



www.altus.com.br

APOSTILA DE TREINAMENTO

SÉRIE NEXTO

AVANÇADO



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 é marca registrada 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	7
O Curso MP10a	8
Estruturação do Tutorial	9
Ementa do curso	9
Referências	11
Documentos relacionados	11
Avaliação do curso	12
Terminologia	12
Convenções Utilizadas	12
Suporte Técnico	14
Base de Conhecimento ALTUS	15
Cursos ALTUS	15
Revisões deste Tutorial	15
Training Box Nexto	16
1. Programação avançada no CP Nexto	1-1
A norma IEC61131	1-1
Estrutura da IEC61131	1-1
Elementos comuns	1-2
Linguagens de programação	1-4
Aplicação da IEC 61131-3	1-6
O Nexto no contexto da IEC61131-3	1-7
O ambiente de programação MasterTool IEC XE	1-7
Conceitos Básicos	1-8
Funcionalidades Avançadas	1-8
Perfis de projeto	1-10
Projeto	1-13
Dispositivos	1-13
Aplicação	1-14
Task Configuration	1-14
Geração de Código e Informações de Compilação	1-15
Alterações Online	1-15
Aplicação de Inicialização (Projeto de Inicialização)	1-15
Monitoração	1-16
Depuração	1-16
Unidade de Organização de Programa (POU)	1-17
POU	1-18
Programa	1-19
Função	1-20
Bloco Funcional	1-21
Estudo Dirigido 1-1: declaração de blocos funcionais	1-25
Unidades de Tipo de Dados	1-28
Estudo Dirigido 1-2: declaração de unidades de tipos de dados	1-28
Método	1-30
Propriedade	1-31
Ação	1-32
Lista de Variáveis Globais - GVL	1-33

Variáveis Persistentes	1-33
Arquivo Externo	1-34
POUs para Verificações Implícitas	1-34
Gerenciamento de Bibliotecas	1-34
Linguagens de Programação IEC61131-3	1-35
Diagrama de Blocos Funcionais (FBD)	1-35
Lista de Instruções (IL)	1-35
Texto Estruturado (ST)	1-36
Estudo Dirigido 1-3: declaração do bloco funcional Lim_Taxa	1-37
Sequenciamento Gráfico de Funções (SFC)	1-39
Um pouco de teoria... IEC61131 – Parte 8: orientações para aplicação e implementação das linguagens de programação	1-40
2. Comunicação avançada no CP Nexto	2-1
Configuração das Interfaces Seriais	2-1
Protocolo MODBUS na Série Nexto	2-3
MODBUS RTU	2-6
MODBUS Ethernet	2-14
Estudo Dirigido 2-1: configuração do protocolo MODBUS Ethernet	2-24
Estudo Dirigido 2-2: aplicação do protocolo MODBUS Ethernet	2-28
Protocolo OPC na Série Nexto	2-31
OPC DA Servidor	2-31
OPC UA Servidor	2-38
Estudo Dirigido 2-3: configurando o Nexto como servidor OPC UA	2-43
Outros Protocolos das Interfaces Ethernet	2-45
EtherCAT Mestre	2-45
EtherNet/IP	2-50
IEC 60870-5-104 Servidor	2-54
Protocolo PROFIBUS na Série Nexto	2-58
Um pouco de teoria... Protocolo Profibus	2-58
Mestre PROFIBUS DP NX5001	2-60
Estudo Dirigido 2-4: configuração Profibus	2-63
3. Recursos avançados no CP Nexto	3-1
Memórias Não-Voláteis	3-1
Memória de Arquivos de Usuário	3-1
Cartão de Memória (SD)	3-4
Variáveis Remanentes	3-6
Gerenciamento de Usuários e Acesso	3-9
Gerenciamento de Usuários	3-9
Gerenciamento de Direitos de Acesso	3-13
Gerenciamento de Usuários e Direitos de Acesso da UCP	3-14
Easy Connection	3-16
Listas de Variáveis Globais (GVLs)	3-17
GVL System_Diagnostics	3-17
GVL Disables	3-17
GVL IOQualities	3-18
GVL Module_Diagnostics	3-18
GVL Qualities	3-18
GVL ReqDiagnostics	3-19
Visualização	3-21
Visão Geral	3-21
Comandos de Visualização	3-24
Editor de Visualização	3-25
Gerenciamento de Visualização com Clientes	3-29
Visualização em Modo Online	3-29
Estudo Dirigido 3-1: criação de uma visualização WEB	3-29
4. Funcionalidades avançadas no CP Nexto	4-1
Um pouco de teoria... Visão geral do controle industrial	4-1

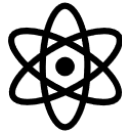
Bloco Funcional PID	4-6
Entradas do Bloco Funcional	4-6
Saídas do Bloco Funcional	4-7
Considerações Gerais sobre a Programação do Bloco Funcional PID	4-7
PID_fixcycle	4-7
PID Control	4-7
Ambiente Gráfico	4-8
Estudo Dirigido 4-1: utilização do PID Control	4-8
Estudo Dirigido 4-2: bloco funcional PID_S	4-20
5. Aplicações, Avaliação e Encerramento	5-1
Estudo de caso	5-1
Especificação funcional da automação proposta	5-1
Especificações do processo	5-2
Condições de contorno e restrições	5-2
Decomposição do controle	5-3
Lista de entradas e saídas	5-3
Modos de operação	5-3
Utilização de funções	5-4
Declaração de blocos funcionais	5-4
Acesso aos sinais dos sensores	5-4
Programas para sequenciamento e estruturação	5-4
Depuração	5-4
Tarefas para controle de execução	5-4
Árvore de POUs da Aplicação	5-4
Documentação	5-5
Situações de Aprendizagem	5-6
Avaliação do Treinamento	5-12
Parabéns...	5-12
Glossário	A

Figuras

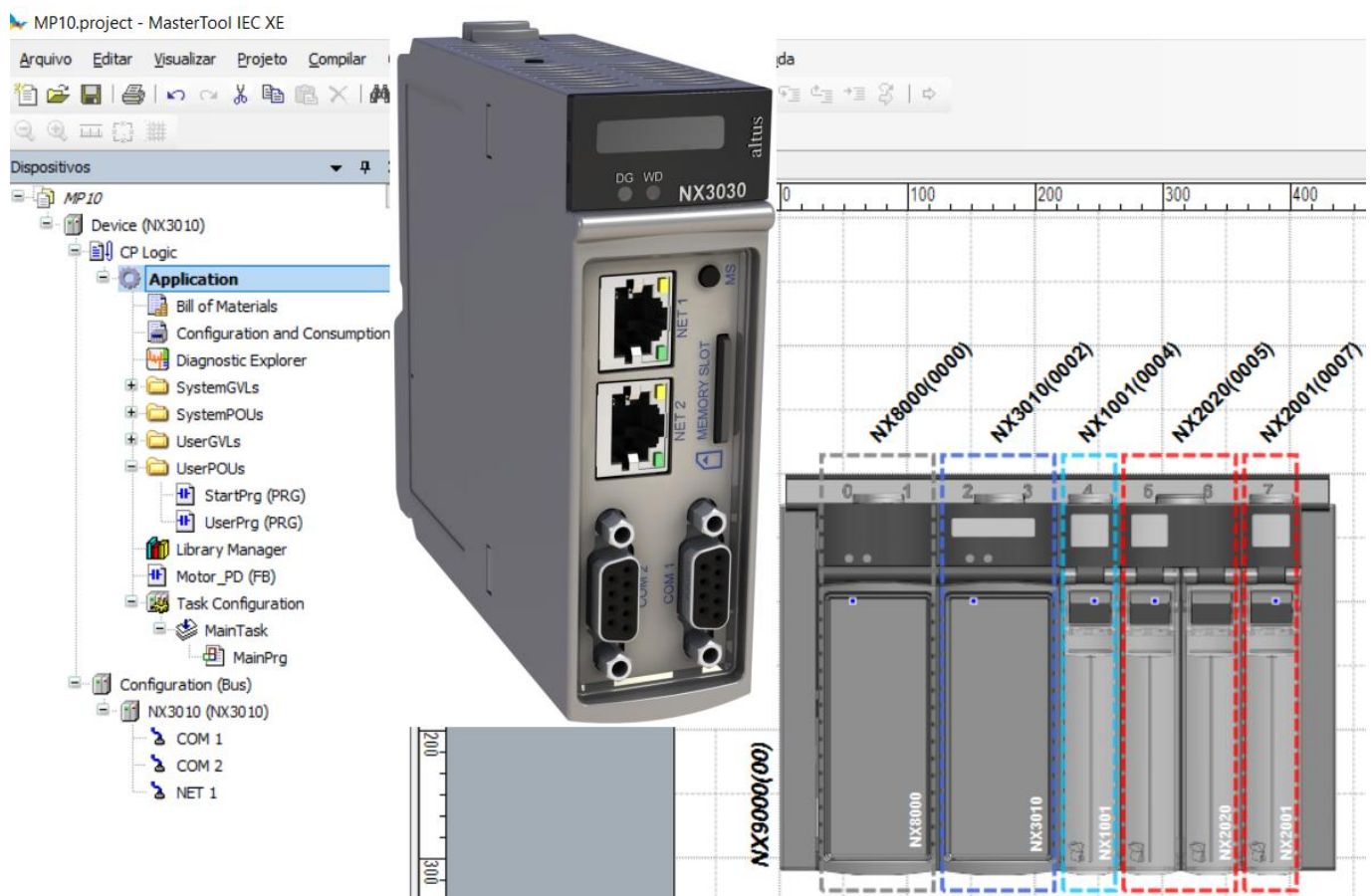
Figura 1-1. Diálogo Acrescentar POU _____	1-18
Figura 1-2. Exemplo de Programa _____	1-19
Figura 1-3. Propriedade Segundos Acrescentada ao Bloco Funcional fb _____	1-32
Figura 1-4. Ação em um Bloco Funcional _____	1-32
Figura 1-5. Redes de Diagrama de Blocos Funcionais _____	1-35
Figura 1-6. Exemplo de Programa IL no Editor Tabular IL _____	1-36
Figura 1-7. Exemplo para uma Sequência de Passos em um Módulo SFC _____	1-39
Figura 2-1. Interfaces seriais _____	2-1
Figura 2-2. Hierarquia Modbus _____	2-3
Figura 2-3. Protocolo Modbus _____	2-4
Figura 2-4. Comunicação Mestre-Escravo _____	2-4
Figura 2-5. Transmissão Modbus _____	2-5
Figura 2-6. Formato da Mensagem _____	2-5
Figura 2-7. Adicionando a Instância MODBUS _____	2-6
Figura 2-8. Seleção do protocolo MODBUS RTU Slave _____	2-7
Figura 2-9. Seleção do protocolo MODBUS RTU Slave _____	2-8
Figura 2-10. Tela de Configuração Parâmetros Gerais MODBUS RTU Mestre _____	2-8
Figura 2-11. Tela de Configurações dos Parâmetros Gerais do Dispositivo _____	2-9
Figura 2-12. Tela de Mapeamentos de Dados MODBUS _____	2-9
Figura 2-13. Tela de Requisições de dados MODBUS Mestre _____	2-9
Figura 2-14. Tela de Configuração MODBUS RTU Mestre _____	2-10
Figura 2-15. Configuração do dispositivo MODBUS _____	2-10
Figura 2-16. Tipo de dado MODBUS _____	2-11
Figura 2-17. Função MODBUS _____	2-11
Figura 2-18. Configuração do Escravo _____	2-12
Figura 2-19. Tela de Mapeamentos de dados MODBUS _____	2-12
Figura 2-20. Tela de Configuração MODBUS RTU Escravo _____	2-13
Figura 2-21. Adicionando Relações MODBUS _____	2-13
Figura 2-22. Configurando a Relação MODBUS _____	2-14
Figura 2-23. Rede de Comunicação MODBUS TCP _____	2-15
Figura 2-24. Adicionando a Instância MODBUS Ethernet _____	2-16
Figura 2-25. Selecionando o Protocolo MODBUS Ethernet _____	2-17
Figura 2-26. Tela de Configuração Parâmetros Gerais MODBUS Cliente _____	2-18
Figura 2-27. Tela de Configurações dos Parâmetros Gerais do Dispositivo MODBUS _____	2-18
Figura 2-28. Tela de Mapeamentos de dados MODBUS Cliente _____	2-18
Figura 2-29. Tela de Requisições de dados MODBUS _____	2-19
Figura 2-30. Tela de Configuração – MODBUS Cliente _____	2-19
Figura 2-31. Configurando Cliente MODBUS _____	2-20
Figura 2-32. Tipo de dado MODBUS _____	2-20
Figura 2-33. Função MODBUS _____	2-20
Figura 2-34. Tela de Configuração Parâmetros Gerais MODBUS Servidor _____	2-21
Figura 2-35. Tela de Configuração MODBUS Servidor _____	2-22
Figura 2-36. Tela de Configuração MODBUS Servidor _____	2-23
Figura 2-37. Tipo de dado MODBUS _____	2-23
Figura 2-38. Função MODBUS Servidor _____	2-24
Figura 2-39. Arquitetura OPC DA _____	2-32
Figura 2-40. Objeto Symbol Configuration _____	2-33
Figura 2-41. Selecionando Variáveis na Symbol Configuration _____	2-34
Figura 2-42. Configuração do OPC DA Servidor _____	2-35
Figura 2-43. Selecionando Servidor OPC DA na Configuração do Cliente _____	2-37
Figura 2-44. Lista de Símbolos Consultados pelo Cliente OPC DA _____	2-38

Figura 2-45. Arquitetura Típica OPC UA	2-39
Figura 2-46. Objeto Symbol Configuration (OPC UA)	2-40
Figura 2-47. Habilitando OPC UA no Objeto Symbol Configuration	2-41
Figura 2-48. Selecionando Servidor OPC UA na Configuração do Cliente	2-42
Figura 2-49. Lista de Símbolos Consultados pelo Cliente OPC UA	2-43
Figura 2-50. Exemplo de Configuração EtherCAT	2-46
Figura 2-51. Diálogo Procurar Dispositivos para EtherCAT	2-47
Figura 2-52. Diálogo de Configuração do Mestre EtherCAT	2-48
Figura 2-53. Diálogo de Configuração do Escravo EtherCAT	2-49
Figura 2-54. Adicionando uma Interface EtherNet/IP	2-51
Figura 2-55. Adicionando um Adapter ou Scanner EtherNet/IP	2-52
Figura 2-56. Adicionando um Adapter EtherNet/IP sob o Scanner	2-53
Figura 2-57. Adicionando um Módulo EtherNet/IP Abaixo do Adapter	2-54
Figura 2-58. Tela de Parâmetros Gerais do Servidor IEC 60870-5-104	2-56
Figura 2-59. Tela de Mapeamento dos Dados do Servidor IEC 60870-5-104	2-56
Figura 2-60. Tela de Configuração da Camada de Enlace do Servidor IEC 60870-5-104	2-56
Figura 2-61. Tela de Configuração da Camada de Aplicação do Servidor IEC 60870-5-104	2-57
Figura 2-62. Mestre PROFIBUS DP NX5001	2-60
Figura 2-63. Configuração PROFIBUS simples	2-62
Figura 2-64. Diagrama Elétrico do Mestre PROFIBUS DP NX5001	2-63
Figura 3-1. Acesso aos Arquivos de Usuário	3-2
Figura 3-2. Transferindo Arquivos	3-3
Figura 3-3. Opções de Utilização	3-4
Figura 3-4. Configurações do Cartão de Memória	3-5
Figura 3-5. Diretório com Cartão de Memória Inserido com Projeto	3-6
Figura 3-6. Configurações do Projeto, Diálogo Usuários	3-10
Figura 3-7. Acrescentar Usuário	3-11
Figura 3-8. Configurações de Projeto, Diálogo Grupos	3-12
Figura 3-9. Acrescentar Grupo	3-12
Figura 3-10. Configurações de Projeto, Diálogo Configuração de Usuários e Grupos	3-13
Figura 3-11. Permissões	3-14
Figura 3-12. Acesso às Abas Usuários e Grupos e Direitos de Acesso	3-15
Figura 3-13. Easy Connection	3-16
Figura 3-14. Mecanismo Geral	3-22
Figura 3-15. Servidor de Dados para fornecer dados de visualização	3-22
Figura 3-16. Cliente de visualização WebVisu	3-23
Figura 3-17. Caixa de diálogo de Configuração de Frame	3-24
Figura 3-18. Editor de Visualização	3-25
Figura 3-19. Diálogo 'Propriedades', Categoria 'Visualização'	3-27
Figura 4-1. Papel do Controlador em Sistemas de Controle Industriais	4-1
Figura 4-2. Malha de Controle	4-2
Figura 4-3. Sistema de Controle com Realimentação	4-3
Figura 4-4. Controle PID para Temperatura	4-4
Figura 4-5. Resposta típica ao degrau	4-5
Figura 4-6. Detalhe do período crítico (Pu)	4-5
Figura 4-7. Bloco Funcional PID	4-6
Figura 4-8. Aba Configurações Avançadas	4-16
Figura 5-1. Diagrama do estudo de caso 2	5-1
Figura 5-2. Árvore de POUs	5-5

Introdução



Este Tutorial fornece um suporte didático-pedagógico ao treinamento presencial CP-NEXTO avançado, possibilitando ao usuário um contato orientado com as funcionalidades estendidas do equipamento e enfatizando a programação do mesmo nas linguagens normatizadas pela IEC61131-3 via programador MasterTool IEC XE. Esta seção contém as divisões do tutorial e também as simbologias e convenções utilizadas. O objetivo é informar ao usuário como o documento pode ser consultado.



O Curso MP10a



O curso MP10a explora os recursos avançados do controlador programável Nexto, tais como funcionalidades de rede, protocolos de comunicação adotados, recursos interessantes e inovadores e utilização do bloco funcional PID. Além disso, o curso propõe o desenvolvimento de programas aplicativos multi-linguagens compatíveis com a IEC61131-3 através da utilização do ambiente de programação MasterTool IEC XE em perfis de usuário avançados.



Os objetivos desse curso são:

- Explorar as funcionalidades de rede do dispositivo através da descrição dos canais de comunicação disponíveis e detalhar os protocolos de comunicação adotados pelo NEXTO.
- Apresentar alguns recursos interessantes e inovadores do dispositivo, tais como retentividade, cartão SD e controle de usuário.
- Analisar algumas 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 6 linguagens gráficas e textuais de programação, completamente compatíveis com a norma IEC 61131-3.

Esse curso é destinado aos profissionais em automação, clientes da Altus, potenciais usuários, profissionais técnicos e engenheiros de áreas afins, assim como todos os interessados em adquirir conhecimentos sobre automação.

O curso tem uma duração de 24 horas e demanda como pré-requisito, a participação prévia do aluno no Curso MP10 — Série Nexto (Básico).

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 contato orientado avançado com o(s) equipamento(s) e enfatizando a programação do(s) mesmo(s).

O documento inclui ainda considerações sobre instalação, conexões elétricas e configurações avançadas. Ele é dividido em capítulos numa estrutura padronizada conforme esquematizado a seguir.

- ▷ 1. Programação avançada no CP Nexto
- ▷ 2. Comunicação avançada no CP Nexto
- ▷ 3. Recursos avançados no CP Nexto
- ▷ 4. Funcionalidades avançadas no CP Nexto
- ▷ 5. 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: Programação avançada no CP Nexto

Este capítulo explora a programação do CP NEXTO no contexto da norma IEC61131-3 através do ambiente de programação MasterTool IEC XE. Tópicos abordados:

- ▲ 1. Programação avançada no CP Nexto
 - ▷ A norma IEC61131
 - ▷ O Nexto no contexto da IEC61131-3
 - ▷ Unidade de Organização de Programa (POU)
 - ▷ Linguagens de Programação IEC61131-3

Capítulo 2: Comunicação avançada no CP Nexto

Neste capítulo são exploradas as funcionalidades de comunicação em rede do controlador programável, enfatizando os protocolos MODUBUS, OPC e PROFIBUS. Tópicos abordados:

- ▲ 2. Comunicação avançada no CP Nexto
 - ▷ Configuração das Interfaces Seriais
 - ▷ Protocolo MODBUS na Série Nexto
 - ▷ Protocolo OPC na Série Nexto
 - ▷ Outros Protocolos das Interfaces Ethernet
 - ▷ Protocolo PROFIBUS na Série Nexto

Capítulo 3: Recursos avançados no CP Nexto

Este capítulo explora algumas das características exclusivas e inovadoras que fazem da Série Nexto uma poderosa e completa série de Controladores Programáveis (CPs). Tópicos abordados:

- ▲ 3. Recursos avançados no CP Nexto
 - ▷ Memórias Não-Voláteis
 - ▷ Gerenciamento de Usuários e Acesso Easy Connection
 - ▷ Listas de Variáveis Globais (GVLs)
 - ▷ Visualização

Capítulo 4: Funcionalidades avançadas no CP Nexto

Este capítulo inclui a análise de algumas funcionalidades avançadas do dispositivo no que se refere às aplicações industriais, enfatizando a utilização do bloco funcional PID. Tópicos abordados:

- ▲ 4. Funcionalidades avançadas no CP Nexto
 - Um pouco de teoria... Visão geral do controle industrial
 - ▷ Bloco Funcional PID
 - ▷ PID Control

Capítulo 5: Aplicações, Avaliação e Encerramento

O tutorial finaliza com a realização de um estudo de caso para consolidação das competências desenvolvidas no treinamento, incluindo a proposição de situações de aprendizagem para aplicação das capacidades envolvidas. Tópicos abordados:

- ▲ 5. Aplicações, Avaliação e Encerramento
 - ▷ Estudo de caso
 - ▷ Avaliação do Treinamento
 - Glossário

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 (conforme indicado na sequência)

Documentos relacionados

Para obter informações adicionais sobre a Série Nexto podem ser consultados outros documentos (manuais e características técnicas) além deste. Estes documentos encontram-se disponíveis em www.altus.com.br. Cada produto possui um documento denominado Características Técnicas (CT), e neste documento encontram-se as características do produto em questão. Caso o produto possua mais informações, ele pode ter também um manual de utilização. Por exemplo, o módulo NX2020 tem todas as informações de características, utilização e de compra, na sua CT. Por outro lado, o NX5001 possui, além da CT, um manual de utilização. A tabela abaixo indica a lista de todos os documentos relacionados à Série Nexto.

Código	Descrição	Idioma
CE114000	Nexto Series – Technical Characteristics	English
CT114000	Série Nexto – Características Técnicas	Portuguese
CS114000	Serie Nexto – Especificaciones y Configuraciones	Spanish
CE114100	Nexto Series CPUs Technical Characteristics	English
CT114100	Características Técnicas UCPs Série Nexto	Portuguese
CS114100	Especificaciones y Configuraciones UCPs Serie Nexto	Spanish
CE114200	NX8000 Power Supply Module Technical Characteristics	English
CT114200	Características Técnicas Fonte de Alimentação NX8000	Portuguese
CS114200	Características Tecnicas del Fuente de Alimentación NX8000	Spanish
CE114810	Nexto Series Accessories for Backplane Rack Technical Characteristics	English
CT114810	Características Técnicas Acessórios para Bastidor Série Nexto	Portuguese
CS114810	Características Tecnicas del Cierres Laterales para el Bastidor	Spanish
CE114902	Nexto Series PROFIBUS-DP Master Technical Characteristics	English
CT114902	Características Tecnicas do Mestre PROFIBUS-DP da Série Nexto	Portuguese
CS114902	Características Técnicas del Módulo Profibus-DP Maestro	Spanish
CE114903	Nexto Series Ethernet Module Technical Characteristics	English
CT114903	Características Técnicas Módulo Ethernet Série Nexto	Portuguese
CS114903	Características Tecnicas del Modulo Ethernet Série Nexto	Spanish
CE114700	Nexto Series Backplane Rack Technical Characteristics	English
CT114700	Características Técnicas dos Bastidores da Série Nexto	Portuguese
CS114700	Características Tecnicas de los Bastidores de la Serie Nexto	Spanish
CE114900	NX4010 Redundancy Link Module Technical Characteristics	English
CT114900	Características Técnicas do Módulo de Redundância NX4010	Portuguese
CS114900	Características Tecnicas del Módulo de Redundancia NX4010	Spanish
MU216600	Nexto Xpress User Manual	English
MU216000	Manual de Utilização Nexto Xpress	Portuguese
MU214600	Nexto Series User Manual	English
MU214000	Manual de Utilização Série Nexto	Portuguese
MU214605	Nexto Series CPUs User Manual	English
MU214100	Manual de Utilização UCPs Série Nexto	Portuguese
MU299609	MasterTool IEC XE User Manual	English
MU299048	Manual de Utilização MasterTool IEC XE	Portuguese
MP399609	MasterTool IEC XE Programming Manual	English
MP399048	Manual de Programação MasterTool IEC XE	Portuguese

MU214601	NX5001 PROFIBUS DP Master User Manual	English
MU214001	Manual de Utilização Mestre PROFIBUS-DP Nexto	Portuguese
MU214608	Nexto PROFIBUS-DP Head Utilization Manual	English
MU214108	Manual de Utilização da Cabeça PROFIBUS-DP Nexto	Portuguese

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.

UCP ou CPU: Unidade Central de Processamento é o 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:

O rótulo **PERIGO** indica que risco de vida, danos pessoais graves ou prejuízos materiais substanciais resultarão se as precauções necessárias não forem tomadas.

CUIDADO:

O rótulo **CUIDADO** indica que risco de vida, danos pessoais graves ou prejuízos materiais substanciais podem resultar se as precauções necessárias não forem tomadas.

ATENÇÃO:

O rótulo **ATENÇÃO** indica que danos pessoais ou prejuízos materiais mínimos podem resultar se as precauções necessárias não forem tomadas.

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...



Home Produtos ▾ Soluções Suporte & Downloads Quem somos Contato ▾

SUPORTE & DOWNLOADS

Home > Suporte & Downloads

Precisa de ajuda? Fale com nosso Suporte Técnico!

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

Estamos 100% disponíveis para resolver problemas, tirar dúvidas e lhe ajudar a otimizar o desempenho da sua aplicação.



Help Desk

Atendimento através do
0800 510 9500 | +55 51 3589 9546
De segunda à sexta – das 8h às 22h
Sábados – das 8h às 17h
Indisponível em domingos e feriados



Downloads

Softwares, Documentação Técnica, Drivers, entre outros. Clique aqui e encontre as informações necessárias para você ter a melhor experiência com os produtos da Altus.



Integradores

Conheça os profissionais treinados e certificados para utilizar a tecnologia Altus. Descubra o integrador mais próximo de você e garanta a qualidade do seu projeto!



Base de conhecimento

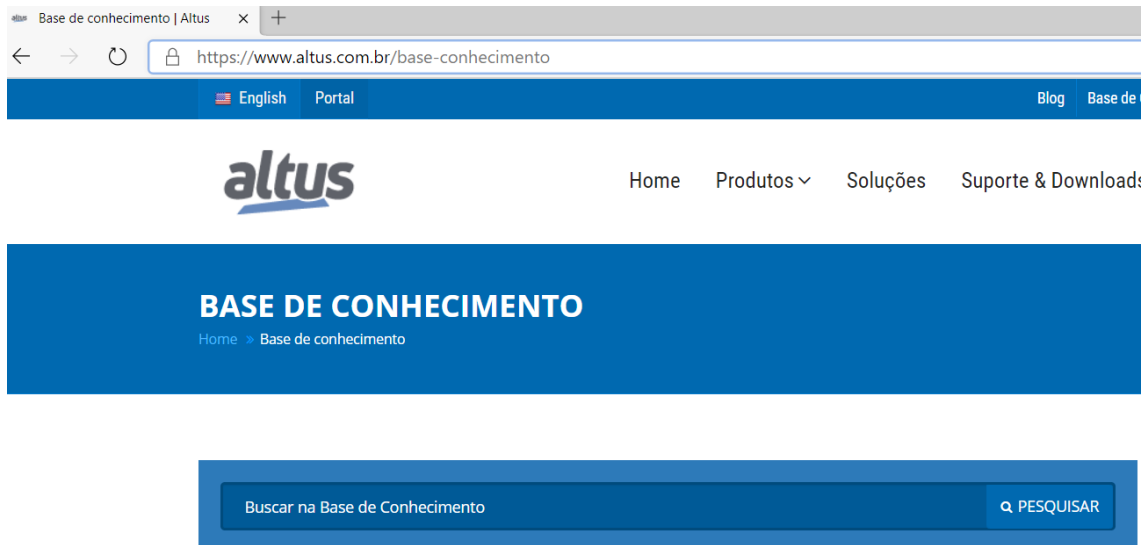
Aqui você encontra tutoriais, dicas e outros materiais de apoio criados por nossos especialistas para qualificar a sua experiência com os produtos e soluções Altus!

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 desenvolver 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	Junho/12	Emissão do Documento
B	Setembro/20	Atualização do documento

Training Box Nexto

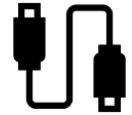


Na execução desse curso é utilizada a Maleta de Treinamento (TRAINING BOX) Nexto baseada no controlador programável de última geração Nexto Altus interligado a uma IHM da série X2 conforme mostrado na figura a seguir.



Para detalhes relacionados à funcionalidade da maleta de treinamento consulte o instrutor ou acesse o site Altus para DOWNLOAD da documentação associada.

1. Programação avançada no CP Nexto



A norma IEC61131



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.

Estrutura da IEC61131

IEC 61131-3 é o primeiro esforço real para a padronização das linguagens de programação para a automação industrial. Como este é um apelo mundial, esta é uma norma independente de qualquer empresa. Ela é a terceira parte da família IEC 61131.

A norma IEC 61131 é estruturada conforme indicado a seguir.

- Parte 1 – Informações Gerais
- Parte 2 – Requisitos de Equipamentos e Testes
- Parte 3 – Linguagens de Programação
- Parte 4 – Orientações para o Usuário
- Parte 5 – Comunicações
- Parte 7 – Programação de Controle FUZZY
- Parte 8 – Orientações para Aplicação e Implementação das Linguagens de Programação

Outra elegante forma é dividir a norma em duas partes: Elementos comuns e Linguagens de programação conforme explorado na sequência.

Elementos comuns

Conceitos básicos da IEC 61131-3

A programação baseada na norma IEC é orientada para o desenvolvimento de programas a partir da abordagem de cima para baixo (*Top-down*) e de baixo para cima (*Bottom-up*), fundamentada por três princípios básicos: modularização, estruturação e reutilização. Nesse contexto, os itens descritos a seguir devem ser considerados.

Configurações

A configuração define todos os elementos de software que interagem entre si para desempenhar as funções de controle.

Recursos

Um recurso é, basicamente, qualquer elemento com capacidade de processamento. Um recurso pode existir fisicamente ou ser apenas uma máquina virtual.

Programas

Um programa IEC pode ser constituído a partir de diferentes elementos de software cada qual escrito em qualquer uma das diferentes linguagens da norma.

Tarefas

Uma tarefa pode ser configurada para controlar a execução de programas ou blocos funcionais, de forma periódica ou disparada por eventos.

Blocos funcionais

O uso de blocos funcionais é uma forma de utilizar na programação de CPs os modernos conceitos da orientação a objetos. Eles podem ser utilizados para criação de elementos de software totalmente reutilizáveis. As principais características dos blocos funcionais são que estes possuem um conjunto de dados que podem ser alterados por um algoritmo interno. Somente o conjunto de dados é mantido na memória para uma determinada instância do bloco funcional (tais dados possuem persistência). Exemplos típicos de blocos funcionais são blocos PID, temporizadores, contadores e blocos para controle de motores.

Funções

Funções não possuem persistência, existindo apenas em tempo de execução, assim como sub-rotinas convencionais. Além da ENO, funções podem ter apenas uma saída. São exemplos de funções: blocos aritméticos, lógicos e comparadores.

Variáveis

A norma exige a declaração de variáveis dentro de diferentes elementos de software. As variáveis podem utilizar nomes (simbólicos) e ser de diferentes tipos de dados. As variáveis podem ser associadas também a posições físicas de memória (representação direta), tais como entradas, saídas e endereços internos. Variáveis globais podem ser acessadas por todos os elementos de software. Variáveis locais têm acesso restrito ao elemento de software em que estão declaradas.

Caminhos de acesso

Os caminhos de acesso permitem a transferência de dados entre diferentes configurações através de variáveis do tipo leitura ou leitura-escrita.

Unidades de Organização de Programa (POUs)

Elementos de software reutilizáveis (programas, blocos funcionais e funções) são abreviados como POU's.

Forma hierárquica

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. Isso permite a portabilidade de soluções para diferentes sistemas de controle.

Identificadores

Os identificadores servem para nominar elementos de software (variáveis, tipos de dados, POU's e tarefas). Exemplos de identificadores válidos: CURSO1, PROG_1. O padrão de caracteres adotado é o ISO646. As palavras-chave são usadas para definir diferentes construções ou para iniciar e terminar elementos de software particulares. Os identificadores não podem coincidir com as palavras-chave. Os comentários, por sua vez, devem estar entre “(” e “)”.

Tipos de dados

Diversos tipos de dados, para os quais estão definidos números de bits, faixa de valores, tamanhos e formas de representação, podem ser utilizados. Todo tipo de dado deve ser primeiramente definido para informar ao controlador como alocar espaço de memória para uma variável a ser declarada. A declaração de uma variável consiste em associar o seu nome a um tipo de dado definido. Os tipos de dados podem ser:

- Inteiros (finalizam com INT)
- Ponto flutuante (REAL)
- Tempo (TIME)
- Data e hora (DATE, TOD, DT)
- Cadeia de caracteres (STRING)
- Cadeia de bits (BOOL, BYTE, WORD e derivadas desta última)
- Genéricos (ANY)
- Derivados (novos tipos de dados que podem ser definidos – TYPE)
- Estruturas (tipos de dados compostos – STRUCT)
- Enumerações (permitem que sejam atribuídos diferentes nomes para os diferentes estados de uma variável – TYPE)
- Sub-faixas (pode-se definir uma faixa restrita de valores para variáveis do tipo inteiro)
- Matrizes (é um agrupamento de elementos do mesmo tipo de dados)

Quando um tipo de dado é criado, pode-se definir um valor inicial padrão para o tipo de dado derivado.

Variáveis

Todas as variáveis devem ser declaradas.

Variáveis locais são de uso interno na POU e são declaradas usando a palavra-chave VAR.

Variáveis de entrada de uma POU (parâmetros de entrada) são supridas por uma fonte interna, declaradas utilizando VAR_INPUT.

Variáveis de saída de uma POU fornecem valores que serão escritos em variáveis externas, declaradas utilizando VAR_OUTPUT. Estas últimas são requeridas por programas e blocos funcionais, mas não por funções.

Variáveis de entrada/saída agem tanto como parâmetros de entrada como saída, podendo ser modificadas em uma POU. São declaradas utilizando VAR_IN_OUT.

Variáveis temporárias são usadas para armazenamento de dados intermediários. São declaradas via VAR_TEMP.

Variáveis globais podem ser declaradas no nível de configuração, recurso ou programa via VAR_GLOBAL. São acessadas nas POU's por meio de uma declaração de variável externa (VAR_EXTERNAL), usando nomes idênticos.

Variáveis de representação direta permitem acessar diretamente as posições de memória do CP, tais como imagem das entradas (I), imagem das saídas (Q) e memórias internas (M). Exigem o prefixo % para identificação. A segunda letra da identificação caracteriza a subdivisão: X (bit – 1 bit), B (byte – 8 bits), W (word – 16 bits), D (double word – 32 bits) e L (long word – 64 bits).

Os atributos aceitos para as variáveis são:

- Normal – tem seu valor definido pelo valor inicial e por manipulações no programa
- Retentiva – mantém seu estado em caso de perda da alimentação do CP. Na declaração, agregam o termo RETAIN.
- Constante – não podem ser modificadas em tempo de execução, somente em tempo de programação do CP. Na declaração agregam o termo CONSTANT.
- AT – esta é a forma de associar uma variável do CP a uma posição fixa de memória.
- R_EDGE/ F_EDGE – caracterizam a detecção da borda de subida/ descida respectivamente de uma variável.

Variáveis de acesso (VAR_ACCESS): permitem referências para acesso por dispositivos remotos estabelecendo um Caminho de Acesso.

Variáveis de configuração: existem no domínio da configuração e são declaradas via VAR_CONFIG.

Linguagens de programação

A norma IEC 61131-3 define cinco linguagens de programação. Duas são textuais: Texto Estruturado (ST) e Lista de Instruções (IL). As outras três são gráficas: Sequenciamento Gráfico de Funções (SFC), Diagrama de Blocos Funcionais (FBD) e Diagrama Ladder (LD).

Estas linguagens têm sua semântica e sintaxe padronizadas, visando garantir ao programador uma interação homogênea com os programas de CP. Foram padronizadas as principais linguagens de programação utilizadas para controle industrial.

Entretanto, as diferentes linguagens da norma servem, num propósito final, como instrumentos para que o usuário possa implementar os diferentes elementos de software definidos por ela.

Nesse contexto, ressalta-se que a utilização de somente uma das linguagens no programa aplicativo não reflete o diferencial da norma, no que se refere à estruturação, modularização e reutilização dos elementos de software definidos.

A norma estimula que os ambientes de programação permitam que o usuário possa utilizar as cinco linguagens de programação para codificação das diferentes partes do problema de controle, de forma a tirar proveito do potencial de cada uma!

Aplicação da IEC 61131-3

As implicações técnicas da norma IEC 61131-3 são variadas, deixando bastante margem para crescimento e diferenciação. Isso se traduz em uma evolução contínua.

A norma IEC 61131-3 tem grande impacto em toda indústria de controle industrial. Certamente a norma não fica restrita ao mercado de CPs convencionais. Atualmente, a norma é adotada em Motion Control, sistemas distribuídos e sistemas de controle baseados em PC (SOFTLOGIC/SOFTPLC), incluindo pacotes SCADA. E as áreas de aplicação continuam crescendo.

Ter uma norma sobre uma ampla área de aplicação proporciona muitos benefícios para os usuários e programadores. Os benefícios da adoção da norma são vários, dependendo da área de aplicação. Alguns exemplos são:

- Reduzir o desperdício de recursos humanos, no treinamento, depuração, manutenção e consultoria;
- Destinar maior atenção para a solução de problemas através da reutilização de software em alto nível;
- Eliminar erros e dificuldade de entendimento;
- Utilizar as melhores práticas de programação em um ambiente mais amplo, qual seja, indústria de controle em geral;
- Combinar diferentes componentes de diferentes programas, locais, empresas e mesmo países.

O Nexto no contexto da IEC61131-3

O Nexto consiste em um poderoso e completo Controlador Programável (CP) com características únicas e inovadoras. Devido a sua flexibilidade, design inteligente, ampla capacidade de diagnósticos e arquitetura modular, os CPs da Série Nexto podem ser utilizados para aplicações de controle de sistemas de pequeno, médio e grande porte com ampla aderência à IEC 61131-3 através do ambiente de programação (PSE) denominado MasterTool IEC XE. O MasterTool IEC XE é um software programador de CP independente de dispositivo. A sua compatibilidade com a IEC 61131-3 permite suporte a todas as linguagens de programação definidas na norma.

O ambiente de programação MasterTool IEC XE

O MasterTool IEC XE é uma ferramenta completa para programação, depuração, configuração e simulação das aplicações do usuário. O software é baseado no conceito de ferramenta integrada, contudo provê flexibilidade e facilidade de uso permitindo aos usuários a programação em seis linguagens definidas pela norma IEC 61131-3: Texto Estruturado (ST), Sequenciamento Gráfico de Funções (SFC), Diagrama de Blocos Funcionais (FBD), Diagrama Ladder (LD), Lista de Instruções (IL) e Gráfico Contínuo de Funções (CFC). O MasterTool IEC XE permite o uso de diferentes linguagens na mesma aplicação, fornecendo ao usuário uma poderosa maneira de organizar a sua aplicação além de reutilizar código desenvolvido em aplicações anteriores.

Este produto oferece características para todas as etapas de desenvolvimento uma aplicação para um sistema de automação, começando por uma análise gráfica da topologia da arquitetura, passando por um poderoso ambiente de programação com suporte às linguagens da norma IEC61131-3 e 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.

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 de 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.

O MasterTool IEC XE torna o uso de interfaces para redes de campo uma prática tão simples quanto nunca visto anteriormente. O usuário não precisa um software especial para configurar a rede de campo porque o MasterTool IEC XE atende este requisito através de uma única ferramenta reduzindo tempo de desenvolvimento e simplificando a aplicação.

Com o objetivo de aumentar a produtividade do usuário, outras características importantes também estão disponíveis: Módulo de impressão que consiste de um relatório com os parâmetros específicos dos módulos e as configurações da aplicação; Impressão de lógicas que consiste de 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.

Conceitos Básicos

Considere os seguintes conceitos básicos que 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.

A organização do projeto é também determinada pela orientação ao objeto: um projeto do MasterTool IEC XE contém um programa de CP composto de vários objetos de programação e também da definição dos "recursos" necessários para executar as instâncias do programa (aplicação) nos sistemas dos dispositivos definidos (dispositivos, CPs).

Assim sendo, existem dois tipos de objetos em um projeto:

- **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 POU's. Alguns exemplos destes objetos de programação são: programas, funções, blocos funcionais, métodos, ações, definições de tipos de dados, entre outros. A instanciação é feita 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.
- **Objetos de recurso**: estes 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.

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.

Funcionalidades Avançadas

As seguir, são apresentadas as funcionalidades avançadas disponíveis no MasterTool IEC XE.

Orientação a Objetos na Programação e na Estrutura do Projeto

- Extensões para blocos funcionais: Propriedades, Métodos, Herança e Invocação de método.
- Aplicações vinculadas a dispositivos como instâncias de objetos de programação independentes.

Tipos de Dados Especiais

- ANY_TYPE
- UNION
- LTIME
- Referências
- Enumerações: tipos de dados básicos podem ser especificados
- DI: DINT := DINT#16#FFFFFFFF

Operadores e Variáveis Especiais

- Operadores de escopo: namespaces estendidos
- Ponteiros de função: substituindo o operador INSTANCE_OF
- Método Init: substituindo o operador INI
- Método Exit
- Variáveis de saída em funções e chamadas de métodos
- VAR_TEMP/VAR_STAT/VAR_RETAIN/ VAR_PERSISTENT...
- Expressões arbitrárias para inicialização de variáveis
- Atribuição como expressão
- 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

- Editor ST: recursos de edição, quebra de linha, autocompletar, monitoração e atribuição SET/RESET na linha.
- Editores FBD, LD e IL reversíveis e programáveis em um editor combinado.
- Editor IL como editor de tabela.
- Editores FBD, LD e IL: possibilidade de alteração da saída principal em caixas com várias saídas.
- Editores FBD, LD e IL sem atualização automática dos parâmetros da caixa.
- Editores FBD, LD e IL: ramificações e “redes dentro de redes”.
- Editor SFC: somente um tipo de passo, macros, seleção múltipla de elementos independentes, sem verificação sintática durante a edição e declaração automática de variáveis sinalizadoras.

Versões de Bibliotecas

- Várias versões de bibliotecas podem ser usadas no mesmo projeto utilizando o recurso de contextos.
- 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”.
- Notas de pré-compilação referentes a erros sintáticos.

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. Com a diversidade de tipos de aplicações suportadas pelo Runtime System (RTS) da Série Nexto, seguir um perfil é uma forma de reduzir a complexidade na programação. As aplicações podem ser criadas conforme um dos seguintes perfis:

- Simple;
- Básico;
- Normal;
- Experiente;
- Personalizado;
- Perfil de Máquina.

O MasterTool IEC XE pode disponibilizar inúmeros templates compatíveis para cada perfil definido para o RTS. Quando o usuário seleciona um template como modelo na criação de um projeto, a nova aplicação será desenvolvida conforme um determinado perfil, adotando as regras, características e padrões definidos pelo perfil associado ao template.

Cada perfil de projeto define nomes padronizados para as tarefas e programas, os quais são pré-criados pelos TEMPLATES de projeto. O desenvolvedor é obrigado a seguir rigorosamente a nomenclatura para tarefas, mas pode seguir ou não os nomes sugeridos para os respectivos programas.

Para garantir a compatibilidade de um projeto a um determinado perfil ao longo do desenvolvimento, são utilizadas duas abordagens:

1. O MasterTool IEC XE somente permite a criação de projetos baseados em um TEMPLATE, selecionando ao mesmo tempo o perfil a ser utilizado.
2. Na geração de código, o MasterTool IEC XE realiza a verificação de todas as regras definidas para o perfil válido para o projeto.

Os parágrafos a seguir detalham as características ou padrões de cada perfil de projeto que seguem uma escala gradual de complexidade. Com base nestas definições, recomenda-se que o usuário sempre procure utilizar o perfil mais simples que atenda às necessidades da sua aplicação, migrando para outro mais sofisticado apenas quando as regras correspondentes estiverem sendo mais entraves ao desenvolvimento do que simplificações didáticas. Cabe ressaltar que a ferramenta de programação permite a alteração do perfil de um projeto existente, mas caberá ao desenvolvedor realizar qualquer ajuste necessário para que o projeto se torne compatível com as regras do novo perfil selecionado.

Simples

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

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

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

Básico

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

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

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

Normal

No perfil de projeto Normal, a aplicação possui uma tarefa de usuário do tipo Cíclica denominada MainTask. Essa tarefa é responsável pela execução de uma única unidade de programação do tipo Programa denominada MainPrg. Este programa e esta tarefa são equivalentes a única tarefa e único programa do perfil Simples, mas aqui a aplicação pode integrar tarefas adicionais de usuário. Estas outras tarefas são denominadas CyclicTask00 e CyclicTask01, cada qual

responsável pela execução exclusiva do respectivo programa `CyclicPrg<nn>`. As tarefas `CyclicTask<nn>` são sempre do tipo cíclica (Cyclic) com prioridade fixada como 13 (treze), prioridade idêntica a tarefa `MainTask`. Estes dois tipos formam um conjunto denominado de tarefas básicas, cujos programas associados podem chamar outras POUs do tipo Programa, Função ou Bloco Funcional.

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

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

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

Experiente

O perfil de projeto Experiente inclui as mesmas tarefas básicas, `MainTask`, `CyclicTask<nn>`, `ExternInterruptTask00` e `TimeInterruptTask00`, com as mesmas prioridades (13, 02 e 01 respectivamente), mas é uma expansão dos anteriores, pois admite múltiplas tarefas de evento. Ou seja, a aplicação pode incluir várias tarefas `ExternInterruptTask<nn>` ou `TimeInterruptTask<nn>` executando os programas `ExternInterruptPrg<nn>` e `TimeInterruptPrg<nn>`. As prioridades das tarefas de evento adicionais podem ser livremente selecionadas na faixa de 08 a 12. Neste perfil, além dos programas padrões, cada tarefa pode executar programas adicionais.

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

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

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

Personalizado

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

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

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

Perfil de Máquina

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

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

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

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



Pesquisa na Documentação: perfis de projeto

Consulte, na documentação do produto, as características das tarefas associadas a cada um dos perfis analisados anteriormente. Estas características incluem POU, Prioridade, Tipo, Intervalo ou Evento.

Fonte: Manual de Utilização UCPs Série Nexto NX30xx – MU214100

Projeto

Um projeto contém os objetos das POU's 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 POU's podem ser gerenciados na janela de visualização das POU's ou na janela de visualização dos dispositivos, POU's 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.

Dispositivos

Cada "dispositivo" representa um hardware específico (destino). Exemplos: controlador, módulos de E/S, monitor.


Cada dispositivo é definido por uma descrição e deve ser instalado no sistema local para que possa ser inserido na árvore de Dispositivos. 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 POU's - serem gerenciados SOMENTE na árvore de Dispositivos e, neste caso, estarem disponíveis apenas na sua aplicação ou nas suas "aplicações secundárias".

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 POU's. 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.



NOTAS:

Em alguns sistemas são realizadas algumas multitarefas preferenciais. Neste caso, as seguintes observações devem ser consideradas.

- Todas as tarefas compartilham o mesmo mapa de processo, pois um mapa para cada tarefa prejudicaria a performance. Entretanto, o mapa do processo sempre consiste em apenas uma tarefa. Assim, ao criar um projeto, o usuário deve explicitamente observar que, em caso de conflitos, os dados de entrada serão copiados para a área de salvamento (o mesmo ocorre com as saídas). Por exemplo, módulos da biblioteca "SysSem" podem ser usados para resolver problemas de sincronização.

- Ao acessar outros objetos globais (variáveis globais, módulos), podem ocorrer problemas de consistência, se o tamanho dos objetos excederem a capacidade do processador (estruturas ou ARRAYS formando uma unidade lógica). Neste caso, os módulos da biblioteca “SysSem” também podem ser usados para resolver os problemas.

Geração de Código e Informações de Compilação

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.

Alterações Online

Dependendo da alteração no projeto em execução no controlador, apenas os objetos modificados serão carregados para ele.

ATENÇÃO:

Alterações online modificam o programa aplicativo em execução e não provocam a reinicialização do mesmo. Certifique-se de que o novo código da aplicação não afetará o comportamento esperado do sistema. Dependendo do controlador, podem ocorrer danos nas máquinas ou nas suas partes, assim como pode haver risco à saúde e à vida das pessoas.



NOTAS:

- Quando é realizada uma alteração online, as inicializações específicas da aplicação não serão executadas, pois a máquina mantém o seu estado. Por esta razão, o novo código de programa pode não funcionar conforme o desejado.
- Variáveis de ponteiro mantêm os valores do ciclo anterior. Se houver um ponteiro em uma variável que tenha alterado seu tamanho devido a uma alteração online, o valor não estará mais correto. Certifique-se de que as variáveis de ponteiro sejam reatribuídas a cada ciclo.

Aplicação de Inicialização (Projeto de Inicialização)

Uma aplicação de inicialização é o projeto que será iniciado automaticamente quando o controlador é inicializado. Para tanto, o projeto deve estar disponível no CP em um arquivo “<nome do projeto>.app”. Este arquivo pode ser criado no modo offline através do comando *Criar Aplicação de Inicialização* (menu *Comunicação*).

A cada envio bem sucedido, a aplicação ativa será automaticamente armazenada no arquivo “<aplicação>.app” na pasta do sistema do dispositivo, ficando assim disponível como uma aplicação de inicialização. O comando *Criar Aplicação de Inicialização* também permite salvar esta aplicação em um arquivo no modo offline.

Pesquisa na Documentação: métodos de envio e login de projetos

Consulte, na documentação do produto, os métodos de envio e login de projetos sem diferença nos mesmos. O objetivo aqui é garantir que o usuário não terá problemas ao enviar projetos iguais e fazer login em UCPs em execução a partir de diferentes estações e para fazer login em UCPs em execução sem gerar alterações de projeto a partir de diferentes estações.



Fonte: Manual de Programação IEC 61131 – MP399048

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.



Pesquisa na Documentação: configuração de Breakpoints

Consulte, na documentação do produto, as configurações típicas associadas à funcionalidade de Breakpoints.

Fonte: Manual de Utilização MasterTool IEC XE – MU299048

Unidade de Organização de Programa (POU)

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

POUs gerenciadas na Visualização das POU's não são específicas do dispositivo, mas devem ser instanciadas para uso em um dispositivo (aplicação). Para tanto, POU's de programa devem ser chamadas por uma tarefa da respectiva aplicação.

As POU's inseridas na árvore de Dispositivos explicitamente em uma aplicação SOMENTE são gerenciadas na Visualização dos Dispositivos, ou seja, podem ser instanciadas apenas por aplicações recuadas abaixo desta aplicação (aplicação secundária).

No entanto, POU também é o nome de uma determinada subcategoria destes objetos no menu Acrescentar Objeto, que aí compreende programas, blocos funcionais e funções.

Assim sendo, uma POU - Unidade de Organização de Programa, em geral, é uma unidade de programação, um objeto gerenciado através de dispositivos não específicos (na janela das POU's) e específicos (na janela dos Dispositivos) e que pode ser visualizado e editado em uma janela do editor.

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

Considere a possibilidade de configurar determinadas propriedades (como por exemplo, condições de compilação etc.) da POU. Por padrão, os seguintes tipos de POU podem ser usados:

- POU propriamente dita;
- Ação;
- DUT (Unidade de Tipo de Dado);
- Arquivo Externo;
- Lista de Variáveis Globais;
- Método;
- Propriedades;
- Programa;
- Função;
- Bloco Funcional;
- Variáveis Persistentes;
- POU's para Verificações Implícitas;

POU

Uma POU é uma Unidade de Organização de Programa do tipo Programa, Função ou Bloco Funcional. A figura a seguir ilustra a criação de uma POU do tipo Bloco Funcional.

Figura 1-1. Diálogo Acrescentar POU

NOTAS:

- Dependendo do tipo, a POU pode ser complementada por métodos, propriedades, ações e transições. Para tal, também é utilizado o comando Acrescentar Objeto.
- A ordem hierárquica de processamento das instâncias da POU de uma aplicação depende da configuração específica do dispositivo (pilha de chamadas).
- Cada POU consiste de uma parte de declaração e uma parte de implementação. O corpo é escrito em uma das linguagens de programação disponíveis, que são IL, ST, SFC, FBD, LD ou CFC.
- O MasterTool IEC XE suporta todas as POUs descritas pela norma IEC 61131-3. Para usar estas POUs no projeto, deve-se incluir a biblioteca standard.library. Os projetos criados a partir do modelo Projeto MasterTool Padrão já possuem esta biblioteca carregada.

- POU's podem chamar outras POU's, mas não são permitidas recorrências.
- Quando uma POU atribuída a uma aplicação chama outra POU apenas pelo seu nome (sem nenhum contexto adicionado), é adotada a seguinte ordem de busca: aplicação, gerenciador de bibliotecas da aplicação, visualização das POU's e gerenciador de bibliotecas na visualização das POU's.

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.

Abaixo do nome, seguem as declarações de variáveis de entrada, saída e variáveis de programa. Opcionalmente, também podem ser declaradas variáveis de acesso. A figura a seguir mostra um exemplo de programa.

```

1  PROGRAM PRGexample
2  VAR_INPUT
3      in_var:INT;
4  END_VAR
5  VAR_OUTPUT
6      out_var:INT;
7  END_VAR
8  VAR
9      ivar:INT;
10     bvar:BOOL;
11 END_VAR

1  out_var:=in_var+ivar;
2  IF out_var:=23
3      THEN bvar:=TRUE;
4  END_IF;
    
```

Figura 1-2. Exemplo de Programa

Chamada de Programas

Exemplo de chamada de programa em IL:

```

CAL      PRGexample      (
         in_var:= 33      )
LD      PRGexample.out_var
ST      erg
    
```

Atribuindo os parâmetros (assistente de entrada Com argumentos):

```

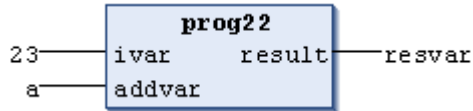
CAL      PRGexample      (
         in_var:= 33      ,
         out_var=> erg    )
    
```

Exemplo em ST:

```
PRGEXAMPLE();
ERG := PRGEXAMPLE.OUT_VAR;
```

Atribuindo os parâmetros (assistente de entrada Com argumentos):
 PRGEXAMPLE (IN_VAR:=33, OUT_VAR=>ERG);

Em FBD:



Função

Uma função é uma POU que produz exatamente um elemento de dados (que pode consistir de 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.

Via comando *Acrescentar Objeto* e *Acrescentar POU*, pode-se adicionar uma função ao projeto. Para atribuí-la a uma aplicação existente, selecione-a na visualização dos *Dispositivos* e use o comando do menu de contexto. Caso contrário, ela será adicionada à visualização das *POUs*. No diálogo *Acrescentar POU*, escolha o tipo *Função*, digite um nome (<nome da função>) e o tipo de dados de retorno (<tipo de dados>) para a nova função e escolha a linguagem de implementação desejada. O editor estará disponível via botão *Abrir*, permitindo, na sequência, a edição da POU.

Declaração:

A declaração da função inicia com a palavra-chave **FUNCTION**. Devem ser definidos um nome e um tipo de dados.

Sintaxe:

```
FUNCTION <NOME DA FUNÇÃO>:<TIPO DE DADOS>
```

Esta sintaxe é seguida pelas declarações das variáveis de entrada e das variáveis da função.

Um resultado deve ser atribuído a uma função, ou seja, o nome da função é usado como variável de saída.

NOTA:

Se uma variável local for declarada como retentiva em uma função, isto não terá efeito. A variável não será incluída na área de retenção.

Exemplo de uma Função em ST:

```
FUNCTION FCT:INT
VAR_INPUT
IVAR1:INT;
IVAR2:INT;
IVAR3:INT;
END_VAR
FCT:=IVAR1+IVAR2*IVAR3;
```

Esta função lê três variáveis de entrada e retorna o produto das duas últimas, adicionadas à primeira.

Chamada de Função

A chamada de uma função em ST pode aparecer como um operando em expressões.

Em IL, uma chamada de função somente pode estar posicionada dentro das ações de um passo ou dentro de uma transição.

Funções (ao contrário dos programas ou blocos funcionais) 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.

Exemplos para chamadas de função.

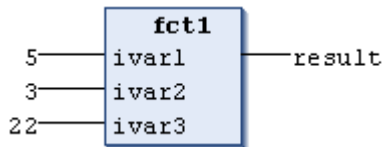
Em IL:

LD	5	
Fct	3	r
	22	
ST	result	

Em ST:

RESULT := FCT1(5, 3, 22);

Em FBD:



Em chamadas de função, não é possível misturar atribuições de parâmetros explícitos com implícitos. Isto permite alterar a ordem das atribuições dos parâmetros de entrada.

Exemplo:

FUN(FORMAL1 := ACTUAL1, ACTUAL2); // -> Mensagem de erro.
 FUN(FORMAL2 := ACTUAL2, FORMAL1 := ACTUAL1); // Mesma semântica que:
 FUN(FORMAL1 := ACTUAL1, FORMAL2 := ACTUAL2);

De acordo com a norma IEC 61131-3, as funções podem ter saídas adicionais. Estas saídas devem ser atribuídas na chamada de uma função, por exemplo, em ST, conforme a seguinte sintaxe:

OUT1 => <VARIÁVEL DE SAÍDA 1> | OUT2 => <VARIÁVEL DE SAÍDA 2> |... OUTRAS VARIÁVEIS DE SAÍDA

Exemplo:

A função FUN é definida com duas variáveis de entrada in1 e in2. O valor de retorno de FUN será escrito nas variáveis de saída (VAR_OUTPUT) loc1 e loc2 declaradas localmente.

FUN(IN1 := 1, IN2 := 2, OUT1 => LOC1, OUT2 => LOC2);

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 das variáveis internas necessárias 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.

Além da funcionalidade descrita na norma IEC 61131-3, a programação orientada a objetos é suportada e os blocos funcionais podem ser definidos como extensões de outros blocos funcionais. Isto significa que a herança pode ser usada ao programar com blocos funcionais.

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

Via comando Acrescentar Objeto e Acrescentar POU, um bloco funcional pode ser adicionado ao projeto. Para atribuí-lo a uma aplicação existente, selecione-o na visualização dos dispositivos e use o comando do menu de contexto. Caso contrário, ele será adicionado à visualização das POU's.

No diálogo Acrescentar POU, escolha o tipo Bloco Funcional, digite o nome do bloco (<identificador>) e escolha a linguagem de implementação.

Opções adicionais:

- Estender: digite o nome de outro bloco funcional disponível no projeto, que deve ser a base do bloco atual (corrente);
- Implementar: não é suportado.

O editor estará disponível via botão Abrir, permitindo, na sequência, a edição da POU.

Sintaxe de declaração:

```
FUNCTION_BLOCK <NOME DO BLOCO FUNCIONAL> | EXTENDS <NOME DO BLOCO FUNCIONAL>
```

A sintaxe acima é seguida pela declaração das variáveis.

Exemplo:

O bloco funcional FBexample, mostrado na figura a seguir, apresenta duas variáveis de entrada e duas de saída, out1 e out2. Out1 é a soma das duas entradas, out2 é o resultado de uma comparação de igualdade.

Exemplo de um Bloco Funcional em ST:

```
FUNCTION_BLOCK FBEXAMPLE
VAR_INPUT
INP1:INT;
INP2:INT;
END_VAR
VAR_OUTPUT
OUT1:INT;
OUT2:BOOL;
END_VAR
OUT1:=INP1+INP2;
OUT2:= INP1=INP2;
```

Instância de Bloco Funcional

Blocos funcionais são sempre chamados através de uma instância, a qual é uma reprodução (cópia) de um 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.

Sintaxe para declaração de uma instância de bloco funcional:

```
<identificador>:<nome do bloco FUNCIONAL>;
```

Exemplo:

Declaração (por exemplo, na parte de declaração de um programa) da instância INSTANCE do bloco funcional FUB:

```
INSTANCE: FUB;
```

As partes de declaração dos blocos funcionais e programas podem conter declarações de instâncias, no entanto, nas funções, as declarações de instâncias não são permitidas.

Chamando um Bloco Funcional

Instâncias de blocos funcionais devem ser declaradas local ou globalmente. A variável do bloco funcional pode ser acessada usando-se a sintaxe abaixo.

<NOME DA INSTÂNCIA>.<NOME DA VARIÁVEL>

NOTAS:

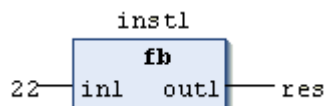
- Somente as variáveis de entrada e saída de um bloco funcional podem ser acessadas externamente a uma instância do bloco funcional. Suas variáveis internas não são acessíveis.
- O acesso a uma instância do bloco funcional é limitado à POU na qual ela foi declarada, a menos que tenha sido declarada globalmente.
- Na chamada da instância, os valores desejados podem ser atribuídos aos parâmetros dos blocos funcionais.
- As variáveis de entrada e saída (VAR_IN_OUT) de um bloco funcional são passadas como ponteiros.
- No SFC, as chamadas de blocos funcionais somente podem ocorrer em ações associadas aos passos.
- O nome da instância do bloco funcional pode ser usado como um parâmetro de entrada para uma função ou outro bloco funcional.
- Todos os valores de um bloco funcional ficam retidos até o próximo processamento deste, portanto, as chamadas de blocos funcionais nem sempre retornam os mesmos valores de saída, mesmo que tenham sido utilizados os mesmos argumentos.
- Se no mínimo uma das variáveis do bloco funcional for retentiva, a instância total é armazenada na área de dados retentivos.

Exemplos de acessos a variáveis de blocos funcionais: suponha que o bloco funcional fb tem uma variável de entrada in1 do tipo INT. Veja a chamada desta variável a partir do programa prog.

Declaração e implementação em ST:

```
PROGRAM PROG
VAR
INST1:FB;
RES:INT;
END_VAR
INST1.IN1:=22; (* fb é chamado e a variável in1 assume o valor 22 *)
INST1(); (* fb é chamado, isto é necessário para o acesso à variável de saída mostrado a seguir *)
RES:=FBINST.OUTL; (* Variável de saída do fb é lida *)
```

Exemplo em FBD:



Atribuindo Parâmetros na Chamada

Nas linguagens textuais IL e ST, os parâmetros de entrada/saída podem ser configurados imediatamente na chamada de um bloco funcional. Os valores podem ser atribuídos aos

parâmetros entre parênteses após o nome da instância do bloco funcional. Para parâmetros de entrada, esta atribuição ocorre usando “:=”, assim como ocorre com a inicialização das variáveis na declaração. Para parâmetros de saída, deve ser usado “=>” .

Veja um exemplo de chamada com atribuições, onde um bloco funcional de temporização (instância CMD_TMR) é chamado com atribuições para os parâmetros IN e PT. A variável resultante Q é atribuída à variável A com o nome da instância do bloco funcional, seguido de um ponto e do nome da variável:

```
CMD_TMR(IN := %IX5.0, PT := 300);  
A:=CMD_TMR.Q
```

Se a instância é inserida via *Assistente de Entrada* (<F2>), através da opção *Inserir com Argumentos* na janela de implementação de uma POU ST ou IL, ela será automaticamente exibida de acordo com a sintaxe mostrada a seguir, com todos os seus parâmetros, mas não necessariamente deve atribuir estes parâmetros. Para o exemplo acima, a chamada deve aparecer da seguinte forma:

```
CMD_TMR(in:=, pt:=, q=>) (* Nenhum parâmetro atribuído *)  
CMD_TMR(in:=bvar, pt:=t#200ms, q=>bres); (* Parâmetros atribuídos *)
```

Estudo Dirigido 1-1: declaração de blocos funcionais



Tomando como base o Estudo de Caso proposto no capítulo 5 declare um bloco funcional chamado Motor_PD_S que permite a operação remota pelo supervisor a partir da reutilização do bloco funcional Motor_PD desenvolvido no Curso MP10 (Básico). A área de interfaceamento da POU com o sistema de supervisão é estabelecida de forma implícita na declaração das variáveis internas do bloco funcional. Essas variáveis são acessadas via rede. O bloco funcional inclui o RESET de determinados comandos provenientes do supervisor utilizando um bloco de atribuição após o processamento do código principal.



DICA: consulte o Instrutor para orientações adicionais.

ANOTAÇÕES

ANOTAÇÕES

Resolução do Estudo Dirigido 1-1:

Etapa 1: Declaração do tipo de bloco funcional Motor_PD_S

FUNCTION_BLOCK Motor_PD_S						
	Scope	Name	Address	Data type	Initialization	Comment
1	VAR_INPUT	Liga_L		BOOL	FALSE	Liga local
2	VAR_INPUT	Desliga_L		BOOL	FALSE	Desliga local
3	VAR_INPUT	Funcionando		BOOL	FALSE	Funcionando
4	VAR_INPUT	Sobrecarga,Defeito		BOOL	TRUE	Sensores de falha
5	VAR_INPUT	MAN_auto		BOOL	FALSE	Modo manual (TRUE) e automático (FALSE)
6	VAR_INPUT	Liberacao		BOOL	FALSE	Liberação em automático
7	VAR_OUTPUT	Ligar		BOOL	FALSE	Comando ligar
8	VAR_OUTPUT	Modo_LR		BOOL	FALSE	Modo remoto (TRUE) e local (FALSE)
9	VAR_OUTPUT	Falha		BOOL	FALSE	Falha
10	VAR	Liga_S		BOOL	FALSE	Liga remoto (supervisório)
11	VAR	Desliga_S		BOOL	FALSE	Desliga remoto (supervisório)
12	VAR	Modo_LR_S		BOOL	FALSE	Modo (supervisório)
13	VAR	Pulso		F_TRIG		Pulso seleção local/remoto
14	VAR	M_PD		Motor_PD		Instância motor partida direta

Etapa 2: Corpo do bloco funcional

A partir da declaração proposta anteriormente para o bloco funcional em questão implemente o corpo do mesmo.

Extensão de um Bloco Funcional

A programação orientada ao objeto permite a derivação de um bloco funcional a partir de outro. Desta forma, as propriedades de um bloco funcional podem ser estendidas a outro.

A extensão é feita usando-se a palavra-chave EXTENDS na declaração do bloco funcional. É possível escolher a opção EXTENDS ao acrescentar o bloco ao projeto através do diálogo *Acréscimo de Objeto*. Sintaxe:

```
FUNCTION_BLOCK <nome do bloco funcional> EXTENDS <nome do bloco funcional>
```

A declaração das variáveis é então realizada na sequência. Exemplo de definição do bloco funcional fbA:

```
FUNCTION_BLOCK fbA
VAR_INPUT
x:int;
...
```

Definição do bloco funcional fbB:

```
FUNCTION_BLOCK fbB EXTENDS fbA
VAR_INPUT
ivar:int;
...
```

No exemplo acima:

- FbB contém todos os dados e métodos definidos por fbA. Uma instância de fbB pode agora ser usada em qualquer contexto onde um bloco funcional do tipo fbA é esperado.
- FbB pode substituir os métodos definidos em fbA. Isto significa que fbB pode declarar um método com o mesmo nome e as mesmas entradas e saídas declaradas em fbA.
- Não é permitido que fbB use as variáveis do bloco funcional com o mesmo nome usado em fbA. Neste caso, o compilador apontará um erro.
- Variáveis e métodos de fbA podem ser acessados diretamente dentro do escopo fbB através da palavra-chave SUPER (SUPER^.<método>).

**NOTA:**

Não são permitidas heranças múltiplas. Ou seja, não é possível estender mais do que um bloco funcional.

Invocação de Método

A programação orientada ao objeto, quando aplicada em blocos funcionais, além da opção ESTENDER, permite o uso de herança. Ela requer invocações dinâmicas de método, conhecidas como chamadas de função virtual.

Chamadas de função virtual precisam de mais tempo que as chamadas de função normais e são usadas quando:

- Uma chamada é realizada através de um ponteiro para um bloco funcional (pfub^.method);
- Um método chama outro método do mesmo bloco funcional.

Chamadas de função virtual possibilitam que a mesma chamada em um código fonte do programa invoque diferentes métodos durante a execução.

De acordo com a norma IEC 61131-3, métodos como funções normais podem ter saídas adicionais. Estas devem ser atribuídas na chamada do método, conforme a seguinte sintaxe:

<MÉTODO>(IN1:=<VALOR> |, ATRIBUIÇÕES DE ENTRADA ADICIONAIS, OUT1 => <VARIÁVEL DE SAÍDA 1> | OUT2 => <VARIÁVEL DE SAÍDA 2> | ...VARIÁVEIS DE SAÍDA ADICIONAIS)

Isto faz com que a saída do método seja escrita nas variáveis de saída declaradas localmente, conforme definido na chamada. Exemplo: Suponha que os blocos funcionais fub1 e fub2 ESTENDEM o bloco funcional fubbase. O método method1 está incluído. Possível uso das chamadas de método:

```
VAR_INPUT
B : BOOL;
END_VAR
VAR
PINST : POINTER TO FUBBASE;
INSTBASE : FUBBASE;
INST1 : FUB1;
INST2 : FUB2;
END_VAR
IF B THEN
PINST := ADR(INSTBASE);
ELSE
PINST := ADR(INST1);
END_IF
PINST^.METHOD1(); (* Se B é TRUE, FUBBASE.METHOD1 é chamado, senão FUB1.METHOD1 é chamado *)
```

Agora, suponha que fubbase do exemplo acima contém dois métodos, method1 e method2. fub1 inclui method2 mas não method1. method1 é chamado, conforme mostrado no exemplo acima, como segue:

```
PINST^.METHOD1(); (*Se B é TRUE, FUBBASE.METHOD1 é chamado, senão FUB1.METHOD1 é chamado*)
```

Chamada via ponteiros - Implementação de method1:

```
METHOD METHOD1 : BOOL
VAR_INPUT
END_VAR
METHOD1 := METHOD2();
```


Ao chamarmos fubbase.method1 via ponteiros devemos considerar o que segue:

- Se o ponteiro for do tipo fubbase, fubbase.method2 será chamado;
- Se o ponteiro for do tipo fub1, fub1.method2 será chamado.

Informações detalhadas sobre a funcionalidade “Método” podem ser conferidas na sequência desse tutorial.

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
```

Estudo Dirigido 1-2: declaração de unidades de tipos de dados



Tomando como base o Estudo de Caso proposto no capítulo 5 declare uma Unidade de Tipo de Dado (DUT) contendo os dados de receita do produto final. Os dados da receita devem incluir: disponibilidade e estado de atualização de dados, taxa e proporção de matéria-prima, velocidade de passagem e temperatura do forno, massa desejada para o produto final, quantidade de rejeito e código do produto.




DICA: consulte o Instrutor para orientações adicionais.


```


1  TYPE Dados_Receita :
2  STRUCT
3      Disponivel:BOOL := FALSE; (* Estado de disponibilidade de dados, TRUE = Disponível *)
4      Atualiza:BOOL := FALSE; (* Estado de atualização de dados TRUE = Atualiza *)
5      Taxa_Mat_A:REAL := 0.0; (* Taxa para a matéria prima "A" *)
6      Proporcao_AB:REAL := 1.0; (* Proporção entre as matérias-primas "A" e "B" *)
7      Velocidade:REAL := 0.0; (* Velocidade de passagem pelo forno *)
8      Temperatura:REAL := 0.0; (* Temperatura do forno para o produto final *)
9      Quantidade:REAL := 0.0; (* Quantidade em [kg] para o produto final *)
10     Rejeito:REAL := 0.0; (* Quantidade do rejeito em [kg] *)
11     Codigo:STRING; (* Código do produto *)
12 END_STRUCT
13 END_TYPE

```

Método

Assim como na programação orientada a objetos, os *Métodos* () 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.

Inserindo Métodos

Para atribuir um método a um bloco funcional, selecione-o na árvore das *POUs* ou *Dispositivos*. No menu de contexto, use o comando *Acrescentar Objeto/Método*. No diálogo *Acrescentar Método*, digite o nome, o tipo de retorno e a linguagem de implementação desejados. Os tipos de dados de retorno podem ser visualizados usando o botão  para abrir o diálogo do *Assistente de Entrada*. O editor do método estará disponível via botão *Abrir*. A declaração segue a sintaxe indicada a seguir:

```

METHOD <NOME DO MÉTODO> : <TIPO DE DADO>
VAR_INPUT
X: INT;
END_VAR

```

Chamadas de Método

Chamadas de método são também denominadas chamadas de função virtual conforme estudado anteriormente no tópico blocos funcionais.



NOTAS:

- Todos dados de um método são temporários e somente são válidos durante a sua execução (variáveis de pilha). No corpo de um método são permitidas variáveis e o acesso à instância do bloco funcional.
- Ponteiro THIS: o identificador THIS é usado para apontar diretamente para a instância implícita do bloco funcional disponível. Observe que, em uma variável declarada localmente, pode estar escondida uma variável do bloco funcional. Então, na descrição da sintaxe do exemplo mostrado acima, THIS^.x não se refere à entrada x do método, mas sim à variável x do bloco funcional. As variáveis VAR_IN_OUT ou VAR_TEMP do bloco funcional não podem ser acessadas em um método.
- Métodos, assim como funções podem ter saídas adicionais. Estas podem ser atribuídas durante a invocação do método.

Métodos Especiais para um Bloco Funcional

- Método Init: um método FB_Init pode ser declarado implícita ou explicitamente. Ele contém o código de inicialização para o bloco funcional, conforme declarado na sua parte de declaração.
- Método Reinit: se o método denominado FB_Reinit for declarado para uma instância de um bloco funcional, ele será chamado após esta ter sido copiada e reinicializará o novo módulo da instância.
- Método Exit: se o usuário deseja um método de saída (FB_Exit), por exemplo, para desalocação, deve declará-lo explicitamente. Não há declaração implícita. O método de saída será chamado para cada instância do bloco funcional antes de um novo envio, reset ou durante uma alteração online para todas as instâncias movidas ou apagadas.
- Propriedades: para os métodos Set e Get (consulte o tópico “Propriedade” – na sequência – para informações adicionais).

Chamada de Método (Aplicação Interrompida)


No arquivo de descrição do dispositivo, pode-se definir que um determinado método seja sempre chamado ciclicamente por uma determinada instância do bloco funcional (ou de um módulo de biblioteca). Se este método contiver os parâmetros de entrada abaixo, ele será processado também quando a aplicação corrente ativa não estiver em execução:

```
VAR_INPUT
PTASKINFO : POINTER TO DWORD;
PAPPLICATIONINFO: POINTER TO _IMPLICIT_APPLICATION_INFO;
END_VAR
```

O programador pode, então, verificar o status da aplicação via pApplicationInfo e definir o que deve acontecer. Exemplo:

```
IF PAPPLICATIONINFO^.STATE=RUNNING THEN <INSTRUÇÕES> END_IF
```

Propriedade

Uma *Propriedade*  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;.

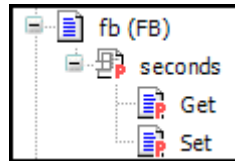


Figura 1-3. Propriedade Segundos Acrescentada ao Bloco Funcional fb

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

Monitorando uma Propriedade

As propriedades podem ser monitoradas no modo online com a ajuda da *Monitoração Online* ou da *Lista de Monitoração*. Para monitorar uma propriedade, é necessário adicionar o pragma {attribute 'monitoring':='variable'} no topo da sua definição.

Ação

Ações (A) podem ser definidas e atribuídas a blocos funcionais e programas através do comando *Acrescentar 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 contêm 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 POUs e, por isso, não contém declarações próprias.

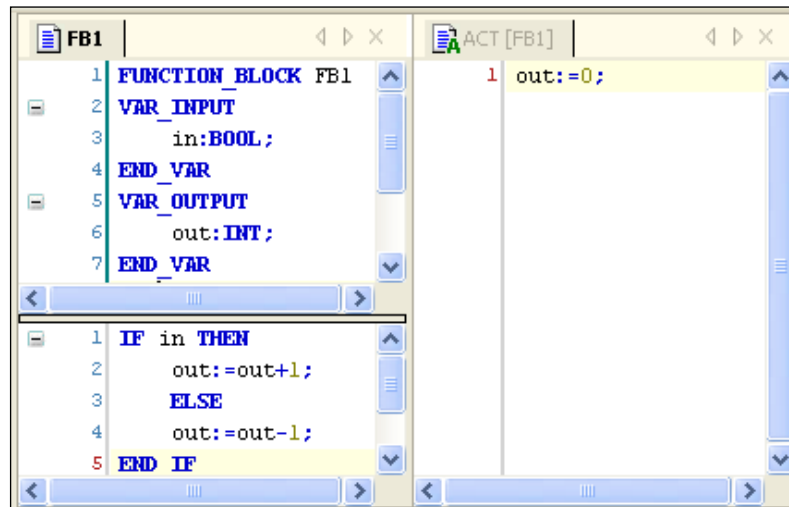


Figura 1-4. Ação em um Bloco Funcional

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 *Acrescentar Objeto/Ação*, quando o respectivo programa ou bloco funcional for selecionado na árvore dos Dispositivos ou das POUs. No diálogo *Acrescentar Ação*, defina o nome da ação e a linguagem de implementação desejada.

Chamando uma Ação

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 MAINPRG
VAR
  INST : COUNTER;
END_VAR
```

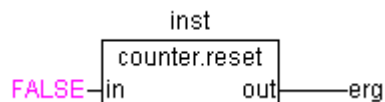
Chamada da ação Reset em outra POU programada em IL:

```
CAL INST.RESET(IN := FALSE)
LD INST.OUT
ST ERG
```

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:



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 POU, as variáveis estarão disponíveis para todo o projeto. Se uma GVL estiver atribuída a uma determinada aplicação, as variáveis serão válidas dentro desta aplicação.

O editor GVL é usado para editar uma Lista de Variáveis Globais.

Se o dispositivo suportar a funcionalidade de rede, as variáveis contidas em uma GVL podem ser configuradas para estarem disponíveis como variáveis de rede, isto é, para uma troca de dados via broadcast com outros dispositivos na rede. Para tanto, as propriedades de rede apropriadas devem estar configuradas para a GVL.

Considere que as variáveis declaradas nas GVLs sempre são inicializadas antes das variáveis locais das POU.

Variáveis Persistentes

O objeto \mathbb{T} representa uma lista de variáveis globais contendo apenas as variáveis persistentes da aplicação, e pode ser atribuído a uma aplicação através da inserção do mesmo na árvore de dispositivos via comando *Acrescentar Objeto*.

Variáveis persistentes são reinicializadas somente em um *Reset Origem* <aplicação>.


A lista de variáveis persistentes é editada no editor GVL e VAR_GLOBAL PERSISTENT RETAIN aparece na primeira linha.

Arquivo Externo

Qualquer arquivo externo (com qualquer extensão) pode ser adicionado ao projeto na janela das *POUs* através do comando *Acréscentar Objeto*, opção *Arquivo Externo*. O diálogo *Acréscentar Arquivo Externo* será aberto.

POUs para Verificações Implícitas

Caso o usuário deseje utilizar a funcionalidade de verificação de ARRAYS, limites de intervalo, divisões por zero e ponteiros durante a execução, POU's especiais poderão ser adicionadas à aplicação. Esta funcionalidade está implícita.


Para tanto, o menu *Acréscentar Objeto* na categoria *POUs para Verificações Implícitas*  apresenta as seguintes funções: *CheckBounds*, *CheckDivDInt*, *CheckDivLInt*, *CheckDivReal*, *CheckDivLReal*, *CheckRangeSigned*, *CheckRangeUnsigned* e *CheckPointer*.

Após ter sido inserida uma POU de verificação, ela será aberta no editor correspondente na linguagem de implementação selecionada. No editor ST, está disponível uma implementação padrão adaptável aos requisitos do usuário.

A opção de inserção de uma determinada POU de verificação só está disponível uma vez. Se todos os tipos de POU de verificação já tiverem sido adicionados na aplicação, o diálogo *Acréscentar Objeto* não mais apresentará a categoria *POUs para Verificações Implícitas*.

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 POU's e variáveis definidas diretamente no projeto. A extensão padrão para um arquivo de biblioteca é ".library".

O gerenciamento das bibliotecas em um projeto é feito no *Library Manager* , cuja instalação no sistema é feita previamente no diálogo do *Repositório de Bibliotecas*.

Pesquisa na Documentação: gerenciamento de bibliotecas

Consulte, na documentação do produto, as configurações típicas associadas à funcionalidade de gerenciamento de bibliotecas, incluindo instalação e inclusão no projeto, bibliotecas referenciadas, versões de biblioteca, acesso único aos módulos da biblioteca ou às variáveis e criação, codificação e documentação de bibliotecas.

Fonte: Manual de Programação IEC 61131 – MP399048



Linguagens de Programação IEC61131-3

Além das linguagens de programação tradicionais, tais como Diagrama LADDER (LD), Diagrama de Blocos Funcionais (FBD) e Lista de Instruções (IL), 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.

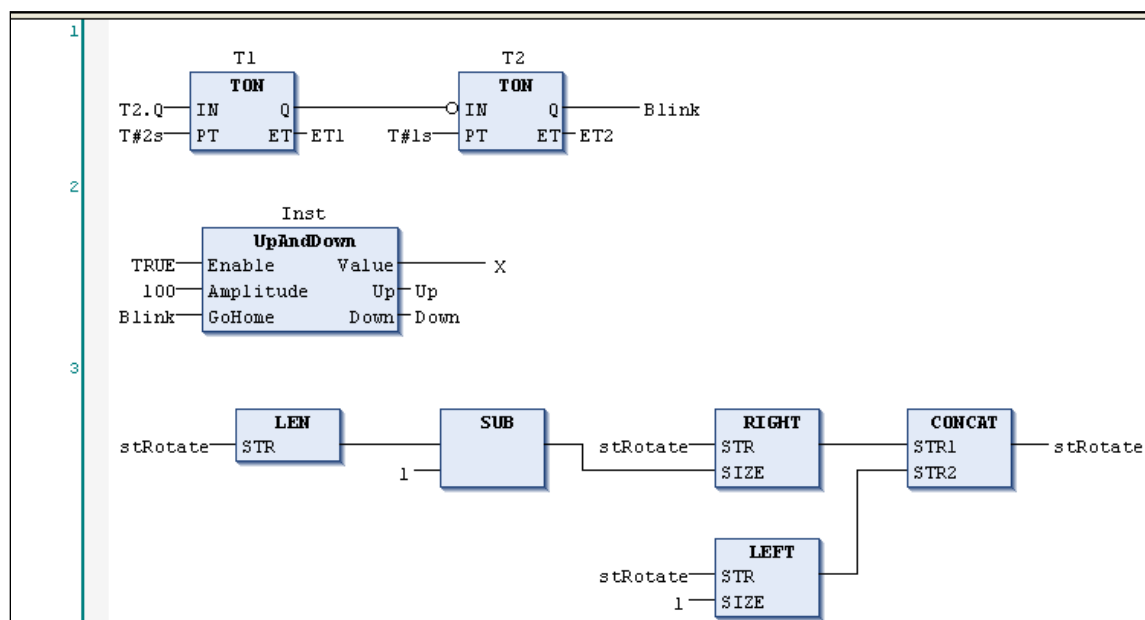


Figura 1-5. Redes de Diagrama de Blocos Funcionais

Lista de Instruções (IL)

A Lista de Instruções é similar à linguagem de programação Assembly. Esta linguagem suporta programações baseadas em um acumulador. Todos operadores IEC 61131-3 são permitidos, assim como as várias entradas/saídas, negações, comentários, saídas set/reset e saltos incondicionais/condicionais. Cada instrução, fundamentalmente, é baseada no carregamento de valores no acumulador através do uso do operador LD. Em seguida, a operação é executada com o primeiro parâmetro retirado do acumulador. O resultado da operação novamente é disponibilizado no acumulador, a partir de onde o usuário deve armazená-lo com a instrução ST.

Para programar execuções condicionais ou laços, a IL suporta tanto operadores de comparação (EQ, GT, LT, GE, LE, NE), quanto saltos. Estes últimos podem ser incondicionais (JMP) ou condicionais (JMPC/JMPCN). Para saltos condicionais, o valor do acumulador é verificado em TRUE ou FALSE.

Cada instrução inicia em uma nova linha e contém um operador. Dependendo do tipo da operação, ela pode conter também um ou mais operandos separados por vírgulas. O operador deve ser estendido por um modificador.

Na linha anterior à instrução, pode haver uma marca de identificação (rótulo) seguida de dois pontos, por exemplo “m1:” no exemplo mostrado abaixo. Um rótulo pode ser o destino de uma instrução de salto, por exemplo “JMPC next” no exemplo mostrado abaixo. Um comentário deve ser posicionado como o último elemento de uma linha. Linhas vazias podem ser inseridas entre instruções.

```

1  PROGRAM IL
2  VAR
3      inst1: TON;
4      dwVar: DWORD;
5      dwRes: DWORD;
6      t1: TIME;
7      tout1: TIME;
8      inst2: TON;
9      bVar: BOOL;
10 END_VAR

```

```

1
LD      bVar                                variable
ST      inst1.IN                            starts timer with risin...
JMPC    m1
CAL     inst1(
        PT:=t1,
        ET:=>tout1)
LD      inst1.Q                              is TRUE, PT seconds aft...
ST      inst2.IN                            starts timer with risin...

2

m1:
LD      dwVar
ADD     230
ST      dwRes

```

Figura 1-6. Exemplo de Programa IL no Editor Tabular IL

O editor IL é um editor tabular integrado ao editor FBD/LD/IL.



Pesquisa na Documentação: editor FBD/LD/IL

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

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. Ao contrário da IL (Lista de 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



Estudo Dirigido 1-3: declaração do bloco funcional Lim_Taxa

Utilizando o Editor ST implemente um bloco funcional para a limitação da taxa de variação de um sinal analógico chamado Lim_Taxa. Ele será reutilizado no Estudo de Caso do Capítulo 5.



DICA: consulte o Instrutor para orientações adicionais.

ANOTAÇÕES

Implementação do Estudo Dirigido 1-3:

Etapa 1: Declaração do tipo de bloco funcional Lim_Taxa

```

1  FUNCTION_BLOCK Lim_Taxa
2  VAR_INPUT
3      Habilita:BOOL; (* Habilita a limitação de taxa *)
4      XIn:REAL; (* Entrada *)
5      X0:REAL; (* Valor inicial *)
6      Fim_ESC:REAL; (* Fim de escala positivo *)
7      Taxa:REAL; (* Taxa/segundo - 0 a 100% *)
8      Ciclo:TIME; (* Tempo de ciclo *)
9      Reset:BOOL; (* Reseta saída *)
10 END_VAR
11 VAR_OUTPUT
12     Variando:BOOL; (* Em rampa = TRUE *)
13     XOUT:REAL; (* Valor de saída *)
14 END_VAR
15 VAR
16     XIn_Corr:REAL; (* Entrada corrigida *)
17     X_Ult:REAL; (* Último valor da saída *)
18     X_Alvo:REAL; (* Valor alvo *)
19     Taxa_Atual:REAL; (* Taxa atual *)
20     Taxa_Calc:REAL; (* Taxa calculada *)
21 END_VAR

```

Etapa 2: Corpo do bloco funcional

```

1  Taxa_Calc:=(Taxa/100.0)*Fim_Esc*TIME_TO_REAL(Ciclo)/1000.0;
2  IF (ABS(XIn)/Fim_Esc) > 1.0 THEN
3      XIn_Corr := XIn / ABS(XIn) * Fim_Esc;
4  ELSE XIn_Corr := XIn;
5  END_IF
6  IF Reset=TRUE OR Taxa_Calc<0.0 OR Taxa_Calc>100.0 THEN
7      XOUT := X0;
8      Variando := FALSE;
9  ELSE
10     IF Habilita=TRUE THEN X_Alvo := XIn_Corr;
11     ELSE X_Alvo := X0;
12     END_IF
13     Taxa_Atual := (X_Alvo - X_Ult) / Fim_Esc * 100.0;
14     IF Taxa_Calc < ABS(Taxa_Atual) THEN
15         XOUT := X_Ult + Taxa_Calc * Taxa_Atual / ABS(Taxa_Atual);
16         Variando := TRUE;
17     ELSE XOUT := X_Alvo;
18         Variando:= FALSE;
19     END_IF
20 END_IF
21 X_Ult := XOUT;

```

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.

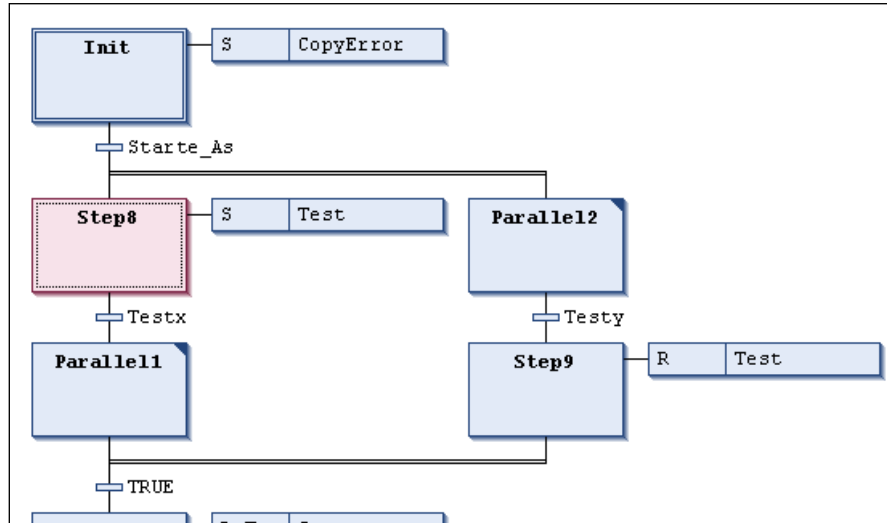


Figura 1-7. Exemplo para uma Sequência de Passos em um Módulo SFC

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





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 (item 1).

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, de forma 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.

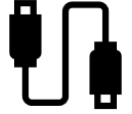
Requisitos para Ambientes de Programação (PSE)

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.

2. Comunicação avançada no CP Nexto



Neste capítulo são exploradas as funcionalidades de comunicação em rede do controlador programável, enfatizando os protocolos MODBUS, OPC e PROFIBUS.

Configuração das Interfaces Seriais

As interfaces seriais COM1 e COM2, são configuradas na árvore de dispositivos abaixo da UCP conforme mostrado na figura a seguir

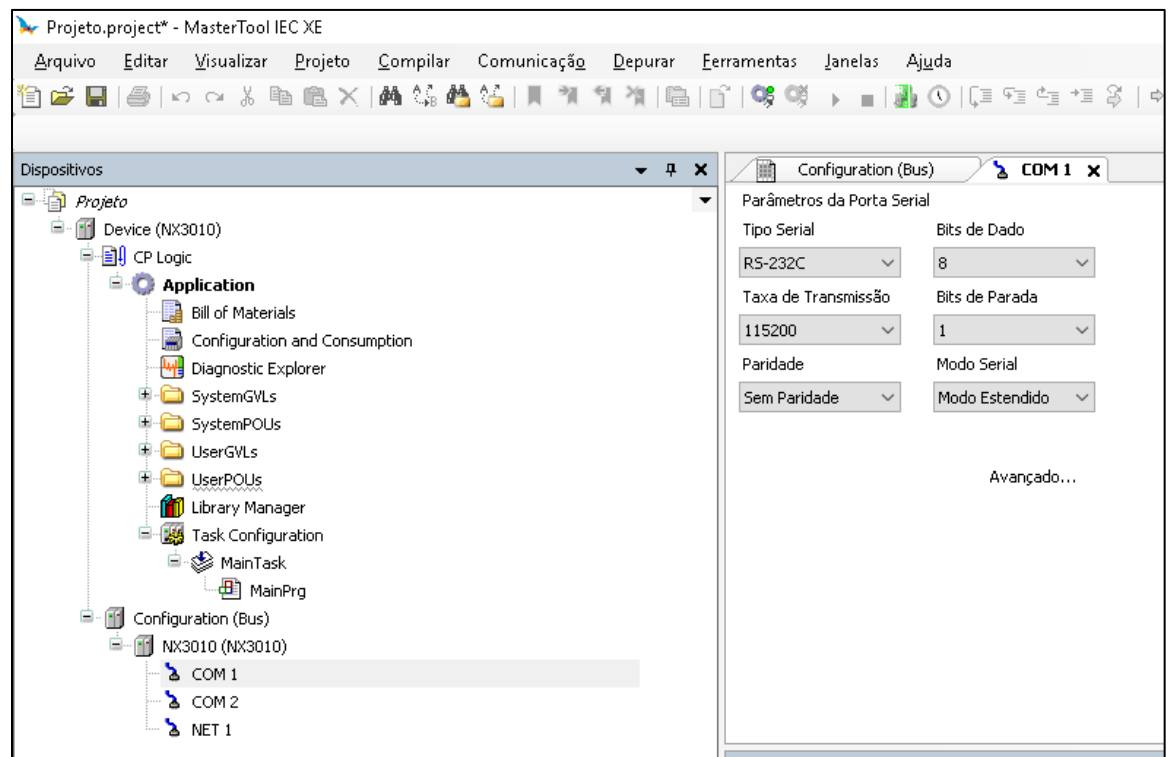


Figura 2-1. Interfaces seriais

A interface de comunicação COM 1, presente nas UCPs NX3010, NX3020 e NX3030, é composta por um conector tipo DB9 fêmea para o padrão RS-232C. Permite a comunicação ponto a ponto (ou em rede utilizando conversor) nos protocolos abertos MODBUS RTU.

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

Além disso, a interface de comunicação COM 2, também presente nas UCPs NX3010, NX3020 e NX3030, dependendo da configuração do cabo usado, permite escolher o tipo de interface física: RS-422 ou RS-485. Por outro lado, as UCPs NX3003, NX3004 e NX3005 não

têm a interface serial COM 2, permitindo somente a comunicação por meio da interface física RS-485 (NX3003) e RS-422 ou RS-485 (dependendo do cabo selecionado) no caso das UCPs NX3004 e NX3005.



Pesquisa na Documentação: configuração das interfaces seriais

Consulte, na documentação do produto, as configurações típicas associadas às interfaces seriais COM 1 / 2 versus modelos de UCPs. Alguns pontos importantes para considerar: tipo de conector e interface, sinais de modem, taxas de transmissão, protocolos suportados etc.

Fonte: Manual de Utilização UCPs Série Nexto NX30xx – MU214100

Protocolo MODBUS na Série Nexto

Como vimos anteriormente, o padrão MODBUS define um Protocolo de mensagens na camada de aplicação, posicionado no 7º nível do Modelo de Referência ISO/OSI que provê comunicação “cliente/servidor” entre dispositivos conectados a diferentes tipos de topologias de rede. Este padrão também especifica um protocolo de comunicação serial para requisições MODBUS entre um mestre e um ou vários escravos. Vamos analisar agora a implementação deste protocolo na plataforma Nexto inicialmente descrevendo a inserção de instâncias do protocolo e, na sequência, explorando os editores MODBUS.



Um pouco de teoria... O protocolo MODBUS

O protocolo Modbus foi desenvolvido originalmente para comunicar um dispositivo mestre com outros dispositivos escravos. Embora seja utilizado normalmente sobre conexões seriais (Ponto a Ponto com RS-232 ou Barramento Multiponto com RS-485), ele também pode ser usado como um protocolo da camada de aplicação de redes industriais tais como TCP/IP sobre Ethernet e MAP. Este é talvez o protocolo de mais larga utilização em automação industrial, pela sua simplicidade e facilidade de implementação. A figura a seguir mostra uma hierarquia Modbus típica.

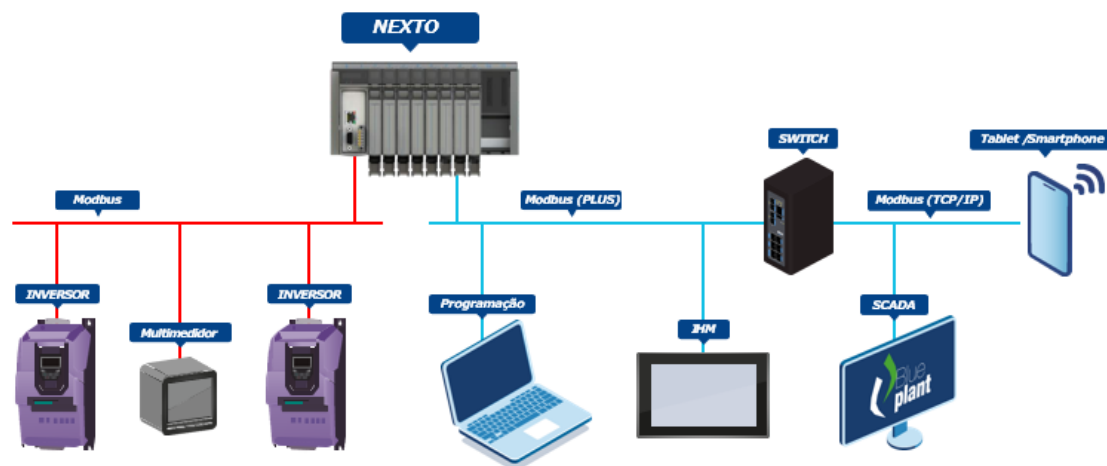


Figura 2-2. Hierarquia Modbus

O MODBUS TCP/IP é usado para comunicação entre sistemas de supervisão e controladores lógicos programáveis. O protocolo Modbus é encapsulado no protocolo TCP/IP e transmitido através de redes padrão Ethernet com controle de acesso ao meio por CSMA/CD.

O MODBUS PLUS é usado para comunicação entre si de controladores lógicos programáveis, módulos de E/S, chaves de partida eletrônica de motores, interfaces homem máquina etc. O meio físico é o RS-485 com taxas de transmissão de 1 Mbps, controle de acesso ao meio por HDLC (High Level Data Link Control).

O MODBUS PADRÃO é usado para comunicação dos CPs com os dispositivos de entrada e saída de dados, instrumentos eletrônicos inteligentes (IEDs) como relés de proteção, controladores de processo, atuadores de válvulas, transdutores de energia etc. O meio físico é o RS-232 ou RS-485 em conjunto com o protocolo mestre-escravo.

Do ponto de vista do modelo de camadas OSI o protocolo Modbus pode ser enquadrado conforme mostrado na figura a seguir.

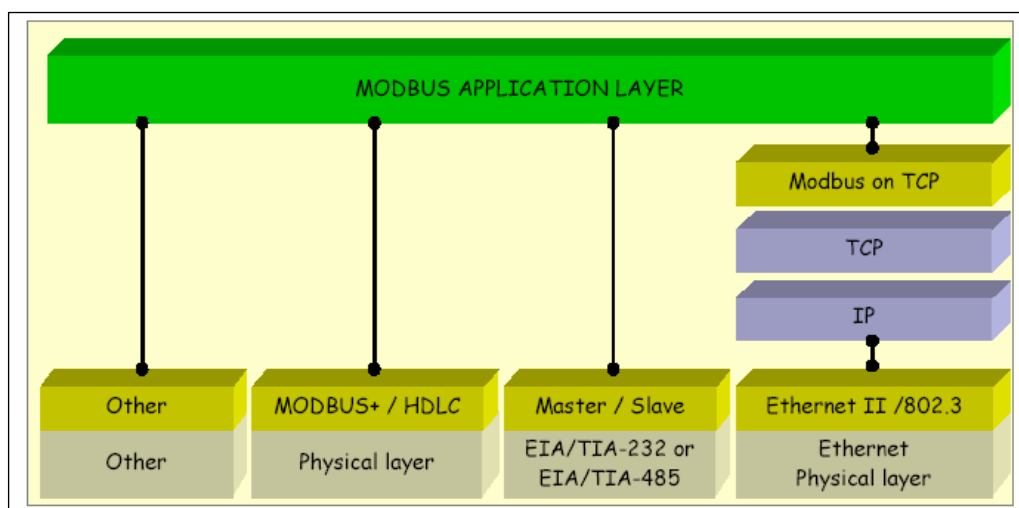


Figura 2-3. Protocolo Modbus

O protocolo Modbus é baseado em um modelo de comunicação mestre-escravo, onde um único dispositivo, o mestre, pode iniciar transações denominadas *queries*. Os demais dispositivos da rede (escravos) respondem, suprimindo os dados requisitados pelo mestre ou executando uma ação por ele comandada.

O mestre, por exemplo, pode ser um sistema supervisor e os escravos, controladores programáveis. Os papéis de mestre e escravo são fixos, quando se utiliza comunicação serial, mas em outros tipos de rede, um dispositivo pode assumir ambos os papéis, embora não simultaneamente. A figura a seguir ilustra uma comunicação mestre-escravo.

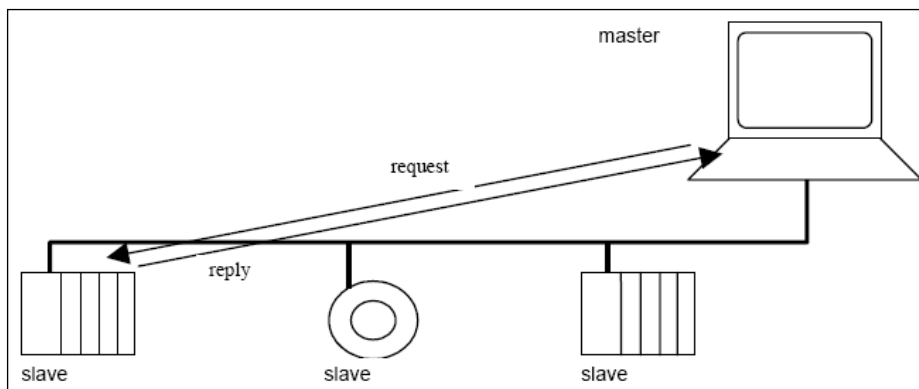


Figura 2-4. Comunicação Mestre-Escravo

Existem dois modos de transmissão: ASCII (American Code for Information Interchange) e RTU (Remote Terminal Unit), que são selecionados durante a configuração dos parâmetros de comunicação. No modo ASCII cada byte de mensagem é enviado como dois caracteres ASCII. Durante a transmissão, intervalos de até um segundo entre caracteres são permitidos, sem que a mensagem seja truncada. Algumas implementações fazem uso de tais intervalos de silêncio como delimitadores de fim de mensagem. No modo RTU cada byte de mensagem é enviado como um byte de dados. A mensagem deve ser transmitida de maneira contínua, já que pausas maiores que 1,5 caractere provoca truncamento da mesma. A figura a seguir mostra uma transmissão MODBUS típica.

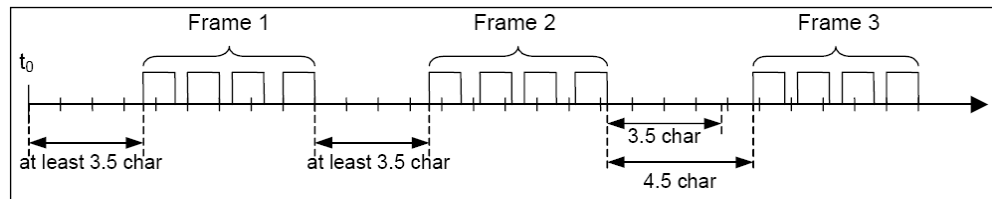


Figura 2-5. Transmissão Modbus

No modo RTU (o mais adotado no meio industrial) a mensagem tem o formato indicado na figura a seguir.

Start	Endereço	Função	Dados	CRC	END
Silêncio 3..5 chars	← 8 bits →	← 8 bits →	← N x 8 bits →	← 16 bits →	Silêncio 3..5 chars

Figura 2-6. Formato da Mensagem

A faixa de endereços válidos vai de 0 a 247 (0x00 a 0xf7 hexadecimal), sendo que os dispositivos recebem endereços de 1 a 247. O endereço zero é reservado para *broadcast*, ou seja, mensagens com esse valor de endereço são reconhecidas por todos os elementos da rede. Quando há outros protocolos de rede abaixo do Modbus, o endereço é normalmente convertido para o formato utilizado por eles. Nesse caso, pode ser que o mecanismo de *broadcast* não seja suportado. Quando o mestre envia uma mensagem para os escravos, este campo contém o endereço do escravo. Quando o escravo responde, coloca seu próprio endereço neste campo.

O código de função varia de 1 a 255 (0x01 a 0xff), mas apenas a faixa de um a 127 (0x01 a 0x7f) é utilizada, já que o *bit* mais significativo é reservado para indicar respostas de exceção. Normalmente, uma resposta inclui o código de função da requisição que lhe deu origem. No entanto, em caso de falha, o *bit* mais significativo do código é ativado para indicar que o conteúdo do campo de dados não é a resposta esperada, mas sim um código de diagnóstico.

O tamanho e conteúdo do campo de dados variam com a função e o papel da mensagem, requisição ou resposta, podendo mesmo ser um campo vazio. O campo de frame “check sequence” contém um valor de oito ou dezesseis bits, dependendo do modo de transmissão serial, que é utilizado para detecção de erros na mensagem. Quando eventuais camadas de rede subjacentes provêm mecanismos de detecção de erros próprios, o FCS é dispensado. O LRC corresponde ao complemento de 2 da soma de todos os bytes da mensagem excluídos o byte de start e end. O CRC é uma variante do CRC16.

Os únicos identificadores através dos quais o dispositivo mestre pode reconhecer a resposta para uma determinada mensagem são o endereço do dispositivo escravo e a função solicitada. Assim, o envio de múltiplas requisições, em que tais parâmetros coincidam, deve ser feito ordenadamente, isto é, cada mensagem só deve ser enviada, depois que a resposta para a mensagem anterior for recebida. Não há problema em se enviar simultaneamente comandos iguais para dispositivos diferentes ou comandos diferentes para um mesmo dispositivo, embora nesse último caso possam surgir problemas dependendo do equipamento específico.

As principais funções do MODBUS são:

- Read Coil Status: leitura de estados das saídas discretas do dispositivo escravo, endereços na forma 0nnnn;
- Read Input Status: leitura de estados das entradas discretas do dispositivo escravo, endereços na forma 1nnnn;
- Read Holding Registers: leitura de valores dos holdings registers do dispositivo escravo, endereços na forma 4nnnn;
- Read Input Registers: leitura de valores dos input registers do dispositivo escravo, endereços na forma 3nnnn;

- Force Single Coil: escrita de um valor de saída discreta, endereço 0nnnn, no dispositivo escravo. Esse valor permanece constante enquanto não for alterado por uma nova operação de escrita ou pela programação interna do dispositivo;
- Preset Single Register: escrita de um valor de holding register, endereço 4nnnn. Assim como acontece para as saídas, o valor no registrador permanece constante enquanto não for alterado por operações de escrita ou pelo próprio dispositivo.

Há dois mecanismos para detecção de erros no protocolo Modbus serial: *bits* de paridade em cada caractere e o *frame check sequence* ao final da mensagem. A verificação de paridade é opcional em ambos os modos de transmissão, ASCII e RTU. Um *bit* extra é adicionado a cada caractere de modo que ele contenha um número par ou ímpar de *bits* ativos, caso sejam adotadas respectivamente paridade par ou ímpar. A principal desvantagem desse método é sua incapacidade de detectar números pares de inversões de *bit*. Caso não seja utilizado, o *bit* de paridade é substituído por um *stop bit* adicional.

Fonte: modbus.org.

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.

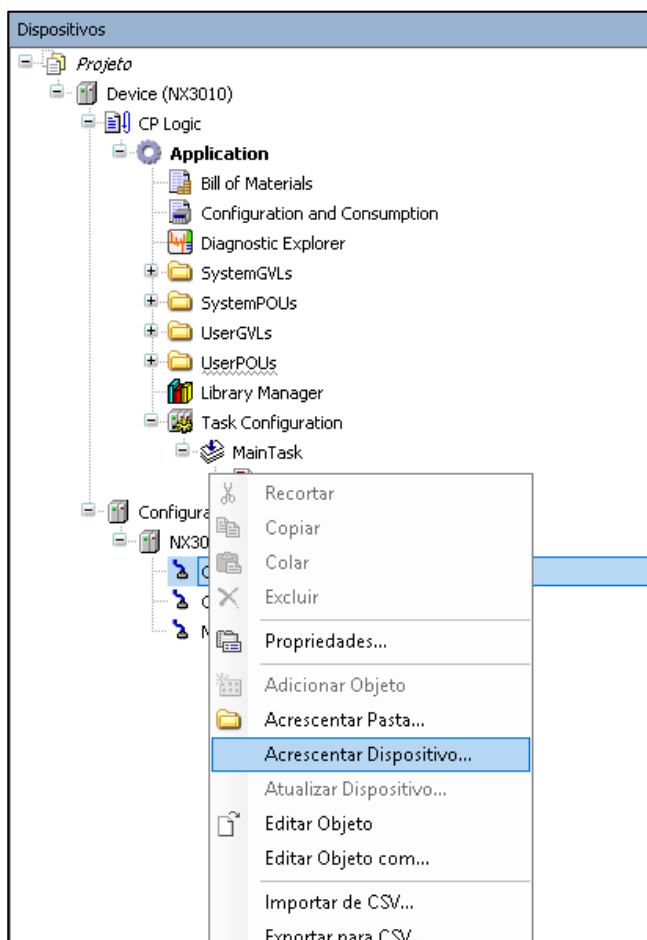


Figura 2-7. Adicionando a Instância MODBUS

Após, surgirão na tela os protocolos disponíveis ao usuário. Nesse caso, pode-se selecionar o MODBUS RTU Slave e clicar em Acrescentar, conforme mostrado a seguir.

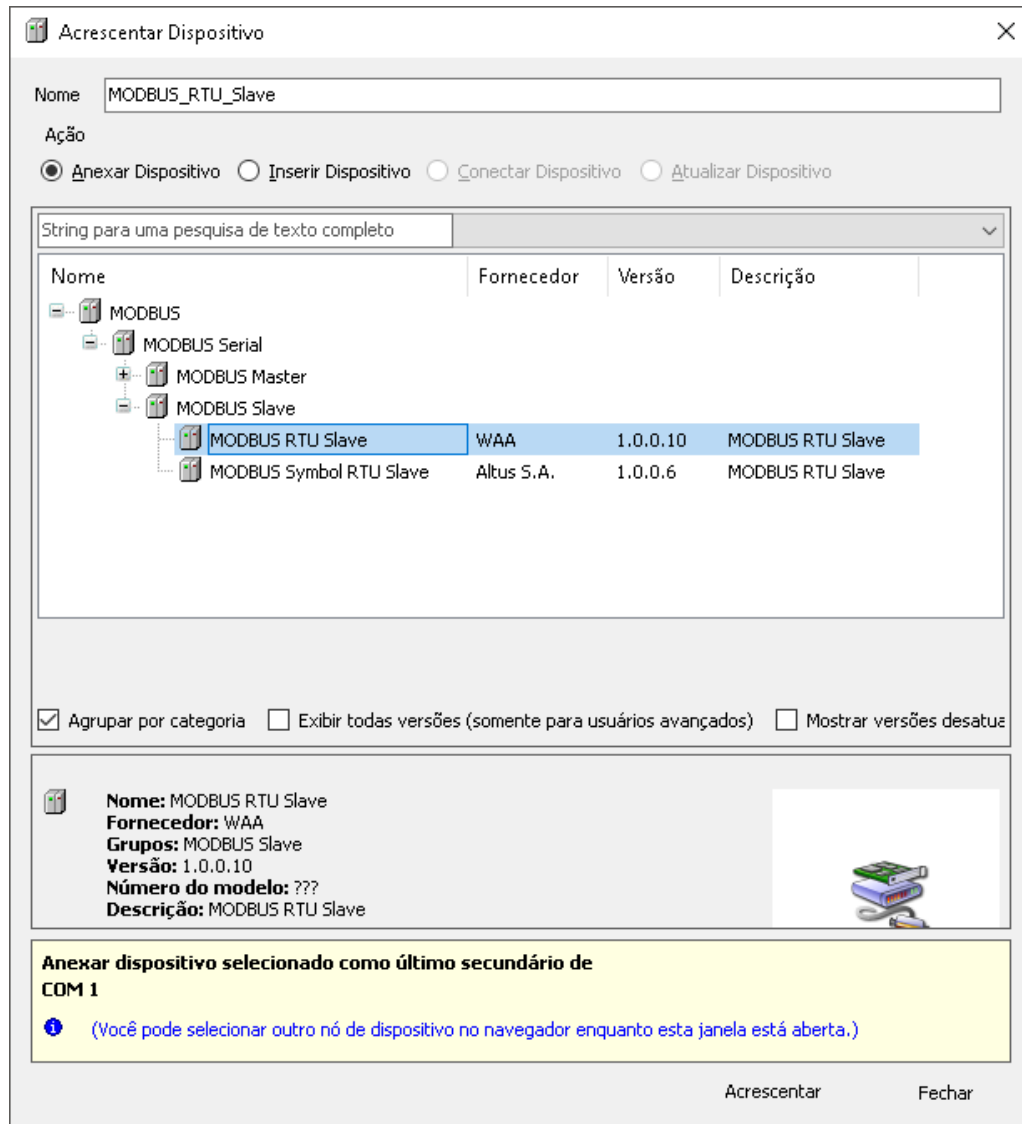


Figura 2-8. Seleção do protocolo MODBUS RTU Slave

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. A seguir, clicar em Acrescentar, conforme mostrado na sequência.

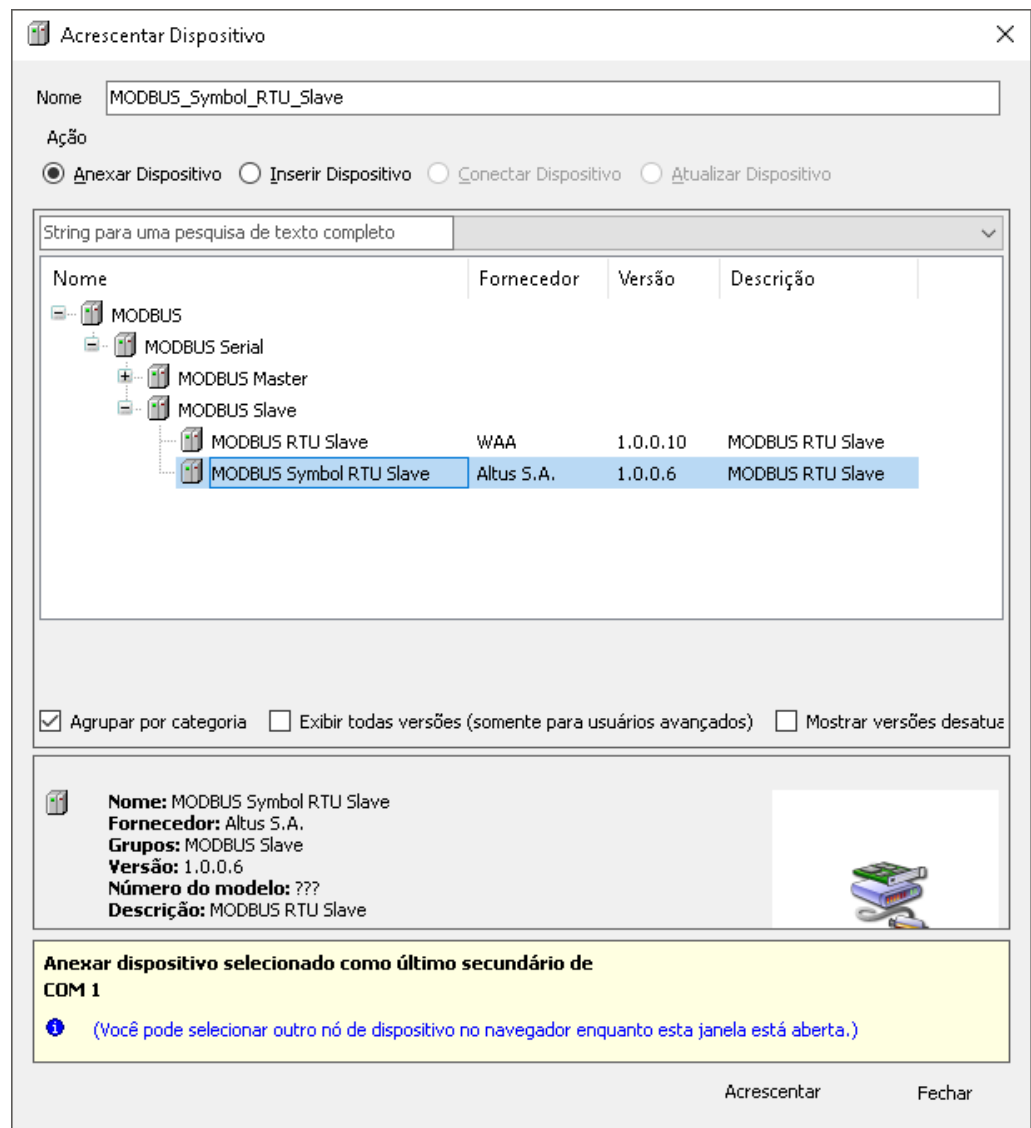


Figura 2-9. Seleção do protocolo MODBUS RTU Slave

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 (essa tela está ilustrada na figura inicial do capítulo).

Configuração do Protocolo MODBUS Mestre por Mapeamento Simbólico

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.

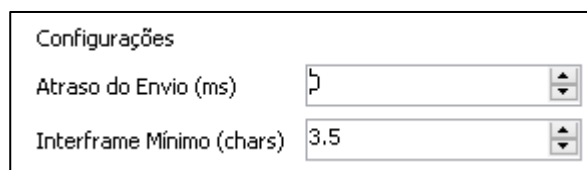


Figura 2-10. Tela de Configuração Parâmetros Gerais MODBUS RTU Mestre

- 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.

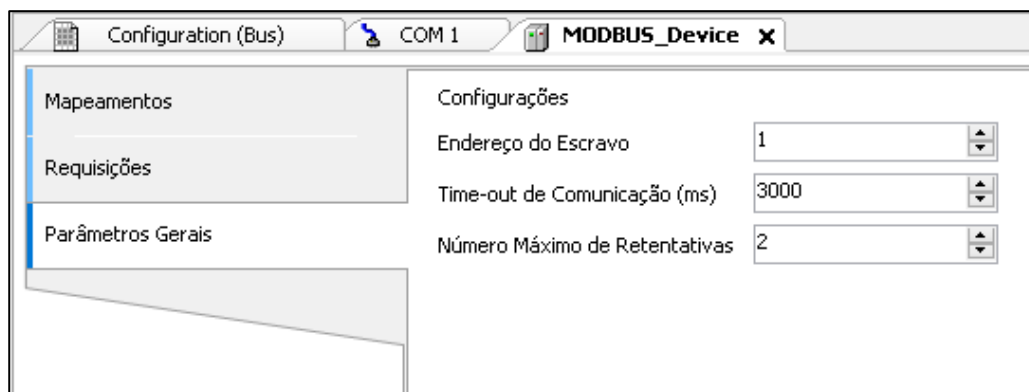


Figura 2-11. Tela de Configurações dos Parâmetros Gerais do Dispositivo

- 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.

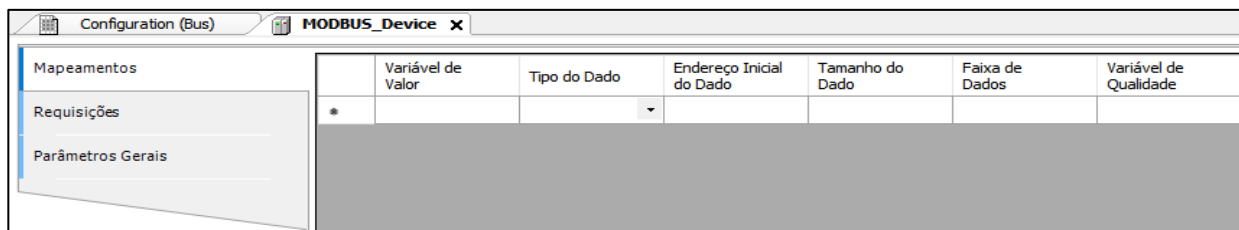


Figura 2-12. Tela de Mapeamentos de Dados MODBUS

- 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.

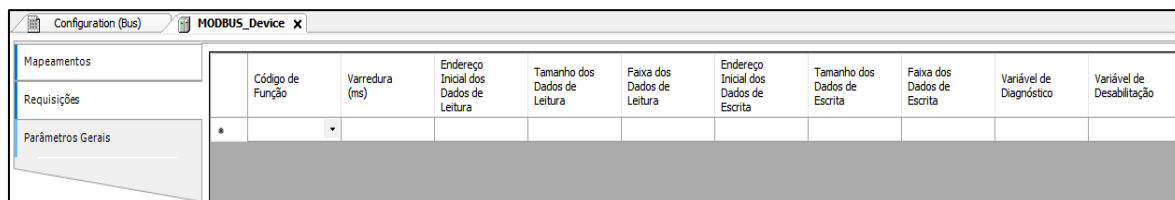


Figura 2-13. Tela de Requisições de dados MODBUS Mestre

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

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

- Configurar os parâmetros gerais do protocolo MODBUS, como: tempos de comunicação e variáveis de representação direta (%Q) para receber os diagnósticos. A figura a seguir ilustra essa configuração.

Configurações do Mestre

Endereço Inicial de Diagnósticos em %Q
21606

Tamanho
20 Avançado...

Faixa usada: %QB21606..%QB21625

Dispositivos

Nome	Endereço do Escravo

Acrescentar... Remover Editar...

Tipo de Dado	Endereço Inicial de Leitura	Tamanho dos Dados de Leitura	Variável IEC de Leitura	Endereço Inicial de Escrita	Tamanho dos Dados de Escrita	Variável IEC de Escrita

Acrescentar... Remover Editar...

Figura 2-14. Tela de Configuração MODBUS RTU Mestre

- Adicionar e configurar dispositivos, definindo endereço, variáveis de representação direta (%Q) para desabilitar as relações, timeouts de comunicação etc. A figura a seguir ilustra a adição do dispositivo.

MODBUS_Device_01 Configurações

Configurações do Dispositivo

Nome: MODBUS_Device_01

Endereço do Escravo: 1

Time-out de Comunicação (ms): 1000

Número Máximo de Retentativas: 2

Desabilitação dos Mapeamentos: 4096

Faixa usada: %QX4096.0..%QX4099.7

OK Cancelar

Figura 2-15. Configuração do dispositivo MODBUS

- Adicionar e configurar relações MODBUS, especificando o tipo de dado e função MODBUS, timeouts, variáveis de representação direta (%Q) para receber os diagnósticos

da relação e outras para receber/escrever os dados, quantidade de dados a comunicar e varredura da relação. As figuras a seguir ilustram essa configuração.

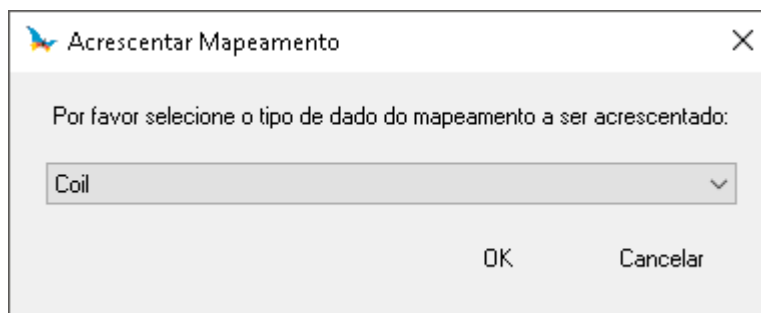


Figura 2-16. Tipo de dado MODBUS

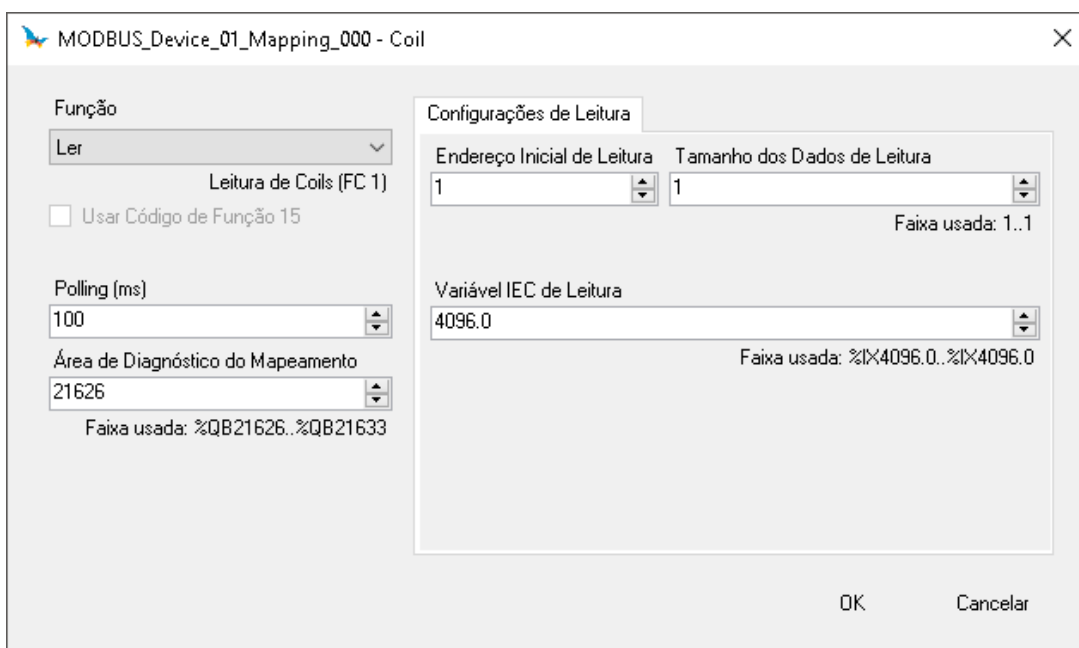


Figura 2-17. Função MODBUS



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.

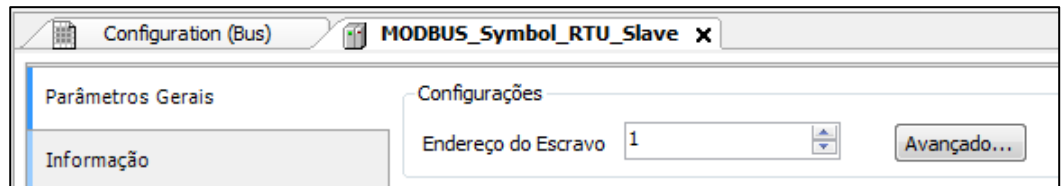


Figura 2-18. Configuração do Escravo

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 2-19. Tela de Mapeamentos de dados MODBUS

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

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

1. Configurar os parâmetros gerais do protocolo MODBUS escravo, como: tempos de comunicação, endereço e variáveis de representação direta (%Q) para receber os diagnósticos e controlar as relações. Veja a figura a seguir.

Configurações do Escravo

Endereço Inicial de Diagnósticos em %Q: 21606 Endereço do Escravo: 1

Tamanho: 20 Desabilitação dos Mapeamentos: 4096 Avançado...

Faixa usada: %QB21606..%QB21625 Faixa usada: %QX4096.0..%QX4099.7

Mapeamentos do Escravo

	Tipo de Dado	Endereço Inicial do Dado	Tamanho do Dado	Variável IEC
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				

Acrescentar... Remover Editar...

Figura 2-20. Tela de Configuração MODBUS RTU Escravo

- Adicionar e configurar relações MODBUS, especificando o tipo de dado MODBUS, variáveis de representação direta (%Q) para receber/escrever os dados e quantidade de dados a comunicar. Isso está ilustrado na figura a seguir. As figuras a seguir ilustram esse processo.

Acrescentar Mapeamento

Por favor selecione o tipo de dado do mapeamento a ser acrescentado:

Coil

OK Cancelar

Figura 2-21. Adicionando Relações MODBUS

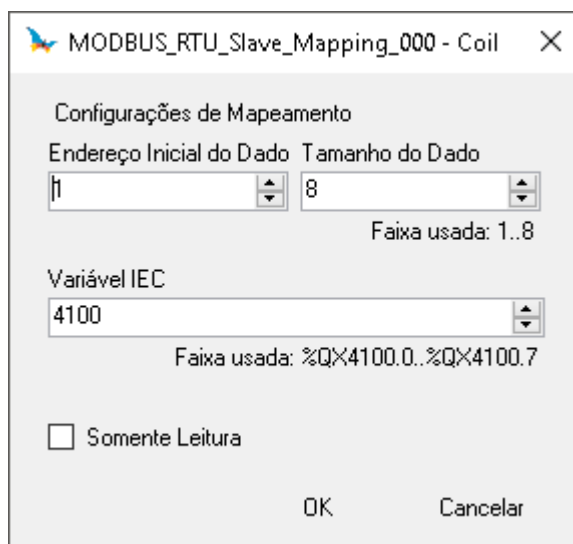


Figura 2-22. Configurando a Relação MODBUS



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 leiam ou escrevam variáveis MODBUS em outros controladores ou IHMs compatíveis com os protocolos MODBUS TCP ou MODBUS RTU via TCP. A UCP Nexto pode, simultaneamente,

ser cliente e servidor em uma mesma rede de comunicação, ou até mesmo ter mais instâncias associadas à interface Ethernet, indiferente se 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.

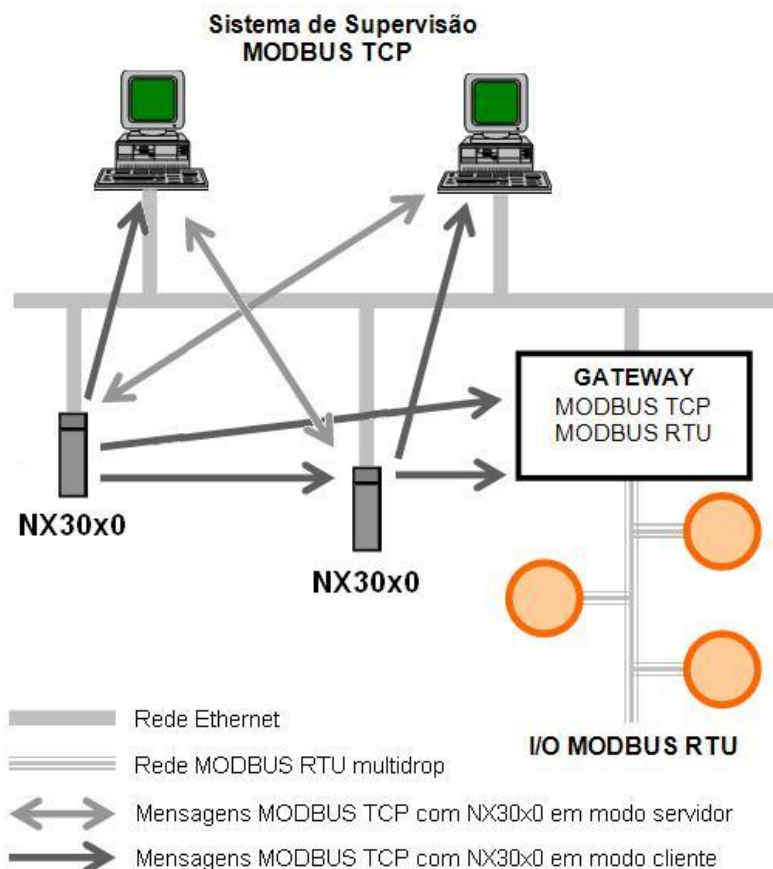


Figura 2-23. Rede de Comunicação MODBUS TCP

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.

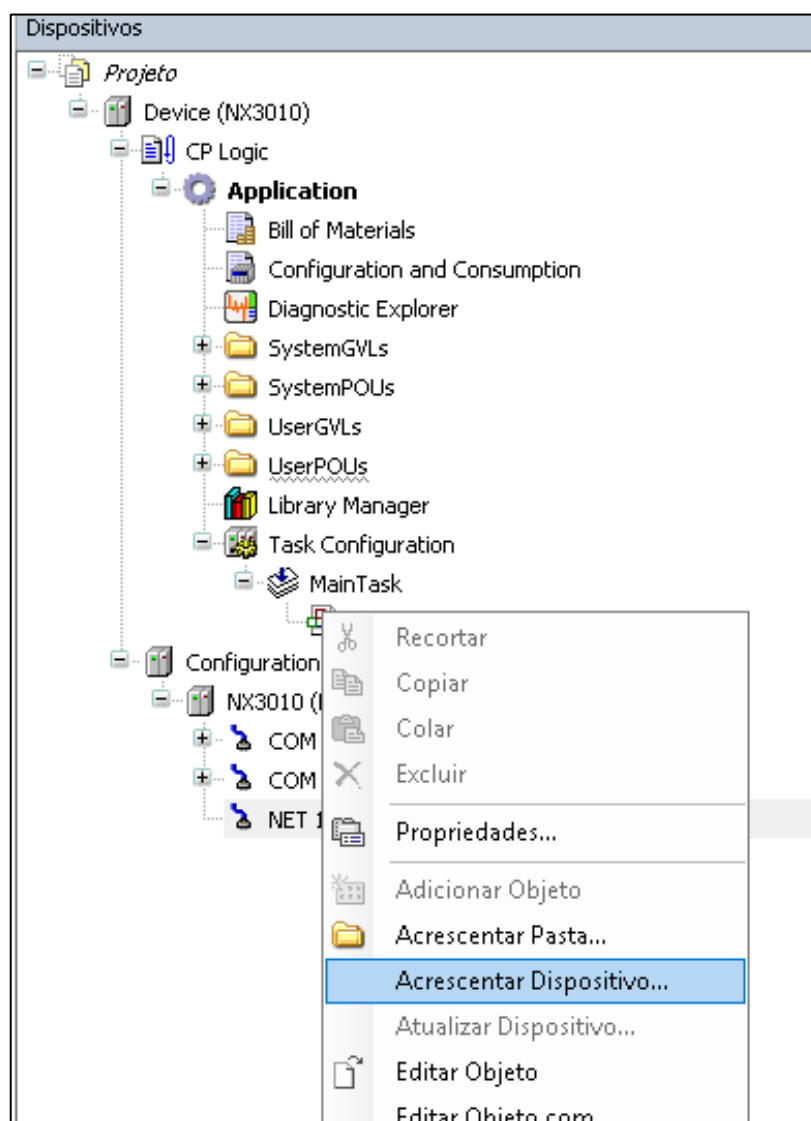


Figura 2-24. Adicionando a Instância MODBUS Ethernet

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, ou MODBUS Client, para configuração por Representação Direta (%Q). A seguir, clicar em Acrescentar, conforme mostra a figura a seguir.

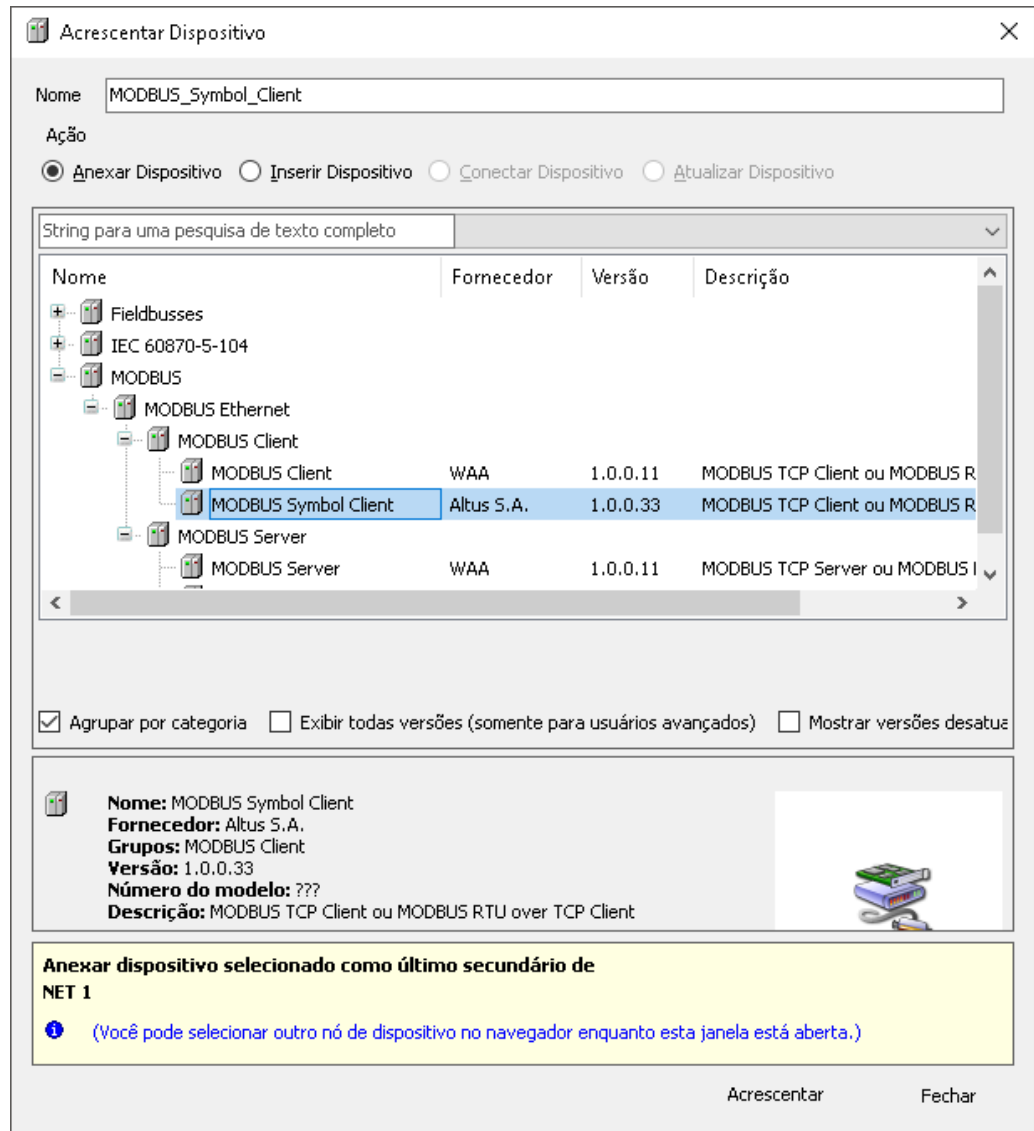


Figura 2-25. Selecionando o Protocolo MODBUS Ethernet

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.

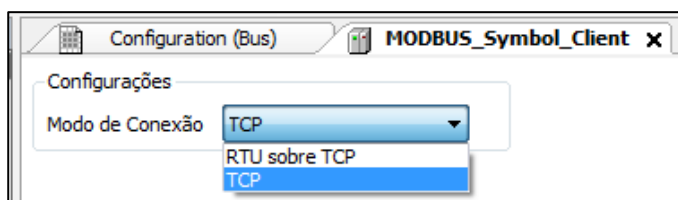


Figura 2-26. Tela de Configuração Parâmetros Gerais MODBUS Cliente

- 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.

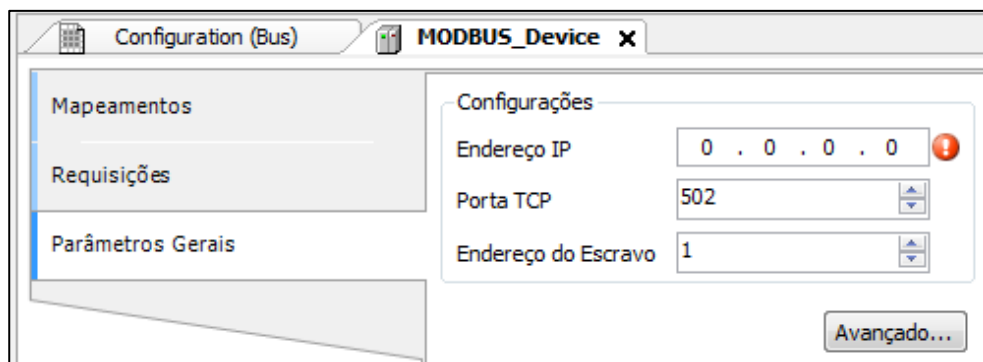


Figura 2-27. Tela de Configurações dos Parâmetros Gerais do Dispositivo MODBUS

- 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.

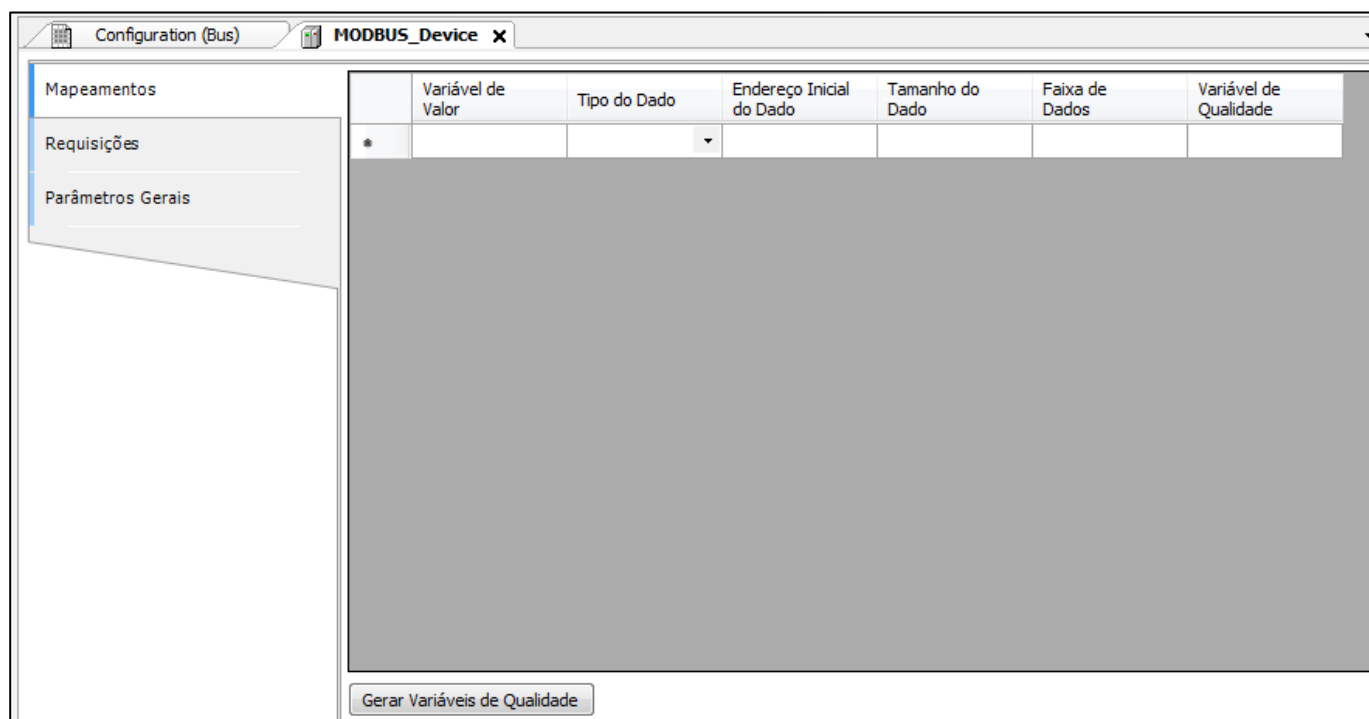


Figura 2-28. Tela de Mapeamentos de dados MODBUS Cliente

- 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	Variável de Diagnóstico	Variável de Desabilitação
*										

Figura 2-29. Tela de Requisições de dados MODBUS

Configuração do Protocolo MODBUS Ethernet Cliente por Representação Direta (%Q)

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

1. Configurar os parâmetros gerais do protocolo MODBUS, como: tempos de comunicação e variáveis de representação direta (%Q) para receber os diagnósticos. A figura a seguir ilustra essas configurações.

Configurações do Cliente

Endereço Inicial de Diagnósticos em %Q: 21626

Tamanho: 20

Faixa usada: %QB21626..%QB21645

Protocolo: RTU via TCP TCP

Dispositivos

Nome	IP de Destino	Porta TCP

Acrescentar... Remover Editar...

Tipo de Dado	Endereço do Escravo	Endereço Inicial de Leitura	Tamanho dos Dados de Leitura	Variável IEC de Leitura	Endereço Inicial de Escrita	Tamanho dos Dados de Escrita	Variável IEC de Escrita

Acrescentar... Remover Editar...

Figura 2-30. Tela de Configuração – MODBUS Cliente

2. Adicionar e configurar dispositivos, definindo endereço, variáveis de representação direta (%Q) para desabilitar as relações, timeouts de comunicação etc. Veja a figura a seguir.

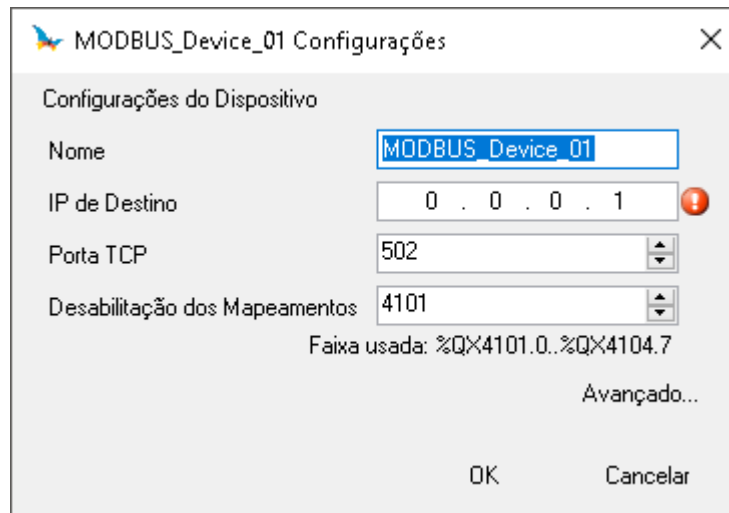


Figura 2-31. Configurando Cliente MODBUS

3. Adicionar e configurar relações MODBUS, especificando o tipo de dado e função MODBUS, timeouts, variáveis de representação direta (%Q) para receber os diagnósticos da relação e outras para receber/escrever os dados, quantidade de dados a comunicar e varredura da relação. As figuras a seguir ilustram esse processo.

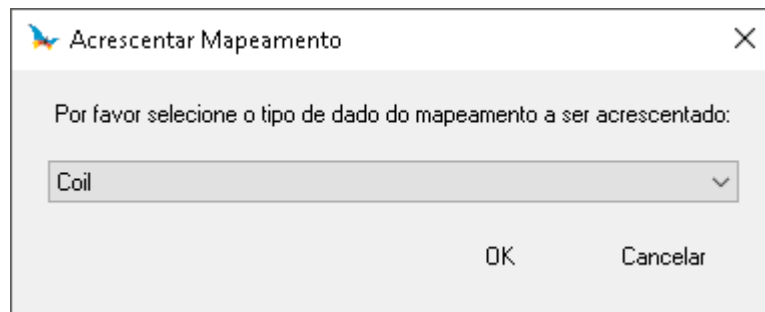


Figura 2-32. Tipo de dado MODBUS

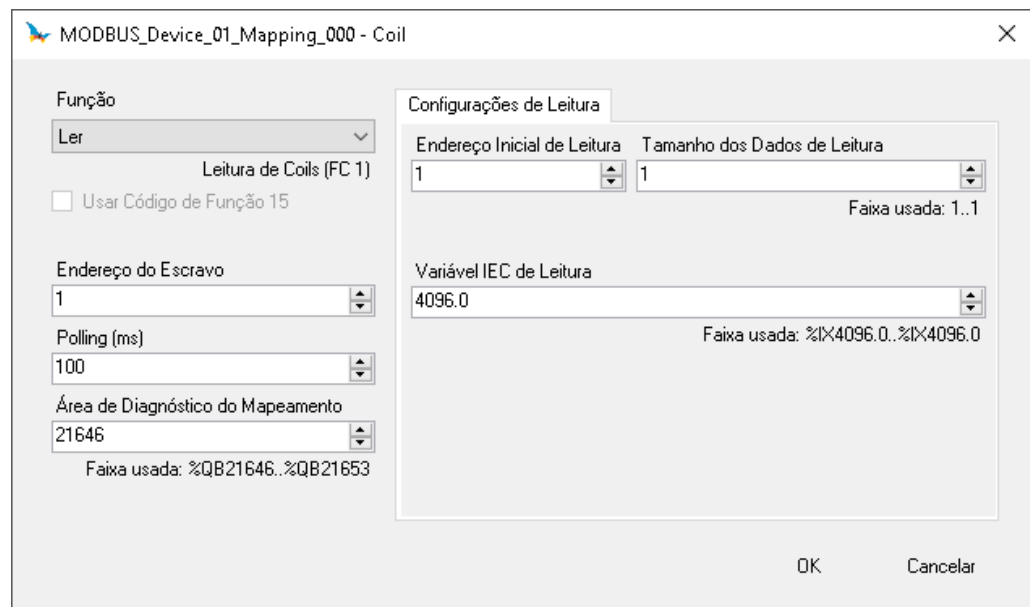


Figura 2-33. Função MODBUS

Disparo de Relações MODBUS Cliente de Forma Acíclica

Para disparar relações MODBUS Cliente de forma acíclica, sugere-se o seguinte método, que pode ser implementado de maneira simples no programa da aplicação do usuário:

- Definir tempo máximo de polling para as relações;
- Manter a relação normalmente desabilitada;
- Habilitar a relação no momento em que se deseja executá-la;
- Esperar pela confirmação de término da execução da relação, e neste momento desabilitá-la novamente.



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.

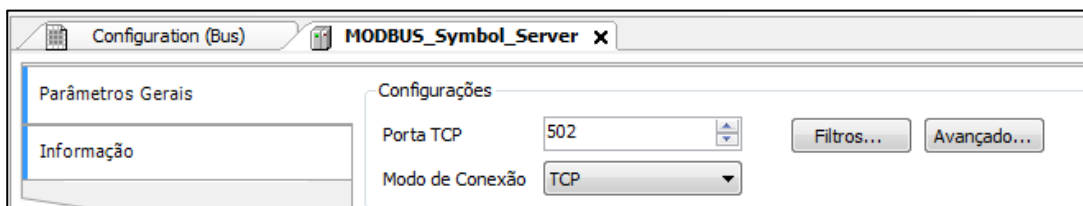


Figura 2-34. Tela de Configuração Parâmetros Gerais MODBUS Servidor

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.

Configurações do Servidor

Endereço Inicial de Diagnósticos em %Q: 1626 Porta TCP: 502 Protocolo: RTU via TCP TCP

Tamanho: 20 Desabilitação dos Mapeamentos: 4101 Avançado...

Faixa usada: %QB21626..%QB21645 Faixa usada: %QX4101.0..%QX4104.7

Mapeamentos do Servidor

	Tipo de Dado	Endereço Inicial do Dado	Tamanho do Dado	Variável IEC
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				

Acrescentar... Remover Editar...

Figura 2-35. Tela de Configuração MODBUS Servidor

Configuração do Protocolo MODBUS Ethernet Servidor por Representação Direta (%Q)

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

1. Configurar os parâmetros gerais do protocolo MODBUS servidor, como: tempos de comunicação, endereço e variáveis de representação direta (%Q) para receber os diagnósticos e controlar as relações. Veja a figura a seguir.

Configurações do Servidor

Endereço Inicial de Diagnósticos em %Q: 21626 Porta TCP: 502 Protocolo: RTU via TCP TCP

Tamanho: 20 Desabilitação dos Mapeamentos: 4101 Faixa usada: %QB21626..%QB21645 Faixa usada: %QX4101.0..%QX4104.7

Mapeamentos do Servidor

	Tipo de Dado	Endereço Inicial do Dado	Tamanho do Dado	Variável IEC
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				

Acrescentar... Remover Editar...

Figura 2-36. Tela de Configuração MODBUS Servidor

- Adicionar e configurar relações MODBUS, especificando o tipo de dado MODBUS, variáveis de representação direta (%Q) para receber/escrever os dados e quantidade de dados a comunicar.

Acrescentar Mapeamento

Por favor selecione o tipo de dado do mapeamento a ser acrescentado:

Coil

OK Cancelar

Figura 2-37. Tipo de dado MODBUS

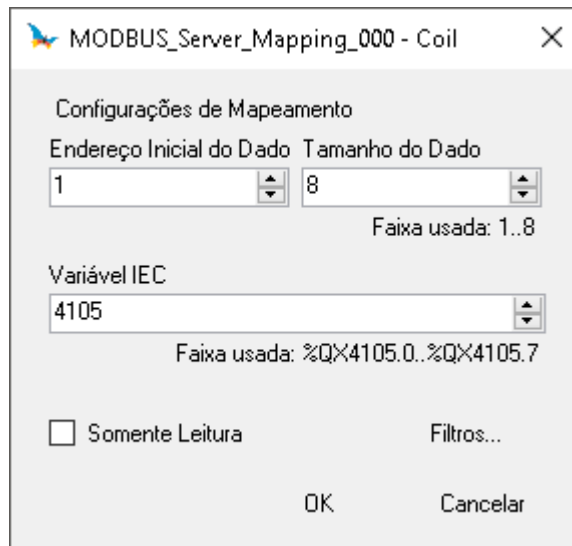


Figura 2-38. Função MODBUS Servidor



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 2-1: configuração do protocolo MODBUS Ethernet

Desenvolva uma aplicação através do protocolo MODBUS Ethernet Cliente, conforme orientações a seguir. Para a realização desta situação de aprendizagem utilizar o monitor Modbus 32 MDBUS.



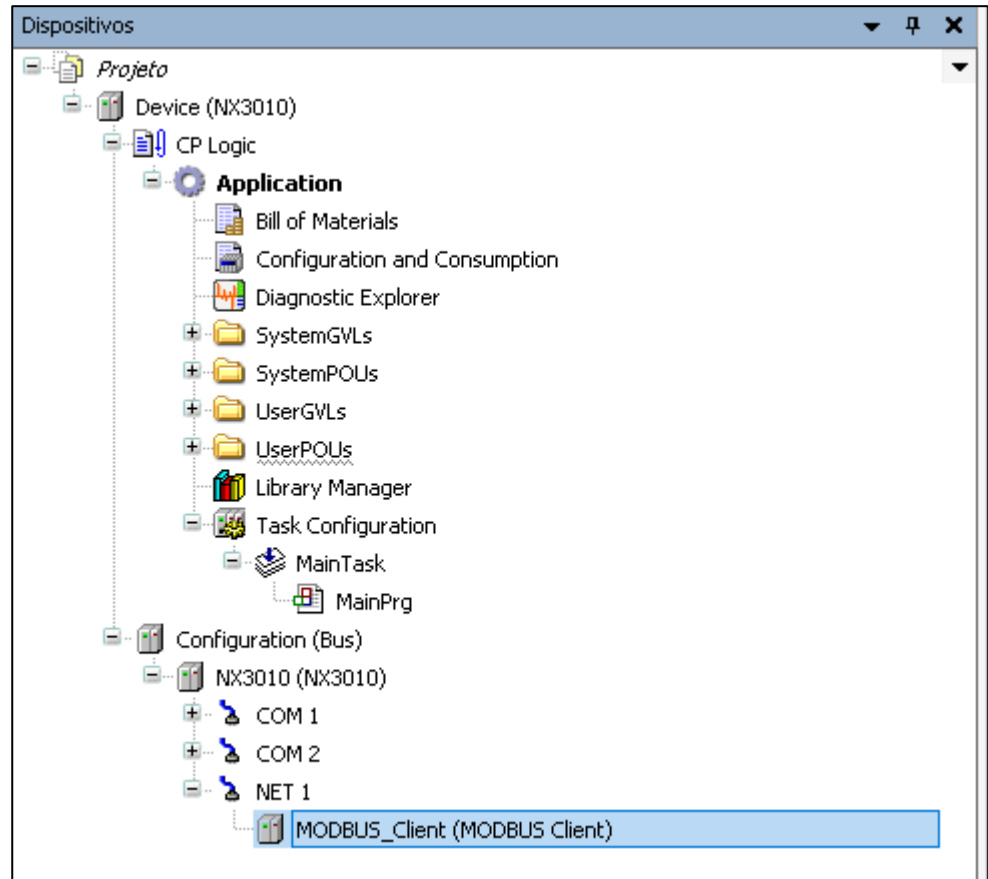
DICA: consulte o instrutor para orientações adicionais referentes à execução desta atividade.

ANOTAÇÕES

Implementação do Estudo Dirigido 2-1:

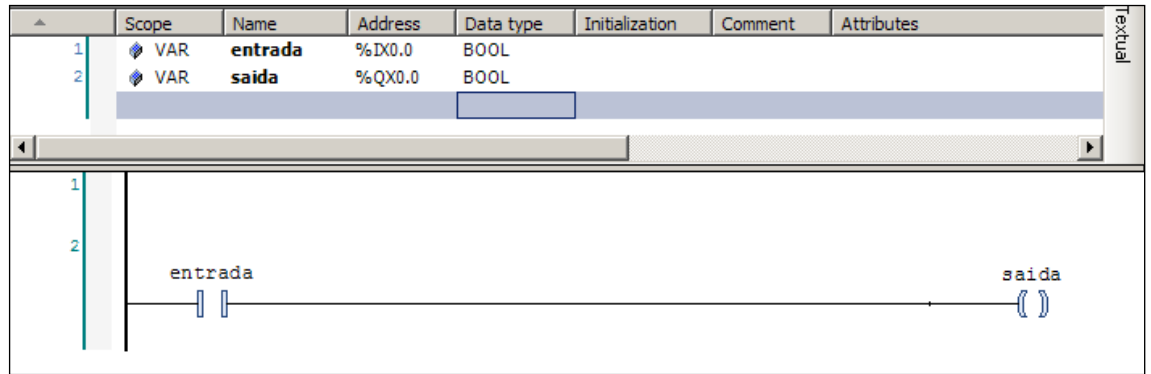
O objetivo principal desta aplicação é mostrar as facilidades na configuração das instâncias MODBUS na plataforma NEXTO.

- Como acrescentar em NET1 um dispositivo MODBUS Ethernet Cliente?
- Como configurar o dispositivo MODBUS Ethernet Cliente?
- Como monitorar uma aplicação na ferramenta 32 MDBUS?



Com um duplo click na aba NET1 em MODBUS_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 fabrica.

Desenvolver o programa LADDER da figura abaixo com as respectivas variáveis IEC.



Utilizar os parâmetros que seguem no mapeamento MODBUS

- Como configurar o IP e a porta TCP?

Configurações do Cliente

Endereço Inicial de Diagnósticos em %Q:

Tamanho:

Faixa usada: %QB21626..%QB21645

Protocolo:

RTU via TCP

TCP

Dispositivos

	Nome	IP de Destino	Porta TCP
0	MODBUS_Device_01	192.168.15.169	502

Acrescentar... Remover Editar...

Mapeamentos do Dispositivo de MODBUS_Device_01

	Tipo de Dado	Endereço do Escravo	Endereço Inicial de Leitura	Tamanho dos Dados de Leitura	Variável IEC de Leitura	Endereço Inicial de Escrita	Tamanho dos Dados de Escrita	Variável IEC de Escrita
0	Coil	1	1	1	%IX0.0			
1								
2								
3								
4								
5								

Acrescentar... Remover Editar...

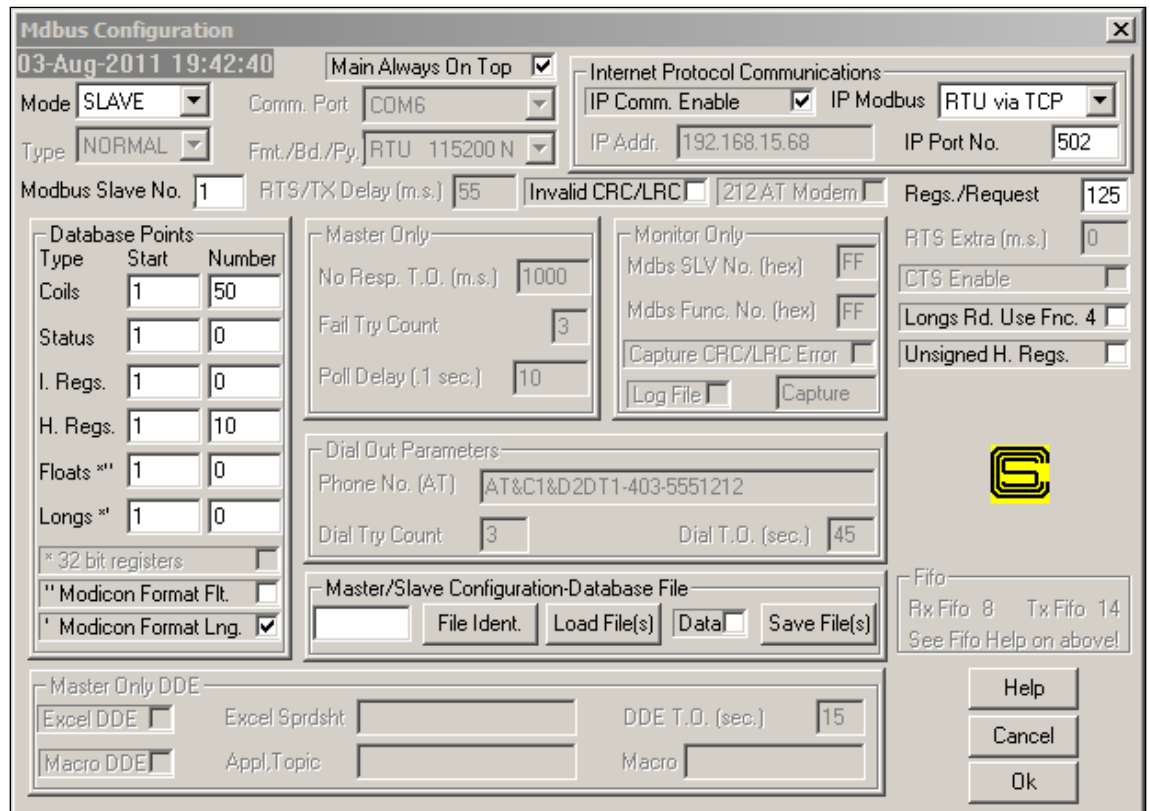
Configurar os parâmetros conforme a tabela abaixo.

Configuração (Menu em Inglês)	Descrição	Possibilidades
Data Type	Função MODBUS	Coil
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	%iX0.0

Observar que a função MODBUS (Read Data) corresponde ao endereço 1 em MODBUS Coils da ferramenta de monitoração 32MDBUS.

Configuração da ferramenta 32MDBUS.

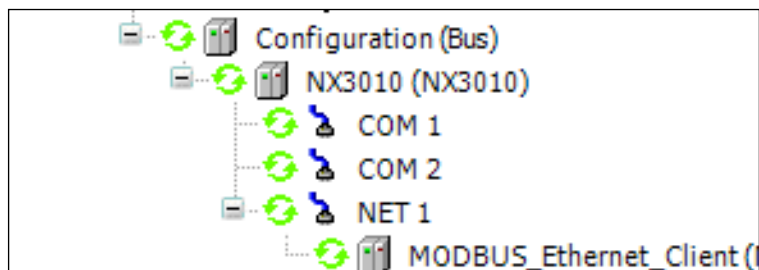
- Como configurar a ferramenta?
- Qual endereço de IP colocar?
- Qual porta devo configurar?
- Qual modo devo selecionar?



Selecionar o modo *Slave*, comunicação via RTU via TCP e porta 502 conforme configuração padrão mostrado na figura acima.

Observação: utilizar este mesmo padrão de configuração da ferramenta de monitoração 32MDBUS para as demais aplicações.

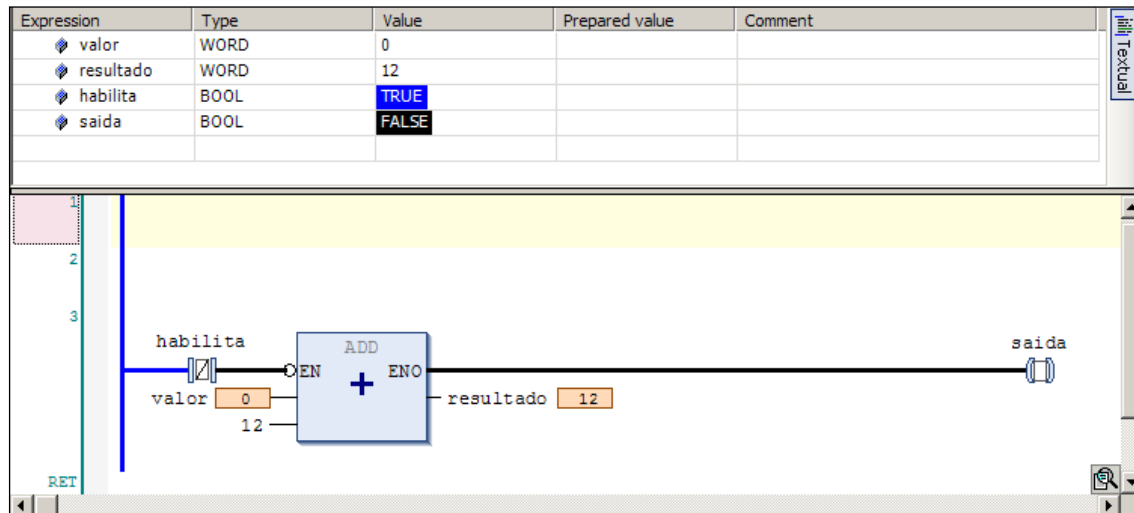
Para escrever os dados nos registradores utilize a ferramenta 32MDBUS. Selecione na aba Displays, da ferramenta, a opção *Holding Regs* para a realização desta experiência prática. Configure o protocolo MODBUS Ethernet Client no MasterTool IEC XE.



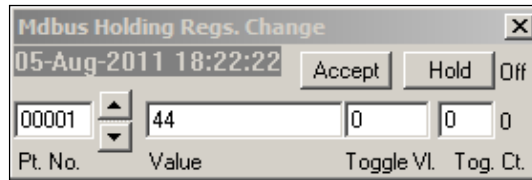
Configurar o cliente MODBUS com os parâmetros da tabela.

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 MainPrg editar o programa conforme a figura abaixo e colocar o sistema em *run* através do *login*. Observar no painel da NEXTO o led indicativo DG na cor verde indicando o estado *run*.



Observar que a variável "valor" está com o valor "0". Através da ferramenta 32MDBUS selecione em Mdbus *Holding Regs* conforme a figura abaixo.



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.

Observar no painel que o led DG sinaliza a CPU NEXTO em estado de forçamento. Este experimento funciona tanto com a CPU NEXTO em estado de forçamento ou em *run*.

Expression	Type	Value	Prepared value	Comment
valor	WORD	44		
resultado	WORD	56		
habilita	BOOL	F FALSE		
saida	BOOL	TRUE		

Protocolo OPC na Série Nexto

Para comunicar com as UCPs da Série Nexto é possível utilizar a tecnologia OPC DA (*Open Platform Communications Data Access*). Esta plataforma de comunicação aberta foi desenvolvida para ser o padrão utilizado nas comunicações industriais.

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

Uma analogia muito comum, utilizada para descrever a tecnologia OPC DA, é a de uma impressora. Quando corretamente conectada, o computador precisa de um driver para ter a interface com o equipamento. Muito similar, o OPC DA auxilia na interface entre o sistema de supervisão com os dados de campo no CP.

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

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.

OPC DA Servidor

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

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.

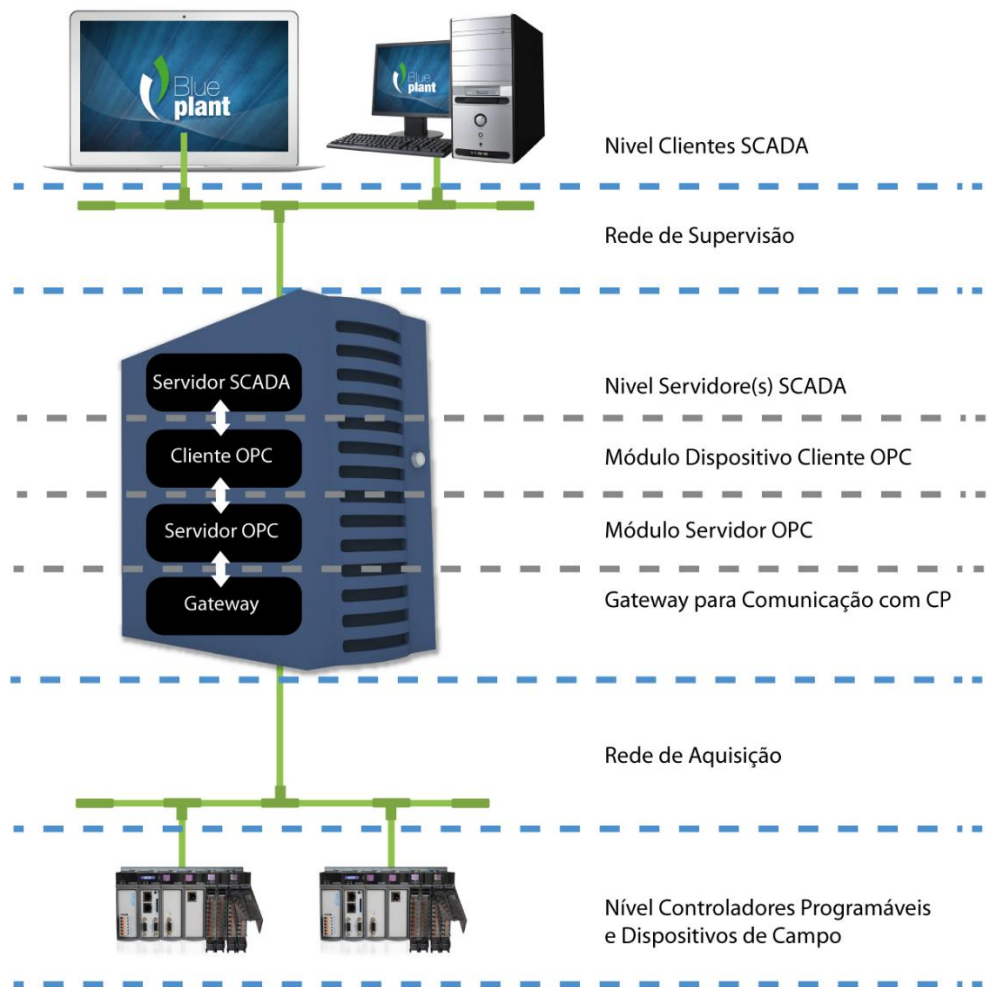


Figura 2-39. Arquitetura OPC DA

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 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.

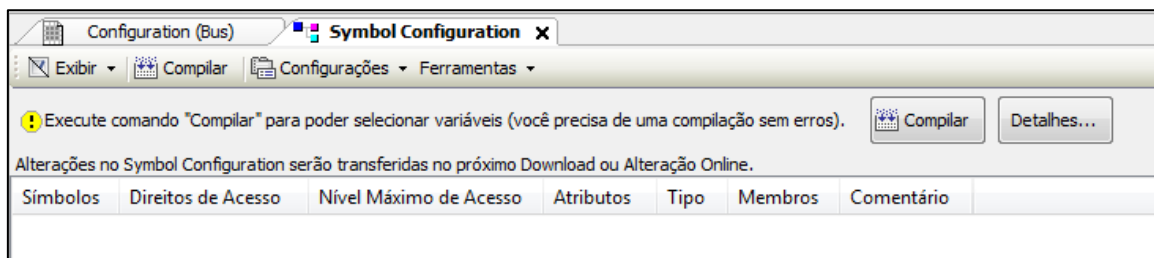


Figura 2-40. Objeto Symbol Configuration

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.

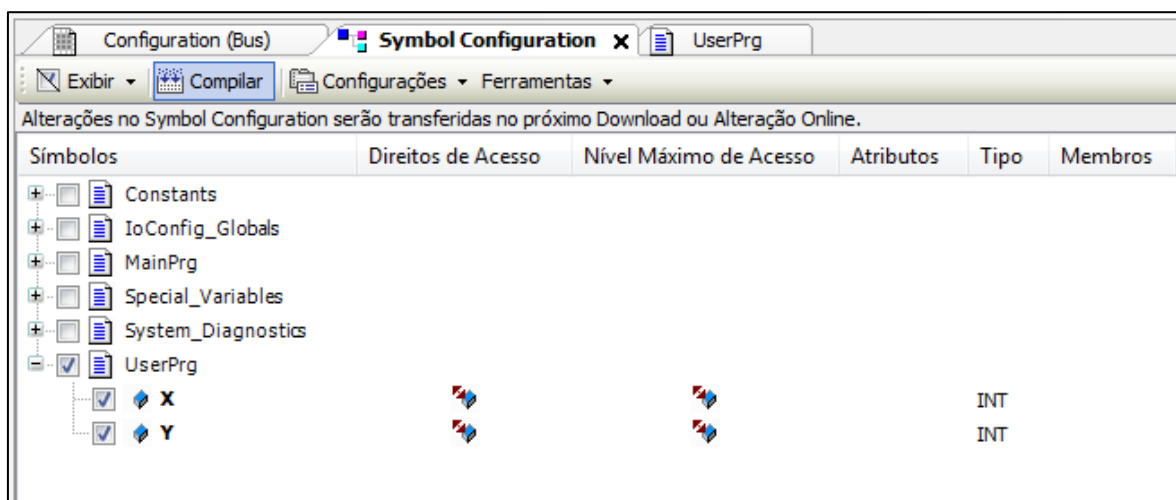


Figura 2-41. Selecionando Variáveis na Symbol Configuration

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 onde a variável é declarada. Por exemplo, no caso de uma instância DUT declarada dentro de uma POU ou GVL que possuem um atributo, este não irá impactar no comportamento dos elementos da instância desta DUT.

Será necessário aplicar o mesmo nível de acesso na declaração da DUT.

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.

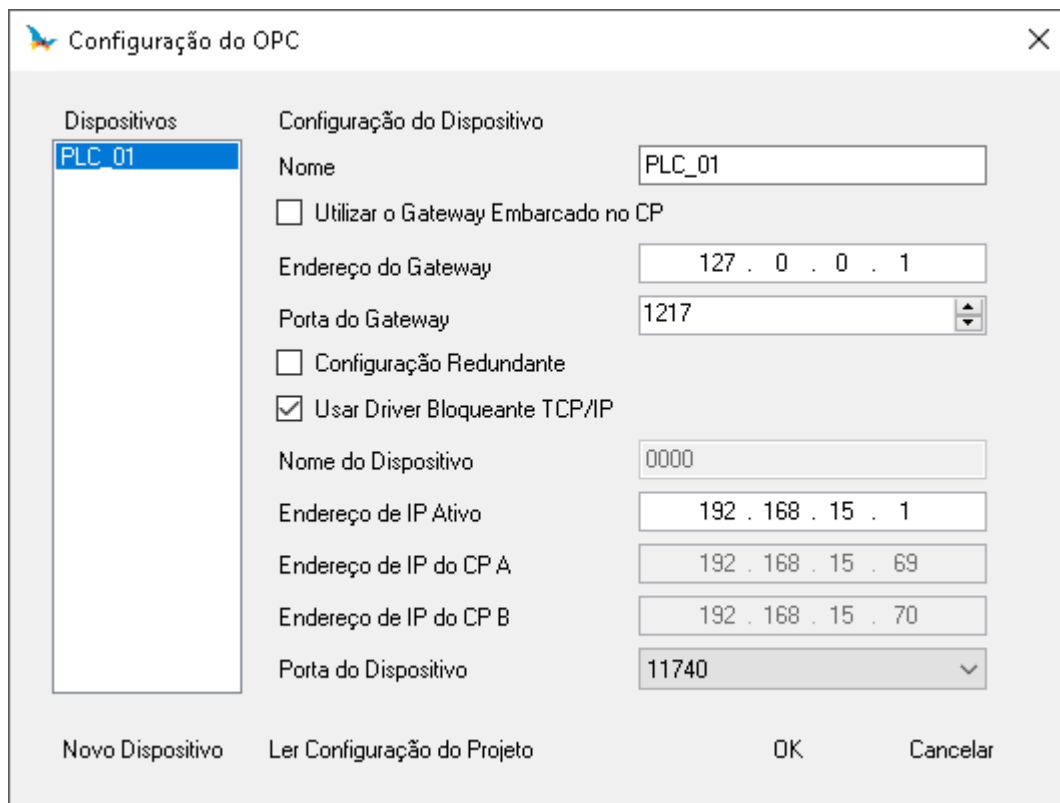


Figura 2-42. Configuração do OPC DA Servidor

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 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.

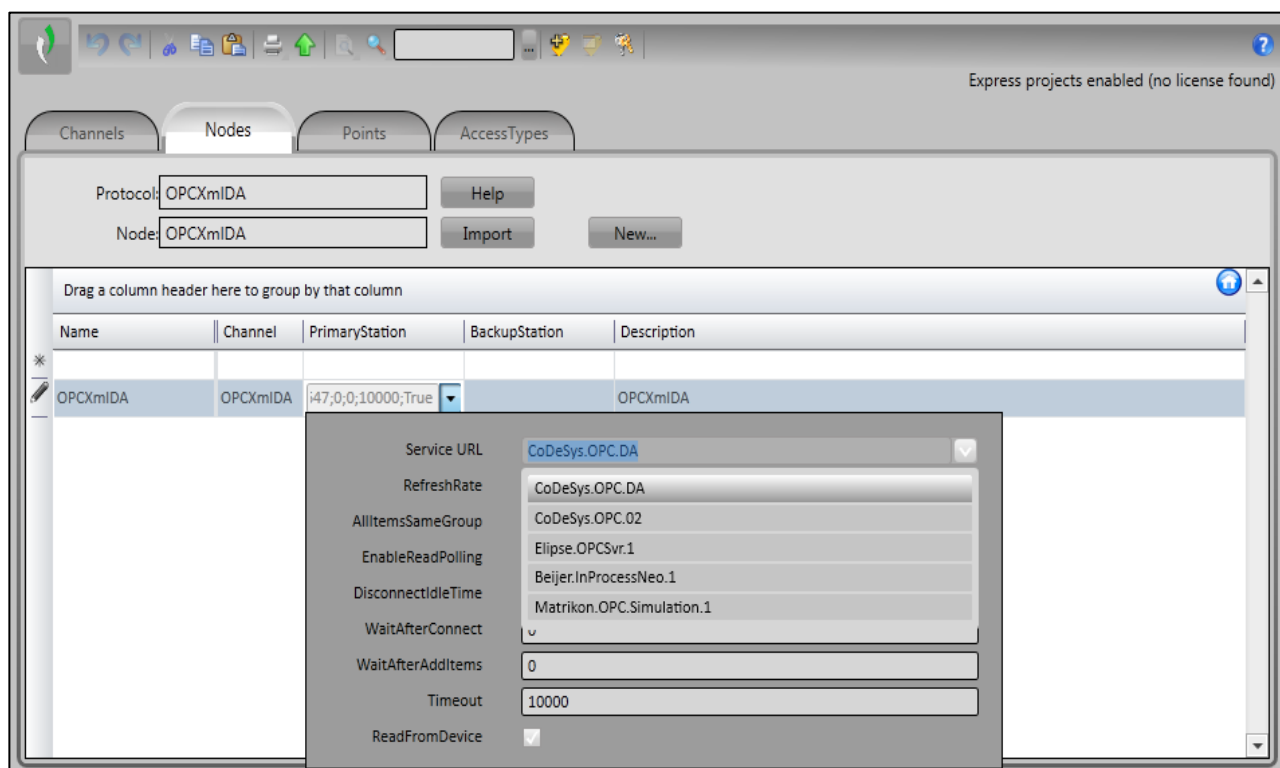


Figura 2-43. Selecionando Servidor OPC DA na Configuração do Cliente

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.

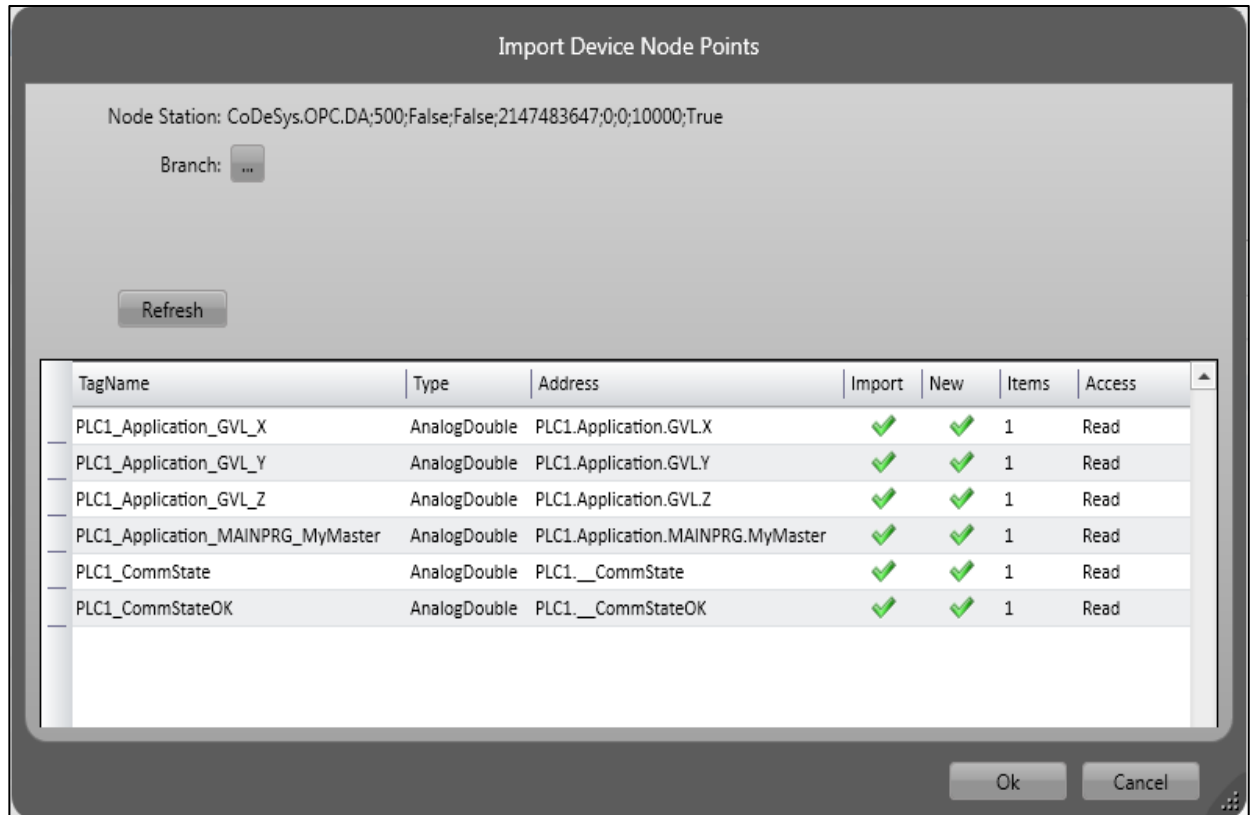


Figura 2-44. Lista de Símbolos Consultados pelo Cliente OPC DA

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.

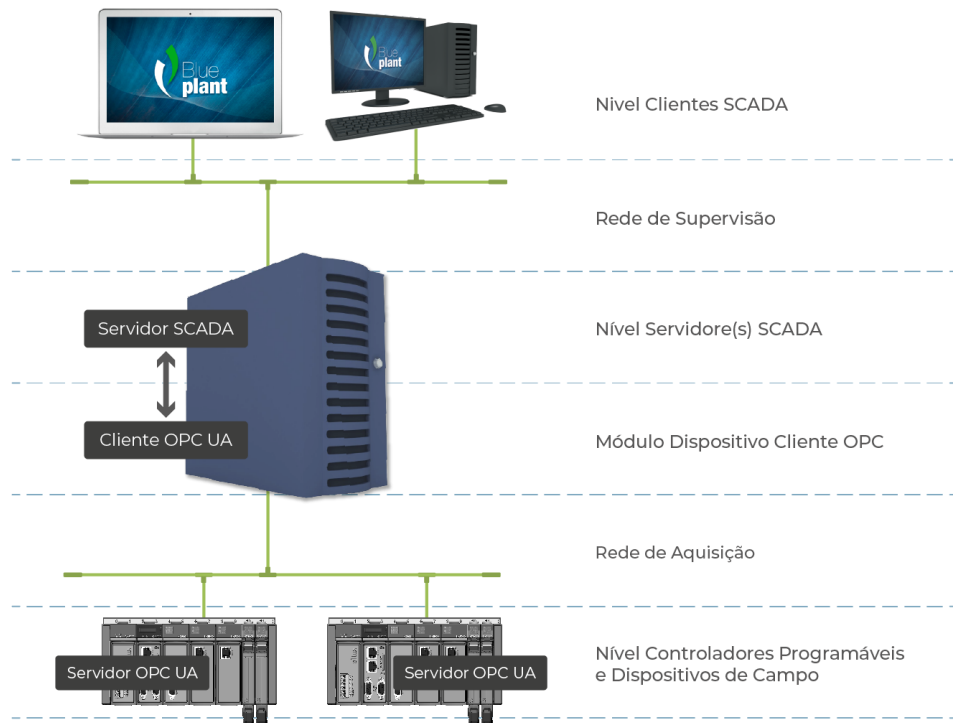


Figura 2-45. Arquitetura Típica OPC UA

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. 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.

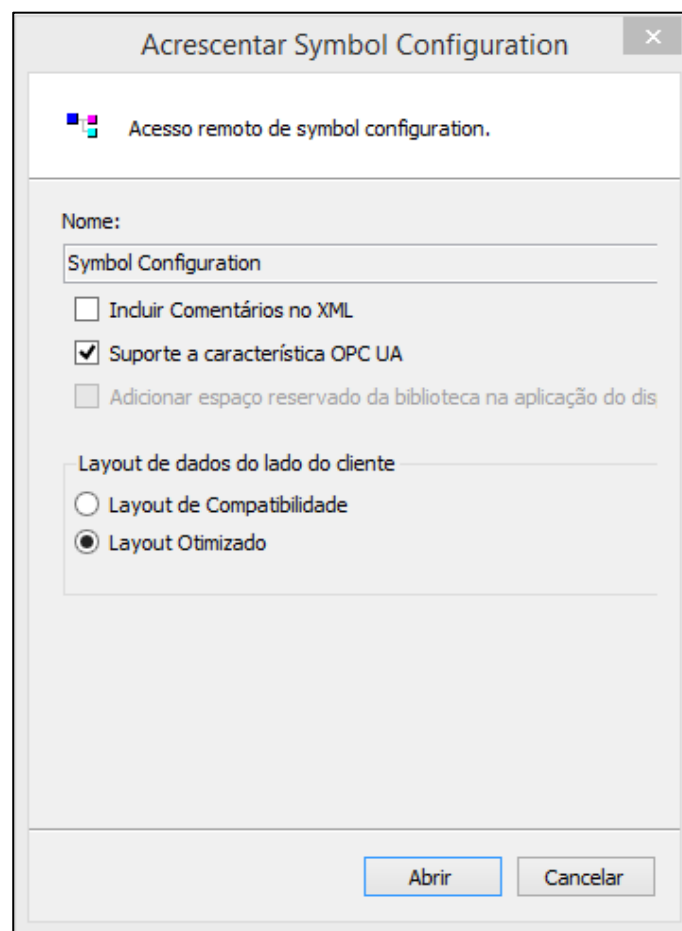


Figura 2-46. Objeto Symbol Configuration (OPC UA)

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.

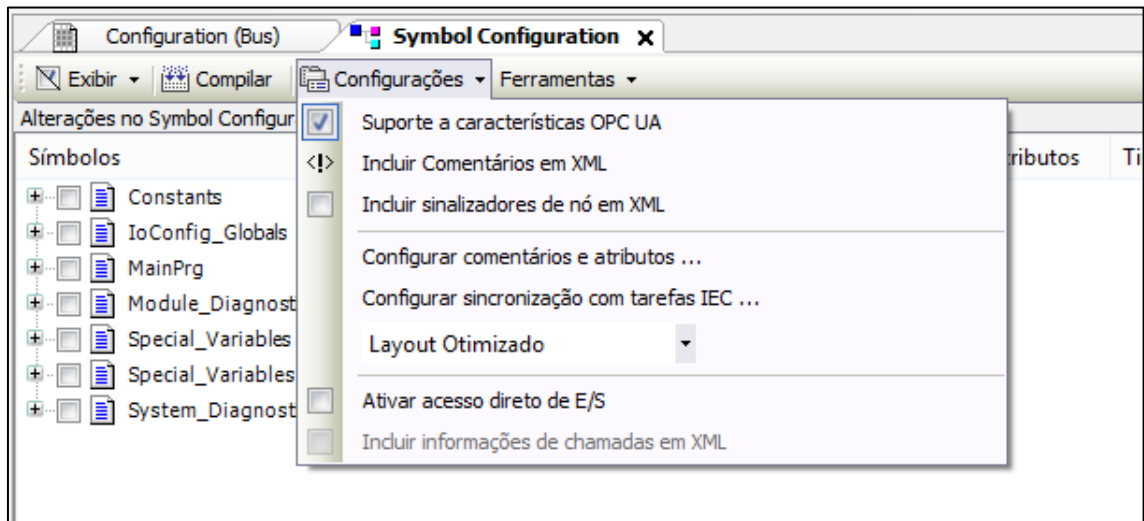


Figura 2-47. Habilitando OPC UA no Objeto Symbol Configuration

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



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.

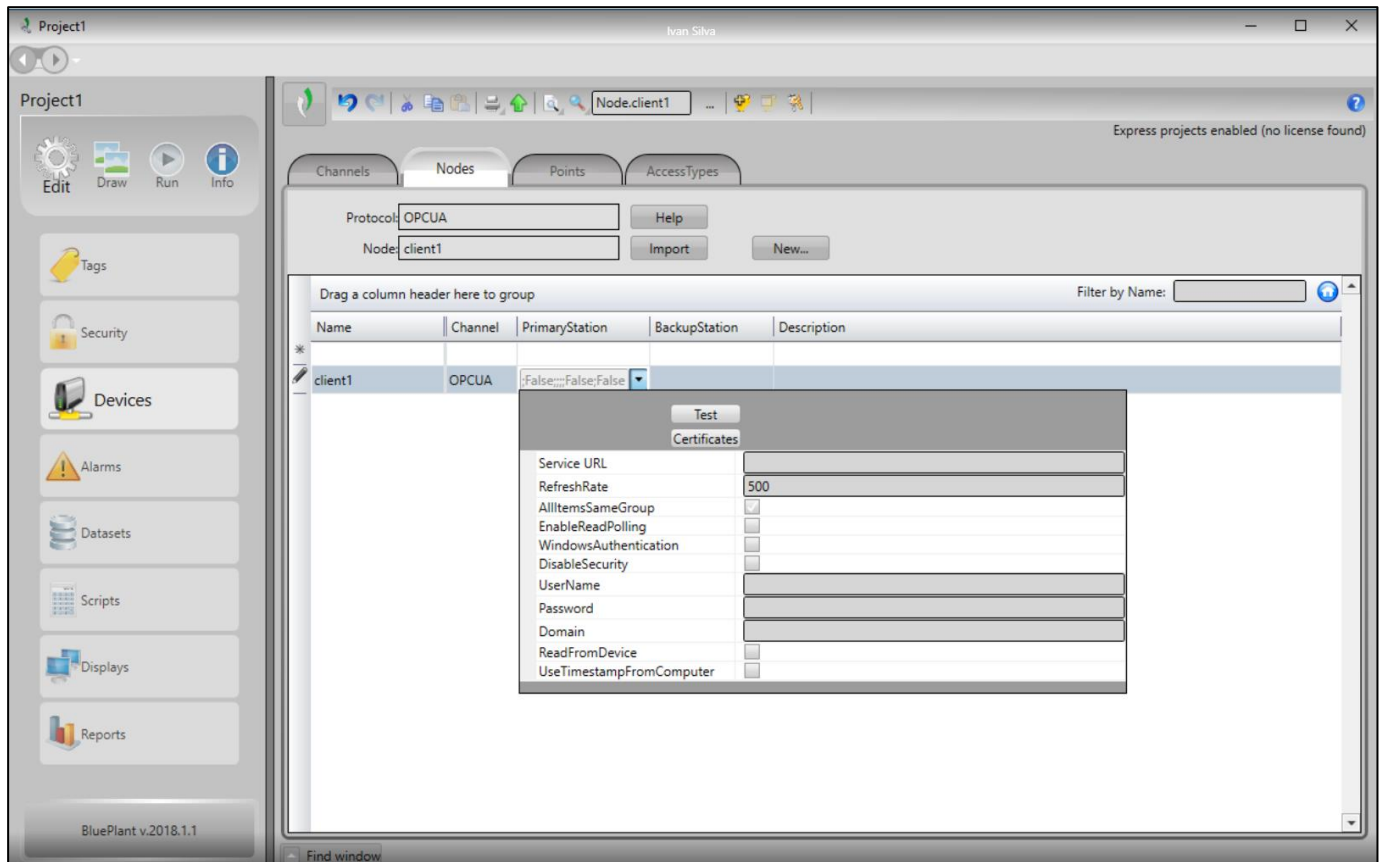


Figura 2-48. Selecionando Servidor OPC UA na Configuração do Cliente

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.

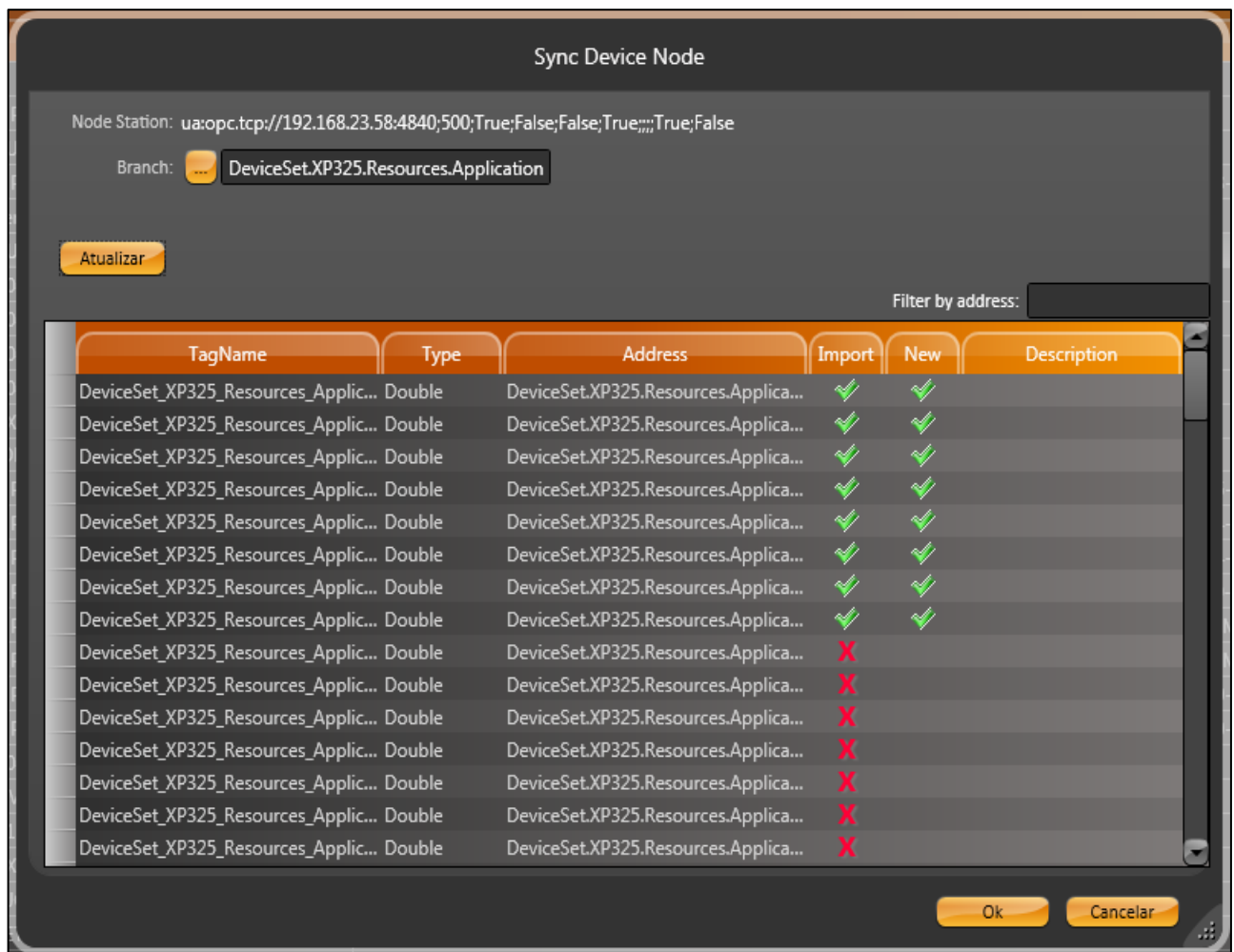


Figura 2-49. Lista de Símbolos Consultados pelo Cliente OPC UA



Estudo Dirigido 2-3: configurando o Nexto como servidor OPC UA

Desenvolva uma aplicação com ênfase no protocolo OPC UA que permita a comunicação entre o CP Nexto (servidor) e o SCADA BluePlant (cliente).




DICA: consulte o instrutor para orientações adicionais referentes à execução desta atividade.

ANOTAÇÕES

Implementação do Estudo Dirigido 2-3:...

Consulte na Base de Conhecimento Altus o tutorial “Como configurar um controlador Nexto como servidor OPC UA”. Nesse tutorial você irá aprender a configurar um CP da Série Nexto como servidor OPC UA, permitindo comunicar-se com outros dispositivos através desse protocolo de

comunicação. O tutorial reforça também, a importância do OPC UA nas aplicações da indústria e contempla um exercício prático de comunicação entre o CP Nexto e o SCADA BluePlant.

 **DICA:** o tutorial foi apresentado em um Webinar Altus conforme mostrado a seguir.



Outros Protocolos das Interfaces Ethernet

EtherCAT Mestre

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

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

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

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

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

Instalando e Inserindo Dispositivos EtherCAT

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

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

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

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

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

Além das topologias de linha e de árvore o MasterTool IEC XE também suporta a topologia em estrela de EtherCAT. Para a configuração de uma topologia em estrela EtherCAT junções especiais EtherCAT são necessárias (no exemplo: Beckhoff EK1122). Uma estrela EtherCAT modular pode ser realizada utilizando várias junções. Dispositivos individuais ou uma seção de

linha EtherCAT completa podem ser conectados nas portas de junção. Uma junção EtherCAT está marcada com o ícone. O exemplo de Árvore de Dispositivos na figura a seguir mostra diferentes possibilidades.

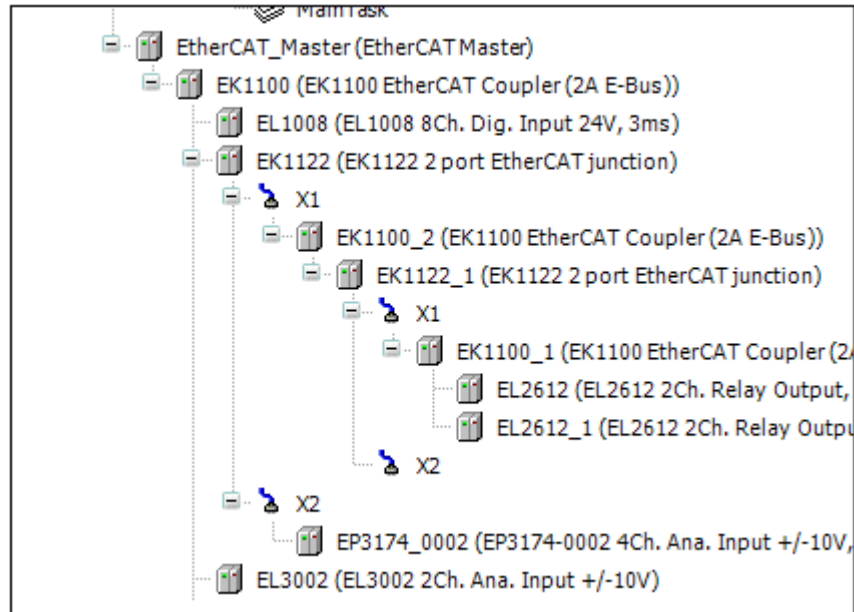


Figura 2-50. Exemplo de Configuração EtherCAT

Procurar Dispositivos

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

É importante salientar que, quando o comando Procurar Dispositivos é selecionado, uma conexão com o CP será estabelecida automaticamente antes da busca iniciar e será encerrada quando a busca terminar. Assim, para que este comando seja executado pela primeira vez, a conexão do Gateway deve ser configurada e deve-se fazer um download do projeto com um Mestre EtherCAT configurado no CP. No caso de futuras adições de dispositivos Escravos, para executar este comando é necessário que a rede EtherCAT esteja parada, para isso, colocar em TRUE o bit bStopBus presente nas variáveis de Diagnóstico do Mestre EtherCAT.

Quando o comando for executado, o campo Dispositivos Mapeados vai conter uma lista de todos os dispositivos e módulos encontrados durante a última verificação. Para adicioná-los ao projeto, basta clicar no botão Copiar Tudo Para o Projeto. É possível, também, executar uma comparação entre os dispositivos encontrados na busca com os presentes no projeto, selecionando a caixa Mostrar Diferenças Para o Projeto.

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

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

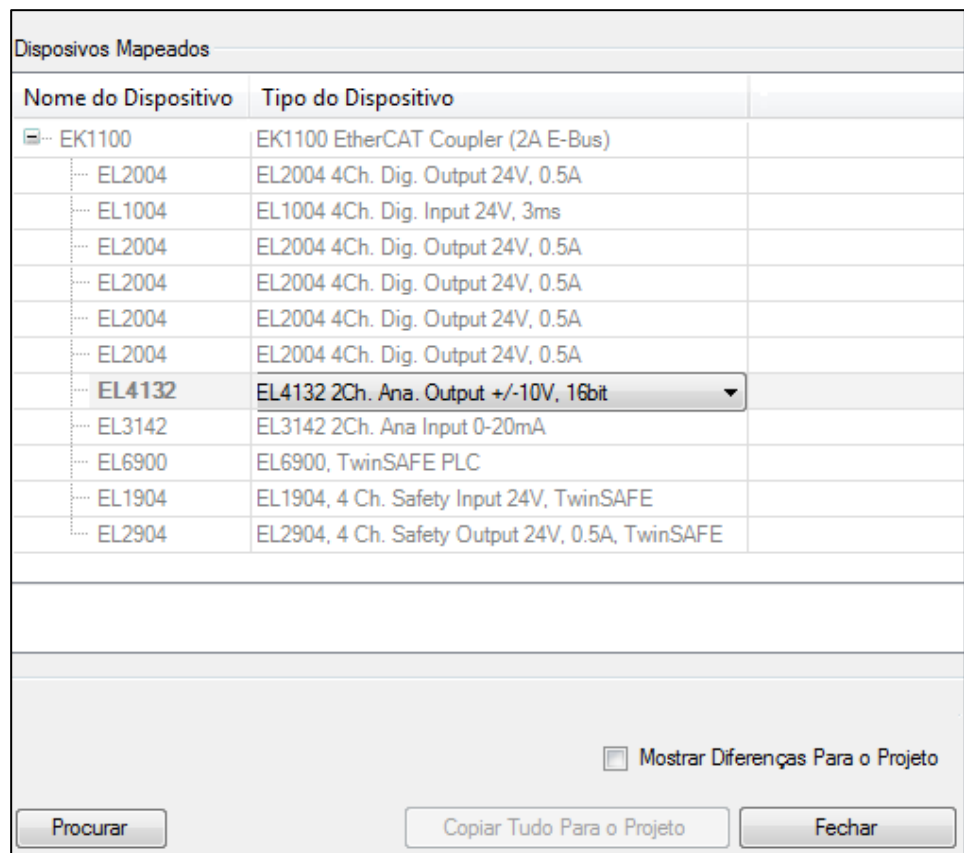


Figura 2-51. Diálogo Procurar Dispositivos para EtherCAT

Diagnóstico de Variáveis

Através da inserção de um Mestre e Escravo EtherCAT é adicionada uma variável de diagnóstico para o dispositivo na GVL System_Diagnostics. Esta variável fornece informações sobre o estado do dispositivo. Existem dois tipos de variáveis, uma para o Mestre EtherCAT e outra para os escravos EtherCAT. Cada variável tem informações específicas sobre o dispositivo.

Configuração do Mestre EtherCAT

A figura a seguir mostra os parâmetros gerais associados à configuração de um Mestre EtherCAT, tal como definido no Arquivo de Descrição do Dispositivo.

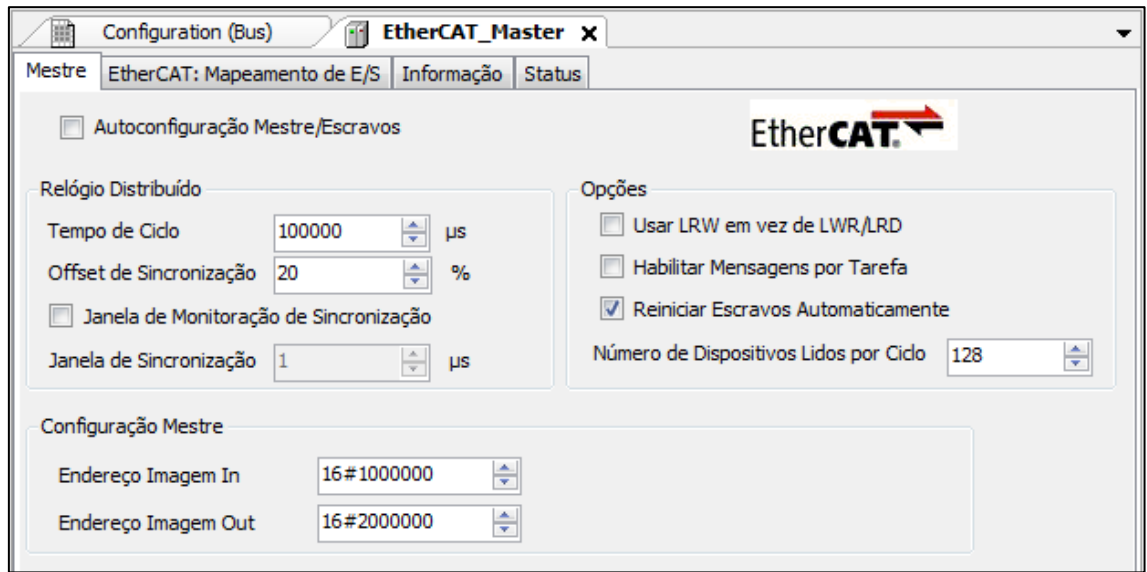


Figura 2-52. Diálogo de Configuração do Mestre EtherCAT

A guia EtherCAT Mestre - Mapeamento de E/S do editor de configuração do Mestre EtherCAT oferece a possibilidade de atribuir as variáveis de projeto para respectivas entradas e saídas do EtherCAT. Assim, as variáveis do Mestre EtherCAT podem ser controladas pela Aplicação de Usuário.

Além disso, podemos escolher a opção de Tarefa Cíclica de Barramento através da lista de seleção, onde podemos alterar a tarefa que se deseja utilizar. Todas as tarefas adicionadas ao projeto estarão presentes nesta lista. Essa tarefa serve para efetuar as operações do Mestre EtherCAT. A MainTask é a opção padrão deste campo.

Configuração do Escravo EtherCAT

A figura a seguir mostra as principais opções de configuração (parâmetros) para um Escravo EtherCAT, tal como definido no Arquivo de Descrição do Dispositivo. Estes campos estarão disponíveis somente se o modo Autoconfiguração (Mestre) não estiver ativado.

Endereço

Endereço AutoInc: 0

Endereço EtherCAT: 1001

Adicional

Habilitar Config. Avançadas

Opcional

Relógio Distribuído

Selecionar DC: DC unused

Habilitar 4000 Sinc. Unidade de Ciclo (µs)

Sync0:

Habilitar Sync 0

Sinc. Unidade de Ciclo [] Tempo de Ciclo (µs)

Definido pelo Usuário 0 Tempo de Deslocamento

Sync1:

Habilitar Sync 1

Sinc. Unidade de Ciclo [] Tempo de Ciclo (µs)

Definido pelo Usuário 0 Tempo de Deslocamento

Verificando Inicialização

Verificar ID do Fornecedor

Verificar ID do Produto

Timeouts

Acesso SDO 1000 ms

I -> P 3000 ms

P -> S / S -> O 10000 ms

Controle de unidade de ciclo DC: atribuir a µC local.

Unidade de Ciclo Latch Unit 0 Latch Unit 1

Figura 2-53. Diálogo de Configuração do Escravo EtherCAT

Os diálogos associados à configuração do escravo estão listados na sequência.

- **FMMU/Sync:** este diálogo só será fornecido em uma guia de um editor de configuração Escravo EtherCAT, se o modo Autoconfiguração no Mestre estiver desativado. Ele mostra os FMMUs e o Gerenciador Sync do Escravo, conforme definido pelo Arquivo de Descrição do Dispositivo. Essas configurações podem ser retrabalhadas, por exemplo, para configurar uma comunicação Escravo-para-Escravo. Nesse contexto, a tabela FMMU/Sync – FMMU mostra as Unidades do Gerenciador de Memória de Rede de Campo (FMMU - Fieldbus Memory Management Unit) do Escravo que são utilizadas para o tratamento dos dados do processo e a tabela FMMU/Sync - Gerenciador Sync mostra o(s) Gerenciador(es) de Sincronização do Escravo.
- **Dados do Processo e Dados de Processo Avançados:** a guia Dados do Processo do editor de configuração do Escravo EtherCAT mostra os dados do processo de entrada e de saída do Escravo, cada um definido por nome, tipo e índice do arquivo de descrição do dispositivo. As entradas selecionadas (a serem lidas) e saídas (a serem escritas) do dispositivo estarão disponíveis no diálogo EtherCAT Escravo - Mapeamento de E/S como saídas e entradas para o CP, no qual as variáveis do projeto podem ser mapeadas. A fim de modificar a seleção atual, você deve primeiro clicar com o mouse sobre a caixa de

seleção dos dados selecionados no momento, para cancelar a seleção. Depois disso, você pode selecionar outros. O diálogo Dados de Processo Avançados só será visível no editor de configuração do Escravo EtherCAT se a opção Habilitar Config. Avançadas do Escravo estiver ativada. Ele fornece uma visão mais detalhada dos dados do processo, além do que é apresentado na aba Dados do Processo. Além disso, nesta aba é possível habilitar o download da Atribuição PDO e a Configuração PDO.

- **Dados do Processo e Dados de Processo Avançados - Editando a Lista PDO:** este diálogo é aberto através do menu de contexto da área Lista PDO.
- **Dados do Processo e Dados de Processo Avançados - Definição do Conteúdo do PDO:** este diálogo é acessado através do menu de contexto da área PDO Content e o seu conteúdo, além da possibilidade de acesso desta janela, varia de acordo com o Escravo EtherCAT utilizado.
- **Parâmetros de Inicialização:** na guia Parâmetros de Inicialização podem ser definidos parâmetros para o dispositivo, que serão repassados pelos SDOs (Service Data Objects) ou IDN na inicialização do sistema. As opções presentes neste ambiente, além da possibilidade de acesso, variam de acordo com o Escravo EtherCAT utilizado e estão presentes no seu Arquivo de Descrição do Dispositivo.
- **Online:** o diálogo Online só será visível no editor de configuração Escravo EtherCAT, se a opção Habilitar Config. Avançadas do Escravo estiver ativada e a Aplicação estiver conectada ao dispositivo. Ele fornece uma visão com informações de status dos Escravos e funções para transferir arquivos para os Escravos sobre EtherCAT (FoE).
- **EtherCAT Escravo - Mapeamento de E/S:** Esta guia do editor de configuração do Escravo EtherCAT oferece a possibilidade de atribuir as variáveis de projeto para respectivas entradas e saídas do EtherCAT. Assim, as variáveis do Escravo EtherCAT podem ser controladas pela Aplicação de Usuário.
- **Guias Status e Informação:** A guia Status do editor de configuração do Escravo EtherCAT fornece informações de estado (por exemplo, 'Executando', 'Parado') e mensagens de diagnóstico específicas do dispositivo e do sistema de barramento interno. A guia Informação, presente no editor de configuração do Escravo EtherCAT, exibe, caso disponíveis, as seguintes informações gerais para o módulo: Nome, Fornecedor, Tipo, Número de Versão, Categorias, Número de Ordem, Descrição, Imagem.

EtherNet/IP

O EtherNet/IP é um protocolo de arquitetura mestre-escravo, o qual consiste de 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.

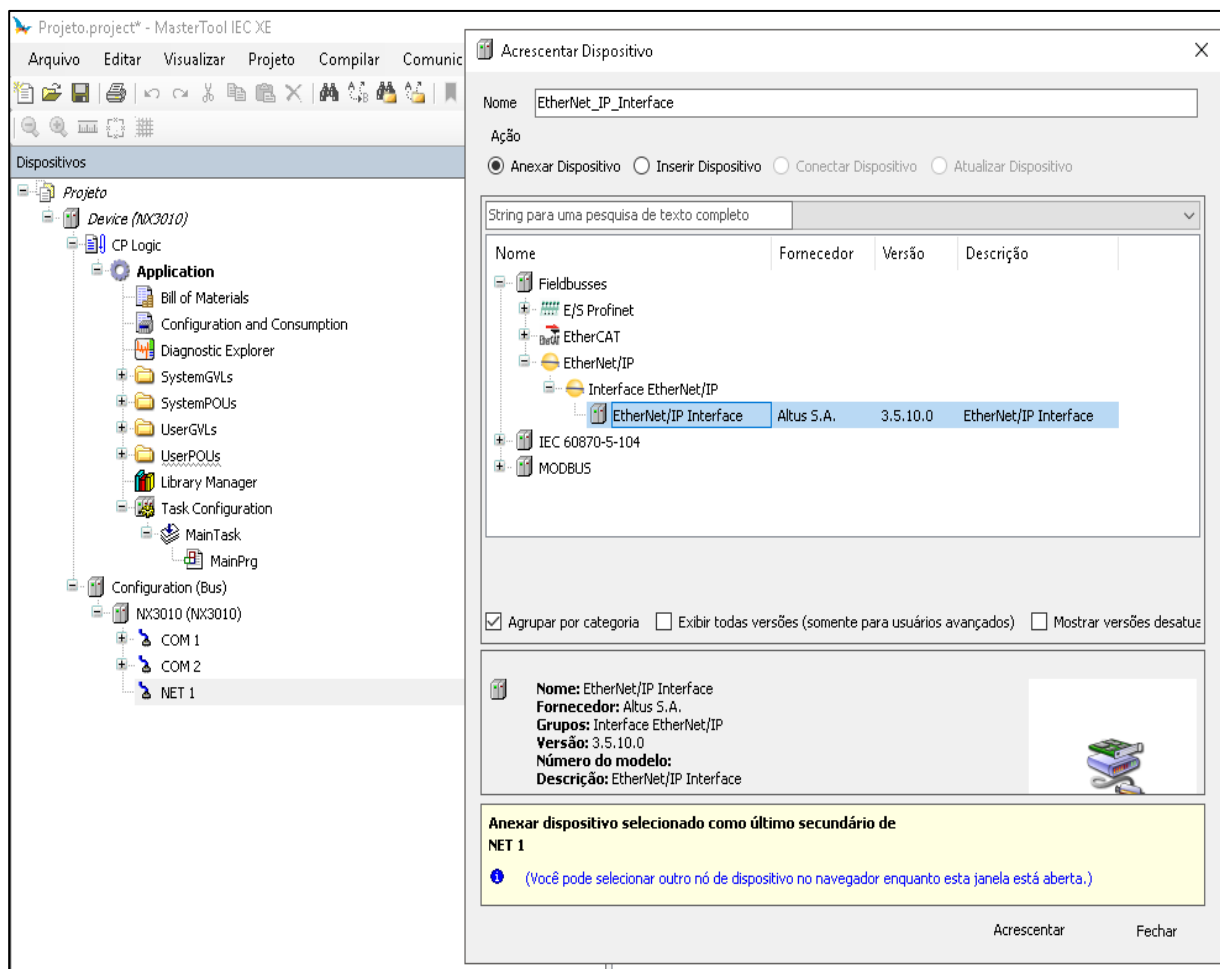


Figura 2-54. Adicionando uma Interface EtherNet/IP

Abaixo desta interface EtherNet/IP é possível adicionar um Scanner ou um Adapter conforme mostrado na figura a seguir.

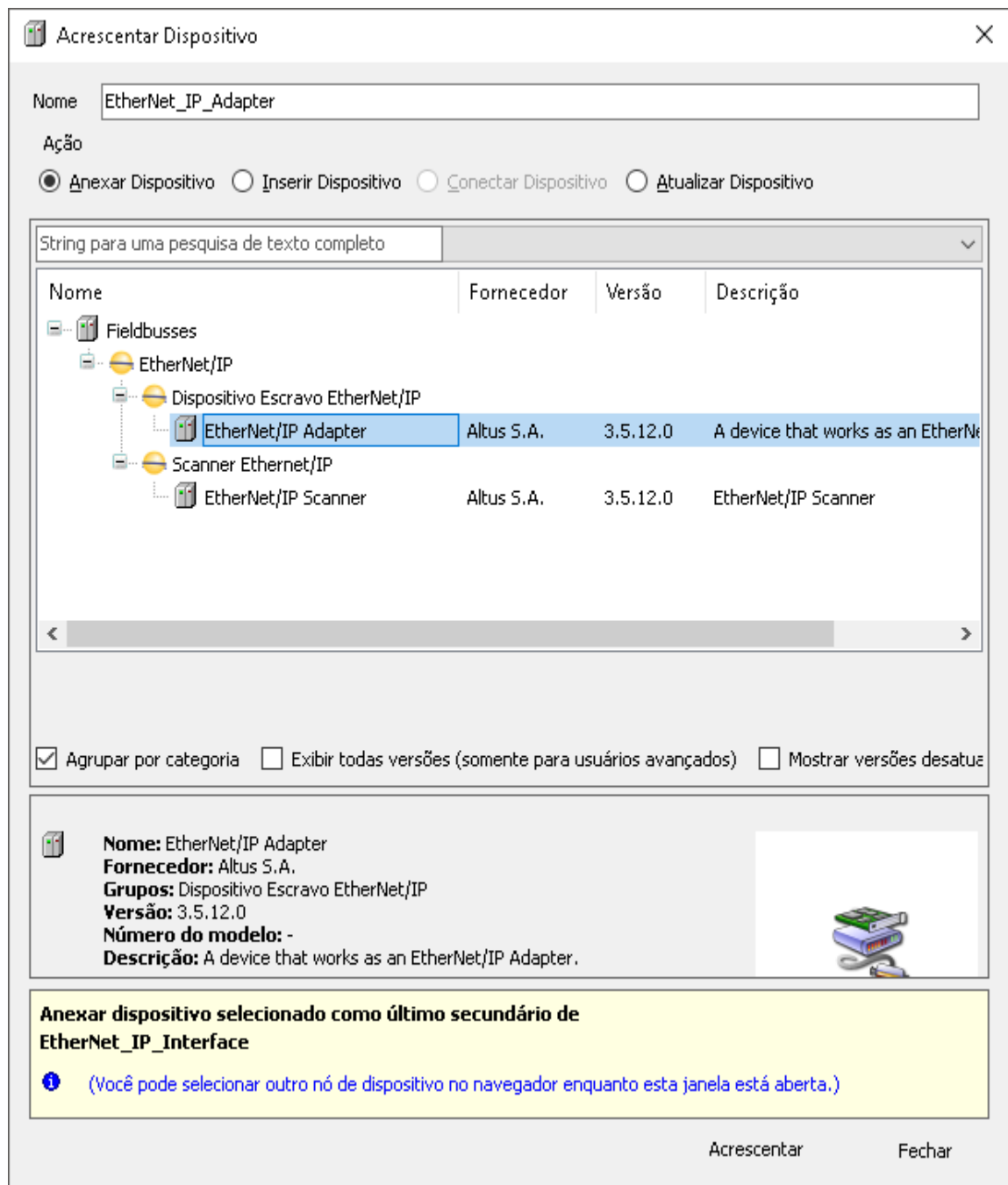


Figura 2-55. Adicionando um Adapter ou Scanner EtherNet/IP

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.

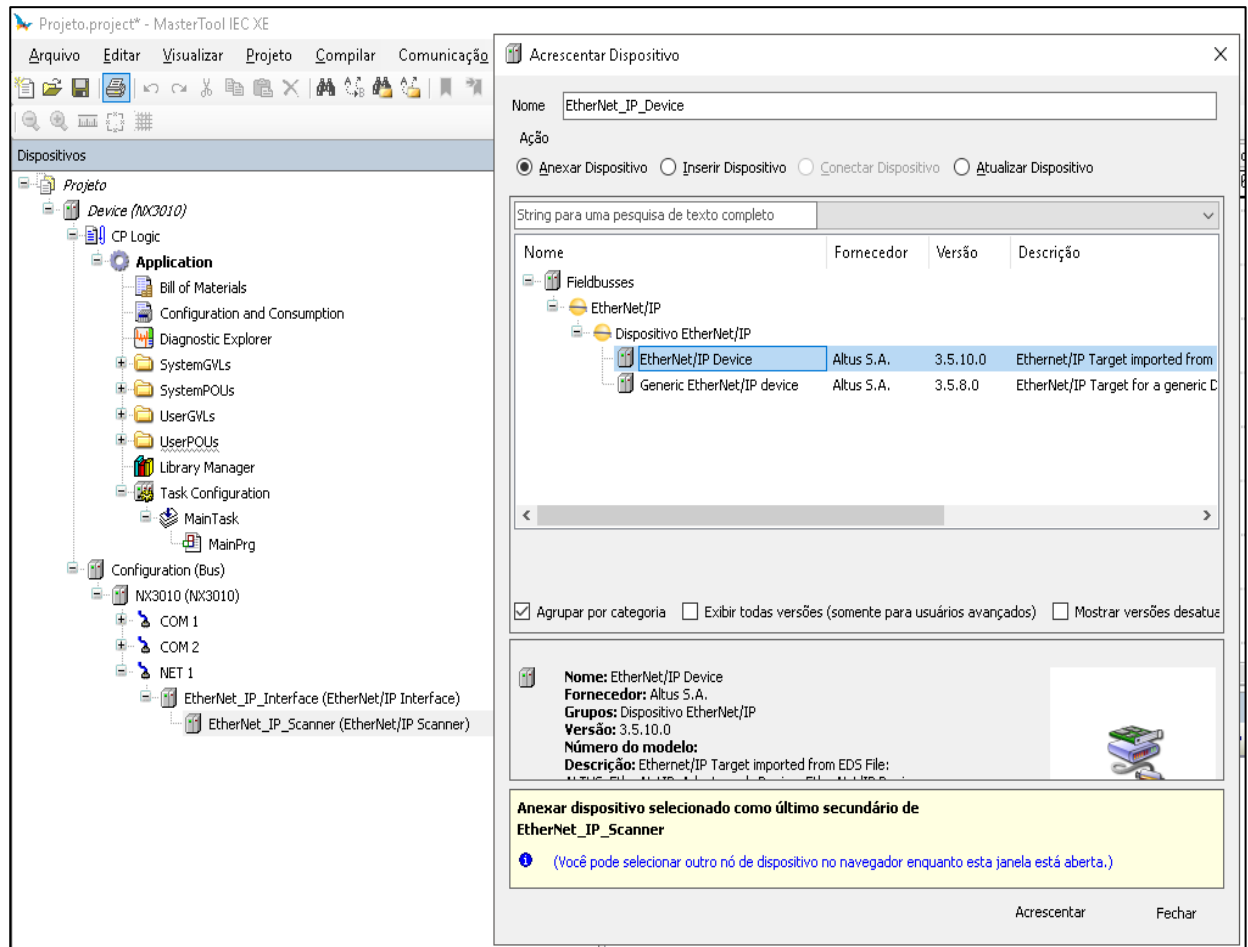


Figura 2-56. Adicionando um Adapter EtherNet/IP sob o Scanner

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.

Configuração do Adapter EtherNet/IP

O Adapter EtherNet/IP requer módulos EtherNet/IP. Os módulos vão prover mapeamentos de E/S que podem ser manipulados pela aplicação do usuário pelos endereços %I ou %Q de acordo com sua configuração (INPUT BYTE, OUTPUT BYTE etc.). Veja a figura a seguir.

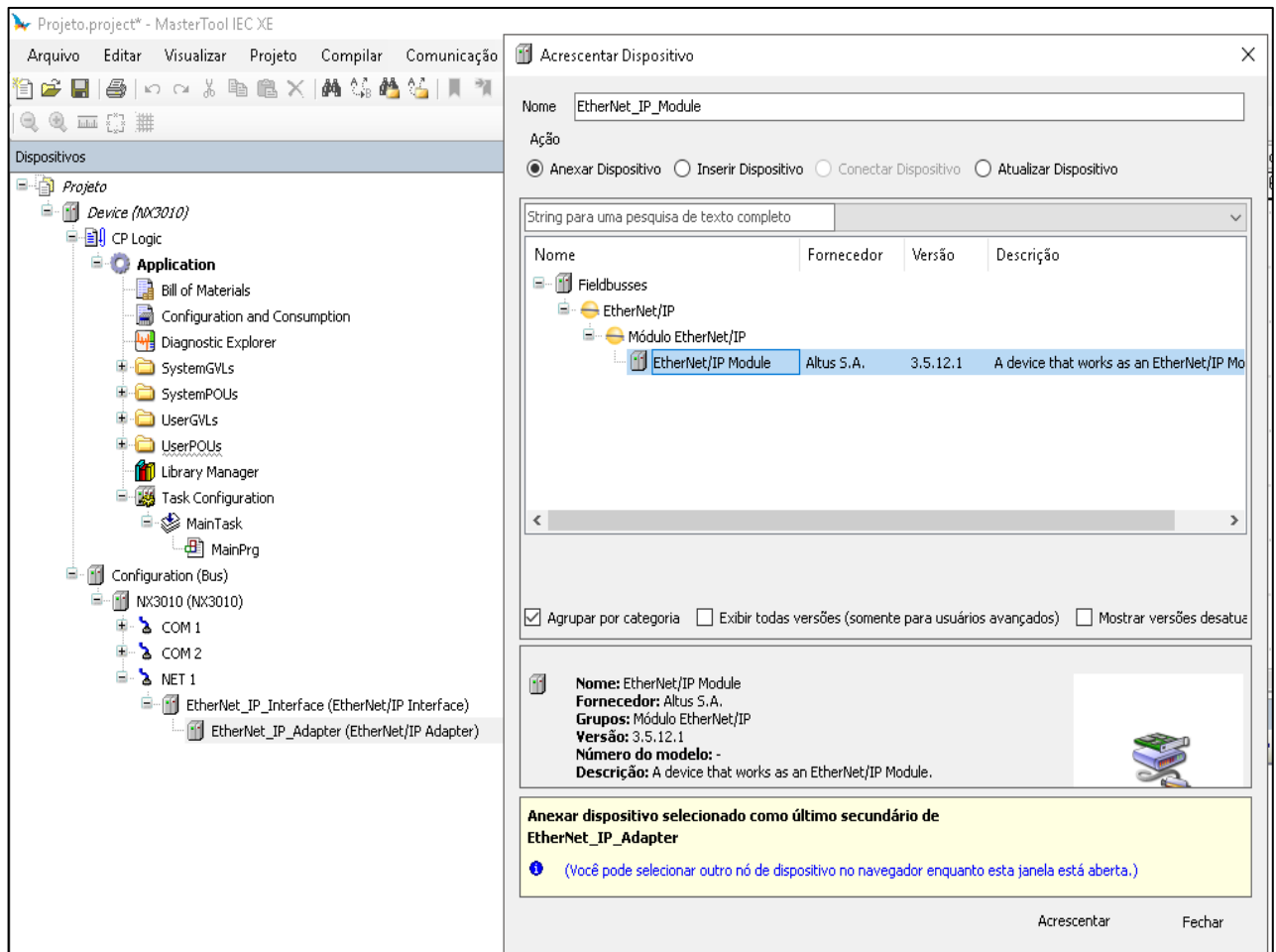


Figura 2-57. Adicionando um Módulo EtherNet/IP Abaixo do Adapter

Existem 18 módulos diferentes, os quais podem ser adicionados abaixo do Adapter. Nove de entradas e nove de saídas. Eles são do tipo BYTE, WORD, DWORD, REAL, SINT, INT, DINT e BIG. Estes tipos podem ser escolhidos na aba Geral do módulo.

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 do usuário. A opção Sempre Atualizar Variáveis deve ser mantida com o valor padrão Ativado 1.

IEC 60870-5-104 Servidor

Este protocolo está disponível para a UCP da Série Nexto, modelos NX3030, NX3020 e NX3005, nos seus canais 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 a um dos canais Ethernet disponíveis (NET 1 ou NET 2);
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 os módulos de entrada e saída digital da Série Nexto não suportam o mapeamento de pontos DBP, são necessárias algumas artimanhas de aplicação para tornar isto possível. Lembrando que também não é possível a utilização da função PulsedCommand, definido na biblioteca LibRtuStandard, para acionamento dos pontos digitais duplos na Série Nexto.

A figura a seguir ilustra a tela de Parâmetros Gerais do Servidor IEC 60870-5-104.

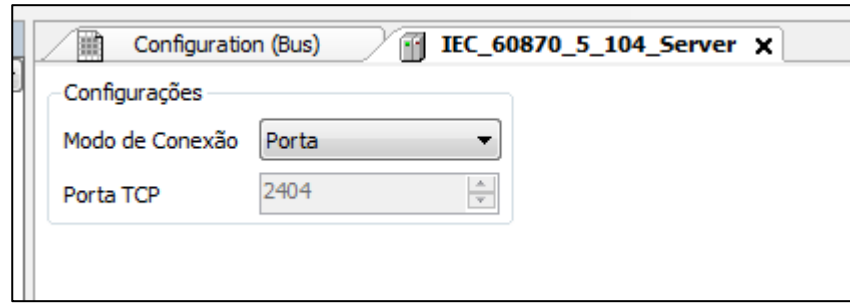


Figura 2-58. Tela de Parâmetros Gerais do Servidor IEC 60870-5-104

Na sequência está mostrada a tela de Mapeamento dos Dados do Servidor IEC 60870-5-104.

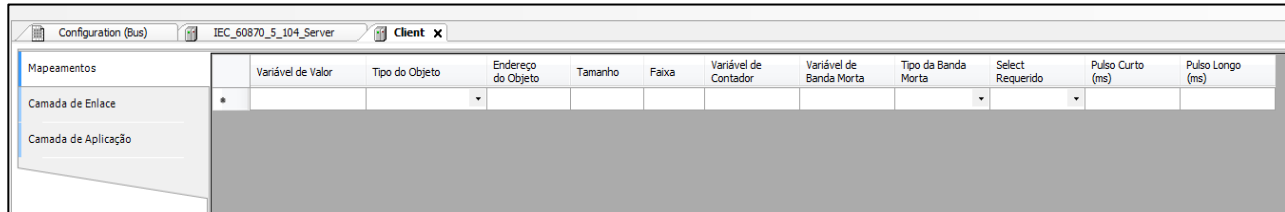


Figura 2-59. Tela de Mapeamento dos Dados do Servidor IEC 60870-5-104

A tela de configuração da camada de enlace do Servidor IEC 60870-5-104 está mostrada a seguir.

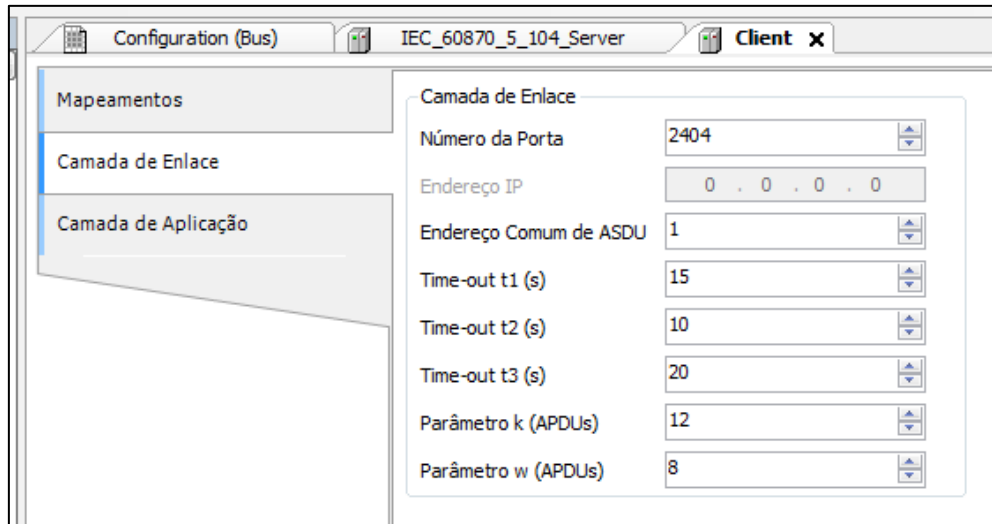


Figura 2-60. Tela de Configuração da Camada de Enlace do Servidor IEC 60870-5-104

A tela de configuração da camada de aplicação do Servidor IEC 60870-5-104 está mostrada a seguir.

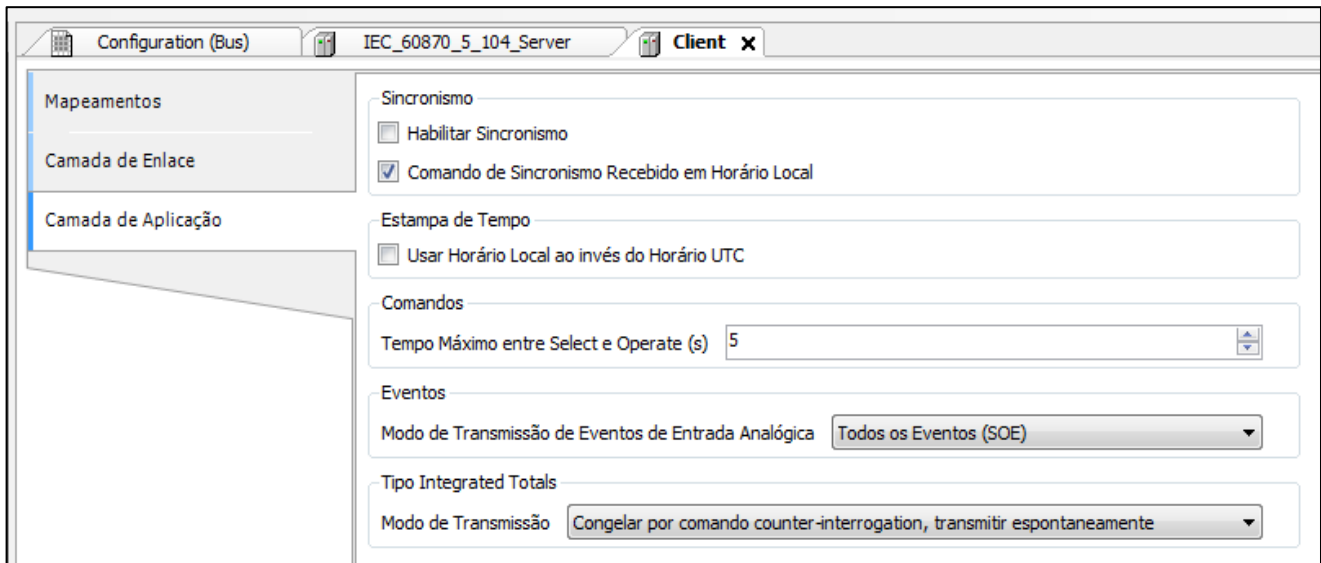


Figura 2-61. Tela de Configuração da Camada de Aplicação do Servidor IEC 60870-5-104

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. Cada tipo de objeto tem um comportamento específico para cada qualificador de comando.



Pesquisa na Documentação: outros protocolos das interfaces Ethernet

Consulte, na documentação do produto, o detalhamento das configurações dos protocolos abordados nessa seção: EtherCAT, Ethernet/IP e IEC 60870-5-104.

Fonte: Manual de Utilização UCPs Série Nexto NX30xx – MU214100

Protocolo PROFIBUS na Série Nexto



Um pouco de teoria... Protocolo Profibus

Profibus é hoje um dos Standards de rede mais empregados no mundo. Esta rede foi concebida a partir de 1987 em uma iniciativa conjunta de fabricantes, usuários e do governo alemão. A rede está padronizada através da norma DIN 19245 incorporada na norma europeia Cenelec EN 50170. No âmbito do Fieldbus, o padrão Profibus subdivide-se em duas famílias: Profibus-DP e Profibus-PA. Profibus é uma rede multimestres. A especificação fieldbus distingue dois tipos de dispositivos:

- Dispositivo Mestre - Um mestre é capaz de enviar mensagens independente de solicitações externas quando tiver a posse do token. São também chamados de estações ativas.
- Dispositivos Escravos - Não possuem direito de acesso ao barramento e podem apenas confirmar o recebimento de mensagens ou responder a uma mensagem enviada por um mestre. São também chamadas de estações passivas. Sua implementação é mais simples e barata que a dos mestres.

Existem atualmente três níveis físicos que definem os métodos de transmissão disponíveis para o Profibus:

- RS-485 para aplicações gerais da automação da manufatura.
- IEC 1158-2 para uso na automação de processos
- Fibra ótica para maior imunidade a ruído e maiores distâncias

Profibus DP

O protocolo DP utiliza as camadas 1 e 2 e a camada de usuário. Esta arquitetura otimizada assegura uma transmissão de dados eficiente e rápida. O Profibus-DP foi desenvolvido em 1994, para fazer a comunicação entre os sistemas de controle (controladores) e os elementos de campo através da configuração mestre-escravo. O sistema pode ser configurado como mono-master (apenas um mestre) ou multi-master (com vários mestres), sendo que neste último, as entradas podem ser lidas por todos os mestres e cada mestre acionando apenas suas respectivas saídas. A topologia utilizada é em linha, utilizando o par trançado ou fibra ótica como meio físico. A transmissão dos dados é feita através de RS485 e a taxa de transmissão está relacionada com a distância do cabo (9,6 Kbits - 1200m, 500 Kbits - 400m, 12000 Kbits - 100m, por exemplo). O sistema comporta 32 estações sem a utilização de repetidores e até 127 estações com a utilização de repetidores. Quando do término do meio físico da rede, ela necessita da colocação de um terminador de rede (resistor de terminação), responsável por garantir a imunidade a ruídos e determinar o final da rede.

As topologias de rede mais utilizadas são: Estrutura em Linha (Daisy Chain) e Linha com Derivações Lineares.

Fibra ótica pode ser utilizada para aumentar a imunidade ao ruído ou para alcançar maiores distâncias. Segmentos Profibus utilizando fibra ótica como meio físico devem adotar uma topologia em estrela ou anel. Alguns fabricantes oferecem ainda redes redundantes com a troca automática de rota em caso de falha. Existem também acopladores entre rede de fibra ótica e RS485, o que permite trocar de meio de transmissão sempre que desejado.

O mecanismo de acesso ao meio é implementado pela camada 2, que no caso do Profibus é denominado Fieldbus Data Link (FDL). A camada MAC no Profibus opera segundo dois princípios básicos:

- Na comunicação entre sistemas de automação complexos (mestres) deve-se buscar que cada estação tenha tempo suficiente para realizar suas tarefas de comunicação dentro

de intervalos de tempo estabelecidos. Para este tipo de comunicação é adotado o protocolo token passing.

- Na comunicação cíclica entre um mestre tal como um CLP e seus periféricos (escravos), a transmissão deve ser a mais simples e rápida possível. Neste tipo de transação utiliza-se o protocolo mestre-escravo.

Profibus PA

Já o Profibus-PA foi desenvolvido em 1995, de acordo com a norma IEC 1158-2, a qual é utilizada na automação e controle de processos contínuos, principalmente no setor químico e petroquímico. O Profibus-PA permite que os sensores e atuadores sejam conectados a rede de comunicação mantendo a segurança intrínseca dos elementos requerida pelo processo. No Profibus-PA cada segmento possui apenas uma fonte de alimentação e todo o equipamento possui consumo constante de corrente. No Profibus-PA são permitidas as topologias em linha, estrela ou árvore. A rede Fieldbus PA é uma rede para interligar válvulas, transmissores de pressão diferencial etc., portanto geralmente dispositivos escravos. A alimentação dos dispositivos pode se dar pela própria rede. Caso se deseje interligar esta rede de baixa velocidade a uma rede de alta velocidade (DP) ou a um CLP, deve-se utilizar um acoplador. O protocolo é muito simples, o que facilita a interoperabilidade. A distribuição do controle depende sempre de um mestre externo. O mestre deve ler as PVs dos transmissores, executar os algoritmos de controle e definir a abertura da válvula de controle. A Profibus PA permite ligar 32 dispositivos por segmento sem segurança intrínseca (IS) ou até 9 dispositivos com segurança intrínseca (Ex ia/ib). Os dispositivos podem ser conectados e desconectados para manutenção com a rede em operação, mesmo quando operando em áreas classificadas.

A rede Profibus-PA obedece ao padrão IEC 1158-2 que utiliza como meio de transmissão um par trançado blindado, e apresenta a velocidade de 31.25 kbit/s. Este padrão de nível físico é o mesmo da rede H1 da Foundation Fieldbus. Ele permite alimentar os dispositivos diretamente, usando o barramento de dois fios e apresenta segurança intrínseca. Esta rede pode ser usada em áreas classificadas e atende ao modelo FISCO (*Fieldbus Intrinsically Safe Concept*) definido pelo Federal Physical Technical Institute da Alemanha.

Os princípios fundamentais são:

- Cada segmento deve ter uma única fonte de potência: a fonte de alimentação.
- Nenhuma potência é alimentada no barramento quando a estação está enviando dados.
- Cada dispositivo de campo consome uma potência fixa conhecida em regime.
- Os dispositivos de campo funcionam como consumidores passivos de corrente.
- A terminação passiva de linha é realizada nos dois extremos da linha.
- Topologias linear, árvore e estrela são permitidas.

As topologias de rede mais utilizadas são: Estrutura em Linha (Daisy chain), Linha com Derivações Lineares e Estrutura em Árvore. Para se determinar o comprimento máximo da linha, vários fatores devem ser analisados, mas uma regra básica seria calcular a potência necessária a cada dispositivo a ser conectado e a classificação da área de processo.

O perfil PA suporta a intercambialidade e interoperabilidade de dispositivos de campo PA de diferentes fornecedores.

As funções parâmetros de cada dispositivo são descritas através modelo de blocos de função. Os parâmetros de entrada e saída dos blocos de função são utilizados diretamente pelas aplicações.

Existem duas maneiras de se realizar a conexão da rede Profibus PA à rede Profibus DP: via acoplador de segmento e via DP/PA link. Acopladores são conversores de sinais que adaptam os sinais RS-485 para o nível de sinal do IEC 1158-2, não possuem endereço de rede e permitem endereçar os dispositivos das sub-redes diretamente. Sua maior desvantagem é limitar a

velocidade da rede DP a 93.75 kbps. Alguns fornecedores do mercado limitam esta velocidade para um valor ainda menor.

Já o link é um equipamento que se conecta na rede DP a 12 Mbps e na Profibus PA na velocidade nominal da rede H1 (31.25 kbps). O link possui um endereço na rede DP e é um dispositivo inteligente. Eles representam todos os dispositivos conectados à rede IEC 1158-2 como um único escravo no segmento RS-485.

O Profibus define uma folha de dados eletrônica denominada General Slave Data (GSD) que são proporcionados pelo fabricante do dispositivo Profibus. O GSD se divide em três partes: especificações gerais, informações relacionadas ao mestre (para dispositivos mestres), informações relacionadas ao escravo. As especificações gerais definem informações do fabricante, velocidade de comunicação, pinagem de conectores etc. As especificações do mestre definem o número máximo de escravos permitidos e opções de upload e download. As especificações do escravo definem os parâmetros do escravo: número e tipo de canais de E/S, especificação de textos de diagnósticos etc. Um editor de GSDs está disponível no site oficial da rede Profibus. GSDs são visíveis até o nível de controle e são usados pelas ferramentas de configuração para visualizar os dados do instrumento.

Fonte: www.profibus.org.br.

Mestre PROFIBUS DP NX5001

O Mestre PROFIBUS DP NX5001 (ver figura a seguir) é um avançado mestre de interface desenvolvido para ser usado juntamente com as UCPs da Série Nexto. O módulo suporta o protocolo de troca de dados cíclicos e pode ser usado em qualquer Bastidor da Série Nexto. A interface permite acesso das UCPs da Série Nexto a qualquer dispositivo de campo compatível com esse protocolo, tais como sistemas modulares de E/S remoto, sensores, transmissores.



Figura 2-62. Mestre PROFIBUS DP NX5001

Não é necessário o uso de software adicional para a parametrização dos módulos que compõem a rede PROFIBUS, nem a exigência de um cabo especial, pois todas as configurações e parametrizações são feitas, de forma simples e fácil, diretamente no programador MasterTool

IEC XE e enviadas ao Mestre PROFIBUS DP NX5001 via UCP Nexto, sendo necessário que os arquivos de configuração (GSD) de todos os dispositivos a serem conectados ao mestre NX5001 estejam disponíveis e instalados.



DICA: informações adicionais sobre instalação (montagem mecânica, instalação elétrica, instalação na rede PROFIBUS), troca do módulo a quente, descrição técnica, configuração, diagnósticos, operação, manutenção e instalação de arquivos GSD podem ser obtidas no Manual de Utilização Mestre PROFIBUS DP NX5001- MU214001.

A figura a seguir ilustra uma configuração PROFIBUS simples.

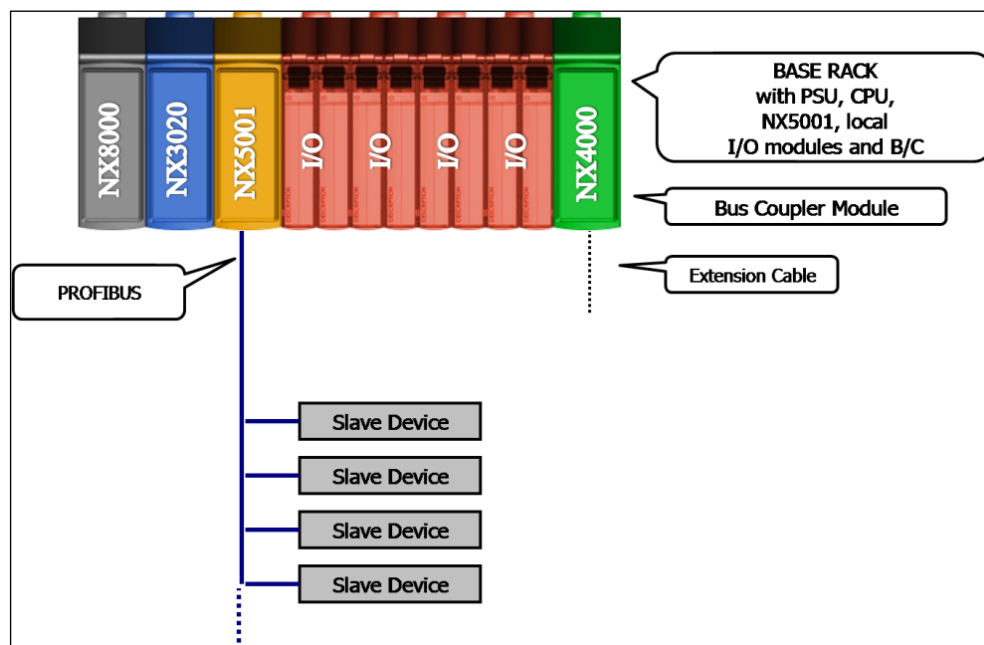


Figura 2-63. Configuração PROFIBUS simples

Diagnósticos

O mestre NX5001 fornece um diagnóstico completo da rede PROFIBUS, bem como diagnósticos gerais de funcionamento do módulo. Os diagnósticos são subdivididos em quatro grandes grupos:

1. Diagnósticos Gerais do Módulo... Fornecem ao usuário informações sobre o comportamento do módulo Mestre PROFIBUS DP NX5001.
2. Diagnósticos do Mestre PROFIBUS DP... Informam a situação geral da rede sob o ponto de vista do mestre, com um panorama geral das estações ativas ou em erro na rede PROFIBUS.
3. Diagnóstico de Redundância de Rede... Somente é gerado se houver operação redundante.
4. Diagnósticos dos Dispositivos Escravos... Diagnóstico gerado, por exemplo, pelos escravos PROFIBUS, cabeças PROFIBUS da Série PONTO e demais escravos da rede.

Os diagnósticos dos dispositivos escravos mostram com detalhe a situação de erro num determinado escravo selecionado.

Para cabeças PROFIBUS da Série PONTO os diagnósticos são decodificados e exibidos no formato mapa de bits, conforme descrito nos capítulos *Diagnóstico* dos manuais das cabeças e das CTs dos módulos que compõem o barramento escravo. O programador MasterTool IEC XE disponibiliza estruturas simbólicas pré-definidas para diagnóstico das cabeças PROFIBUS da Série PONTO e seus respectivos módulos.

Para demais escravos PROFIBUS o programador MasterTool IEC XE disponibiliza os diagnósticos conforme o Formato do Frame de Diagnósticos PROFIBUS.

Comandos de Usuário

Ao adicionar um dispositivo Mestre PROFIBUS DP NX5001 à configuração do barramento, o programador MasterTool IEC XE mapeia operandos %QB (n), na aba *Nexto Bus I/O Mapping*, destinados a comandos que o usuário pode disparar para o dispositivo mestre.

Instalação Elétrica

A alimentação do Mestre PROFIBUS DP NX5001 é proveniente do módulo Fonte de Alimentação, o qual fornece tensão através da conexão ao bastidor. O aterramento do módulo é realizado através do contato entre a mola de aterramento do módulo e o bastidor. A figura a seguir ilustra o diagrama elétrico padrão dos módulos da Série Nexto instalada em um bastidor da série.

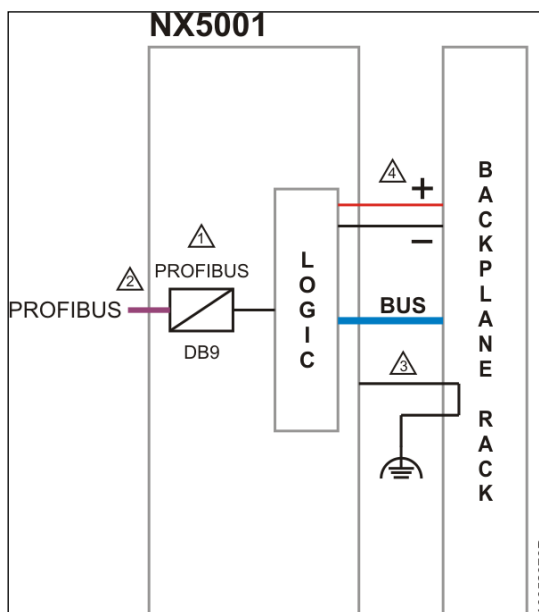


Figura 2-64. Diagrama Elétrico do Mestre PROFIBUS DP NX5001

Depois de inserido no barramento, o módulo é conectado a rede PROFIBUS de acordo com a Norma EN 50170 através do conector derivador AL-2601 (sem terminação) ou AL-2602 (com terminação) e cabo de rede AL-2302.

DICA: como erros de endereçamento em dispositivos escravos são difíceis de identificar, a rede PROFIBUS pode não detectar erros quando dois escravos estão com o mesmo endereço e separados por alguns metros de cabos de rede. Recomenda-se verificar muito bem as chaves de endereçamentos de cada dispositivo, antes de ativar a rede.

Pesquisa na Documentação: instalação de arquivos GSD

Consulte na documentação do produto o procedimento para instalação de arquivos GSD. Alguns pontos importantes para avaliar: Como selecionar o repositório de dispositivos? Como instalar dispositivos? Como adicionar o novo Escravo PROFIBUS ao NX5001?

Fonte: Manual de Utilização Mestre PROFIBUS DP NX5001- MU214001

Estudo Dirigido 2-4: configuração Profibus

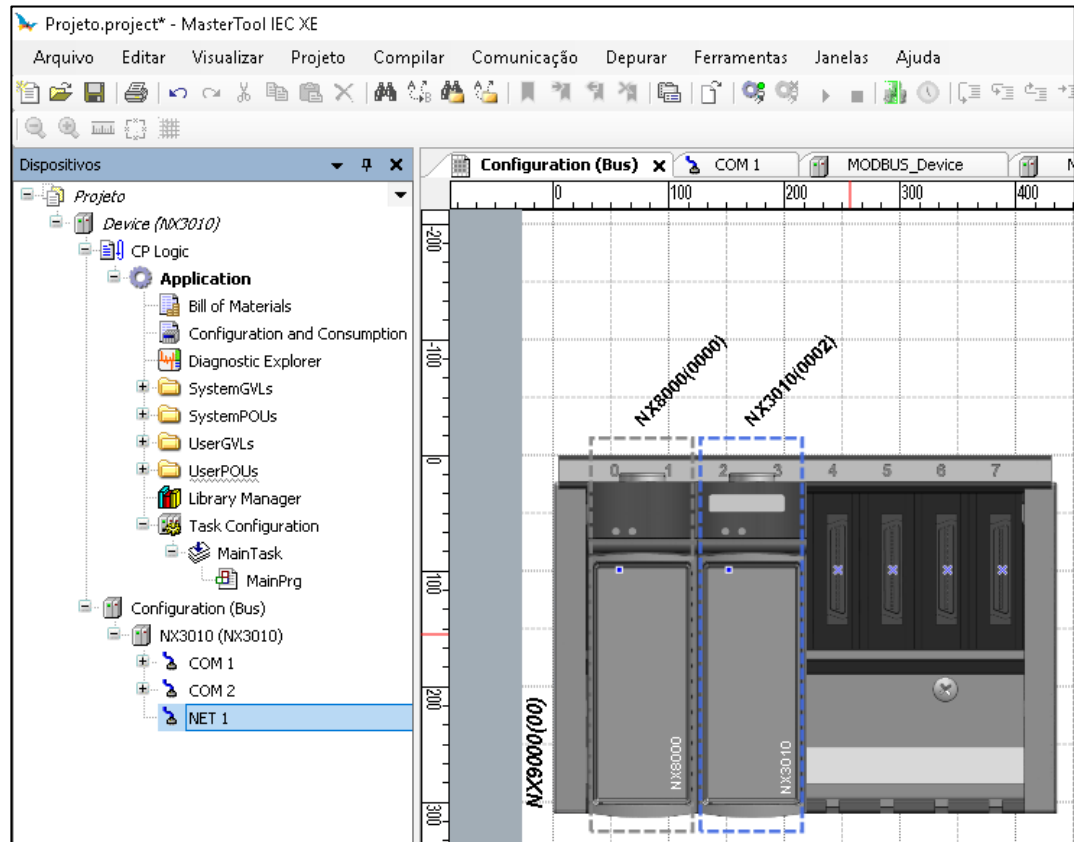
Tomando como base uma arquitetura PROFIBUS simples execute a configuração completa da rede com dois escravos PROFIBUS, sendo que o primeiro dispositivo escravo é formado por uma cabeça PO5063 e mais quatro módulos. Já o segundo dispositivo escravo é formado por uma cabeça PO5064 e mais dois módulos.

DICA: consulte o Manual de Utilização Mestre PROFIBUS DP NX5001-MU214001.

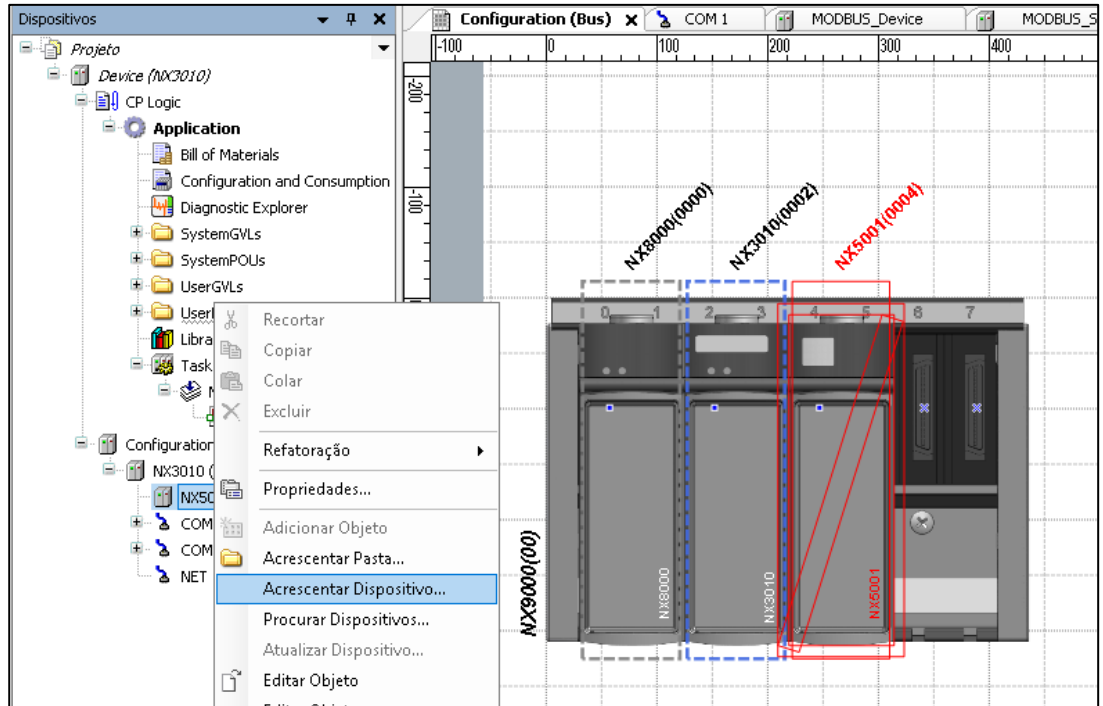
ANOTAÇÕES

Implementação do Estudo Dirigido 2-4:

Etapa 1: configuração do barramento Nexto



Etapa 2: configuração da Rede PROFIBUS



Etapa 3: parametrização da Rede PROFIBUS

The screenshot shows the configuration interface for a PROFIBUS network. The left sidebar displays a project tree with the following structure:

- Projeto
 - Device (NX3010)
 - CP Logic
 - Application
 - Bill of Materials
 - Configuration and Consumption
 - Diagnostic Explorer
 - SystemGVLS
 - SystemPOUs
 - UserGVLS
 - UserPOUs
 - Library Manager
 - Task Configuration
 - MainTask
 - MainPrg
 - Configuration (Bus)
 - NX3010 (NX3010)
 - NX5001 (NX5001)
 - POS063 (POS063)
 - PO1000_16DI_24_Vdc_
 - PO2022_16DO_NO_Dry
 - PO1212_8AI_Universal
 - PO2132_4AO_Universal
 - COM 1
 - COM 2
 - NET 1

The main configuration area is titled 'Parâmetros DP' and includes the following sections:

- Endereços**:
 - Endereço da Estação: 0
 - Maior Endereço de Estação: 125
 - Modo:
 - Modo Autolimpar
 - Inicialização
- Parâmetros**:
 - Taxa de Transmissão [kBits/s]: 12000
 - Usar Padrão

A table of parameters is displayed at the bottom:

Parâmetro	Valor	Unidade	Descrição
T_SL	1000	Bit	Tempo de slot
T_SDR Min	11	Bit	Atraso mínimo no tempo de resposta da estação
T_SDR Máx	800	Bit	Atraso máximo no tempo de resposta da estação.
T_QUI	9	Bit	Tempo de silêncio
T_SET	16	Bit	Tempo de inicialização
T_TR	6647	Bit	Tempo de rotação do dispositivo
Gap	10		Fator de atualização de Gap
Limite de Tentativas	4		Número máximo de tentativas em caso de falha
Intervalo de Escravo	6	100 µs	Intervalo de escravo mínimo
Time-out de Varredura	10	10 ms	Mínimo time-out de varredura
Tempo de Controle de Dados	600	ms	Tempo de controle de dados

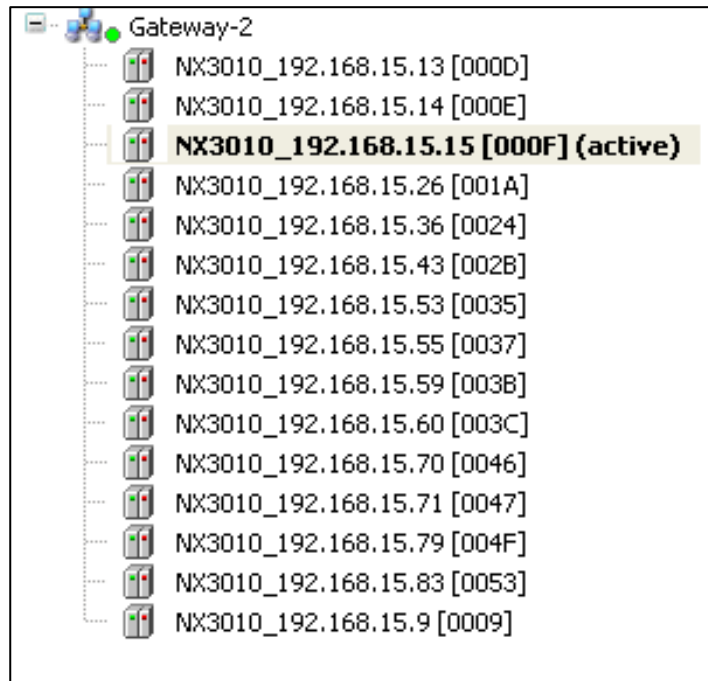
Etapa 4: mapeamento (E/S dos módulos, diagnósticos e comandos do usuário)

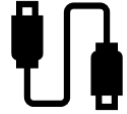
The screenshot shows the 'PO1212_8AI_Universal_Isolated' window with the 'Localizar' tab selected. A table lists variables and their mappings. The table has columns for 'Variável', 'Mapeamento', 'Canal', 'Endereço', 'Tipo', 'Unidade', and 'Descrição'. The 'Canal' column is currently set to 'Input0'. The 'Endereço' column shows addresses like %IW6, %IX6.0 through %IX7.7, and %IW8 through %IW20. The 'Tipo' column shows 'WORD' and 'BOOL'. The 'Unidade' column is empty. The 'Descrição' column is empty. Below the table, there are buttons for 'Resetar Mapeamento', 'Sempre atualizar variáveis', and 'Ativado 1 (usar tarefa de ciclo de barrament)'. There are also icons for 'Criar nova variável' and 'Mapear para variável existente'.

Variável	Mapeamento	Canal	Endereço	Tipo	Unidade	Descrição
		Input0	%IW6			
		Word0	%IW6	WORD		
		Bit0	%IX6.0	BOOL		
		Bit1	%IX6.1	BOOL		
		Bit2	%IX6.2	BOOL		
		Bit3	%IX6.3	BOOL		
		Bit4	%IX6.4	BOOL		
		Bit5	%IX6.5	BOOL		
		Bit6	%IX6.6	BOOL		
		Bit7	%IX6.7	BOOL		
		Bit8	%IX7.0	BOOL		
		Bit9	%IX7.1	BOOL		
		Bit10	%IX7.2	BOOL		
		Bit11	%IX7.3	BOOL		
		Bit12	%IX7.4	BOOL		
		Bit13	%IX7.5	BOOL		
		Bit14	%IX7.6	BOOL		
		Bit15	%IX7.7	BOOL		
		Word1	%IW8	WORD		
		Word2	%IW10	WORD		
		Word3	%IW12	WORD		
		Word4	%IW14	WORD		
		Word5	%IW16	WORD		
		Word6	%IW18	WORD		
		Word7	%IW20	WORD		

Etapa 5: carga da configuração PROFIBUS na UCP via canal Ethernet

The screenshot shows the 'Projeto.project* - MasterTool IEC XE' window. The 'Dispositivos' tree on the left shows the project structure, including 'Projeto', 'Device (NX3010)', 'CP Logic', 'Application', 'Bill of Materials', 'Configuration and Consumption', 'Diagnostic Explorer', 'SystemGVLs', 'SystemPOUs', 'UserGVLs', 'UserPOUs', 'Library Manager', 'Task Configuration', 'MainTask', 'MainPrg', 'Configuration (Bus)', 'NX3010 (NX3010)', and 'NX5001 (NX5001)'. The main window displays the 'Configurações de Comunicação' dialog box. The 'Seleção o caminho de rede para o controlador' section shows a list of network paths, with 'Gateway-1' selected. The 'Gateway-1' section shows a list of network paths, with 'Gateway-1 (Mapeando...)' selected.





3. Recursos avançados no CP Nexto

Memórias Não-Voláteis

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

Para aplicações que necessitam funcionalidades de memória não volátil a Série Nexto possibilita a utilização de variáveis de representação direta de memória retentivas (%M), memória retentiva de dados, memória persistente de dados, memória de programa (para armazenamento do programa do usuário), memória de código fonte, memória de web Server, memória de arquivos de usuário (Doc, pdf, data) e interface para cartão de memória.



NOTA: A área da memória de código fonte é utilizada como backup do projeto. Se o usuário desejar importar a sua aplicação, o software MasterTool IEC XE buscará as informações necessárias nessa área. É importante salientar que o usuário deve estar atento para não se esquecer de atualizar o projeto que está salvo como backup, evitando que informações importantes sejam perdidas.

Na sequência vamos explorar essas funcionalidades.

Para mais informações sobre a utilização dessas características do produto consultar o Manual de Utilização Série Nexto – MU214000.

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 utilizado.

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

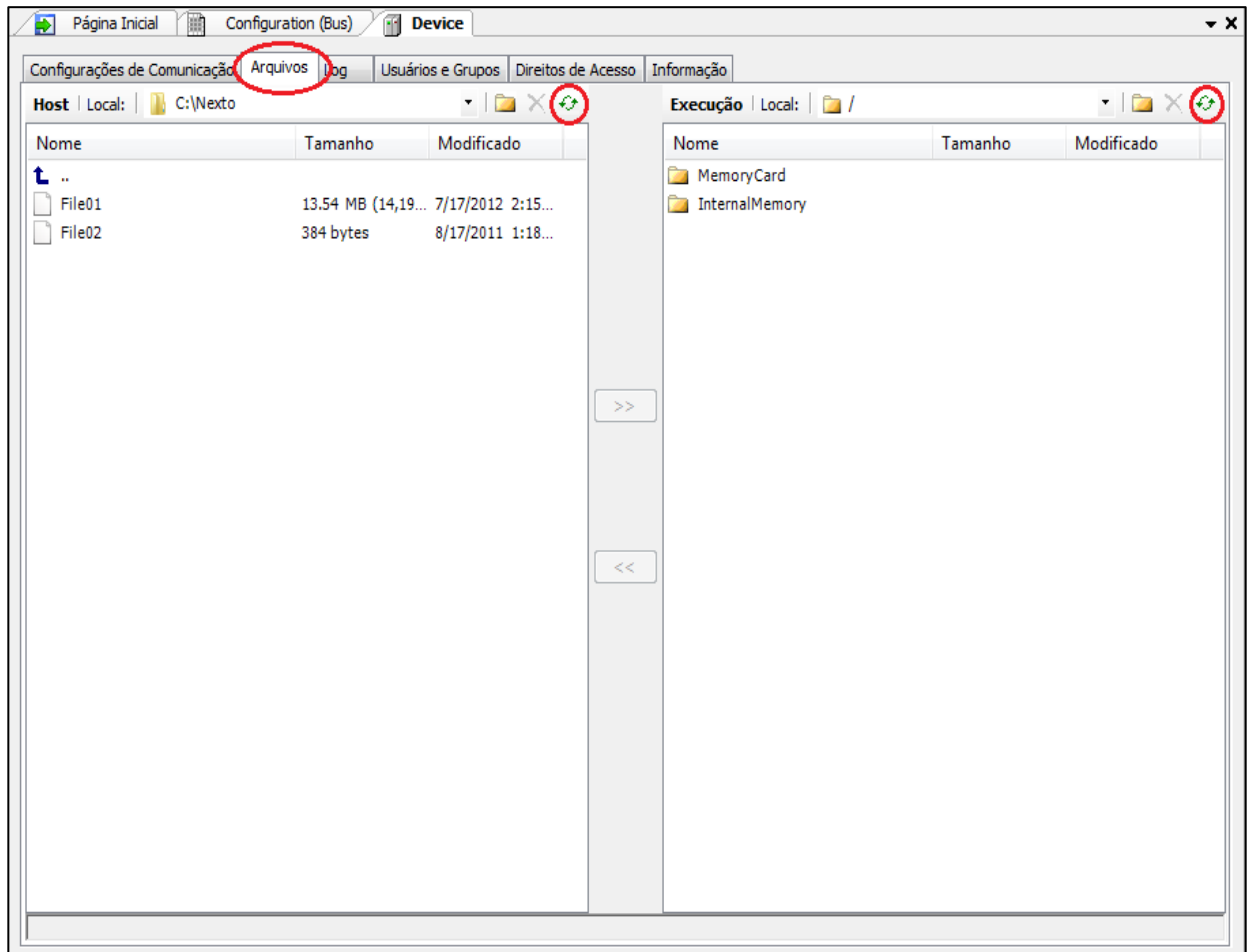


Figura 3-1. Acesso aos Arquivos de Usuário

Após atualizar a coluna de arquivos da UCP, será exibido o diretório raiz de arquivos armazenados na UCP e poderá ser selecionada a pasta para onde os arquivos serão transferidos. A pasta “InternalMemory” é uma pasta padrão, a ser utilizada para armazenar arquivos na memória interna da UCP, uma vez que não é possível transferir arquivos para o diretório raiz. Caso seja necessário, podem ser criadas outras pastas no diretório raiz ou subpastas dentro da pasta “InternalMemory”. Já a pasta “MemoryCard” é o diretório onde o cartão de memória estará montado, caso ele esteja inserido na UCP. Arquivos transferidos para a pasta “MemoryCard” serão transferidos diretamente para dentro do cartão de memória. Conforme novas funcionalidades forem sendo adicionadas ao produto, algumas pastas podem aparecer e que devem ser ignoradas pelo usuário.



NOTA: No caso em que o cartão de memória seja inserido após a inicialização da UCP, será requisitado um usuário e senha para realizar o acesso e/ou transferências de arquivos do MasterTool IEC XE para o cartão de memória ou vice-versa. O usuário padrão com privilégios para acesso à UCP é “Owner” e a senha padrão desse usuário é “Owner”.

Para realizar a transferência de algum arquivo do microcomputador para a UCP, basta selecionar o arquivo desejado na coluna da esquerda e pressionar o botão “»”, localizado no centro da tela, conforme figura abaixo. O tempo de transferência irá variar de acordo com o tamanho do arquivo e com o tempo de ciclo (execução) da aplicação atual da UCP, podendo levar vários minutos. O usuário não precisa estar em Modo Run ou conectado à UCP para realizar as transferências, pois ela possui a capacidade de se conectar automaticamente quando o usuário realizar a transferência.

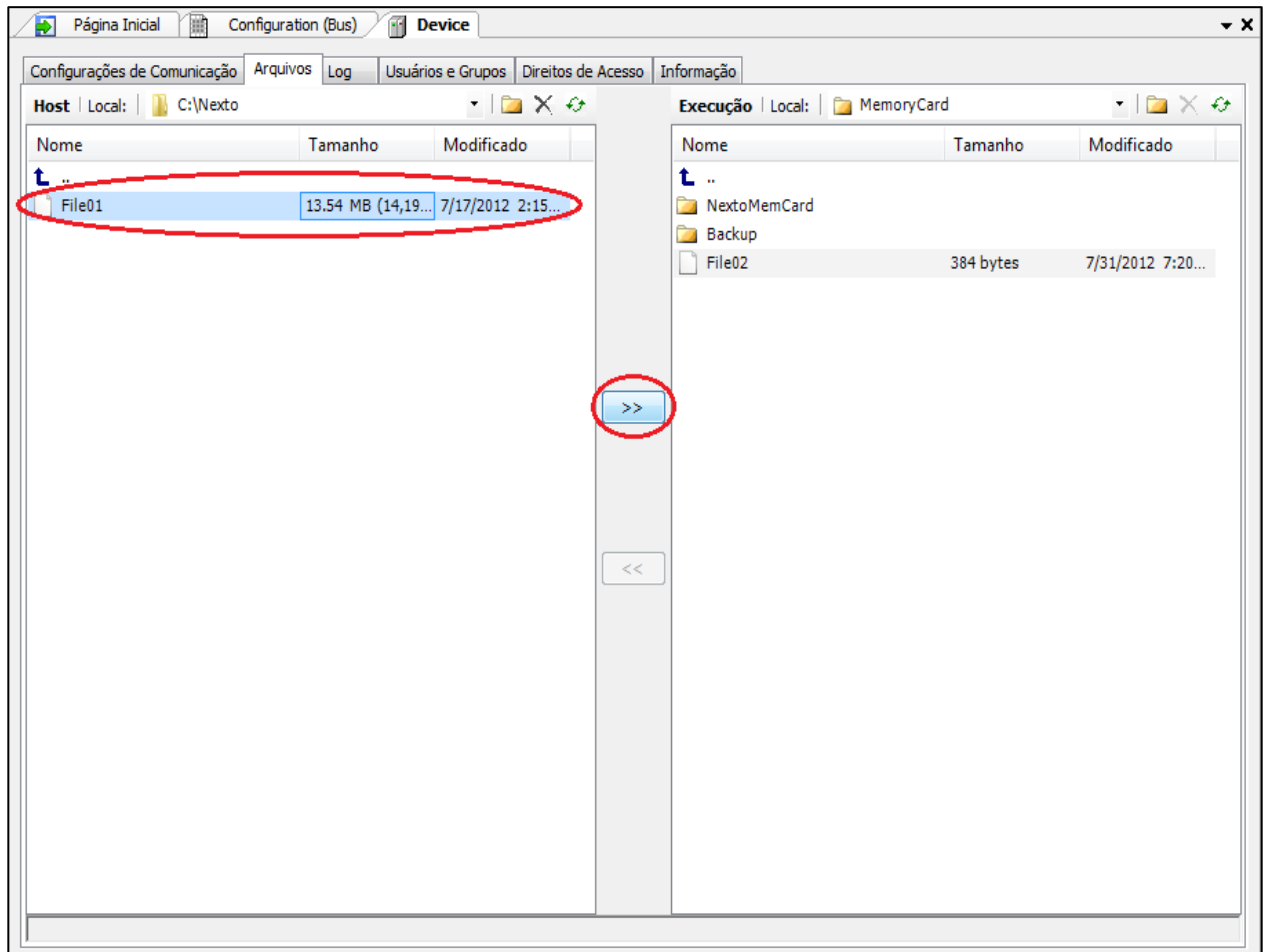





Figura 3-2. Transferindo Arquivos

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

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

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

- Novo diretório : permite a criação de uma nova pasta na área de memória de usuário.
- Excluir item : permite a exclusão de arquivos os diretórios da área de memória de usuário.
- Atualizar : permite atualizar, na tela do MasterTool IEC XE, os arquivos presentes na memória de usuário.

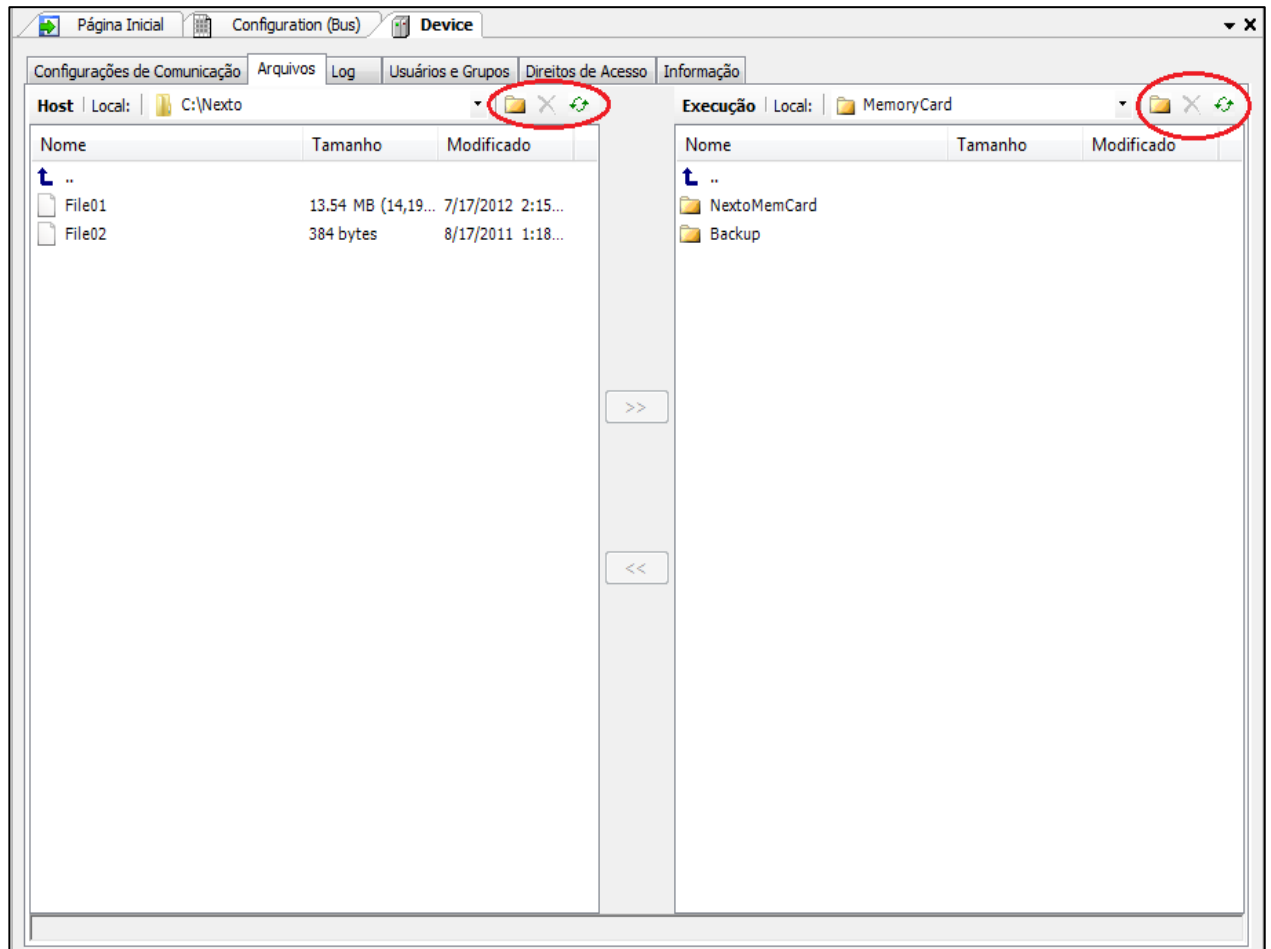


Figura 3-3. Opções de Utilização

Cartão de Memória (SD)

Entre outras funcionalidades, os modelos NX3010, NX3020 e NX3030 das UCPs da Série Nexto possibilitam ao usuário a utilização de um cartão de memória, o qual serve para armazenar, entre outros arquivos, o projeto e a aplicação que está na memória interna da UCP.

Quando o cartão for inserido na UCP e estiver com sistema de arquivos diferente de FAT32, ela automaticamente identifica e pergunta ao usuário se ele deseja formatar. Em caso negativo, ele não poderá utilizar o cartão (o cartão não será montado, aparecerá uma mensagem dizendo que o formato não foi reconhecido, e o visor não indicará a presença do cartão). Caso seja selecionada a opção de formatação, a UCP irá levar alguns minutos, dependendo do tempo de ciclo (execução) da aplicação rodando na UCP, para executar a operação. Assim que o cartão de memória for montado, a UCP irá ler informações gerais do mesmo, deixando o acesso ao cartão de memória mais lento nos primeiros minutos. Esse procedimento é feito apenas quando o cartão é inserido ou a UCP reinicializada. Recomenda-se realizar o procedimento de formatação do cartão de memória diretamente na UCP Nexto para evitar possíveis problemas de utilização, aumento do tempo de montagem ou até mesmo funcionamento incorreto. Não é recomendado remover o cartão de memória ou desenergizar a UCP durante a formatação ou durante a transferência de arquivos, pois pode causar a perda de dados bem como danos irreversíveis ao cartão.

Preparação do Projeto

Para utilização da funcionalidade, durante a configuração do projeto, dentro do software MasterTool IEC XE, o usuário deve habilitar a opção de cópia do projeto da UCP para o cartão de memória e cartão de memória para a UCP e configurar senhas de acesso. Estas senhas serão solicitadas pela UCP no momento da execução da respectiva transferência. Caso a UCP não tenha aplicação carregada, o Menu “Cartão de Memória” estará disponível para permitir a realização da transferência do projeto do cartão para a UCP sem que seja necessário realizar qualquer tipo de preparação prévia da UCP.

Para utilização da funcionalidade é necessário executar os passos a seguir.

1. Deve-se navegar até o menu Comunicação e executar o comando Criar Aplicação de Inicialização, lembrando que não se pode estar logado na UCP para realizar esse procedimento. Após executar esse comando, dois arquivos são criados na pasta onde o projeto está salvo. Um com a extensão “app” e outro com a extensão “crc”.
2. Após gerados os arquivos no item anterior deve-se navegar até a configuração dos Parâmetros Gerais da UCP e clicar no botão Cartão de Memória. Uma nova tela será aberta conforme a figura a seguir. Nesta tela deve-se habilitar a(s) operação(ões) de transferência desejada(s) e, caso necessário, configurar a(s) senha(s), apenas com caracteres numéricos. A utilização de senha não é obrigatória.
3. Para concluir a operação de configuração é necessário clicar no botão Localizar Arquivo... e então localizar o arquivo com extensão “.crc” gerado no passo anterior.

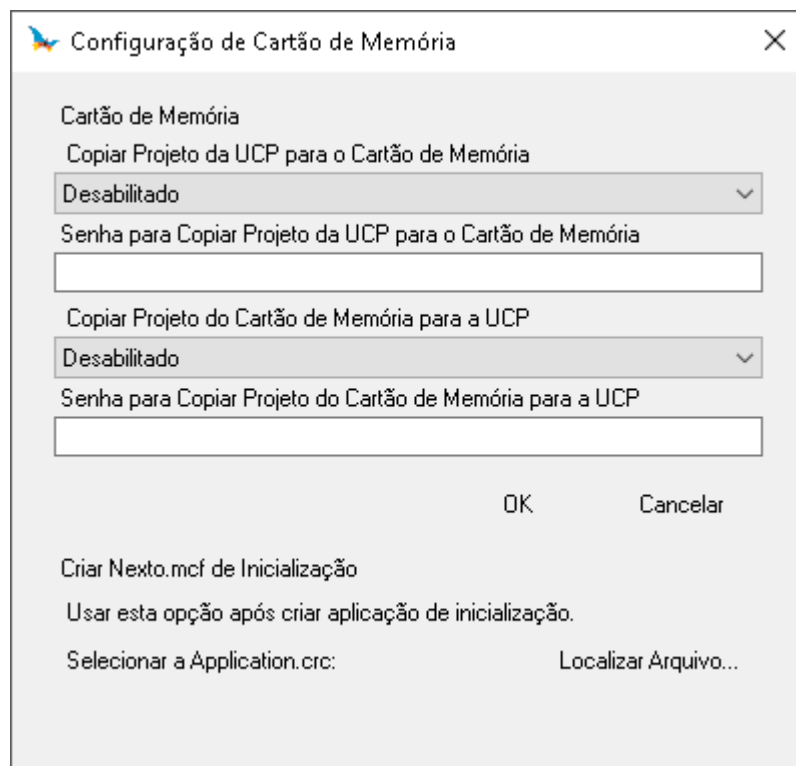


Figura 3-4. Configurações do Cartão de Memória

Executados esses passos, o MasterTool IEC XE irá enviar todos os arquivos necessários para realizar as operações de envio e recebimento de projetos via cartão de memória. Caso o cartão esteja montado, a senha será gravada no mesmo. Caso contrário, a senha configurada no MasterTool será solicitada se o usuário tentar transferir o projeto da UCP para o cartão.

Envio do Projeto

Para realizar o envio do projeto da UCP para o cartão de memória ou vice-versa, o usuário, além de habilitar no software MasterTool IEC XE o uso da funcionalidade, terá que acessar o Menu Cartão de Memória na UCP, utilizando a tecla de diagnósticos, e selecionar a opção de transferência desejada.

O envio do projeto para o cartão de memória, deve ser realizado apenas usando a tecla de diagnósticos da UCP.

Após, será solicitada a senha, caso o usuário tenha configurado durante a configuração da aplicação. Então, com um pressionamento curto na tecla de diagnósticos, os dígitos são incrementados e com um pressionamento longo são confirmados. No sexto dígito confirmado, a UCP irá consistir a senha e iniciará o processo. Após a transferência do cartão de memória para a UCP, caso haja uma aplicação em RUN ela será mantida em STOP por motivos de segurança. Para colocar a UCP em RUN, ela deve ser reinicializada.

Quando as senhas, tanto da aplicação que está na UCP quanto da aplicação que está no cartão de memória, forem iguais, não é requisitado a inserção das senhas no menu da UCP para realizar as transferências das aplicações.

Para remover o cartão de memória, basta pressionar a tecla MS com um pressionamento longo, e aguardar até que o ícone do cartão desapareça da tela de status do visor gráfico.

Caso o cartão de memória seja removido sem ser desmontado via menu da UCP, durante a transferência de arquivos, este processo pode acarretar na perda de dados no cartão, bem como corromper os arquivos nele contidos. Este processo pode implicar na necessidade de uma nova formatação do cartão quando inserido novamente à UCP.

Caso exista algum arquivo na raiz do cartão de memória nomeado "NextoMemCard" ou "Backup", ele será excluído para a criação das pastas de mesmo nome, utilizadas pela UCP para armazenamento do projeto de aplicação e do Project archive. Pastas com estes nomes não serão sobrescritas.

Acesso no MasterTool

O acesso à memória do cartão está vinculado à mesma tela de memória de usuário no software MasterTool IEC XE, sendo ele montado na pasta MemoryCard. Os diretórios NextoMemCard e Backup são criados dentro do cartão de memória toda a vez que ele for inserido na UCP. Caso eles já existam, o sistema irá reconhecer e não sobrescreverá as pastas.

Na pasta NextoMemCard no cartão de memória, encontram-se os arquivos referentes à aplicação, nesta janela você ainda tem a opção de salvar em um diretório de preferência o seu projeto (caso tenha enviado o código fonte). No MasterTool, na opção "Arquivo/ Arquivo de Projeto/ Extrair Arquivo..."pode-se abrir no MasterTool a aplicação salva, que está localizada no diretório escolhido anteriormente. A pasta Backup não é utilizada pelo usuário.

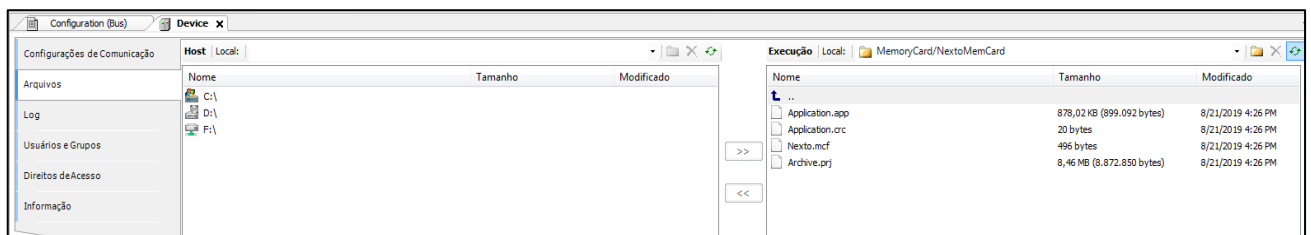


Figura 3-5. Diretório com Cartão de Memória Inserido com Projeto

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 declarada 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 App1 *)
bVarPers : BOOL; (* 2. Variável Persistente+Retentiva App1 *)
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.

Situação	Comportamento		
Após comando online	VAR	VAR RETAIN	VAR PERSISTENT VAR RETAIN PERSISTENT VAR PERSISTENT RETAIN
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

Tabela 3-1. Comportamento das Variáveis Remanentes

Gerenciamento de Usuários e Acesso

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.

Gerenciamento de Usuários

A configuração dos usuários e grupos é feita no diálogo Projeto na janela Configurações do Projeto.

Os projetos contam, automaticamente, com um grupo padrão, chamado Everyone. Todos os usuários de outros grupos são membros deste grupo específico. Assim, cada conta de usuário é fornecida automaticamente com, no mínimo, as configurações padrão. O grupo Everyone não pode ser excluído, (somente renomeado) e os seus membros não podem ser removidos.

O projeto apresenta também o chamado grupo Owner que contém o usuário Owner. Usuários podem ser acrescentados ou removidos deste grupo, mas ao menos um usuário deve permanecer. Este grupo também não pode ser excluído e sempre tem todos os direitos de acesso. Tanto o grupo Owner, como o usuário Owner podem ser renomeados.

Ao iniciar o programador e um projeto, não há nenhum usuário conectado. Entretanto, o logon do usuário pode ser realizado através de uma conta definida com nome e senha e, assim, ele pode obter direitos de acesso específicos.

Observe que cada projeto tem o seu próprio gerenciamento de usuários, por exemplo, para ter direitos de acesso específicos em uma biblioteca, o usuário deve realizar o logon nesta biblioteca separadamente. Os usuários e grupos definidos em projetos diferentes não são idênticos, mesmo que tenham nomes iguais. Por padrão, em novos projetos, a senha do usuário *Owner* é vazia.



NOTA: As senhas dos usuários são armazenadas de forma irreversível. Caso uma senha seja perdida, a respectiva conta do usuário não mais poderá ser utilizada. Se a senha Owner for perdida, o projeto inteiro pode ser inutilizado.

Usuários

Os usuários atuais registrados são listados em uma estrutura de árvore. Através dos comandos Acrescentar ou Editar é possível exibir, além do Nome (logon), o Nome completo e a Descrição do usuário. As propriedades de cada usuário podem ser visualizadas ou não (ficam ocultas) através do sinal de mais e menos respectivamente. Cada usuário, por padrão, é membro do grupo "Everyone". Para definir uma nova conta de usuário, use o botão Acrescentar para abrir o diálogo Acrescentar Usuário conforme mostrado na figura a seguir.

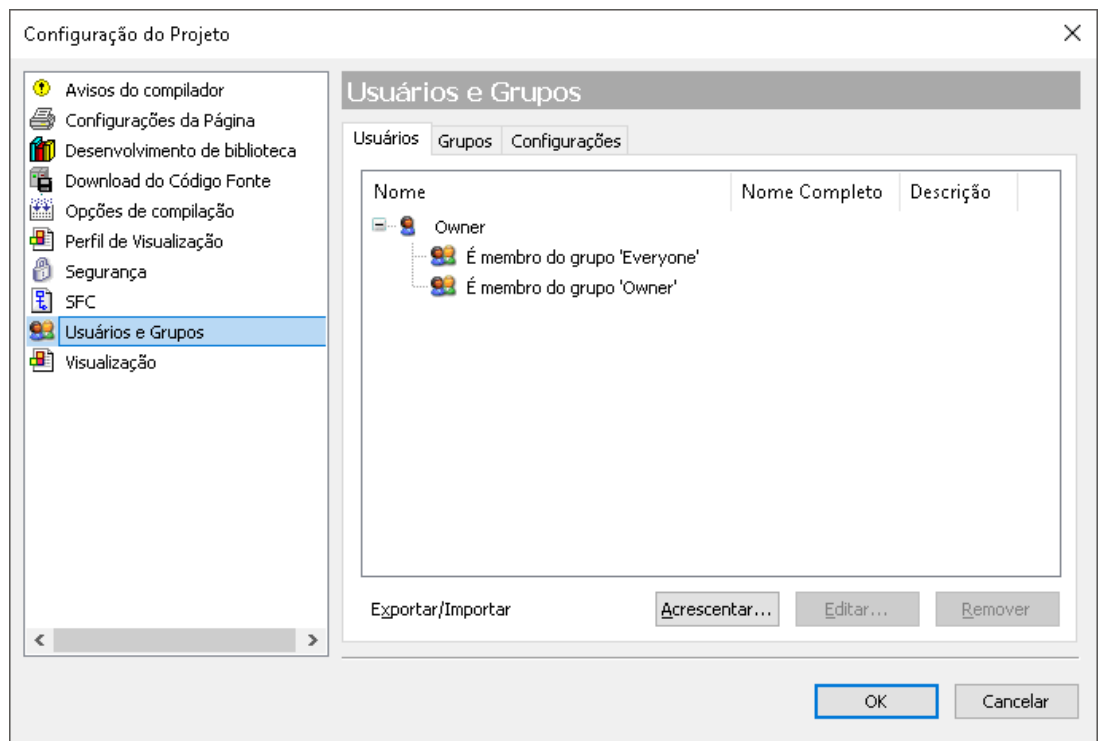


Figura 3-6. Configurações do Projeto, Diálogo Usuários

O diálogo “Propriedades das Contas” contém os campos mostrados na figura a seguir.

Acrescentar Usuário

Propriedades das Contas

Nome do logon: usuario1

Nome completo: usuario1

Descrição: usuário chefe

Senha antiga:

Senha: *****

Confirmar senha: *****

Ativo:

Membro de

- Owner

Este usuário também é membro do grupo 'Everyone'.

OK Cancelar

Figura 3-7. Acrescentar Usuário

Para configurar o novo usuário, feche o diálogo com *OK*. No caso de alguma incoerência (senha incorreta, ausência de nome de logon, usuário já existente), será apresentada uma mensagem de erro.

Para modificar uma conta de usuário existente, use o botão *Editar* e abra o diálogo correspondente. Estes campos são iguais aos do diálogo *Acrescentar Usuário*. Os campos da senha - por questões de segurança - apresentarão 32 asteriscos (*). Após ter modificado os itens desejados, feche o diálogo com *OK* para aplicar as novas configurações.

Para excluir uma ou várias contas de usuários, selecione os respectivos usuários na lista apropriada e tecla *Remover*. Note que esta ação não exige confirmação. Não é possível excluir todos os usuários do grupo (no mínimo um deve permanecer). Caso o usuário tente fazer isto, aparecerá uma mensagem de erro.

Grupos

Os grupos atuais disponíveis são listados em uma estrutura de árvore. Os membros de cada grupo podem ser visualizados ou não através do sinal de mais e menos respectivamente. Lembre-se que o membro deve ser parte de um grupo (ver figura a seguir).

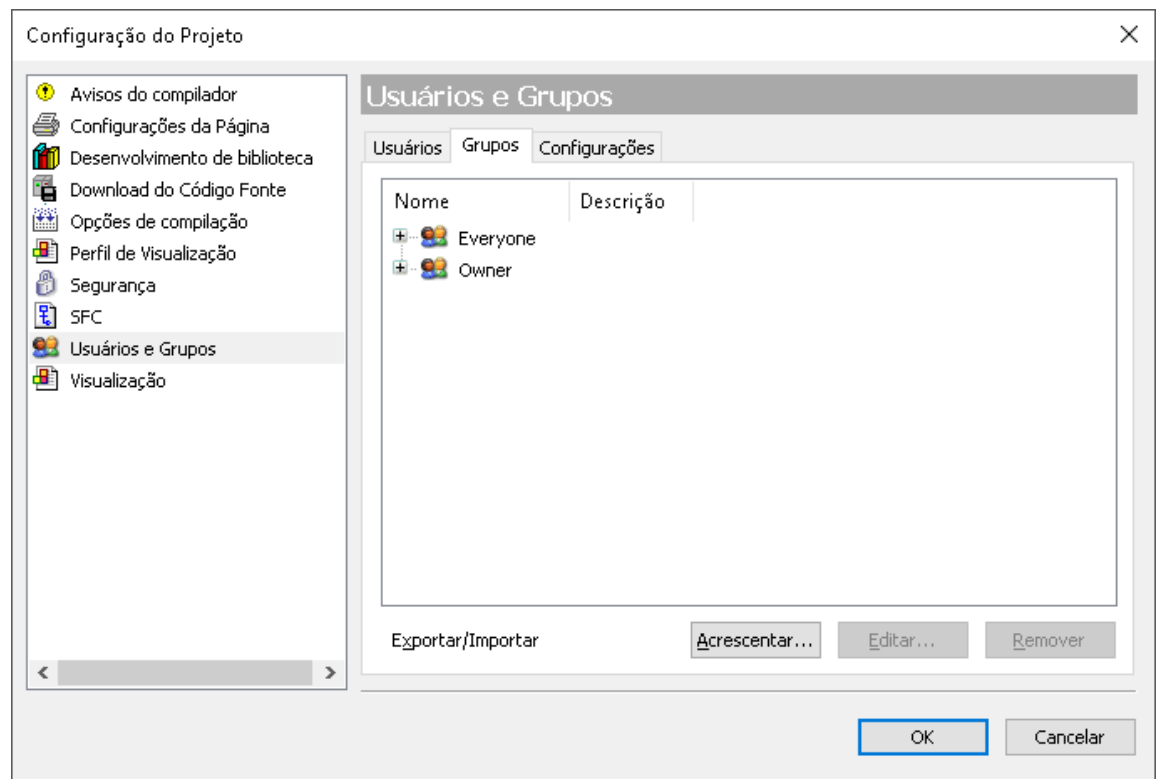


Figura 3-8. Configurações de Projeto, Diálogo Grupos

Para acrescentar um novo grupo, utilize o botão *Acrescentar* e abra o diálogo correspondente.

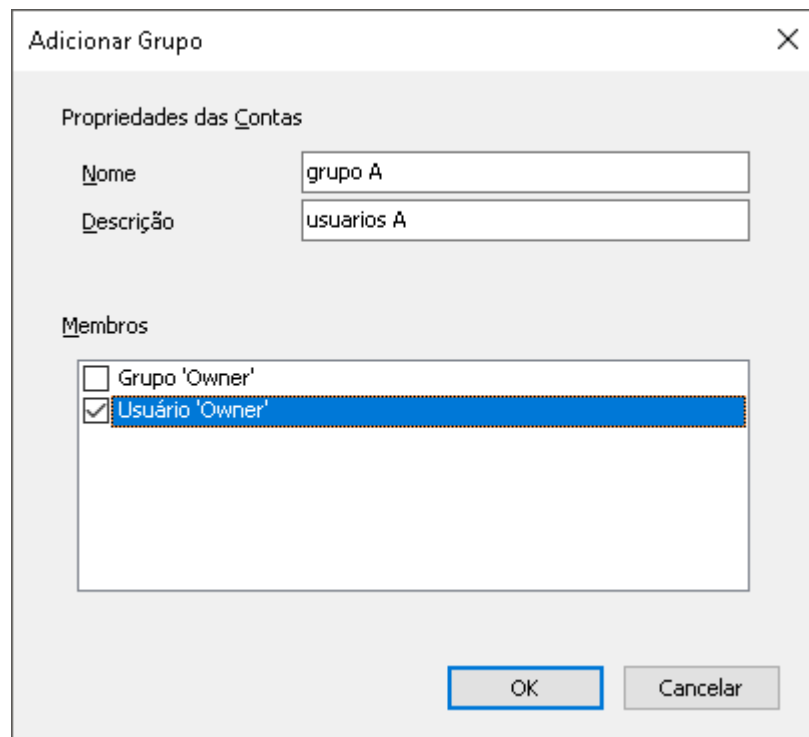


Figura 3-9. Acrescentar Grupo

Para configurar o novo grupo, feche o diálogo com *OK*. No caso de alguma incoerência (ausência de nome, grupo já existente, seleção de um grupo que provocaria um “ciclo de grupo”), será apresentada uma mensagem de erro.

Para modificar um grupo existente, use o botão *Editar* para abrir o diálogo *Editar Grupo*. Os campos são iguais aos do diálogo *Acréscimo de Grupo*. Os campos da senha - por questões de segurança - apresentarão 32 asteriscos (*). Após ter modificado os itens desejados, feche o diálogo com *OK* para aplicar as novas configurações.

Para remover um ou vários grupos, selecione os respectivos grupos na árvore e tecla *Remover*. Note que esta ação não exige confirmação. Os membros dos grupos excluídos permanecerão inalterados. Não é possível excluir os grupos *Everyone* e/ou *Owner*. Caso o usuário tente fazer isto, aparecerá uma mensagem de erro.

Configurações

As configurações básicas disponíveis referentes às contas do usuário podem ser conferidas na figura a seguir.

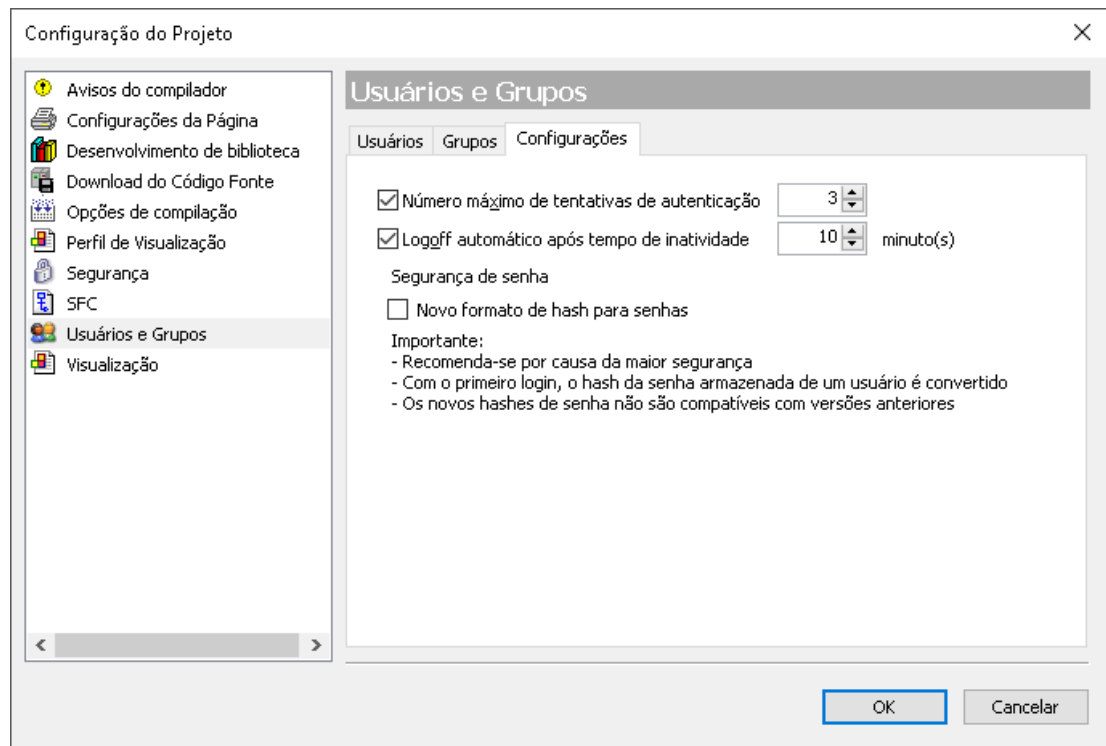


Figura 3-10. Configurações de Projeto, Diálogo Configuração de Usuários e Grupos

Gerenciamento de Direitos de Acesso

O gerenciamento de usuários em um projeto somente é útil se combinado com o gerenciamento dos direitos de acesso.

Em um novo projeto, basicamente todos os direitos de acesso não são definidos automaticamente, mas configurados para um valor padrão, ou seja, normalmente os direitos estão “garantidos”.

Durante a execução do projeto, cada direito pode ser explicitamente garantido ou negado e configurado novamente para o padrão. O gerenciamento dos direitos de acesso é feito no diálogo *Permissões* ou - para os direitos de acesso aos objetos - no diálogo *Controle de Acesso* (que faz parte do diálogo *Propriedades do Objeto*).

Os direitos de acesso aos objetos são “herdados”. Se o objeto tem um objeto “principal”, os direitos de acesso deste tornam-se as configurações padrão do objeto secundário (exemplo: se uma ação é atribuída a um programa, ela é inserida na sua estrutura. Assim, o programa é o

objeto “principal” da ação). No que se refere aos direitos de acesso, normalmente as relações dos objetos principais - secundários correspondem às relações mostradas na árvore das POU's ou Dispositivos e são indicadas no diálogo *Permissões* através da sintaxe “<objeto principal>. <objeto secundário>”. Exemplo: Ação ACT é atribuída ao objeto MainPrg (POU). Assim, na janela das POU's, ACT é mostrada na árvore de objetos em MainPrg. No diálogo *Permissões*, ACT é representada por “MainPrg.ACT” indicando que MainPrg é o “principal” da ACT. Se o direito “modificar” fosse explicitamente negado para MainPrg e a um determinado grupo de usuários, o valor padrão deste direito para ACT também seria “negado” automaticamente.

Para acessar a tela *Permissões* deve-se clicar nessa opção no menu *Projeto > Gerenciamento de Usuário*. Será aberta a tela da Figura 3-11 figura mostrada a seguir.

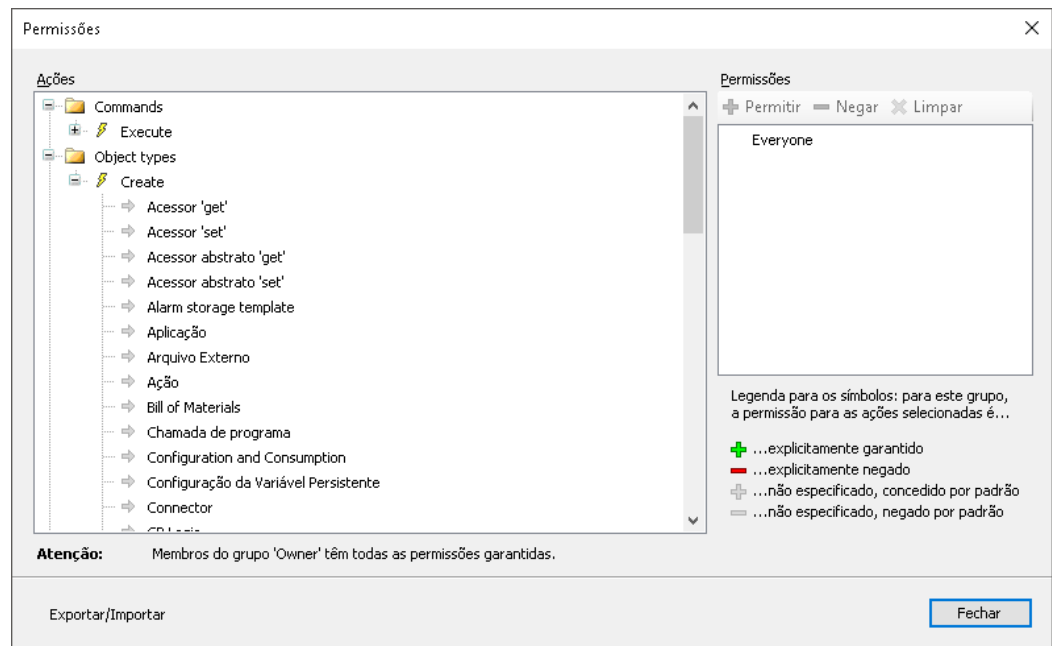


Figura 3-11. Permissões

A janela *Ações* exibe todas as possibilidades de direitos, ou seja, todas as ações que podem ser executadas nos objetos.

Os ícones não serão exibidos caso sejam selecionadas várias ações sem as configurações únicas referentes ao grupo selecionado no momento.

Para configurar os direitos para um grupo selecione a ação e o grupo desejados nas janelas *Ações* e *Permissões* respectivamente. Após, use os botões apropriados na barra de ferramentas da janela *Permissões*

Gerenciamento de Usuários e Direitos de Acesso da UCP

As UCP's Nexto possuem um sistema de gerenciamento de permissões de usuário, que bloqueia ou permite certas ações para cada grupo de usuários na UCP. Para editar estes direitos na UCP, o usuário necessita acessar um projeto no software MasterTool IEC XE, não sendo necessário estar logado na UCP. Deverá então clicar na Árvore de Dispositivos, localizada à esquerda do programa, dar dois cliques no item *Device* e, após, selecionar a UCP na aba *Configurações de Comunicação* que será aberta. Apenas as abas *Usuários e Grupos* e *Direitos de Acesso* se relacionam com este tópico. A figura a seguir exemplifica os passos para acessar estas abas da UCP.

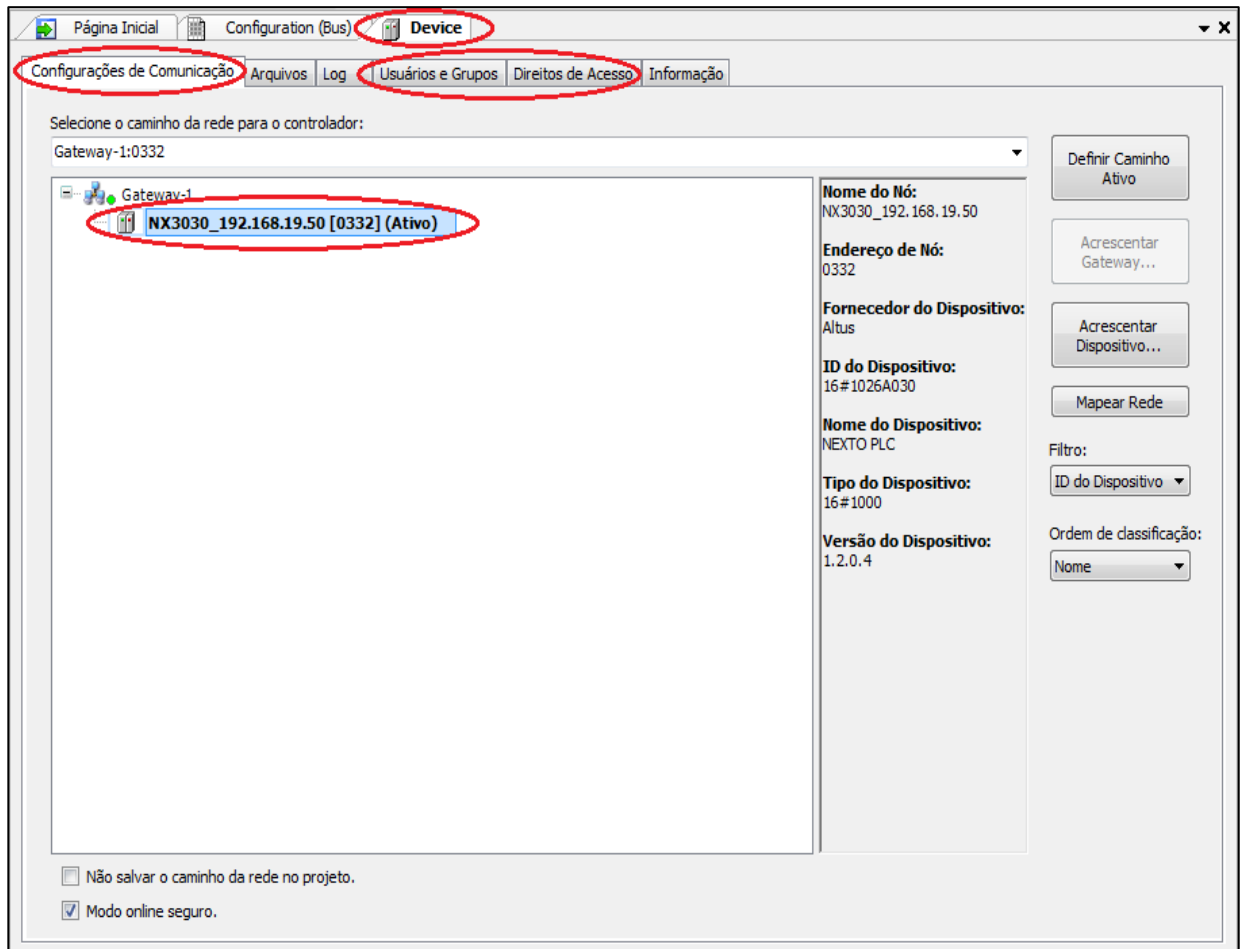


Figura 3-12. Acesso às Abas Usuários e Grupos e Direitos de Acesso

ATENÇÃO:

Caso o usuário esqueça a(s) senha(s) da(s) conta(s) com acesso à UCP, o único modo de recuperar este acesso será atualizando o firmware da mesma.

ATENÇÃO:

Após executar o comando Logoff de um usuário da UCP, deve-se fechar a aba "Device" desse mesmo projeto para efetivamente encerrar os seus direitos de acesso.

Pesquisa na Documentação: Usuários e Grupos



Consulte, na documentação do produto, as configurações típicas associadas ao diálogo *Usuários e Grupos*, o qual é fornecido em uma guia do diálogo *Device*. Ele permite configurar contas de usuários e grupos que, em conjunto com o gerenciamento dos direitos de acesso controlam o acesso aos objetos no CP no modo online.

Fonte: Manual de Utilização MasterTool IEC XE MT8500 – MU299048

Easy Connection

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

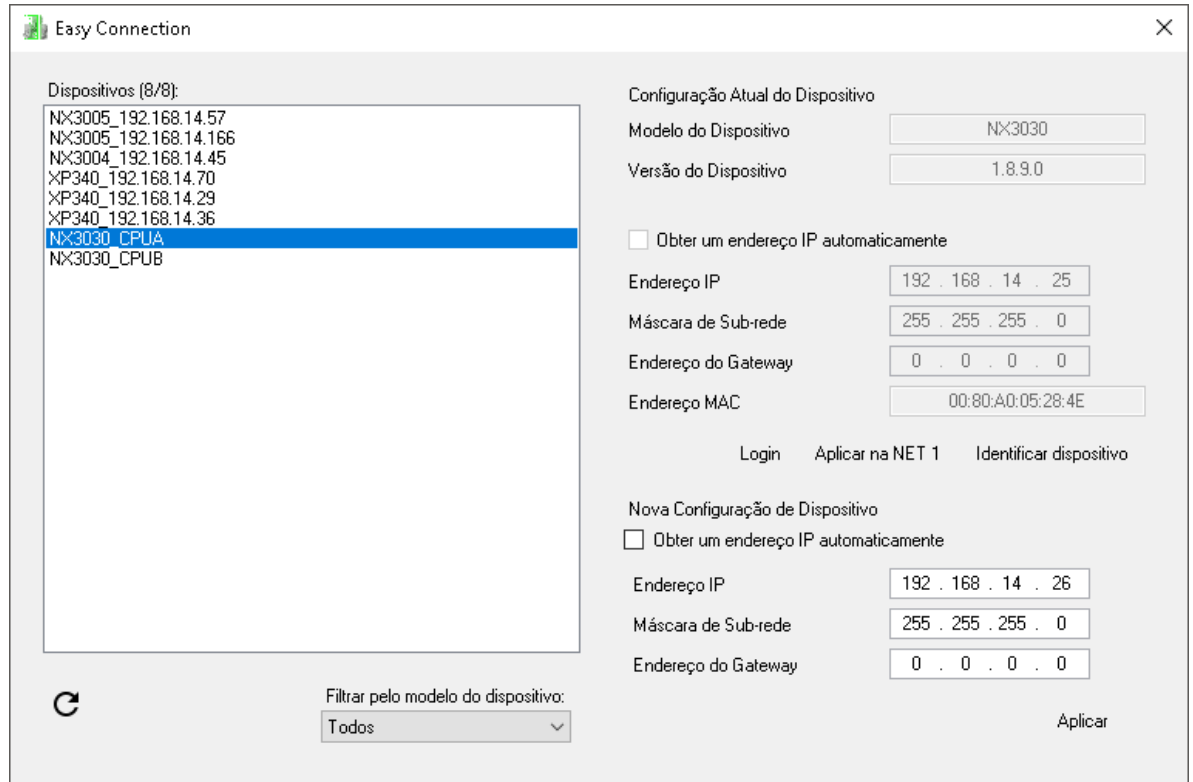


Figura 3-13. Easy Connection

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

Listas de Variáveis Globais (GVLs)

O projeto criado pela ferramenta MasterTool IEC XE contém 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.

GVL System_Diagnostics

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

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

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

GVL Disables

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

```
1 [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
1 VAR_GLOBAL
2 MODBUS_Device_DISABLE_0001 : BOOL;
3 MODBUS_Device_DISABLE_0002 : BOOL;
4 MODBUS_Device_DISABLE_0003 : BOOL;
5 MODBUS_Device_1_DISABLE_0001 : BOOL;
6 MODBUS_Device_1_DISABLE_0002 : BOOL;
7 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.

GVL IOQualities

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

```
Device.Application.IOQualities
1 VAR_GLOBAL
2 QUALITY_NX1001: ARRAY[0..15] OF LibDataTypes.QUALITY;
3 QUALITY_NX2020: ARRAY[0..15] OF LibDataTypes.QUALITY;
4 QUALITY_NX6000: ARRAY[0..7] OF LibDataTypes.QUALITY;
5 QUALITY_NX6100: ARRAY[0..3] OF LibDataTypes.QUALITY;
6 END_VAR
```

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

GVL Module_Diagnostics

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

GVL Qualities

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

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

Onde:

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

ATENÇÃO:

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

Exemplo:

```
Device.Application.Qualities
1 VAR_GLOBAL
2 MODBUS_Device_QUALITY_0001: LibDataTypes.QUALITY;
3 MODBUS_Device_QUALITY_0002: LibDataTypes.QUALITY;
4 MODBUS_Device_QUALITY_0003: LibDataTypes.QUALITY;
5 END_VAR
```

A GVL Qualities é editável, portanto as variáveis de qualidade dos mapeamentos podem ser criadas manualmente não precisando seguir o modelo criado pela declaração automática e podem ser usadas as duas maneiras ao mesmo tempo, mas devem sempre ser do tipo LibDataTypes.QUALITY e deve-se tomar o cuidado para não excluir ou alterar as variáveis declaradas automaticamente, pois elas podem estar sendo usadas por algum dispositivo. Se a

variável for excluída ou alterada será gerado um erro ao compilar o projeto. Para corrigir o nome de uma variável declarada automaticamente, deve-se seguir o modelo exemplificado acima de acordo com o dispositivo e o mapeamento aos quais pertence.

ATENÇÃO:

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

GVL ReqDiagnostics

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

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

Onde:

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

ATENÇÃO:

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

Exemplo:

```
Device.Application.ReqDiagnostics
1 VAR_GLOBAL
2 MODBUS_Device_REQDG_0001 : NXMODBUS_DIAGNOSTIC_STRUCTS.
3 T_DIAG_MODBUS_RTU_MAPPING_1;
4 MODBUS_Device_REQDG_0002 : NXMODBUS_DIAGNOSTIC_STRUCTS.
5 T_DIAG_MODBUS_RTU_MAPPING_1;
6 MODBUS_Device_REQDG_0003 : NXMODBUS_DIAGNOSTIC_STRUCTS.
7 T_DIAG_MODBUS_RTU_MAPPING_1;
8 MODBUS_Device_1_REQDG_0001 : NXMODBUS_DIAGNOSTIC_STRUCTS.
9 T_DIAG_MODBUS_ETH_MAPPING_1;
10 MODBUS_Device_1_REQDG_0002 : NXMODBUS_DIAGNOSTIC_STRUCTS.
11 T_DIAG_MODBUS_ETH_MAPPING_1;
12 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.

Visualização

A *Visualização* totalmente integrada é a solução ideal 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 pode sem 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 visualização 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.

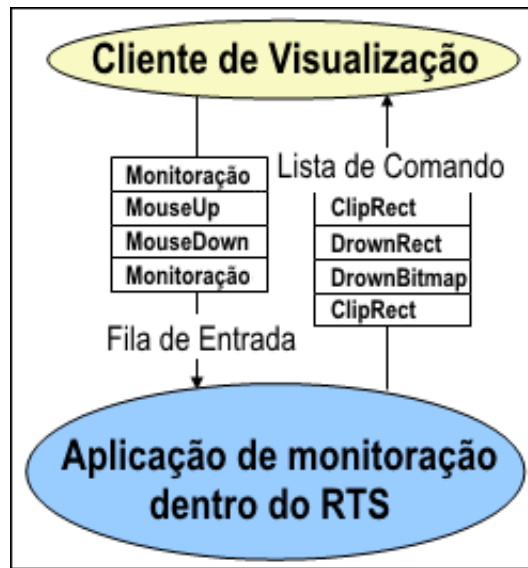


Figura 3-14. Mecanismo Geral

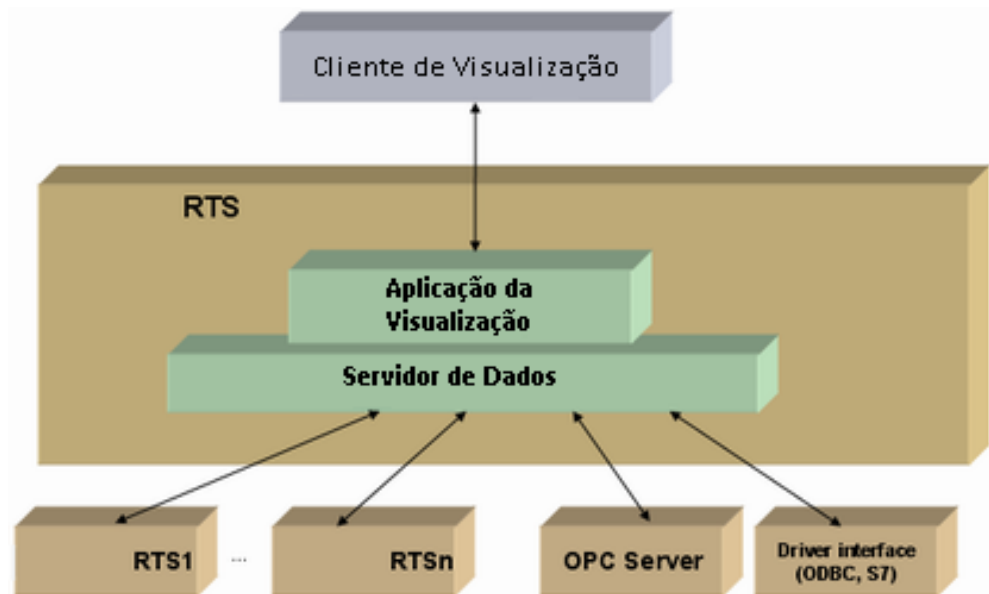


Figura 3-15. Servidor de Dados para fornecer dados de visualização

Uma aplicação de visualização é criada em código IEC no editor de visualização do sistema de programação. No caso da visualização estar executando no dispositivo, o código desta será gerado e carregado lá.

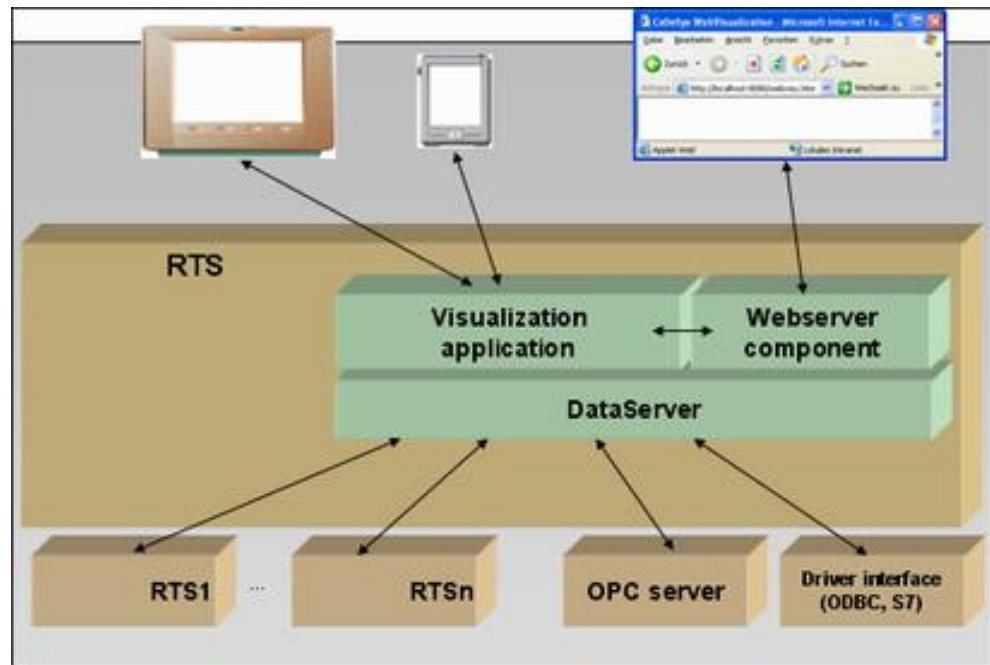


Figura 3-16. Cliente de visualização WebVisu

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 POU's, 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 em Modo de Simulação.

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.

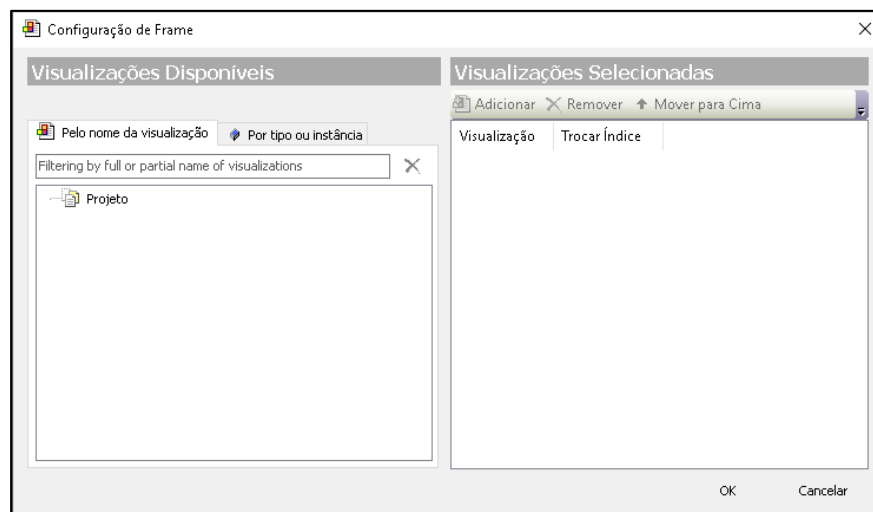


Figura 3-17. Caixa de diálogo de Configuração de Frame

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*.

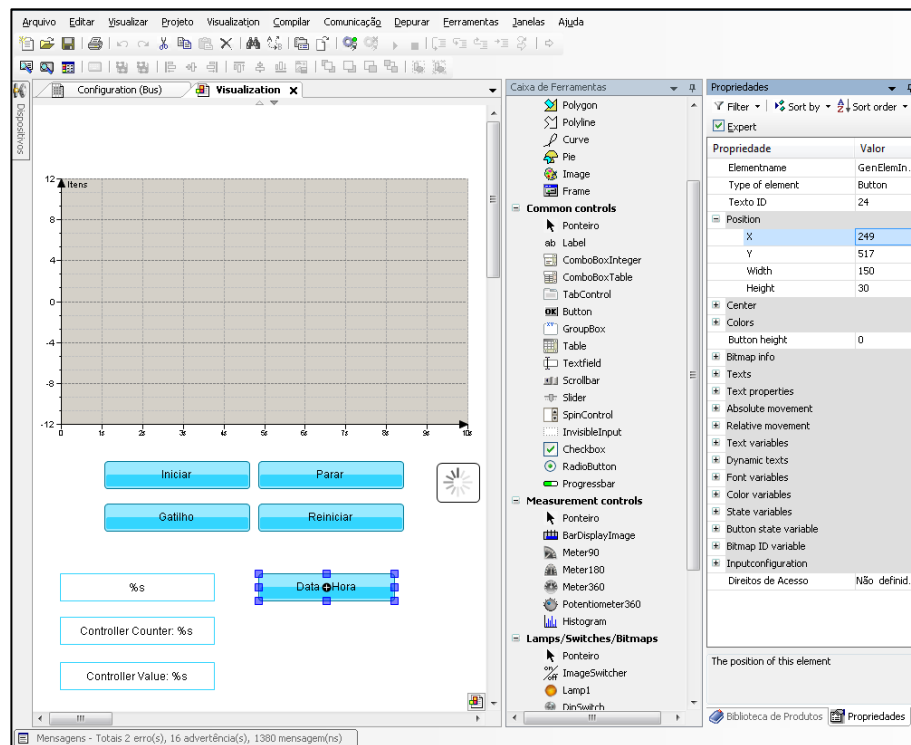



Figura 3-18. Editor de Visualização

A visualização pode ser criada com a ajuda dos seguintes editores e ferramentas adicionais:

A janela do editor é intitulado  <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.

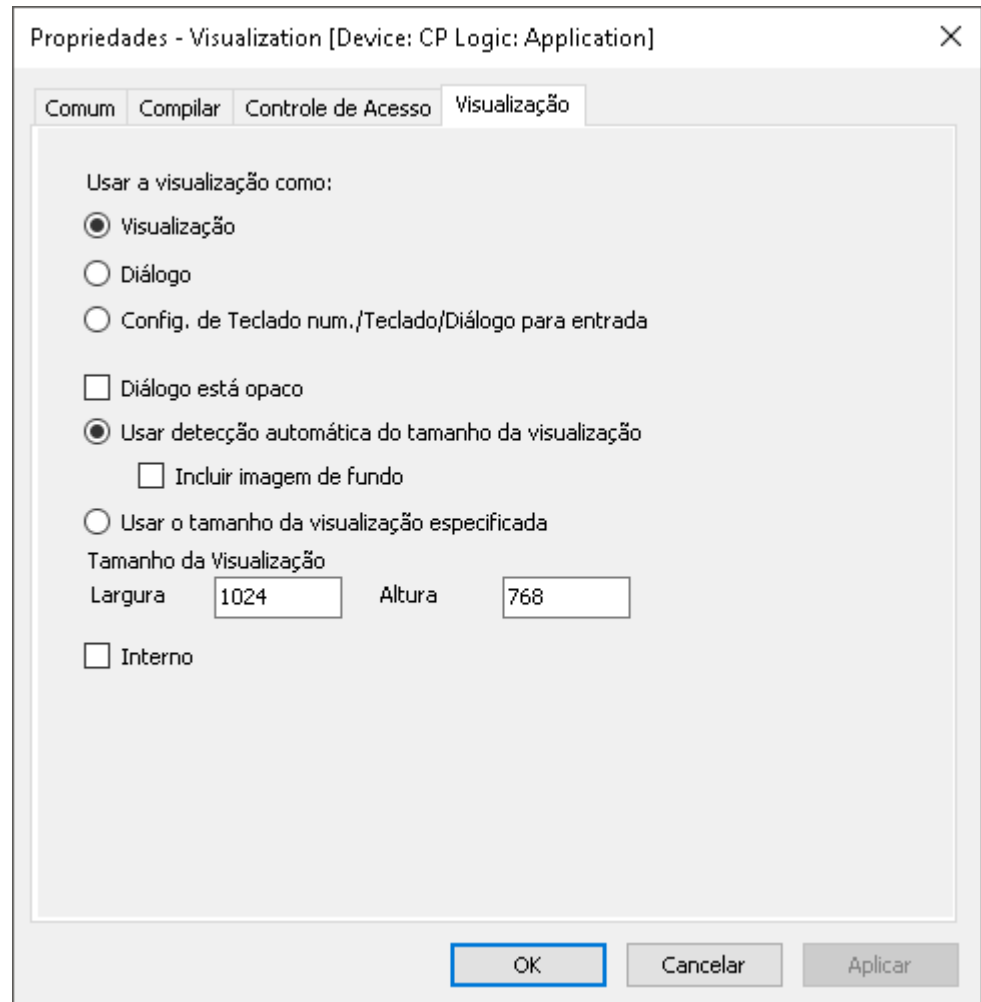


Figura 3-19. Diálogo 'Propriedades', Categoria 'Visualização'

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 serão adicionadas 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 3-1: criação de uma visualização WEB

A visualização web é uma funcionalidade avançada que está disponível nas CPUs NX3005 e XP340 da linha Nexto (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.



DICA: consulte o Instrutor para orientações adicionais.

ANOTAÇÕES

ANOTAÇÕES

altus Webinar Web Server embarcado em CLP: Pr... Assistir m... Compartilh...

WEBINAR

Quando: 23/04
Horário: 10h

WEB SERVER EMBARCADO EM CLP:
PRATICIDADE + ECONOMIA

altus

4. Funcionalidades avançadas no CP Nexto

Este capítulo inclui a análise de algumas funcionalidades avançadas do dispositivo no que se refere às aplicações industriais, enfatizando a utilização do bloco funcional PID.



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.

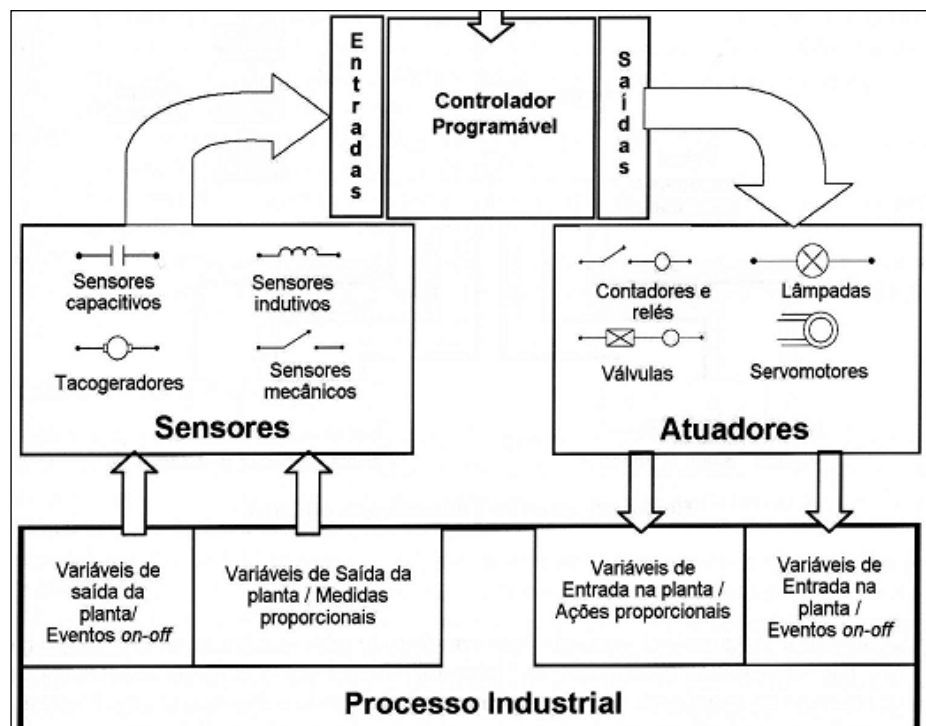


Figura 4-1. Papel do Controlador em Sistemas de Controle Industriais

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.

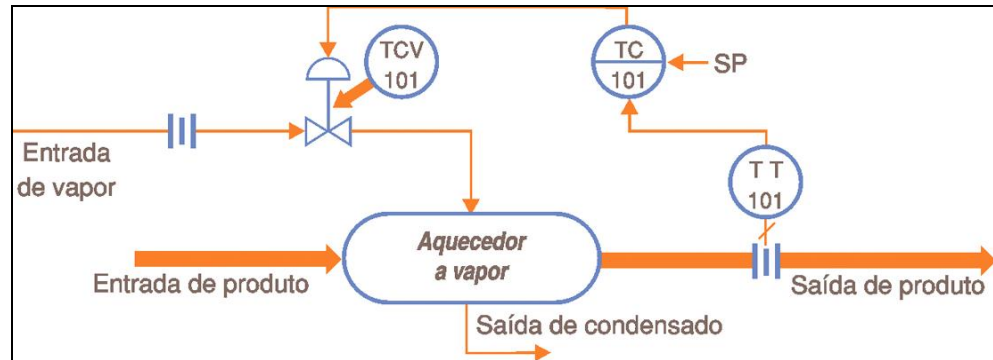


Figura 4-2. Malha de Controle

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.

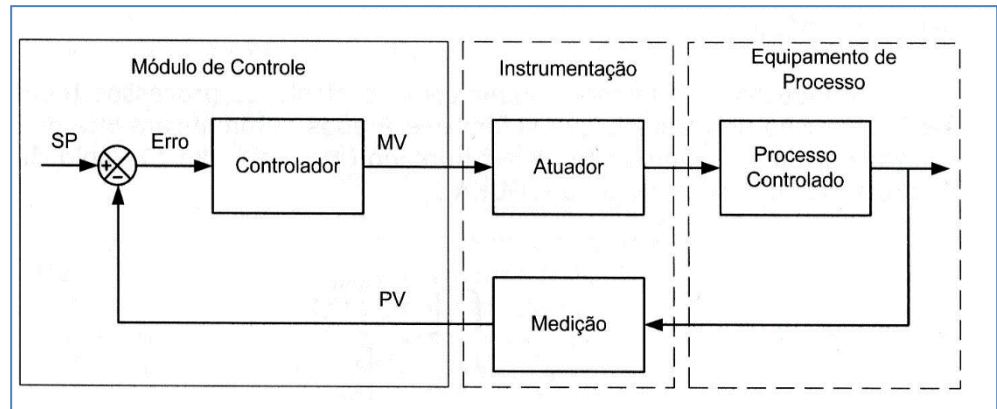


Figura 4-3. Sistema de Controle com Realimentação

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.

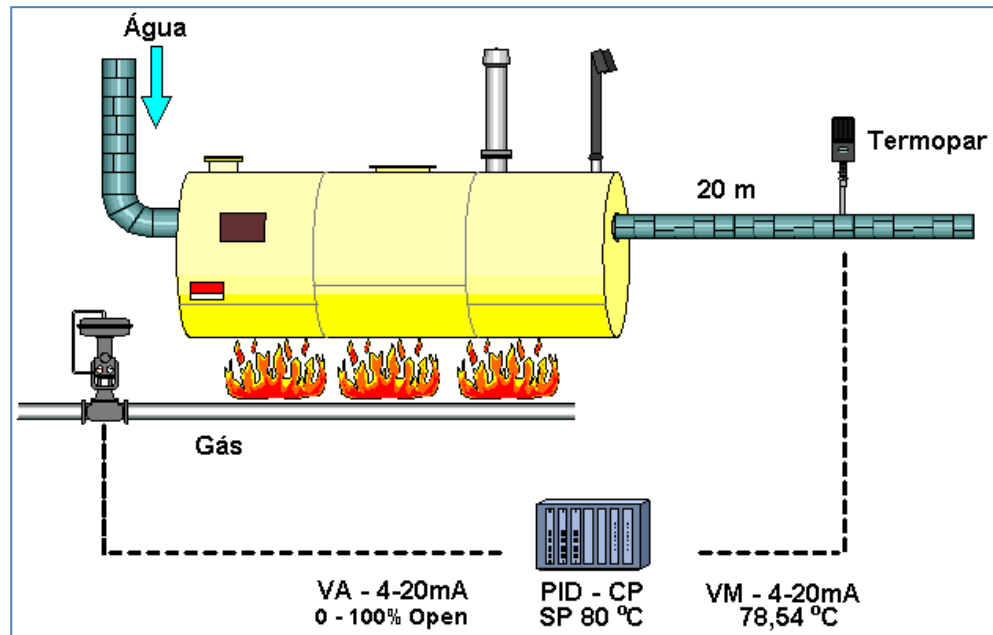


Figura 4-4. Controle PID para Temperatura

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.

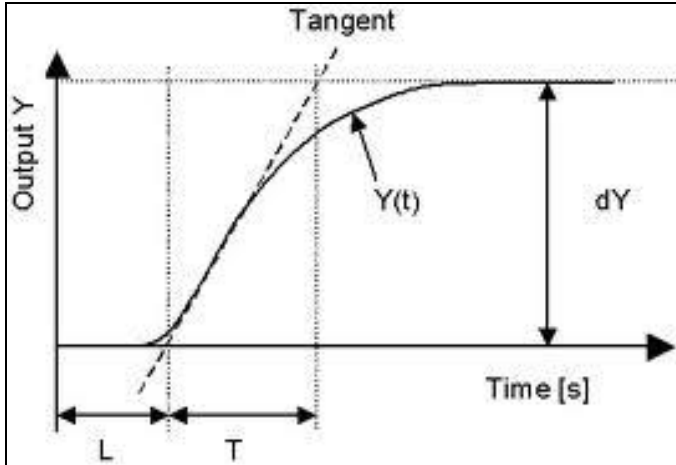


Figura 4-5. Resposta típica ao degrau

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).

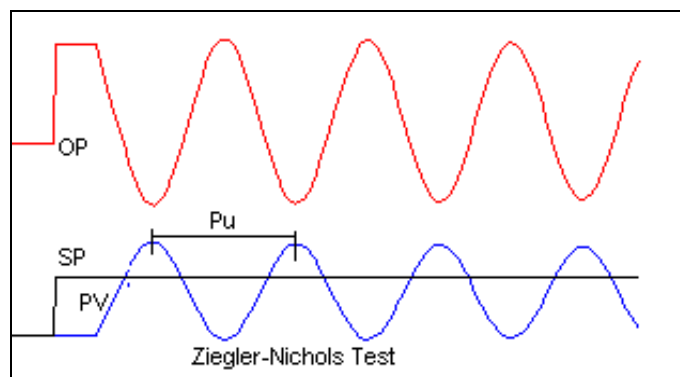
Assim sendo, os valores de K_p , T_i e T_d podem ser obtidos a partir das fórmulas apresentadas na tabela a seguir.

Tipo de controlador	K_p	T_i	T_d
P	T / L	INFINITO	0
PI	$0,9 T / L$	$L / 0,3$	0
PID	$1,2 T / L$	$2 L$	$0,5 L$

Tabela 4-1. Regra de sintonia de Ziegler-Nichols (Método 1)

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 4-6. 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.

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

Tabela 4-2. Regra de sintonia de Ziegler-Nichols (Método 1)

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.

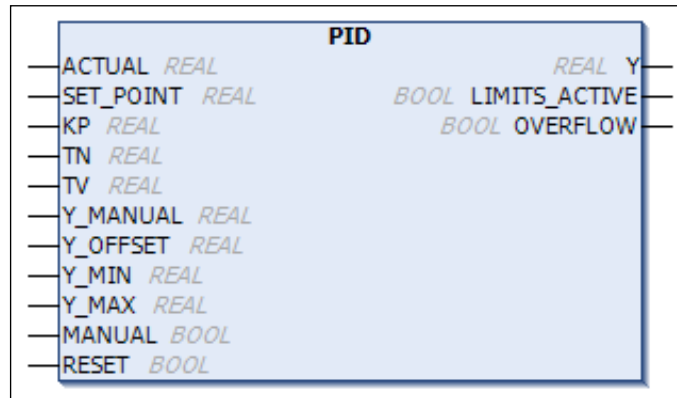


Figura 4-7. Bloco Funcional PID

Entradas do Bloco Funcional

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 milisegundos.
TV	REAL	Tempo de ação derivativa, ganho de unidade da parte D em segundos, por exemplo, "0,5" para 500 milisegundos.
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.

Tabela 4-3. Entradas do Bloco Funcional PID

Saídas do Bloco Funcional

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).

Tabela 4-4. Saídas do Bloco Funcional PID

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 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 4-1: 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.

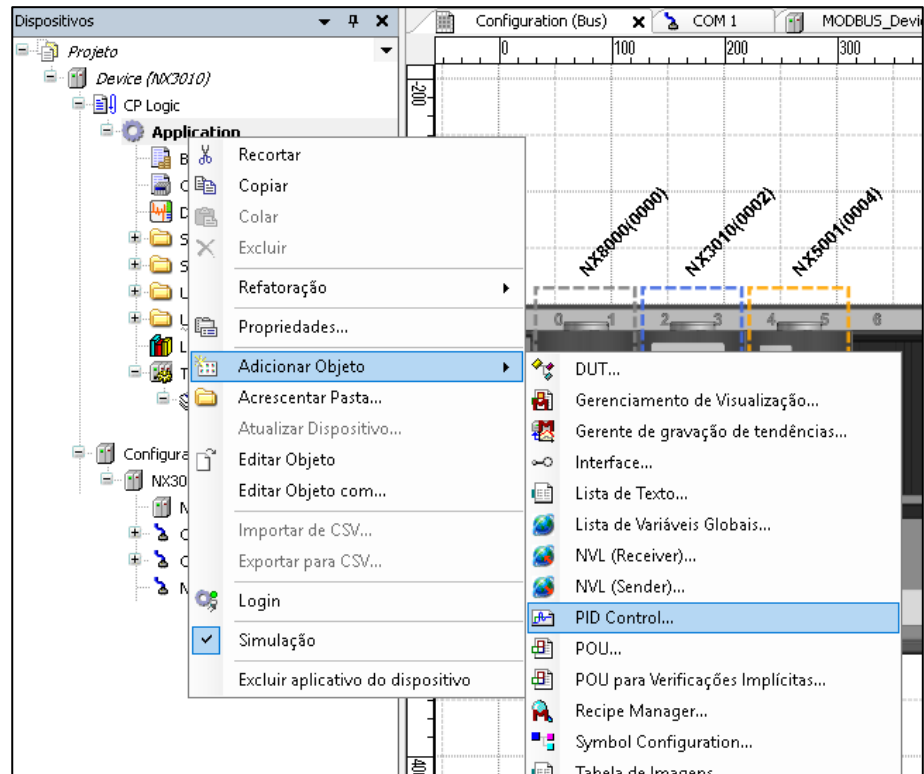


DICA: consulte o Instrutor para orientações adicionais.

ANOTAÇÕES

Implementação do Estudo Dirigido 4-1:


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.

Acrescentar PID Control ×

 Controle Proporcional Integrativo Derivativo

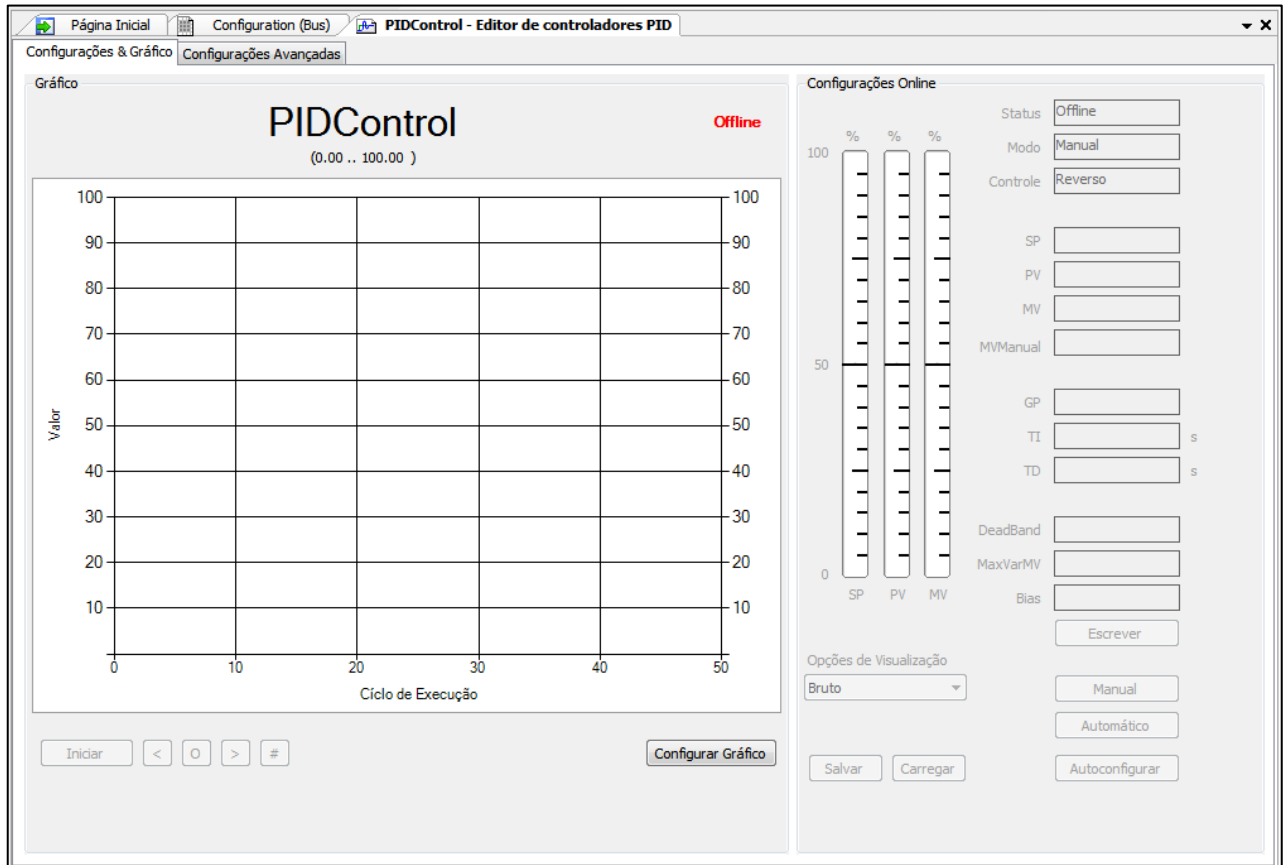
Nome:

Adicionar Cancelar

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:

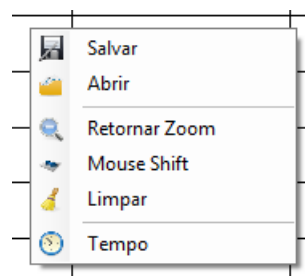
- deslocar o gráfico para a esquerda (<).
- deslocar o gráfico para a direita (>).
- voltar para a posição normal do gráfico (O).
- executar operação de autofit do gráfico (#).



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.

Configuração do Gráfico

Eixo X	Eixo Y	Detalhes do Gráfico
Modo de Visualização: Exibição Normal	Modo de Visualização: Exibição Normal	Pontos Armazenados: 1000
Pontos no Gráfico: 50	Y Min: 0	Tempo de Amostragem (ms): 100
Deslocamento em X: 5	Y Máx: 100	

OK Padrão Cancelar

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.

Configurações Online

Status: Offline

Modo: Automático

Controle: Reverso

SP: 0.00

PV: 0.00

MV: 0.00

MVManual: 0.00

GP: 0.900

TI: 10.000 s

TD: 0.275 s

DeadBand: 0.00

MaxVarMV: 0.00

Bias: 0.00

Opções de Visualização: Bruto

Escrever

Manual

Automático

Autoconfigurar

Salvar Carregar

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.

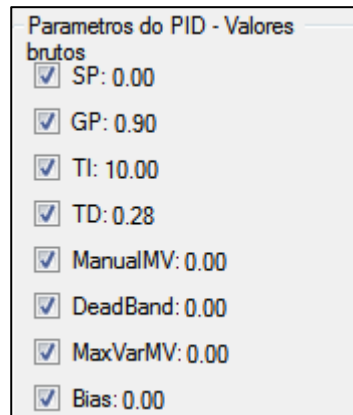
Os campos afetados pelas opções de visualização são: *SP*, *PV*, *MV*, *MVManual*, *DeadBand*, *MaxVarMV* e *Bias*. Estes campos são afetados apenas quando exibidos na caixa de texto, o gráfico de tendência e os gráficos de barra sempre são mostrados em porcentagem.

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.

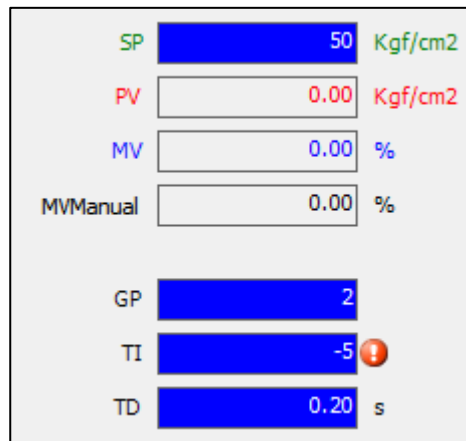


Parâmetro	Valor
SP	0.00
GP	0.90
TI	10.00
TD	0.28
ManualMV	0.00
DeadBand	0.00
MaxVarMV	0.00
Bias	0.00

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.

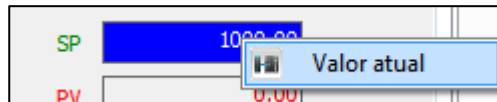


Parâmetro	Valor	Unidade
SP	50	Kgf/cm2
PV	0.00	Kgf/cm2
MV	0.00	%
MVManual	0.00	%
GP	2	
TI	-5	s
TD	0.20	s

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 4-8. 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.

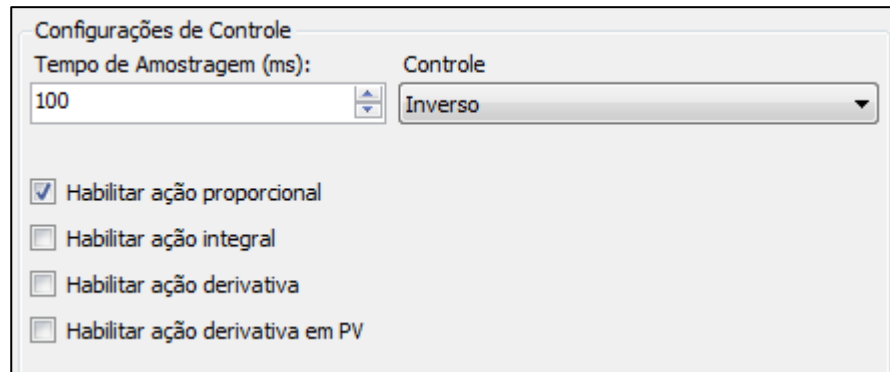
Os campos *Mínimo* e *Máximo*, definem a faixa de operação das variáveis SP, PV e MV. Os campos *Mínimo SP* e *Máximo SP*, não permitem edição. Estes campos assumem os valores dos campos *Mínimo Engenharia de PV* e *Máximo Engenharia de PV*, respectivamente.

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

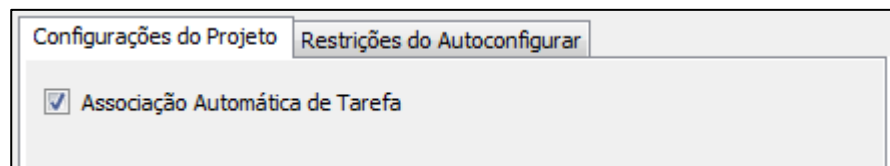
Habilitar ação proporcional

Habilitar ação integral

Habilitar ação derivativa

Habilitar ação derivativa em PV

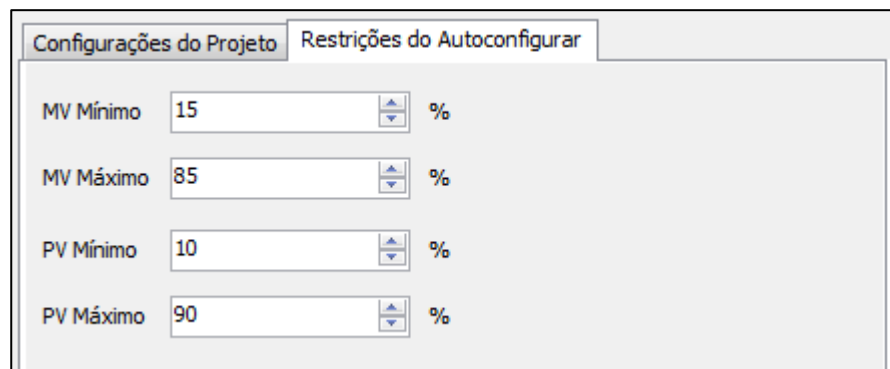
Esta aba contém a opção *Associação Automática de Tarefa*, com a opção habilitada o controlador é associado automaticamente a uma tarefa do sistema, o que permite que ele seja utilizado normalmente. Se a opção estiver desabilitada o controlador deve ser associado a alguma tarefa manualmente ou chamado 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 exhibe 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 a este 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*.



Estudo Dirigido 4-2: bloco funcional PID_S

Tomando como base o Estudo de Caso proposto no capítulo 5 declare um bloco funcional chamado PID_S permite a operação remota pelo supervisor a partir da reutilização do bloco funcional PID da biblioteca inicial.



DICA: consulte o Instrutor para orientações adicionais.

ANOTAÇÕES

Implementação do Estudo Dirigido 4-2:

Etapa 1: Declaração do tipo de bloco funcional PID_S.

```
1  FUNCTION_BLOCK PID_S
2  VAR_INPUT
3      AUTO:BOOL; (* FALSE = Manual *)
4      PV:REAL; (* Variável Processo *)
5      SP_L:REAL; (* SP local *)
6      XO_L:REAL; (* Ajuste manual p/ saída local *)
7      KP,TR,TD:REAL; (* Ganho KP, tempos TR e TD *)
8      CYCLE:TIME; (* Tempo de ciclo do bloco *)
9  END_VAR
10 VAR_OUTPUT
11     XOUT:REAL := 0.0; (* Valor de saída *)
12     Modo_LR:BOOL := FALSE; (* FALSE = Local *)
13 END_VAR
14 VAR
15     SP_S:REAL; (* SP remoto (pelo supervisor) *)
16     XO_S:REAL; (* Ajuste manual (pelo supervisor) *)
17     Modo_LR_S:BOOL; (* Ajuste l/R (pelo supervisor) *)
18     AUTO_S:BOOL; (* Ajuste m/A (pelo supervisor) *)
19     C_PID:PID; (* Instância do bloco funcional PID *)
20 END_VAR
```

Etapa 2: Corpo do bloco funcional

A partir da declaração proposta anteriormente para o bloco funcional em questão implemente o corpo deste.

5. Aplicações, Avaliação e Encerramento

Estudo de caso

O objetivo desse Estudo de Caso é aplicar as principais funcionalidades avançadas do binômio NEXTO – MasterTool IEC XE através da proposição da Automação de um processo de mistura e secagem a ser implementado à luz da norma IEC61131-3. A automação proposta foi adaptada da Referência.

Especificação funcional da automação proposta

O processo a ser automatizado tem por objetivo a produção de determinado produto como resultado da mistura de duas matérias-primas e posterior aquecimento e secagem via resfriamento do material resultante. O processo consiste em duas unidades: Misturador e Secador. A figura a seguir mostra o diagrama de processo e instrumentação do sistema. Consulte o instrutor para orientações quanto à funcionalidade do mesmo.

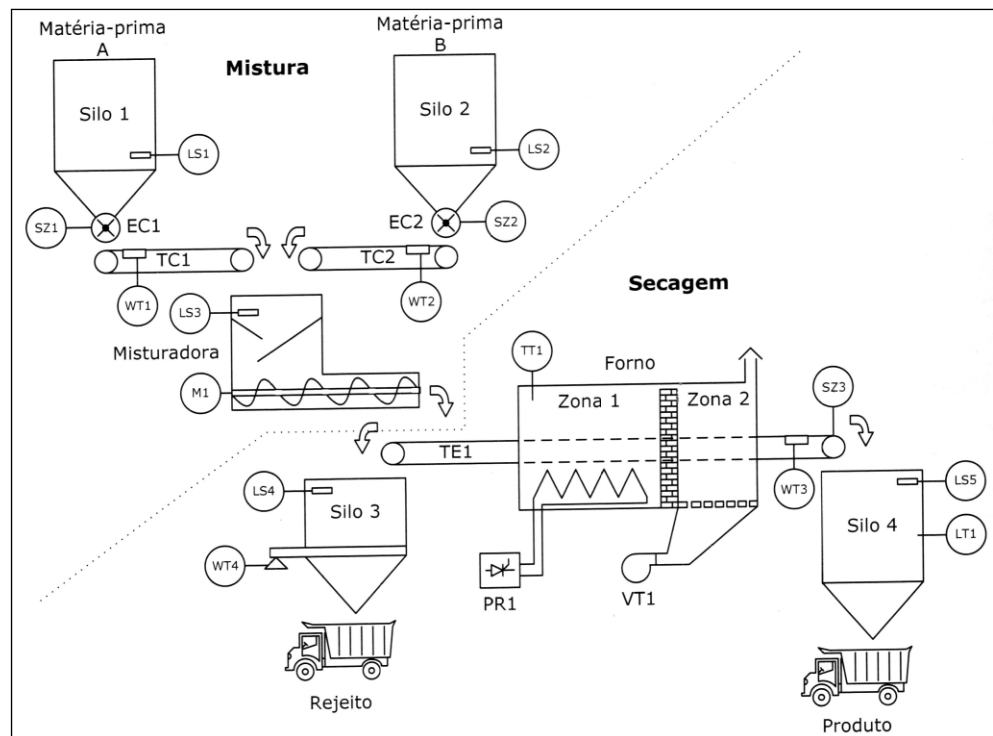


Figura 5-1. Diagrama do estudo de caso 2

Especificações do processo

Item	Especificação
Misturador	As eclusas fazem a retirada das matérias-primas dos silos numa taxa determinada pelos inversores, depositando-as nas correias transportadoras. Balanças dinâmicas pesam o material nas correias. As correias levam o material para mistura e homogeneização no misturador. Um eixo helicoidal transporta o material composto até a esteira de saída.
Secador	O secador é formado por uma esteira transportadora que conduz o material composto através do forno. Este último é composto de duas zonas (uma para aquecimento e outra para resfriamento e secagem). Uma balança totaliza a produção. O material resultante do processo é depositado no silo apropriado. A esteira transportadora tem é comandada por um acionamento reversível que permite também o ajuste de velocidade de transporte do material. Outro silo com abertura manual recebe o rejeito da produção, o qual pode ser pesado por uma balança.

Condições de contorno e restrições

- As taxas de adição das matérias-primas devem ser constantes, obedecendo a receita do produto no ritmo de produção adequado.
- A velocidade da esteira transportadora do secador deve ser compatível com a receita e o ritmo de produção desejado.
- Os dados referentes à receita são escritos no CP pelo sistema supervisor e definem a taxa de alimentação da matéria-prima "A", a proporção entre matérias-primas, a velocidade de passagem no forno, o set point de aquecimento, a quantidade a ser produzida e o código da receita.
- No início do processamento ou em uma situação de alarme (mistura/forno) a esteira transportadora do secador reverterá seu sentido para limpeza do sistema. Nessa situação, as esteiras transportadoras do misturador e o próprio devem permanecer acionados.
- Uma interrupção momentânea no sistema deve reduzir a temperatura do forno a um valor mínimo, visando economia de energia e aumento da vida útil do mesmo.
- O acionamento simultâneo dos equipamentos deve ser evitado quando da partida do sistema elétrico. A parada deve aguardar o esvaziamento das esteiras e do misturador. As eclusas devem ser ligadas/desligadas de forma simultânea para prevenir problemas associados à proporção de adição de matérias-primas.

Decomposição do controle

A tabela a seguir pode ser utilizada para o modelamento do processo. A partir dela, as classes de controle podem ser extraídas, para o encaminhamento da solução.

Unidade	Equipamento	Controle	Classe de controle
Misturador	Silo – Produto A	Nível LS1	Sensor discreto
		Eclusa A	Controle da Taxa EC1
	Motor EC1		Partida direta + Inversor
	Silo - Produto B	Nível LS2	Sensor discreto
		Eclusa B	Controle Taxa EC2
	Motor EC2		Partida direta + Inversor
	Esteira A	Motor TC1	Motor partida direta
		Balança WT1	Monitoração analógica
	Esteira B	Motor TC2	Motor partida direta
		Balança WT2	Monitoração analógica
Mistura	Nível LS3	Sensor discreto	
	Motor Mistura	Motor partida direta	
Secador	Esteira - Secagem	Motor TE1	Partida direta rev. + Inversor
		Balança WT3	Monitoração analógica
	Forno	Temperatura TT1	Monitoração analógica
		Temperatura PR1	PID
		Ventilador VT1	Motor partida direta
	Silo - Rejeito	Nível LS4	Sensor discreto
		Massa WT4	Monitoração analógica
	Silo - Produto	Nível LS5	Sensor discreto
Nível LT1		Monitoração analógica	

Lista de entradas e saídas

A tabela a seguir deve ser completada com as entradas e saídas do sistema. Consulte o instrutor para orientações adicionais.

Descrição do sinal	Variável	Tipo de ponto
Botão de emergência	Geral	Entrada digital
...		

Modos de operação

Modo de operação	Descrição
Manual	Através da seleção do operador, o sistema poderá operar no modo manual, ficando o operador responsável pelo comando dos dispositivos e equipamentos e pela verificação das condições de funcionamento.
Automático	Através da seleção do operador, o sistema poderá operar no modo automático, ficando o operador responsável pela partida/parada do processo ou agrupamento de equipamentos. Um intertravamento apropriado deverá nortear o funcionamento do processo.
Local/Remoto	Através do sistema supervisorário é possível fazer essa seleção. A opção pela operação local só é permitida com o sistema em manual. A operação remota permite o comando via sistema de supervisão.

Utilização de funções

Recomenda-se a utilização das funções disponíveis nas bibliotecas de POU's do MasterTool IEC XE.

Declaração de blocos funcionais

As classes de controle identificadas anteriormente no modelamento do processo podem ser implementadas através de blocos funcionais. A reutilização de software deve ser priorizada. Para implementação dos blocos funcionais recomendamos, por motivos didáticos, a utilização da linguagem FBD.

Acesso aos sinais dos sensores

Para implementação da classe de controle "Sensor Discreto" optou-se por acessar os sinais dos sensores via variáveis atribuídas diretamente aos endereços físicos através de "AT".

Programas para sequenciamento e estruturação

As funcionalidades do sistema serão implementadas através de programas que serão responsáveis pelo sequenciamento dos equipamentos e a estruturação do aplicativo. Os programas a serem criados são:

- Mistura;
- Secagem;
- Receitas.

Depuração

Antes de serem agregadas à biblioteca todas as POU's devem ser depuradas (teste e validação). Para tanto, recomenda-se a simulação da mesma no próprio ambiente de programação.

Tarefas para controle de execução

A tabela a seguir caracteriza as tarefas que serão criadas e associadas às POU's para atendimento das exigências de tempo de resposta específicas para o problema de controle.

Tarefa	Intervalo	Prioridade	POUs associadas (WITH)
CTRL_Discreto	50 ms	1	MISTURA SECAGEM
CTRL_Analogico	200 ms	2	Balanca_WT1/Balanca_WT2/Balanca_WT3 Controle_Taxa_EC1/Controle_Taxa_EC2 Temperatura_TT1 Peso_WT4 Nivel_LT1
Comando	500 ms	3	RECEITAS

Árvore de POU's da Aplicação

A figura a seguir ilustra a estruturação da árvore de POU's para o Estudo de Caso.

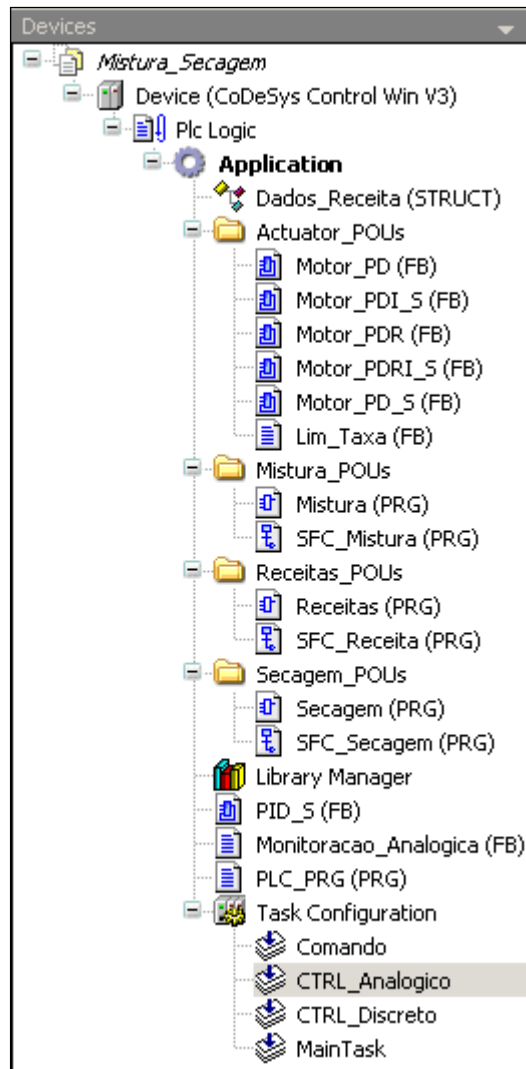
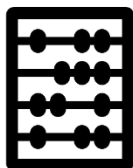


Figura 5-2. Árvore de POUs

Documentação

O MasterTool IEC XE permite um alto grau de documentação do projeto. A utilização das características preconizadas pela norma IEC 61131-3, em conjunto com os recursos gráficos disponibilizados pelo ambiente de programação dispensam a geração de documentação externa e levam ao desenvolvimento de um projeto de automação com alta qualidade de documentação. Para tanto, explore os recursos de documentação associados ao MasterTool IEC XE (consulte o instrutor em caso de dúvida), gerando a documentação do Estudo de Caso ora proposto.



Situações de Aprendizagem

As situações de aprendizagem propostas nessa seção complementam a implementação da aplicação proposta no estudo dirigido. O instrutor pode variar a estratégia de desenvolvimento das mesmas dependendo do perfil de competências do aluno, ou mesmo da disponibilidade de tempo para eventuais aprofundamentos no tema.

Situação de Aprendizagem 5-1: bloco funcional Motor_PD

Importe o bloco funcional Motor_PD desenvolvido no curso MP10 Básico, o qual será reutilizado na implementação de outras POU's de controle de motor a serem utilizadas no projeto do MasterTool IEC XE do Estudo de Caso ora proposto.

O motor pode funcionar em manual/automático. Em manual, duas entradas são utilizadas para ligar/desligar o motor. Em automático, a entrada de liberação comanda a ligação do motor.

Uma saída comanda o circuito elétrico do motor e outra indica falha (sobrecarga, defeito, falha na partida, etc.). Entradas apropriadas monitoram e sinalizam as ocorrências de falhas quando em FALSE.

Uma entrada deve detectar que o motor entrou em funcionamento após a saída de comando do motor ter sido acionada. Um tempo de 2,5 s (por exemplo) pode ser aguardado antes do bloco funcional acionar a saída apropriada devido a uma falha na partida.

A borda de subida do sinal será utilizada para ligar o motor.

Situação de Aprendizagem 5-2: importação do bloco funcional Monitoracao_Analogica

Importe o bloco funcional Monitoracao_Analogica desenvolvido no curso MP10 Básico, o qual será reutilizado na implementação da classe de controle Monitoração Analógica que será utilizada no projeto do MasterTool IEC XE do Estudo de Caso ora proposto.

O bloco funcional deve respeitar as seguintes especificações:

- converte uma variável analógica que varia entre 0 – 4095 para uma faixa de unidades de engenharia escolhida;
- utiliza uma faixa de histerese comum para caracterização dos sinais de alarme;
- gera os sinais de alarme para os níveis alto-alto (HH), alto (H), baixo (L), baixo-baixo (LL), em relação aos limites definidos.

Situação de Aprendizagem 5-3: importação do bloco funcional Lim_Taxa

Importe o bloco funcional Lim_Taxa desenvolvido no Estudo Dirigido 1-3, o qual será reutilizado na implementação de outras POU's de controle de motor a serem utilizadas no projeto do MasterTool IEC XE do Estudo de Caso ora proposto.

O bloco funcional deve realizar a limitação da taxa de variação de um sinal analógico

Situação de Aprendizagem 5-4: importação do bloco funcional Motor_PD_S

Importe o bloco funcional Motor_PD_S desenvolvido no Estudo Dirigido 1-1, o qual será reutilizado na implementação de outras POU's de controle de motor a serem utilizadas no projeto do MasterTool IEC XE do Estudo de Caso ora proposto.

O bloco funcional Motor_PD_S permite a operação remota pelo supervisor a partir da reutilização do bloco funcional Motor_PD. A área de interfaceamento da POU com o sistema de supervisão é estabelecida de forma implícita na declaração das variáveis internas do bloco funcional. Essas variáveis são acessadas via rede. O bloco funcional inclui o RESET de

determinados comandos provenientes do supervisor utilizando um bloco de atribuição após o processamento do código principal.

Situação de Aprendizagem 5-5: importação do bloco funcional PID_S

Importe o bloco funcional PID_S desenvolvido no Estudo Dirigido 4-2, o qual será reutilizado no projeto do MasterTool IEC XE do Estudo de Caso ora proposto para implementar a classe de controle PID.

PID_S permite a operação remota pelo supervisor a partir da reutilização do bloco funcional PID da biblioteca inicial.



DICA: para as situações de aprendizagem anteriores utilize os comandos Projeto... Exportar... Importar...

Situação de Aprendizagem 5-6: declaração do bloco funcional Motor_PDI_S

Implemente o bloco funcional Motor_PDI_S permite a operação remota pelo supervisor a partir da reutilização do bloco funcional Motor_PD e Lim_Taxa. Esta POU realiza o comando do conversor de frequência. A declaração do tipo de bloco funcional está mostrada na sequência.

```

1  FUNCTION_BLOCK Motor_PDI_S
2  VAR_INPUT
3      Liga_L, Desliga_L:BOOL; (* Liga/Desliga local *)
4      Freq_Des_L, Freq_Ini:REAL; (* Frequência inicial e desejada local *)
5      Fim_Esc, Taxa:REAL; (* Fim de escala positivo e Taxa/s percentual *)
6      Ciclo:TIME; (* Tempo de ciclo *)
7      Funcionando:BOOL := FALSE; (* Inversor operando *)
8      Sobrecarga, Defeito:BOOL := TRUE; (* Sobrecarga e defeito no motor *)
9      MAN_auto:BOOL := FALSE; (* Modo manual (TRUE) e automático (FALSE) *)
10     Liberacao:BOOL := FALSE; (* Liberação em automático *)
11 END_VAR
12 VAR_OUTPUT
13     Ligar:BOOL; (* Comando ligar inversor *)
14     Ref_Freq:REAL; (* Referência de frequência *)
15     Modo_LR:BOOL; (* Modo remoto (TRUE) e local (FALSE) *)
16     Falha:BOOL; (* Falha no inversor *)
17 END_VAR
18 VAR
19     Liga_S,Desliga_S,Modo_LR_S,MAN_AUTO_S:BOOL; (* Comandos do supervisor *)
20     Freq_Des_S:REAL; (* Frequência desejada remota (supervisor) *)
21     Pulso:F_TRIG; (* Pulso de seleção local/remoto *)
22     M_PD:MOTOR_PD; (* Instância do bloco funcional Motor_PD *)
23     L_T:Lim_Taxa; (* Instância do bloco funcional Lim_Taxa *)
24 END_VAR

```

Situação de Aprendizagem 5-7: declaração do bloco funcional Motor_PDR

Elabore o bloco funcional Motor_PDR a partir da reutilização do bloco funcional Motor_PD para implementar a lógica de reversão de motor. A declaração do tipo de bloco funcional está mostrada na sequência.

```
1  FUNCTION_BLOCK Motor_PDR
2  VAR_INPUT
3      Sel_Dir_Esq:BOOL:=FALSE;(* Seletor de giro esq/dir *)
4      Liga, Desliga:BOOL:=FALSE; (* Liga/desliga motor *)
5      Funcionando_D,Funcionando_E:BOOL:=FALSE; (* Retorno motr *)
6      Sobrecarga, Defeito:BOOL:=TRUE; (* Indicadores de falha *)
7      MAN_auto:BOOL:=FALSE; (* Modo MANUAL=TRUE / AUTO=FALSE *)
8      Liberacao:BOOL:=FALSE; (* Liberação do motor em AUTO *)
9  END_VAR
10 VAR_OUTPUT
11     Ligar_Dir, Ligar_Esq:BOOL:=FALSE; (* Comando para ligar *)
12     Falha:BOOL:=FALSE; (* Motor em falha *)
13 END_VAR
14 VAR
15     Pulsol: R_TRIG; (* Detector 0-1 seletor giro *)
16     Pulso2: F_TRIG; (* Detector 1-0 seletor giro *)
17     M_PD: MOTOR_PD; (* Instância de MOTOR_PD *)
18     TP_1: TP; (* Tempo para reversão *)
19 END_VAR
```

Situação de Aprendizagem 5-8: declaração do bloco funcional Motor_PDRI_S

Elabore o bloco funcional Motor_PDRI_S obtido a partir da reutilização do bloco funcional Motor_PDR e Lim_Taxa, o qual executa o comando reversível de inversor com operação remota pelo supervisor. A declaração do tipo de bloco funcional está mostrada na sequência.

```

1  FUNCTION_BLOCK Motor_PDRI_S
2  VAR_INPUT
3      Sel_Dir_Esq_L:BOOL:=FALSE;(* Seletor de giro esq/dir local *)
4      Liga_L, Desliga_L:BOOL; (* Liga/desliga motor *)
5      Freq_Des_L, Freq_Ini:REAL; (* Frequência inicial e desejada local *)
6      Fim_Esc, Taxa:REAL; (* Fim de escala positivo e Taxa/s percentual *)
7      Ciclo:TIME; (* Tempo de ciclo *)
8      Funcionando_D,Funcionando_E:BOOL:=FALSE; (* Retorno motr *)
9      Sobrecarga, Defeito:BOOL:=TRUE; (* Indicadores de falha *)
10     MAN_auto:BOOL:=FALSE; (* Modo MANUAL=TRUE / AUTO=FALSE *)
11     Liberacao:BOOL:=FALSE; (* Liberação do motor em AUTO *)
12 END_VAR
13 VAR_OUTPUT
14     Ligar_Dir, Ligar_Esq:BOOL; (* Comando para ligar *)
15     Ref_Freq:REAL; (* Referência de frequência *)
16     Modo_LR:BOOL; (* Modo remoto (TRUE) e local (FALSE) *)
17     Falha:BOOL:=FALSE; (* Motor em falha *)
18 END_VAR
19 VAR
20     Sel_Dir_Esq_S:BOOL; (* Seleciona sentido de giro via supervisor *)
21     Liga_S,Desliga_S,Modo_LR_S,MAN_AUTO_S:BOOL; (* Comandos do supervisor *)
22     Freq_Des_S:REAL; (* Frequência desejada remota (supervisor) *)
23     Pulso:F_TRIG; (* Pulso de seleção local/remoto *)
24     M_PDR:MOTOR_PDR; (* Instância do bloco funcional Motor_PD *)
25     L_T:Lim_Taxa; (* Instância do bloco funcional Lim_Taxa *)
26     Memoria:SR; (* Memória comando desligar *)
27 END_VAR

```

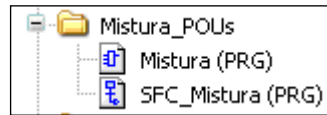
Situação de Aprendizagem 5-9: declaração do programa MISTURA

Elabore o programa MISTURA, o qual será implementado mesclando redes do FBD com diagramas do SFC. Ele realiza a instanciação dos blocos de controle dos equipamentos da unidade misturadora e o seqüenciamento automático de partida e parada. As redes (FBD) contêm as instâncias, juntamente com as conexões de dados e a lógica básica. O comando remoto é realizado via supervisor. As redes do FBD do programa MISTURA estão assim divididas:

- Rede_1 – Eclusa EC1 (formada por instâncias dos FBs Monitoração_Analógica, PID_S e Motor_PDI_S);
- Rede_2 – Transportador TC1 (a base é o bloco funcional Motor_PD_S);
- Rede_3 – Eclusa EC2 (formada por instâncias dos FBs Monitoração_Analógica, PID_S e Motor_PDI_S);
- Rede_4 – Transportador TC2 (a base é o bloco funcional Motor_PD_S);
- Rede_5 – Misturadora (a base é o bloco funcional Monitoração_Analógica).

O programa SFC_MISTURA (seqüenciamento em automático) implementa os automatismos do processo, separando a operação automática das demais funcionalidades (comando e proteção). A seleção do modo manual no SFC permite o comando individual dos equipamentos. Nesta implementação foi adotada a técnica de considerar todas as condições de RESET do diagrama

em cada transição. A declaração do tipo de programa (detalhe da árvore de POU's) está mostrada na sequência.

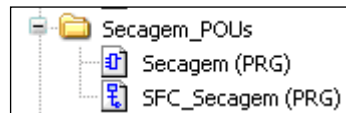


Situação de Aprendizagem 5-10: declaração do programa SECAGEM

Elabore o programa SECAGEM, o qual será implementado mesclando redes do FBD com diagramas do SFC. Ele realiza a instanciação dos blocos de controle dos equipamentos da unidade secagem e o seqüenciamento automático de partida e parada. As redes (FBD) contém as instâncias, juntamente com as conexões de dados e a lógica básica. O comando remoto é realizado via supervisor. As redes do FBD do programa SECAGEM estão assim divididas:

- Rede_1 – Ventilador VT1 (a base é o bloco funcional Motor_PD_S);
- Rede_2 – Transportador TE1 (a base é o bloco funcional Motor_PDRI_S);
- Rede_3 – Controle de temperatura PR1 (formada por instâncias dos FBs Monitoração_Analógica e PID_S);
- Rede_4 – Medições analógicas (formada por instâncias do FB Monitoração_Analógica).

A rede SFC_SECAGEM (seqüenciamento em automático) implementa os automatismos do processo, separando a operação automática das demais funcionalidades (comando e proteção). A seleção do modo manual no SFC permite o comando individual dos equipamentos. Nesta implementação foi adotada a técnica de considerar todas as condições de RESET do diagrama em cada transição. Quando a variável SFC_Secagem.Reset for TRUE o scan do programa forçará o diagrama para o passo inicial. A declaração do tipo de programa (detalhe da árvore de POU's) está mostrada na sequência.

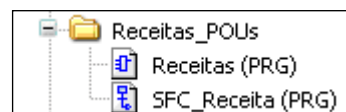


Situação de Aprendizagem 5-11: declaração do programa RECEITAS

Elabore o programa RECEITAS, o qual será implementado mesclando redes do FBD com diagramas do SFC. Ele gerencia o funcionamento em automático dos demais programas. O programa está dividido em duas partes:

- Rede_1 (FBD) – Detecção de pulso das balanças;
- SFC_Receita (SFC) – Seqüenciamento em automático.

Para organizar todos os dados da receita importe a estrutura desenvolvida no Estudo Dirigido 1-2 onde foi declarado o tipo de dados Dados_Receita no nível do recurso. A declaração do tipo de programa (detalhe da árvore de POU's) está mostrada na sequência.



DICA: consulte o Instrutor para orientações adicionais.

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.