



[www.altus.com.br](http://www.altus.com.br)

# APOSTILA DE TREINAMENTO

## SÉRIE HADRON XTORM



---

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, ele 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 dele. 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](http://www.ethercat.org)).

#### **DIREITOS AUTORAIS**

Nexto, Hadron Xtorm e MasterTool são marcas registradas da Altus Sistemas de Automação S.A. Windows, Windows NT e Windows Vista são marcas registradas da Microsoft Corporation.

#### **NOTIFICAÇÃO DE USO DE SOFTWARE ABERTO**

Para obter o código fonte de componentes de software contidos neste produto que estejam sob licença GPL, LGPL, MPL, entre outras, favor entrar em contato através do e-mail [opensource@altus.com.br](mailto: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	x
O Curso MP12	xi
Estruturação do Tutorial	xii
Ementa do curso	xii
Referências	xiv
Documentos relacionados	xiv
Avaliação do curso	xv
Terminologia	xv
Convenções Utilizadas	xvi
Suporte Técnico	xvii
Base de Conhecimento ALTUS	xviii
Cursos ALTUS	xviii
Revisões deste Tutorial	xviii
1. Hardware da Hadron Xtorm	1-1
Características Técnicas da Hadron Xtorm	1-1
O que é a Hadron Xtorm?	1-1
Porque a Hadron Xtorm?	1-1
Quais São Seus Diferenciais?	1-2
Quais são as principais características do produto?	1-3
Instalação da Hadron Xtorm	1-7
Instalação Mecânica	1-7
Instalação Elétrica	1-11
Instalação de Rede	1-15
Instalação do Cartão de Memória	1-19
Estudo Dirigido 1-1: anteprojeto de uma rede RS-422	1-20
Projeto do Painel Elétrico	1-20
Manutenção da Hadron Xtorm	1-21
Diagnósticos do Módulo UCP	1-21
Diagnósticos via Função	1-31
Estudo Dirigido 1-2: diagnóstico via Função	1-31
Diagnósticos dos Módulos de E/S	1-33
Manutenção Preventiva	1-37
2. Software da Hadron Xtorm	2-1
O Projeto no MasterTool Xtorm	2-1
Organização e Acesso à Memória	2-1
Perfis de Projeto	2-1
Novo Projeto	2-3
Acrescentando Módulos	2-10
Os Elementos de Software IEC61131-3	2-12
Unidades de Organização de Programa	2-12
Tarefas	2-14
Parametrização da Aplicação	2-20
Configurando a UCP	2-20
Bibliotecas	2-21
Inserindo uma Instância de Protocolo	2-22
Compilação, Execução e Teste da Aplicação	2-33

---

Compilando um Projeto	2-33
Login	2-34
Modo Run	2-36
Modo Stop	2-38
Monitoração, Escrita e Forçamento de Variáveis	2-39
Variáveis utilizadas em mais de uma origem	2-40
Logout	2-40
Modo Simulação	2-41
Upload do Projeto	2-43
Estados de Operação da UCP	2-44
Estudo Dirigido 2-1: criação de um projeto no MasterTool Xtorm	2-46
3. Configuração da Hadron Xtorm	3-1
Configuração da UCP	3-1
Parâmetros Gerais	3-1
Sincronismo de Tempo	3-4
Pontos Internos	3-6
Conversão de Engenharia	3-8
Alarmes	3-9
Agrupamento de Eventos	3-9
Configuração das Interfaces Seriais	3-11
COM 1	3-11
COM 2	3-11
Configuração das Interfaces Ethernet	3-13
Interfaces Ethernet Locais NET 1 à NET 6	3-13
Configuração dos Pontos Duplos	3-15
Configuração dos Protocolos	3-16
Fila de Eventos da UCP	3-16
Interceptação de Comandos Oriundos do Centro de Controle	3-18
Estudo Dirigido 3-1: aplicação do interceptador de comandos	3-19
MODBUS RTU Mestre	3-19
MODBUS RTU Escravo	3-24
MODBUS Ethernet	3-26
MODBUS Ethernet Cliente	3-27
MODBUS Ethernet Servidor	3-32
DNP3 Ethernet Cliente	3-34
DNP3 Ethernet Servidor	3-41
IEC 60870-5-104 Cliente	3-49
IEC 60870-5-104 Servidor	3-56
IEC 61850 Servidor	3-62
Desempenho da Comunicação	3-77
Desempenho do Sistema	3-79
Escaneamento de E/S	3-79
Cartão de Memória	3-80
Configuração do Relógio RTC	3-81
Funções de Leitura do RTC	3-81
Blocos Funcionais e Funções de Escrita e Configuração do RTC	3-82
Estruturas de Dados do RTC	3-82
Estudo Dirigido 3-2: leitura e configuração do relógio	3-83
Memória de Arquivos de Usuário	3-84
Configuração do Cartão de Memória	3-87
Menu Informativo e de Configuração da UCP	3-91
Configuração via Elementos de Software	3-92
Atualização de Entradas e Saídas	3-92
Timer Retentivo	3-93
Timer Não-Redundante	3-94
Log de Usuário	3-95
Estudo Dirigido 3-3: configuração via elementos de software	3-97

Gerenciamento de Usuários e Direitos de Acesso	3-98
Gerenciamento de Usuários	3-98
Gerenciamento de Direitos de Acesso	3-103
Gerenciamento de Usuários e Direitos de Acesso da UCP	3-104
4. Redundância na Hadron Xtorm	4-1
Redundância Hot-Standby	4-1
Configurações da UCP Redundante	4-3
Identificação de uma UCP HX3040	4-3
Princípio de Funcionamento	4-3
Projeto Redundante Único	4-3
Estrutura do Projeto Redundante	4-3
Mapeamentos Múltiplos	4-7
Estruturas de Dados de Diagnósticos, Comandos e Usuário	4-8
Serviços de Sincronização Cíclicos através de Canais de Sincronismo de Redundância	4-8
Serviços de Sincronização Esporádicos Através dos Canais de Sincronismo de Redundância	4-9
Redes Ethernet Redundantes com NIC Teaming	4-10
Métodos de Troca de IP	4-11
Uso Combinado de NIC Teaming e IP Ativo	4-12
Estados de uma UCP Redundante	4-12
Comandos do menu de redundância do visor da UCP	4-15
Transições entre Estados de Redundância	4-15
Primeiros Instantes em Estado Ativo	4-19
Falhas mais Comuns Causadoras de Switchover Automático entre UCPs	4-19
Falhas Associadas à Switchover entre UCPs Gerenciados pelo Usuário	4-20
Tolerância a Falhas	4-21
Overhead da Redundância	4-22
Programação de uma UCP Redundante	4-24
Wizard para Criação de um Novo Projeto Redundante	4-24
Configuração do projeto com redundância de UCP	4-29
Configurações das Portas Ethernet da UCP HX3040 (NET 1 até NET 6)	4-29
Configurações de I/O Drivers	4-30
Configurações da MainTask	4-30
Configurações da ProtTask	4-39
GVLs com Variáveis Simbólicas Redundantes	4-40
POUs do Tipo Programa com Variáveis Simbólicas Redundantes	4-40
Utilização de Breakpoints em Sistemas Redundantes	4-40
Limitações na Programação de uma UCP Redundante	4-40
Obtendo o Estado da Redundância de uma UCP	4-41
Leitura de Diagnósticos Não-Redundantes	4-41
Carga de Programas em uma UCP Redundante	4-41
Carga Inicial de um Projeto Redundante	4-41
Conexão do MasterTool com uma UCP HX3040 de uma UCP Redundante	4-43
Carga de Modificações em um Projeto Redundante	4-44
Carga de Modificações Off-Line e On-Line	4-44
Carga On-Line de Modificações	4-45
Carga Off-Line de Modificações com Interrupção do Controle do Processo	4-45
Manutenção de UTRs Redundantes	4-47
Mensagens de Advertência do MasterTool	4-47
Interação com a Redundância Através do Visor Gráfico da UCP HX3040	4-47
Estrutura de Diagnósticos da Redundância	4-48
Logs de Eventos da Redundância	4-50
5. Aplicações, Avaliação e Encerramento	5-1
Estudos Dirigidos no MasterTool Xtorm	5-1
Estudo Dirigido 5-1: Projeto-exemplo no MasterTool Xtorm	5-1
Estudo Dirigido 5-2: Utilização do Recurso de Retentividade na Hadron Xtorm	5-3
Estudos de Caso da Hadron Xtorm	5-5
Estudo de Caso 5-1: modernização da UHE Fontes Novas	5-5

Estudo de Caso 5-2: modernização da UHE Santa Branca	5-6
Estudo de Caso 5-3: modernização da PCH Pai Joaquim	5-6
Resolução dos Estudos Dirigidos	5-8
Resolução do Estudo Dirigido 1-1	5-8
Resolução do Estudo Dirigido 1-2	5-8
Resolução do Estudo Dirigido 2-1	5-8
Resolução do Estudo Dirigido 3-1	5-9
Resolução do Estudo Dirigido 3-2	5-10
Resolução do Estudo Dirigido 3-3	5-11
Resolução do Estudo Dirigido 5-1	5-14
Resolução do Estudo Dirigido 5-2	5-14
Avaliação do Treinamento	5-17
Parabéns...	5-17
Glossário	A

# Figuras

Figura 1-1. Os módulos Hadron Xtorm	1-1
Figura 1-2. Ilustração dos recursos avançados de diagnóstico	1-3
Figura 1-3. Posicionamento da parte inferior do módulo em relação ao bastidor	1-7
Figura 1-4. Movimento para Encaixe do Módulo no Bastidor	1-8
Figura 1-5. Módulo Corretamente Encaixado no Bastidor	1-8
Figura 1-6. Movimentos do cursor de fixação e do módulo para remoção deste	1-9
Figura 1-7. Módulo sendo Removido do Bastidor	1-9
Figura 1-8. Inserção da Tampa do Conector do Bastidor	1-10
Figura 1-9. Remoção da Tampa de Conector de Bastidor	1-11
Figura 1-10. Área da Placa de Circuito Impresso (HX9001 e HX9003)	1-12
Figura 1-11. Borne Mola	1-13
Figura 1-12. Borne Mola do Módulo Fonte	1-13
Figura 1-13. Borne Mola do Módulo de E/S (I/O)	1-13
Figura 1-14. Montagem dos cabos no conector de 10 vias	1-14
Figura 1-15. Diagrama Elétrico da UCP da Série Hadron Xtorm	1-15
Figura 1-16. Exemplo de Sincronismo para Sistema com Expansões	1-18
Figura 1-17. Exemplo de Sincronismo para Sistema Redundante	1-18
Figura 1-18. Conector de Interligação para o Sinal IRIG-B	1-19
Figura 1-19. Inserção do Cartão de Memória na UCP	1-19
Figura 1-20. Botão de Diagnóstico	1-22
Figura 1-21. Visualização dos Diagnósticos da UCP	1-23
Figura 1-22. Tela Inicial	1-25
Figura 1-23. Informações do Sistema	1-26
Figura 1-24. Quantidade de Diagnósticos por módulo e por canal	1-27
Figura 1-25. Lista de Diagnósticos do Módulo/Canal	1-28
Figura 1-26. Tela do Diagnostic Explorer	1-29
Figura 1-27. Função GetTaskInfo	1-31
Figura 1-28. Tela de Status da UCP HX3040	1-32
Figura 1-29. Visualização dos diagnósticos da UCP	1-34
Figura 1-30. Visor do Módulo	1-34
Figura 1-31. Exemplo de Diagnóstico	1-35
Figura 2-1. Novo Projeto	2-4
Figura 2-2. Classificação do Projeto	2-5
Figura 2-3. Opções do Projeto	2-6
Figura 2-4. Opções de projeto com redundância de UCP habilitada	2-7
Figura 2-5. Opções de módulos de E/S do Projeto	2-8
Figura 2-6. Seleção das Características de Perfil do Projeto	2-9
Figura 2-7. Linguagem de Programação	2-10
Figura 2-8. Visualizando a Biblioteca	2-11
Figura 2-9. Acrescentando Módulos	2-11
Figura 2-10. Inserindo POUs	2-12
Figura 2-11. Classificando a POU	2-13
Figura 2-12. Editando a POU	2-14
Figura 2-13. Criando uma Tarefa	2-15
Figura 2-14. Nomeando a Tarefa	2-16
Figura 2-15. Configurando a Tarefa Criada	2-18
Figura 2-16. Vinculando POUs às Tarefas	2-19
Figura 2-17. Configuração da UCP	2-20
Figura 2-18. Configurando a Porta de Comunicação da UCP	2-21
Figura 2-19. Aviso de configuração IP	2-21
Figura 2-20. Inclusão de uma Biblioteca no Projeto	2-22

---

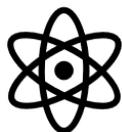
Figura 2-21. Adicionando a Instância	2-23
Figura 2-22. Selecionando o Protocolo	2-24
Figura 2-23. Adicionando a Instância	2-25
Figura 2-24. Selecionando o Protocolo	2-26
Figura 2-25. Adicionando a Instância do DNP3 Servidor	2-27
Figura 2-26. Selecionando o Protocolo DNP3 Servidor	2-28
Figura 2-27. Adicionando a Instância do IEC 60870-5-104 Servidor	2-29
Figura 2-28. Selecionando o Protocolo IEC 60870-5-104 Servidor	2-30
Figura 2-29. Localizando a UCP	2-31
Figura 2-30. Ativando a UCP	2-31
Figura 2-31. Compilando o Projeto	2-33
Figura 2-32. Mensagens da Compilação	2-33
Figura 2-33. Incluindo as Mensagens na Tela	2-34
Figura 2-34. Enviando o Projeto para a UCP	2-35
Figura 2-35. Atualização do Projeto na UCP	2-35
Figura 2-36. Alteração de Variáveis	2-36
Figura 2-37. Primeiro Envio de Aplicação	2-36
Figura 2-38. Iniciando a Aplicação	2-37
Figura 2-39. Programa em Execução	2-38
Figura 2-40. Parando a Aplicação	2-39
Figura 2-41. Interrompendo a Comunicação com a UCP	2-41
Figura 2-42. Modo Simulação	2-42
Figura 2-43. Opção de Upload de Projeto	2-43
Figura 2-44. Selecionando a UCP	2-44
Figura 3-1. Tela de Configuração Parâmetros Gerais da UCP	3-1
Figura 3-2. Configurações do Cartão de Memória	3-4
Figura 3-3. Configuração de Sincronismo de Tempo	3-5
Figura 3-4. Tela de Configuração dos Pontos Internos	3-7
Figura 3-5. Exemplo de Configuração de um Ponto Interno	3-7
Figura 3-6. Tela de Configuração dos Parâmetros de Conversão de Engenharia	3-8
Figura 3-7. Tela de Configuração dos Alarmes	3-9
Figura 3-8. Funcionamento do Agrupamento de Eventos	3-10
Figura 3-9. Tela de Configuração do Agrupamento de Eventos	3-10
Figura 3-10. Tela de Configurações Avançadas das Interfaces Ethernet	3-14
Figura 3-11. Fila de Eventos da UCP	3-17
Figura 3-12. Tela de Configuração Parâmetros Gerais MODBUS RTU Mestre	3-20
Figura 3-13. Tela de Configurações dos Parâmetros Gerais do Dispositivo	3-21
Figura 3-14. Tela de Mapeamentos de Dados MODBUS	3-22
Figura 3-15. Tela de Requisições de Dados MODBUS Mestre	3-23
Figura 3-16. Tela de Configuração do Escravo	3-24
Figura 3-17. Tela de Mapeamentos de Dados MODBUS	3-25
Figura 3-18. Rede de Comunicação MODBUS TCP	3-26
Figura 3-19. Tela de Configuração Parâmetros Gerais MODBUS TCP Cliente	3-28
Figura 3-20. Tela de Configurações dos Parâmetros Gerais do Dispositivo	3-28
Figura 3-21. Tela de Mapeamentos de Dados MODBUS Cliente	3-29
Figura 3-22. Tela de Requisições de dados MODBUS	3-30
Figura 3-23. Tela de Configuração Parâmetros Gerais MODBUS Servidor	3-32
Figura 3-24. Tela de Mapeamentos de Dados MODBUS Servidor	3-33
Figura 3-25. Tela de Parâmetros Gerais do DNP3 Cliente	3-34
Figura 3-26. Tela de Mapeamentos de Dados DNP3 Cliente	3-35
Figura 3-27. Tela de Configurações da Camada de Enlace DNP3 Cliente	3-36
Figura 3-28. Tela de Configurações da Camada de Aplicação DNP3 Cliente	3-38
Figura 3-29. Tela de Mensagens Requisições Customizadas DNP3 Cliente	3-40
Figura 3-30. Tela de Parâmetros Gerais do DNP3 Servidor	3-42
Figura 3-31. Tela de Mapeamentos de Dados DNP3 Servidor	3-42
Figura 3-32. Tela de Configurações da Camada de Enlace DNP3 Servidor	3-44
Figura 3-33. Tela de Configurações da Camada de Aplicação DNP3 Servidor	3-45

Figura 3-34. Tela de Mensagens Não Solicitadas DNP3 Servidor	3-47
Figura 3-35. Tela de Inserção de dispositivo Controlled Station abaixo de um IEC 60870-5-104 Cliente	3-50
Figura 3-36. Tela de Inserção de setor abaixo de um dispositivo Controlled Station IEC 60870-5-104	3-51
Figura 3-37. Tela de Configuração da Camada de Enlace de uma Controlled Station IEC 60870-5-104	3-52
Figura 3-38. Tela de Mapeamentos de Dados de um Setor IEC 60870-5-104	3-53
Figura 3-39. Tela de Configurações da Camada de Enlace de um Setor IEC 60870-5-104	3-54
Figura 3-40. Tela de Configurações da Camada de Aplicação IEC 60870-5-104 Cliente	3-54
Figura 3-41. Tela de Parâmetros Gerais do IEC 60870-5-104 Servidor	3-57
Figura 3-42. Tela de Mapeamento dos Dados do IEC 60870-5-104 Servidor	3-57
Figura 3-43. Tela de Configurações da Camada de Enlace IEC 60870-5-104 Servidor	3-58
Figura 3-44. Tela de Configurações da Camada de Aplicação do IEC 60870-5-104 Servidor	3-59
Figura 3-45. Modelo de dados estruturado	3-63
Figura 3-46. Implementação do modelo de dados	3-64
Figura 3-47. Exemplo de configuração com atributos do tipo STRING	3-65
Figura 3-48. Configurando o nome do Physical Device	3-65
Figura 3-49. Adicionando e configurando o nome do Logical Device	3-66
Figura 3-50. Adicionando um Logical Node	3-67
Figura 3-51. Configurando um Logical Node	3-68
Figura 3-52. Configurando os Datasets	3-69
Figura 3-53. Configurando os Reports	3-70
Figura 3-54. Configurando a comunicação GOOSE Publisher	3-72
Figura 3-55. Importando um arquivo SCL	3-74
Figura 3-56. Configurando a comunicação GOOSE Subscriber	3-75
Figura 3-57. Mapeamento de Variáveis IEC 61850	3-76
Figura 3-58. Blocos de Escrita e Leitura do Relógio	3-81
Figura 3-59. Configura Data e Hora com Milissegundos	3-82
Figura 3-60. Acesso aos Arquivos de Usuário	3-84
Figura 3-61. Transferindo Arquivos	3-85
Figura 3-62. Opções de Utilização	3-86
Figura 3-63. Configurações do Cartão de Memória	3-87
Figura 3-64. Diretório Raiz com Cartão de Memória Inserido	3-89
Figura 3-65. Arquivos Salvos no Cartão de Memória	3-89
Figura 3-66. Exemplo de Operação: Ajuste do Contraste	3-91
Figura 3-67. Função para Atualizar as Entradas	3-92
Figura 3-68. Função para Atualizar as Saídas	3-93
Figura 3-69. Bloco Funcional TOF_RET	3-93
Figura 3-70. Bloco Funcional TON_RET	3-94
Figura 3-71. Comportamento Gráfico do Bloco Funcional TON_RET	3-94
Figura 3-72. Bloco Funcional TP_RET	3-94
Figura 3-73. Comportamento Gráfico do Bloco Funcional TP_RET	3-94
Figura 3-74. Função UserLogAdd	3-95
Figura 3-75. Função UserLogDeleteAll	3-96
Figura 3-76. Configurações do Projeto, Diálogo Usuários	3-99
Figura 3-77. Acrescentar Usuário	3-100
Figura 3-78. Configurações de Projeto, Diálogo Grupos	3-101
Figura 3-79. Acrescentar Grupo	3-102
Figura 3-80. Configurações de Projeto, Diálogo Configuração de Usuários e Grupos	3-103
Figura 3-81. Permissões	3-104
Figura 3-82. Acesso às Abas Usuários e Grupos e Direitos de Acesso	3-105
Figura 3-83. Diálogo Device, Usuários e Grupos	3-106
Figura 3-84. Acrescentar Usuário	3-106
Figura 3-85. Acrescentar Grupo	3-107
Figura 3-86. Dispositivo, Direitos de Acesso	3-109
Figura 4-1. Exemplo de Arquitetura Redundante com UCP HX3040	4-2
Figura 4-2. Exemplo de rede Ethernet redundante, com NIC Teaming	4-11
Figura 4-3. Método IP Ativo – HX3040 Redundante	4-12
Figura 4-4. Máquina de Estados da Redundância	4-16

---

Figura 4-5. Novo Projeto	4-24
Figura 4-6. Configuração Inicial de Hardware	4-25
Figura 4-7. Configuração dos pontos de E/S	4-26
Figura 4-8. Perfil de Projeto e Linguagem Padrão	4-27
Figura 4-9. Linguagem dos Programas Específicos	4-28
Figura 4-10. Parâmetros da Porta Ethernet NET 1	4-29
Figura 4-11. GVL Disable em Modo Online	4-32
Figura 4-12. GVL Module_Diagnostics em Modo Online	4-34
Figura 4-13. GVL Qualities em Modo Online	4-36
Figura 4-14. GVL ReqDiagnostics em Modo Online	4-38
Figura 4-15. GVL System_Diagnostics em Modo Online	4-39

# Introdução



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



## O Curso MP12



O curso MP12 apresenta a Série Hadron Xtorm, a qual consiste na solução no estado-da-arte em Unidades Terminais Remotas (UTRs). A Série apresenta um conjunto ideal de recursos de alto desempenho, facilidades de uso, diagnósticos precisos e completos, design moderno e robusto, além de diversas características inovadoras. A Hadron Xtorm possui uma arquitetura inteligente e versátil, oferecendo modularidade em pontos de entrada e saída (E/S), opções em redundância, protocolos de comunicação de alta velocidade, como IEC 61850, implementação de lógica em conformidade com a norma IEC 61131-3, sendo uma UTR ideal para todas as aplicações de supervisão e controle de sistemas de geração, transmissão e distribuição de energia elétrica, como usinas hidrelétricas (UHEs), e subestações de energia.

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

Os objetivos desse curso são:

- Fazer conhecer as principais características técnicas e de desempenho, os requisitos de instalação, as conexões de rede, as estratégias de manutenção (preventiva e corretiva) com ênfase nos diagnósticos associados e as considerações relativas ao projeto do painel elétrico.
- Apresentar a sequência de tarefas envolvidas na configuração da UCP via software MasterTool Xtorm visando definir o comportamento e modos de utilização dos periféricos e características especiais da UCP.
- Indicar os passos iniciais associados à programação do equipamento incluindo os perfis de projeto, os elementos de software e a instanciação e execução do programa aplicativo.
- Detalhar a funcionalidade de redundância disponível no equipamento para aumentar a tolerância a falhas e, conseqüentemente, aumentar a disponibilidade do sistema de automação.

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

O curso tem uma duração de 24 horas e demanda como pré-requisito, a participação prévia do aluno no Curso MP1 — Introdução à Automação.

O projeto didático do curso é concebido de forma a embasar o ensino direto, utilizando metodologias que propiciam a integração entre a teoria e a prática e favorecem a capacidade de construção e gestão do conhecimento e o autodesenvolvimento contínuo.

Os recursos didáticos associados (Tutorial, Slides e Documentação de Produto) contemplam elementos de instrução de alta qualidade pedagógica e suficientemente compreensíveis, dinâmicos e atrativos. No seu contexto está prevista a flexibilização do ensino para o desenvolvimento de habilidades de autoaprendizagem.

# Estruturação do Tutorial



O Tutorial fornece um suporte didático-pedagógico ao treinamento presencial, possibilitando ao aluno um primeiro contato orientado com o equipamento e enfatizando a instalação, configuração e programação deste.

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

- ▷ 1. Hardware da Hadron Xtorm
- ▷ 2. Software da Hadron Xtorm
- ▷ 3. Configuração da Hadron Xtorm
- ▷ 4. Redundância na Hadron Xtorm
- ▷ 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: Hardware da Hadron Xtorm

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

- ▲ 1. Hardware da Hadron Xtorm
  - ▷ Características Técnicas da Hadron Xtorm
  - ▷ Instalação da Hadron Xtorm
  - ▷ Manutenção da Hadron Xtorm

### Capítulo 2: Software da Hadron Xtorm

O capítulo versa sobre o software de programação MasterTool Xtorm e sua utilização. São fornecidas orientações sobre as configurações básicas para o seu correto funcionamento. O capítulo apresenta a sequência básica de tarefas envolvidas no planejamento de um projeto de automação para aplicação no dispositivo, incluindo os elementos de software da IEC61131-3 e a parametrização e execução da aplicação. Tópicos abordados:

- ▲ 2. Software da Hadron Xtorm
  - ▷ O Projeto no MasterTool Xtorm
  - ▷ Os Elementos de Software IEC61131-3
  - ▷ Parametrização da Aplicação
  - ▷ Compilação, Execução e Teste da Aplicação

### Capítulo 3: Configuração da Hadron Xtorm

Este capítulo detalha as configurações da unidade central de processamento, suas interfaces e protocolos de comunicação, assim como outras funcionalidades (memórias, menus, configuração via elementos de software e gerenciamento e acesso de usuários). Tópicos abordados:

- ▲ 3. Configuração da Hadron Xtorm
  - ▷ Configuração da UCP
  - ▷ Configuração das Interfaces Seriais
  - ▷ Configuração das Interfaces Ethernet
    - Configuração dos Pontos Duplos
  - ▷ Configuração dos Protocolos
  - ▷ Configuração do Relógio RTC
  - Memória de Arquivos de Usuário
  - ▷ Configuração do Cartão de Memória
  - Menu Informativo e de Configuração da UCP
  - ▷ Configuração via Elementos de Software
  - ▷ Gerenciamento de Usuários e Direitos de Acesso

### Capítulo 4: Redundância na Hadron Xtorm

Este capítulo descreve, em detalhes, a funcionalidade de redundância de UCPs da Série Hadron Xtorm. Tópicos abordados:

- ▲ 4. Redundância na Hadron Xtorm
  - Redundância Hot-Standby
  - ▷ Configurações da UCP Redundante
  - ▷ Princípios de Funcionamento
  - ▷ Programação de uma UCP Redundante
  - ▷ Carga de Programas em uma UCP Redundante
  - ▷ Manutenção de UTRs Redundantes

### Capítulo 5: Aplicações, Avaliação e Encerramento

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

- ▲ 5. Aplicações, Avaliação e Encerramento
  - ▷ Estudos Dirigidos no MasterTool Xtorm
  - ▷ Estudos de Caso da Hadron Xtorm
  - ▷ Resolução dos Estudos Dirigidos
  - ▷ Avaliação do Treinamento

## Referências



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

### Documentos relacionados

Para obter informações adicionais sobre a Série Hadron Xtorm podem ser consultados outros documentos (manuais e características técnicas) além deste. Estes documentos encontram-se disponíveis em [www.altus.com.br](http://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 (MU). A tabela a seguir indica a lista de todos os documentos relacionados à Série Hadron Xtorm.

Código	Descrição	Idioma
<b>CE108804</b>	MasterTool Xtorm Technical Characteristics	Inglês Português Espanhol
<b>CT108804</b>	Características Técnicas MasterTool Xtorm	
<b>CS108804</b>	Características Técnicas MasterTool Xtorm	
<b>CE123000</b>	Hadron Xtorm Series Technical Characteristics	Inglês Português Espanhol
<b>CT123000</b>	Características Técnicas Série Hadron Xtorm	
<b>CS123000</b>	Características Técnicas Serie Hadron Xtorm	
<b>CE123100</b>	CPU 6 ETH, 2 SERIALS, IRIG-B, RED Module Technical Characteristics	Inglês Português Espanhol
<b>CT123100</b>	Características Técnicas do Módulo UCP 6 ETH, 2 SERIAIS, IRIG-B, RED.	
<b>CS123100</b>	Características Técnicas del UCP 6 ETH, 2 SERIALES, IRIG-B, RED.	
<b>CE123701</b>	Hadron Xtorm Series Backplane Racks Technical Characteristics	Inglês Português Espanhol
<b>CT123701</b>	Características Técnicas dos Bastidores da Série Hadron Xtorm	
<b>CS123701</b>	Características Técnicas de los Bastidores de la Serie Hadron Xtorm	
<b>CE123200</b>	Redundant Power Supply 60 W Modules Technical Characteristics	Inglês Português Espanhol
<b>CT123200</b>	Características Técnicas dos Módulos Fonte de Alimentação Redundante 60 W	
<b>CS123200</b>	Características Técnicas delos Módulos Fuente de Alimentación Redundante 60 W	
<b>CE123300</b>	Módulo 32 DI 125 Vdc w/ event log Module Technical Characteristics	Inglês Português Espanhol
<b>CT123300</b>	Características Técnicas do Módulo 32 ED 125 Vdc c/ registro de eventos	
<b>CS123300</b>	Características Técnicas del Módulo 32 ED 125 Vdc c/ registro de eventos	
<b>CE123400</b>	16 SD Relay 125 Vdc w/ CBO Module Technical Characteristics	Inglês Português Espanhol
<b>CT123400</b>	Características Técnicas do Módulo 16 SD Relé 125 Vdc c/ CBO	
<b>CS123400</b>	Características Técnicas del Módulo 16 SD Relé 125 Vdc c/ CBO	
<b>CE123310</b>	16 AI Voltage/Current Module Technical Characteristics	Inglês Português Espanhol
<b>CT123310</b>	Características Técnicas do Módulo 16 EA Tensão/Corrente	
<b>CS123310</b>	Características Técnicas del Módulo 16 EA Tensión/Corriente	

<b>CE123311</b>	8 AI RTD Module Technical Characteristics	Inglês
<b>CT123311</b>	Características Técnicas do Módulo 8 EA RTD	Português
<b>CS123311</b>	Características Técnicas del Módulo 8 EA RTD	Espanhol
<b>CE123901</b>	Hadron Xtorm Series Connectors Technical Characteristics	Inglês
<b>CT123901</b>	Características Técnicas dos Conectores da Série Hadron Xtorm	Português
<b>CS123901</b>	Características Técnicas de los Conectores de la Serie Hadron Xtorm	Espanhol
<b>CE123900</b>	Rack Connector Cover Technical Characteristics	Inglês
<b>CT123900</b>	Características Técnicas da Tampa para conector de bastidor	Português
<b>CS123900</b>	Características Técnicas de la Tapa para conector de bastidor	Espanhol
<b>MU223600</b>	Hadron Xtorm Utilization Manual	Inglês
<b>MU223000</b>	Manual de Utilização Hadron Xtorm	Português
<b>MU223300</b>	Manual de Utilización Hadron Xtorm	Espanhol
<b>MU223601</b>	Hadron Xtorm DNP3 Server Device Profile Document	Inglês
<b>MU223602</b>	Hadron Xtorm DNP3 Client Device Profile Document	Inglês
<b>MU223603</b>	Hadron Xtorm IEC 60870-5-104 Server Device Profile Document	Inglês
<b>MU223604</b>	Hadron Xtorm IEC 60870-5-104 Client Device Profile Document	Inglês
<b>MU223605</b>	Hadron Xtorm IEC 61850 Server Device Profile Document	Inglês
<b>MP399609</b>	IEC 61131 Programming Manual	Inglês
<b>MP399048</b>	Manual de Programação IEC 61131	Português
<b>MP399800</b>	Manual de Programación IEC 61131	Espanhol

## 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:** Unidade Central de Processamento é o módulo principal do CP, que realiza o processamento dos dados.

**MasterTool Xtorm:** identifica o programa para microcomputador executável em ambiente WINDOWS®, que permite o desenvolvimento de aplicativos para a UTRs da Série Hadron Xtorm.

## 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...

Suporte e Downloads | Altus

https://www.altus.com.br/suporte

English Portal Blog Base de Conhecimento f in yt ig Q

altus 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**

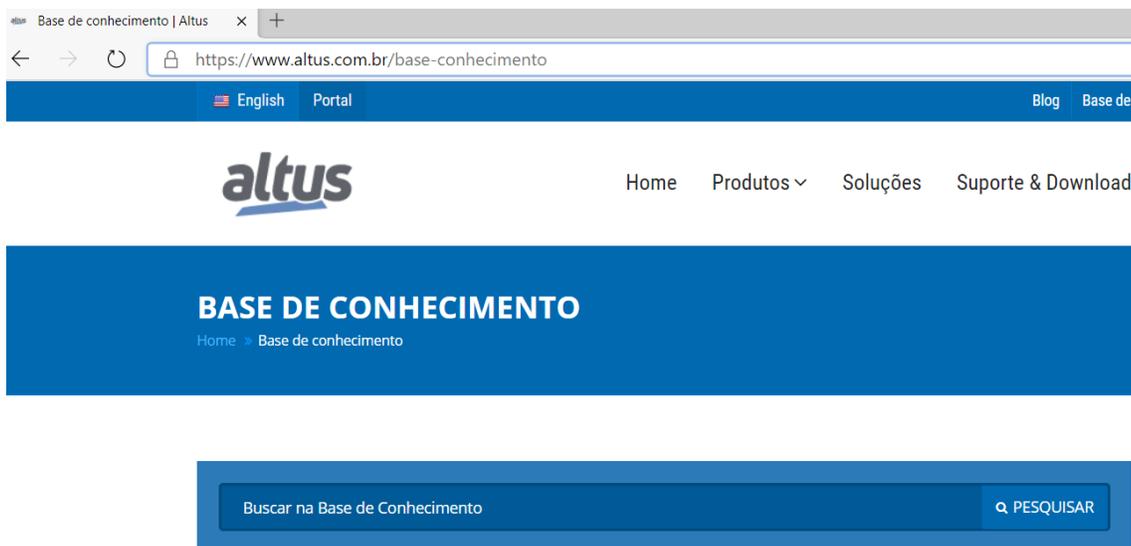
Aqui você encontra tutoriais, dicas e outros materiais de apoio criados por nossos especialistas para qualificar a sua experiência com os produtos e soluções Altus!

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.

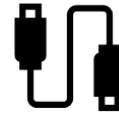
## Revisões deste Tutorial



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	Julho/21	Emissão do Documento

# 1. Hardware da Hadron Xtorm



## Características Técnicas da Hadron Xtorm



### O que é a Hadron Xtorm?

A Série Hadron Xtorm consiste na solução no estado-da-arte em Unidades Terminais Remotas (UTRs). A Série apresenta um conjunto ideal de recursos de alto desempenho, facilidades de uso, diagnósticos precisos e completos, design moderno e robusto, além de diversas características inovadoras. A Hadron Xtorm possui uma arquitetura inteligente e versátil, oferecendo modularidade em pontos de entrada e saída (E/S), opções em redundância, protocolos de comunicação de alta velocidade, como IEC 61850, implementação de lógica em conformidade com a norma IEC 61131-3, sendo uma UTR ideal para todas as aplicações de supervisão e controle de sistemas de geração, transmissão e distribuição de energia elétrica, como usinas hidrelétricas (UHEs), e subestações de energia.



Figura 1-1. Os módulos Hadron Xtorm

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

### Porque a Hadron Xtorm?

Concebidas com base nas normas internacionais IEC 61850 e IEC 61131-3, as Unidades Terminais Remotas (UTR's) da Série Hadron Xtorm são a nova geração da consolidada Série

Hadron, um produto presente há anos no mercado de Energia Elétrica, com vasta base instalada e diversidade de aplicações. Voltada para Geração, Transmissão e Distribuição de Energia Elétrica, a série conta com funcionalidades de destaque, como redundância de fontes de alimentação, redundância de CPUs em um bastidor, protocolos DNP3 e IEC 61850 (MMS e GOOSE) e atendimento ao submódulo 2.7 da ONS.



### Rápida

Processamento veloz e respostas rápidas em aplicações de Geração, Transmissão e Distribuição de Energia Elétrica



### Robusta

Produtos com design prático, de fácil instalação e manutenção. Módulos desenvolvidos para operação em ambientes críticos



### Confiável

Diagnósticos inteligentes e precisos para sistemas que não podem parar. Redundância e módulos de E/S para alto tráfego de dados



### Versátil

Com módulos de E/S de 24 Vdc e 125 Vdc, os produtos Hadron Xtorm podem ser utilizados em sistemas de pequeno a grande porte

## Quais São Seus Diferenciais?

### Recursos para aplicações de alta velocidade

A Série possui uma UCP, baseada em PowerPC de 32 bits, com alta velocidade no processamento e comunicação de dados. Ela atende aos requisitos dos protocolos mais modernos usados nestas aplicações. Funções integradas, como programação online, alta capacidade de memória e dois canais seriais, também são disponibilizadas. Seis portas Ethernet, um slot de cartão de memória e atualização de programa integram ainda os recursos da UCP.

Ela conta com diversas funções integradas, como programação online, alta capacidade de memória e dois canais seriais. Seis portas Ethernet são disponibilizadas para programação, uso em redes IEC 61850, IEC 60870-5-104, DNP3, MODBUS TCP, OPC DA e servidor web embarcado. Um slot de cartão de memória para armazenamento de código-fonte da aplicação e atualização de programa integra os recursos da UCP.

### Redundância para aplicações que não podem parar

Seu design é extremamente robusto, permitindo o uso em aplicações com ambientes severos. Projetada para ter maior durabilidade, pode ser instalada em locais com presença de vibração mecânica e temperatura de operação estendida. A Série está qualificada para aplicações em usinas, em casas de força ou próximo a grandes dispositivos elétricos, possui requisitos elevados de imunidade a descargas eletrostáticas e ruídos eletromagnéticos, comumente presentes nestas aplicações. Seu design oferece estas possibilidades sem comprometimento dos procedimentos de instalação e manutenção.

Confiável, a Série oferece diversas arquiteturas, redundantes ou não, que permitem o seu uso desde sistemas simples até grandes e complexos projetos onde é fundamental ter alta disponibilidade. Além disso, a troca a quente facilita a substituição de módulos sem desenergização do sistema, características essenciais para aplicações que não podem parar.

### Ampla linha de módulos e recursos de expansão

A Série possui uma arquitetura versátil, permitindo a construção de sistemas adequados às necessidades de aplicações independentemente do seu tamanho. Com a opção de expansão de pontos de E/S de forma remota e distribuída, o produto suporta até 8 bastidores remotos, podendo uma única UCP possuir 64 módulos de E/S, ou seja, até 2048 pontos de entradas em sua máxima aplicação.

A linha possui uma ampla linha de módulos de E/S e módulos especiais. Sua alta densidade de pontos e visor gráfico permitem verificar o estado de cada ponto para diagnósticos locais. Estas informações podem ser acessadas pela UCP, por protocolos de rede ou pela ferramenta de

configuração MasterTool Xtorm. Para manusear o produto não são necessárias ferramentas. Os bornes são extraíveis e possuem fácil instalação devido a um sistema com mola para inserção da fiação de campo.

### Desenvolvimento com o MasterTool Xtorm

Ferramenta para configuração, programação, depuração e simulação das aplicações do usuário, o software MasterTool Xtorm oferece flexibilidade e facilidade de uso. Ele permite a importação de dados de planilhas eletrônicas para parametrização de módulos ou mapeamento de variáveis nos protocolos de comunicação disponíveis na UCP. Entre os protocolos e serviços integrados, estão o MODBUS RTU, MODBUS TCP, DNP3, IEC 61850 (MMS e GOOSE), IEC 60870-5-104 e sincronismo de tempo, além do agrupamento de eventos que pode ser realizado de forma gráfica.

O usuário pode conferir informações de diagnóstico de qualquer módulo do sistema diretamente no visor gráfico, com apenas um toque no botão de diagnóstico através do *One Touch Diag*. Outra característica inovadora é o *Multiple Block Storage*. Diversos tipos de memória (voláteis e não voláteis) estão disponíveis nas UCPs da Série Hadron Xtorm, o que fornece melhores opções para cada necessidade.



Figura 1-2. Ilustração dos recursos avançados de diagnóstico

### Quais são as principais características do produto?

#### UCPs

As UCPs apresentam várias funções integradas, programação online, alta capacidade de memória e canais seriais. Estes dispositivos possuem um moderno processador de 32 bits que proporciona um excelente desempenho e a integração de diversas funcionalidades. A UCP conta com porta Ethernet para programação, uso em redes MODBUS, DNP3, IEC 60870-5-104, IEC 61850 e servidor web embarcado. Apresentam duas interfaces seriais para conexão de IHMs locais e redes MODBUS. As UCPs têm uma posição de cartão de memória para hospedagem das páginas da web, código fonte da aplicação, atualização de firmware entre outras características.

#### Módulos

Os módulos apresentam E/S de alta densidade. Cada módulo de E/S apresenta um visor para diagnósticos locais no qual é mostrado o estado de cada ponto de E/S. Há também os diagnósticos multifuncionais sobre o status dos módulos. Todas as informações de diagnóstico

também podem ser acessadas remotamente pela UCP, protocolos de comunicação ou pela ferramenta de configuração MasterTool Xtorm.

### **Barramento de Alta Velocidade**

A arquitetura da Série Hadron Xtorm apresenta um barramento no estado da arte de barramentos baseados em Ethernet de 100 Mbps. A alta taxa de transferência permite a atualização de grandes quantidades de entradas em um curto período de tempo. Os módulos são endereçados e identificados automaticamente evitando erros durante a configuração da aplicação e manutenção de campo. O barramento fornece características especiais que permitem redundância da UCP no mesmo bastidor entre outras características listadas abaixo.

- Endereçamento e identificação de módulo automática;
- Troca a quente;
- Barramento serial baseado em Ethernet 100 Mbps;
- Sincronização de tempo para atualização de E/S ou estampa de tempo precisa;
- Solução de hardware de chip único.

### **Troca a Quente**

A característica de troca a quente permite a substituição de módulos sem desenergização do sistema. A UCP mantém o controle de todo o processo e os módulos podem ser substituídos sempre que necessário.

### **Alta Disponibilidade**

A Série Hadron Xtorm oferece diversas arquiteturas diferentes de redundância, onde UCPs, Fontes de Alimentação e Interfaces de Rede podem ser montadas em uma aplicação redundante. Com esta flexibilidade o sistema pode ser ajustado desde sistemas simples sem nenhuma redundância até complexas e críticas aplicações onde a alta disponibilidade é essencial.

### **Diagnósticos Avançados**

Cada módulo contém seus próprios diagnósticos: as UCPs, Interfaces de Rede, Fontes de Alimentação e módulos de E/S apresentam vários diagnósticos disponíveis. Cada módulo tem um visor multifuncional que informa o status do mesmo. Além disto, cada módulo conta com um botão na sua parte frontal para fornecer informações de diagnósticos diferentes para a equipe de manutenção. Estes diagnósticos podem ser monitorados no campo através de visores ou através da ferramenta de configuração. Alguns exemplos são:

- Módulo localizado em uma posição incorreta no bastidor;
- Ausência de fonte de alimentação;
- Curto circuito nas saídas;
- Não há configuração para um módulo que necessite ser configurado para operação normal;
- Visualização de tag e descrição de E/S da UTR;
- Visualização de endereço IP.

### **Capacidades**

Na Série Hadron Xtorm o maior bastidor pode conter até 18 módulos, sendo que devemos tomar o cuidado de que a combinação dentre os módulos escolhidos não ultrapasse o limite de corrente da fonte de alimentação do bastidor. A corrente que cada módulo da Série Hadron Xtorm

consumo do barramento pode ser encontrado no documento com as características técnicas de cada módulo, ou então podemos utilizar a funcionalidade de “Configuração e Consumo” do Mastertool Xtorm para montar a arquitetura com os módulos desejados e visualizar o consumo de corrente de cada módulo, a corrente total necessária para os módulos escolhidos e o valor fornecido pela fonte de alimentação. Com esta arquitetura uma UCP única pode controlar 512 pontos de E/S usando apenas um bastidor. É possível expandir para até 16 bastidores (bastidor principal + 15 bastidores de expansão) utilizando a funcionalidade de expansão de barramento. Neste caso, o limite máximo de módulos contabilizados dentre todos os bastidores de expansão não podem ultrapassar 80 módulos.

### Programação da UCP e Atualização de Firmware

A Série Hadron Xtorm permite a programação da UCP e a atualização do firmware através da porta Ethernet da UCP. Esta abordagem oferece algumas funcionalidades, tais como:

- Porta Ethernet multifuncional usada para compartilhar programação, intercâmbio de dados ponto a ponto, protocolo de dispositivo de terceiros na camada de aplicação, intercâmbio de dados das variáveis de rede etc.;
- Acesso direto às variáveis locais da UCP;
- Acesso remoto via interface Ethernet;
- Atualização de firmware via interface Ethernet.

### Características Inovadoras

A Série Hadron Xtorm traz aos usuários diversas inovações na utilização, supervisão e manutenção do sistema. Estas características foram desenvolvidas focando um novo conceito em automação de usinas hidrelétricas e subestações. A lista abaixo mostra algumas destas características que o usuário encontrará na Série Hadron Xtorm.

Tabela 1-1. Inovações da Série Hadron Xtorm

<b>Battery Free Operation</b>	A Série Hadron Xtorm não requer nenhum tipo de bateria para manutenção de memória e operação de relógio de tempo real. Esta funcionalidade é extremamente importante porque reduz a necessidade de manutenção do sistema e permite o uso em locais remotos de difícil manutenção. Além disso, esta característica é ambientalmente correta.	
<b>Multiple Block Storage</b>	Diversos tipos de memória estão disponíveis nas UCPs da Série Hadron Xtorm, oferecendo a melhor opção para cada necessidade. Estas memórias são divididas em memórias voláteis e memórias não voláteis. Para uso de memórias voláteis, as UCPs da Série Hadron Xtorm oferecem variáveis de entrada de representação direta (%I), variáveis de saída de representação direta (%Q), variáveis de memória de representação direta (%M), memória de dados e memória de dados redundantes. Para aplicações que necessitam funcionalidades de memória não volátil, a Série Hadron Xtorm possibilita a utilização de variáveis de representação direta de memória retentiva (%Q), memória retentiva de dados, variáveis de representação direta de memória persistente (%Q), memória persistente de dados, memória de programa, memória de código fonte, sistema de arquivo na UCP (Doc, pdf dados) e interface para cartão de memória.	
<b>One Touch Diag TM</b>	Esta é uma característica exclusiva das UTRs da Série Hadron Xtorm. Através deste novo conceito, o usuário pode checar as informações de diagnóstico de qualquer módulo do sistema diretamente no visor gráfico da UCP, mediante apenas um pressionamento no botão de diagnóstico do respectivo módulo. A OTD é uma poderosa ferramenta de diagnóstico que pode ser usada offline (sem supervisor ou programador) e reduz os tempos de manutenção e comissionamento.	

<b>OFD – On Board Full Documentation</b>	As UCPs da Série Hadron Xtorm têm a capacidade de armazenar a documentação completa do projeto na sua memória. Este é um recurso interessante para fins de backup e manutenção, já que informação completa fica armazenada em um único e seguro local.
<b>ETD – Electronic Tag on Display</b>	Esta funcionalidade possibilita a verificação da tag de qualquer ponto ou módulo de E/S usado no sistema, diretamente no visor gráfico das UCPs. Juntamente com esta informação, o usuário pode também verificar a descrição. Este é um recurso extremamente útil durante a manutenção e resolução de problemas.



### **Pesquisa na Documentação: especificações técnicas da Série Hadron Xtorm**

Consulte nas tabelas de características do produto as especificações associadas aos seguintes itens: Características Comuns, Características Gerais Específicas, Interfaces Seriais, Interfaces Ethernet, IRIG-B, Visor Gráfico, Interface do Cartão de Memória, Compatibilidade com Demais Produtos, Desempenho (Tempos de Aplicação, Tempos para Execução de Instruções e Tempos de Inicialização), assim como os Produtos Relacionados.

Fonte: Manual de Utilização Hadron Xtorm – MU223000!

# Instalação da Hadron Xtorm



## Instalação Mecânica

O detalhamento dos procedimentos relacionados à instalação mecânica da série pode ser conferido no Manual de Utilização Hadron Xtorm – MU223000, incluindo a furação para fixação, montagem e remoção do bastidor. Esta seção sumariza as etapas envolvidas na instalação mecânica.

## Inserção dos Módulos

O exemplo a seguir mostra um módulo da Série Hadron Xtorm genérico de modo que o procedimento deve ser seguido para todos os módulos da Série. As figuras que seguem ilustram os procedimentos de inserção dos módulos no bastidor.

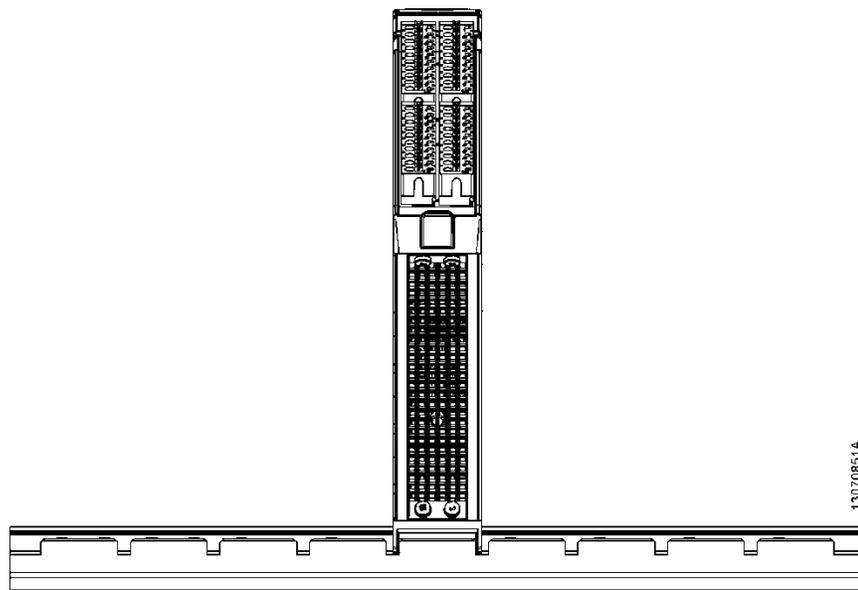


Figura 1-3. Posicionamento da parte inferior do módulo em relação ao bastidor

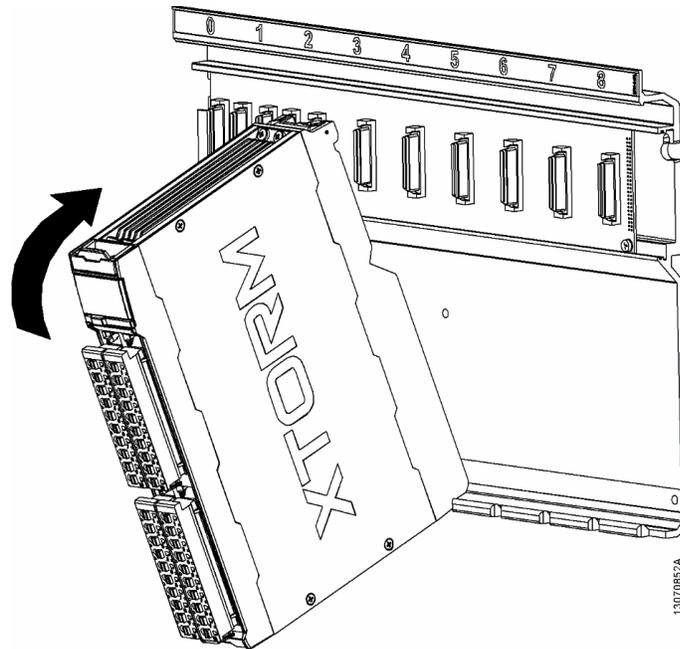


Figura 1-4. Movimento para Encaixe do Módulo no Bastidor

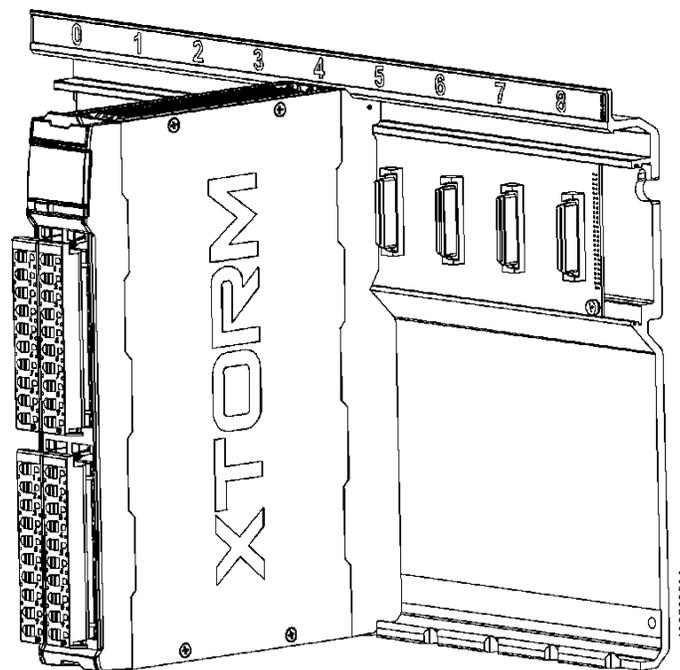


Figura 1-5. Módulo Corretamente Encaixado no Bastidor

### Remoção dos Módulos

O procedimento ilustrado nas figuras a seguir mostra um módulo da Série Hadron Xtorm genérico e deve ser seguido para todos os módulos da Série.

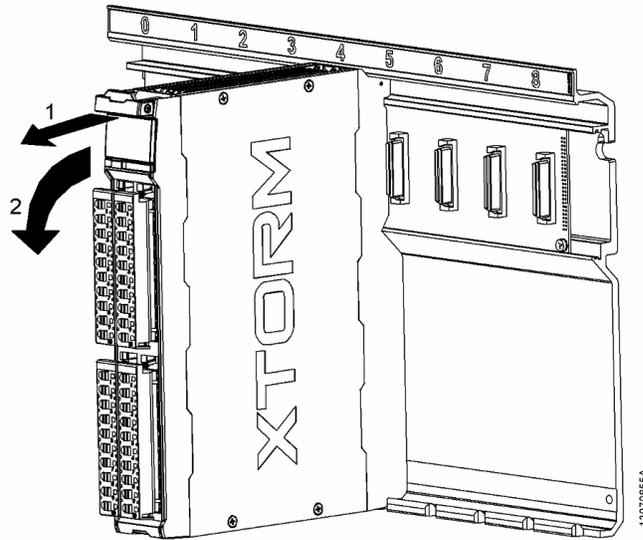


Figura 1-6. Movimentos do cursor de fixação e do módulo para remoção deste

Caso o usuário siga corretamente o procedimento descrito acima, o módulo será desconectado do barramento e basta retirar o mesmo conforme indicado na figura a seguir. Deve ser removido apenas um módulo por vez. Após a conclusão deste procedimento, deve-se aguardar no mínimo dois segundos para a execução de um novo procedimento de inserção/remoção.

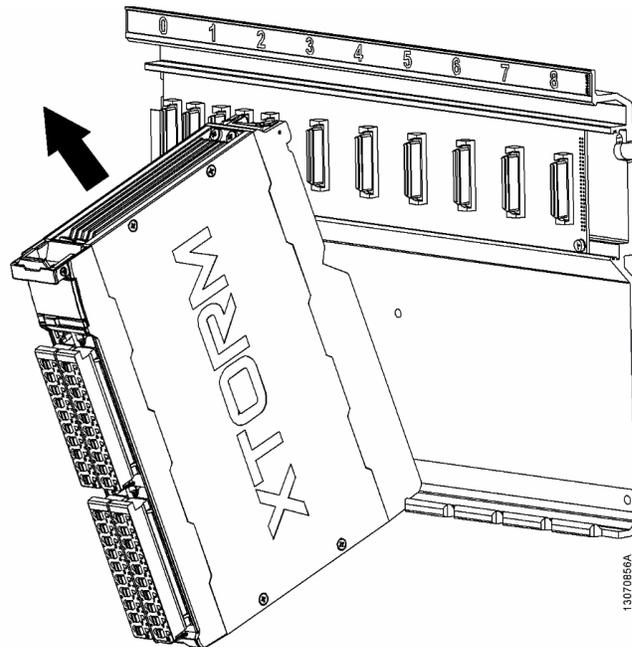


Figura 1-7. Módulo sendo Removido do Bastidor

### Módulos de E/S

Os módulos de entrada e saída da Série Hadron Xtorm possuem algumas características específicas conforme indicado a seguir:

- Bornes de E/S - Os bornes dos módulos de E/S da Série Hadron Xtorm utilizam o sistema de fixação de fixação com bornes tipo mola não necessitando de parafuso para tal fim.
- Identificação - Todos os pinos dos bornes de E/S possuem numeração de 1 a 10. A relação entre o número do pino e a respectiva funcionalidade do mesmo para um módulo específico é definida no documento Características Técnicas do respectivo módulo.
- Diagrama de Instalação - O diagrama de instalação de cada módulo é definido no documento Características Técnicas do respectivo módulo.
- Tampa de Conector de Bastidor - A tampa de conector de bastidor deve ser utilizada nas posições não utilizadas do bastidor para proteção do conector contra contatos indevidos e impurezas.

### Inserção e Remoção da Tampa de Conector de Bastidor

A tampa de conector HX9102 da Série Hadron Xtorm, deve ser encaixada no conector iniciando-se pela parte inferior e finalizando pela parte superior, pressionando-a até encaixar completamente conforme ilustrado na figura a seguir.

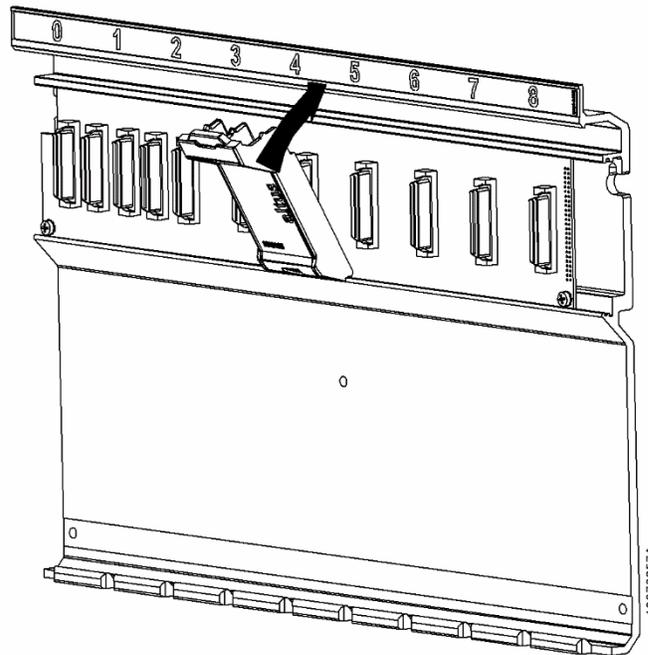


Figura 1-8. Inserção da Tampa do Conector do Bastidor

Para remover a tampa de conector HX9102 da Série Hadron Xtorm, basta puxá-la pela borda superior. A figura a seguir mostra esse procedimento.

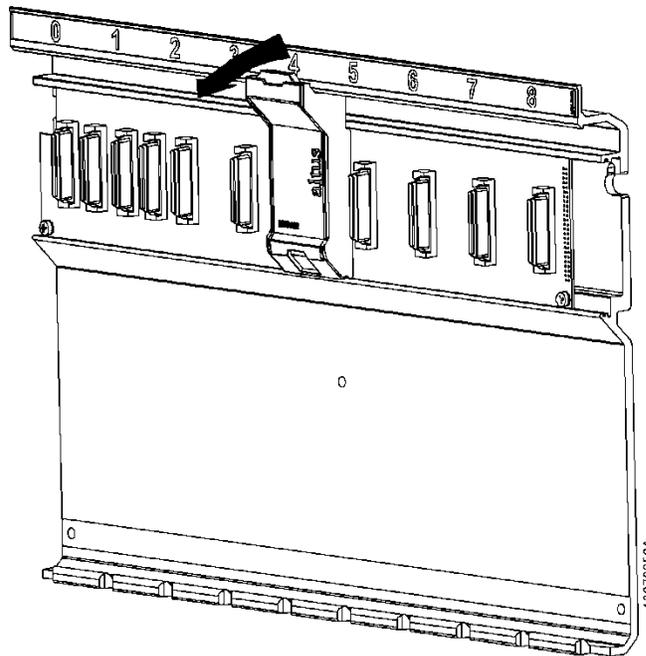


Figura 1-9. Remoção da Tampa de Conector de Bastidor

## Instalação Elétrica

**PERIGO:**

Ao realizar qualquer instalação em um painel elétrico, certifique-se de que a alimentação geral do armário esteja **DESLIGADA**.

**PERIGO:**

Os módulos HX8320, HX1120 e HX2320 utilizam tensões entre 91 e 135 Vdc nas interfaces de alimentação, entradas digitais e saídas digitais, respectivamente. Portanto, deve-se tomar cuidado ao manusear os cabos e bornes destes módulos durante as etapas de instalação, utilização e manutenção, verificando sempre se eles estão em bom estado de conservação, sem cortes ou rompimentos. Para garantir maior segurança, os bornes dos módulos têm classificação IP20.

**PERIGO:**

Nunca se deve substituir um fusível por outro de maior valor de corrente, sob pena de causar sérios danos ao equipamento.

## Segurança Elétrica

No tocante à instalação elétrica é importante considerar os aspectos indicados a seguir relacionados à segurança:

- O pino 5 do borne dos módulos HX8300 e HX8320 é o terra de proteção e obrigatoriamente deve ser ligado ao aterramento local com uma boa conexão, garantindo uma impedância máxima de  $0,1 \Omega$ ;
- A interface serial COM 1 (RS-232/RS-485) da UCP HX3040 possui um conector DB9 que tem classificação IP20, e, portanto exige o operador de risco de choque elétrico ao tocar neste. Entretanto, deve-se atentar para não tocar nos pinos do cabo que estiver ligado à esta interface. E ainda, em função desta ser perigosa em caso de falha simples, ela deve ser ligada apenas a produtos que contenham isolamento superior a 820 Vac desta interface para outras;
- As Placas de Circuito Impresso dos bastidores HX9001 e HX9003, que podem, em condições de falha simples, apresentar tensões perigosas estão representadas nas regiões destacadas em verde na figura a seguir. Portanto, estas regiões devem ser tocadas apenas com o painel desenergizado.

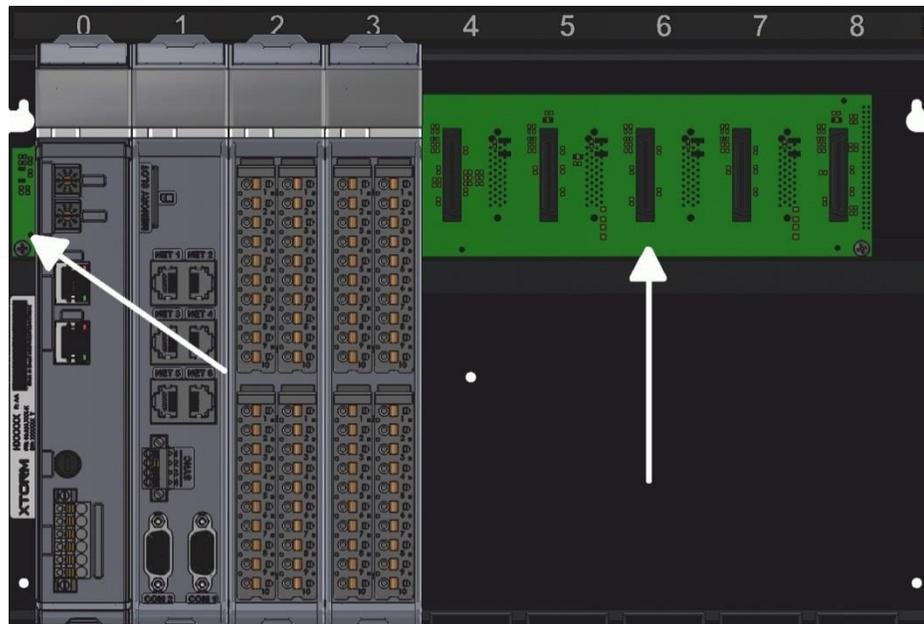


Figura 1-10. Área da Placa de Circuito Impresso (HX9001 e HX9003)

## Bornes com Mola

Este tipo de borne possui um sistema de fixação baseado em uma mola, de elevada confiabilidade, mesmo em ambientes sujeitos a vibração. Para sua montagem recomenda-se utilizar uma chave de fenda com 3,5 mm de largura e cabo isolado. A vantagem de seu emprego é a facilidade e rapidez de montagem dos cabos elétricos. Para montar o cabo no borne insira a chave de fenda no acionador do terminal para abrir a mola do borne, insira o terminal do cabo no borne e retire a chave para fechar o borne. As figuras a seguir ilustram o sistema de fixação dos bornes.

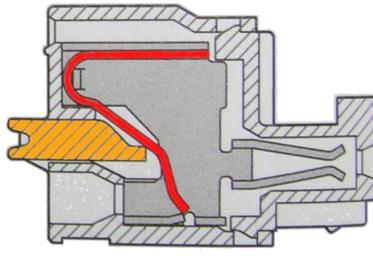


Figura 1-11. Borne Mola

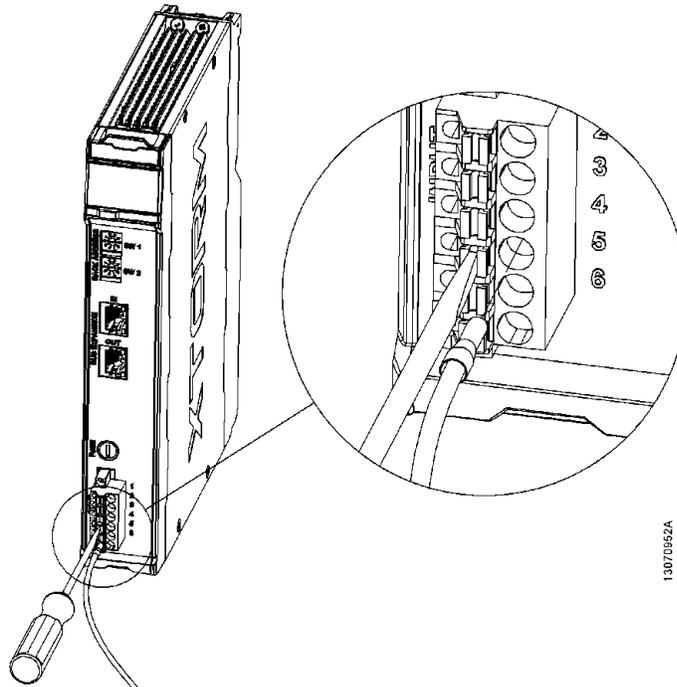


Figura 1-12. Borne Mola do Módulo Fonte

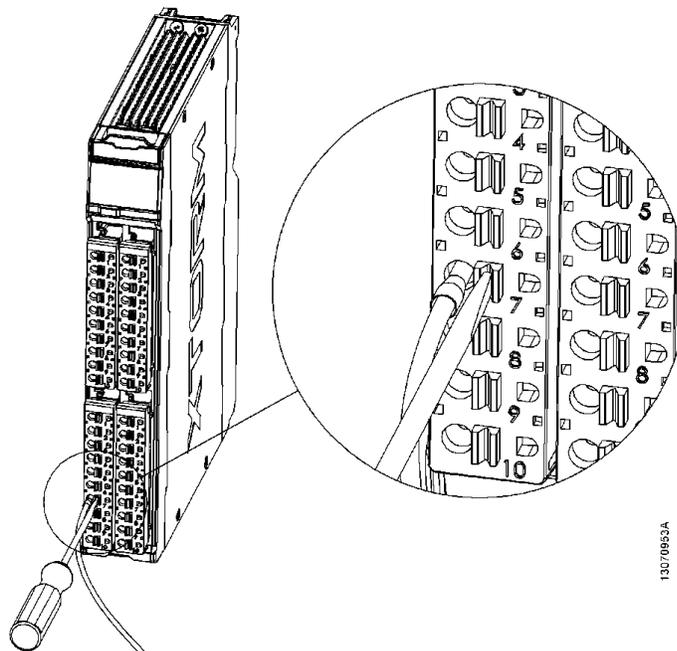


Figura 1-13. Borne Mola do Módulo de E/S (I/O)

## Instalação do Cabeamento

De forma a se obter um comprimento semelhante de todos os fios conectados a um mesmo borne de E/S, indica-se seguir o procedimento previsto no Manual de Utilização Hadron Xtorm – MU223000. É importante ressaltar que o maior fio deverá ser sempre o fio conectado ao pino 1 do borne de E/S. O documento indicado detalha ainda as instruções de instalação dos bornes de 4, 6 e 10 vias, bem como a montagem, fixação e remoção dos cabos. A figura a seguir mostra uma montagem típica no conector de 10 vias.

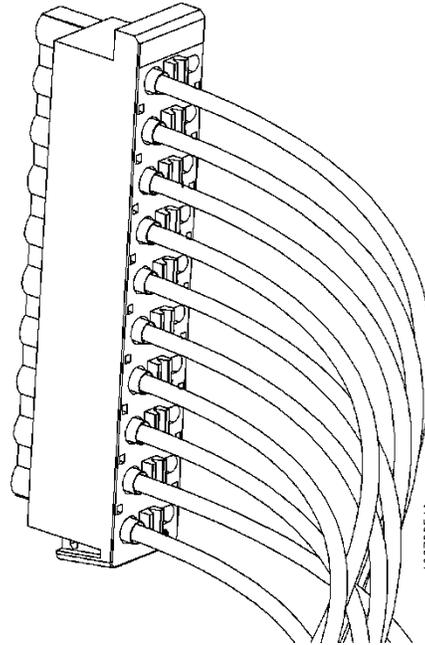


Figura 1-14. Montagem dos cabos no conector de 10 vias

Por fim, os seguintes itens devem ser verificados para garantir a segurança do equipamento e seu correto funcionamento:

- Os cabos junto aos bornes de ligação do painel de montagem devem estar com conexão segura e firme;
- Os bornes de alimentação e aterramento das partes do sistema devem estar firmes e bem conectados, assegurando boa passagem de corrente;
- A conexão do terra dos equipamentos ao terra do painel de montagem deve estar firme e com a bitola de cabo correta, para garantir bom aterramento e imunidade a ruído;
- O encaixe entre os conectores e os módulos deve ser verificado de forma a garantir que ele tenha sido inserido completamente, garantido uma correta conexão;
- As tensões das alimentações devem estar dentro dos valores especificados nas características técnicas. Onde houver alta tensão, colocar etiqueta de aviso e proteções que não permitam o fácil acesso;
- Os fusíveis do sistema devem ser verificados, certificando-se que eles estejam em bom estado e com valor e tipo correto, antes de energizar o sistema.

## Instalação Elétrica UCP

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

A figura a seguir ilustra o diagrama elétrico da UCP da Série Hadron Xtorm instalada em um bastidor da Série Hadron Xtorm. A mesma imagem pode ser visualizada na lateral esquerda da mecânica da UCP. A disposição dos conectores e bornes na figura é meramente ilustrativa.

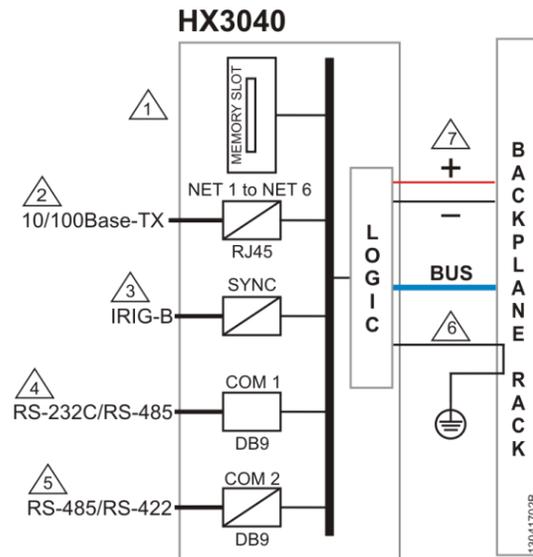


Figura 1-15. Diagrama Elétrico da UCP da Série Hadron Xtorm

#### NOTAS:

1. Interface para cartão de memória.
2. Interface Ethernet padrão 10/100Base-TX para programação, depuração e conexão aos protocolos suportados.
3. Interface IRIG-B padrão para conexão ao sinal de sincronismo de tempo. Recomenda-se a utilização de cabos de seção de 1,5mm<sup>2</sup>.
4. Interface serial padrão RS-232C/RS-485 para conexão à rede MODBUS RTU ou outros protocolos.
5. Interface serial padrão RS-485/RS-422 para conexão à rede MODBUS RTU ou outros protocolos. A escolha do tipo de interface física depende do cabo utilizado.
6. O módulo é aterrado através dos bastidores da Série Hadron Xtorm.
7. A alimentação do módulo é proveniente da conexão ao bastidor, não necessitando de conexões externas.

## Instalação de Rede

### Conexão com a Rede Ethernet

As interfaces isoladas de comunicação NET 1 ao NET 6 possibilitam a conexão com uma rede Ethernet, onde a interface NET 1 (ou NET 2 quando estiver configurada em par NIC Teaming com a NET 1) deve ser usada na comunicação com o MasterTool Xtorm.

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

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

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

Os parâmetros Endereço IP e Máscara de Subrede podem ser visualizados no visor gráfico da UCP através do menu de parâmetros. Inicialmente, deve-se conectar a interface NET 1 da UCP a uma rede ou microcomputador na mesma subrede para comunicação com o MasterTool Xtorm, onde os parâmetros de rede podem ser alterados.

As interfaces Ethernet NET 2 à NET 6 são utilizadas para comunicação Ethernet e vêm configuradas de fábrica com os seguintes parâmetros:

- Endereço IP: 192.168.XX.1;
- Máscara de subrede: 255.255.255.0;
- Endereço do Gateway: 192.168.XX.253.

O parâmetro XX corresponde à 16 na NET 2 e assim por diante até 20 na NET 6.

Os parâmetros de rede das interfaces NET 1 à NET 6 podem ser alterados através do MasterTool Xtorm.

### **ARP Gratuito**

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

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

### **Instalação do Cabo de Rede**

As portas Ethernet da UCP da Série Hadron Xtorm, identificadas no painel por NET 1 ao NET 6, possuem pinagem padrão, sendo a mesma utilizada, por exemplo, em computadores pessoais.

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

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

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

### **Conexão com a Rede Serial**

No tocante à conexão das interfaces seriais, as seguintes opções estão disponíveis:

- A interface não isolada de comunicação COM 1 possibilita a conexão com uma rede RS-232C ou RS-485;
- Para conexão a uma rede RS-485 sem terminação na interface COM 1, deve-se conectar os terminais identificados do cabo AL-1763 nos respectivos bornes do dispositivo;
- A interface COM 1 não permite terminação interna. Para conexão a uma rede RS-485, utilizando a terminação interna, deve ser utilizada a interface COM 2;
- Para conexão a uma rede RS-485, utilizando a terminação externa da interface COM 1, deve-se conectar os terminais identificados do cabo AL-1763 nos respectivos bornes do dispositivo;
- A interface isolada de comunicação COM 2 possibilita a conexão com uma rede RS-485/422 via conector DB9 fêmea da UCP Hadron Xtorm;
- Para conexão a uma rede RS-485 sem terminação na interface COM 2, deve-se conectar os terminais identificados do cabo AL-1763 nos respectivos bornes do dispositivo;
- Para conexão a uma rede RS-485, utilizando a terminação interna da interface COM 2, deve-se conectar os terminais identificados do cabo AL-1763 nos respectivos bornes do dispositivo;
- Para conexão a uma rede RS-485, utilizando a terminação externa da interface COM 2, deve-se conectar os terminais identificados do cabo AL-1763 nos respectivos bornes do dispositivo;
- Para conexão a uma rede RS-422 sem terminação na interface COM 2, deve-se conectar os terminais identificados do cabo AL-1763 nos respectivos bornes do dispositivo;
- Para conexão a uma rede RS-422, utilizando a terminação interna da interface COM 2, deve-se conectar os terminais identificados do cabo AL-1763 nos respectivos bornes do dispositivo;
- Para conexão a uma rede RS-422, utilizando a terminação externa da interface COM 2, deve-se conectar os terminais identificados do cabo AL-1763 nos respectivos bornes do dispositivo.



### **Pesquisa na Documentação: conexão com a rede serial**

Consulte, na documentação do produto, os requisitos (cabearamento, pinagem etc.) relacionados à conexão com a rede Serial (COM 1) via interfaces RS-232C e RS-485, bem como à conexão com a rede Serial (COM 2) via interfaces RS-485 e RS-422.

Fonte: Manual de Utilização Hadron Xtorm – MU223000!

### **Conexão com Sinal IRIG-B**

Para aplicações onde é necessário o sincronismo de tempo com outros equipamentos, a UCP da Série Hadron Xtorm possui uma porta de entrada e uma porta de saída para o sinal IRIG-B. Através da porta de entrada, a UCP irá receber os dados de sincronismo de tempo e sincronizar com o seu relógio interno. E através da porta de saída, a UCP Xtorm poderá sincronizar outros equipamentos, retransmitindo na porta de saída, o sinal recebido na porta de entrada. Para os casos em que for utilizada a expansão de barramento da Série Hadron Xtorm, o sincronismo do relógio interno da UCP se encarregará de sincronizar as estampas de tempo para a geração de eventos dos módulos de E/S presentes na expansão de barramento, conforme a figura a seguir.

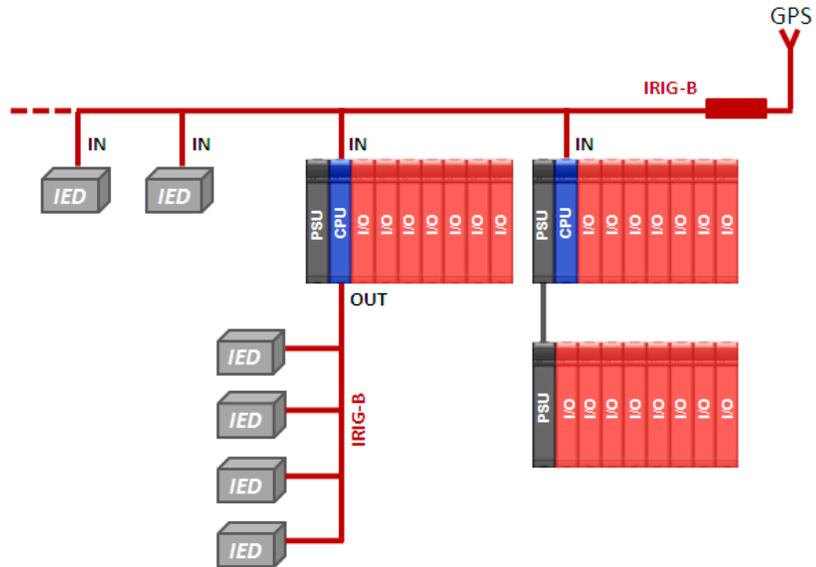


Figura 1-16. Exemplo de Sincronismo para Sistema com Expansões

Para os casos de aplicações onde for utilizada redundância de UCP no mesmo bastidor, o sinal de sincronismo de tempo gerado pelo GPS deve ser conectado na entrada IRIG-B de cada uma das UCPs, independente de qual UCP esteja em modo ativo e qual esteja em modo reserva, garantindo assim o sincronismo de tempo do sistema. Caso seja necessário que a UCP execute a função de repetidora do sinal IRIG-B para outros IEDs, deve-se utilizar a saída de uma UCP somente. Na figura a seguir podemos visualizar um exemplo deste tipo de arquitetura.

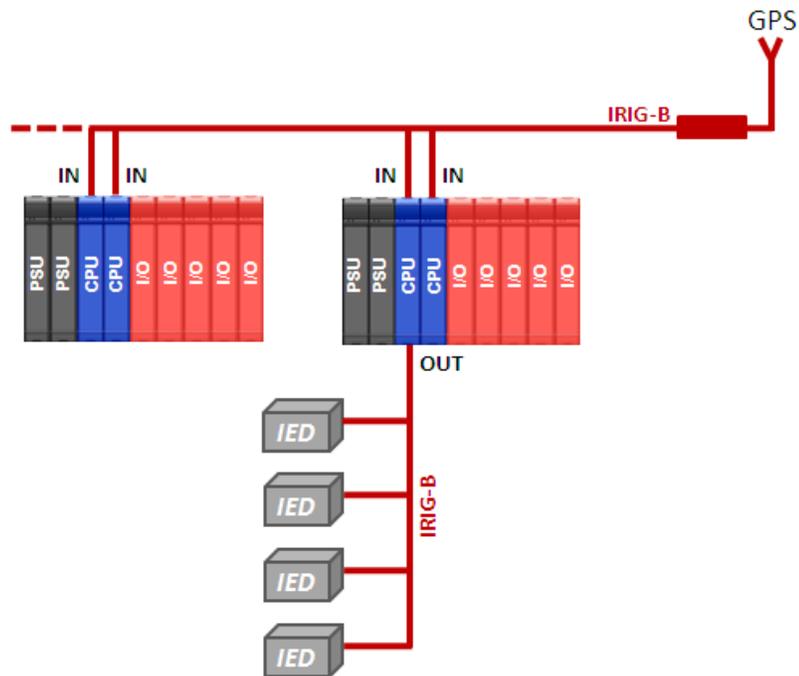


Figura 1-17. Exemplo de Sincronismo para Sistema Redundante

Para conexão a um sinal de sincronismo de tempo IRIG-B deve-se conectar os terminais do cabo contendo o sinal IRIG-B nos respectivos bornes da UCP HX3040. Na figura a seguir podemos visualizar a representação do conector IRIG-B da UCP HX3040.

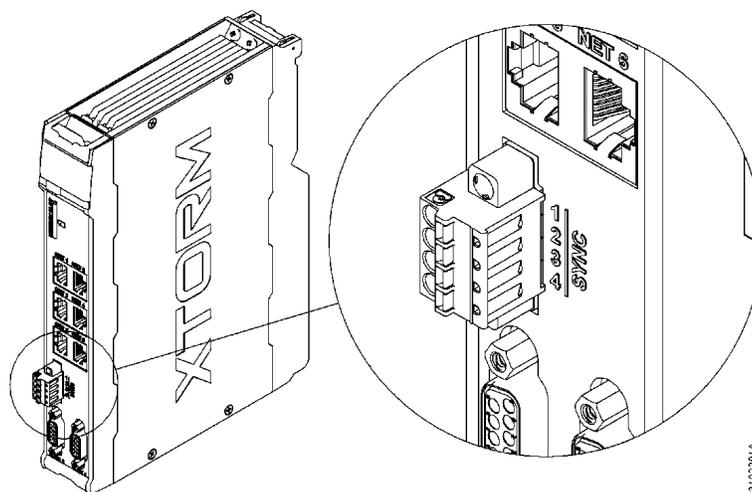


Figura 1-18. Conector de Interligação para o Sinal IRIG-B

### Instalação do Cartão de Memória

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

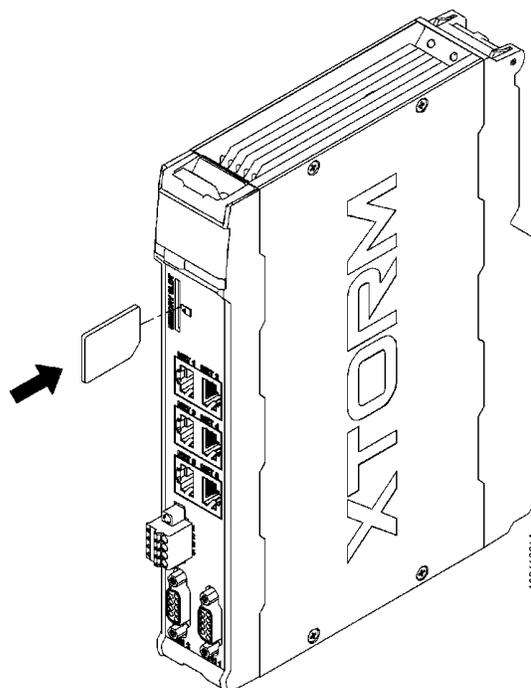


Figura 1-19. Inserção do Cartão de Memória na UCP

Quando o cartão estiver instalado corretamente aparecerá um símbolo no visor gráfico da UCP. Para remover o cartão de memória, basta acessar o Menu “Cartão de Memória” na UCP, utilizando o botão de diagnósticos, e selecionar a opção de remoção do cartão de memória, e aguardar até que o ícone do cartão desapareça da tela de status do visor gráfico. Para que isso seja possível, deve-se pressionar o cartão contra a UCP até que seja gerado um clique. Então, basta soltá-lo e retirá-lo do compartimento.



## Estudo Dirigido 1-1: anteprojeto de uma rede RS-422

Elabore o anteprojeto de uma rede RS-422 composta por uma UCP Hadron Xtorm atuando como dispositivo mestre conectada a dois outros dispositivos escravos via interface RS-422. Utilize a solução Altus para derivadores e terminadores (AL-2600) e cabos AL-2306. Os módulos AL-2600 que estão nas extremidades da rede fazem a função de terminadores. Neste caso deve-se configurar as chaves do AL-2600 em terminação PROFIBUS. Considere que todos os métodos de terminação são polarizados para comunicação RS-485 e RS-422 – não existindo terminações passivas.

🔑 **DICA:** consulte as características técnicas dos elementos utilizados na rede e o Instrutor para orientações adicionais. A resolução do estudo dirigido encontra-se no final deste Tutorial.

## Projeto do Painel Elétrico

Para efeitos de projeto do painel elétrico, a dimensão de cada módulo da Série Hadron Xtorm pode ser encontrada no documento de Características Técnicas de cada módulo. Além disso, o Manual de Utilização Hadron Xtorm traz orientações relativas aos seguintes aspectos relacionados ao projeto do painel elétrico:

- profundidade do módulo montado no bastidor;
- espaçamento entre módulos e outros equipamentos do painel;
- dimensionamento da calha;
- montagem horizontal e vertical;
- projeto térmico;
- considerações sobre interferências eletromagnéticas (EMI);
- alimentação do armário;
- distribuição dos cabos no armário;
- iluminação do armário;
- aterramento;
- blindagem;
- supressores de ruído;
- distribuição das alimentações fora do armário;
- proteção contra raios.

No que tange ao projeto do painel elétrico, em casos considerados críticos, consulte diretamente o serviço de suporte da Altus.



### Pesquisa na Documentação: projeto do painel elétrico

Consulte, na documentação da série, as diretrizes, orientações e informações relativas ao projeto do painel elétrico.

Fonte: Manual de Utilização Hadron Xtorm – MU223000!



### Pesquisa na Documentação: instalação do software MasterTool Xtorm

Consulte, na documentação da série, a sequência de passos para instalação do software MasterTool Xtorm.

Fonte: Manual de Utilização Hadron Xtorm – MU223000!

# Manutenção da Hadron Xtorm



## Diagnósticos do Módulo UCP

Uma das características da Série Hadron Xtorm é a geração de diagnósticos de comportamentos atípicos, sejam eles falhas, erros ou modos de operação, possibilitando ao operador identificar e solucionar problemas que venham a ocorrer com o sistema com grande facilidade.

A UCP Hadron Xtorm proporciona diversas maneiras de se visualizar os diagnósticos gerados pelo sistema, são elas:

- One Touch Diag;
- Diagnósticos via LED;
- Diagnósticos via WEB;
- Diagnósticos via Variáveis;
- Diagnósticos via Blocos Funcionais.

O primeiro deles é uma característica da Série Hadron Xtorm, a qual possibilita um rápido acesso às condições anormais da aplicação. O segundo é puramente visual, gerado através de um LED bicolor (DL) (azul / vermelho) presente no frontal do módulo. A próxima característica é a visualização gráfica em uma página WEB do bastidor e os respectivos módulos configurados, sendo permitido o acesso individual do estado de operação e os diagnósticos ativos. Os diagnósticos também são armazenados diretamente em variáveis simbólicas da UCP, e podem ser utilizados pela aplicação do usuário, por exemplo, sendo apresentados em um sistema de supervisão. Os últimos apresentam condições específicas de funcionamento do sistema.

A função destes diagnósticos é apontar possíveis problemas de instalação ou configuração do sistema, e de problemas ou deficiências das redes de comunicação. O capítulo de Manutenção deve ser consultado pelo usuário sempre que necessário.

### One Touch Diag

One Touch Diag (OTD), ou seja, diagnóstico com um único pressionamento, é uma característica que a Série Hadron Xtorm traz para a Unidade Terminal Remota. Com este conceito, o usuário pode verificar os diagnósticos de qualquer módulo presente no sistema diretamente no visor gráfico da UCP, com um único pressionamento no botão de diagnóstico do respectivo módulo. Essa é uma poderosa ferramenta de diagnóstico que pode ser usada off-line (sem sistema de supervisão ou software programador), tornando mais fácil encontrar e resolver rapidamente possíveis problemas.

O botão de diagnóstico (ver figura a seguir) está localizado na parte frontal da UCP, em local de fácil acesso, e, além de fornecer os diagnósticos ativos, permite o acesso ao menu de navegação.

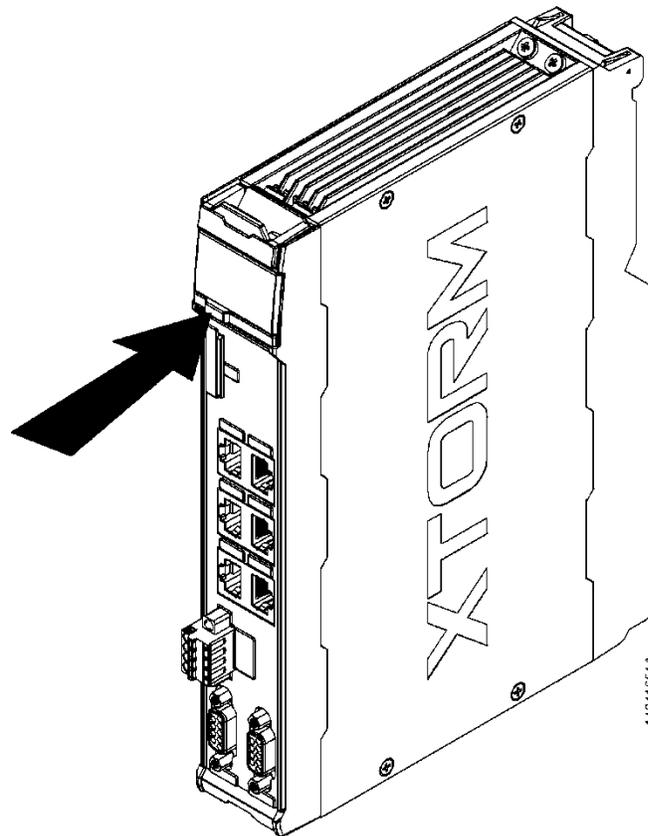


Figura 1-20. Botão de Diagnóstico

Com apenas um pressionamento curto, a UCP começa a mostrar os diagnósticos do barramento quando ativos, caso contrário exibe a mensagem "SEM DIAG". Inicialmente, será visualizada a Tag (configurada nas propriedades do módulo no software MasterTool Xtorm, seguindo a IEC 61131-3), ou seja, o nome atribuído à UCP. Em seguida, serão mostrados todos os diagnósticos, através de mensagens no visor da UCP. Esse processo será executado por duas vezes no visor. Tudo ocorre de forma automática, sendo que o usuário somente deverá executar o pressionamento curto inicial e a UCP será responsável por exibir os diagnósticos. Os diagnósticos de outros módulos presentes no barramento também serão exibidos no visor gráfico da UCP, através de um pressionamento curto no botão de diagnóstico deles, no mesmo modelo da apresentação dos diagnósticos da UCP.

A figura a seguir mostra todo o processo a partir do pressionamento curto, sendo a condição e os tempos da UCP representados nos retângulos menores. É importante salientar que os diagnósticos poderão ter mais de uma tela, ou seja, o tempo especificado no fluxograma abaixo é válido para cada uma delas.

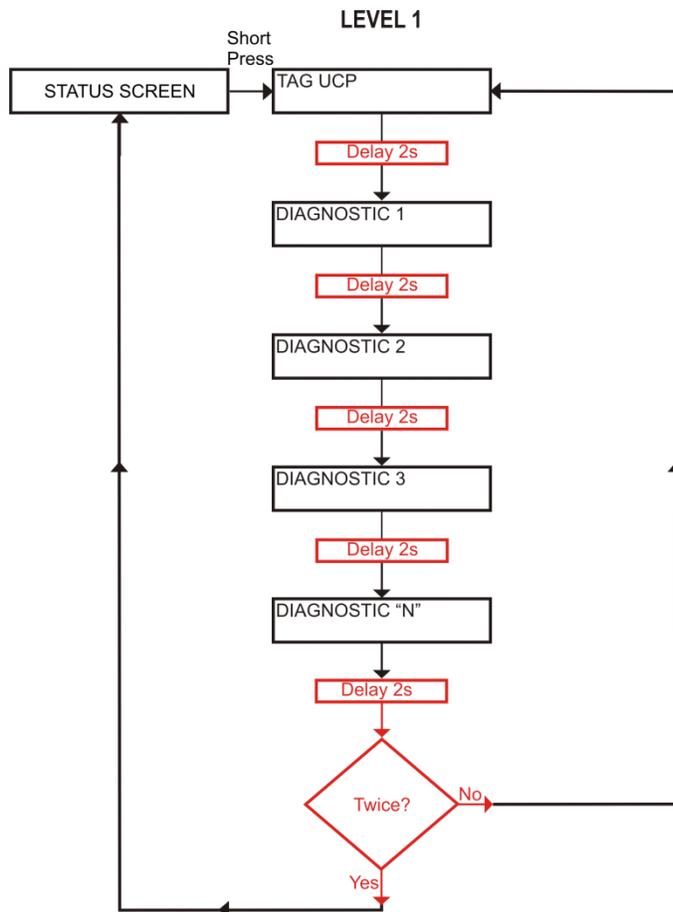


Figura 1-21. Visualização dos Diagnósticos da UCP

Para finalizar, antes de todo o processo de visualização ser efetuado, basta dar um pressionamento curto no botão de diagnóstico, em qualquer momento, ou pressionar o botão diagnóstico de algum módulo de E/S presente no barramento. Caso seja efetuado um pressionamento longo, a UCP entrará no menu de navegação, o qual está descrito na seção relativa ao Menu Informativo e de Configuração da UCP neste Tutorial. As mensagens exibidas no visor gráfico da UCP Hadron Xtorm, correspondentes aos diagnósticos, estão descritas na seção Diagnósticos via Variáveis neste Tutorial.

Caso ocorra uma situação de tecla presa de algum módulo de E/S presente no barramento, o botão de diagnóstico deste deixará de indicar diagnósticos no visor gráfico da UCP quando for pressionado, neste caso a UCP irá indicar que há módulos com diagnóstico ativo. Para que seja possível eliminar este diagnóstico da UCP, será necessário realizar uma troca a quente do módulo com o diagnóstico ativo. É importante ressaltar que essa opção somente estará disponível ao usuário quando o módulo estiver em modo operacional.

### Diagnósticos via LED

A UCP da Série Hadron Xtorm possui um LED (DL) para indicação de diagnósticos. A tabela a seguir mostra o significado de cada estado do LED.

Tabela 1-2. Descrição dos Estados do LED de Diagnóstico

DL (Cor)	Descrição	Causas	Prioridade
<b>Desligado</b>	Módulo desligado ou falha no visor	Sem fonte de alimentação Problema de hardware	-
<b>Ligado (Azul)</b>	Aplicação em execução (Modo Run)	-	4 (Mais Baixo)
<b>Piscando 2x (Azul)</b>	Módulos do barramento ou UCP com diagnóstico	No mínimo um módulo do barramento, incluindo a UCP, está com algum diagnóstico ativo	2
<b>Piscando 3x (Azul)</b>	Forçamento de dados	Alguma área de memória está sendo forçada pelo usuário via MasterTool Xtorm	3
<b>Ligado (Vermelho)</b>	Aplicação parada (Modo Stop)	-	4 (Mais Baixo)
<b>Piscando 1x (Vermelho)</b>	Cão-de-guarda de Software	Cão-de-guarda da aplicação de usuário	1
<b>Piscando 4x (Vermelho)</b>	Erro de Configuração ou de Hardware no barramento.	O barramento está danificado ou não está configurado corretamente	0 (Mais Alta)

⚠ **NOTA:**

Para remover a indicação de cão-de-guarda, deve-se efetuar um reset da aplicação ou desligar e ligar novamente a UCP. Esse cão-de-guarda ocorre quando o tempo de execução da aplicação de usuário for maior que o tempo de cão-de-guarda configurado.

### Diagnósticos via WEB

Além das características apresentadas anteriormente, a Série Hadron Xtorm traz para o usuário uma ferramenta inovadora de acesso aos diagnósticos e estados de operação do sistema, através de uma página WEB.

A utilização, além de dinâmica, é bastante intuitiva e facilita as operações do usuário. Em outras palavras, pode substituir o uso de um sistema de supervisão quando o uso for restrito a verificação de status do sistema.

Para acessar a página WEB da UCP desejada, basta utilizar um navegador padrão (Internet Explorer, Mozilla Firefox 3.0 ou superior e Google Chrome 8 ou superior) e digitar, na barra de endereço, o endereço IP correspondente a UCP (Ex.: <http://192.168.1.1>). Inicialmente, será apresentada as informações da UCP, conforme mostra a figura a seguir.

altus English | Español | Português

Hadron XTORM

CPU Overview System Overview Firmware Update

Updated on 30th September 2014, 11:26:20. To refresh this page click on the icon.

Model	HX3040
Tag	HX3040
Description	
Configured Racks	1
Rack	0
Slot	1
Firmware Version	1.0.1.28
Bootloader Version	1.0.0.1
Auxiliar Processor Version	1.0.1.2
Application State	Run
Active Diagnostics	1
Forced Values	No

Figura 1-22. Tela Inicial

Também existe a aba “System Overview”, a qual pode ser visualizada através do Bastidor ou da lista dos módulos presentes (opção do lado direito da tela). Quando não houver nenhuma aplicação na UCP, será exibida nesta página uma configuração com o maior Bastidor disponível e uma fonte de alimentação padrão, juntamente com a UCP conectada. Quando a visualização pelo bastidor é utilizada, os módulos que têm diagnóstico ficam piscando e assumem a cor vermelha, conforme mostra a figura a seguir. Caso contrário será exibida uma lista com os módulos presentes no sistema, Tags correspondentes e número de diagnósticos ativos.

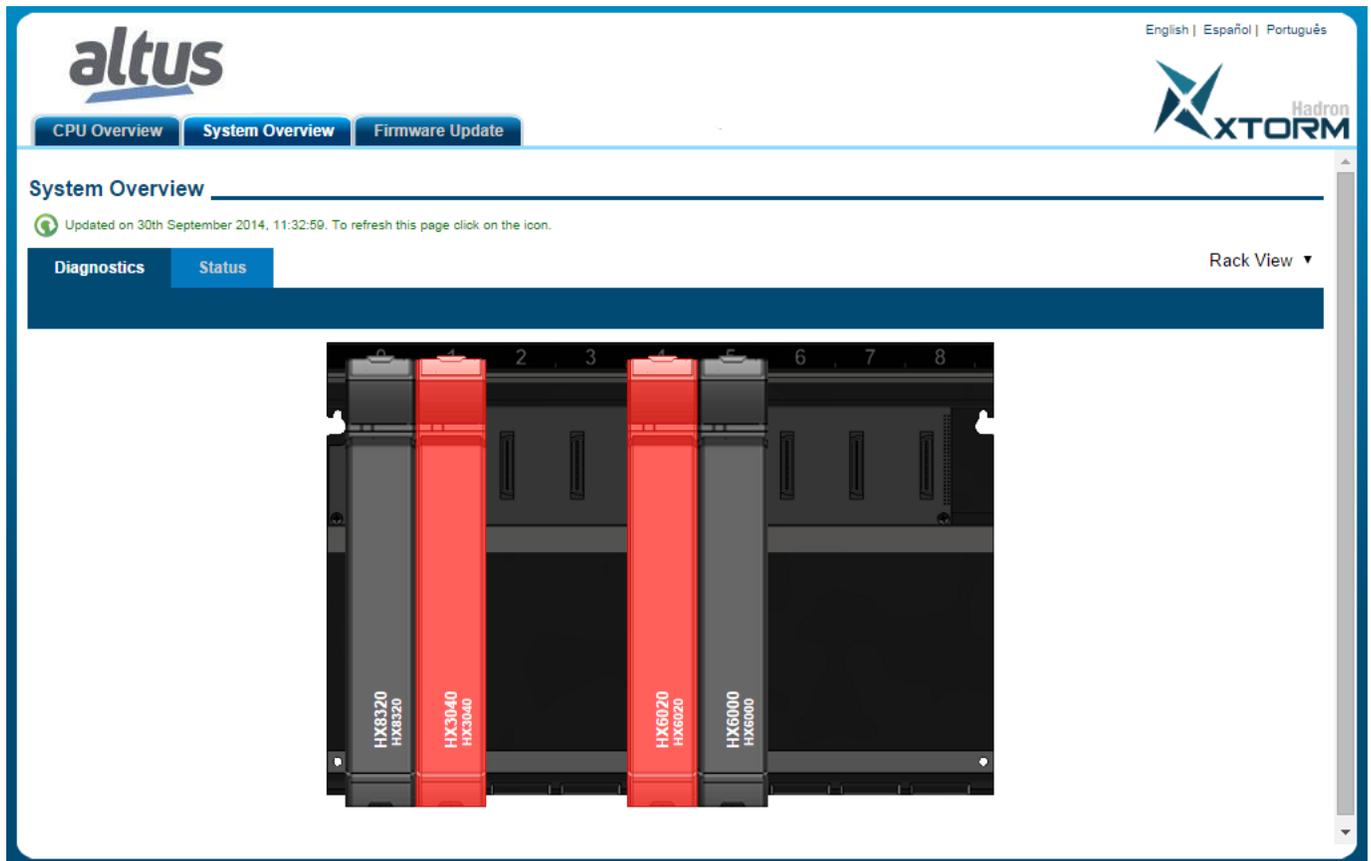


Figura 1-23. Informações do Sistema

Ao clicar em um dos módulos, no mesmo instante é mostrada uma tela que mostra a quantidade de diagnósticos do módulo e a quantidade de diagnóstico para cada canal do módulo, conforme mostra a figura a seguir.

Nota:

Quando uma UCP for reiniciada e a aplicação entrar em exceção na partida do sistema, os diagnósticos não estarão válidos. É necessário corrigir o problema que gera a exceção da aplicação para que os diagnósticos sejam atualizados.

🔔 **NOTA:**

Quando uma UCP for reiniciada e a aplicação entrar em exceção na partida do sistema, os diagnósticos não estarão válidos. É necessário corrigir o problema que gera a exceção da aplicação para que os diagnósticos sejam atualizados.

The screenshot shows the 'System Overview' page of the Altus web interface. At the top, there are navigation tabs for 'CPU Overview', 'System Overview', and 'Firmware Update'. The 'System Overview' tab is active. Below the tabs, there is a refresh icon and a message: 'Updated on 30th September 2014, 13:31:40. To refresh this page click on the icon.' Below this, there are two sub-tabs: 'Diagnostics' and 'Status'. The 'Diagnostics' sub-tab is active, displaying a table with the following data:

Channel	Tag	Description	Diagnostics
General	HX6020		8
00	Input00	Input00 Description	1
01	Input01	Input01 Description	1
02	Input02	Input02 Description	1
03	Input03	Input03 Description	1
04	Input04	Input04 Description	1
05	Input05	Input05 Description	1
06	Input06	Tag 06 Description	1
07	Input07	Input07 Description	1

Figura 1-24. Quantidade de Diagnósticos por módulo e por canal

Ao clicar em uma das linhas da tabela da figura anterior, no mesmo instante é mostrada uma tela que detalha a descrição de cada diagnóstico, conforme mostra a figura a seguir.

The screenshot shows the Altus Hadron Xtorm web interface. At the top, there are navigation tabs for 'CPU Overview', 'System Overview', and 'Firmware Update'. The 'System Overview' tab is active. Below the tabs, there is a language selection menu (English | Español | Português) and the Altus Hadron Xtorm logo. The main content area is titled 'System Overview' and includes a refresh icon and text: 'Updated on 30th September 2014, 13:33:08. To refresh this page click on the icon.' Below this, there are two sub-tabs: 'Diagnostics' and 'Status'. The 'Diagnostics' tab is selected, and the breadcrumb path is 'Diagnostic list >> Rack 0 >> HX6020 @ Slot 4 >> Channel >> 00'. The 'Firmware Version' is displayed as '01.00.00.16'. A table with two columns, 'ID' and 'Diagnostic', contains one row: '00' and 'Over range'.

ID	Diagnostic
00	Over range

Figura 1-25. Lista de Diagnósticos do Módulo/Canal

A lista completa de diagnósticos de cada módulo da Série Hadron Xtorm pode ser encontrada no documento de Características Técnicas de cada módulo.

Caso a aba Status seja selecionada, o estado de todos os diagnósticos detalhados é exibido na tela.

Além disso, o usuário pode optar por três opções de idioma: Português, Inglês e Espanhol. Basta alterar o menu superior direito para o idioma desejado.

A aba Atualização de Firmware é restrita ao usuário, ou seja, apenas para uso interno da Altus. Em casos em que a atualização é realizada remotamente (através de uma conexão de rádio ou satélite por exemplo, a velocidade mínima deste link deve ser de 128 Kbps).

A visualização dos diagnósticos Web é compatível somente até a versão 9 do navegador Internet Explorer, sendo que nesta versão é necessário habilitar o modo de exibição de compatibilidade do navegador. Caso contrário à exibição de algumas páginas poderá ser afetada.

### Diagnostic Explorer

O Diagnostic Explorer é a inclusão dos diagnósticos via WEB dentro do MasterTool Xtorm, a fim de que o acesso seja mais rápido e objetivo. O acesso a essa funcionalidade ocorre de duas maneiras:

1. Acessando a opção "Diagnostic Explorer" na árvore de dispositivos, localizada na esquerda da tela do MasterTool Xtorm, colocando o IP correto no campo indicado na figura a seguir. Lembrando que para a página de diagnósticos ser exibida, o usuário deverá estar conectado à UCP;

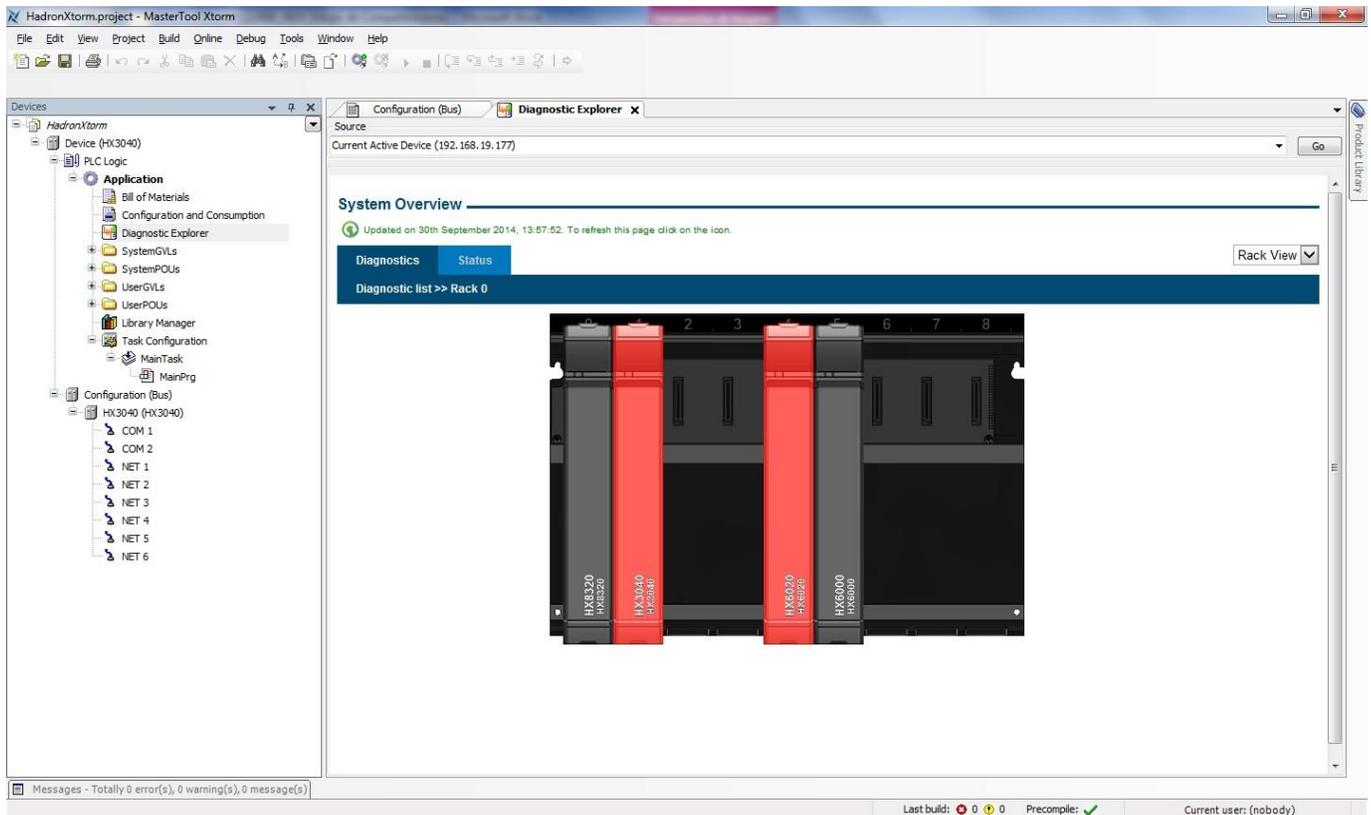


Figura 1-26. Tela do Diagnostic Explorer

2. Clicando com o botão direito do mouse sobre o módulo, e selecionando “Diagnósticos”, o Diagnostic Explorer será aberto, direcionando para a página de status do respectivo módulo.

### Diagnósticos via Variáveis

A UCP da Série Hadron Xtorm possui um conjunto de variáveis simbólicas globais onde são disponibilizadas uma série de informações de diagnóstico relacionadas ao hardware e software. Estas estruturas de dados são criadas automaticamente pelo MasterTool Xtorm. A funcionalidade de diagnóstico pode ser subdividida em dois tipos: diagnósticos resumidos e detalhados.

### Pesquisa na Documentação: Diagnósticos Resumidos

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

Fonte: Manual de Utilização Hadron Xtorm – MU223000!

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

- Sem Diag - Não existe diagnóstico ativo na UCP
- Configuração incompatível - O diagnóstico de configuração incompatível é gerado caso um ou mais módulos presentes (fisicamente), não confere(m) com o que foi declarado. O diagnóstico de módulos trocados ligará junto caso haja a troca de 2 módulos entre si (é uma informação complementar a configuração incompatível). Os módulos inseridos no barramento, além dos módulos declarados no projeto, não ligam este bit de diagnóstico, apenas em caso de módulos ausentes ou diferentes.
- Módulos Ausentes - Se um ou mais módulos declarados estão ausentes no barramento então o diagnóstico de Módulos Ausentes será verdadeiro.

- Módulos trocados - Se somente dois módulos estiverem trocados entre si no respectivo barramento, então o diagnóstico de troca pode ser identificado. Caso contrário, o problema será tratado como “Configuração Incompatível”.
- Módulos não declarados - Se um ou mais módulos presentes no barramento não estão declarados então o diagnóstico de Módulos não declarados será verdadeiro.
- Módulos com diagnósticos - Se um ou mais módulos presentes no barramento estão com diagnósticos ativos então o diagnóstico de Módulos com diagnósticos será verdadeiro.
- Módulos c/ erro fatal - Caso o diagnóstico de Módulos com Erro Fatal seja verdadeiro, verificar qual é o módulo que está com problema no barramento e encaminhar para a Assistência Técnica da Altus, pois ele está apresentando falha do hardware.
- Módulos c/ erro param - Caso o diagnóstico de Módulo com Erro de Parametrização seja verdadeiro, verificar se os módulos do barramento estão configurados corretamente e se as versões de firmware e do software MasterTool Xtorm estão adequadas.
- Erro no barramento - Considerado um erro fatal, interrompendo o acesso aos módulos do barramento. Caso o diagnóstico de erro no barramento seja verdadeiro, talvez uma situação anormal devido a configuração de troca a quente selecionada pode ter acontecido ou talvez haja um problema de hardware nas linhas de comunicação do barramento, sendo assim, deve-se entrar em contato com a Assistência Técnica da Altus.
- Bastidor Ausente - Se um ou mais bastidores declarados estão ausentes então o diagnóstico de Bastidor Ausente será verdadeiro.
- Bastidor Duplicado - Se há algum bastidor com o número de identificação duplicado então o diagnóstico de Bastidor Duplicado será verdadeiro.
- Bastidor inválido - Se há algum bastidor com o número de identificação inválido então o diagnóstico de Bastidor inválido será verdadeiro.
- Bastidor não declarado - Se há algum bastidor com o número de identificação não declarado então o diagnóstico de Bastidor não declarado será verdadeiro.
- Slot duplicado - Se há algum endereço de posição duplicado então o diagnóstico de Slot duplicado será verdadeiro.
- Falha de hardware - Caso o diagnóstico de Falha de Hardware seja verdadeiro, encaminhar a UCP para Assistência Técnica da Altus, pois ela apresenta problemas no RTC, processador auxiliar, ou outros recursos de hardware.
- Exceção no software - Caso o diagnóstico de exceção no software seja verdadeiro, o usuário deverá verificar a sua aplicação para garantir que ela não esteja acessando indevidamente a memória. Se o problema persistir, o setor de Suporte da Altus deverá ser consultado. Os códigos de exceção no software estão descritos após a tabela de diagnósticos detalhados da UCP.
- Perda de Retentividade - Este diagnóstico indica que houve perda dos dados não voláteis (variáveis retentivas, persistentes e fila de eventos). É ligado somente quando a perda for causada por problemas de hardware (falha no hardware, ou remoção à quente da fonte de alimentação ou da UCP). Comandos de reset cold e reset origin disparados através da ferramenta MasterTool Xtorm não causam o acionamento deste diagnóstico.

### **Pesquisa na Documentação: Diagnósticos Detalhados**

Consulte, na documentação do produto, a descrição dos diagnósticos detalhados e os códigos de exceção gerados a partir destes.

Fonte: Manual de Utilização Hadron Xtorm – MU223000!

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



- Visualização das Estruturas de Diagnóstico - As Estruturas de Diagnóstico adicionadas ao projeto podem ser visualizadas acessando o item "Library Manager" na treeview da janela do MasterTool Xtorm. Com isso, é possível visualizar todos os datatypes definidos na estrutura.
- Contadores - Todos os contadores dos diagnósticos da UCP retornam a zero quando o seu valor limite é ultrapassado.
- Reset por Brownout - O diagnóstico de reset por brownout somente será verdadeiro quando a alimentação da fonte exceder o limite mínimo exigido nas características técnicas dela, mantendo-se com tensão baixa, ou seja, sem sofrer uma interrupção. A UCP irá identificar a queda da alimentação e indicará o diagnóstico de falha na alimentação. Quando a tensão for restabelecida, a UCP será reinicializada automaticamente e indicará o diagnóstico de reset por brownout.

## Diagnósticos via Função

O diagnóstico via Função proporciona a visualização de alguns parâmetros que não podem ser acessados de outra maneira. Um exemplo desse tipo de função é a `GetTaskInfo`, a qual retorna informações sobre uma tarefa de uma determinada aplicação. A figura a seguir mostra essa Função.

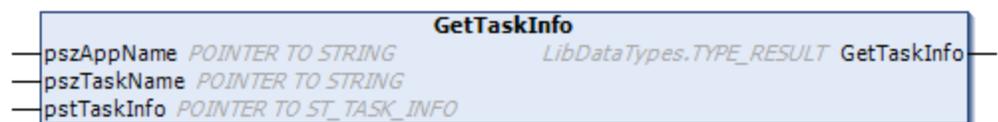


Figura 1-27. Função `GetTaskInfo`



### Pesquisa na Documentação: Diagnóstico via Função

Consulte, na documentação do produto, a descrição das variáveis de entrada e variáveis retornadas das funções que proporcionam a visualização de parâmetros específicos.

Fonte: Manual de Utilização Hadron Xtorm – MU223000!



### Estudo Dirigido 1-2: diagnóstico via Função

Implemente, na Linguagem ST (Texto Estruturado), a chamada da Função `GetTaskInfo` comentando as linhas de código.

📌 **DICA:** consulte o Instrutor para orientações adicionais. A resolução do estudo dirigido encontra-se no final deste Tutorial.

### Visor Gráfico

O visor gráfico disponível na UCP da Série Hadron Xtorm é uma importante ferramenta para o controle de processo, pois através dele podem ser reconhecidas as possíveis condições de erro, presença de componentes ou de diagnósticos ativos. Além disso, é através do visor gráfico que todos os diagnósticos, inclusive dos módulos de E/S, são exibidos ao usuário. Para maiores detalhes sobre a utilização do botão de diagnósticos e da visualização dos mesmos consulte a seção `One Touch Diag`. Na figura a seguir é possível visualizar todos os caracteres disponíveis no visor gráfico da UCP Hadron Xtorm e, a seguir, os seus respectivos significados.

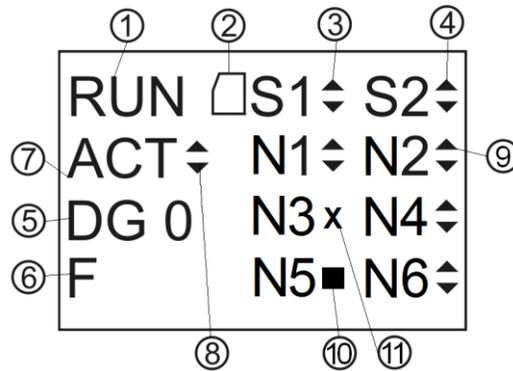


Figura 1-28. Tela de Status da UCP HX3040

Legenda:

Indicação do estado de operação da UCP. Caso a aplicação da UCP esteja em execução, o estado será Run. Caso a aplicação da UCP esteja parada, o estado será STOP e quando estiver parado em marcas de depuração da aplicação o estado será BRKP. Para maiores detalhes, consultar a seção Estados de Operação da UCP

Indicação da presença do Cartão de Memória. Para maiores detalhes sobre a instalação do cartão, consultar o capítulo Instalação.

Indicação de Tráfego na COM 1 (S1). A seta para cima (▲) indica transmissão de dados e a seta para baixo (▼) indica recepção de dados. Para maiores informações sobre a interface COM 1, consultar seção Interfaces Seriais

Indicação de Tráfego na COM 2 (S2). A seta para cima (▲) indica transmissão de dados e a seta para baixo (▼) indica recepção de dados. Para maiores informações sobre a interface COM 2, consultar seção Interfaces Seriais

Indicação da quantidade de diagnósticos ativos na UCP. Caso o número mostrado seja diferente de 0 (zero), existem diagnósticos ativos na UCP. Para maiores detalhes sobre a visualização dos mesmos no visor gráfico da UCP, através do botão de diagnósticos, consultar seção One Touch Diag

Indicação de variáveis forçadas na UCP. Caso o caractere F esteja exibido no visor gráfico, alguma variável está sendo forçada pelo usuário, seja ela variável simbólica ou variável de representação direta mapeada em uma variável simbólica. Para maiores detalhes sobre forçamento de variáveis, consultar seção Modo Run

Identificação do estado da redundância na UCP (mensagem válida somente na HX3040 em modo redundante). Caso a UCP seja a UCP Ativa, a informação ACT será apresentada. Os demais estados possíveis são NCF (Não Configurado), STR (Inicializando), INA (Inativo) e SBY (Reserva). Para UCP simples e redundante as letra BRKP indicam (BreakPoint)

Indicação de que está sendo executada a sincronização de projeto. A seta para cima (▲) indica transmissão de dados do projeto e a seta para baixo (▼) indica recepção de dados de um projeto. Para maiores informações sobre a sincronização de projetos, consultar seção Sincronização de Projetos

Indicação de tráfego na porta Ethernet (N) (NET 1 ao NET 6). A seta para cima (▲) indica transmissão de dados e a seta para baixo (▼) indica recepção de dados. Para maiores informações sobre as interfaces Ethernet, consultar o capítulo Configuração das Interfaces Ethernet

Indicação de que a interface faz parte de um par NIC Teaming e que ela está em estado de reserva (parada). Lembrando que os pares de NIC Teaming serão sempre N1/N2, N3/N4 e N5/N6. Para maiores informações sobre os modos de configuração das interfaces Ethernet, consultar o capítulo de Configuração das Interfaces Ethernet

Indicação de ausência de link de rede (x) na porta Ethernet (N) (NET 1 ao NET 6). Para maiores informações sobre as interfaces Ethernet, consultar o capítulo Configuração das Interfaces Ethernet.

Além dos caracteres descritos acima, a UCP Hadron Xtorm poderá apresentar algumas mensagens no visor gráfico, correspondentes a algum processo que está sendo executado no momento.

### Log de Sistema

O Log de Sistema é um recurso disponível no programador MasterTool Xtorm. É uma importante ferramenta para o controle de processo, pois através dele é possível localizar eventos na UCP que podem indicar condições de erro, presença de componentes ou de diagnósticos ativos. Tais eventos podem ser visualizados em ordem cronológica com uma resolução de milissegundos, com uma capacidade de armazenamento de até mil entradas de Log. Para acessar esses Logs, basta ir à Árvore de Dispositivos e dar um duplo-click em Device, e em seguida ir na aba Log.

Para que os Logs possam ser visualizados, basta estar conectado a uma UCP (Caminho Ativo selecionado) e clicar em . Quando este botão é pressionado, os Logs serão exibidos e a atualização deles é feita automaticamente. Quando o botão não está pressionado os Logs serão mantidos na tela, mas não serão atualizados. Ou seja, este botão tem dois estados, sendo que um estado mantém os logs sendo atualizados e no outro estado a atualização está desabilitada. Para deixar de exibir os Logs, basta clicar em .

É possível filtrar os Logs em 4 tipos diferentes: Advertência(s), Erro(s), Exceção(ões), Informação(ões).

Outra maneira de filtrar as mensagens exibidas para o usuário é selecionar qual o componente queremos visualizar.

A Marca de Tempo da aba de Log é apresentada pela ferramenta MasterTool a partir de informações fornecidas pelo dispositivo (UCP). A ferramenta MasterTool pode apresentar a Marca de Tempo no horário local (do computador) ou no horário UTC, se marcada a opção Tempo UTC.

#### NOTAS:

Se o horário ou o parâmetro de fuso horário do dispositivo estiverem incorretos, a Marca de Tempo apresentada pela ferramenta MasterTool não corresponderá ao tempo esperado.

Log de Sistema nas UCPs: Os logs de sistema da UCP da Série Hadron Xtorm não são recarregados no caso de uma reinicialização da UCP ou por uma reinicialização do Runtime, isto é, este log não é retentivo, não será possível visualizar os logs mais antigos quando uma dessas condições ocorrer.

### Diagnósticos dos Módulos de E/S

As funções de acesso da tag, Descrição e Diagnósticos dos módulos e pontos de E/S estão intimamente relacionadas entre si. Existem três componentes principais que serão amplamente referências nesta explicação, são eles:

Visor gráfico da UCP;

Visor do módulo a ser acessado;

Botão de diagnóstico do módulo a ser acessado.

### Acesso ao modo de diagnóstico

Independentemente do que está sendo mostrado no visor da UCP, após um pressionamento curto no botão de diagnóstico de um determinado módulo, será mostrado no visor da UCP o tag

e os diagnósticos ativos do respectivo módulo. Estes dados serão mostrados no visor da UCP na ordem indicada na figura a seguir.

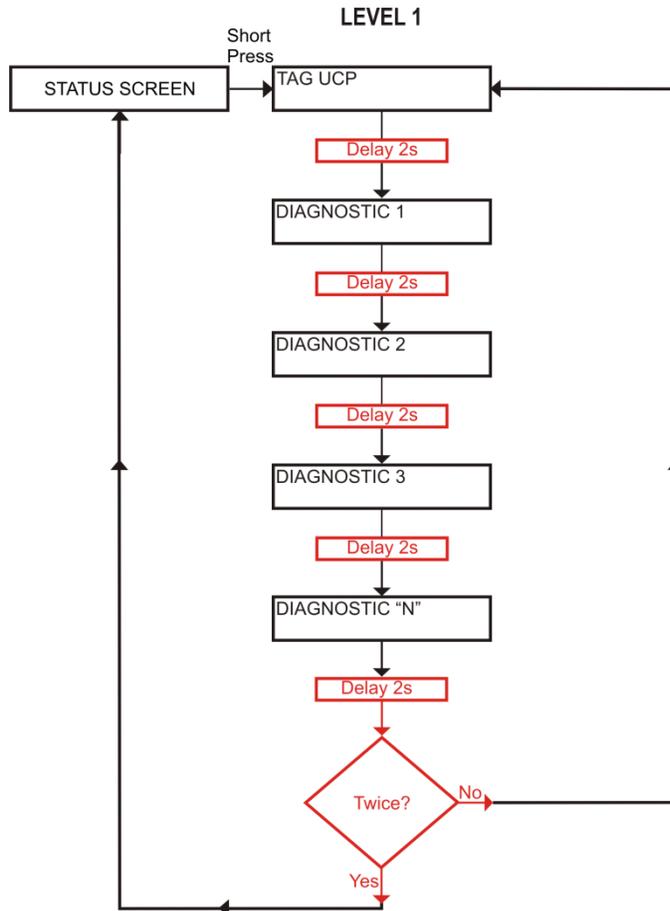


Figura 1-29. Visualização dos diagnósticos da UCP

Conforme mostrado na figura anterior, tanto o tag quanto a lista de todos os diagnósticos ativos relacionados ao módulo serão mostrados duas vezes no visor da UCP, após, o respectivo módulo sairá do modo diagnóstico e o visor da UCP voltará a indicar informações referentes à UCP.

É possível identificar que um determinado módulo está em modo diagnóstico quando os segmentos do visor do módulo, indicados na figura a seguir, estiverem piscando. A quantidade de segmentos depende da quantidade de pontos de cada um dos módulos de E/S.



Figura 1-30. Visor do Módulo

### Identificando pontos de E/S com diagnósticos individuais

Após entrar no modo diagnóstico, os módulos de E/S indicarão quais pontos que possuem diagnósticos individuais, piscando o segmento respectivo ao determinado ponto de E/S. Por exemplo, se o segmento indicado na figura a seguir estiver piscando, indicará que o segundo

ponto de E/S do módulo possui algum diagnóstico individual ativo. Tal funcionalidade se torna útil para a detecção rápida de diagnósticos individuais.

D	0	1	2	3	4	5	6	7
E	1	0	1	2	3	4	5	6
	2	0	1	2	3	4	5	6
	3	0	1	2	3	4	5	6

Figura 1-31. Exemplo de Diagnóstico

### Acessando pontos de E/S

Após entrar no modo diagnóstico, o próximo pressionamento curto do botão de diagnósticos do módulo em questão selecionará o primeiro ponto de E/S, neste momento o visor apagará a indicação de diagnósticos individuais ativos (explicado anteriormente) e indicará qual é o ponto de E/S que está selecionado. Para a seleção do próximo ponto de E/S basta um novo pressionamento curto no botão de diagnósticos do respectivo módulo. Quando o último ponto de E/S estiver selecionado um novo pressionamento curto no botão de diagnóstico irá finalizar o modo de diagnóstico.

Ao acessar um ponto de E/S, o visor da UCP indicará o tag do respectivo ponto de E/S e todos os diagnósticos ativos dele da mesma forma como indicada no fluxograma mostrado anteriormente.

Para módulo que não possuem pontos de E/S, um novo pressionamento curto no botão de diagnósticos, após a entrada no modo diagnóstico, finalizara a operação.

### Acessando a descrição do módulo e dos pontos de E/S

Além do tag, os módulos e pontos de E/S podem possuir uma descrição. Indica-se utilizar a descrição quando existe a necessidade de adicionar uma informação extra, além do tag para um determinado módulo ou ponto de E/S como, por exemplo: “Módulo de leitura de temperaturas” para a descrição de um módulo ou “Acionamento bomba principal” para a descrição de um determinado ponto de saída. O limite de caracteres para o tag, descrição da tag ou nome do módulo é de 255 caracteres.

Para alterar o nome e a descrição de cada módulo inserido na aplicação, clique com o botão direito sobre o módulo, no item “Propriedades”, na guia “Comum”, altere o nome ou a descrição sendo ambos limitados a 255 caracteres.

Para alterar a tag, acesse a aba “Bus I/O Mapping” de cada módulo, e dê um duplo clique na coluna da tabela destinada à inserção da tag de cada ponto. Para alterar a descrição da tag, faça o mesmo na coluna destinada para descrição da tag, e insira a informação desejada.

Para acessar a descrição, basta um pressionamento longo no botão de diagnóstico quando o módulo estiver indicando o tag ou os diagnósticos ativos do módulo. Nesse caso aparecerá no visor da UCP a descrição do módulo. Da mesma forma, quando estiver selecionado um ponto de E/S e for feito um pressionamento longo no botão de diagnósticos, aparecerá no visor da UCP a descrição do determinado ponto de E/S.

#### NOTAS:

É recomendável que a tag, e o nome do módulo possuam apenas os caracteres alfanuméricos (maiúsculos ou minúsculos e o texto não iniciar com números). Para a descrição o recomendável é os caracteres alfanuméricos (maiúsculos ou minúsculos), o espaço em branco e o caractere ponto “.”. A utilização de qualquer outro caractere diferente dos acima citados não é recomendada.

Quando utilizar a função ETD – Electronic Tag on Display, Diagnostic Explorer ou Web Server para visualizar a tag dos pontos de E/S, o nome da tag será truncado nos primeiros 24 caracteres após a string “Application.” do nome da tag. Ex. Para o tag "Application.UserPrg.MyTest.ON", somente a string “UserPrg.MyTest.ON” será visualizada no visor gráfico.

Quando utilizar a função ETD – Electronic Tag on Display, Diagnostic Explorer ou Web Server para visualizar a descrição do tag dos pontos de E/S, a descrição da tag será truncada nos primeiros 48 caracteres da descrição dada para o tag.

Quando utilizar a função ETD – Electronic Tag on Display, Diagnostic Explorer ou Web Server para visualizar o nome do módulo, o nome do módulo será truncado nos primeiros 24 caracteres da descrição dada para o nome do módulo.

### **Não Carregar a Aplicação na Inicialização**

Caso seja necessário, o usuário pode optar por não carregar uma aplicação já existente na UCP durante sua inicialização. Para isto, basta energizar a UCP com o botão de diagnóstico pressionado e mantê-lo assim por, pelo menos, dois segundos. Durante a inicialização da UCP será exibida no visor uma mensagem avisando que a aplicação não será carregada e inicializará em Modo Stop. Caso seja feito um Login, o software MasterTool Xtorm indicará que não existe nenhuma aplicação na UCP. Para voltar a carregar a aplicação, a UCP deve ser reinicializada ou um novo download da aplicação deve ser feito.

### **Falha na Alimentação**

A fonte de alimentação da Série Hadron Xtorm possui um sistema de detecção de falha, de acordo com os níveis definidos em suas características técnicas (consultar Características Técnicas do Módulo de Alimentação). Alguns pontos devem ser observados:

1 – Caso a fonte seja energizada com tensão inferior ao limite mínimo exigido, será gerado um diagnóstico de falha de alimentação, o qual será reconhecido pela UCP e a mensagem “FALHA NA ALIMENTAÇÃO” será exibida em seu visor. Quando a alimentação estiver dentro dos limites estabelecidos, a UCP irá reconhecer e, automaticamente, será reiniciada com a aplicação do usuário. O diagnóstico ainda estará ativo para mostrar ao usuário que na última inicialização ocorreu falha na alimentação.

2 – Caso a fonte tenha uma queda de tensão para um valor inferior ao limite mínimo exigido e retornar para um valor acima em até 10 ms, a falha de alimentação não será reconhecida pela UCP e não será gerado o diagnóstico, pois o sistema se mantém intacto durante esse tempo.

### **Problemas mais Comuns**

Se, ao energizar a UCP, ela não entra em funcionamento, os seguintes itens devem ser verificados:

A temperatura ambiente está dentro da faixa suportada pelos equipamentos?

A fonte de alimentação do bastidor está sendo alimentada com a tensão correta?

A fonte de alimentação é o módulo mais à esquerda no bastidor (bastidor sendo visto de frente), e a sua direita é seguido pela UCP da Série Hadron Xtorm? Caso a arquitetura utilizada seja com fontes redundantes, o módulo mais à esquerda no bastidor será a fonte, a sua direita é seguida por outra fonte, e na próxima posição a direita deverá ser UCP da Série Hadron Xtorm.

Os equipamentos da rede, como hubs, switches ou roteadores, estão alimentados, interligados, configurados e funcionando corretamente?

O cabo de rede Ethernet está devidamente conectado à porta NET 1 a NET 6 da UCP Hadron Xtorm e ao equipamento de rede?

A UCP da Série Hadron Xtorm está ligada, em modo de execução (Run) e sem diagnósticos relacionados à hardware?

Se a UCP Hadron Xtorm indica o estado execução (Run), mas não responde às comunicações solicitadas, sejam elas pelo MasterTool Xtorm ou através de protocolos, os seguintes itens devem ser verificados:

A configuração dos parâmetros Ethernet da UCP estão corretos?

O respectivo protocolo de comunicação está configurado corretamente na UCP?

As variáveis que habilitam as relações MODBUS / DNP3 estão devidamente habilitadas?

Se nenhum problema for identificado, consulte o Suporte a Clientes Altus.

### Solução de Problemas

A tabela a seguir mostra os sintomas de alguns problemas com suas possíveis causas e soluções. Se o problema persistir, entre em contato com o Suporte Técnico da Altus.

Tabela 1-3. Tabela de Solução de Problemas

Sintoma	Possível Causa	Solução
Não liga	Falta de alimentação ou alimentado incorretamente.	Verificar se a UCP está conectada corretamente no bastidor. Desenergizar e retirar todos os módulos do barramento, menos a fonte de alimentação e a UCP. Energizar o barramento e verificar o funcionamento da fonte de alimentação, tanto a externa como a fonte conectada ao barramento. Verificar se a tensão de alimentação chega ao borne da fonte de alimentação Hadron Xtorm, e se a polarização está correta.
Não comunica	Mal contato ou mal configurado.	Verificar todas as conexões dos cabos de comunicação. Verificar as configurações das interfaces serial e Ethernet no software MasterTool Xtorm.
Não reconhece o cartão de memória	Mal conectado ou desmontado	Verificar se o cartão de memória está conectado corretamente no compartimento, . Verificar se o cartão de memória foi colocado do lado correto, conforme a indicação no painel frontal da UCP. Verificar se o cartão de memória não foi desmontado através do menu específico na UCP, visualizando a indicação no visor gráfico da UCP.

### Manutenção Preventiva

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

- Deve ser verificado, a cada ano, se os cabos de interconexão estão conectados firmemente, sem acumulação de poeira, principalmente os dispositivos de proteção
- Em ambientes sujeitos a contaminação excessiva, o equipamento deve ser periodicamente limpo de poeira, detritos etc.
- Os diodos TVS usados para proteção contra transientes causados por descargas atmosféricas devem ser inspecionados periodicamente, pois podem ser danificados ou destruídos caso a energia absorvida esteja acima do limite. Em muitos casos, a falha pode não ser visual. Em aplicações críticas, é recomendável a substituição periódica dos diodos TVS, mesmo que não mostrem sinais visuais de falha
- Aperto e limpeza do bloco conector a cada seis meses.

## 2. Software da Hadron Xtorm



As UCPs da Série Hadron Xtorm usam as linguagens padrão IEC 61131-3, IL, ST, LD, SFC e FBD. IL e ST são linguagens textuais similares a Assembly e linguagem C respectivamente. LD, SFC e FBD são linguagens gráficas. O LD usa a representação de relés e blocos e é similar a diagramas de relé. O SFC usa a representação de um diagrama sequencial e permite uma visualização clara das funções executadas em cada ação. As UCPs desta Série oferecem também uma sexta linguagem chamada CFC.

### O Projeto no MasterTool Xtorm



A programação é feita na interface MasterTool Xtorm. O MasterTool Xtorm possibilita o uso de seis linguagens no mesmo projeto, fornecendo, assim, os melhores recursos que cada linguagem pode oferecer. Como resultado disto, o usuário tem eficientes desenvolvimentos de aplicações e facilidade na documentação e em futuras manutenções.

A configuração física bem como a ordem de disposição dos módulos no(s) barramento(s) deve ser a mesma configurada via MasterTool para que não sejam gerados diagnósticos e o sistema tenha condições de funcionamento correto.

#### Organização e Acesso à Memória

A Série Hadron Xtorm utiliza uma inovadora característica de organização e acesso à memória denominada big-endian, ou seja, o octeto mais significativo é armazenado primeiramente e sempre será o de menor endereço.

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



#### Pesquisa na Documentação: organização e acesso à memória

Consulte, na documentação do produto, as tabelas de organização e acesso à memória no padrão big-endian, que utiliza a palavra UCPHADRON XTORM, bem como a significância dos bytes (Significance) e a disposição dos demais tipos de variável, inclusive a sobreposição (Overlapping).

Fonte: Manual de Utilização Hadron Xtorm – MU223000!

#### Perfis de Projeto

Um perfil de projeto no MasterTool Xtorm é um conjunto de regras, características comuns e padrões utilizados no desenvolvimento de uma solução de automação, um perfil influencia a forma de implementação da aplicação. Seguir um perfil é uma forma de reduzir a complexidade na programação. As aplicações podem ser criadas conforme um dos seguintes perfis:

- Perfil para UTR;

- Perfil Personalizado.

O MasterTool Xtorm pode disponibilizar 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 Xtorm 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 Xtorm realiza a verificação de todas as regras definidas para o perfil válido para o projeto.

As próximas seções detalham as características ou padrões de cada perfil de projeto. 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 o outro mais sofisticado apenas quando as regras correspondentes estiverem sendo mais entesouradas 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.

### Perfil para UTR

O “Perfil para UTR” pode ser tanto criado para projetos redundantes, quanto para projetos não redundantes. Nesse tipo de perfil, a aplicação pode ter até duas tarefas de usuário, a “MainTask” que é sempre criada como padrão para projetos simples e redundantes, e a “ProtTask” que somente será criada se o usuário definir seu uso no projeto. A tarefa “MainTask” é responsável pela execução de uma única unidade de programação do tipo “Program” denominada “MainPrg”. Este programa não pode ser editado e é responsável por chamar outras quatro unidades de programação do tipo “Program”, denominadas “StartPrg”, “EngineeringPrg”, “AlarmPrg” e “UserPrg”.

Os programas “StartPrg” e “UserPrg” são os únicos da tarefa “MainTask” que podem conter código de usuário podendo chamar outras unidades de programação do tipo “Program”, “Function” ou “Function Block”, mas o código de usuário será executado pela tarefa “MainTask”. Já os programas “EngineeringPrg” e “AlarmPrg” são criados automaticamente e são responsáveis, respectivamente, por realizar as conversões de engenharia e sinalizar os alarmes mapeados pelo usuário nas telas de configuração da UCP.

A “ProtTask” também é responsável por executar uma única unidade de programação do tipo “Program”, cujo nome é “ProtPrg”. Este programa não pode ser editado e é responsável por chamar a “UserProtPrg” que, por sua vez, pode chamar outras unidades de programação e é responsável por executar as rotinas de proteção do projeto.

Neste perfil, a tarefa “MainTask” será do tipo cíclica “Cyclic” com prioridade fixada como 13 (treze) e executa exclusivamente o programa “MainPrg”, enquanto a tarefa “ProtTask” também será do tipo cíclica, mas terá sua prioridade fixada como 1 (um), sendo assim, mais prioritária que a anterior, e executa exclusivamente o programa “ProtPrg”.

As tarefas “MainTask” e “ProtTask” já estão completamente definidas e o desenvolvedor precisa criar o programa “StartPrg”, “UserPrg” e “UserProtPrg” 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 programa com o mesmo nome em substituição que seja construída em linguagem diversa.

A opção padrão do MasterTool Xtorm é utilizar o “MasterTool Standard Project” associado ao perfil “Perfil para UTR”, o qual inclui os programas “StartPrg”, “UserPrg” e “UserProtPrg” criados nas linguagens escolhidas durante a criação do projeto, e os programas “ProtPrg”, “MainPrg”, “AlarmPrg” e “EngineeringPrg” sempre na linguagem ST (Structured Text).

### Perfil Personalizado

O perfil de projeto “Personalizado” permite ao desenvolvedor explorar todas as potencialidades do Runtime System implantado na unidade central de processamento da Série Hadron Xtorm. Nenhuma das funcionalidades é desabilitada, nenhuma prioridade, associação entre tarefas e programas é imposta. Também é possível criar tarefas e programas com quaisquer nomes, porém, a “MainTask” e os programas “StartPrg”, “MainPrg”, “AlarmPrg” e “EngineeringPrg”, devem sempre existir com estes nomes neste perfil, além de que, os programas “MainPrg”, “AlarmPrg” e “EngineeringPrg” não podem ser editados.

Além das tarefas em tempo real com prioridade 00 a 15, as quais são escalonadas por prioridade, neste perfil também é possível definir tarefas com prioridades menores na faixa de 16 a 31.

No perfil “Personalizado”, o usuário pode gerar algumas tarefas que não estão presentes no perfil “Perfil para UTR”. Duas destas outras tarefas são denominadas “CyclicTask00”, “CyclicTask01” cada qual responsável pela execução do respectivo programa “CyclicPrg<nn>”. As tarefas “CyclicTask<nn>” são do tipo cíclica “Cyclic”. Estes dois tipos formam um conjunto denominado de tarefas básicas.

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

Quanto ao tempo de cão-de-guarda o ideal é manter o tempo médio de execução das tarefas em, no máximo, 50% do tempo de cão-de-guarda. Assim, diminuem as chances de ocorrerem erros de cão-de-guarda por eventuais picos de tempo na execução da tarefa. Este parâmetro somente é editável no perfil personalizado.

As tarefas chamadas “ExternInterruptTask<nn>” são tarefas de interrupção do tipo Externa “Extern” cuja execução é disparada por algum evento externo, como a variação de um sinal de controle numa porta serial. Essas tarefas são responsáveis pela execução dos programas “ExternInterruptPrg<nn>”. As tarefas chamadas “TimeinterruptTask<nn>” são tarefas de interrupção do tipo cíclica “Cyclic”, sendo responsáveis pela execução dos programas “TimeInterruptPrg<nn>”.

Neste perfil de projeto, a aplicação ainda pode incluir a tarefa de usuário “FreeTask” do tipo “Freewheeling”, responsável pela execução do programa “FreePrg”. Esta é uma tarefa de baixa prioridade e pode ser interrompida por todas as demais, ela é capaz de executar códigos que podem ficar bloqueados.

No modelo “Perfil Personalizado”, existem oito tarefas e suas POUs já completamente definidas conforme apresentado, bem como os respectivos programas associados criados na linguagem que o usuário selecionar. Os intervalos, eventos de disparo e prioridades de qualquer tarefa podem ser configurados pelo usuário.

#### ¶ NOTA:

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

### Novo Projeto

Como descrito anteriormente, existem diferentes tipos de perfis de projeto, sendo dependentes do conhecimento técnico de cada usuário. Neste capítulo, será abordada somente a criação de um novo projeto a partir do perfil Simples, utilizando a ferramenta “Wizard”, a qual apresenta as opções de configuração do sistema ao usuário.

Inicialmente, o usuário deverá criar um projeto novo no MasterTool Xtorm a partir do menu Arquivo e logo em seguida, “New Project...”, conforme mostra a figura a seguir.

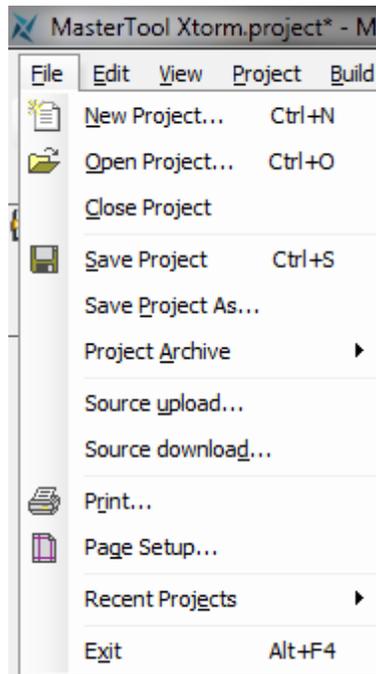


Figura 2-1. Novo Projeto

Após, uma janela será apresentada ao usuário, solicitando que ele selecione o tipo de projeto que deseja fazer e, em seguida, escreva um nome e a localização para armazenar o projeto no computador. Clicar em OK para prosseguir ou Cancel para interromper.

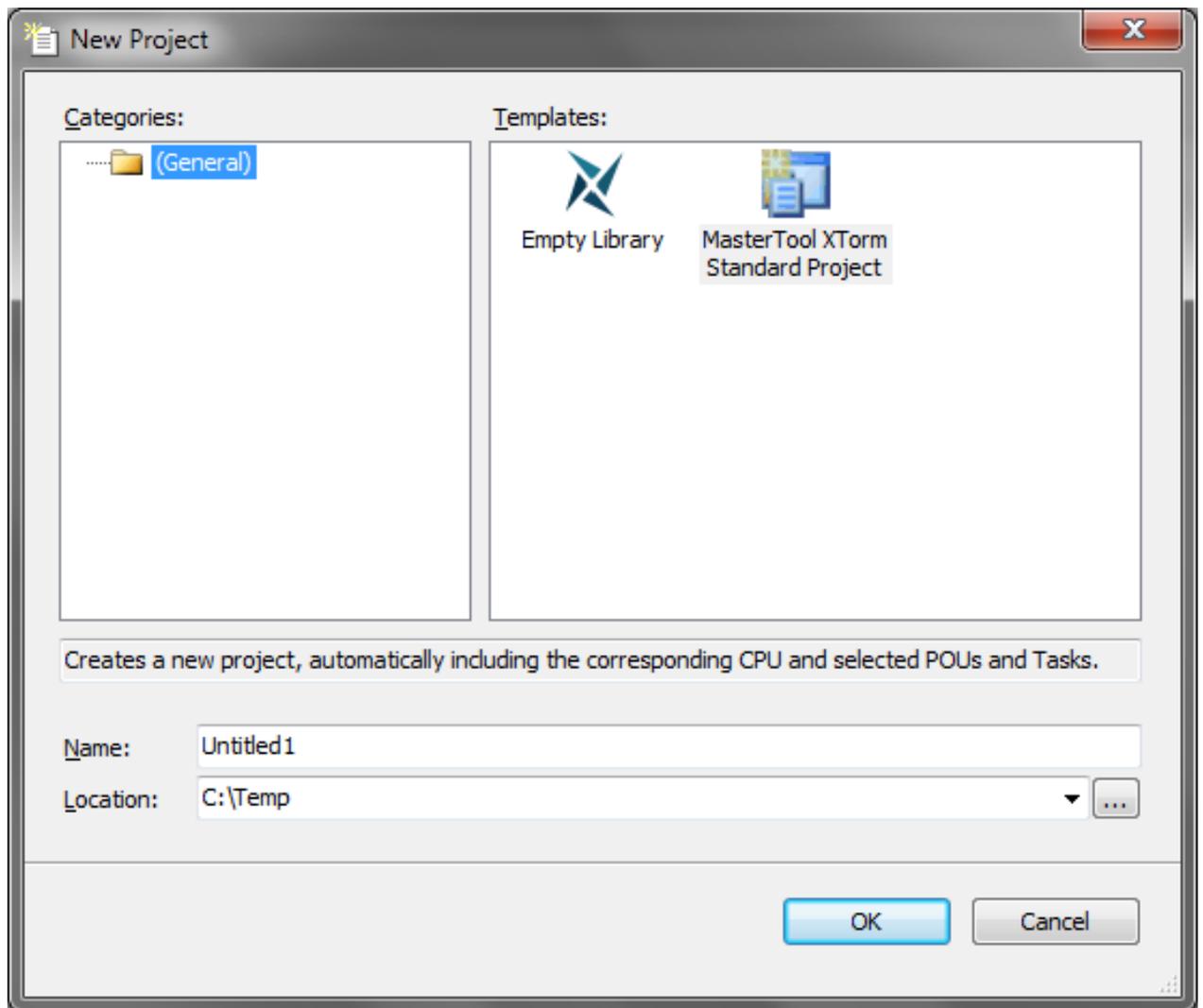


Figura 2-2. Classificação do Projeto

A seguir, o usuário deverá selecionar a UCP desejada, os módulos de hardware básicos que compõem o barramento, ou seja, o modelo de bastidor e de fonte de alimentação e a configuração de redundância. Nesse caso, será utilizada a UCP HX3040 (sem redundância), um bastidor HX9001 e uma fonte de alimentação HX8320 (sem redundância).

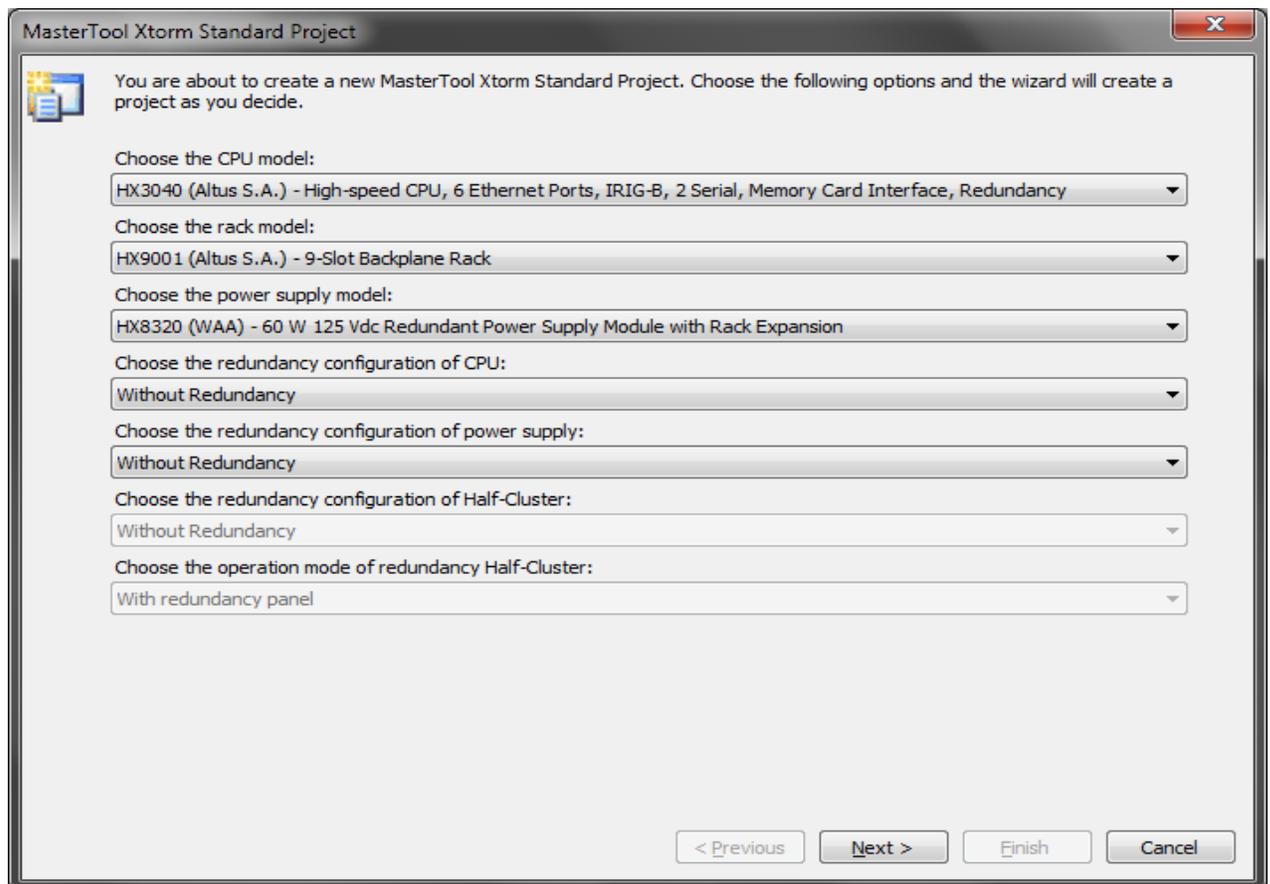


Figura 2-3. Opções do Projeto

Caso o usuário selecione a opção para redundância de UCP, a opção para redundância de fonte de alimentação será automaticamente marcada. Esta é uma característica de projeto, sendo que para projetos com redundância de UCP, será obrigatoriamente necessária a redundância de fonte de alimentação. Na figura a seguir podemos visualizar uma tela com a opção de redundância de UCP selecionada.

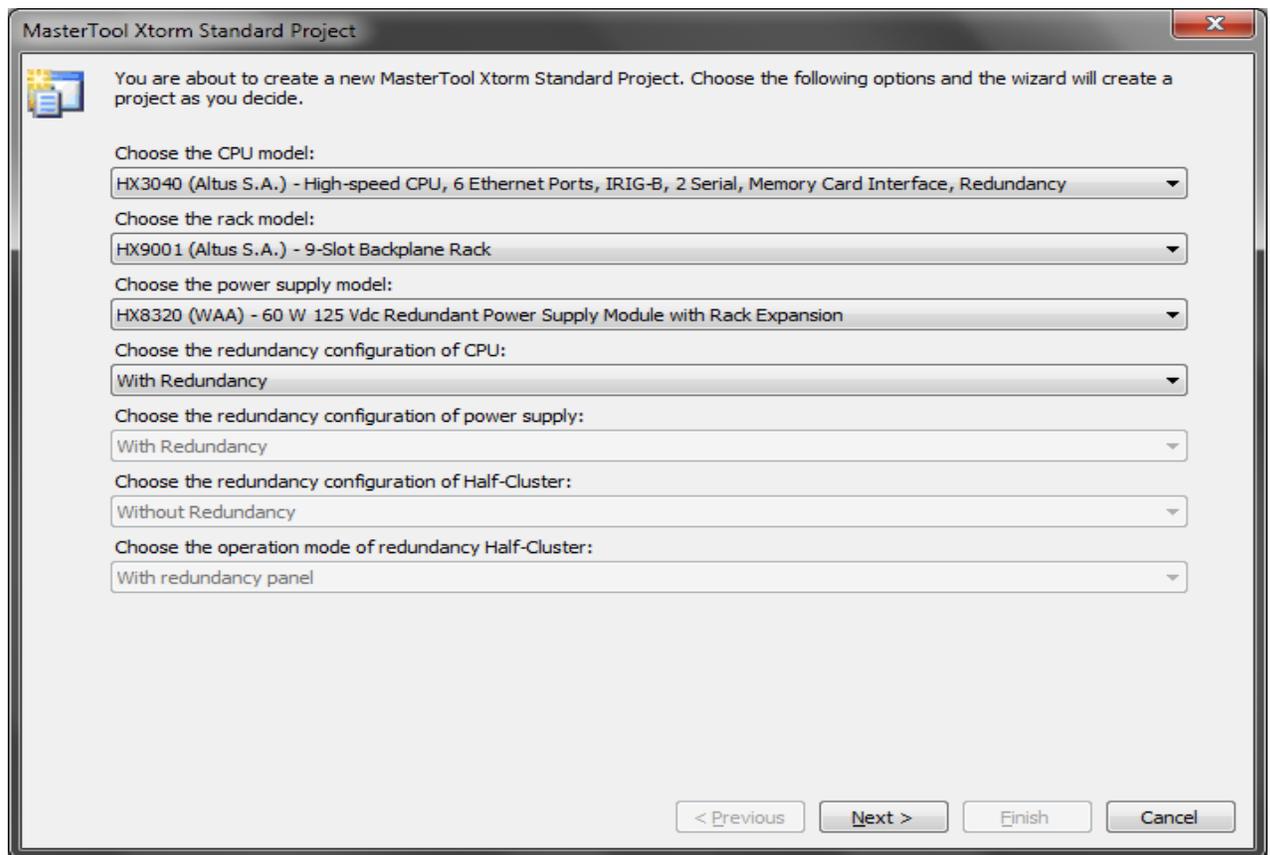


Figura 2-4. Opções de projeto com redundância de UCP habilitada

Na próxima tela, o usuário deverá configurar a quantidade de módulos que usará no projeto, sendo assim o Wizard criará os objetos destes módulos dentro do projeto automaticamente.

MasterTool Xtorm Standard Project

Choose the quantity of I/O points that will be used and the wizard will create the following objects within the project.

Quantity of 125 Vdc digital input points: 16

Quantity of 125 Vdc digital output points: 16

Quantity of 24 Vdc digital input points: 0

Quantity of 24 Vdc digital output points: 0

Quantity of V/I analog input points: 8

Quantity of RTD analog input points: 8

Expansion cable type:  
Not Connected

Quantity of I/O modules that will be used: 4 of 80.  
Quantity of racks that will be used: 1 of 9.

< Previous    Next >    Finish    Cancel

Figura 2-5. Opções de módulos de E/S do Projeto

Então, o usuário deverá selecionar o perfil para o projeto, e a linguagem padrão para as POU's (programas). Nesse caso o novo projeto está exemplificado perfil RTU sem redundância e linguagem ST. Clicar em Next para prosseguir ou Cancel para interromper a criação do projeto.

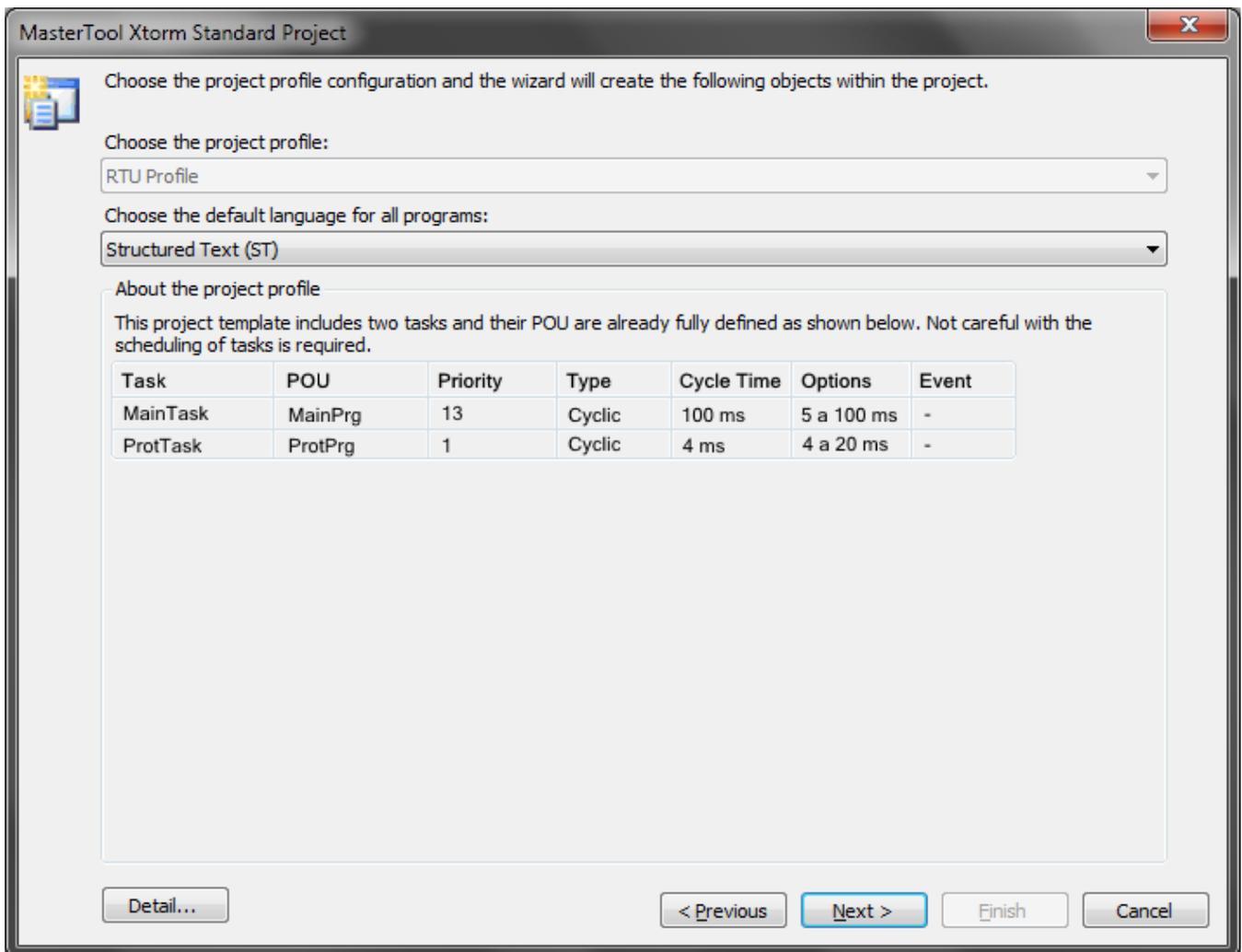


Figura 2-6. Seleção das Características de Perfil do Projeto

A próxima tela define a linguagem da POU criada pelo perfil selecionado. Como o projeto criado não é redundante, existem apenas duas POUs (UserPrg) e (StartPrg) a linguagem ST foi mantida. Clicar em Anterior para voltar à tela antecedente, Finish para finalizar ou Cancel para interromper.

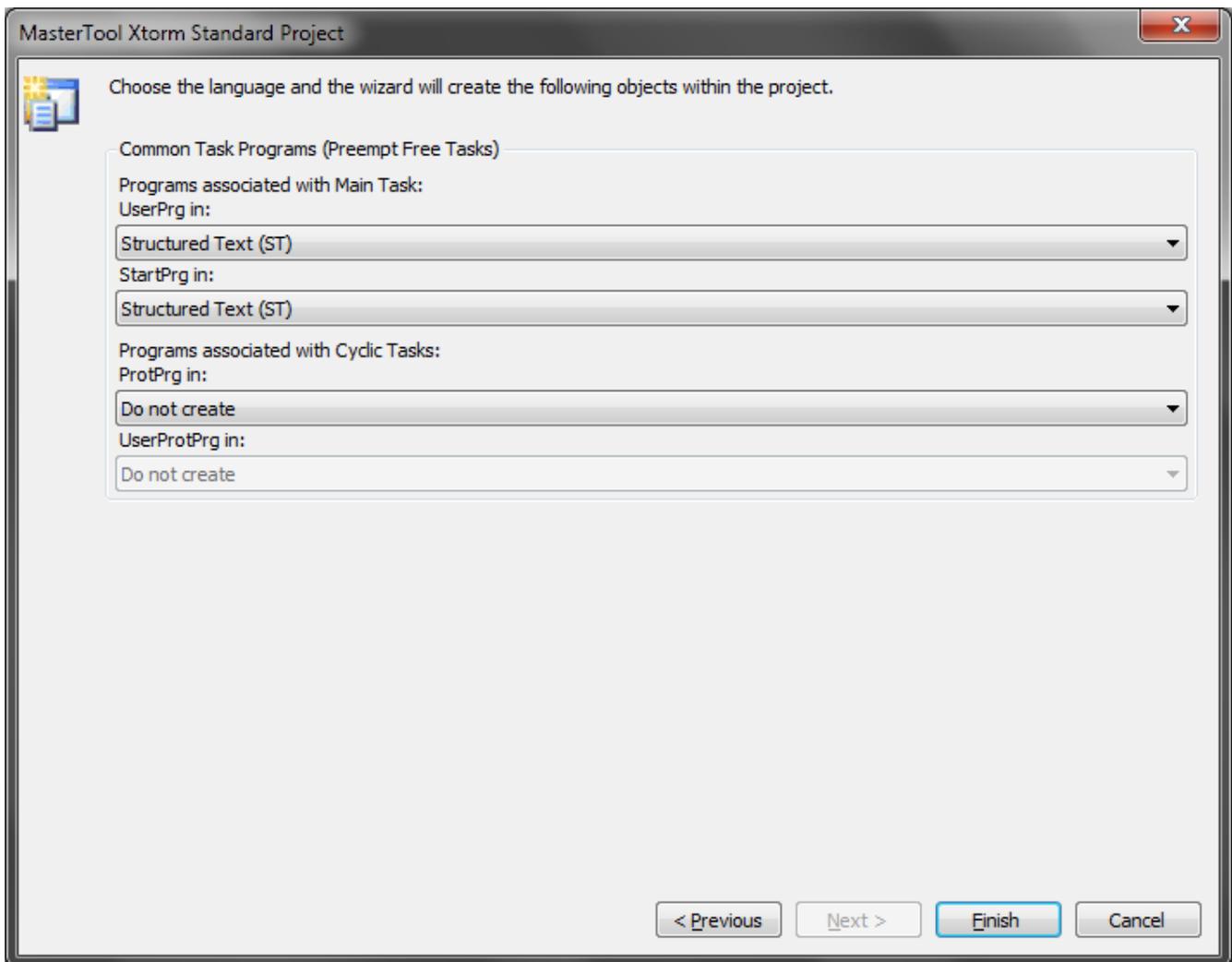


Figura 2-7. Linguagem de Programação

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

### Acrescentando Módulos

Por padrão, a UCP e os módulos de hardware selecionados na criação do projeto já são inseridos na configuração do sistema. Caso necessário, o usuário ainda pode incluir outros módulos necessários. A UCP Xtorm pode ter até 80 módulos de entradas e saídas distribuídos num total de 16 bastidores. Para mais informações consulte a CT123200 das fontes da série Xtorm.

Caso a aba Biblioteca de Produtos não esteja disponível na tela do MasterTool Xtorm, a mesma deve ser incluída, através do menu Visualizar, clicando sobre o item “Product Library”, conforme mostra a figura a seguir.

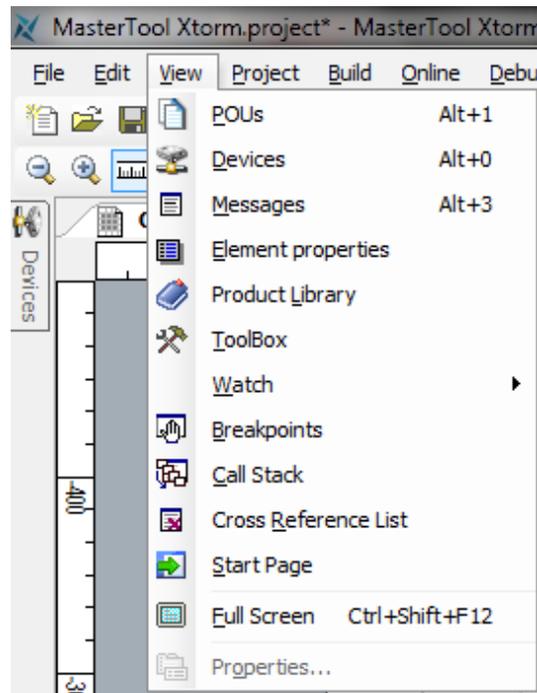


Figura 2-8. Visualizando a Biblioteca

Após, basta selecionar o módulo a ser inserido no projeto e, mantendo pressionada a tecla esquerda do mouse, arrastá-lo para a área de configuração do barramento.

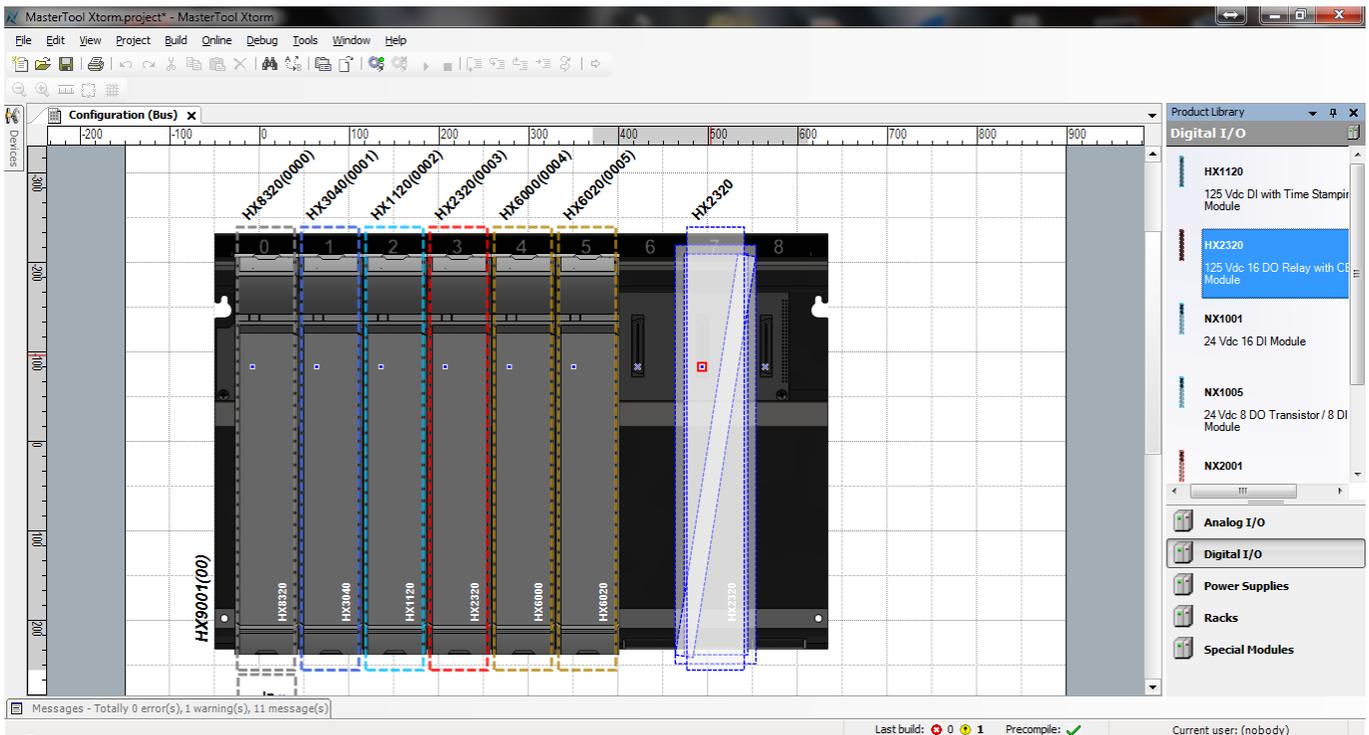


Figura 2-9. Acrescentando Módulos

# Os Elementos de Software IEC61131-3



## Unidades de Organização de Programa

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

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

Para inserir uma nova POU, basta clicar com o botão direito sobre o nome da aplicação, (nome padrão criado para a aplicação é "Application") , selecionar "Add Object" e, então, "POU...", conforme mostra a figura a seguir.

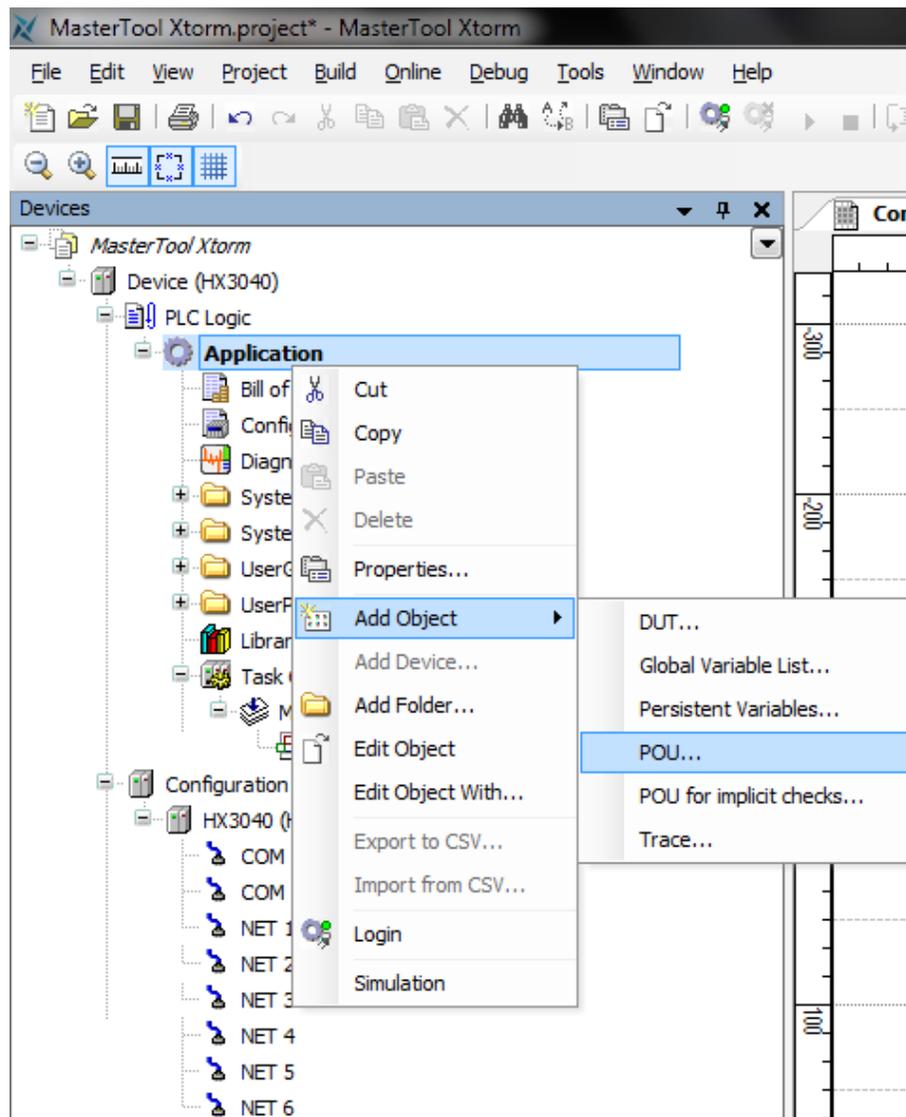


Figura 2-10. Inserindo POU's

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

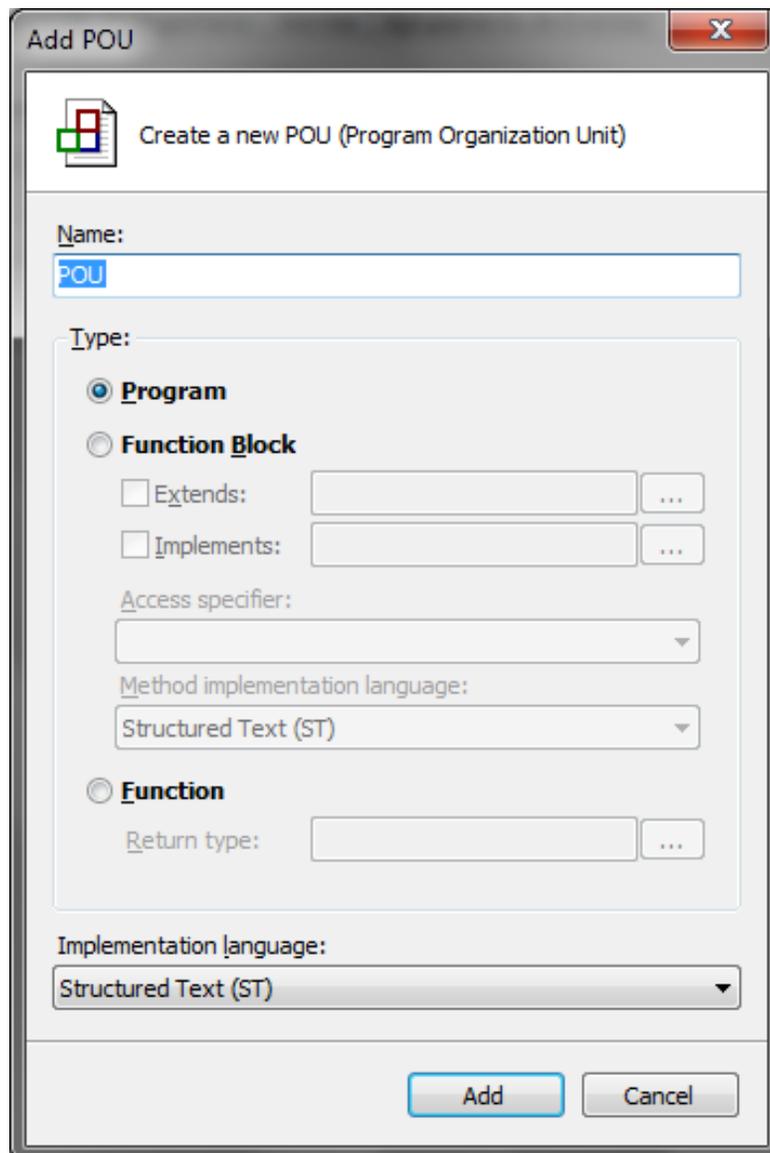


Figura 2-11. Classificando a POU

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

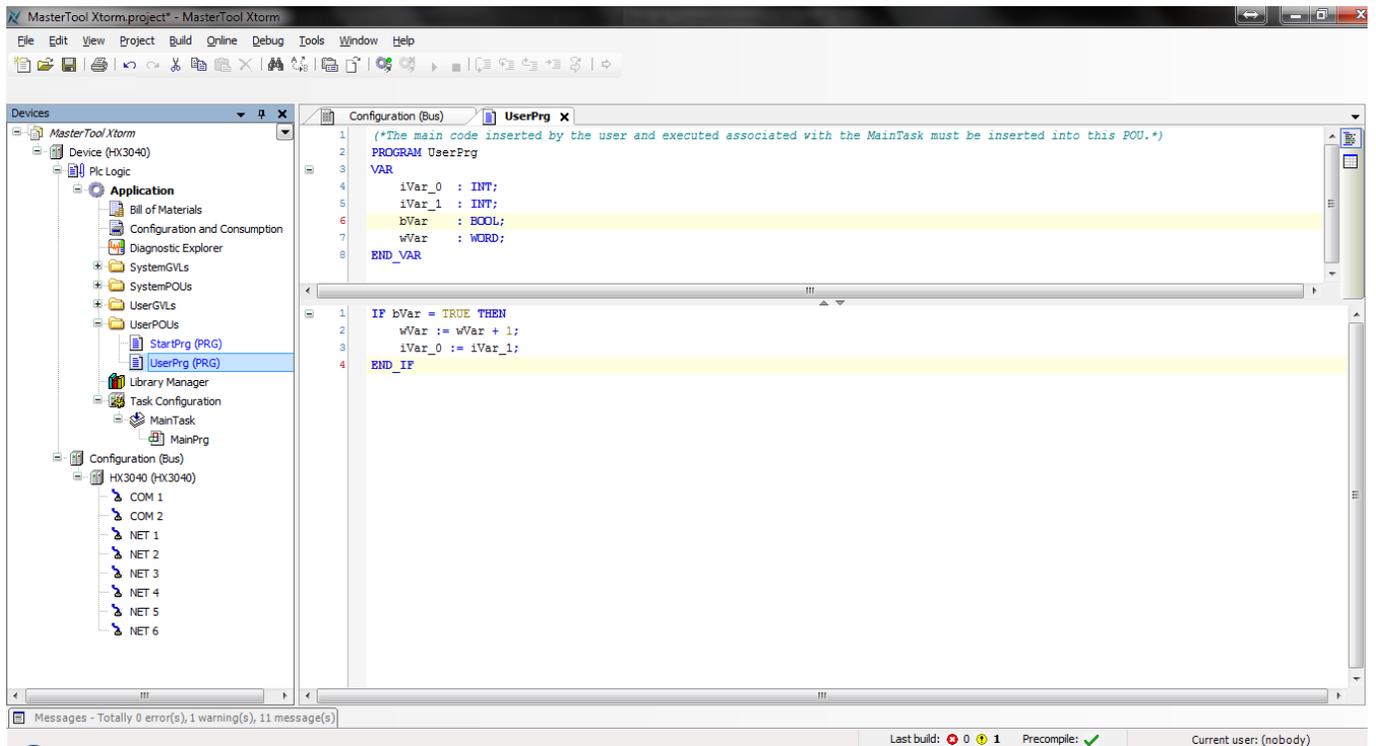


Figura 2-12. Editando a POU

## Tarefas

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

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

Desta forma, para incluir uma nova tarefa (caso o perfil selecionado permita), basta clicar com o botão direito sobre Configuração de tarefas, selecionar “Add Object” e então “Task...”, conforme mostra a figura a seguir.

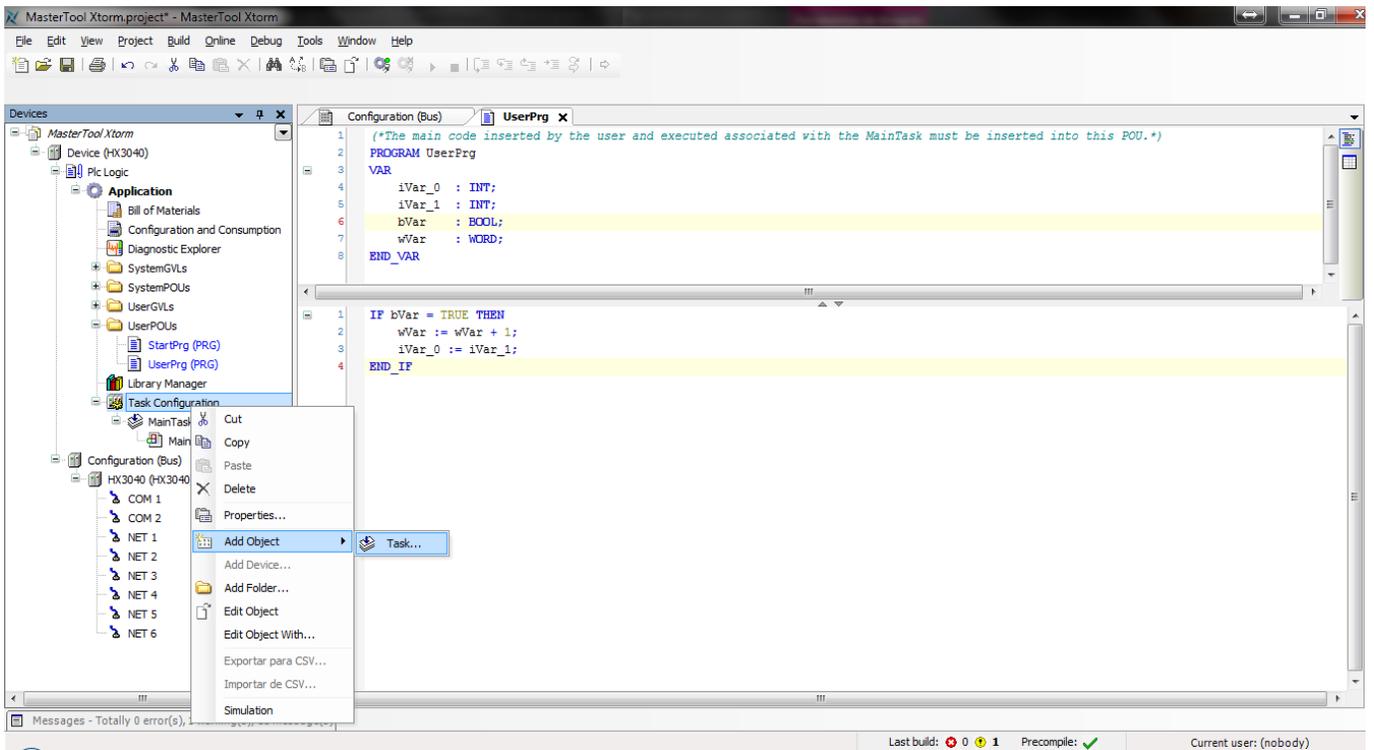


Figura 2-13. Criando uma Tarefa

Em seguida, surgirá uma tela para colocar o nome da tarefa. Após, clicar em Add para finalizar a criação desta.

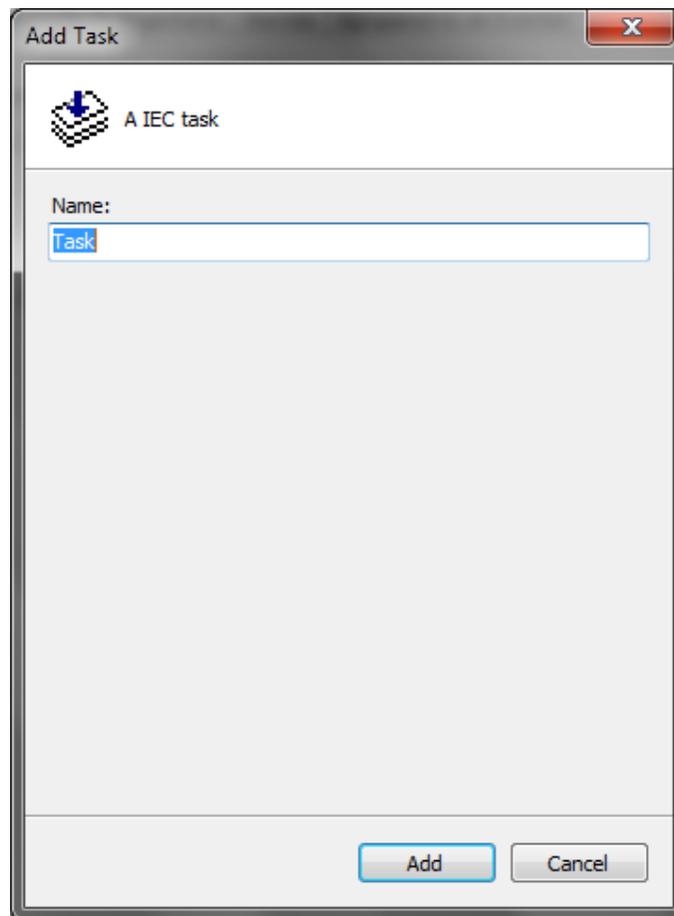


Figura 2-14. Nomeando a Tarefa

### Configurando as Tarefas

Após abrir a tarefa, a janela de configuração surgirá para o usuário definir e classificar o funcionamento dela.

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

O espaço "Type" define qual o tipo e o método para a tarefa ser executada, sendo que podem ser selecionados os seguintes itens:

- **Cyclic** - A tarefa é executada ciclicamente, ou seja, é chamada a cada intervalo de tempo configurado no campo ao lado. Ex: t#100 ms.
- **Event** - A tarefa é executada quando a variável do tipo BOOL, configurada no campo ao lado, recebe uma borda de subida, ou seja, a variável passe de FALSE para TRUE.
- **External** - A tarefa é executada quando uma interrupção externa ocorre, a qual é configurada no campo ao lado.
- **Freewheeling** - A tarefa sempre é executada, de acordo com a sua prioridade, ou seja, tarefas com maior prioridade são executadas primeiramente.
- **Status** - A tarefa é executada quando a variável do tipo BOOL, configurada no campo ao lado, for verdadeira.

Além dos campos mencionados acima, ainda deve ser configurado o Interval (apenas para tarefas cíclicas), ou seja, o intervalo de tempo em que a tarefa é chamada para executar (tempo máximo

para a MainTask é igual a 750 ms e o tempo mínimo igual a 5 ms). Recomenda-se configurar o intervalo da tarefa de um modo que o tempo de ciclo (execução) seja, no máximo, 80% do intervalo.

O cão-de-guarda da UCP é configurado para evitar o travamento das tarefas de usuário.

O campo "Time", define o tempo máximo permitido para a execução da tarefa. Caso a tarefa leve um tempo maior que o tempo de cão-de-guarda para ser executada, a aplicação irá para STOP e entrará em exceção por cão-de-guarda.

O campo "Sensitivity" refere-se a quantas vezes o tempo do cão-de-guarda deverá ser atingido para que o diagnóstico bWatchdogReset seja ativado, por sua vez somente editável em um projeto com perfil personalizado. Caso o tempo de execução da tarefa atinja o valor do campo Sensivity multiplicado pelo campo Time, o diagnóstico também será indicado.

Deve-se atentar ao fato de que o cão-de-guarda da UCP não é utilizado para proteger a aplicação do usuário de picos no tempo de execução e sim de travamentos.

Visando proteger o sistema quanto a possíveis erros de configuração, o MasterTool Xtorm verifica em todas as tarefas cíclicas, durante a compilação, o cão-de-guarda (Cão-de-Guarda de Software) e os limites mínimo e máximo do intervalo da tarefa. É importante destacar que o usuário deverá ter cuidado ao alterar os valores pré-definidos pelos perfis de projeto, uma vez que alterações indevidas podem colocar em risco a execução do sistema. Então, recomenda-se utilizar os valores padrão, ou seja, manter o tempo de cão-de-guarda em 1000 ms.

The screenshot shows the 'Configuration (Bus)' window with a 'Task' tab selected. The configuration fields are as follows:

- Priority ( 0..31 ): 1
- Type: Cyclic
- Interval (e.g. t#200ms): t#20ms
- Watchdog:
  - Enable
  - Time (e.g. t#200ms): 1000 ms
  - Sensitivity: 1

Below the configuration fields is a toolbar with the following actions: Add Call, Remove Call, Change Call, Move Up, Move Down, and Open POU. At the bottom, there is a table with two columns: POU and Comment.

POU	Comment

Figura 2-15. Configurando a Tarefa Criada

### Vínculo de POUs com Tarefas

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

Para associar uma POU criada, basta acessar a tarefa desejada, clicando duas vezes sobre ela na árvore de dispositivos, e clicar sobre "Add POU". Após, surgirá uma tela denominada "Input Assistant", ou seja, Assistente de Entrada na qual deverá ser selecionada a POU desejada, conforme mostra a figura a seguir.

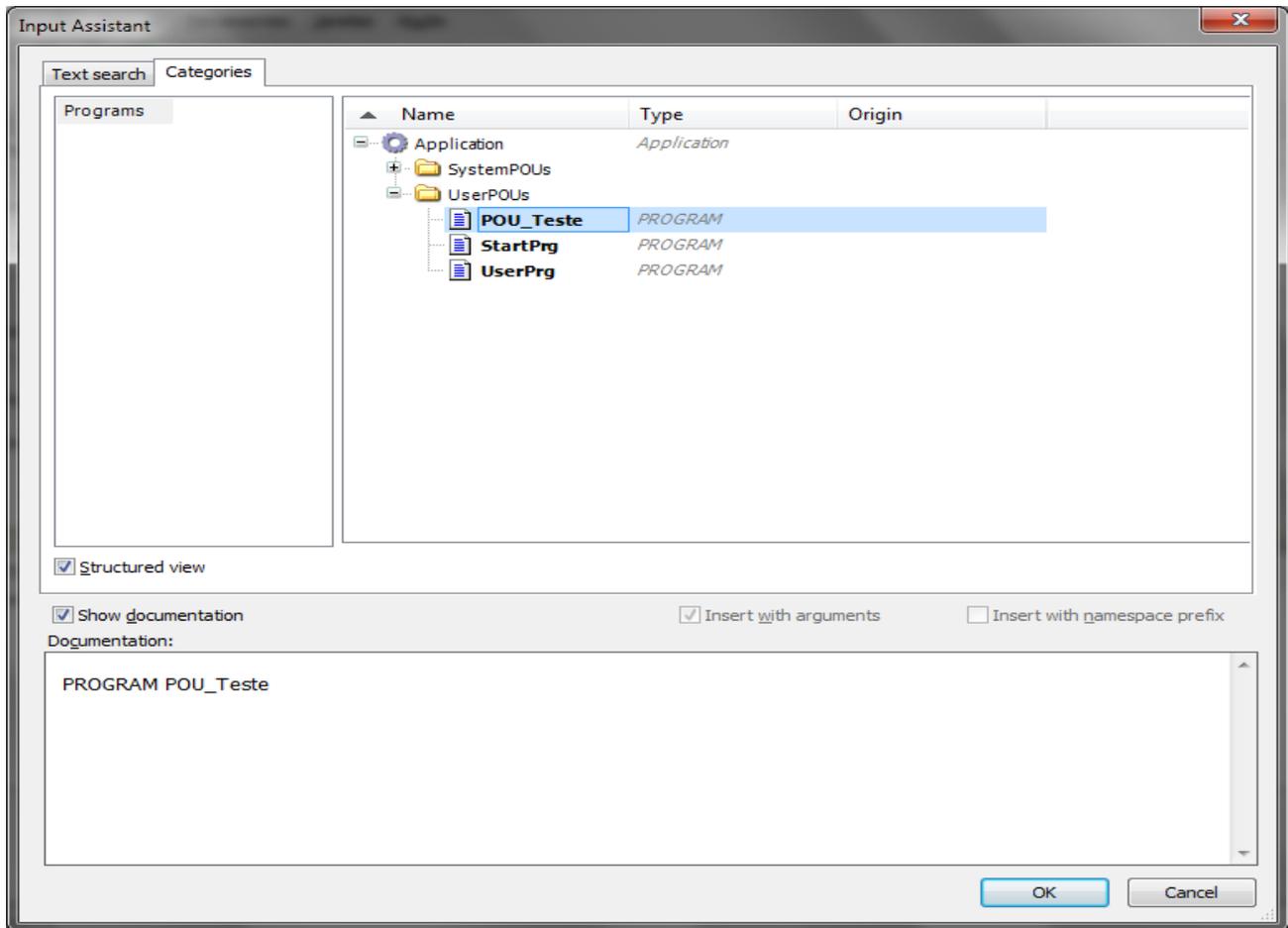


Figura 2-16. Vinculando POUs às Tarefas

Outra opção é a chamada de uma POU dentro de uma POU já instanciada no programa, segue exemplo onde a POU\_Teste será chamada dentro da área destinada a código da POU UserPrg:

```
POU_Teste ();
```

### Número Máximo de Tarefas

O número máximo de tarefas que o usuário poderá criar somente está definido para o perfil Personalizado, ou seja, o único que tem essa permissão. Os demais já têm as suas tarefas criadas e configuradas.

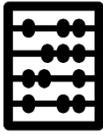
### Pesquisa na Documentação: número máximo de tarefas

Consulte, na documentação do produto, a quantidade máxima de tarefas IEC por UCP e perfil de projeto, sendo que as instâncias de protocolo também são consideradas tarefas de comunicação pela UCP.

Fonte: Manual de Utilização Hadron Xtorm – MU223000!



# Parametrização da Aplicação



## Configurando a UCP

A configuração da UCP Hadron Xtorm baseia-se em configurar os parâmetros da UCP sobre o Cão de guarda, troca a quente, sincronismo de tempo, pontos internos, conversão de engenharia, alarmes e agrupamento de eventos.

O usuário deverá dar dois cliques na UCP Hadron Xtorm, localizada na árvore de dispositivos, conforme mostra a figura a seguir, e configurar os campos associados. Esse tópico será explorado em detalhes na sequência desse Tutorial.

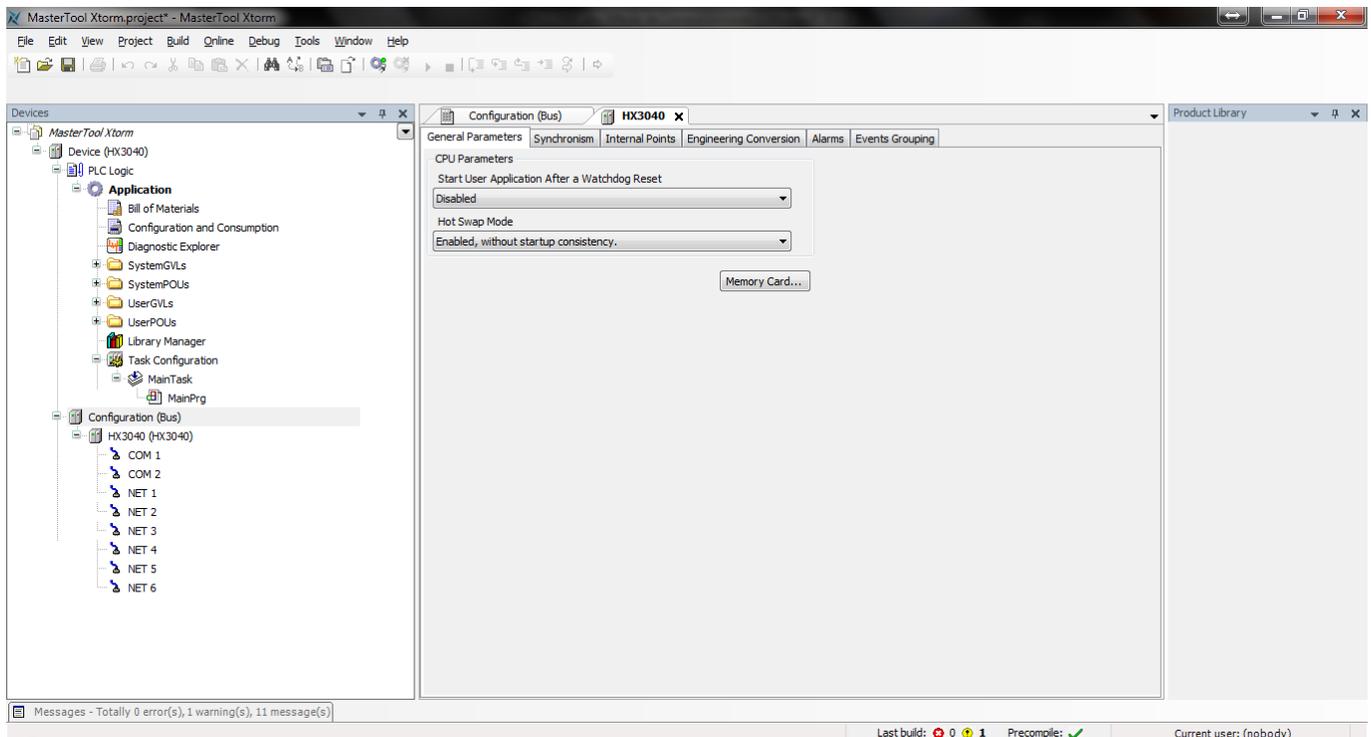


Figura 2-17. Configuração da UCP

Além disso, para que seja possível a comunicação entre a UCP e o MasterTool Xtorm, a interface Ethernet NET 1 deverá ser configurada. Esse tópico será explorado em detalhes na sequência desse Tutorial. Clicando duas vezes sobre o ícone NET 1 da UCP, na árvore de dispositivos, surgirá uma nova aba para configuração da rede de comunicação em que o módulo está conectado.

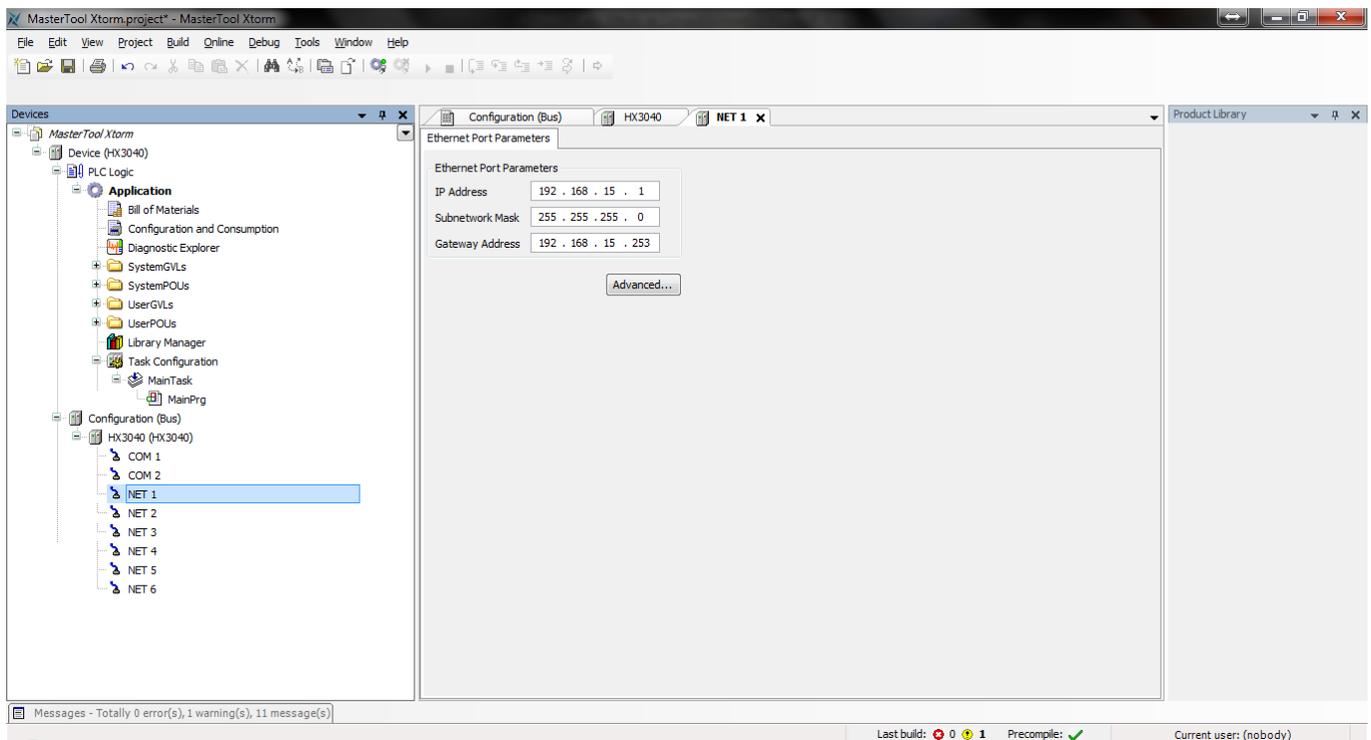


Figura 2-18. Configurando a Porta de Comunicação da UCP

Caso a UCP com o IP configurado não seja encontrada na rede ou a UCP que está atualmente ativa tenha um IP diferente da ativa configurada no projeto, uma mensagem surgirá na tela durante o Login, solicitando ao usuário a possibilidade de substituir o IP antigo pelo configurado (opção Sim) ou Não para não enviar o projeto.

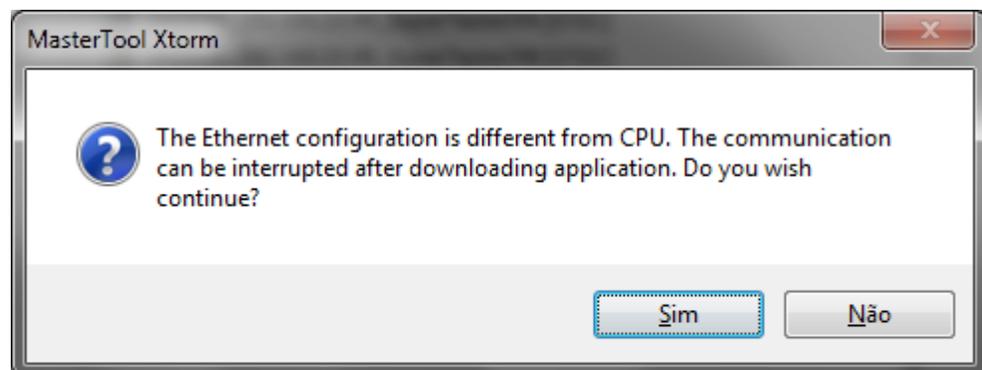


Figura 2-19. Aviso de configuração IP

## Bibliotecas

Existem diversos recursos da ferramenta de programação que estão disponíveis através de bibliotecas. Sendo assim, os mesmos devem ser inseridos no projeto para que a sua utilização seja possível. O procedimento de inserção é bastante simples, sendo necessário selecionar o item "Library Manager", disponível na treeview do menu Dispositivo e selecionar "Add Library", conforme mostra a figura a seguir.

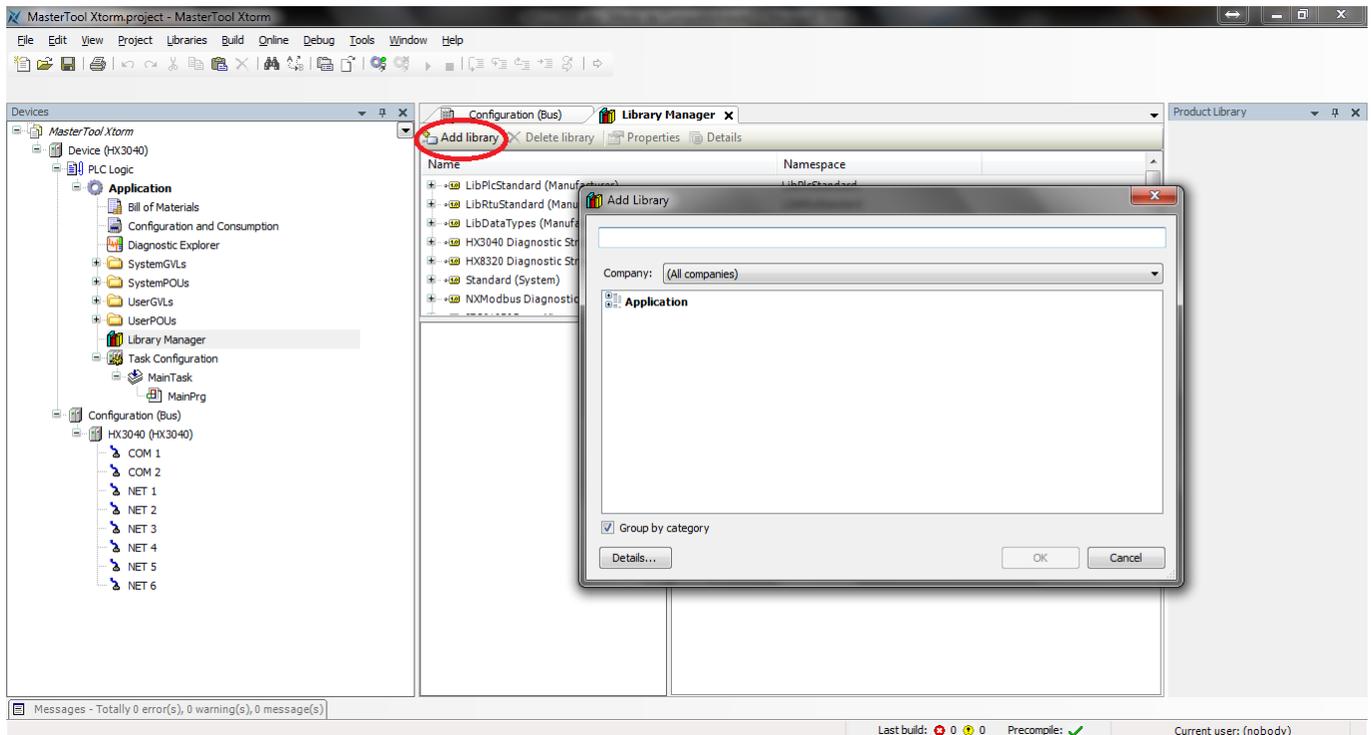


Figura 2-20. Inclusão de uma Biblioteca no Projeto

Na sequência, deve-se selecionar a biblioteca desejada para inclusão no projeto, pressionando em seguida, o botão OK.

## Inserindo uma Instância de Protocolo

A UCP da Série Hadron Xtorm disponibiliza alguns protocolos, como o MODBUS. Basta adicionar e configurar a instância do protocolo desejado na interface de comunicação. Esse tópico será explorado em detalhes na sequência deste Tutorial.

Abaixo são descritos dois casos de inserção do protocolo MODBUS, sendo um na interface serial e outro na interface Ethernet, um caso de inserção do protocolo DNP3 Servidor e um caso de inserção do protocolo IEC 60870-5-104 Servidor. A inserção de protocolos Cliente é feita de forma similar.

### MODBUS RTU

O primeiro passo para configurar o MODBUS RTU, em modo escravo, é incluir a instância na porta COM desejada (nesse caso, COM 1). Clicar com o botão direito sobre a COM e selecionar "Add Device...", conforme mostra a figura a seguir.

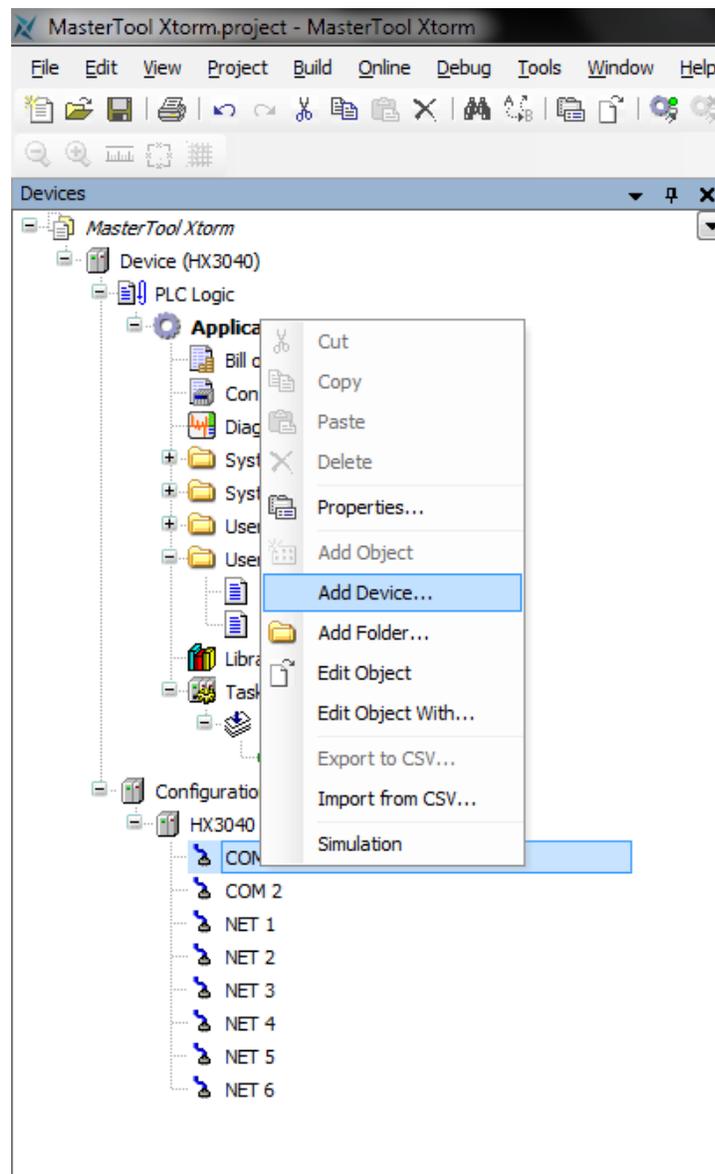


Figura 2-21. Adicionando a Instância

Após, surgirão na tela os protocolos disponíveis ao usuário. Nesse caso, selecionar o “MODBUS Slave” expandir a árvore de opções, selecionar o dispositivo e clicar em Add Device, conforme mostra a figura a seguir.

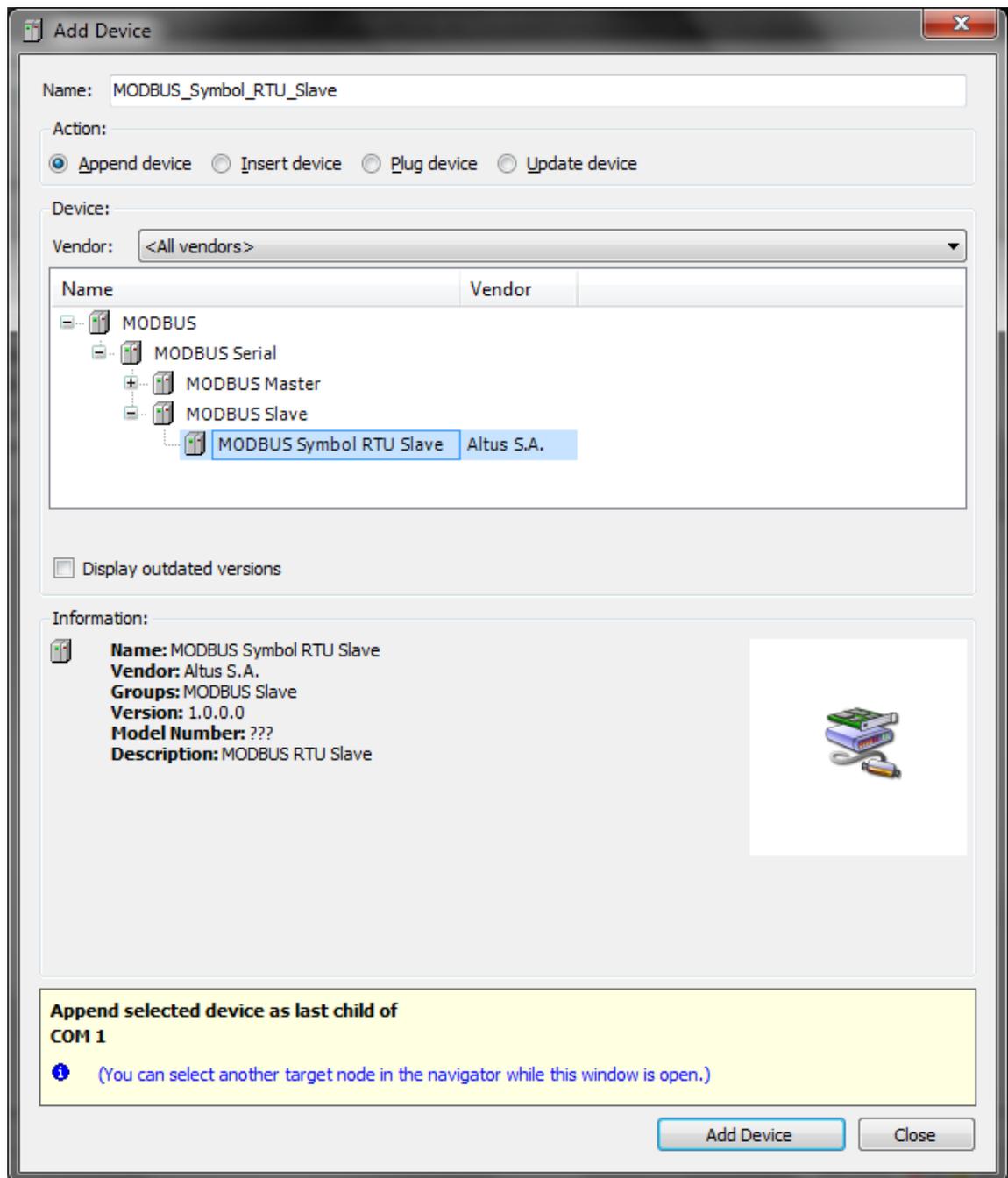


Figura 2-22. Selecionando o Protocolo

### MODBUS Ethernet

O primeiro passo para configurar o MODBUS Ethernet, em modo cliente, é incluir a instância na NET desejada (nesse caso NET 1, a UCP HX3040 possui seis interfaces Ethernet). Clicar com o botão direito sobre a NET e selecionar “Add Device...”, conforme mostra a figura a seguir.

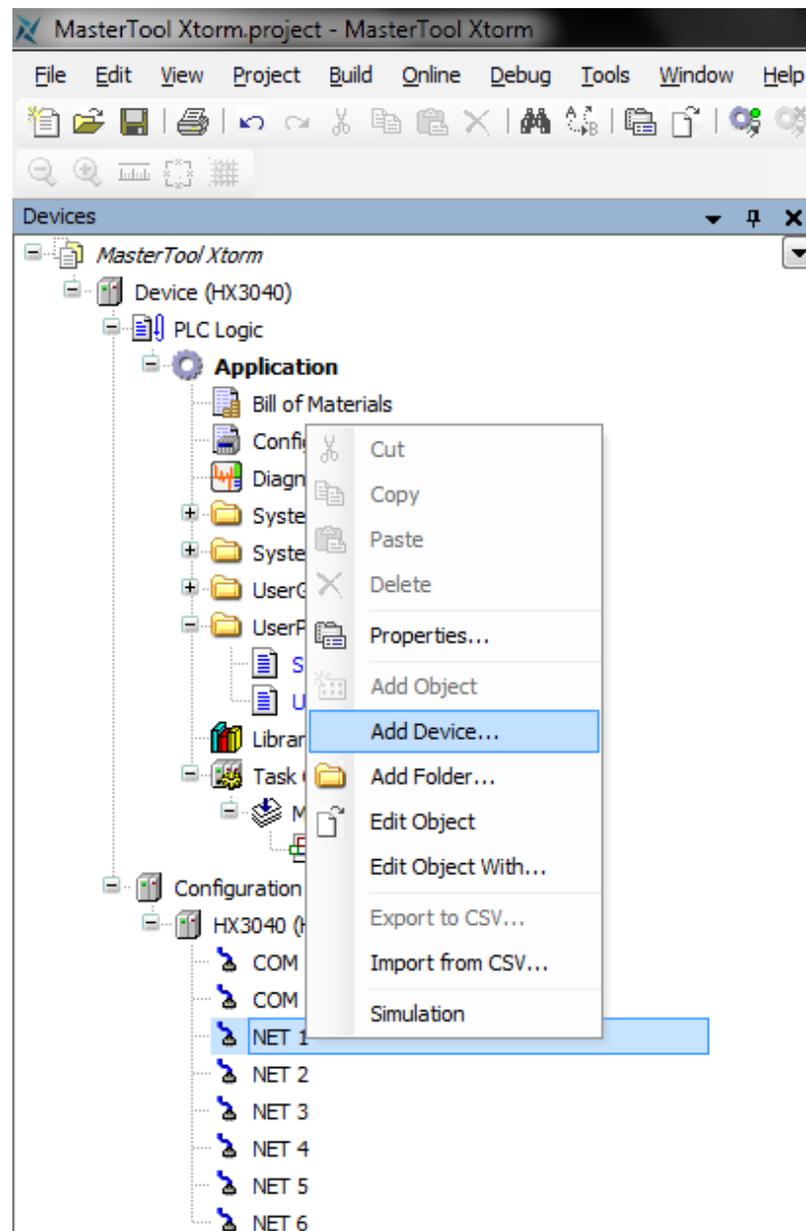


Figura 2-23. Adicionando a Instância

Após, surgirá na tela os protocolos disponíveis ao usuário. Nesse caso, selecionar "MODBUS > MODBUS Ethernet > MODBUS Client > MODBUS Symbol Client" e clicar em Add Device, conforme mostra a figura a seguir.

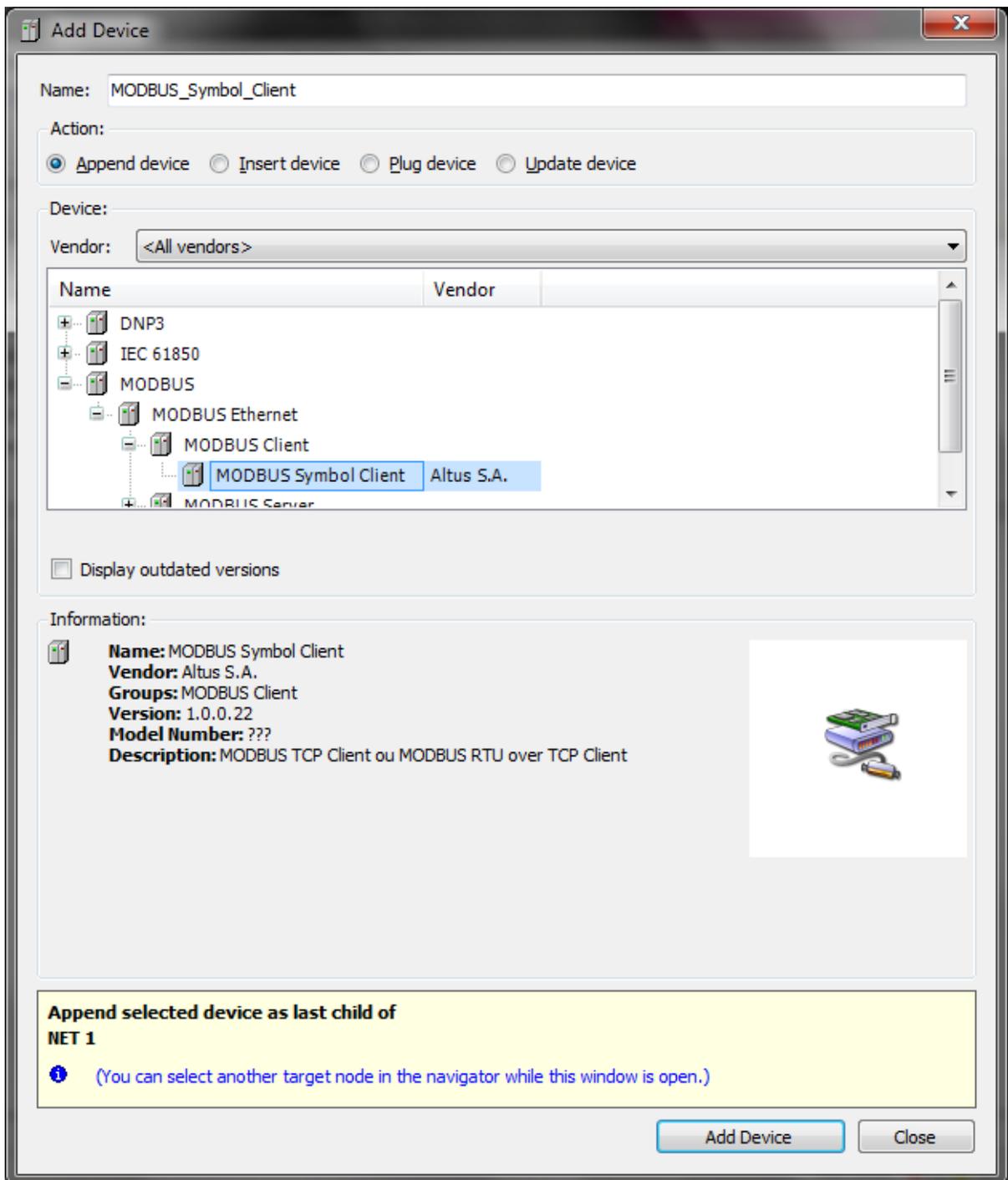


Figura 2-24. Selecionando o Protocolo

### DNP3 Servidor

O primeiro passo para configurar o DNP3, em modo servidor, é incluir a instância na NET desejada (nesse caso NET 1, a UCP HX3040 possui seis interfaces Ethernet). Clicar com o botão direito sobre a NET e selecionar “Add Device...”, conforme mostra a figura a seguir.

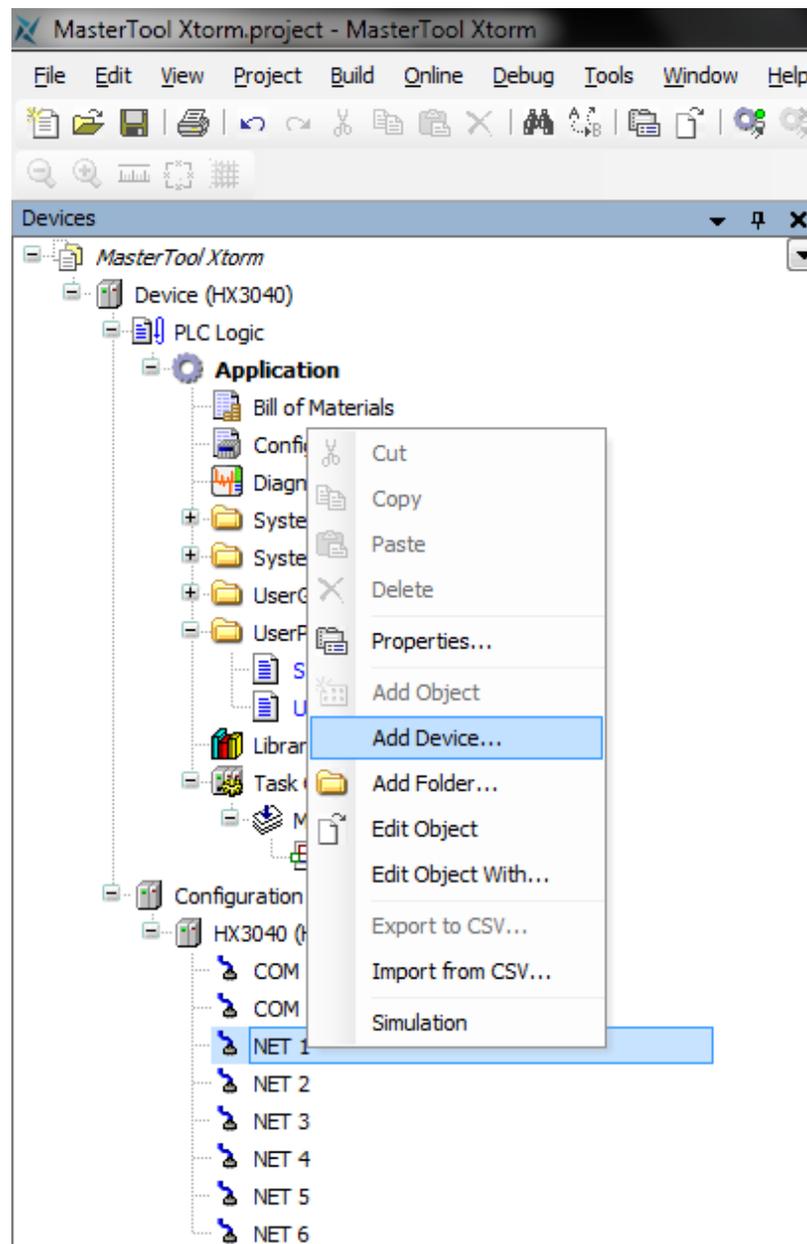


Figura 2-25. Adicionando a Instância do DNP3 Servidor

Após, surgirá na tela os protocolos disponíveis ao usuário. Nesse caso, selecionar o “DNP3 > DNP3Ethernet > DNP3 Server > DNP3 Server” e clicar em Add Device, conforme mostra a figura a seguir.

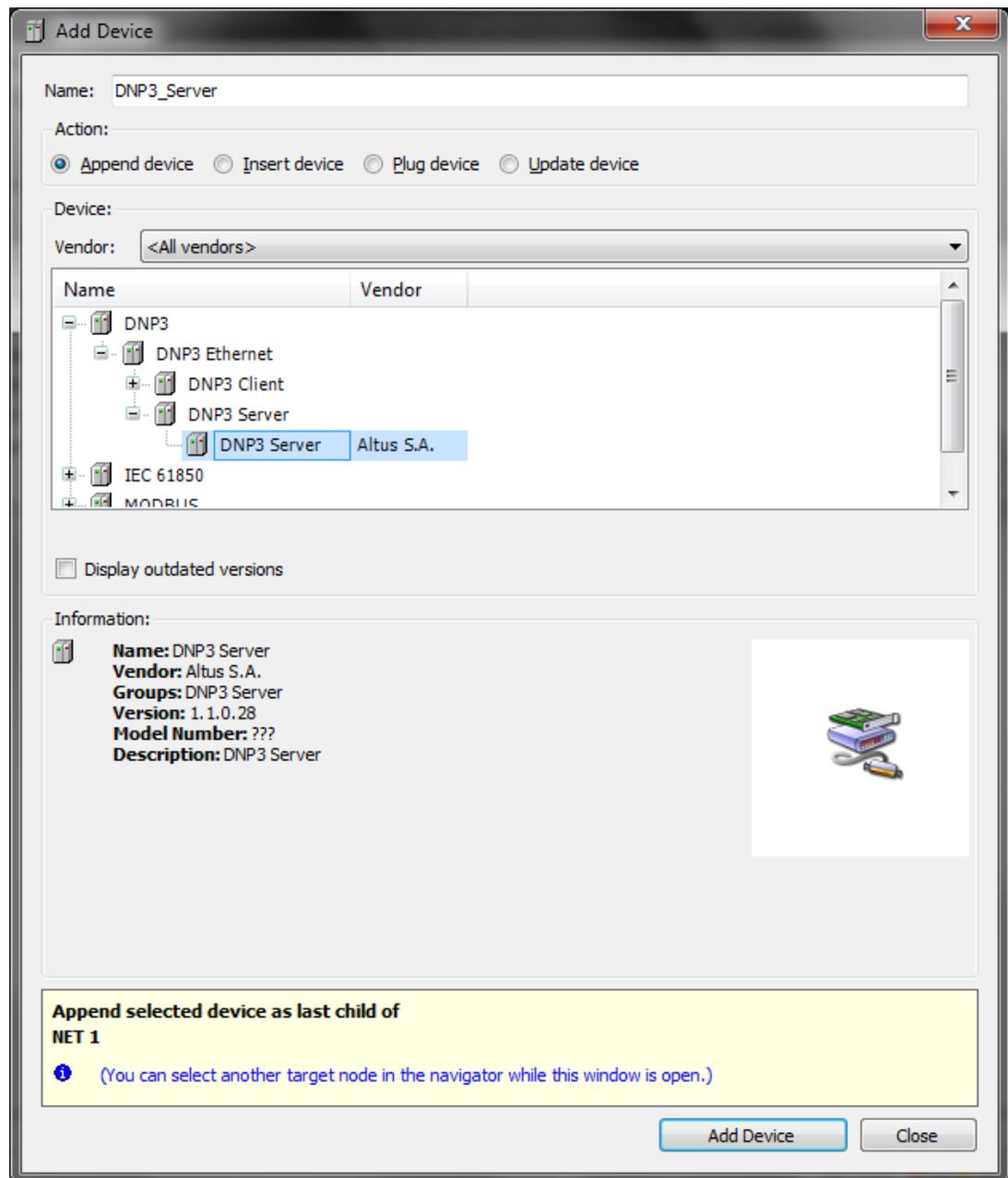


Figura 2-26. Selecionando o Protocolo DNP3 Servidor

### IEC 60870-5-104 Servidor

O primeiro passo para configurar o driver de comunicação IEC 60870-5-104, em modo servidor, é incluir a instância na NET desejada (nesse caso NET 1, a UCP HX3040 possui seis interfaces Ethernet). Clicar com o botão direito sobre a NET e selecionar “Add Device...”, conforme mostra a figura a seguir.

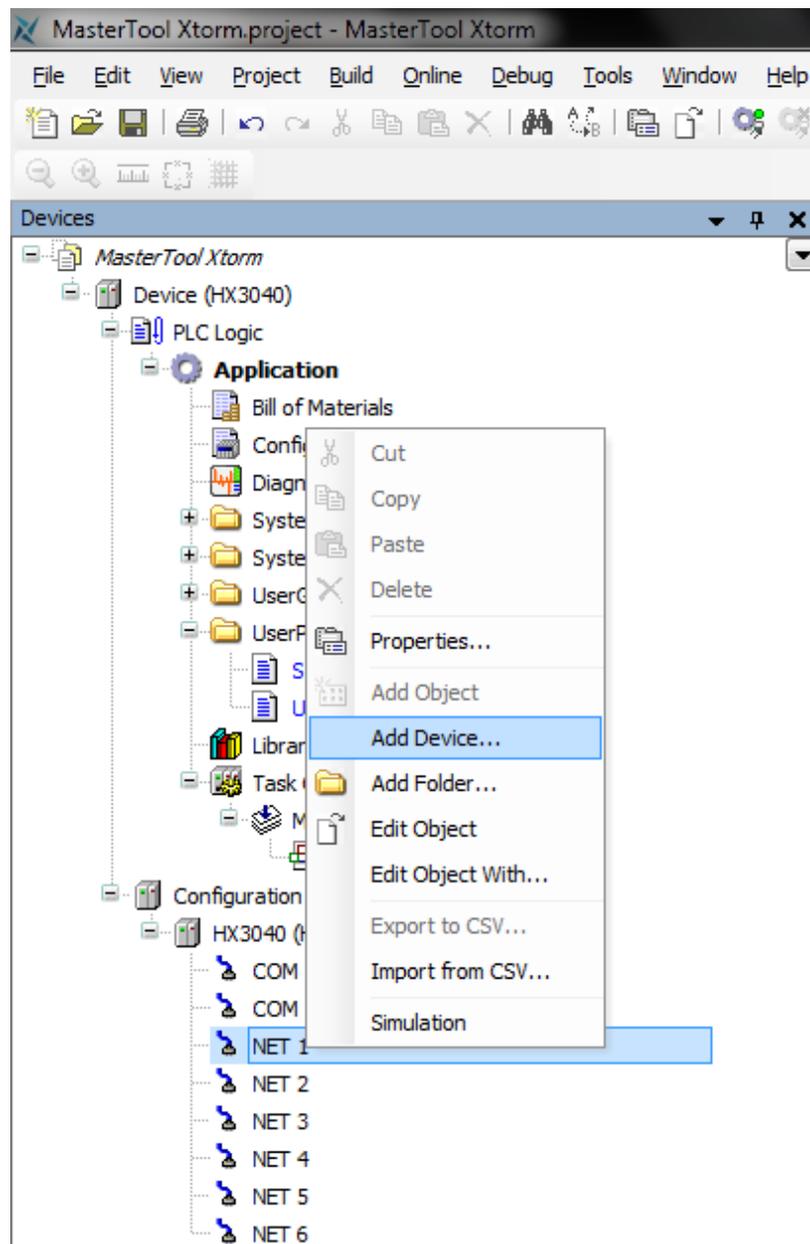


Figura 2-27. Adicionando a Instância do IEC 60870-5-104 Servidor

Após, surgirá na tela os protocolos disponíveis ao usuário. Nesse caso, selecionar o "IEC 60870-5-104 > IEC 60870-5-104 Server > IEC 60870-5-104 Server" e clicar em Add Device, conforme mostra a figura a seguir.

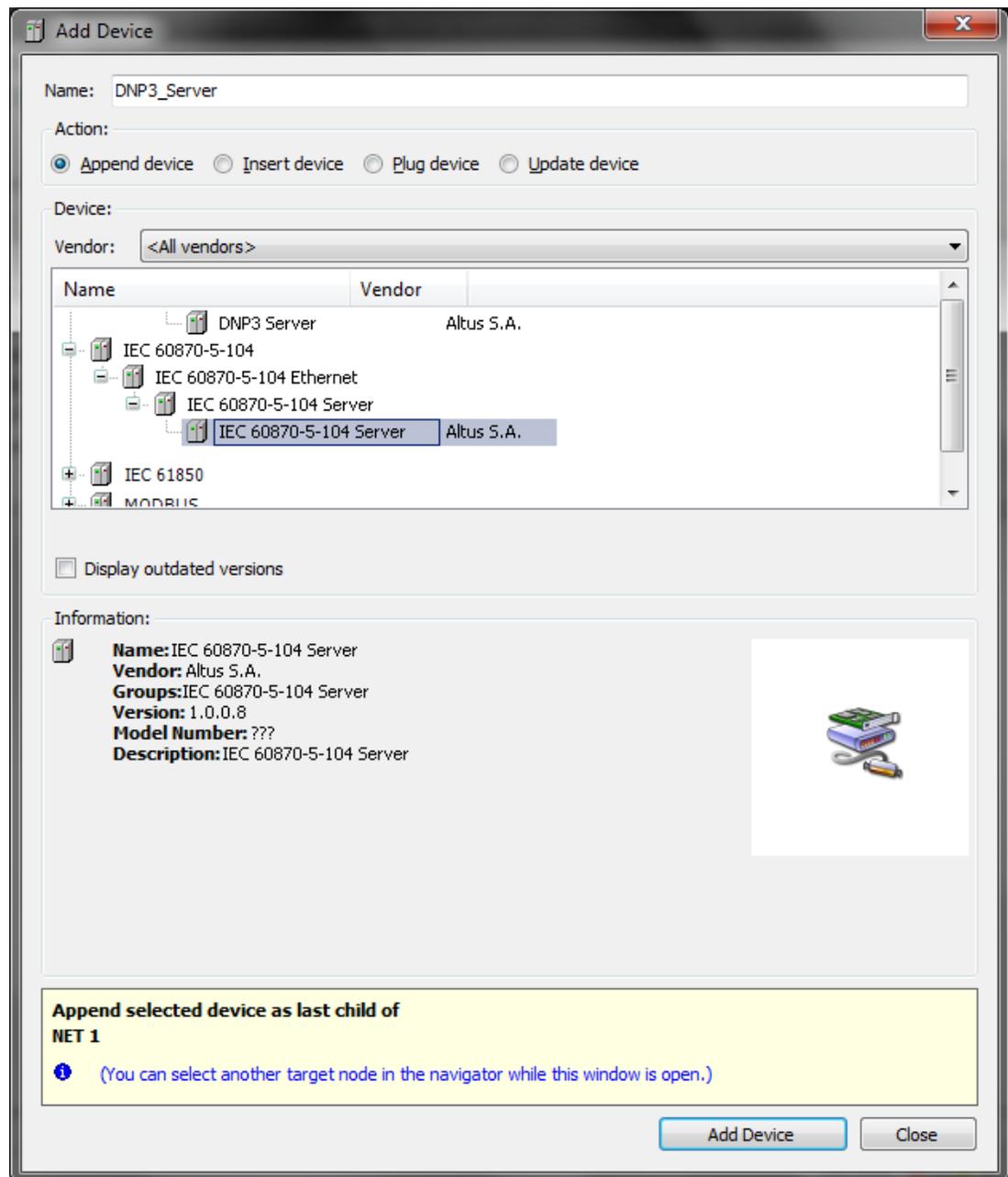


Figura 2-28. Selecionando o Protocolo IEC 60870-5-104 Servidor

### Localizando a Rede

Como existe a possibilidade de mais UCPs estarem conectadas à rede, o usuário deve localizar todas as unidades de comunicação e selecionar a desejada.

Inicialmente, será necessário acessar a opção Device, na árvore de dispositivos, clicando duas vezes. Na aba "Communication Settings", selecionar o Gateway, ou, caso nenhum Gateway exista ou o usuário queira adicionar um novo Gateway, clicar no botão "Add Gateway", configurando seu IP na janela que será aberta. Para mapear os dispositivos presentes na rede, deve-se então clicar no botão "Scan Network".

Então, deve-se aguardar enquanto o software MasterTool Xtorm procura e apresenta as UCPs disponíveis na rede.

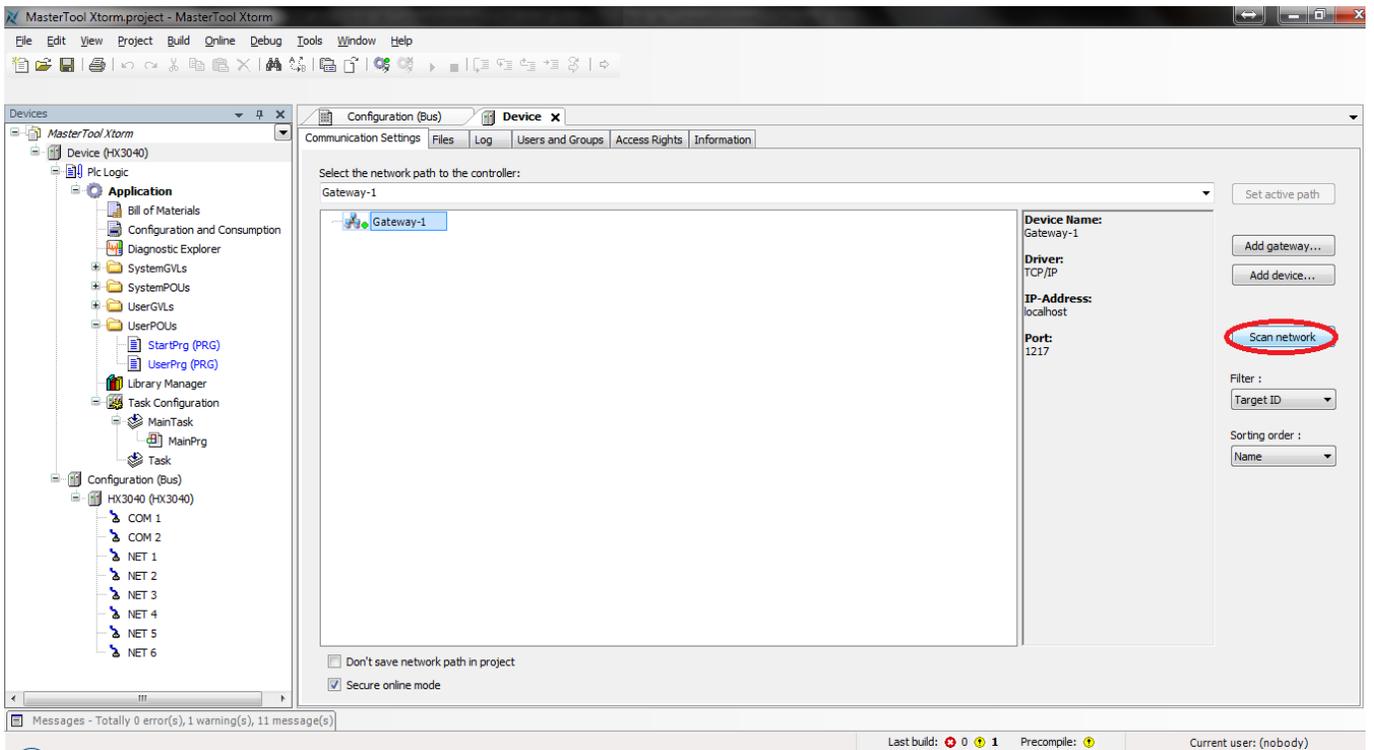


Figura 2-29. Localizando a UCP

Após, basta selecionar a UCP desejada e clicar na opção “Set Active Path”, fazendo com que ela fique ativa e o configurador saiba com quem deve comunicar e enviar o projeto.

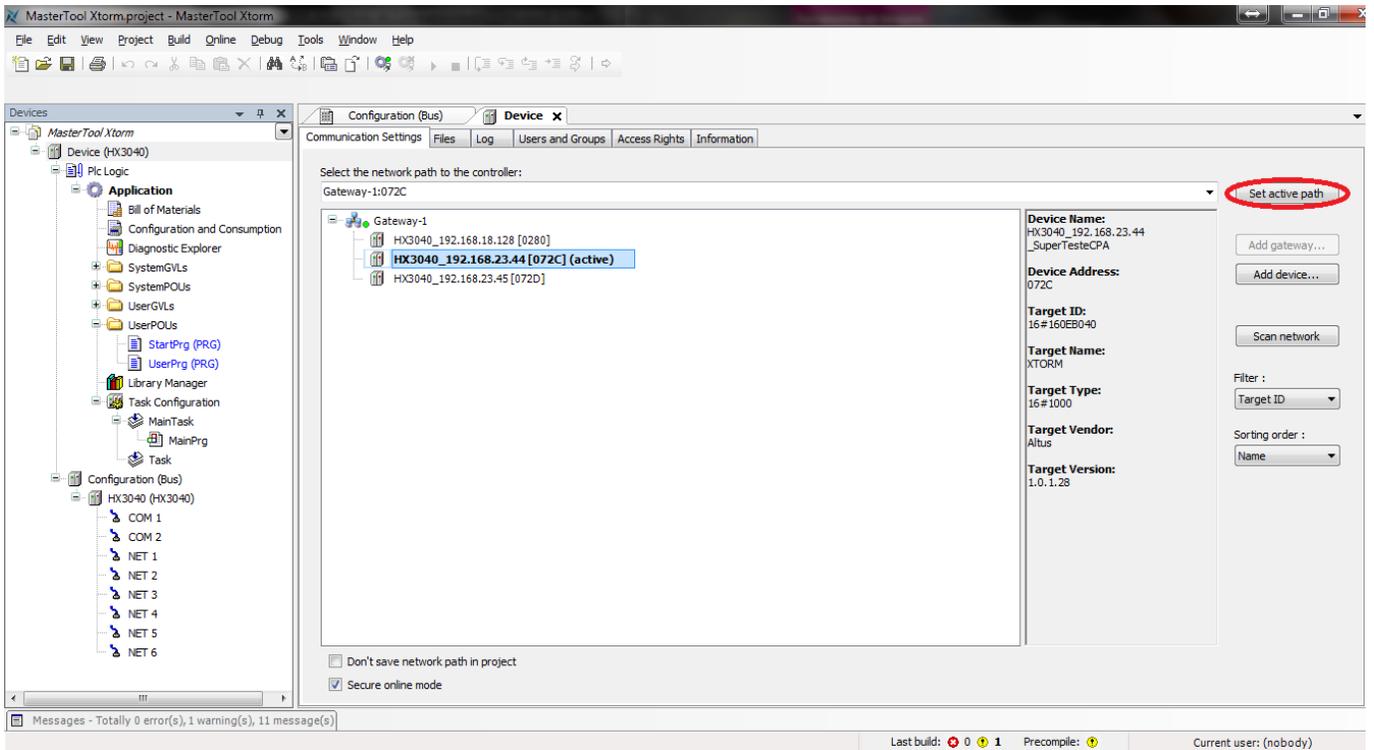


Figura 2-30. Ativando a UCP

Caso seja necessário, o usuário pode alterar o nome padrão do dispositivo que é exibido. Para isso, deve clicar com o botão direito do mouse sobre o dispositivo desejado e selecionar a opção "Change Device Name". Após uma mudança de nome, o dispositivo não voltará ao nome padrão em nenhuma circunstância.

# Compilação, Execução e Teste da Aplicação

## Compilando um Projeto



A fim de realizar a verificação da aplicação criada, o usuário deve executar a compilação do projeto. Essa é a forma mais eficaz de se encontrar problemas ou receber avisos sobre alguns equívocos cometidos durante a configuração do produto e edição da aplicação. Para executar tal procedimento, basta acessar o menu Build e clicar em “Generate Code” conforme mostra a figura a seguir.

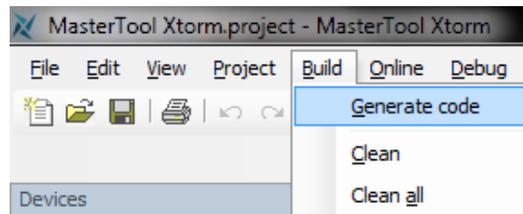


Figura 2-31. Compilando o Projeto

Depois do tempo de processamento, o qual vai variar de acordo com o tamanho da aplicação do usuário, os erros e mensagens de alerta, caso sejam necessários, serão mostrados abaixo, conforme ilustrado na sequência.

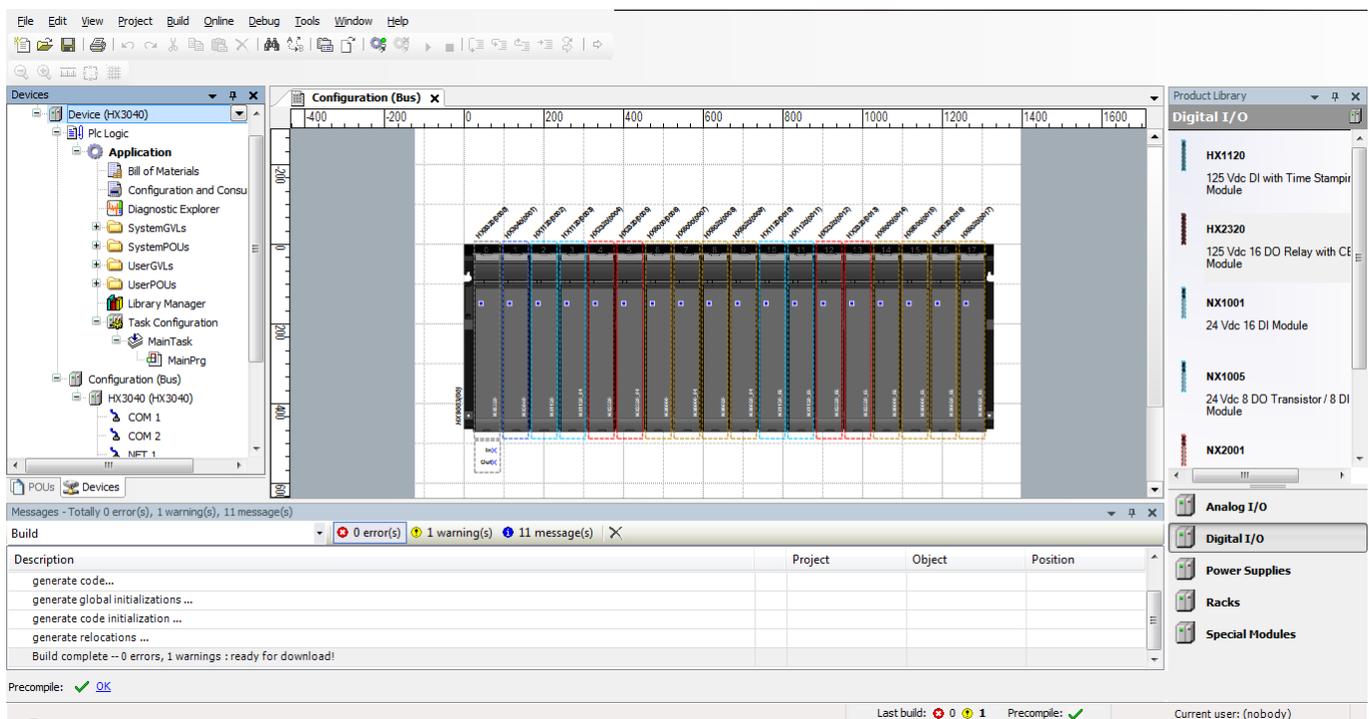


Figura 2-32. Mensagens da Compilação

Caso os erros e mensagens não estejam visíveis na tela, basta acessar o menu Visualizar e clicar em “Messages”, conforme mostra a figura a seguir.

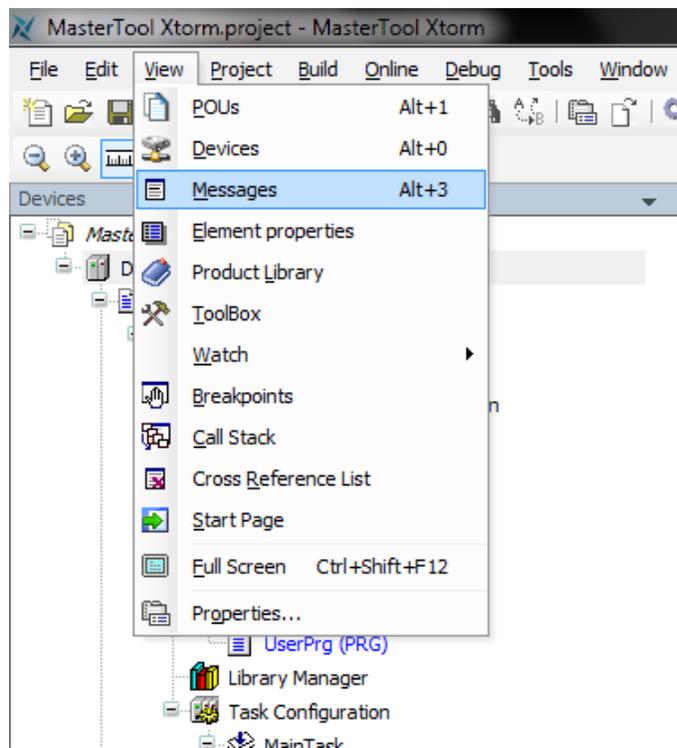


Figura 2-33. Incluindo as Mensagens na Tela

## Login

Após compilar a aplicação e corrigir todos os erros encontrados, é o momento de enviar o projeto para a UCP. Para que isso seja possível, basta efetuar a operação de Login no software MasterTool Xtorm. Essa operação pode levar alguns segundos, dependendo do tamanho do arquivo gerado.

Para executar o Login, basta acessar o menu Online e clicar na opção "Login", conforme mostra o exemplo da figura a seguir.

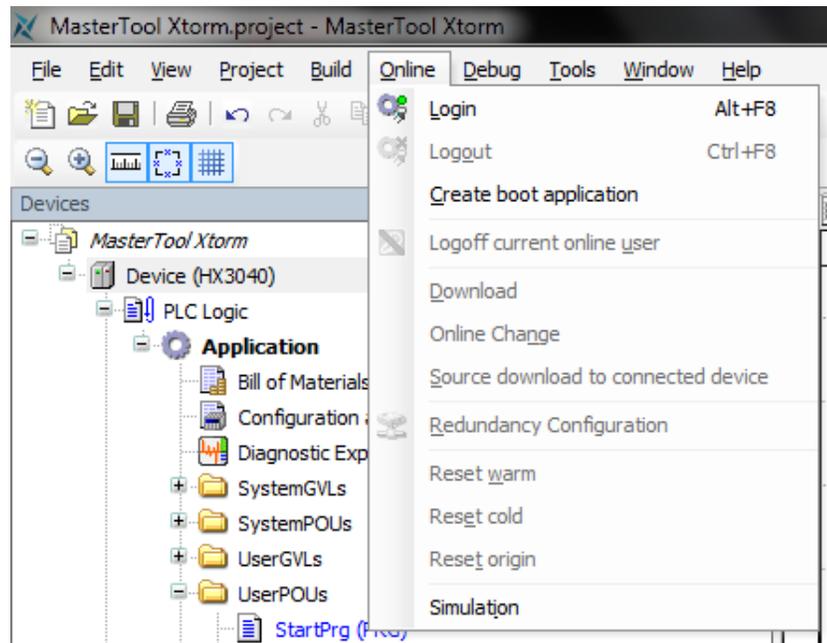


Figura 2-34. Enviando o Projeto para a UCP

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

A figura a seguir mostra a mensagem que o MasterTool Xtorm irá apresentar em caso do novo projeto, que está sendo enviado seja diferente de um projeto já existente dentro da UCP. As opções disponíveis são as seguintes:

- **Login with Online Change** - Executa o Login e envia o novo projeto sem que a aplicação atual da UCP seja parada (ver item Modo Run), atualizando as alterações quando um novo ciclo é executado;
- **Login with Download** - Executa o Login e envia o novo projeto com a UCP parada (ver item Modo Stop). Quando a aplicação for iniciada, as atualizações já terão sido realizadas;
- **Login Without Any Change** - Executa o Login sem enviar o novo projeto.

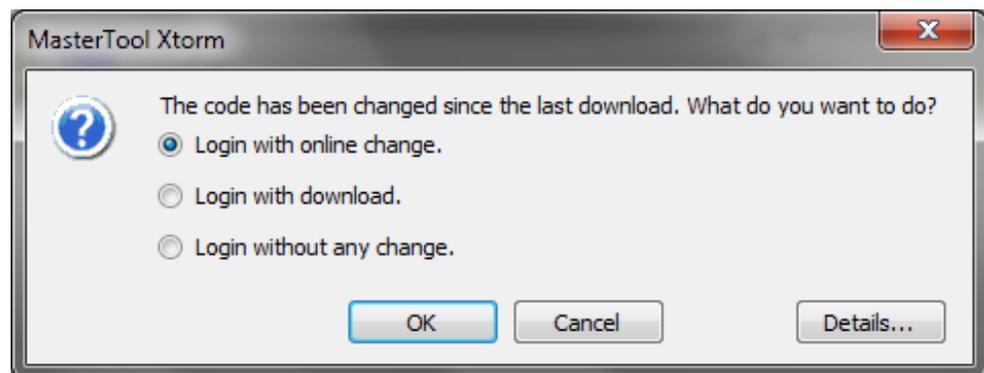


Figura 2-35. Atualização do Projeto na UCP

A figura a seguir mostra a mensagem que o MasterTool Xtorm exibe quando somente foram realizadas alterações em variáveis da aplicação, fazendo com que não seja possível enviar o novo projeto e o mesmo ser atualizado em um novo ciclo da UCP, a qual está em execução (ver tópico: Modo Run). Dessa forma, o MasterTool Xtorm solicita ao usuário se o Login deve ser

executado como download e a UCP para (ver tópico: Modo Stop) ou a operação deve ser cancelada.

O botão “Details...” apresenta as modificações realizadas na aplicação.

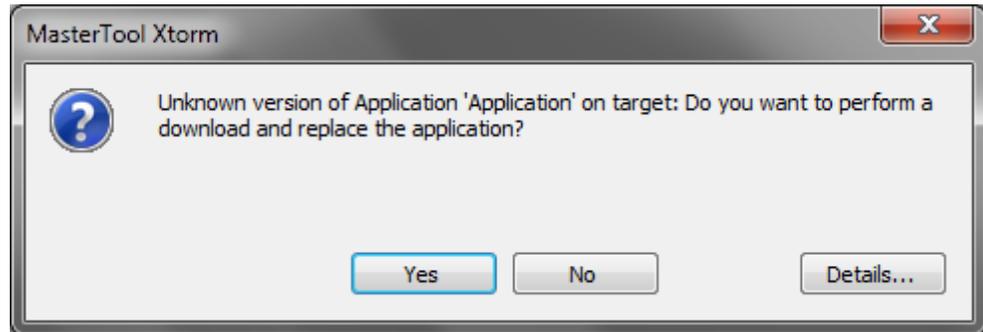


Figura 2-36. Alteração de Variáveis

Caso seja o primeiro envio de aplicação para a UCP, a mensagem da figura a seguir aparecerá na tela do MasterTool Xtorm.

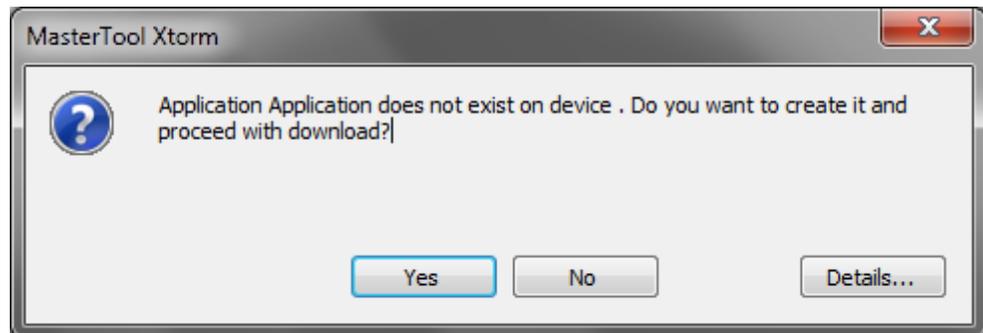


Figura 2-37. Primeiro Envio de Aplicação

### Modo Run

Logo após enviar o projeto para a UCP, a aplicação não será executada imediatamente (somente se for realizada uma alteração online), antes que seja selecionado um novo comando “Iniciar”. Essa função possibilita ao usuário controlar a execução da aplicação enviada para a UCP. Além disso, permite que valores iniciais sejam pré-configurados, para que, no primeiro ciclo já sejam atualizados na UCP.

Para selecionar tal funcionalidade, basta acessar o menu Debug e clicar em “Start”, conforme mostra a figura a seguir.

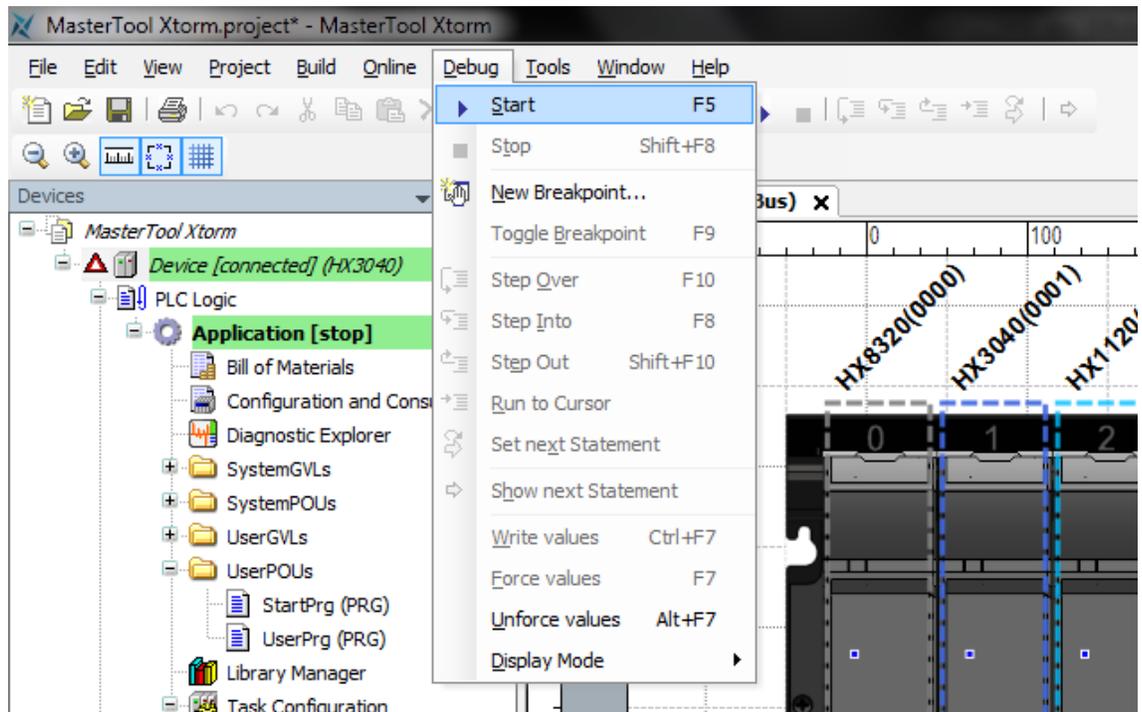


Figura 2-38. Iniciando a Aplicação

A figura a seguir mostra a aplicação em execução. Caso seja selecionada a aba de uma POU, as variáveis criadas serão listadas em uma janela de monitoração, na qual valores podem ser forçados e visualizados pelo usuário. Caso as variáveis sejam forçadas através do comando F7 do teclado, a UCP indicará essa condição no visor gráfico.

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

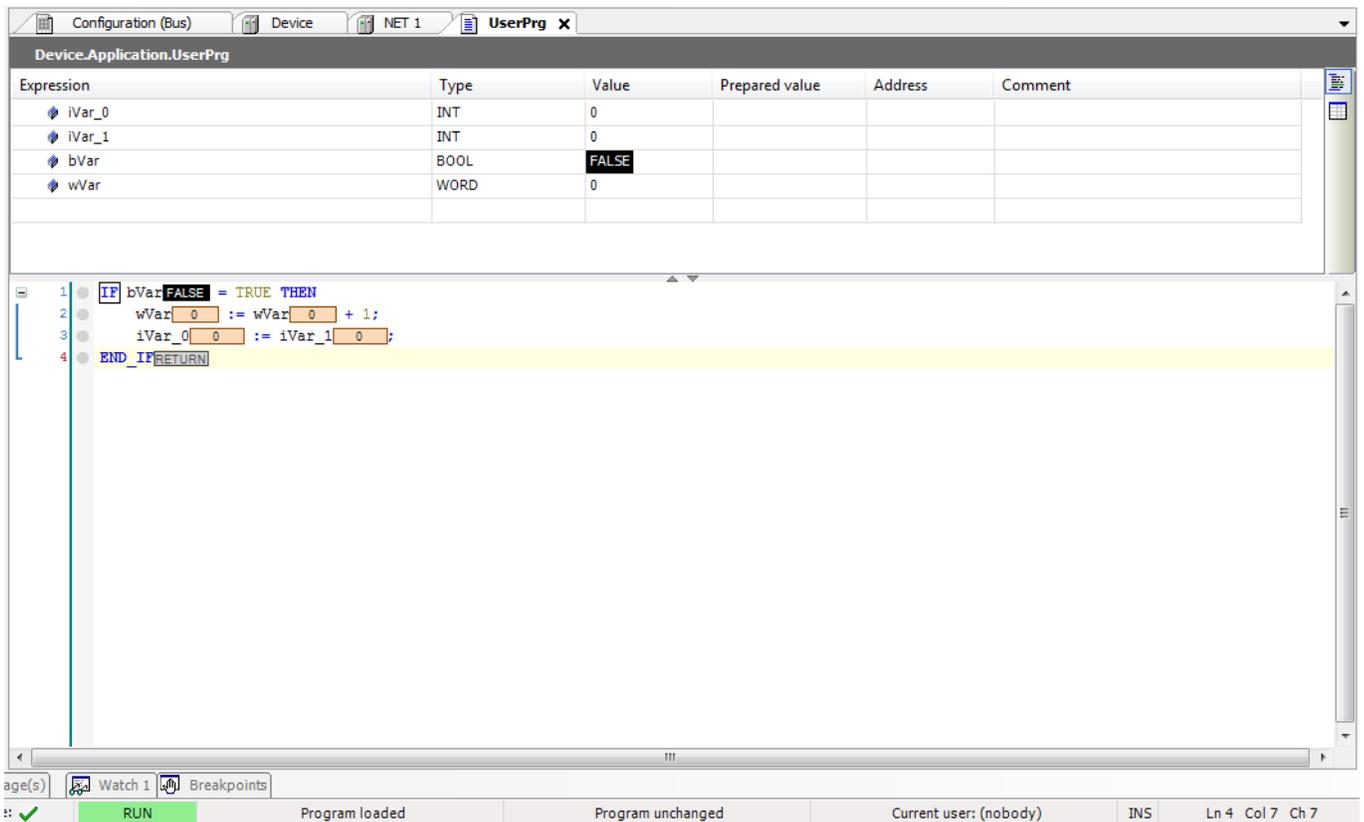


Figura 2-39. Programa em Execução

Caso a UCP seja inicializada com uma aplicação já gravada internamente, ela automaticamente entra em Modo Run, sem a necessidade de realizar o comando via MasterTool Xtorm.

### Modo Stop

Para interromper a execução da UCP, sem que o software MasterTool Xtorm perca a conexão com ela, o usuário deve seleccionar a opção Parar, disponível no menu Debug, "Stop", conforme mostra a figura a seguir.

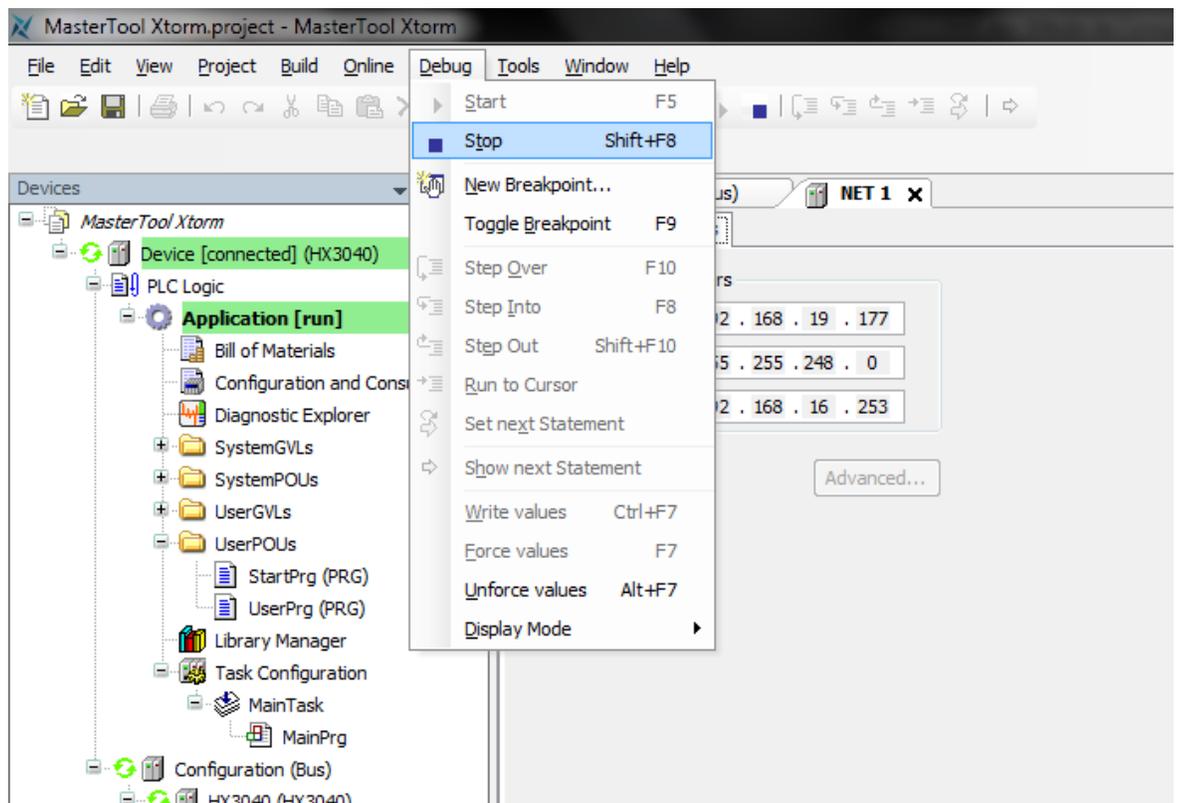


Figura 2-40. Parando a Aplicação

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

## Monitoração, Escrita e Forçamento de Variáveis

Após efetuar um Login em uma UCP, o usuário poderá monitorar variáveis do projeto de 3 formas distintas:

1. Diretamente na POU (na área de declaração de variáveis ou na área do programa);
2. Via janela de monitoração (menu View -> Watch);
3. Utilizando a funcionalidade Trace, a qual realiza a gravação e exibição dos valores das variáveis internamente na UCP.

### NOTA:

No modo online há uma limitação de 25000 entradas de variáveis monitoráveis em POUs editadas com o editor ST, o usuário será alertado quando o limite for ultrapassado com um erro de compilação.

Além disto, também é possível escrever ou forçar valores em variáveis do projeto.

O comando de escrita (CTRL + F7) escreve um valor em uma variável, podendo este valor ser sobrescrito pelas instruções executadas na aplicação no próximo ciclo de execução.

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

O forçamento de variáveis pode ser realizado somente em modo Online na UCP. Variáveis de diagnóstico não podem ser forçadas, apenas escritas, pois diagnósticos são providos pela UCP e devem poder ser sobrescritos por ela.

Quando for efetuada uma escrita forçada em uma variável redundante da UCP Ativa, a MainTask da aplicação sofrerá um impacto em seu tempo de execução, tanto na UCP Ativa, quando na UCP Reserva. Isto porque as duas UCPs irão trocar a cada ciclo informações sobre as variáveis forçadas. Portanto, quando for forçar variáveis em um sistema redundante, deve-se considerar o acréscimo que pode ter a execução da tarefa.

Para mais informações sobre forçamentos de variáveis no sistema redundante, consulte o tópico Sincronização de Dados Redundantes neste Tutorial.

🔔 **NOTA:**

Quando uma UCP está com variáveis forçadas e é desenergizada, na próxima inicialização as variáveis perderão o forçamento. O limite de forçamentos para a UCP Hadron Xtorm é de 128 variáveis, independente do modelo de UCP ou configuração utilizados.

### **Variáveis utilizadas em mais de uma origem**

Uma origem define-se com um ponto de um módulo de E/S ou de um driver de comunicação que escreve em uma variável.

Quando um projeto é compilado, o MasterTool Xtorm verifica todos os pontos e informa se alguma variável está sendo utilizada em mais de uma origem. Essa verificação é realizada somente para os seguintes campos:

- HX3040;
- Dispositivo MODBUS do MODBUS Cliente;
- Dispositivo MODBUS do MODBUS Mestre;
- Outstation do DNP3 Cliente;
- Módulo de E/S.

### **Logout**

Caso a opção do usuário seja finalizar a comunicação com a UCP deve ser utilizado comando "Logout", localizado no menu Online, "Logout", conforme mostra a figura a seguir.

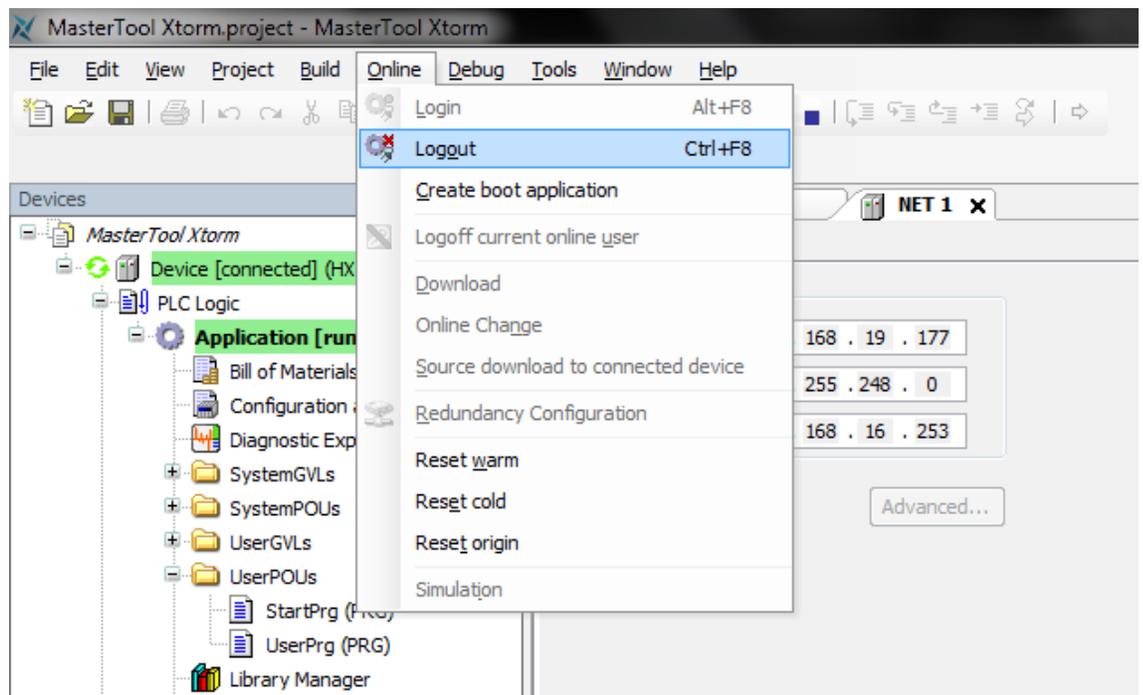


Figura 2-41. Interrompendo a Comunicação com a UCP

## Modo Simulação

O MasterTool Xtorm possui um importante recurso de simulação que permite ao usuário testar sua aplicação sem a utilização do equipamento, conferindo maior agilidade no desenvolvimento do programa. No entanto, alguns recursos específicos, que dependem do hardware da UCP Hadron Xtorm, não são passíveis de simulação:

- Relógio RTC;
- Varredura do Barramento;
- Módulos de E/S;
- Portas Seriais;
- Comunicação Ethernet;
- Protocolos de comunicação MODBUS;
- Protocolos de comunicação DNP3;
- Protocolo de comunicação MMS;
- Protocolo de comunicação GOOSE;
- Operações em cartão de memória;
- Diagnósticos em variáveis;
- Diagnostic Explorer;
- Fila e agrupamento de eventos;
- Outras funções que acessem o hardware da UCP.

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

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

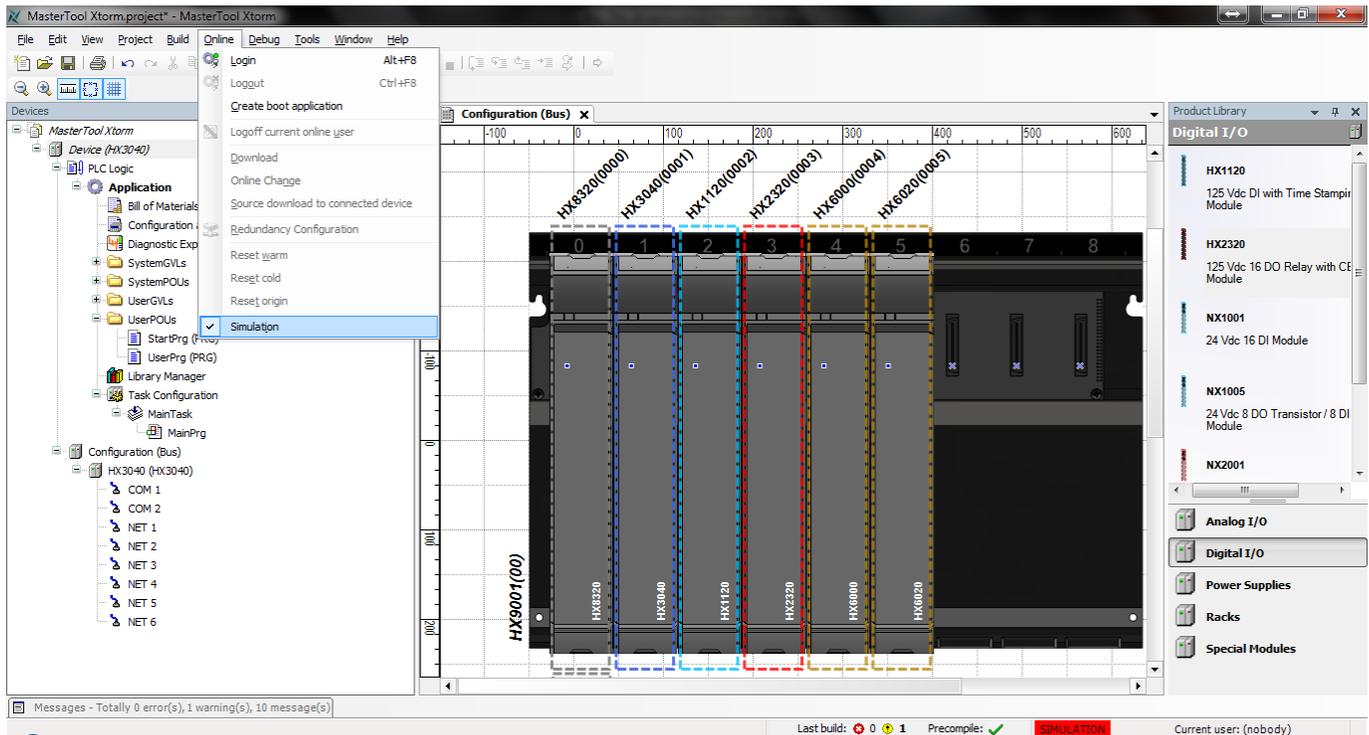


Figura 2-42. Modo Simulação

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

Neste caso um mesmo dado escrito por exemplo em %QD0, será escrito de forma diferente na simulação e na UCP da Série Hadron Xtorm. Se o dado escrito for 16#1234ABCD, a distribuição dos dados na memória da UCP ficará da seguinte maneira:

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

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

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

Tendo em vista estas diferenças e para facilitar o desenvolvimento da aplicação, utilizando os recursos do MasterTool Xtorm e da UCP da Série Hadron Xtorm é recomendado o uso de variáveis simbólicas. Neste caso as diferenças entre o Modo Simulação e o comportamento com a UCP da Série Hadron Xtorm não são verificadas. Portanto a melhor prática é evitar o uso de variáveis de representação direta sempre que possível para evitar o retrabalho quando desenvolver uma lógica que será testada em simulação e depois carregada em uma UCP.

O modo Simulação também pode ser utilizado para simular um projeto redundante, porém, terá as mesmas limitações citadas anteriormente, podendo ser testada apenas a lógica que não dependa do hardware. Neste caso, sempre serão executadas as POUs NonSkippedPrg e UserPrg, como se a UCP simulado fosse a UCP Ativa.

### Upload do Projeto

A UCP da Série Hadron Xtorm possibilita a gravação de um projeto na memória do produto, o qual pode ser recuperado e reutilizado através do software MasterTool Xtorm.

Para armazenar um projeto na memória do produto, a UCP deve estar conectada (Login) e a opção de envio de projeto, juntamente com o aplicativo, deve estar selecionada.

Para recuperar o projeto previamente armazenado, deve ser selecionado as opções, conforme mostra a figura a seguir.

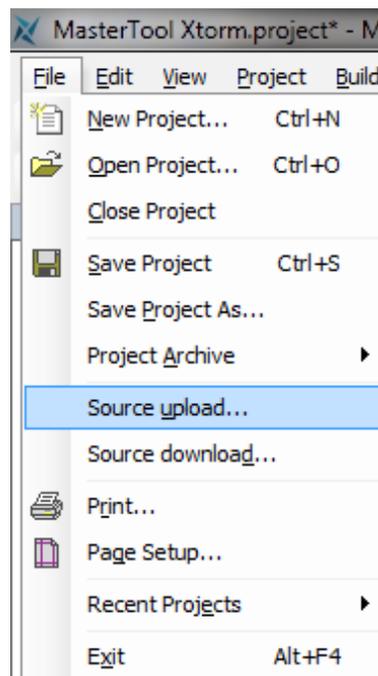


Figura 2-43. Opção de Upload de Projeto

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

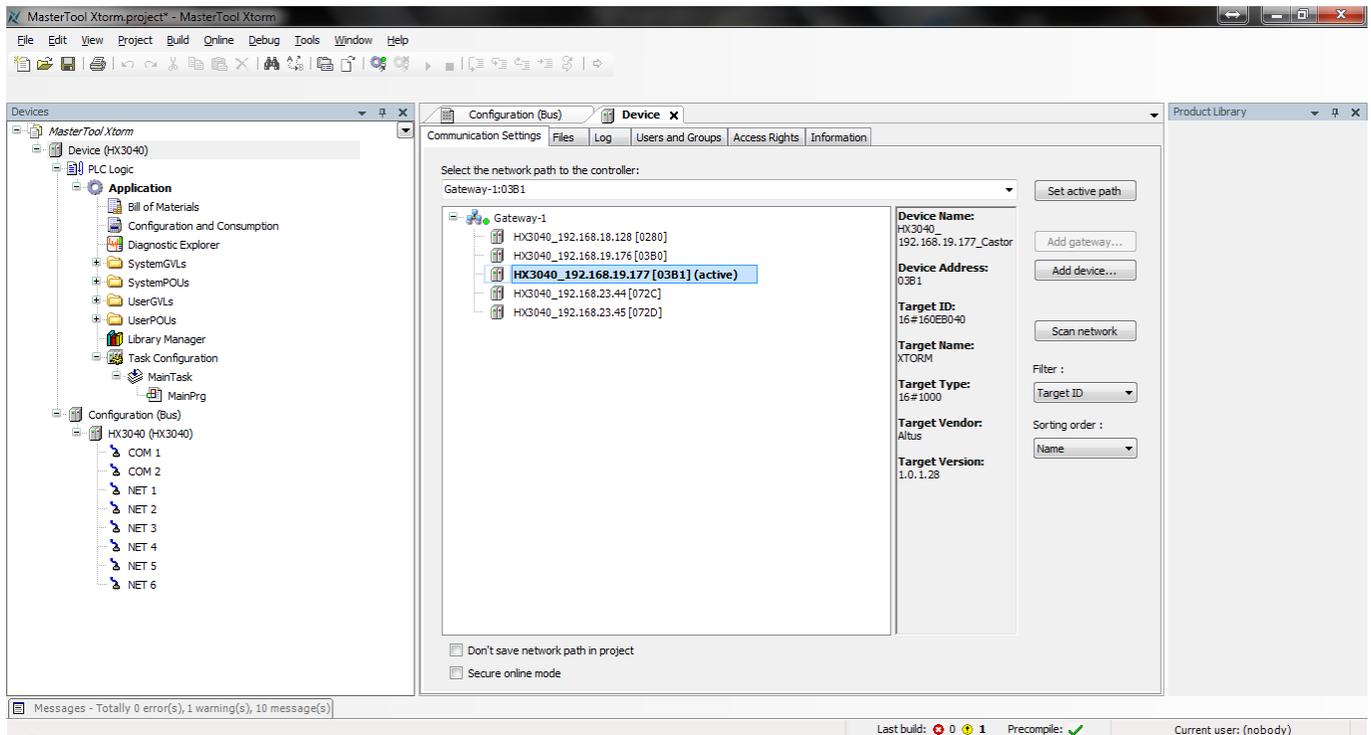


Figura 2-44. Selecionando a UCP

**NOTA:**

O Upload recupera o último projeto armazenado no controlador conforme descrito nos parágrafos anteriores. Caso ocorra apenas o carregamento para execução de um determinado aplicativo, o mesmo não poderá ser recuperado pelo procedimento de Upload.

## Estados de Operação da UCP

### Run

Quando uma UCP está em modo Run indica que todas as tarefas da aplicação estão em execução.

### Stop

Quando uma UCP está em modo Stop, indica que as tarefas da aplicação estão paradas. O valor das variáveis nas tarefas é mantido com o valor atual e as variáveis de saída assumem valores definidos pelo usuário.

Quando uma UCP passa para modo Stop a partir do envio de uma aplicação, as variáveis nas tarefas da aplicação serão perdidas exceto as variáveis do tipo persistente. As variáveis de saída assumirão o valor definido pelo usuário e em seguida o valor das saídas passará para o estado seguro. Assim que a nova aplicação for carregada as variáveis de saída assumirão novamente o valor definido pelo usuário.

### Breakpoint

Quando uma marca de depuração é atingida em uma tarefa, essa tarefa é interrompida. Todas as demais tarefas ativas na aplicação não irão ser interrompidas, elas continuarão a sua execução. Neste modo é possível percorrer um programa no modo Online. Um passo a passo pode ser executado e as posições das interrupções de depuração dependem do editor.

Quando em modo de ponto de interrupção (BRKP), o serviço de barramento fica completamente parado, logo a UCP que controla o barramento terá alguns comportamentos específicos neste estado, são eles:

- A troca a quente de qualquer módulo não é suportado em Breakpoint, o módulo ficará sem parametrização caso ocorra uma troca quente nesse estado;
- Os diagnósticos de todos os módulos serão mantidos congelados na aplicação;
- O Botão de diagnóstico dos módulos não estará disponível, ou seja, caso o usuário aperte o botão no estado de breakpoint, nada acontecerá.

### **Exception**

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

### **Reset a Quente**

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

### **Reset a Frio**

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

### **Reset Origem**

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

#### **🔔 NOTAS:**

1. Quando um Reset é executado os breakpoints definidos na aplicação são desabilitados.
2. Para executar um comando de Reset Quente, Frio ou Origem, é necessário estar com o MasterTool em modo Online na UCP.

### **Reset Process Command (IEC 60870-5-104)**

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

No caso de UCPs redundantes, o comando de reset do processo é sincronizado com a UCP não ativa, resultando na reinicialização das duas UCPs.

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

## Estudo Dirigido 2-1: criação de um projeto no MasterTool Xtorm



Crie um projeto-base no MasterTool Xtorm para servir de referência no decorrer do treinamento.

📌 **DICA:** consulte o Instrutor para orientações adicionais. A resolução do estudo dirigido encontra-se no final deste Tutorial.

## 3. Configuração da Hadron Xtorm



### Configuração da UCP



A UCP da Série Hadron Xtorm é configurada e programada através do software MasterTool Xtorm. A configuração realizada define o comportamento e modos de utilização dos periféricos e características especiais da UCP. A programação representa a aplicação desenvolvida pelo usuário, também chamada de aplicativo.

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

Além destes parâmetros, é possível alterar o nome e a descrição de cada módulo inserido na aplicação, para isto, clique com o botão direito sobre o módulo, no item “Propriedades”, na guia “Comum”, altere o nome ou a descrição sendo ambos limitados a 255 caracteres.

#### Parâmetros Gerais

Os parâmetros gerais (General Parameters) podem ser ajustados na tela inicial de configuração da UCP conforme mostrado na figura a seguir.

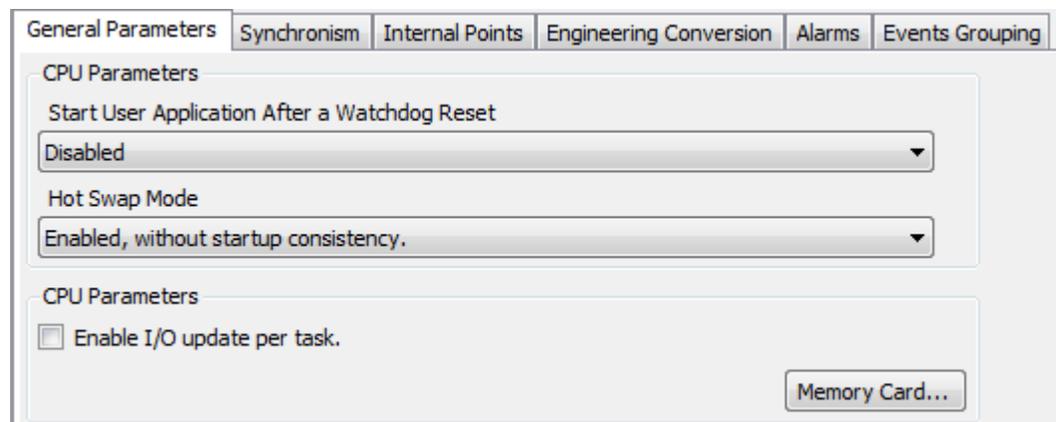


Figura 3-1. Tela de Configuração Parâmetros Gerais da UCP

Os parâmetros ajustáveis estão descritos na sequência.

- **Start User Application After a Watchdog Reset:** quando habilitado, inicia a aplicação do usuário após o reset do cão-de-guarda de hardware ou pela reinicialização do Runtime, porém mantêm a indicação através do LED de diagnóstico (botão) e através das variáveis.
- **Hot Swap Mode:** seleção do modo de troca a quente dos módulos.
- **Enable I/O update per task:** quando habilitado, realiza atualização de entradas e saídas no contexto da tarefa em que estiverem sendo utilizados.

#### Configuração da Troca a Quente

A UCP da Série Hadron Xtorm apresenta a possibilidade de troca dos módulos de E/S do barramento sem a necessidade de desligamento do sistema e sem perda de informações. Esta característica é conhecida como troca a quente. Na troca a quente, o comportamento do sistema relacionado se modifica conforme a configuração definida pelo usuário. Assim, o usuário pode escolher o comportamento que o sistema deverá apresentar em situações anormais de barramento e quando a UCP estiver em Modo Run. Na sequência, estão descritas as opções de configuração para essa funcionalidade.

- **Disabled for declared modules only:** nesta configuração, quando ocorre uma situação anormal de barramento, a UCP entra em Modo Stop em um tempo de até 2 segundos, quando o LED vermelho começa a piscar 4x. Para que a UCP volte ao estado normal Run, além de desfazer o que causou a situação anormal, é necessário executar um Reset de qualquer tipo (isso pode ser feito pelo MasterTool Xtorm no menu Comunicação). Esta opção não está disponível para projetos com redundância de UCP. Para projetos sem redundância de UCP, a utilização da troca a quente desabilitada também não é indicada quando se utiliza fontes de alimentação redundantes.
- **Enabled, with startup consistency for declared modules only:** é considerada “partida” o intervalo entre a energização da UCP (ou comando de reset ou download de aplicação) até a primeira vez em que ela entra em modo Run. Esta configuração verifica se ocorreu alguma situação anormal de barramento durante a partida; em caso positivo, a UCP entra em Modo Stop e o LED DL começa a piscar 4x. Posteriormente, para que a UCP possa ser colocada em modo Run, além de corrigir o que ocasionou a situação anormal, é necessário executar um comando de Reset de qualquer tipo, que pode ser feito pelo MasterTool Xtorm (menu Comunicação). Após a partida, se algum módulo apresentar alguma das situações anormais previstas, o sistema continuará trabalhando normalmente e sinalizará o problema via diagnóstico. Nesta configuração, quando ocorrer falta de alimentação (mesmo que temporária), comando Reset a Quente, comando Reset a Frio ou ter sido realizado o Download de uma nova aplicação, e algum módulo estiver em uma situação anormal de barramento; a UCP entrará em Modo Stop e o LED DL começa a piscar 4x, pois estas são consideradas situações de partida. Esta é a opção mais recomendada, pois garante a integridade do sistema na sua inicialização e permite a troca de módulos com o sistema funcionando.
- **Enabled, without startup consistency:** permite que o sistema entre em operação mesmo quando algum módulo estiver em uma situação anormal de barramento. As situações anormais são relatadas via diagnóstico, tanto durante, como após a partida. Esta opção é recomendada para a fase de implantação do sistema, pois permite que cargas de novas aplicações e o desligamento da alimentação sejam feitos sem a presença de todos os módulos configurados.

O procedimento de troca a quente está detalhado na documentação da série, incluindo as orientações, a caracterização das situações anormais de barramento, os cuidados necessários, a relação entre as condições e a troca a quente etc.

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

No caso de troca a quente de fontes de alimentação, é conveniente que a alimentação externa do módulo seja desligada antes de remover o módulo do barramento, a fim de reduzir a geração de arcos no conector do módulo. Para a inserção do módulo fonte de alimentação no barramento, é recomendado que primeiro seja feito o encaixe mecânico do módulo no barramento e em seguida a conexão do conector da alimentação externa ao módulo.

É importante salientar que, nos casos em que a UCP entra em Modo Stop e o LED DL começa a piscar 4x, devido a alguma situação anormal de barramento; os módulos de saída colocam seus pontos em modo seguro de falha, ou seja, desligam o ponto (0 Vdc).

No caso dos módulos de entrada digitais, caso ele seja removido do barramento energizado, o estado lógico dos pontos permanecerá no último valor. Caso o(s) conector(es) sejam removidos, o estado lógico dos pontos é colocado em estado seguro, ou seja, zero.

No caso dos módulos de saída digitais, caso ele seja removido do barramento energizado, o estado lógico dos pontos não é alterado. Porém, o ponto físico será desenergizado. Caso o(s) conector(es) sejam removidos, o estado lógico dos pontos não será alterado. Porém, o ponto físico será desenergizado.

Para os módulos de entradas analógicas RTD, caso ele seja removido do barramento energizado, o estado lógico dos pontos permanecerá no último valor. Caso o(s) conector(es) sejam removidos, o estado lógico dos pontos irá para o valor de fundo de escala configurado para cada ponto, visto que a impedância de entrada de cada canal sem o conector, tenderá para o infinito.

Para os módulos de entradas analógicas tensão/corrente, caso ele seja removido do barramento energizado, o estado lógico dos pontos permanecerá no último valor. Caso o(s) conector(es) sejam removidos, o estado lógico dos pontos dependerá do tipo de configuração de cada ponto, que vai depender do tipo de entrada selecionado e dos valores configurados para escala de engenharia. Para a escala de corrente 4-20 mA o estado lógico do ponto dependerá do valor configurado no campo “Valor de Circuito Aberto”.

#### **Configuração da Cartão de Memória**

Os parâmetros encontrados na tela de configuração da cartão de memória são definidos conforme indicado a seguir.

- **Copy Project from CPU to Memory Card:** copia o projeto da memória interna da UCP para o cartão de memória.
- **Password to Copy Project from CPU to Memory Card:** senha para copiar o projeto da memória interna da UCP para o cartão de memória.
- **Copy Project from Memory Card to CPU:** copia o projeto do cartão de memória para a memória interna da UCP.
- **Password to Copy Project from Memory Card to CPU:** senha para copiar o projeto do cartão de memória para a memória interna da UCP.

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

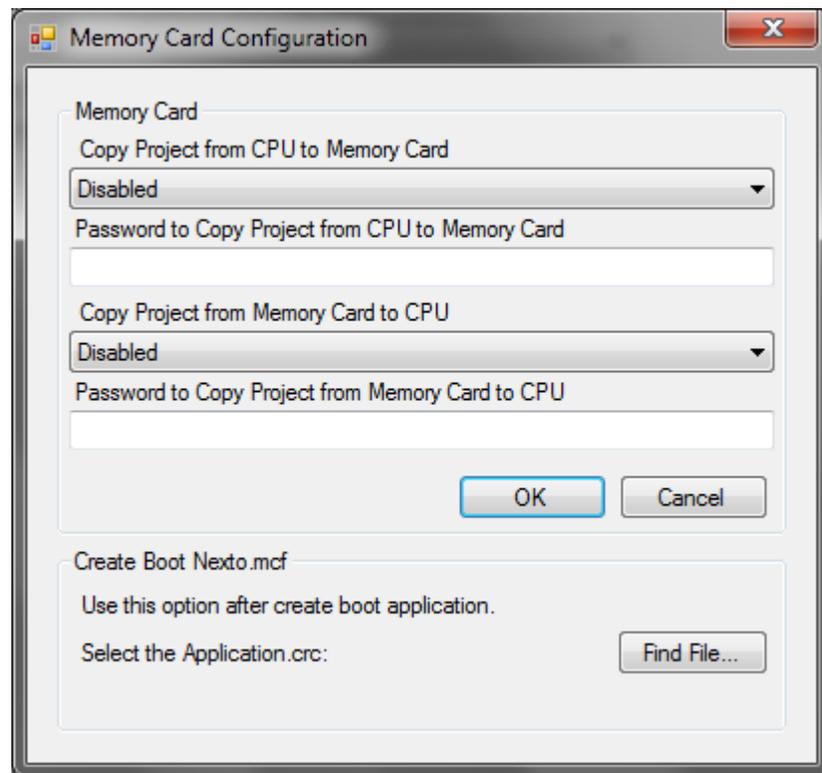


Figura 3-2. Configurações do Cartão de Memória

### Sincronismo de Tempo

Para a sincronização de tempo, a UCP HX3040 utiliza o protocolo IRIG-B, SNTP (Simple Network Time Protocol) ou centros de controle (servidores DNP3 ou IEC 60870-5-104), sendo que para cada um deles, o usuário terá a opção de habilitar/desabilitar o protocolo, como também escolher a prioridade de recebimento do sinal de sincronismo dentre cada um dos protocolos.

Para a utilização de protocolos de sincronismo de tempo, o usuário deve configurar os seguintes parâmetros na aba de Synchronism (Sincronismo), o qual é acessado via UCP HX3040 na árvore de dispositivos conforme mostrado a seguir.

The screenshot shows the configuration window for the HX3040 device, specifically the 'Synchronism' tab. The window has a title bar with 'Configuration (Bus)', 'NET 1', and 'HX3040'. Below the title bar are tabs for 'General Parameters', 'Synchronism', 'Internal Points', 'Engineering Conversion', 'Alarms', and 'Events Grouping'. The 'Synchronism' tab is active, displaying several configuration sections:

- General Configuration:** Time Zone is set to -03 : 00.
- IRIG-B Configuration:** 'Enable IRIG-B' is unchecked. Priority is set to 1. Time-out (s) is set to 3.
- Control Center Configuration:** Priority is set to 2.
- SNTP Configuration:** 'Enable SNTP' is unchecked. Priority is set to 3. Ethernet Interface is set to NET 1. Period for SNTP Synchronization (s) is 60. Minimum Error Before Clock Update (ms) is 200. Maximum Number of Retries is 3. IP Address of First SNTP Server is 192 . 168 . 15 . 10. IP Address of Second SNTP Server is 192 . 168 . 15 . 11.

Figura 3-3. Configuração de Sincronismo de Tempo

As configurações associadas à essa funcionalidade são: prioridade da fonte de sincronismo, Habilitar e Prioridade SNTP, interface Ethernet, período para sincronização SNTP, mínimo erro antes da atualização do relógio, número máximo de tentativas, endereço IP do primeiro e segundo servidores SNTP.

Funcionalidades associadas ao sincronismo de tempo:

- **IRIG-B** - O sincronismo de tempo através da interface IRIG-B permite o acerto do horário da UTR com erro máximo na ordem de poucos microssegundos, sendo recomendado para aplicações que necessitam de alta precisão na estampa de tempo dos eventos (1ms ou menos). O horário é enviado pelo dispositivo gerador (tipicamente um receptor de GPS) periodicamente a cada 1 segundo. O horário da UTR é atualizado imediatamente após a recepção e já passa a ser utilizado para a estampa de tempo dos próximos eventos que ocorrerem. Contudo, o RTC interno (que mantém o horário quando a UTR é desligada) é atualizado somente a cada 10 segundos.
- **Control Center** - No caso do sincronismo através do centro de controle, o usuário deve habilitar o sincronismo de tempo na tela de configuração do protocolo de comunicação que se deseja utilizar (DNP3 Servidor, IEC 60870-5-104 Servidor) para receber a sincronização de relógio. Este método de sincronismo deve ser utilizado apenas como um método auxiliar de sincronismo, uma vez que a precisão do processo de sincronização do relógio da UTR depende muito do atraso e tráfego de rede, bem como da carga de processamento da UCP, uma vez que este mecanismo é tratado por uma tarefa de baixa prioridade.

- **SNTP** - Para o sincronismo via SNTP a UCP irá se comportar como um cliente SNTP, ou seja, enviará requisições de sincronização de tempo para um servidor SNTP/NTP, que pode estar na rede local ou na internet. A precisão do sincronismo de tempo por SNTP depende das características da rede Ethernet onde está sendo aplicado. Caso o SNTP Server esteja na mesma rede local em que está a UTR, em uma situação ideal é possível obter uma precisão de 1ms. Contudo, quando o SNTP Server estiver em uma rede remota (internet etc.) este número pode chegar a dezenas de milissegundos. Ou seja, quando uma sincronização for efetuada, o horário atualizado no cliente pode estar alguns milissegundos adiantado ou atrasado em relação ao servidor. A UCP HX3040 envia as requisições de sincronização cíclicas, de acordo com o tempo configurado no campo Período de Sincronização do SNTP. Na primeira tentativa de sincronização, logo após a inicialização do serviço, a requisição é para o primeiro servidor, configurado em Endereço de IP do 1º Servidor. Caso este não responda, as requisições são direcionadas para o segundo servidor configurado em Endereço de IP do 2º Servidor, fornecendo uma redundância de servidores SNTP. Caso o segundo servidor também não responda, o mesmo processo de tentativa de sincronização é executado novamente, mas apenas após o Período de Sincronização ter passado. Ou seja, a cada período de sincronização, a UCP tenta se conectar uma vez em cada servidor, ela tenta o segundo caso o primeiro não responda. O tempo de espera por uma resposta do servidor SNTP é definido por padrão em 5 segundos e não pode ser modificado. Caso, após uma requisição de sincronização, a diferença entre o horário atual da UCP HX3040 e o recebido pelo servidor for maior que o valor configurado no parâmetro Erro Mínimo Antes da Atualização do Relógio, o horário da UCP HX3040 é atualizado. Como o tempo de espera pela resposta do servidor SNTP é de 5 s, deve-se prestar atenção ao configurar valores menores do que 10 s para o Período de Sincronização. Caso o servidor primário não responda, o tempo para a sincronização irá ser de, no mínimo, 5 s (espera pela resposta do servidor primário e tentativa de sincronização com o servidor secundário). Caso nem o servidor primário nem o secundário respondam, o tempo para a sincronização irá ser de, no mínimo, 10 s (espera pela resposta dos dois servidores e nova tentativa de conexão com o 1º servidor).

#### Configuração do Horário de Verão

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

#### Pontos Internos

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

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

#### ¶ NOTA:

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

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

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

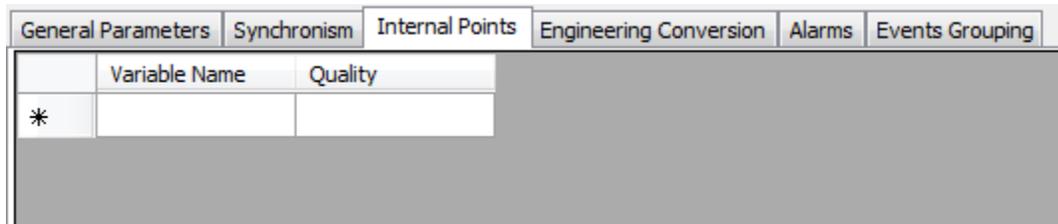


Figura 3-4. Tela de Configuração dos Pontos Internos

Configuração dos pontos internos:

- **Variable Name** - Variável simbólica que armazena o valor do ponto interno.
- **Quality** - Variável simbólica que armazena a qualidade do ponto interno representado em "Nome da Variável".

A figura a seguir mostra um exemplo de configuração de um ponto interno.

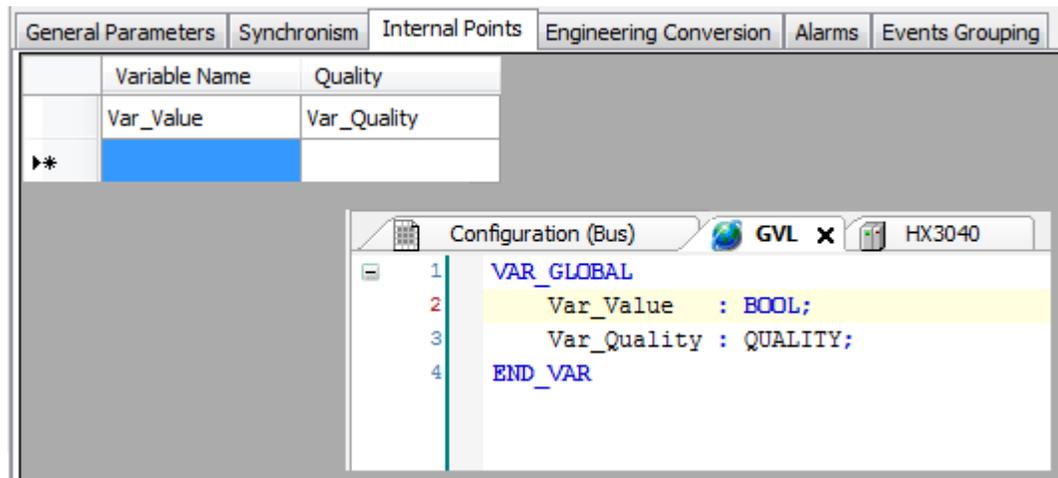


Figura 3-5. Exemplo de Configuração de um Ponto Interno

### Conversões de Qualidade

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

As normas IEC 61850, DNP3 e IEC 60870-5-104 possuem os seus próprios formatos para representação da informação de qualidade de um ponto. A Série Hadron Xtorm, por sua vez, possui um formato de qualidade próprio (mas muito similar ao IEC 61850) chamado de Internal Quality. Este formato é definido pelo tipo QUALITY (biblioteca LibRtuStandard) e é utilizado internamente para armazenamento da qualidade, permitindo que sejam realizadas conversões entre protocolos sem que haja perda de informação.

Quando se realiza o mapeamento de um mesmo ponto de comunicação entre dois drivers, a conversão de qualidade é realizada automaticamente em dois estágios. Por exemplo: caso um ponto de comunicação seja mapeado de um driver DNP3 Cliente para um driver IEC 61850 Servidor, primeiro a qualidade será convertida do formato DNP3 para o formato interno (e

armazenada internamente na UCP), e depois será convertida do formato interno para o formato IEC 61850.



### Pesquisa na Documentação: pontos internos (conversões dos formatos proprietários dos protocolos para o formato interno)

Consulte, na documentação do produto, as conversões dos formatos proprietários dos protocolos para o formato interno incluindo as opções Internal Quality, Conversão DNP3, Conversão IEC 60870-5-104, Conversão IEC 61850 e Qualidade Interna MODBUS .

❗ **DICA:** caso seja necessário consultar a conversão entre os protocolos, é necessário realizar a análise em dois estágios consultando cada uma das tabelas para o formato interno e depois realizando a correlação entre elas.

Fonte: Manual de Utilização Hadron Xtorm – MU223000!

### Qualidade dos Módulos de E/S

Para auxiliar no uso dos diagnósticos de cada ponto de E/S, a Série Hadron Xtorm cria automaticamente uma estrutura de qualidade para cada módulo utilizado no projeto da UTR, através de uma estrutura interna própria acessível via structure QUALITY, disponível na GVL IOQualities criada automaticamente pelo template do HD8500.

### Conversão de Engenharia

A aba de Engineering Conversion tem por objetivo facilitar a conversão linear de escalas. É comum que determinado sinal de entrada ou saída precise ser convertido de uma escala que se refere aos limites de um sensor, por exemplo, para uma escala diferente que tenha representatividade no contexto em que esse sinal é interpretado. Com este objetivo o usuário de um sistema de controle normalmente se depara com a demanda de converter unidades de engenharia. Essa conversão geralmente está associada à relação entre pontos físicos e pontos internos.

A configuração das Conversões de Engenharia, visualizada na figura a seguir, segue os parâmetros descritos na sequência. É possível a configuração de até 5120 entradas na tabela de Conversão de Engenharia.

	General Parameters	Synchronism	Internal Points	Engineering Conversion	Alarms	Events Grouping
	Input	Engineering	Minimum Input Value	Maximum Input Value	Minimum Engineering Value	Maximum Engineering Value
*						

Figura 3-6. Tela de Configuração dos Parâmetros de Conversão de Engenharia

Configuração dos Parâmetros de Conversão de Engenharia:

- **Input** - Variável simbólica de entrada da conversão.
- **Engineering** - Variável simbólica que armazenará o resultado da conversão.
- **Minimum Input Value** - Menor valor da entrada.
- **Maximum Input Value** - Maior valor de entrada.
- **Minimum Engineering Value** - Menor valor de engenharia.
- **Maximum Engineering Value** - Maior valor de engenharia.

## Alarmes

Alarmes são variáveis que armazenam um valor binário, que podem ser do tipo verdadeiro ou falso, sendo este valor o resultado da comparação de uma variável de entrada com um dado setpoint (valor de referência). O valor de um alarme geralmente é utilizado para disparar eventos de informação de estados da aplicação ou processo.

A configuração dos Alarmes, visualizada na figura a seguir, segue os parâmetros descritos na sequência. É possível a configuração de até 5120 entradas na tabela de Alarmes.

General Parameters	Synchronism	Internal Points	Engineering Conversion	Alarms	Events Grouping
	Input	Type	Setpoint	Alarm	
*		▼			

Figura 3-7. Tela de Configuração dos Alarmes

Configuração dos Alarmes:

- **Input** - Variável simbólica que possui a informação utilizada para geração do alarme.
- **Type** - Tipo de comparação que será realizada entre a entrada e o setpoint.
- **Setpoint** - Valor de referência utilizado para identificar se o valor da entrada gerou ou não um alarme.
- **Alarm** - Variável simbólica que receberá o valor VERDADEIRO ou FALSO, dependendo do resultado da comparação entre "Entrada" e "Setpoint".

## Agrupamento de Eventos

O agrupamento de eventos permite resumir um grupo de eventos em um único e novo evento, usando para isso um novo ponto digital. Um agrupamento de eventos é definido por um conjunto de pontos de entrada digital, por uma variável que armazena o resultado do agrupamento e pelo tempo de atraso máximo para a geração do evento.

O funcionamento do agrupamento de eventos consiste em calcular o valor e a qualidade da variável resumo através de uma lógica utilizando os pontos de entrada do agrupamento. Esta lógica é realizada a partir dos eventos ocorridos para as entradas, os quais são armazenados em uma fila interna dedicada para esta funcionalidade. A estampa de tempo da variável de resultado será determinada em função da estampa de tempo das entradas.

Para eventos relacionados à mudança de valor, é realizada uma lógica do tipo "OU". Desta forma o evento da transição "0" para "1" da variável resultado é datado pelo evento de "0" para "1" mais antigo das entradas do grupo, enquanto o evento da transição "1" para "0" da variável resultado é datado pelo evento de "1" para "0" mais recente das entradas. Para eventos relacionados à mudança da qualidade, é realizada uma lógica do tipo "E", isto é, basta que apenas uma entrada mude a qualidade para um valor diferente de GOOD para que a qualidade do resultado mude. Desta forma, o evento de qualidade da variável resultado é datado sempre pelo evento de qualidade mais antigo.

Após a ocorrência de um ou mais eventos que resultem em uma alteração do valor ou qualidade da variável de resultado, é iniciada a contagem de um tempo denominado tempo de atraso máximo. Ao final desta janela de tempo, é realizado o cálculo do valor e qualidade da variável resultado através da avaliação (em ordem cronológica) de todos os eventos previamente armazenados na fila interna dedicada para esta funcionalidade. Caso um evento que tenha ocorrido durante a janela de tempo definido pelo tempo de atraso máximo chegue após este tempo ter expirado, é gerado um alarme de atraso excessivo. Este alarme indica que o resultado do agrupamento pode não ser válido, pois um ou mais eventos não foram utilizados no cálculo por terem chegado atrasados.

A fila utilizada para armazenar os eventos relacionados às entradas possui capacidade para armazenar 8192 eventos, o que é suficiente para garantir o correto funcionamento em condições normais de operação. Caso seja gerada uma grande quantidade de eventos durante o tempo de atraso máximo de forma a encher esta fila, os novos eventos serão descartados (isto é, mantém os mais antigos).

Se um agrupamento de pontos for utilizado como entrada para outro agrupamento, é necessário se certificar de que o parâmetro "Atraso Máximo" do primeiro agrupamento seja menor do que o do grupo no qual ele está sendo inserido.

A figura a seguir demonstra o funcionamento do agrupamento de eventos. Neste exemplo, o resultado do agrupamento é armazenado na variável R, enquanto as entradas do grupo são compostas pelas variáveis P1, P2 e P3.

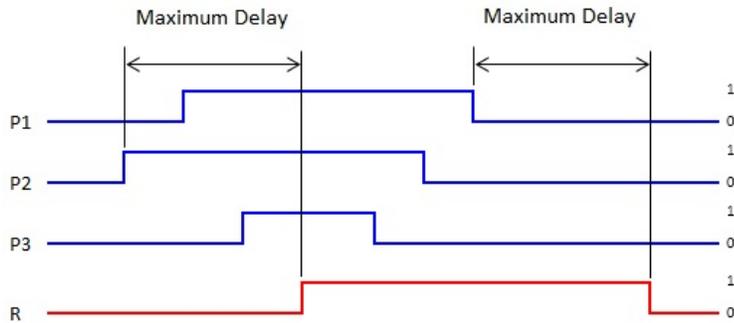


Figura 3-8. Funcionamento do Agrupamento de Eventos

A configuração do agrupamento de eventos, visualizada na figura a seguir permite a configuração de até 128 entradas na tabela de Agrupamento de Eventos. Os parâmetros configuráveis estão descritos na sequência.

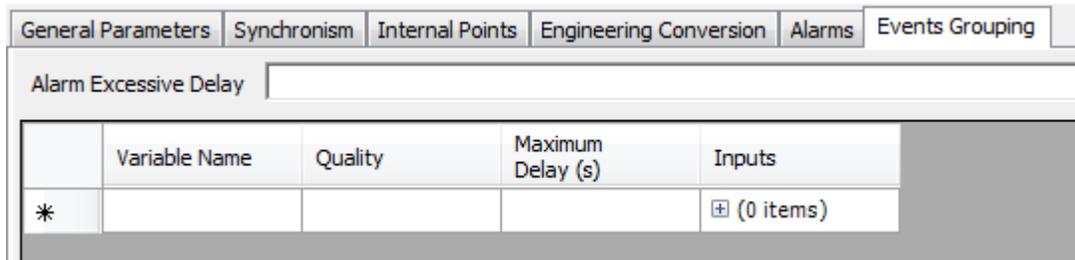


Figura 3-9. Tela de Configuração do Agrupamento de Eventos

Configuração do Agrupamento de Eventos:

- **Alarm Excessive Delay** - Nome da variável digital que indica se um evento chegou tarde demais e foi descartado na composição do resumo de eventos.
- **Variable Name** - Armazena o resultado do agrupamento de eventos.
- **Quality** - Armazena a qualidade do agrupamento de eventos em uma variável.
- **Maximum Delay (s)** - Tempo em segundos a partir da chegada do primeiro evento que será aguardado para que os demais eventos do grupo cheguem.
- **Inputs** - Configuração dos pontos digitais que compõem o grupo de eventos monitorados. Cada linha corresponde a um ponto digital que será monitorado. Um ponto digital só pode estar em um grupo.

# Configuração das Interfaces Seriais



## COM 1

A interface de comunicação COM 1 é composta por um conector tipo DB9 fêmea para o padrão RS-232C e RS-485 (não isolada). Permite a comunicação ponto a ponto (ou em rede utilizando conversor) nos protocolos abertos, MODBUS RTU escravo ou MODBUS RTU mestre.

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

Os parâmetros associados a essa configuração estão indicados a seguir.

- **Tipo Serial** - Configuração do tipo do canal serial.
- **Taxa de Transmissão** - Velocidade da porta de comunicação serial.
- **Paridade** - Configura a paridade da porta serial.
- **Bits de Dados** - Configura o número de bits de dados em cada caractere da comunicação serial.
- **Bits de Parada** - Configura os bits de parada da porta serial.
- **Modo Serial** - Configura o modo de operação da porta serial. O Modo Normal é o padrão. Já no Modo Estendido de operação da comunicação serial são fornecidas informações sobre o frame de dados recebido. Consulte a documentação da série para detalhamento nessa parametrização.

## Configurações Avançadas

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

Parâmetros de Porta Avançados:

- **Sinais de Modem** - Realiza o controle de requisição para transmissão de um comando, através da interface RS-232C.
- **Threshold de RX da UART** - Quantidade de bytes que devem ser recebidos para gerar uma nova interrupção na UART. Valores baixos fazem o TIMESTAMP mais preciso quando o Modo Estendido é utilizado e minimiza os erros de OVERRUN. No entanto, valores baixos podem causar muitas interrupções que podem retardar a UCP.

Eventos Seriais:

- **RX em TX** - Se habilitado todos os bytes recebidos durante a transmissão serão descartados em vez de ir para a fila de RX. Utilizado para desabilitar a operação full-duplex na interface RS-232C.
- **Evento RX DCD** - Se habilitado gera um evento externo devido à mudança do sinal DCD.
- **Evento RX CTS** - Se habilitado gera um evento externo devido à mudança do sinal CTS.

## COM 2

A interface de comunicação COM 2 é composta por um conector tipo DB9 fêmea para os padrões RS-422C e RS-485. Permite a comunicação ponto a ponto ou em rede nos protocolos abertos, MODBUS RTU escravo ou MODBUS RTU mestre.

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

Os parâmetros associados a essa configuração são similares aos indicados na COM 1.

No tocante às configurações avançadas, é possível ajustar somente o Threshold de RX da UART de forma análoga à COM 1.

# Configuração das Interfaces Ethernet



A UCP Hadron Xtorm possui seis interfaces Ethernet locais denominadas NET 1 à NET 6, as quais podem operar de forma independente ou sob a forma de pares. Cada interface ( ou cada par) deve ser configurada em uma subrede distinta.

## Interfaces Ethernet Locais NET 1 à NET 6

As interfaces NET 1 à NET 6 são compostas por um conector tipo RJ45 de comunicação no padrão 10/100Base-TX. Permitem a comunicação ponto a ponto ou em rede nos protocolos suportados. A NET 1 está sempre habilitada, e é através dela (ou do par NET 1+2) que é realizada a comunicação com a ferramenta MasterTool Xtorm. Já as interfaces NET 2 à NET6 podem ser habilitadas/desabilitadas através da configuração. Os parâmetros configuráveis estão indicados na sequência.

- **Habilitar** - Habilita a interface Ethernet (somente para NET 2 à NET 6).
- **Endereço IP** - Endereço IP do Controlador no barramento Ethernet.
- **Máscara de Subrede** - Máscara de subrede do Controlador no barramento Ethernet.
- **Endereço do Gateway** - Endereço do Gateway do Controlador no barramento Ethernet.

### NOTAS:

1. Não é possível configurar duas ou mais interfaces Ethernet de uma UCP na mesma subrede, sendo este tipo de configuração bloqueada pela ferramenta MasterTool Xtorm. Desta forma, cada interface Ethernet deve ser configurada em uma subrede distinta.
2. As seguintes portas TCP das interfaces Ethernet, tanto locais quanto remotas, são utilizadas por serviços da UCP e, portanto, são reservadas e não devem ser utilizadas pelo usuário: 80, 8080, 1217, 1740, 1741, 1742,1743 e 11740.

## Configurações Avançadas das Interfaces Ethernet

Os canais Ethernet da UCP HX3040 podem ser configurados em três modos distintos de funcionamento: de forma individual, ou organizados em pares NIC Teaming ou modo Switch. Os pares possíveis são NET 1+2, NET 3+4 e NET 5+6, sendo que a configuração do par fica armazenada sempre na NET ímpar. Desta forma, quando configuradas como parte de um par, os campos de configuração das NETs 2, 4 e 6 ficam desabilitados. As configurações avançadas das interfaces Ethernet, visualizadas na figura a seguir, seguem os parâmetros descritos na sequência.

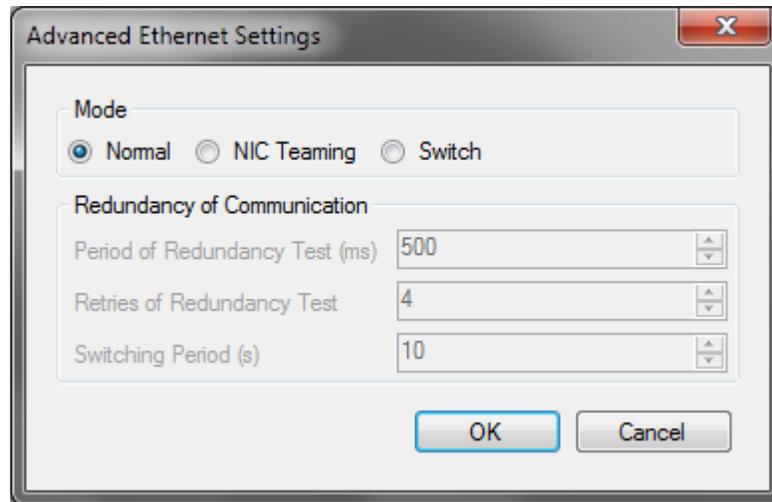


Figura 3-10. Tela de Configurações Avançadas das Interfaces Ethernet

Parâmetros de configurações avançadas das interfaces Ethernet:

- **Modo Normal** - Neste modo, a interface opera como uma porta Ethernet independente, não havendo qualquer relação com a interface subsequente.
- **Modo NIC Teaming** - Neste modo, a interface forma um par redundante com a interface subsequente, operando em um esquema do tipo ativa/reserva. Um par NIC Teaming possui um único endereço IP, associado à interface que estiver ativa no momento. Desta forma, um cliente como um SCADA ou MasterTool Xtorm, conectado a um servidor na UCP, não precisa se preocupar em trocar o endereço IP caso haja falha em alguma das portas do par. Além disso, cada uma das interfaces que formam um par NIC Teaming possuem diagnósticos separados, com o intuito de facilitar a depuração de falhas que eventualmente possam surgir. Quando este modo for selecionado, na mesma tela - automaticamente - outros parâmetros serão habilitados e deverão ser configurados: Período de Teste da Redundância (ms), Número de Retentativas do Teste da Redundância e Período para Chaveamento (s). Caso o tempo de resposta do Teste da Redundância atinja Período de Teste vezes o Número de Retentativas e a interface ativa permaneça por um tempo maior que o Período para Chaveamento sem receber nenhum pacote, ocorrerá um SWITCHOVER, tornando ativa a interface que até então estava inativa. É importante ressaltar que há um DELAY entre a detecção da falha e ativação da interface inativa devido ao tempo necessário para sua configuração. Tal DELAY pode chegar a algumas dezenas de milissegundos. Quando uma das NETs estiver ativa, esta assumirá o endereço de IP configurado, permanecendo a NET inativa com seus parâmetros de Endereço de IP, Máscara de Subrede e Endereço do Gateway em branco nos diagnósticos da UCP.

¶ **NOTA:**

Quando um comando de Reset Origem for executado em uma UCP configurada com NIC Teaming nas interfaces frontais NETx, apenas a interface que estava ativa antes do comando continuará acessível. Após o comando, a interface acessível poderá ser visualizada no Menu Informativo e de Configuração da UCP.

- **Modo Switch** - Neste modo, a interface forma um par com a interface subsequente operando como um switch Ethernet, permitindo então a comunicação por ambas às portas. Desta forma, este modo permite a ligação em “cascata” de diversas UTRs Xtorm, possibilitando a implementação de uma topologia de rede em anel. Contudo, para realizar o fechamento deste anel, é obrigatório que as extremidades desta “cascata” sejam conectadas a um switch externo que possua suporte aos protocolos STP ou RSTP. A vantagem desta arquitetura é a redução do número de portas do switch externo quando comparada à topologia de rede em estrela. Contudo, a principal desvantagem está no

tempo para recuperação da rede em caso de uma falha (tempo de convergência), o qual pode ser de até 50 segundos utilizando STP ou de até 6 segundos utilizando RSTP. Desta forma, é importante avaliar se o sistema que está sendo desenvolvido pode conviver sem maiores problemas com estes tempos de interrupção da rede. Caso seja um sistema crítico que não possa conviver com interrupções tão longas (por exemplo, sistemas de controle ou proteção com tempos de resposta muito rápidos), deve ser utilizada a topologia de rede em estrela.

## Configuração dos Pontos Duplos



A representação de Pontos digitais duplos de entrada e saída é realizada através de um tipo de dados especial chamado DBP (definido na biblioteca LibDataTypes). Este tipo consiste basicamente em uma estrutura de dois elementos do tipo BOOL chamados OFF e ON (equivalentes a TRIP e CLOSE respectivamente). Variáveis deste tipo podem ser associadas à módulos de Entrada e Saída que possuem suporte a eventos e comandos pulsados (HX1100, HX2200, HX2300 e HX2320 por exemplo), onde cada um dos elementos OFF e ON deve ser mapeado individualmente para cada entrada/saída física do módulo. No caso dos módulos de entrada, a memória %I é atualizada normalmente mesmo para os pontos que estiverem configurados como DBP. Já no caso dos módulos de saída, os pontos configurados como DBP não consideram mais o valor presente na memória %Q.

Mais informações sobre a configuração de pontos duplos nos módulos de entrada e saída podem ser obtidas nos seus respectivos documentos de Características Técnicas.

# Configuração dos Protocolos



A documentação da série informa os limites máximos de configuração relacionados às interfaces de comunicação e aos protocolos de comunicação da UCP, bem como os limites máximos de configuração e operação dos protocolos de comunicação e das interfaces Seriais e Ethernet.

## Pesquisa na Documentação: configuração de protocolos



Consulte, na documentação do produto, os limites dos protocolos por interface e por UCP, os limites máximos de configuração e operação dos protocolos de comunicação, das interfaces Seriais e Ethernet e o comportamento assumido para cada protocolo de comunicação existente na UCP HX3040 em todos seus estados de funcionamento.

Fonte: Manual de Utilização Hadron Xtorm – MU223000!

## Fila de Eventos da UCP

A UCP possui uma fila de eventos do tipo FIFO (First In, First Out) utilizada para armazenar temporariamente os eventos relacionados aos pontos de comunicação até que eles sejam transferidos ao seu destino. Todos os eventos de pontos de comunicação gerados na UTR são direcionados e armazenados na fila da UCP. Esta fila possui as seguintes características:

- Tamanho: 4500 eventos;
- Política de estouro: mantém os mais recentes.

Esta fila de eventos é armazenada em uma área de memória retentiva (não volátil) exclusiva para este fim, e não ocupa a área de variáveis simbólicas retentivas/persistentes. Desta forma, os eventos presentes na fila da UCP e que ainda não foram transmitidos ao centro de controle não serão perdidos em um eventual desligamento da UTR.

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

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

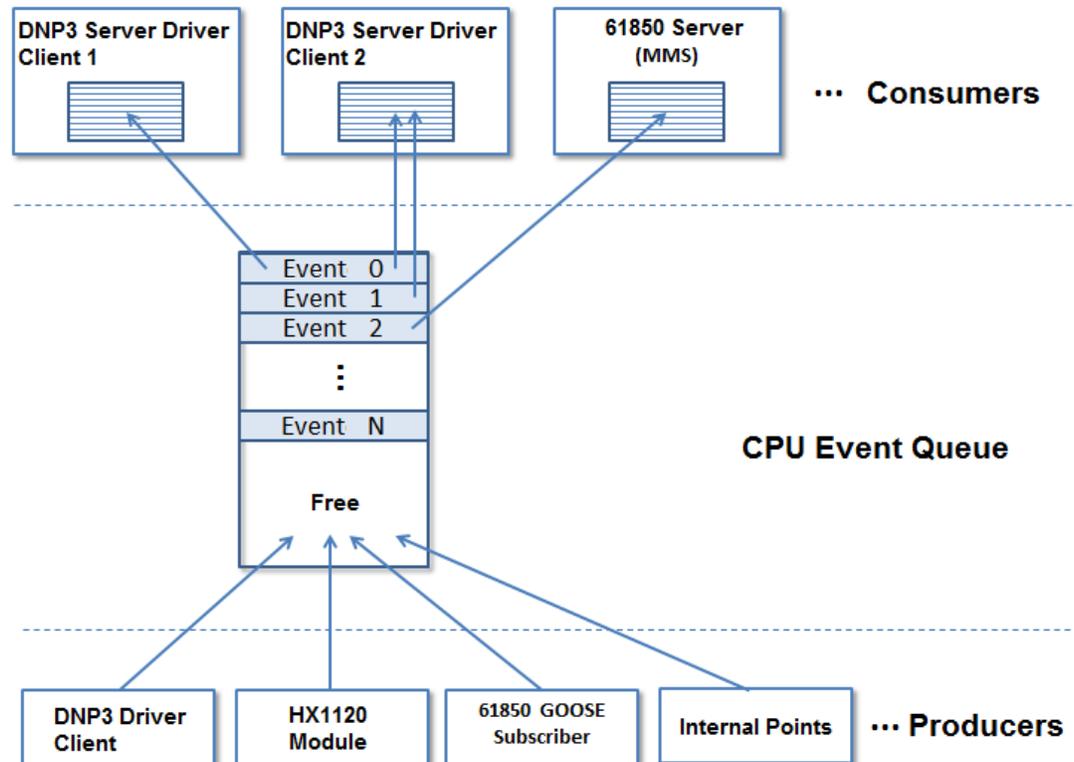


Figura 3-11. Fila de Eventos da UCP

### Consumers

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

### Princípios de Funcionamento da Fila

Uma vez armazenado na fila da UCP, cada evento é transmitido para o consumidor que possui este ponto de comunicação na sua base de dados. Na figura acima, o Evento 0 é referente a um ponto de comunicação mapeado para 2 centros de controle DNP3 (Cliente 1 e 2). Já o Evento 1 é referente a um ponto de comunicação mapeado apenas para um centro de controle DNP3 (Cliente 2). Por sua vez, o Evento 2 é referente a um ponto de comunicação mapeado para um centro de controle IEC 61850.

Os eventos permanecem armazenados na fila da UCP até que todos os seus consumidores confirmem a sua recepção. O critério utilizado para confirmar a recepção é específico de cada consumidor. No caso específico do DNP3 Servidor, a confirmação ocorre apenas quando o centro de controle (SCADA) confirmar a recepção da mensagem (o que oferece uma boa segurança contra perda de eventos em caso de falhas de rede). Já para outros tipos de consumidores (como é o caso dos BUFFERED REPORTS no Servidor IEC 61850), a confirmação ocorre quando o evento foi transferido para a fila interna do driver. A ocupação da fila de eventos da UCP pode ser monitorada através dos diagnósticos disponibilizados na área de diagnósticos da UCP através da variável `DG_HX3040.tDetailed.Events.*`, a qual fornece também informações sobre estouro da fila, entre outras.

#### Sinalização de Overflow

A sinalização de overflow para a fila de eventos da UCP ocorre em duas situações:

1. Se a fila de eventos da UCP já estiver parcialmente ocupada e o espaço disponível não for suficiente para armazenar os novos eventos ocorridos no ciclo de execução;
2. Se a UCP abortou a geração de eventos (porque ocorreram mais eventos em um único ciclo de execução do que o tamanho total da sua fila de eventos).

Já a sinalização de overflow para a fila de eventos dos consumidores ocorre em duas situações:

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

#### Producers

Os produtores são tipicamente drivers de comunicação ou elementos internos da UTR que são capazes de gerar eventos. A figura acima mostra alguns exemplos:

- **Módulo HX1100 e HX1120** - Os eventos são gerados e armazenados internamente no módulo, e então são lidos pelo barramento local da UTR e inseridos na fila da UCP.
- **Driver DNP3 Cliente** - Os eventos gerados e armazenados nos OUTSTATIONS, e então são lidos pelo driver e inseridos na fila da UCP.
- **Driver IEC 60870-5-104 Cliente** - Os eventos gerados e armazenados nos CONTROLLED STATIONS, e então são lidos pelo driver e inseridos na fila da UCP.
- **Driver IEC 61850 GOOSE Subscriber** - Os eventos são gerados no momento da recepção de uma mensagem GOOSE (desde que os dados recebidos causem uma alteração de valor em alguma variável), e então são inseridos na fila da UCP.
- **Pontos Internos** - Este é um elemento interno no firmware da UCP, o qual realiza a detecção de eventos a cada ciclo de execução (MainTask) para aqueles pontos de comunicação que não possuem uma origem definida e então insere os eventos na fila da UCP. O número máximo de eventos que podem ser detectados a cada ciclo da MainTask é igual ao tamanho da fila de eventos da UCP. Caso seja gerado um número de eventos maior que este em um mesmo ciclo da MainTask, os eventos excedentes serão perdidos.
- **Driver MODBUS (Cliente/Servidor/Mestre/Escravo)** - As alterações de valor em variáveis causadas por leituras/escritas MODBUS são detectadas a cada ciclo da MainTask e então os eventos são inseridos na fila da UCP. No caso dos drivers Cliente/Mestre, são gerados também eventos de qualidade quando houver falha de comunicação com o dispositivo escravo

#### 📌 NOTA:

Os módulos HX1100 e HX1120 armazenam internamente os eventos em uma estrutura que pode conter de 1 a 32 eventos. Quando apenas 1 entrada muda de estado, esta estrutura armazenará somente 1 evento. Agora, quando todas as 32 entradas mudarem de estado simultaneamente, esta estrutura armazenará 32 eventos. O módulo possui uma fila interna com capacidade para armazenar até 420 estruturas (isto é, de 420 a 13440 eventos). A UCP realiza, através do barramento local, a leitura de 38 estruturas a cada 3 ciclos de execução da UTR (MainTask).

#### Interceptação de Comandos Oriundos do Centro de Controle

A UTR Xtorm possui um bloco de função que permite que comandos de seleção e operação para pontos de saída recebidos por drivers servidores (DNP3, IEC 60870-5-104, IEC 61850, etc...)

sejam tratados pela lógica de usuário. Este recurso permite a implementação de intertravamentos, bem como a manipulação dos dados do comando recebido na lógica do usuário ou ainda o redirecionamento do comando para diferentes IEDs. A interceptação de comandos é implementada pelo Bloco Funcional CommandReceiver.



#### Pesquisa na Documentação: Bloco Funcional CommandReceiver

Consulte, na documentação do produto, a implementação do tratamento na lógica do usuário dos comandos de seleção e operação para pontos de saída recebidos por drivers servidores via Bloco Funcional CommandReceiver, incluindo os parâmetros de entrada e saída, os comandos suportados, os parâmetros que compõem as estruturas sSelectParameters, sOperateParameters e sCancelParameters.

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

Fonte: Manual de Utilização Hadron Xtorm – MU223000!



#### Estudo Dirigido 3-1: aplicação do interceptador de comandos

Utilizando a linguagem ST implemente um interceptador de comandos para encaminhar um comando recebido pelo protocolo Servidor (DNP3 ou IEC 60870-5-104) através da variável “dbpDoublePointServer”, para um ponto duplo de uma saída digital da Série Hadron Xtorm, em que a variável “dbpDoublePointIO” está mapeada.

##### 🔍 **DICAS:**

1. Para estimular o código pode ser utilizado o comando “C\_DC\_NA” em modo “Execute Only” e com o qualificador “Short pulse” do protocolo IEC 60870-5-104 Servidor, ou a função “Direct Operate” do tipo “Control Relay Output Block” e com código de controle “Trip” ou “Close” do protocolo DNP3 Servidor.
2. Utilize a função PulsedCommand, da biblioteca LibRtuStandard, detalhado no documento de características técnicas dos módulos de saídas digitais da Série Xtorm (consultar a CT do módulo HX2320).
3. Consulte o Instrutor para orientações adicionais.
4. A resolução do estudo dirigido encontra-se no final deste Tutorial.

#### MODBUS RTU Mestre

Este protocolo está disponível para a UCP da Série Hadron Xtorm nos seus canais seriais. Ao selecionar esta opção no MasterTool Xtorm, a UCP passa a ser mestre da comunicação MODBUS, possibilitando o acesso a outros dispositivos com o mesmo protocolo, quando esta estiver em modo de execução (Modo Run). A documentação da série indica os tipos de variáveis suportadas pela HX3040 (com as restrições associadas) para cada um dos tipos de dados do protocolo, aplicável para todos os dispositivos MODBUS.

Procedimento de configuração do protocolo:

1. Adicionar a instância do protocolo MODBUS RTU Mestre ao canal serial COM 1 ou COM 2 (ou ambos, em caso de duas redes de comunicação);
2. Seguir as etapas descritas anteriormente na seção “Configuração das Interfaces Seriais”;
3. Configurar a interface serial, escolhendo a velocidade de comunicação, o comportamento dos sinais RTS/CTS, a paridade, bits de parada do canal, entre outros através de um duplo clique sobre o canal serial COM 1 ou COM 2;
4. Configurar os parâmetros gerais do protocolo MODBUS Mestre, tais como: tempos de atraso de envio e interframe mínimo;
5. Adicionar e configurar dispositivos, definindo endereço do escravo, timeout de comunicação e número de tentativas de comunicação;
6. Adicionar e configurar os mapeamentos MODBUS, especificando o nome da variável, tipo de dados, endereço inicial do dado, tamanho do dado e variável que receberá os dados de qualidade
7. Adicionar e configurar as requisições MODBUS, especificando a função desejada, o tempo de varredura da requisição, o endereço inicial (leitura/escrita), o tamanho dos dados (Leitura/Escrita), a variável que receberá os dados de qualidade, e a variável responsável por desabilitar a requisição.

#### Parâmetros Gerais do Protocolo MODBUS Mestre

Os parâmetros gerais, encontrados na tela inicial de configuração do protocolo MODBUS (ver figura a seguir) estão indicados na sequência.

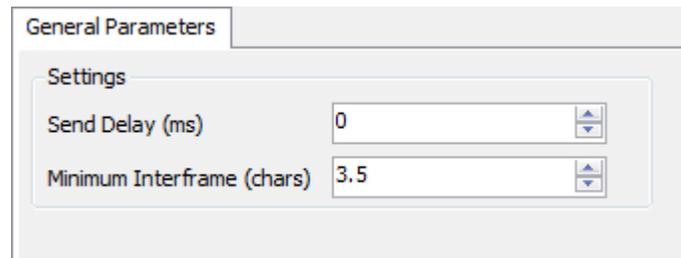


Figura 3-12. Tela de Configuração Parâmetros Gerais MODBUS RTU Mestre

Parâmetros associados:

- **Send Delay (ms)** - Tempo de atraso para envio da resposta. A resposta a uma requisição MODBUS pode causar problemas em certos momentos, como, por exemplo, na interface RS-485 ou outra half-duplex. Às vezes existe um atraso entre o tempo de resposta do escravo e o silêncio na linha física (atraso no escravo para colocar RTS em zero e colocar o transmissor RS-485 em alta impedância). Para resolver o problema, o mestre pode esperar o tempo determinado nesse campo antes de enviar a nova requisição. Caso contrário, os primeiros bytes transmitidos pelo mestre, durante a requisição, podem ser perdidos.
- **Minimum Interframe (chars)** - Tempo mínimo de silêncio entre diferentes frames. A norma MODBUS define esse tempo como 3,5 caracteres, porém esse parâmetro é configurável para atender aos dispositivos que não estão de acordo com o padrão.

#### NOTA:

É recomendável que o usuário faça uso das duas estruturas de diagnósticos disponíveis para maior confiabilidade e abrangência dos diagnósticos, caso esteja consistindo a comunicação do Mestre RTU, através dos diagnósticos e comandos armazenados em variáveis do tipo T\_DIAG\_MODBUS\_RTU\_MASTER\_1 e a estrutura de diagnósticos das requisições do tipo T\_DIAG\_MODBUS\_RTU\_MAPPING\_1.

### Configuração dos Dispositivos Escravos

A configuração dos dispositivos escravos está ilustrada na figura a seguir.

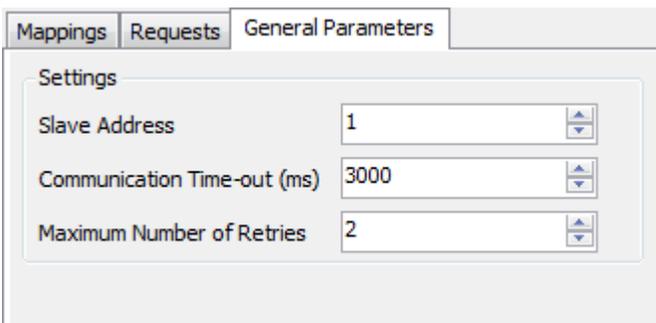


Figura 3-13. Tela de Configurações dos Parâmetros Gerais do Dispositivo

Parâmetros associados:

- **Slave Address** - Endereço do escravo MODBUS. De acordo com a Norma MODBUS, a faixa de endereços válidos para escravos é de 0 a 247, sendo os endereços 248 a 255 reservados. Quando o mestre envia um comando de escrita com o endereço configurado como zero, ele está realizando requisições broadcast na rede;
- **Communication Time-out (ms)** - Define o timeout do nível de aplicação. O timeout da comunicação é o tempo que o mestre aguardará por uma resposta do escravo à requisição. Para um dispositivo MODBUS RTU Mestre devem ser levadas em consideração ao menos as seguintes variáveis do sistema: o tempo que o escravo leva para transmitir o frame (de acordo com a taxa de transmissão), o tempo que o escravo leva para processar a requisição e o Delay de envio da resposta caso seja configurado no escravo, o tempo de transmissão da resposta, possíveis atrasos causados por dispositivos intermediários, como repetidores. É recomendado que o timeout seja igual ou maior que o tempo para transmitir o frame somado ao Delay de envio da resposta e a duas vezes o tempo de processamento da requisição. A seguinte fórmula deve ser adotada para cálculo estimado de Tempo de Transmissão (ms): Tempo de Transm. (ms) =  $[1000 \times (\text{contagem de caracteres}) \times (\text{bits por caractere})] / \text{Taxa de Transm.};$
- **Maximum Number of Retries** - Define o número de tentativas antes de reportar um erro de comunicação. Por exemplo, se o escravo não responder a um pedido e o mestre estiver configurado para enviar três retentativas, o número do contador de erros será incrementado por 1 unidade ao final da execução destas três retentativas.

### Configuração dos Mapeamentos

A configuração das relações MODBUS, visualizadas na figura a seguir, segue os parâmetros descritos na sequência.

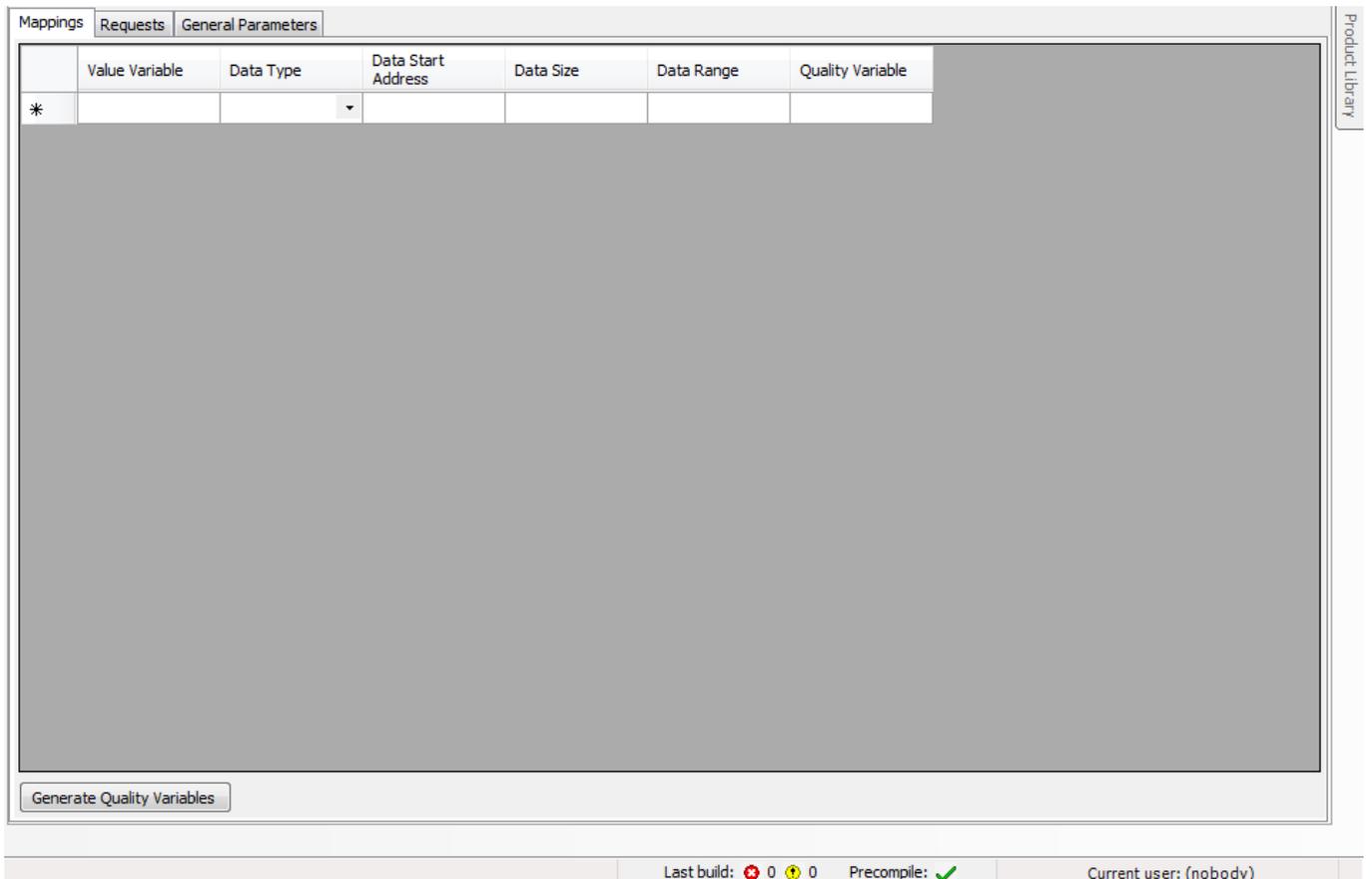


Figura 3-14. Tela de Mapeamentos de Dados MODBUS

Parâmetros associados:

- **Value Variable** - Nome da variável simbólica;
- **Data Type** - Tipo de dado MODBUS;
- **Data Start Address** - Endereço inicial dos dados MODBUS;
- **Data Size** - Tamanho do dado MODBUS. O valor de Tamanho especifica a quantidade máxima de dados que uma relação MODBUS poderá acessar, a partir do endereço inicial. Sendo assim, para ler uma faixa de endereços contínua, é necessário que todos os endereços estejam declarados em uma única relação. Este campo varia de acordo com o tipo de dado MODBUS configurado;
- **Data Range** - Faixa de endereços do dado configurado;
- **Quality Variable** - Nome da variável que vai conter os dados de qualidade do mapeamento MODBUS;
- **Generate Quality Variables** - Gera automaticamente na GVL IOQualities variáveis do tipo QUALITY para cada Variável de Valor declarada. Para mais informações acesse o capítulo Conversões de Qualidade.

### Configuração das Requisições

A configuração das requisições MODBUS, visualizadas na figura a seguir, segue os parâmetros descritos na sequência.

	Function Code	Polling (ms)	Read Data Start Address	Read Data Size	Read Data Range	Write Data Start Address	Write Data Size	Write Data Range	Diagnostic Variable
*									

Diagnostics Variable Type:

Last build: 0 0 Precompile: ✓ Current user: (nobody)

Figura 3-15. Tela de Requisições de Dados MODBUS Mestre

Parâmetros associados:

- **Function Code** - Tipo de função MODBUS. Consulte a documentação da série para informações relativas às funções MODBUS suportadas;
- **Polling (ms)** - Período de comunicação (ms). Este parâmetro indica com que frequência a comunicação definida por esta requisição deve ser executada. Ao ser finalizada uma comunicação será aguardado um tempo igual ao configurado no campo varredura e, após, será executada uma nova comunicação;
- **Read Data Start Address** - Endereço inicial dos dados de leitura MODBUS;
- **Read Data Size** - Tamanho dos dados de leitura MODBUS. O valor mínimo para o tamanho dos dados de leitura é 1 e o valor máximo depende da função MODBUS (FC) utilizada;
- **Read Data Range** - Faixa de endereço dos dados de leitura MODBUS. Este campo mostra a faixa de dados de leitura MODBUS configurada para cada requisição. O endereço inicial de leitura, somado ao tamanho do dado de leitura resultará na faixa de dados de leitura de cada uma das requisições;
- **Write Data Start Address** - Endereço inicial dos dados de escrita MODBUS;
- **Write Data Size** - Tamanho dos dados de escrita MODBUS. O valor mínimo para o tamanho dos dados de escrita é 1 e o valor máximo depende da função MODBUS (FC) utilizada;
- **Write Data Range** - Faixa de endereço dos dados de escrita MODBUS. Este campo mostra a faixa de dados de escrita MODBUS configurada para cada requisição. O

endereço inicial de escrita, somado ao tamanho do dado de escrita resultará na faixa de dados de escrita de cada uma das requisições;

- **Diagnostic Variable** - Nome da variável de diagnóstico. Os diagnósticos da requisição MODBUS configurada, são armazenados em variáveis do tipo T\_DIAG\_MODBUS\_RTU\_MAPPING\_1;
- **Disabling Variable** - Campo destinado a variável utilizada para desabilitar, individualmente, as requisições MODBUS configuradas no espaço Requisições. A requisição é desabilitada quando a variável, correspondente a requisição, for igual a 1, caso contrário, o a requisição está habilitada.

#### 🔔 **NOTA:**

Diferentemente de outras tarefas de uma aplicação, quando for atingida uma marca de depuração na MainTask, a tarefa de uma instância MODBUS RTU Mestre, e qualquer outra tarefa MODBUS, para de ser executada quando tentar efetuar uma escrita em uma área de memória. Isto ocorre para manter a consistência dos dados das áreas de memória enquanto a MainTask não estiver em execução.

## MODBUS RTU Escravo

Este protocolo está disponível para a UCP da Série Hadron Xtorm nos seus canais seriais. Ao selecionar esta opção no MasterTool Xtorm, a UCP passa a ser escravo da comunicação MODBUS, permitindo a conexão com dispositivos mestre MODBUS RTU. Este protocolo somente está disponível quando a UCP estiver em modo de execução (Modo Run).

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

1. Adicionar a instância do protocolo MODBUS RTU Escravo ao canal serial COM 1 ou COM 2 (ou ambos, em caso de duas redes de comunicação);
2. Realizar as etapas indicadas na seção “Inserindo uma Instância de Protocolo”;
3. Configurar a interface serial, escolhendo a velocidade de comunicação, o comportamento dos sinais RTS/CTS, a paridade, os bits de parada do canal, entre outros através de um duplo clique sobre o canal serial COM 1 ou COM 2;
4. Configurar os parâmetros gerais do protocolo MODBUS escravo, como: Endereço do Escravo e tempos de comunicação (disponível no botão de configurações avançadas do Escravo);
5. Adicionar e configurar as relações MODBUS, especificando o nome da variável, tipo de dado MODBUS, o endereço inicial do dado e o tamanho do dado.

### Parâmetros Gerais do Protocolo MODBUS Slave

Os parâmetros gerais, encontrados na tela inicial de configuração do protocolo MODBUS estão ilustrados na figura a seguir. O parâmetro-base a ser ajustado é o Slave Address, ou seja, o endereço do escravo MODBUS.



Figura 3-16. Tela de Configuração do Escravo

### Tempos de comunicação

Os tempos de comunicação do protocolo MODBUS Escravo, encontrados no botão “Advanced...” da tela de configuração, estão divididos em: Ciclo da Tarefa, Atraso do Envio e Interframe Mínimo, os quais são explicados na sequência.

- **Task Cycle (ms)** - Tempo para execução da instância dentro do ciclo, sem considerar o tempo de execução dela. O usuário deverá ter cuidado a alterar esse parâmetro, pois ele interfere diretamente no tempo de resposta, volume de dados por varredura e, principalmente, no balanceamento dos recursos da UCP entre comunicações e outras tarefas.
- **Send Delay (ms)** - Tempo de atraso para envio da resposta. Deve-se atentar para o fato de que a resposta à uma requisição MODBUS pode causar problemas em certos momentos, como, por exemplo, na interface RS-485 ou outra half-duplex. Às vezes existe um atraso entre o tempo da requisição do mestre e o silêncio na linha física (atraso no mestre para colocar RTS em zero e colocar o transmissor RS-485 em alta impedância). Para resolver o problema, o escravo pode esperar o tempo determinado nesse campo antes de enviar a resposta. Caso contrário, os primeiros bytes transmitidos pelo escravo, durante a resposta, podem ser perdidos.
- **Minimum Interframe (chars)** - Tempo mínimo de silêncio entre diferentes frames. A norma MODBUS define esse tempo como 3,5 caracteres, porém esse parâmetro é configurável para atender aos dispositivos que não estão de acordo com o padrão.

Os diagnósticos e comandos do protocolo MODBUS RTU Slave configurado, são armazenados em variáveis do tipo T\_DIAG\_MODBUS\_RTU\_SLAVE\_1. É importante ressaltar que todos os contadores dos diagnósticos do MODBUS RTU Escravo retornam à zero quando o valor limite 65535 é ultrapassado.

A configuração das relações MODBUS, visualizadas na figura a seguir, segue os parâmetros descritos na sequência.

	Value Variable	Data Type	Data Start Address	Data Size	Data Range	Quality Variable
*		▼				

Figura 3-17. Tela de Mapeamentos de Dados MODBUS

Parâmetros associados:

- **Value Variable** - Esse campo é utilizado para especificar uma variável simbólica na relação MODBUS;
- **Data Type** - Esse campo é utilizado para especificar o tipo de dado utilizado na relação MODBUS;
- **Data Start Address** - Endereço inicial do dado de uma relação MODBUS;
- **Data Size** - O valor de Tamanho especifica a quantidade máxima de dados que uma relação MODBUS poderá acessar, a partir do endereço inicial. Sendo assim, para ler uma faixa de endereços contínua, é necessário que todos os endereços estejam declarados em uma única relação. Este campo varia de acordo com o tipo de dado MODBUS configurado;
- **Data Range** - Este campo mostra ao usuário a faixa de endereços de memória utilizada pela relação MODBUS;
- **Quality Variable** - Nome da variável simbólica que armazena a qualidade do mapeamento, que deve ser do tipo QUALITY, disponível na biblioteca LibDataTypes.

### NOTAS:

1. Cada COM tem limite de 20000 pontos de comunicação, ou seja, caso o usuário utilize os 20000 pontos em uma única relação na COM1, não será possível utilizar mais pontos de comunicação nesta COM. Entretanto uma boa prática é utilizar quantidades e tipos de relações apropriadas às necessidades da aplicação. Para mais informações consulte a seção “Pontos Duplos”.
2. Diferentemente de outras tarefas de uma aplicação, quando for atingida uma marca de depuração na MainTask, a tarefa de uma instância MODBUS RTU Escravo, e qualquer outra tarefa MODBUS, irá parar de ser executada quando tentar efetuar uma escrita em uma área de memória. Isto ocorre para manter a consistência dos dados das áreas de memória enquanto a MainTask não estiver em execução.

## MODBUS Ethernet

A rede de comunicação multimestre permite que a UCP Hadron Xtorm leia ou escreva variáveis MODBUS em outros controladores ou IHMs compatíveis com os protocolos MODBUS TCP ou MODBUS RTU via TCP. A UCP Hadron Xtorm 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 representa algumas das possibilidades de comunicação utilizando-se o protocolo MODBUS TCP simultaneamente com o protocolo MODBUS RTU via TCP.

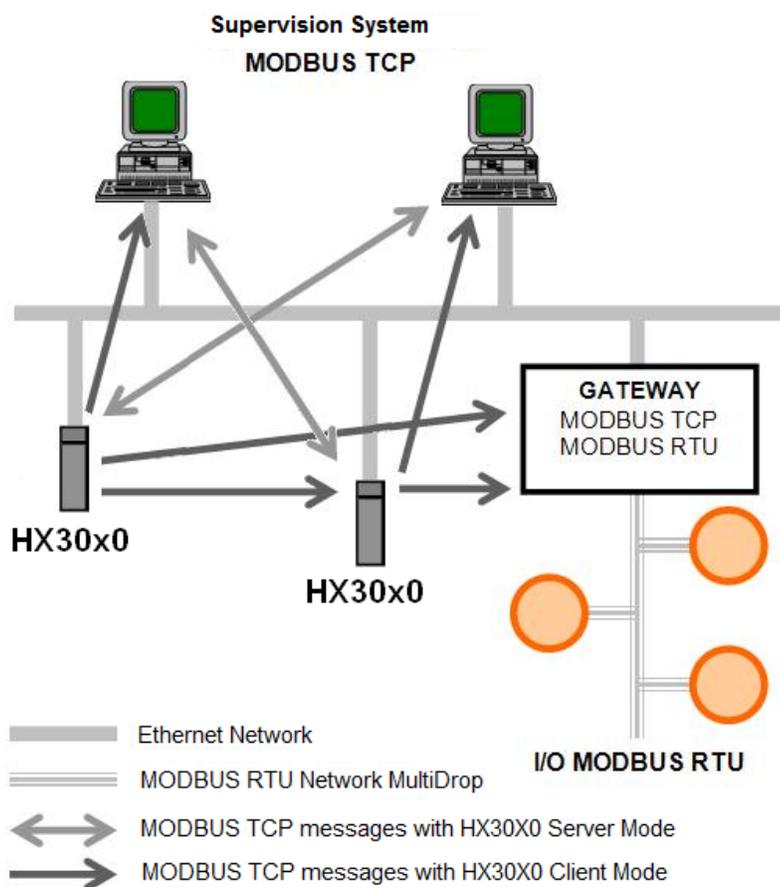


Figura 3-18. Rede de Comunicação MODBUS TCP

A associação de variáveis MODBUS com variáveis simbólicas da UCP é realizada pelo usuário através da definição de relações via configurador MasterTool Xtorm. 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 simbólicas identificadas na coluna Desabilitação no MasterTool Xtorm. A desabilitação ocorre através destes 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. 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.

#### **MODBUS Ethernet Cliente**

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

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

1. Adicionar a instância do protocolo MODBUS Ethernet Cliente a um dos canais Ethernet disponíveis (NET 1 ao NET 6);
2. Realizar as etapas indicadas na seção “Inserindo uma Instância de Protocolo”;
3. Configurar a interface Ethernet. Para realizar esse procedimento, consultar o capítulo Configuração das Interfaces Ethernet;
4. Configurar os parâmetros gerais do protocolo MODBUS Cliente, com o protocolo TCP ou RTU via TCP;
5. Adicionar e configurar dispositivos, definindo endereço IP, porta, endereço do escravo e timeout de comunicação (disponível no botão de configurações avançadas do Dispositivo);
6. Adicionar e configurar os mapeamentos MODBUS, especificando o nome da variável, tipo de dados, endereço inicial do dado, tamanho do dado e variável que receberá os dados de qualidade;
7. Adicionar e configurar as requisições MODBUS, especificando a função desejada, o tempo de varredura da requisição, o endereço inicial (leitura/escrita), o tamanho dos dados (Leitura/Escrita), a variável que receberá os dados de qualidade, e a variável responsável por desabilitar a requisição.

#### **Parâmetros Gerais do Protocolo MODBUS Cliente**

Os parâmetros gerais, encontrados na tela inicial de configuração do protocolo MODBUS estão ilustrados na figura a seguir. O parâmetro-base a ser ajustado é o Connection Mode, ou seja, a seleção do tipo de protocolo. Os diagnósticos e comandos do protocolo MODBUS TCP Cliente

configurado são armazenados em variáveis do tipo T\_DIAG\_MODBUS\_ETH\_CLIENT\_1. Todos os contadores dos diagnósticos do MODBUS TCP Cliente retornam à zero quando o valor limite 65535 é ultrapassado.

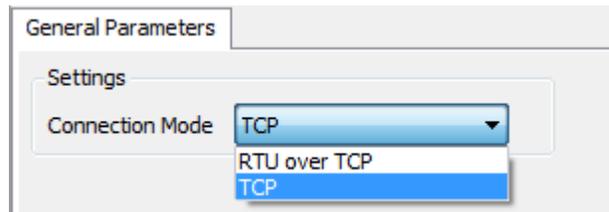


Figura 3-19. Tela de Configuração Parâmetros Gerais MODBUS TCP Cliente

### Configuração dos Dispositivos

A configuração dos dispositivos escravos se dá conforme ilustrado na figura a seguir.

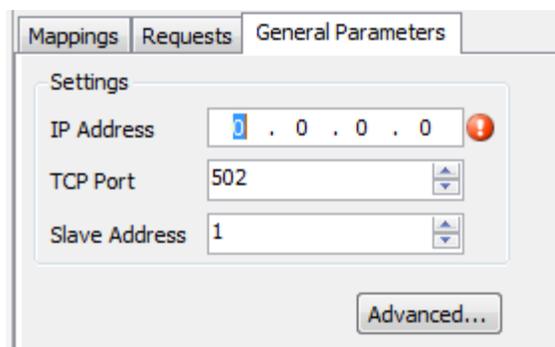


Figura 3-20. Tela de Configurações dos Parâmetros Gerais do Dispositivo

Parâmetros associados:

- **IP Address** - Endereço IP do dispositivo servidor Modbus;
- **TCP Port** – Porta TCP. Caso sejam adicionadas várias instâncias do protocolo em uma única interface Ethernet, diferentes portas TCP devem ser selecionadas para cada instância. Algumas portas TCP são reservadas;
- **Slave Address** - Endereço do Escravo MODBUS. De acordo com a Norma MODBUS, a faixa de endereços válidos para escravos é de 0 a 247, sendo os endereços 248 a 255 reservados. Quando o mestre envia um comando de escrita com o endereço configurado como zero, ele está realizando requisições broadcast na rede.

### Timeout de comunicação

Os parâmetros de Timeout de comunicação do protocolo MODBUS Cliente, encontrados no botão “Avançado...” da tela de configuração, estão divididos em: número máximo de requisições simultâneas, timeout de comunicação, modo de timeout de conexão e tempo de inatividade. Esses parâmetros são explicados na sequência.

- **Número Máximo de Requisições Simultâneas:** é utilizado em servidores com um alto ciclo de varredura. Esse parâmetro é fixado em 1 (não editável), quando o protocolo configurado é MODBUS RTU via TCP.
- **Timeout de Comunicação:** o timeout da comunicação é o tempo que o cliente aguardará por uma resposta do servidor à requisição. Para um dispositivo MODBUS Cliente, duas variáveis do sistema devem ser consideradas: o tempo que o servidor leva para processar a requisição e o atraso de envio da resposta caso seja configurado no

servidor. É recomendado que o timeout seja igual ou maior que duas vezes a soma destes parâmetros.

- **Modo:** define quando a conexão com o servidor é finalizada pelo cliente.
- **Tempo de Inatividade:** tempo de inatividade da conexão.

📌 **NOTA:**

Caso o usuário utilize uma variável que abranja o endereço 5 a 10 de uma função declarada na aba de Mapeamentos, será válida a leitura/escrita de Requisições que atinjam as extremidades ou parte do Mapeamento. Exemplo: a leitura/escrita dessa função do endereço 1 ao 5, onde somente o endereço 5 está declarado. Este endereço receberá valores válidos e o restante será desconsiderado, sem valores válidos.

### Configuração dos Mapeamentos

A configuração das relações MODBUS Cliente, visualizadas na figura a seguir segue os parâmetros descritos na sequência.

	Value Variable	Data Type	Data Start Address	Data Size	Data Range	Quality Variable
*						

Generate Quality Variables

Figura 3-21. Tela de Mapeamentos de Dados MODBUS Cliente

Parâmetros associados:

- **Value Variable** - Esse campo é utilizado para especificar uma variável simbólica na relação MODBUS;
- **Data Type** - Esse campo é utilizado para especificar o tipo de dado utilizado na relação MODBUS;
- **Data Start Address** - Endereço inicial do dado de um mapeamento MODBUS;
- **Data Size** - O valor de Tamanho especifica a quantidade máxima de dados que uma relação MODBUS poderá acessar, a partir do endereço inicial. Sendo assim, para ler uma faixa de endereços contínua, é necessário que todos os endereços estejam declarados em uma única relação. Este campo varia de acordo com o tipo de dado MODBUS configurado;
- **Data Range** - Este campo mostra ao usuário a faixa de endereços de memória utilizada pela relação MODBUS;
- **Quality Variable** - Este campo é utilizado para especificar uma variável que vai conter os dados da qualidade da relação MODBUS;
- **Generate Quality Variables** - Gera automaticamente na GVL Qualities variáveis do tipo QUALITY para cada Variável de Valor declarada.

### Configuração das Requisições

A configuração das requisições MODBUS, visualizadas na figura a seguir segue os parâmetros descritos na sequência.

	Function Code	Polling (ms)	Read Data Start Address	Read Data Size	Read Data Range	Write Data Start Address	Write Data Size	Write Data Range	Diagnostic Variable	Disabling Variable
*										

Diagnostics Variable Type:

Figura 3-22. Tela de Requisições de dados MODBUS

Parâmetros associados:

- **Function Code** - Tipo de função MODBUS. Inclui: Leitura de coils (FC 01), Leitura de input status (FC 02), Leitura de holding registers (FC 03), Leitura de input registers (FC 04), Escrita de um coil (FC 05), Escrita de um holding register (FC 06), Escrita de múltiplos

coils (FC 15), Escrita de holding registers (FC 16), Máscara de escrita do register (FC 22) e Leitura/escrita holding registers (FC 23);

- **Polling (ms)** - Período de comunicação (ms). Este parâmetro indica com que frequência à comunicação definida por esta requisição deve ser executada. Ao ser finalizada uma comunicação será aguardado um tempo igual ao configurado no campo varredura e, após, será executada uma nova comunicação;
- **Read Data Start Address** - Campo destinado ao endereço inicial dos dados de leitura MODBUS;
- **Read Data Size** - Tamanho dos dados de leitura MODBUS. O valor mínimo para o tamanho dos dados de leitura é 1 e o valor máximo depende da função MODBUS (FC) utilizada;
- **Read Data Range** - Este campo mostra a faixa de dados de leitura MODBUS configurada para cada requisição. O endereço inicial de leitura, somado ao tamanho do dado de leitura resultará na faixa de dados de leitura de cada uma das requisições;
- **Write Data Start Address** - Campo destinado ao endereço inicial dos dados de escrita MODBUS;
- **Write Data Size** - Tamanho dos dados de escrita MODBUS. O valor mínimo para o tamanho dos dados de escrita é 1 e o valor máximo depende da função MODBUS (FC) utilizada;
- **Write Data Range** - Este campo mostra a faixa de dados de escrita MODBUS configurada para cada requisição. O endereço inicial de escrita, somado ao tamanho do dado de escrita resultará na faixa de dados de escrita de cada uma das requisições;
- **Diagnostic Variable** - Nome da variável de diagnóstico. As variáveis simbólicas de diagnósticos podem ser geradas automaticamente através do botão "Gerar Variáveis de Diagnósticos". Ao clicar nesse botão, será criada uma variável do tipo "T\_DIAG\_MODBUS\_ETH\_MAPPING\_1", na GVL "ReqDiagnostics", para cada uma das requisições;
- **Disabling Variable** - Variável utilizada para desabilitar a relação MODBUS.

Os diagnósticos da requisição MODBUS configurada, são armazenados em variáveis do tipo T\_DIAG\_MODBUS\_ETH\_CLIENT\_1.

🔔 **NOTA:**

Diferentemente de outras tarefas de uma aplicação, quando for atingida uma marca de depuração na MainTask, a tarefa de uma instância MODBUS Ethernet Cliente, e qualquer outra tarefa MODBUS, irá parar de ser executada quando tentar efetuar uma escrita em uma área de memória. Isto ocorre para manter a consistência dos dados das áreas de memória enquanto a MainTask não estiver em execução.

#### **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 aplicativo do usuário:

1. Definir tempo máximo de Polling para as relações;
2. Manter a relação normalmente desabilitada;
3. Habilitar a relação quando se deseja executá-la;
4. Esperar pela confirmação de término da execução da relação, e neste momento desabilitá-la novamente.

## MODBUS Ethernet Servidor

Este protocolo está disponível para a UCP da Série Hadron Xtorm nos seus canais Ethernet. Ao selecionar esta opção no MasterTool Xtorm, a UCP passa a ser servidor da comunicação MODBUS, permitindo a conexão com dispositivos cliente MODBUS. Este protocolo somente está disponível quando a UCP estiver em modo de execução (Modo Run). Para configurar este protocolo, é necessário executar os seguintes passos:

1. Adicionar a instância do protocolo MODBUS Servidor a um dos canais Ethernet disponíveis (NET 1 ao NET 6);
2. Realizar o procedimento previsto na seção “Inserindo uma Instância de Protocolo”;
3. Configurar a interface Ethernet conforme procedimento descrito neste Tutorial;
4. 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);
5. Adicionar e configurar os mapeamentos MODBUS, especificando o nome da variável, tipo de dados, endereço inicial do dado e tamanho do dado.

### Parâmetros Gerais do Protocolo MODBUS Servidor

Os parâmetros gerais, encontrados na tela inicial de configuração do protocolo MODBUS estão ilustrados na figura a seguir.



Figura 3-23. Tela de Configuração Parâmetros Gerais MODBUS Servidor

Parâmetros associados:

- **TCP Port** - Porta TCP. Caso sejam adicionadas várias instâncias do protocolo em uma única interface Ethernet, diferentes portas TCP devem ser selecionadas para cada instância. Algumas portas TCP, são reservadas;
- **Connection Mode** - Seleção do protocolo;
- **Filters...** - Os filtros são utilizados para estabelecer um intervalo de endereços IP que têm acesso de escrita ou leitura nas relações MODBUS, sendo individualmente configurados. O critério de permissão é realizado através de uma operação lógica AND entre o Filtro de Máscara para Escrita e o endereço IP do cliente. Caso o resultado seja igual ao Filtro de IP para Escrita, o cliente tem direito de escrita. Por exemplo, se o Filtro de IP para Escrita = 192.168.15.0 e o Filtro de Máscara para Escrita = 255.255.255.0, então somente clientes com endereço IP = 192.168.15.x terão direito de escrita. O mesmo procedimento é aplicado nos parâmetros de Filtro de Leitura para definir os direitos de leitura;
- **Advanced...** - Os tempos de comunicação do protocolo MODBUS Servidor, encontrados no botão “Avançado...” da tela de configuração, estão divididos em:
  - **Ciclo da Tarefa (ms)** - Tempo para execução da instância dentro do ciclo, sem considerar o tempo de execução desta. O usuário deverá ter cuidado a alterar esse parâmetro, pois ele interfere diretamente no tempo de resposta, volume de dados

por varredura e, principalmente, no balanceamento dos recursos da UCP entre comunicações e outras tarefas.

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

Os diagnósticos e comandos do protocolo MODBUS TCP Servidor configurado, são armazenados em variáveis do tipo T\_DIAG\_MODBUS\_ETH\_SERVER\_1. Todos os contadores dos diagnósticos do MODBUS Ethernet Servidor retornam a zero quando o valor limite 65535 é ultrapassado.

### Configuração dos Mapeamentos

A configuração das relações MODBUS Servidor está ilustrada na figura a seguir e os parâmetros associados estão descritos na sequência.

	Value Variable	Data Type	Data Start Address	Absolute Data Start Address	Data Size	Data Range
*						

Figura 3-24. Tela de Mapeamentos de Dados MODBUS Servidor

Parâmetros associados:

- **Value Variable** - Esse campo é utilizado para especificar uma variável simbólica na relação MODBUS;
- **Data Type** - Esse campo é utilizado para especificar o tipo de dado utilizado na relação MODBUS;
- **Data Start Address** - Endereço inicial do dado de um mapeamento MODBUS;
- **Absolute Data Start Address** - Endereço inicial absoluto dos dados MODBUS conforme o seu tipo. Por exemplo, o Holding Register com endereço 5 possui endereço absoluto 400005. Este é campo é apenas leitura e está disponível para auxiliar na configuração do MODBUS Cliente/Mestre que irá comunicar-se com este dispositivo. Os valores dependem do endereço base (offset) de cada tipo de dado MODBUS e do endereço permitido para cada tipo de dado;
- **Data Size** - O valor de Tamanho especifica a quantidade máxima de dados que uma relação MODBUS poderá acessar, a partir do endereço inicial. Sendo assim, para ler uma faixa de endereços contínua, é necessário que todos os endereços estejam declarados em uma única relação. Este campo varia de acordo com o tipo de dado MODBUS configurado;
- **Data Range** - É um campo somente de leitura e informa a faixa de endereços que está sendo usada por esse mapeamento. Ele é formado pela soma dos campos "Endereço Inicial" e "Tamanho". Não pode haver sobreposições de faixa com outros mapeamentos do mesmo "Tipo de Dado".

#### NOTAS:

1. Cada NET tem limite de 20000 pontos de comunicação, ou seja, caso o usuário utilize os 20000 pontos em uma única relação na NET1, não será possível utilizar mais pontos de comunicação nesta NET. Entretanto uma boa prática é utilizar quantidades e tipos de relações apropriadas às necessidades da aplicação.
2. Diferentemente de outras tarefas de uma aplicação, quando for atingida uma marca de depuração na MainTask, a tarefa de uma instância MODBUS Ethernet Servidor, e qualquer outra tarefa MODBUS, irá parar de ser executada quando tentar efetuar uma escrita em uma área de memória. Isto ocorre para manter a consistência dos dados das áreas de memória enquanto a MainTask não estiver em execução.

#### DNP3 Ethernet Cliente

Este protocolo está disponível para a UCP da Série Hadron Xtorm nos seus canais Ethernet. Ao selecionar esta opção no MasterTool Xtorm, a UCP passa a ser cliente da comunicação DNP3. O limite máximo de instâncias de drivers DNP3 Cliente depende de quantos drivers de protocolos estão instanciados e são suportados pelas UCPs HX3040 no total. Este protocolo está ativo somente quando a UCP estiver em modo de execução (modo RUN). Para configurar este protocolo, é necessário executar os seguintes passos:

1. Adicionar a instância do protocolo DNP3 Cliente a um dos canais Ethernet disponíveis (NET 1 ao NET 6). Para realizar esse procedimento, consultar a seção “Inserindo uma Instância de Protocolo”;
2. Configurar a interface Ethernet. Para realizar esse procedimento, consultar a seção “Configuração das Interfaces Ethernet”;
3. Configurar os parâmetros gerais do protocolo DNP3 Cliente com o modo Endereço de Enlace;
4. Adicionar e configurar os dispositivos Outstation, definindo os devidos parâmetros. Para cada driver instanciado, existe um limite máximo de 32 outstations;
5. Adicionar e configurar os mapeamentos DNP3, especificando o nome da variável, grupo estático, índices, tamanho, faixa, classe e qualidade;
6. Configurar os parâmetros da camada de enlace, especificando os endereçamentos e as mensagens de confirmação;
7. Configurar os parâmetros da camada de aplicação, especificando as configurações da camada de aplicação, varredura das classes e solicitações automáticas.

#### Configuração dos Mapeamentos DNP3 Cliente

A configuração dos Parâmetros Gerais de um DNP3 Cliente está ilustrada na figura a seguir e contempla a definição do parâmetro Address (Master), o qual define o endereço do mestre DNP3.

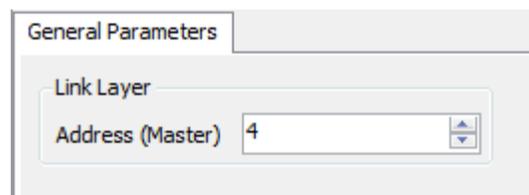


Figura 3-25. Tela de Parâmetros Gerais do DNP3 Cliente

A figura a seguir mostra a configuração das relações DNP3 Cliente.

	Value Variable	Static Group	Index	Size	Range	Quality Variable
*						

Generate Quality Variables

Figura 3-26. Tela de Mapeamentos de Dados DNP3 Cliente

Parâmetros associados:

- **Value Variable** - Nome da variável simbólica a ser mapeada. Quando um comando de leitura é enviado, o retorno enviado na resposta será armazenado nesta variável. Quando for um comando de escrita, o valor escrito será copiado desta variável. A variável pode ser simples, array, elemento de array, elemento simples de estrutura, ou elemento array de estruturas;
- **Static Group** - É utilizado para selecionar em qual grupo DNP3 que o mapeamento será criado;
- **Index** - O campo é utilizado para configurar o índice do primeiro ponto do mapeamento no protocolo DNP3;
- **Size** - É um campo somente leitura que informa quantos pontos estão sendo usados por esse mapeamento. O valor deste campo varia conforme o tipo da variável configurada no mapeamento;
- **Range** - É um campo somente de leitura que informa a faixa de endereços que está sendo usada por esse mapeamento. Ele é formado pela soma dos campos "Índices" e "

Tamanho". Não pode haver sobreposições de faixa com outros mapeamentos do mesmo tipo;

- **Quality Variable** - Nome da variável simbólica que armazenará a qualidade do mapeamento, sendo esta, sempre do tipo QUALITY, disponível a partir da biblioteca LibDataTypes. A variável pode ser simples, array ou elemento de array e pode estar em estruturas;
- **Generate Quality Variable** - Este botão gera automaticamente na GVL Qualities variáveis do tipo QUALITY para cada Variável de Valor declarada. Para mais informações acesse o capítulo Conversões de Qualidade.

#### NOTAS:

1. Quando variáveis do grupo 30 (Analogicas de entrada) e grupo 40 (Analogicas de saída) recebem valores do Outstation que não podem ser armazenados devido à diferença entre as variações dos dados recebidos e da variável de armazenamento mapeada no DNP3 Cliente, a qualidade daquele ponto será considerada INVALID e o valor do ponto será o último valor válido lido pelo DNP3 Cliente. Isto ocorrerá somente no caso do protocolo DNP3 Servidor responder em uma variação maior do que a mapeada no DNP3 Cliente. Se o DNP3 Servidor responder na mesma variação ou em uma variação inferior, o valor e a qualidade do ponto serão armazenados conforme a resposta do DNP3 Servidor. Por exemplo: Variável do tipo INT (16-bit) mapeada no DNP3 Cliente e DNP3 Servidor responde na variação de 32-bit. Neste caso, o valor contido na resposta do DNP3 Servidor não será atualizado e a qualidade será INVALID.
2. O driver DNP3 Cliente não gera eventos baseados nos valores de seus mapeamentos. Os eventos devem ser gerados pelo respectivo IED. Sendo assim, caso o IED não tenha suporte a geração de eventos, o driver DNP3 Cliente não será capaz de gerá-los.

### Configuração da Camada de Enlace DNP3 Cliente

A figura a seguir mostra a configuração dos parâmetros da camada de enlace do DNP3 Cliente. Os parâmetros associados estão descritos na sequência.

Figura 3-27. Tela de Configurações da Camada de Enlace DNP3 Cliente

Parâmetros associados:

- **Address (Outstation)** - Endereço DNP3 de origem deste Outstation;
- **IP Address** - Endereço IP deste Outstation;

- **TCP Port** - Endereço da porta Listen para conexão com este Outstation;
- **Sends Confirmed User Data Frames** - Seleciona o modo de confirmação de mensagens (Nunca, Sempre ou Às vezes);
- **Maximum Retries** - Número de vezes as mensagens serão transmitidas caso confirmação do Outstation não seja recebida. É habilitado quando, na opção "Modo de Confirmação da Camada de Enlace", estiver selecionado "Sempre" ou "Às vezes";
- **Confirmation Time-Out(s)** - Determina qual é o tempo (em segundos) a ser aguardado para que o Outstation envie a mensagem de confirmação. É habilitado quando, na opção "Modo de Confirmação da Camada de Enlace", estiver selecionado "Sempre" ou "Às vezes";
- **Connection Time-Out(s)** - Tempo máximo para que este Outstation se conecte com o DNP3 Cliente através do protocolo TCP;
- **Keep Alive (s)** - Tempo do teste de link. Intervalo de tempo para envio de mensagens de teste de link ("TCP Keep Alive" ou "Serial Link Status") após última mensagem DNP3 recebida. Caso não seja recebida uma resposta para a mensagem de teste de link, a conexão atual será encerrada.

#### **Configuração da Camada de Aplicação DNP3 Cliente**

A figura a seguir mostra a configuração dos parâmetros da camada de aplicação do DNP3 Cliente. Os parâmetros associados estão descritos na sequência.

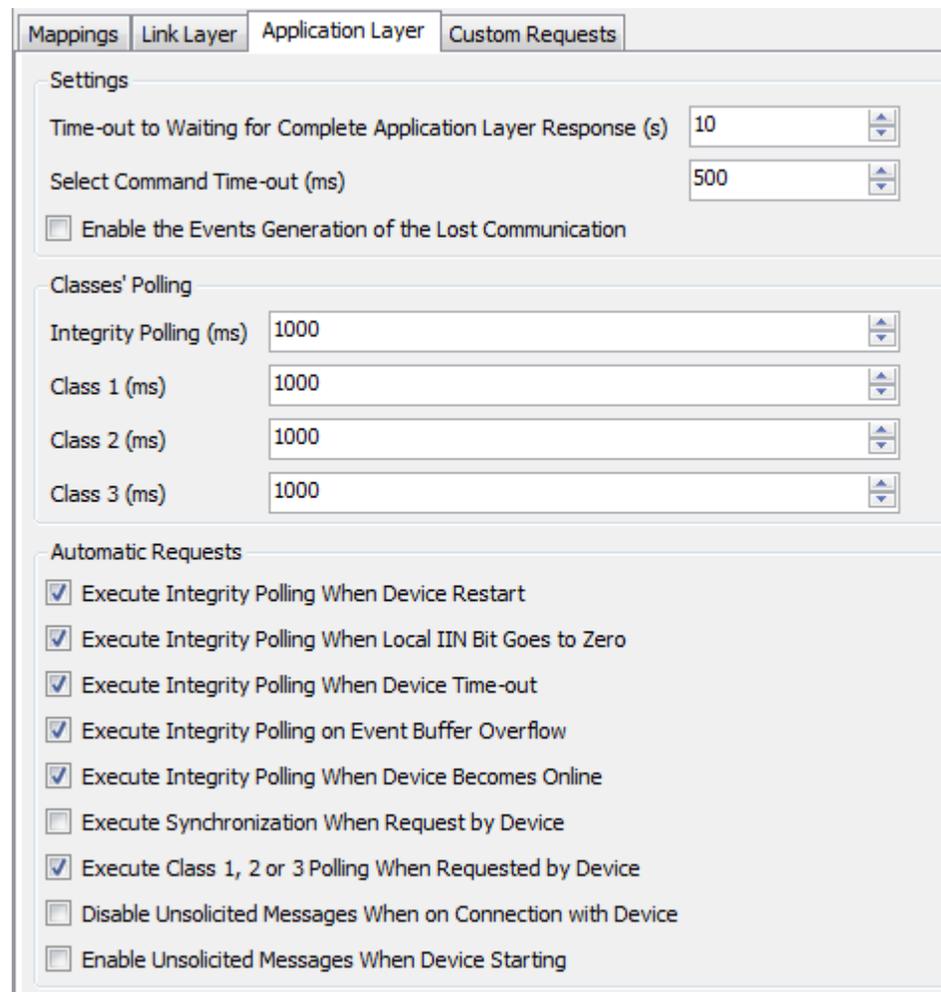


Figura 3-28. Tela de Configurações da Camada de Aplicação DNP3 Cliente

Parâmetros associados:

- **Time-out to Waiting for Complete Application Layer Response (s)** - Tempo máximo que o Outstation aguardará resposta completa da camada de aplicação;
- **Select Command Time-out (ms)** - Tempo máximo para a execução do comando de seleção;
- **Enable the Events Generation of the Lost Communication** - Habilita a geração de eventos nas situações de perda de comunicação e restabelecimento de comunicação com o Outstation;
- **Integrity Polling (ms)** - Determina o tempo (em milissegundos) em que é executada a varredura de todos os pontos mapeados;
- **Class 1 (ms)** - Tempo em que é executada a varredura dos pontos mapeados para a "Classe 1";
- **Class 2 (ms)** - Tempo em que é executada a varredura dos pontos mapeados para a "Classe 2";
- **Class 3 (ms)** - Tempo em que é executada a varredura dos pontos mapeados para a "Classe 3";
- **Execute Integrity Polling When Device Restart** - Atualiza os valores de todos os pontos do dispositivo Outstation e lê todos os eventos quando o bit DEVICE\_RESTART do IIN estiver em "1";

- **Execute Integrity Polling When Local IIN Bit Goes to Zero** - Executa a varredura de integridade quando o bit LOCAL do IIN passar de um para zero;
- **Execute Integrity Polling When Device Timeout** - Executa a varredura de integridade quando o Outstation estiver operando em modo local, não sendo acessível pelo mestre;
- **Execute Integrity Polling on Event Buffer Overflow** - Executa a varredura de integridade após um estouro na fila de eventos do Outstation, indicado pelo bit EVENT\_BUFFER\_OVERFLOW do IIN;
- **Execute Integrity Polling When Device Becomes Online** - Executa a varredura de integridade quando a comunicação com o mestre for restabelecida após um timeout;
- **Execute Synchronization When Request By Device** - Permite que o mestre envie o comando de acerto de hora quando o Outstation solicitar;
- **Execute Class 1,2 or 3 Polling When Requested by Device** - Executa a leitura dos eventos das três classes quando um dos bits CLASS\_1\_EVENTS, CLASS\_2\_EVENTS e CLASS\_3\_EVENTS do IIN estiver em "1";
- **Disable Unsolicited Messages When on Connection with Device** - Envia o comando de desativação das mensagens não solicitadas quando o Outstation conectar. Quando este campo é marcado, a caixa de seleção "Habilitar mensagens não solicitadas quando o dispositivo iniciar" é desmarcada e desabilitada;
- **Enable Unsolicited Messages When Device Starting** - Envia o comando de ativação das mensagens não solicitadas quando o Outstation conectar.

#### Requisições Customizadas DNP3 Cliente

A figura a seguir mostra a configuração das relações customizadas do DNP3 Cliente. Os parâmetros associados estão descritos na sequência.

	Object Variation	Qualifier	Start	Stop	Count	Polling (ms)	Diagnostic Variable
*							

Diagnostic Variable Type:

Figura 3-29. Tela de Mensagens Requisições Customizadas DNP3 Cliente

Parâmetros associados:

- **Object Variation** - Define o grupo e a variação dos objetos que serão requisitados;
- **Qualifier** - Identifica qual o tipo de requisição que será realizada. As opções disponíveis dependem do tipo de Variação do Objeto escolhido (consulte a documentação de produto para informações adicionais referentes aos qualificadores);
- **Start** - Define qual é o primeiro ponto de uma faixa que será requisitada. Só pode ser preenchido se a coluna "Qualificador" estiver com o valor "Faixa" selecionado;
- **Stop** - Define qual é o último ponto de uma faixa que será requisitada. Só pode ser preenchido se a coluna "Qualificador" estiver com o valor "Faixa" selecionado;
- **Count** - Define quantos pontos serão requisitados. Só pode ser preenchido se a coluna "Qualificador" estiver com o valor "Contador" selecionado;
- **Polling (ms)** - Define o período (em milissegundos) em que ocorre a requisição;
- **Diagnostic Variable** - Nome da variável simbólica que receberá o diagnóstico da requisição;
- **Generated Diagnostic Variables** - As variáveis simbólicas de diagnósticos podem ser geradas automaticamente através do botão "Gerar Variáveis de Diagnósticos". Ao clicar

nesse botão, será criada uma variável do tipo " T\_DIAG\_DNP\_CLIENT\_REQUEST\_1", na GVL "ReqDiagnostics", para cada uma das requisições.

Os diagnósticos e comandos do protocolo DNP3 Cliente configurado, são armazenados em variáveis do tipo T\_DIAG\_DNP\_CLIENT\_REQUEST\_1.

🔔 **NOTA:**

Diagnósticos de Requisição: de todos os bits disponibilizados na estrutura tIIN, apenas três deles são efetivamente relacionados à requisição: NO\_FUNC\_CODE\_SUPPORT, OBJECT\_UNKNOWN e PARAMETER\_ERROR. Todos os outros estão relacionados à Outstation para a qual a requisição está endereçada. Desta forma, caso o usuário deseje avaliar os diagnósticos relacionados a uma Outstation, devem ser consultados os bits de diagnóstico de uma requisição que esteja ativa e que seja periodicamente executada.

#### Comandos DNP3 para IEDs

Nesta seção, estão listados os comandos de usuário disponíveis para serem enviados via DNP3 para IEDs. Comando Implementados:

- Código de Função 03: Select;
- Código de Função 04: Operate;
- Código de Função 05: Direct Operate;
- Código de Função 06: Direct Operate – No Acknowledgement;
- Código de Função 13: Cold Restart.

Devido às características do protocolo, não é suportado o envio de comandos simultâneos, mesmo que destinados a diferentes pontos ou mesmo a diferentes IEDs. Caso um comando seja disparado pela aplicação do usuário enquanto houver outro comando em execução, o novo comando aguardará o término do primeiro por um período definido no bloco funcional (udiCommandTimeout). Se o tempo para o fim da execução do primeiro comando for maior que udiCommandTimeout, o novo comando retornará uma mensagem de erro (DNP3\_COMMAND\_ERROR\_STATUS\_TIMEOUT). Ou seja, um comando começa a contabilizar o tempo de timeout desde o momento em que foi disparado pela aplicação do usuário, e não somente ao ser enviado pelo DNP3 Cliente.

Os comandos para as saídas digitais são definidos no bloco funcional DNP3\_BinaryCommand. Os comandos para as saídas analógicas são definidos em dois blocos funcionais: DNP3\_AnalogCommandInt e DNP3\_AnalogCommandReal. O comando para Cold Restart é definido no bloco funcional DNP3\_ColdCommand.

Os comandos DNP3 retornam status de erro (eErrorStatus) ao serem executados (consulte a documentação do produto para informações adicionais referentes aos códigos de erro).

#### DNP3 Ethernet Servidor

Este protocolo está disponível para a UCP da Série Hadron Xtorm nos seus canais Ethernet. Ao selecionar esta opção no MasterTool Xtorm, a UCP passa a ser servidor da comunicação DNP3, permitindo a conexão com até cinco dispositivos clientes DNP3. Para cada cliente, o driver possui uma fila de eventos exclusiva com as seguintes características:

- Tamanho: 4500 eventos;
- Política de estouro: mantém o mais recente.

Para configurar este protocolo é necessário executar os seguintes passos:

1. Adicionar a instância do protocolo DNP3 Servidor a um dos canais Ethernet disponíveis (NET 1 ao NET 6);

2. Realizar o procedimento indicado na seção “Inserindo uma Instância de Protocolo”;
3. Configurar a interface Ethernet conforme indicado anteriormente;
4. Configurar os parâmetros gerais do protocolo DNP3 Servidor, com o modo de conexão Porta ou IP, e o número da porta TCP quando o modo de conexão selecionado for o modo IP;
5. Adicionar e configurar dispositivos, definindo devidos parâmetros;
6. Adicionar e configurar os mapeamentos DNP3, especificando o nome da variável, variação do grupo estático, variação do grupo de eventos, índice, tamanho, faixa, classe do evento, banda morta, e tipo da banda morta;
7. Configurar os parâmetros da camada de enlace, especificando os endereçamentos e as mensagens de confirmação;
8. Configurar os parâmetros da camada de aplicação, especificando a fragmentação de respostas, mensagens de confirmação, Polling de integridade, sincronismo, comandos e fila de eventos;
9. Configurar os parâmetros das mensagens não solicitadas, especificando os parâmetros de controle e as condições de disparo.

### Configuração dos Mapeamentos DNP3 Servidor

A figura a seguir mostra as configurações dos Parâmetros Gerais de um DNP3 Servidor.

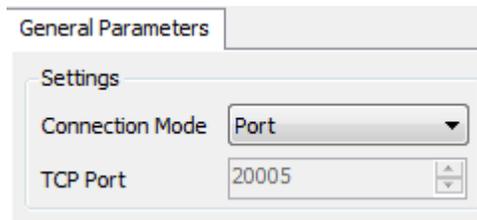


Figura 3-30. Tela de Parâmetros Gerais do DNP3 Servidor

Parâmetros associados:

- **Connection Mode** - Configura o modo de conexão com os módulos Cliente Conectados. Os modos disponíveis para conexão são "IP" ou "Porta";
- **TCP Port** - Define qual o número da porta TCP da UTR será utilizada para a comunicação com os módulos Cliente Conectados, caso no campo "Modo de Conexão" esteja selecionado "IP".

A figura a seguir mostra as configurações das relações DNP3 Servidor.

Mappings	Link Layer	Application Layer	Unsolicited Messages						
	Value Variable	Static Group and Default Variation	Event Group and Default Variation	Event Class	Index	Size	Range	Dead Band Variable	Dead Band Type
*									

Figura 3-31. Tela de Mapeamentos de Dados DNP3 Servidor

Parâmetros associados:

- **Value Variable** - Nome da variável simbólica a ser mapeada. Quando um comando de leitura é enviado, o retorno enviado na resposta será armazenado nesta variável. Quando for um comando de escrita, o valor escrito será armazenado nesta variável. A variável pode ser simples, array ou elemento de array e pode estar em estruturas;

- **Static Group and Default Variation** - O campo é utilizado para configurar a variação dos dados estáticos que serão retornados caso o DNP3 Cliente não especifique uma variação na requisição dos dados, ou no caso de mensagens não solicitadas;
- **Event Group and Default Variation** - O campo é utilizado para configurar a variação para os eventos que serão retornados caso o DNP3 Cliente não especifique uma variação na requisição dos dados, ou no caso de mensagens não solicitadas;
- **Event Class** - O campo é utilizado para configurar em qual classe os eventos deste mapeamento serão armazenados. Sendo que para pontos configurados como classe 0, não serão reportados eventos. Lembrando que para os grupos 10 e 40, o valor deste campo é fixo em zero, visto que tratam de saídas digital e analógicas respectivamente;
- **Index** - O campo é utilizado para configurar o índice (endereço) do primeiro ponto do mapeamento no protocolo DNP3;
- **Size** - Este campo mostra a quantidade máxima de dados que a relação DNP3 poderá acessar, a partir do endereço inicial. Sendo assim, para ler uma faixa de endereços contínua, é necessário que todos os endereços estejam declarados em uma única relação. Este campo varia de acordo com o tipo de dado DNP3 configurado, se variável simples ou array, e é calculado automaticamente pela ferramenta de configuração a partir da declaração da Value Variable;
- **Range** - Este campo mostra ao usuário a faixa de endereços de memória utilizada pela relação DNP3, calculado automaticamente pela ferramenta de configuração a partir dos campos Index e Size.
- **Dead Band Variable** - Este campo é utilizado para especificar uma variável que vai conter os dados da banda morta da relação DNP3. Ela deve ter o mesmo tipo e tamanho da variável mapeada na coluna Value Variable. Novos valores da variável de banda morta serão considerados somente quando a variável analógica de entrada mudar de valor.
- **Dead Band Type** - Os tipos de configuração da Banda Morta disponíveis são os seguintes:
  - **Desabilitada** - Nesta opção, qualquer modificação de valor num ponto do grupo, por menor que seja, gera evento para este ponto;
  - **Absoluta** - Nesta opção, se o módulo da variação do valor de um ponto do grupo for maior do que o valor configurado pela variável no campo "Banda Morta", um evento para este ponto é gerado;
  - **Integrada** - Nesta opção, se o módulo da integral da variação do valor de um ponto do grupo for maior do que o valor configurado pela variável no campo "Banda Morta", um evento para este ponto é gerado. O intervalo de integração é de um segundo.

‡ **NOTA:**

A banda morta configurada para variáveis mapeadas no DNP3 Servidor não é aplicada quando estes pontos forem oriundos de um IED (através do driver DNP3 Cliente, IEC 61850 GOOSE Subscriber etc.), pois neste caso a detecção dos eventos ocorre no próprio IED.

#### **Configuração da Camada de Enlace DNP3 Servidor**

A figura a seguir mostra a configuração dos parâmetros da camada de enlace do DNP3 Servidor.

Figura 3-32. Tela de Configurações da Camada de Enlace DNP3 Servidor

Parâmetros associados:

- **Address (Outstation)** - O campo é utilizado para configurar o endereço DNP3 deste servidor;
- **Address (Master)** - O campo é utilizado para configurar o endereço DNP3 do cliente;
- **IP Address (Master)** - O campo é utilizado para configurar o endereço IP do cliente conectado, utilizado quando a conexão do cliente é por IP;
- **Listen Port** - Endereço da porta listen para conexão do cliente. Utilizado quando a conexão do cliente não é por endereçamento IP;
- **Enable Master Address Validation** - Se essa opção estiver marcada, o DNP3 Servidor só irá aceitar mensagens cujos endereços de origem e destinos coincidem com os valores configurados respectivamente nos campos “Endereço de Origem” e “Endereço de Destino”. Ou seja o campo “Endereço de Origem” da mensagem deve coincidir com o valor do campo “Endereço de Destino”. configurado para o Cliente. E o campo “Endereço de Destino” da mensagem deve coincidir com o valor do campo “Endereço de Origem” configurado pelo cliente;
- **Enable Self-Address** - Habilita a recepção de mensagens com o endereço 65532 do DNP3 Cliente;
- **Sends Confirmed User Data Frames** - Os modos de confirmação da camada de enlace disponíveis são os seguintes:
  - **Never** - Nunca solicitará confirmação das mensagens;
  - **Always** - Sempre solicitará confirmação das mensagens;
  - **Sometimes** - Solicitará confirmação somente para mensagens multifragmentadas;
- **Maximum Retries** - Número de vezes que o DNP3 Servidor irá retransmitir a mensagem caso ele não receba a confirmação do cliente;
- **Confirmation Timeout (s)** - Tempo de espera para enviar uma nova tentativa;

- **Keep Alive (s)** - Intervalo de tempo para envio de mensagens de teste de link ("TCP Keep Alive" ou "Serial Link Status") após última mensagem DNP3 recebida. Caso não seja recebida uma resposta para a mensagem de teste de link, a conexão atual será encerrada. Este mecanismo de teste deve ser corretamente configurado, pois é através dele que problemas de conexão são identificados.

### Configuração da Camada de Aplicação DNP3 Servidor

A figura a seguir mostra a configuração da camada de aplicação DNP3 Servidor.

Figura 3-33. Tela de Configurações da Camada de Aplicação DNP3 Servidor

Parâmetros associados:

- **Allow Multiple Fragmented Responses** - Opção para Habilitar/Desabilitar a Transmissão de Respostas Múltiplos Fragmentos. Se essa opção estiver marcada, o DNP3 Servidor irá permitir a divisão de uma mensagem de aplicação em mensagens menores e enviá-las uma a uma;
- **Request Confirmation of Multiple Fragmented Responses** - Opção para Habilitar/Desabilitar Requisição de Confirmação de Resposta com Múltiplos Fragmentos. Se essa opção estiver marcada, o DNP3 Servidor irá requisitar ao cliente uma confirmação que recebeu a mensagem com o fragmento da mensagem de aplicação;
- **TX Fragmented Length (bytes)** - Tamanho máximo dos fragmentos;
- **Confirmation Timeout (s)** - Determina qual é o tempo máximo de espera para as mensagens de confirmação. O timeout de mensagens de confirmação é o tempo que o servidor aguardará por uma confirmação de recebimento da mensagem pelo cliente;
- **Enable Time Synchronization** - Opção para Habilitar/Desabilitar a solicitação de sincronismo de tempo. Se essa opção estiver marcada, o DNP3 Servidor estará

habilitado a solicitar o sincronismo de tempo do cliente. Esta solicitação é feita na primeira mensagem transmitida ao cliente depois que estourar o tempo configurado no campo "Timeout de Sincronismo";

- **Time Synchronization Command Received in Local Time** - Opção para Habilitar/Desabilitar o tratamento do comando de sincronização em horário local;
- **Synchronization Timeout (s)** - Define o tempo que o relógio local continuará válido após ter recebido uma sincronização de tempo;
- **Use Local Time Instead of UTC Time** - Opção para Habilitar/Desabilitar a estampa de tempo em horário local para eventos;
- **Maximum Time Between Selected and Operate (s)** - Tempo máximo que o comando Select permanecerá ativo aguardando o comando Operate, ou seja, é o período no qual o comando de seleção permanecerá ativo (inicia a contagem a partir da confirmação da recepção do comando de seleção);
- **Transmission Mode of Analog Inputs Events** - Seleciona o modo de transmissão dos eventos de entradas analógicas. Os modos de transmissão dos eventos de entradas analógicas disponíveis são os seguintes:
  - **All Events (SOE)** - São enviados todos os eventos analógicos gerados;
  - **Most Recent Events** - É enviado somente o evento mais recente de cada ponto analógico.

‡ **NOTA:**

Quando a opção de sincronismo de tempo for marcada em mais de um Outstation (parâmetro "Habilitar Sincronismo") os tempos recebidos nos diferentes outstations serão sobrescritos no relógio do sistema em um curto espaço de tempo, podendo causar comportamentos indesejáveis devido a atrasos no tempo de propagação das mensagens e carga do sistema.

#### **Configuração das Mensagens Não Solicitadas DNP3 Servidor**

A figura a seguir mostra a configuração das mensagens não solicitadas DNP3 Servidor.

Figura 3-34. Tela de Mensagens Não Solicitadas DNP3 Servidor

Parâmetros associados:

- **Enable Unsolicited Messages** - Opção para Habilitar/Desabilitar as mensagens não solicitadas;
- **Number of Unsolicited Retries** - Número de vezes que o DNP3 Servidor irá tentar transmitir a mensagem;
- **Confirmation Timeout (s)** - Intervalo a ser aguardado depois de um timeout. Especifica o tempo a ser aguardado depois de ocorrer um timeout para a mensagem não solicitada que foi transmitida. Transcorrido esse intervalo a mensagem será retransmitida. Logo o tempo de retransmissão, será dado pelos parâmetros : “Timeout de Mensagens de Confirmação” + “Timeout”;
- **Try Again (s)** - Intervalo a ser aguardado após atingir o número máximo de tentativas. Após o número máximo de tentativas para transmitir a mensagem ser atingido (ver parâmetro “Number of Unsolicited Retries”), as retransmissões restantes serão baseadas nesse parâmetro de tempo. Sendo assim, o tempo de retransmissão passará a ser dado pelos parâmetros “Time-out de Mensagens de Confirmação” + “Tentar Novamente”;
- **Number of Class 1 Events** - Quantidade mínimo de eventos na Classe 1 para disparar uma mensagem não solicitada;
- **Number of Class 2 Events** - Quantidade mínimo de eventos na Classe 2 para disparar uma mensagem não solicitada;
- **Number of Class 3 Events** - Quantidade mínimo de eventos na Classe 3 para disparar uma mensagem não solicitada;
- **Hold Time After Class 1 Event** - Tempo máximo que um evento pode ficar armazenado na classe 1 sem ser enviado por uma mensagem não solicitada;
- **Hold Time After Class 2 Event** - Tempo máximo que um evento pode ficar armazenado na classe 2 sem ser enviado por uma mensagem não solicitada;

- **Hold Time After Class 3 Event** - Tempo máximo que um evento pode ficar armazenado na classe 3 sem ser enviado por uma mensagem não solicitada.

Os diagnósticos e comandos do protocolo DNP3 Servidor configurado, são armazenados em variáveis do tipo T\_DIAG\_DNP\_SERVER\_1.

#### **Comando Cold Restart**

O comando de Cold Restart força a reinicialização da remota. Os eventos da fila interna não são perdidos, ou seja, os valores das variáveis do tipo retentivas e persistentes não são alterados. Porém os eventos, que por motivo de overflow, estejam apenas nas filas dos drivers, serão perdidos. Uma vez que a remota é reinicializada, todas as configurações realizadas em tempo de execução, específicas de cada protocolo, serão perdidas, e as configurações vigentes serão aquelas originais do projeto, realizadas com o auxílio do software MasterTool.

#### **Comandos para Pontos de Saída e Contadores**

Os comandos recebidos pelo driver DNP3 Servidor podem ter os seguintes destinos:

- Pontos Internos;
- Módulo de saída (por ex. HX2320);
- Driver DNP3 Cliente;
- Interceptação (bloco de função CommandReceiver).

Comandos para congelamento de contadores para uma faixa de pontos são apenas suportados caso todos os pontos contidos na faixa sejam Pontos Internos. Caso um ou mais destes pontos não seja interno (redirecionado para um IED conectado a um driver DNP3 Cliente por exemplo), o comando não será aceito. Nestes casos, o comando de congelamento deve ser executado individualmente para cada ponto.

#### **Pontos Internos**

Pontos internos são considerados aqueles pontos representados por variáveis que não estão associadas a nenhum dos outros destinos descritos acima. Para este tipo de ponto, são suportados apenas comandos para pontos simples. Caso o DNP3 Servidor receba um comando Trip/Close para um ponto interno, será retornado o código de erro 0x04 (Not supported), a menos que seja utilizado o interceptador de comandos.

#### **Módulo de Saída**

Caso a variável mapeada para o ponto de saída DNP3 esteja associada a um cartão de saída (por ex. HX2320), o comando será redirecionado e executado pelo próprio cartão. No caso de comandos trip/close, este redirecionamento funcionará apenas se a variável for do tipo DBP. Caso seja utilizada uma variável do tipo BOOL, o DNP3 Servidor retornará o código de erro 0x03 (wrong format) para comandos pulsados, assim como quando um comando Latched é direcionado a uma variável DBP mapeada no cartão de saída. Com a interceptação e tratamento destes comandos pela aplicação pode evitar o retorno de mensagens negativas (falha).

#### **Driver DNP3 Cliente**

Caso a variável esteja associada a um ponto de um IED em um driver DNP3 Cliente, o comando será redirecionado e enviado para ser executado pelo próprio IED e o valor da variável armazenada na memória da UCP não é atualizado. No caso de comandos Trip/Close, tipicamente a variável deve ser do tipo DBP, contudo o redirecionamento funcionará da mesma forma mesmo que seja utilizada uma variável do tipo BOOL.

Deve-se considerar que, devido às características do protocolo, o driver DNP3 Cliente é capaz de executar somente um comando por vez. Isto é, enquanto um comando para um determinado ponto não for finalizado (resposta ou timeout), não é possível disparar um novo comando, mesmo que seja para um ponto diferente. Desta forma, caso um comando seja redirecionado do DNP3 Servidor para um IED associado a um driver DNP3 Cliente e este último esteja ocupado executando um comando, será retornado o código de erro 0x05 (already active).

#### **Driver IEC 60870-5-104 Cliente**

Caso a variável esteja associada a um ponto de um IED em um driver IEC 60870-5-104 Cliente, o comando será redirecionado e enviado para ser executado pelo próprio IED. No caso de comandos trip/close, tipicamente a variável deve ser do tipo DBP, contudo o redirecionamento funcionará da mesma forma mesmo que seja utilizada uma variável do tipo BOOL.

#### **Interceptação de Comandos**

A interceptação de comandos recebidos pelo driver DNP3 Servidor é realizada através do bloco de função CommandReceiver conforme descrito na seção Interceptação de Comandos Oriundos do Centro de Controle. Este recurso permite a interceptação de comandos de seleção e operação (isto é, funções 3, 4, 5 e 6 do protocolo DNP3).

#### **IEC 60870-5-104 Cliente**

Este protocolo está disponível para a UCP da Série Hadron Xtorm nas suas interfaces Ethernet (NET1 a NET6). Ao selecionar esta opção no MasterTool Xtorm, a UCP passa a ser cliente da comunicação IEC 60870-5-104. O limite máximo de instanciamento de drivers IEC 60870-5-104 Cliente depende de quantos drivers de protocolos estão instanciados e são suportados pelas UCPs HX3040.

O protocolo IEC 60870-5-104 Cliente está ativo somente quando a UCP estiver em modo de execução (RUN).

Para configurar um instância deste protocolo é necessário executar os seguintes passos:

1. Selecionar a interface Ethernet (NET1 a NET6) onde o IEC 60870-5-104 Cliente será instanciado, e configurar esta interface. Para realizar esse procedimento, consultar a seção Configuração das Interfaces Ethernet;
2. Adicionar uma instância do protocolo IEC 60870-5-104 Cliente abaixo da interface Ethernet selecionada. Para realizar esse procedimento, consultar a seção "Inserindo uma Instância de Protocolo";
3. Adicionar e configurar dispositivos Controlled Station abaixo da instância do IEC 60870-5-104 Cliente. Existe um limite máximo de 32 Controlled Stations abaixo de cada instância do IEC 60870-5-104 Cliente;
4. Configurar os parâmetros da camada de enlace de cada dispositivo Controlled Station;
5. Adicionar e configurar setores (Sectors) abaixo do dispositivo Controlled Station. Existe um limite máximo de 5 setores para cada dispositivo;
6. Executar as seguintes configurações para cada setor adicionado:
  - a. Adicionar e configurar os mapeamentos IEC 60870-5-104 do setor;
  - b. Configurar os parâmetros da camada de enlace do setor;
  - c. Configurar os parâmetros da camada de aplicação do setor.

As figuras a seguir ilustram esse procedimento.

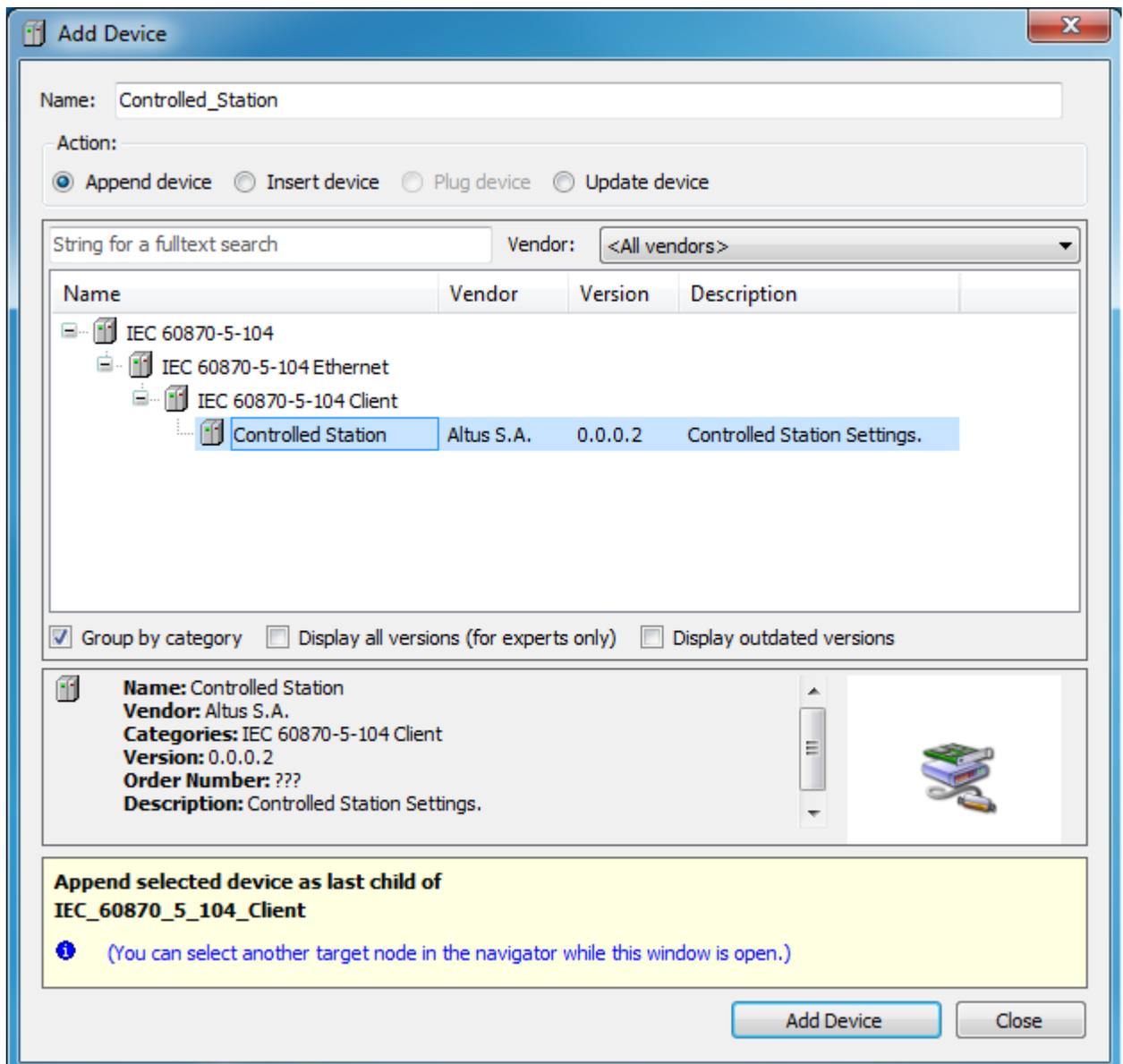


Figura 3-35. Tela de Inserção de dispositivo Controlled Station abaixo de um IEC 60870-5-104 Cliente

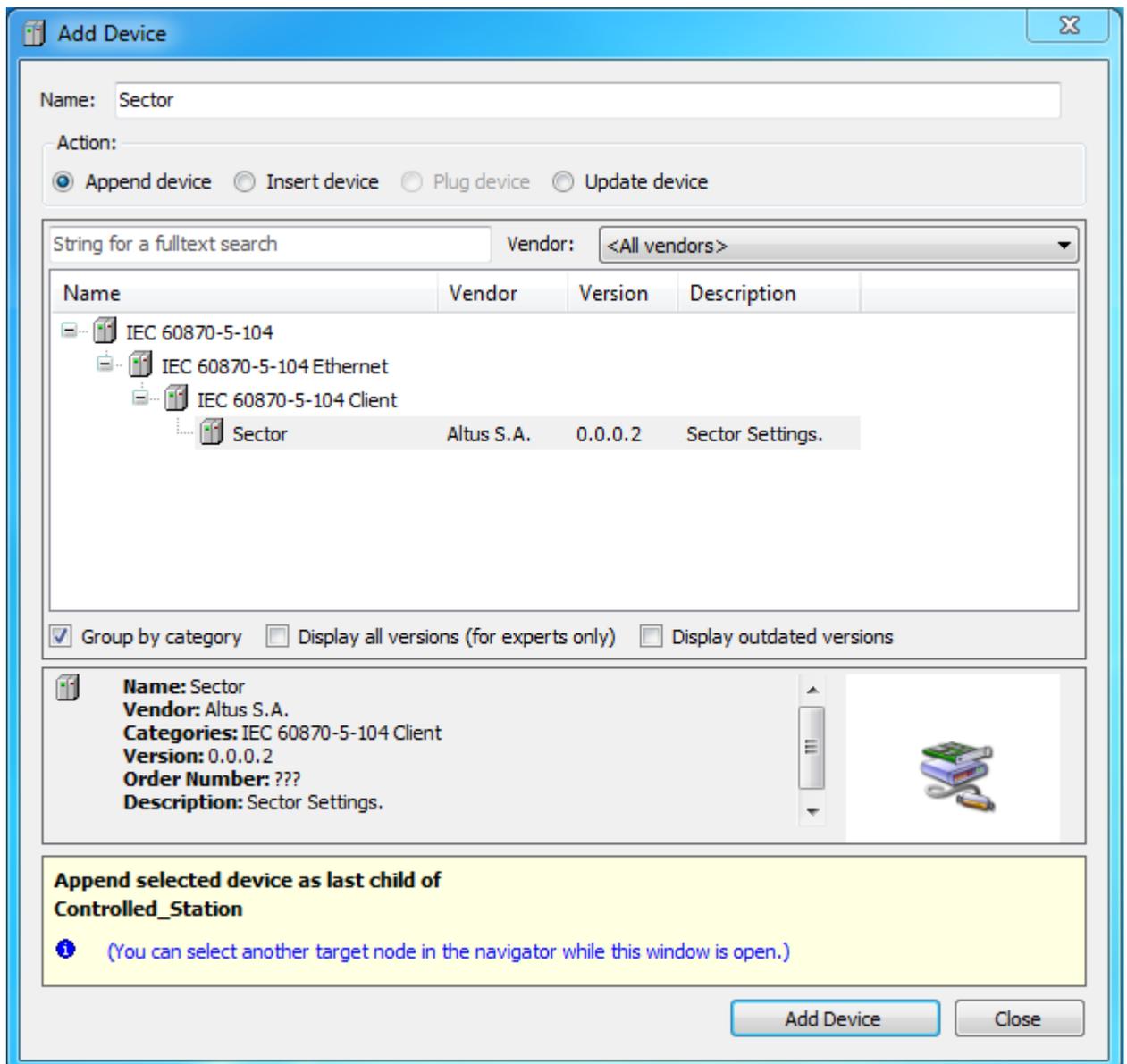
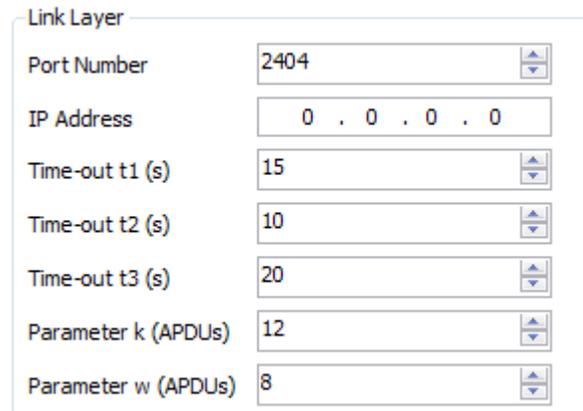


Figura 3-36. Tela de Inserção de setor abaixo de um dispositivo Controlled Station IEC 60870-5-104

### Configuração da Camada de Enlace de uma Controlled Station

A configuração dos parâmetros da camada de enlace de uma Controlled Station, adicionada abaixo de um IEC 60870-5-104 Cliente está mostrada na figura a seguir.



Link Layer	
Port Number	2404
IP Address	0 . 0 . 0 . 0
Time-out t1 (s)	15
Time-out t2 (s)	10
Time-out t3 (s)	20
Parameter k (APDUs)	12
Parameter w (APDUs)	8

Figura 3-37. Tela de Configuração da Camada de Enlace de uma Controlled Station IEC 60870-5-104

Parâmetros associados:

- **Port Number** - Endereço da porta TCP listen para conexão com esta Controlled Station;
- **IP Address** - Endereço IP desta Controlled Station;
- **Time-out t1(s)** - Período (em segundos) que o dispositivo aguarda o recebimento de uma mensagem de confirmação após o envio de uma mensagem APDU do tipo I ou U (dados) antes de fechar a conexão;
- **Time-out t2(s)** - Período (em segundos) que o dispositivo aguarda para enviar uma mensagem supervisória (S-Frame) confirmando o recebimento de frames de dados;
- **Time-out t3(s)** - Período (em segundos) em que será enviada uma mensagem para teste do link caso não haja nenhuma transmissão por ambos os lados;
- **Parameter k (APDUs)** - Número máximo de mensagens de dados (I-Frame) transmitidas e não confirmadas;
- **Parameter w (APDUs)** - Número máximo de mensagens de dados (I-Frame) recebidas e não confirmadas.

### Configuração dos Mapeamentos de um Setor IEC 60870-5-104

A configuração dos mapeamentos de um setor IEC 60870-5-104, adicionado abaixo de uma Controlled Station está mostrada na figura a seguir.

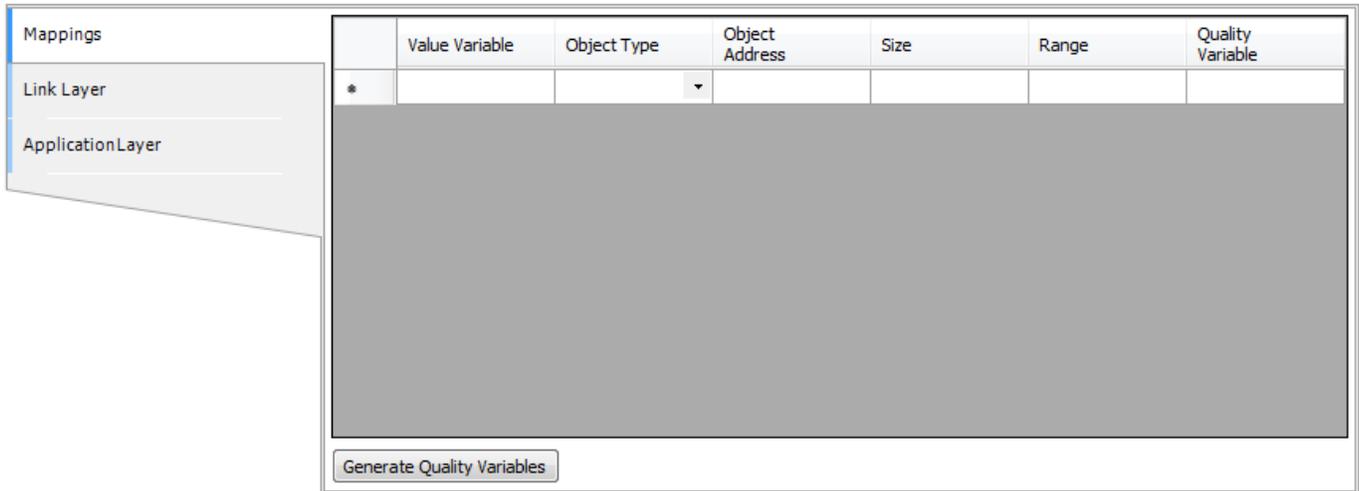


Figura 3-38. Tela de Mapeamentos de Dados de um Setor IEC 60870-5-104

Parâmetros associados:

- **Value Variable** - Nome da variável simbólica a ser mapeada, Quando um comando de leitura é enviado, o retorno enviado na resposta será armazenado nesta variável. Quando for um comando de escrita, o valor escrito será copiado desta variável. A variável pode ser simples, array, elemento de array, elemento simples de estrutura, ou elemento array de estruturas. A variável simbólica só pode ser mapeada em um setor e não deve estar mapeada em nenhum outro protocolo cliente;
- **Object Type** - Seleciona o tipo de objeto IEC 60870-5-104 vinculado ao mapeamento;
- **Object Address** - O campo é utilizado para configurar o endereço do primeiro ponto do mapeamento no protocolo IEC 60870-5-104;
- **Size** - Especifica a quantidade máxima de dados que um mapeamento IEC 60870-5-104 poderá acessar. É um campo somente leitura que informa quantos pontos estão sendo usados por esse mapeamento. O valor deste campo é calculado automaticamente conforme o tipo da variável configurada no mapeamento;
- **Range** - Faixa de endereços do dado configurado. É um campo somente de leitura que informa a faixa de endereços que está sendo usada por esse mapeamento. O valor deste campo é calculado automaticamente pela soma dos campos "Object Address" e "Size". Não pode haver sobreposições de faixa com outros mapeamentos;
- **Quality Variable** - Nome da variável simbólica que armazenará a qualidade do mapeamento, sendo ela sempre do tipo QUALITY, disponível a partir da biblioteca LibDataTypes. A variável pode ser simples, array ou elemento de array e pode estar em estruturas. A variável simbólica só pode ser mapeada em um setor e não deve estar mapeada em nenhum outro protocolo cliente. Não é permitido mapear variáveis de qualidade para os tipos de comando IEC 60870-5-104;
- **Generate Quality Variable** - Este botão gera automaticamente na GVL Qualities variáveis do tipo QUALITY para cada Variável de Valor declarada. Para mais informações acesse o capítulo Conversões de Qualidade.

🔔 **NOTA:**

O driver IEC 60870-5-104 Cliente não gera eventos baseados nos valores de seus mapeamentos. Os eventos devem ser gerados pelo respectivo IED. Sendo assim, caso o IED não tenha suporte a geração de eventos, o driver IEC 60870-5-104 Cliente não será capaz de gerá-los.

### Configuração da Camada de Enlace de um Setor IEC 60870-5-104

A configuração dos parâmetros da camada de enlace de um setor IEC 60870-5-104, adicionado abaixo de uma Controlled Station está mostrada na figura a seguir. O parâmetro configurado é o Common Address of ASDU equivalente ao Endereço IEC 60870-5-104 deste setor.

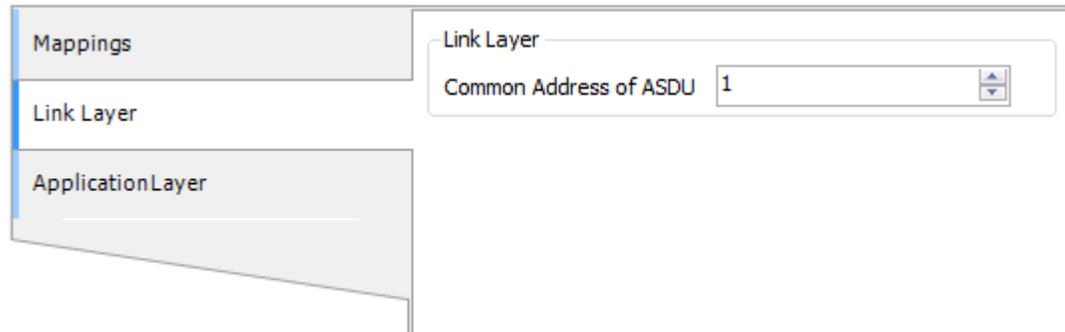


Figura 3-39. Tela de Configurações da Camada de Enlace de um Setor IEC 60870-5-104

### Configuração da Camada de Aplicação de um Setor IEC 60870-5-104

A configuração dos parâmetros da camada de aplicação de um setor IEC 60870-5-104, adicionado abaixo de uma Controlled Station está mostrada na figura a seguir.

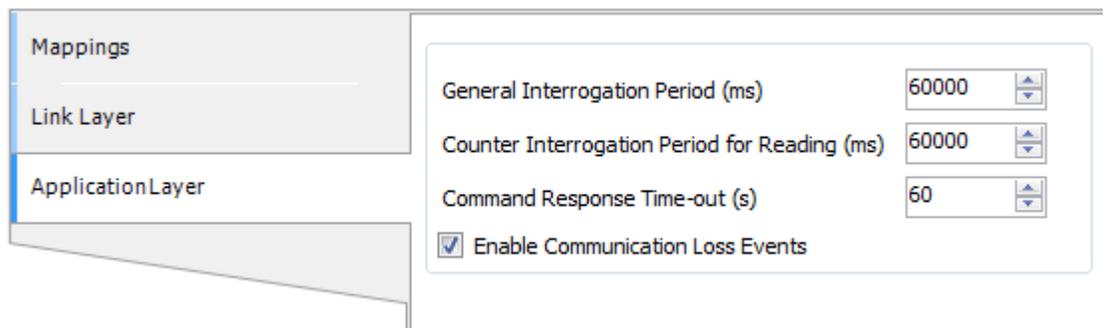


Figura 3-40. Tela de Configurações da Camada de Aplicação IEC 60870-5-104 Cliente  
Parâmetros associados:

- **General Interrogation Period (ms)** - Período para execução do comando “General Interrogation” que faz a leitura de todos os pontos mapeados (exceto contadores). O valor especial 0 desabilita a execução do comando;
- **Counter Interrogation Period for Reading (ms)** - Período para execução do comando “Counter Interrogation” para a leitura de todos os contadores mapeados. O valor especial 0 desabilita a execução do comando;
- **Command Response Time-out (s)** - Tempo máximo para recebimento da resposta de um comando. Caso esse tempo seja excedido, é incrementado o diagnóstico contador de erros de comando;
- **Enable Communication Loss Events** - Habilita a geração de eventos de perda de comunicação com as Controlled Stations.

Os diagnósticos do protocolo IEC 60870-5-104 Cliente configurado, são divididos em duas estruturas, uma para o Cliente e outra para cada Controlled Station declarada abaixo deste. Os diagnósticos do cliente são armazenados em variáveis do tipo T\_DIAG\_IEC104\_CLIENT\_1. Os diagnósticos para Controlled Station são armazenados em variáveis do tipo T\_DIAG\_IEC104\_CONTROLLED\_STATION\_1.

#### **Comandos IEC 60870-5-104 para IEDs**

Nesta seção são listados os comandos de usuário disponíveis para serem enviados via IEC 60870-5-104 para IEDs. Tais comandos são executados com o auxílio de blocos funcionais, descritos a seguir.

Devido às características do protocolo IEC 60870-5-104 Cliente, não é possível enviar comandos simultâneos de usuário para um mesmo Setor de um IED, mesmo que destinados a pontos diferentes do Setor. Desta forma, ao ser enviado um comando para um Setor de um IED, o próximo comando para este mesmo Setor de IED só será enviado após o término da execução do primeiro comando.

Então, caso um comando seja disparado pela aplicação do usuário enquanto houver outro comando em execução para o mesmo Setor do IED, o novo comando aguardará o término do primeiro por um período definido no bloco funcional (udiCommandTimeOut). E se o tempo para o fim da execução do primeiro comando for maior que udiCommandTimeOut, o novo comando retornará uma mensagem de erro (IEC104\_COMMAND\_ERROR\_STATUS\_TIMEOUT). Ou seja, um comando começa a contabilizar o tempo de timeout desde o momento em que foi disparado pela aplicação do usuário, e não somente ao ser enviado pelo IEC 60870-5-104 Cliente.

#### **Comandos para Saídas Digitais**

Os comandos para as saídas digitais são executados com o bloco funcional IEC104\_DigitalCommand. Este bloco funcional permite executar comandos sobre os seguintes tipos de objetos:

- Single Command (C\_SC\_NA);
- Double Command (C\_DC\_NA);
- Regulating Step Command (C\_RC\_NA).

#### **Comando para Saídas Analógicas**

Os comandos para as saídas digitais são executados com os blocos funcionais IEC104\_AnalogCommandInt e IEC104\_AnalogCommandReal. O bloco funcional IEC104\_AnalogCommandInt permite executar comandos sobre os seguintes tipos de objetos:

- Setting Point Command, normalized Value (C\_SE\_NA );
- Setting Point Command, scaled Value (C\_SE\_NB ).

O bloco funcional IEC104\_AnalogCommandReal permite executar comandos sobre o seguinte tipo de objeto: Setting Point Command, short floating point Value (C\_SE\_NC).

#### **Comando Reset Process**

O bloco funcional IEC104\_ResetProcess permite reinicializar um dispositivo IEC 60870-5-104 Servidor.

#### **Comando SynchronizeIED**

Através do bloco funcional IEC104\_SynchronizeIED é possível comandar um sincronismo do relógio de um IED (IEC 60870-5-104 Servidor).

#### **Comando General Interrogation**

Na configuração da camada de aplicação do IEC 60870-5-104 Cliente é possível configurar a execução periódica de comandos General Interrogation apenas para leitura de todos os pontos

mapeados (exceto contadores). Esse comando também pode ser executado através do bloco funcional IEC104\_GeneralInterrogation.

#### **Comando Counter Interrogation**

Na configuração da camada de aplicação do IEC 60870-5-104 Cliente é possível configurar a execução periódica de comandos Counter Interrogation apenas para leitura dos contadores. Através do bloco funcional IEC104\_CounterInterrogation é possível executar outras opções do comando "Counter Interrogation" sobre contadores:

- Reed;
- Freeze;
- Reset;
- Reset e Freeze.

#### **Códigos de Erro para Comandos IEC 60870-5-104**

Os comandos IEC 60870-5-104 retornam status de erro (eErrorStatus) ao serem executados. Consulte a documentação da série para detalhamento dessa funcionalidade.

#### **IEC 60870-5-104 Servidor**

Este protocolo está disponível para a UCP da Série Hadron Xtorm nos seus canais Ethernet. Ao selecionar esta opção no MasterTool Xtorm, a UCP passa a ser servidor da comunicação IEC 60870-5-104, permitindo a conexão com até cinco dispositivos clientes IEC 60870-5-104. Para cada cliente, o driver possui uma fila de eventos exclusiva com as seguintes características:

- Tamanho: 4500 eventos;
- Política de estouro: mantém o mais recente.

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

1. Adicionar a instância do protocolo IEC 60870-5-104 Servidor a um dos canais Ethernet disponíveis (NET 1 ao NET 6). Para realizar esse procedimento, consultar a seção "Inserindo uma Instância de Protocolo";
2. Configurar a interface Ethernet. Para realizar esse procedimento, consultar a seção "Configuração das Interfaces Ethernet";
3. Configurar os parâmetros gerais do protocolo IEC 60870-5-104 Servidor, 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.

#### **Parâmetros Gerais**

A figura a seguir mostra as configurações dos Parâmetros Gerais do IEC 60870-5-104 Servidor.

The screenshot shows a 'Settings' dialog box with two fields: 'Connection Mode' is a dropdown menu currently showing 'Port', and 'TCP Port' is a text input field containing the number '2404'.

Figura 3-41. Tela de Parâmetros Gerais do IEC 60870-5-104 Servidor

Parâmetros associados:

- **Connection Mode** - Configura o modo de conexão com os módulos Cliente Conectados;
- **TCP Port** - Define qual o número da porta TCP da UTR será utilizada para a comunicação com os módulos Cliente Conectados, caso o campo "Modo de Conexão" esteja selecionado "IP".

### Mapeamentos dos Dados

A figura a seguir mostra a configuração das relações de dados do IEC 60870-5-104 Servidor.

Mappings	Value Variable	Object Type	Object Address	Size	Range	Counter Variable	Dead Band Variable	Dead Band Type	Select Required	Short Pulse (ms)	Long Pulse (ms)
Link Layer	*										
Application Layer											

Figura 3-42. Tela de Mapeamento dos Dados do IEC 60870-5-104 Servidor

Parâmetros associados:

- **Value Variable** - Nome da variável simbólica a ser mapeada. Quando um comando de leitura é enviado, o retorno enviado na resposta está armazenado nesta variável. Quando for um comando de escrita, o valor escrito será armazenado nesta variável. A variável pode ser simples, array ou elemento de array e pode estar em estruturas;
- **Object Type** - Configuração do tipo do objeto IEC 60870-5-104;
- **Object Address** - Índice do primeiro ponto do mapeamento IEC 60870-5-104;
- **Size** - Especifica a quantidade máxima de dados que um mapeamento IEC 60870-5-104 poderá acessar;
- **Range** - Faixa de endereços do dado configurado;
- **Counter Variable** - Nome da variável simbólica que vai conter o dado da variável contadora. Este campo se aplica somente no mapeamento de objetos do tipo Integrated Totals, sendo esta última, a variável contadora, a ser manipulada no processo. Ela deve ter o mesmo tipo e tamanho da variável mapeada na coluna Value Variable, cujo valor será lido pelo cliente ou reportado ao cliente em caso de eventos. Quando a Counter Variable tem associada uma variável de qualidade, para que esta qualidade seja transferida para a variável congelada no comando de freeze, deve ser associada uma variável de qualidade à variável congelada. Este procedimento pode ser feito através da aba Internal Points;
- **Dead Band Variable** - Nome da variável simbólica que vai conter o dado da banda morta. Este campo se aplica somente à mapeamento de variáveis analógicas de entrada (objetos tipo Measured Value). Ela deve ter o mesmo tipo e tamanho da variável mapeada na coluna Value Variable. Novos valores da variável de banda morta serão considerados somente quando a variável analógica de entrada mudar de valor. A banda morta configurada para variáveis mapeadas no IEC 60870-5-104 Servidor não é aplicada quando estes pontos forem oriundos de um IED (através do driver DNP3 Cliente, IEC 61850 GOOSE Subscriber etc.), pois neste caso a detecção dos eventos ocorre no próprio IED;

- **Dead Band Type** - Define o tipo de Banda Morta a ser utilizada no mapeamento. Os tipos de configuração da Banda Morta disponíveis são os seguintes:
- **Desabilitada** - Nesta opção, qualquer modificação de valor num ponto do grupo, por menor que seja, gera evento para este ponto;
- **Absoluta** - Nesta opção, se o módulo da variação do valor de um ponto do grupo for maior do que o valor configurado pela variável no campo “Banda Morta”, um evento para este ponto é gerado;
- **Integrada** - Nesta opção, se o módulo da integral da variação do valor de um ponto do grupo for maior do que o valor configurado pela variável no campo “Banda Morta”, um evento para este ponto é gerado. O intervalo de integração é de um segundo;
- **Select Required** - Define se para a execução de um comando é requerido ou não um Select prévio;
- **Short Pulse** - Define o tempo do pulso curto de um comando digital IEC 60870-5-104, em milissegundos;
- **Long Pulse** - Define o tempo do pulso longo de um comando digital IEC 60870-5-104, em milissegundos.

🔗 **NOTA:**

Ao definir o tempo de duração dos pulsos curtos e longos deve se levar em consideração os limites suportados pelo dispositivo que irá tratar o comando. Por exemplo, caso o destino seja um cartão de saída, deve se verificar na CT do módulo quais os tempos mínimos e máximos, bem como a resolução, para a execução de comandos pulsados.

### Camada de Enlace

A figura a seguir mostra a configuração dos parâmetros da camada de enlace do IEC 60870-5-104 Servidor.

Link Layer	
Port Number	2404
IP Address	0 . 0 . 0 . 0
Common Address of ASDU	1
Time-out t1 (s)	15
Time-out t2 (s)	10
Time-out t3 (s)	20
Parameter k (APDUs)	12
Parameter w (APDUs)	8

Figura 3-43. Tela de Configurações da Camada de Enlace IEC 60870-5-104 Servidor

- **Port Number** - Endereço da porta listen para conexão do cliente. Utilizado quando a conexão do cliente não é por IP;
- **IP Address (Master)** - IP do cliente conectado, utilizado quando a conexão do cliente é por IP Common Address of ASDU Endereço IEC 60870-5-104 do servidor;

- **Timeout t1(s)** - Período (em segundos) que o dispositivo aguarda o recebimento de uma mensagem de confirmação após o envio de uma mensagem APDU do tipo I ou U (dados) antes de fechar a conexão;
- **Timeout t2(s)** - Período (em segundos) que o dispositivo aguarda para enviar uma mensagem supervisória (S-Frame) confirmando o recebimento de frames de dados;
- **Timeout t3(s)** - Período (em segundos) em que será enviada uma mensagem para teste do link caso não haja nenhuma transmissão por ambos os lados;
- **Parameter k (APDUs)** - Número máximo de mensagens de dados (I-Frame) transmitidas e não confirmadas;
- **Parameter w (APDUs)** - Número máximo de mensagens de dados (I-Frame) recebidas e não confirmadas.

🔔 **NOTA:**

Os campos "Time-out t1 (s)", "Timeout t2 (s)" e "Timeout t3 (s)" são dependentes entre si e devem ser configurados de tal forma que "Timeout t1 (s)" seja maior do que "Time-out t2 (s)" e "Timeout t3 (s)" seja maior do que "Time-out t1 (s)". Se alguma destas regras não for respeitada, mensagens de erro serão exibidas na compilação do projeto.

### Camada de Aplicação

A figura a seguir mostra a configuração dos parâmetros da camada de aplicação do IEC 60870-5-104 Servidor.

Figura 3-44. Tela de Configurações da Camada de Aplicação do IEC 60870-5-104 Servidor

Parâmetros associados:

- **Enable Time Synchronization** - Opção para Habilitar/Desabilitar a solicitação de sincronismo de tempo. Uma vez habilitado, permite que o IEC 60870-5-104 Servidor ajuste o relógio da UCP ao receber um comando de sincronização. Quando a opção de sincronismo de tempo for marcada em mais de um servidor, os tempos recebidos nos diferentes servidores serão sobrescritos no relógio do sistema em um curto espaço de tempo, podendo causar comportamentos indesejáveis devido a atrasos no tempo de propagação das mensagens e carga do sistema;

- **Time Synchronization Command Received in Local Time** - Opção para Habilitar/Desabilitar o tratamento do comando de sincronização em horário local. Quando habilitado, o IEC 60870-5-104 Servidor ajusta o relógio da UCP tratando o horário recebido no comando de sincronização como horário local. Caso contrário esse horário é considerado UTC;
- **Use Local Time Instead of UTC Time** - Opção para Habilitar/Desabilitar a estampa de tempo em horário local para eventos. Uma vez habilitado, a estampa de tempo dos eventos gerados pelo IEC 60870-5-104 Servidor será enviada conforme o horário local da UCP;
- **Maximum Time Between Select and Operate (s)** - Período no qual o comando de seleção permanecerá ativo (inicia a contagem a partir da confirmação da recepção do comando de seleção) aguardando o comando de Operate.
- **Maximum Time to Complete a Command Execution (s)** - Tempo máximo para tratar um comando quando transferido para um driver de comunicação Cliente ou mesmo um cartão de saída, antes que seja respondido falha por timeout. Este parâmetro não se aplica quando o comando é interceptado pelo bloco de função de interceptação de comandos (CommandReceiver). Neste caso o tempo máximo para processar o comando é definido por um parâmetro do bloco de função;
- **Transmission Mode of Analog Inputs Events** - Modo de transmissão dos eventos de entradas analógicas. Os modos são os seguintes:
  - **All Events(SOE)** - São enviados todos os eventos analógicos gerados;
  - **Most Recent Events** - É enviado somente o evento mais recente de cada ponto analógico;
- **Transmission Mode** - Modo de transmissão dos dados dos contadores. Os modos de transmissão dos contadores congelados (Integrated Totals) disponíveis são os seguintes:
  - **Freeze by Counter Interrogation Command, Transmit Spontaneously** - Equivale ao Modo D de aquisição de contadores (Integrated Totals) definido pela norma IEC 60870-5-104. Neste modo, comandos de interrogação de contadores (Freeze) da estação de controle realizam o congelamento dos contadores. Caso os valores congelados tenham se modificado, eles são reportados através de eventos;
  - **Freeze and Transmit by Counter Interrogation Command** - Equivale ao Modo C de aquisição de contadores (Integrated Totals) definido pela norma IEC 60870-5-104. Neste modo, comandos de interrogação de contadores (freeze) da estação de controle realizam o congelamento dos contadores. Comandos subsequentes de interrogação dos contadores (read) são enviados pela estação de controle para obter os valores congelados.

📌 **NOTA:**

A norma IEC 60870-5-104, seção “Transmission Control using Start/Stop”, prevê a utilização de comandos STARTDT e STOPDT para controle do tráfego de dados entre o cliente e o servidor, utilizando conexões simples ou múltiplas conexões. Apesar da Xtorm suportar tais comandos, a utilização dos mesmos não é recomendada para controlar a transmissão dos dados, principalmente com UCPs redundantes, pois tais comandos não são sincronizados entre as duas UCPs. Em vez de utilizar múltiplas conexões entre o cliente e o servidor Xtorm, sugere-se a utilização do recurso de NIC Teaming para prover canais Ethernet (fisicamente) redundantes e preservar os recursos da UCP (centros de controle por UCP).

#### Diagnósticos do Servidor

Os diagnósticos e comandos do protocolo IEC 60870-5-104 Servidor configurado, são armazenados em variáveis do tipo T\_DIAG\_IEC104\_SERVER\_1.

#### Comandos para Pontos de Saída

Os comandos recebidos pelo driver IEC 60870-5-104 Servidor podem ter os seguintes destinos:

- **Pontos Internos** - Pontos internos são considerados aqueles pontos representados por variáveis que não estão associadas a nenhum dos outros destinos descritos acima. Para este tipo de ponto, não são suportados comandos pulsados. Caso o IEC 60870-5-104 Servidor receba um comando pulsado (simples ou duplo) para um ponto interno, será retornado um código de falha (confirmação negativa), a menos que seja utilizado o interceptador de comandos. Comandos destinados às variáveis mapeadas em um cartão de saída da Série Hadron Xtorm (por exemplo um módulo HX2320) são tratados como comandos para Pontos Internos;
- **Módulo de Saída** (por ex. HX2320, HX2300) - Caso a variável mapeada para o ponto de saída IEC 60870-5-104 esteja associada a um cartão de saída (por ex. HX2320) da Série Xtorm, o comando será redirecionado e executado pelo próprio cartão. No caso de comandos pulsados, este redirecionamento funcionará apenas se a variável for do tipo DBP. Caso seja utilizada uma variável do tipo BOOL, o servidor retornará uma mensagem negativa de confirmação (falha) para comandos pulsados, assim como quando um comando persistente é direcionado a uma variável DBP mapeada no cartão de saída. Com a interceptação e tratamento destes comandos pela aplicação pode evitar o retorno de mensagens negativas (falha);
- **Driver DNP3 Cliente** - Caso a variável esteja associada a um ponto de um IED em um driver DNP3 Cliente, o comando será redirecionado e enviado para ser executado pelo próprio IED e o valor da variável armazenada na memória da UCP não é atualizado. Importante observar que o DNP3 Cliente não suporta comandos persistentes para pontos duplos, neste caso é retornada uma mensagem negativa de confirmação. Deve-se considerar que, devido às características do protocolo, o driver DNP3 Cliente é capaz de executar somente um comando por vez. Isto é, enquanto um comando para um determinado ponto não for finalizado (resposta ou timeout), não é possível disparar um novo comando, mesmo que seja para um ponto diferente. Desta forma, caso um comando seja redirecionado do IEC 60870-5-104 Servidor ou mesmo do DNP3 Servidor para um IED associado a um driver DNP3 Cliente e este último esteja ocupado executando um comando, será retornada uma mensagem negativa de confirmação;
- **Driver IEC 60870-5-104 Cliente** - Caso a variável esteja associada a um ponto de um IED em um driver IEC 60870-5-104 Cliente, o comando será redirecionado e enviado para ser executado pelo próprio IED. No caso de comandos trip/close, tipicamente a variável deve ser do tipo DBP, contudo o redirecionamento funcionará da mesma forma mesmo que seja utilizada uma variável do tipo BOOL
- **Interceptação (bloco CommandReceiver)** - A interceptação de comandos recebidos pelo driver IEC 60870-5-104 Servidor é realizada através do bloco CommandReceiver conforme descrito na seção Interceptação de Comandos Oriundos do Centro de Controle. Este recurso permite a interceptação de comandos de seleção e execução.

#### Qualificador dos Comandos

A norma IEC 60870-5-104 prevê quatro diferentes qualificadores de comandos para os objetos Single Command, Double Command e Regulating Step, todos suportados pelo Servidor da Xtorm. Cada tipo de objeto tem um comportamento específico para cada qualificador de comando, como pode ser observado na tabela apresentada a seguir.

Qualificador	Tipo do objeto do protocolo IEC 60870-5-104		
	Single Command	Double Command	Regulating Step Command
No additional definition (padrão)	Mesmo comportamento do qualificador persistente.	Mesmo comportamento do qualificador de pulso curto.	Mesmo comportamento do qualificador de pulso curto.
Short pulse duration (pulso curto)	Requer interceptação do comando para tratamento pela aplicação, caso contrário irá retornar uma mensagem de confirmação negativa (falha).	A saída correspondente ao comando (ON ou OFF) será ligada pelo tempo correspondente à duração do pulso (curto ou longo) configurado para o ponto.	A saída correspondente ao comando (LOWER ou HIGHER) será ligada pelo tempo correspondente à duração do pulso (curto ou longo) configurado para o ponto.
Long pulse duration (pulso longo)			
Persistent output (persistente)	A saída será ligada ou desligada, e assim permanecerá até novo comando, conforme valor (ON ou OFF) comandado pelo cliente.	Requer interceptação do comando para tratamento pela aplicação, caso contrário irá retornar uma mensagem de confirmação negativa (falha).	Requer interceptação do comando para tratamento pela aplicação, caso contrário irá retornar uma mensagem de confirmação negativa (falha).

Tabela 3-1. Qualificador dos Comandos do IEC 60870-5-104 Servidor

## IEC 61850 Servidor

Este protocolo está disponível para a UCP da Série Hadron Xtorm nos seus canais Ethernet. Ao selecionar esta opção no MasterTool Xtorm, a UCP passa a ser servidor da comunicação IEC 61850, permitindo a conexão com clientes IEC 61850 através do protocolo MMS e possibilitando o envio e recepção de mensagens rápidas através do protocolo GOOSE.

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

1. Adicionar a instância do protocolo IEC 61850 Servidor a um dos canais Ethernet disponíveis (NET 1 ao NET 6). Para realizar esse procedimento, consultar a seção "Inserindo uma Instância de Protocolo";
2. Configurar a interface Ethernet. Para realizar esse procedimento, consultar a seção "Configuração das Interfaces Ethernet";
3. Adicionar um Logical Device;
4. Adicionar os Logical Nodes desejados, incluindo objetos e atributos opcionais caso necessário;
5. Criar e configurar Datasets, selecionando elementos dos Logical Nodes previamente configurados;
6. Criar e configurar Reports (buferizados e não-buferizados), utilizando os Datasets definidos anteriormente;
7. Criar e configurar o envio de mensagens GOOSE (GOOSE Publisher), utilizando os Datasets definidos anteriormente;
8. Configurar a recepção de mensagens GOOSE (GOOSE Subscriber) através da importação de arquivos .CID;
9. Configurar o mapeamento de pontos de comunicação da UTR para atributos dos Logical Nodes.

## Implementação do Modelo de Dados IEC 61850

Uma das principais diferenças da norma IEC 61850 quando comparada aos outros protocolos conhecidos (DNP3, MODBUS etc.) está no modelo de dados utilizado para comunicação. Na IEC 61850, os dados são definidos sob a forma de grandes estruturas de dados simbólicas que representam um elemento físico ou lógico presente no sistema. Estas estruturas são chamadas de Logical Nodes, os quais são formados por Objetos que, por sua vez, possuem um conjunto de Atributos. Por fim, os atributos podem ser simples ou estruturados. A figura a seguir ilustra a

estrutura hierárquica deste modelo de dados. Os Logical Nodes são representados pelos símbolos na cor laranja, enquanto os objetos possuem a cor verde e os atributos a cor azul.

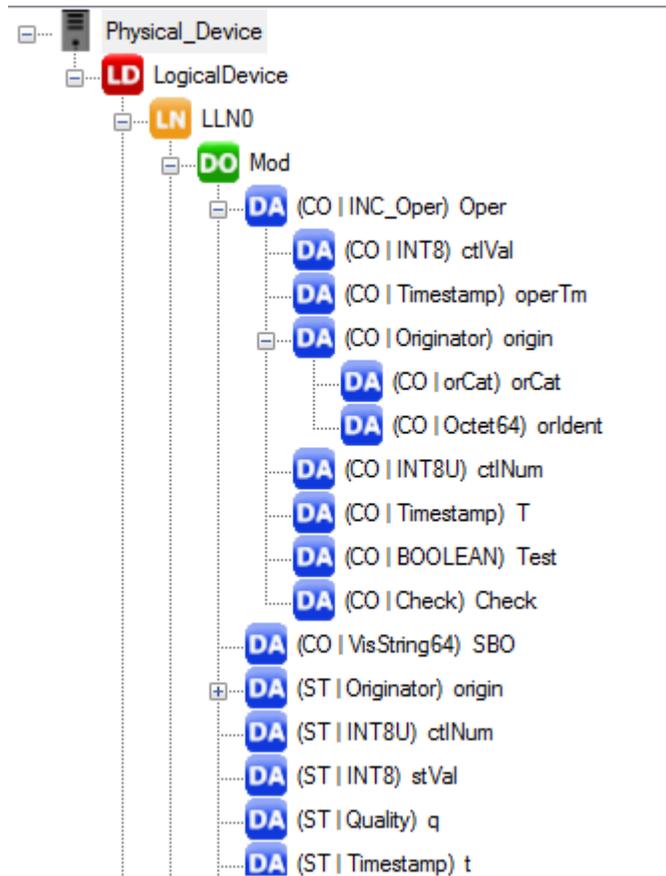


Figura 3-45. Modelo de dados estruturado

Apesar de serem representados sob a forma de uma estrutura de dados, os Logical Nodes muitas vezes possuem um algoritmo associado, o qual pode ser executado em um IED específico ou também de forma distribuída entre vários IEDs.

A implementação desta estrutura de dados IEC 61850 no ambiente de programação IEC 61131-3 do MasterTool Xtorm é realizada através de POU's do tipo Bloco Funcional. Os Logical Devices, Logical Nodes e os Objetos são representados por um Bloco Funcional, enquanto os Atributos tornam-se variáveis simbólicas de tipos elementares conforme a IEC 61131-3. Todos estes dados são declarados em uma lista de variáveis globais (GVL) criada automaticamente pela ferramenta MasterTool Xtorm e são armazenadas na pasta "IEC61850 Generated POU's" localizada na árvore do projeto.

A figura a seguir mostra um exemplo das POU's geradas a partir de uma configuração IEC 61850.

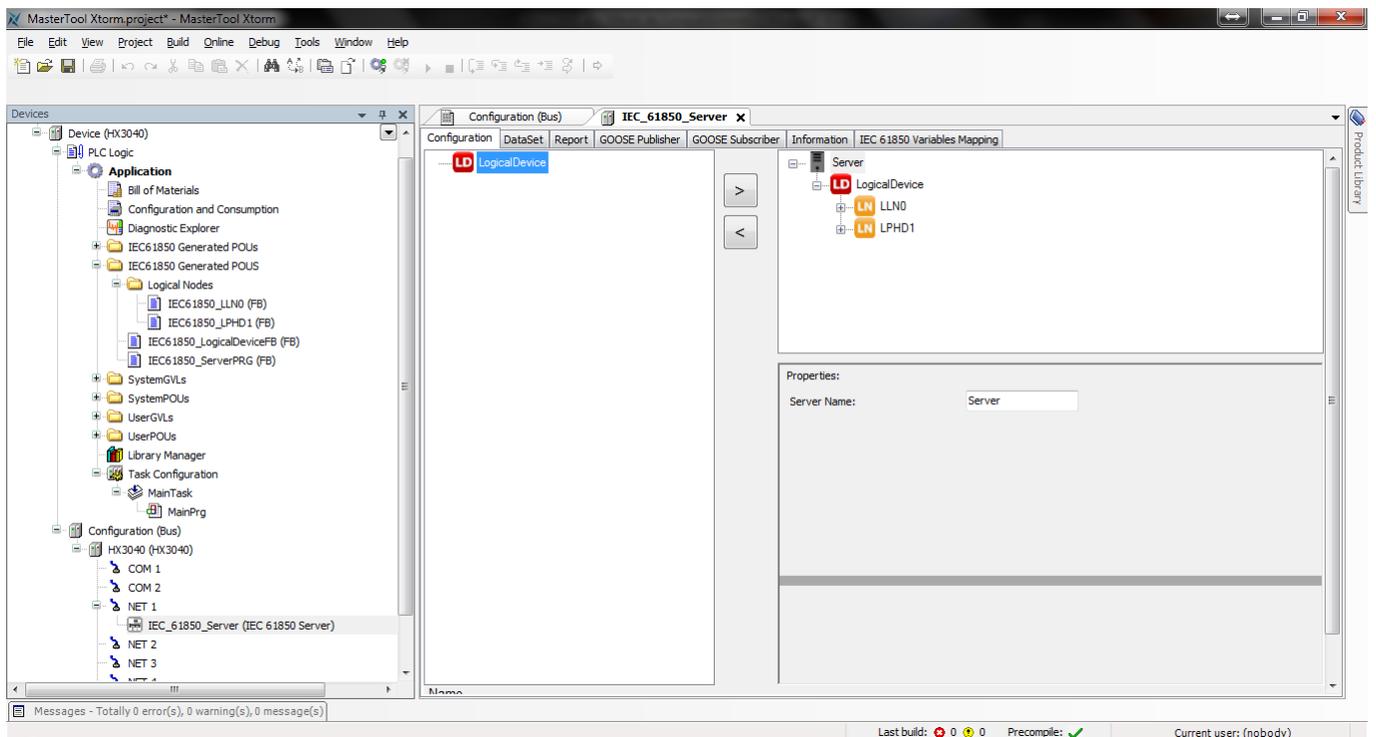


Figura 3-46. Implementação do modelo de dados

Para cada instância de um Logical Node presente na configuração IEC 61850, é criado um tipo correspondente de Bloco Funcional (armazenado na pasta “Logical Nodes”). Isto é necessário pois cada Logical Node pode ser customizado através da inclusão de Objetos e Atributos opcionais. Cada Atributo torna-se uma variável simbólica no ambiente de programação IEC 61131-3. Os tipos de dados enumerados possuem um conjunto de elementos cujos valores recebem um nome.

Em função da flexibilidade de tamanho dos Atributos do tipo STRING, a implementação destes elementos no ambiente IEC 61131-3 é realizada de uma forma diferente de todos os outros. Ao invés de armazenar diretamente o conjunto de caracteres da string, a variável simbólica que representa o atributo armazena apenas uma referência para uma tabela onde se encontra de fato a string propriamente. Desta forma, todos os atributos do tipo string possuem um sufixo “\_REF”. Esta referência é na verdade o índice para acesso a uma tabela de strings implementada sob a forma de uma GVL. Existem no total 7 tabelas de string separadas por tipo.

Por exemplo, vejamos a configuração mostrada na figura a seguir.

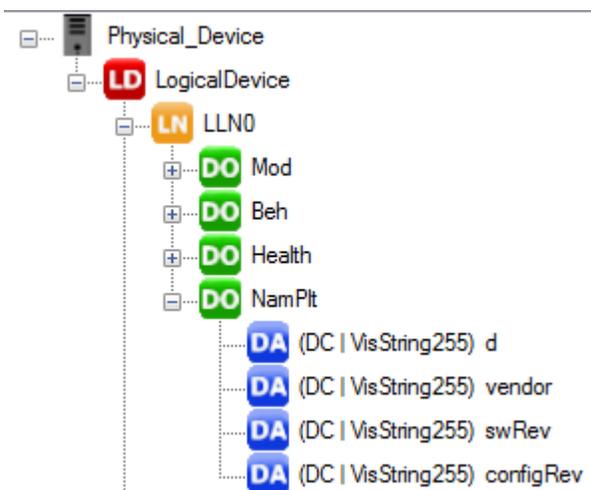


Figura 3-47. Exemplo de configuração com atributos do tipo STRING

Neste exemplo, o Atributo “d” do Objeto NamPlt é do tipo VisString255 e a sua variável simbólica correspondente é:

```
gfbIEC61850_LogicalDevice.LLN0.fbNamPlt.d_REF
```

Desta forma, para atribuir o valor “TEST” a este atributo em linguagem ST:

```
gaVisSTRING255[gfbIEC61850_LogicalDevice.GGIO4.fbNamPlt.d_REF] := 'TEST';
```

### Configuração da Base de Dados

A configuração da Base de Dados do protocolo IEC 61850 Servidor é realizada através da aba Configuration a partir da inclusão de um Logical Device e de seus respectivos Logical Nodes conforme ilustrado na figura a seguir.

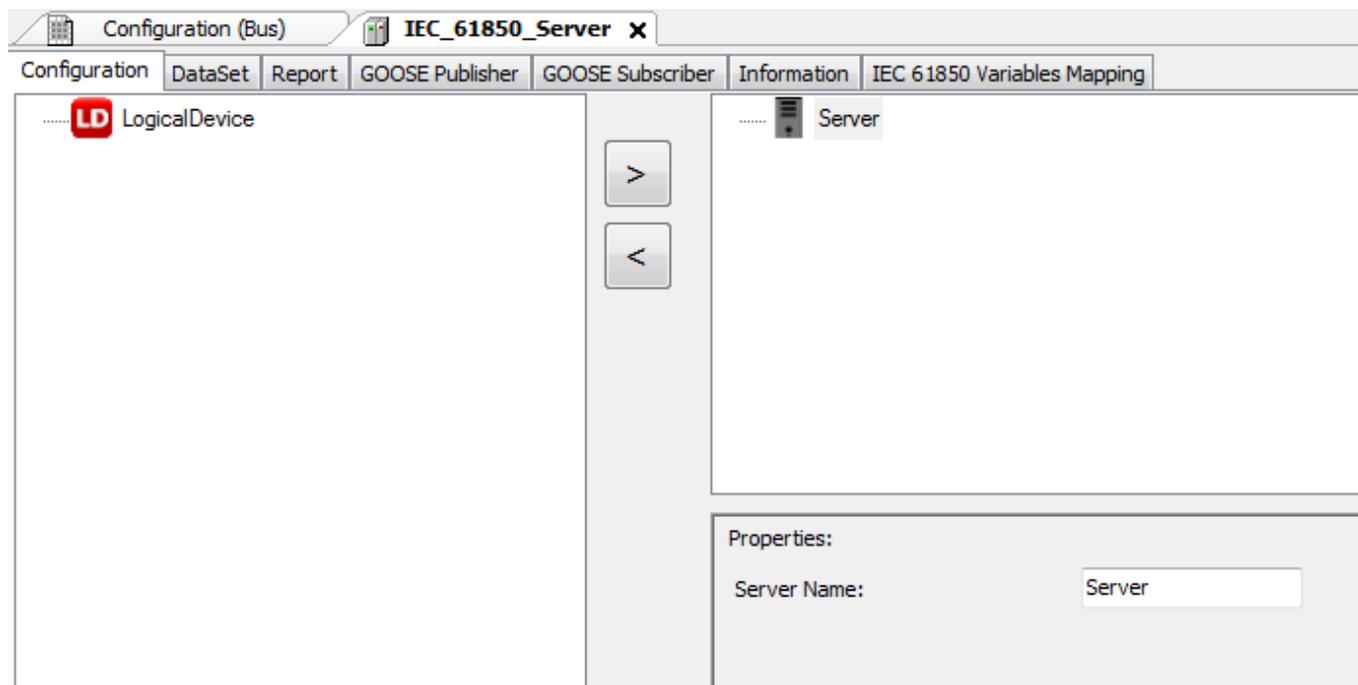


Figura 3-48. Configurando o nome do Physical Device

Esta tela é dividida em duas partes. O lado direito representa a configuração atual do dispositivo. No lado esquerdo encontra-se a lista de objetos disponíveis que podem ser adicionados/removidos conforme objeto que foi selecionado na configuração.

O primeiro passo nesta tela é definir o nome do Physical Device através do campo Server Name. Recomenda-se a utilização de um nome específico de acordo com o projeto do sistema onde a UTR será utilizada, pois este será o nome que irá identificar este dispositivo na configuração da comunicação com dispositivos MMS Cliente e na configuração da troca de mensagens GOOSE com outros IEDs.

O próximo passo é adicionar um Logical Device. Para isto, deve-se selecionar o objeto LogicalDevice na parte esquerda da tela e depois clicar no botão “>” conforme mostrado na sequência.

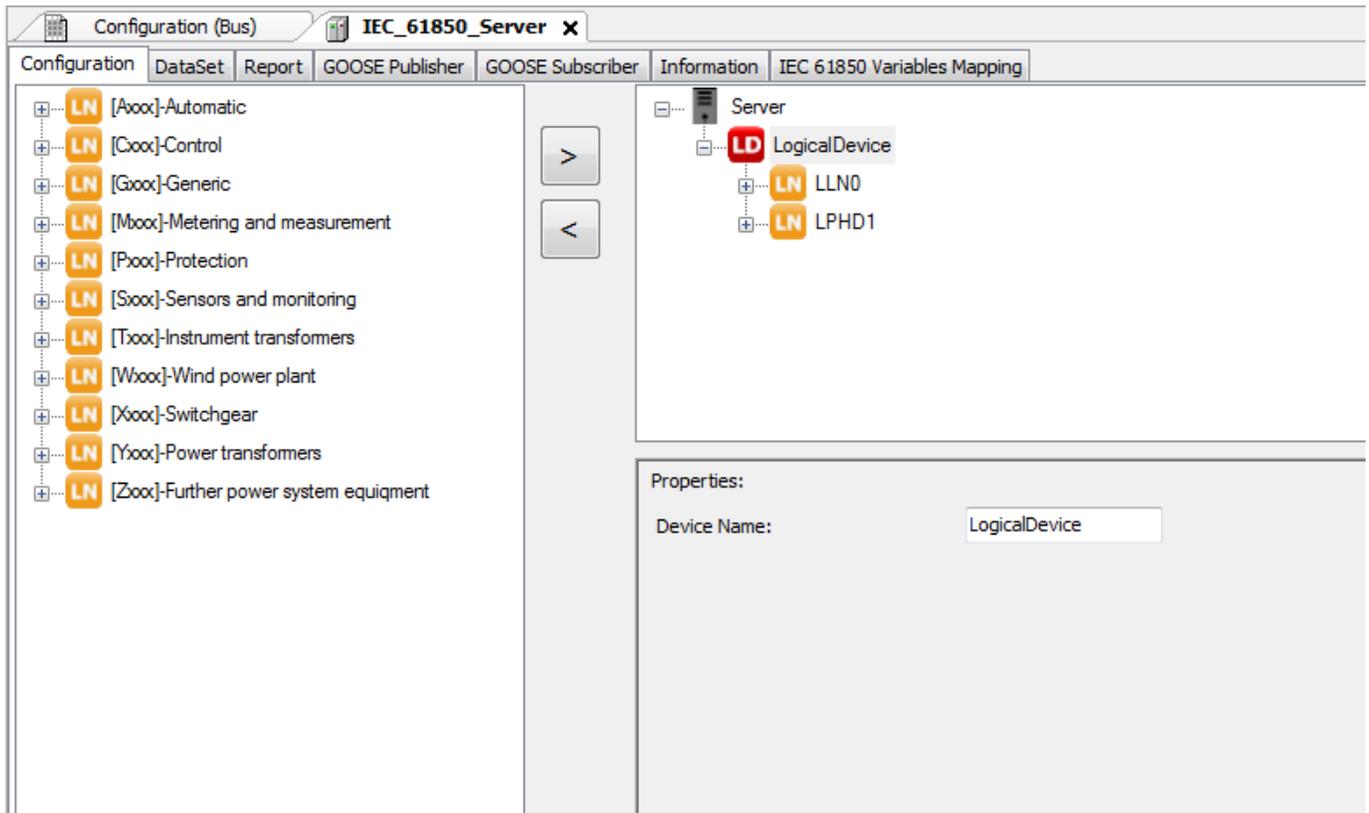


Figura 3-49. Adicionando e configurando o nome do Logical Device

O Logical Device é um nível de abstração do modelo de dados definido na IEC 61850, e tem como objetivo a organização dos dados. O driver IEC 61850 Servidor suporta apenas um Logical Device, e o seu nome pode ser alterado através do campo Device Name. O Logical Device vem por padrão com dois Logical Nodes que armazenam configurações e informações do dispositivo (LLN0 e LPHD1), os quais não podem ser removidos.

A partir deste ponto, o usuário deve incluir os Logical Nodes conforme a necessidade do seu projeto. Para instanciar novos Logical Nodes, basta selecioná-lo na lista que aparece do lado esquerdo da tela conforme indicado a seguir.

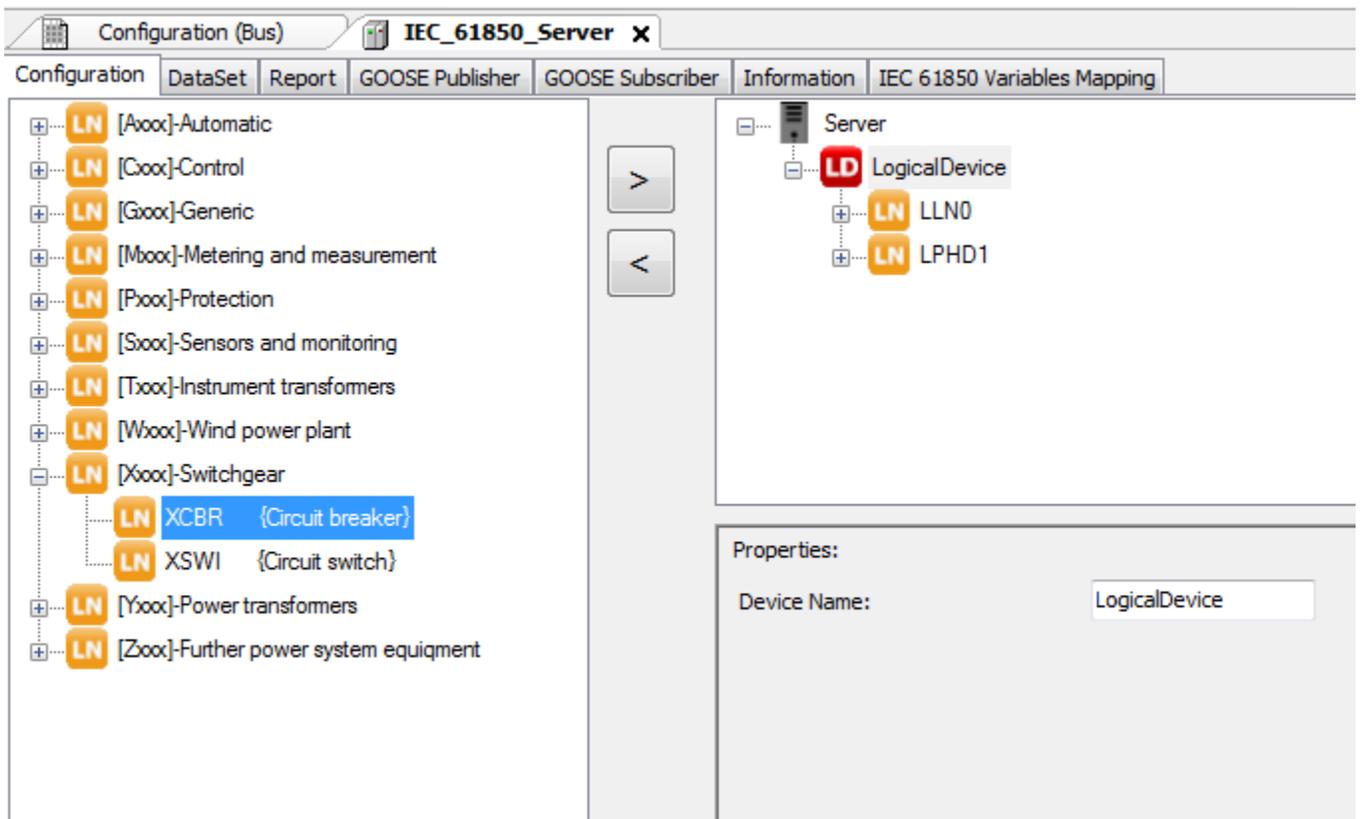


Figura 3-50. Adicionando um Logical Node

Ao clicar sobre um Logical Node configurado, é possível modificar o seu prefixo e o seu índice através dos campos Node Prefix e Logical Node Index. Além disto, é possível também adicionar e remover Data Objects opcionais. Da mesma forma, ao clicar sobre um Data Object, é possível adicionar e remover Data Attributes opcionais.

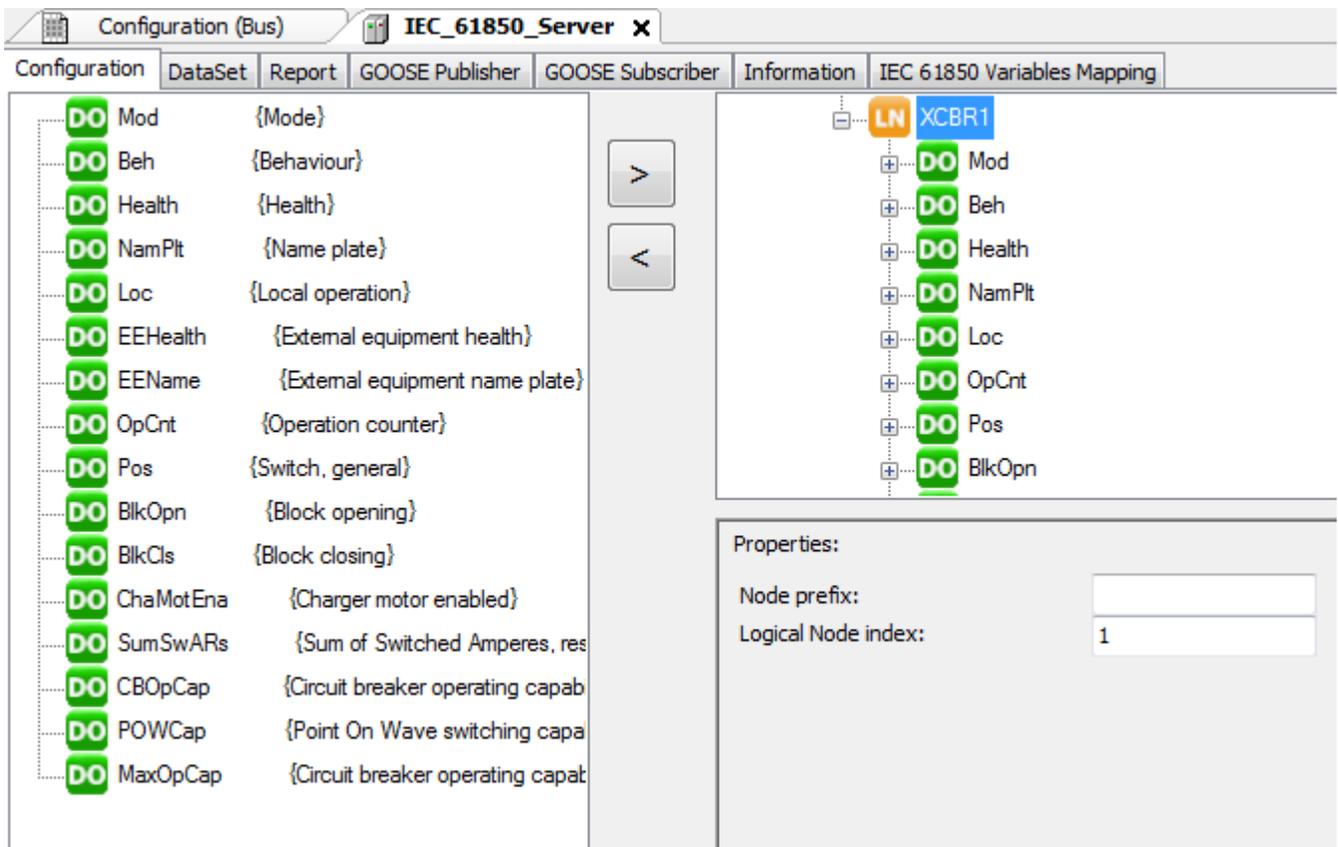


Figura 3-51. Configurando um Logical Node

Parâmetros configuráveis nessa tela:

- **Server Name** - Define o nome do Physical Device;
- **Device Name** - Define o nome do Logical Device;
- **Node Prefix** - Define o prefixo do Logical Node;
- **Logical Node Index** - Define o índice do Logical Node.

### Configuração dos Datasets

Os Datasets são listas de variáveis que são utilizadas na comunicação através de mensagens GOOSE e MMS (Reports etc.). Em outras palavras, consistem em um subconjunto de dados que contém referências para Atributos e/ou Objetos da base de dados configurada no driver. A configuração dos Datasets é realizada através da aba DataSet conforme mostrado na figura a seguir.

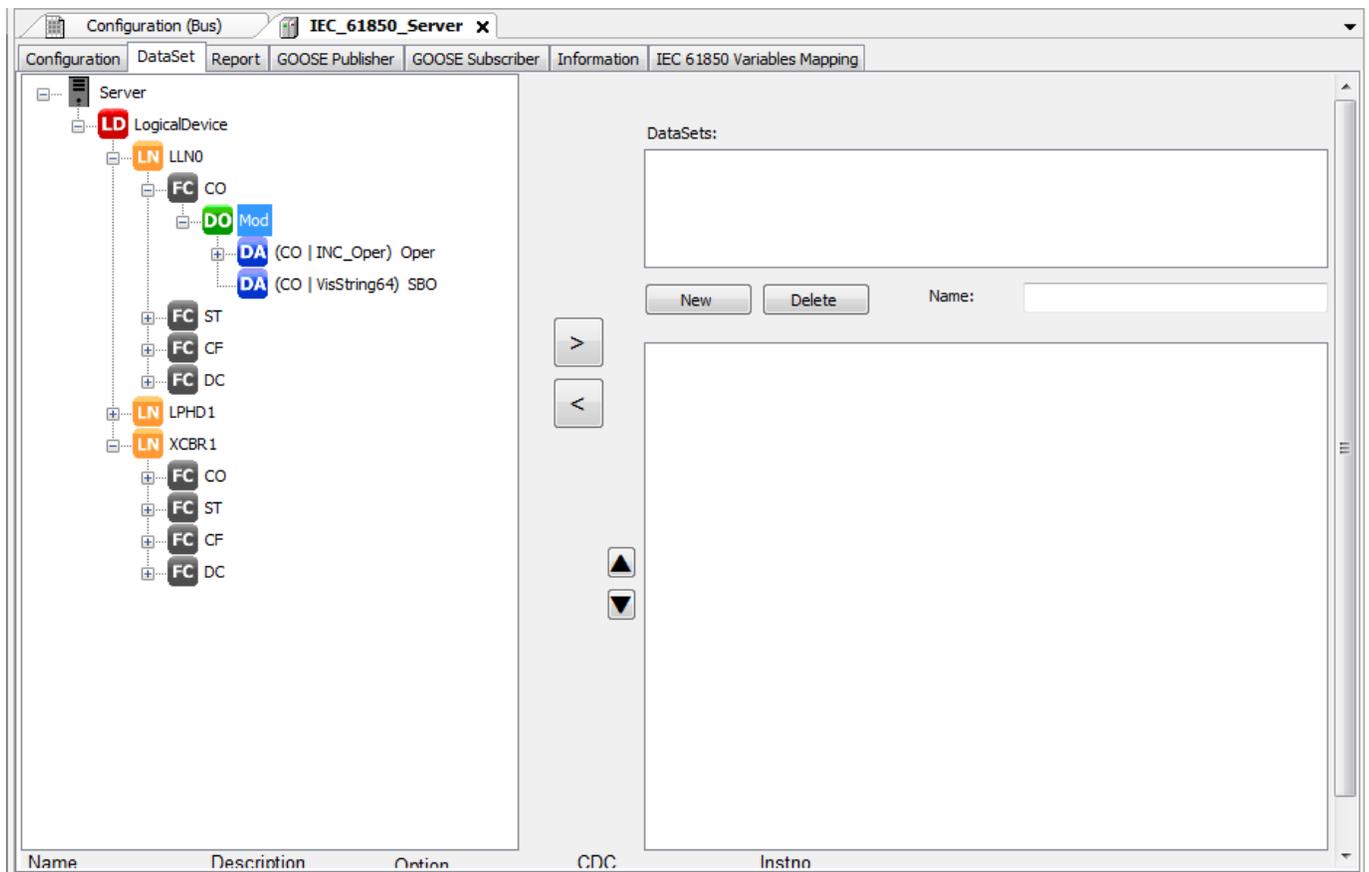


Figura 3-52. Configurando os Datasets

Esta tela é dividida em duas partes. No lado direito é apresentada a lista de Datasets atualmente configurados no dispositivo, onde existem também os comandos para criar, remover e renomear Datasets. No lado esquerdo, encontra-se a lista de elementos disponíveis que podem ser adicionados/removidos em cada Dataset a partir da base de dados que foi configurada.

Adicionalmente, a UTR Hadron Xtorm possui suporte ao recurso de Datasets Dinâmicos, o qual permite a criação de Datasets em tempo de execução pelo IEC 61850 Cliente (SCADA etc.). Após serem criados, é possível associá-los a um Report Control Block já existente. A configuração dos Datasets Dinâmicos é volátil, isto é, será perdida em caso de desligamento ou reconfiguração da UTR.

### Configuração dos Reports

O Report é um mecanismo de envio de um Dataset de forma não solicitada para um IEC 61850 Cliente. As suas configurações (Dataset associado, condições de disparo etc.) são todas armazenadas em uma entidade chamada Report Control Block (RCB). A configuração dos Reports é realizada através da aba Report conforme ilustrado a seguir.

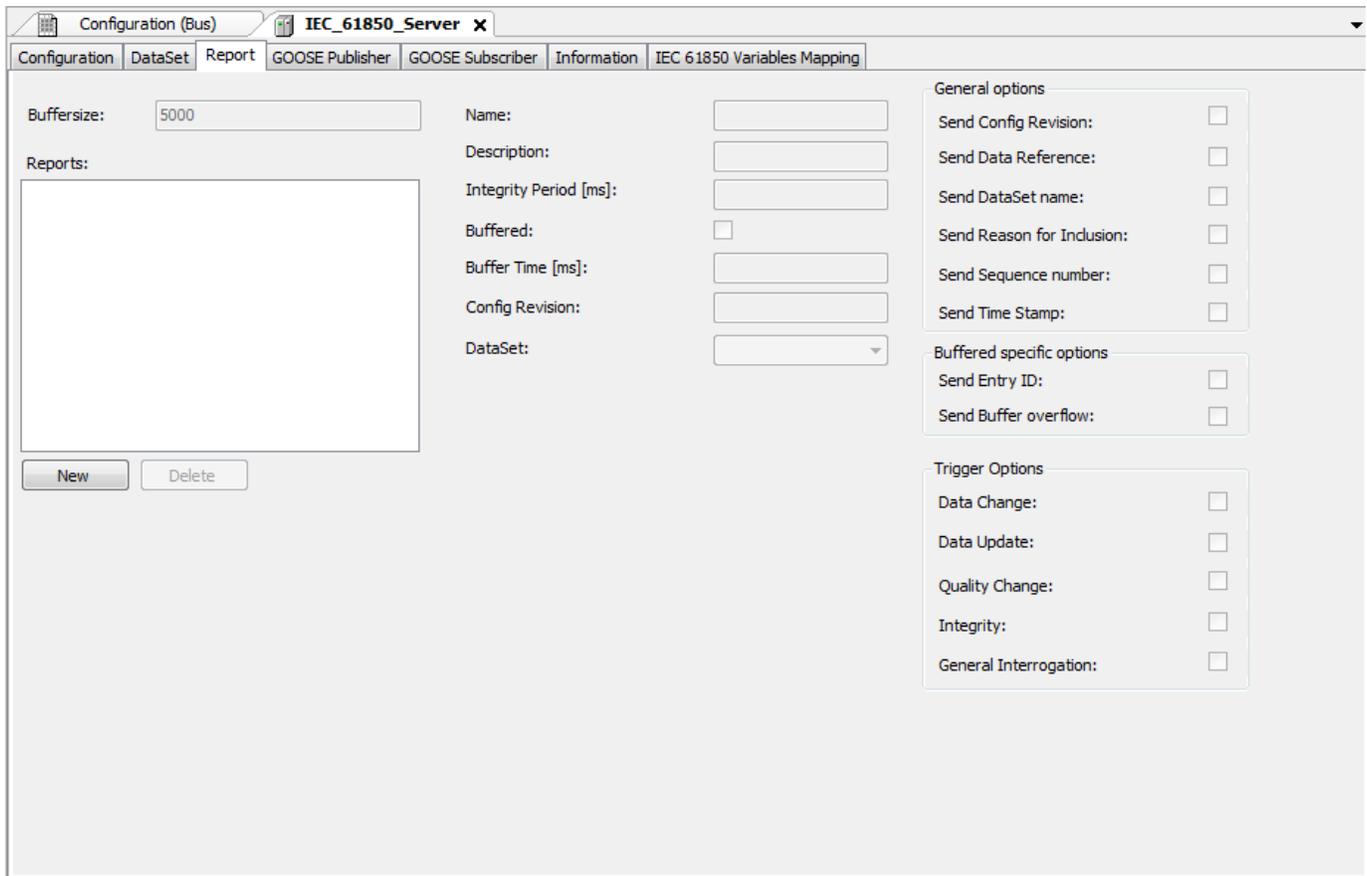


Figura 3-53. Configurando os Reports

No lado esquerdo da tela é apresentada a lista de RCBs, onde existem também os comandos para criar e remover Reports. No lado direito encontram-se as configurações do RCB.

O princípio de funcionamento do Report está baseado no envio das informações para o Cliente MMS sempre que houver qualquer alteração nos elementos contidos no Dataset daquele Control Block. Adicionalmente, o Report pode ser configurado para ser enviado periodicamente a partir do parâmetro Integrity Period. O Report pode também ser configurado no modo buferizado, onde as mensagens são armazenadas em um buffer interno da UTR para envio posterior caso o Cliente MMS não esteja conectado. Este buffer possui tamanho fixo de 20 kbytes para cada Report Control Block, sendo que a política de estouro é “manter os mais antigos” (isto é, caso o buffer esteja cheio e seja gerado um novo report, os reports armazenados no buffer serão mantidos, e o novo report será descartado).

Parâmetros configuráveis:

- **Buffersize** - Define o tamanho (em bytes) do buffer para armazenamento dos eventos. Contudo, este parâmetro não é suportado pelo driver, que utiliza um tamanho fixo de 20Kbytes para cada Buffered Report Control Block independentemente do valor configurado na tela;
- **Name** - Nome que identificará este Report Control Block para fins de configuração e de comunicação com outros dispositivos;
- **Description** - Descrição do Report Control Block, para auxiliar na documentação do projeto;
- **Integrity Period (ms)** - Período no qual o Report será disparado automaticamente caso nenhuma das outras condições de disparo tenha ocorrido;

- **Buffered** - Seleciona o modo Bufferizado, no qual os Reports são armazenados internamente na UTR para envio posterior caso o Cliente MMS não esteja conectado;
- **Buffer Time (ms)** - Tempo que será aguardado após a ocorrência de um evento para que o Report seja transmitido;
- **Config. Revision** - Número que será utilizado para identificar a revisão da configuração do Report Control Block;
- **Dataset** - Dataset que será monitorado e transmitido por este Report Control Block;
- **Send Config Revision** - Caso selecionado, envia o parâmetro Revisão da Configuração juntamente com o Report;
- **Send Data Reference** - Caso selecionado, envia a referência de dados juntamente com o Report;
- **Send Dataset Name** - Caso selecionado, envia o nome do Dataset juntamente com o Report;
- **Send Reason for Inclusion** - Caso selecionado, envia o motivo da inclusão juntamente com o Report;
- **Send Sequence Number** - Caso selecionado, envia o número de sequência juntamente com o Report;
- **Send Time Stamp** - Caso selecionado, envia a estampa de tempo juntamente com o Report;
- **Send Entry ID** - Caso selecionado, envia o ID da Entrada juntamente com o Report;
- **Send Buffer Overflow** - Caso selecionado, envia a informação de Buffer Overflow juntamente com o Report;
- **Data Change** - Caso selecionado, habilita a condição de disparo do Report em função da alteração de valor;
- **Data Update** - Habilita a condição de disparo do Report em função da atualização do valor da variável (independente se o valor foi alterado). Contudo, esta opção não é suportada pelo driver e, caso habilitada, não terá efeito algum;
- **Quality Change** - Caso selecionado, habilita a condição de disparo do Report em função da alteração de qualidade;
- **Integrity** - Caso selecionado, habilita a condição de disparo do Report caso solicitado o Polling de Integridade pelo Cliente MMS;
- **General Interrogation** - Habilita a condição de disparo do Report em função do recebimento de um comando de Interrogação Geral enviado pelo Cliente MMS. Contudo, esta opção não é suportada pelo driver e, caso habilitada, não terá efeito algum.

#### Configuração GOOSE

O GOOSE (Generic Object Oriented Substation Events) é uma forma de comunicação definida pela IEC 61850 criada para permitir a rápida troca de informações entre IEDs. A comunicação GOOSE utiliza o modelo Publisher/Subscriber (Produtor/Consumidor), isto é, as mensagens transmitidas por um IED são recebidas por todos os outros dispositivos presentes na rede, cabendo a eles então avaliar se desejam armazenar e utilizar os dados contidos na mensagem. As mensagens GOOSE são encapsuladas diretamente sobre o frame Ethernet utilizando endereços MAC Multicast ou Broadcast, não havendo relação com as camadas TCP/IP. A configuração das mensagens GOOSE que serão transmitidas pela UTR Xtorm é realizada na aba GOOSE Publisher conforme mostrado na sequência.

The screenshot shows the configuration window for a GOOSE Publisher in the IEC 61850 Server. The window title is 'IEC\_61850\_Server'. The tabs at the top are 'Configuration', 'DataSet', 'Report', 'GOOSE Publisher', 'GOOSE Subscriber', 'Information', and 'IEC 61850 Variables Mapping'. The 'GOOSE Publisher' tab is active. The interface is divided into three main sections: 'General', 'Publisher', and 'DataSet'. The 'General' section contains fields for Name, Description, GOOSE ID, DataSet, MAC, and APPID. The 'Publisher' section contains checkboxes for Needs Commissioning and Fast Goose, and input fields for DataSet Config Revision, Max. Time [ms], Min. Time [ms], VLAN, VLAN-ID, and VLAN-Priority. The 'DataSet' section is currently empty. On the left side, there are 'New' and 'Delete' buttons.

Figura 3-54. Configurando a comunicação GOOSE Publisher

As configurações de uma mensagem GOOSE a ser transmitida pela UTR Xtorm (Dataset associado, tempos de envio etc.) são todas armazenadas em uma entidade chamada GOOSE Control Block (GCB). No lado esquerdo da tela é apresentada a lista de GCBs, onde existem também os comandos para criar e remover GCBs. No lado direito encontram-se as configurações do GCB. Na parte inferior da tela, é possível visualizar os elementos que constituem o Dataset que foi associado a esta mensagem.

O princípio de funcionamento do GOOSE Publisher está baseado no envio da mensagem quando houver qualquer alteração nos elementos contidos no Dataset daquele Control Block. Após o envio da mensagem, são realizadas retransmissões utilizando a base de tempo definida pelo parâmetro Min. Time. O intervalo de tempo para a primeira retransmissão é de 1X Min. Time, e depois vai sendo dobrado (2X, 4X, 8X etc.) até alcançar o intervalo definido pelo parâmetro Max. Time. Caso o Dataset não apresente alterações dentro deste tempo, a mensagem é retransmitida continuamente com o intervalo de Max. Time (conhecido também como heartbeat).

As atividades de comparação do Dataset e envio da mensagem executadas por cada Control Block estão associadas a uma tarefa da UTR. Caso a opção Fast Goose esteja habilitada, o Control Block executa estas atividades na ProtTask, a qual é uma tarefa cíclica de alta prioridade com intervalo bastante curto (o default é 4ms), permitindo então configurar mensagens GOOSE extremamente rápidas para atender requisitos de desempenho exigidos por partes críticas do sistema. Caso esta opção não esteja habilitada, o Control Block executa estas atividades na tarefa principal da UTR, a MainTask.

Caso um Dataset esteja associado a um GOOSE Control Block cuja opção Fast Goose esteja habilitada, não é possível utilizar este mesmo Dataset em um Report Control Block. Caso este tipo de configuração seja realizado, o Report não funcionará corretamente.

Parâmetros configuráveis:

- **Name** - Nome que identificará este GOOSE Control Block para fins de configuração e de comunicação com outros dispositivos;
- **Description** - Descrição do GOOSE Control Block, para auxiliar na documentação do projeto;
- **GOOSE ID** - Também é conhecido como AppID, é uma string de identificação definida pelo usuário;
- **Dataset** - Dataset que será monitorado e transmitido por este GOOSE Control Block;
- **MAC** - Endereço MAC de destino (Multicast);
- **APP ID** - Número de identificação da aplicação. Este número pode ser utilizado para distinguir mensagens GOOSE enviadas por diferentes partes do sistema de automação;
- **Needs Commissioning** - Deve ser habilitada pelo usuário para indicar que a configuração deste GOOSE Control Block ainda não foi concluída;
- **Dataset Config Revision** - Número que deve ser utilizado para indicar a revisão do Dataset associado a este GOOSE Control Block;
- **Max. Time (ms)** - Intervalo de tempo máximo para a retransmissão de uma mensagem GOOSE;
- **Min. Time (ms)** - Intervalo de tempo mínimo para a retransmissão de uma mensagem GOOSE;
- **Fast GOOSE** - Define em qual tarefa da UTR será executado este Control Block. Caso selecionado, a comparação dos valores e envio da mensagem GOOSE serão executados na ProtTask. Caso contrário, esta atividade será realizada na MainTask;
- **VLAN** - Caso selecionado, habilita a utilização de VLAN na mensagem GOOSE;
- **VLAN ID** - Número de identificação da VLAN;
- **VLAN Priority** - Número prioridade da VLAN.

A configuração das mensagens GOOSE que serão recebidas pela UTR Xtorm é realizada na aba GOOSE Subscriber através da importação de arquivos SCL (Substation Configuration Language). Estes arquivos são gerados pela ferramenta de configuração do IED responsável por enviar a mensagem, e armazenam internamente as informações dos GOOSE Control Blocks relacionados. A importação é realizada através do botão localizado no canto superior esquerdo da tela conforme indicado a seguir.

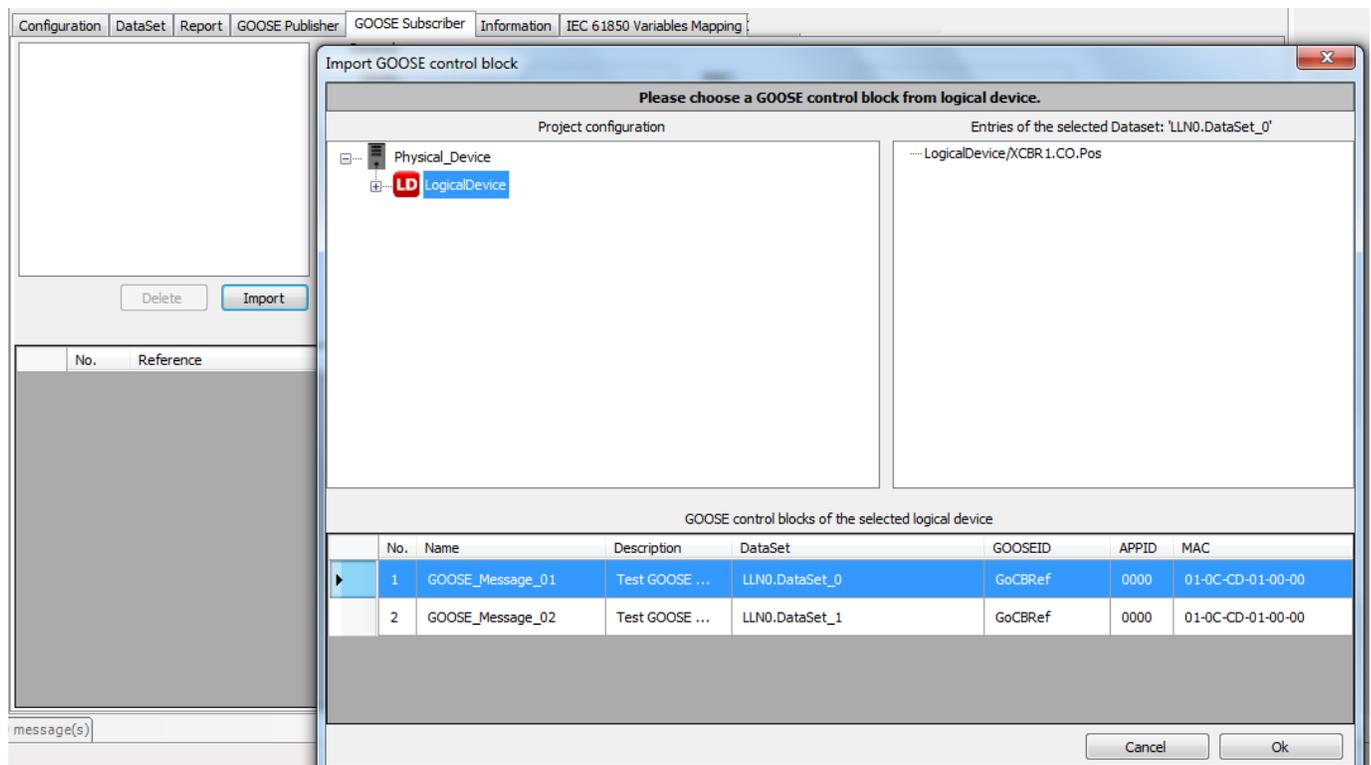


Figura 3-55. Importando um arquivo SCL

Após clicar no botão de importação e selecionar o arquivo SCL desejado, é aberta uma tela onde é apresentada uma lista de Physical Devices e Logical Devices contidos neste arquivo. Ao clicar sobre um Logical Device, é apresentada na parte inferior da tela a lista de mensagens GOOSE (Control Blocks) disponíveis para serem recebidos. Após selecionar a mensagem que se deseja receber, basta clicar em OK. Retornando à aba GOOSE Subscriber, no canto superior esquerdo é apresentada a lista de mensagens configuradas para serem recebidas.

No.	Reference	Type	Varname	Use default name
1	LogicalDevice/XCBR1.CO.Pos.Oper.ctlVal	BOOLEAN	GOOSE_Message_01_E...	<input checked="" type="checkbox"/>
2	LogicalDevice/XCBR1.CO.Pos.Oper.operTm	Timestamp		<input type="checkbox"/>
3	LogicalDevice/XCBR1.CO.Pos.Oper.origin.orCat	orCat		<input type="checkbox"/>
4	LogicalDevice/XCBR1.CO.Pos.Oper.origin.orI...	Octet64		<input type="checkbox"/>
5	LogicalDevice/XCBR1.CO.Pos.Oper.ctlNum	INT8U		<input type="checkbox"/>
6	LogicalDevice/XCBR1.CO.Pos.Oper.T	Timestamp		<input type="checkbox"/>
7	LogicalDevice/XCBR1.CO.Pos.Oper.Test	BOOLEAN		<input type="checkbox"/>
8	LogicalDevice/XCBR1.CO.Pos.Oper.Check	Check		<input type="checkbox"/>
9	LogicalDevice/XCBR1.CO.Pos.SBO	VisString64		<input type="checkbox"/>

Figura 3-56. Configurando a comunicação GOOSE Subscriber

Ao selecionar uma mensagem, no lado direito são apresentadas as informações relacionadas à mensagem que será recebida. Estes campos são preenchidos automaticamente através da importação dos dados contidos no GOOSE Control Block associado à mensagem, desta forma não devem ser modificados pelo usuário.

Na parte inferior desta tela são apresentados os dados contidos na mensagem selecionada, onde é possível configurar quais elementos do Dataset recebido serão efetivamente armazenados na memória da UTR. Este mapeamento é realizado através do campo Varname, onde deve ser informado o nome da variável para a qual será copiado o dado no momento da recepção da mensagem. Este campo pode ser preenchido manualmente pelo usuário com o nome de uma variável já existente na sua aplicação, ou também pode ser preenchido automaticamente através da opção Use default name. Ao marcar esta opção, a ferramenta MasterTool Xtorm cria automaticamente uma variável na GVL IEC61850\_Generated\_GVL para armazenar o elemento correspondente recebido pela mensagem GOOSE. O nome desta variável é formado pelo nome do GOOSE Control Block adicionando o sufixo “\_EntryX”, onde X é um número que indica a posição do elemento na mensagem.

### Configuração do Mapeamento de Variáveis IEC 61850

A aba IEC 61850 Variables Mapping permite realizar a associação entre variáveis internas da UTR e Atributos dos Logical Nodes. Este recurso é essencial para permitir a implementação de Logical Nodes a partir de dados oriundos de outros protocolos de comunicação, uma vez que os tipos de dados entre os protocolos podem não ser totalmente compatíveis.

IEC 61850 Variable	Origin Variable
gfbIEC61850_LogicalDevice.LLN0.fbBeh.stVal	Input00

Figura 3-57. Mapeamento de Variáveis IEC 61850

A configuração do mapeamento de Variáveis IEC 61850 consiste basicamente em uma tabela com duas colunas, uma contendo a variável que representa o Atributo do Logical Node (IEC 61850 Variable) e a outra contendo a variável da UTR com a qual o Atributo será associado (Origin Variable). Cada linha consiste em um mapeamento.

No exemplo da figura acima, o Atributo stVal do objeto Beh do Logical Node LLN0 foi mapeado para uma variável chamada Input00. Visto que esta associação pode implicar em uma conversão de tipo de dado, uma Variável Origem pode somente ser associada a um Atributo cujo tipo tenha a mesma sinalização (signed/unsigned) e tamanho (em bits) igual ou maior. O único tipo que não segue esta regra é o DBP, pois consiste em uma estrutura de duas variáveis BOOL mas, mesmo assim pode ser mapeado para Atributos que representam o valor de pontos duplos e possuem 8 bits. Existem várias possibilidades de mapeamento entre uma Variável Origem e uma Variável IEC 61850.

Caso a Variável Origem esteja associada a uma fonte capaz de gerar eventos com estampa de tempo própria (IED DNP3 ou módulo de entrada digital por exemplo), o driver IEC 61850 Servidor receberá estes eventos e atualizará o atributo "t" (Timestamp) do Objeto referente ao atributo que foi mapeado, se comportando como um "consumidor" conforme descrito na seção Fila de Eventos da UCP.

### Diagnósticos IEC 61850 Servidor

Os diagnósticos do protocolo IEC 61850 Servidor configurado são armazenados em variáveis do tipo T\_DIAG\_IEC61850\_SERVER. As informações relacionadas à transmissão/recepção de mensagens GOOSE são armazenadas em arrays com dimensão correspondente à quantidade máxima que pode ser configurada. Cada posição do array possui uma estrutura que possui informações correspondes a um GOOSE Control Block ou a um Subscription.

### Interceptação de Comandos

A interceptação de comandos recebidos pelo driver IEC 61850 Servidor é realizada através do bloco de função CommandReceive conforme descrito na seção Interceptação de Comandos Oriundos do Centro de Controle. Este recurso permite a interceptação de comandos de seleção e operação para os seguintes CDCs (Common Data Classes): SPC, DPC, INC, BSC, ISC.

A seguir está mostrado um exemplo de aplicação utilizando a linguagem ST detalhando a interceptação de um comando para o objeto BlkOpn de um Logical Node XCBR, respondendo "SUCCESS" ao IEC 61850 Cliente.

```
PROGRAM UserPrg
VAR
CRReceive: CommandReceiver;
CCommand: COMMAND_T;
CRResult: COMMAND_RESULT:= COMMAND_RESULT.SUCCESS;
END_VAR
```

```
CRReceive.dwVariableAddr:=
ADR(gfbIEC61850_LogicalDevice.XCBR1.fbBlkOpn.Oper.ctlVal);
CRReceive.bExec:= TRUE;
CRReceive.dwTimeout:= 500;

// Caso um comando seja capturado:
IF CRReceive.bCommandAvailable THEN
// Salva os dados do comando
CCommand:= CRReceive.sCommand;
// Envia resposta "SUCCESS"
CRReceive.eCommandResult:= CRResult;
CRReceive.bDone:= TRUE;
END_IF

CRReceive();

IF CRReceive.bDone THEN
CRReceive.bDone:= FALSE;
END_IF
```

## Desempenho da Comunicação

### Comunicação MODBUS

Os dispositivos MODBUS configuráveis na UCP Hadron Xtorm executam em segundo plano, com uma prioridade abaixo da aplicação de usuário e de forma cíclica. Sendo assim, seu desempenho irá variar de acordo com o tempo restante, levando em consideração a diferença entre o intervalo e tempo que a aplicação leva para ser executada. Também deve-se levar em conta o número de ciclos que o dispositivo, escravo ou servidor, leva para responder a uma requisição.

Para processar e transmitir uma resposta, um MODBUS RTU Escravo irá levar dois ciclos (tempo do ciclo da tarefa MODBUS), enquanto um MODBUS Ethernet Servidor levará apenas um ciclo. Porém, esse é o tempo mínimo entre a recepção de uma requisição e o envio da resposta. Caso a requisição seja enviada logo após a execução de um ciclo da tarefa MODBUS, o tempo poderá ser equivalente a 2 ou 3 vezes o tempo de ciclo para o MODBUS Escravo, e de 1 a 2 vezes o tempo de ciclo para o MODBUS Servidor.

Neste caso:

Tempo Máximo de Resposta = 3\*(tempo de ciclo) + (tempo de execução das tarefas) + tempo interframe chars) + (Send Delay)

Para os dispositivos MODBUS RTU Mestre e MODBUS Ethernet Cliente o princípio de funcionamento é exatamente igual, mas leva em consideração o tempo de varredura (ms) configurado na relação MODBUS e não o tempo do ciclo da tarefa MODBUS. Para esses casos, o tempo de ciclo da tarefa MODBUS é especificado conforme indicado a seguir.

### MODBUS RTU Mestre

Tempo de ciclo da tarefa = (Menor Tempo de Varredura (ms) configurado nas relações / 5)

Caso o tempo de ciclo da tarefa, calculado de acordo com a expressão acima, fique maior que o valor configurado no campo "Time-out de Comunicação" disponível na aba de Configurações Avançadas do Dispositivo, o Tempo de ciclo da tarefa será:

Tempo de ciclo da tarefa = Time-out de Comunicação (ms) configurado no dispositivo

#### MODBUS Ethernet Cliente

Tempo de ciclo da tarefa = (Menor Tempo de Varredura (ms) configurado nas relações / 3)

Caso o tempo de ciclo da tarefa, calculado de acordo com a expressão acima, fique maior que o valor configurado no campo "Time-out de Comunicação" disponível na aba de Configurações Avançadas do Dispositivo, o Tempo de ciclo da tarefa será:

Tempo de ciclo da tarefa = Time-out de Comunicação (ms) configurado no dispositivo

Para estes casos, o pior desempenho de uma relação será ser executada após o seu tempo de varredura, somado ao tempo de execução da aplicação de usuário.

#### Servidor MODBUS

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

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

$$N = C \times (Z - (Z / (T \times 1000)))$$
$$Z = 1 / T$$

Onde "N" é o número médio de respostas por segundo, "C" é o número de conexões e T é igual ao tempo de ciclo da tarefa MODBUS (em segundos).

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

#### DNP3 Servidor

O driver DNP3 Servidor é executado pela UCP da mesma forma como os outros drivers de comunicação Servidores, isto é, em segundo plano, com uma prioridade abaixo da aplicação de usuário e de forma cíclica. A tarefa deste driver especificamente executa a cada 50ms, e basta 1 ciclo de execução do driver para processar e responder requisições. Contudo, por se tratar de uma tarefa de baixa prioridade, não é garantido que consiga executar com esta periodicidade pois depende do percentual de UCP livre (diferença entre o intervalo da MainTask e tempo que a aplicação do usuário leva para ser executada) e da concorrência com tarefas de outros protocolos configurados na UCP.

#### DNP3 Cliente

O driver DNP3 Cliente também é executado pela UCP em segundo plano, com uma prioridade abaixo da aplicação de usuário, de forma cíclica a cada 50 ms, sendo necessários 3 ciclos de execução do driver para realizar as requisições pendentes. Diferente do driver MODBUS Cliente, o período desta tarefa é fixo em 50 ms, independente do período de Polling configurado nas requisições. Desta forma, o período de Polling efetivamente executado pelo driver pode apresentar uma imprecisão de até 150 ms.

#### IEC 60870-5-104 Servidor

O driver IEC 60870-5-104 Servidor é executado pela UCP da mesma forma como os outros drivers de comunicação Servidores, isto é, em segundo plano, com uma prioridade abaixo da aplicação de usuário e de forma cíclica. A tarefa deste driver especificamente executa a cada 50ms, e basta 1 ciclo de execução do driver para processar e responder requisições. Contudo,

por se tratar de uma tarefa de baixa prioridade, não é garantido que consiga executar com esta periodicidade pois depende do percentual de UCP livre (diferença entre o intervalo da MainTask e tempo que a aplicação do usuário leva para ser executada) e da concorrência com tarefas de outros protocolos configurados na UCP.

#### **IEC 60870-5-104 Cliente**

O driver IEC 60870-5-104 Cliente também é executado pela UCP em segundo plano, com uma prioridade abaixo da aplicação de usuário, de forma cíclica a cada 50ms.

#### **Desempenho do Sistema**

Em casos em que a aplicação possui apenas uma tarefa de usuário MainTask, responsável pela execução de uma única unidade de programação do tipo Programa denominada MainPrg, a UCP consome um determinado tempo para que a tarefa seja processada. A esse tempo damos o nome de Tempo de Execução.

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

Device -> Application -> Task Configuration na aba Monitor, na coluna Tempo de Ciclo Médio.

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

O MasterTool Xtorm tem ainda a opção de limitar o Tempo de Execução da tarefa de usuário MainTask, para que o Tempo de Execução mencionado acima seja respeitado. Isto é feito consistindo o Cão-de-Guarda da tarefa de usuário em 80% do intervalo.

Caso o usuário esteja ciente das informações acima e queira utilizar esta consistência do Cão-de-Guarda, isso será possível marcando o checkbox Gerar erro na consistência do cão-de-guarda das tarefas na aba de configuração da UCP. Porém, o recomendado é manter o tempo de cão-de-guarda configurado com o padrão de 1000 ms e o checkbox para gerar erro na consistência desmarcado.

#### **☪ NOTA:**

Caso o percentual de 80% não seja respeitado e o tempo de execução da tarefa de usuário se aproxime ou exceda o intervalo configurado da MainTask, o visor e o botão de diagnóstico podem não responder, uma vez que sua prioridade na execução do sistema é menor do que as tarefas de usuário. Caso uma aplicação com erros seja carregada na UCP, pode ser necessário reinicializá-la sem carregar esta aplicação.

#### **Escaneamento de E/S**

A atualização das entradas e saídas associadas aos módulos presentes no barramento ocorre primariamente na tarefa principal da UTR, a MainTask, conforme a sequência descrita a seguir:

1. Leitura das Entradas;
2. Escrita das Saídas;
3. Execução da aplicação do usuário;
4. Espera até completar o intervalo configurado para a MainTask.

O Tempo de Execução será resultante da soma de 3 parcelas: leitura das entradas, escrita das saídas e execução da aplicação do usuário. Desta forma, o tempo de execução da aplicação é

afetado pela quantidade de módulos de entrada e saída presentes no barramento. A duração da parcela relacionada à atualização de E/S é de aproximadamente 1 ms para cada 20 módulos inseridos no projeto. A última etapa da MainTask (aguarda completar o intervalo configurado) é onde os drivers de comunicação e todos os outros serviços de baixa prioridade executam.

Adicionalmente, é possível configurar a UTR para realizar a atualização de entradas e saídas dos módulos em outras tarefas através da opção “Habilita atualização de E/S por tarefa”, localizada nas configurações da UCP. Ao habilitar esta opção, a UCP realiza um acesso ao barramento local no contexto de cada tarefa para atualização das entradas e saídas que estiverem sendo utilizadas nela. Esta opção pode ser muito útil em casos em que se utiliza a tarefa ProtTask e seja necessária uma rápida atuação em entradas e saídas (lógicas de proteção por exemplo). A ordem das ações executadas é levemente diferente da MainTask, conforme descrito a seguir:

1. Execução da aplicação do usuário;
2. Escrita das Saídas;
3. Leitura das Entradas;
4. Aguarda completar o intervalo configurado para a tarefa.

#### ¶ NOTAS:

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

A lista de seleção “Sempre atualizar variáveis” define a atualização de variáveis de E/S:

- Usar configuração do dispositivo pai: atualizar de acordo com a configuração do dispositivo pai;
- Ativar 1 (usar a tarefa do ciclo de barramento se não for usada em outra tarefa): as variáveis de E/S são atualizadas na tarefa de ciclo de barramento se elas não forem usadas em qualquer outra tarefa;
- Ativar 2 (sempre na tarefa de ciclo de barramento): todas as variáveis são atualizadas a cada ciclo da tarefa do barramento, independentemente de serem usadas e se elas são mapeadas para um canal de entrada ou de saída.

### Cartão de Memória

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

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

# Configuração do Relógio RTC



A UCP da Série Hadron Xtorm possui um relógio interno que pode ser utilizado através da biblioteca LibPlcStandard.lib. Essa biblioteca é carregada automaticamente durante a criação de um novo projeto. A figura a seguir mostra como incluir os blocos e funções disponíveis na biblioteca para acessar o relógio.

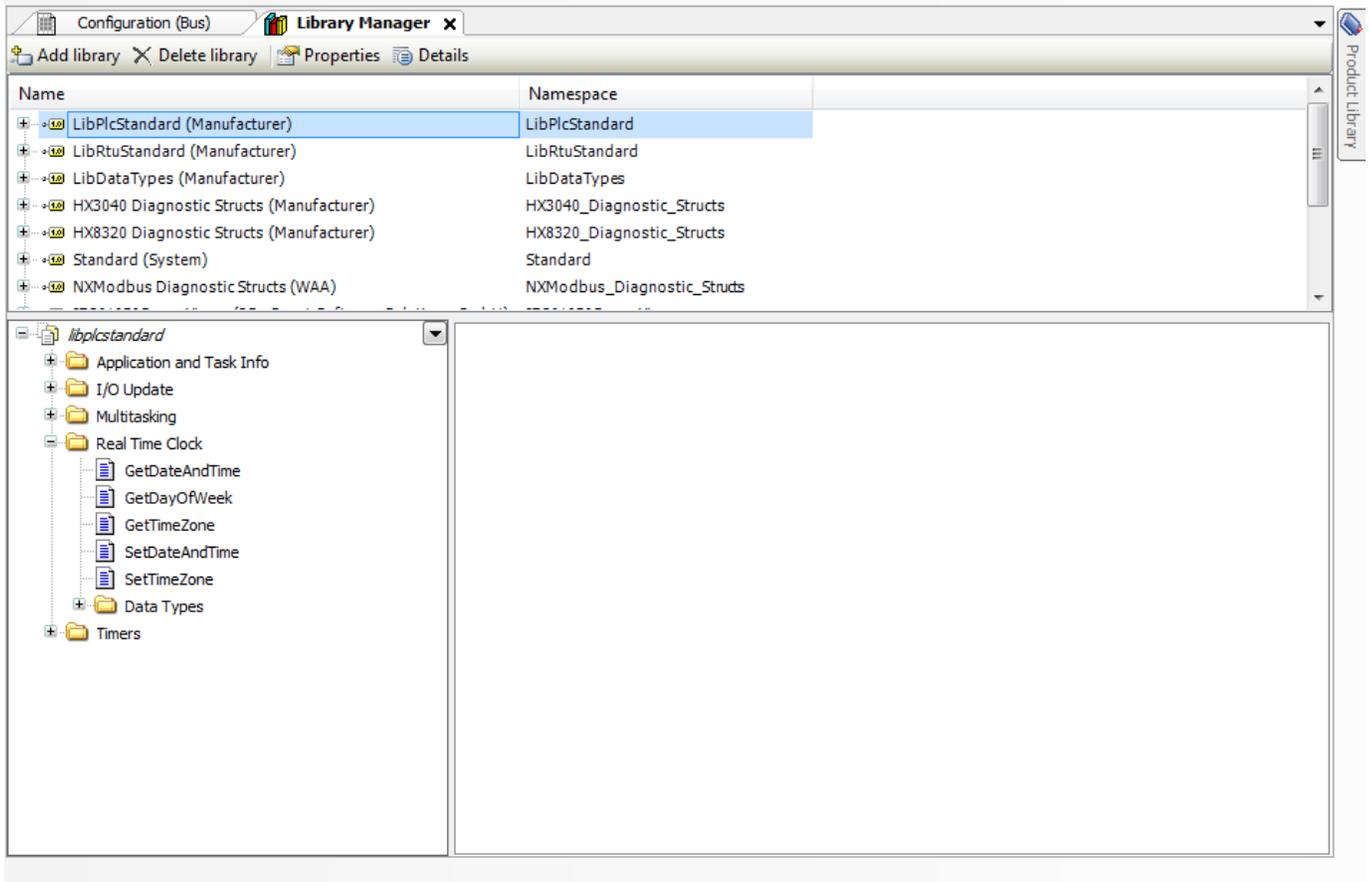


Figura 3-58. Blocos de Escrita e Leitura do Relógio

Entre outros blocos funcionais e funções, existem quatro muito importantes utilizados para a leitura do relógio (`GetDateAndTime` e `GetTimeZone`) e para configurar novos valores de data e hora (`SetDateAndTime` e `SetTimeZone`). Estes elementos de software utilizam sempre o horário local, isto é, levam em consideração o valor definido para o Time Zone.

## Funções de Leitura do RTC

### GetDateAndTime

A função `GetDateAndTime` é utilizada para fazer a leitura da data e hora com o retorno em milissegundos. Quando chamada, a função irá ler o valor atual do relógio e preencher a estrutura `DATEANDTIME`. O resultado da leitura é retornado pela função.

## GetTimeZone

A função GetTimeZone é utilizada para fazer a leitura das configurações de fuso horário. Quando chamada, a função irá ler a configuração atual de fuso horário e preencher a estrutura TIMEZONE. O resultado da leitura é retornado pela função.

## Blocos Funcionais e Funções de Escrita e Configuração do RTC

### SetDateAndTime

O Bloco Funcional SetDateAndTime é utilizado para realizar a configuração do relógio conforme ilustrado na figura a seguir.



Figura 3-59. Configura Data e Hora com Milissegundos

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

#### ⚠ NOTA:

Se o usuário tentar escrever valores de hora fora do intervalo do RTC, os valores serão convertidos para valores válidos, desde que não ultrapasse a faixa válida de 01/01/2000 até 31/12/2035. Por exemplo, se o usuário tentar escrever o valor 2000 ms, ele será convertido para 2 segundos, se escrever o valor 100 segundos, ele será convertido para 1 min e 40 segundos. Se escrever o valor de 30 horas, ele será convertido para 1 dia e 6 horas, e assim por diante. Existe uma exceção para a geração de erros, para valores "0" dos campos de entrada de valores, a função não retornará erro. Neste caso, a data será convertida para a data anterior mais próxima. Por exemplo: se o usuário escrever a data 00/04/2014, a função irá converter para 31/03/2014.

### SetTimeZone

A função SetTimeZone realiza a escrita das configurações de fuso horário. Quando chamada, a função irá configurar o valor de TIMEZONE como a nova configuração de fuso horário do sistema. O resultado da configuração é retornado pela função.

#### ⚠ NOTA:

Para realizar o acerto do relógio, devem-se utilizar valores de hora e datas dentro da seguinte faixa válida: -12h59min até 13h59min, caso contrário, será reportado um erro através do parâmetro de saída STATUS. Para maiores detalhes do parâmetro de saída STATUS, consultar a seção RTC\_CMD\_STATUS.

## Estruturas de Dados do RTC

Os blocos funcionais de leitura e configuração do RTC da UCP da Série Hadron Xtorm utilizam as seguintes estruturas de dados na sua configuração:

- **EXTENDED\_DATE\_AND\_TIME** - Esta estrutura é utilizada para armazenar a data do RTC quando utilizados os blocos funcionais para leitura/configuração do horário.

- **DAYS\_OF\_WEEK** - Esta estrutura é utilizada para armazenar o dia da semana quando utilizada a função para leitura do dia da semana.
- **RTC\_CMD\_STATUS** - Este enumerador é utilizado para retornar o tipo de erro na configuração ou leitura do RTC.
- **TIME\_ZONE SETTINGS** - Esta estrutura é utilizada para armazenar o valor do fuso horário nas requisições de leitura/configuração dos blocos funcionais do RTC.

🔔 **NOTA:**

Bibliotecas diferentes da LibPlcStandard, que contenham blocos funcionais ou funções que possam fazer acesso de leitura e escrita da data e hora no sistema, não são indicados. A biblioteca LibPlcStandard possui as interfaces adequadas para escrever e ler a data e hora do sistema e informar os diagnósticos corretos.



#### **Pesquisa na Documentação: leitura e configuração do relógio**

Consulte, na documentação do produto, os parâmetros de entrada e saída dos Blocos Funcionais e Funções para leitura e escrita do RTC, bem como os itens que compõem as Estruturas de Dados do RTC.

Fonte: Manual de Utilização Hadron Xtorm – MU223000!



#### **Estudo Dirigido 3-2: leitura e configuração do relógio**

Utilizando a linguagem ST implemente os seguintes códigos:

- A. Leitura da data e hora com o retorno em milissegundos;
- B. Leitura do dia da semana;
- C. Leitura das configurações de fuso horário;
- D. Configuração da data e hora;
- E. Configuração do fuso horário.

🔔 **DICAS:**

1. Consulte o Instrutor para orientações adicionais;
2. A resolução do estudo dirigido encontra-se no final deste Tutorial.

## Memória de Arquivos de Usuário



A UCP da Série Hadron Xtorm possui uma área de memória destinada ao armazenamento de dados de uso geral, ou seja, o usuário poderá gravar diversos arquivos de qualquer formato na memória da UCP. Esta área de memória pode variar de acordo com o modelo de UCP utilizado. Consultar capítulo Características Gerais.

Para usar esta área, o usuário deverá acessar um projeto no software MasterTool Xtorm 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.

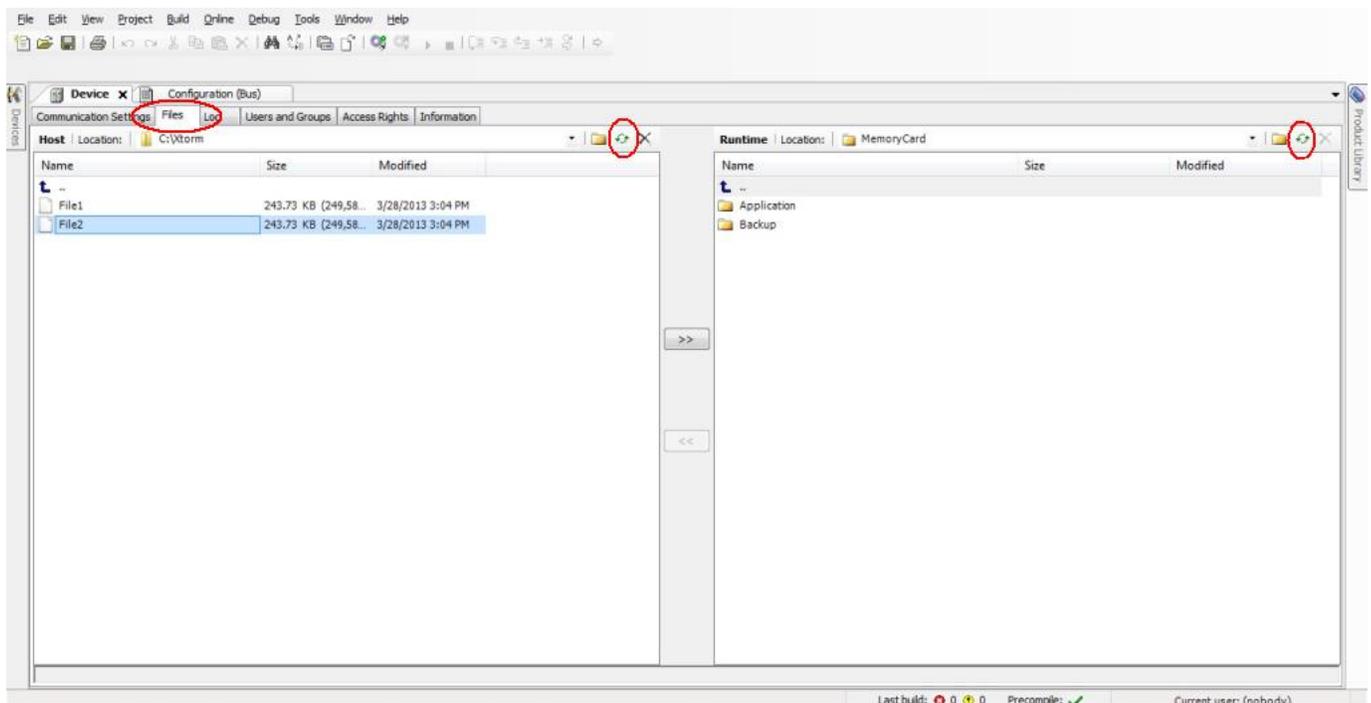


Figura 3-60. Acesso aos Arquivos de Usuário

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 (32 Mbytes), 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.

### ❗ 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 Xtorm 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 a seguir. 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.

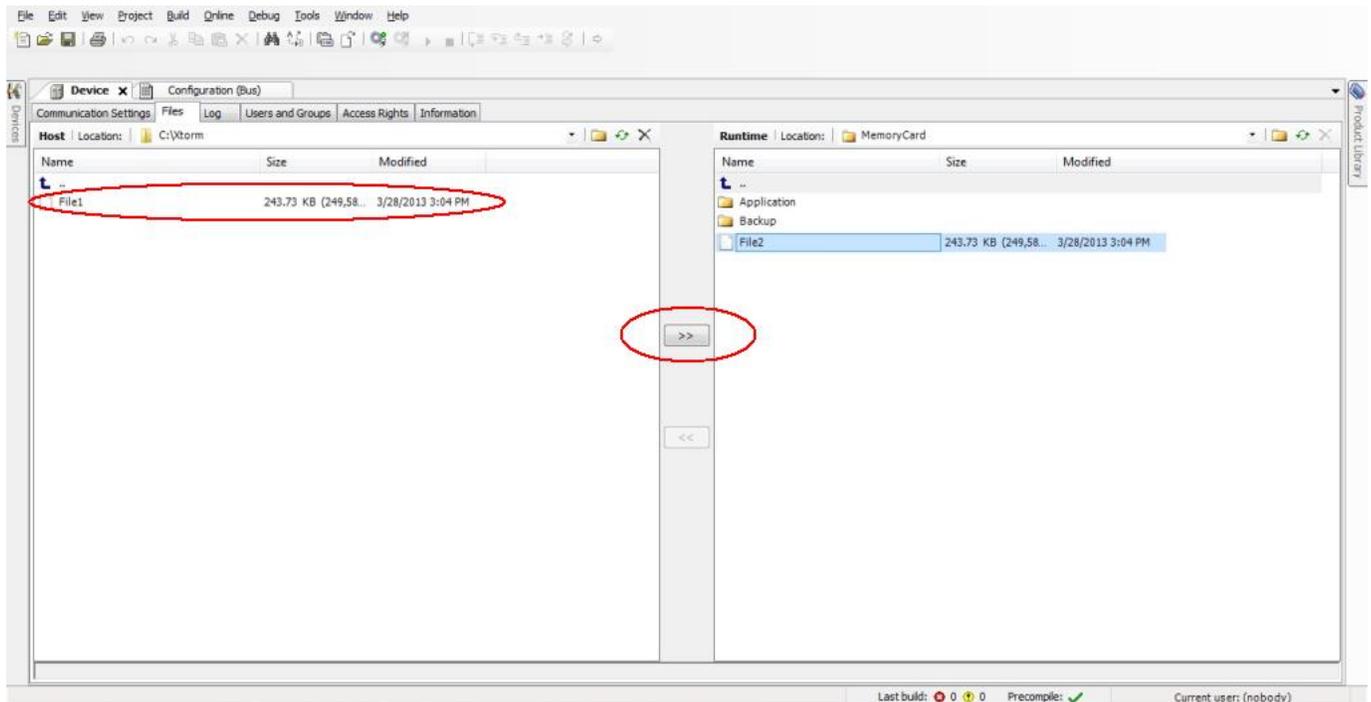


Figura 3-61. Transferindo Arquivos

**NOTA:**

Os arquivos contidos dentro da pasta de um projeto criado pela ferramenta MasterTool Xtorm 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 Xtorm o usuário deve realizar um procedimento muito semelhante ao anterior, ou seja, selecionando o arquivo na coluna da direita e pressionar o botão "<<", localizado no centro da tela.

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

- **Novo diretório**  - Permite a criação de uma nova pasta na área de memória de usuário;
- **Excluir item**  - Permite a exclusão de arquivos nos diretórios da área de memória de usuário;
- **Atualizar**  - Permite atualizar, na tela do MasterTool Xtorm, os arquivos presentes na memória de usuário e no microcomputador.

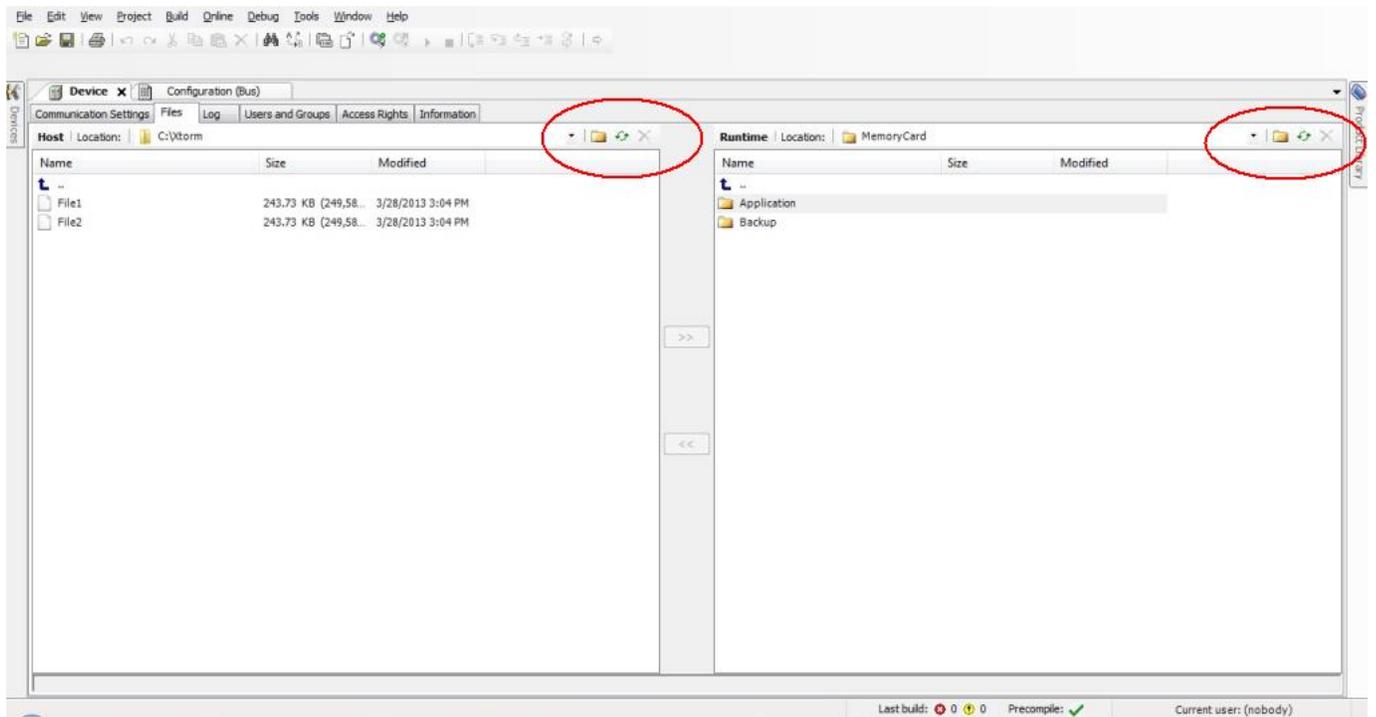


Figura 3-62. Opções de Utilização

**NOTA:**

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

## Configuração do Cartão de Memória



A UCP da Série Hadron Xtorm possibilita 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 dele, 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 reinicializa

### NOTAS:

1. Recomenda-se realizar o procedimento de formatação do cartão de memória diretamente na UCP Hadron Xtorm para evitar possíveis problemas de utilização, aumento do tempo de montagem ou até mesmo funcionamento incorreto.
2. 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.

Durante a configuração do projeto, dentro do software MasterTool Xtorm, o usuário habilita a opção de cópia do projeto da UCP para o cartão de memória ou a cópia do projeto que está no cartão para a UCP. Nessa mesma tela o usuário poderá configurar senhas de acesso, controlando o uso dessas informações.

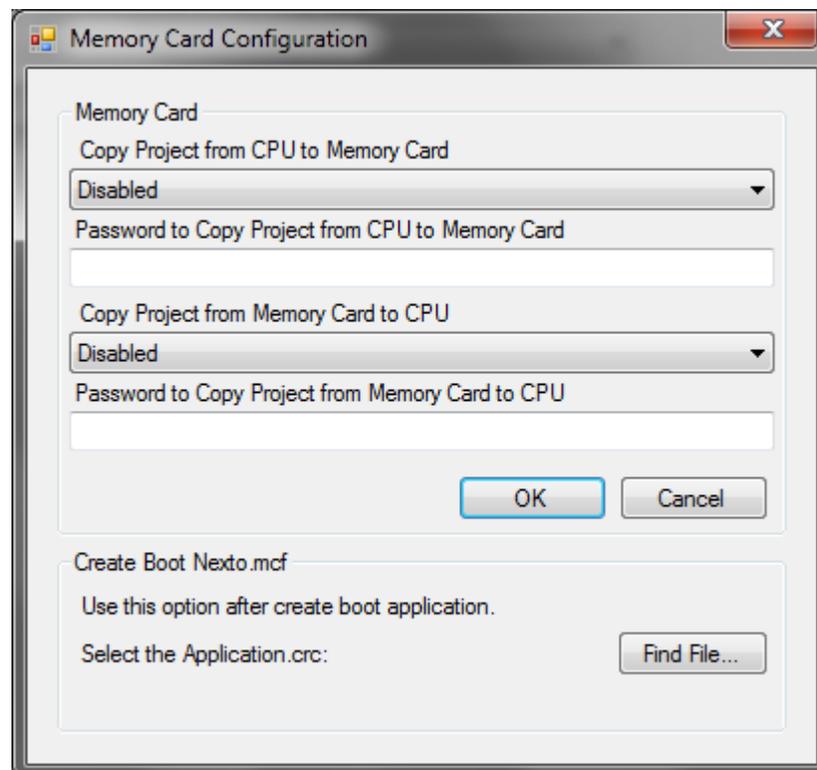


Figura 3-63. Configurações do Cartão de Memória

Quando uma senha é configurada para o cartão de memória no MasterTool, é necessário executar os passos a seguir para que, ao enviar o projeto, o arquivo criptografado que é gerado pelo MasterTool tenha a senha incluída no seu conteúdo e seja enviado.

Primeiramente configurar a(s) senha(s) e clicar no botão “OK”. Neste momento as senhas foram registradas e o próximo passo é executar no menu Comunicação o comando “Criar Aplicação de Inicialização”, lembrando que não se pode estar logado na UCP para realizar esse procedimento. Após executar esse comando, três arquivos são criados. Um com a extensão “app” e outro com a extensão “crc”, a pasta onde o projeto está salvo também criará um arquivo com extensão “.ncf” necessário para a transferência do projeto no cartão para a UCP. Para concluir a operação de configuração da(s) senha(s) é necessário clicar novamente no botão “Cartão de Memória”, que está na configuração dos Parâmetros Gerais da UCP e então localizar o arquivo com extensão “crc” gerado no passo anterior, utilizando o botão “Localizar Arquivo...”. Executados esses passos, o MasterTool Xtorm 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.

Para realizar o envio da UCP para o cartão de memória ou vice-versa, o usuário, além de habilitar no software MasterTool Xtorm e colocar a senha, terá que acessar o Menu “Cartão de Memória” na UCP, utilizando o botão de diagnósticos, e selecionar a opção de transferência desejada. Após, será solicitada a senha, caso o usuário tenha configurado durante a configuração da aplicação. Então, com um pressionamento curto no botão de diagnósticos os dígitos são incrementados e com um pressionamento longo são confirmados. No sexto dígito confirmado, a UCP irá verificar a consistência da senha e iniciará o processo. 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 acessar o Menu “Cartão de Memória” na UCP, utilizando o botão de diagnósticos, e selecionar a opção de remoção do cartão de memória, e aguardar até que o ícone do cartão desapareça da tela de status do visor gráfico.

#### 🔔 NOTAS:

1. Caso o cartão de memória seja removido sem ser desmontado via menu da UCP, durante a transferência de arquivos ou formatação, este processo pode acarretar a perda de dados no cartão, bem como corromper os arquivos nele contido. Este processo pode implicar na necessidade de uma nova formatação do cartão quando inserido novamente à UCP.
2. Caso exista algum arquivo na raiz do cartão de memória nomeado “Application” ou “Backup”, ele será excluído para a criação das pastas de mesmo nome, utilizadas pela UCP para armazenamento do projeto aplicativo 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 Xtorm, sendo ele montado na pasta “MemoryCard”. Os diretórios “Application” 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.

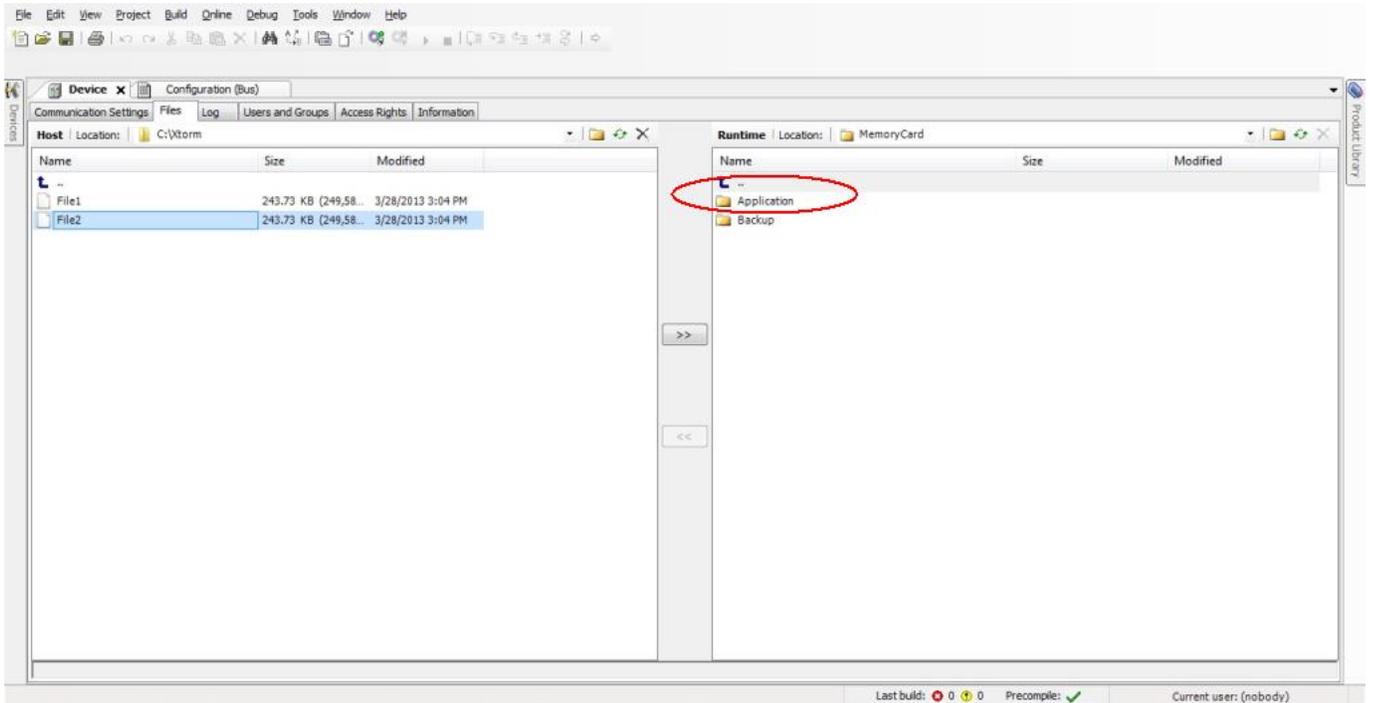


Figura 3-64. Diretório Raiz com Cartão de Memória Inserido

A transferência de arquivos ocorre de maneira semelhante à utilização da memória de usuário (Memória de Arquivos de Usuário), basta acessar o diretório “MemoryCard” e enviar os arquivos, conforme mostra a figura a seguir.

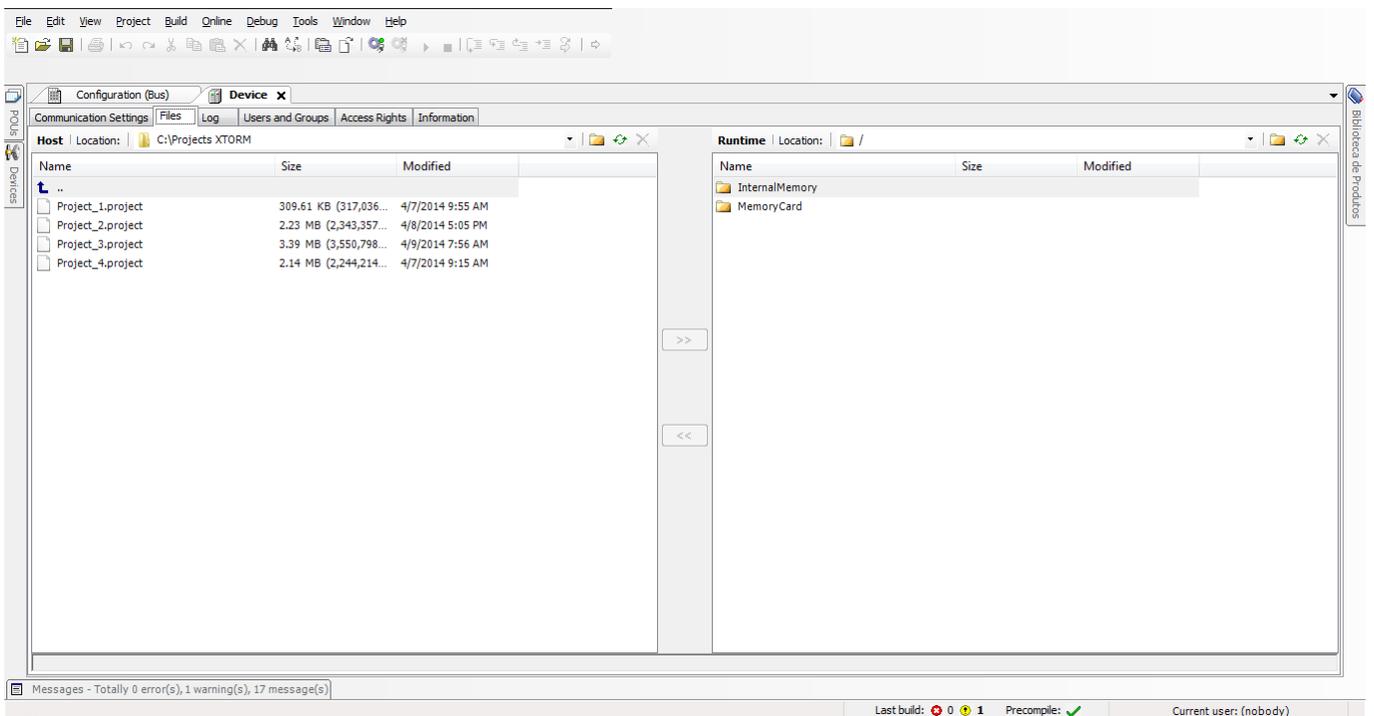


Figura 3-65. Arquivos Salvos no Cartão de Memória

Dentro do diretório do cartão de memória, além dos arquivos que estiverem armazenados no cartão, estarão as pastas “Application” e “Backup”. Nestas pastas ficarão salvos o aplicativo e o

projeto atual caso o usuário escolha transferi-los ou fazer um backup deles através do menu da UCP.

🔔 **NOTAS:**

1. O tempo de transferência de arquivos depende da diferença do tempo de intervalo menos o tempo de execução médio da(s) tarefa(s) em execução (tempo disponível até o próximo ciclo da tarefa), isto é, quanto maior for essa diferença para cada tarefa em uma aplicação, mais rápida deverá ser a transferência de um dado a partir do cartão de memória para a UCP/MasterTool Xtorm ou vice-versa. A transferência de arquivos para o cartão de memória será mais lenta que a transferência para a memória interna da UCP. Para uma UCP em Modo Stop ou sem nenhuma aplicação, a taxa de transferência se aproxima de 100 kbytes/s.
2. O MasterTool Xtorm possui alguns nomes de arquivos reservados que não podem ser transferidos para o cartão de memória, são eles: Application.app, Application.crc, Archive.prj, Stdlogger.csv

## Menu Informativo e de Configuração da UCP



O acesso ao Menu Informativo e de Configuração da UCP Hadron Xtorm, assim como o acesso detalhado aos diagnósticos, estão disponíveis através de níveis, sendo que para acessar as informações do menu, trocar de nível e modificar alguma configuração, basta dar um pressionamento longo no botão de diagnóstico e, para navegar pelos itens de mesmo nível, basta dar um pressionamento curto no botão de diagnóstico.

A figura a seguir descreve um exemplo de como operar o menu da UCP Hadron Xtorm, através do procedimento de ajuste do contraste a partir da tela de Status. Além de facilitar a configuração, é possível identificar todos os níveis de tela e o tipo de pressionamento para navegar entre elas, sendo que para modificar os outros parâmetros, como Idioma e inserir a(s) senha(s) no Cartão de Memória, basta seguir a mesma lógica de acesso. O pressionamento curto mostra que o contraste está sendo incrementado (mais claro), sendo que no próximo pressionamento após o seu valor máximo, ele retorna ao valor mínimo (menos claro). O pressionamento longo mostra a confirmação do contraste desejado e o retorno ao nível anterior.

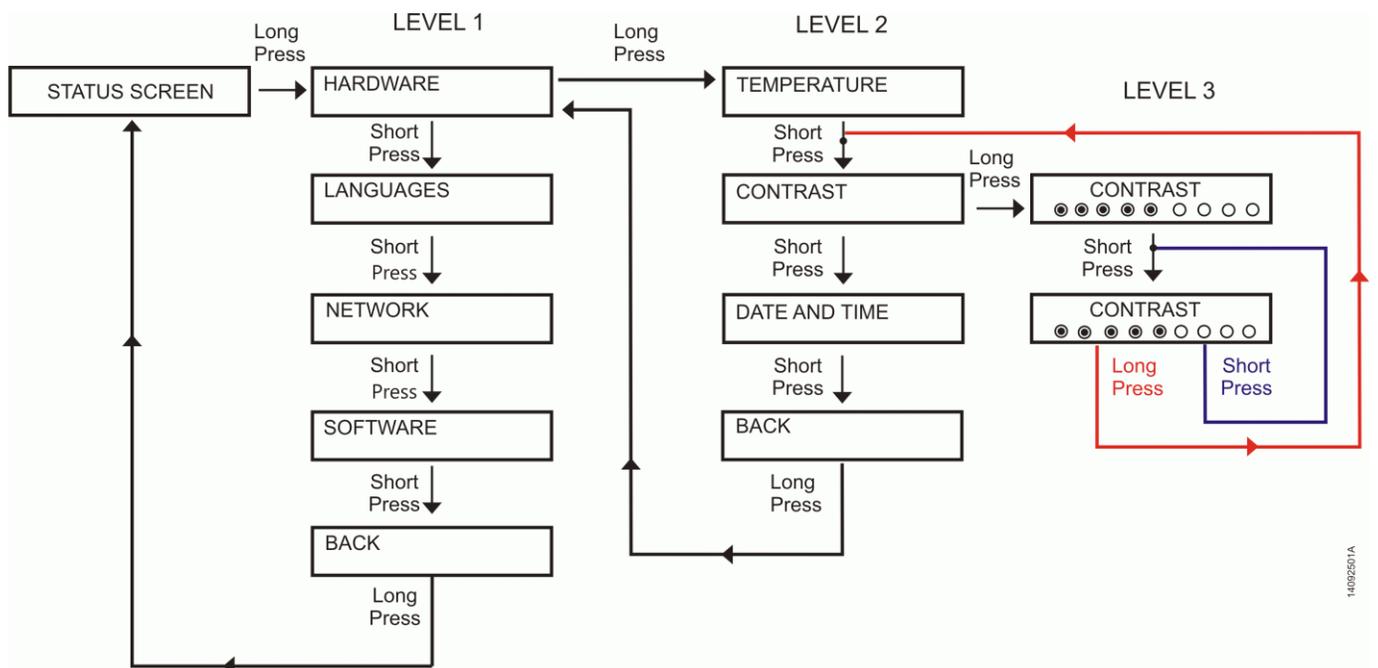


Figura 3-66. Exemplo de Operação: Ajuste do Contraste

Além do menu da UCP Hadron Xtorm ser encerrado através de um pressionamento longo no botão de diagnósticos na tela VOLTAR do nível 1, também existem outras condições de saída, as quais estão descritas abaixo:

- Pressionamento curto, em qualquer momento, nos outros módulos presentes no barramento, faz com que a UCP saia do menu e mostre os diagnósticos do módulo desejado;
- Tempo de inatividade, em qualquer nível, superior a 5 s.

# Configuração via Elementos de Software



## Atualização de Entradas e Saídas

Funcionalidade utilizada para atualizar entradas e saídas no decorrer do aplicativo, não sendo necessário aguardar até que seja completado um ciclo. Quando os blocos funcionais para atualizar as entradas e saídas não são utilizados, a atualização é realizada a cada ciclo da MainTask.

### 🔔 NOTA:

Na inicialização de uma UCP da Série Hadron Xtorm, as entradas e saídas somente estarão atualizadas para leitura e preparadas para escrita quando a MainTask for executada. Todas as demais tarefas do sistema que executarem antes da MainTask estarão com as entradas e as saídas inválidas.

## REFRESH\_INPUT

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

A função de REFRESH\_INPUT deve ser utilizada somente na tarefa MainTask. Caso seja necessária a atualização de E/S em outras tarefas, deve-se utilizar a opção de atualização de E/S por tarefa disponível na tela de configuração da UCP. Além disso, ela não suporta a atualização de entradas que tenham sido mapeadas para variáveis simbólicas. Para o correto funcionamento é necessário que a entrada esteja mapeada para uma variável dentro da memória de variáveis de entrada de representação direta (%I).

A figura a seguir ilustra a declaração dessa função.



Figura 3-67. Função para Atualizar as Entradas

## REFRESH\_OUTPUT

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

A função de REFRESH\_OUTPUT deve ser utilizada somente na tarefa MainTask. Caso seja necessária a atualização de E/S em outras tarefas, deve-se utilizar a opção de atualização de E/S por tarefa disponível na tela de configuração da UCP. Ela não suporta a atualização de saídas que tiverem sido mapeadas para variáveis simbólicas. Para o correto funcionamento é necessário que a saída esteja mapeada para uma variável dentro da memória de variáveis de saída de representação direta (%Q).

A figura a seguir ilustra a declaração dessa função.



Figura 3-68. Função para Atualizar as Saídas

## Timer Retentivo

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

É importante destacar que, para o correto funcionamento dos blocos funcionais Timer Retentivo, as variáveis de controle devem ser declaradas como retentivas (VAR RETAIN).

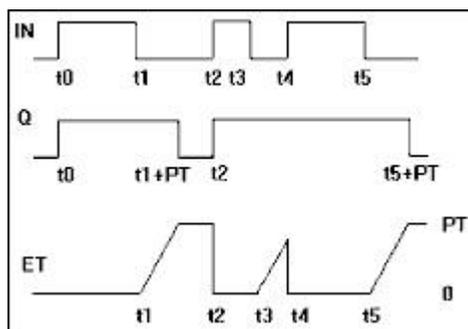
Abaixo, são descritos os três tipos de blocos disponíveis na biblioteca LibPlcStandard do software MasterTool Xtorm.

## TOF\_RET

O bloco funcional TOF\_RET implementa um tempo de atraso para desabilitar uma saída. Quando a entrada IN tem seu estado alterado de verdadeiro (TRUE) para falso (FALSE), ou seja, uma borda de descida, o tempo especificado PT irá transcorrer até que a saída Q também seja falsa (FALSE). Quando a entrada IN tem nível lógico 1 (TRUE), a saída Q também permanecerá no mesmo estado (TRUE), mesmo que isso aconteça no meio de uma contagem. O tempo PT pode ser alterado durante a contagem, pois o bloco funcional assumirá o novo valor, desde que a contagem não tenha chegado ao final. As figuras a seguir mostram a declaração do bloco TOF\_RET e o seu comportamento gráfico.



Figura 3-69. Bloco Funcional TOF\_RET



## TON\_RET

O bloco funcional TON\_RET implementa um tempo de atraso para habilitar uma saída. Quando a entrada IN tem seu estado alterado de falso (FALSE) para verdadeiro (TRUE), ou seja, uma borda de subida, o tempo especificado PT irá transcorrer até que a saída Q também seja verdadeira (TRUE). Quando a entrada IN tem nível lógico 0 (FALSE), a saída Q também permanecerá no mesmo estado (FALSE), mesmo que isso aconteça no meio de uma contagem.

O tempo PT pode ser alterado durante a contagem, pois o bloco funcional assumirá o novo valor, desde que a contagem não tenha chegado ao final. As figuras a seguir mostram a declaração do bloco TON\_RET e o seu comportamento gráfico.



Figura 3-70. Bloco Funcional TON\_RET

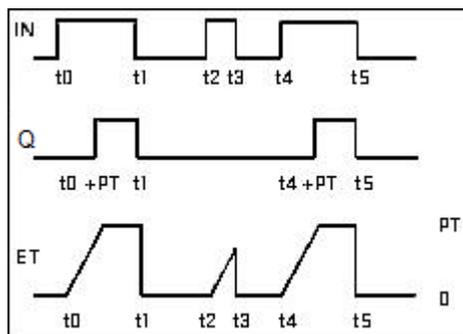


Figura 3-71. Comportamento Gráfico do Bloco Funcional TON\_RET

### TP\_RET

O bloco funcional TP\_RET trabalha como um trigger. O timer, que inicia quando a entrada IN tem seu estado alterado de falso (FALSE) para verdadeiro (TRUE), ou seja, uma borda de subida; é incrementado até que o limite de tempo PT seja atingido. Durante a contagem, a saída Q é verdadeira (TRUE), caso contrário ela é falsa (FALSE). O tempo PT pode ser alterado durante a contagem, pois o bloco assumirá o novo valor, desde que a contagem não tenha chegado ao final. As figuras a seguir mostram a declaração do bloco TP\_RET e o seu comportamento gráfico.

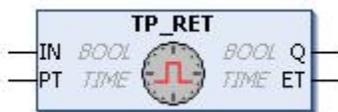


Figura 3-72. Bloco Funcional TP\_RET

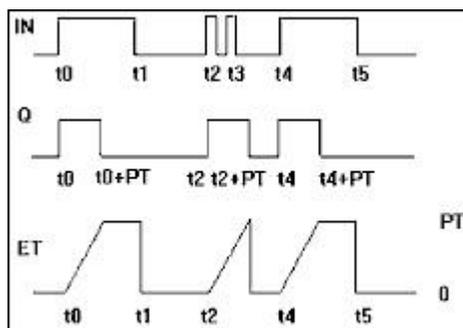


Figura 3-73. Comportamento Gráfico do Bloco Funcional TP\_RET

### Timer Não-Redundante

O timer não redundante é utilizado em aplicações para a UCP HX3040 redundante que necessitam de um timer no programa não redundante em uma arquitetura com redundância de UCP. Este timer não utiliza o timer IEC, portanto, não irá ser sincronizado na arquitetura de

redundância de UCP caso a UCP reserva assumir o estado ativo e o ativo passar para reserva. Os três tipos de blocos já disponíveis na biblioteca LibPlcStandard do software MasterTool Xtorm são:

1. **TOF\_NR** - O bloco funcional TOF\_NR implementa um tempo de atraso para desabilitar uma saída e tem o funcionamento e configuração parecidos com o bloco funcional TOF\_RET, se diferenciando apenas por não ser redundante e nem retentivo.
2. **TON\_NR** - O bloco funcional TON\_NR implementa um tempo de atraso para habilitar uma saída e tem o funcionamento e configuração parecidos com o bloco funcional TON\_RET, se diferenciando apenas por não ser redundante e nem retentivo.
3. **TP\_NR** - O bloco funcional TP\_NR trabalha como um trigger e tem o funcionamento e configuração parecidos com o bloco funcional TP\_RET, se diferenciando apenas por não ser redundante e nem retentivo.

## Log de Usuário

Recurso que permite ao usuário criar seus próprios registros e gravar em arquivos de log no cartão de memória presente na UCP. Os arquivos são gerados em um diretório específico do cartão de memória no formato CSV, permitindo a visualização em editores de texto e planilhas. O separador adotado foi o caractere ponto e vírgula. Há duas funções disponíveis, uma para registrar informações e outra para remover todos os registros. A seguir uma descrição dos tipos de dados utilizados pelas funções:

### UserLogAdd

Essa função é utilizada para adicionar uma nova mensagem de log do usuário, acrescentando em uma nova linha ao arquivo de log no cartão de memória. A mensagem deve ter um tamanho máximo de 150 caracteres e o tipo de evento da mensagem. Variáveis da aplicação podem ser registradas utilizando conversão para string e concatenação com a mensagem principal. É adicionado automaticamente na mensagem a informação de data e hora em UTC (Timestamp) com resolução de milissegundos em que o evento foi registrado. A informação de data e hora também é utilizada na formação dos nomes dos arquivos de log.

A função UserLogAdd pode ser utilizada para inserir várias mensagens dentro de uma mesma tarefa e em tarefas diferentes da aplicação. Porém independente de cada execução da função na aplicação, sendo ela na mesma tarefa ou em tarefas diferentes, todas utilizam o mesmo recurso para gravar as mensagens desejadas. Por esse motivo é recomendado que a adição de mensagens utilizando a função UserLogAdd na aplicação seja realizada a cada 50 ms para evitar retorno de buffer cheio. Caso a função seja executada em períodos menores que o indicado, mas respeite o tempo médio de 50 ms entre cada adição de mensagem ao término do intervalo da tarefa, também evita o retorno de buffer cheio. Para que os logs sejam adicionados corretamente, é importante respeitar os limites de tempo quando o cartão é inserido e na inicialização da UCP, mencionados no capítulo Configuração-Cartão de Memória.

Após a operação a função retorna as opções para o tipo de dado USER\_LOG\_ERROR\_CODES. Quando o retorno da função for diferente de USER\_LOG\_OK, a mensagem não foi registrada, devendo ser executada novamente a função UserLogAdd com a mensagem desejada. Em caso de retorno de falhas consecutivas de escrita, o cartão de memória pode estar danificado. A substituição por um cartão de memória íntegro garante que as últimas mensagens registradas serão gravadas no cartão que não está danificado, desde que a UCP não seja reiniciada.

A figura a seguir mostra a declaração dessa função.

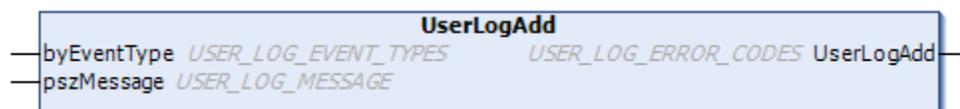


Figura 3-74. Função UserLogAdd

Os arquivos de log são gerados e organizados no cartão de memória em um caminho de diretórios específico que depende do número de série da UCP e da versão de firmware instalada. Por exemplo, para uma UCP com número de série 445627 e versão de firmware 1.1.0.12, o local onde os arquivos de log devem ser gravados no cartão de memória é `MemoryCard/UserLog/445627/1.1.0.12/`.

Os nomes dos arquivos de log são formados pela data e hora (Timestamp) da primeira mensagem. Exceto quando há algum problema para utilizar esse nome, como por exemplo, outro arquivo existente com o mesmo nome, nessa situação é utilizado a data e hora instantânea. O nome do arquivo segue o seguinte padrão: `ano/mês/dia/hora/minuto/segundo/milissegundo.CSV`. Caso um arquivo apresente problema de acesso por setor defeituoso e não seja possível continuar com escrita, será adicionado ao nome desse arquivo a extensão "corrupted" e um novo arquivo será criado. A quantidade de logs por arquivo não é fixa, variando em função do tamanho das mensagens. A quantidade de arquivos criados é limitada em 1024 com tamanho máximo de 1MB cada, portanto o cartão de memória necessita de 1GB de espaço livre. Quando atingir o limite 1024 arquivos criados no cartão de memória, durante a operação da UCP, os arquivos mais antigos são removidos para que os arquivos com logs mais recentes sejam preservados, mesmo nos casos de remoção manual parcial dos arquivos no diretório onde os arquivos estão sendo gravados. A visualização dos arquivos de log pode ser realizada através de planilhas ou editores de texto convencionais. As informações concatenadas, para melhor visualização, podem utilizar ponto e vírgula entre as strings da mensagem para separá-las. Deve-se ter o cuidado na formatação de células com valores em ponto flutuante.

#### UserLogDeleteAll

A função `UserLogDeleteAll` realiza a exclusão dos arquivos de log presentes no diretório criado especificamente para a UCP em que está inserido o cartão de memória, ou seja, são excluídos apenas os logs contidos no diretório nomeado com a versão de firmware da UCP que existe dentro do diretório com a versão de série da UCP. Os arquivos de log excluídos são apenas os arquivos existentes no momento da montagem do cartão de memória e os gerados pela função `UserLogAdd`. Registros de outras UCPs e arquivos adicionados manualmente pelo usuário durante a execução não são apagados.

A figura a seguir mostra a declaração dessa função.



```
void UserLogDeleteAll(  
    USER_LOG_ERROR_CODES UserLogDeleteAll
```

Figura 3-75. Função `UserLogDeleteAll`

#### ❗ NOTA:

O retorno da função `UserLogDeleteAll` não indica operação concluída, apenas confirmação de execução, o que pode levar um longo período no caso de muitos arquivos de log no diretório. A função para registro de novos logs de usuário estará indisponível nesse momento, retornando o código `USER_LOG_PROCESSING` para qualquer operação. O resultado da operação também pode ser verificado no log de sistema.

#### ClearRtuDiagnostic

O bloco funcional `ClearRtuDiagnostic` disponibilizado pela biblioteca `LibRtuStandard` pode ser utilizado quando for necessário limpar os diagnósticos que indicam Perda de Retentividade, Overflow da Fila de Eventos e pode resetar a flag de fila de eventos limpa. A ação que a função executará é definido pelo código do ENUM atribuído na entrada `eDiagnostic`.

#### ClearEventQueue

A função ClearEventQueue disponibilizado pela biblioteca LibRtuStandard pode ser utilizado quando for necessário limpar a fila de eventos da UCP e de todos os drivers instanciados. O status de execução da função será mostrado na sua variável de saída.



#### Pesquisa na Documentação: configuração via elementos de software

Consulte, na documentação do produto, os parâmetros de entrada e saída dos Blocos Funcionais e Funções relacionados às configurações de atualização de entradas/saídas, temporizadores retentivos/não-retentivos e log de usuário.

Fonte: Manual de Utilização Hadron Xtorm – MU223000!



#### Estudo Dirigido 3-3: configuração via elementos de software

Utilizando a linguagem ST implemente os seguintes códigos:

- A. Atualização das entradas de um módulo especificado, sem aguardar o ciclo ser completado;
- B. Atualização das saídas de um módulo especificado, sem aguardar o ciclo ser completado;
- C. Definição de um tempo de atraso para desabilitar uma saída;
- D. Definição de um tempo de atraso para habilitar uma saída;
- E. Definição de um disparador. O timer deve iniciar quando a entrada IN tem seu estado alterado de falso (FALSE) para verdadeiro (TRUE), ou seja, uma borda de subida e é incrementado até que o limite de tempo PT seja atingido. Durante a contagem, a saída Q é verdadeira (TRUE), caso contrário ela é falsa (FALSE);
- F. Definição de um tempo de atraso para desabilitar uma saída (não retentivo);
- G. Definição de um tempo de atraso para habilitar uma saída (não retentivo);
- H. Definição de um disparador (não retentivo);
- I. Adição de uma nova mensagem de log do usuário, acrescentando em uma nova linha ao arquivo de log no cartão de memória;
- J. Exclusão dos arquivos de log presentes no diretório criado especificamente para a UCP em que está inserido o cartão de memória;
- K. Limpeza dos diagnósticos que indicam Perda de Retentividade, Overflow da Fila de Eventos e zeramento da flag de fila de eventos limpa;
- L. Limpeza da fila de eventos da UCP e de todos os drivers instanciados.

#### 🔑 DICAS:

3. Consulte o Instrutor para orientações adicionais;
4. A resolução do estudo dirigido encontra-se no final deste Tutorial.

# Gerenciamento de Usuários e Direitos de Acesso



Fornecem funções para definir contas dos usuários e configurar os direitos de acesso ao projeto. Observe que o gerenciamento de usuários específico do dispositivo deve ser suportado para controlar os direitos de acesso no sistema de arquivos do CP e objetos durante a execução.

Os direitos para acessar os objetos do projeto via ações especificadas são atribuídos somente a grupos e, portanto, cada usuário deve pertencer a um grupo.

## 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.

### NOTAS:

1. 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.
2. Por padrão, em novos projetos, a senha do usuário Owner é vazia.

## Usuários

A figura a seguir ilustra a janela de configuração de usuários.

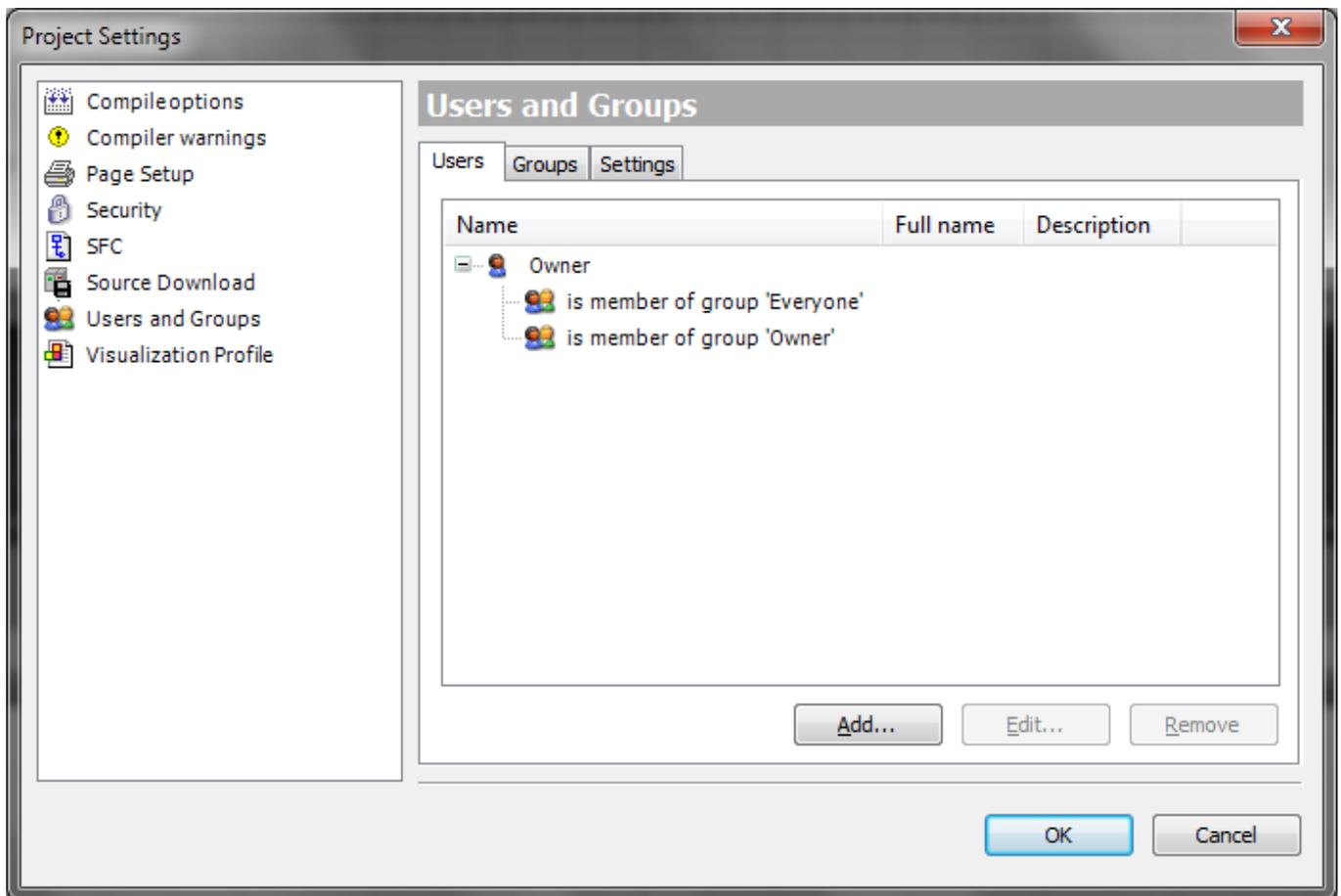


Figura 3-76. Configurações do Projeto, Diálogo Usuários

Os usuários atuais registrados são listados em uma estrutura de árvore. Através dos comandos Add ou Edit 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 Add para abrir o diálogo Add User (ver figura a seguir).

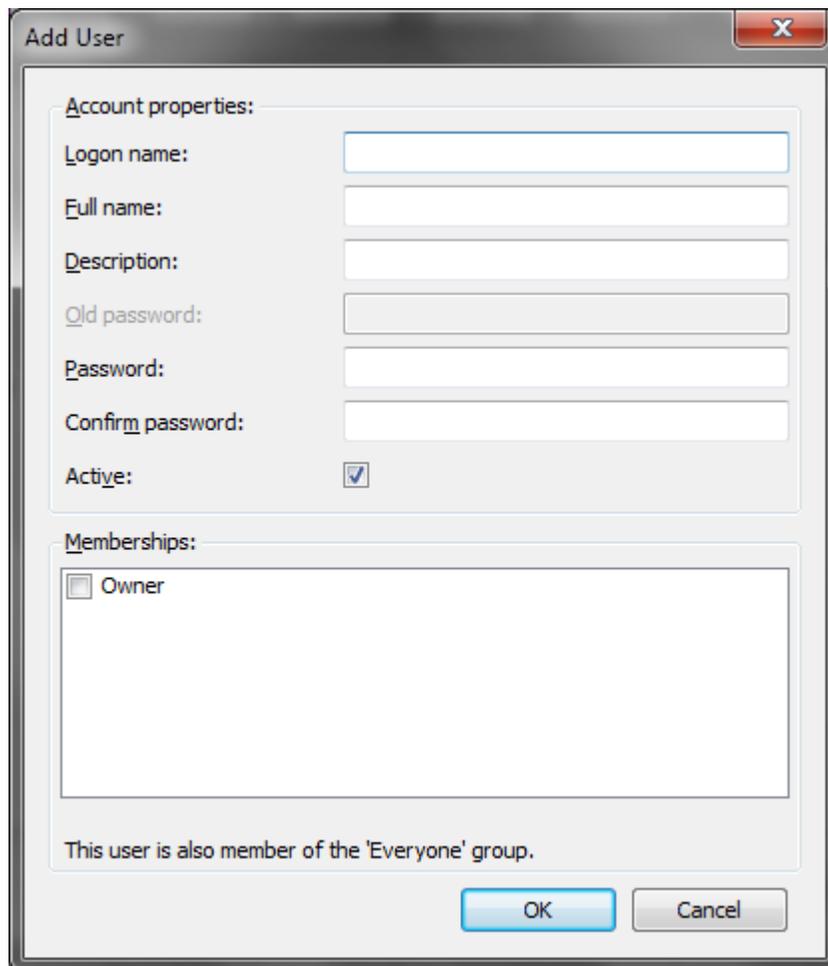


Figura 3-77. Acrescentar Usuário

A funcionalidade Propriedades das Contas, contém os seguintes campos:

- **Logon name** - Nome de logon do novo usuário;
- **Full name** - Nome completo do novo usuário. Serve somente como uma informação adicional;
- **Description** - Descrição do novo usuário. Serve somente como uma informação adicional;
- **Old password** - Este campo é editável somente quando o diálogo é usado para modificar a conta de um usuário existente. Antes de alterar a senha, entretanto, é necessário digitar a senha atual válida;
- **Password** - Senha para o novo usuário. A digitação é representada por asteriscos (\*);
- **Confirm password** - A senha digitada anteriormente deve ser repetida e, caso as duas senhas não coincidam, será mostrada uma mensagem de erro. Esta digitação também é representada por asteriscos (\*);
- **Active** - Esta opção ativa a conta do usuário, tornando-a válida. Quando a conta não é válida, o usuário não pode realizar o logon. A conta será automaticamente desativada quando forem feitas repetidas tentativas de logon com a senha incorreta;
- **Memberships** - Nesta lista são apresentados todos os grupos existentes, além do grupo "Everyone", ao qual pertence o novo usuário automaticamente. Selecionando os respectivos itens no Check Box se define a quais grupos o novo usuário deve pertencer.

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 *Edit* e abra o diálogo correspondente. Estes campos são iguais aos do diálogo *Add User*. 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 teclie *Remove*. 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

A figura a seguir ilustra a janela de configuração de grupos.

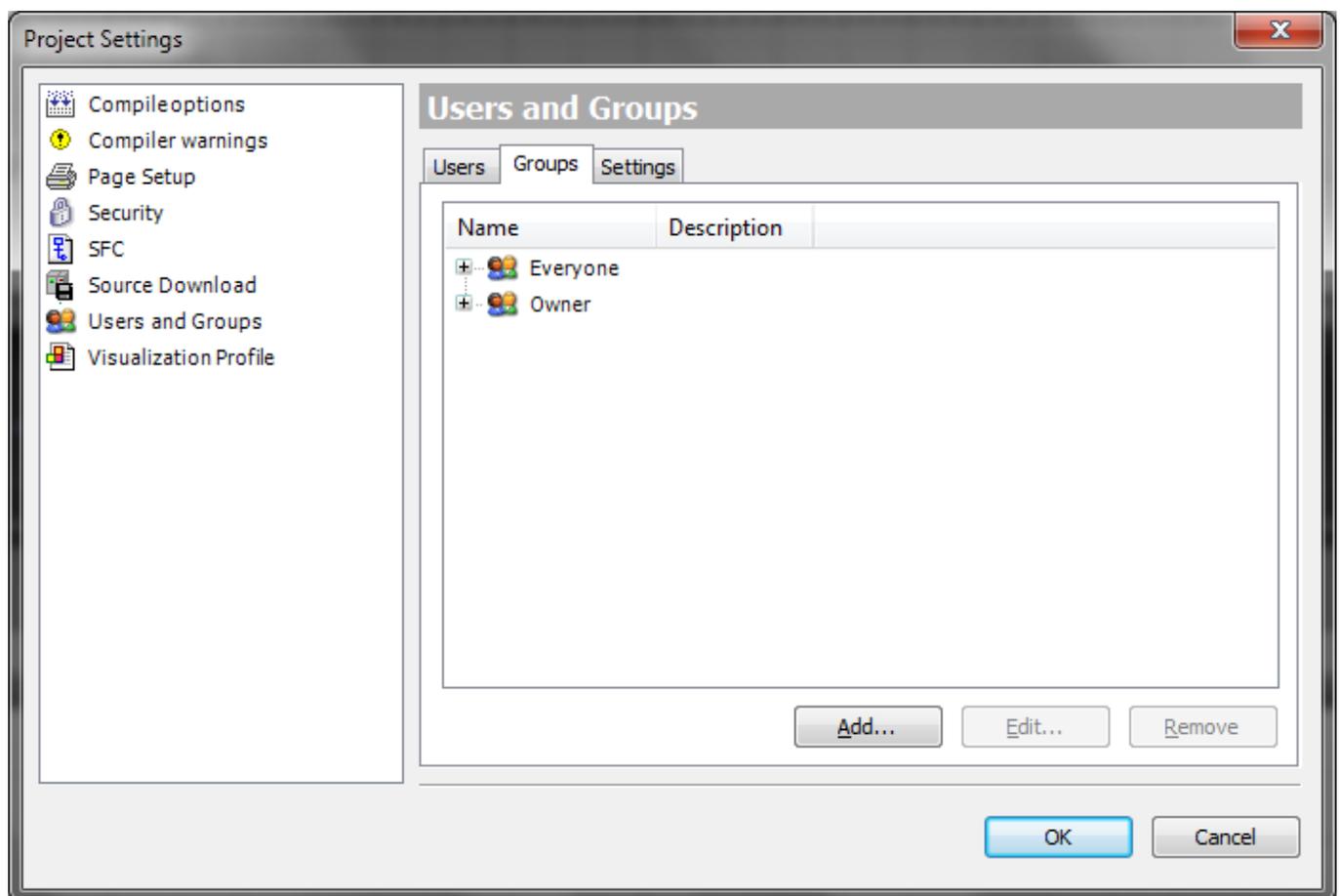


Figura 3-78. Configurações de Projeto, Diálogo Grupos

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.

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

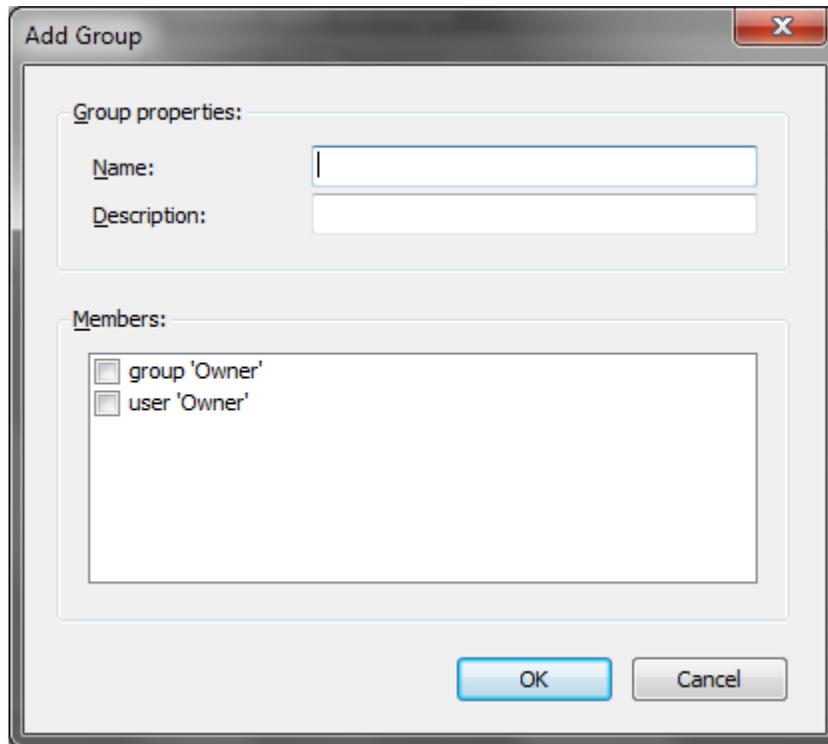


Figura 3-79. Acrescentar Grupo

Os seguintes campos devem ser preenchidos:

- **Name** - Nome do novo grupo;
- **Description** - Descrição do novo grupo. Serve somente como uma informação adicional;
- **Members** - Nesta lista estão apresentados todos os usuários e grupos. Selecione aqueles que devem fazer parte do grupo atual no Check Box.

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 Edit Group. Os campos são iguais aos do diálogo Add Group (figura anterior). 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 tecele Remove. 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

A figura a seguir ilustra a janela de configurações.

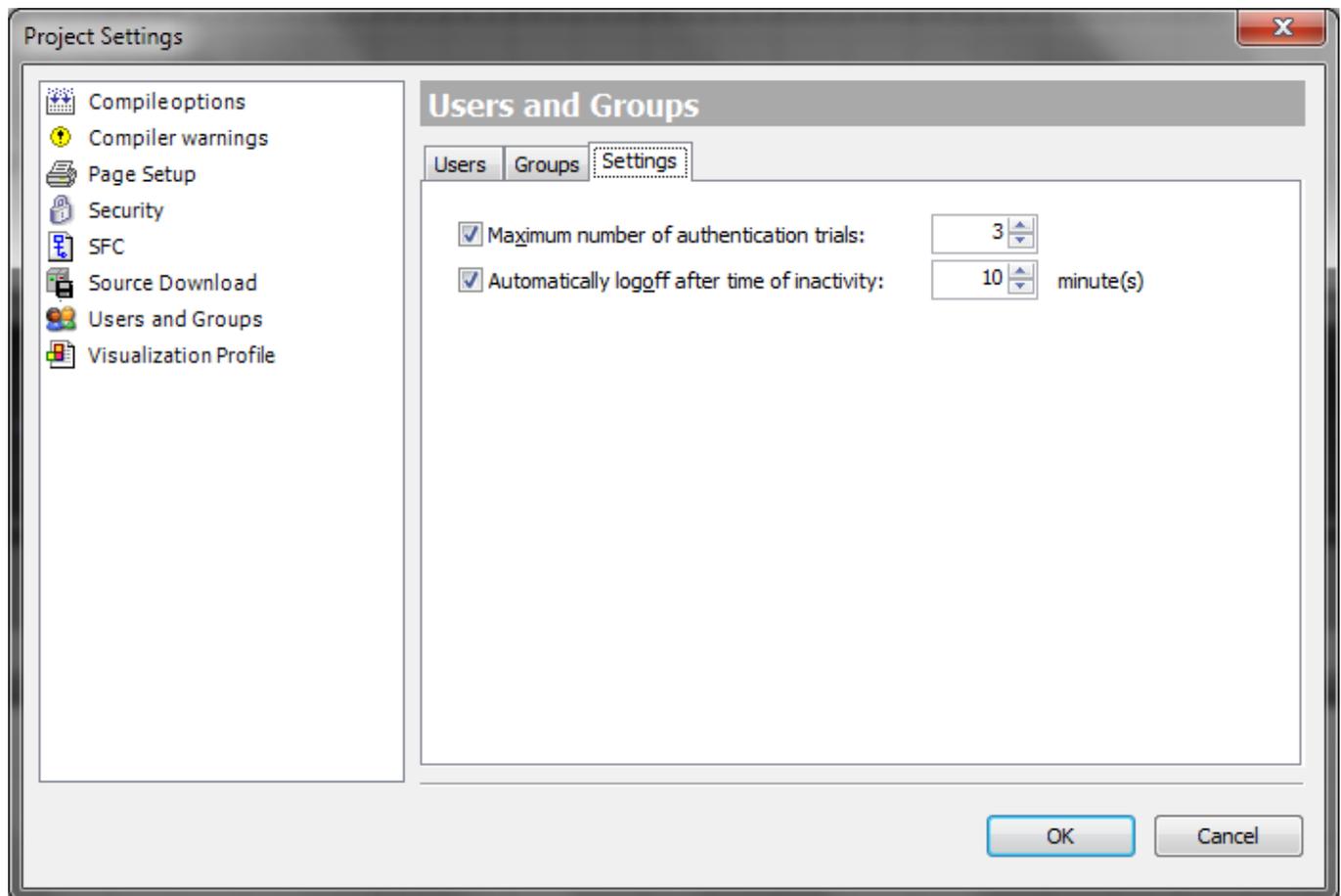


Figura 3-80. Configurações de Projeto, Diálogo Configuração de Usuários e Grupos

As seguintes configurações básicas referentes às contas do usuário podem ser realizadas:

- **Maximum number of authentication trials** - Se esta opção está ativa, a conta do usuário se tornará inválida após o número especificado de tentativas de realizar o logon com a senha incorreta. Se a opção não estiver ativada, o usuário poderá realizar tantas tentativas quantas quiser. Padrão: opção ativada (3 tentativas). Valores permitidos: 1-10;
- **Automatically logoff after time of inactivity** - Se esta opção está ativa, a conexão da conta do usuário será perdida automaticamente após determinado tempo de inatividade (ausência de ações via mouse ou teclado). Padrão: opção ativada (10 minutos). Valores de tempo permitidos: 1-180 minutos.

## 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 *Permissions* ou - para os direitos de acesso aos objetos - no diálogo *Access Control* (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 *Permissions* 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 *Permissions*, 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 *Permissions* deve-se clicar nessa opção no menu *Project > User Management*. Será aberta a tela mostrada na figura a seguir.

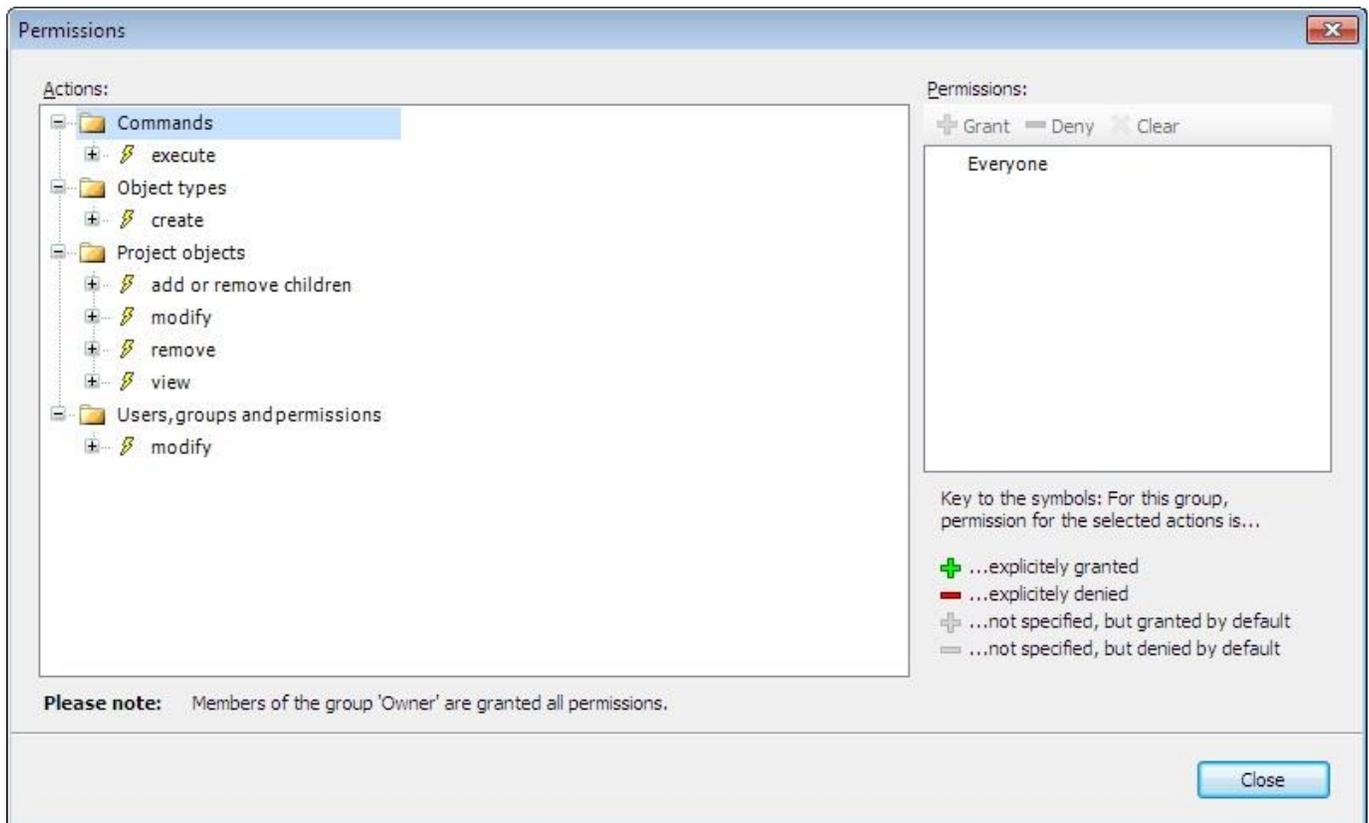


Figura 3-81. Permissões

## Gerenciamento de Usuários e Direitos de Acesso da UCP

As UCPs HX3040 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 Xtorm, 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 *Communication Settings* que será aberta. Apenas as abas *Users and Groups* e *Access Rights* se relacionam com este tópico. A figura a seguir exemplifica os passos para acessar estas abas da UCP.

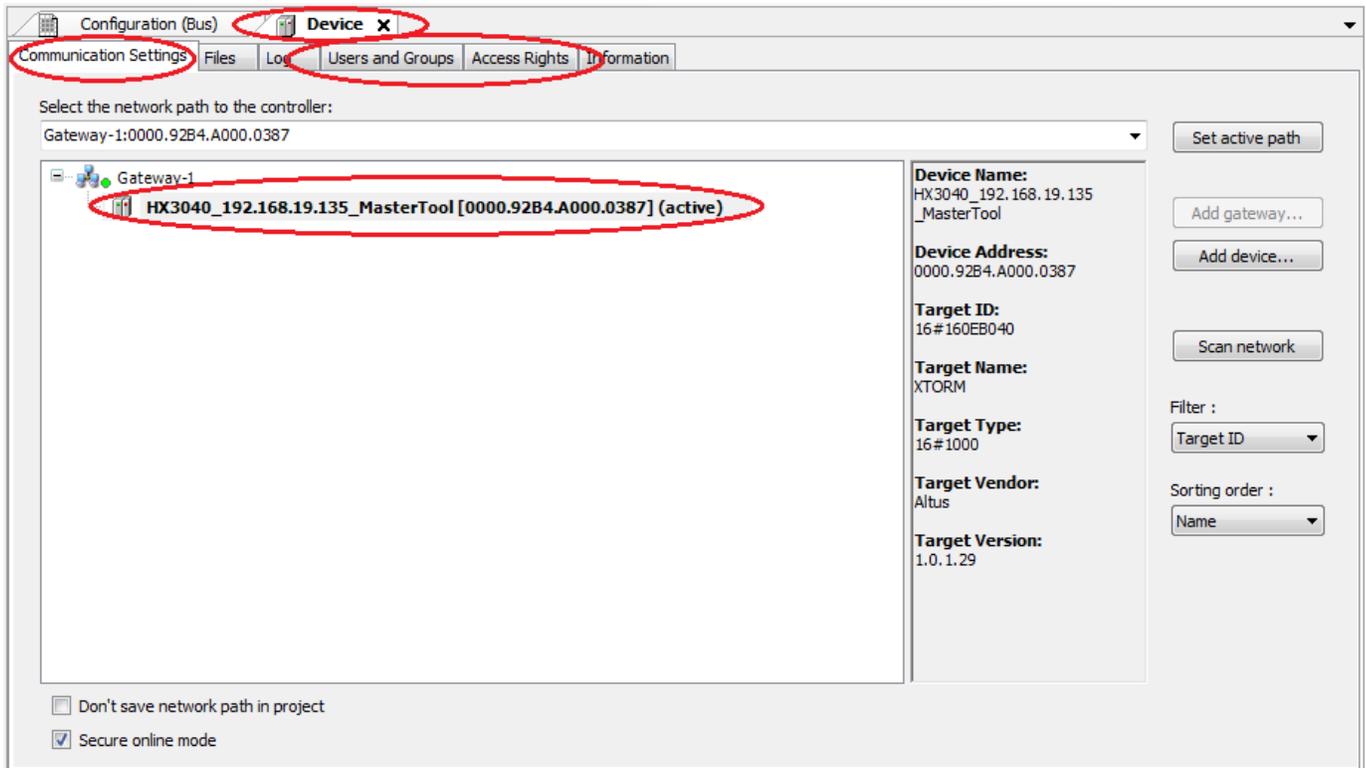


Figura 3-82. Acesso às Abas Usuários e Grupos e Direitos de Acesso

#### ⚠ NOTA:

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

### Usuários e Grupos

O diálogo *Users e Groups* 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.

Para que algumas funções de um controlador possam ser executadas apenas por usuários autorizados, utiliza-se o *Online User Management*. Esta opção fornece a possibilidade de definir contas de usuários, atribuir direitos de acesso para grupos e forçar a autenticação do usuário no login.

O gerenciamento de usuários específico do dispositivo pode ser pré-definido pela descrição deste. Depende também da descrição do dispositivo quais definições podem ser editadas nos diálogos de configuração no programador.

Da mesma forma que no gerenciamento de usuários do projeto, os usuários devem ser membros dos grupos e somente grupos de usuários podem obter determinados direitos de acesso.

Basicamente, o tratamento dos diálogos de gerenciamento de usuários online é similar ao do gerenciamento de usuários do projeto. Há a possibilidade de “importar” definições de contas de usuários a partir do gerenciamento de usuários do projeto.

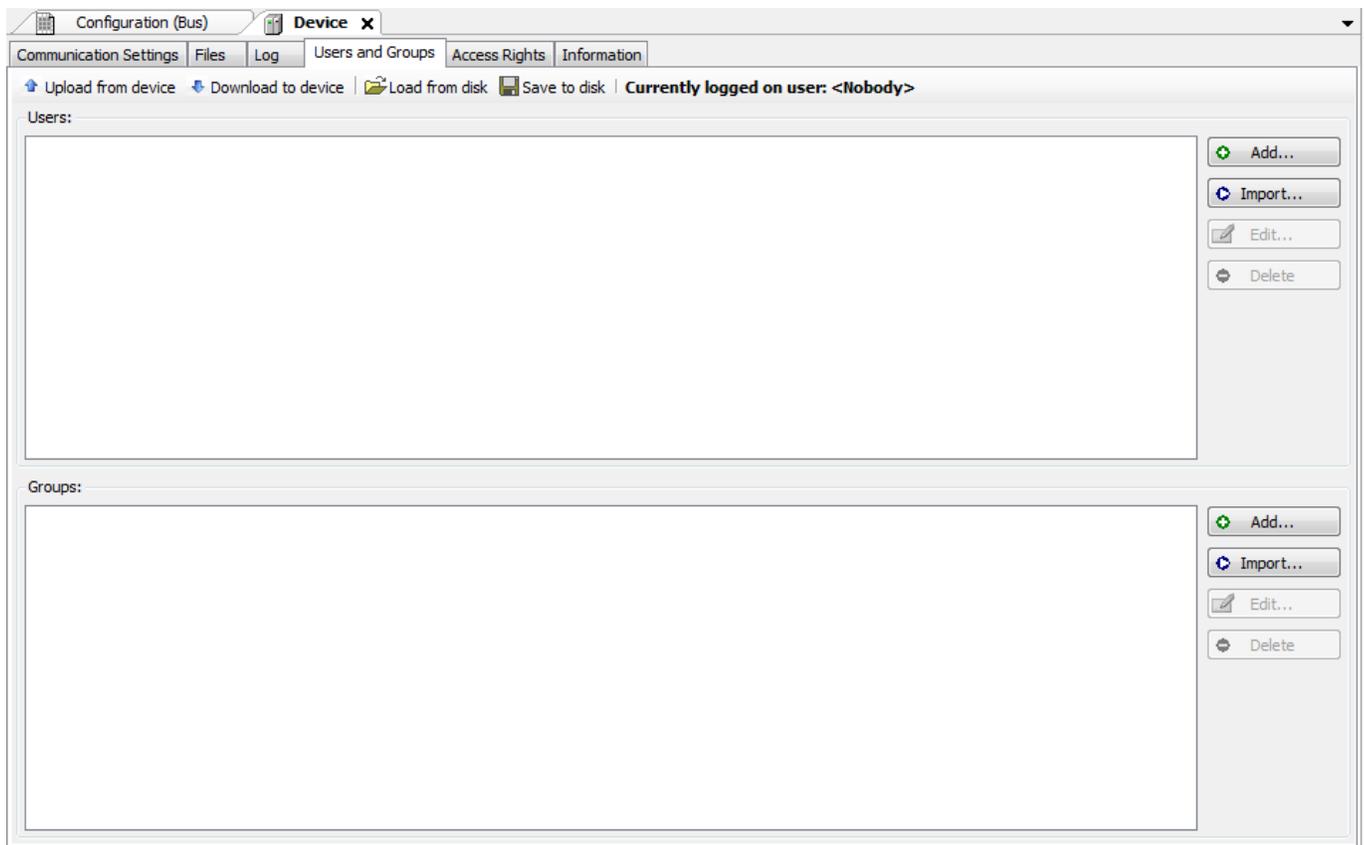


Figura 3-83. Diálogo Device, Usuários e Grupos

### Usuários

Os seguintes botões estão disponíveis para configurar contas de usuários:

-  - O diálogo *Add User* define um nome de usuário e uma senha. A senha deve ser repetida no campo *Confirm password*. Ao abrir esse diálogo os campos *Password* e *Confirm password* estarão preenchidos com caracteres fictícios, o usuário deve substituir esses caracteres por uma senha válida;

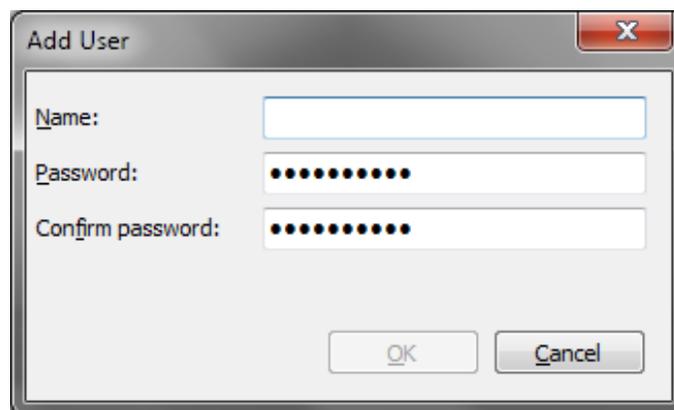
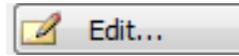


Figura 3-84. Acrescentar Usuário



- O diálogo Import Users mostra todos os nomes de usuários atualmente definidos no gerenciamento de usuários do projeto. Selecione um ou mais itens e confirme com OK. No diálogo Enter Password, digite a senha (conforme definido no gerenciamento) para que a conta do usuário seja importada para o gerenciamento de usuário específico do dispositivo;



- O nome e senha da conta do usuário atualmente selecionada pode ser modificada. O diálogo *Edit User* <nome do usuário> corresponde ao diálogo *Add User*;



- A conta do usuário atualmente selecionada será excluída.

## Grupos

Os seguintes botões estão disponíveis para configurar os grupos:



- O diálogo *Add Group* define um novo nome para este e seleciona, a partir dos usuários atualmente definidos, aqueles que devem fazer parte do grupo;

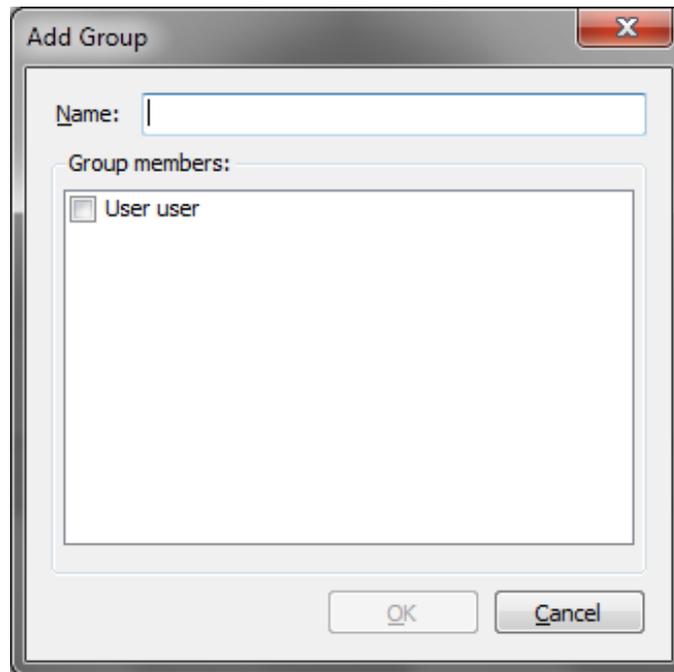
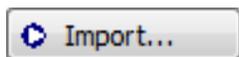
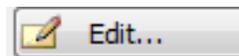


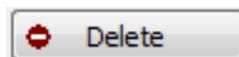
Figura 3-85. Acrescentar Grupo



- O diálogo Import Groups apresenta uma lista com os grupos atualmente definidos no gerenciamento de usuários do projeto. Selecione um ou mais itens e confirme com OK para integrá-los à lista de grupos do gerenciamento de usuários específico do dispositivo;



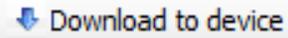
- O grupo atualmente selecionado pode ser modificado no que se refere ao seu nome e usuários associados. Para tanto, usa-se o diálogo *Edit Group* <nome do grupo>, o qual corresponde ao diálogo *Add Group*;



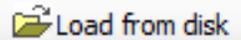
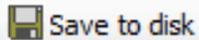
- O grupo atualmente selecionado será excluído.

#### Aplicando e Armazenando a Configuração Atual

Os botões relacionados à aplicação e armazenamento da configuração atual estão na barra superior do diálogo.

- A configuração atual do gerenciamento de usuários deve ser enviada ao dispositivo para torná-la efetiva. A configuração atualmente aplicada no dispositivo pode ser carregada no diálogo de configuração.

- A configuração atual pode ser armazenada em um arquivo xml (\*.dum) e recarregada a partir deste arquivo, o que é útil para configurar a mesma configuração de usuários em vários sistemas. O diálogo padrão para procurar arquivos no sistema será fornecido com esta finalidade. O filtro do arquivo é configurado automaticamente para "\*.dum" (arquivos específicos de "gerenciamento de usuários").

As configurações atuais podem ser impressas ou documentadas através dos comandos *Print...* (menu *Arquive*) e *Document...* (menu *Project*), respectivamente.

#### Considerações sobre Usuários e Grupos Padrão

Nas versões atuais do firmware existem os usuários: *Administrator* e *Everyone*; e os grupos: *Administrator*, *Developer*, *Everyone*, *Service* e *Watch*. Esta divisão em um maior número de grupos visa apresentar uma proposta inicial de diferentes níveis de usuários que possam acessar a UCP. Os seguintes grupos e usuários estão definidos por padrão nas UCPs da Série Hadron Xtorm:

- **Grupo Administrator** - Este grupo possui todos os privilégios e não é possível removê-lo. O grupo *Developer* faz parte deste grupo;
- **Grupo Developer** - Grupo criado para definir direitos de acesso a usuários que sejam desenvolvedores de aplicação. O grupo *Service* faz parte deste grupo. Se não for utilizado este grupo pode ser excluído;
- **Grupo Everyone** - Este é o grupo padrão para realizar os acessos em uma UCP enquanto não existam grupos e usuários definidos;
- **Grupo Service** - Grupo criado para definir direitos de acesso a usuários que prestem algum tipo de serviço nas UTRs, como por exemplo, equipes de manutenção. O grupo *Watch* faz parte deste grupo. Se não for utilizado este grupo pode ser excluído;
- **Grupo Watch** - Grupo criado para definir direitos de acesso a usuários que apenas podem visualizar sem realizar nenhum tipo de modificação na aplicação, se não for utilizado este grupo pode ser excluído;
- **Usuário Administrator** - O usuário *Administrator* está definido nos grupos *Everyone* e *Administrator*. A senha padrão do usuário *Administrator* é "Administrator" e pode ser modificada;
- **Usuário Everyone** - O usuário *Everyone* está definido nos grupos *Everyone* e *Administrator*. Este usuário não possui senha definida.

#### Usuário e Grupos de Projetos Antigos

Para manter os usuários e grupos de projetos antigos em novos projetos após a atualização de firmware da UCP ou em novas UCPs Xtorm é necessário no projeto antigo com o firmware original executar o comando *Upload from Device*, buscando assim as configurações da UCP, e após, o comando *Salvar em Disco*, salvando assim a configuração atual em um arquivo.

Na nova UCP Xtorm ou na UCP atualizada, executar o comando *Load from Disk*, e selecionar o arquivo gerado anteriormente, por fim, executar o comando *Download to Device*, enviando assim as configurações para a UCP.

## Direitos de Acesso

Este diálogo é fornecido em uma guia do diálogo *Device* (Editor do dispositivo). Ele faz parte do *Online Users Management* e serve para conceder e negar certas permissões ao grupo de usuários definido no momento, determinando assim os direitos de acesso a arquivos e objetos (a uma aplicação, por exemplo) quando o UCP está em execução.

Observe que estas permissões somente podem ser atribuídas a grupos e não a usuários únicos. Por isto, um usuário deve estar definido como membro de um grupo. A configuração dos usuários e grupos é feita na guia *Users e Groups* do editor do dispositivo.

A figura a seguir mostra a permissão para adicionar e remover nós secundários para/do objeto PLC Logic concedido ao grupo de usuários *Everyone* e *Owner*.

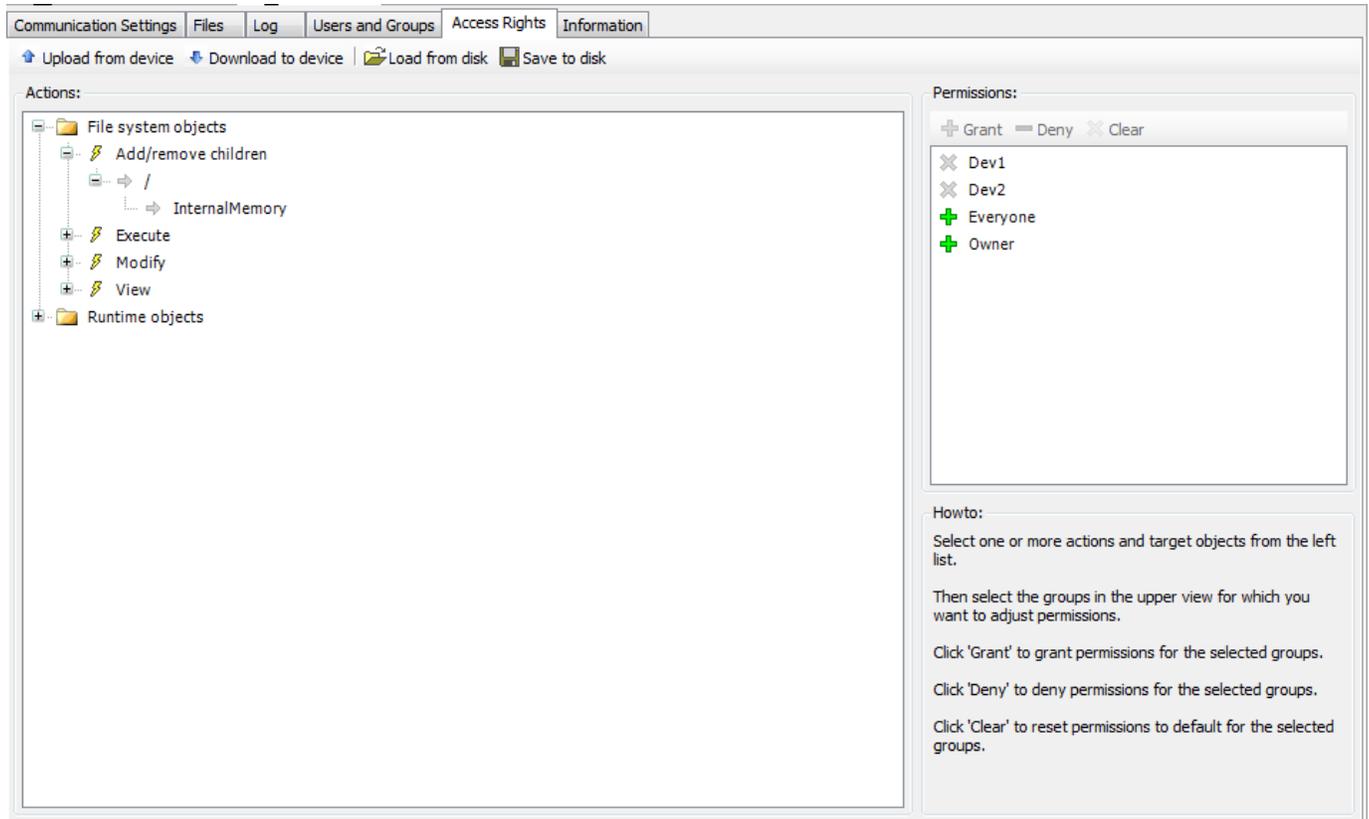


Figura 3-86. Dispositivo, Direitos de Acesso

Os direitos de acesso (acrescentar / remover secundários, executar, modificar e visualizar) são configurados para cada dispositivo e habilitam para cada usuário as ações.

Veja a seguir como definir as permissões de acesso e como fazer com que elas sejam carregadas para o dispositivo ou armazenadas em um arquivo recarregável.

## Definindo as Permissões de Acesso

Para definir a permissão para executar uma ação em um ou vários objetos, selecione-os abaixo do tipo de ação desejada na janela *Action*. Então, selecione o grupo desejado na janela *Permissions* e clique no botão *Grant* ou *Deny* (também na janela *Permissions*).

Veja a seguir a descrição das janelas de diálogo específicas.

#### Ações

Esta parte do diálogo lista as ações que podem ser realizadas durante a execução em arquivos no sistema de arquivos do CP e objetos de execução, como por exemplo, aplicações. A árvore é estruturada da seguinte forma:

-  **Categorias de objetos** - No nível superior, para fins de estruturação, encontram-se as “pastas” referentes aos objetos do sistema de arquivos e objetos de execução.
-  **Tipos de ação** - Nesta “pasta”, existem nós para os quatro tipos de ações passíveis de execução nos objetos específicos. Estes nós servem apenas para fins estruturais:
-  **Add/remove children** (adição ou remoção de objetos “secundários” para um objeto existente)
-  **Execute** (por exemplo, iniciar/parar aplicações, configuração de breakpoints etc.)
-  **Modify** (por exemplo, envio de aplicações etc.)
-  **View** (monitoração)
-  **Objetos** (ação “dispositivos”) - Em cada nó de tipo de ação, encontram-se os “dispositivos” (objetos) da ação (por exemplo,  Dispositivo). Estes objetos mapeados na árvore de dispositivos ou na estrutura do sistema de arquivos, são exibidos de forma estruturada.

#### 🔍 NOTA:

Atribuir um direito de acesso a um “nó principal”, na árvore de objetos, geralmente significa que o “nó secundário” vai herdar esta definição enquanto não receber uma definição específica própria. Entretanto, dependendo do dispositivo, isto pode ser tratado diferentemente. De qualquer forma, as heranças não são visualizadas nos diálogos.

#### Permissões

Este campo mostra os grupos de usuários definidos. Cada grupo é precedido por ícones que indicam a permissão atribuída no momento referente ao dispositivo atualmente selecionado na janela *Actions*.

-  - As ações selecionadas no momento na janela *Actions* são concedidas para o grupo.
-  - As ações selecionadas no momento na janela *Actions* são negadas para o grupo.
-  - Não há definição explícita de direito de acesso para as ações selecionadas no momento, na janela *Actions*.

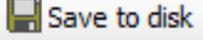
Se várias ações sem configurações exclusivas (em relação ao grupo) estiverem selecionadas, nenhum ícone será exibido.

**Barra de botões** - Após ter selecionado os objetos e o grupo desejados (janela *Actions* e *Permissions*), um dos seguintes botões pode ser usado:

-  **Grant** - Permissão de acesso explicitamente concedida;
-  **Deny** - Permissão de acesso explicitamente negada;
-  **Clear** - O direito de acesso concedido para as ações atualmente selecionadas na janela *Actions* será apagado, ou seja, ele retornará para o padrão.

#### Aplicando e Armazenando a Configuração Atual

Os botões relacionados à aplicação e armazenamento da configuração atual estão na barra superior do diálogo.

-   - As definições dos direitos de acesso configuradas devem ser enviadas para o dispositivo para se tornarem efetivas. As configurações atualmente aplicadas no dispositivo são carregadas no diálogo de configuração.
-   - A configuração atual pode ser armazenada em um arquivo-xml (\*.drm) e recarregada a partir deste arquivo, o qual é útil para definir a mesma configuração de usuário em vários sistemas. Para tanto, será fornecido o diálogo padrão para busca no sistema de arquivo. O filtro do arquivo automaticamente é configurado para "\*.drm" (arquivos de "direitos de acesso de dispositivo").

As configurações atuais podem também ser documentadas em versões impressas via comando *Print* (menu *Archive*) ou *Document* (menu *Project*).

#### Direitos de Acesso de Projetos Antigos

Para manter os direitos de acesso de projetos antigos em novos projetos após a atualização de firmware da UCP ou em novas UCPs Xtorm é necessário no projeto antigo com o firmware original executar o comando *Upload From Device*, buscando assim as configurações da UCP, e após, o comando *Save to Disk*, salvando assim a configuração atual em um arquivo.

Na nova UCP Xtorm ou na UCP atualizada, executar o comando *Load from Disk*, e selecionar o arquivo gerado anteriormente, por fim, executar o comando *Download to Device*, enviando assim as configurações para a UCP.

## 4. Redundância na Hadron Xtorm



### Redundância Hot-Standby



Este capítulo aborda a redundância de UCPs da Série Hadron Xtorm. Trata-se de uma redundância do tipo hot-standby, onde as UCPs são duplicadas no mesmo bastidor. Uma das UCPs normalmente está em estado Ativo e controlando o processo, enquanto a outra UCP normalmente está em estado Reserva, mantendo-se sincronizado com o controlador Ativo. Esta sincronização ocorre através de uma interface de sincronismo localizada no bastidor e dedicada para esta função. Esta interface é redundante, possuindo dois canais independentes (canal A e canal B). Caso ocorra uma falha na UCP Ativa, que o impeça de continuar controlando o processo, a UCP Reserva chaveia automaticamente para Ativa, em um tempo suficientemente baixo para não perturbar o processo, sem causar descontinuidades nas saídas que controlam o processo.

A redundância hot-standby é um método utilizado para aumentar a tolerância a falhas e, conseqüentemente, aumentar a disponibilidade do sistema de automação. A ideia básica é que nenhuma falha simples em componentes duplicados cause a interrupção do controle do processo.

A redundância hot-standby é muito aplicada em:

- Usinas de Geração de Energia;
- Subestações.

Cada UCP pode ter um ou mais protocolos de rede configurados para a comunicação com o centro de controle. Quando ocorre um erro com a UCP Ativa, a UCP reserva assume o controle das conexões ativas com o centro de controle. Com a duplicação destas redes, obtém-se uma disponibilidade ainda maior.

A redundância hot-standby de UCPs da Série Hadron Xtorm não prevê duplicação de módulos de E/S. Caso a redundância de módulos de E/S seja desejável, ela pode ser tratada em nível de aplicação, pelo usuário final. Por exemplo, o usuário pode duplicar ou até mesmo triplicar um módulo de entradas analógicas, e criar um algoritmo de votação para determinar qual das entradas irá ser considerada em determinado momento em sua aplicação.

A figura a seguir mostra um exemplo típico de arquitetura redundante com a UCP HX3040, onde:

1. Topologia de rede Ethernet;
2. A configuração de cada UCP deve ser idêntica.

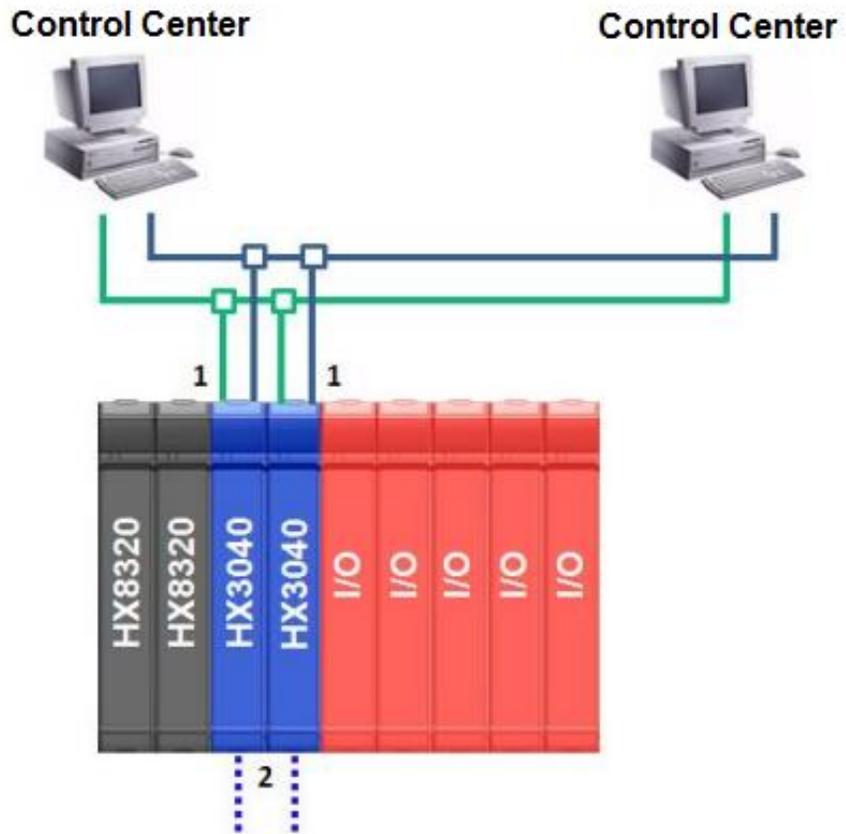


Figura 4-1. Exemplo de Arquitetura Redundante com UCP HX3040

## Configurações da UCP Redundante

Para configurar uma UCP para o modo redundante, é necessário executar os seguintes passos:

1. Criar um projeto selecionando a opção para uso de UCPs redundantes;
2. Para realizar esse procedimento, consultar o capítulo Wizard para Criação de um Novo Projeto Redundante;
3. Configurar as interfaces Ethernet. Para realizar esse procedimento, consultar o capítulo Configuração das Interfaces Ethernet;
4. Opcionalmente, instanciar protocolos DNP e MODBUS. As configurações de cada protocolo podem ser consultadas nas suas respectivas seções deste Tutorial.

### Identificação de uma UCP HX3040

Na sequência deste Tutorial citaremos o par de UCPs redundantes como UCPA e UCPB. Identificamos uma UCP HX3040 que contém um projeto com redundância de UCP como:

- UCPA - A UCP inserida na posição 2 do barramento;
- UCPB - A UCP inserida na posição 3 do barramento.



### Pesquisa na Documentação: características gerais de uma UCP redundante

Consulte, na documentação do produto, as características gerais de uma UCP redundante incluindo os tipos, estados, falhas, desempenho, diagnósticos, comandos, eventos e demais parâmetros associados à redundância.

Fonte: Manual de Utilização Hadron Xtorm – MU223000!

## Princípio de Funcionamento

Nesta seção, descrevem-se as funções de uma UCP redundante, seu comportamento e estados. Também são apresentados conceitos e restrições de programação e configuração.

### Projeto Redundante Único

Fazendo uso da identificação descrita anteriormente, existe um único projeto para a UCP redundante, idêntico para os UCPA e UCPB.

Parâmetros de configuração que devem ser diferentes entre UCPA e UCPB (ex: endereços IP de interfaces Ethernet) aparecem duplicados no projeto da UCP redundante (um para UCPA, outro para UCPB). Cada UCP irá considerar aquele que lhe corresponde, após analisar sua identificação.

### Estrutura do Projeto Redundante

#### Redundancy Template

O projeto de uma UCP redundante é criado automaticamente a partir de um modelo, denominado Redundancy Template. Além disso, são criadas tarefas e POU's básicas do tipo programa, conforme descrito nas próximas seções.

### Tarefas Cíclicas: MainTask e ProtTask

O projeto de uma UCP redundante pode possuir duas tarefas: MainTask e ProtTask (criação opcional no Wizard), que são cíclicas. O usuário pode ajustar o intervalo de cada tarefa.

### Programa MainPrg

A tarefa MainTask está associada a uma única POU do tipo programa, denominada MainPrg. O programa MainPrg é criado automaticamente.

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

```
IF isFirstCycle THEN
  StartPrg();
  isFirstCycle := FALSE;
ELSE
  NonSkippedPrg();
  IF fbRedundancyManagement.bActiveCPU THEN
    EngineeringPrg();
    AlarmPrg();
    UserPrg();
  END_IF;
END_IF;
```

A MainPrg chama outras POUs do tipo programa:

No primeiro ciclo de execução, apenas StartPrg é chamada. Ela é executada tanto na UCP Ativa quanto na UCP Não-Ativa pois as variáveis da MainPrg não são redundantes

Após o primeiro ciclo de execução, NonSkippedPrg sempre é chamada, pois é executada tanto na UCP Ativa quanto na UCP Não-Ativa. Já as POUs EngineeringPrg, AlarmPrg e UserPrg só são chamadas quando a condição “ fbRedundancyManagement.bActiveCPU = TRUE” é verdadeira, ou seja, quando a UCP se encontra em estado Ativa

Portanto, o programa NonSkippedPrg será sempre executado em ambas as UCPs (UCPA e UCPB), independente do estado de redundância desta UCP. Já os programas EngineeringPrg, AlarmPrg e UserPrg serão executados somente na UCP que se encontra em estado Ativa.

Ao contrário do programa MainPrg, que não pode e não deve ser modificado, o usuário pode modificar os outros programas. Inicialmente, quando o projeto redundante é criado a partir do Redundancy Template, estes programas são criados “vazios”, mas o usuário poderá inserir código nos mesmos.

### Programa StartPrg

Esta POU tem por objetivo ser executada somente uma única vez no primeiro ciclo de cada uma das UCPs.

### Programa EngineeringPrg

Esta POU é bloqueada para edição e é gerada automaticamente pelo programa, ela mapeia as variáveis utilizadas para conversões de engenharia, característica da UCP de Conversão de Engenharia.

### Programa UserPrg

O principal objetivo deste programa, que é executado somente na UCP Ativa, é controlar o processo do usuário final.

Este programa normalmente atua sobre as variáveis redundantes, entre as quais encontram-se as variáveis de representação direta %I e %Q associadas ao sistema de E/S local.

### Programa NonSkippedPrg

Este programa, que é executado em ambas UCPs (UCPA e UCPB) independente do estado de redundância, é tipicamente utilizado para funções como:

Gerenciar condições de switchover normalmente não contempladas automaticamente pelas UCPs redundantes, que podem variar de usuário para usuário. Por exemplo, determinado usuário poderá executar um switchover para a UCP Reserva se a UCP Ativa não estiver se comunicando com o sistema SCADA, enquanto outro usuário pode não desejar um switchover nesta situação.

Outras atividades que, por algum motivo, precisam ser executadas tanto na UCP Ativa como na UCP Reserva.

### Programa AlarmPrg

Nesta POU é gerado o código automático responsável pela atribuição do alarme, alarme este que pode ser criado nas configurações da aba de Alarmes da UCP HX3040 .

### Programa ProtPrg

A tarefa ProtTask está associada a uma única POU do tipo programa, denominada ProtPrg. O programa ProtPrg é criado somente caso o usuário escolha seu uso ao criar o projeto, para redundância.

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

```
NonSkippedProtPrg ();  
IF fbRedundancyManagement.bActiveCPU THEN  
UserProtPrg ();  
END_IF
```

A ProtPrg chama outras POUs do tipo programa:

A NonSkippedProtPrg sempre é chamada, pois é executada tanto na UCP Ativa quanto na UCP Não-Ativa e possui prioridade maior que a NonSkippedPrg. Já a POU UserProtPrg só é chamada quando a condição “ fbRedundancyManagement.bActiveCPU = TRUE” é verdadeira, ou seja, quando a UCP se encontra em estado Ativa.

Portanto, o programa NonSkippedProtPrg será sempre executado em ambas as UCPs (UCPA e UCPB), independente do estado de redundância desta UCP. Já o programa UserProtPrg será executado somente na UCP que se encontra em estado Ativa, esta POU tem prioridade maior em relação à UserPrg.

Ao contrário do programa ProtPrg, que não pode e não deve ser modificado, o usuário pode modificar os outros programas. Inicialmente, quando o projeto redundante é criado a partir do Redundancy Template, se selecionado sua criação, estes programas são criados “vazios”, mas o usuário poderá inserir código nos mesmos.

### Programa UserProtPrg

O principal objetivo deste programa, que é executado somente na UCP Ativa, é controlar processos com prioridade superior a UserPrg do usuário final.

Esta POU é executada somente na UCP Ativa e destina-se ao controle do processo do usuário. A "UserProtPrg" é criada somente em projetos com redundância de UCP e possui uma prioridade maior em relação a "UserPrg".

### Programa NonSkippedProtPrg

Este programa, que é executado em ambas UCPs (UCPA e UCPB) independente do estado de redundância com prioridade superior a NonSkippedPrg, é tipicamente utilizado para funções como:

Atividades que, por algum motivo, precisam ser executadas tanto na UCP Ativa como na UCP Reserva com alta prioridade.

### **Variáveis Redundantes e Não-Redundantes**

As variáveis de uma UCP redundante podem ser classificadas entre redundantes e não-redundantes. Variáveis redundantes são copiadas da UCP Ativa para a UCP Não-Ativa, no início de cada ciclo da MainTask, através dos canais de sincronismo entre as UCPs. Por outro lado, variáveis não-redundantes não são copiadas entre as UCPs e, portanto, podem ter valores diferentes na UCP Ativa e UCP Não-Ativa .

As variáveis não-redundantes são utilizadas para armazenar informações privativas de cada UCP (UCPA ou UCPB), tais como diagnósticos de drivers de comunicação e diagnósticos da UCP, incluindo os diagnósticos da redundância (estado da redundância desta UCP etc.).

As variáveis redundantes dizem respeito às informações compartilhadas e relativas ao controle do processo. As variáveis associadas aos módulos de E/S são exemplos típicos de variáveis redundantes.

### **Variáveis %I Redundantes**

Tipicamente as variáveis de entrada de representação direta %I são alocadas para armazenar os sinais lidos dos módulos de entrada digitais e analógicos do barramento local.

Nos projetos com redundância de UCP que possuem módulos de entrada no barramento local, temos uma faixa de variáveis %I que é automaticamente redundante. Esta faixa sempre inicia com a variável %I de endereço 0 e se estende até a variável %I de maior endereço utilizado nos módulos de entrada locais.

Projetos com redundância de UCP que não possuem módulos de entrada no barramento local, não irão possuir variáveis de entrada de representação direta %I redundantes.

### **Variáveis %Q Redundantes**

Tipicamente as variáveis de saída de representação direta %Q são alocadas para armazenar os sinais escritos nos módulos de saída digitais e analógicos do barramento local.

Nos projetos com redundância de UCP que possuem módulos de saída no barramento local, temos uma faixa de variáveis %Q que é automaticamente redundante. Esta faixa sempre inicia com a variável %Q de endereço 0 e se estende até a variável %Q de maior endereço utilizado nos módulos de saída locais.

Projetos com redundância de UCP que não possuem módulos de saída no barramento local, não irão possuir variáveis de saída de representação direta %Q redundantes.

### **Variáveis Simbólicas Redundantes e Não-Redundantes**

Além das variáveis de representação direta (%I, %Q) que são alocadas automaticamente, o usuário pode declarar explicitamente variáveis simbólicas, dentro de POU's ou GVL's. O tamanho máximo permitido para alocação de variáveis simbólicas redundantes é de 512 Kbytes.

#### **¶ NOTA:**

Não se deve confundir variáveis simbólicas com variáveis simbólicas endereçadas através da diretiva AT. Variáveis simbólicas que utilizam a diretiva AT são nomes simbólicos atribuídos as variáveis de representação direta (%I, %Q e %M). Portanto, variáveis que utilizam a diretiva AT não alocam nenhuma memória de variáveis simbólicas.

Variáveis simbólicas são redundantes nos seguintes casos:

1. Quando declaradas em POUs do tipo “programa” criadas na aplicação do usuário, com exceção dos programas NonSkippedPrg e NonSkippedProtPrg;
2. Quando declaradas em GVLs criadas na aplicação do usuário e estas GVLs marcadas como redundantes.

Variáveis simbólicas são não-redundantes nos seguintes casos:

1. Quando declaradas nos programas NonSkippedPrg e NonSkippedProtPrg, já descritos anteriormente;
2. Quando declaradas em POUs do tipo “função”. Observar que tais tipos de POUs normalmente deveriam alocar variáveis apenas na pilha (não estáticas), que conseqüentemente não precisam ser redundantes. Mesmo sabendo que o usuário pode declarar variáveis estáticas (VAR STATIC) dentro de POUs do tipo “função”, isto é considerado uma má prática de programação. Tais variáveis estáticas, caso sejam criadas, serão consideradas não redundantes;
3. Quando declaradas em POUs do tipo “bloco funcional”. Observar que a mera declaração de um “bloco funcional” não aloca memória (o que aloca memória é instanciar um Bloco Funcional).

Deve-se observar que instâncias de blocos funcionais, declaradas dentro de POUs do tipo programa ou dentro de GVLs, comportam-se como variáveis simbólicas, ou seja, alocam memória redundante. Da mesma maneira que variáveis simbólicas, quando instâncias de bloco funcional são declaradas nos programas NonSkippedPrg e NonSkippedProtPrg, ou ainda quando a GVL não é marcada como redundante, tais instâncias são não-redundantes.

## Mapeamentos Múltiplos

Se o usuário desejar mapear as variáveis de comandos da redundância em mais de uma porta de comunicação (COMx ou NETx) será necessária a implementação de um controle pelo usuário na própria aplicação.

A lógica de controle a ser implementada deverá escrever nas variáveis de controle da redundância baseado nos valores das variáveis (comandos) provenientes de cada uma das portas de comunicação (COMx ou NETx). Além disso, a lógica de controle deve reinicializar as variáveis de comandos das portas de comunicação, uma vez que o controle da redundância só reinicializa suas próprias variáveis de comandos.

Segue um exemplo dessa implementação:

```
VAR
var_comando_StandBy_relacao_Ethernet : BOOL;
var_comando_StandBy_relacao_Serial : BOOL;
END_VAR

//Lógica para colocar a UCP Local para StandBy
IF var_comando_StandBy_relacao_Ethernet = TRUE THEN
DG_HX3040_01.RedCmdLoc.bStandbyLocal:=TRUE;
var_comando_StandBy_relacao_Ethernet:=FALSE;
END_IF
IF var_comando_StandBy_relacao_Serial = TRUE THEN
DG_HX3040_01.RedCmdLoc.bStandbyLocal:=TRUE;
var_comando_StandBy_relacao_Serial:=FALSE;
END_IF
```

Acima temos um exemplo de uma lógica em linguagem ST, onde o comando de switchover da redundância pode ser feito por duas variáveis, que provêm de portas de comunicação diferentes em que:

- **var\_comando\_StandBy\_relacao\_Ethernet** - Variável do tipo Bool atribuída a um Coil da comunicação Ethernet que realizará o comando para colocar a UCP Local para StandBy;
- **var\_comando\_StandBy\_relacao\_Serial** - Variável do tipo Bool atribuída a um Coil da comunicação Serial que realizará o comando para colocar a UCP Local para StandBy.
- **DG\_HX3040\_01.tRedundancy.RedCmdLoc.bStandbyLocal** - Este comando produz uma ação equivalente ao comando de "Trocar para Reserva" disparado através do menu do visor, na UCP local.

### Estruturas de Dados de Diagnósticos, Comandos e Usuário

Cada UCP dispõe de diversas estruturas de dados relacionadas com a redundância. As seguintes estruturas pertencem à variável simbólica DG\_HX3040\_01, alocada automaticamente e disponível para o usuário:

- **RedDgnLoc** - Contém diagnósticos da UCP local relacionados com a redundância, como por exemplo, o estado da redundância desta UCP;
- **RedDgnRem** - É uma cópia de RedDgnLoc da outra UCP, recebida via canais de sincronismo da redundância. Desta forma, esta UCP (local) tem acesso aos diagnósticos da outra UCP (remota);
- **RedCmdLoc** - Contém comandos que devem ser aplicados nesta UCP (quando têm sufixo Local) ou na outra UCP (quando têm sufixo Remote). Por exemplo o campo StandbyLocal desta estrutura de dados corresponde a um comando que deve ser executado nesta UCP (local), enquanto o campo StandbyRemote corresponde a um comando que deve ser executado na outra UCP (remota);
- **RedCmdRem** - É uma cópia de RedCmdLoc da outra UCP, recebida via canais de sincronismo da redundância, utilizada somente para visualização ou informação
- **RedUsrLoc** - Contém 128 bytes de dados preenchidos livremente pelo usuário (ex - diagnóstico de comunicação com um sistema SCADA). Estes 128 bytes de dados serão intercambiados com a outra UCP;
- **RedUsrRem** - É uma cópia de RedUsrLoc da outra UCP, recebida via canais de sincronismo.

### Serviços de Sincronização Cíclicos através de Canais de Sincronismo de Redundância

Esta seção descreve os serviços de sincronização que ocorrem ciclicamente em uma UCP redundante, entre UCPA e UCPB, através de dois canais de sincronismo interno feito através do barramento entre as UCPs. Os canais de sincronismo têm por função sincronizar variáveis redundantes, diagnósticos, área de memória de usuário redundante, fila de eventos, sincronização de projetos e comandos.

Estes serviços são executados no início de cada ciclo da MainTask, e na sequência em que aparecem abaixo, ou seja:

1. Primeiro, executa-se o serviço Troca de Diagnósticos e Comandos;
2. Segundo, executa-se o serviço Sincronização de Dados Redundantes.

Estes serviços são executados tanto em STOP quanto em RUN. Modificações e forçamentos feitos em variáveis através da monitoração no MasterTool são sincronizadas entre as UCPs mesmo em STOP.

### Troca de Diagnósticos e Comandos

Este serviço é responsável pelo intercâmbio das seguintes estruturas de dados, em cada ciclo da MainTask:

- Copiar RedDgnLoc da UCPA para RedDgnRem da UCPB;
- Copiar RedCmdLoc da UCPA para RedCmdRem da UCPB;
- Copiar RedUsrLoc da UCPA para RedUsrRem da UCPB;
- Copiar RedDgnLoc da UCPB para RedDgnRem da UCPA;
- Copiar RedCmdLoc da UCPB para RedCmdRem da UCPA;
- Copiar RedUsrLoc da UCPB para RedUsrRem da UCPA.

Este serviço sempre será executado, desde que haja duas UCPs no barramento (nos slots corretos) e desde que a aplicação presente nas duas UCPs seja redundante.

### Sincronização de Dados Redundantes

Este serviço é responsável pela transferência de dados da UCP Ativa para a UCP NÃO-Ativa, o que inclui:

- Fila de eventos;
- Variáveis redundantes (simbólicas e de representação direta - %I e %Q);
- Lista de forçamentos de variáveis redundantes.

Para que este serviço seja executado, diversas condições devem ser satisfeitas:

- O serviço de sincronização anterior deste ciclo da MainTask (Troca de Diagnósticos e Comandos) deve ter sido completado com sucesso;
- Caso esta UCP esteja em estado Ativo, a outra deve estar em estado Não-Ativo. Por outro lado, caso esta UCP esteja em estado Não-Ativa, a outra deve estar em estado Ativa;
- Os projetos das duas UCPs devem estar idênticos. Caso uma UCP seja trocada (troca-a-quente) e a nova UCP não contenha um programa idêntico, essa sincronização somente será executada depois que o Sincronismo de Projeto substituir o programa da nova UCP.

#### 🔔 NOTAS:

1. A sincronização da fila de eventos é realizada sempre para todos os registros de eventos contidos nela, independentemente se estes eventos foram gerados a partir de variáveis redundantes ou de variáveis não redundantes.
2. A lista de forçamentos redundantes contém apenas informações de forçamentos sobre variáveis redundantes. Em cada uma das UCPs (UCPA e UCPB), pode existir uma lista diferente de forçamentos sobre variáveis não redundantes. Forçamentos de variáveis não redundantes não são sincronizados entre os UCPs. Para forçar uma variável não redundante, deve-se fazer o Login no UCP desejado e forçá-la.

### Serviços de Sincronização Esporádicos Através dos Canais de Sincronismo de Redundância

Os seguintes serviços de sincronização são executados de forma esporádica. Ou seja, não são executados a cada ciclo da tarefa MainTask, mas outra tarefa do sistema executa estes serviços esporádicos em segundo plano.

### **Sincronização de Projetos**

Este serviço é responsável por manter sincronizados os projetos das UCPs Ativa e Não-Ativa. Isto ocorre apenas quando os projetos estão diferentes. A sincronização ocorre sempre no sentido da UCP Ativa para a UCP Não-Ativa e basta que um dos canais de sincronismo esteja operacional para que este serviço seja executado. Quando a sincronização estiver habilitada, serão sincronizados os seguintes arquivos e serviços:

- Projeto aplicativo (código executável);
- Project Archive (código fonte);
- Usuários e grupos;
- Direitos de acesso;
- Trace.

O serviço de sincronização irá iniciar em até trinta segundos, após uma das UCPs entrar no estado Ativo, e, após seu início, irá checar o CRC dos projetos a cada cinco segundos.

Quando uma sincronização for iniciada, a UCP Não-Ativa, se estiver em RUN, irá para o modo STOP, no estado de Não Configurado. Após a transferência de todos os arquivos necessários, a UCP Não-Ativa irá para o mesmo modo da UCP Ativa (STOP ou RUN), no estado de Inicializando.

O tempo que a sincronização levará para ser efetuada dependerá do tamanho do projeto. Em média, a taxa de transferência entre os canais de sincronismo é de aproximadamente 500 Kbytes/s.

Caso a sincronização seja interrompida por falha de comunicação entre os canais de sincronismo durante a transferência dos arquivos da UCP Ativa para o Não-Ativo, o procedimento será abortado e reinicializado quando a comunicação for restaurada. Somente após a conclusão de todo o procedimento a UCP Não-Ativa irá para o mesmo modo que a UCP Ativa (STOP ou RUN).

Além de manter sincronizados os projetos, a Sincronização do Projeto também irá impedir que a UCP Não-Ativa assuma estados posteriores a Inicializando, caso o CRC esteja diferente ou algum Online Change esteja pendente na UCP Ativa.

Uma sincronização de projeto terá os mesmos efeitos de um download na UCP Não-Ativa.

### **Redes Ethernet Redundantes com NIC Teaming**

Cada UCP pode ter um ou mais protocolos de rede configurados para a comunicação com o centro de controle conforme exemplo ilustrado na figura a seguir.

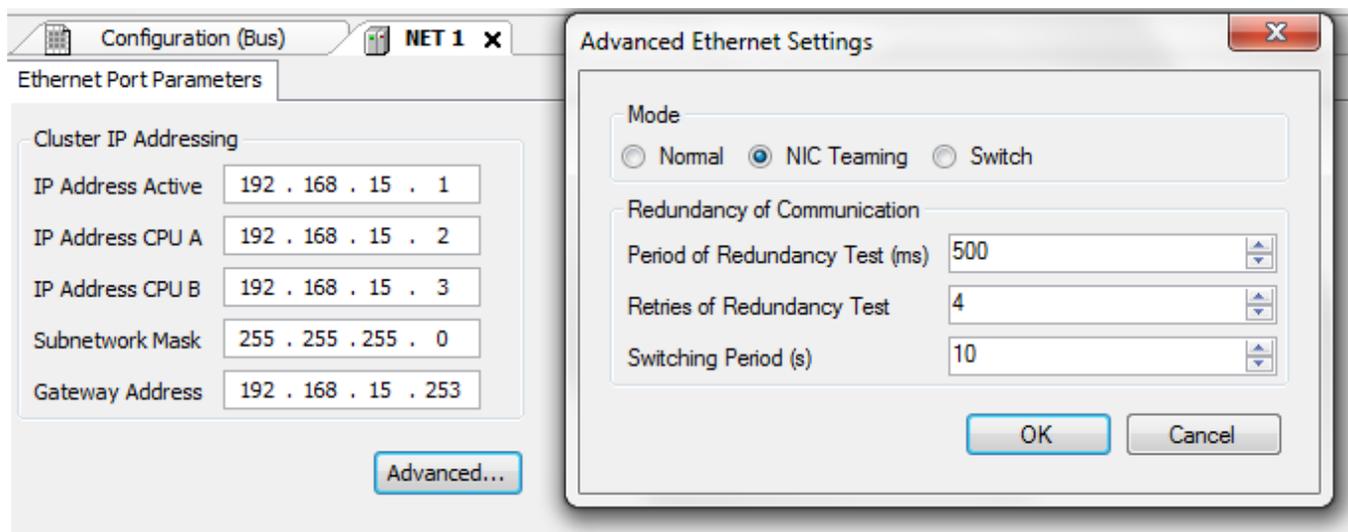


Figura 4-2. Exemplo de rede Ethernet redundante, com NIC Teaming

Neste exemplo a UCP HX3040 conecta-se à rede de supervisão (SCADA). As duas portas Ethernet da UCP HX3040 (NET 1 e NET 2) formam um par redundante NIC Teaming, interligadas em dois switches diferentes (Switch 1 e Switch 2). Em algum ponto, estes dois switches devem ser interligados, para que exista conexão entre as duas portas NIC Teaming e disponibilidade ainda maior (contra falhas duplas).

Tal arquitetura Ethernet possibilita elevada disponibilidade de comunicação do sistema, sendo fortemente indicada para transpor falhas nas portas Ethernet, em cabos e em switches.

📌 **NOTA:**

Quando duas interfaces Ethernet formam um par redundante NIC Teaming, a configuração dos parâmetros básicos e adição de protocolos de comunicação (MODBUS ,DNP3, etc.) possível é feita apenas na primeira interface, sendo que os parâmetros da segunda interface ficam bloqueados para edição.

## Métodos de Troca de IP

Um projeto com UCP redundante da Série Hadron Xtorm dispõe de um método para a troca de IP das portas NET 1 a NET 6. Esse método, chamado de IP Ativo, define o comportamento das portas, no que se refere ao IP dela, de acordo com o estado atual de redundância da UCP (Ativo ou Não-Ativo) e a identificação da UCP em questão (UCPA ou UCPB).

Será necessário configurar três endereços IP por interface NET. Caso a NET esteja trabalhando em NIC Teaming ou modo Switch, será necessário configurar os três endereços IPs para o par de NETs ao invés de fazer uma configuração individual.

### IP Ativo

Esse método é o utilizado nas NETs da UCP HX3040 redundante. Nesse método há um IP para a UCP Ativa e mais dois IPs, um para a UCPA e outro para a UCPB. Nas NETs da UCP HX3040 redundante, o Endereço de IP Ativo será associado à interface da UCP Ativa.

O protocolo de comunicação MODBUS, tanto cliente quanto servidor, utilizam o IP local da UCP, já para o protocolo DNP3 o IP Ativo é usado para se comunicar.

Já para a comunicação com o MasterTool, deve-se usar o endereço de IP específico da UCPA ou UCPB.

Ethernet Port Parameters	
Cluster IP Addressing	
IP Address Active	192 . 168 . 15 . 1
IP Address CPU A	192 . 168 . 15 . 2
IP Address CPU B	192 . 168 . 15 . 3
Subnetwork Mask	255 . 255 . 255 . 0
Gateway Address	192 . 168 . 15 . 253
Advanced...	

Figura 4-3. Método IP Ativo – HX3040 Redundante

Parâmetros que devem ser configurados no método de IP Ativo para as NETs de uma UCP HX3040 Redundante:

- **IP Address Active** - Endereço IP associado à interface quando a UCP estiver em estado Ativo;
- **IP Address CPU A** - Endereço para comunicação com a UCPA, independente do seu estado atual;
- **IP Address CPU B** - Endereço para comunicação com a UCPB, independente do seu estado atual;
- **Subnetwork Mask** - Máscara de subrede das UCPs no barramento Ethernet;
- **Gateway Address** - Endereço do gateway da subrede do barramento Ethernet.

### Uso Combinado de NIC Teaming e IP Ativo

Caso determinado par de portas forme um par NIC Teaming em uma UCP redundante, estas portas podem implementar, ao mesmo tempo, as estratégias NIC Teaming e IP Ativo.

Por exemplo, se as portas NET 1 e NET 2 da UCP HX3040 formam um par NIC Teaming, então:

- **IP Address CPU A** - Endereço IP das portas NET 1 + NET 2 da UCP HX3040 que se encontra na posição 2 do barramento;
- **IP Address CPU B** - Endereço IP das portas NET 1 + NET 2 da UCP HX3040 que se encontra na posição 3 do barramento;
- **IP Address Active** - Endereço IP das portas NET 1 + NET 2 daquela UCP HX3040 que se encontra no estado Ativo.

Desta maneira, associa-se a excelente disponibilidade da estratégia NIC Teaming com a praticidade da estratégia de IP Ativo, que dispensa scripts em sistemas SCADA ou em outros clientes conectados a servidores na UCP Ativa.

### Estados de uma UCP Redundante

Em um sistema redundante, uma UCP (UCPA ou UCPB) pode assumir os seguintes estados:

- Ativo;
- Reserva;
- Inativo;

- Não-Configurado;
- Inicializando.

### 🔔 **NOTA:**

Neste Tutorial utilizaremos a designação “Não-Ativo” para qualquer estado diferente de Ativo, ou seja, para designar qualquer um dos outros quatro estados: Reserva, Inativo, Não-Configurado e Inicializando.

### **Estado Não-Configurado**

Este é o estado de redundância inicial. A UCP encontra-se neste estado de redundância:

- Por convenção, enquanto a UCP estiver desligada;
- Antes de iniciar a MainTask;
- Antes de chavear para o estado Inicializando;
- Caso ocorra uma reinicialização através de um comando como Reset a Quente, Reset a Frio ou Reset Origem.

Caso a MainTask esteja executando no estado Não-Configurado, as seguintes tarefas são executadas:

- Os módulos no barramento não são controlados pela UCP;
- Os serviços de sincronização cíclicos são executados;
- Os serviços de sincronização esporádicos também podem ser executados.

A UCP ficará bloqueada no estado Não-Configurado se o outra UCP estiver em estado Ativo e o projeto deste UCP for diferente do projeto da UCP Ativa. Caso esta situação não se confirme, ocorre uma transição do estado Não-Configurado para o estado Inicializando assim que chegar uma requisição de configuração.

Algumas vezes, a UCP entra no estado Não-Configurado já tendo recebido uma requisição de configuração automática, não sendo necessária uma nova requisição para mudar para o estado Inicializando. Isto ocorre, por exemplo, na energização da UCP.

Outras vezes, o usuário deverá requisitar manualmente esta configuração, por exemplo, usando o menu do visor da UCP. Requisições de configuração manuais tipicamente são necessárias quando alguma manutenção por parte do usuário é necessária, antes de sair do estado Não-Configurado, por exemplo, se a UCP atingiu o estado Não-Configurado devido a alguma falha.

Depois de sair do estado Não-Configurado, a UCP pode voltar a este estado, devido a eventos tais como:

- Reinicialização (Reset a Quente, Reset a Frio ou Reset Origem);
- Desligamento da UCP;
- Diferença de projeto entre esta UCP e a UCP Ativa.

### **Estado Inicializando**

Diferente de todos os outros quatro estados que podem ter duração indeterminada, o estado Inicializando é temporário, durando poucos segundos. Este estado é sempre alcançado a partir do estado Não-Configurado, através de uma requisição de configuração.

Ao entrar no estado Inicializando, diversas ações, testes e verificações são executados, para decidir qual será o próximo estado:

- A UCP apenas escuta o barramento local, sem interferir no controle dos módulos;

- Verifica se a identificação da UCP está correta (deve ser UCPA ou UCPB, de acordo com o onde a UCP está inserida);
- Verifica se há problemas nos parâmetros de configuração extraídos do projeto MasterTool;
- Os serviços de sincronização cíclicos são executados, desde que as condições para sua execução estejam presentes;
- Verifica compatibilidade de versões de firmware entre os dois UCPs;
- Verifica igualdade de projetos entre os dois UCPs;
- Caso a outra UCP esteja em estado Ativo, verifica a possibilidade de estabelecer comunicação passiva com ele, usando o barramento local dos módulos. O modo passivo serve para testar os circuitos de transmissão e recepção e o meio físico, para evitar a ocorrência de uma falha oculta;
- Caso a outra UCP esteja em estado desconhecido devido a falhas no canal de sincronismo de redundância, verifica a possibilidade de estabelecer comunicação passiva com ele, usando o barramento local dos módulos.

Dependendo do resultado destas verificações e testes, a UCP pode ir do estado Inicializando para qualquer um dos outros quatro estados.

### **Estado Ativo**

Neste estado, a UCP controla o processo automatizado, usando o programa UserPrg, executado somente neste estado. A UCP Ativa também controla os módulos de E/S, mantendo-os operacionais, lendo as entradas e escrevendo as saídas.

A UCP Ativa também verifica seus diagnósticos internos e requisições de switchover do usuário para determinar se um switchover é necessário. A UCP normalmente só sairá do estado Ativo se souber que a outra UCP está em estado Reserva, e apta para assumir como Ativo.

No entanto, existem algumas situações em que a UCP Ativa poderá sair do estado Ativo mesmo sem ter certeza de que a outra UCP está no estado Reserva (por exemplo, se esta UCP for removida do bastidor).

### **Estado Reserva**

Neste estado, a UCP está pronta para chavear para o estado Ativo, caso haja uma demanda para isso, tal como uma falha na UCP Ativa atual.

A UCP Reserva também verifica seus próprios diagnósticos, e poderá chavear para o estado Não-Configurado ou Inativo em função de algumas falhas.

O controle do barramento é mantido em estado passivo. O modo passivo serve para testar os circuitos de transmissão e recepção e o meio físico, para evitar a ocorrência de uma falha oculta. Falhas totais causam um chaveamento para o estado Inativo.

### **Estado Inativo**

Este estado normalmente é alcançado depois de alguns tipos de falhas, ou devido a alguma solicitação manual antes de uma manutenção programada.

O controle do barramento é mantido em estado passivo. O modo passivo serve para testar os circuitos de transmissão e recepção e o meio físico, para evitar a ocorrência de uma falha oculta.

Antes de deixar este estado, primeiro deve-se corrigir as falhas diagnosticadas ou executar as manutenções programadas, que causaram a transição para o estado Inativo. Depois, deve-se causar uma transição para o estado Não-Configurado já solicitando uma configuração, para logo em seguida chavear para o estado Inicializando. Depois do estado Inicializando, a UCP pode:

- Retornar ao estado Inativo se determinados tipos de falhas persistem;
- Retornar ao estado Não-Configurado para outros tipos de falhas;
- Ir para o estado Reserva se o outra UCP está no estado Ativo;
- Ir para o estado Ativo se o outra UCP não está no estado Ativo.

### Comandos do menu de redundância do visor da UCP

Esta seção descreve as funções dos comandos do menu de redundância, acessado pelo visor da UCP.

O comando “Trocar para Reserva” tem as seguintes funções:

- Solicitar um chaveamento do estado Ativo para o estado Reserva, o que pode ser útil para executar alguma manutenção programada na UCP Ativa. Depois que o UCP Ativa chaveia para Reserva (e conseqüentemente a UCP Reserva chaveia para Ativo), é possível chaveá-lo de Reserva para Inativo usando o comando “Trocar para Inativo”, e então executar a manutenção programada no estado Inativo;
- Solicitar uma configuração que provoca um chaveamento do estado Não-Configurado para o estado Inicializando, tipicamente depois de reparar falhas que provocaram a transição para o estado Não-Configurado. Depois do estado Inicializando, normalmente espera-se que a UCP vá para o estado Reserva (ou Ativo, se o outra UCP não estiver no estado Ativo);
- Solicitar um chaveamento do estado Inativo para o estado Não-Configurado já solicitando uma configuração. Isto ocorre tipicamente depois de corrigir falhas que provocaram a transição para o estado Inativo. Depois do estado Não-Configurado, a configuração já deve levar ao estado Inicializando. Depois do estado Inicializando, normalmente espera-se que a UCP vá para o estado Reserva (ou Ativo, se o outra UCP não estiver no estado Ativo).

O comando “Trocar para Inativo” solicita um chaveamento do estado Reserva para o estado Inativo, o que pode ser útil para executar alguma manutenção programada na UCP Reserva. Depois desta manutenção, pode-se utilizar o comando “Trocar para Reserva” para tentar voltar ao estado Reserva, passando pelos estados Não-Configurado e Inicializando (ver descrição anterior das funções do comando “Trocar para Reserva”).

#### ¶ NOTA:

Existem maneiras alternativas de gerar os mesmos efeitos dos comandos “Trocar para Reserva” e “Trocar para Inativo”. Podem-se utilizar comandos gerados por esta UCP ou pela UCP remota, conforme descrito preliminarmente na seção Estruturas de Dados de Diagnósticos, Comandos e Usuário.

### Transições entre Estados de Redundância

A figura seguinte mostra a máquina de estados da redundância, ilustrando todas as possíveis transições entre estados de redundância.

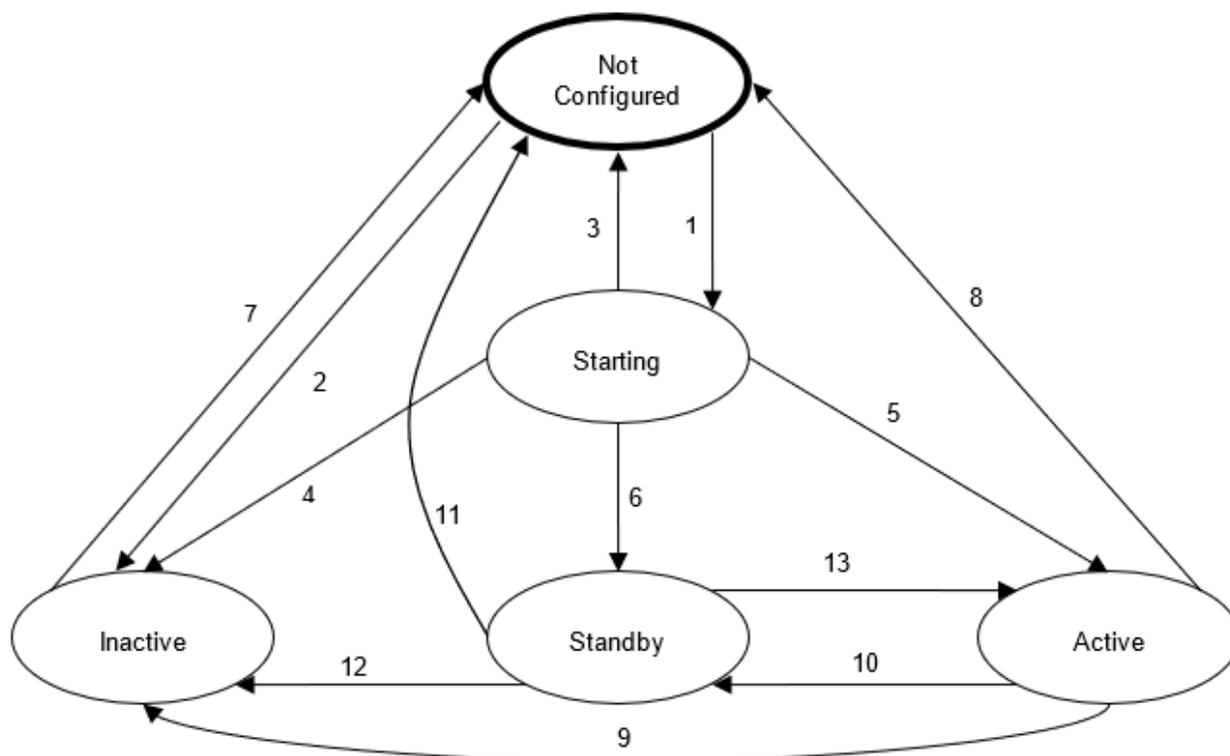


Figura 4-4. Máquina de Estados da Redundância

As seguintes subseções descreverão todas estas transições, e as causas que podem dispará-las. Para interpretar corretamente o funcionamento desta máquina de estados, é preciso estabelecer algumas regras e sequências:

- Transições que se originam do mesmo estado devem ser avaliadas na sequência dada pelo número da transição. Por exemplo, as transições 3, 4, 5 e 6 se originam do estado Inicializando. Neste exemplo, avalia-se primeiro a transição 3, depois a 4, depois a 5 e finalmente a 6. Caso a transição 3 seja disparada, as transições 4, 5 e 6 nem serão avaliadas;
- Dentro de uma subseção específica que descreve uma transição, diversas condições podem disparar esta transição. Estas condições devem ser avaliadas na ordem que aparecem na subseção. Qualquer uma destas condições que se tornar verdadeira pode causar a transição. Se uma condição causar a transição, as próximas condições não serão avaliadas;
- As transições ocorrem tanto com a MainTask em STOP como em RUN;
- Em diversos casos, mencionam-se transições causadas pelos comandos disponíveis no menu do visor da UCP. Deve-se lembrar de que existem alternativas para estes comandos, que são os comandos internos provenientes desta UCP ou da outra UCP (através do link interno de redundância). Nas subseções seguintes, por simplicidade, estes comandos alternativos não serão citados, mas deve-se ter consciência que eles podem causar as mesmas transições do comando disponível no menu do visor;
- Ao executar Reset Origem a aplicação é removida, e sendo assim, o estado da redundância é removido do visor.

#### Transição 1 - Não-Configurado para Inicializando

Um pedido de configuração já existia ao entrar no estado Não-Configurado. Isto ocorre na energização da UCP, e em outras situações, descritas nas próximas subseções ou o comando

de STAND-BY foi solicitado durante o estado Não-Configurado. Isto causa um pedido de configuração manual. O usuário tipicamente solicita o comando de STAND-BY depois de reparar falhas que anteriormente levaram esta UCP ao estado Não-Configurado ou Inativo.

### **Transição 2 – Não-Configurado para Inativo**

Ocorreu alguma exceção de software nesta UCP (cão-de-guarda, access violation, illegal instruction).

### **Transição 3 - Inicializando para Não-Configurado**

As seguintes situações estão associadas a esse tipo de transição:

- Esta UCP foi desligada ou reinicializada (Reset a Quente, Reset a Frio ou Reset Origem);
- A UCP está inserida numa posição incorreta;
- Existem erros lógicos de configuração no projeto recebido do MasterTool Xtorm;
- A outra UCP está no estado Ativo, e a versão de firmware desta UCP é incompatível com a versão de firmware da UCP Ativa;
- A outra UCP está em estado Ativo, e o projeto desta UCP é diferente do projeto da UCP Ativa. Além de ir para o estado Não-Configurado, uma solicitação de configuração é feita. Desta maneira, depois que os projetos forem sincronizados, a UCP sai automaticamente do estado Não-Configurado para o estado Inicializando.

### **Transição 4 - Inicializando para Inativo**

As seguintes situações estão associadas a esse tipo de transição:

- Ocorreu alguma exceção de software nesta UCP (cão-de-guarda, access violation, illegal instruction);
- Algum dos canais de sincronismo do link interno de redundância (canal A ou B) está em falha, e esta UCP sabe que esta falha foi causada por componentes de hardware ou software internos (falhas internas do canal A ou B);
- A outra UCP está em estado Ativo. No entanto, não é possível sincronizar os dados redundantes ou a lista de forçamentos redundantes;
- O estado da outra UCP não pode ser descoberto via link interno de redundância, mas esta UCP consegue monitorar atividade no barramento. Desta maneira, parece que a outra UCP está controlando o processo, embora o link interno de redundância não esteja funcionando adequadamente para ter certeza disso;
- O estado RUN/STOP da aplicação está diferente da UCP Ativa. Por alguma razão de falha, a UCP não sincronizou o estado RUN/STOP da aplicação com a outra UCP Ativa.

### **Transição 5 - Inicializando para Ativo**

A outra UCP encontra-se em estado Não-Ativo. Antes de fazer esta transição, esta condição deve permanecer verdadeira durante algum tempo. Quando a UCPA e UCPB são energizadas simultaneamente, a UCP que terminar a inicialização do sistema antes assume como Ativo. Ou, o estado da outra UCP não pode ser descoberto via link interno de redundância, além disso, a UCP não consegue monitorar atividade no barramento. Portanto, parece realmente que a outra UCP está ausente ou fora de execução (entrou em exceção de software). Por razões de segurança, a UCP em reserva irá chavear para Ativo. Esta condição deve se manter durante algum tempo antes de fazer esta transição.

### **Transição 6 - Inicializando para Reserva**

A outra UCP encontra-se em estado Ativo e os serviços de sincronização de dados redundantes (fila de eventos, variáveis redundantes e lista de forçamentos de variáveis redundantes) estão funcionando corretamente.

### **Transição 7 - Inativo para Não-Configurado**

Esta UCP foi desligada ou reinicializada (Reset a Quente, Reset a Frio ou Reset Origem) ou o comando requisitando transição para STAND-BY foi executado. Além de ir para o estado Não-Configurado, uma solicitação de configuração é feita. Desta maneira, a UCP sai automaticamente do estado Não-Configurado para o estado Inicializando. O usuário tipicamente solicita este comando depois de reparar uma falha que levou a UCP anteriormente ao estado Inativo.

### **Transição 8 - Ativo para Não-Configurado**

Esta UCP foi desligada ou reinicializada (Reset a Quente, Reset a Frio ou Reset Origem).

### **Transição 9 - Ativo para Inativo**

Ocorreu alguma exceção de software nesta UCP (cão-de-guarda, access violation, illegal instruction) ou esta UCP não consegue controlar o barramento, e sabe que a outra UCP está em estado Reserva. Esta condição não é avaliada nos primeiros 2 segundos em estado Ativo.

### **Transição 10 - Ativo para Reserva**

As duas UCPs, por algum motivo, estão no estado Ativo, e este conflito deve ser resolvido. A UCPA chaveará para o estado Reserva caso este conflito dure certo tempo, e a UCPB fará o mesmo após um tempo diferente, que é menor do que o tempo da UCPA. Desta maneira, em caso de conflito, a UCPA tem prioridade para continuar em estado Ativo. Ou, o botão STAND-BY foi pressionado, e esta UCP sabe que a outra UCP encontra-se no estado Reserva. Esta condição não é avaliada nos primeiros 2 segundos em estado Ativo.

### **Transição 11 – Reserva para Não-Configurado**

As seguintes situações estão associadas a esse tipo de transição:

- Esta UCP foi desligada ou reinicializada (Reset a Quente, Reset a Frio ou Reset Origem);
- A outra UCP encontra-se em estado Ativo e o projeto desta UCP é diferente do projeto da UCP Ativa. Além de ir para o estado Não-Configurado, uma solicitação de configuração é feita. Existem casos em que a UCP pode permanecer em Não-Configurado, como quando o Archive é enviado para a UCP Ativa, desta maneira um comando manual de sincronização será necessário. Depois que os projetos forem sincronizados, a UCP sai automaticamente do estado Não-Configurado para o estado Inicializando;
- A outra UCP está no estado Ativo, e a versão de firmware desta UCP é incompatível com a versão de firmware da UCP Ativa.

### **Transição 12 – Reserva para Inativo**

As seguintes situações estão associadas a esse tipo de transição:

- Ocorreu alguma exceção de software nesta UCP (cão-de-guarda, access violation, illegal instruction);
- O comando para transacionar para Inativo foi executado. Isto é feito tipicamente para executar uma manutenção programada na UCP Não-Ativa. Deve-se evitar fazer

manutenções programadas na UCP Reserva, por isso, antes é aconselhável chaveá-lo para Inativo;

- A outra UCP encontra-se em estado Ativo. No entanto, o serviço de sincronização de dados redundantes não funcionou corretamente nos últimos três ciclos da MainTask.

### **Transição 13 – Reserva para Ativo**

As seguintes situações estão associadas a esse tipo de transição:

- O estado da outra UCP é desconhecido devido a falhas no link interno de redundância ou o serviço de sincronização de diagnósticos redundantes não funcionou corretamente nos últimos dois ciclos da MainTask;
- O estado da outra UCP é conhecido e diferente de Ativo;
- Esta UCP não detectou atividades no barramento.

### **Primeiros Instantes em Estado Ativo**

Nos primeiros 2 segundos em estado Ativo, diversas transições que normalmente poderiam tirar a UCP do estado Ativo não são avaliadas (ver subseções anteriores que definem transições a partir do estado Ativo). Por exemplo, durante este tempo, não adianta solicitar o comando de “Trocar para Reserva” para tentar fazer a UCP ir para o estado Reserva.

Apenas duas condições permitem que a UCP saia do estado Ativo nos primeiros 2 segundos. Estas condições são as seguintes:

1. Esta UCP foi desligada ou reinicializada (Reset a Quente, Reset a Frio ou Reset Origem), causando sua transição para Não-Configurado, ou ocorreu alguma exceção de software nesta UCP (cão-de-guarda, access violation, illegal instruction), causando transição para Inativo;
2. As duas UCPs, por algum motivo, estão no estado Ativo, e este conflito deve ser resolvido. A UCPA chaveará para o estado Reserva caso este conflito dure certo tempo, e a UCPB fará o mesmo após um tempo diferente, que é menor do que o tempo da UCPA. Desta maneira, em caso de conflito, a UCPA tem prioridade para continuar em estado Ativo.

Além disso, nos primeiros instantes que uma UCP assume o estado Ativo, alguns diagnósticos que não são redundantes podem não ser válidos. O método para não considerar estes diagnósticos possivelmente “inválidos” é descrito na seção Leitura de Diagnósticos Não-Redundantes.

### **Falhas mais Comuns Causadoras de Switchover Automático entre UCPs**

Nesta seção, são listadas as falhas mais comuns que, de forma automática, causam um switchover da UCP Ativa para Não-Ativo, e da UCP Reserva para Ativo. Estas falhas disparam um subconjunto daquelas transições examinadas na seção Transições entre Estados de Redundância. São elas:

- Falha no circuito interno de alimentação na UCP Ativa;
- Falha no acesso ao barramento do bastidor pela UCP Ativa;
- Falhas na UCP HX3040 da UCP Ativa, tais como
  - Cão-de-guarda;
  - Reinicialização (Reset a Quente, Reset a Frio ou Reset Origem);

- Falha nas interfaces de barramento em um ou ambos os canais de sincronismo NETA e NETB;
- Exceção de software na UCP Ativa (cão-de-guarda de tarefa, erro de acesso em memória etc.).

## Falhas Associadas à Switchover entre UCPs Gerenciados pelo Usuário

Entre as transições examinadas na seção Transições entre Estados de Redundância, algumas possibilitam que o usuário gerencie switchovers entre as UCPs, devido a falhas que normalmente não geram switchovers de forma automática.

Existem casos muito particulares, que dependem da filosofia de cada cliente. Considere-se um exemplo em que o sistema SCADA perde comunicação com a UCP Ativa.

Alguns clientes poderão preferir um switchover manual, onde o operador executa o comando através do menu no visor da UCP. O switchover provoca a retomada da comunicação com a nova UCP Ativa.

Também seria possível a própria UCP Ativa detectar sua perda de comunicação com o SCADA, e ativar um comando em RedCmdLocal, usando as estruturas de dados RedCmdLocal para transportar um comando equivalente ao comando do menu no visor da UCP. Esta seria uma solução totalmente automática e sem intervenção do operador, que tipicamente seria implementada na POU UserPrg.

Através de estruturas de dados como aquelas citadas na seção Estruturas de Dados de Diagnósticos, Comandos e Usuário, é possível trocar diagnósticos e comandos entre as UCPs via link interno de redundância e, desta maneira, o usuário pode executar gerenciamentos especiais de redundância, para falhas que normalmente não causariam switchovers.

Abaixo é exemplificado como o usuário pode gerenciar falhas e executar um switchover devido a um erro de "link down" (cabo de rede rompido) nas interfaces Ethernet da UCP Ativa (o código deve ser utilizado na POU UserPrg) e por controle de erros de comunicação de protocolo das portas seriais:

```
//Verifica se NIC Teaming está habilitado ou não.
IF ((DG_HX3040.tDetailed.Ethernet.NET[1].szIP = '0.0.0.0') OR
(DG_HX3040.tDetailed.Ethernet.NET[2].szIP = '0.0.0.0')) THEN
//NIC Teaming habilitado: erro nas duas NETs para executar switchover.
IF (DG_HX3040.tDetailed.Ethernet.NET[1].bLinkDown AND
DG_HX3040.tDetailed.Ethernet.NET[2].bLinkDown) THEN
//Coloca a UCP Local em StandBy.
DG_HX3040_01.RedCmdLoc.bStandbyLocal := TRUE;
END_IF
ELSE
//NIC Teaming desabilitado: erro em uma NET para executar switchover.
IF (DG_HX3040.tDetailed.Ethernet.NET[1].bLinkDown OR
DG_HX3040.tDetailed.Ethernet.NET[2].bLinkDown) THEN
//Coloca a UCP Local em StandBy.
DG_HX3040_01.RedCmdLoc.bStandbyLocal := TRUE;
END_IF
END_IF
```

```
-----
IF ((DG_HX3040.tDetailed.Serial.COM[1].byProtocol <> 0) OR
(DG_HX3040.tDetailed.Serial.COM[2].byProtocol <> 0)) THEN
//Se ocorre de comunicação para executar switchover.
IF MODBUS_Device_REQDG_0001.byStatus.bCommError THEN
//Coloca a UCP Local em StandBy.
DG_HX3040_01.RedCmdLoc.bStandbyLocal := TRUE;
```

END\_IF  
END\_IF

#### NOTAS:

1. Quando duas interfaces formarem um par NIC Teaming, a interface inativa sempre terá o endereço de IP 0.0.0.0. Este não é um endereço de IP válido e não é possível configurar manualmente uma interface com este endereço.
2. O diagnóstico MODBUS\_Device\_REQDG\_0001 usado na lógica anterior pode ter variação de nome conforme o usuário cria estes diagnósticos, aconselha-se o uso de todas as variáveis de diagnóstico de cada requisição usado no projeto nesta lógica.

## Tolerância a Falhas

O objetivo principal de uma UCP redundante é o aumento da disponibilidade do sistema. A disponibilidade é a razão entre o tempo em que o sistema está funcionando corretamente e o tempo total desde a implantação do sistema. Por exemplo, se um sistema foi implantado há 10 anos, e durante este período esteve parado no tempo total de um ano devido a falhas, então sua disponibilidade foi de apenas 90%. Disponibilidades desta ordem são geralmente inaceitáveis para sistemas críticos, sendo que valores da ordem de 99,99% ou ainda superiores podem ser solicitados nestes sistemas.

Para atingir disponibilidades desta ordem, são necessárias diversas estratégias:

1. Utilização de componentes mais confiáveis (com alto MTBF, ou tempo médio entre falhas), o que contribuirá para o aumento do MTBF do sistema como um todo;
2. Utilização de redundância pelo menos para os componentes mais críticos ou componentes com menor MTBF, de tal maneira que a falha de um componente possa ser tolerada sem parar o sistema. Se a redundância for implementada através da duplicação de componentes, será necessário que os dois falhem para que o sistema como um todo fique indisponível;
3. Alta cobertura de diagnósticos, em especial de componentes redundantes. A redundância de componentes é pouco útil para o aumento da disponibilidade quando não se pode descobrir que um componente redundante falhou. Neste caso, a primeira falha em um dos componentes ainda não compromete o sistema, mas por permanecer oculta, algum dia acontecerá a segunda falha que comprometerá o sistema, já que a primeira falha ainda não foi reparada. As falhas podem ser classificadas entre diagnosticáveis e ocultas, sendo altamente desejável que todas as falhas de componentes redundantes sejam diagnosticáveis;
4. Também é importante que componentes não redundantes tenham ampla cobertura de diagnósticos, pois muitas vezes o sistema pode continuar funcionando mesmo com a falha de um componente não redundante. O componente pode não estar sendo solicitado. Por exemplo, um relé com contato aberto, que raramente tem sua bobina acionada, não tem sua falha detectada até o momento em que o sistema solicitar seu fechamento;
5. Baixo tempo de reparo de componentes não redundantes. A falha de um componente não redundante pode comprometer o sistema, e durante o reparo, o sistema estará indisponível;
6. Possibilidade de reparar ou substituir um componente redundante sem parar o sistema. Se esta possibilidade existe, obtém-se um grande aumento de disponibilidade. Do contrário, deve-se programar uma parada do sistema para substituir o componente, e o tempo de reparo conta como tempo indisponível;
7. Baixo tempo de reparo de componentes redundantes. A falha de um componente redundante não compromete o sistema, mas durante seu reparo, eventualmente pode ocorrer uma falha em seu par redundante. Por este motivo, é importante que a falha seja

reparada brevemente, depois de diagnosticada. Quanto maior o tempo de reparo, maior a probabilidade de acontecer uma segunda falha no componente redundante durante o reparo da primeira falha, o que comprometeria o sistema. Portanto, quanto maior o tempo de reparo, menor a disponibilidade do sistema;

8. Programar testes off-line periódicos em componentes para detectar falhas não diagnosticáveis automaticamente pelo sistema. O objetivo é detectar falhas ocultas, especialmente em componentes redundantes ou componentes simples que não estejam sendo solicitados (por exemplo, um relé de segurança). Testes off-line às vezes implicam em paradas no sistema, o que diminui a disponibilidade. Normalmente aproveitam-se ocasiões especiais, tais como paradas programadas de manutenção da planta. Quanto maior o período entre testes off-line, maior o tempo em que uma falha pode permanecer oculta, e portanto maior a probabilidade de uma falha comprometer o sistema, ou seja, menor a disponibilidade do sistema.

Estes princípios foram considerados no projeto de UTRs redundantes com UCP HX3040.

As próximas subseções analisam diversos tipos de falhas e como são toleradas ou não toleradas, e se existem switchovers associados às falhas toleradas.

### **Falhas Simples com Indisponibilidade**

Alguns componentes, por não serem duplicados, não toleram nem sequer falha simples sem causar algum tipo de indisponibilidade. Em uma UTR redundante com UCP HX3040, trata-se dos seguintes componentes:

- Módulos de E/S;
- Bastidores (HX9001 ou HX9003).

Quanto à indisponibilidade de um módulo de E/S, deve-se observar que ela não se constitui em uma indisponibilidade total do sistema. Ela se constitui em uma indisponibilidade parcial, somente das malhas de controle que utilizam este módulo de E/S.

Embora não haja previsão de redundância de módulos de E/S, a aplicação do usuário pode gerenciá-la em casos especiais. Por exemplo, o usuário pode inserir três módulos de entradas analógicas diferentes, e implementar um esquema de votação entre trios de entradas analógicas, para algum sistema crítico. No entanto, tais soluções, como foi enfatizado, devem ser gerenciadas pelo usuário. Não existe nenhum suporte automático para elas. Tais soluções, em geral, também implicam na redundância de transdutores e atuadores no campo.

### **Falhas Simples sem Indisponibilidade Causando um Switchover**

UCPs HX3040 redundantes toleram falhas simples sem causar indisponibilidade, mas causam switchover.

### **Falhas Simples sem Indisponibilidade**

Fontes de alimentação HX8300 e HX8320 redundantes toleram falhas simples, sem causar indisponibilidade e sem gerar switchover.

### **Overhead da Redundância**

Uma aplicação redundante causa um aumento no tempo de processamento de uma aplicação, quando comparado ao tempo necessário para uma aplicação equivalente não-redundante.

Este tempo adicional deve-se principalmente à execução dos serviços de sincronização cíclicos, descritos na seção Serviços de Sincronização Cíclicos através de Canais de Sincronismo de Redundância, além de um tempo menor para o próprio gerenciamento da redundância (máquina de estados etc.). O tempo adicional total, devido à redundância (overhead da redundância), é

estimado pelo MasterTool e exibido na janela de Mensagens, após compilar o projeto da UCP redundante.

O tempo para transferência dos dados varia conforme a quantidade de dados e segue uma proporção aproximada de 6,4 ms a cada 100 Kb de dados. Para o cálculo efetivo do tempo máximo de transferência, é estimado um tempo mínimo de 7 ms (constante) e calcula-se o restante usando a quantidade total de dados (incluindo a fila de eventos, de 117000 bytes). Sendo assim, tipicamente numa aplicação redundante vazia temos o tempo máximo de transferência calculado em  $7 \text{ ms} + 6,4 \text{ ms/Kbytes} * 117 \text{ Kbytes} = 15 \text{ ms}$ . Este tempo é mostrado pelo MasterTool como “overhead máximo de redundância”.

🔔 **NOTA:**

O overhead calculado pelo MasterTool considera uma lista de forçamentos de variáveis redundantes vazia.

O tempo de ciclo da MainTask deve ser ajustado levando em conta o tempo de “overhead máximo de redundância” calculado pelo MasterTool, mais 30 % de folga. Ou seja, no caso de 15 ms, o tempo de ciclo deverá ser de pelo menos 20 ms. Contudo, em aplicações redundantes, o tempo mínimo permitido é de 50 ms, devido ao tempo necessário para processamento da fila de eventos no switchover.

🔔 **NOTA:**

Caso a folga de 30% de tempo não seja respeitada, o sistema não garantirá um funcionamento de redundância adequado. O que normalmente ocorrerá é a troca de estado de um dos UCPs para INATIVO, devido às falhas de sincronização dos dados entre os UCPs.

Além disso, cabe ao usuário definir um intervalo para a MainTask que acomode:

- O tempo necessário para executar as POUs principais (NonSkippedPrg e UserPrg). Este tempo normalmente é medido após o desenvolvimento do projeto (descontando o tempo adicional para redundância);
- Alguma folga do ciclo de MainTask, para execução de outras tarefas da UCP (sistema operacional, I/O drivers etc.). O percentual desta folga pode variar de acordo com o desempenho solicitado destas outras tarefas. Por exemplo, se a comunicação MODBUS com o sistema SCADA precisa alocar muito processamento para atingir um desempenho satisfatório, esta folga deverá ser aumentada;
- O tempo necessário para executar as POUs de proteção (NonSkippedProtPrg e UserProtPrg), quando utilizadas. Como a prioridade da tarefa ProtTask, associada a estas POUs, é maior que a prioridade da MainTask, a frequência/intervalo da tarefa ProtTask deve ser levada em consideração. O tempo médio de consumo da ProtTask deverá ser de no máximo 20% do seu tempo de ciclo. Isso equivale a 800 us se ela estiver trabalhando com 4ms de interval. Caso isso não seja respeitado, o sistema não conseguirá executar a transferência dos dados redundantes. A ProtTask também não deverá ter picos muito maiores que os 20% do tempo de ciclo, pois isso pode inclusive gerar um switchover. Assim, recomenda-se não ter lógicas com laços que possam divergir e consumir muito tempo.

🔔 **NOTA:**

Dependendo do alinhamento de memória, o número de bytes utilizados no cálculo do overhead de redundância poderá ser maior do que o total de bytes declarados nas variáveis.

# Programação de uma UCP Redundante



## Wizard para Criação de um Novo Projeto Redundante

Para criar um projeto redundante, deve-se utilizar o comando Arquivo / Novo Projeto, e selecionar o Projeto MasterTool Padrão.

Inicialmente, o usuário deve informar o nome que deseja dar ao projeto e o diretório onde deseja armazená-lo conforme mostrado na figura a seguir.

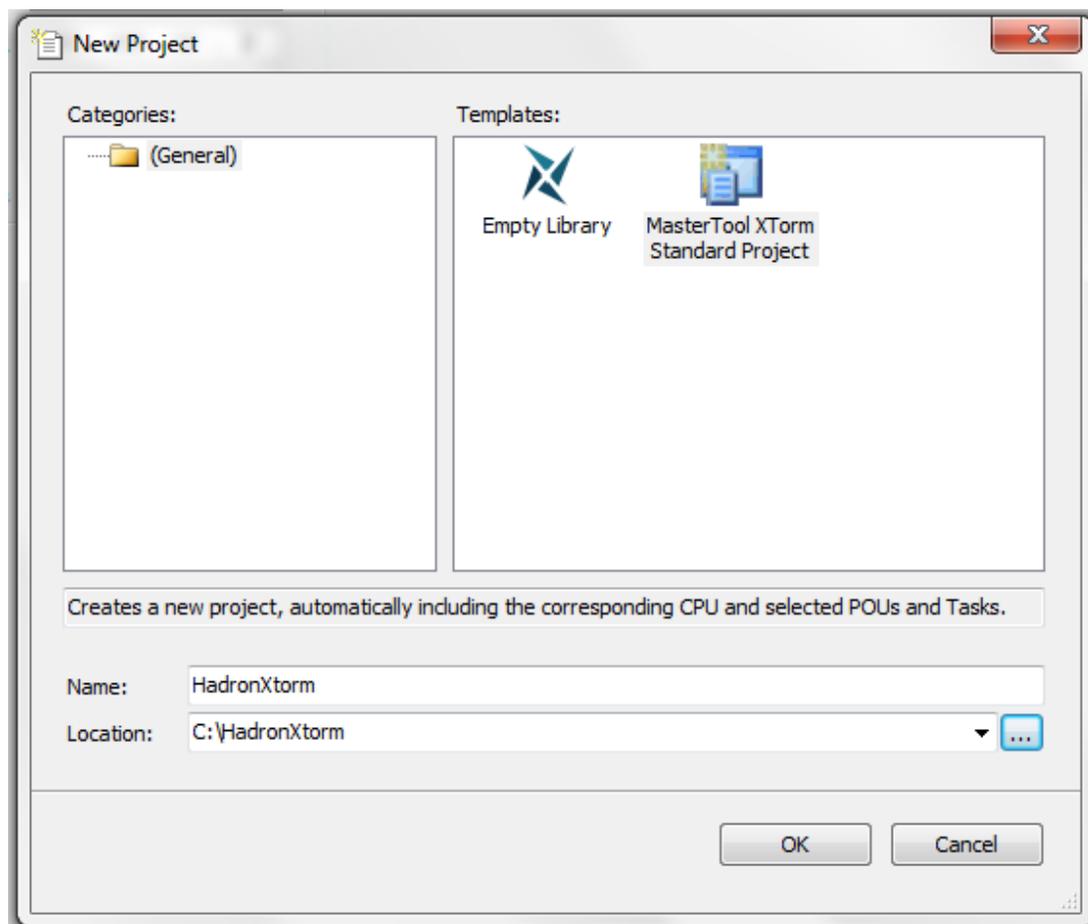


Figura 4-5. Novo Projeto

A seguir, o Wizard que gera o projeto de redundância realiza alguns questionamentos ao usuário quanto à configuração desejada. O primeiro ponto a ser definido é a configuração inicial de hardware da UTR. Os seguintes itens devem ser configurados:

1. Selecionar o modelo da UCP - Como a redundância está implementada somente na HX3040, a mesma deve ser selecionada pelo usuário;
2. Selecionar o modelo do bastidor - Existem duas opções de bastidores disponíveis e a escolha dele dependerá da quantidade de módulos utilizados;
3. Selecionar o modelo da fonte de alimentação;
4. Selecionar a configuração da redundância de UCP;
5. Selecionar a configuração da redundância de fonte de alimentação (essa tipo de configuração para a fonte de alimentação independe da configuração de redundância de UCP).

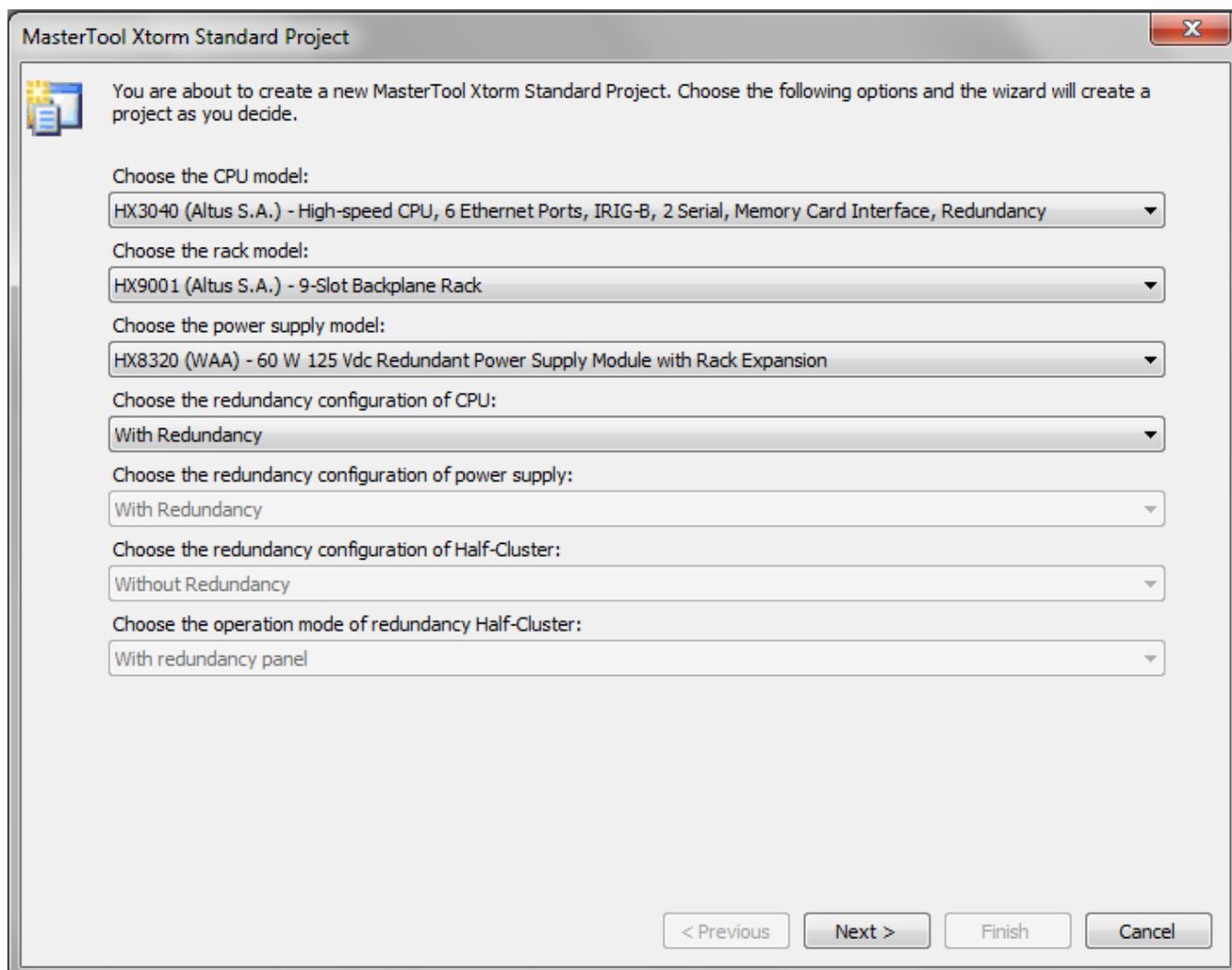


Figura 4-6. Configuração Inicial de Hardware

Após, o usuário deve definir a quantidade e os tipos de E/S utilizadas na aplicação conforme sequência indicada a seguir:

1. Selecionar o número de pontos de entradas digitais;
2. Selecionar o número de pontos de saídas digitais;
3. Selecionar o número de pontos de entradas analógicas V/I;
4. Selecionar o número de pontos de entradas analógicas RTD.

MasterTool Xtorm Standard Project

Choose the quantity of I/O points that will be used and the wizard will create the following objects within the project.

Quantity of digital input points: 32

Quantity of digital output points: 16

Quantity of V/I analog input points: 16

Quantity of RTD analog input points: 8

Expansion cable type:  
Not Connected

Quantity of I/O modules that will be used: 4 of 64.  
Quantity of racks that will be used: 1 of 9.

< Previous   Next >   Finish   Cancel

Figura 4-7. Configuração dos pontos de E/S

Então, deve ser selecionado o perfil de projeto e a linguagem padrão para a criação dos programas conforme itens indicados a seguir:

1. Selecionar o perfil do projeto - Somente é possível utilizar o perfil de projeto “Perfil para UTR” para a redundância, logo a opção de seleção está bloqueada para edição;
2. Selecionar a linguagem padrão para todos os programas - A linguagem selecionada pelo usuário será a padrão para todas os programas, porém pode optar por utilizar qualquer outra para uma POU específica.

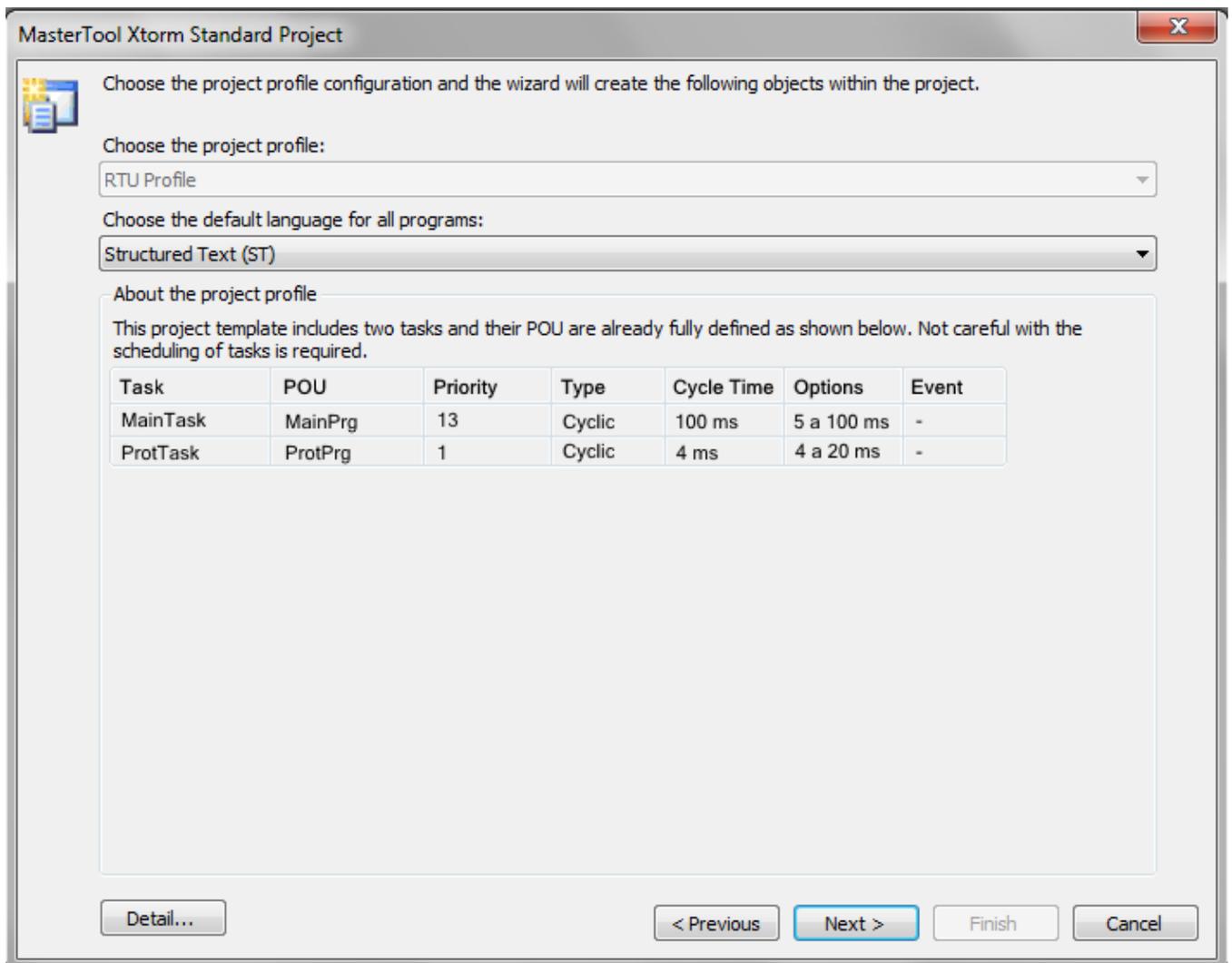


Figura 4-8. Perfil de Projeto e Linguagem Padrão

Para finalizar, o usuário deve selecionar a linguagem dos programas comuns e daqueles associados à redundância:

1. Programas associados à MainTask (MainPrg);
2. Programas associados a tarefas cíclicas: deverá ser, obrigatoriamente, em linguagem ST, sendo que o MasterTool desabilita as outras opções;
3. Programas associados às tarefas principais da redundância.

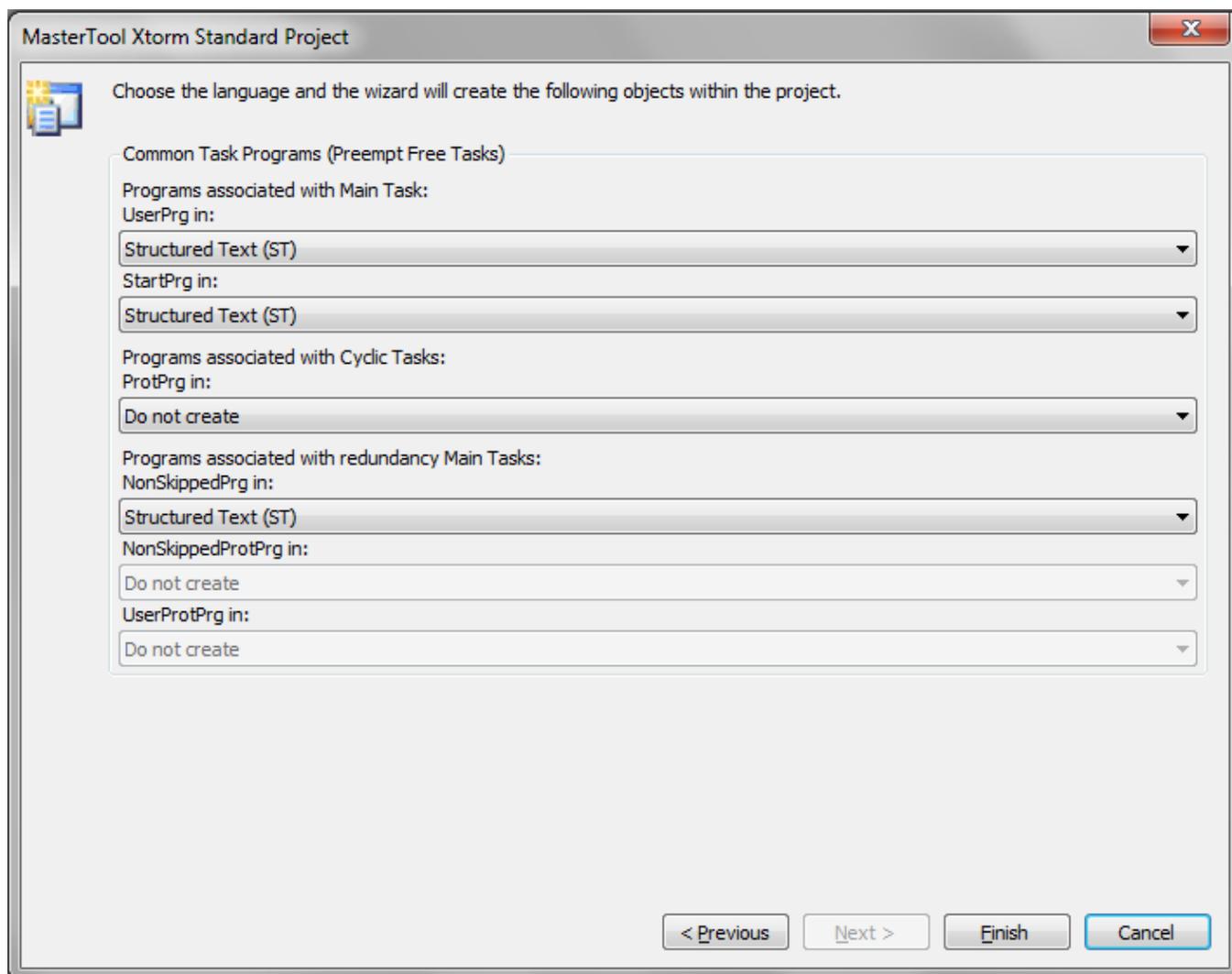


Figura 4-9. Linguagem dos Programas Específicos

#### NOTAS:

1. As POU's UserPrg e NonSkippedPrg são criadas automaticamente, vazias, na linguagem selecionada nas perguntas anteriores. Em POU's criadas manualmente pelo usuário, poderá ser utilizada qualquer uma das linguagens disponíveis, exceto em POU's redundantes, que não podem ser escritas na linguagem SFC, pois esta utiliza o timer IEC em segundo plano.
2. A POU MainPrg será sempre gerada automaticamente em linguagem ST, e não pode ser alterada pelo usuário. Esta POU chama as POU's UserPrg (somente na UCP Ativa) e NonSkippedPrg (em ambas UCP's). De forma equivalente, a POU ProtPrg, quando requisitada pelo usuário para ser criada, é gerada automaticamente em linguagem ST, e não pode ser alterada pelo usuário. Esta POU chama as POU's UserProtPrg (somente na UCP Ativa) e NonSkippedProtPrg (em ambas UCP's).

Depois de obter respostas para as perguntas anteriores, o Wizard gera o projeto inicial, definindo um sistema com a seguinte configuração básica de hardware:

- Bastidor selecionado;
- Fonte de alimentação (posições 0 e 1);
- UCP HX3040 (posições 2 e 3);

- Módulos de E/S (demais posições disponíveis no bastidor).

### Configuração do projeto com redundância de UCP

O Wizard sempre é utilizado para gerar a primeira versão de um projeto redundante. Isso garante que a versão inicial do projeto será gerada rápida e corretamente.

No entanto, é possível que algumas modificações sejam necessárias, tal como a inserção de novos módulos de E/S. As seções a seguir mostram como adicionar e configurar os módulos.

Algumas regras e precauções devem ser seguidas para um projeto redundante, conforme descreve a seção seguinte.

### Configuração Fixa nas Posições 0 a 3 do Bastidor

Nas posições 0 a 3 do bastidor selecionado sempre devem estar instalados os seguintes módulos:

- Fonte de alimentação redundante (posição 0 e posição 1);
- UCP HX3040 redundante (posição 2 e posição 3);
- Módulos de E/S (posição 4 em diante).

Estes módulos não devem ser removidos do projeto original gerado pelo Wizard.

Qualquer configuração diferente nestas posições resultará em um erro notificado pelo MasterTool na compilação do projeto.

### Configurações das Portas Ethernet da UCP HX3040 (NET 1 até NET 6)

#### Configuração do Endereço de IP

A figura a seguir mostra as configurações da porta NET 1 da UCP HX3040. Para abrir esta tela, deve-se dar um duplo clique sobre a NET desejada, abaixo da UCP HX3040 na árvore de dispositivos.

Ethernet Port Parameters	
Cluster IP Addressing	
IP Address Active	192 . 168 . 15 . 1
IP Address CPU A	192 . 168 . 15 . 2
IP Address CPU B	192 . 168 . 15 . 3
Subnetwork Mask	255 . 255 . 255 . 0
Gateway Address	192 . 168 . 15 . 253
Advanced...	

Figura 4-10. Parâmetros da Porta Ethernet NET 1

A seguir devem ser editados os parâmetros básicos para a interface desejada. O endereçamento será de acordo com o método de troca IP Ativo, conforme descrito nesse Tutorial.

#### NOTAS:

1. Os três endereços IPs (Ativo, da UCPA e da UCPB), bem como o endereço do Gateway, devem pertencer à mesma subrede.
2. As NETs ímpares (1, 3 e 5) possuem o botão Avançado, enquanto as NETs pares (2, 4 e 6) não o possuem. Através deste botão é possível configurar redundância de NETs (NIC Teaming) e selecionar o modo Switch.

### Configurações de I/O Drivers

Por definição, os I/O Drivers são todos os drivers de comunicação utilizados pela UCP, como Modbus Cliente, Servidor, Mestre, Escravo, drivers DNP3 Cliente, Outstation, IEC 61850 Servidor etc. Estes drivers funcionam de forma diferente em UCPs com redundância.

Para que um I/O Driver esteja rodando, a UCP deve estar em modo RUN e em estado de redundância Ativo. Isto quer dizer que uma UCP Inicializando, Inativa ou em estado Reserva ainda não terá seus I/O Drivers rodando, ou seja, clientes e servidores (MODBUS, DNP3 etc.) estarão parados.

Desta maneira, não é possível utilizar a comunicação de I/O Drivers numa UCP em estado Não-Ativo.

### Configurações da MainTask

A tela de configurações associadas à tarefa principal de uma UCP redundante, denominada MainTask, a qual é cíclica, pode ser acessada clicando sobre MainTask na árvore de dispositivos.

Dois parâmetros devem ser ajustados nesta tela:

1. Intervalo da MainTask;
2. Tempo de cão-de-guarda.

Diversas considerações devem ser feitas para ajustar adequadamente o intervalo da MainTask:

- O intervalo deve ser suficientemente baixo para controlar o processo efetivamente, observando os tempos de resposta de todos os laços de controle;
- O intervalo deve ser suficientemente alto para acomodar, no mínimo, a soma dos dois seguintes tempos:
  - O tempo máximo de execução das POUs NonSkippedPrg e UserPrg, em conjunto;
  - O tempo necessário para gerenciar a redundância (overhead da redundância);
- O intervalo deve ter uma folga adicional, necessária para que outros processos tenham tempo de ser executados (comunicação Ethernet com sistemas SCADA etc.).

O MasterTool tem condições de calcular o tempo necessário para gerenciar a redundância (overhead de redundância), depois que o projeto estiver terminado (todas POUs desenvolvidas, e áreas de memória redundantes definidas).

Quanto ao tempo máximo de execução das POUs NonSkippedPrg e UserPrg, é possível medir os tempos depois que estas POUs forem desenvolvidas. Após cada compilação do projeto, o MasterTool soma o overhead de redundância calculado com o parâmetro que informa os tempos das POUs (NonSkippedPrg e UserPrg), e verifica se a folga mínima parametrizada está sendo obedecida. Exemplo:

Parâmetros configurados na tela da MainTask:

- Intervalo da MainTask: 100 ms;
- Tempo estimado para POUs NonSkippedPrg + UserPrg: 10 ms;
- Folga Mínima: 30%;

- Overhead calculado para redundância: 50 ms.

Neste caso, o tempo total utilizado é de 60 ms (10 ms + 50 ms), o que consiste em 60% do ciclo da MainTask (100 ms). Desta forma, a folga é de 40%, e portanto a folga mínima de 30% está sendo respeitada.

### Programa StartPrg

Nesta POU o usuário pode criar lógicas, laços, iniciar variáveis etc. que serão executados somente uma única vez no primeiro ciclo de cada uma das UTRs, não sendo chamado novamente durante a execução do projeto.

Caso o usuário carregue uma nova aplicação, ou se o PLC for desenergizado, bem como em condições de Reset Origin, Reset Cold e Reset Warm, esta POU será executada novamente.

### Programa UserPrg

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

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

Deve-se lembrar que todas as variáveis simbólicas definidas na POU UserPrg, assim como instâncias de blocos funcionais ali definidas, serão variáveis redundantes.

Variáveis simbólicas definidas em POU's adicionais do tipo programa, mesmo que chamadas dentro de UserPrg, só serão variáveis redundantes se o usuário desejar, marcando-a como redundante na configuração do objeto Redundancy Configuration, na árvore de dispositivos do projeto. Por padrão, todas as POU's criadas pelo usuário serão, inicialmente, redundantes.

#### 🔔 **NOTA:**

Variáveis do tipo VAR\_TEMP não devem ser utilizadas no programa redundante.

### Programa NonSkippedPrg

Esta POU é destinada para controles que devem ser executados em ambos UCPs (UCPA e UCPB), independente do seu estado de redundância. Esta POU é chamada pela POU principal (MainPrg).

Deve-se lembrar que todas as variáveis simbólicas definidas na POU NonSkippedPrg, assim como instâncias de blocos funcionais ali definidas, serão variáveis não-redundantes.

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

#### 🔔 **NOTAS:**

1. Ao chamar POU's adicionais do tipo programa dentro de NonSkippedPrg, desmarque as mesmas na janela Redundancy Configuration do MasterTool. Por padrão as variáveis simbólicas declaradas dentro destas POU's serão redundantes e, dentro da NonSkippedPrg, normalmente desejam-se variáveis não redundantes. Normalmente o código de NonSkippedPrg é pequeno, dispensando a chamada de POU's adicionais do tipo programa para sua estruturação, mas, caso a sua estruturação seja necessária, o mais recomendável é utilizar bloco funcional ou funções.
2. Não se recomenda utilizar os blocos funcionais TOF\_RET, TON\_RET, TOF e TON no programa NonSkippedPrg.

## GVL Disables

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

Declaração de variável de desabilitação de requisição

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

Onde:

- **Nome do dispositivo** - Nome que aparece na Treeview 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:

Disables

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

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

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

Enviando um projeto para a UCP e colocando-a em "Execução", através da GVL "Disables" é possível monitorar e alterar os valores das variáveis de desabilitação. Se o valor da variável for TRUE significa que a requisição, à qual a variável pertence, está desabilitada e o inverso é válido para quando o valor da variável for FALSE.

Expressão	Tipo	Valor	Valor Preparado	Comentário
 MODBUS_Device_DISABLE_0001	BOOL	TRUE		
 MODBUS_Device_DISABLE_0002	BOOL	FALSE		
 MODBUS_Device_DISABLE_0003	BOOL	TRUE		
 MODBUS_Device_1_DISABLE_0001	BOOL	TRUE		
 MODBUS_Device_1_DISABLE_0002	BOOL	FALSE		

Figura 4-11. GVL Disable em Modo Online

## GVL IOQualities

Na GVL "IOQualities" são declaradas as variáveis de qualidade dos módulos de E/S. 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:

```
IOQualities
```

```
VAR_GLOBAL
QUALITY_HX1120: ARRAY[0..31] OF LibDataTypes.QUALITY;
QUALITY_HX2320: ARRAY[0..15] OF LibDataTypes.QUALITY;
QUALITY_HX6000: ARRAY[0..15] OF LibDataTypes.QUALITY;
QUALITY_HX6020: ARRAY[0..7] OF LibDataTypes.QUALITY;
END_VAR
```

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

### **GVL Module\_Diagnostics**

Na GVL "Module\_Diagnostics" são declaradas as variáveis de diagnóstico dos módulos 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. Exemplo:

#### **Module\_Diagnostics**

```
VAR_GLOBAL
DG_HX1120 : T_DIAG_HX1120_1;
DG_HX2320 : T_DIAG_HX2320_1;
DG_HX6000 : T_DIAG_HX6000_1;
DG_HX6020 : T_DIAG_HX6020_1;
DG_HX8320 : T_DIAG_HX8320_1;
END_VAR
```

Enviando uma aplicação para a UCP HX3040 e passando-a para "Execução", através da GVL "Module\_Diagnostics" é possível monitorar os diagnósticos de cada módulo que foi adicionado ao projeto, como é possível visualizar na figura a seguir.

Expression	Type	Value
[-] 🌐 DG_HX1120	T_DIAG_HX1120_1	
[-] 🌐 tGeneral	T_DIAG_GENERAL_HX1120_1	
🌐 bReserved_08	BOOL	FALSE
🌐 bReserved_09	BOOL	FALSE
🌐 bReserved_10	BOOL	FALSE
🌐 bReserved_11	BOOL	FALSE
🌐 bReserved_12	BOOL	FALSE
🌐 bReserved_13	BOOL	FALSE
🌐 bReserved_14	BOOL	FALSE
🌐 bReserved_15	BOOL	FALSE
🌐 bActiveDiagnostics	BOOL	FALSE
🌐 bFatalError	BOOL	FALSE
🌐 bConfigMismatch	BOOL	FALSE
🌐 bWatchdogError	BOOL	FALSE
🌐 bOTDSwitchError	BOOL	FALSE
🌐 bReserved_05	BOOL	FALSE
🌐 bReserved_06	BOOL	FALSE
🌐 bCommunicationError	BOOL	FALSE
[+] 🌐 tDetailed	T_DIAG_DETAILED_HX1120_1	
[+] 🌐 DG_HX2320	T_DIAG_HX2320_1	
[+] 🌐 DG_HX6000	T_DIAG_HX6000_1	
[+] 🌐 DG_HX6020	T_DIAG_HX6020_1	
[+] 🌐 tGeneral	T_DIAG_GENERAL_HX6020_1	
[+] 🌐 tSpecific	T_DIAG_SPECIFIC_HX6020_1	
🌐 bReserved_08	BOOL	FALSE
🌐 bReserved_09	BOOL	FALSE
🌐 bReserved_10	BOOL	FALSE
🌐 bReserved_11	BOOL	FALSE
🌐 bReserved_12	BOOL	FALSE
🌐 bReserved_13	BOOL	FALSE
🌐 bReserved_14	BOOL	FALSE
🌐 bReserved_15	BOOL	FALSE
🌐 bActiveDiagnosticsInput00	BOOL	FALSE
🌐 bActiveDiagnosticsInput01	BOOL	FALSE

Figura 4-12. GVL Module\_Diagnostics em Modo Online

### GVL Qualities

Na GVL "Qualities" são declaradas as variáveis de qualidade dos mapeamentos de variáveis internas, MODBUS Mestre/Cliente e DNP3 Cliente. 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 para a declaração de variável de qualidade de mapeamento:

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

Onde:

- **Nome do dispositivo** - Nome que aparece na Treeview para o dispositivo;

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

Exemplo:

```
Qualities
VAR_GLOBAL
MODBUS_Device_QUALITY_0001: LibDataTypes.QUALITY;
MODBUS_Device_QUALITY_0002: LibDataTypes.QUALITY;
MODBUS_Device_QUALITY_0003: LibDataTypes.QUALITY;
Outstation_QUALITY_0001: LibDataTypes.QUALITY;
END_VAR
```

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

### 🔔 **NOTA:**

Se uma variável dos drivers DNP3 Cliente ou MODBUS Mestre/Cliente de mapeamento simbólico for mapeada nos drivers DNP3 Servidor ou IEC 60870-5-104 Servidor, é necessário que as variáveis de qualidade dos mapeamentos DNP3 ou MODBUS tenham sido criadas para que sejam gerados eventos de qualidade válidos para tais pontos dos servidores DNP3 ou IEC 60870-5-104. Caso contrário, não serão gerados eventos de qualidade "ruim" para os clientes dos servidores DNP3 e IEC 60870-5-104 nas situações que o DNP3 Cliente ou MODBUS Mestre/Cliente não consiga comunicar com os seus escravos/servidores, por exemplo.

Enviando uma aplicação para a UCP HX3040 e passando-a para "Execução", através da GVL "Qualities", é possível monitorar os diagnósticos os valores das variáveis de diagnósticos de mapeamentos dos dispositivos MODBUS Mestre/Cliente e DNP3 Cliente, conforme mostra a figura a seguir.

Expression	Type	Value
MODBUS_Device_QUALITY_0001	Rtu_Standard.QUALITY	
VALIDITY	QUALITY_VALIDITY	VALIDITY_INVALID
FLAGS	QUALITY_FLAGS	
FLAG_OUT_OF_RANGE	BOOL	FALSE
FLAG_INACCURATE	BOOL	FALSE
FLAG_OLD_DATA	BOOL	FALSE
FLAG_FAILURE	BOOL	FALSE
FLAG_OPERATOR_BLOCKED	BOOL	FALSE
FLAG_TEST	BOOL	FALSE
FLAG_RESERVED_0	BOOL	FALSE
FLAG_RESERVED_1	BOOL	FALSE
FLAG_RESTART	BOOL	TRUE
FLAG_COMM_FAIL	BOOL	FALSE
FLAG_REMOTE_SUBSTITUTED	BOOL	FALSE
FLAG_LOCAL_SUBSTITUTED	BOOL	FALSE
FLAG_FILTER	BOOL	FALSE
FLAG_OVERFLOW	BOOL	FALSE
FLAG_REFERENCE_ERROR	BOOL	FALSE
FLAG_INCONSISTENT	BOOL	FALSE
MODBUS_Device_QUALITY_0002	Rtu_Standard.QUALITY	
VALIDITY	QUALITY_VALIDITY	VALIDITY_INVALID
FLAGS	QUALITY_FLAGS	
MODBUS_Device_QUALITY_0003	Rtu_Standard.QUALITY	
VALIDITY	QUALITY_VALIDITY	VALIDITY_INVALID
FLAGS	QUALITY_FLAGS	
Outstation_QUALITY_0001	Rtu_Standard.QUALITY	
VALIDITY	QUALITY_VALIDITY	VALIDITY_QUESTIONABLE
FLAGS	QUALITY_FLAGS	

Figura 4-13. GVL Qualities em Modo Online

### GVL ReqDiagnostics

Na GVL "ReqDiagnostics" são declaradas as variáveis de diagnóstico das requisições MODBUS Mestre/Cliente e DNP3 Cliente. Não é obrigatória, mas é recomendada a utilização da geração automática destas variáveis, o que é feito clicando no botão "Gerar Variáveis de Diagnóstico" na aba de requisições do dispositivo. A declaração destas variáveis segue a seguinte estrutura para a declaração de variável de diagnóstico de requisição:

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

Onde:

- **Nome do dispositivo** - Nome que aparece na Treeview 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** - "T\_DIAG\_MODBUS\_RTU\_MAPPING\_1" para MODBUS Master, "T\_DIAG\_MODBUS\_ETH\_MAPPING\_1" para MODBUS Cliente e "T\_DIAG\_DNP\_CLIENT\_REQUEST\_1" para DNP3 Cliente.

Exemplo:

```
ReqDiagnostics
VAR_GLOBAL
```

```
MODBUS_Device_REQDG_0001 : T_DIAG_MODBUS_RTU_MAPPING_1;  
MODBUS_Device_REQDG_0002 : T_DIAG_MODBUS_RTU_MAPPING_1;  
MODBUS_Device_REQDG_0003 : T_DIAG_MODBUS_RTU_MAPPING_1;  
MODBUS_Device_1_REQDG_0001 : T_DIAG_MODBUS_ETH_MAPPING_1;  
MODBUS_Device_1_REQDG_0002 : T_DIAG_MODBUS_ETH_MAPPING_1;  
Outstation_REQDG_0001 : T_DIAG_DNP_CLIENT_REQUEST_1;  
END_VAR
```

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

Enviando uma aplicação para a UCP HX3040 e passando-a para "Execução", através da GVL "ReqDiagnostics" é possível monitorar os diagnósticos os valores das variáveis de diagnósticos de mapeamentos dos dispositivos MODBUS Mestre/Cliente e DNP3 Cliente, conforme mostra a figura a seguir.

Expression	Type	Value
MODBUS_Device_REQDG_0001	T_DIAG_MODBUS_RTU_MAPPING_1	
byStatus	T_DIAG_MODBUS_RTU_MAPPING_STATUS	
bCommIdle	BOOL	FALSE
bCommExecuting	BOOL	FALSE
bCommPostponed	BOOL	FALSE
bCommDisabled	BOOL	FALSE
bCommOk	BOOL	FALSE
bCommError	BOOL	FALSE
bDiag_6_reserved	BOOL	FALSE
bDiag_7_reserved	BOOL	FALSE
eLastErrorCode	MASTER_ERROR_CODE	NO_ERROR
eLastExceptionCode	MODBUS_EXCEPTION	NO_EXCEPTION
byDiag_3_reserved	BYTE	0
wCommCounter	WORD	0
wCommErrorCounter	WORD	0
MODBUS_Device_REQDG_0002	T_DIAG_MODBUS_RTU_MAPPING_1	
MODBUS_Device_REQDG_0003	T_DIAG_MODBUS_RTU_MAPPING_1	
MODBUS_Device_1_REQDG_0001	T_DIAG_MODBUS_ETH_MAPPING_1	
MODBUS_Device_1_REQDG_0002	T_DIAG_MODBUS_ETH_MAPPING_1	
Outstation_REQDG_0001	T_DIAG_DNP_CLIENT_REQUEST_1	
eConnectionStatus	CONNECTION_STATUS	CONNECTING
eRequestStatus	REQUEST_STATUS	SUCCESS
tIIN	T_DIAG_DNP_CLIENT_IIN_BITS	
ALL_STATIONS	BOOL	FALSE
CLASS_1_EVENTS	BOOL	FALSE
CLASS_2_EVENTS	BOOL	FALSE
CLASS_3_EVENTS	BOOL	FALSE
NEED_TIME	BOOL	FALSE
LOCAL_CONTROL	BOOL	FALSE
DEVICE_TROUBLE	BOOL	FALSE
DEVICE_RESTART	BOOL	FALSE
NO_FUNC_CODE_SUPPORT	BOOL	FALSE
OBJECT_UNKNOWN	BOOL	FALSE
PARAMETER_ERROR	BOOL	FALSE

Figura 4-14. GVL ReqDiagnostics em Modo Online

### GVL System\_Diagnostics

Na GVL "System\_Diagnostics" são declaradas as variáveis de diagnóstico da UCP e dos 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. Exemplo:

```
VAR_GLOBAL
DG_HX3040 : T_DIAG_HX3040_1;
DG_DNP3_Client : T_DIAG_DNP_CLIENT_1;
DG_MODBUS_Symbol_Client : T_DIAG_MODBUS_ETH_CLIENT_1;
DG_MODBUS_Symbol_RTU_Master : T_DIAG_MODBUS_RTU_MASTER_1;
END_VAR
```

Enviando uma aplicação para a UCP e passando-a para "Execução", através da GVL "System\_Diagnostics" é possível monitorar os valores das variáveis de diagnósticos da UCP HX3040 e dos dispositivos de comunicação DNP3 e MODBUS, conforme mostra a figura a seguir.

Expression	Type	Value
[-] DG_HX3040	T_DIAG_HX3040_1	
[+] tSummarized	T_DIAG_SUMMARIZED_1	
[+] tDetailed	T_DIAG_DETAILED_1	
[+] DG_DNP3_Client	T_DIAG_DNP_CLIENT_1	
[-] DG_MODBUS_Symbol_Client	T_DIAG_MODBUS_ETH_CLIENT_1	
[+] tDiag	T_DIAG_MODBUS_DIAGNOSTICS	
byDiag_1_reserved	BYTE	0
[-] tCommand	T_DIAG_MODBUS_COMMANDS	
bStop	BOOL	FALSE
bRestart	BOOL	FALSE
bResetCounter	BOOL	FALSE
bDiag_19_reserved	BOOL	FALSE
bDiag_20_reserved	BOOL	FALSE
bDiag_21_reserved	BOOL	FALSE
bDiag_22_reserved	BOOL	FALSE
bDiag_23_reserved	BOOL	FALSE
byDiag_3_reserved	BYTE	0
[+] tStat	T_DIAG_MODBUS_ETH_CLIENT_STATS	
[-] DG_MODBUS_Symbol_RTU_Master	T_DIAG_MODBUS_RTU_MASTER_1	
[+] tDiag	T_DIAG_MODBUS_DIAGNOSTICS	
eErrorCode	SERIAL_STATUS	NO_ERROR
[+] tCommand	T_DIAG_MODBUS_COMMANDS	
byDiag_03_reserved	BYTE	0
[-] tStat	T_DIAG_MODBUS_RTU_MASTER_STATS	
wTXRequets	WORD	0
wRXNormalResponses	WORD	0
wRXExceptionResponses	WORD	0
wRXIllegalResponses	WORD	0
wRXOverrunErrors	WORD	0
wRXIncompleteFrames	WORD	0
wCTSTimeOutErrors	WORD	0
wDiag_18_reserved	WORD	0

Figura 4-15. GVL System\_Diagnostics em Modo Online

## Configurações da ProtTask

A "ProtTask" é responsável por executar uma única unidade de programação do tipo "Program", cujo nome é "ProtPrg". Este, por sua vez, pode chamar outras unidades de programação e é responsável por executar as rotinas de proteção do projeto. A "ProtTask" é uma tarefa do tipo cíclica e terá sua prioridade fixada como 1.

## Programa UserProtPrg

Esta POU é executada somente na UCP Ativa e destina-se ao controle do processo do usuário. A "UserProtPrg" é criada somente em projetos com redundância de UCP e possui uma prioridade maior em relação a "UserPrg".

## Programa NonSkippedProtPrg

Esta POU é semelhante a "NonSkippedPrg", com a diferença de possuir uma prioridade maior. A "NonSkippedProtPrg" é criada somente em projetos com redundância de UCP.

## **GVLs com Variáveis Simbólicas Redundantes**

O usuário pode criar outras GVLs, diferentes das citadas anteriormente, para declarar variáveis simbólicas redundantes. Para isso, após a criação da GVL, deve-se marcá-la na configuração do objeto Redundancy Configuration, na árvore de dispositivos do projeto. Por padrão, todas as GVLs criadas pelo usuário serão, inicialmente, redundantes. É recomendado evitar a utilização da diretiva AT em GVLs que contenham declarações de variáveis simbólicas redundantes, para evitar o mapeamentos de variáveis em áreas não redundantes.

## **POUs do Tipo Programa com Variáveis Simbólicas Redundantes**

O usuário pode declarar variáveis simbólicas redundantes em POU's do tipo programa, com exceção da POU NonSkippedPrg e da NonSkippedProtPrg onde as variáveis simbólicas declaradas são consideradas não-redundantes, mas pode criar outras POU's, diferentes das citadas anteriormente, para declarar variáveis simbólicas redundantes. Para isso, após a criação da POU, deve marcá-la na configuração do objeto Redundancy Configuration, na árvore de dispositivos do projeto. Por padrão, todas as POU's criadas pelo usuário serão, inicialmente, redundantes. É recomendado evitar a utilização da diretiva AT em POU's que sejam redundantes, a fim de evitar o mapeamentos de variáveis em áreas não redundantes.

## **Utilização de Breakpoints em Sistemas Redundantes**

Para sistemas redundantes, só é possível utilizar breakpoints na UCP Ativa, com a outra UCP fora do barramento ou em estado Inativo. O MasterTool bloqueará a operação caso ela não respeite esta condição.

## **Limitações na Programação de uma UCP Redundante**

Em uma UCP redundante, irão existir algumas limitações quanto à programação. Estas limitações são tratadas nas subseções a seguir.

### **Limitações em GVLs e POU's Redundantes**

Em uma GVL ou em uma POU do tipo programa que sejam redundantes, as seguintes limitações devem ser respeitadas para um correto funcionamento dos UCPs:

- Não utilizar variáveis do tipo VAR\_TEMP;
- Não misturar tipos de variáveis (VAR, VAR RETAIN, VAR PERSISTENT etc.), devendo ser utilizado somente um dos tipos em cada GVL ou POU;
- Não misturar declaração de variáveis simbólicas com ATs nas GVLs. Devem ser criadas GVLs separadas, declarando em uma as variáveis do tipo AT e em outra as variáveis simbólicas;
- Não armazenar o endereço de uma variável em uma variável redundante (fazer de uma variável redundante um ponteiro para um endereço), pois os endereços das variáveis podem ser diferentes na UCPA e na UCPB.

### **Limitações no Programa Não-Redundante (StartPrg, NonSkippedPrg e NonSkippedProtPrg)**

Em uma POU do tipo programa que não seja redundante, no caso, as POU's StartPrg, NonSkippedPrg e NonSkippedProtPrg, as seguintes limitações devem ser respeitadas para um correto funcionamento dos UCPs:

- Não podem ser utilizados os blocos funcionais TON e TOF tradicionais, pois eles utilizam o timer IEC. Quando a UCP Reserva entrar em estado ativo (com a outra UCP saindo do

estado Ativo), o timer IEC será sincronizado, causando uma descontinuidade no valor do timer. Deve-se optar pelos blocos funcionais TON\_NR e TOF\_NR disponibilizados na biblioteca LibPlcStandard;

- Não podem ser utilizadas POU's do tipo programa escritas na linguagem SFC (Sequenciamento Gráfico de Funções), pois estas utilizam o timer IEC para temporizar as transições;
- Não misturar declaração de variáveis simbólicas com ATs nas GVLs. Devem ser criadas GVLs separadas, declarando em uma as variáveis do tipo AT e em outra as variáveis simbólicas.

### Obtendo o Estado da Redundância de uma UCP

É possível verificar o estado da redundância de uma UCP a partir da Estrutura de Diagnósticos da Redundância:

```
VAR
eRedStateLocal : REDUNDANCY_STATE;
END_VAR
eRedStateLocal :=
DG_HX3040_01.RedDgnLoc.RedundancyDiagnostics.eRedState;
```

Deste modo, o usuário poderá fazer o controle de lógicas que dependam do estado da redundância da UCP, o valor recebido será um valor numérico dado em byte, os significados de todos os estados são: Não-Configurado (0), Inicializando (2), Reserva (3), Ativo (4) e Inativo (5).

### Leitura de Diagnósticos Não-Redundantes

Um projeto redundante, além de apresentar diagnósticos redundantes (Estrutura de Diagnósticos da Redundância e diagnósticos dos módulos do barramento), apresenta também diagnósticos que não são redundantes (diagnósticos específicos da UCP HX3040). Estes diagnósticos não são trocados entre as UCPs. Por exemplo, se no programa o usuário quiser saber o estado do link das portas ethernet da outra UCP, ele deverá fazê-lo utilizando bytes disponíveis na estrutura de dados de usuário dos diagnósticos da redundância.

## Carga de Programas em uma UCP Redundante



Nesta seção, discutem-se métodos e etapas para carregar este projeto em uma UCP redundante, considerando diversas situações, tais como:

- Carga do projeto em uma UCP HX3040 nova, retirada da caixa, ou em uma UCP contendo um projeto desconhecido;
- Carga on-line de modificações sem interrupção do controle do processo;
- Carga off-line de modificações com interrupção do controle do processo, durante uma parada programada do processo.

### Carga Inicial de um Projeto Redundante

Esta seção descreve os passos necessários para fazer a primeira carga de projeto redundante em uma UCP HX3040. Isto é necessário, por exemplo, para uma UCP nova de fábrica, ou para uma UCP que contenha um projeto desconhecido.

Os passos seguintes só precisam ser executados para uma das duas UCPs (UCPA ou UCPB) que compõem o sistema redundante. Ao se introduzir no barramento a segunda UCP, ela - automaticamente - irá se sincronizar com a UCP já em funcionamento, caso ela esteja em estado Ativo.

### **Primeiro Passo - Descoberta do Endereço IP para Conexão do MasterTool**

O primeiro passo é descobrir o endereço IP do canal NET 1 desta UCP, para conexão ao MasterTool. Isto deve ser feito através do visor e botão da UCP HX3040, conforme descrito anteriormente nesse Tutorial. O menu REDE informa os endereços IPs de todas as portas NET da UCP HX3040, especialmente a da NET1 por ser a única utilizada para comunicação com o MasterTool.

### **Segundo Passo – Verificar Rede e IP do Computador para Programação**

Para que todos os próximos estágios possam ser seguidos, existe a necessidade de que o computador que irá programar e enviar o projeto para a UCP HX3040 esteja na mesma subrede ethernet e que possua um IP diferente e único do que será configurado para a UCP.

### **Terceiro Passo – Verificar Conflito de Endereços IP**

Antes de executar o próximo passo, deve-se ter certeza de que não existe, na rede, outro equipamento com o mesmo endereço IP descoberto nos passos anteriores. O usuário inicialmente deve verificar se o firewall do Windows está desabilitado, conforme seguem as instruções do site oficial da Microsoft <http://windows.microsoft.com/pt-br/windows/turn-windows-firewall-on-off#turn-windows-firewall-on-off=windows-7>, alguns sistemas operacionais bloqueiam por padrão o eco dessas mensagens e necessitam de um cuidado especial. Feito isso, o IP pode ser descoberto, por exemplo, desconectando a UCP da rede e executando um comando “ping” no seu endereço IP. Como a UCP está desconectada, espera-se que este “ping” falhe. Se o “ping” responder, existe outro equipamento com o mesmo endereço IP.

Caso o endereço IP já esteja em uso por outro equipamento na rede, deve-se executar o próximo passo, e alguns passos seguintes, utilizando um cabo de crossover para conectar diretamente o PC com o software MasterTool Xtorm à UCP, evitando assim conflitos de endereços IP. Em um dos passos seguintes, ao carregar o projeto na UCP, serão atualizados os endereços IP definitivos da UCP.

### **Quarto Passo – Preparar Conexão do MasterTool (Definir Caminho Ativo)**

O quarto passo consiste em dar um duplo clique sobre o Device (HX3040) na árvore de dispositivos, entrar na aba “Configurações de Comunicação”, clicar sobre o Gateway, e pressionar o botão “Mapear Rede” para listar todos os UCPs detectados pelo MasterTool na rede.

Neste momento, espera-se encontrar uma lista de UTRs cuja identificação contém o endereço IP descoberto no primeiro passo e o tipo da UCP (HX3040). Caso o usuário tenha trocado anteriormente o nome da UCP na rede, este nome será o visualizado neste momento. A seção Localizando a Rede descreve com maiores detalhes as possíveis identificações que podem ser observadas nesta lista, bem como a seleção e envio do projeto.

### **Quinto Passo – Carga do Projeto Redundante**

Este passo descreve a carga do projeto redundante na UCP. Este projeto deve ser preparado conforme descrito na seção Programação de uma UCP Redundante. Um projeto redundante básico pode ser preparado seguindo, no mínimo, as seguintes subseções desta seção:

- Wizard para Criação de um Novo Projeto Redundante;
- Configurações das Portas Ethernet da UCP HX3040.

Obviamente, também é possível fazer um projeto redundante completo e só depois carregá-lo na UCPA e UCPB, por exemplo, caso o hardware destas UTRs não esteja disponível durante o desenvolvimento do projeto com o MasterTool.

A primeira carga de um projeto redundante em uma UCP ainda deve ser feita utilizando aquele endereço IP descoberto no primeiro passo deste procedimento, e selecionado nos passos que se sucedem a este procedimento.

A carga do projeto é feita através do menu Comunicação / Login.

### 🔗 NOTAS:

1. Ao fim do envio do projeto para uma das UCPs, a outra UCP fará download automaticamente do projeto por canais de sincronismo internos.
2. Dentro do projeto desenvolvido com o MasterTool e carregado na UCP neste passo, foram definidos novos endereços IP para a interface NET 1 da UCPA e UCPB (Endereço de IP da UCPA e Endereço de IP da UCPB), assim como um endereço IP para a interface NET 1 da UCP Ativa (Endereço de IP Ativo). Portanto, depois desta carga inicial, aquele endereço IP descoberto no primeiro passo deste procedimento normalmente não será mais válido. Esta mudança do endereço IP em NET 1 provocará uma perda da conexão do MasterTool com a UCP, que será notificada, somente a NET1 pode ser utilizada para comunicação com a UCP via MasterTool.

## Conexão do MasterTool com uma UCP HX3040 de uma UCP Redundante

Depois de executar o procedimento descrito na seção Carga Inicial de um Projeto Redundante nas duas UCPs (UCPA e UCPB), a conexão ao MasterTool, através da interface NET 1 da UCP HX3040, poderá ser feita através de um dos seguintes endereços:

- **Endereço de IP da UCPA** - Endereço de NET 1 exclusivo para a UCPA;
- **Endereço de IP da UCPB** - Endereço de NET 1 exclusivo para a UCPB.

Independente do estado da UCP, o MasterTool só consegue se conectar ao mesmo utilizando o endereço exclusivo da UCP da NET1, configurado em Endereço de IP da UCPx. Porém, caso a UCP se encontre em estado Ativo, todos os outros serviços poderão se conectar à UCP tanto pelo Endereço de IP da UCPx quanto pelo Endereço de IP Ativo.

Para conectar-se a determinada UCP, em primeiro lugar deve-se dar um duplo clique sobre o Device (HX3040) na árvore de dispositivos, entrar na aba “Configurações de Comunicação”, clicar sobre o Gateway, e pressionar o botão “Mapear Rede” para listar todos as UTRs detectados pelo MasterTool na rede.

Nesta lista, será possível encontrar as seguintes identificações padrão, caso o nome da UCP na rede não tenha sido alterado anteriormente pelo usuário:

- **HX3040\_<IP address>\_PLCA** - Identificação relacionada à UCPA. Neste caso, o campo <IP address> deve coincidir com o Endereço de IP da UCPA configurado no projeto;
- **HX3040\_<IP address>\_PLCB** - Identificação relacionada à UCPB. Neste caso, o campo <IP address> deve coincidir com o Endereço de IP da UCPB configurado no projeto.

A seguir, deve-se selecionar a UCP desta lista onde o MasterTool deve se conectar, e pressionar o botão “Definir Caminho Ativo”. Posteriormente, ao executar o comando do menu Comunicação / Login, o MasterTool se conectará a esta UCP.

### 🔗 NOTA:

O MasterTool só consegue se conectar a uma UCP de cada vez. Para conectar-se a diversas UCPs, devem-se abrir múltiplas instâncias do MasterTool, cuidando sempre para abrir o projeto correto em cada instância, por exemplo, conectando-se a UCPA em uma instancia do MasterTool,

abrindo o mesmo projeto e conectando-se a UCPB. Alterações não devem ser feitas na condição de múltiplas instancias do mesmo projeto aberto.

### **Carga de Modificações em um Projeto Redundante**

Depois que as duas UCPs (UCPA e UCPB) que compõe o sistema redundante já receberam uma carga inicial, conforme descrito na seção Carga Inicial de um Projeto Redundante, é possível carregar modificações sucessivas do projeto, à medida que tais modificações forem necessárias.

A conexão do MasterTool às UCPs para executar a carga de modificações deve ser feita conforme descreve-se na seção Conexão do MasterTool com uma UCP HX3040 de uma UCP Redundante. Nesta seção explica-se como é possível conectar-se a uma UCP específico (UCPA ou UCPB), ou à UCP Ativa, ou à UCP Não-Ativa.

As modificações sempre devem ser carregadas na UCP Ativa, que se responsabilizará por transferi-las automaticamente para a UCP Não-Ativa, através dos canais de sincronismo entre as UCPs. Portanto, o MasterTool normalmente deve usar o endereço de IP exclusivo da UCP que se encontra no estado Ativo (endereço de IP da UCPx), para se conectar à porta NET 1 da UCP HX3040 da UCP Ativa.

#### **☞ NOTA:**

Não é possível carregar um projeto na UCP Não-Ativa, enquanto a outra UCP estiver no estado Ativo. Só será possível carregar um projeto ou fazer modificações online na UCP Não-Ativa se a outra UCP não estiver no estado Ativo (tipicamente no estado Não-Configurado ou Inativo) ou se estiver ausente do barramento. Da mesma forma, só é possível carregar um projeto na UCP Ativa, caso a outra UCP encontre-se no estado Inativo ou não esteja presente no barramento.

### **Carga de Modificações Off-Line e On-Line**

Modificações de projeto podem ser carregadas off-line ou on-line.

Cargas off-line requerem a parada da UCP onde a modificação deve ser carregada. Por outro lado, cargas on-line permitem que a UCP continue executando sua aplicação enquanto a modificação é carregada.

Alguns tipos de modificações exigem carga off-line, ou seja, não podem ser carregadas on-line na UCP onde o MasterTool está conectado.

### **Modificações que Demandam Carga Off-Line com Interrupção do Controle do Processo**

As seguintes modificações em um projeto farão com que se torne impossível a carga deste num sistema redundante sem a interrupção do controle do processo:

- Adicionar ou remover dispositivos na árvore de dispositivos, por exemplo: módulos de E/S e instâncias de protocolos de comunicação, como MODBUS, DNP3 etc.;
- Modificar parâmetros dentro de dispositivos existentes na árvore de dispositivos, por exemplo: endereços IP e outros parâmetros de interfaces Ethernet, parâmetros de módulos de E/S e parâmetros de protocolos instanciados, seus mapeamentos etc.;
- Modificações nas configurações das tarefas.

### **Modificações que Permitem Carga On-Line**

A seguir serão citadas as principais modificações que permitem carga on-line na UCP onde o MasterTool está conectado. As modificações citadas abaixo valem para variáveis, POU's e GVL's, redundantes ou não.

- Adicionar POUs do tipo programa, desde que estas POUs não precisem ser associadas a alguma tarefa;
- Remover POUs do tipo programa, desde que estas POUs não estejam associadas a alguma tarefa;
- Adicionar ou remover POUs do tipo função ou bloco funcional;
- Modificar o código de quaisquer tipos de POU (programa, função ou bloco funcional);
- Adicionar ou remover variáveis simbólicas em quaisquer tipos de POU (programa, função ou bloco funcional, sendo elas redundantes ou não);
- Adicionar ou remover instâncias de bloco funcional em POUs do tipo programa ou bloco funcional;
- Adicionar ou remover GVLs;
- Adicionar ou remover variáveis simbólicas ou instâncias de bloco funcional em GVLs.

Existem ainda dois métodos de cargas on-line, descritos a seguir, um para compilação e envio da Aplicação de Inicialização feito manualmente e o outro automaticamente ao fim do envio da carga On-line. Para maiores informações o Manual de Utilização do MasterTool IEC XE MT8500 - MU299048 deve ser consultado.

### **Carga On-Line de Modificações**

Uma carga on-line deve ser feita conectando o MasterTool ao canal NET 1 da UCP Ativa, utilizando seu endereço exclusivo de IP (IP da UCPA ou IP da UCPB). É necessário trocar o estado de redundância da UCP Reserva para Inativo antes de enviar a modificação On-Line para a UCP Ativa, caso contrário o MasterTool bloqueará a operação. Após o envio das modificações, é criada automaticamente a aplicação de inicialização na memória não volátil da UCP, e não se corre o risco de perder as modificações no caso de a UCP ser desligada ou no caso de falta de energia.

Depois de finalizado o processo de carga On-Line, pode-se tirar a outra UCP do estado Inativo, por exemplo, usando o comando de “Trocar para Reserva” no menu do seu visor. Isso provocará a transição da UCP para o estado Não-Configurado. A UCP permanecerá no estado Não-Configurado até que o processo automático de sincronização de projetos termine. Depois disso, a UCP passa para o estado Inicializando e depois voltará para o estado Reserva. ou voltar para o estado Inativo se ocorrer alguma falha.

### **Carga Off-Line de Modificações com Interrupção do Controle do Processo**

Nesta seção, define-se o procedimento para executar uma carga off-line que interromperá o controle do processo. Tal situação é aceitável em determinados tipos de processos e durante paradas programadas dos processos.

Uma carga off-line deste tipo deve ser feita conectando o MasterTool ao canal NET 1 da UCP Ativa, utilizando o endereço exclusivo da UCP em estado Ativo (Endereço de IP da UCPx). Antes de iniciar uma carga off-line na UCP Ativa, é necessário passar a UCP reserva para o estado Inativo. Se isto não for feito, o MasterTool não permitirá que a modificação seja aplicada.

Para executar a carga off-line, exige-se então que conscientemente a UCP reserva seja colocada para o estado Inativo, impedindo então que uma carga acidental de programa seja executada.

No momento que carga off-line se iniciar a UCP Ativa irá para o estado Não-Configurado.

Quando a carga off-line terminar, é possível reiniciar a execução do programa na UCP onde a aplicação foi carregada (colocar no modo RUN novamente). Depois de alguns segundos, esta UCP reassume o estado Ativo.

Depois que esta UCP reassume o estado Ativo, pode-se tirar a outra UCP do estado Inativo, por exemplo, usando o comando de “Trocar para Reserva” no menu do seu visor. Isso provocará a transição da UCP para o estado Não-Configurado. Esta UCP permanecerá no estado Não-Configurado até que o processo automático de sincronização de projetos termine. Depois disso, a UCP passa para o estado Inicializando e depois voltará para o estado Reserva.

# Manutenção de UTRs Redundantes



## Mensagens de Advertência do MasterTool

Quando o MasterTool for efetuar uma conexão ou já estiver conectado a uma UCP com projeto redundante, algumas mensagens de advertência especiais poderão ocorrer, conforme se descreve nas próximas subseções.

### Bloqueio Antes de Comandos que Podem Parar a UCP Ativa

Alguns comandos, como os seguintes, podem parar uma UCP:

- Carga offline após Comunicação / Login;
- Depurar / Parar;
- Depurar / Novo Breakpoint;
- Comunicação / Reset (quente, frio, origem).

Estes comandos só serão disparados caso a UCP Ativa esteja operando com a redundância desarmada, ou seja, a outra UCP deve estar no estado Inativo (ou ausente do barramento). Caso contrário, uma mensagem de bloqueio será mostrada: “O Login não pode ser realizado, pois o projeto atual é diferente do projeto em uso na UCP. Caso realmente deseje executar esta operação, configure a outra UCP para o estado inativo e execute novamente este comando”.

### Bloqueios de Operações na UCP Não-Ativa

Em algumas circunstâncias, certas operações são bloqueadas na UCP Não-Ativa. Caso a outra UCP esteja no estado Ativo, o que é o usual no caso da redundância, certas operações serão bloqueadas na UCP Não-Ativa:

- Carga offline e online;
- Troca de estado de aplicação (RUN/STOP);
- Depurar / Parar;
- Depurar / Novo Breakpoint.

Quando houver uma tentativa de execução de um comando deste tipo, o MasterTool emite a seguinte mensagem: “ Esta operação só poderá ser executada na outra UCP, porque ela está controlando o processo”.

Por outro lado, é possível logar-se na UCP Não-Ativa para monitorar ou forçar variáveis não-redundantes.

## Interação com a Redundância Através do Visor Gráfico da UCP HX3040

O estado da redundância e operações disponíveis no menu REDUNDANCIA podem ser acessadas no visor da UCP HX3040.

### Estado de Redundância da UCP

O estado de redundância da UCP pode ser conferido nos três caracteres iniciais da segunda linha da tela principal (veja a seção “Manutenção – Visor Gráfico” deste Tutorial). A tela é apresentada após a inicialização da UCP, e volta a ser apresentada alguns segundos depois de terminada qualquer navegação (sem apertar o botão da UCP HX3040).

## Telas Abaixo do Menu Redundância

Existe um menu denominado Redundância, abaixo do qual existem algumas telas. A descrição e o acesso às telas de redundância estão disponíveis na seção “Menu Informativo e de Configuração da UCP” deste Tutorial.

## Estrutura de Diagnósticos da Redundância

A área de diagnósticos da redundância corresponde a variável simbólica DG\_HX3040\_01, alocada automaticamente pelo MasterTool na GVL System\_Diagnostics. Esta área é dividida em outras seis estruturas de dados:

1. **RedDgnLoc** - Contém diagnósticos de redundância da UCP local (onde o MasterTool está conectado), como por exemplo, o estado da redundância da UCP;
2. **RedDgnRem** – É uma cópia de RedDgnLoc da outra UCP, recebida via canais de sincronismo. Desta forma, a UCP local tem acesso aos diagnósticos da UCP remota;
3. **RedCmdLoc** - Contém comandos de redundância gerados nesta UCP (local), por exemplo, através de escritas a partir de um sistema SCADA, ou gerados em POU's desta UCP (ex: UserPrg ou NonSkippedPrg);
4. **RedCmdRem** - Trata-se de uma cópia de RedCmdLoc da outra UCP (remota), recebida via canais de sincronismo;
5. **RedUsrLoc** - Área de memória de 128 bytes da UCP local, que será transferida para a estrutura de dados RedUsrRem da UCP remota. É utilizada para que o usuário troque informações entre UCPA e UCPB;
6. **RedUsrRem** - Área de memória de 128 bytes na UCP local, que corresponde à estrutura de dados RedUsrLoc da UCP remota. É utilizada para que o usuário troque informações entre UCPA e UCPB.

É importante ressaltar que as estruturas de diagnósticos da redundância da UCP remota são atualizadas somente quando ocorre uma sincronização de dados com sucesso. Portanto, enquanto uma nova sincronização não ocorrer, os diagnósticos irão permanecer com o valor congelado, correspondente à última troca de dados realizada com sucesso.

Além disso, as estruturas da UCP remota são somente para leitura, isto é, valores escritos nestas estruturas serão sobrescritos, sem serem considerados, na próxima sincronização de dados. Sendo assim, não é possível utilizar a estrutura RedCmdRem para executar um comando na UCP remota. A estrutura utilizada para executar comandos, a ser escrita, deve ser sempre RedCmdLoc.

## Diagnósticos da Redundância

Os diagnósticos da Redundância podem ter várias utilidades. Algumas delas são:

- Podem ser consultados para verificar a existência de algum problema que precisa ser sanado;
- Podem ser utilizados para consultar os status do sistema, bem como a tomada de ações em caso de falhas ou de variações de valores de diagnósticos. Os principais eventos de redundância relacionados a diagnósticos do sistema são apresentados como eventos nos logs de sistema. Consultando a sequência histórica de tais eventos pode-se descobrir, por exemplo, a causa de um switchover.

### ¶ NOTA:

O diagnóstico DG\_HX3040\_01.RedDgnLoc.sGeneral\_Diag.bExchangeSync (definido logo a seguir) deve ser testado para verificar se a estrutura de dados RedDgnRem foi lida com sucesso da UCP remota no último ciclo da MainTask. Caso o valor deste diagnóstico seja FALSE, isso

significa que a estrutura de dados RedDgnRem não foi lida com sucesso da UCP remota e, portanto, os valores de RedDgnRem podem ser inválidos ou estar desatualizados.

Como RedDgnRem é uma cópia de RedDgnLoc da outra UCP, as duas estruturas têm o mesmo formato. Estas ainda são divididas em quatro subestruturas:

1. **RedundancyDiagnostics** - Diagnósticos gerais da redundância;
2. **SyncLinkDiags[1]** - Diagnósticos do canal de sincronismo A;
3. **SyncLinkDiags[2]** - Diagnósticos do canal de sincronismo B;
4. **SyncLinkStatistics** - Estatísticas comuns para os canais de sincronismo A e B, para contagem de sucessos e falhas dos serviços de sincronização.

### 🔔 **NOTA:**

As estruturas de diagnóstico adicionadas ao projeto podem ser visualizadas acessando o item “Library Manager” na treeview da janela do MasterTool Xtorm. Com isso, é possível visualizar todos os datatypes definidos na estrutura.

A subestrutura “SyncLinkStatistics” contém estatísticas de falhas e sucessos dos serviços. As estatísticas das UCPs local e remota podem ser reiniciadas através dos comandos:

```
//UCP Local  
DG_HX3040_01.RedCmdLoc.bResetNETStatisticsLocal := TRUE;  
//UCP Remota  
DG_HX3040_01.RedCmdLoc.bResetNETStatisticsRemote := TRUE;
```

Várias mensagens podem ser apresentadas para os estados e trocas de redundância, acessíveis ao usuário por meio do diagnóstico “DG\_HX3040\_01.RedDgnLoc.RedundancyDiagnostics.eStateChangeReason.”. As mensagens apresentadas na descrição são idênticas às mensagens que serão apresentadas no Log.



### **Pesquisa na Documentação: diagnósticos da redundância**

Consulte, na documentação do produto, os campos para diagnósticos gerais da redundância associados à subestrutura “RedundancyDiagnostics”, os campos para diagnósticos dos dois canais de sincronismo (array 1 e 2) via subestrutura “SyncLinkDiags”, os contadores de diagnósticos específicos da interface da subestrutura “SyncLinkStatistics” e as razões de troca de estados da redundância.

Fonte: Manual de Utilização Hadron Xtorm – MU223000!

### **Comandos da Redundância**

Os campos de comandos das estruturas RedCmdLoc e RedCmdRem possuem sempre um sufixo que pode ser Local ou Remote. Por exemplo, existem os campos de comando StandbyLocal e StandbyRemote, que têm efeito equivalente ao comando de “Trocar para Reserva” do visor da UCP. Um comando com sufixo Local gerado em RedCmdLoc será executado na própria UCP (local). Por outro lado, um comando com sufixo Remote gerado em RedCmdLoc será executado na outra UCP (remota). Isto funciona da seguinte maneira:

1. A UCP remota, a cada ciclo da MainTask, recebe uma cópia de RedCmdLoc da UCP local via canais de sincronismo, e esta cópia é chamada de RedCmdRem na UCP remota;
2. A UCP remota somente executa comandos de RedCmdRem que tenham o sufixo Remote.

Exemplo 1: se a UCP local estiver em estado Ativo, e deseja-se chaveá-la para o estado Reserva, deve-se ligar o bit DG\_HX3040\_01.RedCmdLoc.bStandbyLocal na UCP local.

Exemplo 2: se a UCP remota estiver em estado Ativo, e deseja-se chaveá-la para o estado Reserva, deve-se ligar o bit DG\_HX3040\_01.RedCmdLoc.bStandbyRemote na UCP local.

**❗ NOTA:**

Se o diagnóstico DG\_HX3040\_01.RedDgnLoc.RedundancyDiagnostics.bExchangeSync estiver indicando falha no serviço Troca de Diagnósticos e Comandos, um comando com sufixo Remote não poderá ser repassado para a UCP remota, e portanto não será executado.

Para disparar um comando, deve-se sempre ligar o bit correspondente em RedCmdLoc. Isto pode ser feito por um sistema SCADA, fazendo uma escrita via MasterTool, ou até mesmo ligando o bit dentro de uma POU como UserPrg ou NonSkippedPrg.

O usuário não precisa se preocupar com o desligamento do bit de comando, que será feito automaticamente pelo gerenciador de redundância:

- No caso de comandos executados na UCP local (RedCmdLoc + comando com sufixo Local), o bit é desligado assim que o comando for percebido e executado;
- No caso de comandos executados na UCP remota (RedCmdRem + comando com sufixo Remote);
- Na UCP remota, o comando é executado quando o gerenciador de redundância percebe uma borda de subida no bit de comando.

**Pesquisa na Documentação: comandos da redundância**

Consulte, na documentação do produto, os campos das estruturas RedCmdLoc e RedCmdRem, os quais compõem os comandos da redundância.

Fonte: Manual de Utilização Hadron Xtorm – MU223000!

**Informações do Usuário Trocados entre UCPA e UCPB**

O serviço de sincronização Troca de Diagnósticos e Comandos, troca as seguintes estruturas de dados entre as duas UCPs em cada ciclo da MainTask, usando os canais de sincronismo A/B:

- Diagnósticos de Redundância (RedDgnLoc e RedDgnRem);
- Comandos de Redundância (RedCmdLoc e RedCmdRem);
- Informações do Usuário Trocados entre UCPA e UCPB (RedUsrLoc e RedUsrRem).

Os dois primeiros itens já foram abordados nesse Tutorial. Já, as estruturas RedUsrLoc e RedUsrRem são simplesmente um array de 128 bytes, cuja utilização pode ser livremente definida pelo usuário. Elas permitem que o usuário transfira, a cada ciclo, 128 bytes de informação da UCPA para a UCPB, e outros 128 bytes da UCPB para a UCPA.

RedUsrRem é uma cópia de RedUsrLoc da outra UCP, recebida via canais de sincronismo internos. Determinada UCP escreve informações em RedUsrLoc, que serão recebidas na outra UCP em RedUsrRem.

**Logs de Eventos da Redundância**

O MasterTool permite observar diversos logs para uma UCP Hadron Xtorm, entre os quais encontra-se os logs de eventos da redundância. Estas mensagens, específicas para a redundância, são registradas no Log de Sistema.

São registradas modificações relevantes nos campos das estruturas de dados de diagnósticos e comandos de redundância, que são as seguintes: RedDgnLoc, RedDgnRem, RedCmdLoc, RedCmdRem.

No caso das estruturas de diagnósticos, apenas os seguintes campos não geram logs de eventos: wRedStateDuration, wCycleCounter, dwIECTimer e SyncLinkStatistics.

Cada linha mostrada no log possui as seguintes colunas:

- **Marca de Tempo** - Data e hora do evento, com resolução de milissegundos;
- **Severidade** - Informação, advertência, erro ou exceção;
- **Descrição** - Texto que descreve o evento;
- **Componente** - Componente que gerou o evento, que no caso do log de eventos da redundância será o “Redundancy Management”.

O texto na coluna “Descrição” contém informações referentes ao evento. Por exemplo, no caso da troca de estado da redundância de Reserva para Ativo, será gerada uma entrada no log indicando que houve a troca de estado, do estado Reserva para o Ativo.

Para acessar esta tela, deve-se dar um duplo clique sobre o dispositivo (HX3040) na árvore de dispositivos, e depois selecionar a aba “Log”. Existe um filtro que permite selecionar somente o componente “Redundancy Management”, para exibir somente os eventos de redundância.

🔔 **NOTA:**

Alguns diagnósticos podem apontar possíveis falhas durante a inicialização do sistema redundante e nos primeiros ciclos das tarefas. No entanto, em um correto funcionamento do sistema, estes diagnósticos voltam a indicar a ausência de erros logo após a inicialização do sistema.

## 5. Aplicações, Avaliação e Encerramento

### Estudos Dirigidos no MasterTool Xtorm



#### Estudo Dirigido 5-1: Projeto-exemplo no MasterTool Xtorm

Esta seção aborda um projeto-exemplo no MasterTool Xtorm, o qual explora as principais funcionalidades associadas à Série Hadron Xtorm.

O projeto prevê a seguinte configuração-base para o barramento (bus) no bastidor HX9001 (00):

- HX8320 (0);
- HX3040 (1);
- HX1120 (2);
- HX2320 (3);
- HX6000 (4);
- HX6000\_01 (5).

Assim sendo, implemente a configuração indicada, conferindo, no explorador de projeto, a Lista de Materiais e a Configuração e Consumo do Bastidor.

No âmbito da comunicação, habilite a NET 2 com os parâmetros adequados, visto que o projeto deve prever que a UCP seja servidor da comunicação IEC 61850, permitindo a conexão com clientes IEC 61850 através do protocolo MMS e possibilitando o envio e recepção de mensagens rápidas através do protocolo GOOSE.

As Unidades de Organização de Programa (POUs) associadas ao protocolo IEC61850 podem ser conferidas na pasta IEC61850 Generated POU's, assim como as variáveis globais correspondentes (IEC61850\_Generated\_GVL).

A Configuração de Tarefas deve incluir, além da tarefa principal, uma tarefa cíclica denominada IEC61850Task1 que chama a POU IEC61850\_ServerPRG.

As GVLs e POU's do sistema podem ser conferidas nas pastas respectivas.

Da mesma forma, as GVLs e POU's do usuário podem ser conferidas nas pastas respectivas, com ênfase nas GVLs denominadas como DadosComunica, as quais contemplam os mapeamentos de E/S e protocolos e na POU UserPrg, a qual inclui a lógica Ladder para os parâmetros de qualidade.

Em um nível mais acima no Explorador de Projeto, o Gateway deve ser definido e mapeado.

A figura a seguir ilustra o barramento e uma visão geral do Explorador de Projeto.



## Estudo Dirigido 5-2: Utilização do Recurso de Retentividade na Hadron Xtorm

Essa situação de aprendizagem tem como objetivo fazer conhecer o recurso de Retentividade de dados oferecido pelas UCPs da Série Hadron Xtorm.

### Considerações Iniciais

Conforme abordado anteriormente nesse Tutorial, as UCPs da Série Hadron Xtorm possuem essencialmente dois tipos de memória: a memória de programa (FLASH), onde é armazenado o código da aplicação desenvolvida pelo usuário, e a memória de dados (RAM), onde são armazenadas as variáveis utilizadas pela aplicação.

A memória de dados é, por padrão, uma memória do tipo volátil, isto é, as variáveis alocadas nesta área retornam ao seu valor inicial após um evento de reinicialização da UCP (interrupção da alimentação, por exemplo). Contudo, em diversos casos, é necessário que algumas variáveis mantenham o seu valor mesmo após a inicialização da UCP. Isto ocorre, por exemplo, para variáveis que armazenam algum tipo de ajuste, como um setpoint de controle ou parâmetros de uma máquina ou processo. Para atender a esta necessidade, as UCPs possuem o recurso de Retentividade de dados, o qual permite que um determinado conjunto de variáveis se torne não volátil (isto é, os valores são mantidos mesmo após um evento de reinicialização da UCP).

É importante reforçar que as UCPs da Série Hadron Xtorm oferecem apenas memória de dados não volátil do tipo simbólica.

### Arquitetura de Referência

Para essa situação de aprendizagem, você deverá utilizar uma arquitetura simples composta apenas de barramento, fonte e UCP. A resolução do estudo dirigido encontra-se no final deste Tutorial.

### Parte 1 - Memória de Variáveis Simbólicas Retentivas

Variáveis simbólicas retentivas podem ser declaradas como locais ou globais. Para criar uma variável simbólica retentiva local, basta colocá-la em uma seção "VAR RETAIN" "END\_VAR" na área de declarações da POU em questão. Assim sendo, pede-se:

1. Declare uma variável simbólica retentiva do tipo BYTE com o nome "VariavelRetentiva".
2. Declare duas variáveis simbólicas retentivas GLOBAIS do tipo BYTE denominadas: "VariavelRetentiva01" e "VariavelRetentiva02".

### Parte 2 - Memória de Variáveis Simbólicas Persistentes

Variáveis simbólicas Persistentes devem ser declaradas única e exclusivamente em uma lista especial chamada "Variáveis Persistentes", onde todas as variáveis declaradas são globais por definição. Para tanto, pede-se:

3. Declare duas variáveis simbólicas persistentes do tipo BYTE denominadas: "VariavelPersistente01" e "VariavelPersistente02".

**⚠️DICA:** para que o recurso de Retentividade de dados das UCPs seja utilizado da melhor forma possível, é importante que algumas boas práticas de utilização sejam seguidas, assim como é essencial compreender as restrições inerentes a esta funcionalidade. Os seguintes aspectos devem ser considerados:

1. A lista de variáveis persistentes foi desenvolvida para armazenar um conjunto específico e limitado de variáveis. Em função de algumas características intrínsecas ao seu mecanismo de funcionamento, esta lista possui certas restrições relacionadas à alteração de tipos de variáveis conforme descrito na seção Alteração de Tipo em Variáveis

Retentivas e Persistentes. Adicionalmente, esta lista necessita que as variáveis nela declaradas sejam alocadas de forma sequencial e ordenada na memória;

2. Caso uma variável seja incluída no início ou no meio da lista, o compilador alocará esta variável no primeiro endereço de memória livre (no final da lista), garantindo que as outras variáveis não serão afetadas. De forma semelhante, quando o tipo de uma variável, estrutura ou Function Block já declarado é alterado, o compilador desloca esta variável para o primeiro endereço de memória livre (final da lista). Como se pode observar, estas ações acabam criando "buracos" na área de memória, consequentemente causando a sua fragmentação, o que pode levar ao esgotamento prematuro desta área de memória. Desta forma, recomenda-se que, durante o desenvolvimento da aplicação, novas variáveis sejam incluídas no final da lista, assim como se deve evitar a alteração do tipo de variáveis já declaradas, sob a pena de causar fragmentação da área de memória;
3. Adicionalmente, o comando Compilar -> Limpar Tudo realiza a desfragmentação desta área. Contudo, a execução desta operação implicará na reinicialização de todas as variáveis da lista no próximo download;
4. Não é recomendado realizar a declaração de todas as variáveis do projeto de forma indiscriminada na lista de Variáveis Persistentes. Recomenda-se realizar uma análise da aplicação e declarar nesta lista somente aquelas variáveis que realmente precisam manter o seu valor em caso de reinicialização da UCP;
5. A utilização de Variáveis Retentivas dentro de Function Blocks possui uma característica importante e que deve ser considerada no projeto da aplicação, pois caso o Function Block possua Variáveis Retentivas declaradas (VAR RETAIN), todas as outras variáveis do bloco também se tornam Retentivas (VAR, VAR\_INPUT e VAR\_OUTPUT);
6. Alterações de tipo em variáveis, estruturas de dados e blocos de função já existentes na aplicação requerem um cuidado especial. Para tanto, a documentação da Série deve ser consultada.

ANOTAÇÕES

## Estudos de Caso da Hadron Xtorm



### Estudo de Caso 5-1: modernização da UHE Fontes Novas

Mesmo com o constante surgimento de diferentes fontes de energia alternativas, as usinas hidrelétricas com reservatórios de acumulação seguem como os principais empreendimentos energéticos do Brasil. Hoje, as UHEs respondem por cerca de 60 % de toda energia gerada no país, e a tecnologia Altus é responsável por um importante percentual desta produção. Há mais de 30 anos presentes no mercado nacional de Energia Elétrica, nossos produtos são utilizados pelos principais players do setor como solução de controle e aquisição de dados em processos de Geração, Transmissão e Distribuição.

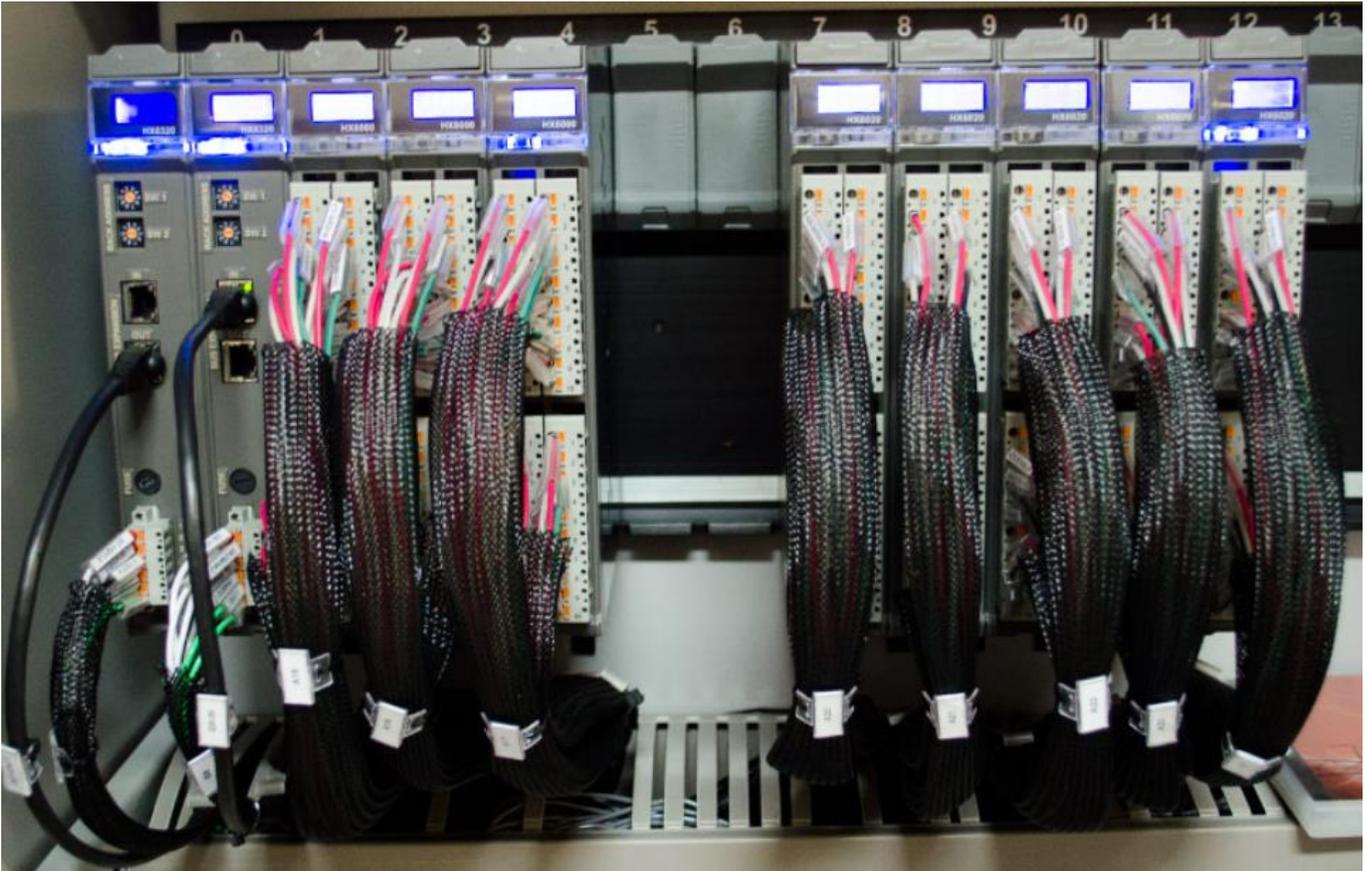


Um dos projetos liderado pela equipe de Engenharia da Altus é a modernização das UACs (Unidades de Controle e Automação) da Subestação e dos Serviços Auxiliares da UHE Fontes Novas, empreendimento do Grupo Light integrante do Complexo de Lajes. Localizada no Ribeirão das Lajes, região sudeste do estado do Rio de Janeiro, a usina conta com três unidades geradoras que, somadas, entregam uma capacidade instalada de 132 MW. O upgrade na aplicação visa aumentar a integridade e a confiabilidade operacional, ao mesmo tempo que promove a padronização dos sistemas da UHE, viabilizando melhores condições de manutenção preventiva e corretiva.

A modernização projetada substitui os CLPs AL-2000, que controlavam as UACs da Subestação e dos Serviços Auxiliares, pelas UTRs da Série Hadron Xtorm. Este mesmo processo de substituição foi realizado nas Unidades Geradoras A, B e C da usina. Desde então, as unidades estão operando com base na tecnologia Hadron Xtorm e em completa integração com o CAO (Centro de Atendimento a Operação) e o COG (Centro de Operação da Geração).

### Estudo de Caso 5-2: modernização da UHE Santa Branca

Outro empreendimento da Light que também teve sua base tecnológica atualizada com os produtos Hadron Xtorm foi a UHE Santa Branca. O projeto da usina engloba a completa modernização do SDSC, contemplando desde o fornecimento das UACs das Unidades Geradoras 1 e 2, da Subestação e dos Serviços Auxiliares, até o desenvolvimento da aplicação do CLP e do sistema de supervisão. O TAF do projeto, conduzido pela equipe da Altus e com a presença de engenheiros e técnicos do cliente e foi realizado na Fábrica de Painéis, em Sapucaia do Sul/RS.



### Estudo de Caso 5-3: modernização da PCH Pai Joaquim

Com mais de 30 anos de atuação no mercado energético nacional, o time de engenharia da Altus possui ampla experiência no desenvolvimento e integração de sistemas para geração, transmissão e distribuição de energia. Uma dessas iniciativas foi a modernização do Sistema Digital de Supervisão e Controle (SDSC) da PCH Pai Joaquim, empreendimento instalado às margens do rio Araguari, em Minas Gerais. Com capacidade produtiva de 23 MW, a pequena central hidrelétrica faz parte do sistema de produção energética da CEMIG, uma das principais concessionárias de energia elétrica do Brasil.

Neste projeto, a Altus foi responsável pela fabricação dos produtos e pelo desenvolvimento do novo SDSC em comissionamento na PCH. Para a aplicação de controle, foram utilizadas duas das nossas principais séries de produtos: as Unidades Terminais Remotas Hadron Xtorm e os Controladores Lógicos Programáveis Nexto. As UTRs Hadron Xtorm são responsáveis pelo controle da unidade geradora e os produtos da Série Nexto atuam na aquisição e transmissão de sinais entre todos os dispositivos que fazem parte da aplicação, como sensores, RTVX e as próprias UTRs.

Já a aplicação de supervisão foi desenvolvida utilizando como base a tecnologia do software SCADA/HMI BluePlant. Solução da Altus para supervisão, controle e aquisição de dados, o BluePlant alia design, desempenho e alta conectividade, possui uma arquitetura cliente-servidor com inúmeras opções de quantidades de tags de comunicação, além de simulador e ferramentas de depuração. Assim como no sistema de controle, todas as telas e lógicas do sistema também foram projetadas e modeladas pelos especialistas da nossa equipe de engenharia.

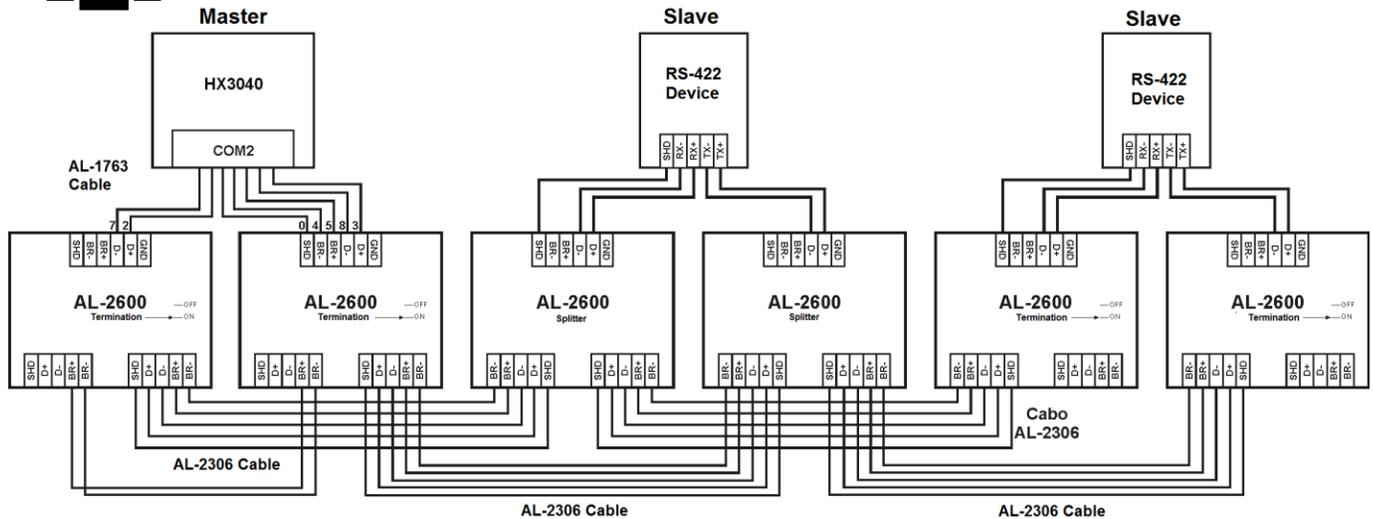


📌 **DICA:** consulte o Instrutor para informações adicionais sobre os Estudos de Caso descritos ou mesmo de outras aplicações da Série Hadron Xtorm.

# Resolução dos Estudos Dirigidos



## Resolução do Estudo Dirigido 1-1



## Resolução do Estudo Dirigido 1-2

```

PROGRAM UserPrg
VAR
sAppName : STRING;
psAppName : POINTER TO STRING;
sTaskName : STRING;
psTaskName : POINTER TO STRING;
pstTaskInfo : POINTER TO ST_TASK_INFO;
TaskInfo : ST_TASK_INFO;
Info : TYPE_RESULT;
END_VAR
//ENTRADAS:
sAppName := 'Application'; //Variável recebe o nome da aplicação.
psAppName := ADR(sAppName); //Ponteiro com o nome da aplicação.
sTaskName := 'MainTask'; //Variável recebe o nome da tarefa.
psTaskName := ADR(sTaskName); //Ponteiro com o nome da tarefa.
pstTaskInfo := ADR(TaskInfo); //Ponteiro que irá receber as
informações da tarefa.
//FUNÇÃO:
//Chamada da função.
Info := GetTaskInfo (psAppName, psTaskName, pstTaskInfo);
//Variável 'Info' recebe possíveis erros da função.

```

## Resolução do Estudo Dirigido 2-1

A criação de um projeto no MasterTool Xtorm segue os seguintes passos básicos:

1. Inicialmente, o usuário deverá criar um projeto no MasterTool Xtorm a partir do menu Arquivo e logo em seguida, “New Project...”;

2. Após, uma janela será apresentada ao usuário, solicitando que ele selecione o tipo de projeto que deseja fazer e, em seguida, escreva um nome e a localização para armazenar o projeto no computador. Clicar em OK para prosseguir ou Cancel para interromper;
3. A seguir, o usuário deverá selecionar a UCP desejada, os módulos de hardware básicos que compõem o barramento, ou seja, o modelo de bastidor e de fonte de alimentação e a configuração de redundância;
4. Caso o usuário selecione a opção para redundância de UCP, a opção para redundância de fonte de alimentação será automaticamente marcada. Esta é uma característica de projeto, sendo que para projetos com redundância de UCP, será obrigatoriamente necessária à redundância de fonte de alimentação;
5. Na próxima tela, o usuário deverá configurar a quantidade de módulos que usará no projeto, sendo assim o Wizard criará os objetos destes módulos dentro do projeto automaticamente;
6. Então, o usuário deverá selecionar o perfil para o projeto, e a linguagem padrão para as POU's (programas). Nesse caso o novo projeto está exemplificado perfil RTU sem redundância e linguagem ST. Clicar em Next para prosseguir ou Cancel para interromper a criação do projeto;
7. A próxima tela define a linguagem da POU criada pelo perfil selecionado. Como o projeto criado não é redundante, existem apenas duas POU's (UserPrg) e (StartPrg) a linguagem ST foi mantida. Clicar em Anterior para voltar à tela antecedente, Finish para finalizar ou Cancel para interromper;
8. Ao pressionar o botão Concluir, o MasterTool Xtorm iniciará a criação do ambiente de desenvolvimento do projeto. Esse procedimento pode levar alguns segundos.

### Resolução do Estudo Dirigido 3-1

```

VAR
CRReceive: CommandReceiver; // Instância do interceptador
CCommand: COMMAND_T; // Estrutura com os dados do comando
CRResult: COMMAND_RESULT; // Resultado do comando interceptado
byResult: BYTE; // Resultado da função Trip/Close
byPulseTimeON: BYTE; // Tempo do pulso ligado
byPulseTimeOFF: BYTE; // Tempo do pulso desligado
dbpDoublePointIO: DBP; // Variável mapeada no módulo de saída
dbpDoublePointServer: DBP; // Variável mapeada no driver servidor
END_VAR

CRReceive.dwVariableAddr:= ADR(dbpDoublePointServer);
CRReceive.bExec:= TRUE;
CRReceive.dwTimeout:= 1000;

// Caso um comando seja capturado:
IF CRReceive.bCommandAvailable THEN
// Salva os dados do comando
CCommand:= CRReceive.sCommand;
// Envio do comando para o cartão pela função PulsedCommand
IF CCommand.sOperateParameters.sValue.sDoublePoint.bValue THEN
// Captura o valor configurado para o tempo do pulso
byPulseTimeON:=
DWORD_TO_BYTE(CCommand.sOperateParameters.sValue.sDoublePoint.sPulseCo
nfig.dwOnDuration / 10);
// Executa a função de PulsedCommand para o módulo de saída
byResult:= PulsedCommand(bCommandType:= 101, bRackNumber:= 0,
bSlotNumber:= 4, bPairIndex:= 0, bPulseTime:= byPulseTimeON);

```

```

ELSE
// Captura o valor configurado para o tempo do pulso
byPulseTimeOFF:=
DWORD_TO_BYTE(CCommand.sOperateParameters.sValue.sDoublePoint.sPulseCo
nfig.dwOffDuration / 10);
// Executa a função de PulsedCommand para o módulo de saída
byResult:= PulsedCommand(bCommandType:= 102, bRackNumber:= 0,
bSlotNumber:= 4, bPairIndex:= 0, bPulseTime:= byPulseTimeOFF);
END_IF
// Trata o resultado da função de comando pulsado e gera a resposta
para o comando interceptado
CASE byResult OF
1: // Tipo inválido de comando
CRResult:= COMMAND_RESULT.NOT_SUPPORTED;
CRReceive.eCommandResult:= CRResult;
CRReceive.bDone:= TRUE;
2, 3: // Parâmetros de entrada inválidos
CRResult:= COMMAND_RESULT.INCONSISTENT_PARAMETERS;
CRReceive.eCommandResult:= CRResult;
CRReceive.bDone:= TRUE;
4: // Módulo não respondeu ao comando
CRResult:= COMMAND_RESULT.TIME_LIMIT_OVER;
CRReceive.eCommandResult:= CRResult;
CRReceive.bDone:= TRUE;
6: // Outro comando foi enviado para este ponto e está em execução
CRResult:= COMMAND_RESULT.BLOCKED_BY_COMAND;
CRReceive.eCommandResult:= CRResult;
CRReceive.bDone:= TRUE;
7, 5: // Comando finalizado com sucesso
CRResult:= COMMAND_RESULT.SUCCESS;
CRReceive.eCommandResult:= CRResult;
CRReceive.bDone:= TRUE;
END_CASE
END_IF

CRReceive();

IF CRReceive.bDone THEN
CRReceive.bDone:= FALSE;
END_IF

```

### Resolução do Estudo Dirigido 3-2

```

PROGRAM UserPrg
VAR
xStatus : RTC_CMD_STATUS;
DateAndTime : EXTENDED_DATE_AND_TIME;
xRead : BOOL;
END_VAR
-----
IF (xRead = TRUE) THEN
xStatus := GetDateAndTime(DateAndTime);
xRead := FALSE;
END_IFPROGRAM UserPrg
VAR
DayOfWeek : DAYS_OF_WEEK;
xRead : BOOL;

```

```

END_VAR
-----
IF (xRead = TRUE) THEN
DayOfWeek := GetDayOfWeek();
xRead := FALSE;
END_IFPROGRAM UserPrg
VAR
RTCStatus : RTC_CMD_STATUS;
TimeZone : TIMEZONESETTINGS;
xRead : BOOL;
END_VAR
-----

//FB GetTimeZone
IF (xRead = TRUE) THEN
GetTimeZone (TimeZone);
xRead := FALSE;
END_IFPROGRAM UserPrg
VAR
SetDateAndTime : SetDateAndTime;
xRequest : BOOL;
DateAndTime : EXTENDED_DATE_AND_TIME;
xDone : BOOL;
xExec : BOOL;
xError : BOOL;
RTCStatus : RTC_CMD_STATUS;
xWrite : BOOL;
END_VAR
-----

IF (xWrite = TRUE) THEN
SetDateAndTime(
Request := xRequest,
DateAndTime := DateAndTime,
Done => xDone,
Exec => xExec,
Error => xError,
Status => RTCStatus);
xWrite := FALSE;
END_IFPROGRAM UserPrg
VAR
RTCStatus : RTC_CMD_STATUS;
TimeZone : TIMEZONESETTINGS;
xWrite : BOOL;
END_VAR
-----

//FB SetTimeZone
IF (xWrite = TRUE) THEN
RTCStatus := SetTimeZone (TimeZone);
xWrite := FALSE;
END_IF

```

### Resolução do Estudo Dirigido 3-3

```

PROGRAM MainPrg
VAR
Info: TYPE_RESULT;
byRackNumber: BYTE;
bySlotNumber: BYTE;
END_VAR

```

```

//ENTRADAS:
byRackNumber := 0;
bySlotNumber := 10;
//FUNÇÃO:
Info:= REFRESH_INPUT (byRackNumber, bySlotNumber); //Chamada da
função.
//Variável 'Info' recebe possíveis erros da função.PROGRAM UserPrq
VAR
Info: TYPE_RESULT;
byRackNumber: BYTE;
bySlotNumber: BYTE;
END_VAR
//ENTRADAS:
byRackNumber := 0;
bySlotNumber := 10;
//FUNÇÃO:
//Chamada da função.
Info := REFRESH_OUTPUT (byRackNumber, bySlotNumber);
//Variável 'Info' recebe possíveis erros da função.PROGRAM UserPrq
VAR RETAIN
bStart : BOOL := TRUE;
TOF_RET : TOF_RET;
END_VAR

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

// Executa ações ao final da contagem
IF (TOF_RET.Q = FALSE) THEN
bStart := TRUE;
END_IFPROGRAM UserPrq
VAR RETAIN
bStart : BOOL;
TON_RET : TON_RET;
END_VAR

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

// Executa ações ao final da contagem
IF (TON_RET.Q = TRUE) THEN
bStart := FALSE;
END_IFPROGRAM UserPrq
VAR RETAIN
bStart : BOOL;
TP_RET : TP_RET;
END_VAR

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

bStart := FALSE;

// Ações durante a contagem

```

```

IF (TP_RET.Q = TRUE) THEN
// Executa enquanto o contador estiver ativado
ELSE
// Executa somente quando o contador estiver desativado
END_IFPROGRAM NonSkippedProg
VAR
bStart : BOOL := TRUE;
TOF_NR : TOF_NR;
END_VAR

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

// Executa ações ao final da contagem
IF (TOF_NR.Q = FALSE) THEN
bStart := TRUE;
END_IFPROGRAM NonSkippedProg
VAR
bStart : BOOL;
TON_NR : TON_NR;
END_VAR

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

// Executa ações ao final da contagem
IF (TON_NR.Q = TRUE) THEN
bStart := FALSE;
END_IFPROGRAM NonSkippedProg
VAR
bStart : BOOL;
TP_NR : TP_NR;
END_VAR

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

bStart := FALSE;

// Ações durante a contagem
IF (TP_NR.Q = TRUE) THEN
// Executa enquanto o contador estiver ativado
ELSE
// Executa somente quando o contador estiver desativado
END_IFPROGRAM MainPrg
VAR
eLogError : USER_LOG_ERROR_CODES;
sMessage : USER_LOG_MESSAGE;
END_VAR

IF (m_rTemperature > MAX_TEMPERATURE_ACCEPT) THEN
sMessage := 'Temperatura acima do esperado: ';
sMessage := concat(sMessage, REAL_TO_STRING(m_rTemperature));
sMessage := concat(sMessage, '°');

```

```
eLogError := UserLogAdd(USER_LOG_EVENT_WARN, sMessage);  
//Variável 'eLogError' recebe possíveis erros da função.  
END_IF
```

Exemplo de conteúdo do arquivo de log: (UserLog-201308271506245738.csv)

Model; HX3040

Serial number; 445627

Firmware version; 1.1.0.12

27/08/2013 15:06:24.5738; WARN; Temperatura acima do esperado: 25°

27/08/2013 16:37:45.3476; WARN; Temperatura acima do esperado: 25°

28/08/2013 09:10:55.4201; WARN; Temperatura acima do esperado: 26°

```
PROGRAM MainPrg  
VAR  
eLogError : USER_LOG_ERROR_CODES;  
END_VAR  
  
IF (m_DeleteLogs = TRUE) THEN  
eLogError := UserLogDeleteAll();m_DeleteLogs := FALSE;  
//Variável 'eLogError' recebe possíveis erros da função.  
END_IFPROGRAM StartPrg  
VAR  
Resultado : TYPE_RESULT;  
END_VAR  
  
Resultado := ClearRtuDiagnostic(eDiagnostic := 0);PROGRAM UserPrg  
VAR  
ClearEventQueueStatus : Type_Result;  
END_VAR  
  
ClearEventQueueStatus := ClearEventQueue();
```

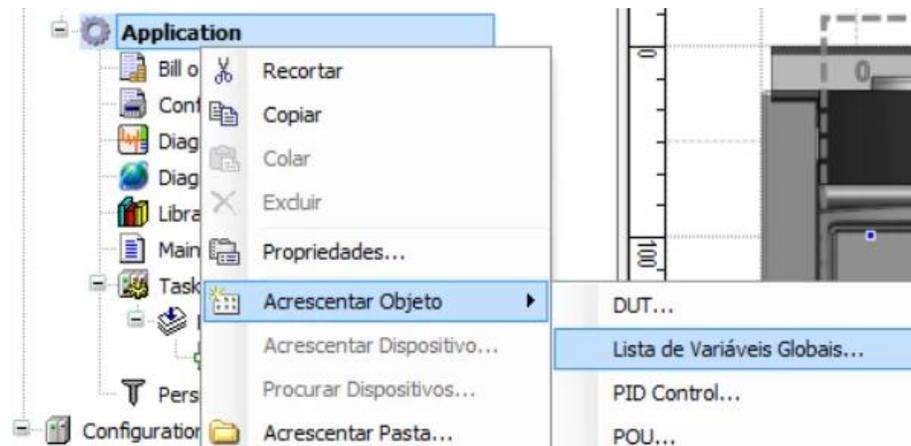
### Resolução do Estudo Dirigido 5-1

Consulte o Instrutor para acesso ao projeto MasterTool Xtorm "ProjetCompExpl.project", que implementa a aplicação proposta.

### Resolução do Estudo Dirigido 5-2

#### Parte 1: variáveis retentivas

Para criar uma variável simbólica retentiva global, primeiro é necessário criar uma Lista de Variáveis Globais (GVL) clicando com o botão direito em Application, em seguida "Acrescentar Objeto > Lista de Variáveis Globais..." conforme apresentado na figura a seguir.



Uma vez criada a GVL, ela aparecerá na árvore de dispositivos.

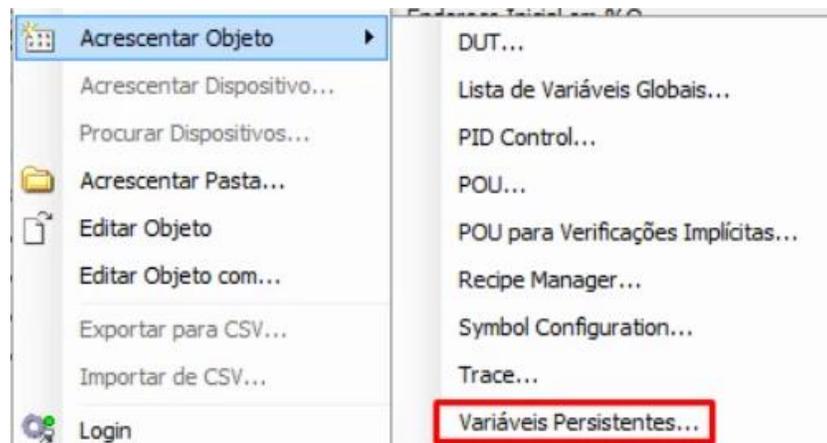


Em seguida, é necessário editar a GVL e acrescentar o termo "RETAIN" na declaração, de forma que ela fique "VAR\_GLOBAL RETAIN". Após realizado este passo, basta incluir as variáveis dentro desta declaração e elas serão retentivas e globais conforme mostra o exemplo abaixo:

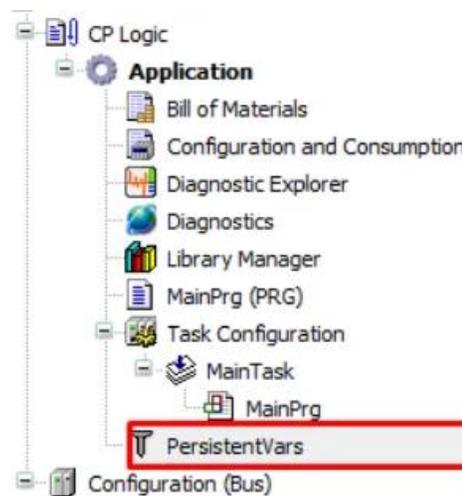
```
VAR_GLOBAL RETAIN
  VariavelRetentiva01: BYTE;
  VariavelRetentiva02: BYTE;
END_VAR
```

## Parte 2: variáveis simbólicas persistentes

Variáveis simbólicas Persistentes devem ser declaradas única e exclusivamente em uma lista especial chamada "Variáveis Persistentes", onde todas as variáveis declaradas são globais por definição. Esta lista é disponibilizada pelo objeto "Variáveis Persistentes", que é adicionado clicando com o botão direito em Application, em seguida "Acrescentar Objeto > Variáveis Persistentes...", conforme a figura a seguir.



Uma vez incluída, a lista aparecerá na árvore de dispositivos conforme a figura abaixo:



Note que, ao editar este objeto, é possível observar a declaração "VAR\_GLOBAL PERSISTENT RETAIN", confirmando que se trata de uma lista de variáveis globais. A sintaxe desta declaração não deve ser alterada pelo usuário. Para adicionar variáveis, basta incluí-las logo abaixo da declaração conforme indicado a seguir.

```
VAR_GLOBAL PERSISTENT RETAIN
  VariavelPersistente01: BYTE;
  VariavelPersistente02: BYTE;
END_VAR
```

Neste exemplo, foram declaradas duas variáveis simbólicas persistentes do tipo BYTE. Conforme citado anteriormente, estas variáveis são globais, desta forma podem ser acessadas em qualquer POU simplesmente utilizando o nome declarado, não sendo necessário indicar o seu caminho completo.

# 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

<b>Algoritmo</b>	Sequência finita de instruções bem definidas, objetivando à resolução de problemas.
<b>Árvore</b>	Estrutura de dados para configuração do hardware.
<b>Backup</b>	Cópia de segurança de dados.
<b>Barramento</b>	Conjunto de módulos de E/S interligados a uma UCP ou cabeça de rede de campo.
<b>Barramento local</b>	Conjunto de módulos de E/S interligados a uma UCP.
<b>Barramento remoto</b>	Conjunto de módulos de E/S interligados a uma cabeça de rede de campo.
<b>Bit</b>	Unidade básica de informação, podendo estar no estado 0 ou 1.
<b>Breakpoint</b>	Ponto de parada no aplicativo para depuração.
<b>Byte</b>	Unidade de informação composta por oito bits.
<b>Canal serial</b>	Interface de um equipamento que transfere dados no modo serial.
<b>Ciclado</b>	Modo de execução do CP passo-a-passo, onde cada passo é um ciclo do CP.
<b>Ciclo de varredura</b>	Uma execução completa do programa aplicativo de um controlador programável.
<b>Circuito de cão-de-guarda</b>	Circuito eletrônico destinado a verificar a integridade do funcionamento de um equipamento.
<b>Controlador programável</b>	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.
<b>CP</b>	Veja controlador programável.
<b>Default</b>	Valor predefinido para uma variável, utilizado em caso de não haver definição.
<b>Diagnóstico</b>	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.
<b>Download</b>	Carga de programa ou configuração no CP.
<b>E/S</b>	Veja entrada/saída.
<b>Entrada/saída</b>	Também chamado de E/S. Dispositivos de E/S de dados de um sistema. No caso de UTRs, correspondem tipicamente a módulos digitais ou analógicos de entrada ou saída que monitoram ou acionam o dispositivo controlado.
<b>Escravo</b>	Equipamento ligado a uma rede de comunicação que só transmite dados se for solicitado por outro equipamento denominado mestre.
<b>Frame</b>	Uma unidade de informação transmitida na rede.
<b>Gateway</b>	Equipamento ou software para a conexão de duas redes de comunicação com diferentes protocolos.
<b>Hardware</b>	Equipamentos físicos usados em processamento de dados onde normalmente são executados programas (software).
<b>Hiperlink</b>	Atalho de navegação para uma nova página do help.
<b>IEC 61131</b>	Norma genérica para operação e utilização de CPs e UTRs. Antiga IEC 1131.
<b>Interface</b>	Dispositivo que adapta elétrica e/ou logicamente a transferência de sinais entre dois equipamentos.
<b>Interrupção</b>	Evento com atendimento prioritário que temporariamente suspende a execução de um programa e desvia para uma rotina de atendimento específica
<b>kbytes</b>	Unidade representativa de quantidade de memória. Representa 1024 bytes.
<b>LED</b>	Sigla para light emitting diode. É um tipo de diodo semiconductor que emite luz quando estimulado por eletricidade. Utilizado como indicador luminoso.
<b>Linguagem de programação</b>	Um conjunto de regras e convenções utilizado para a elaboração de um programa.
<b>Local host</b>	Máquina, PC ou sistema que está em uso.
<b>Login</b>	Ação de estabelecer um canal de comunicação com o CP.
<b>Menu</b>	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.
<b>Menu de Contexto</b>	Menu dinâmico com o conteúdo de acordo com o contexto atual.
<b>Mestre</b>	Equipamento ligado a uma rede de comunicação de onde se originam solicitações de comandos para outros equipamentos da rede.
<b>Módulo (referindo-se a hardware)</b>	Elemento básico de um sistema completo que possui funções bem definidas. Normalmente é ligado ao sistema por conectores, podendo ser facilmente substituído.
<b>Módulo de E/S</b>	Módulo pertencente ao subsistema de entradas e saídas.
<b>Multicast</b>	Disseminação simultânea de informação a um determinado grupo de nós interligados a uma rede de comunicação.

---

<b>Nó</b>	Qualquer estação de uma rede com capacidade de comunicação utilizando um protocolo estabelecido.
<b>Operandos</b>	Elementos sobre os quais as instruções atuam. Podem representar constantes, variáveis ou um conjunto de variáveis.
<b>PC</b>	Sigla para programmable controller. É a abreviatura de controlador programável em inglês.
<b>PDO</b>	Do inglês, é uma mensagem do protocolo CAN contendo os dados operacionais.
<b>Ping</b>	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.
<b>POU</b>	, ou Unidade de Organização de Programa, é uma subdivisão do programa aplicativo que pode ser escrito em qualquer uma das linguagens disponíveis.
<b>Programa aplicativo</b>	É o programa carregado em um CP, que determina o funcionamento de uma máquina ou processo.
<b>Protocolo</b>	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.
<b>RAM</b>	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.
<b>Rede de comunicação</b>	Conjunto de equipamentos (nós) interconectados por canais de comunicação.
<b>Reset</b>	Comando para reinicializar o CP.
<b>RUN</b>	Comando para colocar o CP em modo de execução.
<b>Set</b>	Ação para atribuir o estado de nível lógico alto para uma variável booleana.
<b>Software</b>	Programas de computador, procedimentos e regras relacionadas à operação de um sistema de processamento de dados.
<b>STOP</b>	Comando para congelar o CP em seu estado atual.
<b>Sub-rede</b>	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.
<b>Time-out</b>	Tempo preestabelecido máximo para que uma comunicação seja completada. Se for excedido procedimentos de retentiva ou diagnóstico serão ativados.
<b>Tooltip</b>	Caixa de texto com uma ajuda ou local onde pode-se entrar com a ajuda.
<b>UCP</b>	Sigla para unidade central de processamento. Controla o fluxo de informações, interpreta e executa as instruções do programa e monitora os dispositivos do sistema.
<b>UCP ativa</b>	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.
<b>Upload</b>	Leitura do programa ou configuração do CP.
<b>Visualização</b>	Conjunto de telas do CP.
<b>XML</b>	Do inglês, Extensible Markup Language, é um padrão para gerar linguagens de marcação.
<b>Zoom</b>	No contexto da janela de função do teclado, é utilizado para a troca de telas.