



www.altus.com.br

APOSTILA DE TREINAMENTO NEXTO XPRESS



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 descritos 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 e MasterTool 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

Introdução	10
O Curso MP11	10
Estruturação do Tutorial	11
Ementa do curso	12
Referências	13
Documentos relacionados	13
Avaliação do Curso	14
Terminologia	14
Convenções Utilizadas	15
Suporte Técnico	15
Base de Conhecimento ALTUS	16
Cursos ALTUS	16
Revisões deste Tutorial	16
1. Hardware do Nexto Xpress	1-1
Características do Nexto Xpress	1-1
O que é o Nexto Xpress?	1-1
Porque o Nexto Xpress?	1-2
Quais São Seus Diferenciais?	1-2
Quais são as principais características do produto?	1-4
Instalação do Nexto Xpress	1-6
Instalação Mecânica	1-6
Instalação Elétrica	1-8
Instalação de Rede	1-9
Instalação dos Módulos de Expansão	1-10
Manutenção do Nexto Xpress	1-13
Diagnósticos	1-13
Diagnósticos via LED	1-13
Pesquisa na Documentação: LEDs de diagnóstico	1-13
Diagnósticos via Web	1-13
Diagnósticos via Variáveis	1-14
Diagnósticos via Função	1-15
Manutenção Preventiva	1-16
2. Software do Nexto Xpress	2-1
Características do MasterTool IEC XE	2-1
Recursos básicos do MasterTool IEC XE	2-3
Recursos avançados do MasterTool IEC XE	2-3
Versões do MasterTool IEC XE	2-4
Informações Gerais sobre o Programador	2-4
Conceitos Básicos Associados aos Ambientes de Programação IEC61131	2-5
Objetos de Dispositivos	2-5
Objetos de Programação	2-5
Janela das POUs	2-5
Dispositivos, Árvore de Dispositivos	2-5
Aplicação	2-6
Task Configuration	2-7
Geração de Código e Alteração Online	2-7

Monitoração	2-7
Depuração	2-7
Segurança	2-7
Funcionalidades do MasterTool IEC XE	2-8
Instalação	2-8
Projeto	2-8
Perfis de Projeto	2-8
Configurando a UCP	2-9
Bibliotecas	2-11
Inserindo uma Instância de Protocolo	2-11
Localizando o Dispositivo	2-13
Login	2-16
Modo Run	2-19
Modo Stop	2-21
Escrita e Forçamento de Variáveis	2-22
Logout	2-23
Upload do Projeto	2-23
Estados de Operação da UCP	2-25
Programas (POUs) e Listas de Variáveis Globais (GVLs)	2-27
Pasta de Eventos do Sistema (SystemEvents)	2-34
Interface do Usuário do MasterTool IEC XE	2-36
Janelas, Visualizações e Janelas do Editor	2-37
Customizando a Interface do Usuário	2-38
Organizando Barras de Menu e Barras de Ferramentas	2-38
Zoom	2-38
Interface do Usuário no Modo Online	2-38
Menus e Comandos Padronizados	2-39
Memória de Arquivos de Usuário	2-40
Gerenciamento de Usuários e Direitos de Acesso	2-40
Editores	2-41
Primeiros Passos com o MasterTool IEC XE	2-42
Iniciando o MasterTool IEC XE	2-42
Criando POUs	2-45
Criando Tarefas	2-48
Configurando Tarefas	2-50
Vinculando uma POU a uma Tarefa	2-52
Configurando a UCP	2-53
Bibliotecas	2-53
Inserindo uma Instância de Protocolo	2-53
Compilando um Projeto	2-59
Simulando um Projeto	2-60
Criação e Execução de Projetos	2-63
Declarar Variáveis na UserPrg	2-63
Digitar o Código no Corpo da UserPrg	2-64
Criar a POU de Programação Adicional (ST Bloco Funcional FB1)	2-65
Definir os Recursos para Executar e Controlar o Programa no Nexto Xpress	2-66
Executar e Monitorar a Aplicação no Nexto Xpress	2-66
Depurar uma Aplicação via Breakpoint e Varredura do Programa	2-69
Estudo Dirigido 2-1: criação de projeto no MasterTool IEC XE	2-69
Ajuda	2-70
Ajuda Sensível ao Contexto	2-70
3. Programação do Nexto Xpress	3-1
Elementos da Aplicação	3-1
Unidade de Organização de Programa	3-1
Lista de Variáveis Globais - GVL	3-2
Gerenciamento de Bibliotecas	3-2
Operadores	3-2

Operandos	3-3
Declaração de Variáveis	3-3
Tipos de Variáveis	3-4
Variáveis Remanentes	3-4
Unidades de Tipo de Dados	3-6
Linguagem de Programação Diagrama Ladder (LD)	3-6
Comandos Básicos do LD	3-9
Criação de POUs	3-13
Estudo Dirigido 3-1: programa para controle de um carro Automático de máquina operatriz	3-13
Estudo Dirigido 3-2: programa para controle de um portão automático	3-14
Utilização de blocos funcionais definidos pelo usuário	3-15
Estudo Dirigido 3-3: bloco funcional de controle de motores de partida direta	3-15
Um pouco de teoria... Programação Padronizada Multi-linguagens	3-16
Linguagens de Programação IEC61131-3	3-17
Diagrama de Blocos Funcionais (FBD)	3-17
Texto Estruturado (ST)	3-17
Sequenciamento Gráfico de Funções (SFC)	3-18
Estudo Dirigido 3-4: bloco funcional de controle de motores com reversão	3-19
Um pouco de teoria... IEC61131 – Parte 8: orientações para aplicação e implementação das linguagens de programação	3-19
Elementos Avançados da Aplicação	3-22
4. Configuração do Nexto Xpress	4-1
Configuração da UCP do Controlador	4-1
Parâmetros Gerais	4-1
Sincronização Temporal	4-1
Pontos Internos	4-3
Configuração das Interfaces	4-6
Configuração da Interface Serial (COM 1)	4-6
Configuração da Interface Ethernet (NET 1)	4-6
Configuração da Interface CAN (Controller Area Network)	4-6
Entradas e Saídas Integradas	4-6
Porta USB	4-19
Protocolos de Comunicação	4-29
MODBUS RTU	4-29
MODBUS Ethernet	4-33
Estudo Dirigido 4-1: configuração do protocolo MODBUS Ethernet	4-40
Estudo Dirigido 4-2: aplicação do protocolo MODBUS Ethernet	4-44
Servidor OPC DA	4-45
OPC UA Servidor	4-52
Estudo Dirigido 4-3: configurando o Nexto Xpress como servidor OPC UA	4-59
EtherNet/IP	4-60
CANOpen Manager	4-66
Modo de E/S Remotas	4-70
SNMP	4-74
PROFINET	4-76
Relógio RTC	4-76
Blocos Funcionais Especiais para Comunicação Serial	4-78
Atualização de Entradas e Saídas	4-85
Temporizador Retentivo	4-86
Servidor FTP	4-88
Firewall	4-89
OpenVPN	4-93
5. Funcionalidades do Nexto Xpress	5-1
Bloco Funcional PID	5-6
Entradas do Bloco Funcional	5-6
Saídas do Bloco Funcional	5-7
Considerações Gerais sobre a Programação do Bloco Funcional PID	5-7

PID_fixcycle	5-7
PID Control	5-8
Ambiente Gráfico	5-8
Estudo Dirigido 5-1: utilização do PID Control	5-8
Blocos Funcionais de Controle Avançado	5-21
Aplicação do Bloco Funcional PIDA_REAL	5-21
Aplicação do Bloco Funcional PIDA_INT	5-26
Aplicação do Bloco Funcional PIDA_TUNE_REAL	5-27
Aplicação do Bloco Funcional PIDA_TUNE_INT	5-32
Bloco Funcional Datalogger	5-36
Modos de registro dos dados	5-36
Características Gerais	5-36
Layout do arquivo gerado	5-39
Estrutura do diretório	5-39
Nome do arquivo	5-40
Intervalo de valores para o formato de saída	5-40
Funcionamento geral no modo Contínuo	5-40
Funcionamento geral no modo Circular	5-42
Funcionamento geral no modo Trigger	5-42
Motion Control	5-44
Um pouco de teoria... Aplicações de controle de movimento	5-44
Alteração de IP via Aplicação ou Página Web	5-46
Alteração de IP via Aplicação	5-46
Alteração de IP via Página Web	5-49
Conexão do CP com Banco de Dados SQL	5-51
Parâmetros de Entrada do Bloco Funcional	5-51
Parâmetros de Saída do Bloco Funcional	5-51
Escrevendo em um banco de dados SQL	5-53
Visualização	5-56
Visão Geral	5-56
Comandos de Visualização	5-59
Editor de Visualização	5-60
Gerenciamento de Visualização com Clientes	5-65
Visualização em Modo Online	5-65
Estudo Dirigido 5-2: criação de uma visualização web	5-65
6. Aplicações, Avaliação e Encerramento	6-1
Automação e controle de processo com o Nexto Xpress	6-1
Automação do Processo de Preparação de Solução de NaCl	6-1
Descrição do processo	6-1
Detalhamento do processo	6-1
Esquematização do Processo	6-2
Situação de Aprendizagem Complementar 6-1: armazenamento dos dados da aplicação de Preparação de Solução de NaCl (DATALOGGER)	6-4
Um pouco de teoria... Conheça os nove pilares da Indústria 4.0 e sua relevância para a atividade industrial	6-7
Situação de Aprendizagem Complementar 6-2: o Nexto Xpress e a Indústria 4.0	6-9
Situação de Aprendizagem Complementar 6-3: utilização do protocolo MQTT	6-10
Situação de Aprendizagem Complementar 6-4: implementação de uma simulação de uma aplicação de controle PID com auto sintonia	6-15
Avaliação do Treinamento	6-17
Parabéns...	6-17
Glossário	A

Figuras

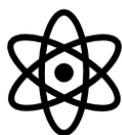
Figura 1-1. O CP Nexto Xpress	1-1
Figura 1-2. Ciclo típico de um controlador programável	1-5
Figura 1-3. Movendo as travas para a posição aberta (1)	1-6
Figura 1-4. Posicionando o Controlador no Trilho DIN	1-6
Figura 1-5. Travando o Controlador no Trilho DIN	1-7
Figura 1-6. Destravando o Controlador no Trilho DIN	1-7
Figura 1-7. Diagrama de Instalação Elétrica do XP3xx	1-8
Figura 1-8. Exemplo de informações do controlador disponíveis na tela inicial	1-13
Figura 2-1. Ambiente de programação MasterTool IEC XE	2-1
Figura 2-2. Janela dos POU's	2-5
Figura 2-3. Árvore de Dispositivos	2-6
Figura 2-4. Configuração da UCP	2-9
Figura 2-5. Configurando a Porta de Comunicação	2-10
Figura 2-6. Configurando as E/S Integradas	2-11
Figura 2-7. Adicionando a Instância	2-12
Figura 2-8. Selecionando o Protocolo	2-13
Figura 2-9. Localizando o Dispositivo	2-14
Figura 2-10. Selecionando o Controlador	2-15
Figura 2-11. Easy Connection	2-16
Figura 2-12. Enviando o Projeto para o controlador	2-17
Figura 2-13. Aviso de configuração IP	2-17
Figura 2-14. Sem aplicação no dispositivo	2-17
Figura 2-15. Download de Nova Aplicação	2-18
Figura 2-16. Alteração de Configuração	2-18
Figura 2-17. Download de código fonte	2-19
Figura 2-18. Iniciando a Aplicação	2-20
Figura 2-19. Programa em Execução	2-21
Figura 2-20. Parando a Aplicação	2-22
Figura 2-21. Encerrando a comunicação online com o controlador	2-23
Figura 2-22. Opção de Upload de Projeto	2-24
Figura 2-23. Selecionando o controlador	2-25
Figura 2-24. GVL IntegratedIO quando em modo Online	2-28
Figura 2-25. GVL System_Diagnostics em Modo Online	2-29
Figura 2-26. GVL Disable em Modo Online	2-30
Figura 2-27. GVL Qualities em Modo Online	2-32
Figura 2-28. GVL ReqDiagnostics em Modo Online	2-34
Figura 2-29. Exemplo do MasterTool IEC XE Interface do Usuário	2-36
Figura 2-30. Linha Pontilhada no Menu e Barras de Ferramentas	2-38
Figura 2-31. Diálogo de Estado Online	2-39
Figura 2-32. Barra de Menu Padrão	2-40
Figura 2-33. Utilização da Memória de Arquivos de Usuário	2-40
Figura 2-34. Editores	2-41
Figura 2-35. Novo Projeto	2-42
Figura 2-36. Classificação do Projeto	2-43
Figura 2-37. Seleção dos Módulos de Hardware	2-44
Figura 2-38. Seleção do Perfil	2-45
Figura 2-39. Inserindo POU's	2-46
Figura 2-40. Classificando a POU	2-47
Figura 2-41. Editando a POU	2-48
Figura 2-42. Criando uma Tarefa	2-49
Figura 2-43. Nomeando a Tarefa	2-50

Figura 2-44. Configurando a Tarefa Criada	2-51
Figura 2-45. Vinculando POUs às Tarefas	2-52
Figura 2-46. Inclusão de uma Biblioteca no Projeto	2-53
Figura 2-47. Adicionando a Instância	2-54
Figura 2-48. Seleção do protocolo MODBUS RTU Slave	2-55
Figura 2-49. Adicionando a Instância	2-56
Figura 2-50. Selecionando o Protocolo	2-57
Figura 2-51. Selecionando o Protocolo MODBUS Ethernet	2-58
Figura 2-52. Compilando o Projeto	2-59
Figura 2-53. Mensagens da Compilação	2-60
Figura 2-54. Incluindo as Mensagens na Tela	2-60
Figura 2-55. Modo Simulação	2-61
Figura 2-56. Janela do Editor ST	2-64
Figura 2-57. Diálogo Autodeclarar	2-65
Figura 2-58. Declaração da Variável “erg”	2-65
Figura 2-59. Expressões de Monitoração e Linhas de Código	2-67
Figura 2-60. Lista de Monitoração	2-68
Figura 2-61. Exibição do Valor Atual	2-68
Figura 2-62. Aplicação em “Stop”	2-69
Figura 2-63. Aplicação no Modo “Run”	2-69
Figura 3-1. Instância “Inst1”	3-2
Figura 3-2. Rede LD	3-7
Figura 3-3. Comandos Editores FBD/LD/ID	3-8
Figura 3-4. Caixa de Ferramentas	3-8
Figura 3-5. Título, Comentário e Rótulo	3-9
Figura 3-6. Contato	3-9
Figura 3-7. Contato Negado	3-9
Figura 3-8. Bobina	3-10
Figura 3-9. Bobina Negada	3-10
Figura 3-10. Bobinas Set e Reset	3-10
Figura 3-11. Visualização Online do Editor LD	3-11
Figura 3-12. Diálogo Preparar Valor	3-11
Figura 3-13. Redes de Diagrama de Blocos Funcionais	3-17
Figura 3-14. Exemplo para uma Sequência de Passos em um Módulo SFC	3-18
Figura 3-15. Propriedade Segundos Acrescentada ao Bloco Funcional fb	3-22
Figura 3-16. Ação em um Bloco Funcional	3-23
Figura 4-1. Configuração da UCP	4-1
Figura 4-2. Configuração Sntp	4-2
Figura 4-3. Tela de Configuração dos Pontos Internos	4-4
Figura 4-4. Exemplo de Configuração dos Pontos Internos	4-4
Figura 4-5. E/S Integradas na árvore de projeto	4-7
Figura 4-6. Parâmetros de Entradas Digitais	4-8
Figura 4-7. Configuração de Entradas Rápidas	4-9
Figura 4-8. Estrutura dos contadores	4-10
Figura 4-9. Bloco Funcional LibIntegratedIo.COUNTER	4-11
Figura 4-10. Configurações da interrupção do contador	4-11
Figura 4-11. Configurações da Interrupção do Contador	4-12
Figura 4-12. Configurações das Interrupções de Entradas Rápidas	4-13
Figura 4-13. Parâmetros das Saídas Rápidas	4-13
Figura 4-14. Estrutura das Saídas Rápidas	4-14
Figura 4-15. Forma de Onda VFO/PWM	4-14
Figura 4-16. Bloco Funcional LibIntegratedIo.PWM	4-14
Figura 4-17. PTO com perfil trapezoidal	4-15
Figura 4-18. PTO com perfil “S”	4-15
Figura 4-19. Parada suave da PTO no perfil trapezoidal	4-16
Figura 4-20. Parada suave da PTO no perfil S	4-16
Figura 4-21. Bloco Funcional LibIntegratedIo.PTO	4-17

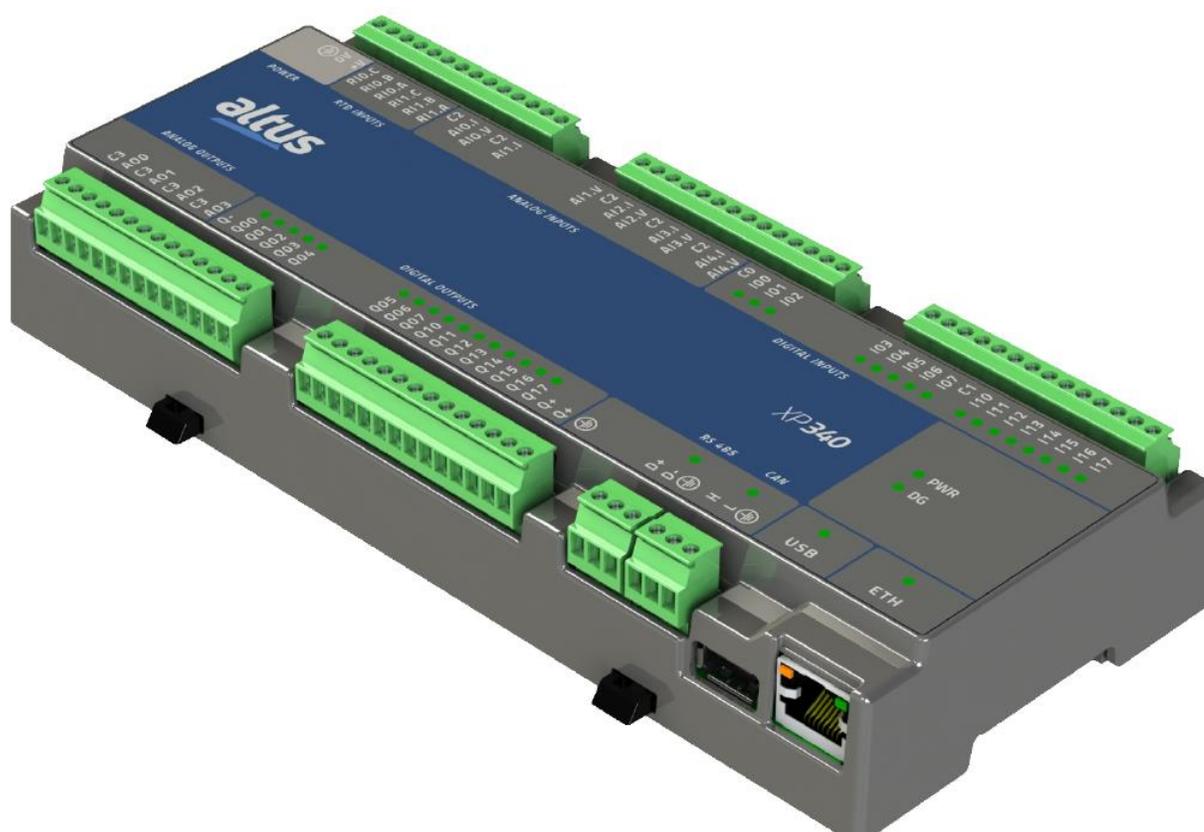
Figura 4-22. Aplicação típica das entradas/saídas rápidas	4-17
Figura 4-23. Parâmetros das Entradas Analógicas	4-18
Figura 4-24. Parâmetros das entradas RTD	4-18
Figura 4-25. Parâmetros das Saídas Analógicas	4-18
Figura 4-26. Mapeamento de tags para E/S	4-19
Figura 4-27. Seção de dispositivos USB	4-20
Figura 4-28. Dispositivos USB - Desconhecido	4-20
Figura 4-29. Informações sobre o dispositivo de armazenamento em massa	4-20
Figura 4-30. Pasta de armazenamento em massa USB	4-21
Figura 4-31. DontBootApplication no dispositivo de armazenamento em massa	4-21
Figura 4-32. Aplicação no dispositivo de armazenamento em massa	4-22
Figura 4-33. Dispositivos USB - Conversor Serial	4-23
Figura 4-34. Configuração da Página do Modem USB	4-24
Figura 4-35. Configurando o Gateway com Endereço IP do Modem	4-25
Figura 4-36. Página do adaptador USB WiFi	4-26
Figura 4-37. Configuração do adaptador USB WiFi	4-27
Figura 4-38. Adaptador USB WiFi conectado a uma rede	4-28
Figura 4-39. Adicionando a Instância MODBUS	4-30
Figura 4-40. Seleção do protocolo MODBUS RTU Master	4-31
Figura 4-41. Tela de Configuração Parâmetros Gerais MODBUS RTU Mestre	4-31
Figura 4-42. Tela de Configurações dos Parâmetros Gerais do Dispositivo	4-32
Figura 4-43. Tela de Mapeamentos de Dados MODBUS	4-32
Figura 4-44. Tela de Requisições de dados MODBUS Mestre	4-32
Figura 4-45. Configuração do Escravo	4-33
Figura 4-46. Tela de Mapeamentos de dados MODBUS	4-33
Figura 4-47. Rede de Comunicação MODBUS TCP	4-34
Figura 4-48. Adicionando a Instância MODBUS Ethernet	4-36
Figura 4-49. Selecionando o Protocolo MODBUS Ethernet	4-37
Figura 4-50. Tela de Configuração Parâmetros Gerais MODBUS Cliente	4-37
Figura 4-51. Tela de Configurações dos Parâmetros Gerais do Dispositivo MODBUS	4-38
Figura 4-52. Tela de Mapeamentos de dados MODBUS Cliente	4-38
Figura 4-53. Tela de Requisições de dados MODBUS	4-39
Figura 4-54. Tela de Configuração Parâmetros Gerais MODBUS Servidor	4-39
Figura 4-55. Tela de Configuração MODBUS Servidor	4-40
Figura 4-56. Arquitetura OPC DA	4-46
Figura 4-57. Objeto Symbol Configuration	4-47
Figura 4-58. Selecionando Variáveis na Symbol Configuration	4-48
Figura 4-59. Configuração do OPC DA Servidor	4-49
Figura 4-60. Selecionando Servidor OPC DA na Configuração do Cliente	4-51
Figura 4-61. Lista de Símbolos Consultados pelo Cliente OPC DA	4-52
Figura 4-62. Arquitetura Típica OPC UA	4-53
Figura 4-63. Objeto Symbol Configuration (OPC UA)	4-55
Figura 4-64. Habilitando OPC UA no Objeto Symbol Configuration	4-55
Figura 4-65. Possíveis vulnerabilidades de um típico sistema de automação	4-56
Figura 4-66. Selecionando Servidor OPC UA na Configuração do Cliente	4-58
Figura 4-67. Lista de Símbolos Consultados pelo Cliente OPC UA	4-59
Figura 4-68. Adicionando uma Interface EtherNet/IP	4-61
Figura 4-69. Adicionando um Adapter ou Scanner EtherNet/IP	4-62
Figura 4-70. Adicionando um Adapter EtherNet/IP sob o Scanner	4-63
Figura 4-71. Tela de Parâmetros Gerais do Servidor IEC 60870-5-104	4-65
Figura 4-72. Tela de Mapeamento dos Dados do Servidor IEC 60870-5-104	4-65
Figura 4-73. Tela de Configuração da Camada de Enlace do Servidor IEC 60870-5-104	4-65
Figura 4-74. Tela de Configuração da Camada de Aplicação do Servidor IEC 60870-5-104	4-66
Figura 4-75. Adicionando o CANopen Manager	4-67
Figura 4-76. Adicionando um Dispositivo CANopen Escravo	4-68
Figura 4-77. Parâmetros gerais do CANopen Manager	4-68
Figura 4-78. Configuração da tarefa cíclica do barramento CANopen	4-69

Figura 4-79. Parâmetros de Escravo CANopen	4-70
Figura 4-80. Tela de Configuração Modo de Operação	4-70
Figura 4-81. Tela de Configuração do Escravo CANopen	4-71
Figura 4-82. Tela de Configuração de E/S Expandida	4-72
Figura 4-83. Modo de Operação na Tela de Informações do CP	4-72
Figura 4-84. Escravo CANopen em Operational - RUN	4-73
Figura 4-85. Escravo CANopen em Pre-Operational - STOP	4-73
Figura 4-86. Exemplo de Gerente SNMP	4-74
Figura 4-87. Tela de Login - SNMP	4-75
Figura 4-88. Tela de Configuração de Estados SNMP	4-75
Figura 4-89. Página de Configuração do Servidor FTP	4-89
Figura 4-90. Página de Configuração do Firewall	4-90
Figura 4-91. Tabela de Aplicação de Regras de Usuário do Firewall	4-92
Figura 4-92. Página de Configuração do OpenVPN	4-93
Figura 4-93. Opções de Configuração do OpenVPN (Servidor)	4-94
Figura 4-94. Opções de Configuração do OpenVPN (Cliente)	4-94
Figura 4-95. Tabela de Configurações de Redes Privadas OpenVPN	4-96
Figura 4-96. Exemplo de Arquitetura com Redes Privadas	4-96
Figura 4-97. Tabela de Arquivos de Segurança OpenVPN	4-98
Figura 4-98. Tabela de Status OpenVPN com Funcionalidade Desativada	4-98
Figura 4-99. Tabela de Status OpenVPN com Funcionalidade Ativada	4-99
Figura 4-100. Seção de Downloads OpenVPN	4-99
Figura 4-101. Exemplo de arquitetura Host-to-Host	4-100
Figura 4-102. Exemplo de arquitetura Host-to-Site	4-100
Figura 4-103. Exemplo de arquitetura Site-to-Site	4-101
Figura 5-1. Papel do Controlador em Sistemas de Controle Industriais	5-1
Figura 5-2. Malha de Controle	5-2
Figura 5-3. Sistema de Controle com Realimentação	5-3
Figura 5-4. Controle PID para Temperatura	5-4
Figura 5-5. Resposta típica ao degrau	5-5
Figura 5-6. Detalhe do período crítico (Pu)	5-5
Figura 5-7. Bloco Funcional PID	5-6
Figura 5-8. Aba Configurações Avançadas	5-16
Figura 5-9. Bloco Funcional PIDA_REAL	5-22
Figura 5-10. Ilustração da técnica de sintonização automática	5-28
Figura 5-11. Exemplo de arquitetura de controle de movimento	5-45
Figura 5-12. Tabela de Interfaces - Modo Aplicação	5-46
Figura 5-13. Seleção da opção de inserção da biblioteca no gerenciador	5-47
Figura 5-14. Seleção da opção "Avançado" no gerenciador	5-47
Figura 5-15. Tabela de Interfaces - Modo Web	5-49
Figura 5-16. Biblioteca LibSQLClient e bloco funcional MSSQL_CLIENT	5-51
Figura 5-17. Conexão com o servidor do banco de dados	5-53
Figura 5-18. Seleção de nova consulta ao banco de dados	5-53
Figura 5-19. Criação da tabela do banco de dados	5-54
Figura 5-20. Código para a nova consulta	5-55
Figura 5-21. Nnn	5-55
Figura 5-22. Mecanismo Geral	5-57
Figura 5-23. Servidor de Dados para fornecer dados de visualização	5-57
Figura 5-24. Cliente de visualização WebVisu	5-58
Figura 5-25. Caixa de diálogo de Configuração de Frame	5-59
Figura 5-26. Editor de Visualização	5-61
Figura 5-27. Diálogo 'Propriedades', Categoria 'Visualização'	5-63
Figura 6-1. Processo de Preparação de Solução de NaCl	6-3

Introdução



Este Tutorial fornece suporte didático-pedagógico ao treinamento presencial do controlador programável Nexto Xpress, possibilitando ao usuário um primeiro contato orientado com o equipamento e enfatizando a programação do mesmo via programador MasterTool IEC XE. O documento inclui ainda considerações sobre instalação do CP, conexões elétricas e configurações básicas. Esta seção contém as divisões do tutorial e as simbologias e convenções utilizadas. O objetivo é informar ao usuário como o documento pode ser consultado.



O Curso MP11



O curso MP11 apresenta o controlador programável Nexto Xpress enfatizando seus diferenciais no que tange a um avançado sistema de automação criado para atender às demandas de máquinas e pequenos processos. O CP compacto Nexto Xpress é uma solução completa para as aplicações mais avançadas, entregando desempenho e alta conectividade com um excelente custo benefício.

Os objetivos desse curso são:

- Fazer conhecer as principais características, requisitos de instalação e as estratégias de manutenção (preventiva e corretiva) com ênfase nos diagnósticos associados;
- Apresentar a sequência de tarefas envolvidas no planejamento de um projeto de automação para aplicação no dispositivo através da utilização do ambiente de programação MasterTool IEC XE;
- Estabelecer considerações sobre sistemas de controle industriais, definir os conceitos básicos da norma IEC 61131-3, descrever os itens comuns previsto no modelo de software, bem como apresentar as linguagens de programação disponíveis;
- Explorar as funcionalidades de rede do dispositivo através da descrição dos canais de comunicação disponíveis e detalhar os protocolos de comunicação;
- Apresentar alguns recursos interessantes e inovadores, assim como funcionalidades avançadas do dispositivo, enfatizando a utilização do bloco funcional PID;
- Utilizar o software de programação e supervisão MasterTool IEC XE, ressaltando sua programação gráfica e amigável, com recursos de edição avançados e prática de programação em 5 linguagens gráficas e textuais de programação, compatíveis com a norma IEC 61131-3.

Esse curso é destinado aos iniciantes em automação, clientes da Altus, potenciais usuários, profissionais técnicos e engenheiros de áreas afins, assim como todos os interessados em adquirir conhecimentos sobre automação. Ele tem uma duração de 24 horas e demanda como pré-requisito, a participação prévia do aluno no Curso MP1 — Introdução à Automação.

O projeto didático do curso é concebido de forma a embasar o ensino direto, utilizando metodologias que propiciam a integração entre a teoria e a prática e favorecem a capacidade de construção e gestão do conhecimento e o autodesenvolvimento contínuo. Os recursos didáticos associados (Tutorial, Slides e Documentação de Produto) contemplam elementos de instrução de alta qualidade pedagógica e suficientemente compreensíveis, dinâmicos e atrativos. No seu contexto está prevista a flexibilização do ensino para o desenvolvimento de habilidades de autoaprendizagem.

Estruturação do Tutorial



O Tutorial fornece um suporte didático-pedagógico ao treinamento presencial, possibilitando ao aluno um primeiro contato orientado com o equipamento, enfatizando a sua programação. O documento inclui ainda considerações sobre instalação, conexões elétricas e configurações. Ele é dividido em capítulos numa estrutura padronizada conforme esquema mostrado a seguir.

- ▷ 1. Hardware do Nexto Xpress
- ▷ 2. Software do Nexto Xpress
- ▷ 3. Programação do Nexto Xpress
- ▷ 4. Configuração do Nexto Xpress
- ▷ 5. Funcionalidades do Nexto Xpress
- ▷ 6. Aplicações, Avaliação e Encerramento

Alguns capítulos contêm Estudos Dirigidos cuja implementação permitirá ao aluno consolidar os conceitos abordados no Tutorial, aplicando-os em situações-problema. Os estudos dirigidos são exercícios teórico/práticos de aprofundamento e aplicação dos temas já tratados. O objetivo é que os alunos resolvam as tarefas propostas de modo relativamente independente, a partir das diretrizes definidas pelo Instrutor.

Outra estratégia de aprendizado prevista nos cursos é a pesquisa na Documentação de Produto, permitindo ao aluno uma familiarização com o acesso a um vasto acervo de manuais, características técnicas, notas de aplicação etc.

EMENTA DO CURSO

Capítulo 1: Hardware do Nexto Xpress

Este capítulo descreve o hardware do Nexto Xpress. Informa suas principais características, diferenciais e características do produto. Apresenta também a instalação mecânica, elétrica e de rede. E, por último, aborda os diagnósticos disponíveis como suporte à manutenção (corretiva e preventiva). Tópicos abordados:

- ▲ 1. Hardware do Nexto Xpress
 - ▷ Características do Nexto Xpress
 - ▷ Instalação do Nexto Xpress
 - ▷ Manutenção do Nexto Xpress

Capítulo 2: Software do Nexto Xpress

O capítulo versa sobre o software de programação MasterTool IEC XE e sua utilização. São fornecidas orientações sobre as configurações básicas para o seu correto funcionamento. O capítulo apresenta a sequência básica de tarefas envolvidas no planejamento de um projeto de automação para aplicação no dispositivo. São realizadas algumas considerações sobre sistemas de controle industriais e apresentados os conceitos básicos da norma IEC 61131-3. Além disso, são descritos os itens comuns às linguagens de programação e introduzidos os conceitos de Configurações, Recursos, Programas, Tarefas, Blocos funcionais, Funções, Variáveis, Caminhos de acesso, POUs e Forma hierárquica. Tópicos abordados:

- ▲ 2. Software do Nexto Xpress
 - ▷ Características do MasterTool IEC XE
 - ▷ Conceitos Básicos Associados aos Ambientes de Programação IEC61131
 - ▷ Funcionalidades do MasterTool IEC XE
 - ▷ Interface do Usuário do MasterTool IEC XE
 - ▷ Criação e Execução de Projetos
 - ▷ Ajuda

Capítulo 3: Programação do Nexto Xpress

Este capítulo aborda a programação do dispositivo apresentando os elementos de uma aplicação e a criação de unidades de organização de programas. Ele explora ainda as linguagens de programação IEC61131-3. Tópicos abordados:

- ▲ 3. Programação do Nexto Xpress
 - ▷ Elementos da Aplicação
 - ▷ Criação de POUs
 - ▷ Linguagens de Programação IEC61131-3

Capítulo 4: Configuração do Nexto Xpress

Este capítulo detalha as configurações da unidade central de processamento do controlador, bem como suas interfaces e protocolos de comunicação. Tópicos abordados:

- ▲ 4. Configuração do Nexto Xpress
 - ▷ Configuração da UCP do Controlador
 - ▷ Configuração das Interfaces
 - ▷ Protocolos de Comunicação

Capítulo 5: Funcionalidades do Nexto Xpress

Este capítulo inclui a análise de algumas funcionalidades avançadas do dispositivo no que se refere às aplicações industriais, enfatizando a utilização do mesmo em sistemas de controle e explorando as potencialidades da visualização web. Tópicos abordados:

- ▲ 5. Funcionalidades do Nexto Xpress
 - ▷ Bloco Funcional PID
 - ▷ PID Control
 - ▷ Blocos Funcionais de Controle Avançado
 - ▷ Bloco Funcional Datalogger
 - ▷ Motion Control
 - ▷ Alteração de IP via Aplicação ou Página Web
 - ▷ Conexão do CP com Banco de Dados SQL
 - ▷ Visualização

Capítulo 6: Aplicações, Avaliação e Encerramento

O tutorial finaliza com a proposição de estudos de caso para consolidação das competências desenvolvidas no treinamento. Tópicos abordados:

- ▲ 6. Aplicações, Avaliação e Encerramento
 - ▷ Automação e controle de processo com o Nexto Xpress
 - ▷ Avaliação do Treinamento

Referências



1. Fonseca, M. Apostila de suporte de curso: Norma IEC 61131-3 para Programação de Controladores. ISA Distrito 4 – América do Sul, 2008.
2. INTERNET (Sites diversos).
3. Documentos relacionados à Série Nexto Xpress (conforme indicado na sequência)

Documentos relacionados

Para obter informações adicionais sobre o Nexto Xpress podem ser consultados outros documentos (manuais e características técnicas) além deste. Estes documentos encontram-se disponíveis em www.altus.com.br. Cada produto possui um documento denominado Características Técnicas (CT), e neste documento encontram-se as características do produto

em questão. Caso o produto possua mais informações, ele pode ter também um manual de utilização. Por exemplo, o módulo NX2020 tem todas as informações de características, utilização e de compra, na sua CT. Por outro lado, o NX5001 possui, além da CT, um manual de utilização. A tabela a seguir indica a lista de todos os documentos relacionados à Série Nexto Xpress.

Código	Descrição	Idioma
CE114000	Nexto Series – Technical Characteristics	Inglês
CT114000	Série Nexto – Características Técnicas	Português
CS114000	Serie Nexto – Especificaciones y Configuraciones	Espanhol
MU216600	Nexto Xpress User Manual	Inglês
MU216000	Manual de Utilização Nexto Xpress	Português
MU214600	Nexto Series User Manual	Inglês
MU214000	Manual de Utilização Série Nexto	Português
MU214605	Nexto Series CPUs User Manual	Inglês
MU214100	Manual de Utilização UCPs Série Nexto	Português
MU299609	MasterTool IEC XE User Manual	Inglês
MU299048	Manual de Utilização MasterTool IEC XE	Português
MP399609	MasterTool IEC XE Programming Manual	Inglês
MP399048	Manual de Programação MasterTool IEC XE	Português
MU214606	MQTT User Manual	Inglês
MU223603	IEC 60870-5-104 Server Device Profile	Inglês
NAP151	Utilização do Tunneller OPC	Português
NAP165	Comunicação OPC UA com Controladores ALTUS	Português
NAP165_ing	OPC UA Communication with ALTUS Controllers	Inglês

Avaliação do Curso



A avaliação continuada e individualizada é uma prática corrente adotada pelo Instrutor no curso. Desta forma, as seguintes estratégias de avaliação são consideradas: acompanhamento do desenvolvimento do aluno (frequência, participação, postura, interesse e construção do conhecimento) durante o curso pelo Suporte Técnico e autoavaliação via resolução das tarefas propostas caracterizadas pelos Estudos Dirigidos, Pesquisa na Documentação e Aplicações.

Terminologia



Neste Tutorial, as palavras “software” e “hardware” são empregados livremente, por sua generalidade e frequência de uso. Por este motivo, apesar de serem vocábulos em inglês, aparecerão no texto sem aspas.

As seguintes expressões podem ser empregadas no texto da Tutorial.

CP: Controlador Programável - equipamento composto por uma UCP, módulos de entrada e saída e fonte de alimentação. É também abreviado como **CLP** (Controlador Lógico Programável).

UCP ou CPU: Unidade Central de Processamento - módulo principal do CP, que realiza o processamento dos dados.

MasterTool IEC XE: identifica o programa para microcomputador executável em ambiente WINDOWS®, que permite o desenvolvimento de aplicativos para os CPs da Série Nexto.

Convenções Utilizadas



Os símbolos utilizados ao longo deste manual possuem os seguintes significados:

- Este marcador indica uma lista de itens ou tópicos.

MAIÚSCULAS PEQUENAS indicam nomes de teclas, por exemplo, ENTER.

TECLA1+TECLA2 é usado para teclas a serem pressionadas simultaneamente. Por exemplo, a digitação simultânea das teclas CTRL e END é indicada como CTRL+END.

TECLA1, TECLA2 é usado para teclas a serem pressionadas sequencialmente. Por exemplo, a mensagem “Digite ALT, F10” significa que a tecla ALT deve ser pressionada e liberada e então a tecla F10 pressionada e liberada.

Maiúsculas GRANDES indicam nomes de arquivos e diretórios.

Itálico indica palavras e caracteres que são digitados no teclado ou vistos na tela. Por exemplo, se for solicitado a digitar *FACAO*, estes caracteres devem ser digitados exatamente como aparecem no manual.

NEGRITO é usado para nomes de comandos ou opções, ou para enfatizar partes importantes do texto.

As mensagens de advertência apresentam 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.

Suporte Técnico



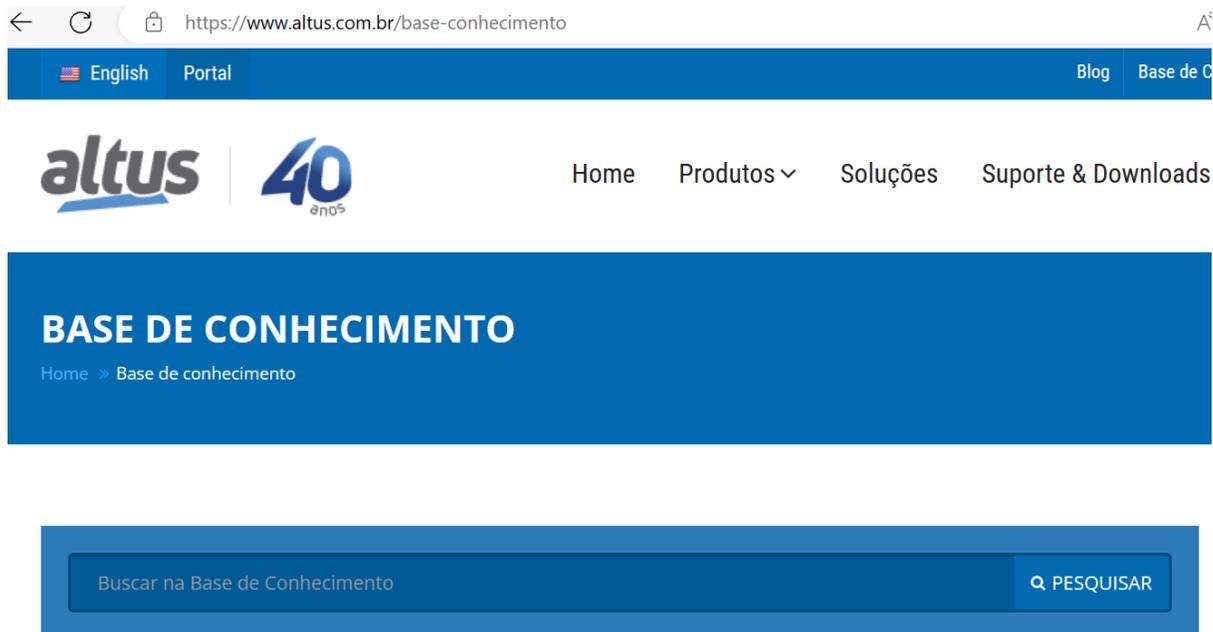
Formada por especialistas experientes e com grande conhecimento, a equipe de Suporte Técnico da ALTUS é altamente capacitada para prestar assistência técnica, preventiva e corretiva, para todos os produtos comercializados pela empresa.

Para mais informações acesse: <https://www.altus.com.br/suporte>.

Base de Conhecimento ALTUS



No site ALTUS você encontra também vários tutoriais que auxiliam na implementação de aplicações específicas no espaço conhecido como Base de Conhecimento ALTUS (<https://www.altus.com.br/base-conhecimento>). Isso inclui: conexões, instalações, utilização de módulos função e configurações diversas...



Cursos ALTUS



A Altus possui um calendário de treinamentos anual para profissionais da área de automação industrial interessados em desenvolver aplicações, realizar reformas e prestar manutenção em sistemas de controle de processos. Os cursos, que abordam as linhas de produtos da empresa, têm como objetivo introduzir conceitos sobre automação industrial e capacitar os participantes a desenvolverem aplicações para controle de processos. Também são realizados cursos especiais e IN COMPANY, de acordo com a solicitação e necessidade dos clientes. Confira as opções de treinamento disponíveis no site ALTUS.

Revisões deste Tutorial



O código de referência, a revisão e a data do presente documento estão indicadas na capa. A mudança da revisão pode significar alterações da especificação funcional ou melhorias no mesmo. A seguir as alterações correspondentes a cada revisão deste:

Revisão	Data	Descrição
A	Novembro/20	Emissão do Documento
B	Novembro/23	Revisão do Documento

1. Hardware do Nexto Xpress



Características do Nexto Xpress



O que é o Nexto Xpress?

O Nexto Xpress é um poderoso Controlador Programável pertencente à família de controladores e módulos de E/S da Série Nexto. O Nexto Xpress fornece poder de processamento de alta velocidade em um design compacto com E/S integradas. Existem várias opções para escolher, permitindo a melhor solução para aplicações básicas. Esse portfólio de produtos visa sistemas de controle pequenos, oferecendo modelos contendo desde algumas entradas e saídas digitais até opções com 43 pontos de E/S concentrados em um único controlador, incluindo entradas e saídas analógicas com suporte de temperatura (sensores RTD). No caso de necessidades adicionais de E/S, o sistema pode ser facilmente expandido através dos módulos de expansão. Adicionalmente, a quantidade de pontos de E/S pode ser ampliada ainda mais através dispositivos de E/S remotos (distribuído) comunicando-se através de protocolos como CANopen, EtherNet/IP, PROFINET e MODBUS.

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

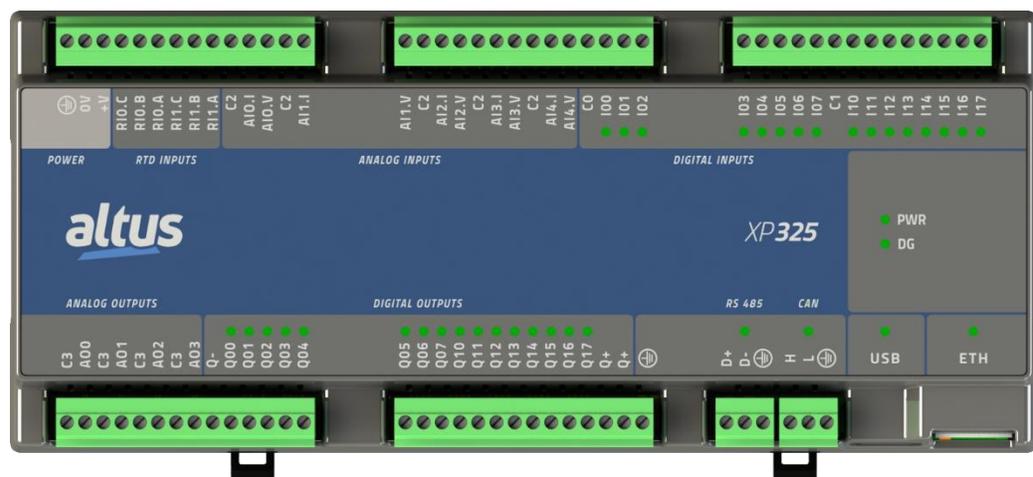


Figura 1-1. O CP Nexto Xpress

Porque o Nexto Xpress?

Criados para atender às demandas da Indústria 4.0, os controladores programáveis vêm em cinco opções de modelos que se diferenciam em número de pontos e recursos: XP300, XP315, XP325, XP340 e XP350. Todos contam com 16 entradas digitais, 16 saídas digitais a transistor, uma porta Ethernet, um canal Serial padrão RS-485, uma porta CAN e uma interface USB.

O CLP XP315 ainda possui 2 entradas analógicas RTD de dois e três fios e 5 entradas analógicas para tensão e corrente.

Os modelos XP325 e XP340 contam com todos estes recursos acrescidos de 4 saídas analógicas, com o diferencial que o CLP XP340 ainda possui suporte a WebServer, ferramenta que permite a criação de telas de supervisão e monitoramento de aplicação sem a necessidade de usar um sistema SCADA.

Já o CLP XP350 conta com os mesmos recursos disponíveis no XP315, acrescidos de ferramentas de Motion Control, como suporte a blocos de função PLCopen Motion Control Part 1 para comando de eixo único, sincronização de multi-eixos, engrenagem eletrônica (CAME), editor especial para planejar movimentos (CAM), entre outros.

Assim como os demais membros da família Nexto, o CLP Xpress também utiliza a plataforma tecnológica CODESYS com configuração via MasterTool IEC XE, software GRATUITO de programação da série com escalabilidade para projetos de micro a grande porte.



Versátil	Compacto	Veloz	Flexível
Recursos variados para aplicações de saneamento, predial, têxtil, alimentício, transporte e em máquinas	Design maciço (212,5 x 90,1 x 32,2 mm) para utilização em espaços extremamente reduzidos	Tempo de atualização inferior a um milissegundo, um dos mais rápidos do mercado	Variadas opções de interfaces de comunicação e suporte aos principais protocolos Ethernet e Fieldbus

Software de programação MasterTool IEC XE baseado em CODESYS	Módulos específicos para expansão de Entradas e Saídas	Produto em conformidade com a marcação CE e as diretivas ROHS	Conformal coating para que possa ser utilizada em ambientes severos

Quais São Seus Diferenciais?

Coleta de dados e integração de processos



Projetada para atender às demandas de ambientes altamente conectados, a solução conta com suporte a uma ampla variedade de protocolos de comunicação, o que lhe permite interagir com dispositivos inteligentes, como conversores de mídia, leitores de código de barra, modems de comunicação sem fio e dispositivos de armazenamento de dados, através da interface USB. Com interfaces de comunicação Ethernet, Serial RS 485, CAN e USB embarcadas, os CLPs Xpress ainda contam com suporte a uma variada lista de protocolos de comunicação, como MODBUS TCP e RTU, PROFINET, CANopen e low level, IEC 60870-5-104, EtherNet/IP, OPC DA, OPC UA e MQTT.

Os recursos também permitem ao controlador comunicar diretamente com os serviços de hospedagem na nuvem mais relevantes do mercado, como Microsoft Azure, IBM Cloud, Google One e Amazon AWS.

O modelo XP340 ainda conta com suporte a WebServer, ferramenta que permite a criação de telas de supervisão e monitoramento de aplicação sem a necessidade de usar um sistema SCADA. Embarcado na memória do controlador programável, o recurso pode ser acessado através do navegador web de qualquer dispositivo (computador, tablet ou smartphone) conectado à rede.

Perfeito para máquinas e pequenas aplicações

Com design compacto (212,5 x 90,1 x 32,2 mm) e alimentação 24 Vdc, a solução Xpress é ideal para o controle embarcado de máquinas e aplicações de pequeno porte.

Seu processador ARM 32 bits, aliado ao relógio de tempo real (RTC) e a alta densidade de pontos de E/S digitais e analógicos de múltiplo propósito (tensão, corrente e temperatura) dão à solução uma elevada capacidade de processamento, precisão e agilidade na aquisição de dados.

Estas características colocam os equipamentos como a solução perfeita para dar mais velocidade e economia no controle das mais variadas máquinas do mercado, como rotuladoras, sopradoras, lavadoras, injetoras, ensacadoras, moedores, prensas, entre outras.

Os CLPs ainda têm LEDs para indicação de estado e diagnósticos, montagem em trilho DIN e conectores extraíveis.

Alto desempenho para aplicações distribuídas

Os controladores programáveis Nexto Xpress também possuem saídas digitais com suporte de 1,5 A por canal (12 A no máximo) e entradas analógicas com resolução 12 bits e atualização de todos os canais em menos de um milissegundo, um dos mais rápidos do mercado.

Esta profusão de recursos dá à plataforma a capacidade de controlar sistemas um pouco maiores, como os encontrados em projetos dos segmentos de automação predial e saneamento.

A precisão no controle de variáveis possibilita que o CLP atue tanto em estações de bombeamento e controle de nível de reservatório, quanto no controle dos processos inerentes ao Sistema de Automação Predial (BMS), como iluminação e climatização, supervisão de processos, transportadoras e controle de acesso.

Com design compacto, montagem em trilho DIN e baixo consumo de energia, os módulos de expansão de entradas e saídas Nexto Xpress possibilitam aumentar a quantidade de sensores e atuadores utilizados nas mais diferentes aplicações.

Os equipamentos da linha são indicados para soluções de automação predial, água e esgoto, indústrias de alimentos, têxteis, máquinas e diferentes soluções OEM. Além disso, os produtos podem ser aplicados em conjunto com todas as CPUs do portfólio da Série Nexto, atuando como um complemento para arquiteturas de pequenas a grandes aplicações, uma importante característica para integradores de sistemas com projetos de diferentes escalas.

Recursos valiosos de engenharia e programação

Assim como os demais produtos da família Nexto, a solução também utiliza o software de programação MasterTool, o que dá um amplo leque de recursos para os CLPs.

Os controladores Xpress contam com depuração e monitoramento online, o que permite ao desenvolvedor do projeto ou operador do sistema realizar alterações online e, também, simulações off-line.

Com configuração de hardware extremamente fácil, os CLPs possibilitam a utilização de variáveis simbólicas e ainda disponibilizam uma ampla variedade de diagnósticos de sistema e listas de variáveis globais.

Em termos de programação, a solução Xpress aceita linguagens do tipo Sequential Function Chart (SFC), Function Block Diagram (FBD), Ladder Diagram (LD), Instruction List (IL) e

Continuous Function Chart (CFC), além de possibilitar o uso de linguagens diferentes no mesmo projeto. Os produtos ainda possuem uma área de memória específica para armazenamento do código fonte, o que dá mais segurança e disponibilidade para sua aplicação.

Quais são as principais características do produto?

O painel frontal contém a identificação das interfaces de comunicação e E/S disponíveis nos controladores Nexto Xpress.

As interfaces de E/S digitais têm um LED para cada ponto para indicar o estado lógico, enquanto as interfaces de comunicação têm um LED cada para indicar atividade.

A disponibilidade dessas interfaces em cada modelo é descrita na próxima seção. Além disso, no lado direito do painel frontal, há 2 LEDs usados para indicar alimentação e diagnósticos.

Para desenvolver uma aplicação para controladores Nexto Xpress, é necessário verificar a versão do MasterTool IEC XE. Além disso, ao longo do roteiro de desenvolvimento do MasterTool IEC XE, alguns recursos podem ser incluídos (como Blocos Funcionais especiais etc.), que podem introduzir um requisito da versão mínima do firmware.

Durante o download da aplicação, o MasterTool IEC XE verifica a versão do firmware instalada no controlador e, se não atender ao requisito mínimo, exibirá uma mensagem solicitando atualização.

A versão mais recente do firmware pode ser baixada no site da Altus e é totalmente compatível com aplicações anteriores.

O desempenho do controlador Nexto Xpress depende dos seguintes fatores:

- **Tempo de Intervalo da Aplicação** – A aplicação e a atualização de E/S são executadas em uma tarefa cíclica (periódica) chamada MainTask. O tempo de intervalo desta tarefa pode ser configurado de 5 a 100 ms. O tempo gasto para essas operações é chamado de Tempo de Ciclo e sempre deve ser menor que o intervalo, porque o tempo livre é usado para comunicação e outras tarefas de baixa prioridade do controlador. Além disso, as E/S integradas podem ser atualizadas de forma assíncrona em qualquer ponto do código de aplicação do usuário, usando as funções de atualização;
- **Tempo da Aplicação do Usuário** – O tempo de execução da aplicação (tempo de ciclo) depende das seguintes variáveis: tempo de leitura de entradas integradas, tempo de execução de tarefa e tempo de escrita de saídas integradas. O tempo necessário para ler e escrever na E/S integrada depende do número e do tipo de canais de E/S habilitados. Para E/S digitais, todos os canais estão sempre ativados e o tempo adicionado à MainTask não é relevante. Para E/S analógicas, o tempo adicionado à MainTask é determinado pelo tempo de conversão (para entradas analógicas) e pelo tempo de atualização (para saídas analógicas);
- **Tempo para a Execução de Instruções** – Esse tempo varia de acordo com o tipo de instrução, a linguagem de programação utilizada e o tipo de variável, conforme valores indicados na documentação do produto;
- **Tempo de Inicialização** – O tempo de inicialização do controlador Nexto Xpress é de aproximadamente 40 s.

A figura a seguir ilustra o ciclo típico de um controlador programável.

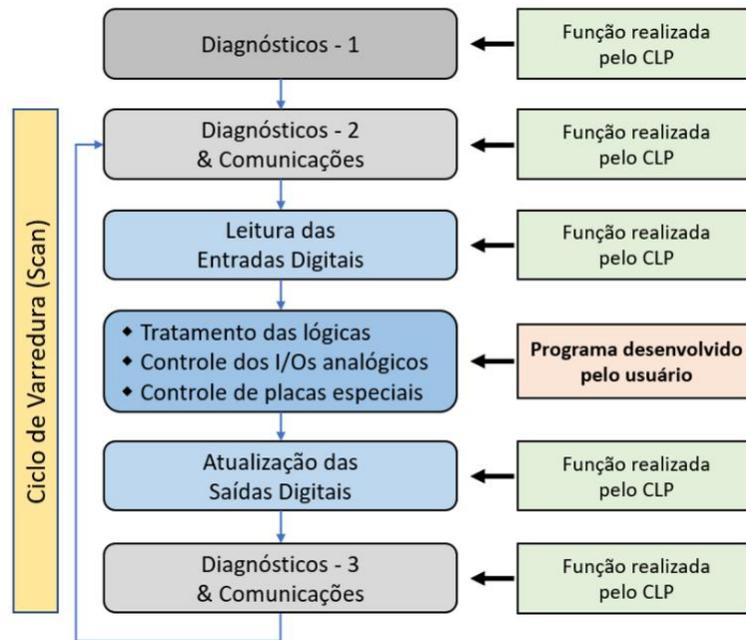


Figura 1-2. Ciclo típico de um controlador programável



Pesquisa na Documentação: características do produto

Consulte as tabelas de descrição técnica do produto: painéis e conexões, características técnicas, normas e certificações, compatibilidade com outros produtos e desempenho.

Fonte: Manual de Utilização Nexto Xpress – MU216000

Instalação do Nexto Xpress



Instalação Mecânica

Os controladores Nexto Xpress foram projetados para serem instalados em um trilho DIN padrão. Além disso, o usuário deve fornecer um painel adequado que atenda aos requisitos de proteção e segurança do sistema.

Para obter a especificação de temperatura do controlador, a instalação deve fornecer um espaço livre ao redor do dispositivo, conforme descrito na seção “Projeto do Painel” do Manual de Utilização da Série Nexto - MU214000. As figuras que seguem ilustram a sequência de instalação do controlador.

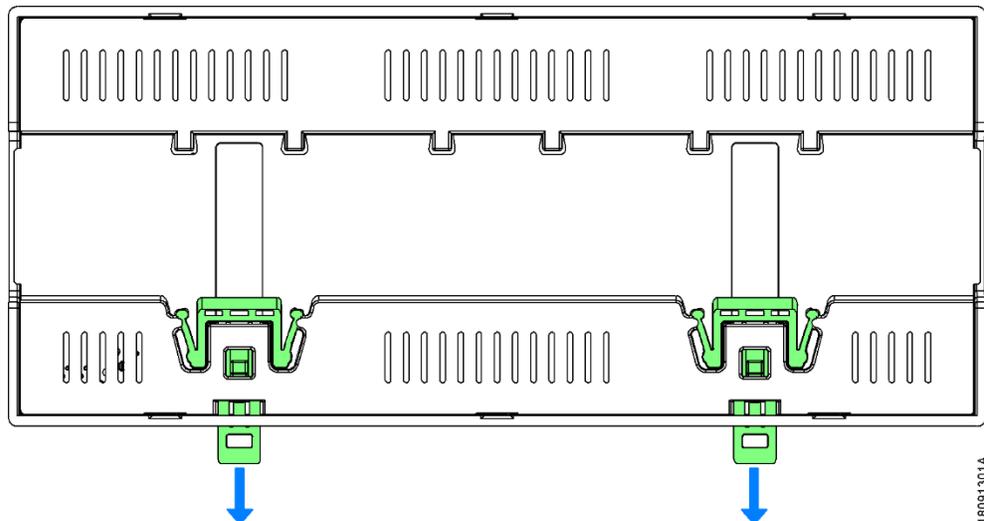


Figura 1-3. Movendo as travas para a posição aberta (1)

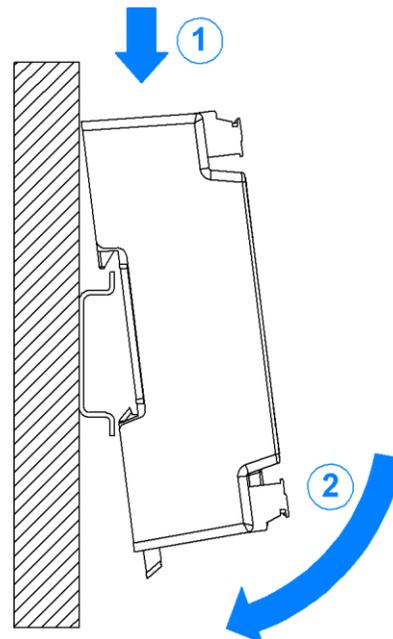


Figura 1-4. Posicionando o Controlador no Trilho DIN

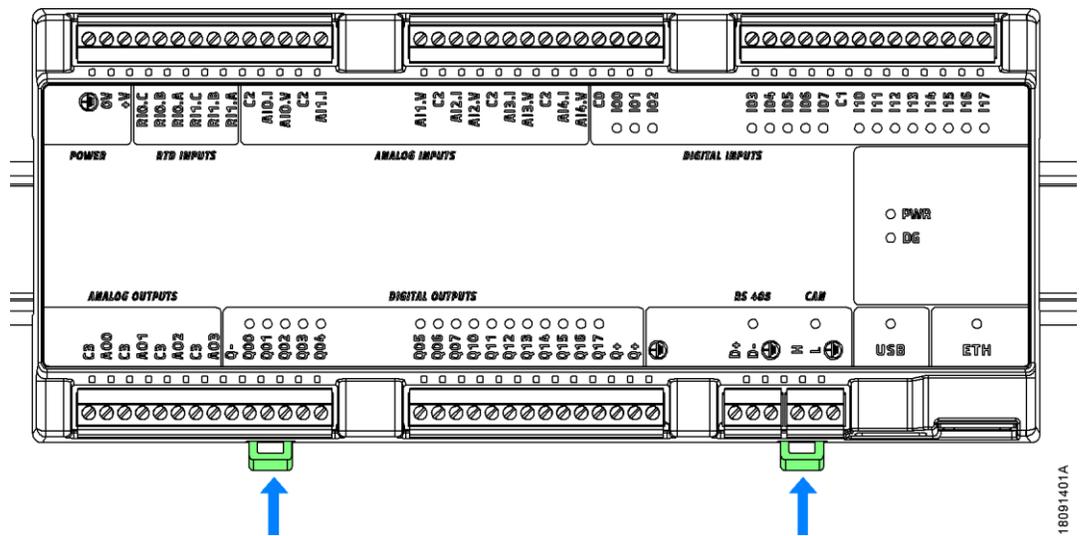


Figura 1-5. Travando o Controlador no Trilho DIN

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

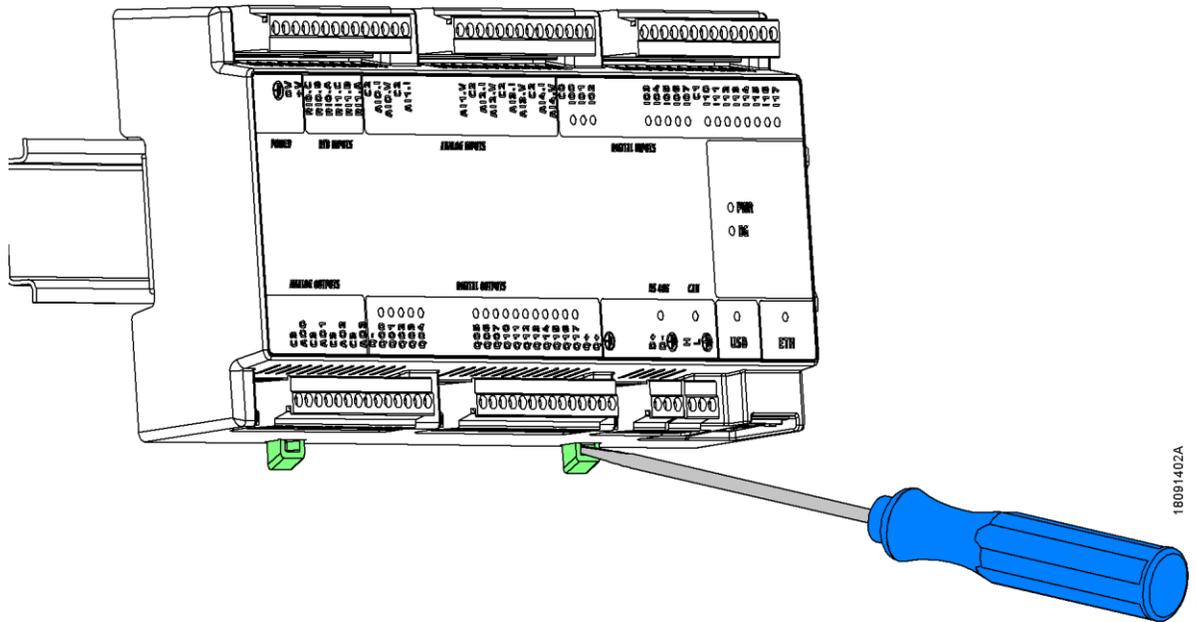


Figura 1-6. Destravando o Controlador no Trilho DIN

Instalação Elétrica

A figura a seguir mostra o diagrama de instalação elétrica do Nexto Xpress.

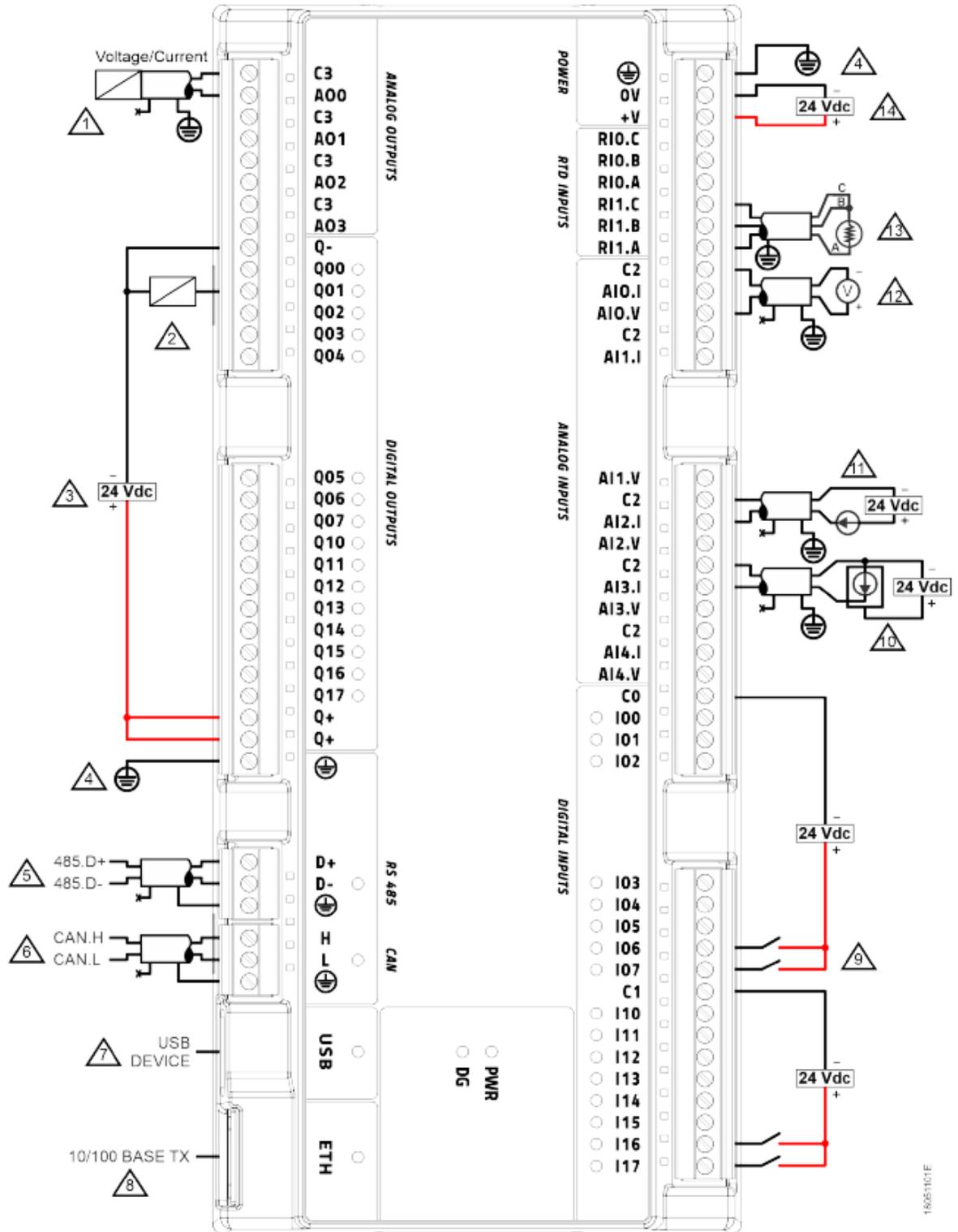


Figura 1-7. Diagrama de Instalação Elétrica do XP3xx

**NOTAS:**

1. Conexão típica da saída analógica no modo de tensão / corrente.
2. Conexão típica de saída digital (tipo fonte).
3. Fonte de alimentação externa para alimentação das saídas Q00 à Q17, os bornes Q+ devem ser conectados ao +24 Vdc, e borne Q- deve ser conectado ao 0 Vdc.
4. Terminais de aterramento de proteção para a fonte de alimentação e portas de comunicação. Ambos devem estar conectados externamente à terra.
5. Conexão típica da interface serial RS-485.
6. Conexão típica da interface CAN.
7. Verifique a tabela de características técnicas da porta USB para obter a lista de dispositivos suportados.
8. Use cabos Ethernet informados na documentação do produto.
9. Conexão típica de entrada digital (tipo *sink*). C0 e C1 são os pontos comuns para os grupos isolados I0x e I1x respectivamente.
10. Conexão típica da entrada analógica de corrente (dispositivo de campo com alimentação fornecida separadamente do sinal analógico).
11. Conexão típica da entrada analógica de corrente (dispositivo de campo com alimentação fornecida com o sinal analógico, 2 fios).
12. Conexão típica das entradas analógicas de tensão.
13. Conexão típica de entrada analógica RTD (3 fios).
14. Conexão da fonte de alimentação externa.

PERIGO

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

Instalação de Rede

Conexão de Rede Ethernet

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

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

A interface Ethernet (NET 1) vem com a seguinte configuração de parâmetros padrão:

- Endereço IP: 192.168.15.1;
- Máscara de sub-rede: 255.255.255.0;
- Endereço do Gateway: 192.168.15.253.

Primeiro, a interface NET 1 deve estar conectada a uma rede de PC com a mesma máscara de sub-rede para se comunicar com o MasterTool IEC XE, onde os parâmetros de rede podem ser modificados.

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

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

A porta Ethernet do Nexto Xpress possui pinagem padrão que é a mesma usada em PCs. O tipo de conector, o tipo de cabo, o nível físico, entre outros detalhes, é definido na tabela de Características Gerais da documentação do produto.

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

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

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

Conexão de Rede Serial RS-485 e CAN

As interfaces RS-485 e CAN usam dois sinais de comunicação e terra. O cabo recomendado é o AL-2306, usando um dos dois pares e a blindagem. Se o controlador for colocado em uma das extremidades da rede, a terminação interna deverá ser habilitada.

Instalação dos Módulos de Expansão

Os modelos XP101, XP106, XP201, XP600 e XP610 são módulos de E/S que oferecem soluções de expansão para a família do Controlador Programável Nexto Xpress.

Com um design compacto, eles se conectam com o Nexto Xpress através do protocolo CANopen de comunicação, assumindo o modo Escravo. Esses modelos são configurados por meio do software de programação MasterTool IEC XE.

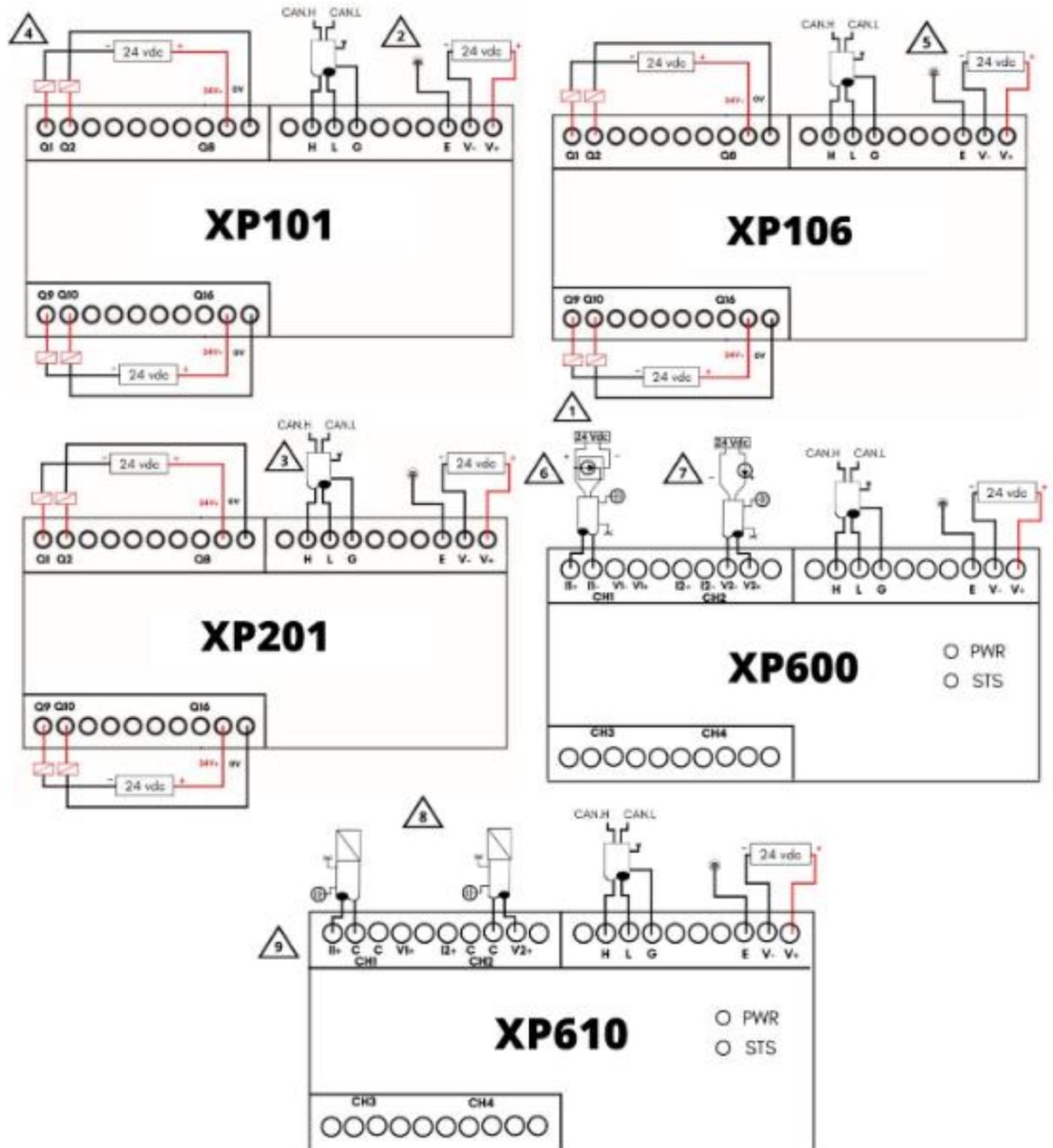
O XP101 apresenta 16 pontos de entrada digital, o XP106 oferece 8 pontos de entrada e 6 de saída digital a relé, o XP201 apresenta 16 de saída digital a transistor, o XP600 adiciona 4 Canais de Entrada Analógica tensão/corrente, enquanto o XP610 acrescenta 4 Canais de Saída Analógica tensão/corrente.

Estes modelos podem ser utilizados em segmentos como infraestrutura, automação predial, água, efluentes, alimentos, têxtil, automação de fábrica, máquinas e várias outras soluções OEM. Além disso, são soluções ideais para complementar grandes aplicações junto com o portfólio da Série Nexto, utilizando a mesma tecnologia e ambiente de engenharia. Esta é uma grande vantagem para OEMs e integradores de sistemas com necessidades de pequenas a grandes aplicações.

Para a configuração da interface CANopen devem ser utilizadas as chaves S1 até S7 para endereço e chaveamento e S8 até S10 para taxa de transmissão (Baud Rate).

As chaves do tipo DIP devem ser ajustadas de acordo com as orientações disponíveis na documentação de características técnicas desses módulos.

A figura a seguir mostra as instalações elétricas típicas desses módulos de expansão.



NOTAS:

1. Fonte de alimentação externa para alimentação das saídas Q00 à Q05 do XP106, o borne C deve ser conectado ao +24 Vdc.

2. Terminais de aterramento de proteção para a fonte de alimentação e portas de comunicação. Ambos devem estar conectados externamente ao conector terra.
3. Conexão típica da interface CAN.
4. Conexão típica de entrada digital (tipo sink). Os comuns (C) são pontos comuns para os grupos isolados I0x e I1x respectivamente.
5. Conexão da fonte de alimentação externa.
6. Dispositivo de campo com alimentação fornecida separadamente do sinal analógico.
7. Dispositivo de campo com alimentação fornecida com o sinal analógico, 2 fios.
8. Conexão típica de saída analógica.
9. Nas expansões analógicas os canais "CHx" possuem dois modos de sinal analógico, de tensão "Vx" e de corrente "Ix", porém apenas um dos tipos de sinal pode ser utilizado por canal, o de tensão OU o de corrente.

Manutenção do Nexto Xpress



Diagnósticos

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

- Diagnósticos via LED: é puramente visual, gerado através de dois LEDs colocados no painel frontal (PWR e DG);
- Diagnósticos via web: é a visualização gráfica em uma página da web;
- Diagnósticos via Variáveis: são apresentados como variáveis simbólicas globais a serem utilizadas na aplicação do usuário, por exemplo, sendo apresentadas em um sistema de supervisão;
- Diagnósticos via Blocos Funcionais: apresentam condições específicas do funcionamento do sistema.

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

Diagnósticos via LED

Os controladores Nexto Xpress possuem um LED de energia (PWR) e um LED de indicação de diagnóstico (DG).



Pesquisa na Documentação: LEDs de diagnóstico

Consulte, na documentação do produto, a descrição dos Estados do LED de Diagnóstico.

Fonte: Manual de Utilização Nexto Xpress – MU216000

Diagnósticos via Web

O Nexto Xpress fornece acesso aos estados de operação e diagnósticos do sistema por meio de uma página web. A utilização, além de dinâmica, é muito intuitiva e facilita as operações do usuário. O uso de um sistema de supervisão pode ser substituído quando estiver restrito à verificação do estado do sistema.

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

Modelo	XP340
Descrição	Nexto PLC
Versão de Firmware	1.9.7.0
Versão do Bootloader	1.0.0.15
Modo de Operação	Controlador Programável
Estado da Aplicação	Stop (Parada)
Diagnósticos Ativos	0
Valores Forçados	Não
Dispositivo USB	Não encontrado

Figura 1-8. Exemplo de informações do controlador disponíveis na tela inicial

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

Diagnósticos via Variáveis

Os controladores Nexto Xpress oferecem um conjunto de variáveis simbólicas globais, que fornecem diversas informações de diagnóstico relacionadas ao hardware e ao software. Essas estruturas de dados simbólicas são criadas automaticamente pelo ambiente de programação MasterTool IEC XE. A funcionalidade de diagnóstico pode ser subdividida em dois tipos: diagnósticos resumidos e detalhados.



Pesquisa na Documentação: Diagnósticos Resumidos

Consulte, na documentação do produto, a descrição dos diagnósticos resumidos.

Fonte: Manual de Utilização Nexto Xpress – MU216000

Algumas observações importantes devem ser feitas em relação à manutenção (diagnósticos resumidos) conforme descrito a seguir.

- **Falha de Hardware:** caso o diagnóstico de falha de hardware seja verdadeiro, o controlador deve ser enviado para a Assistência Técnica da Altus, pois há problemas no RTC ou em outros recursos de hardware.
- **Exceção de Software:** caso o diagnóstico de exceção de software seja verdadeiro, o usuário deve verificar sua aplicação para garantir que não está acessando a memória incorretamente. Se o problema persistir, o setor de Suporte Técnico da Altus deve ser consultado. Os códigos de exceção do software são descritos a seguir na tabela de diagnósticos detalhados do controlador.
- **Erro de Retentividade:** O CP grava dados na memória retentiva a cada 5 segundos em tempo de execução. Quando esse bit for TRUE, a causa raiz mais provável é um erro de hardware na memória retentiva. Nesse caso, a UCP deve ser enviada à Assistência Técnica da Altus. Os comandos Reset a Frio e Reset Origem acionados pelo programador MasterTool não causam a indicação deste diagnóstico.



Pesquisa na Documentação: Diagnósticos Detalhados

Consulte, na documentação do produto, a descrição dos diagnósticos detalhados, os códigos de exceção gerados pelo RTS (Runtime System) e entradas RTD.

Fonte: Manual de Utilização Nexto Xpress – MU216000

Em relação aos diagnósticos detalhados, os aspectos a seguir devem ser observados.

- **Visualização das Estruturas de Diagnóstico:** as Estruturas de Diagnósticos adicionadas ao Projeto podem ser vistas no item Gerenciador de bibliotecas da visualização em árvore do MasterTool IEC XE. Lá, é possível ver todos os tipos de dados definidos na estrutura
- **Contadores:** todos os contadores de diagnósticos do controlador retornam a zero quando seu valor limite é excedido.
- **Reset por Brownout:** o diagnóstico de reset de queda de energia (brownout) só é verdadeiro quando a fonte de alimentação excede o limite mínimo requerido em suas características técnicas, permanecendo em baixa tensão, sem sofrer qualquer interrupção. O controlador identifica a quebra de tensão e indica o diagnóstico de falha

da fonte de alimentação. Quando a tensão é restabelecida, o controlador é reiniciado automaticamente e indica o diagnóstico de reinicialização.

- **Contador de Erro de Paridade:** quando a serial COM 1 é configurada sem paridade, esse contador de erros não será incrementado quando receber uma mensagem com uma paridade diferente. Neste caso, um erro de frame será indicado.
- **Partição do Usuário:** a partição do usuário é uma área de memória reservada para o armazenamento de dados na UCP. Por exemplo: arquivos com extensão PDF, arquivos com extensão DOC e outros dados.

Diagnósticos via Função

Algumas funções localizadas na biblioteca NextoStandard permitem a visualização de parâmetros que não podem ser acessados de outra maneira, tais como a GetNoOfApplication que retorna o número total de aplicações válidas que estão na UCP Nexto Xpress, a GetAppInfo: que informações sobre a aplicação desejada, e a GetTaskInfo que retorna informações sobre uma tarefa de uma determinada aplicação.

Pesquisa na Documentação: Diagnósticos via Função

Consulte, na documentação do produto, as funções que proporcionam a visualização de parâmetros específicos com ênfase na GetTaskInfo conforme apresentado a seguir.



GetTaskInfo: essa função retorna informações sobre uma tarefa de uma determinada aplicação.

```

GetTaskInfo
- psAppName  POINTER TO STRING      ERRORCODE GetTaskInfo
- psTaskName  POINTER TO STRING
- pstTaskInfo POINTER TO stTaskInfo
    
```

Parâmetros de entrada	Descreva os parâmetros que devem ser repassados...
psAppName	
psTaskName	
pstTaskInfo	

Dados de retorno	Descreva os dados que a função retorna, via ponteiro informado nos parâmetros de entrada...
dwCurScanTime	
dwMinScanTime	
dwMaxScanTime	
dwAvgScanTime	
dwLimitMaxScan	
dwIECCycleCount	

Códigos de erro (ERRORCODE)	Descreva os códigos de erro possíveis...
NoError	
TaskNotPresent	

O trecho de código, na linguagem ST (Texto Estruturado), mostrado a seguir exemplifica o uso da função GetTaskInfo.

```
PROGRAM UserPrg
VAR
sAppName : STRING;
psAppName : POINTER TO STRING;
sTaskName : STRING;
psTaskName : POINTER TO STRING;
pstTaskInfo : POINTER TO stTaskInfo;
TaskInfo : stTaskInfo;
Info : 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.
```

Fonte: Manual de Utilização Nexto Xpress – MU216000

Manutenção Preventiva

Os diagnósticos abordados anteriormente estão relacionados principalmente ao monitoramento geral do funcionamento do controlador e fornecem um suporte consistente à manutenção corretiva do sistema. No tocante à manutenção preventiva, os seguintes aspectos devem ser observados:

- Deve ser verificado, a cada ano, se os cabos de interconexão estão conectados firmemente, sem acumulação de poeira, principalmente os dispositivos de proteção;
- Em ambientes sujeitos a contaminação excessiva, o equipamento deve ser periodicamente limpo de poeira, detritos etc.;
- Os diodos TVS usados para proteção contra transientes causados por descargas atmosféricas devem ser inspecionados periodicamente, pois podem ser danificados ou destruídos caso a energia absorvida esteja acima do limite;
- O aperto e limpeza do bloco conector deve ser executado a cada seis meses.

2. Software do Nexto Xpress



Características do MasterTool IEC XE



Assim como para os demais dispositivos da Série Nexto, a programação dos controladores Nexto Xpress é feita através da interface de desenvolvimento do MasterTool IEC XE (IDE).

O MT8500 – MasterTool IEC XE é o software para programação, configuração, diagnóstico e comissionamento da Série Nexto. Ele é um software programador de CP independente de dispositivo.

Este produto oferece características para todas as etapas de desenvolvimento de um sistema de automação, começando por uma análise gráfica da topologia da arquitetura, passando por um ambiente de programação com suporte às linguagens da norma IEC 61131-3, com uma ferramenta de simulação realística, onde o usuário pode verificar o comportamento da aplicação antes de executá-la em um sistema real, e, finalmente, provê uma completa interface para visualização de diagnósticos e status. A figura a seguir ilustra o ambiente de programação MasterTool IEC XE.

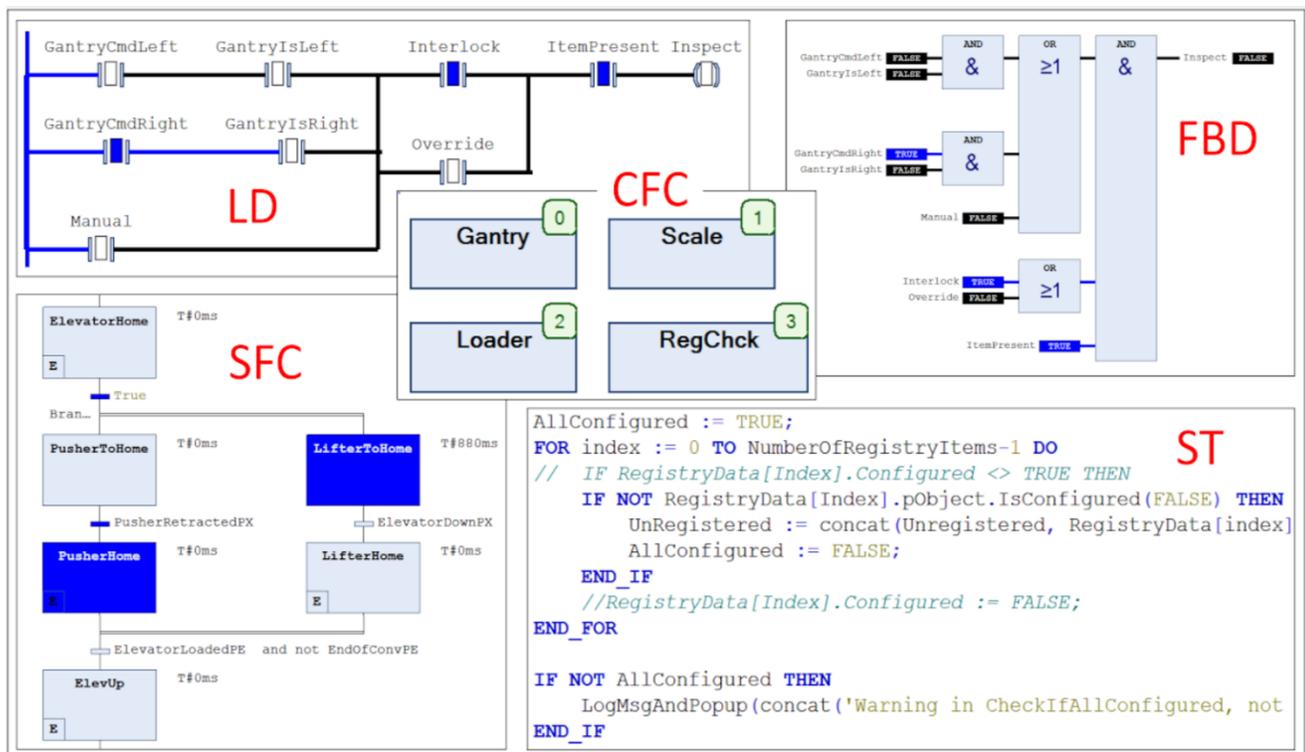


Figura 2-1. Ambiente de programação MasterTool IEC XE

As suas principais características estão sumarizadas na tabela a seguir.

Tabela 2-1. Características do MasterTool IEC XE

Linguagens de Programação IEC 61131-3	O MasterTool IEC XE oferece todos os editores definidos na norma IEC para desenvolvimento da aplicação: Texto Estruturado (ST), Sequenciamento Gráfico de Funções (SFC), Diagrama de Blocos Funcionais (FBD), Diagrama Ladder (LD) e Gráfico Funcional Contínuo (CFC). Todos os editores foram especialmente desenvolvidos para garantir um ótimo gerenciamento. Ideias e sugestões de usuários experientes foram incorporadas ao processo de desenvolvimento.
Editores para Configuração de Projeto e de Hardware	Com a ajuda de editores especiais, um projeto pode ser facilmente configurado no MasterTool IEC XE. A ferramenta gráfica permite uma rápida e amigável maneira de configurar o sistema. Adicionalmente, o usuário tem a visualização completa da arquitetura da aplicação com a posição física e as informações de cada módulo. O MasterTool IEC XE torna o uso de interfaces para redes de campo uma prática tão simples como nunca visto anteriormente. O usuário não precisa um software especial para configurar a rede de campo porque o aplicativo atende este requisito por meio de uma única ferramenta reduzindo tempo de desenvolvimento e simplificando a aplicação.
Programação Orientada ao Objeto	O aplicativo oferece uma programação orientada a objeto com as vantagens conhecidas das modernas linguagens de alto nível, como JAVA ou C++: classes, interfaces, métodos, herança, polimorfismo etc. Os blocos funcionais IEC são perfeitamente estendidos e as extensões estão disponíveis a todos os aspectos da engenharia. A programação orientada a objeto oferece grandes vantagens ao usuário, como por exemplo, quando se deseja reutilizar partes existentes de uma aplicação ou quando deseja trabalhar em uma aplicação com vários desenvolvedores.
Funcionalidades: Online, Depuração e Comissionamento	O código gerado a partir da aplicação é enviado para o dispositivo com um simples clique de mouse. A partir do momento em que o MasterTool IEC XE está online, são disponibilizadas várias funções importantes para realizar uma depuração rápida e eficiente, assim como para teste e comissionamento.
Simulação	Uma característica que permite ao usuário avaliar e testar diversas lógicas e algoritmos é a ferramenta de simulação. Este recurso permite que as aplicações do usuário sejam projetadas e testadas sem a necessidade de um controlador conectado. Isto é também interessante para treinamento, documentação e avaliação de casos de teste.
Desenvolvimento de páginas Web (acessíveis através de protocolo HTTP)	Além de permitir que o usuário crie a lógica responsável pelo controle de processos segundo a norma IEC61131-3, o MasterTool IEC XE também oferece um editor para o desenvolvimento de telas que permitem supervisão e operação do processo automatizado. Estas telas são desenvolvidas de maneira simples através de um editor gráfico que possui diversos objetos predefinidos e que são armazenados de forma vetorial juntamente com o projeto da lógica. Entre estes objetos se encontram formas geométricas, medidores de ponteiro e barras, controles para edição e visualização de variáveis, objetos que permitem carregar imagens entre outras funcionalidades. O editor das telas está integrado ao MasterTool IEC XE de forma que a interação destas com a lógica do CP é feita de forma transparente e intuitiva reduzindo significativamente o tempo de desenvolvimento e integração destas. Uma vez criadas as telas, elas são carregadas no CP em conjunto com o restante do programa e são acessadas utilizando um navegador web através do protocolo HTTP. Apesar das telas serem armazenadas e exibidas no formato web, não é necessário que os usuários possuam qualquer conhecimento sobre linguagens de programação, sendo toda a configuração feita de forma gráfica e intuitiva.
Documentação de Usuário & Arquivos de Ajuda	Considerando que a programação do CP de acordo com as linguagens da norma IEC 61131-3 é uma tarefa complexa, o MasterTool IEC XE oferece um extenso arquivo de ajuda com várias dicas e descrições para orientar e servir como uma primeira base de dados de resolução de problemas para o usuário na criação de códigos de lógica ou no uso das características do software. Este arquivo de ajuda é disponibilizado em diferentes idiomas, de acordo com as opções de instalação. O MasterTool IEC XE também oferece suporte a múltiplos idiomas, permitindo ao usuário selecionar o seu idioma preferido a partir das opções disponíveis. A ideia é minimizar os possíveis problemas de entendimento decorrentes do uso de uma língua estrangeira. Como parte da documentação do usuário, o MT8500 pode imprimir documentos de aplicação tais como listas de materiais (BOM), POUs e parâmetros de configuração.
Diagnósticos Avançados	Uma das inovações chave da Série Nexto é o seu amplo suporte de diagnósticos. Esta ideia surgiu das demandas de extensas e complexas aplicações onde o correto uso de cada informação é fundamental para a manutenção, resolução de problemas e prevenção de potenciais problemas. Esta característica também está presente no MasterTool IEC XE onde o usuário, enquanto estiver conectado a uma UCP em execução, poderá acessar estruturas de diagnóstico completo através de janelas de monitoração e páginas da web.
Visualização que utiliza o conceito de abas (tecnologia Docking View)	A tecnologia <i>Docking View</i> permite ao usuário customizar o ambiente MasterTool IEC XE de acordo com suas necessidades pessoais. Adicionalmente, o usuário pode editar a estrutura do menu, as atribuições das teclas e a barra de ferramentas conforme desejado. Esta característica fornece uma interface de usuário amigável para maximizar a experiência com a ferramenta do software.

Mecanismos de proteção da aplicação e características de segurança	O MasterTool IEC XE também oferece dois diferentes mecanismos de proteção da aplicação e características de segurança: Proteção da Propriedade Intelectual e Login Seguro no CP. A Proteção da Propriedade Intelectual tem por objetivo proteger a propriedade intelectual do usuário, permitindo a ele proteger todo o projeto ou arquivos específicos dentro do projeto através da definição uma senha de acesso. Isso significa que estes arquivos estarão disponíveis (para operação de leitura e escrita) apenas depois de desbloqueados com a senha correta. Já o Login Seguro no CP provê uma maneira de proteger a aplicação do usuário de qualquer acesso não autorizado. Habilitando esta característica, a UCP da Série Nexto irá solicitar uma senha de usuário antes de executar quaisquer comandos entre MasterTool IEC XE e a UCP, como parar e programar a aplicação ou forçar pontos de saída em um módulo.
Funcionalidades adicionais para aumento da produtividade	Outras características importantes também estão disponíveis para aumentar a produtividade do usuário como: <ul style="list-style-type: none"> • Módulo de impressão que consiste em um relatório com os parâmetros específicos dos módulos e as configurações da aplicação; • Impressão de lógicas que consiste em um relatório com todo o código da aplicação; Verificação de Projeto que auxilia o usuário a verificar diferentes condições durante a programação como: sintaxe do programa, consumo de corrente da fonte de alimentação, regras de posicionamento dos módulos da Série Nexto, parametrização e configuração de módulos; • Depuração em tempo real que provê uma maneira de verificar a funcionalidade da aplicação passo-a-passo, verificar o conteúdo de variáveis ou ainda adicionar e remover breakpoints durante a programação da UCP da Série Nexto.

Recursos básicos do MasterTool IEC XE

Os seguintes princípios caracterizam a programação via MasterTool IEC XE:

- **Orientação a objetos** - A orientação a objetos é contemplada pela disponibilidade das características e dos elementos de programação associados, pelo tratamento da estrutura e pela forma como o projeto é organizado;
- **Estrutura do programador baseada em componentes** - A funcionalidade disponível na interface do usuário (editores, menus) depende dos componentes utilizados. Existem componentes essenciais e opcionais;
- **Organização do projeto também determinada pela orientação ao objeto** - Um projeto do MasterTool IEC XE contém um programa de CP composto de vários objetos de programação e da definição dos "recursos" necessários para executar as instâncias do programa (aplicação) nos sistemas dos dispositivos definidos (dispositivos, CPs);
- **Geração de código** - A geração de código é feita através de compiladores integrados e otimização do código de máquina versus tempos de execução;
- **Transferência de dados entre o MasterTool IEC XE e o dispositivo (controlador)** - Esta operação é feita via gateway (componente) e um sistema de execução;
- **Interface padrão e profissional** - Configurações pré-definidas oferecem a possibilidade de escolher entre uma interface de usuário "padrão" (seleção reduzida de configurações com menor complexidade) ou um ambiente "profissional", o qual suporta todas as configurações. A escolha entre estas opções é feita quando o programador é inicializado após a primeira instalação no sistema, porém é possível alterar esta opção posteriormente, assim como se pode também adotar uma customização definida pelo usuário.

Recursos avançados do MasterTool IEC XE

A seguir, são apresentadas as funcionalidades avançadas disponíveis no MasterTool IEC XE:

- **Orientação a objetos na programação e na estrutura do projeto** – Esta funcionalidade inclui extensões para blocos funcionais (propriedades, métodos, herança, invocação de método) e aplicações vinculadas a dispositivos como instâncias de objetos de programação independentes;

- **Tipos de dados especiais** – Esses tipos incluem ANY_TYPE, UNION, LTIME, referências, enumerações (tipos de dados básicos podem ser especificados) e DI: DINT := DINT#16#FFFFFFFF;
- **Operadores e variáveis especiais** – Esses elementos de software contemplam operadores de escopo (NAMESPACES estendidos), ponteiros de função que substituem o operador INSTANCE_OF, método Init (substituindo o operador INI), método Exit, variáveis de saída em funções e chamadas de métodos, variáveis do tipo VAR_TEMP/VAR_STAT/VAR_RETAIN/ VAR_PERSISTENT, expressões arbitrárias para inicialização de variáveis, atribuição como expressão e acesso de índice com ponteiros e STRINGS;
- **Conceito de gerenciamento de usuário e direitos de acesso** - Contas de usuários, grupos de usuários, direitos específicos de grupos para acesso e ações em objetos específicos;
- **Características em editores** – Vários recursos de edição avançados nos editores das linguagens de programação;
- **Versões de bibliotecas** - Várias versões de bibliotecas podem ser usadas no mesmo projeto utilizando o recurso de contextos, bem como instalação em repositórios, atualização automática e depuração;
- **Funcionalidades adicionais** - Menus, barra de ferramentas e uso de teclado, possibilidade de incluir componentes específicos do usuário, configuração do CP e configuração de tarefas integradas na árvore de dispositivos, suporte a UNICODE, comentários de linha, cão-de-guarda, seleção múltipla na árvore de objetos do projeto, ajuda online integrada na interface do usuário, compilação condicional, breakpoints condicionais, depuração (passo para o cursor e retorno à chamada anterior), driver de barramento de campo em conformidade com a norma IEC 61131-3, configuração do CP e de símbolos disponíveis na aplicação, alocação livre de memória de código e dados, cada objeto pode ser especificado como “interno” ou “externo” (link posterior no sistema de execução) e notas de pré-compilação referentes a erros sintáticos.

Versões do MasterTool IEC XE

O software MasterTool IEC XE MT8500 possui quatro versões de distribuição, cada uma com um portfólio otimizado, de acordo com a necessidade do usuário. São elas:

1. **Lite:** software programador gratuito que permite a programação e carga de projetos de até 320 pontos de E/S no barramento Nexto. Será utilizado nesse treinamento;
2. **Basic:** software que permite a programação e carga de projetos de até 2048 pontos de E/S no barramento Nexto;
3. **Professional:** software programador para todas as UCPs da Série Nexto;
4. **Advanced:** software programador com ferramentas para aplicações de grande porte com redundância de *half-cluster*.

Informações Gerais sobre o Programador

Para informações detalhadas sobre o aplicativo incluindo compatibilidade com os demais produtos, requisitos mínimos e recomendáveis, idiomas suportados e dados para compra consulte o Manual de Utilização MasterTool IEC XE MT8500 - MU299048.

Conceitos Básicos Associados aos Ambientes de Programação IEC61131



Objetos de Dispositivos

São objetos de dispositivos, aplicações, configurações de tarefas e são gerenciados na árvore de dispositivos ou no editor gráfico, dependendo do tipo do dispositivo. Ao inserir objetos, o hardware a ser controlado deve ser mapeado de acordo com determinadas regras.

Objetos de Programação

Os objetos de programação (POUs) que podem ser instanciados no projeto (em todas as aplicações definidas no mesmo) devem ser gerenciados na janela das POUs. Alguns exemplos destes objetos de programação são: programas, funções, blocos funcionais, métodos, ações, definições de tipos de dados, entre outros. O instanciamento é feito ao chamar uma POU de programa através de uma tarefa atribuída à aplicação. Os objetos de programação gerenciados na janela de dispositivos (atribuídos diretamente a uma aplicação) não podem ser instanciados por outra aplicação inserida abaixo.

Janela das POUs

Na janela *POUs* podem ser acrescentadas POUs, arquivos externos, entre outros. Além de exibir objetos de configurações e informações de projeto conforme ilustrado na figura a seguir.

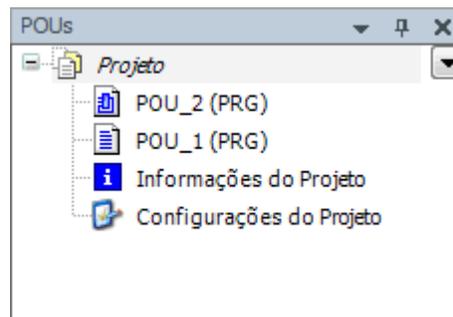


Figura 2-2. Janela dos POUs

Dispositivos, Árvore de Dispositivos

Na janela *Dispositivos* (“árvore de dispositivos”) define-se o hardware onde a aplicação será executada. Cada “dispositivo” representa um hardware específico (destino). Exemplos: controlador, módulos de E/S, monitor. Ele é definido por uma descrição e deve ser instalado no sistema local para que possa ser inserido na árvore de dispositivos (conforme mostrado na figura a seguir).

O arquivo de descrição define as propriedades referentes à configuração, programação e possíveis conexões com outros dispositivos.

Na árvore de dispositivos são gerenciados os objetos necessários para executar uma aplicação no dispositivo (controlador, CP), incluindo aplicação, configuração de tarefas e tarefas. Entretanto, objetos de programação específicos (POUs, listas de variáveis globais e gerenciador de bibliotecas) podem - em vez de serem gerenciados como unidades instanciáveis globais de projeto na janela das POUs - serem gerenciados SOMENTE na árvore de dispositivos e, neste caso, estarem disponíveis apenas na sua aplicação ou nas suas “aplicações secundárias”.

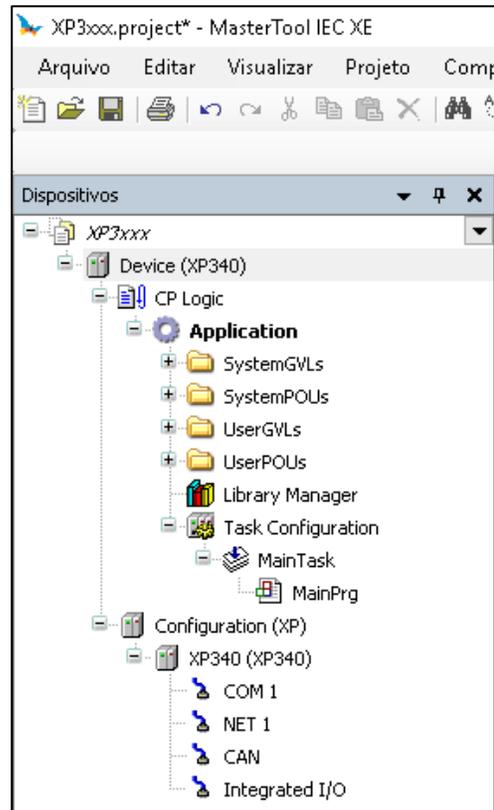


Figura 2-3. Árvore de Dispositivos

Aplicação

Uma “aplicação” é o conjunto dos objetos necessários para executar uma instância específica do programa do CP em um determinado dispositivo de hardware (CP, controlador). Para isto, objetos “independentes” gerenciados na visualização das POU’s são instanciados e atribuídos a um dispositivo na janela visualização dos *Dispositivos*. Isto está em conformidade com a programação orientada a objetos. Entretanto, POU’s específicas da aplicação também podem ser utilizadas.

Uma aplicação é representada por um objeto de aplicação (🌀) na árvore de dispositivos inserido abaixo de um nó do dispositivo programável (PLC Logic). Os objetos que definem o “conjunto de recursos” da aplicação podem ser inseridos abaixo de um item da aplicação.

A aplicação padrão, “Application”, é criada junto com novos projetos criados a partir do modelo *Projeto MasterTool Padrão* ela é acrescentada à árvore de dispositivos abaixo do item Device e PLC Logic.

Uma parte essencial da aplicação é a Task Configuration que controla a execução de um programa (instâncias de POU ou POU’s específicas da aplicação). Adicionalmente, podem estar atribuídos objetos de recursos, tais como listas de variáveis globais, bibliotecas, entre outros, os quais - ao contrário daqueles gerenciados na janela das POU’s - somente podem ser usados pela aplicação específica e seus subitens.

A compatibilidade dos parâmetros da aplicação no CP com os parâmetros da aplicação do projeto é verificada no momento do login em um dispositivo (CP ou dispositivo de simulação). Em caso de incompatibilidade, é exibida uma mensagem apropriada.

Task Configuration

A Task Configuration () define uma ou várias tarefas para controlar o processamento de um programa aplicativo.

Ela é um objeto de recurso essencial para uma aplicação e é inserido automaticamente ao criar um novo projeto a partir do modelo *Projeto MasterTool Padrão*. Uma tarefa pode chamar uma POU de programa específica da aplicação que esteja disponível na árvore de dispositivos, assim como o programa gerenciado na janela das POUs. Neste último caso, o programa do projeto global disponível será instanciado pela aplicação.

Uma task configuration pode ser editada no editor de tarefas, sendo as opções disponíveis específicas do dispositivo.

No modo online o editor de tarefas fornece uma visualização da monitoração e informações sobre os ciclos, tempos e status.

Geração de Código e Alteração Online

O código fonte não será gerado até que o projeto da aplicação seja enviado para o dispositivo (CP ou dispositivo de simulação). A cada envio, as informações de compilação contendo o código e a ID de referência da aplicação carregada serão armazenadas em um diretório do projeto em um arquivo "`<nome do projeto>.<nome do dispositivo>.<ID da aplicação>.compileinfo`". As informações de compilação serão apagadas quando os comandos *Limpar* e *Limpar Tudo* forem executados. No caso da opção de "Alteração Online", dependendo da modificação que for realizada no projeto em execução no controlador, apenas os objetos modificados serão carregados para o mesmo.

Monitoração

No modo online existem várias possibilidades de exibir os valores atuais das expressões de monitoração de um objeto no CP.

Depuração

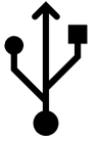
Para avaliar erros de programação use a funcionalidade de depuração do MasterTool IEC XE no modo online. Neste contexto, considere a possibilidade de verificar a aplicação no modo de simulação, isto é, sem a necessidade de conectar-se com um dispositivo de hardware real.

Breakpoints podem ser configurados em determinadas posições para forçar uma interrupção na execução. Algumas condições podem ser configuradas para cada breakpoint, como por exemplo, quais as tarefas associadas ou em quais ciclos o breakpoint deve atuar. Funções de passos estão disponíveis para que um programa seja executado em passos controlados. A cada interrupção, os valores atuais das variáveis podem ser examinados. Uma pilha de chamadas pode ser visualizada na posição do passo atual.

Segurança

Através dos diálogos Configurações do Projeto e Propriedades do Objeto, são configurados e gerenciados o controle de acesso aos projetos e objetos, assim como o direito de executar determinadas ações em um projeto. Um projeto pode ser protegido por uma senha codificada. Os Direitos de Acesso referentes a objetos são sempre atribuídos a grupos de usuários específicos, e não a usuários únicos. Para cada objeto, pode ser definida uma lista de ações permitidas para cada grupo. Cada usuário, no entanto, pode ter uma senha própria. Dependendo do dispositivo pode haver também um controle de acesso relacionado a objetos e arquivos no CP.

Funcionalidades do MasterTool IEC XE



A seguir, são apresentadas algumas funcionalidades disponíveis no MasterTool IEC XE.

Instalação

Para realizar a instalação do software de desenvolvimento MasterTool IEC XE é necessário efetuar o download do arquivo de instalação no site <http://www.altus.com.br>. Após, feche todos os programas que estejam em execução no seu computador e em seguida dê um duplo clique no arquivo de instalação e siga as orientações do aplicativo instalador.

Para desinstalar o programador e seus componentes ou para modificar a instalação atual, execute o arquivo de instalação.

Projeto

Um projeto contém os objetos das POUs que compõem um programa do CP, assim como as definições dos objetos de recursos necessários para executar uma ou mais instâncias do programa (aplicação) em determinados sistemas-destino (CPs, dispositivos). Objetos de POUs podem ser gerenciados na janela de visualização das POUs ou na janela de visualização dos dispositivos, POUs criadas a partir do wizard, aparecem na janela visualização dos dispositivos, e os objetos de recursos específicos do dispositivo são gerenciados na janela de visualização dos dispositivos. Um projeto é salvo no arquivo <nome do projeto>.project. A configuração básica de um novo projeto (estrutura de menu, objetos pré-definidos) é determinada pelo modelo do projeto usado. Este modelo é escolhido na criação de um novo arquivo de projeto com o comando *Novo Projeto*.

Perfis de Projeto

Um perfil de projeto no MasterTool IEC XE é um conjunto de regras, características comuns e padrões utilizados no desenvolvimento de uma solução de automação industrial, um perfil que influencia a forma de implementação da aplicação. Para os controladores Nexto Xpress, existe apenas um perfil de projeto disponível: Perfil de Máquina.

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

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 Program denominada MainPrg. Este programa pode chamar outras unidades de programação do tipo Program, Function ou Function Block, mas todo o código do usuário será executado exclusivamente pela tarefa MainTask.

Este perfil se caracteriza por permitir intervalos menores na tarefa MainTask, possibilitando a execução mais rápida do código do usuário. Este perfil pode adicionalmente incluir uma tarefa de interrupção de tempo, denominada TimeInterruptTask00, com maior prioridade que a tarefa MainTask e que conseqüentemente pode interromper a execução dela a qualquer momento.

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

Configurando a UCP

A configuração da UCP do controlador está localizada na árvore de dispositivos, conforme mostrado na figura a seguir, e pode ser acessada com um clique duplo no objeto correspondente. Nesta aba é possível configurar o comportamento do cão de guarda, sincronismo do clock, entre outros parâmetros.

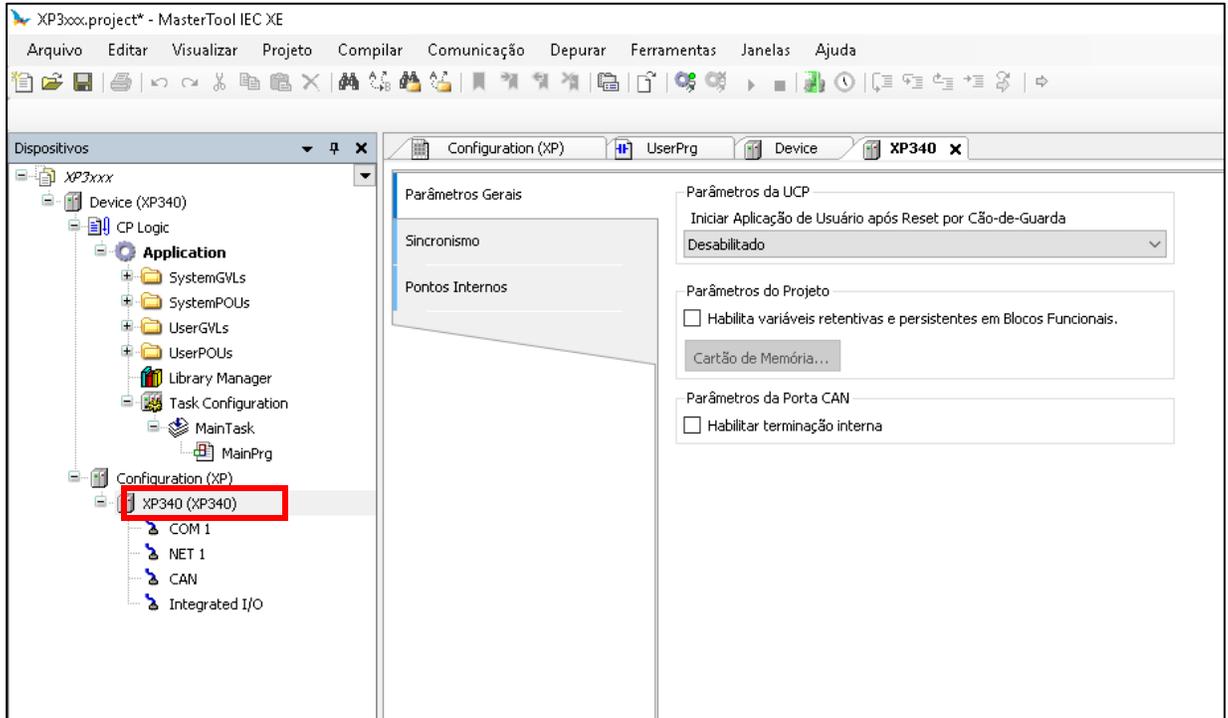


Figura 2-4. Configuração da UCP

Além disso, ao clicar duas vezes no ícone NET 1 do controlador, é possível configurar a interface Ethernet que será utilizada para comunicação entre o controlador e o software MasterTool IEC XE conforme mostrado na figura a seguir.

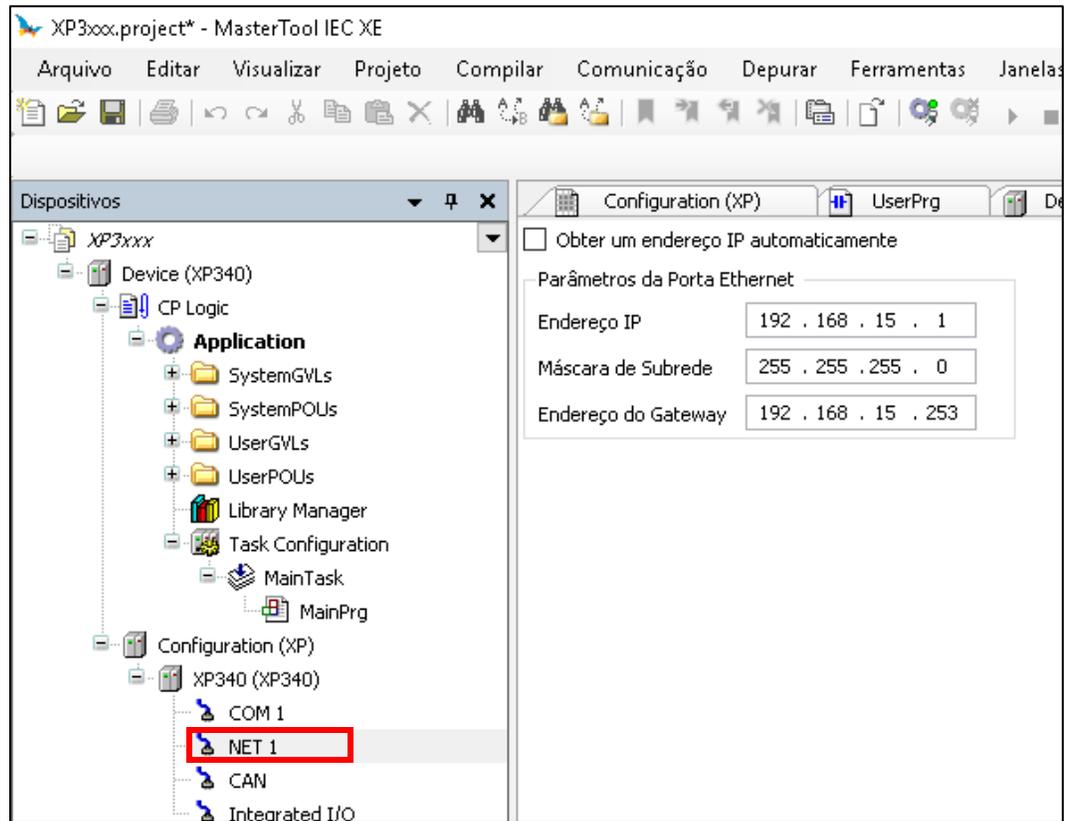


Figura 2-5. Configurando a Porta de Comunicação

A configuração definida nesta guia será aplicada ao dispositivo somente ao enviar a aplicação para o dispositivo (download).

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

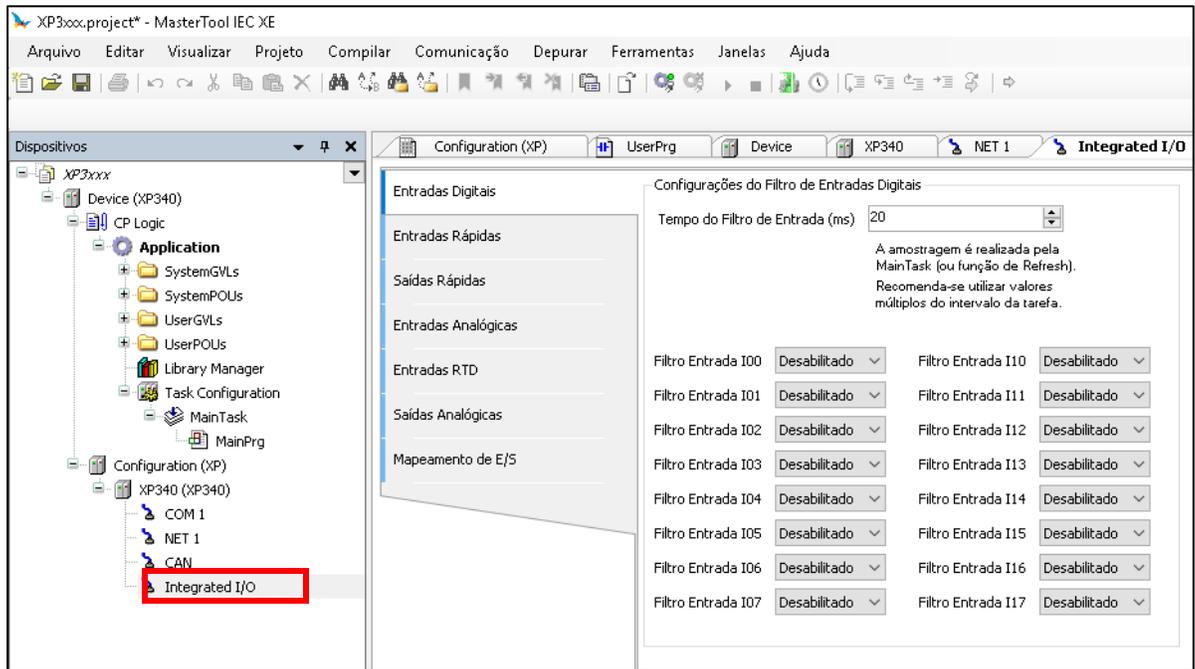


Figura 2-6. Configurando as E/S Integradas

Bibliotecas

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

Inserindo uma Instância de Protocolo

Os controladores Nexto Xpress oferecem diversos protocolos de comunicação.

Exceto as comunicações OPC, que possuem um procedimento de configuração diferente, para inserir um protocolo basta clicar com o botão direito sobre a interface de comunicação desejada, adicionar o dispositivo e finalmente configurar a instância do protocolo. O detalhamento da configuração das instâncias dos protocolos pode ser conferido mais adiante nesse tutorial. A seguir são apresentados alguns exemplos básicos.

MODBUS Ethernet

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

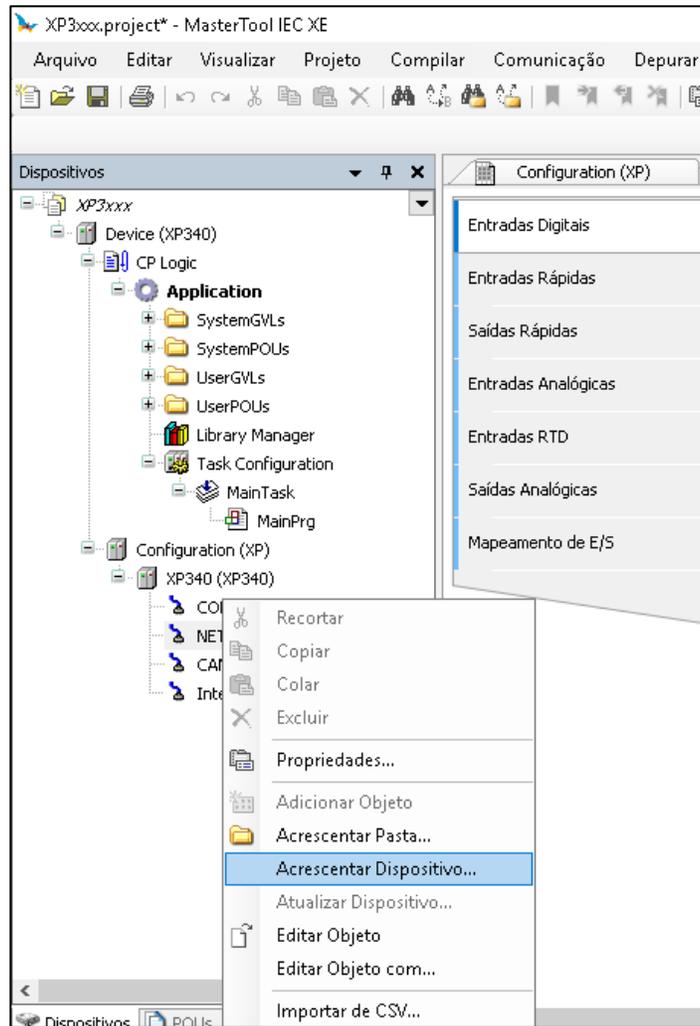


Figura 2-7. Adicionando a Instância

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

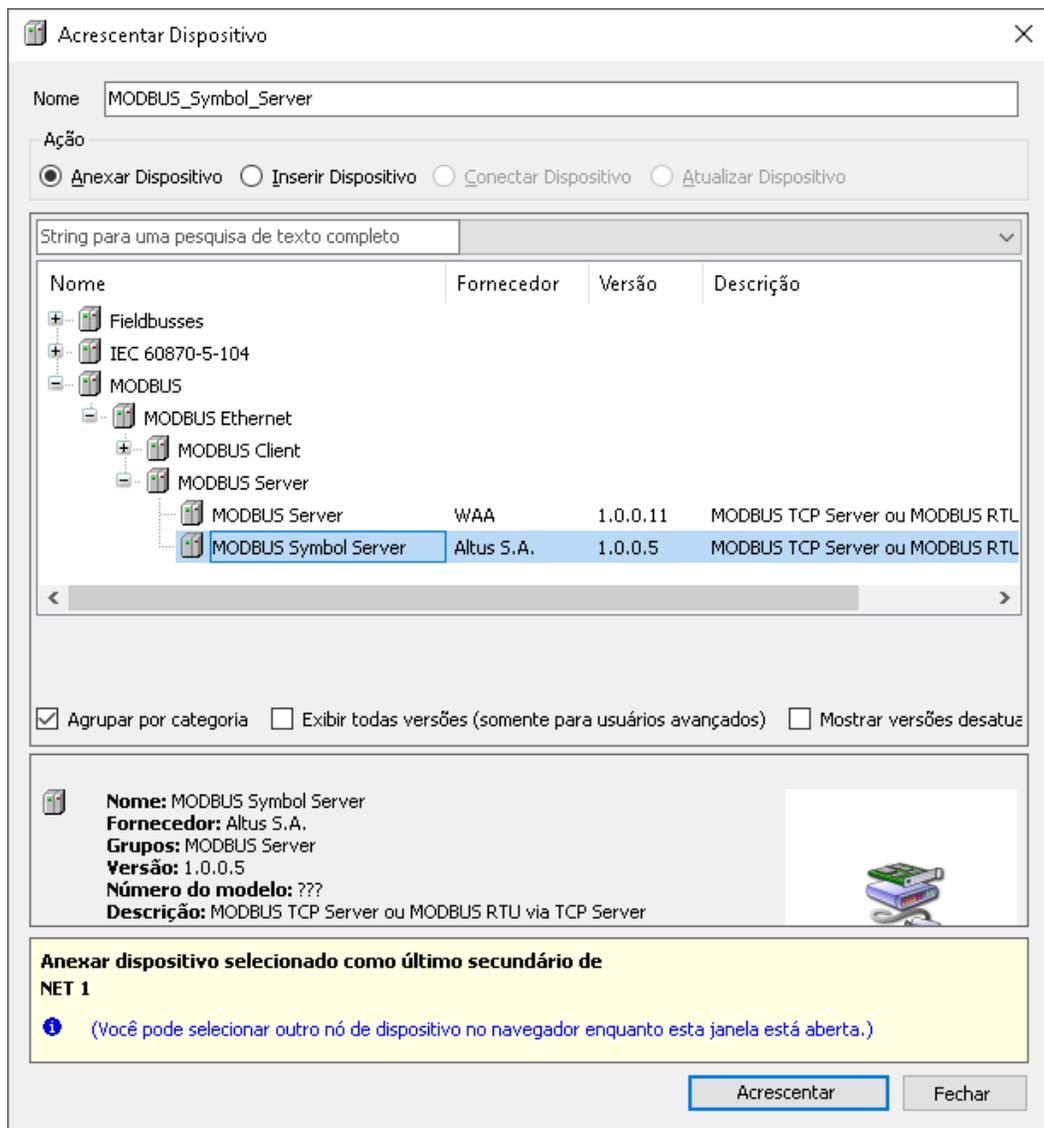


Figura 2-8. Selecionando o Protocolo

Localizando o Dispositivo

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

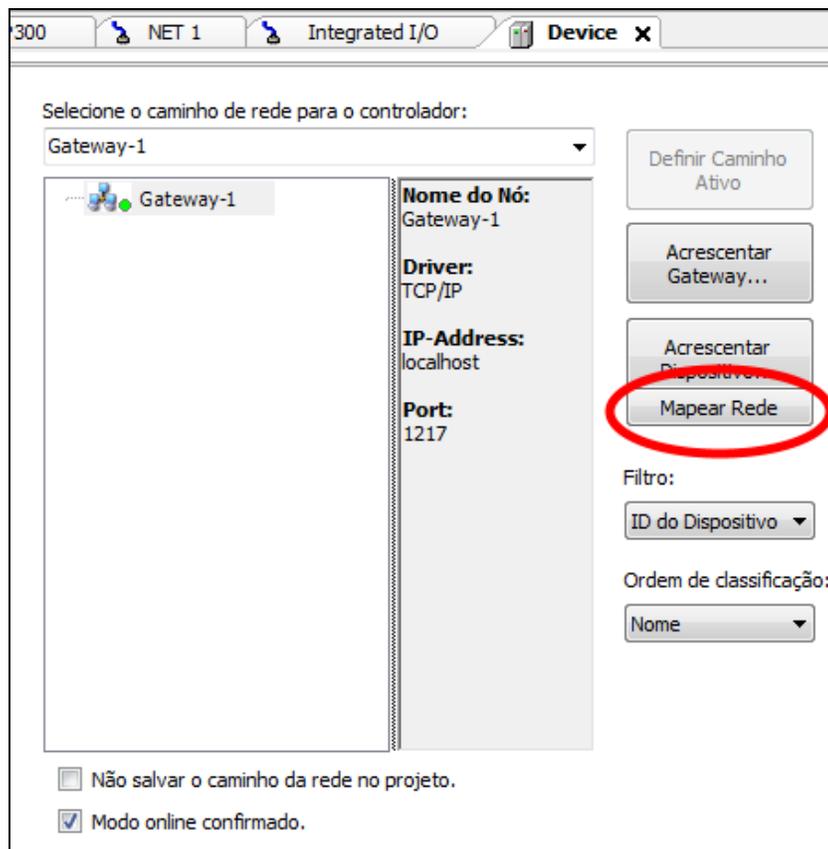


Figura 2-9. Localizando o Dispositivo

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

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

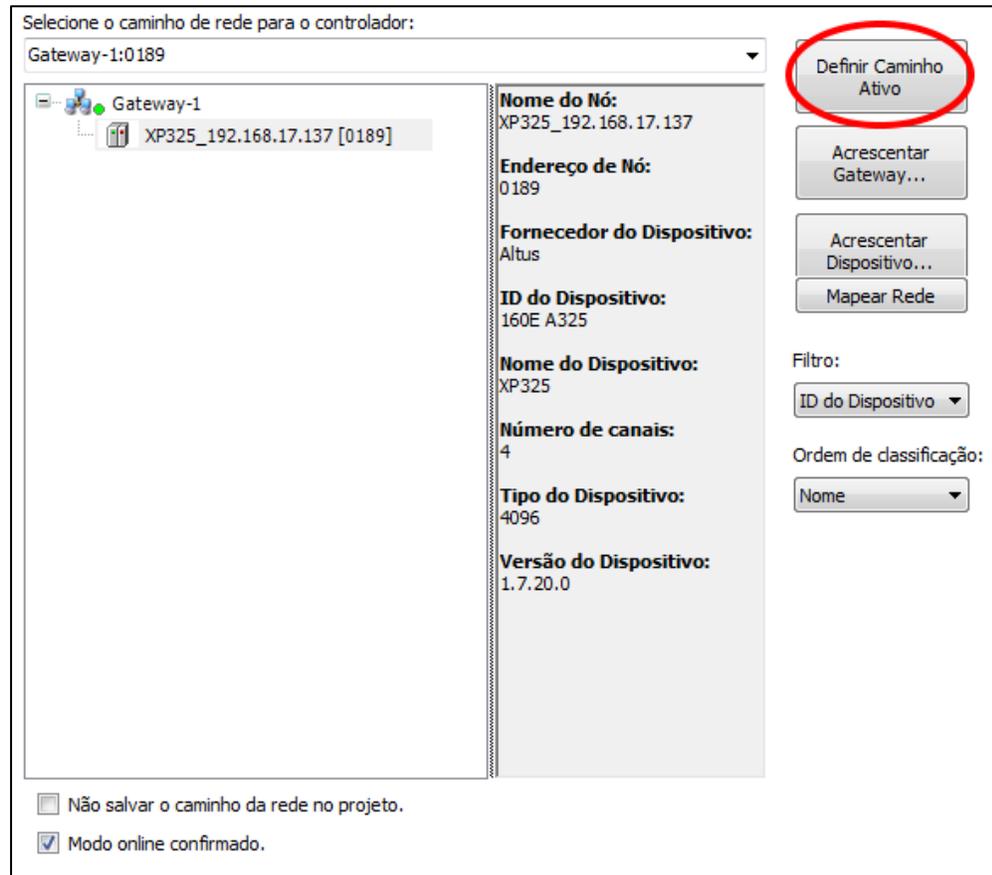


Figura 2-10. Selecionando o Controlador

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

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

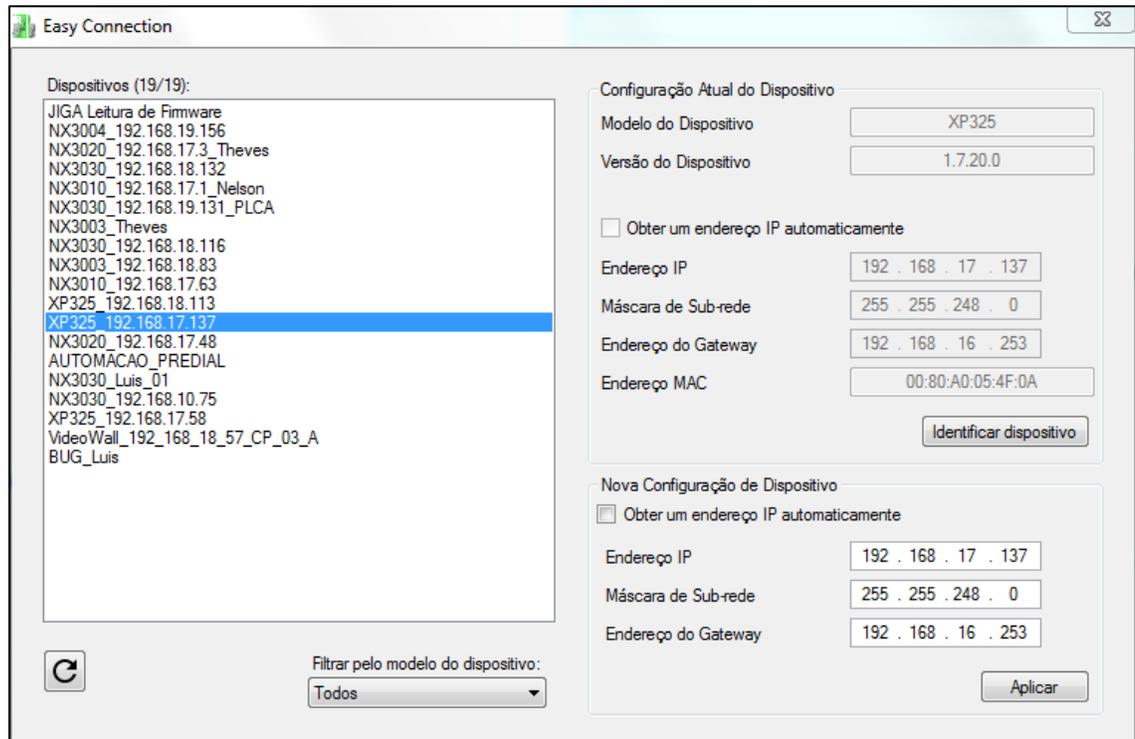


Figura 2-11. Easy Connection

Este comando realiza uma comunicação em nível MAC com o dispositivo, permitindo alterar permanentemente a configuração da interface Ethernet do controlador, independentemente da configuração IP da estação e daquela previamente existente no dispositivo. Com isso, é possível configurar o dispositivo para que fique na mesma rede da interface Ethernet da estação onde está sendo executada a ferramenta MasterTool IEC XE, permitindo localizar e selecionar o dispositivo para comunicação.

A descrição completa do comando Easy Connection pode ser encontrada no Manual de Utilização do MasterTool IEC XE - MU299048.

Login

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

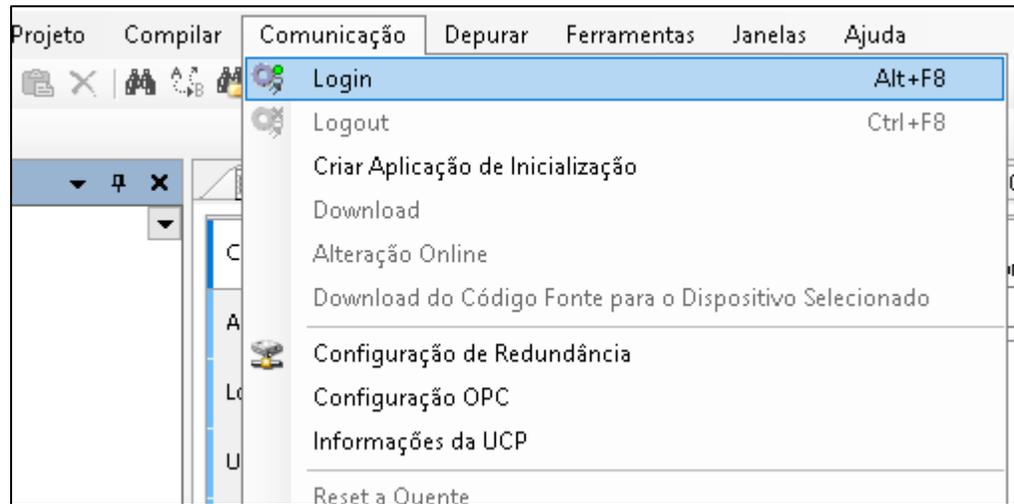


Figura 2-12. Enviando o Projeto para o controlador

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

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

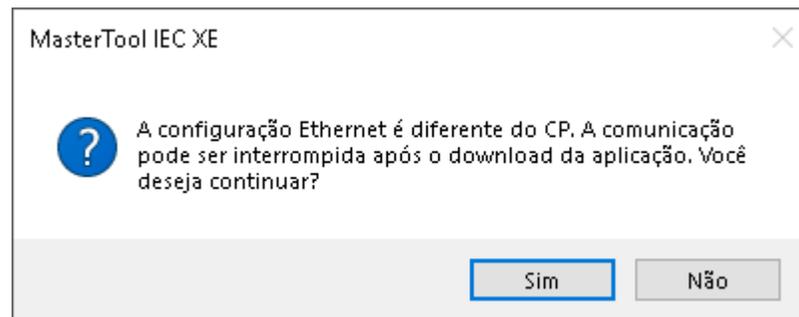


Figura 2-13. Aviso de configuração IP

Se não houver aplicação rodando no controlador, a mensagem ilustrada na figura a seguir será apresentada.

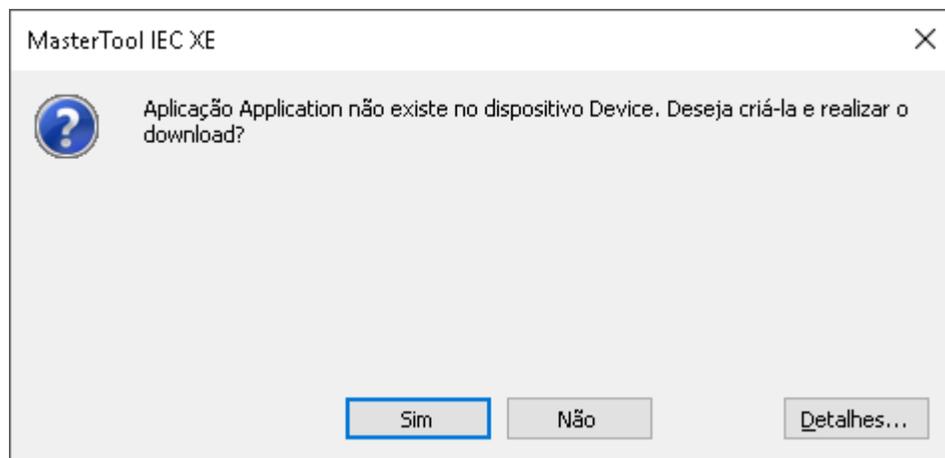


Figura 2-14. Sem aplicação no dispositivo

Caso já exista uma aplicação no controlador, dependendo das diferenças entre os projetos, serão disponibilizadas as seguintes opções:

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

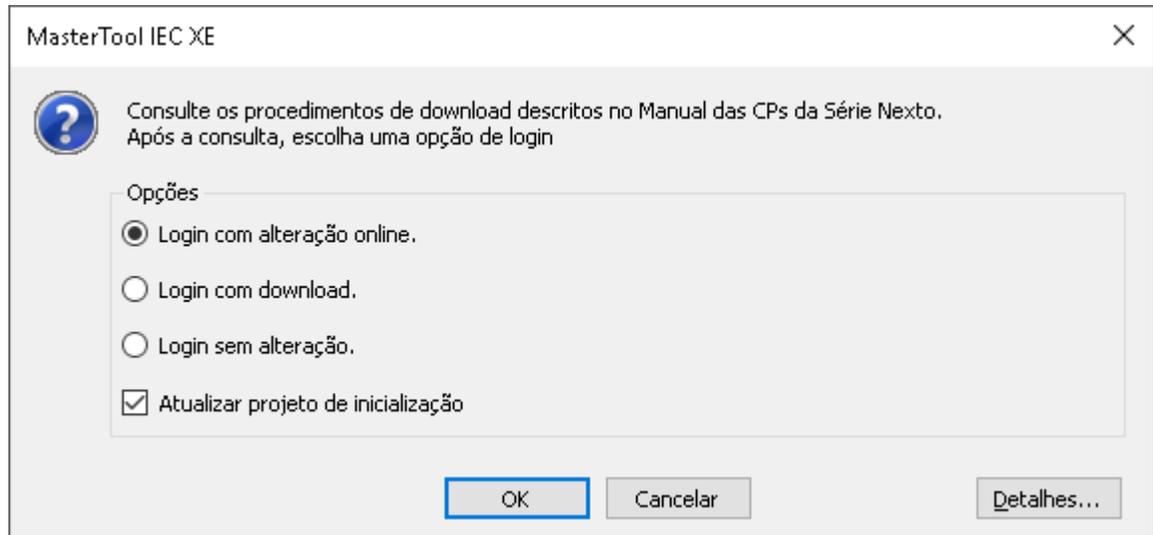


Figura 2-15. Download de Nova Aplicação

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

Caso a nova aplicação tenha sofrido alterações de configuração, a alteração online não será possível. Neste caso, o MasterTool IEC XE solicita ao usuário se o Login deve ser executado como download (parando a execução da aplicação) ou se a operação deve ser cancelada, conforme mostra a figura a seguir. Vale ressaltar que o botão Detalhes... apresenta as modificações realizadas na aplicação.

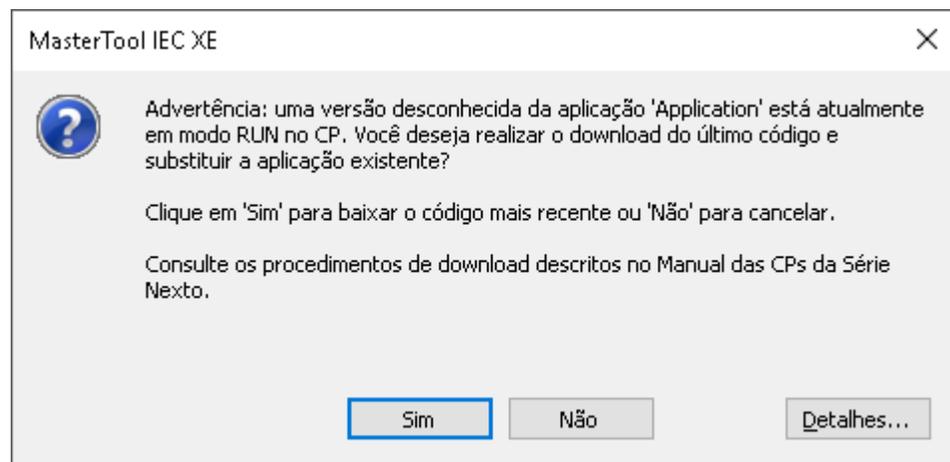


Figura 2-16. Alteração de Configuração

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

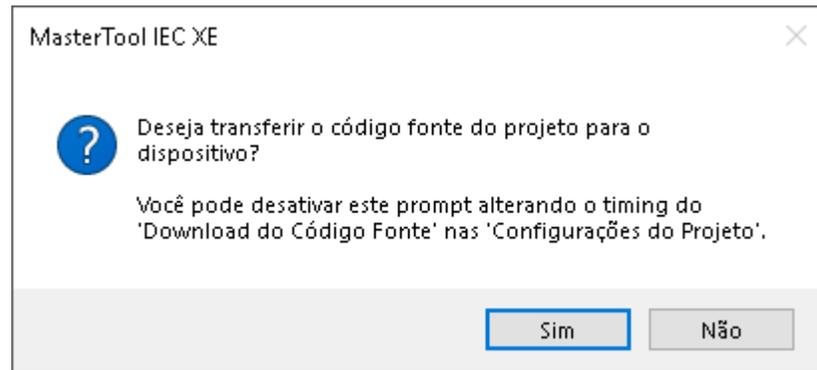


Figura 2-17. Download de código fonte

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.

Modo Run

Logo após enviar o projeto para o controlador, a aplicação não será executada imediatamente (exceto no caso de uma alteração online). Para que isto aconteça, o comando Iniciar deve ser executado. Desta forma, o usuário pode controlar a execução da aplicação enviada para o controlador, permitindo que valores iniciais sejam pré-configurados para serem utilizados no controlador no primeiro ciclo. Para executar este comando, basta acessar o menu Depurar e clicar em Iniciar, conforme mostra a figura a seguir.

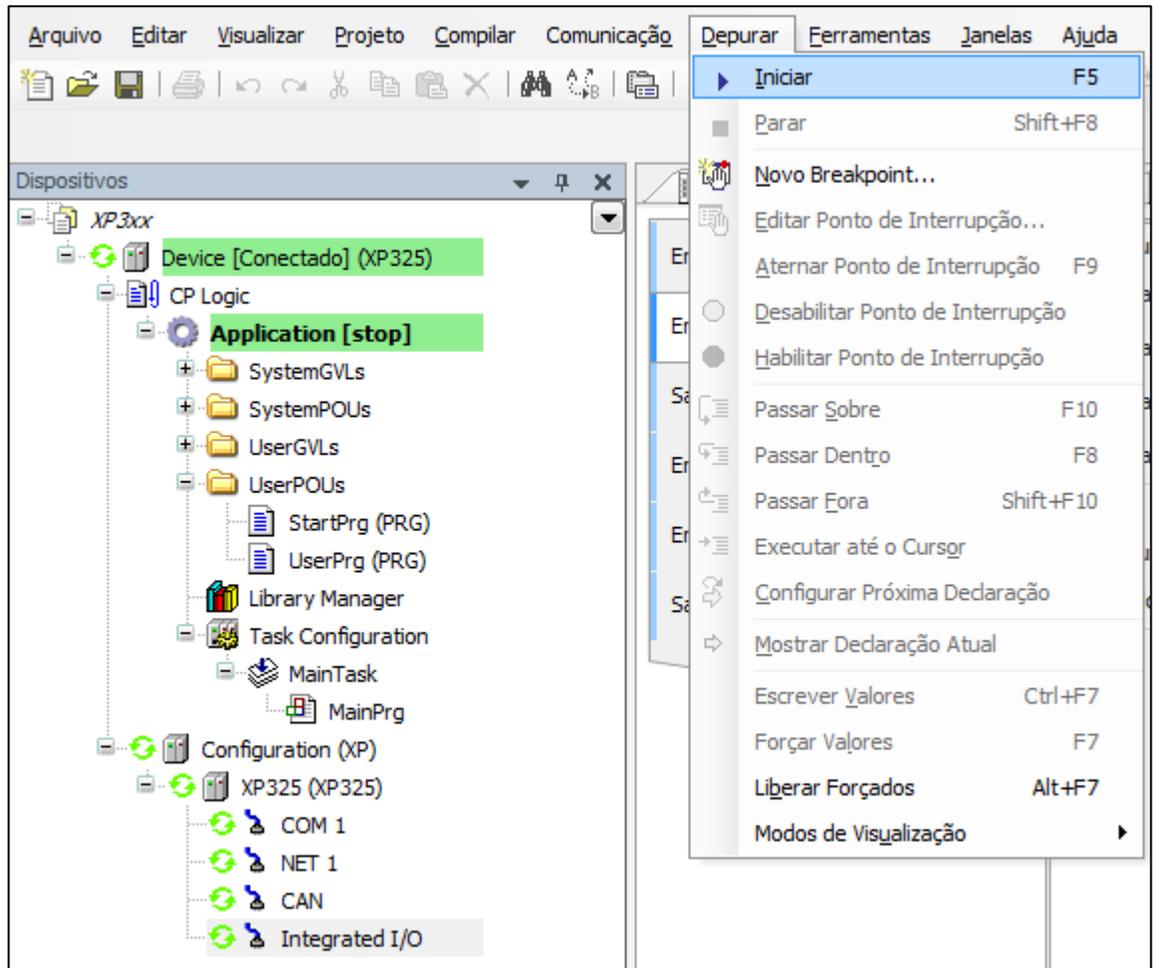


Figura 2-18. Iniciando a Aplicação

A figura a seguir ilustra 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.

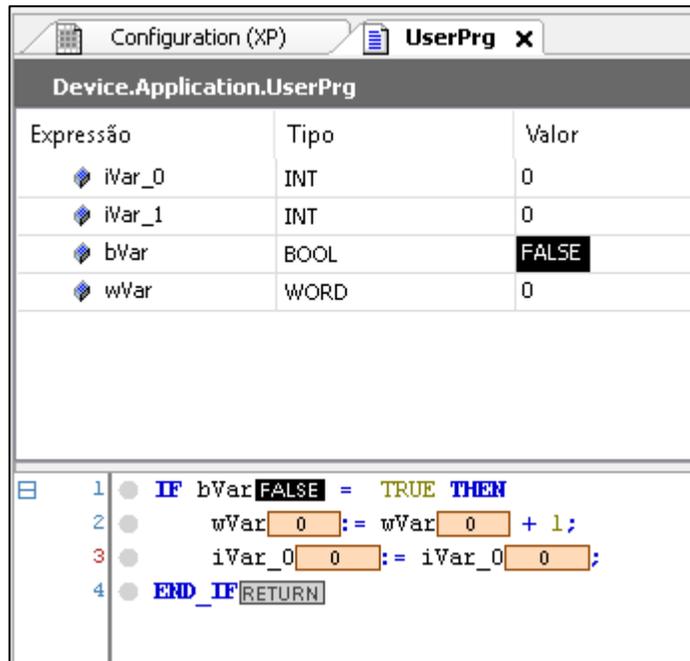


Figura 2-19. Programa em Execução

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

Modo Stop

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

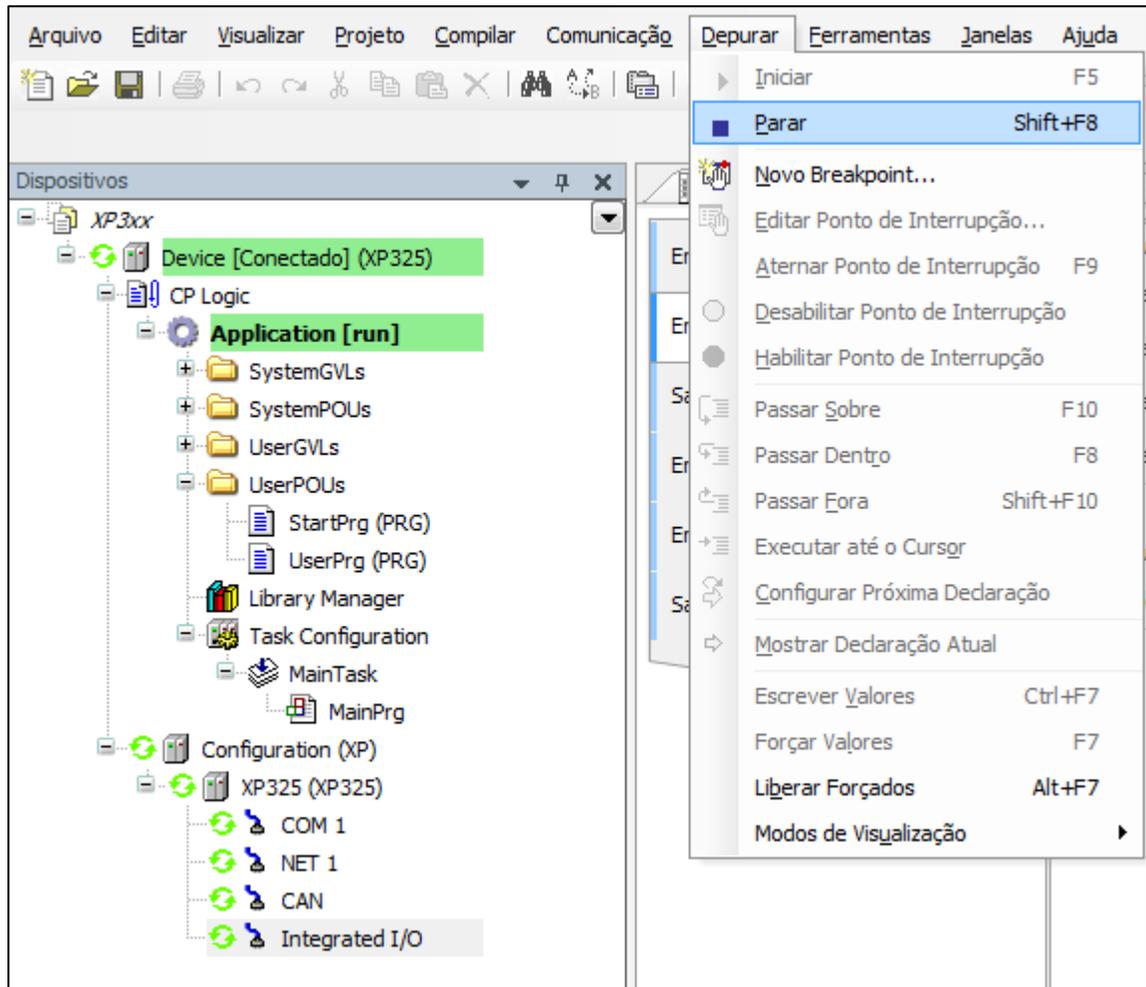


Figura 2-20. Parando a Aplicação

Caso o controlador seja inicializado sem aplicação gravada, ele automaticamente entra em Modo Stop, assim como quando ocorre uma exceção de software.

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 por ele.

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

Logout

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

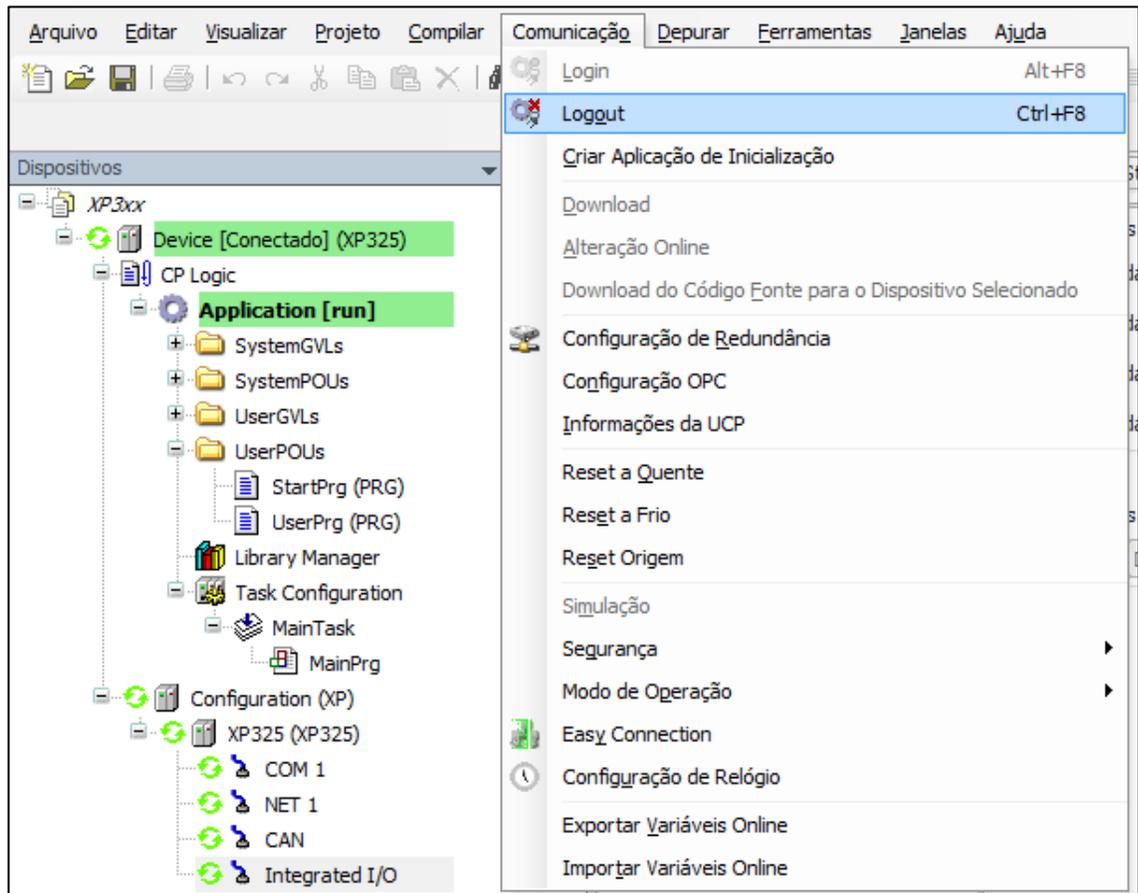


Figura 2-21. Encerrando a comunicação online com o controlador

Upload do Projeto

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

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

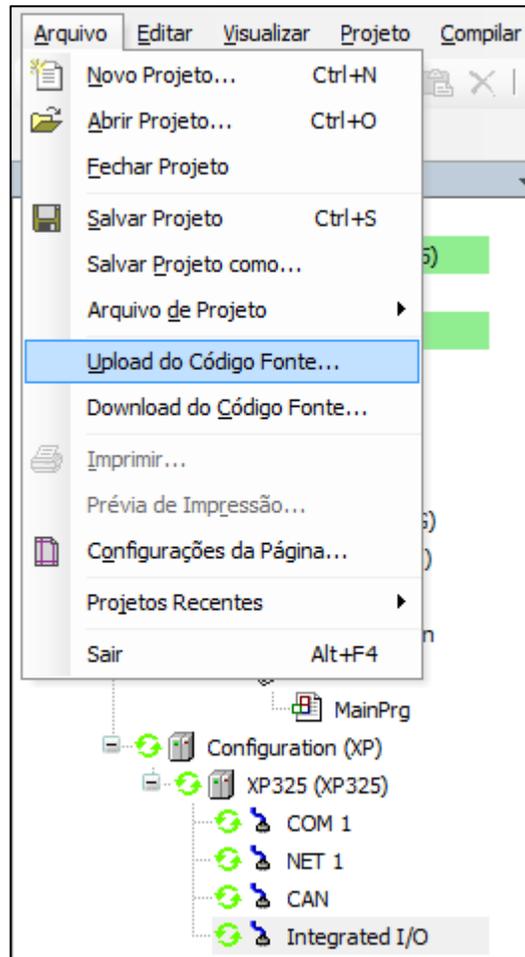


Figura 2-22. Opção de Upload de Projeto

Após, basta selecionar o controlador desejado e clicar em OK, conforme ilustrado na sequência.

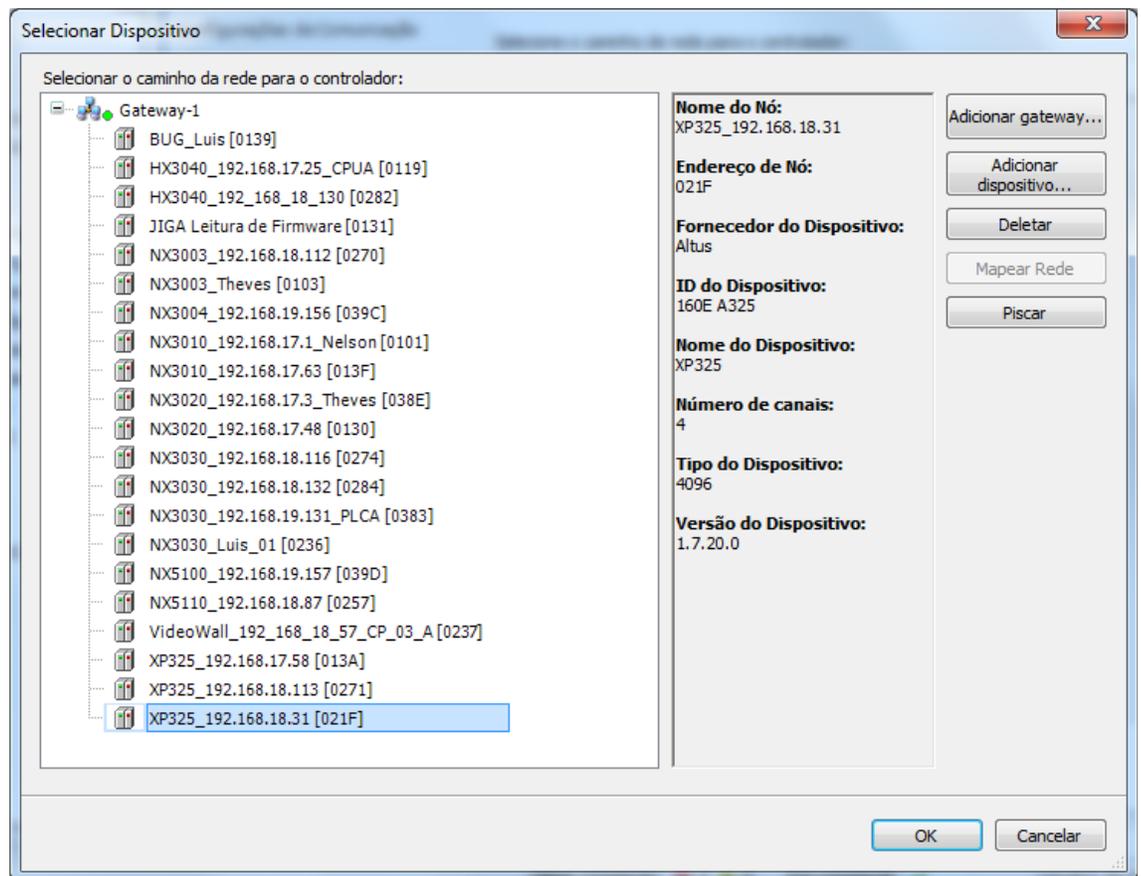


Figura 2-23. Selecionando o controlador

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

O tamanho da área de memória para armazenar um projeto nos controladores Nexto Xpress está definido na documentação do produto.

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.

Estados de Operação da UCP

Run

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

Stop

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

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

Breakpoint

Quando uma marca de depuração é atingida em uma tarefa, essa tarefa é interrompida. Todas as demais tarefas ativas na aplicação não serão interrompidas, continuando a sua execução. Com este recurso, é possível percorrer e investigar a execução de um programa passo a passo no modo Online conforme as posições de interrupção incluídas através do editor. Para maiores informações sobre a utilização de marcas de depuração, favor consultar o Manual de Utilização do MasterTool IEC XE – MU299048.

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.

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).

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).

Reset Origem

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



NOTAS:

1. Reset: quando um Reset é executado, os breakpoints definidos na aplicação são desabilitados.
2. Comando: para executar um comando de Reset a Quente, Reset a Frio ou Reset Origem, é necessário estar com o MasterTool em modo Online no controlador.

Reset Process Command (IEC 60870-5-104)

Este comando de reset do processo pode ser solicitado pelos clientes IEC 60870-5-104. Depois de responder ao cliente, a UCP entra num procedimento de reinicialização, como se estivesse sendo executado um ciclo de energização. A norma IEC 60870-5-104 prevê a passagem de um valor de qualificação (0..255) com o comando de reset do processo, mas este parâmetro não é considerado pela UCP.

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

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

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

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

PROGRAM MainPrg
VAR
  isFirstCycle : BOOL := TRUE;
END_VAR

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

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

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.

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.

GVL IntegratedIO

A GVL IntegratedIO contém as variáveis correspondentes aos canais físicos integrados de entrada e saída. A figura seguinte mostra um exemplo da apresentação desta GVL quando em modo Online.

Device.Application.IntegratedIO				
Expression	Type	Value	Prepared value	Address
[-] DigitalInputs	T_DIGITAL_INPUTS_1			
I00	BOOL	FALSE		
I01	BOOL	FALSE		
I02	BOOL	FALSE		
I03	BOOL	FALSE		
I04	BOOL	FALSE		
I05	BOOL	FALSE		
I06	BOOL	FALSE		
I07	BOOL	FALSE		
I10	BOOL	FALSE		
I11	BOOL	FALSE		
I12	BOOL	FALSE		
I13	BOOL	FALSE		
I14	BOOL	FALSE		
I15	BOOL	FALSE		
I16	BOOL	FALSE		
I17	BOOL	FALSE		
[-] DigitalOutputs	T_DIGITAL_OUTPUT...			
Q00	BOOL	FALSE		
Q01	BOOL	FALSE		
Q02	BOOL	FALSE		
Q03	BOOL	FALSE		
Q04	BOOL	FALSE		
Q05	BOOL	FALSE		
Q06	BOOL	FALSE		
Q07	BOOL	FALSE		
Q10	BOOL	FALSE		
Q11	BOOL	FALSE		
Q12	BOOL	FALSE		
Q13	BOOL	FALSE		
Q14	BOOL	FALSE		
Q15	BOOL	FALSE		
Q16	BOOL	FALSE		
Q17	BOOL	FALSE		
[+] FastInputs	T_FAST_INPUTS_1			
[+] FastOutputs	T_FAST_OUTPUTS_1			
[-] AnalogInputs	T_ANALOG_INPUTS_1			
A10	INT	0		
A11	INT	0		
A12	INT	0		
A13	INT	0		
A14	INT	0		
[-] AnalogOutputs	T_ANALOG_OUTPUT...			
A00	INT	0		
A01	INT	0		
A02	INT	0		
A03	INT	0		
[-] RtdInputs	T_RTD_INPUTS_1			
RI0	INT	0		
RI1	INT	0		

Figura 2-24. GVL IntegratedIO quando em modo Online

GVL System_Diagnostics

Na GVL System_Diagnostics estão presentes as variáveis de diagnóstico da UCP do controlador, das interfaces de comunicação e E/S. Essa GVL não é editável e as variáveis são declaradas automaticamente com tipo especificado pelo dispositivo ao qual pertence quando este é adicionado ao projeto. Na GVL System_Diagnostics também são declaradas as variáveis de diagnóstico das requisições MODBUS Mestre/Cliente por representação direta.

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

Device.Application.System_Diagnostics				
Expression	Type	Value	Prepared value	Address
[-] DG_XP325	T_DIAG_...			
[-] tSummarized	T_DIAG_...			
[-] bHardwareFailure	BIT	FALSE		
[-] bSoftwareException	BIT	FALSE		
[-] bCOM1ConfigError	BIT	FALSE		
[-] bNET1ConfigError	BIT	FALSE		
[-] bInvalidDateTime	BIT	FALSE		
[-] bRuntimeReset	BIT	FALSE		
[-] bRetentivityError	BIT	FALSE		
[-] bIntegratedIODiagnostic	BIT	FALSE		
[-] tDetailed	T_DIAG_...			
+ Target	T_DIAG_...			
+ Hardware	T_DIAG_...			
+ Exception	T_DIAG_...			
+ RetainInfo	T_DIAG_...			
+ Reset	T_DIAG_...			
+ Serial	T_DIAG_...			
+ CAN	T_DIAG_...			
+ USB	T_DIAG_...			
+ Ethernet	T_DIAG_...			
+ UserFiles	T_DIAG_...			
+ UserLogs	T_DIAG_...			
+ Application	T_DIAG_...			
+ ApplicationInfo	T_DIAG_...			
+ SNTP	T_DIAG_...			
[-] IntegratedIO	T_DIAG_I...			
+ AnalogInputs	T_DIAG_...			
+ AnalogOutputs	T_DIAG_...			
+ RTDInputs	T_DIAG_...			

Figura 2-25. GVL System_Diagnostics em Modo Online

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 estrutura indicada a seguir.

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

Onde:

- Nome do dispositivo: nome que aparece na visualização em árvore para o dispositivo MODBUS;
- Número da Requisição: número da requisição que foi declarada na tabela de requisições do dispositivo MODBUS seguindo a sequência de cima para baixo, começando em 0001.

Exemplo (Device.Application.Disables):

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

A geração automática através do botão Gerar Variáveis de Desabilitação apenas cria variáveis, e não remove automaticamente. Desta forma, caso alguma relação seja removida, a sua respectiva variável de desabilitação deve ser removida manualmente. A GVL Disables é editável, portanto, as variáveis de desabilitação das requisições podem ser criadas manualmente não precisando seguir o modelo criado pela declaração automática e podem ser usadas as duas maneiras ao mesmo tempo, mas devem sempre ser do tipo BOOL e deve-se tomar o cuidado para não excluir ou alterar as variáveis declaradas automaticamente, pois elas podem estar sendo usadas por algum dispositivo MODBUS. Se a variável for excluída ou alterada será gerado um erro ao compilar o projeto. Para corrigir o nome de uma variável declarada automaticamente, deve-se seguir o modelo exemplificado acima de acordo com o dispositivo e a requisição aos quais pertence. A figura a seguir mostra um exemplo da apresentação desta GVL quando em modo Online. Se o valor da variável for TRUE significa que a requisição, à qual a variável pertence, está desabilitada e o inverso é válido para quando o valor da variável for FALSE.

Device.Application.Disables			
Expression	Type	Value	Prepared
 MODBUS_Slave_1_DISABLE_0001	BOOL	FALSE	
 MODBUS_Slave_1_DISABLE_0002	BOOL	TRUE	
 MODBUS_Slave_1_DISABLE_0003	BOOL	FALSE	
 MODBUS_Slave_1_DISABLE_0004	BOOL	FALSE	
 MODBUS_Server_1_DISABLE_0001	BOOL	FALSE	
 MODBUS_Server_1_DISABLE_0002	BOOL	FALSE	
 MODBUS_Server_1_DISABLE_0003	BOOL	FALSE	
 MODBUS_Server_1_DISABLE_0004	BOOL	TRUE	

Figura 2-26. GVL Disable em Modo Online

GVL Qualities

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

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

Onde:

- Nome do dispositivo: nome que aparece na TreeView para o dispositivo;
- Número do Mapeamento: número do mapeamento que foi declarado na tabela de mapeamentos do dispositivo seguindo a sequência de cima para baixo, começando em 0001.

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

Exemplo (Device.Application.Qualities):

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

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

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

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

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

Figura 2-27. GVL Qualities em Modo Online

GVL ReqDiagnostics

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

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

Onde:

- Nome do dispositivo: nome que aparece na visualização em árvore para o dispositivo;
- Número da Requisição: número da requisição que foi declarada na tabela de requisições do dispositivo seguindo a sequência de cima para baixo, começando em 0001;
- Tipo da Variável: NXMODBUS_DIAGNOSTIC_STRUCTS. T_DIAG_MODBUS_RTU_MAPPING_1 para MODBUS Master e NXMODBUS_DIAGNOSTIC_STRUCTS. T_DIAG_MODBUS_ETH_MAPPING_1 para MODBUS Cliente.

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 2-28. GVL ReqDiagnostics em Modo Online

Pasta de Eventos do Sistema (SystemEvents)

Esta pasta, localizada logo abaixo do item Application, no diretório de dispositivos contém algumas funções associadas a eventos do sistema, as quais estão descritas na sequência.

Função Prepare_Start

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

Função Prepare_Stop

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

Função Start_Done

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

Função Stop_Done

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

Esta aparência depende da estruturação das janelas que podem ser alteradas a qualquer momento pelo usuário (ações relacionadas à minimização/restauração, redimensionamento do tamanho ou fechamento de janelas).

Interface do Usuário do MasterTool IEC XE



A interface de programação do usuário MasterTool IEC XE é estruturada através de “componentes” (veja abaixo na figura).

Esta aparência depende da estruturação das janelas que podem ser alteradas a qualquer momento pelo usuário (ações relacionadas à minimização/restauração, redimensionamento do tamanho ou fechamento de janelas).

A interface do usuário fornece menus e barras de ferramentas, organização de editores e objetos, janelas de monitoração e mensagens, além de uma linha de informação e status.

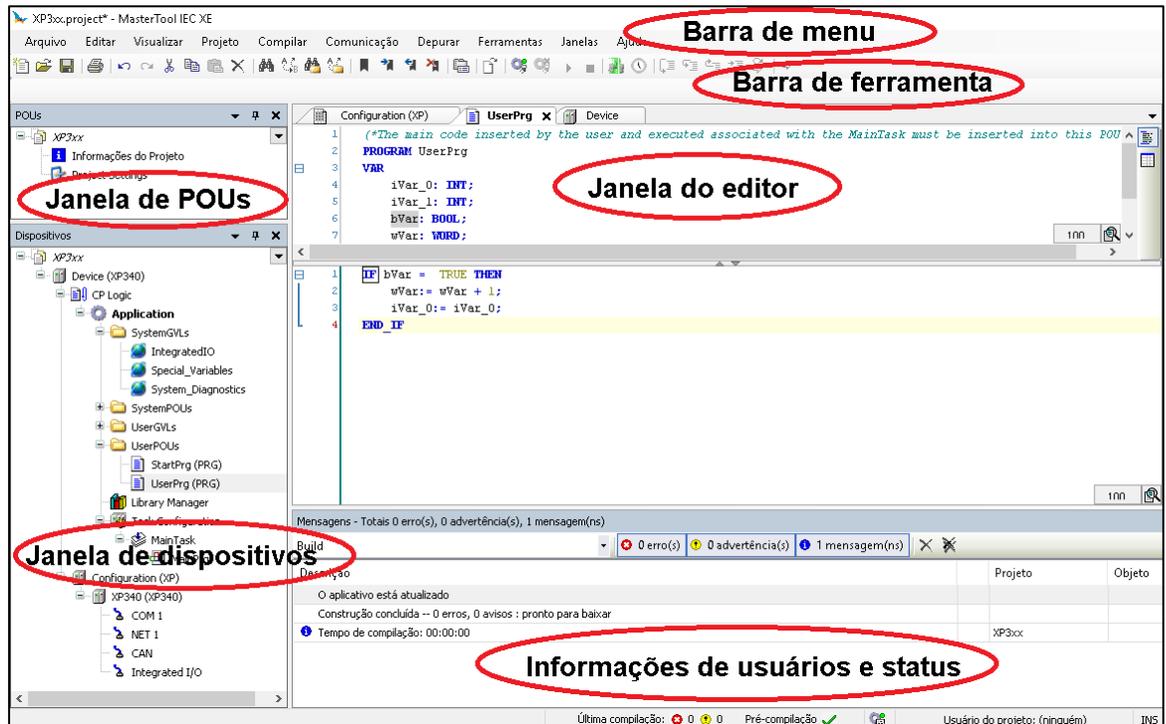


Figura 2-29. Exemplo do MasterTool IEC XE Interface do Usuário

Os componentes padrão são listados a seguir:

Barra de menu: fornece menus com todos os comandos atualmente disponíveis.

Barra de ferramentas: contém botões de ferramentas para todas as ferramentas atuais disponíveis.

Janela de POU's: organiza as unidades de programação (POUs, DUTs, etc.) de um projeto em uma estrutura de árvore (aberta no menu Visualizar).

Janela de dispositivos: organiza os objetos dos recursos dos dispositivos de um projeto em uma estrutura de árvore (aberta no menu Visualizar).

Janela do editor: cria um objeto específico no respectivo editor. No caso de editores de linguagem (por exemplo, Editores ST ou CFC), em geral, estas janelas estão na parte inferior, enquanto no editor de declaração estão na parte superior. No caso de outros editores, esta janela também fornece diálogos (por exemplo, Editor de tarefas, Editor de dispositivos, Editor de UCP). O nome das POU's ou dos objetos dos recursos sempre são exibidos na barra de títulos. Os objetos podem ser abertos na janela do editor nos modos online ou off-line através do comando Editar Objeto.

Nos componentes abaixo, encontram-se mais informações sobre o projeto nos modos online ou off-line:

Janela de Mensagens: os comandos Precompilar, Compilar, Montar, Enviar mensagens, entre outros, são exibidos nesta janela.

Janelas de Monitoração e visualizações online dos editores: mostram uma visualização da monitoração de uma POU e uma lista de expressões de monitoração definida pelo usuário.

Linha de informação e status: a linha na borda inferior da interface do usuário fornece informações sobre o usuário conectado no momento. A posição do cursor e o status do modo de edição também serão exibidos se o usuário estiver trabalhando em uma janela do editor. No modo online, é indicado o status do programa.

Usuário Atual: cada projeto tem um usuário e gerenciamento de acesso. O usuário conectado no momento será nomeado na linha de status.

Posição: considerada a partir da margem superior esquerda da janela do editor em:

- Ln = número de linhas.
- Col = número de colunas (uma coluna inclui exatamente 1 espaço, caractere ou dígito).
- Ch = número de caracteres (neste contexto, um caractere pode ser um dígito ou caractere único, assim como também pode ser uma guia incluindo, por exemplo, 4 colunas).

Com um duplo clique em um dos campos é exibido o diálogo Ir para a linha, onde o usuário pode direcionar o cursor para outra posição.

Status do modo de edição: INS refere-se ao modo de inserção e OVR refere-se ao modo de sobrescrita. Com um duplo clique neste campo o usuário alterna entre as duas configurações.

Informação no modo online: status da aplicação no dispositivo conforme descrito a seguir:

- **RUN** = programa em execução.
- **STOP** = programa parado.
- **HALT ON BP** = programa interrompido em um breakpoint.
- Programa Carregado = o programa está carregado em um dispositivo.
- Programa Inalterado = o programa no dispositivo coincide com o programa do programador.
- Programa Modificado (alteração online) = o programa no dispositivo é diferente do programador e requer uma alteração online.
- Programa Modificado (envio completo) = o programa no dispositivo difere do programador e um envio completo se faz necessário.

Janelas, Visualizações e Janelas do Editor

As janelas podem ser exibidas dentro ou fora da moldura da interface do usuário. Embora à primeira vista pareçam iguais, existem dois tipos de janelas:

Algumas podem ser encolhidas para qualquer margem da moldura ou podem ser posicionadas na tela como janelas inteiras, independentemente da moldura da janela. Elas também podem estar "ocultas" - situação representada por uma guia na borda da moldura. Estas janelas exibem informações que não dependem de um único objeto do projeto, por exemplo, Mensagens, Dispositivos, POUs, Caixa de Ferramentas. Elas podem ser acessadas através dos comandos do menu *Visualizar* e são chamadas "visualizações". A maioria das visualizações inclui uma barra de ferramentas não configurável com botões para classificar, visualizar e pesquisar na janela.

Outras abrem na visualização ou na edição de um objeto específico do projeto no respectivo editor. Estas janelas são exibidas em uma área do editor, ou dependendo das configurações da interface do usuário. Estas janelas não podem ser ocultadas nem encolhidas na moldura da janela e podem ser acessadas através dos comandos do menu *Janelas*.

Tipos adicionais de janelas ou visualizações podem ser acrescentados através de componentes específicos do fornecedor.

Customizando a Interface do Usuário

A aparência da interface do usuário, ou seja, a estruturação e a configuração dos seus componentes específicos, dependem dos itens abaixo:

Pré-configurações padrão para menus, funções de teclado e barras de ferramentas. As configurações padrão vêm instaladas no MasterTool IEC XE.

Propriedades de um editor definidas nos respectivos diálogos de Opções: estas pré-configurações podem ser igualmente substituídas pelo usuário e a configuração atual também será salva no sistema local.

Organização das visualizações ou das janelas de edição no projeto, feitas pelo usuário: as posições atuais serão salvas com o projeto.

Organizando Barras de Menu e Barras de Ferramentas

A barra de menu está sempre posicionada no topo da interface do usuário, entre a barra de título e as janelas de visualização. Uma barra de ferramentas pode ser posicionada na mesma área que a barra de menu (fixa) ou como uma janela independente em qualquer lugar da tela.

Para reposicionar uma barra, clique com o cursor na linha pontilhada na extremidade final esquerda da barra e, mantendo-o pressionado, conduza a barra até a posição desejada.



Figura 2-30. Linha Pontilhada no Menu e Barras de Ferramentas



NOTA: Nas janelas de visualização, como por exemplo, POUs, Dispositivos ou Propriedades de elementos visuais, está disponível uma barra de ferramentas especial que fornece botões para classificação, visualização e localização. Esta barra não pode ser configurada.

Zoom

Cada janela dos editores das linguagens de programação apresentam uma função de zoom (), exceto no editor da linguagem ST. Este botão, localizado no canto inferior direito da janela, abre uma lista com os níveis permitidos: 25, 50, 75, 100, 150, 200, 400 %. A impressão refere-se à visualização em 100%. O usuário também pode definir o nível de zoom de acordo com o seu interesse, bastando para isto digitar o valor desejado no campo correspondente.

Interface do Usuário no Modo Online

Assim que o login for realizado no projeto, todos os objetos abertos no modo off-line serão automaticamente visualizados no modo online.

Para abrir um objeto no modo online (já aberto no modo off-line), execute um duplo clique no mesmo, na janela das POUs ou dispositivos ou utilize o comando *Editar Objeto*.

Se este procedimento foi executado corretamente, o objeto será aberto no modo online. Caso contrário, por exemplo, se haviam várias instâncias do objeto selecionado (blocos funcionais, etc.) contidas no projeto, o diálogo *Selecionar Estado Online <nome do objeto>* será exibido para que se possa escolher o que deve ser visualizado (uma instância ou a implementação básica do objeto). Neste diálogo, também se define o modo de exibição do objeto (modo online ou off-line).

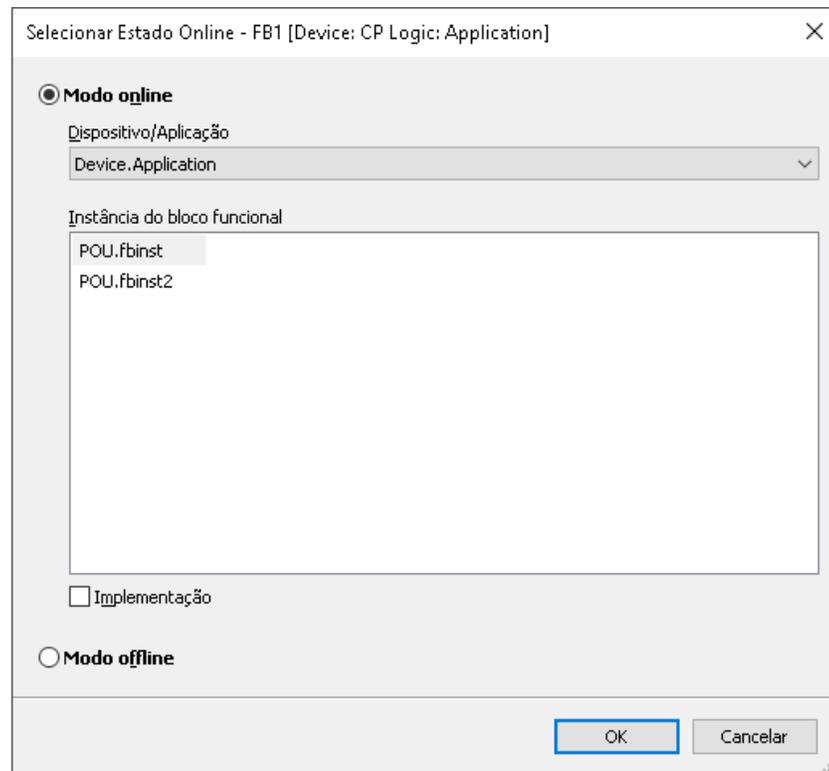


Figura 2-31. Diálogo de Estado Online

Nos campos *Dispositivo/Aplicação* veja o Dispositivo e a Aplicação aos quais o respectivo objeto está associado.

Para abrir a visualização online do objeto, ative a opção *Modo Online* e pressione *OK*. Para a visualização off-line, ative a opção *Modo Offline*.

Se o objeto é um bloco funcional, no campo *Instância do bloco funcional* haverá uma lista de todas as instâncias atuais usadas na aplicação. Neste caso, o usuário pode selecionar uma das instâncias e ativar os modos online ou off-line, ou selecionar a opção *Implementação*, que, independentemente da instância selecionada, abrirá a visualização da implementação básica do bloco funcional. A implementação não tem efeito para objetos não instanciados.

Para mais informações sobre visualizações online dos editores específicos, consulte a descrição dos respectivos editores.

A barra de status fornecerá informações sobre o status atual da aplicação.

Menus e Comandos Padronizados

Veja a seguir uma visão geral sobre a estrutura dos menus principais e comandos.

Os comandos especiais para um determinado editor estarão disponíveis no menu correspondente quando este editor for aberto (exemplo: ao editar um objeto no editor *SFC*, o menu *SFC* será acrescentado à barra de menu).

Figura

Arquivo Editar Visualizar Projeto Compilar Comunicação Depurar Ferramentas Janelas Ajuda

Figura 2-32. Barra de Menu Padrão



Pesquisa na Documentação: menus e comandos padronizados

Consulte, na documentação do produto, os menus e comandos padronizados para cada um dos itens da barra de menu padrão.

Fonte: Manual de Utilização Manual de Utilização MasterTool IEC XE (MT8500) - MU299048.

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 projeto na memória da UCP utilizada. Essa área de memória irá variar de acordo com o modelo de UCP utilizada.

Essa funcionalidade é acessada com um duplo-clique sobre o item *Device* e selecionando a aba *Arquivos*, conforme mostram as indicações da figura a seguir.

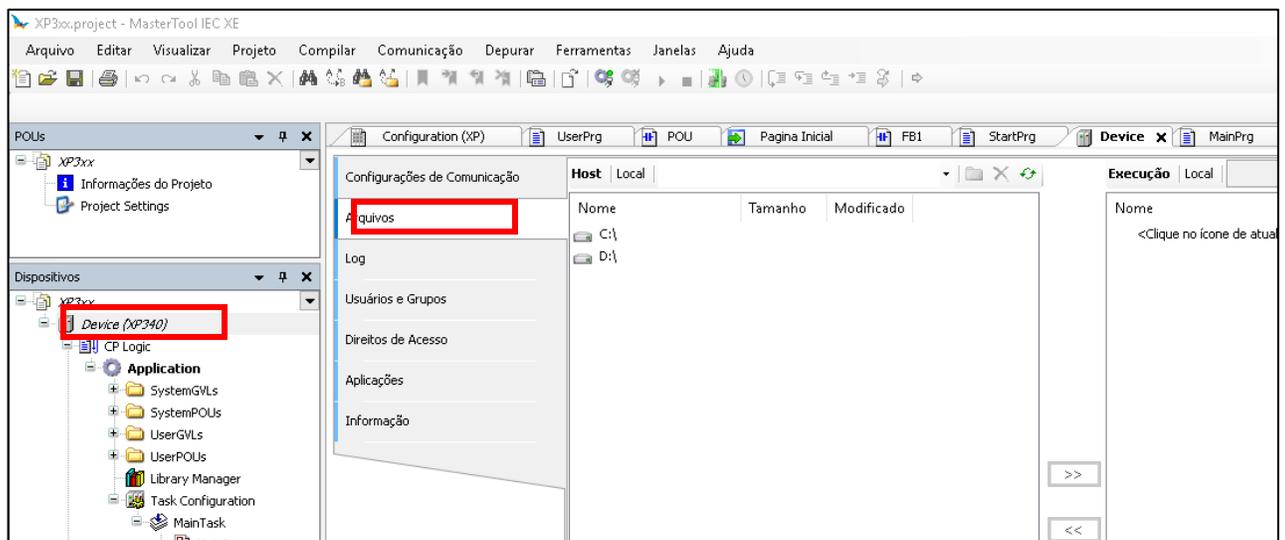


Figura 2-33. Utilização da Memória de Arquivos de Usuário

Gerenciamento de Usuários e Direitos de Acesso

Esta funcionalidade fornece funções para definir contas dos usuários e configurar os direitos de acesso ao projeto. Observe que o gerenciamento de usuários específico do dispositivo deve ser suportado para controlar os direitos de acesso no sistema de arquivos do CP e objetos durante a execução. Os direitos para acessar os objetos do projeto via ações especificadas são atribuídos somente a grupos e, portanto, cada usuário deve pertencer a um grupo.



Pesquisa na Documentação: gerenciamento de usuários e direitos de acesso

Consulte, na documentação do produto, as configurações associadas ao gerenciamento de usuários e direitos de acesso, tanto do projeto quanto da UCP, incluindo as opções de usuários e grupos.

Fonte: Manual de Utilização Manual de Utilização MasterTool IEC XE (MT8500) - MU299048.

Editores

O MasterTool IEC contempla vários editores para configuração de objetos, dispositivos, parametrização de rede e outros tipos de ajustes. Cada tipo de editor possui as suas peculiaridades. Entretanto algumas características são gerais e válidas para todos os editores.

Nos editores sempre que existirem campos numéricos sendo configurados estes campos possuem um limite mínimo e máximo de valor que depende da funcionalidade do campo. Por exemplo, um campo de time-out pode ter valor mínimo de 10 s e o valor máximo de 65535 s. Nesses casos não é possível configurar valores fora dessa faixa e esta consistência é executada durante a configuração. Para alguns campos numéricos esta consistência é executada em uma das etapas de geração de código, isso também se dá quando existe dependência entre diferentes campos, como por exemplo entre o tempo de ciclo de uma tarefa e o seu valor de cão-de-guarda.

Já nos casos que parâmetros representam endereços de mapeamentos diretos nas áreas de %I, %Q e %M a consistência é executada apenas durante a geração de código. Como os projetos podem ser feitos para quaisquer modelos de UCP e estes possuem tamanhos de áreas diferentes os campos que representam faixas de endereço dentro destas áreas estão limitados entre 0 e 2147483647. Durante a geração de código os valores atribuídos na configuração são consistidos com os limites disponíveis para cada modelo de UCP e caso estejam fora destes limites uma mensagem de erro é gerada durante o processo.

Em módulos de entrada e saída a guia Mapeamentos de E/S não mostrará os mapeamentos com seus valores atuais quando ela se encontrar aberta e o comando Reset Origem for executado, exibirá <Bad> na coluna Valor Atual. Para que os valores sejam devidamente exibidos e atualizados deve-se realizar o comando Download (menu Comunicação), fechar e abrir a janela do editor do módulo. Para mais informações consulte Reset Origem e Download.

Os editores disponíveis no programador estão listados na sequência.

▷ Editor de Barramento	▷ Editor Recipe Manager
▷ Editor de Módulos de E/S Digital	▷ Editor Recipe Definition
▷ Bill of Materials	▷ Editor de Tarefas
Configuration and Consumption	▷ Editor Trace
Editores das Linguagens de Programação	▷ Editor de Lista de Monitoração
▷ Editor de Declaração	▷ Editores MODBUS
▷ Editor de Dispositivos	Editor PROFIBUS
▷ Editores para Comunicação OPC	Editor UCP
Editor DUT	Interfaces Seriais
Editor FBD/LD/IL	Interfaces Ethernet
Editor de Lista de Variáveis Globais	▷ Editor do Controle PID
▷ Editor Library Manager	▷ Visualização

Figura 2-34. Editores

Pesquisa na Documentação: editores

Consulte, na documentação do produto, o detalhamento dos editores listados anteriormente, identificando as características comuns a todos e as peculiaridades de cada um.

Fonte: Manual de Utilização Manual de Utilização MasterTool IEC XE (MT8500) - MU299048.



Primeiros Passos com o MasterTool IEC XE

O objetivo desta seção é indicar os passos básicos para a programação do Nexto Xpress. Serão explorados os seguintes itens:

- Criar e executar projetos;
- Obter ajuda.

Iniciando o MasterTool IEC XE

A partir do menu Iniciar do PC selecione a opção MasterTool IEC XE. A inicialização também pode ser feita através do ícone MasterTool IEC XE que está disponível no desktop após a instalação. Inicialmente, o usuário deverá criar um novo projeto no MasterTool IEC XE a partir do menu Arquivo e logo em seguida, Novo Projeto..., conforme mostra a figura a seguir.

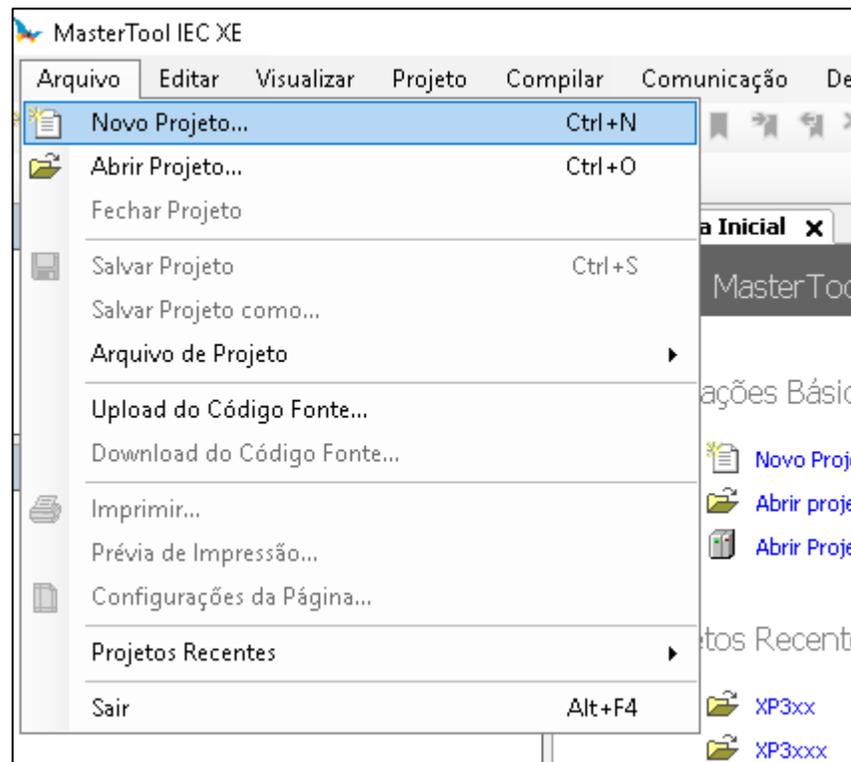


Figura 2-35. Novo Projeto

Após, uma janela será apresentada ao usuário, solicitando que ele selecione o tipo de projeto que deseja fazer e, em seguida, escreva um nome e a localização para armazenar o projeto no computador. Clicar em *OK* para prosseguir. A opção de cancelamento também está disponível nessa janela via botão *Cancelar*.

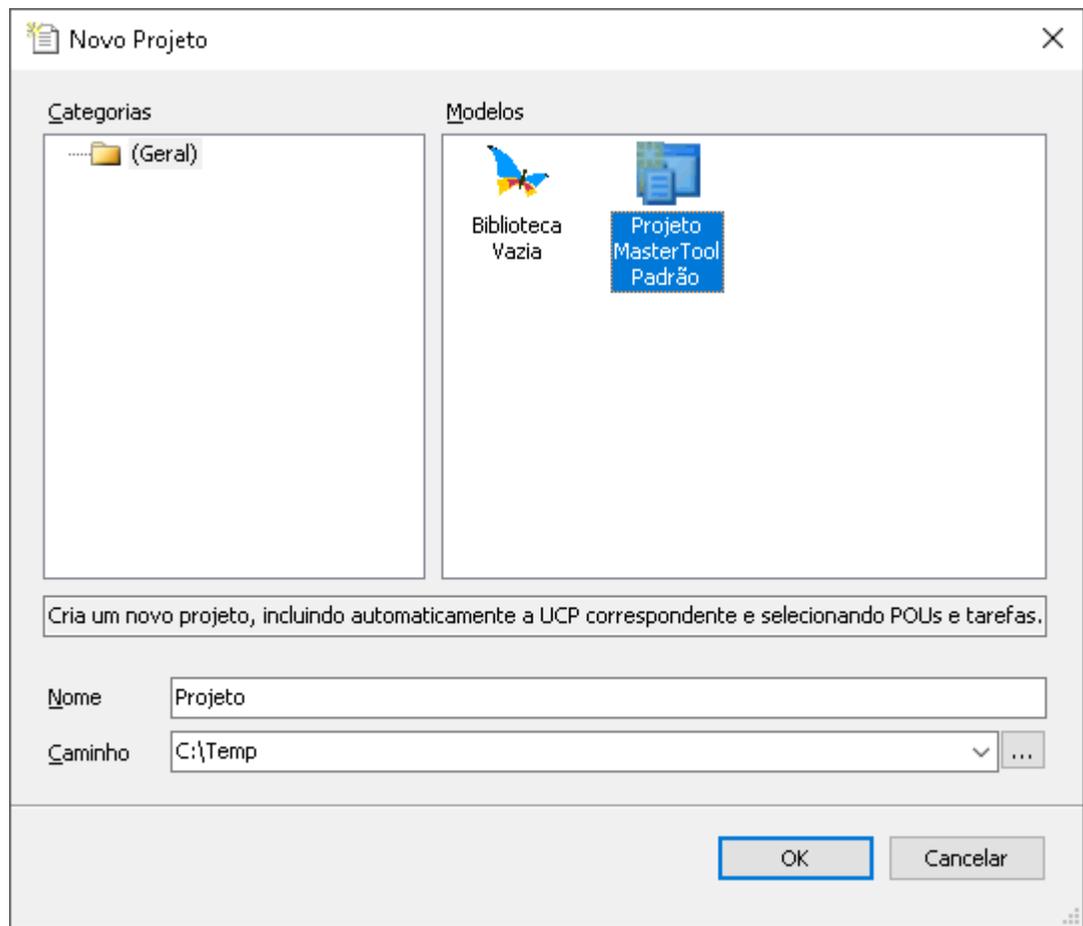
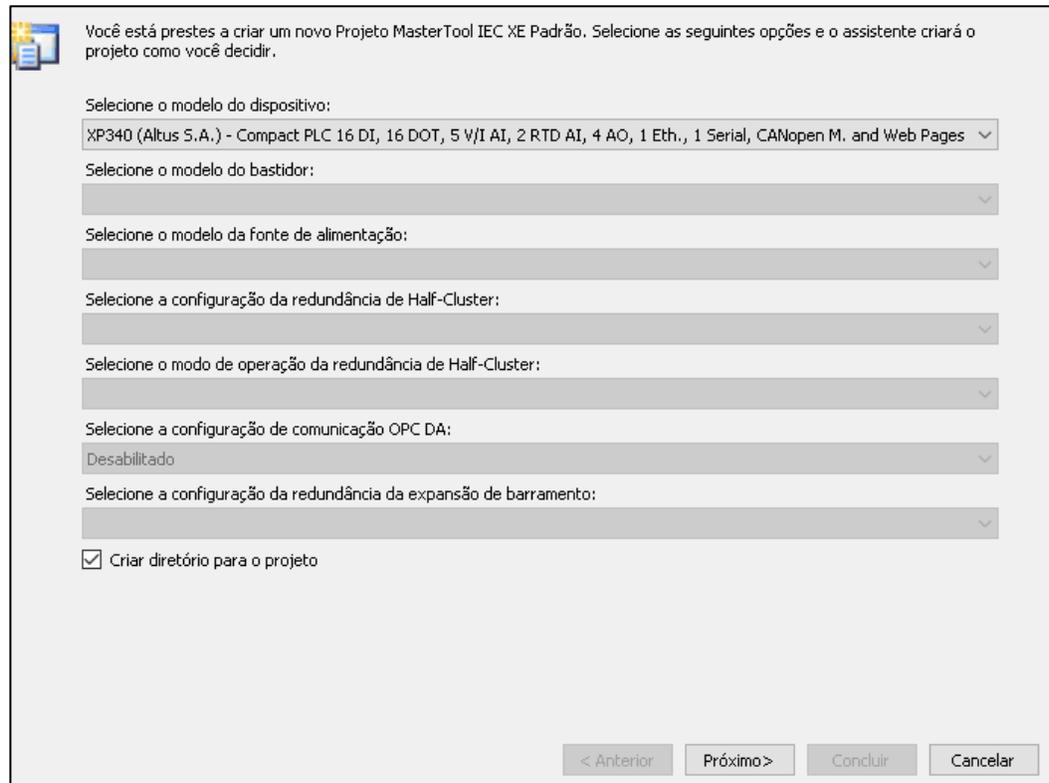


Figura 2-36. Classificação do Projeto

Escolhendo o modelo *Projeto MasterTool Padrão* será aberto um wizard para criação de projetos, onde o usuário deverá selecionar a UCP desejada. Nesse caso, será utilizada a UCP XP340.



Você está prestes a criar um novo Projeto MasterTool IEC XE Padrão. Selecione as seguintes opções e o assistente criará o projeto como você decidir.

Selecione o modelo do dispositivo:
XP340 (Altus S.A.) - Compact PLC 16 DI, 16 DOT, 5 V/I AI, 2 RTD AI, 4 AO, 1 Eth., 1 Serial, CANopen M. and Web Pages ▾

Selecione o modelo do bastidor:
▾

Selecione o modelo da fonte de alimentação:
▾

Selecione a configuração da redundância de Half-Cluster:
▾

Selecione o modo de operação da redundância de Half-Cluster:
▾

Selecione a configuração de comunicação OPC DA:
Desabilitado ▾

Selecione a configuração da redundância da expansão de barramento:
▾

Criar diretório para o projeto

< Anterior Próximo > Concluir Cancelar

Figura 2-37. Seleção dos Módulos de Hardware

A seguir, o usuário deverá selecionar o perfil para o projeto, e a linguagem padrão para as POU's (programas). Nesse caso, será utilizado o *perfil de Máquina* e linguagem ST. Clicar em *Próximo* para prosseguir. A opção de cancelamento também está disponível nessa janela via botão *Cancelar*.

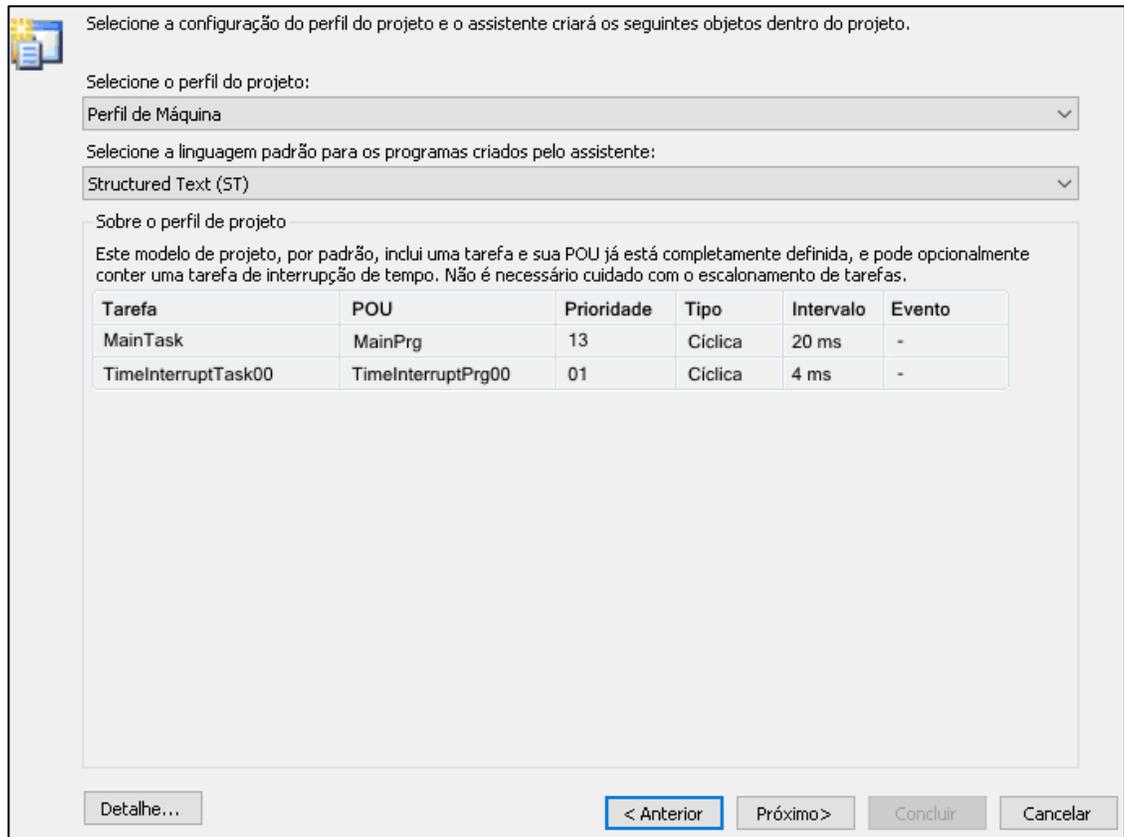


Figura 2-38. Seleção do Perfil

A próxima tela define a linguagem da POU criada pelo perfil selecionado. Como o perfil é *perfil de Máquina*, existe apenas a POU (UserPrg) e a POU (StartPrg), e a linguagem ST foi mantida. Clicar em *Próximo* para ir à tela de criação da POU de interrupção por tempo, selecionar *Não Criar*.

Ao pressionar o botão *Concluir*, o MasterTool IEC XE iniciará a criação do ambiente de desenvolvimento do projeto. Esse procedimento pode levar alguns segundos.

Criando POUs

Uma POU (Program Organization Unit, ou Unidade de Organização de Programa), é uma subdivisão do programa aplicativo que pode ser escrito em qualquer uma das linguagens disponíveis no software MasterTool IEC XE.

Com a criação do projeto através de um perfil selecionado, algumas POUs já são criadas, porém o usuário poderá criar quantas quiser, limitado pelo tamanho máximo da memória de programa.

Para inserir uma nova POU, basta clicar com o botão direito do mouse sobre Application (nome padrão criado para a aplicação), selecionar *Acréscitar Objeto* e, então, *POU...*, conforme mostra a figura a seguir.

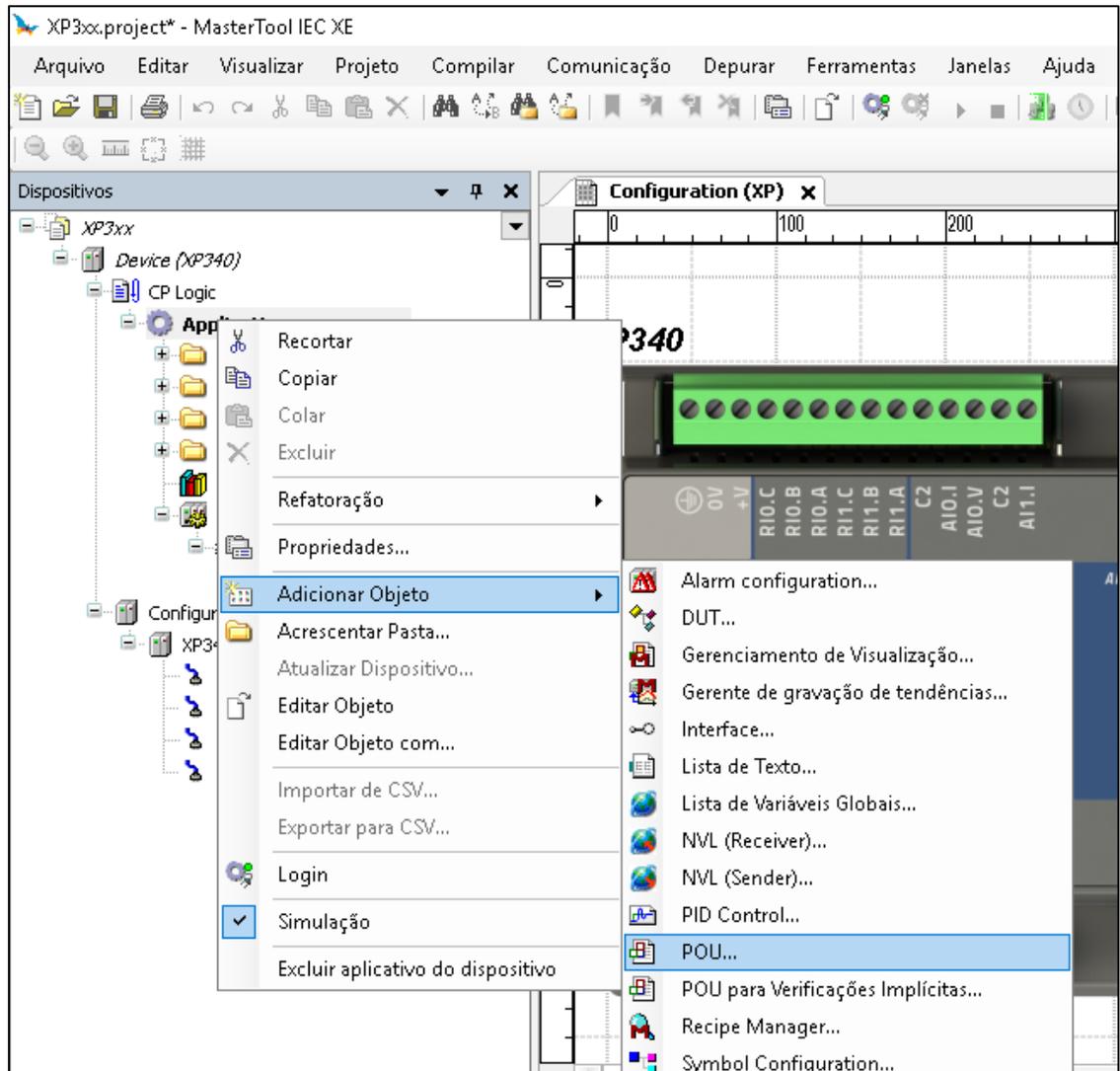


Figura 2-39. Inserindo POU's

Uma janela de configuração surgirá na tela, na qual o usuário deve colocar o nome da POU e selecionar o tipo e a linguagem que se deseja implementar. A seguir, deve clicar em *Abrir*.

Acrescentar POU

Criar uma nova POU (Unidade de Organização de Programa)

Nome
POU

Tipo

Programa

Bloco funcional

Estender

Implementar

Final Abstrato

Especificador de acesso

Linguagem de implementação do método
Diagrama de Lógica Ladder (LD)

Função

Tipo de retorno

Linguagem de implementação
Diagrama de Lógica Ladder (LD)

Adicionar Cancelar

Figura 2-40. Classificando a POU

Para editar a POU basta selecionar a aba com o nome correspondente e iniciar o desenvolvimento da aplicação na linguagem escolhida anteriormente. O mesmo procedimento é válido para as POUs criadas automaticamente pelo perfil do projeto.

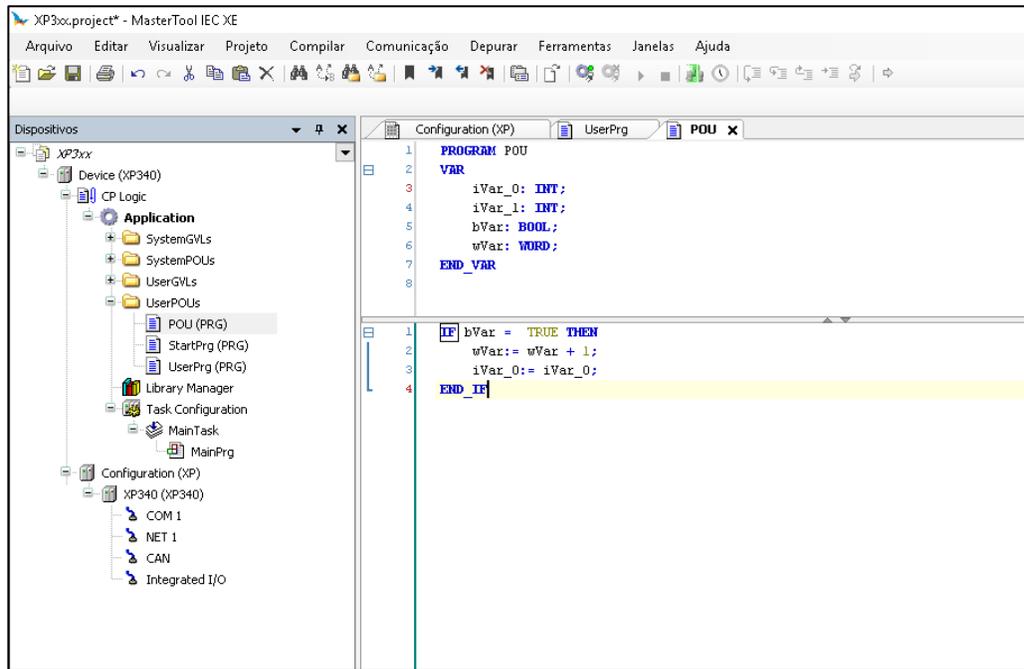


Figura 2-41. Editando a POU

Criando Tarefas

Para que uma POU seja executada, ela deve estar vinculada a alguma tarefa. Esse mecanismo de escalonamento, denominado de Tarefa, é muito útil para sistemas de tempo real que definem a execução periódica ou em resposta a um evento (mudança de estado de alguma variável booleana). As tarefas controlam a execução de programas em diferentes taxas, em função das características da aplicação. A necessidade de executar programas em taxas diferentes tem por objetivo as exigências de tempo de resposta do processo sob controle e otimizar o uso da capacidade de processamento da UCP. Os controladores que fazem o uso de tarefas são denominados sistemas multitarefa.

Somente será permitida a criação de novas tarefas quando o perfil de projeto selecionado for o Personalizado (Custom), sendo que nos demais perfis as tarefas possíveis são criadas e configuradas automaticamente.

Desta forma, para incluir uma nova tarefa (caso o perfil selecionado permita), basta clicar com o botão direito do mouse sobre o objeto *Task Configuration*, selecionar *Acrescentar Objeto* e então *Tarefa...*, conforme mostra a figura a seguir.

ATENÇÃO:

No caso da Série Nexto Xpress, o único perfil permitido é o "Perfil de máquina", para mais detalhes sobre as características desse perfil, consultar seção: *Funcionalidades do MasterTool IEC XE*

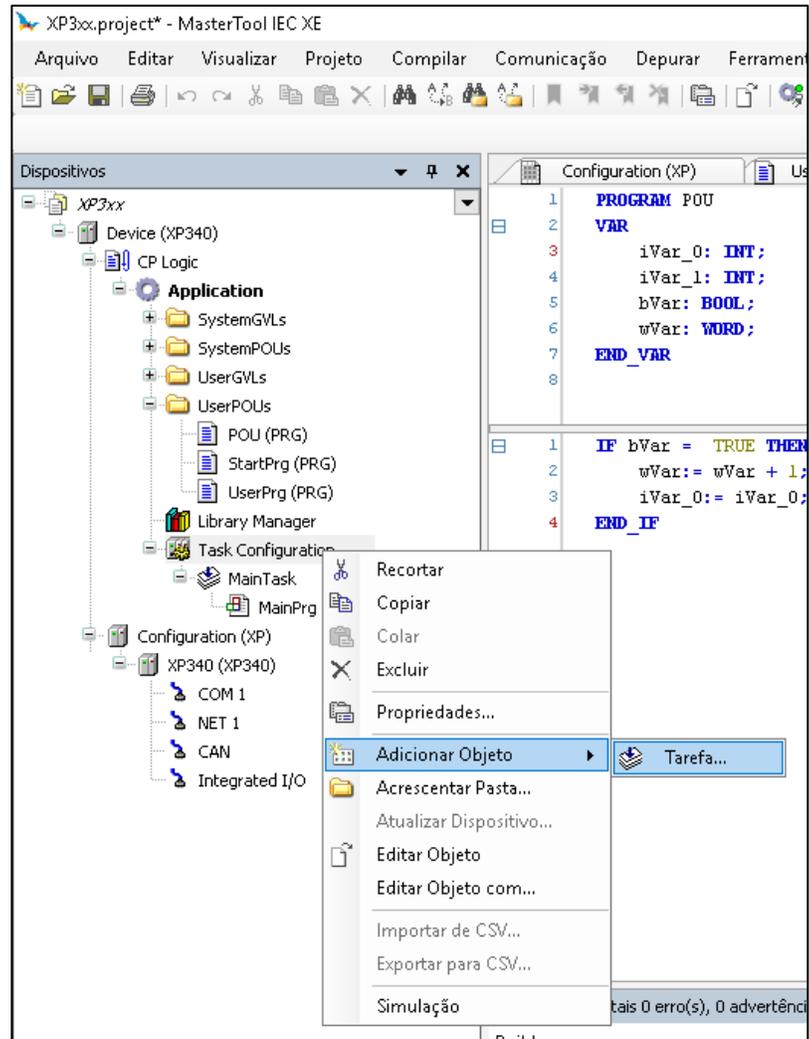


Figura 2-42. Criando uma Tarefa

Em seguida, surgirá uma tela para colocar o nome da tarefa. Após, clicar em *Abrir* para finalizar a criação dela.

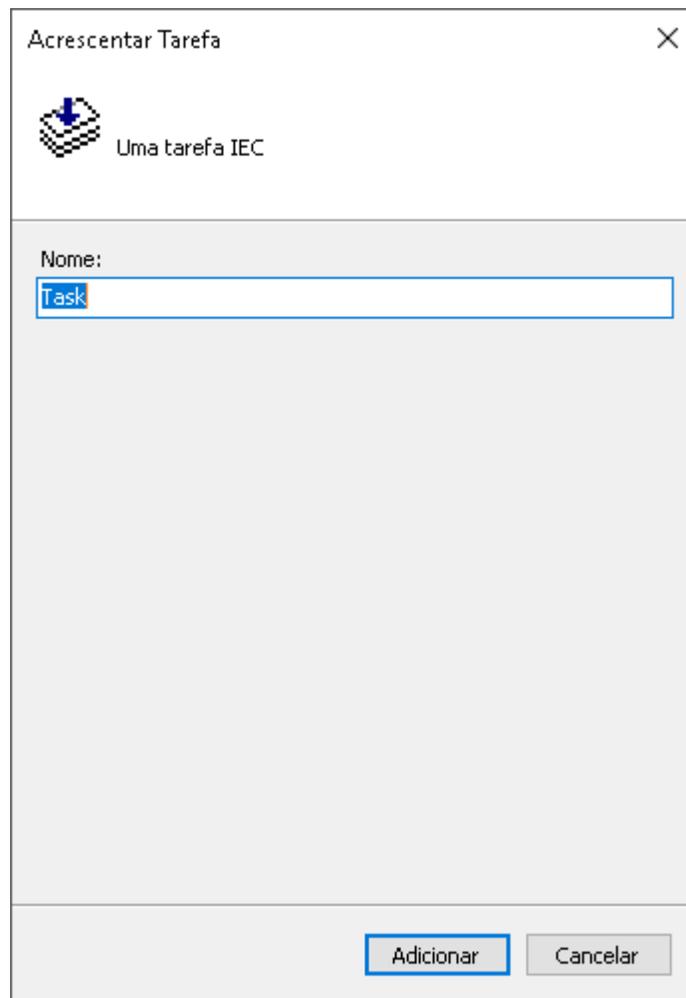


Figura 2-43. Nomeando a Tarefa

Configurando Tarefas

Após abrir a tarefa, a janela de configuração surgirá para que o usuário defina e classifique o funcionamento da mesma.

O campo *Prioridade* (0..31) estabelece a prioridade em que a tarefa será executada na aplicação, sendo o 0 (zero) mais prioritário. Por exemplo, a *MainTask*, criada na maioria dos perfis de projeto, tem prioridade 13, ou seja, essa tarefa é considerada prioritária para o sistema.

O campo *Tipo* define qual o tipo e o método para a tarefa ser executada, sendo que podem ser selecionados os seguintes itens:

- *Cíclica*: a tarefa é executada ciclicamente, ou seja, é chamada a cada intervalo de tempo configurado no campo ao lado. Ex: t#20ms.
- *Evento*: a tarefa é executada quando a variável do tipo BOOL, configurada no campo ao lado, recebe uma borda de subida, ou seja, a variável passe de FALSE para TRUE.
- *Externa*: a tarefa é executada quando uma interrupção externa ocorre, a qual é configurada no campo ao lado.
- *Contínua*: a tarefa sempre é executada, de acordo com a sua prioridade, ou seja, tarefas com maior prioridade são executadas primeiramente.

- *Status*: a tarefa é executada quando a variável do tipo BOOL, configurada no campo ao lado, for verdadeira.

Além dos campos mencionados acima, ainda deve ser configurado o Intervalo (apenas para tarefas cíclicas), ou seja, o intervalo de tempo em que a tarefa é chamada para executar. O tempo máximo para a MainTask nos perfis Simples, Básico, Normal, Experiente e Personalizado é igual a 750 ms e o tempo mínimo igual a 5 ms. No Perfil de Máquina o tempo máximo para a MainTask é igual a 100 ms e o tempo mínimo 5 ms. Recomenda-se configurar o intervalo da tarefa de no mínimo duas vezes o tempo de ciclo (execução) da mesma.

O cão-de-guarda da UCP é configurado para evitar o travamento das tarefas de usuário. O campo *Tempo*, define o tempo máximo permitido para a execução da tarefa. Caso a tarefa leve um tempo maior que o tempo de cão-de-guarda para ser executada, a aplicação irá para STOP e entrará em exceção por cão-de-guarda. O campo *Sensibilidade* refere-se a quantas vezes o tempo do cão-de-guarda deverá ser atingido para que o diagnóstico seja ativado. Caso o tempo de execução da tarefa atinja o valor do campo Sensibilidade multiplicado pelo campo Tempo, o diagnóstico também será indicado. Deve-se atentar ao fato de que o cão-de-guarda da UCP não é utilizado para proteger a aplicação do usuário de picos no tempo de execução e sim de travamentos. Portanto, seu tempo deve ser configurado com um valor alto, se comparado ao tempo de execução da tarefa a que está relacionado. O ideal é manter o tempo médio de execução das tarefas em, no máximo, 50% do tempo de cão-de-guarda. Assim, diminuem as chances de ocorrerem erros de cão-de-guarda por eventuais picos de tempo na execução da tarefa.

Visando proteger o sistema quanto a possíveis erros de configuração, o MasterTool IEC XE verifica em todas as tarefas cíclicas, durante a compilação, o cão-de-guarda (Cão-de-Guarda de Software) e os limites mínimo e máximo do intervalo da tarefa. É importante destacar que o usuário deverá ter cuidado ao alterar os valores pré-definidos pelos perfis de projeto, uma vez que alterações indevidas podem colocar em risco a execução do sistema. Então, recomenda-se utilizar os valores padrão.

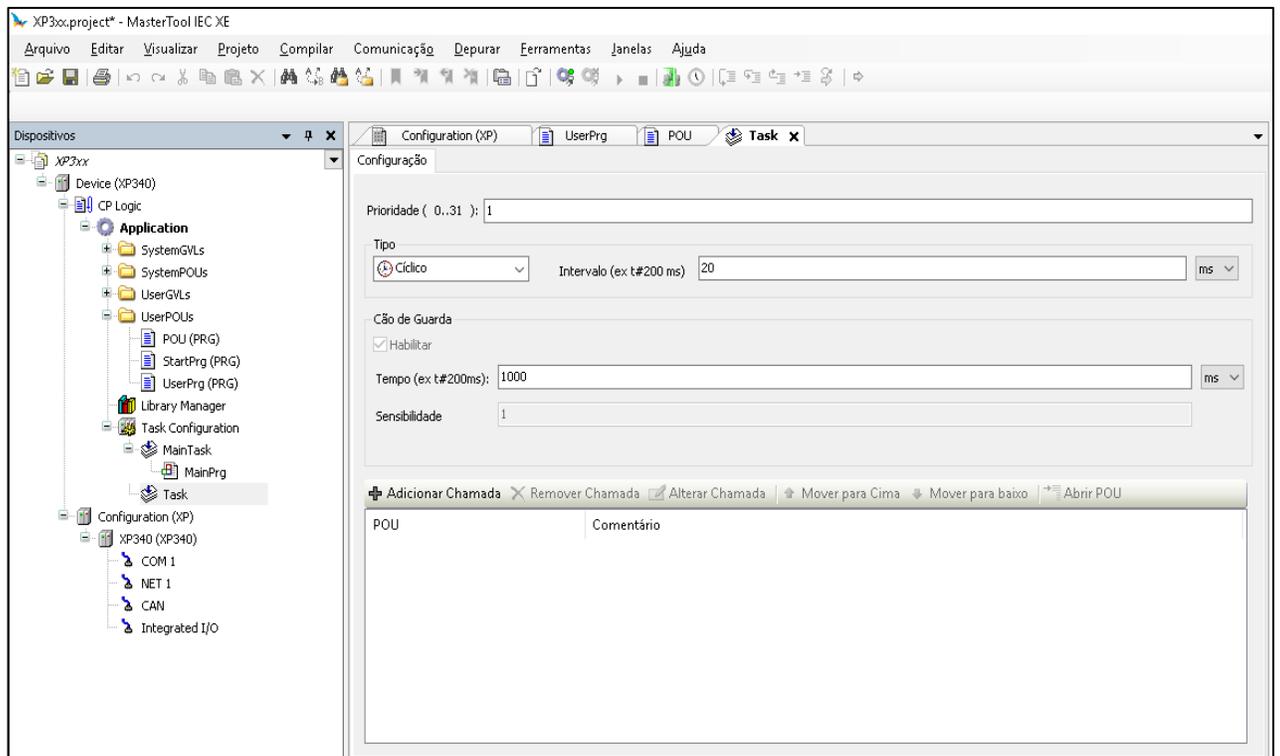


Figura 2-44. Configurando a Tarefa Criada

A tabela a seguir exibe as verificações realizadas pelo MasterTool IEC XE para a configuração do campo Intervalo das tarefas cíclicas, quando o campo Sensibilidade é igual a 1. Para o Perfil Personalizado não é feita consistência no intervalo da tarefa e no tempo de cão-de-guarda.

Tabela 2-2. Limites das tarefas

Tarefa	Tipo	Intervalo Mínimo	Intervalo Máximo
MainTask	Cíclica	5 ms	750 ms
CyclicTask	Cíclica	5 ms	2147483 ms
TimeInterruptTask00	Cíclica	500 us	2147483 ms

Vinculando uma POU a uma Tarefa

Conforme mencionado anteriormente, para que uma POU seja executada na aplicação, ela deve estar associada a uma tarefa. Nos perfis de projeto (sem considerar o Personalizado), as POUs já estão associadas as suas respectivas tarefas. Porém, caso seja utilizado o perfil Personalizado (Custom) ou caso novas POUs sejam criadas, as mesmas devem ser vinculadas a tarefas.

Para associar uma POU criada basta acessar a tarefa desejada clicando duas vezes sobre a mesma na árvore de dispositivos e, em seguida, clicar sobre *Acréscenatar*. Após, surgirá uma tela denominada *Assistente de Entrada*, na qual deverá ser selecionada a POU desejada, conforme mostra a figura a seguir.

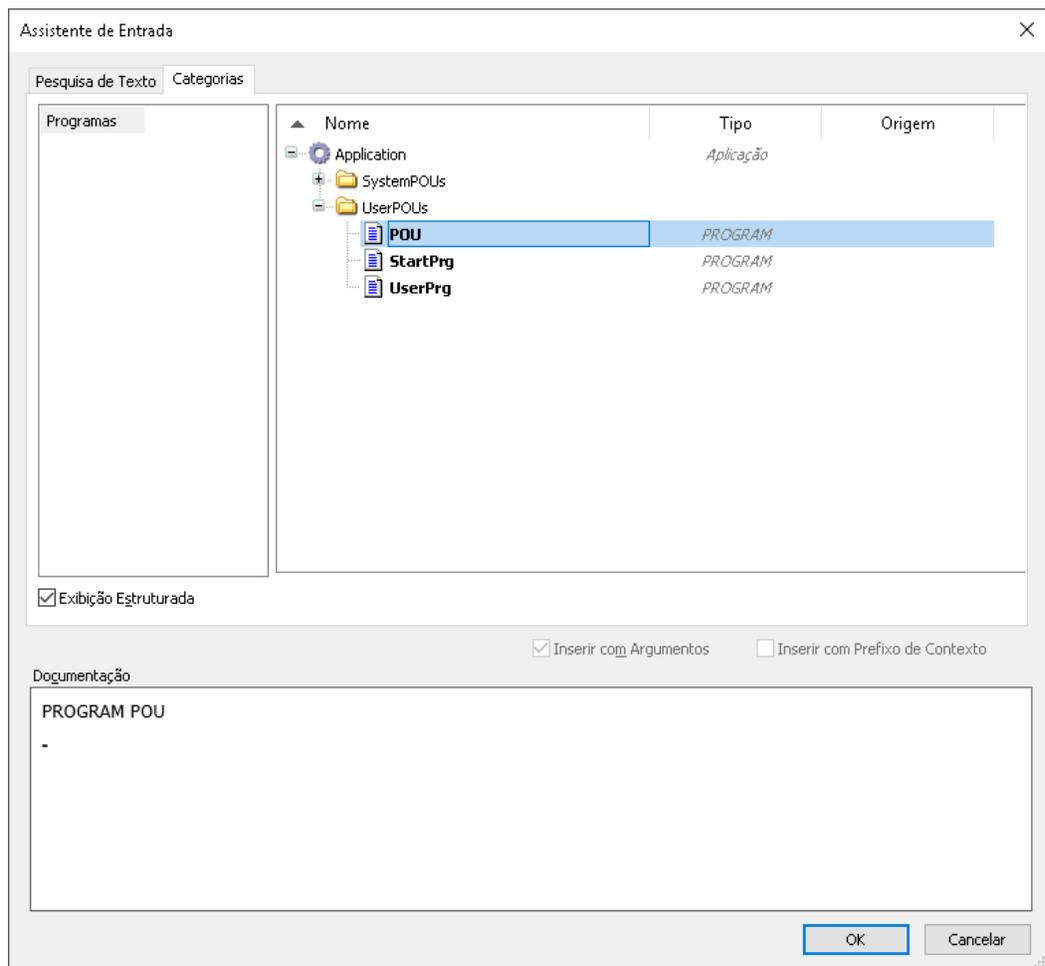


Figura 2-45. Vinculando POUs às Tarefas

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. Para informações adicionais sobre a quantidade máxima de tarefas IEC por UCP e perfil de projeto, consulte o Manual de Utilização UCPs Série Nexto (MU214100).

Configurando a UCP

A configuração da UCP Nexto Xpress baseia-se em estruturar a área de diagnósticos, a área de memória retentiva e persistente, entre outros parâmetros.

O usuário deverá dar dois cliques no ícone da UCP Nexto Xpress (localizado na árvore de dispositivos) e configurar os campos conforme descrito no Manual de Utilização UCPs Série Nexto – MU214100, capítulo Configuração.

Bibliotecas

Existem diversos recursos da ferramenta de programação disponibilizados através de bibliotecas. Sendo assim, os mesmos devem ser inseridos no projeto para que a sua utilização seja possível. O procedimento de inserção é bastante simples, sendo necessário selecionar o item *Library Manager*, disponível no menu da esquerda e selecionar *Acrescentar Biblioteca*, conforme mostra a figura a seguir.

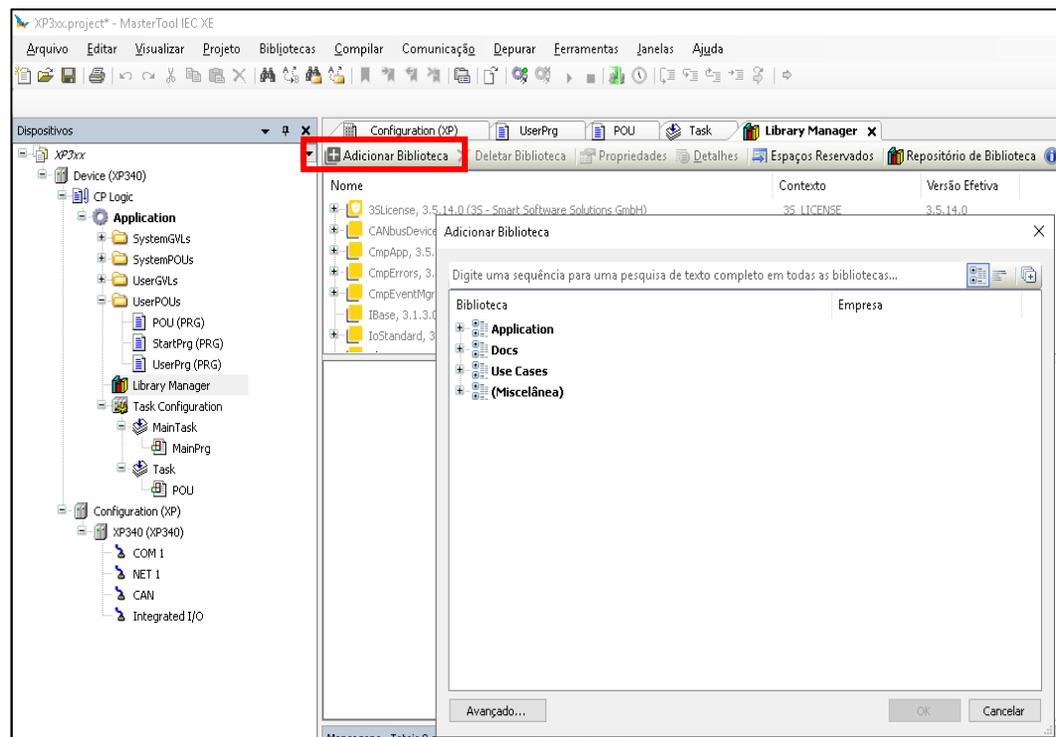


Figura 2-46. Inclusão de uma Biblioteca no Projeto

Na sequência, deve-se selecionar a biblioteca desejada para inclusão no projeto, pressionando em seguida, o botão OK. Dependendo das opções selecionadas nas opções de características esse diálogo pode ter mais ou menos opções.

Inserindo uma Instância de Protocolo

As UCPs da Série Nexto Xpress disponibilizam protocolos, como o MODBUS. Basta adicionar e configurar a instância do protocolo desejado na interface de comunicação.

Na sequência, são descritos dois casos de inserção do protocolo MODBUS, sendo um na interface serial e outro na interface Ethernet.

Inserção de uma instância do protocolo MODBUS RTU Escravo

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 do mouse sobre a COM e selecionar Acrescentar Dispositivo..., conforme mostra a figura a seguir.

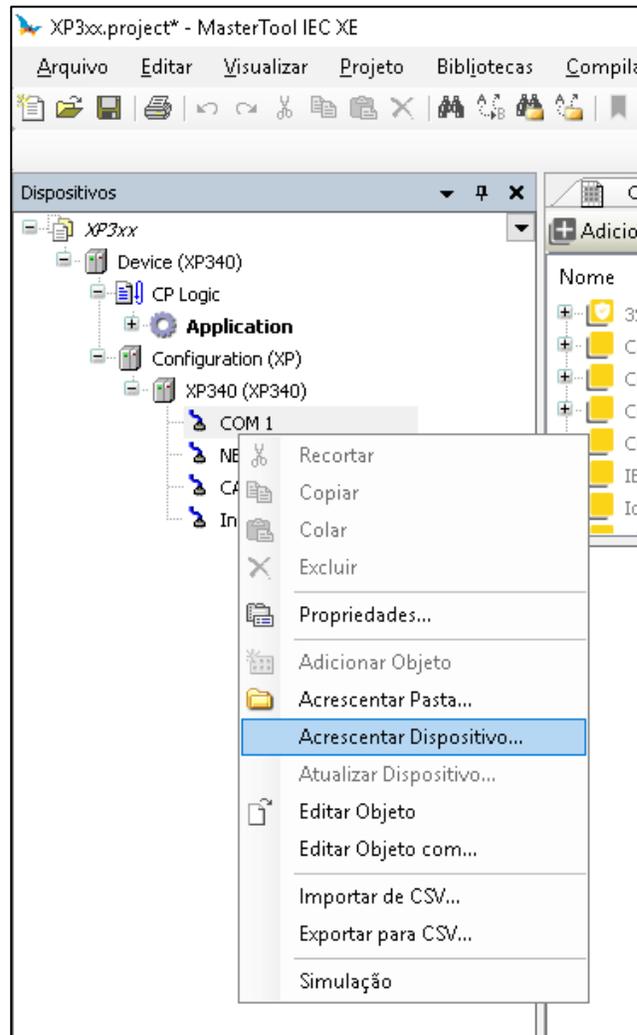


Figura 2-47. Adicionando a Instância

Após, surgirão na tela os protocolos disponíveis ao usuário. Nesse caso, selecionar o *MODBUS RTU Slave* e clicar em *Acrescentar*, conforme mostra a figura a seguir.

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.

Desta forma, como alternativa à configuração MODBUS RTU Slave por Representação Direta (%Q) mostrada anteriormente, pode-se selecionar a opção MODBUS Symbol RTU Slave, para configuração por Mapeamento Simbólico.

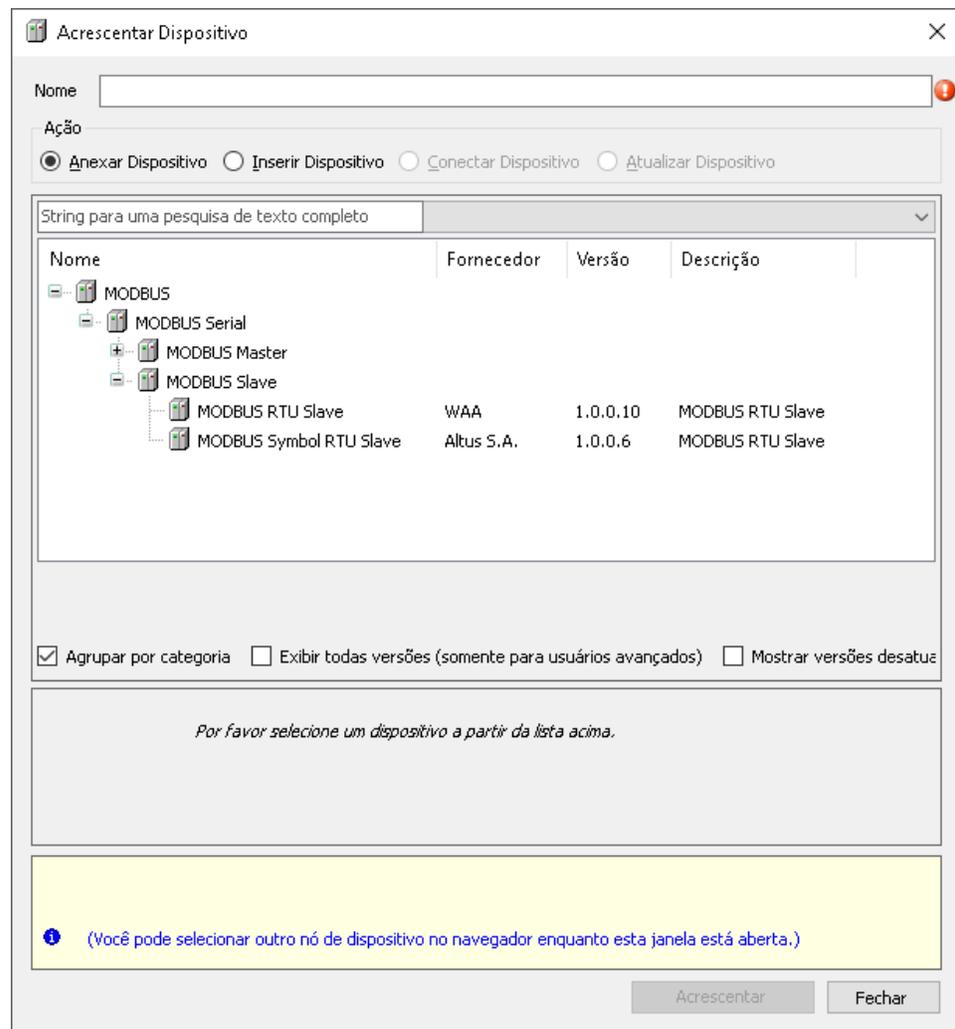


Figura 2-48. Seleção do protocolo MODBUS RTU Slave

Configuração do Protocolo MODBUS Mestre

O Nexto Xpress também permite a configuração do protocolo MODBUS Mestre (a figura anterior ilustra essa possibilidade) nas opções de mapeamento simbólico e por representação direta (%Q). Essa configuração é abordada em detalhes no Módulo de Treinamento CP Nexto Avançado ALTUS MP10A, ou mesmo na documentação do produto.

Inserção de uma instância do protocolo MODBUS Ethernet Cliente

O primeiro passo para configurar o MODBUS Ethernet, em modo cliente, é incluir a instância na NET desejada (nesse caso NET 1, pois a UCP NX3010 possui uma interface Ethernet). Clicar com o botão direito do mouse sobre a NET e selecionar *Acrescentar Dispositivo...*, conforme mostra a figura a seguir.

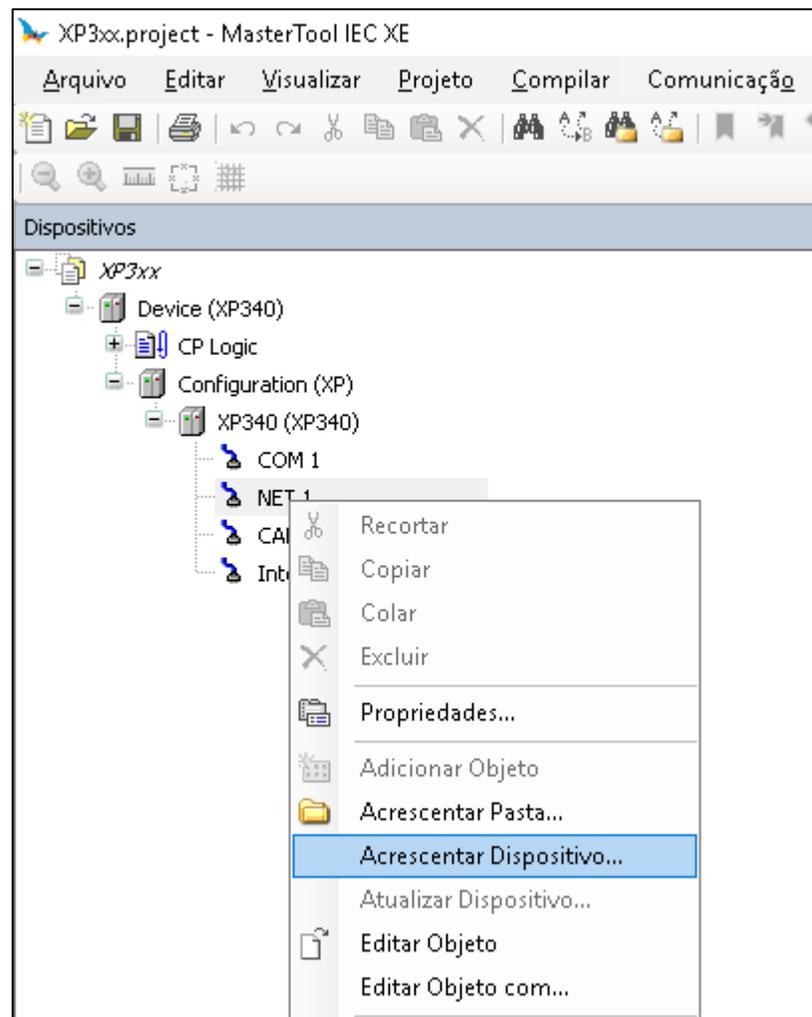


Figura 2-49. Adicionando a Instância

Após, surgirão na tela os protocolos disponíveis ao usuário. Nesse caso, selecionar o *MODBUS Client* e clicar em *Acrescentar*, conforme mostra a figura a seguir.

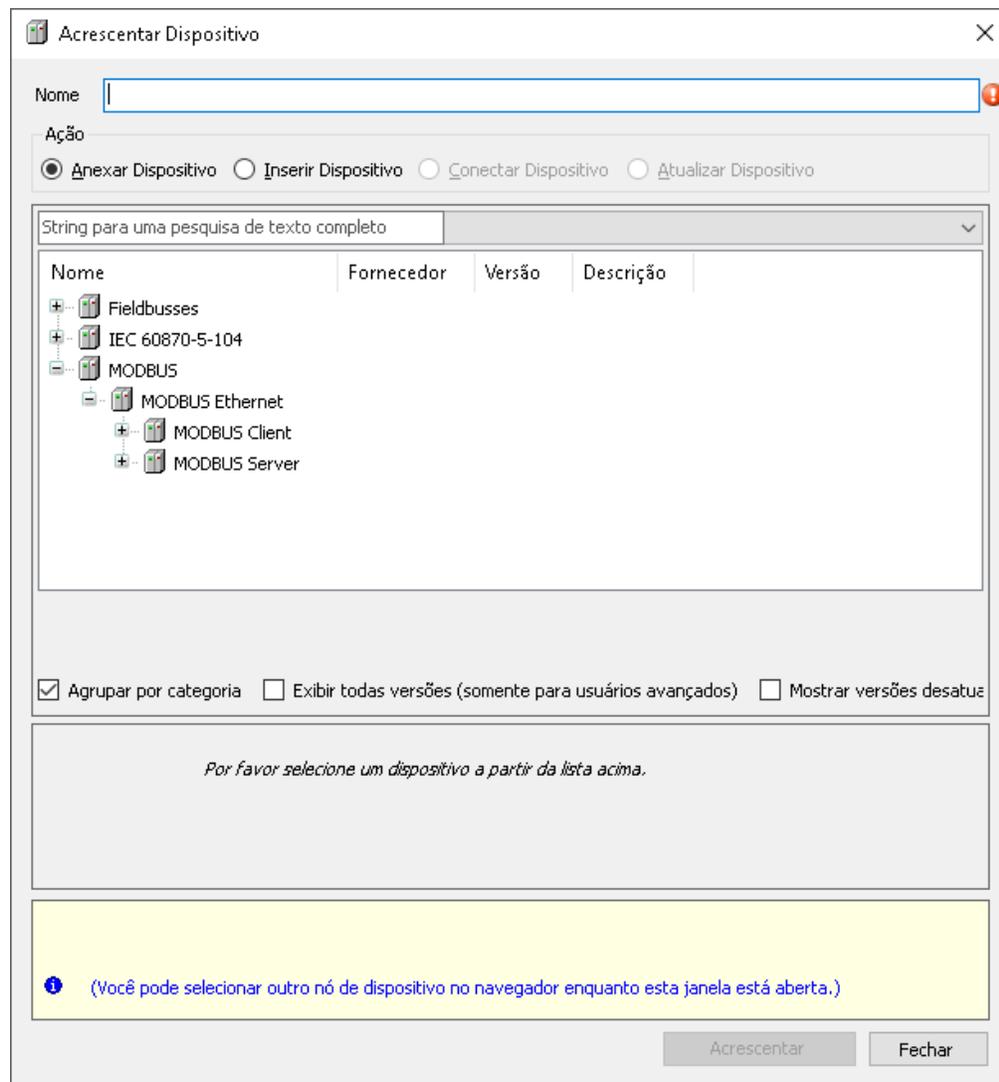


Figura 2-50. Selecionando o Protocolo

Como alternativa à configuração MODBUS Client por Representação Direta (%Q) mostrada anteriormente, pode-se selecionar a opção MODBUS Symbol Client, para configuração por Mapeamento Simbólico conforme ilustrado na figura a seguir.

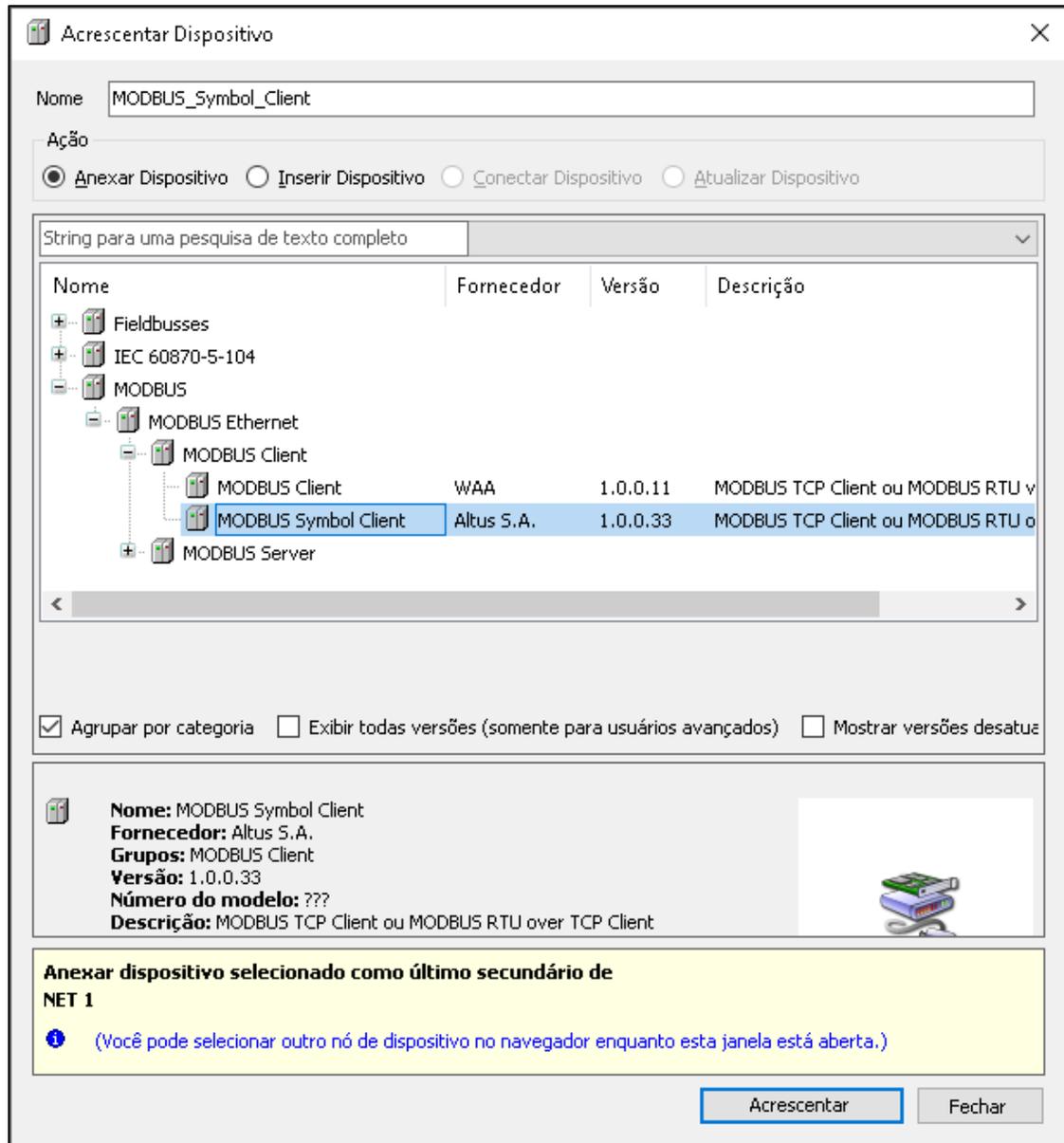


Figura 2-51. Selecionando o Protocolo MODBUS Ethernet

Configuração do Protocolo MODBUS Ethernet Servidor

O Nexto Xpress também permite a configuração do protocolo MODBUS Ethernet Servidor (a figura anterior ilustra essa possibilidade) nas opções de mapeamento simbólico e por representação direta (%Q). Essa configuração é abordada em detalhes no Módulo de Treinamento CP Nexto Avançado ALTUS MP10A, ou mesmo na documentação do produto.

Protocolo OPC DA na Série Nexto Xpress

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.

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

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.

O protocolo OPC DA na Série Nexto é abordado em detalhes no Módulo de Treinamento CP Nexto Avançado ALTUS MP10A, ou mesmo na documentação do produto.

Protocolo OPC UA na Série Nexto Xpress

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.

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`.

O protocolo OPC UA na Série Nexto é abordado em detalhes no Módulo de Treinamento CP Nexto Avançado ALTUS MP10A, ou mesmo na documentação do produto.

Compilando um Projeto

A fim de realizar a verificação da aplicação criada, o usuário deve executar a compilação do projeto. Essa é a forma mais eficaz de se encontrar problemas ou receber avisos sobre alguns equívocos cometidos durante a configuração do produto e edição da aplicação. Para executar tal procedimento, basta acessar o menu *Compilar* e clicar na opção *Gerar Código*.

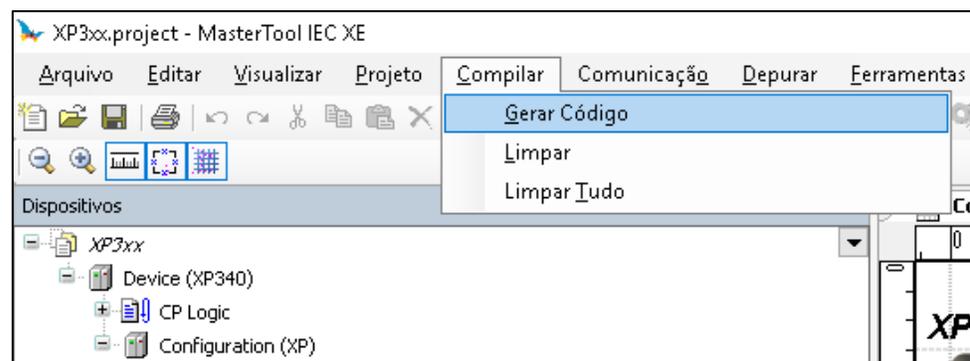


Figura 2-52. Compilando o Projeto

Depois do tempo de processamento, o qual vai variar de acordo com o tamanho da aplicação do usuário, os erros e mensagens de alerta, caso sejam necessários, serão mostrados conforme mostra a figura a seguir.

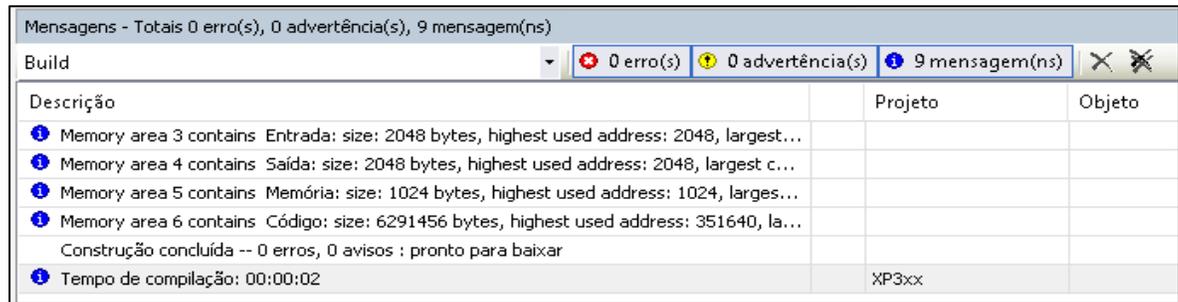


Figura 2-53. Mensagens da Compilação

Caso os erros e mensagens não estejam visíveis na tela, basta acessar o menu *Visualizar* e clicar em *Mensagens*, conforme mostrado na figura a seguir.

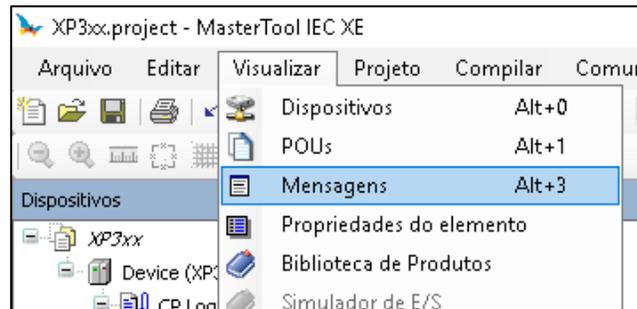


Figura 2-54. Incluindo as Mensagens na Tela

Simulando um Projeto

O MasterTool IEC XE possui um importante recurso de simulação que permite ao usuário testar sua aplicação sem a utilização do equipamento, conferindo maior agilidade no desenvolvimento do programa. No entanto, alguns recursos específicos, que dependem do hardware das UCPs Nexto, não são possíveis de serem simulados.

Seguem abaixo, os recursos indisponíveis no modo de simulação:

- Relógio RTC;
- Varredura do Barramento;
- Módulos de E/S;
- Interrupção de Barramento;
- Portas Seriais;
- Comunicação Ethernet;
- Protocolos de comunicação como Modbus;
- Interfaces PROFIBUS;
- Escravos PROFIBUS;
- Operações em cartão de memória;
- Diagnósticos em variáveis;
- Diagnostics Explorer;
- Outras funções que acessem o hardware do CP.

Por esta razão o modo simulação deve ser utilizado para testar a lógica da aplicação no que não depender de funções de acesso ao hardware. Estes recursos devem ser testados com o hardware para garantir o funcionamento da aplicação neste sentido.

Para alterar o MasterTool IEC XE para Modo Simulação é necessário selecionar esta opção no Menu Comunicação conforme figura a seguir. Após isso é exibido um aviso na barra inferior do MasterTool IEC XE que indica que a ferramenta está operando em Modo Simulação.

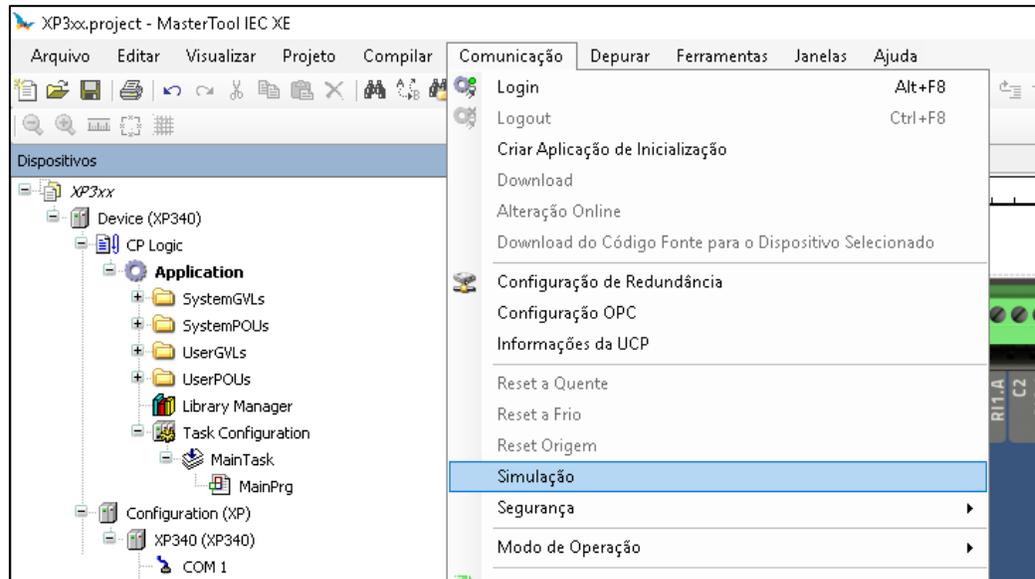


Figura 2-55. Modo Simulação

Em Modo Simulação a aplicação é executada em um dispositivo virtual no computador onde está instalado o MasterTool IEC XE. Por esta razão algumas características apresentadas estão relacionadas à arquitetura de hardware do computador e não das UCPs da Série Nexto Xpress. A principal característica neste sentido está relacionada ao formato dos dados nas áreas de memória de representação direta. O Modo Simulação trabalha com o formato little endian onde o primeiro endereço de memória é o menos significativo do dado. Por outro lado, as UCPs da Série Nexto trabalham com o formato big endian onde o primeiro endereço de memória é o mais significativo do dado.

Neste caso um mesmo dado escrito por exemplo em %QD0, será escrito de forma diferente na simulação e nas UCPs da Série Nexto. Se o dado escrito for 16#1234ABDC a distribuição dos dados na memória do CP ficará da seguinte maneira:

```
%QW0 = 16#1234
%QW2 = 16#ABCD
%QB0 = 16#12
%QB1 = 16#34
%QB2 = 16#AB
%QB3 = 16#CD
```

Para o mesmo dado escrito em %QD0 no Modo Simulação a distribuição dos dados na memória ficarão da seguinte maneira:

```
%QW0 = 16#ABCD
%QW2 = 16#1234
%QB0 = 16#CD
%QB1 = 16#AB
%QB2 = 16#34
%QB3 = 16#12
```

Tendo em vista estas diferenças e para facilitar o desenvolvimento da aplicação, utilizando os recursos do MasterTool IEC XE e das UCPs da Série Nexto é recomendado o uso de variáveis simbólicas. Neste caso as diferenças entre o Modo Simulação e o comportamento com as UCPs

da Série Nexto não são verificadas. Portanto a melhor prática é evitar o uso de variáveis de representação direta sempre que possível para evitar o retrabalho quando desenvolver uma lógica que será testada em simulação e depois carregada em uma UCP.

O modo Simulação pode ser utilizado para simular um projeto redundante, porém, terá as mesmas limitações citadas anteriormente, podendo ser testada apenas a lógica que não dependa do hardware. Neste caso, sempre serão executadas as POU's NonSkippedPrg e ActivePrg, como se o CP simulado fosse o CP Ativo.

Também é importante salientar que, em função das diferenças entre as arquiteturas dos dispositivos, o mesmo código gerado utilizando o dispositivo de simulação pode ter tamanhos nas áreas de dados e código diferentes daqueles gerados para uma UCP da Série Nexto.

Criação e Execução de Projetos



Na descrição a seguir será mostrado como criar um projeto simples contendo um programa aplicativo, como carregá-lo para a UCP através de um Gateway Server e também como executá-lo e monitorá-lo. O sistema de tempo de execução do Nexto Xpress usado para este projeto exemplo é fornecido com as configurações do MasterTool IEC XE.

O programa exemplo será escrito em linguagem de Texto Estruturado (ST) e consiste em um programa (UserPrg) e um bloco funcional (FB1). UserPrg conterá uma variável de contador (ivar) e uma chamada de bloco funcional (FB1). FB1 obterá a entrada “in” a partir do UserPrg, adicionará “2” a esta entrada e escreverá o resultado na saída “out”. Out será lida pelo UserPrg.

Em linhas gerais, a sequência abaixo sumariza o procedimento de criação e execução do projeto.

1. Iniciar o MasterTool IEC XE;
2. Criar um projeto
3. Declarar variáveis na UserPrg;
4. Digitar o código no corpo da UserPrg;
5. Criar a POU de programação adicional (ST bloco funcional FB1);
6. Definir os recursos para executar e controlar o programa no Nexto;
7. Definir a aplicação ativa;
8. Configurar o canal de comunicação para o Nexto;
9. Compilar e carregar a aplicação no Nexto;
10. Iniciar a aplicação;
11. Monitorar a aplicação;
12. Depurar uma aplicação;
13. Configurar um breakpoint e realizar a varredura do programa.

Declarar Variáveis na UserPrg

A POU “UserPrg” disponível na janela *Dispositivos* do editor de linguagem ST (parte central da interface do usuário MasterTool IEC XE) é automaticamente aberta com um duplo clique em seu item correspondente na árvore dos Dispositivos.

As POUs, em geral, sempre podem ser abertas na visualização do editor desta forma (clique duplo no item correspondente da árvore).

O editor ST consiste em uma parte de declaração (parte superior) e um “corpo” (parte inferior), separados por um divisor de tela móvel.

A parte da declaração mostra números de linha no canto esquerdo, o tipo e nome das POUs (PROGRAM UserPrg) e as palavras-chave “VAR” e “END_VAR” para a declaração das variáveis.

O corpo é vazio, somente a linha número 1 é exibida na figura a seguir.

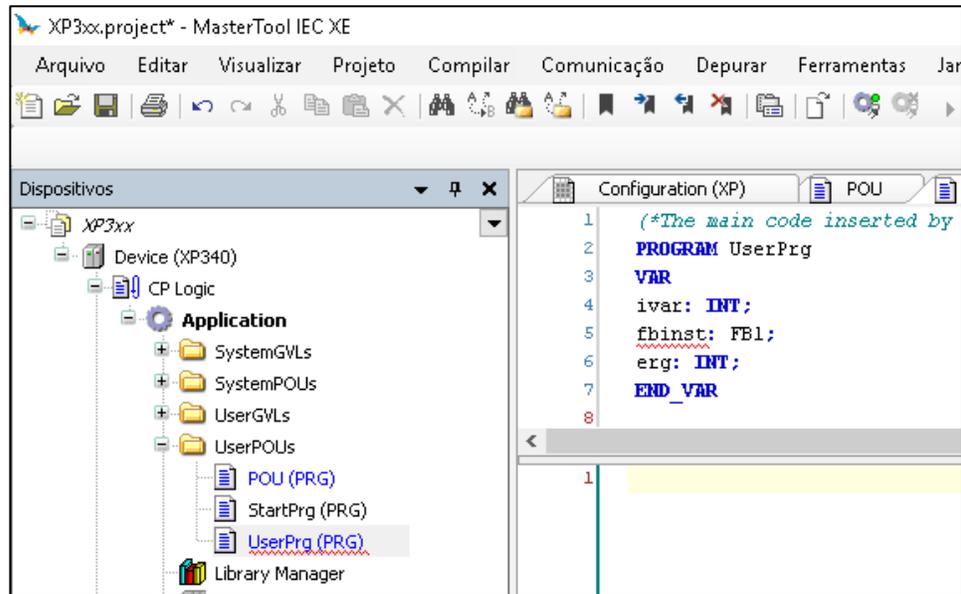


Figura 2-56. Janela do Editor ST

Na parte de declaração do editor posicione o cursor após VAR e pressione a tecla <ENTER>. Será inserida uma linha vazia para entrar com a declaração das variáveis “ivar” e “erg” do tipo INT e fbinst do tipo FB1:

```
PROGRAM UserPrg
VAR
  iver: INT;
  fbinst: FB1;
  erg: INT;
  END_VAR
```

Outra opção é digitar diretamente uma instrução na parte de implementação do editor (corpo) e utilizar a função Autodeclarar.

Digitar o Código no Corpo da UserPrg

```
  iver := iver+1; // contador
  fbinst(in:=11, out=>erg); // chama o bloco funcional do tipo FB1,
  // com o parâmetro de entrada "in"
  // a saída é escrita em "erg"
```

Outra possibilidade é usar a configuração da Autodeclaração. Sem nenhuma declaração anterior, entre com uma instrução no corpo do programa e pressione a tecla <ENTER>. Para cada variável não declarada encontrada na linha de implementação, o diálogo *Autodeclarar* se abre para realizar as configurações da declaração.

Figura 2-57. Diálogo Autodeclarar

O escopo e nome das variáveis, assim como a POU atual (objeto) serão preenchidos automaticamente. Digite o tipo desejado e o valor de inicialização de acordo com a declaração descrita anteriormente e adicione um comentário. Adicionalmente poderia ser inserido o endereço de uma variável de representação direta no campo *Endereço*.

Confirme o diálogo com *OK*. Isto fará com que a declaração de “erg” seja inserida na parte de declaração da POU com os comentários, conforme a figura a seguir.

```

(*The main code inserted *)
PROGRAM UserPrg
VAR
ivar: INT;
fbinst: FB1;
// result
    erg: INT;
END_VAR

```

Figura 2-58. Declaração da Variável “erg”

Criar a POU de Programação Adicional (ST Bloco Funcional FB1)

O bloco funcional FB1 executa a seguinte expressão: $out = in + 2$.

Escolha o comando *Acréscitar Objeto* no menu *Projeto*.

Selecione *POU* na parte esquerda do diálogo *Acréscitar Objeto*. Insira o nome “FB1” para a POU e ative a opção *Bloco funcional* na seção *Tipo*.

Escolha *Texto Estruturado* para a linguagem de implementação.

Pressione o botão *Abrir* para confirmar as configurações do objeto.

Outra janela do editor será aberta para o novo bloco funcional FB1. Da mesma forma que em *UserPrg*, declare as seguintes variáveis:

```

FUNCTION_BLOCK FB1
VAR_INPUT
    in: INT;
END_VAR
VAR_OUTPUT
    out: INT;
END_VAR

```

```
VAR
  ivar:INT:=2;
END_VAR
```

Na parte de implementação do editor, digite:

```
out:=in+ivar;
```

Definir os Recursos para Executar e Controlar o Programa no Nexto Xpress

Iniciar Servidor de Gateway

O servidor de gateway é iniciado automaticamente na inicialização do sistema como um serviço. O ícone (🏠) na barra do sistema, indica que o servidor está rodando. O ícone (✖) indica que o gateway está interrompido. Este ícone é parte do programa GatewaySysTray, disponível para controle e monitoramento do serviço de Gateway. Ele fornece um menu com comandos *Start Gateway* e *Stop Gateway*, permitindo ao usuário parar e reiniciar o serviço manualmente. O menu também inclui o comando *Exit Gateway SysTray* do controle de gateway, o qual encerra o programa GatewaySysTray (mas não o serviço de Gateway). O programa GatewaySysTray é iniciado automaticamente quando o Windows é iniciado, entretanto também pode ser inicializado manualmente através do menu *Programas*.

Iniciar o Nexto Xpress (Simulação)

O dispositivo de simulação está disponível na inicialização do sistema. Na barra do sistema, os ícones (🛑) e (🏠) representam os status “parado” e “em execução” respectivamente. Se o sistema permitir, este dispositivo será iniciado automaticamente na inicialização do sistema. Caso contrário ele deverá ser iniciado manualmente via comando *Iniciar CP* (duplo clique no ícone). Este ícone é parte integrante do MasterTool IEC XE. Ele fornece um menu com os mesmos comandos *Iniciar* e *Parar* do Servidor de Gateway. O comando *Sair* do controle do CP encerra o programa GatewaySysTray (mas não o serviço de Gateway).

Configurar um Canal de Comunicação com o Nexto Xpress

A conexão resultante aparecerá na linha seguinte à opção “Selecionar caminho na rede para o controlador Nexto”.

Execute um clique duplo no *Device* na janela *Dispositivos*. O diálogo *Device* será aberto com o subdiálogo *Configurações de Comunicação*. Configure a conexão entre o Nexto e o programador conforme indicado anteriormente no item *Localizando a Rede*.

Executar e Monitorar a Aplicação no Nexto Xpress

Compilar e Carregar a Aplicação no Nexto Xpress

Para verificar erros sintáticos na aplicação ativa, use o comando *Gerar Código* (menu *Compilar*).

NOTA: não será gerado nenhum código neste caso. Informações, avisos e mensagens de erro serão exibidos na janela de Mensagens localizada na parte inferior da interface do usuário (padrão).

Mesmo que esta verificação sintática não tenha sido feita antes, é possível realizar o login no Nexto Xpress. Certifique-se de que o Nexto Xpress está em execução (símbolo colorido na barra do sistema).

Use o comando *Login* (menu *Comunicação*). Se as configurações de comunicação foram configuradas corretamente a seguinte mensagem aparecerá (caso contrário será necessário corrigi-las):

“Não existe aplicação no Dispositivo. Você deseja criá-la e continuar com o envio?”

Confirme com *Sim* para iniciar a compilação e o envio da aplicação.

As mensagens de compilação serão exibidas na janela de *Mensagens*. Se o projeto foi criado corretamente, não deve ocorrer erro e a aplicação pode ser iniciada.

Após a criação do “projeto padrão” e do envio da aplicação conforme descrito nos passos anteriores, a mesma é iniciada no dispositivo.

Iniciar a Aplicação no Nexto Xpress

Execute o comando *Iniciar* no menu *Depurar*. O programa iniciará a execução e a sinalização RUN (em verde) será exibida na barra de status na parte inferior da interface do usuário.

Monitorar a Aplicação

Existem três possibilidades de monitoração das variáveis do programa aplicativo:

- Listas de monitoração
- Escrita e forçamento de variáveis
- Visualização online de POUs

Abrir uma Janela de Instância do Programa

A visualização de uma POU fornece todas as expressões de monitoração daquela instância em uma tabela na parte de declaração e - se ativada como monitoração em linha - também na parte de implementação.

Para abrir a visualização online dê um clique duplo na POU *MainPrg* na janela *Dispositivos* ou selecione o item e execute o comando *Editar Objeto* (menu de contexto).

Na parte inferior da visualização aparecerão as linhas de código conforme digitadas no modo off-line, acrescidas de pequenas janelas de monitoração em linha após cada variável, mostrando o valor atual. Na parte superior, uma tabela mostrará as expressões de monitoração da POU, que consistem nos valores atuais das variáveis da aplicação no Nexto Xpress.

Device.Application.UserPrg			
Expressão	Tipo	Valor	Valor preparado
ivar	INT	253	
fbinst	FB1		
erg	INT	13	


```

1  ivar 253 := ivar 253 +1; // contador
2  fbinst(in 11 :=11, out 13 =>erg 13); // chama o bloco funcional do tipo FB1,
3  // com o parâmetro de entrada "in"
4  // a saída é escrita em "erg"
5  RETURN

```

Figura 2-59. Expressões de Monitoração e Linhas de Código

Escrever e Forçar Variáveis

É possível escrever ou forçar um Valor Preparado em uma variável no Nexto Xpress, ou seja, “ivar” assumirá este valor no início do próximo ciclo. Dê um duplo clique na coluna Valor Preparado, digite o valor inteiro desejado e saia do campo com <Enter> ou clicando com o mouse fora deste campo. Execute o comando Escrever Valores ou Forçar Valores (menu Depurar) para escrever ou forçar este valor no Nexto. O resultado aparecerá na coluna Valor.

Usar as Visualizações de Monitoração

Janelas de expressões de monitoração podem ser usadas para configurar tabelas específicas de monitoração da aplicação para, por exemplo, a depuração do programa aplicativo.

A partir do menu *Visualizar* use o comando *Janelas de Monitoração, Monitorar 1* para abrir a janela de monitoração.

Na coluna *Expressão* clique na primeira linha da tabela para abrir o campo de edição. Digite o caminho completo para a variável a ser monitorada (“ivar”): “Device.Application.UserPrg.ivar”.

É recomendado usar o assistente de entrada via botão específico para tal. Feche o campo de edição com <ENTER>. O tipo da variável será incluído automaticamente.

Faça o mesmo para as demais variáveis. A lista de monitoração mostrada na próxima figura contém somente expressões da UserPrg, mas, obviamente, é possível listar várias variáveis do projeto. Note que para instâncias, por exemplo do bloco funcional FB1, basta digitar “Device.Application.UserPrg.fbinst”. As variáveis específicas destas aparecerão automaticamente e as linhas correspondentes podem ser abertas via símbolo “+”. O valor atual da variável é mostrado na coluna *Valor*.

Monitoração 1			
Expressão	Aplicação	Tipo	Valor
UserPrg.erg	Device.Application	INT	13
+ UserPrg.fbinst	Device.Application	FB1	
UserPrg.ivar	Device.Application	INT	4437

Figura 2-60. Lista de Monitoração

Se ainda não foi feito, selecione o objeto da aplicação e execute o comando *Iniciar* no menu de contexto. A aplicação será iniciada no Nexto Xpress e o valor atual será exibido na coluna *Valor*.

Monitoração 1			
Expressão	Aplicação	Tipo	Valor
UserPrg.erg	Device.Application	INT	13
- UserPrg.fbinst	Device.Application	FB1	
in		INT	11
out		INT	13
ivar		INT	2
UserPrg.ivar	Device.Application	INT	5931

Figura 2-61. Exibição do Valor Atual

Também é possível realizar a escrita e o forçamento de valores.

Para desconectar-se do Nexto Xpress, execute o comando *Logout* no menu *Comunicação*.



Depurar uma Aplicação via Breakpoint e Varredura do Programa

No modo online, os breakpoints podem ser configurados como posições de parada na execução do programa.

Quando o programa alcançar um breakpoint, o programa poderá ser executado em etapas. Em cada posição de parada, poderá ser visto o valor atual das variáveis nas visualizações de monitoração.

Selecione a linha 1 de UserPrg. Pressione a tecla <F9> (idem ao comando *Alternar Breakpoint*, menu *Depurar*). O breakpoint será indicado.

```

1  ● ivar 1277 := ivar 1277 + 1; // CONTADOR
2  fbinst(IN 11 :=11, OUT 13 =>ERG 13); // CHAMA O BLOCO FUNCIONAL DO TIPO FB1,
3  // COM O PARÂMETRO DE ENTRADA "IN"
4  // A SAÍDA É ESCRITA EM "ERG"RETURN

```

Figura 2-62. Aplicação em "Stop"

Uma aplicação em execução parará em um breakpoint:

```

1  ● ivar 1277 := ivar 1277 + 1; // CONTADOR
2  fbinst(IN 11 :=11, OUT 13 =>ERG 13); // CHAMA O BLOCO FUNCIONAL DO TIPO FB1,
3  // COM O PARÂMETRO DE ENTRADA "IN"
4  // A SAÍDA É ESCRITA EM "ERG"RETURN

```

Figura 2-63. Aplicação no Modo "Run"

A tecla <F8> permite percorrer os passos (equivalente ao comando *Passar Dentro* no menu *Depurar*) e, portanto, percorrerá também instâncias de blocos funcionais.

Para pular as etapas do bloco funcional, use <F10> (o mesmo que o comando *Passar Sobre*). Todo os valores da variável do Nexto Xpress serão exibidos.

O diálogo breakpoints também está disponível via comando *Breakpoints*, no menu *Visualização*. Aqui, os breakpoints configurados no momento podem ser visualizados e editados, assim como novos breakpoints também podem ser inseridos.

Observe também que as posições de breakpoint serão memorizadas ao fazer o logout. Elas serão indicadas por marcadores vermelho-claro.

Estudo Dirigido 2-1: criação de projeto no MasterTool IEC XE

Seguindo os passos indicados anteriormente crie um projeto no MasterTool IEC XE utilizando o Perfil de Máquina salvando-o com o nome "MP11_[ALUNO].project" e a linguagem Ladder como padrão. Este projeto será utilizado mais adiante em outros estudos dirigidos.



DICA: caso você queira saber mais sobre o desenvolvimento de um projeto no Nexto Xpress, assista o Webinar a seguir (<https://www.youtube.com/watch?v=uCk4v540CSQ>). Nele é demonstrado o desenvolvimento de uma aplicação simples no software MasterTool IEC XE, onde uma entrada digital aciona uma saída digital e o valor de uma entrada analógica é transferido para uma saída analógica. Como não há nenhuma chave/sensor conectado às entradas (digital e analógica) do CLP, utilizaremos um recurso muito interessante do software, o modo Simulação.



ANOTAÇÕES

Ajuda



Uma versão não dinâmica da Ajuda online está instalada. Por padrão, ela pode ser acessada através do menu *Ajuda*. A linguagem na qual as páginas de ajuda são exibidas podem ser alteradas no diálogo *Opções no menu Configurações Internacionais*.

Ajuda Sensível ao Contexto

Atalho padrão: <F1>

Para abrir a página de ajuda online pressione <F1> na janela ativa, caixa de diálogo ou comando do menu.

Ao selecionar um comando do menu, será exibida a página de ajuda correspondente.

Se optar por utilizar <F1>, da mesma forma a página de ajuda correspondente será exibida (por exemplo uma palavra-chave, uma função básica ou uma mensagem de erro dentro da janela de mensagens).

3. Programação do Nexto Xpress



O MasterTool IEC XE é totalmente compatível com a IEC 61131-3 e contempla todas as linguagens de programação definidas na norma, incluindo a linguagem LADDER (LD) que utilizaremos nesse tutorial para programação do CP Nexto Xpress. Antes disso, vamos estudar alguns conceitos básicos comuns a todas as linguagens de programação da norma.

Elementos da Aplicação



Uma aplicação é o conjunto dos objetos necessários para executar uma instância específica do programa no Nexto Xpress. É também chamada de Programa Aplicativo. Os principais elementos que constituem uma aplicação estão indicados na sequência.

Unidade de Organização de Programa

O termo POU é utilizado basicamente para todos os objetos usados para criar um programa do CP, e significa Unidade de Organização de Programa.

Uma POU pode ser uma função, bloco funcional, método, ação, interface, DUT ou ainda um arquivo externo de qualquer formato.

Cada POU consiste em uma parte de declaração (janela superior) e uma parte de implementação (janela inferior). O corpo é escrito em uma das linguagens de programação disponíveis, que são ST, SFC, FBD, LD ou CFC.

Programa

Um programa é uma POU que retorna um ou vários valores durante a operação. Todos os valores são mantidos desde a última vez que o programa foi executado até a próxima execução. Um programa pode ser chamado por outra POU, mas a chamada de programa não é permitida em uma função. Não existem instâncias (reproduções/cópias) de programas. No contexto de POUs em forma de programa PRG, é recomendável que estas possuam diversos tipos de variáveis, mas é aconselhável evitar a utilização de VAR_INPUT ou VAR_OUTPUT. Idealmente, as POUs de programa são otimizadas para lógica, utilizando apenas variáveis internas. Para POUs do tipo função, é recomendado apenas um dado de retorno, enquanto para POUs do tipo Bloco Funcional, é permitido o uso de diversos tipos de variáveis, incluindo VAR_INPUT, VAR_OUTPUT, VAR e VAR_IN_OUT.

Função

Uma função é uma POU que produz exatamente um elemento de dados (que pode consistir em um ou vários elementos, tais como campos ou estruturas) ao ser processada e cuja chamada em linguagens textuais pode ocorrer como um operador em expressões. Funções não contêm informações de estado, ou seja, a invocação de uma função com os mesmos argumentos (parâmetros de entrada) sempre produzirá os mesmos valores (saída). Por esta razão, as funções não devem conter variáveis globais e endereços.

Bloco Funcional

Um bloco funcional é uma POU que fornece um ou mais valores durante o processamento de um programa do CP. Ao contrário da função, os valores das variáveis de saída e as necessárias variáveis internas mantêm-se de uma execução do bloco funcional até a próxima execução. Assim sendo, a chamada de um bloco funcional com os mesmos argumentos (parâmetros de entrada) nem sempre produz os mesmos valores de saída.

Um bloco funcional sempre é chamado através de uma instância, que é uma reprodução (cópia) do bloco funcional.

Cada instância tem o seu próprio identificador (nome da instância) e uma estrutura de dados contendo entradas, saídas e variáveis internas.

Assim como as variáveis, as instâncias são declaradas local ou globalmente e, por meio delas, o nome do bloco funcional é indicado como o tipo de dado de um identificador.

A figura a seguir ilustra a chamada da instância "inst1" do bloco funcional "fb", o qual tem uma entrada (in1) e uma saída (out1).

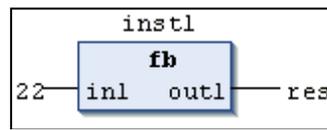


Figura 3-1. Instância "Inst1"

Lista de Variáveis Globais - GVL

Uma Lista de Variáveis Globais, a GVL (🌐) é usada para declarar variáveis globais. Se uma GVL for definida na visualização das POUs, as variáveis estarão disponíveis para todo o projeto. Se uma GVL estiver atribuída a uma determinada aplicação, as variáveis serão válidas dentro desta aplicação.

Gerenciamento de Bibliotecas

Bibliotecas podem fornecer funções, blocos funcionais, tipos de dados, variáveis globais e até mesmo visualizações, que podem ser usadas em um projeto assim como em outras POUs e variáveis definidas diretamente no projeto.

A expressão padrão para um arquivo de biblioteca é .library.

O gerenciamento das bibliotecas em um projeto é feito no Gerenciador de bibliotecas (📚), cuja instalação no sistema é feita previamente no diálogo do Repositório de bibliotecas. A biblioteca standard.library já vem instalada por padrão. Ela contém todas as funções e blocos funcionais necessários em conformidade com a norma IEC61131-3 assim como com as POUs para o programador IEC.

Operadores

Operadores são usados como funções em uma POU. Exemplos de operadores:

- Operadores de atribuição
- Operadores aritméticos
- Operadores de bitstring
- Operadores de deslocamento de bits
- Operadores de seleção
- Operadores de comparação

- Operadores de endereço
- Operadores de chamada
- Funções de conversão de tipo
- Funções numéricas
- Operadores adicionais à norma IEC
- Operadores de escopo adicionais à norma IEC

Operandos

Os operandos são classificados como segue:

- Constantes (BOOL, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME, Número, REAL/LREAL, STRING, Literais Tipados)
- Variáveis
- Endereços
- Funções

Declaração de Variáveis

Para declarar variáveis utilizamos o editor de declaração, o qual consiste em um editor textual ou tabular usualmente disponível em combinação com os editores de linguagem. A classe (ou escopo, no diálogo de declaração) da variável a ser declarada é especificada pelas palavras-chave no âmbito de sua parte de declaração. A declaração de uma variável comum, por exemplo, aparece entre as palavras VAR e END_VAR.

Sintaxe:

```
<IDENTIFICADOR> {AT <ENDEREÇO>};<TIPO> {:=<INICIALIZAÇÃO>};
```

As partes entre chaves {} são opcionais. Por exemplo: a variável pode ser diretamente vinculada a um endereço definido usando a palavra-chave AT. Além disso, o valor de inicialização padrão é 0 para todas as declarações, porém valores de inicialização definidos pelo usuário podem ser adicionados na declaração de cada variável via operador de atribuição ":=".

O identificador é o nome da variável. Algumas regras referentes a indicadores:

- Não deve conter espaços nem caracteres especiais.
- Não diferencia maiúsculas e minúsculas, o que significa que VAR1, Var1 e var1 são todas a mesma variável.
- A_BCD e AB_CD são considerados diferentes identificadores, porém um identificador não deve conter mais de um sublinhado por linha.
- O número de caracteres do identificador (identificação) é ilimitado.

Cada identificador é atribuído a um tipo de dado que determina quanto de espaço de memória será reservado e quais tipos de valores ele armazena. Exemplos:

- BOOL
- Tipos de dados inteiros
- REAL / LREAL
- STRING / WSTRING
- Tipos de dados de tempo

- Tipos de dados definidos pelo usuário

Exemplo de declaração de uma variável:

```
CONTADOR:INT := 12; (* VARIÁVEL INTEIRA COM VALOR INICIAL 12 *)
```

Tipos de Variáveis

Variável de Representação Direta

Variável de representação direta significa que a variável pode ser acessada diretamente na memória utilizando o endereço desejado. Por exemplo: %QB0, %MW100.

Variável Simbólica

Variáveis simbólicas são variáveis IEC criadas em POU's e GVL's durante o desenvolvimento do aplicativo, as quais não são endereçadas diretamente na memória.

Variável de Entrada/Saída de Representação Direta

Variável de entrada/saída de representação direta pode ser utilizada para mapear pontos de entrada/saída analógicos ou digitais. Como referência, 8 pontos de entrada/saída digital podem ser representados por um byte e um ponto de entrada/saída analógica pode ser representado por dois bytes.

Variáveis Remanentes

Variáveis remanentes podem reter os seus valores durante o período de execução normal do programa. A declaração determina o grau de resistência de uma variável remanente no caso de reset, envio ou reinicialização do CP. Principalmente nas aplicações, será requisitado uma combinação de ambas as memórias remanentes.

Variáveis Retentivas

Variáveis declaradas como retentivas serão mantidas dependentes do CP, mas em uma área separada de memória. Elas apresentam a palavra-chave RETAIN na sua declaração em uma POU e em uma lista de variáveis globais. Exemplo:

```
VAR RETAIN
iRem1 : INT; (* 1. Variável retentiva*)
END_VAR
```

Variáveis retentivas são identificadas pela palavra-chave RETAIN. Estas variáveis mantêm seu valor após o desligamento do controlador (inesperado ou normal) e no comando online *Reset a Quente*. Quando o programa for executado novamente, os valores armazenados serão processados posteriormente. Todas outras variáveis são novamente inicializadas, tanto com seus valores inicializados, quanto com suas inicializações padrão.

Ao contrário das variáveis persistentes, as retentivas são reinicializadas em um novo envio do programa. Variáveis retentivas, entretanto, são reinicializadas em um *Reset Origem* e, diferentemente das persistentes, em um *Reset a Frio* e em um envio da aplicação. A propriedade *Retentiva* pode ser combinada com a propriedade *Persistente*.



NOTAS:

- Se uma variável local em um programa for declarada como VAR RETAIN, ela será salva na área retentiva (como uma variável retentiva global).

- Se uma variável local em um bloco funcional for declarada como VAR RETAIN, a instância completa do bloco funcional será salva na área retentiva (todos os dados da POU), onde somente a variável retentiva declarada será tratada como retentiva.
- Se uma variável local em uma função for declarada como VAR RETAIN, então isto não terá efeito. A variável não será salva na área retentiva. Se uma variável local for declarada como persistente em uma função, então a mesma não terá efeito também.

Variáveis Persistentes

Variáveis persistentes são identificadas pela palavra-chave PERSISTENT (VAR_GLOBAL PERSISTENT). Elas são reinicializadas somente após um comando de *Reset Origem*. Ao contrário das variáveis retentivas, elas mantêm o seu valor após um envio. Esse tipo de variável SOMENTE pode ser declarado em uma lista de variáveis globais especial do tipo de objeto *Variáveis Persistentes*, atribuída a uma aplicação. Somente pode haver uma única lista.



NOTA: Uma declaração com VAR_GLOBAL PERSISTENT tem o mesmo efeito que uma declaração com VAR_GLOBAL PERSISTENT RETAIN ou VAR_GLOBAL RETAIN PERSISTENT.

Assim como as variáveis retentivas, as persistentes são armazenadas em uma área de memória separada. Exemplo:

```
VAR GLOBAL PERSISTENT RETAIN
iVarPers1 : DINT; (* 1. Variável Persistente+Retentiva Appl *)
bVarPers  : BOOL; (* 2. Variável Persistente+Retentiva Appl *)
END_VAR
```

NOTA: Somente é possível o uso de variáveis persistentes globais, a não ser que exista algum mecanismo no CP utilizado que permita este tipo de operação.

O dispositivo deve fornecer uma área separada de memória para a lista de variáveis persistentes de cada aplicação. A cada reenvio da aplicação, a lista de variáveis persistentes no CP será verificada no projeto. A lista no CP é identificada pelo nome da aplicação. Se houver inconsistências, o usuário deverá reinicializar todas as variáveis persistentes da aplicação. Inconsistências podem ser resultantes de renomeação ou exclusão de outras modificações na lista de declarações existentes.

Novas declarações somente podem ser adicionadas ao final da lista. Em um envio, elas serão detectadas como novas e não demandarão uma reinicialização da lista completa.

Tabela 3-1. Comportamento das Variáveis Remanentes

Situação	Comportamento		
	VAR	VAR RETAIN	VAR PERSISTENT VAR RETAIN PERSISTENT VAR PERSISTENT RETAIN
Após comando online			
Reset a Quente	-	x	x
Reset a Frio	-	-	x
Reset Origem	-	-	-
Envio da Aplicação	-	-	x
Alteração Online da Aplicação	x	x	x
Reinicializar CP	-	x	x

Unidades de Tipo de Dados

Além dos tipos de dados padrão, o usuário pode utilizar tipos criados por ele. Tipos de estruturas, enumeração e referências podem ser criados como Unidades de Tipos de Dados (DUT) em um editor DUT.

Uma DUT () pode ser adicionada ao projeto através do comando *Acrescentar Objeto*. Para atribuí-la a uma aplicação existente, selecione a aplicação na árvore de *Dispositivos*. Caso contrário, a DUT será adicionada à árvore das *POUs*. No diálogo *Acrescentar DUT*, digite um nome para ela (nova unidade de tipos de dados) e escolha o tipo desejado de *Estrutura*, *Enumeração*, *Pseudônimo* ou *União*.

No caso do tipo *Estrutura*, é possível utilizar o princípio da herança como suporte à programação orientada ao objeto. Opcionalmente, pode ser especificado que uma DUT se estenda à outra DUT que esteja definida no projeto. Isto significa que as definições da DUT estendida serão automaticamente válidas dentro da DUT atual. Para este propósito, ativa-se a opção *Estender* e digita-se o nome da outra DUT.

Após confirmar a configuração, pressione o botão *Abrir*. O editor da nova DUT estará disponível e pode-se iniciar a edição.

Sintaxe para declaração de uma DUT:

```
TYPE <IDENTIFICADOR> : <DECLARAÇÃO DOS COMPONENTES DA DUT>
END_TYPE
```

A declaração dos componentes da DUT depende do seu tipo, por exemplo, se é uma estrutura ou uma enumeração. Exemplo: veja nas duas DUTs abaixo, estruturas de definição struct1 e struct, sendo que struct2 estende struct1, o que significa que se pode usar struct2.A na implementação para acessar a variável A.

```
TYPE STRUCT1 :
STRUCT
A:INT;
B:BOOL;
END_STRUCT
END_TYPE

TYPE STRUCT2 EXTENDS STRUCT1 :
STRUCT
C:DWORD;
D:STRING;
END_STRUCT
END_TYPE
```

Linguagem de Programação Diagrama Ladder (LD)

O Diagrama Ladder é uma linguagem de programação gráfica similar à estrutura de um circuito elétrico. Ele é utilizado para construir intertravamentos lógicos estruturados na forma de redes. Portanto, o LD é bastante útil para controlar a chamada de outras POU's.

O Diagrama Ladder consiste de uma série de redes, sendo cada uma delas limitada a sua direita e esquerda por linhas de corrente verticais (barras de energia). Uma rede contém um diagrama de circuito formado por contatos, bobinas, opcionalmente POU's adicionais (caixas) e por linhas conectoras. No lado esquerdo há uma série de contatos que transmitem a condição ON ou OFF da esquerda para a direita (o que corresponde aos valores booleanos TRUE e FALSE). Para cada contato é atribuída uma variável Booleana. Se esta variável for TRUE, a condição será transmitida da esquerda para a direita ao longo da linha conectora. Caso contrário, o valor OFF transmitirá ON. Assim sendo, as bobinas localizadas na parte direita da rede recebem o valor ON ou OFF vindo da esquerda para a direita (correspondentemente a TRUE ou FALSE para uma variável booleana atribuída).

Exemplo de uma rede LD:

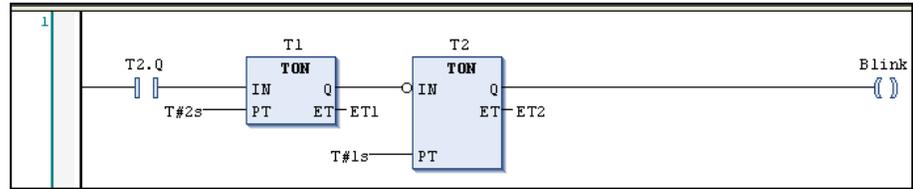


Figura 3-2. Rede LD

Inserindo e Organizando Elementos

Elementos também podem ser diretamente arrastados com o mouse a partir da caixa de ferramentas para a janela do editor ou de uma posição para outra (arrastar&soltrar). Para isto, selecione o elemento com um clique de mouse, mantenha-o pressionado e arraste o mesmo para a respectiva rede na visualização do editor. Assim que a rede tiver sido alcançada, todas as possíveis posições de inserção serão indicadas por marcadores verdes. Ao posicionar o cursor em um destes pontos, o marcador mudará para verde e, soltando o botão, o elemento poderá ser ali posicionado.

Os comandos Recortar, Copiar, Colar e Excluir, por padrão, disponíveis no menu Editar, podem ser usados para organizar elementos. A cópia também pode ser feita via arrastar&soltrar: selecione o elemento na rede com um clique de mouse e, mantendo-o pressionado, arraste-o para a posição destino. Assim que a posição for alcançada (marcador de posição verde), um símbolo de mais será acrescentado ao símbolo do cursor. Solte o botão do mouse e insira o elemento.

Menu LD

Quando o cursor está localizado na janela do editor LD o menu deste está disponível na barra do menu e fornece os comandos para programação na visualização do editor atualmente configurado. A figura abaixo mostra os comandos comuns aos editores FBD/LD/IL.

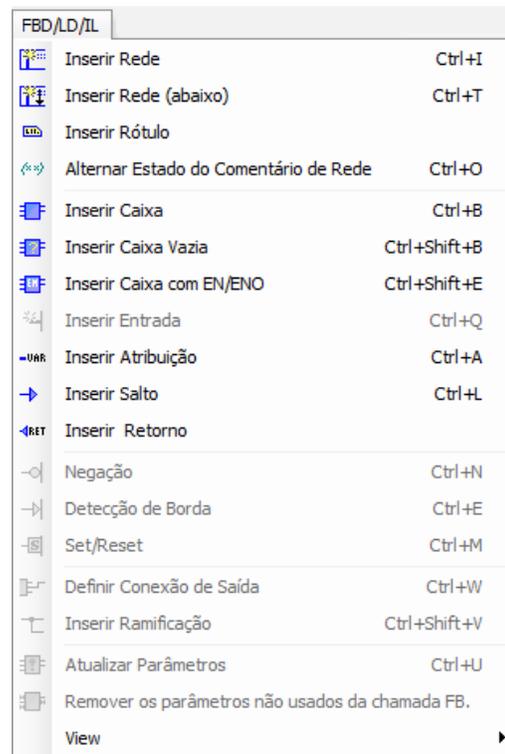


Figura 3-3. Comandos Editores FBD/LD/ID

O editor fornece uma caixa de ferramentas com elementos de programação a serem inseridos na janela do editor (arrastar&soltar). Por padrão, esta caixa de ferramentas pode ser aberta via comando Caixa de Ferramentas no menu Visualização. A figura a seguir mostra a caixa de ferramentas.

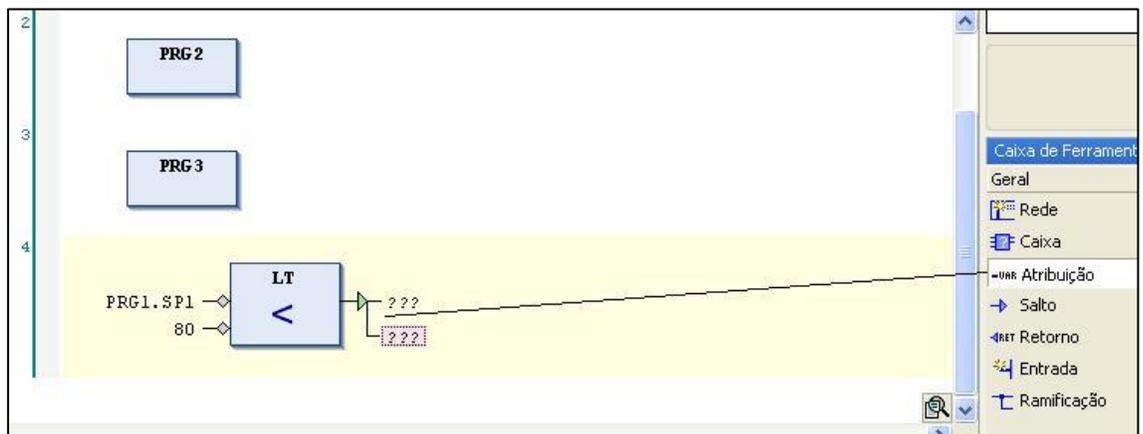


Figura 3-4. Caixa de Ferramentas

Em uma rede podem ser atribuídos, opcionalmente, um título, um comentário e um rótulo. A disponibilidade destes dois primeiros campos título e comentário pode ser ativada ou não no diálogo Opções do Editor FBD, LD. Para acrescentar um rótulo, o qual pode ser endereçado por um salto, usa-se o comando Inserir rótulo. Veja esses itens na figura a seguir.

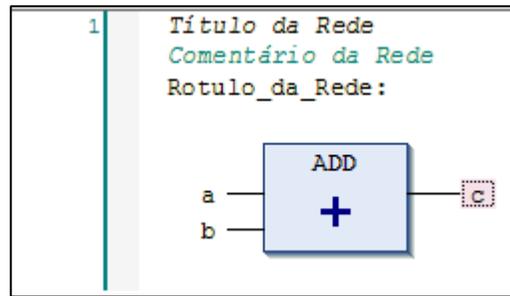


Figura 3-5. Título, Comentário e Rótulo

As redes podem ser configuradas em estado de comentário, o que faz com que a rede não seja processada, mas sim exibida e tratada como um comentário.

Comandos Básicos do LD

Contato

Cada rede LD contém um ou vários contatos na sua parte esquerda. Estes contatos são representados por duas linhas paralelas:

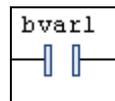


Figura 3-6. Contato

Contatos passam da condição ON (TRUE) ou OFF (FALSE) da esquerda para a direita.

Uma variável booleana é atribuída a cada contato. Se esta variável for TRUE, primeiramente a condição é transmitida da esquerda para a direita e a seguir para uma bobina na parte direita da rede. Caso contrário a conexão direita recebe o valor FALSE.

Múltiplos contatos podem ser conectados tanto em série quanto em paralelo. No caso de dois contatos paralelos somente um deles deve transmitir o valor TRUE para que a ramificação paralela transmita o valor TRUE. No caso de contatos conectados em série todos os contatos devem transmitir a condição TRUE para que o último contato transmita a condição TRUE.

Assim, a organização do contato corresponde tanto a um paralelo elétrico quanto a um circuito série.

Um contato também pode ser negado (barra no símbolo do contato):

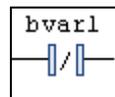


Figura 3-7. Contato Negado

Um contato negado transmite a condição de entrada (TRUE ou FALSE) somente se a variável Booleana atribuída for FALSE. Observe que a caixa de ferramentas fornece diretamente elementos de contatos negados.

Um contato pode ser inserido em uma rede LD através dos comandos:

- Inserir Contato
- Inserir Contato (Direita)
- Inserir Contato Paralelo (Acima)

- Inserir Contato Paralelo (Abaixo)

Estes comandos, por padrão, fazem parte do menu FBD/LD/IL. Alternativamente o elemento pode ser inserido via arrastar e soltar a partir da caixa de ferramentas ou a partir de outra posição no editor.

Bobina

Ao lado direito de uma rede LD pode haver um número indeterminado de bobinas, as quais são representadas como segue.

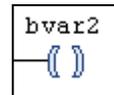


Figura 3-8. Bobina

Elas podem ser organizadas somente **em paralelo**. Uma bobina transmite o valor das conexões da esquerda para a direita e copia os mesmos para uma **variável Booleana** apropriada. Na linha de entrada podem estar presentes os valores ON (TRUE) ou OFF (FALSE). Contatos e bobinas também podem ser negados (barra no símbolo da bobina):

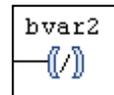


Figura 3-9. Bobina Negada

Neste caso o valor negado do sinal de entrada será copiado para a variável Booleana apropriada. Desta forma um contato negado somente será conectado se o valor Booleano apropriado for FALSE.

Uma bobina pode ser inserida em uma rede através do comando Inserir Atribuição, o qual, por padrão, faz parte do menu FBD/LD. Alternativamente o elemento pode ser inserido via arrastar e soltar a partir da caixa de ferramentas (elementos Ladder) ou a partir de outra posição no editor.

As bobinas do tipo Set (S) e Reset (R) permitem reter o estado a elas atribuído (TRUE ou FALSE). Veja um exemplo na figura abaixo.

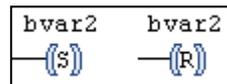


Figura 3-10. Bobinas Set e Reset

Editor LD no Modo Online

No modo online o Editor LD fornece visualizações para a monitoração, escrita e forçamento das variáveis e expressões no CP. A funcionalidade de depuração (breakpoints, passo a passo etc.) também está disponível. A figura a seguir mostra um exemplo de visualização online do Editor LD.

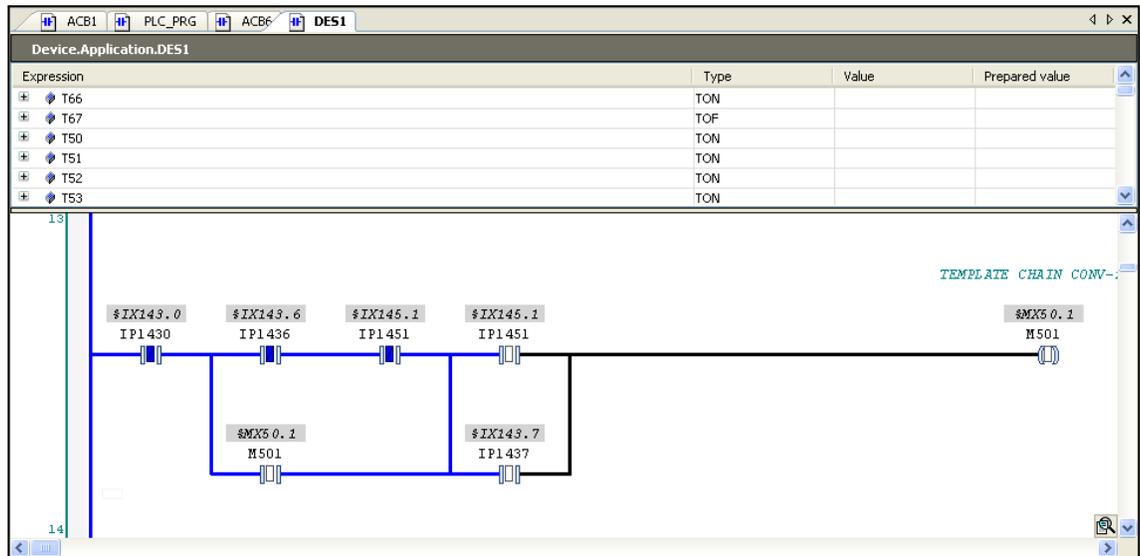


Figura 3-11. Visualização Online do Editor LD

Forçamento/Escreita de Variáveis

No modo online é possível preparar um valor para forçamento ou escrita de uma variável tanto no editor de declaração quanto na parte de implementação. Na parte de implementação com um clique de mouse na variável, o seguinte diálogo se abrirá:

The dialog box 'Preparar Valor' contains the following fields and options:

- Expressão: UserPrg.fbinst.out
- Tipo: INT
- Current value: 13
- O que você quer fazer?
 - Prepara um novo valor para a próxima operação de escrita ou força
 - 13
 - Remover a preparação com um valor.
 - Liberar o forçamento, sem modificar o valor.
 - Liberar o forçamento e restaurar a variável ao valor que tinha antes de forçá-la.

Buttons: OK, Cancelar

Figura 3-12. Diálogo Preparar Valor

Neste diálogo encontra-se nome da variável complementado pelo seu caminho na árvore de dispositivos (Expressão), seu tipo e seu valor atual. Ativando o item correspondente o usuário escolhe se deseja:

- Preparar um novo valor que deve ser inserido no campo de edição
- Remover um valor preparado,

- Liberar a variável atualmente forçada,
- Liberar a variável atualmente forçada e levá-la ao valor a ela atribuído antes do forçamento (reset).

A ação selecionada será efetivada com a execução do comando Forçar Valores (menu Online) ou através da tecla F7.

Criação de POUs

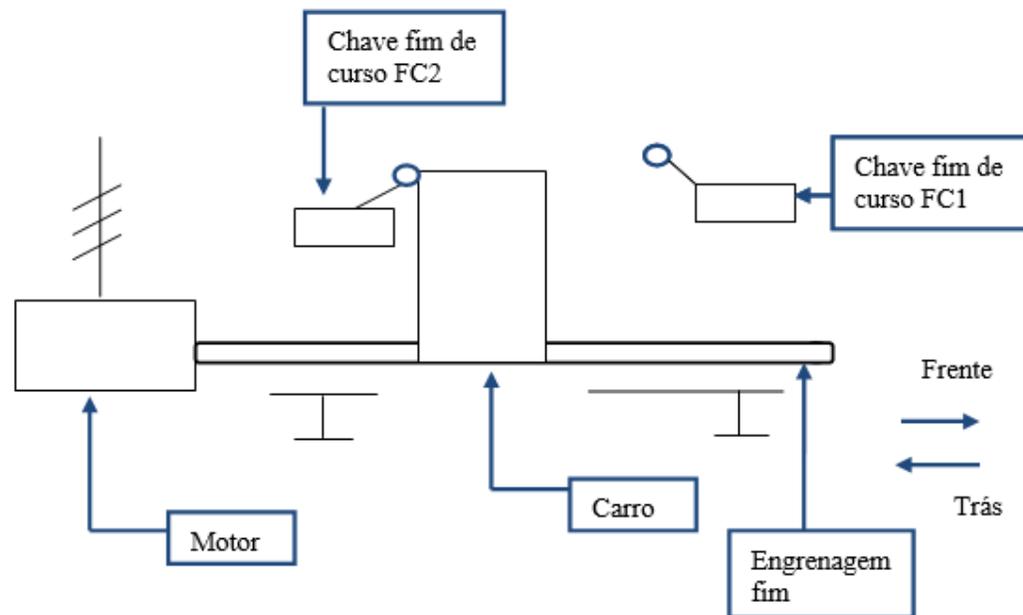


Para acrescentar uma POU (☐), selecione o item correspondente na árvore das POUs ou Dispositivos (uma aplicação, por exemplo), use o comando Acrescentar objeto no menu de contexto e selecione POU no submenu que aparecerá. O diálogo Acrescentar POU abrirá para que se configure o seu nome, tipo e linguagem de implementação.

Estudo Dirigido 3-1: programa para controle de um carro Automático de máquina operatriz



Para ilustrar a utilização de unidades de organização de programa implemente - no projeto "MP11_[ALUNO].project" - uma aplicação referente a um "Carro Automático para Máquina Operatriz" utilizando a linguagem LD. A figura a seguir ilustra esse sistema.



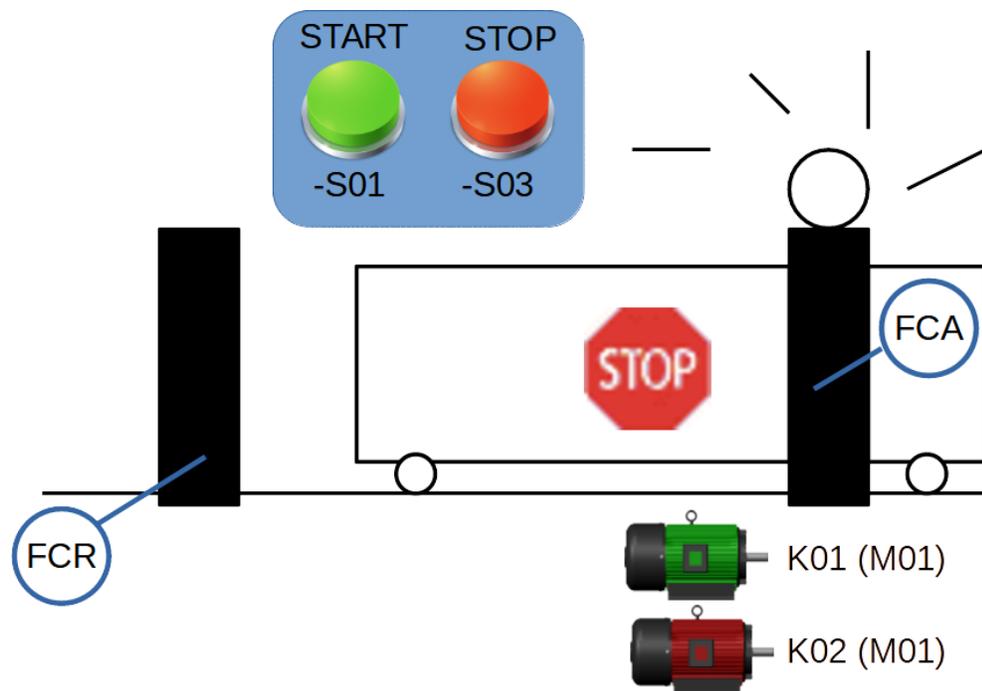
O sistema prevê dois modos de operação: automático e manual. No comando manual, pressionando-se o botão de avanço (S01) é energizado o contator de avanço do carro (K01). Por outro lado: S02 (botão de recuo) energiza K02 (contator de recuo). Neste modo, o avanço/recuo somente se processa pressionando-se o botão de parada (S03) antes de cada reversão. No comando automático, ao acionar o fim de curso de recuo (FCR) inicia a temporização de avanço (Temporizador_A). Transcorrido um tempo de 10 s, o contator de avanço (K01) é energizado, movimentando o carro. Ao atingir o fim de curso de avanço (FCA) o motor para. Neste ponto inicia-se a temporização de recuo (Temporizador_R). Transcorrido um tempo de 10 s, o contator de recuo (K02) é energizado, retornando o carro. Atingindo novamente o fim de curso de recuo (FCR) o ciclo se repete (contatores K01 e K02 desenergizados). O botão de STOP para tudo a qualquer momento.

Estudo Dirigido 3-2: programa para controle de um portão automático



Desenvolva - no projeto "MP11_[ALUNO].project", uma aplicação para controle de um "Portão automático com sistema de batente", conforme figura e orientações abaixo. Trata-se de um sistema para abrir e fechar automaticamente um portão industrial. O funcionamento deste portão se dá de acordo com a operação descrita abaixo:

1. A bobina do contador K01 é energizada através do botão S01 de START (ou de um controle remoto). Esta ação movimenta a abertura do portão.
2. O portão parará quando o batente acionar mecanicamente a chave de fim de curso FCA.
3. O timer é ativado no período em que o portão permanece aberto. Finalizado este tempo a bobina do contador K02 de fechamento do portão é energizada.
4. Ao tocar a chave fim de curso FCR, o portão se fecha e o ciclo é concluído.
5. Um botão de STOP (S03) paralisa tudo a qualquer momento.



ANOTAÇÕES

Utilização de blocos funcionais definidos pelo usuário

As POUs do tipo Bloco Funcional também podem ser criadas pelo usuário. Tais POUs são muito utilizadas para reutilização de código e permitem a estruturação de programas em conformidade com as diretrizes da norma IEC61131-3.

Estudo Dirigido 3-3: bloco funcional de controle de motores de partida direta



Para compreender a utilização de uma POU do tipo FUNCTION_BLOCK implemente - no projeto "MP11_[ALUNO].project" - um bloco funcional para o controle de motores com partida direta (Motor_PD), utilizando a linguagem LD. O funcionamento do bloco funcional deve obedecer às diretivas indicadas a seguir.

Deve ser previsto o controle do motor nos modos manual e automático via variável MAN_Auto. Em manual (MAN_Auto = TRUE) o motor pode ser comandado pelas entradas Liga/Desl. Em automático o motor é comandado (ligado/desligado) unicamente pela entrada Libera, a qual permite que lógicas externas possam ligar e desligar o motor de acordo com as necessidades de intertravamento. As saídas do bloco são: Ligar para comando do contator do motor e Falha que, em TRUE, indica algum defeito no motor (sobrecarga, problema na partida etc.). A entrada "Funcionando" serve para detectar que o motor entrou em funcionamento após a saída "Ligar" ser acionada. Se esta entrada não estiver em TRUE 1,5 s depois do acionamento do motor - monitorado pela instância Retardo do bloco funcional TON - a saída de falha será acionada. A instância Pulso do bloco funcional R_TRIG assegura que somente a transição do sinal será utilizada para ligar o motor em manual.

Um exemplo de declaração de variáveis para esse bloco funcional está mostrado na figura a seguir.

```

1  FUNCTION_BLOCK Motor_PD
2  VAR_INPUT
3      Liga, Desl : BOOL; (* Comando em Manual *)
4      Libera     : BOOL; (* Comando em Automático *)
5      Funcionando : BOOL; (* Motor funcionando *)
6      MAN_auto   : BOOL; (* Manual = TRUE & Automático = FALSE *)
7      Defeito    : BOOL; (* Defeito no motor *)
8      Sobrecarga : BOOL; (* Sobrecarga no motor *)
9  END_VAR
10 VAR_OUTPUT
11     Ligar : BOOL; (* Comando elétrico *)
12     Falha : BOOL; (* Motor em falha *)
13 END_VAR
14 VAR
15     Pulso : R_TRIG; (* Transição para ligar em manual *)
16     Mem01 : RS;     (* Memoriza acionamento *)
17     Retardo : TON;  (* Retardo alarme partida *)
18     Desliga : BOOL; (* Lógica de desligamento *)
19 END_VAR
20

```

ANOTAÇÕES



Um pouco de teoria... Programação Padronizada Multi-linguagens

Desde o advento dos controladores programáveis, muitas linguagens têm sido utilizadas para escrever programas para máquinas e processos.

O resultado desta falta de padronização acaba se refletindo na necessidade de treinamentos em diferentes equipamentos e formação de equipes de manutenção específicas em determinados fabricantes. A consequência direta, muitas vezes não percebida pelos usuários, é a perda de tempo e dinheiro.

Para atenuar este problema, um grupo formado pela organização internacional IEC (*International Electrotechnical Commission*) definiu uma norma para vários aspectos dos controladores, desde características do hardware, instalação, testes, comunicação e programação.

Especificamente a norma IEC61131-3 (parte 3) estabelece as principais características para programação de controladores. Estas características definem o modelo de software e cobre as 5 linguagens mais utilizadas em todo mundo: Function Block Diagram (FBD), Ladder Diagram (LD), Sequential Function Chart (SFC), Structured Text (ST) e Instruction List (IL).

Dentre as principais vantagens da norma podemos destacar a facilidade que o usuário tem em modularizar e estruturar a programação em elementos funcionais ou "POUs" (Program Organization Units), bem como poder definir a linguagem em que irá programar determinada parte do projeto, além de estar utilizando um ambiente de programação world-wide onde o usuário, aprendendo as linguagens da norma, poderá usar este conhecimento em diferentes ambientes de programação (fabricantes). Além disso, o modelo de software permite a reutilização de código através da utilização de biblioteca de blocos funcionais, facilitando o desenvolvimento, implantação e manutenção dos sistemas e aumentando a qualidade do software.

Os programas ou parte deles poderão ser usados entre os ambientes de programação através da importação e exportação de módulos.

Linguagens de Programação IEC61131-3

IOIO
IOIO

Além das linguagens de programação tradicionais, tais como Diagrama LADDER (LD), e Diagrama de Blocos Funcionais (FBD), o MasterTool IEC XE permite a edição de POU's utilizando linguagens de alto nível e muito poderosas. Nesse contexto estão incluídas o Texto Estruturado (ST) e o Sequenciamento Gráfico de Funções (SFC). Esta seção explora o leque de linguagens suportadas pelo MasterTool IEC XE.

Diagrama de Blocos Funcionais (FBD)

O Diagrama de Blocos Funcionais é uma linguagem de programação orientada graficamente. Ele funciona com uma lista de redes, cada qual com uma estrutura gráfica de caixas e linhas de conexão que representam tanto uma expressão lógica quanto aritmética, uma chamada de um bloco funcional, um salto ou uma instrução de retorno.

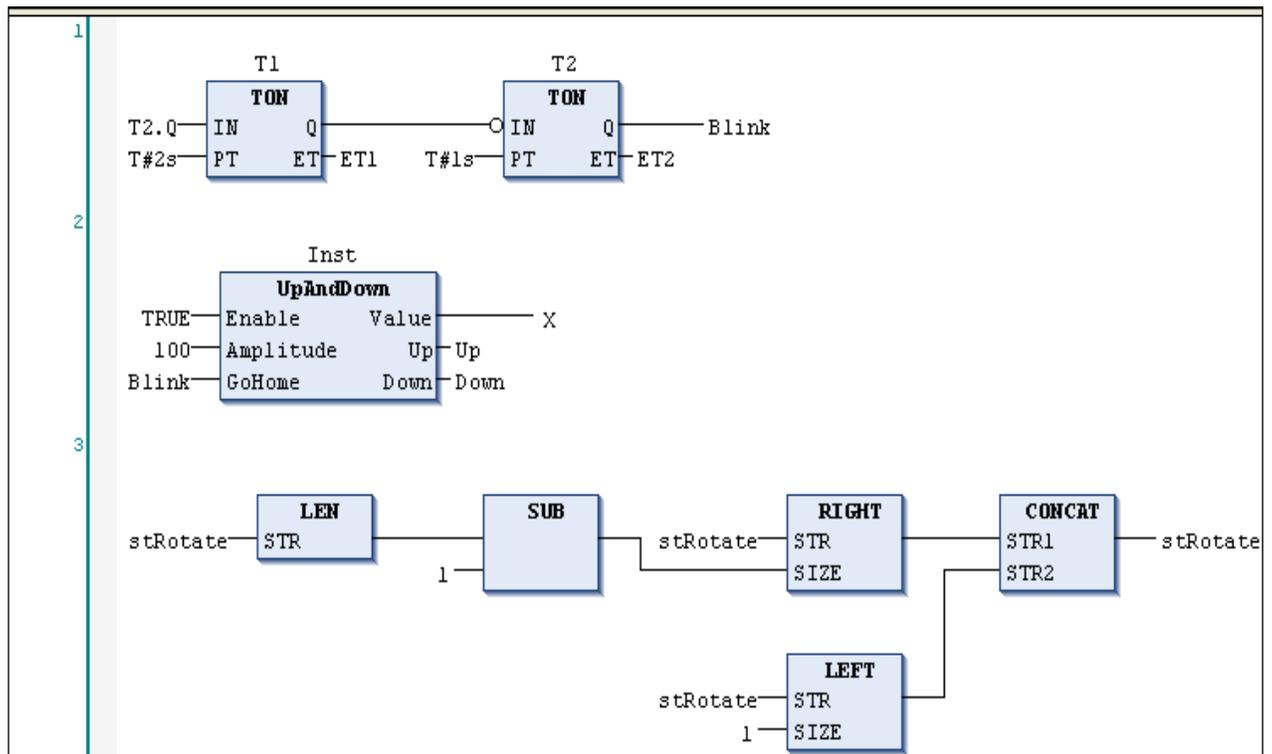


Figura 3-13. Redes de Diagrama de Blocos Funcionais

Pesquisa na Documentação: editor FBD/LD

Consulte, na documentação do produto, o seguinte tópico relacionado à linguagem de programação Diagrama de Blocos Funcionais: Editor FBD/LD.

Fonte: Manual de Programação IEC 61131 – MP399048

Texto Estruturado (ST)

Texto estruturado é uma linguagem de programação textual de alto nível, similar a PASCAL ou C. O código de programa é composto de expressões e instruções. Inúmeras construções podem ser usadas para laços de programação, permitindo, assim, o desenvolvimento de complexos algoritmos.

Exemplo:

```

IF value < 7 THEN
WHILE value < 8 DO
value:=value+1;
END_WHILE;
END_IF;

```

**Pesquisa na Documentação: editor ST**

Consulte, na documentação do produto, os seguintes tópicos relacionados à linguagem de programação Texto Estruturado:

- Texto Estruturado (ST) / Texto Estruturado Estendido (ExST);
- Editor ST.

Fonte: Manual de Programação IEC 61131 – MP399048

Sequenciamento Gráfico de Funções (SFC)

O Sequenciamento Gráfico de Funções (SFC) é uma linguagem gráfica orientada que permite descrever a ordem cronológica de ações determinadas em um programa. Estas ações estão disponíveis como objetos de programação separados, escritos em qualquer linguagem de programação disponível. Em um SFC, elas são atribuídas a elementos de passo e a sequência de processamento é controlada por elementos de transição.

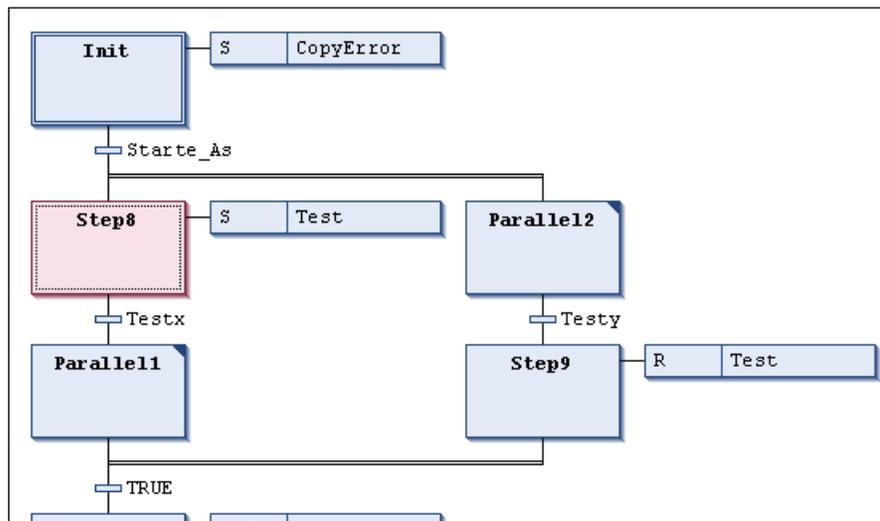


Figura 3-14. Exemplo para uma Sequência de Passos em um Módulo SFC

**Pesquisa na Documentação: editor SFC**

Consulte, na documentação do produto, os seguintes tópicos relacionados à linguagem de programação Sequenciamento Gráfico de Funções:

- Editor SFC;
- Trabalhando no Editor SFC;
- Elemento SFC / Caixa de ferramentas;
- Qualificador;
- Variáveis Implícitas- Memórias SFC;

- Sequência de Processamento no SFC.

Fonte: Manual de Programação IEC 61131 – MP399048

Estudo Dirigido 3-4: bloco funcional de controle de motores com reversão



Implemente - no projeto “MP11_[ALUNO].project” - um bloco funcional (MOTOR_RV) para o controle de motores com partida reversível, utilizando a linguagem SFC. O funcionamento do bloco funcional deve obedecer às seguintes diretrizes:

- O motor elétrico pode ser acionado nos dois sentidos com reversão do sentido de rotação.
- O motor não deve entrar em funcionamento ou deve ser desligado no caso de sobrecarga no motor.
- O ligamento/desligamento do motor pode ser feito utilizando-se memórias biestáveis.
- Para evitar que o motor que se encontra em um sentido de rotação seja ligado em sentido contrário, deve ser incluído um intertravamento no circuito.
- O desligamento deve ser único.

Este Bloco Funcional deverá ser chamado na aplicação “Carro Automático para Máquina Operatriz” desenvolvido anteriormente, o qual prevê a utilização de um motor reversível para movimentação do carro. O desenvolvimento do bloco funcional pode ser realizado seguindo a sequência indicada a seguir.

1. Inclusão da nova POU na Árvore de Dispositivos;
2. Criação da POU tipo Bloco Funcional *Motor_RV*;
3. Declaração das Variáveis de Entrada (VAR_INPUT), saída (VAR_OUTPUT) e internas (VAR);
4. Edição do corpo (redes) do bloco funcional, por exemplo, comando à direita (rede 1) e comando à esquerda (rede 2);
5. Chamada do bloco funcional na aplicação.

ANOTAÇÕES

ANOTAÇÕES



Um pouco de teoria... IEC61131 – Parte 8: orientações para aplicação e implementação das linguagens de programação

A parte 8 da norma IEC foi elaborada com o objetivo de orientar a implementação dos elementos comuns e das linguagens de programação definidas na parte 3 em sistemas de CP em ambientes de programação (PSE). A norma IEC 61131 foi elaborada com o propósito de estimular a aplicação das melhores práticas para sistemas de controle de processos usando CPs. O avanço tecnológico dos CPs juntamente com o da engenharia de software e das redes de comunicação permite a utilização de vários recursos e funcionalidades que são abordados pela norma. Os usuários de CP devem ficar atentos para a quebra de paradigmas oriundos das características e restrições dos controladores programáveis tradicionais, sejam estes CLPs, SDCDs ou outros. Várias orientações sobre como devem ser entendidos e utilizados os elementos de software da

parte 3 da norma são apresentadas na parte 8. Algumas orientações de destaque são mostradas a seguir.

1. Aplicação

Utilização de Linguagens Adicionais

A criação de Funções e Blocos Funcionais pode ser feita através do uso de linguagens adicionais, como C e Pascal, desde que sejam obedecidas as exigências da interface de dados e as formas de chamadas definidas pela norma.

Execução de Ações em SFC

A norma define que uma ação irá executar no mínimo duas vezes, uma quando a ação é ativada e uma última quando a ação é desativada, com exceção para os qualificadores PO e P1. Para os casos de ações simples que referenciam diretamente variáveis booleanas isto não é relevante porque a saída do bloco de ação estará em FALSE na última execução. Entretanto, para ações que compreendem a execução de um conjunto de instruções, o usuário deve tomar as precauções necessárias para que a última execução não provoque efeitos indesejados,

Varredura de Redes em SFC

A título de sugestão, a parte 8 apresenta uma possibilidade para implementação da forma de varredura de uma rede SFC:

- Determinar o conjunto de passos ativos. Esse conjunto será o dos passos iniciais, após a inicialização da rede ou, caso contrário, determinado pela desativação dos passos precedentes e pela ativação dos passos seguintes para a lista de transições disparadas.
- Determinar o estado da saída de todos os blocos de ações e executar a varredura final qualquer ação associada com a transição de descida para a saída Q do respectivo bloco de ação.
- Fazer a varredura de todas as ações associadas à saída Q com valor TRUE dos respectivos blocos de ação.
- Determinar as transições a serem disparadas na próxima varredura.

Sistemas Operacionais para Tempo Real

Os sistemas operacionais para uso em CPs devem ser adequados para atendimento das exigências de aplicações de tempo real. O crescente aumento de velocidade e complexidade para as aplicações usando CPs restringe muito a utilização de sistemas mono tarefas de scan simples. A norma IEC utiliza elementos para usufruir das capacidades de multiprocessamento e multitarefa dos modernos CPs, deforma a garantir a responsividade do sistema. A utilização de escalonamento preemptivo aumenta a capacidade de responsividade e eficiência do PC, mesmo sendo necessário o chaveamento de contexto durante a preempção de tarefas.

Uso de Variáveis Globais

O uso excessivo de variáveis globais vai contra os princípios de encapsulamento, reduzindo confiabilidade, manutenibilidade e reutilização de software. A escrita de variáveis globais em mais de um lugar no programa deve ser evitada. Variáveis globais nunca devem ser usadas para comunicação assíncrona entre programas quando a concorrência de dados for um fator determinante. Recomenda-se o uso de variáveis globais principalmente para: definição de caminhos de acesso para comunicação externa e fornecimento de valores comuns para POUS, principalmente programas.

Invocações Múltiplas de uma Instância em FBD

A invocação múltipla da mesma instância de um bloco funcional é permitida nas linguagens, mas deve ser evitada na linguagem FBD, por questões de coerência com a cultura dos usuários desta linguagem.

Acoplamento de Redes em SFC

Efeitos inesperados podem ocorrer para máquinas de estados acoplados, incluindo SFCs. O abraço mortal (deadly embrace), semelhante ao deadlock, é um efeito que pode ocorrer quando uma rede SFC fica esperando por uma transição de outra rede e esta condição nunca ocorre. As medidas a seguir são recomendadas para evitar e diagnosticar tais condições:

- Evitar o uso de flags de passo "Passo X em transições, com o objetivo de detectar que ações de outros passos foram completadas, especialmente para qualificadores diferentes de "N".
- Sempre que possível, evitar o uso de ações nas quais a variável de indicação de um passo de um SFC interno seja utilizada como a condição de transição do passo externo em que a ação esteja associada, pois isso cria um acoplamento.
- Sempre que possível, encapsular o SFC em um bloco funcional e usar a programação em FBD para externar aos acoplamentos do SFC interno à instância com os outros blocos funcionais em SFC.
- Garantir que a rede SFC seja terminável e segura.

Modificação Dinâmica da Prioridade de Tarefas

Como as tarefas definidas pela norma não são, necessariamente, mapeamentos diretos para tarefas do sistema operacional, é altamente recomendável que não seja permitida a modificação dinâmica de prioridade de tarefas, pois isso pode levar a efeitos imprevisíveis.

2. Implementação

Funções

No caso da ocorrência de erros no processamento de uma função, a ação esperada para o sistema é que a saída ENO da função seja resetada para FALSE. Dependendo da implementação, uma condição de erro pode disparar uma tarefa do usuário ou do sistema para tratamento de erros no final da execução da função. Um programa do usuário poderá ser associado a esta tarefa.

Blocos Funcionais

Nos casos em que uma instância de bloco funcional estiver associada a uma tarefa diferente do programa que faz a sua invocação, a invocação pelo programa fará apenas a passagem de parâmetros de entrada, sem disparar a execução do código do corpo do bloco funcional, que acontecerá sob o controle da tarefa associada à instância. No caso da ocorrência de erros no processamento de funções pertencentes ao corpo de blocos funcionais, o mesmo tratamento de erros mencionado anteriormente para as funções é esperado.

Tratamento de Erros

A parte 3 define os tratamentos de erros para sistemas compatíveis com a norma. Os erros associados à sintaxe ou à configuração no programa fonte podem ser detectados durante a edição do programa pelo usuário no ambiente de programação, durante a análise de sintaxe (parsing) na compilação, durante o ligamento (linking) das POUS ou durante o carregamento do aplicativo para execução no CP. Os erros em tempo de execução devem ser tratados pelo sistema CP destacando a classificação do seu tipo e da sua fonte, uma "divisão por zero" em uma POU, por exemplo. O sistema pode permitir a criação de procedimentos definidos pelo usuário para tratamento de erros.

Requisitos para Ambientes de Programação (PSE)

A parte 8 enfatiza alguns requisitos que os implementadores devem observar para criação de ambientes de programação. Os requisitos apresentados abrangem os seguintes tópicos:

- Interface com o usuário.
- Programação de POUS.
- Criação da configuração, independente da programação da aplicação.

- Compilação separada da interface e do corpo de POUS.
- Ligamento (linking) de elementos de configuração com programas.
- Gerenciamento de bibliotecas.
- Ferramentas de análise.
- Requisitos de documentação.
- Proteção de dados e programas.
- Facilidades para uso on-line.

Fonte: Aplicando a norma IEC 61131 na automação de processos / Marcos de Oliveira Fonseca, Constantino Seixas Filho, João Aristides Bottura Filho. - São Paulo: ISA Distrito 4, 2008.

Elementos Avançados da Aplicação

Método

Assim como na programação orientada a objetos, os *Métodos* (M) podem ser usados para descrever uma sequência de instruções. Assim como a função, o método não é uma POU independente, mas deve ser atribuído a um bloco funcional. Pode ser considerado como uma função que contém uma instância do respectivo bloco funcional. Esta funcionalidade somente está disponível se suportada pela configuração atual.

Propriedade

Uma *Propriedade* (P) é um tipo de objeto que pode ser inserido em um programa ou bloco funcional através do comando *Acrescentar Objeto* e após *Propriedade...* no menu de contexto. No diálogo *Acrescentar Propriedade*, o nome, o tipo de retorno e a linguagem de implementação desejados devem ser especificados. Esta funcionalidade somente está disponível se suportada pela configuração atual. Ela contém dois métodos especiais que serão inseridos automaticamente na árvore de objetos no item correspondente:

- O método Set é chamado quando a propriedade é escrita, ou seja, o nome da propriedade é usado como entrada
- O método Get é chamado quando a propriedade é lida, ou seja, o nome da propriedade é usado como saída

Veja o exemplo onde o bloco funcional FB1 usa uma variável local MILLI. Esta variável é determinada pelas propriedades Set e Get:

(* Código na propriedade Get: *)

```
SECONDS := MILLI / 1000;
```

(* Código na propriedade Set: *)

```
MILLI := SECONDS * 1000;
```

A propriedade do bloco funcional pode ser escrita (método Set), por exemplo, por fbinst.seconds := 22; (fbinst é a instância de FB1) ou pode ser lida (método Get), por exemplo, por testvar := fbinst.seconds;.

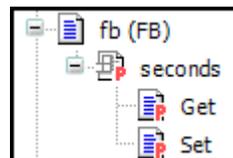


Figura 3-15. Propriedade Segundos Acrescentada ao Bloco Funcional fb

Propriedades podem ter variáveis locais adicionais, porém não tem entradas adicionais e - diferentemente da função ou método - não tem saídas adicionais.

Ação

Ações (A) podem ser definidas e atribuídas a blocos funcionais e programas através do comando *Acréscitar Objeto*. A ação é uma implementação adicional que pode ser criada em uma linguagem diferente da implementação básica. A cada ação, é referenciado um nome.

A norma IEC não reconhece outras ações que não sejam do Sequenciamento Gráfico de Funções (SFC). Nesta linguagem, as ações são parte essencial e contém instruções a serem processadas nos passos específicos do gráfico.

Uma ação manipula os dados do bloco funcional ou programa ao qual ela pertence. Ela utiliza as variáveis de entrada/saída e variáveis locais definidas nessas POU's e, por isso, não contém declarações próprias.

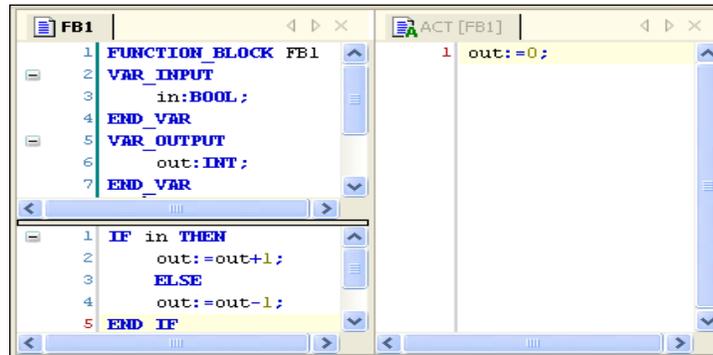


Figura 3-16. Ação em um Bloco Funcional

Neste exemplo, cada chamada do bloco funcional FB1 incrementará ou decrementará a variável de saída Out, dependendo do valor da variável de entrada In. Chamando a ação Reset deste bloco funcional, a variável de saída "Out" será zerada. É escrito na mesma variável "Out" nos dois casos.

Pode-se adicionar uma ação via comando *Acréscitar Objeto/Ação*, quando o respectivo programa ou bloco funcional for selecionado na árvore dos Dispositivos ou das POU's. No diálogo *Acréscitar Ação*, defina o nome da ação e a linguagem de implementação desejada.

Uma ação é chamada com:

```
<NOME_PROGRAMA>.<NOME_AÇÃO> OU <NOME_INSTÂNCIA>.<NOME_AÇÃO>.
```

Observe as particularidades da notação em FBD (exemplo abaixo).

Para chamar uma ação dentro do seu próprio bloco (dentro do programa ou bloco funcional ao qual pertence), basta referenciar o nome da ação.

Exemplos para a chamada da ação descrita acima a partir de outra POU:

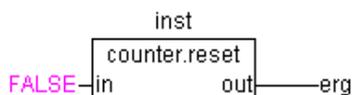
Declaração para todos os exemplos:

```
PROGRAM USERPRG
VAR
  INST : COUNTER;
END_VAR
```

Chamada da ação Reset em outra POU programada em ST:

```
INST.RESET (IN := FALSE);
ERG := INST.OUT;
```

Chamada da ação Reset em outra POU programada em FBD:



4. Configuração do Nexto Xpress



Configuração da UCP do Controlador



Os controladores Nexto Xpress são configurados e programados através do software MasterTool IEC XE. A configuração define o comportamento e o uso de periféricos e recursos especiais do controlador. A programação representa a aplicação desenvolvida pelo usuário, também conhecida como Application.

Parâmetros Gerais

Os parâmetros relativos à UCP do controlador estão localizados na visualização em árvore sob o item XP3xx, logo abaixo de *Configuration*. Cada item deve ser propriamente verificado para a correta execução do projeto conforme ilustrado na figura a seguir.

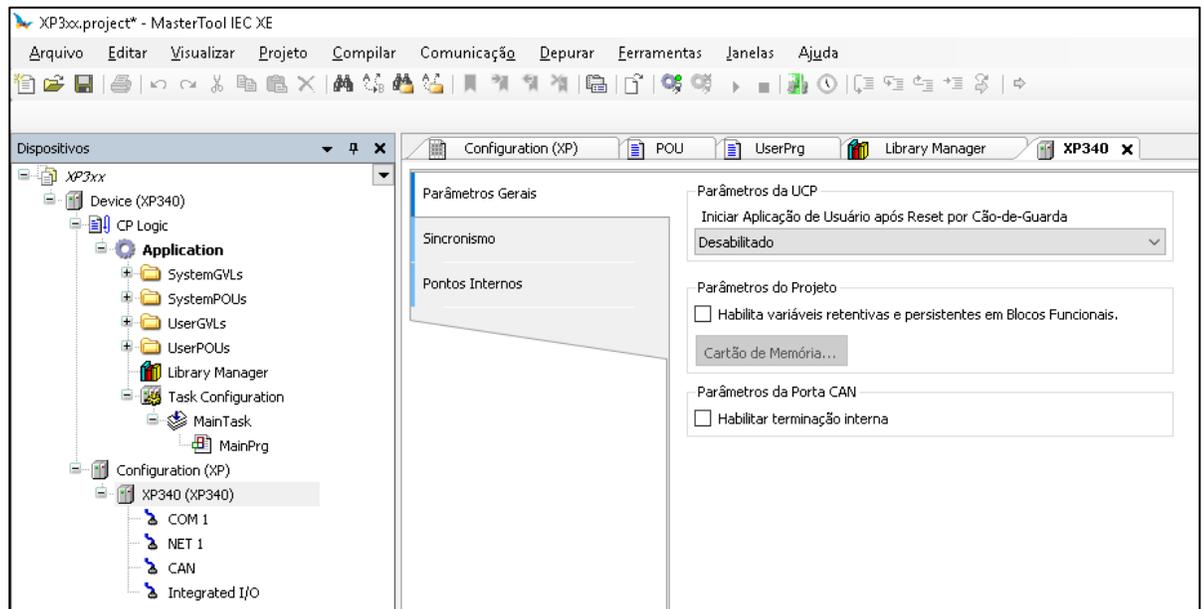


Figura 4-1. Configuração da UCP

Sincronização Temporal

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

Para usar os protocolos de sincronismo, o usuário deve configurar os parâmetros na aba Sincronismo localizada dentro da Configuração da UCP, em XP3xx, na visualização em árvore de acordo com o ilustrado na figura a seguir.

Figura 4-2. Configuração SNTP

Caso o sincronismo seja realizado através do protocolo IEC 60870-5-104, o usuário deve ativar a sincronização de horário na tela de configuração do protocolo para receber a sincronização do relógio. Para definir esta opção no dispositivo, verifique o parâmetro Habilitar Sincronismo disponível na seção Camada de Aplicação do Manual de Utilização UCPs Série Nexto código MU214100.

Se o CP receber um comando de sincronização de tempo do centro de controle e esta opção estiver desativada, uma resposta de erro será retornada para esse comando. Mas se essa opção estiver ativada, uma mensagem de sucesso será retornada ao centro de controle, mesmo que o comando de sincronização seja descartado, pois existe outro método de sincronismo ativo com prioridade mais alta.

Esse método de sincronismo deve ser usado apenas como um método auxiliar de sincronismo, uma vez que a precisão do processo de sincronização do relógio depende muito de atrasos e tráfego na rede, bem como da carga do processador na UCP, pois esse mecanismo é tratado por uma tarefa de baixa prioridade.

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

O controlador envia as requisições de sincronização cíclicas, de acordo com o tempo configurado no campo Período de Sincronização do SNTP. Na primeira tentativa de sincronização, logo após a inicialização do serviço, a requisição é para o primeiro servidor, configurado em Endereço de IP do 1º Servidor. Caso este não responda, as requisições são direcionadas para o segundo servidor configurado em Endereço de IP do 2º Servidor, fornecendo uma redundância de servidores SNTP.

Caso o segundo servidor também não responda, o mesmo processo de tentativa de sincronização é executado novamente, mas apenas após o Período de Sincronização ter passado. Ou seja, a cada período de sincronização, o controlador tenta se conectar uma vez em cada servidor, ele tenta o segundo caso o primeiro não responda. O tempo de espera por uma resposta do servidor SNTP é definido por padrão em 5 segundos e não pode ser modificado.

Caso, após uma requisição de sincronização, a diferença entre o horário atual do controlador e o recebido pelo servidor for maior que o valor configurado no parâmetro Erro Mínimo Antes da Atualização do Relógio, o horário do controlador é atualizado.

O SNTP usa o horário no formato UTC (*Universal Time Coordinated*), logo o parâmetro de Fuso Horário deve ser configurado corretamente, para que o horário lido pelo SNTP seja convertido corretamente para a hora local. O processo de execução do cliente SNTP pode ser exemplificado com os seguintes passos:

1. Tentativa de sincronização através do primeiro servidor. Caso a sincronização ocorra com sucesso, o controlador aguarda o tempo para a nova sincronização (Período de Sincronização) e tentará sincronizar-se novamente com este servidor, utilizando, então, este como servidor primário. Em caso de falha (o servidor não responde em menos de 5 s) o passo 2 é executado;
2. Tentativa de sincronização através do segundo servidor. Caso a sincronização ocorra com sucesso, o controlador aguarda o tempo para a nova sincronização (Período de Sincronização) e tentará sincronizar-se novamente com este servidor, utilizando, então, este como servidor primário. Em caso de falha (o servidor não responde em menos de 5 s) é aguardado o tempo referente ao Período de Sincronização e executado novamente o passo 1.

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

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



NOTA:

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

Pontos Internos

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

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

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

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

A configuração dos pontos internos está mostrada na figura a seguir. É possível a configuração de até 5120 entradas na tabela de Pontos Internos.

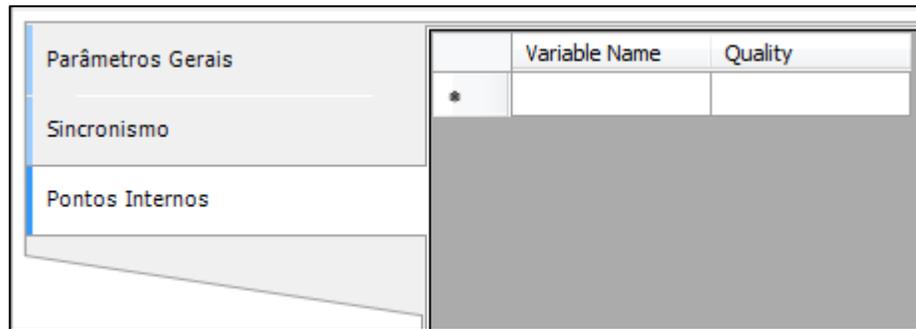


Figura 4-3. Tela de Configuração dos Pontos Internos

A figura a seguir mostra um exemplo de configuração de dois pontos internos:

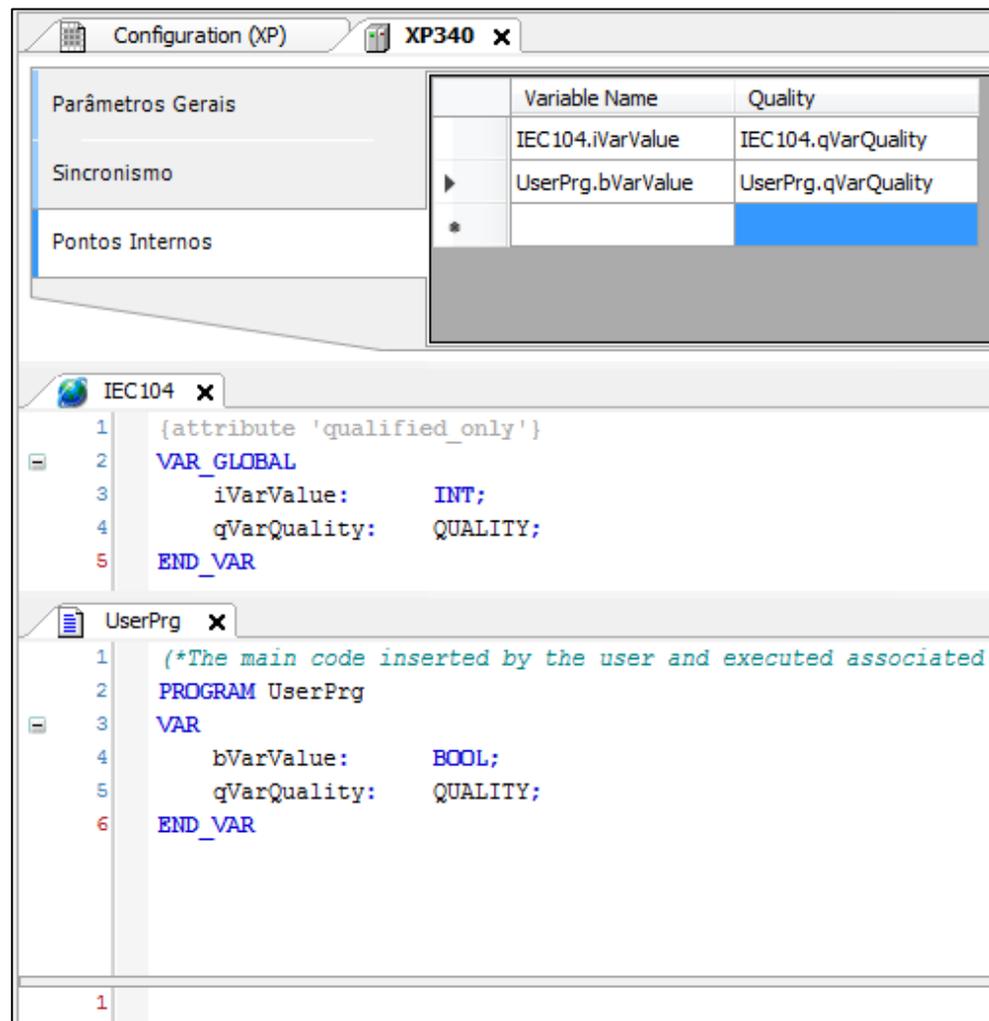


Figura 4-4. Exemplo de Configuração dos Pontos Internos

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

As normas como IEC104 possuem os seus próprios formatos para representação da informação de qualidade de um ponto. A Série Nexto, por sua vez, possui um formato de qualidade próprio (mas muito similar ao IEC 61850) chamado de Qualidade Interna. Este formato é definido pelo tipo QUALITY (biblioteca LibRtuStandard) e é utilizado internamente para armazenamento da

qualidade, permitindo que sejam realizadas conversões entre protocolos sem que haja perda de informação.



Pesquisa na Documentação: configuração da UCP

Consulte as tabelas de configuração da UCP do controlador os seguintes tópicos: parâmetros gerais, configurações SNTP, configuração de pontos internos, estrutura QUALITY, conversão interna para IEC 60870-5-104 de pontos digitais, analógicos e contadores e qualidade MODBUS.

Fonte: Manual de Utilização Nexto Xpress – MU216000

Configuração das Interfaces



Configuração da Interface Serial (COM 1)

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

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

Configuração da Interface Ethernet (NET 1)

A interface NET 1 é composta por um conector de comunicação RJ45 10/100Base-TX padrão. Permite comunicação ponto-a-ponto ou por rede com vários protocolos, por exemplo: MODBUS, OPC UA etc.

As seguintes portas TCP das interfaces Ethernet, locais e remotas, são usadas pelos serviços das UCPs, portanto, são reservadas e não podem ser usadas pelo usuário: 80, 161, 8080, 1217, 1740, 1741, 1742, 1743 e 11740.

Configuração da Interface CAN (Controller Area Network)

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



Pesquisa na Documentação: configuração das interfaces

Consulte nas tabelas de configuração das interfaces, os seguintes tópicos: configuração serial padrão RS-485, configurações avançadas da comunicação serial RS-485, configuração da NET 1 e configuração CAN.

Fonte: Manual de Utilização Nexto Xpress – MU216000

Entradas e Saídas Integradas

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

Existem dois objetos na estrutura do projeto relacionados às E/S Integradas, conforme mostrado na figura a seguir.

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

O outro objeto é o conector Integrated I/O, que contém a configuração para cada tipo de ponto de E/S.

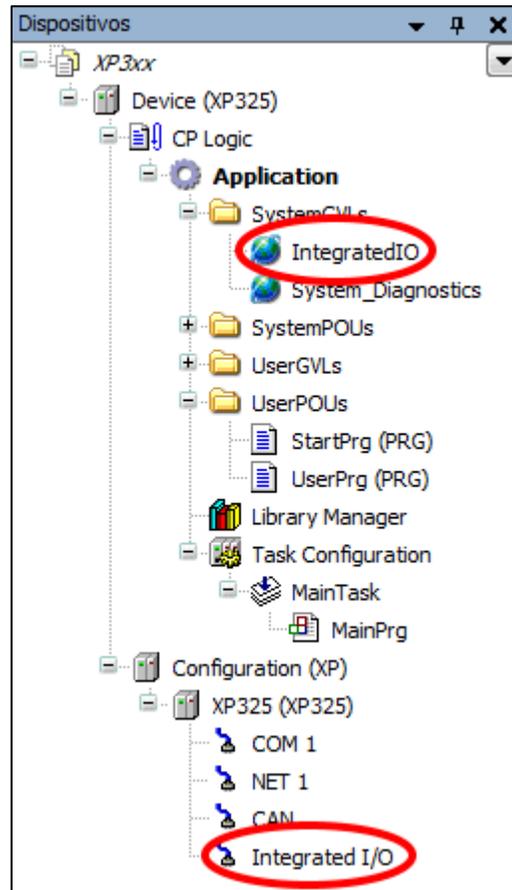


Figura 4-5. E/S Integradas na árvore de projeto

Entradas Digitais

Os parâmetros relacionados às Entradas Digitais estão localizados na tela ilustrada na figura a seguir (exemplo do XP325), tanto para entradas padrão quanto entradas rápidas (quando configuradas como entradas digitais normais).

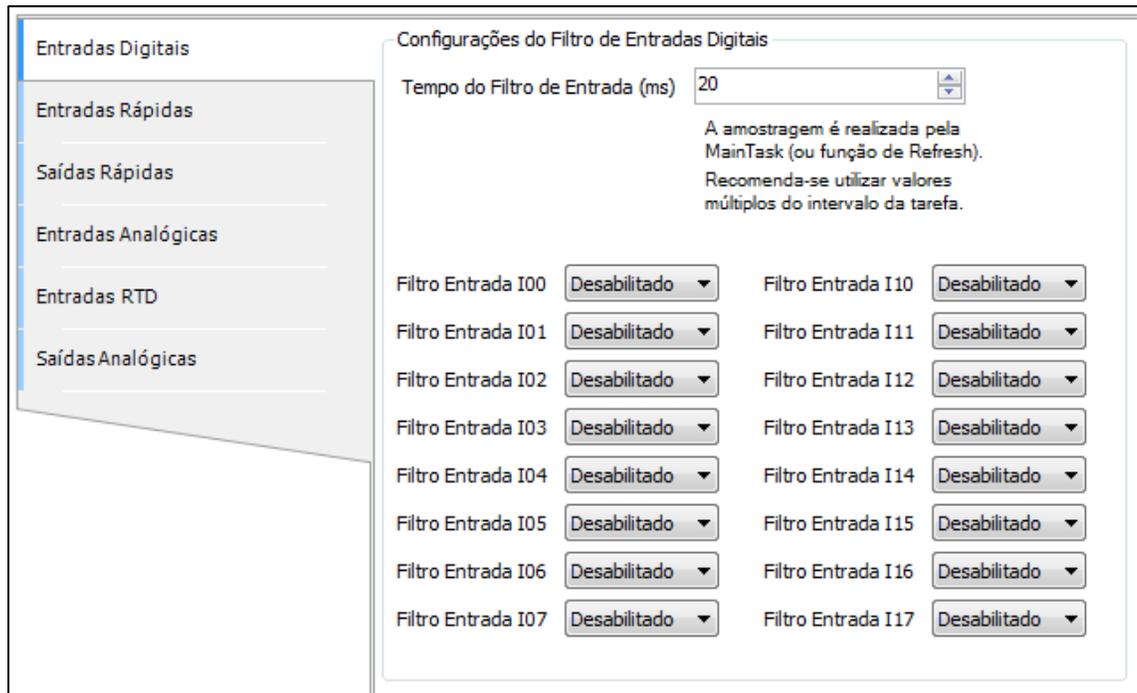


Figura 4-6. Parâmetros de Entradas Digitais

Entradas Rápidas

As entradas rápidas são sinais de entrada especiais que podem ser usados para funções especiais de alta velocidade. Essas entradas físicas especiais podem ser atribuídas a dois tipos de elementos lógicos: contadores rápidos e interrupção externa.

Cada um desses elementos lógicos consome certa quantidade de sinais de entrada rápida. Para a unidade de contadores rápidos, depende do modo selecionado (*Up / Down*, *Quadratura* etc.), enquanto cada interrupção externa usa um sinal de entrada rápida.

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

Tabela 4-1. Alocação de Contadores Rápidos e Entradas

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

Para cada configuração descrita acima, os sinais de entrada rápida restantes (não usados pelas unidades de contador rápido) podem ser usados como interrupção externa, respeitando o número máximo deste tipo de elemento lógico especificado na documentação do produto.

A configuração de contadores rápidos e interrupções está localizada na tela ilustrada na figura a seguir.

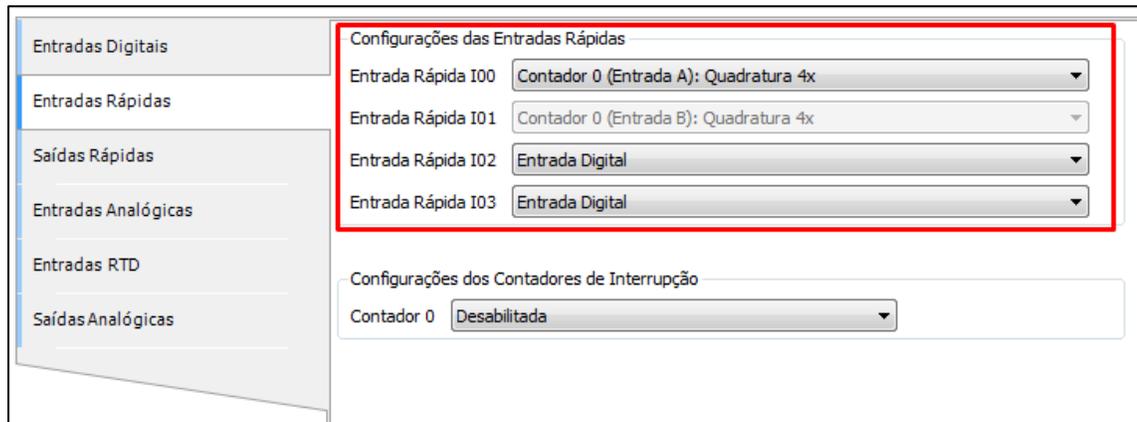


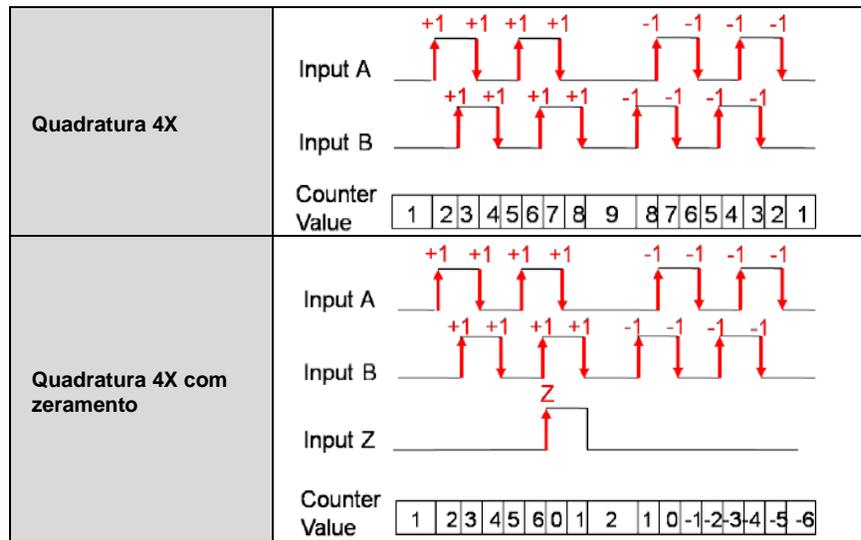
Figura 4-7. Configuração de Entradas Rápidas

Mesmo que uma entrada rápida seja configurada como um contador ou interrupção, seu valor digital ainda pode ser lido por meio da variável `IntegratedIo.DigitalInputs`.

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

Tabela 4-2. Modos dos Contadores Rápidos

Modo do Contador	Formas de Onda de Contagem
Up/Down (A conta, B sentido) com zeramento	<p>Input A: $+1$ $+1$ -1 -1</p> <p>Input B</p> <p>Input Z: Z</p> <p>Counter Value: 31 32 0 1 0 -1</p>
Quadratura 2X	<p>Input A: $+1$ $+1$ $+1$ $+1$ -1 -1 -1 -1</p> <p>Input B</p> <p>Counter Value: 29 30 31 32 33 32 31 30 29</p>
Quadratura 2X com zeramento	<p>Input A: $+1$ $+1$ $+1$ $+1$ -1 -1 -1 -1</p> <p>Input B</p> <p>Input Z: Z</p> <p>Counter Value: 5 6 7 8 0 1 0 -1 -2 -3</p>



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

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

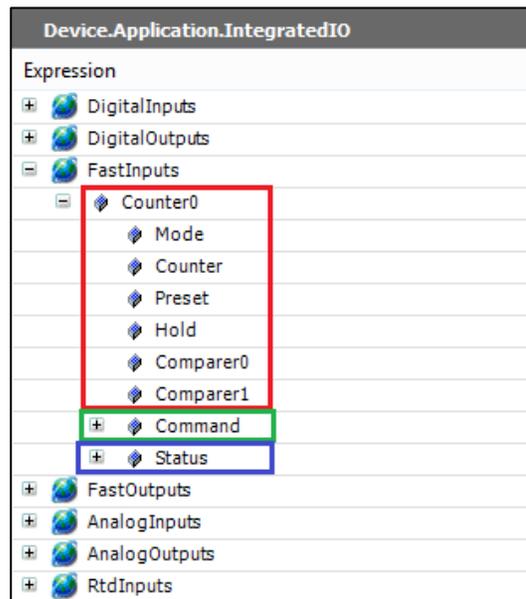


Figura 4-8. Estrutura dos contadores

O comando e o estado são estruturas de bits que permitem ao programa do usuário controlar a operação do contador.

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

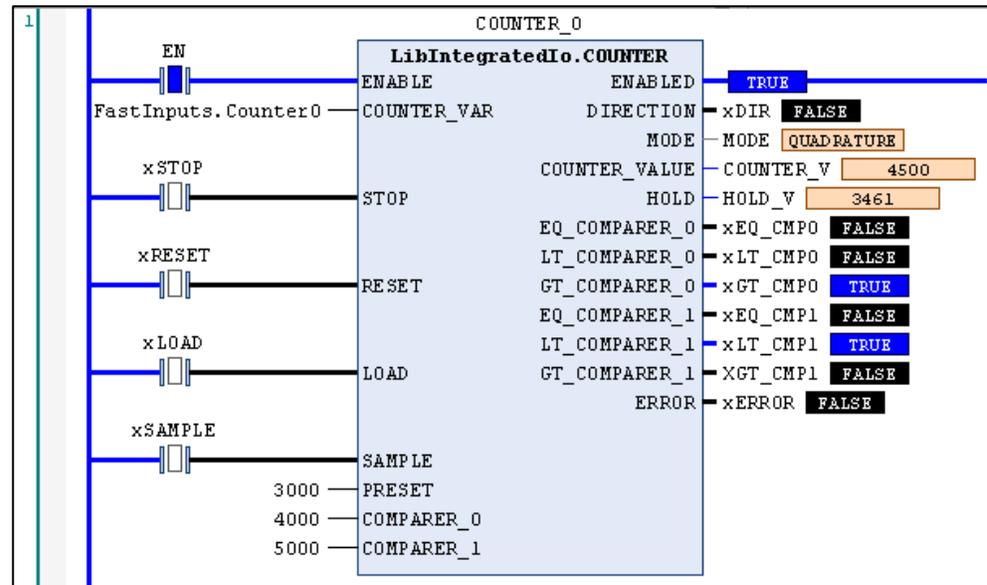


Figura 4-9. Bloco Funcional LibIntegratedIo.COUNTER

Os contadores rápidos têm a capacidade de gerar interrupções por comparação, isto é, quando o contador alcança um determinado valor de comparação, uma tarefa específica será executada e interromperá a execução do programa principal.

Cada unidade de contador rápido possui dois valores de comparação, chamados Comparer0 e Comparer1, que estão presentes na estrutura de dados simbólicos global correspondente ou Bloco Funcional, conforme descrito nas seções anteriores. A configuração da interrupção do contador para cada unidade de contador rápido está localizada na tela ilustrada na figura a seguir.

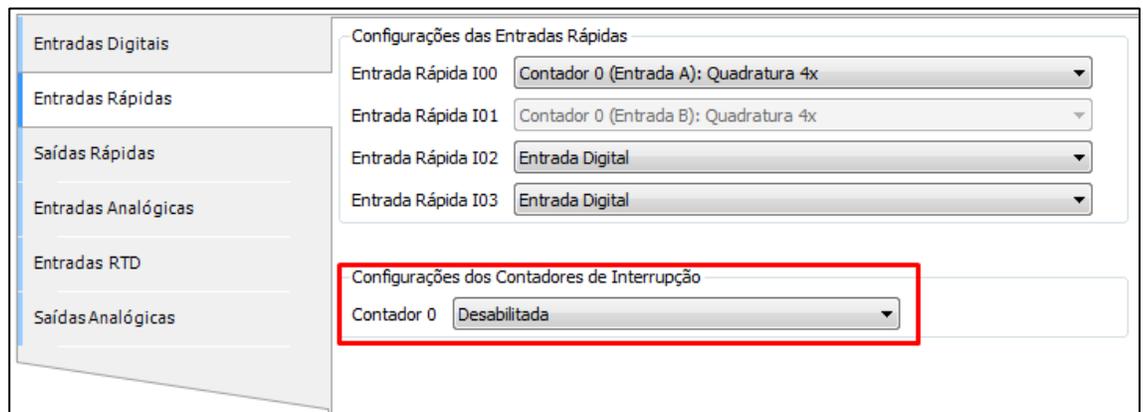


Figura 4-10. Configurações da interrupção do contador

A interrupção por comparação irá gerar um evento específico. Esse evento deve acionar a execução de uma tarefa de evento externo, que deve chamar uma POU específica. Por exemplo, o evento de comparação gerado para o contador 0 é chamado COUNTER0_EVT. Portanto, uma tarefa de evento externo chamada Counter0InterruptTask deve ser configurada para ser acionada por esse evento e deve chamar uma POU chamada Counter0InterruptPrg que conterá o programa do usuário a ser executado. A figura a seguir mostra este cenário de configuração no MasterTool IEC XE.

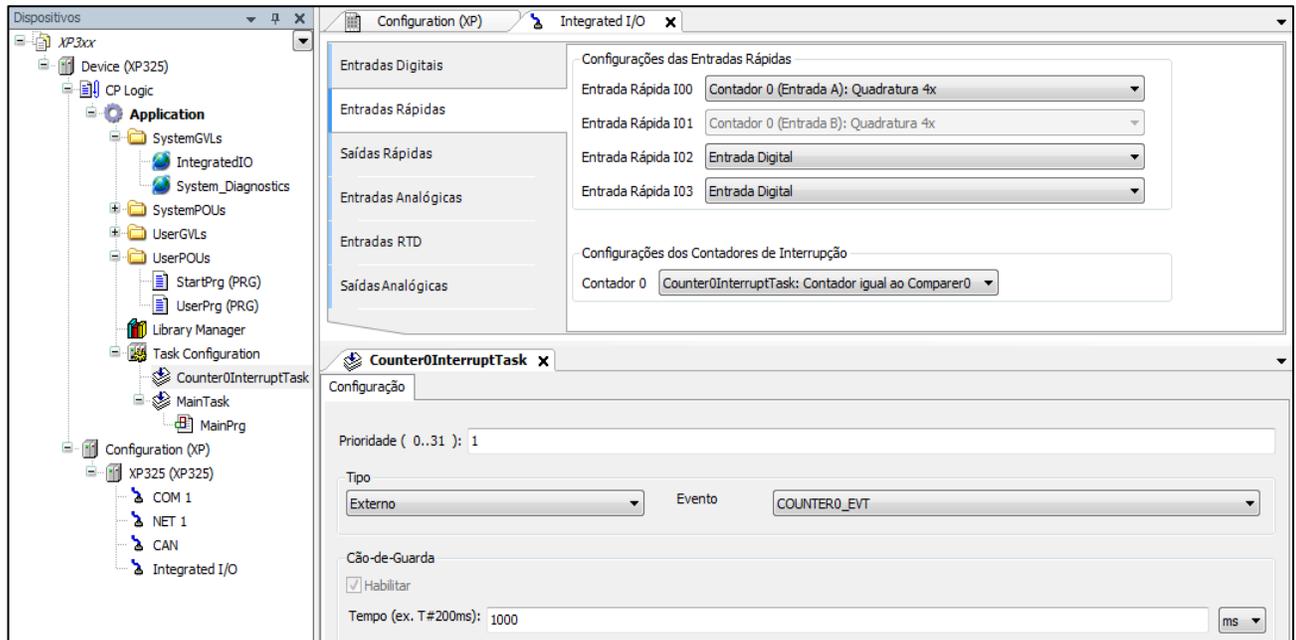


Figura 4-11. Configurações da Interrupção do Contador

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

Cada interrupção externa irá gerar um evento específico. Esse evento deve acionar a execução da tarefa de evento externo, que deve chamar uma POU específica. Por exemplo, o evento de interrupção externa gerado para a entrada rápida I02 é chamado FIN2_EVT. Portanto, uma tarefa de evento externo chamada FastInputI02InterruptTask deve ser configurada para ser acionada por esse evento e deve chamar uma POU chamada FastInputI02InterruptPrg que conterá o programa do usuário a ser executado.

ATENÇÃO:

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

ATENÇÃO:

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

A figura a seguir mostra este cenário de configuração no MasterTool IEC XE.

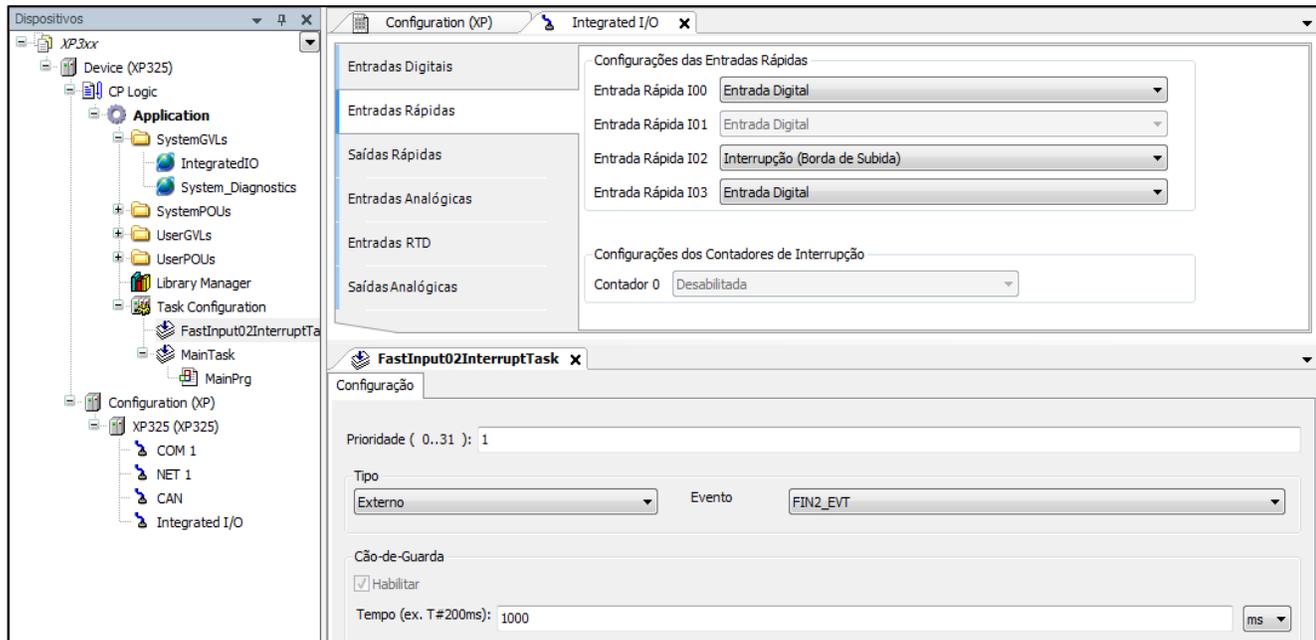


Figura 4-12. Configurações das Interrupções de Entradas Rápidas

Saídas Rápidas

As saídas rápidas são sinais de saída especiais que podem ser usados para geração de pulsos. Essas saídas físicas especiais podem ser associadas a dois tipos de elementos lógicos: VFO / PWM (frequência variável / largura de pulso) e PTO (saída de trem de pulso). Cada um desses elementos lógicos consome um sinal de saída rápida. A configuração de saídas rápidas está localizada na tela ilustrada na figura a seguir:

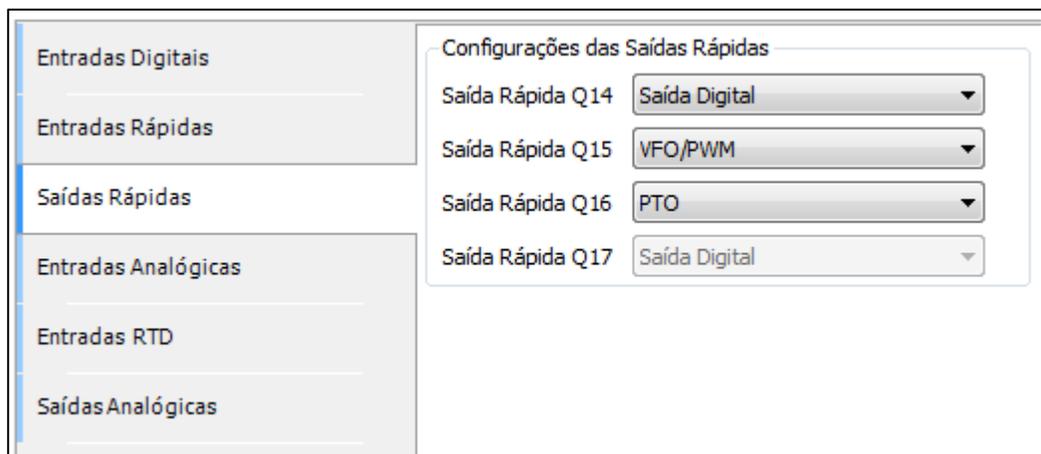


Figura 4-13. Parâmetros das Saídas Rápidas

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

As saídas rápidas podem ser configuradas como saída digital normal. Nesse caso, seu valor digital pode ser definido usando a variável global padrão `IntegratedIo.DigitalOutputs`.

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

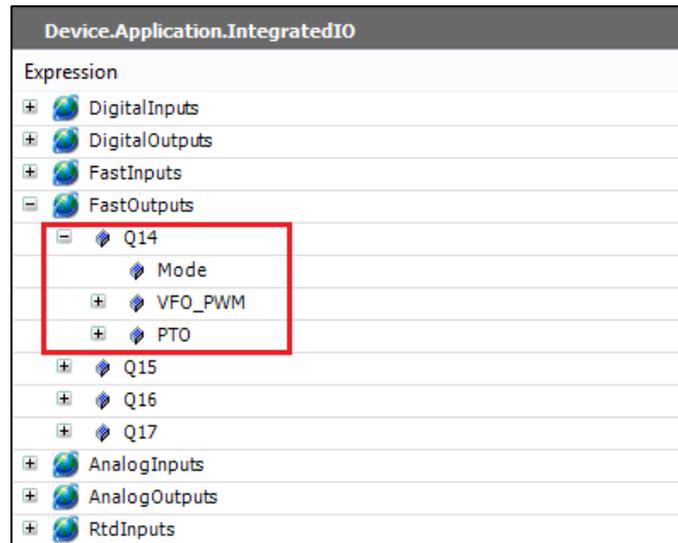


Figura 4-14. Estrutura das Saídas Rápidas

O VFO / PWM (saída de frequência variável / modulador de largura de pulso) é um modo de saída geradora de pulsos onde a frequência e o ciclo de trabalho podem ser controlados pelo programa do usuário. É aplicável, por exemplo, para controlar a energia transferida para uma carga elétrica ou controlar o ângulo de um servo motor. Tomando como base a forma de onda pulsada que é mostrada na figura a seguir, onde T é o período dos pulsos e τ é a largura do pulso. Esses são os parâmetros de pulso que podem ser alterados no modo VFO / PWM. A frequência (f) é definida como o inverso do período ($f = 1/T$) e o ciclo de trabalho (*duty cycle*) é a razão entre a largura do pulso e o período ($D = 100\% * \tau / T$). Para controlar a saída do VFO/PWM, o programa do usuário deve acessar a variável VFO_PWM da estrutura de saída rápida.

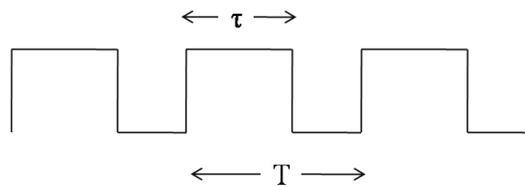


Figura 4-15. Forma de Onda VFO/PWM

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

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

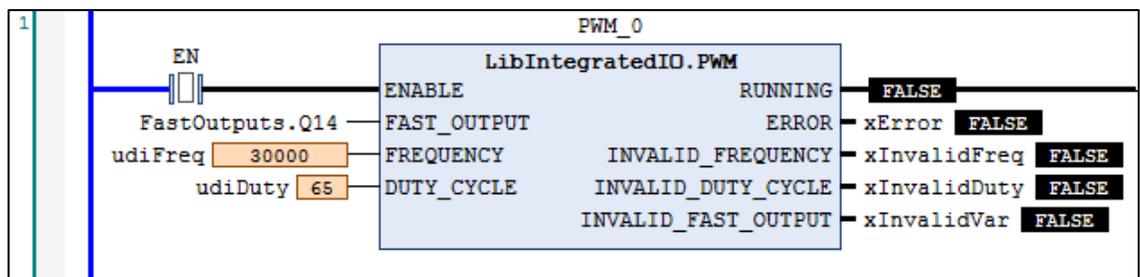


Figura 4-16. Bloco Funcional LibIntegratedIo.PWM

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

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

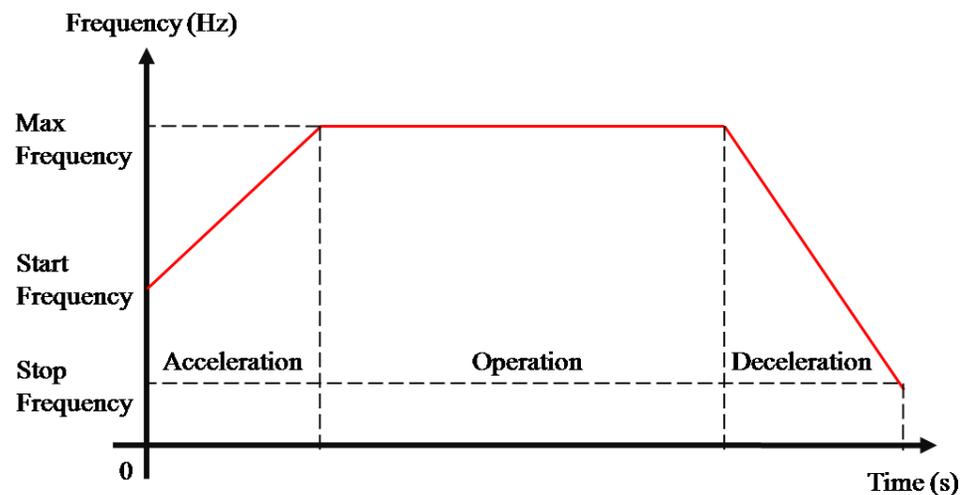


Figura 4-17. PTO com perfil trapezoidal

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

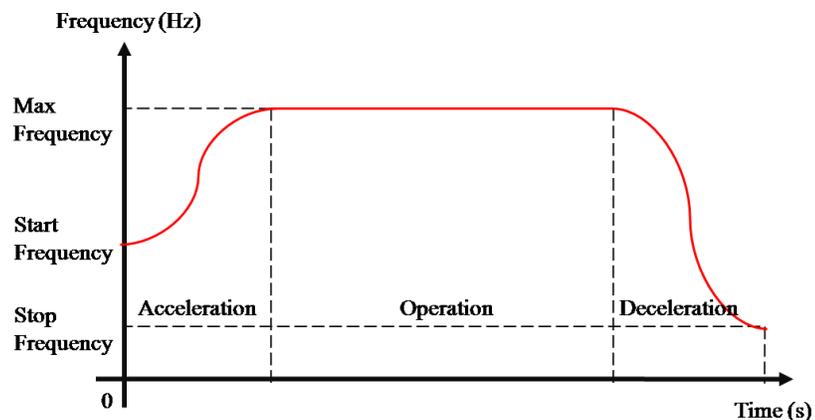


Figura 4-18. PTO com perfil "S"

Além dos parâmetros da PTO, há informações de estados e comandos que o programa do usuário pode usar para controlar a saída. Algumas informações de estados importantes são o contador de pulsos (proporcional a uma posição), a etapa (step) do trem de pulso (aceleração, operação, desaceleração) e, mesmo, se a saída está funcionando bem. Os comandos necessários para controlar a PTO são para iniciar o trem de pulsos (start), parar o trem de pulsos (stop) e parar o trem de pulsos suavemente (softstop). O comando de parada suave é muito importante, uma vez que pode ser usado para emergências em que o sistema não pode parar abruptamente. As figuras

a seguir mostram como o comando de parada suave altera o trem de pulso quando ele é executado. As linhas azuis tracejadas representam a PTO se o comando de parada suave for executado nas etapas de aceleração e operação. O comando de parada suave na etapa de desaceleração não tem efeito, uma vez que o sistema já está parando.

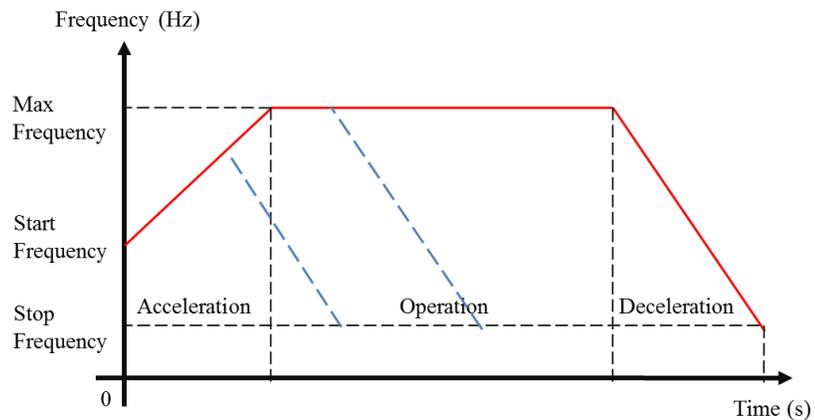


Figura 4-19. Parada suave da PTO no perfil trapezoidal

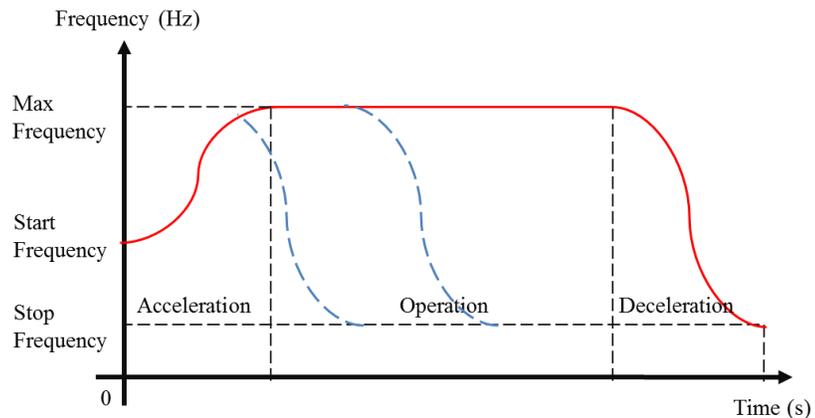


Figura 4-20. Parada suave da PTO no perfil S

Para controlar a PTO, o programa do usuário deve acessar a variável PTO da estrutura de saída rápida.

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

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

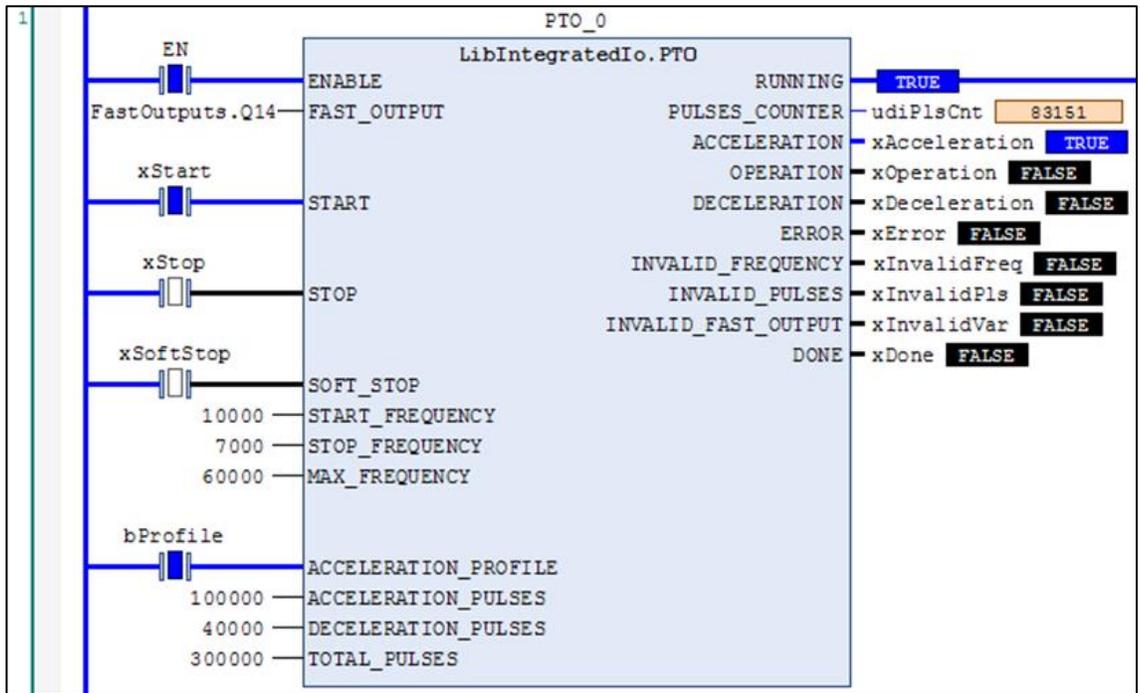


Figura 4-21. Bloco Funcional LibIntegratedIo.PTO

A figura a seguir ilustra uma aplicação típica dentro desse contexto de entradas e saídas rápidas.

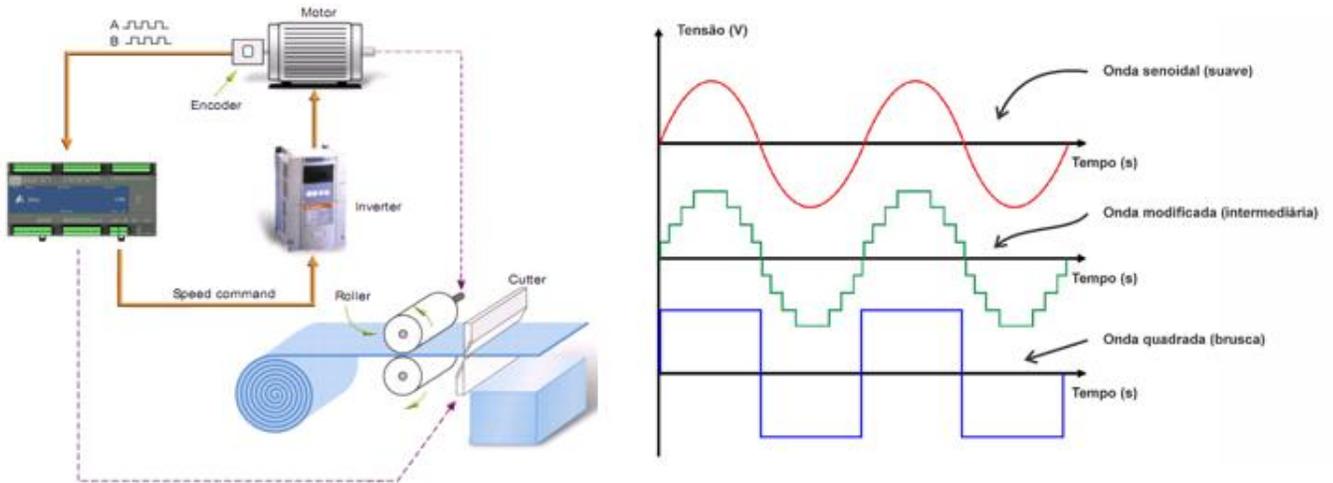


Figura 4-22. Aplicação típica das entradas/saídas rápidas

Entradas Analógicas

Os parâmetros relacionados às Entradas Analógicas são mostrados na figura a seguir.

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

Figura 4-23. Parâmetros das Entradas Analógicas

Entradas RTD

Os parâmetros relacionados às entradas do RTD são mostrados na figura a seguir.

Configurações das Entradas RTD		
Unidade de Temperatura		
	Tipo de Entrada	Filtro Digital
Entrada RI0	Não Configurado	Desabilitado
Entrada RI1	Não Configurado	Desabilitado

Figura 4-24. Parâmetros das entradas RTD

Saídas Analógicas

Os parâmetros relacionados às saídas analógicas são mostrados na figura a seguir.

Configurações das Saídas	
	Tipo de Saída
Saída AO0	Não Configurado
Saída AO1	Não Configurado
Saída AO2	Não Configurado
Saída AO3	Não Configurado

Figura 4-25. Parâmetros das Saídas Analógicas

Mapeamento de E/S

Na aba Mapeamento de E/S é possível configurar nome e descrição para cada variável de entrada e saída do controlador conforme ilustrado na figura a seguir.

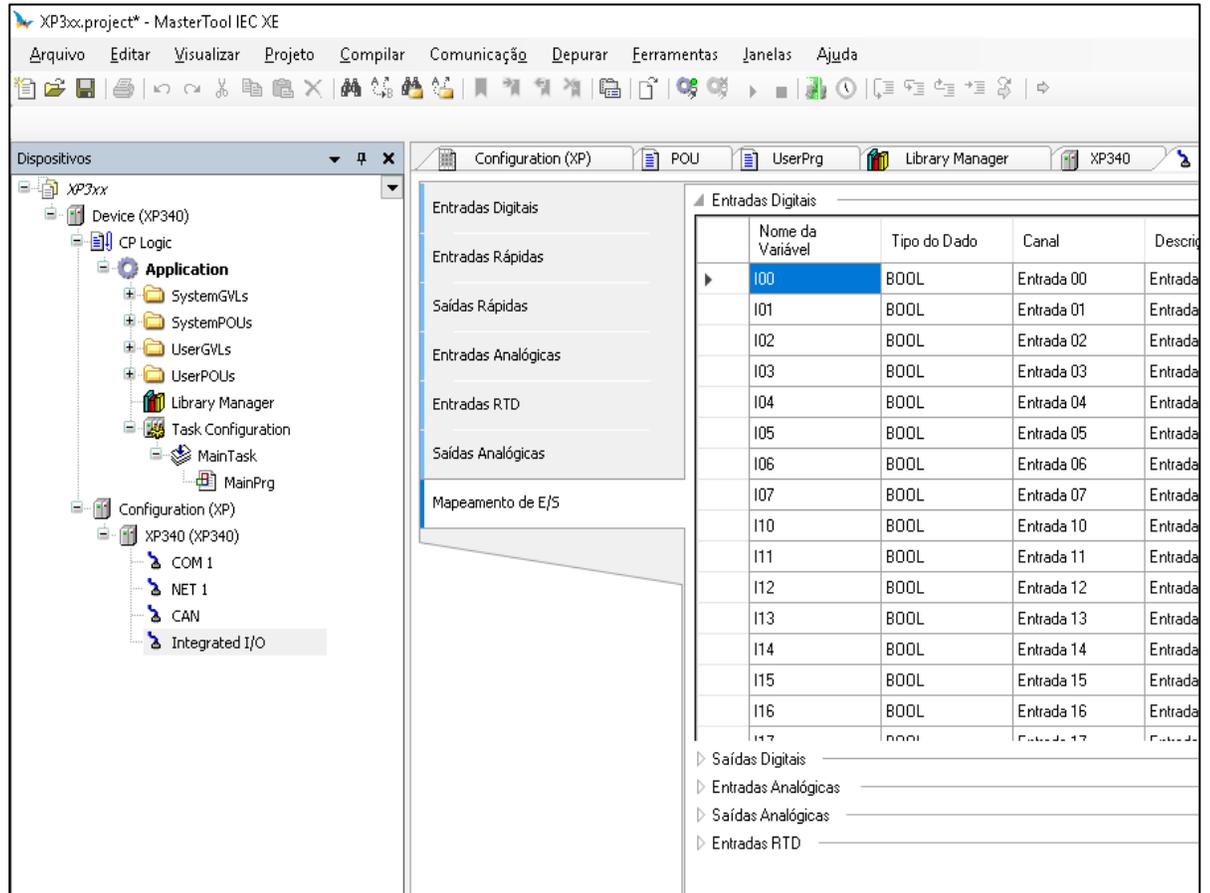


Figura 4-26. Mapeamento de tags para E/S



Pesquisa na Documentação: configuração das E/S integradas

Consulte, na documentação do produto, a configuração das entradas digitais, a alocação de contadores rápidos e suas entradas rápidas, os parâmetros das entradas rápidas, a estrutura de variáveis, comandos e estados do contador rápido, as variáveis de entrada e saída do bloco funcional LibIntegratedIo.COUNTER, os parâmetros da interrupção do contador, os parâmetros das saídas rápidas, a estrutura das variáveis de saída rápida, a estrutura de variáveis, comandos e estados do VFO_PWM, as variáveis de entrada e saída do bloco funcional LibIntegratedIo.PWM, a estrutura de variáveis, comandos e estados do PTO, as variáveis de entrada e saída do bloco funcional LibIntegratedIo.PTO, os parâmetros das entradas analógicas, os parâmetros das entradas RTD e os parâmetros das saídas analógicas.

Fonte: Manual de Utilização Nexto Xpress – MU216000

Porta USB

A porta Host USB presente nos controladores Nexto Xpress permite ampliar as funcionalidades do controlador utilizando diversos tipos de dongles USB. Devido à grande variedade de

dispositivos USB disponíveis no mercado (unidades flash, adaptadores Ethernet /Wifi, modem 3G / 4G etc.), o suporte para cada dispositivo específico é fornecido por uma atualização de firmware.

O gerenciamento de dispositivos USB é feito através de uma seção dedicada localizada na guia Gerenciamento da página web de sistema do controlador, conforme mostrado na figura a seguir.



Figura 4-27. Seção de dispositivos USB

O conteúdo desta página muda dinamicamente de acordo com o tipo de dispositivo USB conectado. No exemplo anterior, não há nenhum dispositivo conectado. Esta página requer login, semelhante à seção de atualização de firmware. O usuário e a senha padrão são "admin", sem as aspas, para os dois campos. As seções a seguir descrevem todos os tipos de dispositivos USB atualmente suportados. Se um dispositivo não suportado estiver conectado, a página informará que o dispositivo é desconhecido conforme ilustrado na figura a seguir.

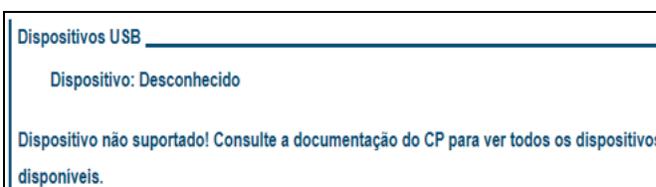


Figura 4-28. Dispositivos USB - Desconhecido

Dispositivos de Armazenamento em Massa Gerais

Os dispositivos de armazenamento em massa podem ser usados para expandir a memória flash do controlador para armazenar uma grande quantidade de dados, como em aplicações de registrador de dados, por exemplo. Para usar um dispositivo de armazenamento em massa USB, basta conectá-lo à porta USB. Após alguns segundos, quando o dispositivo for detectado e montado corretamente, o LED USB será ligado e as informações do dispositivo aparecerão na seção Dispositivos USB localizada na guia Gerenciamento da página web de diagnóstico do controlador, conforme mostrado na figura a seguir.



Figura 4-29. Informações sobre o dispositivo de armazenamento em massa

As informações mostradas na seção de Status desta página também estão disponíveis na estrutura de diagnóstico de variáveis simbólicas.

ATENÇÃO:

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

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

Depois que o dispositivo for detectado e montado corretamente, uma nova pasta chamada Mass_Storage aparecerá na memória do controlador, conforme mostrado na figura a seguir.

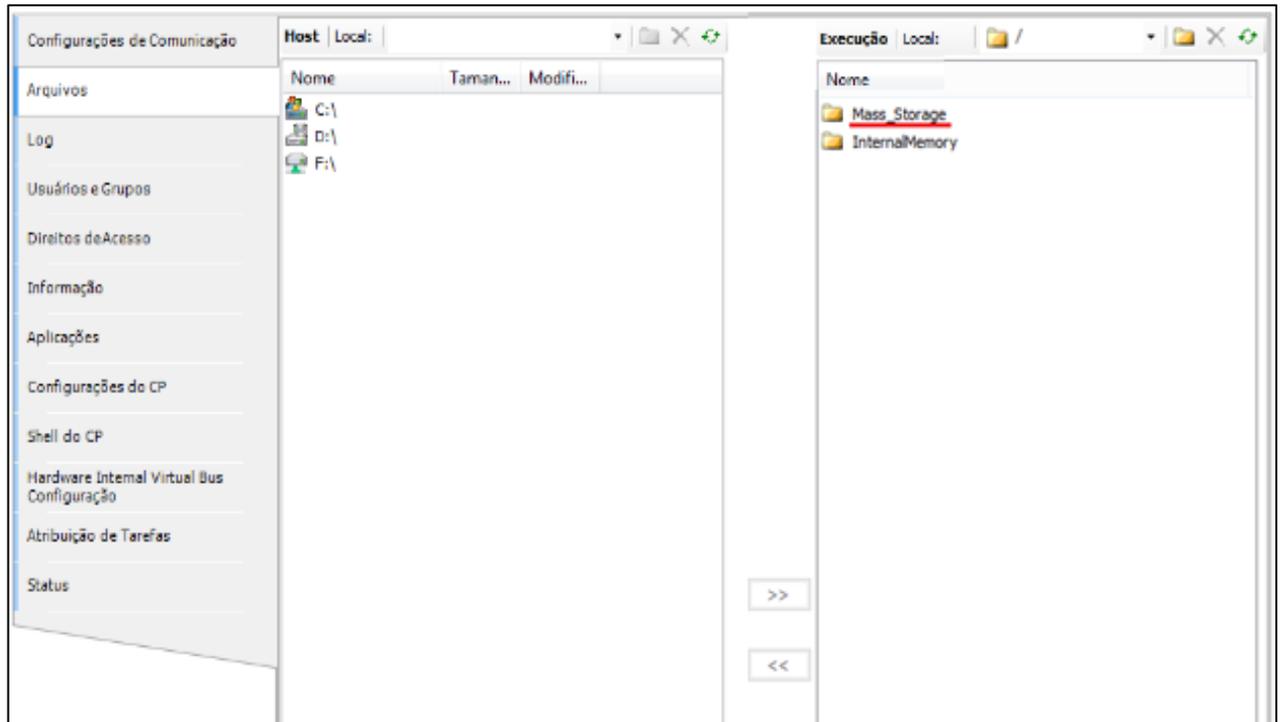


Figura 4-30. Pasta de armazenamento em massa USB

O dispositivo de armazenamento em massa USB pode ser usado para impedir que o controlador carregue automaticamente a aplicação após a energização. Para fazer isso, basta colocar um arquivo de texto vazio chamado dontbootapp.txt na pasta raiz do dispositivo de armazenamento em massa. A presença desse arquivo é informada no campo Arquivos Especiais na página web de sistema do controlador, conforme mostrado na figura a seguir.



Figura 4-31. DontBootApp no dispositivo de armazenamento em massa

O dispositivo de armazenamento em massa USB também pode ser usado para transferir uma aplicação para o controlador. Para fazer isso, coloque os dois arquivos Application.app e Application.crc na pasta raiz do dispositivo de armazenamento em massa (esses arquivos são criados usando o MasterTool IEC XE executando o comando Comunicação -> Criar Aplicação de

Inicialização quando off-line). Após a energização, se o controlador detectar a presença desses arquivos no dispositivo de armazenamento em massa USB, a seguinte sequência de ações ocorrerá:

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

A presença da aplicação é informada no campo Arquivos Especiais na página web de sistema do controlador, conforme mostrado a seguir.

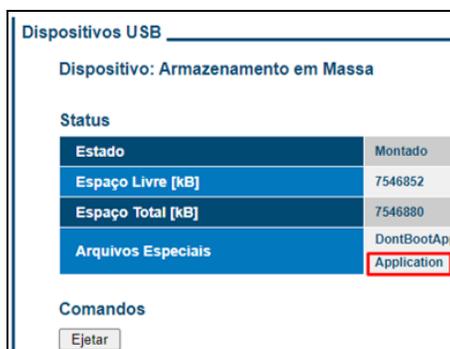


Figura 4-32. Aplicação no dispositivo de armazenamento em massa

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

Conversor USB para RS-232

O Nexto Xpress permite implementar uma porta RS-232 usando um conversor USB para Serial. Esses conversores são todos baseados em um chip controlador interno.

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

Depois de conectar o conversor à porta USB, o LED USB pode acender indicando que o dispositivo foi detectado e montado corretamente e as informações do dispositivo aparecerão na seção Dispositivos USB, localizada na guia Gerenciamento do CP da página web de diagnósticos do controlador como mostrado na figura a seguir.



Dispositivos USB	
Dispositivo: Conversor Serial	
Status	
Bytes Recebidos	0
Bytes Transmitidos	0
Bytes Restantes no Buffer RX	0
Bytes Restantes no Buffer TX	0
Número de Erros de Break	0
Número de Erros de Paridade	0
Número de Erros de Frame	0
Número de Erros de RX Overrun	0

Figura 4-33. Dispositivos USB - Conversor Serial

As informações mostradas nesta página também estão disponíveis na estrutura de diagnóstico de variáveis simbólicas.

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

Dispositivos Modem

Um modem USB com chip SIM pode ser usado para conectar o CP à Internet usando a rede de dados celular (serviços telefônicos, como o envio de SMS, não são implementados). Esse recurso permite usar os controladores Nexto Xpress para implementar aplicações de telemetria e IoT. Existem basicamente dois tipos de modems: bridge e roteador.

O modem bridge é um dispositivo não gerenciado que implementa uma conexão direta (passagem) à rede de dados móveis, para que todas as solicitações de conexão provenientes da Internet cheguem ao sistema operacional do controlador. Para esse tipo de dispositivo, a configuração é realizada na página web do sistema do controlador, conforme descrito mais adiante nesta seção.

Por outro lado, o modem roteador é um dispositivo gerenciado que implementa um firewall com políticas de rede configuráveis. Para esse tipo de dispositivo, toda a configuração é realizada por meio de uma página da web proprietária incorporada ao dispositivo. Por padrão, o modem bloqueia qualquer tipo de conexão de entrada (ou seja, não permitirá o acesso remoto). Para permitir isso, o usuário deve configurar (através da página Web incorporada ao dispositivo) uma regra de encaminhamento de porta para a porta TCP relacionada ao serviço desejado.

O gerenciamento da funcionalidade do modem USB é feito através da página Dispositivos USB na guia Gerenciamento do CP da página web de sistema do controlador. Depois que o dispositivo Modem for detectado e montado corretamente, o LED USB acenderá e as informações do dispositivo aparecerão como mostrado na figura a seguir.

Dispositivos USB

Dispositivo: Modem

Status

Configuração	Não Configurado
Conexão	Desconectado
Endereço IP	-

Configuração

APN do Provedor:

Código PIN:

Buttons: Importar, Exportar, Redefinir, Aplicar

Figura 4-34. Configuração da Página do Modem USB

Esta área contém basicamente duas seções: Status e Configuração.

A seção Status é onde todos os diagnósticos relacionados ao modem USB são exibidos: estado da configuração (que não depende do estado do dispositivo), estado da conexão com a Internet e o endereço IP do modem. Esses campos são atualizados automaticamente sempre que um valor é atualizado (não é necessário recarregar a página manualmente). As mesmas informações também são fornecidas na estrutura de diagnóstico de variáveis simbólicas, que também contém a descrição detalhada dos possíveis valores para cada campo.

A seção Configuração é onde o usuário executa a configuração do modem. Para modems em bridge, esta seção é usada para inserir as informações do chip SIM. Para modems roteadores, esta seção mostrará apenas um botão chamado Abrir Página do Modem que será redirecionado para a página web interna do modem (verifique na lista de dispositivos compatíveis se o redirecionamento é suportado; caso contrário, o modem deve ser configurado externamente em um PC). Esta seção contém os seguintes botões:

- **Importar:** carrega a configuração de um arquivo externo;
- **Exportar:** salva a configuração atual em um arquivo externo;
- **Redefinir:** apaga a configuração do modem da memória do controlador;
- **Aplicar:** escreve a configuração na memória do controlador.

Os campos APN do provedor e código PIN são obrigatórios para cada chip SIM. Se o fornecedor informar esses parâmetros, eles deverão ser utilizados. Por outro lado, sabe-se que vários chips SIM simplesmente não se importam com o conteúdo desses campos, usando valores predefinidos internos. Nesse caso, esses campos da página da web podem ser deixados com os valores padrão e a conexão prosseguirá com sucesso. Valores como "zero" para o código PIN e "vazio" para o APN do provedor não são permitidos.

O princípio de operação da funcionalidade do modem USB é bastante simples. Para dispositivos bridge, depois que o dispositivo é detectado corretamente e os parâmetros do SIM configurados, o controlador inicia um processo em segundo plano, que controla continuamente o modem para mantê-lo conectado à Internet. Isso elimina a necessidade de qualquer tipo de intervenção manual. Portanto, se a conexão for interrompida por algum motivo (má recepção, interrupção da operadora etc.), o controlador tentará se reconectar automaticamente. O status desse processo pode ser observado no campo Status da conexão. Para modems roteadores, o dispositivo contém um processo semelhante executado internamente (independente do controlador) chamado Auto Connect. Geralmente é ativado por padrão, mas pode ser desativado na página web do modem.

Um aspecto importante a considerar é que, se o modem USB estiver configurado, o sistema do controlador o configurará como o gateway padrão para toda a comunicação Ethernet. Isso significa que, se o controlador estiver conectado simultaneamente a uma rede local (NET1), que também tem acesso à Internet, todas as mensagens Ethernet endereçadas à IPs externos serão roteadas pelo modem USB (e não pela NET1). A NET1 retorna como o gateway padrão quando o modem USB tiver sido removido.

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

1. Conecte o dispositivo na porta USB. Após alguns segundos, o campo Dispositivo irá mostrar "Modem". Caso contrário, o dispositivo pode não ter suporte ou estar com defeito.
2. (Para dispositivos bridge) Defina os parâmetros do SIM (APN do provedor - Nome do Ponto de Acesso e o código PIN do chip) conforme informado pelo fornecedor do chip SIM. Após clicar no botão Aplicar, o processo de conexão será iniciado. Depois de configurados, não é mais necessário definir essas informações. Elas serão salvas na memória do controlador.
3. Verifique o status da conexão no campo correspondente. Se tudo correr bem, após alguns segundos, ele informará que o modem está conectado à Internet e um endereço IP aparecerá em seu campo.

Uma vez conectado à Internet, o controlador pode ser usado para vários tipos de aplicações. Um caso de uso típico é implementar solução de telemetria usando o Bloco Funcional do Cliente MQTT para publicar dados. Outro caso de uso é acessar o controlador remotamente. Nesse caso, é necessário conhecer o endereço IP do modem na internet. No entanto, o IP é dinâmico e será alterado a cada processo de conexão. Uma maneira de solucionar esse problema é publicar o endereço IP (disponível nas variáveis de diagnóstico do modem) por meio do MQTT.

Com o endereço IP do modem, é possível executar acesso remoto à página da web do sistema do controlador para exibir o status e os diagnósticos (a atualização de firmware não é suportada). Além disso, permite programar o controlador remotamente usando o MasterTool IEC XE. Para isso, o gateway deve ser configurado com o endereço IP do modem, como mostrado na figura a seguir.

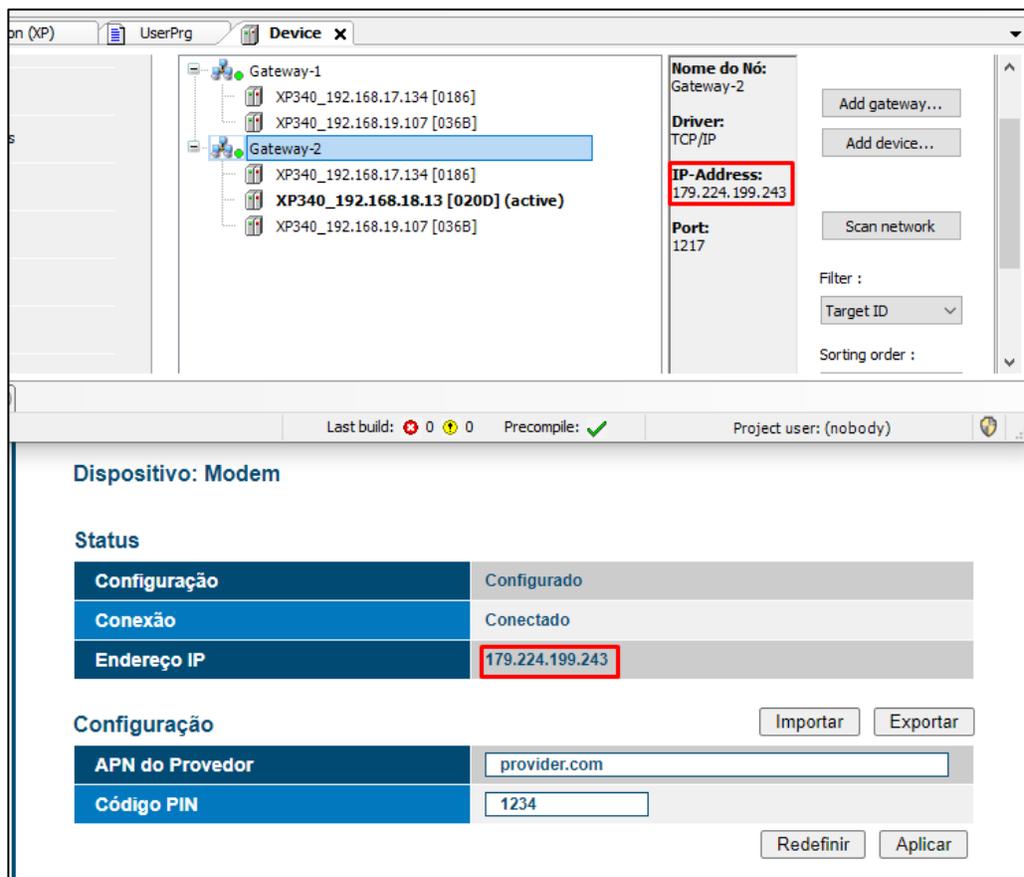


Figura 4-35. Configurando o Gateway com Endereço IP do Modem

Para dispositivos em bridge ou roteadores com acesso externo ativado (encaminhamento de porta), uma vez conectado à Internet, qualquer pessoa que conheça o endereço IP do modem poderá acessar o controlador remotamente. Portanto, por motivos de segurança, é EXTREMAMENTE importante e recomendado configurar os Direitos do Usuário no controlador para restringir as operações online do MasterTool IEC XE com login e senha.

Adaptadores WiFi

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

O gerenciamento da funcionalidade do adaptador WiFi é feito através da página Dispositivos USB na guia Gerenciamento do CP da página web do sistema do controlador. Depois que o dispositivo adaptador WiFi for detectado e montado corretamente, o LED USB acenderá e as informações do dispositivo aparecerão como mostrado na figura a seguir.

Dispositivos USB	
Dispositivo: Adaptador WiFi	
Status	
Configuração	Não Configurado
Conexão	Desconectado
Endereço IP	-
Máscara de Rede	-
Gateway	-
Endereço MAC	E8:94:F6:26:CA:EA
Configuração	
Modo de Operação	Dispositivo Cliente
Modo de Descobrimto	Auto
SSID da Rede	<input type="text"/> Escanear
Senha	<input type="password"/>

Figura 4-36. Página do adaptador USB WiFi

Esta página contém basicamente duas seções: Status e Configuração.

A seção Status mostra todos os diagnósticos relacionados ao adaptador WiFi: estado de configuração, estado da conexão, endereço IP, máscara de rede, gateway e endereço MAC. Esses campos são atualizados assim que os valores mudam. Essas informações também são fornecidas na estrutura de diagnóstico de variáveis simbólicas.

A seção Configuração é composta pelos seguintes parâmetros:

- **Modo de Operação:** define como o adaptador WiFi funcionará (atualmente, apenas o modo cliente é suportado).
- **Modo de Descobrimto:** define qual é o método para definir a rede WiFi. Se selecionado como "Auto", o botão "Escanear" deve ser usado para escolher a rede desejada. Se selecionado como "Manual", o nome do SSID e do Tipo de segurança deve ser digitado manualmente.
- **SSID da Rede:** Quando o modo de descobrimto estiver definido como "Auto", esse campo mostrará as redes disponíveis encontradas no processo de verificação, classificadas do melhor ao pior nível de sinal (de cima para baixo). Quando o modo de descobrimto está definido como "Manual", este campo deve ser preenchido com o SSID da rede desejada.
- **Tipo de Segurança:** Esse campo estará disponível apenas quando o modo de descobrimto estiver definido como "Manual" (o modo "Auto" seleciona

automaticamente o tipo de segurança fornecido pela rede escaneada). Este campo define o tipo de segurança usada na rede Wi-Fi, que pode ser "Pública" ou "WPA2-Personal".

- **Senha:** é aqui que você precisa digitar a senha da rede WiFi. O campo será bloqueado automaticamente se o Tipo de segurança estiver definido como "Pública" ou se a rede escaneada escolhida não usar um protocolo de segurança.
- **Definição de IP:** define se o adaptador WiFi definirá o endereço IP dinamicamente (atribuído pelo servidor DHCP da rede) ou estaticamente (onde o usuário precisará inserir as configurações de IP manualmente).
- **Endereço IP, Máscara de Rede e Gateway:** disponível apenas quando a definição de IP é definida como "Estático". Esses campos serão usados para configurar os parâmetros de rede do adaptador WiFi.
- **Gateway Padrão:** esse campo define qual interface de rede será usada como gateway para acessar a Internet. É possível escolher o "Adaptador WiFi" ou a "Ethernet Local" para esta função.

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

Figura 4-37. Configuração do adaptador USB WiFi

Além disso, a seção Configuração contém os seguintes botões:

- **Importar:** carrega um arquivo de configuração;
- **Exportar:** baixa um arquivo com a configuração atual;
- **Redefinir:** retorna à configuração padrão;
- **Aplicar:** aplica a configuração atual.

O princípio de operação do adaptador USB WiFi é semelhante ao modem USB. Depois que o dispositivo for detectado corretamente e os parâmetros de rede estiverem configurados, o controlador sempre tentará manter-se conectado à rede Wi-Fi. O status desse processo pode ser observado no campo Status da Conexão.

Se o Gateway Padrão foi definido como "Adaptador WiFi", o Gateway no diagnóstico da NET1 do MasterTool mostrará zero (0.0.0.0). Caso contrário, se foi definido como "Ethernet Local", o gateway do adaptadorWiFi será zero.

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

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

A imagem a seguir mostra a página de um controlador conectado a uma rede WiFi.

The screenshot shows a web interface titled "Dispositivos USB". It contains two main sections: a status table and a configuration form.

Dispositivos USB	
Configuração	Configurado
Conexão	Conectado
Endereço IP	192.168.13.175
Máscara de Rede	255.255.255.0
Gateway	192.168.13.253
Endereço MAC	E8:94:F6:26:CA:EA

Configuração Importar Exportar

Modo de Operação	Dispositivo Cliente ▾
Modo de Descobrimto	Auto ▾
SSID da Rede	Altus ▾ Escanear
Senha	<input type="text"/>
Definição de IP	Dinâmico (DHCP) ▾
Gateway Padrão	Adaptador WIF ▾

Redefinir Aplicar

Figura 4-38. Adaptador USB WiFi conectado a uma rede

Uma vez conectado à rede WiFi, esse canal de comunicação pode ser usado para vários propósitos. Para programar o CP com o MasterTool IEC XE, o gateway deve ser configurado com o endereço IP atribuído ao adaptador WiFi (semelhante ao modem USB). Esse endereço IP também pode ser usado para acessar a página da web do sistema do controlador, onde é possível executar uma atualização de firmware, que não está disponível ao usar o modem USB. Além disso, esse canal de comunicação também pode ser usado com o bloco funcional do cliente MQTT para relatar dados para um broker externo na Internet (nesse caso, o adaptador WiFi como Gateway Padrão).

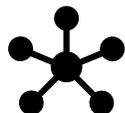


Pesquisa na Documentação: configuração da porta USB

Consulte, na documentação do produto, os conversores USB para RS-232 suportados e as limitações da solução, os dispositivos MODEM USB suportados, os números das portas TCP para os principais serviços do CP e os chipsets suportados para adaptadores USB WiFi.

Fonte: Manual de Utilização Nexto Xpress – MU216000

Protocolos de Comunicação



Os controladores Nexto Xpress oferecem diversos protocolos de comunicação, incluindo MODBUS (exclusivamente simbólico), OPC UA e outros.



NOTAS:

1. **Pontos Mapeados:** Cada variável ou item de um determinado tipo de dado é considerado um mapeamento. Isso é considerado para cada posição do tipo ARRAY. Isso significa que, se uma variável simples for declarada, ela será considerada um mapeamento e, se um tipo ARRAY for declarado, a contagem será igual ao tamanho do ARRAY declarado. A quantidade de mapeamentos é incrementada em um quando há um tipo simples de variável sendo declarado independentemente do tamanho do tipo dado. Então, o mapeamento de uma variável do tipo INT (16-bit) em um Holding Register de drivers MODBUS simbólicos ou uma variável do tipo LINT (64-bit) em quatro Holding Register de drivers simbólicos MODBUS é contabilizado como apenas um mapeamento.
2. **Mapeamentos:** Um mapeamento é um relacionamento entre uma variável interna da aplicação e um objeto de protocolo da aplicação. O valor limite para os mapeamentos do projeto corresponde à soma de todos os mapeamentos feitos dentro das instâncias dos protocolos de comunicação e seus respectivos dispositivos.
3. **Requisições:** A soma das requisições dos protocolos de comunicação, declaradas nos dispositivos, não pode ultrapassar a quantidade máxima de requisições suportadas pela UCP.
4. **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 à soma de todos os dispositivos clientes das instâncias de protocolos de comunicação Servidor IEC 60870-5-104 (não inclui os mestres e clientes dos protocolos MODBUS RTU Escravo ou MODBUS Servidor).

MODBUS RTU

Inserção de uma instância do protocolo MODBUS RTU

O primeiro passo para configurar o MODBUS RTU é incluir a instância na COM desejada (nesse caso, COM 1). Clicar com o botão direito sobre a COM e selecionar Acrescentar Dispositivo..., conforme mostra a figura a seguir.

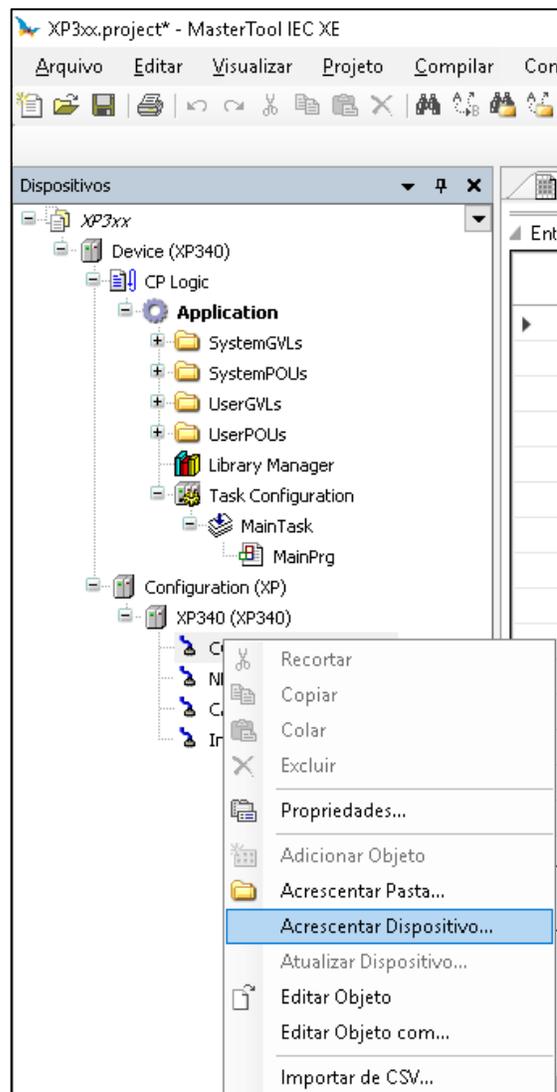


Figura 4-39. Adicionando a Instância MODBUS

Após, surgirão na tela os protocolos disponíveis ao usuário. Nesse caso, pode-se selecionar a opção MODBUS Symbol RTU Master, para configuração por Mapeamento Simbólico. A seguir, clicar em Acrescentar, conforme mostrado na sequência.

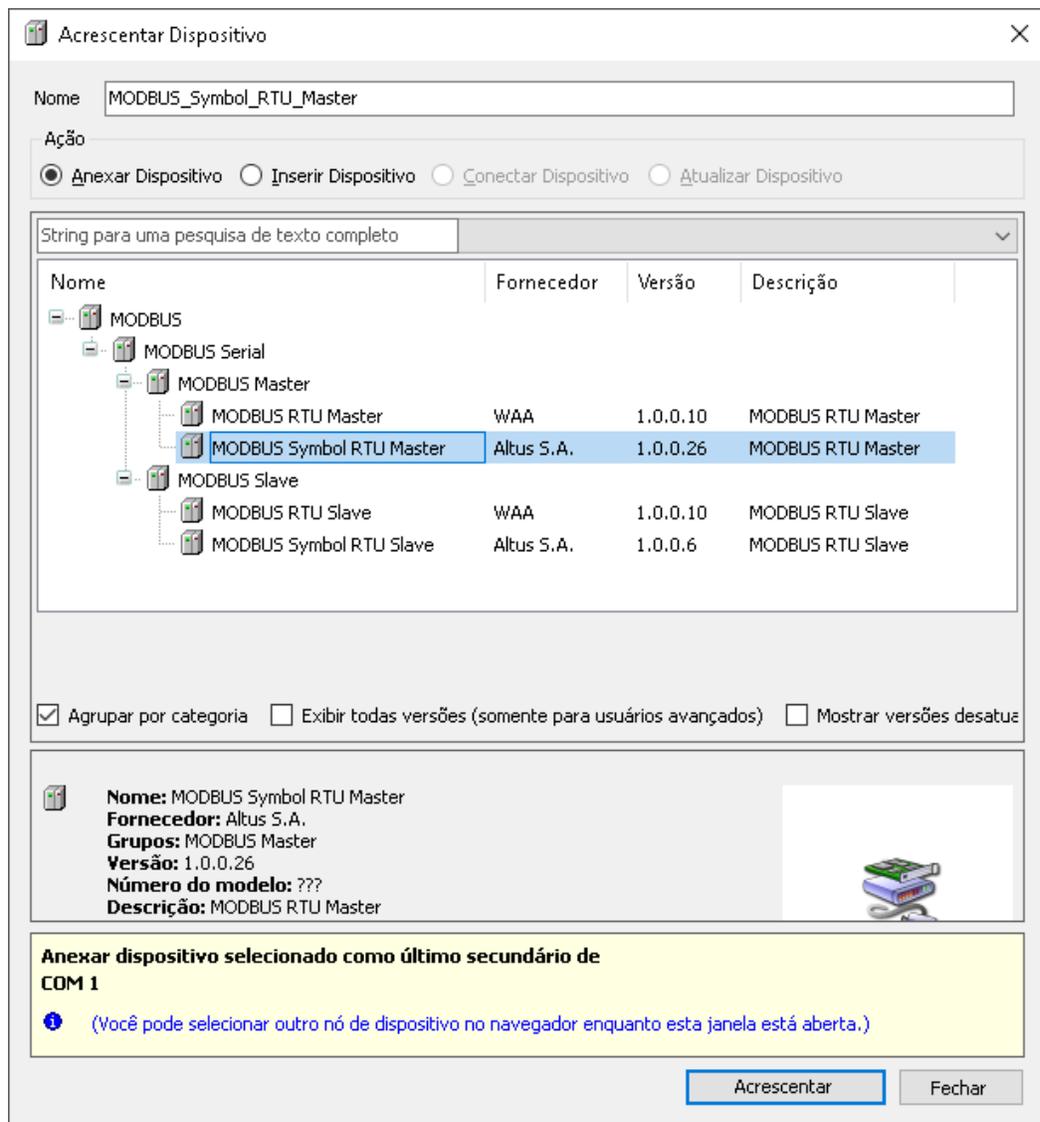


Figura 4-40. Seleção do protocolo MODBUS RTU Master

Na sequência, deve-se 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.

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

1. Configurar os parâmetros gerais do protocolo MODBUS Mestre, como: tempos de atraso de envio e interframe mínimo conforme mostrado na figura a seguir.

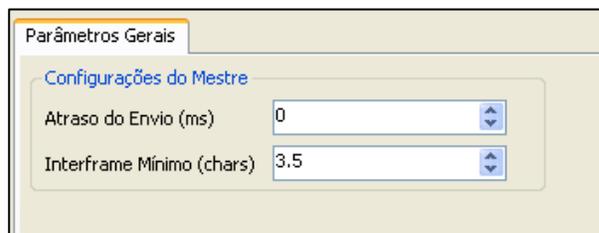


Figura 4-41. Tela de Configuração Parâmetros Gerais MODBUS RTU Mestre

- Adicionar e configurar dispositivos através da aba Parâmetros Gerais, definindo endereço do escravo, time-out de comunicação e número de tentativas de comunicação como pode ser visto na figura a seguir.

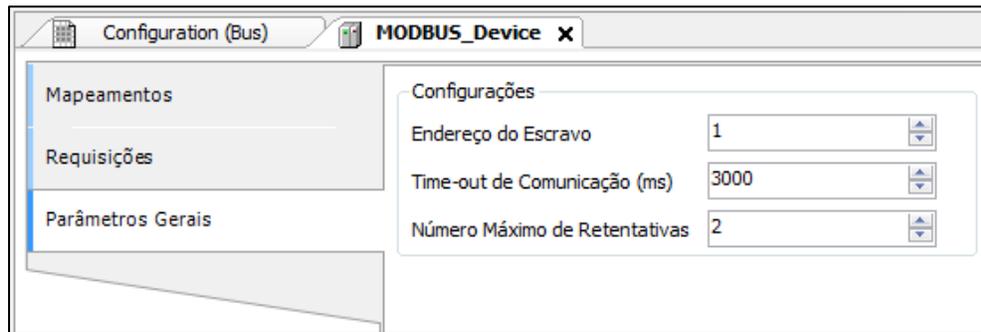


Figura 4-42. Tela de Configurações dos Parâmetros Gerais do Dispositivo

- Adicionar e configurar os mapeamentos MODBUS na aba Mapeamentos conforme figura a seguir, especificando o nome da variável, tipo de dados e endereço inicial do dado, o tamanho do dado e a faixa são preenchidos automaticamente.

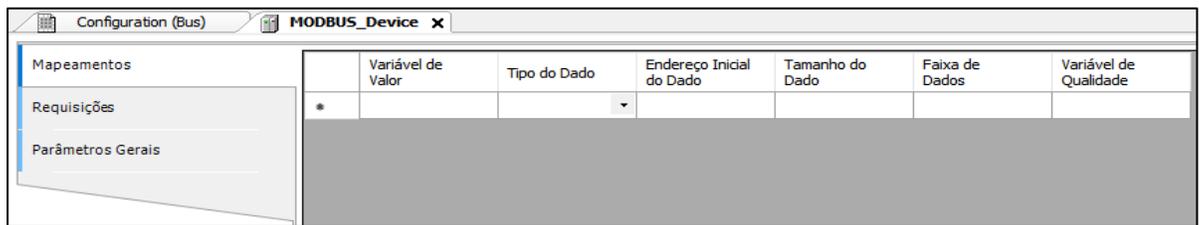


Figura 4-43. Tela de Mapeamentos de Dados MODBUS

- Adicionar e configurar as requisições MODBUS como apresentado na figura a seguir, 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.

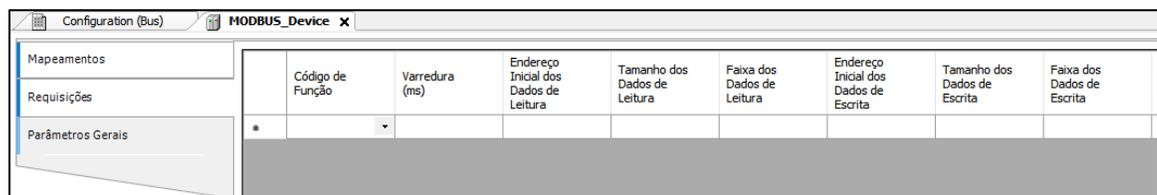


Figura 4-44. Tela de Requisições de dados MODBUS Mestre



Pesquisa na Documentação: editores MODBUS Mestre

Consulte, na documentação do produto, as configurações típicas associadas à configuração do protocolo MODBUS Mestre. Alguns itens importantes para análise: Configurações Gerais MODBUS RTU Mestre, Diagnósticos MODBUS RTU Mestre, Configuração dos Mapeamentos MODBUS, Tipos de Dados suportados no MODBUS e Configuração, Diagnósticos e Códigos de Erro das Relações MODBUS.

Fonte: Manual de Utilização UCPs Série Nexto NX30xx – MU214100

Configuração do Protocolo MODBUS Escravo por Mapeamento Simbólico

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

1. Configurar os parâmetros gerais do protocolo MODBUS escravo, como: endereço do escravo e tempos de comunicação (disponível no botão de configurações avançadas do Escravo). A figura a seguir mostra essa configuração.

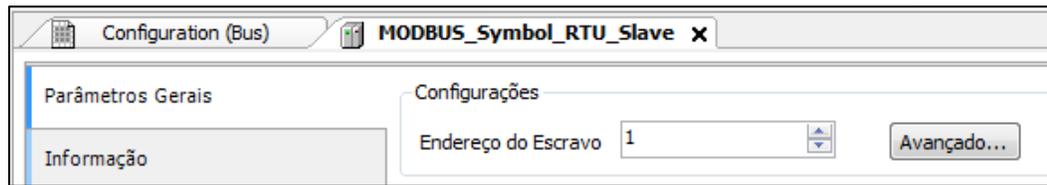


Figura 4-45. Configuração do Escravo

2. 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. Essa etapa está mostrada na figura a seguir.

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

Figura 4-46. Tela de Mapeamentos de dados MODBUS



Pesquisa na Documentação: editores MODBUS Escravo

Consulte, na documentação do produto, as configurações típicas associadas à configuração do protocolo MODBUS Escravo. Alguns itens importantes para análise: Configurações Gerais e Avançadas, Diagnósticos, Configuração dos Mapeamentos, Tipos de Dados suportados no MODBUS e Configurações das Variáveis de Representação Direta e Endereçamento (mapeamento).

Fonte: Manual de Utilização UCPs Série Nexto NX30xx – MU214100

MODBUS Ethernet

A rede de comunicação multimestre permite que as UCPs Nexto Xpress leiam ou escrevam variáveis MODBUS em outros controladores ou IHMs compatíveis com os protocolos MODBUS TCP ou MODBUS RTU via TCP. A UCP Nexto Xpress pode, simultaneamente, ser cliente e servidor em uma mesma rede de comunicação, ou até mesmo ter mais instâncias associadas à interface Ethernet, indiferente se elas são MODBUS TCP ou MODBUS RTU via TCP. A figura a seguir ilustra algumas das possibilidades de comunicação utilizando-se o protocolo MODBUS TCP simultaneamente com o protocolo MODBUS RTU via TCP.

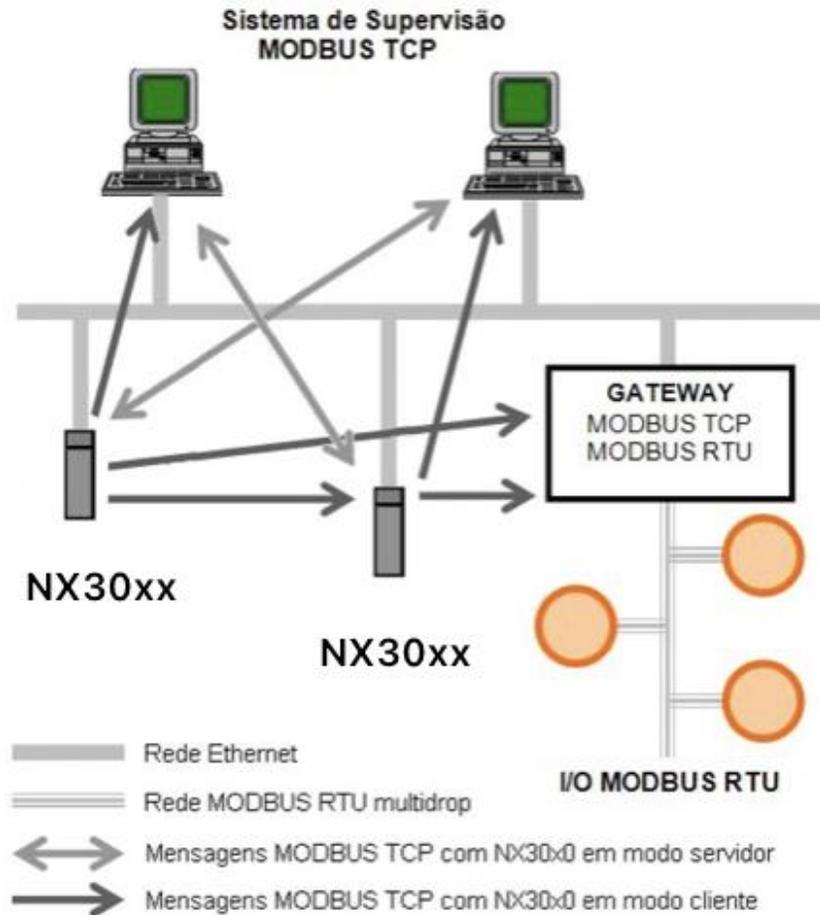


Figura 4-47. Rede de Comunicação MODBUS TCP

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

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

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

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

Independentemente do modo de configuração, os passos para inserir uma instância do protocolo e configurar a interface ethernet são iguais.

Este protocolo está disponível para as UCPs da Série Nexto nos seus canais Ethernet. Ao seleccionar esta opção no MasterTool IEC XE, a UCP passa a ser cliente ou servidor 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.

Inserção de uma instância do protocolo MODBUS Ethernet

O primeiro passo para configurar o MODBUS Ethernet, em modo cliente, é incluir a instância na NET desejada (nesse caso NET 1, pois a UCP NX3010 possui uma interface Ethernet). Clicar com o botão direito sobre a NET e seleccionar Acrescentar Dispositivo..., conforme mostra a figura a seguir.

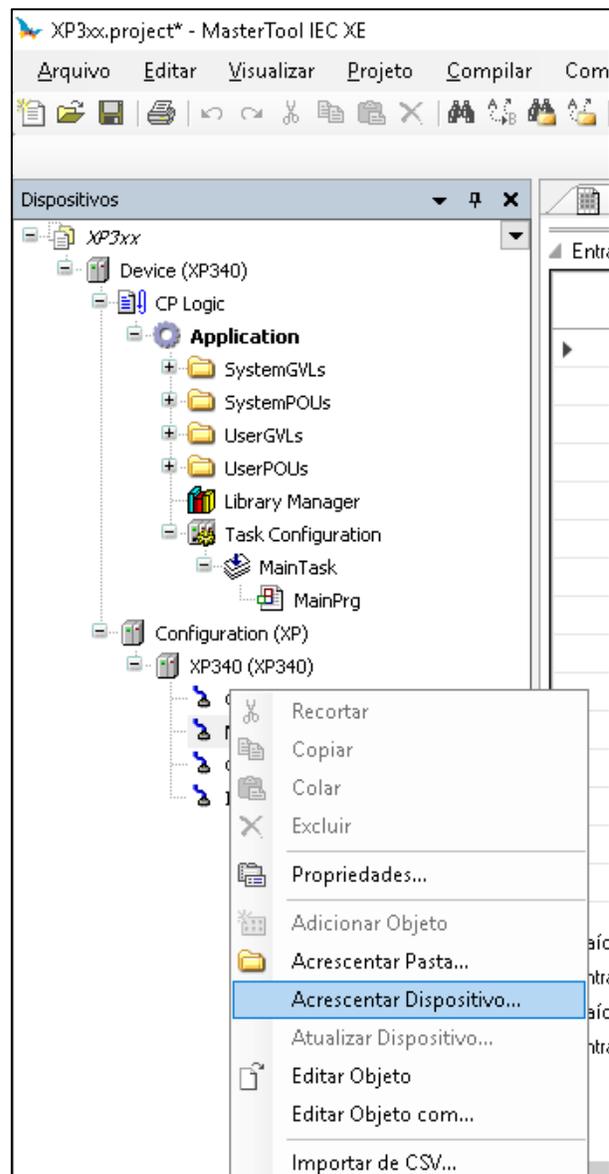


Figura 4-48. Adicionando a Instância MODBUS Ethernet

Após irá surgir na tela os protocolos disponíveis ao usuário. Definir o modo de configuração do protocolo, selecionar a opção desejada, por exemplo, MODBUS Symbol Client, para configuração por Mapeamento Simbólico. A seguir, clicar em Acrescentar, conforme mostra a figura a seguir.

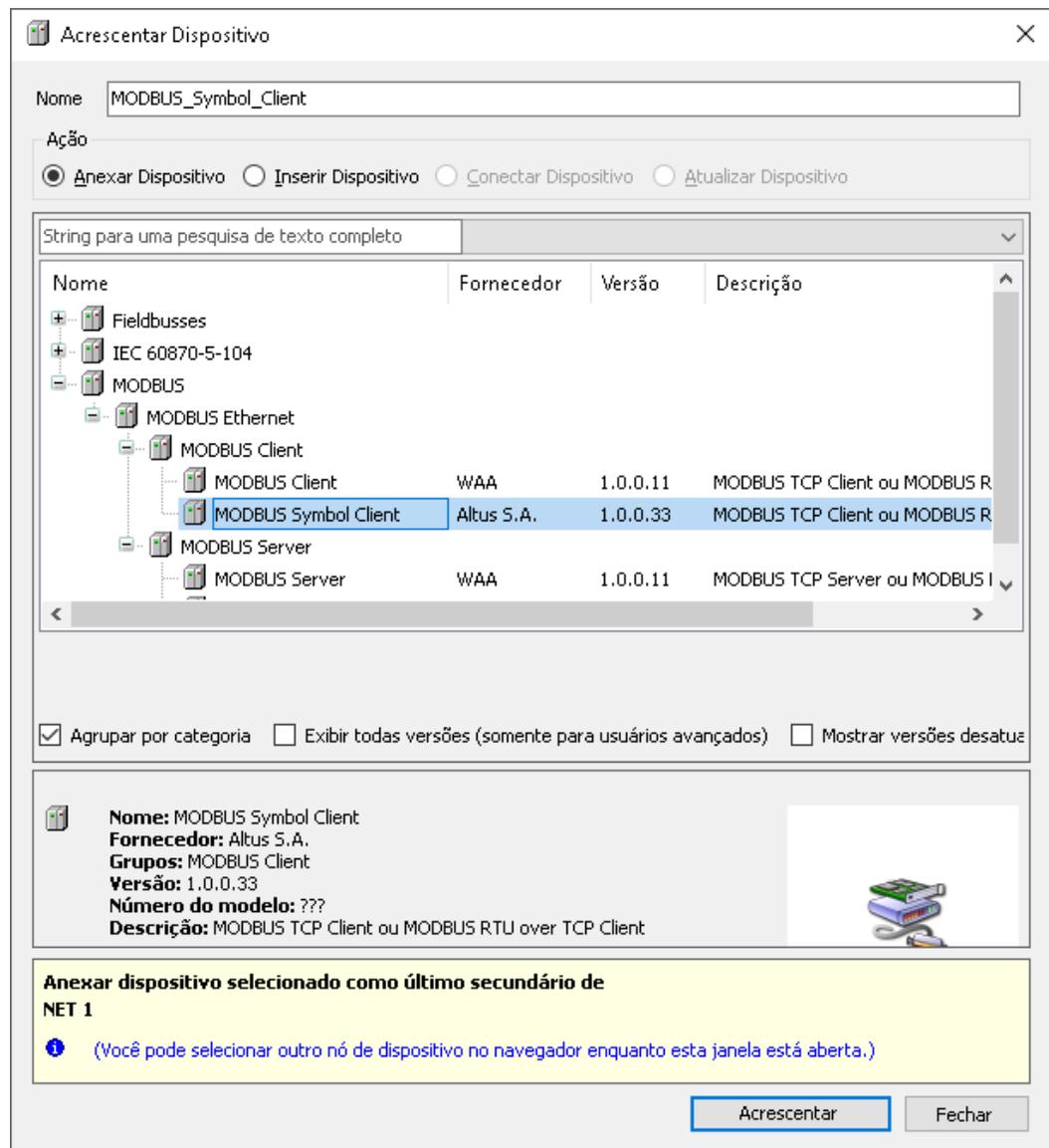


Figura 4-49. Selecionando o Protocolo MODBUS Ethernet

Configuração do Protocolo MODBUS Ethernet Cliente por Mapeamento Simbólico

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

1. Configurar os parâmetros gerais do protocolo MODBUS Cliente, com o protocolo TCP ou RTU via TCP. A figura na sequência mostra essa seleção.



Figura 4-50. Tela de Configuração Parâmetros Gerais MODBUS Cliente

- Adicionar e configurar dispositivos, definindo endereço IP, porta, endereço do escravo e time-out de comunicação (disponível no botão de configurações avançadas do Dispositivo). A figura a seguir mostra essa configuração.

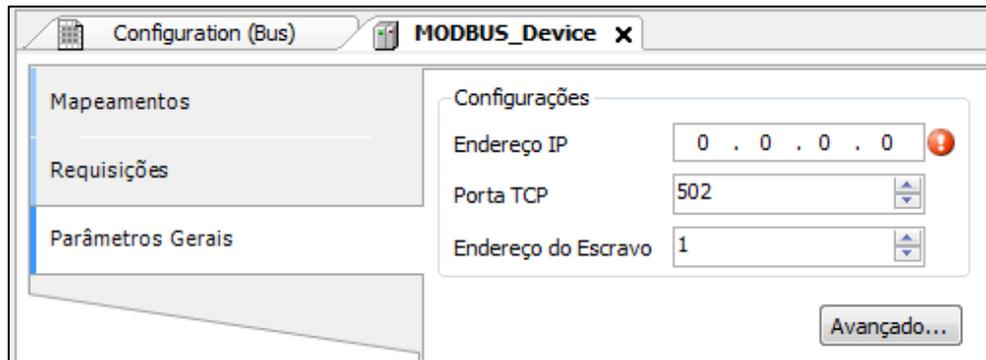


Figura 4-51. Tela de Configurações dos Parâmetros Gerais do Dispositivo MODBUS

- Adicionar e configurar os mapeamentos MODBUS (ver figura a seguir), especificando o nome da variável, tipo de dados, endereço inicial do dado, tamanho do dado e variável que receberá os dados de qualidade.

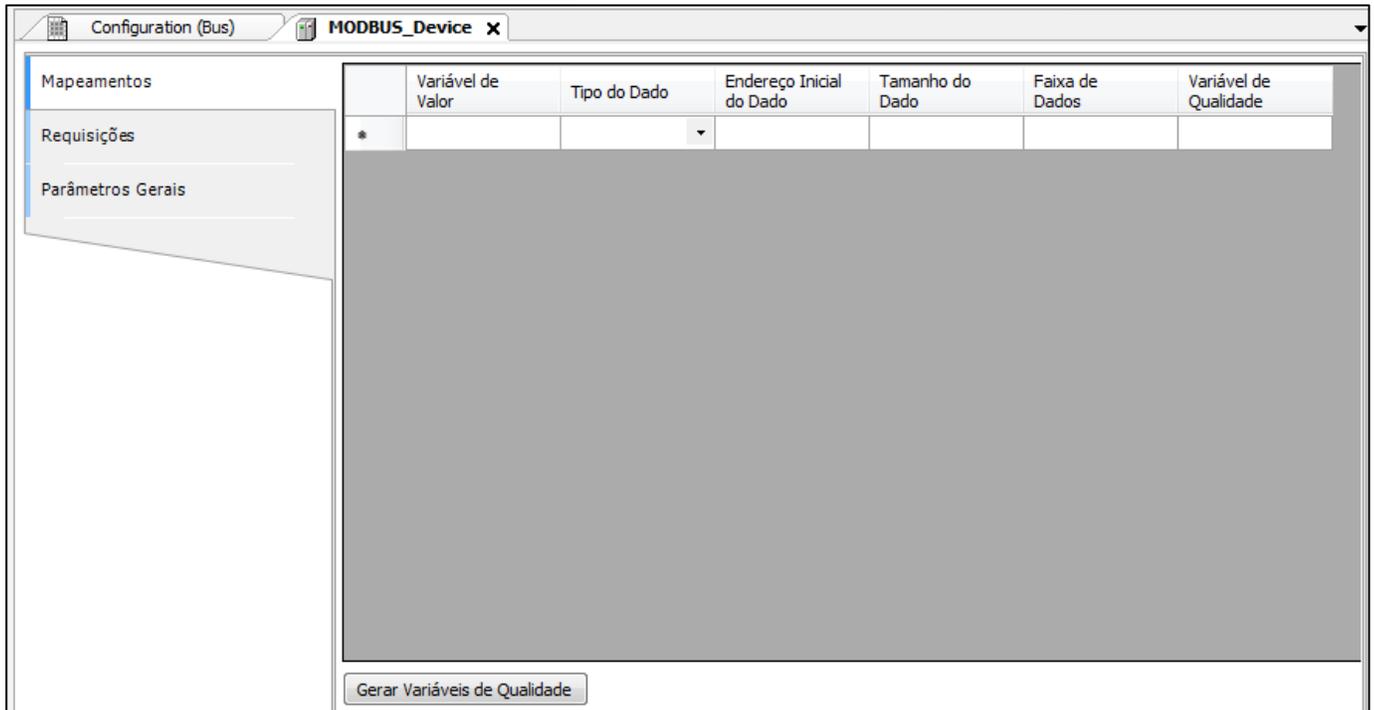
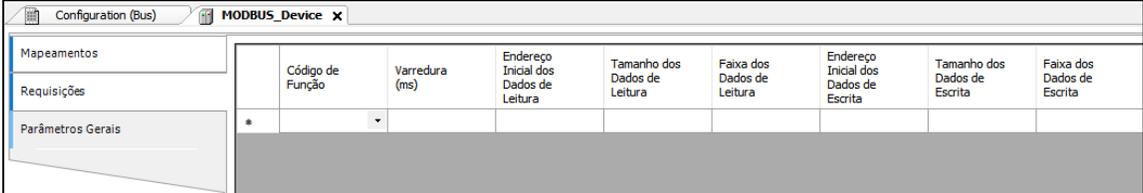


Figura 4-52. Tela de Mapeamentos de dados MODBUS Cliente

- Adicionar e configurar as requisições MODBUS conforme figura a seguir, 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.



Código de Função	Varredura (ms)	Endereço Inicial dos Dados de Leitura	Tamanho dos Dados de Leitura	Faixa dos Dados de Leitura	Endereço Inicial dos Dados de Escrita	Tamanho dos Dados de Escrita	Faixa dos Dados de Escrita
*							

Figura 4-53. Tela de Requisições de dados MODBUS



Pesquisa na Documentação: editores MODBUS Ethernet cliente

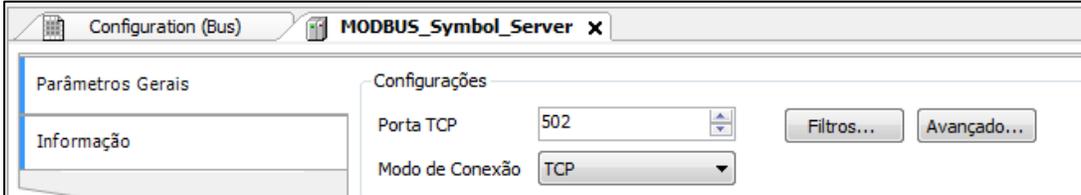
Consulte, na documentação do produto, as configurações típicas associadas à configuração do protocolo MODBUS Ethernet cliente. Alguns itens importantes para análise: Configurações Gerais e do Dispositivo, Diagnósticos, Parâmetros Gerais, Configurações Avançadas, Configuração dos Mapeamentos, Tipos de Dados suportados, Configuração das Relações, Funções MODBUS, Diagnósticos e Códigos de Erro das Relações e Mapeamentos do Dispositivo.

Fonte: Manual de Utilização UCPs Série Nexto NX30xx – MU214100

Configuração do Protocolo MODBUS Ethernet Servidor por Mapeamento Simbólico

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

1. 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). Ver figura a seguir.



Configuration (Bus) MODBUS_Symbol_Server x

Parâmetros Gerais

Configurações

Porta TCP: 502

Modo de Conexão: TCP

Filtros... Avançado...

Figura 4-54. Tela de Configuração Parâmetros Gerais MODBUS Servidor

2. Adicionar e configurar os mapeamentos MODBUS, especificando o nome da variável, tipo de dados, endereço inicial do dado e tamanho do dado. A figura a seguir ilustra essa configuração.

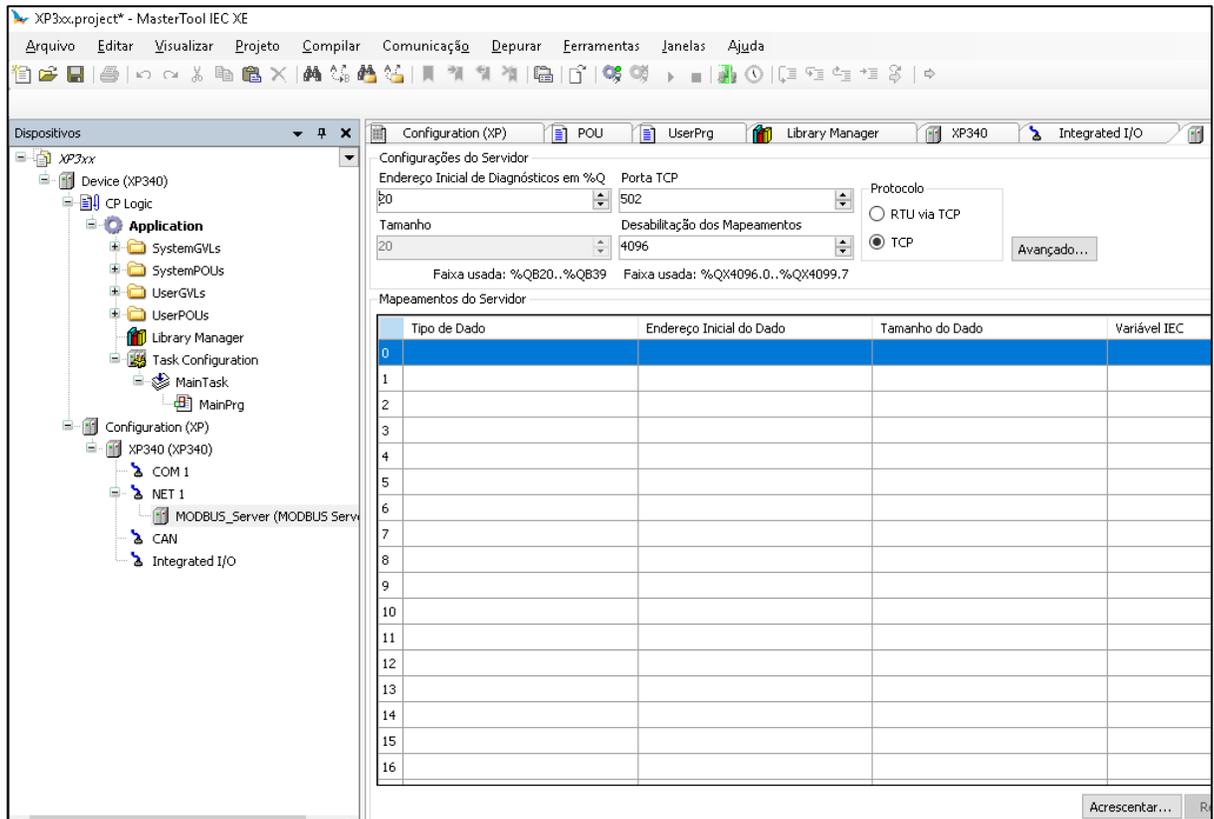


Figura 4-55. Tela de Configuração MODBUS Servidor



Pesquisa na Documentação: editores MODBUS Ethernet servidor

Consulte, na documentação do produto, as configurações típicas associadas à configuração do protocolo MODBUS Ethernet servidor. Alguns itens importantes para análise: Configurações Gerais, Filtros de IP, Configurações Avançadas, Diagnósticos, Configuração dos Mapeamentos, Configurações das Variáveis de Representação Direta e Endereçamento e Mapeamentos do Servidor.

Fonte: Manual de Utilização UCPs Série Nexto NX30xx – MU214100



Estudo Dirigido 4-1: configuração do protocolo MODBUS Ethernet

Desenvolva uma aplicação através do protocolo MODBUS Ethernet Symbol Cliente, conforme orientações a seguir. Para a realização desta situação de aprendizagem utilizar o monitor Modbus 32 MDBUS.

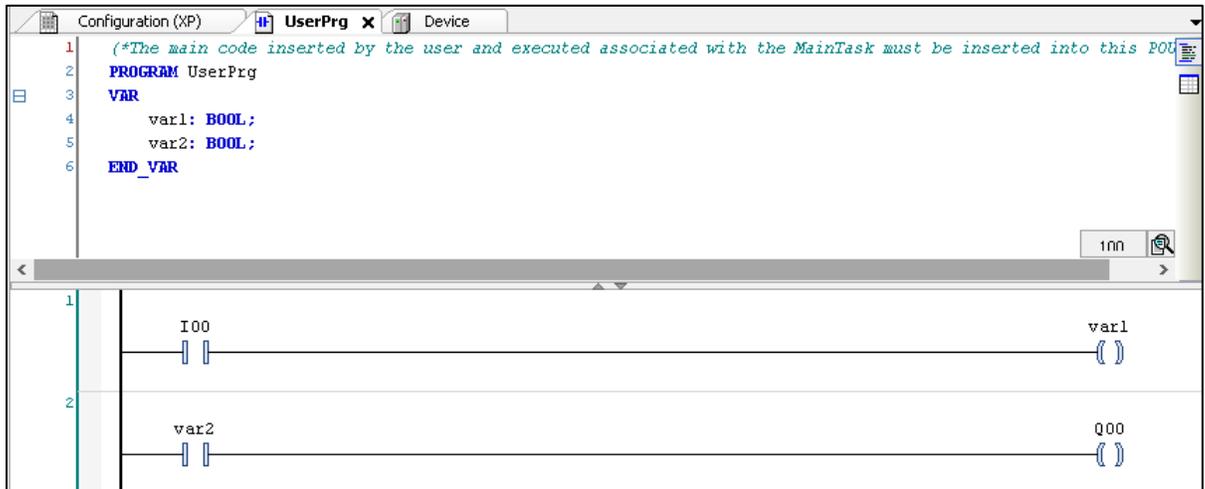
Orientações gerais referentes à realização da atividade

O objetivo principal desta aplicação é mostrar as facilidades na configuração das instâncias MODBUS na plataforma Nexto Xpress.

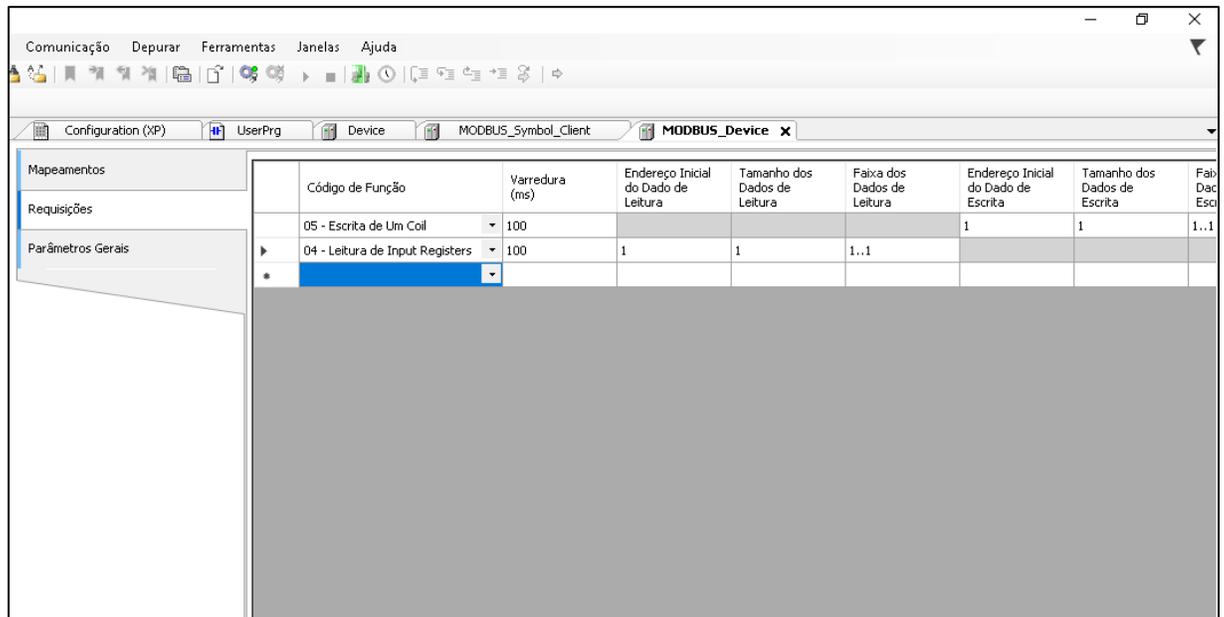
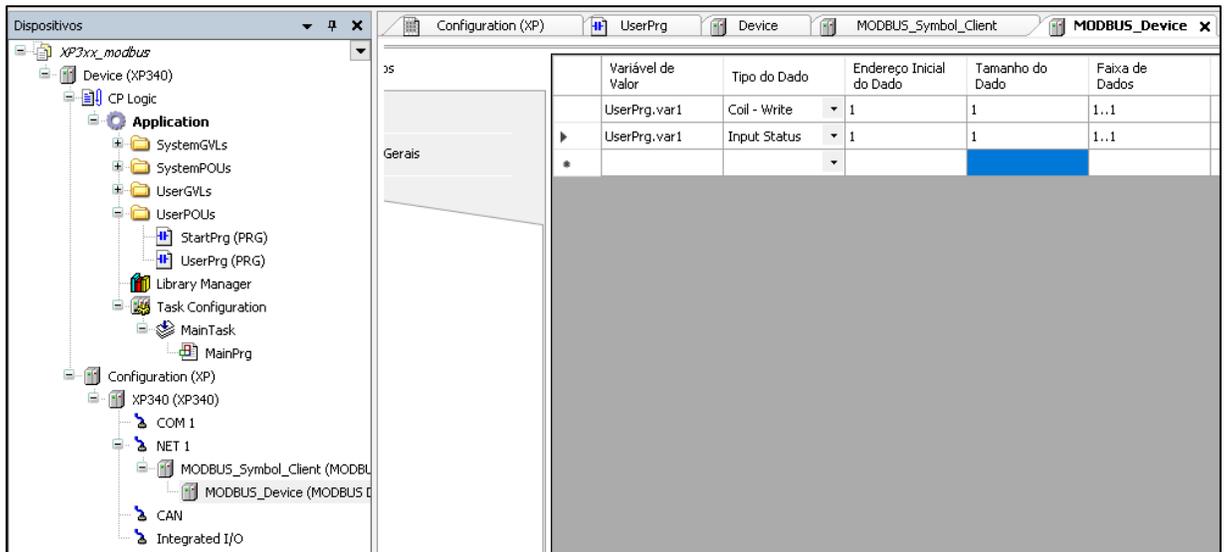
- Como acrescentar em NET1 um dispositivo MODBUS Ethernet Symbol Cliente?
- Como configurar o dispositivo MODBUS Ethernet Symbol Cliente?
- Como monitorar uma aplicação na ferramenta 32 MDBUS?

Com um duplo click na aba NET1 em MODBUS_Ethernet_Symbol_Client, o usuário deve configurar a instância MODBUS, o endereço IP (do computador aonde o monitor 32 MDBUS será executado) e a porta de comunicação (502) por padrão de fábrica.

Desenvolver o programa LADDER da figura a seguir com as respectivas variáveis.



Utilizar os parâmetros que seguem no mapeamento MODBUS.



Configuração da ferramenta 32MDBUS.

- Como configurar a ferramenta?
- Qual endereço de IP colocar?
- Qual porta devo configurar?
- Qual modo devo selecionar?

Mdbus Configuration
03-Aug-2011 19:42:40

Main Always On Top

Mode: **SLAVE** Comm. Port: **COM6**

Type: **NORMAL** Fmt./Bd./Py: **RTU 115200 N**

Internet Protocol Communications
IP Comm. Enable IP Modbus: **RTU via TCP**
IP Addr: **192.168.15.68** IP Port No.: **502**

Modbus Slave No.: **1** RTS/TX Delay (m.s.): **55** Invalid CRC/LRC 212 AT Modem Regs./Request: **125**

Database Points		
Type	Start	Number
Coils	1	50
Status	1	0
I. Regs.	1	0
H. Regs.	1	10
Floats **	1	0
Longs **	1	0

* 32 bit registers
** Modicon Format Flt.
* Modicon Format Lng.

Master Only
No Resp. T.O. (m.s.): **1000**
Fail Try Count: **3**
Poll Delay (.1 sec.): **10**

Monitor Only
Mdbus SLV No. (hex): **FF**
Mdbus Func. No. (hex): **FF**
Capture CRC/LRC Error
Log File Capture

Dial Out Parameters
Phone No. (AT): **AT&C1&D2DT1-403-5551212**
Dial Try Count: **3** Dial T.O. (sec.): **45**

Master/Slave Configuration-Database File
 File Ident. Load File(s) Data Save File(s)

Master Only DDE
Excel DDE Excel Sprdsht: DDE T.O. (sec.): **15**
Macro DDE Appl,Topic: Macro:

RTS Extra (m.s.): **0**
CTS Enable
Longs Rd. Use Fnc. **4**
Unsigned H. Regs.



Fifo
Rx Fifo **8** Tx Fifo **14**
See Fifo Help on above!

Help
Cancel
Ok

Selecionar o modo *Slave*, comunicação via RTU via TCP e porta 502 conforme configuração padrão mostrado na figura acima. Importante: utilizar este mesmo padrão de configuração da ferramenta de monitoração 32MDBUS para as demais aplicações.



DICA: consulte o instrutor para orientações adicionais referentes à execução desta atividade.

Estudo Dirigido 4-2: aplicação do protocolo MODBUS Ethernet



Desenvolva uma aplicação através do protocolo MODBUS Ethernet Client, conforme orientações a seguir para escrever via ferramenta de simulação MDBUS32 em uma memória de dados no Nexto Xpress. Para a realização desta situação-problema utilizar o monitor Modbus 32 MDBUS com as mesmas configurações do estudo dirigido anterior. Esta situação de aprendizagem vai explorar os seguintes tópicos:

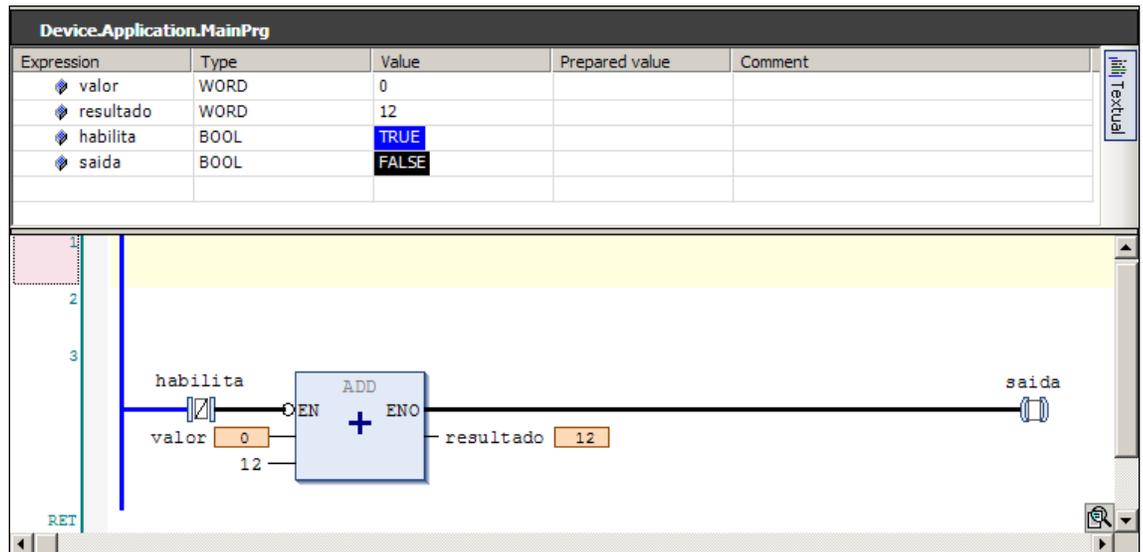
- Como escrever dados em um registrador no MODBUS Ethernet Cliente?
- Como forçar valores e monitorar estados de variáveis no ambiente MasterTool IEC XE?
- Como monitorar dados via ferramenta de simulação 32MDBUS e confirmar na plataforma MasterTool IEC XE?

Orientações gerais referentes à realização da atividade

Para escrever os dados nos registradores utilize a ferramenta 32MDBUS. Selecione na aba Displays, da ferramenta, a opção *Holding Regs* para a realização desta experiência prática. Configure o protocolo MODBUS Ethernet Client no MasterTool IEC XE. Configure o cliente MODBUS com os parâmetros da tabela.

Configuração (Menu em inglês)	Descrição	Possibilidades
Data Type	Função MODBUS	Holding Register
Slave Address	Endereço MODBUS do Escravo	1
Read Data	Endereço de Leitura de dados	0
Read Data Size	Tamanho da variável	1
Read IEC Variable	Correlação com a variável IEC	%IW2

Na área de desenvolvimento edite o programa conforme a figura abaixo e coloque o sistema em *run* através do *login*.



Observe que a variável "valor" está com o valor "0". Através da ferramenta 32MDBUS selecione em Mdbus *Holding Regs* conforme a figura a seguir.



Selecione um valor e *click* em *Accept* para enviar o valor para o endereço da CPU do NEXTO. Verificar e forçar o contato habilita para *False* para enviar o valor armazenado no endereço “valor” para ser adicionado ao número 12 da função ADD. Após verificar que a variável resultado armazenará o valor 56 em seu endereço, conforme pode ser visualizado na figura que segue.

Expression	Type	Value	Prepared value	Comment
valor	WORD	44		
resultado	WORD	56		
habilita	BOOL	F FALSE		
saida	BOOL	TRUE		

DICA: consulte o instrutor para orientações adicionais referentes à execução desta atividade.

Servidor OPC DA

Para comunicar com as UCPs da Série Nexto Xpress 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).

Em outras palavras, a relação entre as tags dos sistemas de supervisão e os dados do processo nas variáveis do controlador é totalmente transparente. Isso significa que se as

áreas de dados são alteradas no decorrer do desenvolvimento do projeto, não há a necessidade de refazer relações entre as informações do CP com o SCADA. Basta utilizar a nova variável disponibilizada pelo CP nos sistemas que requisitam esse dado.

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

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

A figura a seguir apresenta uma arquitetura para comunicação de sistema SCADA e CPs em projeto de automação. Todos os papéis presentes na comunicação estão explícitos nesta figura independentemente do local onde estejam executando, eles podem estar em um mesmo equipamento ou em equipamentos diferentes.

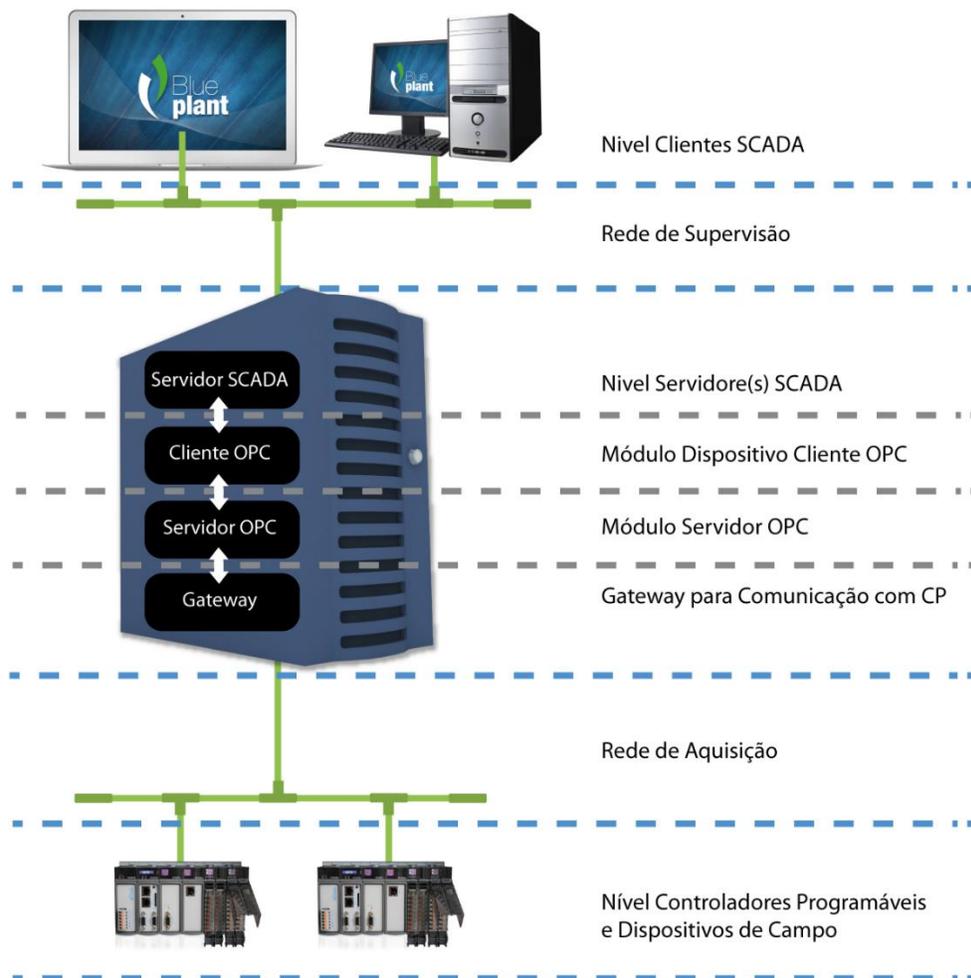


Figura 4-56. Arquitetura OPC DA

Cada um dos papéis desta arquitetura é descrito na sequência.

- **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 armazenadas. 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 Xpress é utilizado um gateway que permite esta comunicação.
- **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.
- **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.

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. A figura a seguir ilustra o resultado dessa seleção.

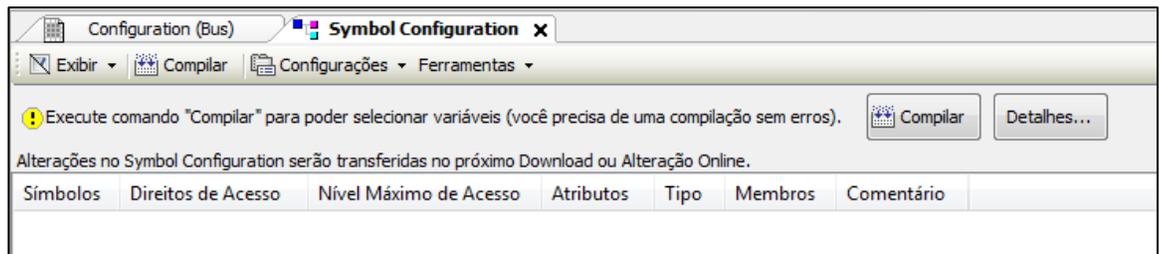


Figura 4-57. Objeto Symbol Configuration

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 anterior desapareça. Após isso todas as variáveis disponíveis no projeto, sejam declaradas em POU's, GVL's e diagnósticos, serão exibidas e podem ser selecionadas (veja figura a seguir). As variáveis selecionadas estarão disponíveis no Servidor OPC DA para acesso pelos Clientes.

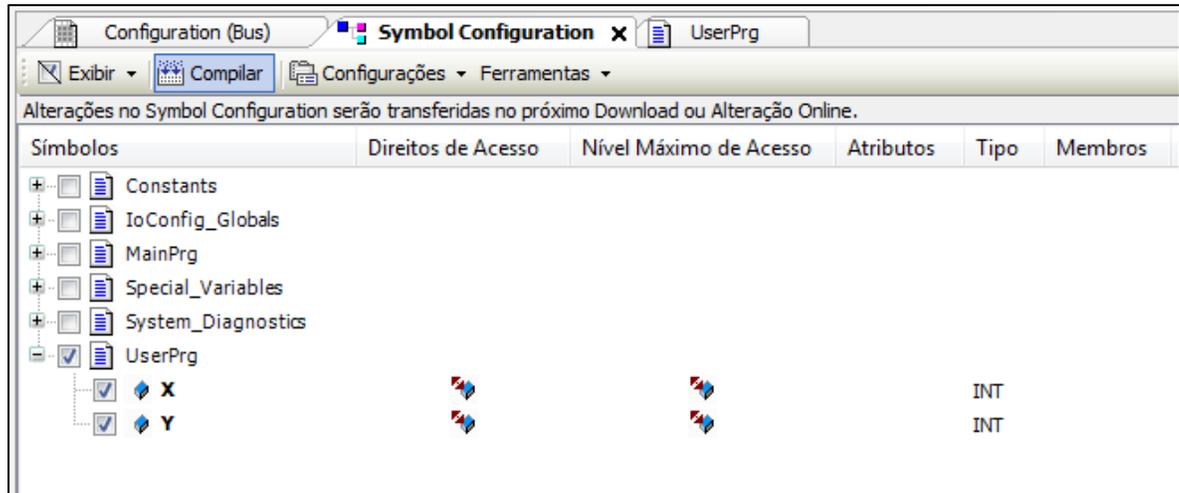


Figura 4-58. Selecionando Variáveis na Symbol Configuration

Após este procedimento o projeto pode ser carregado no CP e as variáveis selecionadas estarão disponíveis para comunicação com o Servidor OPC DA. Se a tela do Objeto Symbol Configuration estiver aberta e alguma das variáveis, POU's ou GVL's selecionadas for alterada, os nomes destes objetos aparecerão na cor vermelha. As situações nas quais isso acontece são caso a variável seja deletada ou o valor do atributo tenha sido modificado.

Também é possível configurar quais variáveis estarão disponíveis no Servidor OPC DA através de um atributo inserido diretamente nas POU's ou GVL's onde as variáveis são declaradas. Quando o atributo '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.

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 em que 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.

Configurando um CP no Servidor OPC DA

A configuração de um CP é executada dentro do MasterTool IEC XE através da opção disponível no menu Comunicação.

É necessário que o MasterTool IEC XE seja executado como administrador.

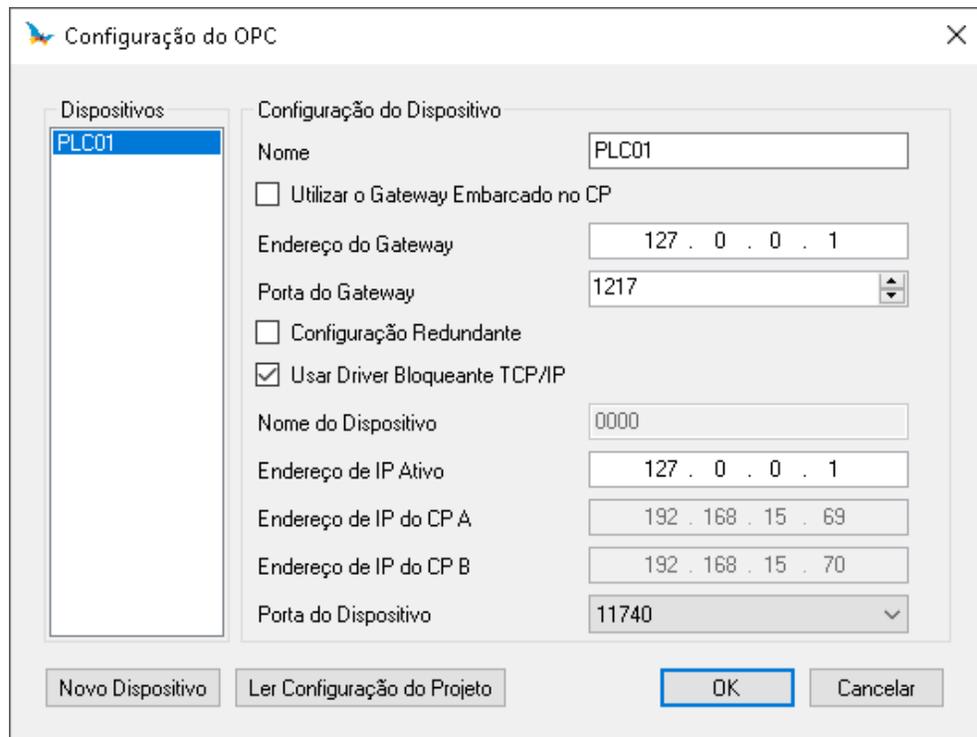


Figura 4-59. Configuração do OPC DA Servidor

A Configuração do Gateway é a mesma configurada no Gateway utilizado para comunicação entre o MasterTool IEC XE e o CP. Se a configuração utilizada for localhost o Endereço do Gateway deve ser preenchido com 127.0.0.1. Esta configuração é necessária, pois o Servidor OPC utiliza o mesmo gateway de comunicação e o mesmo protocolo utilizados na comunicação entre CP e MasterTool IEC XE.

Existe a opção de 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.

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

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

Importando uma Configuração do Projeto

Utilizando o botão Ler Configuração do Projeto 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. 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.

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 Xpress 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 à descrita anteriormente. 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 IEC XE. 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 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 atributo '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 mostrada anteriormente. 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. 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.



Pesquisa na Documentação: OPC DA Servidor

Consulte, na documentação do produto, as configurações típicas associadas à configuração do protocolo OPC DA Servidor. Alguns itens importantes para análise: Descrição dos Campos da Tela do Objeto Symbol Configuration, Parâmetros de Configuração de Cada CP para o Servidor OPC DA Descrição dos estados da comunicação do Servidor OPC DA com o CP, Descrição do valor da Qualidade OPC DA e Limites da Comunicação com o Servidor OPC DA.

Fonte: Manual de Utilização UCPs Série Nexto NX30xx – MU214100

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 a seguir exibe a seleção do servidor no driver cliente do software SCADA BluePlant.

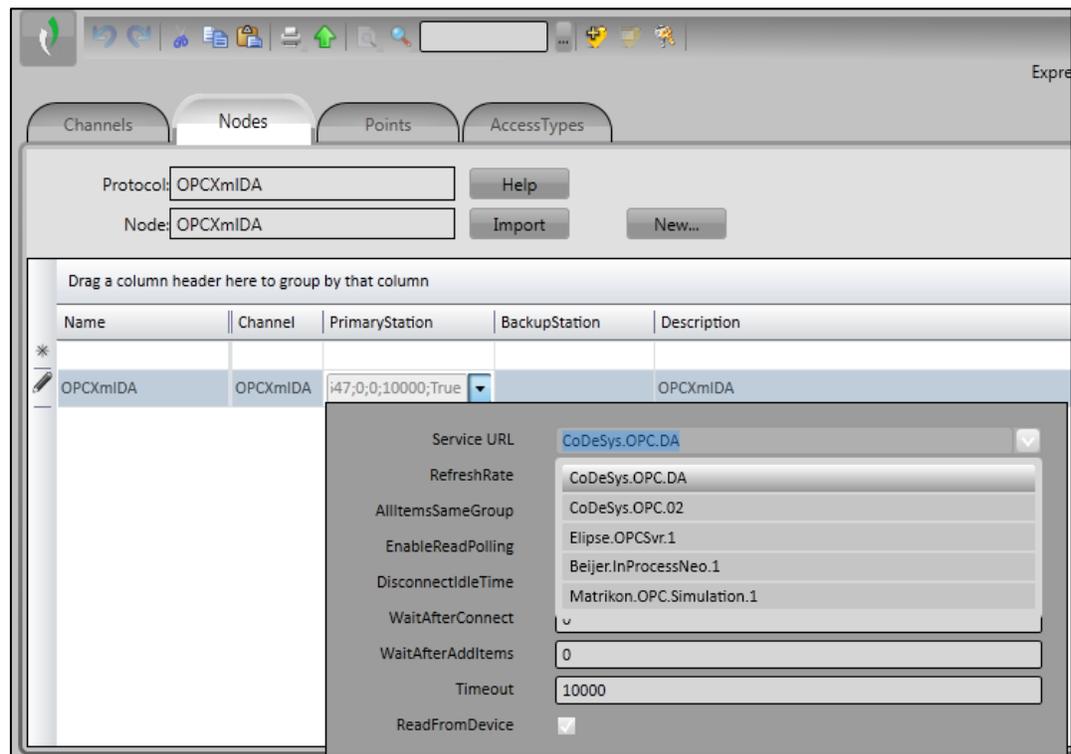


Figura 4-60. Selecionando Servidor OPC DA na Configuração do Cliente

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.

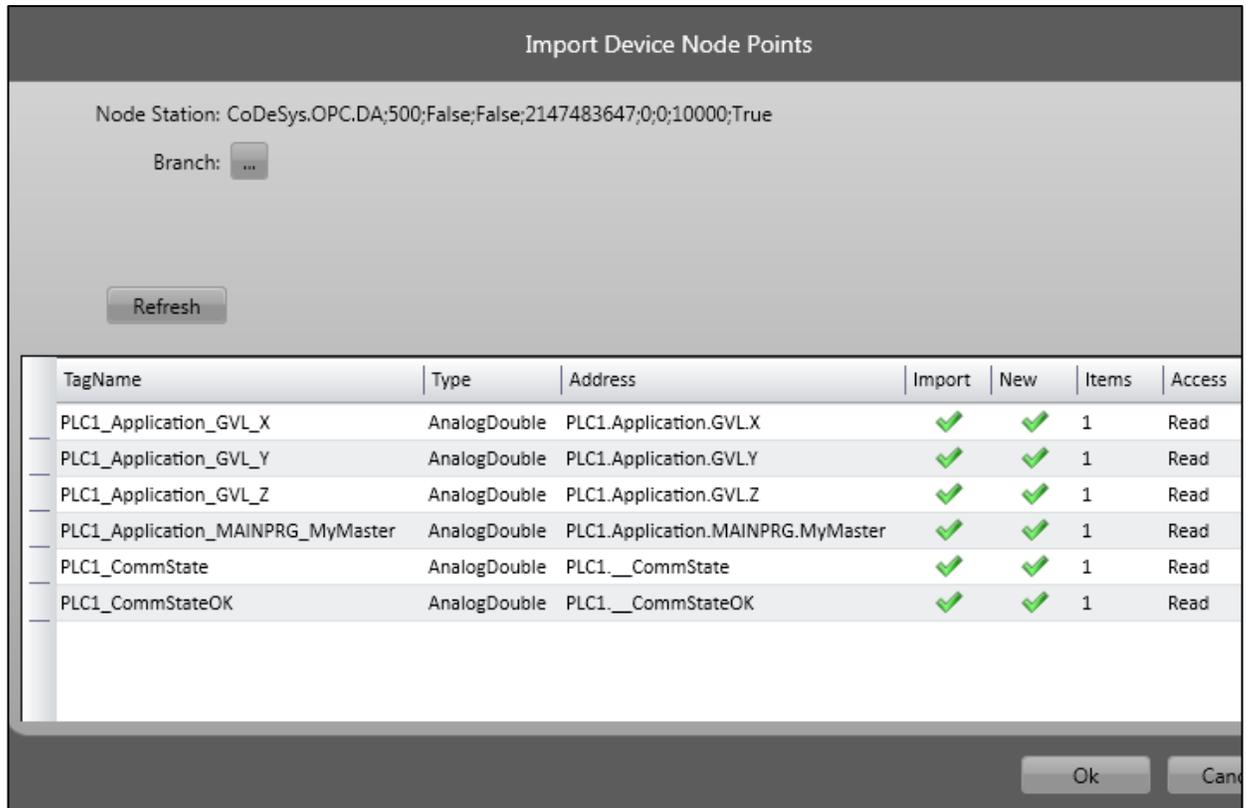


Figura 4-61. Lista de Símbolos Consultados pelo Cliente OPC DA

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.

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.

A figura a seguir 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 independentemente do local onde estejam executando, eles podem estar em um mesmo equipamento ou em equipamentos diferentes.

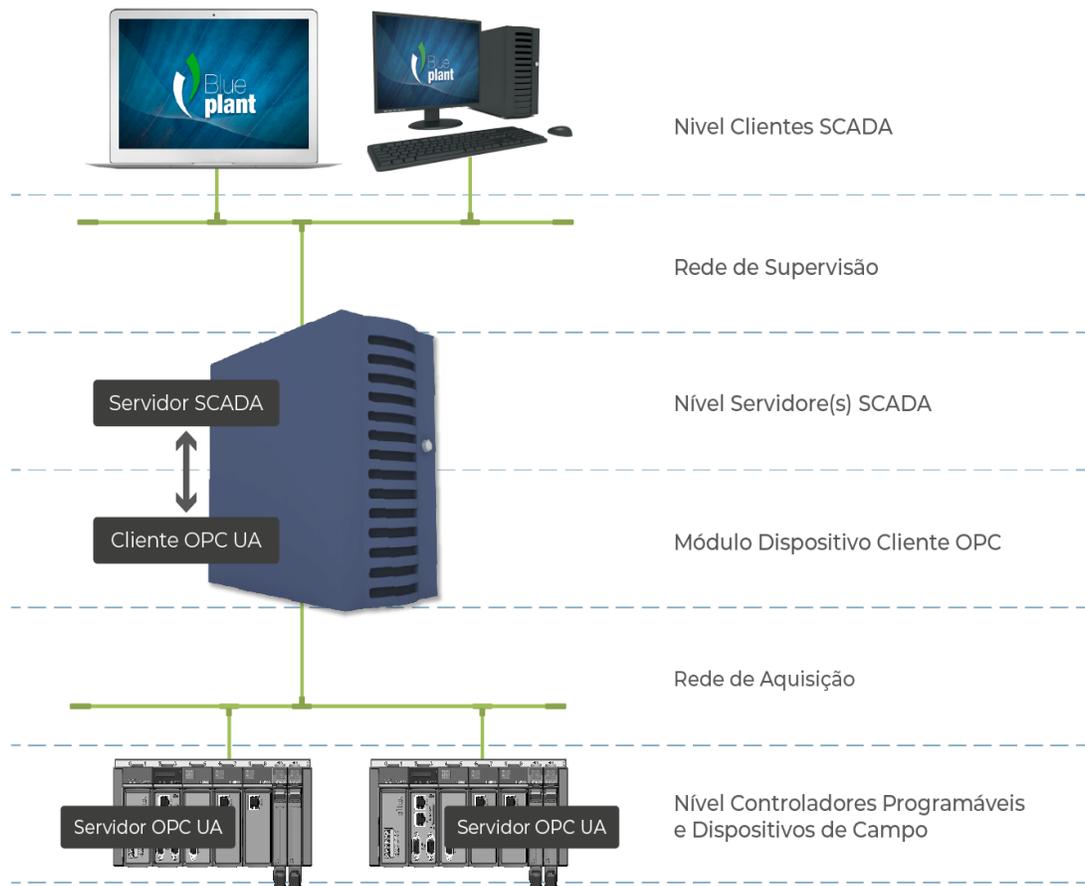


Figura 4-62. Arquitetura Típica OPC UA

Cada um dos papéis desta arquitetura é descrito na sequência.

- **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.

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

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

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

Criando um Projeto para Comunicação OPC UA

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 a seguir.

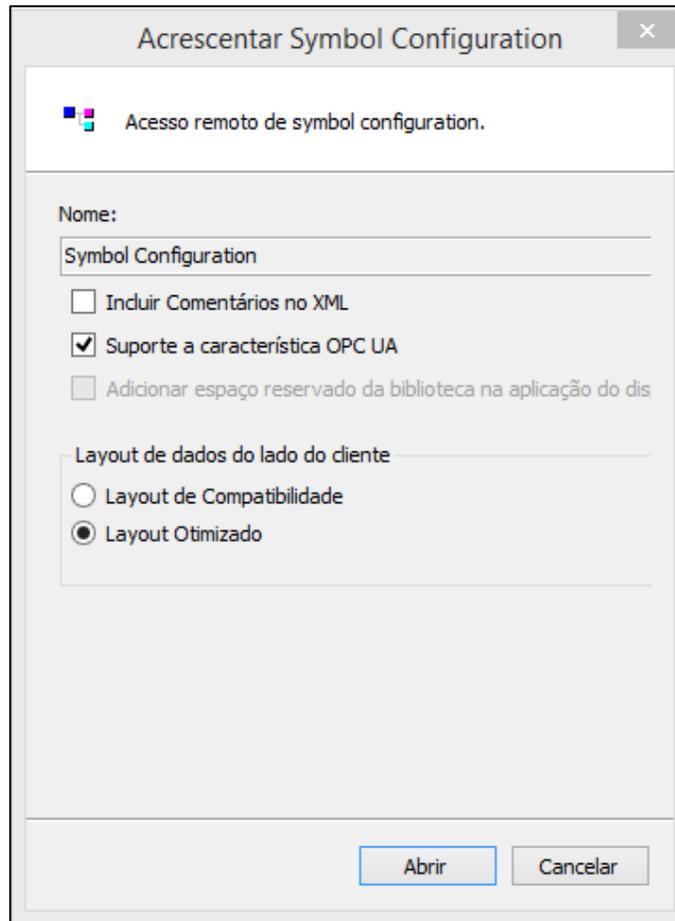


Figura 4-63. Objeto Symbol Configuration (OPC UA)

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 a seguir.

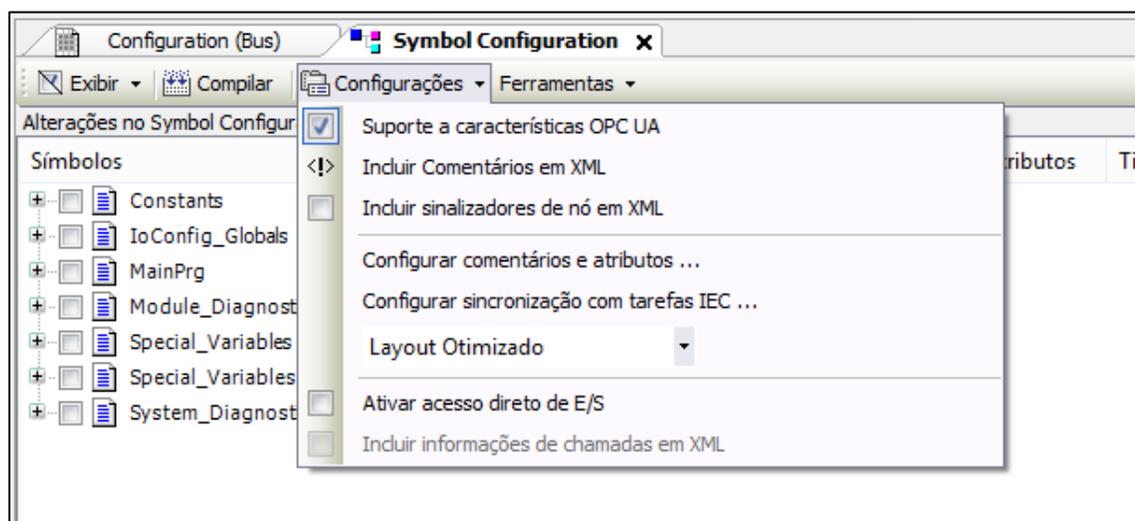


Figura 4-64. Habilitando OPC UA no Objeto Symbol Configuration

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



Um pouco de teoria... Segurança cibernética

A segurança cibernética desempenha um papel crucial no ambiente de automação industrial. Com o aumento alarmante de incidentes de segurança em fábricas, plantas e outras aplicações automatizadas, medidas efetivas para proteger esses sistemas tornaram-se imperativas. Instituições governamentais, como a ICS-Cert e o Departamento Federal Alemão para Segurança da Informação (BSI), têm acompanhado de perto o aumento desses incidentes. Diante desse cenário, a elaboração de metodologias que assegurem a integridade e proteção dos sistemas tornou-se uma necessidade urgente. Um marco importante nesse sentido é a diretriz de padrão internacional IEC 62443, inicialmente publicada pelo comitê de segurança de sistemas de controle e automação industrial (ISA99) da Sociedade de Automação Industrial (ISA) e frequentemente referida como norma ISA/IEC 62443.

O escopo das medidas de cibersegurança abrange a proteção de vários aspectos, incluindo a disponibilidade das funcionalidades do controlador, a funcionalidade da aplicação, a confidencialidade do código fonte e da aplicação, a integridade das funções de aplicação, do sistema de desenvolvimento e dos componentes empregados, além da autenticidade do controlador e seus dados.

A figura a seguir ilustra as possíveis vulnerabilidades de um típico sistema de automação.

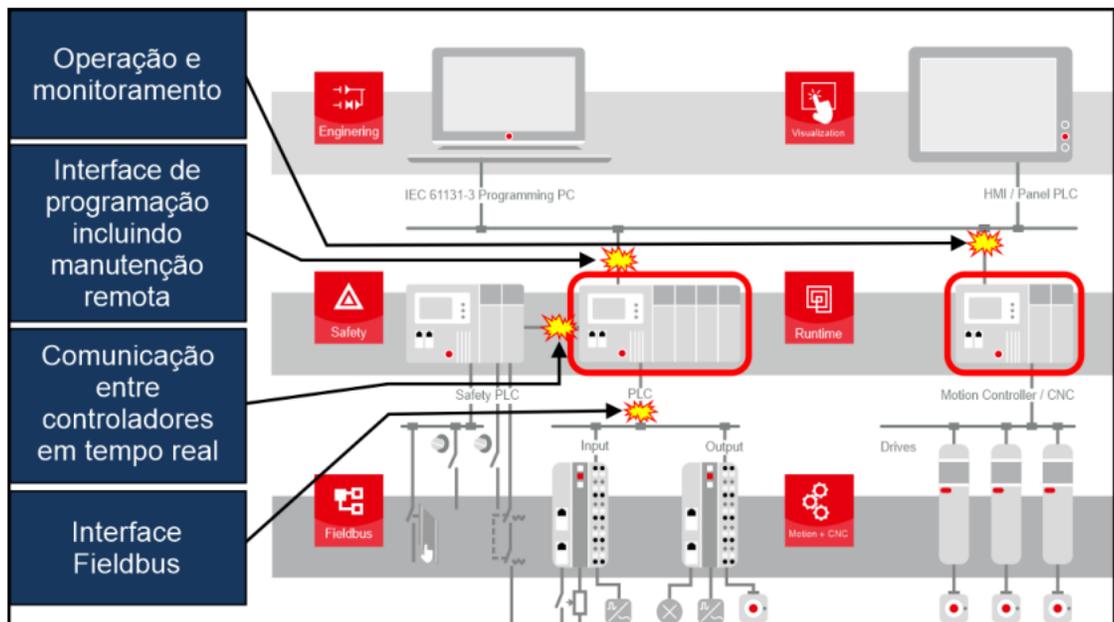


Figura 4-65. Possíveis vulnerabilidades de um típico sistema de automação

Todas as medidas de segurança comumente conhecidas para computadores devem ser aplicadas em redes com equipamentos de automação industrial, tais como:

- Proteção contra vírus;
- Senhas fortes que são regularmente alteradas;
- Proteção de firewall;
- Uso de túneis VPN para conexões entre redes;
- Cautela ao lidar com dispositivos de armazenamento removíveis, como pen drives USB.

Além disso, é obrigatório ter um gerenciamento de usuários e permissões bem definidos para o acesso aos controladores e suas redes interconectadas. a colaboração entre fornecedores, integradores, operadores e usuários é fundamental para promover uma cultura de segurança robusta e eficiente.

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 em um servidor OPC UA deve-se criar um certificado, 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 e
 - 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, Unstruted 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.



Pesquisa na Documentação: OPC UA Servidor

Consulte, na documentação do produto, as configurações típicas associadas à configuração do protocolo OPC UA Servidor. Alguns itens importantes para análise: Tipos de Variáveis Suportadas, Limite de Clientes Conectados no Servidor OPC UA, Limite de Variáveis de Comunicação no Servidor OPC UA, Configurações de Criptografia e Principais Parâmetros de Comunicação Ajustados em um Cliente OPC UA.

Fonte: Manual de Utilização UCPs Série Nexto NX30xx – MU214100

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 a seguir exibe a seleção do servidor no driver cliente do software SCADA BluePlant.

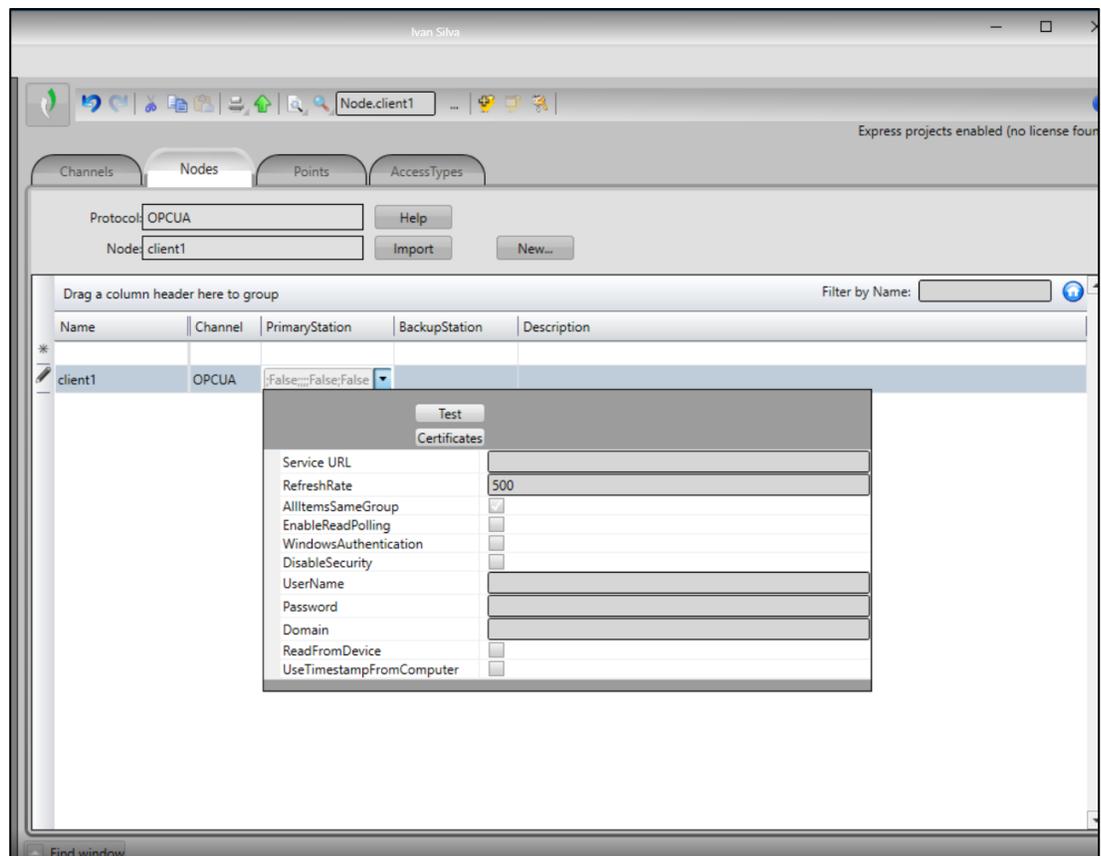


Figura 4-66. Selecionando Servidor OPC UA na Configuração do Cliente

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.

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

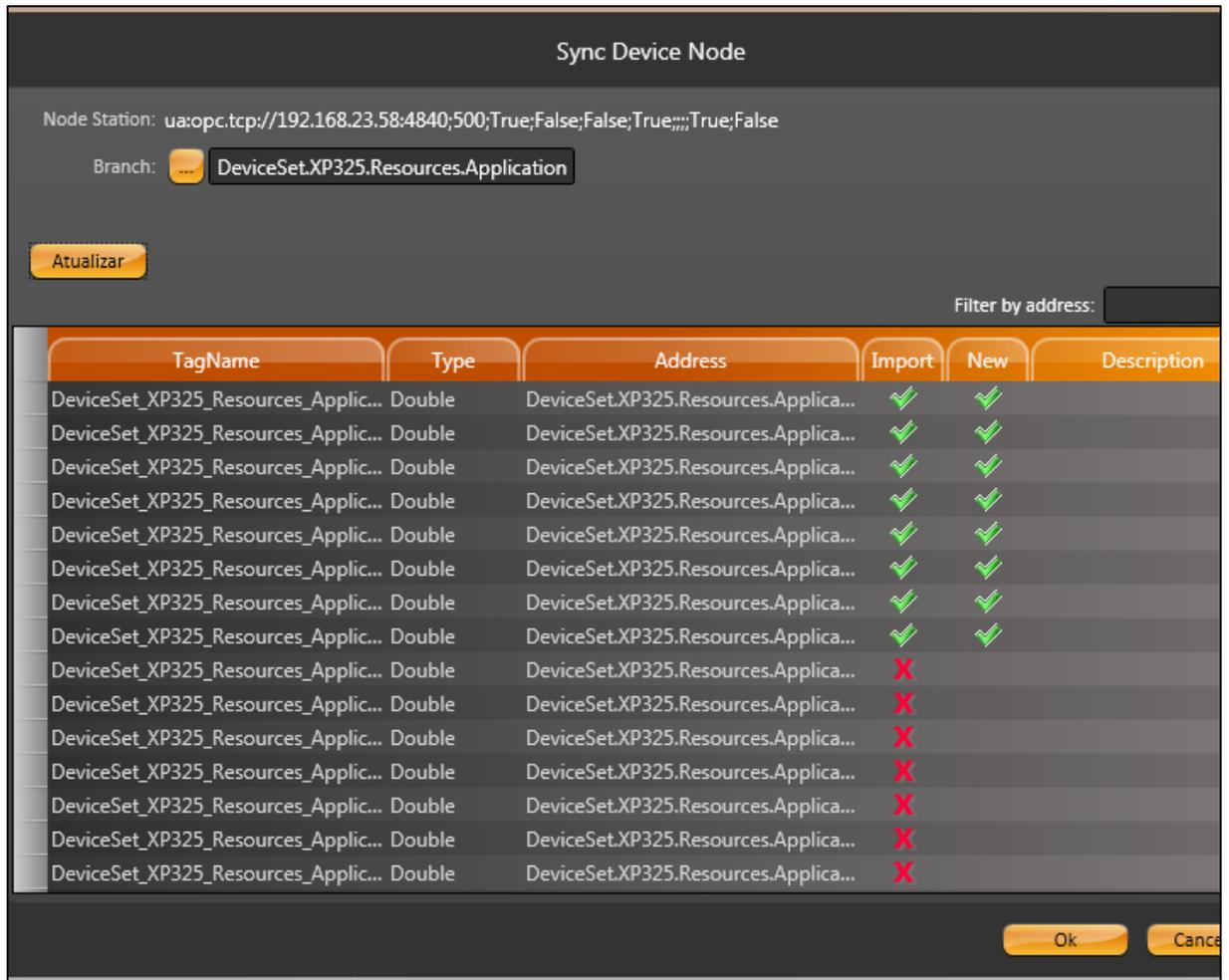


Figura 4-67. Lista de Símbolos Consultados pelo Cliente OPC UA



Estudo Dirigido 4-3: configurando o Nexto Xpress como servidor OPC UA

Desenvolva uma aplicação com ênfase no protocolo OPC UA que permita a comunicação entre o CP Nexto Xpress (servidor) e o SCADA BluePlant (cliente).



DICA: consulte o instrutor para orientações adicionais referentes à execução desta atividade.

EtherNet/IP

O EtherNet/IP é um protocolo de arquitetura mestre-escravo, o qual consiste em um EtherNet/IP Scanner (o mestre) e mais um EtherNet/IP Adapter (o escravo). O editor do EtherNet/IP fornece diálogos para configurar parâmetros e mapear entradas/saídas para variáveis.

O EtherNet/IP é um protocolo baseado na CIP (Common Industrial Protocol), o qual tem dois propósitos primários: (1) O transporte de dados de controle-orientado associados com dispositivos de E/S e (2) o transporte de outras informações relacionadas ao sistema sendo controlado, tais como parâmetros de configuração e diagnósticos. O primeiro é realizado por mensagens implícitas, enquanto o segundo é realizado através de mensagens explícitas.

O sistema em execução, das UCPs, pode atuar tanto como Scanner como Adapter. Cada interface NET das UCPs suporta apenas uma instância EtherNet/IP e ele não pode ser instanciado em um módulo de expansão Ethernet.

Uma UCP suporta até duas instâncias EtherNet/IP Scanner, uma por interface NET, mas no máximo 32 dispositivos Adapters no total, seja sob apenas uma ou sob duas instâncias EtherNet/IP Scanner. E cada um destes dispositivos Adapters não pode ter mais do que 500 Entradas/Saídas (assemblies).

Uma instância EtherNet/IP Adapter suporta até 64 módulos de entrada ou saída. Estes módulos podem ser do tipo BYTE, WORD, DWORD ou REAL. O intervalo da MainTask de um dispositivo rodando como Adapter deve ser menor ou igual ao RPI.

Interface EtherNet/IP

Para adicionar um EtherNet/IP Scanner ou um Adapter é necessário adicionar uma Interface EtherNet/IP abaixo da interface NET (ver figura a seguir). Isto pode ser feito através do comando Acrescentar Dispositivo.

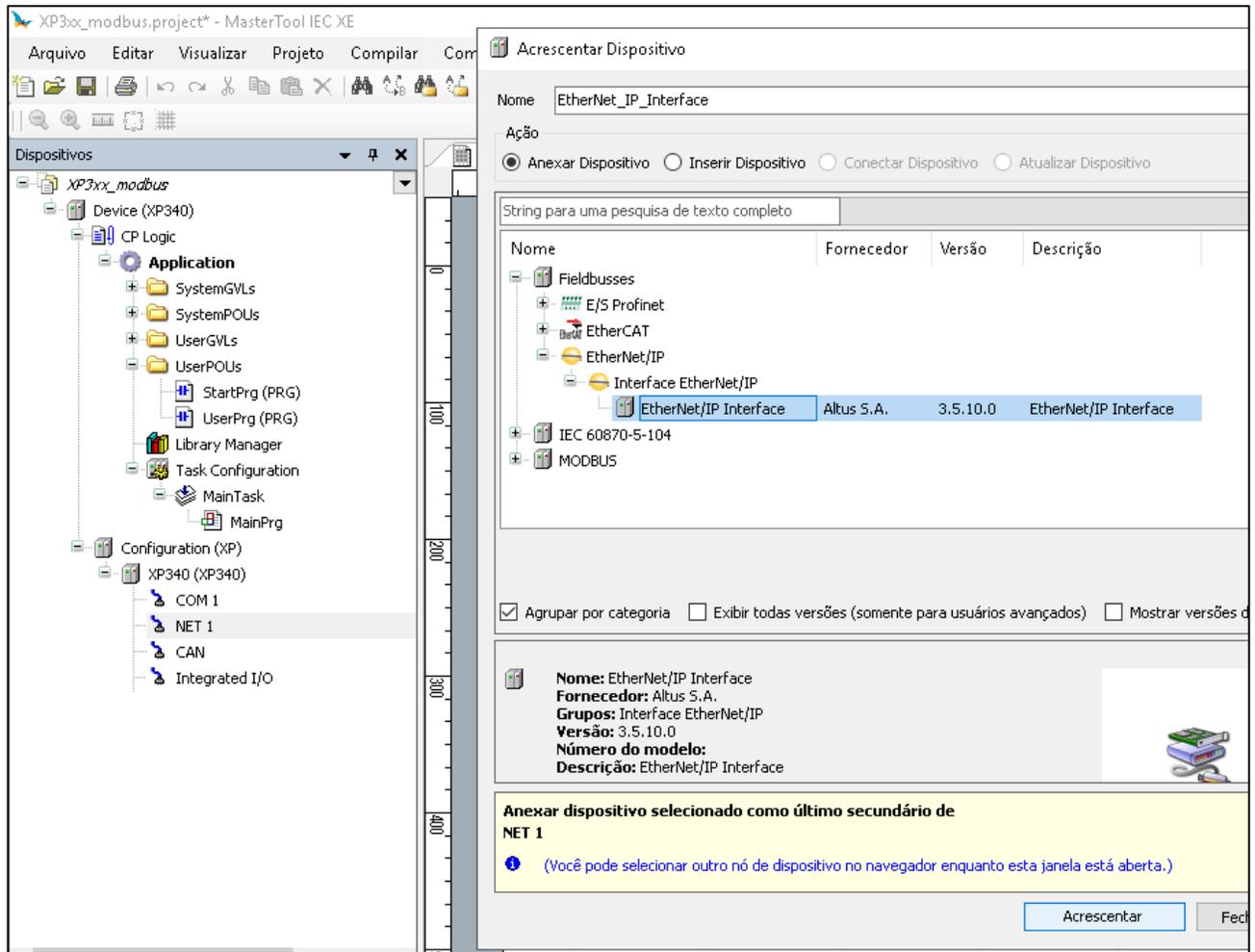


Figura 4-68. Adicionando uma Interface EtherNet/IP

Abaixo desta interface EtherNet/IP é possível adicionar um Scanner ou um Adapter conforme mostrado na figura a seguir.

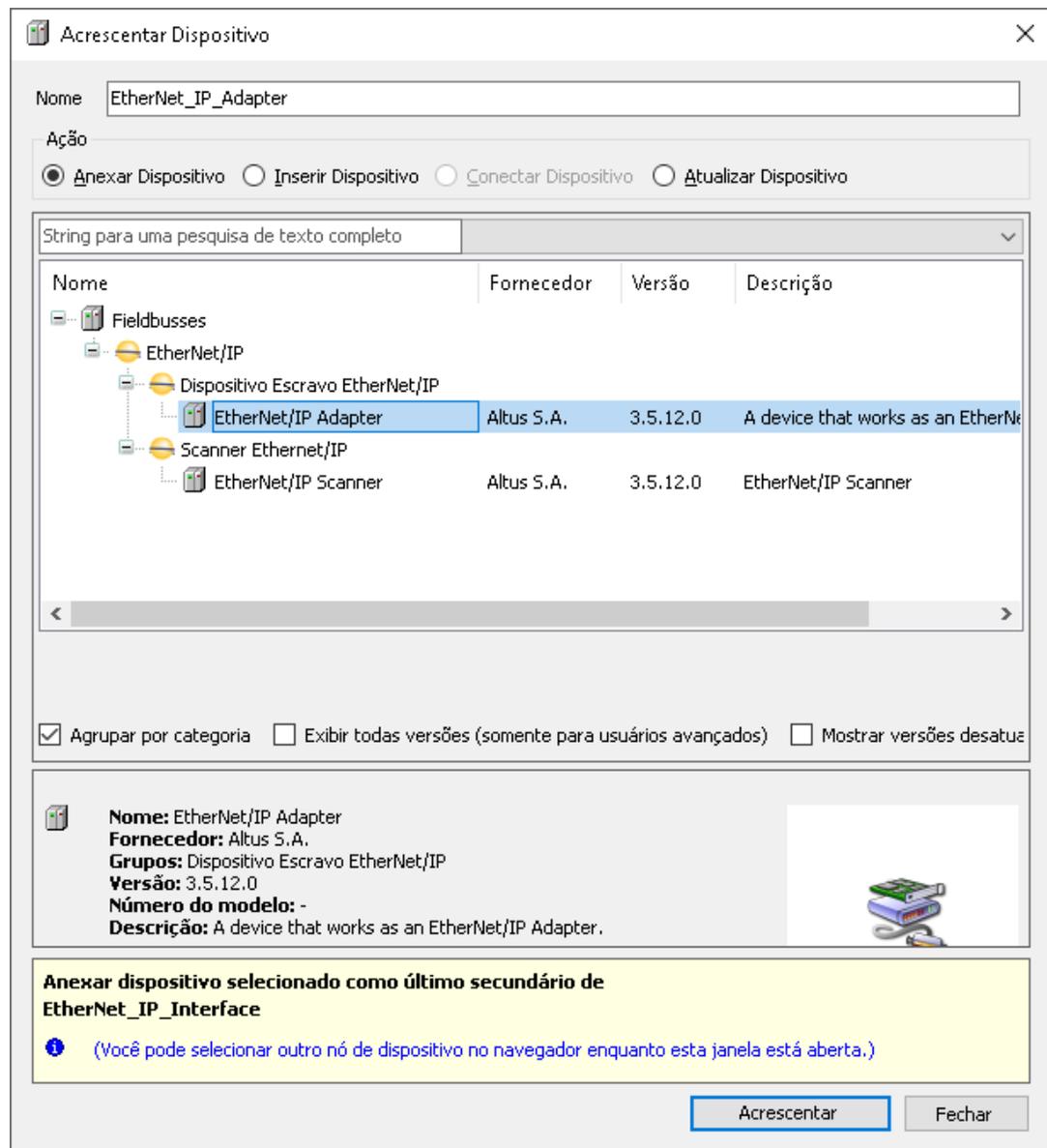


Figura 4-69. Adicionando um Adapter ou Scanner EtherNet/IP

Configuração do Scanner EtherNet/IP

O Scanner requer ao menos um Adapter declarado, com o qual ele vai trocar dados. Novos Adapters podem ser instalados no MasterTool com os arquivos EDS e DCF. As opções de configuração podem divergir dependendo do arquivo de descrição do dispositivo do Adapter adicionado.

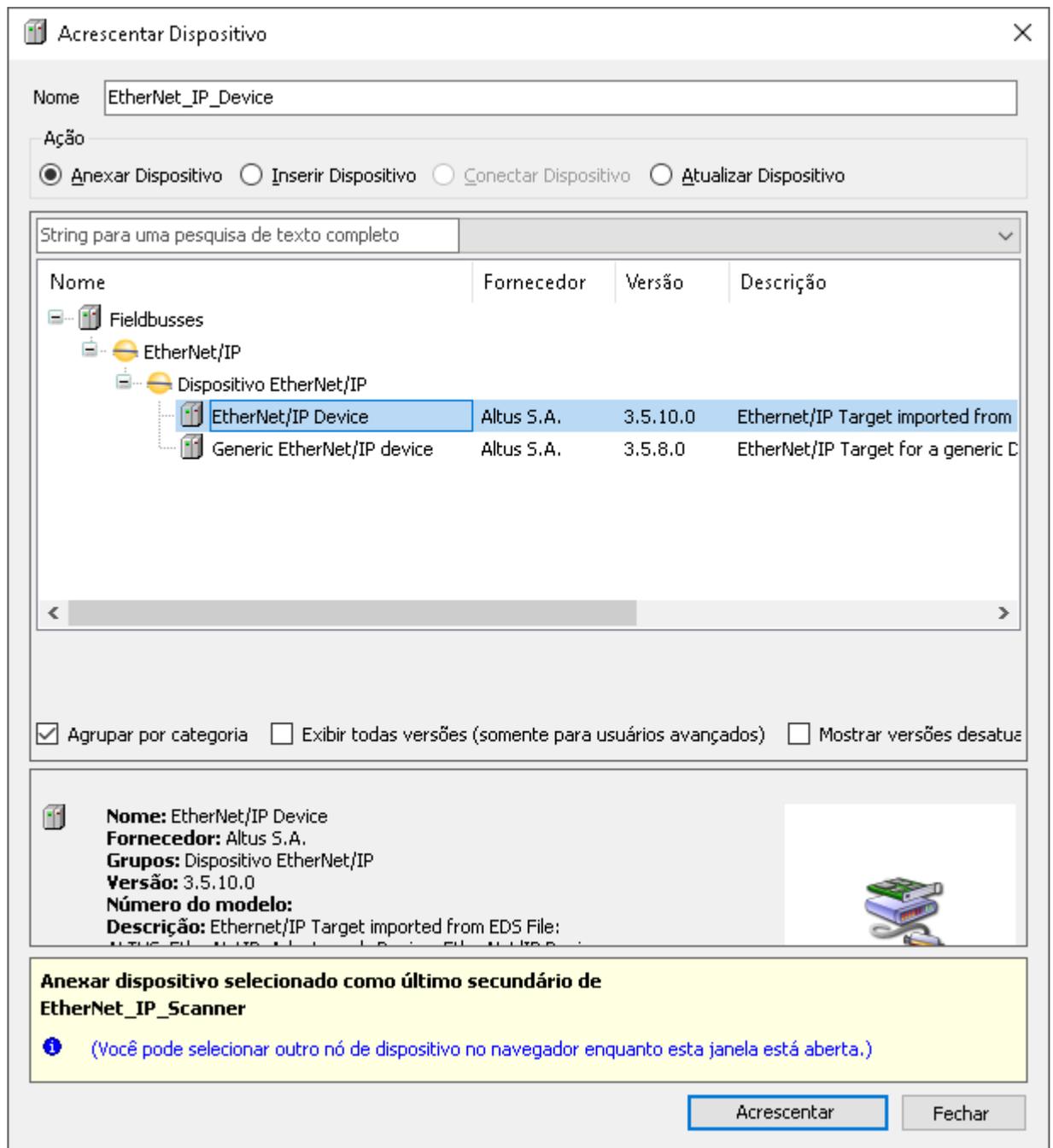


Figura 4-70. Adicionando um Adapter EtherNet/IP sob o Scanner

Depois de abrir o Adapter declarado sob o Scanner é possível configurá-lo. As opções estão indicadas na sequência.

- **Geral:** nesta aba 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 Verificar Tipo do Dispositivo, Verificar Código do Fornecedor, Verificar Código do Produto, Verificar Revisão Principal e Verificar Revisão Secundária devem ser preenchidos com a informação correta e as caixas marcadas tanto quanto necessário. A verificação pode ser alternada entre Verificar Compatibilidade e Verificar Identidade Estrita.

- **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 abaixo nesta janela.
- **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.
- **EtherNet/IP: Mapeamento de E/S:** a 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. A opção Sempre Atualizar Variáveis deve ser mantida com o valor padrão Ativado 1.

Servidor IEC 60870-5-104

Este protocolo está disponível para a UCP XP340, no seu canal Ethernet.

Ao selecionar esta opção no MasterTool, a UCP passa a ser servidor da comunicação IEC 60870-5-104, permitindo a conexão com até três dispositivos clientes IEC 60870-5-104. Para cada cliente, o driver possui uma fila de eventos exclusiva com as seguintes características:

- Tamanho: 1000 eventos;
- Retentividade: não é retentiva;
- Política de estouro: mantém o mais recente.

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

1. Adicionar a instância do protocolo Servidor IEC 60870-5-104 ao canal Ethernet (NET 1);
2. Configurar a interface Ethernet;
3. Configurar os parâmetros gerais do protocolo Servidor IEC 60870-5-104, com o modo de conexão Porta ou IP, e o número da porta TCP quando o modo de conexão selecionado for o modo IP;
4. Adicionar e configurar dispositivos, definindo os devidos parâmetros;
5. Adicionar e configurar os mapeamentos IEC 60870-5-104, especificando o nome da variável, tipo do objeto, endereço do objeto, tamanho, faixa, banda morta, e tipo da banda morta;
6. Configurar os parâmetros da camada de enlace, especificando os endereçamentos, os timeouts de comunicação e os parâmetros de comunicação;
7. Configurar os parâmetros da camada de aplicação, configuração do sincronismo, comandos, bem como o modo de transmissão dos objetos Integrated Totals.



NOTA:

Com exceção dos objetos digitais, os tipos de dados de objetos analógicos e contadores do protocolo IEC 60870-5-104 são diferentes dos tipos de dados dos módulos analógicos e contadores PROFIBUS, não sendo possível mapear tais tipos de variáveis PROFIBUS diretamente para os clientes IEC 60870-5-104. Nestes casos é necessário criar uma variável intermediária, a ser mapeada no cliente IEC 60870-5-104, e converter os tipos adequadamente.

Pontos Duplos

Os pontos digitais duplos são utilizados para indicar a posição de equipamentos como válvulas, disjuntores e seccionadoras, onde a transição entre os estados aberto e fechado demandam um

determinado tempo, podendo assim indicar um estado intermediário de transição entre os dois estados finais.

Pontos digitais duplos também são utilizados como saídas e, de uma forma análoga, é necessário manter uma das saídas acionada por determinado tempo para a transição ser completada. Tal acionamento é realizado através de pulsos, também conhecido por comandos trip/close, com determinada duração (suficiente para o chaveamento do dispositivo sob controle).

Uma vez que as entradas e saídas digitais do Nexto Xpress não suportam o mapeamento de pontos DBP, são necessárias algumas artimanhas de aplicação para tornar isto possível. Lembrando que também não é possível a utilização da função PulsedCommand, definido na biblioteca LibRtuStandard, para acionamento dos pontos digitais duplos na Série Nexto.

A figura a seguir ilustra a tela de Parâmetros Gerais do Servidor IEC 60870-5-104.

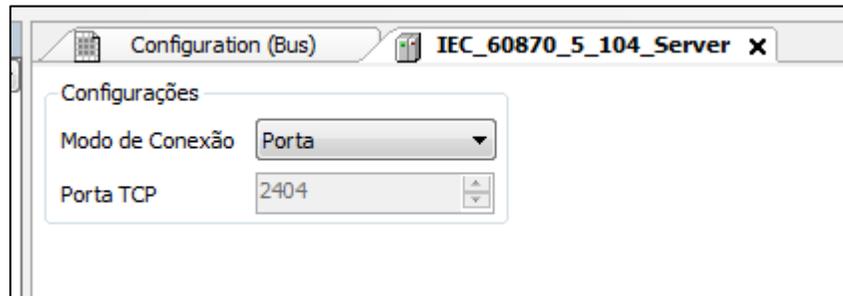


Figura 4-71. Tela de Parâmetros Gerais do Servidor IEC 60870-5-104

Na sequência está mostrada a tela de Mapeamento dos Dados do Servidor IEC 60870-5-104.



Figura 4-72. Tela de Mapeamento dos Dados do Servidor IEC 60870-5-104

A tela de configuração da camada de enlace do Servidor IEC 60870-5-104 está mostrada a seguir.

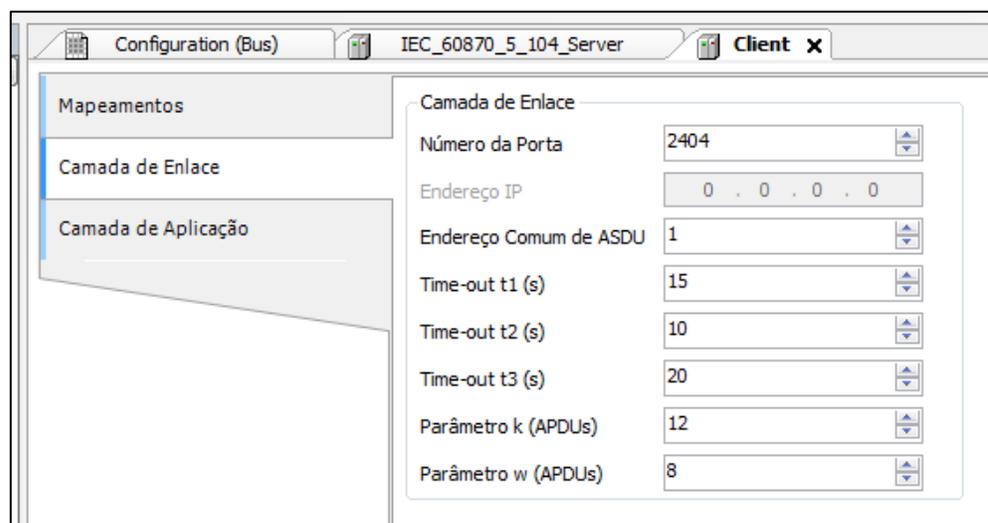


Figura 4-73. Tela de Configuração da Camada de Enlace do Servidor IEC 60870-5-104

A tela de configuração da camada de aplicação do Servidor IEC 60870-5-104 está mostrada a seguir.

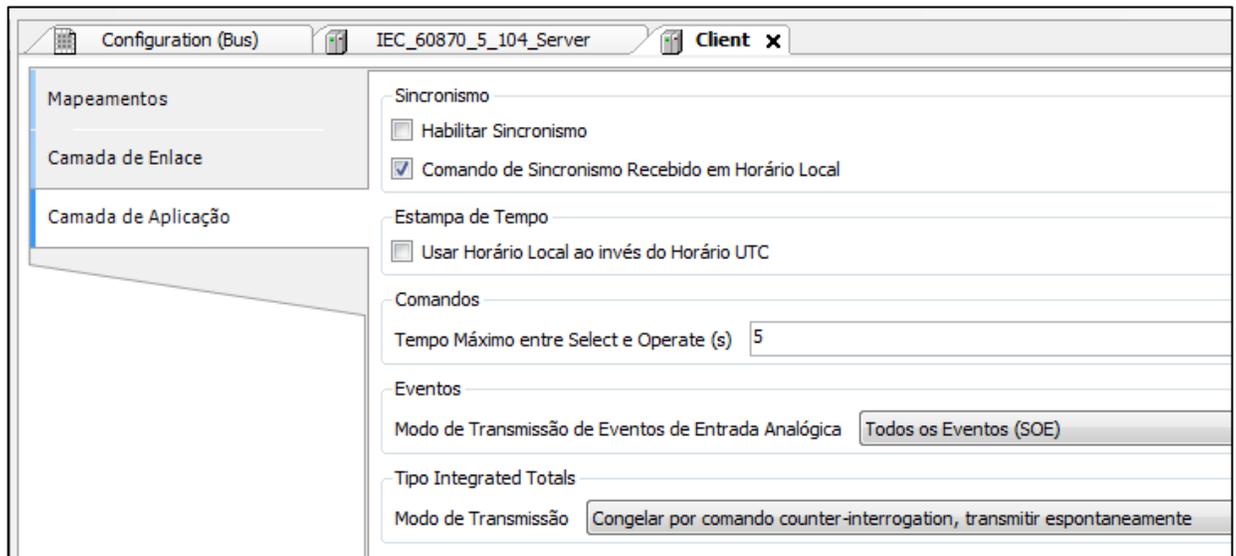


Figura 4-74. Tela de Configuração da Camada de Aplicação do Servidor IEC 60870-5-104

Os diagnósticos do protocolo Servidor IEC 60870-5-104, são armazenados em variáveis do tipo T_DIAG_IEC104_SERVER_1.

A norma IEC 60870-5-104 prevê quatro diferentes qualificadores de comandos para os objetos Single Command, Double Command e Regulating Step Command, todos suportados pelo Servidor do Nexto Xpress. Cada tipo de objeto tem um comportamento específico para cada qualificador de comando.



Pesquisa na Documentação: protocolo IEC 60870-5-104

Consulte, na documentação do produto, o detalhamento das configurações do protocolo IEC 60870-5-104.

Fonte: Manual de Utilização UCPs Série Nexto NX30xx – MU214100

CANOpen Manager

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

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

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

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

ATENÇÃO:

- Somente uma instância do CANOpen Manager por projeto é permitida
- Embora a especificação CANOpen permita até 127 nós (incluindo o Manager), as aplicações com o Nexto Xpress não devem exceder um máximo de 64 dispositivos escravos.

Um cuidado especial deve ser tomado considerando o comprimento do barramento físico e a taxa de transmissão selecionada.

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

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

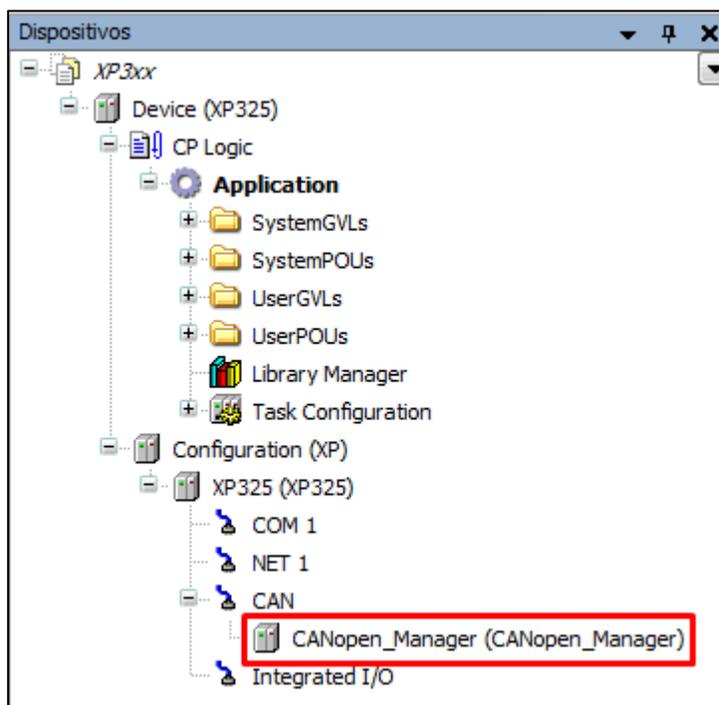


Figura 4-75. Adicionando o CANopen Manager

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

Depois disso, clique com o botão direito do mouse no dispositivo CANopen_Manager e clique em Acrescentar Dispositivo.

Pesquise os dispositivos que você deseja e clique no botão Acrescentar, como mostrado na figura a seguir.

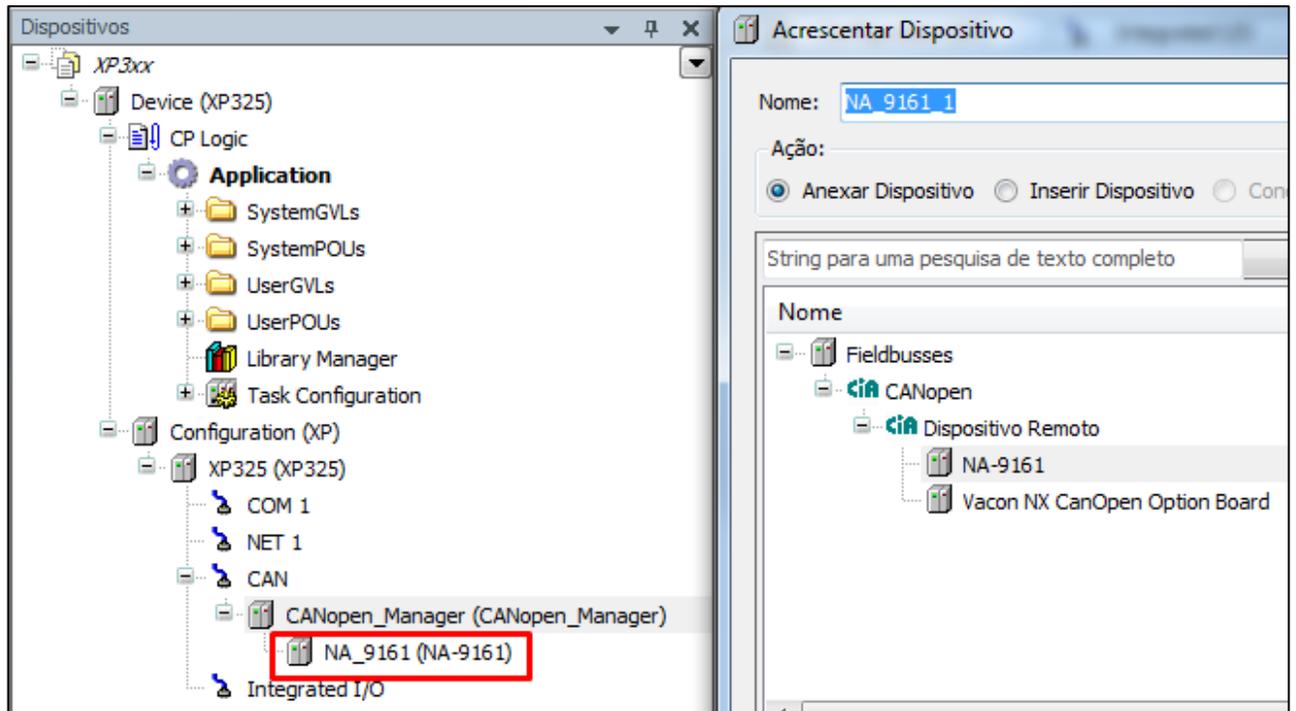


Figura 4-76. Adicionando um Dispositivo CANopen Escravo

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

Os principais parâmetros do CANOpen Manager estão localizados na guia General conforme ilustrado na sequência.

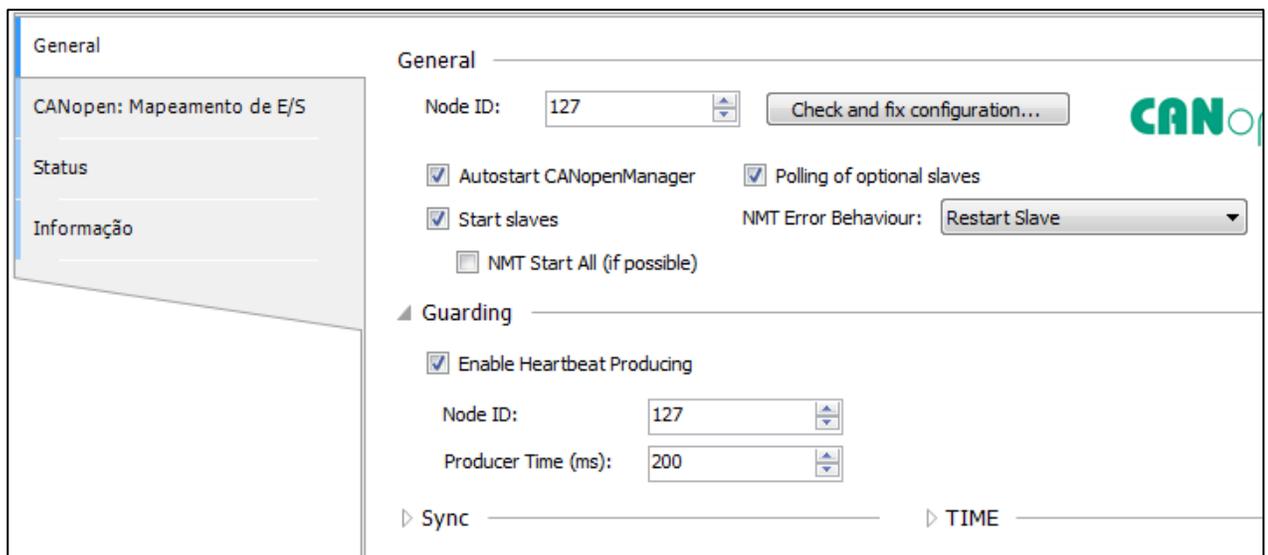


Figura 4-77. Parâmetros gerais do CANopen Manager

A descrição detalhada dos parâmetros gerais do CANOpen Manager pode ser encontrada na seção Editores de Dispositivos → CANopen da Ajuda Online do MasterTool IEC XE (F1).

Além disso, a guia CANopen: Mapeamento de E/S permite alterar a tarefa do ciclo de barramento de acordo com o mostrado na figura a seguir.

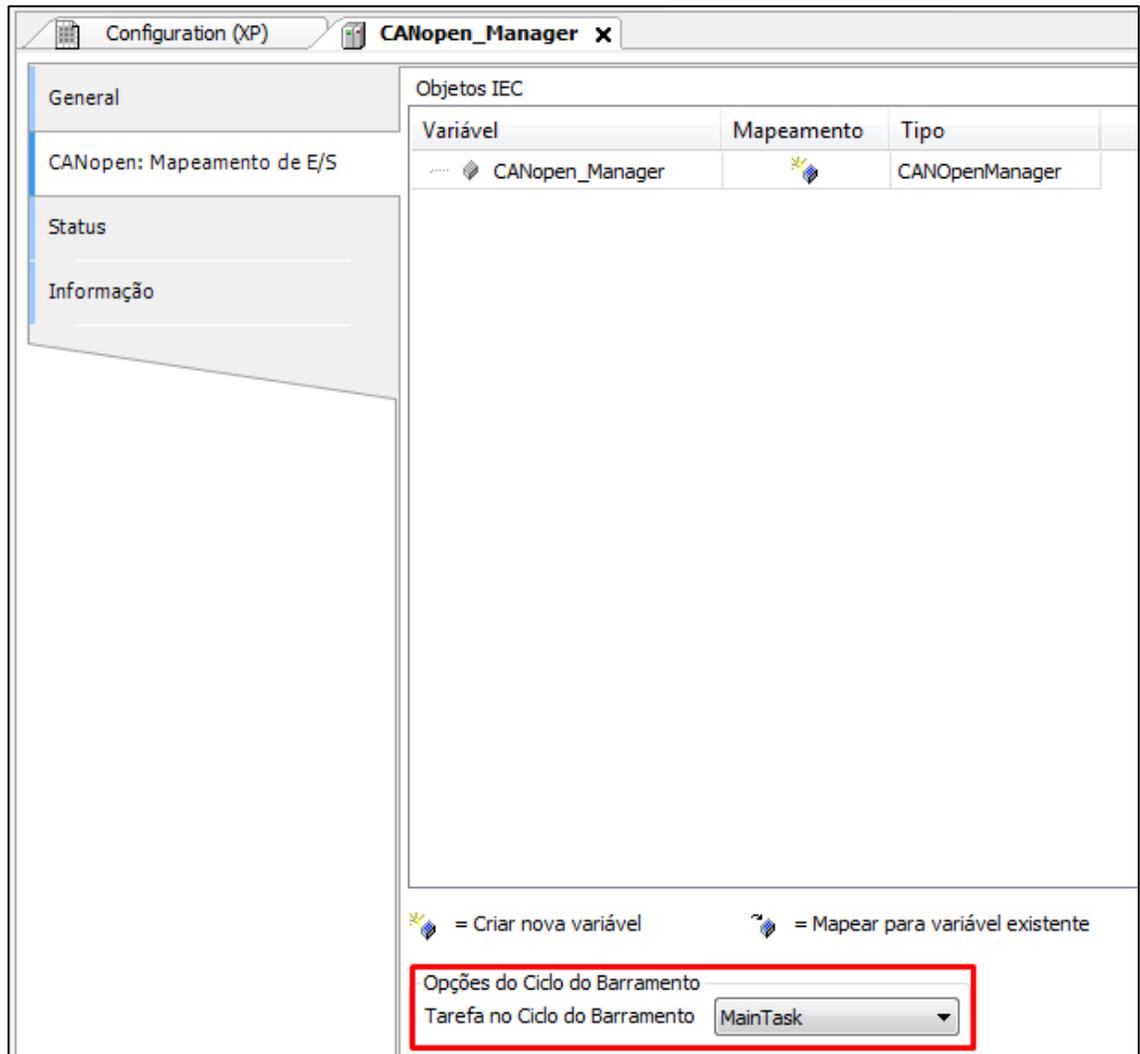


Figura 4-78. Configuração da tarefa cíclica do barramento CANopen

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

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

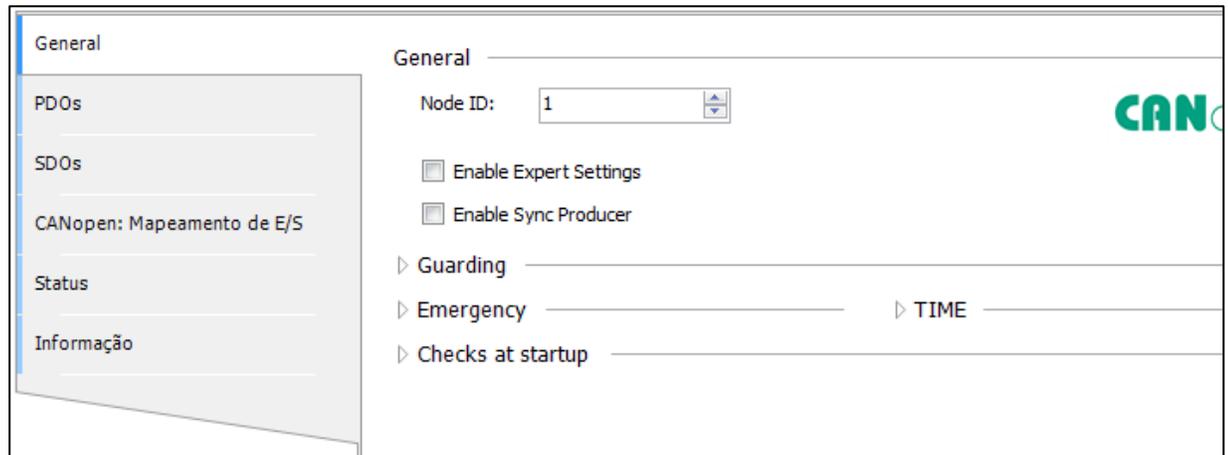


Figura 4-79. Parâmetros de Escravo CANopen

A guia General contém o endereço do escravo (Node ID), Nodeguarding e Emergency. A guia PDO contém a configuração dos dados do processo (dados de E / S) que serão trocados.

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

A descrição detalhada dos parâmetros do Escravo CANopen pode ser encontrada na seção Editores de Dispositivos → CANopen da Ajuda Online do MasterTool IEC XE (F1).



Pesquisa na Documentação: CANopen Manager

Consulte, na documentação do produto, o detalhamento das configurações do protocolo CANopen (CANopen manager) no que se refere ao Baudrate versus Comprimento do Barramento.

Fonte: Manual de Utilização Nexto Xpress – MU216000

Modo de E/S Remotas

Os controladores Nexto Xpress dispõem de um modo de operação, utilizado para expansão do número de pontos de E/S.

Esta expansão é realizada com base no protocolo CANopen, onde o controlador deixa de operar como um CP comum e atua como um escravo remoto. Para configurar o seu Xpress como uma expansão de E/S remotas, utilize a página Web, na guia de Gerenciamento, em Modo de Operação.

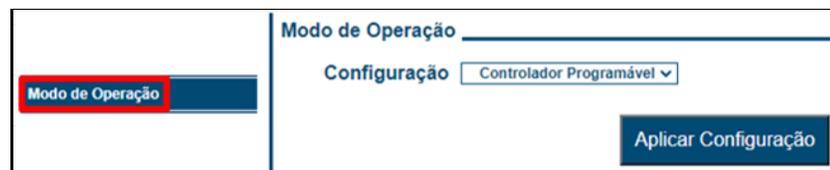


Figura 4-80. Tela de Configuração Modo de Operação

Nesta página, é possível selecionar o modo de operação do controlador por meio do combo box, quando o mesmo se encontra em STOP. Utilize o botão Aplicar Configuração para realizar a troca para o modo desejado, onde o Xpress será reiniciado, a fim de configurá-lo no novo modo de operação. As opções disponíveis para configuração são:

- Controlador Programável: função padrão do controlador, podendo ser programado conforme as necessidades do usuário.

- Escravo CANopen: função de expansão de E/S remotas, onde o controlador torna-se um escravo CANopen, podendo ser comandado por outros controladores com CANopen Manager.

O modo de operação remoto utiliza uma aplicação desenvolvida apenas para fins de expansão de E/S, que executa com um ciclo de MainTask de 5 ms, não sendo possível alterá-la ou realizar a carga de uma nova aplicação durante sua execução.

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

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

The screenshot shows a web interface titled 'Modo de Operação'. At the top, there is a dropdown menu set to 'Escravo CANopen' and three buttons: 'Baixar EDS', 'Importar', and 'Exportar'. Below this, the page is divided into three main sections:

- Configuração da Rede:** Contains three input fields: 'Endereço IP' (192.168.15.1), 'Máscara de Rede' (255.255.255.0), and 'Gateway' (192.168.15.253).
- Configuração do Escravo CANopen:** Contains three fields: 'Node ID' (1), 'Baudrate (kbits/s)' (250), and 'Terminação' (Desabilitado).
- Configuração de E/S:** A list of expandable sections: 'Entradas Digitais', 'Entradas Analógicas', 'Entradas RTD', and 'Saídas Analógicas', each with a '+' icon on the right.

Figura 4-81. Tela de Configuração do Escravo CANopen

Clique nos itens com um + à direita para expandir a janela de configuração. Os parâmetros apresentados na Configuração de E/S são os mesmos das E/S Integradas. Enquanto os parâmetros do Escravo CANopen são semelhantes aos do CANopen Manager conforme mostrado na figura a seguir.

Modo de Operação					
Entradas Analógicas					
Entrada AI0	Não Configurada	Circuito Aberto	Desabilitado	Filtro Digital	Desabilitado
Entrada AI1	Não Configurada	Circuito Aberto	Desabilitado	Filtro Digital	Desabilitado
Entrada AI2	Não Configurada	Circuito Aberto	Desabilitado	Filtro Digital	Desabilitado
Entrada AI3	Não Configurada	Circuito Aberto	Desabilitado	Filtro Digital	Desabilitado
Entrada AI4	Não Configurada	Circuito Aberto	Desabilitado	Filtro Digital	Desabilitado
Entradas RTD					
Unidade de Temperatura	Graus Celsius				
Entrada RI0	Não Configurada	Filtro Digital	Desabilitado		
Entrada RI1	Não Configurada	Filtro Digital	Desabilitado		
Saídas Analógicas					
Saída AO0	Não Configurada				
Saída AO1	Não Configurada				
Saída AO2	Não Configurada				
Saída AO3	Não Configurada				
Aplicar Configuração					

Figura 4-82. Tela de Configuração de E/S Expandida

Após as configurações realizadas, é possível utilizar o botão Exportar Configuração para realizar o download de um arquivo chamado *WebRemoteConfiguration.config*, contendo os parâmetros configurados na tela. Posteriormente, este arquivo pode ser carregado utilizando-se o botão Importar Configuração. Além disso, é possível fazer o download do arquivo Electronic Data Sheet (EDS) do Escravo CANopen Remoto diretamente pela Web, utilizando o botão Baixar EDS.

Concluída a etapa de configuração, clique em Aplicar Configuração, o que fará o controlador se reiniciar. A página Web será automaticamente atualizada para o IP configurado, onde será possível ver que o Modo de Operação está como Escravo CANopen, na guia de Informações do CP, confirmando a troca do modo de operação (veja na figura a seguir).

Modelo	XP340
Descrição	Nexto PLC
Versão de Firmware	1.9.7.0
Versão do Bootloader	1.0.0.15
Modo de Operação	Escravo CANopen
Estado da Aplicação	Stop (Parada)
Diagnósticos Ativos	0
Valores Forçados	Não
Dispositivo USB	Não encontrado

Figura 4-83. Modo de Operação na Tela de Informações do CP

Com isto, é possível utilizar um controlador com a funcionalidade CANopen Manager (por exemplo, o XP340), a fim de acessar os pontos de E/S no Escravo CANopen. Para isto, realize os mesmos passos descritos na seção CANopen Manager deste tutorial.

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

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

Como o Escravo CANopen não é acessível ao usuário pelo MasterTool, o estado RUN e STOP da aplicação é controlado pelo estado de operação do escravo CANopen. Para colocar o Escravo CANopen em RUN, é necessário que o mesmo esteja em estado Operational (marcado pelo

símbolo verde ao lado do dispositivo). Para colocá-lo em STOP, utilize o Bloco Funcional NMT da biblioteca CiA405 - consulte a Ajuda On-line (F1), se necessário - para trocar o estado de operação do escravo CANopen (recomendado), ou remova o conector CAN do controlador em modo remoto. O LED do CAN pode permanecer piscando, pois ele indica a transmissão e recepção de mensagens CAN, não o estado de operação do protocolo CANopen. As figuras a seguir ilustram essa configuração.

Aba informações Gerais (Página web):	
Modelo	XP340
Descrição	Nexto PLC
Versão de Firmware	1.9.7.0
Versão do Bootloader	1.0.0.15
Modo de Operação	Escravo CANopen
Estado da Aplicação	Run (Executando)
Diagnósticos Ativos	0
Valores Forçados	Não
Dispositivo USB	Não encontrado

Figura 4-84. Escravo CANopen em Operacional - RUN

Aba informações Gerais (Página web):	
Modelo	XP340
Descrição	Nexto PLC
Versão de Firmware	1.9.7.0
Versão do Bootloader	1.0.0.15
Modo de Operação	Escravo CANop
Estado da Aplicação	Stop (Parada)
Diagnósticos Ativos	0
Valores Forçados	Não
Dispositivo USB	Não encontrado

Figura 4-85. Escravo CANopen em Pre-Operacional - STOP

Pesquisa na Documentação: modo de E/S Remotas

Consulte, na documentação do produto, o detalhamento das configurações do modo de E/S Remotas no que se refere à Organização dos PDOs do Escravo CANopen, Diagnósticos da AIX/RIX/AOX.

Fonte: Manual de Utilização Nexto Xpress – MU216000



SNMP

SNMP (Simple Network Management Protocol) é um protocolo amplamente utilizado pelos administradores de rede para fornecer informações importantes e equipamentos de diagnóstico presentes em uma determinada rede Ethernet.

Esse protocolo usa o conceito de agente e gerente, no qual o gerente envia solicitações de leitura ou grava certos objetos para o agente. Por meio de uma MIB (Management Information Base), o gerente está ciente dos objetos existentes no agente e, portanto, pode fazer solicitações desses objetos, respeitando as permissões de leitura ou escrevendo as mesmas. MIB é uma coleção de informações organizadas hierarquicamente, na qual cada objeto desta árvore é chamado OID (identificador de objeto). Para todos os equipamentos com SNMP, é obrigatório o suporte ao MIB-II, que possui informações importantes para o gerenciamento de redes Ethernet.

Os controladores Nexto Xpress comportam-se como agentes na comunicação SNMP, com suporte aos protocolos SNMPv1, SNMPv2c, SNMPv3 e suportam o MIB-II, onde os objetos requeridos são descritos no RFC-1213. As informações fornecidas pelo SNMP não podem ser manipuladas ou acessadas por meio da aplicação do usuário, exigindo um gerenciador SNMP externo para executar o acesso.

Por padrão, o agente SNMP é ativado, ou seja, o serviço é inicializado no momento em que o controlador é iniciado. O acesso às informações do agente é feito através da interface Ethernet, porta TCP 161. A figura a seguir mostra um exemplo de um gerenciador SNMP lendo alguns valores.

The screenshot displays the PowerSNMP Free Manager interface. On the left, there is a tree view of 'Discovered Devices' and 'SNMP Agents'. The main area shows a table of 'Variable Watches' with columns for Agent Address, Variable (OID), and Value. Below the table is a 'Traps' section with a 'Log' button and a table with columns for Time, Sender, Originator, Enterprise/OID, Generic Trap, and Specific Trap. On the right, there is a tree view of the MIB-II structure, showing a hierarchy from '1' to '6 snmpV2'.

Agent Address	Variable (OID)	Value
v3_192.168.19.226	sysDescr (1.3.6.1.2.1.1.1)	CPU NX3030
v3_192.168.19.226	sysName (1.3.6.1.2.1.1.5)	nexto-default
v3_192.168.19.226	sysContact (1.3.6.1.2.1.1.4)	who@where
v3_192.168.19.226	snmpInPkts (1.3.6.1.2.1.11.1)	2972
v3_192.168.19.226	snmpOutPkts (1.3.6.1.2.1.11.2)	2952
v3_192.168.19.226	ipInReceives (1.3.6.1.2.1.4.3)	1128746
v3_192.168.19.226	ipInDelivers (1.3.6.1.2.1.4.9)	1015910
v3_192.168.19.226	ipInReceives (1.3.6.1.2.1.4.3)	1128679
v3_192.168.19.226	ipOutRequests (1.3.6.1.2.1.4.10)	53509
v3_192.168.19.226	tcpOutSegs (1.3.6.1.2.1.6.11)	597
v3_192.168.19.226	udpInDatagrams (1.3.6.1.2.1.7.1)	173537

Figura 4-86. Exemplo de Gerente SNMP

Para SNMPv3, em que há autenticação de usuário e senha para solicitações via protocolo SNMP, é fornecido um usuário padrão.

Se você deseja desabilitar o serviço, alterar o usuário SNMPv3 ou comunidade para SNMPv1 / v2c predefinido, você deve acessar a página da web do controlador conforme descrito na sequência.

As configurações do SNMP podem ser alteradas através da página da Web do controlador, na guia Gerenciamento do CP. Para acessar as configurações, você deve primeiro fazer o login, como mostrado na figura a seguir.

Figura 4-87. Tela de Login - SNMP

Após o login bem-sucedido, o estado atual do serviço (ativado ou desativado), bem como as informações do usuário SNMPv3 e comunidades para SNMPv1 / v2c, podem ser visualizados.

O usuário pode ativar ou desativar o serviço através de uma caixa de seleção na parte superior da tela.

Também é possível alterar as informações do SNMPv3 clicando no botão Alterar abaixo das informações do usuário. Será aberto um formulário no qual você deverá preencher o nome de usuário e a senha antigos e o novo nome de usuário e senha.

As outras informações do usuário SNMPv3 não podem ser alteradas.

Para alterar os dados das comunidades SNMPv1 / v2c, o processo é semelhante, basta clicar no botão Alterar abaixo das informações das comunidades. Uma nova tela será aberta onde os novos dados para os campos de rocommunity e rwcommunity serão inseridos. Se você falhar em algum dos campos em branco, a comunidade deles será desativada. Dessa forma, se o usuário deixar os dois campos em branco, o acesso ao agente SNMP só será possível por meio do SNMPv3.

Se o usuário quiser retornar às configurações padrão, deve-se reconfigurar manualmente de acordo com a seção Usuário e Comunidades SNMP. Portanto, todas as configurações atuais do SNMP serão mantidas no processo de atualização do firmware. Essas opções são mostradas na figura a seguir.

Figura 4-88. Tela de Configuração de Estados SNMP

ATENÇÃO:

O usuário e a senha para efetuar login na página da web de configurações SNMP e para acessar o agente por meio do protocolo SNMP são os mesmos.

Para acessar o SNMPv1/v2c dos controladores Nexto Xpress, existem duas comunidades. É possível acessar o SNMP v3 usando o usuário padrão. Para todas as configurações de comunidades, usuário e senha, alguns limites devem ser respeitados.

**Pesquisa na Documentação: SNMP**

Consulte, na documentação do produto, o detalhamento das configurações do protocolo SNMP no que se refere aos Objetos MIB II – Agente SNMP Nexto Xpress, Informação Padrão das Comunidades SNMP v1/v2c, Informação do usuário SNMP v3 e Configuração de Limites SNMP.

Fonte: Manual de Utilização UCPs Série Nexto NX30xx – MU214100

PROFINET

No âmbito dos protocolos de comunicação, a opção PROFINET Controller está disponível na interface NET1 permitindo a implementação de redes simples (sem anel) com até oito dispositivos. Para a implementação de aplicações maiores, o suporte técnico deve ser consultado.

Relógio RTC

As UCPs da Série Nexto 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.

**Pesquisa na Documentação: Relógio RTC**

Consulte, na documentação do produto, os elementos de software (funções e blocos funcionais) que são 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. Além disso, identifique as estruturas de dados utilizadas nessas configurações (EXTENDED_DATE_AND_TIME, DAYS_OF_WEEK, RTC_STATUS e TIMEZONESETTINGS).

- GetDateAndTime: retorna o valor de data e hora do RTC, assim como o eventual estado de erro da função.

```

GetDateAndTime
DATEANDTIME EXTENDED_DATE_AND_TIME RTC_STATUS GetDateAndTime

```

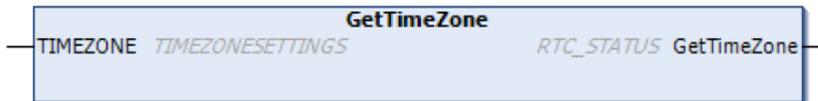
O trecho de código, na linguagem ST (Texto Estruturado), mostrado a seguir exemplifica o uso desta função.

```

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

```

- **GetTimeZone**: 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.



O trecho de código, na linguagem ST (Texto Estruturado), mostrado a seguir exemplifica o uso desta função.

```

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

```

- **GetDayOfWeek**: é utilizada para fazer a leitura do dia da semana. Quando chamada, a função lerá o dia da semana e preencherá a estrutura DAYS_OF_WEEK.



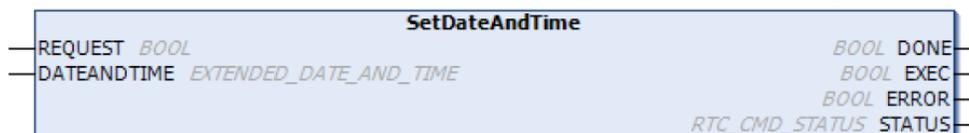
O trecho de código, na linguagem ST (Texto Estruturado), mostrado a seguir exemplifica o uso desta função.

```

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

```

- **SetDateAndTime**: este bloco funcional é utilizado para realizar o ajuste do relógio. Tipicamente a precisão é da ordem de centenas de milissegundos.



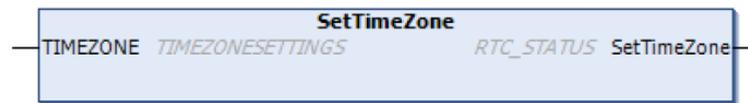
O trecho de código, na linguagem ST (Texto Estruturado), mostrado a seguir exemplifica o uso deste bloco funcional.

```

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

```

- SetTimeZone: esta função realiza a escrita do ajuste de fuso horário.



O trecho de código, na linguagem ST (Texto Estruturado), mostrado a seguir exemplifica o uso desta função.

```

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

```

Fonte: Manual de Utilização Nexto Xpress – MU216000

Blocos Funcionais Especiais para Comunicação Serial

Os blocos funcionais especiais para comunicação serial possibilitam o acesso às portas seriais locais (COM), bem como o acesso remoto (módulos de expansão). Dessa forma, o usuário poderá criar seus próprios protocolos e manipular as portas seriais como quiser, seguindo as linguagens da IEC 61131-3 disponíveis no software MasterTool IEC XE. Os blocos estão disponíveis dentro da biblioteca NextoSerial, a qual deve ser adicionada ao projeto para que seja possível utilizá-los.

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.



Pesquisa na Documentação: Blocos Funcionais para Comunicação Serial

Consulte, na documentação do produto, os elementos de software que são utilizados para configuração dos canais seriais, bem como os tipos de dados utilizados por esses blocos (Datatypes).

- 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.



O trecho de código, na linguagem ST (Texto Estruturado), mostrado a seguir exemplifica o uso deste bloco.

```
PROGRAM UserPrg
VAR
Config: SERIAL_CFG;
Port: SERIAL_PORT := COM1;
Parameters: SERIAL_PARAMETERS := (BAUDRATE := BAUD9600,
DATABITS := DATABITS_8,
STOPBITS := STOPBITS_1,
PARITY := PARITY_NONE,
HANDSHAKE := RS232_RTS,
UART_RX_THRESHOLD := 8,
MODE :=NORMAL_MODE,
ENABLE_RX_ON_TX := FALSE,
ENABLE_DCD_EVENT := FALSE,
ENABLE_CTS_EVENT := FALSE);
Status: SERIAL_STATUS;
END_VAR

//ENTRADAS:
Config.REQUEST := TRUE;
Config.PORT := Port;
Config.PARAMETERS := Parameters;
//FUNÇÃO:
Config();
//SAÍDAS:
Config.DONE;
Config.EXEC;
Config.ERROR;
Status := Config.STATUS; //Caso seja necessário tratar o erro.
```

- SERIAL_GET_CFG: esse bloco funcional é utilizado para capturar as configurações da porta serial desejada.



O trecho de código, na linguagem ST (Texto Estruturado), mostrado a seguir exemplifica o uso deste bloco.

```
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.
```

- 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.



O trecho de código, na linguagem ST (Texto Estruturado), mostrado a seguir exemplifica o uso deste bloco.

```

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;

```

- 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.



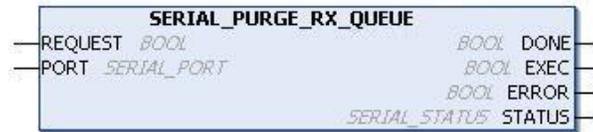
O trecho de código, na linguagem ST (Texto Estruturado), mostrado a seguir exemplifica o uso deste bloco.

```

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.

```

- 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.



O trecho de código, na linguagem ST (Texto Estruturado), mostrado a seguir exemplifica o uso deste bloco.

```

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.

```

- 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.



O trecho de código, na linguagem ST (Texto Estruturado), mostrado a seguir exemplifica o uso deste bloco.

```

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;

```

- SERIAL_RX_EXTENDED: esse bloco funcional é utilizado para receber um buffer da porta serial utilizando o modo estendido da fila RX.



O trecho de código, na linguagem ST (Texto Estruturado), mostrado a seguir exemplifica o uso deste bloco.

```

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;

```

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

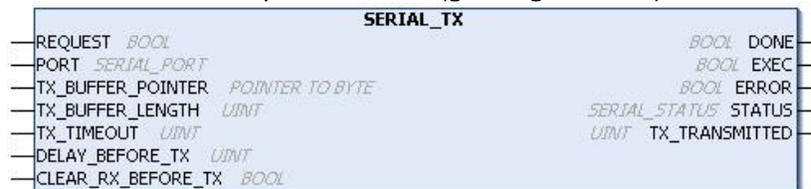


O trecho de código, na linguagem ST (Texto Estruturado), mostrado a seguir exemplifica o uso deste bloco.

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

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

- **SERIAL_TX**: esse bloco funcional é utilizado para transmitir um buffer de dados pela porta serial, sendo que ele somente é finalizado depois de todos os bytes serem transmitidos ou após o time-out (gera alguns erros).



O trecho de código, na linguagem ST (Texto Estruturado), mostrado a seguir exemplifica o uso deste bloco.

```

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;

```

Fonte: Manual de Utilização Nexto Xpress – MU216000

Atualização de Entradas e Saídas

Esta funcionalidade é utilizada para atualizar entradas e saídas no decorrer da aplicação, não sendo necessário aguardar até que seja completado um ciclo. Quando as funções para atualizar as entradas e saídas não são utilizados, a atualização é realizada a cada ciclo da MainTask. 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.

Pesquisa na Documentação: Funções de Atualização de Entradas e Saídas

Consulte, na documentação do produto, os elementos de software que são utilizados para atualização das entradas e saídas conforme indicado na sequência.

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

RefreshIntegratedIoInputs
UDINT RefreshIntegratedIoInputs

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

RefreshIntegratedIoOutputs
UDINT RefreshIntegratedIoOutputs

Fonte: Manual de Utilização Nexto Xpress – MU216000



Temporizador Retentivo

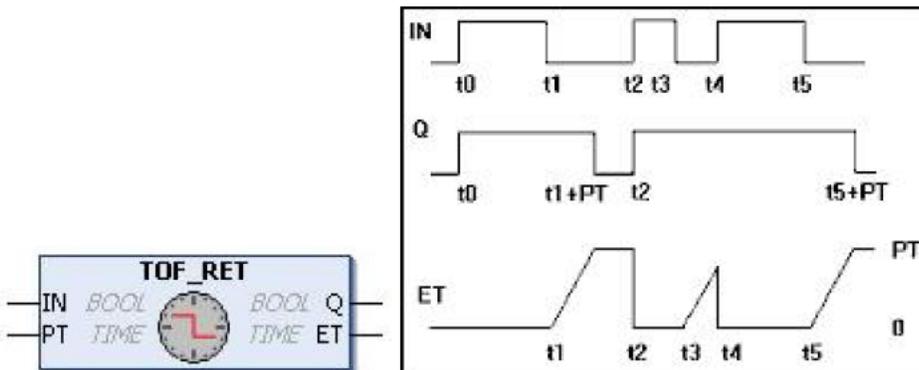
O Temporizador Retentivo é um elemento de software desenvolvido para aplicações como relógios de linha de produção, as quais necessitam armazenar o seu valor e reiniciar a contagem do mesmo ponto em caso de falha na alimentação. Os valores, guardados pelos blocos funcionais associados a essa funcionalidade somente serão zerados em caso de um Reset a Frio, Reset Origem ou Download de uma nova aplicação, sendo que os contadores continuam em funcionamento mesmo que a aplicação esteja parada (Modo Stop).

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 conectada. Os três tipos de blocos funcionais de temporização retentiva estão disponíveis na biblioteca NextoStandard.

Pesquisa na Documentação: Blocos Funcionais de Temporização Retentiva

Consulte, na documentação do produto, os elementos de software que são utilizados para temporização retentiva conforme indicado na sequência.

- 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.



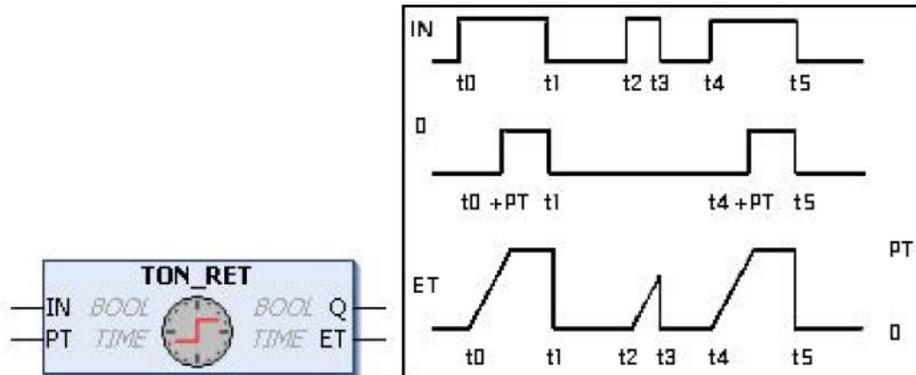
O trecho de código, na linguagem ST (Texto Estruturado), mostrado a seguir exemplifica o uso deste bloco.

```
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
```

- **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.



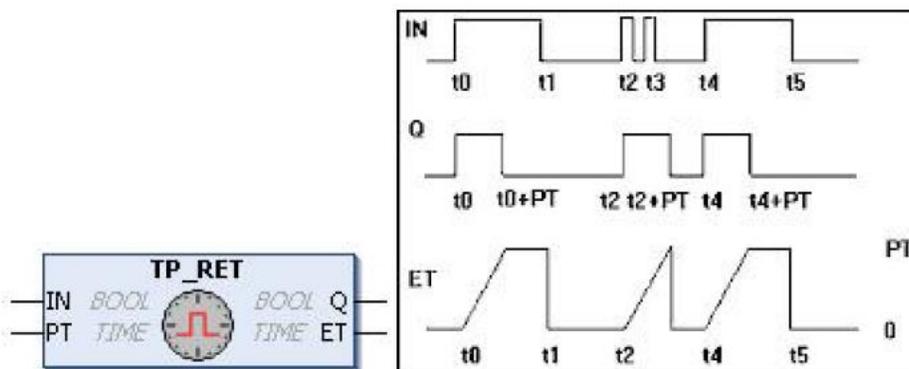
O trecho de código, na linguagem ST (Texto Estruturado), mostrado a seguir exemplifica o uso deste bloco.

```
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
```

- **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.



O trecho de código, na linguagem ST (Texto Estruturado), mostrado a seguir exemplifica o uso deste bloco.

```

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

```

Fonte: Manual de Utilização Nexto Xpress – MU216000

Servidor FTP

A principal função do FTP (File Transfer Protocol) é realizar a transferência de arquivos do CP para outro dispositivo e vice-versa. É um protocolo rápido e de fácil utilização, o qual realiza operações como download e upload de arquivos em conexões do tipo cliente/servidor. A configuração do FTP é feita através de uma seção específica localizada na aba Gerenciamento da Página Web de Sistema do controlador, conforme ilustrado na figura a seguir.

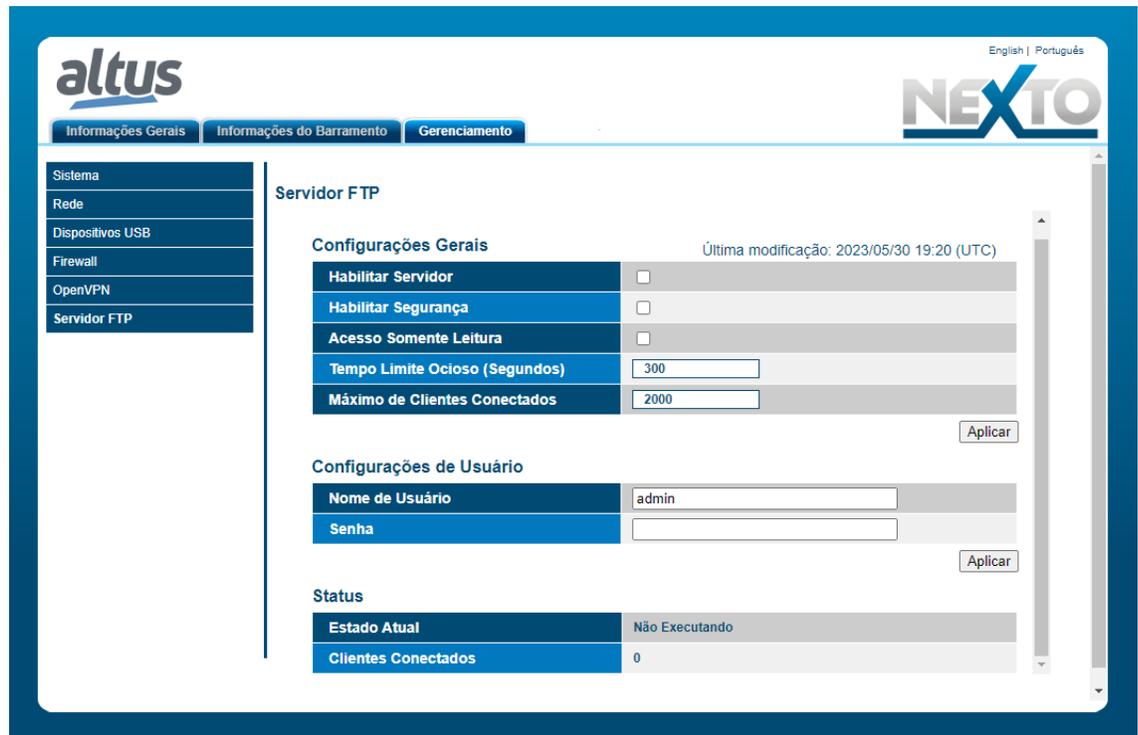


Figura 4-89. Página de Configuração do Servidor FTP

O FTP é uma funcionalidade à parte do MasterTool IEC XE, isto é, não necessita de interação alguma com a ferramenta de programação. As configurações aplicadas na Página Web de Sistema passam a valer quando confirmadas através do botão Aplicar e, são salvas automaticamente no controlador. Contudo que a funcionalidade esteja habilitada, voltará a operar mesmo após a reinicialização do dispositivo.



Pesquisa na Documentação: Configurações Gerais e de Usuário do Servidor FTP

Consulte, na documentação do produto, as configurações típicas do Servidor FTP, as quais incluem:

- **Configurações Gerais:** Habilitar Servidor, Habilitar Segurança, Acesso Somente Leitura, Tempo Limite Ocioso (segundos) e Máximo de Clientes Conectados;
- **Configurações de Usuário:** Nome de Usuário e Senha;
- **Status:** Estado Atual e Clientes Conectados.

Fonte: Manual de Utilização Nexto Xpress – MU216000

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. A configuração do Firewall é feita através de uma seção dedicada localizada na aba Gerenciamento da Página Web de Sistema do controlador, conforme mostrado na figura a seguir.

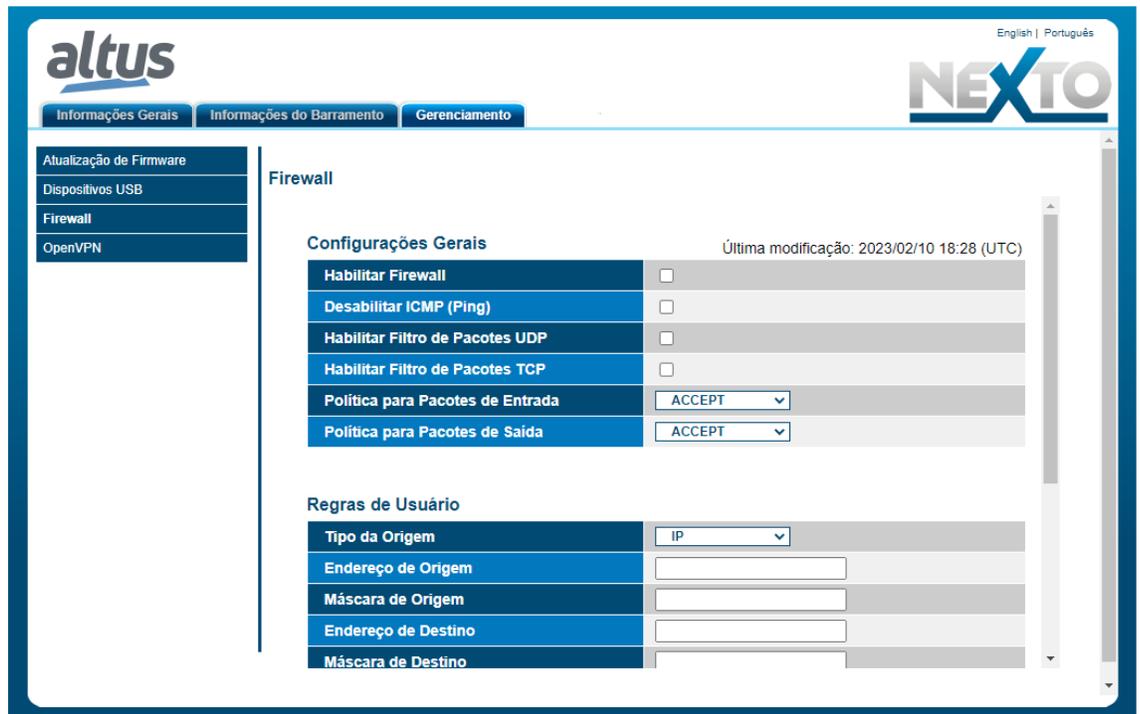


Figura 4-90. Página de Configuração do Firewall

Tabela de Configurações Gerais

A tabela disponível na parte de Configurações Gerais da Página Web do Firewall se 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 de **Habilitar Filtro de Pacotes UDP e TCP**, quando habilitados, realizam o filtro destes protocolos dentro dos limites configurados nos seus respectivos campos. A regra que realiza a filtragem dos pacotes possui o seguinte funcionamento: para que um pacote seja aceito, é necessário que existam “créditos” disponíveis, um crédito é utilizado para aceitar um pacote de dado.

A configuração do campo **Estouro de Pacotes XXX** configura o valor inicial de pacotes, ou créditos, que serão aceitos. Dessa forma, é possível configurar um limite de estouro destes pacotes, onde caso haja um fluxo muito grande de pacotes, somente serão aceitos a quantidade configurada.

O campo de **Pacotes XXX por Segundo**, configura quantos créditos aquela regra irá ganhar por segundo, ou seja, se for configurado o valor 5, a cada segundo a regra receberá cinco novos créditos, logo poderá aceitar mais cinco pacotes. A limitação para esse incremento no número de créditos é a própria configuração de Estouro de Pacotes XXX, o limite estabelecido aqui não é ultrapassado, mesmo com o incremento de pacotes a cada segundo. Estas configurações são

aplicadas como um estoque, onde ao receber um pacote de dados, primeiro é verificado se há algum crédito disponível no estoque e então é tomada a decisão de aceitar ou não o pacote. Se o pacote é aceito neste filtro de quantidade, ele é direcionado para o filtro das demais regras de Firewall.

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

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

Tabela de Regras de Usuário

A tabela Regras de Usuário, disponível logo abaixo da tabela de Configurações Gerais na Página Web do Firewall, 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.

Esta tabela altera o seu formato de acordo com o **Tipo de Origem** selecionado, podendo ser IP ou MAC. Quando o tipo é IP, a tabela possui os itens exibidos na figura acima, porém, quando é selecionado o tipo como MAC os campos de máscaras de origem e destinos desaparecem, assim como o campo de Endereço do Destino. O item Endereço de Origem passa a aceitar um endereço MAC como entrada. 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 à regra.

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

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

O campo **Porta de Serviço** serve para indicar quais as portas serão configuradas nesta regra. Todas as portas de serviços que possuem um determinado protocolo ou comunicação padrão para o controlador, como por exemplo o protocolo MODBUS que tem a porta padrão 502, estão disponíveis com o nome do serviço e a porta utilizada ao lado. Assim, caso seja configurada a

regra para o protocolo MODBUS, será aplicada a porta 502, caso seja configurada a regra para o serviço webvisu, será aplicada a porta 8080 e assim segue para os demais protocolos listados no campo de seleção. Este campo também possui outras duas configurações, que são Qualquer e Outra. Quando é selecionada a opção Qualquer, a regra é aplicada para todas as portas de serviços, exceto a porta 80, então são criadas duas regras utilizando as seguintes faixas de portas: 1:79 e 81:65535. Caso seja selecionada a opção Outra, será exibido uma caixa de texto na qual é possível configurar a porta que deseja, exceto a porta 80. Para configurar uma porta, basta escrever o seu número na caixa de texto, se desejar adicionar mais de uma única porta, é necessário utilizar o separador "&" e caso queira inserir uma faixa de portas, basta informar a porta inicial e final utilizando o separador ":".

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

120 & 144 & 1300:1450.

Este campo não aceita valores fora da faixa 1:65535, a porta 80 ou repetições de portas. A porta HTTP, 80, somente pode ser configurada através da seleção na lista de protocolos conhecidos e não pode ser aplicada a interface NET1. Sendo assim, caso o protocolo HTTP seja escolhido, os campos NET1 e Qualquer do campo de Interface não serão possíveis de serem selecionados. No campo de Protocolo é possível selecionar entre os protocolos UDP, TCP e UDP|TCP. Caso seja selecionado a opção UDP|TCP, serão criadas duas regras no Firewall, uma para cada protocolo de transporte. No campo de Direção é possível selecionar entre INPUT, OUTPUT e INPUT|OUTPUT. Estas opções fazem com que a regra seja aplicada para os pacotes que estão chegando no dispositivo, opção de INPUT, ou que estão saindo do dispositivo, opção de OUTPUT. Caso seja configurado a opção conjunta, serão criadas duas regras, uma com cada opção de direção. A figura a seguir demonstra como é feita a aplicação de uma regra.

Tipo da Origem	IP
Endereço de Origem	192.168.18.120
Máscara de Origem	255.255.255.0
Endereço de Destino	192.168.18.17
Máscara de Destino	255.255.248.0
Interface	Qualquer
Ação	ACCEPT
Porta do Serviço	WebVisu - 8080
Protocolo	UDP TCP
Direção	INPUT

[Adicionar à lista](#)

ID	Ação	Regras de Usuário
0	Accepta na entrada	Fonte é 192.168.18.120/24 destino é 192.168.18.17/21 na porta 8080 sobre U... X
1	Accepta na entrada	Fonte é 192.168.18.120/24 destino é 192.168.18.17/21 na porta 8080 sobre T...

[Aplicar](#)

Figura 4-91. 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.

Assim como no caso do Servidor FTP, o Firewall se constitui em uma funcionalidade à parte do MasterTool IEC XE, isto é, não necessita de interação alguma com a ferramenta de programação. As configurações aplicadas na Página Web de Sistema passam a valer quando confirmadas através do botão Aplicar e, são salvas automaticamente no controlador. Contudo que a funcionalidade esteja habilitada, voltará a operar mesmo após a reinicialização do dispositivo.

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 CPU conforme é mostrado a seguir.

Configurações Gerais	
<input type="checkbox"/> Ativado	<input type="button" value="Importar"/>
Modo	Servidor
Endereço de IP	10.8.123.0
Endereço de Máscara	255.255.255.0
Comunicação entre Clientes	Ativado
Máximos Clientes Conectados	7
Protocolo	UDP
Topologia	subnet
Nível de Logs	5
Keep Alive Ping	10
Keep Alive Timeout	30
Certificado CA	sample_server_inline_ca

Figura 4-92. Página de Configuração do OpenVPN

Por estar localizada dentro da aba Gerenciamento, o acesso a esta página é protegido por senha.

Para realizar a configuração da VPN de forma rápida e fácil no dispositivo, pode-se utilizar o botão Importar que aparece na figura anterior 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.

As figuras a seguir mostram todas as configurações permitidas nas opções Servidor e Cliente, respectivamente.

Modo	Servidor
Endereço de IP	10.8.123.0
Endereço de Máscara	255.255.255.0
Comunicação entre Clientes	Ativado
Máximos Clientes Conectados	7
Protocolo	UDP
Topologia	subnet
Nível de Logs	5
Keep Alive Ping	10
Keep Alive Timeout	30
Certificado CA	sample_server_inline_ca
Certificado de Dispositivo	sample_server_inline_device
Chave de Dispositivo	sample_server_inline_device
Chave TA	sample_server_inline_ta
Chave do TA	SHA256

Figura 4-93. Opções de Configuração do OpenVPN (Servidor)

Modo	Cliente
IP remoto	192.168.16.127
Protocolo	UDP
Nível de Logs	3
Keep Alive Ping	5
Keep Alive Timeout	20
Certificado CA	device_1 @altus_com_br
Certificado de Dispositivo	device_1 @altus_com_br
Chave de Dispositivo	device_1 @altus_com_br
Chave TA	device_1 @altus_com_br
Chave do TA	SHA256

Figura 4-94. Opções de Configuração do OpenVPN (Cliente)

Configurações Comuns (válidas para as opções Servidor e Cliente)

- **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 necessários para cada modo de operação;
- **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. A configuração do servidor e de todos os seus clientes deve ser a mesma;
- **Nível de Logs:** este campo configura qual será o nível (numérico) que o arquivo de logs receberá. A configuração varia de 0 até 5, sendo 0 o nível mais básico e 5 o nível mais

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

- **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-se desabilitar o ping entre o servidor e os clientes, configura-se o valor 0;
- **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-se desabilitar esta funcionalidade, configura-se o valor 0;
- **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;
- **Chave do TA:** no campo Chave do TA é configurada qual será o tipo de criptografia aplicada para a Chave TA. Este campo se mantém oculto até que seja selecionado algum arquivo para a chave TLS, pois, ele somente é utilizado em conjunto com esta chave. O valor padrão deste parâmetro é SHA1, porém é possível selecionar entre os seguintes valores: SHA256, SHA512 e MD5, além do padrão SHA1. É necessário que esta configuração seja igual entre os clientes e o servidor da mesma rede OpenVPN.

Configurações Exclusivas do Servidor

- **Endereço de Rede:** A faixa de IP que será utilizada para atribuir os endereços do servidor e dos clientes da rede VPN é configurado pelo servidor através da configuração dos campos Endereço de IP e Endereço de Máscara. Todos os IPs que serão atribuídos para os clientes e ao servidor serão retirados da faixa especificada. O endereço IP do servidor é sempre o endereço inicial da faixa configurada, para a atribuição de IP dos clientes, são utilizados os valores ainda disponíveis da faixa. Por exemplo, caso seja configurada uma rede com o endereço IP 10.8.12.4 e máscara 255.255.255.248, o servidor assumirá o IP 10.8.12.5 que é o primeiro endereço disponível da faixa configurada. Porém, caso seja configurada a máscara 255.255.255.0, o servidor assumirá o IP 10.8.12.1, que é o primeiro endereço disponível da faixa. Os campos de endereço IP e máscara, somente aceitam as configurações que tenham a sintaxe de um endereço de IP e endereço de máscara, respectivamente. Caso seja configurado algo fora do padrão, será exibida uma mensagem de alerta, informando que houve algum erro;

- **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;
- **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;
- **Redes Privadas:** ao selecionar o modo de operação do OpenVPN como servidor, será exibido uma tabela, normalmente oculta, que permite a configuração de redes privadas que podem estar abaixo do servidor e de cada cliente. 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 a seguir, 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. Na sequência está ilustrada essa arquitetura.

Redes Privadas				
Tipo	Common Name	Endereço	Máscara	
Servidor	server	192.168.80.0	255.255.255.0	✖
Cliente	client2	192.168.70.0	255.255.255.0	
Servidor				+

Figura 4-95. Tabela de Configurações de Redes Privadas OpenVPN

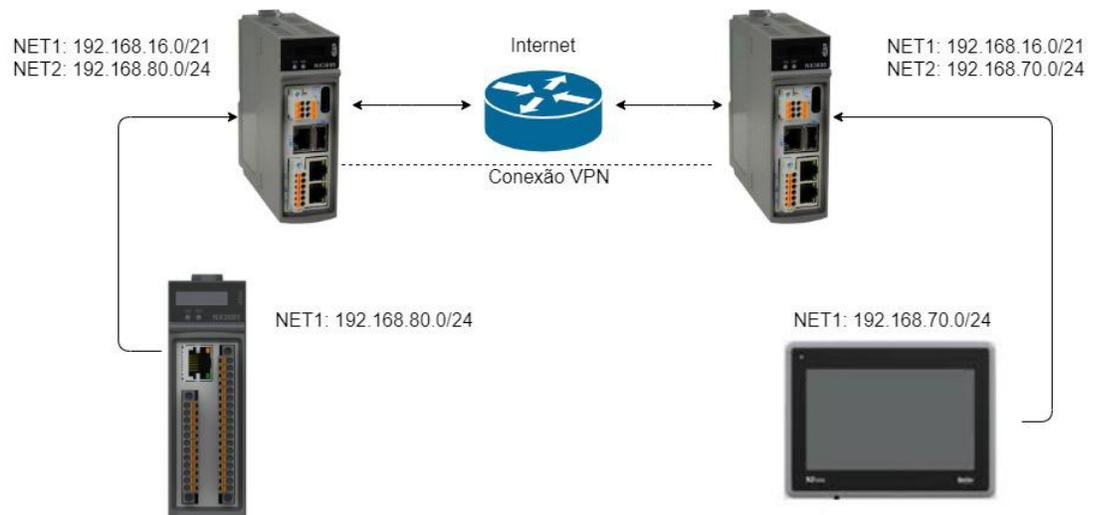


Figura 4-96. Exemplo de Arquitetura com Redes Privadas

No exemplo da figura anterior, a NX3008 da esquerda possui a rede privada 80 configurada na sua NET 2 e conectado a ela existe uma NX3003 na mesma rede. A NX3008 da direita possui uma rede privada 70 configurada na NET 2 e conectado a ela existe uma IHM, na mesma rede.

A arquitetura exemplo realiza a comunicação entre os dispositivos NX3003 e IHM pela VPN, através da configuração das suas respectivas redes privadas. Após preencher os campos de configuração, 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-se excluir alguma regra, arrasta-se o mouse sobre a regra que se deseja remover e, então, um X (vermelho) irá aparecer à direita. 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.

Configurações Exclusivas do Cliente

Existe somente uma única configuração exclusiva de clientes OpenVPN na seção associada. Trata-se do **IP Remoto**. O campo de IP Remoto, configura qual é o endereço cujo 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.

Em função da necessidade de permitir parâmetros de natureza diferente (IPs e host names), a única verificação que ocorre neste campo é sobre a existência ou não de dados. Deve-se ter atenção ao realizar a configuração.

Aplicação de Configurações

Para habilitar a funcionalidade, deve-se marcar o checkbox 'Ativado' disponível na seção. Caso se deseje apenas aplicar as configurações realizadas e não habilitar o OpenVPN, desmarca-se este checkbox. Após realizar todas as configurações desejadas, as configurações devem ser aplicadas no dispositivo, para isso utiliza-se o botão 'Aplicar' no canto inferior direito da página.

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.

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. A figura a seguir mostra a seção responsável pelo gerenciamento dos arquivos de segurança.

Arquivos de Segurança

Selecione certificados e chaves: Nenhum arquivo escolhido

Nome do Arquivo	Tipo	Common Name	Começa em (GMT)	Termina em (GMT)
device_1_bruno_ber...	Certificado	berwanger/brun...	2022/03/09 15:57	2047/03/10 15:57
device_1_bruno_ber...	Chave	-	-	-
device_1_bruno_ber...	Certificado	CloudVPN Prod ...	2019/10/16 18:04	2049/10/08 18:04
device_1_bruno_ber...	Chave	-	-	-
sample_server_inlin...	Certificado	Easy-RSA CA	2022/03/14 18:17	2032/03/11 18:17
sample_server_inlin...	Certificado	server	2022/03/14 18:21	2032/03/11 18:21
sample_server_inlin...	Chave	-	-	-
sample_server_inlin...	Chave	-	-	-

Status

Estado Atual	Executando
---------------------	------------

Figura 4-97. 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. Há um limite de importação de 12 arquivos para o controlador.

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

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

Tabela de Status

Esta tabela foi 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 do VPN atribuído àquele dispositivo, os dados trafegados e os arquivos de segurança que estão sendo utilizados na comunicação.

Status

Estado Atual	Não Executando
Common Name do CA	Easy-RSA CA
Certificado CA	sample_server_inline_ca.crt - 3536 Dia(s) Restante(s)
Common Name do Dispositivo	server
Certificado Atual do Dispositivo	sample_server_inline_device.crt - 3536 Dia(s) Restante(s)
Chave Privada Atual do Dispositivo	sample_server_inline_device.key

Figura 4-98. 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.

Status	
Estado Atual	Executando
Estado da Conexão	Conectado
Tempo de Conexão	00:00:45
Endereço IP	10.8.123.1
Número de Clientes Conectados	1
Bytes Enviados	6411
Bytes Recebidos	15007
Common Name do CA	Easy RSA CA
Certificado CA	sample_server_inline_ca.crt - 3536 Dia(s) Restante(s)
Common Name do Dispositivo	server
Certificado Atual do Dispositivo	sample_server_inline_device.crt - 3536 Dia(s) Restante(s)
Chave Privada Atual do Dispositivo	sample_server_inline_device.key

Figura 4-99. Tabela de Status OpenVPN com Funcionalidade Ativada

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

O item 'Estado Atual' possui os estados de 'Não Executando', 'Iniciando serviço...' e 'Executando', que indicam, respectivamente, que a VPN está desativada, que está iniciando, e que ela está ativada. O item 'Estado da Conexão' possui os estados de 'Não conectado', 'Conectando...' e 'Conectado'.

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

Seção de Downloads

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.

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

Figura 4-100. Seção de Downloads OpenVPN

A lista de arquivos para download somente é exibida quando há o arquivo para ser baixado, caso não exista o arquivo, o item da lista permanece oculto. Ao clicar sobre qualquer um dos links, será realizado o download do arquivo solicitado, através do navegador.

Configuração de Arquiteturas

As possibilidades de configurações para o OpenVPN incluem as arquiteturas Host-to-Host, Host-to-Site e Site-to-Site.

- **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. A figura a seguir ilustra essa topologia.



Figura 4-101. Exemplo de arquitetura Host-to-Host

- **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. 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 a seguir. Também é preciso informar o endereço da rede privada, Lan B, que realizará a comunicação através da VPN. Esta configuração é feita utilizando o comando 'push "route IP_da_Lan_B Mascara_da_Lan_B"', e é obrigatória independente se a rede privada está localizada abaixo do cliente ou do servidor do OpenVPN, porém, caso a rede privada esteja abaixo do cliente VPN, é necessário adicionar, além deste comando, a seguinte configuração: 'route IP_da_Lan_B Mascara_da_Lan_B'. O comando 'route' informa o servidor sobre qual é a rede privada que está conectada a rede da VPN e o comando utilizando o 'push' faz com que os clientes daquele servidor VPN obtenham a mesma informação. Estas configurações são escritas no arquivo de configuração do servidor VPN.

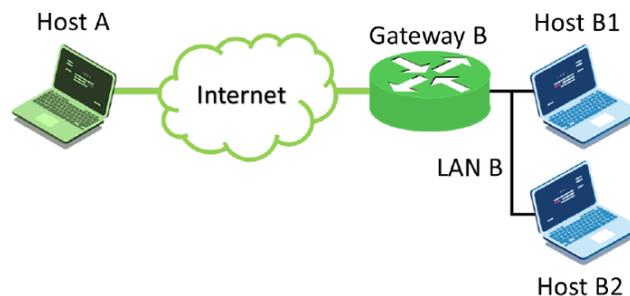


Figura 4-102. Exemplo de arquitetura Host-to-Site

- **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. As configurações para esta arquitetura precisam das mesmas configurações específicas descritas na seção Configuração 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, devem ser adicionados 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"'. A figura a seguir ilustra essa topologia.

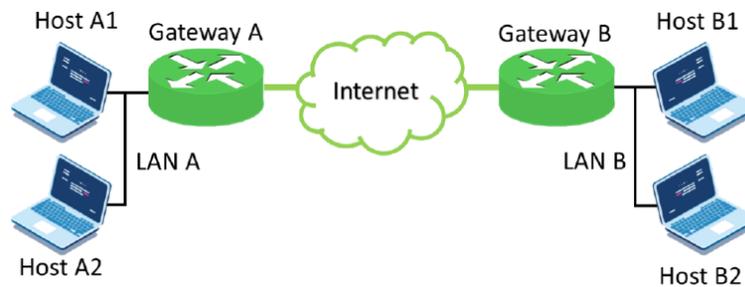
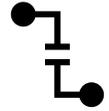


Figura 4-103. Exemplo de arquitetura Site-to-Site

5. Funcionalidades do Nexto Xpress



Este capítulo inclui a análise de algumas funcionalidades avançadas do dispositivo no que se refere às aplicações industriais, enfatizando a utilização do mesmo em sistemas de controle e explorando as potencialidades da visualização web.

Um pouco de teoria... Visão geral do controle industrial



Quase toda planta industrial necessita de algum tipo de controlador para garantir uma operação segura e economicamente viável. Todos os sistemas de controle podem ser divididos em três partes: transdutores, controladores e atuadores. A figura a seguir ilustra a relação entre essas partes.

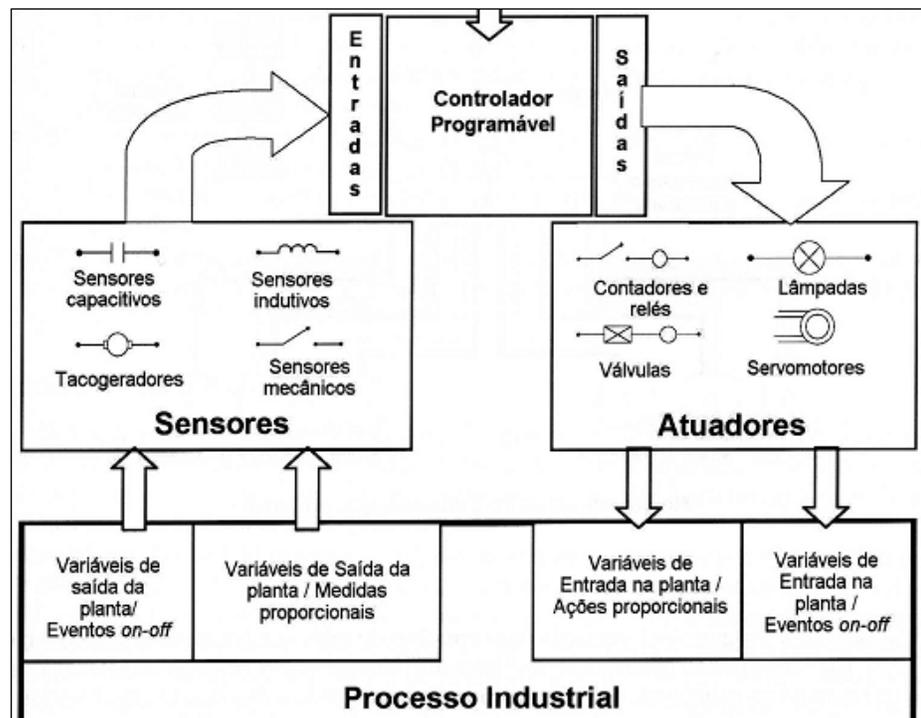


Figura 5-1. Papel do Controlador em Sistemas de Controle Industriais

O controlador monitora o estado real do processo através dos transdutores. Com base nos estados das suas entradas, o controlador utiliza um algoritmo de controle embutido para calcular os estados das suas saídas. O controlador programável (CP) manipula tanto controles digitais quanto malhas analógicas.

Os processos industriais podem ser divididos em: monitoração, sequenciamento e malha de controle.

O subsistema de monitoração mostra os estados do processo para o operador e destaca as condições anormais em que alguma ação deva ser tomada por parte do mesmo.

O sequenciamento, diferentemente da lógica combinacional usual, é baseado em uma sequência pré-definida de ações que devem ser executadas em uma determinada ordem. Para tanto, funções do tipo memória (estados e eventos) são utilizadas.

A malha de controle é útil para manipulação contínua de variáveis analógicas, as quais devem ser mantidas, automaticamente, em um determinado valor ou seguir o valor de outro sinal. A figura a seguir mostra o diagrama em blocos de uma malha de controle.

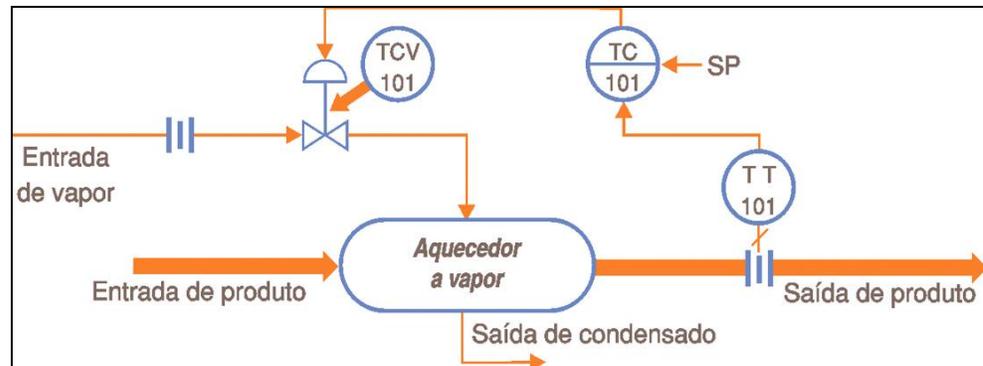


Figura 5-2. Malha de Controle

A função lógica de um sistema de controle baseado em relés é descrita no diagrama chamado de Ladder, mostrando como os contatos dos transdutores e os atuadores são interligados.

Atualmente, os CPs permitem que sejam utilizadas linguagens de programação mais estruturadas e de mais alto nível. Estas linguagens permitem que o problema de controle seja analisado e programado de forma mais intuitiva. Exemplos dessas linguagens são: o Sequenciamento Gráfico de Funções (SFC), o Diagrama de Blocos Funcionais (FBD) e o Texto Estruturado (ST).

Uma característica importante do CP é que este é projetado para trabalhar no ambiente industrial. Muitos controladores têm uma unidade de E/S modularizada, para conexão direta com os sinais dos transdutores e atuadores.

Sistemas de controle industrial são sistemas de tempo real, o que significa que alterações nos sinais de entrada exigem uma imediata ação no sinal de saída correspondente. Para garantir isso, o programa de controle deve monitorar, constantemente, os sinais de entrada provenientes do processo. Assim sendo, o programa é executado ciclicamente num período de tempo definido pelo tamanho do mesmo, denominado varredura (Scan).

Em grandes plantas industriais, é necessária a distribuição da função de controle para vários CPs e computadores de processo. São os sistemas de controle distribuídos. A interligação das E/S remotas é feita através de uma rede de campo (Fieldbus).

Sistemas de Controle de Malha Fechada

Um sistema que estabeleça uma relação de comparação entre a saída e a entrada de referência, utilizando a diferença como meio de controle, é denominado sistema de controle com realimentação, o qual, com frequência é denominado também de sistema de controle de malha fechada. A figura a seguir ilustra um sistema desse tipo.

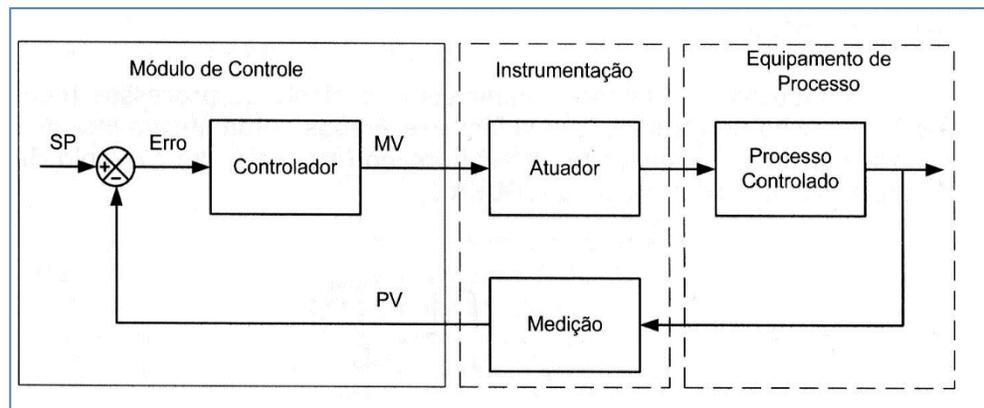


Figura 5-3. Sistema de Controle com Realimentação

Ação de controle PID

A combinação das ações de controle proporcional, integral e derivativo caracteriza o controlador PID. Essa ação combinada tem as vantagens individuais de cada uma das três ações de controle. A equação de um controlador desse tipo é dada por:

$$u(t) = K_p * e(t) + (K_p / T_i) \int e(t)dt + K_p * T_d de(t)/dt$$

Nessa equação K_p é o ganho proporcional, T_i é o tempo integrativo e T_d é o tempo derivativo.

Controle PID de uma Planta Industrial

A utilidade dos controles PID reside na sua aplicabilidade geral à maioria dos sistemas de controle.

Se um modelo matemático da planta pode ser obtido, então é possível aplicar técnicas de projeto na determinação dos parâmetros do controlador PID que vão impor as especificações do regime transitório e do regime permanente do sistema de malha fechada.

Contudo, para plantas muito complexas, onde não é possível obter o modelo matemático, recorreremos a abordagens experimentais de sintonia de controladores PID.

O processo de seleção de parâmetros do controlador que garantam uma dada especificação de desempenho é conhecido como sintonia do controlador.

Algumas regras para sintonia fornecem estimativas dos valores dos parâmetros (K_p , T_i e T_d) e proporcionam um ponto de partida na sintonia fina.

A maioria dos controladores PID são ajustados em campo. A figura a seguir mostra um exemplo de controle PID para temperatura.

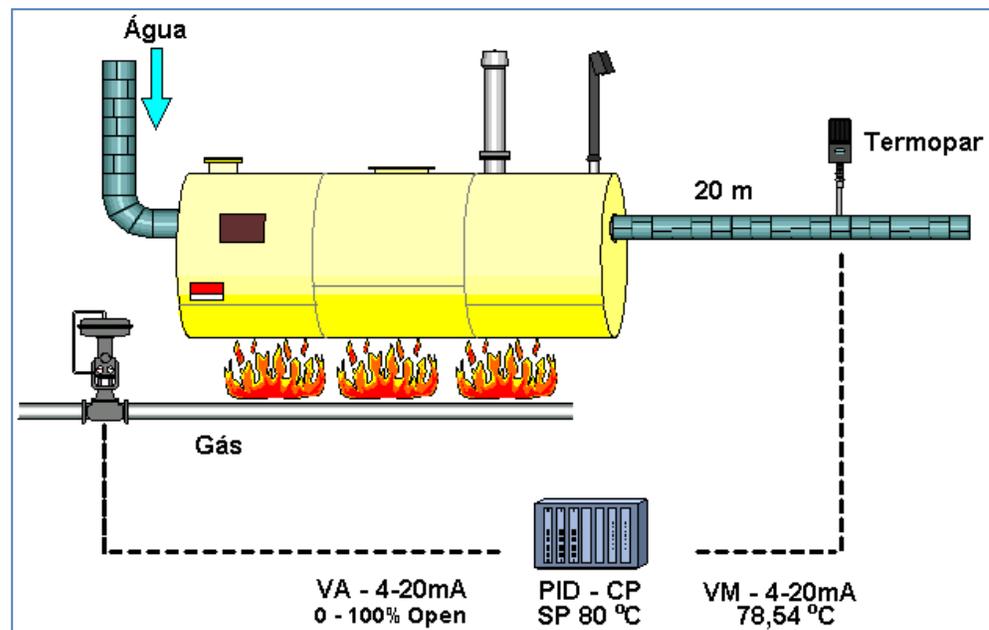


Figura 5-4. Controle PID para Temperatura

Regras de sintonia para controladores PID

Os autores Ziegler e Nichols sugeriram algumas regras para sintonizar controladores PID, ou seja, ajustar os valores de K_p , T_i e T_d baseadas na resposta experimental da planta quando submetida a uma entrada do tipo degrau* ou no valor de K_p que resulta em uma estabilidade marginal, quando consideramos somente a ação proporcional. Tais regras sugerem um conjunto de valores para K_p , T_i e T_d que conferem estabilidade ao sistema de controle associado à planta.

*A função degrau é definida conforme segue: $f(t) = 0$, para $t < 0$ e $f(t) = \text{constante}$, para $t > 0$.

Um dos efeitos resultantes da utilização desse tipo de controle pode ser o aparecimento de um sobressinal máximo de valor elevado, o que não é interessante. Nesse caso, recomenda-se a realização de várias sintonias finas até a obtenção de um resultado aceitável, ou seja, as regras fornecem estimativas iniciais para os parâmetros.

Primeiro método de sintonia de Ziegler-Nichols

Esse método de aplica se a curva de resposta ao degrau de entrada apresentar o aspecto mostrado na figura a seguir*.

*Essa curva de resposta ao degrau pode ser gerada experimentalmente ou a partir da simulação dinâmica da planta.

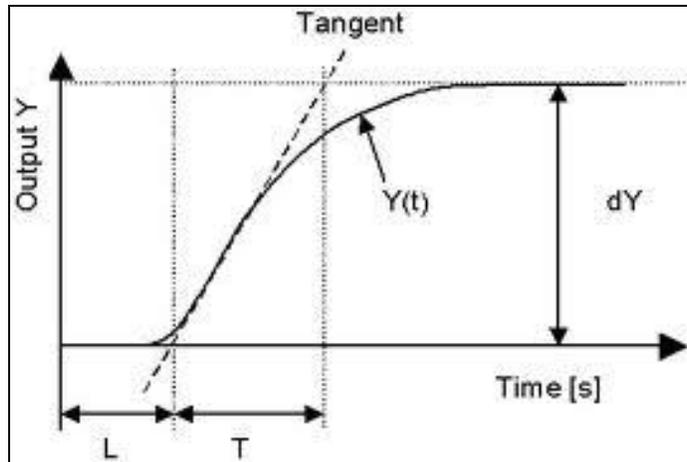


Figura 5-5. Resposta típica ao degrau

Conforme podemos observar no gráfico, a curva de resposta pode ser caracterizada por duas constantes: o tempo de atraso (L) e a constante de tempo (T).

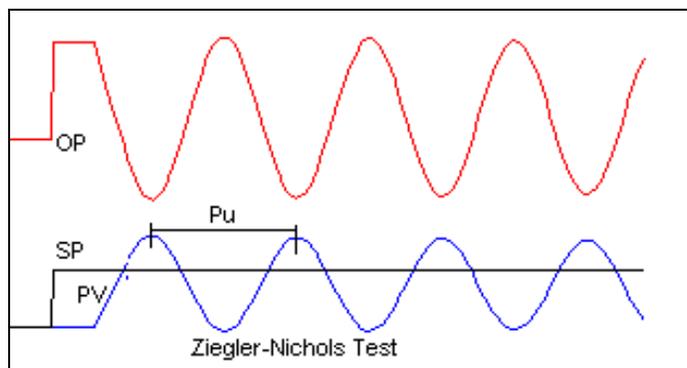
Assim sendo, os valores de K_p , T_i e T_d podem ser obtidos a partir das fórmulas apresentadas na tabela a seguir.

Tabela 5-1. Regra de sintonia de Ziegler-Nichols (Método 1)

Tipo de controlador	K_p	T_i	T_d
P	T / L	INFINITO	0
PI	$0,9 T / L$	$L / 0,3$	0
PID	$1,2 T / L$	$2 L$	$0,5 L$

Segundo método de sintonia de Ziegler-Nichols

No segundo método de sintonia, utilizando apenas a ação de controle proporcional aumenta-se K_p até um valor crítico (K_{cr}) no qual a saída apresenta uma oscilação sustentada pela primeira vez (se isso não ocorrer, o método não é aplicável). A figura a seguir ilustra o formato da resposta oscilatória.

Figura 5-6. Detalhe do período crítico (P_u)

As fórmulas para determinação dos parâmetros K_p , T_i e T_d são baseadas no ganho crítico (K_{cr}) e período crítico (P_u) da oscilação mostrado na figura anterior.

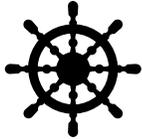
A tabela a seguir apresenta as fórmulas do segundo método.

Tabela 5-2. Regra de sintonia de Ziegler-Nichols (Método 2)

Tipo de controlador	Kp	Ti	Td
P	0,5 Kcr	INFINITO	0
PI	0,45 Kcr	1/1,2 Pu	0
PID	0,6 Kcr	0,5 Pu	0,125 Pu

Fonte: referências indicadas no início deste tutorial.

Bloco Funcional PID



O bloco funcional PID pertencente à biblioteca Util.library do MasterTool IEC XE implementa a funcionalidade controlador PID. A figura a seguir ilustra esse bloco funcional.

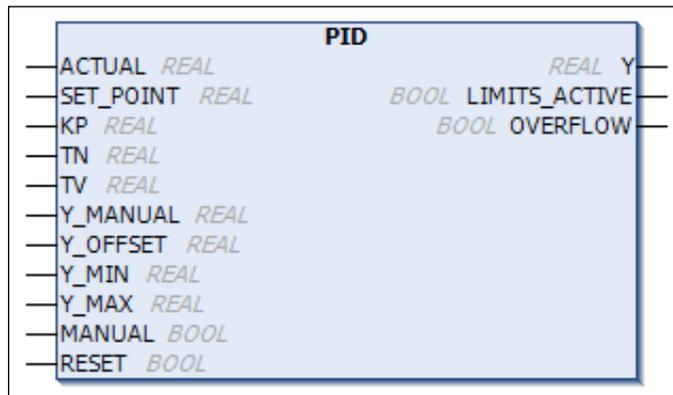


Figura 5-7. Bloco Funcional PID

Entradas do Bloco Funcional

Tabela 5-3. Entradas do Bloco Funcional PID

Variável	Tipo de dado	Descrição
ACTUAL	REAL	Valor atual da variável controlada.
SET_POINT	REAL	Valor desejado, variável de comando.
KP	REAL	Coeficiente de proporcionalidade, ganho de unidade da parte P.
TN	REAL	Tempo de reset, ganho de unidade recíproco da parte I. Tempo dado em segundos, por exemplo, "0,5" para 500 milissegundos.
TV	REAL	Tempo de ação derivativa, ganho de unidade da parte D em segundos, por exemplo, "0,5" para 500 milissegundos.
Y_MANUAL	REAL	Define o valor da saída Y em caso da variável MANUAL = TRUE.
Y_OFFSET	REAL	Offset para a variável Y manipulada.
Y_MIN, Y_MAX	REAL	Limites superior e inferior para a variável Y manipulada. Se Y exceder estes limites, a saída LIMITS_ACTIVE será definida para TRUE e Y será mantido neste intervalo determinado. Este controle somente funcionará se Y_MIN < Y_MAX.
MANUAL	BOOL	Se TRUE, a operação manual será ativada, ou seja, o valor manipulado será definido por Y_MANUAL.
RESET	BOOL	TRUE causa um reset no controlador. Durante a reinicialização, Y= Y_OFFSET.

Saídas do Bloco Funcional

Tabela 5-4. Saídas do Bloco Funcional PID

Variável	Tipo de dado	Descrição
Y	REAL	Valor manipulado, calculado pelo bloco funcional (veja abaixo).
LIMITS_ACTIVE	BOOL	TRUE indica que Y excedeu os limites determinados (Y_MIN, Y_MAX).
OVERFLOW	BOOL	TRUE indica um overflow (veja abaixo).

Considerações Gerais sobre a Programação do Bloco Funcional PID

Y_OFFSET, Y_MIN e Y_MAX servem para a transformação da variável manipulada em um intervalo determinado.

MANUAL pode ser usada para alterar para a operação manual. RESET pode ser usado para reinicializar o controlador.

Na operação normal (MANUAL = RESET = LIMITS_ACTIVE = FALSE) o controlador calcula o erro do controlador ("e") como a diferença de SET_POINT – ACTUAL e gera a derivada em relação ao tempo de/dt. Estes valores são armazenados internamente.

A saída, isto é, a variável Y manipulada, diferentemente do controlador PD, contém uma parte integral adicional e é calculada da seguinte forma:

$$Y = KP \times (D + 1/TN \int edt + TV dD/dt) + Y_OFFSET$$

Assim, além da parte P, também a alteração atual do erro do controlador (parte D) e o seu histórico (parte I) influenciam a variável manipulada.

O controlador PID pode ser facilmente convertido para um controlador PI definindo-se TV=0.

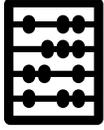
Devido à parte integral adicional, um overflow pode advir da parametrização incorreta do controlador se a integral do erro D tornar-se maior. Portanto, por questões de segurança, uma saída Booleana chamada OVERFLOW está presente (neste caso, com o valor TRUE). Isto somente acontecerá se o sistema de controle for instável devido à parametrização incorreta. Ao mesmo tempo, o controlador será suspenso e somente será ativado novamente através da reinicialização.

Nota: enquanto a limitação para a variável manipulada (Y_MIN e Y_MAX) estiver ativa, a parte integral será adaptada (o histórico dos valores de entrada afetando automaticamente o valor de saída limitado). Se este comportamento não for o desejado, o seguinte é possível: desligue a limitação no controlador PID (Y_MIN >= Y_MAX) e, em vez disto, aplique o operador LIMIT (norma IEC) no valor Y da saída.

PID_fixcycle

O bloco funcional PID_fixcycle também implementa a funcionalidade controlador PID exceto pelo fato de que o tempo do ciclo não é medido automaticamente por uma função interna, mas sim definido pela entrada CYCLE (em segundos)

PID Control



A ALTUS disponibiliza um objeto para implementação do algoritmo PID no Nexto Xpress chamado de PID Control, o qual permite, entre outras facilidades, a visualização gráfica do processo, configuração dos parâmetros do controlador, procedimento de sintonia automática e configuração das variáveis utilizadas pelo controlador.

Ao inserir um objeto PID Control é inserida uma POU do tipo Programa ao projeto. Este programa contém um bloco funcional do tipo PID assim como todos os parâmetros e lógicas necessárias para a sua utilização. Dentro do objeto podem ser configuradas as variáveis que serão utilizadas como entradas e saídas assim como tempo de amostragem utilizado no controle.

Ambiente Gráfico

O ambiente gráfico do objeto PID Control é formado por uma tela composta por duas abas:

- Aba Settings & Chart - esta é a principal aba de trabalho, onde são configurados os principais parâmetros e onde está localizado o gráfico de tendência.
- Advanced Settings - esta aba contém as configurações secundárias do laço PID.

Para detalhamento referente à aplicação do PID Control consulte o Manual de Utilização do MasterTool IEC XE (MT8500) - MU299048.

Sua aplicação será explorada a seguir.

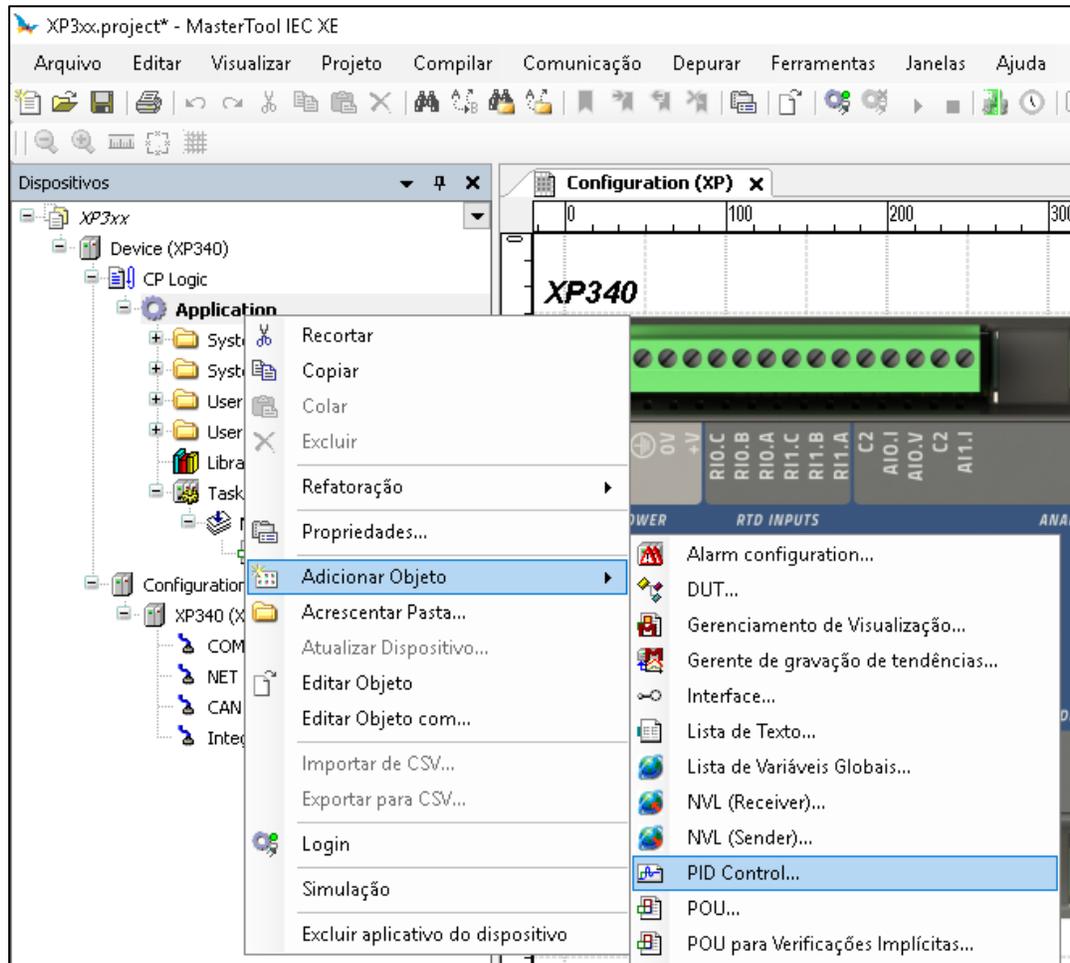


Estudo Dirigido 5-1: utilização do PID Control

Este objeto permite que seja inserido um controlador PID de fácil edição em uma aplicação do MasterTool IEC XE. A seguir serão apresentadas todas as funcionalidades encontradas no objeto PID Control. Entre elas, podem ser citadas: visualização gráfica do processo, configuração dos parâmetros do controlador, procedimento de sintonia automática, configuração das variáveis utilizadas pelo controlador, entre outras.

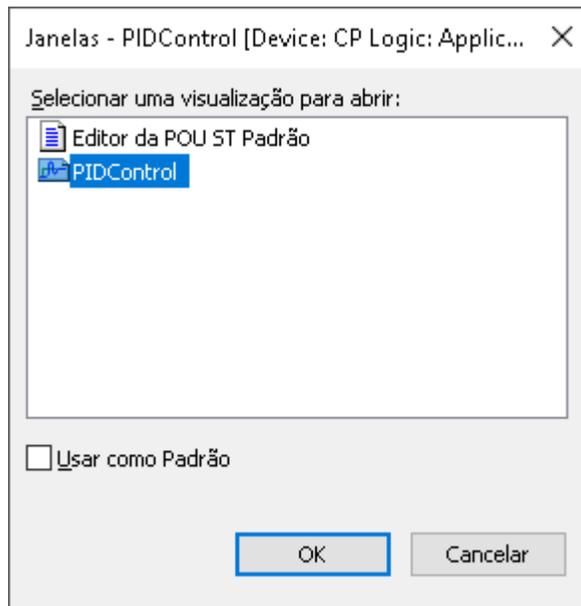
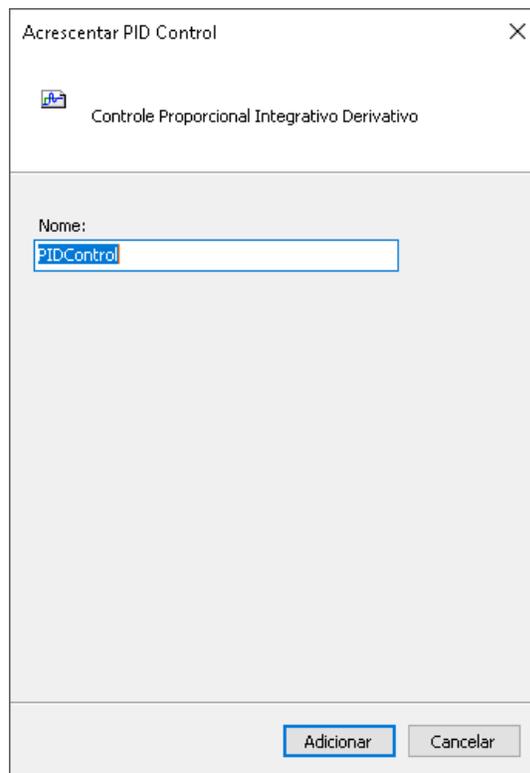
Orientações gerais referentes à realização da atividade

Etapa 1: acrescentar objeto PID Control na aplicação.



Etapa 2: selecionar o ambiente gráfico.

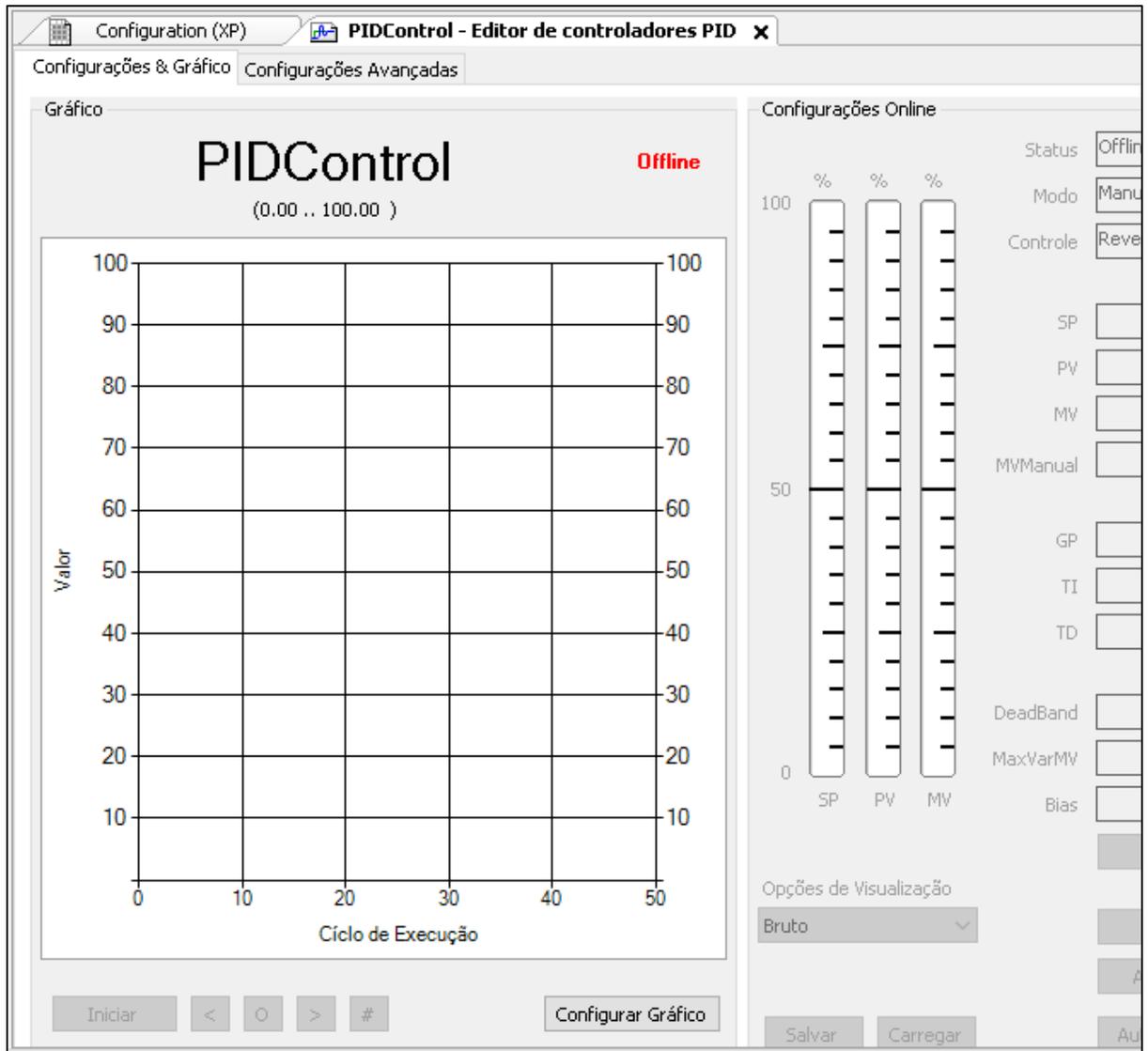
Clique duas vezes com o botão esquerdo do mouse sobre o objeto PID Control, localizado na treeview da aplicação e selecione a componente gráfica desejada.



Etapa 3: visualizar gráfico de tendência.

Os valores apresentados no gráfico de tendência são sempre exibidos em percentual. Este gráfico é composto por três penas que representam as seguintes variáveis

- **SP**: valor de referência do controlador (verde)
- **PV**: variável de processo do controlador (vermelha)
- **MV**: variável manipulada do controlador (azul)



Etapa 4: acessar as funcionalidades do gráfico de tendência

O botão *Iniciar* é utilizado para iniciar a monitoração do processo. Neste caso, as penas do gráfico irão desenhar o comportamento dinâmico das variáveis SP, PV e MV. Após o início da monitoração o nome deste botão é alterado para *Parar*, onde o monitoramento poderá ser finalizado. Após parar a monitoração o nome do botão retorna para *Iniciar*.

O botão *Iniciar* e todos os demais botões que estão localizados imediatamente a sua direita apenas estão habilitados com o CP em modo *Online*. Estes outros botões possuem as seguintes funções:

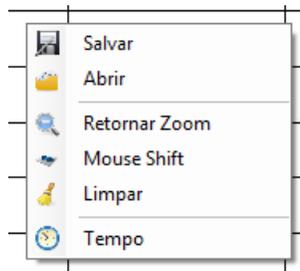
- deslocar o gráfico para a esquerda (<).
- deslocar o gráfico para a direita (>).
- voltar para a posição normal do gráfico (O).
- executar operação de autofit do gráfico (#).



Etapa 5: habilitar o menu de contexto do gráfico de tendência.

Clicando com o botão direito do mouse sobre o gráfico de tendência surge um menu de contexto. Opções do menu:

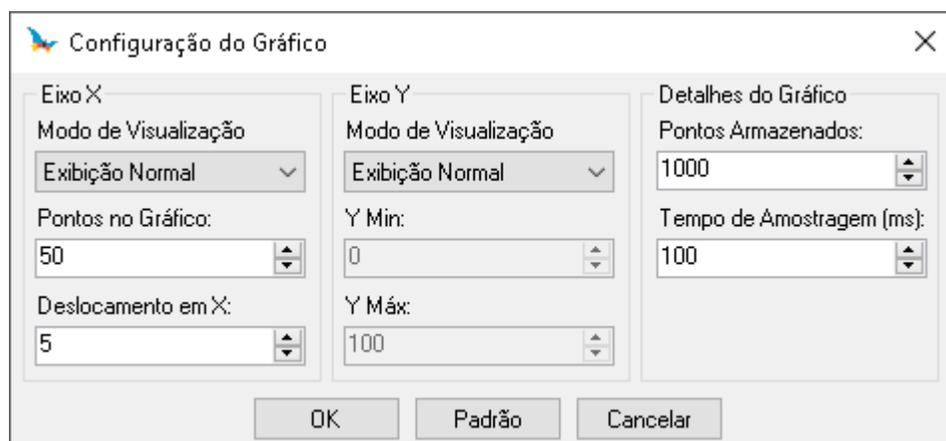
- A opção Salvar somente é habilitada quando existe alguma informação presente no gráfico de tendência. Esta opção permite que sejam armazenados em arquivo .CSV os dados desenhados pelas penas do gráfico.
- A opção Abrir permite que dados anteriormente salvos em arquivo .CSV através da opção Salvar sejam lidas e desenhadas novamente no gráfico de tendência. Esta opção é desabilitada quando está ocorrendo o monitoramento do processo (inicializado pelo botão Iniciar).
- A opção Retornar Zoom habilita a aplicação de zoom no gráfico. Esta opção é habilitada quando existem informações desenhadas no gráfico.
- A opção Mouse Shift permite que a partir do mouse seja possível realizar deslocamentos no gráfico de tendência. Esta opção é habilitada quando existem informações desenhadas no gráfico.
- A opção Limpar permite que o gráfico seja limpo, apagando todas as informações por ele contidas. Esta opção é habilitada quando existem informações desenhadas no gráfico.
- A opção Tempo permite alternar o tipo de dado do eixo "x" entre ciclo de execução e tempo em segundos.



Etapa 6: configurar gráfico de tendência.

A janela Configuração do Gráfico é acessada pressionando o botão Configurar Gráfico do grupo Gráfico, localizado na aba Configurações & Gráfico. Este botão encontra-se habilitado somente com o CP em Offline.

A janela Configuração do Gráfico permite configurar algumas características visuais do gráfico de tendência.



Etapa 7: configurar parâmetros do controlador PID.

O grupo Configurações Online é responsável por mostrar e possibilitar a configuração dos principais parâmetros do controlador PID. As funcionalidades deste grupo apenas estão habilitadas com o CP em modo Online.

No canto superior esquerdo do grupo, estão presentes os gráficos de barra que exibem os valores atuais das variáveis SP, PV e MV em percentual. No lado direito do grupo *Configurações Online*, encontram-se os campos não editáveis:

- Status: informa o status do CP, pode assumir: Offline, Parado ou Executar.
- Modo: informa se o controlador PID está configurado em modo Manual ou Automático.
- Controle: informa a direção certa da ação MV para fornecer uma realimentação negativa.

Quando em controle *Direto* indica que MV deve aumentar em resposta a um aumento de PV, já em controle *Inverso* indica que MV deve diminuir em resposta a um aumento de PV.

Logo abaixo, o campo *SP* permite visualizar o valor de referência atual do controlador PID, assim como o seu ajuste quando o controlador está operando em modo automático. No campo *PV* é possível visualizar o valor da variável de processo do controlador PID. Este campo não permite edição. No campo *MV* é possível visualizar o valor da variável manipulada do controlador PID. Este campo não permite edição.

Quando o controlador PID encontra-se em modo automático, o campo *MVManual* não permite edição. Entretanto, quando o controlador está em modo manual, o valor da variável *MV* pode ser ajustado através deste campo.

Os campos *GP*, *TI* e *TD* permitem a edição dos parâmetros ganho proporcional, tempo integrativo e tempo derivativo do controlador PID. Nos campos *DeadBand*, *MaxVarMV* e *Bias* são configurados, respectivamente, a banda morta, a máxima variação permitida para a variável *MV* e o offset adicionado à *MV*.

O botão *Escrever* é responsável por enviar ao CP todos os parâmetros que foram modificados, concretizando a alteração dos parâmetros no controlador PID.

Os botões *Manual* e *Automático* alteram o modo de funcionamento do controlador para manual e automático, respectivamente.

O botão *Autoconfigurar* abre a janela de execução do procedimento de sintonia automática do controlador.

O campo *Opções de Visualização* é utilizado para controlar o modo de exibição dos valores dos parâmetros e variáveis. Os valores possíveis são:

- Bruto: exibe os valores tal qual está no CP.
- Porcento: exibe os valores na forma de percentual, na faixa de 0% a 100% dentro da faixa de valores máximos e mínimos do parâmetro ou variável.
- Engenharia: exibe os valores na forma de escala de engenharia configurada para o parâmetro ou variável.

Os campos afetados pelas opções de visualização são: *SP*, *PV*, *MV*, *MVManual*, *DeadBand*, *MaxVarMV* e *Bias*. Estes campos são afetados apenas quando exibidos na caixa de texto, o gráfico de tendência e os gráficos de barra sempre são mostrados em porcentagem.

A unidade de engenharia exibida ao lado da caixa de texto dos campos citados também muda de acordo com a seleção do campo *Opções de Visualização*, exibindo a unidade de engenharia (se configurada), percentual “%” ou nada caso a opção seja *Bruto*.

Para os campos *MV*, *MVManual*, *MaxVarMV* e *Bias* não faz sentido a visualização em escala de engenharia, neste caso será exibido em percentual.

O botão *Salvar* salva em arquivo .CSV as configurações atuais do controlador. O botão *Carregar* lê de arquivo .CSV com configurações anteriormente salvas e carrega-as no controlador. A presença dos botões *Salvar* e *Carregar* permitem que parâmetros do controlador sejam salvos e carregados posteriormente, por exemplo, após um procedimento de manutenção do CP. Para carregar no controlador os parâmetros que foram previamente armazenados em um arquivo .CSV deve-se clicar no botão *Carregar*. Logo após ser selecionado o arquivo .CSV adequado, uma pequena janela irá ser apresentada ao usuário, permitindo que este selecione os parâmetros que deseja carregar no controlador.

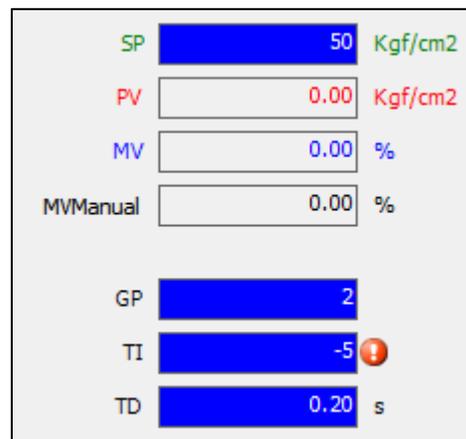
A figura a seguir exibe a janela mencionada. Pode-se notar os valores dos parâmetros que podem ser salvos e carregados no objeto PIDControl. Ao pressionar o botão *OK*, os parâmetros marcados serão automaticamente carregados no CP, reconfigurando o controlador.



A escrita ou edição dos parâmetros do controlador PID no CP é realizada a partir do botão *Escrever* localizado no grupo *Configurações Online*.

O fundo dos campos do grupo *Configurações Online* que permitem edição, quando sofrem alteração de seus valores, passam para a cor azul e a fonte para a cor branca. Isto indica que o valor do parâmetro ou variável foi modificada, mas o novo valor ainda não foi enviado para o CP. Ao pressionar o botão *Escrever* todos os parâmetros e variáveis nesta condição e que não apresentam erro são enviados para o CP e a cor do fundo e da fonte originais são restauradas.

A figura a seguir demonstra os campos *SP*, *GP*, *TI* e *TD* foram alterados, mas que ainda não tiveram os seus novos valores enviados para o CP. Pode-se notar que o campo *TI* apresenta uma mensagem de erro, pois o seu valor é negativo (a mensagem de erro pode ser visualizada passando-se o mouse sobre o sinal de exclamação). Dessa forma, quando o botão *Escrever* for pressionado, somente os valores dos campos que não apresentam erros são realmente modificados no CP.



A figura a seguir apresenta a visualização desses campos após o botão *Escrever* ser pressionado.

SP	50.00	Kgf/cm2
PV	0.00	Kgf/cm2
MV	0.00	%
MVManual	0.00	%
GP	2.00	
TI	-5	
TD	0.20	s

Caso o valor de algum dos campos tenha sido alterado e o seu valor ainda não tenha sido enviado para o CP, é possível restaurar o seu valor atual do CP clicando com o botão direito do mouse sobre o campo e em seguida selecionando *Valor Atual*. Esta operação é exibida na sequência.



Etapa 8: configurações avançadas

Depois de inserido o objeto PID Control na aplicação, o primeiro passo a ser tomado para se utilizar o controlador PID é ajustar as configurações do laço PID de acordo com a aplicação. Para que isso seja realizado é preciso que seja acessada a aba *Configurações Avançadas*. A figura a seguir exibe esta aba.

Configurações de Entrada/Saída		Variável	Mínimo	Máximo	Unidade
SP Bruto	SP		0.00	30000.00	
Engenharia de SP			0.00	100.00	
PV Bruto	PV		0.00	30000.00	
Engenharia de PV			0.00	100.00	
MV Bruto	MV		0.00	30000.00	

Configurações de Controle		Configurações do Projeto
Tempo de Amostragem (ms):	100	Restrições do Autoconfigurar
Controle:	Inverso	<input checked="" type="checkbox"/> Associação Automática de Tarefa
<input checked="" type="checkbox"/> Habilitar ação proporcional		
<input type="checkbox"/> Habilitar ação integral		
<input type="checkbox"/> Habilitar ação derivativa		
<input type="checkbox"/> Habilitar ação derivativa em PV		

Figura 5-8. Aba Configurações Avançadas

Observando a figura anterior nota-se que as configurações desta aba são divididas em dois grupos: *Configurações de Entrada/Saída* e *Configurações de Controle*, e duas abas: *Configurações do Projeto* e *Restrições do Autoconfigurar*.

É importante ressaltar que todas as alterações realizadas na aba *Configurações Avançadas* devem ser feitas com o CP em *Offline*. Desta forma, após alterar os parâmetros é necessário que o projeto seja carregado no CP. Quando o CP estiver em *Online* todos os campos estarão desabilitados, não permitindo edição.

Grupo: Configurações de Entrada/Saída

Este grupo é utilizado para configurar as faixas de operação da entrada e da saída do PID, a PV e a MV, respectivamente. A figura a seguir exhibe o grupo.

Configurações de Entrada/Saída				
	Variável	Mínimo	Máximo	Unidade
SP Bruto	SP	0.00	30000.00	
Engenharia de SP		0.00	100.00	
PV Bruto	PV	0.00	30000.00	
Engenharia de PV		0.00	100.00	
MV Bruto	MV	0.00	30000.00	

Nos campos *Variável* são configuradas as variáveis que serão utilizadas como entrada (PV) e saída (MV) do PID. O objeto PID Control aceita somente variáveis do tipo REAL.

O campo variável SP é o único que pode ser deixado em branco. Neste caso, será assumido que uma variável interna do objeto PID Control deverá ser utilizada para este campo. Estas variáveis devem ser variáveis globais declaradas em outros objetos da aplicação como configuração de redes de campo ou objetos GVL. A utilização de variáveis externas neste campo permite, por exemplo, que estratégias de controle em cascata possam ser aplicadas com o objeto PID Control.

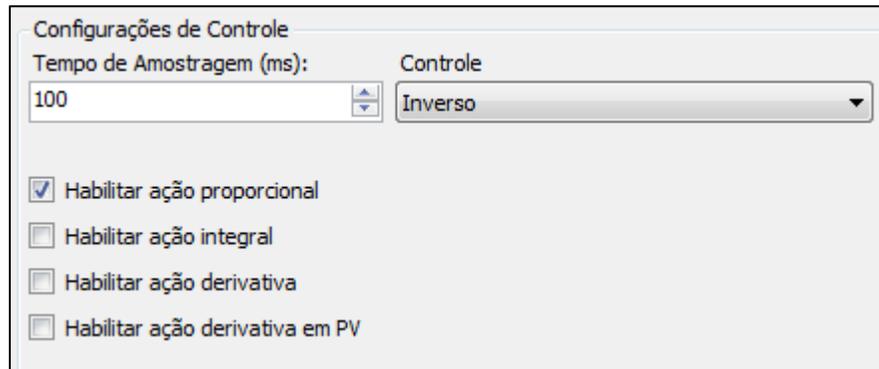
Os campos *Mínimo* e *Máximo*, definem a faixa de operação das variáveis SP, PV e MV. Os campos *Mínimo SP* e *Máximo SP*, não permitem edição. Estes campos assumem os valores dos campos *Mínimo Engenharia de PV* e *Máximo Engenharia de PV*, respectivamente.

O ajuste correto dessas informações é de grande importância para o funcionamento adequado do laço PID. É importante também dizer que estes valores são também utilizados para validar a entrada de dados dos campos do grupo *Configurações Online* da aba *Configurações e Gráfico*, de acordo com a configuração do parâmetro *Opções de Visualização*.

Grupo: Configurações de Controle

Este grupo permite a configuração de alguns parâmetros ligados ao modo de funcionamento do controlador PID, são eles:

- Tempo de Amostragem (ms): configura o intervalo de tempo que o PID é executado, podendo variar de 1 ms a 1.000.000 ms.
- Controle: este parâmetro de entrada seleciona a direção certa da ação MV para fornecer uma realimentação negativa. Se uma seleção errada é feita, a realimentação resultante será positiva, e o PID não será capaz de controlar o processo. O controle Direto deve ser selecionado, quando MV deve aumentar em resposta a um aumento de PV. O controle Inverso deve ser selecionado, quando MV deve diminuir em resposta a um aumento de PV.
- Habilitar...: estes campos habilitam individualmente as quatro ações (proporcional, integrativa, derivativa e derivativa em PV) que compõem o bloco PID.



Configurações de Controle

Tempo de Amostragem (ms): 100

Controle: Inverso

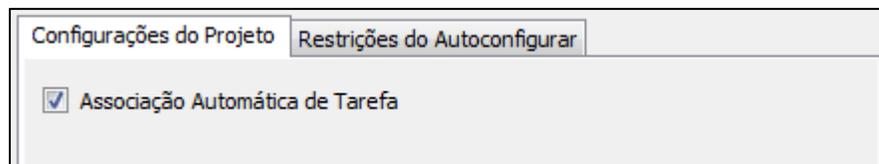
Habilitar ação proporcional

Habilitar ação integral

Habilitar ação derivativa

Habilitar ação derivativa em PV

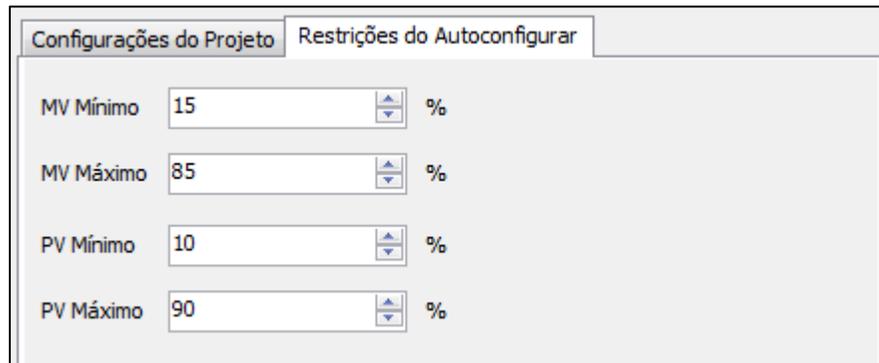
Esta aba contém a opção *Associação Automática de Tarefa*, com a opção habilitada o controlador é associado automaticamente a uma tarefa do sistema, o que permite que ele seja utilizado normalmente. Se a opção estiver desabilitada o controlador deve ser associado a alguma tarefa manualmente ou chamado em alguma POU do usuário.



Configurações do Projeto Restrições do Autoconfigurar

Associação Automática de Tarefa

Esta aba contém campos que definem os valores mínimos e máximos que a sintonia automática pode atribuir para as variáveis MV e PV.



Configurações do Projeto Restrições do Autoconfigurar

MV Mínimo 15 %

MV Máximo 85 %

PV Mínimo 10 %

PV Máximo 90 %

Etapa 9: Sintonia Automática

O procedimento de sintonia automática do objeto PID Control é realizado acessando a janela *Autoconfiguração*. Isto é realizado clicando no botão *Autoconfigurar* localizado no grupo *Configurações Online* da aba *Configurações & Gráfico*. A figura a seguir exibe a janela *Autoconfigurar*.

Através da janela *Autoconfigurar* é possível realizar o procedimento de sintonia dos parâmetros do controlador PID utilizando o método de síntese. Para a aplicação deste método é necessário que sejam conhecidos os parâmetros de um sistema de primeira ordem que represente satisfatoriamente o processo. Desta forma, é preciso que estes parâmetros sejam identificados através da realização de um experimento em malha aberta, controlado por um operador. Este experimento consiste em aplicar um sinal degrau ao processo e aguardar que o mesmo venha a entrar em regime permanente. A seguir serão apresentados os passos necessários para efetuar o procedimento de sintonia automática.

No campo *Tf* é configurada a constante de tempo em malha fechada desejada. É importante ressaltar que o método de síntese não produz bons resultados quando $Tf < Tc < 10$ e quando $Tc / (Tde + SampleTime / 2) < 10$. Onde *Tc* e *Tde* correspondem à constante de tempo e ao atraso de transporte do processo e *SampleTime* é o intervalo de execução do PID.

No campo *Passo* é possível configurar o degrau em percentual que será aplicado ao processo no experimento. Ou seja, o sinal aplicado na saída do controlador será correspondente a $(MV + MV * Step)$.

No campo *Método* é possível selecionar entre dois métodos para efetuar a identificação dos parâmetros do processo. *PureStat*, utiliza apenas informações estatísticas, enquanto *PolyStat* utiliza informações estatísticas em conjunto com aproximação polinomial. As informações estatísticas são utilizadas para tentar amenizar presença de ruídos no processo.

A opção *Parada Automática* habilita o término automático do experimento necessário para a obtenção dos parâmetros aproximados de um modelo de primeira ordem a serem utilizados pelo método de síntese. Para determinar o término o algoritmo monitora o sinal de PV aguardando pela sua estabilidade. Para aguardar a estabilidade o algoritmo monitora o sinal a cada amostragem. Se a variação de uma amostra para outra for inferior a 2% da variação total da escala o algoritmo aguarda 800 amostras dentro da faixa para considerar o sinal estável. Caso seja percebido pela representação gráfica que o sinal não estabiliza o experimento pode ser alternado para modo manual.

Os campos *GP*, *TI* e *TD* da aba *Parâmetros do PID*, apresentam, inicialmente, os valores atuais configurados para os parâmetros do controlador.

Os campos *Gs*, *Tc* e *Tde* da aba *Processo*, apresentam, os valores do ganho estático do processo, da constante de tempo do processo e do tempo morto do processo, respectivamente.

Antes de iniciar o experimento de identificação dos parâmetros do processo e de sintonia dos parâmetros do controlador, é importante que o processo esteja em regime permanente. Garantida essa situação o experimento pode ser iniciado pressionando o botão *Iniciar*.

Tendo sido o experimento iniciado, o gráfico de tendência começa a monitorar o processo para que o operador acompanhe o que está ocorrendo. No experimento, inicialmente, o controlador PID é passado automaticamente para o seu modo manual de funcionamento.

Em seguida, após certo período de tempo o sinal degrau é aplicado. É importante notar que o sinal degrau não é aplicado imediatamente. Neste intervalo em que a MV permanece inalterada estão sendo coletadas algumas informações estatísticas que serão utilizadas para minimizar a possível presença de ruídos no processo.

Depois de aplicado o degrau, o processo irá começar a responder ao estímulo até que entre novamente em regime permanente. Deve-se manter o experimento em execução com o processo em regime permanente por certo período de tempo, quando então o procedimento pode ser parado pressionando o botão *Parar*. Deve-se lembrar que se a opção *Parada Automática* estiver habilitada em algum momento, o experimento irá ser finalizado automaticamente pelo procedimento de sintonia. Entretanto, o operador ainda é capaz de finalizar o experimento quando achar conveniente, mesmo antes da finalização automática.

Ao pressionar o botão *Parar* o experimento é finalizado e os novos parâmetros sugeridos para o controlador são apresentados nos campos *GP*, *TI* e *TD*. Para que estes parâmetros sejam enviados para o CP o botão *Escrever* deve ser pressionado.

Após o procedimento de sintonia automática, o controlador permanecerá funcionando em seu modo manual. Para que retorne ao modo automático deve-se pressionar o botão *Automático* do grupo *Configurações Online* da aba *Configurações & Gráfico*.



DICA: consulte o Instrutor para orientações adicionais.

Blocos Funcionais de Controle Avançado



Diversos elementos de software de controle avançado estão disponíveis na biblioteca NextoPID. Esses blocos funcionais (FB) ou funções (FUN) estão descritos na tabela a seguir.

Tabela 5-5. Lista de funções avançadas de controle na biblioteca NextoPID

Elemento de Software	Tipo	Descrição
PIDA_INT	FB	Controle PID com a maioria dos parâmetros do tipo REAL, com suporte para configurações de restrição e cascata
PIDA_INT	FB	Controle PID com a maioria dos parâmetros do tipo INT, com suporte para configurações de restrição e cascata
PIDA_TUNE_REAL	FB	Controle PID com a maioria dos parâmetros do tipo REAL, com suporte para configurações de restrição e cascata e com sintonia automática
PIDA_TUNE_INT	FB	Controle PID com a maioria dos parâmetros do tipo INT, com suporte para configurações de restrição e cascata e com sintonia automática
ControlON_OFF	FB	Controle liga-desliga
ControlPWM	FB	Converte uma saída analógica do tipo REAL em uma saída digital do tipo BOOL com modulação de largura de pulso (PWM)
ControlLowFilter	FB	Aplica um filtro passa-baixa a uma variável do tipo REAL
ControlDelay	FB	Aplica um atraso a uma variável do tipo REAL
ControlLag	FB	Aplica um atraso de primeira ordem a uma variável do tipo REAL com ganho e limites
ControlDelayLag	FB	Combinação dos blocos de função ControlDelay e ControlLag
ControlLeadLag	FB	Implementa tempos de avanço e atraso com ganho e limites (por exemplo, para uso em controles de avanço)
ControlSelectMax	FUN	Seleciona a saída PID máxima para uso em configurações de controle de restrição
ControlSelectMin	FUN	Seleciona a saída PID mínima para uso em configurações de controle de restrição
ControlSplitRange	FUN	Divide a saída PID em dois resultados para diferentes válvulas

Aplicação do Bloco Funcional PIDA_REAL

O objetivo de um bloco de função PIDA_REAL é manter a variável do processo (PV) próxima ao ponto de ajuste selecionado (SP ou SP_CASC), controlando a saída MV (variável manipulada). O erro é computado como: SP (valor desejado ou referência) – PV (variável de processo). O objetivo principal do controle PIDA_REAL é manter a magnitude do erro tão baixa quanto possível. A figura a seguir mostra um bloco funcional PIDA_REAL com suas entradas e saídas.

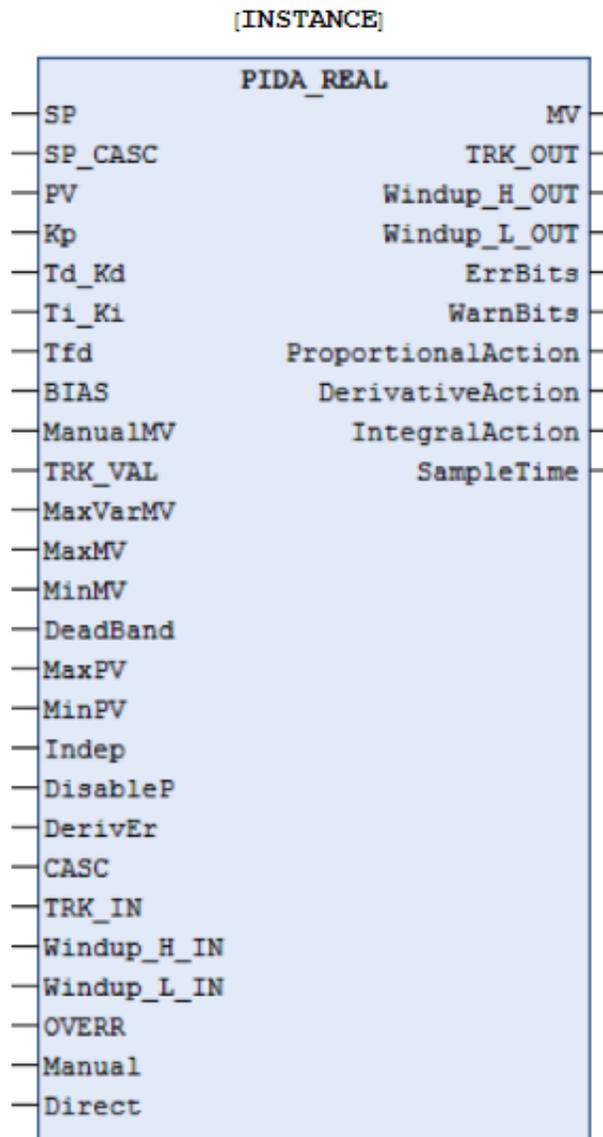


Figura 5-9. Bloco Funcional PIDA_REAL

Descrição das Entradas

- SP - Define o valor desejado para a variável de processo (PV).
- SP_CASC - Funciona como SP quando o PIDA_REAL está em modo cascata.
- PV - Corresponde à variável do processo controlada pelo PIDA_REAL. Normalmente, ela é lida de um transmissor, por exemplo, utilizando uma entrada analógica.
- Kp – Este ganho se aplica ao termo proporcional do controle PIDA_REAL. Na forma dependente (Indep = FALSE), ele também afeta os termos integral e derivativo. Na forma independente (Indep = TRUE), afeta apenas o termo proporcional.
- Td_Kd – A entrada de Tempo Derivativo ou Ganho Derivativo se aplica ao termo derivativo do controle PIDA_REAL. Na forma dependente (Indep = FALSE), significa um tempo derivativo, e o ganho derivativo é calculado como: $Kp * Td_Kd$. Note que o ganho proporcional (Kp) afeta o ganho derivativo, na forma dependente. Na forma independente (Indep = TRUE), Td_Kd expressa diretamente o ganho derivativo.

- **Ti_Ki** - Este tempo ou ganho se aplica ao termo integral do controle PIDA_REAL. Na forma dependente (Indep = FALSE), significa um tempo integral, e o ganho integral é calculado como:
 - Se $Ti_Ki \geq 0.001$ segundos: Kp/Ti_Ki ;
 - A ação integral é desativada se $Ti_Ki < 0.001$ segundos: 0.

Note que o ganho proporcional (Kp) afeta o ganho integral, na forma dependente. Na forma independente (Indep = TRUE), **Ti_Ki** expressa diretamente o ganho integral.

- **Tfd** - Tempo de Filtro para Ação Derivativa: este parâmetro pode ser utilizado para filtrar a ação derivativa. É útil para filtrar ruídos e para suavizar solavancos na ação derivativa.
- **BIAS** - É um deslocamento (offset) adicionado à saída do PIDA_REAL (MV).
- **ManualMV** - Este parâmetro de entrada é copiado para a saída MV quando o modo manual é selecionado. No modo manual, o PIDA_REAL não tenta fazer com que a variável do processo (PV) se aproxime do ponto de ajuste selecionado (SP ou SP_CASC). O operador ou outro sistema é responsável pelo controle no modo manual.
- **TRK_VAL** - Este parâmetro de entrada é copiado para a saída MV quando o modo de rastreamento é selecionado. O modo de rastreamento é utilizado em configurações em cascata para indicar a um mestre de cascata que seu escravo não está em modo automático/cascata. Este parâmetro de entrada também é utilizado quando o modo de restrição é selecionado, para um cálculo diferente da ação integral.
- **MaxVarMV** - Este parâmetro de entrada pode ser utilizado para impor um limite na variação de MV entre dois ciclos consecutivos. Pode ser útil para evitar variações rápidas em válvulas, por exemplo. Se MaxVarMV for igual a 0, essa limitação é desativada. Se MaxVarMV for maior que 0, o valor absoluto da variação de MV é limitado a ser igual a MaxVarMV.
- **MaxMV** - Este parâmetro de entrada impõe um limite máximo à saída MV. Se MV for calculado e acabar sendo maior do que MaxMV, então o valor de MaxMV é atribuído a MV.
- **MinMV** - Este parâmetro de entrada impõe um limite mínimo à saída MV. Se MV for calculado e acabar sendo menor do que MinMV, então o valor de MinMV é atribuído a MV.
- **DeadBand** - Este parâmetro de entrada define um valor absoluto mínimo para qualquer erro que ocorra para causar uma ação de controle. Se o valor absoluto do erro (SP - PV) for menor do que a Zona Morta (DeadBand), o controlador PIDA_REAL assume como se tivesse um erro igual a zero e, portanto, não altera a saída (MV). Também pode ser útil para evitar movimentos desnecessários da válvula quando o valor do erro está em uma faixa aceitavelmente baixa.
- **MaxPV** - Este parâmetro de entrada impõe um limite máximo aos parâmetros de entrada PV, SP e SP_CASC. Se PV, SP ou SP_CASC forem maiores do que MaxPV, então o valor de MaxPV é atribuído ao respectivo parâmetro de entrada (PV, SP ou SP_CASC) para uso interno no PIDA_REAL.
- **MinPV** - Este parâmetro de entrada impõe um limite mínimo aos parâmetros de entrada PV, SP e SP_CASC. Se PV, SP ou SP_CASC forem menores do que MinPV, então o valor de MinPV é atribuído ao respectivo parâmetro de entrada (PV, SP ou SP_CASC) para uso interno no bloco de função PIDA_REAL.
- **Indep** - Quando TRUE, este parâmetro de entrada indica que os ganhos proporcional, derivativo e integral são independentes; caso contrário, são dependentes. É útil

definir `Indep` como `TRUE` se o usuário deseja alterar `Kp` para afetar apenas a ação proporcional, sem afetar as ações derivativa ou integral.

- `DisableP` - Quando `TRUE`, este parâmetro de entrada desabilita a ação proporcional, mas apenas com ganhos dependentes (`Indep = FALSE`). Isso pode ser necessário, por exemplo, se o usuário deseja um controlador integral puro com um ganho dependente (Kp / Ti_Ki). Neste caso, o usuário não pode atribuir $Kp = 0$ para desabilitar a ação proporcional, pois isso também desabilitaria a ação integral. Ao usar ganhos independentes (`Indep = TRUE`), o parâmetro de entrada `DisableP` é ignorado. Neste caso, é possível desativar a ação proporcional atribuindo $Kp = 0$, sem afetar as ações derivativa e integral.
- `DerivEr` - Quando `TRUE`, a ação derivativa é calculada com base no erro; caso contrário, é calculada com base no PV (Variável de Processo). Assim, quando `DerivEr` é `TRUE`, uma ação derivativa pode ser desencadeada por uma mudança no PV ou por uma mudança no Setpoint. Quando `DerivEr` é `FALSE`, uma ação derivativa só pode ser desencadeada por uma mudança no PV. Às vezes, ocorrem grandes alterações no Setpoint entre dois ciclos consecutivos do CP. Por outro lado, grandes mudanças no PV não são esperadas entre dois ciclos consecutivos do CP. Portanto, na maioria das aplicações, calcula-se a ação derivativa com base no PV (`DerivEr = FALSE`), para evitar um impacto no MV (Variável Manipulada) quando o Setpoint muda abruptamente.
- `CASC` - Quando `TRUE`, indica que `PIDA_REAL` está operando no modo cascata. Neste caso, o parâmetro de entrada `SP_CASC` corresponde ao Setpoint, em vez de `SP`.
- `TRK_IN` – Esta entrada está relacionada à indicação de modo de rastreamento de um escravo em cascata. Quando `TRUE`, indica que a saída MV deve ser copiada a partir do parâmetro de entrada `TRK_VAL`.
- `Windup_H_IN` - Indicação de windup alto na ação integral de um escravo em cascata: este parâmetro de entrada deve ser utilizado em um mestre em cascata e está conectado à saída `Windup_H_OUT` do correspondente escravo em cascata. É necessário para evitar problemas de windup na ação integral. Se a entrada for `TRUE`, a ação integral do mestre em cascata não é aumentada, mas pode ser diminuída.
- `Windup_L_IN` - Indicação de windup baixo na ação integral de um escravo em cascata: este parâmetro de entrada deve ser utilizado em um mestre em cascata e está conectado à saída `Windup_L_OUT` do correspondente escravo em cascata. É necessário para evitar problemas de windup na ação integral. Se a entrada for `TRUE`, a ação integral do mestre em cascata não é diminuída, mas pode ser aumentada.
- `OVERR` - Quando `TRUE`, indica que o `PIDA_REAL` está sendo substituído por outro bloco do mesmo tipo, em configurações de restrição. Nessa situação, a ação integral do bloco `PIDA_REAL` é calculada de maneira diferente usando o parâmetro de entrada `TRK_VAL`.
- `Manual` - Quando `TRUE`, indica que o `PIDA_REAL` está no modo manual, e a MV (Variável Manipulada) é copiada do parâmetro de entrada `ManualMV`.
- `Direct` – O `PIDA_REAL` pode ser configurado para controle direto (`Direct = TRUE`) ou controle reverso (`Direct = FALSE`). Alguns processos controlados requerem controle direto, enquanto outros requerem controle reverso. Se este parâmetro não for configurado corretamente, um controle de realimentação positiva será estabelecido, e `PIDA_REAL` não conseguirá controlar o processo. As seguintes regras devem ser seguidas para obter uma configuração correta com realimentação negativa:
 - O controle direto deve ser usado quando a MV (Variável Manipulada) deve ser aumentada para controlar o processo quando a PV (Variável de

Processo) aumenta. Em outras palavras, um aumento na MV faz com que a PV diminua;

- O controle reverso deve ser usado quando a MV deve ser diminuída quando a PV aumenta. Em outras palavras, uma diminuição na MV faz com que a PV diminua.

Descrição das Saídas

- MV - Corresponde à variável manipulada para controlar o processo. Normalmente, o seu valor é escrito em um atuador, por exemplo, utilizando uma saída analógica.
- TRK_OUT - Indicação de modo de rastreamento para um mestre em cascata: em configurações de cascata, esta saída é configurada por um escravo em cascata quando este não está em modo automático ou não está em modo cascata. Esta saída deve ser conectada ao parâmetro de entrada TRK_IN do mestre em cascata correspondente, para que o mestre saiba que não está controlando o escravo em cascata.
- Windup_H_OUT - Indicação de windup alto para um mestre em cascata: este parâmetro de saída deve ser utilizado em um escravo em cascata e está conectado à entrada Windup_H_IN do mestre em cascata correspondente. É necessário para evitar problemas de windup na ação integral do mestre em cascata.
- Windup_L_OUT - Indicação de windup baixo para um mestre em cascata: este parâmetro de saída deve ser utilizado em um escravo em cascata e está conectado à entrada Windup_L_IN do mestre em cascata correspondente. É necessário para evitar problemas de windup na ação integral do mestre em cascata.
- ErrBits - Bits de indicação de erro fatal: este parâmetro de saída é uma estrutura do tipo PIDA_ERROR_BITS com vários bits de erro fatal que indicam parâmetros de entrada inadequados ou um tempo de amostragem ruim. Esses erros impedem a execução do bloco de função PIDA_REAL. Os bits são:
 - O bit MaxMinMV é ativado quando $\text{MaxMV} \leq \text{MinMV}$;
 - O bit MaxMinPV é ativado quando $\text{MaxPV} \leq \text{MinPV}$;
 - O bit Kp é ativado quando $\text{Kp} \leq 0$ com $\text{Indep} = \text{FALSE}$, ou $\text{Kp} < 0$ com $\text{Indep} = \text{TRUE}$;
 - O bit Td_Kd é ativado quando $\text{Td_Kd} < 0$;
 - O bit Ti_Ki é ativado quando $\text{Ti_Ki} < 0$;
 - O bit Tfd é ativado quando $\text{Tfd} < 0$;
 - O bit MaxVarMV é ativado quando $\text{MaxVarMV} < 0$;
 - O bit DeadBand é ativado quando $\text{DeadBand} < 0$;
 - O bit SampleTime é ativado quando a saída SampleTime é maior que 60 s ou menor que 0,001 s, ou se o bloco funcional for chamado dentro de uma tarefa não cíclica.
- WarnBits - Bits de indicação de aviso: este parâmetro de saída é uma estrutura do tipo PIDA_WARNING_BITS com vários bits de aviso que indicam limites atingidos. Esses avisos não impedem a execução do bloco de função PIDA_REAL. Os bits são:
 - O bit PV_MaxPV é ativado quando o PV está sendo limitado por MaxPV;
 - O bit PV_MinPV é ativado quando o PV está sendo limitado por MinPV;
 - O bit SP_MaxPV é ativado quando o SP está sendo limitado por MaxPV;
 - O bit SP_MinPV é ativado quando o SP está sendo limitado por MinPV;

- O bit SP_CASC_MaxPV é ativado quando o SP_CASC está sendo limitado por MaxPV;
 - O bit SP_CASC_MinPV é ativado quando o SP_CASC está sendo limitado por MinPV;
 - O bit MV_MaxMV é ativado quando a MV está sendo limitada por MaxMV;
 - O bit MV_MinMV é ativado quando a MV está sendo limitada por MinMV;
 - O bit MV_P_MaxVarMV é ativado quando uma variação positiva da MV está sendo limitada por MaxVarMV;
 - O bit MV_N_MaxVarMV é ativado quando uma variação negativa da MV está sendo limitada por MaxVarMV.
- ProportionalAction - Mostra a ação proporcional calculada pelo bloco de função PIDA_REAL e é meramente informativo.
 - DerivativeAction - Mostra a ação derivativa calculada pelo bloco de função PIDA_REAL e é meramente informativo.
 - IntegralAction - Mostra a ação integral calculada pelo bloco de função PIDA_REAL e é meramente informativo.
 - SampleTime - O FB calcula automaticamente o ciclo utilizado para chamar o bloco lendo o intervalo da tarefa e mostra seu valor (em [s]) nesta variável de saída apenas para fins informativos.

Aplicação do Bloco Funcional PIDA_INT

O PIDA_INT funciona de forma semelhante ao PIDA_REAL, descrito anteriormente. Ambos os blocos são usados para controle PID e suportam recursos como configurações em cascata e controle de restrição.

No PIDA_REAL, todos os parâmetros numéricos têm o tipo REAL. No PIDA_INT, por sua vez, a maioria dos parâmetros numéricos tem o tipo INT, e poucos parâmetros têm o tipo REAL. Se o usuário selecionar PIDA_INT em vez de PIDA_REAL, as seguintes vantagens podem ser alcançadas:

- Cada instância consome menos memória. Isso pode ser importante em aplicações com controladores redundantes com centenas de instâncias PIDA_INT, para economizar tempo de sincronização de memória entre os CPs redundantes;
- Para alguns protocolos de comunicação, como o MODBUS TCP, a quantidade de bytes trocados ciclicamente com o sistema SCADA pode ser significativamente reduzida, especialmente se a aplicação no controlador tiver centenas de instâncias PIDA_INT;
- A aplicação fica mais fácil quando o CP gerencia unidades de campo, e as conversões entre unidades de engenharia e unidades de campo são executadas pelo sistema SCADA.

Por outro lado, PIDA_INT não deve ser usado se o CP precisar gerenciar unidades de engenharia internamente. O significado dos parâmetros do bloco funcional é o mesmo em PIDA_REAL e PIDA_INT. Portanto, para usar PIDA_INT, o usuário deve ler a documentação do PIDA_REAL e considerar que alguns parâmetros mudaram seu tipo de REAL para INT.

Os seguintes parâmetros de entrada têm tipo REAL em PIDA_REAL e tipo INT em PIDA_INT: SP, SP_CASC, PV, BIAS, ManualMV, TRK_VAL, MaxVarMV, MaxMV, MinMV, DeadBand, MaxPV e MinPV. Em relação aos parâmetros de saída, somente o MV alterou o seu tipo para INT.

Aplicação do Bloco Funcional PIDA_TUNE_REAL

O PIDA_TUNE_REAL funciona de forma semelhante ao PIDA_REAL para controle PID, mas possui a sintonização automática como uma função adicional. Essa função calcula automaticamente os parâmetros de sintonização para o controlador PID (K_p , T_d , K_d e T_i , K_i).

Todos os parâmetros que existem no PIDA_REAL também existem no PIDA_TUNE_REAL, com os mesmos tipos e significados. Além disso, o PIDA_TUNE_REAL possui alguns parâmetros adicionais para gerenciar a função de sintonização automática.

Princípio de Funcionamento da Solução de Sintonização Automática

Esta função de sintonização automática é baseada no método de relé e tem como objetivo calcular os parâmetros de sintonização do controlador PID (K_p , T_d , K_d e T_i , K_i). A função pode ser iniciada no modo manual ou no modo automático, mas o modo manual é mais seguro. Antes de iniciar a função, o usuário deve verificar se o processo está estável (PV e MV estáveis). Na figura a seguir, assume-se que o PV está estável em RefPV e o MV em RefMV. Algum ruído pode existir no PV, mas as oscilações do PV devido a esse ruído devem ser menores que o parâmetro de histerese configurado pelo usuário para a função. Na figura a seguir, assume-se que a histerese configurada é H, definindo uma janela de tamanho $2 * H$ ao redor de RefPV.

A função é iniciada no tempo t_0 na figura e assume que o valor atual de MV é RefMV. Durante 3 segundos (de t_0 a t_1), a função calcula RefPV como o valor médio de PV neste intervalo e verifica se PV permanece dentro da faixa $RefPV \pm H$ durante este intervalo. Se o PV sair desta janela, a sintonização automática é abortada. O usuário também deve configurar um limite de segurança (S na figura) ao redor de RefPV para evitar grandes oscilações do PV. Este limite de segurança define uma janela de tamanho $2 * S$ ao redor de RefPV. Após t_1 até o final da sintonização automática, PV deve permanecer dentro da faixa $RefPV \pm S$. Se o PV sair desta janela, a sintonização automática também é abortada.

A função de sintonização automática controla MV. Durante a execução da função, MV pode assumir três valores diferentes: RefMV (valor inicial); $RefMV + M$ ou $RefMV - M$. O valor de M vem de outro parâmetro de configuração do usuário. De t_1 a t_2 , a sintonização automática faz algumas inicializações. Após t_2 , a função realiza vários ciclos de medição. No exemplo mostrado na figura, 3 ciclos foram configurados pelo usuário (primeiro ciclo de t_2 a t_3 , segundo ciclo de t_3 a t_4 , terceiro ciclo de t_4 a t_5). O objetivo de ter vários ciclos é fazer médias das medições e, portanto, produzir melhores resultados de sintonização. Se o usuário desejar acelerar a sintonização automática, pode ser configurado um único ciclo.

Após o término do último ciclo (tempo t_5 na figura), os resultados da sintonização automática são calculados. Os resultados podem ser válidos ou inválidos. Diagnósticos indicam a validade dos resultados, bem como o motivo de possíveis falhas. Os parâmetros de sintonização calculados não são copiados automaticamente sobre os ganhos atuais. O usuário deve fornecer um pequeno trecho de código para fazer essa cópia.

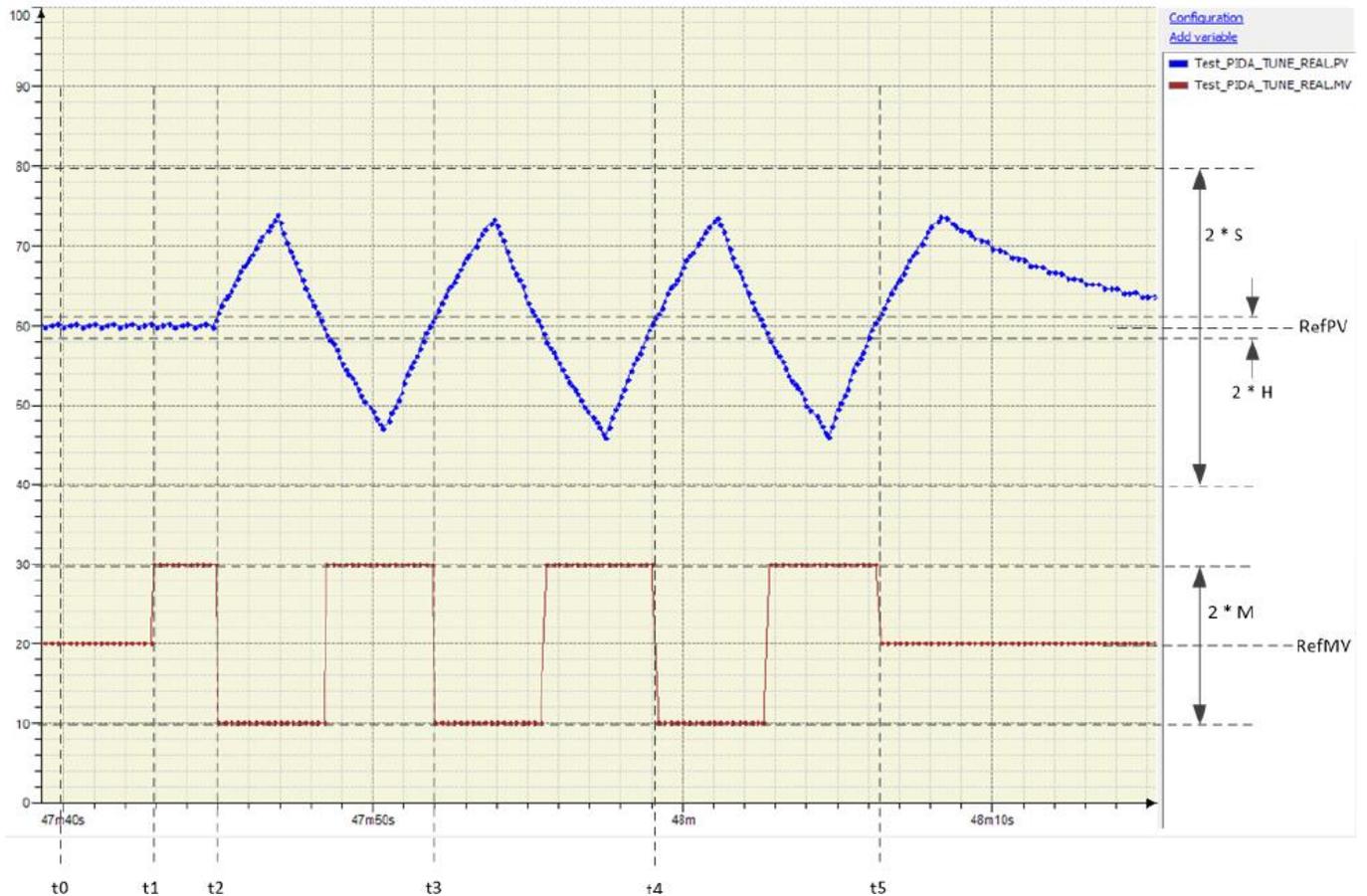


Figura 5-10. Ilustração da técnica de sintonização automática

Essa função de sintonização automática não ajustará adequadamente qualquer tipo de processo. As principais limitações conhecidas são as seguintes:

- Funciona melhor para processos que podem ser aproximados a um processo linear de primeira ordem com tempo morto. Esse tipo de processo é caracterizado por um ganho estático, um tempo morto e uma constante de tempo;
- Pode gerar resultados de sintonização inadequados se o tempo morto for muito curto em comparação com a constante de tempo ou em comparação com o tempo de amostragem do CP;
- Também pode gerar resultados de sintonização inadequados se o tempo morto for muito longo em comparação com a constante de tempo;
- Antes de iniciar a sintonização automática, o processo deve estar estável, ou seja, PV e MV devem ser constantes;
- Durante a execução da sintonização automática, o processo não deve sofrer perturbações significativas; ou seja, apenas MV deve influenciá-lo.

A função de sintonização automática tentará detectar automaticamente condições em que os resultados de sintonização calculados não são bons. No entanto, algumas situações que produzem resultados de sintonização inadequados podem não ser detectadas automaticamente. Portanto, é importante seguir algumas recomendações de segurança antes de utilizar os resultados de sintonização calculados pela função:

- Embora a sintonização automática possa ser iniciada tanto no modo automático quanto no manual, é mais seguro iniciá-la no modo manual. Se executada no modo automático, um controle inadequado pode começar assim que os parâmetros de sintonização calculados forem copiados para os parâmetros ativos de sintonização PIDA_TUNE_REAL;
- Durante a sintonização automática, é importante limitar a oscilação do PV, usando um parâmetro de configuração (consulte S na figura anterior);
- Antes de executar o PIDA_TUNE_REAL no modo automático pela primeira vez, após utilizar os parâmetros de sintonização calculados pela sintonização automática, é recomendável criar um comando e um trecho de código que retorne ao modo manual com um valor seguro em ManualMV. Esse trecho de código pode ser iniciado por um comando do SCADA ou por um botão conectado a uma entrada digital.

Parâmetros de Entrada Adicionais do PIDA_TUNE_REAL

- AutoTune - Este parâmetro de entrada é utilizado para iniciar, encerrar e abortar a função de sintonização automática. Quando AutoTune é FALSE, a função de sintonização automática não está em execução e os seguintes parâmetros de saída assumem os valores padrão de reinicialização: AutoTuneDone é FALSE, AutoTuneError é FALSE e todos os resultados de sintonização automática que compõem AutoTuneResult são reinicializados (0 para valores numéricos, FALSE para valores booleanos). Na borda de subida de AutoTune, a função de sintonização automática é iniciada. Enquanto AutoTune for TRUE, a saída MV de PIDA_TUNE_REAL é controlada pela função de sintonização automática. MV é alternado pela função de sintonização automática entre três valores possíveis: RefMV, RefMV + M e RefMV – M. Se o usuário redefinir a entrada AutoTune para FALSE antes que a função de sintonização automática termine o último ciclo, a função é abortada, e MV retorna para RefMV. Após isso, MV é controlada por PIDA_TUNE_REAL no modo manual ou automático, e assim MV pode possivelmente mudar. Quando a função de sintonização automática termina o último ciclo, ou se aborta devido à detecção de um erro:
 - O parâmetro de saída AutoTuneDone muda para TRUE para indicar que a função completou o cálculo dos resultados;
 - O parâmetro de saída AutoTuneError é FALSE se a sintonização automática terminou normalmente sem erros; caso contrário, é definido como TRUE;
 - O parâmetro de saída AutoTuneResult mostra os parâmetros de sintonização calculados e os erros detectados (se houver);
 - MV retorna para RefMV.

É importante observar que a entrada AutoTune mantém o valor como TRUE depois que a função de sintonização automática termina (AutoTuneDone = TRUE). Normalmente, um pequeno código do usuário deve ser criado para copiar os parâmetros de sintonização calculados sobre os parâmetros de sintonização atuais, se nenhum erro for detectado (AutoTuneDone = TRUE e AutoTuneError = FALSE). Após isso, a entrada AutoTune pode ser redefinida, e então PIDA_TUNE_REAL retomará o controle de MV novamente no modo automático ou manual.

- AutoTuneParam - Este parâmetro de entrada é uma estrutura que contém os parâmetros de configuração para a sintonização automática.
 - PercAmpMV - Este parâmetro de configuração indica a amplitude da oscilação de MV (0,1% a 50%) durante a sintonização automática, expressa como uma porcentagem da faixa de MV (MaxMV - MinMV). Este parâmetro é utilizado para calcular o valor de M:

$$M = \text{PercAmpMV} \times \frac{(\text{MaxMV} - \text{MinMV})}{100}$$

Se este parâmetro for muito pequeno, os picos correspondentes em PV também podem ser muito pequenos. Nesse caso, a função de sintonização automática pode detectar um erro. Se este parâmetro for muito grande, os picos correspondentes em PV também podem ser muito altos. Nesse caso, a função de sintonização automática também pode detectar um erro.

- PercHystPV - Este parâmetro de configuração indica a histerese do PV durante a sintonização automática, expressa como uma porcentagem da faixa de PV (MaxPV - MinPV), ou seja, 0,01% a 1%. Este parâmetro é utilizado para calcular o valor de H:

$$H = \text{PercHystPV} \times \frac{(\text{MaxPV} - \text{MinPV})}{100}$$

Este parâmetro deve ser maior que o nível de ruído no PV, caso contrário, a função de sintonização automática pode detectar um erro. Por outro lado, ele deve ser o menor possível para produzir resultados de sintonização mais precisos e para evitar outros erros que podem ser detectados pela função.

- PercMaxPeakPV - Este parâmetro de configuração indica o pico máximo de PV durante a sintonização automática para segurança, expresso como uma porcentagem (0,1% a 50%) da faixa de PV (MaxPV - MinPV). Este parâmetro é utilizado para calcular o valor de S.

$$S = \text{PercMaxPeakPV} \times \frac{(\text{MaxPV} - \text{MinPV})}{100}$$

Se um pico de PV sair da faixa $\text{RefPV} \pm S$, a sintonização automática é abortada e MV retorna ao valor inicial RefMV. Uma boa configuração deste parâmetro é importante para a segurança, pois limita as oscilações do PV durante a sintonização automática.

- NumCycles - Este parâmetro de configuração indica o número de ciclos de medição (1 a 10) usados pela função de sintonização automática para a média dos resultados. No exemplo abordado anteriormente, seu valor foi definido como 3. Valores maiores devem produzir resultados mais exatos, mas levam mais tempo.
- Mode - Este parâmetro de configuração indica a configuração do PID (P, PI ou PID). Outras configurações (D, PD, I etc.) não são suportadas pela função de sintonização automática. A enumeração AUTOTUNE_MODE está detalhada na sequência.

```

TYPE AUTOTUNE_MODE :
(
  AUTOTUNE_PID := 0, // PID configuration
  AUTOTUNE_PI  := 1, // PI configuration
  AUTOTUNE_P   := 2  // P configuration
);
END_TYPE

```

Parâmetros de Saída Adicionais do PIDA_TUNE_REAL

- AutoTuneDone - Este parâmetro de saída indica que a função de sintonização automática foi concluída e os resultados foram calculados no parâmetro de saída AutoTuneResult. Pode ser concluída com ou sem erros.
- AutoTuneError - Este parâmetro de saída indica que a função de sintonização automática foi encerrada (ou abortada) devido a um erro detectado pela função. Nesse caso, o parâmetro de saída AutoTuneResult fornece mais informações sobre o(s) erro(s) detectado(s).

- AutoTuneResult - Este parâmetro de saída é uma estrutura (AUTOTUNE_RESULT) que contém os parâmetros de sintonização calculados e indicações de erro:
 - Kp – Ganho proporcional calculado pela sintonização automática.
 - Ti_Ki - Tempo integral ou ganho integral calculado pela sintonização automática. Se o parâmetro de entrada Indep for TRUE durante a sintonização automática, um ganho é calculado; caso contrário, um tempo é calculado.
 - Td_Kd - Tempo derivativo ou ganho derivativo calculado pela sintonização automática. Se o parâmetro de entrada Indep for TRUE durante a sintonização automática, um ganho é calculado; caso contrário, um tempo é calculado.
 - Bad_PercAmpMV - Este bit retorna TRUE se o parâmetro de entrada AutoTuneParam.PercAmpMV estiver fora da faixa permitida.
 - Bad_PercHystPV - Este bit retorna TRUE se o parâmetro de entrada AutoTuneParam.PercHystPV estiver fora da faixa permitida.
 - Bad_PercMaxPeakPV - Este bit retorna TRUE se o parâmetro de entrada AutoTuneParam.PercMaxPeakPV estiver fora da faixa permitida.
 - Bad_NumCycles - Este bit retorna TRUE se o parâmetro de entrada AutoTuneParam.NumCycles estiver fora da faixa permitida.
 - Bad_Mode - Este bit retorna TRUE se o parâmetro de entrada AutoTuneParam.Mode estiver fora da faixa permitida.
 - MV_TOO_HIGH - Este bit retorna TRUE se $(RefMV + M)$ for maior que MaxMV.
 - MV_TOO_LOW - Este bit retorna TRUE se $(RefMV - M)$ for menor que MinMV.
 - PV_TOO_HIGH - Este bit retorna TRUE se $(RefPV + H)$ for maior que MaxPV.
 - PV_TOO_LOW - Este bit retorna TRUE se $(RefPV - H)$ for menor que MinPV.
 - PEAK_TOO_HIGH - Este bit retorna TRUE se PV sair da faixa $(RefPV \pm S)$ durante os ciclos de medição da sintonização automática. A função é abortada por segurança, e MV retorna para RefMV.
 - PEAK_TOO_LOW - Este bit retorna TRUE se os picos de PV durante os ciclos de medição forem muito baixos em comparação com a histerese H (consulte a Figura 14). Nesse caso, os parâmetros de sintonização calculados não podem ser confiáveis. Isso pode ocorrer, por exemplo, quando o tempo morto é muito baixo em comparação com a constante de tempo do processo ou quando o ganho estático do processo é muito baixo. Às vezes, é possível evitar esse erro aumentando o valor do parâmetro AutoTuneParam.PercAmpMV ou diminuindo o valor do parâmetro AutoTuneParam.PercHystPV.
 - PV_NOT_STEADY - Este bit retorna TRUE se o PV sair da faixa $(RefPV \pm H)$ nos primeiros 3 segundos após o início da sintonização automática. Durante este intervalo, a função verifica se o PV está estável. É possível evitar esse erro aumentando o valor do parâmetro AutoTuneParam.PercHystPV.
 - CYCLE_TOO_FAST - Este bit retorna TRUE se a duração de um ciclo de medição for muito curta em comparação com o tempo de ciclo do PLC (parâmetro de saída SampleTime). Nesse caso, os parâmetros de sintonização calculados não podem ser confiáveis. Isso normalmente acontece quando o tempo morto é muito baixo em comparação com a constante de tempo do processo.
 - HIGH_DEAD_TIME - Este bit retorna TRUE se a duração de um ciclo de medição for muito curta em comparação com o tempo morto medido. Nesse caso, os parâmetros de sintonização calculados não podem ser confiáveis. Isso pode acontecer quando o tempo morto é muito maior em comparação com a constante de tempo do processo.

Interface SCADA Recomendada e Código para Copiar os Parâmetros de Sintonização Calculados

Após o término da sintonização automática, os parâmetros de sintonização calculados não são copiados automaticamente para os parâmetros de sintonização ativos do PIDA_TUNE_REAL. O método a seguir é recomendado para copiar os parâmetros de sintonização:

1. Crie um comando como CopyParam (variável booleana) que deve ser definido como TRUE pelo usuário (por exemplo, no sistema SCADA) para iniciar a cópia dos parâmetros de sintonização calculados.
2. Utilize um trecho de código semelhante ao mostrado na sequência a seguir (neste exemplo, em linguagem ST - Structured Text).

```
IF CopyParam THEN
  CopyParam := FALSE;
  AutoTune := FALSE;
  IF AutoTuneDone AND NOT(AutoTuneError) THEN
    Kp := AutoTuneResult.Kp;
    Ti_Ki := AutoTuneResult.Ti_Ki;
    Td_Kd := AutoTuneResult.Td_Kd;
  END_IF
END_IF
```

Aplicação do Bloco Funcional PIDA_TUNE_INT

O PIDA_TUNE_INT funciona de maneira semelhante ao PIDA_TUNE_REAL, descrito anteriormente. Ambos os blocos são utilizados para controle PID com a função de sintonização automática. No PIDA_TUNE_REAL, todos os parâmetros numéricos têm o tipo REAL. No PIDA_TUNE_INT, a maioria dos parâmetros numéricos tem o tipo INT, enquanto alguns têm o tipo REAL. Se o usuário escolher o PIDA_TUNE_INT em vez do PIDA_TUNE_REAL, algumas vantagens podem ser alcançadas:

- Menor Consumo de Memória - Cada instância consome menos memória. Isso pode ser crucial em aplicações PLC redundantes com centenas de instâncias PIDA_TUNE_INT, economizando tempo de sincronização de memória entre os controladores redundantes;
- Redução na Troca de Dados - Para alguns protocolos de comunicação, como o MODBUS TCP, a quantidade de bytes trocados ciclicamente com o sistema SCADA pode ser significativamente reduzida, especialmente se a aplicação PLC tiver centenas de instâncias PIDA_TUNE_INT;
- Facilidade na Gestão de Unidades de Campo - A aplicação fica mais fácil quando o PLC gerencia unidades de campo, e as conversões entre unidades de engenharia e unidades de campo são executadas pelo sistema SCADA.

Por outro lado, o PIDA_TUNE_INT não deve ser usado se o PLC precisar gerenciar unidades de engenharia internamente. O significado de cada um dos parâmetros, quando comparado entre PIDA_TUNE_REAL e PIDA_TUNE_INT, é o mesmo. Portanto, para usar o PIDA_TUNE_INT, o usuário deve ler a documentação do PIDA_TUNE_REAL e apenas considerar que alguns parâmetros mudaram seu tipo de REAL para INT.

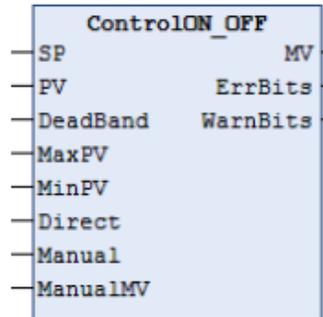
Os seguintes parâmetros de entrada têm o tipo REAL no PIDA_TUNE_REAL e o tipo INT no PIDA_TUNE_INT: SP, SP_CASC, PV, BIAS, ManualMV, TRK_VAL, MaxVarMV, MaxMV, MinMV, DeadBand, MaxPV e MinPV. Em relação aos parâmetros de saída, somente o MV alterou o seu tipo para INT.



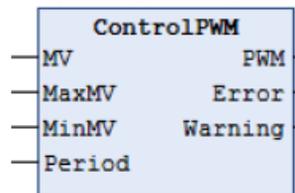
Pesquisa na Documentação: Elementos de Software de Sistemas de Controle

Consulte, na documentação do produto, os elementos de software que são utilizados em aplicações de controle complementarmente ao PID conforme indicado na sequência. Caracterize os parâmetros de entrada e saída dos blocos funcionais e funções associadas.

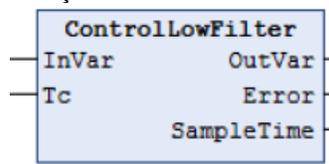
- ControlON_OFF: é utilizado para controle ON-OFF (Liga-Desliga).



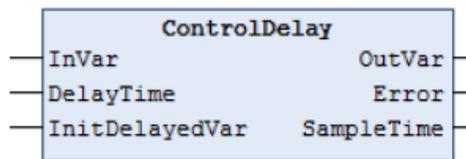
- ControlPWM: é utilizado para converter uma saída analógica típica (MV, tipo REAL) em uma saída modulada por largura de pulso (PWM, tipo BOOL).



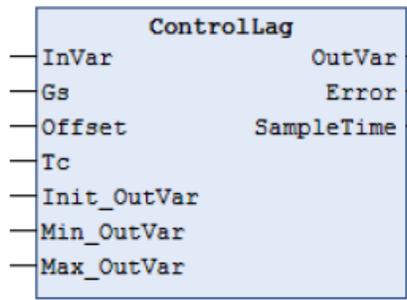
- ControlLowFilter: implementa um filtro de primeira ordem com uma constante de tempo (Tc). Pode ser utilizado, por exemplo, para filtrar ruídos na entrada PV antes de conectá-la a um bloco de função de controle PID.



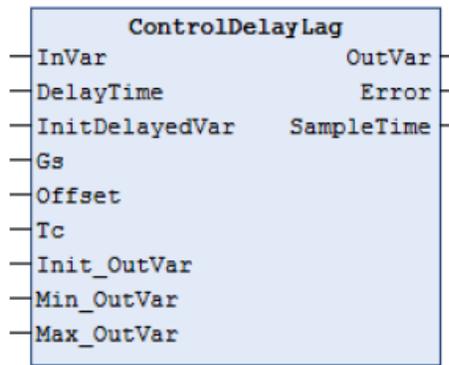
- ControlDelay: cria um atraso, especificado em segundos, para uma variável de entrada do tipo REAL. Ele deve ser chamado em uma tarefa cíclica. O atraso é limitado a MaxSamplesDelay períodos de amostragem deste bloco funcional, devido a limites de alocação de memória. MaxSamplesDelay é um parâmetro que pode ser ajustado na biblioteca NextoPID.



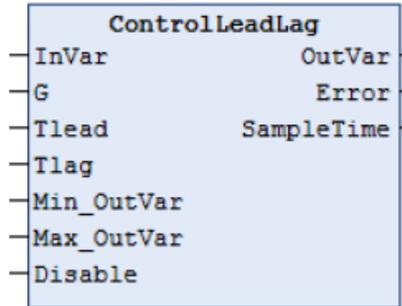
- ControlLag: aplica um atraso de primeira ordem a uma entrada. Pode ser utilizado, por exemplo, para simular um processo de primeira ordem sem tempo morto.



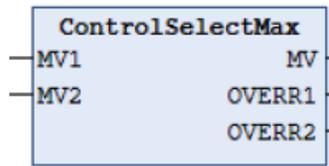
- ControlDelayLag: combina um bloco funcional ControlDelay com um ControlLag. Pode ser usado, por exemplo, para simular um processo de primeira ordem com tempo morto. A variável de entrada é primeiro atrasada e, em seguida, um atraso é aplicado ao sinal atrasado. Este bloco deve ser chamado em uma tarefa cíclica.



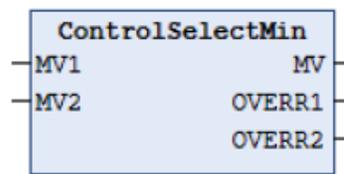
- ControlLeadLag: pode ser utilizado como componente em algumas estratégias avançadas de controle, como o feed-forward.



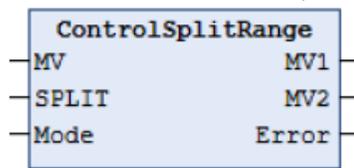
- ControlSelectMax: esta função é usada em conjunto com dois blocos funcionais de controle PID (PIDA_REAL, PIDA_INT, PIDA_TUNE_REAL, PIDA_TUNE_INT) em uma configuração com restrições, onde a saída máxima de dois controles PID (MV1 e MV2) deve ser selecionada.



- ControlSelectMin: esta função é usada em conjunto com dois blocos funcionais de controle PID (PIDA_REAL, PIDA_INT, PIDA_TUNE_REAL, PIDA_TUNE_INT) em uma configuração com restrições, onde a saída mínima de dois controles PID (MV1 e MV2) deve ser selecionada.



- ControlSplitRange: esta função é usada quando a saída MV de um bloco de controle PID (PIDA_REAL, PIDA_INT, PIDA_TUNE_REAL, PIDA_TUNE_INT) precisa ser dividida para comandar duas válvulas diferentes, controladas pelas saídas MV1 e MV2.



Fonte: *Advanced Control Functions User Manual* - MU214610



Pesquisa na Documentação: Dicas de Sintonização do PID

Consulte, na documentação do produto, algumas dicas básicas para sintonizar controladores PID, incluindo: caracterização do processo em malha aberta, unidades de campo e engenharia, controlabilidade de um processo, máximo tempo de amostragem do controlador, seleção dos modos do PID, método de síntese, minimização da integral do erro absoluto e sintonização do PID em cascata.

A sintonia de controladores PID pode ser uma tarefa complexa para alguns processos. Portanto, tenha em mente que essas dicas são muito básicas e podem não ser úteis para todos os tipos de processos. Essas dicas se concentram em processos lineares de primeira ordem com tempo morto. Aliás, esse tipo de processo pode ser simulado com o bloco funcional ControlDelayLag.

Fonte: *Advanced Control Functions User Manual* - MU214610

Bloco Funcional Datalogger



O bloco funcional Datalogger permite ao usuário criar seus próprios registros de dados e gravar em arquivos de log no cartão de memória ou na memória interna presente na CPU. Os arquivos são gerados em um diretório específico do cartão de memória ou na memória interna da CPU no formato CSV, permitindo a visualização em editores de texto e planilhas. A figura a seguir mostra o bloco e suas entradas e saídas.



Modos de registro dos dados

O Datalogger suporta dois modos de registro dos dados: o modo contínuo e o modo com trigger.

- Modo Contínuo** - Neste modo, os dados são registrados e escritos no arquivo de log continuamente em um intervalo de amostragem definido pelo usuário. O número de registros por arquivo é de no máximo 65535 (FILE_RECORDS) e no mínimo 100, sendo o limite de arquivos (NUMBER_OF_FILES) limitado pela memória ou em 65535 arquivos, que é o valor máximo que a variável pode assumir. Esse modo suporta a funcionalidade CIRCULAR, que será explicada posteriormente.
- Modo Trigger** - Idem ao modo contínuo, porém os dados serão escritos no arquivo de log somente na ocorrência de um trigger. Ao ocorrer um trigger, um novo arquivo é criado, onde serão escritos os registros anteriores e posteriores à ocorrência, conforme definição do usuário. É possível armazenar 2001 registros por arquivo, ou seja, 1000 registros correspondentes ao TRIGGER_BEFORE, 1000 ao TRIGGER_AFTER e mais 1 no momento do trigger. No modo TRIGGER, o número máximo de arquivos (NUMBER_OF_FILES) também será limitado pelo espaço de memória e limite da variável, que é de 65535 arquivos, isso se deve ao fato de que cada borda de subida na entrada TRIGGER cria um arquivo. Esse modo também suporta a funcionalidade CIRCULAR.

Características Gerais

O Datalogger é composto por 268 entradas e 6 saídas. Ele suporta até 256 entradas para o registro dos dados que podem assumir qualquer tipo numérico com resolução menor ou igual ao tipo REAL (32 bits).

A entrada HEADER é de preenchimento obrigatório, pois o número de entradas de dados é calculado pela quantidade de cabeçalhos definidos nesta entrada.

A consistência das entradas é feita na borda de subida da entrada ENABLE, caso algum parâmetro não atenda as especificações, um erro é informado e o bloco função só vai funcionar corretamente após a correção desse parâmetro e uma nova borda de subida na entrada ENABLE. O bloco funcional Datalogger não pode ser utilizado na área de dados redundantes de projetos redundantes. O mesmo deve ser utilizado em POU's não redundantes, como a POU NonSkippedPrg.

Parâmetros de Entrada do Datalogger

Tabela 5-6. Parâmetros de entrada do Datalogger

Parâmetros de entrada	Tipo	Descrição
ENABLE	BOOL	Habilita o FB Datalogger. TRUE=Habilitado FALSE=Bloqueado
FILE_NAME	STRING[20]	Nome do arquivo sem a extensão CSV. O nome do arquivo ficará limitado aos primeiros 20 caracteres da string.
STORAGE	DATALOGGER_STORAGE	Define o dispositivo onde os registros serão armazenados. INTERNAL_MEMORY: os dados serão armazenados na memória interna da UCP. MEMORY_CARD: os dados serão armazenados no cartão de memória inserido na UCP.
FILE_RECORDS	UINT	Número de registros que serão escritos no arquivo. Intervalo de valores permitidos: de 100 até 65535. Parâmetro válido para o modo contínuo.
NUMBER_OF_FILES	UINT	Número máximo de arquivos a serem armazenados. Intervalo permitido: de 1 até 65535.
MODE	DATALOGGER_MODE	Modo de funcionamento do Datalogger. CONTINUOUS_MODE: o armazenamento dos registros será contínuo. Se nada for configurado, esse modo é definido como padrão. TRIGGER_MODE: o armazenamento será ativado por um disparo externo.
CIRCULAR	BOOL	Habilita o modo circular. TRUE = Habilitado FALSE = Desabilitado
TRIGGER_BEFORE	UINT	Número de registros anteriores à ocorrência do disparo do trigger que devem ser escritos no arquivo. Intervalo de valores permitidos: de 1 até 1000. Parâmetro válido para o modo trigger.
TRIGGER_AFTER	UINT	Número de registros posteriores à ocorrência do disparo do trigger que devem ser escritos no arquivo. Intervalo de valores permitidos: de 1 até 1000. Parâmetro válido para o modo trigger.
TRIGGER	BOOL	Entrada do disparo externo, ativado por uma borda de subida.
REC_INTERVAL	TIME	Intervalo em que os dados serão registrados. Valores permitidos: ≥ 10ms no modo Trigger ≥ 100ms no modo Contínuo
FILE_HEADER	STRING[3327]	Cabeçalhos dos dados que serão armazenados no arquivo. Campo de preenchimento obrigatório, com os cabeçalhos separados por ponto e vírgula. O número máximo de caracteres suportado por esta entrada é de 3327. Este valor permite configurar até 256 cabeçalhos com 12 caracteres cada.
INPUT1...INPUTN, N≤256	REAL	Entrada de dados. O Datalogger suporta até 256 entradas de dados.

Parâmetros de Saída do Datalogger

Tabela 5-7. Parâmetros de saída do Datalogger

Parâmetros de saída	Tipo	Descrição
DONE	BOOL	Indica o estado do bloco função: TRUE: a função foi executada. FALSE: a função está sendo executada.
BUSY	BOOL	TRUE: função está registrando os dados na memória. FALSE: função não está registrando os dados na memória, ou entrou no estado de erro.
ERROR	BOOL	TRUE: ocorrência de erro. Função abortada. FALSE: nenhum erro.
STATUS	DATALOGGER_STATUS	Status do Datalogger, descritos na próxima tabela.
CURRENT_RECORDS	UDINT	Número de registros desde o início do processo.
MISSED_TRIGGERS	UDINT	Contador de disparos ignorados pelo Datalogger durante período em que a função está funcionando no modo trigger.

Parâmetros de Status do Datalogger

Tabela 5-8. Parâmetros de status do Datalogger

Tipo de dado	Opção	Descrição
STATUS	NO_ERROR	Nenhum erro. Bloco função executado com sucesso.
	ERROR_FILE_OPEN	Erro ao abrir arquivo.
	ERROR_FILE_WRITE	Erro durante a escrita no arquivo.
	ERROR_FILE_CLOSE	Erro ao fechar arquivo.
	ERROR_OVERLOAD	Erro que indica sobrecarga do buffer.
	ERROR_NUMBER_OF_FILES_REACHED	Erro que indica que o número máximo de arquivos foi atingido.
	ERROR_STORAGE_DEVICE_UNMOUNTED	Erro que indica que o dispositivo de armazenamento não está montado. Exemplo: sem o cartão de memória.
	ERROR_NOSPACE	Indica que não tem mais espaço disponível no dispositivo de armazenamento.
	ERROR_RTC_READING	Erro na leitura de data e hora.
	ERROR_INTERNAL	Erro interno.
	INVALID_FILE_NAME	Nome do arquivo é inválido.
	INVALID_FILE_HEADER	Header Inválido. Este erro pode acontecer caso a entrada HEADER esteja vazia ou contendo um número de cabeçalhos maior que 255.
	INVALID_STORAGE	Indica que o dispositivo de armazenamento é inválido.
	INVALID_REC_INTERVAL	Intervalo de amostragem inválido.
	INVALID_MODE	Modo inválido
	INVALID_TRIGGER_BEFORE	Entrada fora do intervalo especificado.
INVALID_TRIGGER_AFTER	Entrada fora do intervalo especificado.	
INVALID_FILE_RECORDS	Entrada fora do intervalo especificado.	
INVALID_NUMBER_OF_FILES	Indica que a entrada do número de arquivos é inválida.	

Layout do arquivo gerado

A tabela a seguir mostra o layout do arquivo gerado pelo Datalogger para o modo contínuo. Todos os registros são ordenados de forma sequencial. A primeira coluna é formada pelo número do registro em ordem crescente, a segunda coluna é composta pela data e hora (UTC) em que os registros foram coletados. Os valores dos processos coletados pelo Datalogger são inseridos da coluna Header1 até a Header256 de forma sequencial conforme definido nas entradas do bloco funcional, isto é, os dados da coluna Header1 corresponde a entrada INPUT1, os dados da coluna Header2 corresponde a entrada INPUT2 e assim sucessivamente até o Header256, correspondente a entrada INPUT256.

Tabela 5-9. Formato do arquivo gerado pelo Datalogger para o modo contínuo

Record	Date	Time	Header1	...	Header256
1	17/11/2017	15:00:00.000	10	...	177.0
2	17/11/2017	15:00:00.100	20	...	178.0
...
100	17/11/2017	15:00:09.900	1000	...	276.0

A tabela a seguir mostra o layout do arquivo para o modo trigger. Neste modo, a coluna Record é preenchida com valores negativos para indicar que estes registros foram coletados antes do disparo do trigger. Da mesma forma como no modo contínuo, os registros são ordenados em ordem crescente. No exemplo descrito na tabela 5 levou-se em conta que o usuário tenha configurado a entrada TRIGGER_BEFORE com valor 50 e a entrada TRIGGER_AFTER com valor 100. Vale lembrar que o total de registros coletados neste modo é igual a TRIGGER_BEFORE + TRIGGER_AFTER + 1. O valor 1 somado no final corresponde ao registro coletado no momento do disparo do trigger (registro número 0).

Tabela 5-10. Formato do arquivo gerado pelo Datalogger para o modo trigger

Record	Date	Time	Header1	...	Header256
-50	17/11/2017	15:00:00.000	10	...	0.5
-49	17/11/2017	15:00:00.100	20	...	0.52
...
-1	17/11/2017	15:00:04.900	500	...	0.7
0	17/11/2017	15:00:05.000	510	...	0.75
1	17/11/2017	15:00:05.100	520	...	0.82
...
100	17/11/2017	15:00:15.000	1510	...	0.9

Para visualizarmos de forma correta a coluna Time nas ferramentas de edição e visualização (Microsoft Excel, LibreOffice Calc, entre outros), a célula deve estar no seguinte formato: hh:mm:ss.000.

Estrutura do diretório

Por padrão, os arquivos são salvos na memória interna (diretório InternalMemory/Datalogger). Caso a entrada STORAGE seja configurada com o valor MEMORY_CARD os arquivos serão salvos no cartão de memória (diretório MemoryCard/Datalogger).

Por exemplo, se o bloco função estiver com a entrada STORAGE configurada como INTERNAL_MEMORY, a estrada MODE configurada como CONTINUOUS_MODE, após uma borda de subida na entrada ENABLE no dia 17/11/2017 a partir das 15:00:01 no horário UTC, ficarão no diretório InternalMemory /Datalogger /2017_11_17_15_00_01. Horário UTC é utilizado para se evitar os possíveis conflitos durante o horário de verão e UCP's em diferentes zonas de horários.

Nome do arquivo

O arquivo será nomeado com o seguinte formato FILE_NAME_XXXXX.csv, com XXXXX variando de 00001 até o valor definido para NUMBER_OF_FILES. Por exemplo, podemos supor que um arquivo cujo nome é MyFileName, o primeiro arquivo armazenado ficará da seguinte forma: MyFileName_00001. O nome do arquivo não pode conter os caracteres ".", "/" e "~". Caso contrário é gerado o erro INVALID_FILE_NAME.

Intervalo de valores para o formato de saída

O intervalo de valores mínimo e máximo, para o formato de saída, vai de -10000000 a 10000000 para o tipo DINT. Como todos os tipos numéricos são convertidos para o tipo REAL e posteriormente para STRING, o limite deste tipo é propagado para os outros tipos de dados, que estão descritos na tabela a seguir.

Tabela 5-11. Tipos suportados pelas entradas de dados e os respectivos intervalo de valores

Tipo de dado	Limite inferior	Limite superior
BIT	0	1
BYTE	-	-
WORD	-	-
DWORD	-	-
LWORD	-	-
SINT	-128	127
USINT	0	255
INT	-32768	32767
UINT	0	65535
DINT	-10000000	10000000
UDINT	0	10000000
LINT	-10000000	10000000
ULINT	0	10000000
REAL	-10000000	10000000
LREAL	-10000000	10000000

Funcionamento geral no modo Contínuo

Para que seja possível realizar a escrita dos dados em arquivos de log é necessário que as entradas do Datalogger estejam configuradas de forma correta, essa conferência é feita somente na borda de subida da entrada ENABLE. Além disso, deve haver espaço na memória interna ou no cartão de memória para que não ocorram problemas de escrita. Caso a entrada MODE não for definida, esse será o modo de operação por padrão.

Os parâmetros FILE_NAME, REC_INTERVAL, FILE_RECORDS, NUMBER_OF_FILES, FILE_HEADER e CIRCULAR devem ser obrigatoriamente configurados pelo usuário para que não ocorra nenhum dos erros listados na tabela 3.

A entrada FILE_NAME deve conter somente o nome do arquivo sem uso de caracteres especiais, como mencionado na seção anterior.

O usuário deverá inserir o intervalo em que as amostras serão coletadas configurando a entrada REC_INTERVAL com um intervalo mínimo de 100 milissegundos. Caso um intervalo menor seja configurado, o bloco função retornará erro INVALID_REC_INTERVAL e se o valor do

REC_INTERVAL for menor do que o intervalo da main task, a captura das entradas é feita no mesmo intervalo da main task.

A entrada FILE_RECORDS é utilizada para limitar o número de registros que um arquivo deve possuir. O usuário deve ficar atento ao limite de valores permitidos para esta entrada, caso contrário o bloco função retornará o erro INVALID_FILE_RECORDS.

As entradas de dados devem ser configuradas de acordo com o número de cabeçalhos definidos pela entrada FILE_HEADER. Cada cabeçalho deve ser separado por ponto e vírgula do cabeçalho seguinte para que o Datalogger identifique a diferença entre as entradas de dados. Os cabeçalhos devem ser ordenados da esquerda para a direita e devem seguir a mesma sequência das entradas de dados conforme mencionado na seção anterior. O usuário deve tomar o cuidado de inserir o mesmo número de entradas e de cabeçalhos, pois se o número de cabeçalhos for maior ou menor que o número de entradas de dados, as seguintes condições de operação poderão ocorrer:

- Se o número de cabeçalhos for menor que o número de entradas de dados, as entradas excedentes não serão registradas pelo bloco função;
- Se o número de cabeçalhos for maior que o número de entradas de dados, as entradas correspondentes aos cabeçalhos excedentes serão registradas com valor 0.

Além das entradas obrigatórias listadas acima, o usuário tem a opção de escolher a memória onde os arquivos serão armazenados configurando a entrada STORAGE. Caso esta entrada esteja desabilitada ou o valor configurado pelo usuário seja INTERNAL_MEMORY, os dados serão armazenados na memória interna da UCP. Se o valor da entrada for MEMORY_CARD, os dados serão armazenados no cartão de memória.

Deve-se atentar para o fato de que a opção para escrita em cartão de memória só irá funcionar em UCPs que suportam cartão de memória.

O usuário deve ficar atento ao tamanho máximo da string da entrada FILE_HEADER descrito na tabela 1. Caso esta entrada possua uma string maior que o limite máximo, a string será truncada no valor máximo ocasionando um funcionamento indesejado do bloco funcional.

Após a devida configuração das entradas, o comando para dar início ao registro dos logs deve ser acionado (borda de subida da entrada ENABLE, ou seja, de FALSE para TRUE. Uma vez que o registro de dados é iniciado, o mesmo só para se for gerado uma borda de descida na entrada ENABLE (TRUE para FALSE) do bloco função. Esta operação redefine todas as variáveis internas do bloco funcional e salva o arquivo onde estava sendo realizado o processo de escrita. Caso o usuário deixe a entrada ENABLE como TRUE indefinidamente, um novo arquivo será criado quando o número de registros escritos no arquivo atingir o valor definido na entrada FILE_RECORDS.

É possível o usuário acompanhar o número de registros coletados e escritos no arquivo através da saída CURRENT_RECORDS.

O código ilustrado a seguir foi criado no MasterTool IEC XE 3.02 e mostra a implementação de um bloco funcional Datalogger para o modo contínuo.

```

PROGRAM UserPrg
VAR
    FB_Datalogger    :LibDatalogger.DATALOGGER;
    Status           :LibDatalogger.DATALOGGER_STATUS;
    Records          :UDINT;
    EnableFB         :BOOL;
    Error            :BOOL;
    Data             :REAL;
END_VAR
Data := 5*SIN(2*3.14159*TIME_TO_REAL(TIME()));
FB_Datalogger(
    ENABLE := EnableFB,
    FILE_NAME := 'MyFileName',
    REC_INTERVAL := T#100MS,
    NUMBER_OF_FILES := 1,
    MODE := LibDatalogger.DATALOGGER_MODE.CONTINUOUS_MODE,
    CIRCULAR := 0,
    STORAGE := LibDatalogger.DATALOGGER_STORAGE.INTERNAL_MEMORY,
    FILE_RECORDS := 100,
    FILE_HEADER := 'H1;H2',
    INPUT001 := Data,
    INPUT002 := Data + 1,
    STATUS => Status,
    ERROR => Error,
    CURRENT_RECORDS => Records
);

```

Funcionamento geral no modo Circular

O modo circular possui a funcionalidade de, após atingir a quantidade máxima de arquivos, reabrir o primeiro arquivo e sobrescrevê-lo, ou seja, a aplicação fica sempre gravando de forma a manter a quantidade de arquivos, sendo assim, circular.

Para que o modo circular seja ativado, devemos colocar nessa entrada o valor "1" ou "TRUE", caso contrário "0" ou "FALSE" para desabilitá-lo.

Essa funcionalidade está disponível tanto para o modo Trigger, quanto para o Contínuo.

Funcionamento geral no modo Trigger

O modo de Trigger permite a coleta de até 1000 registros antes do disparo de um trigger, 1000 registros após o disparo e o registro coletado no momento do disparo.

Da mesma forma como foi descrito para o modo de funcionamento contínuo, as entradas FILE_NAME, STORAGE, MODE, REC_INTERVAL, FILE_HEADER, CIRCULAR, NUMBER_OF_FILES se o CIRCULAR estiver habilitado e as entradas de dados devem ser configuradas como descrito para o modo contínuo. Além disso, as seguintes entradas adicionais devem ser configuradas para que o Datalogger opere no modo trigger:

- A entrada MODE deve estar configurada com o valor TRIGGER_MODE;
- A entrada TRIGGER_BEFORE deve ser maior que 0 e menor ou igual a 1000;
- A entrada TRIGGER_AFTER deve ser maior que 0 e menor ou igual a 1000.

A entrada FILE_RECORDS não se aplica ao modo Trigger.

Após a devida configuração das entradas do bloco função, o registro dos dados só ocorre quando a entrada ENABLE estiver habilitada (valor TRUE) e após a borda de subida da entrada TRIGGER, continuando a captura dos registros mesmo se a entrada TRIGGER for desativada.

O exemplo ilustrado no código a seguir foi criado no MasterTool IEC XE 3.02 e mostra a implementação de um bloco funcional Datalogger para o modo trigger.

```

PROGRAM UserPrg
VAR
    FB_Datalogger    :LibDatalogger.DATALOGGER;
    EnableFB         :BOOL;
    TriggerFB        :BOOL;
    Data             :REAL;
    Records          :UDINT;
    Status           :LibDatalogger.DATALOGGER_STATUS;
    Error            :BOOL;
END_VAR

Data := 5*SIN(2*3.14159*TIME_TO_REAL(TIME()));

FB_Datalogger(
    ENABLE := EnableFB,
    TRIGGER := TriggerFB,
    FILE_NAME := 'MyFileName',
    REC_INTERVAL := T#100MS,
    NUMBER_OF_FILES := 1,
    MODE := LibDatalogger.DATALOGGER_MODE.TRIGGER_MODE,
    CIRCULAR := 0,
    TRIGGER_BEFORE := 100,
    TRIGGER_AFTER := 100,
    STORAGE := LibDatalogger.DATALOGGER_STORAGE.INTERNAL_MEMORY,
    FILE_HEADER := 'H1;H2',
    INPUT001 := Data,
    INPUT002 := Data + 1,
    STATUS => Status,
    ERROR => Error,
    CURRENT_RECORDS => Records
);

```

Motion Control



Criado para agregar segurança e flexibilidade a máquinas e processos que demandam controle preciso de movimento, o CLP XP350(351) Nexto Xpress tem capacidade para operar em malhas com até 4 eixos de motion control (X, Y, Z e A).

Utilizado em conjunto com os novos servodrivens de resposta rápida e alta performance da Altus, estes modelos compõem uma solução completa de motion control para aplicações em onde potência, eficiência e precisão de movimento são essenciais.

Para atender esta demanda de alta precisão em máquinas e processos, o novo controlador da Altus, assim como os servos drivers, conta com suporte a EtherCAT Master e CANopen, dois dos principais protocolos de comunicação utilizados no segmento.

Além disso, possui suporte a blocos funcionais PLCopen Motion Control Part 1 para comando de eixo único, sincronização de multi-eixos, engrenagem eletrônica (CAME), editor especial para planejar movimentos (CAM), entre outros.



Um pouco de teoria... Aplicações de controle de movimento

O controle de movimento, ou Motion Control, é um campo da automação industrial que engloba os sistemas e subsistemas responsáveis pela movimentação de partes de máquinas com alto grau de precisão.

Otimização e precisão são essenciais na automação industrial moderna para aumentar a eficiência e a flexibilidade de máquinas e processos. Em situações como essa, equipamentos capazes de controlar com precisão o movimento são essenciais para garantir precisão e reduzir o desperdício no setor.

Servo drivers e servo motores são dois produtos essenciais para aplicações de soluções de motion control.

Os servomotores são dispositivos eletromecânicos projetados para transformar sinais de controle em movimentos controlados e precisos. O servo motor é composto por um motor de corrente contínua (DC) ou corrente alternada (AC), um sensor de posição (normalmente um encoder) e um sistema de controle integrado. Ele oferece precisão, torque e resposta dinâmica excepcionais. Essas características o tornam ideal para aplicações que exigem movimentos precisos e repetitivos, como bobinas de rotulagem, tambores de enchimento, máquinas CNC, robótica industrial e impressoras 3D.

O servomotor é controlado por um dispositivo eletrônico chamado servodriver. Ele funciona como o "cérebro" do sistema, recebendo comandos do controle e enviando as instruções para o servo motor executar movimentos específicos.

O servo driver é responsável por fornecer a corrente adequada ao motor, ajustar a velocidade, a posição e a direção do movimento e monitorar o feedback do sensor de posição. Além disso, ele protege o servo motor de sobrecargas, curtos-circuitos e outras anomalias de operação.

Os servomotores e servodrivens funcionam juntos para fornecer controle preciso e dinâmico em processos de movimentação industrial. No topo do processo, o servo driver recebe comandos do CLP, uma peça central do sistema de controle, e os processa para acionar o servo motor.

Os algoritmos de controle complexos permitem que o driver interprete e altere a velocidade, a posição e a direção do movimento do motor. Além disso, o sensor de posição, que funciona com o servo motor, fornece feedback ao servo driver para assegurar que o movimento esteja de acordo com as instruções do controle.

A figura a seguir ilustra uma arquitetura típica desse tipo de aplicação.



Figura 5-11. Exemplo de arquitetura de controle de movimento

DICA: consulte o instrutor para orientações adicionais referentes à funcionalidade de controle de movimento no Nexto Xpress.

Alteração de IP via Aplicação ou Página Web



Nas configurações da seção de rede da Página web na aba Gerenciamento, 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.

A tabela para configuração de rede é exibida somente quando não possui 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.

A figura a seguir mostra a tabela de interfaces desabilitada com o modo Definido pela aplicação selecionado.

The screenshot shows the Altus NEX web interface. The top navigation bar includes 'Informações Gerais' and 'Gerenciamento'. The left sidebar lists system components: Sistema, Rede, SNMP, Modo de Operação, Dispositivos USB, Firewall, and OpenVPN. The main content area is titled 'Rede' and contains a 'Configuração' section with a 'Modo' dropdown menu set to 'Definido pela aplicação'. Below this, there is a table for 'Sniffer' configuration with two rows: 'Número de Pacotes' (100) and 'Tempo Limite de Ocio (segundos)' (10). Buttons for 'Aplicar', 'Baixar', and 'Executar' are visible.

Figura 5-12. Tabela de Interfaces - Modo Aplicação

Alteração de IP via Aplicação

Uma das funcionalidades disponíveis no Nexto Xpress é a possibilidade de alterar o IP, a máscara de rede e o gateway do controlador, em tempo de execução, por meio de elementos de software disponibilizados pela biblioteca SysSocket do CODESYS, no MasterTool IEC XE. Para tanto, é necessário inserir uma biblioteca do MasterTool chamada SysSocket_Implementation para utilizar as funções SysSockSetIpAddress, SysSockSetSubnetMask e SysSockSetDefaultGateway.

As figuras a seguir ilustram o passo-a-passo para a inserção da biblioteca SysSocket na aplicação.

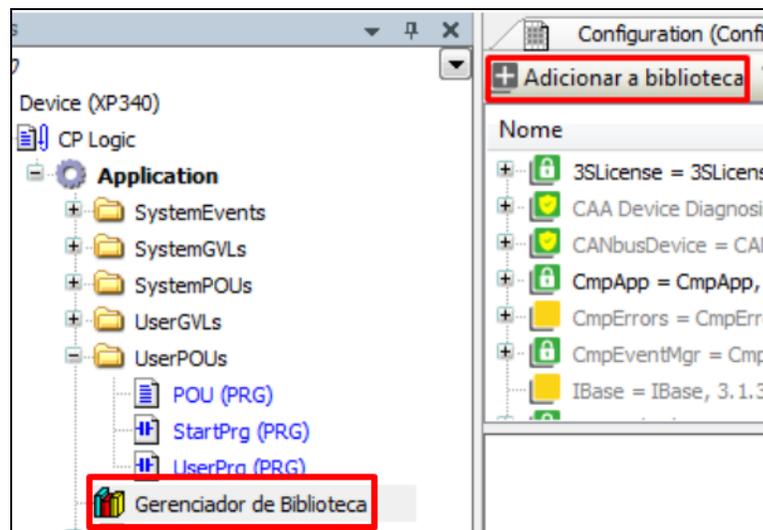


Figura 5-13. Seleção da opção de inserção da biblioteca no gerenciador

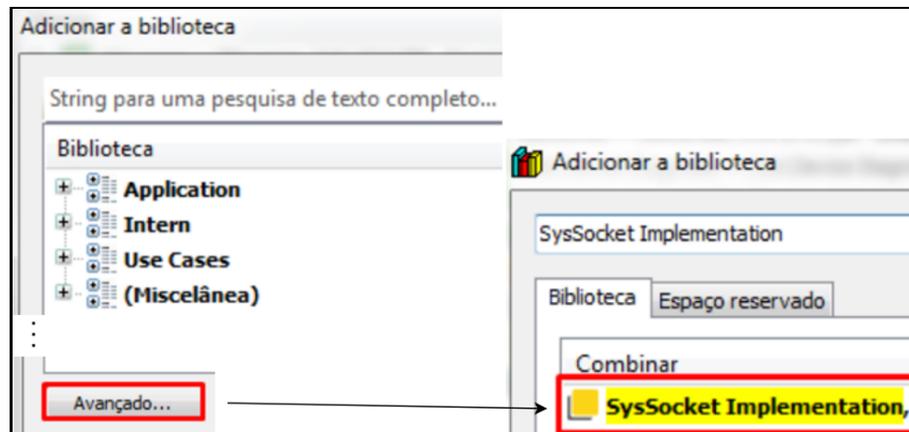


Figura 5-14. Seleção da opção "Avançado" no gerenciador

Além disso, será necessária a biblioteca SysTypes2 Interfaces, portanto, deve-se repetir os passos anteriores com esta outra biblioteca.

Declarando as Funções da Biblioteca SysSocket

Serão utilizadas três funções da biblioteca SysSocket: SysSockSetIpAddress, SysSockSetSubnetMask e SysSockSetDefaultgateway. O código a seguir mostra um exemplo de declaração de variáveis para utilização da funcionalidade de alteração do IP.

```

PROGRAM POU
VAR
    bChangeIp : BOOL;
    szInterface : STRING := 'eth0'; //representa a NET1 dos Xpress
    szInterfaceGateway : WSTRING := "eth0";
    szNewIp: STRING := '192.168.14.22'; //novo IP
    szNewMask: STRING := '255.255.248.0'; //nova mascara de rede
    NewGatewayAddr : SysSocket_Implementation.SysSocket_Interfaces.INADDR;

    Result: RTS_IEC_RESULT;
    Result2: RTS_IEC_RESULT;
    Result3: RTS_IEC_RESULT;

END_VAR

```

Na sequência, está exemplificado – em linguagem ST – o código que implementa esta funcionalidade em uma CPU XPxxx.

```

IF bChangeIp THEN //se changeIP for 1
    Result := SysSockSetIpAddress(szInterface, szNewIp); //troca o IP pelo novo IP
    Result := 0; //coloca o resultado em 0 (zero)

    IF Result = 0 THEN //se o resultado for 0 (zero)
        Result2 := SysSockSetSubnetMask(szNewIp, szNewMask); //coloca a nova mascara de rede
        Result2 := 0; //coloca o resultado 2 for 0 (zero)

        IF Result2 = 0 THEN //se o resultado 2 for 0 (zero)

            NewGatewayAddr.S_un_b.s_b1:=192;
            NewGatewayAddr.S_un_b.s_b2:=168;
            NewGatewayAddr.S_un_b.s_b3:=014;
            NewGatewayAddr.S_un_b.s_b4:=015;

            Result3 := SysSockSetDefaultGateway (szInterfaceGateway, NewGatewayAddr);
            // adiciona toda a nova gateway que foi colocada aqui

            Result3 := 0; //coloca o resultado 3 em 0 (zero)
        END_IF
    END_IF
    bChangeIp := FALSE; //coloca o change IP em 0 (zero)
END_IF

```

Neste exemplo, quando bChangeIp for para TRUE, a interface 'eth0' (que representa NET1 do Xpress) irá receber as configurações de IP, máscara e gateway. As funções são chamadas em sequência, condicionado aos seus retornos (Result, Result2 e Result3) pois, quando em zero, indicam que as funções executaram com sucesso.

A string de identificação da interface Ethernet é específica para cada modelo de controlador. No caso do XP3xxx é "eth0" (NET1).

É importante considerar que, ao confirmar a nova configuração via aplicação, haverá desconexão com o MasterTool (time out). Para logar novamente no controlador, é necessário realizar outro scan no Gateway e selecionar o dispositivo. Não é necessário alterar a configuração da NET1 no projeto.

Caso seja alterado apenas o endereço gateway não haverá desconexão.

Também deve-se ter em mente que esta alteração de configuração não é retentiva, isto é, ao desligar e ligar o controlador, ele retornará com o IP configurado na NET1 do projeto.

Alteração de IP via Página Web

Para o modo Definido pela Página Web a tabela de interfaces permanece habilitada, conforme mostra a figura a seguir. Neste modo, o usuário pode configurar Endereço IP, Máscara de Rede e Gateway de cada uma das interfaces Ethernet disponíveis.

The screenshot shows the 'altus NEX' web interface. On the left is a navigation menu with options: Sistema, Rede, SNMP, Modo de Operação, Dispositivos USB, Firewall, and OpenVPN. The main content area is titled 'Rede' and contains two sections: 'Configuração' and 'Sniffer'.

Configuração

Modo:

NET1	
Endereço IP	<input type="text" value="192.168.18.120"/>
Máscara de Rede	<input type="text" value="255.255.248.0"/>
Gateway	<input type="text" value="192.168.16.253"/>

Sniffer

Número de Pacotes	<input type="text" value="100"/>
Tempo Limite de Ocio (segundos)	<input type="text" value="10"/>

Figura 5-15. Tabela de Interfaces - Modo Web

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.

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 página web serão carregados. O modo de operação Definido pela Página Web configura as interfaces para operarem em Modo Simplex.

Nesta mesma página de configuração, o sniffer de rede pode ser utilizado para observar o tráfego nas interfaces físicas, exceto para os dispositivos USB como modems e adaptadores wifi. Ele possui duas configurações básicas:

1. Número de Pacotes - É o número de pacotes que se deseja capturar. O valor configurado desse parâmetro deve estar dentro da faixa de 1 a 25000 pacotes;
2. 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.

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, mostrado na figura anterior, se tornará o botão Parar.

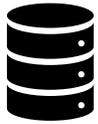
O botão Parar pode ser utilizado para encerrar a execução do sniffer a qualquer momento após ter sido iniciada.

Para cada uma das interfaces em que o Sniffer é executado, ele gera um arquivo .pcap. Estes arquivos são nomeados de acordo com o nome do controlador e a interface que foi analisada. Estes arquivos são encontrados dentro de um .zip, também nomeado de acordo com o nome do controlador.

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 acessados por meio 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, cancelamento manual.

Conexão do CP com Banco de Dados SQL



A biblioteca LibSQLClient permite que seja estabelecida uma conexão direta entre os CLPs da série Nexto e Nexto Xpress com o servidor de um banco de dados Microsoft SQL Server. Após estabelecer essa conexão, é possível realizar operações de escrita e leitura de informações em base de dados SQL Server, utilizando a linguagem de consulta estruturada SQL.

A figura a seguir mostra a LibSQLClient e o bloco funcional MSSQL_CLIENT disponível nessa biblioteca.

Nome	Contexto	Versão Efetiva
LibDatabaseManager = LibDatabaseManager, 1.1.0.3 (Manufacturer)	LibDatabaseManager	1.1.0.3
LibDatalogger = LibDatalogger, 1.0.0.5 (Manufacturer)	LibDatalogger	1.0.0.5
LibDataTypes = LibDataTypes, 1.0.0.0 (Manufacturer)	LibDataTypes	1.0.0.0
LibIntegratedIo = LibIntegratedIo, 1.0.0.14 (Manufacturer)	LibIntegratedIo	1.0.0.14
LibNextoNet = LibNextoNet, 1.3.0.4(WAA)	LibNextoNet	1.3.0.4
LibSQLClient = LibSQLClient, 1.0.0.2 (Manufacturer)	LibSQLClient	1.0.0.2
NextoStandard = NextoStandard, 1.1.0.26 (WAA)	NextoStandard	1.1.0.26
Standard, 3.5.12.0 (System)	Standard	3.5.12.0
XP315 Diagnostic Structs = XP315 Diagnostic Structs, 1.0.0.7 (Manufacturer)	XP315_Diagnostic_Structs	1.0.0.7



Figura 5-16. Biblioteca LibSQLClient e bloco funcional MSSQL_CLIENT

Parâmetros de Entrada do Bloco Funcional

- EXECUTE - Ativa o bloco funcional na borda de subida
- DB_LOGIN_PARAMS - Parâmetros para realizar login no banco de dados, incluído o endereço ou nome do servidor, porta TCP, timeout, nome e senha de usuário e nome do banco de dados;
- SQL_COMMAND - String que abriga o comando SQL a ser enviado para o banco de dados;
- QUERY_DATA - Variável utilizada para armazenar os dados retornados pelo banco de dados após a execução de um comando SELECT. Nesse caso, o usuário deve instanciar uma variável do tipo C_QUERY_DATA e colocá-la na entrada do bloco QUERY_DATA para o mesmo não retornar erro de parâmetro inválido (MSSQL_INVALID_PARAMETERS). Essa estrutura contempla o número de colunas e linhas recebidas do banco de dados, tipos de dados e nomes das colunas, bem como os valores dos dados das células da tabela.

Parâmetros de Saída do Bloco Funcional

- DONE - Indica se a operação foi completada com êxito;
- BUSY - Indica se há uma operação ocorrendo (ocupado);

- ERROR - Indica se há um erro na execução do comando;
- STATUS - String que indica o status de funcionamento do bloco. A documentação do produto pode ser consultada para acesso aos códigos de status.

Também é possível redimensionar a tabela dos dados, ajustando o número máximo de linhas e colunas que a estrutura QUERY_DATA irá armazenar. Para fazer isso, o usuário deve alterar o valor das constantes globais gc_dwMSSQLMaxRows e gc_dwMSSQLMaxColumns, descritas a seguir.

Nome	Tipo	Valor (editável)	Comentário
P  gc_dwMSSQLMaxRows	UINT	15	Maximum rows of table array
P  gc_dwMSSQLMaxColumns	UINT	10	Maximum columns of table array
P  gc_dwMSSQLStringLength	UINT	200	String length of data values
P  gc_dwMSSQLStrLengthColumnName	UINT	40	String length of column names

Além disso, o usuário também tem a opção de alterar o tamanho máximo das strings que armazenam os nomes das colunas, assim como as strings que armazenam os dados retornados pelo servidor. Por padrão, as constantes estão configuradas para armazenar 15 linhas, 10 colunas, strings de 200 caracteres dentro do array de dados (asTableQuery) e 40 caracteres no array com os nomes das colunas.

O código a seguir exemplifica uma operação de leitura no banco de dados através do Nexto utilizando a linguagem ST. É importante notar alguns pontos importantes do código abaixo:

- Na operação de leitura é importante destacar a adição da variável QD do tipo QUERY_DATA para armazenar os dados lidos pela query.
- Após configurar todos os parâmetros e compilar a aplicação, a operação no banco se dará na borda de subida da variável "xExec", e a variável "xError" permanecerá em FALSE. Caso ela venha a se tornar TRUE, algum erro ocorreu na operação, que pode ser visualizado na variável "msgError".

```
PROGRAM Database
VAR
FB_QUERY      : MSSQL_CLIENT;
  insertInstruction: STRING(255) := `select * from dbo.myTeste`;
  QD:C_QUERY_DATA;

  loginParams: DB_LOGIN_PARAMETERS:= (sServer:=`192.168.1.100`,
    uiTcpPort:=1433,
    sUser:=`sa`,
    sPassword:=`senha`,
    sDataBaseName:=`myDatabase`,
    uiTimeout:=1);

  xExec: BOOL;
  xBusy: BOOL;
  xError: BOOL;
  msgError: BOOL;
  xDone: BOOL;

END_VAR
```

```

FB_QUERY(EXECUTE:=xExec,DB_LOGIN_PARAMS:=loginParams,SQL_COMMAND:=insertInstruction, QUERY_DATA:=QD);
xBusy:=FB_QUERY.BUSY;
xError:=FB_QUERY.ERROR;
xDone:=FB_QUERY.DONE;

```

Escrevendo em um banco de dados SQL

Nesta seção, iremos escrever em um banco de dados criado no Microsoft SQL, utilizando o comando `insert`. Primeiramente, abra o Microsoft SQL Server e na opção `Autenticação`, escolha `Autenticação do SQL Server`. Preencha seu login e senha e clique em `Conectar`.

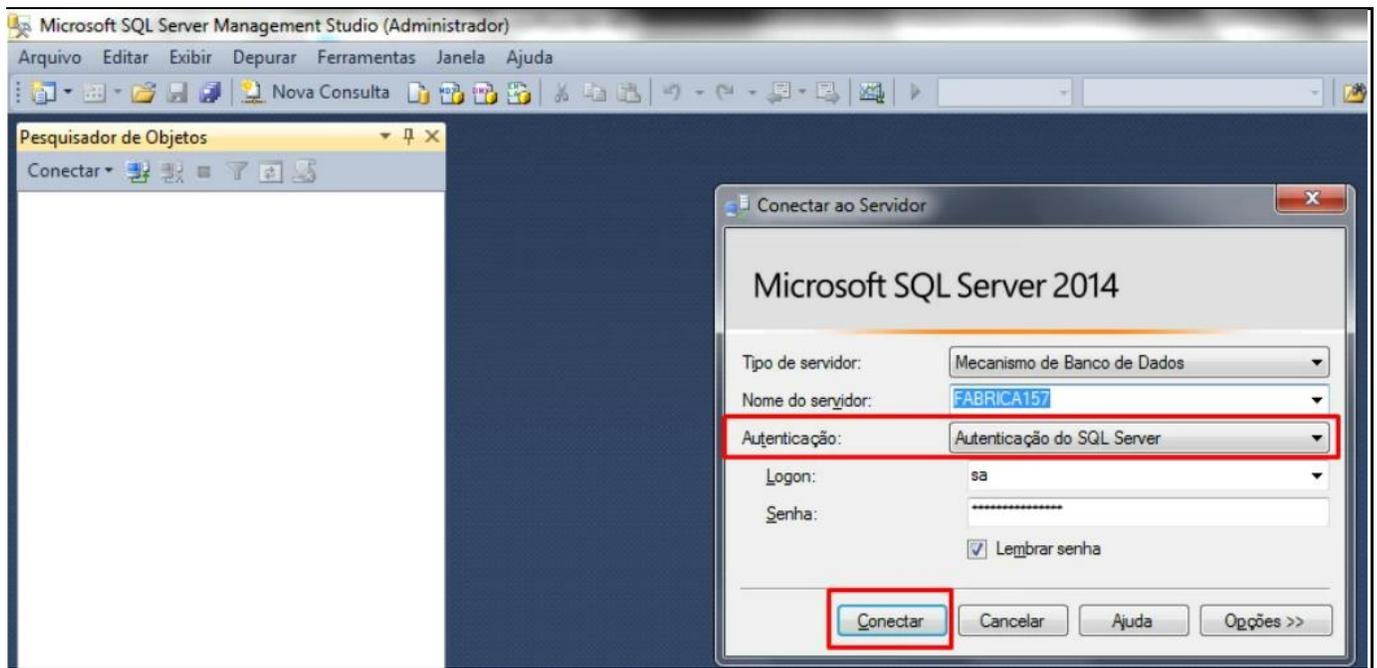


Figura 5-17. Conexão com o servidor do banco de dados

Após conectar-se ao servidor, clique em `Nova Consulta` para criar a tabela que será escrita através dos comandos do MasterTool IEC XE.

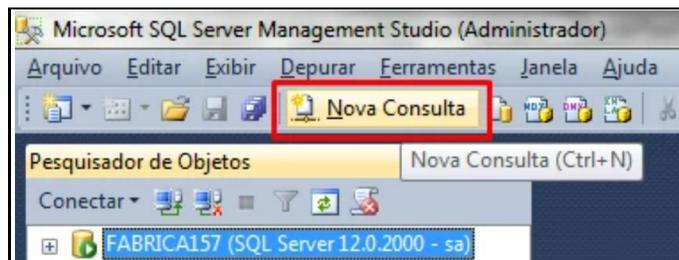


Figura 5-18. Seleção de nova consulta ao banco de dados

Na aba que será aberta, escreva o programa da imagem abaixo e clique em `Executar` para que a tabela seja criada.

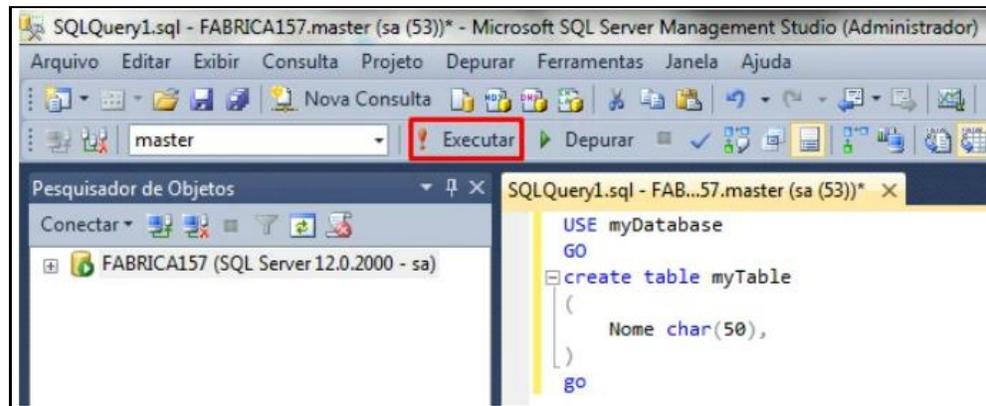


Figura 5-19. Criação da tabela do banco de dados

Com a tabela devidamente criada, abra um novo projeto no MasterTool IEC XE 3.20, inclua a biblioteca `LibSQLClient` conforme visto no início deste tutorial, crie uma nova POU em texto estruturado e então escreva o programa a seguir nesta POU criada.

```
PROGRAM Database
```

```
VAR
```

```

FB_QUERY : MSSQL_CLIENT;
insertInstruction1: STRING[255] := 'insert into dbo.myTable values ($'Altus$)';
QD : C_QUERY_DATA;
loginParams6 : DB_LOGIN_PARAMETERS := (sServer := '192.168.14.102',
                                       uiTopPort := 1433,
                                       sUser := 'sa',
                                       sPassword := '██████████',
                                       sDataBaseName := 'myDatabase',
                                       uiTimeout := 1);

xExec : BOOL := 0;
xBusy : BOOL;
xError : BOOL;
msgError : SQL_STATUS_CODES;
xDone : BOOL;

```

```

FB_QUERY(EXECUTE:=xExec, DB_LOGIN_PARAMS:=loginParams6, SQL_COMMAND:=insertInstruction1, QUERY_DATA:=QD);
xBusy := FB_QUERY.BUSY;
xError := FB_QUERY.ERROR;
xDone := FB_QUERY.DONE;
msgError := FB_QUERY.STATUS;

```

Na variável `sServer` (sublinhado em vermelho) deve ser colocado o IP do computador onde o banco de dados Microsoft SQL está operando.

Passa a variável `xExec` para TRUE (através da opção de forçar valores ou vinculado-a à uma chave física em uma entrada digital), pois ela irá ativar a variável `EXECUTE`, enviando o comando `insert` para o banco SQL.

Por fim, abra uma `Nova Consulta` no Microsoft SQL Server e escreva o programa a seguir.

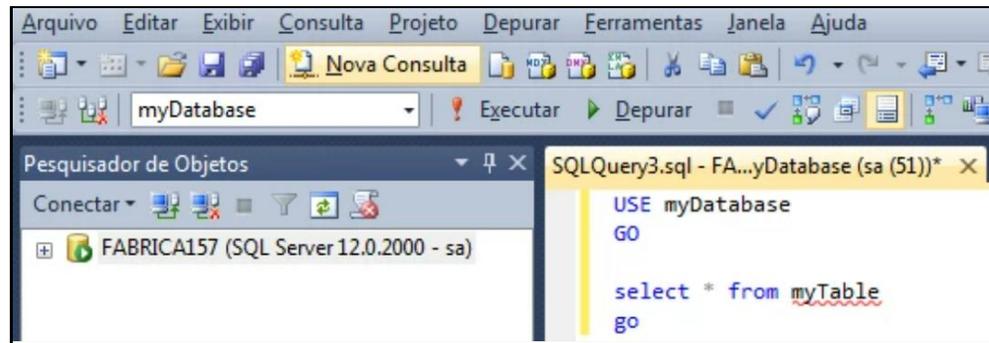


Figura 5-20. Código para a nova consulta

O comando `select` irá exibir a tabela atualizada. Para isso, após escrever o programa acima, clique em `Executar`.

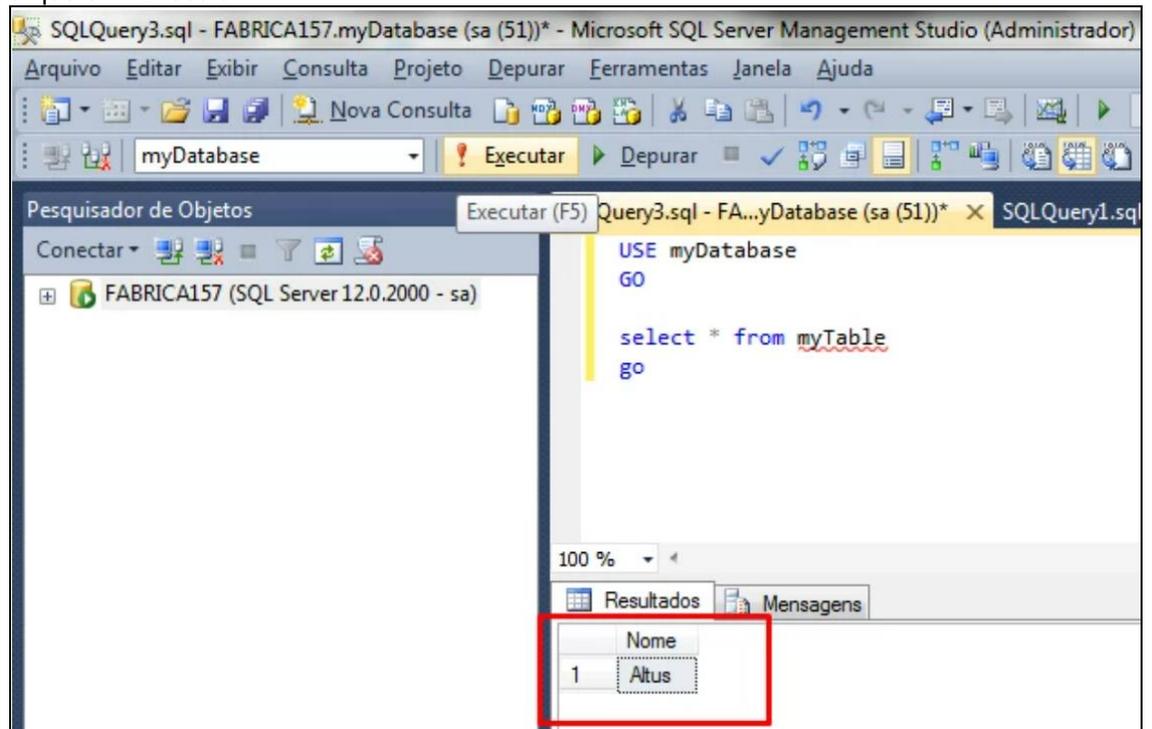


Figura 5-21. Nnn

O valor enviado pelo MasterTool foi escrito corretamente na tabela do banco de dados.

DICA: consulte o instrutor para orientações adicionais referentes à instalação do banco de dados MSSQL na sua máquina.

Visualização



A *Visualização* totalmente integrada é a solução ideal para simular, operar ou monitorar máquinas ou plantas. Um conceito de visualização modular oferece um cliente de visualização que pode, de forma flexível e com pouco esforço, ser utilizado para as mais diversas aplicações personalizadas. O editor da visualização oferece elementos de visualização completos, prontos, suportados por uma série de comandos. Use esses elementos para criar modernas máscaras de visualização com apenas alguns cliques. Os elementos podem ser organizados em bibliotecas.

Visão Geral

Ferramentas e Editores

Uma visualização baseada em um conceito de visualização modular fornece as seguintes ferramentas/editores:

- Comandos de Visualização: aqui é possível criar uma visualização/máscara/painel na ferramenta IEC 61131-3 com apenas alguns cliques.
- Elementos de Visualização: elementos pré-fabricados estão disponíveis para utilização, numa grande quantidade.
- Perfil de Visualização: permite definir um perfil de bibliotecas de visualização. Cada projeto de visualização contendo pelo menos um objeto de visualização deve se basear em um.
- Bibliotecas de Visualização: aqui, os elementos podem ser resumidos para os tornar adequados para o uso em outros projetos.
- Configurações do Projeto: configurações para projetos de visualização.
- Gerenciamento de Visualização com Clientes: aqui, os clientes de visualização são gerenciados e configurados.

A visualização com MasterTool permite que você:

- Localize visualizações por referência e/ou alterne entre diferentes visualizações: Frame;
- Use interfaces para transferência de parâmetro para instanciar visualizações complexas: Visão Geral do Editor de Interface;
- Obtenha o suporte a multilinguagem por um editor de lista de texto integrado: Texto e Linguagem na Visualização;
- Instale um sistema de gerenciamento de usuários: Gerenciamento do Usuário.

Mecanismo Geral

Os clientes de visualizações remotas nada mais são que intérpretes de comandos do desenho. Cada cliente terá as mesmas instruções, de modo que as visualizações resultantes serão todas idênticas.



Figura 5-22. Mecanismo Geral

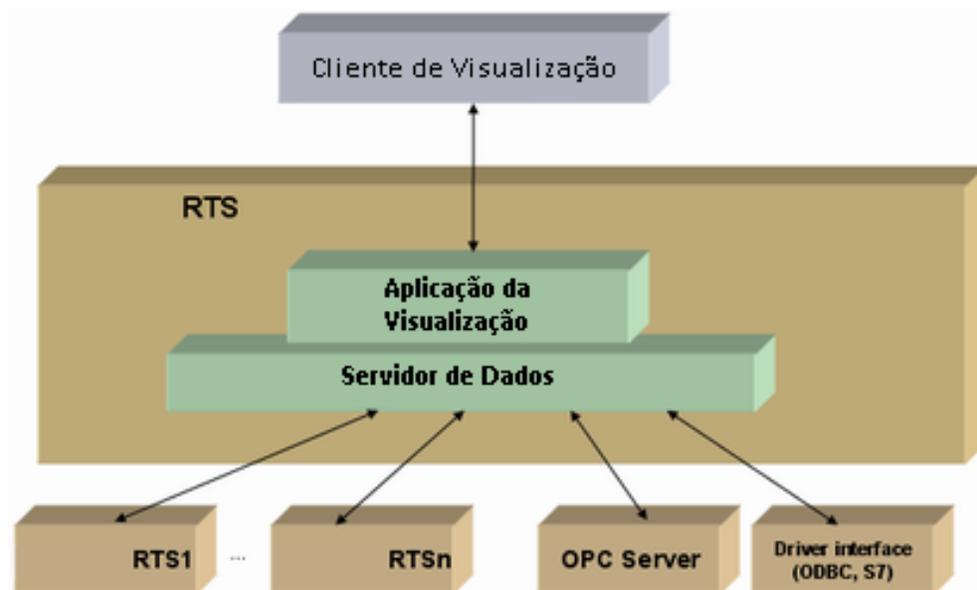


Figura 5-23. Servidor de Dados para fornecer dados de visualização

Uma aplicação de visualização é criada em código IEC no editor de visualização do sistema de programação. No caso de a visualização estar executando no dispositivo, o código desta será gerado e carregado lá.

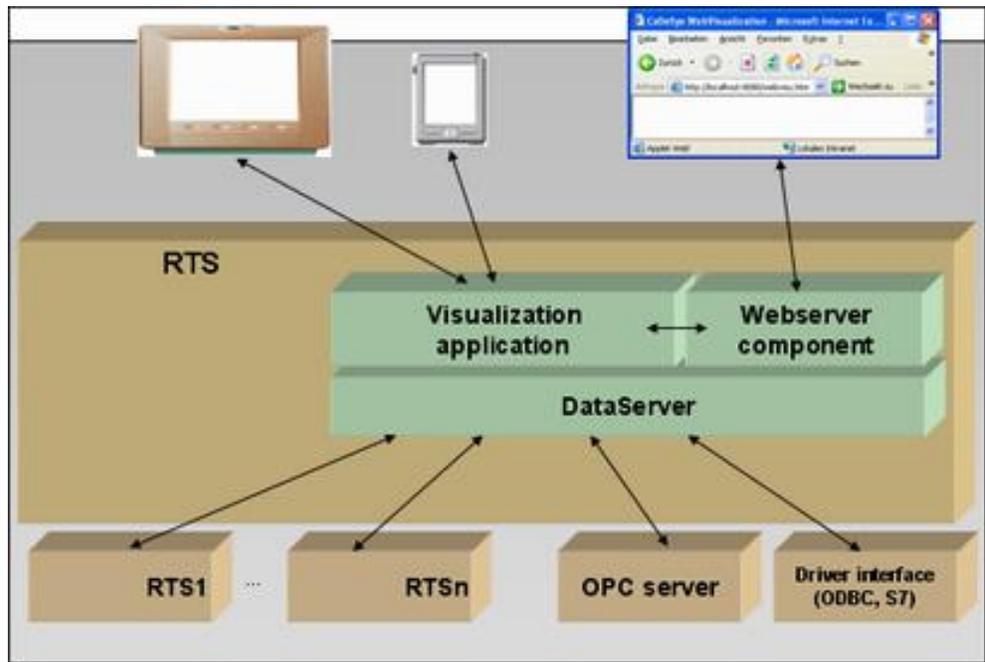


Figura 5-24. Cliente de visualização WebVisu

O WebVisu é para o uso de um servidor web para se conectar à aplicação relacionada.

Fontes de dados externas (remotas) podem ser usadas em visualizações. Para este efeito, um Servidor de Dados deve estar disponível para a aplicação local lidar com as respectivas fontes de dados.

ATENÇÃO:

O WebVisu e o download da aplicação são dois processos que gastam uma grande quantidade de memória, quando os dois ocorrem juntos, é normal que a memória livre diminua significativamente. Em alguns casos, terá uma mensagem no log: Memória livre alcançou estado crítico.

Orientamos fazer um Reset Origem ou uma inicialização sem aplicação (botão pressionado), antes de fazer um download quando o problema acontece.

Criar uma Visualização

Um objeto de visualização pode ser inserido à aplicação atribuindo na árvore de dispositivos abaixo de um aplicativo ou na janela de visualização das POUs, usando o comando Acrescentar objeto no menu Projeto (também no menu de contexto). A janela de um novo objeto de visualização é intitulada com <nome do objeto visu> e mostra uma área vazia primeiro.

Uma Visualização Inicial, que é o objeto de visualização, que deve ser aberto primeiro, após o login no CP com uma aplicação, deve ser inserido na Árvore de Dispositivos abaixo do respectivo objeto de aplicação.

Quando o primeiro objeto de visualização é adicionado no projeto, as respectivas Bibliotecas de Visualização serão adicionadas automaticamente no Editor Library Manager.

Visualização, Diálogo ou Teclado Numérico/Keypad

Cada visualização particular pode ter Propriedades de um Objeto de Visualização como a sua utilização designada: Visualização, Teclado Numérico/Keypad, Diálogo ou o tamanho de exibição. Note que, neste contexto, a visualização pode ser criada e configurada explicitamente para ser utilizada como diálogo de entrada do usuário (Configuração de Entrada) em outras visualizações. Implicitamente também um teclado numérico padrão e um teclado padrão estão disponíveis para esta finalidade. A utilização de tais teclados e diálogo pode ser definido na propriedade de um Elemento de Visualização.

**NOTAS:**

- **Ficar online:** a visualização funciona com a ajuda de um sistema de tempo de execução integrado. Assim, durante ações de edição primárias, você receberá mensagens sobre como iniciar e fazer download.
- **Modo de Simulação:** o Web Visu não está disponível no navegador em Modo de Simulação, porém permanece disponível no software.

Comandos de Visualização

Os comandos visuais são fornecidos com o plug-in Visual Editor para a categoria Comandos Visuais. Eles servem para editar um objeto de visualização no editor de visualização. A maioria deles são parte do Menu Visualization e, geralmente, também do menu de contexto no editor de visualização. Seguem os comandos Visuais que não são listados no Menu Visualization:

- Adicionar Elemento Visual: este comando (categoria *Comandos Visuais*) é usado para inserir um elemento na visualização atual e, portanto, corresponde a inserção através de arrastar-e-soltar a partir da *ToolBox* (Caixa de Ferramentas).
- Seleção de Frame: este comando (categoria *Comandos Visuais*) é usado para configurar o conteúdo de um *Frame*. Um elemento de frame é utilizado para definir uma subárea de uma visualização que inclui um ou vários outros efeitos visuais. No modo on-line que pode ser alternada entre a exibição dessas visualizações particulares. Basicamente, a primeira visualização na lista de visualizações atribuídas ao frame é exibida. Mas por uma entrada em outro - devidamente configurado (*Switch Framevisualization*) - elemento de visualização que o usuário pode efetuar que uma das outras visualizações atribuídas serão exibidas no frame. Assim alternar entre diferentes telas dentro de uma visualização é possível.

As visualizações incluídas em um *Frame* são referências (instâncias das visualizações originais). Espaços reservados definidos nos objetos de visualização originais podem ser substituídos por valores localmente aplicáveis.

A seleção das visualizações em um frame é feita na caixa de diálogo Configuração de Entrada, que é aberto por este comando.

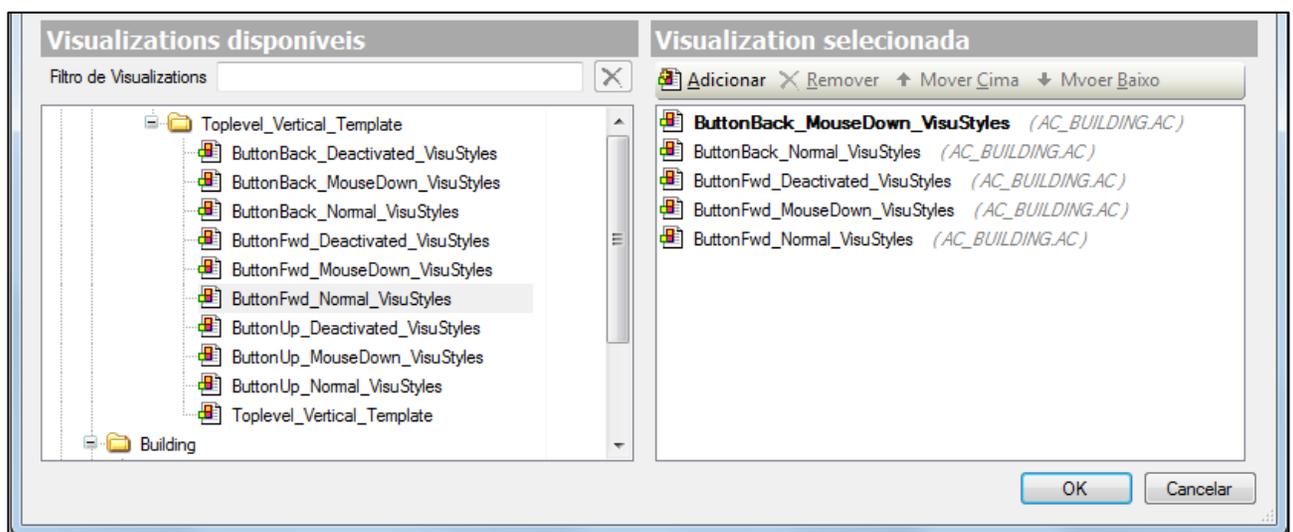


Figura 5-25. Caixa de diálogo de Configuração de Frame

No lado esquerdo as *Visualizações Disponíveis* do projeto estão listadas. Selecione as que devem ser incluídas no frame e coloque elas na lista *Visualizações selecionadas* no lado direito com um duplo clique ou com o botão *Adicionar*. Visualizações selecionadas podem ser removidas com um duplo clique ou com o botão *Deletar*. Selecionar vários dos elementos também é possível. A ordem dentro da lista pode ser alterada pelo comando *Mover para cima* e *Mover para baixo*.

Recomenda-se atribuir apenas as visualizações para um frame que são geridas no conjunto global. Caso contrário, podem ocorrer problemas, se em um momento posterior qualquer objeto do tipo dispositivo ou aplicação for renomeado e devido a isso o caminho das visualizações atribuídas não serão mais válidos.

A ordem das visualizações selecionadas de cima para baixo determina os números de índice implícitos gerados automaticamente para as visualizações. O mais superior fica no índice 0, os seguintes 1, 2, etc. Os números de índice são necessários para a configuração da troca da função *Chavear Framevisualization* de outro elemento. Inicialmente a visualização com índice 0 será exibido.

Exemplo:

1. Através de uma barra de menu que pretende determinar qual das várias visualizações atualmente devem ser exibidas no modo online.
2. Adicione um elemento de frame.
3. Atribua este frame através de *Seleção de Frame* na caixa de diálogo *Configuração de Frame* as visualizações para serem alternadas.
4. Adicione elementos de controle para cada visualização que permite mudar para essa visualização. Por exemplo adicionar uma barra de menu com elementos de controle.
5. Cada elementos de controle é configurado definindo sua Configuração de Entrada: Adicionar na propriedade *OnClick* após a ação de *Chavear Framevisualization* e atribuir a visualização em particular à estrutura respectiva.

Editor de Visualização

O editor de visualização está disponível através do plug-in do *Editor de Visualização*.

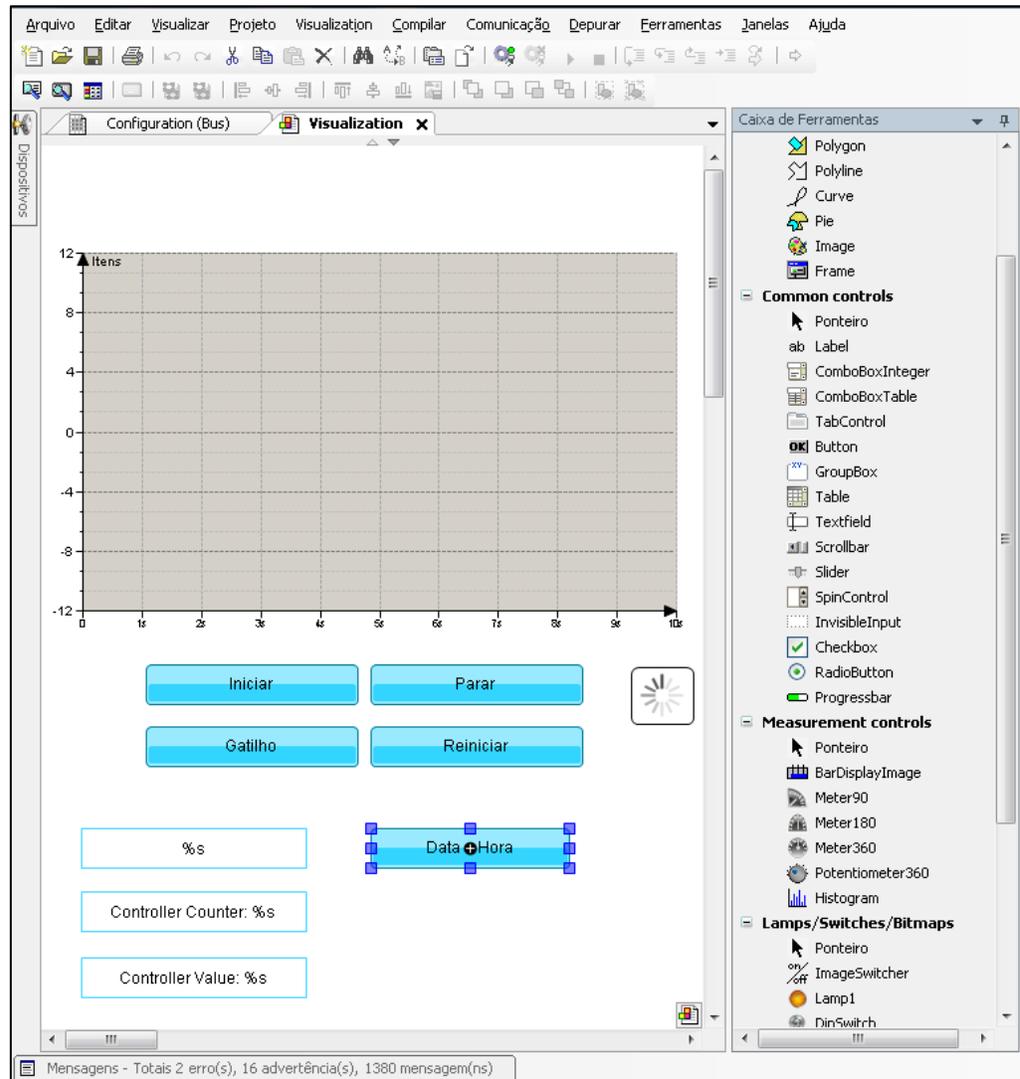


Figura 5-26. Editor de Visualização

A visualização pode ser criada com a ajuda dos seguintes editores e ferramentas adicionais:

A janela do editor é intitulada  <nome de visualização>[<caminho do objeto>] e exibe a visualização pintada/editada/programada.

A Caixa de Ferramentas, que contém todos os elementos de visualização atualmente disponíveis.

A janela Propriedades contendo as propriedades do elemento de visualização, que está atualmente selecionado na janela do editor.

Os comandos fornecidos pelo menu Visualização ou pelo menu de contexto no editor.

O Editor de Interface para definir espaços reservados. Se a visualização se destina a ser inserida em outra visualização.

A Configuração Hotkeys para atribuir ações para teclas ou combinações de teclas. Em conta, contudo, que o dispositivo usando a visualização deve suportar as respectivas chaves.

A Lista de Elementos fornecendo uma visão geral sobre todos os elementos da visualização atual e permitindo a seleção, exclusão e alteração da posição do elemento de trás para frente e vice-versa.

Se o *Editor de Interface*, *Configuração Hotkeys*, ou *Lista de Elementos* é ativada através do respectivo comando (por padrão no menu *Visualização*), o Editor de Visualização será bipartido

e fornecerá as respectivas abas na parte superior. Os elementos de visualização podem ser animados pela utilização direta das variáveis de projeto, ou sob a forma de expressão, que é combinado com os operadores e constantes. Por exemplo, este permite dimensionar as variáveis para o uso na visualização. Dentro de expressões arbitrárias da visualização, até mesmo chamadas de função são válidas.

Caixa de Ferramentas

A Caixa de Ferramentas, que é utilizada em conjunto com o Editor de Visualização, fornece elementos de visualização para serem inseridos na janela de edição. Os elementos são fornecidos via Bibliotecas de Visualização e a seleção atual, no caso de visualizações associadas a dispositivos, depende da descrição do dispositivo. A Caixa de Ferramentas pode ser aberta pelo Menu Visualizar. Ela consiste nas seguintes categorias:

- Basic;
- Common Controls;
- Measurement Controls;
- Lamps, Switches, Bitmaps;
- Special Controls.

Lá, os elementos de visualização associados estão listados com nomes e ícones. Os elementos podem ser inseridos arrastando-os até a janela de edição atualmente aberta. Um sinal de adição no cursor ao arrastar indica que o elemento será adicionado à janela de edição. Após soltar o cursor, o elemento será exibido na visualização. Para inserir outras visualizações no elemento de frame da visualização atual, use o comando Seleção de Frame.

Propriedades do Elemento

Símbolo: 

Ao selecionar um elemento de visualização as suas propriedades passam a ser apresentadas nesta aba.

Objeto de Visualização: Propriedades de um Objeto de Visualização

O diálogo Propriedades abre com um clique no comando Propriedades do Elemento do Menu Visualizar quando um objeto de visualização é selecionado, ou no menu de contexto de um objeto de visualização selecionado.

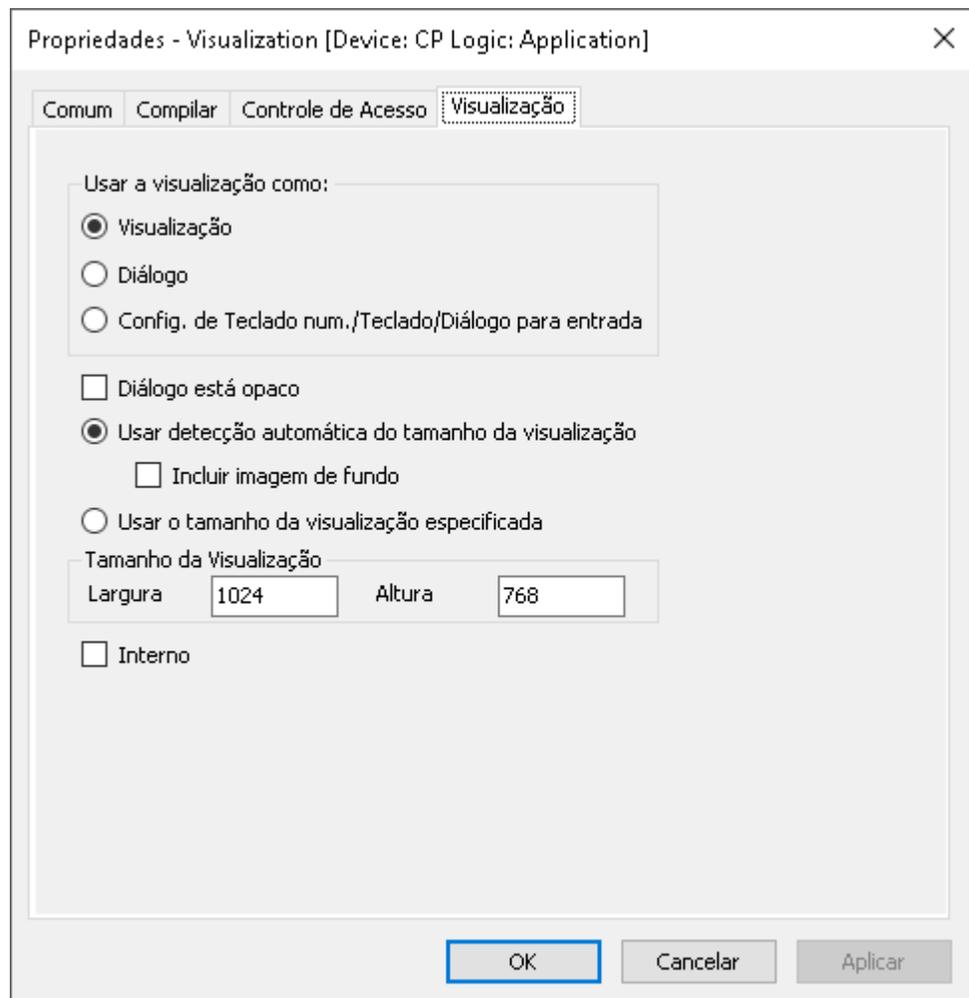


Figura 5-27. Diálogo 'Propriedades', Categoria 'Visualização'

Elementos de Visualização: O que Pode ser Feito com um Elemento de Visualização?

As seguintes opções estão disponíveis para um elemento de visualização:

- Entradas do Usuário ao Editar um Elemento;
- Texto e Linguagem na Visualização;
- Configuração de Entrada;
- Direitos de Acesso.

Elementos de Visualização: Propriedades

As seguintes propriedades estão disponíveis para um elemento de visualização:

- Gerais;
- Position;
- Center;
- Colors;
- Element Look;

- Texts;
- Absolut Movement;
- Relative Movement;
- Text Variables;
- Dynamic Texts;
- Font Variables;
- Color Variables;
- Look Variables;
- State Variables;
- Access Rights;
- Configurações de Entrada.

Elementos de Visualização: Basic

O conjunto de elementos Basic é composto pelos tipos Rectangle, Round Rectangle, Ellipse, Line, Polygon, Polyline, Bézier Curve, Pie, Image e Frame.

Elementos de Visualização: Common Controls

O conjunto de elementos Common Controls é composto pelos tipos Label, Combo Box Integer, Combo Box Array, Tab Control, Button, Group Box, Table, Text Field, Scrollbar, Slider, Spin Control, Invisible Input, Progress Bar, Radio Button e Checkbox.

As configurações e informações sobre Entradas do Usuário ao Editar um Elemento e Interface do Usuário Durante a Edição de Propriedades são as mesmas dos objetos do grupo Basic.

Elementos de Visualização: Measurement Controls

O conjunto de elementos Measurement Controls é composto pelos tipos Bar Display, Meter 90°, Meter 180°, Meter 360°, Potentiometer e Histogram.

As configurações e informações sobre Entradas do Usuário ao Editar um Elemento e Interface do Usuário Durante a Edição de Propriedades são as mesmas dos objetos do grupo Basic.

Elementos de Visualização: Lamps, Switches, Bitmaps

O conjunto de elementos Lamps, Switches, Bitmaps é composto pelos tipos Image Switcher, Lamp, Dip Switch, Power Switch, Push Switch, Push Switch LED, Rocker Switch e Rotary Switch.

As configurações e informações sobre Entradas do Usuário ao Editar um Elemento e Interface do Usuário Durante a Edição de Propriedades são as mesmas dos objetos do grupo Basic.

Elementos de Visualização: Special Controls

O conjunto de elementos Special Controls é composto pelos tipos Trace, Símbolo de Espera e Editor de Texto.

As configurações e informações sobre Entradas do Usuário ao Editar um Elemento e Interface do Usuário Durante a Edição de Propriedades são as mesmas dos objetos do grupo Basic.

Bibliotecas de Visualização

Devido ao fato de que os elementos de visualização são implementados como blocos funcionais, que são fornecidos através de bibliotecas, ao adicionar um objeto de visualização em um projeto, um certo conjunto de bibliotecas de visualização será adicionado ao Editor Library Manager na Janela de POUs.

Uma biblioteca de visualização sempre é configurada como um tipo especial de biblioteca de espaço reservado, para os efeitos que não forem resolvidos até que a versão exata da biblioteca a ser usada é incluída no projeto. Só então o perfil ativo no momento vai definir qual versão, na verdade, é necessária. Note-se que este tipo de bibliotecas é diferente a partir das bibliotecas de espaço reservado específicas do dispositivo onde os espaços reservados são resolvidos pela descrição do dispositivo.

Bibliotecas básicas são incluídas por padrão quando é adicionado um objeto de visualização no projeto.

Gerenciamento de Visualização com Clientes

Se uma visualização é inserida na *Árvore de Dispositivos* abaixo de uma aplicação, automaticamente um objeto de *Gerenciador de Visualização*  será adicionado. Ele lida com as definições comuns para todos os efeitos visuais atribuídos à aplicação atual por clientes. Para abrir o objeto Gerenciamento de Visualização dê um duplo clique sobre a entrada na árvore de dispositivos ou selecione a entrada e use o comando Editar Objeto do menu de contexto.

Visualização em Modo Online

Para operar uma visualização no modo online pelo teclado, há algumas teclas padrão suportadas por cada dispositivo. Adicionalmente, podem existir teclas específicas para cada dispositivo.

Pesquisa na Documentação: visualização

Consulte, na documentação do produto, as configurações detalhadas associadas à funcionalidade de visualização.

Fonte: Manual de Utilização MasterTool IEC XE MT8500 – MU299048



Estudo Dirigido 5-2: criação de uma visualização web

A visualização web é uma funcionalidade avançada que está disponível na CPU XP340 (WebServer). Este estudo dirigido aborda a inclusão de visualizações, assim como objetos básicos como Labels, Retângulos e Elipses, além de como editá-los para deixar da forma e cor desejada.



Orientações gerais referentes à realização da atividade

Consulte na Base de Conhecimento Altus a série de tutoriais “Criando uma visualização web com o NX3005 ou XP340”. Nessa série de tutoriais você irá aprender a criar uma visualização web, função disponível nas CPUs NX3005 e XP340 da linha Nexto Xpress (WebServer). Na primeira parte iremos aprender a criar um projeto utilizando o MasterTool IEC XE e adicionar as visualizações, assim como objetos básicos como Labels, Retângulos e Elipses, além de como editá-los para deixar da forma e cor que desejar.



DICA: Caso você queira saber mais sobre as diferenças entre Web Server embarcado em CLP, Terminais de Operação (IHM) e software SCADA, assista o Webinar a seguir (<https://www.youtube.com/watch?v=bieRMoJKEZo>).



 **DICA:** consulte o Instrutor para orientações adicionais.

ANOTAÇÕES

6. Aplicações, Avaliação e Encerramento

Automação e controle de processo com o Nexto Xpress

Utilizando os conhecimentos desenvolvidos no treinamento implemente a automação do sistema indicado a seguir.



Automação do Processo de Preparação de Solução de NaCl

Projetar a automação (em conformidade com a IEC61131-3) do processo de preparação de solução de cloreto de sódio em uma indústria eletroquímica utilizado posteriormente na produção de cloro e soda.

Você deve procurar, na medida do possível, atender os seguintes aspectos que norteiam - Segundo Fonseca (2008) - um projeto baseado na norma IEC61131-3:

- Decomposição do problema de controle de cima para baixo (TOP-DOWN);
- Estruturação da aplicação de baixo para cima (BOTTOM-UP);
- Modularização dos elementos de controle e de programação;
- Ampliação/reutilização da biblioteca de POU's (Unidade de Organização de Programa);
- Utilização de linguagens de programação mais adequadas ao problema de controle a ser solucionado e às necessidades dos usuários;
- Priorização da auto documentação do programa do CP, proporcionada pelo ambiente de programação (PSE).



DICA: O conceito de decomposição (TOP-DOWN) e estruturação (BOTTOM-UP) implica na utilização de blocos funcionais e funções padrões através de bibliotecas que podem ser fornecidas pelo fabricante do CP ou definidas pelo usuário.

Descrição do processo

A correia transportadora acionada pelo motor M2 conduz o sal a ser dissolvido para o reservatório misturador e, com a abertura da válvula, é despejado um solvente (água) no reservatório. As duas substâncias são misturadas por meio do agitador acionado pelo motor M1. O esvaziamento do reservatório com a solução efetua-se ao ligar a bomba B2 e ao abrir a válvula Y2. Um sensor de nível mínimo (S0) interrompe o esvaziamento. O processo completo se repete ciclicamente se, após um tempo da última etapa, não ocorrer uma ordem manual de parada.

Detalhamento do processo

1. Deve haver a possibilidade de iniciar o processo manual ou automaticamente.

2. A válvula Y1 se abre dando início ao processo que admite água no tanque até o volume QA fixado previamente e medido por um medidor de vazão na saída da válvula Y1.
3. O volume QA deve ser inferior ao nível máximo (S2) e superior ao nível baixo (S1) do tanque.
4. Ao atingir o nível baixo (S1), o motor do agitador (M1) deve ser ligado.
5. Ao atingir o volume prefixado QA de água, deve-se dar o fechamento da válvula Y1. Se o volume QA não for atingido em um tempo TQA deve ser gerado um alarme.
6. Com a válvula Y1 fechada, é ligado o motor M2 que aciona a correia transportadora a qual leva o sal ao tanque. O peso OS (função de QA) é previamente medido por uma balança de correlação peso/volume controlada na saída de Y3, dando origem ao volume QS.
7. Ao atingir o volume QS, a válvula Y3 deve ser fechada e 30 s depois deve ser desligado o motor M2.
8. Depois do desligamento de M2 deve transcorrer um tempo TM de agitação da mistura.
9. Ao término do tempo TM, um sensor deve indicar se a correlação condutividade e temperatura atingiu um valor M, caso contrário, deve ser gerado um alarme.
10. Atingindo o valor M ou por um comando manual deve-se dar a descarga do tanque, se o tanque de eletrólise estiver vazio (sensor ND não acionado).
11. A descarga se inicia com a abertura da válvula Y2 e com o acionamento da bomba B2. O motor M1 (agitador) deve ser desligado quando o nível do tanque for baixo (S1).
12. O tanque deve ser esvaziado até o fim (S0), então a bomba B2 deve ser desligada e a válvula Y2 deve ser fechada.
13. Se após 30 s do término do processo não for recebida ordem manual, o processo deve ser reiniciado.

Esquematização do Processo

A figura a seguir ilustra o esboço do processo não automatizado elaborado pelo “cliente”. Alguns itens indicados não têm relação direta com a automação proposta e são apenas ilustrativos do processo existente.

Situação de Aprendizagem Complementar 6-1: armazenamento dos dados da aplicação de Preparação de Solução de NaCl (DATALOGGER)

Essa situação de aprendizagem complementa a aplicação relativa à Preparação de Solução de NaCl desenvolvida anteriormente e propõe a implementação de um DATALOGGER para registro das variáveis, eventos e parâmetros do processo, enfatizando o alinhamento do Nexto Xpress com o contexto da Indústria 4.0.

Orientações gerais referentes à realização da atividade

O bloco funcional Datalogger permite ao usuário criar seus próprios registros de dados e gravar em arquivos de log no cartão de memória ou na memória interna presente na CPU. Os arquivos são gerados em um diretório específico do cartão de memória ou na memória interna da CPU no formato CSV, permitindo a visualização em editores de texto e planilhas.

Para tanto, você precisa acrescentar a biblioteca LibDatalogger, o que, na sequência, permitirá adicionar a função Datalogger.

Confira abaixo um exemplo de como declarar e chamar a função em texto estruturado.

```

2  PROGRAM UserPrg
3  VAR
4      FB_Datalogger  :DATALOGGER;
5      Status        :DATALOGGER_STATUS;
6      Records       :UDINT;
7      EnableFB      :BOOL;
8      Error         :BOOL;
9      Data          :REAL;
10 END_VAR

1  Data := 5*SIN(2*3.14159*TIME_TO_REAL(TIME()));
2
3  FB_Datalogger(
4      ENABLE := EnableFB,
5      FILE_NAME := 'MyFileName',
6      REC_INTERVAL := T#100MS,
7      NUMBER_OF_FILES := 1,
8      MODE := DATALOGGER_MODE.CONTINUOUS_MODE,
9      CIRCULAR := 0,
10     STORAGE := DATALOGGER_STORAGE.INTERNAL_MEMORY,
11     FILE_RECORDS := 100,
12     COMMA_DELIMITER := TRUE,
13     FILE_HEADER := 'H1;H2',
14     INPUT001 := Data,
15     INPUT002 := Data + 1,
16     STATUS => Status,
17     ERROR => Error,
18     CURRENT_RECORDS => Records
19 );

```

Na parte de VAR, declarar o bloco de função em si e uma variável para armazenar o status, como mostrado. OBS: os nomes podem ser diferentes

No código, chamar a função como demonstrado. OBS: Para uma descrição de cada parâmetro, consultar próxima página.

Os parâmetros de entrada estão descritos na sequência.

ENABLE: o parâmetro ENABLE habilita ou desabilita o bloco funcional. O parâmetro é do tipo BOOL e assume os valores TRUE e FALSE para ligado e desligado respectivamente.

FILE_NAME: o parâmetro FILE_NAME é do tipo STRING e define o nome do arquivo (sem o .csv) e permite um valor de até 20 caracteres. (caso o parâmetro possua mais de 20 caracteres, o bloco considerará apenas as 20 primeiras caracteres).

STORAGE: esse parâmetro é do tipo `DATALOGGER_STORAGE` e define o dispositivo onde os registros serão armazenados. O valor `INTERNAL_MEMORY` armazenará os dados na memória interna da CPU e o valor `MEMORY_CARD` armazenará os dados no cartão de memória (quando inserido). No Nexto Xpress, o valor `USB_MASS_STORAGE` salva os dados no dispositivo conectado na porta USB.

FILE_RECORDS: parâmetro do tipo `UINT` que define o número de registros que serão escritos no arquivo. Intervalo de valores permitidos: de 100 até 65535. Esse parâmetro é válido somente para o modo contínuo.

NUMBER_OF_FILES: parâmetro do tipo `UINT` cuja função é definir o número máximo de arquivos a serem armazenados. Intervalos permitidos: de 1 até 65535.

MODE: este parâmetro do tipo `DATALOGGER_MODE` pode assumir dois valores:

- **CONTINUOUS_MODE:** os dados são registrados e escritos no arquivo de log continuamente em um intervalo de amostragem definido pelo usuário. O número de registros por arquivo é de no máximo 65535 (`FILE_RECORDS`) e no mínimo 100, sendo o limite de arquivos (`NUMBER_OF_FILES`) limitado pela memória ou em 65535 arquivos, que é o valor máximo que a variável pode assumir. Esse modo suporta a funcionalidade `CIRCULAR`, que será explicada posteriormente.
- **TRIGGER_MODE:** idem ao modo contínuo, porém os dados serão escritos no arquivo de log somente na ocorrência de um trigger. Ao ocorrer um trigger, um novo arquivo é criado, onde serão escritos os registros anteriores e posteriores à ocorrência, conforme definição do usuário. É possível armazenar 2001 registros por arquivos, ou seja, 1000 registros correspondentes ao `TRIGGER_BEFORE`, 1000 ao `TRIGGER_AFTER` e mais 1 no momento do trigger. No modo `TRIGGER`, o número máximo de arquivos (`NUMBER_OF_FILES`) também será limitado pelo espaço de memória e limite da variável, que é de 65535 arquivos, isso se deve ao fato de que cada borda de subida na entrada `TRIGGER` cria um novo arquivo. Esse modo também suporta a funcionalidade `CIRCULAR`.

CIRCULAR: parâmetro do tipo `BOOL` que habilita o modo circular. Ele pode assumir os valores `TRUE` e `FALSE` para habilitado e desabilitado respectivamente. O modo circular possui a funcionalidade de, após atingir a quantidade máxima de arquivos, reabrir o primeiro arquivo e sobrescrevê-lo, ou seja, a aplicação fica sempre gravando de forma a manter a mesma quantidade de arquivos, sendo assim, circular.

TRIGGER_BEFORE: do tipo `UINT`, define o número de registros anteriores à ocorrência do disparo do trigger que devem ser escritos no arquivo. Intervalo de valores permitidos: de 1 até 1000. Parâmetro válido apenas para o modo trigger.

TRIGGER_AFTER: do tipo `UINT`, define o número de registros posteriores à ocorrência do disparo do trigger que devem ser escritos no arquivo. Intervalo de valores permitidos: de 1 até 1000. Parâmetro válido apenas para o modo trigger.

TRIGGER: entrada do disparo externo, ativado por uma borda de subida. Tipo `BOOL`.

REC_INTERVAL: do tipo `TIME`, define o Intervalo em que os dados serão registrados. Para o modo `TRIGGER`, o valor deve ser $\geq 10\text{ms}$ e para o modo `CONTÍNUO` o valor deve ser $\geq 100\text{ms}$.

COMMA_DELIMITER: parâmetro do tipo `BOOL`, quando seu valor é `TRUE` as colunas do arquivo no formato CSV são delimitadas por vírgulas e o ponto decimal é um ponto. Caso contrário (valor `FALSE`), as colunas do arquivo no formato CSV são delimitadas por ponto e vírgula, e o ponto decimal é uma vírgula.

FILE_HEADER: do tipo `STRING`, define os Cabeçalhos dos dados que serão armazenados no arquivo. Campo de preenchimento obrigatório, com os cabeçalhos separados por ponto e vírgula. O número máximo de caracteres suportado por esta entrada é de 3307.

INPUT1...INPUTN, N \leq 256: tipo `REAL`. Entrada de dados. O Datalogger suporta até 256 entradas de dados.

Os parâmetros de saída são detalhados a seguir.

DONE: esse parâmetro do tipo BOOL retorna TRUE caso a função foi executada e FALSE se a função não foi executada.

BUSY: do tipo BOOL assume TRUE se a função está registrando os dados na memória e FALSE se a função não está registrando os dados na memória, ou entrou no estado de erro.

ERROR: parâmetro que retorna TRUE caso ocorra um erro ou a função seja abortada e FALSE quando não há nenhum erro. Parâmetro do tipo BOOL.

STATUS: indica o status do Datalogger. Tipo DATALOGGER_STATUS.

CURRENT_RECORDS: do tipo UDINT, indica o número de registros desde o início do processo.

MISSED_TRIGGERS: contador de disparos ignorados pelo Datalogger durante período em que a função está funcionando no modo trigger. Tipo UDINT.

Após compilar o código para o CP, para acessar o arquivo de log no computador basta ir em "Device" -> "Arquivos" e na parte de Execução, selecionar "Internal Memory" caso tenha salvado em memória interna e "Memory Card" caso tenha salvado em cartão de memória. Dentro da pasta haverá outra pasta chamada "Datalogger" e dentro dessa pasta estarão os logs. Para transferi-los para o computador, selecione o diretório desejado em Host e clique em "Carreg".

Após isso, seu arquivo de log deverá estar no diretório selecionado. Para abri-lo, será necessário um editor de tabelas como o Microsoft Excel ou similar.

Para auxiliá-lo na realização dessa atividade você pode consultar na Base de Conhecimento Altus, o Webinar "Seu PLC pode ser um DataLogger" ilustrado a seguir (<https://www.youtube.com/watch?v=L9cG8vx1Vw4>).



WEBINAR
SEU PLC PODE SER
UM DATALOGGER

Quando: 24/11
Horário: 10h

altus

The banner features a line graph with two data series, 'Spalte B' (red) and 'Spalte C' (blue), plotted against a y-axis from 0 to 70 and an x-axis with dates from 2017/05/09 to 2017/05/14. The graph shows 'Spalte C' fluctuating between approximately 40 and 65, while 'Spalte B' remains relatively flat around 10. The Altus logo is in the bottom right corner.



Um pouco de teoria... Conheça os nove pilares da Indústria 4.0 e sua relevância para a atividade industrial

Surgido na Alemanha por volta de 2012, o conceito da Indústria 4.0 envolve as inovações tecnológicas nos campos de automação e tecnologia da informação para manufatura. Com o objetivo base de criar processos mais rápidos, flexíveis e eficientes, a quarta revolução industrial promove a união dos recursos físicos e digitais, conectando máquinas, sistemas e ativos a fim de produzir itens de maior qualidade a custos reduzidos. Para alcançar estes resultados, é necessário gerar um alto nível de articulação entre as principais tecnologias que formam o conceito, os chamados pilares.

A figura a seguir ilustra a interconexão dessas tecnologias.



Confira abaixo quais são os nove pilares da Indústria 4.0 e qual sua relevância para a atividade industrial:

Análise de dados

Análise e gestão de grandes quantidades de dados propicia o aumento de performance e otimização dos processos industriais, equalizando o consumo de energia com a qualidade de produção ao propiciar uma melhor leitura de cenários e tomadas de decisão mais velozes.

Robótica

Ao incorporar robôs inteligentes aos processos da Indústria, o setor ganha em desempenho e disponibilidade, deixando a execução de tarefas de produção logísticas e repetitivas a cargo das máquinas. Além de reduzir os custos, estes robôs representam um importante aumento na produção.

Simulação

Na indústria 4.0, a simulação computacional é utilizada em plantas industriais para análise dados em tempo real, aproximando o mundo físico e virtual, e no aperfeiçoamento em configurações de máquinas para testar o próximo produto na linha de produção virtual antes de qualquer mudança real, gerando otimização de recursos, melhor performance e mais economia.

Integração de sistemas

Atualmente, nem todos os sistemas são totalmente integrados, faltando uma coesão entre empresa-clientes e até mesmo o processo de produção de uma indústria carece de uma integração plena. A indústria 4.0 propõe uma melhor harmonia entre todos que façam parte do ecossistema, garantindo uma gestão integral de experiência para que cadeias de valor sejam realmente automatizadas.

Internet das Coisas (IoT)

A internet das coisas (em inglês, IoT – Internet of Things) consiste na conexão entre rede de objetos físicos, ambientes, veículos e máquinas por meio de dispositivos eletrônicos embarcados, permitindo uma coleta e troca de informações mais rápida e efetiva. Na indústria de produtos e serviços, a IoT representa a integração de tecnologias que antes não estavam conectadas e que agora estão interligadas por meio de uma rede baseada em IP.

Cibersegurança

A indústria do futuro demanda que todas as áreas da empresa estejam conectadas, tanto as redes corporativas (TI) quanto as de automação e operacionais (TA). Desta forma, é fundamental que as empresas contem com sistemas de cibersegurança robustos para proteger sistemas e informações de possíveis ameaças e falhas que podem causar transtornos na produção.

Cloud computing

O número de tarefas relacionadas à produção de bens e serviços na Indústria tem crescido cada vez mais, demandando o uso de aplicativos e dados compartilhados entre diferentes localidades e sistemas para além dos limites dos servidores de uma empresa. A computação em nuvem fornece recursos que refletem em uma importante redução de custo, tempo e eficiência na execução destas tarefas.

Manufatura aditiva

Também conhecida como impressão em 3D, este pilar envolve a produção de peças a partir de camadas sobrepostas de material, normalmente em forma de pó, para se obter um modelo 3D. Esta estratégia pode ser utilizada para criar produtos personalizados que oferecem vantagens de construção e desenhos complexos.

Realidade aumentada

Utilizando os recursos deste pilar, é possível, por exemplo, enviar instruções de montagem via celular para o desenvolvimento de peças de protótipo e utilizar óculos de realidade aumentada para a gestão e operação de determinadas máquinas, melhorando procedimentos de trabalho.

Fonte: <https://www.altus.com.br/post/212/conheca-os-nove-pilares-da-industria-4-0-e-sua-relevancia-para-a-atividade-industrial>

Situação de Aprendizagem Complementar 6-2: o Nexto Xpress e a Indústria 4.0

Essa situação de aprendizagem visa familiarizá-lo com os principais conceitos associados à Indústria 4.0 correlacionando-os com as funcionalidades disponíveis no Nexto Xpress. Para tanto, consulte a seção de “Soluções” no site da Altus onde são apresentadas as soluções para controle de processos e tomada de decisão para Indústria 4.0.

O LINK para acesso é: <https://www.altus.com.br/solucoes/37/industria-4-0>.

Situação de Aprendizagem Complementar 6-3: utilização do protocolo MQTT

Essa situação de aprendizagem objetiva apresentar os passos necessários para utilização do protocolo MQTT enfatizando o uso da biblioteca LibMQTT.

Orientações gerais referentes à realização da atividade

Primeiramente é necessário ativar a biblioteca LibMQTT e logo após criar uma POU chamada MQTT_START do tipo texto estruturado, onde vamos criar variáveis para satisfazer as condições do bloco MQTT_CLIENT. Na PRG MQTT_START, vamos declarar as seguintes variáveis:

```
PROGRAM MQTT_START
VAR
  // Bloco de funções -> Bibliotecas LibMQTT
  MQTT : MQTT_CLIENT;
  CONN : MQTT_CONN_CONFIG;

  //Habilitar
  Enable : BOOL;

  // Parâmetros do cliente (conexão)
  ClientId : STRING(255) := '';
  CleanSession : BOOL := TRUE;

  // Parâmetros do Broker (Conexão)
  Hostname : STRING(255) := 'XXX.XXX.XX.XXX'; // IP da máquina ou do servidor.
  Port : UINT := 1883;
  KeepAlive : UINT := 60;

  // Credenciais (conexão)
  Username : STRING(255) := '';
  Password : STRING(255) := '';

  // Última Mensagem (Conexão)
  LastWillTopic : STRING(1024) := '';
  LastWillMessage : STRING := ''; // Pode ser de qualquer tipo (por exemplo, STRING, DWORD, BYTE ...)
  LastWillMsgSize : UINT := 0;
  LastWillMsgQoS : MQTT_QoS := MQTT_QOS_0;
  LastWillMsgRet : BOOL := FALSE;

  // Criptografia TLS (conexão)
  EnableTLS : BOOL := FALSE;
  Certificate : STRING(255) := '';

  // Saídas (Conexão)
  ConnectionState : BOOL;
  ErrorState : BOOL;
  ErrorCode : MQTT_ERR_CODE;
  Publish_Status : ARRAY [1..gc_uiMaxPubs] OF BOOL;

  // Saídas (mensagem recebida)
  LastTopic : STRING(1024);
END_VAR
```

```

  // Saídas (Conexão)
  ConnectionState : BOOL;
  ErrorState : BOOL;
  ErrorCode : MQTT_ERR_CODE;
  Publish_Status : ARRAY [1..gc_uiMaxPubs] OF BOOL;

  // Saídas (mensagem recebida)
  LastTopic : STRING(1024);
END_VAR
```

Não se esqueça de substituir o Hostname ('xxx.xxx.xx.xxx') pelo IP do seu computador. Na estrutura da MQTT_START (PRG), vamos ligar as variáveis criadas com as variáveis do bloco.

```

CONN.sClientId      := ClientId;
CONN.bCleanSession  := CleanSession;
CONN.sHostname     := Hostname;
CONN.uiPort        := Port;
CONN.uiKeepAlive    := KeepAlive;
CONN.sUser         := Username;
CONN.sPass         := Password;
LastWillMsgSize    := SIZEOF (LastWillMessage);
CONN.sLastWillTopic := LastWillTopic;
CONN.pbLastWillPayload := ADR (LastWillMessage);
CONN.uiLastWillPayloadSize := LastWillMsgSize;
CONN.eLastWillMessageQoS := LastWillMsgQoS;
CONN.bLastWillRetain := LastWillMsgRet;
CONN.bEnableTLS    := EnableTLS;
CONN.sCertFilename := Certificate;

IF ErrorState = TRUE THEN
  Enable := FALSE;
END_IF

MQTT.ENABLE      := Enable;
MQTT.CONNECTION_CONFIG := CONN;
MQTT.SUBSCRIBE   := SUBS;
MQTT.PUBLISH     := PUBS;
ConnectionState := MQTT.CONNECTED;
ErrorState       := MQTT.ERROR;
ErrorCode        := MQTT.ERROR_CODE;
Publish_Status   := MQTT.PUBLISH_STATUS;
LastTopic        := MQTT.LAST_TOPIC;
RCV              := MQTT.RECEIVED_MESSAGES;

```

```

MQTT();
Enable := TRUE;

```

As variáveis que devem ser declaradas em uma GVL são mostradas na sequência.

```

VAR GLOBAL
  PubEnable : BOOL;

  SUBS : ARRAY[1..gc_uiMaxSubs] OF MQTT_SUBSCRIBE;
  PUBS : ARRAY[1..gc_uiMaxPubs] OF MQTT_PUBLISH;
  RCV  : ARRAY[1..gc_uiMaxSubs] OF MQTT_RECEIVED;
  AIO_Real: REAL;
  AII_Real: REAL;
END_VAR

```

Agora vamos criar uma POU para publicar em alguns tópicos. nessa situação de aprendizagem nomearemos como MQTT_PUBLISHER utilizando a linguagem texto estruturado assim como a outra. Nesta POU, vamos declarar as variáveis indicadas a seguir.

```

PROGRAM MQTT_PUBLISHER
VAR
  PubMessage : ARRAY[1..gc_uiMaxPubs] OF STRING;
  N_Pub      : UINT;

  //Esta variável armazenará o último valor publicado.
  PubMessage_old: STRING;
END_VAR

```

E na sua estrutura, vamos definir os tópicos em que desejamos escrever. Abaixo temos dois exemplos de códigos usados para publicar as informações.

```

//TÓPICO 1 - Entradas Digitais IO
N_Pub := 1;
FUBS[N_Pub].bEnablePublish := NOT MQTT_Start.Publish_Status[N_Pub];
PubMessage[N_Pub][0] := 48 + BOOL_TO_BYTE(I00); //48 é o decimal que repres
PubMessage[N_Pub][1] := 48 + BOOL_TO_BYTE(I01);
PubMessage[N_Pub][2] := 48 + BOOL_TO_BYTE(I02);
PubMessage[N_Pub][3] := 48 + BOOL_TO_BYTE(I03);
PubMessage[N_Pub][4] := 48 + BOOL_TO_BYTE(I04);
PubMessage[N_Pub][5] := 48 + BOOL_TO_BYTE(I05);
PubMessage[N_Pub][6] := 48 + BOOL_TO_BYTE(I06);
PubMessage[N_Pub][7] := 48 + BOOL_TO_BYTE(I07);

FUBS[N_Pub].sPublishTopic      := 'Altus2/I0';
FUBS[N_Pub].pbPublishPayload  := ADR(PubMessage[N_Pub]);
FUBS[N_Pub].uiPublishPayloadSize := 8;
FUBS[N_Pub].ePublishMessageQoS := MQTT_QOS_0; //Valor padrão
FUBS[N_Pub].bPublishRetain    := FALSE;      // Valor padrão

```

```

(* TÓPICO 2 - Saídas Digitais Q0*)
N_Pub := 2;
FUBS[N_Pub].bEnablePublish := NOT MQTT_Start.Publish_Status[N_Pub];
PubMessage[N_Pub][0] := 48 + BOOL_TO_BYTE(Q00); //48 é o decimal que represent
PubMessage[N_Pub][1] := 48 + BOOL_TO_BYTE(Q01);
PubMessage[N_Pub][2] := 48 + BOOL_TO_BYTE(Q02);
PubMessage[N_Pub][3] := 48 + BOOL_TO_BYTE(Q03);
PubMessage[N_Pub][4] := 48 + BOOL_TO_BYTE(Q04);
PubMessage[N_Pub][5] := 48 + BOOL_TO_BYTE(Q05);
PubMessage[N_Pub][6] := 48 + BOOL_TO_BYTE(Q06);
PubMessage[N_Pub][7] := 48 + BOOL_TO_BYTE(Q07);

// Somente publicará o valor, se o valor atual for diferente do valor antigo
IF (PubMessage[N_Pub] <> PubMessage_old) THEN

    PubMessage_old      := PubMessage[N_Pub];
    FUBS[N_Pub].bEnablePublish := PubEnable;

END_IF

```

```

FUBS[N_Pub].sPublishTopic      := 'Altus2/Q0';
FUBS[N_Pub].pbPublishPayload  := ADR(PubMessage[N_Pub]);
FUBS[N_Pub].uiPublishPayloadSize := 8;
FUBS[N_Pub].ePublishMessageQoS := MQTT_QOS_0; // Valor padrão
FUBS[N_Pub].bPublishRetain    := FALSE;      // Valor padrão

```

```

(* TÓPICO 3 - Entradas Analógicas AI0*)
N_Pub := 3;
FUBS[N_Pub].bEnablePublish      := NOT MQTT_Start.Publish_Status[N_Pub];
PubMessage[N_Pub]               := INT_TO_STRING(AI0);
FUBS[N_Pub].sPublishTopic       := 'Altus2/AI0';
FUBS[N_Pub].pbPublishPayload    := ADR(PubMessage[N_Pub]);
FUBS[N_Pub].uiPublishPayloadSize := LEN(PubMessage[N_Pub]);

```

Agora devemos criar a POU que irá ler os tópicos, no nosso caso a nomeamos como MQTT_SUBSCRIBER e ela deve estar na linguagem ST. Nesta POU, vamos declarar as variáveis indicadas a seguir.

```

PROGRAM MQTT_SUBSCRIBER
VAR
    N_Sub :   UINT;
    RcvMsg1 : STRING; //Pode ser de qualquer tipo (por exemplo, STRING, DWORD, BYTE ...)
    RcvMsg2 : STRING; //Pode ser de qualquer tipo (por exemplo, STRING, DWORD, BYTE ...)
END_VAR

```

Da mesma forma como definimos a estrutura de escrita, vamos definir a de leitura de tópicos conforme mostrado a seguir.

```

//TÓPICO 1 - Saídas Digitais Q0
N_Sub := 1;
SUBS[N_Sub].sSubscribeTopic := 'Altus2/Q0';
SUBS[N_Sub].pbPayloadBuffer := ADR(RcvMsg1);
SUBS[N_Sub].uiMaxPayloadSize := SIZEOF(RcvMsg1);
SUBS[N_Sub].eSubscribeQoS := MQTT_QOS_0; //Valor p
Q00 := BYTE_TO_BOOL(RcvMsg1[0] - 48); (*48 é o decimal
Q01 := BYTE_TO_BOOL(RcvMsg1[1] - 48);
Q02 := BYTE_TO_BOOL(RcvMsg1[2] - 48);
Q03 := BYTE_TO_BOOL(RcvMsg1[3] - 48);
Q04 := BYTE_TO_BOOL(RcvMsg1[4] - 48);
Q05 := BYTE_TO_BOOL(RcvMsg1[5] - 48);
Q06 := BYTE_TO_BOOL(RcvMsg1[6] - 48);
Q07 := BYTE_TO_BOOL(RcvMsg1[7] - 48);

//-----
/** TÓPICO 2 - Saída Analógica A00*/
N_Sub := 2;
SUBS[N_Sub].sSubscribeTopic := 'Altus/A00';
SUBS[N_Sub].pbPayloadBuffer := ADR(RcvMsg2);
SUBS[N_Sub].uiMaxPayloadSize := SIZEOF(RcvMsg2);
A00 := STRING_TO_INT(RcvMsg2);

```

Devemos também definir em MQTTClient_Parameters a quantidade de tópicos em Subs (2 UINT) e em Pubs (6 UINT) que estamos utilizando.

Por último, vamos chamar as POU's na UserPRG.

Realizados os passos descritos anteriormente, você poderá realizar download do projeto no CP. Para testar o projeto podemos utilizar o broker mosquitto, usando o CMD do Windows. O mosquitto é um broker gratuito e pode ser facilmente instalado. Abra a pasta do mosquito localizada no diretório (C:), e digite o comando CMD na barra de diretórios e pressione ENTER. Abra três janelas prompt de comando.

Na primeira execute o mosquitto. Para executar o mosquito: C:\Program Files\mosquitto>mosquitto -v

Na segunda você deve receber as mensagens: C:\Program Files\mosquitto>mosquitto_sub -h localhost -p 1883 -t "Altus2/Q0", onde: Sub: Receber, -h: Servidor local, -p: porta 1883 e -t: Tópico de leitura.

Na terceira você irá publicar as mensagens: C:\Program Files\mosquitto>mosquitto_pub -h localhost -p 1883 -t "Altus2/Q0" -m "00010001" onde: Pub: Publicar, -h: Servidor local, -p: porta 1883, -t: Tópico de escrita e -m: mensagem. (Deve ser digitada entre aspas duplas "00011100").

Para auxiliá-lo na realização dessa atividade você pode consultar na Base de Conhecimento Altus, o Webinar “Desmistificando o MQTT” cujas partes 1 e 2 estão ilustradas na sequência (https://www.youtube.com/watch?v=3F_zVZoPU-M) e (https://www.youtube.com/watch?v=gYKq_9Tbm_0).



altus Webinar Desmistificando o MQTT (parte 1)

Assistir m... Compartilh...

WEBINAR

DESMISTIFICANDO O MQTT (PARTE 1)

Quando: 03/04
Horário: 10h

altus

This banner features a dark blue background with a subtle circuit pattern. It includes the Altus logo in the top left, a title bar, navigation icons, a large 'WEBINAR' header, a subtitle, a date and time box, and the Altus logo in the bottom right. A red play button icon is centered over the main text.



altus Webinar Desmistificando o MQTT (parte 2)

Assistir m... Compartilh...

WEBINAR

DESMISTIFICANDO O MQTT (PARTE 2)

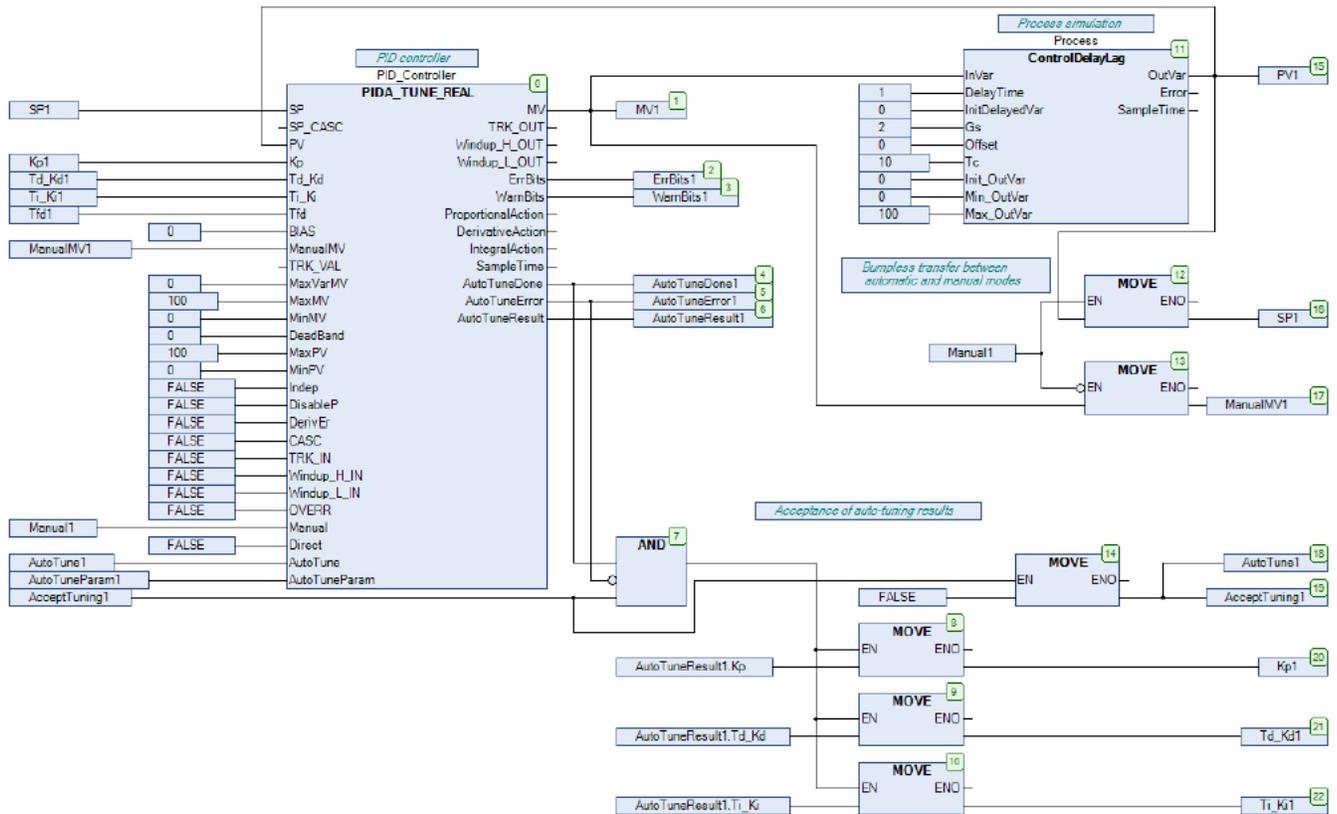
Quando: 09/04
Horário: 10h

altus

This banner features a dark green background with a subtle circuit pattern. It includes the Altus logo in the top left, a title bar, navigation icons, a large 'WEBINAR' header, a subtitle, a date and time box, and the Altus logo in the bottom right. A red play button icon is centered over the main text.

Situação de Aprendizagem Complementar 6-4: implementação de uma simulação de uma aplicação de controle PID com auto sintonia

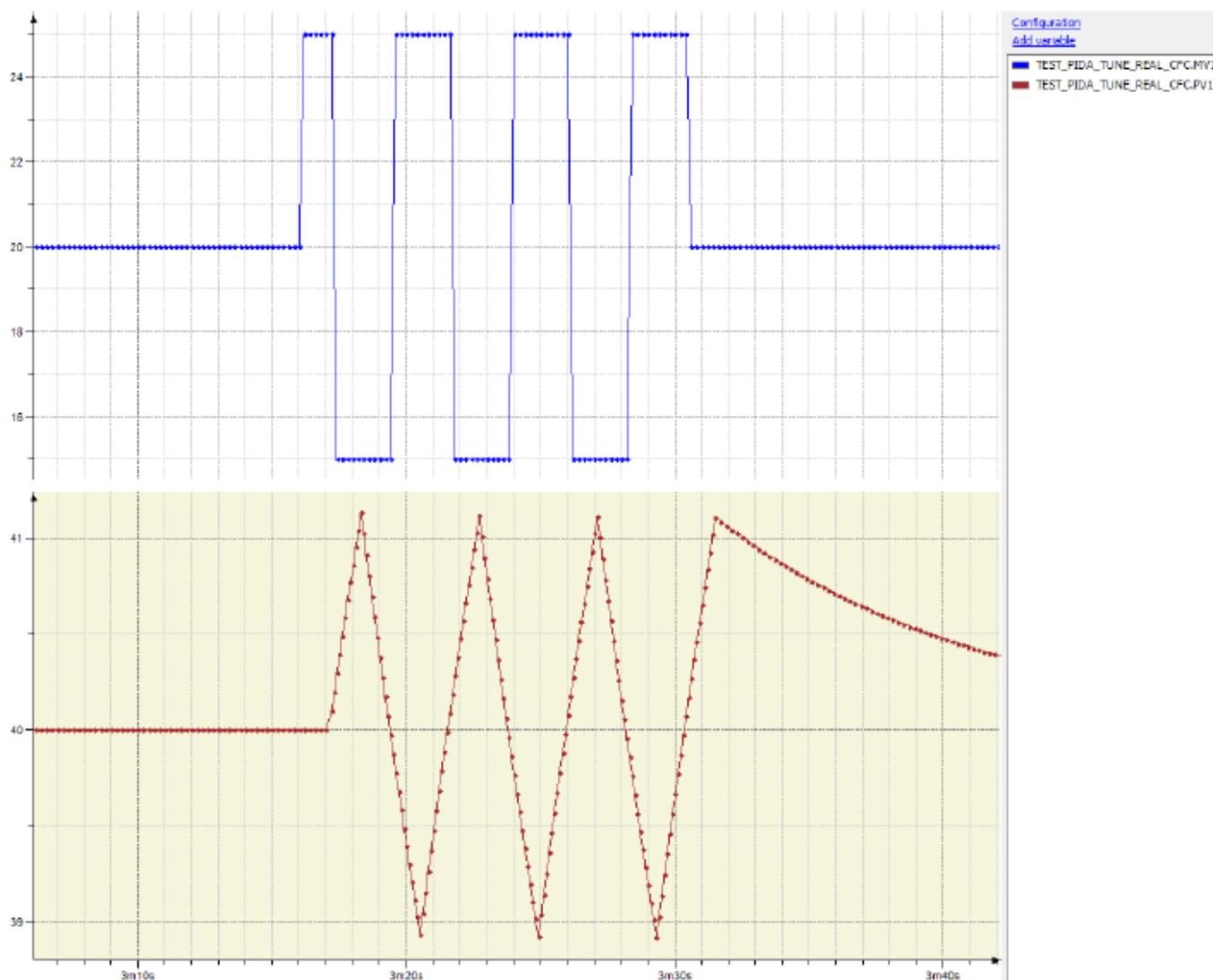
Esta situação de aprendizagem objetiva implementar uma simulação de uma aplicação de controle PID, usando a linguagem CFC, onde um bloco funcional PIDA_TUNE_REAL está conectado a um processo simulado de primeira ordem com tempo morto, usando o bloco funcional ControlDelayLag. Instruções adicionais foram adicionadas para uma transferência suave entre modos automático e manual, e para a sintonia automática. A figura a seguir mostra o código utilizado.



A figura a seguir mostra um exemplo de auto sintonia (saídas MV e PV durante o processo de auto-sintonia), considerando o código apresentado anteriormente. O operador SCADA executou o seguinte procedimento neste caso:

1. Ele mudou o controlador para o modo manual, com **ManualMV = 20**. Também seria possível executar a auto-sintonia no modo automático, mas é mais seguro no modo manual.
2. Ele ajustou os parâmetros de auto-sintonia dentro de **AutoTuneParam1**:
 - **PercAmpMV = 5%** (o passo de MV é 5, porque **MaxMV = 100** e **MinMV = 0**);
 - **PercHystPV = 0.1%** (histerese de PV é 0.1, porque **MaxPV = 100** e **MinPV = 0**);
 - **PercMaxPeakPV = 10%** (picos máximos de PV são 10, porque **MaxPV = 100** e **MinPV = 0**);
 - **NumCycles = 3** (3 ciclos de medição);
 - **Mode = AUTOTUNE_PID** (configuração PID).

3. Ele esperou até que PV estivesse quase estacionária. Neste caso, o estado estacionário é 40 ($MV = 20$, $G_s = 2$, $Offset = 0$).
4. Ele enviou um comando para definir AutoTune1, iniciando o processo de auto-sintonia.
5. Ele esperou até que AutoTuneDone1 fosse TRUE, indicando que a auto-sintonia foi concluída.
6. Ele enviou um comando para definir AcceptTuning1. Este comando copiará os parâmetros de sintonia, se AutoTuneError1 for FALSE.



Os parâmetros de sintonia calculados nesta aplicação foram:

- $K_p = 1.819$;
- $T_i_{K_i} = 9.680$ s;
- $T_d_{K_d} = 0.698$ s.

Consulte a documentação de produto (*Advanced Control Functions User Manual - MU214610*) e o instrutor para orientações adicionais referentes ao desenvolvimento dessa situação de aprendizagem.

Avaliação do Treinamento



Preencha a ficha de avaliação do treinamento.

Parabéns...

Se você seguiu a sequência indicada no tutorial e realizou as tarefas propostas com sucesso, você atingiu plenamente os objetivos desse treinamento!!!

Glossário

Algoritmo	Sequência finita de instruções bem definidas, objetivando à resolução de problemas.
Árvore	Estrutura de dados para configuração do hardware.
Backup	Cópia de segurança de dados.
Barramento	Conjunto de módulos de E/S interligados a uma UCP ou cabeça de rede de campo.
Barramento local	Conjunto de módulos de E/S interligados a uma UCP.
Barramento remoto	Conjunto de módulos de E/S interligados a uma cabeça de rede de campo.
Bit	Unidade básica de informação, podendo estar no estado 0 ou 1.
Breakpoint	Ponto de parada no aplicativo para depuração.
Byte	Unidade de informação composta por oito bits.
Canal serial	Interface de um equipamento que transfere dados no modo serial.
Ciclado	Modo de execução do CP passo-a-passo, onde cada passo é um ciclo do CP.
Ciclo de varredura	Uma execução completa do programa aplicativo de um controlador programável.
Circuito de cão-de-guarda	Circuito eletrônico destinado a verificar a integridade do funcionamento de um equipamento.
Controlador programável	Também chamado de CP. Equipamento que realiza controle sob o comando de um programa aplicativo. É composto de uma UCP, uma fonte de alimentação e uma estrutura de E/S.
CP	Veja controlador programável.
Default	Valor predefinido para uma variável, utilizado em caso de não haver definição.
Diagnóstico	Procedimento utilizado para detectar e isolar falhas. É também o conjunto de dados usados para tal determinação, que serve para a análise e correção de problemas.
Download	Carga de programa ou configuração no CP.
E/S	Veja entrada/saída.
Entrada/saída	Também chamado de E/S. Dispositivos de E/S de dados de um sistema. No caso de CPs, correspondem tipicamente a módulos digitais ou analógicos de entrada ou saída que monitoram ou acionam o dispositivo controlado.
Escravo	Equipamento ligado a uma rede de comunicação que só transmite dados se for solicitado por outro equipamento denominado mestre.
Frame	Uma unidade de informação transmitida na rede.
Gateway	Equipamento ou software para a conexão de duas redes de comunicação com diferentes protocolos.
Hardware	Equipamentos físicos usados em processamento de dados onde normalmente são executados programas (software).
Hiperlink	Atalho de navegação para uma nova página do help.
IEC 61131	Norma genérica para operação e utilização de CPs. Antiga IEC 1131.
Interface	Dispositivo que adapta elétrica e/ou logicamente a transferência de sinais entre dois equipamentos.
Interrupção	Evento com atendimento prioritário que temporariamente suspende a execução de um programa e desvia para uma rotina de atendimento específica
kbytes	Unidade representativa de quantidade de memória. Representa 1024 bytes.
LED	Sigla para light emitting diode. É um tipo de diodo semiconductor que emite luz quando estimulado por eletricidade. Utilizado como indicador luminoso.
Linguagem de programação	Um conjunto de regras e convenções utilizado para a elaboração de um programa.
Local host	Máquina, PC ou sistema que está em uso.
Login	Ação de estabelecer um canal de comunicação com o CP.
Menu	Conjunto de opções disponíveis e exibidas por um programa no vídeo e que podem ser selecionadas pelo usuário a fim de ativar ou executar uma determinada tarefa.
Menu de Contexto	Menu dinâmico com o conteúdo de acordo com o contexto atual.
Mestre	Equipamento ligado a uma rede de comunicação de onde se originam solicitações de comandos para outros equipamentos da rede.
Módulo (referindo-se a hardware)	Elemento básico de um sistema completo que possui funções bem definidas. Normalmente é ligado ao sistema por conectores, podendo ser facilmente substituído.
Módulo de E/S	Módulo pertencente ao subsistema de entradas e saídas.
Multicast	Disseminação simultânea de informação a um determinado grupo de nós interligados a uma rede de comunicação.

Nó	Qualquer estação de uma rede com capacidade de comunicação utilizando um protocolo estabelecido.
Operandos	Elementos sobre os quais as instruções atuam. Podem representar constantes, variáveis ou um conjunto de variáveis.
PC	Sigla para programmable controller. É a abreviatura de controlador programável em inglês.
PDO	Do inglês, é uma mensagem do protocolo CAN contendo os dados operacionais.
Ping	Do inglês, é um comando usado pelo protocolo ICMP que serve para testar a conectividade entre equipamentos e foi criado para o uso em redes com a pilha de protocolo TCP/IP.
POU	, ou Unidade de Organização de Programa, é uma subdivisão do programa aplicativo que pode ser escrito em qualquer uma das linguagens disponíveis.
Programa aplicativo	É o programa carregado em um CP, que determina o funcionamento de uma máquina ou processo.
Protocolo	Regras de procedimentos e formatos convencionais que, mediante sinais de controle, permitem o estabelecimento de uma transmissão de dados e a recuperação de erros entre equipamentos.
RAM	Sigla para random access memory. É a memória onde todos os endereços podem ser acessados diretamente de forma aleatória e com a mesma velocidade. É volátil, ou seja, seu conteúdo é perdido quando o equipamento é desenergizado, a menos que se possua uma bateria para a retenção dos valores.
Rede de comunicação	Conjunto de equipamentos (nós) interconectados por canais de comunicação.
Reset	Comando para reinicializar o CP.
RUN	Comando para colocar o CP em modo de execução.
Set	Ação para atribuir o estado de nível lógico alto para uma variável booleana.
Software	Programas de computador, procedimentos e regras relacionadas à operação de um sistema de processamento de dados.
STOP	Comando para congelar o CP em seu estado atual.
Sub-rede	Segmento de uma rede de comunicação que interliga um grupo de equipamentos (nós) com o objetivo de isolar o tráfego local ou utilizar diferentes protocolos ou meio físicos.
Time-out	Tempo preestabelecido máximo para que uma comunicação seja completada. Se for excedido procedimentos de retentiva ou diagnóstico serão ativados.
Tooltip	Caixa de texto com uma ajuda ou local onde pode-se entrar com a ajuda.
UCP	Sigla para unidade central de processamento. Controla o fluxo de informações, interpreta e executa as instruções do programa e monitora os dispositivos do sistema.
UCP ativa	Em um sistema redundante, a UCP ativa realiza o controle do sistema, lendo os valores dos pontos de entrada, executando o programa aplicativo e acionando os valores das saídas.
Upload	Leitura do programa ou configuração do CP.
Visualização	Conjunto de telas do CP.
XML	Do inglês, Extensible Markup Language, é um padrão para gerar linguagens de marcação.
Zoom	No contexto da janela de função do teclado, é utilizado para a troca de telas.