotas Compatíveis com Rede PROFIBUS Redundante . . . 81
5.6.3.2. Configuração de Cão-de-Guarda em Remotas PROFIBUS em CPs Redundantes . . 82
5.7. Configuração de Protocolos . . . 82
5.7.1. Comportamento dos Protocolos x Estados da UCP . . . 84
5.7.2. MODBUS RTU Mestre . . . 85
5.7.2.1. Configuração do Protocolo MODBUS Mestre por Mapeamento Simbólico . . . 85
5.7.2.1.1. Parâmetros Gerais do Protocolo MODBUS Mestre – Configuração por Mapeamento Simbólico . . . 86
5.7.2.1.2. Configuração dos Dispositivos – Configuração por Mapeamento Simbólico 87
5.7.2.1.3. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico 88
5.7.2.1.4. Configuração das Requisições – Configuração por Mapeamento Simbólico . 89
5.7.2.2. Configuração do Protocolo MODBUS Mestre por Representação Direta (%Q) . . . 92
5.7.2.2.1. Parâmetros Gerais do Protocolo MODBUS Mestre – Configuração por Representação Direta (%Q) . . . 92
5.7.2.2.2. Configuração dos Dispositivos – Configuração por Representação Direta (%Q) . . . 93
5.7.2.2.3. Configuração dos Mapeamentos – Configuração por Representação Direta (%Q) . . . 94
5.7.3. MODBUS RTU Escravo . . . 96
5.7.3.1. Configuração do Protocolo MODBUS Escravo por Mapeamento Simbólico . . . 97
5.7.3.1.1. Parâmetros Gerais do Protocolo MODBUS Escravo – Configuração por Mapeamento Simbólico . . . 97
5.7.3.1.2. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico 100
5.7.3.2. Configuração do Protocolo MODBUS Escravo por Representação Direta (%Q) . . 101
5.7.3.2.1. Parâmetros Gerais do Protocolo MODBUS Escravo – Configuração por Representação Direta (%Q) . . . 101
5.7.3.2.2. Configuração dos Mapeamentos – Configuração por Representação Direta (%Q) . . . 102
5.7.4. MODBUS Ethernet . . . 104
5.7.5. MODBUS Ethernet Cliente . . . 105
5.7.5.1. Configuração do Protocolo MODBUS Ethernet Cliente por Mapeamento Simbólico 106
5.7.5.1.1. Parâmetros Gerais do Protocolo MODBUS Cliente – Configuração por Mapeamento Simbólico . . . 106
5.7.5.1.2. Configuração dos Dispositivos – Configuração por Mapeamento Simbólico 107
5.7.5.1.3. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico 108
5.7.5.1.4. Configuração das Requisições – Configuração por Mapeamento Simbólico . 109
5.7.5.2. Configuração do Protocolo MODBUS Ethernet Cliente por Representação Direta (%Q) . . . 112
5.7.5.2.1. Parâmetros Gerais do Protocolo MODBUS Cliente – Configuração por Representação Direta (%Q) . . . 113
5.7.5.2.2. Configuração dos Dispositivos – configuração por Representação Direta (%Q) . . . 114
5.7.5.2.3. Configuração dos Mapeamentos – Configuração por Representação Direta (%Q) . . . 114
5.7.5.3. Disparo de Relações MODBUS Cliente de Forma Acíclica . . . 117

5.7.6. MODBUS Ethernet Servidor . . . 117
5.7.6.1. Configuração do Protocolo MODBUS Ethernet Servidor por Mapeamento Simbólico 117
5.7.6.1.1. Parâmetros Gerais do Protocolo MODBUS Servidor – Configuração por Mapeamento Simbólico . . . 117
5.7.6.1.2. Diagnósticos MODBUS Servidor – Configuração por Mapeamento Simbólico . . . 119
5.7.6.1.3. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico 119
5.7.6.2. Configuração do Protocolo MODBUS Ethernet Servidor por Representação Direta (%Q) . . . 120
5.7.6.2.1. Parâmetros Gerais do Protocolo MODBUS Servidor – Configuração por Representação Direta (%Q) . . . 121
5.7.6.2.2. Configuração dos Mapeamentos – Configuração por Representação Direta (%Q) . . . 122
5.7.7. OPC DA Servidor . . . 124
5.7.7.1. Criando um Projeto para Comunicação OPC DA . . . 126
5.7.7.2. Configurando um CP no Servidor OPC DA . . . 129
5.7.7.2.1. Importando uma Configuração do Projeto . . . 131
5.7.7.3. Configuração com CP no Servidor OPC DA com Redundância de Conexão . . . 131
5.7.7.4. Variáveis de Status e Qualidade da Comunicação OPC DA . . . 132
5.7.7.5. Limites da Comunicação com Servidor OPC DA . . . 134
5.7.7.6. Acessando Dados Através de um Cliente OPC DA . . . 134
5.7.8. OPC UA Servidor . . . 136
5.7.8.1. Criando um Projeto para Comunicação OPC UA . . . 137
5.7.8.2. Tipos de Variáveis Suportadas . . . 139
5.7.8.3. Limite de Clientes Conectados no Servidor OPC UA . . . 139
5.7.8.4. Limite de Variáveis de Comunicação no Servidor OPC UA . . . 139
5.7.8.5. Configurações de Criptografia . . . 139
5.7.8.6. Principais Parâmetros de Comunicação Ajustados em um Cliente OPC UA . . . 140
5.7.8.6.1. Endpoint URL . . . 140
5.7.8.6.2. Publishing Interval (ms) e Sampling Interval (ms) . . . 140
5.7.8.6.3. Lifetime Count e Keep-Alive Count . . . 141
5.7.8.6.4. Queue Size e Discard Oldest . . . 141
5.7.8.6.5. Filter Type e Deadband Type . . . 141
5.7.8.6.6. PublishingEnabled, MaxNotificationsPerPublish e Priority . . . 142
5.7.8.7. Acessando Dados Através de um Cliente OPC UA . . . 142
5.7.9. Mestre EtherCAT . . . 143
5.7.9.1. Instalando e Inserindo Dispositivos EtherCAT . . . 143
5.7.9.1.1. EtherCAT - Scan Devices . . . 144
5.7.9.2. Configuração do Mestre EtherCAT . . . 145
5.7.9.2.1. EtherCAT Master - General . . . 145
5.7.9.2.2. EtherCAT Master - Sync Unit Assignment . . . 147
5.7.9.2.3. EtherCAT Master - Overview . . . 147
5.7.9.2.4. EtherCAT Master - I/O Mapping . . . 147
5.7.9.2.5. EtherCAT Master - IEC Objects . . . 147
5.7.9.2.6. EtherCAT Master - Status / Informations . . . 147
5.7.9.3. Configuração do Escravo EtherCAT . . . 147
5.7.9.3.1. EtherCAT Slave - General . . . 147
5.7.9.3.2. EtherCAT Slave - Process Data . . . 151
5.7.9.3.3. EtherCAT Slave - Edit PDO List . . . 153

5.7.9.3.4. EtherCAT Slave - Startup Parameters . . . 153
5.7.9.3.5. EtherCAT Slave - Module I/O Mapping . . . 153
5.7.9.3.6. EtherCAT Slave - Status and Information . . . 154
5.7.10. EtherNet/IP . . . 154
5.7.10.1. Adaptador Ethernet . . . 155
5.7.10.2. Configuração do Scanner EtherNet/IP . . . 157
5.7.10.2.1. Geral . . . 157
5.7.10.2.2. Conexões . . . 158
5.7.10.2.3. Assemblies . . . 160
5.7.10.2.4. Parâmetros Definidos pelo Usuário . . . 161
5.7.10.2.5. EtherNet/IP: Mapeamento de E/S . . . 163
5.7.10.3. Configuração do Adapter EtherNet/IP . . . 163
5.7.10.3.1. Geral . . . 163
5.7.10.3.2. EtherNet/IP Adapter: Mapeamento de E/S . . . 164
5.7.10.4. Configuração do Módulo EtherNet/IP . . . 164
5.7.10.4.1. Assemblies . . . 164
5.7.10.4.2. EtherNet/IP Module: Mapeamento de E/S . . . 164
5.7.11. PROFINET Controller . . . 165
5.8. Desempenho de Comunicação . . . 165
5.8.1. Servidor MODBUS . . . 165
5.8.1.1. Interfaces Integradas da UCP . . . 165
5.8.1.2. Interfaces Remotas . . . 166
5.8.1.3. Taxa de Comunicação . . . 166
5.8.2. Servidor OPC UA . . . 167
5.9. Desempenho do Sistema . . . 167
5.9.1. Escaneamento de E/S . . . 168
5.9.2. Cartão de Memória . . . 168
5.10. Relógio RTC . . . 168
5.10.1. Blocos Funcionais e Funções para Leitura e Escrita do RTC . . . 169
5.10.1.1. Funções de Leitura do RTC . . . 170
5.10.1.1.1. GetDateAndTime . . . 170
5.10.1.1.2. GetTimeZone . . . 170
5.10.1.1.3. GetDayOfWeek . . . 171
5.10.1.2. Funções de Escrita do RTC . . . 172
5.10.1.2.1. SetDateAndTime . . . 172
5.10.1.2.2. SetTimeZone . . . 174
5.10.2. Estruturas de Dados do RTC . . . 175
5.10.2.1. EXTENDED_DATE_AND_TIME . . . 175
5.10.2.2. DAYS_OF_WEEK . . . 175
5.10.2.3. RTC_STATUS . . . 175
5.10.2.4. TIMEZONESETTINGS . . . 176
5.10.3. Limites de Operação do RTC . . . 176
5.11. Memória de Arquivos de Usuário . . . 177
5.12. Cartão de Memória . . . 178
5.12.1. Configuração do Cartão de Memória . . . 179
5.12.1.1. Formato Não Suportado . . . 180
5.12.1.2. Formatação do Cartão de Memória . . . 181
5.12.1.3. Desmontagem do Cartão de Memória . . . 182

5.12.1.4. Gerenciamento da Interface do Cartão de Memória . . . 184
5.12.1.5. Gerenciamento da Interface do Cartão de Memória via Aplicação . . . 185
5.13. Menu Informativo e de Configuração da UCP . . . 186
5.14. Blocos Funcionais e Funções . . . 189
5.14.1. Blocos Funcionais Especiais para Comunicação Serial . . . 189
5.14.1.1. SERIAL_CFG . . . 192
5.14.1.2. SERIAL_GET_CFG . . . 194
5.14.1.3. SERIAL_GET_CTRL . . . 195
5.14.1.4. SERIAL_GET_RX_QUEUE_STATUS . . . 197
5.14.1.5. SERIAL_PURGE_RX_QUEUE . . . 198
5.14.1.6. SERIAL_RX . . . 199
5.14.1.7. SERIAL_RX_EXTENDED . . . 201
5.14.1.8. SERIAL_TX . . . 204
5.14.2. Atualização de Entradas e Saídas . . . 206
5.14.2.1. REFRESH_INPUT . . . 206
5.14.2.2. REFRESH_OUTPUT . . . 207
5.14.3. Temporizador Retentivo . . . 208
5.14.3.1. TOF_RET . . . 208
5.14.3.2. TON_RET . . . 210
5.14.3.3. TP_RET . . . 211
5.14.4. Temporizador Não-Redundante . . . 213
5.14.4.1. TOF_NR . . . 213
5.14.4.2. TON_NR . . . 213
5.14.4.3. TP_NR . . . 214
5.14.5. Log de Usuário . . . 215
5.14.5.1. UserLogAdd . . . 216
5.14.5.2. UserLogDeleteAll . . . 217
5.15. Acesso a Aba de Gerenciamento . . . 218
5.15.1. Seção de Sistema . . . 218
5.15.1.1. Ajuste do Relógio . . . 219
5.15.1.1.1. Horário do Computador (UTC) . . . 219
5.15.1.1.2. Horário Personalizado (UTC) . . . 219
5.15.2. Seção de Rede . . . 219
5.15.2.1. Configurações da Seção de Rede . . . 220
5.15.2.1.1. Modo Definido pela aplicação . . . 220
5.15.2.1.2. Modo Definido pela Página Web . . . 221
5.15.2.2. Sniffer de Rede . . . 222
5.16. SNMP . . . 223
5.16.1. SNMP nas UCPs Nexto . . . 224
5.16.2. Configuração SNMP . . . 225
5.16.3. Usuário e Comunidades SNMP . . . 226
5.17. Firewall . . . 227
5.17.1. Configuração . . . 227
5.17.2. Configurações Gerais . . . 227
5.17.3. Regras de Usuário . . . 229
5.17.3.1. Modo Switch ou Nic Teaming . . . 231
5.18. OpenVPN . . . 231
5.18.1. Importação de Configurações . . . 232

5.18.2. Configuração do OpenVPN . . . 233
5.18.2.1. Configurações Comuns . . . 234
5.18.2.1.1. Modo . . . 234
5.18.2.1.2. Protocolo . . . 234
5.18.2.1.3. Nível de Logs . . . 234
5.18.2.1.4. Keep Alive Ping . . . 234
5.18.2.1.5. Keep Alive Timeout . . . 234
5.18.2.1.6. Arquivos de Segurança . . . 234
5.18.2.1.7. Chave do TA . . . 235
5.18.2.2. Configurações Exclusivas do Servidor . . . 235
5.18.2.2.1. Endereço de Rede . . . 235
5.18.2.2.2. Comunicação entre Clientes . . . 235
5.18.2.2.3. Máximos Clientes Conectados . . . 235
5.18.2.2.4. Redes Privadas . . . 235
5.18.2.3. Configurações Exclusivas de Cliente . . . 237
5.18.2.3.1. IP Remoto . . . 237
5.18.2.4. Aplicação de Configurações . . . 237
5.18.3. Arquivos de Segurança . . . 237
5.18.4. Tabela de Status . . . 238
5.18.5. Arquivos para Baixar . . . 240
5.18.6. Configuração de Arquiteturas . . . 240
5.18.6.1. Host-to-Host . . . 240
5.18.6.2. Host-to-Site . . . 241
5.18.6.3. Site-to-Site . . . 242
5.19. Servidor FTP . . . 242
5.19.1. Configuração . . . 243
5.19.1.1. Configurações Gerais . . . 243
5.19.1.1.1. Habilitar Servidor . . . 243
5.19.1.1.2. Habilitar Comunicação Criptografada (FTPS) . . . 243
5.19.1.1.3. Acesso Somente Leitura . . . 244
5.19.1.1.4. Tempo Limite Ocioso (Segundos) . . . 244
5.19.1.2. Configurações de Usuário . . . 244
5.19.1.2.1. Nome de Usuário . . . 244
5.19.1.2.2. Senha . . . 244
5.19.1.3. Status . . . 244
5.19.1.3.1. Estado Atual . . . 244
5.19.1.3.2. Conexões Ativas . . . 244
5.20. SysLog . . . 245
5.20.1. Configuração do SysLog . . . 245
6. Redundância com UCP NX3035 . . . 246
6.1. Introdução . . . 246
6.2. Descrição Técnica e Configuração . . . 247
6.2.1. Configuração Mínima de um CP Redundante . . . 247
6.2.2. Canais de Sincronismo entre CPA e CPB . . . 248
6.2.3. Configurações Típicas de um CP Redundante . . . 249
6.2.3.1. Adição de Módulos NX5001 para Redes PROFIBUS . . . 249
6.2.3.2. Adição de Módulos NX5000 para Redes Ethernet . . . 250
6.2.3.3. Adição de Módulos de E/S . . . 250

6.2.4. Características Gerais de um CP Redundante . . . 250
6.2.5. Dados para Compra da Redundância . . . 253
6.3. Princípios de Funcionamento . . . 254
6.3.1. Identificação de uma UCP NX3035 . . . 254
6.3.2. Projeto Redundante Único . . . 254
6.3.3. Funcionalidades Configuradas na Página Web . . . 254
6.3.4. Estrutura do Projeto Redundante . . . 254
6.3.4.1. Redundancy Template . . . 254
6.3.4.2. Tarefa Única e Cíclica MainTask . . . 254
6.3.4.3. Programa MainPrg . . . 255
6.3.4.4. Programa ActivePrg . . . 255
6.3.4.5. Programa NonSkippedPrg . . . 255
6.3.4.6. Variáveis Redundantes e Não-Redundantes . . . 256
6.3.4.7. Variáveis %I Redundantes e Não-Redundantes . . . 257
6.3.4.8. Variáveis %Q Redundantes e Não-Redundantes . . . 257
6.3.4.9. Variáveis %M Redundantes e Não-Redundantes . . . 260
6.3.4.10. Variáveis Simbólicas Redundantes e Não-Redundantes . . . 261
6.3.4.11. Variáveis Simbólicas Endereçadas com a Diretiva AT . . . 262
6.3.5. Estruturas de Dados de Diagnósticos, Comandos e Informações do Usuário . . . 262
6.3.6. Serviços de Sincronização Cíclicos através de NETA e NETB . . . 262
6.3.6.1. Troca de Diagnósticos e Comandos . . . 263
6.3.6.2. Sincronização de Dados Redundantes . . . 263
6.3.6.3. Sincronização da Lista de Forçamentos Redundantes . . . 263
6.3.7. Serviços de Sincronização Esporádicos através de NETA e NETB . . . 263
6.3.7.1. Sincronização de Projetos . . . 264
6.3.8. Desabilitação da Sincronização de Projetos . . . 264
6.3.9. Configurações de Redes PROFIBUS . . . 265
6.3.10. Redes Ethernet Redundantes com NIC Teaming . . . 266
6.3.11. Configuração de Endereços IP nas portas integradas NET1 ... NET6 de uma UCP NX3035 em um CP Redundante . . . 266
6.3.12. Configuração de Endereços IP em Módulos NX5000 em um CP Redundante . . . 266
6.3.12.1. IP Fixo . . . 267
6.3.12.2. Troca Automática de IP . . . 267
6.3.12.3. IP Ativo . . . 268
6.3.12.4. Múltiplos IPs . . . 268
6.3.13. Uso Combinado de NIC Teaming e IP Ativo . . . 269
6.3.14. Uso de Comunicação OPC DA com Projetos Redundantes . . . 269
6.3.15. Estados de um CP Redundante . . . 270
6.3.15.1. Ativo . . . 270
6.3.15.2. Reserva . . . 270
6.3.15.3. Inativo . . . 270
6.3.15.4. Não-Configurado . . . 270
6.3.15.5. Inicializando . . . 270
6.3.16. Máquina de Estados de Redundância . . . 270
6.3.16.1. Estado Não Configurado . . . 271
6.3.16.1.1. Transição 1 - Não-Configurado para Inicializando . . . 272
6.3.16.2. Estado Inicializando . . . 272
6.3.16.2.1. Transição 2 - Inicializando para Não-Configurado . . . 272

6.3.16.2.2. Transição 3 - Inicializando para Inativo . . . 272
6.3.16.2.3. Transição 4 - Inicializando para Ativo . . . 272
6.3.16.2.4. Transição 5 - Inicializando para Reserva . . . 273
6.3.16.3. Estado Inativo . . . 273
6.3.16.3.1. Transição 6 - Inativo para Não-Configurado . . . 273
6.3.16.4. Estado Ativo . . . 273
6.3.16.4.1. Transição 7 - Ativo para Não-Configurado . . . 273
6.3.16.4.2. Transição 8 - Ativo para Inativo . . . 273
6.3.16.4.3. Transição 9 - Ativo para Reserva . . . 274
6.3.16.5. Estado Reserva . . . 274
6.3.16.5.1. Transição 10 – Reserva para Não-Configurado . . . 274
6.3.16.5.2. Transição 11 – Reserva para Inativo . . . 274
6.3.16.5.3. Transição 12 – Reserva para Ativo . . . 274
6.3.17. Primeiros Instantes em Estado Ativo . . . 275
6.3.18. Falhas mais Comuns Causadoras de Switch-overs Automáticos entre Half-Clusters . . . 275
6.3.19. Switch-overs entre Half-Clusters Gerenciados pelo Usuário . . . 275
6.3.20. Tolerância a Falhas . . . 276
6.3.20.1. Falhas Simples com Indisponibilidade . . . 276
6.3.20.2. Falhas Simples sem Indisponibilidade Causando um Switch-over . . . 277
6.3.20.3. Falhas Duplas sem Indisponibilidade Causando um Switch-over . . . 277
6.3.21. Overhead da Redundância . . . 277
6.4. Programação de um CP Redundante . . . 277
6.4.1. Assistente para Criação de um Novo Projeto Redundante . . . 277
6.4.2. Configuração dos Half-Clusters . . . 281
6.4.2.1. Configuração Fixa nas Posições 0 a 5 do Bastidor . . . 281
6.4.3. Configurações das Portas Ethernet Integradas de uma UCP NX3035 Redundante (NET 1 a NET 6) . . . 282
6.4.3.1. Configuração de Endereços IP . . . 282
6.4.3.2. Configuração de NIC Teaming . . . 282
6.4.3.3. Configurações dos Módulos NX5001 . . . 282
6.4.3.4. Configurações dos Módulos NX5000 . . . 282
6.4.3.5. Configurações da UCP NX3035 Relacionadas com Redundância . . . 282
6.4.4. Configurações de Drivers de E/S . . . 283
6.4.5. Configurações da MainTask . . . 284
6.4.5.1. Programa ActivePrg . . . 284
6.4.5.2. Programa NonSkippedPrg . . . 285
6.4.6. Objeto *Redundancy Configuration* . . . 285
6.4.7. GVL *Module_Diagnostics* e GVL *System_Diagnostics* . . . 286
6.4.8. GVLs com Variáveis Simbólicas Redundantes ou Não-Redundantes . . . 286
6.4.9. GVLs com Diretivas AT . . . 286
6.4.10. POUs do Tipo Programa com Variáveis Simbólicas Redundantes ou Não-Redundantes . . . 286
6.4.11. Utilização de Breakpoints em Sistemas Redundantes . . . 286
6.4.12. Limitações na Programação de um CP Redundante . . . 286
6.4.12.1. Limitações em GVLs e POUs Redundantes . . . 287
6.4.12.2. Limitações em POUs Não-Redundantes . . . 287
6.4.13. Obtendo o Estado da Redundância de um Half-Cluster . . . 287
6.4.14. Utilização de Diagnósticos Não-Redundantes . . . 287
6.4.15. Exemplo de Gerenciamento de um Cliente MODBUS TCP num CP Redundante . . . 288

6.4.15.1. Configuração das Interfaces NET3 / NET4 do CP Redundante . . . 289
6.4.15.2. Configuração do Servidor MODBUS TCP para Comunicação de Teste . . . 289
6.4.15.3. Configuração do Cliente MODBUS TCP . . . 290
6.4.15.4. GVLs e POUs para Gerenciamento do Cliente MODBUS TCP . . . 290
6.5. Carga de Programas em um CP Redundante . . . 295
6.5.1. Carga Inicial de um Projeto Redundante . . . 295
6.5.2. Conexão do Mastertool com uma UCP NX3035 de um CP Redundante . . . 298
6.5.3. Carga de Modificações em um Projeto Redundante . . . 299
6.5.4. Carga de Modificações Offline e Online . . . 299
6.5.4.1. Modificações que Demandam Carga Offline com Interrupção do Controle do Processo . . . 299
6.5.4.2. Modificações que Demandam Carga Offline . . . 300
6.5.4.3. Modificações que Permitem Carga Online . . . 300
6.5.5. Carga Online de Modificações . . . 300
6.5.6. Carga Offline de Modificações com Interrupção do Controle do Processo . . . 301
6.6. Manutenção da Redundância . . . 302
6.6.1. Troca a Quente de Módulos em um CP Redundante . . . 302
6.6.2. Mensagens de Advertência do Mastertool . . . 303
6.6.2.1. Bloqueio de Carga de um Projeto Redundante ou Não-Redundante . . . 303
6.6.2.2. Alertas antes de Comandos que Podem Parar o CP Ativo . . . 303
6.6.2.3. Alerta antes de Logar-se ao CP Não-Ativo . . . 304
6.6.3. Diagnósticos da Redundância no Visor Gráfico da UCP NX3035 . . . 304
6.6.3.1. Estado de Redundância do CP . . . 304
6.6.3.2. Telas abaixo do Menu REDUNDANCIA . . . 305
6.6.4. Estrutura de Diagnósticos, Comandos e Informações do Usuário para a Redundância . . . 305
6.6.4.1. Diagnósticos da Redundância . . . 305
6.6.4.2. Comandos da Redundância . . . 311
6.6.4.3. Informações do Usuário Trocadas entre CPs Local e Remoto . . . 313
6.6.4.4. Diagnósticos MODBUS utilizados na Redundância . . . 313
6.6.4.5. Log de Eventos da Redundância . . . 314
7. Manutenção . . . 315
7.1. Diagnósticos do Módulo . . . 315
7.1.1. One Touch Diag . . . 315
7.1.2. Diagnósticos via LED . . . 318
7.1.2.1. DG (Diagnóstico) . . . 318
7.1.2.2. WD (Cão-de-guarda) . . . 318
7.1.2.3. LEDs Conector RJ45 . . . 319
7.1.3. Diagnósticos via Página Web de Sistema . . . 319
7.1.4. Diagnósticos via Variáveis . . . 322
7.1.4.1. Diagnósticos Resumidos . . . 322
7.1.4.2. Diagnósticos Detalhados . . . 324
7.1.4.2.1. Diagnósticos Detalhados Grupo Target . . . 324
7.1.4.2.2. Diagnósticos Detalhados Grupo Hardware . . . 325
7.1.4.2.3. Diagnósticos Detalhados Grupo Exception . . . 325
7.1.4.2.4. Diagnósticos Detalhados Grupo RetainInfo . . . 326
7.1.4.2.5. Diagnósticos Detalhados Grupo Reset . . . 326
7.1.4.2.6. Diagnósticos Detalhados Grupo Thermometer . . . 326
7.1.4.2.7. Diagnósticos Detalhados Grupo Serial . . . 327

7.1.4.2.8. Diagnósticos Detalhados Grupo Ethernet . . . 327
7.1.4.2.9. Diagnósticos Detalhados Grupo UserFiles . . . 329
7.1.4.2.10. Diagnósticos Detalhados Grupo UserLogs . . . 329
7.1.4.2.11. Diagnósticos Detalhados Grupo MemoryCard . . . 329
7.1.4.2.12. Diagnósticos Detalhados Grupo WHSB . . . 330
7.1.4.2.13. Diagnósticos Detalhados Grupo Application . . . 331
7.1.4.2.14. Diagnósticos Detalhados Grupo SNTP . . . 331
7.1.4.2.15. Diagnósticos Detalhados Grupo Rack . . . 332
7.1.4.2.16. Diagnósticos Detalhados Grupo ApplicationInfo . . . 332
7.1.4.2.17. Diagnósticos Detalhados Grupo Firewall . . . 332
7.1.4.2.18. Diagnósticos Detalhados Grupo OpenVPN . . . 333
7.1.4.2.19. Diagnósticos Detalhados Grupo FTP . . . 333
7.1.5. Diagnósticos via Blocos Funcionais . . . 333
7.1.5.1. GetTaskInfo . . . 333
7.2. Visor Gráfico . . . 335
7.3. Log de Sistema . . . 337
7.4. Não Carregar a Aplicação na Inicialização . . . 337
7.5. Falha na Alimentação . . . 337
7.6. Problemas mais Comuns . . . 338
7.7. Solução de Problemas . . . 338
7.8. Manutenção Preventiva . . . 339
8. Apêndices . . . 340
8.1. Gerenciamento de Certificados e Chaves TLS . . . 340
8.1.1. Geração de Certificados por Easy-RSA . . . 340
8.1.2. Geração de Certificados por OpenSSL . . . 345
8.1.3. Geração de Chave TA pelo OpenVPN . . . 347
8.2. Carga Offline de Programas sem Interrupção do Controle do Processo . . . 347
8.2.1. Recomendações Gerais . . . 348
8.2.2. Alterações de PROFIBUS . . . 348
8.2.3. Condição Inicial . . . 348
8.2.4. Procedimento para Alteração . . . 349
8.2.5. Passos . . . 349
8.2.5.1. Etapa 1 – Preparação da alteração . . . 349
8.2.5.2. Etapa 2 – Download no CP Reserva . . . 349
8.2.5.3. Etapa 3 – Troca de controle . . . 349
8.2.5.4. Etapa 4 – Validação final . . . 349

1. Introdução

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

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

Com processamento de 64 bits e ampla capacidade de memória, o NX3035 é projetado para atender aplicações que demandam alta performance, disponibilidade e confiabilidade. Com 384 Kbytes para pontos %I e %Q e 8 Mbytes de memória retentiva ou persistente, a UCP oferece suporte a aplicações robustas, enquanto seu desempenho é assegurado pela unidade de ponto flutuante integrada. Equipada com seis interfaces Ethernet de alta velocidade, duas interfaces SFP para sincronismo de redundância e uma serial RS-485, o NX3035 garante conectividade confiável e suporte de redundância, ideal para sistemas críticos. Sua compatibilidade com protocolos como OPC DA/UA, PROFINET, EtherCAT, MODBUS TCP, SNMP e EtherNet/IP, permite integração total com diferentes plataformas e dispositivos. Recursos como sincronismo de relógio via SNTP e um relógio em tempo real proporcionam maior precisão. Combinando design compacto e diagnóstico avançado, seu suporte para cartões microSD amplia a flexibilidade de armazenamento, e o modo de redundância reforça a disponibilidade em operações contínuas.



Figura 1: NX3035

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.

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

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

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

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

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

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

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

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

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

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

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

1.3. Documentos Relacionados a este Manual

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

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

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

Código	Descrição	Idioma
CE114000	Nexto Series – Technical Characteristics	Inglês
CT114000	Série Nexto – Características Técnicas	Português
CE114108	NX3035 Technical Characteristics	Inglês
CT114108	Características Técnicas NX3035	Português
CE114200	NX8000 Power Supply Module Technical Characteristics	Inglês
CT114200	Características Técnicas Fonte de Alimentação NX8000	Português
CE114700	Nexto Series Backplane Racks Technical Characteristics	Inglês
CT114700	Características Técnicas dos Bastidores da Série Nexto	Português
CE114810	Nexto Series Accessories for Backplane Rack Technical Characteristics	Inglês
CT114810	Características Técnicas Acessórios para Bastidor Série Nexto	Português
CE114902	Nexto Series PROFIBUS-DP Master Technical Characteristics	Inglês
CT114902	Características Técnicas do Mestre PROFIBUS-DP da Série Nexto	Português
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
CE157204	NX9500 / NX9501 Technical Characteristics	Inglês
CT157204	Características Técnicas NX9500 / NX9501	Português
MU214600	Nexto Series User Manual	Inglês
MU214000	Manual de Utilização Série Nexto	Português
MU214619	NX3035 CPU User Manual	Inglês
MU214107	Manual de Utilização UCP NX3035	Português
MU299609	MasterTool IEC XE User Manual	Inglês

Código	Descrição	Idioma
MU299048	Manual de Utilização MasterTool IEC XE	Português
MP399609	IEC 61131 Programming Manual	Inglês
MP399048	Manual de Programação IEC 61131	Português
MU214601	NX5001 PROFIBUS DP Master User Manual	Inglês
MU214001	Manual de Utilização Mestre PROFIBUS-DP NX5001	Português
MU214608	Nexto PROFIBUS-DP Head Utilization Manual	Inglês
MU214108	Manual de Utilização da Cabeça PROFIBUS-DP Nexto	Português
MU219000	Ponto Series Utilization Manual	Inglês
MU209000	Manual de Utilização da Série Ponto	Português
MU209508	Manual de Utilização Cabeça PROFIBUS PO5063V1 e Cabeça Redundante PROFIBUS PO5063V5	Português
MU219511	PO5064 PROFIBUS Head and PO5065 Redundant PROFIBUS Head Utilization Manual	Inglês
MU209511	Manual de Utilização Cabeça PROFIBUS PO5064 e Cabeça Redundante PROFIBUS PO5065	Português
MU209020	Manual de Utilização Rede HART sobre PROFIBUS	Português
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
NAP169	RSTP in Nexto CPUs	Inglês

Tabela 1: Documentos Relacionados

1.4. Inspeção Visual

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

CUIDADO

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

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

1.5. Suporte Técnico

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

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

1.6. Mensagens de Advertência Utilizadas neste Manual

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

PERIGO

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

CUIDADO

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

ATENÇÃO

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

2. Descrição Técnica

Este capítulo apresenta todas as características técnicas da UCP NX3035 da Série Nexto.

2.1. Painéis e Conexões

A figura a seguir mostra o painel frontal da UCP.



Figura 3: NX3035

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

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

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

Tabela 2: Descrição dos LEDs

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

Tecla	Descrição
Tecla de Diagnóstico	Tecla situada na parte superior do módulo. Utilizada para visualização dos diagnósticos no visor gráfico ou para navegação no menu informativo e de configurações da UCP.

Tabela 3: Descrição da Tecla

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

Interfaces	Descrição
NET 1	Conector tipo RJ45 de comunicação no padrão 10/100/1000Base-TX. Permite a comunicação ponto a ponto ou em rede. Para obter mais informações sobre utilização, consulte a seção Configuração das Interfaces Ethernet Integradas.
NET 2	Conector tipo RJ45 de comunicação no padrão 10/100/1000Base-TX. Permite a comunicação ponto a ponto ou em rede. Para obter mais informações sobre utilização, consulte a seção Configuração das Interfaces Ethernet Integradas.
NET 3	Conector tipo RJ45 de comunicação no padrão 10/100/1000Base-TX. Permite a comunicação ponto a ponto ou em rede. Para obter mais informações sobre utilização, consulte a seção Configuração das Interfaces Ethernet Integradas.
NET 4	Conector tipo RJ45 de comunicação no padrão 10/100/1000Base-TX. Permite a comunicação ponto a ponto ou em rede. Para obter mais informações sobre utilização, consulte a seção Configuração das Interfaces Ethernet Integradas.
NET 5	Conector tipo RJ45 de comunicação no padrão 10/100/1000Base-TX. Permite a comunicação ponto a ponto ou em rede. Para obter mais informações sobre utilização, consulte a seção Configuração das Interfaces Ethernet Integradas.
NET 6	Conector tipo RJ45 de comunicação no padrão 10/100/1000Base-TX. Permite a comunicação ponto a ponto ou em rede. Para obter mais informações sobre utilização, consulte a seção Configuração das Interfaces Ethernet Integradas.
NET A	Conector tipo SFP de comunicação para o link de redundância. (Parte de baixo do produto.)
NET B	Conector tipo SFP de comunicação para o link de redundância. (Parte de baixo do produto.)
COM 1	Conector tipo DB9 fêmea para comunicação no padrão RS-485. Para obter mais informações sobre utilização, consulte a seção Configuração da Interface Serial.
MEMORY SLOT	Conector para cartão de memória. Permite utilizar um cartão de memória para diferentes tipos de armazenamento de dados como: logs de usuário, páginas Web, 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

	NX3035
Ocupação do bastidor	4 posições sequenciais
Consumo de corrente do bastidor	2600 mA
Fonte de alimentação integrada	Não
Ethernet TCP/IP interface integrada	6
Interface Serial	1
Interface CAN	Não
Porta USB Host	Não
Interface Cartão de Memória	1
Relógio de tempo real (RTC)	Sim Resolução de 1 ms, máx. variação de 2 segundos por dia.
Cão de guarda	Sim
Indicação de estado e diagnóstico	Visor gráfico LEDs Páginas Web 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) Contínua (execução livre) Disparada por estado (interrupção de software)
Alterações online	Sim
Número máximo de tarefas	32
Número máximo de barramentos de expansão	24
Suporte a redundância de expansão de barramento	Sim
Número máximo total de módulos de E/S nos barramentos locais da UCP	128
Número máximo de módulos adicionais de interface Ethernet TCP/IP (NX5000)	6
Suporte a redundância de interfaces Ethernet TCP/IP	Sim
Número máximo de redes PROFIBUS-DP (usando módulos mestres PROFIBUS-DP)	6
Suporte a redundância de rede PROFIBUS-DP	Sim
Suporte a redundância (half-clusters)	Sim
Suporte a troca a quente	Sim
Registro de eventos (SOE) **Protocolo** **Tamanho máximo da fila de eventos**	Não - -
Páginas web de usuário (Webvisu)	Não
Firewall	Sim

	NX3035
VPN	Sim
FTP	Sim
Docker	Não
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: O número máximo de módulos de E/S refere-se a soma de todos os módulos do barramento local e das expansões. Módulos adicionais de E/S podem ser instalados em remotas (PROFIBUS, MODBUS, PROFINET, Ethernet/IP, etc).

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)
	TR 004/2011 (LVD) CU TR 020/2011 (EMC)

Tabela 6: Normas e Certificações

2.2.3. Memória

	NX3035
Memória de variáveis de entrada de representação direta (%I)	384 Kbytes
Memória de variáveis de saída de representação direta (%Q)	384 Kbytes
Memória de variáveis de representação direta (%M)	128 Kbytes
Memória de variáveis simbólicas	48 Mbytes
Memória de variáveis retentivas ou persistentes	8 Mbytes
Memória de dados redundantes total	2912 Kbytes
Memória de variáveis de entrada de representação direta (%I)	368 Kbytes
Memória de variáveis de saída de representação direta (%Q)	368 Kbytes
Memória de variáveis de representação direta (%M)	128 Kbytes
Memória de variáveis simbólicas	2912 Kbytes
Memória de programa	64 Mbytes
Memória de código fonte (backup)	256 Mbytes
Memória de arquivos de usuário	2 Gbytes

Tabela 7: Memória

Notas:

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

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

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

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

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

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

ATENÇÃO

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

A lista completa de quando as variáveis persistentes simbólicas mantêm seus valores e quando o valor é perdido pode ser encontrada na tabela abaixo. Além do tamanho da área persistente declarado na tabela acima, esses 44 bytes são reservados para armazenar informações sobre as variáveis persistentes (não disponíveis para uso).

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

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

Tabela 8: Comportamento das Variáveis após Evento

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

Memória de dados redundantes total: *Memória de dados redundantes total* 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 (*Memória de variáveis de entrada de representação direta*, *Memória de variáveis de saída de representação direta*, *Memória de variáveis de representação direta* e *Memória de variáveis simbólicas*) deve ser igual ou menor que a *Memória de dados redundantes total*.

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 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. A capacidade utilizável pode ser menor que a capacidade nominal especificada devido a arquivos de sistema e formatação. Maiores informações no capítulo Configuração - Memória de Arquivos de Usuário.

2.2.4. Protocolos

	NX3035 CP Simples	NX3035 CP Redundante	Interface
Protocolo aberto	Sim	Sim	COM1
MODBUS RTU Mestre	Sim	Sim	COM1
MODBUS RTU Escravo	Sim	Sim	COM1
MODBUS TCP Cliente	Sim	Sim	NET1 ... NET6
MODBUS TCP Servidor	Sim	Sim	NET1 ... NET6
MODBUS RTU via TCP Cliente	Sim	Sim	NET1 ... NET6
MODBUS RTU via TCP Servidor	Sim	Sim	NET1 ... NET6
CANopen Mestre	Não	Não	-
CANopen Escravo	Não	Não	-
CAN low level	Não	Não	-
SAE J-1939	Não	Não	-
OPC DA Servidor	Sim	Sim	NET1 ... NET6
OPC UA Servidor	Sim	Não	NET1 ... NET6
EtherCAT Mestre	Sim	Não	NET1 ... NET6
SNMP Agente	Sim	Sim	NET1 ... NET6
SOE (dados orientados ao evento)	Não	Não	-
DNP3 Servidor	Não	Não	-
IEC 60870-5-104 Servidor	Não	Não	-
EtherNet/IP Scanner	Sim	Não	NET1 ... NET6
EtherNet/IP Adapter	Sim	Não	NET1 ... NET6
MQTT Cliente	Sim	Sim	NET1 ... NET6
SparkPlugB	Sim	Não	NET1 ... NET6
SNTP Cliente (para sincronismo do relógio)	Sim	Sim	NET1 ... NET6
PROFINET Controller	Sim	Não	NET1 ... NET6
PROFINET Device	Não	Não	-
OpenVPN Client	Sim	Sim	NET1 ... NET6
OpenVPN Server	Sim	Sim	NET1 ... NET6
FTP Server	Sim	Sim	NET1 ... NET6
RSTP	Sim	Não	NET1 ... NET6
MRP	Sim	Não	NET1 ... NET6
SysLog	Sim	Sim	NET1 ... NET6

Tabela 9: Protocolos

2.2.5. Interface Serial

2.2.5.1. COM 1

	COM 1
Conector	DB9 fêmea blindado
Interface Física	RS-485
Direção de Comunicação	half duplex
Máx. Transmissores RS-485	32
Terminação	Sim (opcional via seleção de cabo)
Taxa de Transmissão	9600, 19200, 38400, 57600, 115200 bps
Isolação **Lógica para porta serial** **Porta serial para terra de proteção** ⏚	 1000 Vac / 1 minuto 1000 Vac / 1 minuto

Tabela 10: Características da Interface Serial COM 1

Nota:

Meio físico: A lista dos cabos pode ser encontrada na seção Produtos Relacionados.

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 6

	NET 1 / NET 2 / NET 3 / NET 4 / NET 5 / NET 6
Conector	RJ45 fêmea blindado
Auto crossover	Sim
Máximo Comprimento de Cabo	100 m
Tipo de Cabo	UTP ou ScTP, categoria 5e ou superior
Taxa de Transmissão	10/100/1000 Mbps
Camada Física	10BASE-TE/100BASE-TX/1000BASE-T
Camada de Enlace	LLC (Controle de Enlace Lógico)
Camada de Rede	IP (Protocolo de Internet)
Camada de Transporte	TCP (Protocolo de Controle de Transmissão) UDP (Protocolo de Datagrama de Usuário)
Diagnósticos	LED - verde 1000 Mbps (link/atividade) LED – amarelo 100 Mbps (link/atividade) LEDs – verde e amarelo 10 Mbps (link/atividade)
Isolação **Interfaces Ethernet para lógica e terra** **Interface Ethernet para Interface Ethernet**	 1500 Vac / 1 minuto 1500 Vac / 1 minuto

Tabela 11: Características da Interface Ethernet NET 1 à NET 6

A Interface NET 1 é a interface utilizada para programação usando a ferramenta Mastertool.

2.2.7. Link de Redundância

2.2.7.1. NET A / NET B

	NET A / NET B
Conector	Receptáculo SFP
Taxa de dados interna	1.25 Gbps
Proteção de sobre corrente	Sim
Troca a quente	Sim

Tabela 12: Características do Link de Redundância NET A e NET B

Nota:
A lista dos transceptores SFP compatíveis podem ser encontradas na seção Produtos Relacionados.

2.2.8. Interface do Cartão de Memória

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

	Cartão de Memória
Capacidade máxima	32 Gbytes
Capacidade mínima	2 Gbytes
Tipo	MicroSD
Sistema de arquivos	FAT32
Remover cartão de forma segura	Sim, através de menu específico para essa função.

Tabela 13: 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.

2.2.9. Características Ambientais

	NX3035
Dissipação	11 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)	72,20 x 114,63 x 115,30 mm
Dimensões da embalagem (L x A x P)	77 x 119 x 145 mm

	NX3035
Peso	490 g
Peso com embalagem	620 g

Tabela 14: Características Ambientais

Nota:

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

2.3. Compatibilidade com Outros Produtos

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

UCPs da Série Nexto	Mastertool	Versão de Firmware
NX3035	3.75 ou superior	1.14.0.0 ou superior

Tabela 15: Compatibilidade com Outros Produtos

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

2.4. Desempenho

O desempenho das UCPs da Série Nexto depende:

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

2.4.1. Tempo de Intervalo da MainTask

A configuração de tempo de intervalo da MainTask depende do perfil de projeto selecionado. Para 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	Tipo da Variável	Tempo (μs)
1000 Contatos	LD	BOOL	1,05
1000 Divisões	LD, ST	INT	4,0
		REAL	2,3
1000 Multiplicações	LD, ST	INT	2,3
		REAL	2,3
1000 Somas	LD, ST	INT	2,3
		REAL	2,3

Tabela 16: 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.

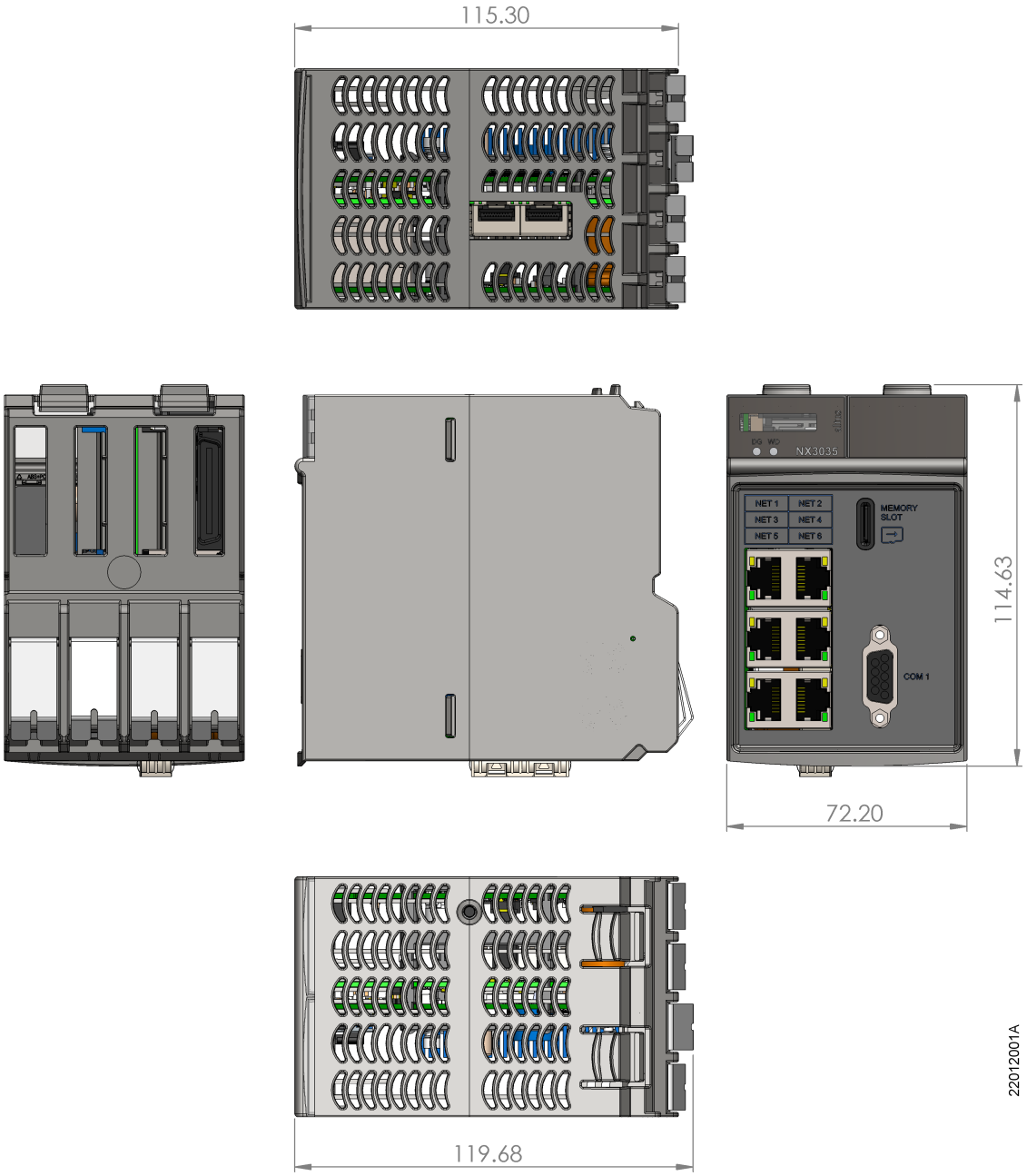


Figura 4: Dimensões Físicas

2.6. Dados para Compra

2.6.1. Itens Integrantes

A embalagem do produto contém os seguintes itens:

- Módulo NX3035

2.6.2. Código do Produto

Os seguintes códigos devem ser usados para compra do produto:

Código	Descrição
NX3035	UCP de alta velocidade, 6 portas Ethernet, 2 portas SFP, 1 canal serial, interface para cartão de memória, suporte à expansão de barramento e suporte à redundância

Tabela 17: Código do Produto

2.7. Produtos Relacionados

Os seguintes produtos devem ser adquiridos separadamente quando necessário:

Código	Descrição
MT8500	MasterTool IEC XE
AL-2600	Derivador e terminador de rede RS-485
AL-2306	Cabo RS-485 p/ rede MODBUS ou CAN
AL-2319	Cabo RJ45-RJ45
AL-1763	Cabo CMDB9-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
NX9500	Transceptor de fibra multimodo SFP Gigabit (550m)
NX9501	Transceptor de fibra monomodo SFP Gigabit (10Km)

Tabela 18: Produtos Relacionados

Notas:

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

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

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

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

AL-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.

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, bem como os cuidados que se deve ter com outras instalações existentes no painel elétrico ocupado pelo controlador.

CUIDADO

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

3.1. Instalação Mecânica

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.

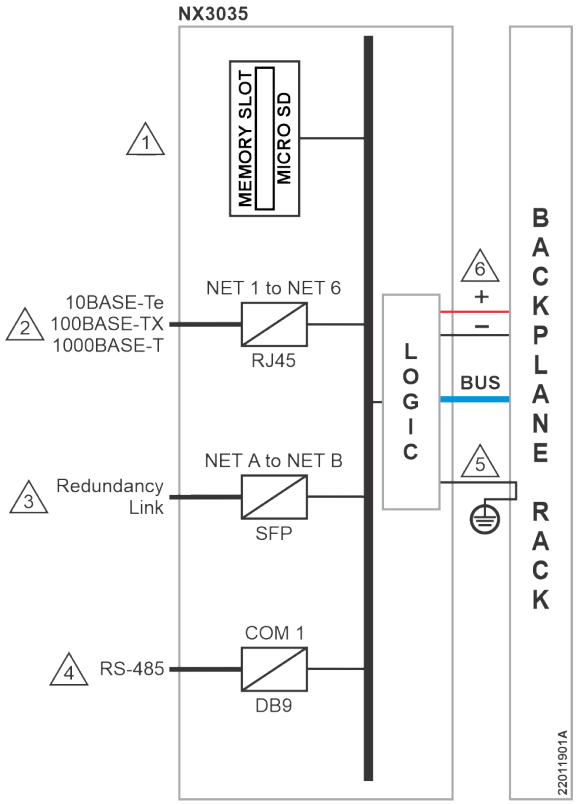


Figura 5: Diagrama Elétrico da UCP NX3035

Notas do Diagrama:

1. Interface para cartão microSD.
2. Interfaces Ethernet padrão 10BASE-Te/100BASE-TX/1000BASE-T.
3. Link de Redundância.
4. Interface RS-485.
5. A UCP é aterrada através dos bastidores da Série Nexto.
6. A alimentação do módulo é proveniente da conexão ao bastidor, não necessitando de conexões externas.

3.3. Conexão com a Rede Ethernet

As interfaces isoladas de comunicação NET 1 à NET 6 possibilitam a conexão com uma rede Ethernet, onde a interface NET 1 é mais indicada para comunicação com o Mastertool.

A conexão com a rede Ethernet utiliza cabos tipo par trançado (10BASE-Te/100BASE-TX/1000BASE-T), sendo que a detecção da velocidade é realizada automaticamente pela UCP Nexto. Este cabo deve ter uma de suas extremidades ligadas à interface que se pretende utilizar e a outra ao HUB, switch, microcomputador ou outro ponto de rede Ethernet.

3.3.1. Endereço IP

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

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

Tabela 19: Configuração de Parâmetros Padrão para a Interface Ethernet NET 1

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

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

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

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

Tabela 20: Configuração de Parâmetros Padrão para a Interface Ethernet NET 2

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

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

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

Tabela 21: Configuração de Parâmetros Padrão para a Interface Ethernet NET 3

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

Os parâmetros de rede da interface NET 3 podem ser alterados através do Mastertool. Para maiores detalhes sobre configuração e alteração de parâmetros de rede, verifique a seção Configuração das Interfaces Ethernet Integradas.

	NET 4
Endereço IP	192.168.18.1
Máscara de subrede	255.255.255.0
Endereço do Gateway	0.0.0.0

Tabela 22: Configuração de Parâmetros Padrão para a Interface Ethernet NET 4

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 4 podem ser alterados através do Mastertool. Para maiores detalhes sobre configuração e alteração de parâmetros de rede, verifique a seção Configuração das Interfaces Ethernet Integradas.

	NET 5
Endereço IP	192.168.19.1
Máscara de subrede	255.255.255.0
Endereço do Gateway	0.0.0.0

Tabela 23: Configuração de Parâmetros Padrão para a Interface Ethernet NET 5

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 5 podem ser alterados através do Mastertool. Para maiores detalhes sobre configuração e alteração de parâmetros de rede, verifique a seção Configuração das Interfaces Ethernet Integradas.

	NET 6
Endereço IP	192.168.20.1
Máscara de subrede	255.255.255.0
Endereço do Gateway	0.0.0.0

Tabela 24: Configuração de Parâmetros Padrão para a Interface Ethernet NET 6

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 6 podem ser alterados através do Mastertool. Para maiores detalhes sobre configuração e alteração de parâmetros de rede, verifique a seção Configuração das Interfaces Ethernet Integradas.

3.3.2. ARP Gratuito

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

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

3.3.3. Instalação do Cabo de Rede

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

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

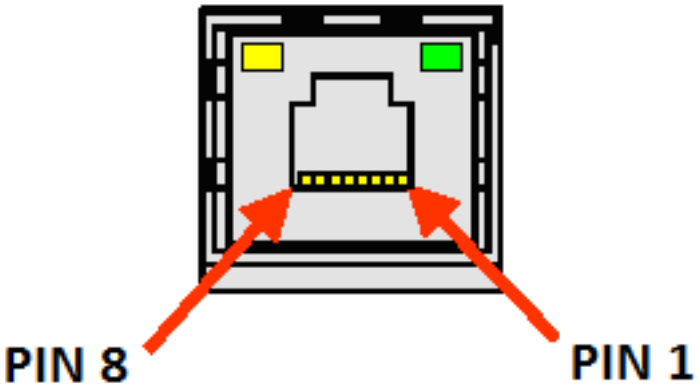


Figura 6: Conector Fêmea RJ45

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

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

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

Tabela 26: Pinagem do Conector Fêmea RJ45 - 1GBASE-T

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

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

Normalmente estes cabos possuem uma trava de conexão que garante uma perfeita conexão entre o conector fêmea da interface e o conector macho do cabo. No momento da instalação, o conector macho do cabo deve ser inserido no fêmea do

módulo até que se ouça um som característico ("click"), garantindo a atuação da trava. Para desconectar os mesmos deve-se utilizar a alavanca presente no conector macho.

3.4. Conexão com a Rede Serial RS-485

A interface isolada de comunicação COM 1, possibilita a conexão com uma rede RS-485 e utiliza o conector DB9 fêmea para a conexão serial, cuja identificação e a descrição dos sinais é mostrada a seguir.

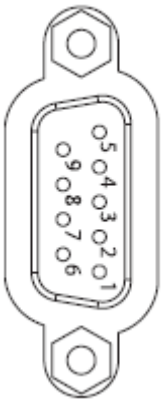


Figura 7: Conector DB9 Fêmea

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 27: COM 1

3.4.1. 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

Terminais do AL-1763	Sinais na Borneira da UCP
7	D-
8	D-

Tabela 28: 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.

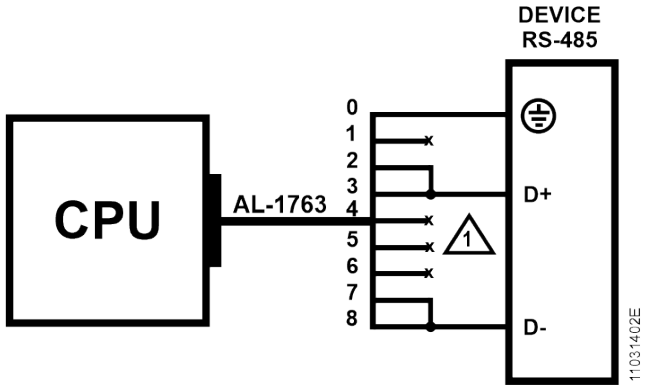


Figura 8: Diagrama de Conexão da Interface RS-485 sem Terminação

Nota do Diagrama:

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

3.4.2. 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 29: 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.

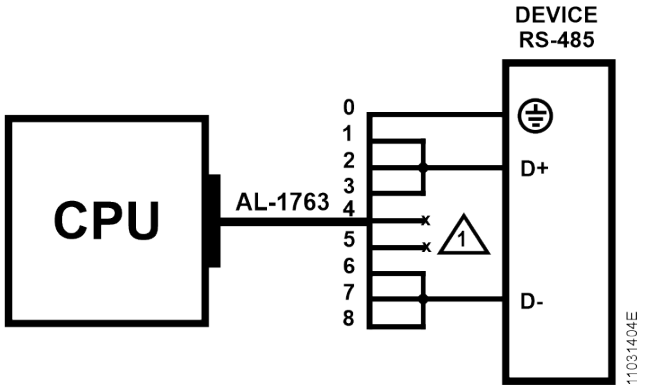


Figura 9: Diagrama de Conexão da Interface RS-485 com Terminação Interna

Nota do Diagrama:

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

3.4.3. 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 30: 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.

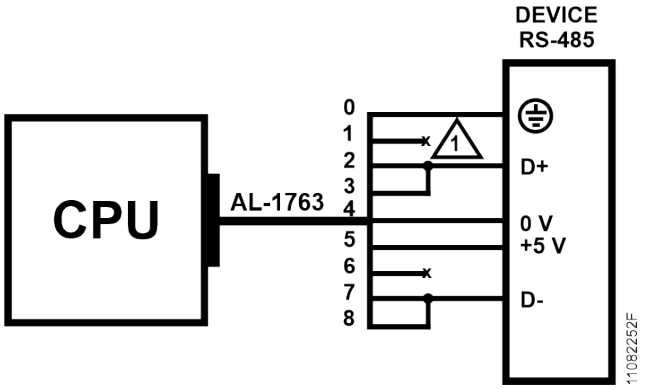


Figura 10: Diagrama de Conexão da Interface RS-485 com Terminação Externa

Nota do Diagrama:

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

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

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

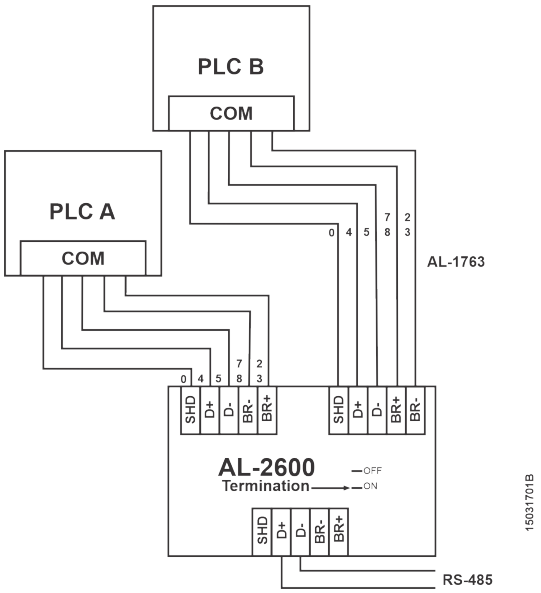


Figura 11: Diagrama de Conexão de Rede RS-485 com Terminação Externa e Redundância de Mestre

3.5. Instalação do Cartão de Memória

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

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

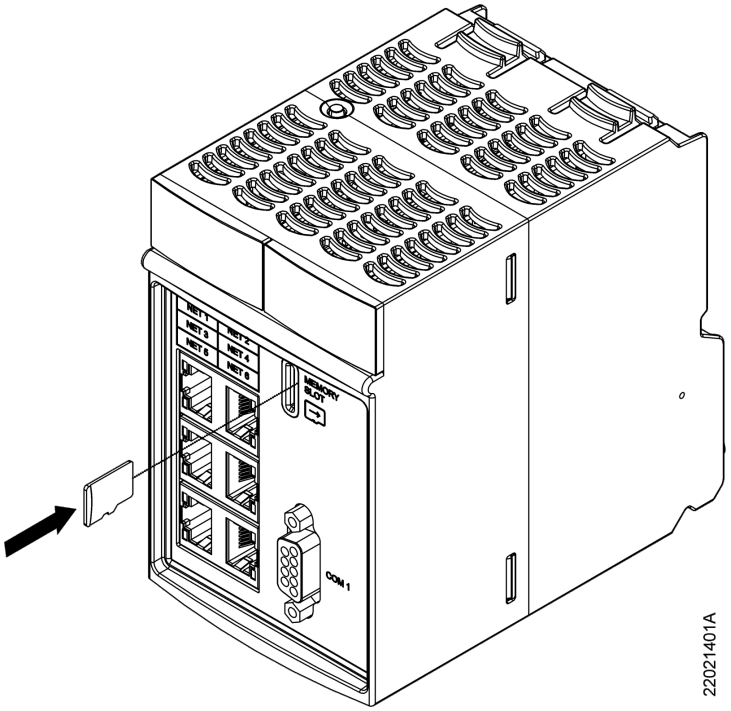


Figura 12: Inserção do Cartão de Memória na UCP

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

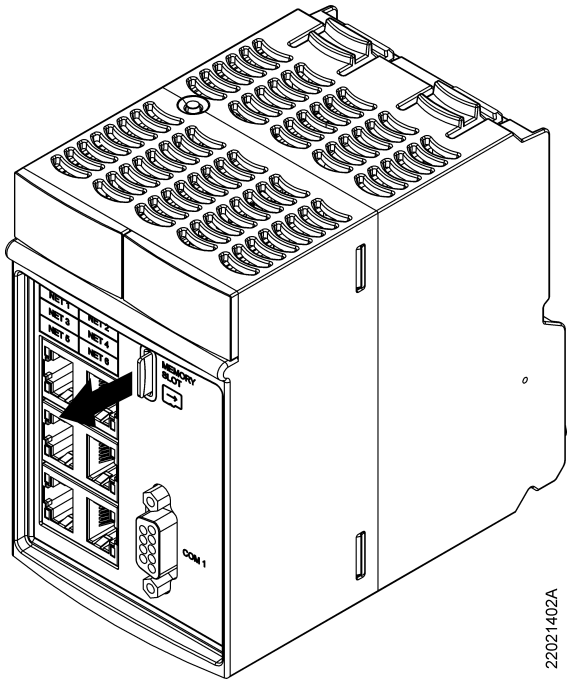


Figura 13: Remoção do Cartão de Memória

3.6. Instalação da Arquitetura

3.6.1. Instalação de Módulos no Bastidor Principal

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

3.7. Instalação do Programador

Para realizar a instalação do software de desenvolvimento Mastertool é necessário efetuar o download do arquivo de instalação no site http://www.altus.com.br. Para mais informações e o passo a passo a ser seguido, consulte o Manual de Utilização Mastertool, no capítulo Instalação.

4. Programação Inicial

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

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

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

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

4.1. Organização e Acesso à Memória

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

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

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

Tabela 31: Exemplo Organização e Acesso à Memória

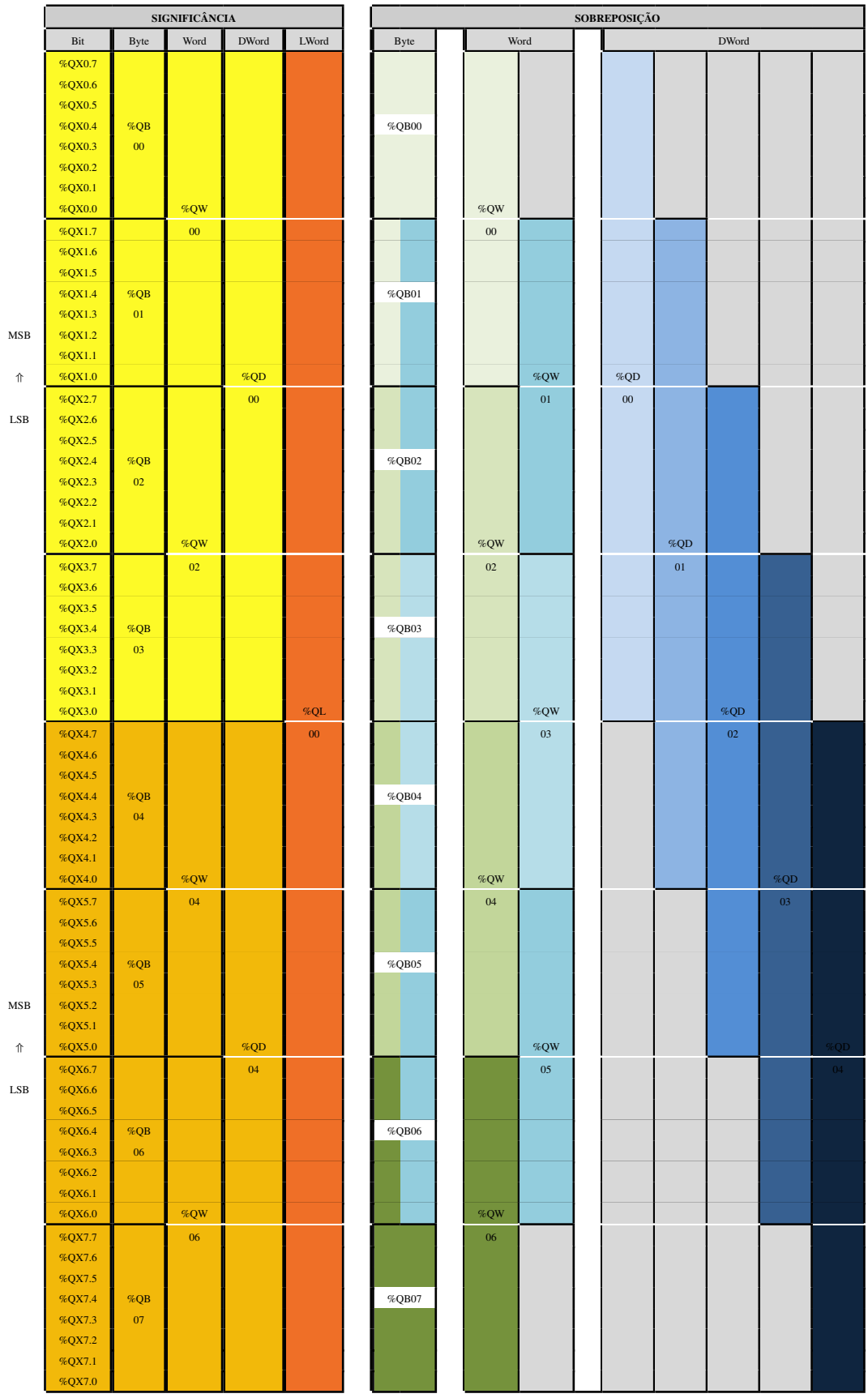


Tabela 32: Organização e Acesso à Memória

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 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 33: 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 34: 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 35: Tarefas no Perfil Normal

4.2.4. Experiente

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

adicionais podem ser livremente selecionadas na faixa de 08 a 12. Neste perfil, além dos programas padrões, cada tarefa pode executar programas adicionais.

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

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

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

Tarefa	POU	Prioridade	Tipo	Intervalo	Evento
MainTask	MainPrg	13	Cíclica	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 36: 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 37: 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 38: 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 39: 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	NX3035					
		S	B	N	E	P	M
Configuration Task (Tarefa WHSB)	Cíclica	1	1	1	1	1	0
Tarefas de Usuário	Cíclica	1	1	31	31	31	2
	Disparada por Evento	0	0	0	0	31	0
	Disp. Evento Externo	0	1	0	30	31	0
	Contínua	0	1	0	1	31	0
	Disparada por Estado	0	0	0	0	31	0
NETs – Instâncias Cliente ou Servidor	Cíclica	16					
COM (n) – Instâncias Mestre ou Escravo	Cíclica	1					
TOTAL		32					

Tabela 40: Número Máximo de Tarefas IEC NX3035

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.

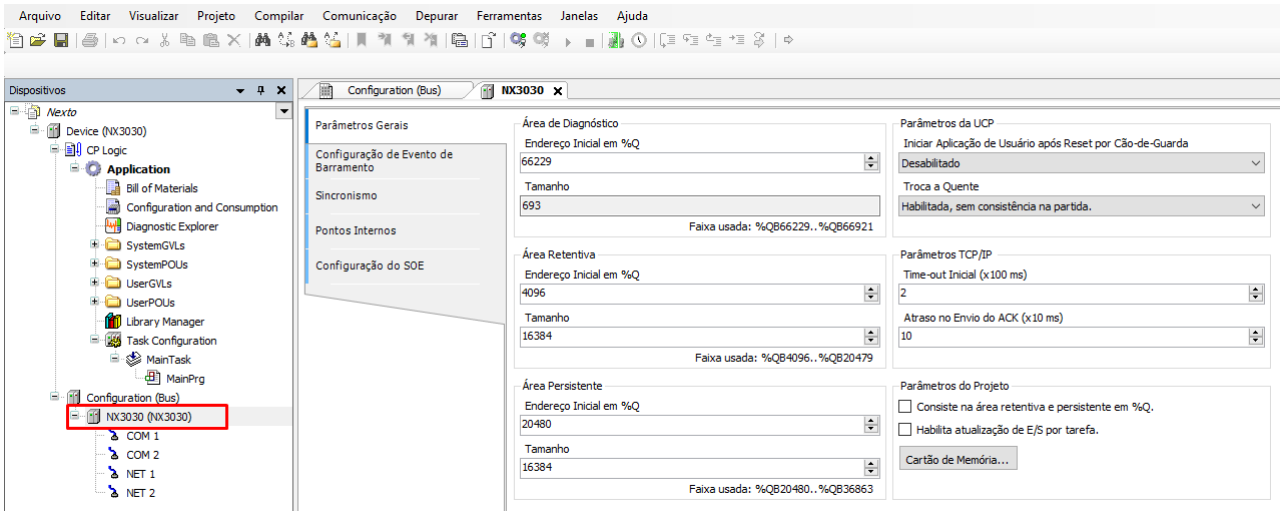


Figura 14: Configuração da UCP

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

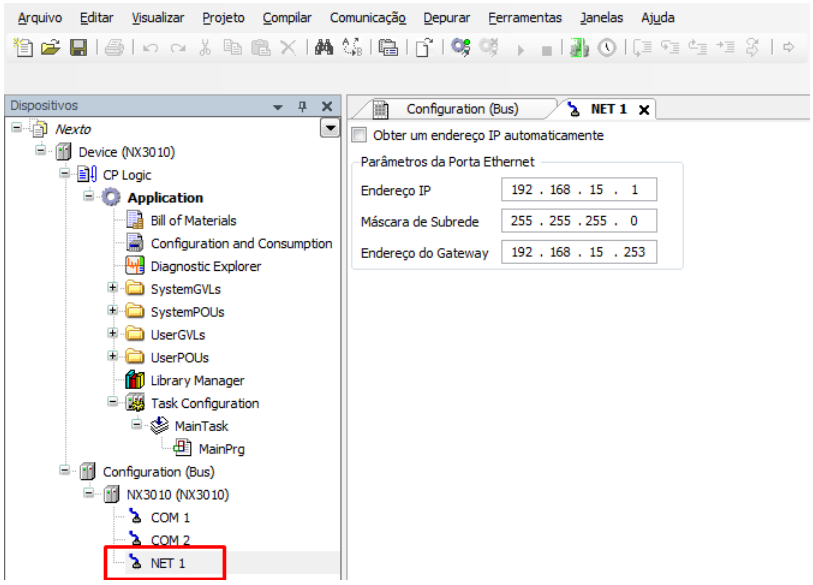


Figura 15: Configurando a Porta de Comunicação da UCP

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

4.4. Bibliotecas

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

4.5. Inserindo uma Instância de Protocolo

As UCPs da Série Nexto, conforme 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 RTU

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

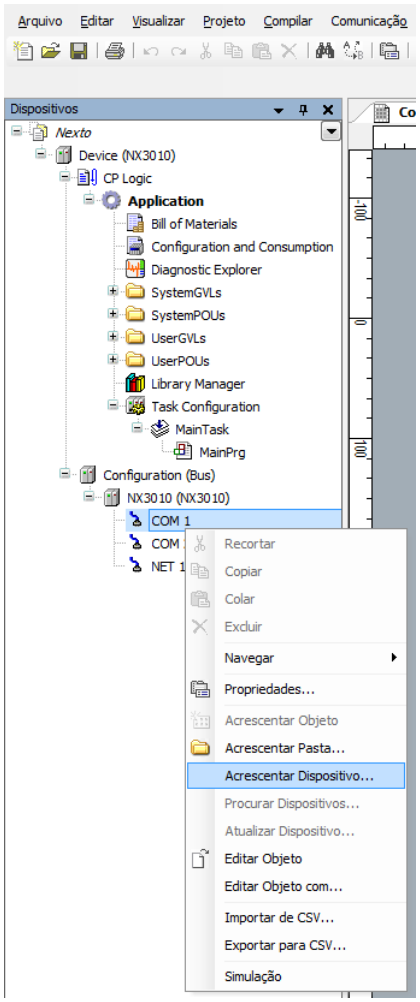


Figura 16: Adicionando a Instância

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

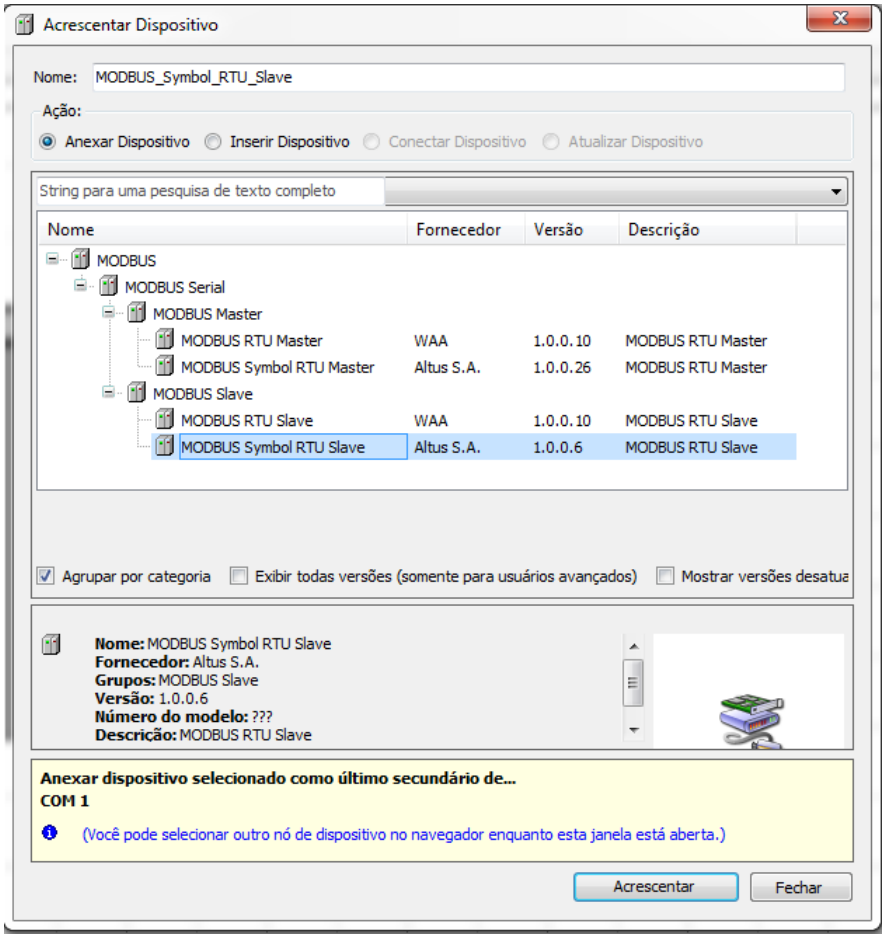


Figura 17: Selecionando o Protocolo

4.5.2. MODBUS Ethernet

O primeiro passo para configurar o MODBUS Ethernet (Cliente neste exemplo), é incluir a instância na NET desejada (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.

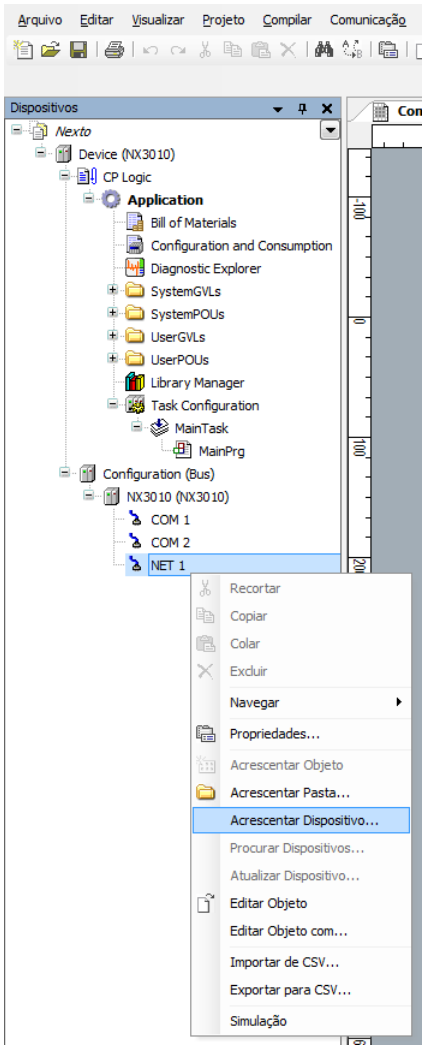


Figura 18: Adicionando a Instância

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.

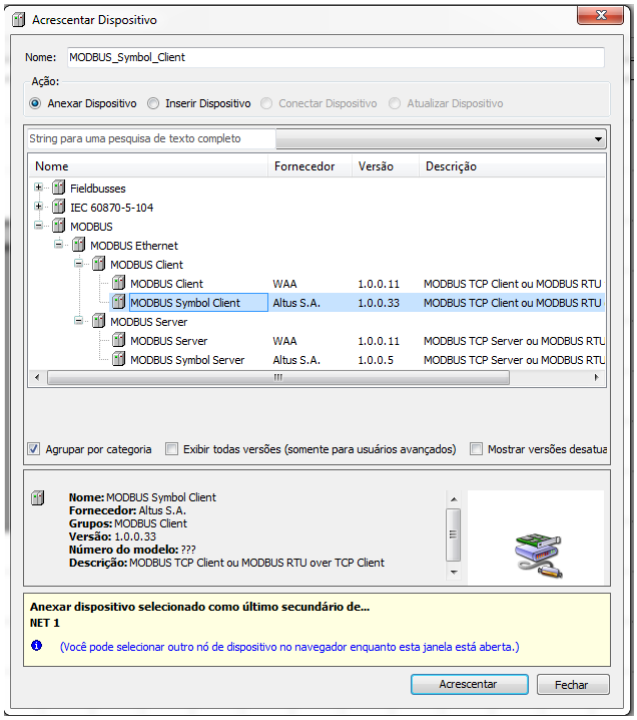


Figura 19: Selecionando o Protocolo

4.6. Localizando o Dispositivo

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

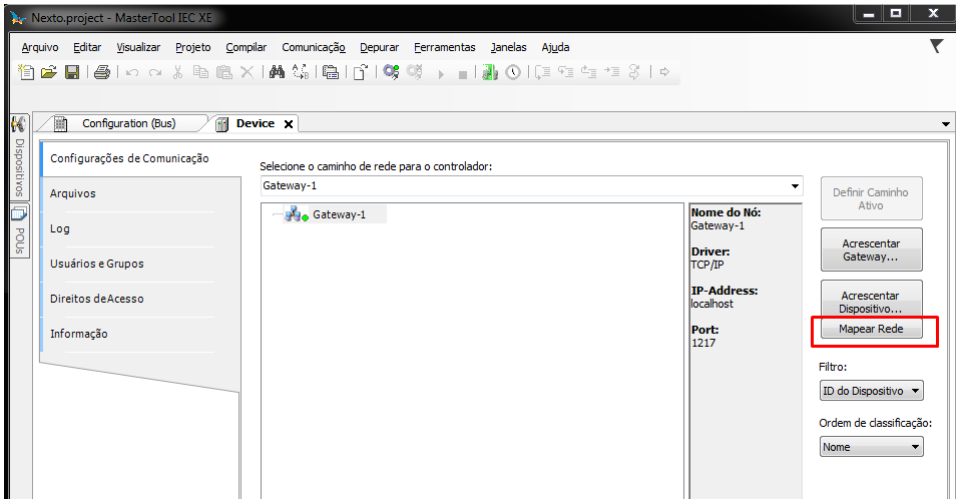


Figura 20: Localizando a UCP

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.

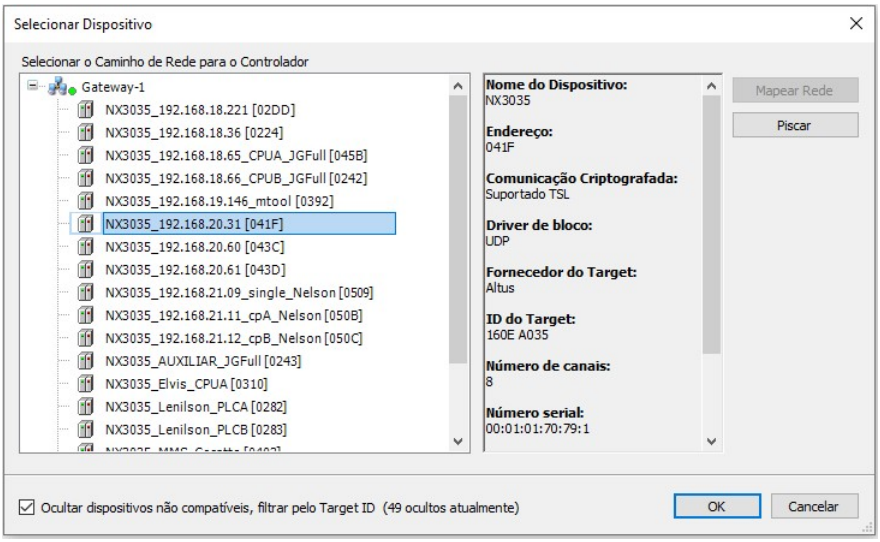


Figura 21: Selecionando a UCP

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

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

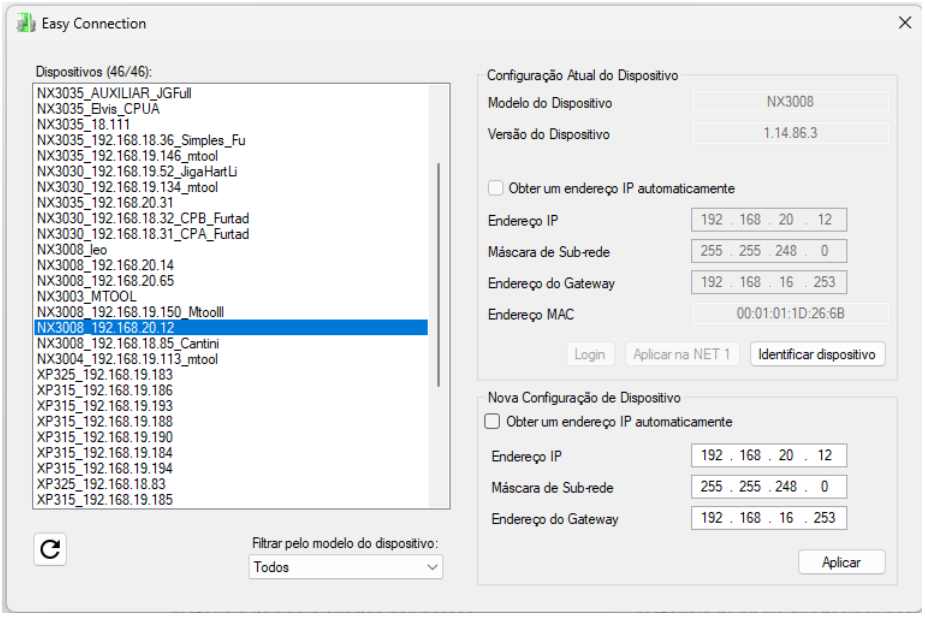


Figura 22: Easy Connection

Este comando realiza uma comunicação em nível MAC com a interface NET 1 do dispositivo, permitindo alterar permanentemente a configuração da interface Ethernet da UCP, independentemente da configuração IP da estação e daquela previamente existente no dispositivo. Com isso, é possível configurar o dispositivo para que fique na mesma rede da interface Ethernet da estação onde está sendo executada a ferramenta Mastertool, permitindo localizar e selecionar o dispositivo para comunicação. A descrição completa do comando *Easy Connection* pode ser encontrada no Manual de Utilização do Mastertool.

4.7. Login

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

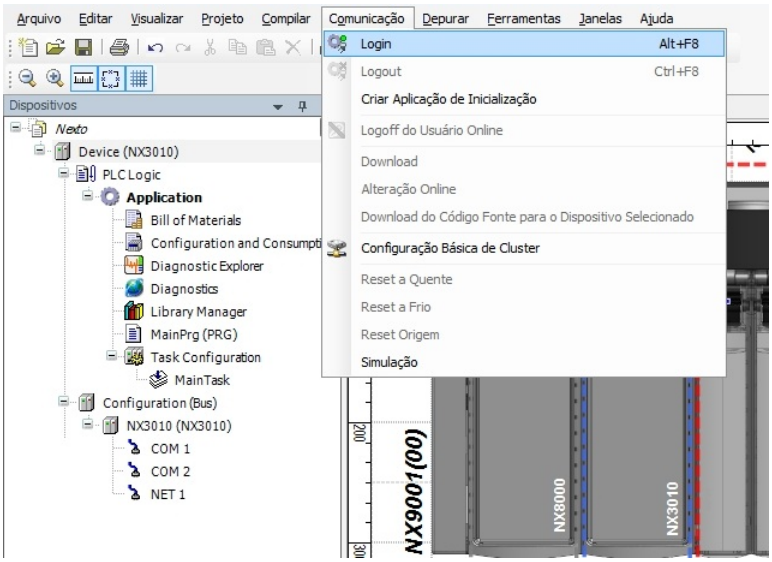


Figura 23: Enviando o Projeto para a UCP

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.

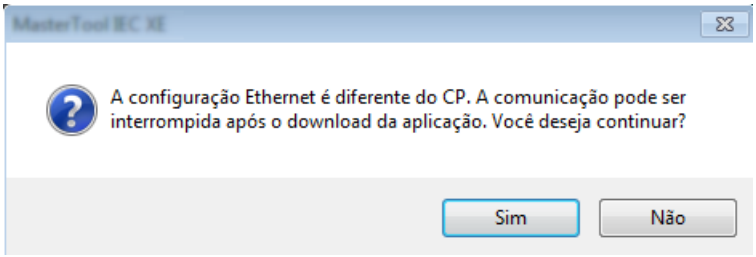


Figura 24: Aviso de configuração IP

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

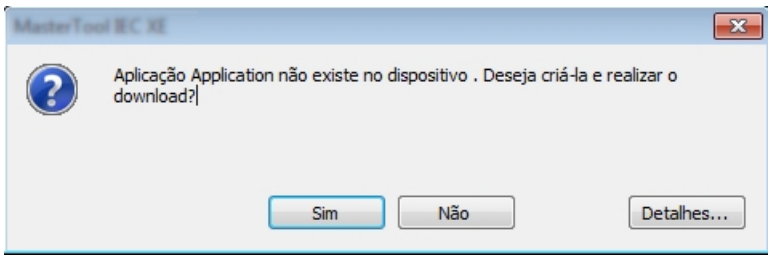


Figura 25: Sem aplicação no dispositivo

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.

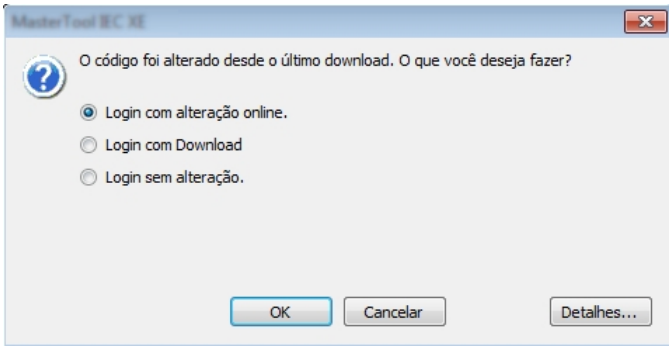


Figura 26: Atalização do Projeto na UCP

ATENÇÃO

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

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

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

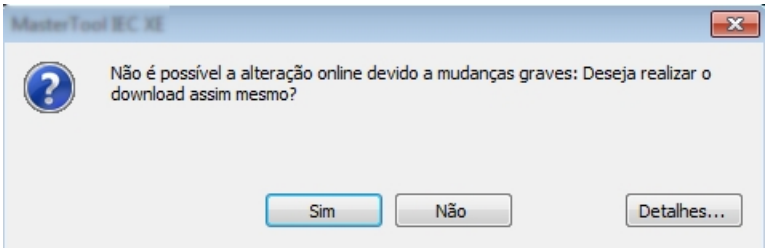


Figura 27: Alteração de Configuração

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

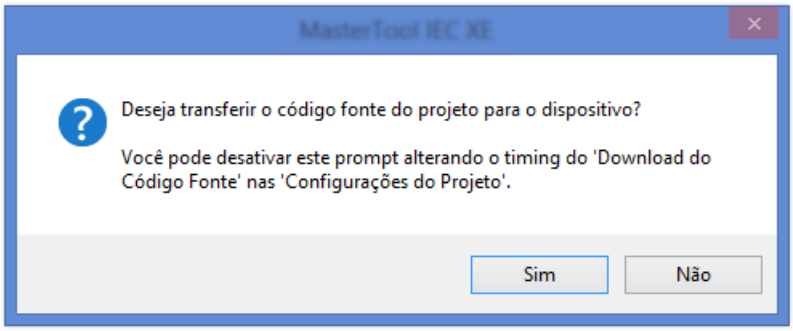


Figura 28: Download de código fonte

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.

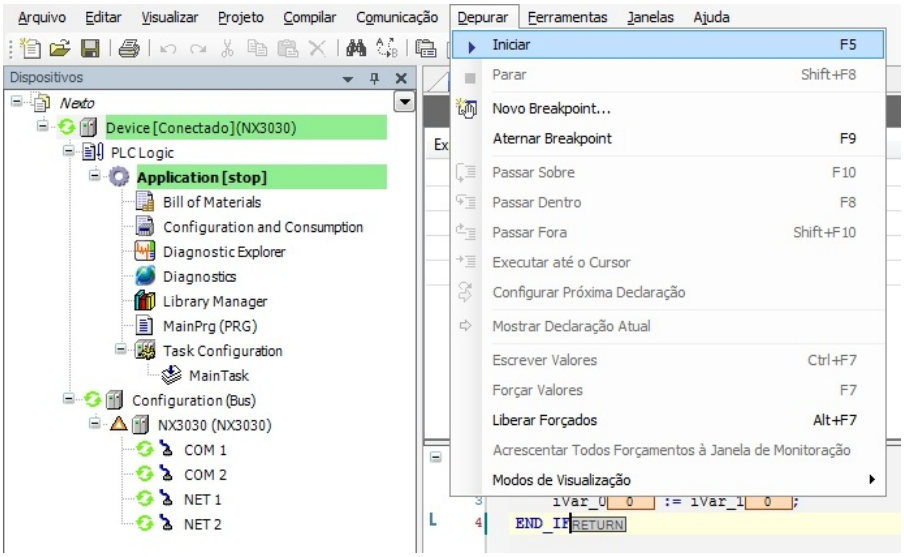


Figura 29: Iniciando a Aplicação

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.

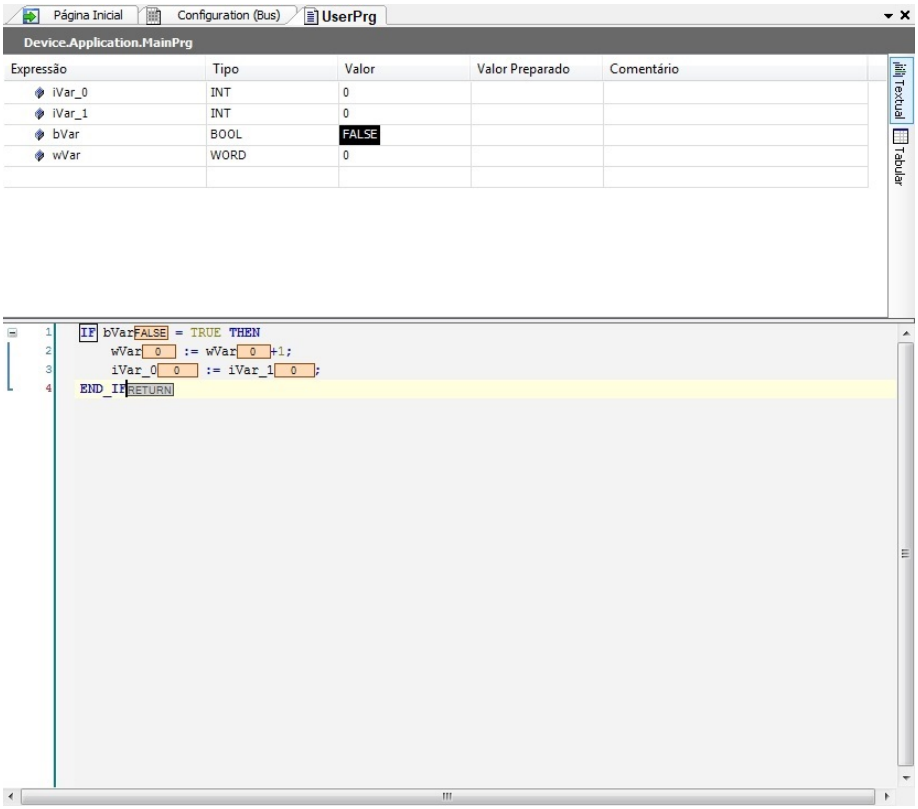


Figura 30: Programa em Execução

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

4.9. Modo Stop

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

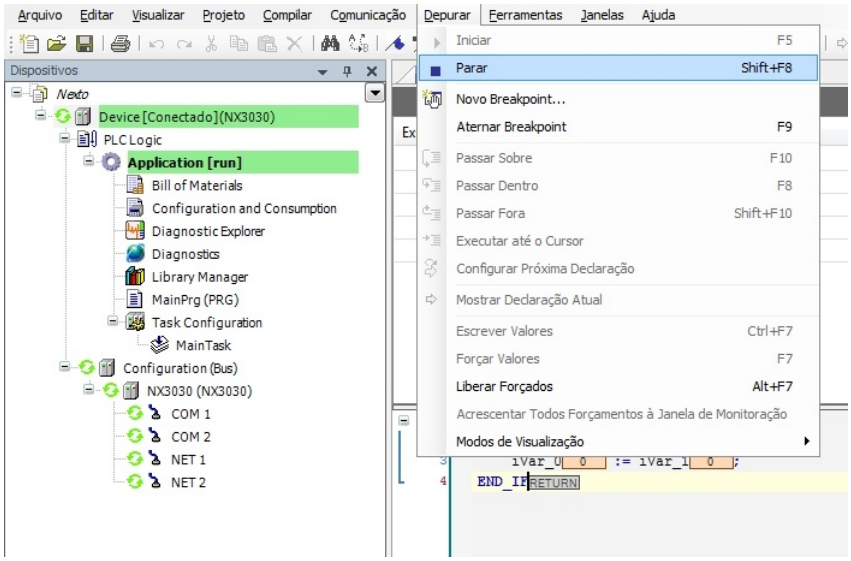


Figura 31: Parando a Aplicação

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.

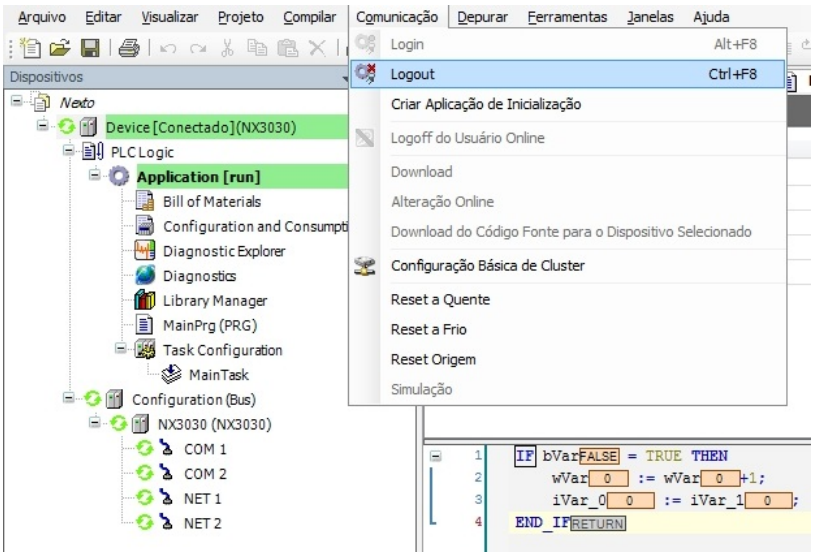


Figura 32: Encerrando a comunicação online com a UCP

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.

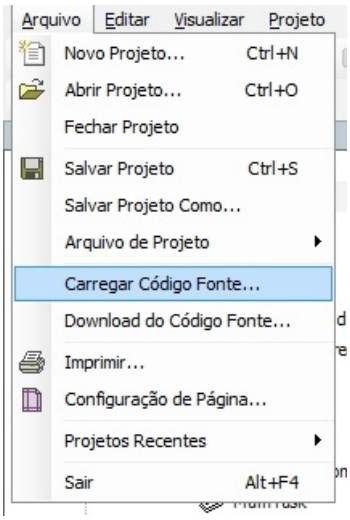


Figura 33: Opção de Upload de Projeto

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

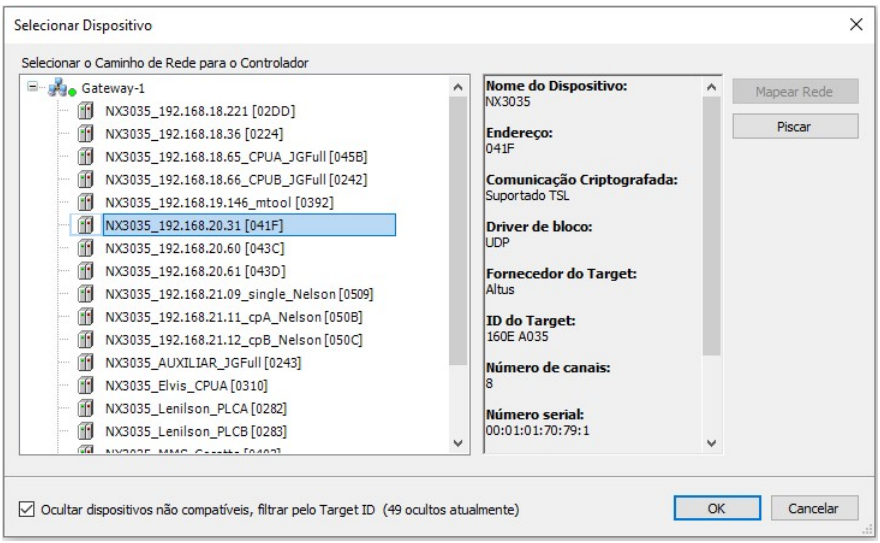


Figura 34: Selecionando a UCP

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

ATENÇÃO

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

ATENÇÃO

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

4.13. Estados de Operação da UCP

4.13.1. Run

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

4.13.2. Stop

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

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

4.13.3. Breakpoint

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

Para maiores informações sobre a utilização de marcas de depuração, favor consultar o Manual de Utilização do Mastertool.

4.13.4. Exception

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

4.13.5. Reset a Quente

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

4.13.6. Reset a Frio

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

4.13.7. Reset Origem

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

Notas:

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

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

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

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

4.14.1. Programa MainPrg

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

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

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

PROGRAM MainPrg
VAR
   isFirstCycle : BOOL := TRUE;
END_VAR

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

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

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

4.14.2. Programa StartPrg

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

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

4.14.3. Programa UserPrg

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

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

4.14.4. GVL System_Diagnostics

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

ATENÇÃO

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

Alguns dispositivos, como o driver de comunicação MODBUS de mapeamento simbólico, não têm seus diagnósticos alocados em variáveis %Q com a diretiva AT. O mesmo ocorre com drivers de comunicação mais recentes.

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

Device.Application.System_Diagnostics

Expression	Type	Value	Address	Con
DG_IEC_60870_5_104_Server	T_DIAG_IEC104_SERVER_1			DG_I
DG_MODBUS_Symbol_Client	T_DIAG_MODBUS_ETH_CLIENT_1			DG_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 35: 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.

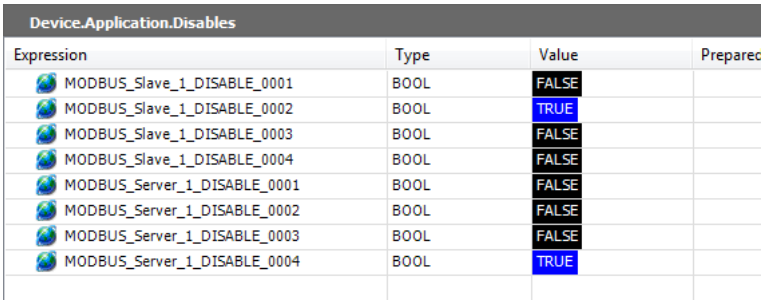


Figura 36: GVL Disables em Modo Online

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 37: GVL Module_Diagnostics em Modo Online

4.14.8. GVL ReqDiagnostics

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

Declaração de variável de diagnóstico de requisição:

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

Onde:

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

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

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

> **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 38: GVL ReqDiagnostics em Modo Online

5. Configuração

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

5.1. Device

5.1.1. Gerênciamento de Usuários e Direitos de Acesso

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

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

5.1.2. Configurações do CP

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

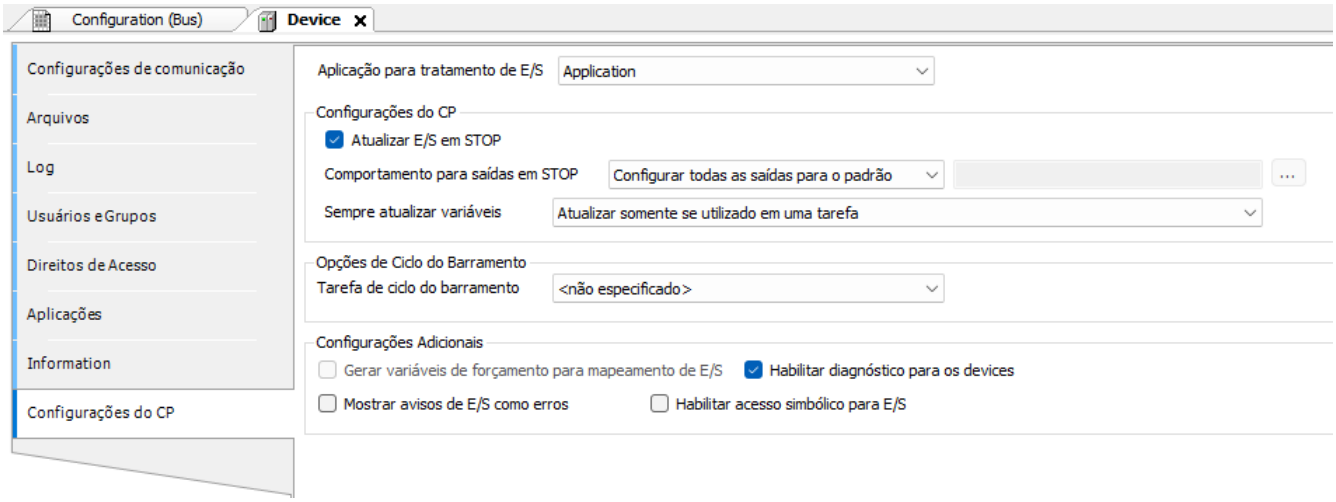


Figura 39: Configurações do CP

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

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

Tabela 41: Configurações do CP

ATENÇÃO

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

5.2. Configuração da UCP

5.2.1. Parâmetros Gerais

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

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

Configuração	Descrição	Padrão	Opções
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
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
Endereço Inicial em %Q	**Área de Diagnósticos (%Q)**		
	Endereço inicial dos diagnósticos da UCP (%Q)	Alocado automaticamente na criação do projeto.	0 a 393217
Tamanho	Tamanho da área de diagnósticos em bytes	Variável, depende do projeto.	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)	0	0 a 393215
Tamanho	Tamanho da memória de dados retentivos em bytes	0	0 a 393216
Endereço Inicial em %Q	**Área Persistente (%Q)**		
	Endereço inicial da memória de dados persistentes (%Q)	0	0 a 393215

Configuração	Descrição	Padrão	Opções
Tamanho	Tamanho da memória de dados persistentes em bytes	0	0 a 393216

Tabela 42: Configurações da UCP

ATENÇÃO

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

ATENÇÃO

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

5.2.1.1. Troca a Quente

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

CUIDADO

As UCPs da Série Nexto não garantem a retenção de variáveis persistentes e retentivas caso a alimentação seja removida ou a própria UCP seja retirada do backplane 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 43: 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 43) a UCP entra em Modo *Stop* em um tempo de até 2 segundos, quando o LED DG começa a piscar 4x (conforme a Tabela 44). Para que a UCP volte ao estado normal *Run*, além de desfazer o que causou a situação anormal é necessário executar um *Reset a Quente* ou um *Reset a Frio*. Caso seja realizado um *Reset Origem*, será necessário realizar o download para que a UCP possa voltar ao estado normal *Run*. Os comandos de *Reset a Quente*, *Reset a Frio* e *Reset Origem* podem ser feitos pelo Mastertool no menu *Comunicação*.

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

5.2.1.1.2. Troca a Quente Desabilitada

Esta configuração não permite qualquer situação anormal de barramento (conforme a Tabela 43) 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 44). Para que, nesses casos, a UCP volte ao estado normal *Run*, além de desfazer o que causou a situação anormal é necessário executar um *Reset a Quente* ou um *Reset a Frio*. Caso seja realizado um *Reset Origem*, será necessário realizar o download para que a UCP possa voltar ao estado normal *Run*. Os comandos de *Reset a Quente*, *Reset a Frio* e *Reset Origem* podem ser feitos pelo Mastertool no menu *Comunicação*.

5.2.1.1.3. Troca a Quente Desabilitada, sem Consistência na Partida

Permite que o sistema entre em operação mesmo quando algum módulo estiver em uma situação anormal de barramento (conforme Tabela 43). 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 44). Para que, nesses casos, a UCP volte ao estado normal *Run*, é necessário executar um *Reset a Quente* ou um *Reset a Frio*. Caso seja realizado um *Reset Origem*, será necessário realizar o download para que a UCP possa voltar ao estado normal *Run*. Os comandos de *Reset a Quente*, *Reset a Frio* e *Reset Origem* podem ser feitos pelo Mastertool no menu *Comunicação*.

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

É considerada "partida" o intervalo entre a energização da UCP (ou comando de reset ou download de aplicação) até a primeira vez em que a mesma entra em modo *Run*. Esta configuração verifica se ocorreu alguma situação anormal de barramento (conforme a Tabela 43) durante a partida; em caso positivo, a UCP entra em Modo *Stop* e o LED DG começa a piscar 4x (conforme a Tabela 44). Posteriormente, para que a UCP possa ser colocada em modo *Run*, além de corrigir o que ocasionou a situação anormal, é necessário executar um comando de *Reset a Quente* ou um *Reset a Frio*. Caso seja realizado um *Reset Origem*, será necessário realizar o download para que a UCP possa voltar ao estado normal *Run*. Os comandos de *Reset a Quente*, *Reset a Frio* e *Reset Origem* podem ser feitos pelo Mastertool no menu *Comunicação*.

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

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

ATENÇÃO

Nesta configuração, quando ocorrer falta de alimentação (mesmo que temporária), comando *Reset a Quente*, comando *Reset a Frio* ou ter sido realizado o *Download* de uma nova aplicação, e algum módulo estiver em uma situação anormal de barramento; a UCP entrará em Modo *Stop* e o LED DG começa a piscar 4x (conforme a Tabela 44), 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 43) 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 44). Posteriormente, para que a UCP possa ser colocada em modo *Run*, além de corrigir o que ocasionou a situação anormal, é necessário executar um comando de *Reset a Quente* ou um *Reset a Frio*. Caso seja realizado um *Reset Origem*, será necessário realizar o download para que a UCP possa voltar ao estado normal *Run*. Os comandos de *Reset a Quente*, *Reset a Frio* e *Reset Origem* podem ser feitos pelo Mastertool no menu *Comunicação*.

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

Permite que o sistema entre em operação mesmo quando algum módulo estiver em uma situação anormal de barramento (conforme Tabela 43). 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 44, devido a alguma situação anormal de barramento, conforme a Tabela 43; 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 44 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 44: Relação entre Condições e Troca a Quente

Nota:

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

5.2.1.2. Áreas de Memória Retentiva e Persistente

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

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

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

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

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

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

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

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

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

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

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

5.2.2. Sincronização de Tempo

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

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

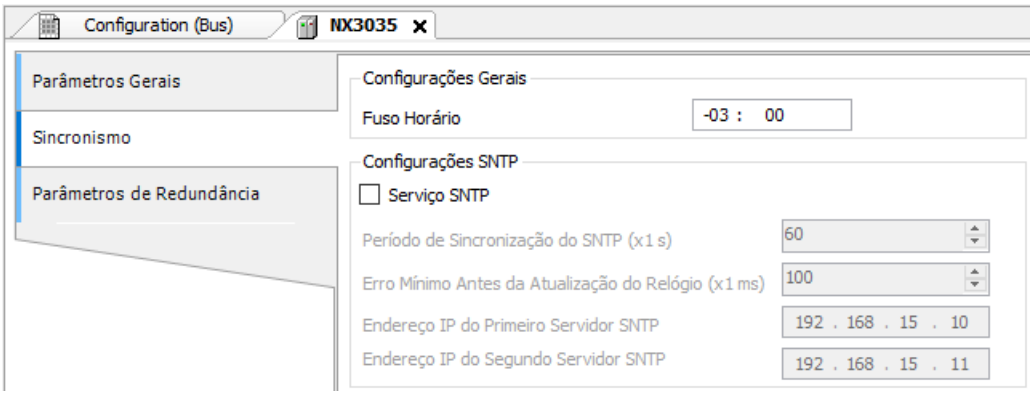


Figura 40: Configuração de SNTP

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

Tabela 45: 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 a mais prioritária. Assim, quando houver mais fontes habilitadas e o servidor SNTP estiver presente, este será responsável pela sincronização do relógio da UCP e qualquer comando de sincronismo de outras fontes será negado.

5.2.2.1. 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.

CUIDADO

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

5.2.2.2. Horário de Verão

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

Para maiores informações, consultar a seção Relógio RTC.

5.3. Configuração da Interface Serial

5.3.1. COM 1

A interface de comunicação COM 1 é composta por um conector tipo DB9 fêmea para o padrão RS-485. Permite a comunicação ponto a ponto ou em rede nos protocolos abertos MODBUS RTU Escravo ou MODBUS RTU Mestre.

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

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

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

Tabela 46: Configuração da Interface Serial Padrão RS-485

Notas:

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

- Um byte para o dado recebido (RX_CHAR : BYTE): Armazena os 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 47: Configurações Específicas

5.3.1.1. Configurações Avançadas

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

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

Tabela 48: Configurações Avançadas da Interface Serial RS-485

5.4. Configuração das Interfaces Ethernet Integradas

A UCP NX3035 possui seis interfaces Ethernet integradas (NET 1 até NET 6). Cada uma destas interfaces possui um conector tipo RJ45 de comunicação no padrão 10BASE-Te/100BASE-TX/1000BASET. Estas interfaces permitem a comunicação usando protocolos abertos, tais como MODBUS TCP Cliente, MODBUS RTU via TCP Cliente, MODBUS TCP

Servidor e MODBUS RTU via TCP Servidor. Estas interfaces também podem ser utilizadas para a programação do CP através do programador Mastertool.

5.4.1. Configurações Básicas das Portas Ethernet

A tela seguinte mostra as configurações básicas das portas Ethernet NET1 a NET6.

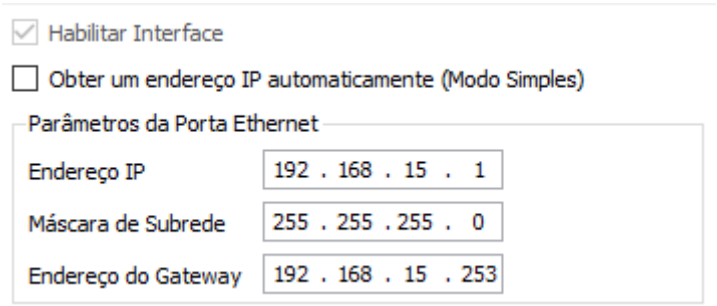


Figura 41: Configurações básicas de portas Ethernet

O checkbox *Habilitar Interface* deve estar marcado para que a porta entre em funcionamento. No caso específico da porta NET1, esta habilitação é fixa, pois trata-se da porta padrão para conexão do programador Mastertool. As demais portas (NET2 ... NET6), por padrão, estão desabilitadas.

A tabela seguinte descreve o preenchimento dos demais campos da tela anterior.

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	NET1: 192.168.15.1 NET2: 192.168.16.1 NET3: 192.168.17.1 NET4: 192.168.18.1 NET5: 192.168.19.1 NET6: 192.168.20.1	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	NET1: 192.168.15.253 NET2: 0.0.0.0 NET3: 0.0.0.0 NET4: 0.0.0.0 NET5: 0.0.0.0 NET6: 0.0.0.0	0.0.0.0 a 223.255.255.254

Tabela 49: Endereçamento IP de portas integradas da UCP NX3035

ATENÇÃO

Não é possível configurar duas portas Ethernet diferentes da UCP NX3035 na mesma subrede, sendo este tipo de configuração bloqueada pela ferramenta Mastertool. Desta forma, cada porta Ethernet da UCP NX3035 deve ser configurada em uma subrede distinta.
Num CP redundante (ver capítulo Redundância com UCP NX3035), o parâmetro *Endereço IP* é substituído por três endereços IP diferentes, que devem estar na mesma subrede Ethernet. Estes três endereços IP são descritos na seção Configuração de Endereços IP nas portas integradas NET1 ... NET6 de uma UCP NX3035 em um CP Redundante.

5.4.2. Configuração de Modo das Portas Ethernet

A tela seguinte mostra as configurações de modo das portas Ethernet NET1 a NET6.

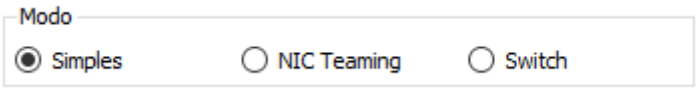


Figura 42: Configuração de modo de portas Ethernet

É possível selecionar um dos três seguintes modos para uma porta Ethernet:

- O modo *Simples* indica que a porta Ethernet opera de forma individual (sem redundância).
- O modo *NIC Teaming* indica que esta porta Ethernet, e a seguinte, formam um par com redundância NIC Teaming. Para maiores detalhes, ver seção Modo NIC Teaming na UCP NX3035.
- O modo *Switch* é uma previsão futura, e ainda não deve ser utilizado.

5.4.2.1. Modo Simples

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

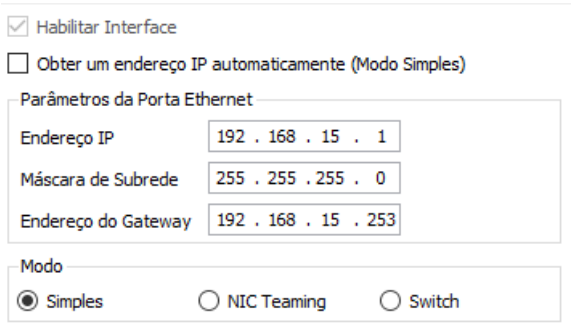


Figura 43: Configurações Avançadas das Interfaces Ethernet - Modo Simples

5.4.2.2. Modo NIC Teaming na UCP NX3035

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

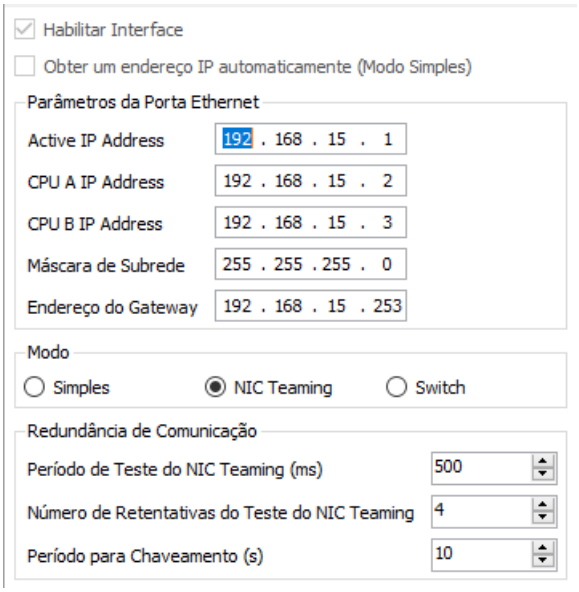


Figura 44: Configurações Avançadas das Interfaces Ethernet - Modo NIC Teaming

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

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

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

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

Uma redundância NIC Teaming entre duas portas Ethernet da UCP NX3035 possui as seguintes características:

- A redundância é formada por um par de portas da UCP, que pode ser:
  - NET1 + NET2
  - NET3 + NET4
  - NET5 + NET6
- Num determinado instante, uma das portas do par estará em estado ativo, e a outra em estado reserva.
- As comunicações com outros equipamentos via Ethernet utilizando protocolos TCP/IP ou UDP/IP ocorrerão através da porta ativa.
- Sobre o endereçamento IP de um par de portas NIC Teaming:
  - A porta em estado reserva não possui nenhum endereço IP.
  - A porta em estado ativo pode ter um ou dois endereços IP. Somente terá dois endereços IP caso a UCP NX3035 esteja instalada em um CP redundante, e caso o half-cluster com este NX3035 esteja momentaneamente em estado Ativo (ver o capítulo Redundância com UCP NX3035 e a seção Configuração de Endereços IP nas portas integradas NET1 ... NET6 de uma UCP NX3035 em um CP Redundante).
  - Desta forma, um cliente como um SCADA ou Mastertool, conectado a um servidor no CP, não precisa se preocupar em trocar o endereço IP caso haja falha em alguma das portas do par NIC Teaming.
- A porta ativa poderá passar para reserva, e a reserva para ativa, nas seguintes situações:
  - Perda de conexão da porta ativa (link-down).

  - Verificação de inatividade de comunicação nas conexões TCP/IP da porta ativa, em conjunto com a impossibilidade de troca de mensagens de teste entre as portas ativa e reserva.
- As duas portas que formam um par NIC Teaming possuem diagnósticos separados, para a indicação de falhas individuais das mesmas.
- A configuração dos parâmetros da porta, e a adição de dispositivos tais como MODBUS, apenas será possível na primeira porta do par (NET1, NET3 ou NET5). A segunda porta (NET2, NET4 ou NET6) terá seus parâmetros de configuração e inserção de dispositivos bloqueados, pois compartilha estes parâmetros e dispositivos com a primeira porta do par NIC Teaming.

Arquiteturas com NIC Teaming visam alta disponibilidade. Para tanto, é importante que as duas portas do par sejam conectadas em dois switches diferentes, para que a falha de um único switch não comprometa a comunicação. Estes dois switches devem estar interligados de alguma forma, para que as duas portas do par NIC Teaming possam trocar mensagens de teste entre si. Estas mensagens de teste periódicas são utilizadas para controlar a troca entre estados das portas (ativa e reserva) em determinadas situações.

5.4.2.3. Modo Switch

Neste modo, a interface forma um par com a interface subsequente operando como um switch Ethernet, permitindo então a comunicação por ambas às portas. Desta forma, este modo permite a ligação em "cascata" de diversas UCPs, possibilitando a implementação de uma topologia de rede em anel. São duas as opções de configuração de proteção de loop no modo switch.

5.4.2.3.1. Modo Switch - Disabled

Nesse modo, as NETs 2 e 3 da UCP atuam como um switch simples, sem qualquer proteção de loop.

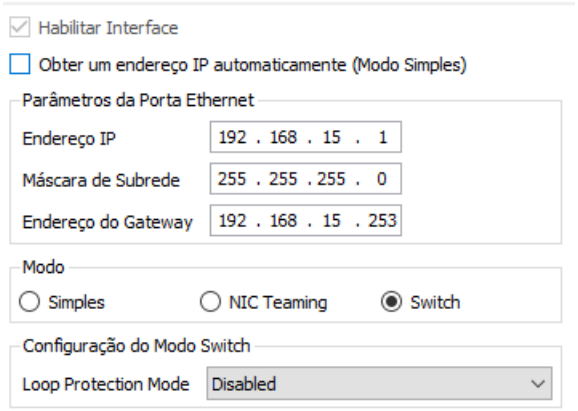


Figura 45: Configurações Avançadas das Interfaces Ethernet - Modo Switch - Loop Protection Disabled

5.4.2.3.2. Modo Switch - MRP

Para informações detalhadas sobre a configuração e o funcionamento do MRP consulte a seção *Configure MRP Ring with ALTUS Controller* do Nexto Series PROFINET Manual - MU214621.

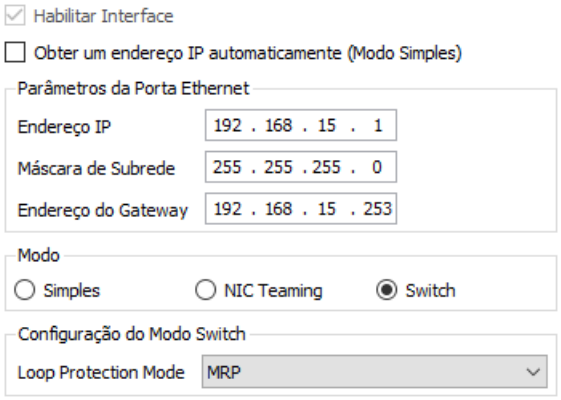


Figura 46: Configurações Avançadas das Interfaces Ethernet - Modo Switch - MRP

5.4.2.3.3. Modo Switch - RSTP

Neste modo, o par de interfaces de rede utiliza o *Rapid Spanning-Tree protocol* (RSTP) para tratar os loops de rede e a redundância de comunicação entre os dispositivos. Quando selecionado, o dispositivo força o protocolo RSTP na bridge. Contudo, se for recebido um frame STP pelas suas portas, o protocolo de *Spanning-Tree* é automaticamente rebaixado para *Spanning-Tree Protocol* (STP), mantendo assim a comunicação com a rede.

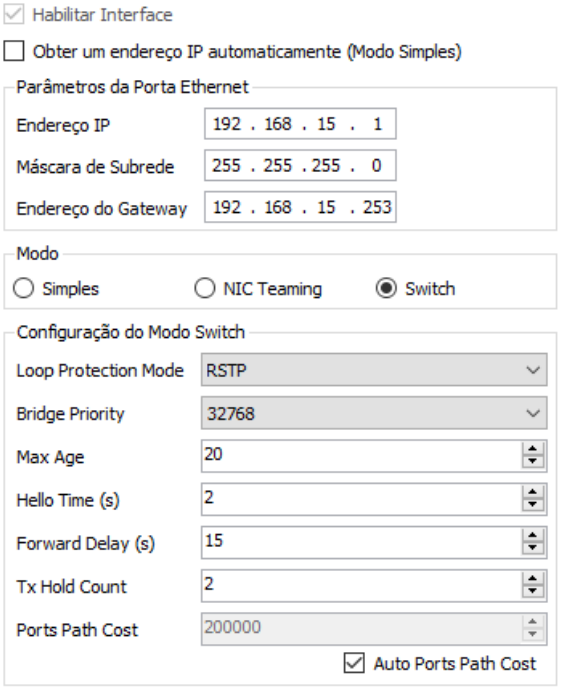


Figura 47: Configurações Avançadas das Interfaces Ethernet - Modo Switch - Loop Protection RSTP

Quando o *Loop Protection Mode* for definido como *RSTP*, outros parâmetros serão habilitados automaticamente na mesma tela e deverão ser configurados:

- **Bridge Priority:** Define a prioridade da bridge. Quanto menor o valor, maior é a prioridade. O valor padrão é 32768.
- **Max Age:** O valor máximo referente a idade de um BDPU. Quando operado no modo RSTP, esse valor trata-se do número máximo de bridges entre o Root e a Bridge que recebeu o BPDU. Porém, quando operado no modo STP, trata-se do tempo máximo para que possa considerar um frame BPDU válido. Pode ser configurado com valores entre 6 e 40, padrão 20. O valor deve seguir a seguinte regra: Max Age $\leq$ 2×(Forward Delay - 1).
- **Hello Time (s):** Período de envio dos frames BPDU's. Pode ser configurado com valores entre 1 e 10, padrão 2.

- **Forward Delay (s):** Tempo que a *Root Port* e a *Designated Port* levam para transitar para o estado de *Forwarding* quando estiver operando no modo STP. Pode ser configurado com valores entre 4 e 30, padrão 15.
- **Tx Hold Count:** Limite máximo de frames BPDUs que a bridge pode enviar a cada hello-time. Pode ser configurado com valores entre 1 e 10, padrão 6.
- **Ports Path Cost:** Define o custo das portas para serem utilizadas no caminho da rede RSTP/STP. Pode ser configurado com valores entre 1 e 20000000, padrão 200000.
- **Auto Ports Path Cost:** Quando habilitado, o custo das portas é definido automaticamente de acordo com a velocidade de conexão delas, o que corresponde a (2×10^{12})/(link speed). Nesta configuração as portas podem assumir custos diferentes. Quando desabilitado, ambas as portas vão assumir o mesmo custo, definido no parâmetro anterior.

5.4.3. Configuração de Modo de Falha das Portas Ethernet

A tela seguinte somente aparece no projeto de um CP redundante (ver capítulo Redundância com UCP NX3035).

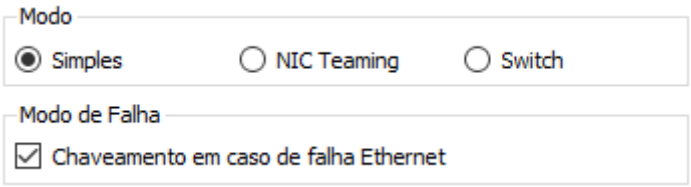


Figura 48: Configuração de modo de falha de portas Ethernet

5.4.4. Portas TCP/UDP Reservadas

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

Serviço	TCP	UDP
Página Web de Sistema	80	-
SNTP	-	123
SNMP	-	161
MODBUS TCP	502*	-
Mastertool	1217*	1740:1743
SQL Server	1433	-
MQTT	1883* / 8883*	-
EtherNet/IP	44818	2222
IEC 60870-5-104	2404*	-
IEC 61850	102*	-
DNP3	20000* / 20005*	-
OPC UA	4840	-
WEBVISU	8080	-
CODESYS ARTI	11740	-
PROFINET	-	34964
Portainer Docker	9000	-
SysLog	-	514
LibHART	1234	-

Tabela 50: Portas TCP/UDP reservadas

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

5.4.5. Roteamento de Rede

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

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

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

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

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

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

5.5. Configuração de Interfaces Ethernet com Módulos NX5000

Além das seis interfaces Ethernet integradas da UCP NX3035 (NET1 ... NET6), é possível inserir até seis módulos NX5000 no mesmo bastidor da UCP NX3035, em posições do bastidor à direita da UCP NX3035. Isso pode ser necessário caso as seis portas integradas da UCP NX3035 não sejam suficientes.

Portas Ethernet de módulos NX5000 podem ser usadas de forma individual, ou organizadas em pares NIC Teaming. Pares NIC Teaming são utilizados para prover interfaces Ethernet redundantes, e demandam a instalação de dois módulos NX5000 em posições vizinhas do bastidor. Desta forma, pode-se adicionar até seis portas Ethernet simples, ou três portas Ethernet redundantes com NIC Teaming, ou outras combinações que não ultrapassem seis módulos NX5000 (ex: duas portas redundantes com NIC Teaming e duas portas simples).

5.5.1. Configurações Básicas da Porta Ethernet do NX5000

A tela seguinte mostra as configurações básicas da porta Ethernet NET1 do NX5000.

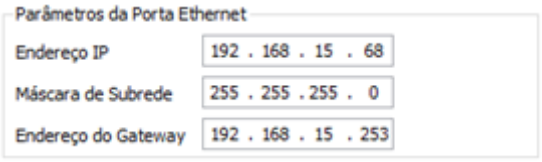


Figura 49: Configurações básicas da porta Ethernet do NX5000

5.5.1.1. Endereçamento IP de Portas Ethernet Remotas do NX5000 (NET1)

5.5.1.1.1. NET 1

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.

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 51: Configurações NET 1 Remota NX5000

Num CP redundante (ver capítulo Redundância com UCP NX3035), o parâmetro *Endereço IP* pode ser substituído por diversos endereços IP diferentes, que devem estar na mesma subrede Ethernet. A quantidade de endereços IP configurados depende do método de endereçamento de IP do cluster, cuja seleção é visualizada apenas em projetos de CPs redundantes, e mostrada na figura seguinte.

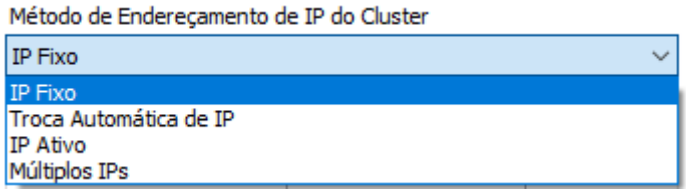


Figura 50: Método de endereçamento IP do NX5000 em CPs redundantes

Estes endereços IP, e os quatro métodos de endereçamento, são descritos na seção Configuração de Endereços IP em Módulos NX5000 em um CP Redundante.

5.5.2. Configuração de Modo da Porta Ethernet do NX5000

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

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

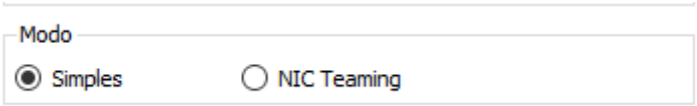


Figura 51: Configuração de modo da porta Ethernet do NX5000

É possível selecionar um dos dois seguintes modos para uma porta Ethernet:

- O modo *Simples* indica que o NX5000 opera de forma individual (sem redundância).
- O modo *NIC Teaming* indica que este NX5000, e o NX5000 à sua direita no bastidor, formam um par com redundância NIC Teaming. Para maiores detalhes, ver seção Modo NIC Teaming no NX5000.

5.5.2.1. Modo NIC Teaming no NX5000

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:

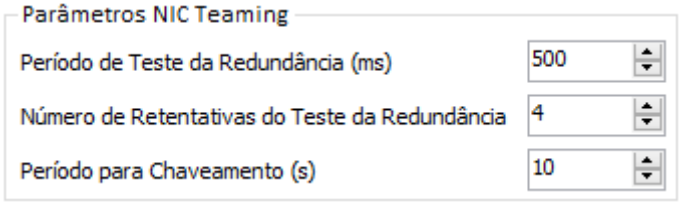


Figura 52: Configuração de parâmetros adicionais para NIC Teaming no NX5000

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

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

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

5.5.3. Configuração de Modo de Falha das Portas Ethernet

A tela seguinte somente aparece no projeto de um CP redundante.



Figura 53: Configuração de modo de falha de portas Ethernet NX5000

5.5.4. Outras Configurações do Módulo NX5000

Ao fazer um duplo clique sobre um módulo NX5000, abrem-se algumas abas com configurações adicionais.

5.5.4.1. Aba "Parâmetros do Módulo"

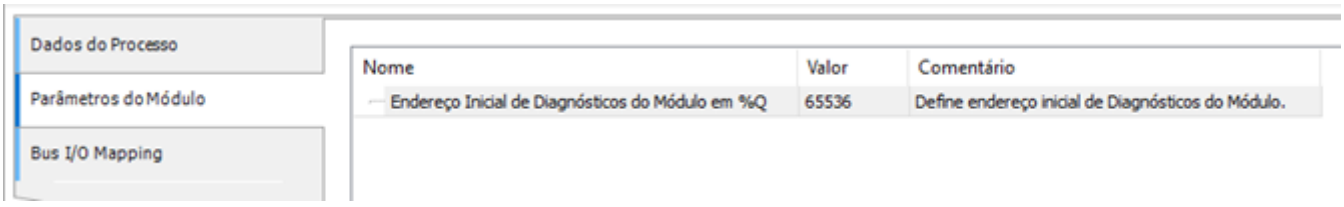


Figura 54: Aba *Parâmetros do Módulo* do NX5000

O *Endereço Inicial de Diagnósticos do Módulo em %Q* define o endereço %Q inicial onde serão armazenados os diagnósticos deste NX5000. Este endereço normalmente é ajustado de forma satisfatória pelo Mastertool, mas o usuário pode alterá-lo se desejar.

5.5.4.2. Aba “Bus I/O Mapping”

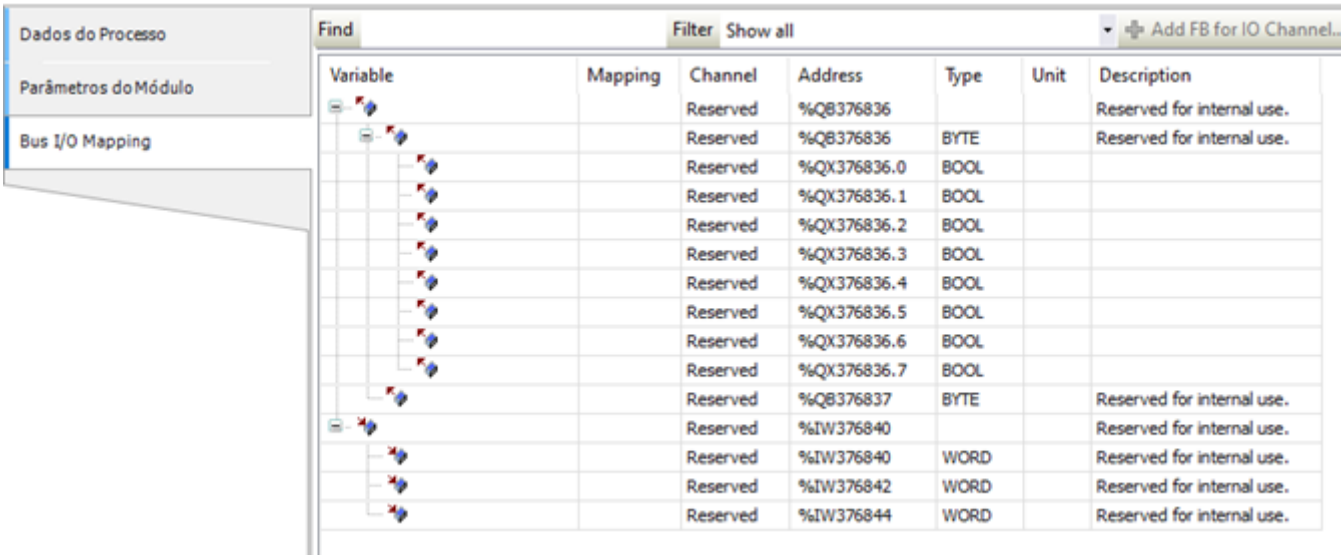


Figura 55: Aba *Bus I/O Mapping* do NX5000

Esta aba aloca 2 bytes de %Q para comandos do usuário, além de 6 bytes de %I para diagnósticos rápidos do NX5000. Estes endereços normalmente são ajustados de forma satisfatória pelo Mastertool, mas o usuário pode alterá-los se desejar.

5.6. Configuração de Redes PROFIBUS com Módulos NX5001

A UCP NX3035 pode ser conectada em redes PROFIBUS DP através do módulo mestre PROFIBUS NX5001.

É possível instalar até seis módulos NX5001 no mesmo bastidor da UCP NX3035. As redes PROFIBUS conectadas podem ser simples ou redundantes. Para conexão a uma PROFIBUS simples, é necessário um único NX5001. Para conexão a uma rede PROFIBUS redundante, são necessários dois NX5001 posicionados lado a lado no bastidor.

Desta maneira, pode-se ter até seis redes PROFIBUS simples, ou até três redes PROFIBUS redundantes. Outras possibilidades também existem desde que não se ultrapassem o limite de seis módulos NX5001 (por exemplo, duas redes PROFIBUS redundantes e duas redes PROFIBUS simples). Caso a rede PROFIBUS “n” (sendo “n” um número de 1 a 6) seja redundante, as duas redes que a compõem são nomeadas PROFIBUS “n”A e PROFIBUS “n”B. Caso a rede PROFIBUS “n” seja simples, a rede que a compõe é nomeada PROFIBUS “n”A.

A figura seguinte mostra um exemplo de arquitetura onde a UCP NX3035 se conecta a duas redes PROFIBUS DP, utilizando três módulos NX5001. A primeira rede é redundante (PROFIBUS 1A e PROFIBUS 1B). A segunda rede é simples (PROFIBUS 2A).

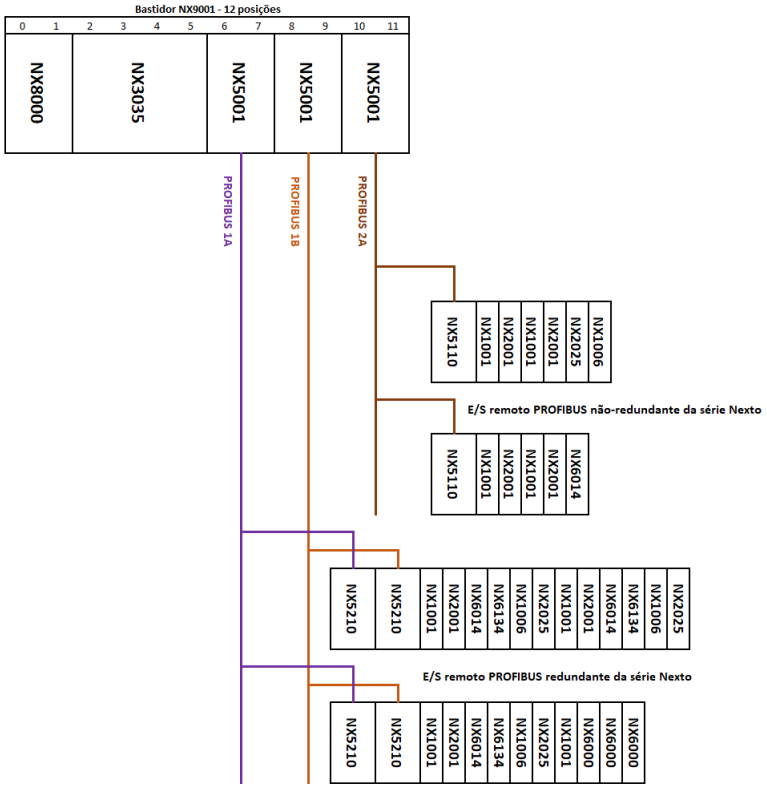


Figura 56: Exemplo de conexão da UCP NX3035 a duas redes PROFIBUS

5.6.1. Inserção ou Remoção de Módulos NX5001

Pode-se inserir ou remover módulos NX5001 no bastidor de um CP com a UCP NX3035. Para executar esta operação de forma correta, deve-se ter consciência das seguintes regras:

- Módulos NX5001 só podem ser instalados no mesmo bastidor da UCP NX3035, numa posição à direita da UCP NX3035.
- O número de módulos NX5001 neste bastidor pode variar entre zero e seis.
- Quando uma rede PROFIBUS é simples, ela necessita de um único módulo NX5001. Quando é redundante, ela necessita de dois módulos NX5001.
- Dois módulos NX5001 utilizados para compor uma rede PROFIBUS redundante devem ocupar posições vizinhas no bastidor.

5.6.2. Ajuste de Parâmetros dos Módulos NX5001

Cada módulo NX5001 utilizado em uma rede PROFIBUS simples, ou cada par de módulos NX5001 utilizados em uma rede PROFIBUS redundante, possui diversos parâmetros que devem ser ajustados. Depois de um duplo clique no módulo NX5001, aparecem diversas abas com configurações que são descritas nas seguintes subseções.

Detalhes avançados sobre a configuração de redes PROFIBUS podem ser obtidos no **Manual de Utilização Mestre PROFIBUS DP NX5001**.

5.6.2.1. Aba "Geral"

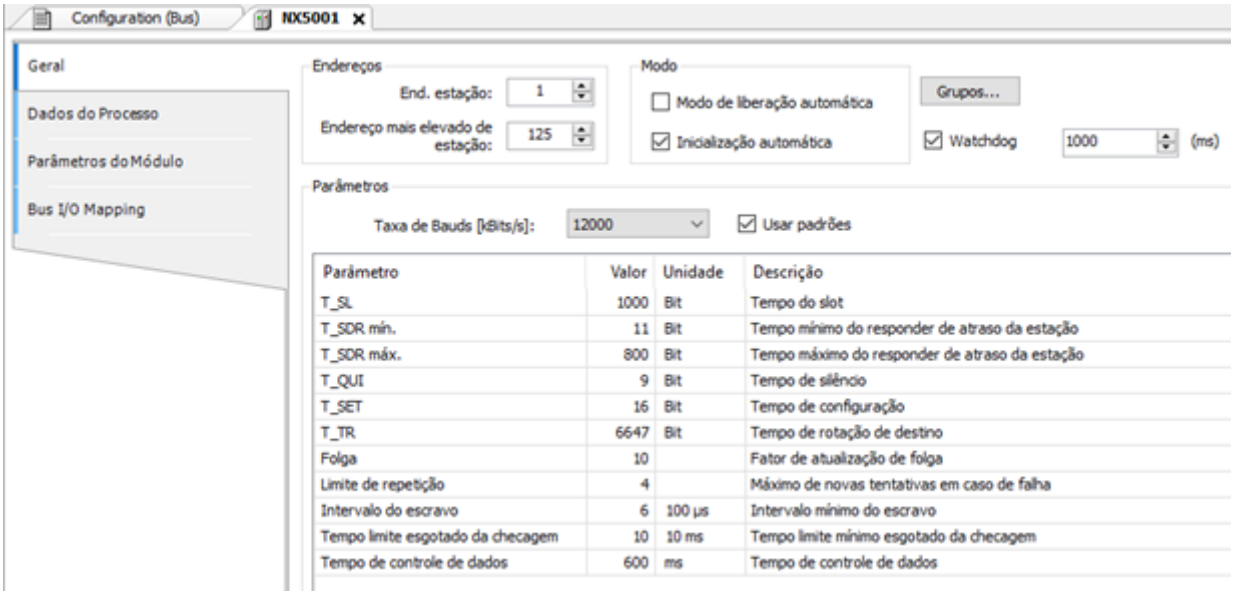


Figura 57: Aba *Geral* do NX5001

O parâmetro *End. estação* indica o endereço do mestre NX5001 na rede PROFIBUS.

Recomenda-se manter o valor 1 sempre que possível.

> **ATENÇÃO**
>
> Em um CP redundante (ver capítulo Redundância com UCP NX3035), o parâmetro *End. estação* corresponde ao endereço do NX5001 instalado no CP Ativo, enquanto este endereço subtraído de 1 corresponde ao endereço do NX5001 instalado no CP Não-Ativo. Portanto, em um CP redundante onde *End. estação* vale 1, dois endereços são alocados para a rede PROFIBUS: o endereço 1 para o NX5001 no CP Ativo, e o endereço 0 para o NX5001 no CP Não-Ativo.

Os demais endereços PROFIBUS normalmente devem ser alocados da seguinte maneira:

- **2:** para um mestre classe 2, por exemplo, utilizado para coletar dados de gerenciamento de ativos dos escravos PROFIBUS via protocolo DP-V1.
- **3 até 125:** para escravos do NX5001 na rede PROFIBUS, tais como estações de E/S remoto PROFIBUS da Série Nexto.
- O endereço 126 é muito utilizado por dispositivos escravos quando saem de fábrica, por isso deve-se evitar utilizá-lo.
- O endereço 127 é utilizado para enviar frames em broadcast, e também não deve ser utilizado.

O parâmetro *Endereço mais elevado de estação* pode ficar em 125, mas para otimizar a performance da rede PROFIBUS ele pode ser igual ao maior endereço de um escravo do NX5001 na rede PROFIBUS.

Para mais detalhes de outros campos, consultar o **Manual de Utilização Mestre PROFIBUS DP NX5001.**

O checkbox *Watchdog* e o tempo de watchdog ao lado do mesmo não fazem efeito sobre as remotas PROFIBUS tipicamente utilizadas (Série Nexto ou Série Ponto). Estas configurações se destinam a outras remotas especiais. Os parâmetros de *Watchdog* para remotas PROFIBUS das séries Nexto e Ponto devem ser configurados nas telas destas remotas (ex: NX5210, NX5110, PO5065).

O parâmetro *Taxa de Bauds [kBits/s]* seleciona a velocidade da rede PROFIBUS, com várias opções entre 9.6 kbps e 12000 kbps. Ao lado desta taxa de bauds, recomenda-se deixar marcado o checkbox *Usar padrões*, para que o Mastertool calcule automaticamente diversos parâmetros mostrados abaixo (*T_SL*, *T_SDR mín.*, etc). Do contrário, o usuário poderá alterar tais parâmetros, o que exige um conhecimento profundo da rede PROFIBUS.

5.6.2.2. Aba "Parâmetros do Módulo"



Figura 58: Aba *Parâmetros do Módulo* do NX5001

O *Endereço Inicial de Diagnósticos do Módulo em %Q* define o endereço %Q inicial onde serão armazenados os diagnósticos deste NX5001. Este endereço normalmente é ajustado de forma satisfatória pelo Mastertool, mas o usuário pode alterá-lo se desejar.

O *Endereço Inicial de Diagnósticos dos Escravos em %Q* define o endereço %Q inicial a partir do qual serão colocados os diagnósticos de cabeças remotas e módulos de E/S de remotas PROFIBUS instaladas abaixo deste NX5001. Este endereço normalmente é ajustado de forma satisfatória pelo Mastertool, mas o usuário pode alterá-lo se desejar.

Para agrupar dois módulos NX5001 em uma rede PROFIBUS redundante, deve-se selecionar um módulo NX5001 desagrupado que tenha outro módulo NX5001 também desagrupado a sua direita no bastidor.

A seguir deve-se ajustar o parâmetro *Redundância de Rede* com o valor *True*. Para desagrupá-los, basta seguir o mesmo procedimento, porém marcando o parâmetro *Redundância de Rede* com o valor *False*. Se este parâmetro estiver marcado como *True*, os parâmetros do módulo NX5001 a sua direita estarão bloqueados para edição, pois serão calculados automaticamente a partir dos parâmetros deste NX5001 à esquerda.

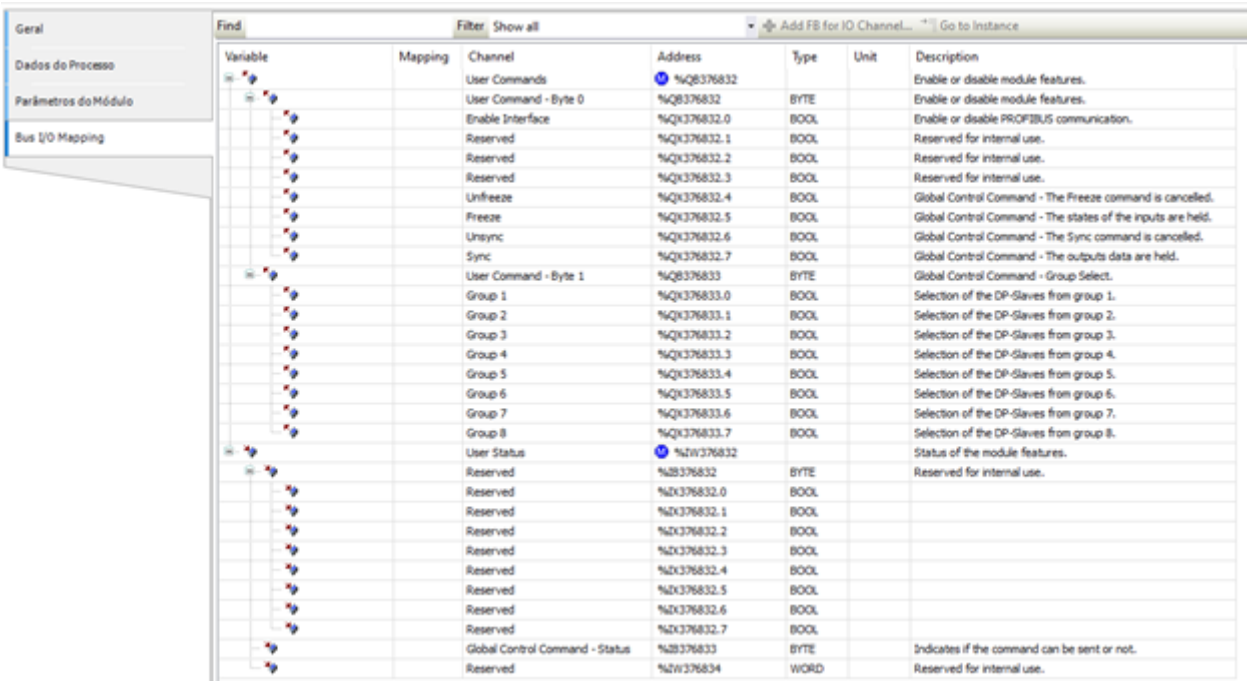
O parâmetro *Modo de Falha* se aplica somente a um CP redundante (ver capítulo Redundância com UCP NX3035), e possui duas opções:

- **True:** uma falha total da rede PROFIBUS causará um switch-over entre CPs Ativo e Reserva. A falha total se caracteriza quando o mestre NX5001 da rede simples, ou os dois mestres NX5001 da rede redundante, não conseguem comunicação com nenhum dos seus escravos.
- **False:** uma falha total da rede PROFIBUS não causará um switch-over entre CPs Ativo e Reserva.

O parâmetro *Alocar Área de Diagnósticos Conforme Descrição do Dispositivo* define a quantidade de bytes de diagnósticos redundantes em %Q que serão alocados para remotas de terceiros (não se aplica a remotas PROFIBUS da Altus):

- Se o valor for *False*, alocará sempre 244 bytes de diagnóstico para remotas de terceiros.
- Se o valor for *True*, alocará a quantidade de bytes definida no arquivo GSD da remota de terceiros através do parâmetro *Max_Diag_Data_Len*.

5.6.2.3. Aba "Bus I/O Mapping"



Variable	Mapping	Channel	Address	Type	Unit	Description
		User Commands	%QB376832			Enable or disable module features.
		User Command - Byte 0	%QB376832	BYTE		Enable or disable module features.
		Enable Interface	%QX376832.0	BOOL		Enable or disable PROFIBUS communication.
		Reserved	%QX376832.1	BOOL		Reserved for internal use.
		Reserved	%QX376832.2	BOOL		Reserved for internal use.
		Reserved	%QX376832.3	BOOL		Reserved for internal use.
		Unfreeze	%QX376832.4	BOOL		Global Control Command - The Freeze command is cancelled.
		Freeze	%QX376832.5	BOOL		Global Control Command - The states of the inputs are held.
		Unsync	%QX376832.6	BOOL		Global Control Command - The Sync command is cancelled.
		Sync	%QX376832.7	BOOL		Global Control Command - The outputs data are held.
		User Command - Byte 1	%QB376833	BYTE		Global Control Command - Group Select.
		Group 1	%QX376833.0	BOOL		Selection of the DP-Slaves from group 1.
		Group 2	%QX376833.1	BOOL		Selection of the DP-Slaves from group 2.
		Group 3	%QX376833.2	BOOL		Selection of the DP-Slaves from group 3.
		Group 4	%QX376833.3	BOOL		Selection of the DP-Slaves from group 4.
		Group 5	%QX376833.4	BOOL		Selection of the DP-Slaves from group 5.
		Group 6	%QX376833.5	BOOL		Selection of the DP-Slaves from group 6.
		Group 7	%QX376833.6	BOOL		Selection of the DP-Slaves from group 7.
		Group 8	%QX376833.7	BOOL		Selection of the DP-Slaves from group 8.
		User Status	%IW376832			Status of the module features.
		Reserved	%IB376832	BYTE		Reserved for internal use.
		Reserved	%IX376832.0	BOOL		
		Reserved	%IX376832.1	BOOL		
		Reserved	%IX376832.2	BOOL		
		Reserved	%IX376832.3	BOOL		
		Reserved	%IX376832.4	BOOL		
		Reserved	%IX376832.5	BOOL		
		Reserved	%IX376832.6	BOOL		
		Reserved	%IX376832.7	BOOL		
		Global Control Command - Status	%IB376833	BYTE		Indicates if the command can be sent or not.
		Reserved	%IW376834	WORD		Reserved for internal use.

Figura 59: Aba *Bus I/O Mapping* do NX5001

Esta aba aloca 2 bytes de %Q para comandos do usuário, além de 4 bytes de %I para diagnósticos rápidos do NX5001. Estes endereços normalmente são ajustados de forma satisfatória pelo Mastertool, mas o usuário pode alterá-los se desejar.

ATENÇÃO

Em *User Commands*, *User Commands – Byte 0*, o usuário não deve modificar o valor do bit 0 (*Enable Interface*), pois este bit é manipulado pela aplicação.

5.6.3. Configurações de Remotas PROFIBUS

Para configurar remotas PROFIBUS abaixo de um mestre NX5001, deve-se consultar, além do **Manual de Utilização Mestre PROFIBUS DP NX5001**, os seguintes manuais:

- Manual de Utilização da Série Ponto
- Manual de Utilização Cabeça PROFIBUS PO5064 e Cabeça Redundante PROFIBUS PO5065
- Manual de Utilização Rede HART sobre PROFIBUS
- Manual de Utilização da Cabeça PROFIBUS-DP Nexto

5.6.3.1. Remotas Compatíveis com Rede PROFIBUS Redundante

Atualmente, apenas três modelos de cabeças remotas fabricadas pela Altus podem ser conectadas em uma rede PROFIBUS redundante:

- PO5063V5
- PO5065
- NX5210

5.6.3.2. Configuração de Cão-de-Guarda em Remotas PROFIBUS em CPs Redundantes

Em um CP redundante (ver capítulo Redundância com UCP NX3035), deve-se ter um cuidado especial na configuração do parâmetro de cão-de-guarda das remotas. Este cuidado vale para redes PROFIBUS simples e redundantes, visto que é possível utilizar redes PROFIBUS simples em CPs redundantes.

Caso, na tela de configuração da remota, esteja habilitado o checkbox *Controle de Cão-de-Guarda*, o campo *Tempo* deverá ser configurado corretamente. O tempo de cão-de-guarda deve ser maior ou igual do que o maior tempo entre WD1 e WD2:

- WD1 = I x 2 + 500 ms
- WD2 = I x 3

Nas equações anteriores, I é o intervalo configurado para a tarefa MainTask. Por exemplo, para um intervalo de 200 ms para a MainTask, o valor mínimo do cão-de-guarda configurado em cada remota deve ser de 900 ms. Portanto, um cão-de-guarda de 1000 ms é adequado.

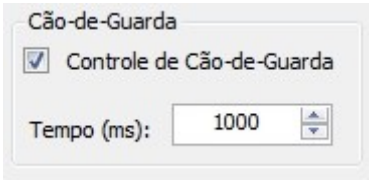


Figura 60: Configuração de Cão-de-Guarda de uma Remota PROFIBUS em um CP redundante

5.7. 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:

	NX3035
Pontos Mapeados	512000
Mapeamentos (Por Instância / Total)	5120 / 20480
Requisições	512
NETs – Instâncias Cliente ou Servidor (Por NET / Total)	4 / 16
COM (n) – Instâncias Mestre ou Escravo	1
Centros de Controle	-

Tabela 52: 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 53 e na Tabela 54, 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 53: 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 52.

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

Limitações	MODBUS RTU Mestre	MODBUS RTU Escravo	MODBUS Ethernet Cliente	MODBUS Ethernet Servidor
Requisições simultâneas por dispositivo	-	-	8	64

Tabela 54: 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.

5.7.1. Comportamento dos Protocolos x Estados da UCP

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

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

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
EtherNet/IP	Scanner	✓	✓	×	✓	NA	×
	Adapter	×	✓	×	✓	NA	×
PROFINET	Controller	×	✓	×	✓	NA	✓
MQTT	Client	×	×	×	✓	✓	×
SparkPlugB	Client	×	×	×	✓	✓	×
OpenVPN	Server	✓	✓	✓	✓	✓	✓
	Client	✓	✓	✓	✓	✓	✓
FTP	Server	✓	✓	✓	✓	✓	✓
Syslog	Client	✓	✓	✓	✓	✓	✓

Tabela 55: 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*".

5.7.2. MODBUS RTU Mestre

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

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

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

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

5.7.2.1. Configuração do Protocolo MODBUS Mestre por Mapeamento Simbólico

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

- Configurar os parâmetros gerais do protocolo MODBUS Mestre, como: tempos de atraso de envio e interframe mínimo como na Figura 61.
- 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 62.
- Adicionar e configurar os mapeamentos MODBUS na aba Mapeamentos conforme Figura 63, 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 64, especificando a função, o tempo de varredura da requisição, o endereço inicial (leitura/escrita), o tamanho dos dados (Leitura/Escrita) e gerar as variáveis de diagnóstico e desabilitação das requisições através dos botões na parte inferior da janela.

5.7.2.1.1. 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:

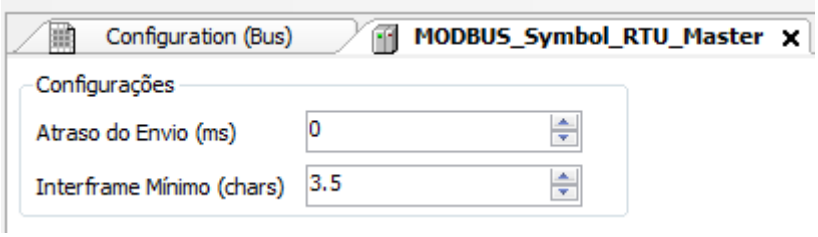


Figura 61: Tela de Configuração Parâmetros Gerais MODBUS RTU Mestre

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 56: 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:		
		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

T_DIAG_MODBUS_RTU_MASTER_1.*	Tamanho	Descrição
eErrorCode	SERIAL_STATUS (BYTE)	20: tamanho do buffer de transmissão inválido 21: método de sinal de modem inválido 22: time-out de CTS = verdadeiro 23: time-out de CTS = falso 24: erro de time-out na transmissão 30: tamanho do buffer de recepção inválido 31: erro de time-out na recepção 32: controle de fluxo configurado diferente de manual 33: controle de fluxo inválido para a porta serial configurada 34: recepção de dados não permitida no modo normal 35: recepção de dados não permitida no modo estendido 36: interrupção DCD não permitida 37: interrupção CTS não permitida 38: interrupção DSR não permitida 39: porta serial não configurada 50: erro interno na porta serial
Bits de comando, reiniciados automaticamente:		
tCommand.bStop	BIT	Parar o mestre.
tCommand.bRestart	BIT	Reiniciar o mestre.
tCommand.bResetCounter	BIT	Reiniciar as estatísticas dos diagnósticos (contadores).
Estatísticas de comunicação:		
tStat.wTXRequests	WORD	Contador de requisições transmitidas pelo mestre (0 a 65535).
tStat.wRXNormalResponses	WORD	Contador de respostas normais recebidas pelo mestre (0 a 65535).
tStat.wRXExceptionResponses	WORD	Contador de respostas com códigos de exceção recebidas pelo mestre (0 a 65535).
tStat.wRXIllegalResponses	WORD	Contador de respostas ilegais recebidas pelo mestre – sintaxe inválida, número insuficiente de bytes recebidos, CRC inválido – (0 a 65535).
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 57: Diagnósticos MODBUS RTU Mestre

Nota:

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

5.7.2.1.2. Configuração dos Dispositivos – Configuração por Mapeamento Simbólico

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

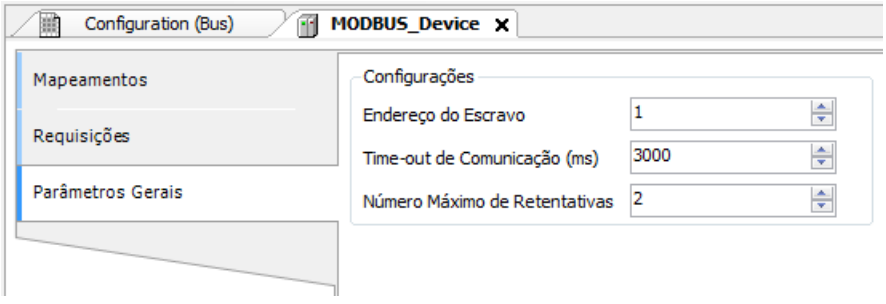


Figura 62: Tela de Configurações dos Parâmetros Gerais do Dispositivo

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 58: Configurações do Dispositivo

Notas:

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

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

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

5.7.2.1.3. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico

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

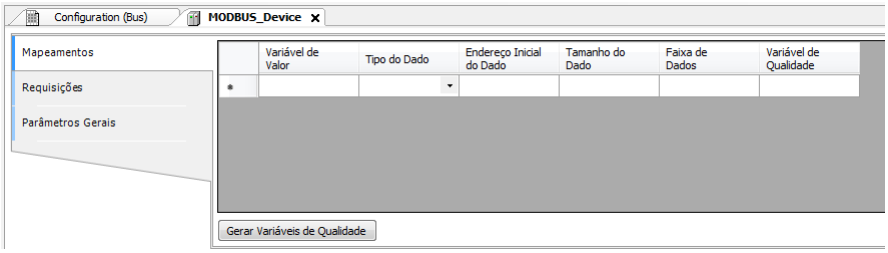


Figura 63: Tela de Mapeamentos de Dados MODBUS

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

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

Tabela 59: 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 60: Tipos de Dados suportados no MODBUS

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

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

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

5.7.2.1.4. Configuração das Requisições – Configuração por Mapeamento Simbólico

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

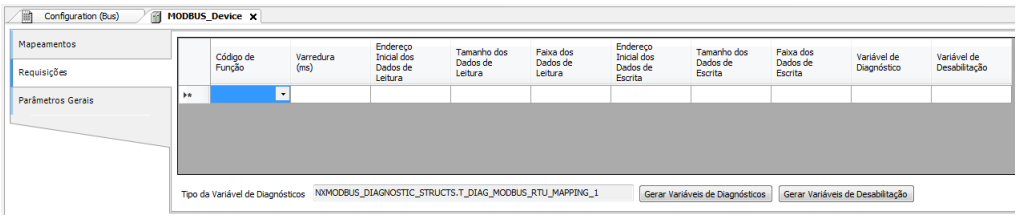


Figura 64: Tela de Requisições de dados MODBUS Mestre

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

Configuração	Descrição	Padrão	Opções
Varredura (ms)	Período de comunicação (ms)	100	0 a 3600000
Endereço Inicial do Dado de Leitura	Endereço inicial dos dados de leitura MODBUS	-	1 a 65536
Tamanho dos Dados de Leitura	Tamanho dos dados de leitura MODBUS	-	Depende da função utilizada
Faixa dos Dados de Leitura	Faixa de endereço dos dados de leitura MODBUS	-	0 a 2147483646
Endereço Inicial do Dado de Escrita	Endereço inicial dos dados de escrita MODBUS	-	1 a 65536
Tamanho dos Dados de Escrita	Tamanho dos dados de escrita MODBUS	-	Depende da função utilizada
Faixa dos Dados de Escrita	Faixa de endereço dos dados de escrita MODBUS	-	0 a 2147483647
Variável de Diagnóstico	Nome da variável de diagnóstico	-	Nome de uma variável declarada em um programa ou GVL
Variável de Desabilitação	Variável utilizada para desabilitar a relação MODBUS	-	Campo destinado a variável simbólica utilizada para desabilitar, individualmente, as requisições MODBUS configuradas. Esta variável deve ser do tipo BOOL. A variável pode ser simples ou elemento de array e pode estar em estruturas.

Tabela 61: 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 62: 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 63.

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 86 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 63: 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 64: Códigos de Erro das relações MODBUS

ATENÇÃO

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

5.7.2.2. 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.

5.7.2.2.1. 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:

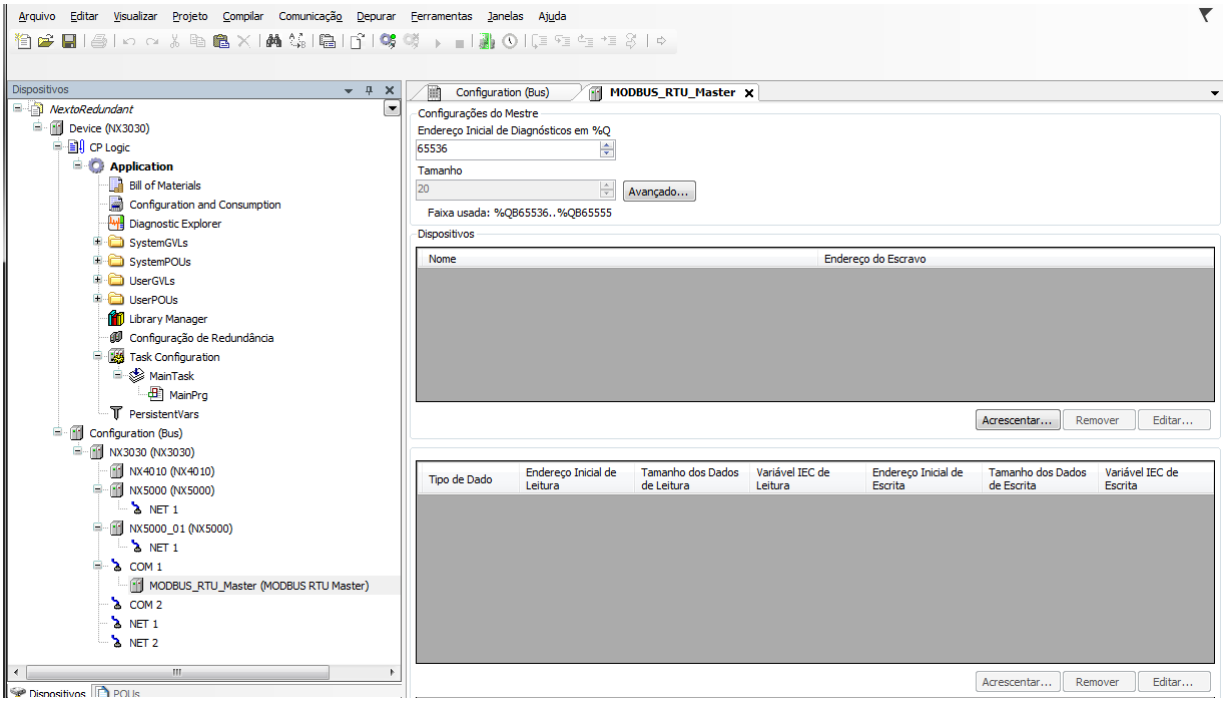


Figura 65: Tela de Configuração MODBUS RTU Mestre

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 65: 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 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 57.

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.

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

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

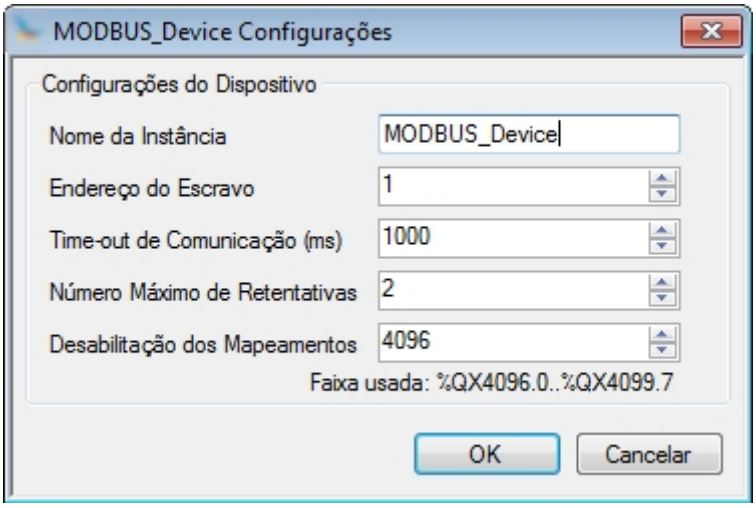


Figura 66: Configurações do Dispositivo

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 66: 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 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.

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

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

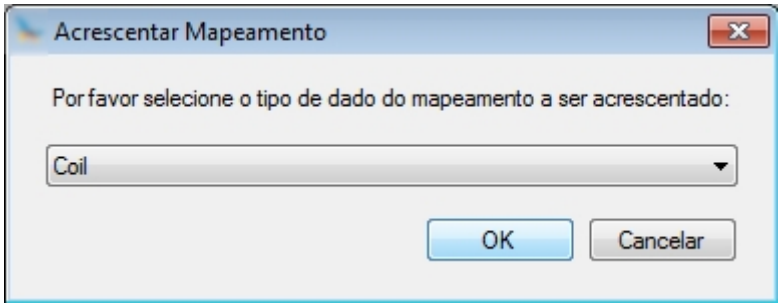


Figura 67: Tipo de Dado MODBUS

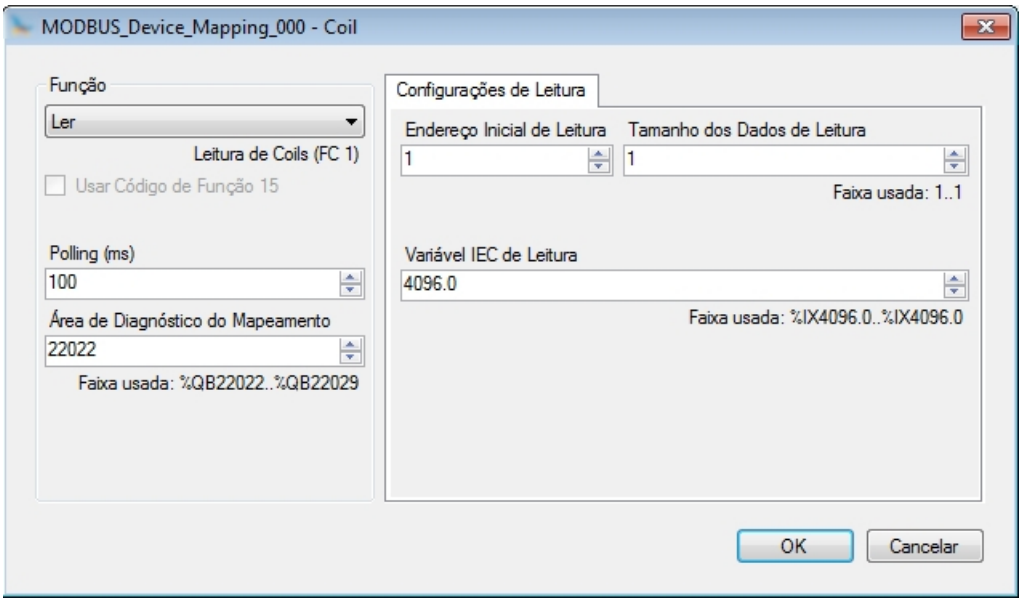


Figura 68: Função MODBUS

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

Configuração	Descrição	Padrão	Opções
Máscara de Escrita das Variáveis IEC	Endereço inicial das variáveis para a máscara de escrita (%Q)	-	0 a 2147483644

Tabela 67: Mapeamentos do Dispositivo

Notas:

Função: Os tipos de dados disponíveis estão detalhados na Tabela 86 e as funções MODBUS (FC) disponíveis na Tabela 84.

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 63.

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 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.

5.7.3. MODBUS RTU Escravo

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

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

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

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

5.7.3.1. Configuração do Protocolo MODBUS Escravo por Mapeamento Simbólico

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

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

5.7.3.1.1. 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.

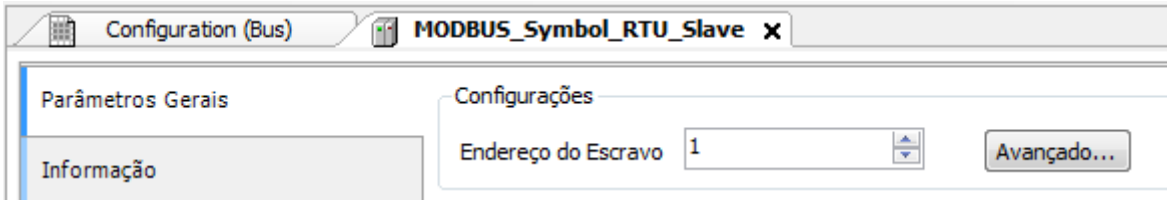


Figura 69: Configuração do Escravo

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

Tabela 68: 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.

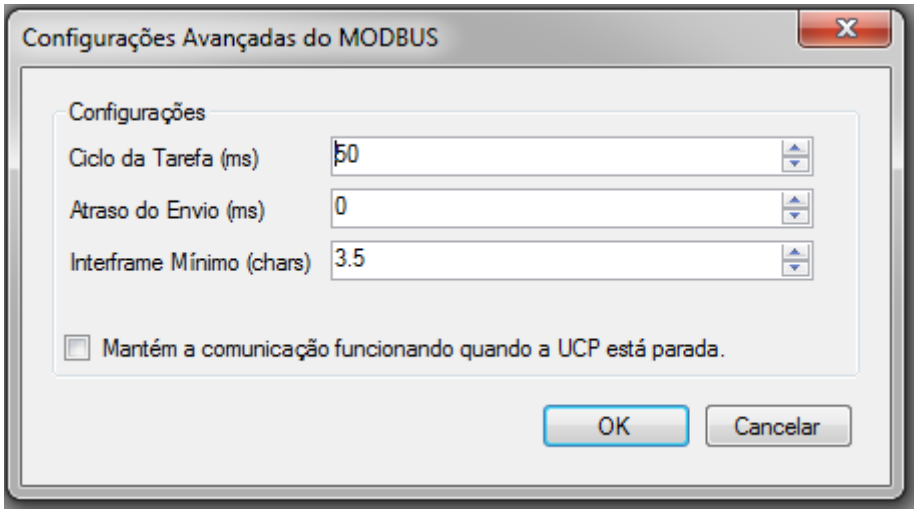


Figura 70: Configurações Avançadas MODBUS do Escravo

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 69: 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.

T_DIAG_MODBUS_RTU_SLAVE_1.*	Tamanho	Descrição
Códigos de erro:		
eErrorCode	SERIAL_STATUS (BYTE)	0: não existem erros 1: porta serial inválida 2: modo da porta serial inválido 3: taxa de transmissão inválida 4: bits de dado inválidos 5: bits de paridade inválidos 6: bits de parada inválidos 7: parâmetro de sinal de modem inválido 8: parâmetro de Threshold de RX da UART inválido 9: parâmetro de time-out inválido 10: porta serial ocupada 11: erro de hardware na UART 12: erro de hardware remoto 20: tamanho do buffer de transmissão inválido 21: método de sinal de modem inválido 22: time-out de CTS = verdadeiro 23: time-out de CTS = falso 24: erro de time-out na transmissão 30: tamanho do buffer de recepção inválido 31: erro de time-out na recepção 32: controle de fluxo configurado diferente de manual 33: controle de fluxo inválido para a porta serial configurada 34: recepção de dados não permitida no modo normal 35: recepção de dados não permitida no modo estendido 36: interrupção DCD não permitida 37: interrupção CTS não permitida 38: interrupção DSR não permitida 39: porta serial não configurada 50: erro interno na porta serial
Bits de comando, reiniciados automaticamente:		
tCommand.bStop	BIT	Parar o escravo.
tCommand.bRestart	BIT	Reiniciar o escravo.
tCommand.bResetCounter	BIT	Reiniciar as estatísticas dos diagnósticos (contadores).
Estatísticas de comunicação:		
tStat.wRXRequests	WORD	Contador de requisições normais recebidas pelo escravo e respondidas normalmente. No caso de um comando broadcast, esse contador é incrementado, porém não é transmitida a resposta (0 a 65535).
tStat.wTXExceptionResponses	WORD	Contador de requisições normais recebidas pelo escravo e respondidas com códigos de exceção. No caso de um comando broadcast, esse contador é incrementado, porém não é transmitida a resposta (0 a 65535).
tStat.wRXFrames	WORD	Contador de frames recebidos pelo escravo. Considera-se frame, algo que é processado e seguido de um período mínimo de silêncio interframes (Interframe Mínimo), ou seja, uma mensagem ilegal também é computada (0 a 65535).
tStat.wRXIllegalRequests	WORD	Contador de requisições ilegais. Estes são frames que iniciam com o endereço 0 (broadcast) ou com o endereço MODBUS do escravo, mas não são requisições legais – sintaxe inválida, frames menores, CRC inválido – (0 a 65535).
tStat.wRXOverrunErrors	WORD	Contador de frames com erros de overrun durante a recepção – UART FIFO ou fila RX – (0 a 65535).
tStat.wRXIncompleteFrames	WORD	Contador de frames com erro de construção, paridade ou falha durante a recepção (0 a 65535).
tStat.wCTSTimeOutErrors	WORD	Contador de erro de time-out no CTS, utilizando o handshake RTS/CTS, durante a transmissão (0 a 65535).

Tabela 70: Diagnósticos MODBUS RTU Escravo

Nota:

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

5.7.3.1.2. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico

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

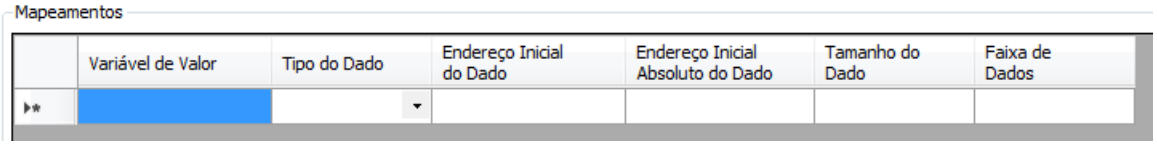


Figura 71: Tela de Mapeamentos de dados MODBUS

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 71: Configuração dos Mapeamentos MODBUS

Notas:

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

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

Tipo de Dado	Tamanho [bits]	Descrição
Coil	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 72: Tipos de Dados MODBUS suportados pelas UCPs Nexto

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

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

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

ATENÇÃO

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

5.7.3.2. 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.

5.7.3.2.1. 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:

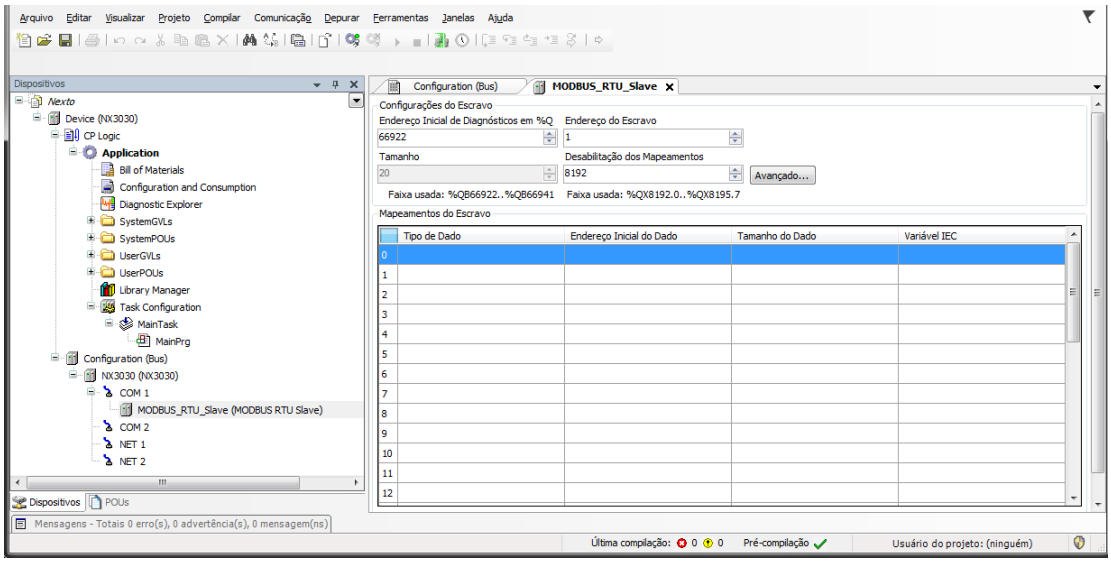


Figura 72: Tela de Configuração MODBUS RTU Escravo por Representação Direta

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 73: 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 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 70.

Os tempos de comunicação do protocolo MODBUS Escravo, encontrados no botão *Avançado...* da tela de configuração, estão descritos na Tabela 69.

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

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

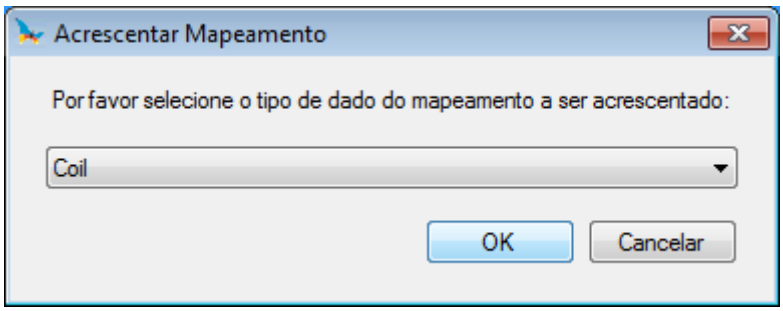


Figura 73: Adicionando Relações MODBUS

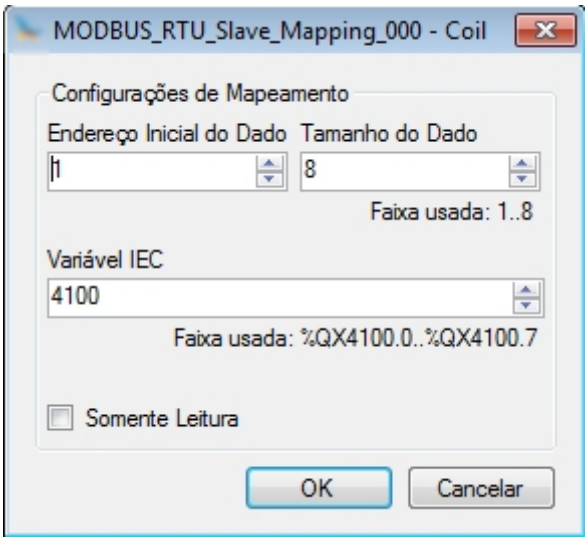


Figura 74: Configurando a Relação MODBUS

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 74: 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 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.

5.7.4. 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 52.

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

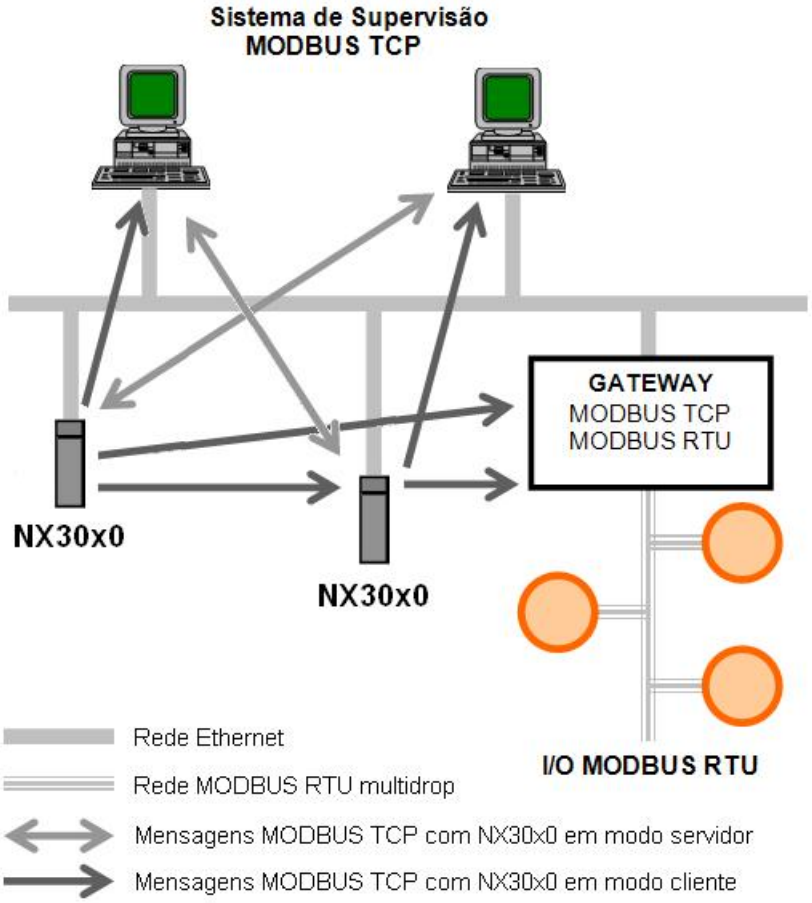


Figura 75: Rede de Comunicação MODBUS TCP

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

Todas as relações, em modo cliente ou servidor, podem ser desabilitadas através de variáveis de representação direta (%Q) identificadas como Desabilitação dos Mapeamentos pelo Mastertool. A desabilitação pode ocorrer através de bits gerais, os

quais afetam todas as relações de um modo de operação, ou através de bits específicos, afetando relações específicas.

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

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

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

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

Tabela 75: 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 76: 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 Integradas.

5.7.5. MODBUS Ethernet Cliente

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

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

O procedimento para inserir uma instância de protocolo é encontrado com detalhes no Manual de Utilização do Mastertool ou na seção Inserindo uma Instância de Protocolo.

5.7.5.1. Configuração do Protocolo MODBUS Ethernet Cliente por Mapeamento Simbólico

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

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

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

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

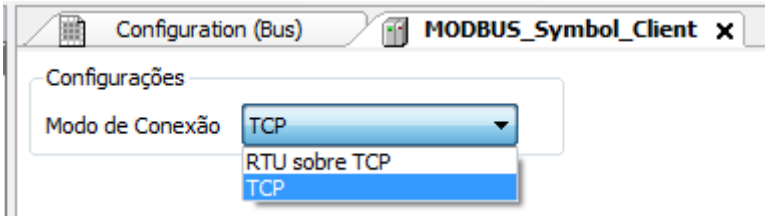


Figura 76: Tela de Configuração Parâmetros Gerais MODBUS Cliente

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

Tabela 77: 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).

T_DIAG_MODBUS_ETH_CLIENT_1.*	Tamanho	Descrição
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 78: Diagnósticos MODBUS Cliente

Nota:

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

5.7.5.1.2. Configuração dos Dispositivos – Configuração por Mapeamento Simbólico

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

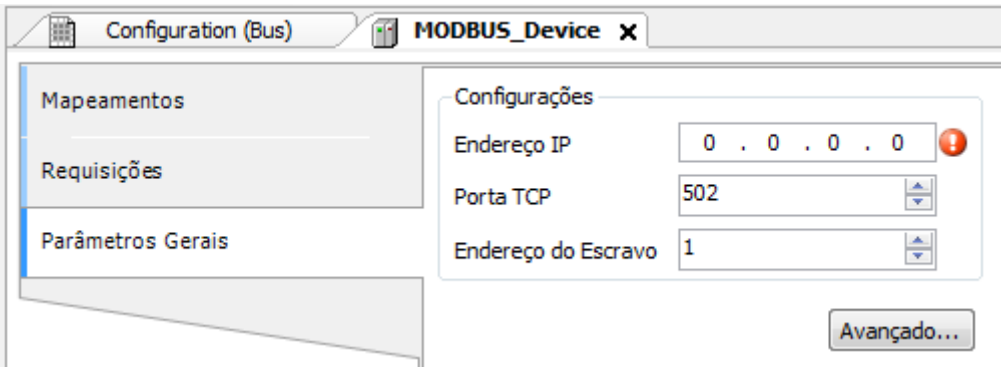


Figura 77: Tela de Configurações dos Parâmetros Gerais do Dispositivo

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 79: 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

Configuração	Descrição	Padrão	Opções
Time-out de Comunicação	Time-out do nível da aplicação em ms	3000	10 a 65535
Modo	Define quando a conexão com o servidor é finalizada pelo cliente	Conexão permanece aberta enquanto não houver um timeout.	Conexão permanece aberta enquanto não houver um timeout. Conexão é fechada ao final de cada comunicação. Conexão é fechada após um tempo de inatividade de (s): 10 a 3600.
Tempo de Inatividade	Tempo de inatividade	10	10 a 3600

Tabela 80: Configurações Avançadas MODBUS Cliente

Notas:

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

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

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

- **Conexão permanece aberta enquanto não houver um timeout:** Esta opção apresenta o mesmo comportamento do Cliente fechar a conexão devido ao fato do Servidor não ter respondido a uma requisição antes do Time-out de Comunicação ter se esgotado.
- **Conexão é fechada ao final de cada comunicação:** A conexão é fechada pelo Cliente após concluir cada requisição.
- **Conexão é fechada após um tempo de inatividade:** A conexão será fechada pelo Cliente caso ele fique por um tempo igual ao Tempo de Inatividade sem realizar requisição para o Servidor.

Tempo de Inatividade: Tempo de inatividade da conexão.

5.7.5.1.3. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico

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

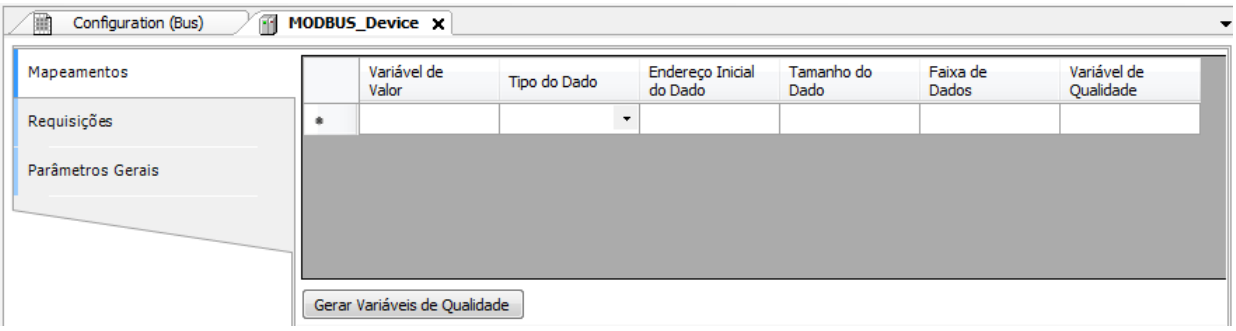


Figura 78: Tela de Mapeamentos de dados MODBUS Cliente

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

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 81: 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 82: Tipos de Dados suportados no MODBUS

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

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

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

5.7.5.1.4. Configuração das Requisições – Configuração por Mapeamento Simbólico

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

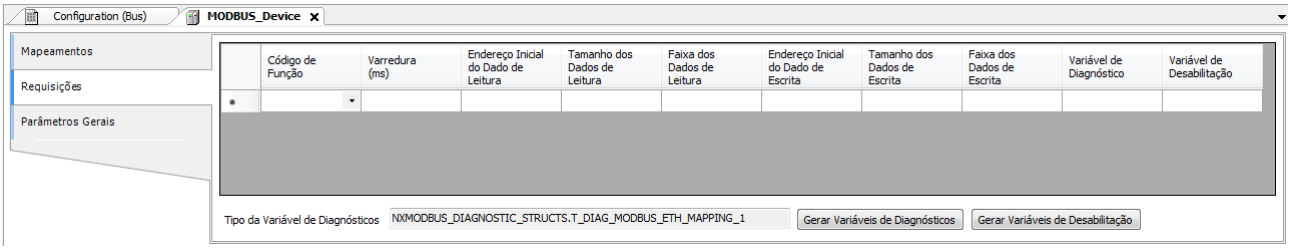


Figura 79: Tela de Requisições de dados MODBUS

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

Tabela 83: 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 84: 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 63.

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 86 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)*

T_DIAG_MODBUS_ETH_MAPPING_1.*	Tamanho	Descrição
		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 85: 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 86: Códigos de Erro das relações MODBUS

ATENÇÃO

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

5.7.5.2. 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.

5.7.5.2.1. 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:

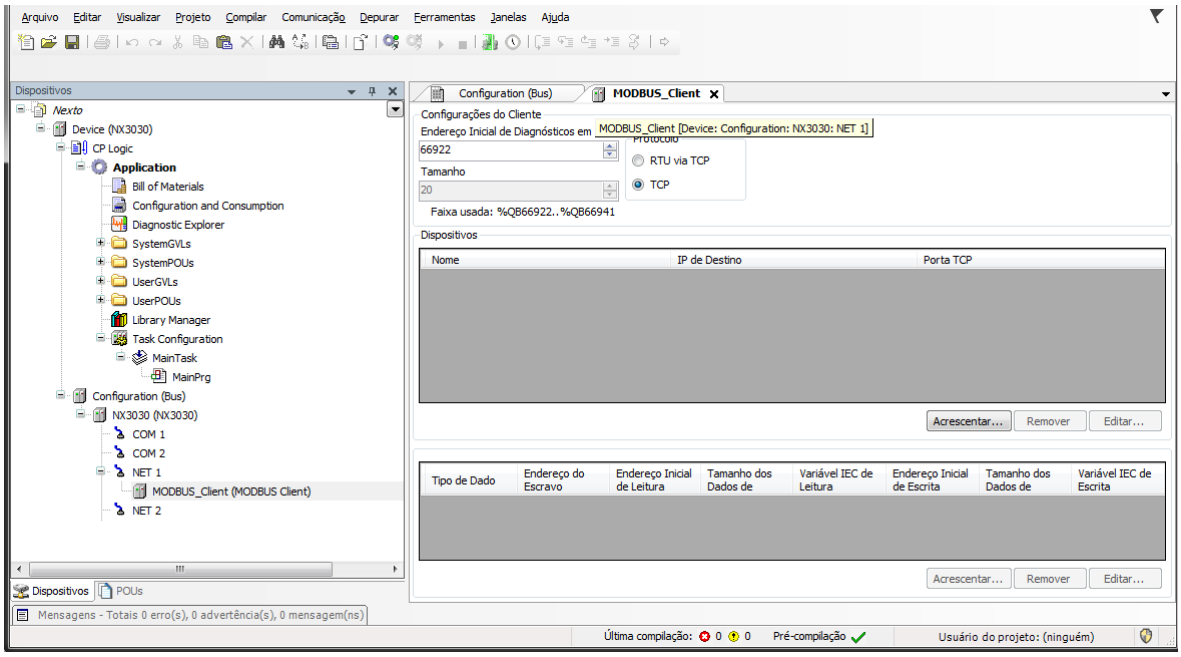


Figura 80: Tela de Configuração – MODBUS Cliente

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 87: Configurações do MODBUS Cliente

Notas:

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

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

Os diagnósticos e comandos do protocolo MODBUS estão descritos na Tabela 78.

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

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

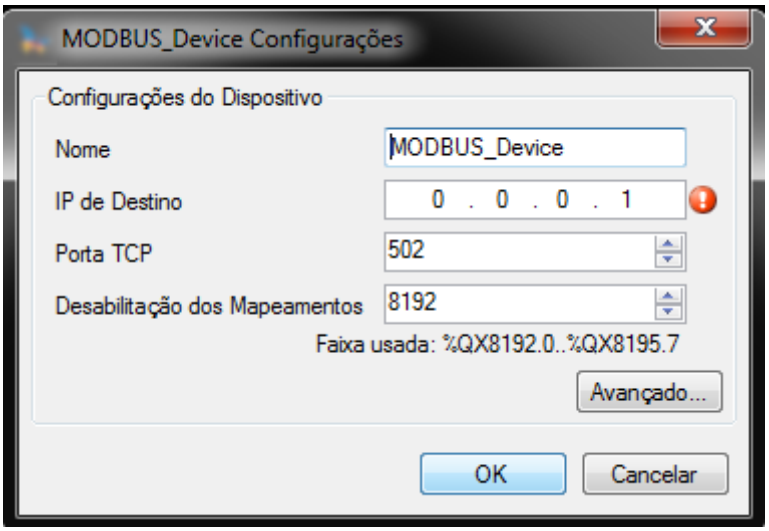


Figura 81: Configurando Cliente MODBUS

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 88: 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 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.

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

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

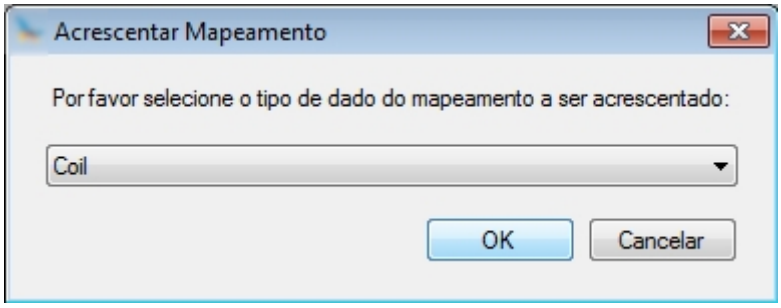


Figura 82: Tipo de Dado MODBUS

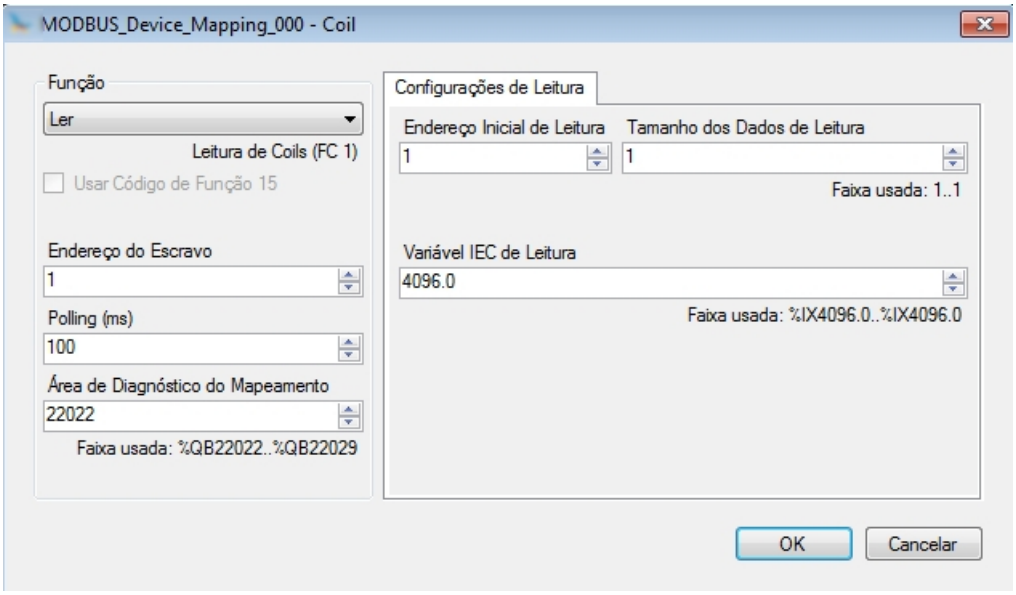


Figura 83: Função MODBUS

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

Configuração	Descrição	Padrão	Opções
Função	Tipo de função MODBUS	Ler	Ler Escrever Ler/Escrever Máscara de Escrita
Endereço do Escravo	Endereço do escravo MODBUS.	1	0 a 255
Polling (ms)	Período de comunicação (ms)	100	0 a 3600000
Área de Diagnóstico do Mapeamento	Endereço inicial dos diagnósticos da relação MODBUS (%Q)	-	0 a 2147483640
Endereço Inicial de Leitura	Endereço inicial dos dados de leitura MODBUS	1	1 a 65536
Tamanho dos Dados de Leitura	Número de dados de leitura MODBUS	-	Depende da função utilizada
Variável IEC de Leitura	Endereço inicial das variáveis de leitura (%I)	-	0 a 2147483647
Endereço Inicial de Escrita	Endereço inicial dos dados de escrita MODBUS	1	1 a 65536
Tamanho dos Dados de Escrita	Número de dados de escrita MODBUS	-	Depende da função utilizada

Configuração	Descrição	Padrão	Opções
Variável IEC de Escrita	Endereço inicial das variáveis de escrita (%Q)	-	0 a 2147483647
Máscara de Escrita das Variáveis IEC	Endereço inicial das variáveis para a máscara de escrita (%Q)	-	0 a 2147483644

Tabela 89: 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 63.

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 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.

5.7.5.3. 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.

5.7.6. MODBUS Ethernet Servidor

Este protocolo está disponível para as UCPs da Série Nexto nos seus canais Ethernet. Ao selecionar esta opção no Mastertool, a UCP passa a ser servidor da comunicação MODBUS, permitindo a conexão com dispositivos cliente MODBUS. Este protocolo somente está disponível quando a UCP estiver em modo de execução (Modo *Run*).

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

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

5.7.6.1. Configuração do Protocolo MODBUS Ethernet Servidor por Mapeamento Simbólico

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

- Configurar os parâmetros gerais do protocolo MODBUS servidor, como: porta TCP, seleção de protocolo, filtros de IP para Escrita e para Leitura (disponível no botão de configuração de filtros) e tempos de comunicação (disponível no botão de configurações avançadas do Servidor).
- Adicionar e configurar os mapeamentos MODBUS, especificando o nome da variável, tipo de dados, endereço inicial do dado e tamanho do dado.

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

5.7.6.1.1. Parâmetros Gerais do Protocolo MODBUS Servidor – Configuração por Mapeamento Simbólico

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

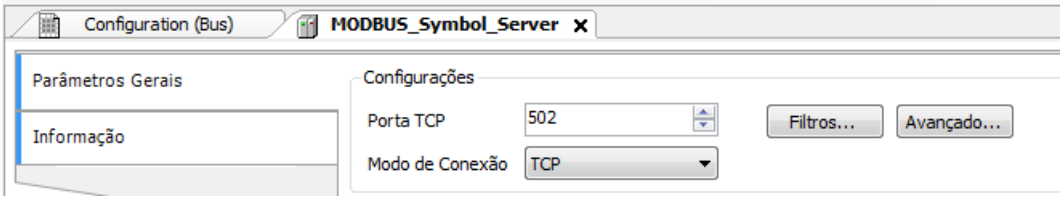


Figura 84: Tela de Configuração Parâmetros Gerais MODBUS Servidor

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

Tabela 90: 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 91: 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*.

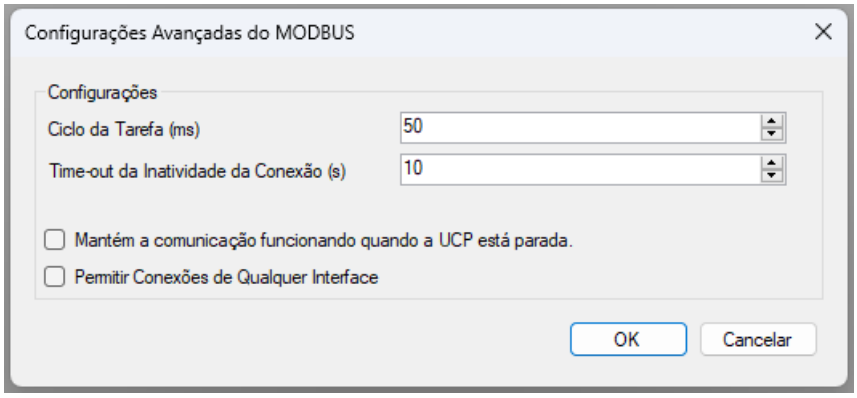


Figura 85: Janela de Configurações Avançadas do MODBUS Servidor

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

Tabela 92: 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.

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

5.7.6.1.2. Diagnósticos MODBUS Servidor – Configuração por Mapeamento Simbólico

Os diagnósticos e comandos do protocolo MODBUS Servidor configurado, seja por mapeamento simbólico ou por representação direta, são armazenados em variáveis do tipo *T_DIAG_MODBUS_ETH_SERVER_1* e ainda para o mapeamento por representação direta estão em 4 bytes e 8 words, as quais estão descritas na tabela abaixo:

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

Tabela 93: Diagnósticos MODBUS Servidor

Nota:

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

bModuleFailure: Diagnóstico implementado apenas para MODBUS simbólico.

5.7.6.1.3. Configuração dos Mapeamentos – Configuração por Mapeamento Simbólico

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

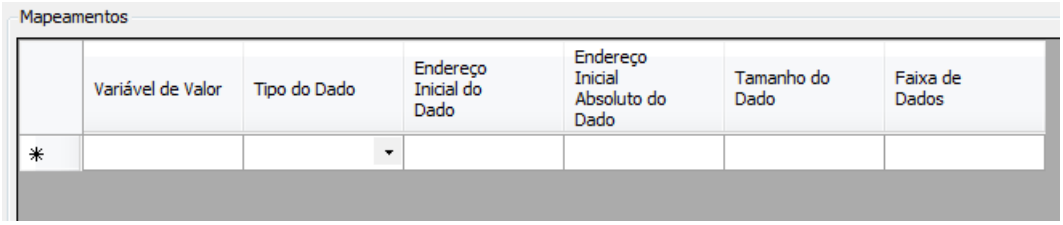


Figura 86: Tela de Mapeamentos de dados MODBUS Servidor

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

Tabela 94: Configuração dos Mapeamentos MODBUS

Notas:

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

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

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

Endereço Inicial Absoluto do Dado: Endereço inicial absoluto dos dados MODBUS conforme o seu tipo. Por exemplo, o Holding Register com endereço 5 possui endereço absoluto 400005. Este campo é apenas de leitura e está disponível para auxiliar na configuração do Cliente/Mestre MODBUS que irá comunicar-se com este dispositivo. Os valores dependem do endereço base (offset) de cada tipo de dado MODBUS e do endereço permitido para cada tipo de dado.

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

Faixa de Dados: É um campo somente de leitura e informa a faixa de endereços que está sendo usada por esse mapeamento. Ele é formado pela soma dos campos *Endereço Inicial* e *Tamanho do Dado*. Não podem haver sobreposições de faixa com outros mapeamentos do mesmo *Tipo de Dado*.

ATENÇÃO

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

5.7.6.2. 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.

5.7.6.2.1. *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:

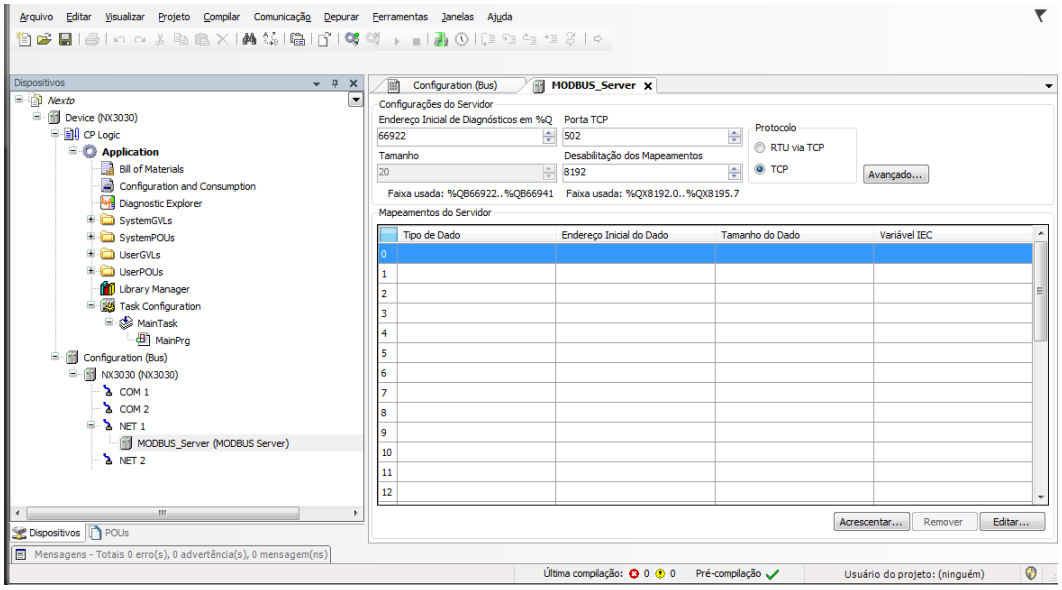


Figura 87: Tela de Configuração MODBUS Servidor

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 95: 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 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 93.

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

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

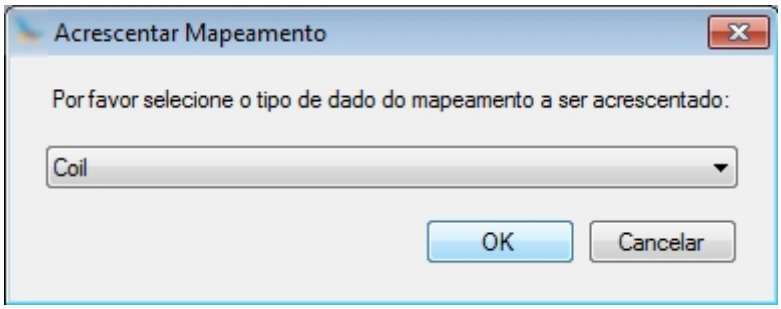


Figura 88: Tipo de Dado MODBUS

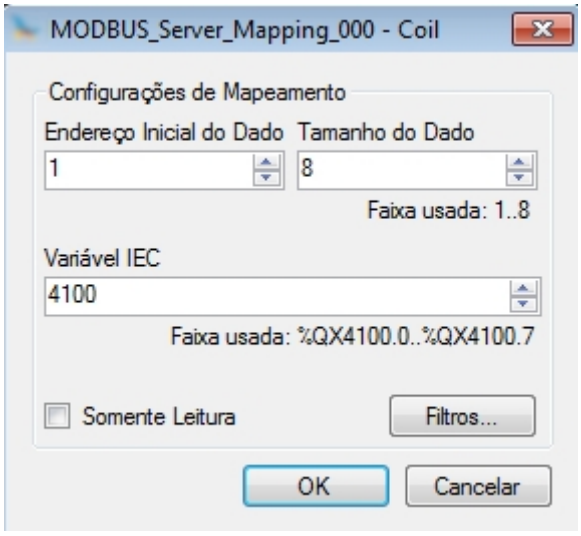


Figura 89: Função MODBUS Servidor

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 96: 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 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 97: 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.

5.7.7. 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).

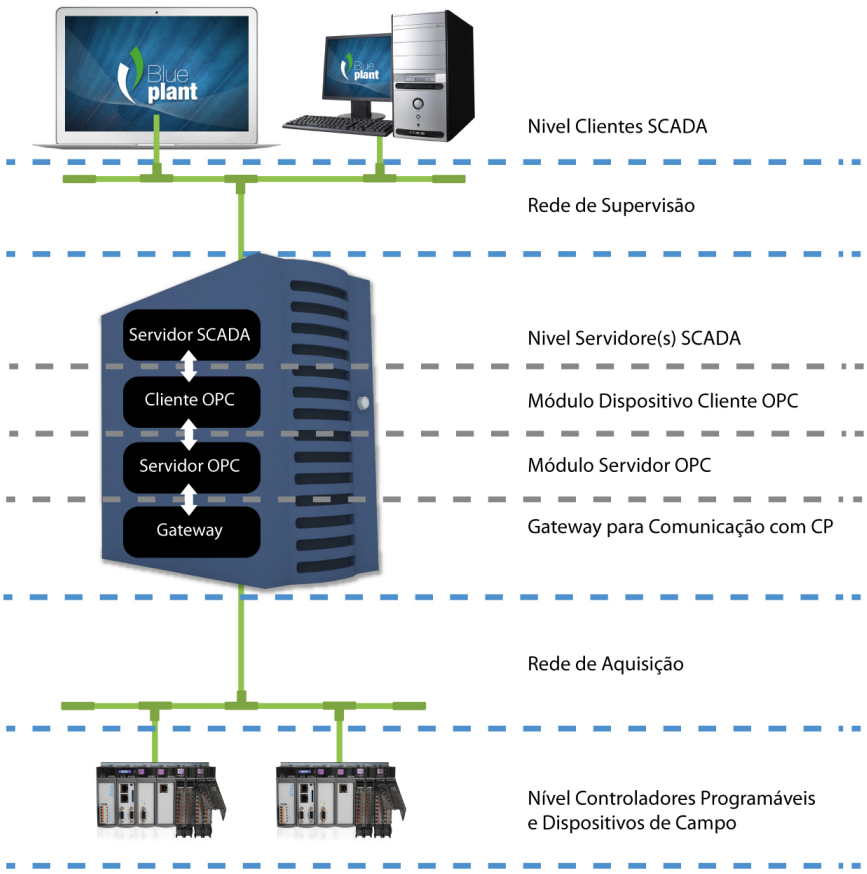


Figura 90: Arquitetura OPC DA

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

Papel	Descrição
Nível Controladores Programáveis e Dispositivos de Campo	Os dispositivos de campo e os CPs são os dispositivos nos quais as informações do estado de operações e controle da planta são armazenados. Os sistemas SCADA acessam as informações nestes dispositivos e armazenam nos servidores SCADA para consulta pelos Clientes SCADA durante a operação da planta.
Rede de Aquisição	A rede de aquisição é a rede na qual trafegam as mensagens para solicitar os dados que são coletados dos dispositivos de campo.
Gateway para Comunicação com CP	Para a comunicação entre o Servidor OPC DA e os CPs da Série Nexto é utilizado um gateway que permite esta comunicação. Sempre é necessário existir um gateway na mesma subrede do CP como descrito no capítulo Configurações de Comunicação, no Manual de Utilização Mastertool.
Módulo Servidor OPC	O Servidor OPC DA é um Módulo responsável por receber as requisições OPC DA e traduzi-las para a comunicação com os dispositivos de campo.

Papel	Descrição
Módulo Dispositivo Cliente OPC	O módulo Dispositivo do Cliente OPC DA é responsável por fazer requisições aos Servidores OPC DA utilizando o protocolo OPC DA. Os dados coletados por ele são armazenados na base de dados do Servidor SCADA.
Nível Servidor SCADA	O Servidor SCADA é responsável por se conectar aos diversos dispositivos de comunicação e armazenar os dados coletados destes dispositivos em uma base de dados para que possam ser consultados pelos Clientes SCADA.
Rede de Supervisão	A rede de supervisão é a rede pela qual os Clientes SCADA estão conectados aos Servidores SCADA. Em uma topologia na qual não se usa diversos Clientes ou que o servidor e o Cliente estejam instalados em um mesmo equipamento, não existe este tipo de rede.
Nível Clientes SCADA	Os clientes SCADA são responsáveis por solicitar aos servidores SCADA os dados necessários para exibir em uma tela onde é executada a operação de uma planta. Através deles é possível executar leituras e escritas em dados armazenados na base de dados do Servidor SCADA.

Tabela 98: Descrição dos Papéis em uma Arquitetura com Servidor OPC DA

A relação entre as tags dos sistemas de supervisão e os dados do processo nas variáveis do controlador é totalmente transparente. Isso significa que se as áreas de dados são alteradas no decorrer do desenvolvimento do projeto, não há a necessidade de refazer relações entre as informações do CP com o SCADA. Basta utilizar a nova variável disponibilizada pelo CP nos sistemas que requisitam esse dado.

O uso do OPC DA oferece maior produtividade e conectividade com os sistemas SCADA. Contribui na redução do tempo de desenvolvimento de aplicações e nos custos com manutenção. Possibilita, ainda, inserção de novos dados na comunicação de forma simplificada com maior flexibilidade e interoperabilidade entre os sistemas de automação por ser um padrão aberto.

O Servidor OPC DA é instalado juntamente com a instalação do Mastertool e sua configuração é realizada dentro da ferramenta. Vale salientar que o OPC DA está disponível somente nas interfaces Ethernet integradas das UCPs Nexto. Os módulos de expansão Ethernet não suportam essa funcionalidade.

5.7.7.1. 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.

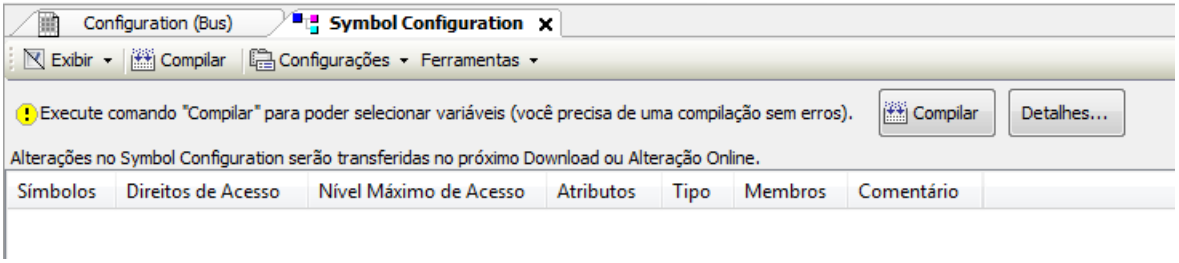


Figura 91: Objeto Symbol Configuration

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 99: 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 91 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.

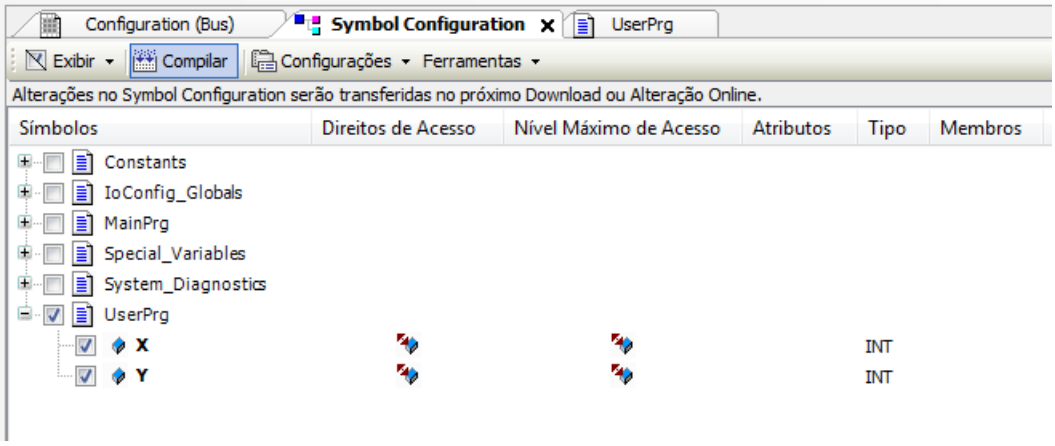


Figura 92: Selecionando Variáveis na Symbol Configuration

Após este procedimento o projeto pode ser carregado em um CP e as variáveis selecionadas estarão disponíveis para comunicação com o Servidor OPC DA. Se a tela do Objeto Symbol Configuration estiver aberta e alguma das variáveis, POUs ou GVLs selecionadas for alterada, os nomes destes objetos aparecerão na cor vermelha. As situações nas quais isso acontece é caso a variável seja deletada ou o valor do atributo tenha sido modificado.

Também é possível configurar quais variáveis estarão disponíveis no Servidor OPC DA através de um atributo inserido diretamente nas POUs ou GVLs onde as variáveis são declaradas. Quando o atributo *attribute 'symbol'* está presente na declaração das variáveis, podendo estar antes da definição do nome da POU ou GVL, ou para cada variável individualmente, estas são enviadas diretamente para o objeto *Symbol Configuration*, as quais são apresentadas com um símbolo na coluna *Atributos*. Antes de carregar o projeto neste caso é necessário executar o comando *Compilar* dentro do objeto *Symbol Configuration*.

As sintaxes válidas para uso do atributo são:

- *attribute 'symbol' := 'none'* – quando o valor do atributo for igual a *'none'* as variáveis não serão disponibilizadas para o Servidor OPC DA e não serão exibidas na tela do objeto *Symbol Configuration*.
- *attribute 'symbol' := 'read'* - quando o valor do atributo for igual a *'read'* as variáveis serão disponibilizadas para o Servidor OPC DA com direito de acesso somente de leitura.
- *attribute 'symbol' := 'write'* - quando o valor do atributo for igual a *'write'* as variáveis serão disponibilizadas para o Servidor OPC DA com direito de acesso somente de escrita.
- *attribute 'symbol' := 'readwrite'* – quando o valor do atributo for igual a *'readwrite'* as variáveis serão disponibilizadas para o Servidor OPC DA com direito de acesso de leitura e escrita.

No exemplo a seguir de declaração de variáveis, a configuração das variáveis A e B permite que um Servidor OPC DA acesse as mesmas com direito de acesso para leitura e escrita. Em contraponto a variável C não pode ser acessada, enquanto a variável D é acessada com direito de acesso apenas para leitura.

```
{attribute 'symbol' := 'readwrite'}
PROGRAM UserPrg
VAR
A: INT;
B: INT;
{attribute 'symbol' := 'none'}
C: INT;
{attribute 'symbol' := 'read'}
D :INT;
END_VAR
```

Quando uma variável diferente dos tipos básicos é definida, o uso do atributo deve ser feito dentro da declaração desta DUT e não somente no contexto onde a variável é declarada. Por exemplo, no caso de uma instância DUT declarada dentro de uma POU ou GVL que possuem um atributo, este não irá impactar no comportamento dos elementos da instância desta DUT. Será necessário aplicar o mesmo nível de acesso na declaração da DUT.

ATENÇÃO

As configurações dos símbolos que serão disponibilizados ao Servidor OPC DA são armazenadas dentro do projeto do CP. Ao modificar estas configurações é necessário carregar a aplicação no CP para que seja possível acessar estas variáveis.

ATENÇÃO

Quando uma variável é excluída do projeto e carregada no CP desmarcando a mesma do objeto *Symbol Configuration*, a variável não pode mais ser lida com o Cliente OPC DA. Se a variável for adicionada ao projeto novamente, com o mesmo nome e o mesmo contexto, e inserida no objeto *Symbol Configuration*, será necessário reinicializar o Cliente OPC DA para atualizar a referência do endereço da variável, que passa a ser criada em outra área de memória no CP.

5.7.7.2. Configurando um CP no Servidor OPC DA

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

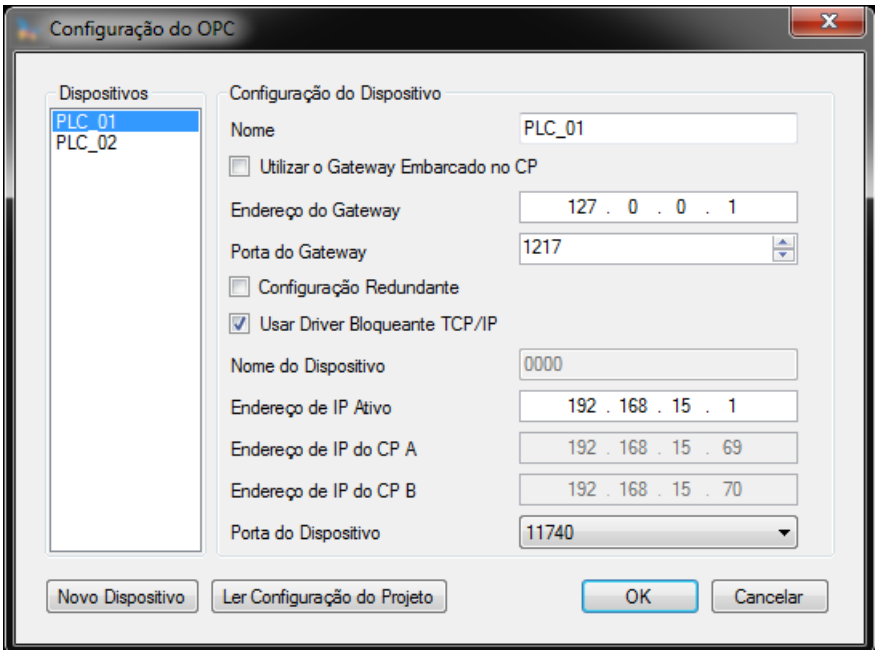


Figura 93: Configuração do OPC DA Servidor

A *Configuração do Gateway* é a mesma configurada no Gateway utilizado para comunicação entre o Mastertool e o CP e descrita em Configurações de Comunicação, presente no Manual de Utilização Mastertool. Se a configuração utilizada for *localhost* o *Endereço do Gateway* deve ser preenchido com 127.0.0.1. Esta configuração é necessária, pois o Servidor OPC utiliza o mesmo gateway de comunicação e o mesmo protocolo utilizados na comunicação entre CP e Mastertool.

Existe a opção *Utilizar o Gateway Embarcado no CP* que pode ser selecionada quando se deseja utilizar o Gateway que fica no próprio CP. Esta opção pode ser empregada para otimizar a comunicação, pois ela evita o excesso de tráfego através de uma determinada estação, quando mais de uma estação, com Cliente OPC DA, esteja conectada ao mesmo CP.

Para a configuração do CP, são possíveis dois tipos de configuração, conforme a seleção do checkbox *Usar Driver Bloqueante TCP/IP*. Quando a opção não está selecionada o nome do CP deve ser colocado no campo *Nome do Dispositivo*. Este é o nome exibido pelo CP selecionado como ativo na tela de *Configurações de Comunicação*.

A outra opção é usar o *Endereço de IP* das Interfaces Ethernet. O mesmo endereço configurado nas telas de configuração deve ser colocado neste campo. Além disso, quando for utilizado este método deve ser colocado o número da porta 11740. A 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 100: Parâmetros de Configuração de Cada CP para o Servidor OPC DA

Quando um novo CP precisar ser configurado no Servidor OPC DA basta pressionar o botão *Novo Dispositivo* que a configuração será criada. Sempre que a tela de configuração for acessada será exibida uma lista com todos os CPs já configurados no Servidor OPC DA. As configurações existentes podem ser editadas selecionando o CP na lista *Dispositivos* e editando os parâmetros. As configurações de CPs que não são mais utilizadas podem ser excluídas. O número máximo de CPs configurados em um Servidor OPC DA é 16.

Caso a arquitetura de automação utilizada preveja que o servidor OPC DA deve ser executado em um computador onde não é executada a comunicação com o CP via Mastertool, a ferramenta deve ser instalada neste computador para permitir a configuração do Servidor OPC DA da mesma maneira como é feito nas outras situações.

ATENÇÃO

Para armazenar a configuração do Servidor OPC DA, o Mastertool precisa ser executado com direitos de administrador no Sistema Operacional. Dependendo da versão do Sistema Operacional este direito deve ser autorizado ao executar o programa. Para essa operação clique com o botão direito sobre o executável do Mastertool e escolha a opção *Executar como administrador*.

ATENÇÃO

As configurações de um CP no Servidor OPC DA não são armazenadas no projeto criado no Mastertool. Por esta razão podem ser realizadas com um projeto aberto ou fechado. As configurações são armazenadas em um arquivo de configuração onde o Servidor OPC DA está instalado. Quando alterar as configurações não é necessária carga de aplicação no CP, mas dependendo do Cliente OPC DA é possível que seja necessário conectar novamente ao Servidor ou carregar as configurações para que os dados sejam atualizados corretamente.

5.7.7.2.1. Importando uma Configuração do Projeto

Utilizando o botão *Ler Configuração do Projeto*, conforme a Figura 93, é possível atribuir a configuração do projeto aberto à configuração do CP que está em edição. Para que esta opção funcione corretamente deve existir um projeto aberto e deve ser definido um *Caminho Ativo* conforme descrito em *Configurações de Comunicação*, presente no Manual de Utilização Mastertool. Caso alguma destas condições não seja atendida será exibida uma mensagem de erro e nenhum dado será modificado.

Quando as condições anteriores são válidas, as configurações do CP recebem os parâmetros do projeto aberto. As informações de *Endereço de IP* e *Porta do Gateway* são configuradas conforme descrito em *Configurações de Comunicação* de acordo com o *Caminho Ativo*. Entretanto, as configurações de *Endereço de IP* são lidas das configurações da interface Ethernet NET1. A porta para conexão com o CP é sempre atribuída neste caso como 11740.

5.7.7.3. Configuração com CP no Servidor OPC DA com Redundância de Conexão

É possível configurar o Servidor OPC DA para que este opere com redundância de conexão. Desta forma o Servidor OPC DA irá se comunicar preferencialmente com um CP, mas quando, por alguma razão, não conseguir estabelecer uma comunicação com este CP um segundo CP também configurado será acessado. Esta configuração é especialmente importante para a comunicação de sistemas SCADA com os CPs da Série Nexto que utilizam redundância de Half-Cluster, onde existe um CP em estado ativo executando o processo e outro CP em estado reserva apto a assumir o controle do processo quando ocorrer algum tipo de falha.

A configuração do projeto nestes casos é semelhante a descrita em Criando um Projeto para Comunicação OPC DA. Contudo, quando um Projeto é criado com Redundância de Half-Cluster e a comunicação com o sistema de supervisão se dará através do Servidor OPC DA, é necessário selecionar a opção da *Configuração de comunicação OPC DA* como habilitada durante o Assistente de criação de projeto do Mastertool. Ao habilitar esta opção o projeto criado terá o código necessário para funcionamento da comunicação com redundância de conexão OPC DA.

No caso redundante, uma variável é declarada dentro da POU, chamada NonSkippedPrg. Esta POU é executada em ambos os CPs, independente do estado de redundância. Dentro desta POU é declarada uma variável do tipo BOOL, utilizada para o controle da conexão com o Servidor OPC DA chamada *OPCRedundancyActive*. Esta variável pode ser acessada de qualquer ponto da aplicação, através de todo o contexto, ou seja, *Application.NonSkippedPrg.OPCRedundancyActive*. Ela é declarada dentro do objeto *Symbol Configuration* com direito apenas de leitura por parte do SCADA. Quando o valor da variável for igual a TRUE os dados são lidos através da conexão com este CP. Desta forma toda vez que ocorre uma troca de estado entre os CPs a variável tem seu estado alterado, permanecendo no estado TRUE no CP que está no estado ativo de redundância.

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

```
PROGRAM NonSkippedPrg
VAR
  {attribute 'symbol' := 'read'}
  OPCRedundancyActive : BOOL;
END_VAR

IF fbRedundancyManagement.m_fbDiagnosticsLocal.eRedState = REDUNDANCY_STATE.
   ACTIVE THEN
  OPCRedundancyActive := TRUE;
ELSE
  OPCRedundancyActive := FALSE;
END_IF
```

O código do programa *NonSkippedPrg* pode ser editado tomando-se o cuidado de manter o código acima sem alterações. Este código testa o estado da redundância e preenche uma variável do tipo BOOL chamada *OPCRedundancyActive*, em função deste estado. Caso o CP seja o Ativo, o valor da variável será TRUE, caso contrário será FALSE. Esta variável recebe o *attribute 'symbol' := 'read'* para permitir que o Servidor OPC DA acesse o seu conteúdo e defina de onde a informação deve ser lida.

Caso se decida adicionar comunicação OPC DA após um projeto ter sido criado, é possível configurar o OPC DA adicionando o código anterior no programa *NonSkippedPrg* e adicionando o objeto *Symbol Configuration* ao projeto.

Para a configuração do CP redundante no Servidor OPC DA é necessário selecionar a opção *Configuração Redundante* na tela de configuração conforme exibido na Figura 93. Quando esta opção é selecionada, será sempre utilizada a opção *Usar Driver Bloqueante TCP/IP*. Além disso, serão habilitados os campos *Endereço de IP do CPA* e *Endereço de IP do CPB* conforme descrito na Tabela 100. Estes *Endereços de IP* são os mesmos configurados nas interfaces Ethernet dentro do projeto do CP com redundância de Half-Cluster. Para facilitar a configuração quando um projeto redundante estiver aberto, o botão *Ler Configuração do Projeto* pode ser utilizado.

ATENÇÃO

A redundância de conexão do Servidor OPC DA é realizada por apenas um Servidor. Nos casos em que se deseje uma maior disponibilidade dos dados para os sistemas de supervisão deve ser usada uma arquitetura com Servidores SCADA redundantes. Nestes casos não é necessária nenhuma configuração no Servidor OPC DA. Consulte as documentações do sistema SCADA para verificar quais as configurações necessárias para o funcionamento da arquitetura redundante.

5.7.7.4. Variáveis de Status e Qualidade da Comunicação OPC DA

Para cada um dos CPs criados no Servidor OPC DA são geradas variáveis de status chamadas de *_CommState* e *_CommStateOK*. A variável *_CommState* indica o estado da comunicação do Servidor OPC DA com CP. Este estado pode ser interpretado pelo Cliente OPC DA conforme a tabela abaixo.

Estado	Valor	Descrição
STATE_TERMINATE	-1	Se a comunicação do Servidor OPC DA com o Cliente OPC DA for encerrada este valor será retornado. Quando houver mais de um Cliente OPC DA conectado ao mesmo tempo o retorno ocorrerá na desconexão do último Cliente conectado. Este status apesar de estar na variável não pode ser visualizado, pois só é alterado quando não existe mais conexão com um cliente.
STATE_PLC_NOT_CONNECTED	0	O CP configurado no Servidor OPC DA não está conectado. Pode acontecer caso a configuração esteja incorreta (Endereço de IP do CP e/ou Gateway errados) ou o CP não esteja disponível naquele momento.
STATE_PLC_CONNECTED	1	O CP configurado no Servidor OPC DA está conectado. Este é um estado transitório durante conexão.
STATE_NO_SYMBOLS	2	Não existem símbolos (variáveis) disponíveis no CP configurado no Servidor OPC DA. Pode acontecer na situação de não existirem símbolos ou ainda quando não houver um projeto carregado no CP.
STATE_SYMBOLS_LOADED	3	Finalizado o processo de leitura dos símbolos (variáveis) do CP configurado no Servidor OPC DA. Este é um estado transitório durante conexão.
STATE_RUNNING	4	Após leitura dos símbolos (variáveis) o Servidor OPC DA está executando a atualização periódica dos valores dos símbolos disponíveis em cada CP configurado.
STATE_DISCONNECT	5	Ocorreu desconexão com o CP configurado no servidor OPC DA.
STATE_NO_CONFIGURATION	6	Quando a configuração do OPC DA (armazenada em um arquivo OPCServer.ini) possuir uma sintaxe errada este será o status da variável. De maneira geral, este comportamento não é observado pois o Mastertool mantém esta configuração íntegra.

Tabela 101: 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 102: Descrição do valor da Qualidade OPC DA

A Tabela 102 apresenta os valores possíveis de qualidade. O Servidor OPC DA retorna apenas valor com Qualidade *Good* e *Bad*. Um Cliente OPC DA pode manter a qualidade como *Uncertain* em caso de alguma falha na qual ele não consiga uma conexão com o Servidor. No caso de monitoração dos 8 bits de qualidade diretamente do Servidor OPC DA os campos *Substatus* e *Limite* serão nulos e a Qualidade *Good* será representada com o valor 192 e a qualidade *Bad* com o valor 0.

5.7.7.5. Limites da Comunicação com Servidor OPC DA

A tabela abaixo apresenta os limites de configuração do Servidor OPC DA.

Número máximo de variáveis comunicando com um único CP	Ver Nota
Número máximo de CPs em um Servidor OPC DA	16
Número máximo de conexões simultâneas de Servidor OPC DA em um mesmo CP	8

Tabela 103: Limites da Comunicação com o Servidor OPC DA

Nota:

Número máximo de variáveis comunicando com um único CP: Não há limite de configuração. A quantidade máxima possível de variáveis depende da capacidade de processamento do dispositivo.

> **ATENÇÃO**
>
> O número máximo de conexões simultâneas de Servidor OPC DA em um mesmo CP é compartilhado com as conexões realizadas com o Mastertool. Ou seja, a soma de conexões com Servidor OPC DA e Mastertool não deve ultrapassar o número máximo definido na Tabela 103.

A comunicação entre o Servidor OPC DA e o CP utiliza o mesmo protocolo utilizado para comunicação do Mastertool com o CP. Este protocolo só está disponível para as interfaces Ethernet das UCPs da Série Nexto, não sendo possível estabelecer este tipo de comunicação com módulos de expansão Ethernet.

Quando uma comunicação é estabelecida entre o Servidor OPC DA e o CP estes dois elementos iniciam uma série de transações que visam resolver o endereço de cada variável declarada, otimizando a comunicação em regime de leitura de dados. Além disso, nesta fase também são resolvidas as classificações dos grupos de comunicação usados por alguns Clientes com o intuito de otimizar a comunicação. Este processo inicial demanda algum tempo e depende da quantidade de variáveis mapeadas e da capacidade de processamento do dispositivo.

5.7.7.6. Acessando Dados Através de um Cliente OPC DA

Após a configuração do Servidor OPC DA os dados disponíveis em todos os CPs podem ser acessados via um Cliente OPC DA. Na configuração do Cliente OPC DA deve ser selecionado o nome do Servidor OPC DA correto. Neste caso o nome é *CoDeSys.OPC.DA*. A figura abaixo exibe a seleção do servidor no driver cliente do software SCADA BluePlant.

> **ATENÇÃO**
>
> Da mesma forma que o Mastertool 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*.

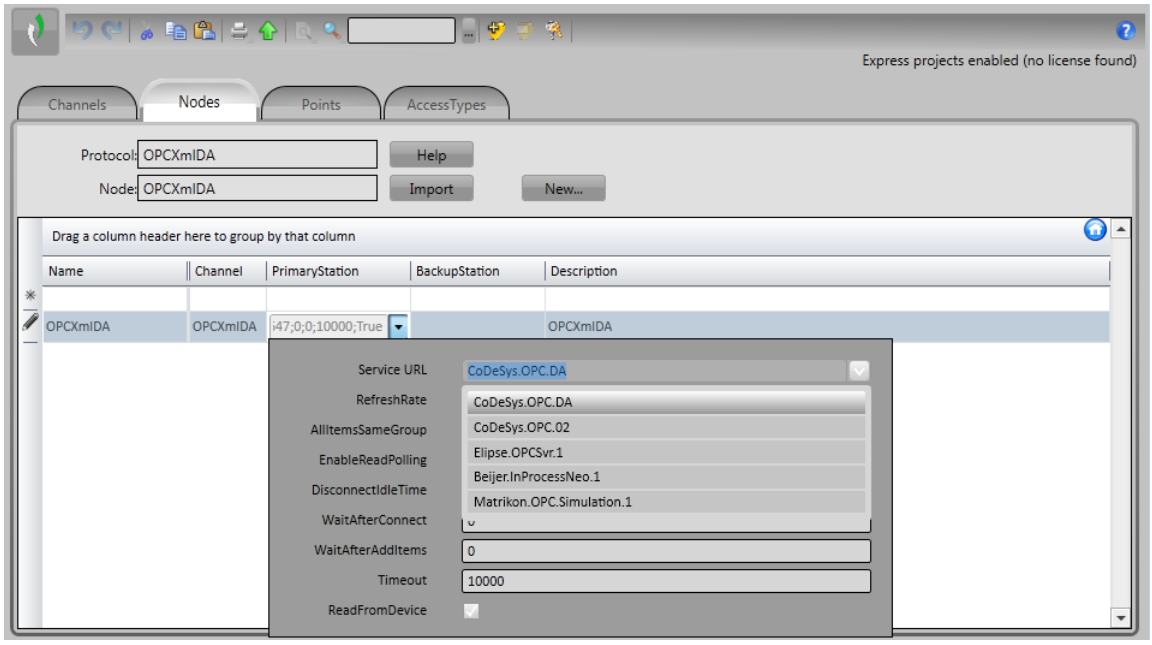


Figura 94: Selecionando Servidor OPC DA na Configuração do Cliente

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.

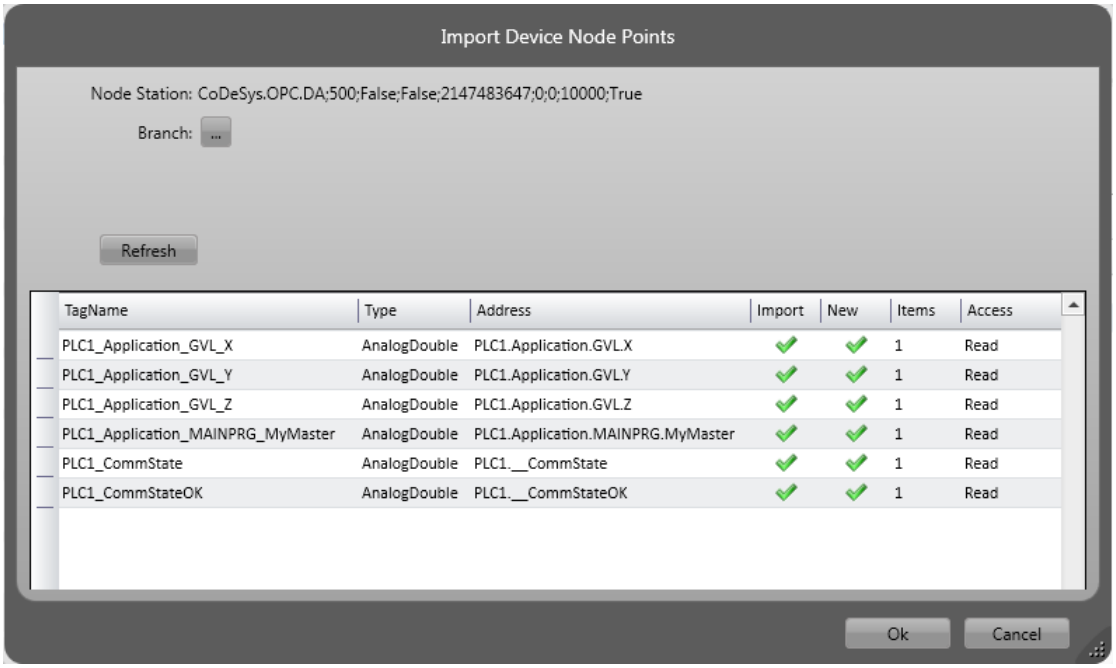


Figura 95: Lista de Símbolos Consultados pelo Cliente OPC DA

A lista de variáveis selecionadas será incluída na lista de comunicações do Cliente e podem ser utilizadas, por exemplo, em telas de um sistema SCADA.

ATENÇÃO

O modo de simulação do software Mastertool pode ser utilizado para testes da comunicação OPC DA. As informações sobre como configurá-lo estão presentes na seção *Testando a Comunicação OPC com o Uso do Simulador*, do Manual de Utilização Mastertool.

5.7.8. OPC UA Servidor

O protocolo OPC UA é uma evolução da família OPC. Independente de plataforma, foi concebido para ser o novo padrão utilizado nas comunicações industriais.

Baseado na arquitetura cliente/servidor, o protocolo OPC UA oferece inúmeras vantagens no desenvolvimento de projeto e facilidades na comunicação com os sistemas de automação.

Quando se trata do desenvolvimento de projetos, configurar a comunicação e trocar informações entre os sistemas é extremamente simples utilizando tecnologia OPC UA. Utilizando outros drivers, baseados em endereços, é necessário criar tabelas para relacionar as tags do sistema de supervisão e as variáveis do controlador programável. Quando as áreas de dados são alteradas, no decorrer do desenvolvimento do projeto, é necessário refazer os mapeamentos e novas tabelas com as relações entre as informações do CP com o sistema SCADA.

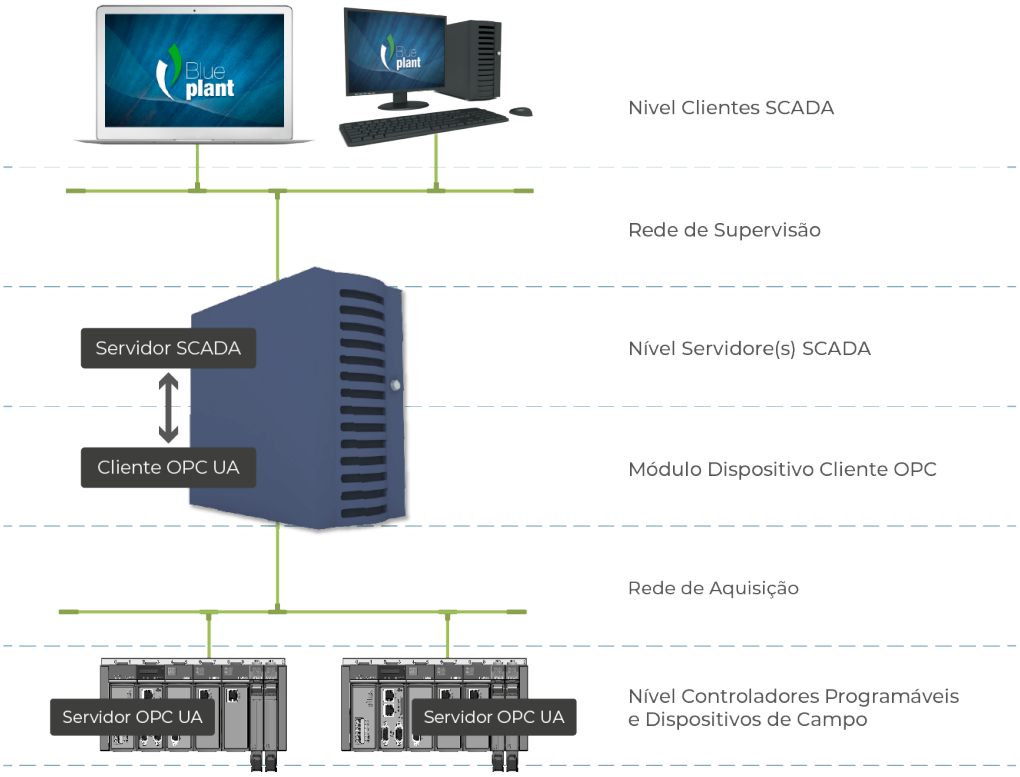


Figura 96: Arquitetura Típica OPC UA

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 104: Descrição dos Papéis em uma Arquitetura com Servidor OPC UA

Ao utilizar o protocolo OPC UA, a relação entre as tags dos sistemas de supervisão e os dados do processo nas variáveis do controlador é totalmente transparente. Isso significa que se as áreas de dados são alteradas no decorrer do desenvolvimento do projeto, não há a necessidade de refazer relações entre as informações do CP com o SCADA. Basta utilizar a nova variável disponibilizada pelo CP nos sistemas que requisitam esse dado.

O uso do OPC UA oferece maior produtividade e conectividade com os sistemas SCADA. Contribui na redução do tempo de desenvolvimento de aplicações e nos custos com manutenção. Possibilita, ainda, inserção de novos dados na comunicação de forma simplificada com maior flexibilidade e interoperabilidade entre os sistemas de automação por ser um padrão aberto.

Vale salientar que o OPC UA está disponível somente nas interfaces Ethernet integradas das UCPs Nexto. Os módulos de expansão Ethernet não suportam essa funcionalidade.

5.7.8.1. 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.

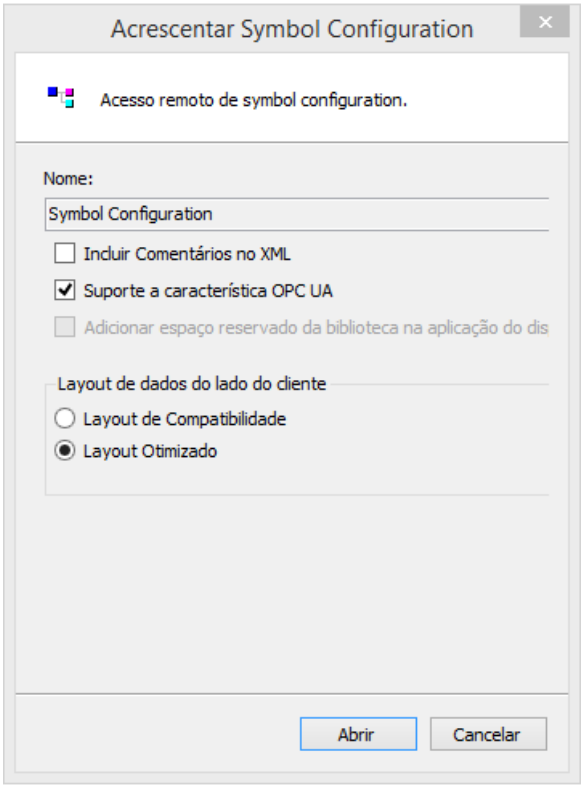


Figura 97: Objeto Symbol Configuration

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.

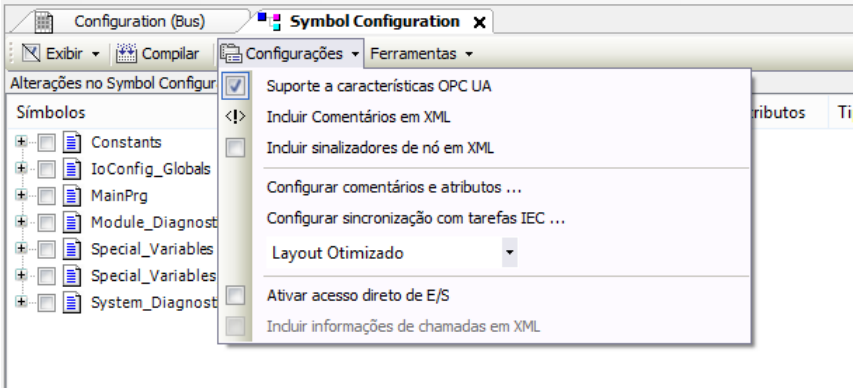


Figura 98: Habilitando OPC UA no Objeto Symbol Configuration

Após este procedimento o projeto pode ser carregado em um CP e as variáveis selecionadas estarão disponíveis para comunicação com o Servidor OPC UA.

5.7.8.2. 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.

5.7.8.3. Limite de Clientes Conectados no Servidor OPC UA

O número máximo de clientes OPC UA conectados simultaneamente num CP é 8 (oito).

5.7.8.4. 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.

5.7.8.5. 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.

5.7.8.6. 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*.

5.7.8.6.1. 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.

5.7.8.6.2. 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.

5.7.8.6.3. 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*.

5.7.8.6.4. 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 UCP, 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.

5.7.8.6.5. 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.

5.7.8.6.6. *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.

5.7.8.7. 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, 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*.

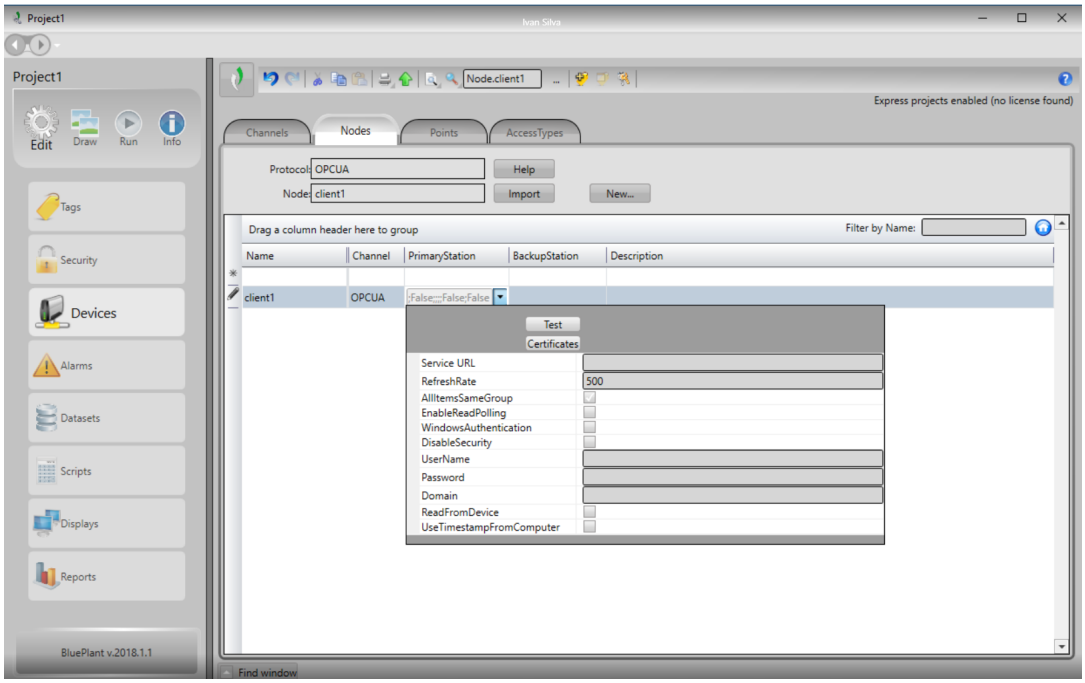


Figura 99: Selecionando Servidor OPC UA na Configuração do Cliente

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.

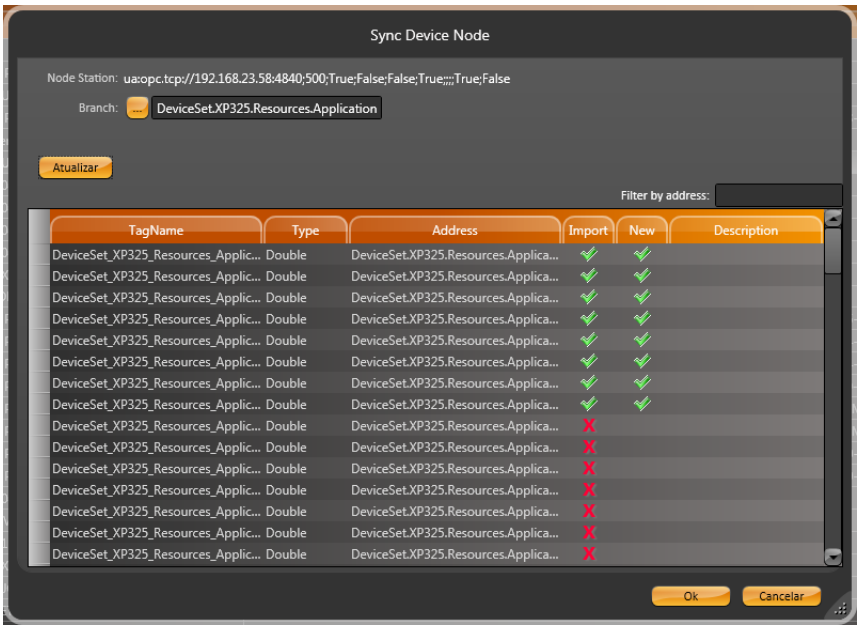


Figura 100: Lista de Símbolos Consultados pelo Cliente OPC UA

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.

5.7.9. Mestre EtherCAT

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

Um frame Ethernet pode ser processado em tempo real em vez de ser recebido, interpretado e copiado como dados do processo em cada conexão. O FMMU (*Fieldbus Memory Management Unit*) em cada nó Escravo lê os dados que lhe são endereçados ao mesmo tempo em que o telegrama é encaminhado para o próximo dispositivo. De uma maneira similar, os dados de entrada são inseridos enquanto o telegrama é passado. Devido a isso, os frames são atrasados apenas por alguns nanosegundos. Acessos nos terminais de Ethernet podem ser feitos em qualquer ordem, porque a sequência de dados é independente da ordem física. É possível executar comunicação Broadcast, Multicast e comunicação entre os Escravos.

O protocolo EtherCAT permite uma sincronização com precisão, que é necessária, por exemplo, em aplicações onde vários eixos realizam movimentos coordenados simultaneamente, esta pode ser realizada através de um ajuste exato do *Distributed Clock*. Há também a possibilidade de configuração de dispositivos que, em contraste com a comunicação síncrona, têm um elevado grau de tolerância dentro do sistema de comunicação.

A configuração de módulos EtherCAT inicialmente é determinada pelos *Arquivos de Descrição de Dispositivo* dos dispositivos Mestre e Escravo utilizados e pode ser modificada pelo usuário nas caixas de diálogo do *Editor de Configuração*. No entanto, para aplicações convencionais e com o objetivo de ter uma manipulação tão fácil quanto possível, a configuração em larga escala pode ser automatizada, escolhendo o modo *Autoconfig master/slaves* em EtherCAT Master - General.

Observe a possibilidade de modificação dos parâmetros de configuração do Mestre e Escravo também no modo operacional, através das instâncias de Mestre e Escravo, de acordo com a disponibilidade do dispositivo em questão.

5.7.9.1. Instalando e Inserindo Dispositivos EtherCAT

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

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

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

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

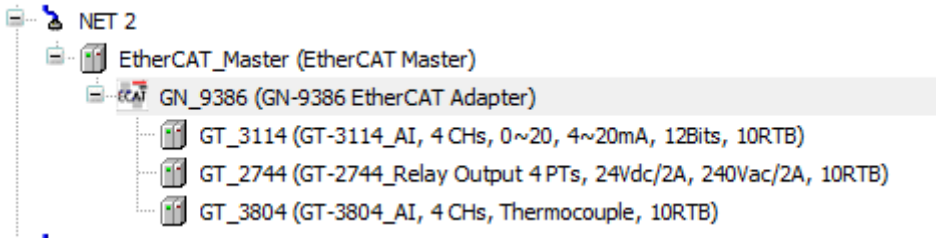


Figura 101: Exemplo de Configuração EtherCAT

ATENÇÃO

- É permitida apenas uma instância do Mestre EtherCAT por projeto.
- Permitido apenas em portas NET integradas das UCPs.
- Não pode ser usado quando as NETs estiverem configuradas como redundantes/switch.
- Não pode ser usado quando o projeto tiver redundância de cluster.
- Não podem ser instanciados outros drivers na mesma porta NET que o Mestre EtherCAT.

5.7.9.1.1. EtherCAT - Scan Devices

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

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

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

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

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

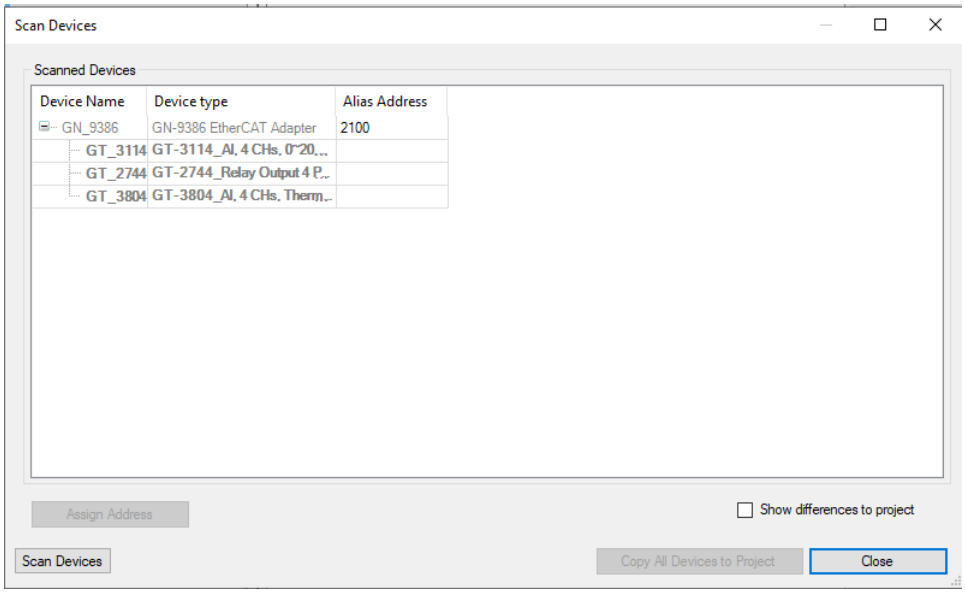


Figura 102: Diálogo Procurar Dispositivos para EtherCAT

5.7.9.2. Configuração do Mestre EtherCAT

A seguir são listadas as opções para a execução da configuração de um Mestre EtherCAT, tal como definido no *Arquivo de Descrição do Dispositivo*.

5.7.9.2.1. EtherCAT Master - General

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

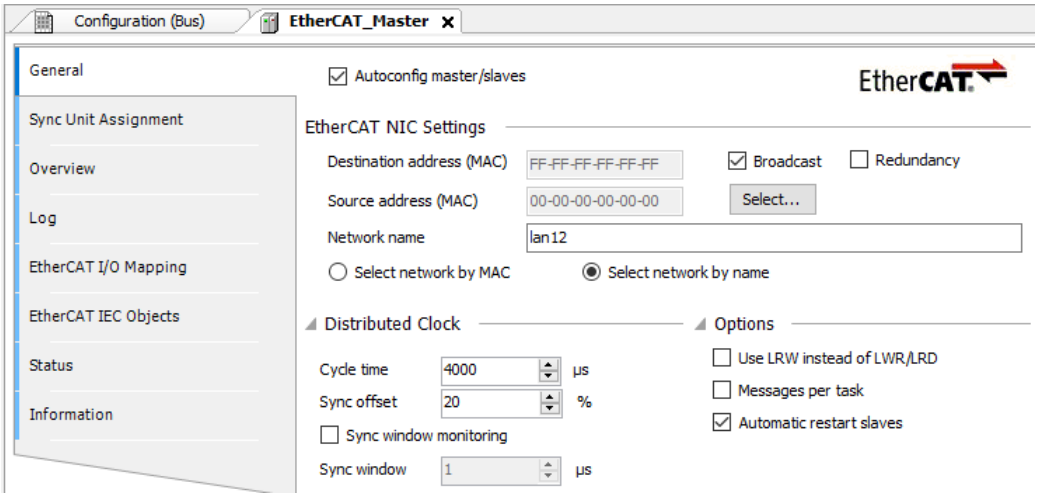


Figura 103: Diálogo de Configuração do Mestre EtherCAT

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

Configuração	Descrição	Padrão	Opções
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 105: Configurações do Mestre EtherCAT

Notas:

Autoconfig master/slaves: Se esta opção estiver ativada, a maior parte da configuração do Mestre e Escravo será feita automaticamente, baseando-se nos arquivos de descrição e cálculos implícitos. Neste caso, o diálogo de configurações FMMU / Sync não estará disponível. Se esta estiver desmarcada as opções *Endereço Imagem In* e *Endereço Imagem Out* estarão disponíveis para o usuário.

ATENÇÃO

O modo *Autoconfig* é ativado por padrão e, geralmente, suficiente e fortemente recomendado para aplicações padrão. Se a opção estiver desativada, todas as definições de configuração para o Mestre e o(s) Escravo(s) terão de ser feitas manualmente e um conhecimento especializado é necessário. Para a configuração de uma comunicação Escravo-para-Escravo a opção *Autoconfiguração* tem de ser desativada.

Cycle time: Período de tempo após o qual, um novo telegrama de dados deve ser enviado ao barramento. Se a funcionalidade *Relógio Distribuído* estiver ativada, o valor deste parâmetro será transferido para os clocks dos Escravos. Assim, uma sincronização precisa de troca de dados pode ser alcançada, o que particularmente é importante nos casos em que os processos distribuídos especialmente exigem ações simultâneas. Então, uma base de tempo muito precisa para toda a rede, com um jitter significativamente menor do que um microssegundo, pode ser alcançada.

Sync offset: Este valor permite ajustar o deslocamento da interrupção de sincronização do Escravo EtherCAT para o ciclo do CP. Normalmente, o ciclo de tarefa do CP começa 20% mais tarde do que a sincronização de interrupção dos Escravos. Isto significa que a tarefa do CP pode ser atrasada por 80% do tempo de ciclo e nenhuma mensagem será perdida.

Sync window monitoring: Se a sincronização de todos os Escravos está dentro desta janela de tempo o diagnóstico do Mestre EtherCAT *bDistributedClockInSync* será definido como TRUE, caso contrário será FALSE. Quando o Relógio Distribuído é utilizado, é altamente recomendado usar uma tarefa dedicada com alta prioridade como *Tarefa Cíclica de Barramento* do mestre EtherCAT. Desta maneira, é necessário utilizar Perfis de Projeto que possibilitem a criação de novas tarefas, criar uma tarefa cíclica com prioridade 0 (tarefa de tempo real), e atribuir a *Tarefa Cíclica de Barramento* do mestre para esta nova tarefa na aba EtherCAT Master - I/O Mapping do Mestre EtherCAT. O usuário também poderá alterar o valor da variável *wDCInSyncWindow*, configurando qual o jitter máximo permitido na sincronização entre mestre e escravos.

Use LRW instead of LWR/LRD: A ativação desta opção habilita a comunicação Escravo-para-Escravo, pois, ao invés de comandos separados de leitura (LRD) e escrita (LWR), comandos combinados leitura/escrita (LRW) serão utilizados.

Automatic restart slaves: Habilitando esta opção o Mestre reiniciará os Escravos assim que a comunicação for abortada.

5.7.9.2.2. EtherCAT Master - Sync Unit Assignment

Esta guia do editor de configuração do Mestre EtherCAT mostra todos os escravos que são inseridos abaixo de um mestre específico com uma atribuição para as unidades de sincronização.

Com o EtherCAT sync units, múltiplos escravos são configurados em grupos e subdivididos em unidades menores. Para cada grupo, o contador de trabalho pode ser monitorado para uma melhor e mais exata detecção de erros. Assim que um escravo está em falta num grupo de unidades de sincronização, os outros escravos do grupo também são mostrados como desaparecidos. A detecção ocorre imediatamente no próximo ciclo de barramento, porque o contador de trabalho é verificado continuamente. Com os diagnósticos do dispositivo, o grupo em falta pode ser remediado o mais rapidamente possível.

Os grupos não afectados permanecem operáveis sem qualquer interferência.

5.7.9.2.3. EtherCAT Master - Overview

Esta guia do editor de configuração do Mestre EtherCAT fornece uma visão geral dos estados de todos os escravos, que são inseridos abaixo deste mestre e têm um endereço. Os módulos não são exibidos.

5.7.9.2.4. EtherCAT Master - I/O Mapping

Esta guia do editor de configuração do Mestre EtherCAT oferece a possibilidade de mudar a tarefa que será utilizada para atualização do barramento.

5.7.9.2.5. EtherCAT Master - IEC Objects

Esta guia do editor de configuração do Mestre EtherCAT lista *objetos* que permitem o acesso ao dispositivo a partir da aplicação IEC. Em modo online, este é utilizado para monitoração.

5.7.9.2.6. EtherCAT Master - Status / Informations

A guia Status do editor de configuração do Mestre EtherCAT fornece informações de estado (por exemplo, '*Executando*', '*Parado*') e mensagens de diagnóstico específicas do dispositivo e do sistema de barramento interno.

A guia Informação, presente no editor de configuração do Mestre EtherCAT, exibe, caso disponíveis, as seguintes informações gerais para o módulo: *Nome, Fornecedor, Tipo, Número de Versão, Categorias, Número de Ordem, Descrição, Imagem*.

5.7.9.3. Configuração do Escravo EtherCAT

A seguir são listadas as principais opções de configuração para um Escravo EtherCAT, tal como definido no *Arquivo de Descrição do Dispositivo*.

5.7.9.3.1. EtherCAT Slave - General

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

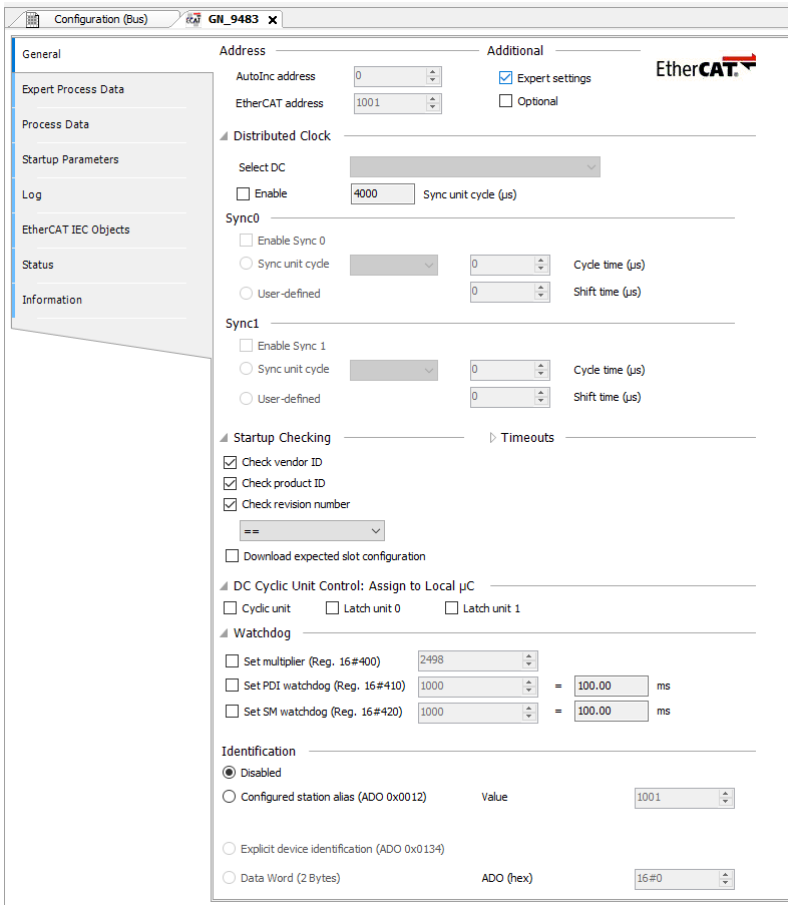


Figura 104: Diálogo de Configuração do Escravo EtherCAT

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 106: Configurações do Escravo EtherCAT

Notas:

AutoInc address: Este endereço é usado apenas durante a inicialização, quando o Mestre está atribuindo os endereços EtherCAT para os Escravos. Quando para este propósito, o primeiro telegrama percorre os Escravos, cada Escravo de leitura rápida aumenta seu *Endereço AutoInc* por 1. O Escravo com o endereço 0, finalmente, vai receber os dados.

Optional: Se um Escravo é declarado como *Opcional* nenhuma mensagem de erro será criada caso o dispositivo não exista no sistema de barramento. Se essa opção for ativada, um endereço de estação deve ser armazenado no dispositivo Escravo. Assim, um endereço *Pseudônimo de Estação* deve ser definido e gravado na EEPROM. Esta opção só está disponível se a opção *Autoconfiguração Mestre/Escravos* estiver ativada nas configurações do Mestre EtherCAT, e se esta função for suportada pelo Escravo EtherCAT.

Enable Distributed Clock: Se a funcionalidade *Relógio Distribuído* for ativada, o tempo de ciclo de troca de dados, exibido no campo *Unidade de Ciclo Sync(μs)* será determinado pelo *Tempo de Ciclo* do Mestre. Assim, o clock Mestre pode sincronizar a troca de dados dentro da rede. As definições para manusear a(s) unidade(s) de sincronização dependem do Escravo.

Enable Sync 0: Se essa opção for ativada, a unidade de sincronização *Sync0* é utilizada. Uma unidade de sincronização descreve um conjunto de dados do processo que são trocados de forma síncrona.

Sync unit cycle Sync 0: Se esta opção estiver ativada, o *Tempo de Ciclo* do Mestre, multiplicado pelo fator escolhido será utilizado como tempo de ciclo de sincronização para o Escravo. O campo *Tempo de Ciclo(μs)* mostra o tempo de ciclo atualmente definido.

Shift time: O Tempo de deslocamento descreve o tempo entre os eventos de sincronização (*Sync0, Sync1*) e as *Outputs Valid* ou *Input Latch*. Valor gravável, se o escravo suporta transferência de *Outputs Valid* ou *Input Latch*.

Enable Sync 1: Se essa opção for ativada, a unidade de sincronização *Sync1* é utilizada. Uma unidade de sincronização descreve um conjunto de dados do processo que são trocados de forma síncrona.

Sync unit cycle Sync1: Se essa opção for ativada, o *Tempo de Ciclo* do Mestre, multiplicado pelo fator escolhido será utilizado como tempo de ciclo de sincronização para o Escravo. O campo *Tempo de Ciclo(μs)* mostra o tempo de ciclo definido atualmente.

Check vendor ID and Check product ID: Por padrão, na inicialização do sistema o *ID do Fornecedor* e/ou o *ID do Produto* serão verificados contra as configurações atuais. Se for detectada alguma diferença, o barramento será interrompido e nenhuma ação adicional será executada. Isso serve para evitar o download de uma configuração errada. Estas opções têm a finalidade de desativar esta verificação.

SDO access: Por padrão não há tempo limite definido para a ação de envio de uma lista SDO ao iniciar o sistema. Porém, se houver a necessidade de verificar se esta ação excede certo tempo, este deve ser especificado neste campo.

I -> P: Por padrão não há tempo limite definido para a ação de troca do modo *Inicial* para o modo *Pré-Operacional*. Porém, se houver a necessidade de verificar se esta ação excede certo tempo, este deve ser especificado neste campo.

P -> S / S-> O: Por padrão não há tempo limite definido para a ação de troca dos modos *Pré-Operacional* para *Segurança-Operacional* e respectivamente de *Segurança-Operacional* para *Operacional*. Porém, se houver a necessidade de verificar se esta ação excede certo tempo, este deve ser especificado neste campo.

DC Cyclic Unit Control: Escolha a(s) opção(ões) desejada(s), relativa às funções de *Relógio Distribuído*, a fim de definir o que deve ser atribuído ao microprocessador local. O controle é feito no registro 0x980 do Escravo EtherCAT. As configurações possíveis são: *Unidade Cíclica, Unidade Latch 0, Unidade Latch 1*.

Enable: Se a definição de Opcional não estiver ativada, essa configuração pode ser ativada se for suportada pela descrição do dispositivo Escravo. Ele permite a atribuição direta de um endereço de pseudônimo, a fim de obter o endereço dos Escravos independente de sua posição dentro do barramento. Se a opção *Opcional* estiver ativada, esta caixa de seleção estará desativada.

5.7.9.3.2. EtherCAT Slave - Process Data

A guia *Process Data* do editor de configuração do Escravo EtherCAT mostra os dados do processo de entrada e de saída do Escravo, cada um definido por nome, tipo e índice do arquivo de descrição do dispositivo, como pode ser visto na figura abaixo.

As entradas selecionadas (a serem lidas) e saídas (a serem escritas) do dispositivo estarão disponíveis no diálogo de EtherCAT Slave - Module I/O Mapping como saídas e entradas para o CP, no qual as variáveis do projeto podem ser mapeadas.

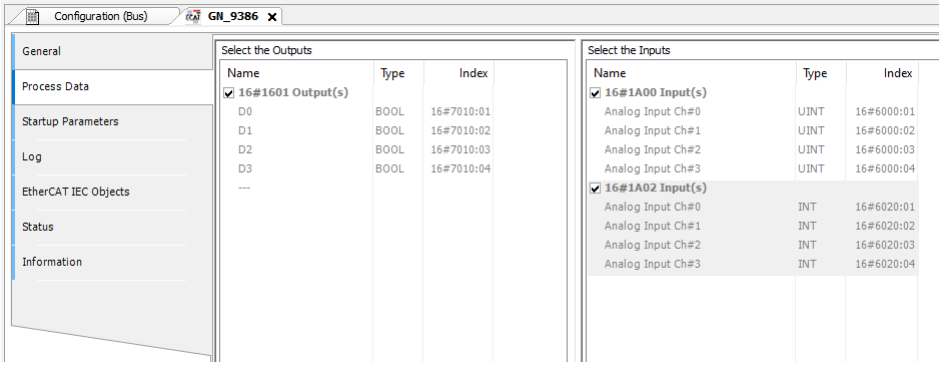


Figura 105: Diálogo Dados do Processo

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.

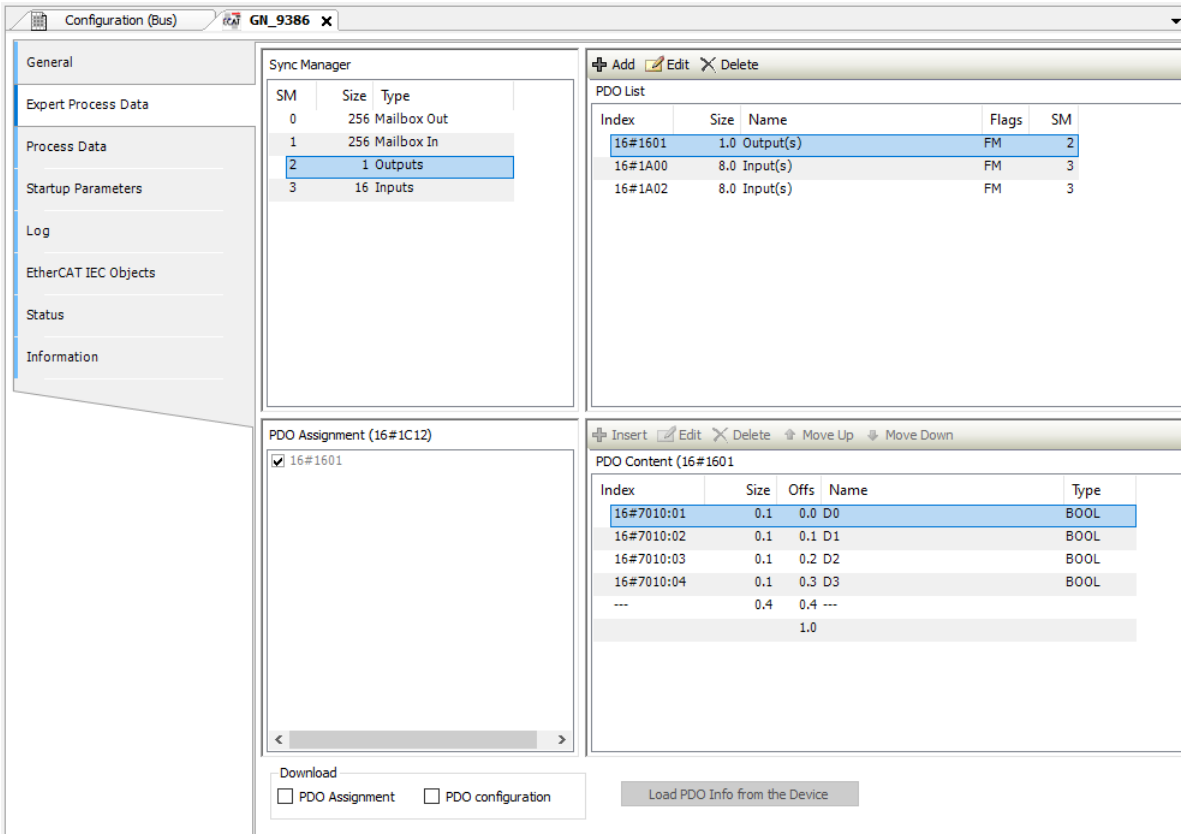


Figura 106: Diálogo Dados de Processo Avançados

Esta guia está dividida em quatro seções e duas opções:

- *Sync Manager*: Lista de SM *Sync Manager* com o tamanho dos dados e tipo de PDO.
- *PDO Assignment*: Lista de PDOs atribuídos ao *Sync Manager* selecionado. A caixa de seleção ativa o PDO e canais de E/S são criados. É parecido com as janelas de configuração de PDO simples. Aqui apenas PDOs podem ser ativados ou desativados.
- *PDO List*: Lista de todos os PDOs definidos no arquivo de descrição do dispositivo. PDOs individuais podem ser apagados, editados ou adicionados por meio da execução do respectivo comando no menu de contexto.
- *PDO Content*: Exibe o conteúdo do PDO selecionado na seção acima. As inscrições podem ser apagadas, editadas ou adicionadas por meio da execução do respectivo comando a partir do menu de contexto.
- *PDO Assignment*: Se ativado, um comando de escrita CoE será adicionado ao índice 0x1CXX para escrever a configuração PDO 0x16XX ou 0x1A00.
- *PDO configuration*: Se ativado, vários comandos da gravação CoE serão adicionados ao escrever o mapeamento PDO no Escravo.

ATENÇÃO

Se o Escravo não suportar a configuração PDO, o download desta pode resultar em um erro de Escravo. Esta função deve ser usada apenas por especialistas.

5.7.9.3.3. *EtherCAT Slave - Edit PDO List*

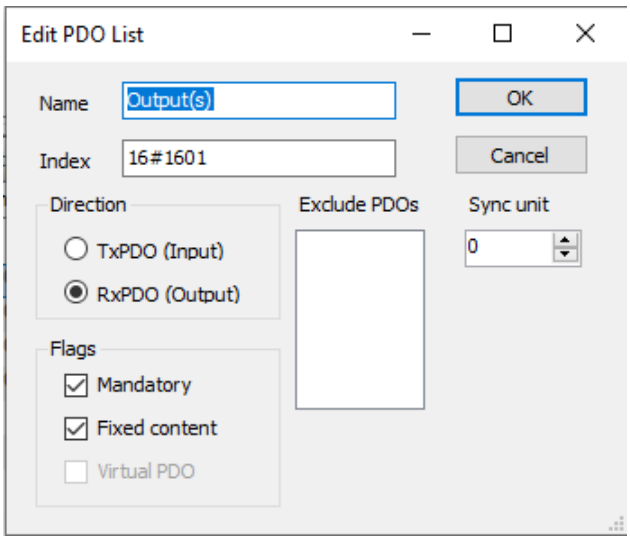


Figura 107: Diálogo Editar Lista PDO

Este diálogo é aberto através do menu de contexto da área Lista PDO, apresentada na Figura 106. Seguem algumas explicações das opções de configuração presentes neste diálogo.

- *Name:* Nome da entrada PDO.
- *Index:* Índice do PDO em edição.
- *TxPDO (Input):* Se ativado, o PDO será transferido do Mestre para o Escravo.
- *RxPDO (Output):* Se ativado, o PDO será transferido do Escravo para o Mestre.
- *Mandatory:* O PDO é necessário e não pode ser desmarcado na área PDO *Assignment*.
- *Fixed content:* O conteúdo do PDO é fixo e não pode ser alterado. Não é possível adicionar entradas no painel PDO *Fixed content*.
- *Virtual PDO:* Reservado para uso futuro.
- *Exlude PDOs:* É possível definir PDOs que podem, ou não, serem selecionados juntos ao PDO em edição na área PDO *Assignment*, ou na guia *Process data*. Se um PDO é marcado nesta lista este não poderá ser selecionado, ficando cinza, no ambiente PDO *Assignment* quando o PDO em edição for selecionado.
- *Sync unit:* ID do *Sync Manager* do PDO para o qual deve ser atribuído.

5.7.9.3.4. *EtherCAT Slave - Startup Parameters*

Na guia *Startup Parameters* podem ser definidos parâmetros para o dispositivo, que serão repassados pelos SDOs (*Service Data Objects*) ou IDN na inicialização do sistema. As opções presentes neste ambiente, além da possibilidade de acesso, variam de acordo com o Escravo EtherCAT utilizado e estão presentes no seu *Arquivo de Descrição do Dispositivo*.

5.7.9.3.5. *EtherCAT Slave - 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*.

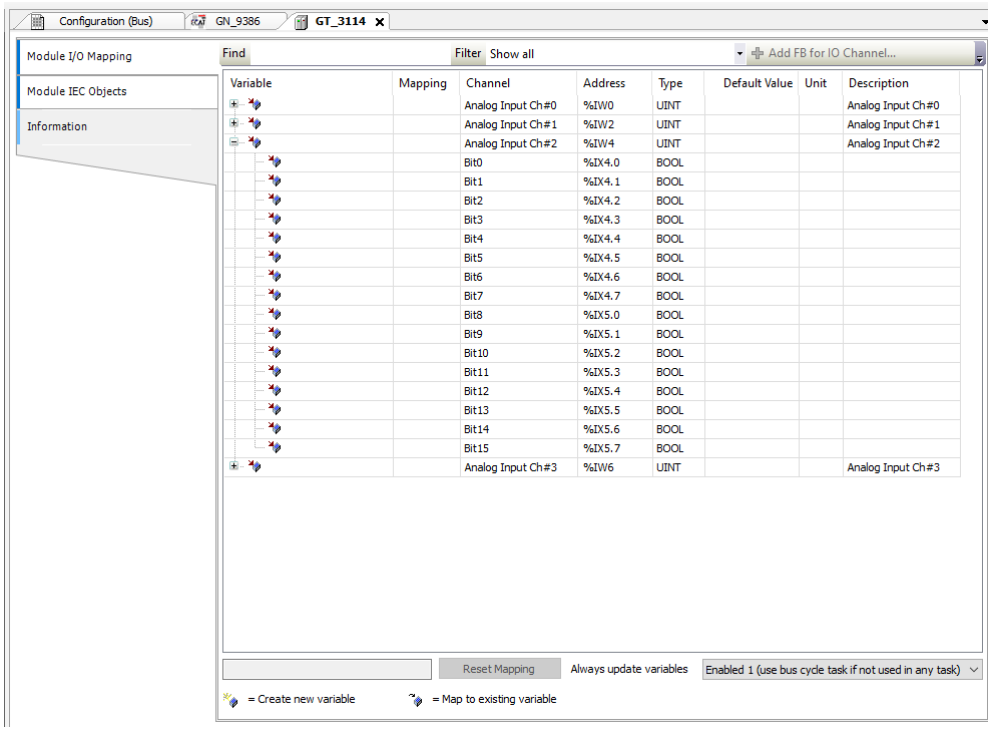


Figura 108: Diálogo Mapeamento de E/S Escravo

5.7.9.3.6. EtherCAT Slave - Status and Information

A guia *Status* do editor de configuração do Escravo EtherCAT fornece informações de estado (por exemplo, '*Executando*', '*Parado*') e mensagens de diagnóstico específicas do dispositivo e do sistema de barramento interno.

A guia *Information*, presente no editor de configuração do Escravo EtherCAT, exibe, caso disponíveis, as seguintes informações gerais para o módulo: *Nome, Fornecedor, Tipo, Número de Versão, Categorias, Número de Ordem, Descrição, Imagem*.

5.7.10. EtherNet/IP

O EtherNet/IP é um protocolo de arquitetura mestre-escravo, o qual consiste de um EtherNet/IP Scanner (o mestre) e um, ou mais, EtherNet/IP Adapters (o escravo).

O EtherNet/IP é um protocolo baseado na CIP (*Common Industrial Protocol*), o qual tem dois propósitos primários: O transporte de dados de controle-orientado associados com dispositivos de E/S e o transporte de outras informações relacionadas ao sistema sendo controlado, tais como parâmetros de configuração e diagnósticos. O primeiro é realizado por mensagens implícitas, enquanto o segundo é realizado através de mensagens explicitas.

O sistema em execução, 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.

5.7.10.1. Adaptador Ethernet

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

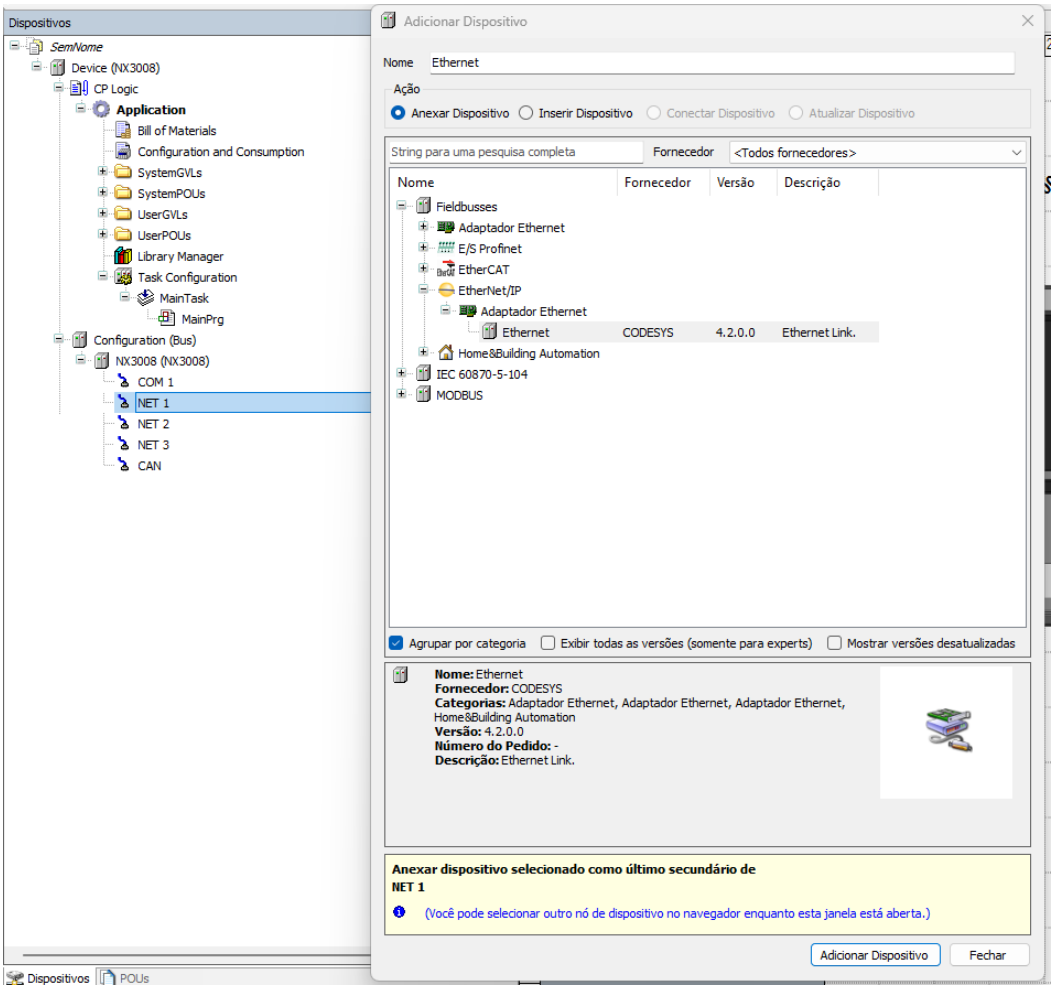


Figura 109: Adicionando um Adaptador Ethernet

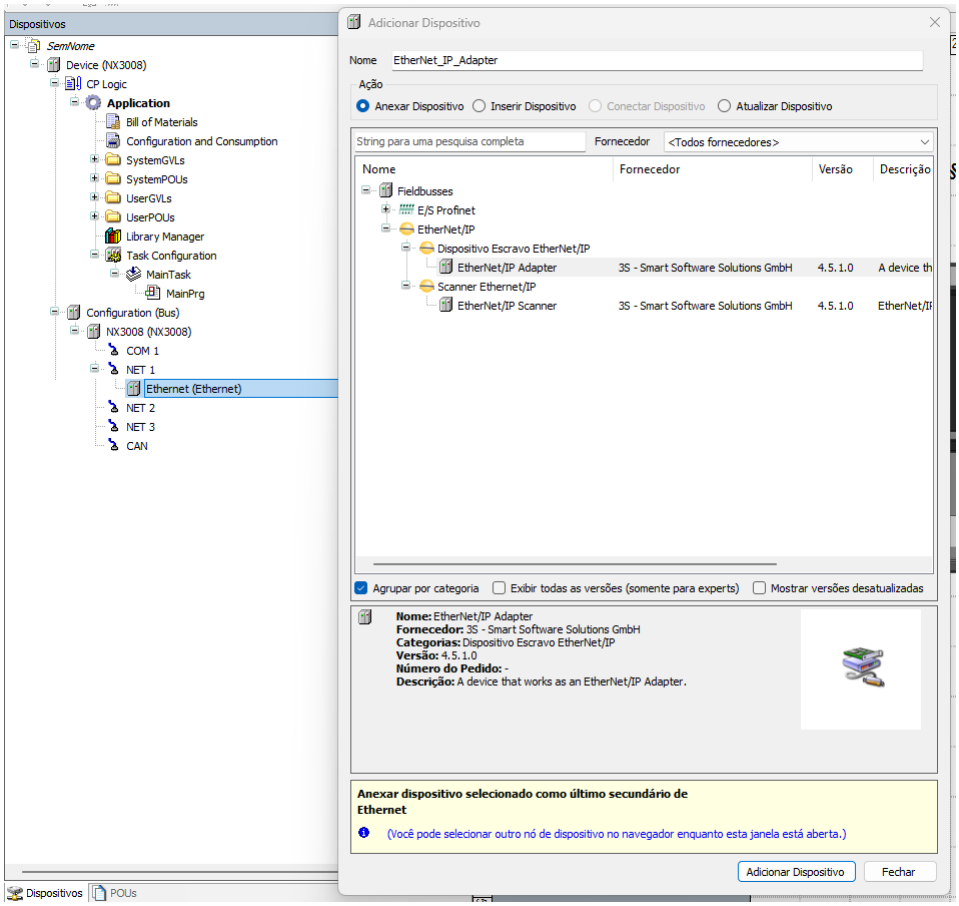


Figura 110: Adicionando um Adapter ou Scanner EtherNet/IP

5.7.10.2. Configuração do Scanner EtherNet/IP

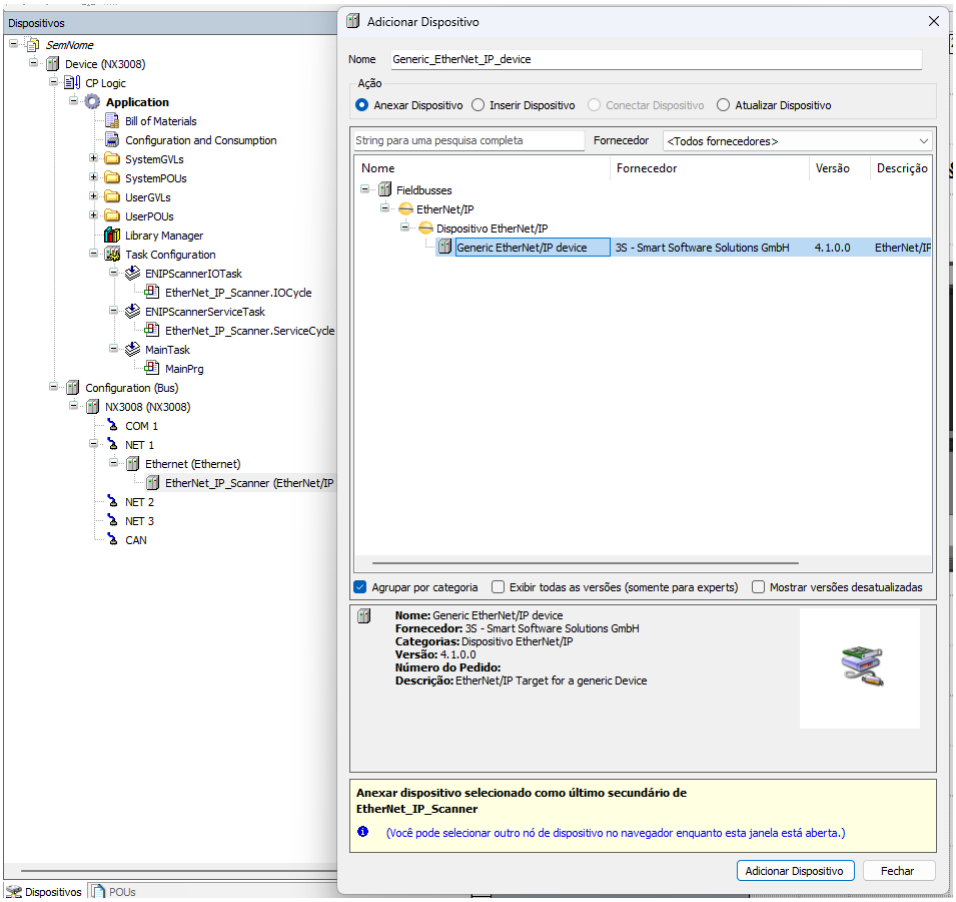


Figura 111: Adicionando um Adapter EtherNet/IP Sob o Scanner

5.7.10.2.1. Geral

Depois de abrir o Dispositivo declarado sob o Scanner é possível configurá-lo como for necessário. A primeira aba é *Geral*, nela é possível configurar o *Endereço IP* e o parâmetro *Chave Eletrônica*. Esses parâmetros devem ser marcados ou desmarcados se um Adapter sendo usado está instalado no Mastertool. De outra forma, se o Adapter usado é do tipo *Generic*, então os campos *ID do Fornecedor*, *Tipo do Dispositivo*, *Código do Produto*, *Revisão grande* e *Revisão pequena* devem ser preenchidos com as informações corretas do fabricante e as caixas marcadas tanto quanto necessário. A Altus por sua vez, tem um ID próprio, que é o ID *"1454"*.

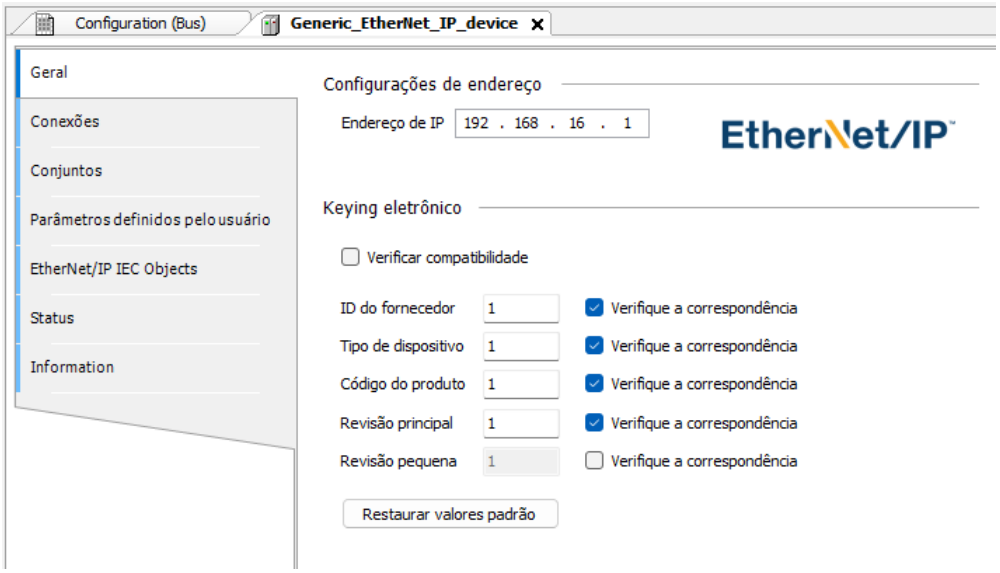


Figura 112: Aba Geral - EtherNet/IP

5.7.10.2.2. *Conexões*

A área superior da aba *Conexões* mostra uma lista de todas as conexões configuradas. Quando há uma conexão *Exclusive Owner* no arquivo EDS, ela é inserida automaticamente quando o Adapter é adicionado. Os dados de configuração para estas conexões podem ser mudados mais a baixo nesta janela.

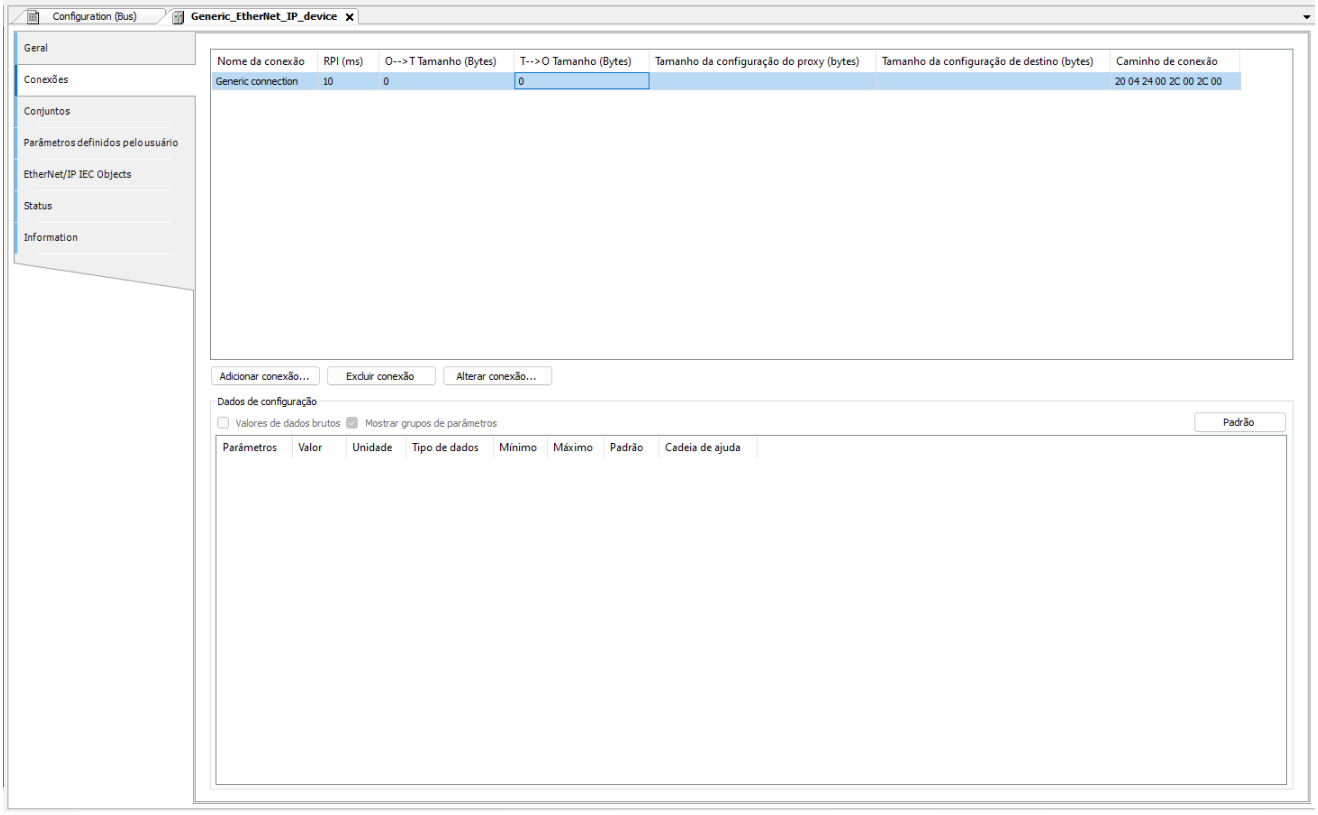


Figura 113: Aba Conexões - EtherNet/IP

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 107: 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.

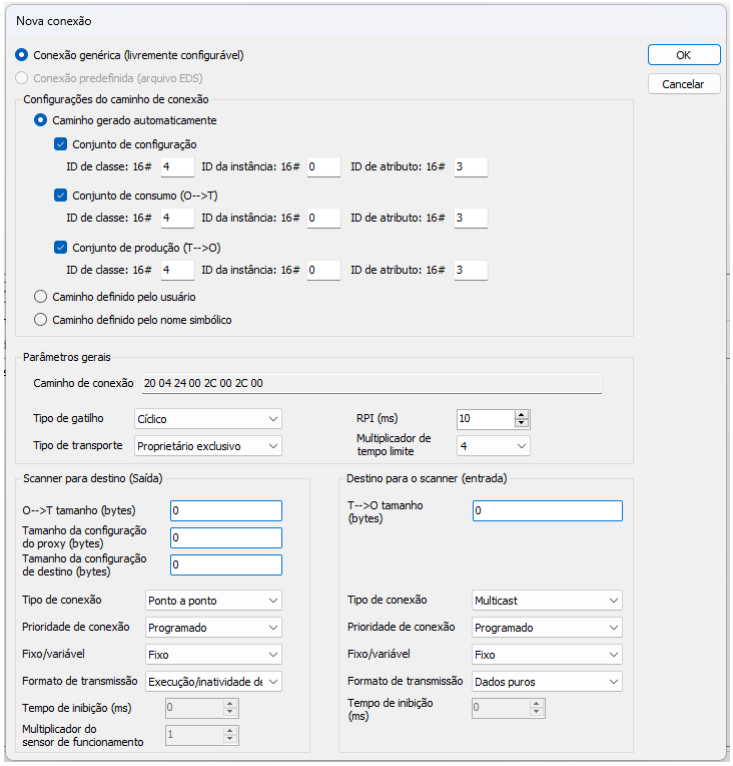


Figura 114: Janela de Novas Conexões EtherNet/IP

5.7.10.2.3. *Assemblies*

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

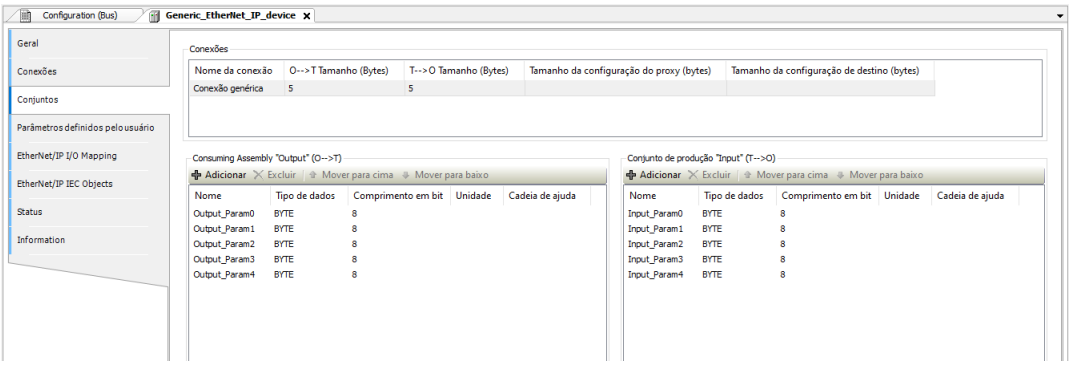


Figura 115: Aba Assemblies - EtherNet/IP

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 108: 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 109: Janela "Adicionar Entrada/Saída" – EtherNet/IP

5.7.10.2.4. Parâmetros Definidos pelo Usuário

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

ATENÇÃO

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

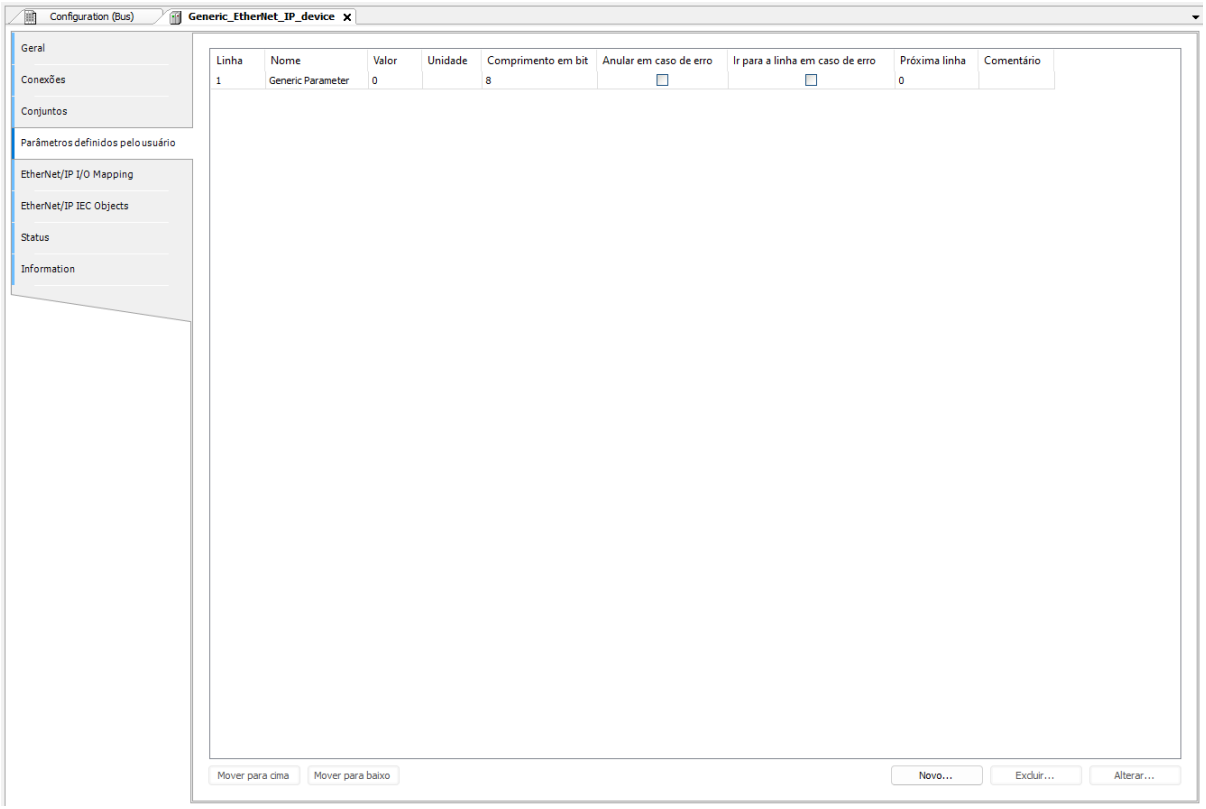


Figura 116: Aba Parâmetros Definidos pelo Usuário - EtherNet/IP

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

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

Tabela 110: Aba Parâmetros definidor pelo Usuário - EtherNet/IP

Note:

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

5.7.10.2.5. EtherNet/IP: Mapeamento de E/S

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

5.7.10.3. Configuração do Adapter EtherNet/IP

O Adapter EtherNet/IP requer módulos EtherNet/IP. Os módulos vão prover mapeamentos de E/S que podem ser manipulados pela aplicação do usuário pelos endereços %I ou %Q de acordo com sua configuração.

Novos Adapters podem ser instalados no Mastertool com os arquivos EDS. As opções de configuração podem divergir dependendo do arquivo de descrição do dispositivo do Adapter adicionado.

5.7.10.3.1. Geral

A primeira aba do EtherNet/IP Adapter é a aba *Geral*. Nela é possível configurar os parâmetros da *Chave Eletrônica*, utilizada no Scanner para verificar a compatibilidade. Nesta aba também é possível instalar o EDS do dispositivo diretamente no repositório de dispositivos do Mastertool ou exporta-lo.

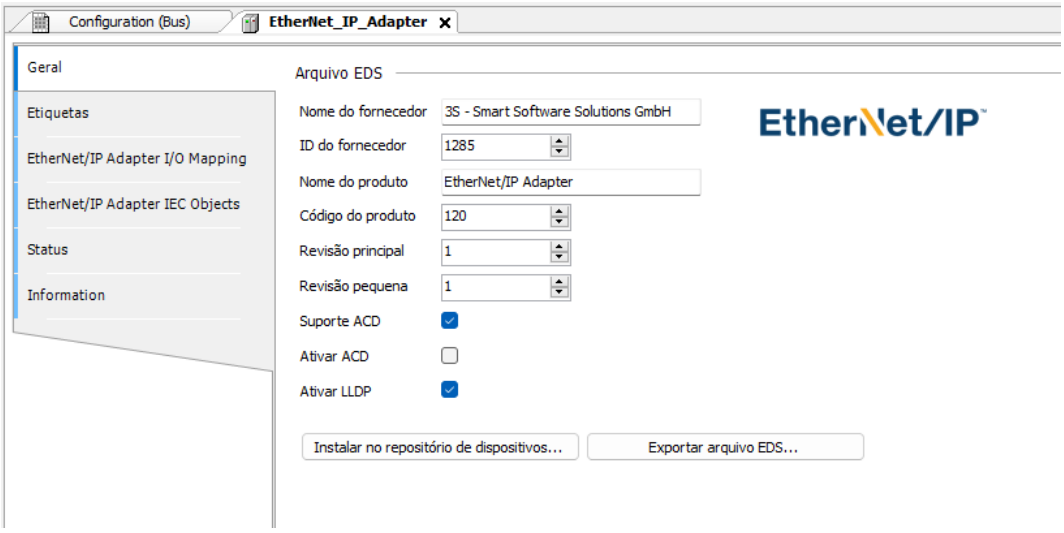


Figura 117: Aba Geral - EtherNet/IP_Adapter

5.7.10.3.2. *EtherNet/IP Adapter: Mapeamento de E/S*

Na aba *Mapeamento de E/S*, é possível configurar em qual tarefa de ciclo do barramento o Adapter irá executar.

5.7.10.4. Configuração do Módulo EtherNet/IP

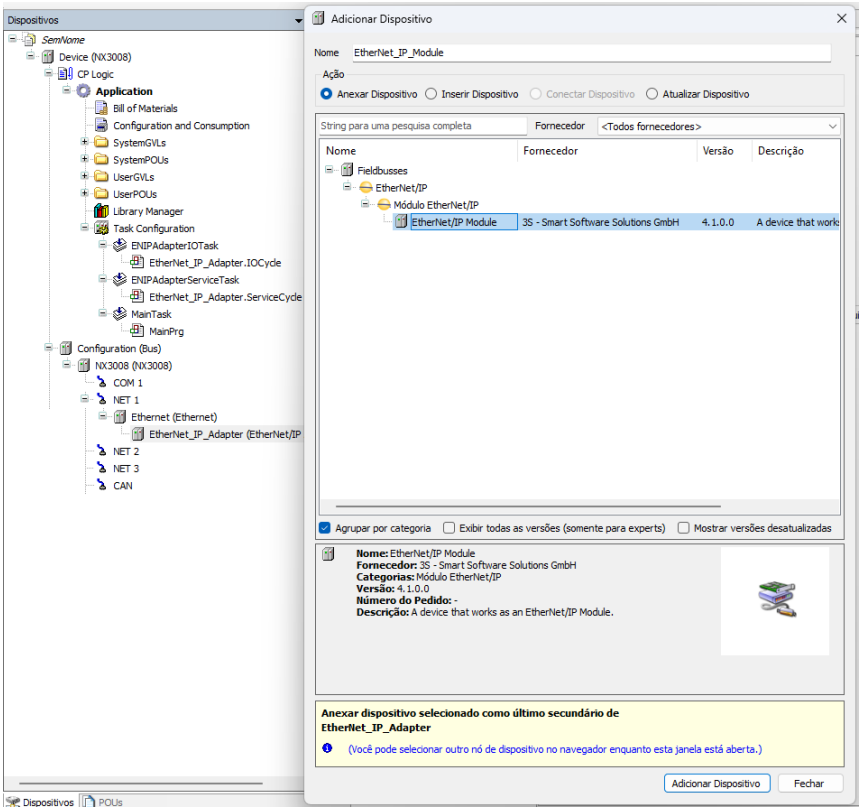


Figura 118: Adicionando um Módulo EtherNet/IP Abaixo do Adapter

5.7.10.4.1. *Assemblies*

Os parâmetros da aba *Geral* do módulo seguem as mesmas regras descritas nas tabelas 108 e 109.

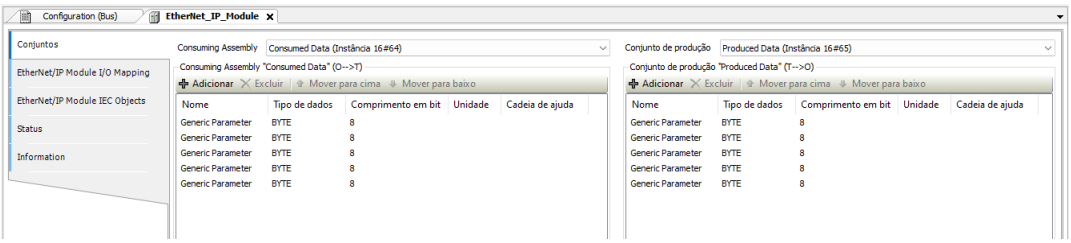


Figura 119: Aba Assemblies Módulo - EtherNet/IP

5.7.10.4.2. *EtherNet/IP Module: Mapeamento de E/S*

Aba *Mapeamento de E/S* mostra, na coluna *Variável*, o nome das instâncias de Adapter, automaticamente geradas. Desta forma, a instância pode ser acessada pela aplicação do usuário.

5.7.11. PROFINET Controller

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

5.8. Desempenho de Comunicação

5.8.1. Servidor MODBUS

Os dispositivos MODBUS configuráveis 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.

5.8.1.1. Interfaces Integradas da UCP

Para um dispositivo MODBUS Ethernet Servidor, podemos afirmar que ele é capaz de responder a um número x de requisições por segundo, ou seja, será capaz de transferir n bytes por segundo, dependendo do tamanho de cada requisição. Quanto menor for o ciclo da tarefa do Servidor MODBUS, maior será o impacto do número de conexões em sua taxa de resposta. Porém, para tempos de ciclo menores que 20 ms este impacto não é linear, devendo ser consultada a tabela abaixo para informações.

A tabela abaixo exemplifica o número de requisições respondidas por um Servidor MODBUS inserido em uma interface integrada 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

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

Tabela 111: Taxa de Comunicação de um Servidor MODBUS em uma Interface Integrada

ATENÇÃO

Os desempenhos de comunicação mencionados neste capítulo são exemplos utilizando uma UCP apenas com um dispositivo MODBUS TCP Servidor, sem nenhuma lógica na aplicação que possa atrasar a comunicação. Portanto, estes desempenhos devem ser tomados como máximos.

Já para tempos de ciclo iguais ou maiores do que 20 ms, o crescimento da taxa de respostas é linear, podendo ser calculada através da fórmula:

N = C x (1 / T)

Onde:

N é o número médio de respostas por segundo;

C é o número de conexões estabelecidas;

T é o intervalo da tarefa MODBUS em segundos.

Tomando como exemplo um Servidor MODBUS com uma conexão e um tempo de ciclo de 50 ms temos:

C = 1; T = 0,05 s;

N = 1 x (1 / (0,05))

N = 20

Ou seja, nesta configuração o Servidor MODBUS será capaz de responder, em média, 20 requisições por segundo.

Caso este valor obtido seja multiplicado pelo número de bytes em cada requisição, será obtida uma taxa de transferência de n bytes por segundo.

5.8.1.2. Interfaces Remotas

O desempenho de um dispositivo MODBUS Servidor em uma interface Ethernet remota é parecido com o desempenho nas interfaces integradas da UCP.

Porém, devido ao tempo de comunicação entre a UCP e os módulos, o máximo desempenho será limitado. Para apenas uma conexão aberta, o número de respostas é limitado em, no máximo, 18 respostas por segundo. Com mais conexões abertas, o número de respostas aumentará linearmente, exatamente conforme para as interfaces integradas, 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.

5.8.1.3. Taxa de Comunicação

A tabela abaixo exemplifica alguns valores calculados:

Número de Conexões	Requisições respondidas por segundo com tempo de ciclo da tarefa MODBUS em 5 ms, 40 ms e 55 ms (calculado)	Requisições respondidas por segundo com tempo de ciclo da tarefa MODBUS em 70 ms (calculado)	Requisições respondidas por segundo com tempo de ciclo da tarefa MODBUS em 100 ms (calculado)
1 Conexão	17,85	14,29	9,99
2 Conexões	35,69	28,58	19,98
4 Conexões	71,4	57,16	39,96
7 Conexões	124,95	100,03	69,93
10 Conexões	178,5	142,9	99,9

Tabela 112: Taxa de Comunicação de um Servidor MODBUS em uma Interface Integrada - Calculado

A seguinte tabela mostra o resultado de teste prático de performance:

Número de Conexões	Requisições respondidas por segundo com tempo de ciclo da tarefa MODBUS em 5 ms (medido)	Requisições respondidas por segundo com tempo de ciclo da tarefa MODBUS em 40 ms (medido)	Requisições respondidas por segundo com tempo de ciclo da tarefa MODBUS em 55 ms (medido)	Requisições respondidas por segundo com tempo de ciclo da tarefa MODBUS em 70 ms (medido)	Requisições respondidas por segundo com tempo de ciclo da tarefa MODBUS em 100 ms (medido)
1 Conexão	18,3	18,3	16,1	14,3	10,6
2 Conexões	36,6	27,6	32,7	28,6	20
4 Conexões	70,9	57,3	69,7	57	40,1
7 Conexões	120,7	97,8	114,1	99,9	70
10 Conexões	175,4	134,7	162,1	141,9	100,2

Tabela 113: Taxa de Comunicação de um Servidor MODBUS em uma Interface Integrada - Medido

5.8.2. Servidor OPC UA

O manual do OPC UA Server MU214609 analisa o desempenho da comunicação OPC UA com maiores detalhes, inclusive abordando o consumo de banda de comunicação Ethernet. Este manual também aborda conceitos sobre o funcionamento do protocolo OPC UA.

5.9. Desempenho do Sistema

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

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

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

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

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.

5.9.1. 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 114: 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.

5.9.2. 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.

5.10. Relógio RTC

As UCPs possuem um relógio interno que pode ser utilizado através da biblioteca *NextoStandard.lib*. Essa biblioteca é carregada automaticamente durante a criação de um novo projeto (para realizar o procedimento de inserção de uma biblioteca, consultar a seção Bibliotecas). A figura abaixo mostra os blocos e funções disponibilizados:

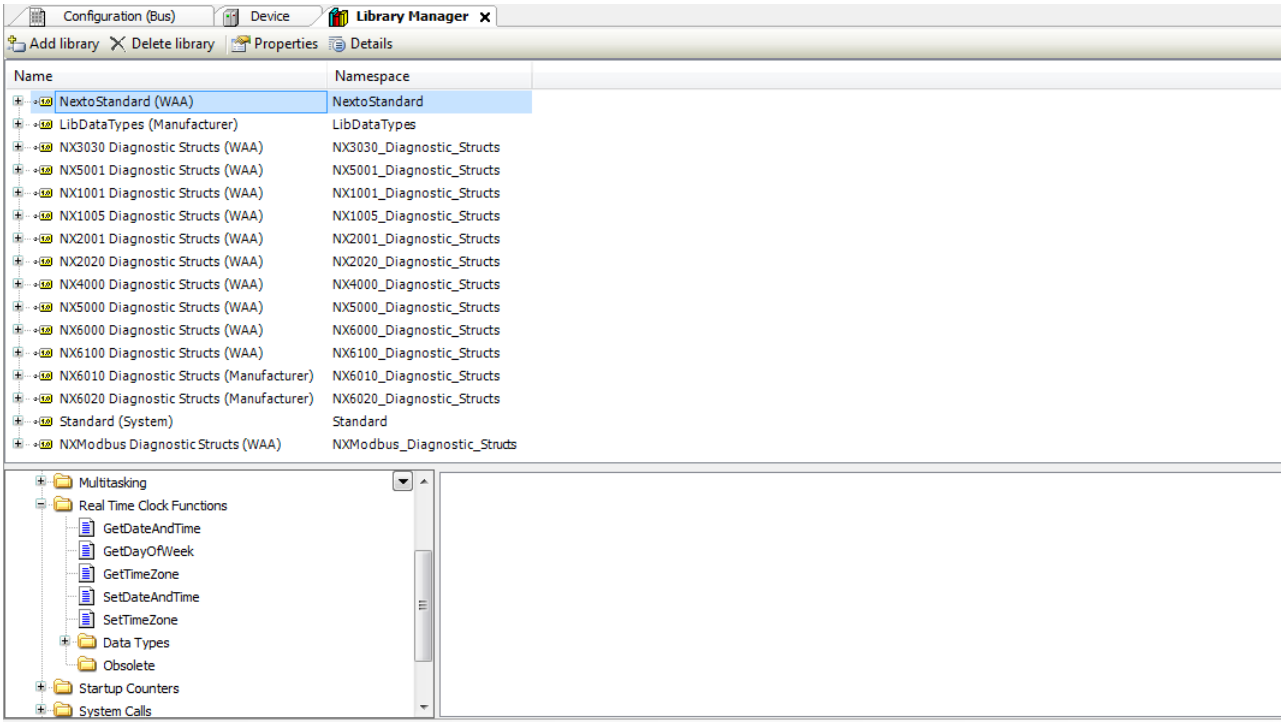


Figura 120: Blocos e Funções de Escrita e Leitura do Relógio

ATENÇÃO

Blocos funcionais para Leitura e Escrita do RTC, disponíveis em versões anteriores a 2.00 do Mastertool tornaram-se obsoletos da versão 2.00 em diante, os blocos que ficaram obsoletos são:
NextoGetDateAndTime, *NextoGetDateAndTimeMs*, *NextoGetTimeZone*, *NextoSetDateAndTime*, *NextoSetDateAndTimeMs* e *NextoSetTimeZone*.

5.10.1. Blocos Funcionais e Funções para Leitura e Escrita do RTC

Entre outros blocos funcionais e funções, existem alguns muito importantes utilizados para a leitura do relógio (*GetDateAndTime*, *GetDayOfWeek* e *GetTimeZone*) e para configurar novos valores de data e hora (*SetDateAndTime* e *SetTimeZone*). Estas funções utilizam sempre o horário local, isto é, levam em consideração o valor definido para o *Fuso Horário*.

Abaixo, descrevem-se os procedimentos utilizados para configurar as funções.

ATENÇÃO

Os blocos funcionais obsoletos para leitura e escrita do RTC (*NextoGetDateAndTime*, *NextoGetDateAndTimeMs*, *NextoSetDateAndTime* e *NextoSetDateAndTimeMs*) não podem ser utilizados na área de dados redundantes em projetos redundantes. Os mesmos devem ser utilizados em POUs não redundantes, como a POU *NonSkippedPrg*. Mais detalhes sobre o funcionamento da POU *NonSkippedPrg* podem ser encontrados em Programa NonSkippedPrg.

5.10.1.1. Funções de Leitura do RTC

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

5.10.1.1.1. GetDateAndTime

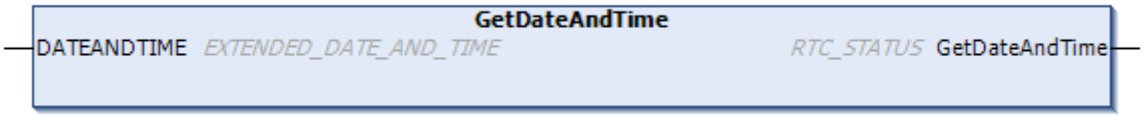


Figura 121: Leitura da Data e Hora

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 124.

Tabela 115: Parâmetros de Entrada GetDateAndTime

Parâmetros de saída	Tipo	Descrição
GETDATEANDTIME	RTC_STATUS	Retorna o estado de erro da função, ver Tabela 126.

Tabela 116: Parâmetros de Saída GetDateAndTime

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
Result : RTC_STATUS;
DATEANDTIME : EXTENDED_DATE_AND_TIME;
xEnable : BOOL;
END_VAR
-------------------------------------------------------------------------------
IF xEnable = TRUE THEN
Result := GetDateAndTime(DATEANDTIME);
xEnable := FALSE;
END_IF
```

5.10.1.1.2. GetTimeZone

A função a seguir faz a leitura das configurações de fuso horário, esta função está diretamente relacionada com o tempo de fuso horário configurado no serviço de sincronismo do SNTP:

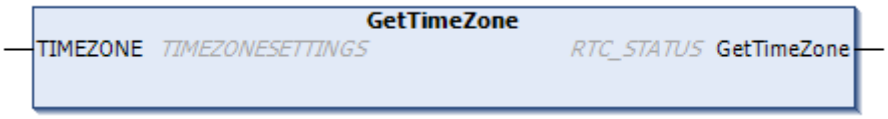


Figura 122: Leitura das Configurações de Fuso Horário

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

Tabela 117: Parâmetros de Entrada GetTimeZone

Parâmetros de saída	Tipo	Descrição
GetTimeZone	RTC_STATUS	Retorna o estado de erro da função, ver Tabela 126.

Tabela 118: Parâmetros de Saída GetTimeZone

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
GetTimeZone_Status  : RTC_STATUS;
TimeZone      : TIMEZONESETTINGS;
xEnable : BOOL;
END_VAR
-------------------------------------------------------------------------------
IF xEnable = TRUE THEN
GetTimeZone_Status := GetTimeZone(TimeZone);
xEnable := FALSE;
END_IF
```

5.10.1.1.3. GetDayOfWeek

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



Figura 123: Leitura do Dia da Semana

Parâmetros de saída	Tipo	Descrição
GetDayOfWeek	DAYS_OF_WEEK	Retorna o dia da semana. Ver Tabela 125.

Tabela 119: Parâmetros de Saída GetDayOfWeek

Quando chamada, a função lerá o dia da semana e preencherá a estrutura *DAYS_OF_WEEK*.
Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
DayOfWeek : DAYS_OF_WEEK;
END_VAR
--------------------------------------------------------------------------------
DayOfWeek := GetDayOfWeek();
```

5.10.1.2. Funções de Escrita do RTC

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

5.10.1.2.1. SetDateAndTime

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

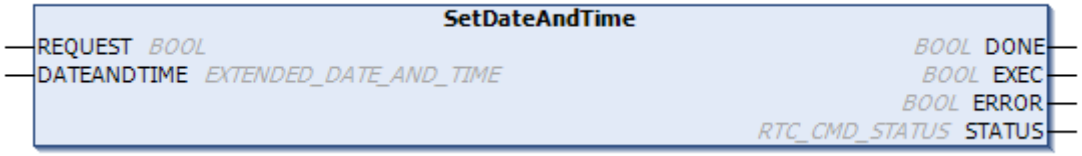


Figura 124: Ajuste de Data e Hora em Milissegundos

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 124.

Tabela 120: 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 126.

Tabela 121: Parâmetros de Saída SetDateAndTime

Quando ocorrer uma borda de subida na entrada *REQUEST*, o bloco funcional irá escrever o novo valor *DATEANDTIME* no relógio. Caso a escrita seja realizada com sucesso, a saída *DONE* será igual a *TRUE*. Caso contrário, a saída *ERROR* será igual a *TRUE* e o erro será apresentado na variável *STATUS*.

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
SetDateAndTime : SetDateAndTime;
xRequest : BOOL;
DateAndTime : EXTENDED_DATE_AND_TIME;
xDone : BOOL;
xExec : BOOL;
xError : BOOL;
xStatus : RTC_STATUS;
END_VAR
-------------------------------------------------------------------------------
IF xRequest THEN
  SetDateAndTime.REQUEST:=TRUE;
  SetDateAndTime.DATEANDTIME:=DateAndTime;
  xRequest:= FALSE;
END_IF
SetDateAndTime();
SetDateAndTime.REQUEST:=FALSE;
IF SetDateAndTime.DONE THEN
  xExec:=SetDateAndTime.EXEC;
  xError:=SetDateAndTime.ERROR;
  xStatus:=SetDateAndTime.STATUS;
END_IF
```

ATENÇÃO

Se o usuário tentar escrever valores de hora fora do intervalo do RTC, os valores serão convertidos para valores válidos, desde que não ultrapasse a faixa descrita na seção Limites de Operação do RTC. Por exemplo, se o usuário tentar escrever o valor 2000 ms, o mesmo será convertido para 2 segundos, se escrever o valor 100 segundos, o mesmo será convertido para 1 min e 40 segundos. Se escrever o valor de 30 horas, o mesmo será convertido para 1 dia e 6 horas, e assim por diante.

5.10.1.2.2. *SetTimeZone*

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

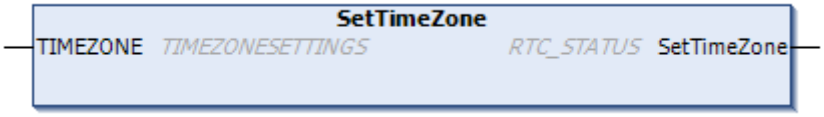


Figura 125: Ajuste de Fuso Horário

Parâmetros de entrada	Tipo	Descrição
TIMEZONE	TIMEZONESETTINGS	Estrutura com o valor de fuso horário a ser configurado. Ver Tabela 127.

Tabela 122: Parâmetros de Entrada SetTimeZone

Parâmetros de saída	Tipo	Descrição
SetTimeZone	RTC_STATUS	Retorna o erro ocorrido durante a leitura/configuração. Ver Tabela 126.

Tabela 123: Parâmetros de Saída SetTimeZone

Quando chamada, a função, irá configurar o valor de *TIMEZONE* como a nova configuração de fuso horário do sistema. O resultado da configuração é retornado pela função.

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
Status : RTC_STATUS;
TimeZone : TIMEZONESETTINGS;
xWrite : BOOL;
END_VAR
-------------------------------------------------------------------------------
//FB SetTimeZone
IF (xWrite = TRUE) THEN
Status := SetTimeZone(TimeZone);
  IF Status = RTC_STATUS.NO_ERROR THEN
    xWrite := FALSE;
  END_IF
END_IF
```

ATENÇÃO

Para realizar o acerto do relógio, devem-se utilizar valores de hora e datas dentro da faixa descrita na seção Limites de Operação do RTC, caso contrário, será reportado um erro através do parâmetro de saída *STATUS*. Para maiores detalhes do parâmetro de saída *STATUS*, consultar a seção RTC_STATUS.

5.10.2. Estruturas de Dados do RTC

Os blocos funcionais de leitura e configuração do RTC das UCPs da Série Nexto utilizam as seguintes estruturas de dados na sua configuração:

5.10.2.1. EXTENDED_DATE_AND_TIME

Esta estrutura é utilizada para armazenar a data do RTC quando utilizados os blocos funcionais para leitura/configuração da data com precisão de milissegundos e é descrita na tabela abaixo:

Estrutura	Tipo	Variável	Descrição
EXTENDED_DATE_AND_TIME	BYTE	byDayOfMonth	Armazena o dia do mês da data configurada.
	BYTE	ByMonth	Armazena o mês da data configurada.
	WORD	wYear	Armazena o ano da data configurada.
	BYTE	byHours	Armazena a hora da data configurada.
	BYTE	byMinutes	Armazena os minutos da data configurada.
	BYTE	bySeconds	Armazena os segundos da data configurada.
	WORD	wMilliseconds	Armazena os milissegundos da data configurada.

Tabela 124: Estrutura EXTENDED_DATE_AND_TIME

5.10.2.2. DAYS_OF_WEEK

Esta estrutura é utilizada para armazenar o dia da semana quando utilizada a função para leitura do dia da semana:

Enumerador	Valor	Descrição
DAYS_OF_WEEK	0	INVALID_DAY
	1	SUNDAY
	2	MONDAY
	3	TUESDAY
	4	WEDNESDAY
	5	THURSDAY
	6	FRIDAY
	7	SATURDAY

Tabela 125: Estrutura DAYS_OF_WEEK

5.10.2.3. RTC_STATUS

Este enumerador é utilizado para retornar o tipo de erro na configuração ou leitura do RTC e é descrito na tabela abaixo:

Enumerador	Valor	Descrição
	NO_ERROR (0)	Não há erro.
	UNKNOWN_COMMAND (1)	Comando desconhecido.
	DEVICE_BUSY (2)	Dispositivo está ocupado.
	DEVICE_ERROR (3)	Dispositivo com erro.

Enumerador	Valor	Descrição
RTC_STATUS	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 126: RTC_STATUS

5.10.2.4. TIMEZONESETTINGS

Esta estrutura é utilizada para armazenar o valor do fuso horário nas requisições de leitura/configuração dos blocos funcionais do RTC e é descrita na tabela abaixo:

Estrutura	Tipo	Variável	Descrição
TIMEZONESETTINGS	INT	iHour	Hora do fuso horário configurado.
	INT	iMinutes	Minuto do fuso horário configurado.

Tabela 127: TIMEZONESETTINGS

Nota:

Blocos funcionais de escrita e leitura de data e hora: Bibliotecas diferentes da *NextoStandard*, que tenham blocos funcionais ou funções que possam fazer acesso de leitura e escrita da data e hora no sistema, não são indicadas. A biblioteca *NextoStandard* possui as interfaces adequadas para escrever e ler a data e hora do sistema adequadamente e informar os diagnósticos corretos.

5.10.3. Limites de Operação do RTC

Os valores apresentados na Tabela 128 indicam o menor e o maior instante de tempo que o relógio RTC consegue representar. O valor mínimo corresponde à data e hora mais antiga que pode ser configurada, enquanto o valor máximo representa a data e hora mais futura válida. Esses limites devem ser respeitados ao definir ou ler o relógio do produto, garantindo que todas as funções de data e hora operem corretamente.

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

Tabela 128: Faixa de valores válidos para o Relógio RTC

5.11. Memória de Arquivos de Usuário

As UCPs da Série Nexto possuem uma área de memória destinada ao armazenamento de dados de uso geral, ou seja, o usuário poderá gravar diversos arquivos de qualquer formato na memória da UCP. Esta área de memória pode variar de acordo com o modelo de UCP utilizado (consultar Memória).

Para usar esta área, o usuário deverá acessar um projeto no software Mastertool e clicar na *Árvore de Dispositivos*, localizada à esquerda do programa. Deverá dar dois cliques sobre o item *Device* e, após selecionar a UCP na aba *Configurações de Comunicação* que será aberta, selecionar a aba *Arquivos* e clicar em *Atualizar*, tanto na coluna de arquivos do computador (esquerda), como na coluna de arquivos da UCP selecionada (direita), conforme mostram as indicações da figura abaixo.

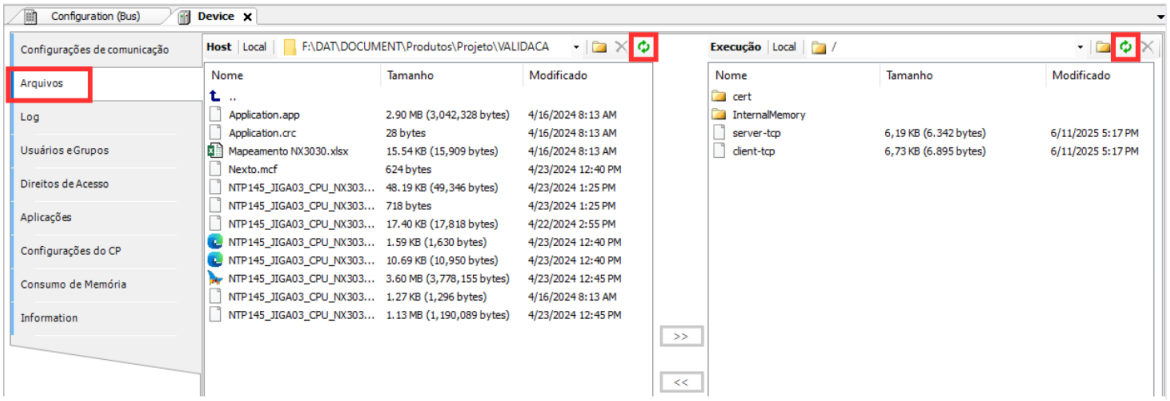


Figura 126: Acesso aos Arquivos de Usuário

Após atualizar a coluna de arquivos da UCP, será exibido o diretório raiz de arquivos armazenados na UCP e poderá ser selecionada a pasta para onde os arquivos serão transferidos. A pasta *"InternalMemory"* é uma pasta padrão, a ser utilizada para armazenar arquivos na memória interna da UCP, uma vez que não é possível transferir arquivos para o diretório raiz. Caso seja necessário, podem ser criadas outras pastas no diretório raiz ou subpastas dentro da pasta *"InternalMemory"*.

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

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

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

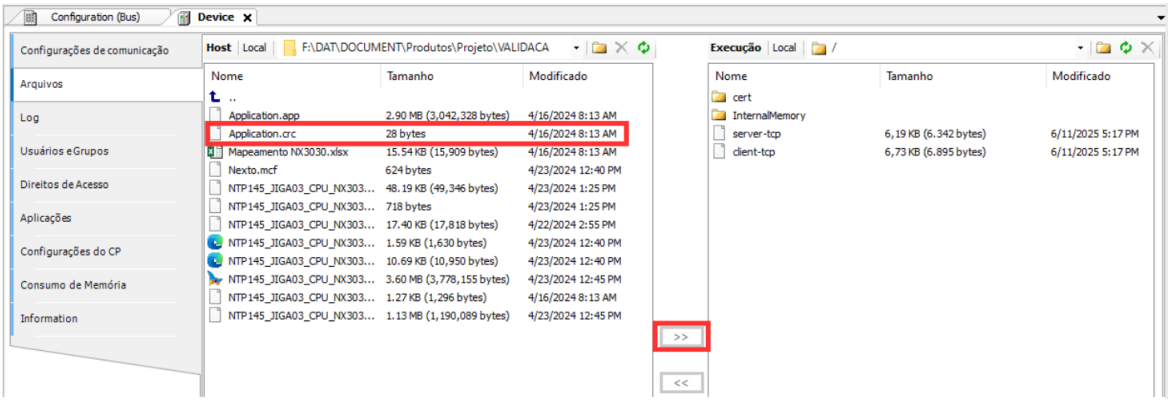


Figura 127: Transferindo Arquivos

ATENÇÃO

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

Caso seja necessário transferir documentos da UCP para o microcomputador em que está instalado o software Mastertool o usuário deve realizar um procedimento muito semelhante ao anterior, ou seja, selecionando o arquivo na coluna da direita e pressionar o botão "«", localizado no centro da tela.

Além disso, o usuário possui algumas opções de operação da área de armazenamento de arquivos, são elas:

- Novo diretório  : permite a criação de uma nova pasta na área de memória de usuário.
- Excluir item  : permite a exclusão de arquivos nos diretórios da área de memória de usuário.
- Atualizar  : permite atualizar, na tela do Mastertool, os arquivos presentes na memória de usuário e no microcomputador.

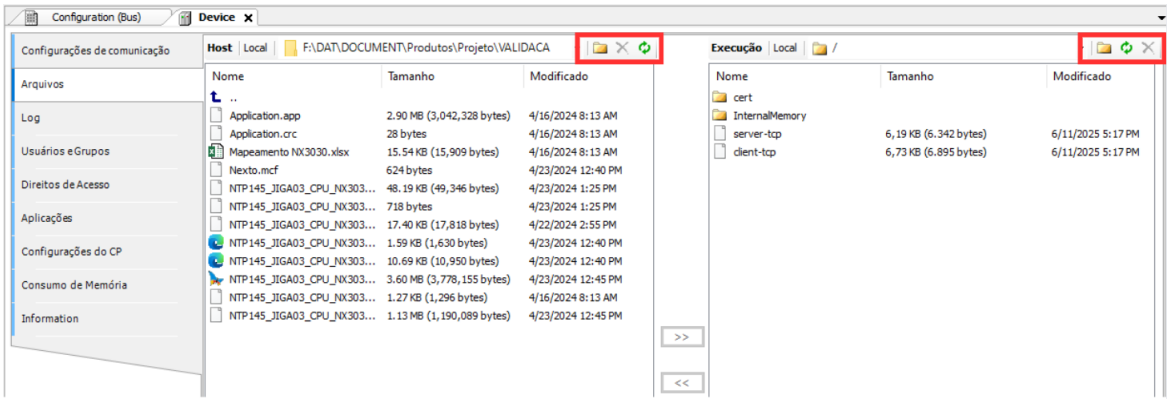


Figura 128: Opções de Utilização

ATENÇÃO

Para uma UCP em Modo Stop ou sem nenhuma aplicação, a taxa de transferência para a memória interna é de aproximadamente 150 Kbytes/s.

5.12. Cartão de Memória

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

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

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.

5.12.1. Configuração do Cartão de Memória

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

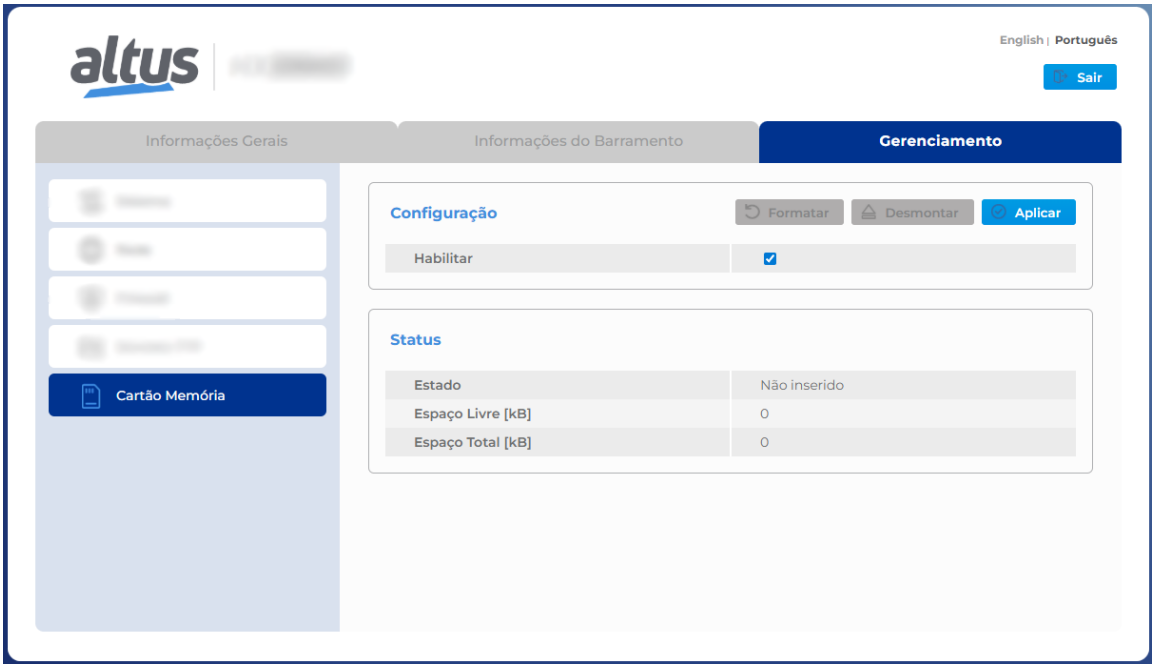


Figura 129: Cartão de Memória Página Inicial

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

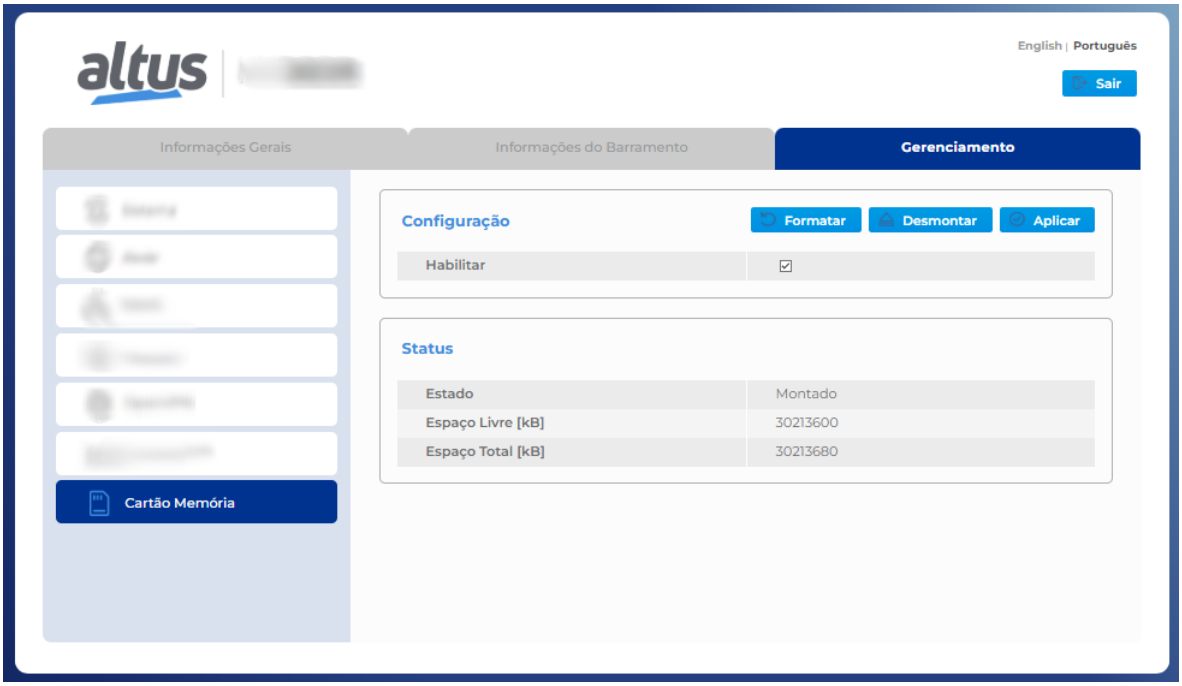


Figura 130: Cartão de Memória com Dispositivo Montado

ATENÇÃO

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

5.12.1.1. Formato Não Suportado

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

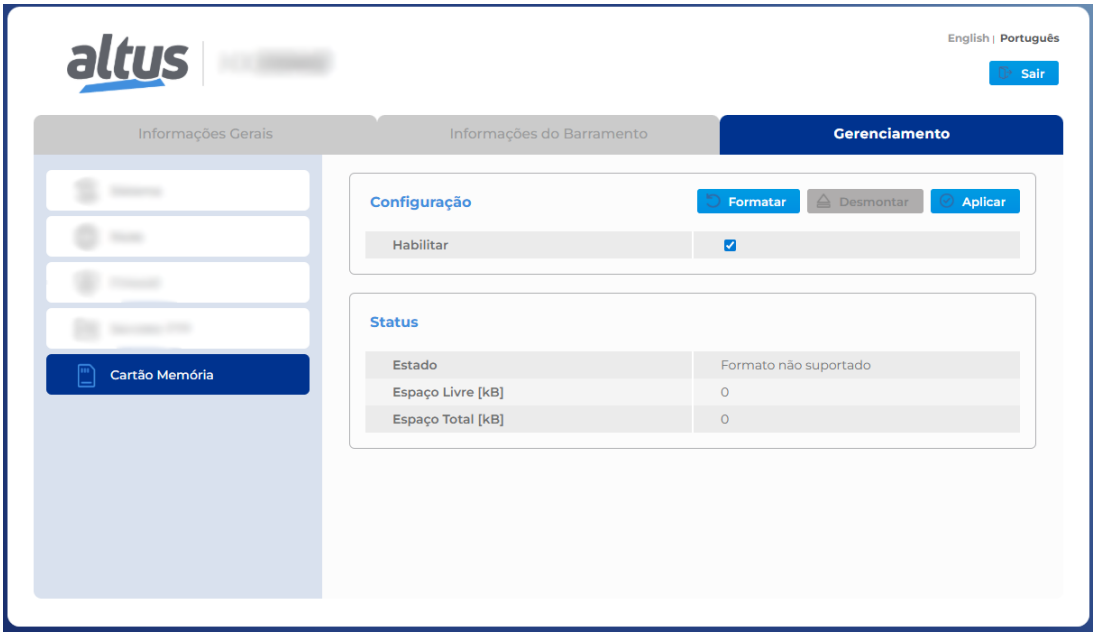


Figura 131: Mensagem de formato não suportado

Para formatar o dispositivo, o botão *Formatar* pode ser usado, conforme mostrado em Formatação do Cartão de Memória.

5.12.1.2. Formatação do Cartão de Memória

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

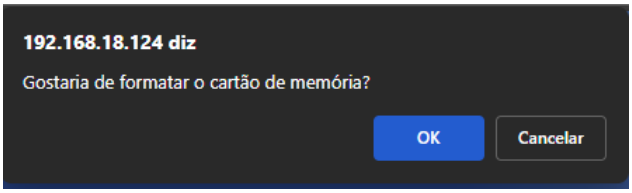


Figura 132: Mensagem Confirmação de Formatação

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

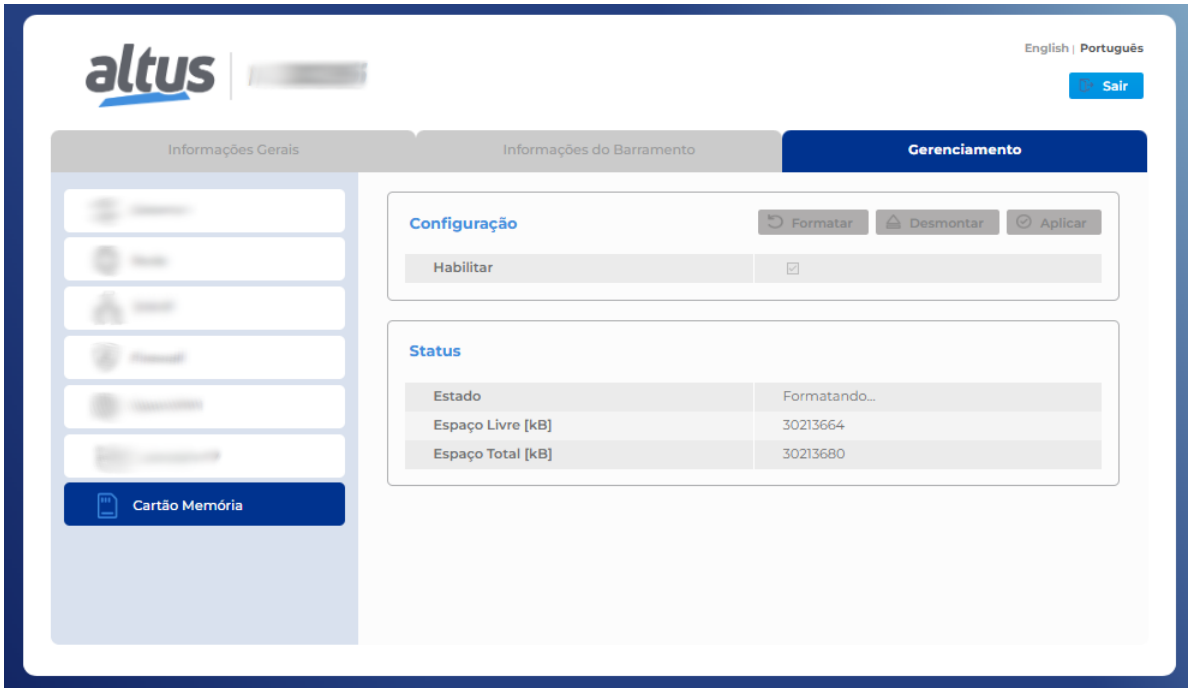


Figura 133: Cartão de Memória Formatando

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

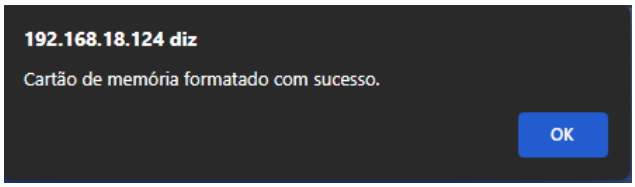


Figura 134: Mensagem de Formatação Concluída

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

5.12.1.3. Desmontagem do Cartão de Memória

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

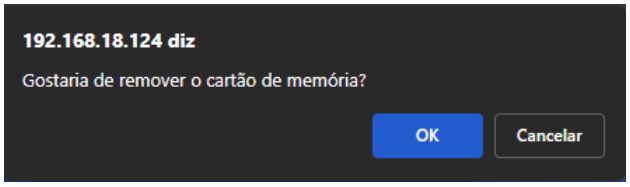


Figura 135: Mensagem Confirmação para Desmontar

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

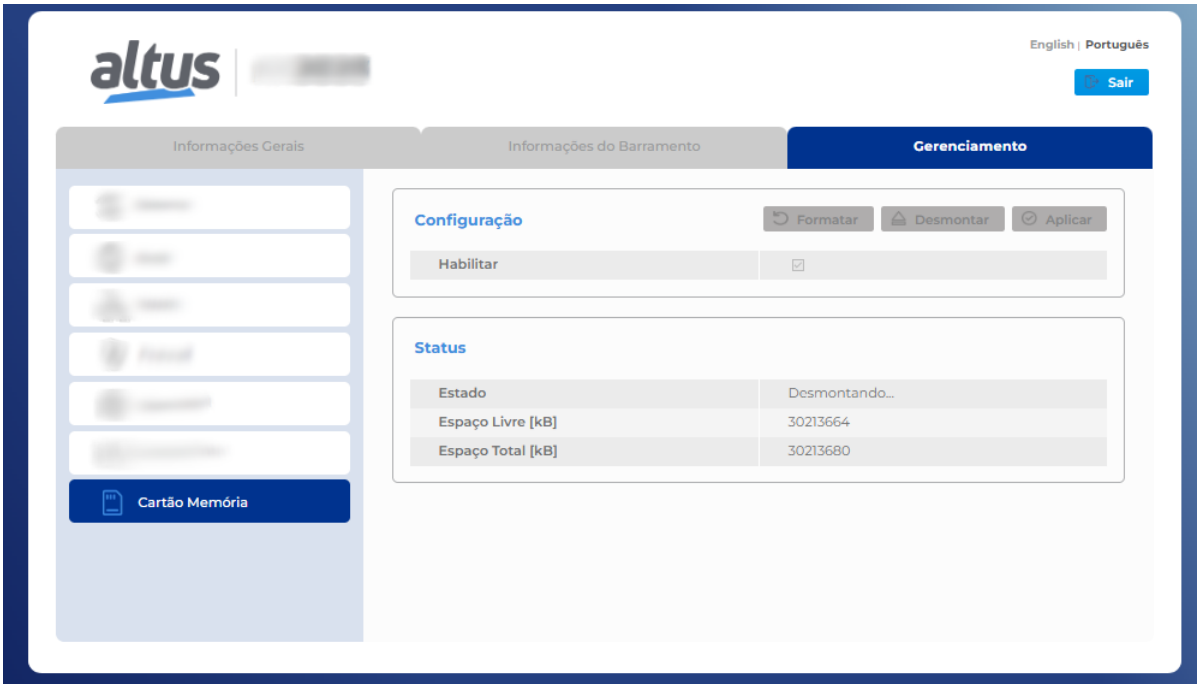


Figura 136: Cartão de Memória Desmontando

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

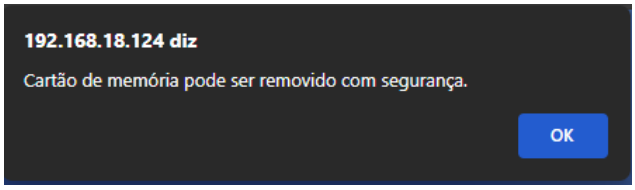


Figura 137: Mensagem de Desmontagem Concluída

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

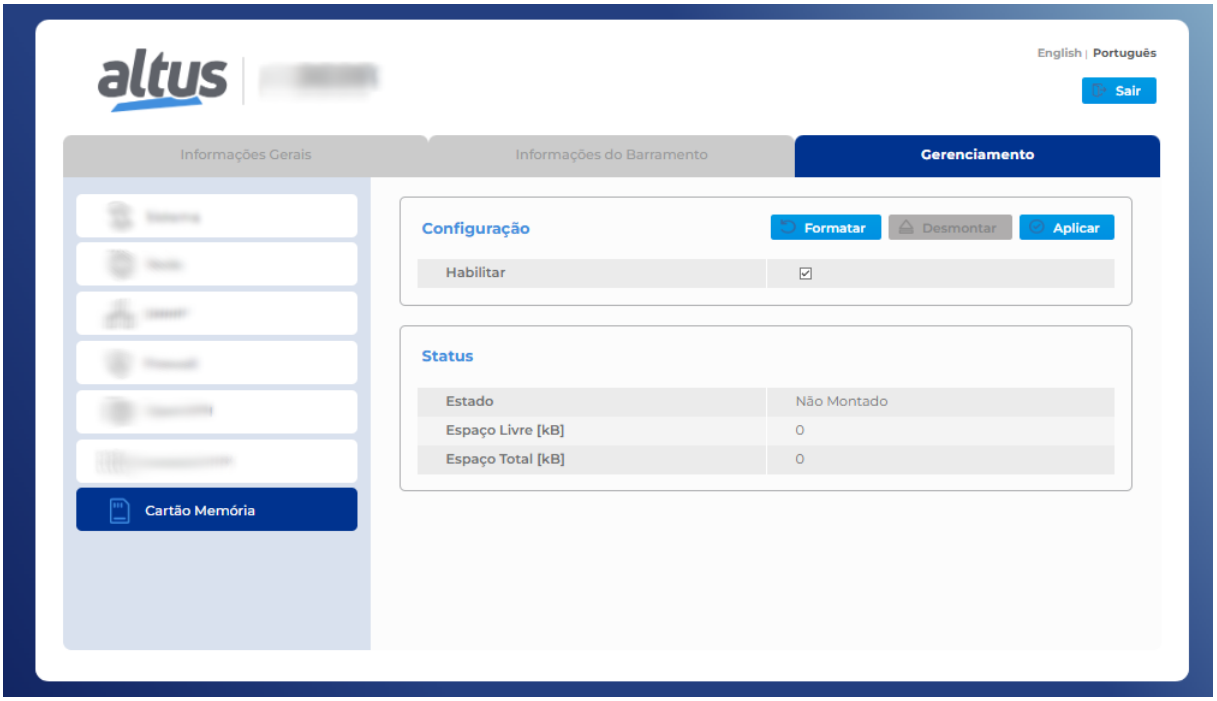


Figura 138: Cartão de Memória Não Montado

5.12.1.4. Gerenciamento da Interface do Cartão de Memória

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

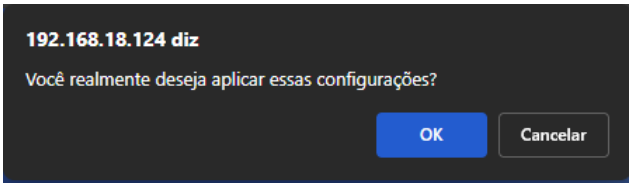


Figura 139: Mensagem Confirmação para Aplicar Configuração

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

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

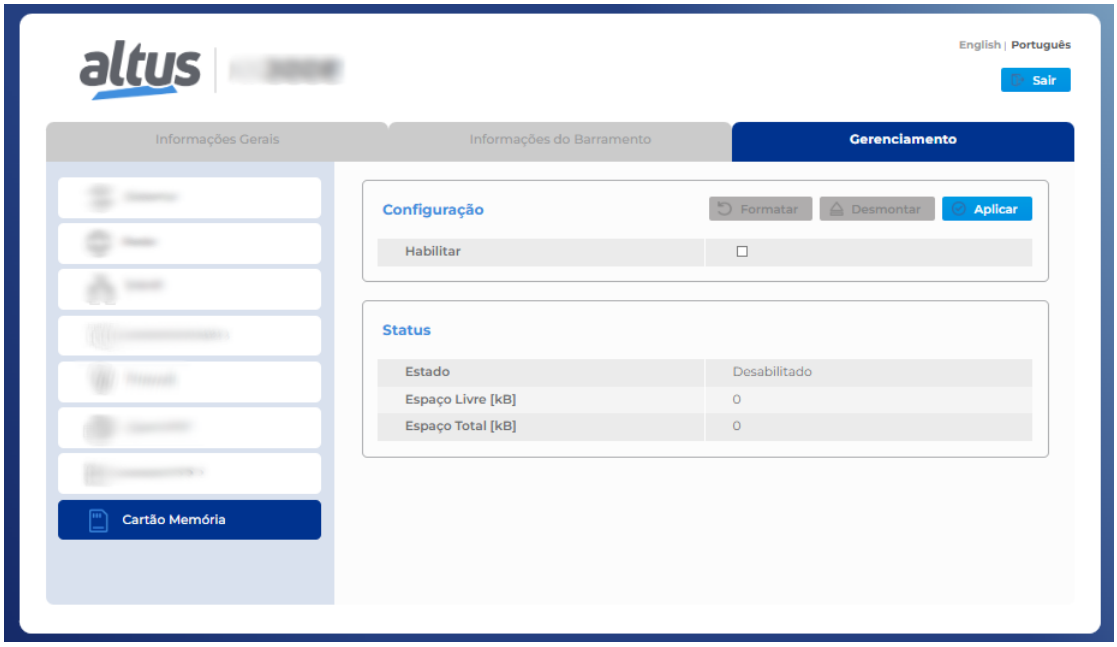


Figura 140: Cartão de Memória Interface Desabilitada

ATENÇÃO

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

5.12.1.5. Gerenciamento da Interface do Cartão de Memória via Aplicação

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

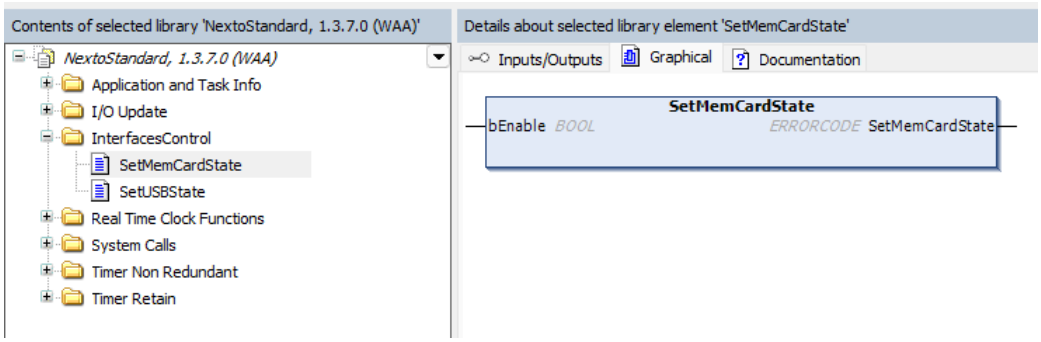


Figura 141: Informações de SetMemCardState no Library Manager

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

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

ATENÇÃO

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

5.13. 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
	VOLTAR	-	Retorna Nível
REDE	END. IP NET 1	-	Informativo
	MASCARA NET 1		Informativo
	END. IP NET 2		Informativo
	MASCARA NET 2		Informativo
	END. IP NET 3		Informativo
	MASCARA NET 3		Informativo
	END. IP NET 4		Informativo
	MASCARA NET 4		Informativo
	END. IP NET 5		Informativo
	MASCARA NET 5		Informativo
	END. IP NET 6		Informativo
	MASCARA NET 6		Informativo
	VOLTAR		Retorna Nível
REDUNDANCIA	IDENT. CP	-	Informativo
	ESTADO REM.	-	Informativo

Nível 1	Nível 2	Nível 3	Tipo
	SINCR. PROJ.	-	Informativo
	TROCAR ESTADO	CONFIRMA?	Configurável
	VOLTAR	-	Retorna Nível
CARTAO DE MEM.	DESMONTAR	-	Configurável
	FORMATAR	CONFIRMA?	Configurável
	VOLTAR	-	Retorna Nível
SOFTWARE	FIRMWARE	-	Informativo
	BOOTLOADER		Informativo
	COPROC		Informativo
	VOLTAR		Retorna Nível

Tabela 129: Níveis do Menu da UCP

Notas:

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

Redundância: O menu "*REDUNDANCIA*" somente estará disponível caso a UCP NX3035 esteja identificada como Redundante.

Conforme já mostrou a Tabela 129, entre as opções disponíveis para visualização e alteração, encontram-se os principais dados necessários ao usuário, como:

- Informações sobre os recursos de hardware:
  - TEMPERATURA – Temperatura interna da UCP (Ex.: 36 C 97 F)
  - CONTRASTE – Ajuste do contraste do visor frontal da UCP
  - DATA E HORA – Data e hora configuradas na UCP (Ex.: 2001.01.31 00:00)
- Alteração do idioma do menu da UCP:
  - PORTUGUES – Altera o idioma para Português
  - ENGLISH – Altera o idioma para Inglês
- Visualização de informações sobre a rede configurada no dispositivo:
  - END. IP NET 1 – Endereço IP (Ex.: 192.168.0.1)
  - MASCARA NET 1 – Máscara de subrede (Ex.: 255.255.255.0)
  - END. IP NET 2 – Endereço IP (Ex.: 192.168.0.2)
  - MASCARA NET 2 – Máscara de subrede (Ex.: 255.255.255.0)
  - END. IP NET 3 – Endereço IP (Ex.: 192.168.0.3)
  - MASCARA NET 3 – Máscara de subrede (Ex.: 255.255.255.0)
  - END. IP NET 4 – Endereço IP (Ex.: 192.168.0.4)
  - MASCARA NET 4 – Máscara de subrede (Ex.: 255.255.255.0)
  - END. IP NET 5 – Endereço IP (Ex.: 192.168.0.5)
  - MASCARA NET 5 – Máscara de subrede (Ex.: 255.255.255.0)
  - END. IP NET 6 – Endereço IP (Ex.: 192.168.0.6)
  - MASCARA NET 6 – Máscara de subrede (Ex.: 255.255.255.0)
- Acesso às informações de redundância do CP:
  - IDENT. CP – Informa a identificação do CP na redundância. Possíveis informações:
    - CPA
    - CPB
  - ESTADO REMOTO – Informa o estado do CP redundante remoto. Possíveis estados:
    - ATIVO

    - RESERVA
    - INATIVO
    - NAO CONFIG.
    - INICIANDO
    - INDISPONIVEL
  - SINCR. PROJETO – Informa o estado do sincronismo de projeto
    - CONECTADO – Sincronismo habilitado e link de redundância operante
    - NAO CONECTADO – Sincronismo habilitado e link de redundância inoperante
    - DESABILITADO – Sincronismo desabilitado
    - SINCRONIZANDO – Sincronismo em andamento
    - SINCRONIZADO – Sincronismo realizado com sucesso
  - TROCAR ESTADO – Permite a troca de estado do CP local. Possíveis opções:
    - RESERVA - Disponível quando o CP local está Ativo/Inativo, e solicita que o CP local mude para o estado Reserva.
    - INATIVO - Disponível quando o CP local está Reserva, e solicita que o CP local mude para o estado Inativo.
- 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)
  - COPROC – Versão do coprocessador da UCP (Ex.: 1.0.0.0)

A figura abaixo descreve um exemplo de como operar o menu das UCPs Nexto, através do procedimento de ajuste do contraste a partir da tela de *Status*. Além de facilitar a configuração, é possível identificar todos os níveis de tela e o tipo de pressionamento para navegar entre as mesmas, sendo que para modificar os outros parâmetros, como *Idioma* e e comandos na *Redundância*, 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.

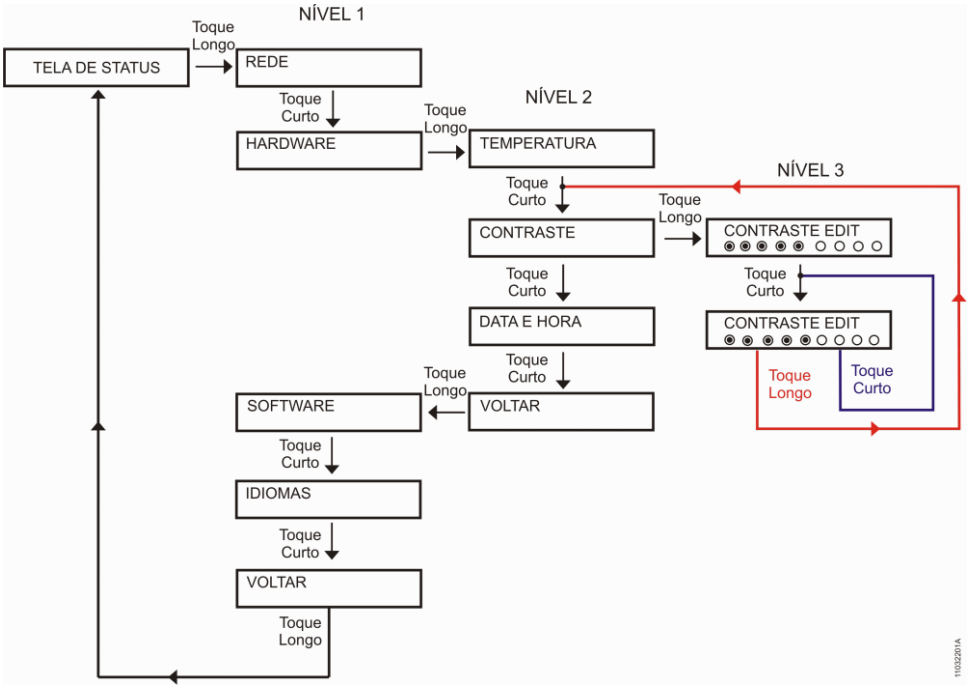


Figura 142: Ajuste do Contraste

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.

5.14. Blocos Funcionais e Funções

5.14.1. Blocos Funcionais Especiais para Comunicação Serial

Os blocos funcionais especiais para comunicação serial possibilitam o acesso local (COM) e também a portas seriais remotas (módulos de expansão). Dessa forma, o usuário poderá criar seus próprios protocolos e manipular as portas seriais como quiser, seguindo as linguagens da IEC 61131-3 disponíveis no software Mastertool. Os blocos estão disponíveis dentro da biblioteca *NextoSerial*, a qual deve ser adicionada ao projeto para que seja possível utilizá-los (para realizar o procedimento de inserção de uma biblioteca, consultar o Manual de Programação IEC 61131 – MP399048, capítulo Bibliotecas).

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

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

Tipo de dado	Opção	Descrição
SERIAL_BAUDRATE	BAUD9600	Lista todas as possibilidades da taxa de transmissão (bits por segundo)
	BAUD19200	
	BAUD38400	
	BAUD57600	
	BAUD115200	
SERIAL_DATABITS	DATABITS_8	Lista todas as possibilidades de bits de dado.
SERIAL_MODE	NORMAL_MODE	Modo normal de operação da comunicação serial.
	EXTENDED_MODE	Modo estendido de operação da comunicação serial, no qual são fornecidas informações sobre o frame de dados recebido.
SERIAL_PARAMETERS	Define todos os parâmetros de configuração da porta serial:	
	BAUDRATE	Definido em SERIAL_BAUDRATE.
	DATABITS	Definido em SERIAL_DATABITS.
	STOPBITS	Definido em SERIAL_STOPBITS.
	PARITY	Definido em SERIAL_PARITY.
	UART_RX_THRESHOLD	Quantidade de bytes que devem ser recebidos para gerar uma nova interrupção na UART. Valores baixos fazem o TIMESTAMP mais preciso quando o MODO ESTENDIDO é utilizado e minimiza os erros de overrun. No entanto, valores baixos podem causar muitas interrupções que podem retardar a UCP.
	MODE	Definido em SERIAL_MODE.

Tipo de dado	Opção	Descrição
	ENABLE_RX_ON_TX	Quando verdadeiro, todos os bytes recebidos durante a transmissão serão descartados em vez de ir para a fila de RX. Utilizado para desabilitar a operação full-duplex na interface RS-422.
SERIAL_PARITY	PARITY_NONE	Lista todas as possibilidades de paridade.
	PARITY_ODD	
	PARITY_EVEN	
SERIAL_PORT	COM 1	Lista todas as portas seriais disponíveis.
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 (Para mais informações ver notas da seção COM 1).
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).

Tipo de dado	Opção	Descrição
	RX_ANY_ERRORS	Soma dos 5 últimos contadores de erros (frame, paridade, interrupção, overrun RX FIFO, overrun fila RX).
	RX_REMAINING	Número de caracteres remanescentes na fila RX.
SERIAL_STATUS	Lista os códigos de erro críticos que podem ser retornados pelos blocos funcionais da serial. Cada bloco retorna erros específicos, os quais serão mencionados na descrição dos mesmos:	
	NO_ERROR	Não existem erros.
	ILLEGAL_*	Retorna os parâmetros com valores inválidos ou fora da faixa: - SERIAL_PORT - SERIAL_MODE - BAUDRATE - DATA_BITS - PARITY - STOP_BITS - UART_RX_THRESHOLD - TIMEOUT - TX_BUFF_LENGTH - 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.
	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_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.
	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	

Tabela 130: Tipos de dados Blocos Funcionais Serial

5.14.1.1. 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 143: 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 131: 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.

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 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 132: 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,
UART_RX_THRESHOLD := 8,
MODE :=NORMAL_MODE,
ENABLE_RX_ON_TX := FALSE);
Status: SERIAL_STATUS;
END_VAR
//ENTRADAS:
Config.REQUEST := TRUE;
Config.PORT := Port;
Config.PARAMETERS := Parameters;
//FUNÇÃO:
Config();
//SAÍDAS:
Config.DONE;
Config.EXEC;
Config.ERROR;
Status := Config.STATUS;    //Caso seja necessário tratar o erro.
```

5.14.1.2. SERIAL_GET_CFG

Esse bloco funcional é utilizado para capturar as configurações da porta serial desejada.



Figura 144: 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 133: Parâmetros de Entrada SERIAL_GET_CFG

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

Tabela 134: 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.
```

5.14.1.3. 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 145: 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 135: 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 136: 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;
```

5.14.1.4. 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 146: 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 137: 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 138: 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.
```

5.14.1.5. 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 147: 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 139: 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.

Parâmetros de saída	Tipo	Descrição
ERROR	BOOL	Essa variável é verdadeira quando o bloco funcional concluir a sua execução com algum erro, caso contrário, ela é falsa. Está vinculada à variável DONE, pois seu estado é exibido após a conclusão do bloco funcional.
STATUS	SERIAL_STATUS	Caso a variável ERROR seja verdadeira, a estrutura STATUS exibirá o erro encontrado na execução do bloco. Os estados, já descritos no tipo de dado SERIAL_STATUS, possíveis são: - NO_ERROR - ILLEGAL_SERIAL_PORT - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - NOT_CONFIGURED

Tabela 140: 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.
```

5.14.1.6. 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 148: Bloco para Ler Valores do Buffer de Recepção

Parâmetros de entrada	Tipo	Descrição
REQUEST	BOOL	Essa variável, quando verdadeira, habilita o uso do bloco funcional.
PORT	SERIAL_PORT	Seleciona a porta serial, conforme descrito no tipo de dado SERIAL_PORT.
RX_BUFFER_ POINTER	POINTER TO BYTE	Ponteiro de um array de bytes para receber os valores do buffer.
RX_BUFFER_ LENGTH	UINT	Especifica o número de caracteres esperados no array de bytes. Caso estejam disponíveis mais bytes do que o esperado, somente a quantidade esperada vai ser lida no array de bytes, sendo que os demais serão deixados na fila RX (tamanho máximo igual a 1024 caracteres).
RX_TIMEOUT	UINT	Especifica o time-out para receber a quantidade de caracteres esperados. Caso ele seja menor do que a quantidade de bytes a receber, será indicado RX_TIMEOUT_ERROR no parâmetro de saída STATUS. Quando o valor especificado, em ms, for igual a zero, a função irá retornar os dados presentes no buffer.

Tabela 141: 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

Parâmetros de saída	Tipo	Descrição
RX_RECEIVED	UINT	Retorna o número de caracteres recebidos. Esse número pode estar entre zero e o valor configurado em RX_BUFFER_LENGTH. Caso seja menor, um erro será indicado pelo bloco funcional.
RX_REMAINING	UINT	Retorna o número de caracteres que ainda estão na fila RX depois que o bloco funcional foi executado.

Tabela 142: 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;
```

5.14.1.7. SERIAL_RX_EXTENDED

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

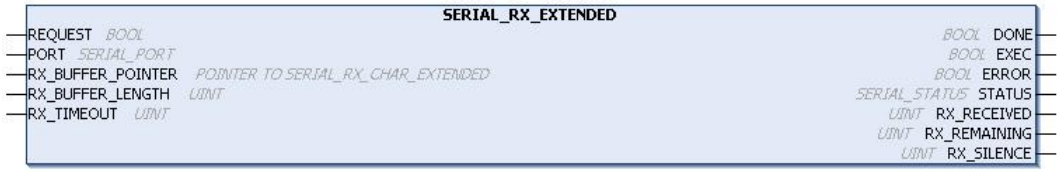


Figura 149: Bloco para Leitura do Buffer de Recepção

Parâmetros de entrada	Tipo	Descrição
REQUEST	BOOL	Essa variável, quando verdadeira, habilita o uso do bloco funcional.
PORT	SERIAL_PORT	Seleciona a porta serial, conforme descrito no tipo de dado SERIAL_PORT.
RX_BUFFER_ POINTER	POINTER TO SERIAL_RX_CHAR _EXTENDED	Ponteiro de um array de SERIAL_RX_CHAR_EXTENDED para receber os valores do buffer.
RX_BUFFER_ LENGTH	UINT	Especifica o número de caracteres esperados no array de SERIAL_RX_CHAR_EXTENDED. Caso estejam disponíveis mais bytes do que o esperado, somente a quantidade esperada vai ser lida no array, sendo que os demais serão deixados na fila RX (tamanho máximo igual a 1024 caracteres).
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 143: 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.

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_RX_BUFF_LENGTH - RX_TIMEOUT_ERROR - FB_SERIAL_RX_EXTENDED_NOT_ALLOWED - NOT_CONFIGURED
RX_RECEIVED	UINT	Retorna o número de caracteres recebidos. Esse número pode estar entre zero e o valor configurado em RX_BUFFER_LENGTH. Caso seja menor, um erro será indicado pelo bloco funcional.
RX_REMAINING	UINT	Retorna o número de caracteres que ainda estão na fila RX depois que o bloco funcional foi executado.
RX_SILENCE	UDINT	Retorna o tempo de silêncio na linha RX, medido desde o fim do último caractere recebido. A unidade de tempo é 1 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 144: 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;
```

5.14.1.8. 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).

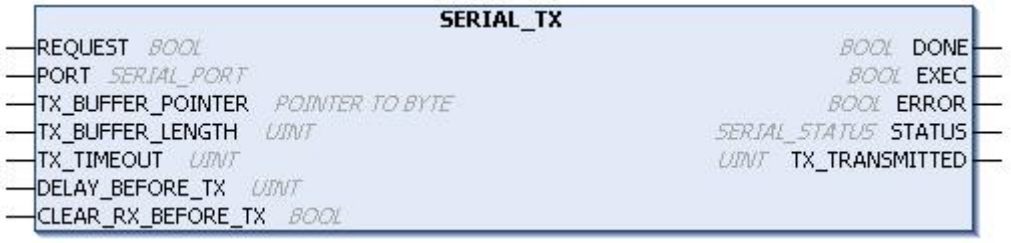


Figura 150: Bloco para Transmitir Valores pela SERIAL_TX

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 145: Parâmetros de Entrada SERIAL_TX

Parâmetros de saída	Tipo	Descrição
DONE	BOOL	Essa variável é verdadeira quando o bloco funcional for executado por completo, caso contrário, ela é falsa.
EXEC	BOOL	Essa variável é verdadeira enquanto o bloco funcional estiver sendo executado, caso contrário, ela é falsa.
ERROR	BOOL	Essa variável é verdadeira quando o bloco funcional concluir a sua execução com algum erro, caso contrário, ela é falsa. Está vinculada à variável DONE, pois seu estado é exibido após a conclusão do bloco funcional.
STATUS	SERIAL_STATUS	Caso a variável ERROR seja verdadeira, a estrutura STATUS exibirá o erro encontrado na execução do bloco. Os estados, já descritos no tipo de dado SERIAL_STATUS, possíveis são: - NO_ERROR - ILLEGAL_SERIAL_PORT - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - ILLEGAL_TX_BUFF_LENGTH - ILLEGAL_TIMEOUT - CTS_TIMEOUT_ON - CTS_TIMEOUT_OFF - TX_TIMEOUT_ERROR - NOT_CONFIGURED
TX_TRANSMITTED	UINT	Retorna o número de bytes transmitidos, o qual deve ser igual ao TX_BUFFER_LENGTH, mas pode ser menor caso ocorra algum erro durante a transmissão.

Tabela 146: 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;
```

5.14.2. 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.

5.14.2.1. REFRESH_INPUT

Essa função é utilizada para atualizar as entradas do módulo especificado, sem aguardar o ciclo ser completado. É importante ressaltar que os filtros, configurados no Mastertool, e o tempo de atualização das entradas do módulo, deverão ser considerados no tempo efetivo de atualização das entradas na aplicação desenvolvida pelo usuário.

ATENÇÃO

A função de *REFRESH_INPUT* deve ser utilizada somente na tarefa MainTask.

ATENÇÃO

A função de *REFRESH_INPUT* não suporta a atualização de entradas que tenham sido mapeadas para variáveis simbólicas. Para o correto funcionamento é necessário que a entrada esteja mapeada para uma variável dentro da memória de variáveis de entrada de representação direta (%I).

ATENÇÃO

A função de *REFRESH_INPUT* atualiza apenas as variáveis diretas %I que estiverem declaradas na aba de "Bus: Mapeamento de E/S"do módulo endereçado no respectivo rack/slot da função. Em se tratando de módulos/interfaces de comunicação (MODBUS, Profibus, etc), a atualização não inclui as variáveis diretas dos mapeamentos dos dispositivos.

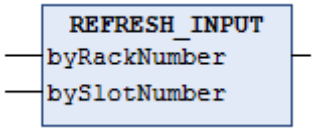


Figura 151: Bloco para Atualizar as Entradas

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

Tabela 147: Parâmetros de Entrada REFRESH_INPUT

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
Info: ERRORCODE;
byRackNumber: BYTE;
bySlotNumber: BYTE;
END_VAR
//ENTRADAS:
byRackNumber := 0;
bySlotNumber := 10;
//FUNÇÃO:
Info := REFRESH_INPUT (byRackNumber, bySlotNumber); //Chamada da função.
//Variável "Info" recebe possíveis erros da função.
```

5.14.2.2. REFRESH_OUTPUT

Essa função é utilizada para atualizar as saídas do módulo especificado, sem aguardar o ciclo ser completado. É importante ressaltar que o tempo de atualização das saídas do módulo deverá ser considerado no tempo efetivo de atualização das saídas na aplicação desenvolvida pelo usuário.

ATENÇÃO

A função de *REFRESH_OUTPUT* deve ser utilizada somente na tarefa MainTask.

ATENÇÃO

A função de *REFRESH_OUTPUT* não suporta a atualização de saídas que tiverem sido mapeadas para variáveis simbólicas. Para o correto funcionamento é necessário que a saída esteja mapeada para uma variável dentro da memória de variáveis de saída de representação direta (%Q).

ATENÇÃO

A função de *REFRESH_OUTPUT* atualiza apenas as variáveis diretas %Q que estiverem declaradas na aba de "Bus: Mapeamento de E/S"do módulo endereçado no respectivo rack/slot da função. Em se tratando de módulos/interfaces de comunicação (MODBUS, Profibus, etc), a atualização não inclui as variáveis diretas dos mapeamentos dos dispositivos.

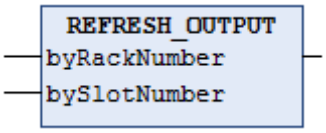


Figura 152: Bloco para Atualizar as Saídas

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

Tabela 148: Parâmetros de Entrada REFRESH_OUTPUT

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
Info: ERRORCODE;
byRackNumber: BYTE;
bySlotNumber: BYTE;
END_VAR
//ENTRADAS:
byRackNumber := 0;
bySlotNumber := 10;
//FUNÇÃO:
//Chamada da função.
Info := REFRESH_OUTPUT (byRackNumber, bySlotNumber);
//Variável "Info" recebe possíveis erros da função.
```

5.14.3. Temporizador Retentivo

O Temporizador Retentivo é um bloco funcional desenvolvido para aplicações como relógios de linha de produção, as quais necessitam armazenar o seu valor e reiniciar a contagem do mesmo ponto em caso de falha na alimentação. Os valores, guardados pelo bloco funcional somente serão zerados em caso de um *Reset a Frio*, *Reset Origem* ou o *Download* de uma nova aplicação (ver Manual de Utilização do Mastertool), sendo que os contadores continuam em funcionamento mesmo que a aplicação esteja parada (Modo Stop).

ATENÇÃO

É importante destacar que, para o correto funcionamento dos blocos funcionais do Temporizador Retentivo, as variáveis de controle devem ser declaradas como retentivas (*VAR RETAIN*). Também é importante ressaltar que em modo simulação os blocos funcionais do Temporizador Retentivo não são executados adequadamente em virtude de necessitarem da UCP Nexto para o correto comportamento.

Abaixo, são descritos os três tipos de blocos disponíveis na biblioteca *NextoStandard* do software Mastertool (para realizar o procedimento de inserção de uma biblioteca, consultar o Manual de Programação IEC 61131 – MP399048, capítulo Bibliotecas).

5.14.3.1. TOF_RET

O bloco funcional *TOF_RET* implementa um tempo de atraso para desabilitar uma saída. Quando a entrada *IN* tem seu estado alterado de verdadeiro (TRUE) para falso (FALSE), ou seja, uma borda de descida, o tempo especificado *PT* irá

transcorrer até que a saída *Q* também seja falsa (FALSE). Quando a entrada *IN* tem nível lógico 1 (TRUE), a saída *Q* também permanecerá no mesmo estado (TRUE), mesmo que isso aconteça no meio de uma contagem. O tempo *PT* pode ser alterado durante a contagem, pois o bloco funcional assumirá o novo valor, desde que a contagem não tenha chegado ao final. A Figura 153 representa o bloco *TOF_RET* e a Figura 154 mostra o comportamento gráfico do mesmo.



Figura 153: 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 149: 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 150: Parâmetros de Saída TOF_RET

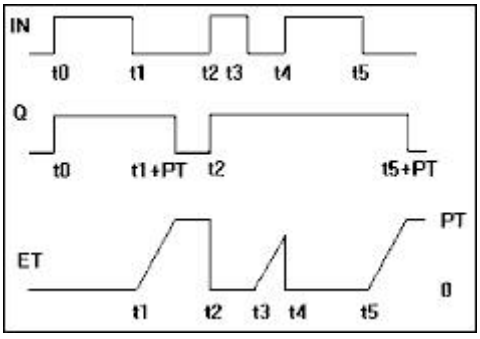


Figura 154: Comportamento Gráfico do Bloco Funcional TOF_RET

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
```

5.14.3.2. TON_RET

O bloco funcional *TON_RET* implementa um tempo de atraso para habilitar uma saída. Quando a entrada *IN* tem seu estado alterado de falso (FALSE) para verdadeiro (TRUE), ou seja, uma borda de subida, o tempo especificado *PT* irá transcorrer até que a saída *Q* também seja verdadeira (TRUE). Quando a entrada *IN* tem nível lógico 0 (FALSE), a saída *Q* também permanecerá no mesmo estado (FALSE), mesmo que isso aconteça no meio de uma contagem. O tempo *PT* pode ser alterado durante a contagem, pois o bloco funcional assumirá o novo valor, desde que a contagem não tenha chegado ao final. A Figura 155 representa o bloco *TON_RET* e a Figura 156 mostra o comportamento gráfico do mesmo.



Figura 155: 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 151: 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 152: Parâmetros de Saída TON_RET

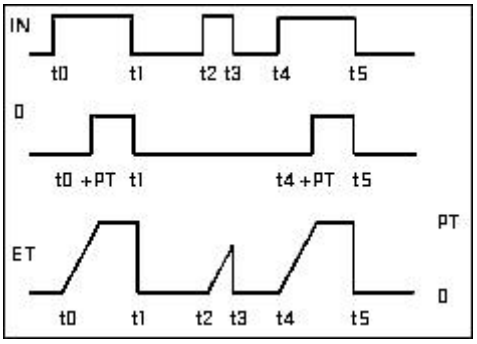


Figura 156: Comportamento Gráfico do Bloco Funcional TON_RET

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
```

5.14.3.3. TP_RET

O bloco funcional *TP_RET* trabalha como um trigger. O temporizador, que inicia quando a entrada *IN* tem seu estado alterado de falso (FALSE) para verdadeiro (TRUE), ou seja, uma borda de subida, é incrementado até que o limite de tempo *PT* seja atingido. Durante a contagem, a saída *Q* é verdadeira (TRUE), caso contrário ela é falsa (FALSE). O tempo *PT* pode ser alterado durante a contagem, pois o bloco assumirá o novo valor, desde que a contagem não tenha chegado ao final. A Figura 157 representa o bloco *TP_RET* e a Figura 158 mostra o comportamento gráfico do mesmo.



Figura 157: 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 153: 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 154: Parâmetros de Saída TP_RET

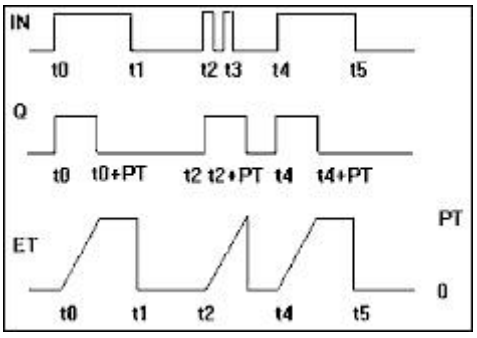


Figura 158: Comportamento Gráfico do Bloco Funcional TP_RET

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
```

5.14.4. Temporizador Não-Redundante

O temporizador não redundante é utilizado em aplicações para as UCPs redundantes que necessitam de um temporizador no programa não redundante de um half-cluster. Este temporizador não utiliza o temporizador IEC, portanto, não irá ser sincronizado no caso de o half-cluster reserva assumir o estado ativo e o ativo passar para reserva.

Abaixo, são descritos os três tipos de blocos já disponíveis na biblioteca *NextoStandard* do software Mastertool (para realizar o procedimento de inserção de uma biblioteca, consultar o Manual de Programação IEC 61131 – MP399048, capítulo Bibliotecas).

5.14.4.1. TOF_NR

O bloco funcional *TOF_NR* implementa um tempo de atraso para desabilitar uma saída e tem o funcionamento e configuração parecidos com o bloco funcional TOF_RET, se diferenciando apenas por não ser redundante e nem retentivo.

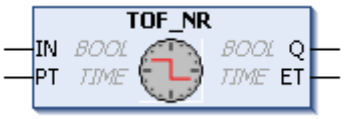


Figura 159: Bloco Funcional TOF_NR

Exemplo de utilização em linguagem ST:

```
PROGRAM NonSkippedPrg
VAR
bStart : BOOL := TRUE;
TOF_NR : TOF_NR;
END_VAR

// Quando bStart=FALSE inicia contagem
TOF_NR( IN := bStart,
PT := T#20S);

// Executa ações ao final da contagem
IF (TOF_NR.Q = FALSE) THEN
bStart := TRUE;
END_IF
```

5.14.4.2. TON_NR

O bloco funcional *TON_NR* implementa um tempo de atraso para habilitar uma saída e tem o funcionamento e configuração parecidos com o bloco funcional TON_RET, se diferenciando apenas por não ser redundante e nem retentivo.

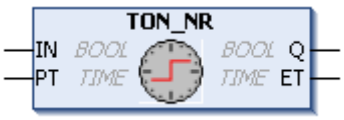


Figura 160: Bloco Funcional TON_NR

Exemplo de utilização em linguagem ST:

```
PROGRAM NonSkippedPrg
VAR
bStart : BOOL;
TON_NR : TON_NR;
END_VAR

// Quando bStart=TRUE inicia contagem
TON_NR( IN := bStart,
PT := T#20S);

// Executa ações ao final da contagem
IF (TON_NR.Q = TRUE) THEN
bStart := FALSE;
END_IF
```

5.14.4.3. TP_NR

O bloco funcional *TP_NR* trabalha como um trigger e tem o funcionamento e configuração parecidos com o bloco funcional TP_RET, se diferenciando apenas por não ser redundante e nem retentivo.

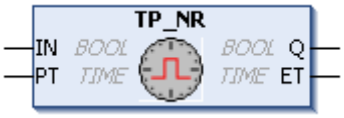


Figura 161: Bloco Funcional TP_NR

Exemplo de utilização em linguagem ST:

```
PROGRAM NonSkippedPrg
VAR
bStart : BOOL;
TP_NR : TP_NR;
END_VAR

// Configura TP_NR
TP_NR( IN := bStart,
PT := T#20S);

bStart := FALSE;

// Ações durante a contagem
IF (TP_NR.Q = TRUE) THEN
// Executa enquanto o contador estiver ativado
ELSE
// Executa somente quando o contador estiver desativado
END_IF
```

5.14.5. Log de Usuário

Recurso que permite ao usuário criar seus próprios registros e gravar em arquivos de log no cartão de memória presente na UCP. Os arquivos são gerados em um diretório específico do cartão de memória no formato CSV, permitindo a visualização em editores de texto e planilhas. O separador adotado foi o caractere ponto e vírgula. Para mais informações sobre a utilização do cartão de memória, ver seção Cartão de Memória.

Há duas funções disponíveis, uma para registrar informações e outra para remover todos os registros. A seguir uma descrição dos tipos de dados utilizados pelas funções:

Tipo de dado	Opção	Descrição
USER_LOG_EVENT_TYPES	USER_LOG_EVENT_ERROR	O usuário é livre para usar a melhor indicação conforme a severidade de sua mensagem de log.
	USER_LOG_EVENT_DEBUG	
	USER_LOG_EVENT_INFO	
	USER_LOG_EVENT_WARN	
USER_LOG_MESSAGE		Mensagem de log com o máximo de 150 caracteres.
USER_LOG_ERROR_CODES	USER_LOG_OK	A operação foi executada com sucesso.
	USER_LOG_FAILED	A operação não foi executada com sucesso. O motivo da falha pode ser verificado nos logs de sistema – ver seção Log de Sistema.
	USER_LOG_BUFFER_FULL	Mensagens estão sendo adicionadas além da capacidade de processamento.
	USER_LOG_NO_MEMORY	No momento, não houve recursos para executar a operação.
	USER_LOG_FILE_SYSTEM_ERROR	Houve algum erro no acesso ao cartão de memória ou não há espaço disponível. Informações do erro podem ser verificadas nos logs de sistema – ver seção Log de Sistema.
	USER_LOG_NO_MEMORY_CARD	Não há um cartão de memória presente na UCP.
	USER_LOG_MEMORY_CARD_FULL	Não há espaço livre disponível no cartão de memória.
	USER_LOG_PROCESSING	O recurso está ocupado executando a última operação, por exemplo, deletando todos os arquivos de log.

Tabela 155: Tipos de dados para Log de Usuário

A seguir, são descritas as duas funções disponíveis na biblioteca *LibLogs* no Mastertool. Para realizar o procedimento de inserção de uma biblioteca, consultar o Manual de Programação IEC 61131 – MP399048, capítulo Bibliotecas.

ATENÇÃO

Os Logs de Usuário estão disponíveis apenas a partir da versão 1.3.0.20 das UCPs da Série Nexto. Da mesma forma, para utilizar esta funcionalidade é necessária a versão 1.40 ou superior do Mastertool.

5.14.5.1. 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 155. 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:

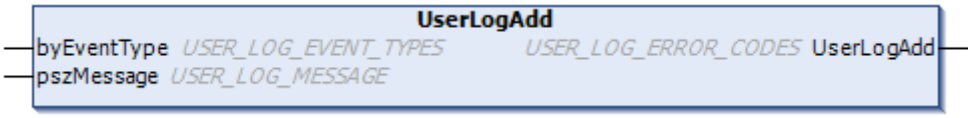


Figura 162: Função UserLogAdd

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 156: 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-20130827150624573 8.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◦
```

5.14.5.2. 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*.

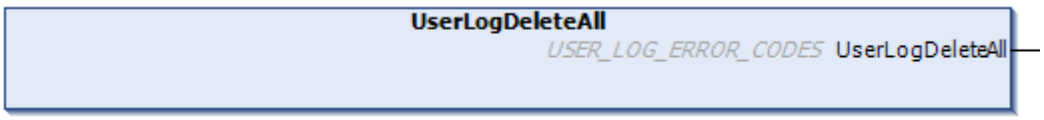


Figura 163: Função UserLogDeleteAll

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.

5.15. Acesso a Aba de Gerenciamento

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

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

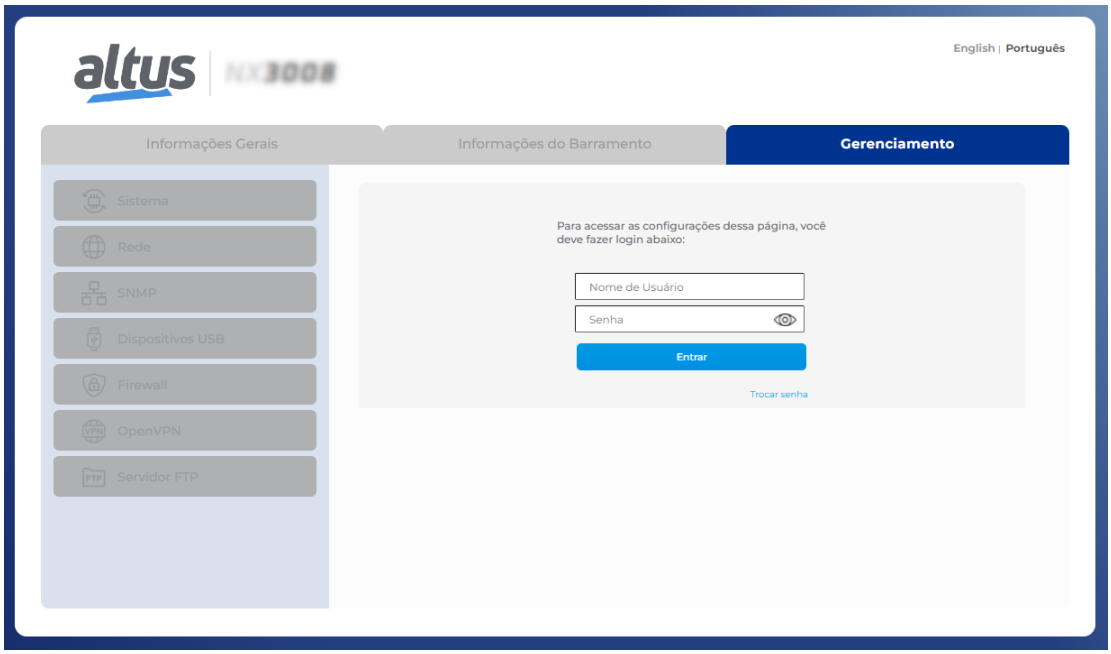


Figura 164: Acesso a aba de Gerenciamento

5.15.1. Seção de Sistema

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

5.15.1.1. Ajuste do Relógio

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

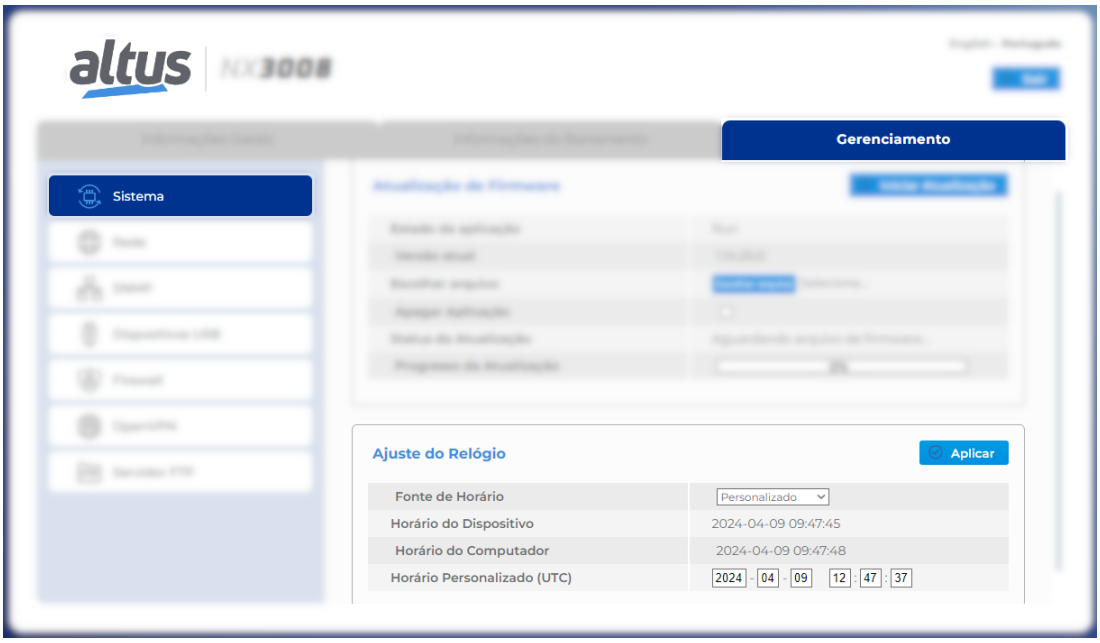


Figura 165: Ajuste do Relógio

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

5.15.1.1.1. Horário do Computador (UTC)

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

5.15.1.1.2. Horário Personalizado (UTC)

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

5.15.2. Seção de Rede

Desenvolvida para auxiliar na usabilidade do controlador, a seção de Rede (figura abaixo) possibilita a alteração dos endereços de rede e a execução do Sniffer de rede.

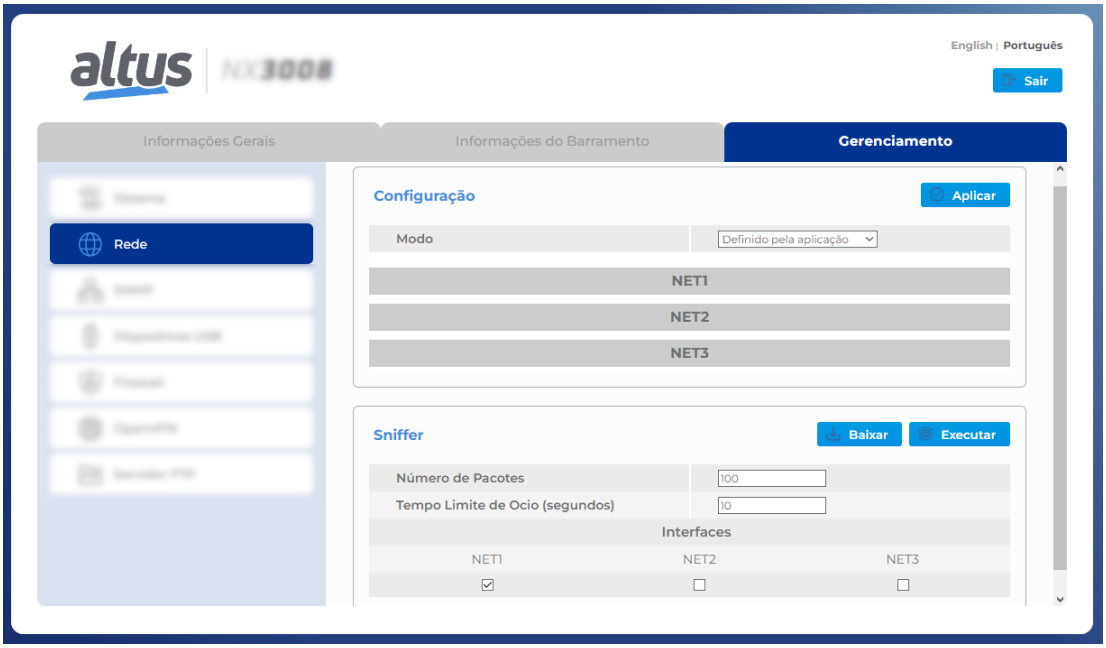


Figura 166: Seção de Rede

5.15.2.1. Configurações da Seção de Rede

ATENÇÃO

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

5.15.2.1.1. Modo Definido pela aplicação

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

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

ATENÇÃO

A tabela para configuração de rede é exibida somente quando não possuir aplicação no controlador ou este não estiver em execução. Não é possível alterar as configurações de rede enquanto houver uma aplicação em execução no controlador.

Abaixo uma imagem com o modo *Definido pela aplicação* selecionado, mostrando a tabela de interfaces desabilitada.

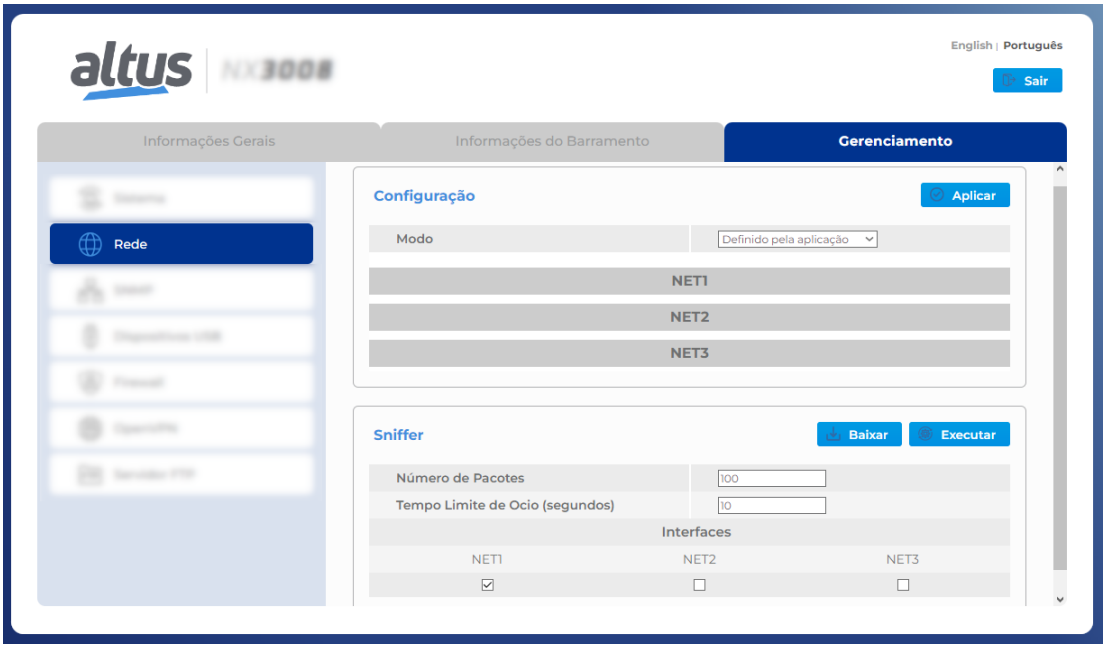


Figura 167: Tabela de Interfaces - Modo Aplicação

5.15.2.1.2. *Modo Definido pela Página Web*

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

Neste modo, o usuário pode configurar Endereço IP, Máscara de Rede e Gateway de cada uma das interfaces Ethernet disponíveis, além de poder habilitar e desabilitar as NETs 2 e 3.



Figura 168: Tabela de Interfaces - Modo Web

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

Quando este checkbox estiver desmarcado, indica que aquela interface NET está desabilitada, ou seja, não receberá configuração e ficará desativada, conforme pode ser visto na NET 3 da figura acima. Já quando a interface está habilitada (com o checkbox marcado), conforme a NET 2 da figura acima, as configurações ficam disponíveis para edição.

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

ATENÇÃO

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

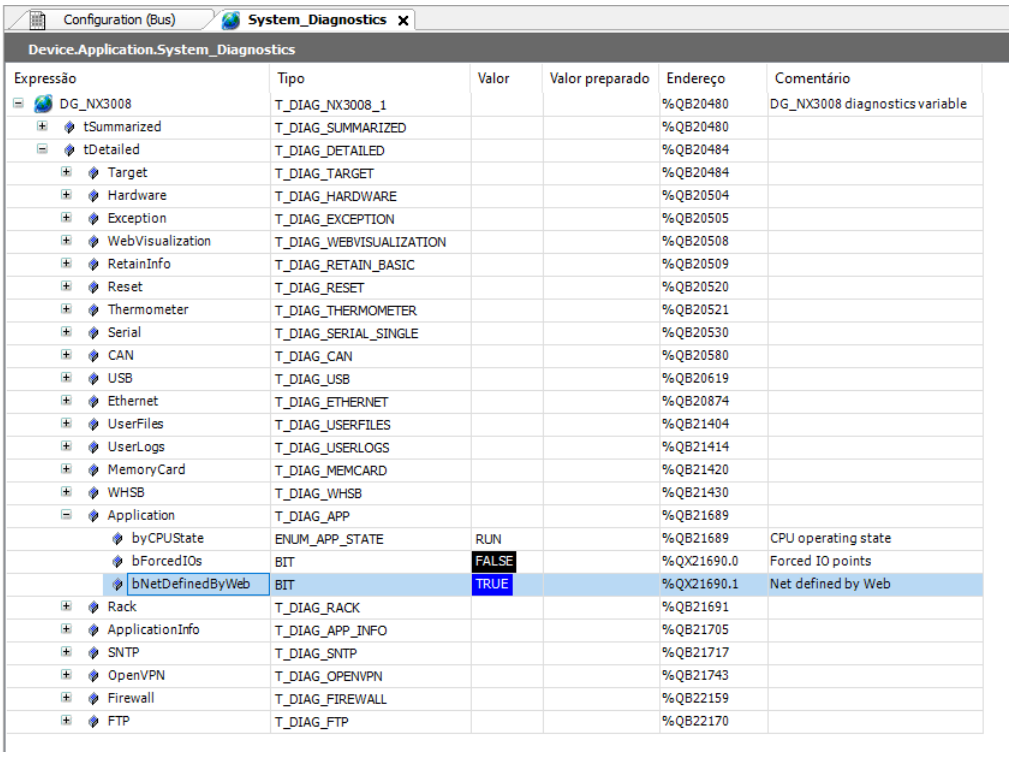
Ao aplicar as configurações utilizando o modo de operação *Definido pela Aplicação*, o controlador vai assumir as configurações que foram definidas pela aplicação já carregada nele anteriormente. Caso não haja nenhuma aplicação no controlador, a configuração atual será mantida, sendo alterado apenas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

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

ATENÇÃO

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

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



Expressão	Tipo	Valor	Valor preparado	Endereço	Comentário
DG_NX3008	T_DIAG_NX3008_1			%QB20480	DG_NX3008 diagnostics variable
tSummarized	T_DIAG_SUMMARIZED			%QB20480	
tDetailed	T_DIAG_DETAILED			%QB20484	
Target	T_DIAG_TARGET			%QB20484	
Hardware	T_DIAG_HARDWARE			%QB20504	
Exception	T_DIAG_EXCEPTION			%QB20505	
WebVisualization	T_DIAG_WEBVISUALIZATION			%QB20508	
RetainInfo	T_DIAG_RETAIN_BASIC			%QB20509	
Reset	T_DIAG_RESET			%QB20520	
Thermometer	T_DIAG_THERMOMETER			%QB20521	
Serial	T_DIAG_SERIAL_SINGLE			%QB20530	
CAN	T_DIAG_CAN			%QB20580	
USB	T_DIAG_USB			%QB20619	
Ethernet	T_DIAG_ETHERNET			%QB20874	
UserFiles	T_DIAG_USERFILES			%QB21404	
UserLogs	T_DIAG_USERLOGS			%QB21414	
MemoryCard	T_DIAG_MEMCARD			%QB21420	
WHSB	T_DIAG_WHSB			%QB21430	
Application	T_DIAG_APP			%QB21689	
byCPUState	ENUM_APP_STATE	RUN		%QB21689	CPU operating state
bForcedIOs	BIT	FALSE		%QX21690.0	Forced IO points
bNetDefinedByWeb	BIT	TRUE		%QX21690.1	Net defined by Web
Rack	T_DIAG_RACK			%QB21691	
ApplicationInfo	T_DIAG_APP_INFO			%QB21705	
SNTP	T_DIAG_SNTP			%QB21717	
OpenVPN	T_DIAG_OPENVPN			%QB21743	
Firewall	T_DIAG_FIREWALL			%QB22159	
FTP	T_DIAG_FTP			%QB22170	

Figura 169: Diagnóstico - IP definido pela Página Web

5.15.2.2. Sniffer de Rede

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

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

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

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

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

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

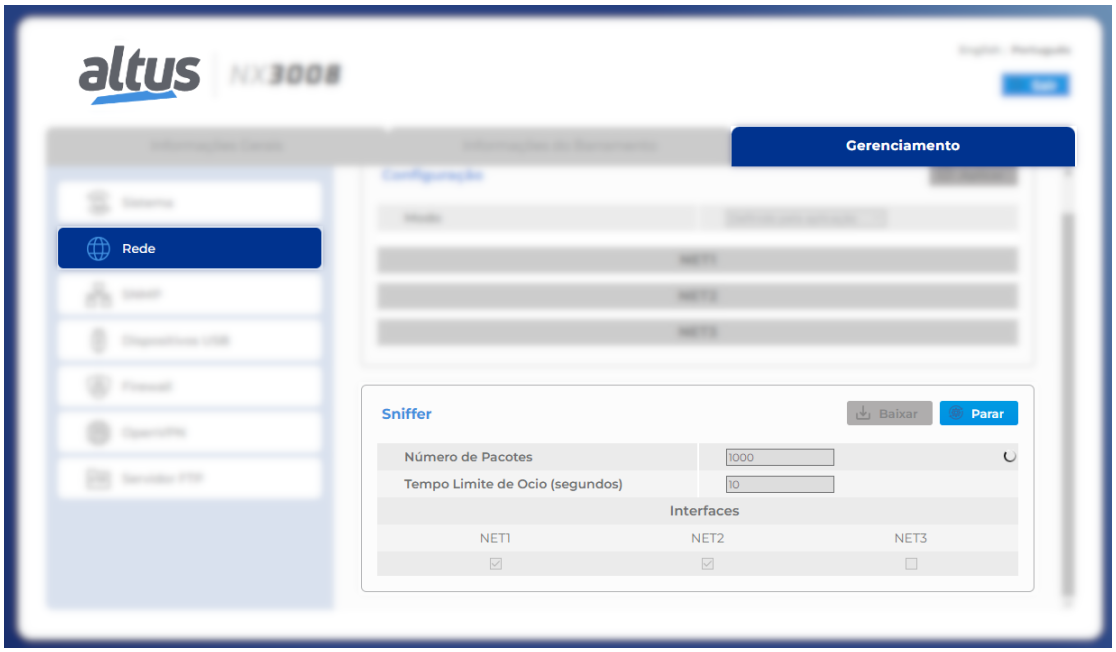


Figura 170: Sniffer de Rede Executando

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

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

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

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

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

5.16. SNMP

SNMP (*Simple Network Management Protocol*) é um protocolo muito utilizado por administradores de redes por fornecer importantes informações e diagnósticos de equipamentos presentes em determinada rede Ethernet.

Este protocolo usa o conceito de agente e gerente, no qual o gerente envia requisições de leitura ou escrita de determinados objetos para o agente. Através de uma MIB (*Management Information Base*) o gerente tem conhecimento dos objetos existentes no agente, e dessa forma poderá fazer requisições desses objetos, respeitando as permissões de leitura ou escrita dos mesmos.

MIB é uma coleção de informações organizadas hierarquicamente, onde cada objeto desta árvore é chamado de OID (*Object Identifier*).

Para todos os equipamentos com SNMP, é obrigatório o suporte a MIB-II. Nesta MIB estão descritas as principais informações para gerenciamento de redes Ethernet.

5.16.1. 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 157: 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.

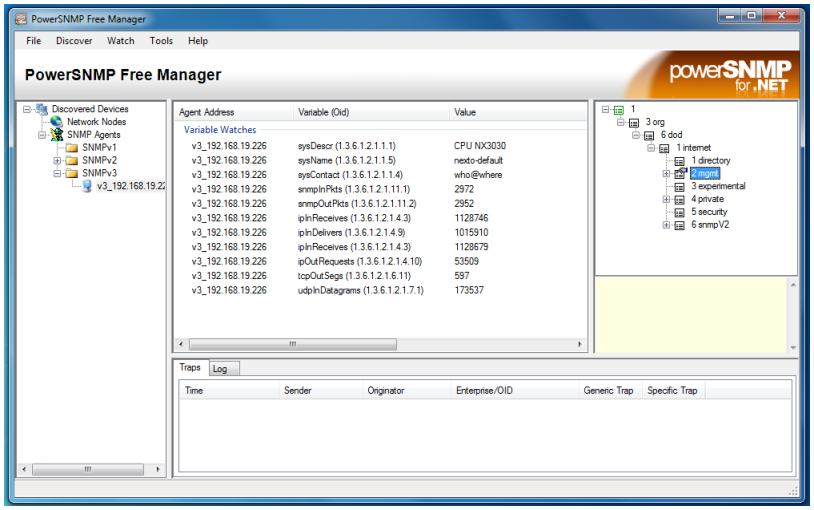


Figura 171: Exemplo de Gerente SNMP

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.

5.16.2. Configuração SNMP

As configurações do SNMP podem ser alteradas através da Página Web de Sistema da UCP, na aba *Gerenciamento*, na seção *SNMP*. Efetuado o login com sucesso, poderá ser visualizado o estado atual do serviço (ativado ou desativado), assim como as informações de usuário SNMPv3 e comunidades para SNMPv1/v2c.

O usuário poderá ativar ou desativar o serviço através de um checkbox na parte superior da tela.

É possível também alterar as informações de SNMPv3, clicando no botão *Alterar* logo abaixo das informações de usuário. Abrirá um formulário onde é necessário preencher o usuário e senha antigos, e o novo usuário e senha. As demais informações de usuário SNMPv3 não podem ser alteradas.

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

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

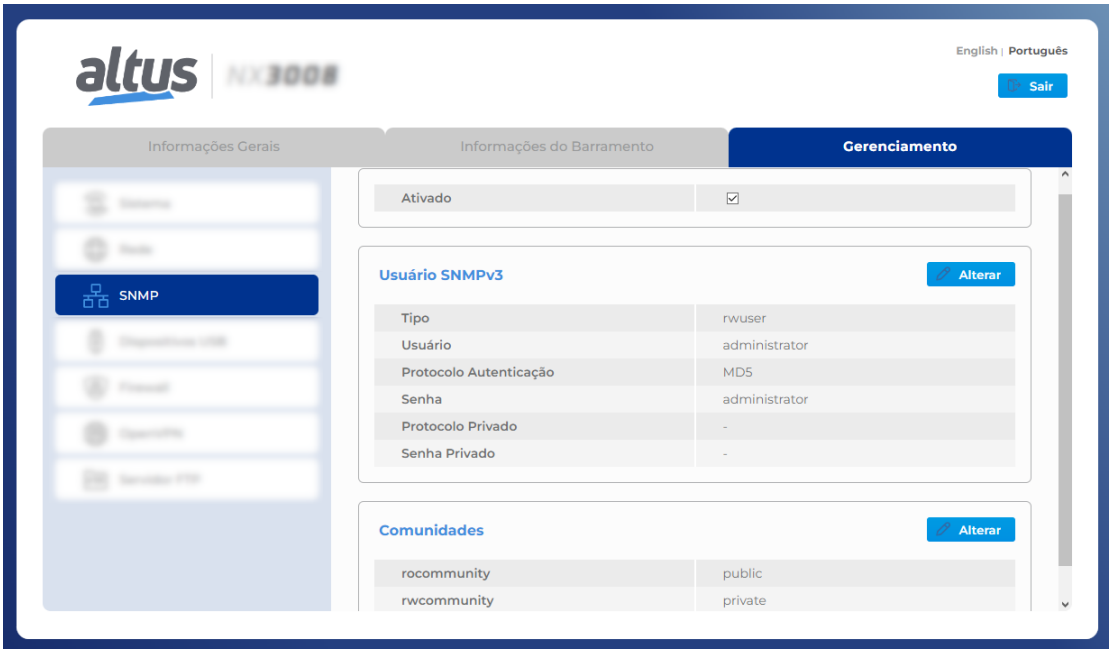


Figura 172: Tela de Configuração e Status SNMP

5.16.3. Usuário e Comunidades SNMP

Para acesso ao SNMPv1/v2c das UCPs da Série Nexto, existem duas comunidades, conforme tabela abaixo.

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

Tabela 158: 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 159: 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 160: Limites para Configurações SNMP

5.17. Firewall

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

ATENÇÃO

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

5.17.1. Configuração

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

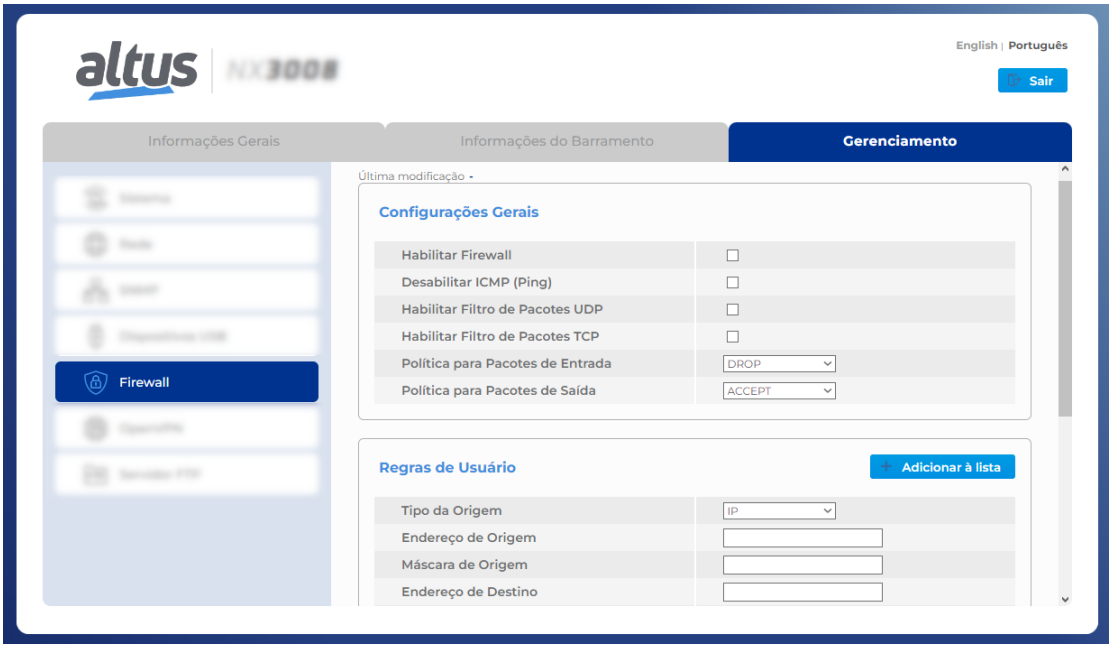


Figura 173: Tela de Configuração do Firewall

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

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

5.17.2. Configurações Gerais

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

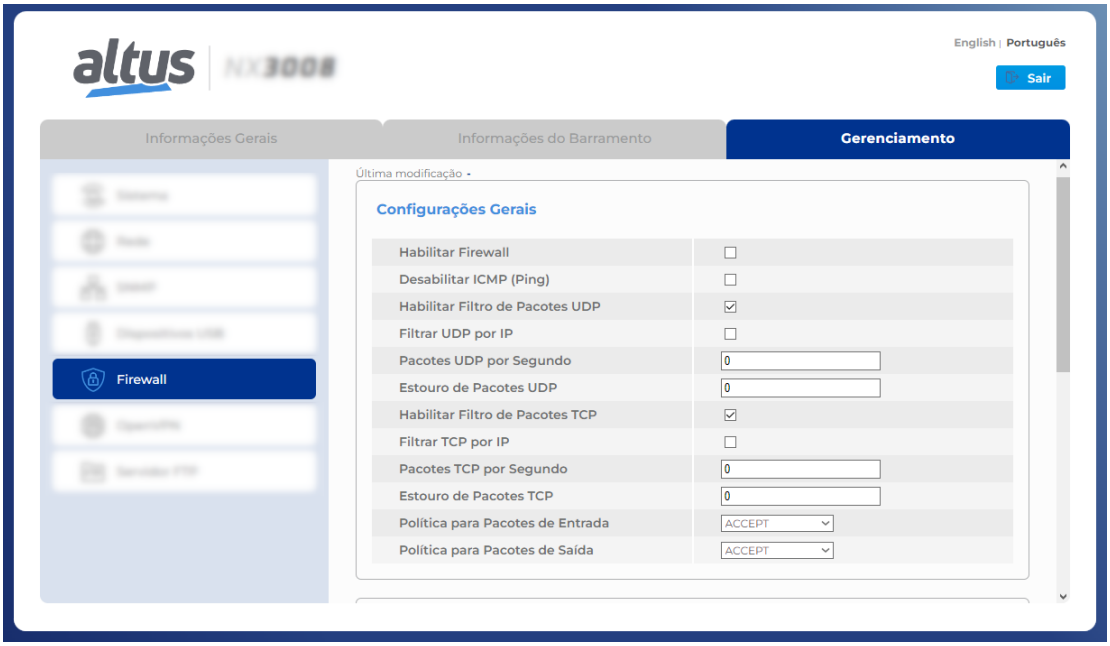


Figura 174: Tabela de Configurações Gerais do Firewall

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

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

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

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

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

ATENÇÃO

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

As configurações desta tabela são aplicadas com o botão de *Aplicar* que aparece na figura 176.

Os campos para selecionar ambas as políticas, de entrada e de saída, possuem as opções de *Accept*, aceitar, e de *Drop*,

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

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

ATENÇÃO

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

ATENÇÃO

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

5.17.3. Regras de Usuário

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

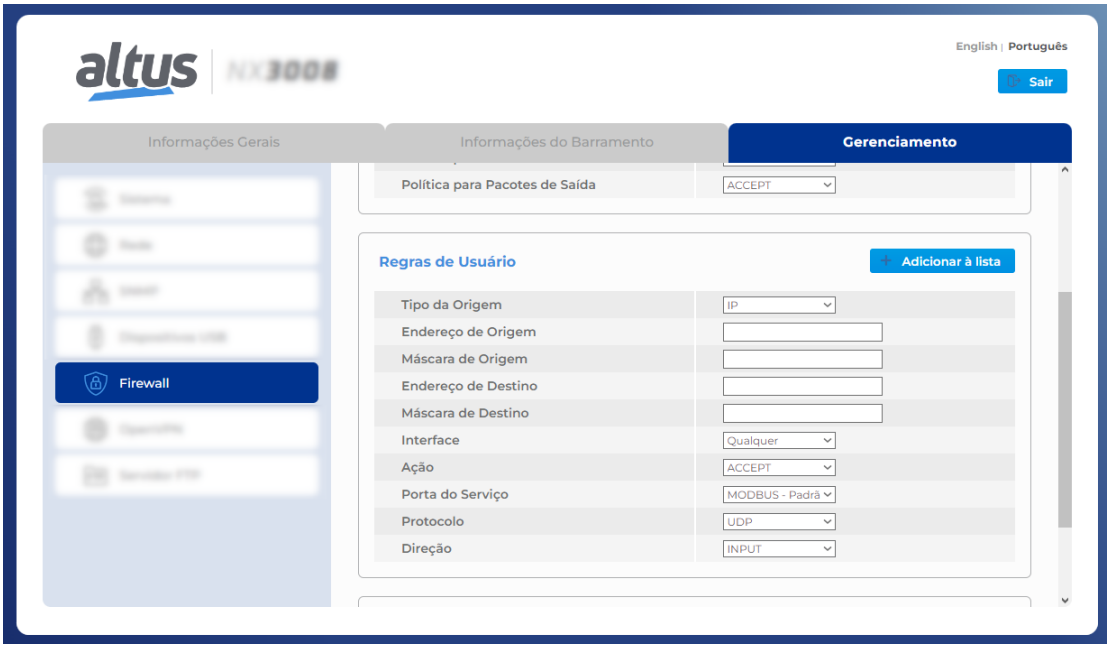


Figura 175: Tabela de Configuração de Regras de Usuário do Firewall

Esta tabela altera o seu formato de acordo com o *Tipo de Origem* selecionado, podendo ser IP ou MAC. Quando o tipo é *IP*, a tabela possuí os itens exibidos na figura acima, porém, quando é selecionado o tipo como *MAC* os campos de máscaras de origem e destinos desaparecem, assim como o campo de *Endereço do Destino*. O item *Endereço de Origem* passa a aceitar um endereço MAC como entrada em um formato de seis grupos de dois digitos hexadecimais separados por dois pontos, ex:

"1A:2B:3C:4D:5E:6F". Além disso, uma regra com base em endereço *MAC* somente pode ser configurada como uma regra de entrada, ou seja, o campo *Direção* será forçado com o valor *INPUT*.

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

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

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

O campo *Porta de Serviço* serve para indicar quais as portas serão configuradas nesta regra. Todas as portas de serviços que possuem um determinado protocolo ou comunicação *padrão* para o controlador, como por exemplo o protocolo MODBUS que tem a porta padrão 502, estão disponíveis com o nome do serviço e a porta utilizada ao lado. Assim, caso seja configurada a regra para o protocolo MODBUS, será aplicada a porta 502, caso seja configurada a regra para o serviço webvisu, será aplicada a porta 8080 e assim segue para os demais protocolos listados no campo de seleção. Este campo também possuí outras duas configurações, que são *Qualquer* e *Outra*. Quando é selecionada a opção *Qualquer*, a regra é aplicada para todas as portas de serviços, exceto a porta 80, então são criadas duas regras utilizando as seguintes faixas de portas: *1:79* e *81:65535*. Caso seja selecionada a opção *Outra*, será exibido uma caixa de texto na qual é possível configurar a porta que deseja, exceto a porta 80. Para configurar uma porta, basta escrever o seu número na caixa de texto, se desejar adicionar mais de uma única porta, é necessário utilizar o separador *"&"* e caso queira inserir uma faixa de portas, basta informar a porta inicial e final utilizando o separador *":"*.

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

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

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

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

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

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

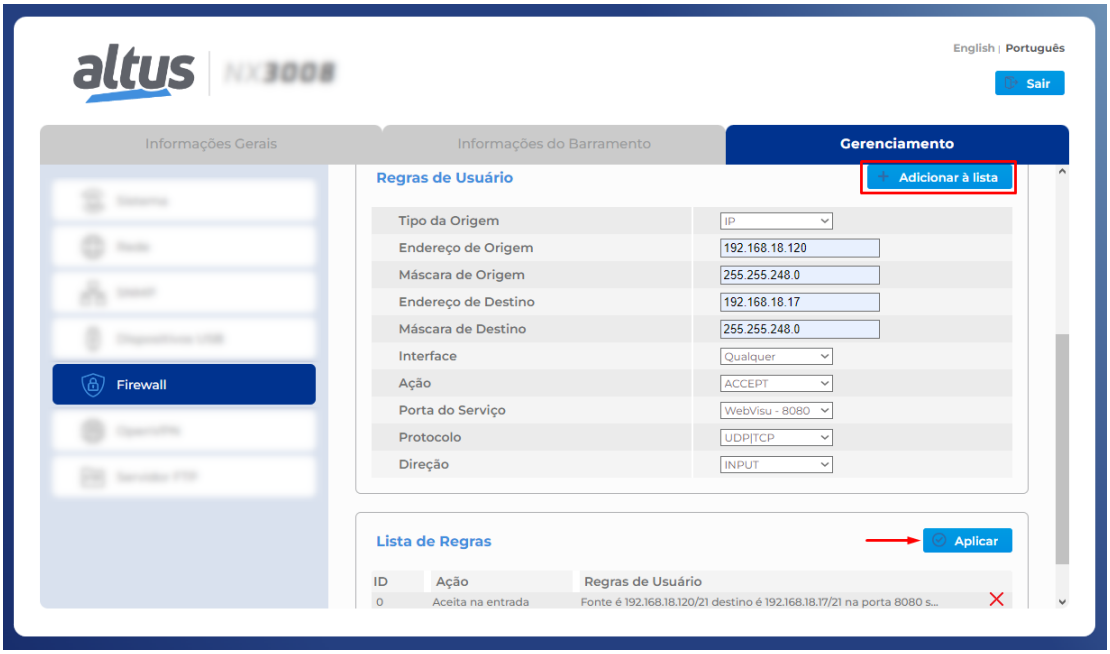


Figura 176: Tabela de Aplicação de Regras de Usuário do Firewall

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

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

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

ATENÇÃO

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

5.17.3.1. Modo Switch ou Nic Teaming

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

ATENÇÃO

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

5.18. OpenVPN

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

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

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

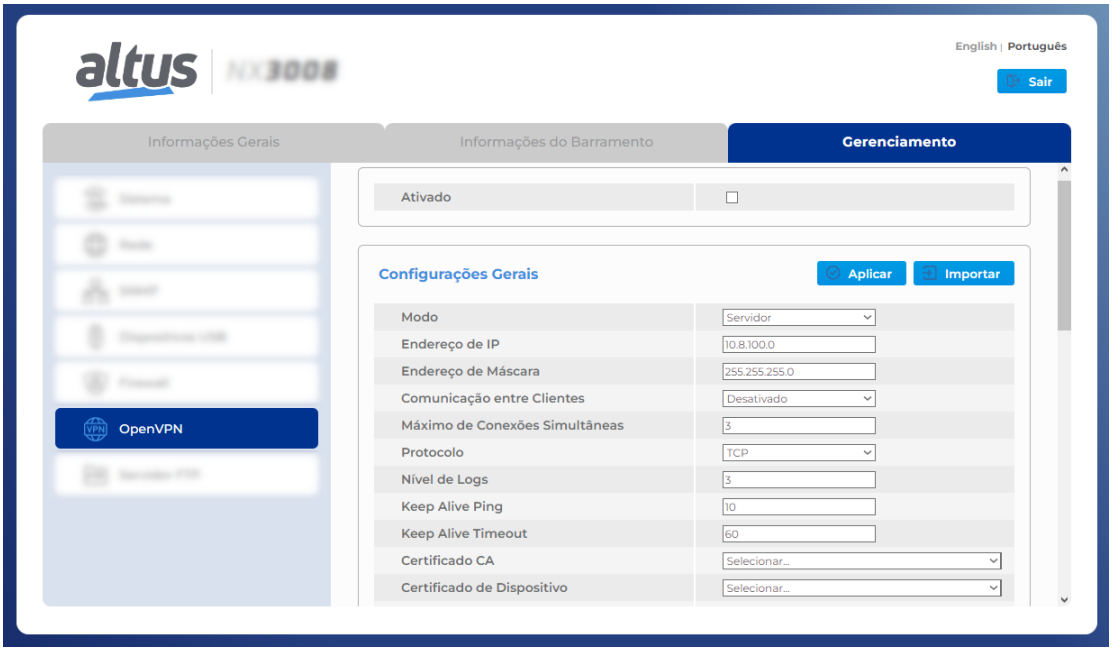


Figura 177: Tela de Configuração de OpenVPN

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

5.18.1. Importação de Configurações

Para realizar a configuração da VPN de forma rápida e fácil no seu dispositivo, pode-se utilizar o botão *Importar* que aparece na figura 177 no canto superior direito da página. Ao clicar sobre este botão, é aberta uma janela, do explorador de arquivos, para que seja feita a seleção de um arquivo de configuração. Devem ser selecionados os arquivos com extensão *conf* ou *ovpn*. Ao selecionar um arquivo, o seu conteúdo será lido e os parâmetros de configuração que estiverem presentes preencherão os seus respectivos campos de configuração na página web.

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

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

ATENÇÃO

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

5.18.2. Configuração do OpenVPN

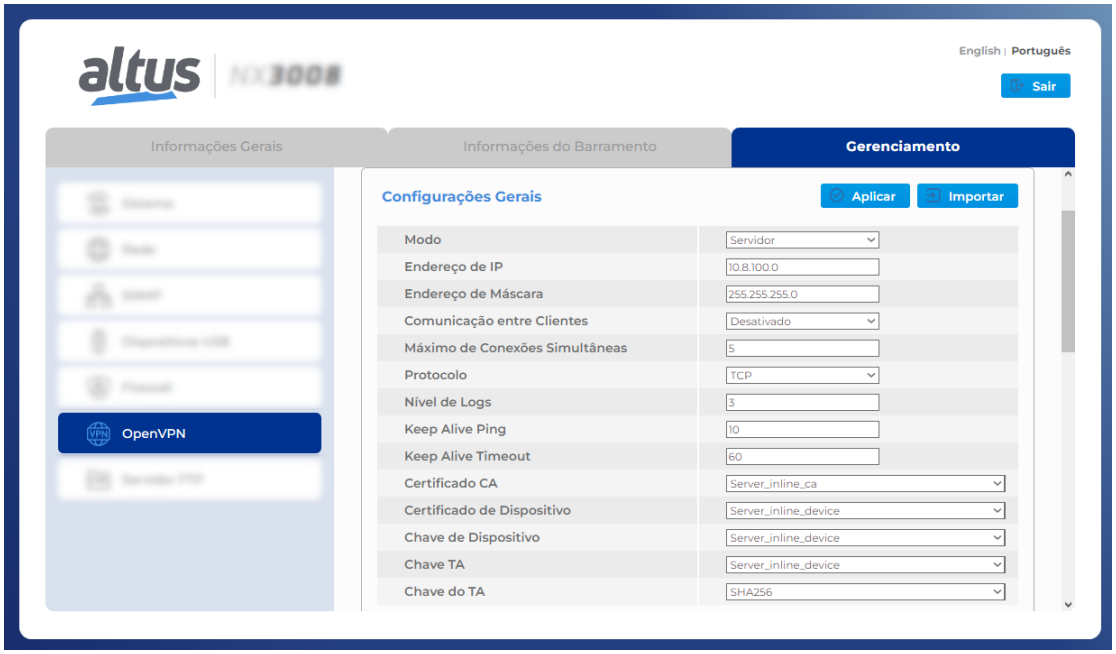


Figura 178: Tabela de Configurações de Servidor OpenVPN

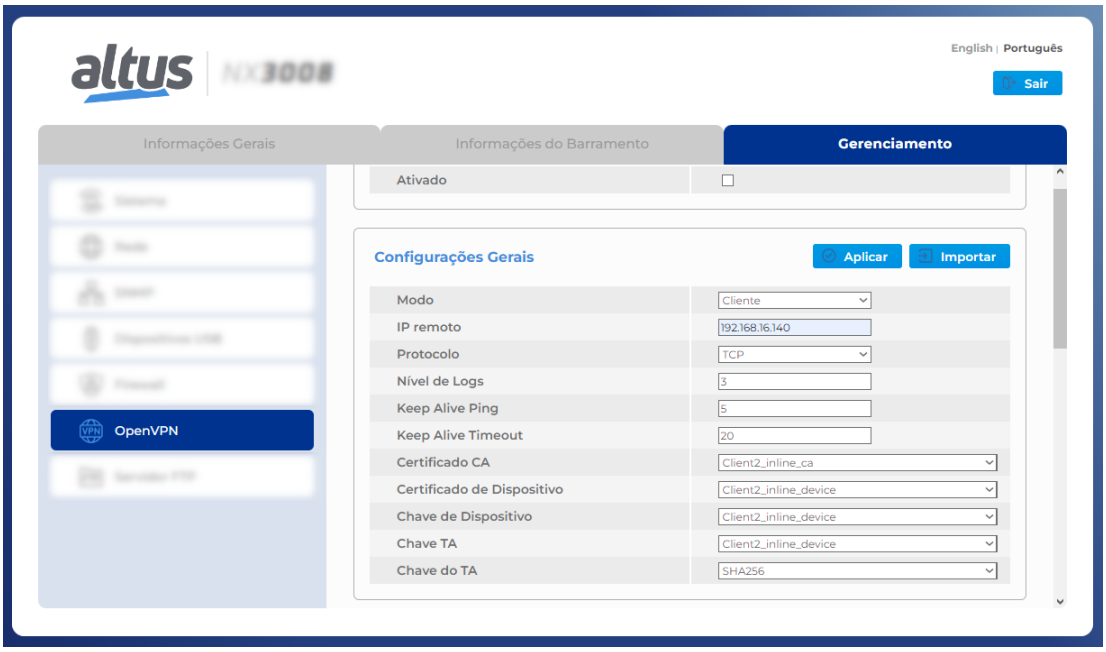


Figura 179: Tabela de Configurações de Cliente OpenVPN

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

5.18.2.1. Configurações Comuns

Observando as figuras com as configurações de cliente, figura 179, e a de servidor, figura 178, é possível identificar que diversos parâmetros são os mesmos para ambas as configurações. São eles:

5.18.2.1.1. Modo

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

5.18.2.1.2. Protocolo

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

ATENÇÃO

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

5.18.2.1.3. Nível de Logs

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

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

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

5.18.2.1.4. Keep Alive Ping

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

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

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

5.18.2.1.5. Keep Alive Timeout

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

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

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

5.18.2.1.6. Arquivos de Segurança

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

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

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

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

5.18.2.1.7. Chave do TA

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

ATENÇÃO

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

5.18.2.2. Configurações Exclusivas do Servidor

As configurações exclusivas do servidor, observadas na figura 178, são descritas a seguir.

5.18.2.2.1. Endereço de Rede

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

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

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

5.18.2.2.2. Comunicação entre Clientes

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

5.18.2.2.3. Máximos Clientes Conectados

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

5.18.2.2.4. Redes Privadas

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

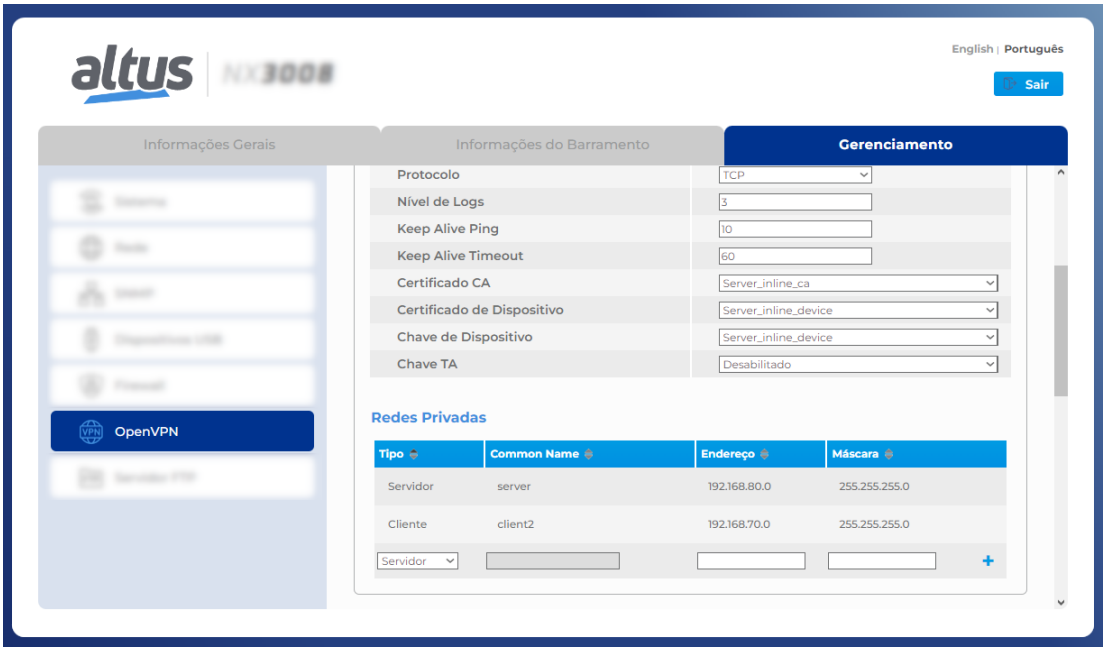


Figura 180: Tabela de Configurações de Redes Privadas OpenVPN

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

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

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

Veja na figura a seguir um exemplo de arquitetura:

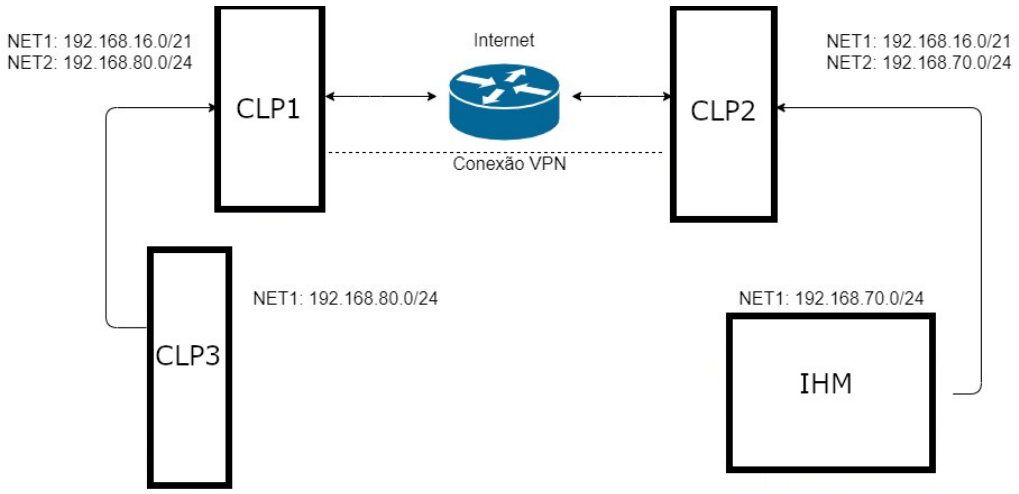


Figura 181: Exemplo de Arquitetura com Redes Privadas

Na Figura 181, o CLP1 da esquerda possui a rede privada 80 configurada na sua NET 2 e conectado a ele existe um CLP3 na mesma rede. O CLP2 da direita possui uma rede privada 70 configurada na NET 2 e conectado a ele existe uma

IHM, na mesma rede. A arquitetura exemplo realiza a comunicação entre os dispositivos CLP3 e IHM pela VPN, através da configuração das suas respectivas redes privadas.

Após preencher os campos, mostrados na Figura 180, com a configuração desejada, deve-se clicar sobre o botão + azul que aparece na extrema direita dos campos de configuração, para que a regra seja adicionada à tabela. Caso deseje excluir alguma regra, arraste o mouse sobre a regra que deseja remover e, então, um *X* vermelho irá aparecer à direita, como é mostrado na Figura 180. Ao clicar sobre este *X*, a regra é removida da tabela.

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

5.18.2.3. Configurações Exclusivas de Cliente

Existe somente uma única configuração exclusiva de clientes OpenVPN na página, que pode ser visualizada na figura 179. Esta configuração é o *IP Remoto*.

5.18.2.3.1. IP Remoto

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

ATENÇÃO

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

5.18.2.4. Aplicação de Configurações

Para habilitar a funcionalidade, deve-se marcar o checkbox *Ativado*, exibido na figura 179. Caso deseje apenas aplicar as configurações realizadas e não habilitar o OpenVPN, desmarque este checkbox.

Após realizar todas as configurações desejadas, as configurações devem ser aplicadas no dispositivo, para isso utilize o botão *Aplicar*. Este botão é mostrado na figura 179, no canto inferior direito. Quando as configurações são aplicadas e a VPN está habilitada, a seção *Firewall* irá realizar um *scroll* automático até a tabela de *status* do OpenVPN, exibida na seção Tabela de Status.

5.18.3. Arquivos de Segurança

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

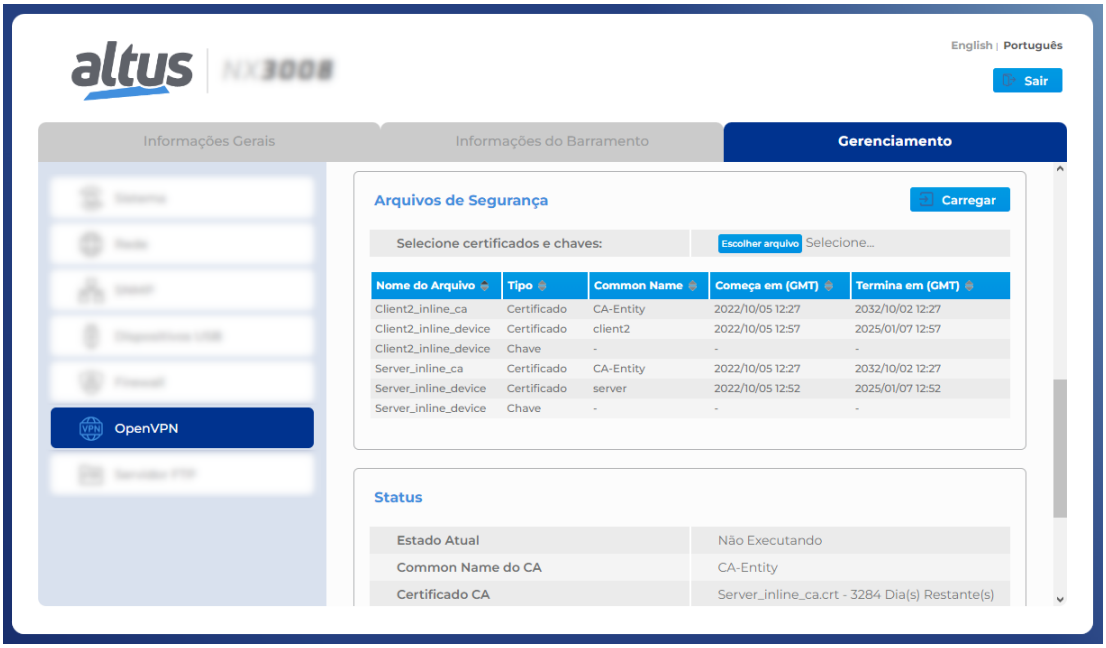


Figura 182: Tabela de Arquivos de Segurança OpenVPN

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

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

ATENÇÃO

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

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

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

5.18.4. Tabela de Status

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

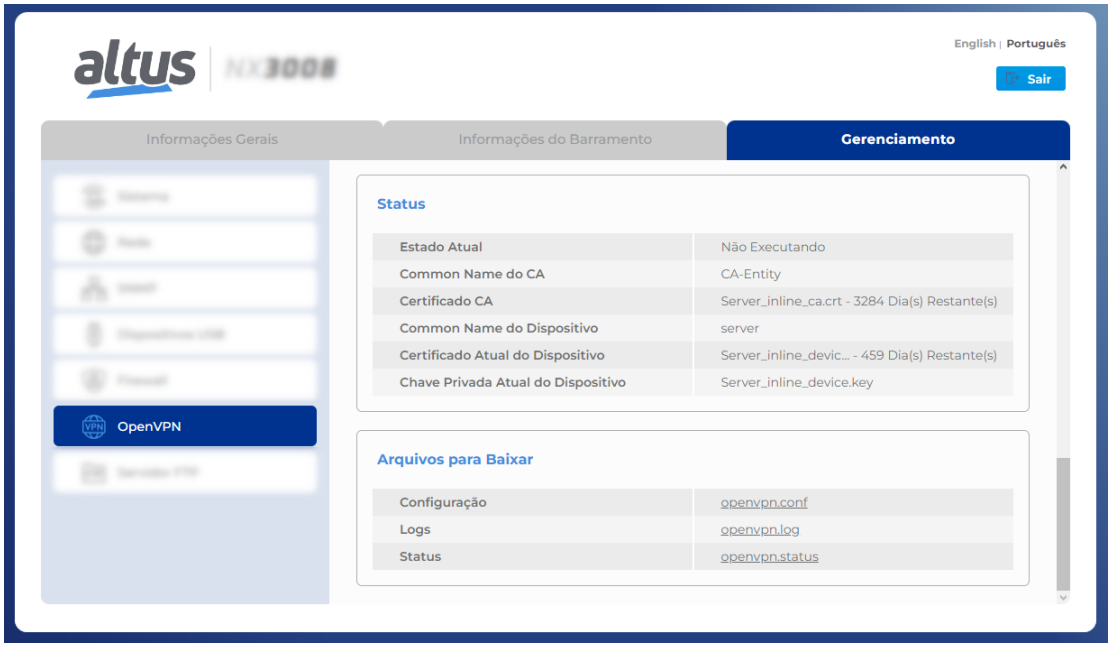


Figura 183: Tabela de Status OpenVPN com Funcionalidade Desativada

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

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

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



Figura 184: Tabela de Status OpenVPN com Funcionalidade Ativada

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

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

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

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

5.18.5. Arquivos para Baixar

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

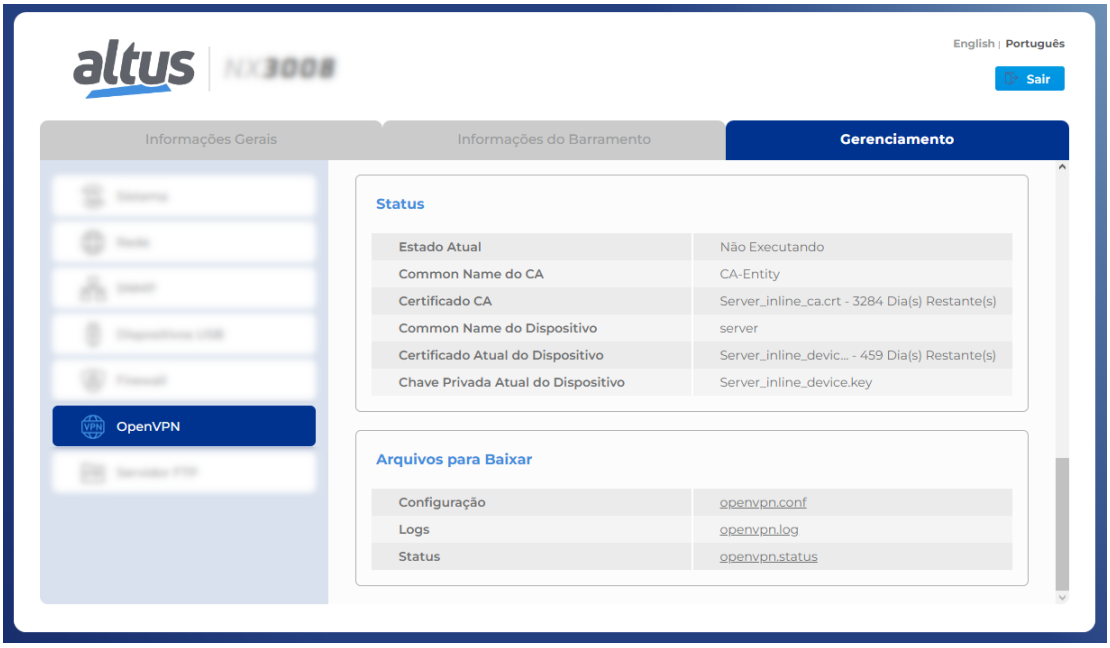


Figura 185: Seção de Downloads OpenVPN

5.18.6. Configuração de Arquiteturas

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

5.18.6.1. Host-to-Host



Figura 186: Exemplo de arquitetura Host-to-Host

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

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

5.18.6.2. Host-to-Site

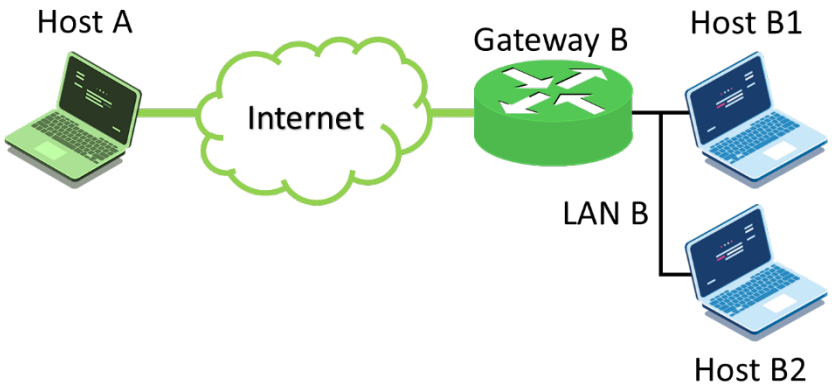


Figura 187: Exemplo de topologia Host-to-Site

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

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

Para que a comunicação entre todos os dispositivos da rede seja possível, é necessário criar regras de roteamento para o túnel da VPN. Consultar a seção Redes Privadas para verificar como criar regras de redes privadas.

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

Também é preciso informar o endereço da rede privada, Lan B, que estará realizando a comunicação através da VPN. Esta configuração é feita utilizando o comando *push "route IP_da_Lan_B Mascara_da_Lan_B"*, e é obrigatória independente se a rede privada está localizada a baixo do cliente ou do servidor do OpenVPN, porém, caso a rede privada esteja abaixo do cliente VPN, é necessário adicionar, além deste comando, a seguinte configuração: *route IP_da_Lan_B Mascara_da_Lan_B*. O comando *route* informa o servidor sobre qual é a rede privada que está conectada a rede da VPN e o comando utilizando o *push* faz com que os clientes daquele servidor VPN obtenham a mesma informação. Estas configurações são escritas no arquivo de configuração do servidor VPN.

5.18.6.3. Site-to-Site

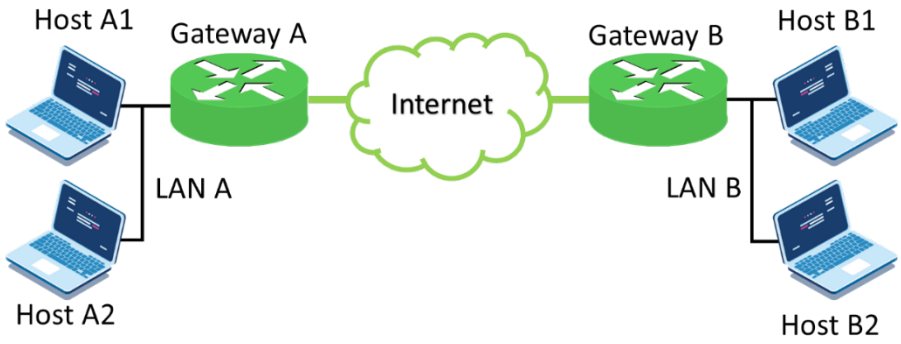


Figura 188: Exemplo de topologia Site-to-Site

Esta topologia permite a conexão entre dois hosts VPN, sendo que ambos atuam como gateways para a rede VPN. Através destes gateways é realizado o acesso para que seja estabelecida a comunicação entre os hosts A1, A2, B1, B2, Gateway A e Gateway B. Neste cenário, qualquer gateway pode assumir o papel de servidor, logo, o outro será o cliente.

Nenhum dos hosts que estão em uma rede privada, abaixo de um dos dois gateways, precisam ter suporte ao OpenVPN para conseguir se comunicar, pois, toda a comunicação é gerida pelos gateways da rede VPN.

Para que a comunicação entre todos os dispositivos da rede seja possível, é necessário criar regras de roteamento para o túnel da VPN. Consultar a seção Redes Privadas para verificar como criar regras de redes privadas.

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

5.19. Servidor FTP

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

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

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

ATENÇÃO

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

ATENÇÃO

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

5.19.1. Configuração

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

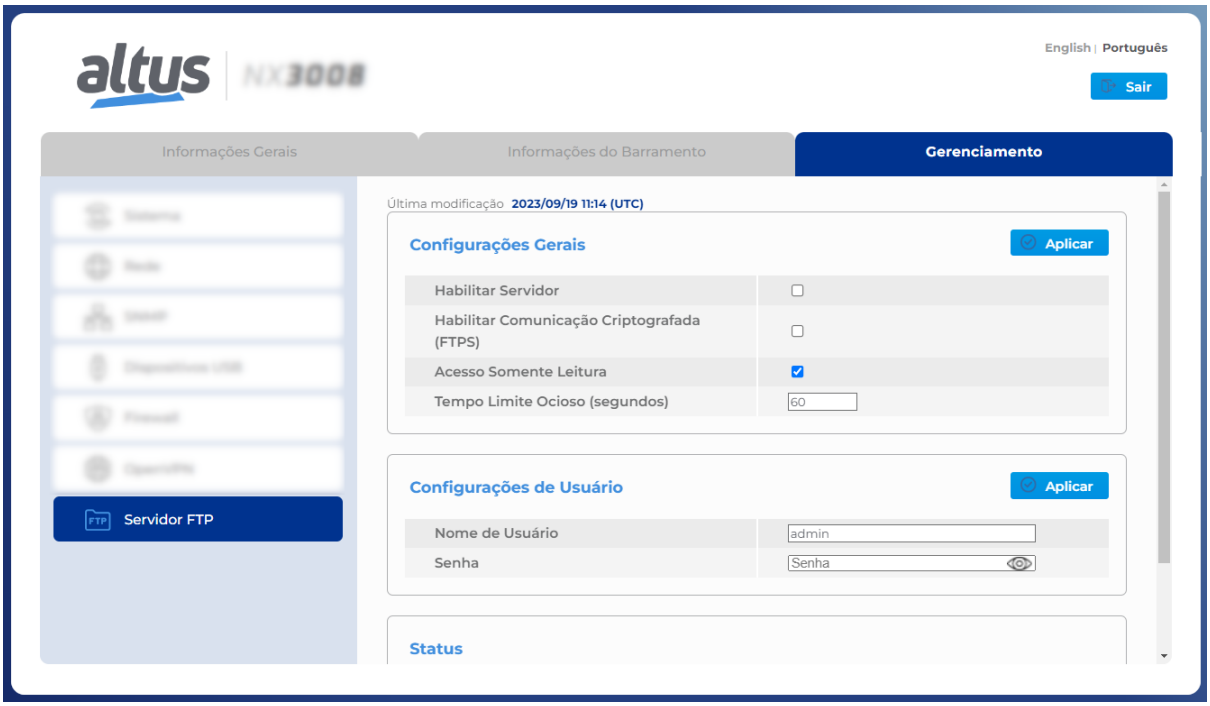


Figura 189: Tela de Configuração do Servidor FTP

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

ATENÇÃO

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

5.19.1.1. Configurações Gerais

5.19.1.1.1. Habilitar Servidor

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

5.19.1.1.2. Habilitar Comunicação Criptografada (FTPS)

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

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

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

ATENÇÃO

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

5.19.1.1.3. Acesso Somente Leitura

Permite habilitar e desabilitar a limitação do acesso de escrita e leitura no Servidor. Por padrão, o parâmetro é habilitado.

Quando o parâmetro está habilitado, o Cliente FTP tem acesso somente a leitura, sem a possibilidade de adicionar ou remover qualquer arquivo do Servidor FTP. Neste caso, a única operação permitida é o upload de arquivos, ou seja, transferência do Servidor para o Cliente. Já quando o parâmetro está desabilitado, é possível realizar todas as operações.

5.19.1.1.4. Tempo Limite Ocioso (Segundos)

Define o tempo máximo em que a conexão será mantida antes do Servidor FTP fecha-la por inatividade. Isto é, após aberta a conexão com um Cliente FTP, se não houver nenhuma atividade, após passado o Tempo Limite Ocioso, a conexão com o Cliente é fechada pelo Servidor.

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

5.19.1.2. Configurações de Usuário

5.19.1.2.1. Nome de Usuário

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

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

5.19.1.2.2. Senha

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

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

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

Tabela 161: Configurações de Usuário FTP

5.19.1.3. Status

5.19.1.3.1. Estado Atual

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

Cada configuração aplicada reiniciará o serviço.

5.19.1.3.2. Conexões Ativas

Mostra ao usuário o número atual de conexões ativas.

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

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

5.20. SysLog

Syslog (System Logging Protocol) é um protocolo que permite que dispositivos enviem mensagens de log para um sistema centralizado de coleta e armazenamento de logs. Essas mensagens podem incluir informações sobre eventos do sistema, erros, avisos, e outras atividades relevantes para a administração e segurança do sistema.

A configuração da UCP como Client SysLog pode ser feita a partir da seção *Sistema* na aba *Gerenciamento* da página Web de Sistema da UCP. Conforme é mostrado na figura abaixo.



Figura 190: Configurações do SysLog

5.20.1. Configuração do SysLog

A configuração do SysLog é composta por apenas dois campos, o checkbox *"Habilitar Serviço"* por padrão é desabilitado, sendo necessário marcar-lo para ativar o serviço. No campo *"Endereço IP do Servidor"* é configurado o endereço IP do servidor SysLog que será utilizado.

Após a configuração do endereço IP do servidor e a habilitação do serviço, as configurações serão aplicadas clicando no botão *"Aplicar"*. É importante que o servidor Syslog esteja devidamente configurado para comunicar com o cliente via protocolo UDP, utilizando a porta 514.

Quando configurado, o SysLog enviará para o servidor todos os logs presentes nos *Logs de Sistema* do Mastertool, como detalhado na seção Log de Sistema. O serviço do SysLog apenas envia logs de prioridade *WARNING* ou superior, como indicado na classificação abaixo:

Categoria do Mastertool	Categoria do Syslog	Prioridade
Exception	Emergency	0
Error	Error	3
Warning	Warning	4

Tabela 162: Prioridade Das Mensagens Do SysLog

6. Redundância com UCP NX3035

6.1. Introdução

Trata-se de uma redundância do tipo hot-standby, onde os controladores são duplicados. Um dos controladores normalmente está em estado Ativo e controlando o processo, enquanto o outro controlador normalmente está em estado Reserva, mantendo-se sincronizado com o controlador Ativo. Caso ocorra uma falha no controlador Ativo, que o impeça de continuar controlando o processo, o controlador Reserva chaveia automaticamente para Ativo, em um tempo suficientemente baixo para não perturbar o processo, sem causar descontinuidades nas saídas que controlam o processo.

A redundância hot-standby é um método utilizado para aumentar a tolerância a falhas e, consequentemente, aumentar a disponibilidade do sistema de automação. A ideia básica é que nenhuma falha simples em componentes duplicados cause a interrupção do controle do processo.

A redundância hot-standby é muito aplicada em:

- Plataformas de exploração de petróleo;
- Sistemas de geração e distribuição de energia;
- Processos contínuos, tais como plantas químicas, refinarias de petróleo, produção de celulose, etc.

Além dos controladores, podem ser também duplicadas, opcionalmente, as redes de campo (PROFIBUS DP e Ethernet), as redes de supervisão Ethernet, e as redes de controle Ethernet HSDN (High Speed Deterministic Network). Optando-se pela duplicação destas redes, obtém-se uma disponibilidade ainda maior.

A redundância hot-standby de CPs da Série Nexto não prevê duplicação de módulos de E/S. Caso a redundância de módulos de E/S seja desejável, ela pode ser tratada em nível de aplicação, pelo usuário final. Por exemplo, o usuário pode duplicar ou até mesmo triplicar um módulo de entradas analógicas, e criar um algoritmo de votação para determinar quais das entradas serão consideradas em determinado momento em sua aplicação.

A figura abaixo mostra um exemplo típico de arquitetura redundante com a UCP NX3035.

A parte central de um CP redundante é formada por dois bastidores idênticos, denominados CPA e CPB. No contexto da redundância, cada bastidor (CPA e CPB) é denominado half-cluster, enquanto o conjunto formado por estes dois bastidores é denominado cluster. A UCP NX3035 do CPA é denominada UCP A, enquanto a UCP NX3035 do CPB é denominada UCP B.

Dois canais de sincronismo conectam os CPA e CPB via fibra óptica através das portas NETA e NETB dos NX3035 (UCP A e UCP B), e são utilizados para o sincronismo da redundância entre os dois half-clusters.

ATENÇÃO

A configuração do PLC realizada através da página web do produto, não é sincronizada entre os dois controladores. Portanto, caso seja realizada alguma alteração na configuração do CP via página web, esta alteração deverá ser replicada no outro controlador, manualmente.

Neste exemplo existem diversas redes adicionais:

- Uma rede de campo PROFIBUS redundante (PROFIBUS 1A e PROFIBUS 1B).
- Uma rede Ethernet redundante de supervisão conectada nas portas NET1 e NET2 dos NX3035.
- Uma rede Ethernet redundante para a HSDN conectada nas portas NET3 e NET4 dos NX3035.
- Uma rede “Ethernet redundante 1” para outros usos conectada nas portas NET5 e NET6 dos NX3035.
- Uma rede “Ethernet redundante 2” para outros usos conectada nos dois primeiros NX5000 de cada half-cluster.
- Uma rede “Ethernet não redundante 3” para outros usos conectada no terceiro NX5000 de cada half-cluster.

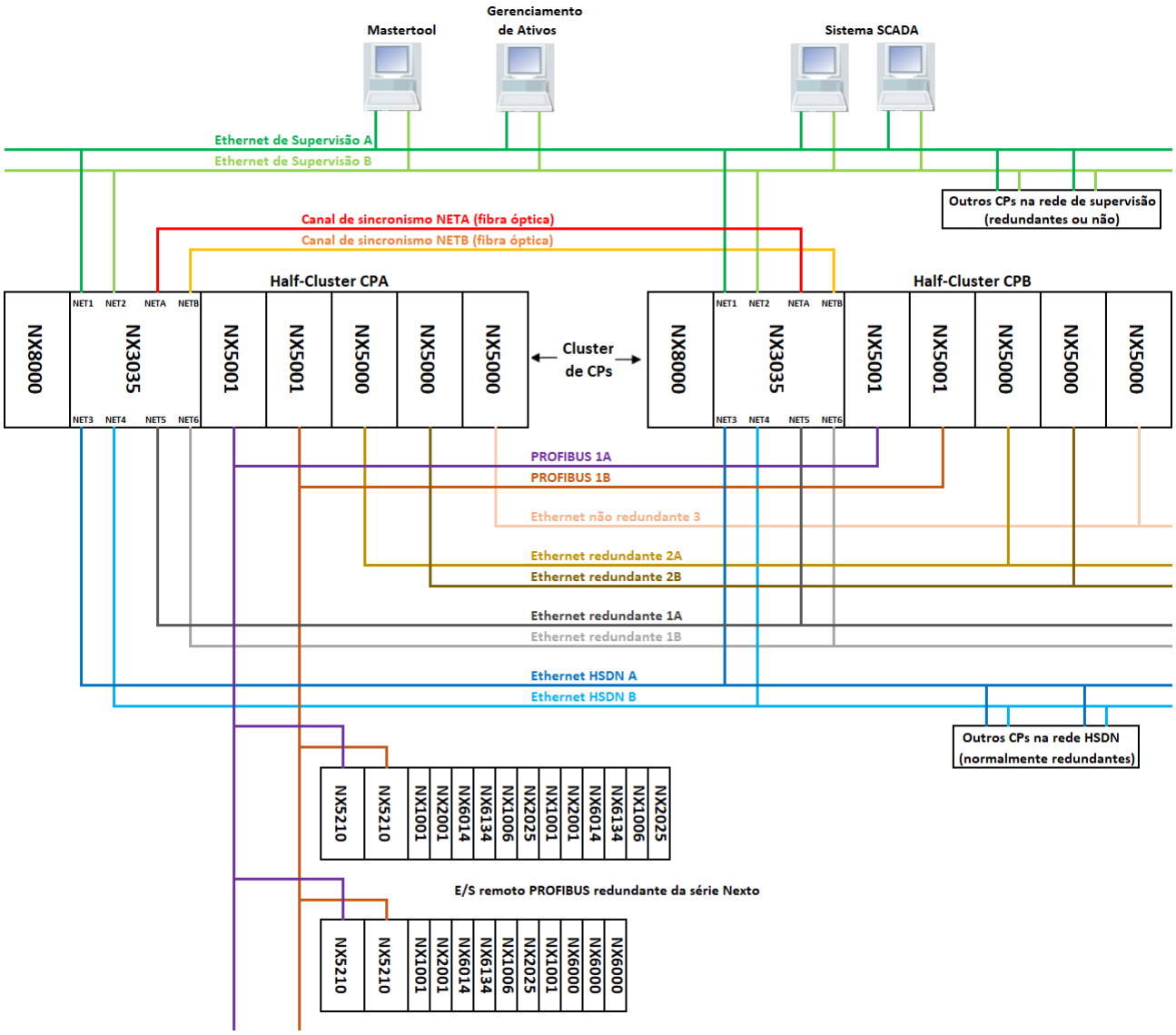


Figura 191: Exemplo de Arquitetura Redundante com UCP NX3035

6.2. Descrição Técnica e Configuração

6.2.1. Configuração Mínima de um CP Redundante

Um CP redundante é composto, no mínimo, de dois half-clusters indênticos.
Cada half-cluster é constituído, no mínimo, dos seguintes módulos:

- O bastidor onde os módulos são inseridos, que pode ser um dos seguintes:
  - NX9000, com 8 posições
  - NX9001, com 12 posições
  - NX9002, com 16 posições
  - NX9003, com 24 posições
- A fonte de alimentação NX8000, nas posições 0 e 1 do bastidor.
- A UCP NX3035, nas posições 2 a 5 do bastidor.

A figura abaixo mostra um exemplo de configuração mínima de um CP redundante, utilizando o bastidor NX9000 (8 posições). Ainda restam livres as posições 6 e 7 deste bastidor.

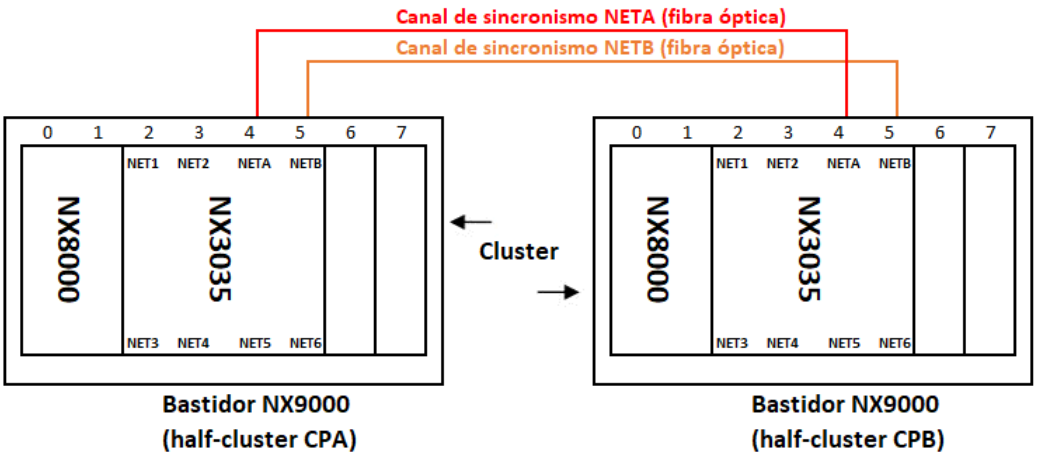


Figura 192: Configuração Mínima de um CP Redundante em Bastidor NX9000

6.2.2. Canais de Sincronismo entre CPA e CPB

A Figura 192 mostra que deve ser feita uma conexão entre os half-clusters CPA e CPB para efeito de sincronização entre os mesmos (troca de dados redundantes, diagnósticos, listas de forçamentos e projetos).

Esta conexão é feita através de duas portas de comunicação de 1 Gbps, denominadas NETA e NETB. Devem ser instalados dois cabos ópticos entre os dois half-clusters:

- Um entre as portas NETA das UCPA e UCP B.
- Outro entre as portas NETB das UCP A e UCP B.

As portas de comunicação NETA e NETB da UCP NX3035 estão localizadas na parte inferior da mesma, e são conectores SFP onde podem ser instalados dois tipos de transceptores:

- NX9500: Transceptor de fibra multimodo SFP Gigabit (550m)
- NX9501: Transceptor de fibra monomodo SFP Gigabit (10Km)

Estes transceptores podem ser instalados ou removidos à quente nos conectores NETA e NETB.

Para maiores detalhes sobre estes transceptores, tais como tipos de conectores ópticos e de fibra óptica compatíveis, ver a CT dos mesmos.

ATENÇÃO

A conexão entre as portas NETA da UCP A e NETA da UCP B, bem como a conexão entre as portas NETB da UCP A e NETB da UCP B, devem ser feitas diretamente entre os NX9500 ou NX9501. Nenhum equipamento intermediário (exemplo: switch) deve ser instalado no meio destas duas conexões.

Nenhum serviço de sincronização entre o CPA e o CPB irá funcionar caso sejam invertidos os cabos dos canais de sincronismo. Por exemplo, conectar o canal NETA da UCP A no NETB da UCP B e o canal NETB da UCP A no NETA da UCP B.

Em alguns casos, os half-clusters poderão estar bem próximos entre si, por exemplo, dentro do mesmo painel elétrico. Neste caso, pode-se utilizar o NX9500 e dois cabos ópticos de curta distância para interligá-los.

Em outros casos, é desejável que os dois half-clusters estejam distantes entre si, para evitar que ambos sejam danificados por um evento tal como incêndio, e a redundância seja perdida. Por exemplo, em aplicações em túneis, pode-se instalar o CPA numa sala de controle numa ponta do túnel, e o CPB numa sala de controle na outra ponta do túnel. Neste caso, dependendo da distância, deve-se utilizar o NX9501 e dois cabos ópticos de longa distância para interligá-los.

6.2.3. Configurações Típicas de um CP Redundante

Uma configuração mínima, como esta mostrada na Figura 192, já pode ser útil como uma "unidade de processamento redundante". Seria possível utilizar o canal de comunicação serial e Ethernet da UCP NX3035, por exemplo, para comunicação MODBUS TCP com um sistema SCADA, e comunicação MODBUS RTU e/ou MODBUS TCP com dispositivos de campo inteligentes ou sistemas de E/S remotas MODBUS.

Em configurações maiores são inseridos módulos adicionais nos half-clusters CPA e CPB, por exemplo, para prover um sistema de E/S remotas PROFIBUS e canais Ethernet adicionais.

Entre os módulos adicionais que, opcionalmente, podem ser inseridos nos half-clusters, pode-se citar:

- Mestres PROFIBUS NX5001
- Interfaces Ethernet NX5000

Também é possível instalar módulos de E/S não-redundantes em cada um dos half-clusters. Isso pode ser útil, por exemplo, para confeccionar um painel de botões e lâmpadas para mostrar o estado de redundância e chavear o estado de redundância.

Caso seja necessário, pode-se utilizar bastidores maiores, como o NX9001 (12 posições), NX9002 (16 posições) e NX9003 (24 posições).

Os módulos que podem ser inseridos nos bastidores ocupam as seguintes quantidades de posições:

- NX8000: 2 posições (sempre as posições 0 e 1)
- NX3035: 4 posições (sempre as posições 2 a 5)
- NX5001: 2 posições (a partir da posição 6)
- NX5000: 2 posições (a partir da posição 6)
- Módulos de E/S (a partir da posição 6): a maioria dos modelos de módulos de E/S ocupa 1 posição no bastidor, e poucos modelos utilizam 2 posições.

Ao inserir módulos em um bastidor, existem alguns limites, que são automaticamente verificados pelo programador Mastertool:

- O número de posições do bastidor selecionado
- A capacidade de corrente fornecida pela fonte NX8000 e consumida pelos demais módulos.
- A quantidade de determinados módulos:
  - NX8000: exatamente 1 deve ser inserido
  - NX3035: exatamente 1 deve ser inserido
  - NX5001: no máximo 6 podem ser inseridos
  - NX5000: no máximo 6 podem ser inseridos

6.2.3.1. Adição de Módulos NX5001 para Redes PROFIBUS

É possível inserir até seis módulos NX5001 em cada half-cluster para a utilização de redes PROFIBUS em um CP redundante.

Para criar uma rede PROFIBUS redundante, é necessário inserir dois módulos NX5001 em cada half-cluster. Para criar uma rede PROFIBUS simples, basta inserir um módulo NX5001 em cada half-cluster.

Instalando seis NX5001 em cada half-cluster, é possível configurar até três redes PROFIBUS redundantes, ou até seis redes PROFIBUS simples. Outras possibilidades também existem desde que não se ultrapassem o limite de seis módulos NX5001 (por exemplo, duas redes PROFIBUS redundantes e duas redes PROFIBUS simples).

Caso a rede PROFIBUS "n" (sendo "n" um número de 1 a 6) seja redundante, as duas redes que a compõem são nomeadas PROFIBUS "n"A e PROFIBUS "n"B. Caso a rede PROFIBUS "n" seja simples, a rede que a compõe é nomeada PROFIBUS "n"A.

No exemplo da Figura 191, existe apenas uma rede PROFIBUS (PROFIBUS 1), e a mesma é redundante (PROFIBUS 1A e PROFIBUS 1B). Neste exemplo, portanto, foram inseridos dois módulos NX5001 em cada half-cluster.

Mais informações sobre a configuração de redes PROFIBUS são fornecidas na seção Configuração de Redes PROFIBUS com Módulos NX5001.

6.2.3.2. Adição de Módulos NX5000 para Redes Ethernet

É possível inserir até seis módulos NX5000 em cada half-cluster para adicionar portas Ethernet remotas em um CP redundante, caso as seis portas Ethernet já presentes no NX3035 não sejam suficientes.

Cada NX5000 inserido nos dois half-clusters adiciona mais uma porta Ethernet de 100/10 Mbps que permite a conexão a uma rede Ethernet simples. Um par redundante de NX5000 inserido nos dois half-clusters adiciona duas portas Ethernet que permitem a conexão a uma rede Ethernet redundante, usando uma estratégia denominada NIC Teaming.

Instalando seis NX5000 em cada half-cluster, é possível configurar até três redes Ethernet redundantes, ou até seis redes Ethernet simples. Outras possibilidades também existem desde que não se ultrapasse o limite de seis módulos NX5000 (por exemplo, duas redes Ethernet redundantes e duas redes Ethernet simples).

No exemplo da Figura 191, existem 3 módulos NX5000 instalados em cada half-cluster. Os dois primeiros formam um par redundante (NIC teaming) conectado a uma rede Ethernet redundante, e o terceiro se conecta a uma rede Ethernet simples.

Mais informações sobre a configuração dos módulos NX5000 são fornecidas na seção Configuração de Interfaces Ethernet com Módulos NX5000.

6.2.3.3. Adição de Módulos de E/S

É possível instalar módulos de E/S nos bastidores locais de um half-cluster, embora isso não seja comum. Uma possível aplicação seria instalar um módulo NX1005 para implementar um painel de comando de redundância com botões e lâmpadas. Com os botões pode-se comandar mudanças de estado de redundância, e com as lâmpadas pode-se mostrar o estado atual da redundância.

6.2.4. Características Gerais de um CP Redundante

	Características Gerais de um CP redundante
UCPs permitidas	NX3035
Tipo de redundância	Hot-standby
Tolerância a falhas	Tolera, no mínimo, falha simples em equipamentos duplicados nos half-clusters. Em casos específicos, pode tolerar múltiplas falhas.
5 estados de redundância de um half-cluster	**Não-Configurado:** estado inicial, também considerado quando CP está desligado ou não está executando a tarefa principal (MainTask). **Inicializando:** estado temporário assumido depois de Não-Configurado, onde alguns testes determinarão o próximo estado (Inativo, Ativo, Reserva – ou de volta para Não-Configurado). **Inativo:** estado atingido depois de determinados tipos de falhas ou para manutenção programada. **Ativo:** controlando o processo do usuário. **Reserva:** apto para chavear para Ativo e controlar o processo do usuário, caso haja uma demanda (ex: falha do CP Ativo).
Principais falhas que causam switch-over entre CP Ativo e CP Reserva. CP Reserva chaveia para Ativo e o Ativo pode ir para Inativo ou Não-Configurado.	- Falta de alimentação no CP Ativo. - Falha na fonte do CP Ativo. - Falha na UCP do CP Ativo (parada de execução da tarefa principal - MainTask). - Falha de ambos os canais de sincronismo (NETA e NETB) com evidências que mostram que o CP Ativo não está mais controlando o processo (exemplo: CP reserva percebe que a rede PROFIBUS está sem mestre). Neste caso, o CP Reserva toma a iniciativa de chavear para Ativo. - Falha de ambos os canais de sincronismo (NETA e NETB) sendo que o CP Ativo sabe que a falha é local (exemplo: falha diagnosticada nos controladores Ethernet das portas NETA e NETB). Além disso, o CP Ativo deve saber que havia um CP em estado Reserva imediatamente antes da detecção desta falha local. Neste caso, o CP Ativo toma a iniciativa de chavear para Reserva, e consequentemente o CP Reserva chaveia para Ativo. - Falha em alguma rede PROFIBUS configurada como vital. - Falha em alguma rede Ethernet configurada como vital.

	Características Gerais de um CP redundante
Comandos que causam switch-over entre CP Ativo e CP Reserva	- Comandos recebidos do MasterTool ou de um sistema SCADA, através deste CP (local) ou do outro CP (remoto). - Comandos gerados pela aplicação do usuário, por exemplo, em função de outros diagnósticos (ex: falha de comunicação Ethernet), através deste CP (local) ou do outro CP (remoto). - Comando gerado via botão OTD da UCP.
Principais falhas que impedem um CP de ficar no estado Reserva, ou assumir o estado Reserva. Tais falhas fazem com que este CP assuma os estados Não-Configurado ou Inativo.	- Falta de alimentação neste CP. - Falha na fonte deste CP. - Falha na UCP deste CP (parada de execução da tarefa principal - MainTask). - Falha de ambos os canais de sincronismo (NETA e NETB) onde percebe-se que o outro CP está controlando o processo (exemplo: este CP percebe que existe um mestre na rede PROFIBUS). - Falha de ambos os canais de sincronismo (NETA e NETB) sendo que este CP sabe que a falha é local (exemplo: falha diagnosticada nos controladores Ethernet das portas NETA e NETB). - Falha no serviço de sincronismo de dados redundantes, mas os canais NETA ou NETB ainda informam que o outro CP está em estado Ativo. - Falha no serviço de sincronismo da lista de forçamentos redundantes, mas os canais NETA ou NETB ainda informam que o outro CP está em estado Ativo. - Falha total em alguma rede PROFIBUS configurada como vital. - Falha total em alguma rede Ethernet configurada como vital. - Projeto diferente daquele do CP Ativo, com sincronização automática de projetos habilitada. - Versão de firmware incompatível com CP Ativo.
Comandos que tiram o CP do estado Reserva	- Comandos recebidos do MasterTool ou de um sistema SCADA, através deste CP (local) ou do outro CP (remoto). - Comandos gerados pela aplicação do usuário, por exemplo, em função de outros diagnósticos (ex: falha de comunicação Ethernet), através deste CP (local) ou do outro CP (remoto). - Comando gerado via botão OTD da UCP.
Tempo de switch-over	- Tipicamente de um a três ciclos da MainTask, dependendo do estímulo para troca de estado (comando ou falha). - Em caso de falha na rede PROFIBUS, dois ciclos da MainTask + 500 ms.
Switch-over sem descontinuidades (bump-less)	- Um switch-over não provoca descontinuidades nas saídas do controlador, nem em variáveis internas.
Overhead da redundância (consumo de UCP a cada ciclo da MainTask adicionado pela redundância)	- Valor máximo calculado automaticamente pelo Mastertool e informado ao usuário. - Valor médio típico de 31 ms para 1 Mbytes de dados redundantes.
Visor da UCP	Entre outros diagnósticos, mostra o estado da redundância (Ativo, Reserva, Inativo, Não-Configurado, Inicializando). Para a identificação do CP, é necessário verificar no menu Redundancia, onde há a identificação do CPA ou CPB.
Diagnósticos da redundância	- Indicam falhas tanto no CPA como no CPB, independente dos seus estados (Ativo ou Não-Ativo). - Evitam "falhas ocultas". - Permitem manutenção rápida, o que é essencial para obter alta disponibilidade.
Comandos da redundância	- Permitem executar as mesmas ações de controle, além de outros comandos (ex: comandar switch-overs). - Podem ser executados no CP local, ou repassados para o outro CP (remoto) via canais de sincronismo NETA / NETB. - Podem ser recebidos a partir do MasterTool ou de um sistema SCADA. - Podem ser executados a partir da aplicação do usuário. - Podem ser executados com o botão OTD do NX3035.
Eventos da redundância	O log acessível via MasterTool registra modificações nos diagnósticos e comandos de redundância, com estampa de tempo, permitindo assim uma investigação das causas de um switch-over.
SNTP (Simple Network Time Protocol)	Permite que os eventos tenham estampa de tempo precisa e ajustada à hora mundial. Também sincroniza o relógio de tempo real do CP para outras aplicações.

	Características Gerais de um CP redundante
Sincronização de comandos e diagnósticos	A cada ciclo da MainTask, CPA e CPB trocam diagnósticos e comandos através dos canais de sincronismo NETA ou NETB. Desta maneira, um CP conhece os diagnósticos e comandos do outro.
Sincronização de dados redundantes	A cada ciclo da MainTask, o CP Ativo copia dados redundantes para o CP Não-Ativo através dos canais de sincronismo NETA e NETB. Dados não-redundantes não são sincronizados.
Sincronização da lista de forçamentos redundantes	A cada ciclo da MainTask, o CP Ativo copia a lista de forçamentos redundantes para o CP Não-Ativo através dos canais de sincronismo NETA e NETB. Esta lista inclui apenas variáveis redundantes forçadas. Desta maneira, CPA e CPB poderão ter diferentes conjuntos de dados não-redundantes forçados, pois estes forçamentos não são sincronizados.
Projeto único para CPA e CPB	Existe um projeto único comum para o CPA e CPB, gerado pelo MasterTool. O projeto é composto do projeto de aplicação (código executável) e do *project archive* (código fonte).
Identificação de um CP	Através do MasterTool, identifica-se uma UCP como CPA, CPB ou CP não-redundante. Esta identificação não faz parte do projeto de aplicação gerado pelo MasterTool, embora seja escrita em um CP usando o MasterTool. A identificação de cada CP viabiliza a característica de ter um projeto único para CPA e CPB.
Sincronização automática do projeto	Se o projeto do CP Ativo se tornar diferente do projeto do CP Não-Ativo, o mesmo é copiado do CP Ativo para o CP Não-Ativo. Esta sincronização ocorre em baixa prioridade e pode demorar vários segundos, e durante a sua execução o CP Não-Ativo não consegue atingir o estado Reserva. Deve-se lembrar que o projeto é composto do projeto de aplicação (código executável) e do código fonte (project archive), sendo ambos sincronizados. Esta sincronização pode ser desabilitada em circunstâncias especiais, para viabilizar modificações de projeto que só podem ser carregadas offline em CPs não-redundantes. Exemplos de tais circunstâncias especiais são descritos no capítulo Carga Offline de Programas sem Interrupção do Controle do Processo.
Sincronização automática das configurações de Firewall, SNMP, FTP, SysLog e Cartão de Memória	Se as configurações de alguma das funcionalidades do CP Ativo se tornarem diferentes das configurações do CP Não-Ativo, as mesmas configurações serão copiadas do CP Ativo para o CP Não Ativo desde que a Sincronização de Projetos esteja habilitada em ambos os CPs. Esta sincronização pode ser desabilitada em circunstâncias especiais, para viabilizar modificações de configurações sem a sincronização imediata. Não é permitida a alteração das configurações dessas funcionalidades enquanto o CP estiver no estado Reserva (SBY). Nos estados Inativo (INA) e Não-Configurado (NCF), a alteração é permitida, mas sujeita a sobrescrita pelo CP Ativo (ACT) caso a Sincronização de Projetos esteja habilitada e este CP assuma o estado Reserva (SBY).
Expansão a quente de módulos e remotas PROFIBUS	Existem modificações de projeto que não podem ser feitas à quente (online) em um CP não-redundante, tais como a inclusão de novos módulos ou remotas PROFIBUS. No entanto, valendo-se da redundância do CP e das redes PROFIBUS, foi definido um procedimento para atingir este objetivo, muito importante para sistemas que necessitam de alta disponibilidade (ver capítulo Carga Offline de Programas sem Interrupção do Controle do Processo).
Endereços IP privados para CPA e CPB	É possível conectar-se a um CP específico (CPA ou CPB) utilizando um endereço IP privado para o mesmo. Isto serve, por exemplo, para obter diagnósticos específicos desse half-cluster. Existe um endereço IP do CPA para cada interface local da UCP A (NET1 ... NET6), e também existe um endereço IP do CPB para cada interface local da UCP B (NET1 ... NET6), conforme descrito na seção Configuração de Endereços IP nas portas integradas NET1 ... NET6 de uma UCP NX3035 em um CP Redundante. Para módulos NX5000, também existem modos de endereçamento que possuem endereços específicos para os CPA e CPB, conforme descrito na seção Configuração de Endereços IP em Módulos NX5000 em um CP Redundante.

	Características Gerais de um CP redundante
IP Ativo	Estratégia que permite que clientes Ethernet se conectem a um servidor no CP Ativo, utilizando sempre o mesmo endereço IP. Isto evita a necessidade de complexos scripts para mudar o endereço IP quando ocorrem switch-overs devido à redundância. Existe um endereço IP Ativo para cada interface local do CP Ativo (NET1 ... NET6), conforme descrito na seção Configuração de Endereços IP nas portas integradas NET1 ... NET6 de uma UCP NX3035 em um CP Redundante. Para módulos NX5000, também existem modos de endereçamento que possuem um endereço IP Ativo, conforme descrito na seção Configuração de Endereços IP em Módulos NX5000 em um CP Redundante.
NIC Teaming	Estratégia que permite que duas interfaces Ethernet de um half-cluster formem um par redundante compartilhando o(s) mesmo(s) endereço(s) IP. Esta estratégia também pode ser utilizada em CPs não redundantes, e já foi descrita nas seções Configuração das Interfaces Ethernet Integradas e Configuração de Interfaces Ethernet com Módulos NX5000. Esta estratégia pode ser definida entre os seguintes pares de portas Ethernet: - NET1 e NET2 - NET3 e NET4 - NET5 e NET6 - Par redundante de módulos NX5000
Redes PROFIBUS e Configuração de Falhas Vitais	O CP suporta até seis redes PROFIBUS simples ou até três redes PROFIBUS redundantes. Para cada uma delas, é possível configurar se a falha total é considerada vital (causa switch-over) ou não. A falha total de uma rede PROFIBUS simples é constatada quando o mestre NX5001 não consegue se comunicar com nenhum dos escravos PROFIBUS desta rede. A falha total de uma rede PROFIBUS redundante é constatada quando os dois mestres NX5001 não conseguem se comunicar com nenhum dos escravos PROFIBUS das duas redes redundantes.
Tarefa de usuário única e cíclica	Somente uma tarefa de usuário é permitida, sendo denominada MainTask. Esta tarefa é cíclica.
POUs principais: NonSkippedPrg e ActivePrg	Na criação de um projeto redundante, o MasterTool gera automaticamente duas POUs do tipo programa vazias que devem ser preenchidas pelo usuário: - NonSkippedPrg: esta POU é executada nos dois CPs (CPA e CPB), independente do estado de redundância (Ativo ou Não-Ativo). Destina-se ao gerenciamento de diagnósticos e comandos especiais. - ActivePrg: esta POU é executada somente no CP Ativo, e destina-se ao controle do processo do usuário final.

Tabela 163: Características Gerais de um CP Redundante

6.2.5. Dados para Compra da Redundância

A configuração mínima de um CP redundante implica na aquisição dos seguintes módulos:

- Dois bastidores, que devem ser escolhidos entre quatro modelos disponíveis de acordo com os módulos que devem ser instalados:
  - NX9000: oito posições
  - NX9001: doze posições
  - NX9002: dezesseis posições
  - NX9003: vinte e quatro posições
- Dois NX8000
- Dois NX3035
- Quatro NX9500 ou NX9501

Além disso, pode ser necessário adquirir os seguintes módulos opcionais:

- Dois módulos NX5001 para cada rede PROFIBUS simples adicional
- Quatro módulos NX5001 para cada rede PROFIBUS redundante adicional
- Dois módulos NX5000 para cada rede Ethernet simples adicional
- Quatro módulos NX5000 para cada rede Ethernet redundante adicional (NIC Teaming)

6.3. Princípios de Funcionamento

Nesta seção, descrevem-se as funções de um CP redundante com UCP NX3035, seu comportamento e estados. Também são apresentados conceitos e restrições de programação e configuração que serão utilizados nas próximas seções.

6.3.1. Identificação de uma UCP NX3035

Uma UCP NX3035 possui um registro de identificação, não volátil, onde é possível identificá-la como:

- **Não-Redundante:** não pode ser utilizada em um CP redundante (configuração padrão de fábrica)
- **CPA:** utilizada no CPA de um CP redundante
- **CPB:** utilizada no CPB de um CP redundante

O registro de identificação pode ser ajustado com o programador MasterTool. A primeira coisa a ser feita em uma UCP NX3035, antes de carregar o projeto do CP redundante na mesma, é identificá-la como CPA ou CPB. Caso a identificação não seja realizada, não é possível carregar um projeto redundante no CP.

ATENÇÃO

O registro de Identificação do CP não faz parte do projeto do CP redundante, e portanto não é salvo como parte deste projeto no computador onde o MasterTool é executado. O registro é salvo apenas na memória não volátil da UCP.

6.3.2. Projeto Redundante Único

Existe um único projeto para o CP redundante, idêntico para os CPA e CPB.

Parâmetros de configuração que devem ser diferentes entre CPA e CPB (ex: endereços IP de interfaces Ethernet) aparecem duplicados no projeto do CP redundante (um para CPA, outro para CPB). Cada CP irá considerar aquele que lhe corresponde, após analisar seu registro de identificação.

6.3.3. Funcionalidades Configuradas na Página Web

Algumas funcionalidades configuradas na página web do CP redundante, serão idêndicas para os CPA e CPB.

Estas funcionalidade estão listadas na tabela de Características Gerais de um CP Redundante.

6.3.4. Estrutura do Projeto Redundante

6.3.4.1. Redundancy Template

O projeto de um CP redundante é criado automaticamente a partir de um modelo, denominado *Redundancy Template*.

O modelo parte da configuração mínima de um CP redundante, conforme definido na seção Configuração Mínima de um CP Redundante. Além disso, alguns diálogos com o usuário são realizados para seleção do bastidor e inserção de módulos adicionais nos half-clusters, tais como mestres PROFIBUS (NX5001) e módulos Ethernet (NX5000).

Remotas PROFIBUS devem ser inseridas pelo usuário, abaixo dos mestres PROFIBUS NX5001 já inseridos.

Adicionalmente, são criadas tarefas e POUs básicas do tipo programa, conforme descrito nas próximas seções.

6.3.4.2. Tarefa Única e Cíclica MainTask

O projeto de um CP redundante possui uma única tarefa, denominada MainTask, que é cíclica. O usuário deve ajustar o intervalo da tarefa conforme suas necessidades.

6.3.4.3. Programa MainPrg

A tarefa MainTask está associada a uma única POU do tipo programa, denominada MainPrg. O programa MainPrg é criado automaticamente.

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

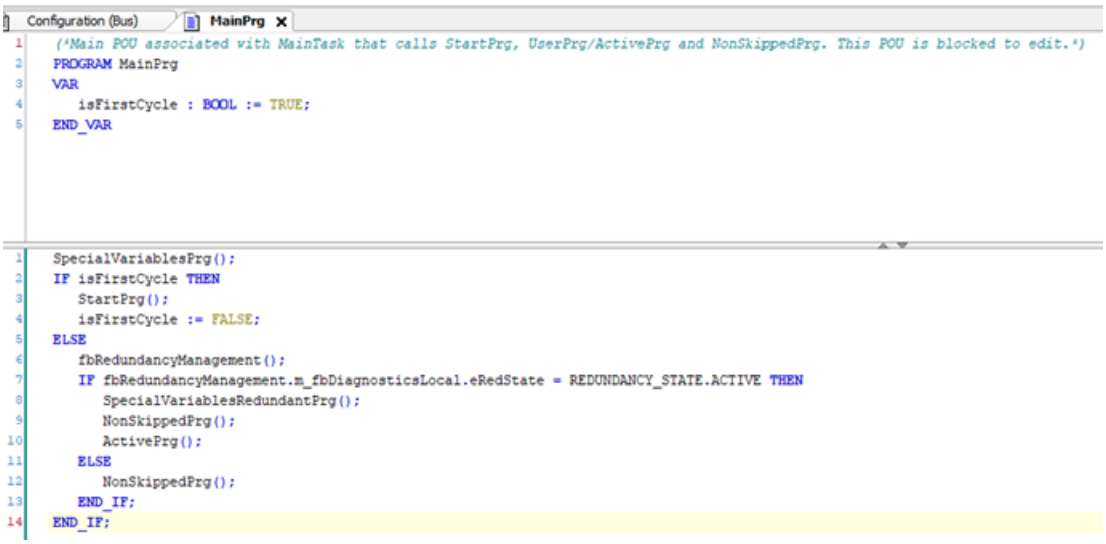


Figura 193: Código do programa MainPrg

MainPrg chama outras três POUs do tipo programa, denominadas *NonSkippedPrg*, *ActivePrg* e *SpecialVariablesRedundantPrg*.

A POU *NonSkippedPrg* sempre é executada, tanto no CP Ativo quanto no Não-Ativo.

As POUs *ActivePrg* e *SpecialVariablesRedundantPrg* somente são executadas quando o CP se encontra em estado Ativo.

As POUs *MainPrg* e *SpecialVariablesRedundantPrg* estão bloqueadas para edição, portanto não podem ser modificadas pelo usuário.

O usuário pode modificar os programas *NonSkippedPrg* e *ActivePrg*. Inicialmente, quando o projeto redundante é criado a partir do *Redundancy Template*, estes dois programas são criados "vazios", mas o usuário poderá inserir código nos mesmos.

ATENÇÃO

Quando a opção de comunicação OPC DA for habilitada durante a criação do projeto, o programa *NonSkippedPrg* não será criado vazio. Para maiores informações, consultar a seção Uso de Comunicação OPC DA com Projetos Redundantes.

6.3.4.4. Programa ActivePrg

O principal objetivo deste programa, que é executado somente no CP Ativo, é controlar o processo do usuário final.

Este programa normalmente atua sobre as variáveis redundantes, entre as quais encontram-se as variáveis de representação direta %I e %Q associadas ao sistema de E/S remotas. Para maiores informações, consultar a seção Programação de um CP Redundante - Configurações da MainTask - Programa ActivePrg.

6.3.4.5. Programa NonSkippedPrg

Este programa, executado em ambos os CPs (CPA e CPB) independente do estado de redundância, é tipicamente utilizado para funções como:

- Organizar diagnósticos não-redundantes para reportar a um sistema SCADA.
- Receber e executar comandos não-redundantes a partir de um sistema SCADA.

- Gerenciar condições de switch-over normalmente não contempladas automaticamente pelo CP redundante, que podem variar de usuário para usuário. Por exemplo, determinado usuário poderá executar um switch-over para o CP Reserva se o CP Ativo não estiver se comunicando com o sistema SCADA, enquanto outro usuário pode não desejar um switch-over nesta situação.
- Habilitar ou desabilitar drivers de E/S em função do estado de redundância, por exemplo, desabilitar um mestre MODBUS RS-485 no CP Não-Ativo.
- Detectar falhas em drivers de E/S em um CP Não-Ativo para evitar falhas ocultas. Alguns drivers de E/S não contemplam detecção automática de tais falhas. Outros drivers de E/S, como o PROFIBUS, o fazem automaticamente.
- Outras atividades que, por algum motivo, precisam ser executadas tanto no CP Ativo como no CP Reserva.

Para maiores informações, consultar a seção Programação de um CP Redundante - Configurações da MainTask - Programa NonSkippedPrg.

6.3.4.6. Variáveis Redundantes e Não-Redundantes

As variáveis de um CP redundante podem ser classificadas entre redundantes e não-redundantes. Variáveis redundantes são copiadas do CP Ativo para o CP Não-Ativo, no início de cada ciclo da MainTask, através dos canais de sincronismo NETA e NETB. Por outro lado, variáveis não-redundantes não são copiadas entre half-clusters e, portanto, podem ter valores diferentes nos CPs A e B.

As variáveis não-redundantes são utilizadas para armazenar informações privativas de cada half-cluster (CPA ou CPB), tais como diagnósticos de um módulo dentro deste half-cluster, incluindo os diagnósticos da redundância (estado da redundância deste half-cluster, etc...). Por exemplo, a falha de um módulo NX5001 instalado na posição 6 do bastidor do CPA pode ter um valor diferente da falha do módulo NX5001 correspondente instalado na posição 6 do bastidor do CPB. Além disso, o estado de redundância costuma ser diferente (por exemplo: CPA em estado Ativo, CPB em estado Reserva).

As variáveis redundantes dizem respeito às informações compartilhadas e relativas ao controle do processo. As variáveis associadas aos módulos de E/S remotos são exemplos típicos de variáveis redundantes. De maneira geral, uma variável precisa ser redundante quando o valor da mesma no ciclo atual da aplicação pode influenciar o comportamento da aplicação no ciclo seguinte. Desta forma, se ela não fosse copiada do CP Ativo para o CP Reserva, haveria uma descontinuidade em caso de switchover (se o CP Reserva passasse para Ativo). Por exemplo, no trecho de código seguinte, a variável COUNTER precisa ser redundante:

```
COUNTER := COUNTER + 1;
```

Se a variável COUNTER não fosse redundante, em caso de switchover o valor da variável COUNTER no CP Reserva poderia ser diferente do valor da variável COUNTER no CP Ativo. Desta forma, o valor da variável COUNTER poderia retroceder para um valor menor do que o valor atual, ou então avançar para um valor maior do que o valor atual, dependendo do valor que estivesse armazenado no CP Reserva. Em ambos os casos, haveria uma descontinuidade no comportamento da aplicação.

Existem também variáveis que podem ser redundantes ou não-redundantes, sem que isso afete a correção da aplicação do usuário. Não é muito fácil identificar este tipo de variáveis. Como exemplo poderíamos citar variáveis auxiliares usadas para cálculos intermediários dentro de uma POU.

É fácil identificar quando uma variável deve ser não-redundante. Conforme explicado anteriormente, trata-se de uma variável que representa uma informação privativa de cada half-cluster, por exemplo, um diagnóstico específico do CPA ou do CPB.

Por outro lado, não é muito fácil identificar variáveis que poderiam ser redundantes ou não redundantes sem afetar a correção de aplicação. Por este motivo, sugere-se a seguinte estratégia:

1. Identifique se a variável deve ser não-redundante. Neste caso, declare-a numa POU ou GVL não redundante.
2. Declare todas as demais variáveis como redundantes, mesmo que elas pudessem ser não-redundantes.

Esta estratégia poderá falhar se não houver quantidade suficiente de memória redundante. Neste caso, será necessário identificar algumas variáveis que poderiam ser redundantes ou não-redundantes sem afetar a correção da aplicação, e declará-las como não redundantes.

As próximas seções descrevem os limites de memória redundante para variáveis de representação direta (%I, %Q e %M) e para variáveis simbólicas, e também descrevem como declará-las com redundantes ou não-redundantes.

6.3.4.7. Variáveis %I Redundantes e Não-Redundantes

A UCP NX3035 aloca 384 Kbytes de variáveis %I (%IB0 ... %IB393215):

- Os primeiros 368 Kbytes podem ser redundantes ou não-redundante (%IB0 ... %IB376831)
- Os últimos 16 kbytes nunca serão redundantes (%IB376832 ... %IB393215)

A área de 368 Kbytes que pode ser redundante é alocada para entradas de sistemas de E/S remotos, tais como:

- Variáveis de entrada de módulos de I/O instalados em remotas PROFIBUS.
- Variáveis de entrada de mapeamentos de leitura de um cliente MODBUS.

A área de 16 Kbytes nunca redundante é alocada para diagnósticos privados rápidos de um half-cluster e para módulos de E/S instalados no mesmo bastidor local onde está instalado o NX3035. Diagnósticos rápidos são aqueles que, necessariamente, devem ser atualizados a cada ciclo da MainTask. Temos os seguintes exemplos de variáveis alocadas nesta área nunca redundante:

- Diagnósticos rápidos de cada NX5001
- Diagnósticos rápidos de cada NX5000
- Variáveis de entrada de módulos de E/S instalados no barramento local do NX3035.

O usuário pode configurar a quantidade de variáveis %I redundantes, entre 0 bytes e 376832 bytes, em múltiplos de 1 byte (o valor padrão é 16384 bytes - %IB0 ... %IB16383).

A figura seguinte ilustra a alocação de variáveis de representação direta %I redundantes e não-redundantes, onde os seguintes parâmetros podem ser configurados no MasterTool:

- **RI:** quantidade em bytes de %I configurados como redundantes.

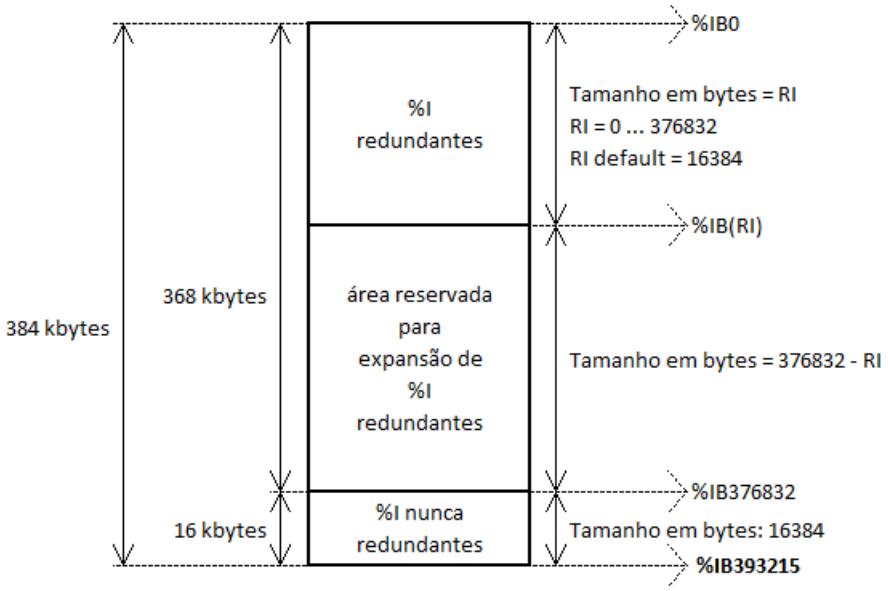


Figura 194: Alocação de %I Redundantes e Não-Redundantes

6.3.4.8. Variáveis %Q Redundantes e Não-Redundantes

A UCP NX3035 aloca 384 Kbytes de variáveis %Q (%QB0 ... %QB393215):

- Os primeiros 368 Kbytes podem ser redundantes ou não-redundante (%QB0 ... %QB376831)
- Os últimos 16 kbytes nunca serão redundantes (%QB376832 ... %QB393215)

A área de 368 Kbytes que pode ser redundante é dividida em duas seções.

A primeira seção que pode ser redundante sempre inicia em %QB0. Seu tamanho pode variar entre 0 e 368 kbytes, mas o tamanho default é de 16 kbytes. Esta seção é alocada para saídas de sistemas de E/S remotos, tais como:

- Variáveis de saída de módulos de I/O instalados em remotas PROFIBUS.
- Variáveis de saída de mapeamentos de escrita de um cliente MODBUS.
- Variáveis disponibilizadas para leitura/escrita em mapeamentos de um servidor MODBUS.

A segunda seção que pode ser redundante pode iniciar a partir de %QB0, mas o endereço inicial default é %QB262144. Seu tamanho pode variar entre 0 e 368 kbytes, mas o tamanho default é de "8 kbytes + NRP * 16 kbytes", onde NRP é o número de redes PROFIBUS geradas pelo Wizard do Mastertool ao criar o projeto (valor máximo de NRP = 6). Esta seção é alocada para diagnósticos de sistemas de E/S redundantes. Ao contrário dos diagnósticos rápidos (alocados em %I), tais diagnósticos alocados em %Q podem levar mais do que um ciclo da MainTask para serem atualizados. Como exemplos de tais diagnósticos podemos citar:

- Diagnósticos de mapeamentos de clientes MODBUS.
- Diagnósticos de escravos PROFIBUS, que se dividem em:
  - Diagnósticos de cabeças remotas PROFIBUS.
  - Diagnósticos de módulos de I/O instalados em remotas PROFIBUS.

Os últimos 16 kbytes, nunca redundantes (%QB376832 ... %QB393215), servem para alocar, por exemplo:

- Diagnósticos do NX3035
- Diagnósticos de cada NX5001
- Comandos especiais para cada NX5001
- Diagnósticos de cada NX5000
- Comandos especiais para cada NX5000
- Diagnósticos de cada cliente MODBUS
- Desabilitação dos mapeamentos de cada device de cada cliente MODBUS
- Diagnósticos de cada servidor MODBUS
- Desabilitação dos mapeamentos de cada servidor MODBUS
- Diagnósticos de módulos de I/O instalados no bastidor local do NX3035
- Variáveis de saída de módulos de I/O instalados no bastidor local do NX3035

A figura seguinte ilustra a alocação de variáveis de representação direta %Q redundantes e não-redundantes, onde os seguintes parâmetros podem ser configurados no MasterTool:

- **RQS:** quantidade em bytes da primeira seção de %Q redundantes, utilizada para saídas de sistemas de E/S remotos.
- **INI_RQD:** endereço inicial da segunda seção de %Q redundantes, utilizada para diagnósticos de sistemas de E/S remotos.
- **RQD:** quantidade em bytes desta segunda seção de %Q redundantes.

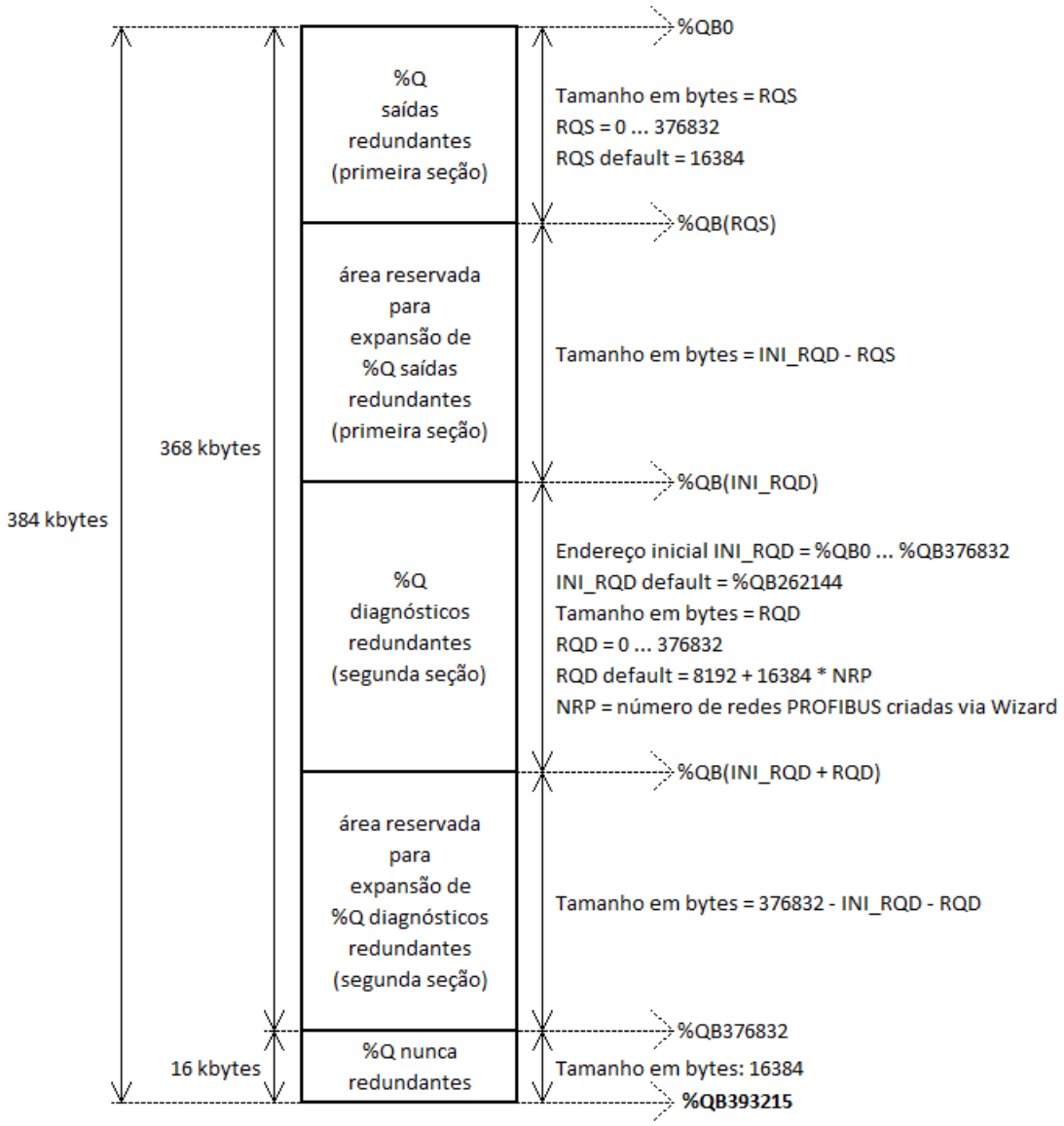


Figura 195: Alocação de %Q Redundantes e Não-Redundantes

A segunda seção de variáveis %Q redundantes, com endereço inicial INI_RQD e tamanho RQD, se divide em "1 + NRP" subseções, onde NRP é o número de redes PROFIBUS configuradas.

- A primeira subseção é destinada para diagnósticos redundantes não associados com redes PROFIBUS. Pode ser utilizada, por exemplo, para diagnósticos de mapeamentos MODBUS clientes. O endereço inicial desta subseção é INI_RQD. Seu tamanho default ao criar o projeto com o Wizard do MasterTool é 8192 bytes.
- Cada uma das próximas NRP subseções é destinada para diagnósticos de escravos de uma rede PROFIBUS. O endereço inicial de cada uma destas subseções é definido por um parâmetro do NX5001 mestre da rede PROFIBUS correspondente. Trata-se do parâmetro "Endereço Inicial de Diagnósticos dos Escravos em %Q". O tamanho default de cada uma destas subseções ao criar o projeto com o Wizard do MasterTool é 16384 bytes

. A figura seguinte mostra a organização das subseções da segunda seção de variáveis %Q redundantes. Através do MasterTool, o usuário consegue configurar os seguintes parâmetros mostrados nesta figura:

- **INI_RQD:** já descrito anteriormente.
- **INI_PB1, INI_PB2, ..., INI_PB(NRP):** parâmetros "Endereço Inicial de Diagnósticos dos Escravos em %Q" dos NX5001 mestres das NRP redes PROFIBUS configuradas.

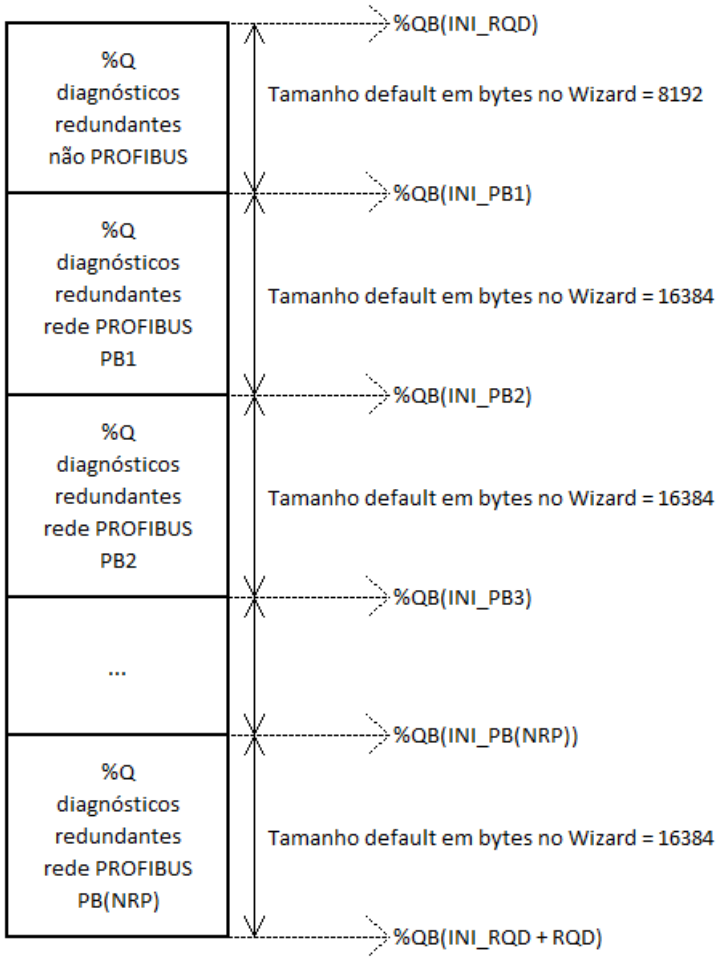


Figura 196: Subseções dos %Q Redundantes para Diagnósticos de E/S Remoto

6.3.4.9. Variáveis %M Redundantes e Não-Redundantes

A UCP NX3035 aloca 128 Kbytes de variáveis %M (%MB0 ... %MB131071).

Todos os 128 kbytes podem ser redundantes. Por padrão, a quantidade de variáveis %M redundantes é 0.

O uso de variáveis %M redundantes e até mesmo não redundantes deve ser evitado, preferindo-se a utilização de variáveis simbólicas (ver seção Variáveis Simbólicas Redundantes e Não-Redundantes).

A figura seguinte ilustra a alocação de variáveis de representação direta %M redundantes e não-redundantes, onde os seguintes parâmetros podem ser configurados no MasterTool:

- **RM:** quantidade em bytes de variáveis %M redundantes.

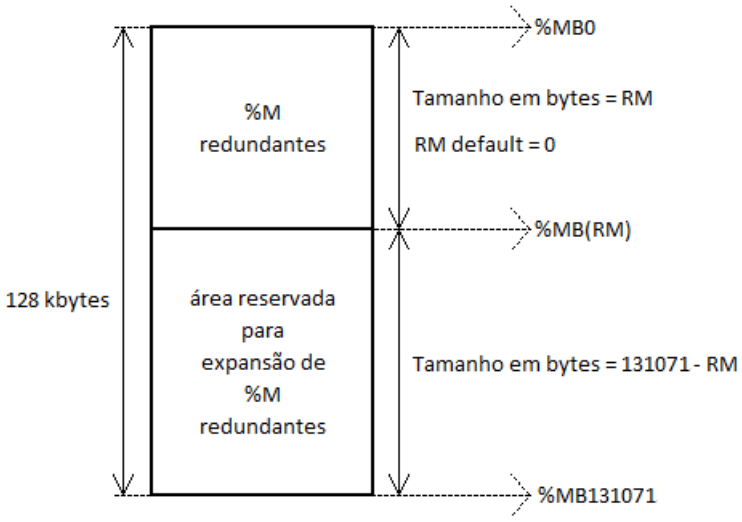


Figura 197: Alocação de %M Redundantes e Não-Redundantes

6.3.4.10. Variáveis Simbólicas Redundantes e Não-Redundantes

Além das variáveis de representação direta (%I, %Q e %M) que são alocadas automaticamente, o usuário pode declarar explicitamente variáveis simbólicas, dentro de POUs ou GVLs.

O tamanho máximo permitido para alocação de variáveis simbólicas redundantes e não-redundantes pode ser consultado na Tabela 7.

ATENÇÃO

Não se deve confundir variáveis simbólicas com variáveis simbólicas endereçadas através da diretiva AT. Variáveis simbólicas que utilizam a diretiva AT são nomes simbólicos atribuídos a variáveis de representação direta (%I, %Q e %M). Portanto, variáveis que utilizam a diretiva AT não alocam nenhuma memória de variáveis simbólicas. Para maiores detalhes, ver seção Variáveis Simbólicas Endereçadas com a Diretiva AT.

Algumas POUs geradas automaticamente pelo Wizard ao criar o projeto são não-redundantes, ou seja, as variáveis simbólicas declaradas dentro da mesma são não-redundantes:

- MainPrg
- NonSkippedPrg
- SpecialVariablesPrg

Algumas GVLs geradas automaticamente pelo Wizard ao criar o projeto são não-redundantes, ou seja, as variáveis simbólicas declaradas dentro da mesma são não-redundantes:

- System_Diagnostics
- Module_Diagnostics
- ReqDiagnostics
- IOQualities
- Special_Variables

Todas as demais POUs e GVLs geradas automaticamente pelo Wizard ao criar o projeto são redundantes.

Quando o usuário cria uma POU ou GVL, a mesma é definida como redundante. Normalmente esta é a situação desejada pelo usuário, mas em alguns casos o usuário poderá desejar modificá-la para não redundante. Por exemplo, uma POU chamada dentro da POU NonSkippedPrg para tratamento de diagnósticos tipicamente deverá ser não redundante.

Através do objeto *Redundancy Configuration* na árvore de dispositivos do MasterTool, o usuário pode definir se uma POU ou GVL será redundante ou não. Algumas POUs e GVLs geradas automaticamente pelo Wizard, no entanto, tem o atributo de redundância fixo (bloqueado para edição do usuário).

Os critérios para definir se uma variável deve ser redundante ou foram discutidos na seção Variáveis Redundantes e Não-Redundantes.

As seguintes observações adicionais sobre a redundância de variáveis simbólicas devem ser consideradas:

- Variáveis definidas em POUs do tipo FUNCTION são não redundantes (mesmo que tenham o atributo VAR STATIC, o que não é recomendado por ser uma má prática de programação).
- A mera declaração de uma POU do tipo FUNCTION BLOCK não aloca nenhuma memória. Mas quando um FUNCTION BLOCK é instanciado dentro de uma POU do tipo PROGRAM ou dentro de uma GVL, esta instância alocará memória. Esta memória será redundante ou não redundante, de acordo com a configuração de redundância da POU ou GVL onde o FUNCTION BLOCK foi instanciado (objeto *Redundancy Configuration* na árvore de dispositivos do MasterTool).

6.3.4.11. Variáveis Simbólicas Endereçadas com a Diretiva AT

A diretiva AT pode ser utilizada dentro de GVLs para designar nomes simbólicos para variáveis de representação direta (%I, %Q e %M).

Por exemplo, se a variável %IW2120 corresponde a uma entrada analógica que mede o nível de um tanque, podemos dar um nome significativo para mesma dentro de um GVL ou POU, usando uma declaração como a seguinte:

```
TANK_LEVEL AT %IW2110 : WORD;
```

Portanto, variáveis declaradas usando a diretiva AT não alocam memória simbólica. Elas apenas funcionam como um "apelido" para uma variável de representação direta alocada como %I, %Q ou %M.

Variáveis nomeadas usando a diretiva AT só podem ser declaradas em POUs ou GVLs marcadas como "não-redundantes" no objeto *Redundancy Configuration* da árvore de dispositivos do Mastertool. Isso é válido mesmo que a variável de representação direta seja redundante.

Caso a diretiva AT seja empregada dentro de um POU ou GVL marcada como redundante, o Mastertool reportará um erro de compilação.

6.3.5. Estruturas de Dados de Diagnósticos, Comandos e Informações do Usuário

Cada CP dispõe de diversas estruturas de dados relacionadas à redundância, abaixo da estrutura de dados *DG_NX3035.tRedundancy*.

As seguintes estruturas são variáveis simbólicas que utilizam a diretiva AT para mapear variáveis do tipo %Q:

- **RedDgnLoc:** contém diagnósticos deste CP (local) relacionados com a redundância, como por exemplo, o estado da redundância deste CP.
- **RedDgnRem:** é uma cópia da estrutura *RedDgnLoc* do outro CP (remoto), recebida via canais de sincronismo NETA/NETB. Desta forma, este CP (local) tem acesso aos diagnósticos do outro CP (remoto).
- **RedCmdLoc:** contém comandos que devem ser aplicados no CP local (quando têm sufixo *Local*) ou então no CP remoto (quando têm sufixo *Remote*). Por exemplo o campo *bStandbyLocal* desta estrutura de dados solicita que o CP local vá para o estado *Standby*, enquanto o campo *bStandbyRemote* solicita que o CP remoto vá para o estado *Standby*.
- **RedCmdRem:** é uma cópia da estrutura *RedCmdLoc* do CP remoto, recebida via canais de sincronismo NETA/NETB. Desta forma, este CP (local) pode executar comandos que tenham o sufixo *Remote* recebidos do outro CP (remoto).
- **RedUsrLoc:** contém 128 bytes de dados preenchidos livremente pelo usuário no CP local e transmitidos para o CP remoto via canais de sincronismo NETA/NETB. Um uso típico desta área é transmitir dados privados num sentido não usual, do CP não-ativo para o CP ativo.
- **RedUsrRem:** é uma cópia da estrutura *RedUsrLoc* do outro CP, recebida via canais de sincronismo NETA / NETB. Um uso típico desta área é transmitir dados privados num sentido não usual, do CP não-ativo para o CP ativo.

Na seção Manutenção da Redundância, a subseção Estrutura de Diagnósticos, Comandos e Informações do Usuário para a Redundância fornecerá maiores detalhes sobre estas estruturas de dados.

6.3.6. Serviços de Sincronização Cíclicos através de NETA e NETB

Esta seção descreve os três serviços de sincronização que ocorrem ciclicamente em um CP redundante, entre CPA e CPB, através dos canais de sincronismo NETA e NETB.

Estes serviços são executados no início de cada ciclo da MainTask, e na sequência em que aparecem abaixo, ou seja:

- Primeiro, executa-se o serviço Troca de Diagnósticos e Comandos.
- Segundo, executa-se o serviço Sincronização de Dados Redundantes.
- Terceiro, executa-se o serviço Sincronização da Lista de Forçamentos Redundantes.

6.3.6.1. Troca de Diagnósticos e Comandos

Este serviço é responsável pelo intercâmbio das seguintes estruturas de dados, em cada ciclo da MainTask:

- Copiar *RedDgnLoc* do CPA para *RedDgnRem* do CPB
- Copiar *RedCmdLoc* do CPA para *RedCmdRem* do CPB
- Copiar *RedUsrLoc* do CPA para *RedUsrRem* do CPB
- Copiar *RedDgnLoc* do CPB para *RedDgnRem* do CPA
- Copiar *RedCmdLoc* do CPB para *RedCmdRem* do CPA
- Copiar *RedUsrLoc* do CPB para *RedUsrRem* do CPA

O serviço será executado utilizando apenas um dos canais de sincronismo (NETA ou NETB). Desta maneira, o serviço pode ser completado mesmo que um dos canais esteja com problemas.

6.3.6.2. Sincronização de Dados Redundantes

Este serviço é responsável pela transferência de variáveis redundantes, do CP Ativo para o CP Não-Ativo. Conforme visto anteriormente, existem variáveis redundantes simbólicas e também variáveis redundantes de representação direta (%I, %M e %Q).

Para que este serviço seja executado, diversas condições devem ser satisfeitas:

- O serviço de sincronização anterior deste ciclo da MainTask (Troca de Diagnósticos e Comando) deve ter completado com sucesso.
- Caso este CP esteja em estado Ativo, o outro deve estar em estado Não-Ativo. Por outro lado, caso este CP esteja em estado Não-Ativo, o outro deve estar em estado Ativo.
- Os projetos dos dois CPs devem estar idênticos, exceto quando a sincronização automática de projetos estiver desabilitada (ver seção Desabilitação da Sincronização de Projetos).
- Pelo menos um canal de sincronismo (NETA ou NETB) deve estar em estado operacional.

6.3.6.3. Sincronização da Lista de Forçamentos Redundantes

Este serviço é responsável pela transferência da lista de forçamentos redundantes do CP Ativo para o CP Não-Ativo.

Para que este serviço seja executado, diversas condições devem ser satisfeitas:

- Os dois serviços de sincronização anteriores deste ciclo (Troca de Diagnósticos e Comando e Sincronização de Dados Redundantes) devem ter completado com sucesso.
- Caso este CP esteja em estado Ativo, o outro deve estar em estado Não-Ativo. Por outro lado, caso este CP esteja em estado Não-Ativo, o outro deve estar em estado Ativo.
- Os projetos dos dois CPs devem estar idênticos, exceto quando a sincronização automática de projetos estiver desabilitada (ver seção Desabilitação da Sincronização de Projetos).
- Pelo menos um canal de sincronismo (NETA ou NETB) deve estar em estado operacional.

ATENÇÃO

A lista de forçamentos redundantes contém apenas forçamentos sobre variáveis redundantes. Além disso, em cada um dos CPs (CPA e CPB), pode existir uma lista diferente de forçamentos sobre variáveis não-redundantes.

6.3.7. Serviços de Sincronização Esporádicos através de NETA e NETB

Os seguintes serviços de sincronização são executados de forma esporádica. Ou seja, não são executados a cada ciclo da tarefa MainTask e outra tarefa do sistema executa estes serviços esporádicos em segundo plano, com menor prioridade.

6.3.7.1. Sincronização de Projetos

Este serviço é responsável por manter sincronizados os projetos dos CPs Ativo e Não-Ativo. Isto ocorre apenas quando os projetos estão diferentes e a sincronização automática de projetos está habilitada em ambos os CPs.

A sincronização ocorre sempre no sentido do CP Ativo para o CP Não-Ativo, e basta que um dos canais de sincronismo (NETA ou NETB) esteja operacional para que este serviço seja executado.

Quando a sincronização estiver habilitada, serão sincronizados os seguintes arquivos e serviços:

- Projeto de aplicação (código executável)
- Project archive (código fonte)
- Usuários e grupos
- Direitos de acesso
- Trace

O serviço de sincronização irá iniciar em até trinta segundos, após um dos CPs entrar no modo Ativo. Após seu início, o serviço irá comparar o CRC dos projetos do CP Ativo e do CP Não-Ativo a cada cinco segundos.

Quando uma sincronização for iniciada, o CP Não-Ativo irá para o modo Stop e assumirá o estado Não-Configurado. Após a transferência de todos os arquivos necessários, o CP Não-Ativo irá para Run e assumirá o estado Inicializando. Caso a transferência falhe, o CP voltará para o estado Não-Configurado. Caso a transferência funcione, o CP normalmente irá para o estado Reserva (caso não haja falhas que impeçam isso).

O tempo que a sincronização de projetos levará para ser efetuada dependerá do tamanho do projeto e da ocupação da UCP do NX3035.

Caso a sincronização seja interrompida (perda da comunicação entre os canais de sincronismo) durante a transferência dos arquivos do CP Ativo para o Não-Ativo, o procedimento será abortado e reinicializado quando a comunicação for restaurada. Somente após a conclusão de todo o procedimento o CP Não-Ativo irá para o modo Run.

Além de manter sincronizados os projetos, a Sincronização do Projeto também irá impedir que o CP Não-Ativo assuma estados superiores a Inicializando caso o CRC esteja diferente ou alguma Alteração Online esteja pendente no CP Ativo.

ATENÇÃO

Uma sincronização de projeto terá os mesmos efeitos de um download no CP Não-Ativo. Este serviço não é executado se a sincronização automática de projetos estiver desabilitada, conforme descreve-se na seção Desabilitação da Sincronização de Projetos.
Nenhum serviço de sincronização entre o CPA e o CPB irá funcionar caso sejam invertidos os cabos dos canais de sincronismo. Por exemplo, conectar o canal NETA da UCP A no NETB da UCP B e o canal NETB da UCP A no NETA da UCP B.

6.3.8. Desabilitação da Sincronização de Projetos

Na seção Serviços de Sincronização Esporádicos através de NETA e NETB foram descritos serviços para a sincronização do projeto de aplicação e do project archive. Estes serviços normalmente devem estar habilitados e são úteis quando as modificações de projeto podem ser carregadas online no CP Ativo, sendo, a seguir, automaticamente retransmitidas para o CP Reserva via canais de sincronismo NETA / NETB.

Entretanto, existem modificações de projeto que não podem ser carregadas online em nenhum CP, como por exemplo a inclusão de módulos em uma remota PROFIBUS, ou a inclusão de uma nova remota PROFIBUS. Nestes casos, valendo-se da redundância do CP e da rede PROFIBUS, pode-se fazer tais modificações sem interromper o controle de processo, utilizando procedimentos descritos no capítulo Carga Offline de Programas sem Interrupção do Controle do Processo. Nestes procedimentos, é necessário desabilitar temporariamente a sincronização de projetos, permitindo que, por algum tempo, um dos CPs opere com a nova versão do projeto, enquanto o outro CP ainda opera com a versão antiga do projeto.

Uma UCP NX3035 possui um registro de *Desabilitação da Sincronização de Projetos*, não volátil, que permite desabilitar os serviços para sincronização do projeto de aplicação e do project archive. Este registro pode ser ajustado com o programador Mastertool. Basta desabilitar a sincronização de projetos em um dos dois CPs para que ela não mais ocorra.

Para desabilitar a sincronização de projetos, deve-se, primeiramente, conectar-se no CP desejado com o software Mastertool (ver seção Conexão do Mastertool com uma UCP NX3035 de um CP Redundante).

A seguir, no menu *Comunicação / Configuração de Redundância*, abre-se a combo-box *Sincronização do Projeto*, que permite selecionar uma das duas seguintes opções:

- Habilitar
- Desabilitar

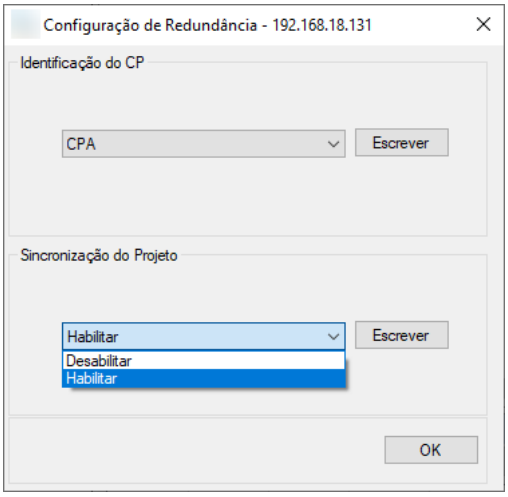


Figura 198: Desabilitar ou Habilitar Sincronização do Projeto

Deve-se selecionar a opção *Desabilitar* e depois pressionar o botão *Escrever*, ao lado desta combo-box. Uma mensagem informará se a operação teve sucesso ou não.

ATENÇÃO

A configuração da desabilitação de sincronismo de projetos não faz parte do projeto redundante desenvolvido com o Mastertool. Tal configuração reside apenas em um registro não-volátil da UCP NX3035, que pode ser lido ou gravado usando o Mastertool. O Mastertool não salva esta configuração em nenhum arquivo.

Esta configuração é copiada a cada ciclo da MainTask, do registro não volátil para o diagnóstico *DG_NX3035.tRedundancy.RedDgnLoc* do CP local. Este diagnóstico valerá TRUE quando a sincronização de projeto estiver desabilitada no CP local. O usuário pode verificar este diagnóstico no CP local para conferir se o comando teve sucesso, desde que este CP esteja em modo RUN. Caso o CP não esteja em modo RUN, é possível verificar esta configuração diretamente no visor da UCP NX3035 (ver seção Diagnósticos da Redundância no Visor Gráfico da UCP NX3035).

Também é possível observar a habilitação de sincronização de projeto do outro half-cluster (CP remoto), através do diagnóstico *DG_NX3035.tRedundancy.RedDgnRem.sGeneral_Diag.bProjectSyncDisable* (desde que o CP remoto esteja em modo RUN e algum dos canais de sincronismo NETA ou NETB esteja funcionando corretamente).

Para que a sincronização de projetos não ocorra, basta que esteja desabilitada num dos 2 CPs redundantes. Para que a sincronização de projetos ocorra, deve estar habilitada nos 2 CPs redundantes.

6.3.9. Configurações de Redes PROFIBUS

É possível instalar até seis módulos mestre PROFIBUS NX5001 em cada half-cluster. Pode-se configurar até três redes PROFIBUS redundantes, ou até seis redes PROFIBUS simples. Outras combinações com seis NX5001 também são possíveis, como por exemplo, duas redes redundantes e duas redes simples.

A Figura 191 mostra um exemplo com uma única rede PROFIBUS (PROFIBUS 1), que é redundante (PROFIBUS 1A e PROFIBUS 1B).

Cada uma das redes PROFIBUS pode ser configurada em dois modos diferentes:

- **Falha vital:** caso esta rede falhe completamente, esta falha poderá determinar uma transição de estado de redundância no CP redundante (switch-over). No caso de uma rede PROFIBUS redundante, uma falha completa consiste na falha das duas redes que a compõem.
- **Falha não vital:** mesmo que esta rede falhe completamente, esta falha não determinará uma transição de estado de redundância no CP redundante (switch-over).

Somente alguns tipos de remotas podem ser conectadas diretamente a uma rede PROFIBUS redundante:

- PO5065: escravo PROFIBUS DP-V1 com HART para remotas da Série Ponto.
- NX5210: escravo PROFIBUS DP-V1 com HART para remotas da Série Nexto.

Estes assuntos já foram abordados, ainda com mais detalhes, na seção Configuração de Redes PROFIBUS com Módulos NX5001.

6.3.10. Redes Ethernet Redundantes com NIC Teaming

A Figura 191 mostra diversos exemplos de rede Ethernet redundantes com NIC Teaming:

- A conexão da UCP NX3035 via NET1 e NET2 a uma rede de supervisão redundante (Ethernet de Supervisão A e B).
- A conexão da UCP NX3035 via NET3 e NET4 a uma rede HSDN redundante (HSDN A e B).
- A conexão da UCP NX3035 via NET5 e NET6 a uma rede Ethernet redundante de uso genérico (Ethernet redundante 1A e 1B).
- A conexão de um par de interfaces NX5000 a uma rede Ethernet redundante de uso genérico (Ethernet redundante 2A e 2B).

Maiores detalhes sobre configuração de portas NIC Teaming são fornecidos na seção Configuração das Interfaces Ethernet Integradas e Configuração de Interfaces Ethernet com Módulos NX5000.

6.3.11. Configuração de Endereços IP nas portas integradas NET1 ... NET6 de uma UCP NX3035 em um CP Redundante

Em um CP redundante, existem algumas diferenças nas configurações de endereços IP das portas integradas (NET1 ... NET6) da UCP NX3035, comparado a um CP não redundante.

Num cluster redundante da Série Nexto, é necessário configurar três endereços IP para cada uma das portas Ethernet da UCP NX3035 (NET1 a NET6), conforme mostra a figura seguinte:

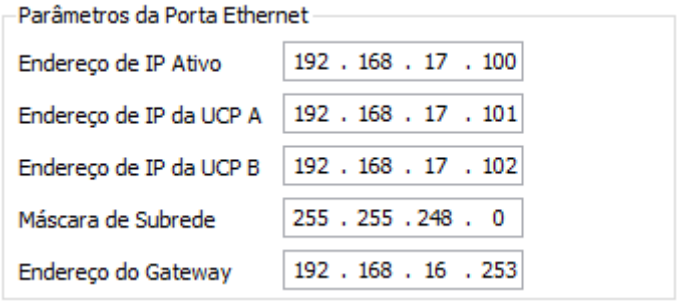


Figura 199: Endereços IP das interfaces Ethernet NET1 a NET6 da UCP NX3035

- **Endereço IP Ativo:** este endereço se aplica ao half-cluster que, no momento, está em estado ativo. Pode ser utilizado, por exemplo, por um sistema SCADA para ler e escrever dados do processo controlado. Graças ao endereço IP Ativo, o sistema SCADA não precisa se preocupar em trocar o endereço IP em caso de switchover entre os half-clusters CPA e CPB.
- **Endereço IP da UCP A:** este endereço IP sempre se aplica ao half-cluster CPA. Deve ser utilizado, por exemplo, por um sistema SCADA para ler diagnósticos ou escrever comandos privados no CPA.
- **Endereço IP da UCP B:** este endereço IP sempre se aplica ao half-cluster CPB. Deve ser utilizado, por exemplo, por um sistema SCADA para ler diagnósticos ou escrever comandos privados no CPB.

Enquanto o CPA estiver em estado ativo:

- A porta Ethernet da UCP A terá dois endereços IP simultâneos: *Endereço IP da UCP A* e *Endereço IP Ativo*.
- A porta Ethernet da UCP B terá um único endereço IP: *Endereço IP da UCP B*.

Enquanto o CPB estiver em estado ativo:

- A porta Ethernet da UCP A terá um único endereço IP: *Endereço IP da UCP A*.
- A porta Ethernet da UCP B terá dois endereços IP simultâneos: *Endereço IP da UCP B* e *Endereço IP Ativo*.

6.3.12. Configuração de Endereços IP em Módulos NX5000 em um CP Redundante

Em um CP redundante, existem algumas diferenças nas configurações de endereços IP dos NX5000, comparado a um CP não redundante.

Ao contrário das portas Ethernet da UCP NX3035 (NET1 ... NET6) que suportam dois endereços IP simultâneos (ver seção anterior), a porta Ethernet de um NX5000 suporta um único endereço IP. Por este motivo, foram desenvolvidos quatro métodos de troca de IP diferenciados para o NX5000. Estes métodos, descritos a seguir, são: IP Fixo, Troca Automática de IP, IP Ativo e Múltiplos IPs.

6.3.12.1. IP Fixo

Este método pode ser configurado nas interfaces Ethernet dos módulos Ethernet NX5000 conforme mostra a figura seguinte:

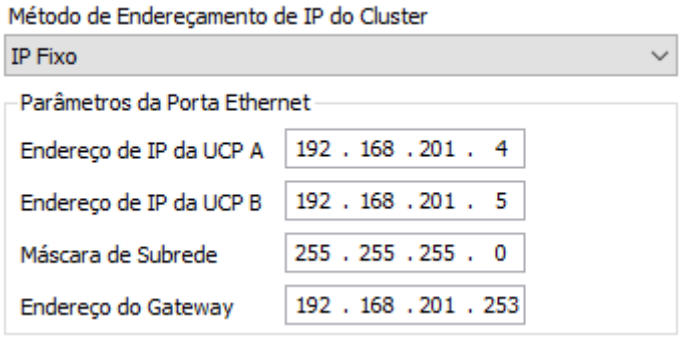


Figura 200: Método IP Fixo

Nesse método simples, o endereço IP de cada half-cluster não depende do seu estado de redundância (Ativo ou Não-Ativo):

- O NX5000 do CPA sempre terá o *Endereço IP da UCP A*.
- O NX5000 do CPB sempre terá o *Endereço IP da UCP B*.

6.3.12.2. Troca Automática de IP

Este método pode ser configurado nas interfaces Ethernet dos módulos Ethernet NX5000 conforme mostra a figura seguinte:

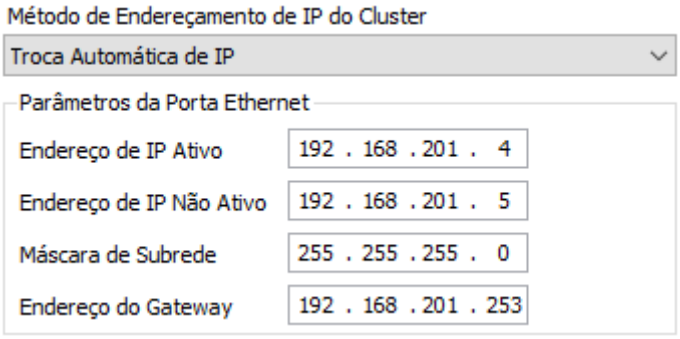


Figura 201: Método de Troca Automática de IP

Nesse método, o endereço IP de cada half-cluster depende do seu estado de redundância (Ativo ou Não-Ativo):

- O NX5000 do CP em estado Ativo terá o *Endereço IP Ativo*.
- O NX5000 do CP em estado Não-Ativo terá o *Endereço IP Não Ativo*.

Enquanto o CPA estiver em estado ativo:

- O NX5000 do CPA terá o *Endereço IP Ativo*.
- O NX5000 do CPB terá o *Endereço IP Não Ativo*.

Enquanto o CPB estiver em estado ativo:

- O NX5000 do CPA terá o *Endereço IP Não Ativo*.
- O NX5000 do CPB terá o *Endereço IP Ativo*.

ATENÇÃO

Neste método de endereçamento, as portas Ethernet de ambos os CPs (CPA e CPB) irão assumir o mesmo *Endereço IP Não Ativo* enquanto os dois CPs estiverem no estado Não-Ativo, provocando conflito de endereçamento na rede. Considerando que essa é uma situação incomum e temporária (ex: power-on simultâneo dos dois half-clusters), onde nenhum CP está controlando o sistema, isso não configura um problema grave.

6.3.12.3. IP Ativo

Este método pode ser configurado nas interfaces Ethernet dos módulos Ethernet NX5000 conforme mostra a figura seguinte:

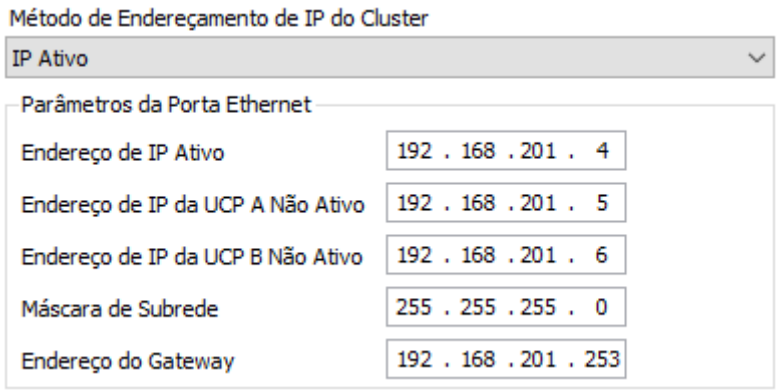


Figura 202: Método IP Ativo

Nesse método, o endereço IP de cada half-cluster depende do seu estado de redundância (Ativo ou Não-Ativo):

- O NX5000 do CP em estado Ativo terá o *Endereço IP Ativo*.
- O NX5000 do CPA, em estado Não-Ativo, terá o *Endereço IP da UCP A Não Ativo*.
- O NX5000 do CPB, em estado Não-Ativo, terá o *Endereço IP da UCP B Não Ativo*.

Enquanto o CPA estiver em estado ativo:

- O NX5000 do CPA terá o *Endereço IP Ativo*.
- O NX5000 do CPB terá o *Endereço IP da UCP B Não Ativo*.

Enquanto o CPB estiver em estado ativo:

- O NX5000 do CPA terá o *Endereço IP da UCP A Não Ativo*.
- O NX5000 do CPB terá o *Endereço IP Ativo*.

6.3.12.4. Múltiplos IPs

Este método pode ser configurado nas interfaces Ethernet dos módulos Ethernet NX5000 conforme mostra a figura seguinte:

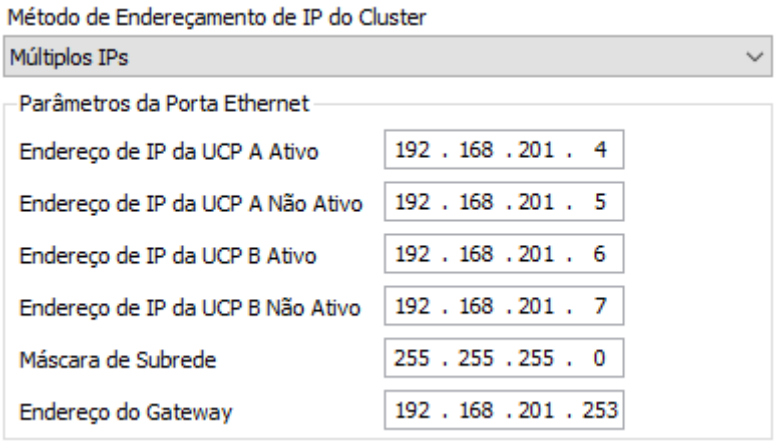


Figura 203: Método Múltiplos IPs

Nesse método, o endereço IP de cada half-cluster depende do seu estado de redundância (Ativo ou Não-Ativo):

- O NX5000 do CPA, em estado Ativo, terá o *Endereço IP da UCP A Ativo*.
- O NX5000 do CPA, em estado Não-Ativo, terá o *Endereço IP da UCP A Não Ativo*.
- O NX5000 do CPB, em estado Ativo, terá o *Endereço IP da UCP B Ativo*.
- O NX5000 do CPB, em estado Não-Ativo, terá o *Endereço IP da UCP B Não Ativo*.

Enquanto o CPA estiver em estado ativo:

- O NX5000 do CPA terá o *Endereço IP da UCP A Ativo*.
- O NX5000 do CPB terá o *Endereço IP da UCP B Não Ativo*.

Enquanto o CPB estiver em estado ativo:

- O NX5000 do CPA terá o *Endereço IP da UCP A Não Ativo*.
- O NX5000 do CPB terá o *Endereço IP da UCP B Ativo*.

6.3.13. Uso Combinado de NIC Teaming e IP Ativo

Caso determinado par de portas forme um par NIC Teaming em um CP redundante, estas portas podem implementar, ao mesmo tempo, as estratégias NIC Teaming e IP Ativo.

Por exemplo, se as portas NET 1 e NET 2 da UCP NX3035 formam um par NIC Teaming, então:

- **Endereço IP do CPA:** endereço IP das portas NET 1 + NET 2 da UCP NX3035 do CPA
- **Endereço IP do CPB**: endereço IP das portas NET 1 + NET 2 da UCP NX3035 do CPB
- **Endereço IP Ativo:** endereço IP das portas NET 1 + NET 2 daquela UCP NX3035 que se encontra no CP Ativo

Desta maneira, associa-se a excelente disponibilidade da estratégia NIC Teaming com a praticidade da estratégia de IP Ativo, que dispensa scripts em sistemas SCADA ou em outros clientes conectados a servidores no CP Ativo.

6.3.14. Uso de Comunicação OPC DA com Projetos Redundantes

O protocolo OPC DA pode ser configurado para comunicação de clusters redundantes com sistemas SCADA. Quando esta opção é selecionada na criação de um projeto redundante, o objeto *Symbol Configuration* é adicionado ao projeto. Neste objeto são configuradas as variáveis do sistema que serão enviadas para o sistema SCADA. Esta opção de comunicação é habilitada nas portas Ethernet da UCP NX3035.

Para mais informações a respeito da configuração da comunicação OPC DA com projetos redundantes consulte a seção Configuração com CP no Servidor OPC DA com Redundância de Conexão deste Manual.

6.3.15. Estados de um CP Redundante

Em um sistema redundante, um CP ou half-cluster (CPA ou CPB) pode assumir os seguintes estados:

- Ativo
- Reserva
- Inativo
- Não-Configurado
- Inicializando

> **ATENÇÃO**
>
> Frequentemente este manual utilizará a designação "Não-Ativo" para qualquer estado diferente de Ativo, ou seja, para designar qualquer um dos outros quatro estados (Reserva, Inativo, Não-Configurado e Inicializando).

A seguir, estes cinco estados são descritos brevemente. Maiores detalhes sobre os estados de um CP redundante, e sobre as transições entre os mesmos, serão abordados na seção Máquina de Estados de Redundância.

6.3.15.1. Ativo

Neste estado, o CP controla o processo automatizado, usando o programa *ActivePrg*, que é executado somente neste estado. O CP Ativo também atualiza o sistema de E/S remotas PROFIBUS, colocando seus mestres PROFIBUS em modo ativo, para estabelecer comunicação com as remotas (escravos) PROFIBUS.

6.3.15.2. Reserva

Neste estado, o CP está pronto para chavear para o estado Ativo, caso haja uma demanda para isso, tal como uma falha no CP Ativo. Além disso, verifica sua própria integridade, podendo chavear para o estado Inativo ou Não-Configurado em caso de alguma falha.

6.3.15.3. Inativo

Este estado é alcançado depois da detecção de uma falha que requer manutenção. Também pode ser alcançado devido a alguma solicitação manual do usuário para executar uma manutenção programada.

6.3.15.4. Não-Configurado

Este é o estado de redundância inicial. Por convenção, também assume-se que o CP está neste estado enquanto estiver desligado.

6.3.15.5. Inicializando

Diferente de todos os outros quatro estados que podem ter duração indeterminada, o estado Inicializando dura poucos segundos. Este estado é sempre alcançado a partir do estado Não-Configurado.

Ao entrar no estado Inicializando, diversas ações, testes e verificações são executados, para decidir qual será o próximo estado.

6.3.16. Máquina de Estados de Redundância

A figura seguinte mostra a máquina de estados da redundância, ilustrando todos os estados de redundância e todas as possíveis transições entre os mesmos.

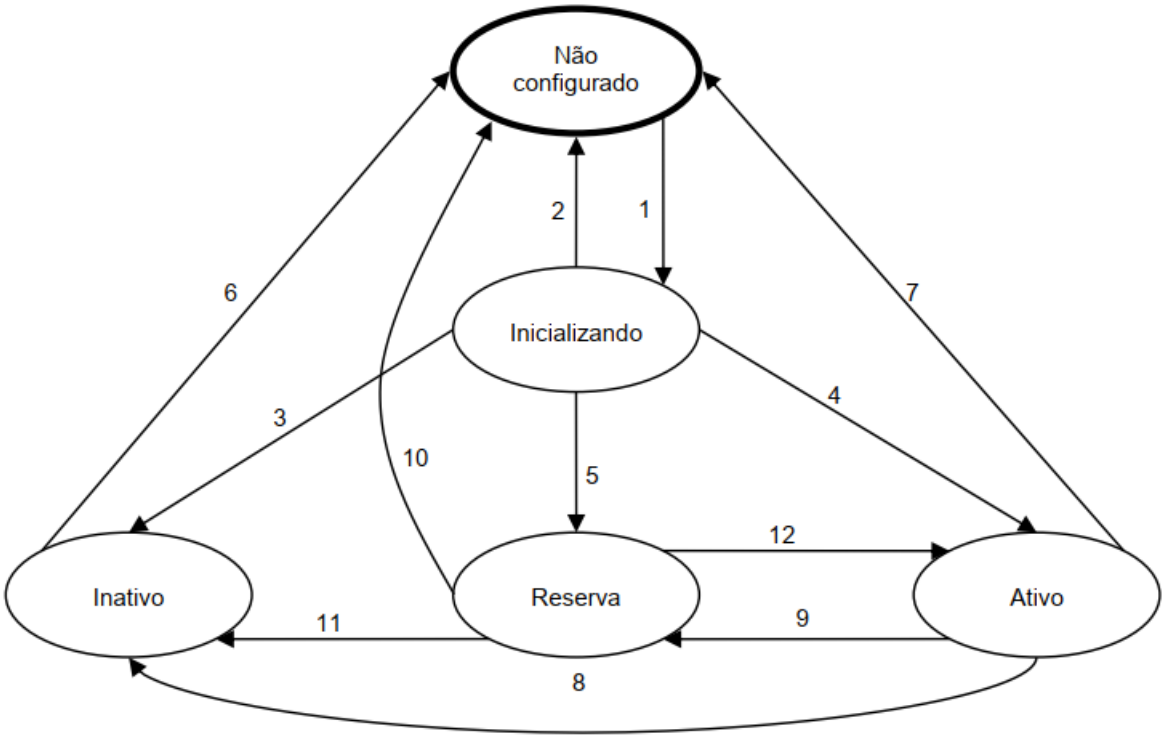


Figura 204: Máquina de Estados da Redundância

As subseções seguintes descrevem o comportamento de cada estado, e as causas das transições deste estado para outros estados.

Para interpretar corretamente o funcionamento desta máquina de estados, é preciso estabelecer algumas regras e sequências:

- Transições que se originam do mesmo estado devem ser avaliadas na sequência dada pelo número da transição. Por exemplo, as transições 2, 3, 4 e 5 se originam do estado Inicializando. Neste exemplo, avalia-se primeiro a transição 2, depois a 3, depois a 4 e finalmente a 5. Caso a transição 2 seja disparada, as transições 3, 4 e 5 não serão avaliadas.
- Transições só podem ser disparadas se o CP estiver energizado e a tarefa MainTask estiver executando. Do contrário, assume-se que o CP esteja no estado Não-Configurado.

6.3.16.1. Estado Não Configurado

Este é o estado inicial da máquina de estados.

O CP assume este estado sempre que tarefa MainTask não estiver em execução, devido a motivos tais como:

- O CP está desligado (desenergizado).
- O CP foi ligado recentemente, mas ainda não iniciou a execução da tarefa MainTask.
- Um comando STOP foi atribuído pelo usuário via programador Mastertool.
- Um comando de reinicialização foi atribuído pelo usuário via programador Mastertool (Reset a Quente, Reset a Frio ou Reset Origem).
- Um watchdog ou outra condição de exceção provocou a parada da MainTask.

Quando a tarefa MainTask está em execução, o estado Não-Configurado pode ser atingido a partir dos outros 4 estados, conforme mostra a máquina de estados da Figura 204.

Durante o estado Não-Configurado:

- Os mestres PROFIBUS NX5001 estão desabilitados.
- Os serviços de sincronização cíclicos são executados (ver seção Serviços de Sincronização Cíclicos através de NETA e NETB), desde que as condições para sua execução estejam presentes.
- Os serviços de sincronização esporádicos também podem ser executados (ver seção Serviços de Sincronização Esporádicos através de NETA e NETB) desde que haja necessidade de executá-los e que as condições para sua execução estejam presentes.

6.3.16.1.1. Transição 1 - Não-Configurado para Inicializando

- Esta transição ocorre assim que todas as seguintes condições forem falsas:
  - A tarefa MainTask não está em execução (CP desligado ou em STOP).
  - O outro CP está em estado Ativo e uma sincronização de projeto é necessária ou está ocorrendo. Uma sincronização de projetos é necessária quando há diferenças entre o projeto deste CP e o projeto do CP ativo e, além disso, a sincronização de projetos está habilitada (ver seção Desabilitação da Sincronização de Projetos).

6.3.16.2. Estado Inicializando

Diferente de todos os outros quatro estados que podem ter duração indeterminada, o estado Inicializando dura poucos segundos. Este estado é sempre alcançado a partir do estado Não-Configurado através da Transição 1.

Ao entrar no estado Inicializando, diversas ações, testes e verificações são executados, para decidir qual será o próximo estado.

Durante o estado Inicializando, o CP:

- Verifica se a identificação do CP está correta (deve ser CPA ou CPB).
- Verifica se há problemas nos parâmetros de configuração extraídos do projeto Mastertool.
- Verifica a integridade da rede de sincronismo (NETA e NETB).
- Os serviços de sincronização cíclicos são executados (ver seção Serviços de Sincronização Cíclicos através de NETA e NETB), desde que as condições para sua execução estejam presentes.
- Verifica a compatibilidade de versões de firmware entre os dois CPs. Considerando que uma versão tem o formato X.Y.Z.W, duas versões serão compatíveis se o dígito X de ambas for igual.
- Verifica a igualdade de projetos entre os dois CPs, se a sincronização automática de projetos estiver habilitada (ver seção Desabilitação da Sincronização de Projetos).
- Habilita seus mestres PROFIBUS (NX5001) em estado passivo e tenta conexões com os respectivos mestres NX5001 do outro CP. Cada conexão terá sucesso caso o outro CP esteja em estado Ativo e não haja problemas nos circuitos de transmissão e recepção PROFIBUS do NX5001 ou no meio físico.

6.3.16.2.1. Transição 2 - Inicializando para Não-Configurado

- Este CP foi desligado ou reinicializado (Reset a Quente, Reset a Frio ou Reset Origem) ou sua UCP entrou em Modo STOP.
- O registro de identificação deste CP é inválido (diferente de CPA ou CPB).
- Existem erros lógicos de configuração no projeto recebido do Mastertool.
- O outro CP está no estado Ativo, e a versão de firmware deste CP é incompatível com a versão de firmware do CP Ativo. Considerando que uma versão tem o formato X.Y.Z.W, duas versões serão compatíveis se o dígito X de ambas for igual.
- O outro CP está em estado Ativo, e o projeto deste CP é diferente do projeto do CP Ativo, e a sincronização automática de projetos está habilitada (ver seção Desabilitação da Sincronização de Projetos).

6.3.16.2.2. Transição 3 - Inicializando para Inativo

- Os dois canais da rede de sincronismo (NETA e NETB) estão em falha, e este CP sabe que esta falha foi causada por componentes de hardware ou software internos (falhas internas de NETA ou NETB).
- O outro CP está em estado Ativo, mas não é possível sincronizar os dados redundantes ou a lista de forçamentos redundantes.
- O estado do outro CP não pode ser descoberto via rede de sincronismo (NETA / NETB), mas algum NX5001 deste CP, em modo passivo, consegue detectar tráfego em alguma das redes PROFIBUS configurada em modo de falha vital. Desta maneira, este CP assume que o outro CP está em estado Ativo e controlando o processo, embora NETA / NETB não estejam funcionando para ter certeza disso.
- O outro CP está em estado Ativo. Além disso, para determinada rede PROFIBUS configurada em modo de falha vital, todos os NX5001 deste CP, em modo passivo, não conseguem detectar tráfego nestas redes PROFIBUS. Isto indica falha em todos os NX5001 deste CP nesta rede PROFIBUS configurada em modo de falha vital.

6.3.16.2.3. Transição 4 - Inicializando para Ativo

- O outro CP encontra-se em estado Não-Ativo. Antes de fazer esta transição, esta condição deve permanecer verdadeira durante alguns segundos. Quando o CPA e CPB são energizados simultaneamente, o CP que terminar a inicialização do sistema antes assumirá o estado Ativo.
- O estado do outro CP não pode ser descoberto via rede de sincronismo (NETA / NETB), e além disso os NX5001 deste CP, em modo passivo, não conseguem detectar tráfego em nenhuma das redes PROFIBUS configuradas em modo de falha vital, ou então não existe nenhuma rede PROFIBUS configurada em modo de falha vital. Portanto, tudo indica que o outro CP está desligado ou fora de execução. Esta condição deve se manter verdadeira durante alguns segundos antes de fazer esta transição.

6.3.16.2.4. Transição 5 - Inicializando para Reserva

- O outro CP encontra-se em estado Ativo e os serviços de sincronização de dados redundantes e de sincronização da lista de forçamentos redundantes estão funcionando corretamente.

6.3.16.3. Estado Inativo

Este estado é alcançado depois da detecção de uma falha que requer manutenção.

Também pode ser alcançado devido a alguma solicitação manual do usuário para executar uma manutenção programada. O comando manual antes de iniciar uma manutenção programada pode ser, por exemplo:

- RedCmdLoc.bInactiveLocal, executado neste CP
- RedCmdLoc.bInactiveRemote, executado no outro CP
- Comando via botão OTD, executado neste CP

Para sair do estado Inativo, depois de corrigir as falhas que levaram a este estado, ou depois de executar a manutenção programada, o usuário deve executar um comando manual, que pode ser, por exemplo:

- RedCmdLoc.bStandbyLocal, executado neste CP
- RedCmdLoc.bStandbyRemote, executado no outro CP
- Comando via botão OTD, executado neste CP

Ao invés de executar o comando manual citado anteriormente, também é possível desligar e religar o CP, ou usar um comando de reinicialização (Reset a Quente, Reset a Frio ou Reset Origem), ou colocar o CP em STOP e depois para RUN.

6.3.16.3.1. Transição 6 - Inativo para Não-Configurado

- Este CP foi desligado ou reinicializado (Reset a Quente, Reset a Frio ou Reset Origem) ou sua UCP entrou em Modo Stop.
- Foi executado um comando manual para sair do estado Inativo, por exemplo:
  - RedCmdLoc.bStandbyLocal, executado neste CP
  - RedCmdLoc.bStandbyRemote, executado no outro CP
  - Comando via botão OTD, executado neste CP, para ir para o estado Reserva

6.3.16.4. Estado Ativo

Neste estado, o CP controla o processo automatizado, usando o programa ActivePrg, que é executado somente neste estado. O CP Ativo também atualiza o sistema de E/S remotas PROFIBUS, colocando seus mestres PROFIBUS NX5001 em modo ativo, que serve para estabelecer comunicação com as remotas (escravos) PROFIBUS.

O CP Ativo também verifica seus diagnósticos internos e requisições de switch-over do usuário para determinar se um switch-over é necessário. O CP normalmente só sairá do estado Ativo se souber que o outro CP está em estado Reserva e, portanto, apto para assumir o estado Ativo.

No entanto, existem algumas situações em que o CP Ativo poderá sair do estado Ativo mesmo sem ter certeza de que o outro CP está em estado Reserva, por exemplo:

- Se este CP for desligado
- Se este CP for colocado em modo STOP
- Se um comando de reinicialização ocorrer neste CP (Reset a Quente, Reset a Frio ou Reset Origem)

6.3.16.4.1. Transição 7 - Ativo para Não-Configurado

- Este CP foi desligado ou reinicializado (Reset a Quente, Reset a Frio ou Reset Origem) ou sua UCP entrou em Modo Stop.

6.3.16.4.2. Transição 8 - Ativo para Inativo

- Este CP perdeu comunicação com o outro CP via rede de sincronismo (NETA e NETB) devido a uma falha diagnosticada como interna deste CP, e além disso sabe que o outro CP estava em estado Reserva pouco antes desta falha ocorrer. Esta condição não é avaliada nos primeiros 2 segundos em estado Ativo.
- Este CP, com seus NX5001 em modo ativo, não consegue se comunicar com nenhum dos escravos PROFIBUS de uma rede PROFIBUS configurada em modo vital. Além disso, este CP sabe que o outro CP está em estado Reserva. Esta condição não é avaliada nos primeiros 2 segundos em estado Ativo.

6.3.16.4.3. *Transição 9 - Ativo para Reserva*

- Os dois CPs, por algum motivo, estão no estado Ativo, e este conflito deve ser resolvido. O CPA chaveará para o estado Reserva caso este conflito dure certo tempo, e o CPB fará o mesmo após um tempo diferente, que é menor do que o tempo do CPA. Desta maneira, em caso de conflito, o CPA tem prioridade para continuar em estado Ativo. Caso o estado do outro CP seja desconhecido devido a uma falha na rede de sincronismo (NETA / NETB), o CP Ativo não irá para o estado Reserva.
- O outro CP está em estado Reserva, e este CP recebeu um comando manual para ir para o estado Reserva. Esta condição não é avaliada nos primeiros 2 segundos em estado Ativo. O comando manual para ir para o estado Reserva pode ser, por exemplo:
  - RedCmdLoc.bStandbyLocal, executado neste CP
  - RedCmdLoc.bStandbyRemote, executado no outro CP
  - Comando via botão OTD, executado neste CP

6.3.16.5. Estado Reserva

Neste estado, o CP está pronto para chavear para o estado Ativo, caso haja uma demanda para isso, tal como uma falha no CP Ativo.

O CP Reserva também verifica seus próprios diagnósticos, e poderá chavear para o estado Não-Configurado ou Inativo em função de algumas falhas.

Mestres PROFIBUS NX5001 são habilitados em estado passivo. O modo passivo serve para testar os circuitos de transmissão e recepção PROFIBUS e o meio físico, para evitar a ocorrência de uma falha oculta. Falhas totais em redes PROFIBUS configuradas como vitais causam um chaveamento para o estado Inativo. Uma falha total em uma rede PROFIBUS atinge as duas redes que a compõem (rede PROFIBUS redundante) ou a única rede que a compõe (rede PROFIBUS não-redundante).

6.3.16.5.1. *Transição 10 – Reserva para Não-Configurado*

- Este CP foi desligado ou reinicializado (Reset a Quente, Reset a Frio ou Reset Origem), ou sua UCP entrou em Modo Stop.
- O outro CP encontra-se em estado Ativo e o projeto deste CP é diferente do projeto do CP Ativo, e a sincronização automática de projetos está habilitada (ver seção Desabilitação da Sincronização de Projetos).
- O outro CP está no estado Ativo, e a versão de firmware deste CP é incompatível com a versão de firmware do CP Ativo

6.3.16.5.2. *Transição 11 – Reserva para Inativo*

- Um comando manual para mudar para o estado Inativo foi executado. Por exemplo:
  - RedCmdLoc.bInactiveLocal, executado neste CP
  - RedCmdLoc.bInactiveRemote, executado no outro CP
  - Comando via botão OTD, executado neste CP

  Isto é feito tipicamente para executar uma manutenção programada no CP Não-Ativo. Deve-se evitar fazer manutenções programadas no CP Reserva, por isso, antes é aconselhável chaveá-lo para o estado Inativo.
- O outro CP encontra-se em estado Ativo. No entanto, o serviço de sincronização de dados redundantes não funcionou corretamente nos últimos quatro ciclos da MainTask, ou o serviço de sincronização de diagnósticos redundantes não funcionou corretamente nos últimos dois ciclos da MainTask.
- O outro CP encontra-se em estado Ativo. No entanto, este CP não consegue monitorar tráfego PROFIBUS em todos os NX5001 de uma rede PROFIBUS configurada em modo de falha vital.
- O estado do outro CP é desconhecido devido a falhas na rede sincronismo (NETA / NETB). Entretanto, este CP consegue monitorar tráfego PROFIBUS em algum dos seus NX5001 em modo passivo, em qualquer uma das redes PROFIBUS configuradas em modo de falha vital. Isso indica que o outro CP ainda está em estado Ativo.

6.3.16.5.3. *Transição 12 – Reserva para Ativo*

- O estado do outro CP é conhecido e diferente de Ativo.
- O estado do outro CP é desconhecido devido a falhas na rede sincronismo (NETA / NETB). Além disso, este CP não consegue monitorar tráfego PROFIBUS em nenhum dos seus NX5001 em modo passivo, em qualquer uma das redes PROFIBUS configuradas em modo de falha vital. Isso indica que o outro CP não está em estado Ativo.

6.3.17. Primeiros Instantes em Estado Ativo

Nos primeiros 2 segundos em estado Ativo, diversas transições que normalmente poderiam tirar o CP do estado Ativo não são avaliadas (ver subseções anteriores que definem transições a partir do estado Ativo).

Apenas duas condições permitem que o CP saia do estado Ativo durantes estes dois segundos:

- Este CP foi desligado ou reinicializado (Reset a Quente, Reset a Frio ou Reset Origem) ou sua UCP entrou em Modo Stop, causando transição para Não-Configurado.
- Os dois CPs, por algum motivo, estão no estado Ativo, e este conflito deve ser resolvido. O CPA chaveará para o estado Reserva caso este conflito dure certo tempo, e o CPB fará o mesmo após um tempo diferente, que é menor do que o tempo do CPA. Desta maneira, em caso de conflito, o CPA tem prioridade para continuar em estado Ativo.

Além disso, nos primeiros instantes que um CP assume o estado Ativo, alguns diagnósticos que não são redundantes podem não ser válidos, por exemplo, os diagnósticos dos módulos NX5000 e NX5001. O método para não considerar estes diagnósticos possivelmente "inválidos" é descrito na seção Utilização de Diagnósticos Não-Redundantes.

6.3.18. Falhas mais Comuns Causadoras de Switch-overs Automáticos entre Half-Clusters

Nesta seção, são listadas as falhas mais comuns que, de forma automática, causam um switch-over do CP Ativo para Não-Ativo, e do CP Reserva para Ativo. Estas falhas disparam um sub-conjunto daquelas transições examinadas na seção Máquina de Estados de Redundância.

- Falta de alimentação no CP Ativo. É importante que os dois CPs tenham sistemas de alimentação redundantes, para que uma falha comum na alimentação não afete ambos.
- Falha na fonte de alimentação NX8000 do CP Ativo.
- Falha no barramento do bastidor (NX9000, NX9001, NX9002 ou NX9003) do CP Ativo.
- Falhas na UCP NX3035 do CP Ativo, tais como:
  - Cão-de-guarda.
  - Reinicialização (Reset a Quente, Reset a Frio ou Reset Origem).
  - Parada (Stop).
  - Falha internas que afetam ambos os canais de sincronismo (NETA e NETB).
- Falha total de uma rede PROFIBUS no CP Ativo, caso esta rede esteja configurada em modo vital. Caso a rede PROFIBUS seja redundante, ambas as redes que a compõem devem estar em falha (falha dupla).

6.3.19. Switch-overs entre Half-Clusters Gerenciados pelo Usuário

Entre as transições examinadas na seção Máquina de Estados de Redundância, algumas possibilitam que o usuário gerencie switch-overs entre half-clusters, devido a falhas que normalmente não geram switch-overs de forma automática.

Existem casos muito particulares, que dependem da filosofia de cada cliente. Considere um exemplo em que o sistema SCADA perde comunicação com o CP Ativo, mas ainda consegue se comunicar com o CP Reserva. O operador pode enviar um comando do SCADA para o CP Reserva, e o CP Reserva pode então utilizar o comando RedCmdLoc.bStandbyRemote para solicitar que o CP Ativo vá para o estado Reserva, provocando um switch-over entre os CPs Ativo e Reserva.

Outro exemplo é ilustrado na seção Exemplo de Gerenciamento de um Cliente MODBUS TCP num CP Redundante. Este exemplo descreve um switch-over gerenciado automaticamente sem nenhuma intervenção do operador. Neste caso, trata-se de um cliente MODBUS TCP, cujas comunicações com o servidor devem ser habilitadas somente no CP Ativo. Caso o CP Ativo perca todas comunicações com o servidor, ou tenha problemas de perda de link (link-down), um switch-over automático deverá ocorrer entre os CPs Ativo e Reserva. Este exemplo também mostra como evitar que haja falha oculta no CP reserva, onde as comunicações com o servidor devem estar desabilitadas.

Através de estruturas de dados como aquelas citadas na seção Estruturas de Dados de Diagnósticos, Comandos e Informações do Usuário, é possível trocar diagnósticos e comandos entre os half-clusters via NETA e NETB e, desta maneira, o usuário pode executar gerenciamentos especiais de redundância, para falhas que normalmente não causariam switch-overs. Maiores detalhes sobre estas estruturas de dados serão fornecidos na seção Estrutura de Diagnósticos, Comandos e Informações do Usuário para a Redundância.

6.3.20. Tolerância a Falhas

O objetivo principal de um CP redundante é o aumento da disponibilidade do sistema. A disponibilidade é a razão entre o tempo em que o sistema está funcionando corretamente e o tempo total desde a implantação do sistema. Por exemplo, se um sistema foi implantado há 10 anos, e durante este período esteve parado no tempo total de um ano devido a falhas, então sua disponibilidade foi de apenas 90%. Disponibilidades desta ordem são geralmente inaceitáveis para sistemas críticos, sendo que valores da ordem de 99,5% ou ainda superiores podem ser solicitados nestes sistemas.

Para atingir disponibilidades desta ordem, são necessárias diversas estratégias:

- Utilização de componentes mais confiáveis (com alto MTBF, ou tempo médio entre falhas), o que contribuirá para o aumento do MTBF do sistema como um todo.
- Utilização de redundância pelo menos para os componentes mais críticos ou componentes com menor MTBF, de tal maneira que a falha de um componente possa ser tolerada sem parar o sistema. Se a redundância for implementada através da duplicação de componentes, será necessário que os dois falhem para que o sistema como um todo fique indisponível.
- Alta cobertura de diagnósticos, em especial de componentes redundantes. A redundância de componentes é pouco útil para o aumento da disponibilidade quando não se pode descobrir que um componente redundante falhou. Neste caso, a primeira falha em um dos componentes ainda não compromete o sistema, mas por permanecer oculta, algum dia acontecerá a segunda falha que comprometerá o sistema, já que a primeira falha ainda não foi reparada. As falhas podem ser classificadas entre diagnosticáveis e ocultas, sendo altamente desejável que a maior parte das falhas de componentes redundantes sejam diagnosticáveis.
- Também é importante que componentes não-redundantes tenham ampla cobertura de diagnósticos, pois muitas vezes o sistema pode continuar funcionando mesmo com a falha de um componente não-redundante. O componente pode não estar sendo solicitado. Por exemplo, um relé com contato aberto, que raramente tem sua bobina acionada, não tem sua falha detectada até o momento em que o sistema solicitar seu fechamento.
- Baixo tempo de reparo de componentes não-redundantes. A falha de um componente não-redundante pode comprometer o sistema, e durante o reparo, o sistema estará indisponível.
- Possibilidade de reparar ou substituir um componente redundante sem parar o sistema. Se esta possibilidade existe, obtém-se um grande aumento de disponibilidade. Do contrário, deve-se programar uma parada do sistema para substituir o componente, e o tempo de reparo conta como tempo indisponível.
- Baixo tempo de reparo de componentes redundantes. A falha de um componente redundante não compromete o sistema, mas durante seu reparo, eventualmente pode ocorrer uma falha em seu par redundante. Por este motivo, é importante que a falha seja reparada brevemente, após diagnosticada. Quanto maior o tempo de reparo, maior a probabilidade de acontecer uma segunda falha no componente redundante durante o reparo da primeira falha, o que comprometeria o sistema. Portanto, quanto maior o tempo de reparo, menor a disponibilidade do sistema.
- Programar testes offline periódicos em componentes para detectar falhas não diagnosticáveis automaticamente pelo sistema. O objetivo é detectar falhas ocultas, especialmente em componentes redundantes ou componentes simples que não estejam sendo solicitados (por exemplo, um relé de segurança). Testes offline às vezes implicam em paradas no sistema, o que diminui a disponibilidade. Normalmente aproveita-se ocasiões especiais, tais como paradas programadas de manutenção da planta. Quanto maior o período entre testes offline, maior o tempo em que uma falha pode permanecer oculta e, portanto, maior a probabilidade de esta falha comprometer o sistema, ou seja, menor a disponibilidade do sistema.

Estes princípios foram considerados no projeto de CPs redundantes com UCP NX3035.

As próximas subseções analisam diversos tipos de falhas e como são toleradas ou não toleradas, e se existem switch-overs associados às falhas toleradas.

6.3.20.1. Falhas Simples com Indisponibilidade

Alguns componentes, por não serem duplicados, não toleram nem sequer falha simples sem causar algum tipo de indisponibilidade. Em um CP redundante com UCP NX3035, trata-se dos seguintes componentes:

- Remotas (escravos) PROFIBUS em uma rede PROFIBUS não-redundante.
- Remotas (escravos) Ethernet em uma rede Ethernet não redundante.
- Módulos de E/S.

A tolerância a falhas de uma rede PROFIBUS pode ser aumentada optando-se por uma rede PROFIBUS redundante, que é aconselhada em sistemas que demandam alta tolerância a falhas. A Figura 191 mostra um exemplo de arquitetura com uma rede PROFIBUS redundante.

Da mesma forma a tolerância a falhas de uma rede Ethernet pode ser aumentada usando-se uma rede Ethernet redundante com a configuração de NIC Teaming.

Quanto à indisponibilidade de um módulo de E/S, deve-se observar que a mesma não se constitui em uma indisponibilidade total do sistema. Ela se constitui em uma indisponibilidade parcial, somente das malhas de controle que utilizam este módulo de E/S.

Embora não haja previsão de redundância de módulos de E/S, a aplicação do usuário pode gerenciá-la em casos especiais. Por exemplo, o usuário pode inserir três módulos de entradas analógicas em três remotas PROFIBUS diferentes, e implementar um esquema de votação entre triplas entradas analógicas, para algum sistema crítico. Tais soluções devem ser gerenciadas pelo usuário. Não existe nenhum suporte automático para as mesmas. Tais soluções, em geral, também implicam na redundância de sensores e atuadores no campo.

6.3.20.2. Falhas Simples sem Indisponibilidade Causando um Switch-over

Alguns componentes redundantes toleram falhas simples sem causar indisponibilidade, mas causam switch-over:

- Bastidores do half-cluster CPA ou CPB (NX9000, NX9001, NX9002 ou NX9003)
- Fontes de alimentação (NX8000) do CPA ou CPB
- UCPs (NX3035) do CPA ou CPB
- Módulos NX5001 (mestres PROFIBUS) em configuração de rede PROFIBUS não-redundante
- Módulos NX5000 (Ethernet) em configurações não redundante (sem NIC Teaming)
- Interface escravo PROFIBUS de uma remota redundante (PO5063V5, PO5065, NX5210 ou AL-3416). Neste caso, diferente dos anteriores, o switch-over acontece dentro da remota, entre as redes PROFIBUS A e B.

6.3.20.3. Falhas Duplas sem Indisponibilidade Causando um Switch-over

Alguns componentes são duplicados em cada half-cluster, desta maneira, antes de causar um switch-over, é necessário que ambos falhem:

- Módulos NX5001 (mestres PROFIBUS) em configuração redundante, configurados em modo de falha vital.
- Módulos NX5000 (Ethernet) em configurações com NIC Teaming (switchover gerenciado pelo usuário).

6.3.21. Overhead da Redundância

Uma aplicação redundante causa um aumento no tempo de processamento de uma aplicação, quando comparado ao tempo necessário para uma aplicação equivalente não-redundante.

Este tempo adicional deve-se principalmente à execução dos serviços de sincronização cíclicos, descritos na seção Serviços de Sincronização Cíclicos através de NETA e NETB, além de um tempo menor para o próprio gerenciamento da redundância (máquina de estados, etc...). O tempo adicional total, devido à redundância (overhead da redundância), é estimado pelo MasterTool e exibido na janela de Mensagens, após compilar o projeto do CP redundante.

Além disso, cabe ao usuário definir um intervalo para a MainTask suficiente para acomodar:

- O tempo necessário para executar as POUs principais (NonSkippedPrg e ActivePrg). Este tempo normalmente é medido após o desenvolvimento do projeto (descontando o overhead da redundância).
- Alguma folga para execução de outras tarefas da UCP (sistema operacional, drivers de E/S PROFIBUS, MODBUS, etc...). O percentual desta folga pode variar de acordo com o desempenho solicitado para estas tarefas. Por exemplo, se a comunicação MODBUS com o sistema SCADA precisa alocar muito processamento para atingir um desempenho satisfatório, esta folga deverá ser aumentada.

6.4. Programação de um CP Redundante

6.4.1. Assistente para Criação de um Novo Projeto Redundante

Para criar um novo projeto redundante, deve-se utilizar o comando *Arquivo/Novo Projeto*, e selecionar o *Projeto Mastertool Padrão*. O usuário também deve informar o nome que deseja dar ao projeto e a pasta onde deseja armazená-lo, conforme exemplo na figura abaixo.

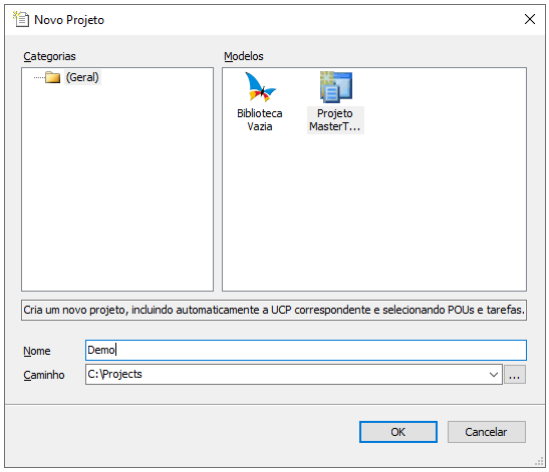


Figura 205: Novo Projeto

A seguir, o *Assistente* que gera o projeto de redundância realiza diversos questionamentos ao usuário quanto à configuração desejada, os quais devem ser respondidos ao longo de uma sequência de telas.

A primeira tela do *Assistente* é mostrada na figura seguinte, já preenchida com algumas opções compatíveis com a redundância do NX3035:

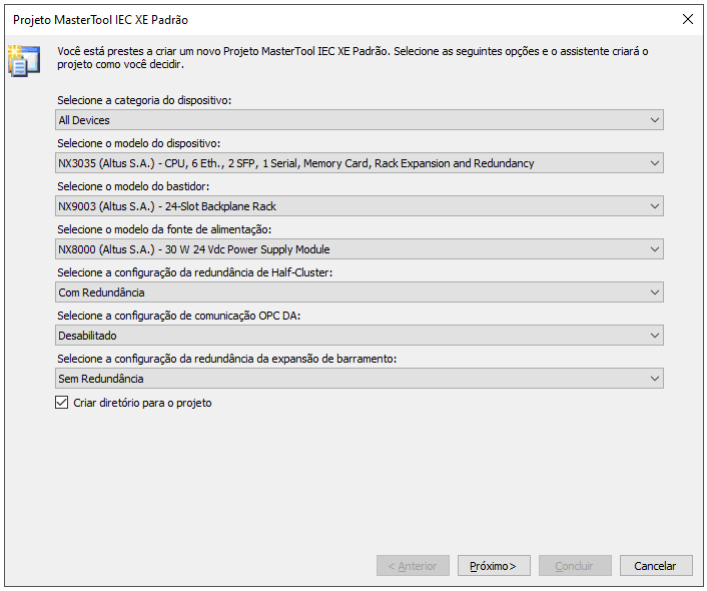


Figura 206: Primeira tela do Assistente

- **Categoria do dispositivo:** selecione sempre *Todos os Dispositivos*.
- *Modelo do dispositivo:* selecione o *NX3035*.
- **Modelo do bastidor:** selecione o bastidor adequado para a aplicação (*NX9000*, *NX9001*, *NX9002* ou *NX9003*), de acordo com o número de posições necessárias para acomodar todos os módulos no half-cluster.
- **Fonte de alimentação:** selecione o *NX8000*.
- **Configuração de redundância de Half-Cluster:** selecione *Com Redundância*.
- **Configuração de comunicação OPC DA:** selecione caso desejar utilizá-la. Mais detalhes na seção Uso de Comunicação OPC DA com Projetos Redundantes.
- **Configuração da redundância da expansão de barramento:** normalmente o usuário seleciona a opção *Sem Redundância* num CP NX3035 redundante. Entretanto, é possível usar um barramento expandido redundante para colocar módulos de I/O em outros bastidores, mas não é comum utilizar módulos de I/O nos bastidores locais de um half-cluster.
- Marque o checkbox *Criar diretório para o projeto*, caso deseje criar uma pasta automaticamente com o nome do projeto.

Clique no botão *Próximo>* para selecionar a segunda tela do Assistente. A segunda tela do Assistente é mostrada na figura seguinte, já preenchida com algumas opções compatíveis com a redundância do NX3035. Esta tela serve para configurar a redundância das portas Ethernet integradas (NET1 ... NET6) da UCP NX3035. No exemplo seguinte, consideramos que haverá dois pares NIC Teaming de portas (NET1+NET2 e NET3+NET4) e duas portas simples (NET5 e NET6).

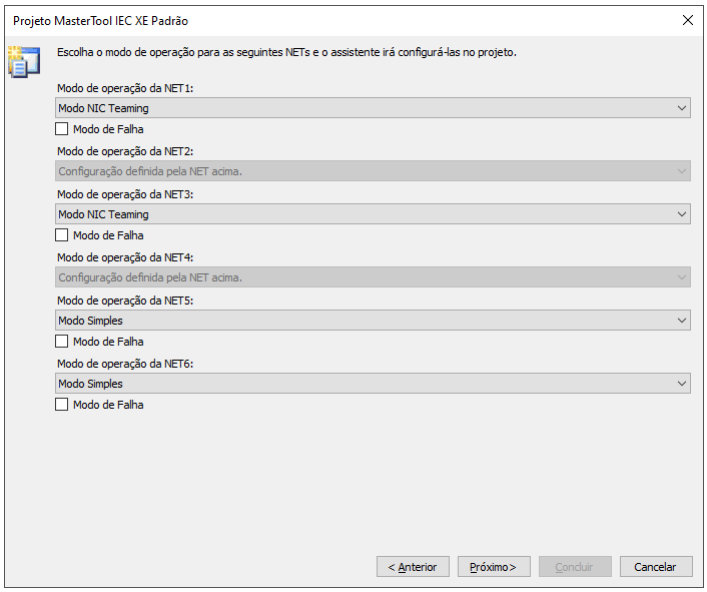


Figura 207: Segunda tela do Assistente

Clique no botão *Próximo>* para selecionar a terceira tela do Assistente. A terceira tela do Assistente é mostrada na figura seguinte, já preenchida com algumas opções compatíveis com a redundância do NX3035. Esta tela tem duas funções:

- Definir a quantidade de redes PROFIBUS (entre 0 e 6), e se cada uma delas é simples (aloca um único NX5001) ou redundante (aloca dois NX5001). No exemplo seguinte, as duas primeiras redes (PROFIBUS 1 e PROFIBUS 2) são redundantes, e a terceira rede (PROFIBUS 3) é simples. Todas as redes PROFIBUS criadas pelo Assistente tem o modo de falha vital habilitado, ou seja, haverá switchover em caso de falha numa destas redes. É possível desabilitar este modo de falha mais tarde em cada rede, se desejável. Observar que existe um limite de no máximo 6 módulos NX5001 por half-cluster (ou seja, embora seja possível definir até 6 redes PROFIBUS, nem todas podem ser redundantes, pois isso exigiria 12 módulos NX5001).
- Definir a quantidade de redes Ethernet adicionais (entre 0 e 6) utilizando módulos NX5000, e se cada uma destas redes é simples (aloca um único NX5000) ou NIC Teaming (aloca dois NX5000). No exemplo seguinte, existe uma única rede NIC Teaming com modo de falha desabilitado (aloca dois NX5000). Observar que existe um limite de no máximo 6 módulos NX5000 por half-cluster (ou seja, embora seja possível definir até 6 redes Ethernet adicionais, nem todas podem ser redundantes, pois isso exigiria 12 módulos NX5000).

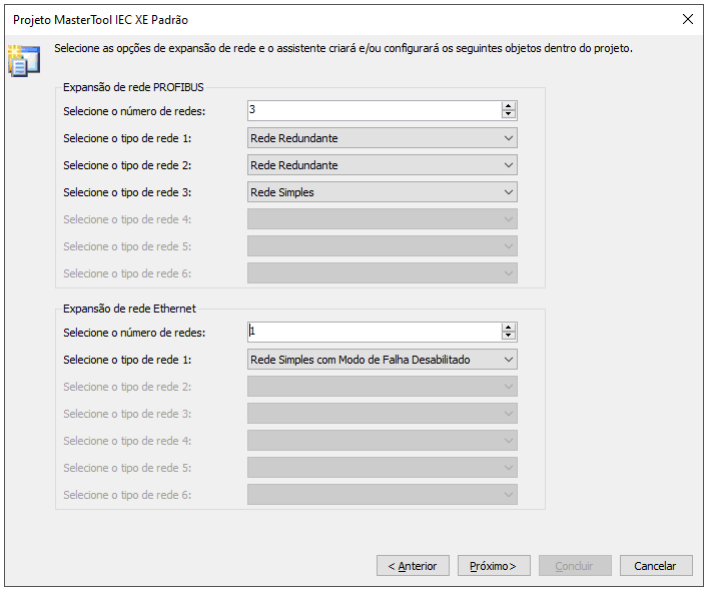


Figura 208: Terceira tela do Assistente

Clique no botão *Próximo>* para selecionar a quarta tela do Assistente. A quarta tela do Assistente é mostrada na figura seguinte, já preenchida com algumas opções compatíveis com a redundância do NX3035. Esta tela permite selecionar a linguagem padrão para criar os programas do usuário. Neste exemplo, foi selecionada a linguagem ST, mas existem outras opções (CFC, CFC page oriented, SFC, LD, FBD).

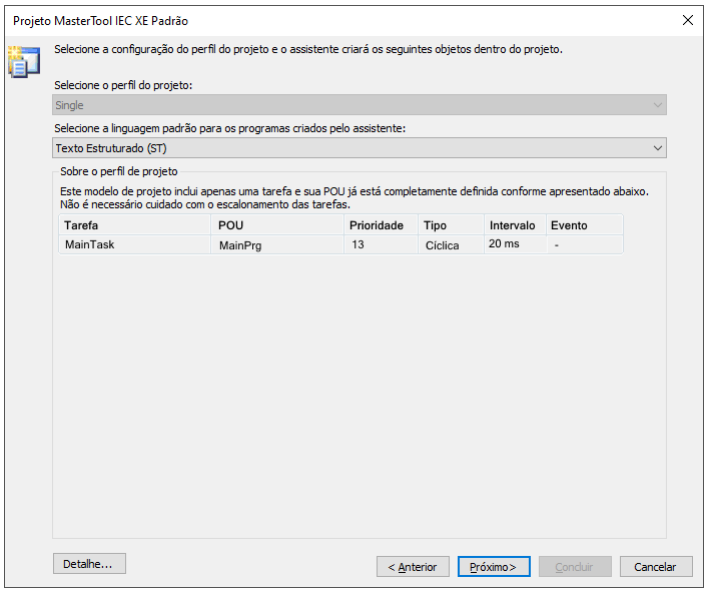


Figura 209: Quarta tela do Assistente

Clique no botão *Próximo>* para selecionar a quinta e última tela do Assistente. A quinta tela do Assistente é mostrada na figura seguinte, já preenchida com algumas opções compatíveis com a redundância do NX3035. Esta tela permite apenas selecionar as linguagens para editar as POUs StartPrg e ActivePrg. Neste exemplo, foi selecionada a linguagem ST para ambas as POUs, mas existem outras opções (CFC, CFC page oriented, SFC, LD, FBD).

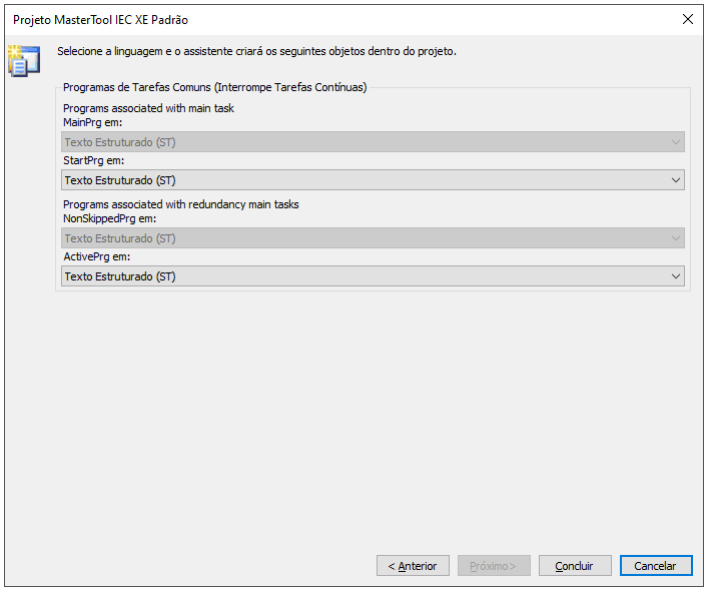


Figura 210: Quinta tela do Assistente

Para finalizar, o usuário deve pressionar o botão *Concluir*. Depois disso, o Assistente vai gerar configurações coerentes com as respostas do usuário para as cinco telas do Assistente. Por exemplo, considerando os exemplos das cinco telas anteriores, a seguinte configuração de hardware será gerada na aba *Configuration*.

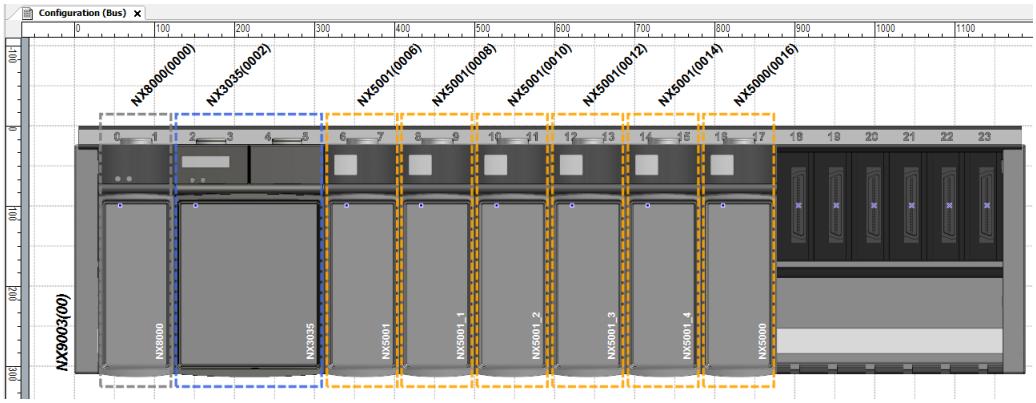


Figura 211: Configuração de hardware do half-cluster gerada pelo Assistente

6.4.2. Configuração dos Half-Clusters

O Assistente sempre é utilizado para gerar a primeira versão de um projeto redundante. Isso garante que a versão inicial do projeto será gerada rápida e corretamente.

No entanto, é possível que algumas modificações sejam necessárias em um half-cluster, tal como a inserção de novos módulos NX5001, NX5000, ou módulos de E/S locais, que podem ser feitas alterando a tela de configuração do half-cluster. Os capítulos a seguir mostram como adicionar e/ou configurar os módulos NX5000 e NX5001.

Algumas regras e precauções devem ser seguidas para um projeto redundante, conforme descreve a seção seguinte.

6.4.2.1. Configuração Fixa nas Posições 0 a 5 do Bastidor

Nas posições 0 a 5 do bastidor selecionado sempre devem estar instalados os seguintes módulos:

- Fonte de alimentação NX8000 (posição 0)
- UCP NX3035 (posição 2)

Estes módulos não devem ser removidos do projeto original gerado pelo Assistente. Qualquer configuração diferente nestas posições resultará em um erro notificado pelo Mastertool na compilação do projeto.

6.4.3. Configurações das Portas Ethernet Integradas de uma UCP NX3035 Redundante (NET 1 a NET 6)

6.4.3.1. Configuração de Endereços IP

Para as portas Ethernet integradas da UCP NX3035, este assunto foi tratado na seção Configuração de Endereços IP nas portas integradas NET1 ... NET6 de uma UCP NX3035 em um CP Redundante.

Para as portas Ethernet remotas dos NX50000, este assunto foi tratado na seção Configuração de Endereços IP em Módulos NX5000 em um CP Redundante.

6.4.3.2. Configuração de NIC Teaming

Para as portas Ethernet integradas da UCP NX3035, este assunto foi tratado na seção Configuração das Interfaces Ethernet Integradas, mais especificamente na seção Configuração de Modo das Portas Ethernet.

Para as portas Ethernet remotas dos NX5000, este assunto foi tratado na seção Endereçamento IP de Portas Ethernet Remotas do NX5000 (NET1), mais especificamente na seção Configuração de Modo da Porta Ethernet do NX5000.

6.4.3.3. Configurações dos Módulos NX5001

Este assunto foi tratado na seção Configuração de Redes PROFIBUS com Módulos NX5001, e se aplica a CPs redundantes e não redundantes.

Existem poucas diferenças na configuração de uma rede PROFIBUS que se aplicam somente a CPs redundantes:

- Existe um único parâmetro de configuração do módulo NX5001 que se aplica exclusivamente a CPs redundantes. Trata-se do parâmetro *Modo de Falha* que encontra-se na aba *Parâmetros do Módulo* do NX5001. Este parâmetro é descrito na seção Aba "Parâmetros do Módulo".
- O cão-de-guarda de cada remota PROFIBUS possui um valor mínimo a ser configurado num CP redundante. Este assunto é tratado na seção Configuração de Cão-de-Guarda em Remotas PROFIBUS em CPs Redundantes.

6.4.3.4. Configurações dos Módulos NX5000

Este assunto já foi tratado na seção Configuração de Interfaces Ethernet com Módulos NX5000.

Existem algumas diferenças na configuração de uma interface NX5000 que se aplicam somente a CPs redundantes:

- Devem ser configurados múltiplos endereços IP num CP redundante, ao invés de um único endereço IP como ocorre num CP não-redundante. O tipo e quantidade de endereços IP depende do parâmetro *Método de Endereçamento de IP do Cluster*, que só existe em projetos de CPs redundantes. Para maiores detalhes, ver seção Configurações Básicas da Porta Ethernet do NX5000 e seção Configuração de Endereços IP em Módulos NX5000 em um CP Redundante.
- Existe uma configuração de modo de falha, descrito na seção Configuração de Modo de Falha das Portas Ethernet.

6.4.3.5. Configurações da UCP NX3035 Relacionadas com Redundância

As configurações de redundância relacionadas às variáveis %I, %Q e %M podem ser acessadas através de um duplo clique sobre a UCP NX3035, seguido da seleção da aba *Parâmetros de Redundância*, mostrada na tela seguinte:

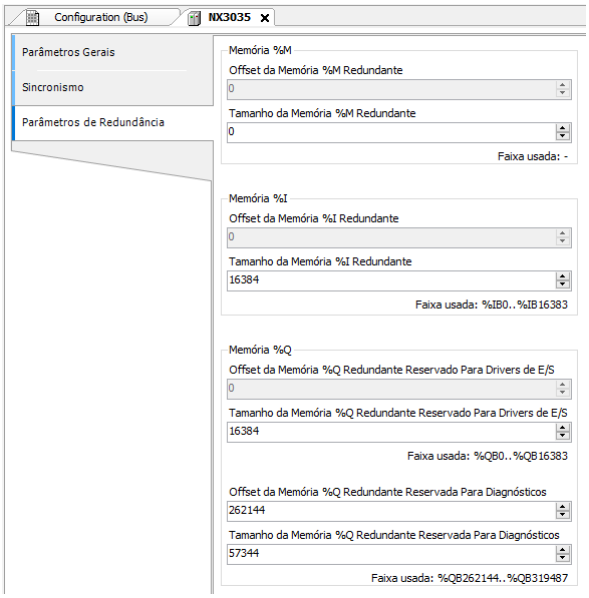


Figura 212: Parâmetros de redundância da UCP NX3035

Para entender estes parâmetros, deve-se ler as seções Variáveis %I Redundantes e Não-Redundantes, Variáveis %Q Redundantes e Não-Redundantes e Variáveis %M Redundantes e Não-Redundantes.

6.4.4. Configurações de Drivers de E/S

A configuração de drivers de E/S apresenta poucas diferenças em relação a um CP não-redundante. Estas diferenças geralmente consistem na utilização de comandos especiais executados no programa *NonSkippedPrg* e em alguns diagnósticos adicionais.

Considere um CP redundante que é cliente MODBUS TCP de um ou mais dispositivos de campo servidores MODBUS TCP. Neste exemplo, as seguintes estratégias diferenciadas podem ser adotadas em um CP redundante:

- Recomenda-se que a comunicação cliente seja desabilitada no CP Não-Ativo, para não sobrecarregar os dispositivos de campo com comunicações desnecessárias, já que basta que tais comunicações sejam realizadas somente entre o CP Ativo e os dispositivos de campo. Para tanto, no CP Não-Ativo, sugere-se desabilitar todos os mapeamentos do driver MODBUS TCP cliente para comunicação com os dispositivos de campo, ligando diversos bits de desabilitação de mapeamentos.
- É importante verificar se as interfaces de comunicação Ethernet do CP Não-Ativo estão íntegras, para evitar falhas ocultas que seriam descobertas somente quando este CP chaveasse para Ativo. Uma possível estratégia para isso é definir um mapeamento de teste adicional, onde o CP Ativo como cliente executa periodicamente uma comunicação de teste onde o CP Não-Ativo é servidor. Este mapeamento de teste deve ser habilitado somente no CP Ativo, e desabilitado no CP Não-Ativo. O diagnóstico de falha deste mapeamento de teste deve ser utilizado pela equipe de manutenção para reparar uma possível falha que impede a comunicação entre o CP Ativo e o CP Não-Ativo.
- Caso o CP Ativo perca comunicação com todos os dispositivos de campo, e além disso perca a comunicação de teste com o CP Não-Ativo, uma opção pode ser chavear o CP Não-Ativo para Ativo, desde que o CP Não-Ativo esteja em estado Reserva. Existem comandos que permitem fazer este switch-over.

A seção Exemplo de Gerenciamento de um Cliente MODBUS TCP num CP Redundante mostra um exemplo prático de aplicação das estratégias definidas anteriormente.

No caso de um servidor MODBUS TCP, algumas estratégias também podem ser utilizadas. Considere, por exemplo, um sistema SCADA que é cliente MODBUS TCP do CP redundante. Geralmente o sistema SCADA fará comunicações utilizando três endereços IP diferentes (IP Ativo para comunicações relacionadas ao processo controlado, IP da UCP A para diagnósticos e comandos exclusivos do CPA, e IP da UCP B para diagnósticos e comandos exclusivos do CPB). Neste caso, pode-se pensar em duas estratégias para tratar o problema:

1. Na primeira estratégia, o operador do SCADA receberá um alarme informando que a comunicação foi perdida com o CP Ativo. Caso a comunicação continue funcionando com o CP Não-Ativo (via IP da UCP A ou IP da UCP B), o operador pode mandar um comando especial do SCADA para o CP Não-Ativo, solicitando que o mesmo chaveie para Ativo. Para executar tal switch-over, o CP Não-Ativo liga o bit de comando *DG_NX3035.tRedundancy.RedCmdLoc.bStandbyRemote*.
2. A segunda estragégia é totalmente automática, sem necessidade de intervenção do operador:

a) O CP Ativo detecta que seu servidor MODBUS TCP não está mais sendo acessado pelo cliente SCADA. Isso é verificado através do congelamento, por alguns segundos, de um diagnóstico do servidor MODBUS (*tStat.wRXRequests*) que conta o número de requisições recebidas do cliente SCADA.

b) O CP Não Ativo pode disponibilizar o seu diagnóstico do servidor MODBUS (*tStat.wRXRequests*) para o CP Ativo, copiando este diagnóstico para dois bytes da estrutura *DG_NX3035.tRedundancy.RedUsrLoc*. No CP Ativo, este diagnóstico é lido em dois bytes da estrutura *DG_NX3035.tRedundancy.RedUsrRem*.

c) Caso o CP Ativo perceba que não está sendo acessado pelo SCADA, mas o CP Não-Ativo continua sendo acessado, pode chavear para Reserva utilizando o comando *DG_NX3035.tRedundancy.RedCmdLoc.bStandbyLocal*.

No caso da rede PROFIBUS, também existem comandos especiais diferenciados para os CPs em estado Ativo e Não-Ativo. Neste caso, no entanto, o gerenciamento da redundância executa tais comandos de forma automática, sem que seja necessário qualquer gerenciamento por parte do usuário.

6.4.5. Configurações da MainTask

Num CP redundante, toda a aplicação do usuário deve ser executada dentro do contexto de uma tarefa cíclica chamada MainTask (dentro das POUs NonSkippedPrg e ActivePrg).

A tela de configurações da tarefa MainTask pode ser acessada clicando sobre o objeto MainTask na árvore de dispositivos. Dois parâmetros devem ser ajustados nesta tela:

- Intervalo da MainTask
- Tempo de Cão-de-Guarda

Diversas considerações devem ser feitas para ajustar adequadamente o intervalo da MainTask:

- O intervalo deve ser suficientemente baixo para controlar o processo efetivamente, observando os tempos de resposta de todos os laços de controle.
- O intervalo deve ser suficientemente alto para acomodar, no mínimo, a soma dos dois seguintes tempos:
  - O tempo máximo de execução das POUs NonSkippedPrg e ActivePrg, em conjunto.
  - O tempo necessário para gerenciar a redundância (overhead da redundância). Uma estimativa deste tempo é calculada pelo Mastertool. Tal estimativa só é confiável depois que o projeto estiver completo, com todas as POUs desenvolvidas, e áreas de memória redundantes definidas.
- Além disso, o intervalo deve ter uma folga adicional, necessária para que outras tarefas da UCP tenham tempo de ser executados (comunicações PROFIBUS, comunicações Ethernet, etc).

O tempo de ciclo médio da tarefa MainTask pode ser medido clicando sobre o objeto *Task Configuration* e depois selecionando a aba *Monitorar*. Sugere-se que isso seja feito depois que o projeto esteja completo, para eventualmente então modificar o intervalo da tarefa MainTask.

Caso o ciclo médio medido da tarefa MainTask supere 70% do intervalo configurado da MainTask, sugere-se aumentar o intervalo para que haja uma folga da ordem de 30% para as demais tarefas.

6.4.5.1. Programa ActivePrg

Nesta POU o usuário deve criar a aplicação principal, responsável pelo controle de seu processo. Esta POU é chamada pela POU principal (MainPrg), sendo executada apenas no CP Ativo.

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

Deve-se lembrar que todas as variáveis simbólicas definidas na POU ActivePrg, assim como instâncias de blocos funcionais definidas, serão variáveis redundantes. Variáveis simbólicas definidas em POUs adicionais do tipo programa que forem chamadas dentro de ActivePrg também serão variáveis redundantes.

ATENÇÃO

Variáveis do tipo VAR_TEMP não devem ser utilizadas no programa redundante.

6.4.5.2. Programa NonSkippedPrg

Esta POU é destinada para controles que devem ser executados em ambos os CPs (CPA e CPB), independente do seu estado de redundância. Esta POU também é chamada pela POU principal (MainPrg).

Deve-se lembrar que todas as variáveis simbólicas definidas na POU NonSkippedPrg, assim como instâncias de blocos funcionais definidas, serão variáveis não-redundantes. O usuário pode também criar POUs adicionais (programa, funções ou bloco funcional), e chamá-las ou instanciá-las dentro da POU NonSkippedPrg, para fins de estruturação de seu programa. Também é possível chamar funções e instanciar blocos funcionais definidos em bibliotecas.

ATENÇÃO

Quando chamar POUs adicionais do tipo programa dentro de NonSkippedPrg, desmarque as mesmas no objeto *Redundancy Configuration* do MasterTool. Por padrão as variáveis simbólicas declaradas dentro destas POUs serão redundantes e, dentro da NonSkippedPrg, normalmente deseja-se variáveis não-redundantes.

Exemplos típicos de controles executados na NonSkippedPrg são os seguintes:

- Criar uma estrutura compacta de diagnósticos (%Q) a serem reportados para um sistema SCADA, a partir de uma estrutura completa de diagnósticos, onde diversos diagnósticos não são de interesse para o sistema SCADA. Estes diagnósticos podem ser extraídos de estruturas de dados como *RedDgnLoc*, *RedDgnRem*, *RedUsrLoc*, *RedUsrRem*, etc.
- Copiar comandos recebidos de um SCADA para os respectivos campos da estrutura de dados *RedCmdLoc*, e intertravar estes comandos se necessário
- Gerenciar switch-overs controlados pelo usuário, em função de falhas tais como a comunicação com um sistema SCADA ou com dispositivos servidores MODBUS TCP.
- Habilitar ou desabilitar comunicações dependendo do estado da redundância (Ativo ou Não-Ativo). Por exemplo, um driver cliente MODBUS TCP poderia desabilitar suas comunicações no CP Não-Ativo para não gerar requisições inúteis para servidores.

ATENÇÃO

Não se recomenda utilizar os blocos funcionais *TOF_RET*, *TON_RET*, *TOF* e *TON* no programa *NonSkippedPrg*. Ver seção Limitações na Programação de um CP Redundante.

6.4.6. Objeto *Redundancy Configuration*

Este objeto, localizado na árvore de dispositivos do projeto, é criado automaticamente pelo Assistente ao criar um novo projeto redundante. Ele é utilizado para determinar quais POUs e GVLs serão redundantes. Variáveis declaradas dentro de POUs e GVLs redundantes serão redundantes e, portanto, serão sincronizadas entre os CPs.

Por padrão, POUs e GVLs criadas pelo usuário são marcadas como redundantes no objeto *Redundancy Configuration*, cabendo ao usuário reverter a marcação quando deseja que sejam não-redundantes. Por exemplo, ao criar uma nova POU que será chamada dentro da POU *NonSkippedPrg*, normalmente esta nova POU deverá ser revertida para não-redundante dentro do objeto *Redundancy Configuration*.

Algumas POUs e GVLs tem configuração fixa (não editável) dentro do objeto *Redundancy Configuration*:

- POUs nunca redundantes:
  - MainPrg
  - NonSkippedPrg
  - SpecialVariablesPrg
- GVLs nunca redundantes:
  - System_Diagnostics
  - Module_Diagnostics
  - ReqDiagnostics
  - IOQualities
  - SpecialVariables

ATENÇÃO

A quantidade máxima de objetos redundantes é limitada a 1000.

6.4.7. GVL *Module_Diagnostics* e GVL *System_Diagnostics*

Estas duas GVLs especiais são criadas e preenchidas automaticamente pelo Assistente ao criar um projeto redundante novo, e também ao inserir ou remover objetos na árvore de dispositivos. Elas não podem ser modificadas pelo usuário.

Dentro destas GVLs, diversas estruturas de dados são criadas automaticamente, contendo diagnósticos e comandos. Estes diagnósticos e comandos residem em variáveis de representação direta (%Q ou %I).

Estas GVLs contêm diversas sentenças com a diretiva AT para definir nomes simbólicos para estes diagnósticos e comandos. Desta maneira, quando o usuário precisar referenciar estas variáveis, recomenda-se usar um nome simbólico ao invés da referência numérica ao lado do %Q ou %I.

6.4.8. GVLs com Variáveis Simbólicas Redundantes ou Não-Redundantes

O usuário pode criar outras GVLs, diferentes das citadas anteriormente, para declarar variáveis simbólicas redundantes ou não-redundantes.

Para que as variáveis sejam redundantes, a GVL deve estar marcada como redundante no objeto *Redundancy Configuration*, na árvore de dispositivos do projeto, significando que essa GVL é redundante.

Por padrão, todas uma GVL criada pelo usuário será, inicialmente, redundante. Caso o usuário a deseje não-redundante, deverá desmarcá-la no objeto *Redundancy Configuration*.

6.4.9. GVLs com Diretivas AT

GVLs com diretivas AT podem ser utilizadas para definir nomes simbólicos para variáveis de representação direta (%I, %Q ou %M).

Caso haja variáveis definidas com diretiva AT dentro de uma GVL:

- Esta GVL deverá ser marcada como não-redundante no objeto *Redundancy Configuration*. Caso contrário, o Mastertool detectará um erro de compilação.
- Mesmo que uma GVL com diretivas AT seja marcada como não-redundante, ela poderá apontar para variáveis %I, %Q ou %M redundantes. Para verificar se uma variável %I, %Q ou %M é redundante ou não-redundante, ver seção Configurações da UCP NX3035 Relacionadas com Redundância.

6.4.10. POUs do Tipo Programa com Variáveis Simbólicas Redundantes ou Não-Redundantes

Variáveis simbólicas declaradas dentro de uma POU do tipo programa serão redundantes ou não, dependendo de como a POU está configurada dentro do objeto *Redundancy Configuration*.

Para definir uma nova POU como redundante deve-se, após a criação da POU, marcá-la na configuração do objeto *Redundancy Configuration*, na árvore de dispositivos do projeto. Por padrão, todas as POUs criadas pelo usuário serão, inicialmente, redundantes.

Caso uma POU seja marcada como redundante, não é possível utilizar a diretiva AT dentro da mesma para definir nomes simbólicos para variáveis de representação direta (%I, %Q ou %M). O Mastertool detectará um erro de compilação neste caso.

6.4.11. Utilização de Breakpoints em Sistemas Redundantes

Para sistemas redundantes, deve-se utilizar breakpoints apenas no half-cluster Ativo enquanto o outro half-cluster estiver num estado diferente de Reserva.

O usuário será impedido de ativar breakpoints no CP Ativo enquanto o outro CP estiver Reserva.

Além disso, o usuário deve lembrar o seguinte:

- Remover todos os breakpoints do CP Ativo ao terminar a depuração.
- Remover todos os breakpoints do CP Ativo antes chavear o outro CP para reserva.

6.4.12. Limitações na Programação de um CP Redundante

Em um CP redundante, existem algumas limitações quanto à programação das UCPs. Estas limitações são tratadas nas subseções abaixo.

6.4.12.1. Limitações em GVLs e POUs Redundantes

Em uma GVL ou em uma POU do tipo programa que sejam redundantes, as seguintes limitações devem ser respeitadas para um correto funcionamento dos half-clusters:

- Não utilizar variáveis do tipo VAR_TEMP.
- Não misturar tipos de variáveis (VAR, VAR RETAIN, VAR PERSISTENT, VAR CONSTANT etc.), devendo ser utilizado somente um dos tipos em cada GVL ou POU.
- Não misturar declaração de variáveis simbólicas com ATs nas GVLs e POUs. Devem ser criadas GVLs ou POUs separadas, declarando em uma as variáveis do tipo AT e em outra as variáveis simbólicas. Normalmente, deve-se utilizar ATs somente em GVLs.
- Não armazenar o endereço de uma variável em uma variável redundante (fazer de uma variável redundante um ponteiro para um endereço), pois os endereços das variáveis podem ser diferentes no CPA e no CPB.
- Não utilizar blocos funcionais de escrita e leitura ao relógio RTC em POUs redundantes. Mais detalhes podem ser encontrados na seção Relógio RTC.

6.4.12.2. Limitações em POUs Não-Redundantes

Em uma POU do tipo programa que não seja redundante, as seguintes limitações devem ser respeitadas para um correto funcionamento dos half-clusters:

- Não podem ser utilizados os blocos funcionais temporizadores TON e TOF tradicionais, pois os mesmos utilizam o temporizador IEC. Quando o CP Reserva entrar em estado ativo (com o outro half-cluster saindo do estado Ativo), o temporizador IEC será sincronizado, causando uma descontinuidade no valor do temporizador. Deve-se optar pelos blocos funcionais TON_NR e TOF_NR disponibilizados na biblioteca NextoStandard. Ver seção Temporizador Não-Redundante.
- Não podem ser utilizadas POUs do tipo programa escritas na linguagem SFC (Sequenciamento Gráfico de Funções), pois estas utilizam o temporizador IEC para temporizar as transições.

6.4.13. Obtendo o Estado da Redundância de um Half-Cluster

É possível verificar o estado da redundância de um half-cluster descrito na seção Estrutura de Diagnósticos, Comandos e Informações do Usuário para a Redundância:

```
VAR
  eRedStateLocal : IoDrvNextoRedundancy.REDUNDANCY_STATE;
END_VAR

eRedStateLocal := DG_NX3035.tRedundancy.RedDgnLoc.sGeneral_Diag.eRedState;
```

Deste modo, o usuário poderá fazer o controle de lógicas que dependam do estado redundante do CP.

6.4.14. Utilização de Diagnósticos Não-Redundantes

Um projeto redundante, além de utilizar diagnósticos redundantes (por exemplo, diagnósticos de uma remota PROFIBUS), também pode utilizar diagnósticos não redundantes (diagnósticos específicos dos módulos NX5000, NX5001, NX3035, etc.). Tais diagnósticos não-redundantes geralmente são utilizados dentro diretamente dentro da POU *NonSkippedPrg*, ou em alguma outra POU não-redundante chamada dentro da POU *NonSkippedPrg*.

Alguns destes diagnósticos não-redundantes poderão não ser válidos nos primeiros instantes após o CP assumir o estado Ativo, pois deverão ser recalculados pelo novo CP em estado Ativo, e isso pode levar algum tempo. Antes de utilizar um diagnóstico não-redundante, deve-se aguardar que o CP esteja em estado Ativo por algum tempo, tipicamente da ordem de 5 segundos. Seguem alguns exemplos de tais diagnóstico:

- **DG_NX5001.tGeneral.bPbusCommFail:** a rede PROFIBUS abaixo deste NX5001 tem falha de comunicação.
- **DG_NX5001.tGeneral.bSlaveWithDiagnostic:** pelo menos um escravo abaixo deste NX5001 tem falha de diagnósticos ativos.

A seguir mostra um exemplo de código na POU NonSkippedPrg para utilizar um diagnóstico deste tipo somente depois de 5 segundos em estado ativo.

```
PROGRAM NonSkippedPrg
VAR
  Active_Time : NextoStandard.TON_NR; // Temporizador não redundante: mede tempo
    em estado Ativo
END_VAR


Active_Time.IN:=System_Diagnostics.DG_NX3035.tRedundancy.RedDgnLoc.sGeneral_Diag
   .eRedState = IoDrvNextoRedundancy.REDUNDANCY_STATE.ACTIVE;
Active_Time.PT := T#5S;
Active_Time();

IF Active_Time.Q THEN
  ; // Trecho de código onde é possível utilizar diasgnósticos não-redundantes (
   mais de 5 segundos em Ativo)
ELSE
  ; // Trecho de código onde não possível utilizar diasgnósticos não-redundantes
    (menor de 5 segundos em Ativo)
END_IF
```

6.4.15. Exemplo de Gerenciamento de um Cliente MODBUS TCP num CP Redundante

Esta seção descreve um exemplo de gerenciamento de um cliente MODBUS TCP em um CP redundante, considerando a arquitetura da figura seguinte.

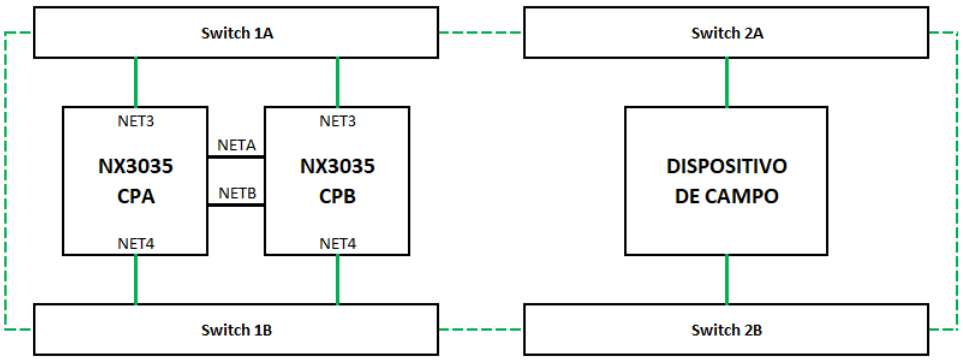


Figura 213: Arquitetura do exemplo de comunicação cliente MODBUS TCP

Na arquitetura, observa-se que os dois half-clusters redundantes (CPA e CPB), como clientes MODBUS TCP, devem se comunicar com um dispositivo de campo, que é um servidor MODBUS TCP. Trata-se de uma arquitetura de alta disponibilidade, devido às seguintes características:

- Os half-clusters CPA e CPB utilizam um par NIC Teaming (NET3+NET4), sendo que as portas NET3 se conectam ao switch 1A enquanto as portas NET4 se conectam ao switch 1B. Desta forma, tolera-se falha individual de uma porta (NET3 ou NET4), bem como falha individual de um switch (1A ou 1B).
- Assume-se que o dispositivo de campo também tenha capacidade de se conectar via NIC Teaming, e suas duas portas se conectam a dois switches diferentes (2A e 2B).
- Finalmente, os quatro switches estão conectados em anel, para tolerância contra uma falha individual de interconexão entre switches.

A aplicação desenvolvida para controlar esta comunicação tem as seguintes características, já mencionadas na seção Configurações de Drivers de E/S:

- Mapeamentos são desabilitados no CP Não-Ativo, para que o mesmo não gere requisições de comunicação desnecessárias para o dispositivo de campo. Apenas o CP Ativo gera requisições de comunicação para o dispositivo de campo. Desta forma, evita-se uma sobrecarga desnecessária de comunicações na rede e para o dispositivo de campo.
- O CP Ativo calculará os seguintes diagnósticos de falha:
  - Falha dos mapeamentos utilizados para a comunicação com o dispositivo de campo. No exemplo que será apresentado, são dois mapamentos. O mapeamento 0 é utilizado para ler 120 holding registers do dispositivo de campo, enquanto o mapeamento 1 é utilizado para escrever 120 holding registers no dispositivo de campo.
  - Falha de comunicação contra o CP Não-Ativo. Neste caso, o CP Ativo age como cliente e o CP Não-Ativo como servidor. O objetivo desta comunicação é fazer um teste da integridade de comunicação do CP Não-Ativo, para evitar que haja falhas ocultas no CP Não-Ativo. Isto é necessário porque o CP Não-Ativo não gera requisições de comunicação como cliente.
  - Falha geral de comunicação, que ocorre quando existe falha simultânea em todos os mapeamentos de comunicação com o dispositivo de campo, e além disso existe falha na comunicação com o CP Não-Ativo.
- Um switch-over poderá levar o CP Ativo para o estado Reserva, e o CP Reserva para o estado Ativo. Para tanto, três condições devem ocorrer simultaneamente:
  1. O CP Ativo, antes de ir para Reserva, deve ter permanecido em Ativo por um tempo mínimo de 15 segundos.
  2. O CP Não-Ativo deve estar em estado Reserva, ou seja, apto a chavear para o estado Ativo.
  3. Deve existir uma falha geral de comunicação com duração mínima de 5 segundos, ou então link-downs simultâneos nas portas NET3 e NET4 do CP Ativo.

A seguinte seções descrevem detalhes da aplicação desenvolvida para este exemplo.

6.4.15.1. Configuração das Interfaces NET3 / NET4 do CP Redundante

A configuração das interfaces NET3 e NET4 do CP Redundante é mostrada na figura seguinte:

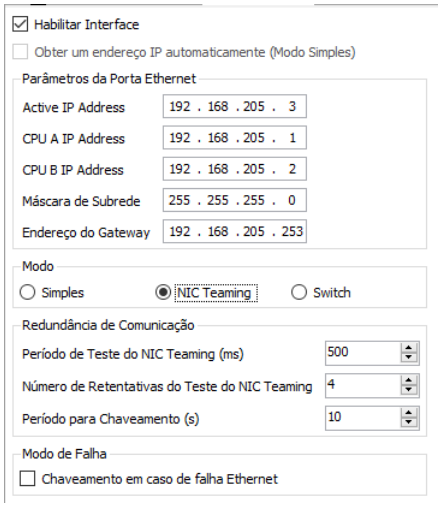


Figura 214: Configurações da NET3 e NET4

Observa-se que NET3+NET4 forma um par NIC Teaming.

6.4.15.2. Configuração do Servidor MODBUS TCP para Comunicação de Teste

É necessário configurar um servidor MODBUS TCP abaixo das portas NET3+NET4 do CP redundante, para que seja realizada a comunicação de teste onde o CP Ativo é cliente e o CP Não-Ativo é servidor.

A figura seguinte mostra a configuração deste servidor, que disponibiliza um único holding register para ser lido pelo cliente.

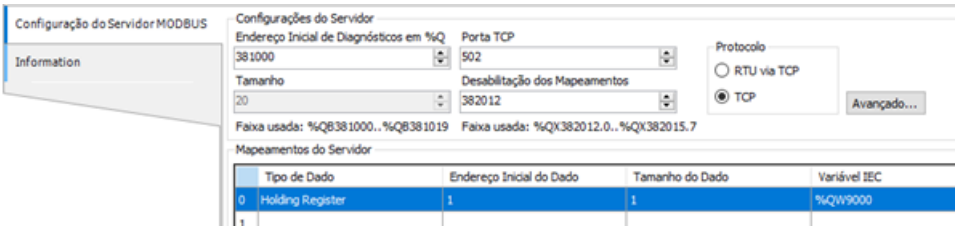


Figura 215: Configurações do servidor MODBUS TCP abaixo da NET3 e NET4

6.4.15.3. Configuração do Cliente MODBUS TCP

A figura seguinte mostra um panorama geral da configuração do cliente MODBUS TCP abaixo das portas NET3+NET4 do CP redundante.

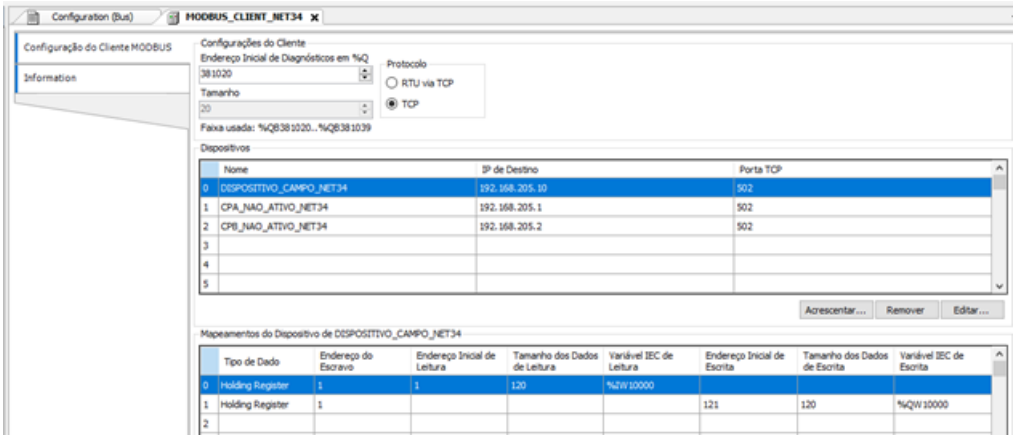


Figura 216: Configurações do cliente MODBUS TCP abaixo da NET3 e NET4

Cabem as seguintes observações sobre os três dispositivos servidores configurados neste cliente MODBUS TCP:

- O primeiro dispositivo servidor (FIELD_DEVICE_NET34) corresponde ao dispositivo de campo (endereço IP = 192.168.205.10), e contém dois mapeamentos (0 e 1). O mapeamento 0 configura a leitura de 120 holding registers, e o mapeamento 1 configura a escrita de 120 holding registers.
- O segundo dispositivo servidor (CPA_NOT_ACTIVE_NET34) corresponde ao CPA (endereço IP = 192.168.205.1), e contém um único mapeamento (0), que configura a leitura de 1 holding register. Este mapeamento só deve ser habilitado quando o CPB for o CP Ativo (cliente), e o CPA for o CP Não-Ativo (servidor). O objetivo deste mapeamento é testar a integridade da comunicação do CPA enquanto o mesmo estiver no estado Não-Ativo.
- O terceiro dispositivo servidor (CPB_NOT_ACTIVE_NET34) corresponde ao CPB (endereço IP = 192.168.205.2), e contém um único mapeamento (0), que configura a leitura de 1 holding register. Este mapeamento só deve ser habilitado quando o CPA for o CP Ativo (cliente), e o CPB for o CP Não-Ativo (servidor). O objetivo deste mapeamento é testar a integridade da comunicação do CPB enquanto o mesmo estiver no estado Não-Ativo.

6.4.15.4. GVLs e POUs para Gerenciamento do Cliente MODBUS TCP

A aplicação para gerenciar o cliente MODBUS TCP deste exemplo consiste de duas GVLs e uma POU.

A figura seguinte mostra a *GVL_AT_MODBUS_CLIENT_NET34*:

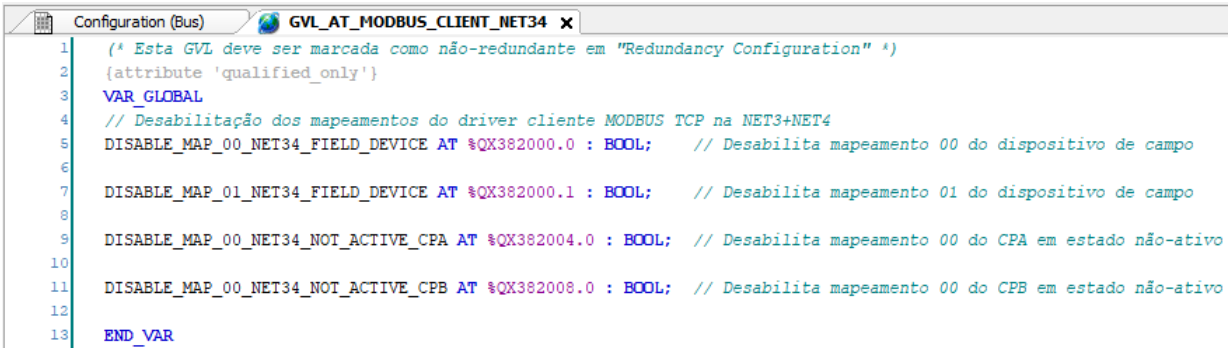


Figura 217: GVL_AT_MODBUS_CLIENT_NET34

```
(*Esta GVL deve ser marcada como não-redundante em "Redundancy Configuration"*)
{attribute 'qualified_only'}
VAR_GLOBAL
  // Desabilitação dos mapeamentos do driver cliente MODBUS TCP na NET3+NET4
  DISABLE_MAP_00_NET34_FIELD_DEVICE AT %QX382000.0 : BOOL;  // Desabilita
   mapeamento 00 do dispositivo de campo

  DISABLE_MAP_01_NET34_FIELD_DEVICE AT %QX382000.1 : BOOL;  // Desabilita
   mapeamento 01 do dispositivo de campo

  DISABLE_MAP_00_NET34_NOT_ACTIVE_CPA AT %QX382004.0 : BOOL;  // Desabilita
   mapeamento 00 do CPA em estado não-ativo

  DISABLE_MAP_00_NET34_NOT_ACTIVE_CPB AT %QX382008.0 : BOOL;  // Desabilita
   mapeamento 00 do CPB em estado não-ativo

END_VAR
```

A GVL da figura anterior simplesmente utiliza diretivas AT para definir nomes simbólicos para os bits de desabilitação dos 4 mapeamentos descritos anteriormente. É importante ressaltar que esta GVL deve ser marcada como não-redundante no objeto *Redundancy Configuration*.

A figura seguinte mostra a GVL_MODBUS_CLIENT_NET34:

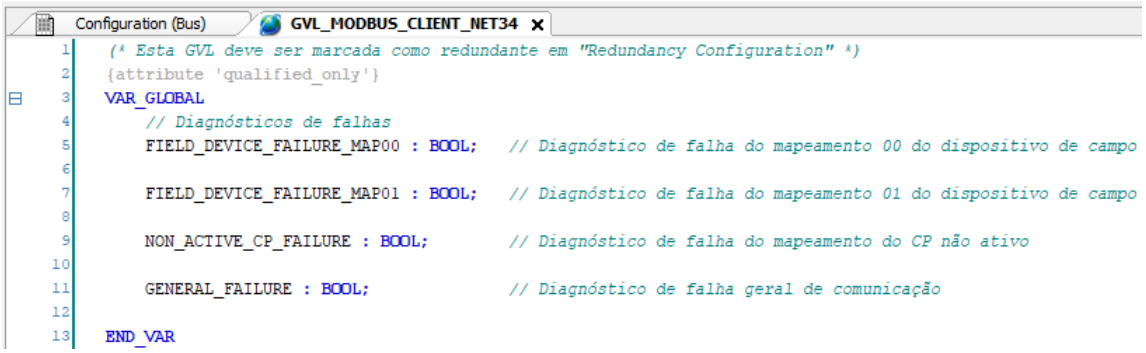


Figura 218: GVL_MODBUS_CLIENT_NET34

```
(* Esta GVL deve ser marcada como redundante em "Redundancy Configuration" *)
{attribute 'qualified_only'}
VAR_GLOBAL
  // Diagnósticos de falhas
  FIELD_DEVICE_FAILURE_MAP00 : BOOL;  // Diagnóstico de falha do mapeamento 00
   do dispositivo de campo

  FIELD_DEVICE_FAILURE_MAP01 : BOOL;  // Diagnóstico de falha do mapeamento 01
   do dispositivo de campo

  NON_ACTIVE_CP_FAILURE : BOOL;   // Diagnóstico de falha do mapeamento do CP nã
   o ativo

  GENERAL_FAILURE : BOOL;       // Diagnóstico de falha geral de comunicação

END_VAR
```

A GVL da figura anterior define variáveis para diagnósticos de falhas. É importante ressaltar que esta GVL deve ser marcada como redundante no objeto *Redundancy Configuration*.

Finalmente, a figura seguinte mostra a POU *Manage_MODBUS_CLIENT_NET34*:

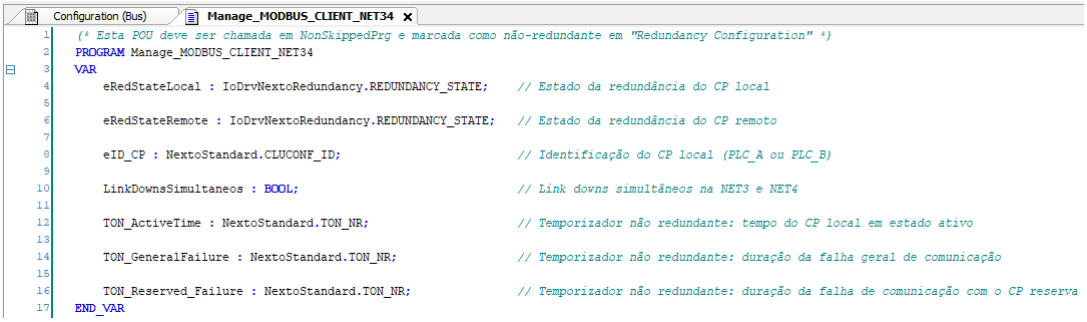


Figura 219: Manage_MODBUS_CLIENT_NET34– seção de variáveis



Figura 220: Manage_MODBUS_CLIENT_NET34– seção de código

```
(* Esta POU deve ser chamada em NonSkippedPrg e marcada como não-redundante em "
   Redundancy Configuration" *)
PROGRAM Manage_MODBUS_CLIENT_NET34
VAR
  eRedStateLocal : IoDrvNextoRedundancy.REDUNDANCY_STATE;   // Estado da redundâ
   ncia do CP local

  eRedStateRemote : IoDrvNextoRedundancy.REDUNDANCY_STATE;  // Estado da redundâ
   ncia do CP remoto

  eID_CP : NextoStandard.CLUCONF_ID;              // Identificação do CP local (
   PLC_A ou PLC_B)

  LinkDownsSimultaneos : BOOL;                 // Link downs simultâneos na NET3
   e NET4

  TON_ActiveTime : NextoStandard.TON_NR;            // Temporizador não
   redundante: tempo do CP local em estado ativo

  TON_GeneralFailure : NextoStandard.TON_NR;          // Temporizador não
   redundante: duração da falha geral de comunicação
```

```
  TON_Reserved_Failure : NextoStandard.TON_NR;        // Temporizador não
    redundante: duração da falha de comunicação com o CP reserva

END_VAR



// Captura estado de redundância dos CPs local e remoto, e identificação do CP
    local
eRedStateLocal := System_Diagnostics.DG_NX3035.tRedundancy.RedDgnLoc.
    sGeneral_Diag.eRedState;
eRedStateRemote := System_Diagnostics.DG_NX3035.tRedundancy.RedDgnRem.
    sGeneral_Diag.eRedState;
eID_CP := System_Diagnostics.DG_NX3035.tRedundancy.RedDgnLoc.sGeneral_Diag.
    ePLC_ID;
// Habilitação e desabilitação dos mapeamentos do cliente MODBUS
IF (eRedStateLocal = IoDrvNextoRedundancy.REDUNDANCY_STATE.ACTIVE) THEN
  // No CP ativo, habilita mapeamentos com dispositivo de campo
  GVL_AT_MODBUS_CLIENT_NET34.DISABLE_MAP_00_NET34_FIELD_DEVICE := FALSE;
  GVL_AT_MODBUS_CLIENT_NET34.DISABLE_MAP_01_NET34_FIELD_DEVICE := FALSE;
  IF eID_CP = PLC_A THEN
    // No CPA como ativo, desabilita mapeamento para diagnóstico do CPA e
    habilita mapeamento para diagnóstico do CPB
    GVL_AT_MODBUS_CLIENT_NET34.DISABLE_MAP_00_NET34_NOT_ACTIVE_CPA := TRUE;
    GVL_AT_MODBUS_CLIENT_NET34.DISABLE_MAP_00_NET34_NOT_ACTIVE_CPB := FALSE;
  ELSE
    // No CPB como ativo, habilita mapeamento para diagnóstico do CPA e
    desabilita mapeamento para diagnóstico do CPB
    GVL_AT_MODBUS_CLIENT_NET34.DISABLE_MAP_00_NET34_NOT_ACTIVE_CPA := FALSE;
    GVL_AT_MODBUS_CLIENT_NET34.DISABLE_MAP_00_NET34_NOT_ACTIVE_CPB := TRUE;
  END_IF
ELSE
  // No CP não-ativo, desabilita todos os mapeamentos
  GVL_AT_MODBUS_CLIENT_NET34.DISABLE_MAP_00_NET34_FIELD_DEVICE := TRUE;
  GVL_AT_MODBUS_CLIENT_NET34.DISABLE_MAP_01_NET34_FIELD_DEVICE := TRUE;
  GVL_AT_MODBUS_CLIENT_NET34.DISABLE_MAP_00_NET34_NOT_ACTIVE_CPA := TRUE;
  GVL_AT_MODBUS_CLIENT_NET34.DISABLE_MAP_00_NET34_NOT_ACTIVE_CPB := TRUE;
END_IF
// Calcula diagnósticos dos mapeamentos do cliente MODBUS que estão habilitados
    no CP ativo
IF (eRedStateLocal = IoDrvNextoRedundancy.REDUNDANCY_STATE.ACTIVE) THEN
  GVL_MODBUS_CLIENT_NET34.FIELD_DEVICE_FAILURE_MAP00 := NOT(
    DG_FIELD_DEVICE_NET34_Mapping_000.byStatus.bCommOk);
  GVL_MODBUS_CLIENT_NET34.FIELD_DEVICE_FAILURE_MAP01 := NOT(
    DG_FIELD_DEVICE_NET34_Mapping_001.byStatus.bCommOk);
  IF eID_CP = PLC_A THEN
    GVL_MODBUS_CLIENT_NET34.NON_ACTIVE_CP_FAILURE := NOT(
    DG_CPB_NOT_ACTIVE_NET34_Mapping_000.byStatus.bCommOk);
  ELSE
    GVL_MODBUS_CLIENT_NET34.NON_ACTIVE_CP_FAILURE := NOT(
    DG_CPA_NOT_ACTIVE_NET34_Mapping_000.byStatus.bCommOk);
  END_IF
  GVL_MODBUS_CLIENT_NET34.GENERAL_FAILURE := GVL_MODBUS_CLIENT_NET34.
    FIELD_DEVICE_FAILURE_MAP00 AND
                        GVL_MODBUS_CLIENT_NET34.FIELD_DEVICE_FAILURE_MAP01 AND
```

```
                        GVL_MODBUS_CLIENT_NET34.NON_ACTIVE_CP_FAILURE;
END_IF
// Calcula link-downs simultâneos em NET3 e NET4 do CP local
LinkDownsSimultaneos := System_Diagnostics.DG_NX3035.tDetailed.Ethernet.NET[3].
    bLinkDown AND DG_NX3035.tDetailed.Ethernet.NET[4].bLinkDown;
// Comanda switchover do CP local de ativo para reserva, quando todas as três
    seguintes condições forem verdadeiras:
// 1) CP local está em estado ativo por tempo superior a 15 segundos.
// 2) CP remoto está em estado reserva e, portanto, apto a chavear para ativo.
// 3) Falha geral por tempo superior a 5 segundos, ou link-downs simultâneos em
    NET3 e NET4.
TON_ActiveTime(IN:= (eRedStateLocal = IoDrvNextoRedundancy.REDUNDANCY_STATE.
    ACTIVE), PT:= T#15S);
TON_GeneralFailure(IN:= (eRedStateLocal = IoDrvNextoRedundancy.REDUNDANCY_STATE.
    ACTIVE) AND GVL_MODBUS_CLIENT_NET34.GENERAL_FAILURE, PT:= T#5S);
IF TON_ActiveTime.Q AND (eRedStateRemote = IoDrvNextoRedundancy.REDUNDANCY_STATE
    .STANDBY) AND (TON_GeneralFailure.Q OR LinkDownsSimultaneos) THEN
  // Comando para chavear de ativo para reserva
  System_Diagnostics.DG_NX3035.tRedundancy.RedCmdLoc.bStandbyLocal := TRUE;
END_IF
```

É importante ressaltar que esta POU deve ser marcada como não-redundante no objeto *Redundancy Configuration*, e que deve ser chamada dentro da POU *NonSkippedPrg*. As figuras anteriores possuem comentários que explicam suficientemente o funcionamento desta POU.

6.5. Carga de Programas em um CP Redundante

A seção Programação de um CP Redundante tratou de aspectos relativos ao desenvolvimento de um projeto para um CP redundante com UCP NX3035.

Nesta seção, discutem-se métodos e etapas para carregar este projeto em um CP redundante, considerando as seguintes situações:

- Carga do projeto em uma UCP nova retirada da caixa, ou em uma UCP contendo um projeto desconhecido.
- Carga on-line de modificações.
- Carga off-line de modificações com interrupção do controle do processo, durante uma parada programada do processo.

Além disso, existe outra situação mais complexa, que será discutida num capítulo separado: Carga Offline de Programas sem Interrupção do Controle do Processo. Neste caso, mesmo que a modificação do projeto demande uma carga off-line, utilizam-se características especiais da redundância para não interromper o controle do processo.

6.5.1. Carga Inicial de um Projeto Redundante

Esta seção descreve um procedimento seguro para fazer a primeira carga de projeto redundante em uma UCP NX3035. Isto é necessário, por exemplo, para uma UCP nova de fábrica recém tirada da caixa, ou para uma UCP que contenha um projeto desconhecido.

Assume-se o pior caso de que a UCP possa contar um projeto desconhecido, e que sua inclusão numa rede de automação possa causar efeitos imprevisíveis, tais como endereços IP conflitantes com outros equipamentos instalados nesta rede de automação. Por este motivo, o procedimento seguro descrito a seguir contém precauções tais como isolar esta UCP completamente da rede de automação, até que um projeto conhecido esteja carregado na mesma.

Os passos seguintes devem ser executados para os dois half-clusters (CPA e CPB) que compõem um CP redundante. Primeiro deve-se executar todos os passos para um dos half-clusters (ex: CPA), e depois para o outro (ex: CPB).

1. Inicie o procedimento com a UCP desenergizada.
2. Desconecte todas as portas Ethernet da UCP (NET1 ... NET6), bem como as portas da rede de sincronismo NETA e NETB.
3. Inicie a UCP sem aplicação, executando o procedimento de Não Carregar a Aplicação na Inicialização.

4. Utilize um computador com o programador Mastertool, e abra nele o projeto que deve ser carregado no CP redundante que utilizará esta UCP.
5. Conecte uma porta Ethernet RJ45 deste computador à porta NET1 da UCP através de um cabo ponto-a-ponto.
6. Temporariamente, altere o endereço IP e máscara de subrede desta porta do computador para que esteja na mesma subrede Ethernet configurada para a porta NET1 da UCP no projeto aberto no passo 4. Considere, por exemplo, que no projeto a porta NET1 da UCP tenha Endereço IP Ativo = 192.168.17.100, Endereço IP da UCP A = 192.168.17.101, Endereço IP da UCP B = 192.168.17.102 e Máscara de Subrede =255.255.255.0. Neste exemplo, a porta do computador pode ser ajustada temporariamente com IP = 192.168.17.10 e máscara = 255.255.255.0.
7. Temporariamente, desconecte quaisquer outras portas Ethernet deste computador, e desabilite quaisquer conexões Wifi do mesmo. Desta forma, evita-se que o Mastertool faça uma conexão indesejada a outras UCPs, através de outras portas Ethernet ou Wifi. O objetivo é que o Mastertool só consiga localizar a UCP onde o projeto aberto deve ser carregado.
8. Utilize a função *Easy Connection* do Mastertool – descrita no manual do Mastertool - para ajustar o endereço IP, a máscara de sub-rede e o endereço do gateway, para a porta NET1. Estes parâmetros devem ser iguais àqueles configurados no projeto aberto no passo 4. Existem duas possibilidades de endereços:
   a) Caso esta UCP deva ser instalada no half-cluster CPA, deve ser o *Endereço IP da UCP A* da porta NET1.
   b) Caso esta UCP deva ser instalada no half-cluster CPB, deve ser o *Endereço IP da UCP B* da porta NET1.
   c) Clique em *Device* na árvore de dispositivos, selecione a aba *Configurações de Comunicação*, e finalmente no botão *Mapear Rede*.
9. Deve aparecer uma lista com uma única UCP detectada. Selecione com duplo-clique esta UCP para definir um caminho ativo até a mesma.

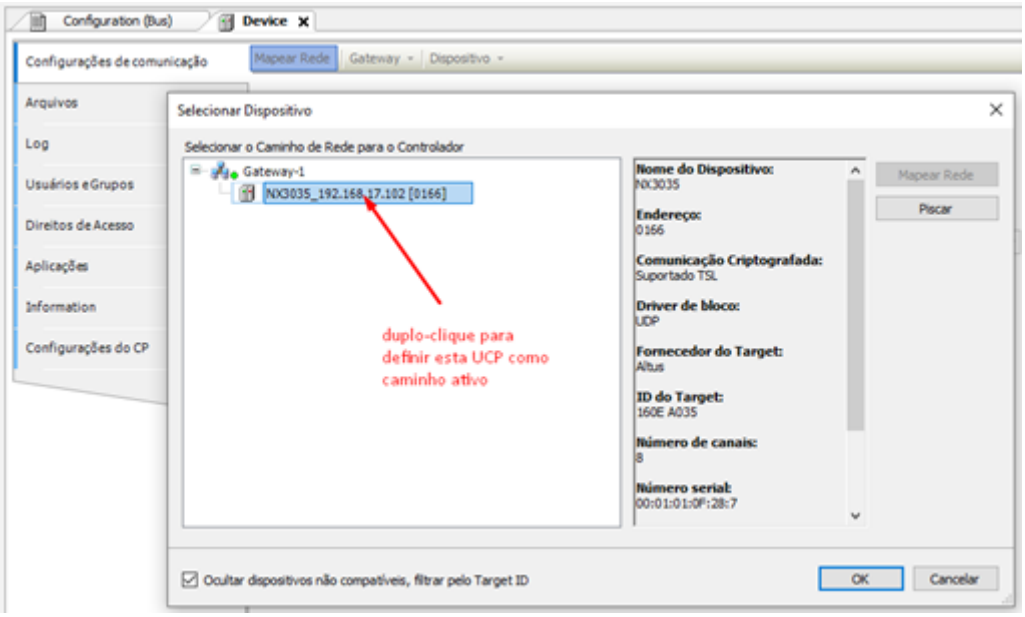


Figura 221: Selecionando o Dispositivo

10. Neste momento, aconselha-se dar um nome significativo para esta UCP, por exemplo, GENERATOR_B para o half-cluster CPB do cluster GENERATOR. Isto pode ser feito em *Device*, aba *Configurações de Comunicação*, *Dispositivo*, *Renomear o dispositivo ativo*.

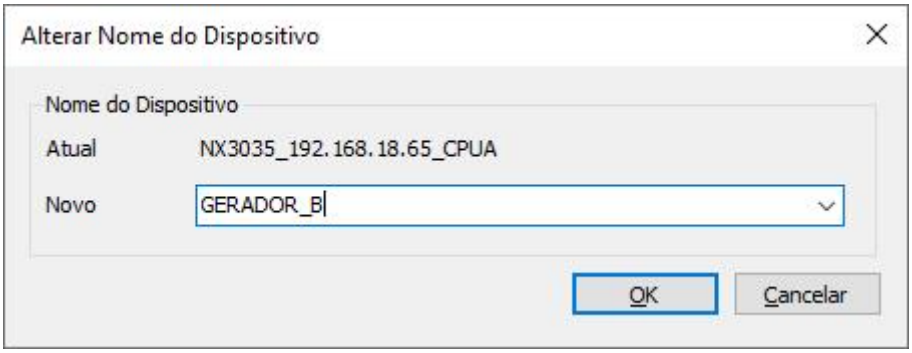


Figura 222: Alterando Nome do Dispositivo

11. Através do menu *Comunicação*, *Configuração de Redundância*, deve-se ajustar a identificação do CP, que num CP redundante pode ser CPA ou CPB. Depois deve-se pressionar o botão *Escrever*. Isso provocará uma reinicialização da UCP.

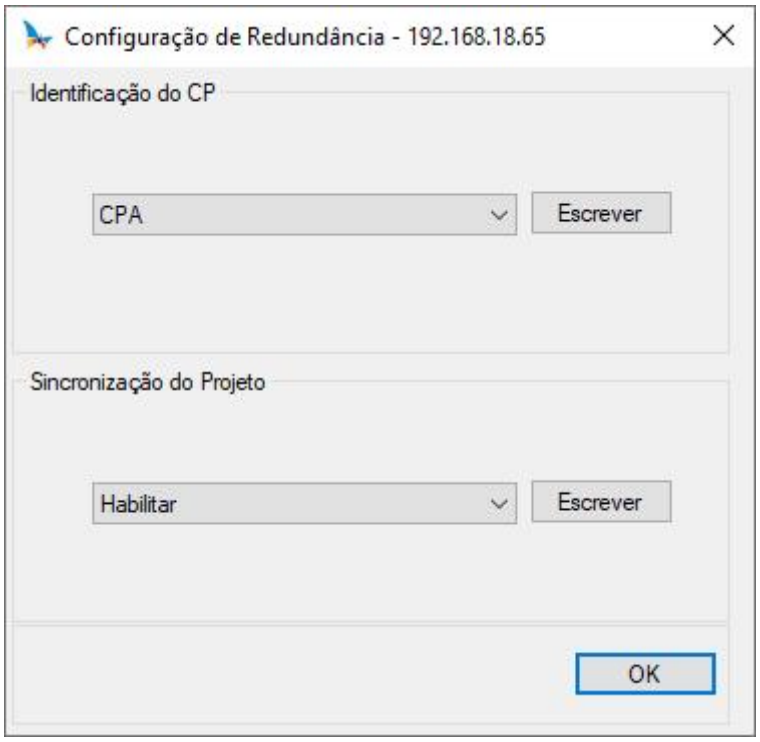


Figura 223: Configuração de Redundância

12. Depois que a UCP reinicializar, use o botão OTD e visor do NX3035 para conferir:
    a) Se o endereço IP está correto, no menu REDE.
    b) Se a identificação do CP (CPA ou CPB) está correta, no menu REDUNDANCIA.
13. Se as conferências do passo anterior estiverem corretas, é possível prosseguir. Caso contrário reinicie este procedimento desde o passo 1.
14. Se necessário, realize as configurações pertinentes a esse CP na página web do produto.
15. Defina o caminho ativo do Mastertool para este CP, repetindo o procedimento descrito no passo 9.
16. Utilize o menu *Comunicação / Login* para carregar o projeto no CP.

Depois de executado este procedimento nas UCPs do CPA e CPB, lembre-se de desfazer as alterações temporárias realizadas no computador, nos passos 6 e 7.

6.5.2. Conexão do Mastertool com uma UCP NX3035 de um CP Redundante

Depois de executar o procedimento descrito na seção Carga Inicial de um Projeto Redundante nos dois CPs (CPA e CPB), a conexão ao Mastertool, através da interface NET 1 da UCP, poderá ser feita através de um dos seguintes endereços:

- Endereço IP da UCP A: endereço IP da NET 1 exclusivo para a UCP instalada no CPA
- Endereço IP da UCP B: endereço IP da NET 1 exclusivo para a UCP instalada no CPB

Para conectar-se a determinado CP, em primeiro lugar deve-se dar um duplo clique sobre o *Device* (NX3035) na árvore de dispositivos, entrar na aba *Configurações de Comunicação*, e pressionar o botão *Mapear Rede* para listar todos os CPs detectados pelo Mastertool na rede. A figura seguinte mostra um exemplo de uma lista deste tipo.

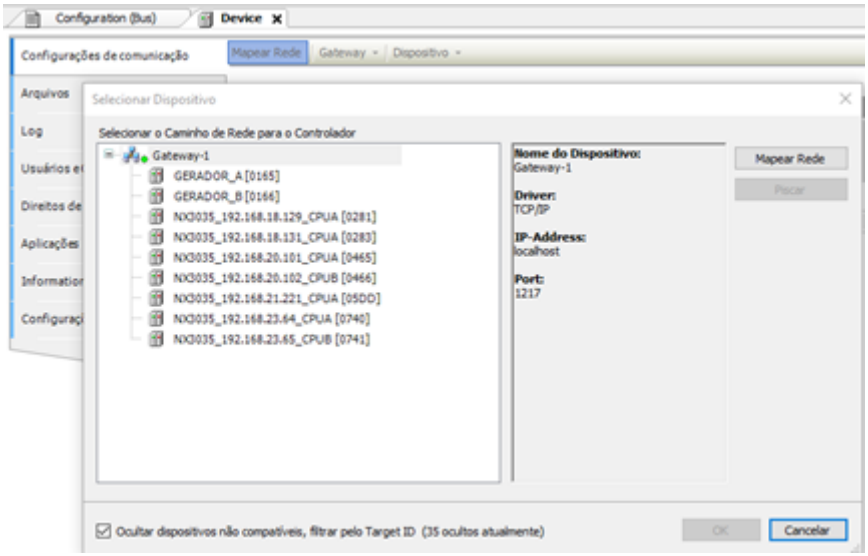


Figura 224: UCPs NX3035 detectadas na rede

O CP correto nesta lista pode ser localizado através do nome significativo que foi atribuído anteriormente ao mesmo (por exemplo: GENERATOR_A – ver passo 10 da seção Carga Inicial de um Projeto Redundante).

Em caso de dúvida quanto à UCP correta nesta lista, que pode ocorrer se um nome significativo ainda não foi atribuído ao CP, pode-se selecionar o CP mais provável com um clique simples, e pressionar o botão *Piscar*. Isso provocará o piscamento do visor da UCP no CP selecionado, permitindo ao usuário descobrir se selecionou o CP correto.

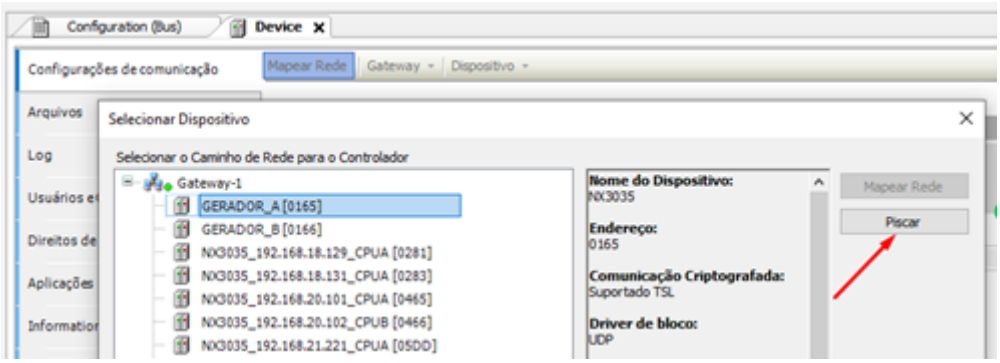


Figura 225: Uso do botão Piscar para verificar a UCP selecionada

Com um duplo clique sobre o CP selecionado na lista, define-se um caminho ativo que conecta o Mastertool a este CP. O caminho ativo serve para que o Mastertool se conecte a um CP para executar diversas funções. A lista seguinte mostra apenas algumas destas funções:

- Fazer um login ao CP. Ao fazer isso, se o projeto no Mastertool for diferente do projeto no CP, o usuário terá a opção de fazer a carga do projeto.

- Dar um nome significativo para o CP. Caso isto ainda não tenha sido feito, recomenda-se fazê-lo o mais breve possível (ver passo 10 da seção Carga Inicial de um Projeto Redundante).
- Executar comandos do menu *Comunicação*, *Configuração de Redundância*, tais como:
  - Alterar a identificação do CP (CPA, CPB ou Não-Redundante)
  - Habilitar a sincronização de projetos

ATENÇÃO

O Mastertool só consegue se conectar a um CP de cada vez (só existe um caminho ativo no Mastertool em determinado instante). Para conectar-se a diversos CPs, deve-se abrir múltiplas instâncias do Mastertool, cuidando sempre para abrir o projeto correto em cada instância. Não é possível abrir o mesmo projeto com duas instâncias diferentes do Mastertool. Caso isto seja desejado (por exemplo, uma instância conectada ao CPA e outra ao CPB para efeito de depuração), deve-se criar uma cópia adicional do projeto no computador, para que cada instância do Mastertool abra um projeto diferente.

6.5.3. Carga de Modificações em um Projeto Redundante

Depois que os dois CPs (CPA e CPB) que formam um CP redundante já receberam uma carga inicial, conforme descrito na seção Carga Inicial de um Projeto Redundante, é possível carregar modificações sucessivas do projeto, à medida que tais modificações forem necessárias.

A conexão do Mastertool a um CP para executar a carga de modificações deve ser feita conforme descreve-se na seção Conexão do Mastertool com uma UCP NX3035 de um CP Redundante. Nesta seção explica-se como é possível conectar-se a um CP específico (CPA ou CPB).

Normalmente as modificações devem ser carregadas no CP Ativo, e a seguir serão automaticamente sincronizadas para o CP Não-Ativo, através dos canais de sincronismo NETA/NETB. Portanto, normalmente o usuário deve descobrir qual dos CPs está em estado Ativo, e fazer login no mesmo para carregar modificações de um projeto. O estado de redundância pode ser descoberto através do visor da UCP, ou através de diagnósticos que podem estar sendo mostrados no sistema SCADA, por exemplo.

ATENÇÃO

Existem algumas situações em que pode ser necessário carregar modificações de projeto no CP Não-Ativo. Tais situações são descritas no capítulo Carga Offline de Programas sem Interrupção do Controle do Processo.

6.5.4. Carga de Modificações Offline e Online

Modificações de projeto podem ser carregadas offline ou online.

Cargas offline requerem a parada do CP onde a modificação deve ser carregada. Por outro lado, cargas online permitem que o CP continue executando sua aplicação enquanto a modificação é carregada.

Alguns tipos de modificações exigem carga offline, ou seja, não podem ser carregadas online no CP onde o Mastertool está conectado. Neste caso, existem duas opções:

- Interromper o controle do processo, executando o procedimento descrito na seção Carga Offline de Modificações com Interrupção do Controle do Processo.
- Valer-se da redundância do CP para não interromper o controle do processo, mesmo com a necessidade de fazer cargas offline em cada um dos half-clusters (CPA ou CPB). Os procedimentos para este caso são descritos no capítulo Carga Offline de Programas sem Interrupção do Controle do Processo.

6.5.4.1. Modificações que Demandam Carga Offline com Interrupção do Controle do Processo

As seguintes modificações em um projeto farão com o que o mesmo se torne impossível de ser carregado num sistema redundante sem a interrupção do controle do processo.

- Modificações nas áreas de memória redundante para variáveis de representação direta (%Q, %I e %M), descritas na seção Configurações da UCP NX3035 Relacionadas com Redundância.

ATENÇÃO

Modificações deste tipo impedem a execução dos procedimentos descritos no capítulo Carga Offline de Programas sem Interrupção do Controle do Processo. Para evitar esta situação, a configuração destas áreas deve ser cuidadosamente planejada com boa reserva de expansão.

6.5.4.2. Modificações que Demandam Carga Offline

As seguintes modificações demandam carga offline no CP, mas não impedem a execução dos procedimentos descritos no capítulo Carga Offline de Programas sem Interrupção do Controle do Processo:

- Adicionar ou remover dispositivos na árvore de dispositivos, por exemplo:
  - Módulos no bastidor principal (mestres PROFIBUS NX5001, interfaces Ethernet NX5000, etc.)
  - Remotas em redes PROFIBUS
  - Módulos de E/S em remotas da rede PROFIBUS da Série Ponto
  - Adicionar módulos de E/S em remotas da rede PROFIBUS da Série Nexto
  - Instâncias MODBUS
- Modificar parâmetros dentro de dispositivos existentes na árvore de dispositivos, por exemplo:
  - Endereços IP e outros parâmetros de interfaces Ethernet
  - Parâmetros de mestres PROFIBUS
  - Parâmetros de remotas PROFIBUS da Série Ponto
  - Parâmetros de módulos de E/S dentro de remotas da rede PROFIBUS da Série Ponto ou Série Nexto
- Modificações nas configurações das tarefas.
- Atualização do projeto devido à atualização do programador Mastertool.
- Atualizações de firmware da UCP NX3035, desde que a nova versão continue compatível com a anterior (exemplo: versão 1.a.b.c para versão 1.e.f.g).

6.5.4.3. Modificações que Permitem Carga Online

Em princípio, as modificações não citadas nas seções Modificações que Demandam Carga Offline com Interrupção do Controle do Processo e Modificações que Demandam Carga Offline, permitem carga online.

Mesmo assim, a seguir serão citadas as principais modificações que permitem carga online no CP onde o Mastertool está conectado. As modificações citadas abaixo valem para variáveis, POUs e GVLs, redundantes ou não.

- Adicionar POUs do tipo programa, desde que estas POUs não precisem ser associadas à alguma tarefa e não estejam marcadas como redundante.
- Remover POUs do tipo programa, desde que estas POUs não estejam associadas à alguma tarefa.
- Adicionar ou remover POUs do tipo função ou bloco funcional.
- Modificar o código de quaisquer tipos de POU (programa, função ou bloco funcional).
- Adicionar ou remover variáveis simbólicas em quaisquer tipos de POU (programa, função ou bloco funcional), sendo elas redundantes ou não.
- Adicionar ou remover instâncias de bloco funcional em POUs do tipo programa ou bloco funcional.
- Adicionar ou remover GVLs.
- Adicionar ou remover variáveis simbólicas ou instâncias de bloco funcional em GVLs.

6.5.5. Carga Online de Modificações

Uma carga online deve ser feita conectando o MasterTool a alguma interface do CP Ativo. Portanto, o usuário deve descobrir qual dos CPs está em estado Ativo, e fazer login no mesmo para carregar modificações online de um projeto. O estado de redundância pode ser descoberto através do visor da UCP, ou através de diagnósticos que podem estar sendo mostrados no sistema SCADA.

O procedimento de conexão a um CP foi descrito na seção Conexão do Mastertool com uma UCP NX3035 de um CP Redundante.

Logo a seguir, o usuário deve utilizar o menu *Comunicação / Login*. Por exemplo, se o Mastertool estiver conectado ao CP GENERATOR_A, a seguinte sequência de mensagens será apresentada:

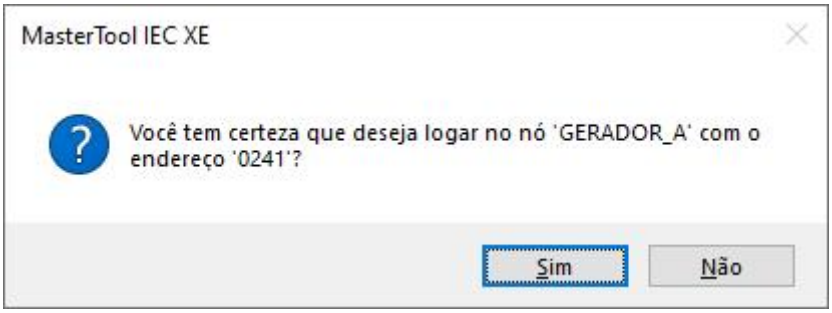


Figura 226: Mensagem de Confirmação do Dispositivo

- Pressionar o botão *Sim*.

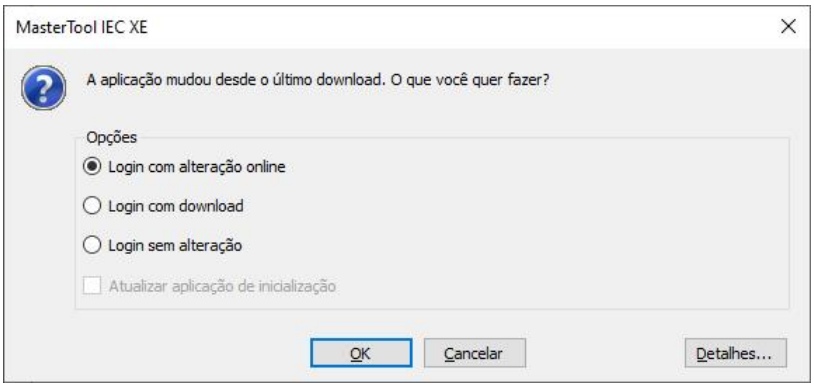


Figura 227: Mensagem Opção de Download

- Selecionar *Login com alteração online* e depois o botão *OK*.

Depois que a carga online termina no CP Ativo, quando o CP Não-Ativo percebe que possui um projeto diferente do CP Ativo, ele executa as seguintes ações:

- Negocia uma sincronização automática de projeto com o CP Ativo.
- Caso esteja no estado Reserva ou Inicializando, chaveia para o estado Não-Configurado, e permanece neste estado até que os projetos estejam novamente sincronizados. Depois disso, volta automaticamente ao estado Reserva.
- Caso esteja no estado Inativo, um comando deve ser executado para solicitar que o CP chaveie para o estado Reserva. Quando isso ocorrer, o CP vai primeiro para o estado Não-Configurado, sincroniza o projeto, e finalmente tenta ir para o estado Reserva. O estado Reserva será atingido se nenhum diagnóstico crítico impedir isso.

6.5.6. Carga Offline de Modificações com Interrupção do Controle do Processo

Nesta seção, define-se o procedimento para executar uma carga offline que interromperá o controle do processo. Tal situação é aceitável em determinados tipos de processos e durante paradas programadas deste processo.

Uma carga offline deve ser feita conectando o Mastertool ao canal NET 1 do CP Ativo. Portanto, o usuário deve descobrir qual dos CPs está em estado Ativo, e fazer login no mesmo para carregar modificações online de um projeto. O estado de redundância pode ser descoberto através do visor da UCP, ou através de diagnósticos que podem estar sendo mostrados no sistema SCADA.

O procedimento de conexão a um CP foi descrito na seção Conexão do Mastertool com uma UCP NX3035 de um CP Redundante.

Logo a seguir, o usuário deve utilizar o menu *Comunicação / Login*. Por exemplo, se o Mastertool estiver conectado ao CP GENERATOR_A, a seguinte sequência de mensagens será apresentada:

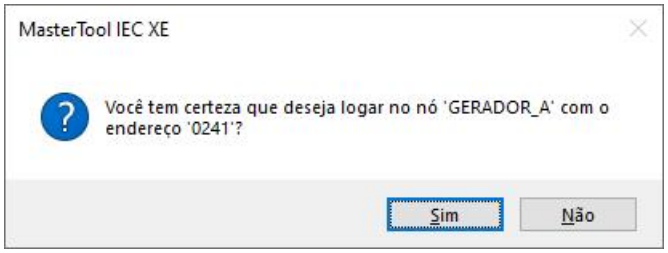


Figura 228: Mensagem Confirmação de Login

- Responder *Sim*.

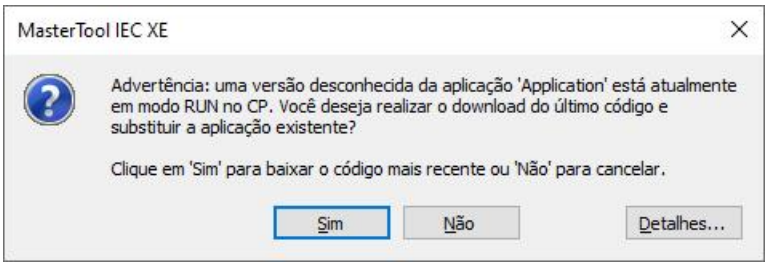


Figura 229: Mensagem Aplicação Desconhecida

- Responder *Sim*.
- Depois disso, se o outro CP estiver em estado Reserva, a seguinte mensagem vai aparecer:

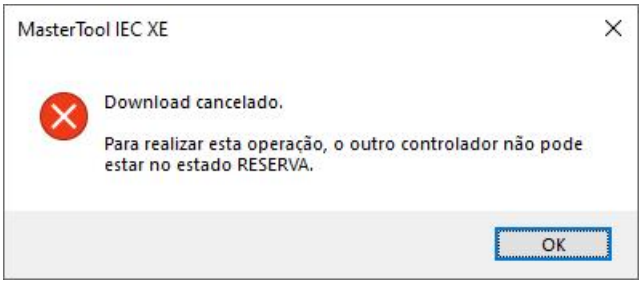


Figura 230: Mensagem Download Cancelado

Caso a mensagem anterior tenha aparecido, modifique o estado do CP Reserva para Inativo, e reinicie o processo de carga offine.

Caso a mensagem anterior não tenha aparecido, a carga off-line é realizada, e o CP Ativo vai para o estado Não-Configurado em modo STOP, interrompendo o controle do processo.

Depois disso:

1. Deve-se reiniciar a aplicação deste CP colocando-o no modo RUN. Depois de algum tempo inicializando, este CP deve voltar ao estado Ativo.
2. Caso o outro CP esteja em estado Inativo, deve-se solicitar que o mesmo vá para o estado Reserva. Neste momento, ocorrerá uma sincronização automática de projeto, e quando a mesma estiver concluída, o CP irá para o estado Reserva.

6.6. Manutenção da Redundância

6.6.1. Troca a Quente de Módulos em um CP Redundante

Em caso de falha em algum módulo de um dos CPs (CPA ou CPB), pode ser necessária uma troca do módulo, sem interromper o controle do processo. Para isso, os seguintes passos devem ser seguidos:

1. Coloque o outro CP, que não sofrerá manutenção, no estado Ativo, se ele já não estiver neste estado.
2. Coloque o CP que sofrerá manutenção no estado Inativo.
3. Efetue as trocas a quente necessárias no CP Inativo, conforme indicado no capítulo Configuração da UCP - Parâmetros Gerais - Como realizar a Troca a Quente.
4. No final, volte o CP Inativo ao estado Reserva.
5. Se desejado, troque este CP Reserva para o estado Ativo.

6.6.2. Mensagens de Advertência do Mastertool

Quando o Mastertool está com um projeto redundante aberto, ou quando está conectado a uma UCP NX3035 identificada como CPA ou CPB, algumas mensagens de advertência especiais poderão ocorrer, conforme descreve-se nas próximas subseções.

6.6.2.1. Bloqueio de Carga de um Projeto Redundante ou Não-Redundante

Em algumas situações, o Mastertool bloquerá a carga de um projeto no CP em que estiver conectado através do caminho ativo:

- A carga de um projeto redundante será bloqueada se modelo da UCP não for NX3035 ou esta UCP não estiver identificada como CPA ou CPB (ver seção Identificação de uma UCP NX3035).
- A carga de um projeto não-redundante será bloqueada se esta UCP estiver identificada como CPA ou CPB (ver seção Identificação de uma UCP NX3035).
- A carga de um projeto redundante será bloqueada se a outra UCP redundante estiver em estado Reserva.

Caso ocorra uma tentativa de executar alguma destas ações ilegais, o Mastertool advertirá com uma mensagem apropriada.

6.6.2.2. Alertas antes de Comandos que Podem Parar o CP Ativo

Os seguintes comandos podem parar um CP:

- Colocar a UCP em modo STOP
- Depurar / Parar
- Comunicação / Reset (a Quente, a Frio, Origem)

Caso o Mastertool esteja logado ao CP ativo e ocorra uma tentativa de execução de um destes comandos, antes de enviá-los, o MasterTool envia a seguinte mensagem e aguarda autorização para enviar o comando (OK) ou para cancelá-lo (Cancelar):

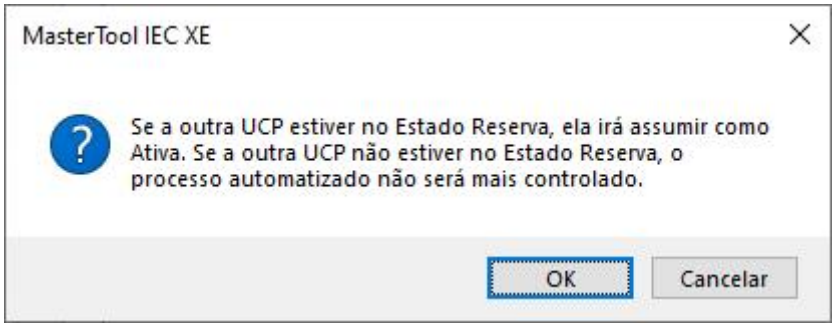


Figura 231: Alerta de parada do CP Ativo

Um CP também pode parar caso o usuário tente uma carga offline de projeto no CP Ativo, e o outro CP não esteja em estado Reserva. Neste caso, o Mastertool envia a seguinte mensagem e aguarda autorização para fazer a carga off-line (Sim) ou para cancelá-lo (Não):

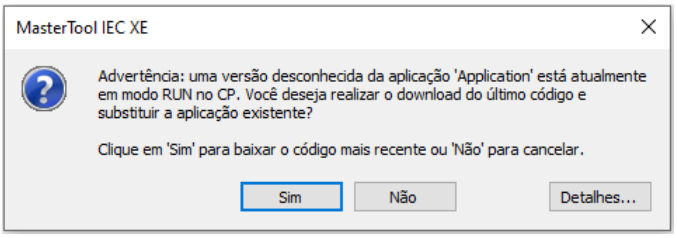


Figura 232: Alerta de parada do CP Ativo após carga offline

Outra situação que pode parar o CP é a ativação de um breakpoint no CP Ativo. Neste caso, a ativação do breakpoint no CP Ativo deve ser bloqueada caso o outro CP esteja em estado Reserva, conforme descrito na seção Utilização de Breakpoints em Sistemas Redundantes.

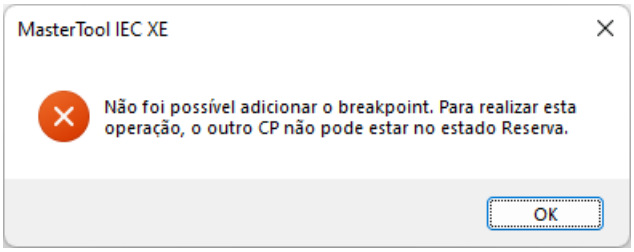


Figura 233: Bloqueio de Ativação de Breakpoint

6.6.2.3. Alerta antes de Logar-se ao CP Não-Ativo

Em circunstâncias normais, não é usual o Mastertool logar-se ao CP Não-Ativo. Quando houver uma tentativa de execução de um comando deste tipo, o Mastertool emite o seguinte aviso para confirmar a operação (OK) ou cancelá-la (Cancelar).

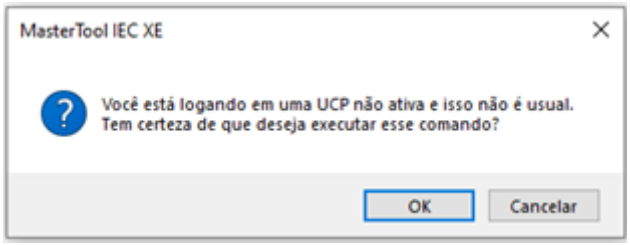


Figura 234: Aviso de tentativa de login no CP Não-Ativo

Existem circunstâncias (não muito usuais) em que é necessário logar-se ao CP Não-Ativo, e nestes casos o usuário deve autorizar o login. Tais circunstâncias podem ocorrer, por exemplo:

- Para configurações iniciais, conforme descrito na seção Carga Inicial de um Projeto Redundante.
- Para carregar offline um projeto diferente no CP Não-Ativo, conforme descrito no capítulo Carga Offline de Programas sem Interrupção do Controle do Processo.
- Para monitorar ou forçar variáveis não-redundantes no CP Não-Ativo.

6.6.3. Diagnósticos da Redundância no Visor Gráfico da UCP NX3035

Diversos diagnósticos relativos à redundância podem ser observados no visor da UCP NX3035.

6.6.3.1. Estado de Redundância do CP

O estado de redundância do CP, descrito em Estados de um CP Redundante, é visto nos três caracteres iniciais da segunda linha da tela principal, conforme mostra a seção Visor Gráfico. A tela principal do visor é apresentada na inicialização, e volta a ser apresentada alguns segundos depois de terminada a navegação (sem apertar a tecla da UCP NX3035).

6.6.3.2. Telas abaixo do Menu REDUNDANCIA

Existe um menu denominado REDUNDANCIA, abaixo do qual existem diversas telas. A descrição e o acesso às telas de redundância estão disponíveis na seção Menu Informativo e de Configuração da UCP do capítulo Configuração.

6.6.4. Estrutura de Diagnósticos, Comandos e Informações do Usuário para a Redundância

A área de diagnósticos, comandos e informações do usuário para a redundância é mapeada sobre variáveis de representação direta %Q, definidas simbolicamente através de diretivas AT, na GVL *System_Diagnostics*.

A estrutura de dados *DG_NX3035.tRedundancy* contém diagnósticos, comandos e informações do usuário específicos da redundância, os quais serão descritos no decorrer desta seção e de suas subseções. Essa estrutura é dividida nas 6 seguintes subestruturas de dados:

- **RedDgnLoc:** Contém diagnósticos de redundância referentes ao CP local, por exemplo, o estado da redundância do CP local. Essa estrutura é descrita na seção Diagnósticos da Redundância.
- **RedDgnRem:** Contém diagnósticos de redundância referentes ao CP remoto, por exemplo, o estado da redundância do CP remoto.É uma cópia de RedDgnLoc do outro CP, recebida via canais de sincronismo NETA / NETB. Desta forma, o CP local tem acesso aos diagnósticos do CP remoto. Essa estrutura é descrita na seção Diagnósticos da Redundância.
- **RedCmdLoc:** Contém comandos de redundância que podem ser escritos no CP local, por exemplo, através de escritas a partir de um sistema SCADA, ou gerados em POUs executadas neste CP (ex: ActivePrg ou NonSkippedPrg). Essa estrutura é descrita na seção Comandos da Redundância.
- **RedCmdRem:** Trata-se de uma cópia de RedCmdLoc do CP remoto, recebida via canais de sincronismo NETA / NETB. No CP local, esta estrutura só pode ser lida (tentativas de escrita na mesma são descartadas). Alguns dos comandos recebidos do CP remoto poderão ser executados no CP local. Essa estrutura é descrita na seção Comandos da Redundância.
- **RedUsrLoc:** Contém informações de livre escolha do usuário que podem ser trocadas entre CPA e CPB. No CP local, essa estrutura pode ser escrita para transmitir informações para o CP remoto. Essa estrutura é descrita na seção Informações do Usuário Trocadas entre CPs Local e Remoto.
- **RedUsrRem:** Trata-se de uma cópia de RedUsrLoc do CP remoto, recebida via canais de sincronismo NETA / NETB. No CP local, esta estrutura só pode ser lida (tentativas de escrita na mesma são descartadas). As informações lidas no CP local via RedUsrRem são as mesmas escritas na estrutura RedUsrLoc do CP remoto. Essa estrutura é descrita na seção Informações do Usuário Trocadas entre CPs Local e Remoto.

É importante ressaltar que as estruturas copiadas do CP remoto são atualizadas somente quando termina, com sucesso, o serviço de Troca de Diagnósticos e Comandos, descrito na seção Serviços de Sincronização Cíclicos através de NETA e NETB. Esse serviço é executado a cada ciclo da tarefa MainTask. Em caso de falha neste serviço:

- Diagnósticos do CP remoto poderão estar com valores obsoletos no CP local.
- Comandos enviados para o CP remoto poderão não ser executados no CP remoto.
- Informações de usuário não serão trocadas entre os CP local e remoto.

As próximas subseções contêm tabelas que definem com detalhes cada uma das estruturas de dados de diagnósticos, comandos e informações de usuários. As seguintes observações devem ser consideradas ao analisar estas tabelas:

- Para ter uma visão detalhada dos tipos de dados de cada variável, deve-se acessar o o item *Library Manager* na árvore de dispositivos da janela do Mastertool. Dentro do *Library Manager*, tais tipos podem ser encontrados na biblioteca *IoDrvNextoRedundancy*.
- As variáveis são de representação direta (%Q), mas possuem nomes simbólicos atribuídos através de diretivas AT na GVL *System_Diagnostics*.
- O endereço %Q(n) corresponde ao endereço inicial da estrutura de dados *DG_NX3035.tRedundancy*.

6.6.4.1. Diagnósticos da Redundância

Os diagnósticos da redundância estão nas seguintes estruturas:

- **DG_NX3035.tRedundancy.RedDgnLoc:** diagnósticos do CP local
- **DG_NX3035.tRedundancy.RedDgnRem:** diagnósticos do CP remoto

Estes diagnósticos têm várias utilidades, por exemplo:

- Podem ser consultados para verificar a existência de algum problema que precisa ser sanado.
- Toda vez que ocorrem variações nos mesmos, tais variações são inseridas como eventos nos logs de usuário. Consultando a sequência histórica de tais eventos pode-se descobrir, por exemplo, a causa de um switch-over.

- Podem ser referenciados na aplicação do usuário (ActivePrg ou NonSkippedPrg). Por exemplo, pode-se testar o estado de redundância do CP e tomar ações específicas em função deste estado.

> **ATENÇÃO**
>
> O diagnóstico *DG_NX3035.tRedundancy.RedDgnLoc.sGeneral_Diag.bExchangeSync* informa se o serviço de Troca de Diagnósticos e Comandos foi concluído com sucesso, caso tenha o valor TRUE. Caso o valor deste diagnóstico seja FALSE, diagnósticos do CP remoto não foram copiados e podem estar com valores obsoletos.

Como RedDgnRem é uma cópia de RedDgnLoc do CP remoto, as duas estruturas têm o mesmo formato.

Estas estruturas são divididas em cinco subestruturas:

- **sGeneral_Diag:** Diagnósticos gerais da redundância.
- **sNETA_Diag:** Diagnósticos do canal de sincronismo NETA.
- **sNETB_Diag:** Diagnósticos do canal de sincronismo NETB.
- **sNET_Stat:** Estatísticas comuns para os canais de sincronismo NETA e NETB, para contagem de sucessos e falhas dos serviços de sincronização.
- **sGeneral_DiagExt:** Diagnósticos gerais da redundância, parte estendida (continuação de *sGeneral_Diag*).

A tabela seguinte detalha a subestrutura *sGeneral_Diag*.

Variável AT DG_NX3035.tRedundancy.RedDgnLoc.sGeneral_Diag.*	**Descrição**
bConfigDone	TRUE – O processo de configuração, executado no estado Não-Configurado, terminou.
	FALSE – O processo de configuração, executado no estado Não-Configurado, ainda não terminou ou não foi executado.
bConfigError	TRUE – O processo de configuração, executado no estado Não-Configurado, terminou com erros. Trata-se de um erro de sistema, normalmente não esperado. Entre em contato com o suporte da Altus para reportá-lo. Informe também o valor do diagnóstico *wConfigErrorCode* para o suporte da Altus.
	FALSE – O processo de configuração ocorreu com sucesso ou não foi realizado.
bTooManyRedAreas	TRUE – O número de áreas redundantes excedeu o número máximo permitido. Trata-se de um erro de sistema, normalmente não esperado. Entre em contato com o suporte da Altus para reportá-lo.
	FALSE – O número de áreas redundantes está dentro do esperado.
bTemporaryBufferTooSmall	TRUE – Estrutura de dados intermediária com tamanho insuficiente. Trata-se de um erro de sistema, normalmente não esperado. Entre em contato com o suporte da Altus para reportá-lo.
	FALSE – O tamanho da estrutura de dados está dentro do esperado.
bExchangeSync	TRUE – O serviço de sincronização Troca de Diagnósticos e Comandos foi executado com sucesso neste ciclo da MainTask.
	FALSE – O serviço de sincronização Troca de Diagnósticos e Comandos não foi executado com sucesso neste ciclo da MainTask. Valores inválidos ou obsoletos podem existir nas estruturas de dados e comandos copiados do CP remoto.
bRedDataSync	TRUE – O serviço Sincronização de Dados Redundantes foi executado com sucesso neste ciclo da MainTask.
	FALSE – O serviço Sincronização de Dados Redundantes não foi executado com sucesso neste ciclo da MainTask.
bRedForceSync	TRUE – O serviço Sincronização da Lista de Forçamentos Redundantes foi executado com sucesso neste ciclo da MainTask.
	FALSE – O serviço Sincronização da Lista de Forçamentos Redundantes não foi executado com sucesso neste ciclo da MainTask.
bApplicationIncompatible	TRUE – A aplicação não é compatível entre os dois CPs. Foi efetuado o download de uma nova aplicação para um dos CPs com uma das seguintes alterações: - Modificação da área de dados redundantes; - Modificação de variáveis simbólicas redundantes. Enquanto este diagnóstico estiver ligado, um dos CPs ficará em estado Inativo até que a mesma aplicação esteja presente nos dois CPs. Isto implica em recarregar a aplicação antiga nos dois CPs ou atualizar os dois CPs com a nova aplicação. Para mais informações sobre como proceder, consultar capítulo Carga de Programas em um CP Redundante.
	FALSE – A aplicação é compatível entre os dois CPs.

Variável AT DG_NX3035.tRedundancy.RedDgnLoc.sGeneral_Diag.*	Descrição
bProjectSyncDisable	TRUE – Não será efetuado o sincronismo do projeto de aplicação e do *Project archive* entre os dois CPs. Trata-se de uma cópia da variável não volátil utilizada para desabilitar a sincronização de projetos, conforme descrito na seção Desabilitação da Sincronização de Projetos. A sincronização de projetos pode ser desabilitada tanto no CP local como no remoto, ou seja, basta executar o comando de desabilitação em um dos CPs para que a sincronização de projetos seja desabilitada. Os comandos de desabilitação do sincronismo de projetos são descritos na seção Desabilitação da Sincronização de Projetos.
	FALSE – Será efetuado o sincronismo do projeto de aplicação e do *Project archive* entre os dois CPs.
bIncompatibleFirmware	TRUE – A versão de firmware é incompatível entre este CP e o CP remoto, ou há uma falha nos links de redundância.
	FALSE – A versão de firmware é compatível entre este CP e o CP remoto.
bApplicationProjectDiff	TRUE – O projeto de aplicação deste CP é diferente do projeto de aplicação do CP remoto, ou há uma falha nos links de redundância.
	FALSE – O projeto de aplicação deste CP é igual ao do CP remoto.
bProjectArchiveDiff	TRUE – O *Project archive* deste CP é diferente do presente no CP remoto, ou há uma falha nos links de redundância.
	FALSE – O *Project archive* deste CP é igual ao do CP remoto.
bOnlineChangeApply	TRUE – Foi realizada alguma alteração online na aplicação e esta ainda não foi sincronizada com o CP reserva.
	FALSE – Não foram realizadas alterações online na aplicação ou estas já foram sincronizadas com o CP reserva.
bFailedRED	TRUE – Falha interna de hardware desta UCP que impede o funcionamento das duas redes de sincronismo (NETA e NETB).
	FALSE – Ausência de falha interna de hardware desta UCP que impede o funcionamento das duas redes de sincronismo (NETA e NETB).
bFailedPBUS1A	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) na rede PROFIBUS 1A. O modo ativo (comunicando com escravos) é assumido pelo CP Ativo. O modo passivo (comunicando com o mestre em modo ativo) é assumido pelo CP Não-Ativo. Esta falha também pode ser indicada caso o módulo NX5001 tenha falha em seu microprocessador, ou caso não consiga se comunicar com a UCP via barramento.
	FALSE – Não existem falhas na rede PROFIBUS 1A.
bFailedPBUS1B	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) na rede PROFIBUS 1B. O modo ativo (comunicando com escravos) é assumido pelo CP Ativo. O modo passivo (comunicando com o mestre em modo ativo) é assumido pelo CP Não-Ativo. Esta falha também pode ser indicada caso o módulo NX5001 tenha falha em seu microprocessador, ou caso não consiga se comunicar com a UCP via barramento.
	FALSE – Não existem falhas na rede PROFIBUS 1B.
bFailureProfibus_1	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) na rede PROFIBUS 1. Caso a rede PROFIBUS 1 seja redundante, *bFailureProfibus_1* resulta de um "E lógico" entre *bFailedPBUS1A* e *bFailedPBUS1B*. Caso a rede PROFIBUS 1 não seja redundante, *bFailureProfibus_1* é uma cópia de *bFailedPBUS1A*.
	FALSE – Não existem falhas na rede PROFIBUS 1.
bFailedPBUS2A	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) na rede PROFIBUS 2A. O modo ativo (comunicando com escravos) é assumido pelo CP Ativo. O modo passivo (comunicando com o mestre em modo ativo) é assumido pelo CP Não-Ativo. Esta falha também pode ser indicada caso o módulo NX5001 tenha falha em seu microprocessador, ou caso não consiga se comunicar com a UCP via barramento.
	FALSE – Não existem falhas na rede PROFIBUS 2A.
bFailedPBUS2B	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) na rede PROFIBUS 2B. O modo ativo (comunicando com escravos) é assumido pelo CP Ativo. O modo passivo (comunicando com o mestre em modo ativo) é assumido pelo CP Não-Ativo. Esta falha também pode ser indicada caso o módulo NX5001 tenha falha em seu microprocessador, ou caso não consiga se comunicar com a UCP via barramento.
	FALSE – Não existem falhas na rede PROFIBUS 2B.
bFailureProfibus_2	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) na rede PROFIBUS 2. Caso a rede PROFIBUS 2 seja redundante, *bFailureProfibus_2* resulta de um "E lógico" entre *bFailedPBUS2A* e *bFailedPBUS2B*. Caso a rede PROFIBUS 2 não seja redundante, *bFailureProfibus_2* é uma cópia de *bFailedPBUS2A*.
	FALSE – Não existem falhas na rede PROFIBUS 2.

Variável AT DG_NX3035.tRedundancy.RedDgnLoc.sGeneral_Diag.*	Descrição
bProfibusVitalFailureAny	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) com pelo menos uma das redes PROFIBUS configuradas em modo de falha vital.
	FALSE – Não existem falhas nas redes PROFIBUS configuradas em modo de falha vital.
bProfibusVitalFailureAll	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) com todas as redes PROFIBUS configuradas em modo de falha vital.
	FALSE – Pelo menos uma das redes PROFIBUS configuradas em modo de falha vital não possui falhas.
ePLC_ID	Este diagnóstico informa a identificação deste CP: - 0 = não-redundante - 2 = PLC_A - 3 = PLC_B Trata-se de uma cópia da variável não-volátil utilizada para identificar o CP, conforme descrito na seção Identificação de uma UCP NX3035. Na seção Carga Inicial de um Projeto Redundante descreve-se o comando do Mastertool utilizado para escrever sobre esta variável não-volátil.
eRedState	Informa o estado de redundância deste CP: - Não-Configurado = 0 - Inicializando = 2 - Reserva = 3 - Ativo = 4 - Inativo = 5
ePreviousRedState	Valor que o diagnóstico RedState possuía antes da última transição de estados.
wRedStateDuration	Mede há quanto tempo (milissegundos) o estado de redundância atual foi assumido. Este tempo para de incrementar quanto atinge 65535 ms.
wConfigErrorCode	Código de erro descoberto durante o processo de configuração no estado Não-Configurado. Ver diagnóstico *bConfigError* descrito anteriormente.
dwApplicationCRC	CRC de 32 bits da aplicação, utilizado para detectar diferenças entre as aplicações dos 2 CPs.
dwArchiveCRC	CRC de 32 bits do *project archive*, utilizado para detectar diferenças entre os *project archives* dos 2 CPs.
dwFirmwareVersion	Versão de firmware deste CP, utilizada para verificar compatibilidade entre firmware dos 2 CPs.
dwIECTimer	A sincronização do Temporizador IEC é necessária para operação sem colisões de alguns blocos de função como TON e TOF. Através deste diagnóstico o Temporizador IEC do CP Ativo é recebido e atualizado no CP Não-Ativo, desde que o serviço Troca de Diagnósticos e Comandos tenha sido executado com sucesso. Sua contagem inicia em 0 e incrementa até 4294967295. Após estouro de contagem, reinicia com valor 0.
wCycleCounter	Contador de 16 bits utilizado como informação auxiliar de sequência nos Logs de Eventos da Redundância. No CP Ativo, é incrementado a cada ciclo da MainTask. No CP Não-Ativo, recebe uma cópia do valor existente no CP Ativo, desde que o serviço Troca de Diagnósticos e Comandos tenha sido executado com sucesso. Sua contagem inicia em 0 e incrementa até 65535. Após estouro de contagem, reinicia com valor 0.

Tabela 164: Subestrutura *sGeneral_Diag*

ATENÇÃO

Quando a variável *DG_NX3035.tRedundancy.RedDgnLoc.sGeneral_Diag.bExchangeSync* tem o valor igual a FALSE não é possível definir o estado de alguns outros diagnósticos da tabela anterior que dependem da comunicação com o CP remoto, como por exemplo *bIncompatibleFirmware*, *bApplicationProjectDiff* e *bProjectArchiveDiff*.

A tabela seguinte detalha a subestrutura *sNETA_Diag*, que contém diagnósticos para o canal de sincronismo NETA:

Variável AT DG_NX3035.tRedundancy.RedDgnLoc.sNETA_Diag.*	Descrição
bGeneralFailure	TRUE – O canal de sincronismo possui algum tipo de falha. Os 3 próximos diagnósticos indicarão a falha específica.
	FALSE – O canal de sincronismo está em correto funcionamento.
bInternalFailure	TRUE – A falha detectada tem sua causa localizada dentro deste CP. Tais falhas são tratadas de forma diferenciada.
	FALSE – Não existem falhas internas para o canal de sincronismo.
bLinkDownFailure	TRUE – Falha na conexão da fibra óptica no canal de recepção, ou defeito no módulo SFP (NX9500 ou NX9501).
	FALSE – Ausência destas falhas.
bTimeoutFailure	TRUE – Esta falha é reportada caso um serviço de sincronização não tenha terminado com sucesso até um time-out especificado, e não tenham sido encontradas falhas do tipo *bInternalFailure* ou *bLinkDownFailure* que justificassem isso.
	FALSE – Sem falhas de timeout.

Tabela 165: Subestrutura *sNETA_Diag*

tabela seguinte detalha a subestrutura *sNETB_Diag*, que contém diagnósticos para o canal de sincronismo NETB:

Variável AT DG_NX3035.tRedundancy.RedDgnLoc.sNETB_Diag.*	Descrição
bGeneralFailure	TRUE – O canal de sincronismo possui algum tipo de falha. Os 3 próximos diagnósticos indicarão a falha específica.
	FALSE – O canal de sincronismo está em correto funcionamento.
bInternalFailure	TRUE – A falha detectada tem sua causa localizada dentro deste CP. Tais falhas são tratadas de forma diferenciada.
	FALSE – Não existem falhas internas para o canal de sincronismo.
bLinkDownFailure	TRUE – Falha na conexão da fibra óptica no canal de recepção, ou defeito no módulo SFP (NX9500 ou NX9501).
	FALSE – Ausência destas falhas.
bTimeoutFailure	TRUE – Esta falha é reportada caso um serviço de sincronização não tenha terminado com sucesso até um time-out especificado, e não tenham sido encontradas falhas do tipo *bInternalFailure* ou *bLinkDownFailure* que justificassem isso.
	FALSE – Sem falhas de timeout.

Tabela 166: Subestr|utura *sNETB_Diag*

A tabela seguinte detalha a subestrutura *sNET_Stat*, que contém contadores estatísticos de falhas e sucessos para os serviços executados através dos canais de sincronismo NETA e NETB.

Variável AT DG_NX3035.tRedundancy.RedDgnLoc.sNET_Stat.*	Descrição
wSuccessExchDgCmdSync	Contagem de sucessos do serviço Troca de Diagnósticos e Comandos. (0 a 65535)
wFailedExchDgCmdSync	Contagem de falhas do serviço Troca de Diagnósticos e Comandos. (0 a 65535)
wSuccessRedDataSync	Contagem de sucessos do serviço Sincronização de Dados Redundantes. (0 a 65535)
wFailedRedDataSync	Contagem de falhas do serviço Sincronização de Dados Redundantes. (0 a 65535)
wSuccessRedForceSync	Contagem de sucessos do serviço Sincronização de Lista de Forçamentos Redundantes. (0 a 65535)
wFailedRedForceSync	Contagem de falhas do serviço Sincronização de Lista de Forçamentos Redundantes. (0 a 65535)

Tabela 167: Subestrutura *sNET_Stat*

Notas sobre os contadores da tabela anterior:

- Todos os contadores retornam ao valor zero quando o seu valor máximo (65535) é ultrapassado.

- É possível zerar todos os contadores estatísticos mostrados na tabela anterior escrevendo TRUE seguintes comandos:
  - **DG_NX3035.tRedundancy.RedCmdLoc.bResetNETStatisticsLocal:** zera contadores do CP local.
  - **DG_NX3035.tRedundancy.RedCmdLoc.bResetNETStatisticsRemote:** zera contadores do CP remoto.

A tabela seguinte detalha a subestrutura *sGeneral_DiagExt*, que é uma continuação dos diagnósticos da subestrutura *sGeneralDiag* descrita anteriormente.

Variável AT DG_NX3035.tRedundancy.RedDgnLoc.sGeneral_DiagExt.*	**Descrição**
bFailedPBUS3A	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) na rede PROFIBUS 3A. O modo ativo (comunicando com escravos) é assumido pelo CP Ativo. O modo passivo (comunicando com o mestre em modo ativo) é assumido pelo CP Não-Ativo. Esta falha também pode ser indicada caso o módulo NX5001 tenha falha em seu microprocessador, ou caso não consiga se comunicar com a UCP via barramento.
	FALSE – Não existem falhas na rede PROFIBUS 3A.
bFailedPBUS3B	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) na rede PROFIBUS 3B. O modo ativo (comunicando com escravos) é assumido pelo CP Ativo. O modo passivo (comunicando com o mestre em modo ativo) é assumido pelo CP Não-Ativo. Esta falha também pode ser indicada caso o módulo NX5001 tenha falha em seu microprocessador, ou caso não consiga se comunicar com a UCP via barramento.
	FALSE – Não existem falhas na rede PROFIBUS 3B.
bFailureProfibus_3	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) na rede PROFIBUS 3. Caso a rede PROFIBUS 3 seja redundante, *bFailureProfibus_3* resulta de um "E lógico" entre *bFailedPBUS3A* e *bFailedPBUS3B*. Caso a rede PROFIBUS 3 não seja redundante, *bFailureProfibus_3* é uma cópia de *bFailedPBUS3A*.
	FALSE – Não existem falhas na rede PROFIBUS 3.
bFailedPBUS4A	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) na rede PROFIBUS 4A. O modo ativo (comunicando com escravos) é assumido pelo CP Ativo. O modo passivo (comunicando com o mestre em modo ativo) é assumido pelo CP Não-Ativo. Esta falha também pode ser indicada caso o módulo NX5001 tenha falha em seu microprocessador, ou caso não consiga se comunicar com a UCP via barramento.
	FALSE – Não existem falhas na rede PROFIBUS 4A.
bFailedPBUS4B	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) na rede PROFIBUS 4B. O modo ativo (comunicando com escravos) é assumido pelo CP Ativo. O modo passivo (comunicando com o mestre em modo ativo) é assumido pelo CP Não-Ativo. Esta falha também pode ser indicada caso o módulo NX5001 tenha falha em seu microprocessador, ou caso não consiga se comunicar com a UCP via barramento.
	FALSE – Não existem falhas na rede PROFIBUS 4B.
bFailureProfibus_4	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) na rede PROFIBUS 4. Caso a rede PROFIBUS 4 seja redundante, *bFailureProfibus_4* resulta de um "E lógico" entre *bFailedPBUS4A* e *bFailedPBUS4B*. Caso a rede PROFIBUS 4 não seja redundante, *bFailureProfibus_4* é uma cópia de *bFailedPBUS4A*.
	FALSE – Não existem falhas na rede PROFIBUS 4.
bFailureCommBusAny	TRUE – Existe falha vital em alguma rede de comunicação Ethernet configurada para falha vital.
	FALSE – Nenhuma rede de comunicação Ethernet configurada para falha vital apresenta falha.
bFailureCommBusAll	TRUE – Existe falha em todas redes de comunicação Ethernet configuradas para falha vital.
	FALSE – Não existe falha em todas redes de comunicação Ethernet configuradas para falha vital.
byFailedCommBusCount	Quantidade de redes de campo de comunicação Ethernet reportando falhas.
bRemCpuKeepAliveNet1	TRUE – A UCP está recebendo pacotes de Keep Alive da UCP do outro half-cluster, através da porta Ethernet NET1. Os pacotes de Keep Alive só são enviados em projetos que não têm nenhuma rede PROFIBUS.
	FALSE – Pacotes de Keep Alive não estão sendo recebidos através da porta Ethernet NET1.

Variável AT DG_NX3035.tRedundancy.RedDgnLoc.sGeneral_DiagExt.*	Descrição
bRemCpuKeepAliveNet2	TRUE – A UCP está recebendo pacotes de Keep Alive da UCP do outro half-cluster, através da porta Ethernet NET2. Os pacotes de Keep Alive só são enviados em projetos que não têm nenhuma rede PROFIBUS.
	FALSE – Pacotes de Keep Alive não estão sendo recebidos através da porta Ethernet NET2.
bFailedPBUS5A	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) na rede PROFIBUS 5A. O modo ativo (comunicando com escravos) é assumido pelo CP Ativo. O modo passivo (comunicando com o mestre em modo ativo) é assumido pelo CP Não-Ativo. Esta falha também pode ser indicada caso o módulo NX5001 tenha falha em seu microprocessador, ou caso não consiga se comunicar com a UCP via barramento.
	FALSE – Não existem falhas na rede PROFIBUS 5A.
bFailedPBUS5B	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) na rede PROFIBUS 5B. O modo ativo (comunicando com escravos) é assumido pelo CP Ativo. O modo passivo (comunicando com o mestre em modo ativo) é assumido pelo CP Não-Ativo. Esta falha também pode ser indicada caso o módulo NX5001 tenha falha em seu microprocessador, ou caso não consiga se comunicar com a UCP via barramento.
	FALSE – Não existem falhas na rede PROFIBUS 5B.
bFailureProfibus_5	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) na rede PROFIBUS 5. Caso a rede PROFIBUS 5 seja redundante, *bFailureProfibus_3* resulta de um "E lógico" entre *bFailedPBUS5A* e *bFailedPBUS5B*. Caso a rede PROFIBUS 5 não seja redundante, *bFailureProfibus_5* é uma cópia de *bFailedPBUS5A*.
	FALSE – Não existem falhas na rede PROFIBUS 5.
bFailedPBUS6A	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) na rede PROFIBUS 6A. O modo ativo (comunicando com escravos) é assumido pelo CP Ativo. O modo passivo (comunicando com o mestre em modo ativo) é assumido pelo CP Não-Ativo. Esta falha também pode ser indicada caso o módulo NX5001 tenha falha em seu microprocessador, ou caso não consiga se comunicar com a UCP via barramento.
	FALSE – Não existem falhas na rede PROFIBUS 6A.
bFailedPBUS6B	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) na rede PROFIBUS 6B. O modo ativo (comunicando com escravos) é assumido pelo CP Ativo. O modo passivo (comunicando com o mestre em modo ativo) é assumido pelo CP Não-Ativo. Esta falha também pode ser indicada caso o módulo NX5001 tenha falha em seu microprocessador, ou caso não consiga se comunicar com a UCP via barramento.
	FALSE – Não existem falhas na rede PROFIBUS 6B.
bFailureProfibus_6	TRUE – Este CP não consegue comunicar-se no papel de mestre (modo ativo ou passivo) na rede PROFIBUS 6. Caso a rede PROFIBUS 6 seja redundante, *bFailureProfibus_4* resulta de um "E lógico" entre *bFailedPBUS6A* e *bFailedPBUS6B*. Caso a rede PROFIBUS 6 não seja redundante, *bFailureProfibus_6* é uma cópia de *bFailedPBUS6A*.
	FALSE – Não existem falhas na rede PROFIBUS 6.
bAdditionalSettingsDiff	TRUE – As configurações da web para Firewall, SNMP, FTP, SysLog, Cartão de Memória ou os direitos de usuário neste CP são diferentes das configurações no CP remoto, ou há uma falha nos links de redundância.
	FALSE – As configurações e direitos de usuários deste CP são iguais ao do CP remoto.
dwAdditionalSettingsCRC	CRC de 32 bits das configurações da web para Firewall, SNMP, FTP, SysLog, Cartão de Memória e direitos de usuário, utilizado para detectar diferenças entre as configurações dos 2 CPs.

Tabela 168: Subestrutura *sGeneral_DiagExt*

6.6.4.2. Comandos da Redundância

Os comandos da redundância estão nas seguintes estruturas:

- **DG_NX3035.tRedundancy.RedCmdLoc:** comandos do CP local
- **DG_NX3035.tRedundancy.RedCmdRem:** comandos do CP remoto

Estes comandos têm várias utilidades, por exemplo, solicitar uma mudança no estado de redundância do CP local ou do CP remoto.

ATENÇÃO

O diagnóstico *DG_NX3035.tRedundancy.RedDgnLoc.sGeneral_Diag.bExchangeSync* informa se o serviço de Troca de Diagnósticos e Comandos foi concluído com sucesso, caso tenha o valor TRUE. Caso o valor deste diagnóstico seja FALSE, comandos do CP local não foram copiados para o CP remoto e, portanto, não serão executados no CP remoto.

Como *RedCmdRem* é uma cópia de *RedCmdLoc* do CP remoto, as duas estruturas têm o mesmo formato.

Os comandos das estruturas *RedCmdLoc* e *RedCmdRem* possuem sempre um sufixo que pode ser Local ou Remote. Comandos com o sufixo Local devem ser executados no CP local, enquanto comandos com o sufixo Remote devem ser executados no CP remoto. Por exemplo:

- Ao escrever TRUE no comando *RedCmdLoc.bInactiveLocal* do CP local, estamos solicitando que o CP local mude seu estado de redundância para Inativo.
- Ao escrever TRUE no comando *RedCmdLoc.bInactiveRemote* do CP local, estamos solicitando que o CP remoto mude seu estado de redundância para Inativo. Este comando será recebido no CP remoto em RedCmdRem.bInactiveRemote, e então o CP remoto executará a mudança de estado para Inativo.

Deve-se lembrar que, em cada ciclo da MainTask, a estrutura *RedCmdLoc* do CP local é transmitida para a estrutura *RedCmdRem* do CP remoto.

Para disparar um comando, deve-se sempre ligar o bit correspondente em *RedCmdLoc*. Isto pode ser feito por um sistema SCADA, fazendo uma escrita via Mastertool, ou até mesmo ligando o bit de comando dentro de uma POU como *ActivePrg* ou *NonSkippedPrg*.

O usuário não precisa se preocupar com o desligamento do bit de comando, que será feito automaticamente pelo gerenciador de redundância:

- No caso de comandos executados no próprio CP local (RedCmdLoc + comando com sufixo Local), o bit é desligado assim que o comando for percebido e executado pelo próprio CP local.
- No caso de comandos enviados para o CP remoto para serem executados no CP remoto (*RedCmdLoc* + comando com sufixo Remote):
  - No CP remoto onde o comando é recebido em *RedCmdRem*, o comando é executado quando o gerenciador de redundância percebe uma borda de subida no bit de comando.
  - No CP local onde o comando foi escrito em *RedCmdLoc*, o bit é desligado automaticamente no próximo ciclo da MainTask.

É importante salientar que todos os acionamentos ou desligamentos de comandos são registrados nos logs de eventos da redundância, permitindo a avaliação histórica, por exemplo, para identificar as causas de um switch-over.

A tabela seguinte detalha as estruturas *RedCmdLoc* e *RedCmdRem*, que possuem formatos idênticos.

Variável AT DG_NX3035.tRedundancy.RedCmdLoc.*	Descrição
bStandbyLocal	TRUE – Solicita que o CP local mude para o estado Reserva.
	FALSE – Sem comando.
bInactiveLocal	TRUE – Solicita que o CP local mude para o estado Inativo.
	FALSE – Sem comando.
bResetNETStatisticsLocal	TRUE – Este comando reseta as estatísticas de NETA / NETB (veja sub-estrutura *SNET_Stat* em *RedDgnLoc* e *RedDgnRem*).
	FALSE – Sem comando.
bStandbyRemote	TRUE – Solicita que o CP remoto mude para o estado Reserva.
	FALSE – Sem comando.
bInactiveRemote	TRUE – Solicita que o CP remoto mude para o estado Inativo.
	FALSE – Sem comando.
bResetNETStatisticsRemote	TRUE – Este comando reseta as estatísticas de NETA / NETB do CP remoto (veja sub-estrutura *SNET_Stat* em *RedDgnRem*).
	FALSE – Sem comando.

Tabela 169: Estruturas *RedCmdLoc* e *RedCmdRem*

6.6.4.3. Informações do Usuário Trocadas entre CPs Local e Remoto

As informações do usuário trocadas entre CPs local e remoto estão nas seguintes estruturas:

- **DG_NX3035.tRedundancy.RedUsrLoc:** informações transmitidas do CP local para o CP remoto
- **DG_NX3035.tRedundancy.RedUsrRem:** informações transmitidas do CP remoto para o CP local

Estas estruturas de dados podem ser utilizadas para trazer informações não redundantes do CP local para o remoto, ou vice-versa. Em cada estrutura (*RedUsrLoc* e *RedUsrRem*) o usuário tem um array de 128 bytes disponível para transmitir as informações que desejar. Como exemplo de tais informações, podemos citar diagnósticos não redundantes que não estejam mapeados nas estruturas *RedDgnLoc* e *RedDgnRem*.

ATENÇÃO

O diagnóstico *DG_NX3035.tRedundancy.RedDgnLoc.sGeneral_Diag.bExchangeSync* informa se o serviço de Troca de Diagnósticos e Comandos foi concluído com sucesso, caso tenha o valor TRUE. Caso o valor deste diagnóstico seja FALSE, informações do usuário do CP remoto não foram copiadas e podem estar com valores obsoletos.

Como *RedUsrRem* é uma cópia de *RedUsrLoc* do CP remoto, as duas estruturas têm o mesmo formato. Trata-se do array:

```
abyUser : ARRAY [1 .. 128] OF BYTE;
```

6.6.4.4. Diagnósticos MODBUS utilizados na Redundância

Para verificar se existe falha em todos os MODBUS Servidores configurados em uma instância MODBUS Cliente, existe um diagnóstico específico em cada instância MODBUS Cliente configurada. Na tabela abaixo, é exibido o diagnóstico para este tipo de falha em uma instância MODBUS Cliente chamada *MODBUS_Symbol_Client*.

Variável DG_MODBUS_Symbol_Client.tDiag.*	Descrição
bAllDevicesCommFailure	TRUE – Todos os Servidores configurados neste Cliente estão apresentando erro.
	FALSE – Existe pelo menos um Servidor configurado neste Cliente que não está apresentando erro.

Tabela 170: Diagnóstico MODBUS Cliente

Quando configurado com falha vital Ethernet, este diagnóstico é consultado e 3 segundos após a alteração do seu estado de FALSE para TRUE, ocorre um switch-over caso as outras condições para este evento estejam satisfeitas.

6.6.4.5. Log de Eventos da Redundância

O Mastertool permite observar diversos logs para um CP Nexto, entre os quais encontra-se os Log de Eventos da Redundância. Estas mensagens, específicas para a redundância, registram no *Log de Sistema* modificações relevantes em campos das estruturas de dados de diagnósticos e comandos de redundância.

Cada linha mostrada no log possui as seguintes colunas:

- **Severidade:** informação, advertência, erro ou exceção.
- **Marca de Tempo:** data e hora do evento.
- **Descrição:** texto que descreve o evento.
- **Componente:** componente que gerou o evento, que no caso do Log de Eventos da Redundância, será o *IoDrvNextoRedundancy*.

O texto na coluna *Descrição* contém informações sobre o evento que ocorreu.

Um exemplo da coluna *Descrição* poderia ser o seguinte:

Redundancy new state (local): Starting

Para acessar esta tela, deve-se dar um duplo clique sobre o dispositivo (NX3035) na árvore de dispositivos, e depois selecionar a aba *Log*. Existe um filtro que permite selecionar somente o componente *IoDrvNextoRedundancy*, para exibir somente os eventos de redundância.

ATENÇÃO

Alguns diagnósticos podem apontar possíveis falhas durante a inicialização do sistema redundante e nos primeiro ciclos das tarefas. Mas, em um correto funcionamento do sistema, estes diagnósticos voltam a indicar a ausência de erros logo após a inicialização do sistema.

7. Manutenção

Uma das características da Série Nexto é a geração de diagnósticos de comportamentos atípicos, sejam eles falhas, erros ou modos de operação, possibilitando ao operador identificar e solucionar problemas que venham a ocorrer com o sistema com grande facilidade.

As UCPs Nexto proporcionam diversas maneiras de se visualizar os diagnósticos gerados pelo sistema, são elas:

- One Touch Diag
- Diagnósticos via LED
- Diagnósticos via Página Web de Sistema
- Diagnósticos via Variáveis
- Diagnósticos via Blocos Funcionais

O primeiro deles é uma característica inovadora da Série Nexto, a qual possibilita um rápido acesso às condições anormais da aplicação. O segundo é puramente visual, gerado através de dois LEDs presentes no painel (DG e WD) e também dos LEDs presentes no conector RJ45 (exclusivos para a conexão Ethernet). A próxima característica é a visualização gráfica do bastidor e dos respectivos módulos configurados na aba *Informações do Barramento* da Página Web de Sistema, sendo permitido o acesso individual do estado de operação e os diagnósticos ativos. Os diagnósticos também são armazenados diretamente em variáveis da UCP, tanto de representação direta (%Q), como as simbólicas mapeadas através da diretiva AT, e podem ser utilizados pela aplicação do usuário, por exemplo, sendo apresentados em um sistema de supervisão. Os últimos apresentam condições específicas de funcionamento do sistema.

A função destes diagnósticos é apontar possíveis problemas de instalação ou configuração do sistema, e de problemas ou deficiências das redes de comunicação. O capítulo de Manutenção deve ser consultado pelo usuário sempre que necessário.

7.1. Diagnósticos do Módulo

7.1.1. One Touch Diag

One Touch Diag (OTD), ou seja, diagnóstico com um único pressionamento, é uma característica exclusiva que a Série Nexto traz para os controladores programáveis. Com este novo conceito, o usuário pode verificar os diagnósticos de qualquer módulo presente no sistema diretamente no visor gráfico da UCP, com um único pressionamento na tecla de diagnóstico do respectivo módulo. Essa é uma poderosa ferramenta de diagnóstico que pode ser usada offline (sem sistema de supervisão ou software programador), tornando mais fácil encontrar e resolver rapidamente possíveis problemas.

A tecla de diagnóstico está localizada na parte superior da UCP, em local de fácil acesso, e, além de fornecer os diagnósticos ativos, permite o acesso ao menu de navegação, descrito no capítulo Configuração – Menu Informativo e de Configuração da UCP.

A figura abaixo mostra a localização da tecla na UCP:

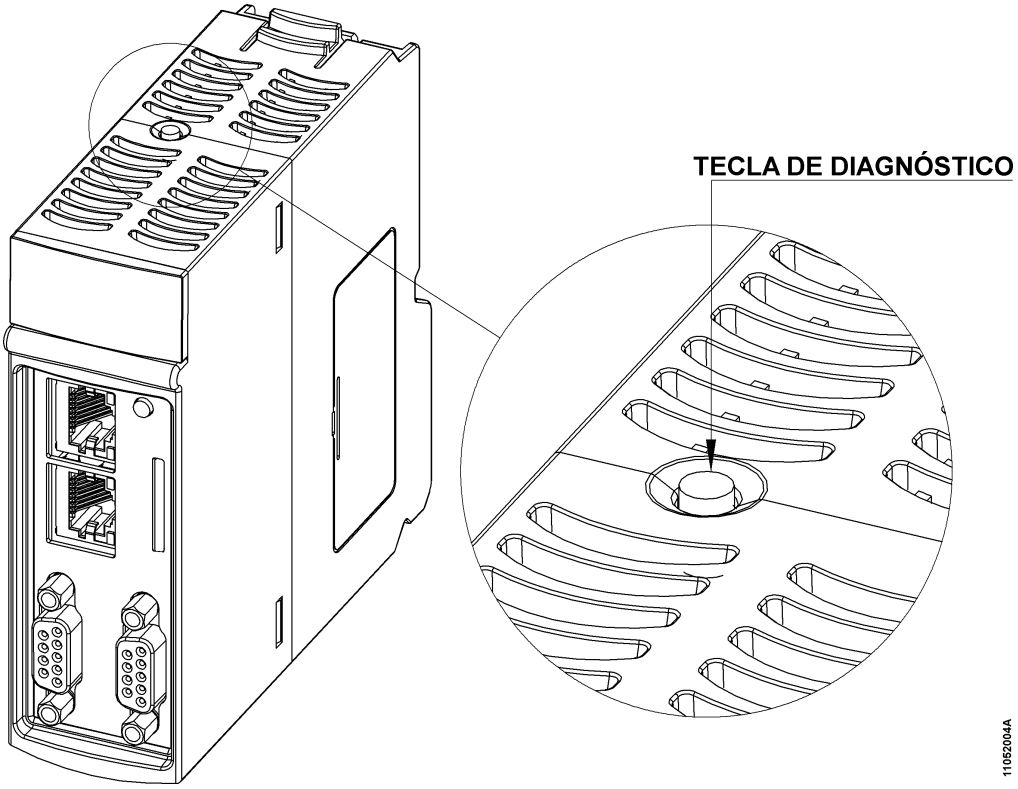


Figura 235: Tecla de Diagnóstico

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.

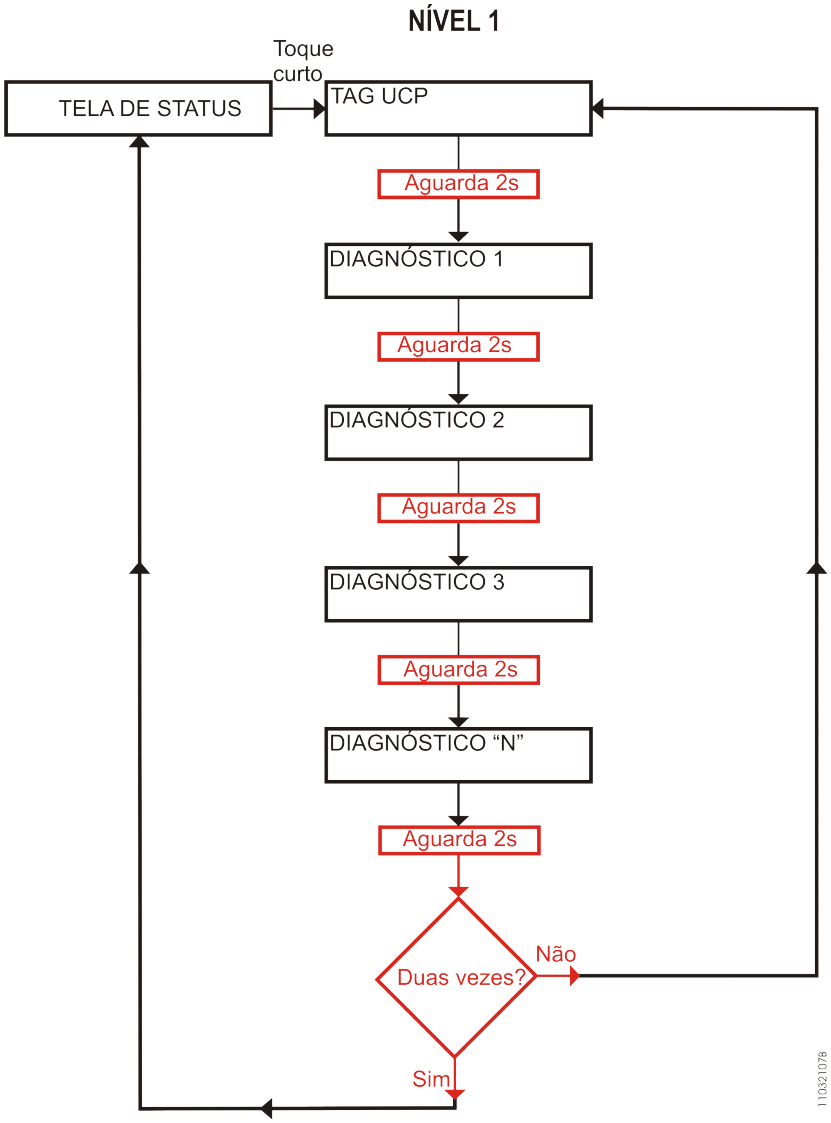


Figura 236: Visualização dos Diagnósticos da UCP

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 175

Tabela 171: 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 175.

Caso ocorra uma situação de tecla presa de algum módulo de E/S presente no barramento, a tecla de diagnóstico deste deixará de indicar diagnósticos no visor gráfico da UCP quando for pressionada, neste caso a UCP irá indicar que há módulos com diagnóstico ativo. Para que seja possível eliminar este diagnóstico da UCP, será necessário realizar uma troca a quente do módulo com o diagnóstico ativo.

Para mais detalhes sobre o procedimento de visualização dos diagnósticos da UCP ou de outros módulos do barramento, ver descrição no Manual de Utilização Série Nexto – MU214000.

7.1.2. Diagnósticos via LED

Este produto possui um LED para indicação de diagnóstico (LED DG) e um LED para indicar ocorrência de cão-de-guarda (LED WD). A Tabela 172 e Tabela 173 mostram o significado de cada estado e suas respectivas descrições.

7.1.2.1. DG (Diagnóstico)

Verde	Vermelho	Descrição	Causas	Prioridade
Desligado	**Desligado**	Não utilizado	Sem fonte de alimentação. Problema de Hardware	-
Ligado	**Desligado**	Aplicação em Execução (Modo Run)	-	3 (Baixa)
Desligado	**Ligado**	Aplicação Parada (Modo Stop)	-	3 (Baixa)
Piscando 2x	**Desligado**	Módulos do barramento com diagnóstico	No mínimo um módulo do barramento, incluindo a UCP, está com algum diagnóstico ativo	1
Piscando 3x	**Desligado**	Forçamento de dados	Alguma área de memória está sendo forçada pelo usuário via Mastertool	2
Desligado	**Piscando 4x**	Erro de configuração ou de hardware no barramento.	O barramento está danificado ou não está configurado corretamente.	0 (Alta)

Tabela 172: Descrição dos Estados do LED de Diagnóstico

7.1.2.2. WD (Cão-de-guarda)

LED Vermelho	Descrição	Causas	Prioridade
Desligado	Sem indicação de cão-de-guarda	Operação normal	3 (Baixa)
Piscando 1x	Cão-de-guarda de Software	Cão-de-guarda da aplicação de usuário	2
Ligado	Cão-de-guarda de Hardware	Módulo danificado e/ou sistema operacional corrompido	1 (Alta)

Tabela 173: 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 179.

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 42.

7.1.2.3. LEDs Conector RJ45

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

Amarelo	Verde	Descrição
○	○	Ausência de LINK de rede
○	●	LINK de rede de 1000 Mbits/s
●	○	LINK de rede de 100 Mbits/s
●	●	LINK de rede de 10 Mbits/s
X	-	Ocorrência de transmissão ou recepção na rede Ethernet, pelo ou para este endereço IP. Pisca sob demanda da UCP Nexto, e não a cada transmissão ou recepção, ou seja, pode piscar com uma frequência menor que a frequência real de transmissão ou recepção.

Tabela 174: Significado dos LEDs Ethernet

7.1.3. Diagnósticos via Página Web de Sistema

Além das características apresentadas anteriormente, a Série Nexto traz para o usuário uma ferramenta inovadora de acesso aos diagnósticos e estados de operação do sistema, através da aba *Informações do Barramento* da Página Web de Sistema.

A utilização, além de dinâmica, é bastante intuitiva e facilita as operações do usuário. Em outras palavras, pode substituir o uso de um sistema de supervisão quando o uso for restrito a verificação de status do sistema.

Para acessar a Página Web de Sistema da UCP desejada, basta utilizar um navegador 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 237:

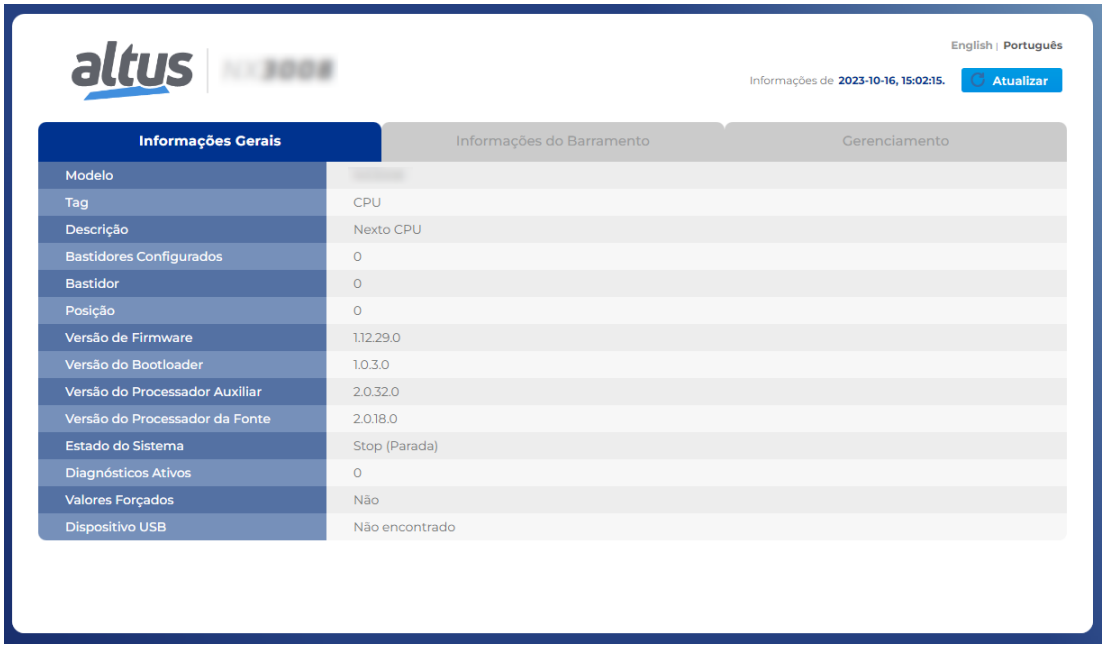


Figura 237: Tela Inicial

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 238. Caso contrário será exibida uma lista com os módulos presentes no sistema, Tags correspondentes e número de diagnósticos ativos:



Figura 238: Informações do Barramento

Ao clicar no módulo com diagnóstico, no mesmo instante são mostrados os diagnósticos ativos do módulo, conforme mostra a Figura 239:

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.

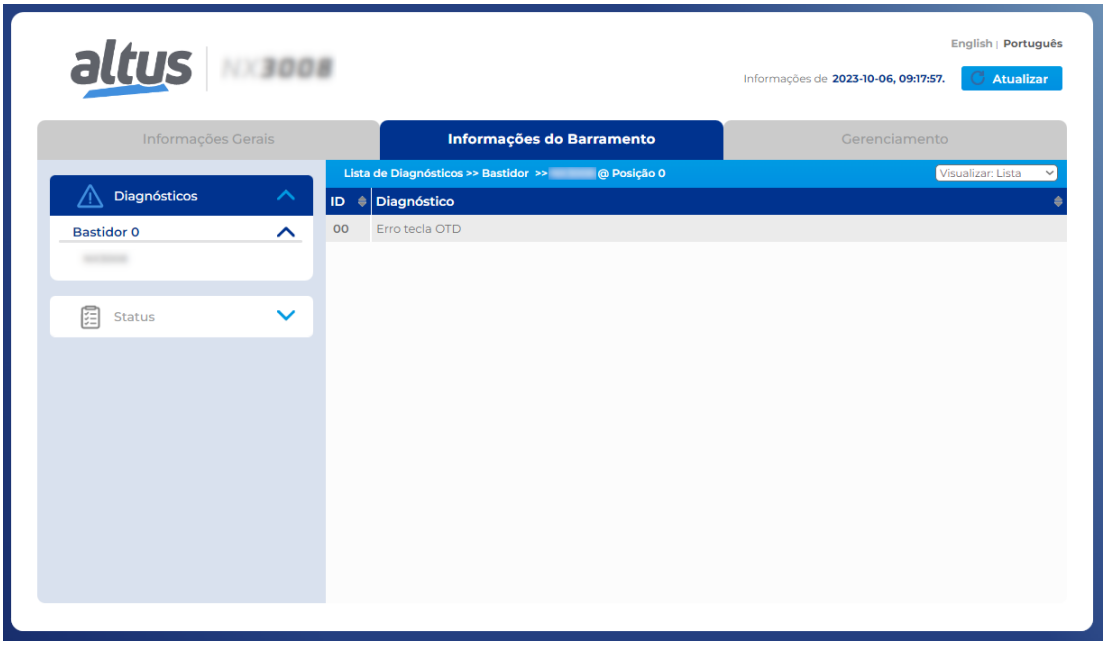


Figura 239: Diagnósticos do Sistema

Caso a seção Status seja selecionada, o estado de todos os diagnósticos detalhados é exibido na tela, conforme mostra a Figura 240:

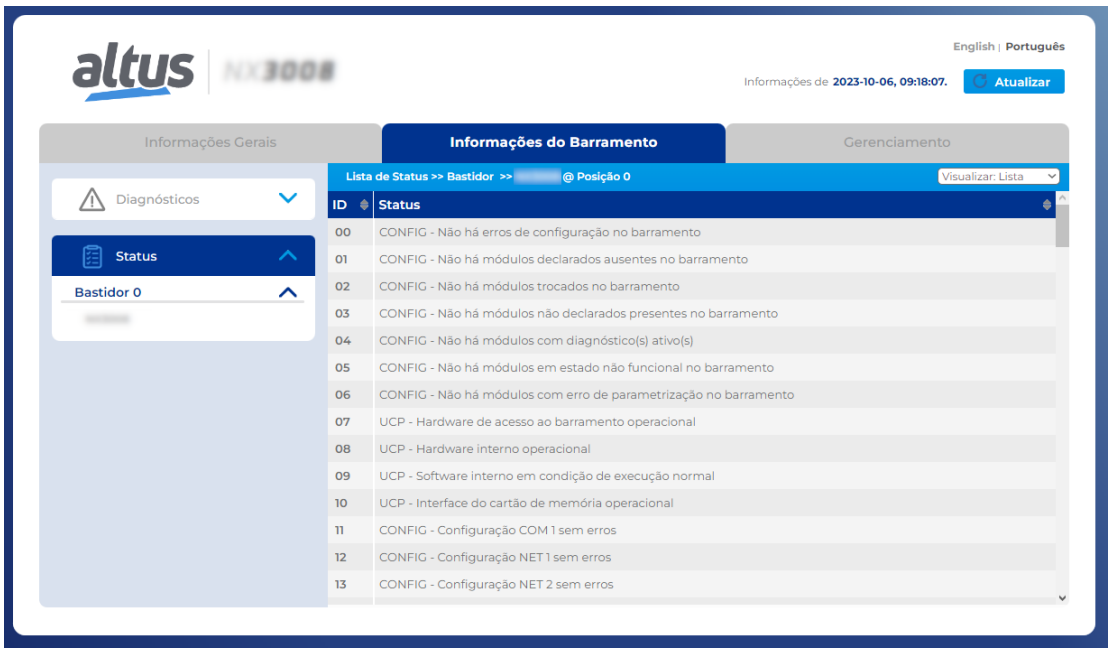


Figura 240: Estado do Sistema

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

7.1.4. Diagnósticos via Variáveis

As UCPs da Série Nexto possuem variáveis para indicação de diagnóstico. Existem estruturas de dados com os diagnósticos de todos os módulos declarados no barramento, mapeadas sobre variáveis de representação direta %Q, e definidas simbolicamente através da diretiva AT, na GVL System_Diagnostics criada automaticamente pelo Mastertool.

Os diagnósticos da UCP estão divididos nas categorias abaixo:

- Diagnósticos Resumidos
- Diagnósticos Detalhados
- Diagnósticos de Redundância

O tamanho em bytes de cada estrutura de diagnóstico deve ser determinado por meio da instrução SIZEOF.

A descrição detalhada dos diagnósticos de redundância está disponível na seção Estrutura de Diagnósticos, Comandos e Informações do Usuário para a Redundância deste manual.

7.1.4.1. 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.

Mensagem de Diagnóstico	Variável DG_Modulo.tSummarized.*	Descrição
EXCECAO NO SOFTWARE	bSoftwareException	TRUE – Uma ou mais exceções geradas pelo software.
		FALSE – Não foram geradas exceções no software.
ERRO CARTAO DE MEMORIA	bMemoryCardError	TRUE – O cartão de memória está inserido na UCP, porém não está funcionando corretamente.
		FALSE – O cartão de memória está funcionando corretamente.
ERRO CONFIG. COM 1	bCOM1ConfigError	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface serial COM 1.
		FALSE – A configuração da interface serial COM 1 está correta.
ERRO CONFIG. NET 1	bNET1ConfigError	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface Ethernet NET 1.
		FALSE – A configuração da interface Ethernet NET 1 está correta.
ERRO CONFIG. NET 2	bNET2ConfigError	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface Ethernet NET 2.
		FALSE – A configuração da interface Ethernet NET 2 está correta.
ERRO CONFIG. NET 3	bNET3ConfigError	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface Ethernet NET 3.
		FALSE – A configuração da interface Ethernet NET 3 está correta.
ERRO CONFIG. NET 4	bNET4ConfigError	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface Ethernet NET 4.
		FALSE – A configuração da interface Ethernet NET 4 está correta.
ERRO CONFIG. NET 5	bNET5ConfigError	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface Ethernet NET 5.
		FALSE – A configuração da interface Ethernet NET 5 está correta.
ERRO CONFIG. NET 6	bNET6ConfigError	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface Ethernet NET 6.
		FALSE – A configuração da interface Ethernet NET 6 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.

Mensagem de Diagnóstico	Variável DG_Modulo.tSummarized.*	Descrição
		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 175: Diagnósticos Resumidos da UCP

Notas:

Configuração incompatível: O diagnóstico de configuração incompatível é gerado caso um ou mais módulos do barramento não conferirem com o que foi declarado, ou seja, nas condições de módulos ausentes ou diferentes. Os módulos inseridos no barramento que não foram declarados no projeto não são considerados.

Módulos trocados: Se somente dois módulos estiverem trocados entre si no respectivo barramento, então o diagnóstico de troca pode ser identificado. Caso contrário, o problema será tratado como "Configuração Incompatível".

Módulos com erro fatal: Caso o diagnóstico de Módulos com Erro Fatal seja verdadeiro, verificar qual é o módulo que está com problema no barramento e encaminhar para a Assistência Técnica da Altus, pois o mesmo está apresentando falha do hardware.

Módulos com erro de parametrização: Caso o diagnóstico de Módulo com Erro de Parametrização seja verdadeiro, verificar se os módulos do barramento estão configurados corretamente e se as versões de firmware e do software Mastertool estão adequadas. Se o problema ocorrer ao inserir um módulo no barramento, verifique se esse módulo suporta troca à quente.

Erro no barramento: Considerado um erro fatal, interrompendo o acesso aos módulos do barramento. Caso o diagnóstico de erro no barramento seja verdadeiro, talvez haja um problema de hardware nas linhas de comunicação do barramento, sendo assim, deve-se entrar em contato com a Assistência Técnica da Altus.

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

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

Mensagem de Diagnóstico: As mensagens de diagnóstico podem ser visualizadas através do visor gráfico da UCP através da tecla OTD ou através da WEB na página de diagnósticos da UCP.

7.1.4.2. Diagnósticos Detalhados

As tabelas a seguir mostram os diagnósticos detalhados das UCPs da Série Nexto, para sua consulta é importante verificar as observações abaixo:

- **Visualização das Estruturas de Diagnóstico:** As Estruturas de Diagnóstico adicionadas ao projeto podem ser visualizadas acessando o item "*Library Manager*" na árvore de dispositivos da janela do Mastertool. 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.

7.1.4.2.1. Diagnósticos Detalhados Grupo Target

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

Tabela 176: Diagnósticos Detalhados Grupo Target

7.1.4.2.2. Diagnósticos Detalhados Grupo Hardware

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 177: Diagnósticos Detalhados Grupo Hardware

7.1.4.2.3. Diagnósticos Detalhados Grupo Exception

DG_Modulo.tDetailed.Exception.*	Tamanho	Descrição
wExceptionCode	WORD	Código de Exceção gerado pelo RTS, o significado dos códigos pode ser conferido na Tabela 179.
byProcessorLoad	BYTE	Nível, em percentagem (%), de carga no processador.

Tabela 178: 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.

Código	Descrição	Código	Descrição
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 179: Códigos de Exceção RTS

7.1.4.2.4. Diagnósticos Detalhados Grupo RetainInfo

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

Tabela 180: Diagnósticos Detalhados Grupo RetainInfo

7.1.4.2.5. Diagnósticos Detalhados Grupo Reset

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

Tabela 181: 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.

7.1.4.2.6. Diagnósticos Detalhados Grupo Thermometer

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

Tabela 182: 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.

7.1.4.2.7. Diagnósticos Detalhados Grupo Serial

DG_Modulo.tDetailed.Serial.COM1.*	Tamanho	Descrição
byProtocol	BYTE	Protocolo selecionado na COM 1: 00: Sem protocolo 01: MODBUS RTU Master 02: MODBUS RTU Slave 03: Outro protocolo
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 183: 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.

7.1.4.2.8. Diagnósticos Detalhados Grupo Ethernet

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

DG_Modulo.tDetailed.Ethernet.*	Tamanho	Descrição
NET[x].dwTransmittedCollisionErrors	DWORD	Contador de erros de colisão na transmissão através da porta NET[x] (0 a 4294967295).
NET[x].dwReceivedBytes	DWORD	Contador de bytes recebidos através da porta NET[x] (0 a 4294967295).
NET[x].dwReceivedPackets	DWORD	Contador de pacotes recebidos através da porta NET[x] (0 a 4294967295).
NET[x].dwReceivedDropErrors	DWORD	Contador de perdas de conexão na recepção através da porta NET[x] (0 a 4294967295).
NET[x].dwReceivedFrameErrors	DWORD	Contador de erros de frame na recepção através da porta NET[x] (0 a 4294967295).

Tabela 184: Diagnósticos Detalhados Grupo Ethernet NET[x]

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

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

Tabela 185: Diagnósticos Detalhados Grupo Ethernet RSTP

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

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

Tabela 186: Diagnósticos Detalhados Grupo Ethernet MRP

Nota:
[x]: é o número do MRP.

7.1.4.2.9. Diagnósticos Detalhados Grupo UserFiles

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

Tabela 187: Diagnósticos Detalhados Grupo UserFiles

Nota:
User Partition: A partição de usuário é uma área da memória reservada para o armazenamento de dados na UCP. Por exemplo: arquivos com extensão PDF, arquivos com extensão DOC e demais dados.

7.1.4.2.10. Diagnósticos Detalhados Grupo UserLogs

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

Tabela 188: Diagnósticos Detalhados Grupo UserLogs

7.1.4.2.11. Diagnósticos Detalhados Grupo MemoryCard

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

Tabela 189: Diagnósticos Detalhados Grupo MemoryCard

7.1.4.2.12. Diagnósticos Detalhados Grupo WHSB

DG_Modulo.tDetailed.WHSB.*	Tamanho	Descrição
byHotSwapAndStartupStatus	BYTE	Informa a situação anormal no barramento que ocasionou a parada da aplicação, para cada Modo de troca a quente. Consulte a Tabela 191 para detalhes das possibilidades.
adwRackIOErrorStatus	DWORD ARRAY (32)	Identificação de erros em módulos de E/S, individualmente. Para mais informações sobre este diagnóstico consultar a nota abaixo.
adwModulePresenceStatus	DWORD ARRAY (32)	Status de presença de módulos declarados de E/S em barramentos, individualmente. Para mais informações sobre este diagnóstico consultar a nota abaixo.
byWHSBBusErrors	BYTE	Contador de falhas no barramento WHSB. Esse contador é reinicializado na energização (0 a 255).

Tabela 190: 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.

Código	Enumerável	Descrição
08	APPL_STOP_MODULES_GETTING_ READY_TO_RUN	Aplicação em Modo Stop e todas as consistências realizadas com sucesso. Os módulos de E/S estão sendo preparados para a partida do sistema. Não é possível colocar a Aplicação em Modo Run.
09	NORMAL_OPERATING_STATE	Aplicação em Modo Run.
10	MODULE_CONSISTENCY_OK	Uso interno.
11	APPL_STOP_DUE_TO_EXCEPTION	Aplicação em modo Stop pois uma exceção ocorreu na UCP.
12	DUPLICATED_SLOT_HOT_SWAP_ DISABLED	Aplicação em Modo Stop devido à ativação do diagnóstico Slots Duplicados, quando configurada troca a quente *"Desabilitada"* ou troca a quente *"Desabilitada, apenas para módulos declarados"*.
13	DUPLICATED_SLOT_HOT_SWAP_ STARTUP_CONSISTENCY	Aplicação em Modo Stop devido à ativação do diagnóstico Slots Duplicados, quando configurada troca a quente *"Habilitada, com consistência na partida"* ou troca a quente *"Habilitada, com consistência na partida somente para os módulos declarados"*.
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 191: Códigos de Situações que Ocasionam Parada da Aplicação

7.1.4.2.13. Diagnósticos Detalhados Grupo Application

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

Tabela 192: Diagnósticos Detalhados Grupo Application

7.1.4.2.14. Diagnósticos Detalhados Grupo SNTP

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

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

Tabela 193: Diagnósticos Detalhados Grupo SNTP

7.1.4.2.15. Diagnósticos Detalhados Grupo Rack

DG_Modulo.tDetailed.Rack.*	Tamanho	Descrição
dwAbsentRacks	DWORD	Cada bit representa um número de identificação de um bastidor, se algum bit for TRUE, isso significa que o bastidor, com o referido número de identificação, está ausente.
dwDuplicatedRacks	DWORD	Cada bit representa um número de identificação de um bastidor, se algum bit for TRUE, isso significa que mais de um bastidor está configurado com o mesmo número de identificação.
dwNonDeclaredRacks	DWORD	Cada bit representa um número de identificação de um bastidor, se algum bit for TRUE, significa que há algum bastidor configurado com um número de identificação que não está declarado no projeto.

Tabela 194: Diagnósticos Detalhados Grupo Rack

7.1.4.2.16. Diagnósticos Detalhados Grupo ApplicationInfo

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

Tabela 195: Diagnósticos Detalhados Grupo ApplicationInfo

7.1.4.2.17. Diagnósticos Detalhados Grupo Firewall

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

Tabela 196: Diagnósticos Detalhados Grupo Firewall

7.1.4.2.18. Diagnósticos Detalhados Grupo OpenVPN

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

Tabela 197: Diagnósticos Detalhados Grupo OpenVPN

7.1.4.2.19. Diagnósticos Detalhados Grupo FTP

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

Tabela 198: Diagnósticos Detalhados Grupo FTP

7.1.5. Diagnósticos via Blocos Funcionais

Os blocos funcionais proporcionam a visualização de alguns parâmetros que não podem ser acessados de outra maneira. A função sobre diagnósticos avançados está localizada na biblioteca *NextoStandard* e está descrita abaixo.

7.1.5.1. GetTaskInfo

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



Figura 241: 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 199: 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 200: 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.
```

7.2. 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.

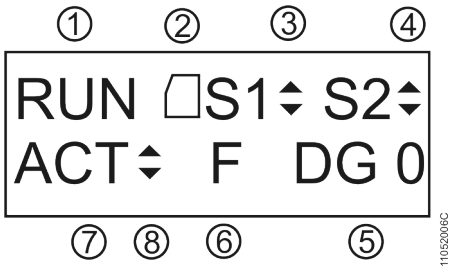


Figura 242: Tela de Status da UCP

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 Interface Serial.
- 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 Interface Serial.
- 5. Indicação da quantidade de diagnósticos ativos na UCP. Caso o número mostrado seja diferente de 0 (zero), existem diagnósticos ativos na UCP. Para maiores detalhes sobre a visualização dos mesmos no visor gráfico da UCP, através da tecla de diagnósticos, consultar seção One Touch Diag.
- 6. Indicação de variáveis forçadas na UCP. Caso o caractere F esteja exibido no visor gráfico, alguma variável está sendo forçada pelo usuário, seja ela variável simbólica ou variável de representação direta mapeada em uma variável simbólica. Para maiores detalhes sobre forçamento de variáveis, consultar seção Escrita e Forçamento de Variáveis.
- 7. Identificação do estado da redundância na UCP (mensagem válida somente na NX3035 em modo redundante). Caso a UCP seja o CP Ativo, a informação ACT será apresentada. Os demais estados possíveis são NCF (Não Configurado), STR (Inicializando), INA (Inativo) e SBY (Reserva).
- 8. Indicação de que está sendo executada a sincronização de projeto. A seta para cima (▲) indica transmissão de dados do projeto e a seta para baixo (▼) indica recepção de dados de um projeto.

Além dos caracteres descritos acima, as UCPs Nexto poderão apresentar algumas mensagens no visor gráfico, correspondentes a algum processo que está sendo executado no momento.

A tabela abaixo mostra as mensagens e suas respectivas descrições:

Mensagem	Descrição
FORMATANDO...	Indica que a UCP está formatando o cartão de memória.
ERRO NA FORMATACAO	Indica que ocorreu um erro durante a formatação do cartão de memória pela UCP.
SEM TAG	Não existe Tag configurada para a UCP no Mastertool.
SEM DESC	Não existe Descrição configurada para a UCP no Mastertool.
ERRO MSG.	Indica que há erro(s) na(s) mensagem(s) de diagnóstico(s) do(s) módulo(s) requisitado(s).
SEM ASSINATURA	Indica que o produto apresentou um problema inesperado. Entre em contato com o setor de Suporte da Altus.
ERRO NA APL. REINICIANDO	Indica que ocorreu um erro na aplicação e o Runtime está reiniciando a aplicação.
APL. NAO CARREGADA	Indica que o Runtime não irá carregar a aplicação.
CARREGANDO APL.	Indica que o Runtime irá carregar a aplicação.
POSICAO INCORRETA	Indica que a UCP está em uma posição incorreta no bastidor.
ERRO FATAL	Indica que há problemas graves na inicialização da UCP como por exemplo, as partições da UCP não foram devidamente montadas. Entre em 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.
REINICIANDO SISTEMA...	Indica que a UCP está sendo reiniciada para que as atualizações tenham efeito.

Tabela 201: Outras Mensagens do Visor Gráfico

7.3. Log de Sistema

O *Log de Sistema* é um recurso disponível no programador Mastertool. É uma importante ferramenta de registro de processo, pois através dele é possível localizar eventos na UCP que podem indicar condições de erro, presença de componentes ou de diagnósticos ativos. Tais eventos podem ser visualizados em ordem cronológica com uma resolução de segundos, com uma capacidade de armazenamento de até mil entradas de log armazenadas na memória interna do CP e não podem ser removidos.

Para acessar esses Logs, basta ir na *Árvore de Dispositivos* e dar um duplo clique em *Device*, e em seguida ir na aba *Log*, onde podem ser visualizadas as centenas de operações como: ciclo máximo das tarefas, acesso de usuários, alteração online, download e upload da aplicação, forçamento das variáveis, sincronização da aplicação em CPs redundantes, atualização de firmware entre outros eventos e ações.

Para que os *Logs* possam ser visualizados, basta estar conectado a uma UCP (Caminho Ativo selecionado) e clicar em . Quando este botão é pressionado os Logs serão exibidos e a atualização deles é feita automaticamente. Quando o botão não está pressionado os Logs serão mantidos na tela, mas não serão atualizados. Ou seja, este botão tem dois estados, sendo que um estado mantém os logs sendo atualizados e no outro estado a atualização está desabilitada. Para deixar de exibir os Logs, basta clicar em .

É possível filtrar os Logs em 4 tipos diferentes: Advertência(s), Erro(s), Exceção (s),Informação (s).

Outra maneira de filtrar as mensagens exibidas para o usuário é selecionar qual componente queremos visualizar.

A *Marca de Tempo* da aba de Log é apresentada pela ferramenta Mastertool a partir de informações fornecidas pelo dispositivo (UCP). A ferramenta Mastertool pode apresentar a Marca de Tempo no horário local (do computador) ou no horário UTC, se marcada a opção *Tempo UTC*.

ATENÇÃO

Se o horário ou o parâmetro de fuso horário do dispositivo estiverem incorretos, a Marca de Tempo apresentada pela ferramenta Mastertool não corresponderá ao tempo esperado.

Para maiores informações sobre os Logs de Sistema, favor consultar o Manual de Utilização do Mastertool e os subcapítulos Relógio RTC e Sincronização de Tempo deste manual.

ATENÇÃO

Os logs de sistema das UCPs, a partir da versão de firmware 1.4.0.33 (Nexto) e 1.14.36.0 (Xtorm), são recarregados no caso de uma reinicialização da UCP ou por uma reinicialização do Runtime System, isto é, será possível visualizar os logs mais antigos quando uma dessas condições ocorrer.

7.4. Não Carregar a Aplicação na Inicialização

Caso seja necessário, o usuário pode optar por não carregar uma aplicação já existente na UCP durante sua inicialização. Para isto, basta energizar a UCP com a tecla de diagnóstico pressionada e mantê-la assim até ser exibida a mensagem na tela "APL. NÃO CARREGADA". Após a inicialização da UCP e a exibição no visor da mensagem avisando que a aplicação não será carregada a UCP estará também em Modo Stop. Caso seja feito um login, o software Mastertool indicará que não existe nenhuma aplicação na UCP. Para voltar a carregar a aplicação, a UCP deve ser reinicializada, ou um novo download da aplicação deve ser feito.

7.5. Falha na Alimentação

A fonte de alimentação da Série Nexto (NX8000) possui um sistema de detecção de falha, de acordo com os níveis definidos em suas características técnicas (consultar Características Técnicas do Módulo de Alimentação 30W 24Vdc – CT114200). Existem duas formas de diagnosticar a falha:

1. Caso a fonte NX8000 seja energizada com tensão inferior ao limite mínimo exigido, será gerado um diagnóstico de falha de alimentação, o qual será reconhecido pela UCP e a mensagem "FALHA NA ALIMENTAÇÃO" será exibida em seu visor. Quando a alimentação estiver dentro dos limites estabelecidos, a UCP irá reconhecer e, automaticamente, será reiniciada com a aplicação do usuário. O diagnóstico ainda estará ativo para mostrar ao usuário que na última inicialização ocorreu falha na alimentação.

2. Caso a fonte NX8000 tenha uma queda de tensão para um valor inferior ao limite mínimo exigido e retornar para um valor acima em até 10 ms, a falha de alimentação não será reconhecida pela UCP e não será gerado o diagnóstico, pois o sistema se mantém intacto durante esse tempo. Porém, se a queda de tensão for superior a 10 ms, a mensagem "FALHA NA ALIMENTAÇÃO" será exibida no visor da UCP e o diagnóstico será ativado.



Figura 243: 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.

7.6. Problemas mais Comuns

Se, ao energizar a UCP, a mesma não entra em funcionamento, os seguintes itens devem ser verificados:

- A temperatura ambiente está dentro da faixa suportada pelos equipamentos?
- A fonte de alimentação do bastidor está sendo alimentada com a tensão correta?
- A fonte de alimentação é o módulo mais à esquerda no bastidor (bastidor sendo visto de frente), e é seguido pela UCP da Série Nexto à sua direita?
- Os equipamentos da rede, como hubs, switches ou roteadores, estão alimentados, interligados, configurados e funcionando corretamente?
- O cabo de rede Ethernet está devidamente conectado à porta NET da UCP Nexto e ao equipamento de rede?
- A UCP da Série Nexto está ligada, em modo de execução (Run) e sem diagnósticos relacionados à hardware?

Se a UCP Nexto indica o estado execução (Run), mas não responde às comunicações solicitadas, sejam elas pelo Mastertool ou através de protocolos, os seguintes itens devem ser verificados:

- A configuração dos parâmetros Ethernet da UCP estão corretos?
- O respectivo protocolo de comunicação está configurado corretamente na UCP?
- As variáveis que habilitam as relações MODBUS estão devidamente habilitadas?

Se nenhum problema for identificado, consulte o Suporte à Clientes Altus.

7.7. Solução de Problemas

A tabela abaixo mostra os sintomas de alguns problemas com suas possíveis causas e soluções. Se o problema persistir, entre em contato com o Suporte Técnico da Altus.

Sintoma	Possível Causa	Solução
Não liga	Falta de alimentação ou alimentado incorretamente.	Verificar se a UCP está conectada corretamente no bastidor. Desenergizar e retirar todos os módulos do barramento, menos a fonte de alimentação e a UCP. Energizar o barramento e verificar o funcionamento da fonte de alimentação, tanto a externa como a fonte conectada ao barramento. Verificar se a tensão de alimentação chega ao borne da fonte de alimentação Nexto, e também se a polarização está correta.

Sintoma	Possível Causa	Solução
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.
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 202: Solução de Problemas

7.8. Manutenção Preventiva

- Deve-se verificar, a cada ano, se os cabos de interligação estão com as conexões firmes, sem depósitos de poeira, principalmente os dispositivos de proteção.
- Em ambientes sujeitos a contaminação excessiva, deve-se limpar periodicamente o equipamento, retirando resíduos, poeira, etc.
- Os diodos TVS utilizados para a proteção contra transientes causados por descargas atmosféricas devem ser inspecionados periodicamente, pois podem estar danificados ou destruídos caso a energia absorvida esteja acima do limite. Em muitos casos, a falha pode não ser evidente ou facilmente visível. Em aplicações críticas, é recomendável a substituição periódica dos diodos TVS, mesmo os que não apresentarem sinais visíveis de falha.
- Aperto e limpeza do barramento a cada 6 meses.
- Para maiores informações, consultar o Manual de Utilização Série Nexto – MU214000.

8. Apêndices

8.1. Gerenciamento de Certificados e Chaves TLS

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

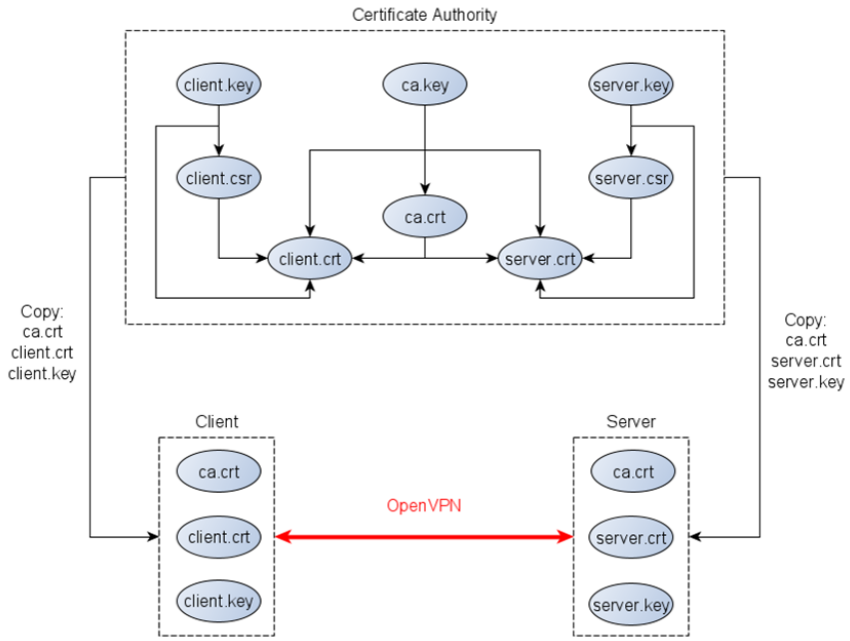


Figura 244: Fluxo de Geração de Certificados TLS

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

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

8.1.1. Geração de Certificados por Easy-RSA

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

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

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

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

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

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

EasyRSA Shell
# _
```

Figura 245: Geração de Certificado utilizando o Easy-RSA (passo 1)

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

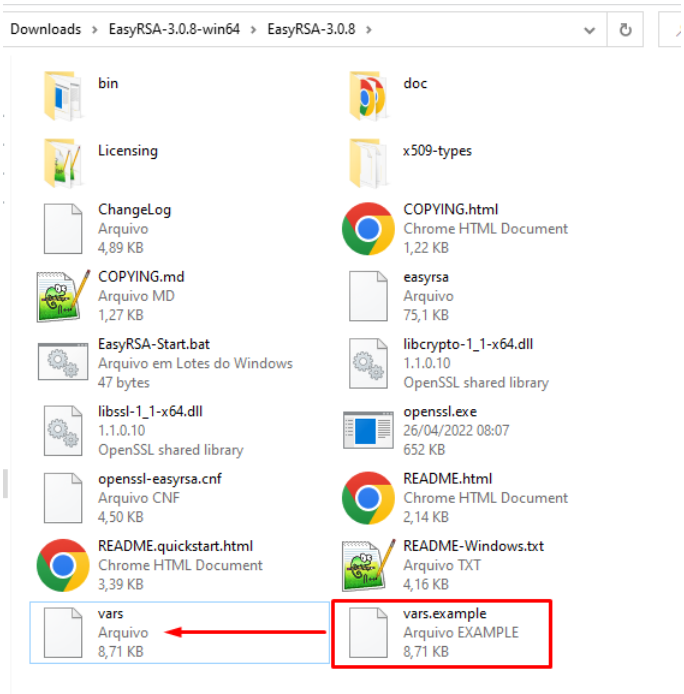


Figura 246: Geração de Certificado utilizando o Easy-RSA (passo 2)

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

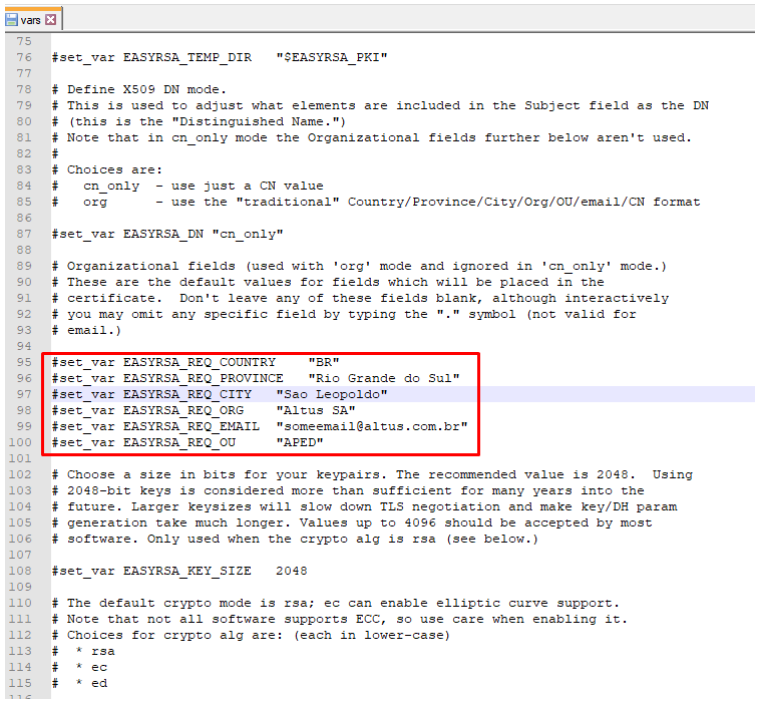


Figura 247: Geração de Certificado utilizando o Easy-RSA (passo 3)

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

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



Figura 248: Geração de Certificado utilizando o Easy-RSA (passo 4)

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

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

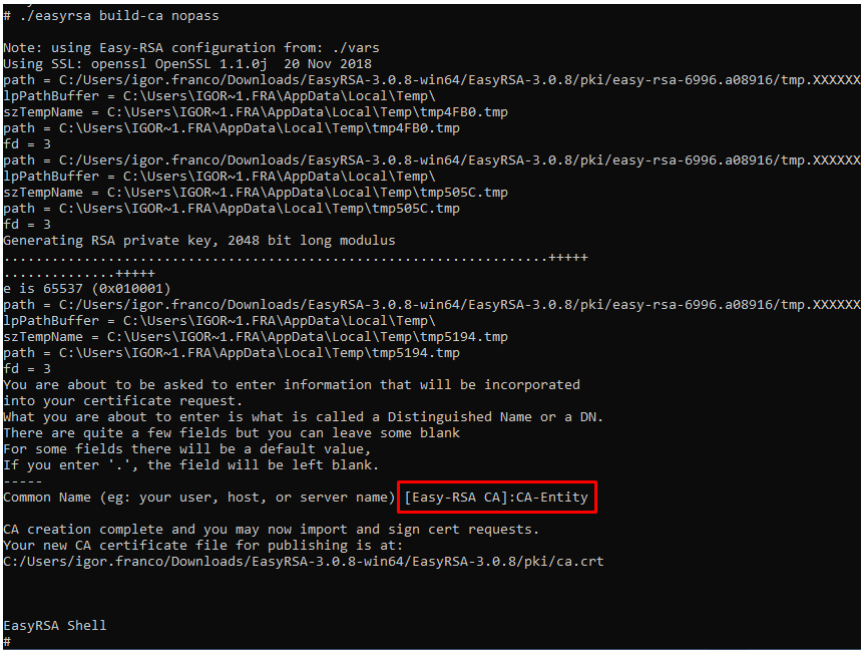


Figura 249: Geração de Certificado utilizando o Easy-RSA (passo 5)

6- Gere a chave do dispositivo e solicite arquivos usando o seguinte comando (altere o *DeviceName* com o nome comum desejado):

```
./easyrsa gen-req DeviceName nopass
```

Novamente, remova o argumento *nopass* para usar uma senha para o arquivo de certificado. Ao entrar com o Nome Comum como argumento, basta pressionar enter quando solicitado (quadrado vermelho).

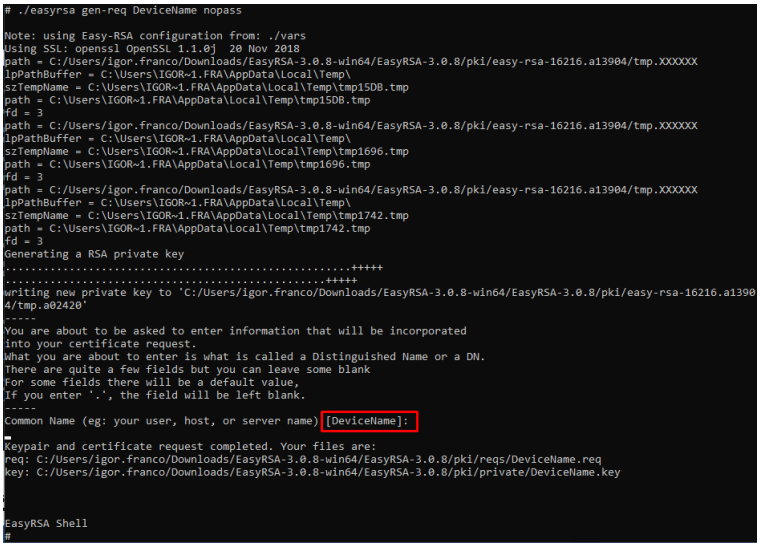


Figura 250: Geração de Certificado utilizando o Easy-RSA (passo 6)

7- Por fim, digite o seguinte comando para gerar o certificado do dispositivo (o *DeviceName* é o nome comum desejado e o *servidor* é o tipo; use *cliente* se estiver gerando para um cliente VPN).

```
./easyrsa sign-req server DeviceName
```

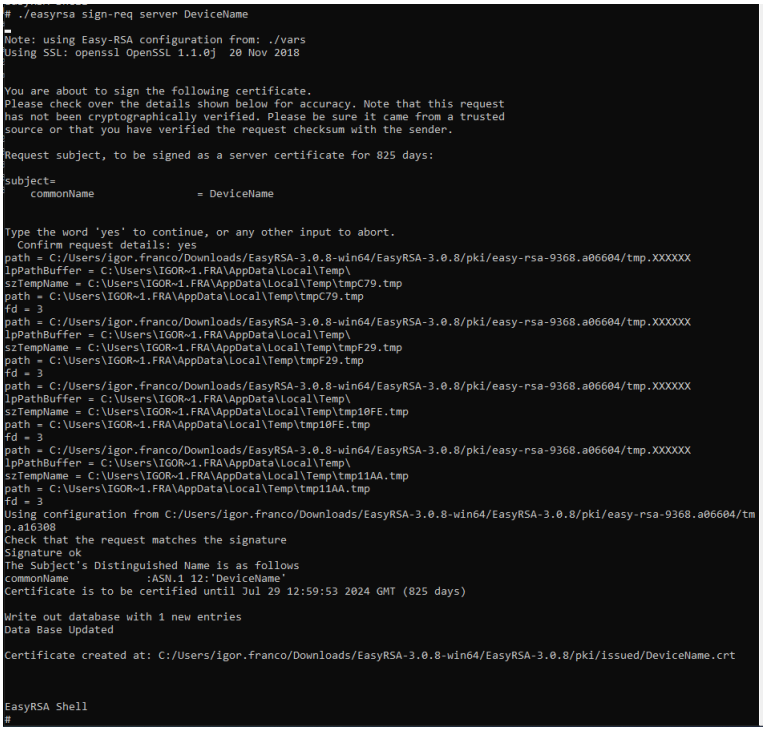


Figura 251: Geração de Certificado utilizando o Easy-RSA (passo 7)

8- Repita as etapas 6 e 7 para gerar mais certificados de dispositivo.

9- Encontre o *ca.crt* na pasta *pki*, as chaves privadas do dispositivo no caminho *pki/private* e os certificados do dispositivo no diretório *pki/issued*.

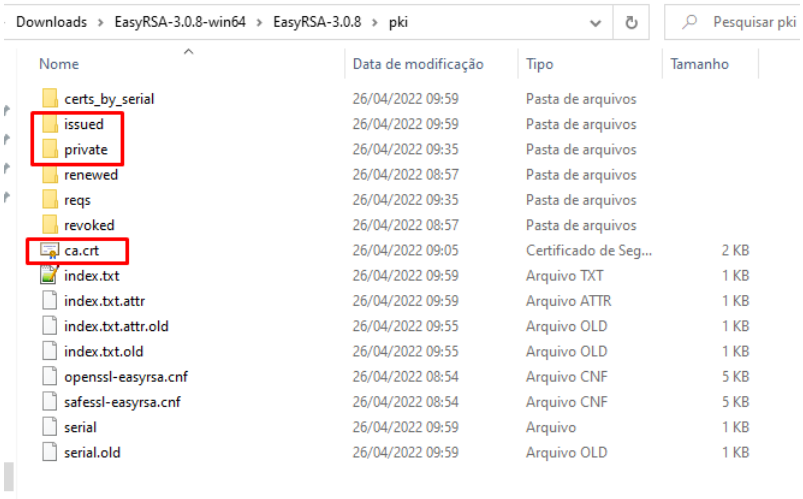


Figura 252: Geração de Certificado utilizando o Easy-RSA (passo 9)

8.1.2. Geração de Certificados por OpenSSL

O OpenSSL é um pacote de código aberto com ferramentas que ajudam a gerar muitos arquivos e recursos de segurança. Este pacote é nativo para a maioria das distribuições do Linux e está disponível para Windows. Apenas lembre-se de definir a pasta OpenSSL no PATH (variável de ambiente) para permitir o uso do comando de qualquer lugar através do prompt. Encontre abaixo o passo a passo utilizando este recurso (todos os arquivos podem ter qualquer nome conforme desejado, os passos consideram apenas um exemplo):

1- Abra um prompt na pasta do certificado (onde criará os arquivos).

2- Gere a chave privada da CA com o seguinte comando:

```
openssl genrsa -out ca.key 4096
```

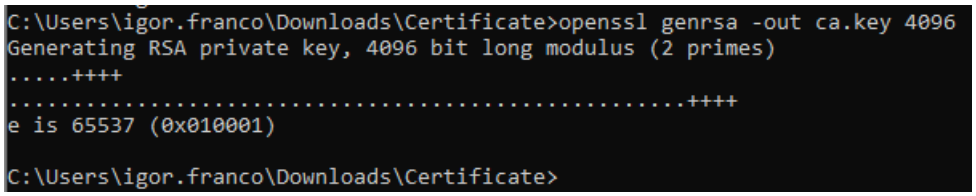


Figura 253: Geração de Certificado utilizando o OpenSSL (passo 2)

3- Em seguida, gere o certificado CA com base na chave privada, utilizando o comando a seguir.

```
openssl req -new -x509 -days 365 -key ca.key -out ca.crt
```

O parâmetro *-days* representa o tempo de expiração do certificado. Configure-o como desejar. Neste exemplo, o certificado é válido por um ano. Preencha os valores solicitados no prompt conforme necessário (pressione enter para usar o padrão, que está entre colchetes []). É obrigatório definir um Nome Comum para o trabalho do certificado.

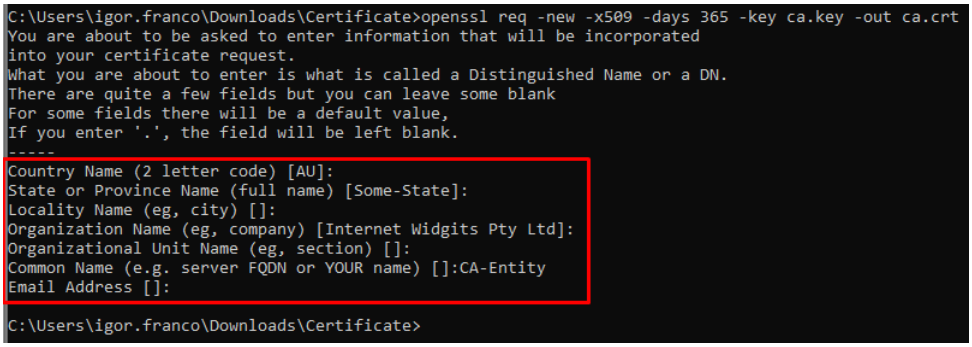


Figura 254: Geração de Certificado utilizando o OpenSSL (passo 3)

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

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

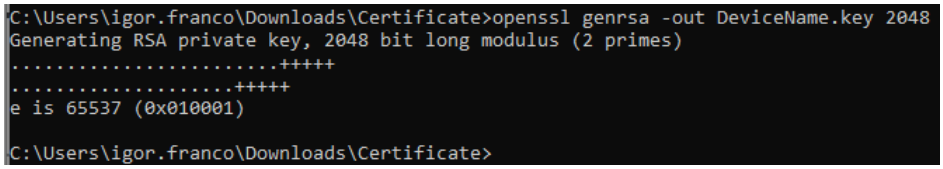


Figura 255: Geração de Certificado utilizando o OpenSSL (passo 4)

5- Depois disso, gere o arquivo de solicitação de certificado com base na chave privada, utilizando o comando a seguir:

```
openssl req -new -key DeviceName.key -out DeviceName.csr
```

Insira as informações desejadas e lembre-se de usar um nome comum diferente da CA.



Figura 256: Geração de Certificado utilizando o OpenSSL (passo 5)

6- Por fim, gere o certificado do dispositivo usando a chave privada da CA, o certificado da CA e o arquivo de solicitação de certificado do dispositivo, utilizando o comando que segue:

```
openssl x509 -req -days 365 -in DeviceName.csr -CA ca.crt -CAkey ca.key -set_serial 01 -out DeviceName.crt
```

Defina a data de expiração conforme desejado com o parâmetro *-days* e o número de série do certificado com o argumento *-set_serial*.

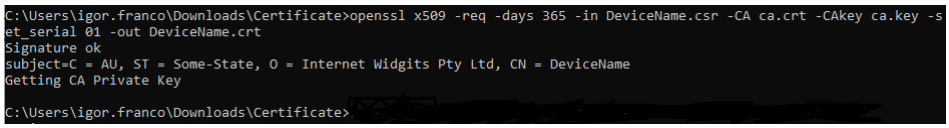


Figura 257: Geração de Certificado utilizando o OpenSSL (passo 6)

7- Repita as etapas 4 a 6 para qualquer novo dispositivo.

8- (Opcional) O OpenSSL oferece uma ferramenta para verificar se o certificado do dispositivo funciona com CA:

Utilize o seguinte comando:

```
openssl verify -purpose sslserver -CAfile ca.crt DeviceName.crt
```

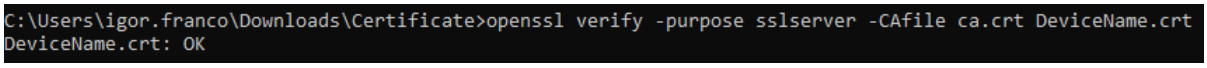


Figura 258: Geração de Certificado utilizando o OpenSSL (passo 8)

8.1.3. Geração de Chave TA pelo OpenVPN

O projeto OpenVPN fornece uma ferramenta para geração de uma chave TLS, comumente chamada de ta.key. Esta chave é uma camada de proteção extra nas portas UDP/TCP de comunicação do OpenVPN, sendo assim, a utilziação desta chave pode ser interpretada como um Firewall de HMAC para a comunicação VPN, fazendo com que seja necessário a existência do parâmetro dos dois lados da comunicação para que ela seja estabelecida.

A geração desta chave no Windows pode ser feita com o comando a seguir:

```
openvpn --genkey secret ta.key
```

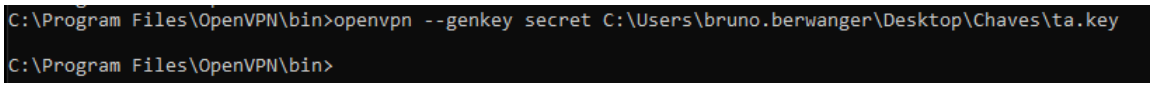


Figura 259: Exemplo de Geração de Chave TA no Windows

Para executar o comando foi utilizado o executável que é instalado juntamente ao pacote do OpenVPN. O diretório utilizado na imagem a cima é um exemplo e é opcional, pode ser utilizado somente o nome do arquivo desejado.

O comando pode ser utilizado para realizar a geração da chave pelo Linux, porém, há uma pequena alteração no comando em relação ao windows. Para gerar a chave no Linux utilize o seguinte comando: *openvpn –genkey –secret ta.key*. Para executa-lo, basta digitar o seguinte comando no terminal:

```
openvpn --genkey --secret ta.key
```

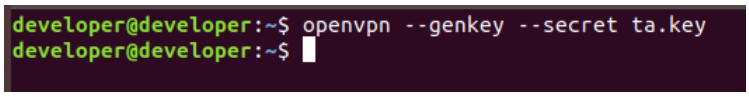


Figura 260: Exemplo de Geração de Chave TA no Linux

Este parâmetro não é obrigatório para a comunicação VPN, porém, caso o servidor esteja utilizando, todos os seus clientes devem também utilizar, sendo que a chave do servidor e dos clientes deve ser a mesma.

8.2. Carga Offline de Programas sem Interrupção do Controle do Processo

A redundância da UCP de controladores programáveis tem como principal objetivo aumentar a disponibilidade do sistema de automação. Para modificações que demandem uma carga off-line sem interrupção do controle do processo, utilizam-se recursos específicos da redundância para não interromper o controle do processo.

8.2.1. Recomendações Gerais

Seguem boas práticas a serem observadas ao efetuar alterações de programação em sistemas redundantes:

- **Antes de alterações importantes:**
  1. Certifique-se de que é possível efetuar login em qualquer CP sem necessidade de download ou online change;
  2. Salve o projeto utilizando os comandos no menu "*Arquivo /Arquivo de Projeto / Salvar/Enviar Arquivo...*", gerando um *projectarchive* que permita retornar ao ponto anterior de backup.
- **Em caso de problemas durante a alteração:**
  1. Recupere o projeto através do menu "*Arquivo / Arquivo de Projeto / Extrair Arquivo...*";
  2. Abra o projeto recuperado. Mesmo que seja feito em outro computador ou usuário, será possível realizar o login sem alterações adicionais.
- **Garantia de consistência:** antes de iniciar qualquer alteração online, confirme que o projeto aberto no Mastertool é idêntico ao que está em execução no CP Ativo (ACT).
- **Alterações de configuração de dispositivo** (qualquer objeto da Treeview no Mastertool exceto GVL ou POU) devem ser descarregadas (download) primeiro no CP Reserva (SBY), com o sincronismo de projeto desabilitado.
- **Não misturar alterações de configuração e dados redundantes:**
  - Primeiro descarregue a nova configuração no CP Reserva com o sincronismo desabilitado;
  - Em seguida, após o CP voltar ao estado Reserva (SBY), promova o switchover;
  - Só então realize alterações de dados redundantes online no novo CP Ativo.
- **Alterações complexas** (exclusão/inserção de variáveis redundantes, movimentação de blocos em POU, mudanças de configuração): faça passo a passo, com compilações intermediárias e backups, garantindo a possibilidade de retornar à versão anterior.
- **Exclusões de código ou variáveis:** recomenda-se inicialmente apenas comentar as linhas, em vez de excluir diretamente.

8.2.2. Alterações de PROFIBUS

Entre as modificações que afetam uma rede PROFIBUS e demandam carga offline, as seguintes **são suportadas** pelo procedimento que permite realizar cargas offline de modificações sem interrupção do controle do processo, desde que a rede PROFIBUS seja redundante:

- Inserir uma nova remota PROFIBUS da Série Nexto ou Série Ponto;
- Inserir um novo módulo de E/S em uma remota PROFIBUS da Série Nexto ou Série Ponto;
- Remover um módulo de E/S em uma remota PROFIBUS da Série Ponto;
- Alterar parâmetro de remotas PROFIBUS da Série Ponto.
- Modificar parâmetros em módulos de E/S de remotas PROFIBUS da Série Nexto ou Série Ponto.

Algumas modificações de projeto demandam a interrupção do controle do processo, tais modificações estão descritas no parágrafo Modificações que Demandam Carga Offline com Interrupção do Controle do Processo. Outras alterações de projeto, relacionadas a rede PROFIBUS, embora suportadas pelo procedimento que permite realizar cargas offline de modificações sem interrupção do controle do processo, podem resultar na descontinuidade das saídas das remotas PROFIBUS. São elas:

- Remover módulo (incluso o "NX9999 Slot Reserved") de remota PROFIBUS da Série Nexto;
- Substituir módulo (incluso o "NX9999 Slot Reserved") de remota PROFIBUS da Série Nexto;
- Alterar parâmetro de remotas PROFIBUS da Série Nexto.

Para estas situações recomenda-se realizar o procedimento que vai exigir a interrupção do controle do processo, descrito na secção Carga Offline de Modificações com Interrupção do Controle do Processo.

8.2.3. Condição Inicial

Antes de iniciar o procedimento descrito a seguir, deve-se garantir o seguinte contexto:

- Um CP está no estado Ativo (ACT);
- O outro CP está no estado Reserva (SBY);
- O sincronismo de projeto está habilitado em ambos os CPs (caso contrário, não haverá transferência do projeto do Ativo para o Reserva);
- O projeto correspondente está aberto no Mastertool, permitindo login sem necessidade de download ou online change;
- Não existem forçamentos aplicados em nenhum CP.

8.2.4. Procedimento para Alteração

Este procedimento é aplicável para alterações exclusivamente de configuração, mas também permite alterações de apenas código, apenas dados não redundantes ou todas estas alterações simultâneas, mas sem qualquer alteração em dados redundantes. No caso de alterações simultâneas de dados redundantes e configuração de dispositivos são necessárias duas etapas: primeiro alterar a configuração utilizando este método e depois alterar os dados redundantes através do método de carga online.

O procedimento a seguir é aplicável para:

- Alterações exclusivamente de configuração;
- Alterações apenas de código ou de dados não redundantes;
- Alterações simultâneas dessas categorias (desde que não envolvam dados redundantes).

ATENÇÃO

Em caso de necessidade de alterar dados redundantes e configuração ao mesmo tempo, o processo deve ser dividido em duas etapas:

1. Alterar a configuração com o método abaixo;
2. Em seguida, alterar os dados redundantes utilizando o procedimento específico para esse caso.

8.2.5. Passos

O procedimento pode ser dividido em quatro etapas principais: preparação, download, troca de controle e validação final.

8.2.5.1. Etapa 1 – Preparação da alteração

1. Editar e compilar o projeto no MasterTool, confirmando ausência de erros.
2. Colocar o CP Reserva (SBY) em estado Inoperante (INA).
3. Selecionar o CP Reserva como caminho ativo no gateway.

8.2.5.2. Etapa 2 – Download no CP Reserva

1. Desabilitar o sincronismo de projeto no CP Reserva.
2. Efetuar login no CP Reserva (o MasterTool permitirá apenas download).
3. Executar o download (o CP Reserva entrará em STOP).
4. Solicitar a volta do CP Reserva para RUN (após inicialização, o CP retorna ao estado Reserva – SBY).

8.2.5.3. Etapa 3 – Troca de controle

1. Executar o switchover entre CP Ativo (ACT) e Reserva (SBY).
2. Habilitar o sincronismo de projeto no novo CP Ativo (ACT).
3. Aguardar a transferência automática do programa para o CP Reserva.

8.2.5.4. Etapa 4 – Validação final

1. Confirmar que ambos os CPs retornaram ao estado Ativo/Reserva (ACT/SBY), restabelecendo a condição inicial.

ATENÇÃO

Este procedimento requer que os dados redundantes da nova versão de projeto sejam idênticos aos do projeto em execução. Caso contrário, o CP Reserva entrará no estado inoperante (INA) após o download.
Além disso, recomenda-se o uso deste procedimento nas seguintes situações:

1. Quando não for mais possível realizar alterações online no CP Ativo;
2. Quando se deseja testar uma nova versão do projeto mantendo a anterior disponível no CP Reserva, possibilitando retorno imediato em caso de falhas.