



Manual de Utilização Hadron Xtorm

MU223000 Rev. G

11 de junho de 2024

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

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

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

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

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

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

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

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

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

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

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

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

DIREITOS AUTORAIS

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

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

NOTIFICAÇÃO DE USO DE SOFTWARE ABERTO

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

Sumário

1.	Introdução	1
1.1.	Série Hadron Xtorm	1
1.1.1.	Lista de Módulos	1
1.1.1.1.	UCPs – Unidades Centrais de Processamento	1
1.1.1.2.	Módulos de Entrada	1
1.1.1.3.	Módulos Mistos de E/S	2
1.1.1.4.	Módulos de Saída	2
1.1.1.5.	Módulos de Fonte de Alimentação	2
1.1.1.6.	Bastidores	2
1.1.1.7.	Software	2
1.1.1.8.	Acessórios	2
1.2.	Características Inovadoras	2
1.3.	Características Principais	3
1.3.0.1.	UCP	3
1.3.0.2.	Módulos	3
1.3.0.3.	Barramento de Alta Velocidade	3
1.3.0.4.	Troca a Quente	3
1.3.0.5.	Alta Disponibilidade	3
1.3.0.6.	Diagnósticos Avançados	3
1.3.0.7.	Capacidades	4
1.3.0.8.	Robustez	4
1.3.0.9.	Programação da UCP & Atualização de Firmware	4
1.4.	Características do Software	4
1.4.0.1.	HD8500 – MasterTool Xtorm	4
1.4.0.1.1.	Linguagens de Programação IEC 61131-3	5
1.4.0.1.2.	Editores para Configuração de Projeto e Configuração de Hardware	6
1.4.0.1.3.	Programação Orientada a Objeto	6
1.4.0.1.4.	Online, Depuração e Características de Comissionamento	7
1.4.0.1.5.	Simulação	7
1.4.0.1.6.	Documentação do Usuário & Arquivos de Ajuda	7
1.4.0.1.7.	Diagnósticos Avançados	8
1.4.0.1.8.	Docking View	8
1.4.0.1.9.	Integração dos Logical Nodes IEC 61850 com a Linguagem IEC 61131-3	8
1.4.1.	Exemplos de Aplicação	8
1.4.1.1.	UCP com E/S Local	9
1.4.1.2.	UCP com E/S Remoto (Expansão de Barramento)	9
1.4.1.3.	UCP com E/S Remoto (Expansão de Barramento com Loopback)	10

1.4.1.4.	UCP com E/S Remoto com Alta Disponibilidade (Redundância de Expansão de Barramento com Loopback)	11
1.4.1.5.	Redundância de UCP e Módulo Fonte de Alimentação	12
1.4.1.6.	Sincronismo de Tempo via IRIG-B	13
1.4.1.7.	Redes Ethernet Redundantes com NIC Teaming	15
1.4.1.8.	Redes Ethernet em Modo Anel	16
1.4.1.9.	Compatibilidade com Demais Produtos	17
1.5.	Documentos Relacionados a este Manual	18
1.6.	Suporte Técnico	19
1.7.	Mensagens de Advertência Utilizadas neste Manual	19
2.	Descrição Técnica	20
2.1.	Painéis e Conexões	20
2.2.	Características Gerais	21
2.2.1.	Características Comuns	21
2.2.2.	Características Gerais Específicas	23
2.2.3.	Interfaces Seriais	25
2.2.3.1.	COM 1	25
2.2.3.2.	COM 2	25
2.2.4.	Interfaces Ethernet	26
2.2.4.1.	NET 1 .. NET 6	26
2.2.5.	IRIG-B	26
2.2.6.	Visor Gráfico	27
2.2.7.	Interface do Cartão de Memória	27
2.3.	Compatibilidade com Demais Produtos	27
2.4.	Desempenho	28
2.4.1.	Tempos de Aplicação	28
2.4.2.	Tempos para Execução de Instruções	28
2.4.3.	Tempos de Inicialização	29
2.5.	Produtos Relacionados	29
3.	Instalação	31
3.1.	Inspeção Visual	31
3.2.	Instalação Mecânica	31
3.2.1.	Fixação do Bastidor	31
3.2.1.1.	Furação para Bastidor 9 Posições	31
3.2.1.2.	Furação para Bastidor 18 Posições	32
3.2.1.3.	Montagem	32
3.2.1.4.	Remoção	33
3.2.2.	Inserção dos Módulos	34
3.2.3.	Remoção dos Módulos	35
3.2.3.1.	Módulos de E/S	36
3.2.3.2.	Bornes de E/S	37
3.2.3.3.	Identificação	37
3.2.3.4.	Diagrama de Instalação	37
3.2.3.5.	Tampa de Conector de Bastidor	37
3.2.3.6.	Inserção da Tampa de Conector de Bastidor	37
3.2.3.7.	Remoção da Tampa de Conector de Bastidor	37
3.3.	Instalação Elétrica	38
3.3.1.	Segurança Elétrica	38

3.3.2.	Bornes com Mola	39
3.3.2.1.	Inserção da Fiação	40
3.3.2.2.	Borne de 4 Vias	40
3.3.2.3.	Borne de 6 Vias	41
3.3.2.4.	Borne de 10 Vias	41
3.3.2.5.	Montagem da Fiação	41
3.3.2.6.	Fixação da Fiação	41
3.3.2.7.	Remoção da Fiação	42
3.3.3.	Conexões	42
3.3.4.	Alimentações	42
3.3.5.	Fusíveis	42
3.4.	Instalação Elétrica UCP	43
3.5.	Conexão com a Rede Ethernet	44
3.5.1.	Endereço IP	44
3.5.2.	ARP Gratuito	45
3.5.3.	Instalação do Cabo de Rede	45
3.6.	Conexão com a Rede Serial (COM 1)	46
3.6.1.	Comunicação RS-232C	47
3.6.2.	Comunicação RS-485 sem Terminação	47
3.6.3.	Comunicação RS-485 com Terminação Interna	48
3.6.4.	Comunicação RS-485 com Terminação Externa	48
3.7.	Conexão com a Rede Serial (COM 2)	49
3.7.1.	Comunicação RS-485 sem Terminação	50
3.7.2.	Comunicação RS-485 com Terminação Interna	51
3.7.3.	Comunicação RS-485 com Terminação Externa	52
3.7.4.	Comunicação RS-422 sem Terminação	53
3.7.5.	Comunicação RS-422 com Terminação Interna	54
3.7.6.	Comunicação RS-422 com Terminação Externa	55
3.7.7.	Exemplo de Rede RS-422	56
3.8.	Conexão com Sinal IRIG-B	57
3.8.1.	Terminais de Ligação do IRIG-B	58
3.9.	Instalação do Cartão de Memória	59
3.10.	Instalação do Programador	60
4.	Configuração	65
4.1.	Configuração da UCP	65
4.1.1.	Parâmetros Gerais	65
4.1.1.1.	Troca a Quente	67
4.1.1.1.1.	Troca a Quente Desabilitada, Apenas para Módulos Declarados	67
4.1.1.1.2.	Troca a Quente Habilitada, com Consistência na Partida Apenas para Módulos Declarados	67
4.1.1.1.3.	Troca a Quente Desabilitada, sem Consistência na Partida	68
4.1.1.1.4.	Como Realizar a Troca a Quente	68
4.1.1.2.	Cartão de Memória	69
4.1.2.	Sincronização de Tempo	70
4.1.2.1.	IRIG-B	72
4.1.2.2.	Control Center	72
4.1.2.3.	SNTP	73
4.1.2.4.	PTP	74

4.1.2.5.	Horário de Verão	74
4.1.3.	Pontos Internos	74
4.1.3.1.	Conversões de Qualidade	76
4.1.3.1.1.	Internal Quality	77
4.1.3.1.2.	Conversão DNP3	78
4.1.3.1.3.	Conversão IEC 60870-5-104	79
4.1.3.1.4.	Conversão IEC 61850	82
4.1.3.1.5.	Qualidade Interna MODBUS	83
4.1.3.1.6.	Qualidade dos Módulos de E/S	83
4.1.4.	Conversão de Engenharia	85
4.1.5.	Alarmes	85
4.1.6.	Agrupamento de Eventos	86
4.1.6.1.	Configuração dos Agrupamentos de Eventos	87
4.1.6.2.	Utilização da Configuração de Atraso Máximo de um Agrupamento	88
4.1.6.3.	Cálculo do Atributo Valor do Evento Resumo de um Agrupamento	88
4.1.6.4.	Cálculo do Atributo Qualidade do Evento Resumo de um Agrupamento	89
4.1.7.	Roteamento de Eventos	90
4.2.	Configuração das Interfaces Seriais	90
4.2.1.	COM 1	90
4.2.1.1.	Configurações Avançadas	92
4.2.2.	COM 2	93
4.2.2.1.	Configurações Avançadas	94
4.3.	Configuração das Interfaces Ethernet	95
4.3.1.	Interfaces Ethernet Locais	95
4.3.1.1.	NET 1 .. NET 6	95
4.3.2.	Portas TCP Reservadas	96
4.3.3.	Configurações Avançadas das Interfaces Ethernet	96
4.4.	Pontos Duplos	97
4.5.	Configuração de Protocolos	97
4.5.1.	Comportamento dos Protocolos x Estado da UCP	100
4.5.2.	Fila de Eventos da UCP	100
4.5.2.1.	Consumers	101
4.5.2.2.	Princípios de Funcionamento da Fila	101
4.5.2.2.1.	Sinalização de Overflow	102
4.5.2.3.	Producers	102
4.5.3.	Interceptação de Comandos Oriundos do Centro de Controle	102
4.5.4.	MODBUS - Tipos de Dados	108
4.5.5.	MODBUS RTU Mestre	108
4.5.5.1.	Parâmetros Gerais do Protocolo MODBUS Mestre	109
4.5.5.2.	Configuração dos Dispositivos	112
4.5.5.3.	Configuração dos Mapeamentos	113
4.5.5.4.	Configuração das Requisições	115
4.5.6.	MODBUS RTU Escravo	119
4.5.6.1.	Parâmetros Gerais do Protocolo MODBUS Escravo	120
4.5.6.2.	Configuração dos Mapeamentos	123
4.5.7.	MODBUS Ethernet	124
4.5.8.	MODBUS Ethernet Cliente	125
4.5.8.1.	Parâmetros Gerais do Protocolo MODBUS Cliente	126

4.5.8.1.1.	Diagnósticos MODBUS Cliente	127
4.5.8.2.	Configuração dos Dispositivos	128
4.5.8.2.1.	Time-out de Comunicação	129
4.5.8.3.	Configuração dos Mapeamentos	130
4.5.8.4.	Configuração das Requisições	131
4.5.8.4.1.	Diagnósticos MODBUS Cliente	134
4.5.8.5.	Disparo de Relações MODBUS Cliente de Forma Acíclica	135
4.5.9.	MODBUS Ethernet Servidor	135
4.5.9.1.	Parâmetros Gerais do Protocolo MODBUS Servidor	136
4.5.9.1.1.	Diagnósticos MODBUS Servidor	138
4.5.9.2.	Configuração dos Mapeamentos	140
4.5.10.	DNP3 - Tipos de Dados	141
4.5.11.	DNP3 Ethernet Cliente	142
4.5.11.1.	Configuração dos Mapeamentos DNP3 Cliente	143
4.5.11.2.	Configuração da Camada de Enlace DNP3 Cliente	145
4.5.11.3.	Configuração da Camada de Aplicação DNP3 Cliente	147
4.5.11.4.	Requisições Customizadas DNP3 Cliente	148
4.5.11.4.1.	Diagnósticos DNP3 Cliente	153
4.5.11.5.	Comandos DNP3 para IEDs	156
4.5.11.5.1.	Comandos para Saídas Digitais	156
4.5.11.5.2.	Comandos para Saídas Analógicas	157
4.5.11.5.3.	Comando Cold Restart	159
4.5.11.6.	Códigos de Status para Comandos DNP3	161
4.5.12.	DNP3 Ethernet Servidor	161
4.5.12.1.	Configuração dos Mapeamentos DNP3 Servidor	162
4.5.12.2.	Configuração da Camada de Enlace DNP3 Servidor	165
4.5.12.3.	Configuração da Camada de Aplicação DNP3 Servidor	167
4.5.12.4.	Configuração das Mensagens Não Solicitadas DNP3 Servidor	169
4.5.12.5.	Diagnósticos DNP3 Servidor	170
4.5.12.6.	Associação dos Objetos com o Protocolo IEC 60870-5-104	172
4.5.12.7.	Comando Cold Restart	172
4.5.12.8.	Comandos para Pontos de Saída e Contadores	172
4.5.12.8.1.	Pontos Internos	173
4.5.12.8.2.	Módulo de Saída	173
4.5.12.8.3.	Driver DNP3 Cliente	173
4.5.12.8.4.	Driver IEC 60870-5-104 Cliente	173
4.5.12.8.5.	Interceptação de Comandos	173
4.5.13.	IEC 60870-5-104 - Tipos de Dados	174
4.5.14.	IEC 60870-5-104 Cliente	175
4.5.14.1.	Configuração da Camada de Enlace de uma Controlled Station	176
4.5.14.2.	Configuração dos Mapeamentos de um Setor IEC 60870-5-104	177
4.5.14.3.	Configuração da Camada de Enlace de um Setor IEC 60870-5-104	179
4.5.14.4.	Configuração da Camada de Aplicação um Setor IEC 60870-5-104	179
4.5.14.5.	Diagnósticos do IEC 60870-5-104 Cliente	180
4.5.14.6.	Comandos IEC 60870-5-104 para IEDs	181
4.5.14.6.1.	Comandos para Saídas Digitais	182
4.5.14.6.2.	Comandos para Saídas Analógicas	183
4.5.14.6.3.	Comando Reset Process	185

4.5.14.6.4.	Comando Synchronize IED	186
4.5.14.6.5.	Comando General Interrogation	186
4.5.14.6.6.	Comando Counter Interrogation	187
4.5.14.6.7.	Exemplo de Utilização de um Bloco Funcional para Comando IEC 60870-5-104	188
4.5.14.6.8.	Códigos de Erro para Comandos IEC 60870-5-104	189
4.5.15.	IEC 60870-5-104 Servidor	189
4.5.15.1.	Parâmetros Gerais	190
4.5.15.2.	Mapeamentos de Dados	190
4.5.15.3.	Camada de Enlace	192
4.5.15.4.	Camada de Aplicação	193
4.5.15.5.	Diagnósticos do Servidor	196
4.5.15.6.	Associação dos Objetos com o Protocolo DNP3	197
4.5.15.7.	Comandos para Pontos de Saída	198
4.5.15.7.1.	Pontos Internos	198
4.5.15.7.2.	Módulo de Saída	198
4.5.15.7.3.	Driver DNP3 Cliente	198
4.5.15.7.4.	Driver IEC 60870-5-104 Cliente	198
4.5.15.7.5.	Interceptação de Comandos	198
4.5.15.7.6.	Qualificador de Comandos	199
4.5.15.7.7.	Interceptação do Comando	199
4.5.16.	IEC 61850 Servidor	199
4.5.16.1.	Implementação do Modelo de Dados IEC 61850	200
4.5.16.2.	Configuração da Base de Dados	204
4.5.16.3.	Configuração dos Datasets	207
4.5.16.4.	Configuração dos Reports	207
4.5.16.5.	Configuração GOOSE	210
4.5.16.6.	Configuração do Mapeamento de Variáveis IEC 61850	213
4.5.16.7.	Diagnósticos IEC 61850 Servidor	214
4.5.16.8.	Interceptação de Comandos	216
4.6.	Desempenho da Comunicação	217
4.6.1.	Comunicação MODBUS	217
4.6.1.1.	Servidor MODBUS	218
4.6.2.	Comunicação DNP3	218
4.6.2.1.	DNP3 Servidor	218
4.6.2.2.	DNP3 Cliente	218
4.6.3.	Comunicação IEC 60870-5-104	218
4.6.3.1.	IEC 60870-5-104 Servidor	218
4.6.3.2.	IEC 60870-5-104 Cliente	219
4.7.	Desempenho do Sistema	219
4.7.1.	Escaneamento de E/S	219
4.7.2.	Cartão de Memória	220
4.8.	Relógio RTC	220
4.8.1.	Blocos Funcionais e Funções para Leitura e Escrita do RTC	221
4.8.1.1.	Funções de Leitura do RTC	221
4.8.1.1.1.	GetDateAndTime	221
4.8.1.1.2.	GetDayOfWeek	222
4.8.1.1.3.	GetTimeZone	223

4.8.1.2.	Blocos Funcionais e Funções de Escrita e Configuração do RTC	223
4.8.1.2.1.	SetDateAndTime	223
4.8.1.2.2.	SetTimeZone	225
4.8.2.	Estruturas de Dados do RTC	226
4.8.2.1.	EXTENDED_DATE_AND_TIME	226
4.8.2.2.	DAYS_OF_WEEK	226
4.8.2.3.	RTC_CMD_STATUS	227
4.8.2.4.	TIME_ZONE_SETTINGS	227
4.9.	Memória de Arquivos de Usuário	228
4.10.	Cartão de Memória	230
4.10.1.	Acesso no MasterTool	231
4.11.	Menu Informativo e de Configuração da UCP	232
4.12.	Blocos Funcionais e Funções	235
4.12.1.	Atualização de Entradas e Saídas	235
4.12.1.1.	REFRESH_INPUT	235
4.12.1.2.	REFRESH_OUTPUT	237
4.12.2.	Timer Retentivo	238
4.12.2.1.	TOF_RET	239
4.12.2.2.	TON_RET	240
4.12.2.3.	TP_RET	241
4.12.3.	Timer Não-Redundante	243
4.12.3.1.	TOF_NR	243
4.12.3.2.	TON_NR	243
4.12.3.3.	TP_NR	244
4.12.4.	Log de Usuário	245
4.12.4.1.	UserLogAdd	245
4.12.4.2.	UserLogDeleteAll	247
4.12.5.	ClearRtuDiagnostic	248
4.12.6.	ClearEventQueue	249
4.13.	Gerenciamento de Usuários e Direitos de Acesso	249
4.13.1.	Gerenciamento de Usuários e Direitos de Acesso do Projeto	249
4.13.1.1.	Gerenciamento de Usuários	249
4.13.1.1.1.	Usuários	250
4.13.1.1.2.	Grupos	252
4.13.1.1.3.	Configurações	253
4.13.1.2.	Gerenciamento de Direitos de Acesso	253
4.13.2.	Gerenciamento de Usuário e Direitos de Acesso da UCP	255
4.13.2.1.	Usuários e Grupos	255
4.13.2.1.1.	Comum	256
4.13.2.1.2.	Usando o Diálogo de Configuração	256
4.13.2.1.3.	Usando o Diálogo de Configuração - Usuários	256
4.13.2.1.4.	Usando o Diálogo de Configuração - Grupos	257
4.13.2.1.5.	Aplicando e Armazenando a Configuração Atual	258
4.13.2.1.6.	Considerações sobre Usuário e Grupos Padrão	258
4.13.2.1.7.	Considerações sobre Usuário e Grupos Padrão - Usuários e Grupos Padrão	258
4.13.2.1.8.	Considerações sobre Usuário e Grupos Padrão - Usuários e Grupos Padrão - Grupo Administrador	258

4.13.2.1.9.	Considerações sobre Usuário e Grupos Padrão - Usuários e Grupos Padrão - Grupo Developer	258
4.13.2.1.10.	Considerações sobre Usuário e Grupos Padrão - Usuários e Grupos Padrão - Grupo Everyone	258
4.13.2.1.11.	Considerações sobre Usuário e Grupos Padrão - Usuários e Grupos Padrão - Grupo Service	258
4.13.2.1.12.	Considerações sobre Usuário e Grupos Padrão - Usuários e Grupos Padrão - Grupo Watch	259
4.13.2.1.13.	Considerações sobre Usuário e Grupos Padrão - Usuários e Grupos Padrão - Usuário Administrador	259
4.13.2.1.14.	Considerações sobre Usuário e Grupos Padrão - Usuários e Grupos Padrão - Usuário Everyone	259
4.13.2.1.15.	Usuário e Grupos de Projetos Antigos	259
4.13.2.2.	Direitos de Acesso	259
4.13.2.2.1.	Definindo as Permissões de Acesso	260
4.13.2.2.2.	Definindo as Permissões de Acesso - Ações	260
4.13.2.2.3.	Definindo as Permissões de Acesso - Permissões	261
4.13.2.2.4.	Aplicando e Armazenando a Configuração Atual	261
4.13.2.2.5.	Direitos de Acesso de Projetos Antigos	261
4.14.	Acesso a Página Web de Gerenciamento	261
4.14.1.	Página de Sistema	262
4.14.1.1.	Ajuste do Relógio	262
4.14.1.1.1.	Horário do Computador (UTC)	263
4.14.1.1.2.	Horário Personalizado (UTC)	263
4.14.2.	Página de Rede	263
4.14.2.1.	Configurações de Rede	264
4.14.2.1.1.	Modo Definido pela Aplicação	264
4.14.2.1.2.	Modo Definido pela Página Web	265
4.14.2.2.	Sniffer de rede	266
4.15.	Servidor FTP	267
4.15.1.	Configuração	268
4.15.1.1.	Configurações Gerais	268
4.15.1.1.1.	Habilitar Servidor	268
4.15.1.1.2.	Habilitar Comunicação Criptografada (FTPS)	268
4.15.1.1.3.	Acesso Somente Leitura	269
4.15.1.1.4.	Tempo Limite Ocioso (Segundos)	269
4.15.1.2.	Configurações de Usuário	269
4.15.1.2.1.	Nome de Usuário	269
4.15.1.2.2.	Senha	269
4.15.1.3.	Status	269
4.15.1.3.1.	Estado Atual	269
4.15.1.3.2.	Conexões Ativas	269
5.	Programação Inicial	270
5.1.	Organização e Acesso à Memória	270
5.2.	Perfis de Projeto	272
5.2.1.	Perfil para UTR	272
5.2.2.	Perfil Personalizado	273
5.3.	Novo Projeto	274
5.4.	Acrescentando Módulos	277

5.5.	Criando POUs	278
5.6.	Criando Tarefas	279
5.6.1.	Configurando as Tarefas	280
5.6.2.	Vínculo de POUs com Tarefas	282
5.6.3.	Número Máximo de Tarefas	282
5.7.	Configurando a UCP	283
5.8.	Bibliotecas	285
5.9.	Inserindo uma Instância de Protocolo	285
5.9.1.	MODBUS RTU	286
5.9.2.	MODBUS Ethernet	287
5.9.3.	DNP3 Servidor	288
5.9.4.	IEC 60870-5-104 Servidor	289
5.10.	Localizando a Rede	290
5.11.	Compilando um Projeto	291
5.12.	Login	292
5.13.	Modo Run	293
5.14.	Modo Stop	294
5.15.	Monitoração, Escrita e Forçamento de Variáveis	294
5.16.	Variáveis Utilizadas em mais de uma Origem	295
5.17.	Logout	296
5.18.	Modo Simulação	296
5.19.	Upload do Projeto	298
5.20.	Estados de Operação da UCP	299
5.20.1.	Run	299
5.20.2.	Stop	299
5.20.3.	Breakpoint	299
5.20.4.	Exception	299
5.20.5.	Reset a Quente	299
5.20.6.	Reset a Frio	299
5.20.7.	Reset Origem	300
5.20.8.	Reset Process Command (IEC 60870-5-104)	300
5.21.	Programas (POUs) e Listas de Variáveis Globais (GVLs)	300
5.21.1.	Programa MainPrg	300
5.21.2.	Programa StartPrg	301
5.21.3.	Programa EngineeringPrg	301
5.21.4.	Programa AlarmPrg	301
5.21.5.	Programa UserPrg	301
5.21.6.	Programa ProtPrg	301
5.21.7.	Programa UserProtPrg	301
5.21.8.	GVL Disables	301
5.21.9.	GVL IOQualities	302
5.21.10.	GVL Module_Diagnostics	302
5.21.11.	GVL Qualities	303
5.21.12.	GVL ReqDiagnostics	305
5.21.13.	GVL System_Diagnostics	306
6.	Redundância com UCP HX3040	308
6.1.	Introdução	308
6.2.	Configurações para uma UCP Redundante	308

6.2.1.	Identificação de uma UCP HX3040	309
6.2.2.	Características Gerais	309
6.3.	Princípios de Funcionamento	312
6.3.1.	Projeto Redundante Único	312
6.3.2.	Estrutura do Projeto Redundante	312
6.3.2.1.	Redundancy Template	312
6.3.2.2.	Tarefas Cíclicas: MainTask e ProtTask	313
6.3.2.3.	Programa MainPrg	313
6.3.2.4.	Programa UserPrg	313
6.3.2.5.	Programa NonSkippedPrg	314
6.3.2.6.	Programa ProtPrg	314
6.3.2.7.	Programa UserProtPrg	314
6.3.2.8.	Programa NonSkippedProtPrg	314
6.3.2.9.	Variáveis Redundantes e Não-Redundantes	315
6.3.2.10.	Variáveis %I Redundantes	315
6.3.2.11.	Variáveis %Q Redundantes	315
6.3.2.12.	Variáveis Simbólicas Redundantes e Não-Redundantes	315
6.3.3.	Mapeamentos Múltiplos	316
6.3.4.	Estruturas de Dados de Diagnósticos, Comandos e Usuário	316
6.3.5.	Serviços de Sincronização Cíclicos através de Canais de Sincronismo de Redundância	317
6.3.5.1.	Troca de Diagnósticos e Comandos	317
6.3.5.2.	Sincronização de Dados Redundantes	317
6.3.6.	Serviços de Sincronização Esporádicos Através dos Canais de Sincronismo de Redundância	318
6.3.6.1.	Sincronização de Projetos	318
6.3.7.	Redes Ethernet Redundantes com NIC Teaming	318
6.3.8.	Métodos de Troca de IP	319
6.3.8.1.	IP Ativo	319
6.3.9.	Uso Combinado de NIC Teaming e IP Ativo	320
6.3.10.	Estados de uma UCP Redundante	320
6.3.10.1.	Estado Não-Configurado	320
6.3.10.2.	Estado Inicializando	321
6.3.10.3.	Estado Ativo	321
6.3.10.4.	Estado Reserva	321
6.3.10.5.	Estado Inativo	321
6.3.11.	Comandos do menu de redundância do visor da UCP	322
6.3.12.	Transições entre Estados de Redundância	322
6.3.12.1.	Transição 1 - Não-Configurado para Inicializando	323
6.3.12.2.	Transição 2 - Não-Configurado para Inativo	323
6.3.12.3.	Transição 3 - Inicializando para Não-Configurado	323
6.3.12.4.	Transição 4 - Inicializando para Inativo	323
6.3.12.5.	Transição 5 - Inicializando para Ativo	323
6.3.12.6.	Transição 6 - Inicializando para Reserva	324
6.3.12.7.	Transição 7 - Inativo para Não-Configurado	324
6.3.12.8.	Transição 8 - Ativo para Não-Configurado	324
6.3.12.9.	Transição 9 - Ativo para Inativo	324
6.3.12.10.	Transição 10 - Ativo para Reserva	324
6.3.12.11.	Transição 11 - Reserva para Não-Configurado	324

6.3.12.12.	Transição 12 – Reserva para Inativo	324
6.3.12.13.	Transição 13 – Reserva para Ativo	324
6.3.13.	Primeiros Instantes em Estado Ativo	325
6.3.14.	Falhas mais Comuns Causadoras de Switchovers Automáticos entre UCPs	325
6.3.15.	Falhas Associadas à Switchovers entre UCPs Gerenciados pelo Usuário	325
6.3.16.	Tolerância a Falhas	326
6.3.16.1.	Falhas Simples com Indisponibilidade	327
6.3.16.2.	Falhas Simples sem Indisponibilidade Causando um Switchover	327
6.3.16.3.	Falhas Simples sem Indisponibilidade	327
6.3.17.	Overhead da Redundância	327
6.4.	Programação de uma UCP Redundante	328
6.4.1.	Wizard para Criação de um Novo Projeto Redundante	328
6.4.2.	Configuração do Projeto com Redundância de UCP	331
6.4.2.1.	Configuração Fixa nas Posições 0 a 3 do Bastidor	332
6.4.3.	Configurações das Portas Ethernet da UCP HX3040 (NET 1 até NET 6)	332
6.4.3.1.	Configuração do Endereço de IP	332
6.4.4.	Configurações de I/O Drivers	332
6.4.5.	Configurações da MainTask	333
6.4.5.1.	Programa StartPrg	333
6.4.5.2.	Programa UserPrg	333
6.4.5.3.	Programa NonSkippedPrg	334
6.4.6.	Configurações da ProtTask	334
6.4.6.1.	Programa UserProtPrg	334
6.4.6.2.	Programa NonSkippedProtPrg	334
6.4.7.	GVLs com Variáveis Simbólicas Redundantes	334
6.4.8.	POUs do Tipo Programa com Variáveis Simbólicas Redundantes	334
6.4.9.	Utilização de Breakpoints em Sistemas Redundantes	335
6.4.10.	Limitações na Programação de uma UCP Redundante	335
6.4.10.1.	Limitações em GVLs e POU's Redundantes	335
6.4.10.2.	Limitações no Programa Não-Redundante (StartPrg, NonSkippedPrg e NonSkippedProtPrg)	335
6.4.11.	Obtendo o Estado da Redundância de uma UCP	335
6.4.12.	Leitura de Diagnósticos Não-Redundantes	336
6.5.	Carga de Programas em uma UCP Redundante	336
6.5.1.	Carga Inicial de um Projeto Redundante	336
6.5.1.1.	Primeiro Passo - Descoberta do Endereço IP para Conexão do MasterTool	336
6.5.1.2.	Segundo Passo – Verificar Rede e IP do Computador para Programação	336
6.5.1.3.	Terceiro Passo – Verificar Conflito de Endereços IP	337
6.5.1.4.	Quarto Passo – Preparar Conexão do MasterTool (Definir Caminho Ativo)	337
6.5.1.5.	Quinto Passo – Carga do Projeto Redundante	337
6.5.2.	Conexão do MasterTool com uma UCP HX3040 de uma UCP Redundante	338
6.5.3.	Carga de Modificações em um Projeto Redundante	338
6.5.4.	Carga de Modificações Off-Line e On-Line	338
6.5.4.1.	Modificações que Demandam Carga Off-Line com Interrupção do Controle do Processo	339
6.5.4.2.	Modificações que Permitem Carga On-Line	339
6.5.5.	Carga On-Line de Modificações	339
6.5.6.	Carga Off-Line de Modificações com Interrupção do Controle do Processo	340

6.6.	Manutenção de UTRs Redundantes	340
6.6.1.	Mensagens de Advertência do MasterTool	340
6.6.1.1.	Bloqueio Antes de Comandos que Podem Parar a UCP Ativa	340
6.6.1.2.	Bloqueios de Operações na UCP Não-Ativa	340
6.6.2.	Interação com a Redundância Através do Visor Gráfico da UCP HX3040	340
6.6.2.1.	Estado de Redundância da UCP	341
6.6.2.2.	Telas Abaixo do Menu Redundância	341
6.6.3.	Estrutura de Diagnósticos da Redundância	341
6.6.3.1.	Diagnósticos da Redundância	341
6.6.3.2.	Comandos da Redundância	349
6.6.3.3.	Informações do Usuário Trocados entre UCPA e UCPB	350
6.6.4.	Logs de Eventos da Redundância	350
7.	Manutenção	352
7.1.	Diagnósticos do Módulo UCP	352
7.1.1.	One Touch Diag	352
7.1.2.	Diagnósticos via LED	355
7.1.3.	Diagnósticos via WEB	356
7.1.4.	Diagnósticos via Variáveis	359
7.1.4.1.	Diagnósticos Resumidos	359
7.1.4.2.	Diagnósticos Detalhados	362
7.1.5.	Diagnósticos via Blocos Funcionais	373
7.1.5.1.	GetTaskInfo	373
7.1.6.	Visor Gráfico	375
7.2.	Log de Sistema	377
7.3.	Diagnósticos dos Módulos de E/S	378
7.3.1.	Acesso ao modo de diagnóstico	378
7.3.2.	Acessando pontos de E/S	379
7.3.3.	Acessando a descrição do módulo e dos pontos de E/S	379
7.3.4.	Pressionamento curto e pressionamento longo	380
7.4.	Não Carregar a Aplicação na Inicialização	380
7.5.	Falha na Alimentação	380
7.6.	Problemas mais Comuns	381
7.7.	Solução de Problemas	382
7.8.	Manutenção Preventiva	382
8.	Projeto do Painel Elétrico	383
8.1.	Projeto Mecânico	383
8.1.1.	Dimensões	383
8.1.1.1.	Bastidores	383
8.1.1.1.1.	Bastidor de 9 Posições	383
8.1.1.1.2.	Bastidor de 18 Posições	383
8.1.1.2.	Módulos	384
8.1.2.	Profundidade do Módulo Montado no Bastidor	384
8.1.3.	Espaçamento entre módulos e outros equipamentos do painel	385
8.1.4.	Dimensionamento da Calha	386
8.1.5.	Montagem Horizontal/ Vertical	386
8.2.	Projeto Térmico	386
8.2.1.	Dissipação de calor em um painel elétrico	386
8.3.	Projeto Elétrico	389

8.3.1.	Informações Gerais	389
8.3.2.	Alimentação do Armário	389
8.3.3.	Distribuição dos Cabos no Armário	389
8.3.4.	Iluminação do Armário	390
8.3.5.	Aterramento	390
8.3.6.	Interferência Eletromagnética	390
8.3.7.	Blindagem	390
8.3.8.	Supressores de Ruído	390
8.3.8.1.	Circuito RC	390
8.3.8.2.	Circuito com Diodo	391
8.3.8.3.	Circuito com Diodo e Zener	391
8.3.8.4.	Circuito com Varistor	392
8.3.8.5.	Circuito com Capacitor	392
8.3.9.	Distribuição das Alimentações fora do Armário	393
8.3.10.	Proteção contra Raios	393

1. Introdução

1.1. Série Hadron Xtorm

A automação de sistemas de energia elétrica é caracterizada pelo uso de equipamentos e dispositivos robustos, confiáveis e que apresentam alta tecnologia com a capacidade de operar em ambientes hostis, onde há presença de níveis significativos de interferência eletromagnética e exposição a temperaturas de operação mais elevadas. Esta é a realidade de aplicações em usinas hidrelétricas (UHÉs), subestações de energia elétrica, parques eólicos, entre outras.

Neste contexto, a Série Hadron Xtorm se apresenta como uma inovadora Unidade Terminal Remota (UTR), perfeita para aplicações em geração, transmissão e distribuição de energia elétrica. A Série possui um conjunto ideal de recursos com alto desempenho e facilidades para as diversas etapas no ciclo de vida de uma aplicação, visando redução de custos de engenharia, instalação e comissionamento e a minimização de tempos de indisponibilidade, e manutenção do sistema quando em operação. Com interfaces intuitivas e amigáveis, diagnósticos precisos e inteligentes, um design moderno e robusto, além de diversas características inovadoras, a Hadron Xtorm supera os requisitos de aplicações deste mercado.

A Série possui uma arquitetura inteligente e versátil, oferecendo modularidade em pontos de entrada e saída (E/S), opções em redundância, troca a quente de módulos, protocolos de comunicação de alta velocidade, como IEC 61850 e DNP3, implementação de lógica em conformidade com a norma IEC 61131-3 e sincronismo de tempo.



Figura 1: Série Hadron Xtorm – Visão Geral

1.1.1. Lista de Módulos

Segue abaixo a lista completa dos módulos. Para mais informações, por favor, consulte a documentação de produto de cada módulo.

1.1.1.1. UCPs – Unidades Centrais de Processamento

- **HX3040:** UCP de alta velocidade, 6 portas Ethernet, 2 canais seriais, interface para cartão de memória, suporte a expansão de barramento e suporte a redundância

1.1.1.2. Módulos de Entrada

- **HX1100:** Módulo 32 ED 24 Vdc c/ Registro de Eventos
- **HX1120:** Módulo 32 ED 125 Vdc c/ Registro de Eventos
- **HX6000:** Módulo 16 EA Tensão/Corrente
- **HX6020:** Módulo 8 EA Temperatura (RTD)

1.1.1.3. Módulos Mistos de E/S

- **HX6065:** Módulo Misto Medição AC / 4S Tensão/Corrente

1.1.1.4. Módulos de Saída

- **HX2200:** Módulo 16 SD Relé
- **HX2300:** Módulo 16 SD Relé 24 Vdc c/ CBO
- **HX2320:** Módulo 16 SD Relé 125 Vdc c/ CBO

1.1.1.5. Módulos de Fonte de Alimentação

- **HX8300:** Fonte de Alimentação Redundante 60 W 24 Vdc
- **HX8320:** Fonte de Alimentação Redundante 60 W 125 Vdc

1.1.1.6. Bastidores

- **HX9001:** Bastidor de 9 Posições
- **HX9003:** Bastidor de 18 Posições

1.1.1.7. Software

- **HD8500/ADV:** MasterTool Xtorm

1.1.1.8. Acessórios

- **HX9102:** Tampa de Conector de Bastidor
- **HX9405:** Conector 04 terminais
- **HX9401:** Conector 06 terminais
- **HX9402:** Conector 10 terminais
- **NX9202:** Cabo RJ45-RJ45 2 m
- **NX9205:** Cabo RJ45-RJ45 5 m
- **NX9210:** Cabo RJ45-RJ45 10 m
- **NX9101:** Cartão de 32 GB microSD com adaptador para miniSD e SD

1.2. 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, subestações e demais aplicações deste segmento. A lista abaixo mostra algumas destas características que o usuário encontrará na Série Hadron Xtorm:



Battery Free Operation: 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 disto, esta característica é ambientalmente correta.



Multiple Block Storage: 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.



One Touch Diag: Esta é uma característica exclusiva dos CPs 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.

OFD – On Board Full Documentation: 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 a informação completa fica armazenada em um único e seguro local.

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

1.3. Características Principais

1.3.0.1. UCP

A UCP apresenta várias funções integradas, programação online, alta capacidade de memória, seis portas Ethernet, dois canais seriais, uma porta de entrada e uma porta de saída para o sinal de sincronismo de tempo. As seis portas Ethernet são disponibilizadas para configuração e programação, uso em redes IEC 61850, DNP3, IEC 60870-5-104, MODBUS TCP e servidor web embarcado. Além disso, as UCPs contam com duas interfaces seriais para conexão de IHMs locais e uso em redes MODBUS RTU, um slot de cartão de memória para armazenamento de código-fonte da aplicação e atualização de programa, uma porta para recebimento do sinal de sincronismo de tempo IRIG-B e outra porta de saída deste sinal, utilizada para que a UCP possa realizar o sincronismo de tempo de outros equipamentos.

1.3.0.2. 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.

1.3.0.3. 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

1.3.0.4. 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.

1.3.0.5. 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.

1.3.0.6. 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

1.3.0.7. 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 consome 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 e o bastidor principal, é igual a 100.

1.3.0.8. Robustez

O design da Série Hadron Xtorm é extremamente robusto e permite o uso em aplicações com ambientes hostis. Capaz de ser instalada em ambientes 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 comportas. Finalmente, possui requisitos elevados de imunidade a descargas eletroestáticas e ruídos eletromagnéticos comumente presentes nestas aplicações. Seu design oferece estas possibilidades, sem comprometimento dos procedimentos de instalação e manutenção.

1.3.0.9. Programação da UCP & 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

1.4. Características do Software

1.4.0.1. HD8500 – MasterTool Xtorm

O MasterTool Xtorm é o software para configuração, programação, simulação, monitoração e depuração da Série Hadron Xtorm. Baseado no conceito de ferramenta integrada, o MasterTool Xtorm proporciona flexibilidade e facilidade de uso, permitindo aos usuários a importação de dados de planilhas eletrônicas para parametrização de módulos ou mapeamento de variáveis nos protocolos de comunicação disponíveis na UCP.

Dentre os protocolos e serviços integrados, podemos citar o MODBUS RTU, MODBUS TCP, DNP3, IEC 60870-5-104, IEC 61850 (Servidor MMS e GOOSE), sincronismo de tempo, além do agrupamento de eventos que pode ser realizado de forma gráfica.

O MasterTool Xtorm oferece também todos os editores definidos na norma IEC 61131-3 para desenvolvimento da aplicação: Texto Estruturado (ST), Sequenciamento Gráfico de Funções (SFC), Diagrama de Blocos Funcionais (FBD), Diagrama Ladder (LD), Lista de Instruções (IL) e Gráfico Funcional Contínuo (CFC). Todos editores foram especialmente desenvolvidos para garantir a ótima escolha para os usuários dependendo da aplicação e do seu perfil e cultura técnica em automação.

Principais Características:

- Linguagens de programação baseadas em IEC 61131-3
- Editores gráficos para configuração de projeto e de hardware
- Programação orientada a objeto
- Simulação

- Documentação de usuário e arquivos de ajuda integrados
- Diagnósticos avançados
- Visualização que utiliza o conceito de abas (tecnologia Docking View)
- Integração dos Logical Nodes (IEC 61850) com a linguagem IEC 61131-3
- Biblioteca de funções para usinas hidrelétricas e subestações

1.4.0.1.1. Linguagens de Programação IEC 61131-3

O MasterTool Xtorm oferece todos os editores definidos na norma IEC para desenvolvimento da aplicação: Texto Estruturado (ST), Sequenciamento Gráfico de Funções (SFC), Diagrama de Blocos Funcionais (FBD), Diagrama Ladder (LD), Lista de Instruções (IL) e Gráfico Funcional Contínuo (CFC). O editor permite a declaração e uso de variáveis simbólicas, sendo que as variáveis são integradas entre todos os editores da ferramenta.

Todos editores foram especialmente desenvolvidos para garantir um ótimo gerenciamento. Idéias e sugestões de usuários experientes foram incorporadas ao processo de desenvolvimento.

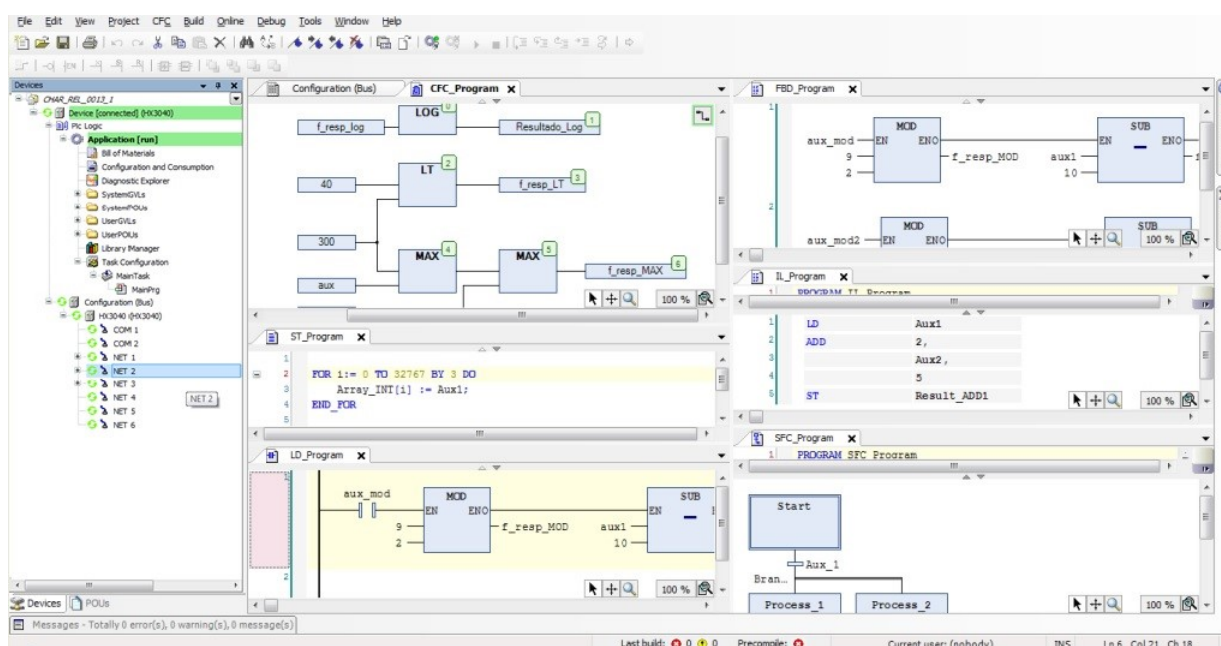


Figura 2: Linguagens de Programação IEC 61131-3

Alguns exemplos:

- Ao trabalhar em FBD, LD ou IL, o usuário pode alternar entre estes editores
- Elementos de linguagens podem ser inseridos diretamente ou arrastados para o editor a partir de uma caixa de ferramentas
- O HD8500 oferece um assistente de entrada inteligente e a funcionalidade auto completar
- Construções típicas de linguagem (declarações IF, laços FOR, classes de variáveis, etc.) podem ser escondidas nos editores textuais
- Construções de linguagem são criadas automaticamente (IF... END_IF)
- O editor SFC pode ser usado conforme definido na norma ou na versão simplificada
- Um tempo adequado de monitoramento para passos e para a funcionalidade de diagnóstico online também é disponibilizado no editor SFC

1. INTRODUÇÃO

1.4.0.1.2. Editores para Configuração de Projeto e Configuração de Hardware

Com a ajuda de editores especiais, um projeto pode ser facilmente configurado no MasterTool Xtorm. A ferramenta gráfica permite uma maneira rápida e amigável de configurar o sistema. Adicionalmente, o usuário tem a visualização completa da arquitetura da aplicação com a posição física e as informações de cada módulo.

A configuração das interfaces de rede e protocolos de comunicação padrão como MODBUS, DNP3, IEC 60870-5-104 e o protocolo IEC 61850 são integrados na ferramenta de programação. Esta característica permite que o usuário defina todos os parâmetros de configuração em um único lugar, não necessitando utilizar diferentes ferramentas de software.

Caso seja necessário, o usuário poderá utilizar o recurso de importação de dados de planilhas eletrônicas para parametrização, recuperação ou comparação de tabelas de mapeamentos de projetos, bastando para isso selecionar e arrastar as tabelas das planilhas eletrônicas para as abas de configuração dos editores.

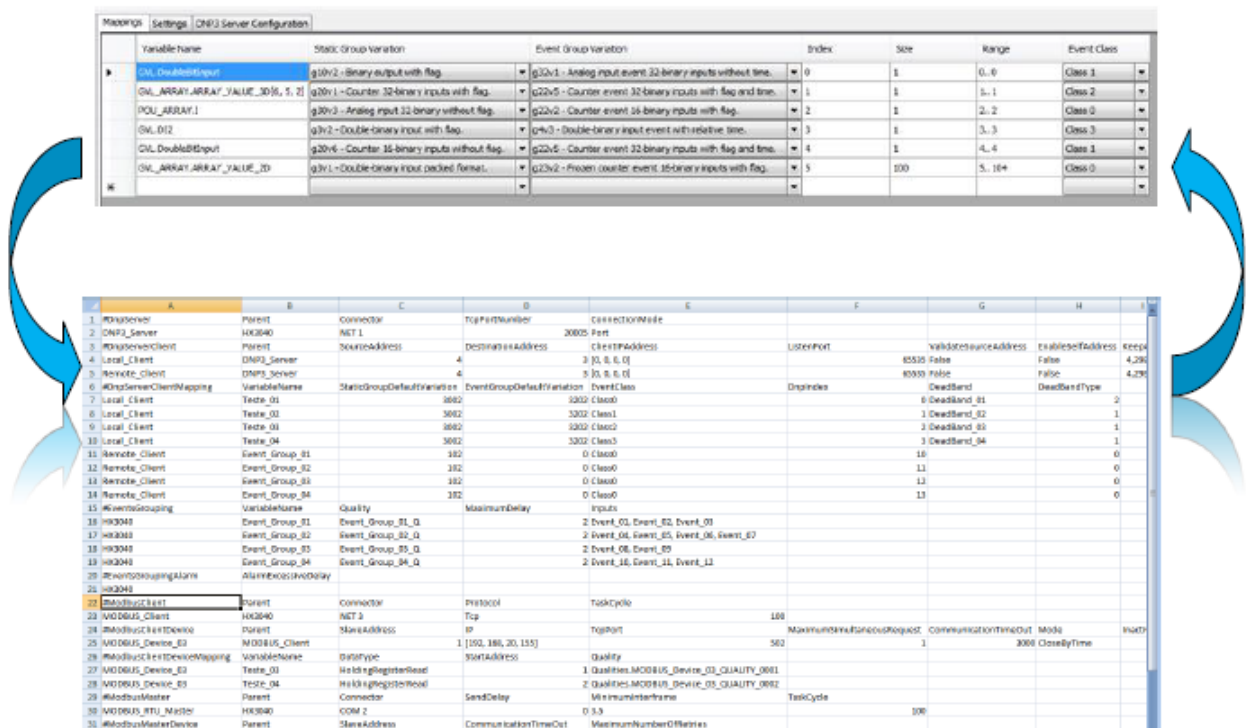


Figura 3: Editores para Configuração de Projeto e Configuração de Hardware

1.4.0.1.3. Programação Orientada a Objeto

O MasterTool Xtorm oferece uma programação orientada a objeto com as vantagens conhecidas das modernas linguagens de alto nível, como JAVA ou C++: classes, interfaces, métodos, herança, polimorfismo, etc. A programação orientada a objeto oferece grandes vantagens ao usuário, como por exemplo, quando se deseja reutilizar partes existentes de uma aplicação, ou quando se trabalha em uma aplicação com vários desenvolvedores.

1. INTRODUÇÃO

1.4.0.1.4. Online, Depuração e Características de Comissionamento

O código gerado a partir da aplicação é enviado para o dispositivo com um simples clique de mouse. A partir do momento em que o MasterTool Xtorm está online, são disponibilizadas várias funções importantes para realizar uma depuração rápida e eficiente, assim como para teste e comissionamento.

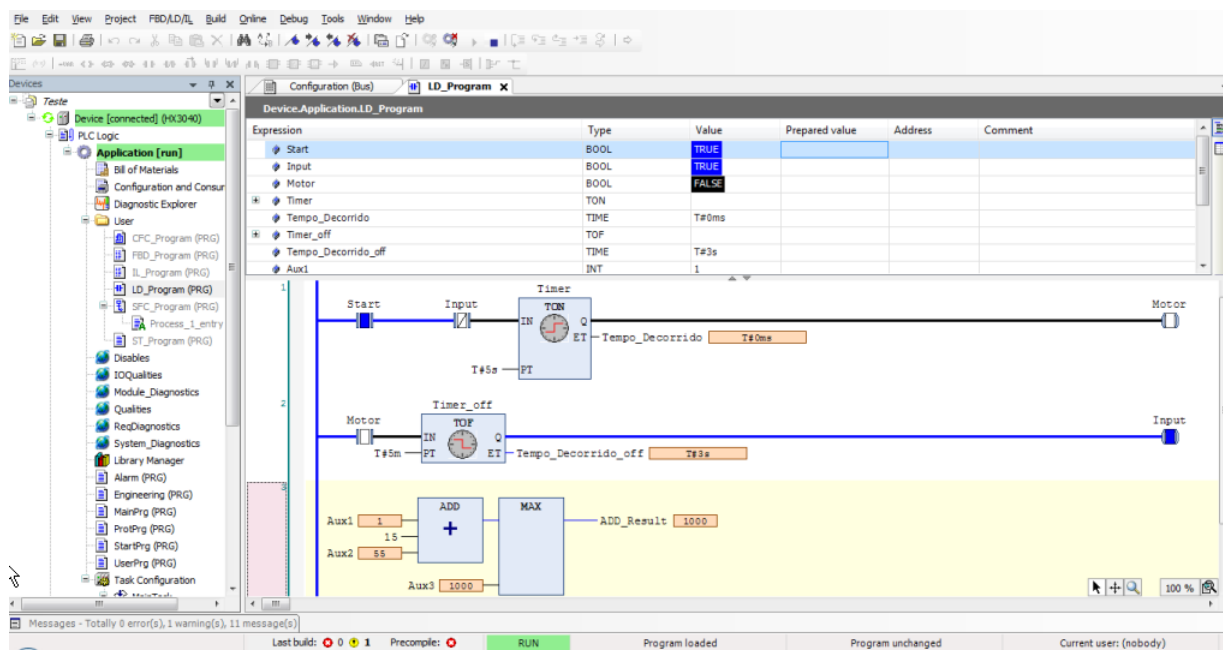


Figura 4: Online, Depuração e Características de Comissionamento

Os valores das variáveis declaradas, por exemplo, são exibidos no código do programa. Estes valores podem ser alterados ou forçados sem nenhuma dificuldade. Definindo breakpoints e percorrendo-os através da linha de código (passo a passo) os erros podem ser facilmente detectados. Os breakpoints, no MasterTool Xtorm, podem também ser atribuídos a determinadas condições para conferir mais precisão ao processo de depuração. Em uma operação de ciclo único, o usuário pode seguir a execução da aplicação através de um ciclo completo.

Se a aplicação for modificada, apenas as modificações serão compiladas, carregadas e então ativadas, sem a necessidade de interromper o controlador e correr o risco de perda dos valores das variáveis. Também são possíveis as alterações em várias POU's (Unidades de Organização de Programa), variáveis ou tipos de dados. Esta funcionalidade é chamada de Alteração Online.

O recurso de Trace é um poderoso recurso que permite ao usuário analisar dados da aplicação de forma gráfica. Possibilitando, desta forma, a fácil obtenção de informações como tendências e dinâmica temporal dos dados da aplicação.

1.4.0.1.5. Simulação

Uma característica que permite ao usuário avaliar e testar diversas lógicas e algoritmos é a ferramenta de simulação. Este recurso permite que as aplicações do usuário sejam projetadas e testadas sem a necessidade de uma UTR conectada. Isto é também interessante para treinamento, documentação e avaliação de casos de teste. Como se trata de um simulador, naturalmente podem haver algumas limitações no desenvolvimento da aplicação em comparação com a UTR real.

1.4.0.1.6. Documentação do Usuário & Arquivos de Ajuda

Considerando que a programação da Unidade Terminal Remota de acordo com as linguagens da norma IEC 61131-3 é uma tarefa complexa, o MasterTool Xtorm oferece um extenso arquivo de ajuda, com várias dicas e descrições para orientar e servir como uma primeira base de dados de resolução de problemas para o usuário, na criação de códigos de lógica ou no uso das características do software. Este arquivo de ajuda é disponibilizado em diferentes idiomas, de acordo com as opções de instalação.

O MasterTool Xtorm também oferece suporte a múltiplos idiomas, permitindo ao usuário selecionar o seu idioma preferido a partir das opções disponíveis.

1. INTRODUÇÃO

Como parte da documentação do usuário, o HD8500 pode imprimir documentos de aplicação tais como listas de materiais (BOM), POUs e parâmetros de configuração.

1.4.0.1.7. Diagnósticos Avançados

Uma das inovações da Série Hadron Xtorm é o seu amplo suporte de diagnósticos. Esta ideia surgiu das demandas de extensas e complexas aplicações, onde o correto uso de cada informação é fundamental para a manutenção, resolução e prevenção de potenciais problemas. Esta característica também está presente no MasterTool Xtorm onde o usuário, enquanto estiver conectado a uma UCP em execução, poderá acessar estruturas de diagnóstico completo através de janelas de monitoração e páginas da web.

1.4.0.1.8. Docking View

A tecnologia Docking View permite ao usuário customizar o ambiente MasterTool Xtorm, de acordo com suas necessidades pessoais. Esta característica fornece uma interface de usuário amigável, para maximizar a experiência com a ferramenta do software.

1.4.0.1.9. Integração dos Logical Nodes IEC 61850 com a Linguagem IEC 61131-3

No Mastertool Xtorm, para cada nó lógico (LN) de configuração de IED, atribuído ao projeto de usuário, o correspondente bloco de função é automaticamente declarado no ambiente de desenvolvimento das lógicas do usuário.

Maiores detalhes de configuração, descrições e funcionamento dos LNs estão descritos no Manual de Utilização da Série Hadron Xtorm. Entretanto, é recomendada a leitura da norma IEC 61850 para melhor uso das lógicas apresentadas.

1.4.1. Exemplos de Aplicação

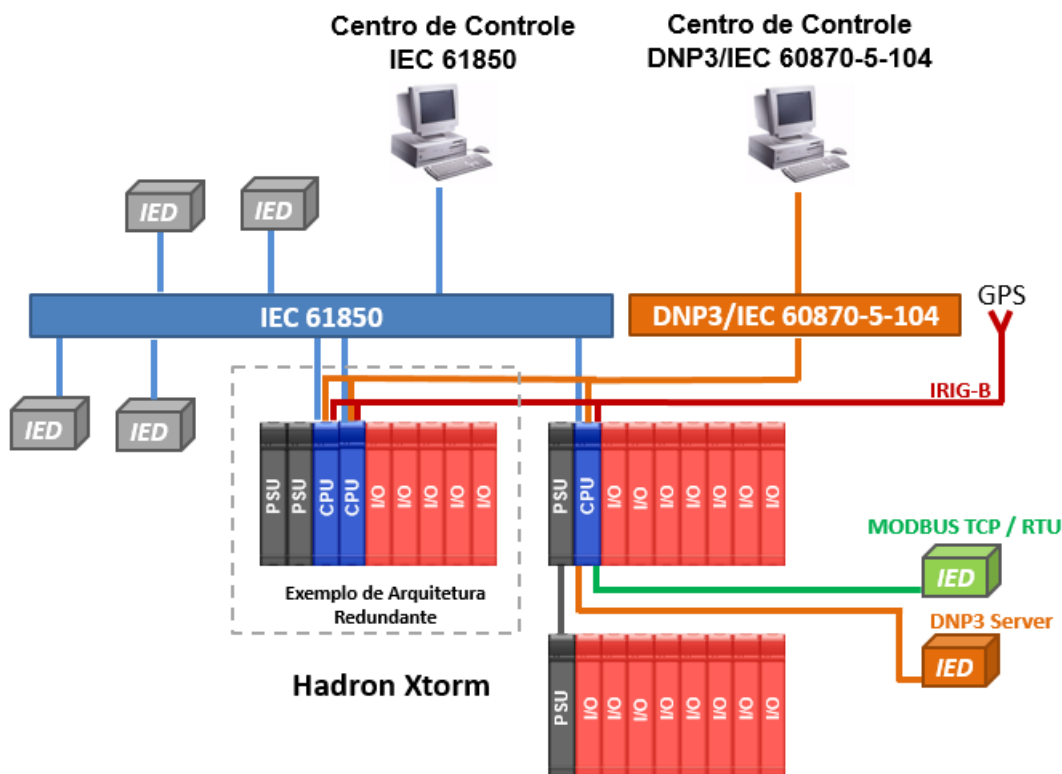


Figura 5: Exemplo de Aplicação

Abaixo podemos visualizar um exemplo de arquitetura típica utilizando a Unidade Terminal Remota (UTR) da Série Hadron Xtorm.

1.4.1.1. UCP com E/S Local

Esta arquitetura é baseada em um único bastidor, chamado bastidor local. Este bastidor é composto por uma UCP, um módulo fonte de alimentação e pelos módulos de E/S exigidos para a aplicação, conforme mostrado abaixo. A ordem dos módulos deve seguir as regras de configuração apresentadas na ferramenta de configuração.

Esta arquitetura destina-se a pequenas aplicações, onde há um baixo número de pontos de E/S.



Figura 6: UCP com E/S Local

1.4.1.2. UCP com E/S Remoto (Expansão de Barramento)

Esta arquitetura é baseada em um bastidor principal (onde está localizada a UCP) e bastidores remotos. A comunicação entre o bastidor local e os bastidores remotos é feita através das portas de expansão, localizadas no módulo HX8300 e HX8320. Cada bastidor remoto necessita de seu próprio módulo de fonte de alimentação. O módulo HX8300 e HX8320 possui duas portas RJ45, sendo uma delas usada para os dados de entrada e a outra para os dados de saída.

Neste exemplo de aplicação, apenas a porta de saída do módulo HX8300 ou HX8320 local é conectada, deixando a porta dos dados de entrada aberta. Já, no último bastidor remoto, é a porta dos dados de saída que fica aberta. Os bastidores remotos entre eles apresentam ambas as portas conectadas: uma porta conectada ao bastidor anterior e a outra ao bastidor seguinte.

Esta arquitetura destina-se a médias e grandes aplicações, onde há um alto número de pontos de E/S.

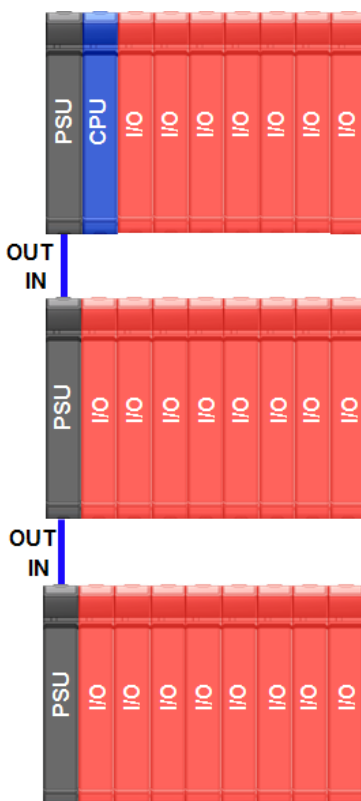


Figura 7: UCP com E/S Remoto (Expansão de Barramento)

1.4.1.3. UCP com E/S Remoto (Expansão de Barramento com Loopback)

Da mesma forma que a anterior, esta arquitetura é baseada em um bastidor local (onde está localizada a UCP) e bastidores remotos. A comunicação entre o bastidor local e os bastidores remotos é feita através das portas de expansão, localizados no módulo HX8300 ou HX8320. A única diferença em relação à arquitetura anterior, é que a porta de dados de saída do último bastidor remoto é conectada à porta de dados de entrada do bastidor principal.

Esta arquitetura permite ao sistema manter o acesso às informações dos bastidores remotos mesmo em caso de falha nos cabos de expansão. A UCP detectará a falha simples em um dos cabos e redirecionará os caminhos dos dados internos para suportar esta falha. Neste caso, também será gerado um alarme de diagnóstico ao usuário. Esta característica possui vantagens na manutenção dos cabos com o sistema energizado, além de aumentar a disponibilidade do sistema geral. Na figura abaixo podemos visualizar esta arquitetura proposta.

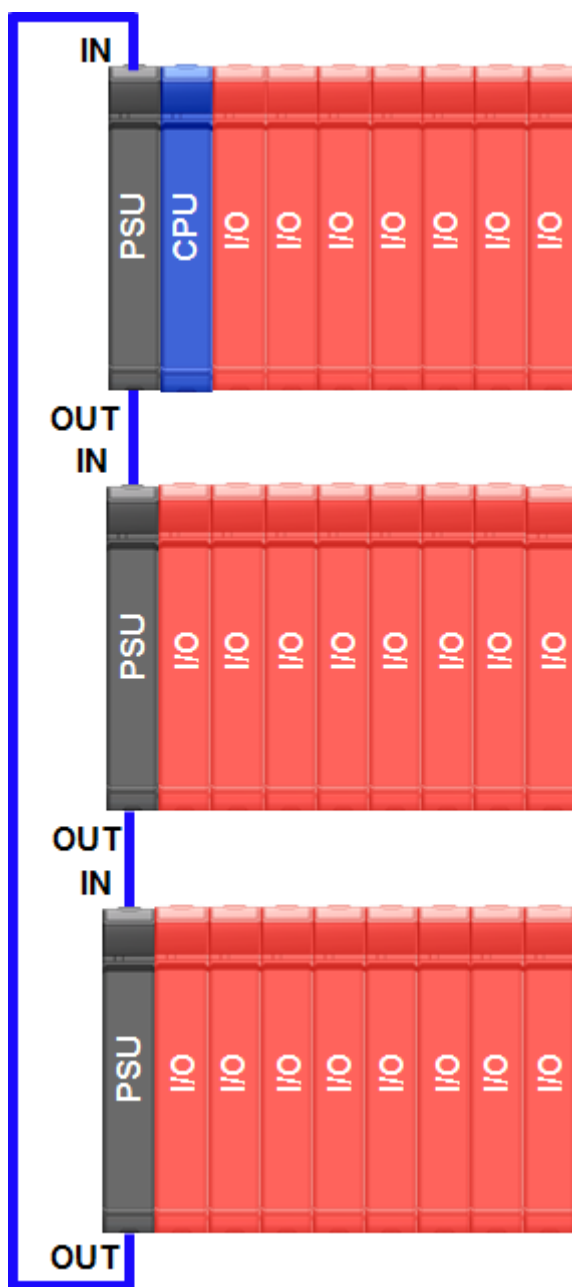


Figura 8: UCP com E/S Remoto (Expansão de Barramento com Loopback)

1.4.1.4. UCP com E/S Remoto com Alta Disponibilidade (Redundância de Expansão de Barramento com Loopback)

Esta arquitetura é baseada no uso de dois módulos HX8300 ou HX8320 por bastidor. Contando com dois módulos de expansão de barramento, o sistema apresenta uma disponibilidade elevada, pois suporta falha nos cabos de expansão de barramento ou no próprio módulo HX8300 ou HX8320.

Assim como a arquitetura anterior, esta arquitetura se destina a sistemas onde a manutenção é crítica e o sistema precisa estar disponível por longos períodos. Nesta arquitetura, os bastidores devem ser montados de acordo com o diagrama abaixo, com os módulos HX8300 ou HX8320 localizados lado a lado nas primeiras posições do bastidor.

Observe que há portas de módulos de expansão de barramento não utilizadas, as quais devem ser deixadas desconectadas.

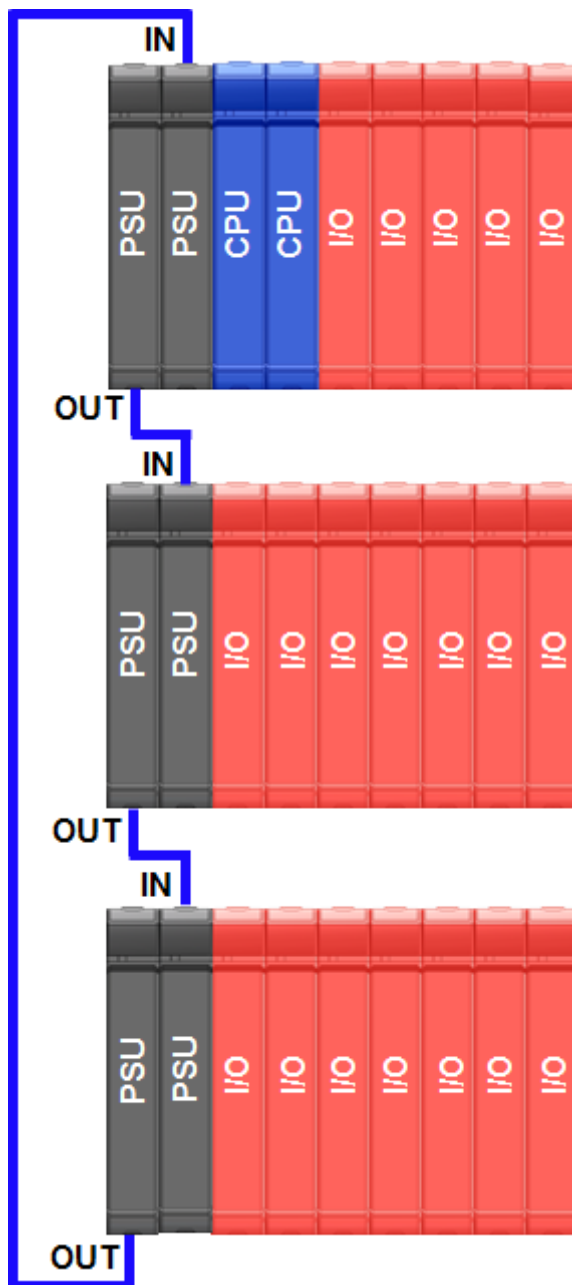


Figura 9: UCP com E/S Remoto com Alta Disponibilidade (Redundância de Expansão de Barramento com Loopback)

1.4.1.5. Redundância de UCP e Módulo Fonte de Alimentação

Para aplicações críticas, a Série Hadron Xtorm oferece redundância de UCPs e de Módulos de Fonte de Alimentação no mesmo bastidor, sendo que quando o usuário optar por utilizar redundância de UCPs em sua arquitetura, deverá utilizar também redundância do Módulo de Fonte de Alimentação.

Nesta arquitetura, o sistema terá uma UCP executando a tarefa de controle (ativo) e outra permanecendo em reserva, em uma topologia hot-standby. Em caso de falha da UCP ativa, o sistema automaticamente irá realizar um switchover (evento onde a UCP em reserva torna-se ativa). Isto significa que aplicações críticas deixam de ser afetadas por eventuais falhas do sistema de controle, garantindo elevada disponibilidade para estas aplicações. Os resultados são aumento na eficiência, produtividade, minimização do tempo de inatividade e menor tempo de manutenção.

A configuração das duas UCPs deve ser idêntica. A UCP ativa e a UCP reserva devem estar em posições do bastidor que suportem tal funcionalidade (posição 3 e 4 do bastidor). Esta aplicação é de fácil configuração e dispensa programação especial ou parametrização.

Para a redundância do Módulo de Fonte de Alimentação, se ocorrer falha em um dos módulos, automaticamente, o segundo assumirá. Em cada módulo haverá uma indicação do seu estado de funcionamento e da tensão de alimentação de entrada. Este estado pode ser usado para detectar e substituir o módulo com falha. Tal status pode ser lido pela UCP através do barramento e informado aos dispositivos de monitoração como um terminal de operação ou Interface Homem-Máquina (IHM) ou um sistema de supervisão e aquisição de dados SCADA. Um módulo em falha pode ser substituído durante a operação normal do sistema, sem necessidade de desenergização ou interrupção da aplicação.



Figura 10: Redundância de UCP e Módulo Fonte de Alimentação

1.4.1.6. Sincronismo de Tempo via 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 poderá sincronizar outros equipamentos, retransmitindo na porta de saída, o sinal recebido na porta de entrada.

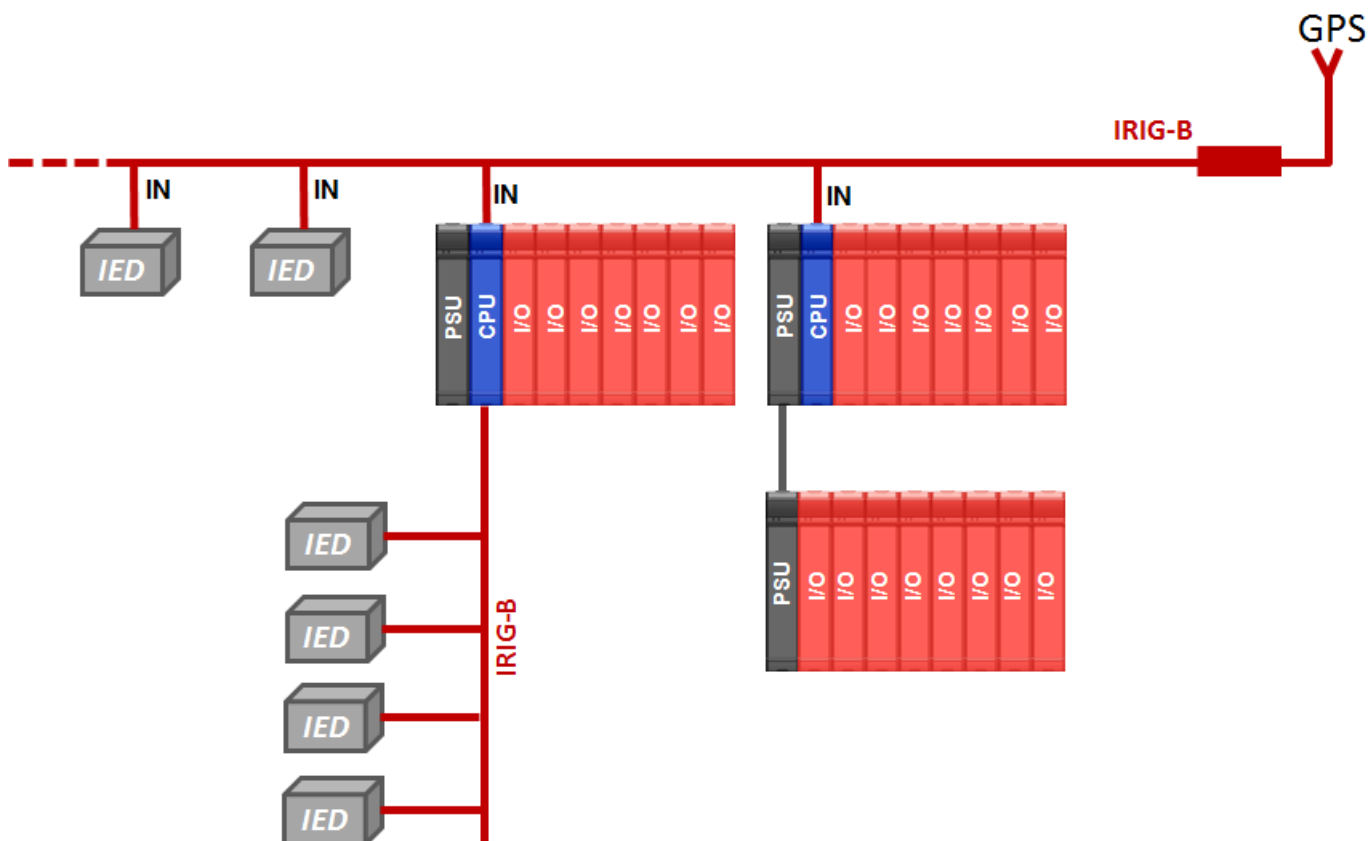


Figura 11: Sincronismo de Tempo via IRIG-B - UCP Simples

Para os casos de aplicações onde for utilizada redundância de UCP, é conveniente que o sinal de sincronismo de tempo seja interligado a 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. Na figura abaixo podemos visualizar um exemplo deste tipo de arquitetura.

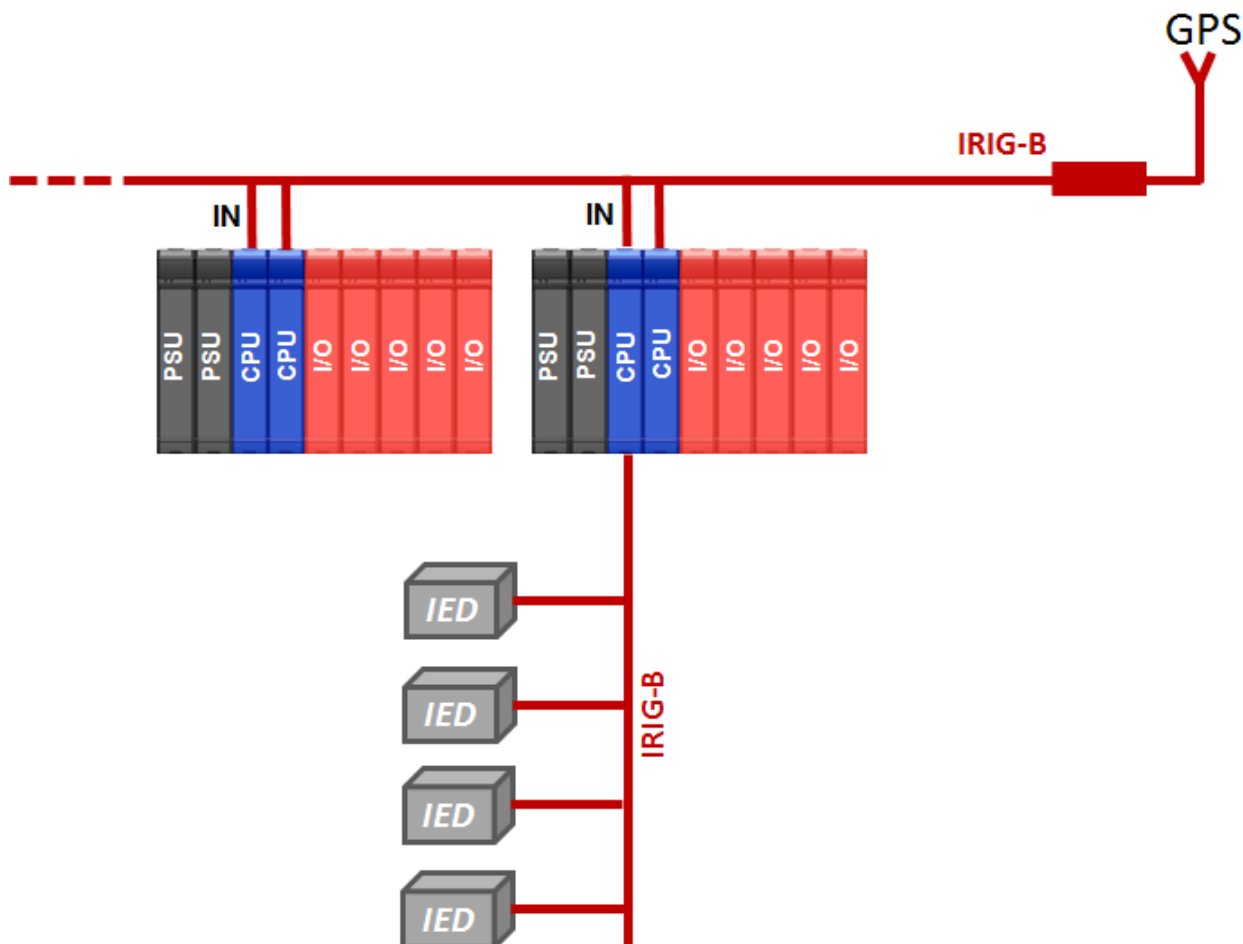


Figura 12: Sincronismo de Tempo via IRIG-B - UCP Redundante

1.4.1.7. 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 ou outros equipamentos ou dispositivos de campo. Para a redundância de redes Ethernet com NIC Teaming, é necessário que duas portas Ethernet da UCP estejam configuradas formando um par redundante. Um conjunto de duas portas Ethernet formando um par NIC Teaming possui um único endereço IP vinculado ao par de portas. Desta forma, o centro de controle não precisa se preocupar em trocar o IP caso haja falha em alguma das portas do par NIC Teaming. Cada uma das portas Ethernet deverá estar interligada a switches diferentes. Se ocorrer falha em uma das portas, automaticamente os pacotes de dados serão redirecionados para a outra porta.

Tal arquitetura Ethernet possibilita elevada disponibilidade de comunicação do sistema, sendo fortemente indicada para transpor falhas nas portas Ethernet, em cabos e em switches.

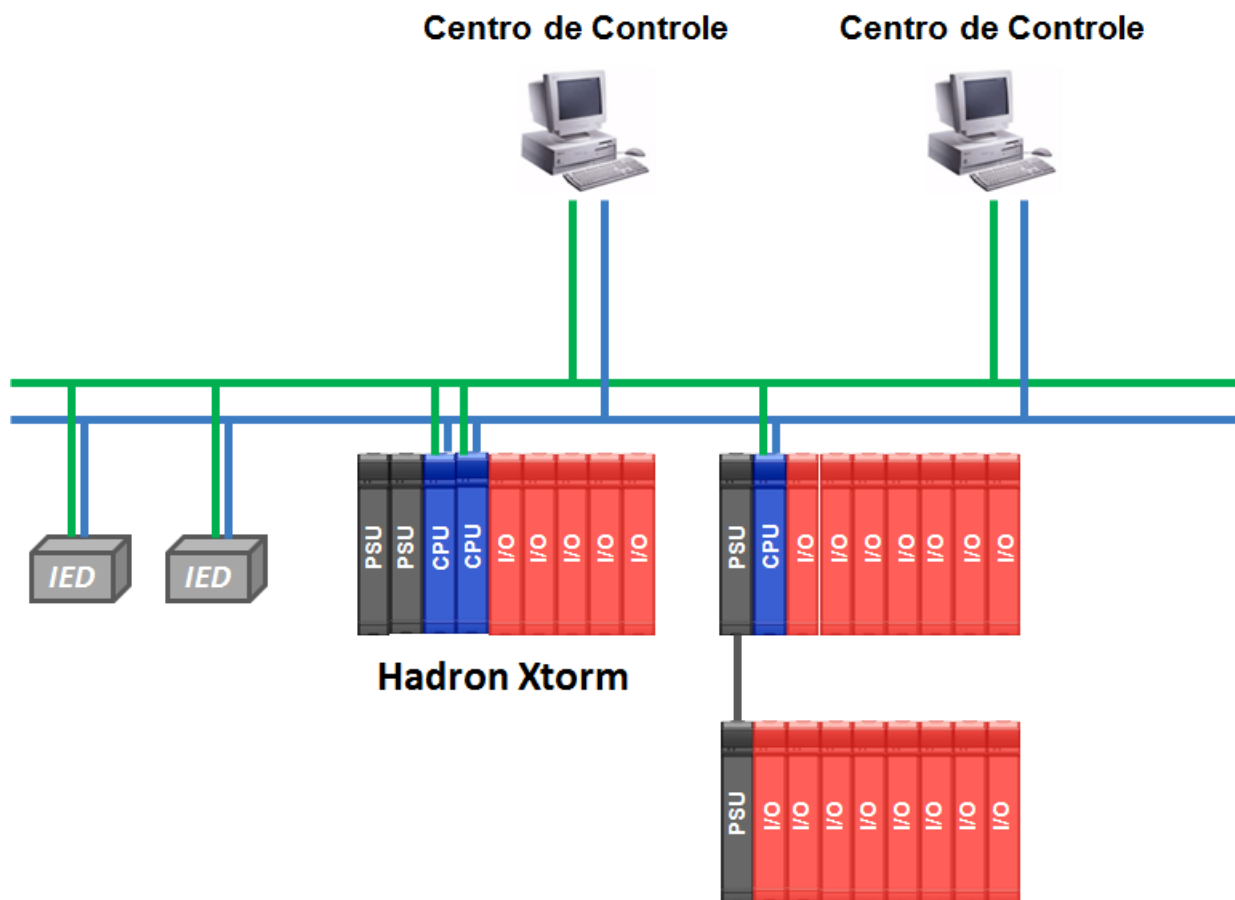


Figura 13: Redes Ethernet Redundantes com NIC Teaming

1.4.1.8. Redes Ethernet em Modo Anel

Há um modo de operação que transforma um par de interfaces Ethernet em um switch, onde a comunicação pode ser feita por ambas as portas. Isto possibilita a implementação de uma topologia de rede em anel.

Neste anel, é necessário incluir um switch externo com capacidade de gerenciá-lo, para evitar o surgimento de loops que degradem o desempenho da rede.

Todas as interfaces Ethernet possuem diagnósticos individuais, facilitando a depuração de eventuais problemas.

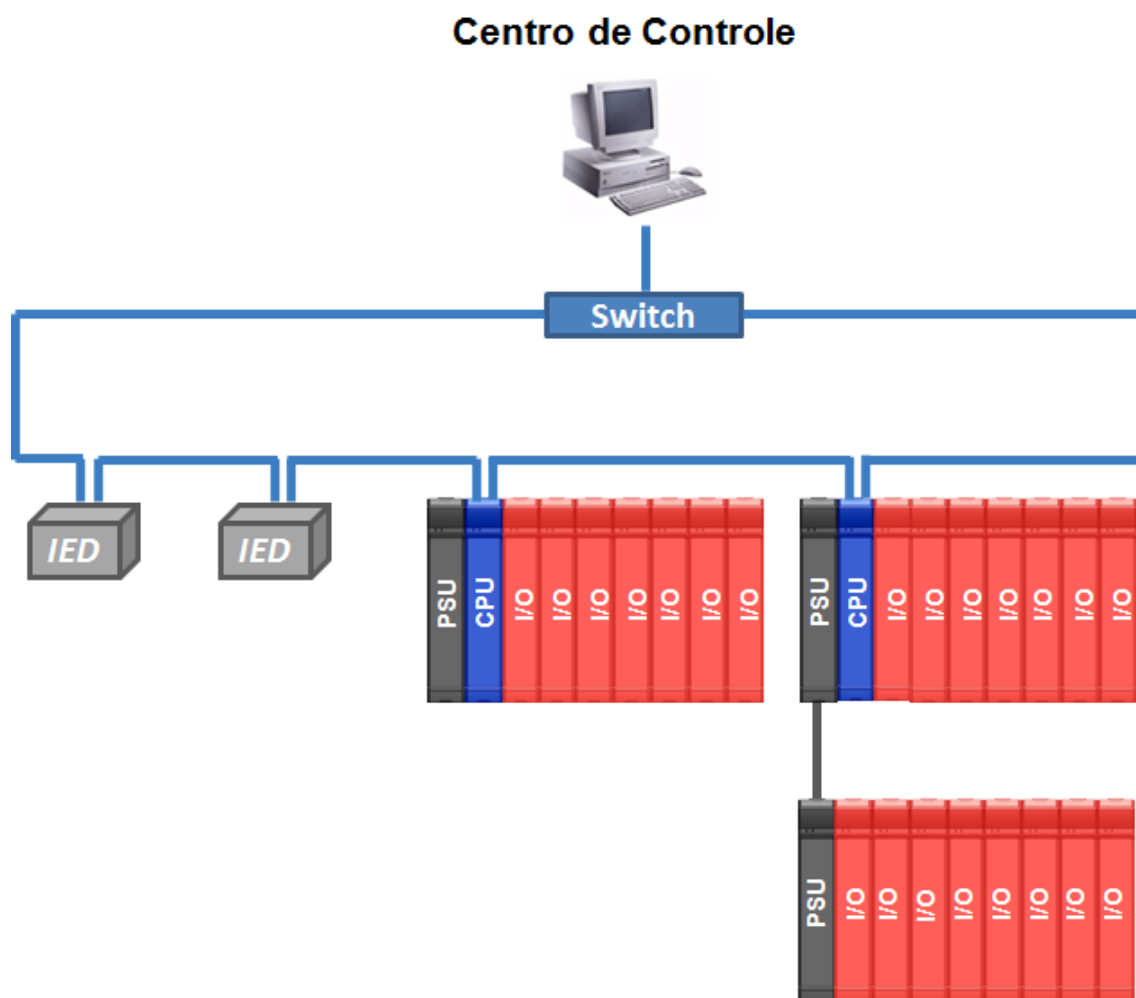


Figura 14: Redes Ethernet em Modo Anel

1.4.1.9. Compatibilidade com Demais Produtos

A Série Hadron Xtorm possui compatibilidade com todas as versões de MasterTool Xtorm, com suporte à expansão de barramento, realizada através do módulo HX8300 ou HX8320. Com o modelo de UCP HX3040 é possível expandir a arquitetura para até 16 bastidores (1 bastidor principal e 15 bastidores de expansão) utilizando a funcionalidade de expansão de barramento. Neste caso, o limite máximo de módulos contabilizados entre todos os bastidores de expansão não pode ultrapassar 80 módulos.

Além disso, é possível também montar uma arquitetura mista utilizando a expansão de barramento da Série Hadron Xtorm para interligar com um bastidor da Série Nexto ou Nexto Jet. Na figura a seguir, pode-se visualizar esta arquitetura proposta.

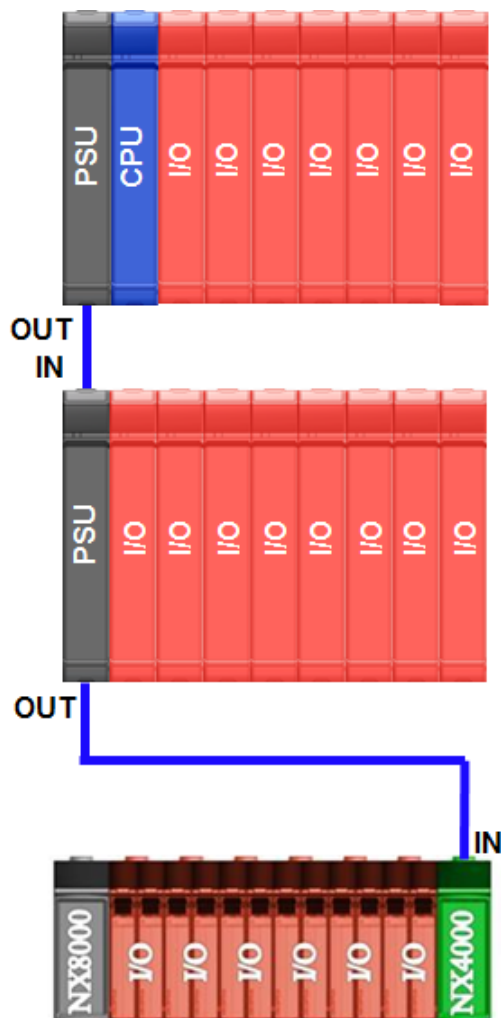


Figura 15: Compatibilidade com Demais Produtos

ATENÇÃO

Para a arquitetura mista, são permitidos apenas módulos de E/S da Série Nexto. Demais módulos da Série Nexto não podem ser instalados em barramentos de expansão misto da Série Hadron Xtorm.

1.5. Documentos Relacionados a este Manual

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 sua última revisão em www.altus.com.br.

Cada produto possui um documento denominado Características Técnicas (CT), onde encontram-se as características do produto em questão.

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

Código	Descrição	Idioma
CE108804	MasterTool Xtorm Technical Characteristics	Inglês
CT108804	Características Técnicas MasterTool Xtorm	Português
CS108804	Características Técnicas MasterTool Xtorm	Espanhol
CE123000	Hadron Xtorm Series Technical Characteristics	Inglês
CT123000	Características Técnicas Série Hadron Xtorm	Português
CS123000	Características Técnicas Serie Hadron Xtorm	Espanhol
CE123100	CPU 6 ETH, 2 SERIALS, IRIG-B, RED Module Technical Characteristics	Inglês
CT123100	Características Técnicas do Módulo UCP 6 ETH, 2 SERIAIS, IRIG-B, RED.	Português
CS123100	Características Técnicas del UCP 6 ETH, 2 SERIALES, IRIG-B, RED.	Espanhol
CE123701	Hadron Xtorm Series Backplane Racks Technical Characteristics	Inglês
CT123701	Características Técnicas dos Bastidores da Série Hadron Xtorm	Português
CS123701	Características Técnicas de los Bastidores de la Serie Hadron Xtorm	Espanhol
CE123200	Redundant Power Supply 60 W Modules Technical Characteristics	Inglês
CT123200	Características Técnicas dos Módulos Fonte de Alimentação Redundante 60 W	Português
CS123200	Características Técnicas delos Módulos Fuente de Alimentación Redundante 60 W	Espanhol
CE123300	Módulo 32 DI 125 Vdc w/ event log Module Technical Characteristics	Inglês
CT123300	Características Técnicas do Módulo 32 ED 125 Vdc c/ registro de eventos	Português
CS123300	Características Técnicas del Módulo 32 ED 125 Vdc c/ registro de eventos	Espanhol
CE123400	16 SD Relay 125 Vdc w/ CBO Module Technical Characteristics	Inglês
CT123400	Características Técnicas do Módulo 16 SD Relé 125 Vdc c/ CBO	Português
CS123400	Características Técnicas del Módulo 16 SD Relé 125 Vdc c/ CBO	Espanhol
CE123310	16 AI Voltage/Current Module Technical Characteristics	Inglês
CT123310	Características Técnicas do Módulo 16 EA Tensão/Corrente	Português
CS123310	Características Técnicas del Módulo 16 EA Tensión/Corriente	Espanhol
CE123313	AC Measurement / 40 Voltage/Current Mixed Module Technical Characteristics	Inglês
CT123313	Características Técnicas do Módulo Misto Medição AC / 4S Tensão/Corrente	Português
CS123313	Características Técnicas del Módulo Mixto Medición AC / 4S Voltaje/Corriente	Espanhol
CE123311	8 AI RTD Module Technical Characteristics	Inglês
CT123311	Características Técnicas do Módulo 8 EA RTD	Português
CS123311	Características Técnicas del Módulo 8 EA RTD	Espanhol
CE123901	Hadron Xtorm Series Connectors Technical Characteristics	Inglês
CT123901	Características Técnicas dos Conectores da Série Hadron Xtorm	Português
CS123901	Características Técnicas de los Conectores de la Serie Hadron Xtorm	Espanhol
CE123900	Rack Connector Cover Technical Characteristics	Inglês
CT123900	Características Técnicas da Tampa para conector de bastidor	Português
CS123900	Características Técnicas de la Tapa para conector de bastidor	Espanhol
MU223600	Hadron Xtorm Utilization Manual	Inglês
MU223000	Manual de Utilização Hadron Xtorm	Português
MU223601	Hadron Xtorm DNP3 Server Device Profile Document	Inglês
MU223602	Hadron Xtorm DNP3 Client Device Profile Document	Inglês
MU223603	IEC 60870-5-104 Server Device Profile Document	Inglês

Código	Descrição	Idioma
MU223604	Hadron Xtorm IEC 60870-5-104 Client Device Profile Document	Inglês
MU223605	Hadron Xtorm IEC 61850 Server Device Profile Document	Inglês
MP399609	MasterTool IEC XE Programming Manual	Inglês
MP399048	Manual de Programação MasterTool IEC XE	Português

Tabela 1: Documentos Relacionados

1.6. Suporte Técnico

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

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

1.7. Mensagens de Advertência Utilizadas neste Manual

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

PERIGO

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

CUIDADO

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

ATENÇÃO

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

2. Descrição Técnica

Este capítulo apresenta todas as características técnicas da UCP da Série Hadron Xtorm HX3040.

2.1. Painéis e Conexões

A figura a seguir mostra o painel frontal da UCP HX3040.



Figura 16: UCP HX3040

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

A UCP da Série Hadron Xtorm apresenta um visor gráfico contendo o status e informações úteis para o usuário, tais como: estados da aplicação (em execução e parado), status do cartão miniSD, atividade nas interfaces seriais (RX e TX) entre outras. Adicionalmente, as UCPs da Série Hadron Xtorm também fornecem um LED bicolor usado para indicar os status e diagnósticos. Para maiores informações a respeito do significado do status do LED de diagnósticos, consulte o documento com as características técnicas da UCP – CT123100.

No painel frontal estão disponíveis as interfaces de conexão da UCP da Série Hadron Xtorm. Essas interfaces são de comunicação Ethernet, comunicação serial, interface do cartão de memória e interface de sincronismo de tempo. A tabela abaixo apresenta uma breve descrição dessas interfaces.

Interfaces	Descrição
NET 1 .. NET 6	Conector tipo RJ45 de comunicação no padrão 10/100Base TX. Permite a comunicação ponto a ponto ou em rede nos protocolos abertos, MODBUS TCP cliente e servidor, MODBUS RTU via TCP cliente e servidor, DNP3 cliente e servidor, IEC 60870 5 104 cliente e servidor, IEC 61850 MMS servidor, IEC 61850 GOOSE publish e subscriber. Para obter mais informações sobre utilização, consulte a seção Configuração das Interfaces Ethernet .
COM 1	Conector tipo DB9 fêmea para comunicação no padrão RS 232C e RS 485. Permite a comunicação ponto a ponto ou em rede nos protocolos abertos, MODBUS RTU escravo ou MODBUS RTU mestre. Para obter mais informações sobre utilização, consulte a seção Configuração das Interfaces Seriais .
COM 2	Conector tipo DB9 fêmea para comunicação nos padrões RS 485 e RS 422. Permite a comunicação ponto a ponto ou em rede nos protocolos abertos, MODBUS RTU escravo ou MODBUS RTU mestre. Para obter mais informações sobre utilização, consulte a seção Configuração das Interfaces Seriais .
MEMORY SLOT	Conector para cartão de memória. Permite utilizar um cartão de memória para diferentes tipos de armazenamento de dados como: logs de usuário, páginas Web, a documentação do projeto e arquivos de origem. Para obter mais informações sobre utilização, consulte seção Cartão de Memória .
IRIG-B	Borne conector com 4 terminais para comunicação no padrão IRIG B. Para obter mais informações sobre utilização, consulte seção IRIG-B .


Tabela 2: Interfaces de Conexão

2.2. Características Gerais

2.2.1. Características Comuns

	HX3040
Linguagens de programação	Lista de Instruções (IL) Texto Estruturado (ST) Diagrama Ladder (LD) Sequenciamento Gráfico de Funções (SFC) Diagrama de Blocos Funcionais (FBD) Gráfico Contínuo de Funções (CFC)
Tipos de tarefas	Cíclica (periódica) Disparada por evento (interrupção de software) Disparada por evento externo (interrupção de hardware) Contínua (execução livre) Disparada por estado (interrupção de software)
Alterações online	Sim
Suporte a troca quente	Sim

2. DESCRIÇÃO TÉCNICA

	HX3040
Suporte a redundância de expansão de barramento	Sim
Interfaces seriais	1 x RS-232C / RS-485 (COM 1) 1 x RS-485 / RS-422 (COM 2)
Protocolo MODBUS	Mestre e escravo RTU (COM 1 e COM 2) Cliente e servidor TCP (NET 1 .. NET 6) Cliente e servidor RTU via TCP (NET 1 .. NET 6)
Protocolo DNP3	Cliente e servidor TCP (NET 1 .. NET 6)
Protocolo IEC 60870-5-104	Cliente e servidor TCP (NET 1 .. NET 6)
Protocolo IEC 61850 MMS	Servidor TCP (NET 1 .. NET 6)
Protocolo IEC 61850 GOOSE	Publish e Subscriber (NET 1 .. NET 6)
Protocolo SNTP	Cliente
Protocolo PTP	Precision Time Protocol (Escravo)
Fila de eventos Tamanho Política de estouro Retentividade	4.500 eventos Mantém os mais recentes Sim
Relógio de tempo real (RTC)	Sim Resolução de 1 ms e variação máxima de 2s por dia
Cão-de-guarda	Sim
Indicação de status e diagnóstico	Visor gráfico, LED, páginas web e memória interna da UCP
One Touch Diag (OTD)	Sim
Electronic Tag on Display (ETD)	Sim
Isolação Lógica para terra de proteção Ⓢ Lógica para interfaces Ethernet Lógica para porta serial (COM 2) Lógica para porta IIRIG-B Interfaces Ethernet para terra de proteção Ⓢ Interfaces Ethernet para porta serial (COM 2) Interface Ethernet para interface Ethernet Porta Serial (COM 2) para terra de proteção Ⓢ	2500 Vac/ 1 minuto 1500 Vac/ 1 minuto 2000 Vac/ 1 minuto 2500 Vac/ 1 minuto 1500 Vac/ 1 minuto 2500 Vac/ 1 minuto 1500 Vac/ 1 minuto 2500 Vac/ 1 minuto
Consumo de corrente do bastidor	1500 mA
Dissipação	7,5 W
Temperatura de operação	-5 a 60°C
Temperatura de armazenamento	-25 a 75°C
Umidade relativa de operação e armazenamento	5 a 96 %, sem condensação
Revestimento de circuitos eletrônicos	Sim
Índice de proteção	IP 20
Normas IEC 61131-2 IEC 61131-3 IEC 61850  2014/35/EU (LVD) e 2014/30/EU (EMC)	Sim Sim Sim Sim

2. DESCRIÇÃO TÉCNICA


	HX3040
 RoHS 2011/65/EU	Sim
Dimensões do módulo (L x A x P)	38,0 x 235,3 x 184,2 mm
Dimensões do módulo (L x A x P)	55,0 x 308,0 x 266,0 mm
Peso	1000 g
Peso com embalagem	1300 g

Tabela 3: Características Comuns

Notas:

Tipos de tarefas: Tarefa é um objeto usado para chamar POU's. Uma tarefa pode ser disparada por período, eventos ou pode ser executada no modo contínuo. Cada tarefa pode chamar uma ou mais POU's.

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

One Touch Diag (OTD): Essa opção somente estará disponível ao usuário quando o módulo estiver em modo operacional.

Isolação: Lógica é o nome para os circuitos internos como processador, memória e interfaces com barramento.

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

2.2.2. Características Gerais Específicas

	HX3040
Memória de variáveis de entrada de representação direta (%I)	96 Kbytes
Memória de variáveis de saída de representação direta (%Q)	96 Kbytes
Memória de variáveis simbólicas	6 Mbytes
Memória de variáveis simbólicas retentivas	8 Kbytes
Memória de variáveis simbólicas persistentes	4 Kbytes
Memória de variáveis redundantes	512 Kbytes
Memória de programa	12 Mbytes
Memória de código fonte (backup)	100 Mbytes
Memória de arquivos de usuário	32 Mbytes
Número máximo de tarefas	32
Número máximo de barramento de expansão	15
Ethernet TCP/IP interface local	6
Suporte a redundância de interfaces Ethernet TCP/IP	Sim
Suporte a redundância de UCP (mesmo bastidor)	Sim
Sincronização do relógio (IRIG-B, SNTP e PTP)	Sim

Tabela 4: Características Gerais Específicas

Notas:

Memória de variáveis de entrada de representação direta (%I): Área onde são alocadas as variáveis de representação direta para o tipo entrada. Variável de representação direta significa que a variável pode ser acessada diretamente na memória utilizando o endereço desejado. Por exemplo: A UCP HX3040 da Série Hadron Xtorm define toda área de memória de variáveis de entrada de representação direta %I como variáveis redundantes, ou seja, o usuário não precisa selecionar tal área.

Memória de variáveis de saída de representação direta (%Q): Área onde são alocadas todas as variáveis de representação direta para o tipo saída. Variável de representação direta significa que a variável pode ser acessada diretamente na

memória utilizando o endereço desejado. Por exemplo: A UCP HX3040 da Série Hadron Xtorm define toda área de memória de variáveis de saída de representação direta %Q como variáveis redundantes, ou seja, o usuário não precisa selecionar tal área.

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

Memória de variáveis simbólicas retentivas: Área onde são alocadas as variáveis simbólicas retentivas. Os dados retentivos mantêm seus respectivos valores mesmo após um ciclo de desenergização e energização da UCP. A lista completa de quando as variáveis retentivas mantêm seus valores e quando o valor é perdido, pode ser encontrada na próxima tabela.

Memória de variáveis simbólicas persistentes: Área onde são alocadas as variáveis simbólicas persistentes. Os dados persistentes mantêm seus respectivos valores mesmo após o download de uma nova aplicação na UCP.

ATENÇÃO

A declaração e utilização de variáveis persistentes deve ser realizada única e exclusivamente através do objeto Persistent Vars, o qual pode ser incluído no projeto através da treeview em Application -> Add Object -> Persistent Variables. Não deve ser utilizada a expressão VAR PERSISTENT no campo de declaração de variáveis das POU's.

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

A tabela mostra o comportamento das variáveis simbólicas, retentivas e persistentes para diferentes situações, onde “-” significa que o valor é perdido e “X” significa que o valor é mantido.

Comando	Variável Simbólica	Variável Retentiva	Variável Persistente
Reset a Quente / Ciclo de energização	-	X	X
Reset a Frio	-	-	X
Reset Origem	-	-	-
Remover UCP ou Fonte de Alimentação do Bastidor enquanto energizado	-	-	-
Download	-	-	X
Alteração online	X	X	X
Reiniciar CP	-	X	X
Limpa tudo	-	-	X
Reset Process (IEC 60870-5-104)	-	X	X

Tabela 5: Comportamento das Variáveis Não Voláteis

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

Memória de variáveis redundantes: Apenas aplicado para projetos com redundância de UCP. São as variáveis sincronizadas entre as UCP's redundantes, o que contempla as variáveis de representação direta (%I e %Q), utilizadas por módulos de E/S, e variáveis simbólicas do programa do usuário, declaradas em GVL's ou POU's redundantes.

Memória de programa: Área da memória que corresponde ao tamanho máximo permitido para a aplicação de usuário. Essa área é compartilhada com a memória de código fonte, sendo a área total a soma de “memória de programa” e “memória de código fonte”.

Memória de código fonte (backup): Área da memória utilizada como backup do projeto, ou seja, caso o usuário deseje importar o seu projeto, o software MasterTool Xtorm irá buscar as informações necessárias nessa área. É importante garantir

2. DESCRIÇÃO TÉCNICA

que o projeto salvo como backup está atualizado para evitar perda de informações críticas. Essa área é compartilhada com a memória de programa sendo a área total a soma de “memória de programa” e “memória de código fonte”.

Memória de arquivos de usuário: Essa área da memória é destinada ao armazenamento de arquivos, como: doc, pdf, imagens, entre outros, ou seja, permite a gravação de dados como se fosse um cartão de memória.

Suporte a redundância (mesmo bastidor): A UCP HX3040 suporta redundância de UCP localizadas no mesmo bastidor. Para maiores informações consulte a seção [Redundância com UCP HX3040](#).

2.2.3. Interfaces Seriais

2.2.3.1. COM 1

COM 1	
Conector	DB9 fêmea blindado
Meio físico	RS-232C ou RS-485 (dependendo do cabo concetado)
Direção da comunicação	RS-232C: full duplex RS-485: half duplex
Número máximo de transceivers RS-485	32
Terminação RS-485	Não (possibilita o uso de terminação ativa externa)
Sinais de modem	RTS, CTS, DCD
Baud rate	600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200 bps
Protocolos	MODBUS RTU (mestre/ escravo) Protocolo aberto

Tabela 6: Interface Serial COM 1

2.2.3.2. COM 2

COM 2	
Conector	DB9 fêmea blindado
Meio físico	RS-232C ou RS-485 (dependendo do cabo concetado)
Direção da comunicação	RS-232C: full duplex RS-485: half duplex
Número máximo de transceivers RS-422	11 (1 transistor e 10 receptores)
Número máximo de transceivers RS-485	32
Terminação	Sim (opcional via seleção de cabo)
Baud rate	600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200 bps
Protocolos	MODBUS RTU (mestre/ escravo) Protocolo aberto

Tabela 7: Interface Serial COM 2

Notas:

Meio físico: Dependendo da configuração do cabo usado, é possível escolher o tipo de interface física: RS-232C ou RS-485 para a COM1, e RS-422 ou RS-485 para a COM2.

Número máximo de transceivers RS-422: Referem-se ao número máximo de interfaces RS-422 que podem ser usadas no mesmo barramento.

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

2.2.4. Interfaces Ethernet

2.2.4.1. NET 1 .. NET 6

NET 1 .. NET 6	
Conector	RJ45 fêmea blindado
Auto negociação (auto crossover)	Sim
Comprimento máximo do cabo	100 m
Tipo de cabo	UTP ou ScTP, categoria 5
Baud rate	10/100 Mbps
Camada física	10/100 Base-TX
Camada de enlace de dados	LLC (controle de enlace lógico)
Camada de rede	IP (protocolo de internet)
Camada de transporte	TCP (protocolo de controle de transmissão) UDP (protocolo de datagrama de usuário)
Camada de aplicação	Cliente / Servidor MODBUS TCP Cliente / Servidor MODBUS RTU via TCP Cliente / Servidor DNP3 Cliente / Servidor IEC 60870 5 104 Envio e recebimento GOOSE (IEC 61850) Servidor MMS (IEC 61850) Servidor HTTP (WEB) Servidor Mastertool Xtorm (programação apenas NET 1) Cliente SNTP (sincronismo de relógio) Escravo PTP (sincronismo de relógio)

Tabela 8: Interfaces Ethernet NET 1 .. NET 6

Notas:

Protocolo de Programação MasterTool Xtorm: Nos casos em que a NET 1 estiver configurada como redundante, a NET 2 também poderá ser utilizada para o Protocolo de Programação MasterTool Xtorm.

2.2.5. IRIG-B

IRIG-B	
Tipo de Conector	Conector borne extraível com 4 terminais (HX9405)
Comprimento máximo do cabo	5 m
Seção do fio	0,5 mm ²
Nível de entrada e saída	TTL
Impedância de entrada	>100 kΩ
Atraso entre entrada e saída	< 10 ns
Corrente máxima na saída	10 mA
Carga máxima na saída	500 Ω
Proteção contra curto circuito	Sim
Níveis de tensão	0 a 1,5 Vdc para nível lógico 0 3,5 a 5 Vdc para nível lógico 1

Tabela 9: Interface IRIG-B

Notas:

Carga Máxima na Saída: A carga total resultante de todos os dispositivos conectados à saída não deve ultrapassar este valor. Não há um limite máximo de dispositivos pré-definido. O mesmo deve ser calculado a partir da impedância de entrada mínima de cada um dos dispositivos conectados à saída IRIG-B da Xtorm.

2.2.6. Visor Gráfico

As UCPs da Série Hadron Xtorm possuem um visor gráfico usado para mostrar o status e diagnóstico do sistema inteiro, inclusive os diagnósticos específicos de cada módulo adicional. O visor também oferece um menu de fácil utilização que dá ao usuário um rápido caminho para ler ou definir alguns parâmetros como: temperatura interna (somente leitura); contraste do visor gráfico e endereço IP para cada interface NET (somente leitura). Mais informações sobre como usar o visor gráfico podem ser encontradas na seção [Visor Gráfico](#).

2.2.7. Interface do Cartão de Memória

Os cartões de memória podem ser usados para diferentes tipos de armazenamento de dados como: logs de usuários, páginas web, documentação de projeto e arquivos fontes. Mais informações sobre como usar interface de cartão de memória podem ser encontradas na seção [Cartão de Memória](#).

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

Tabela 10: Características da Interface com Cartão de Memória

Notas:

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

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

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

2.3. Compatibilidade com Demais Produtos

A tabela a seguir traz informações referente à relação de compatibilidade das versões de UCPs da Série Hadron Xtorm com as versões do MasterTool Xtorm:

UCPs da Série Xtorm	MasterTool Xtorm	Versão de Firmware
HX3040	1.00 à 1.09	1.0.0.0 à 1.7.58.0
HX3040	2.01 ou superior	1.9.4.0 ou superior

Tabela 11: Compatibilidade entre UCPs e MasterTool

A tabela a seguir traz informações referente à compatibilidade dos produtos da Série Nexto com a Hadron Xtorm:

Produto	Versão de Firmware
NX1001	1.2.0.2 ou superior
NX1005	1.2.0.3 ou superior
NX2001	1.2.0.2 ou superior
NX2020	1.2.0.2 ou superior
NX6000	1.2.0.2 ou superior
NX6010	1.0.0.0 ou superior
NX6020	1.0.0.0 ou superior
NX6100	1.2.0.1 ou superior

Tabela 12: Compatibilidade com Demais Produtos

2.4. Desempenho

O desempenho da UCP da Série Hadron Xtorm depende de alguns fatores, entre eles:

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

2.4.1. Tempos de Aplicação

O tempo de execução de uma aplicação da UCP da Série Hadron Xtorm depende das seguintes variáveis:

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

É importante ressaltar que o tempo de execução da tarefa “MainTask” será diretamente influenciado pela tarefa de sistema “Configuration”, uma tarefa de alta prioridade, executada periodicamente pelo sistema. A tarefa “Configuration” poderá interromper a “MainTask”.

2.4.2. Tempos para Execução de Instruções

Na a seguir, encontram-se os tempos necessários para a realização de diferentes instruções na UCP da Série Hadron Xtorm:

Instrução	Linguagem	Variáveis	Tempos da Instrução (μs)
1000 Contatos	LD	BOOL	3
1000 Divisões	ST	INT	22
		REAL	41
	LD	INT	22
		REAL	41
1000 Multiplicações	ST	INT	8
		REAL	12
	LD	INT	8
		REAL	12
1000 Somas	ST	INT	8
		REAL	12
	LD	INT	8
		REAL	12

Tabela 13: Tempos de Instruções

2.4.3. Tempos de Inicialização

A UCP da Série Hadron Xtorm possui tempo de inicialização de 25 segundos, sendo que a tela inicial com o logotipo HADRON XTORM (Splash) é apresentada depois de 7 segundos da energização. O tempo de inicialização dos módulos de E/S depende de vários fatores tais como número de módulos no barramento, intervalo da “MainTask”, etc.

2.5. Produtos Relacionados

Os seguintes produtos devem ser adquiridos separadamente quando necessário:

Código	Descrição
HD8500	MasterTool Xtorm
AL-2600	Derivador e terminador de rede RS-485
AL-2306	Cabo RS-485 p/ rede MODBUS ou CAN
AL-1729	Cabo RJ45-CMDB9
AL-1748	Cabo CMDB9-CFDB9
AL-1752	Cabo CMDB9-CMDB9
AL-1753	Cabo CMDB9-CMDB25
AL-1754	Cabo CMDB9-CFDB9
AL-1762	Cabo CMDB9-CMDB9
AL-1763	Cabo CMDB9-borneira
NX9202	Cabo RJ45-RJ45 2m
NX9205	Cabo RJ45-RJ45 5m
NX9210	Cabo RJ45-RJ45 10m
NX9101	Cartão microSD de 32 GB com adaptadores SD e miniSD
HX9401	Conector com 6 terminais
HX9402	Conector com 10 terminais
HX9405	Conector com 4 terminais
HX9102	Tampa de conector de bastidor

Tabela 14: Produtos Relacionados

Notas:

HD8500: MasterTool Xtorm é a ferramenta de programação utilizada para a Série Hadron.

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

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

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

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

AL-1752: Cabo padrão RS-232C com dois conectores DB9 macho para comunicação entre a UCP da Série Hadron Xtorm e outros produtos Altus da Série H e IHMs da Série IX.

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

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

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

AL-1763: Cabo com um conector DB9 macho e terminais para comunicação entre a UCP da Série Hadron Xtorm e produtos com bornes padrão RS-232C/RS-485/RS-422.

2. DESCRIÇÃO TÉCNICA

NX9202, NX9205 e NX9210: Cabo Ethernet CAT5, blindado, par trançado, com conectores RJ45 macho em ambas extremidades, suporta temperatura de -5 °C a 70 °C, para ser utilizado em redes Ethernet, com comprimento de 2, 5 e 10 metros, respectivamente.

NX9101: Kit contendo um cartão microSD de 32 Gbytes, um adaptador para o padrão SD e outro adaptador para o padrão miniSD.

HX9401, HX9402 e HX9405: Conectores de 6, 10 e 4 terminais, respectivamente, o conector de 6 posições, HX9401, é usado exclusivamente para as fontes HX8300 e HX8320. O conector de 10 posições, HX9402, pode ser usado em qualquer um dos módulos de entrada e saída da Série Hadron Xtorm. O conector de 4 terminais, HX9405, é de uso exclusivo do canal IRIG-B da UCP HX3040. Suportam temperaturas de -5 °C a 70 °C.

HX9102: O HX9102 consiste em uma tampa protetora de conectores dos bastidores da Série Hadron Xtorm. Esta tampa foi desenvolvida para fornecer uma elevada proteção aos conectores não utilizados do bastidor. É recomendado ao usuário, que todas as posições não utilizadas sejam protegidas com o produto HX9102, suporta temperaturas de -5 °C a 70 °C.

3. Instalação

Este capítulo apresenta os procedimentos necessários para a instalação física dos produtos da Série Hadron Xtorm, bem como os cuidados que se deve ter com outras instalações existentes no armário elétrico ocupado pelo UCP.

3.1. Inspeção Visual

Antes de proceder a instalação, é recomendável fazer uma inspeção visual cuidadosa dos equipamentos, verificando se não há danos causados pelo transporte nos mesmos. Verifique se todos os componentes de seu pedido estão em perfeito estado e qualquer problema detectado deve ser informado à companhia transportadora e ao representante ou distribuidor Altus mais próximo.

CUIDADO

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

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

3.2. Instalação Mecânica

3.2.1. Fixação do Bastidor

3.2.1.1. Furação para Bastidor 9 Posições

A fixação do bastidor de 9 posições deve ser feita através de cinco parafusos M4 conforme mostrado na figura a seguir.

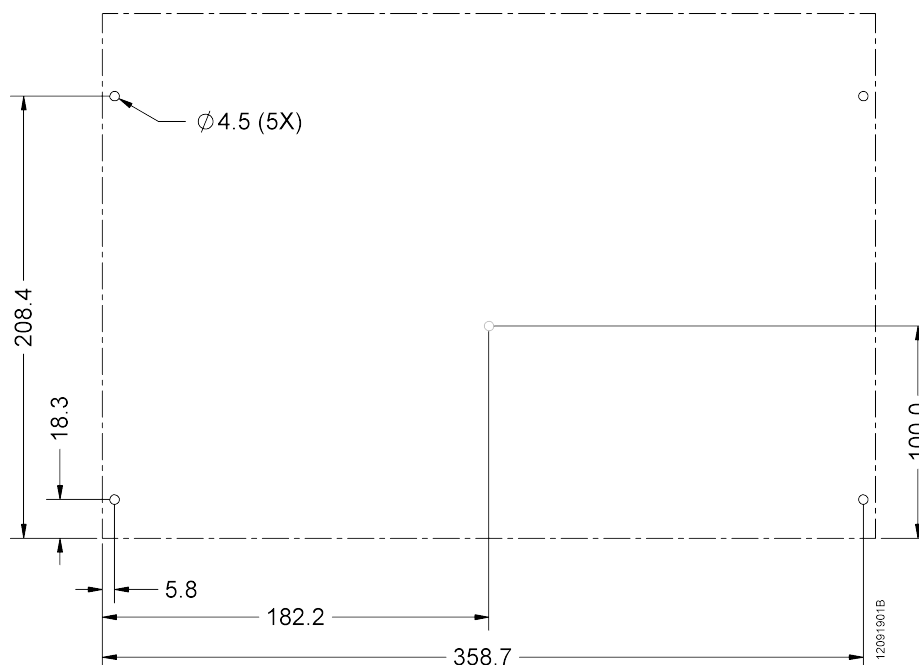


Figura 17: Furação para Fixação Bastidor 9 Posições

3.2.1.2. Furação para Bastidor 18 Posições

A fixação do bastidor de 18 posições deve ser feita através de seis parafusos M4 conforme mostrado na figura a seguir.

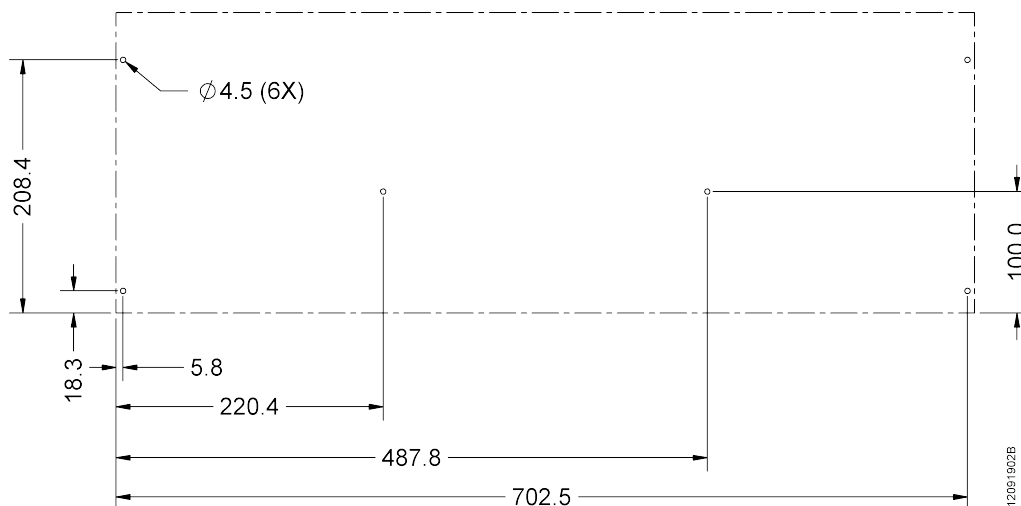


Figura 18: Furação para Fixação Bastidor 18 Posições

Em todos os furos devem ser utilizados parafusos tipo M4 panela fenda cruzada DIN 7985. Estes parafusos podem ser fixados diretamente no painel ou em porcas, quando a espessura do painel for insuficiente para possuir rosca. Ao utilizar porcas, recomenda-se o uso de porcas do tipo auto travante de forma a evitar que a mesma solte.

ATENÇÃO

A seção [Espaçamento entre módulos e outros equipamentos do painel](#) deve ser consultada para verificar as distâncias mínimas que devem existir entre o bastidor da Série Hadron Xtorm e os demais componentes localizados no painel elétrico.

3.2.1.3. Montagem

Antes da inserção do bastidor no painel, os parafusos dos furos tipo 1, conforme mostra a figura 17, devem estar parcialmente inseridos.

Alinhar o bastidor aos dois parafusos tipo 1 e encostar o mesmo no fundo do painel elétrico. A figura 19 indica como deve ser feito tal procedimento.

Obs.: Algumas figuras utilizadas neste item não mostram a placa de circuito impresso do bastidor por motivos de simplificar o entendimento do procedimento.

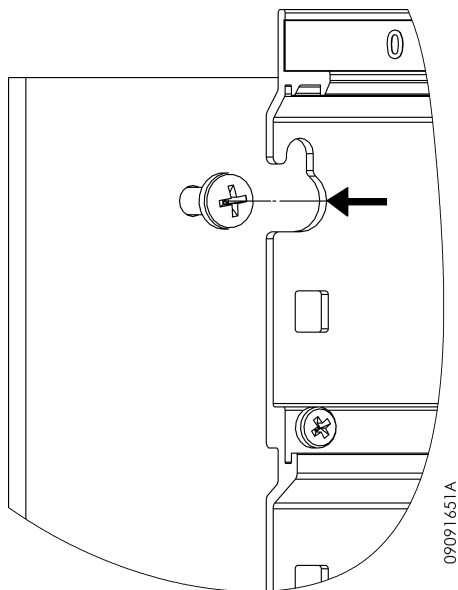


Figura 19: Fixação Bastidor - Alinhamento

Após, deve-se realizar um movimento de tal forma que os parafusos dos furos tipo 1 fiquem encaixados na parte menor do rasgo do bastidor, conforme mostrado na figura a seguir.

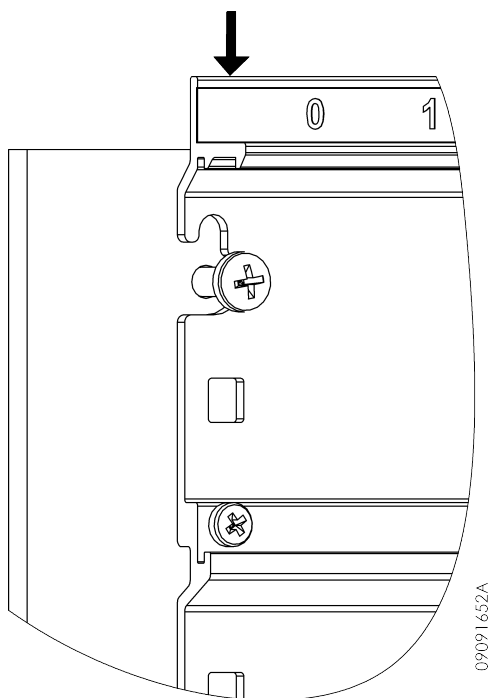


Figura 20: Fixação Bastidor - Encaixe nos Parafusos

Após a completa inserção do bastidor, todos os parafusos utilizados para a fixação do mesmo devem ser montados.

3.2.1.4. Remoção

Para remover o bastidor, é necessário realizar a sequência inversa indicada na seção [Montagem](#).

3.2.2. 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.

Primeiramente deve ser encaixada a parte inferior do módulo, que serve como guia para a correta inserção ao bastidor. Ao encaixar a parte inferior do módulo, deve ser verificado se os pinos guias estão corretamente encaixados aos rasgos do bastidor correspondente a uma determinada posição. A figura abaixo mostra como a parte do inferior do módulo deve estar posicionada em relação ao bastidor para a correta inserção. Deve ser inserido apenas um módulo por vez.

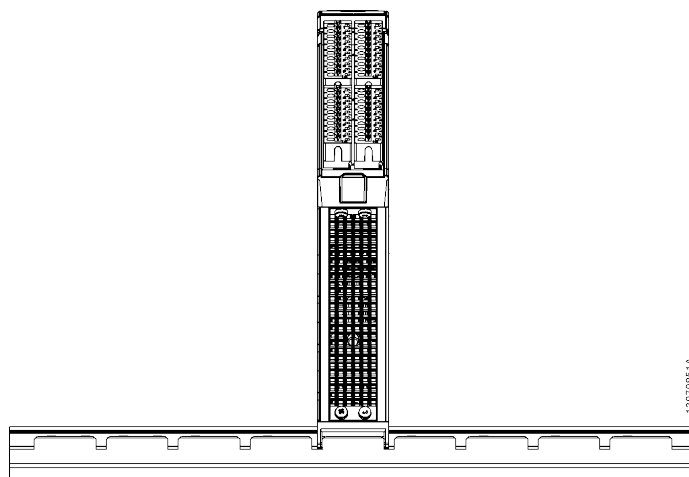


Figura 21: CPU no Bastidor

Após encaixar a parte inferior do módulo conforme descrito acima, deve-se exercer um movimento único e contínuo de rotação, através do cursor de fixação que deverá estar na posição de trava, de forma que a parte inferior do módulo fique no mesmo local e a trava de fixação encaixe na parte superior do bastidor. Após a conclusão deste procedimento, devem-se aguardar no mínimo dois segundos para a execução de um novo procedimento de inserção/remoção. A figura a seguir mostra o movimento que deve ser executado.

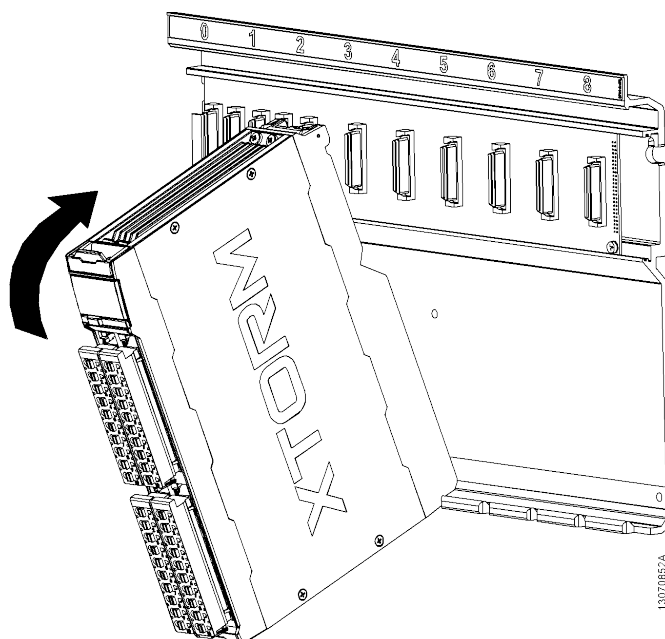


Figura 22: Movimento para Encaixe no Bastidor

3. INSTALAÇÃO

Caso o usuário siga corretamente os procedimentos descritos acima, o módulo estará perfeitamente conectado ao barramento, conforme mostrado na figura abaixo.

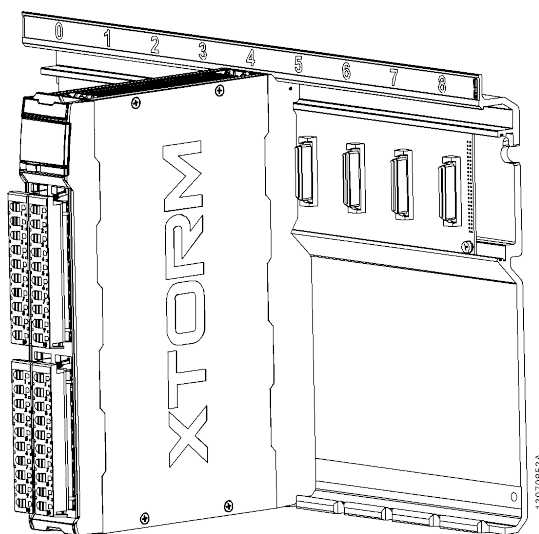


Figura 23: Módulo Encaixado no Bastidor

O módulo não pode ser conectado de outra forma ao bastidor. A tentativa de inserção do módulo de forma errada pode causar danos irreparáveis ao mesmo. A figura abaixo mostra uma maneira de como NÃO se deve conectar os módulos da Série Hadron Xtorm no bastidor.

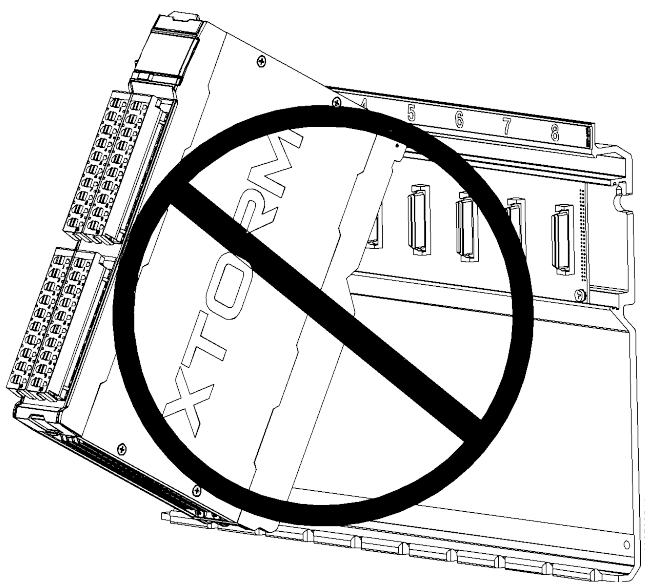


Figura 24: Forma Incorreta de Inserção

3.2.3. Remoção 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.

Primeiramente, deve se puxar o cursor de fixação, conforme indica o número 1 da figura abaixo, a fim de destravá-lo do barramento e então rotacionar em um movimento único e contínuo, no sentido indicado pelo número 2, também da figura abaixo.

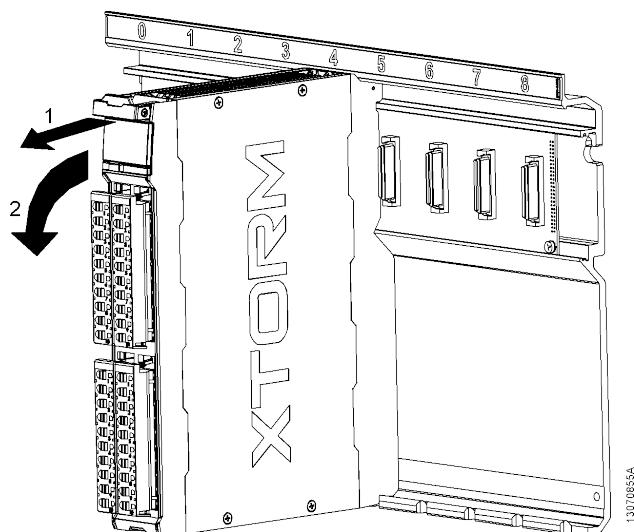


Figura 25: Desconexão do Barramento

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

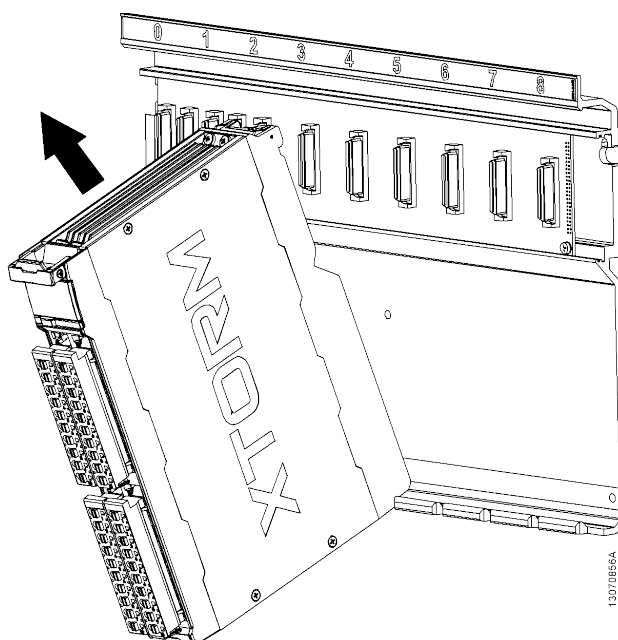


Figura 26: Módulo sendo Removido do Bastidor

3.2.3.1. Módulos de E/S

Os módulos de entrada e saída da Série Hadron Xtorm possuem algumas características específicas que serão abordadas neste item.

3. INSTALAÇÃO

3.2.3.2. Bornes de E/S

Os bornes dos módulos de E/S da Série Hadron Xtorm utilizam o sistema de fixação de fiação com bornes tipo mola não necessitando de parafuso para tal fim.

3.2.3.3. 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.

3.2.3.4. Diagrama de Instalação

O diagrama de instalação de cada módulo é definido no documento Características Técnicas do respectivo módulo.

3.2.3.5. 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.

3.2.3.6. Inserção 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 mostra a figura abaixo.

Na figura 28 a tampa de conector aparece completamente encaixada.

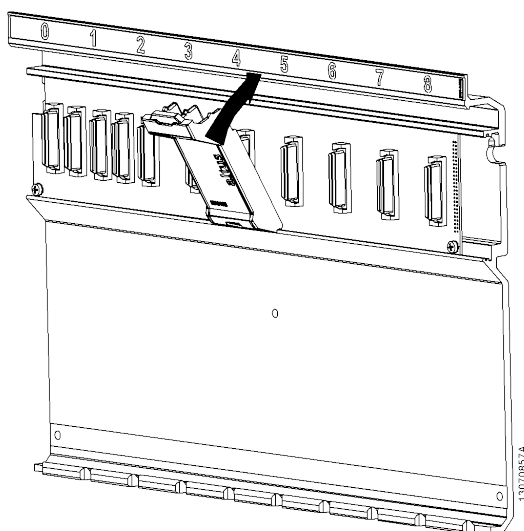


Figura 27: Inserção da Tampa de Conector de Bastidor

3.2.3.7. Remoção da Tampa de Conector de Bastidor

Para remover a tampa de conector HX9102 da Série Hadron Xtorm, basta puxá-la pela borda superior, conforme mostra a figura abaixo.

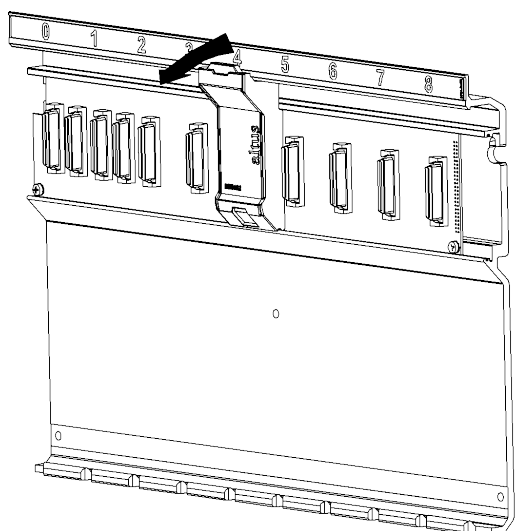


Figura 28: Remoção da Tampa de Conector de Bastidor

3.3. Instalação Elétrica

PERIGO

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

3.3.1. Segurança Elétrica

ATENÇÃO

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 $0,1 \Omega$.

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 tem classificação IP20.

ATENÇÃO

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. É 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.

ATENÇÃO

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 abaixo. Portanto, estas regiões devem ser tocadas apenas com o painel desenergizado.

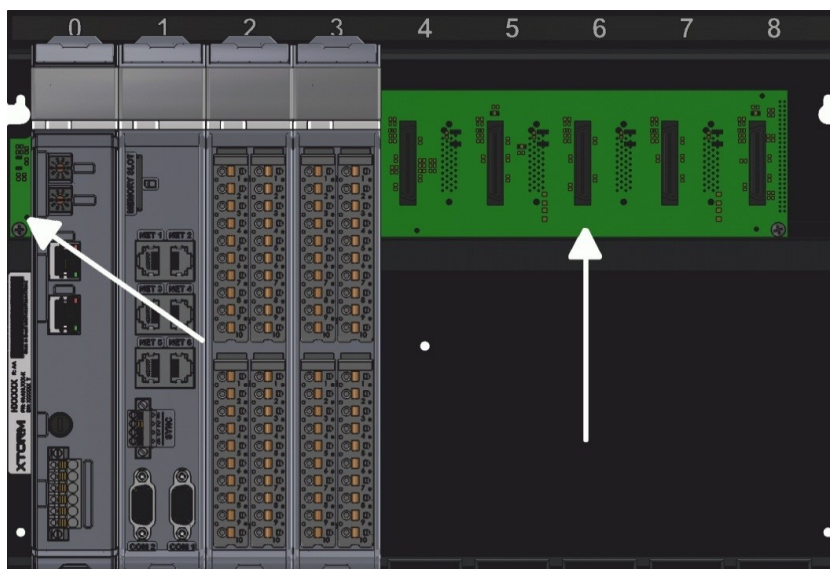


Figura 29: Região das Placas de Circuito Impresso (HX9001 e HX9003)

3.3.2. 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 (Figuras 30 e 31). Para sua montagem recomenda-se utilizar uma chave de fenda com 3,5 mm de largura e cabo isolado (Figura 31). A vantagem de seu emprego é a facilidade e rapidez de montagem dos cabos elétricos.

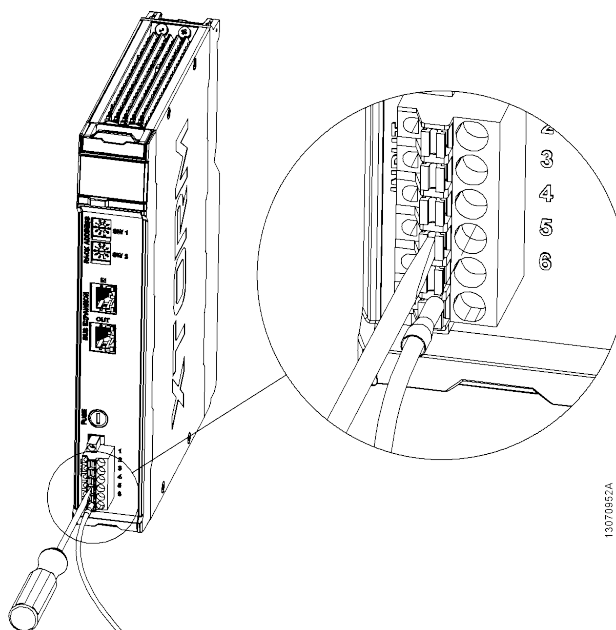


Figura 30: Borne Mola Módulo Fonte

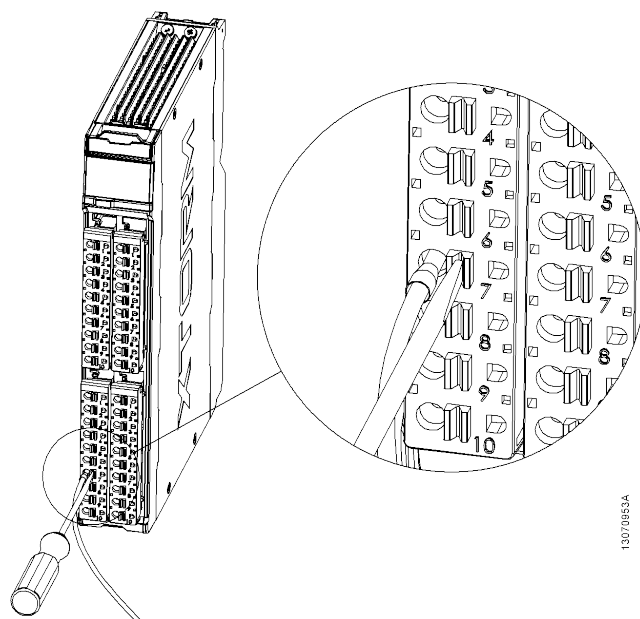


Figura 31: Borne Mola Módulo IO

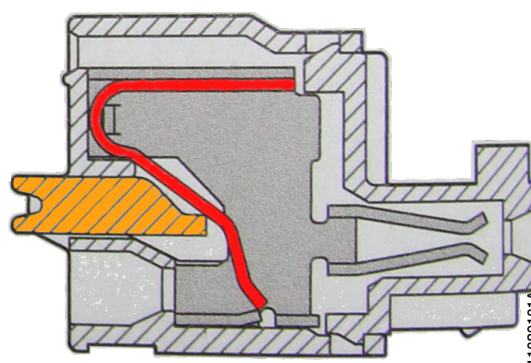


Figura 32: Borne Mola

Para montar o fio no borne:

- Inserir a chave de fenda no acionador do terminal para abrir a mola do borne
- Inserir o terminal do fio no borne
- Retirar a chave para fechar o borne

3.3.2.1. Inserção da Fiação

De forma a obter um comprimento semelhante de todos os fios conectados a um mesmo borne de E/S, indica-se seguir a seguinte definição referente à diferença de comprimento que cada fio subsequente ao outro deve ter antes da instalação. É importante ressaltar que o maior fio devera ser sempre o fio conectado ao pino 1 do borne de E/S.

3.3.2.2. Borne de 4 Vias

Recomenda-se utilizar fio de $0,5 \text{ mm}^2$. Crimpar terminais especificados para o para fio escolhido ($0,5 \text{ mm}^2$) em cada via.

ATENÇÃO

Utilizar terminais simples com comprimento $A = 8 \text{ mm}$ para garantir o efetivo contato (ver figura 34).

3. INSTALAÇÃO

3.3.2.3. Borne de 6 Vias

Recomenda-se utilizar fio de 2,5 mm². Crimpar os terminais especificados para o fio escolhido (2,5 mm²) em cada via.

ATENÇÃO

Utilizar terminais simples com comprimento A = 12 mm para garantir o efetivo contato (ver figura 34).

3.3.2.4. Borne de 10 Vias

Recomenda-se utilizar fio de 0,5 a 1,5 mm². Cortar cada fio com uma diferença de 8 mm, conforme indicado na figura 33. Crimpar terminais especificados para o para fio escolhido (0,5 a 1,5 mm²) em cada via.

ATENÇÃO

Utilizar terminais simples com comprimento A = 12 mm para garantir o efetivo contato (ver figura 34).

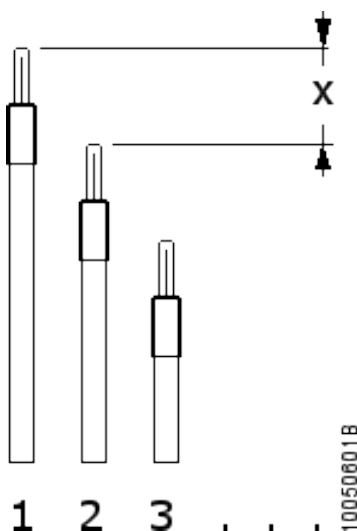


Figura 33: Corte Fiação

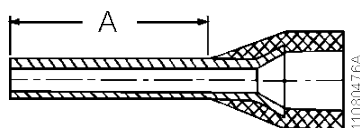


Figura 34: Terminal

3.3.2.5. Montagem da Fiação

Inserir os terminais no borne iniciando pelo pino 10, para o borne de 10 vias. Certifique-se de que os terminais estão completamente inseridos no borne e devidamente conectados.

3.3.2.6. Fixação da Fiação

Girar os cabos para que fiquem acomodados e fixar com amarrilha plástica conforme a figura a seguir.

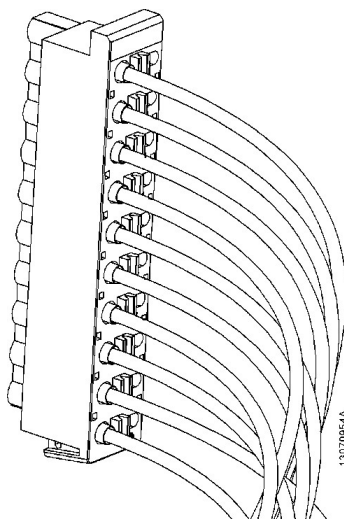


Figura 35: Montagem Cabo no Conector de 10 Vias

3.3.2.7. Remoção da Fiação

Para remover os fios do borne recomenda-se utilizar uma chave de fenda com 3,5 mm de largura e cabo isolado. Inserir a chave de fenda no acionador ao lado do fio enquanto puxa o fio para fora (ver figuras 30 e 31).

3.3.3. Conexões

A correta fixação dos cabos das UCPs e dos módulos do sistema garante a segurança do equipamento e seu correto funcionamento. Para isso, devem ser verificados os seguintes pontos:

- 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 o mesmo tenha sido inserido completamente, garantido uma correta conexão

3.3.4. Alimentações

Conferir se as tensões das alimentações estão dentro dos valores especificados nas características técnicas.

ATENÇÃO

Onde houver alta tensão, colocar etiqueta de aviso e proteções que não permitam o fácil acesso.

3.3.5. Fusíveis

Recomenda-se verificar os fusíveis do sistema, certificando-se que os mesmos estejam em bom estado e com valor e tipo correto, antes de energizar o sistema.

PERIGO

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

3.4. Instalação Elétrica UCP

PERIGO

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

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 abaixo 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.

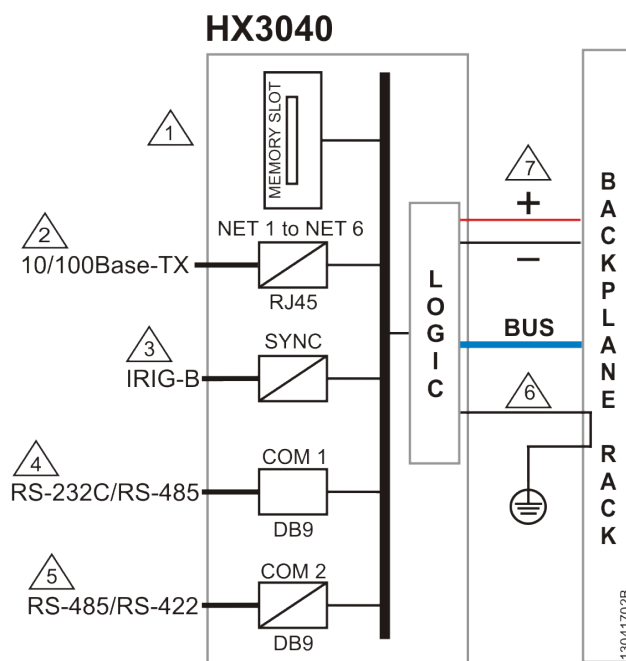


Figura 36: Diagrama Elétrico da UCP da Série Hadron Xtorm

Notas do Diagrama:

- 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 1,5mm².
- 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.

3.5. 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.

3.5.1. Endereço IP

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:

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

Tabela 15: Parâmetros de Fábrica da Interface Ethernet NET 1

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

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

As interface de Ethernet NET 2, NET 3, NET 4, NET 5 e NET 6 são utilizadas para comunicação Ethernet e vem configuradas de fábrica com os seguintes parâmetros:

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

Tabela 16: Parâmetros de Fábrica da Interface Ethernet NET 2

	NET 3
Endereço IP	192.168.17.1
Máscara de Subrede	255.255.255.0
Endereço do Gateway	192.168.17.253

Tabela 17: Parâmetros de Fábrica da Interface Ethernet NET 3

	NET 4
Endereço IP	192.168.18.1
Máscara de Subrede	255.255.255.0
Endereço do Gateway	192.168.18.253

Tabela 18: Parâmetros de Fábrica da Interface Ethernet NET 4

NET 5	
Endereço IP	192.168.19.1
Máscara de Subrede	255.255.255.0
Endereço do Gateway	192.168.19.253

Tabela 19: Parâmetros de Fábrica da Interface Ethernet NET 5

NET 6	
Endereço IP	192.168.20.1
Máscara de Subrede	255.255.255.0
Endereço do Gateway	192.168.20.253

Tabela 20: Parâmetros de Fábrica da Interface Ethernet NET 6

Os parâmetros de rede das interfaces NET 1 .. NET 6 podem ser alterados através do MasterTool Xtorm. Para maiores detalhes sobre configuração e alteração de parâmetros de rede, verifique a seção [Configuração das Interfaces Ethernet](#).

3.5.2. ARP Gratuito

A interface de Ethernet NETx envia espontaneamente pacotes do tipo ARP, em broadcast, informando seu endereço de IP e MAC para todos os dispositivos interligados à rede. Estes pacotes são enviados durante o download de uma nova aplicação pelo software MasterTool 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. Ex.: Primeiro disparo ocorre no tempo 0, o segundo disparo no tempo 200 ms, o terceiro disparo no tempo 600 ms e assim até o quinto disparo no tempo 3 s.

3.5.3. Instalação do Cabo de Rede

As portas Ethernet da UCP da Série Hadron Xtorm, identificadas no painel por NET 1 .. NET 6, possuem pinagem padrão, sendo a mesma utilizada, por exemplo, em computadores pessoais. O tipo de conector, tipo de cabo, nível físico, entre outros detalhes para interligar a UCP ao dispositivo de acesso à rede Ethernet, são definidos na seção [Interfaces Ethernet](#).

A tabela e a figura abaixo apresentam o conector RJ45 fêmea da UCP Hadron Xtorm, com a identificação e a descrição da pinagem válida para os níveis físicos tipo 10Base T e 100Base TX.

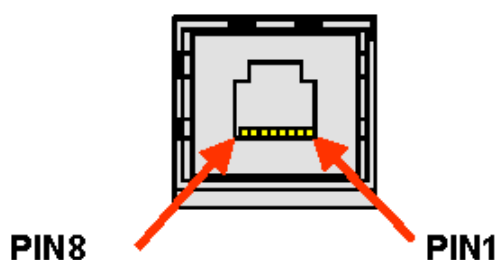


Figura 37: Conector RJ45 Fêmea da UCP Hadron Xtorm

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

Tabela 21: Pinagem do Conector RJ45 Fêmea da UCP Hadron Xtorm

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 Cross-over, o mesmo utilizado para conectar dois computadores pessoais, ponto a ponto, através da porta Ethernet.

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

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

3.6. Conexão com a Rede Serial (COM 1)

ATENÇÃO

Para uma instalação adequada e com segurança da Rede Serial COM 1, deve-se consultar primeiramente a seção [Segurança Elétrica](#), parte integrante deste mesmo capítulo.

A interface não isolada de comunicação COM 1 possibilita a conexão com uma rede RS-232C ou RS-485. A seguir é apresentado o conector DB9 fêmea da UCP Hadron Xtorm, com a identificação e a descrição dos sinais.

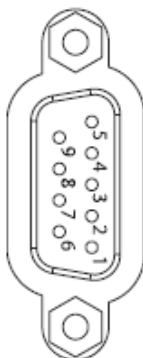


Figura 38: Conector DB9 Fêmea da UCP Hadron Xtorm

3. INSTALAÇÃO

Pino	Sinal	Interface	Descrição
1	DCD	RS-232C	Data Carrier Detect
2	TXT	RS-232C	Transmissão de dados
3	RXD	RS-232C	Recepção de dados
4	D-	RS-485	Transmissão/ Recepção de dados, negativo
5	GND	RS-232C RS-485	RS-232C: Ground RS-485: Referência negativa para terminação externa
6	D+	RS-485	Transmissão/ Recepção de dados, positivo
7	CTS	RS-232C	Clear to Send
8	RTS	RS-232C	Request to Send
9	+5V	RS-485	Referência positiva para terminação externa

Tabela 22: Pinagem do Conector DB9 Fêmea da UCP Hadron Xtorm

3.6.1. Comunicação RS-232C

Para conexão com um dispositivo RS-232C, utilizar o cabo apropriado conforme a seção [Compatibilidade com Demais Produtos](#).

3.6.2. Comunicação RS-485 sem Terminação

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, conforme a tabela abaixo.

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

Tabela 23: Conexão da COM 1 com RS-485 sem Terminação

O diagrama da figura abaixo indica como deve ser realizada a conexão dos terminais do AL-1763 na rede RS-485.

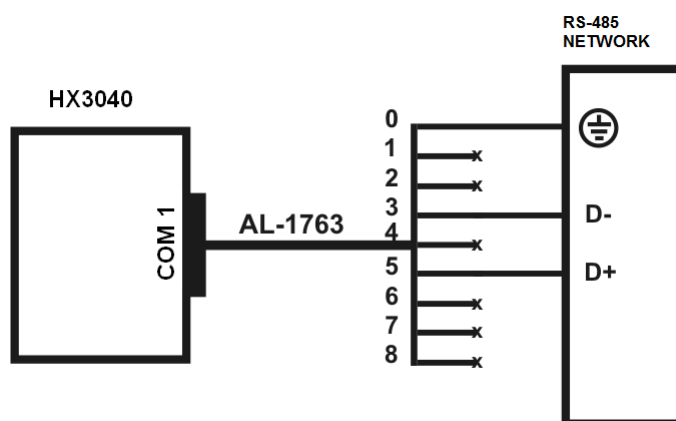


Figura 39: Diagrama de Conexão da COM 1 com RS-485 sem Terminação

Nota do Diagrama:

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

3.6.3. Comunicação RS-485 com Terminação Interna

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.

3.6.4. Comunicação RS-485 com Terminação Externa

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, conforme a tabela abaixo.

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

Tabela 24: Conexão da COM 1 com RS-485 com Terminação Externa

O diagrama da figura abaixo indica como deve ser realizada a conexão dos terminais do AL-1763 na rede RS-485.

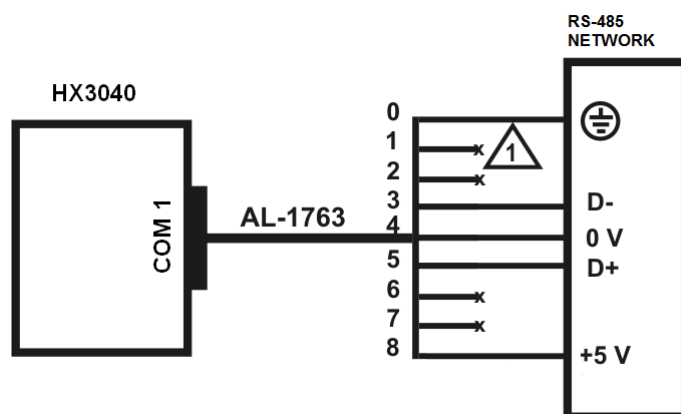


Figura 40: Diagrama de Conexão da COM 1 com RS-485 com Terminação Externa

Nota do Diagrama:

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

3.7. Conexão com a Rede Serial (COM 2)

A interface isolada de comunicação COM 2 possibilita a conexão com uma rede RS-485/422. A seguir é apresentado o conector DB9 fêmea da UCP Hadron Xterm, com a identificação e a descrição dos sinais.

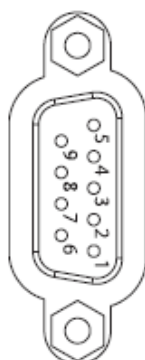


Figura 41: Conector DB9 Fêmea da UCP Hadron Xterm

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

Tabela 25: Pinagem do Conector DB9 Fêmea da UCP Hadron Xterm

3.7.1. Comunicação RS-485 sem Terminação

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, conforme a tabela abaixo.

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

Tabela 26: Conexão da COM 2 com RS-485 sem Terminação

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

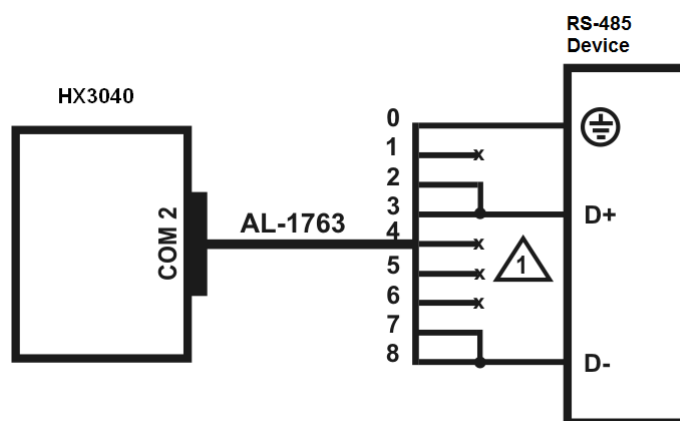


Figura 42: Diagrama de Conexão da COM 2 com RS-485 sem Terminação

Nota do Diagrama:

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

3.7.2. Comunicação RS-485 com Terminação Interna

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, conforme a tabela abaixo.

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

Tabela 27: Conexão da COM 2 com RS-485 com Terminação Interna

Obs.: A terminação interna disponível na COM 2 é do tipo estado seguro em modo aberto.

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

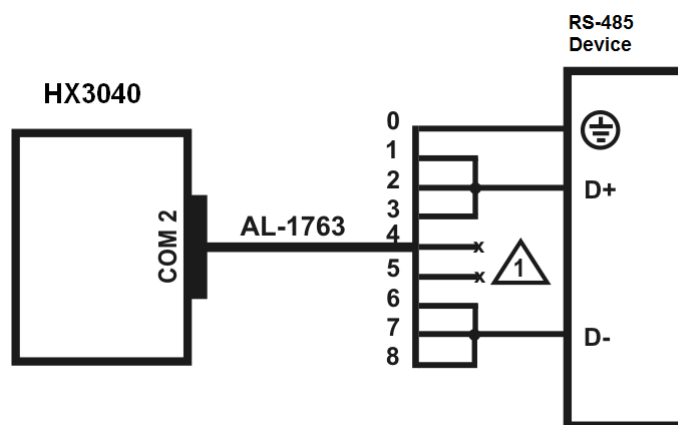


Figura 43: Diagrama de Conexão da COM 2 com RS-485 com Terminação Interna

Nota do Diagrama:

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

3.7.3. Comunicação RS-485 com Terminação Externa

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, conforme a tabela abaixo.

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

Tabela 28: Conexão da COM 2 com RS-485 com Terminação Externa

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

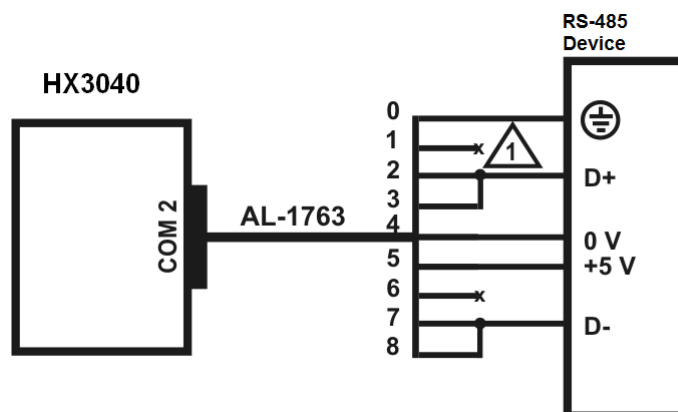


Figura 44: Diagrama de Conexão da COM 2 com RS-485 com Terminação Externa

Nota do Diagrama:

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

3.7.4. Comunicação RS-422 sem Terminação

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, conforme a tabela abaixo.

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

Tabela 29: Conexão da COM 2 com RS-422 sem Terminação

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

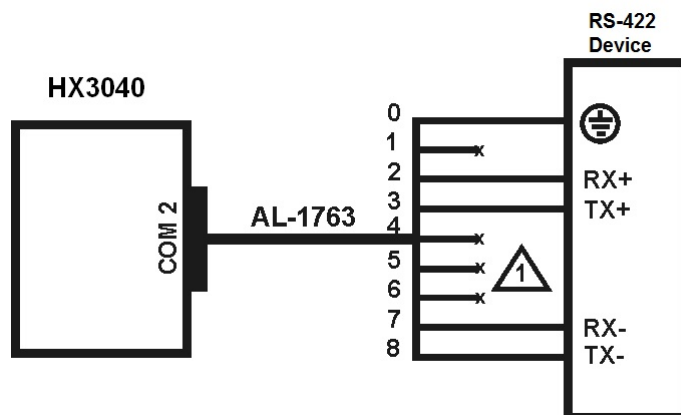


Figura 45: Diagrama de Conexão da COM 2 com RS-422 sem Terminação

Nota do Diagrama:

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

3.7.5. Comunicação RS-422 com Terminação Interna

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, conforme a tabela abaixo.

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

Tabela 30: Conexão da COM 2 com RS-422 com Terminação Interna

Obs.: A terminação interna disponível na COM 2 é do tipo estado seguro em modo aberto.

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

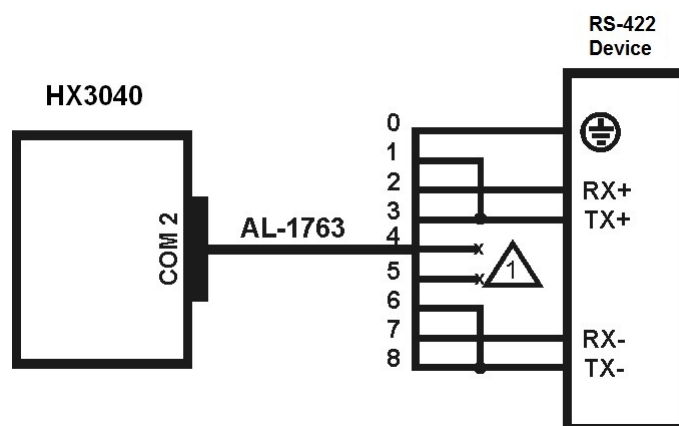


Figura 46: Diagrama de Conexão da COM 2 com RS-422 com Terminação Interna

Nota do Diagrama:

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

3.7.6. Comunicação RS-422 com Terminação Externa

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, conforme a tabela abaixo.

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

Tabela 31: Conexão da COM 2 com RS-422 com Terminação Externa

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

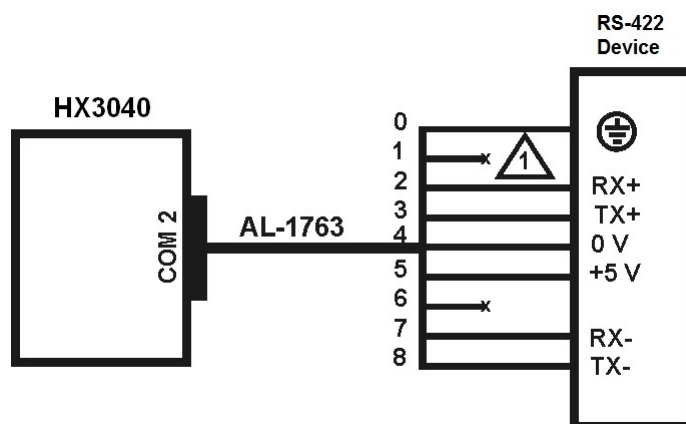


Figura 47: Diagrama de Conexão da COM 2 com RS-422 com Terminação Externa

Nota do Diagrama:

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

3.7.7. Exemplo de Rede RS-422

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

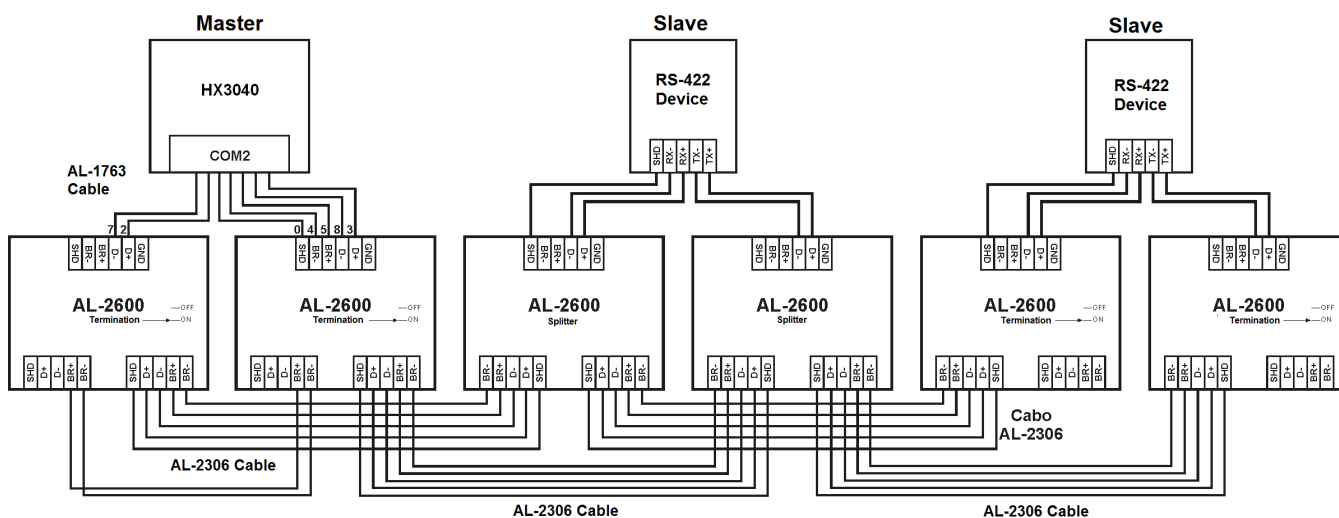


Figura 48: Exemplo de Rede RS-422

Nota do Diagrama:

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

Nota:

Todos os métodos de terminação são polarizados para comunicação RS-485 e RS-422 – não há terminações passivas.

3.8. 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 abaixo.

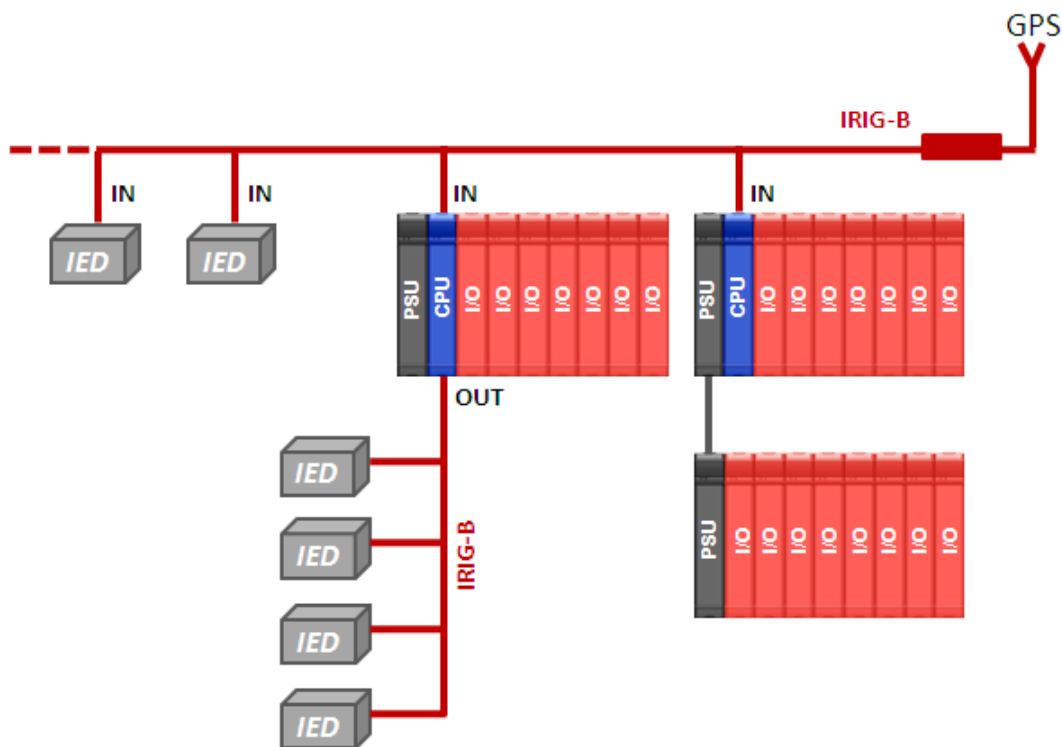


Figura 49: Exemplo de Sincronismo para Sistema com Expansões

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 abaixo podemos visualizar um exemplo deste tipo de arquitetura.

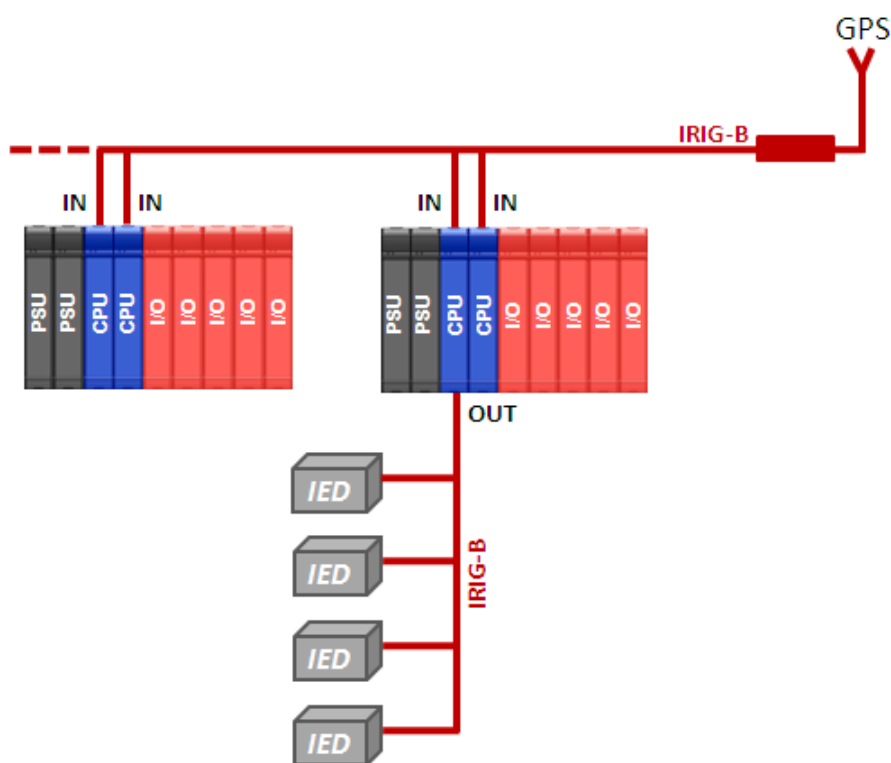


Figura 50: Exemplo de Sincronismo para Sistema Redundante

3.8.1. Terminais de Ligação do IRIG-B

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, conforme a tabela abaixo.

Nomenclatura	Borne da UCP
IRIG-B +IN	1
IRIG-B -IN	2
IRIG-B +OUT	3
IRIG-B -OUT	4

Tabela 32: Conexão dos Sinais IRIG-B na UCP HX3040

ATENÇÃO

Curto-circuitos prolongados nos conectores do IRIG-B podem danificar a saída do mesmo.

Na figura abaixo podemos visualizar a representação do conector IRIG-B da UCP HX3040.

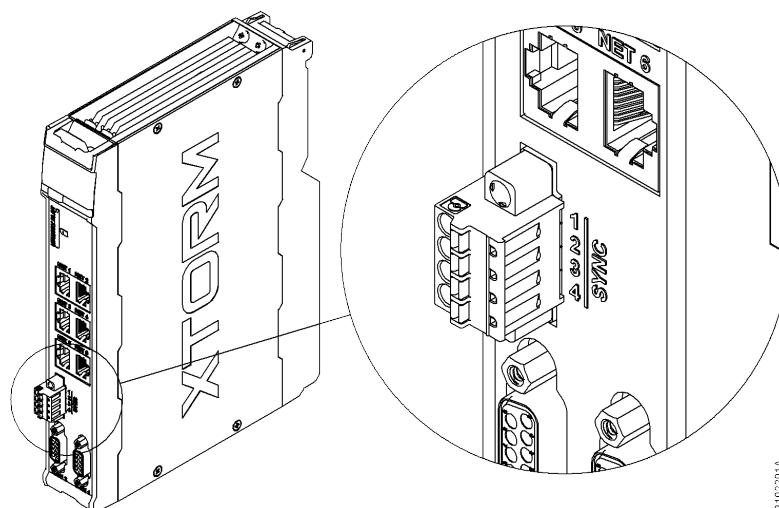


Figura 51: Conector de Interligação para o Sinal IRIG-B

3.9. Instalação do Cartão de Memória

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

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

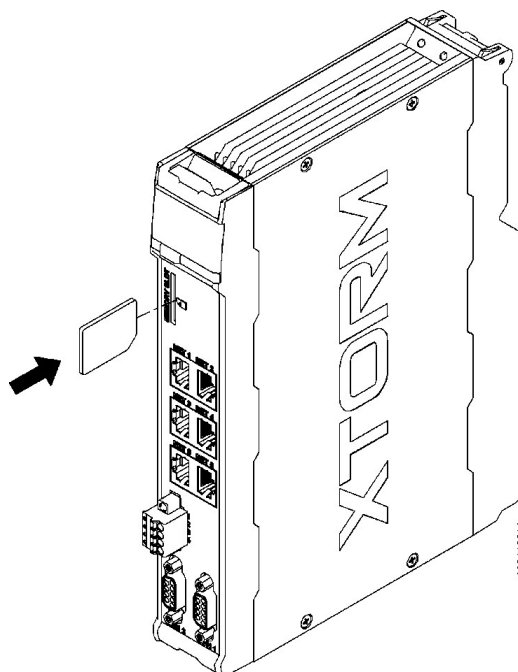


Figura 52: Inserção do Cartão de Memória na UCP

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, conforme mostra a figura abaixo, pois o mesmo estará solto.

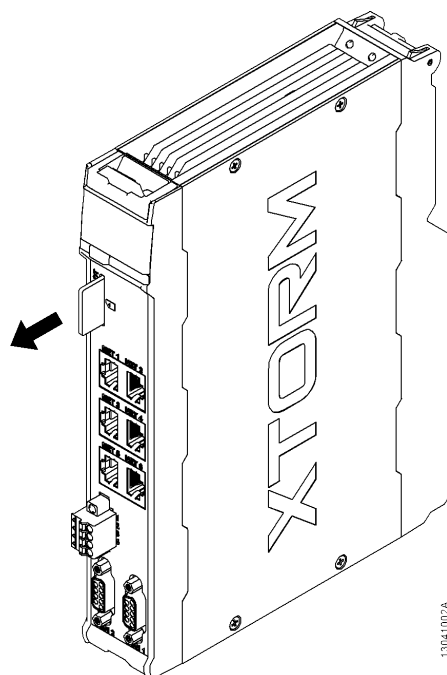


Figura 53: Retirada do Cartão de Memória

3.10. Instalação do Programador

Para realizar a instalação do software de desenvolvimento MasterTool Xtorm, é necessário ter em mãos o CD-ROM de distribuição ou efetuar o download do arquivo de instalação no site www.altus.com.br. Após, feche todos os programas que estejam em execução no seu computador e em seguida dê um duplo clique no arquivo de instalação. O instalador abrirá a seguinte tela de instalação:



Figura 54: Assistente de Instalação

A primeira tela indica o início da instalação e apresenta a versão do software MasterTool Xtorm que será instalada em seu computador. Clique em Next para continuar. A tela que se refere ao contrato de licença será mostrada, o qual deve ser lida atentamente.

Caso você concorde com os termos da licença, clique em *Avançar* para continuar.

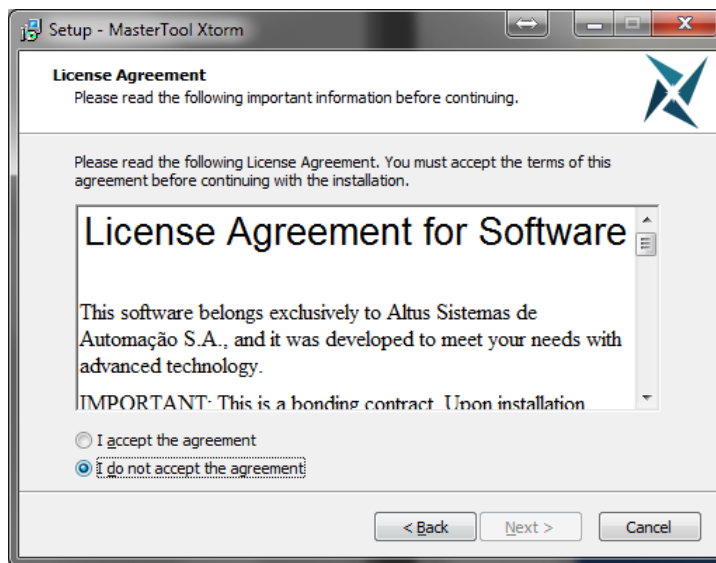


Figura 55: Tela de Licença

Na próxima tela é mostrado o programa que será instalado bem como a versão. Clique em *Next* para continuar.

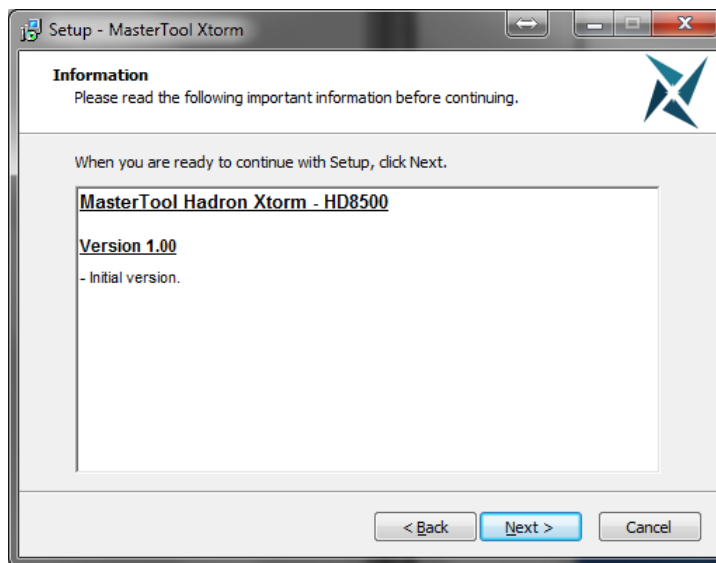


Figura 56: Seleção do Caminho de Instalação

Na próxima tela são apresentadas as opções de componentes que serão instalados. Clique em *Next* para continuar.

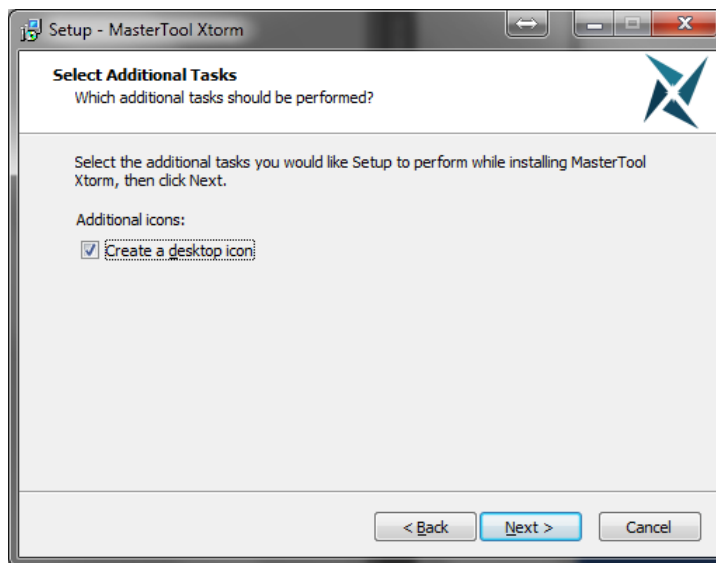


Figura 57: Seleção dos Componentes

Surgirá a tela que se refere a instalação ao nome da pasta onde serão mostrados os caminhos e localizações de todos itens a serem instalados. Após, clique em *Next* para continuar.

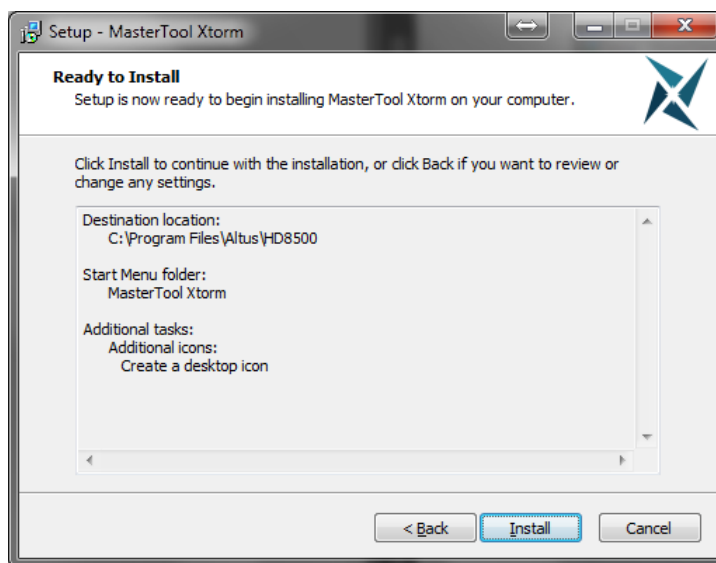


Figura 58: Seleção da Pasta

Nessa etapa a instalação do MasterTool Xtorm foi iniciada. Aguarde enquanto os arquivos necessários são instalados em seu computador, isso pode levar alguns minutos dependendo da configuração de seu computador.

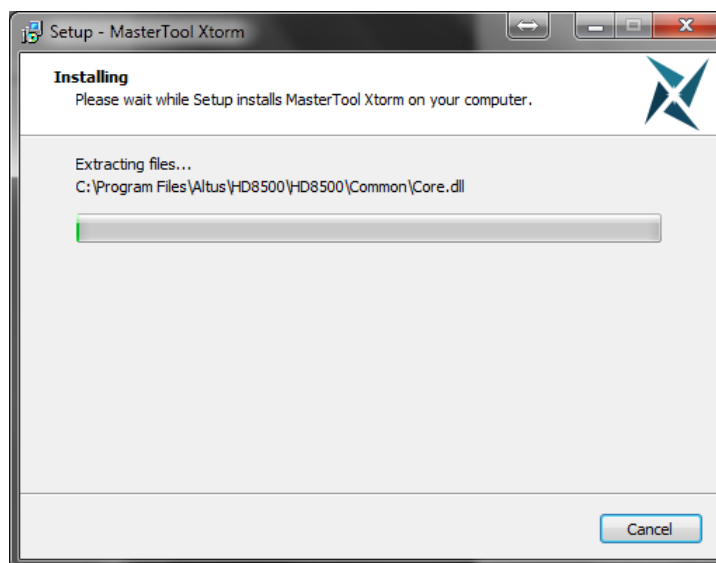


Figura 59: Processo de Instalação

Ao fim da instalação será perguntado se o usuário deseja reinicializar o computador para que as modificações da instalação sejam aplicadas:



Figura 60: Instalação Completa

O MasterTool Xtorm está instalado e pronto para ser utilizado. Para executá-lo, clique no atalho "MasterTool Xtorm" dentro do grupo "Altus" -> "MasterTool Xtorm", criado durante a instalação, no menu *Iniciar*.

Na primeira vez que o software for iniciado, uma tela solicitando as informações de registro será apresentada. Após preencher os campos corretamente, clique em *Confirm* para utilizar o MasterTool Xtorm.

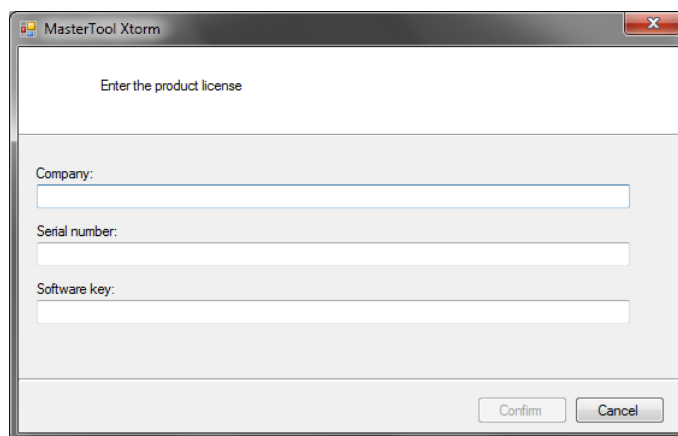


Figura 61: Tela de Registro

Para maiores informações sobre a instalação ou utilização do software, consultar o Manual de Programação IEC 61131 - MP399048.

4. Configuração

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.

4.1. Configuração da UCP

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

Além destes parâmetros, é possível alterar o nome 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.

As regras e orientações sobre tags, descrições e nome dos módulos de E/S podem ser consultadas na seção [Acessando a descrição do módulo e dos pontos de E/S](#).

ATENÇÃO

É recomendável que o nome de cada módulo possua 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 caracter ponto “.”. A utilização de qualquer outro caracter diferente dos acima citados não é recomendada.

ATENÇÃO

Quando utilizar a função *ETD – Electronic Tag on Display* para visualizar a descrição do módulo no visor gráfico, a descrição do módulo será trancada nos primeiros 48 caracteres da descrição dada para o módulo.

ATENÇÃO

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

4.1.1. Parâmetros Gerais

Os parâmetros gerais (General Parameters), encontrados na tela inicial de configuração da UCP, conforme mostra a figura abaixo, são definidos como:

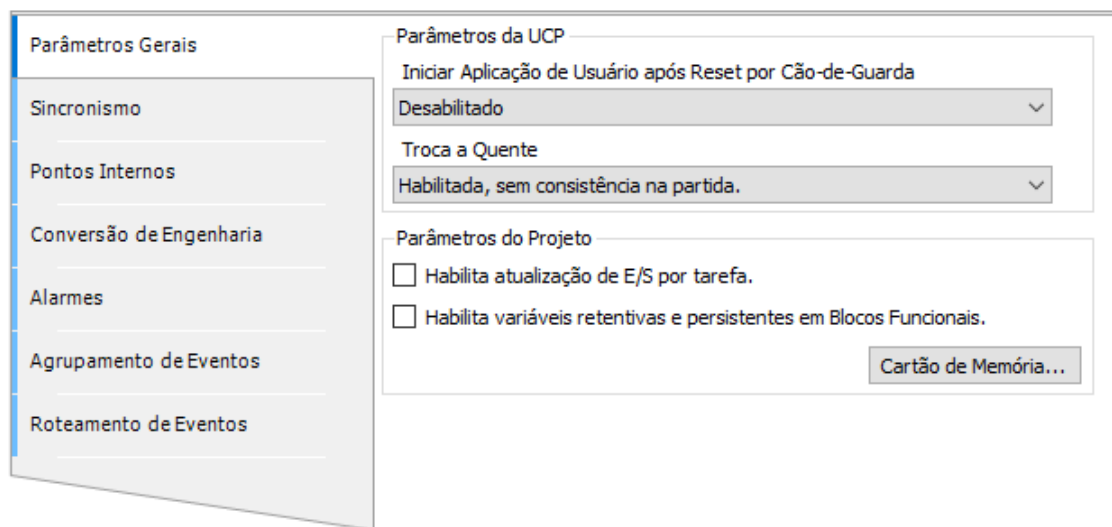


Figura 62: Tela de Configuração Parâmetros Gerais da UCP

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Iniciar aplicação de usuário após reset por Cão-de-guarda	Quando habilitado, inicia a aplicação do usuário após o reset do cão-de-guarda de hardware ou pela reinicialização do Runtime, porém mantém a indicação através do LED de diagnóstico (botão) e através das variáveis.	Desabilitado	Habilitado Desabilitado
Troca a Quente	Modo troca a quente dos módulos.	Habilitada, sem consistência na partida	- Desabilitada, apenas para módulos declarados: Configuração desabilitada para os módulos que foram declarados na configuração. - Habilitada, com consistência na partida apenas para módulos declarados: Configuração habilitada, com consistência na partida apenas para os módulos declarados na configuração. - Habilitada, sem consistência na partida: Configuração habilitada, sem consistência na partida
Habilitar atualização de I/O por tarefa	Quando habilitado, realiza atualização de entradas e saídas no contexto da tarefa em que estiverem sendo utilizados. Para mais informações, consultar capítulo Escaneamento de E/S.	Desabilitado	Habilitado Desabilitado

Tabela 33: Configurações Gerais da UCP

4.1.1.1. 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.

CUIDADO

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

Na troca a quente, o comportamento do sistema relacionado se modifica conforme a configuração definida pelo usuário, que apresenta as seguintes opções, conforme descrito na tabela 33.

- Habilitada, com Consistência na Partida Apenas para Módulos Declarados
- Desabilitada, Apenas para Módulos Declarados
- Desabilitada, sem Consistência na Partida

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

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

Tabela 34: Situações Anormais de Barramento

Para maiores informações sobre os diagnósticos correspondentes às situações descritas acima, consultar seção [Diagnósticos via Variáveis](#).

4.1.1.1.1. Troca a Quente Desabilitada, Apenas para Módulos Declarados

Nesta configuração, quando ocorre uma situação anormal de barramento (conforme a tabela 34) a UCP entra em Modo Stop em um tempo de até 2 segundos, quando o LED vermelho começa a piscar 4x (conforme a tabela 35). 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).

ATENÇÃO

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.

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

É considerada *partida* o intervalo entre a energização da UCP (ou comando de reset ou download de aplicação) até a primeira vez em que a mesma entra em modo Run. Esta configuração verifica se ocorreu alguma situação anormal de barramento durante a partida; em caso positivo, a UCP entra em Modo Stop e o LED DL começa a piscar 4x (conforme a tabela 35). 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 citadas na tabela anterior, o sistema continuará trabalhando normalmente e sinalizará o problema via diagnóstico.

ATENÇÃO

- Nesta configuração, quando ocorrer falta de alimentação (mesmo que temporária), comando Reset a Quente, comando Reset a Frio ou ter sido realizado o Download de uma nova aplicação, e algum módulo estiver em uma situação anormal de barramento; a UCP entrará em Modo Stop e o LED DL começa a piscar 4x (conforme a tabela 35), 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.

4.1.1.1.3. Troca a Quente Desabilitada, sem Consistência na Partida

Permite que o sistema entre em operação mesmo quando algum módulo estiver em uma situação anormal de barramento (conforme a tabela 34). As situações anormais são relatadas via diagnóstico, tanto durante, como após a partida.

ATENÇÃO

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

4.1.1.1.4. Como Realizar a Troca a Quente

CUIDADO

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

ATENÇÃO

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

O procedimento para a troca de módulos a quente está descrito nas seções [Remoção dos Módulos](#) e [Inserção dos Módulos](#).

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 (conforme a tabela 35), devido a alguma situação anormal de barramento (conforme a tabela 34); os módulos de saída colocam seus pontos em modo seguro de falha, ou seja, desligam o ponto (0 Vdc).

No caso dos módulos de saída digitais, caso o mesmo seja removido do barramento energizado, o estado lógico dos pontos não serão alterados. 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 o mesmo seja removido do barramento energizado, o estado lógico dos pontos permanecerão 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 o mesmo 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*.

ATENÇÃO

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

A tabela 35 relaciona as condições de barramento e o estado de operação e do LED DL da UCP Hadron Xtorm. Para maiores informações sobre os estados do LED de diagnóstico, consulte a seção [Diagnósticos via LED](#).

Condições	Habilitada, com Consistência na Partida Apenas para Módulos Declarados	Habilitada, sem Consistência na Partida	Desabilitada, Apenas para Módulos Declarados
Módulo não declarado	LED DL: Pisca 2x Azul Aplicação: Run	LED DL: Pisca 2x Azul Aplicação: Run	LED DL: Pisca 2x Azul Aplicação: Run
Módulo ausente	LED DL: Pisca 2x Azul Aplicação: Run	LED DL: Pisca 2x Azul Aplicação: Run	LED DL: Pisca 4x Vermelho Aplicação: Stop
Módulo ausente (condição partida)	LED DL: Pisca 4x Vermelho Aplicação: Stop	LED DL: Pisca 2x Azul Aplicação: Run	LED DL: Pisca 4x Vermelho Aplicação: Stop
Configuração incompatível	LED DL: Pisca 2x Azul Aplicação: Run	LED DL: Pisca 2x Azul Aplicação: Run	LED DL: Pisca 4x Vermelho Aplicação: Stop
Configuração incompatível (condição partida)	LED DL: Pisca 4x Vermelho Aplicação: Stop	LED DL: Pisca 2x Azul Aplicação: Run	LED DL: Pisca 4x Vermelho Aplicação: Stop
Endereço de slot duplicado	LED DL: Pisca 4x Vermelho Aplicação: Stop	LED DL: Pisca 4x Vermelho Aplicação: Stop	LED DL: Pisca 4x Vermelho Aplicação: Stop
Módulo não operacional	LED DL: Pisca 4x Vermelho Aplicação: Stop	LED DL: Pisca 4x Vermelho Aplicação: Stop	LED DL: Pisca 4x Vermelho Aplicação: Stop

Tabela 35: Relação entre Condições e Troca Quente

4.1.1.2. Cartão de Memória

Os parâmetros encontrados na tela de configuração da cartão de memória (figura 130), são definidos como:

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Copy Project from CPU to Memory Card	Copia o projeto da memória interna da UCP para o cartão de memória	Disabled	- Enabled: Configuração habilitada. - Disabled: Configuração desabilitada.
Password to Copy Project from CPU to Memory Card	Senha para copiar o projeto da memória interna da UCP para o cartão de memória	-	Senha de 6 dígitos (0 a 999999)
Copy Project from Memory Card to CPU	Copia o projeto do cartão de memória para a memória interna da UCP	Disabled	- Enabled: Configuração habilitada. - Disabled: Configuração desabilitada.
Password to Copy Project from Memory Card to CPU	Senha para copiar o projeto do cartão de memória para a memória interna da UCP	-	Senha de 6 dígitos (0 a 999999)

Tabela 36: Parâmetros de Configuração do Cartão de Memória

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 130.

4.1.2. Sincronização de Tempo

Para a sincronização de tempo, a UCP HX3040 utiliza o protocolo IRIG-B, SNTP (Simple Network Time Protocol), centros de controle (servidores DNP3 ou IEC 60870-5-104) ou PTP (Precision Time Protocol), 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), acessado através da UCP HX3040, na árvore de dispositivos:

The screenshot displays the configuration window for the HX3040 device, specifically the 'Synchronism' tab. The interface is divided into several sections for configuring different time synchronization protocols:

- General Configuration:** Time Zone is set to -03 : 00.
- IRIG-B Configuration:** The 'Enable IRIG-B' checkbox is unchecked. Priority is set to 1, and Time-out (s) is set to 3.
- Control Center Configuration:** Priority is set to 2.
- SNTP Configuration:** The 'Enable SNTP' checkbox is unchecked. Priority is set to 3, Ethernet Interface is 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, and IP Address of Second SNTP Server is 192 . 168 . 15 . 11.
- PTP Configuration:** The 'Enable PTP' checkbox is unchecked. Priority is set to 4, and Ethernet Interface is NET 1.

Figura 63: Configuração de Sincronismo de Tempo

4. CONFIGURAÇÃO

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Enable IRIG-B	Habilita o sincronismo pelo protocolo IRIG-B.	Disabled	Disabled Enabled
Time Zone	Fuso horário. Diferença de tempo entre o horário UTC (Universal Time Coordinated) e o horário local.	-03:00	-12:59 a +13:59
Priority (IRIG-B)	Define a prioridade da fonte de sincronismo por IRIG-B com relação às demais. Quando habilitado, o sincronismo pelo protocolo IRIG-B deve exercer prioridade sobre as demais fontes. Devido a isso, o campo está habilitado somente para leitura.	1	1 2 3 4
Time-Out (s)	Em caso de falha do sinal IRIG-B, define o tempo (em segundos) máximo esperado para a restauração do sinal antes de buscar outra fonte de sincronismo.	3	3 a 100
Priority (Control Center)	Define a prioridade da fonte de sincronismo por Centro de Controle com relação às demais. Quanto menor for o valor do campo, mais prioritária é a fonte de sincronismo.	2	1 2 3 4
Enable SNTP	Habilita o sincronismo pelo protocolo SNTP.	Disabled	Disabled Enabled
Priority (SNTP)	Define a prioridade da fonte de sincronismo por SNTP com relação às demais. Quanto menor for o valor do campo, mais prioritária é a fonte de sincronismo.	3	1 2 3 4
Ethernet Interface (SNTP)	Define a porta ethernet pela qual é feita a comunicação com o servidor SNTP usado para a sincronização do relógio.	NET 1	NET 1 NET 2 NET 3 NET 4 NET 5 NET 6
Period for SNTP Synchronization (s)	Período de tempo (em segundos) que deve ser aguardado entre duas solicitações de sincronismo.	60	1 a 255
Minimum Error Before Clock Update (x1 ms)	Antes de atualizar o horário, a UCP compara o horário recebido do servidor SNTP com o seu horário atual. O horário da UCP apenas será atualizado caso a diferença entre o horário recebido e horário atual seja maior que o tempo (em milissegundos) configurado nesse campo.	200	0 a 65535
Maximum Number of Retries	É o número de vezes que a UCP irá tentar estabelecer uma conexão com um dos servidores SNTP. Em caso de erro em todas as tentativas, no próximo período ela irá tentar se conectar com o outro servidor.	3	0 a 100
IP Address of First SNTP Server	Endereço IP do primeiro servidor SNTP a ser utilizado.	192.168.15.10	1.0.0.1 a 223.255.255.254

Configurações	Descrição	Padrão de Fábrica	Possibilidades
IP Address of Second SNTP Server	Endereço IP do segundo servidor SNTP a ser utilizado.	192.168.15.11	1.0.0.1 a 223.255.255.254
Enable PTP	Habilita o sincronismo pelo protocolo PTP.	Disabled	Disabled Enabled
Priority (PTP)	Define a prioridade da fonte de sincronismo por PTP com relação às demais. Quanto menor for o valor do campo, mais prioritária é a fonte de sincronismo.	4	1 2 3 4
Ethernet Interface (PTP)	Define a porta ethernet pela qual é feita a comunicação com o mestre PTP usado para a sincronização do relógio.	NET 1	NET 1 NET 2 NET 3 NET 4 NET 5 NET 6

Tabela 37: Configurações de Sincronismo de Tempo

Notas:

Time Zone: A configuração de fuso horário é utilizada para converter o horário local em horário UTC e vice-versa. Enquanto algumas fontes de sincronismo utilizam o horário local (IRIG-B, protocolo IEC 60870-5-104, função SetDateAndTime), outras utilizam o horário UTC (SNTP, protocolo DNP3). O horário UTC normalmente é utilizado para imprimir os eventos (protocolos DNP3 e IEC 60870-5-104), enquanto que o horário local é utilizado por outras funcionalidades da UCP (função GetDateAndTime, informação de data e hora do OTD).

Minimum Error Before Clock Update: Caso este parâmetro seja igual à zero, o relógio será atualizado em todas as requisições de sincronismo SNTP.

4.1.2.1. 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 microsegundos, 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.

ATENÇÃO

Alterações do fuso-horário em tempo de execução quando o método de sincronismo de tempo é IRIG-B não são recomendadas. Estas alterações podem causar descontinuidades em estampas de tempo, que por sua vez causariam um comportamento não desejado na geração de certos eventos.

4.1.2.2. 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. Para configurar o dispositivo com essa opção, consultar os parâmetros *Enable Synchronism* e *Synchronism Time-Out* disponíveis na seção [Configuração da Camada de Aplicação DNP3 Servidor](#), ou o parâmetro *Enable Time Synchronization* disponível na seção [Camada de Aplicação](#).

ATENÇÃO

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

Este método de sincronismo deve ser utilizado apenas como um método auxiliar de sincronismo, uma vez que a precisão do processo de sincronização do relógio 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.

ATENÇÃO

Em arquiteturas com redundância de UCP, os drivers DNP3 Servidor e IEC 60870-5-104 Servidor são desabilitados na UCP não ativa. Desta forma, não é recomendada a utilização destes métodos de sincronismo em sistemas redundantes, pois a UCP não ativa pode levar vários segundos após um switchover até que o seu relógio seja sincronizado. Em sistemas redundantes, recomenda-se a utilização de SNTP e/ou IRIG-B.

4.1.2.3. 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.

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

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

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

ATENÇÃO

- Dependendo da subrede do servidor SNTP, o cliente irá utilizar a interface Ethernet que esteja na subrede correspondente para fazer as requisições de sincronismo. Caso o servidor esteja fora das duas subredes das interfaces NET 1 e NET 2, a requisição poderá ser feita por qualquer uma das duas interfaces que possa achar uma rota para o servidor.

- O Serviço SNTP depende da aplicação do usuário apenas para a sua configuração. Portanto, este serviço vai ser executado mesmo quando a UCP estiver nos modos STOP ou BREAKPOINT, desde que exista uma aplicação na UCP com o cliente SNTP habilitado e corretamente configurado.

CUIDADO

Caso não seja utilizado o sincronismo por IRIG-B, é vital a configuração de no mínimo um servidor SNTP. O recomendável é configurar dois servidores SNTP (primário e secundário). O sincronismo SNTP é necessário para gerar eventos com estampa de tempo coerente entre as UCPs A e B e com a hora mundial. Outra utilidade é evitar descontinuidades durante um switchover em aplicações que referenciam data e hora, tendo em vista que não existe sincronismo de data e hora entre as UCPs através dos canais de sincronismo da redundância.

4.1.2.4. PTP

O PTP (Precision Time Protocol), foi implementado de acordo com a norma IEEE 1588, como fonte de tempo. Além disso, ele foi construído para atuar apenas no modo escravo e no modo de comunicação End-To-End (E2E).

Para isso é necessário que o usuário habilite o serviço na aba de sincronismo da UCP e configure sua prioridade juntamente da porta de comunicação com o mestre PTP.

O PTP disponível na Série Xtorm foi projetado a nível de software, por isso, apresenta uma precisão de no mínimo 1 ms e, por operar apenas através protocolo de transporte de Ethernet nível 2 (IEEE 802.3), não é necessário que a UCP esteja na mesma subrede do mestre PTP. Dessa maneira, para que seja possível estabelecer uma comunicação sem falhas, os pontos acima citados devem ser observados na configuração do mestre PTP que será utilizado em conjunto com o controlador.

Além disso, para que esse protocolo seja utilizado como fonte primária de sincronismo de tempo, é necessário que exista uma comunicação entre a interface configurada e um mestre PTP, e que essa fonte tenha a menor prioridades entre as disponíveis no momento. O controlador verifica ciclicamente o estado da comunicação entre a UCP e o mestre PTP a fim de rapidamente identificar alguma falha e informar o dispositivo para que a outra fonte de tempo assuma como ativa.

ATENÇÃO

O Serviço PTP depende da aplicação do usuário apenas para a sua configuração. Portanto, ele será executado mesmo quando a UCP estiver nos modos STOP ou BREAKPOINT, desde que exista uma aplicação na UCP com o PTP habilitado e corretamente configurado.

4.1.2.5. 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.

Para maiores informações, consultar a seção [Blocos Funcionais e Funções para Leitura e Escrita do RTC](#).

4.1.3. 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.

ATENÇÃO

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

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

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

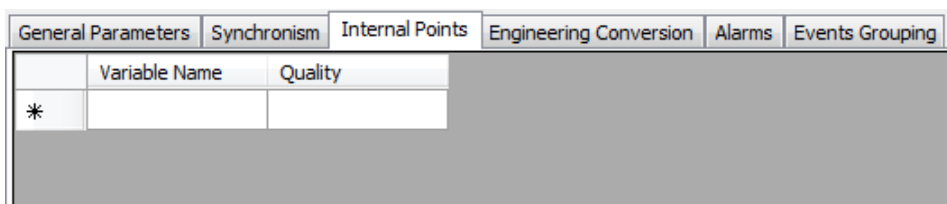


Figura 64: Tela de Configuração dos Pontos Internos

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

Tabela 38: Configuração dos Pontos Internos

A figura abaixo mostra um exemplo de configuração de um ponto interno:

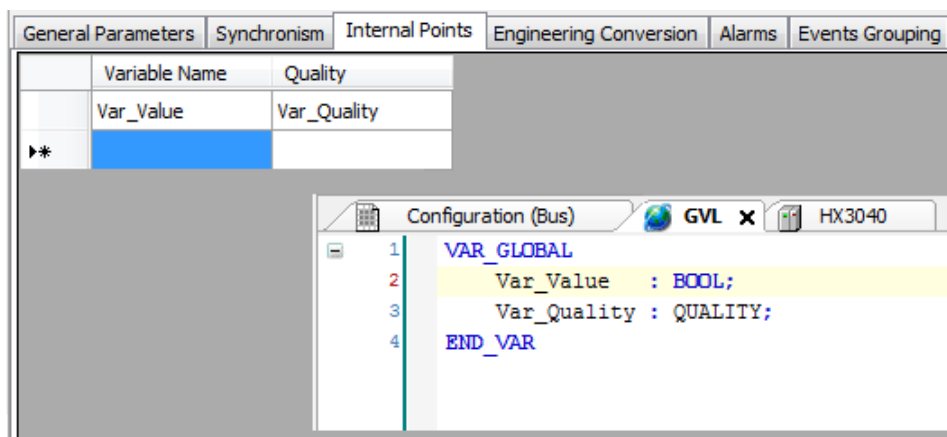


Figura 65: Exemplo de Configuração dos Pontos Internos

4.1.3.1. Conversões de Qualidade

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

As normas IEC 61850, DNP3 e 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.

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

ATENÇÃO

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

4. CONFIGURAÇÃO

4.1.3.1.1. Internal Quality

Esta é a estrutura QUALITY, a tabela abaixo mostra detalhadamente cada uma das suas Variáveis.

Bit	Nome	Tipo	Descrição
0	FLAG_RESTART	BOOL	A flag de RESTART indica que o dado não foi atualizado pelo campo desde o Reset do dispositivo.
1	FLAG_COMM_FAIL	BOOL	Indica que existe uma falha de comunicação no caminho entre o dispositivo onde o dado é originado e o dispositivo de relatórios.
2	FLAG_REMOTE_SUBSTITUTED	BOOL	Se TRUE, o valor dos dados é substituído em um dispositivo de comunicação remota.
3	FLAG_LOCAL_SUBSTITUTED	BOOL	Se TRUE, o valor do dado é substituído pelo dispositivo que gerou essa flag. Este comportamento pode ter ocorrido devido a um funcionamento em diagnóstico ou temporário devido à intervenção humana.
4	FLAG_FILTER	BOOL	Flag utilizada para sinalizar e prevenir a sobrecarga dos canais de comunicação de eventos, como oscilações (mudanças rápidas) nas entradas binárias.
5	FLAG_OVERFLOW	BOOL	Este identificador deve indicar um problema de qualidade de que o valor do atributo que a qualidade for associada está além da capacidade de representação.
6	FLAG_REFERENCE_ERROR	BOOL	Este identificador deve indicar que o valor pode não ser um valor correto devido à referência estar fora de calibração.
7	FLAG_INCONSISTENT	BOOL	Este identificador deve indicar que uma função de avaliação detectou uma inconsistência.
8	FLAG_OUT_OF_RANGE	BOOL	Este identificador deve indicar um erro de qualidade que o atributo ao qual a qualidade foi associada está além da capacidade de valores pré-definidos.
9	FLAG_INACCURATE	BOOL	Este identificador deve indicar que o valor não atende a precisão declarada da fonte.

Bit	Nome	Tipo	Descrição
10	FLAG_OLD_DATA	BOOL	Um valor parece estar desatualizado, caso uma atualização não ocorra em um período de tempo específico.
11	FLAG_FAILURE	BOOL	Este identificador deve indicar que uma função de supervisão detectou uma falha interna ou externa.
12	FLAG_OPERATOR_BLOCKED	BOOL	Atualização foi bloqueada pelo operador.
13	FLAG_TEST	BOOL	Test deve ser um identificador adicional que pode ser usado para classificar um valor sendo um valor teste e que não será usado para fins operacionais.
14-15	RESERVED	-	Reservado
16-17	VALIDITY	QUALITY_VALIDITY	0 – Good (Valor confiável, significa que não há condição anormal) 1 – Invalid (Valor não corresponde ao valor no IED) 2 – Reserved (Reservado) 3 – Questionable (Valor presente pode não ser o mesmo do IED)

Tabela 39: Estrutura QUALITY

4.1.3.1.2. Conversão DNP3

A tabela 40 ilustra a conversão de qualidade DNP3 para pontos internos, e a tabela 41 relaciona a conversão de qualidade de pontos internos para DNP3 da Série Hadron Xtorm disponíveis para HD8500.

DNP3-> Pontos Internos		
	Qualidade Interna	
DNP3 Quality	Flags	VALIDITY
ONLINE	-	Se apenas ONLINE, VALIDITY_GOOD
NOT ONLINE	FLAG_OLD_DATA	VALIDITY_QUESTIONABLE
RESTART	FLAG_RESTART	VALIDITY_QUESTIONABLE
COMM_LOST	FLAG_COMM_FAIL e FLAG_OLD_DATA	VALIDITY_QUESTIONABLE
REMOTE_FORCED	FLAG_REMOTE_SUBSTITUTED e FLAG_OPERATOR_BLOCKED	Se apenas REMOTE_FORCED e ONLINE, VALIDITY_GOOD
LOCAL_FORCED	FLAG_LOCAL_SUBSTITUTED e FLAG_OPERATOR_BLOCKED	Se apenas LOCAL_FORCED e ONLINE, VALIDITY_GOOD
CHATTER_FILTER	FLAG_FILTER	VALIDITY_QUESTIONABLE

4. CONFIGURAÇÃO

DNP3-> Pontos Internos		
	Qualidade Interna	
DNP3 Quality	Flags	VALIDITY
OVER_RANGE	FLAG_OVERFLOW e FLAG_OUT_OF_RANGE	VALIDITY_INVALID
DISCONTINUITY	FLAG_REFERENCE_ERROR	VALIDITY_QUESTIONABLE
REFERENCE_ERR	FLAG_INACCURATE e FLAG_REFERENCE_ERROR	VALIDITY_QUESTIONABLE

Tabela 40: Conversão DNP3 para Pontos Internos

Pontos Internos-> DNP3		
Qualidade Interna		
Flags	VALIDITY	Qualidade DNP3
FLAG_RESTART	ANY	RESTART
FLAG_COMM_FAIL	ANY	COMM_LOST
FLAG_REMOTE_SUBSTITUTED	ANY	REMOTE_FORCED
FLAG_LOCAL_SUBSTITUTED	ANY	LOCAL_FORCED
FLAG_FILTER	ANY	CHATTER_FILTER
FLAG_OVERFLOW	ANY	OVER_RANGE
FLAG_REFERENCE_ERROR	ANY	DISCONTINUITY (para contadores) REFERENCE_ERR (para analógicos)
FLAG_INCONSISTENT	ANY	not ONLINE
FLAG_OUT_OF_RANGE	ANY	OVER_RANGE
FLAG_INACCURATE	ANY	REFERENCE_ERR
FLAG_OLD_DATA	ANY	not ONLINE
FLAG_FAILURE	ANY	not ONLINE
FLAG_OPERATOR_BLOCKED e NOT LOCAL_SUBSTITUTED e/ou NOT REMOTE_SUBSTITUTED	ANY	LOCAL_FORCED
FLAG_TEST	ANY	not ONLINE
NOT FLAG_OLD_DATA e NOT FLAG_FAILURE e NOT FLAG_INCONSISTENT e NOT FLAG_TEST	VALIDITY_INVALID	INVALID

Tabela 41: Conversão Pontos Internos para DNP3

4.1.3.1.3. Conversão IEC 60870-5-104

As tabelas 42 à 44 apresentam respectivamente a conversão de qualidade de pontos digitais, analógicos, Step Position e contadores IEC 60870-5-104 para qualidade de pontos internos. As tabelas 45 à 47 apresentam a conversão de qualidade de pontos internos digitais, analógicos, Step Position, Bitstring e contadores para qualidade IEC 60870-5-104 da Série Hadron Xterm disponíveis para HD8500.

4. CONFIGURAÇÃO

IEC 60870-5-104 Digital-> Pontos Internos		
Qualidade Interna		
Qualidade IEC 60870-5-104	Flags	VALIDITY
BLOCKED	FLAG_OPERATOR_BLOCKED	Se apenas ONLINE, VALIDITY_QUESTIONABLE
SUBSTITUTED	FLAG_REMOTE_SUBSTITUTED	Se apenas SUBSTITUTED, VALIDITY_GOOD
NOT TOPICAL	FLAG_OLD_DATA	VALIDITY_QUESTIONABLE
INVALID	FLAG_FAILURE	VALIDITY_INVALID
Driver IEC 60870-5-104 iniciado, mas ainda não comunicou com sucesso	FLAG_RESTART	VALIDITY_INVALID
Falha de comunicação, sendo que já estava comunicando	FLAG_COMM_FAIL	VALIDITY_QUESTIONABLE

Tabela 42: Conversão IEC 60870-5-104 para Pontos Internos Digitais

IEC 60870-5-104 Analógico e Step Position -> Pontos Internos		
Qualidade Interna		
Qualidade IEC 60870-5-104	Flags	VALIDITY
OVERFLOW	FLAG_OVERFLOW	VALIDITY_INVALID
BLOCKED	FLAG_OPERATOR_BLOCKED	Se apenas BLOCKED, VALIDITY_QUESTIONABLE
SUBSTITUTED	FLAG_REMOTE_SUBSTITUTED	Se apenas SUBSTITUTED, VALIDITY_GOOD
NOT TOPICAL	FLAG_OLD_DATA	VALIDITY_QUESTIONABLE
INVALID	FLAG_FAILURE	VALIDITY_INVALID
Driver IEC 60870-5-104 iniciado, mas ainda não comunicou com sucesso	FLAG_RESTART	VALIDITY_INVALID
Falha de comunicação, sendo que já estava comunicando	FLAG_COMM_FAIL	VALIDITY_QUESTIONABLE

Tabela 43: Conversão IEC 60870-5-104 para Pontos Internos Analógicos e Step Position

IEC 60870-5-104 Contadores -> Pontos Internos		
Qualidade Interna		
Qualidade IEC 60870-5-104	Flags	VALIDITY
CARRY	FLAG_OVERFLOW	VALIDITY_GOOD
INVALID	FLAG_FAILURE	VALIDITY_INVALID
Driver IEC 60870-5-104 iniciado, mas ainda não comunicou com sucesso	FLAG_RESTART	VALIDITY_INVALID
Falha de comunicação, sendo que já estava comunicando	FLAG_COMM_FAIL	VALIDITY_QUESTIONABLE

Tabela 44: Conversão IEC 60870-5-104 para Pontos Internos de Contadores

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

Tabela 45: Conversão Pontos Internos para IEC 60870-5-104 de Pontos Digitais

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

4. CONFIGURAÇÃO

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

Tabela 46: Conversão Pontos Internos para IEC 60870-5-104 de Pontos Analógicos, Step Position e Bitstring

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

Tabela 47: Conversão Pontos Internos para IEC 60870-5-104 de Contadores

4.1.3.1.4. Conversão IEC 61850

A tabela abaixo relaciona o tipo de qualidade dos pontos internos com a qualidade IEC 61850 na Série Hadron Xtorm disponíveis para HD8500.

Pontos Internos -> IEC 61850		
Flags de Qualidade Interna	VALIDITY IEC 61850	Flags IEC 61850
FLAG_RESTART	VALIDITY_QUESTIONABLE	OLDDATE
FLAG_COMM_FAIL	VALIDITY_QUESTIONABLE	OLDDATE
FLAG_REMOTE_SUBSTITUTED	GOOD	Source = SUBSTITUTED
FLAG_LOCAL_SUBSTITUTED	GOOD	Source = SUBSTITUTED
FLAG_FILTER	VALIDITY_QUESTIONABLE ou VALIDITY_INVALID	OSCILLATORY

4. CONFIGURAÇÃO

Pontos Internos -> IEC 61850		
Flags de Qualidade Interna	VALIDITY IEC 61850	Flags IEC 61850
FLAG_OVERFLOW	VALIDITY_INVALID	OVERFLOW
FLAG_REFERENCE_ERROR	VALIDITY_QUESTIONABLE ou VALIDITY_INVALID	BADREFERENCE
FLAG_INCONSISTENT	VALIDITY_QUESTIONABLE	INCONSISTENT
FLAG_OUT_OF_RANGE	VALIDITY_QUESTIONABLE ou VALIDITY_INVALID	OUTOFRANGE
FLAG_INACCURATE	VALIDITY_QUESTIONABLE	INNACURATE
FLAG_OLD_DATA	VALIDITY_QUESTIONABLE	OLDDATE
FLAG_FAILURE	VALIDITY_INVALID	FAILURE
FLAG_OPERATOR_BLOCKED	GOOD	OPERATORBLOCKED
FLAG_TEST	GOOD	TEST

Tabela 48: Conversão Pontos Internos para IEC 61850

O processo de conversão do formato de qualidade interno para IEC 61850, segue as recomendações definidas pela norma IEC 61850. Segundo a norma o campo validity é definido em função das flags de qualidade. Algumas flags permitem que a validity assumam tanto o valor VALIDITY_QUESTIONABLE quanto VALIDITY_INVALID, nestes casos a validade assumida é a definida no ponto a ser convertido.

4.1.3.1.5. Qualidade Interna MODBUS

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

Qualidade Resultante	VALIDITY Resultante	Descrição
FLAG_RESTART	VALIDITY_INVALID	Valor de inicialização. O ponto nunca foi atualizado
No error flag (0)	VALIDITY_GOOD	Comunicação OK. O ponto está atualizado
FLAG_COMM_FAIL e FLAG_RESTART	VALIDITY_INVALID	Erro de comunicação. O ponto nunca foi atualizado
FLAG_COMM_FAIL e FLAG_OLD_DATA	VALIDITY_QUESTIONABLE	Um erro ocorreu mas o ponto estava atualizado e possui um valor antigo.
FLAG_FAILURE e FLAG_RESTART	VALIDITY_INVALID	Recebeu uma resposta de exceção, e o ponto ainda está com o valor inicial.
FLAG_FAILURE e FLAG_OLD_DATA	VALIDITY_QUESTIONABLE	Recebeu uma resposta de exceção, mas o ponto possui um valor antigo válido
FLAG_RESTART e FLAG_OLD_DATA	VALIDITY_QUESTIONABLE	Dispositivo está parado. O ponto contém um valor antigo

Tabela 49: Qualidade MODBUS

4.1.3.1.6. 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

4. CONFIGURAÇÃO

QUALITY, disponível na GVL IOQualities criada automaticamente pelo template do HD8500.

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

Para mais informações consulte a seção [GVL IOQualities](#).

Diagnóstico	Flags Resultantes	VALIDITY Resultante	Descrição
Don't care	FLAG_RESTART	VALIDITY_INVALID	A qualidade tem este valor antes de ser lida ou escrita pela primeira vez.
None	-	VALIDITY_GOOD	Comunicação OK. O ponto está atualizado.
None	FLAG_OLD_DATA e FLAG_FAILURE	VALIDITY_QUESTIONABLE	Módulo não operacional, entretanto dados foram lidos/escritos pelo menos uma vez.
bOverRange ou bUnderRange	FLAG_OUT_OF_RANGE	VALIDITY_INVALID	Valor da entrada está acima ou abaixo da faixa permitida.
bInputNotEnabled ou bOutputNotEnabled	FLAG_OPERATOR_BLOCKED	VALIDITY_INVALID	Entrada/saída não está habilitada.
bOpenLoop	FLAG_FAILURE	VALIDITY_INVALID	Entrada/saída em laço aberto.
bDisconnected	FLAG_FAILURE	VALIDITY_INVALID	Entrada/saída desconectada.
bHwFailure	FLAG_FAILURE	VALIDITY_INVALID	Entrada/saída com falha de hardware.
bFatalError	FLAG_FAILURE	VALIDITY_INVALID	Falha fatal de hardware
bNoExternalSupply	FLAG_FAILURE	VALIDITY_INVALID	Alimentação externa está abaixo do limite mínimo de funcionamento.
bShortCircuit ou bOutputShortCircuit	FLAG_FAILURE	VALIDITY_INVALID	Curto-circuito na saída.
bCalibrationError	FLAG_INACCURATE	VALIDITY_INVALID	Erro de calibração.
bColdJunctionSensorError	FLAG_INACCURATE	VALIDITY_INVALID	Erro no sensor de junta fria.
bWrongSequence	FLAG_FAILURE	VALIDITY_INVALID	Sequência de fases configurada difere da observada. Este diagnóstico afeta a qualidade de um subconjunto das entradas do módulo, cujo cálculo depende de uma sequência de fases correta.

Tabela 50: Qualidade dos Módulos de E/S

4.1.4. 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 onde 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, visualizadas na figura abaixo, segue os parâmetros descritos na tabela a seguir. 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 66: Tela de Configuração dos Parâmetros de Conversão de Engenharia

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Input	Variável simbólica de entrada da conversão.	-	Aceita variáveis simbólicas do tipo INT, DINT, UINT, UDINT, WORD e DWORD. A variável pode ser simples, array ou elemento de array e pode estar em estruturas.
Engineering	Variável simbólica que armazenará o resultado da conversão.	-	Aceita somente variáveis do tipo REAL. A variável pode ser simples, array ou elemento de array e pode estar em estruturas.
Minimum Input Value	Menor valor da entrada.	-	Constante numérica que deve obedecer aos limites do tipo da variável utilizada na "Entrada".
Maximum Input Value	Maior valor de entrada.	-	Constante numérica que deve obedecer aos limites do tipo da variável utilizada na "Entrada".
Minimum Engineering Value	Menor valor de engenharia.	-	Constante numérica que deve obedecer aos limites do tipo REAL.
Maximum Engineering Value	Maior valor de engenharia.	-	Constante numérica que deve obedecer aos limites do tipo REAL.

Tabela 51: Configuração dos Parâmetros de Conversão de Engenharia

4.1.5. 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, visualizadas na figura abaixo, segue os parâmetros descritos na a seguir. É possível a configuração de até 5120 entradas na tabela de Alarmes.

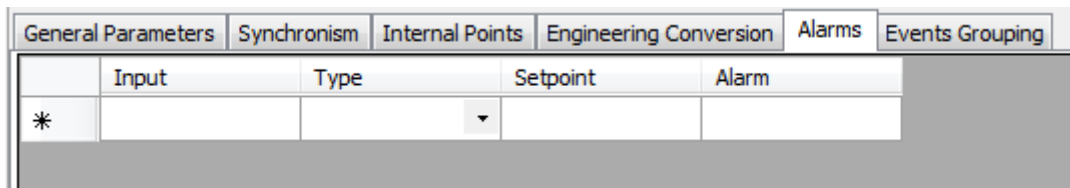


Figura 67: Tela de Configuração dos Alarmes

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Input	Variável simbólica que possui a informação utilizada para geração do alarme.	-	Aceita variáveis do tipo INT, DINT, LINT, UINT, UDINT, ULINT, WORD, DWORD, LWORD, REAL ou LREAL. A variável pode ser simples, array ou elemento de array e pode estar em estruturas.
Type	Tipo de comparação que será realizada entre a entrada e o setpoint.	-	Pode assumir dois valores predefinidos: "Maior que" ou "Menor que".
Setpoint	Valor de referência utilizado para identificar se o valor da entrada gerou ou não um alarme.	-	Aceita variáveis simbólicas do tipo INT, DINT, LINT, UINT, UDINT, ULINT, WORD, DWORD, LWORD, REAL ou LREAL. A variável pode ser simples, array ou elemento de array e pode estar em estruturas. São aceitas também constantes numéricas que respeitem os limites dos tipos LINT, LWORD e LREAL.
Alarm	Variável simbólica que receberá o valor VERDADEIRO ou FALSO, dependendo do resultado da comparação entre "Entrada" e "Setpoint".	-	Aceita somente variáveis do tipo BOOL. A variável pode ser simples, array ou elemento de array e pode estar em estruturas.

Tabela 52: Configuração dos Alarmes

4.1.6. Agrupamento de Eventos

É possível criar diversos agrupamentos de eventos em uma UTR Xtorm. Cada agrupamento de eventos permite resumir até 128 eventos digitais em um único e novo evento digital. Isso permite que um servidor configurado na UTR Xtorm reporte uma quantidade menor de eventos para determinados clientes. Por exemplo, um cliente local numa usina ou subestação pode estar interessado em todos os eventos de proteção que desarmam um disjuntor, enquanto um cliente remoto pode estar interessado apenas num resumo destes eventos de proteção.

Os eventos digitais agrupados podem ser derivados de:

- Entradas digitais locais, instaladas no barramento da UTR Xtorm;
- Entradas digitais remotas (eventos recebidos pela UTR Xtorm de servidores externos, através de drivers de comunicação clientes configurados na UTR Xtorm);
- Pontos digitais internos.

As entradas digitais ou pontos digitais internos, que compõem um agrupamento de eventos, podem ser simples (atributo valor do tipo BOOL) ou duplos (atributo valor do tipo DBP). Além disso, possuem um atributo de qualidade (variável do tipo

QUALITY). O evento resumo de cada agrupamento possui um atributo de valor do tipo BOOL, e um atributo de qualidade do tipo QUALITY.

A UTR Xtorm calcula automaticamente o valor, qualidade e estampa de tempo dos eventos resumo de todos os agrupamentos configurados. Para tanto, considera o valor, qualidade e estampa de tempo dos eventos digitais que compõem estes agrupamentos.

As próximas seções descrevem como os agrupamentos de eventos são configurados, e como são feitos os cálculos de valor, qualidade e estampa de tempo dos eventos resumo dos agrupamentos.

4.1.6.1. Configuração dos Agrupamentos de Eventos

A figura seguinte mostra um exemplo de tela contendo a configuração de dois agrupamentos de eventos.

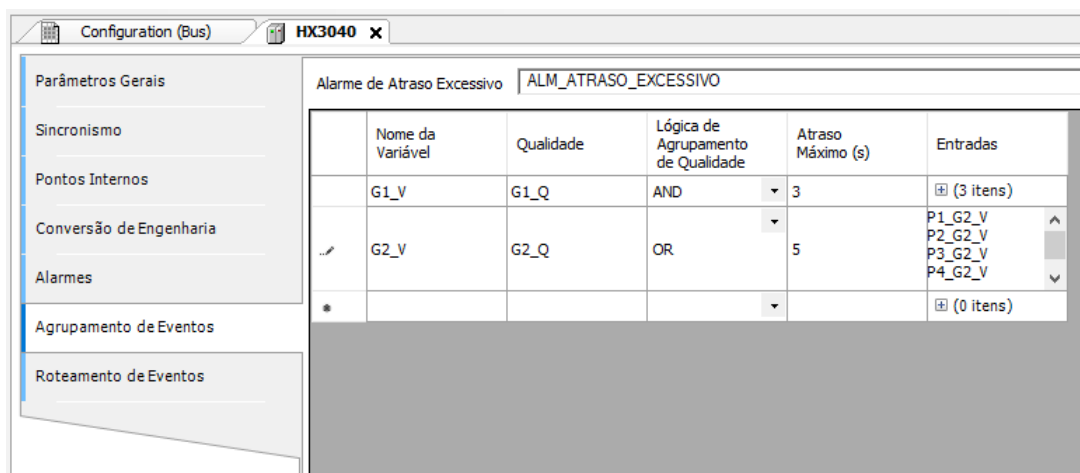


Figura 68: Tela de Configuração do Agrupamento de Eventos

Na parte superior da tela, deve-se configurar um “*Alarme de Atraso Excessivo*”, que é comum para todos os agrupamentos de eventos. Trata-se de uma variável do tipo BOOL que deve ser criada pelo usuário. No exemplo acima, trata-se da variável **ALM_ATRASO_EXCESSIVO**. Para maiores detalhes sobre a função desta variável, ver a seção [Utilização da Configuração de Atraso Máximo de um Agrupamento](#).

Depois deve ser preenchida uma tabela onde cada linha contém a configuração de um agrupamento de eventos. Cada linha contém 5 colunas:

1. A coluna “*Nome da Variável*” contém o nome de uma variável do tipo BOOL que corresponde ao atributo de valor do evento resumo. Ela deve ser criada, por exemplo, numa GVL. O cálculo deste atributo de valor é descrito na seção [Cálculo do Atributo Valor do Evento Resumo de um Agrupamento](#).

2. A coluna “*Qualidade*” contém o nome de uma variável do tipo QUALITY que corresponde ao atributo de qualidade do evento resumo. Ela deve ser criada, por exemplo, numa GVL. O cálculo deste atributo de qualidade é descrito na seção [Cálculo do Atributo Qualidade do Evento Resumo de um Agrupamento](#).

3. A coluna “*Lógica de Agrupamento de Qualidade*” permite selecionar entre dois métodos para o cálculo do atributo de qualidade do evento resumo (AND ou OR). Maiores detalhes na seção [Cálculo do Atributo Qualidade do Evento Resumo de um Agrupamento](#).

4. A coluna “*Atraso Máximo*” permite configurar um atraso que precisa ser utilizado no cálculo de valor e qualidade do evento resumo. O valor do atraso pode variar entre 1 e 60 segundos. Maiores detalhes na seção [Utilização da Configuração de Atraso Máximo de um Agrupamento](#).

5. A coluna “*Entradas*” define uma lista com até 128 variáveis digitais que compõe o agrupamento.

Tratam-se dos atributos de valor destas variáveis, que podem ter o tipo BOOL (variáveis simples) ou DBP (variáveis duplas). Para inserir diversas variáveis nesta coluna, deve-se utilizar as teclas **SHIFT+ENTER** depois de cada variável.

ATENÇÃO

O valor e qualidade resumo do agrupamento não devem ser confundidos com ponto internos. Não devem ser inseridos na aba “Pontos Internos” abaixo da CPU HX3040. Devem ser apenas criados (numa GVL ou POU) e configurados na aba “Agrupamento de Eventos” abaixo da CPU HX3040.

4.1.6.2. Utilização da Configuração de Atraso Máximo de um Agrupamento

Os diversos eventos que compõem um agrupamento podem chegar de servidores diferentes e ter atrasos de propagação diferenciados, de forma que podem chegar à UTR Xtorm fora de sequência cronológica. Para o cálculo correto dos eventos resumo, é essencial que os eventos que compõem cada agrupamento sejam avaliados em ordem cronológica. Isso impõe alguns requisitos:

- É essencial que a UTR Xtorm e todos os servidores que lhe reportam eventos estejam sincronizados com boa precisão.
- Quando um evento que compõe um agrupamento chegar à UTR Xtorm, sua idade não pode ser superior ao atraso máximo configurado para o agrupamento:
 - A idade de um evento corresponde à diferença entre o horário atual da UTR Xtorm e a estampa de tempo do evento.
 - Portanto, o usuário deve configurar um atraso máximo maior do que a soma dos retardos envolvidos desde que um evento é detectado num servidor até que ele chegue na UTR Xtorm (atrasos internos do servidor, atrasos de comunicação, etc.).
 - Caso a idade de um evento seja superior ao atraso máximo quando o evento chegar à UTR Xtorm, este evento é desprezado nos cálculos. Além disso, o “*Alarme de Atraso Excessivo*” é ativado durante um curto espaço de tempo. A ativação deste alarme indica que o cálculo do evento resumo pode estar incorreto.
 - Recomenda-se configurar um evento para a variável de “*Alarme de Atraso Excessivo*”, e reportar este evento para o mesmo cliente que recebe os eventos agrupados. Desta forma, este cliente saberá que podem haver cálculos incorretos de eventos agrupados, nos momentos próximos da estampa de evento deste alarme.
- É possível configurar o evento resumo de um agrupamento X como um evento que compõe outro agrupamento Y, fazendo uma espécie de encadeamento de eventos agrupados. Neste caso, a atraso máximo configurado para o agrupamento Y deve ser maior do que o atraso máximo configurado para o agrupamento X.
- O procedimento de cálculo de um evento resumo requer ordenar cronologicamente os eventos do agrupamento. Além disso, antes de processar um evento recebido que compõe o agrupamento, deve esperar até que a idade deste evento seja igual à configuração de atraso máximo do agrupamento.
 - Para executar este procedimento, a UTR Xtorm utiliza uma fila auxiliar com capacidade de 8192 eventos, compartilhada por todos os agrupamentos. Nesta fila, os eventos são ordenados e armazenados até atingir a idade necessária para serem processados (a configuração de atraso máximo).
 - Este tamanho de fila 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 eventos mais recentes serão descartados, mantendo os mais antigos.

4.1.6.3. Cálculo do Atributo Valor do Evento Resumo de um Agrupamento

O valor do evento resumo corresponde ao “ou lógico” dos valores dos eventos que compõem o agrupamento.

A figura seguinte mostra um exemplo de um agrupamento cujo valor do evento resumo é G, e cujos valores dos eventos componentes do agrupamento são P1, P2 e P3.



Figura 69: Cálculo do atributo valor do evento resumo

Observar que, neste exemplo, são gerados dois eventos para G:

- No primeiro evento, o valor de G sobe para 1 (TRUE), e recebe a estampa de tempo da subida do ponto P2 (o primeiro que sobe).
- No segundo evento, o valor de G desce para 0 (FALSE), e recebe a estampa de tempo da descida do ponto P1 (o último que desce).

Ou seja, o valor G é um “ou lógico” dos valores P1, P2 e P3.

Para um usuário que estiver monitorando os valores de P1, P2, P3 e G utilizando o Mastertool Xtorm, ou na IHM de um cliente conectado ao Xtorm, o diagrama de tempo da figura anterior não parece mostrar a realidade. O usuário deve considerar as seguintes observações:

- Monitorando com o Mastertool Xtorm, o valor G sobe depois que o ponto P2. O retardo pode chegar ao valor do atraso máximo configurado para o agrupamento, devido aos motivos explicados na seção [Utilização da Configuração de Atraso Máximo de um Agrupamento](#). Isso ocorre porque o valor G só é modificado para 1 depois que a idade do evento no ponto P2 for igual ao valor do atraso máximo. No entanto, a estampa de tempo atribuída ao evento de subida em G será a mesma do evento de subida no ponto P2.
- Monitorando com o Mastertool Xtorm, o valor G desce depois que o ponto P1. O retardo pode chegar ao valor do atraso máximo configurado para o agrupamento, devido aos motivos explicados na seção [Utilização da Configuração de Atraso Máximo de um Agrupamento](#). Isso ocorre porque o valor G só é modificado para 0 depois que a idade do evento no ponto P1 for igual ao valor do atraso máximo. No entanto, a estampa de tempo atribuída ao evento de descida em G será a mesma do evento de descida no ponto P1.

Algumas observações adicionais devem ser consideradas quando um evento que compõe o agrupamento é duplo (tipo DBP). O tipo DBP é uma estrutura que possui dois campos: OFF e ON, ambos do tipo BOOL.

- Caso OFF = TRUE e ON = FALSE, o valor do evento é considerado FALSE nos cálculos do agrupamento.
- Caso OFF = FALSE e ON = TRUE, o valor do evento é considerado TRUE nos cálculos do agrupamento.
- Os outros dois casos (OFF = ON = FALSE ou OFF = ON = TRUE) não são utilizados nos cálculos do agrupamento.

4.1.6.4. Cálculo do Atributo Qualidade do Evento Resumo de um Agrupamento

A qualidade do evento resumo de um agrupamento, e as qualidades dos eventos que compõem este agrupamento, são variáveis do tipo *QUALITY*.

Uma variável do tipo *QUALITY* é uma estrutura que possui dois campos: **VALIDITY** e **FLAGS**.

O campo **FLAGS** é uma estrutura que contém 16 campos do tipo BIT. Para a qualidade do evento resumo, cada um destes 16 bits é calculado como o ou lógico dos mesmos bits dos eventos que compõem do agrupamento.

O campo **VALIDITY** é uma enumeração que pode assumir um dos quatro seguintes valores:

- *VALIDITY_GOOD*
- *VALIDITY_QUESTIONABLE*
- *VALIDITY_INVALID*
- *VALIDITY_RESERVED* (este valor é reservado, e em princípio nunca deve ser observado)

Ao configurar o agrupamento, o usuário pode selecionar entre dois métodos na coluna “Lógica de Agrupamento de Qualidade”, e isso influi no cálculo do campo **VALIDITY**:

- *AND*
- *OR*

Funcionamento da lógica *AND* para o campo **VALIDITY**:

- Basta que um evento do agrupamento possua *VALIDITY_INVALID*, para que a qualidade resumo seja *VALIDITY_INVALID*.
- Caso nenhum evento do agrupamento possuir *VALIDITY_INVALID*, basta que um evento do agrupamento possua *VALIDITY_QUESTIONABLE*, para que a qualidade resumo seja *VALIDITY_QUESTIONABLE*.
- Caso nenhum evento do agrupamento possuir *VALIDITY_INVALID* ou *VALIDITY_QUESTIONABLE*, a qualidade resumo será *VALIDITY_GOOD*. Observar que *VALIDITY_RESERVED* é assumido como se fosse *VALIDITY_GOOD*.

Funcionamento da lógica *OR* para o campo **VALIDITY**:

- Basta que um evento do agrupamento possua *VALIDITY_GOOD*, para que a qualidade resumo seja *VALIDITY_GOOD*.
- Caso nenhum evento do agrupamento possuir *VALIDITY_GOOD*, basta que um evento do agrupamento possua *VALIDITY_QUESTIONABLE*, para que a qualidade resumo seja *VALIDITY_QUESTIONABLE*.
- Caso nenhum evento do agrupamento possuir *VALIDITY_GOOD* ou *VALIDITY_QUESTIONABLE*, a qualidade resumo será *VALIDITY_INVALID*. Observar que *VALIDITY_RESERVED* é assumido como se fosse *VALIDITY_INVALID*.

Ao calcular os campos **VALIDITY** e **FLAGS** do evento resumo, conforme as regras explicadas anteriormente, espera-se que a idade do evento componente do grupo atinja o valor configurado como “Atraso Máximo” do agrupamento, conforme explicado na seção [Utilização da Configuração de Atraso Máximo de um Agrupamento](#).

Um evento de mudança da qualidade do evento resumo ocorre quando existe alguma alteração no campo **VALIDITY** ou no campo **FLAGS** da qualidade do evento resumo. A estampa de tempo provém do evento componente do agrupamento que provocou esta alteração.

4.1.7. Roteamento de Eventos

O Roteamento de Eventos é uma funcionalidade que permite ao usuário copiar eventos que ocorrem em uma variável, para outra. Através da tabela Roteamento de Eventos, o usuário associa uma variável de entrada com uma variável de saída que receberá uma cópia do evento com mesmo valor, qualidade e estampa de tempo da variável de entrada. Ambas as variáveis precisam ser simples e de mesmo tipo.

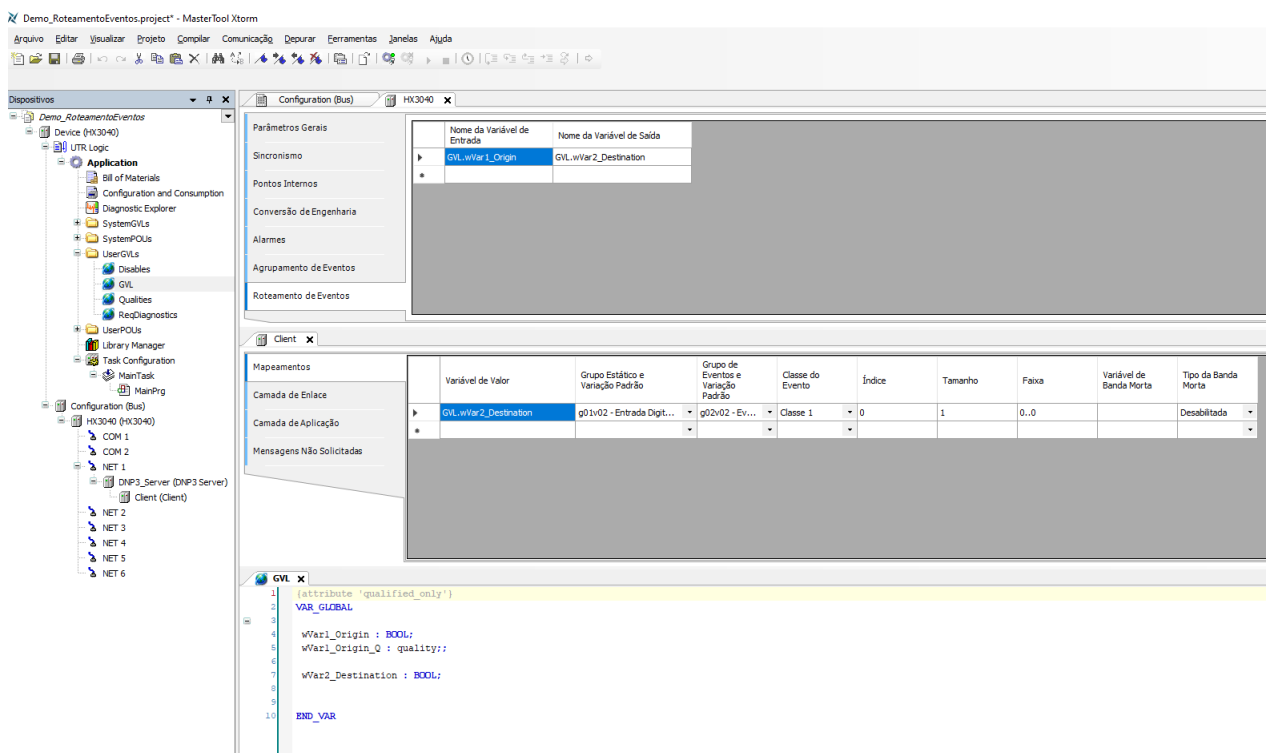


Figura 70: Roteamento de Eventos

O roteamento do evento para variável de saída somente ocorre quando a variável de entrada recebe um novo evento. Na condição de o valor ou a qualidade da variável de saída seja manipulada pelo usuário ou aplicação, através do MasterTool Xtorm, não será gerado evento para a mesma. O mesmo serve para variável de entrada, salvo em caso da mesma ser um ponto interno.

Caso a variável de saída seja mapeada em um servidor e este, por sua vez, não estiver conectado a um cliente, o evento gerado ficará registrado na fila de eventos da UCP.

4.2. Configuração das Interfaces Seriais

4.2.1. 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.

A tabela abaixo segue os parâmetros que devem ser configurados para o bom funcionamento da aplicação.

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.

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Tipo Serial	Configuração do tipo do canal serial.	RS-232C	RS-232C ou RS-485
Taxa de Transmissão	Velocidade da porta de comunicação serial.	115200	600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200 bps
Paridade	Configura a paridade da porta serial.	Sem Paridade	Ímpar Par Paridade sempre Um Paridade sempre Zero Sem Paridade
Bits de Dado	Configura o número de bits de dados em cada caractere da comunicação serial.	8	5, 6, 7 e 8
Bits de Parada	Configura os bits de parada da porta serial.	1	1, 1.5 e 2
Modo Serial	Configura o modo de operação da porta serial.	Modo Normal	- Modo Estendido: Modo estendido de operação da comunicação serial, no qual são fornecidas informações sobre o frame de dados recebido. - Modo Normal: Modo normal de operação da comunicação serial.

Tabela 53: Configurações da COM 1

Notas:

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

Um byte para o dado recebido (RX_CHAR : BYTE): Armazena os cinco, seis, sete ou oito bits dos dados recebidos, dependendo da configuração da serial.

Um byte para os erros no sinal (RX_ERROR : BYTE): Tem o seguinte formato:

- Bit 0: 0 - o caractere nos bits 0 a 7 é válido. 1 - o caractere nos bits 0 a 7 não é válido (ou pode não ser válido), devido aos problemas indicados nos bits 10 a 15.
- Bit 1: Não utilizado.
- Bit 2: Não utilizado.
- Bit 3: Erro de interrupção na UART. A entrada serial permaneceu na lógica 0 (paridade sempre zero) por um tempo maior do que um caractere (bit de partida + bit de dados + bit de paridade + bit de parada).
- Bit 4: Erro no frame UART. A lógica 0 (paridade sempre zero) foi lida quando o primeiro bit de parada era esperado, sendo que deveria ser lógica 1 (paridade sempre um).
- Bit 5: Erro de paridade UART. O bit de paridade lido não está de acordo com o bit de paridade calculado.
- Bit 6: Erro de overrun UART. Dados foram perdidos durante a leitura do FIFO UART, porque novos caracteres foram recebidos antes de os antigos serem removidos. Esse erro somente será indicado no primeiro caractere lido após a indicação do erro de overrun. Isso significa que alguns dados antigos foram perdidos.
- Bit 7: Erro de overrun na fila RX: Esse caractere foi escrito quando a fila RX foi completa, sobrescrevendo caracteres não lidos.

Dois bytes para o sinal de estampa de tempo (RX_TIMESTAMP : WORD): Indica o tempo de silêncio, no intervalo de 0 a 65535, usando como base de tempo 10 us. Satura em 655,35 ms, caso o tempo de silêncio seja maior do que 65535 unidades. O RX_TIMESTAMP de um caractere mede o tempo desde uma referência que pode ser uma das três opções abaixo:

- Na maioria dos casos, o final do caractere anterior.
- Configuração da porta serial.
- O fim da transmissão serial usando o SERIAL_TX FB, ou seja, quando o último caractere foi enviado na linha.

Além de medir o tempo de silêncio antes de cada caractere, o RX_TIMESTAMP também é importante, pois mede o tempo de silêncio do último caractere da fila RX.

A medição do silêncio é importante para implementar corretamente alguns protocolos, como MODBUS RTU. Esse protocolo especifica um inter-frame maior do que 3,5 caracteres e um inter-byte menor do que 1,5 caracteres.

Bits de Dado: A configuração Bits de Dado das interfaces seriais limita os campos Bits de Parada e Paridade da comunicação, ou seja, o número de bits de parada e o método de paridade irão variar de acordo com o número de bits de dado. A Tabela abaixo mostra as configurações permitidas para a interface COM 1.

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

Tabela 54: Configurações Específicas

4.2.1.1. Configurações Avançadas

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

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Parâmetros de Porta Avançados			
Sinais de modem	Realiza o controle de requisição para transmissão de um comando, através da interface RS-232C.	RTS Sempre Desligado	<ul style="list-style-type: none"> - RTS: Habilitado no início da transmissão e reiniciado, o mais rápido possível, após o final da transmissão. Ex: Controle de um conversor RS-232C/RS-485 externo. - RTS Sempre Desligado: Sempre desabilitado. - RTS Sempre Ligado: Sempre habilitado. - RTS/CTS: Caso o CTS esteja desabilitado, o RTS é habilitado. Então, aguarda-se o CTS ser habilitado para a transmissão começar e o RTS é reiniciado, o mais rápido possível, no final da transmissão. Ex: Controle de rádio modems com o mesmo sinal de modem. - RTS Manual: o usuário é responsável por controlar todos os sinais de controle.

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Parâmetros de Porta Avançados			
Threshold de RX da UART	Quantidade de bytes que devem ser recebidos para gerar uma nova interrupção na UART. Valores baixos fazem o Timestamp mais preciso quando o modo Estendido é utilizado e minimiza os erros de overrun. No entanto, valores baixos podem causar muitas interrupções que podem retardar a UCP.	8	1, 4, 8 e 14
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.	Desabilitado	- Habilitado: Configuração habilitada. - Desabilitado: Configuração desabilitada.
Evento RX DCD	Se habilitado gera um evento externo devido à mudança do sinal DCD.	Habilitado	- Habilitado: Configuração habilitada. - Desabilitado: Configuração desabilitada.
Evento RX CTS	Se habilitado gera um evento externo devido à mudança do sinal CTS.	Habilitado	- Habilitado: Configuração habilitada. - Desabilitado: Configuração desabilitada.

Tabela 55: Configurações Avançadas COM 1

Notas:

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

Evento RX DCD: Os eventos externos, como o sinal de DCD da COM 1, só podem ser associados às tarefas do perfil de projeto personalizado, descrito na seção [Perfil Personalizado](#) do capítulo de [Perfis de Projeto](#).

Evento RX CTS: Os eventos externos, como o sinal de CTS da COM 1, só podem ser associados às tarefas do perfil de projeto personalizado, descrito na seção [Perfil Personalizado](#) do capítulo de [Perfis de Projeto](#).

4.2.2. 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.

A tabela abaixo segue os parâmetros que devem ser configurados para o bom funcionamento da aplicação.

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.

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Tipo Serial	Configuração do tipo do canal serial.	RS-485	RS-232C ou RS-485
Taxa de Transmissão	Velocidade da porta de comunicação serial.	115200	600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200 bps
Paridade	Configura a paridade da porta serial.	Sem Paridade	Ímpar Par Paridade sempre Um Paridade sempre Zero Sem Paridade
Bits de Dado	Configura o número de bits de dados em cada caractere da comunicação serial.	8	5, 6, 7 e 8
Bits de Parada	Configura os bits de parada da porta serial.	1	1, 1.5 e 2
Modo Serial	Configura o modo de operação da porta serial.	Modo Normal	- Modo Estendido: Modo estendido de operação da comunicação serial, no qual são fornecidas informações sobre o frame de dados recebido (ver nota seção COM 1). - Modo Normal: Modo normal de operação da comunicação serial.

Tabela 56: Configurações da COM 2

A configuração Bits de Dado das interfaces seriais limita os campos Bits de Parada e Paridade da comunicação, ou seja, o número de bits de parada e o método de paridade irá variar de acordo com o número de bits de dado. A tabela abaixo mostra as configurações permitidas para a interface COM 2.

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

Tabela 57: Configurações Específicas

4.2.2.1. Configurações Avançadas

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

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

Tabela 58: Configurações Avançadas COM 2

4.3. Configuração das Interfaces Ethernet

A UCP Hadron Xtorm possui seis interfaces Ethernet locais denominadas NET 1 .. 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.

4.3.1. Interfaces Ethernet Locais

4.3.1.1. NET 1 .. 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.

A tabela abaixo descreve os parâmetros de configuração para as interfaces Ethernet:

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Habilitar	Habilita a interface Ethernet (somente para NET 2 .. NET 6).	Desabilitado	Desabilitado Habilitado
Endereço IP	Endereço IP do Controlador no barramento Ethernet.	192.168.15.1 (NET 1) 192.168.16.1 (NET 2) 192.168.17.1 (NET 3) 192.168.18.1 (NET 4) 192.168.19.1 (NET 5) 192.168.20.1 (NET 6)	1.0.0.1 a 223.255.255.254
Máscara de Subrede	Máscara de subrede do Controlador no barramento Ethernet.	255.255.255.0	0.0.0.1 a 255.255.255.254
Endereço do Gateway	Endereço do Gateway do Controlador no barramento Ethernet.	192.168.15.253 (NET 1) 192.168.16.253 (NET 2) 192.168.17.253 (NET 3) 192.168.18.253 (NET 4) 192.168.19.253 (NET 5) 192.168.20.253 (NET 6)	1.0.0.1 a 223.255.255.254

Tabela 59: Configurações das Interfaces NET 1 .. NET 6

ATENÇÃO

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.

4.3.2. Portas TCP Reservadas

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.

4.3.3. 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 abaixo, seguem os parâmetros descritos na tabela a seguir.

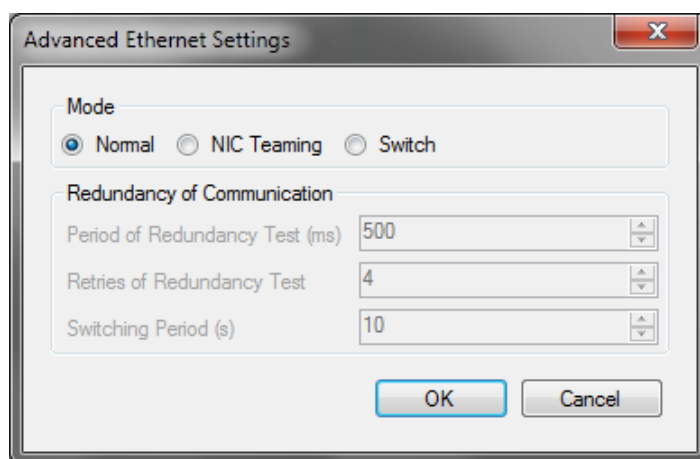


Figura 71: Tela de Configurações Avançadas das Interfaces Ethernet

Configurações	Descrição	Padrão de Fábrica	Possibilidades
Mode	Habilita o modo de funcionamento da Interface Ethernet.	Normal	Normal Nic Teaming Switch

Tabela 60: 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 o Modo NIC Teaming for selecionado, na mesma tela, automaticamente, outros parâmetros serão habilitados e deverão ser configurados:

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

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

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

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.

4.4. 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.

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

4.5. Configuração de Protocolos

A Hadron Xtorm oferece diversos protocolos de comunicação, incluindo MODBUS (exclusivamente simbólico), DNP3, IEC 60870-5-104 e outros. A tabela a seguir descreve os limites de configuração:

	HX3040
Pontos Mapeados	20000
Mapeamentos (Por Instância / Total)	5000 / 5000
Requisições	512
NETs – Instâncias Cliente ou Servidor (Por NET / Total)	4 / 12
COM (n) – Instâncias Mestre ou Escravo	1
Conexões DNP3	100
Conexões IEC 60870-5-104	100
Centros de Controle	5

Tabela 61: Limites dos Protocolos por UCP

Notas:

Pontos mapeados: É o número máximo de pontos mapeados que a UCP suporta. Lembrando que cada mapeamento suporta 1 ou mais pontos mapeados, dependendo do tamanho do dado, quando utilizado com variáveis do tipo ARRAY.

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

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

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

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

Conexões DNP3: É a soma do número máximo de conexões DNP3 Cliente e Servidor de uma UCP HX3040, não podendo exceder o número total de conexões informado na tabela para um correto funcionamento da aplicação.

Conexões IEC 60870-5-104: É a soma do número máximo de conexões IEC 60870-5-104 Cliente e Servidor de uma UCP HX3040, não podendo exceder o número total de conexões informado na tabela para um correto funcionamento da aplicação.

Centros de Controle: “Centro de Controle” é todo dispositivo cliente conectado à UCP através dos protocolos DNP3, IEC 60870-5-104 ou MMS (IEC 61850). Este campo informa a quantidade máxima de dispositivos clientes, do tipo centro de controle, suportada pela UCP. Corresponde a soma de todos os dispositivos clientes das instâncias de protocolos de comunicação DNP3, IEC 60870-5-104 e MMS Servidor (não inclui os mestres e clientes dos protocolos MODBUS RTU Escravo e MODBUS Servidor).

A tabela a seguir, informa os limites máximos de configuração e operação dos protocolos de comunicação, das interfaces Seriais e Ethernet.

	MODBUS RTU Mestre	MODBUS RTU Escravo	MODBUS Cliente	MODBUS Servidor	DNP3 Cliente	DNP3 Servidor	IEC 60870-5-104 Cliente	IEC 60870-5-104 Servidor
Dispositivos por instância	64	1 ⁽¹⁾	64	64 ⁽²⁾	32	5 ⁽³⁾	32	5 ⁽⁴⁾
Requisições por instância	2048	NA	2048	NA	128	NA	NA	NA
Requisições por dispositivo	32	NA	32	NA	32	NA	NA	NA
Requisições simultâneas por instância	1	1	128	64	32	5	160	5
Requisições simultâneas por dispositivo	1	1	8	64	1	1	5	1

Tabela 62: Limites dos Protocolos de Comunicação

Notas:**Dispositivos por instância:**

- **Protocolos Mestre ou Cliente:** Quantidade de dispositivos escravos ou servidores suportados por cada instância de protocolo Mestre ou Cliente.

- **Protocolo MODBUS RTU Escravo:** O limite ⁽¹⁾ informado diz respeito às interfaces seriais, que não permitem que um Escravo possa estabelecer comunicação, simultaneamente, através da mesma interface serial, com mais de um dispositivo Mestre. Não é necessário, e nem é possível, declarar nem configurar o dispositivo Mestre sob a instância do protocolo MODBUS RTU Escravo. O dispositivo Mestre terá acesso a todos os mapeamentos feitos diretamente na instância do protocolo MODBUS RTU Escravo.
- **Protocolo MODBUS Servidor:** O limite ⁽²⁾ informado diz respeito às interfaces Ethernet, que limitam a quantidade de conexões que podem ser estabelecidas com outros dispositivos através de uma mesma interface Ethernet. Não é necessário, e nem é possível, declarar nem configurar os dispositivos Clientes sob a instância do protocolo MODBUS Servidor. Todos os dispositivos Clientes terão acesso a todos os mapeamentos feitos diretamente na instância do protocolo MODBUS Servidor.
- **Protocolo DNP3 Servidor:** Quantidade de dispositivos clientes, do tipo centro de controle, suportados por cada instância de protocolo DNP3 Servidor. O limite ⁽³⁾ informado pode ser menor em função dos limites totais da UCP (consultar a Tabela 61).
- **Protocolo IEC 60870-5-104 Servidor:** Quantidade de dispositivos clientes, do tipo centro de controle, suportados por cada instância de protocolo IEC 60870-5-104 Servidor. O limite ⁽⁴⁾ informado pode ser menor em função dos limites totais da UCP (consultar a Tabela 61).

Requisições por instância: A soma das requisições configuradas em cada um dos dispositivos de uma instância Mestre ou Cliente não pode ultrapassar este limite.

Requisições por dispositivo: Quantidade de requisições, como por exemplo de leitura ou escrita de holding registers, que podem ser configuradas para cada um dos dispositivos (escravos ou servidores) de instâncias de protocolos Mestre ou Cliente. Para o protocolo DNP3 Cliente, este limite refere-se às requisições customizadas, configuradas via MasterTool. Este parâmetro não é aplicável (NA) às instâncias de protocolos Escravo ou Servidor, bem como ao protocolo IEC 61870 5 104 Cliente, cujas requisições customizadas devem ser declaradas e gerenciadas pela própria aplicação do usuário através de Blocos Funções.

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

Requisições simultâneas por dispositivo: Quantidade de requisições que podem ser transmitidas simultaneamente para cada dispositivo MODBUS Servidor, ou que podem ser recebidas simultaneamente de cada dispositivo MODBUS Cliente. Dispositivos MODBUS RTU, Mestre ou Escravo, bem como dispositivos DNP3, Cliente ou Servidor, e IEC 60870-5-104 Servidor, não suportam requisições simultâneas. Dispositivos IEC 60870-5-104 Cliente suportam o envio de requisições simultâneas, desde que para setores distintos.

ATENÇÃO

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

4.5.1. Comportamento dos Protocolos x Estado da UCP

A tabela abaixo mostra detalhadamente cada comportamento assumido para cada protocolo de comunicação existente na UCP HX3040 em todos seus estados de funcionamento, uma legenda especificando maiores informações é apresentada na tabela.

Protocolo	Tipo	Estado operacional da UCP					
		STOP			RUN		
		Depois do download antes da aplicação iniciar	Depois de a aplicação ir para STOP (PAUSE)	Depois de uma exceção	Não redundante ou Ativo	Redundante em Reserva	Após um break-point na MainPrg
MODBUS	Slave/Server	x / ✓	x / ✓	x / ✓	✓	x	✓
	Master/Client	x	x	x	✓	x	x
DNP3	Outstation/Server	x	x	x	✓	x	✓
	Master/Client	x	x	x	✓	x	✓
IEC 60870-5-104	Server	x	x	x	✓	x	✓
	Client	x	x	x	✓	x	✓
MMS	Server	x	x	x	✓	x	✓
GOOSE	Publisher	x	x	x	✓	x	✓
	Subscriber	x	x	x	✓	x	✓
SNTP	Client	✓	✓	✓	✓	✓	✓
PTP	Slave	✓	✓	✓	✓	✓	✓
HTTP	Server	✓	✓	✓	✓	✓	✓
IRIG-B	Transmitter	✓	✓	✓	✓	✓	✓
	Receiver	✓	✓	✓	✓	✓	✓

Tabela 63: Comportamento dos Protocolos x Estados da UCP

Notas:

Símbolo ✓: O protocolo permanece ativo e operando normalmente.

Símbolo x: O protocolo está desabilitado.

Símbolo x / ✓: Para que o protocolo comunique nas condições em que a UCP não está em RUN é necessário que seja marcada a opção “Mantém a comunicação funcionando quando a UCP está parada”. Para mais detalhes, veja a seção [Parâmetros Gerais do Protocolo MODBUS Escravo](#).

4.5.2. Fila de Eventos da UCP

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

Todos os eventos de pontos de comunicação gerados 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 abaixo descreve este funcionamento, incluindo exemplo de alguns consumidores e produtores de eventos:

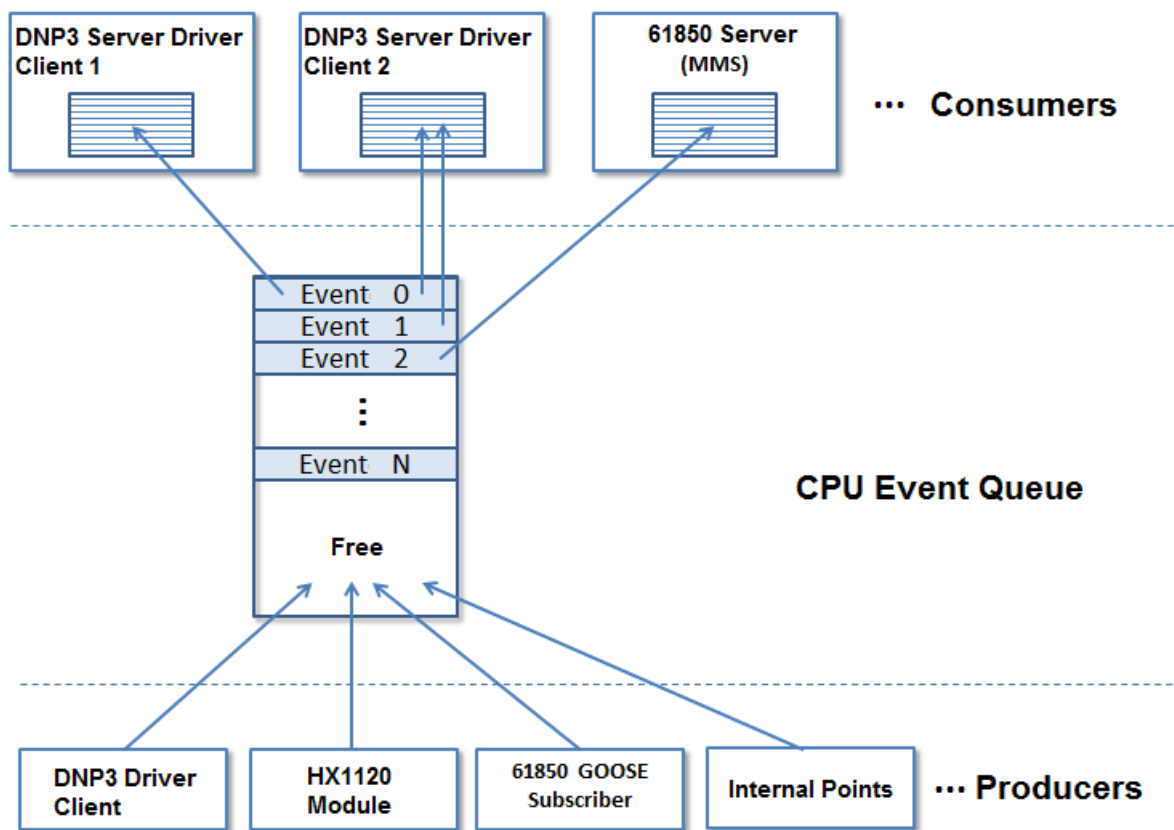


Figura 72: Fila de Eventos da UCP

4.5.2.1. 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.

4.5.2.2. 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 IEC 61850 Servidor), 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.

4.5.2.2.1. Sinalização de Overflow

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

- 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.
- 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:

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

4.5.2.3. 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:

HX1100 e HX1120: 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).

4.5.3. Intercepção 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 intercepção de comandos é implementada pelo bloco de função *CommandReceiver*. Os parâmetros de entrada e saída são descritos nas tabelas a seguir:

Parâmetro	Tipo	Descrição
bExec	BOOL	Quando TRUE, executa a interceptação do comando.
bDone	BOOL	Indica que os dados de saída do comando já foram processados, liberado o bloco de função para recepção de um novo comando.
dwVariableAddr	DWORD	Endereço da variável que receberá o comando.
eCommandResult	ENUM	Ação de entrada definida pelo usuário a partir da seguinte lista: SUCCESS(0); NOT_SUPPORTED(1); BLOCKED_BY_SWITCHING_HIERARCHY(2); SELECT_FAILED(3); INVALID_POSITION(4); POSITION_REACHED(5) PARAMETER_CHANGE_IN_EXECUTION(6); STEP_LIMIT(7); BLOCKED_BY_MODE(8); BLOCKED_BY_PROCESS(9); BLOCKED_BY_INTERLOCKING(10); BLOCKED_BY_SYNCHROCHECK(11); COMMAND_ALREADY_IN_EXECUTION(12); BLOCKED_BY_HEALTH (13); ONE_OF_N_CONTROL(14); ABORTION_BY_CANCEL(15); TIME_LIMIT_OVER(16); ABORTION_BY_TRIP(17); OBJECT_NOT_SELECTED(18); OBJECT_ALREADY_SELECTED(19); NO_ACCESS_AUTHORITY(20); ENDED_WITH_OVERSHOOT(21); ABORTION_DUE_TO_DEVIATION(22); ABORTION_BY_COMMUNICATION_LOSS(23); BLOCKED_BY_COMAND(24); NONE(25); INCONSISTENT_PARAMETERS(26); LOCKED_BY_OTHER_CLIENT(27); HARDWARE_ERROR(28); UNKNOWN(29);
dwTimeout	DWORD	Time-out definido para o usuário, é um parâmetro de entrada para a função.

Tabela 64: Parâmetros de Entrada do Bloco de Função CommandReceiver

Notas:

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

bDone: Após a interceptação do comando, o usuário será responsável por tratá-lo. Ao fim do tratamento, esta entrada deve ser ligada para que um novo comando possa ser recebido. Caso esta entrada não seja ligada antes da passagem do tempo definido na entrada **dwTimeout**, o servidor enviará uma confirmação negativa para o comando no protocolo. Enquanto a entrada **bDone** não for ligada, não será possível receber outros comandos para a variável especificada pela entrada **dwVariableAddr**. Se outros comandos forem recebidos para esta variável, o servidor enviará imediatamente uma confirmação negativa para o comando no protocolo.

eCommandResult: Resultado do tratamento do comando, interceptado pelo usuário. O resultado retornado ao cliente que

enviou o comando, o qual deve ser atribuído juntamente com a entrada *bDone*, é convertido para o formato do protocolo do qual foi recebido o comando. Na interceptação do protocolo IEC 60870-5-104, qualquer retorno diferente de *SUCCESS* resulta em uma confirmação negativa no protocolo.

ATENÇÃO

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

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

Tabela 65: Parâmetros de Saída do Bloco de Função *CommandReceiver***Notas:**

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

Os comandos suportados são descritos na tabela abaixo:

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

Tabela 66: Comandos Suportados do Bloco de Função *CommandReceiver*

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

Parâmetro	Tipo	Descrição
sSelectConfig	STRUCT	Configuração do comando de seleção recebido. Os parâmetros desta estrutura estão descritos na Tabela 68
sValue	STRUCT	Valor recebido em um select, quando recebido um comando de seleção com valor. Os parâmetros desta estrutura estão descritos na Tabela 71

Tabela 67: Parâmetros *sSelectParameters*

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

Tabela 68: Parâmetros sSelectConfig

Parâmetro	Tipo	Descrição
sSelectConfig	STRUCT	Configuração do comando de seleção recebido. Os parâmetros desta estrutura estão descritos na Tabela 70
sValue	STRUCT	Campo de valor referente ao comando de operação recebido. Os parâmetros desta estrutura estão descritos na Tabela 71

Tabela 69: Parâmetros sOperateParameters

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

Tabela 70: Parâmetros sOperateConfig

Parâmetro	Tipo	Descrição
eParamType	ENUM	Informa o tipo de comando que foi recebido: NO_COMMAND(0) SINGLE_POINT_COMMAND(1) DOUBLE_POINT_COMMAND(2) INTEGER_STATUS_COMMAND(3) ENUMERATED_STATUS_COMMAND(4) ANALOGUE_VALUE_COMMAND(5)
sSinglePoint	STRUCT	Quando recebido um comando, em função do tipo do comando recebido, definido por eParamType, é preenchida a estrutura de dados correspondente. Os parâmetros destas estruturas são apresentadas na Tabela 72 a Tabela 76
sDoublePoint	STRUCT	
sIntegerStatus	STRUCT	
sEnumeratedStatus	STRUCT	
sAnalogueValue	STRUCT	

Tabela 71: Parâmetros sValue

Parâmetro	Tipo	Descrição
bValue	BOOL	Valor de operação do ponto.
sPulseConfig	STRUCT	Os parâmetros de configuração de um comando pulsado são armazenados nesta estrutura. Os parâmetros desta estrutura são apresentados na Tabela 77

Tabela 72: Parâmetros sSinglePoint

Parâmetro	Tipo	Descrição
bValue	BOOL	Valor de operação do ponto.
sPulseConfig	STRUCT	Os parâmetros de configuração de um comando pulsado são armazenados nesta estrutura. Os parâmetros desta estrutura são apresentados na Tabela 77

Tabela 73: Parâmetros sDoublePoint

Parâmetro	Tipo	Descrição
diValue	DINT	Valor de operação do ponto.

Tabela 74: Parâmetros sIntegerStatus

Parâmetro	Tipo	Descrição
dwValue	DWORD	Valor de operação do ponto.

Tabela 75: Parâmetros sEnumeratedStatus

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

Tabela 76: Parâmetros sAnalogueValue

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

Tabela 77: Parâmetros sPulseConfig

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 busca-se 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*.

Abaixo, segue um exemplo de aplicação do interceptador de comandos, utilizando a linguagem ST, que encaminha 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.

Para estimular este código exemplo 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.

O exemplo utiliza ainda 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).

```

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.sPulseConfig.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.sPulseConfig.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

```

4.5.4. MODBUS - Tipos de Dados

A tabela abaixo mostra o tipo de variável suportada pela HX3040 para cada um dos tipos de dados do protocolo, aplicável para todos os dispositivos MODBUS.

Tipo de Relação	Paridade
Coil	BOOL
Input Status	BOOL
Holding Registers	INT, UINT, WORD, REAL, DINT, UDINT, DWORD, LREAL, LINT, ULINT ou LWORD
Input Registers	INT, UINT, WORD, REAL, DINT, UDINT, DWORD, LREAL, LINT, ULINT ou LWORD

Tabela 78: Declaração de Variáveis para MODBUS

Nota:

Para escrita mascarada de Holding Register somente os tipos INT, UINT ou WORD podem ser usados. As variáveis do tipo REAL, DINT, UDINT e WORD possuem tamanho 2. Já as variáveis do tipo LREAL, LINT, ULINT e LWORD possuem tamanho 4. As demais variáveis possuem tamanho 1. No caso de mapeamentos com arrays, a quantidade de elementos dos mesmos será multiplicado pelo tamanho do tipo de dado informado anteriormente.

4.5.5. 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).

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

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. Para realizar esse procedimento, consultar a 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, como: tempos de atraso de envio e interframe mínimo.
5. Adicionar e configurar dispositivos, definindo endereço do escravo, time-out 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.

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

4.5.5.1. Parâmetros Gerais do Protocolo MODBUS Mestre

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

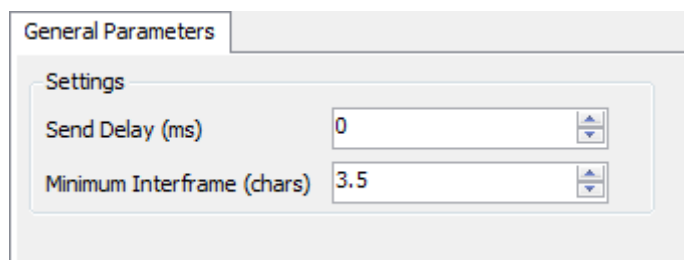


Figura 73: Tela de Configuração Parâmetros Gerais MODBUS RTU Mestre

Configuração	Descrição	Padrão de Fábrica	Possibilidades
Send Delay (ms)	Tempo de atraso para envio da resposta.	0	0 a 65535
Minimum Interframe(chars)	Tempo mínimo de silêncio entre diferentes frames.	3.5	3.5 a 100.0

Tabela 79: Configurações Gerais MODBUS RTU Mestre

Nota:

Send Delay: 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: 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.

ATENÇÃO

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 as quais estão descritas na tabela abaixo e a estrutura de diagnósticos das requisições do tipo T_DIAG_MODBUS_RTU_MAPPING_1, as quais estão descritas na tabela 87.

Variável de diagnóstico do tipo T_DIAG_MODBUS _RTU_MASTER_1.*	Tamanho	Descrição
Bits de diagnóstico:		
tDiag. bRunning	BIT	O mestre está em execução.
tDiag. bNotRunning	BIT	O mestre não está em execução (ver bit: bInterruptedByCommand).
tDiag. bInterruptedByCommand	BIT	O bit bNotRunning foi habilitado, pois o mestre foi interrompido pelo usuário atra- vés de bits de comando.
tDiag. bConfigFailure	BIT	Diagnóstico descontinuado.
tDiag. bRXFailure	BIT	Diagnóstico descontinuado.
tDiag. bTXFailure	BIT	Diagnóstico descontinuado.
tDiag. bModuleFailure	BIT	Indica se há falha no módulo ou o módulo não está presente.
tDiag. bDiag_7_reserved	BIT	Reservado
Códigos de erro:		
eErrorCode	SERIAL_ STATUS (BYTE)	Informa a possível causa de erros. Ver ta- bela 81 para maiores informações.
Bits de comando, reiniciados automaticamente:		
tCommand. bStop	BIT	Parar o mestre.
tCommand. bRestart	BIT	Reiniciar o mestre.
tCommand. bResetCounter	BIT	Reiniciar as estatísticas dos diagnósticos (contadores).
tCommand. bDiag_19_reserved	BIT	Reservado
tCommand. bDiag_20_reserved	BIT	Reservado
tCommand. bDiag_21_reserved	BIT	Reservado
tCommand. bDiag_22_reserved	BIT	Reservado
tCommand. bDiag_23_reserved	BIT	Reservado
byDiag_3_reserved	BYTE	Reservado
Estatísticas de comunicação:		
tStat. wTXRequests	WORD	Contador de requisições transmitidas pelo mestre (0 a 65535).
tStat. wRXNormalResponses	WORD	Contador de respostas normais recebidas pelo mestre (0 a 65535).

Variável de diagnóstico do tipo T_DIAG_MODBUS RTU_MASTER_1.*	Tamanho	Descrição
tStat. wRXExceptionResponses	WORD	Contador de respostas com códigos de exceção recebidas pelo mestre (0 a 65535).
tStat. wRXIllegalResponses	WORD	Contador de respostas ilegais recebidas pelo mestre – sintaxe inválida, número insuficiente de bytes recebidos, CRC inválido – (0 a 65535).
tStat. wRXOverrunErrors	WORD	Contador de erros de overrun durante a recepção – UART FIFO ou fila RX – (0 a 65535).
tStat. wRXIncompleteFrames	WORD	Contador de respostas com erro de construção, paridade ou falha durante a recepção (0 a 65535).
tStat. wCTSTimeOutErrors	WORD	Contador de erro de time-out no CTS, utilizando o handshake RTS/CTS, durante a transmissão (0 a 65535).
tStat. wDiag_18_Reserved	WORD	Reservado

Tabela 80: Diagnósticos MODBUS RTU Mestre

Nota:

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

eErrorCode: Os códigos das possíveis situações que ocasionam erro na comunicação MODBUS podem ser consultados abaixo:

Código	Enumerável	Descrição
0	NO_ERROR	Não existem erros.
1	ILLEGAL_SERIAL_PORT	Porta serial inválida.
2	ILLEGAL_SERIAL_MODE	Modo da porta serial inválido.
3	ILLEGAL_BAUDRATE	Taxa de transmissão inválida.
4	ILLEGAL_DATA_BITS	Bits de dado inválidos.
5	ILLEGAL_PARITY	Paridade inválida.
6	ILLEGAL_STOP_BITS	Bits de parada inválidos.
7	ILLEGAL_HANDSHAKE	Parâmetro de sinal de modem inválido.
8	ILLEGAL_UART_RX_THRESHOLD	Parâmetro de Threshold de RX da UART inválido.
9	ILLEGAL_TIMEOUT	Parâmetro de time-out inválido.
10	PORT_BUSY	Porta serial ocupada.
11	HW_ERROR_UART	Erro de hardware na UART.
12	HW_ERROR_REMOTE	Erro de hardware remoto.
20	ILLEGAL_TX_BUFF_LENGTH	Tamanho do buffer de transmissão inválido.
21	ILLEGAL_HANDSHAKE_METHOD	Método de sinal de modem inválido.
22	CTS_TIMEOUT_ON	Time-out de CTS = verdadeiro.
23	CTS_TIMEOUT_OFF	Time-out de CTS = falso.
24	TX_TIMEOUT_ERROR	Erro de time-out na transmissão.
30	ILLEGAL_RX_BUFF_LENGTH	Tamanho do buffer de recepção inválido.

Código	Enumerável	Descrição
31	RX_TIMEOUT_ERROR	Erro de time-out na recepção.
32	FB_SET_CTRL_NOT_ALLOWED	Controle de fluxo configurado diferente de manual.
33	FB_GET_CTRL_NOT_ALLOWED	Controle de fluxo inválido para a porta serial configurada.
34	FB_SERIAL_RX_NOT_ALLOWED	Recepção de dados não permitida no modo normal.
35	FB_SERIAL_RX_EXTENDED_NOT_ALLOWED	Recepção de dados não permitida no modo estendido.
36	DCD_INTERRUPT_NOT_ALLOWED	Interrupção DCD não permitida.
37	CTS_INTERRUPT_NOT_ALLOWED	Interrupção CTS não permitida.
38	DSR_INTERRUPT_NOT_ALLOWED	Interrupção DSR não permitida.
39	NOT_CONFIGURED	Porta serial não configurada.
50	INTERNAL_ERROR	Erro interno na porta serial.

Tabela 81: Enumerável SERIAL_STATUS

4.5.5.2. Configuração dos Dispositivos

A configuração dos dispositivos escravos, visualizados na figura abaixo, seguem os seguintes parâmetros:

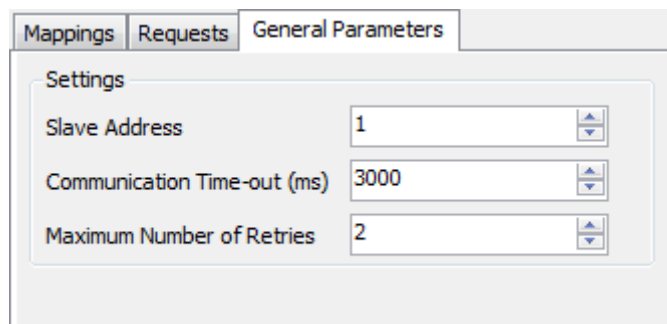


Figura 74: Tela de Configurações dos Parâmetros Gerais do Dispositivo

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

Tabela 82: Configurações do Dispositivo

Notas:

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

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

A seguinte fórmula deve ser adotada para cálculo estimado de Tempo de Transmissão (ms):

$$\text{Tempo de Transm(ms).} = [1000 \times (\text{contagem de caracteres}) \times (\text{bits por caracter})] / \text{Taxa de Transm.}$$

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

4.5.5.3. Configuração dos Mapeamentos

A configuração das relações MODBUS, visualizadas na figura abaixo, seguem os parâmetros descritos na tabela a seguir.

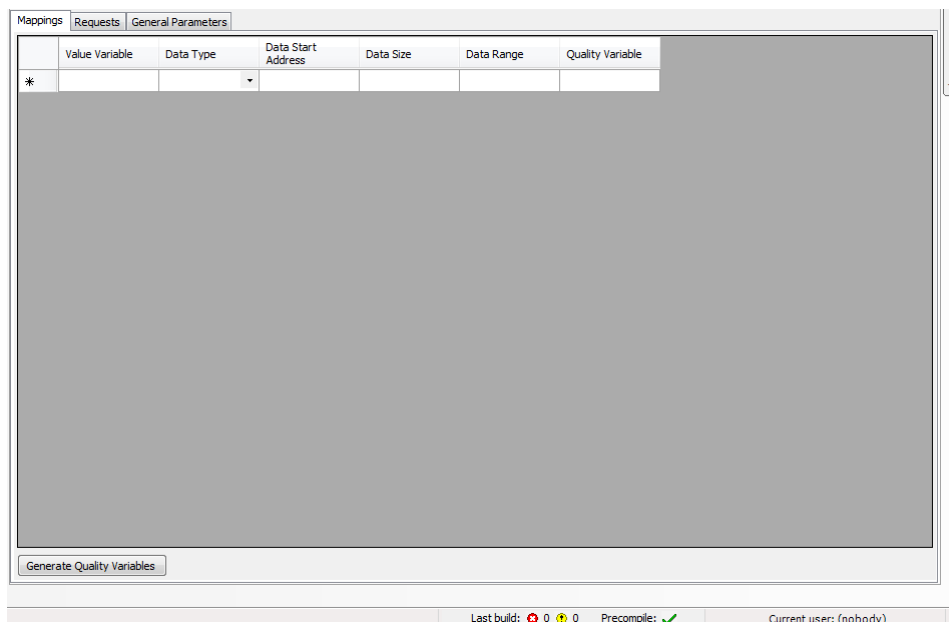


Figura 75: Tela de Mapeamentos de Dados MODBUS RTU Mestre

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

Configuração	Descrição	Padrão	Opções
Endereço Inicial do Dado	Endereço inicial dos dados MODBUS	-	1 a 65536
Tamanho do Dado	Tamanho do dado MODBUS	-	1 a 65536
Faixa de Dados	Faixa de endereços do dado configurado	-	-
Variável de Qualidade	Nome da variável que vai conter os dados de qualidade do mapeamento MODBUS	-	Nome de uma variável declarada em um programa ou GVL. A variável deve ser do tipo DWORD
Gerador de Variáveis de Qualidade	Gera automaticamente na GVL IOQualities variáveis do tipo QUALITY para cada Variável de Valor declarada. Para mais informações acesse o capítulo Conversões de Qualidade	-	-

Tabela 83: Configuração dos Mapeamentos MODBUS

Notas:

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

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

Tipo de Dado	Tamanho [bits]	Descrição
Coil Escrita	1	Saída digital de escrita.
Coil Leitura	1	Saída digital de leitura.
Holding Register de Escrita	16	Saída analógica de escrita.
Holding Register de Leitura	16	Saída analógica de leitura.
Holding Register com mascara AND	16	Saída analógica que pode ser lida ou escrita com máscara AND.
Holding Register com mascara OR	16	Saída analógica que pode ser lida ou escrita com máscara OR.
Input Register	16	Entrada analógica que pode ser apenas lida.
Input Status	1	Entrada digital que pode ser apenas lida.

Tabela 84: Tipos de Dados Suportados no MODBUS RTU Mestre

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

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

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

Variável de Qualidade: Este campo é utilizado para especificar uma variável que vai conter os dados da qualidade da relação MODBUS.

ATENÇÃO

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.

4.5.5.4. Configuração das Requisições

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

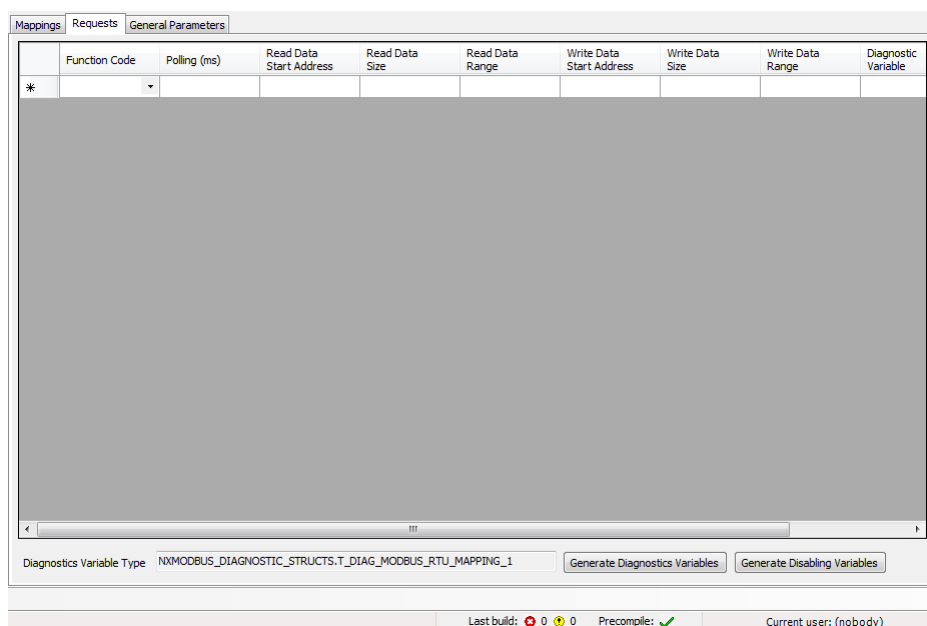


Figura 76: Tela de Requisições de Dados MODBUS Mestre

Configuração	Descrição	Padrão	Opções
Function Code	Tipo de função MODBUS	-	FC01 – Leitura de Coils FC02 – Leitura de Input Status FC03 – Leitura de Holding Registers FC04 – Leitura de Input Registers FC05 – Escrita de um Coil FC06 – Escrita de um Holding Register FC15 – Escrita de múltiplos Coils FC16 – Escrita de Holding Registers FC22 – Máscara de Escrita do Register FC23 – Leitura/Escrita de Holding Register
Polling (ms)	Período de comunicação (ms)	-	0 a 3600000
Read Data Start Address	Endereço inicial dos dados de leitura MODBUS	-	1 a 65536
Read Data Size	Tamanho dos dados de leitura MODBUS	-	Depende da função utilizada

Configuração	Descrição	Padrão	Opções
Read Data Range	Faixa de endereço dos dados de leitura MODBUS	-	0 a 2147483646
Write Data Start Address	Endereço inicial dos dados de escrita MODBUS	1	1 a 65536
Write Data Size	Tamanho dos dados de escrita MODBUS	-	Depende da função utilizada
Write Data Range	Faixa de endereço dos dados de escrita MODBUS	-	0 a 2147483647
Diagnostic Variable	Nome da variável de diagnóstico	-	Nome de uma variável declarada em um programa ou GVL
Disabling Variable	Variável utilizada para desabilitar a relação MODBUS	-	Campo destinado a variável simbólica utilizada para desabilitar, individualmente, as requisições MODBUS configuradas. Esta variável deve ser do tipo BOOL. A variável pode ser simples ou elemento de array e pode estar em estruturas.

Tabela 85: Configuração das Relações MODBUS Mestre

Notas:

Generate 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_MODBUS_RTU_MAPPING_1", na GVL "ReqDiagnostics", para cada uma das requisições.

Generate Disabling Variables: As variáveis simbólicas de desabilitação podem ser geradas automaticamente através do botão "Gerar Variáveis de Desabilitação". Ao clicar nesse botão, será criada uma variável do tipo BOOL, na GVL "Disables", para cada uma das requisições.

Configuração das relações MODBUS: O número de configurações, padrão de fábrica e os valores da coluna possibilidades, podem variar de acordo com o tipo de dado e função MODBUS (FC).

Function Code: As funções MODBUS (FC) disponíveis são os seguintes:

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

Tabela 86: Funções MODBUS Suportadas pela UCP Hadron Xtorm

Notas:

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

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

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

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

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.

Generate Diagnostic Variables: Os diagnósticos da requisição MODBUS configurada, são armazenados em variáveis do tipo T_DIAG_MODBUS_RTU_MAPPING_1. a qual está descrita na tabela abaixo.

Variável de diagnóstico do tipo T_DIAG_MODBUS_RTU_MAPPING_1.*	Tamanho	Descrição
Bits de estado da comunicação:		
byStatus. bCommIdle	BIT	Comunicação inativa (aguardando ser constada).
byStatus. bCommExecuting	BIT	Comunicação ativa.
byStatus. bCommPostponed	BIT	Comunicação adiada, pois o número máximo de requisições simultâneas foi atingido. As comunicações adiadas serão executadas na mesma sequência em que elas foram requisitadas, evitando a indeterminação. O tempo gasto nesse estado não é contabilizado para efeitos de time-out. Os bits bCommIdle e bCommExecuting são falsos quando o bit bCommPostponed é verdadeiro.
byStatus. bCommDisabled	BIT	Comunicação desabilitada. O bit bCommIdle é reiniciado nessa condição.
byStatus. bCommOk	BIT	Comunicação finalizada anteriormente foi realizada com sucesso.
byStatus. bCommError	BIT	Comunicação finalizada anteriormente teve um erro. Verificar código de erro.
byStatus. bCommAborted	BIT	Não utilizado no MODBUS Mestre RTU.
byStatus. bDiag_7_reserved	BIT	Reservado

Variável de diagnóstico do tipo T_DIAG_MODBUS_RTU_MAPPING_1.*	Tamanho	Descrição
Último código de erro (habilitado quando o bCommError = verdadeiro):		
eLastErrorCode	MASTER_ERROR_CODE (BYTE)	Informa a possível causa do último erro ocorrido na relação MODBUS. Consultar tabela 88.
Último código de exceção recebido pelo mestre:		
eLastExceptionCode	MODBUS_EXCEPTION (BYTE)	NO_EXCEPTION (0) FUNCTION_NOT_SUPPORTED (1) MAPPING_NOT_FOUND (2) ILLEGAL_VALUE (3) ACCESS_DENIED (128)* MAPPING_DISABLED (129)* IGNORE_FRAME (255)*
Estatísticas de comunicação:		
byDiag_3_reserved	BYTE	Reservado.
wCommCounter	WORD	Contador de comunicações finalizadas, com ou sem erros. O usuário pode testar quando a comunicação foi finalizada testando a variação desse contador. Quando o valor 65535 é atingido, o contador retorna à zero.
wCommErrorCounter	WORD	Contador de comunicações finalizadas com erros. Quando o valor 65535 é atingido, o contador retorna a zero.

Tabela 87: Diagnósticos das Relações MODBUS

Notas:

Códigos de exceção: Os códigos de exceção apresentados neste campo são os valores retornados pelo escravo. As definições dos códigos de exceção 128, 129 e 255, apresentadas nessa tabela, são válidas apenas na utilização de escravos Altus. Para escravos de outros fabricantes esses códigos de exceção podem ter significados diferentes.

Variável de Desabilitação: Campo destinado a variável do tipo booleana utilizada para desabilitar, individualmente, as requisições MODBUS configuradas na aba Requisições através do botão na parte inferior da janela. A requisição é desabilitada quando a variável, correspondente a requisição, for igual a 1, caso contrário, a requisição está habilitada.

Último código de Erro: Os códigos das possíveis situações que ocasionam erro na comunicação MODBUS podem ser consultados abaixo:

ATENÇÃO

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

Código	Enumerável	Descrição
1	ERR_EXCEPTION	Resposta reportada em um código de exceção (ver eLastExceptionCode = Código de Exceção).
2	ERR_CRC	Resposta com CRC inválido.
3	ERR_ADDRESS	Endereço MODBUS não encontrado. O endereço que respondeu à requisição foi diferente do esperado.
4	ERR_FUNCTION	Código inválido da função. A função recebida na resposta foi diferente da esperada pela requisição.
5	ERR_FRAME_DATA_COUNT	A quantidade de dados da resposta foi diferente da esperada.
7	ERR_NOT_ECHO	Resposta não é eco da pergunta (FC 5 e 6).
8	ERR_REFERENCE_NUMBER	Número de referência inválido (FC 15 e 16).
9	ERR_INVALID_FRAME_SIZE	Resposta menor do que a esperada.
20	ERR_CONNECTION	Erro durante a fase de conexão.
21	ERR_SEND	Erro durante a fase de transmissão.
22	ERR_RECEIVE	Erro durante a fase de recepção.
40	ERR_CONNECTION_TIMEOUT	Time-out no nível de aplicação durante a conexão.
41	ERR_SEND_TIMEOUT	Time-out no nível de aplicação durante a transmissão.
42	ERR_RECEIVE_TIMEOUT	Time-out no nível de aplicação enquanto aguarda resposta.
43	ERR_CTS_OFF_TIMEOUT	Time-out enquanto aguarda CTS = falso na transmissão.
44	ERR_CTS_ON_TIMEOUT	Time-out enquanto aguarda CTS = verdadeiro na transmissão.
128	NO_ERROR	Sem erro desde a inicialização.

Tabela 88: Códigos de Erro das relações MODBUS

4.5.6. 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. Para realizar esse procedimento, consultar a 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.

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

4.5.6.1. Parâmetros Gerais do Protocolo MODBUS Escravo

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

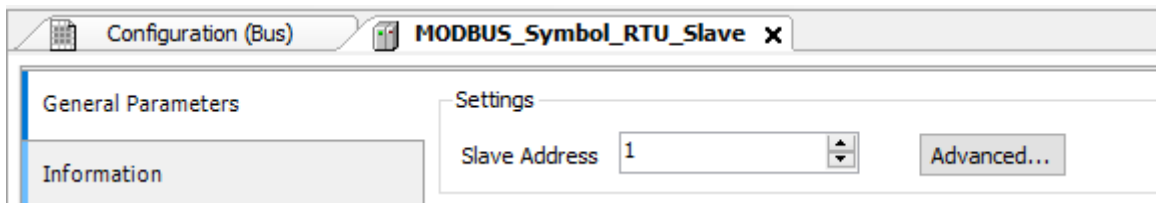


Figura 77: Tela de Configuração Parâmetros Gerais MODBUS Escravo

Configurações	Descrição	Padrão	Possibilidades
Slave Address	Endereço do escravo MODBUS	1	1 a 255

Tabela 89: Configurações do Escravo

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* como pode ser visto na figura abaixo e na tabela abaixo.

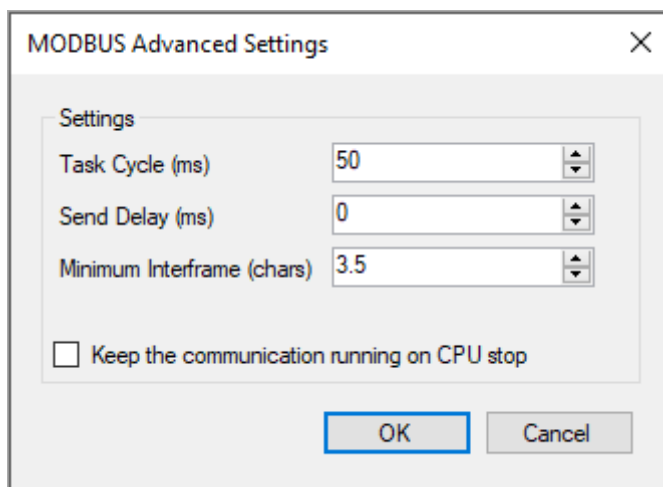


Figura 78: Configurações Avançadas MODBUS Escravo

Configurações	Descrição	Padrão	Possibilidades
Task Cycle (ms)	Tempo para execução da instância dentro do ciclo, sem considerar o tempo de execução da mesma.	50	20 a 100
Send Delay (ms)	Tempo de atraso para envio da resposta.	0	0 a 65535
Minimum Interframe (chars)	Tempo mínimo de silêncio entre diferentes frames.	3.5	3.5 a 100.0
Keep the communication running on CPU stop	Habilita o MODBUS Symbol Slave para comunicar enquanto a UCP estiver em STOP.	desmarcado	marcado ou desmarcado

Tabela 90: Configurações Avançadas MODBUS Escravo

Notas:

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

Send Delay: 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: 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 as quais estão descritas na tabela abaixo: Também através do botão "Advanced...", existe o checkbox "Mantém a comunicação funcionando quando a UCP está parada". Quando este estiver marcado, apesar de a CPU, que está configurada como Server/Slave estar em modo STOP, a comunicação MODBUS permanece funcionando. Ou seja, a CPU continua respondendo as requisições.

Variável de diagnósticos do tipo T_DIAG_MODBUS_RTU_SLAVE_1.*	Tamanho	Descrição
Bits de diagnóstico:		
tDiag. bRunning	BIT	O escravo está em execução.
tDiag. bNotRunning	BIT	O escravo não está em execução (ver bit: bInterruptedByCommand).
tDiag. bInterruptedByCommand	BIT	O bit bNotRunning foi habilitado, pois o escravo foi interrompido pelo usuário através de bits de comando.
tDiag. bConfigFailure	BIT	Diagnóstico descontinuado.
tDiag. bRXFailure	BIT	Diagnóstico descontinuado.
tDiag. bTXFailure	BIT	Diagnóstico descontinuado.
tDiag. bModuleFailure	BIT	Diagnóstico descontinuado.

Variável de diagnósticos do tipo T_DIAG_MODBUS_RTU_SLAVE_1.*	Tamanho	Descrição
tDiag. bDiag_7_reserved	BIT	Reservado
Códigos de erro:		
eErrorCode	SERIAL_STATUS (BYTE)	Informa a possível causa de erros. Ver tabela 81 para maiores informações.
Bits de comando, reiniciados automaticamente:		
tCommand. bStop	BIT	Parar o escravo.
tCommand. bRestart	BIT	Reiniciar o escravo.
tCommand. bResetCounter	BIT	Reiniciar as estatísticas dos diagnósticos (contadores).
tCommand. bDiag_19_reserved	BIT	Reservado
tCommand. bDiag_20_reserved	BIT	Reservado
tCommand. bDiag_21_reserved	BIT	Reservado
tCommand. bDiag_22_reserved	BIT	Reservado
tCommand. bDiag_23_reserved	BIT	Reservado
byDiag_3_reserved	BYTE	Reservado
Estatísticas de comunicação:		
tStat. wRXRequests	WORD	Contador de requisições normais recebidas pelo escravo e respondidas normalmente. No caso de um comando broadcast, esse contador é incrementado, porém não é transmitida a resposta (0 a 65535).
tStat. wTXExceptionResponses	WORD	Contador de requisições normais recebidas pelo escravo e respondidas com códigos de exceção. No caso de um comando broadcast, esse contador é incrementado, porém não é transmitida a resposta (0 a 65535). Códigos de exceção: 1: o código da função (FC) é legal, porém não suportado. 2: relação não encontrada nestes dados MODBUS. 3: valor ilegal para o endereço. 128: o mestre/cliente não tem direito de escrita ou leitura. 129: a relação MODBUS está desabilitada.
tStat. wRXFrames	WORD	Contador de frames recebidos pelo escravo. Considera-se frame, algo que é processado e seguido de um período mínimo de silêncio interframes (Interframe Mínimo), ou seja, uma mensagem ilegal também é computada (0 a 65535).

Variável de diagnósticos do tipo T_DIAG_MODBUS_RTU_SLAVE_1.*	Tamanho	Descrição
tStat. wRXIllegalRequests	WORD	Contador de requisições ilegais. Estes são frames que iniciam com o endereço 0 (broadcast) ou com o endereço MODBUS do escravo, mas não são requisições legais – sintaxe inválida, frames menores, CRC inválido – (0 a 65535).
tStat. wRXOverrunErrors	WORD	Contador de frames com erros de overrun durante a recepção – UART FIFO ou fila RX – (0 a 65535).
tStat. wRXIncompleteFrames	WORD	Contador de frames com erro de construção, paridade ou falha durante a recepção (0 a 65535).
tStat. wCTSTimeOutErrors	WORD	Contador de erro de time-out no CTS, utilizando o handshake RTS/CTS, durante a transmissão (0 a 65535).
tStat. wDiag_18_Reserved	WORD	Reservado

Tabela 91: Diagnósticos MODBUS RTU Escravo

Nota:

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

4.5.6.2. Configuração dos Mapeamentos

A configuração das relações MODBUS, visualizadas na figura abaixo, seguem os parâmetros descritos na tabela a seguir.

- Mappings

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

Figura 79: Tela de Mapeamentos de Dados MODBUS RTU Escravo

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

Tabela 92: Configuração dos Mapeamentos MODBUS

Notas:

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

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

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

Tabela 93: Tipos de Dados Suportados no MODBUS RTU Escravo

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

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

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

Máximo Tamanho de Relação: 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 ver capítulo de [Pontos Duplos](#).

ATENÇÃO

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

4.5.7. 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 as mesmas são MODBUS TCP ou MODBUS RTU via TCP, conforme descreve a tabela 60.

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

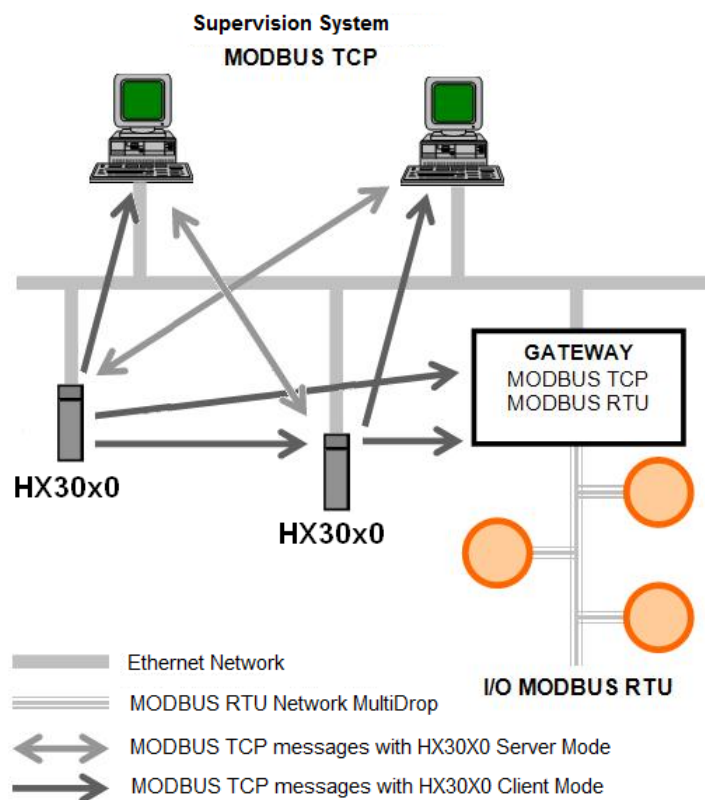


Figura 80: Rede de Comunicação MODBUS TCP

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.

4.5.8. 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 .. 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 MODBUS Cliente, com o protocolo TCP ou RTU via TCP.

4. CONFIGURAÇÃO

4. Adicionar e configurar dispositivos, definindo endereço IP, porta, endereço do escravo e time-out de comunicação (disponível no botão de configurações avançadas do Dispositivo).
5. 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.
6. 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.

As descrições de cada configuração estão relacionadas a seguir, nesta seção.

4.5.8.1. Parâmetros Gerais do Protocolo MODBUS Cliente

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

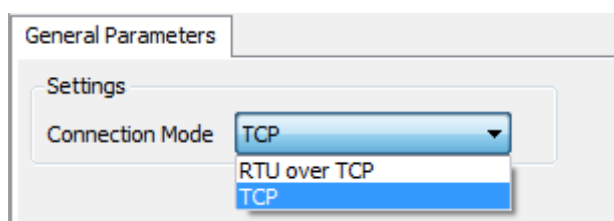


Figura 81: Tela de Configuração Parâmetros Gerais MODBUS TCP Cliente

Configuração	Descrição	Padrão de Fábrica	Possibilidades
Connection Mode	Seleção do protocolo	TCP	RTU over TCP TCP

Tabela 94: Configurações Gerais MODBUS TCP Cliente

4. CONFIGURAÇÃO

4.5.8.1.1. Diagnósticos MODBUS Cliente

Os diagnósticos e comandos do protocolo MODBUS TCP Cliente configurado, são armazenados em variáveis do tipo T_DIAG_MODBUS_ETH_CLIENT_1 as quais estão descritas na tabela abaixo.

Variável de diagnóstico do tipo T_DIAG_MODBUS_ETH_CLIENT_1.*	Tamanho	Descrição
Bits de diagnóstico:		
tDiag. bRunning	BIT	O cliente está em execução.
tDiag. bNotRunning	BIT	O cliente não está em execução (ver bit bInterruptedByCommand).
tDiag. bInterruptedByCommand	BIT	O bit bNotRunning foi habilitado pois o cliente foi interrompido pelo usuário através de bits de comando.
tDiag. bConfigFailure	BIT	Diagnóstico descontinuado.
tDiag. bRXFailure	BIT	Diagnóstico descontinuado.
tDiag. bTXFailure	BIT	Diagnóstico descontinuado.
tDiag. bModuleFailure	BIT	Indica se há falha no módulo ou o módulo não está presente.
tDiag. bAllDevicesCommFailure	BIT	Indica que a comunicação com todos os dispositivos está falhando.
tDiag. bDiag_7_reserved	BIT	Reservado
byDiag_1_reserved	BYTE	Reservado
Bits de comando, reiniciados automaticamente:		
tCommand. bStop	BIT	Parar o cliente.
tCommand. bRestart	BIT	Reiniciar o cliente.
tCommand. bResetCounter	BIT	Reiniciar as estatísticas dos diagnósticos (contadores).
tCommand. bDiag_19_reserved	BIT	Reservado
tCommand. bDiag_20_reserved	BIT	Reservado
tCommand. bDiag_21_reserved	BIT	Reservado
tCommand. bDiag_22_reserved	BIT	Reservado
tCommand. bDiag_23_reserved	BIT	Reservado
byDiag_3_reserved	BYTE	Reservado

Variável de diagnóstico do tipo T_DIAG_MODBUS_ETH_CLIENT_1.*	Tamanho	Descrição
Estatísticas de comunicação:		
tStat. wTXRequests	WORD	Contador de requisições transmitidas pelo cliente (0 a 65535).
tStat. wRXNormalResponses	WORD	Contador de respostas normais recebidas pelo cliente (0 a 65535).
tStat. wRXExceptionResponses	WORD	Contador de respostas com códigos de exceção (0 a 65535).
tStat. wRXIllegalResponses	WORD	Contador de respostas ilegais recebidas pelo cliente – sintaxe inválida, CRC inválido ou número insuficiente de bytes recebidos (0 a 65535).
tStat. wDiag_12_reserved	WORD	Reservado
tStat. wDiag_14_reserved	WORD	Reservado
tStat. wDiag_16_reserved	WORD	Reservado
tStat. wDiag_18_Reserved	WORD	Reservado

Tabela 95: Diagnósticos MODBUS TCP Cliente

Nota:

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

4.5.8.2. Configuração dos Dispositivos

A configuração dos dispositivos escravos, visualizados na figura abaixo, seguem os seguintes parâmetros:

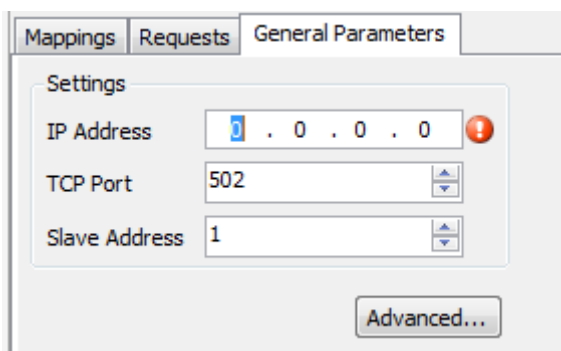


Figura 82: Tela de Configurações dos Parâmetros Gerais do Dispositivo MODBUS Cliente

Configuração	Descrição	Padrão de Fábrica	Possibilidades
IP Address	Endereço IP do servidor	0.0.0.0	1.0.0.1 a 223.255.255.254
TCP Port	Porta TCP	502	2 a 65534
Slave Address	Endereço do Escravo MODBUS	1	0 a 255

Tabela 96: Configurações Gerais MODBUS RTU Cliente

Notas:

IP Address: Endereço IP do dispositivo servidor Modbus.

TCP Port: 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, veja capítulo [Portas TCP Reservadas](#).

Slave Address: 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.

4.5.8.2.1. *Time-out de Comunicação*

Os parâmetros de Time-out de comunicação do protocolo MODBUS Cliente, encontrados no botão “Avançado...” da tela de configuração, estão divididos em: número máximo de requisições simultâneas, Time-out de Comunicação, Modo de time-out de conexão e tempo de inatividade.

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

Tabela 97: Configurações Avançadas MODBUS Cliente

Notas:

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

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

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

Tempo de Inatividade: Tempo de inatividade da conexão.

ATENÇÃO

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.

4.5.8.3. Configuração dos Mapeamentos

A configuração das relações MODBUS, visualizadas na figura abaixo, seguem os parâmetros descritos na tabela a seguir.

Figura 83: Tela de Mapeamentos de Dados MODBUS Cliente

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

Configuração	Descrição	Padrão	Opções
Variável de Qualidade	Nome da variável que vai conter os dados de qualidade do mapeamento MODBUS	-	Nome de uma variável declarada em um programa ou GVL. A variável deve ser do tipo DWORD
Gerador de Variáveis de Qualidade	Gera automaticamente na GVL IOQualities variáveis do tipo QUALITY para cada Variável de Valor declarada. Para mais informações acesse o capítulo Conversões de Qualidade	-	-

Tabela 98: Configuração dos Mapeamentos MODBUS

Notas:

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

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

Tipo de Dado	Tamanho [bits]	Descrição
Coil Escrita	1	Saída digital de escrita.
Coil Leitura	1	Saída digital de leitura.
Holding Register de Escrita	16	Saída analógica de escrita.
Holding Register de Leitura	16	Saída analógica de leitura.
Holding Register com mascara AND	16	Saída analógica que pode ser lida ou escrita com máscara AND.
Holding Register com mascara OR	16	Saída analógica que pode ser lida ou escrita com máscara OR.
Input Register	16	Entrada analógica que pode ser apenas lida.
Input Status	1	Entrada digital que pode ser apenas lida.

Tabela 99: Tipos de Dados Suportados no MODBUS

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

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

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

Variável de Qualidade: Este campo é utilizado para especificar uma variável que vai conter os dados da qualidade da relação MODBUS.

4.5.8.4. Configuração das Requisições

A configuração das relações MODBUS, visualizadas na figura abaixo, seguem os parâmetros descritos na tabela a seguir.

4. CONFIGURAÇÃO

The screenshot shows a software window titled 'General Parameters' with a table for configuring MODBUS data requests. The table has the following columns: 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, and Disabling Variable. A single row with an asterisk (*) is visible in the table. Below the table, there are two buttons: 'Generate Diagnostics Variables' and 'Generate Disabling Variables'. The 'Diagnostics Variable Type' is set to 'NXMODBUS_DIAGNOSTIC_STRUCTS.T_DIAG_MODBUS_ETH_MAPPING_1'.

Figura 84: Tela de Requisições de Dados MODBUS

Configuração	Descrição	Padrão	Opções
Function Code	Tipo de função MODBUS	-	FC01 – Leitura de Coils FC02 – Leitura de Input Status FC03 – Leitura de Holding Registers FC04 – Leitura de Input Registers FC05 – Escrita de um Coil FC06 – Escrita de um Holding Register FC15 – Escrita de múltiplos Coils FC16 – Escrita de Holding Registers FC22 – Máscara de Escrita do Register FC23 – Leitura/Escrita de Holding Register
Polling (ms)	Período de comunicação (ms)	-	0 a 3600000
Read Data Start Address	Endereço inicial dos dados de leitura MODBUS	-	1 a 65536
Read Data Size	Tamanho dos dados de leitura MODBUS	-	Depende da função utilizada
Read Data Range	Faixa de endereço dos dados de leitura MODBUS	-	0 a 2147483646
Write Data Start Address	Endereço inicial dos dados de escrita MODBUS	1	1 a 65536
Write Data Size	Tamanho dos dados de escrita MODBUS	-	Depende da função utilizada
Write Data Range	Faixa de endereço dos dados de escrita MODBUS	-	0 a 2147483647
Diagnostic Variable	Nome da variável de diagnóstico	-	Nome de uma variável declarada em um programa ou GVL

Configuração	Descrição	Padrão	Opções
Disabling Variable	Variável utilizada para desabilitar a relação MODBUS	-	Campo destinado a variável simbólica utilizada para desabilitar, individualmente, as requisições MODBUS configuradas. Esta variável deve ser do tipo BOOL. A variável pode ser simples ou elemento de array e pode estar em estruturas.

Tabela 100: Configuração das Relações MODBUS Cliente

Notas:

Generate 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_MODBUS_RTU_MAPPING_1", na GVL "ReqDiagnostics", para cada uma das requisições.

Generate Disabling Variables: As variáveis simbólicas de desabilitação podem ser geradas automaticamente através do botão "Gerar Variáveis de Desabilitação". Ao clicar nesse botão, será criada uma variável do tipo BOOL, na GVL "Disables", para cada uma das requisições.

Configuração das relações MODBUS: O número de configurações, padrão de fábrica e os valores da coluna possibilidades, podem variar de acordo com o tipo de dado e função MODBUS (FC).

Function Code: As funções MODBUS (FC) disponíveis são os seguintes:

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

Tabela 101: Funções MODBUS Cliente

Notas:

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

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

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

4. CONFIGURAÇÃO

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

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.

4.5.8.4.1. Diagnósticos MODBUS Cliente

Os diagnósticos da requisição MODBUS configurada, são armazenados em variáveis do tipo T_DIAG_MODBUS_ETH_CLIENT_1 as quais estão descritas na tabela abaixo.

Variável de diagnóstico do tipo T_DIAG_MODBUS_ETH_MAPPING_1.*	Tamanho	Descrição
Bits de estado da comunicação:		
byStatus. bCommIdle	BIT	Comunicação inativa (aguardando ser constada).
byStatus. bCommExecuting	BIT	Comunicação ativa.
byStatus. bCommPostponed	BIT	Comunicação adiada, pois o número máximo de requisições simultâneas foi atingido. As comunicações adiadas serão executadas na mesma sequência em que elas foram requisitadas, evitando a indeterminação. O tempo gasto nesse estado não é contabilizado para efeitos de time-out. Os bits bCommIdle e bCommExecuting são falsos quando o bit bCommPostponed é verdadeiro.
byStatus. bCommDisabled	BIT	Comunicação desabilitada. O bit bCommIdle é reiniciado nessa condição.
byStatus. bCommOk	BIT	Comunicação finalizada anteriormente foi realizada com sucesso.
byStatus. bCommError	BIT	Comunicação finalizada anteriormente teve um erro. Verificar código de erro.
byStatus. bCommAborted	BIT	Comunicação finalizada anteriormente foi interrompida devido à falha de conexão.
byStatus. bDiag_7_reserved	BIT	Reservado
Último código de erro (habilitado quando o bCommError = verdadeiro):		
eLastErrorCode	MASTER_ERROR_CODE (BYTE)	Informa a possível causa do último erro ocorrido na relação MODBUS. Consulte a Tabela 88 para detalhes das possibilidades.
Último código de exceção recebido pelo cliente:		

Variável de diagnóstico do tipo T_DIAG_MODBUS_ETH_MAPPING_1.*	Tamanho	Descrição
eLastExceptionCode	MODBUS_EXCEPTION (BYTE)	NO_EXCEPTION (0) FUNCTION_NOT_SUPPORTED (1) MAPPING_NOT_FOUND (2) ILLEGAL_VALUE (3) ACCESS_DENIED (128)* MAPPING_DISABLED (129)* IGNORE_FRAME (255)*
Estatísticas de comunicação:		
byDiag_3_reserved	BYTE	Reservado.
wCommCounter	WORD	Contador de comunicações finalizadas, com ou sem erros. O usuário pode testar quando a comunicação foi finalizada testando a variação desse contador. Quando o valor 65535 é atingido, o contador retorna à zero.
wCommErrorCounter	WORD	Contador de comunicações finalizadas com erros. Quando o valor 65535 é atingido, o contador retorna a zero.

Tabela 102: Diagnósticos das relações MODBUS Cliente

Notas:

Códigos de exceção: Os códigos de exceção apresentados neste campo são os valores retornados pelo servidor. As definições dos códigos de exceção 128, 129 e 255, apresentadas nessa tabela, são válidas apenas na utilização de escravos Altus. Para escravos de outros fabricantes esses códigos de exceção podem ter significados diferentes.

Variável de Desabilitação: Campo destinado à variável do tipo booleana utilizada para desabilitar, individualmente, as requisições MODBUS configuradas na aba Requisições através do botão na parte inferior da janela. A requisição é desabilitada quando a variável, correspondente a requisição, for igual a 1, caso contrário, a requisição está habilitada.

Último código de Erro: Os códigos das possíveis situações que ocasionam erro na comunicação MODBUS podem ser consultados na tabela 88.

ATENÇÃO

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

4.5.8.5. 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:

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

4.5.9. 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:

4. CONFIGURAÇÃO

1. Adicionar a instância do protocolo MODBUS Servidor a um dos canais Ethernet disponíveis (NET 1 .. 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 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).
4. Adicionar e configurar os mapeamentos MODBUS, especificando o nome da variável, tipo de dados, endereço inicial do dado e tamanho do dado.

As descrições de cada configuração estão relacionadas a seguir, nesta seção.

4.5.9.1. Parâmetros Gerais do Protocolo MODBUS Servidor

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

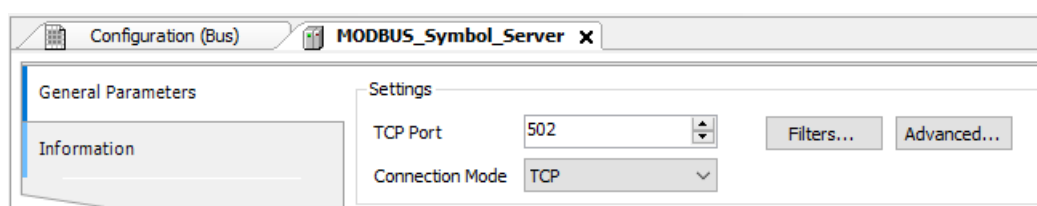


Figura 85: Tela de Configuração Parâmetros Gerais MODBUS Servidor

Configuração	Descrição	Padrão de Fábrica	Possibilidades
TCP Port	Porta TCP	502	2 a 65534
Connection Mode	Seleção do protocolo	TCP	RTU via TCP TCP

Tabela 103: Configurações Gerais MODBUS Servidor

Notas:

TCP Port: 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, veja capítulo [Portas TCP Reservadas](#).

As configurações presentes no botão “Filters...”, descritas na tabela abaixo, são relativas aos filtros de comunicação TCP:

Configuração	Descrição	Padrão de Fábrica	Possibilidades
Write Filter IP Address	Especifica um intervalo de IPs com acesso de escrita nas variáveis declaradas na relação MODBUS.	0.0.0.0	0.0.0.0 a 255.255.255.255
Write Filter Mask	Especifica a máscara de subrede em conjunto com o parâmetro Filtro de IP para Escrita.	0.0.0.0	0.0.0.0 a 255.255.255.255
Read Filter IP Address	Especifica um intervalo de IPs com acesso de leitura nas variáveis declaradas na relação MODBUS.	0.0.0.0	0.0.0.0 a 255.255.255.255
Read Filter Mask	Especifica a máscara de subrede em conjunto com o parâmetro Filtro de IP para Leitura.	0.0.0.0	0.0.0.0 a 255.255.255.255

Tabela 104: Filtros de IP

Notas:

Filtros: Os filtros são utilizados para estabelecer um intervalo de endereços IP que têm acesso de escrita ou leitura nas relações MODBUS, sendo individualmente configurados. O critério de permissão é realizado através de uma operação lógica AND entre o Filtro de Máscara para Escrita e o endereço IP do cliente. Caso o resultado seja igual ao Filtro de 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.

Os tempos de comunicação do protocolo MODBUS Servidor, encontrados no botão *Avançado...* da tela de configuração, estão divididos em: *Ciclo da Tarefa* e *Time-out da Inatividade da Conexão*, como pode ser visto na figura abaixo e na tabela abaixo.

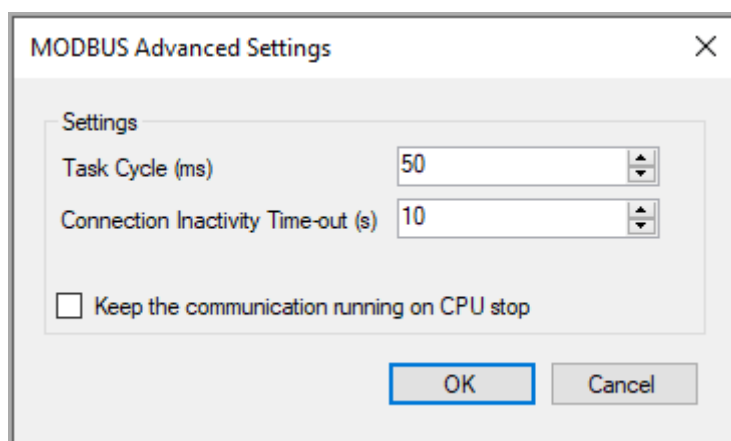


Figura 86: Configurações avançadas MODBUS Servidor

Configuração	Descrição	Padrão de Fábrica	Possibilidades
Task Cycle (ms)	Tempo para execução da instância dentro do ciclo, sem considerar o tempo de execução da mesma.	50	5 a 100
Connection Inactivity Time-out (s)	Tempo máximo de inatividade entre cliente e servidor antes da conexão ser fechada pelo servidor.	10	10 a 3600
Keep the communication running on CPU stop.	Habilita o MODBUS Symbol Server para comunicar enquanto a UCP estiver em STOP.	desmarcado	marcado ou desmarcado

Tabela 105: Configurações Avançadas

Notas:

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

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

4.5.9.1.1. Diagnósticos MODBUS Servidor

Os diagnósticos e comandos do protocolo MODBUS TCP Servidor configurado, são armazenados em variáveis do tipo T_DIAG_MODBUS_ETH_SERVER_1 as quais estão descritas na tabela abaixo.

Variável de diagnóstico do tipo T_DIAG_MODBUS_ETH_SERVER_1.*	Tamanho	Descrição
Bits de diagnóstico:		
tDiag. bRunning	BIT	O servidor está em execução.
tDiag. bNotRunning	BIT	O servidor não está em execução (ver bit bInterruptedByCommand).
tDiag. bInterruptedByCommand	BIT	O bit bNotRunning foi habilitado, pois o servidor foi interrompido pelo usuário através de bits de comando.
tDiag. bConfigFailure	BIT	Diagnóstico descontinuado.
tDiag. bRXFailure	BIT	Diagnóstico descontinuado.
tDiag. bTXFailure	BIT	Diagnóstico descontinuado.
tDiag. bModuleFailure	BIT	Diagnóstico descontinuado.
tDiag. bDiag_7_reserved	BIT	Reservado

Variável de diagnóstico do tipo T_DIAG_MODBUS _ETH_SERVER_1.*	Tamanho	Descrição
byDiag_1_reserved	BYTE	Reservado
Bits de comando, reiniciados automaticamente:		
tCommand. bStop	BIT	Parar o servidor.
tCommand. bRestart	BIT	Reiniciar o servidor.
tCommand. bResetCounter	BIT	Reiniciar as estatísticas dos diagnósticos (contadores).
tCommand. bDiag_19_reserved	BIT	Reservado
tCommand. bDiag_20_reserved	BIT	Reservado
tCommand. bDiag_21_reserved	BIT	Reservado
tCommand. bDiag_22_reserved	BIT	Reservado
tCommand. bDiag_23_reserved	BIT	Reservado
byDiag_3_reserved	BYTE	Reservado
Estatísticas de comunicação:		
tStat. wActiveConnections	WORD	Número de conexões estabelecidas entre cliente e servidor (0 a 64).
tStat. wTimeoutClosedConnections	WORD	Contador de conexões, entre cliente e servidor, interrompidas depois de um período de inatividade – time-out (0 a 65535).
tStat. wClientClosedConnections	WORD	Contador de conexões interrompidas devido à solicitação do cliente (0 a 65535).
tStat. wRXFrames	WORD	Contador de frames Ethernet recebidos pelo servidor, sendo que um frame Ethernet pode conter mais de uma requisição (0 a 65535).
tStat. wRXRequests	WORD	Contador de requisições recebidas pelo servidor e respondidas normalmente (0 a 65535).
tStat. wTXExceptionResponses	WORD	Contador de requisições recebidas pelo servidor e respondidas com códigos de exceção (0 a 65535). Os códigos de exceção estão relacionados abaixo: 1: o código da função (FC) é legal, porém não suportado. 2: relação não encontrada nestes dados MODBUS. 3: valor ilegal para o endereço. 128: o mestre/cliente não tem direito de escrita ou leitura. 129: a relação MODBUS está desabilitada.

Variável de diagnóstico do tipo T_DIAG_MODBUS _ETH_SERVER_1.*	Tamanho	Descrição
tStat. wRXIllegalRequests	WORD	Contador de requisições ilegais. (0 a 65535).
tStat. wDiag_18_Reserved	WORD	Reservado

Tabela 106: Diagnósticos MODBUS Servidor

Nota:

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

4.5.9.2. Configuração dos Mapeamentos

A configuração das relações MODBUS, visualizadas na figura abaixo, seguem os parâmetros descritos na tabela a seguir.

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

Figura 87: Tela de Mapeamentos de Dados MODBUS Servidor

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

Tabela 107: Configuração dos Mapeamentos MODBUS Ethernet

Notas:

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

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

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

Endereço Inicial Absoluto dos Dados: 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.

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

Faixa de Dados: 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 podem haver sobreposições de faixa com outros mapeamentos do mesmo "Tipo de Dado".

Máximo Tamanho de Relação: 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. Para mais informações ver capítulo [Configuração das Interfaces Ethernet](#).

ATENÇÃO

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

4.5.10. DNP3 - Tipos de Dados

A tabela abaixo mostra o tipo de variável suportada pela HX3040 para cada um dos tipos de dados do protocolo DNP3.

Tipo de Mapeamento	Tipos de Variáveis IEC	Variação Válida de StaticGroup	Variação Válida de EventGroup	Event Class Faixa
BI: Binary Input	BOOL BIT	g01v01 - Entrada digital em formato compactado g01v02 - Entrada digital com qualidade	g02v01 - Evento de entrada digital sem tempo g02v02 - Evento de entrada digital com tempo absoluto g02v03 - Evento de entrada digital com tempo relativo	0..3
DBI: Double Binary Input	DBP	g03v01 - Entrada digital dupla em formato compactado g03v02 - Entrada digital dupla com qualidade	g04v01 - Evento de entrada digital dupla sem tempo g04v02 - Evento de entrada digital dupla com tempo absoluto g04v03 - Evento de entrada digital dupla com tempo relativo	0..3
BO: Binary Output	BOOL DBP BIT	g10v01 - Saída digital em formato compactado g10v02 - Saída digital com qualidade	-	0
CN: Counter	UINT	g20v02 - Contador de 16 bits com qualidade g20v06 - Contador de 16 bits sem qualidade	g22v02 - Evento do contador de 16 bits com qualidade g22v06 - Evento do contador de 16 bits com qualidade e tempo	0..3
	UDINT	g20v01 - Contador de 32 bits com qualidade g20v05 - Contador de 32 bits sem qualidade	g22v01 - Evento do contador de 32 bits com qualidade g22v05 - Evento do contador de 32 bits com qualidade e tempo	0..3

Tipo de Mapeamento	Tipos de Variáveis IEC	Variação Válida de StaticGroup	Variação Válida de EventGroup	Event Class Faixa
FCN: Frozen Counter	UINT	g21v02 - Contador congelado de 16 bits com qualidade g21v10 - Contador congelado de 16 bits sem qualidade	g23v02 - Evento do contador congelado de 16 bits com qualidade g23v06 - Evento do contador congelado de 16 bits com qualidade e tempo	0..3
	UDINT	g21v01 - Contador congelado de 32 bits com qualidade g21v09 - Contador congelado de 32 bits sem qualidade	g23v01 - Evento do contador congelado de 32 bits com qualidade g23v05 - Evento do contador congelado de 32 bits com qualidade e tempo	0..3
AI: Analog Input	INT	g30v02 - Entrada analógica de 16 bits com qualidade g30v04 - Entrada analógica de 16 bits sem qualidade	g32v02 - Evento de entrada analógica de 16 bits sem tempo g32v04 - Evento de entrada analógica de 16 bits com tempo	0..3
	DINT	g30v01 - Entrada analógica de 32 bits com qualidade g30v03 - Entrada analógica de 32 bits sem qualidade	g32v01 - Evento de entrada analógica de 32 bits sem tempo g32v03 - Evento de entrada analógica de 32 bits com tempo	0..3
	REAL	g30v05 - Entrada analógica de precisão simples com qualidade	g32v05 - Evento de entrada analógica de precisão simples sem tempo g32v07 - Evento de entrada analógica de precisão simples em ponto flutuante com tempo	0..3
AO: Analog Output	INT	g40v02 - Saída analógica de 16 bits com qualidade	-	0
	DINT	g40v01 - Saída analógica de 32 bits com qualidade	-	0
	REAL	g40v03 - Saída analógica de precisão simples com qualidade	-	0

Tabela 108: Declaração de Variáveis para DNP3

4.5.11. 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 pela UCP 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.

As descrições de cada configuração estão relacionadas a seguir, nesta seção.

4.5.11.1. Configuração dos Mapeamentos DNP3 Cliente

As configurações dos Parâmetros Gerais de um DNP3 Cliente, visualizadas na figura abaixo, seguem os parâmetros descritos na tabela a seguir.

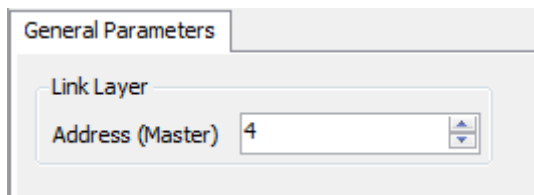


Figura 88: Tela de Parâmetros Gerais do DNP3 Cliente

Configuração	Descrição	Padrão de Fábrica	Possibilidades
Address (Master)	Define o endereço do mestre DNP3.	4	0 a 65519

Tabela 109: Configuração dos Parâmetros Gerais do DNP3 Cliente

A configuração das relações DNP3 Cliente, visualizadas na figura abaixo, seguem os parâmetros descritos na tabela a seguir.

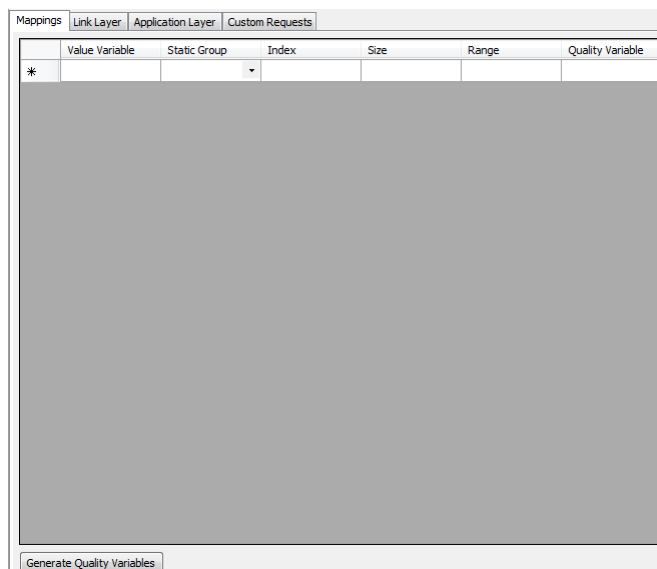


Figura 89: Tela de Mapeamentos de Dados DNP3 Cliente

Configuração	Descrição	Padrão	Opções
Value Variable	Nome da variável simbólica.	-	Nome de uma variável declarada em um programa ou GVL
Static Group	Configuração dos dados estáticos.	-	g01 – Entrada Digital g03 – Entrada Digital Dupla g10 – Saída Digital g20 – Contador g21 – Contador Congelado g30 – Entrada Analógica* g40 – Saída Analógica*
Index	Índice do primeiro ponto de mapeamento DNP3.	-	0 a 65535
Size	Tamanho do dado DNP3.	-	- (calculado automaticamente)
Range	Faixa de endereços do dado configurado.	-	- (calculado automaticamente)
Quality Variable	Nome da variável simbólica que armazenará a qualidade do mapeamento.	-	Variável do tipo QUALITY
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	-	-

Tabela 110: Configuração dos Mapeamentos DNP3 Cliente

Notas:

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.

ATENÇÃO

Quando variáveis do grupo 30 (Analógicas de entrada) e grupo 40 (Analógicas 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 INVALID e o valor do ponto será o último valor válido lido pelo DNP3 Cliente. Isto ocorrerá somente no caso do DNP3 Servidor responder em uma variação maior do que a mapeada no DNP3 Cliente. No caso do 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.

ATENÇÃO

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.

4.5.11.2. Configuração da Camada de Enlace DNP3 Cliente

As configurações dos parâmetros da camada de enlace do DNP3 Cliente, visualizadas na figura abaixo, seguem os parâmetros descritos na tabela a seguir.

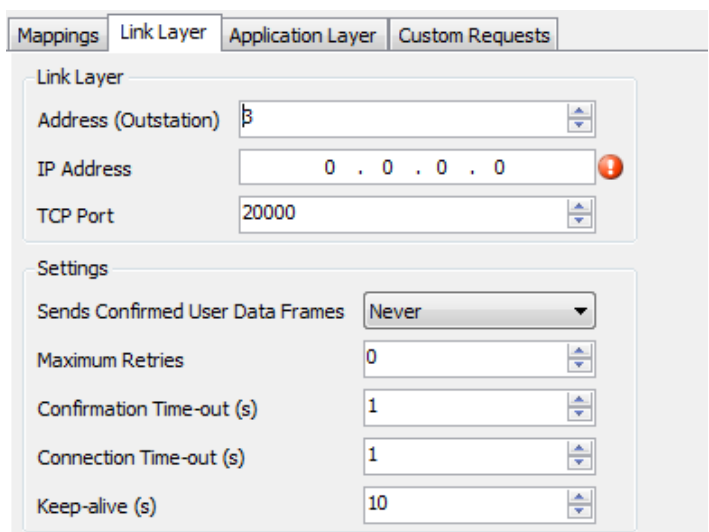


Figura 90: Tela de Configurações da Camada de Enlace DNP3 Cliente

Configuração	Descrição	Padrão	Opções
Address (Outstation)	Endereço DNP3 de origem deste Outstation.	3	0 a 65519
IP Address	Endereço IP deste Outstation.	0.0.0.0	1.0.0.1 a 223.255.255.254
TCP Port	Endereço da porta listen para conexão com este Outstation.	20000	0 a 65535
Sends Confirmed User Data Frames	Seleciona o modo de confirmação de mensagens.	Never	Never Sometimes Always
Maximum Retries	Número de vezes que este Outstation irá retransmitir a mensagem caso ele não receba a confirmação do cliente.	0	0 a 10
Confirmation Time-Out(s)	Tempo para que este Outstation envie a mensagem de confirmação novamente.	1	1 a 86400
Connection Time-Out(s)	Tempo máximo para que este Outstation se conecte com o DNP3 Cliente através do protocolo TCP.	1	1 a 10
Keep Alive (s)	Tempo do teste de link.	10	1 a 60

Tabela 111: Configuração da Camada de Enlace DNP3 Cliente

Notas:

Address (Outstation): Endereço DNP3 de origem deste outstation.

IP Address: Endereço IP deste outstation.

Porta TCP: Endereço da porta listen para conexão com este outstation.

Sends Confirmed User Data Frames: Envio de mensagens solicitando ao outstation que confirme ou não o recebimento. Os modos de confirmação da camada de enlace disponíveis são os seguintes:

	Opção	Descrição
Modo de Confirmação da Camada de Enlace	Nunca	Nunca solicitará confirmação das mensagens
	Sempre	Sempre solicitará confirmação das mensagens
	Às vezes	Solicitará confirmação somente para mensagens multifragmentadas

Tabela 112: Modos de Confirmação da Camada de Enlace do DNP3 Cliente

Notas:

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*”.

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.

4.5.11.3. Configuração da Camada de Aplicação DNP3 Cliente

A configuração da camada de aplicação de um DNP3 Servidor adicionado sob um DNP3 Cliente é visualizada na figura abaixo. Os parâmetros configurados nesta tela são descritos na tabela a seguir.

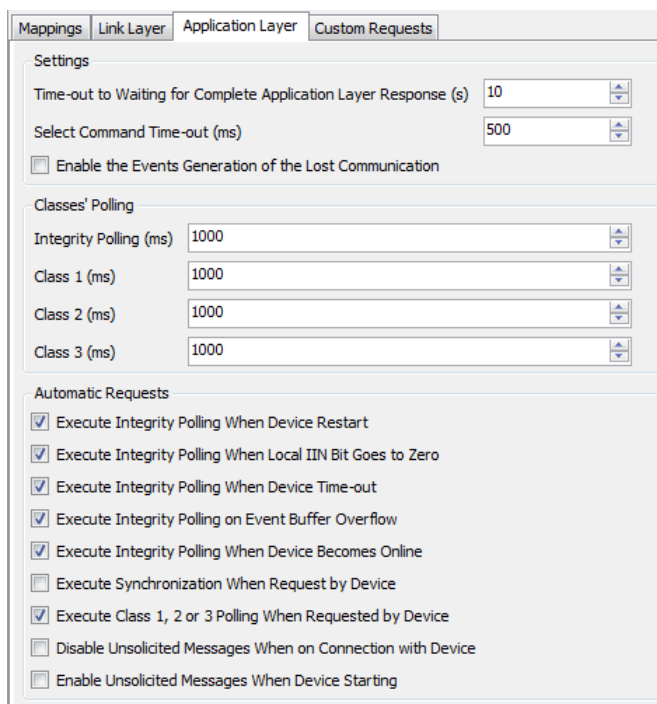


Figura 91: Tela de Configurações da Camada de Aplicação DNP3 Cliente

Configuração	Descrição	Padrão	Opções
Time-out to Waiting for Complete Application Layer Response (s)	Tempo máximo que o outstation aguardará resposta completa da camada de aplicação.	10	1 a 86400
Select Command Time-out (ms)	Tempo máximo para a execução do comando de seleção.	500	100 a 10000
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.	Disabled	Enabled Disabled
Integrity Polling (ms)	Determina o tempo (em milissegundos) em que é executada a varredura de todos os pontos mapeados.	1000	0 a 86400000
Class 1 (ms)	Tempo em que é executada a varredura dos pontos mapeados para a “ Classe 1”.	1000	0 a 86400000
Class 2 (ms)	Tempo em que é executada a varredura dos pontos mapeados para a “ Classe 2”.	1000	0 a 86400000
Class 3 (ms)	Tempo em que é executada a varredura dos pontos mapeados para a “ Classe 3”.	1000	0 a 86400000
Execute Integrity polling When Device Restart	Atualiza os valores de todos os pontos do dispositivo outstation e também lê todos os eventos quando o bit DEVICE_RESTART do IIN estiver em um.	Enabled	Enabled Disabled

Configuração	Descrição	Padrão	Opções
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.	Enabled	Enabled Disabled
Execute Integrity Polling When Device Time-out	Executa a varredura de integridade quando o outstation estiver operando em modo local, não sendo acessível pelo mestre.	Enabled	Enabled Disabled
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.	Enabled	Enabled Disabled
Execute Integrity Polling When Device Becomes Online	Executa a varredura de integridade quando a comunicação com o mestre for restabelecida após um time-out.	Enabled	Enabled Disabled
Execute Synchronization When Request By Device	Permite que o mestre envie o comando de acerto de hora quando o outstation solicitar.	Disabled	Enabled Disabled
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 um.	Enabled	Enabled Disabled
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.	Disabled	Enabled Disabled
Enable Unsolicited Messages When Device Starting	Envia o comando de ativação das mensagens não solicitadas quando o outstation conectar.	Disabled	Enabled Disabled

Tabela 113: Configuração da Camada de Aplicação DNP3 Cliente

4.5.11.4. Requisições Customizadas DNP3 Cliente

As configurações das relações customizadas do DNP3 Cliente, visualizadas na figura abaixo, seguem os parâmetros descritos na tabela a seguir.

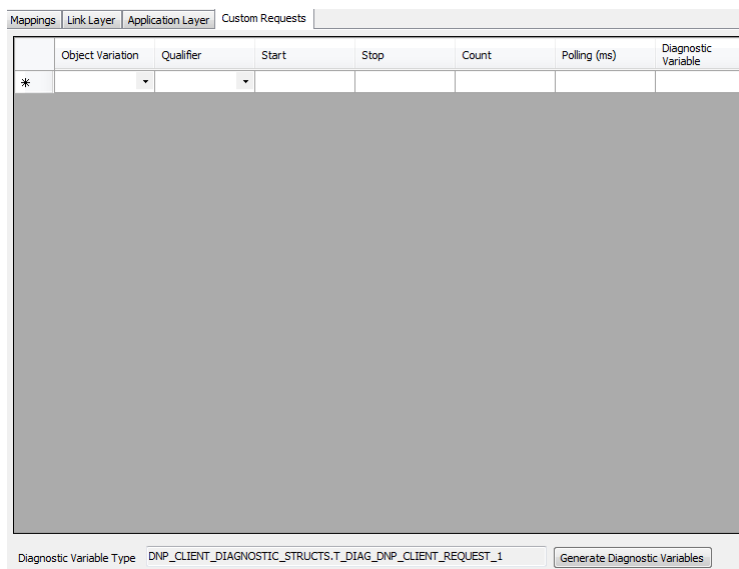


Figura 92: Tela de Mensagens Requisições Customizadas DNP3 Cliente

Configuração	Descrição	Padrão	Opções
Object Variation	Define o grupo e a variação dos objetos que serão requisitados.	-	g01v00 - Entrada digital com variação. g01v01 - Entrada digital em formato compactado. g01v02 - Entrada digital com qualidade g02v00 - Evento de entrada digital com variação. g02v01 - Evento de entrada digital sem tempo. g02v02 - Evento de entrada digital com tempo absoluto. g02v03 - Evento de entrada digital com tempo relativo. g03v00 - Entrada digital dupla com variação. g03v01 - Entrada digital dupla em formato compactado. g03v02 - Entrada digital dupla com qualidade. g04v00 - Evento de entrada digital dupla com variação. g04v01 - Evento de entrada digital dupla sem tempo. g04v02 - Evento de entrada digital dupla com tempo absoluto. g04v03 - Evento de entrada digital dupla com tempo relativo. g10v00 - Saída digital com variação. g10v01 - Saída digital em formato compactado. g10v02 - Saída digital com qualidade. g20v00 - Contador com variação. g20v01 - Contador de 32 bits com qualidade. g20v02 - Contador de 16 bits com qualidade. g20v05 - Contador de 32 bits sem qualidade. g20v06 - Contador de 16 bits sem qualidade. g21v00 - Contador congelado com variação. g21v01 - Contador congelado de 32 bits com qualidade. g21v02 - Contador congelado de 16 bits com qualidade.

4. CONFIGURAÇÃO

Configuração	Descrição	Padrão	Opções
			<p>g21v09 - Contador congelado de 32 bits sem qualidade.</p> <p>g21v10 - Contador congelado de 16 bits sem qualidade.</p> <p>g22v00 - Evento do contador com variação.</p> <p>g22v01 - Evento do contador de 32 bits com qualidade.</p> <p>g22v02 - Evento do contador de 16 bits com qualidade.</p> <p>g22v05 - Evento do contador de 32 bits com qualidade e tempo.</p> <p>g22v06 - Evento do contador de 16 bits com qualidade e tempo.</p> <p>g23v00 - Evento do contador congelado com variação.</p> <p>g23v01 - Evento do contador congelado de 32 bits com qualidade.</p> <p>g23v02 - Evento do contador congelado de 16 bits com qualidade.</p> <p>g23v05 - Evento do contador congelado de 32 bits com qualidade e tempo.</p> <p>g23v06 - Evento do contador congelado de 16 bits com qualidade e tempo.</p> <p>g30v00 - Entrada analógica com variação.</p> <p>g30v01 - Entrada analógica de 32 bits com qualidade.</p> <p>g30v02 - Entrada analógica de 16 bits com qualidade.</p> <p>g30v03 - Entrada analógica de 32 bits sem qualidade.</p> <p>g30v04 - Entrada analógica de 16 bits sem qualidade.</p> <p>g30v05 - Entrada analógica de precisão simples com qualidade.</p> <p>g32v00 - Evento de entrada analógica com variação.</p> <p>g32v01 - Evento de entrada analógica de 32 bits sem tempo.</p> <p>g32v02 - Evento de entrada analógica de 16 bits sem tempo.</p> <p>g32v03 - Evento de entrada analógica de 32 bits com tempo.</p> <p>g32v04 - Evento de entrada analógica de 16 bits com tempo.</p> <p>g32v05 - Evento de entrada analógica de precisão simples sem tempo.</p> <p>g32v07 - Evento de entrada analógica de precisão simples em ponto flutuante com tempo.</p> <p>g40v00 - Saída analógica com variação.</p> <p>g40v01 - Saída analógica de 32 bits com qualidade.</p> <p>g40v02 - Saída analógica de 16 bits com qualidade.</p> <p>g40v03 - Saída analógica de precisão simples com qualidade.</p> <p>g60v01 - Classe de dados 0.</p> <p>g60v02 - Classe de dados 1.</p> <p>g60v03 - Classe de dados 2.</p> <p>g60v04 - Classe de dados 3.</p>
Qualifier	Identifica qual o tipo de requisição que será realizada.	-	Faixa Todos Contador
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.	0	0 a 65535

Configuração	Descrição	Padrão	Opções
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.	0	0 a 65535
Count	Define quantos pontos serão requisitados. Só pode ser preenchido se a coluna "Qualificador" estiver com o valor "Contador" selecionado.	0	0 a 65535
Polling (ms)	Define o período (em milissegundos) em que ocorre a requisição.	1000	0 a 86400000
Diagnostic Variable	Nome da variável simbólica que receberá o diagnóstico da requisição.	-	Nome de uma variável declarada em um programa ou GVL

Tabela 114: Configuração das Requisições Customizadas DNP3 Cliente

Notas:

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.

Qualifier: Identifica o tipo de requisição que será realizada. As opções disponíveis dependem do tipo de Variação do Objeto escolhido. Na tabela está descrito os tipos de Qualificador e na os possíveis valores de qualificador para cada um dos grupos das requisições.

Tipo de Função	Configurações	Descrição
Qualifier	Todos	Todos os pontos do grupo e variação são requisitados.
	Faixa	Serão requisitados somente uma faixa de pontos delimitada pelas colunas "Início" e "Fim".
	Contador	Para requisitar uma quantidade de eventos definida na coluna "Contador", por exemplo, "ler 10 eventos".

Tabela 115: Tipos de Qualificador nos Parâmetros do DNP3 Cliente

4. CONFIGURAÇÃO

Variação do Objeto	Qualificador Permitido
g01v00 - Entrada digital com variação. g01v01 - Entrada digital em formato compactado. g01v02 - Entrada digital com qualidade.	Todos Faixa
g02v00 - Evento de entrada digital com variação. g02v01 - Evento de entrada digital sem tempo. g02v02 - Evento de entrada digital com tempo absoluto. g02v03 - Evento de entrada digital com tempo relativo.	Todos Contador
g03v00 - Entrada digital dupla com variação. g03v01 - Entrada digital dupla em formato compactado. g03v02 - Entrada digital dupla com qualidade.	Todos Faixa
g04v00 - Evento de entrada digital dupla com variação. g04v01 - Evento de entrada digital dupla sem tempo. g04v02 - Evento de entrada digital dupla com tempo absoluto. g04v03 - Evento de entrada digital dupla com tempo relativo.	Todos Contador
g10v00 - Saída digital com variação. g10v01 - Saída digital em formato compactado. g10v02 - Saída digital com qualidade.	Todos Faixa
g20v00 - Contador com variação. g20v01 - Contador de 32 bits com qualidade. g20v02 - Contador de 16 bits com qualidade. g20v05 - Contador de 32 bits sem qualidade. g20v06 - Contador de 16 bits sem qualidade.	Todos Faixa
g21v00 - Contador congelado com variação. g21v01 - Contador congelado de 32 bits com qualidade. g21v02 - Contador congelado de 16 bits com qualidade. g21v09 - Contador congelado de 32 bits sem qualidade. g21v10 - Contador congelado de 16 bits sem qualidade.	Todos Faixa
g22v00 - Evento do contador com variação. g22v01 - Evento do contador de 32 bits com qualidade. g22v02 - Evento do contador de 16 bits com qualidade. g22v05 - Evento do contador de 32 bits com qualidade e tempo. g22v06 - Evento do contador de 16 bits com qualidade e tempo.	Todos Contador
g23v00 - Evento do contador congelado com variação. g23v01 - Evento do contador congelado de 32 bits com qualidade. g23v02 - Evento do contador congelado de 16 bits com qualidade. g23v05 - Evento do contador congelado de 32 bits com qualidade e tempo. g23v06 - Evento do contador congelado de 16 bits com qualidade e tempo.	Todos Contador
g30v00 - Entrada analógica com variação. g30v01 - Entrada analógica de 32 bits com qualidade. g30v02 - Entrada analógica de 16 bits com qualidade. g30v03 - Entrada analógica de 32 bits sem qualidade. g30v04 - Entrada analógica de 16 bits sem qualidade. g30v05 - Entrada analógica de precisão simples com qualidade.	Todos Faixa

4. CONFIGURAÇÃO

Variação do Objeto	Qualificador Permitido
g32v00 - Evento de entrada analógica com variação. g32v01 - Evento de entrada analógica de 32 bits sem tempo. g32v02 - Evento de entrada analógica de 16 bits sem tempo. g32v03 - Evento de entrada analógica de 32 bits com tempo. g32v04 - Evento de entrada analógica de 16 bits com tempo. g32v05 - Evento de entrada analógica de precisão simples sem tempo. g32v07 - Evento de entrada analógica de precisão simples em ponto flutuante com tempo.	Todos Contador
g40v00 - Saída analógica com variação. g40v01 - Saída analógica de 32 bits com qualidade. g40v02 - Saída analógica de 16 bits com qualidade. g40v03 - Saída analógica de precisão simples com qualidade.	Todos Faixa
g60v01 - Classe de dados 0. g60v02 - Classe de dados 1. g60v03 - Classe de dados 2. g60v04 - Classe de dados 3.	Todos Contador

Tabela 116: Possíveis Valores de Qualificador para Todos os Grupos de Requisições

4.5.11.4.1. Diagnósticos DNP3 Cliente

Os diagnósticos e comandos do protocolo DNP3 Cliente configurado, são armazenados em variáveis do tipo T_DIAG_DNP_CLIENT_REQUEST_1 as quais estão descritas na tabela abaixo.

Variável de diagnóstico do tipo T_DIAG_DNP_CLIENT_1.*	Tamanho	Descrição
Bits de comando:		
tCommand. bStop	BIT	Desabilitar Driver.
tCommand. bStart	BIT	Habilitar o Driver.
tCommand. bDiag_01_Reserved	BIT	Reservado.
tCommand. bDiag_02_Reserved	BIT	Reservado.
tCommand. bDiag_03_Reserved	BIT	Reservado.
tCommand. bDiag_04_Reserved	BIT	Reservado.
tCommand. bDiag_05_Reserved	BIT	Reservado.
tCommand. bDiag_06_Reserved	BIT	Reservado.
Bits de diagnóstico:		
bRunning	BOOL	DNP Cliente está sendo executado.
Estatísticas de comunicação:		
tStat. wRXFrames	WORD	Número de quadros (frames) recebidos.

Variável de diagnóstico do tipo T_DIAG_DNP_CLIENT_1.*	Tamanho	Descrição
tStat. wTXFrames	WORD	Número de quadros (frames) enviados.
tStat. wRequestsSent	WORD	Indica o número de solicitações enviadas. Não contabiliza “solicitações automáticas”.
tStat. wCommErrors	WORD	Contador de erros de comunicação incluindo erros na Camada Física, Camada de Enlace e Camada de Transporte.
tStat. dwReserved_0	DWORD	Reservado.
tStat. dwReserved_1	DWORD	Reservado.
dwReserved_0	DWORD	Reservado.
dwReserved_1	DWORD	Reservado.

Tabela 117: Diagnósticos DNP3 Cliente

Os diagnósticos das requisições DNP3 Cliente estão descritos na tabela abaixo.

Variável de diagnóstico do tipo T_DIAG_DNP_CLIENT_REQUEST_1.*	Tamanho	Descrição
Diagnósticos de Requisição:		
eConnection Status. CLOSED	BYTE	Comunicação fechada com o IED.
eConnection Status. CONNECTED	BYTE	Comunicação operante com o IED.
eRequestStatus. SUCCESS	BYTE	Indicação de que a requisição foi concluída com sucesso.
eRequestStatus. PROCESSING	BYTE	Indica que uma resposta foi recebida, mas o comando ainda não foi completado. Isto poderia significar que a resposta é parte de uma resposta multi-fragmentada e não tem o bit final setado ou poderia ser um pedido tal como o “select operate” que requer múltiplas solicitações de respostas.
eRequestStatus. FAILURE	BYTE	Indica que houve uma falha na transmissão da requisição.
eRequestStatus. MISMATCH	BYTE	A resposta a um comando “select” ou “operate” indica erros .
eRequestStatus. STATUS_CODE_CHANGED	BYTE	O status de uma resposta a um comando “select” indica erros.
eRequestStatus. IIN_ERROR	BYTE	A resposta de uma requisição possuiu IIN bits indicando erro.
eRequestStatus. TIMEOUT	BYTE	Sincronização de tempo é necessária.

Variável de diagnóstico do tipo T_DIAG_DNP_CLIENT_REQUEST_1.*	Tamanho	Descrição
eRequestStatus.CANCELED	BYTE	Indica que uma segunda requisição foi feita e, portanto, a primeira requisição é cancelada.
tINN.ALL_STATIONS	BIT	Indica que foi recebida uma mensagem de todas as estações.
tINN.CLASS_1_EVENTS	BIT	Indica que existem eventos de classe 1 na fila do outstation a serem reportados.
tINN.CLASS_2_EVENTS	BIT	Indica que existem eventos de classe 2 na fila do outstation a serem reportados.
tINN.CLASS_3_EVENTS	BIT	Indica que existem eventos de classe 3 na fila do outstation a serem reportados.
tINN.NEED_TIME	BIT	Indica que o IED quer realizar sincronização de tempo usando o DNP3.
tINN.LOCAL_CONTROL	BIT	Um ou mais pontos do outstation estão em modo de controle local.
tINN.DEVICE_TROUBLE	BIT	Outstation com defeito.
tINN.DEVICE_RESTART	BIT	Outstation foi reiniciado.
tINN.NO_FUNC_CODE_SUPPORT	BIT	O outstation não suporta o código de função.
tINN.OBJECT_UNKNOWN	BIT	Outstation não suporta o comando para os objetos da requisição.
tINN.PARAMETER_ERROR	BIT	Foi detectado um erro de parâmetro.
tINN.EVENT_BUFFER_OVERFLOW	BIT	Existe uma condição de overflow na fila no outstation e pelo menos um evento não confirmado foi perdido.
tINN.ALREADY_EXECUTING	BIT	A operação solicitada já está em execução.
tINN.CONFIG_CORRUPT	BIT	O outstation possui uma configuração errada.
tINN.RESERVED_2	BIT	Reservado.
tINN.RESERVED_1	BIT	Reservado.
dwReserved_0	BIT	Reservado.
dwReserved_1	BIT	Reservado.

Tabela 118: Diagnósticos de Requisições DNP3 Cliente

Nota:

Diagnósticos de Requisição: De todos os bits disponibilizados na estrutura tINN, 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.

4.5.11.5. 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 de tempo 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.

4.5.11.5.1. Comandos para Saídas Digitais

Os comandos para as saídas digitais são definidos na função **DNP3_BinaryCommand**. Nas tabelas abaixo, pode-se visualizar os parâmetros de entrada e saída do bloco funcional para as saídas digitais.

Parâmetro	Tipo	Descrição
bRequest	BOOL	Quando TRUE, executa o comando
dwVariableAddr	DWORD	Endereço da variável que receberá o comando
eCommand	ENUM (BYTE)	Comando que será enviado: DNP3_COMMAND_TYPE_SELECT (0) DNP3_COMMAND_TYPE_OPERATE (1) DNP3_COMMAND_TYPE_DIRECT_OPERATE (2) DNP3_COMMAND_TYPE_DIRECT_OPERATE_NO_ACK (3)
eOpType	ENUM (BYTE)	DNP3_OP_TYPE_CTRL_NUL (0) DNP3_OP_TYPE_CTRL_PULSE_ON (1) DNP3_OP_TYPE_CTRL_PULSE_OFF (2) DNP3_OP_TYPE_CTRL_LATCH_ON (3) DNP3_OP_TYPE_CTRL_LATCH_OFF (4)
eTripClose	ENUM (BYTE)	DNP3_TRIP_CLOSE_CTRL_NUL (0) DNP3_TRIP_CLOSE_CTRL_PAISED_CLOSE (64) DNP3_TRIP_CLOSE_CTRL_PAISED_TRIP (128)
byCount	BYTE	Este é o número de vezes que o cliente deve executar a operação.
udiOnTime	UNSIGNED DOUBLE INT	Esta é a duração, expressa em milissegundos, para que a saída digital permaneça ligada.
udiOffTime	UNSIGNED DOUBLE INT	Esta é a duração, expressa em milissegundos, para que a saída digital permaneça desligada.
udiCommandTimeOut	UNSIGNED DOUBLE INT	Tempo para o Time Out deste comando (milissegundos).

Tabela 119: Parâmetros de Entrada Bloco DNP3_BinaryCommand

Parâmetro	Tipo	Descrição
eExecStatus	ENUM (BYTE)	Status da execução, indica se um novo comando pode ser enviado, os valores são os seguintes: DNP3_COMMAND_EXEC_STATUS_DONE (0) DNP3_COMMAND_EXEC_STATUS_RUNNING (1)
eErrorStatus	ENUM (BYTE)	Indica se o último comando foi bem sucedido, os valores são os seguintes: DNP3_COMMAND_ERROR_STATUS_NO_ERROR (0) DNP3_COMMAND_ERROR_STATUS_TIMEOUT (1) DNP3_COMMAND_ERROR_STATUS_NO_SELECT (2) DNP3_COMMAND_ERROR_STATUS_FORMAT_ERROR (3) DNP3_COMMAND_ERROR_STATUS_NOT_SUPPORTED (4) DNP3_COMMAND_ERROR_STATUS_ALREADY_ACTIVE (5) DNP3_COMMAND_ERROR_STATUS_HARDWARE_ERROR (6) DNP3_COMMAND_ERROR_STATUS_UNDEFINED (7) DNP3_COMMAND_ERROR_STATUS_INVALID_ADDRESS (8) DNP3_COMMAND_ERROR_STATUS_WRONG_TIMEOUT (9) DNP3_COMMAND_ERROR_STATUS_INVALID_VALUE (10) DNP3_COMMAND_ERROR_STATUS_UNINITIALIZED (11)

Tabela 120: Parâmetros de Saída Bloco DNP3_BinaryCommand

Nota:

eTripClose: Os comandos de trip/fechamento são exclusivos para os módulos de saída digital que suportem essa funcionalidade. Ex. Módulo HX2320 da Série Hadron Xtorm. Caso o usuário esteja utilizando barramento de expansão misto da Série Hadron Xtorm com a Série Nexto, e utilize um módulo de saída digital que não suporte essa funcionalidade, o código de erro retornado será o "DNP3_COMMAND_ERROR_STATUS_NOT_SUPPORTED".

4.5.11.5.2. Comandos para Saídas Analógicas

Os comandos para as saídas analógicas são definidos em duas funções: **DNP3_AnalogCommandInt** e **DNP3_AnalogCommandReal**. Nas tabelas abaixo, podemos visualizar os parâmetros de entrada e saída para cada um dos blocos de função para as saídas analógicas.

Nome do Bloco de Função: **DNP3_AnalogCommandInt**, usado para números inteiros 16 ou 32 bits. Usados no para o grupo G41V1 e G41V2.

Parâmetro	Tipo	Descrição
bRequest	BOOL	Quando TRUE, executa o comando.
dwVariableAddr	DWORD	Endereço da variável de comando para enviar.
eCommand	ENUM (BYTE)	Comando que será enviado: DNP3_COMMAND_TYPE_SELECT (0) DNP3_COMMAND_TYPE_OPERATE (1) DNP3_COMMAND_TYPE_DIRECT_OPERATE (2) DNP3_COMMAND_TYPE_DIRECT_OPERATE_NO_ACK (3)
dValue	DOUBLE INT	Valor analógico que será escrito: INT ou DINT.
udiCommandTimeOut	UNSIGNED DOUBLE INT	Tempo para o Time Out deste comando (milissegundos).

Tabela 121: Parâmetros de Entrada Bloco DNP3_AnalogCommandInt

Parâmetro	Tipo	Descrição
eExecStatus	ENUM (BYTE)	Status da execução, indica se um novo comando pode ser enviado, os valores são os seguintes: DNP3_COMMAND_EXEC_STATUS_DONE (0) DNP3_COMMAND_EXEC_STATUS_RUNNING (1)
eErrorStatus	ENUM (BYTE)	Indica se o último comando foi bem sucedido, os valores são os seguintes: DNP3_COMMAND_ERROR_STATUS_NO_ERROR (0) DNP3_COMMAND_ERROR_STATUS_TIMEOUT (1) DNP3_COMMAND_ERROR_STATUS_NO_SELECT (2) DNP3_COMMAND_ERROR_STATUS_FORMAT_ERROR (3) DNP3_COMMAND_ERROR_STATUS_NOT_SUPPORTED (4) DNP3_COMMAND_ERROR_STATUS_ALREADY_ACTIVE (5) DNP3_COMMAND_ERROR_STATUS_HARDWARE_ERROR (6) DNP3_COMMAND_ERROR_STATUS_UNDEFINED (7) DNP3_COMMAND_ERROR_STATUS_INVALID_ADDRESS (8) DNP3_COMMAND_ERROR_STATUS_WRONG_TIMEOUT (9) DNP3_COMMAND_ERROR_STATUS_INVALID_VALUE (10) DNP3_COMMAND_ERROR_STATUS_UNINITIALIZED (11)

Tabela 122: Parâmetros de Saída Bloco DNP3_AnalogCommandInt

Nome do Bloco de Função: **DNP3_AnalogCommandReal**, usado para números do tipo real 32 bits. Usado no para o grupo G41V3.

Parâmetro	Tipo	Descrição
bRequest	BOOL	Quando TRUE, executa o comando.
dwVariableAddr	DWORD	Endereço da variável de comando para enviar.
eCommand	ENUM (BYTE)	Comando que será enviado: DNP3_COMMAND_TYPE_SELECT (0) DNP3_COMMAND_TYPE_OPERATE (1) DNP3_COMMAND_TYPE_DIRECT_OPERATE (2) DNP3_COMMAND_TYPE_DIRECT_OPERATE_NO_ACK (3)
rValue	REAL	Valor analógico que será escrito: REAL.
udiCommandTimeOut	UNSIGNED DOUBLE INT	Tempo para o Time Out deste comando (milissegundos).

Tabela 123: Parâmetros de Entrada do Bloco DNP3_AnalogCommandReal

Parâmetro	Tipo	Descrição
eExecStatus	ENUM (BYTE)	Status da execução, indica se um novo comando pode ser enviado, os valores são os seguintes: DNP3_COMMAND_EXEC_STATUS_DONE (0) DNP3_COMMAND_EXEC_STATUS_RUNNING (1)
eErrorStatus	ENUM (BYTE)	Indica se o último comando foi bem sucedido, os valores são os seguintes: DNP3_COMMAND_ERROR_STATUS_NO_ERROR (0) DNP3_COMMAND_ERROR_STATUS_TIMEOUT (1) DNP3_COMMAND_ERROR_STATUS_NO_SELECT (2) DNP3_COMMAND_ERROR_STATUS_FORMAT_ERROR (3) DNP3_COMMAND_ERROR_STATUS_NOT_SUPPORTED (4) DNP3_COMMAND_ERROR_STATUS_ALREADY_ACTIVE (5) DNP3_COMMAND_ERROR_STATUS_HARDWARE_ERROR (6) DNP3_COMMAND_ERROR_STATUS_UNDEFINED (7) DNP3_COMMAND_ERROR_STATUS_INVALID_ADDRESS (8) DNP3_COMMAND_ERROR_STATUS_WRONG_TIMEOUT (9) DNP3_COMMAND_ERROR_STATUS_INVALID_VALUE (10) DNP3_COMMAND_ERROR_STATUS_UNINITIALIZED (11)

Tabela 124: Parâmetros de Saída do Bloco DNP3_AnalogCommandReal

4.5.11.5.3. Comando Cold Restart

Nome do Bloco de Função: **DNP3_ColdCommand**.

Parâmetro	Tipo	Descrição
bRequest	BOOL	Quando TRUE, executa o comando.
dwVariableAddr	DWORD	Endereço da variável de comando para enviar.
udiCommandTimeOut	UNSIGNED DOUBLE INT	Tempo para o Time Out deste comando (milissegundos).

Tabela 125: Parâmetros de Entrada do Bloco DNP3_ColdCommand

Parâmetro	Tipo	Descrição
eExecStatus	ENUM (BYTE)	Status da execução, indica se um novo comando pode ser enviado, os valores são os seguintes: DNP3_COMMAND_EXEC_STATUS_DONE (0) DNP3_COMMAND_EXEC_STATUS_RUNNING (1)
eErrorStatus	ENUM (BYTE)	Indica se o último comando foi bem sucedido, os valores são os seguintes: DNP3_COMMAND_ERROR_STATUS_NO_ERROR (0) DNP3_COMMAND_ERROR_STATUS_TIMEOUT (1) DNP3_COMMAND_ERROR_STATUS_NO_SELECT (2) DNP3_COMMAND_ERROR_STATUS_FORMAT_ERROR (3) DNP3_COMMAND_ERROR_STATUS_NOT_SUPPORTED (4) DNP3_COMMAND_ERROR_STATUS_ALREADY_ACTIVE (5) DNP3_COMMAND_ERROR_STATUS_HARDWARE_ERROR (6) DNP3_COMMAND_ERROR_STATUS_UNDEFINED (7) DNP3_COMMAND_ERROR_STATUS_INVALID_ADDRESS (8) DNP3_COMMAND_ERROR_STATUS_WRONG_TIMEOUT (9) DNP3_COMMAND_ERROR_STATUS_INVALID_VALUE (10) DNP3_COMMAND_ERROR_STATUS_UNINITIALIZED (11)

Tabela 126: Parâmetros de Saída do Bloco DNP3_ColdCommand

Exemplo de Aplicação:

Abaixo um exemplo de aplicação utilizando a linguagem ST, na qual o usuário envia o comando *Select Before Operate* para um IED via DNP3.

```

PROGRAM UserPrg
VAR
  iDNPClientAO: INT;
  iDNPClientAOCmd: LibDNP3.DNP3_AnalogCommandInt;
END_VAR

iDNPClientAOCmd.dwVariableAddr:= ADR(iDNPClientAO);
iDNPClientAOCmd();
IF iDNPClientAOCmd.eExecStatus = LibDNP3.DNP3_COMMAND_EXEC_STATUS_DONE AND
iDNPClientAOCmd.bRequest = TRUE THEN
  IF iDNPClientAOCmd.eCommand = LibDNP3.DNP3_COMMAND_TYPE_SELECT AND
  iDNPClientAOCmd.eErrorStatus = LibDNP3.DNP3_COMMAND_ERROR_STATUS_NO_ERROR
  THEN
    iDNPClientAOCmd.eCommand:= LibDNP3.DNP3_COMMAND_TYPE_OPERATE;
  ELSE
    iDNPClientAOCmd.bRequest:= FALSE;
  END_IF
END_IF

```

4.5.11.6. Códigos de Status para Comandos DNP3

Os comandos DNP3 retornam status de erro (*eErrorStatus*) ao serem executados. A tabela abaixo mostra a lista e a descrição dos possíveis valores para o status de erro.

Código	Status	Descrição
0	DNP3_COMMAND_ERROR_STATUS_NO_ERROR	Pedido aceito, iniciado ou na fila.
1	DNP3_COMMAND_ERROR_STATUS_TIMEOUT	Pedido não aceito porque a mensagem foi recebida após o time out do comando select.
2	DNP3_COMMAND_ERROR_STATUS_NO_SELECT	Pedido não aceito porque o comando select não foi enviado previamente.
3	DNP3_COMMAND_ERROR_STATUS_FORMAT_ERROR	Pedido não aceito porque havia erros de formatação do pedido de controle (select, operate ou direct operate).
4	DNP3_COMMAND_ERROR_STATUS_NOT_SUPPORTED	Pedido não aceito, porque uma operação de controle não é suportada por este ponto.
5	DNP3_COMMAND_ERROR_STATUS_ALREADY_ACTIVE	Pedido não aceito porque a fila de controle já está cheia ou o ponto já está ativo.
6	DNP3_COMMAND_ERROR_STATUS_HARDWARE_ERROR	Pedido não aceito, por causa de problemas de hardware.
7	DNP3_COMMAND_ERROR_STATUS_UNDEFINED	Pedido não aceite por causa de algum outro motivo indefinido.
8	DNP3_COMMAND_ERROR_STATUS_INVALID_ADDRESS	Pedido não aceito porque o endereço da variável é inválido.
9	DNP3_COMMAND_ERROR_STATUS_WRONG_TIMEOUT	Pedido não aceito porque o valor de time out não é válido.
10	DNP3_COMMAND_ERROR_STATUS_INVALID_VALUE	Pedido não aceito porque o valor não é válido.
11	DNP3_COMMAND_ERROR_STATUS_UNINITIALIZED	Pedido não aceito porque o bloco funcional não foi iniciado corretamente.

Tabela 127: Códigos de Status para Comandos DNP3

4.5.12. 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:

4. CONFIGURAÇÃO

1. Adicionar a instância do protocolo DNP3 Servidor a um dos canais Ethernet disponíveis (NET 1 .. 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 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 devidos parâmetros.
5. 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.
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 a fragmentação de respostas, mensagens de confirmação, polling de integridade, sincronismo, comandos e fila de eventos.
8. Configurar os parâmetros das mensagens não solicitadas, especificando os parâmetros de controle e as condições de disparo.

As descrições de cada configuração estão relacionadas a seguir, nesta seção.

4.5.12.1. Configuração dos Mapeamentos DNP3 Servidor

As configurações dos Parâmetros Gerais de um DNP3 Servidor conforme a figura abaixo, seguem os parâmetros apresentados na tabela a seguir.

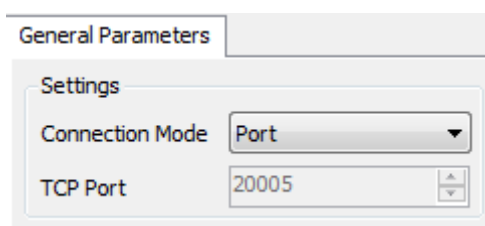


Figura 93: Tela de Parâmetros Gerais do DNP3 Servidor

Configuração	Descrição	Padrão	Opções
Connection Mode	Configura o modo de conexão com os módulos Cliente Conectados. Os modos disponíveis para conexão são "IP" ou "Porta".	Port	Port IP
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".	20005	1 a 65535

Tabela 128: Configuração dos Parâmetros Gerais do DNP3 Servidor

As configurações das relações DNP3 Servidor, visualizadas na figura abaixo, seguem os parâmetros descritos na tabela a seguir.

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 94: Tela de Mapeamentos de Dados DNP3 Servidor

4. CONFIGURAÇÃO

Configuração	Descrição	Padrão	Opções
Value Variable	Nome da variável simbólica.	-	Nome de uma variável declarada em um programa ou GVL.
Static Group and Default Variation	Configuração da variação dos dados estáticos.	-	<p>g10v01 - Saída digital em formato compactado.</p> <p>g10v02 - Saída digital com qualidade.</p> <p>g01v01 - Entrada digital em formato compactado</p> <p>g01v02 - Entrada digital com qualidade</p> <p>g20v01 - Contador de 32 bits com qualidade.</p> <p>g20v02 - Contador de 16 bits com qualidade.</p> <p>g20v05 - Contador de 32 bits sem qualidade.</p> <p>g20v06 - Contador de 16 bits sem qualidade.</p> <p>g21v01 - Contador congelado de 32 bits com qualidade.</p> <p>g21v02 - Contador congelado de 16 bits com qualidade.</p> <p>g21v09 - Contador congelado de 32 bits sem qualidade.</p> <p>g21v10 - Contador congelado de 16 bits sem qualidade.</p> <p>g30v01 - Entrada analógica de 32 bits com qualidade.</p> <p>g30v02 - Entrada analógica de 16 bits com qualidade.</p> <p>g30v03 - Entrada analógica de 32 bits sem qualidade.</p> <p>g30v04 - Entrada analógica de 16 bits sem qualidade.</p> <p>g30v05 - Entrada analógica de precisão simples com qualidade.</p> <p>g03v01 – Entrada digital dupla em formato compactado.</p> <p>g03v02 – Entrada digital dupla com qualidade</p> <p>g40v01 – Saída analógica de 32 bits com qualidade</p> <p>g40v02 – Saída analógica de 16 bits com qualidade</p> <p>g40v03 – Saída analógica de precisão simples com qualidade</p>
Event Group and Default Variation	Configuração da variação para os eventos.	-	<p>g22v01 – Evento do contador de 32 bits com qualidade.</p> <p>g22v02 – Evento do contador de 16 bits com qualidade.</p> <p>g22v05 – Evento do contador de 32 bits com qualidade e tempo.</p> <p>g22v06 – Evento do contador de 16 bits com qualidade e tempo.</p> <p>g23v01 – Evento do contador congelado de 32 bits com qualidade.</p> <p>g23v02 – Evento do contador congelado de 16 bits com qualidade.</p> <p>g23v05 - Evento do contador congelado de 32 bits com qualidade e tempo.</p> <p>g23v06 - Evento do contador congelado de 16 bits com qualidade e tempo.</p> <p>g02v01 – Evento de entrada digital sem tempo.</p> <p>g02v02 – Evento de entrada digital com tempo absoluto.</p> <p>g02v03 – Evento de entrada digital com tempo relativo.</p> <p>g32v01 - Evento de entrada analógica de 32 bits sem tempo.</p> <p>g32v02 - Evento de entrada analógica de 16 bits sem tempo.</p> <p>g32v03 - Evento de entrada analógica de 32 bits com tempo.</p> <p>g32v04 - Evento de entrada analógica de 16 bits com tempo.</p> <p>g32v05 - Evento de entrada analógica de precisão simples sem tempo.</p> <p>g04v01 – Evento de entrada digital dupla sem tempo.</p> <p>g04v02 – Evento de entrada digital dupla com tempo absoluto.</p> <p>g04v03 – Evento de entrada digital com tempo relativo.</p> <p>Nenhum</p>

Configuração	Descrição	Padrão	Opções
Event Class	Configuração da Classe do Evento.	-	Classe 0 Classe 1 Classe 2 Classe 3
Index	Índice do primeiro ponto de mapeamento DNP3.	-	0 a 65535
Size	Tamanho do dado DNP3.	-	- (calculado automaticamente)
Range	Faixa de endereços do dado configurado.	-	- (calculado automaticamente)
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.	-	Nome de uma variável declarada em um programa ou GVL
Dead Band Type	Define o tipo de Banda Morta a ser utilizada no mapeamento.	Desabilitada	Absoluta Desabilitada Integrada

Tabela 129: Configuração dos Mapeamentos DNP3 Servidor

Notas:

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:

Tipo de Função	Configurações	Descrição
Tipo da Banda Morta	Desabilitada	Nesta opção, qualquer modificação de valor num ponto do grupo, por menor que seja, gera evento para este ponto.
	Absoluta	Nesta opção, se o módulo da variação do valor de um ponto do grupo for maior do que o valor configurado pela variável no campo “Banda Morta”, um evento para este ponto é gerado.
	Integrada	Nesta opção, se o módulo da integral da variação do valor de um ponto do grupo for maior do que o valor configurado pela variável no campo “Banda Morta”, um evento para este ponto é gerado. O intervalo de integração é de um segundo.

Tabela 130: Tipos de Banda Morta nos Parâmetros do DNP3 Servidor

ATENÇÃO

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.

4.5.12.2. Configuração da Camada de Enlace DNP3 Servidor

As configurações dos parâmetros da camada de enlace do DNP3 Servidor, visualizadas na figura abaixo, seguem os parâmetros descritos na tabela a seguir.

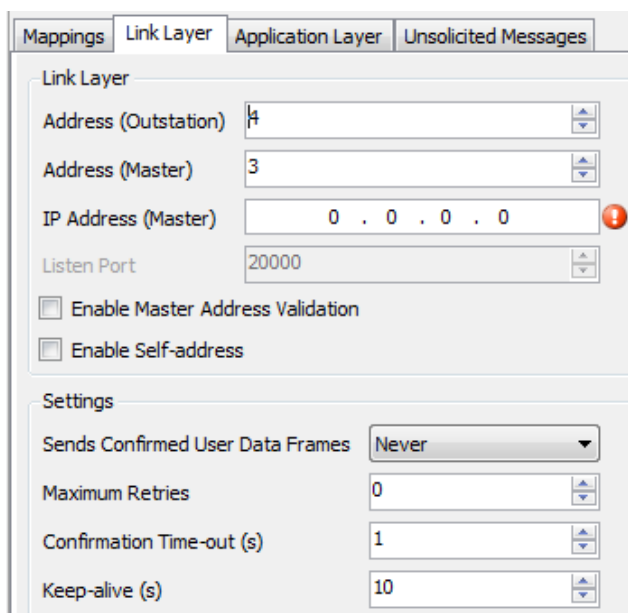


Figura 95: Tela de Configurações da Camada de Enlace DNP3 Servidor

Configuração	Descrição	Padrão	Opções
Address (Outstation)	Endereço DNP3 deste servidor.	4	0 a 65519
Address (Master)	Endereço DNP3 do cliente conectado.	3	0 a 65519
IP Address (Master)	IP do cliente conectado, utilizado quando a conexão do cliente é por IP.	0.0.0.0	1.0.0.1 a 223.255.255.254
Listen Port	Endereço da porta listen para conexão do cliente. Utilizado quando a conexão do cliente não é por IP.	20000	1 a 65535
Enable Master Address Validation	Opção para Habilitar/Desabilitar a validação do endereço de Origem.	Disabled	Enabled Disabled
Enable Self-address	Opção para Habilitar/Desabilitar o Auto Endereçamento.	Disabled	Enabled Disabled
Sends Confirmed User Data Frames	Seleciona o modo de confirmações para as mensagens enviadas para o cliente.	Never	Never Always Sometimes
Maximum Retries	Número Máximo de Tentativas de retransmissão de mensagem.	3	0 a 10
Confirmation Time-out (s)	Tempo de espera para enviar uma nova tentativa.	3	1 a 86400
Keep Alive (s)	Tempo de espera para enviar mensagens de teste de link após receber a mensagem DNP.	10	1 a 86400

Tabela 131: Configuração da Camada de Enlace DNP3 Servidor

Notas:

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:

Tipo de Função	Configurações	Descrição
Modo de Confirmação da Camada de Enlace	Never	Nunca solicitará confirmação das mensagens.
	Always	Sempre solicitará confirmação das mensagens.
	Sometimes	Solicitará confirmação somente para mensagens multifragmentadas.

Tabela 132: Tipos de Modo de Confirmação da Camada de Enlace nos Parâmetros do DNP3 Servidor

Notas:

Maximum Retries: Número de vezes que o DNP3 Servidor irá retransmitir a mensagem caso ele não receba a confirmação do cliente.

Confirmation Time-Out (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.

4.5.12.3. Configuração da Camada de Aplicação DNP3 Servidor

As configurações das relações DNP3 Servidor, visualizadas na figura abaixo, seguem os parâmetros descritos na tabela a seguir.

Figura 96: Tela de Configurações da Camada de Aplicação DNP3 Servidor

Configuração	Descrição	Padrão	Opções
Allow Multiple Fragmented Responses	Opção para Habilitar/Desabilitar a Transmissão de Respostas Múltiplos Fragmentos.	Enabled	Enabled Disabled
Request Confirmation of Multiple Fragmented Responses	Opção para Habilitar/Desabilitar Requisição de Confirmação de Resposta com Múltiplos Fragmentos.	Enabled	Enabled Disabled
TX Fragmented Length (bytes)	Tamanho máximo dos fragmentos.	2048	249 a 2948
Confirmation Time-out (s)	Determina qual é o tempo máximo de espera para as mensagens de confirmação.	5	1 a 86400
Enable Time Synchronization	Opção para Habilitar/Desabilitar a solicitação de sincronismo de tempo.	Disabled	Enabled Disabled
Time Synchronization Command Received in Local Time	Opção para Habilitar/Desabilitar o tratamento do comando de sincronização em horário local.	Disabled	Enabled Disabled
Synchronization Time-out (s)	Define o tempo que o relógio local continuará válido após ter recebido uma sincronização de tempo.	60	1 a 86400

Configuração	Descrição	Padrão	Opções
Use Local Time instead of UTC Time	Opção para Habilitar/Desabilitar a estampa de tempo em horário local para eventos.	Disabled	Enabled Disabled
Maximum Time Between Selected and Operate (s)	Tempo máximo que o comando select permanecerá ativo aguardando o comando Operate.	5	1 a 86400
Transmission Mode of Analog Inputs Events	Seleciona o modo de transmissão dos eventos de entradas analógicas.	All Events (SOE)	All Events (SOE) Most Recent Events

Tabela 133: Configuração da Camada de Aplicação DNP3 Servidor

Notas:

Allow Multiple Fragmented Responses: 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 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): Máximos de tamanhos de fragmentos transmitidos aplicação.

Confirmation Time-out (s): O time-out de mensagens de confirmação é o tempo que o servidor aguardará por uma confirmação de recebimento da mensagem pelo cliente.

Enable Time Synchronization: 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 “Time-Out de Sincronismo”.

Synchronization Time-out (s): Este campo é utilizado para configurar o intervalo de tempo para as solicitações de sincronização de tempo.

Maximum Time Between Selected and Operate (s): Período de tempo no qual o comando de seleção permanecerá ativo (inicia a contagem a partir da confirmação da recepção do comando de seleção).

Transmission Mode of Analog Input Events: Os modos de transmissão dos eventos de entradas analógicas disponíveis são os seguintes:

Tipo de Função	Configurações	Descrição
Transmission Mode of Analog Inputs Events	All Events(SOE)	São enviados todos os eventos analógicos gerados.
	Most Recent Events	É enviado somente o evento mais recente de cada ponto analógico.

Tabela 134: Configurações dos Modos de Transmissão dos Eventos de Entradas Analógicas no DNP3 Servidor

ATENÇÃO

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.

4.5.12.4. Configuração das Mensagens Não Solicitadas DNP3 Servidor

As configurações das relações DNP3 Servidor, visualizadas na figura abaixo, seguem os parâmetros descritos na tabela a seguir.

Figura 97: Tela de Mensagens Não Solicitadas DNP3 Servidor

Configuração	Descrição	Padrão	Opções
Enable Unsolicited Messages	Opção para Habilitar/Desabilitar as mensagens não solicitadas.	Enabled	Enabled Disabled
Number of Unsolicited Retries	Número de tentativas para transmitir a mensagem.	3	0 a 65535
Confirmation Time-out (s)	Intervalo a ser aguardado depois de um time out.	5	1 a 86400
Try Again (s)	Intervalo a ser aguardado após atingir o número máximo de tentativas.	30	1 a 86400
Number of Class 1 Events	Quantidade mínimo de eventos na Classe 1 para disparar uma mensagem não solicitada.	10	1 a 255
Number of Class 2 Events	Quantidade mínimo de eventos na Classe 2 para disparar uma mensagem não solicitada.	10	1 a 255
Number of Class 3 Events	Quantidade mínimo de eventos na Classe 3 para disparar uma mensagem não solicitada.	10	1 a 255
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.	1	1 a 86400
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.	1	1 a 86400

Configuração	Descrição	Padrão	Opções
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.	1	1 a 86400

Tabela 135: Configuração das Mensagens Não Solicitadas DNP3 Servidor

Notas:

Enable Unsolicited Messages: Se essa opção estiver marcada, o DNP3 Servidor irá permitir Mensagens não solicitadas.

Number of Unsolicited Retries: Número de vezes que o DNP3 Servidor irá tentar transmitir a mensagem.

Confirmation Time-out(s): Especifica o tempo a ser aguardado depois de ocorrer um time-out para a mensagem não solicitada que foi transmitida. Transcorrido esse intervalo a mensagem será retransmitida. Logo o tempo de retransmissão, será dado pelos parâmetros : “**Time-out de Mensagens de Confirmação**” + “**Time-Out**”.

Try Again (s): 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 a mensagem não solicitada.

Number of Class 2 Events: Quantidade mínimo de eventos na classe 2 para disparar a mensagem não solicitada.

Number of Class 3 Events: Quantidade mínimo de eventos na classe 3 para disparar a mensagem não solicitada.

Hold Time After Class 1 Event: Tempo máximo que um evento da classe 1 pode ficar armazenado sem ser enviado por uma mensagem não solicitada.

Hold Time After Class 2 Event: Tempo máximo que um evento da classe 2 pode ficar armazenado sem ser enviado por uma mensagem não solicitada.

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

4.5.12.5. Diagnósticos DNP3 Servidor

Os diagnósticos e comandos do protocolo DNP3 Servidor configurado, são armazenados em variáveis do tipo *T_DIAG_DNP_SERVER_I* as quais estão descritas na tabela abaixo.

Variável de diagnóstico do tipo T_DIAG_DNP_SERVER_1.*	Tamanho	Descrição
Bits de comando, reiniciados automaticamente:		
tCommand. bStop	BIT	Desabilitar Driver.
tCommand. bStart	BIT	Habilitar o Driver.
tCommand. bDiag_01_Reserved	BIT	Reservado.
tCommand. bDiag_02_Reserved	BIT	Reservado.
tCommand. bDiag_03_Reserved	BIT	Reservado.
tCommand. bDiag_04_Reserved	BIT	Reservado.
tCommand. bDiag_05_Reserved	BIT	Reservado.
tCommand. bDiag_06_Reserved	BIT	Reservado.

Variável de diagnóstico do tipo T_DIAG_DNP_SERVER_1.*	Tamanho	Descrição
Diagnósticos:		
tClient_X. bRunning	BOOL	DNP3 Servidor está sendo executado.
tClient_X. eConnectionStatus. CLOSED	ENUM (BYTE)	Canal de comunicação fechado. Servidor não irá aceitar solicitação de conexão. Valor do ENUM (0).
tClient_X. eConnectionStatus. LISTENING	ENUM (BYTE)	Servidor está escutando a porta configurada, e não existem clientes conectados. Valor do ENUM (1).
tClient_X. eConnectionStatus. CONNECTED	ENUM (BYTE)	Cliente conectado. Valor do ENUM (2).
tClient_X. tQueueDiags. bOverflow	BOOL	Fila do cliente está com overflow.
tClient_X. tQueueDiags. wSIZE	WORD	Tamanho da fila configurada.
tClient_X. tQueueDiags. wUSAGE	WORD	Número de eventos na fila.
tClient_X. tQueueDiags. dwUnsolEna	DWORD	Máscara que indica para quais classes estão habilitadas as mensagens não solicitadas.
tClient_X. tQueueDiags. dwReserved_1	DWORD	Reservado.
tClient_X. tStats. wRXFrames	WORD	Número de quadros (frames) recebidos.
tClient_X. tStats. wTXFrames	WORD	Número de quadros (frames) enviados.
tClient_X. tStats. wCommErrors	WORD	Contador de erros de comunicação incluindo erros na Camada Física, Camada de Enlace e Camada de Transporte.
tClient_X. tStats. dwReserved_0	DWORD	Reservado.
tClient_X. tStats. dwReserved_1	DWORD	Reservado.

Tabela 136: Diagnósticos DNP3 Servidor

Nota:

dwUnsolEna: Os bits 0, 1 e 2 correspondem respectivamente aos bits de indicação do estado de habilitação das mensagens não solicitadas das classes 1, 2 e 3. Quando ligados indicam que o envio de mensagens não solicitadas foi habilitado para a

correspondente classe.

4.5.12.6. Associação dos Objetos com o Protocolo IEC 60870-5-104

A tabela abaixo apresenta associações permitidas entre os protocolos DNP3 e IEC 60870-5-104.

Grupo e Variação DNP3	Tipo de Objeto IEC 60870-5-104
g01v01 - Entrada digital em formato compactado. g01v02 - Entrada digital com qualidade.	Single Point Information (M_SP_NA)
g03v01 - Entrada digital dupla em formato compactado. g03v02 - Entrada digital dupla com qualidade.	Double Point Information (M_DP_NA)
g10v01 - Saída digital em formato compactado. g10v02 - Saída digital com qualidade.	Single Command (C_SC_NA) Double Command (C_DC_NA)
g30v02 - Entrada analógica de 16 bits com qualidade. g30v04 - Entrada analógica de 16 bits sem qualidade.	Measured Value, normalized value (M_ME_NA) Measured Value, scaled value (M_ME_NB)
g30v01 - Entrada analógica de 32 bits com qualidade. g30v03 - Entrada analógica de 32 bits sem qualidade.	Measured Value, short floating value (M_ME_NC)
g30v05 - Entrada analógica de precisão simples com qualidade.	Measured Value, short floating value (M_ME_NC)
g40v02 - Saída analógica de 16 bits com qualidade.	Setting Point Command, normalized Value (C_SE_NA) Setting Point Command, scaled Value (C_SE_NA)
g40v01 - Saída analógica de 32 bits com qualidade.	-
g40v03 - Saída analógica de precisão simples com qualidade.	Setting Point Command, short floating point Value (C_SE_NC)

Tabela 137: Associação dos Tipos de Objetos DNP3 com Seus Correspondentes no IEC 60870-5-104

ATENÇÃO

g20 (Contadores) e g21 (Contadores Congelados): não é permitido associar uma variável utilizada como contadora, simultaneamente no protocolos DNP3 e IEC 60870-5-104.
g40v01 (Saída analógica de 32 bits com qualidade): não há objeto equivalente para o protocolo IEC 60870-5-104.

4.5.12.7. 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.

4.5.12.8. 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

- Intercepção (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.

4.5.12.8.1. 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.

4.5.12.8.2. 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 intercepção e tratamento destes comandos pela aplicação pode evitar o retorno de mensagens negativas (falha).

4.5.12.8.3. 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 time-out), 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 esteja ocupado executando um comando, será retornado o código de erro 0x05 (already active).

4.5.12.8.4. 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.

4.5.12.8.5. Intercepção de Comandos

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

Abaixo um exemplo de aplicação utilizando a linguagem ST mostrando a intercepção de um comando para o objeto `BlkOpn` de um Logical Node `XCBR`, respondendo `"SUCCESS"` ao IEC 61850 Cliente.

4.5.13. IEC 60870-5-104 - Tipos de Dados

A tabela abaixo mostra o tipo de variável suportada pela HX3040 para cada um dos tipos de dados do protocolo IEC 60870-5-104.

Tipo do Objeto	Tipos de Variáveis IEC
Single Point Information (M_SP_NA)	BOOL BIT
Double Point Information (M_DP_NA)	DBP
Step Position Information (M_ST_NA)	USINT
Measured Value, normalized value (M_ME_NA)	INT
Measured Value, scaled value (M_ME_NB)	INT
Measured Value, short floating point value (M_ME_NC)	INT UINT DINT UDINT REAL
Integrated Totals (M_IT_NA)	INT DINT
Bitstring Information (M_BO_NA)	DWORD
Single Command (C_SC_NA)	BOOL BIT
Double Command (C_DC_NA)	DBP
Regulating Step Command (C_RC_NA)	DBP
Setting Point Command, normalized Value (C_SE_NA)	INT
Setting Point Command, scaled Value (C_SE_NB)	INT
Setting Point Command, short floating point Value (C_SE_NC)	REAL
Bitstring Command (C_BO_NA)	DWORD

Tabela 138: Declaração de Variáveis para IEC 60870-5-104

Nota:

Regulating Step Command: Os estados *Lower* e *Higher* do objeto C_RC_NA estão associados respectivamente aos estados *OFF* e *ON* do tipo interno DBP.

Step Position Information: Conforme definido no item 7.3.1.5 da norma IEC 60870-5-101 essa variável de 8 bits é composta por dois campos : valor (definido pelos 7 bits menos significativos da variável) e transiente (definido como o bit mais significativo, o qual indica quando o dispositivo medido está transicionando).

Abaixo, exemplo de código para manipulação dos campos em uma variável do tipo USINT. Este código não consiste se o valor de entrada está dentro da faixa (entre -64 e + 63), portanto esta consistência fica a cargo do usuário.

```
PROGRAM UserPrg
VAR
  usiVTI: USINT; // Value transient indication (client mapping)
  siValue: SINT; // Value to convert to VTI
  bTransient: BOOL; // Transient to be converted to VTI
END_VAR

usiVTI:= SINT_TO_USINT(siValue) AND 16#3F;
IF siValue < 0 THEN
  usiVTI:= usiVTI OR 16#40;
END_IF
IF bTransient THEN
  usiVTI:= usiVTI OR 16#80;
END_IF
```

4.5.14. 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 pela UCP HX3040.

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

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

1. Selecionar a interface Ethernet (NET1 .. 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, conforme ilustra a figura 98. 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*, conforme ilustra a figura 99. Existe um limite máximo de 5 setores para cada dispositivo.
6. Executar as seguintes configurações para cada setor adicionado:
 - Adicionar e configurar os mapeamentos IEC 60870-5-104 do setor;
 - Configurar os parâmetros da camada de enlace do setor;
 - Configurar os parâmetros da camada de aplicação do setor.

As descrições de cada configuração estão relacionadas a seguir, nesta seção.

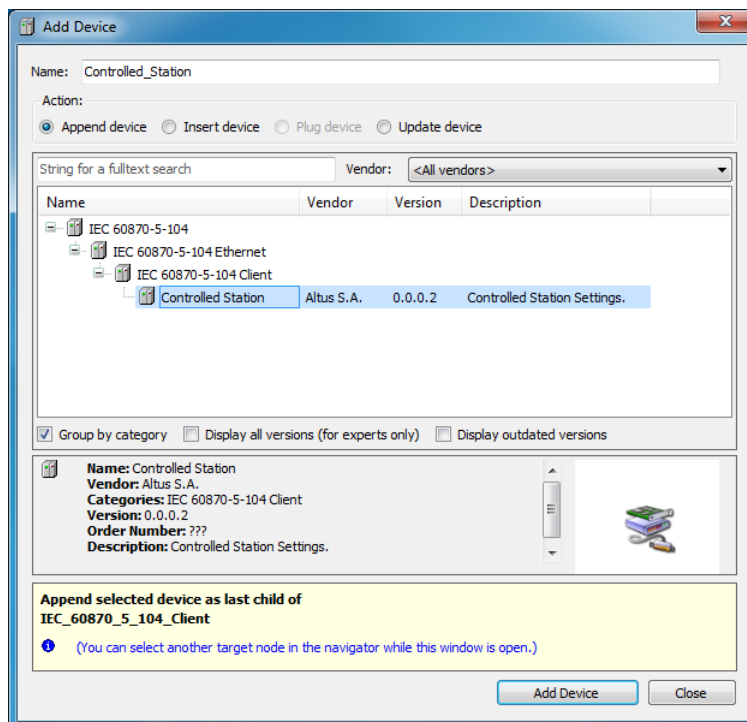


Figura 98: Tela de Inserção de Dispositivo Controlled Station Abaixo de um IEC 60870-5-104 Cliente

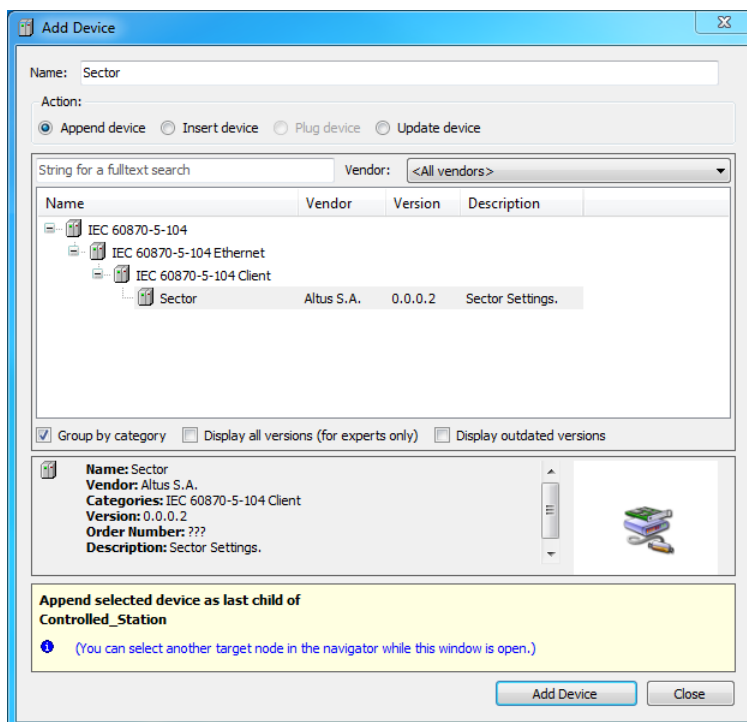


Figura 99: Tela de Inserção de Setor Abaixo de um Dispositivo Controlled Station IEC 60870-5-104

4.5.14.1. 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, é visualizada na figura abaixo. A tabela a seguir, descreve os campos que devem ser preenchidos nesta tela.

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 100: Tela de Configuração da Camada de Enlace de uma Controlled Station IEC 60870-5-104

Configuração	Descrição	Padrão	Opções
Port Number	Endereço da porta TCP listen para conexão com esta Controlled Station.	2404	0 a 65535
IP Address	Endereço IP desta Controlled Station.	0.0.0.0	1.0.0.1 a 223.255.255.254
Time-out t1(s)	Período de tempo (em segundos) que o dispositivo aguarda o recebimento de uma mensagem de confirmação após o envio de uma mensagem APDU do tipo I ou U (dados) antes de fechar a conexão.	15	1 a 180
Time-out t2(s)	Período de tempo (em segundos) que o dispositivo aguarda para enviar uma mensagem supervisória (S-Frame) confirmando o recebimento de frames de dados.	10	1 a 180
Time-out t3(s)	Período de tempo (em segundos) em que será enviada uma mensagem para teste do link caso não haja nenhuma transmissão por ambos os lados.	20	1 a 180
Parameter k (APDUs)	Número máximo de mensagens de dados (I-Frame) transmitidas e não confirmadas.	12	1 a 12
Parameter w (APDUs)	Número máximo de mensagens de dados (I-Frame) recebidas e não confirmadas.	8	1 a 8

Tabela 139: Configuração da Camada de Enlace de uma Controlled Station

4.5.14.2. 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, é visualizada na figura abaixo. A tabela a seguir, descreve as colunas que devem ser preenchidas para cada mapeamento.

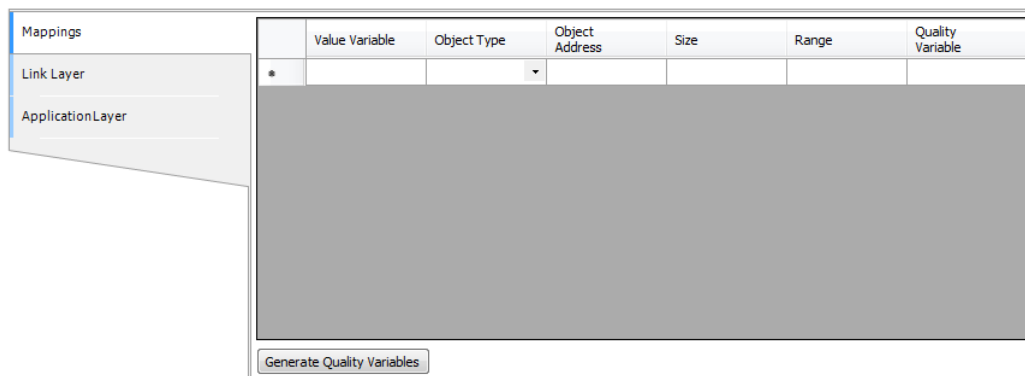


Figura 101: Tela de Mapeamentos de Dados de um Setor IEC 60870-5-104

Configuração	Descrição	Padrão	Opções
Value Variable	Nome da variável simbólica.	-	Nome de uma variável declarada em um programa ou GVL.
Object Type	Configuração do tipo do objeto IEC 60870-5-104.	-	Single Point Information. Double Point Information. Step Position Information. Measured Value (Normalized). Measured Value (Scaled). Measured Value (Short Floating Point). Integrated Totals. Single Command. Double Command. Regulating Step Command. Setting Point Command (Normalized). Setting Point Command (Scaled). Setting Point Command (Short Floating Point).
Object Address	Índice do primeiro ponto do mapeamento IEC 60870-5-104.	-	1 a 65535
Size	Especifica a quantidade máxima de dados que um mapeamento IEC 60870-5-104 poderá acessar.	-	- (calculado automaticamente)
Range	Faixa de endereços do dado configurado.	-	- (calculado automaticamente)
Quality Variable	Nome da variável simbólica que armazenará a qualidade do mapeamento.	-	Variável do tipo QUALITY
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 .	-	-

Tabela 140: Configuração dos Mapeamentos de Dados de um Setor IEC 60870-5-104

Notas:

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: 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: 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 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. 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.

ATENÇÃO

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.

4.5.14.3. 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, é visualizada na figura abaixo. A tabela a seguir, descreve os campos que devem ser preenchidos nesta tela.

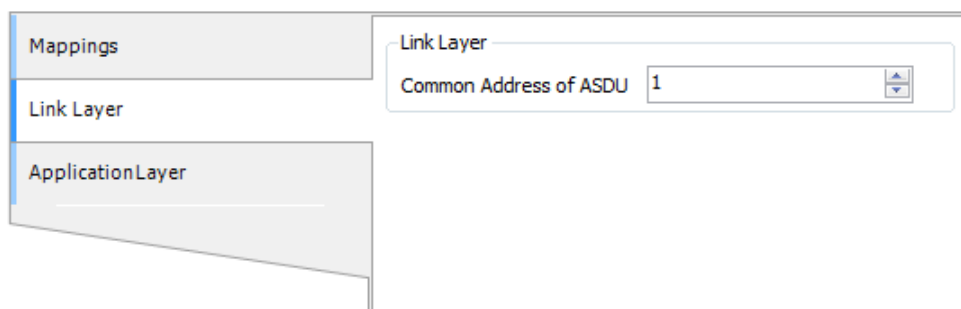


Figura 102: Tela de Configurações da Camada de Enlace de um Setor IEC 60870-5-104

Configuração	Descrição	Padrão de Fábrica	Possibilidades
Common Address of ASDU	Endereço IEC 60870-5-104 deste setor.	1	1 a 65534

Tabela 141: Configuração da Camada de Enlace de um Setor IEC 60870-5-104

4.5.14.4. Configuração da Camada de Aplicação 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, é visualizada na figura abaixo. A tabela a seguir, descreve os campos que devem ser preenchidos nesta tela.

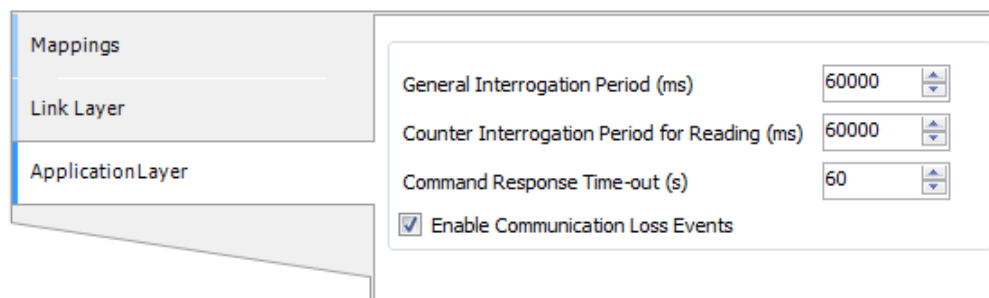


Figura 103: Tela de Configurações da Camada de Aplicação IEC 60870-5-104 Cliente

Configuração	Descrição	Padrão	Opções
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.	60000	0 desabilita. 1 a 86400000. [milissegundos]
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.	60000	0 desabilita. 1 a 86400000. [milissegundos]
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.	60	1 a 3600. [segundos]
Enable Communication Loss Events	Habilita a geração de eventos de perda e restabelecimento de comunicação com as Controlled Stations.	Enabled	Enabled. Disabled.

Tabela 142: Configuração da Camada de Aplicação IEC 60870-5-104 Cliente

4.5.14.5. Diagnósticos do IEC 60870-5-104 Cliente

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 as quais estão descritas na tabela 143. Os diagnósticos para Controlled Station são armazenados em variáveis do tipo T_DIAG_IEC104_CONTROLLED_STATION_1, as quais estão descritas na tabela 144.

Variável de diagnóstico do tipo T_DIAG_IEC104_CLIENT_1.	Tamanho	Descrição
Bits de comando, reiniciados automaticamente:		
tCommand. bStop	BIT	Desabilitar Driver.
tCommand. bStart	BIT	Habilitar o Driver.
tCommand. bDiag_01_Reserved	BIT	Reservado.

Variável de diagnóstico do tipo T_DIAG_IEC104_CLIENT_1.	Tamanho	Descrição
tCommand. bDiag_02_Reserved	BIT	Reservado.
tCommand. bDiag_03_Reserved	BIT	Reservado.
tCommand. bDiag_04_Reserved	BIT	Reservado.
tCommand. bDiag_05_Reserved	BIT	Reservado.
tCommand. bDiag_06_Reserved	BIT	Reservado.
Diagnósticos:		
bRunning	BOOL	IEC 60870-5-104 Cliente está sendo executado.

Tabela 143: Diagnósticos do IEC 60870-5-104 Cliente

Variável de diagnóstico do tipo T_DIAG_IEC104_CONTROLLED_STATION_1.	Tamanho	Descrição
Diagnósticos de conexão:		
eConnectionStatus. CLOSED	BYTE	Comunicação fechada com o IED.
eConnectionStatus. CONNECTED	BYTE	Comunicação operante com o IED.
Estatísticas de comunicação:		
tStat. wRXFrames	WORD	Número de quadros (frames) recebidos.
tStat. wTXFrames	WORD	Número de quadros (frames) enviados.
tStat. wCommErrors	WORD	Contador de erros de comunicação incluindo erros na Camada Física, Camada de Enlace e Camada de Transporte.
tStat. wGeneralInterrogationErrors	WORD	Contador de erros no comando "General Interrogation" cíclico.
tStat. wCounterInterrogationErrors	WORD	Contador de erros no comando "Counter Interrogation" cíclico.
tStat. dwReserved_0	DWORD	Reservado.
tStat. dwReserved_1	DWORD	Reservado.

Tabela 144: Diagnósticos da Controlled Station IEC 60870-5-104

4.5.14.6. 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 de tempo 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.

4.5.14.6.1. 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)

Nas tabelas seguintes pode-se visualizar os parâmetros de entrada e saída deste bloco funcional.

Parâmetro	Tipo	Descrição
bRequest	BOOL	Quando TRUE, executa o comando.
dwVariableAddr	DWORD	Endereço da variável que receberá o comando.
eCommand	ENUM (BYTE)	Comando que será enviado: IEC104_COMMAND_TYPE_SELECT (0) IEC104_COMMAND_TYPE_EXECUTE (1) IEC104_COMMAND_TYPE_DEACTIVATE (2)
eQualifier	ENUM (BYTE)	IEC104_QUALIFIER_DEFAULT (0) IEC104_QUALIFIER_SHORT_PULSE (1) IEC104_QUALIFIER_LONG_PULSE (2) IEC104_QUALIFIER_PERSISTENT (3)
eValue	ENUM (BYTE)	IEC104_VALUE_OFF (0) IEC104_VALUE_ON (1) IEC104_VALUE_LOWER (2) IEC104_VALUE_HIGHER (3)
udiCommandTimeout	UDINT	Tempo para o Time Out deste comando.

Tabela 145: Parâmetros de Entrada Bloco Funcional IEC104_DigitalCommand

Parâmetro	Tipo	Descrição
eExecStatus	ENUM (BYTE)	Status da execução, indica se um novo comando pode ser enviado, os valores são os seguintes: IEC104_COMMAND_EXEC_STATUS_DONE (0) IEC104_COMMAND_EXEC_STATUS_RUNNING (1)
eErrorStatus	ENUM (BYTE)	Indica se o último comando foi bem sucedido. IEC104_COMMAND_ERROR_STATUS_NO_ERROR (0) IEC104_COMMAND_ERROR_STATUS_TIMEOUT (1) IEC104_COMMAND_ERROR_STATUS_FORMAT_ERROR (2) IEC104_COMMAND_ERROR_STATUS_NOT_SUPPORTED (3) IEC104_COMMAND_ERROR_STATUS_INTERNAL_ERROR (4) IEC104_COMMAND_ERROR_STATUS_STATION_ERROR (5) IEC104_COMMAND_ERROR_STATUS_INVALID_ADDRESS (6) IEC104_COMMAND_ERROR_STATUS_WRONG_TIMEOUT (7) IEC104_COMMAND_ERROR_STATUS_UNINITIALIZED (8)

Tabela 146: Parâmetros de Saída Bloco Funcional IEC104_DigitalCommand

Notas:■ **eValue:**

- Os valores IEC104_VALUE_ON e IEC104_VALUE_OFF devem ser utilizados exclusivamente para objetos dos tipos Single Command (C_SC_NA) e Double Command (C_DC_NA);
- Os valores IEC104_VALUE_LOWER e IEC104_VALUE_HIGHER devem ser utilizados exclusivamente para objetos do tipo Regulating Step Command (C_RC_NA).

4.5.14.6.2. Comandos para Saídas Analógicas

Os comandos para as saídas analógicas 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)

Nas tabelas seguintes pode-se visualizar os parâmetros de entrada e saída do bloco funcional **IEC104_AnalogCommandInt**.

Parâmetro	Tipo	Descrição
bRequest	BOOL	Quando TRUE, executa o comando.
dwVariableAddr	DWORD	Endereço da variável de comando para enviar.
eCommand	ENUM (BYTE)	Comando que será enviado: IEC104_COMMAND_TYPE_SELECT (0) IEC104_COMMAND_TYPE_EXECUTE (1) IEC104_COMMAND_TYPE_DEACTIVATE (2)
iValue	INT	Valor analógico que será escrito: INT.
udiCommandTimeOut	UDINT	Tempo para o Time Out deste comando.

Tabela 147: Parâmetros de Entrada Bloco IEC104_AnalogCommandInt

Parâmetro	Tipo	Descrição
eExecStatus	ENUM (BYTE)	Status da execução, indica se um novo comando pode ser enviado, os valores são os seguintes: IEC104_COMMAND_EXEC_STATUS_DONE (0) IEC104_COMMAND_EXEC_STATUS_RUNNING (1)
eErrorStatus	ENUM (BYTE)	Indica se o último comando foi bem sucedido. IEC104_COMMAND_ERROR_STATUS_NO_ERROR (0) IEC104_COMMAND_ERROR_STATUS_TIMEOUT (1) IEC104_COMMAND_ERROR_STATUS_FORMAT_ERROR (2) IEC104_COMMAND_ERROR_STATUS_NOT_SUPPORTED (3) IEC104_COMMAND_ERROR_STATUS_INTERNAL_ERROR (4) IEC104_COMMAND_ERROR_STATUS_STATION_ERROR (5) IEC104_COMMAND_ERROR_STATUS_INVALID_ADDRESS (6) IEC104_COMMAND_ERROR_STATUS_WRONG_TIMEOUT (7) IEC104_COMMAND_ERROR_STATUS_UNINITIALIZED (8)

Tabela 148: Parâmetros de Saída Bloco IEC104_AnalogCommandInt

O bloco funcional **IEC104_AnalogCommandReal** permite executar comandos sobre os seguintes tipos de objetos:

- Setting Point Command, short floating point Value (C_SE_NC).

Nas tabelas seguintes pode-se visualizar os parâmetros de entrada e saída do bloco funcional **IEC104_AnalogCommandReal**.

Parâmetro	Tipo	Descrição
bRequest	BOOL	Quando TRUE, executa o comando.
dwVariableAddr	DWORD	Endereço da variável de comando para enviar.
eCommand	ENUM (BYTE)	Comando que será enviado: IEC104_COMMAND_TYPE_SELECT (0) IEC104_COMMAND_TYPE_EXECUTE (1) IEC104_COMMAND_TYPE_DEACTIVATE (2)
rValue	REAL	Valor analógico que será escrito: REAL.
udiCommandTimeOut	UDINT	Tempo para o Time Out deste comando.

Tabela 149: Parâmetros de Entrada Bloco IEC104_AnalogCommandReal

Parâmetro	Tipo	Descrição
eExecStatus	ENUM (BYTE)	Status da execução, indica se um novo comando pode ser enviado, os valores são os seguintes: IEC104_COMMAND_EXEC_STATUS_DONE (0) IEC104_COMMAND_EXEC_STATUS_RUNNING (1)
eErrorStatus	ENUM (BYTE)	Indica se o último comando foi bem sucedido. IEC104_COMMAND_ERROR_STATUS_NO_ERROR (0) IEC104_COMMAND_ERROR_STATUS_TIMEOUT (1) IEC104_COMMAND_ERROR_STATUS_FORMAT_ERROR (2) IEC104_COMMAND_ERROR_STATUS_NOT_SUPPORTED (3) IEC104_COMMAND_ERROR_STATUS_INTERNAL_ERROR (4) IEC104_COMMAND_ERROR_STATUS_STATION_ERROR (5) IEC104_COMMAND_ERROR_STATUS_INVALID_ADDRESS (6) IEC104_COMMAND_ERROR_STATUS_WRONG_TIMEOUT (7) IEC104_COMMAND_ERROR_STATUS_UNINITIALIZED (8)

Tabela 150: Parâmetros de Saída Bloco IEC104_AnalogCommandReal

4.5.14.6.3. Comando Reset Process

O bloco funcional **IEC104_ResetProcess** permite reinicializar um dispositivo IEC 60870-5-104 Servidor. Nas tabelas seguintes pode-se visualizar os parâmetros de entrada e saída do bloco funcional.

Parâmetro	Tipo	Descrição
bRequest	BOOL	Quando TRUE, executa o comando.
dwVariableAddr	DWORD	Endereço de qualquer variável mapeada no servidor, usada apenas para identificar o servidor.
udiCommandTimeOut	UDINT	Tempo para o Time Out deste comando.

Tabela 151: Parâmetros de Entrada Bloco IEC104_ResetProcess

Parâmetro	Tipo	Descrição
eExecStatus	ENUM (BYTE)	Status da execução, indica se um novo comando pode ser enviado, os valores são os seguintes: IEC104_COMMAND_EXEC_STATUS_DONE (0) IEC104_COMMAND_EXEC_STATUS_RUNNING (1)
eErrorStatus	ENUM (BYTE)	Indica se o último comando foi bem sucedido. IEC104_COMMAND_ERROR_STATUS_NO_ERROR (0) IEC104_COMMAND_ERROR_STATUS_TIMEOUT (1) IEC104_COMMAND_ERROR_STATUS_FORMAT_ERROR (2) IEC104_COMMAND_ERROR_STATUS_NOT_SUPPORTED (3) IEC104_COMMAND_ERROR_STATUS_INTERNAL_ERROR (4) IEC104_COMMAND_ERROR_STATUS_STATION_ERROR (5) IEC104_COMMAND_ERROR_STATUS_INVALID_ADDRESS (6) IEC104_COMMAND_ERROR_STATUS_WRONG_TIMEOUT (7) IEC104_COMMAND_ERROR_STATUS_UNINITIALIZED (8)

Tabela 152: Parâmetros de Saída Bloco IEC104_ResetProcess

4.5.14.6.4. Comando Synchronize IED

Através do bloco funcional **IEC104_SynchronizeIED** é possível comandar um sincronismo do relógio de um IED (IEC 60870-5-104 Servidor).

Nas tabelas seguintes pode-se visualizar os parâmetros de entrada e saída do bloco funcional.

Parâmetro	Tipo	Descrição
bRequest	BOOL	Quando TRUE, executa o comando.
dwVariableAddr	DWORD	Endereço de qualquer variável mapeada no servidor, usada apenas para identificar o servidor.
udiCommandTimeOut	UDINT	Tempo para o Time Out deste comando.

Tabela 153: Parâmetros de Entrada Bloco IEC104_SynchronizeIED

Parâmetro	Tipo	Descrição
eExecStatus	ENUM (BYTE)	Status da execução, indica se um novo comando pode ser enviado, os valores são os seguintes: IEC104_COMMAND_EXEC_STATUS_DONE (0) IEC104_COMMAND_EXEC_STATUS_RUNNING (1)
eErrorStatus	ENUM (BYTE)	Indica se o último comando foi bem sucedido. IEC104_COMMAND_ERROR_STATUS_NO_ERROR (0) IEC104_COMMAND_ERROR_STATUS_TIMEOUT (1) IEC104_COMMAND_ERROR_STATUS_FORMAT_ERROR (2) IEC104_COMMAND_ERROR_STATUS_NOT_SUPPORTED (3) IEC104_COMMAND_ERROR_STATUS_INTERNAL_ERROR (4) IEC104_COMMAND_ERROR_STATUS_STATION_ERROR (5) IEC104_COMMAND_ERROR_STATUS_INVALID_ADDRESS (6) IEC104_COMMAND_ERROR_STATUS_WRONG_TIMEOUT (7) IEC104_COMMAND_ERROR_STATUS_UNINITIALIZED (8)

Tabela 154: Parâmetros de Saída Bloco IEC104_SynchronizeIED

4.5.14.6.5. 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**.

Nas tabelas seguintes pode-se visualizar os parâmetros de entrada e saída do bloco funcional.

Parâmetro	Tipo	Descrição
bRequest	BOOL	Quando TRUE, executa o comando.
dwVariableAddr	DWORD	Endereço de qualquer variável mapeada no servidor, usada apenas para identificar o servidor.
udiCommandTimeOut	UNSIGNED DOUBLE INT	Tempo para o Time Out deste comando.

Tabela 155: Parâmetros de Entrada Bloco IEC104_GeneralInterrogation

Parâmetro	Tipo	Descrição
eExecStatus	ENUM (BYTE)	Status da execução, indica se um novo comando pode ser enviado, os valores são os seguintes: IEC104_COMMAND_EXEC_STATUS_DONE (0) IEC104_COMMAND_EXEC_STATUS_RUNNING (1)
eErrorStatus	ENUM (BYTE)	Indica se o último comando foi bem sucedido. IEC104_COMMAND_ERROR_STATUS_NO_ERROR (0) IEC104_COMMAND_ERROR_STATUS_TIMEOUT (1) IEC104_COMMAND_ERROR_STATUS_FORMAT_ERROR (2) IEC104_COMMAND_ERROR_STATUS_NOT_SUPPORTED (3) IEC104_COMMAND_ERROR_STATUS_INTERNAL_ERROR (4) IEC104_COMMAND_ERROR_STATUS_STATION_ERROR (5) IEC104_COMMAND_ERROR_STATUS_INVALID_ADDRESS (6) IEC104_COMMAND_ERROR_STATUS_WRONG_TIMEOUT (7) IEC104_COMMAND_ERROR_STATUS_UNINITIALIZED (8)

Tabela 156: Parâmetros de Saída Bloco IEC104_GeneralInterrogation

4.5.14.6.6. 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:

- Ler
- Freeze
- Reset
- Reset e Freeze

Nas tabelas seguintes pode-se visualizar os parâmetros de entrada e saída do bloco funcional.

Parâmetro	Tipo	Descrição
bRequest	BOOL	Quando TRUE, executa o comando.
dwVariableAddr	DWORD	Endereço de qualquer variável mapeada no servidor, usado apenas para identificar o servidor.
eCommand	ENUM (BYTE)	Comando que será enviado: IEC104_COMMAND_READ (0) IEC104_COMMAND_FREEZE (1) IEC104_COMMAND_RESET (2) IEC104_COMMAND_FREEZE_RESET (3)
udiCommandTimeOut	UDINT	Tempo para o Time Out deste comando.

Tabela 157: Parâmetros de Entrada Bloco IEC104_CounterInterrogation

Parâmetro	Tipo	Descrição
eExecStatus	ENUM (BYTE)	Status da execução, indica se um novo comando pode ser enviado, os valores são os seguintes: IEC104_COMMAND_EXEC_STATUS_DONE (0) IEC104_COMMAND_EXEC_STATUS_RUNNING (1)
eErrorStatus	ENUM (BYTE)	Indica se o último comando foi bem sucedido. IEC104_COMMAND_ERROR_STATUS_NO_ERROR (0) IEC104_COMMAND_ERROR_STATUS_TIMEOUT (1) IEC104_COMMAND_ERROR_STATUS_FORMAT_ERROR (2) IEC104_COMMAND_ERROR_STATUS_NOT_SUPPORTED (3) IEC104_COMMAND_ERROR_STATUS_INTERNAL_ERROR (4) IEC104_COMMAND_ERROR_STATUS_STATION_ERROR (5) IEC104_COMMAND_ERROR_STATUS_INVALID_ADDRESS (6) IEC104_COMMAND_ERROR_STATUS_WRONG_TIMEOUT (7) IEC104_COMMAND_ERROR_STATUS_UNINITIALIZED (8)

Tabela 158: Parâmetros de Saída Bloco IEC104_CounterInterrogation

4.5.14.6.7. Exemplo de Utilização de um Bloco Funcional para Comando IEC 60870-5-104

Abaixo um exemplo de aplicação utilizando a linguagem ST, na qual o usuário envia o comando *Select Before Execute* para um IED via protocolo IEC 60870-5-104.

```

PROGRAM UserPrg
VAR
    iIec104ClientAO:      INT;
    iIec104ClientAOCmd:  LibIEC104.IEC104_AnalogCommandInt;
END_VAR

iIec104ClientAOCmd.dwVariableAddr:= ADR(iIec104ClientAO);
iIec104ClientAOCmd();
IF iIec104ClientAOCmd.eExecStatus = LibIEC104.IEC104_COMMAND_EXEC_STATUS_DONE
AND
iIec104ClientAOCmd.bRequest = TRUE THEN
    IF iIec104ClientAOCmd.eCommand = LibIEC104.IEC104_COMMAND_TYPE_SELECT AND
    iIec104ClientAOCmd.eErrorStatus = LibIEC104.
IEC104_COMMAND_ERROR_STATUS_NO_ERROR THEN
        iIec104ClientAOCmd.eCommand:= LibIEC104.IEC104_COMMAND_TYPE_EXECUTE;
    ELSE
        iIec104ClientAOCmd.bRequest:= FALSE;
    END_IF
END_IF
END_IF

```

4. CONFIGURAÇÃO

4.5.14.6.8. Códigos de Erro para Comandos IEC 60870-5-104

Os comandos IEC 60870-5-104 retornam status de erro (*eErrorStatus*) ao serem executados. A tabela abaixo mostra a lista e a descrição dos possíveis valores para o status de erro.

Código	Status	Descrição
0	IEC104_COMMAND_ERROR_STATUS_NO_ERROR	Comando aceito, iniciado ou na fila.
1	IEC104_COMMAND_ERROR_STATUS_TIMEOUT	Comando não aceito porque a mensagem foi recebida após o time out do comando.
2	IEC104_COMMAND_ERROR_STATUS_FORMAT_ERROR	Comando não aceito porque há erros de formatação do pedido.
3	IEC104_COMMAND_ERROR_NOT_SUPPORTED	Comando não aceito, porque a operação não é suportada por este ponto.
4	IEC104_COMMAND_ERROR_INTERNAL_ERROR	Comando não aceito devido a falha interna.
5	IEC104_COMMAND_ERROR_STATION_ERROR	Comando não aceito por falha desconhecida na Controlled Station.
6	IEC104_COMMAND_ERROR_INVALID_ADDRESS	Comando não aceito porque o endereço da variável é inválido.
7	IEC104_COMMAND_ERROR_WRONG_TIMEOUT	Comando não aceito porque o valor de time out não é válido.
8	IEC104_COMMAND_ERROR_UNINITIALIZED	Comando não aceito porque o bloco funcional não foi iniciado corretamente.

Tabela 159: Status de Erro para Comandos IEC 60870-5-104

4.5.15. 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 time-outs 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.

As descrições de cada configuração estão relacionadas a seguir, nesta seção.

4.5.15.1. Parâmetros Gerais

Para as configurações dos Parâmetros Gerais do IEC 60870-5-104 Servidor conforme a abaixo, seguem os parâmetros apresentados na tabela a seguir:

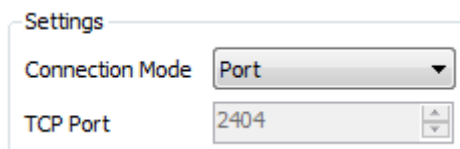


Figura 104: Tela de Parâmetros Gerais do IEC 60870-5-104 Servidor

Configuração	Descrição	Padrão	Opções
Connection Mode	Configura o modo de conexão com os módulos Cliente Conectados.	Port	Port IP
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".	2404	1 a 65535

Tabela 160: Configuração dos Parâmetros Gerais do IEC 60870-5-104 Servidor

4.5.15.2. Mapeamentos de Dados

Para a configuração das relações de dados do IEC 60870-5-104 Servidor, visualizadas na figura abaixo, seguem os parâmetros descritos na tabela a seguir.

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	*										
ApplicationLayer											

Figura 105: Tela de Mapeamento dos Dados do IEC 60870-5-104 Servidor

Configuração	Descrição	Padrão	Opções
Value Variable	Nome da variável simbólica.	-	Nome de uma variável declarada em um programa ou GVL.
Object Type	Configuração do tipo do objeto IEC 60870-5-104.	-	Single Point Information Double Point Information Step Position Information Measured Value (Normalized) Measured Value (Scaled) Measured Value (Short Floating Point) Integrated Totals Bitstring Information (32 Bits) Single Command Double Command Regulating Step Command Setting Point Command (Normalized) Setting Point Command (Scaled) Setting Point Command (Short Floating Point) Bitstring Command (32 Bits)
Object Address	Índice do primeiro ponto do mapeamento IEC 60870-5-104.	-	1 a 65535
Size	Especifica a quantidade máxima de dados que um mapeamento IEC 60870-5-104 poderá acessar.	-	- (calculado automaticamente)
Range	Faixa de endereços do dado configurado.	-	- (calculado automaticamente)
Counter Variable	Nome da variável simbólica que vai conter o dado da variável contadora.	-	Nome de uma variável declarada em um programa, GVL ou módulo contador
Dead Band Variable	Nome da variável simbólica que vai conter o dado da banda morta.	-	Nome de uma variável declarada em um programa ou GVL
Dead Band Type	Define o tipo de Banda Morta a ser utilizada no mapeamento.	Desabilitada	Absoluta Desabilitada Integrada
Select Required	Define se para a execução de um comando é requerido ou não um select prévio.	False	True False
Short Pulse	Define o tempo do pulso curto de um comando digital IEC 60870-5-104, em milisegundos.	1000	1 a 86400000
Long Pulse	Define o tempo do pulso longo de um comando digital IEC 60870-5-104, em milisegundos.	2000	1 a 86400000

Tabela 161: Configuração dos Mapeamentos do IEC 60870-5-104 Servidor

Notas:

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.

Counter Variable: Este campo se aplica somente no mapeamento de objetos do tipo Integrated Totals, sendo esta a variável contadora, a ser manipulada no processo. Ela deve ter o mesmo tipo e tamanho da variável mapeada na coluna Value Variable, cujo valor será lido pelo cliente ou reportado ao cliente em caso de eventos.

ATENÇÃO

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: 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.

Counter Variable: Os tipos de configuração da Banda Morta disponíveis são os seguintes:

Tipo de Função	Configurações	Descrição
Tipo da Banda Morta	Desabilitada	Nesta opção, qualquer modificação de valor num ponto do grupo, por menor que seja, gera evento para este ponto.
	Absoluta	Nesta opção, se o módulo da variação do valor de um ponto do grupo for maior do que o valor configurado pela variável no campo “Banda Morta”, um evento para este ponto é gerado.
	Integrada	Nesta opção, se o módulo da integral da variação do valor de um ponto do grupo for maior do que o valor configurado pela variável no campo “Banda Morta”, um evento para este ponto é gerado. O intervalo de integração é de um segundo.

Tabela 162: Tipos de Banda Morta nos Parâmetros do IEC 60870-5-104 Servidor

Nota:

Short Pulse e Long Pulse: 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.

ATENÇÃO

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.

4.5.15.3. Camada de Enlace

Para a configuração dos parâmetros da camada de enlace do IEC 60870-5-104 Servidor, visualizadas na figura abaixo, seguem os parâmetros descritos na tabela a seguir.

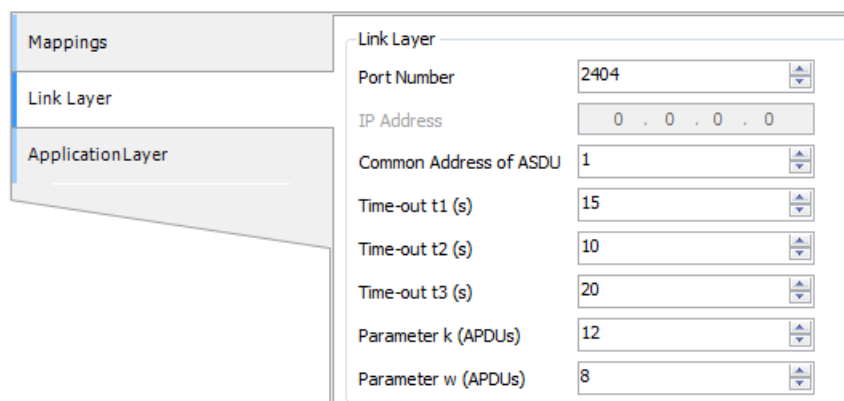


Figura 106: Tela de Configuração da Camada de Enlace IEC 60870-5-104 Servidor

Configuração	Descrição	Padrão	Opções
Port Number	Endereço da porta listen para conexão do cliente. Utilizado quando a conexão do cliente não é por IP.	2404	1 a 65535
IP Address (Master)	IP do cliente conectado, utilizado quando a conexão do cliente é por IP.	0.0.0.0	1.0.0.1 a 223.255.255.254
Common Address of ASDU	Endereço IEC 60870-5-104 do servidor.	1	1 a 65535
Time-out t1(s)	Período de tempo (em segundos) que o dispositivo aguarda o recebimento de uma mensagem de confirmação após o envio de uma mensagem APDU do tipo I ou U (dados) antes de fechar a conexão.	15	1 a 180
Time-out t2(s)	Período de tempo (em segundos) que o dispositivo aguarda para enviar uma mensagem supervisória (S-Frame) confirmando o recebimento de frames de dados.	10	1 a 180
Time-out t3(s)	Período de tempo (em segundos) em que será enviada uma mensagem para teste do link caso não haja nenhuma transmissão por ambos os lados.	20	1 a 180
Parameter k (APDUs)	Número máximo de mensagens de dados (I-Frame) transmitidas e não confirmadas.	12	1 a 12
Parameter w (APDUs)	Número máximo de mensagens de dados (I-Frame) recebidas e não confirmadas.	8	1 a 8

Tabela 163: Configuração da Camada de Enlace do IEC 60870-5-104 Servidor

Nota:

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

4.5.15.4. Camada de Aplicação

Para a configuração dos parâmetros da camada de aplicação do IEC 60870-5-104 Servidor, visualizadas na figura abaixo, seguem os parâmetros descritos na tabela a seguir.

Figura 107: Tela de Configuração da Camada de Aplicação do IEC 60870-5-104 Servidor

Configuração	Descrição	Padrão	Opções
Enable Time Synchronization	Opção para Habilitar/Desabilitar a solicitação de sincronismo de tempo.	Disabled	Enabled. Disabled.
Time Synchronization Command Received in Local Time	Opção para Habilitar/Desabilitar o tratamento do comando de sincronização em horário local.	Enabled	Enabled. Disabled.
Use Local Time instead of UTC Time	Opção para Hablitar/Desabilitar a estampa de tempo em horário local para eventos.	Disabled	Enabled. Disabled.
Maximum Time Between Select and Operate (s)	Período de tempo no qual o comando de seleção permanecerá ativo (inicia a contagem a partir da confirmação da recepção do comando de seleção) aguardando o comando de Operate.	5	1 a 180
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 time-out.	2	1 a 180
Transmission Mode of Analog Inputs Events	Modo de transmissão dos eventos de entradas analógicas.	All Events (SOE)	All Events (SOE). Most Recent Events.
Transmission Mode	Modo de transmissão dos dados dos contadores.	Freeze by counter-interrogation command, transmit spontaneously.	Freeze by counter-interrogation command, transmit spontaneously. Freeze and transmit by counter-interrogation command.

Tabela 164: Configuração da Camada de Aplicação IEC 60870-5-104 Servidor

Notas:

Enable Time Synchronization: Uma vez habilitado, permite o IEC 60870-5-104 Servidor ajustar o relógio da UCP quando recebido um comando de sincronização.

Time Synchronization Command Received in Local Time: 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: Uma vez habilitado, a estampa de tempo dos eventos gerados pelo IEC 60870-5-104 Servidor será enviada conforme o horário local da UCP.

ATENÇÃO

Quando a opção de sincronismo de tempo for marcada em mais de um servidor (parâmetro “Enable Time Synchronization”) 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.

Maximum Time to Complete a Command Execution: 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 Input Events: Os modos de transmissão dos eventos de entradas analógicas disponíveis são os seguintes:

Tipo de Função	Configurações	Descrição
Transmission Mode of Analog Inputs Events	All Events(SOE)	São enviados todos os eventos analógicos gerados.
	Most Recent Events	É enviado somente o evento mais recente de cada ponto analógico.

Tabela 165: Modos de Transmissão dos Eventos de Entradas Analógicas do IEC 60870-5-104 Servidor

Tipo de Função	Configurações	Descrição
Transmission Mode	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.

Tabela 166: Modos de Transmissão dos Contadores Congelados do IEC 60870-5-104 Servidor

ATENÇÃO

A norma IEC 60870-5-104, seção “Transmission control using Start/Stop”, prevê a utilização de comandos STARTDT e STOPDT para controle do tráfego de dados entre o cliente e o servidor, utilizando conexões simples ou múltiplas conexões. Apesar 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).

4.5.15.5. 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* as quais estão descritas na tabela abaixo.

Variável de diagnóstico do tipo <i>T_DIAG_IEC104_SERVER_1</i> .	Tamanho	Descrição
Bits de comando, reiniciados automaticamente:		
tCommand. bStop	BIT	Desabilitar Driver.
tCommand. bStart	BIT	Habilitar o Driver.
tCommand. bDiag_01_Reserved	BIT	Reservado.
tCommand. bDiag_02_Reserved	BIT	Reservado.
tCommand. bDiag_03_Reserved	BIT	Reservado.
tCommand. bDiag_04_Reserved	BIT	Reservado.
tCommand. bDiag_05_Reserved	BIT	Reservado.
tCommand. bDiag_06_Reserved	BIT	Reservado.
Diagnósticos:		
tClient_X. bRunning	BOOL	IEC 60870-5-104 Servidor está sendo executado.
tClient_X. eConnectionStatus. CLOSED	ENUM (BYTE)	Canal de comunicação fechado. Servidor não irá aceitar solicitação de conexão. Valor do ENUM (0).
tClient_X. eConnectionStatus. CONNECTED	ENUM (BYTE)	Servidor está escutando a porta configurada, e não existem clientes conectados. Valor do ENUM (1).
tClient_X. eConnectionStatus. CONNECTED	ENUM (BYTE)	Cliente conectado. Valor do ENUM (2).
tClient_X. tQueueDiags. bOverflow	BOOL	Fila do cliente está com overflow.
tClient_X. tQueueDiags. wSize	WORD	Tamanho da fila configurada.
tClient_X. tQueueDiags. wUsage	WORD	Número de eventos na fila.
tClient_X. tQueueDiags. dwReserved_0	DWORD	Reservado.

Variável de diagnóstico do tipo T_DIAG_IEC104_SERVER_1.	Tamanho	Descrição
tClient_X. tQueueDiags. dwReserved_1	DWORD	Reservado.
tClient_X. tStats. wRXFrames	WORD	Número de quadros (frames) recebidos.
tClient_X. tStats. wTXFrames	WORD	Número de quadros (frames) enviados.
tClient_X. tStats. wCommErrors	WORD	Contador de erros de comunicação incluindo erros na Camada Física, Camada de Enlace e Camada de Transporte.
tClient_X. tStats. dwReserved_0	DWORD	Reservado.
tClient_X. tStats. dwReserved_1	DWORD	Reservado.

Tabela 167: Diagnósticos do IEC 60870-5-104 Servidor

4.5.15.6. Associação dos Objetos com o Protocolo DNP3

A tabela apresenta associações permitidas entre os protocolos IEC 60870-5-104 e DNP3.

Tipo do Objeto IEC 60870-5-104	Grupo DNP3
Single Point Information (M_SP_NA)	g01 - Entrada digital
Double Point Information (M_DP_NA)	g03 - Entrada digital dupla
Step Position Information (M_ST_NA)	-
Measured Value, normalized value (M_ME_NA)	g30 - Entrada analógica
Measured Value, scaled value (M_ME_NB)	g30 - Entrada analógica
Measured Value, short floating point value (M_ME_NC)	g30 - Entrada analógica
Single Command (C_SC_NA)	g10 - Saída digital
Double Command (C_DC_NA)	g10 - Saída digital
Regulating Step Command (C_RC_NA)	g10 - Saída digital
Setting Point Command, normalized Value (C_SE_NA)	g40 - Saída analógica
Setting Point Command, scaled Value (C_SE_NB)	g40 - Saída analógica
Setting Point Command, short floating point Value (C_SE_NC)	g40 - Saída analógica

Tabela 168: Associação dos Tipos de Objetos IEC 60870-5-104 com Seus Correspondentes no DNP3

ATENÇÃO

Integrated Totals (M_IT_NA): não é permitido associar uma variável utilizada como contadora congelada, simultaneamente no protocolos IEC 60870-5-104 e DNP3.
Step Position Information (M_ST_NA): não há grupo equivalente para o protocolo DNP3.

4.5.15.7. Comandos para Pontos de Saída

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

- Pontos Internos
- Módulo de saída (por ex. HX2320, HX2300)
- Driver DNP3 Cliente
- Driver IEC 60870-5-104 Cliente
- Intercepção (bloco de função `CommandReceiver`)

4.5.15.7.1. 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.

4.5.15.7.2. Módulo de Saída

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 intercepção e tratamento destes comandos pela aplicação pode evitar o retorno de mensagens negativas (falha).

ATENÇÃO

Comandos destinados à variáveis mapeadas em um cartão de saída da Série Nexto (por exemplo um módulo NX2020) são tratados como comandos para pontos internos.

4.5.15.7.3. 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 time-out), 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 esteja ocupado executando um comando, será retornada uma mensagem negativa de confirmação.

4.5.15.7.4. 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.

4.5.15.7.5. Intercepção de Comandos

A intercepção de comandos recebidos pelo driver IEC 60870-5-104 Servidor é realizada através do bloco de função `CommandReceiver` conforme descrito na seção [Intercepção de Comandos Oriundos do Centro de Controle](#). Este recurso permite a intercepção de comandos de seleção e execução.

4.5.15.7.6. Qualificador de 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) Long pulse duration (pulso longo)	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.
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 169: Qualificador dos Comandos do IEC 60870-5-104 Servidor

4.5.15.7.7. Interceptação do Comando

Para maiores informações sobre a interceptação de comandos de clientes IEC 60870-5-104, consultar a seção [Interceptação de Comandos Oriundos do Centro de Controle](#), implementada através do bloco de função *CommandReceiver*.

4.5.16. 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 também possibilitando o envio e recepção de mensagens rápidas através do protocolo GOOSE.

A tabela a seguir descreve os limites máximos de configuração relacionados ao protocolo IEC 61850 Servidor.

	IEC 61850 Servidor
Instâncias por Interface Ethernet (NETs)	1
Instâncias por UCP	1
Número máximo de clientes	10
Número máximo de Logical Nodes	255
Número máximo de Datasets	100
Número máximo de Datasets Dinâmicos	10
Número máximo de membros em um Dataset	300
Número máximo de GOOSE Control Blocks	100

IEC 61850 Servidor	
Número máximo de GOOSE Subscriptions	100
Número máximo de Report Control Blocks	100
Número máximo de mapeamentos	1.000

Tabela 170: Limites do Protocolo IEC 61850 Servidor

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 .. 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 (bufferizados e não-bufferizados), 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.

As descrições de cada configuração estão relacionadas a seguir, nesta seção.

4.5.16.1. Implementação do Modelo de Dados IEC 61850

Uma das principais diferenças da norma IEC 61850 quando comparada aos outros protocolos conhecidos (DNP3, MOD-BUS, etc...) está no modelo de dados utilizado para comunicação. Na IEC 61850, os dados são definidos sob a forma de grandes estruturas de dados simbólicos 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 abaixo 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:

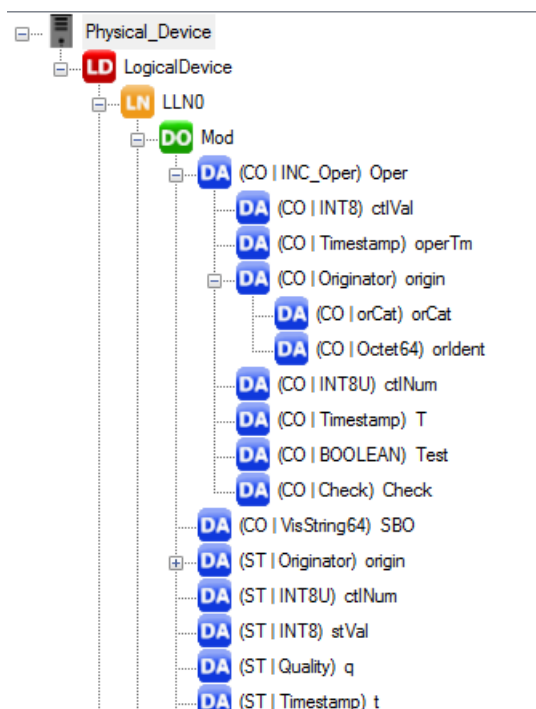


Figura 108: Modelo de Dados Estruturado

4. CONFIGURAÇÃO

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 POUs 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 POUs” localizada na árvore do projeto.

A figura abaixo mostra um exemplo das POUs geradas a partir de uma configuração IEC 61850:

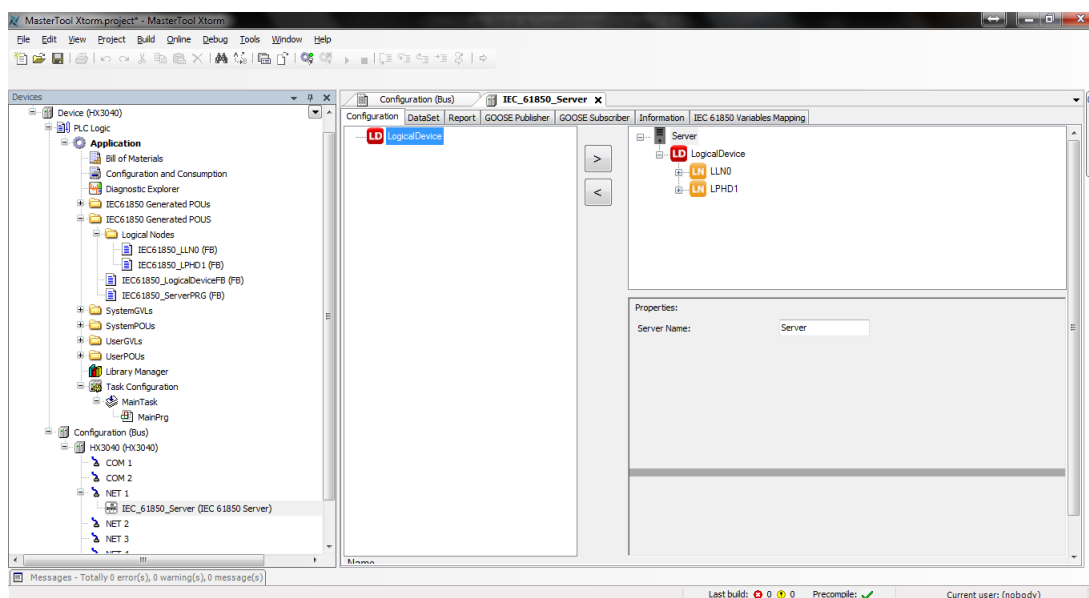


Figura 109: Implementação do Modelo de Dados

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.

Conforme citado anteriormente, cada Atributo torna-se uma variável simbólica no ambiente de programação IEC 61131-3. A tabela a seguir mostra a correspondência entre os tipos de dados básicos da IEC 61850 com os tipos de dados elementares da IEC 61131-3:

Tipo básico IEC 61850	Tipo elementar IEC 61131-3
BOOLEAN	BOOL
INT8	SINT
INT16	INT
INT32	DINT
INT8U	USINT
INT16U	UINT
INT32U	UDINT
FLOAT32	REAL
FLOAT64	LREAL
ENUMERATED	INT
CODED ENUM	INT
OCTET STRING _{nn} VISIBLE STRING _{nn} UNICODE STRING _{nn}	STRING(_{nn})

Tabela 171: Correspondência dos Tipos de Dados IEC 61850 vs IEC 61131-3

4. CONFIGURAÇÃO

O formato do *Timestamp* segue o padrão do IEC61850, no qual utiliza uma estrutura do tipo DATE_AND_TIME que é padrão da IEC61131-3 e mais uma variável para a fração de segundos *dwFracsec*.

A variável *dwFracsec* é uma parte fracionária em um número de 24 bits. O bit menos significativo representa $1/2^{24} = 59,60$ ns. Para chegar no valor fracionário em decimal, o valor presente em *dwFracsec* deve ser dividido por 256 (para reduzir de 32 para 24 bits) e multiplicar por 59,60 ns.

Já os tipos de dados enumerados possuem um conjunto de elementos cujos valores recebem um nome.

A tabela a seguir descreve os valores (em base decimal) que representam na memória da UTR cada um dos nomes dos elementos:

CDC (Common Data Class)	Atributo	Identificador	Valor (decimal)
-	origin.OrCat	not-supported	0
		bay-control	1
		station-control	2
		remote-control	3
		automatic-bay	4
		automaticstation	5
		automatic-remote	6
		maintenance	7
ACD	dirGeneral	unknown	0
	dirPhsA	forward	1
	dirPhsB	backward	2
	dirPhsC	both	3
	dirNeut		
BSC	ctlVal	stop	0
		lower	16384
		higher	-32768
		reseved	-16384
CMV	angRef	V	0
		A	1
		Other	2
CURVE	setCharact	ANSI Extremely Inverse	1
		ANSI Very Inverse	2
		ANSI Normal Inverse	3
		ANSI Moderate Inverse	4
		ANSI Definite Time	5
		Long-Time Extremely Inverse	6
		Long-Time Very Inverse	7
		Long-Time Inverse	8
		IEC Normal Inverse	9
		IEC Very Inverse	10
		IEC Inverse	11
		IEC Extremely Inverse	12
		IEC Short-Time Inverse	13
		IEC Long-Time Inverse	14
		IEC Definite Time	15
		Polynom 1 to 16	17 to 32
Multiline 1 to 16	33 to 48		
DPC DPS	stVal	intermediate	0
		off	16384
		on	-32768

4. CONFIGURAÇÃO

CDC (Common Data Class)	Atributo	Identificador	Valor (decimal)
		BAD	-16384
SEQ MV CMV	Sev range	normal	0
		high	1
		low	2
		high-high	3
		low-low	4
SEQ	phsRef	A	0
		B	1
		C	2
SEQ	T	pos-neg-zero	0
		dir-quad-zero	1
WYE	angRef	Va	0
		Vb	1
		Vc	2
		Aa	3
		Ab	4
		Ac	5
		Vab	6
		Vbc	7
		Vca	8
		Vother	9
		Aother	10

Tabela 172: Elementos Representados na Memória da UTR

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:

Nome da GVL (Tabela)	Descrição
gaVisSTRING32	VISIBLE STRING32
gaVisSTRING64	VISIBLE STRING64
gaVisSTRING129	VISIBLE STRING129
gaVisSTRING255	VISIBLE STRING255
gaUCSTRING255	UNICODE STRING255
gaOCTED64	OCTET STRING64
gaOCTED255	OCTET STRING255

Tabela 173: Tabelas para Acesso aos Atributos do Tipo STRING

Por exemplo, vejamos a configuração na figura abaixo:

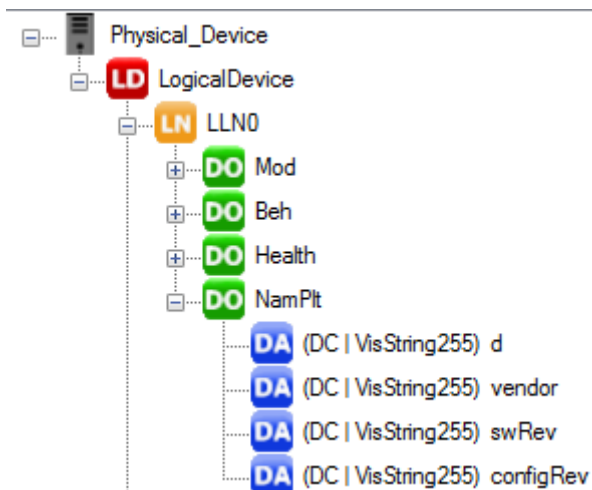


Figura 110: Exemplo de Configuração com Atributos do tipo STRING

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';
```

4.5.16.2. 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:

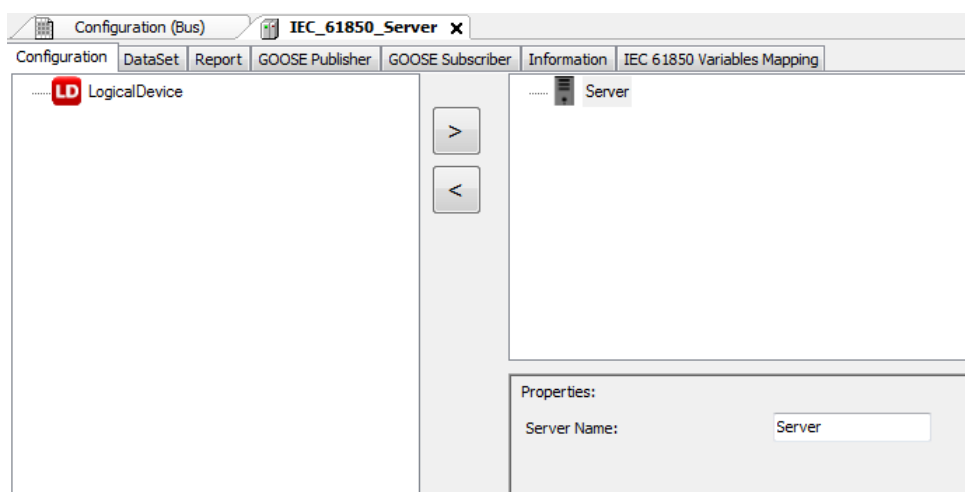


Figura 111: Configurando o Nome do Physical Device

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.

4. 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 esta UTR será utilizada, pois este será o nome que irá identificar este dispositivo na configuração da comunicação com dispositivos MMS Cliente e também 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 “>”:

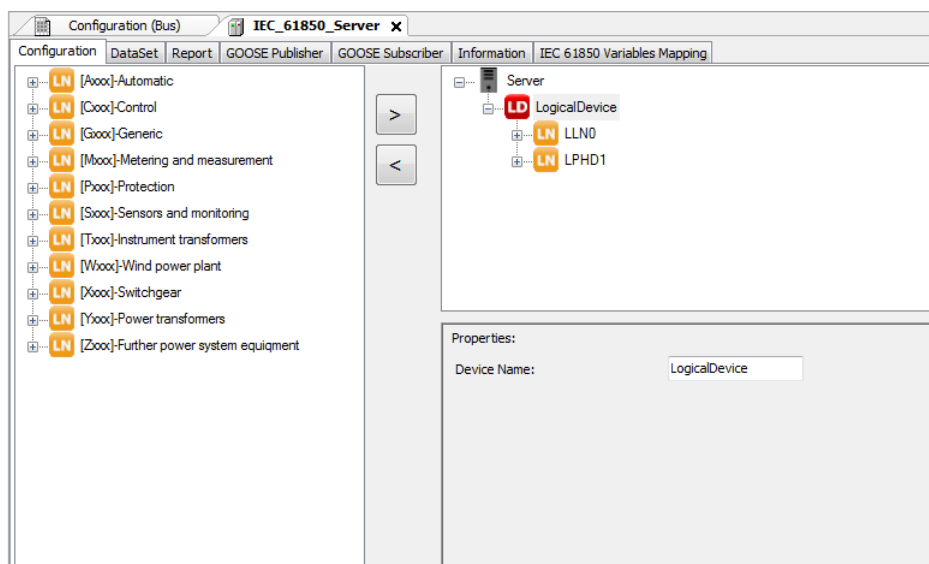


Figura 112: Adicionando e Configurando o Nome do Logical Device

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:

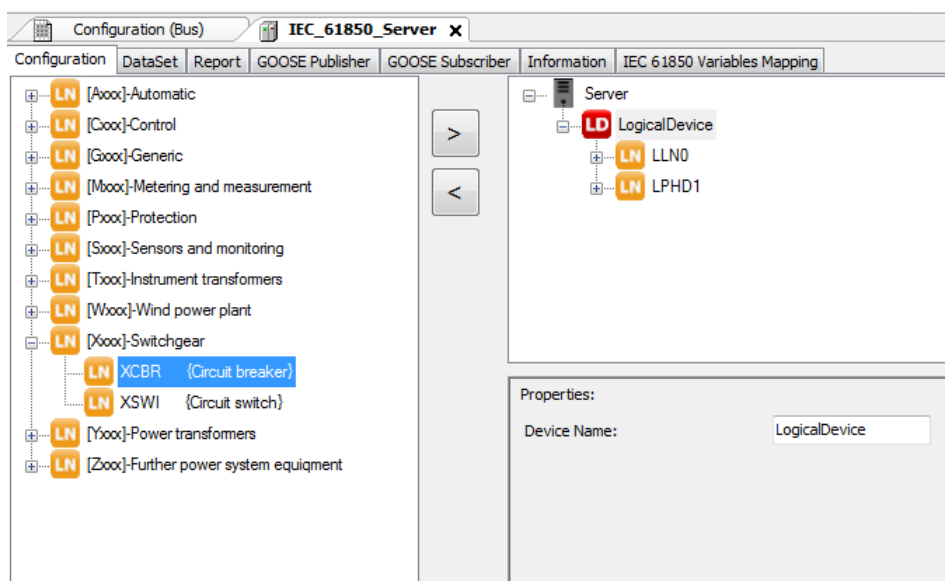


Figura 113: Adicionando um Logical Node

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 disso, 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.

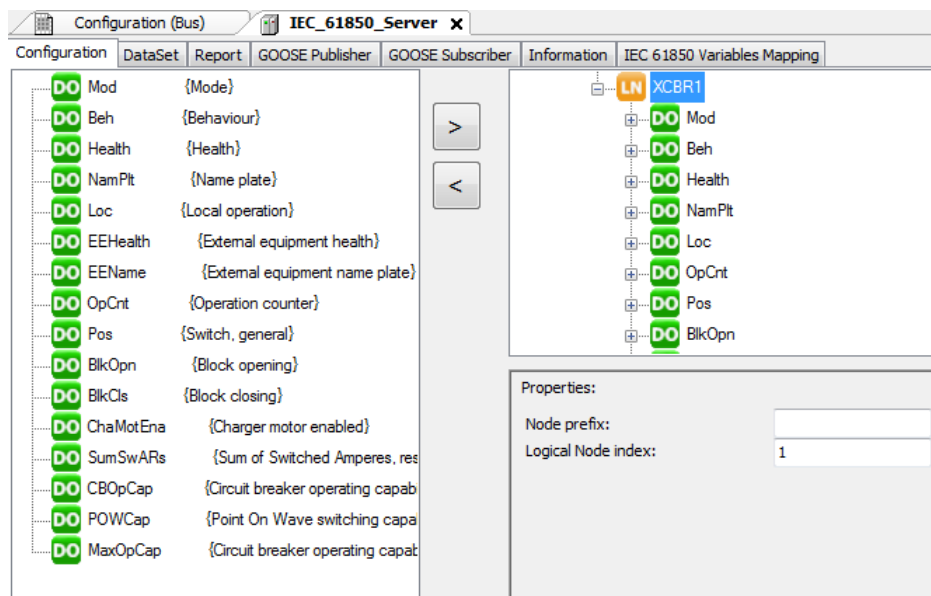


Figura 114: Configurando um Logical Node

A tabela abaixo mostra todos os campos de configuração presentes nesta tela, informando seus valores padrão e as possibilidades de configuração:

Configuração	Descrição	Padrão	Opções
Server Name	Define o nome do Physical Device.	Server	Conjunto de até 32 caracteres (letras, números e underline).
Device Name	Define o nome do Logical Device.	LogicalDevice	Conjunto de até 32 caracteres (letras, números e underline).
Node Prefix	Define o prefixo do Logical Node.	-	Conjunto de caracteres (apenas letras) cuja quantidade somada ao número de caracteres do índice não pode ultrapassar 07 caracteres.
Logical Node Index	Define o índice do Logical Node.	-	Conjunto de caracteres (apenas números) cuja quantidade somada ao número de caracteres do prefixo não pode ultrapassar 07 caracteres.

Tabela 174: Parâmetros da Configuração da Base de Dados

Notas:

A soma do tamanho dos campos **Node Prefix** e **Logical Node Index** não deve ultrapassar 12 caracteres.

O limite do caminho de um atributo, definido pela soma do tamanho dos campos **Device Name**, **Node Prefix**, **Logical Node Index**, bem como todos os separadores definidos na norma e do nome do atributo propriamente não deve ultrapassar 64 caracteres.

4.5.16.3. 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:

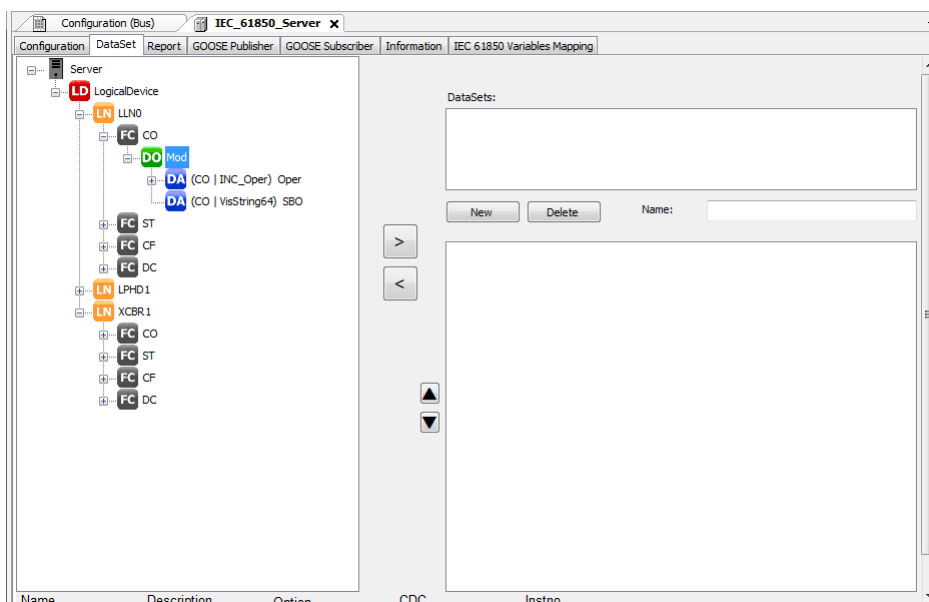


Figura 115: Configurando os Datasets

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.

4.5.16.4. 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:

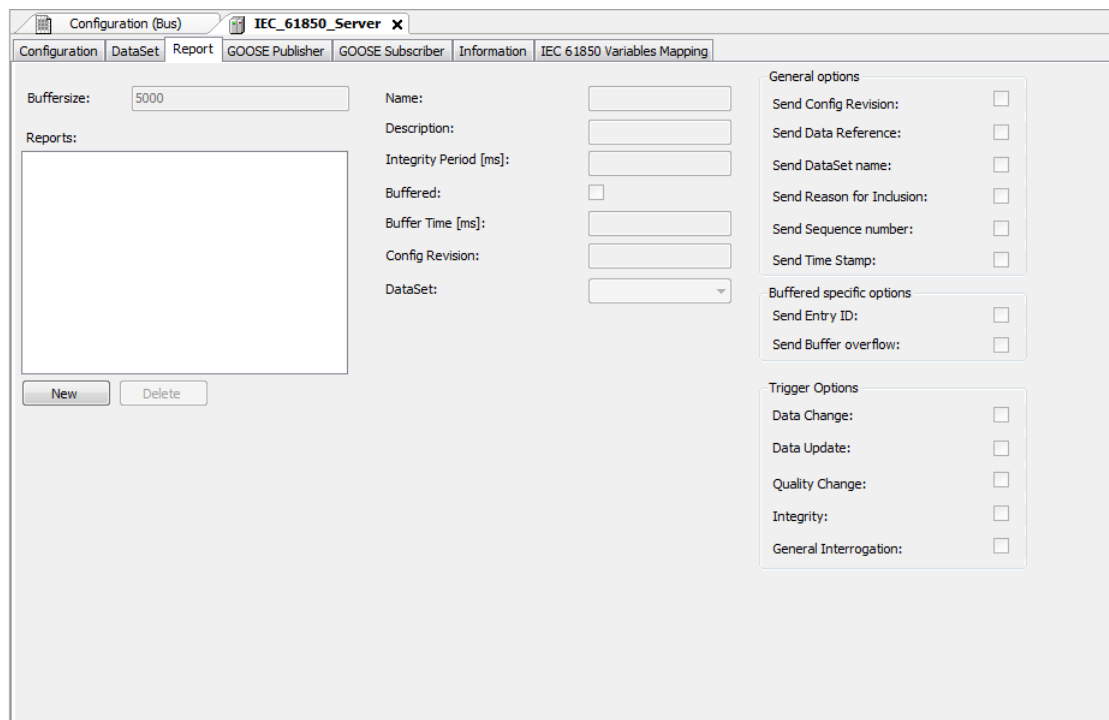


Figura 116: Configurando os Reports

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 20Kbytes 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).

A tabela abaixo mostra todos os campos de configuração presentes nesta tela, informando seus valores padrão e as possibilidades de configuração:

Configuração	Descrição	Padrão	Opções
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.	RCB_n	Conjunto de até 32 caracteres (letras, números e underline).
Description	Descrição do Report Control Block, para auxiliar na documentação do projeto.	-	Conjunto de até 80 caracteres (letras, números e underline).

4. CONFIGURAÇÃO

Configuração	Descrição	Padrão	Opções
Integrity Period (ms)	Período no qual o Report será disparado automaticamente caso nenhuma das outras condições de disparo tenha ocorrido.	0	0 – Desabilitado (nunca envia) 1 a 65535
Buffered	Seleciona o modo Bufferizado, no qual os Reports são armazenados internamente na UTR para envio posterior caso o Cliente MMS não esteja conectado.	Desabilitado	Habilitado/Desabilitado
Buffer Time (ms)	Tempo que será aguardado após a ocorrência de um evento para que o Report seja transmitido.	100	0 a 65535
Config. Revision	Número que será utilizado para identificar a revisão da configuração do Report Control Block.	1	0 a 65535
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.	Habilitado	Habilitado/Desabilitado
Send Data Reference	Caso selecionado, envia a referência de dados juntamente com o Report.	Habilitado	Habilitado/Desabilitado
Send Dataset Name	Caso selecionado, envia o nome do Dataset juntamente com o Report.	Habilitado	Habilitado/Desabilitado
Send Reason for Inclusion	Caso selecionado, envia o motivo da inclusão juntamente com o Report.	Habilitado	Habilitado/Desabilitado
Send Sequence Number	Caso selecionado, envia o número de sequência juntamente com o Report.	Habilitado	Habilitado/Desabilitado
Send Time Stamp	Caso selecionado, envia a estampa de tempo juntamente com o Report.	Habilitado	Habilitado/Desabilitado
Send Entry ID	Caso selecionado, envia o ID da Entrada juntamente com o Report.	Desabilitado	Habilitado/Desabilitado
Send Buffer Overflow	Caso selecionado, envia a informação de Buffer Overflow juntamente com o Report.	Desabilitado	Habilitado/Desabilitado
Data Change	Caso selecionado, habilita a condição de disparo do Report em função da alteração de valor.	Habilitado	Habilitado/Desabilitado

Configuração	Descrição	Padrão	Opções
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.	Habilitado	Habilitado/Desabilitado
Integrity	Caso selecionado, habilita a condição de disparo do Report caso solicitado o Polling de Integridade pelo Cliente MMS.	Habilitado	Habilitado/Desabilitado
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.	-	-

Tabela 175: Parâmetros de Configuração dos Reports

4.5.16.5. 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:

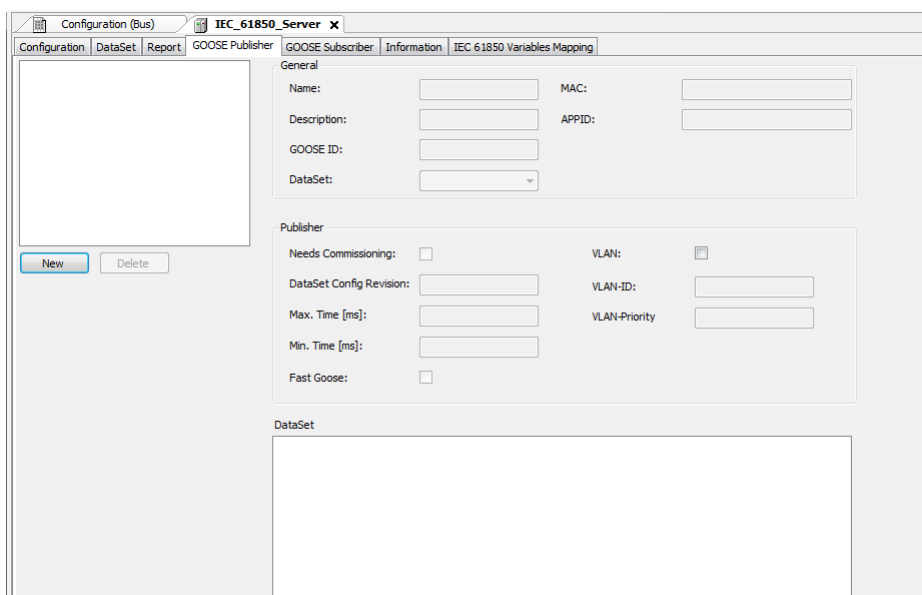


Figura 117: Configurando a Comunicação GOOSE Publisher

4. CONFIGURAÇÃO

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 ProfTask, 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.

ATENÇÃO

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 realizada, o Report não funcionará corretamente.

A tabela abaixo mostra todos os campos de configuração presentes nesta tela, informando seus valores padrão e as possibilidades de configuração:

Configuração	Descrição	Padrão	Opções
Name	Nome que identificará este GOOSE Control Block para fins de configuração e de comunicação com outros dispositivos.	GCB_n	Conjunto de até 32 caracteres (letras, números e underline).
Description	Descrição do GOOSE Control Block, para auxiliar na documentação do projeto.	-	Conjunto de até 80 caracteres (letras, números e underline).
GOOSE ID	Também é conhecido como AppID, é uma string de identificação definida pelo usuário.	GoCBRef	Conjunto de até 65 caracteres (letras, números e underline).
Dataset	Dataset que será monitorado e transmitido por este GOOSE Control Block.	-	-
MAC	Endereço MAC de destino (multicast).	01-0C-CD-01-00-00	01-0C-CD-01-00-00 a 01-0C-CD-01-01-FF
APP ID	Número de identificação da aplicação. Este número pode ser utilizado para distinguir mensagens GOOSE enviadas por diferentes partes do sistema de automação.	0x0000	0x0000 a 0x3FFF
Needs Commissioning	Deve ser habilitada pelo usuário para indicar que a configuração deste GOOSE Control Block ainda não foi concluída.	Desabilitado	Habilitado/Desabilitado
Dataset Config Revision	Número que deve ser utilizado para indicar a revisão do Dataset associado a este GOOSE Control Block.	1	0 a 4294967295

Configuração	Descrição	Padrão	Opções
Max. Time (ms)	Intervalo de tempo máximo para a retransmissão de uma mensagem GOOSE.	4000	0 a 4294967295
Min. Time (ms)	Intervalo de tempo mínimo para a retransmissão de uma mensagem GOOSE.	500	0 a 4294967295
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.	Desabilitado	Habilitado/Desabilitado
VLAN	Caso selecionado, habilita a utilização de VLAN na mensagem GOOSE.	Desabilitado	Habilitado/Desabilitado
VLAN ID	Número de identificação da VLAN.	0x0000	0x0000 a 0x0FFF
VLAN Priority	Número prioridade da VLAN.	4	0 a 7

Tabela 176: Parâmetros de Configuração do GOOSE Publisher

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:

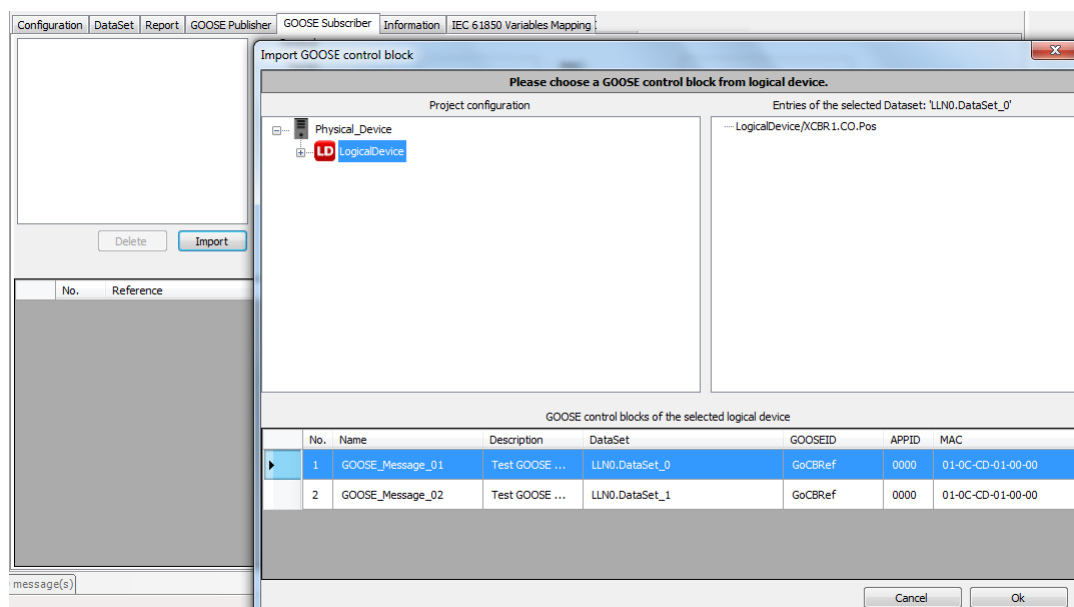


Figura 118: Importando um Arquivo SCL

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 recebidas. 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.

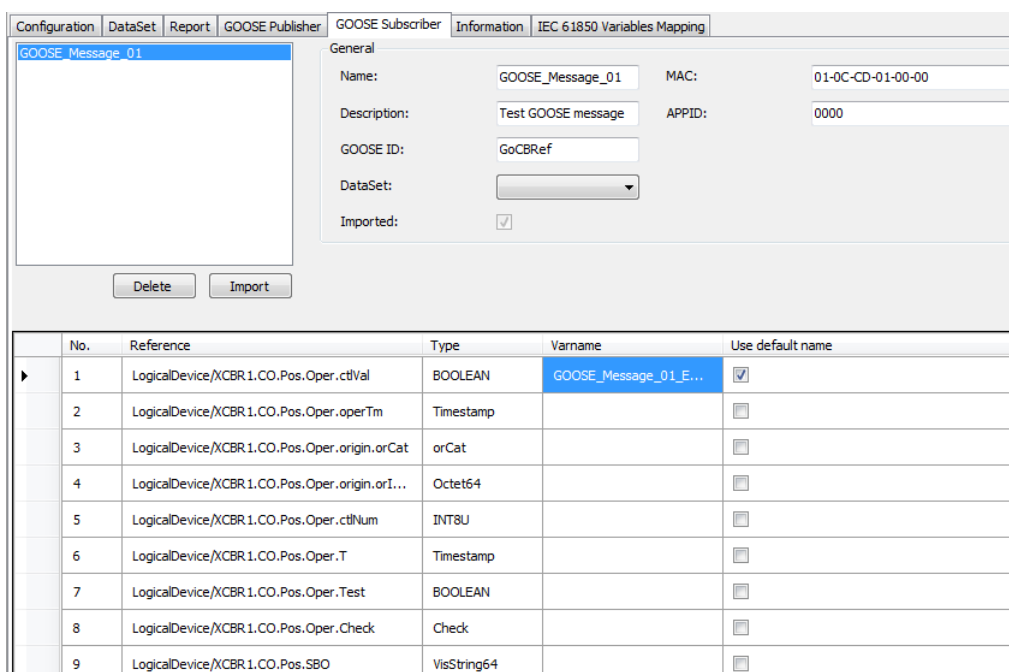


Figura 119: Configurando a Comunicação GOOSE Subscriber

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 Xform 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.

4.5.16.6. 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.

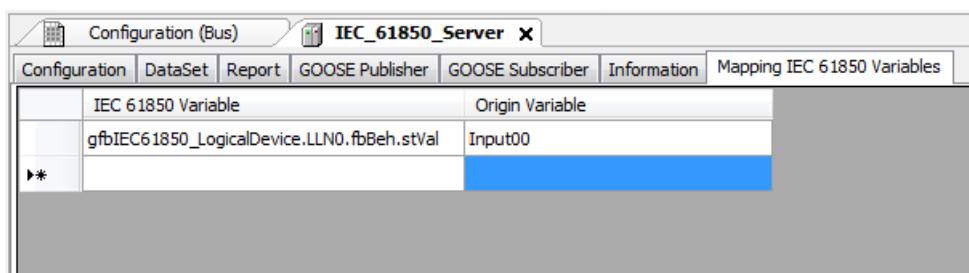


Figura 120: Mapeamento de Variáveis IEC 61850

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, sendo que o limite máximo está definido no início desta seção.

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. A tabela abaixo mostra as possibilidades de mapeamento entre uma *Variável Origem* e uma *Variável IEC 61850*:

Tipo da Variável Origem	Tipo da Variável IEC 61850	
	IEC 61131-3	IEC 61850
BOOL	BOOL	BOOLEAN
SINT	SINT, DBP	INT8
INT	INT, DINT, REAL	INT16, INT32, REAL
DINT	DINT, REAL	INT32, REAL
USINT	USINT	INT8U
UINT	DINT, UDINT, REAL	INT32, INT32U, REAL
UDINT	UDINT, REAL	INT32U, REAL
REAL	REAL	FLOAT32
DBP	SINT	INT8

Tabela 177: Tipos de Dados Possíveis no Mapeamento de Variáveis 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](#).

4.5.16.7. 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*. Os diagnósticos estão descritos na tabela abaixo.

Variável de diagnóstico do tipo <i>T_DIAG_IEC61850_SERVER.*</i>	Tamanho	Descrição
Diagnósticos:		
bDriverConfigured	BOOL	Indica que o driver foi configurado corretamente.
tGooseGenDiagnostic. uiGooseControlBlocks	UINT	Número de GOOSE Control Blocks (GCBs) configurados.
tGooseGenDiagnostic. uiGooseSubscriptions	UINT	Número de mensagens GOOSE configuradas para serem recebidas.
tGooseControlBlock. tGooseControlBlock[n]. bActive	BOOL	Indica que esta estrutura de diagnósticos está ativa.
tGooseControlBlock. tGooseControlBlock[n]. uiAPPID	UINT	Número de identificação do GOOSE Control Block.

Variável de diagnóstico do tipo T_DIAG_IEC61850_SERVER.*	Tamanho	Descrição
tGooseControlBlock. tGooseControlBlock[n]. udiDataChange	UDINT	Número de vezes que foi detectada alguma alteração nos elementos do Dataset.
tGooseControlBlock. tGooseControlBlock[n]. udiRetransmission	UDINT	Número de retransmissões da mensagem GOOSE (é zerado sempre que houver um Data Change).
tGooseControlBlock. tGooseControlBlock[n]. udiTotal	UDINT	Total de mensagens transmitidas (inclui a transmissão inicial e todas as retransmissões).
tGooseSubscription. tGooseSubscription[n]. bActive	BOOL	Indica que esta estrutura de diagnósticos está ativa.
tGooseSubscription. tGooseSubscription[n]. uiAPPID	UINT	Número de identificação do GOOSE Control Block correspondente à mensagem configurada para recepção.
tGooseSubscription. tGooseSubscription[n]. udiReceived	UDINT	Número de recepções da mensagem GOOSE configurada para recepção.
tGooseSubscription. tGooseSubscription[n]. udiDataChange	UDINT	Número de vezes que foi detectada alguma alteração nos elementos do Dataset ao receber a mensagem.
tGooseSubscription. tGooseSubscription[n]. udiTimeout	UDINT	Indica que não houve recepção da mensagem GOOSE dentro do tempo definido por Time Allowed to Live (TAL).
tGooseSubscription. tGooseSubscription[n]. tTimestamp	T_TIMESTAMP	Estampa de tempo do momento em que detectada alguma alteração nos elementos do Dataset ao receber a mensagem.
tReportControlBlock. tReportControlBlock[n]. sRCB_NAME	STRING(65)	Nome do Report Control Block.
tReportControlBlock. tReportControlBlock[n]. sIP_ADDR	STRING(15)	Endereço IP do IEC 61850 Cliente que habilitou e/ou reservou este Report.
tReportControlBlock. tReportControlBlock[n]. bEnabled	BOOL	Indica que este Report foi habilitado por um IEC 61850 Cliente.
tReportControlBlock. tReportControlBlock[n]. bResv	BOOL	Indica que este Report foi reservado por um IEC 61850 Cliente (aplicável somente para Buffered Reports).
tReportControlBlock. tReportControlBlock[n]. bBuffered	BOOL	Indica que este Report é do tipo Buffered.
tReportControlBlock. tReportControlBlock[n]. byOccupation	BYTE	Informa a ocupação do buffer interno (de 0 a 100, em %, aplicável somente para Buffered Reports).

Variável de diagnóstico do tipo T_DIAG_IEC61850_SERVER.*	Tamanho	Descrição
tConnectionStatus. tConnectionStatus[n]. bActive	BOOL	Indica que há conexão estabelecida com um IEC 61850 Cliente.
tConnectionStatus. tConnectionStatus[n]. sIP_ADDR	STRING(15)	Endereço IP do IEC 61850 Cliente que está conectado.
tConnectionStatus. tConnectionStatus[n]. udiFramesTx	UDINT	Informa a quantidade de frames transmitidos na comunicação com o IEC 61850 Cliente que está conectado.
tConnectionStatus. tConnectionStatus[n]. udiFramesRx	UDINT	Informa a quantidade de frames recebidos na comunicação com o IEC 61850 Cliente que está conectado.

Tabela 178: Diagnósticos IEC 61850 Servidor

4.5.16.8. Intercepção de Comandos

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

Abaixo um exemplo de aplicação utilizando a linguagem ST mostrando a intercepção 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;
IEC61850_Generated_GVL.gfbIEC61850_LogicalDevice.XCBR1.fbBlkOpn.ctlModel :=
  IEC61850ServerXtorm.tyIEC61850_AT_ENUM_CtlModels.IEC61850_AT_CTLModels_
  DIRECT_WITH_ENHANCED_SEC;

// 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
```

4.6. Desempenho da Comunicação

4.6.1. 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. Por exemplo, um dispositivo MODBUS em uma aplicação que é executada a cada 100 ms, com um tempo de execução de 50 ms, irá ter um desempenho menor do que com uma aplicação de 25 ms executando a cada 100 ms de intervalo. Isto acontece porque, no segundo caso, a UCP terá um tempo maior entre cada ciclo da MainTask para executar as tarefas com prioridade mais baixa.

Também deve-se levar em conta o número de ciclos que o dispositivo, escravo ou servidor, leva para responder a uma requisição.

Para processar e transmitir uma resposta, um MODBUS RTU Escravo irá levar dois ciclos (tempo do ciclo da tarefa MODBUS), enquanto que um MODBUS Ethernet Servidor levará apenas um ciclo. Porém, esse é o tempo mínimo entre a recepção de uma requisição e o envio da resposta. Caso a requisição seja enviada logo após a execução de um ciclo da tarefa MODBUS, o tempo poderá ser equivalente a 2 ou 3 vezes o tempo de ciclo para o MODBUS Escravo, e de 1 a 2 vezes o tempo de ciclo para o MODBUS Servidor.

Neste caso: Tempo Máximo de Resposta = 3*(tempo de ciclo) + (tempo de execução das tarefas) + (tempo interframe chars) + (send delay).

Por exemplo, para uma tarefa MODBUS Ethernet Servidor com um ciclo de 50 ms, em uma aplicação executada por 60 ms a cada 100 ms, o servidor conseguirá executar apenas um ciclo entre cada ciclo da aplicação. Por outro lado, com a mesma aplicação, sendo executada por 10 ms, mas com um intervalo de 100 ms, o MODBUS terá um desempenho melhor, pois enquanto a aplicação não estiver em execução, o mesmo estará sendo executado a cada 50 ms e apenas a cada ciclo da MainTask ele irá demorar mais para ser executado. Para estes casos, o pior desempenho será a soma do tempo de execução da aplicação do usuário com o tempo do ciclo da tarefa MODBUS.

Para os dispositivos 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 da seguinte maneira:

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 igual:

- 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 igual:

- 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.

É importante ressaltar que o número de dispositivos MODBUS em execução também irá alterar o seu desempenho. Em uma aplicação de usuário com tempo de execução de 60 ms e intervalo de 100 ms, irá restar 40 ms para a UCP executar todas as tarefas de menor prioridade. Portanto, uma UCP com apenas um MODBUS Ethernet Servidor terá um desempenho maior do que uma UCP que utilize quatro destes dispositivos.

4.6.1.1. 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.

ATENÇÃO

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

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

$$N = - C \times (Z - (Z / (T \times 1000)))$$

$$Z = 1 / T$$

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).

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

$$C = 1; T = 0,05 \text{ s}; Z = 1 / 0,05 = 20;$$

$$N = 1 \times (20 - (20 / (0,05 \times 1000))) = 1 \times (20 - (20 / 50)) = 1 \times (20 - 0,4) = 1 \times 19,6$$

$$N = 19,6$$

Ou seja, nesta configuração o Servidor MODBUS será capaz de responder, em média, 19 requisições por segundo.

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

4.6.2. Comunicação DNP3

4.6.2.1. 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 também da concorrência com tarefas de outros protocolos configurados na UCP.

Para auxiliar na compreensão do desempenho do driver DNP3 Servidor, a seguir é apresentado o resultado de testes realizados com um simulador DNP3 Cliente conectado a uma HX3040 com um driver DNP3 Servidor. A base de dados configurada era composta de 4000 pontos digitais e 500 pontos analógicos (todos com qualidade e estampa de tempo), e o tempo de execução da MainTask estava em 70ms (intervalo de 100ms).

- Tempo para completar um polling de integridade: menos de um segundo.
- Tempo para transferir 4000 eventos digitais + 500 eventos analógicos: 4 segundos.

4.6.2.2. 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 50ms, 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 50ms, independente do período de polling configurado nas requisições. Desta forma, o período de polling efetivamente executado pelo driver pode apresentar uma imprecisão de até 150ms.

4.6.3. Comunicação IEC 60870-5-104

4.6.3.1. 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 também da concorrência com tarefas de outros protocolos configurados na UCP.

Para auxiliar na compreensão do desempenho do driver IEC 60870-5-104 Servidor, a seguir é apresentado o resultado de testes realizados com um simulador IEC 60870-5-104 Cliente conectado a uma HX3040 com um driver IEC 60870-5-104 Servidor. A base de dados configurada era composta de 4000 pontos digitais e 500 pontos analógicos (todos com qualidade e estampa de tempo), e o tempo de execução da MainTask estava em 70ms (intervalo de 100ms).

- Tempo para completar um comando de interrogação geral: menos de um segundo.
- Tempo para transferir 4000 eventos digitais + 500 eventos analógicos: 25 segundos.

4.6.3.2. 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.

4.7. Desempenho do Sistema

Em casos onde a aplicação possui apenas uma tarefa de usuário MainTask, responsável pela execução de uma única unidade de programação do tipo Programa denominada MainPrg, o 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.

ATENÇÃO

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, como descrito na seção [Não Carregar a Aplicação na Inicialização](#).

4.7.1. 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. Aguarda completar o intervalo configurado para a MainTask.

Conforme descrito acima, fica claro que 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 onde utiliza-se a tarefa ProfTask e seja necessário 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

ATENÇÃO

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

ATENÇÃO

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.

4.7.2. Cartão de Memória

A transferência de dados envolvendo o cartão de memória é realizada em segundo plano pela UCP, pois esta dá prioridade para a execução da aplicação de usuário e processamento da comunicação. Desta forma, a transferência de arquivos para o cartão poderá sofrer um acréscimo de tempo significativo, dependendo do tempo de execução da aplicação do usuário.

O tempo necessário para ler/escrever arquivos no cartão será afetado diretamente pelo tempo de execução da aplicação de usuário, uma vez que essa aplicação tem prioridade na execução.

Para mais informações sobre a utilização do cartão de memória, ver seção [Cartão de Memória](#).

4.8. Relógio RTC

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

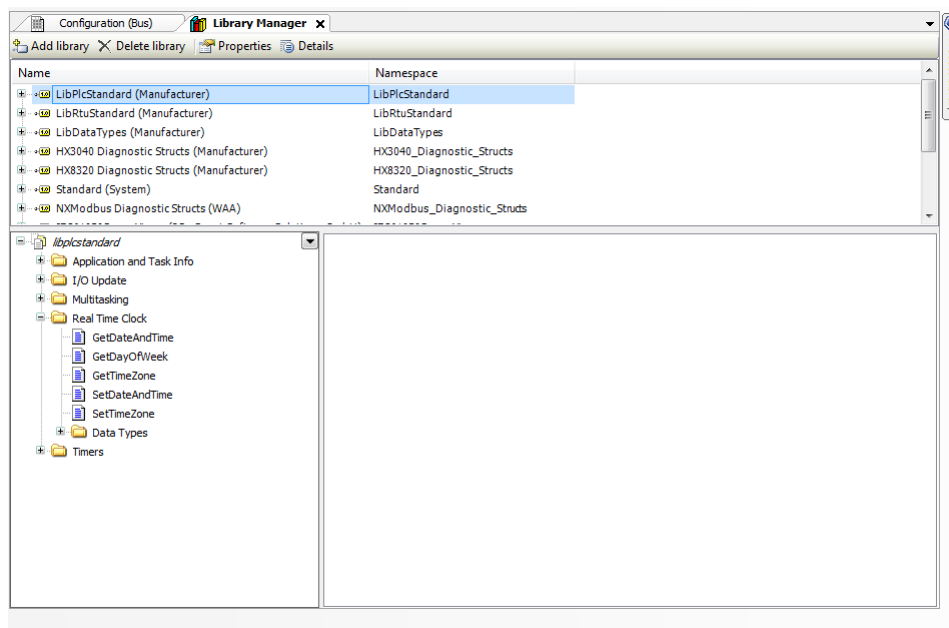


Figura 121: Blocos de Escrita e Leitura do Relógio

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

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

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

4.8.1.1. Funções de Leitura do RTC

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

4.8.1.1.1. *GetDateAndTime*

A função *GetDateAndTime* é utilizada para fazer a leitura da data e hora com o retorno em milissegundos.

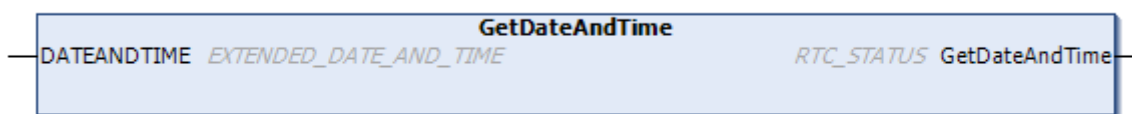


Figura 122: Leitura da Data e Hora

Parâmetros de entrada e saída	Tipo	Descrição
DATEANDTIME	EXTENDED_DATE_AND_TIME	Recebe os valores de data e hora com milissegundos. Ver seção EXTENDED_DATE_AND_TIME .

Tabela 179: Parâmetros de Entrada e Saída Função *GetDateAndTime*

Parâmetros de saída	Tipo	Descrição
GetDateAndTime	RTC_CMD_STATUS	Retorna o resultado da leitura/configuração. Ver seção RTC_CMD_STATUS .

Tabela 180: Parâmetros de Saída Função GetDateAndTime

Exemplo de utilização em Linguagem ST:

```
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_IF
```

4.8.1.1.2. *GetDayOfWeek*

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

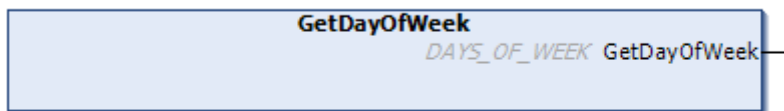


Figura 123: Leitura do Dia da Semana

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

Tabela 181: Parâmetros de Saída Função GetDayOfWeek

Quando chamada, a função lerá o dia da semana e preencherá a estrutura *DAYS_OF_WEEK*.

Exemplo de utilização em Linguagem ST:

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

4. CONFIGURAÇÃO

4.8.1.1.3. GetTimeZone

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

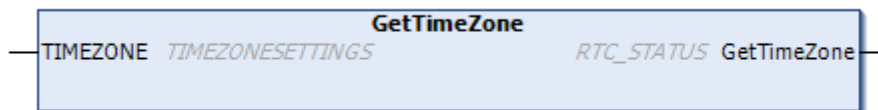


Figura 124: Leitura das Configurações de Fuso Horário

Parâmetros de entrada e saída	Tipo	Descrição
TIMEZONE	TIMEZONESETTINGS	Estrutura com o valor de fuso horário lido. Ver seção TIME ZONE SETTINGS .

Tabela 182: Parâmetros de Entrada e Saída Função GetTimeZone

Parâmetros de saída	Tipo	Descrição
GetTimeZone	RTC_CMD_STATUS	Retorna o resultado da leitura/configuração. Ver seção RTC_CMD_STATUS .

Tabela 183: Parâmetros de Saída Função GetTimeZone

Exemplo de utilização em Linguagem ST:

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.

```
PROGRAM UserPrg
VAR
  RTCStatus    : RTC_CMD_STATUS;
  TimeZone     : TIMEZONESETTINGS;
  xRead        : BOOL;
END_VAR

-----
//FB GetTimeZone
IF (xRead = TRUE) THEN
  GetTimeZone (TimeZone);
  xRead := FALSE;
END_IF
```

4.8.1.2. Blocos Funcionais e Funções de Escrita e Configuração do RTC

4.8.1.2.1. SetDateAndTime

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

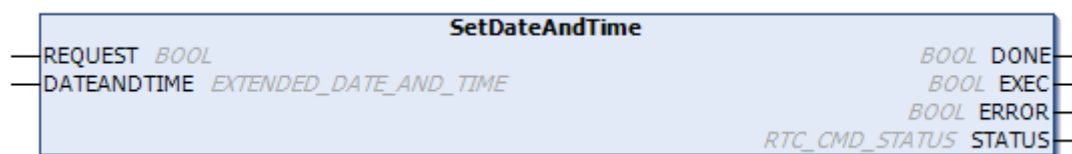


Figura 125: Ajuste de Data e Hora em Milissegundos

Parâmetros de entrada	Tipo	Descrição
REQUEST	BOOL	Essa variável, quando recebe uma borda de subida, habilita a escrita do relógio.
DATEANDTIME	EXTENDED_DATE_AND_TIME	Estrutura a ser preenchida com os valores de data e hora com milissegundos. Ver seção EXTENDED_DATE_AND_TIME .

Tabela 184: Parâmetros de Entrada Função SetDateAndTime

Parâmetros de saída	Tipo	Descrição
DONE	BOOL	Essa variável, quando verdadeira, indica que a ação foi completada.
EXEC	BOOL	Essa variável, quando verdadeira, indica que a função está processando os valores.
ERROR	BOOL	Essa variável, quando verdadeira, indica que ocorreu algum erro durante a Escrita.
STATUS	RTC_CMD_STATUS	Retorna o erro ocorrido durante a leitura/configuração. Ver seção RTC_CMD_STATUS .

Tabela 185: Parâmetros de Saída Função SetDateAndTime

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

Exemplo de utilização em Linguagem ST:

```

PROGRAM UserPrg
VAR
  SetDateAndTime : SetDateAndTime;
  xRequest       : BOOL;
  DateAndTime    : EXTENDED_DATE_AND_TIME;
  xDone          : BOOL;
  xExec          : BOOL;
  xError         : BOOL;
  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_IF
    
```

ATENÇÃO

Se o usuário tentar escrever valores de hora fora do intervalo do RTC, os valores serão convertidos para valores válidos, desde que não ultrapasse a faixa válida de 01/01/2000 até 31/12/2035. Por exemplo, se o usuário tentar escrever o valor 2000 ms, o mesmo será convertido para 2 segundos, se escrever o valor 100 segundos, o mesmo será convertido para 1 min e 40 segundos. Se escrever o valor de 30 horas, o mesmo será convertido para 1 dia e 6 horas, e assim por diante. 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.

4.8.1.2.2. *SetTimeZone*

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

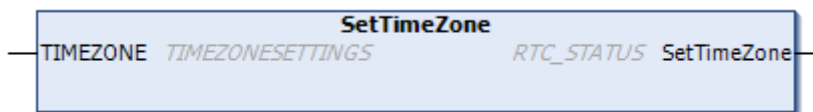


Figura 126: Ajuste de Fuso Horário

Parâmetros de entrada	Tipo	Descrição
TIMEZONE	TIMEZONESETTINGS	Estrutura com o valor de fuso horário a ser configurado. Ver seção TIME ZONE SETTINGS .

Tabela 186: Parâmetros de Entrada

Parâmetros de saída	Tipo	Descrição
SetTimeZone	RTC_CMD_STATUS	Retorna o erro ocorrido durante a configuração. Ver seção RTC_CMD_STATUS .

Tabela 187: Parâmetros de Saída

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

Exemplo de utilização em Linguagem ST:

```

PROGRAM UserPrg
VAR
  RTCStatus   : RTC_CMD_STATUS;
  TimeZone    : TIMEZONESETTINGS;
  xWrite      : BOOL;
END_VAR

-----

//FB SetTimeZone
IF (xWrite = TRUE) THEN
  RTCStatus := SetTimeZone (TimeZone);
  xWrite := FALSE;
END_IF
    
```

ATENÇÃO

Para realizar o acerto do relógio, devem-se utilizar valores de hora e datas dentro da seguinte faixa válida: 00:00:00 horas de 01/01/2000 até 23:59:59 horas de 31/12/2035, caso contrário, será reportado um erro através do parâmetro de saída *STATUS*. Para maiores detalhes do parâmetro de saída *STATUS*, consultar a seção [RTC_CMD_STATUS](#).

4.8.2. 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:

4.8.2.1. EXTENDED_DATE_AND_TIME

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

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

Tabela 188: Estrutura EXTENDED_DATE_AND_TIME

4.8.2.2. DAYS_OF_WEEK

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

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

Tabela 189: Estrutura DAYS_OF_WEEK

4.8.2.3. RTC_CMD_STATUS

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

Enumerador	Valor	Descrição
RTC_CMD_STATUS	NO_ERROR (0)	Não há erro.
	UNKNOWN_COMMAND (1)	Comando desconhecido.
	DEVICE_BUSY (2)	Dispositivo está ocupado.
	DEVICE_ERROR (3)	Dispositivo com erro.
	ERROR_READING_OSF (4)	Erro na leitura do sinalizador de data e hora válidas.
	ERROR_READING_RTC (5)	Erro na leitura da data e hora.
	ERROR_WRITING_RTC (6)	Erro na escrita da data e hora.
	ERROR_UPDATING_SYSTEM_TIME (7)	Erro na atualização de data e hora do sistema.
	INTERNAL_ERROR (8)	Erro interno.
	INVALID_TIME (9)	Data e hora inválidas.
	INPUT_OUT_OF_RANGE (10)	Fora do limite de Data e hora válidas para o sistema.
	TIMEZONE_CFG_NOT_SUPPORTED (11)	Erro gerado quando as configurações de fuso horário não são suportadas pelo dispositivo.

Tabela 190: RTC_CMD_STATUS

4.8.2.4. TIME_ZONE_SETTINGS

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

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

Tabela 191: TIMEZONESETTINGS

Nota:

Blocos funcionais de escrita e leitura de data e hora: Bibliotecas diferentes da *LibPlcStandard*, que tenham blocos funcionais ou funções que possam fazer acesso de leitura e escrita da data e hora no sistema, não são indicadas. A biblioteca

LibPlcStandard possui as interfaces adequadas para escrever e ler a data e hora do sistema adequadamente e informar os diagnósticos corretos.

4.9. 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 a seção [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 abaixo.

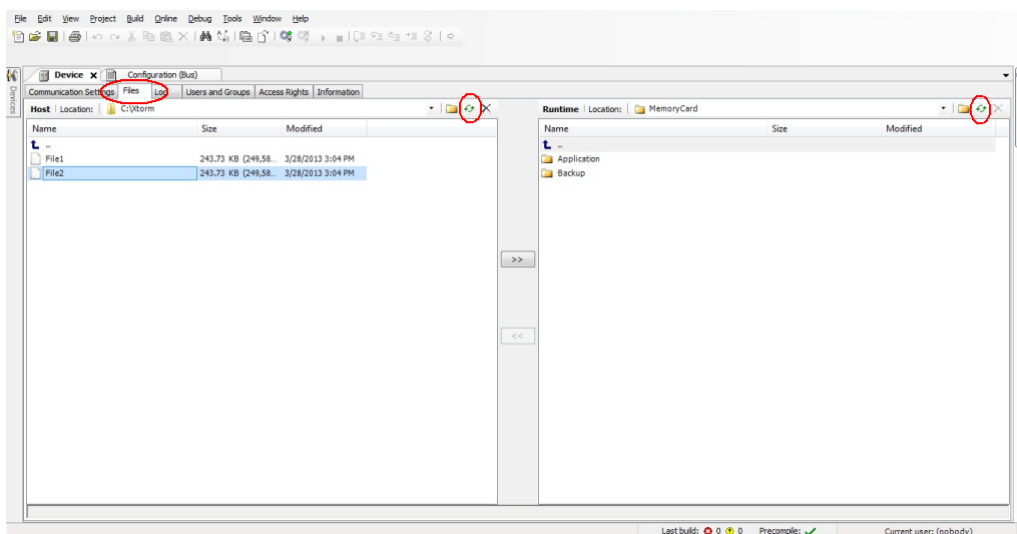


Figura 127: Acesso aos Arquivos de Usuário

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 o mesmo esteja inserido na UCP. Arquivos transferidos para a pasta *MemoryCard* estarão sendo transferidos diretamente para dentro do cartão de memória.

ATENÇÃO

No caso em que o cartão de memória seja inserido após a inicialização da UCP, será requisitado um usuário e senha para realizar o acesso e/ou transferências de arquivos do MasterTool 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 mostra a figura abaixo. O tempo de transferência irá variar de acordo com o tamanho do arquivo e com o tempo de ciclo (execução) da aplicação atual da UCP, podendo levar vários minutos.

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

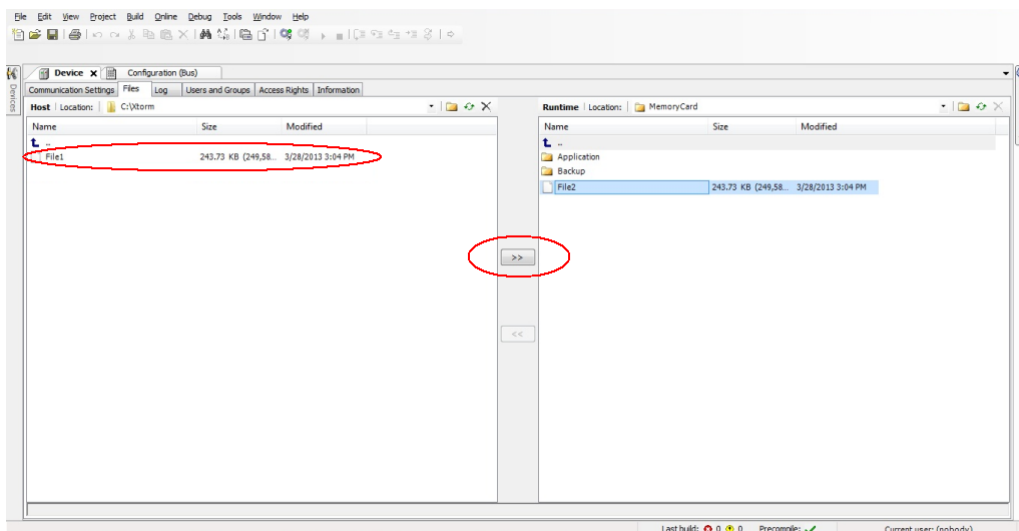





Figura 128: Transferindo Arquivos

ATENÇÃO

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.

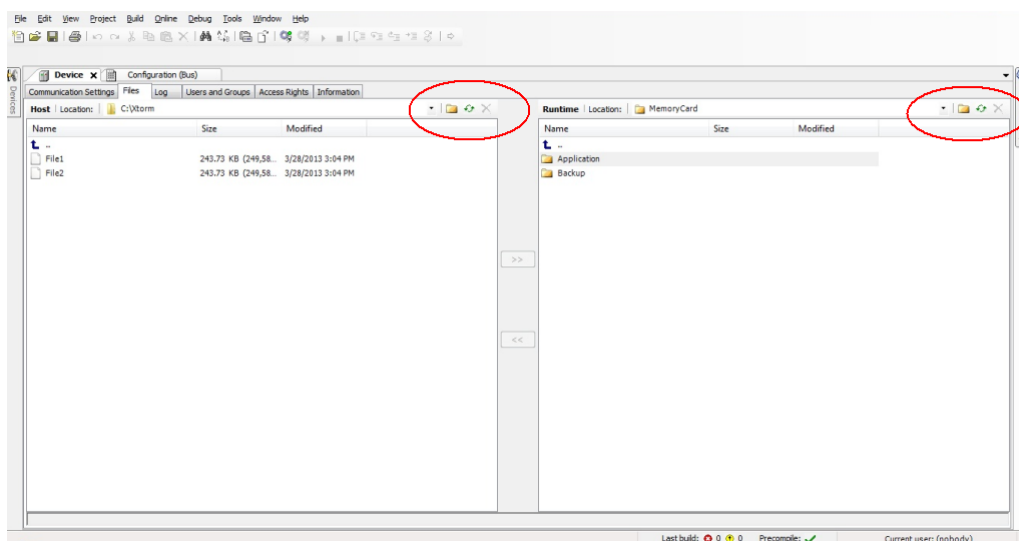


Figura 129: Opções de Utilização

ATENÇÃO

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

4.10. Cartão de Memória

Entre outras memórias, a UCP da Série Hadron Xtorm possibilita ao usuário a utilização de um cartão de memória, definido conforme as características da seção [Interface do Cartão de Memória](#), o qual serve para armazenar, entre outros arquivos, o projeto e a aplicação que está na memória interna da UCP.

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

ATENÇÃO

Recomenda-se realizar o procedimento de formatação do cartão de memória diretamente na UCP Hadron Xtorm para evitar possíveis problemas de utilização, aumento do tempo de montagem ou até mesmo funcionamento incorreto. Não é recomendado remover o cartão de memória ou desenergizar a UCP durante a formatação ou durante a transferência de arquivos, pois pode causar a perda de dados bem como danos irreversíveis ao cartão.

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. Informações sobre a tabela, consultar a seção [Cartão de Memória](#).

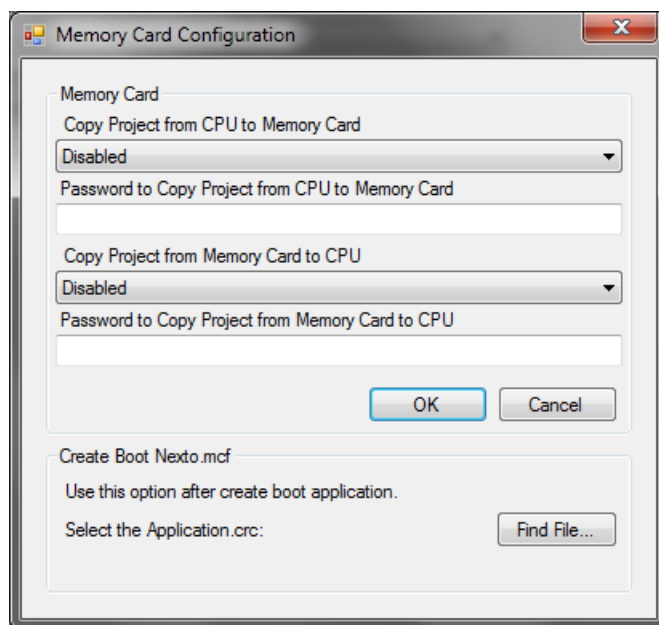


Figura 130: Configurações do Cartão de Memória

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 também seja enviado.

4. CONFIGURAÇÃO

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 consistir a 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. Maiores informações sobre a utilização do botão de diagnósticos, consultar a seção [One Touch Diag](#).

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.

ATENÇÃO

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 na 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.

ATENÇÃO

Caso exista algum arquivo na raiz do cartão de memória nomeado “*Application*” ou “*Backup*”, o mesmo 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.

4.10.1. 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.

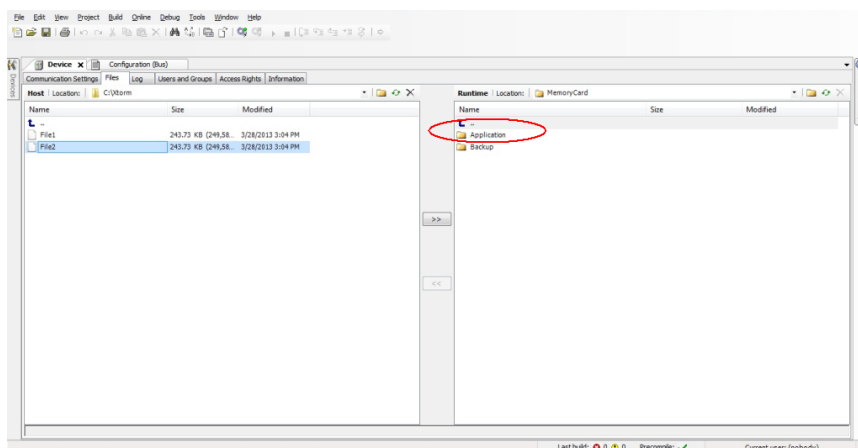


Figura 131: Diretório Raiz com Cartão de Memória Inserido

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 abaixo.

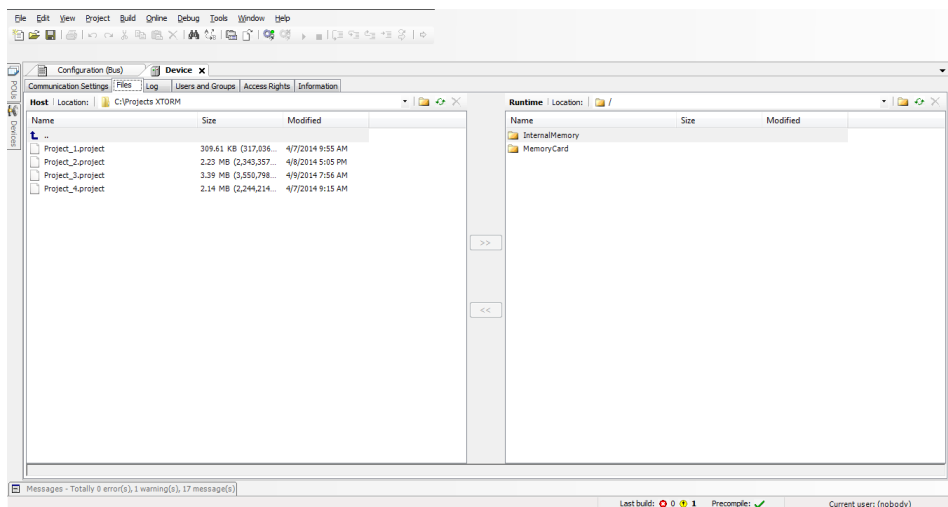


Figura 132: Arquivos Salvos no Cartão de Memória

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 dos mesmos através do menu da UCP.

ATENÇÃO

O tempo de transferência de arquivos depende da diferença do tempo de intervalo menos o tempo de execução médio da(s) tarefa(s) em execução (tempo disponível até o próximo ciclo da tarefa), isto é, quanto maior for essa diferença para cada tarefa em uma aplicação, mais rápida deverá ser a transferência de um dado a partir do cartão de memória para a UCP/MasterTool 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.

ATENÇÃO

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.

4.11. 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. Consultar a seção [One Touch Diag](#) para verificar o funcionamento e a diferença entre tipos de pressionamento no botão de diagnósticos.

A tabela abaixo mostra os níveis do menu e o tipo de cada tela disponível na UCP, ou seja, se ela é de caráter informativo, configurável ou se retorna um nível.

Nível 1	Nível 2	Nível 3	Tipo
HARDWARE	TEMPERATURA	-	Informativo
	CONTRASTE	NIVEL CONTRASTE	Configurável
	DATA E HORA	-	Informativo
	VOLTAR	-	Retorna Nível
IDIOMAS	ENGLISH	>ENGLISH	Configurável
	PORTUGUES	>PORTUGUES	Configurável
	ESPAÑOL	>ESPAÑOL	Configurável
	VOLTAR	-	Retorna Nível
REDE	ENDEREÇO IP NET 1	-	Informativo
	MASCARA NET 1	-	Informativo
	ENDEREÇO IP NET 2	-	Informativo
	MASCARA NET 2	-	Informativo
	ENDEREÇO IP NET 3	-	Informativo
	MASCARA NET 3	-	Informativo
	ENDEREÇO IP NET 4	-	Informativo
	MASCARA NET 4	-	Informativo
	ENDEREÇO IP NET 5	-	Informativo
	MASCARA NET 5	-	Informativo
	ENDEREÇO IP NET 6	-	Informativo
	MASCARA NET 6	-	Informativo
VOLTAR	-	Retorna Nível	
SOFTWARE	FIRMWARE	-	Informativo
	BOOTLOADER	-	Informativo
	PROC. AUX.	-	Informativo
	VOLTAR	-	Retorna Nível
REDUNDANCIA	TROCAR PARA ATIVO?	-	Operacional
	TROCAR PARA RESERVA?	-	Operacional
	VOLTAR	-	Retorna Nível
CARTAO DE MEM.	DESMONTAR	-	Operacional
	UCP > CARTAO	SENHA UCP	Operacional
	BACKUP	-	Operacional
	FORMATAR	CONFIRMA ?	Operacional
	CARTAO > UCP	SENHA CARTAO	Operacional
VOLTAR	-	Retorna Nível	
VOLTAR	-	-	Retorna Nível

Tabela 192: Níveis do Menu da UCP

Nota:

Cartão de Memória: O cartão de memória somente estará disponível no menu caso o mesmo esteja conectado na UCP Hadron Xtorm.

Senha: A senha para acesso aos dados do cartão de memória somente será necessária caso ela seja configurada no software MasterTool Xtorm. Não é possível editar a senha via menu. A senha precisa ter ao menos 6 caracteres.

Redundância: O menu “Redundancia” somente estará disponível caso a UCP HX3040 esteja identificada como Redundante.

Conforme apresentado pela tabela acima, entre as opções disponíveis para visualização e alteração, encontram-se os principais dados necessários ao usuário, como:

- Informações sobre os recursos de hardware:
 - TEMPERATURA – Temperatura interna da UCP (Ex.: 36 C 97 F)

- CONTRASTE – Ajuste do contraste do visor frontal da UCP
- DATA E HORA – Data e hora configuradas na UCP (Ex.: 2001.01.31 00:00)
- Alteração do idioma do menu da UCP:
 - PORTUGUES – Altera o idioma para Português
 - ENGLISH – Altera o idioma para Inglês
 - ESPANOL – Altera o idioma para Espanhol
- Visualização de informações sobre a rede configurada no dispositivo:
 - ENDERECO IP NET 1 – Endereço IP (Ex.: 192.168.0.1)
 - MASCARA NET 1 – Máscara de subrede (Ex.: 255.255.255.0)
 - ENDERECO IP NET 2 – Endereço IP (Ex.: 192.168.1.1)
 - MASCARA NET 2 – Máscara de subrede (Ex.: 255.255.255.0)
 - NET 3: DESAB. (Caso não esteja configurada no projeto de usuário)
 - NET 4: DESAB. (Caso não esteja configurada no projeto de usuário)
 - NET 5: DESAB. (Caso não esteja configurada no projeto de usuário)
 - NET 6: DESAB. (Caso não esteja configurada no projeto de usuário)
- Informações sobre as versões de software:
 - FIRMWARE – Versão de software da UCP (Ex.: 1.0.0.0)
 - BOOTLOADER – Versão do bootloader da UCP (Ex.: 1.0.0.0)
 - PROC. AUX. – Versão do processador auxiliar da UCP (Ex.: 1.0.0.0)
- Acesso ao menu de opções de redundância da UCP:
 - TROCAR PARA ATIVO?
 - TROCAR PARA RESERVA?
- Acesso aos dados do Cartão de Memória:
 - CARTAO>UCP – Transferência do projeto do cartão de memória para a UCP
 - UCP>CARTAO – Transferência do projeto da UCP para o cartão de memória
 - FORMATAR – Formata o cartão para o sistema de arquivos FAT32
 - DESMONTAR – Habilita a remoção segura do cartão de memória da UCP
 - BACKUP – Faz um backup dos dados da UCP no cartão de memória

A figura abaixo 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 as mesmas, sendo que para modificar os outros parâmetros, como Idioma e inserir a(s) senha(s) no Cartão de Memória, basta seguir a mesma lógica de acesso. O pressionamento curto mostra que o contraste está sendo incrementado (mais claro), sendo que no próximo pressionamento após o seu valor máximo, ele retorna ao valor mínimo (menos claro). O pressionamento longo mostra a confirmação do contraste desejado e o retorno ao nível anterior.

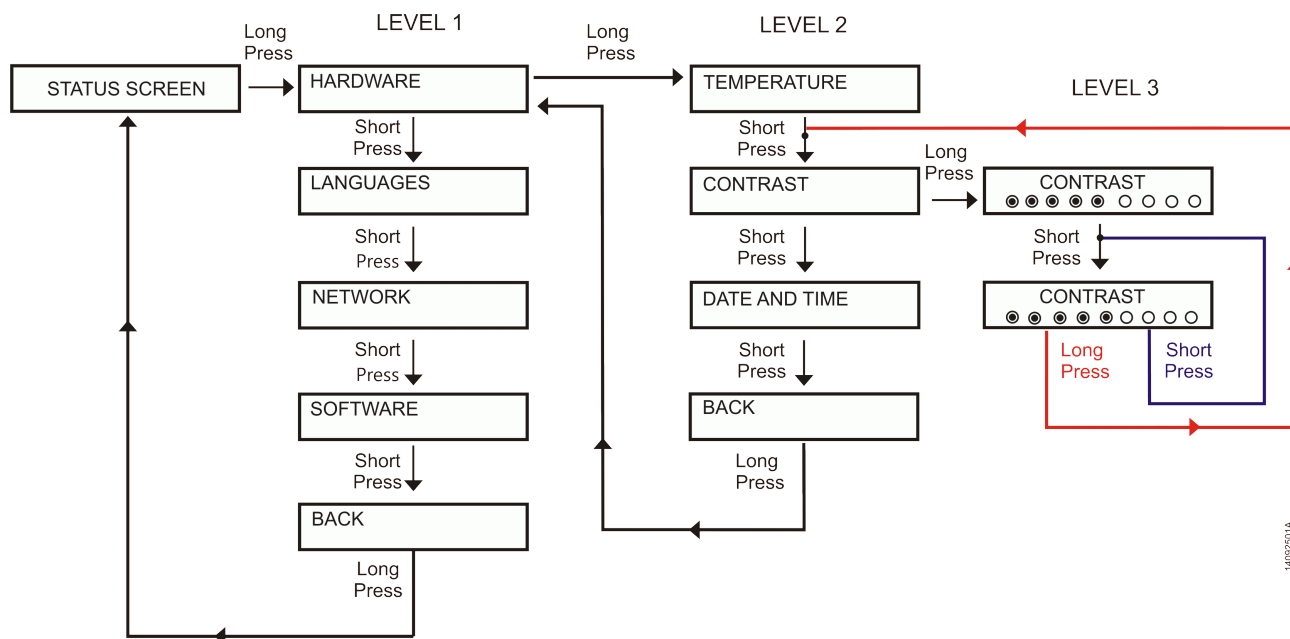


Figura 133: Ajuste do Contraste

Além do menu 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.

4.12. Blocos Funcionais e Funções

4.12.1. Atualização de Entradas e Saídas

Funcionalidade utilizada para atualizar entradas e saídas no decorrer da aplicação, não sendo necessário aguardar até que seja completado um ciclo.

Quando os blocos funcionais para atualizar as entradas e saídas não são utilizados, a atualização é realizada a cada ciclo da MainTask.

ATENÇÃO

Na inicialização de uma UCP desta série, as entradas e saídas somente estarão atualizadas para leitura e preparadas para escrita quando a MainTask for executada. Todas as demais tarefas do sistema que executarem antes da MainTask estarão com as entradas e as saídas inválidas.

4.12.1.1. REFRESH_INPUT

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

ATENÇÃO

A função de *REFRESH_INPUT* deve ser utilizada somente na tarefa MainTask. Para executar a atualização de entradas em outras tarefas, a opção *Habilita atualização de E/S por tarefa* deve ser selecionada, mais informações a respeito desta opção podem ser obtidas na Tabela 33.

ATENÇÃO

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

ATENÇÃO

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

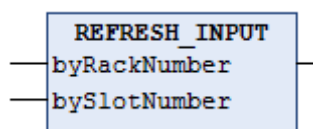


Figura 134: Bloco para Atualizar as Entradas

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

Tabela 193: Parâmetros de Entrada REFRESH_INPUT

Possíveis *TYPE_RESULT*:

- **OK_SUCCESS:** Execução com sucesso.
- **ERROR_FAILED:** Esse erro é retornado caso a função seja chamada para um módulo que possua somente saídas, ou também caso a opção Sempre atualizar variáveis (localizada na tela de configuração do módulo, aba Mapeamento de E/S) não esteja marcada.
- **ERROR_NOTSUPPORTED:** A função não é suportada pelo produto.
- **ERROR_PARAMETER:** Parâmetro inválido / não suportado.
- **ERROR_MODULE_ABSENT:** O módulo está ausente.
- **ERROR_MODULE_NOTCONFIGURED:** O módulo não está configurado na aplicação.
- **ERROR_MODULE_NOTRUNNING:** O módulo não está em funcionamento (não está no estado operacional).
- **ERROR_MODULE_COMMFAIL:** Falha de comunicação com o módulo.
- **ERROR_MODULE_NOTFOUND:** O módulo não foi encontrado na aplicação ou não é suportado.

Exemplo de utilização em Linguagem ST:

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

4.12.1.2. REFRESH_OUTPUT

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

ATENÇÃO

A função de *REFRESH_OUTPUT* deve ser utilizada somente na tarefa MainTask. Para executar a atualização de saídas em outras tarefas, a opção *Habilita atualização de E/S por tarefa* deve ser selecionada, mais informações a respeito desta opção podem ser obtidas na Tabela 33.

ATENÇÃO

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

ATENÇÃO

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

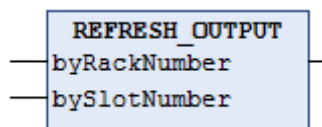


Figura 135: Bloco para Atualizar as Saídas

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

Tabela 194: Parâmetros de Entrada REFRESH_OUTPUT

Possíveis *TYPE_RESULT*:

- **OK_SUCCESS:** Execução com sucesso.
- **ERROR_FAILED:** Esse erro é retornado caso a função seja chamada para um módulo que possua somente entradas, ou também caso a opção Sempre atualizar variáveis (localizada na tela de configuração do módulo, aba Mapeamento de E/S) não esteja marcada.
- **ERROR_NOTSUPPORTED:** A função não é suportada pelo produto.
- **ERROR_PARAMETER:** Parâmetro inválido / não suportado.
- **ERROR_MODULE_ABSENT:** O módulo está ausente.
- **ERROR_MODULE_NOTCONFIGURED:** O módulo não está configurado na aplicação.
- **ERROR_MODULE_NOTRUNNING:** O módulo não está em funcionamento (não está no estado operacional).
- **ERROR_MODULE_COMMFAIL:** Falha de comunicação com o módulo.
- **ERROR_MODULE_NOTFOUND:** O módulo não foi encontrado na aplicação ou não é suportado.

Exemplo de utilização em Linguagem ST:

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

4.12.2. 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).

ATENÇÃO

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 (para realizar o procedimento de inserção de uma biblioteca, consultar a seção [Bibliotecas](#)).

4.12.2.1. TOF_RET

O bloco funcional *TOF_RET* implementa um tempo de atraso para desabilitar uma saída. Quando a entrada *IN* tem seu estado alterado de verdadeiro (TRUE) para falso (FALSE), ou seja, uma borda de descida, o tempo especificado *PT* irá transcorrer até que a saída *Q* também seja falsa (FALSE). Quando a entrada *IN* tem nível lógico 1 (TRUE), a saída *Q* também permanecerá no mesmo estado (TRUE), mesmo que isso aconteça no meio de uma contagem. O tempo *PT* pode ser alterado durante a contagem, pois o bloco funcional assumirá o novo valor, desde que a contagem não tenha chegado ao final. A Figura 136 representa o bloco *TOF_RET* e a Figura 137 mostra o comportamento gráfico do mesmo.



Figura 136: Bloco Funcional TOF_RET

Parâmetros de entrada	Tipo	Descrição
IN	BOOL	Essa variável, quando recebe uma borda de descida, habilita a contagem do bloco funcional.
PT	TIME	Essa variável especifica o limite de contagem do bloco funcional (tempo de atraso).

Tabela 195: Parâmetros de Entrada TOF_RET

Parâmetros de saída	Tipo	Descrição
Q	BOOL	Essa variável vai para FALSE assim que a variável PT (tempo de atraso) atinge o seu valor máximo.
ET	TIME	Essa variável exhibe o valor atual do tempo de atraso.

Tabela 196: Parâmetros de Saída TOF_RET

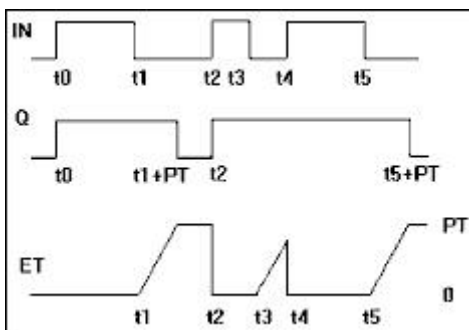


Figura 137: Comportamento Gráfico do Bloco Funcional TOF_RET

Exemplo de utilização em linguagem ST:

```

PROGRAM UserPrg
VAR RETAIN
bStart : BOOL := TRUE;
TOF_RET : TOF_RET;
END_VAR

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

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

4.12.2.2. TON_RET

O bloco funcional *TON_RET* implementa um tempo de atraso para habilitar uma saída. Quando a entrada *IN* tem seu estado alterado de falso (FALSE) para verdadeiro (TRUE), ou seja, uma borda de subida, o tempo especificado *PT* irá transcorrer até que a saída *Q* também seja verdadeira (TRUE). Quando a entrada *IN* tem nível lógico 0 (FALSE), a saída *Q* também permanecerá no mesmo estado (FALSE), mesmo que isso aconteça no meio de uma contagem. O tempo *PT* pode ser alterado durante a contagem, pois o bloco funcional assumirá o novo valor, desde que a contagem não tenha chegado ao final. A Figura 138 representa o bloco *TON_RET* e a Figura 139 mostra o comportamento gráfico do mesmo.



Figura 138: Bloco Funcional TON_RET

Parâmetros de entrada	Tipo	Descrição
IN	BOOL	Essa variável, quando recebe uma borda de subida, habilita a contagem do bloco funcional.
PT	TIME	Essa variável especifica o limite de contagem do bloco funcional (tempo de atraso).

Tabela 197: Parâmetros de Entrada TON_RET

Parâmetros de saída	Tipo	Descrição
Q	BOOL	Essa variável vai para TRUE assim que a variável PT (tempo de atraso) atinge o seu valor máximo.
ET	TIME	Essa variável exibe o valor atual do tempo de atraso.

Tabela 198: Parâmetros de Saída TON_RET

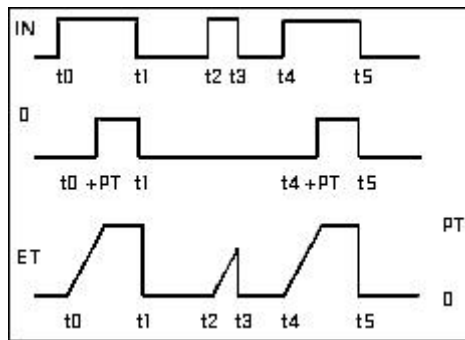


Figura 139: Comportamento Gráfico do Bloco Funcional TON_RET

Exemplo de utilização em linguagem ST:

```

PROGRAM UserPrg
VAR RETAIN
bStart : BOOL;
TON_RET : TON_RET;
END_VAR

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

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

4.12.2.3. TP_RET

O bloco funcional *TP_RET* trabalha como um trigger. O temporizador, que inicia quando a entrada *IN* tem seu estado alterado de falso (FALSE) para verdadeiro (TRUE), ou seja, uma borda de subida, é incrementado até que o limite de tempo *PT* seja atingido. Durante a contagem, a saída *Q* é verdadeira (TRUE), caso contrário ela é falsa (FALSE). O tempo *PT* pode ser alterado durante a contagem, pois o bloco assumirá o novo valor, desde que a contagem não tenha chegado ao final. A Figura 140 representa o bloco *TP_RET* e a Figura 141 mostra o comportamento gráfico do mesmo.



Figura 140: Bloco Funcional TP_RET

Parâmetros de entrada	Tipo	Descrição
IN	BOOL	Essa variável, quando recebe uma borda de subida, habilita a contagem do bloco funcional.
PT	TIME	Essa variável especifica o limite de contagem do bloco funcional (tempo de atraso).

Tabela 199: Parâmetros de Entrada TP_RET

Parâmetros de saída	Tipo	Descrição
Q	BOOL	Essa variável é verdadeira durante a contagem. Caso contrário é falsa.
ET	TIME	Essa variável exibe o valor atual do tempo de atraso.

Tabela 200: Parâmetros de Saída TP_RET

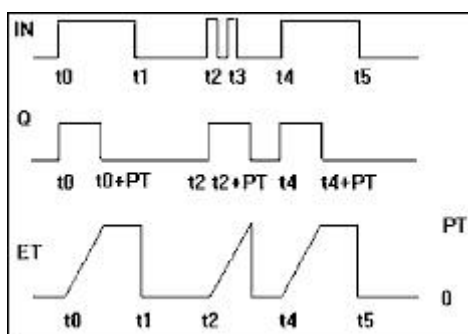


Figura 141: Comportamento Gráfico do Bloco Funcional TP_RET

Exemplo de utilização em linguagem ST:

```

PROGRAM UserPrg
VAR RETAIN
bStart : BOOL;
TP_RET : TP_RET;
END_VAR

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

bStart := FALSE;

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

```

4.12.3. 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.

Abaixo, são descritos os três tipos de blocos já disponíveis na biblioteca *LibPlcStandard* do software MasterTool Xtorm (para realizar o procedimento de inserção de uma biblioteca, consultar a seção [Bibliotecas](#)).

4.12.3.1. TOF_NR

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

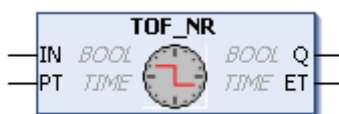


Figura 142: Bloco Funcional TOF_NR

Exemplo de utilização em linguagem ST:

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

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

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

4.12.3.2. TON_NR

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



Figura 143: Bloco Funcional TON_NR

Exemplo de utilização em linguagem ST:

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

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

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

4.12.3.3. TP_NR

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



Figura 144: Bloco Funcional TP_NR

Exemplo de utilização em linguagem ST:

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

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

bStart := FALSE;

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

4.12.4. Log de Usuário

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

Há duas funções disponíveis, uma para registrar informações e outra para remover todos os registros. A seguir uma descrição dos tipos de dados utilizados pelas funções:

Tipo de dado	Opção	Descrição
USER_LOG_EVENT_TYPES	USER_LOG_EVENT_ERROR	O usuário é livre para usar a melhor indicação conforme a severidade de sua mensagem de log.
	USER_LOG_EVENT_DEBUG	
	USER_LOG_EVENT_INFO	
	USER_LOG_EVENT_WARN	
USER_LOG_MESSAGE		Mensagem de log com o máximo de 150 caracteres.
USER_LOG_ERROR_CODES	USER_LOG_OK	A operação foi executada com sucesso.
	USER_LOG_FAILED	A operação não foi executada com sucesso. O motivo da falha pode ser verificado nos logs de sistema – ver seção Log de Sistema .
	USER_LOG_BUFFER_FULL	Mensagens estão sendo adicionadas além da capacidade de processamento.
	USER_LOG_NO_MEMORY	No momento, não houve recursos para executar a operação.
	USER_LOG_FILE_SYSTEM_ERROR	Houve algum erro no acesso ao cartão de memória ou não há espaço disponível. Informações do erro podem ser verificadas nos logs de sistema – ver seção Log de Sistema .
	USER_LOG_NO_MEMORY_CARD	Não há um cartão de memória presente na UCP.
	USER_LOG_MEMORY_CARD_FULL	Não há espaço livre disponível no cartão de memória.
	USER_LOG_PROCESSING	O recurso está ocupado executando a última operação, por exemplo, deletando todos os arquivos de log.

Tabela 201: Tipos de dados para Log de Usuário

A seguir, são descritas as duas funções disponíveis na biblioteca LibLogs no MasterTool Xtorm 1.10. Para realizar o procedimento de inserção de uma biblioteca, consultar a seção [Bibliotecas](#).

ATENÇÃO

Os Logs de Usuário estão disponíveis apenas a partir da versão 1.1.0.12 da UCP da Série Hadron Xtorm. Da mesma forma para utilizar esta funcionalidade é necessária a versão 1.10 ou superior do MasterTool Xtorm.

4.12.4.1. UserLogAdd

Essa função é utilizada para adicionar uma nova mensagem de log do usuário, acrescentando em uma nova linha ao arquivo de log no cartão de memória. A mensagem deve ter um tamanho máximo de 150 caracteres e o tipo de evento da mensagem.

Variáveis da aplicação podem ser registradas utilizando conversão para string e concatenação com a mensagem principal. É adicionada automaticamente na mensagem a informação de data e hora em UTC (estampa de tempo) com resolução de milissegundos em que o evento foi registrado. A informação de data e hora também é utilizada na formação dos nomes dos arquivos de log.

A função *UserLogAdd* pode ser utilizada para inserir várias mensagens dentro de uma mesma tarefa e também em tarefas diferentes da aplicação. Porém independente de cada execução da função na aplicação, sendo ela na mesma tarefa ou em tarefas diferentes, todas utilizam o mesmo recurso para gravar as mensagens desejadas. Por esse motivo é recomendado que a adição de mensagens utilizando a função *UserLogAdd* na aplicação seja realizada a cada 50 ms para evitar retorno de buffer cheio. Caso a função seja executada em períodos menores que o indicado, mas respeite o tempo médio de 50 ms entre cada adição de mensagem ao término do intervalo da tarefa, também evita o retorno de buffer cheio. Para que os logs sejam adicionados corretamente, é importante respeitar os limites de tempo quando o cartão é inserido e na inicialização da UCP, mencionados na seção [Cartão de Memória](#). Após a operação a função retorna as opções para o tipo de dado *USER_LOG_ERROR_CODES*, conforme a Tabela 201. Quando o retorno da função for diferente de *USER_LOG_OK*, a mensagem não foi registrada, devendo ser executada novamente a função *UserLogAdd* com a mensagem desejada. Em caso de retorno de falhas consecutivas de escrita, o cartão de memória pode estar danificado. A substituição por um cartão de memória íntegro garante que as últimas mensagens registradas serão gravadas no cartão que não está danificado, desde que a UCP não seja reiniciada.

A figura abaixo representa a função *UserLogAdd* e a tabela abaixo os parâmetros de entrada:

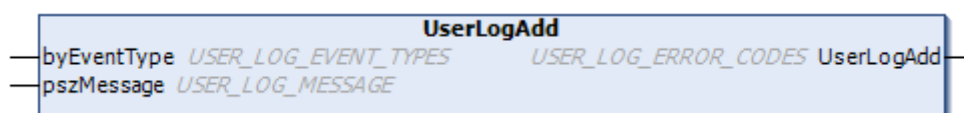


Figura 145: Função UserLogAdd

Parâmetros de entrada	Tipo	Descrição
byEventType	BYTE	Essa variável especifica o tipo de evento do log que está sendo adicionado conforme as opções para o tipo de dado <code>USER_LOG_EVENT_TYPES</code> .
pszMessage	<code>USER_LOG_MESSAGE</code>	Essa variável deve conter o conjunto de caracteres que compõe a mensagem a ser adicionada no arquivo de log. A mensagem deve conter no máximo 150 caracteres.

Tabela 202: Parâmetros de Entrada UserLogAdd

Os arquivos de log são gerados e organizados no cartão de memória em um caminho de diretórios específico que depende do número de série da UCP e da versão de firmware instalada. Por exemplo, para uma UCP com número de série 445627 e versão de firmware 1.4.0.4, o local onde os arquivos de log devem ser gravados no cartão de memória é *MemoryCard/UserLog/445627/1.4.0.4/*.

Os nomes dos arquivos de log são formados pela data e hora (estampa de tempo) da primeira mensagem. Exceto quando há algum problema para utilizar esse nome, como por exemplo, outro arquivo existente com o mesmo nome, nessa situação são utilizadas a data e hora instantâneas. O nome do arquivo segue o seguinte padrão: *ano/mês/dia/hora/minuto/segundo/milissegundo.CSV*. Caso um arquivo apresente problema de acesso por setor defeituoso e não seja possível continuar com escrita, será adicionado ao nome desse arquivo a extensão *“corrupted”* e um novo arquivo será criado. A quantidade de logs por arquivo não é fixa, variando em função do tamanho das mensagens. A quantidade de arquivos criados é limitada em 1024 com tamanho máximo de 1 MB cada, portanto o cartão de memória necessita de 1 GB de espaço livre. Quando atingir o limite de 1024 arquivos criados no cartão de memória, durante a operação da UCP, os arquivos mais antigos são removidos para que os arquivos com logs mais recentes sejam preservados, mesmo nos casos de remoção manual parcial dos arquivos no diretório onde os arquivos estão sendo gravados.

A visualização dos arquivos de log pode ser realizada através de planilhas ou editores de texto convencionais. As informações concatenadas, para melhor visualização, podem utilizar ponto e vírgula entre as strings da mensagem para separá-las. Deve-se ter o cuidado na formatação de células com valores em ponto flutuante.

Exemplo de utilização em Linguagem ST:

4. CONFIGURAÇÃO

```
PROGRAM UserPrg
VAR
  eLogError : USER_LOG_ERROR_CODES;
  sMessage : USER_LOG_MESSAGE;
END_VAR

IF (m_rTemperature > MAX_TEMPERATURE_ACCEPT) THEN
  sMessage := 'Temperatura acima do esperado: ';
  sMessage := concat(sMessage, REAL_TO_STRING(m_rTemperature));
  sMessage := concat(sMessage, '°');
  eLogError := UserLogAdd(USER_LOG_EVENT_WARN, sMessage);
//Variável eLogError recebe possíveis erros da função.
END_IF
```

Exemplo de conteúdo do arquivo de log: (*UserLog-201308271506245738.csv*)

```
Model; NX3030
Serial number; 445627
Firmware version; 1.4.0.4

27/08/2013 15:06:24.5738; WARN; Temperatura acima do esperado: 25°
27/08/2013 16:37:45.3476; WARN; Temperatura acima do esperado: 25°
28/08/2013 09:10:55.4201; WARN; Temperatura acima do esperado: 26°
```

4.12.4.2. UserLogDeleteAll

A função *UserLogDeleteAll* realiza a exclusão dos arquivos de log presentes no diretório criado especificamente para a UCP em que está inserido o cartão de memória, ou seja, são excluídos apenas os logs contidos no diretório nomeado com a versão de firmware da UCP que existe dentro do diretório com a versão de série da UCP. Os arquivos de log excluídos são apenas os arquivos existentes no momento da montagem do cartão de memória e os gerados pela função *UserLogAdd*. Registros de outras UCPs e arquivos adicionados manualmente pelo usuário durante a execução não são apagados. A figura abaixo representa a função *UserLogDeleteAll*.

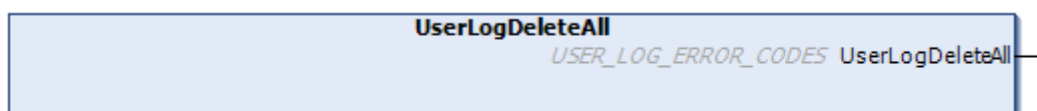


Figura 146: Função UserLogDeleteAll

Exemplo de utilização em Linguagem ST:

```
PROGRAM UserPrg
VAR
  eLogError : USER_LOG_ERROR_CODES;
END_VAR
```

```

IF (m_DeleteLogs = TRUE) THEN
//Variável eLogError recebe possíveis erros da função.
eLogError := UserLogDeleteAll();
m_DeleteLogs := FALSE;
END_IF
    
```

ATENÇÃO

O retorno da função *UserLogDeleteAll* não indica operação concluída, apenas confirmação de execução que pode levar um período de tempo grande caso existam centenas de arquivos de log no diretório. A função para registro de novos logs de usuário estará indisponível nesse momento, retornando a opção *USER_LOG_PROCESSING* para qualquer operação. O resultado da operação também pode ser verificado no log de sistema.

4.12.5. 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 também pode resetar a flag de fila de eventos limpa. Conforme a tabela a seguir, a ação que a função executará é definido pelo código do ENUM atribuído na entrada *eDiagnostic*.

Nome	ENUM (BYTE)	Descrição
DIAG_RETENTIVITY_LOST	0	Limpa diagnóstico de Perda de Retentividade.
DIAG_QUEUE_OVERFLOW	1	Limpa diagnóstico de Fila de Eventos cheia / estourada.
DIAG_QUEUE_CLEARED	2	Limpa diagnóstico de Fila de Eventos limpa.

Tabela 203: Descrição das Ações da Função ClearRtuDiagnostic

A tabela a seguir descreve o status de execução da função será mostrado na sua variável de saída.

Nome	ENUM (UDINT)	Descrição
OK_SUCCESS	0	Sucesso.
ERROR_PARAMETER	3	Erro de parâmetros (valor atribuído na entrada <i>eDiagnostic</i> fora da faixa válida).

Tabela 204: Resultados da Função ClearRtuDiagnostic

Exemplo de utilização em linguagem ST, onde a chamada da função irá limpar o bit *DG_HX3040.tSummarized*. *bRetentivityLost* da estrutura de diagnósticos da UCP:

```

PROGRAM UserPrg
VAR
Resultado : TYPE_RESULT;
END_VAR

Resultado := ClearRtuDiagnostic(eDiagnostic := 0);
    
```

4.12.6. ClearEventQueue

A função *ClearEventQueue* disponibilizada pela biblioteca *LibRtuStandard* pode ser utilizada quando for necessário limpar a fila de eventos da UCP e de todos os drivers instanciados.

Conforme a tabela abaixo, o resultado de execução da função será mostrado na sua variável de saída.

Nome	ENUM (UDINT)	Descrição
OK_SUCCESS	0	Sucesso
ERROR_FAILED	1	Erro
ERROR_NOTSUPPORTED	2	A rotina chamada não é suportada pelo produto
ERROR_PARAMETER	3	Parâmetro inválido/não suportado
ERROR_MODULE_ABSENT	16	O módulo está ausente no barramento
ERROR_MODULE_NOTCONFIGURED	17	O módulo não está configurado na aplicação
ERROR_MODULE_NOTRUNNING	18	O módulo não está rodando (não está em estado operacional)
ERROR_MODULE_COMMFAIL	19	Falha na comunicação com o módulo
ERROR_MODULE_NOTFOUND	20	O módulo não foi encontrado na aplicação ou não é suportado

Tabela 205: Resultados da função ClearEventQueue

Exemplo de utilização em linguagem ST, onde a chamada da função irá limpar a fila de eventos e, conseqüentemente, zerar os diagnósticos de uso das filas de eventos dos drivers de comunicação *T_DIAG_DNP_SERVER_1.tClient_*.tQueueDiags.wUsage*:

```
PROGRAM UserPrg
VAR
  ClearEventQueueStatus : TYPE_RESULT;
END_VAR

ClearEventQueueStatus := ClearEventQueue();
```

4.13. 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.

4.13.1. Gerenciamento de Usuários e Direitos de Acesso do Projeto

4.13.1.1. 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.

ATENÇÃO

- 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.
- Por padrão, em novos projetos, a senha do usuário *Owner* é vazia.

4.13.1.1.1. Usuários

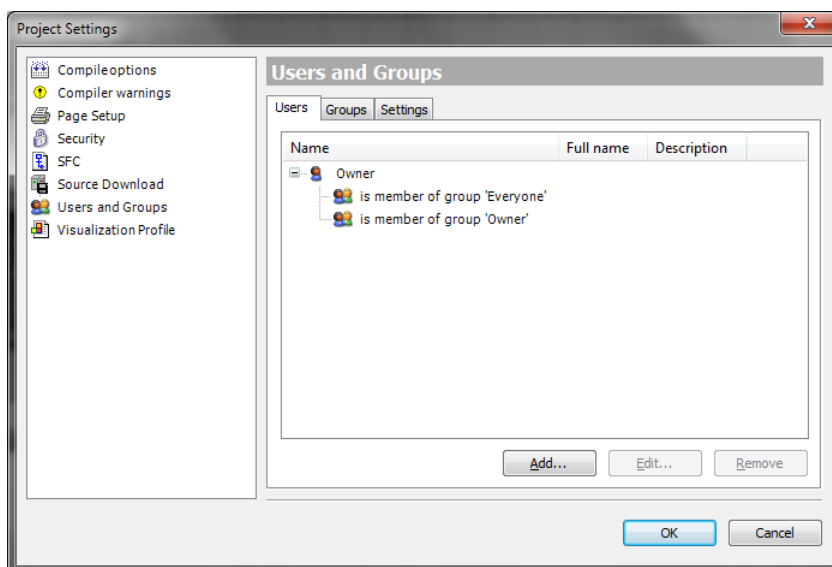


Figura 147: Configurações do Projeto - Usuários

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*.

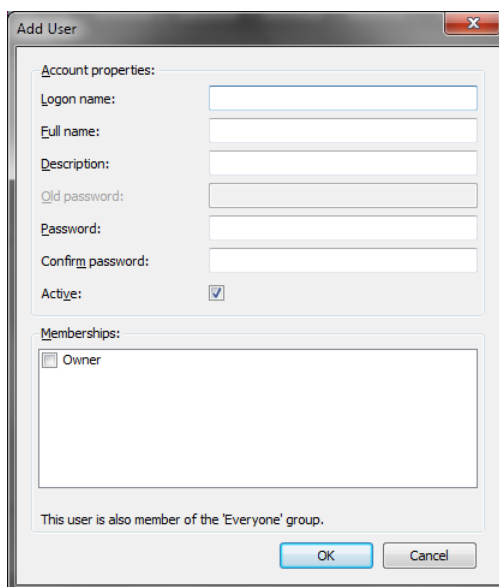


Figura 148: Acrescentar Usuário no Projeto

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 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 tecla *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.

4.13.1.1.2. Grupos

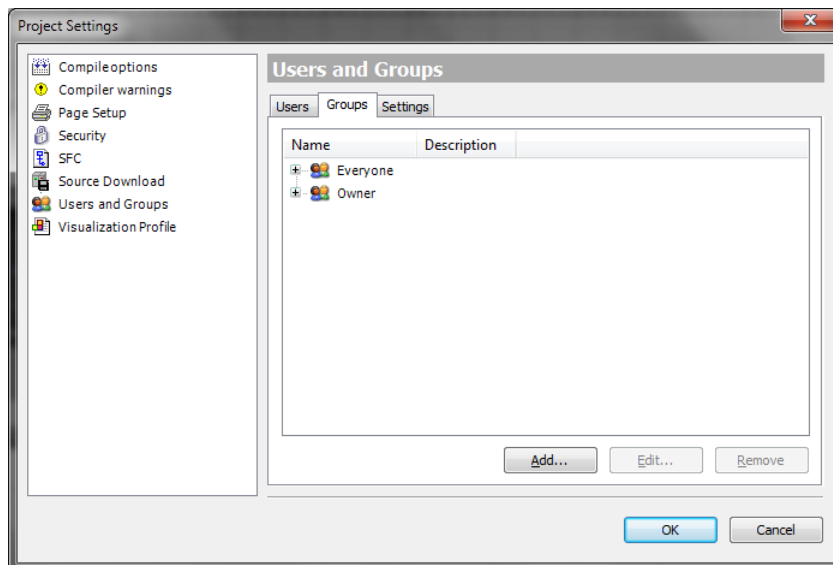


Figura 149: Configurações de Projeto - Grupos

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 que o membro deve ser parte de um grupo.

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

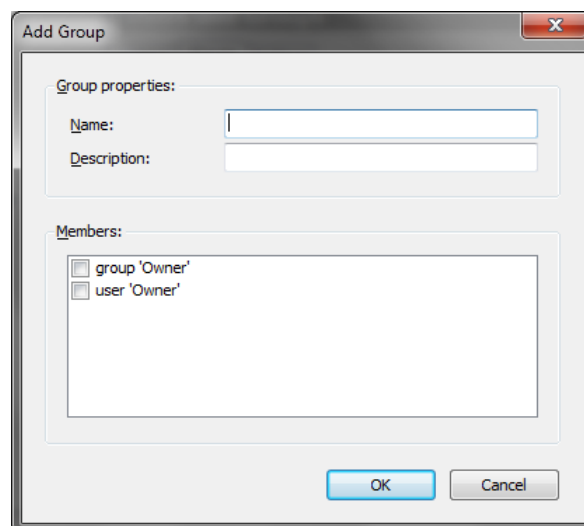


Figura 150: Acrescentar Grupo

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.

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 acima). Os campos da senha - por questões de segurança - apresentarão 32 asteriscos (*). Após ter modificado os itens desejados, feche o diálogo com *OK* para aplicar as novas configurações.

Para remover um ou vários grupos, selecione os respectivos grupos na árvore e tecla *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.

4.13.1.1.3. Configurações

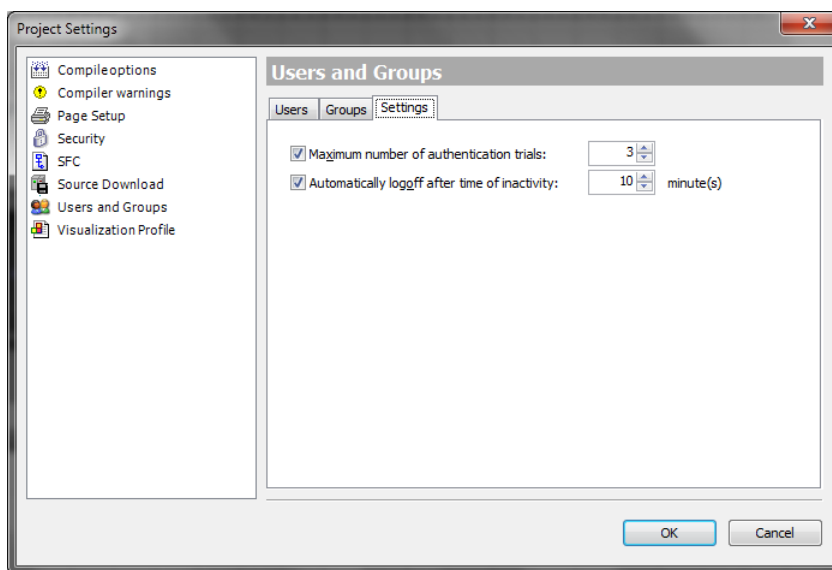


Figura 151: Configurações de Projeto - Configuração de Usuários e Grupos

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.

4.13.1.2. 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.

4. CONFIGURAÇÃO

Para acessar a tela *Permissions* deve-se clicar nessa opção no menu *Project > User Management*. Será aberta a tela da figura abaixo.

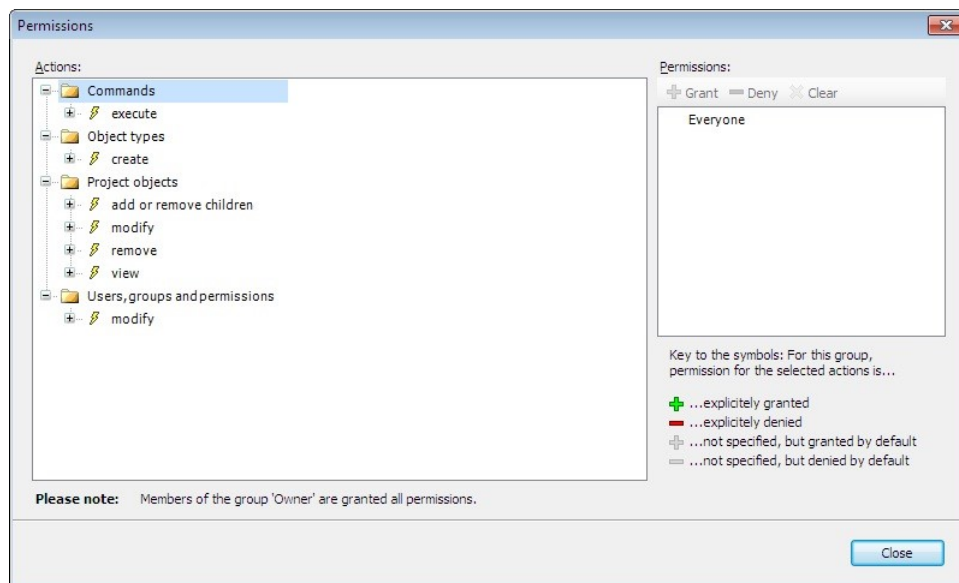











Figura 152: Permissões

A janela *Actions* exibe todas as possibilidades de direitos, ou seja, todas as ações que podem ser executadas nos objetos. A árvore é estruturada da seguinte forma:


- : No topo da árvore encontram-se os nomes de algumas categorias configuradas para estruturar visualmente os direitos de acesso. Estas categorias referem-se à execução de Comandos, à configuração de Contas de usuários e grupos, à criação de Tipos de objetos e também à visualização, edição, exclusão e tratamento de objetos secundários nos Objetos do projeto.
- : Em cada categoria encontram-se nós com as ações específicas que podem ser executadas em relação ao comando, conta do usuário, grupo, tipo de objeto ou objeto do projeto. Estes nós possuem também apenas função visual.




Possíveis ações:

-  *Execute* (execução de um comando do menu).
-  *Create* (criação de um novo objeto no projeto atual).
-  *Add or remove children* (de um objeto existente).
-  *Modify* (edição de um objeto no editor).
-  *Remove* (excluir ou recortar um objeto).
-  *View* (visualização de um objeto no editor).
-  Cada item “ação” pode conter “dispositivos”, ou seja, objetos (do projeto).

A janela *Permissões* fornece uma lista de todos os grupos de usuários disponíveis (exceto o grupo *Owner*) e uma barra de ferramentas para configurar os direitos de acesso ao grupo.




À esquerda de cada nome do grupo encontram-se ícones que indicam a permissão atual referente ao dispositivo selecionado na janela *Actions*:

- : A ação selecionada atualmente na janela *Actions* é garantida ao grupo.

-  : A ação selecionada atualmente na janela *Actions* é negada ao grupo.
-  : O direito de executar uma ou várias ações selecionadas na janela *Actions* não é garantido explicitamente, mas sim por padrão, devido ao referido direito ter sido garantido ao objeto "principal". Basicamente esta é a configuração padrão para todos os direitos que não tenham sido explicitamente configurados.
-  : O direito de executar uma ou várias ações selecionadas na janela *Actions* não é negado explicitamente, mas sim por padrão, por exemplo, nos casos em que o referido direito tenha sido atribuído ao objeto principal.

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

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

-  **Permitir** : Garantir explicitamente.
-  **Negar** : Negar explicitamente.
-  **Limpar** : Os direitos garantidos para a ação selecionada atualmente na janela *Actions* serão excluídos (retorno à configuração padrão).

4.13.2. Gerenciamento de Usuário 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 abaixo exemplifica os passos para acessar estas abas da UCP.

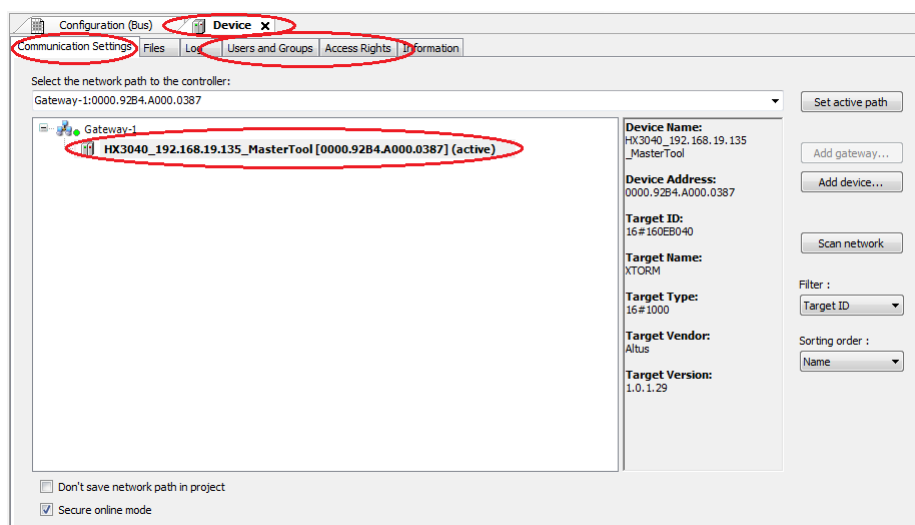


Figura 153: Acesso às Abas Usuários e Grupos e Direitos de Acesso

ATENÇÃO

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

4.13.2.1. 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.

4.13.2.1.1. Comum

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.

4.13.2.1.2. Usando o Diálogo de Configuração

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.

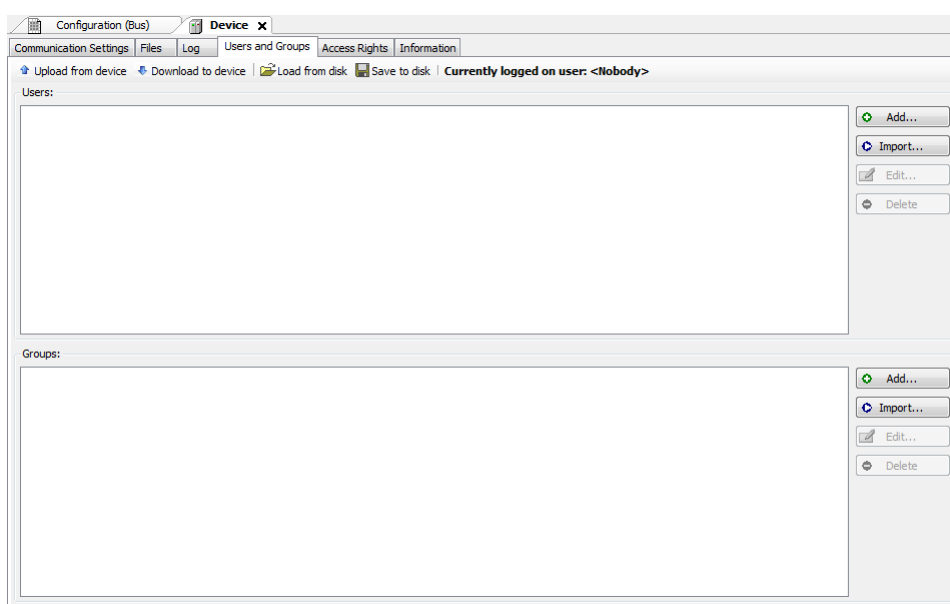
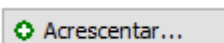


Figura 154: Diálogo Device, Usuários e Grupos

4.13.2.1.3. Usando o Diálogo de Configuração - 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*.

ATENÇÃO

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.

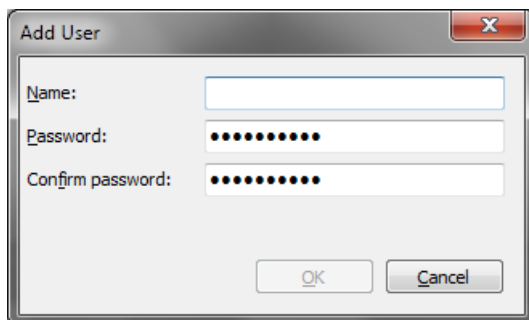
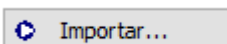
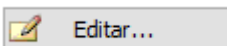


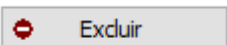
Figura 155: Acrescentar Usuário no Device



: 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.

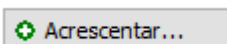


: O nome e senha da conta do usuário atualmente selecionada pode ser modificada. O diálogo *Edit User <nome do usuário>* corresponde ao diálogo *Add User*.



: A conta do usuário atualmente selecionada será excluída.

4.13.2.1.4. Usando o Diálogo de Configuração - 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.

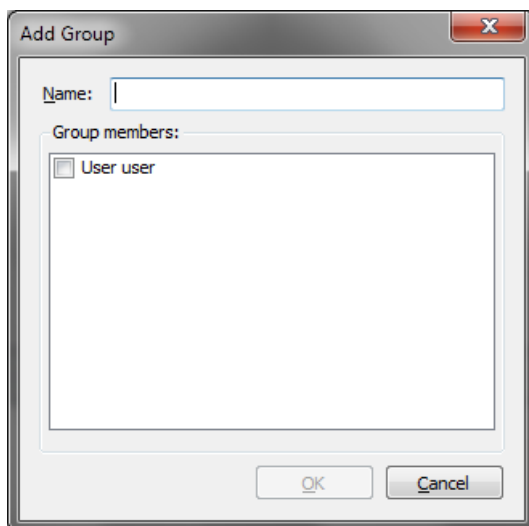
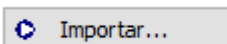
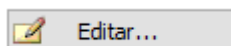


Figura 156: Acrescentar Grupo - UCP

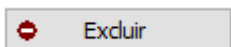


: O diálogo *Import Groups* apresenta uma lista com os grupos atualmente definidos no gerenciamento de usuários do projeto. Selecione um ou mais itens e confirme com *OK* para integrá-los à lista de grupos do gerenciamento de usuários específico do dispositivo.

4. CONFIGURAÇÃO



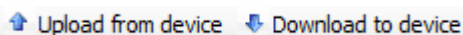
: O grupo atualmente selecionado pode ser modificado no que se refere ao seu nome e usuários associados. Para tanto, usa-se o diálogo *Edit Group <nome do grupo>*, o qual corresponde ao diálogo *Add Group*.



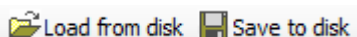
: O grupo atualmente selecionado será excluído.

4.13.2.1.5. Aplicando e Armazenando a Configuração Atual

Veja os botões respectivos 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 *Archive*) e "Document..." (menu *Project*), respectivamente.

4.13.2.1.6. Considerações sobre Usuário e Grupos Padrão

Nas versões atuais do firmware existem os usuários: *Administrator* e *Everyone*. E, também, os grupos: *Administrator*, *Developer*, *Everyone*, *Service* e *Watch*. Conforme mostra a tabela abaixo:

Usuários	Grupos
Administrator	Administrator
Everyone	Developer
	Everyone
	Service
	Watch

Tabela 206: Usuários e Grupos

4.13.2.1.7. Considerações sobre Usuário e Grupos Padrão - Usuários e Grupos Padrão

Os seguintes grupos e usuários estão definidos por padrão nas UCPs da Série Hadron Xtorm. Esta divisão em um maior número de grupos visa apresentar uma proposta inicial de diferentes níveis de usuários que possam acessar a UCP.

4.13.2.1.8. Considerações sobre Usuário e Grupos Padrão - Usuários e Grupos Padrão - Grupo Administrador

Este grupo possui todos os privilégios e não é possível removê-lo. O grupo *Developer* faz parte deste grupo.

4.13.2.1.9. Considerações sobre Usuário e Grupos Padrão - Usuários e Grupos Padrão - 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.

4.13.2.1.10. Considerações sobre Usuário e Grupos Padrão - Usuários e Grupos Padrão - Grupo Everyone

Este é o grupo padrão para realizar os acessos em uma UCP enquanto não existam grupos e usuários definidos.

4.13.2.1.11. Considerações sobre Usuário e Grupos Padrão - Usuários e Grupos Padrão - Grupo Service

Grupo criado para definir direitos de acesso a usuários que prestem algum tipo de serviço nos CPs, como por exemplo, equipes de manutenção. O grupo *Watch* faz parte deste grupo. Se não for utilizado este grupo pode ser excluído.

4.13.2.1.12. Considerações sobre Usuário e Grupos Padrão - Usuários e Grupos Padrão - Grupo Watch

Grupo criado para definir difeitos 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.

4.13.2.1.13. Considerações sobre Usuário e Grupos Padrão - Usuários e Grupos Padrão - Usuário Administrador

O usuário Administrator está definido nos grupos *Everyone* e *Administrator*. A senha padrão do usuário Administrator é "Administrator" e pode ser modificada.

4.13.2.1.14. Considerações sobre Usuário e Grupos Padrão - Usuários e Grupos Padrão - Usuário Everyone

O usuário Everyone está definido nos grupos *Everyone* e *Administrator*. Este usuário não possui senha definida.

4.13.2.1.15. 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.

4.13.2.2. 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 abaixo mostra a permissão para adicionar e remover nós secundários para/do objeto PLC Logic concedido ao grupo de usuários *Everyone* e *Owner*.

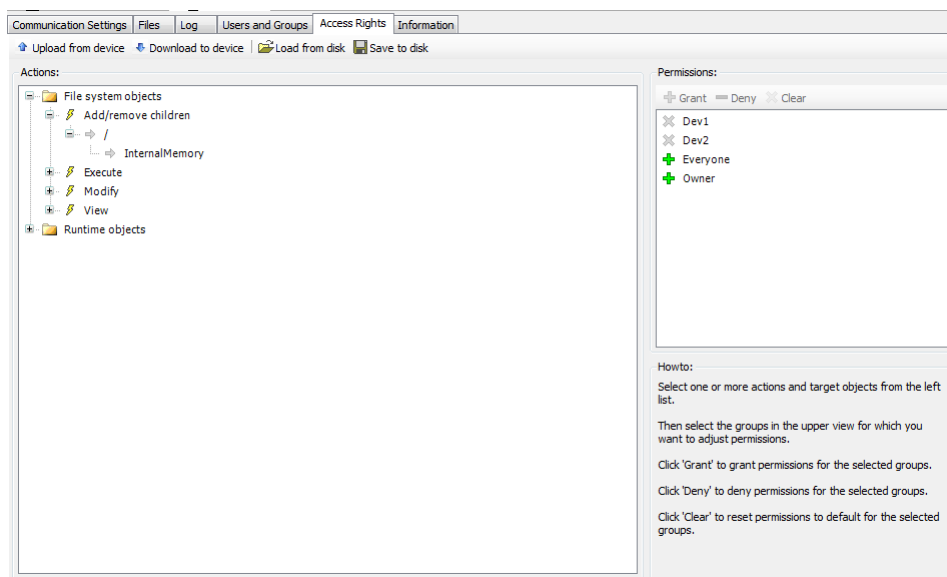


Figura 157: Direitos de Acesso do Dispositivo

Os direitos de acesso são configurados para cada dispositivo e habilitam para cada usuário as ações conforme a tabela abaixo. Eles podem ser definidos de acordo com a necessidade do usuário. O mesmo pode definir o tipo de acesso ao projeto, podendo ser permitido ou não Adicionar / Remover, Modificar, Visualizar e Executar.

Objetos de execução			Direitos de acesso			
			Adicionar / Remover	Modificar	Visualizar	Executar
Device	Logger	-			X	
	PlcLogic	Application	X	X	X	X
		_Backup&Restore		X	X	
	PlcShell	-		X		
	RemoteConnections	-	X	X	X	X
	Settings	-		X	X	
	UserMenagement	-		X	X	
X509	-		X	X		

Tabela 207: Ações e Direitos

Nota:

Direitos de acesso: São as permissões que podem ser configuradas de acordo com as necessidades. A tabela acima assinala, conforme cada tipo de objeto, quais direitos / permissões podem ser definidos. Além disso, é possível definir as permissões de cada tipo de grupo do dispositivo, de acordo com a disponibilidade do objeto de execução. 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.

4.13.2.2.1. 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 *Permitir* ou *Negar* (também na janela *Permissions*).

Veja a seguir a descrição das janelas de diálogo específicas.

4.13.2.2.2. Definindo as Permissões de Acesso - 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 or 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.

ATENÇÃO

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.




4.13.2.2.3. Definindo as Permissões de Acesso - 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*.

- : A ação selecionada atualmente na janela *Actions* é garantida para o grupo.
- : A ação selecionada atualmente na janela *Actions* é negada 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:

-  **Permitir**: Permissão de acesso explicitamente concedida.
-  **Negar**: Permissão de acesso explicitamente negada.
-  **Limpar**: O direito de acesso concedido para as ações atualmente selecionadas na janela *Actions* será apagado, ou seja, ele retornará para o padrão.

4.13.2.2.4. Aplicando e Armazenando a Configuração Atual

Veja os botões respectivos na barra superior do diálogo.

 **Upload from device**  **Download to device**: 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.

 **Load from disk**  **Save to disk**: 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 ser impressas ou documentadas através dos comandos “Print...” (menu *Archive*) e “Document...” (menu *Project*), respectivamente.

4.13.2.2.5. 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.

4.14. Acesso a Página Web de Gerenciamento

Desenvolvida para realizar a configuração e o acesso aos diagnósticos de algumas funcionalidades. A aba *Gerenciamento* de Página Web de Sistema tem seu acesso protegido por usuário e senha, sendo *admin* o valor padrão para ambos os campos.

Na aba *Gerenciamento*, existem outros recursos como, por exemplo, *Sistema*, *Rede*, *Dispositivo USB*, *Firewall*, *OpenVPN* e *Servidor FTP*. Os recursos disponíveis nesta aba variam de acordo com as funcionalidades disponíveis para o controlador utilizado e somente podem ser acessados após o usuário realizar o Login, conforme mostra a figura abaixo.

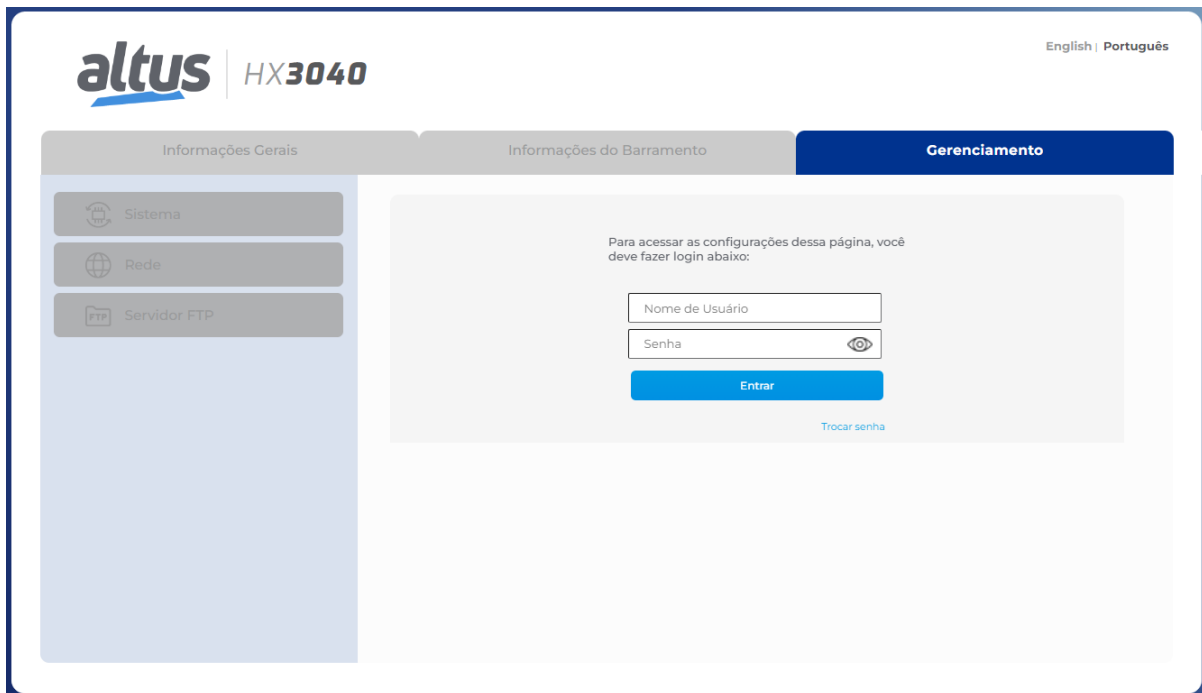


Figura 158: Acesso ao Gerenciamento Web

4.14.1. Página de Sistema

Na seção de *Sistema* da aba *Gerenciamento* é possível realizar a atualização de firmware da CPU. Para casos em que a atualização é realizada remotamente (através de uma conexão de rádio ou satélite por exemplo), a velocidade mínima deste link deve ser de 128Kbps.

4.14.1.1. Ajuste do Relógio

Na Página Web de Sistema, é possível realizar o ajuste de relógio do controlador, que se encontra na Seção Sistema da aba Gerenciamento da Página. O formato de data e horário seguem a norma ISO 8601 para amostragem de data e horário (AAAA/MM/DD hh:mm:ss), como mostra a imagem abaixo:

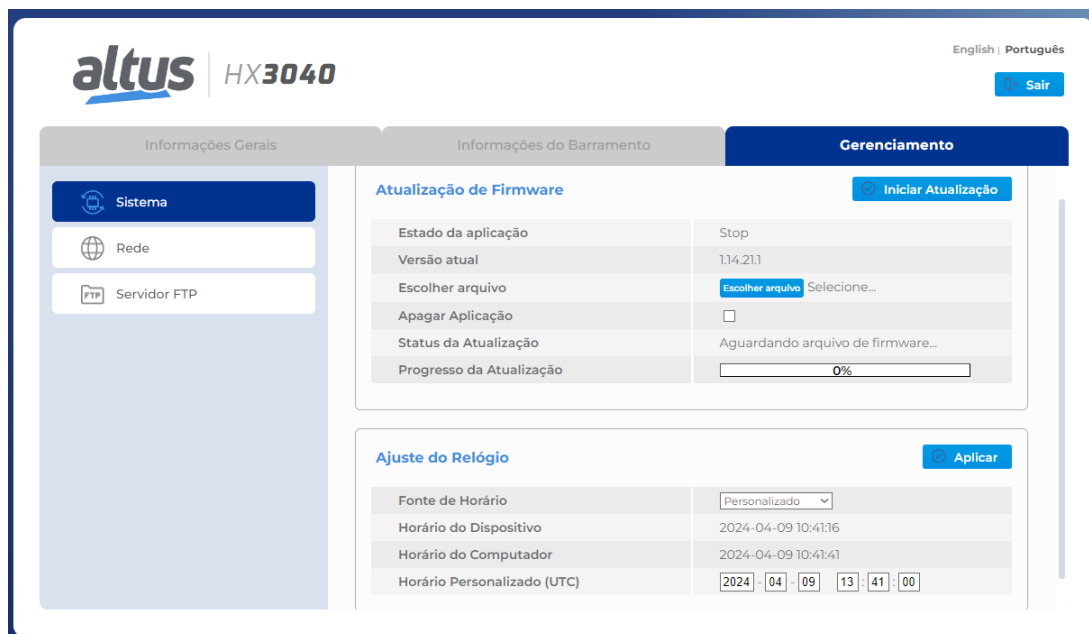


Figura 159: Ajuste do Relógio

Este recurso possui dois modos de ajuste do horário do dispositivo, que podem ser selecionados no item “Fonte de Horário”, disponibilizando ao usuário duas opções para sincronizar o relógio.

4.14.1.1.1. Horário do Computador (UTC)

No modo de horário do computador, o usuário pode aplicar o horário configurado em seu computador em UTC para o seu dispositivo. Para isso, deve-se selecionar no item “Fonte de Horário” a opção “Computador”. Após clicar no botão “Aplicar”, é necessário a validação das credenciais do dispositivo, então a UCP receberá a data e o horário UTC que estão configurados no computador.

4.14.1.1.2. Horário Personalizado (UTC)

No modo de horário personalizado, o usuário pode preparar um horário personalizado no padrão UTC para ser aplicado na data e horário interna de seu dispositivo. Para isso, deve-se selecionar no item “Fonte de Horário” a opção “Personalizado”. Com o modo selecionado, o usuário deverá configurar a data e o horário desejado no item “Horário Personalizado (UTC)”, que será inicializada com o horário local do navegador. Então, após o usuário clicar no botão “Aplicar” e validar as credenciais do dispositivo, o mesmo estará com seu horário interno configurado com o horário configurado no item “Horário Personalizado (UTC)”.

ATENÇÃO

O menor valor de data e horário configurável é 2000/01/01 00:00:00. Já o maior valor de data e horário é 2035/12/31 23:59:59.

4.14.2. Página de Rede

Desenvolvida para auxiliar na usabilidade do controlador, a seção de Rede (figura abaixo) possibilita a alteração dos endereços de rede e a execução do Sniffer de rede.

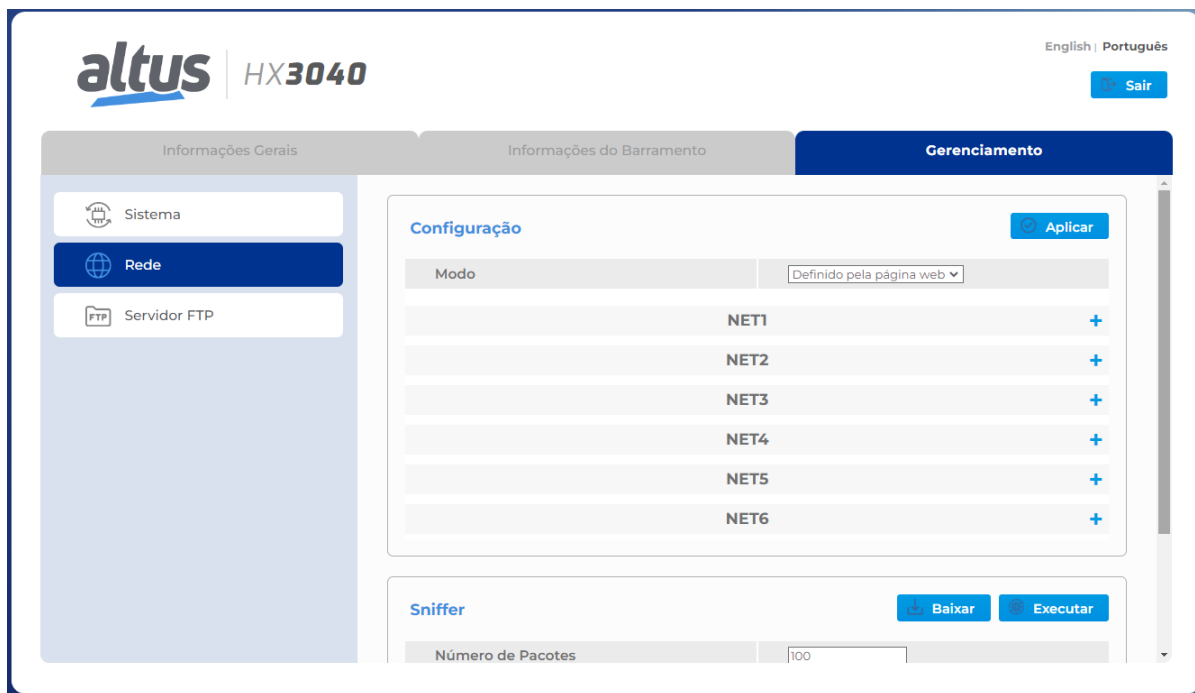


Figura 160: Página Web de Rede

4.14.2.1. Configurações de Rede

4.14.2.1.1. Modo Definido pela Aplicação

O campo Modo define qual configuração o controlador deve carregar para as suas interfaces. Este campo pode ser configurado como *Definido pela página web* ou *Definido pela aplicação*.

Quando configurado como *Definido pela aplicação*, a tabela de interfaces fica desabilitada, não permitindo alterações, conforme mostra a figura abaixo. Neste modo, as configurações aplicadas no controlador são as definidas pela aplicação.

ATENÇÃO

A tabela para configuração de rede é exibida somente quando não possui aplicação no controlador ou este não estiver em execução. Não é possível alterar as configurações de rede enquanto houver uma aplicação em execução no controlador.

Abaixo uma imagem com o modo *Definido pela aplicação* selecionado, mostrando a tabela de interfaces desabilitada.

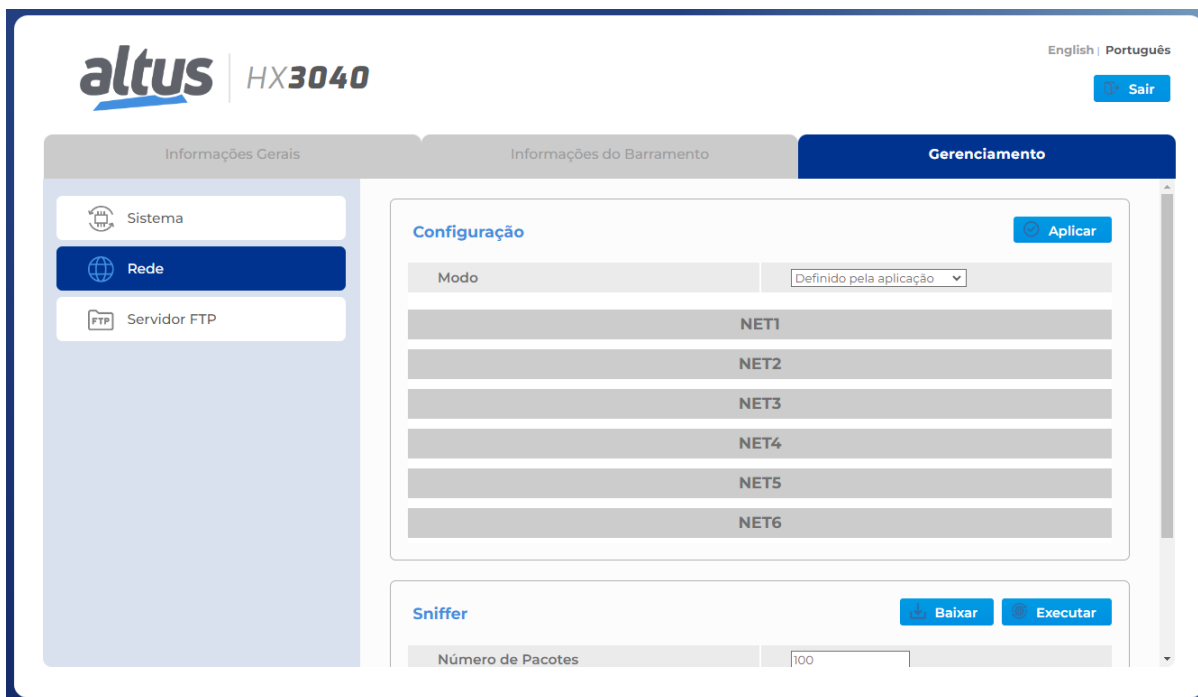


Figura 161: Tabela de Interfaces - Modo Aplicação

4.14.2.1.2. Modo Definido pela Página Web

Para o modo *Definido pela Página Web* a tabela de interfaces permanece habilitada, conforme mostra a figura abaixo.

Neste modo, o usuário pode configurar Endereço IP, Máscara de Rede e Gateway de cada uma das interfaces Ethernet disponíveis, além de poder habilitar e desabilitar todas elas (exceto a NET 1, que permanecerá sempre habilitada).

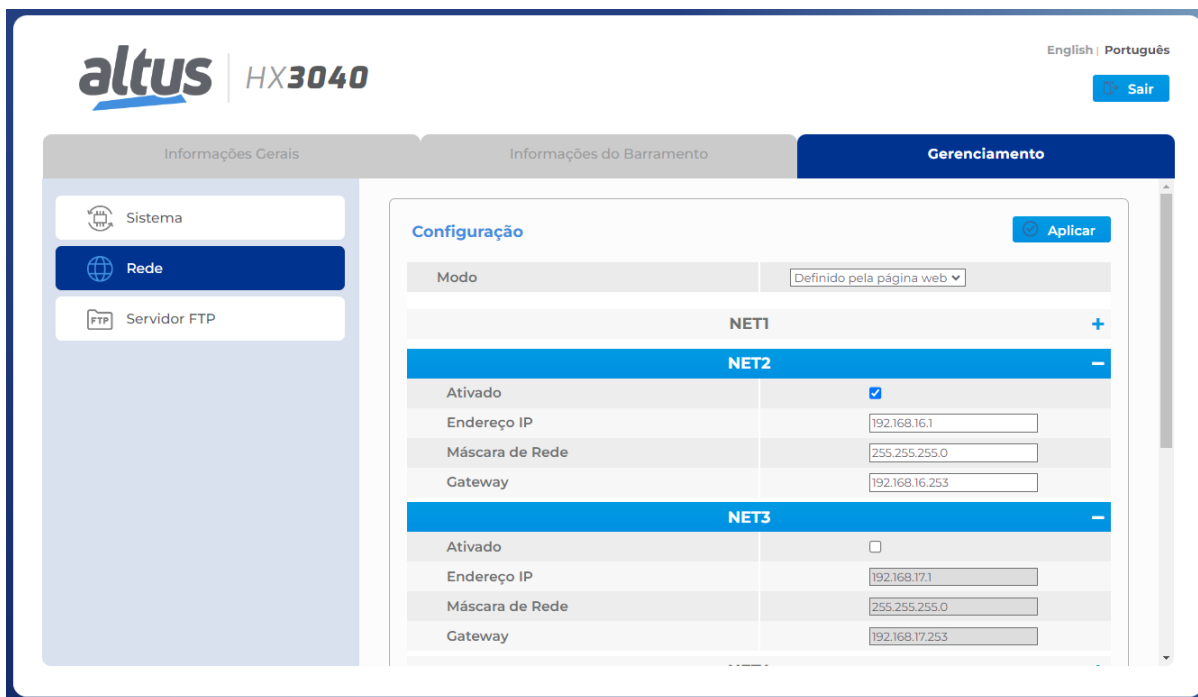


Figura 162: Tabela de Interfaces - Modo Web

O checkbox *Ativado* permite ao usuário habilitar e desabilitar as interfaces Ethernet. Este somente é disponível para ser

marcado ou desmarcado quando a configuração for *Definido pela página web*.

Quando este checkbox estiver desmarcado, indica que aquela interface NET está desabilitada, ou seja, não receberá configuração e ficará desativada, conforme pode ser visto na NET 3 da figura acima. Já quando a interface está habilitada (com o checkbox marcado), conforme a NET 2 da figura acima, as configurações ficam disponíveis para edição.

Para que as configurações sejam aplicadas no controlador, basta clicar no botão *Aplicar*. Este processo verifica se houve algum erro na configuração realizada e, se for o caso, exibe uma mensagem na tela do navegador indicando o erro. Se as configurações estiverem corretas, após clicar em *Aplicar*, será exibida uma janela, no navegador, de confirmação para aplicar as novas configurações. Clicando em *OK*, as configurações são enviadas para o controlador e aplicadas.

ATENÇÃO

Ao realizar alterações de rede no controlador as interfaces serão reiniciadas, o que pode causar queda na comunicação. Especialmente ao ser alterado o valor do endereço IP.

Ao aplicar as configurações utilizando o modo de operação *Definido pela Aplicação*, o controlador vai assumir as configurações que foram definidas pela aplicação já carregada nele anteriormente. Caso não haja nenhuma aplicação no controlador, a configuração atual será mantida, sendo alterado apenas o modo de configuração.

Utilizando o modo *Definido pela Página Web*, os endereços indicados na Seção de Rede serão carregados.

ATENÇÃO

O modo de operação *Definido pela Página Web* configura as interface para operarem em *Modo Simples*.

4.14.2.2. Sniffer de rede

O sniffer de rede, mostrado na figura abaixo, pode ser utilizado para observar o tráfego nas interfaces físicas, exceto para os dispositivos USB como modems e adaptadores wifi. Ele possui duas configurações básicas:

Número de Pacotes: É o número de pacotes que se deseja capturar. O valor configurado desse parâmetro deve estar dentro da faixa de 100 a 25000 pacotes;

Tempo Limite de Ocio (segundos): Se não houver nenhum tráfego de pacotes na interface após este tempo limite configurado, a execução do Sniffer é encerrada. Pode ser configurado com valores entre 1 e 3600 segundos.

Através da tabela *Interfaces*, o usuário pode selecionar em quais interfaces deseja executar o Sniffer, ou seja, fazer a análise da rede. Pode-se selecionar todas as interfaces disponíveis e executa-lo em todas elas, simultaneamente. Para interfaces desabilitadas, não é possível executar o Sniffer, caso a opção selecionada esteja desabilitada, será exibido um erro no navegador.

Somente após alguns instantes, depois da abertura da tela, o botão *Executar*, que inicia a execução do Sniffer, se tornará disponível. O botão *Baixar* será desbloqueado apenas se houver algum arquivo, referente ao Sniffer, disponível para ser baixado. Se o Sniffer nunca foi executado, ou o arquivo for excluído, o botão não estará disponível.

Ao executar o Sniffer de Rede, a página irá desabilitar os campos de edição, o botão *Baixar* será bloqueado e o botão *Executar*, se tornará o botão *Parar*, conforme mostra a figura abaixo.

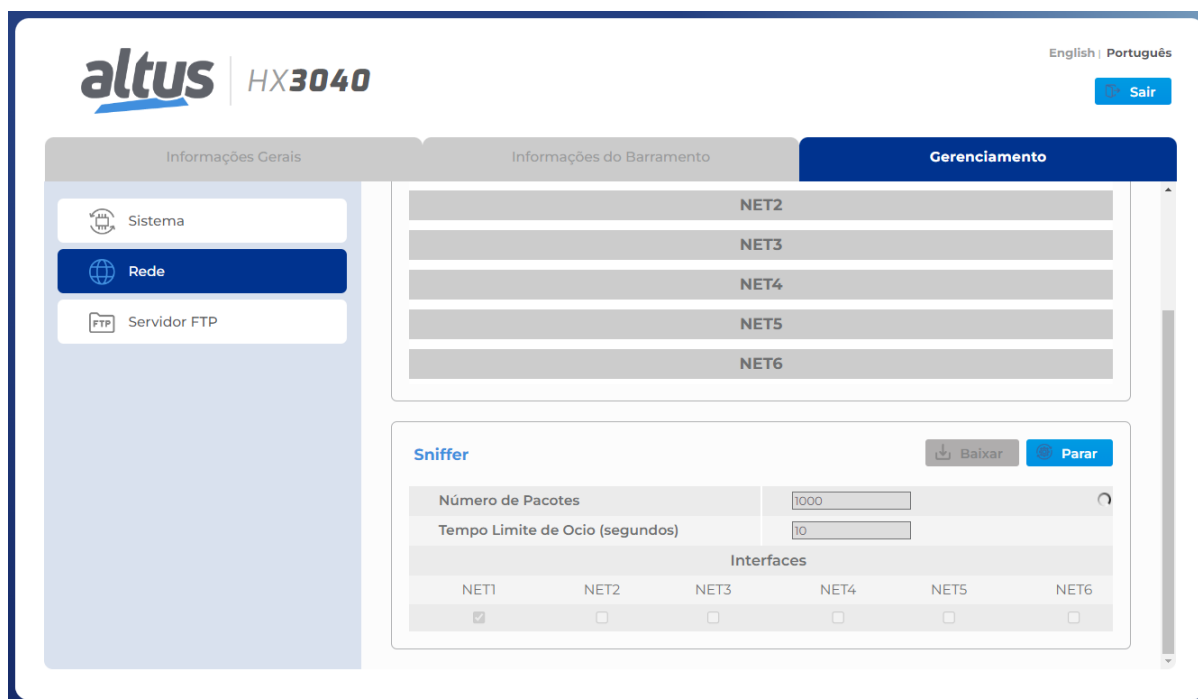


Figura 163: Sniffer de Rede Executando

O botão *Parar* pode ser utilizado para encerrar a execução do sniffer a qualquer momento após ter sido iniciada.

Para cada uma das interfaces em que o Sniffer é executado, ele gera um arquivo **.pcap**. Estes arquivos são nomeados de acordo com o nome do controlador e a interface que foi analisada, por exemplo, **HX3040_NET1.pcap**. Estes arquivos são encontrados dentro de um **.zip**, também nomeado de acordo com o nome do controlador, por exemplo, **HX3040_capture.zip**.

Ao final da execução do sniffer, é exibida uma mensagem questionando se deve ser realizado ou não o download automático dos arquivos gerados. Estes arquivos são armazenados na pasta *InternalMemory* da [Memória de Arquivos de Usuário](#) e podem ser acessados através do software de programação do controlador. O arquivo baixado é sempre o com a extensão **.zip**, que agrupa os demais.

Caso ocorra algum problema em relação a memória insuficiente, ocasionado em função da geração dos arquivos sniffer, será indicado para o usuário. Recomenda-se então executar o analisador outra vez, com uma configuração de *Número de Pacotes* menor.

O sniffer de rede pode encerrar a sua execução por três razões, falta de memória, estouro do tempo limite de ociosidade das interfaces e também o cancelamento manual.

4.15. Servidor FTP

O FTP (File Transfer Protocol) é um protocolo que permite a transferência de arquivos entre dispositivos. Opera no modelo cliente-servidor, onde a UCP torna-se um Servidor FTP, armazenando arquivos para que Clientes FTP possam acessá-los para transferência, isto é, download e upload.

A conexão FTP de controle é uma conexão TCP, estabelecida através da porta 21 do Servidor FTP. Onde o Cliente e o Servidor trocam comandos e repostas para o gerenciamento da sessão de transferência de arquivos.

Através do protocolo FTP, o Cliente FTP consegue ler e escrever arquivos que estão, tanto armazenados na memória interna da UCP, como também nas memórias externas (cartão de memória, por exemplo), caso presentes na arquitetura. O tamanho máximo do arquivo que pode ser transferido varia de acordo com a quantidade de memória disponível, entre memória interna e memórias externas. Ao atingir o limite de memória, a transferência irá parar, e caso o arquivo não tenha sido transferido por completo, o mesmo ficará corrompido.

ATENÇÃO

Download e upload de grandes arquivos via FTP, tanto da memória interna, quanto das memórias externas, podem afetar consideravelmente na performance da UCP.

4.15.1. Configuração

A configuração do Servidor FTP é feita através de uma seção dedicada, localizada na aba *Gerenciamento* da Página Web de Sistema do controlador, conforme mostra a figura abaixo.

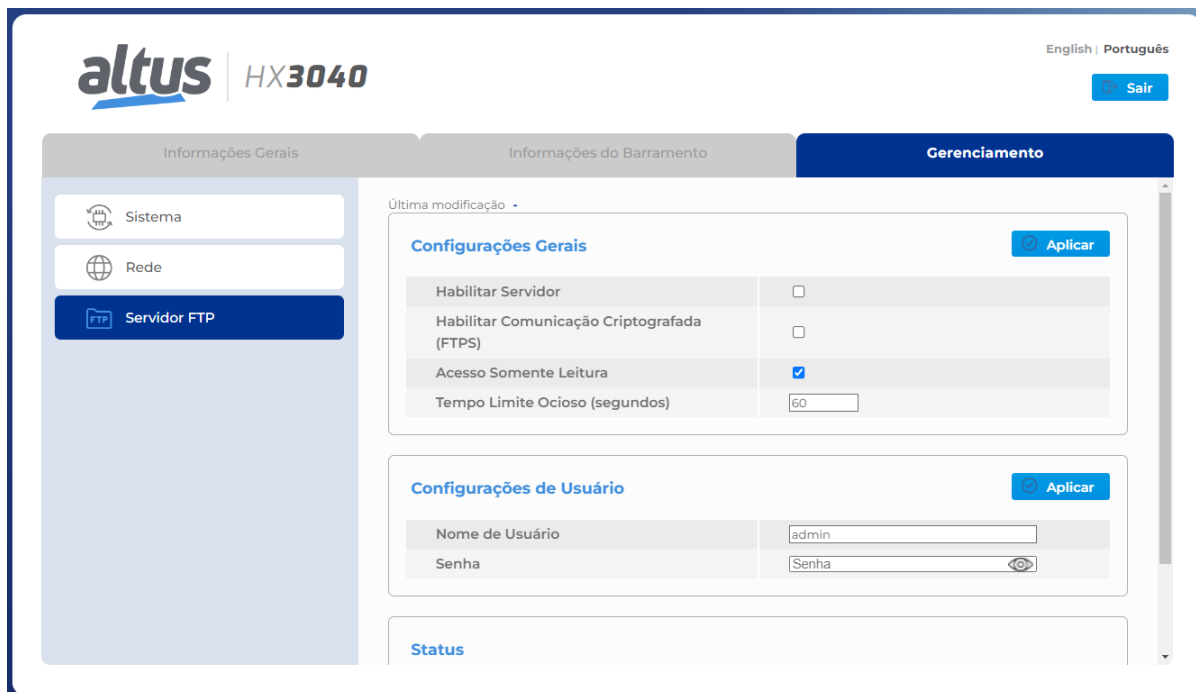


Figura 164: Tela de configuração do Servidor FTP

O FTP trata-se de uma funcionalidade à parte ao MasterTool IEC XE, isto é, não necessita de interação alguma com a ferramenta de programação. As configurações aplicadas na Página Web de Sistema passam a valer quando confirmadas através do botão *Aplicar* e, são salvas automaticamente no controlador. Contanto que a funcionalidade esteja habilitada, voltará a operar mesmo após a reinicialização do dispositivo.

4.15.1.1. Configurações Gerais

4.15.1.1.1. Habilitar Servidor

Permite habilitar e desabilitar a funcionalidade. Quando o Servidor FTP está habilitado, o serviço torna-se disponível e as demais configurações realizadas são aplicadas. Caso o Servidor FTP esteja desabilitado, apesar de o serviço estar indisponível, suas configurações são armazenadas. Por padrão, o Servidor FTP é desabilitado.

4.15.1.1.2. Habilitar Comunicação Criptografada (FTPS)

Permite habilitar e desabilitar a comunicação criptografada. Essa comunicação é feita através do *FTPS Explícito*, também conhecido como *FTPES*. Trata-se de uma extensão segura do protocolo FTP, que adiciona uma camada de criptografia à transferência.

Neste tipo de comunicação, quando estabelecida uma conexão cliente-servidor, em vez de iniciar imediatamente a transferência de dados, o Cliente FTP envia um comando *AUTH SSL*, para solicitar a conexão segura. Então, ao habilitar a comunicação criptografada, a UCP gera um certificado *self-signed* para garantir a comunicação utilizando o protocolo SSL (Secure Sockets Layer).

Quando o Cliente FTP realizar a conexão autenticada, o certificado e suas respectivas informações serão exibidos, permitindo estabelecer a comunicação FTPS Explícita (FTP sobre SSL), através da mesma porta 21.

ATENÇÃO

O certificado FTPES tem uma validade de um ano a partir da data de criação. Para gerar um novo certificado, basta aplicar novamente a configuração.

4.15.1.1.3. Acesso Somente Leitura

Permite habilitar e desabilitar a limitação do acesso de escrita e leitura no Servidor. Por padrão, o parâmetro é habilitado.

Quando o parâmetro está habilitado, o Cliente FTP tem acesso somente a leitura, sem a possibilidade de adicionar ou remover qualquer arquivo do Servidor FTP. Neste caso, a única operação permitida é o upload de arquivos, ou seja, transferência do Servidor para o Cliente. Já quando o parâmetro está desabilitado, é possível realizar todas as operações.

4.15.1.1.4. Tempo Limite Ocioso (Segundos)

Define o tempo máximo em que a conexão será mantida antes do Servidor FTP fecha-la por inatividade. Isto é, após aberta a conexão com um Cliente FTP, se não houver nenhuma atividade, após passado o Tempo Limite Ocioso, a conexão com o Cliente é fechada pelo Servidor.

O parâmetro pode ser configurado com valores entre 10 e 60 segundos sendo, por padrão, 60 segundos.

4.15.1.2. Configurações de Usuário

4.15.1.2.1. Nome de Usuário

Usuário ou *Nome de Usuário* solicitado para que o Cliente realize a conexão com o Servidor.

Caso seja realizada uma nova configuração, a anterior é removida. Isto é, há apenas um único usuário FTP, porém, este pode ser utilizado em múltiplas conexões.

4.15.1.2.2. Senha

Gerencia a chave de autenticação para que o Cliente FTP possa se conectar com o Servidor FTP.

Não há recuperação de senha. Caso esta tenha sido perdida, é necessário adicionar uma nova configuração de usuário.

Item Configurável	Tamanho Mínimo	Tamanho Máximo	Caracteres Permitidos	Configuração Padrão
Usuário	4	30	[a-z][A-Z][0-9]@\$*_.	admin
Senha	4	30	[a-z][A-Z][0-9]@\$*_.	admin

Tabela 208: Configurações de Usuário FTP

4.15.1.3. Status

4.15.1.3.1. Estado Atual

Exibe o estado atual do Servidor FTP. Os possíveis estados são: "*Executando*", "*Não Executando*" e "*Reiniciando Serviço*". Cada configuração aplicada reiniciará o serviço.

4.15.1.3.2. Conexões Ativas

Mostra ao usuário o número atual de conexões ativas.

O Servidor FTP aceita no máximo duas conexões ativas simultaneamente. Alguns Clientes FTP, como por exemplo o *Filezilla*, utilizam um recurso chamado Transferência de Arquivos Multithread, para melhorar a eficiência e a velocidade da transferência. Este recurso, permite que o Cliente FTP abra mais de uma conexão para a transferência de um arquivo. Consequentemente, quando utilizado um Cliente FTP deste tipo, apenas um Cliente já soma duas conexões ativas no Servidor.

Já outros Clientes FTP como por exemplo através de terminal de comandos, ocupam apenas uma conexão ativa, assim possibilitando a conexão simultânea de dois Clientes FTP com o Servidor.

5. Programação Inicial

O objetivo deste capítulo é auxiliar na programação e configuração da UCP da Série Hadron Xtorm, para que o usuário de os primeiros passos antes de iniciar a programação de uma UCP.

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.

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.

ATENÇÃO

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.

5.1. 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 (Ex.: %QB0 sempre será mais significativo que o %QB1, como na tabela abaixo, que utiliza a palavra UCPHADRON XTORM, onde a letra U é o octeto 0 e a letra O é o octeto 7).

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. Organização little e big endian estão na tabela abaixo.

MSB ← Little-endian → LSB								
BYTE	%QB7	%QB6	%QB5	%QB4	%QB3	%QB2	%QB1	%QB0
	U	C	P	X	T	O	R	M
WORD	%QW3		%QW2		%QW1		%QW0	
	UC		PX		TO		RM	
DWORD	%QD1				%QD0			
	UCPX				TORM			
LWORD	%QL0							
	UCPXTORM							
MSB ← Big-endian → LSB								
BYTE	%QB0	%QB1	%QB2	%QB3	%QB4	%QB5	%QB6	%QB7
	U	C	P	X	T	O	R	M
WORD	%QW0		%QW2		%QW4		%QW6	
	UC		PX		TO		RM	
DWORD	%QD0				%QD4			
	UCPX				TORM			
LWORD	%QL0							
	UCPXTORM							

Tabela 209: Exemplo Organização e Acesso à Memória

SIGNIFICÂNCIA					SOBREPOSIÇÃO					
Bit	Byte	Word	DWord	LWord	Byte	Word	DWord			
%QX0.7	%QB 00	%QW			%QB00	%QW				
%QX0.6										
%QX0.5										
%QX0.4										
%QX0.3										
%QX0.2										
%QX0.1										
%QX0.0										
%QX1.7	%QB 01	00			%QB01	%QW				
%QX1.6										
%QX1.5										
%QX1.4										
%QX1.3										
%QX1.2										
%QX1.1										
%QX1.0										
%QX2.7	%QB 02	%QW	00		%QB02	%QW	01		00	
%QX2.6										
%QX2.5										
%QX2.4										
%QX2.3										
%QX2.2										
%QX2.1										
%QX2.0										
%QX3.7	%QB 03	02			%QB03	%QW			01	
%QX3.6										
%QX3.5										
%QX3.4										
%QX3.3										
%QX3.2										
%QX3.1										
%QX3.0										
%QX4.7	%QB 04	%QW	00	%QL	%QB04	%QW	03			02
%QX4.6										
%QX4.5										
%QX4.4										
%QX4.3										
%QX4.2										
%QX4.1										
%QX4.0										
%QX5.7	%QB 05	04			%QB05	%QW	04			03
%QX5.6										
%QX5.5										
%QX5.4										
%QX5.3										
%QX5.2										
%QX5.1										
%QX5.0										
%QX6.7	%QB 06	%QW	04		%QB06	%QW	05			04
%QX6.6										
%QX6.5										
%QX6.4										
%QX6.3										
%QX6.2										
%QX6.1										
%QX6.0										
%QX7.7	%QB 07	06			%QB07		06			
%QX7.6										
%QX7.5										
%QX7.4										
%QX7.3										
%QX7.2										
%QX7.1										
%QX7.0										

Tabela 210: Organização e Acesso à Memória

A tabela acima mostra a organização e acesso à memória, exemplificando a significância dos bytes (Significance) e a disposição dos demais tipos de variável, inclusive a sobreposição (Overlapping):

5.2. 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:

- O MasterTool Xtorm somente permite a criação de projetos baseados em um template, selecionando ao mesmo tempo o perfil a ser utilizado.
- 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 as 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.

ATENÇÃO

No decorrer dos perfis de projeto são nomeados alguns tipos de tarefas, as quais estão descritas na seção [Configurando as Tarefas](#).

5.2.1. 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. Para mais informações sobre alarmes e conversão de engenharia, consulte a seção [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 novo 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).

Tarefa	POU	Prioridade	Tipo	Padrão	Opções	Evento
MainTask	MainPrg	13	Cíclica	20 ms	5 a 100 ms	-
ProtTask	ProtPrg	1	Cíclica	4 ms	4 a 20 ms	-

Tabela 211: Tarefas no Perfil para UTR

5.2.2. 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 consequentemente 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, a mesma 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.

Tarefa	POU	Prioridade	Tipo	Padrão	Evento
MainTask	MainPrg	13	Cíclica	20 ms	-
CyclicTask00	CyclicPrg00	13	Cíclica	200 ms	-
CyclicTask01	CyclicPrg01	13	Cíclica	500 ms	-
ExternInterruptTask00	ExternInterruptPrg00	02	Externa	-	COM1_CTS
TimeInterruptTask00	TimeInterruptPrg00	01	Cíclica	20 ms	-
ExternInterruptTask01	ExternInterruptPrg01	11	Externa	-	COM1_DCD
TimeInterruptTask01	TimeInterruptPrg01	09	Cíclica	30 ms	-
FreeTask	FreePrg	31	Contínua	-	-

Tabela 212: Tarefas no Perfil Personalizado

ATENÇÃO

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

5.3. 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 novo projeto no MasterTool Xtorm a partir do menu *Arquivo* e logo em seguida, “*New Project...*”, conforme mostra a figura abaixo.

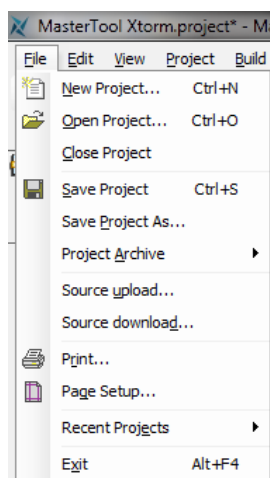


Figura 165: Novo Projeto

Após, uma janela será apresentada ao usuário, solicitando que o mesmo 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.

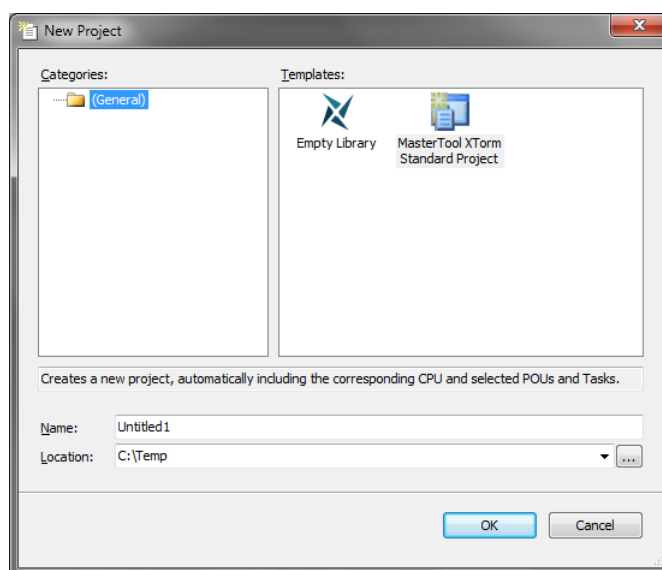


Figura 166: Classificação do Projeto

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).

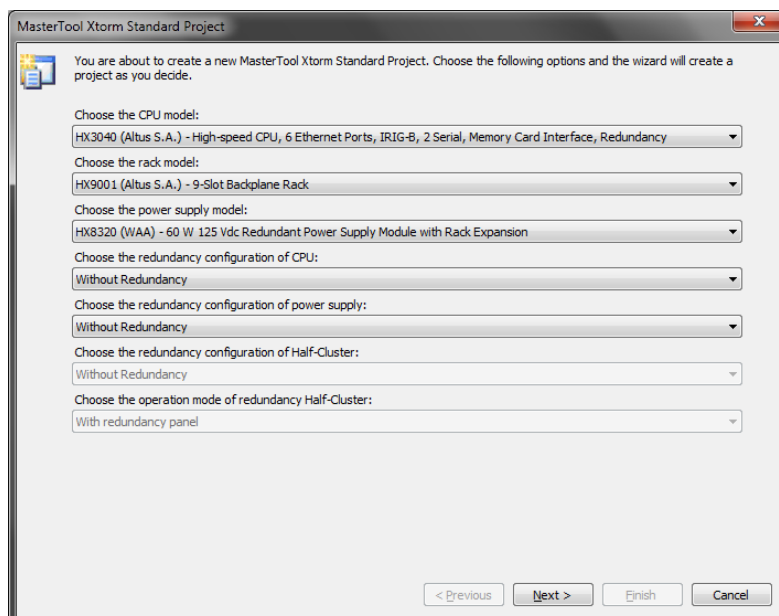


Figura 167: Opções do Projeto

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 abaixo podemos visualizar uma tela com a opção de redundância de UCP selecionada.

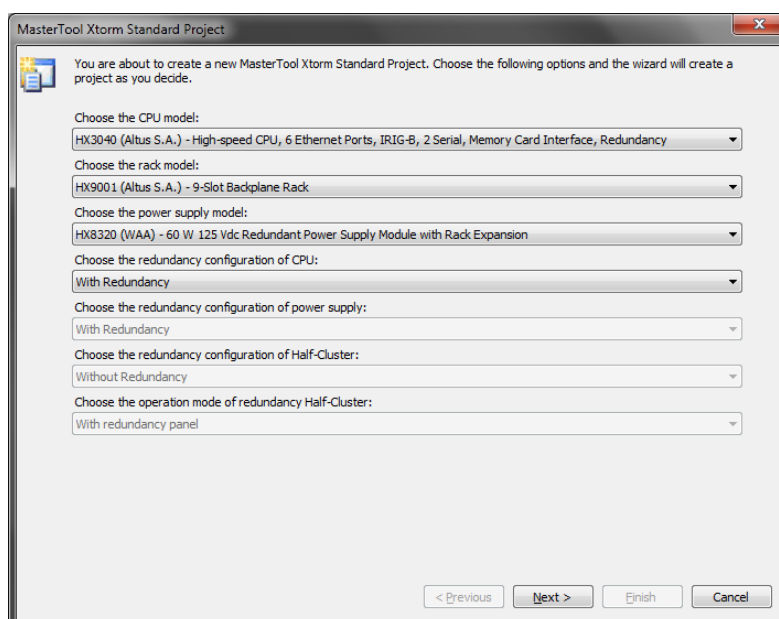


Figura 168: Opções de Projeto com Redundância de UCP Habilitada

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.

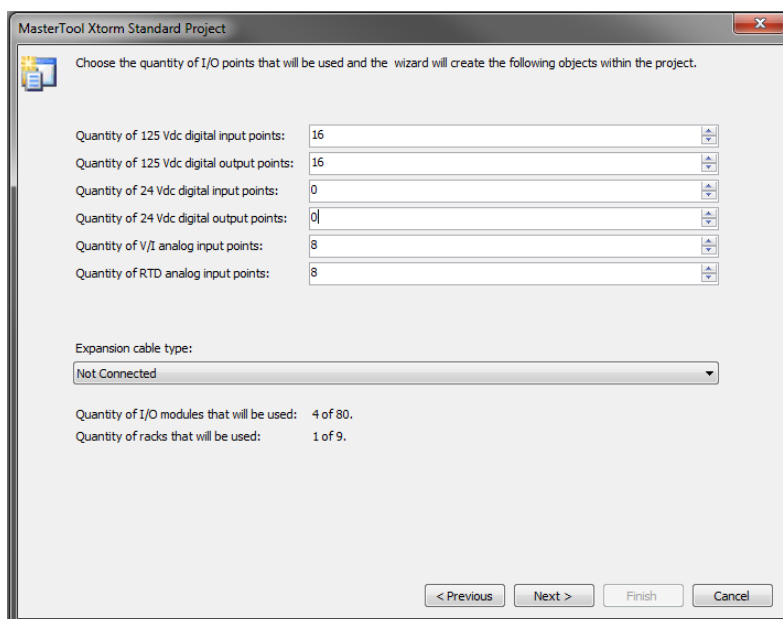


Figura 169: Opções de Módulos de E/S do Projeto

Então, o usuário deverá selecionar o perfil para o projeto, e a linguagem padrão para as POU (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.

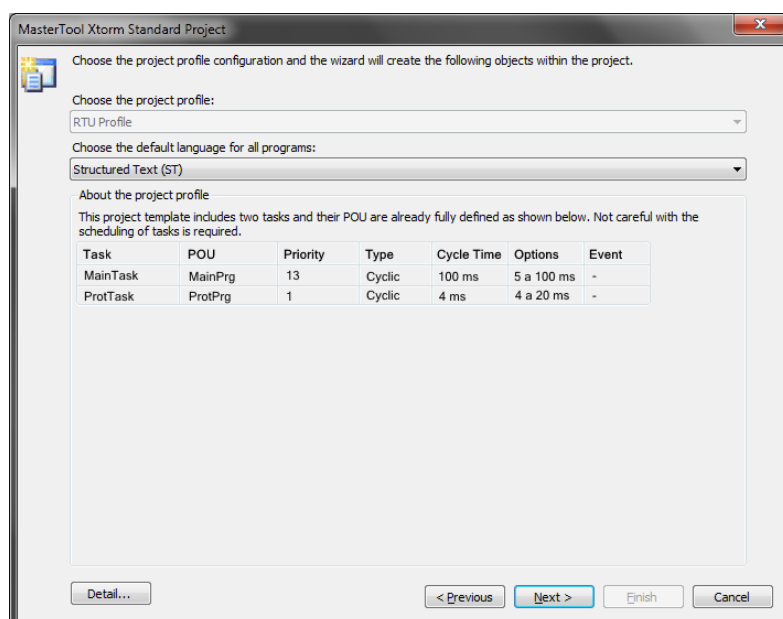


Figura 170: Seleção das Características de Perfil do Projeto

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

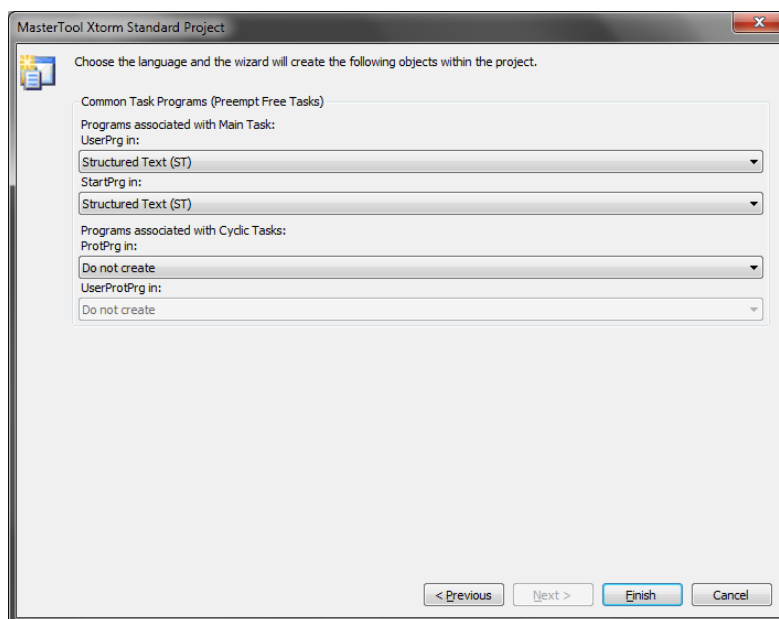


Figura 171: Linguagem de Programação

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

5.4. 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 suporta até 100 módulos de entrada/saída, estes distribuídos em até 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 abaixo:

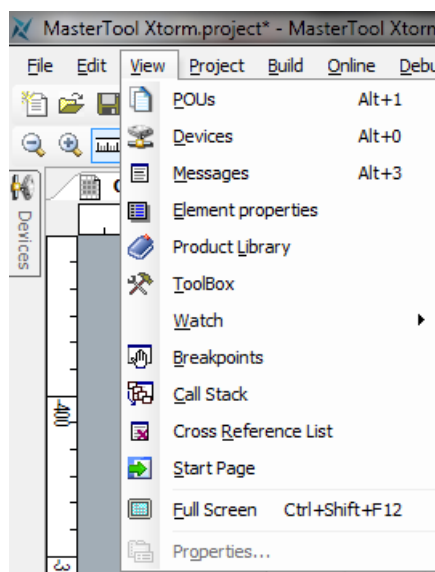


Figura 172: Visualizando a Biblioteca

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.

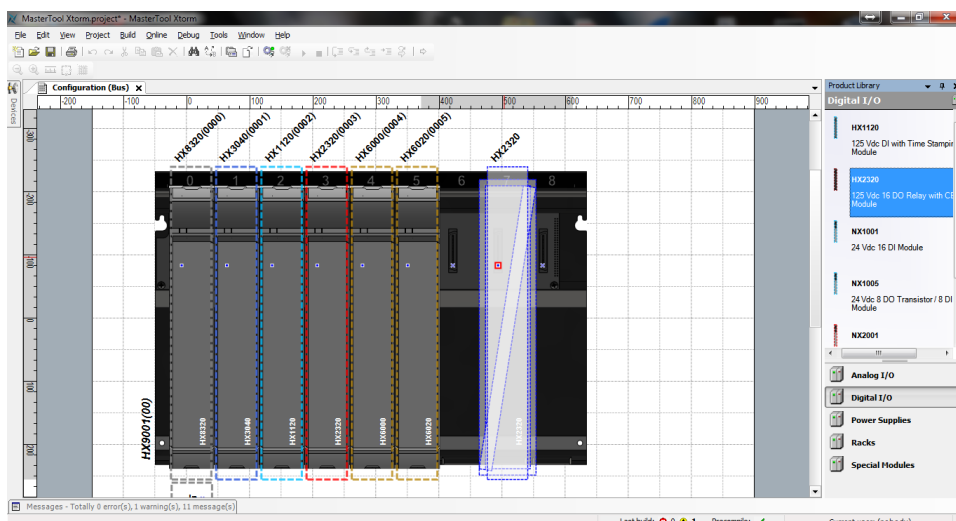


Figura 173: Acrescentando Módulos

5.5. Criando POUs

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

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

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

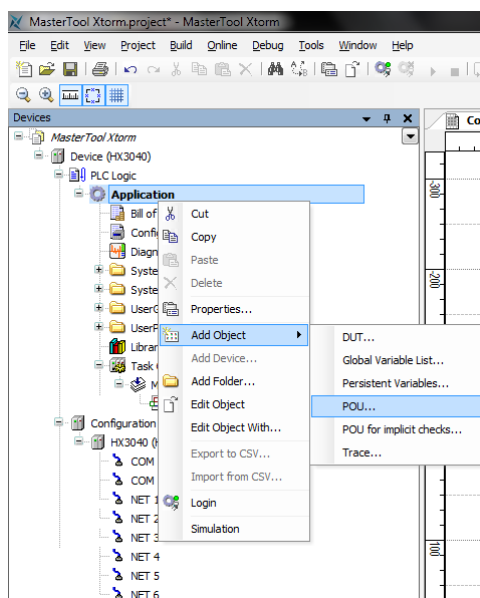


Figura 174: Inserindo POUs

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*.

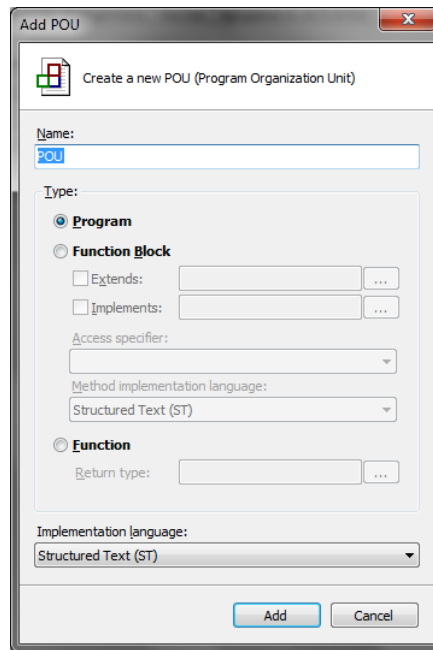


Figura 175: Classificando a POU

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 POU's criadas automaticamente pelo perfil do projeto.

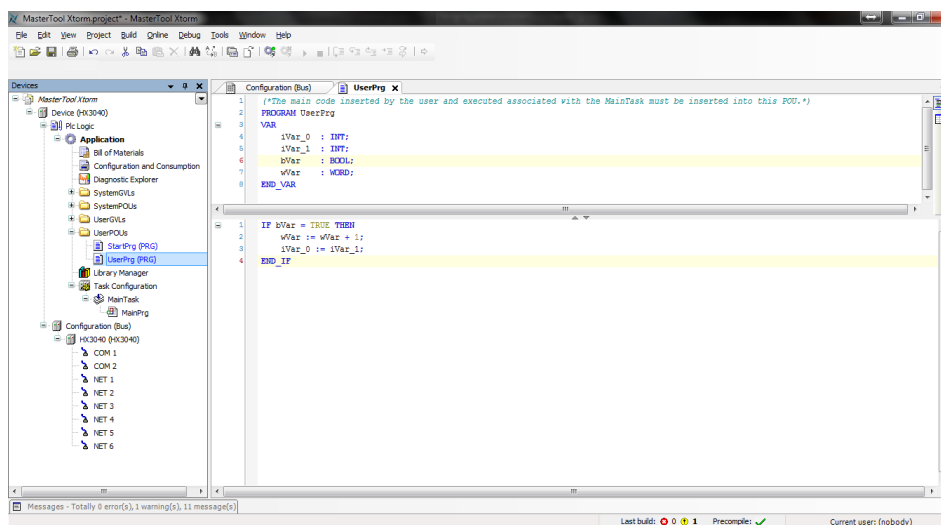


Figura 176: Editando a POU

5.6. Criando 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 abaixo.

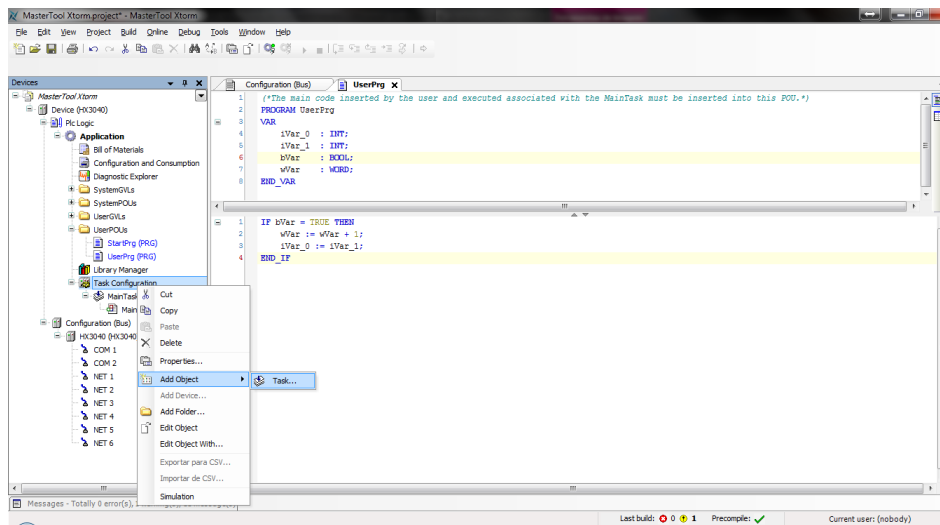


Figura 177: Criando uma Tarefa

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

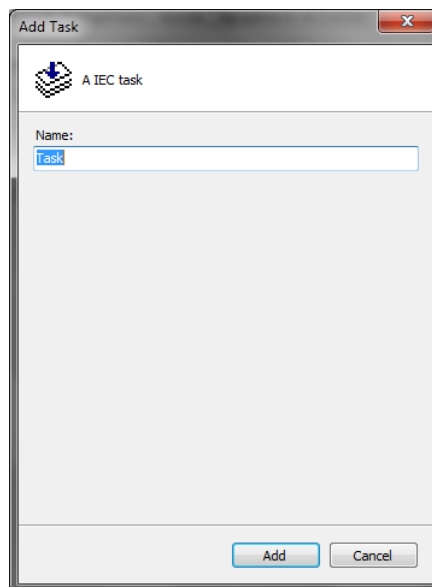


Figura 178: Nomeando a Tarefa

5.6.1. Configurando as Tarefas

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

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 *Sensitivity* 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.

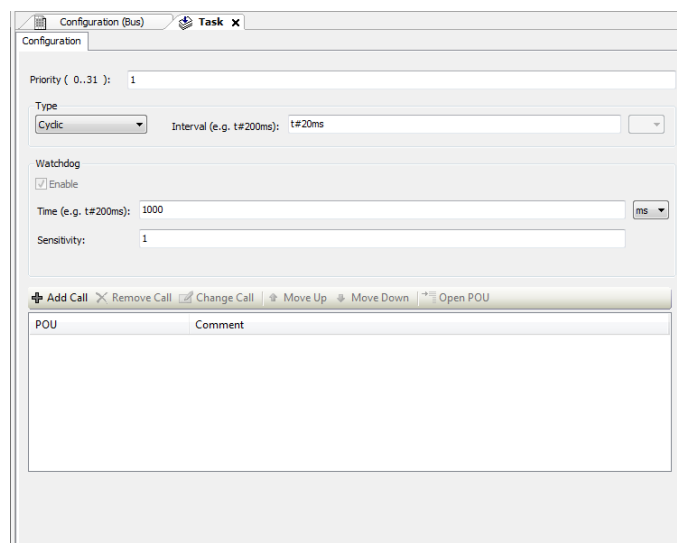


Figura 179: Configurando a Tarefa Criada

A tabela abaixo exibe as verificações realizadas pelo *MasterTool Xtorm*, sendo o campo *Sensitivity* igual a 1. Para o Perfil Personalizado não é feita consistência no intervalo da tarefa e no tempo de *cão-de-guarda*.

Tarefa	Tipo	Intervalo Mínimo	Intervalo Máximo
MainTask	Cíclica	5 ms	750 ms
CyclicTask	Cíclica	5 ms	2147483 ms
TimeInterruptTask	Cíclica	500 μ s	2147483 ms

Tabela 213: Configurações Máximas Permitidas

5.6.2. Vínculo de POU 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 POU já estão associadas as suas respectivas tarefas. Porém, caso utilizado o perfil Personalizado ou novas POU criadas, as mesmas devem ser vinculadas a tarefas.

Para associar uma POU criada, basta acessar a tarefa desejada, clicando duas vezes sobre a mesma na árvore de dispositivos, e 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 abaixo.

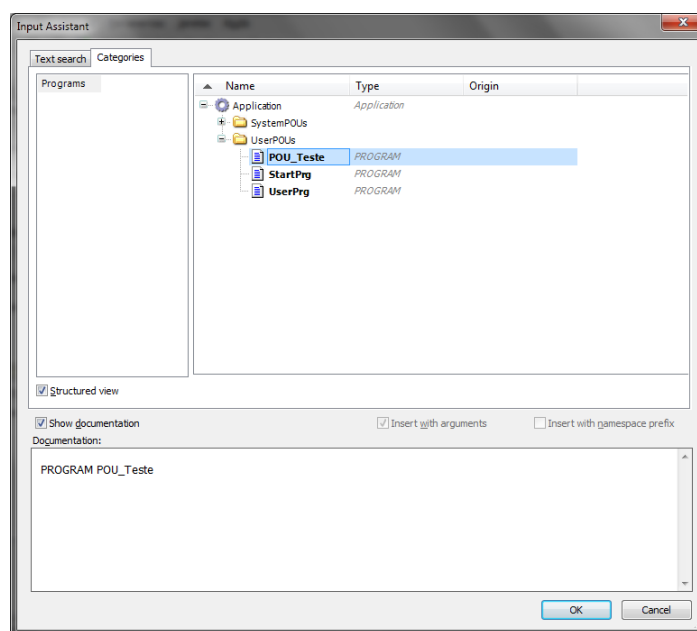


Figura 180: Vinculando POU às Tarefas

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();
```

5.6.3. 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.

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

	Tipo de Tarefa	HX3040	
		UTR	Personalizado
Configuration Task (Tarefa WHSB)	Cíclica	1	1
Tarefas de Usuário	Cíclica	1	15
	Disparada por Evento	0	15
	Disp. por Evento Externo	0	15
	Contínua	0	15
	Disparada por Estado	0	15
NETs – Instâncias Cliente ou Servidor	Cíclica	4	
COM (n) – Instâncias Mestre ou Escravo	Cíclica	1	
TOTAL		16	32

Tabela 214: Número Máximo de Tarefas IEC

Notas:

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

Tarefa WHSB: A tarefa WHSB que é uma tarefa de sistema deve ser considerada para que não seja ultrapassado o valor total. Para perfil UTR utilizando redundância pode se optar por criar também a tarefa ProfTask, totalizando assim até 2 tarefas cíclicas.

NET (n) - Instâncias Cliente ou Servidor: O "n" representa o número da interface Ethernet. Valor máximo definido por interface Ethernet do sistema, ou seja, inclui os módulos de expansão, quando estes são aplicados. Exemplos para esse tipo de tarefa são as instâncias do Protocolo MODBUS.

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

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

5.7. 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 abaixo, e configurar os campos conforme descrito na seção [Configuração da UCP](#).

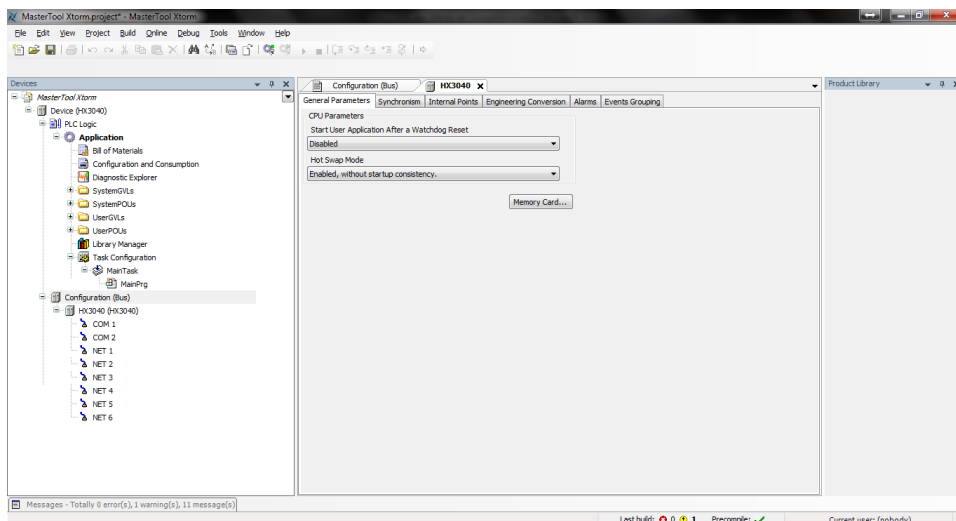


Figura 181: Configuração da UCP

Além disso, para que seja possível a comunicação entre a UCP e o MasterTool Xtorm, a interface Ethernet NET 1 deverá ser configurada, conforme a seção [Configuração das Interfaces Ethernet](#). 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.

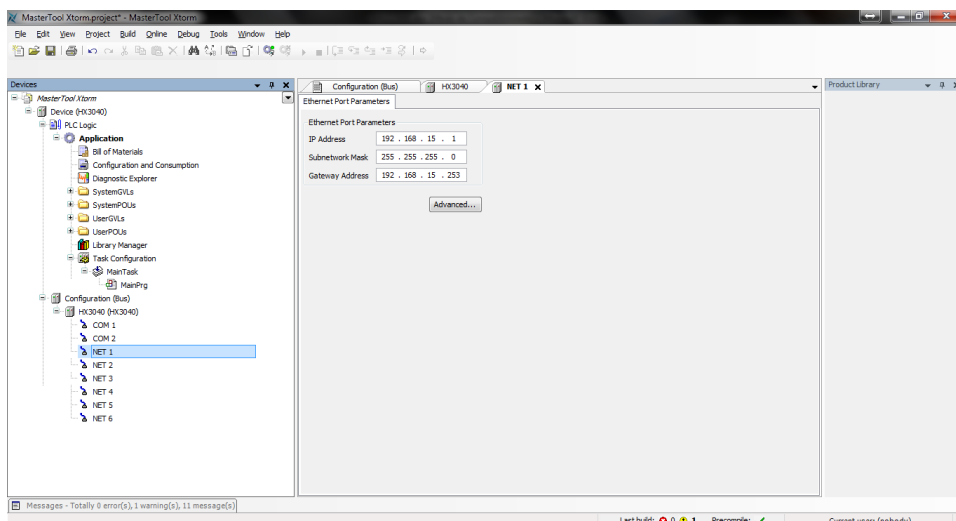


Figura 182: Configurando a Porta de Comunicação da UCP

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.

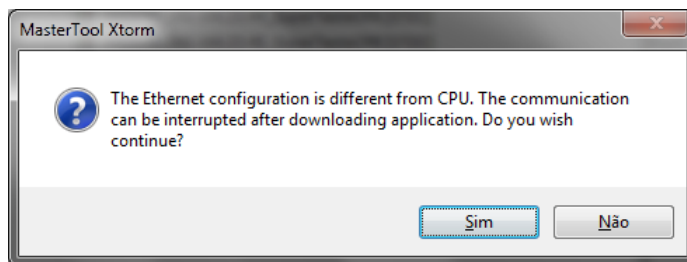


Figura 183: Aviso de Configuração IP

5.8. 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 abaixo:

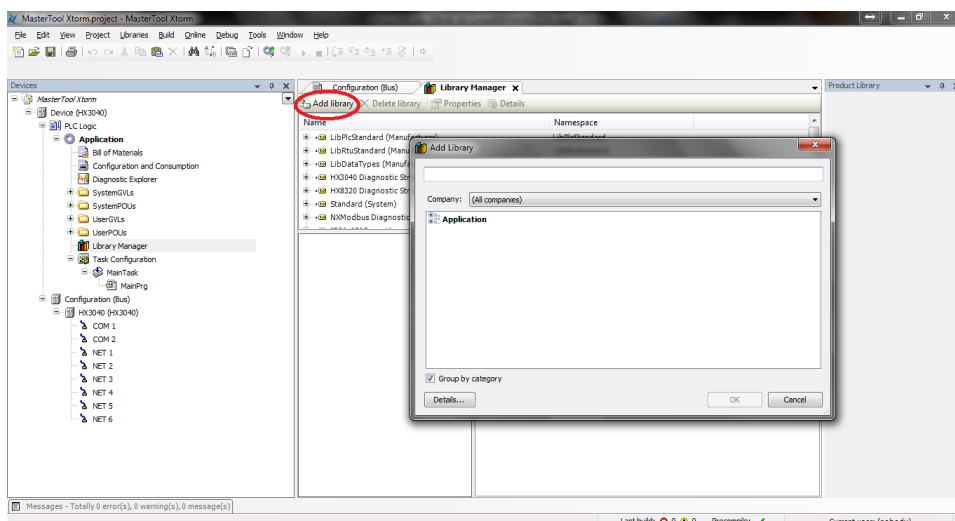


Figura 184: Inclusão de uma Biblioteca no Projeto

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

5.9. Inserindo uma Instância de Protocolo

A própria UCP da Série Hadron Xtorm, conforme a seção [Características Gerais](#), disponibiliza protocolos, como o MODBUS. Basta adicionar e configurar (conforme a seção [Configuração de Protocolos](#)) a instância do protocolo desejado na interface de comunicação.

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.

5.9.1. MODBUS RTU

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

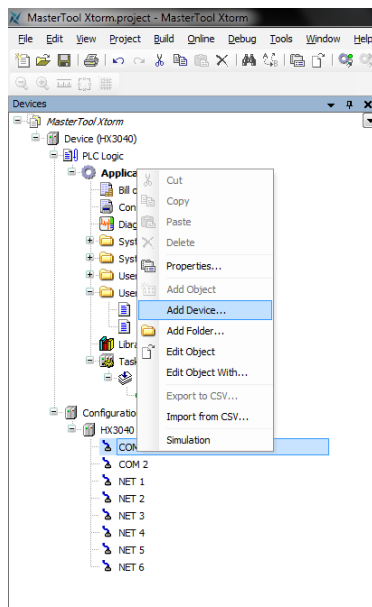


Figura 185: Adicionando a Instância MODBUS RTU

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 abaixo:

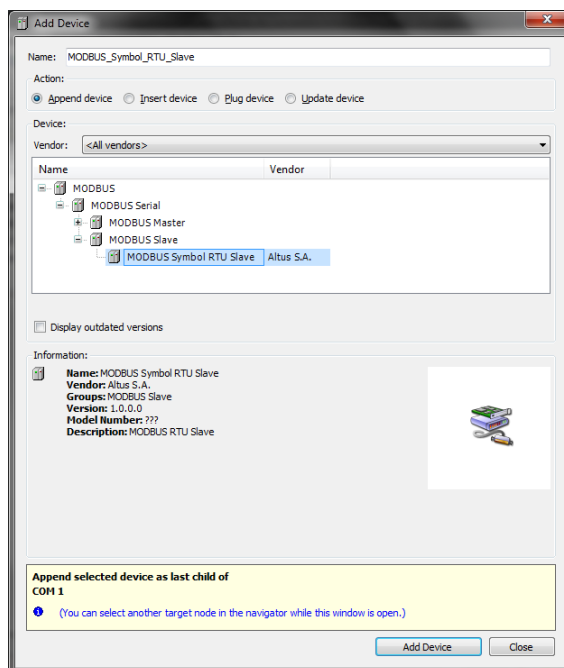


Figura 186: Selecionando o Protocolo MODBUS RTU

5.9.2. 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 abaixo.

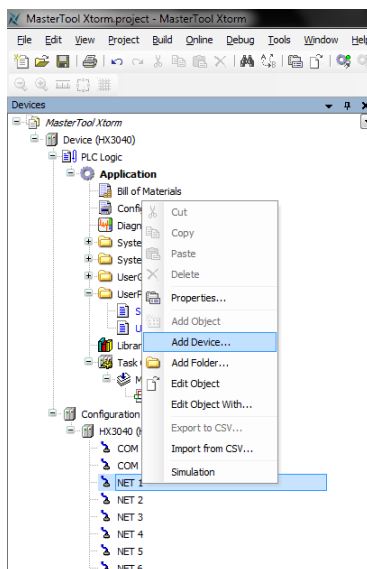


Figura 187: Adicionando a Instância MODBUS Ethernet

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 abaixo:

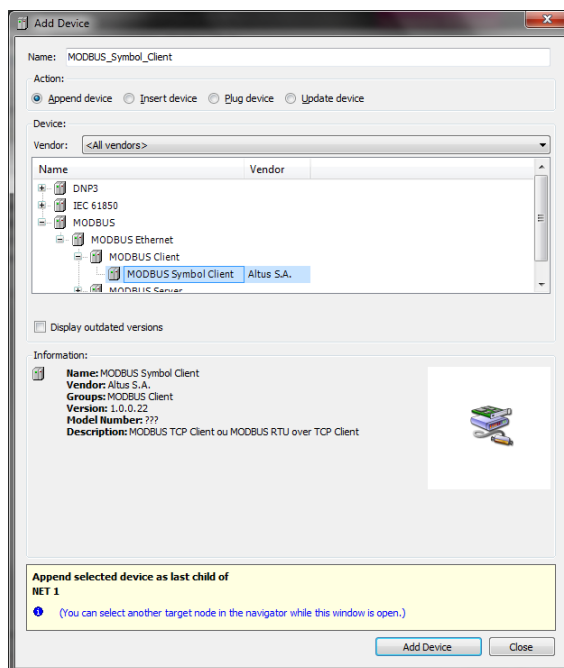


Figura 188: Selecionando o Protocolo MODBUS Ethernet

5.9.3. 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 abaixo:

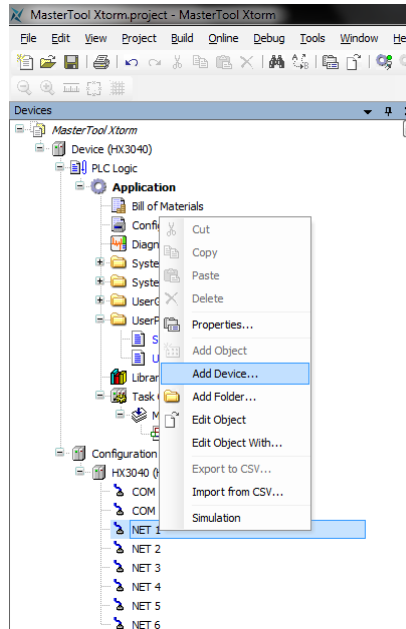


Figura 189: Adicionando a Instância do DNP3 Servidor

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 abaixo:

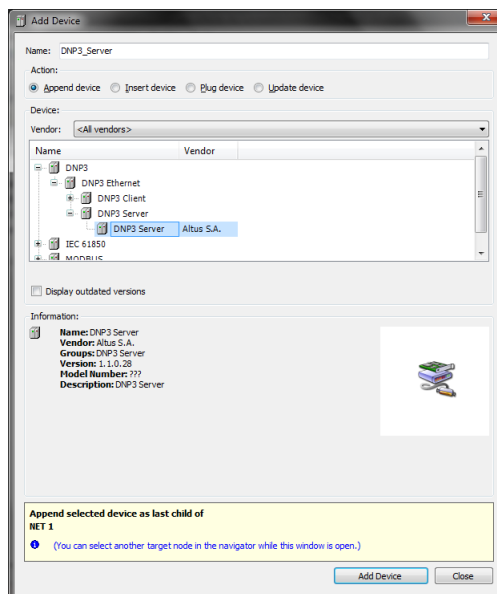


Figura 190: Selecionando o Protocolo DNP3 Servidor

5.9.4. 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 abaixo:

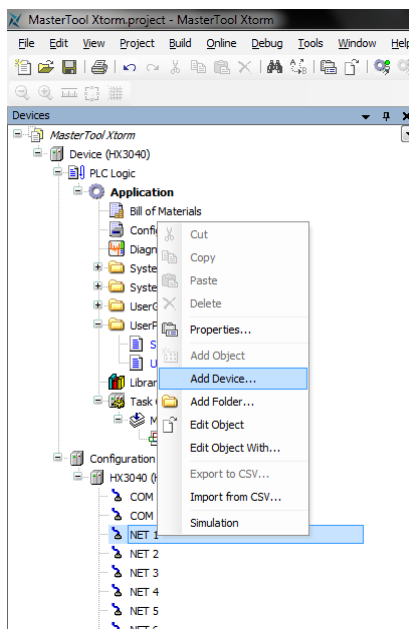


Figura 191: Adicionando a Instância do IEC 60870-5-104 Servidor

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 abaixo:

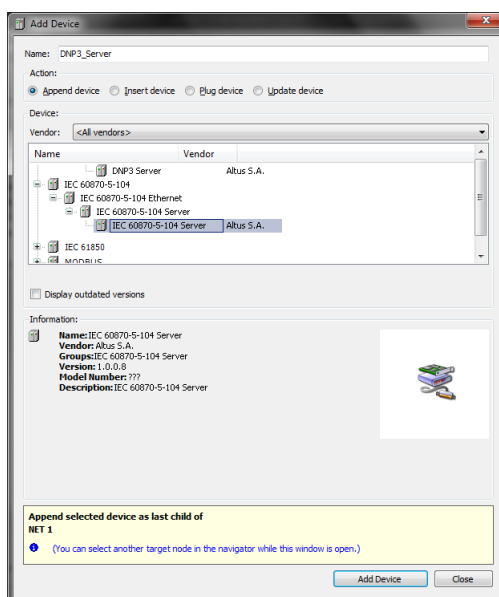


Figura 192: Selecionando o Protocolo IEC 60870-5-104 Servidor

5.10. 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.

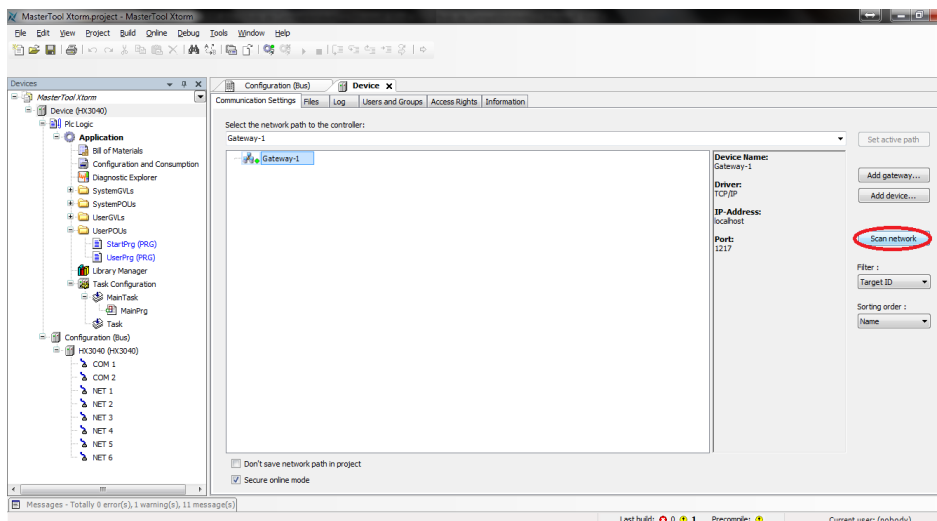


Figura 193: Localizando a UCP

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.

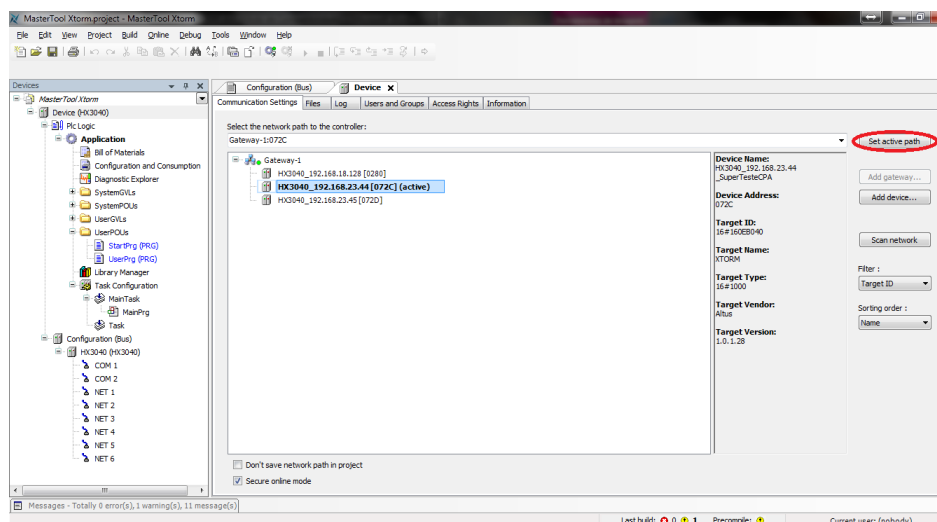


Figura 194: Selecionando o caminho ativo

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.

5.11. 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 abaixo.

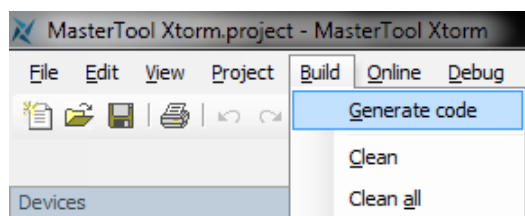


Figura 195: Compilando o Projeto

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 mostra a figura abaixo.

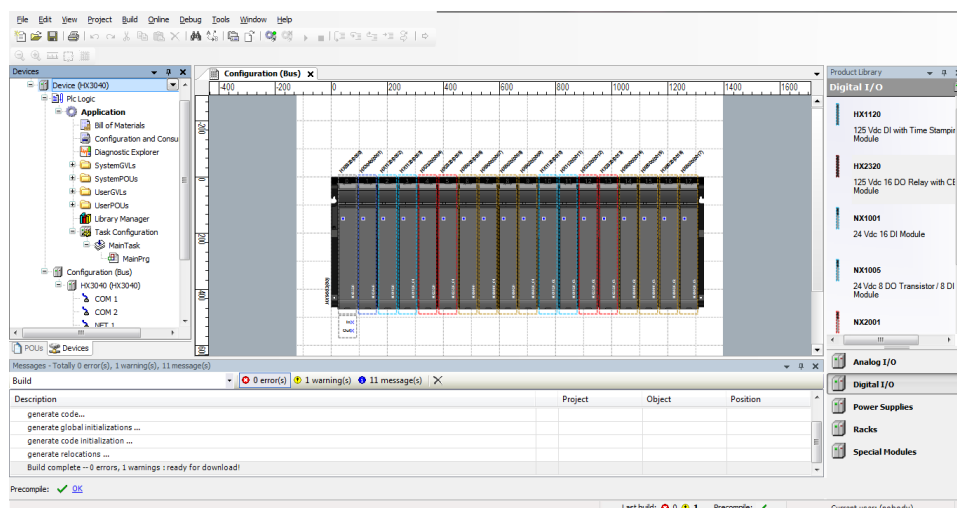


Figura 196: Mensagens da Compilação

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

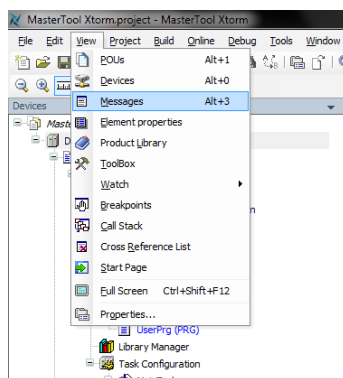


Figura 197: Incluindo as Mensagens na Tela

5.12. 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 abaixo.

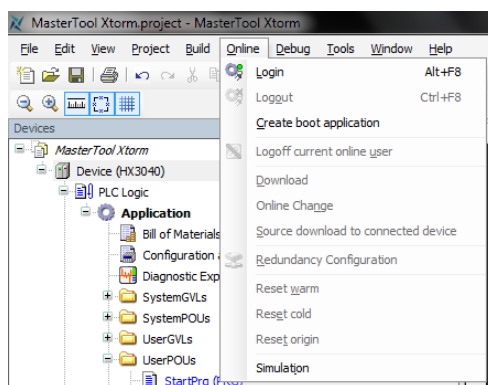


Figura 198: Enviando o Projeto para a UCP

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 abaixo 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:** Executar o Login e enviar o novo projeto sem que a aplicação atual da UCP seja parada (ver seção **Modo Run**), atualizando as alterações quando um novo ciclo é executado.
- **Login with Download:** Executar o Login e enviar o novo projeto com a UCP parada (ver seção **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.

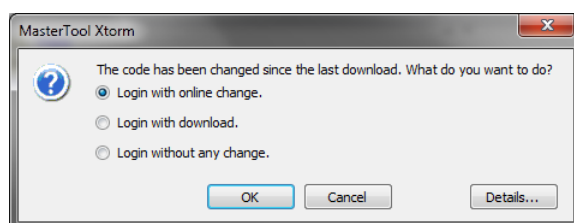


Figura 199: Atualização do Projeto na UCP

A figura abaixo 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 seção **Modo Run**). Dessa forma, o MasterTool Xtorm solicita ao usuário se o Login deve ser executado como download e a UCP parada (ver seção **Modo Stop**) ou a operação deve ser cancelada.

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

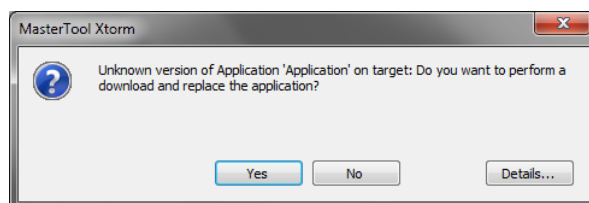


Figura 200: Alteração de Variáveis

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

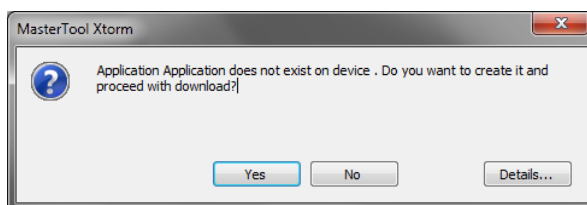


Figura 201: Primeiro Envio de Aplicação

5.13. 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 abaixo.

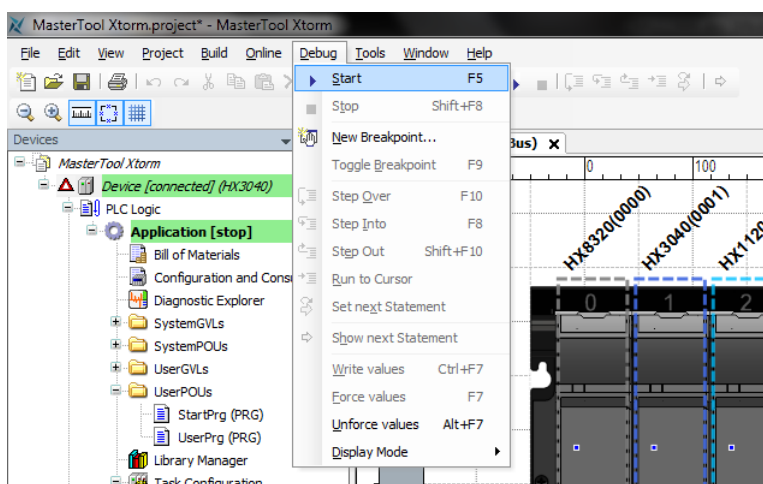


Figura 202: Iniciando a Aplicação

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

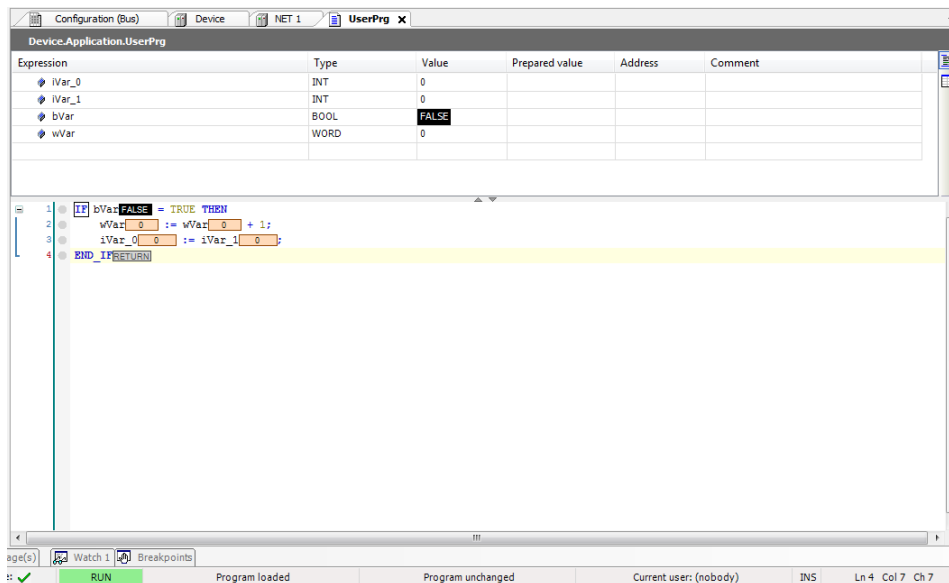


Figura 203: Programa em Execução

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

5.14. Modo Stop

Para interromper a execução da UCP, sem que o software MasterTool Xtorm perca a conexão com a mesma, o usuário deve seleccionar a opção “Stop” (Parar), disponível no menu *Debug* (Depurar), conforme mostra a figura abaixo.

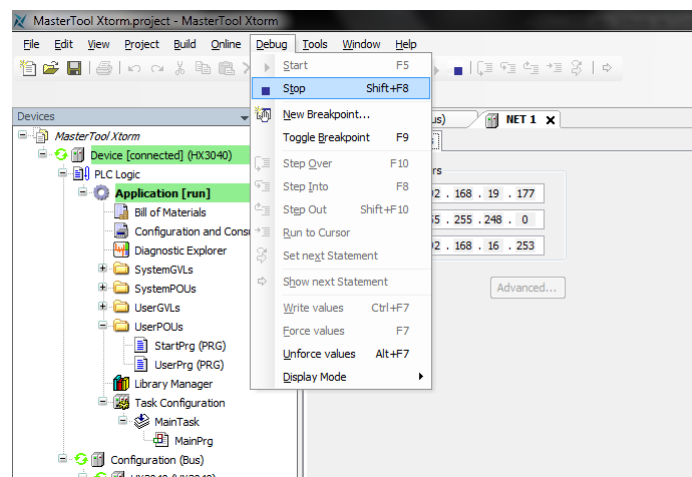


Figura 204: Parando a Aplicação

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

5.15. Monitoração, Escrita e Forçamento de Variáveis

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

- Diretamente na POU (na área de declaração de variáveis ou na área do programa).

- Através da janela de monitoração (menu *View -> Watch*)
- Utilizando a funcionalidade *Trace*, a qual realiza a gravação e exibição dos valores das variáveis internamente na UCP

ATENÇÃO

No modo online há uma limitação de 25000 entradas de variáveis monitoráveis em POU's 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.

ATENÇÃO

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 pela mesma.

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. A tabela abaixo exemplifica em quanto será acrescida, em média, a execução da MainTask quando isto ocorrer:

Tempo de Execução	UCP Ativo			UCP Reserva		
	50 ms	100 ms	200 ms	50 ms	100 ms	200 ms
Acréscimo com 10 forçamentos	2,4 %	2,2 %	1,7 %	4,0 %	3,4 %	2,0 %
Acréscimo com 50 forçamentos	12,0 %	9,2 %	6,0 %	18,0 %	12,0 %	8,0 %
Acréscimo com 128 forçamentos	26,0 %	21,0 %	16,0 %	56,0 %	34,0 %	22,5 %

Tabela 215: Influência do Forçamento de Variáveis em uma UCP Redundante

Para mais informações sobre forçamentos de variáveis no sistema redundante, consulte [Sincronização de Dados Redundantes](#).

ATENÇÃO

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.

5.16. 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 campos informados na tabela abaixo.

Módulo / Driver	Funcionalidade	Campo / Origem
Campos Validados		
HX3040	Agrupamento de Eventos	Nome da Variável Alarme de Atraso Excessivo
Dispositivo MODBUS do MODBUS Cliente	Mapeamentos	Nome da Variável Qualidade
Dispositivo MODBUS do MODBUS Mestre	Mapeamentos	Nome da Variável Qualidade
Outstation do DNP3 Cliente	Mapeamentos	Nome da Variável Qualidade
Módulo de E/S	Bus: Mapeamento de E/S	Variável

Tabela 216: Variáveis Utilizadas em Mais de Uma Origem

5.17. 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 abaixo.

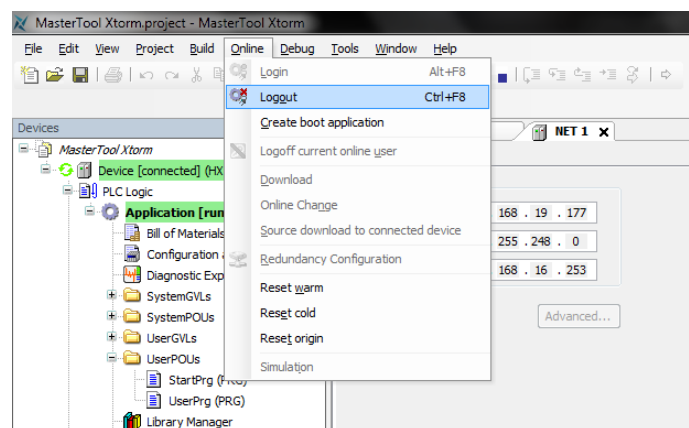


Figura 205: Interrompendo a Comunicação com a UCP

5.18. 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 possíveis de serem simulados.

Seguem abaixo, os recursos indisponíveis no modo 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
- 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 selecionar esta opção no *Menu Comunicação* conforme a figura abaixo. Após isso é exibido um aviso no barra inferior do MasterTool Xtorm que indica que a ferramenta está operando em Modo Simulação.

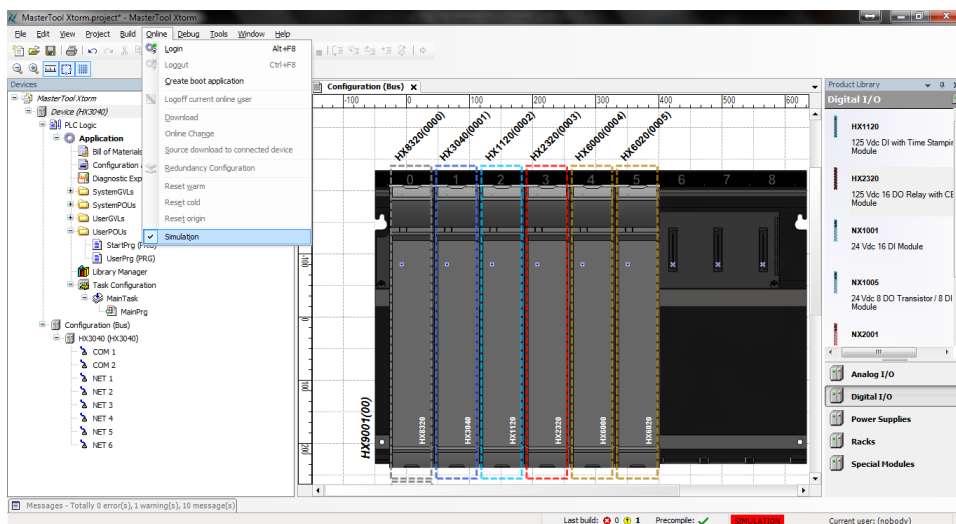


Figura 206: Modo Simulação

Em 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 relacionado 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 ficarão 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 POU's *NonSkippedPrgr* e *UserPrgr*, como se a UCP simulado fosse a UCP Ativa.

5.19. 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 abaixo.

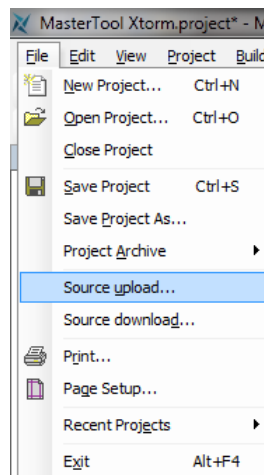


Figura 207: Opção de Upload de Projeto

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

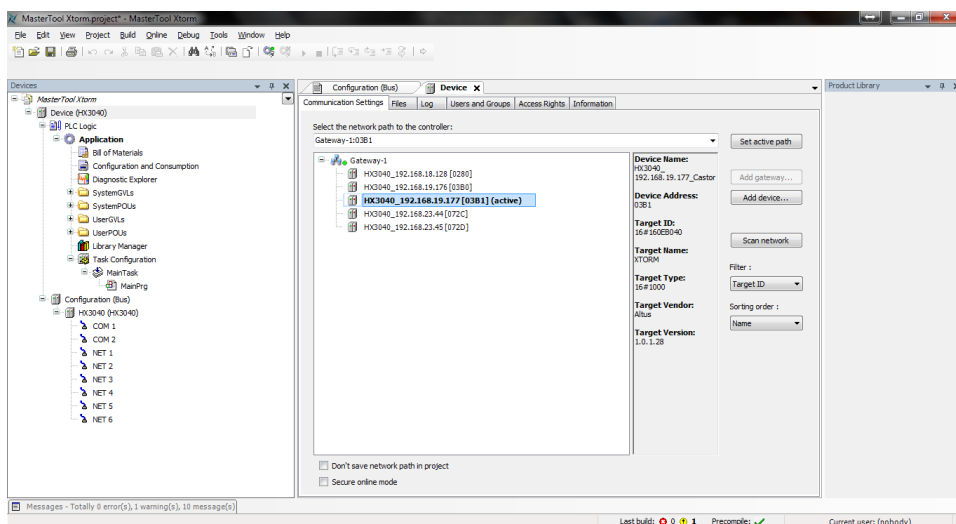


Figura 208: Seleciona a UCP

ATENÇÃO

O tamanho da área de memória para armazenar um projeto na UCP Hadron Xtorm está definido nas [Características Gerais Específicas](#).

ATENÇÃO

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.

5.20. Estados de Operação da UCP

5.20.1. Run

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

5.20.2. Stop

Quando uma UCP está em modo Stop, indica que as tarefas da aplicação estão paradas. O valor das variáveis nas tarefas são mantidas com o valor atual e 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.

5.20.3. 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.

Nota:

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.

5.20.4. Exception

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

5.20.5. Reset a Quente

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

5.20.6. Reset a Frio

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

5.20.7. 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:

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

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

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

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

No caso de 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.

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

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

5.21.1. 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:

```
(*Main POU associated with MainTask that calls StartPrg,
UserPrg, NonSkippedPrg and the system POUs AlarmPrg and EngineeringPrg.
This POU is blocked to edit.*)

PROGRAM MainPrg
VAR
  isFirstCycle : BOOL := TRUE;
END_VAR

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

MainPrg chama outras quatro POUs do tipo programa, denominadas *StartPrg*, *EngineeringPrg*, *AlarmPrg* e *UserPrg*. Enquanto *EngineeringPrg*, *AlarmPrg* e *UserPrg* são sempre chamadas, a *StartPrg* é chamada uma única vez na partida da aplicação do CP.

Diferente dos programas *EngineeringPrg* e *AlarmPrg* que, assim como o *MainPrg* são bloqueados para modificações, os programas *StartPrg* e *UserPrg* podem ser modificados pelo usuário. Inicialmente, quando o projeto é criado a partir do Wizard, estes dois programas editáveis são criados "vazios", mas o usuário poderá inserir código nos mesmos.

5.21.2. Programa StartPrg

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

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

5.21.3. Programa EngineeringPrg

Esta POU é bloqueada para edição e é gerada automaticamente pelo programa. Ela é responsável por mapear as variáveis utilizadas para conversões de engenharia, característica da UCP de Conversão de Engenharia, descrito na seção [Conversão de Engenharia](#).

5.21.4. Programa AlarmPrg

Esta POU também é bloqueada para edição. Nela é gerado automaticamente o código responsável pela atribuição do alarme. Alarme este que pode ser criado nas configurações da aba de Alarmes da UCP. Para maiores informações, veja a seção [Alarmes](#).

5.21.5. Programa UserPrg

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

O usuário pode também criar 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.

5.21.6. 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.

Este programa é bloqueado para edição e responsável por chamar a POU *UserProtPrg*.

5.21.7. Programa UserProtPrg

O programa *UserProtPrg* tem por objetivo controlar os processos com prioridade superior a *UserPrg*. Ou seja, nesta POU o usuário deve inserir suas lógicas de processos que necessitam de uma rápida atuação da UCP.

Este é chamado pelo programa *ProtPrg* que, por sua vez, está associado a tarefa *ProtTask*.

Para maiores informações sobre a tarefa *ProtTask*, veja a seção [Configurações da ProtTask](#).

5.21.8. 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.

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






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 209: GVL Disable em Modo Online

5.21.9. 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:

```

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*.

5.21.10. 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 *RUN*. Através da GVL *Module_Diagnostics* é possível monitorar os diagnósticos de cada módulo que foi adicionado ao projeto, como é possível visualizar na figura abaixo.

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 210: GVL Modules_Diagnostic em Modo Online

5.21.11. 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:

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.

Para os dispositivos de comunicação as variáveis de qualidade se comportam da seguinte maneira, ver a tabela 49.

ATENÇÃO

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 *RUN*. 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 abaixo.

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 211: GVL Qualities em Modo Online

5.21.12. 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:

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 *RUN*. 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 abaixo.

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
iIIN	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 212: GVL ReqDiagnostics em Modo Online

5.21.13. 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 *RUN*. 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 abaixo.

5. PROGRAMAÇÃO INICIAL

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	
[-] wTXRequests	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 213: GVL System_Diagnostics em Modo Online

6. Redundância com UCP HX3040

6.1. Introdução

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 abaixo 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.

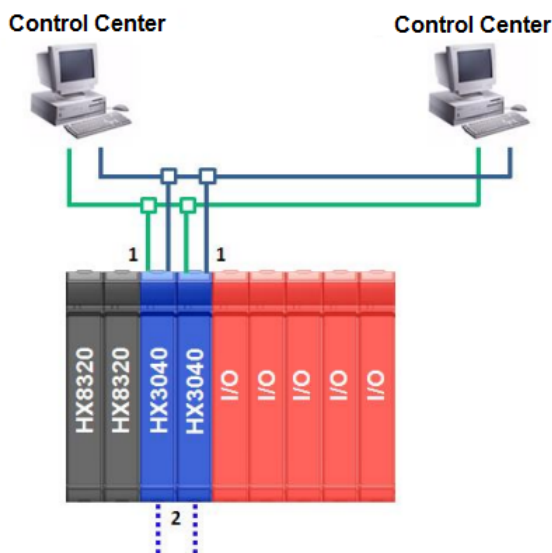


Figura 214: Exemplo de Arquitetura Redundante com UCP HX3040

6.2. Configurações para uma 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. Para realizar esse procedimento, consulte a seção [Wizard para Criação de um Novo Projeto Redundante](#);

2. Configurar as interfaces Ethernet. Para realizar esse procedimento, consultar a seção [Configuração das Interfaces Ethernet](#);
3. Opcionalmente, instanciar protocolos DNP e MODBUS. As configurações de cada protocolo podem ser consultadas nas suas respectivas seções.

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

6.2.1. Identificação de uma UCP HX3040

Ao longo deste manual, 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 no posição 2 do barramento.
- UCPB: a UCP inserida no posição 3 do barramento.

6.2.2. Características Gerais

	Características Gerais de uma UCP redundante
UCPs permitidas	HX3040
Tipo de redundância	Hot-standby
Tolerância a falhas	Tolera, no mínimo, falha simples em equipamentos duplicados nos bastidores. Em casos específicos, pode tolerar múltiplas falhas.
5 estados de redundância	<ul style="list-style-type: none"> - Não-Configurado: estado inicial, também considerado quando a UCP está desligada ou não está executando a tarefa principal (MainTask). - Inicializando: estado temporário assumido depois de Não-Configurado, onde alguns testes determinarão o próximo estado (Inativo, Ativo, Reserva – ou de volta para Não-Configurado). - Inativo: estado atingido depois de determinados tipos de falhas ou para manutenção programada. - Ativo: controlando o processo do usuário. - Reserva: apto para chavear para Ativo e controlar o processo do usuário, caso haja uma demanda (ex: falha da UCP Ativa).
Principais falhas que causam switch-over entre UCP Ativa e UCP Reserva. UCP Reserva chaveia para Ativo e a UCP Ativa pode ir para Inativo ou Não-Configurado.	<ul style="list-style-type: none"> - Falha de alimentação. - UCP (parada de execução da tarefa principal - MainTask). - Falha de ambos os canais de sincronismo entre as UCPs.
Comandos que causam switch-over entre UCP Ativa e UCP Reserva	<ul style="list-style-type: none"> - Comandos via visor das UCPs. - Comandos recebidos do MasterTool Xtorm ou de um sistema SCADA, através desta UTR (local) ou da outra UTR (remota). - Comandos gerados pela aplicação do usuário, por exemplo, em função de outros diagnósticos (ex: falha de comunicação Ethernet), através desta UTR (local) ou da outra UTR (remota).

	Características Gerais de uma UCP redundante
Principais falhas que impedem uma UCP de ficar no estado Reserva, ou assumir o estado Reserva. Tais falhas fazem com que esta UCP assuma os estados Não-Configurado ou Inativo.	<ul style="list-style-type: none"> - Falha de alimentação. - UCP (parada de execução da tarefa principal - MainTask). - Falha em um dos canais de sincronismo (A ou B). - Falha no serviço de sincronismo de dados redundantes. - Projeto diferente daquele da UCP Ativa. - Versão de firmware incompatível com a UCP Ativa.
Comandos que tiram a UCP do estado Reserva	<ul style="list-style-type: none"> - Comandos via visor das UCPs redundantes. - Comandos recebidos do MasterTool Xtorm ou de um sistema SCADA, através desta UTR (local) ou da outra UTR (remota). - Comandos gerados pela aplicação do usuário, por exemplo, em função de outros diagnósticos (ex: falha de comunicação Ethernet), através desta UTR (local) ou da outra UTR (remota).
Tempo de switch-over	- Tipicamente de um a três ciclos da MainTask, dependendo do estímulo para troca de estado (comando ou falha).
Switch-over sem discontinuidades (bump-less)	- Um switch-over não provoca discontinuidades nas saídas do controlador, nem em variáveis internas.
Overhead da redundância (consumo de UCP a cada ciclo da MainTask adicionado pela redundância)	<ul style="list-style-type: none"> - Valor máximo calculado automaticamente pelo MasterTool e informado ao usuário, considerando uma lista de forçamentos redundantes vazia. - Valor médio típico de 15 ms para 100 Kbytes de dados redundantes e fila de 4500 eventos.
Visor da UCP	Entre outros diagnósticos, mostra o estado da redundância (Ativo, Reserva, Inativo, Não-Configurado, Inicializando). Através de opções no menu, permite comandos de switchover ou transições entre estados de redundância para manutenção.
Diagnósticos da redundância	<ul style="list-style-type: none"> - Indicam falhas tanto na UCPA como na UCPB, independente dos seus estados (Ativo ou Não-Ativo). - Evitam “falhas ocultas”. - Permitem manutenção rápida, o que é essencial para obter alta disponibilidade.
Comandos da redundância	<ul style="list-style-type: none"> - Permitem executar as mesmas ações do visor da UCP, além de outros comandos (ex: comandar switchovers). - Podem ser executados na UCP local, ou repassados para a outra UTR (remota) via canais de sincronismo entre as UCPs. - Podem ser recebidos a partir do MasterTool Xtorm ou de um sistema SCADA. - Podem ser executados a partir da aplicação do usuário.
Eventos da redundância	Registram modificações nos diagnósticos e comandos de redundância, com estampa de tempo, permitindo assim uma investigação das causas de um switch-over.
IRIG-B, DNP3 e SNTP (Simple Network Time Protocol)	Permite que os eventos tenham estampa de tempo precisa e ajustada à hora mundial. Também sincroniza o relógio de tempo real da UCP para outras aplicações.
Telas no MasterTool para Diagnósticos, Comandos e Eventos do LOG de redundância	O programador MasterTool Xtorm disponibiliza telas especiais para comandos e diagnósticos da redundância, bem como para mostrar os LOGs de redundância.

	Características Gerais de uma UCP redundante
Sincronização de dados de usuário	A cada ciclo da MainTask, UCPA e UCPB trocam diagnósticos e comandos. O usuário ainda possui 128 bytes de dados redundantes disponíveis dentro dos diagnósticos da UCP redundante, o sincronismo ocorre através dos canais de sincronização entre as UCPs. Desta maneira, uma UCP conhece os diagnósticos e comandos da outra.
Sincronização de dados redundantes	A cada ciclo da MainTask, a UCP Ativa copia dados redundantes para a UCP Não-Ativa através dos canais de sincronismo entre as UCPs. Dados não-redundantes não são sincronizados.
Sincronização da lista de forçamentos redundantes	A cada ciclo da MainTask, a UCP Ativa copia a lista de forçamentos redundantes para a UCP Não-Ativa através dos canais de sincronismo entre as UCPs. Esta lista inclui apenas variáveis redundantes forçadas. Desta maneira, UCPA e UCPB poderão ter diferentes conjuntos de dados não redundantes forçados, pois estes forçamentos não são sincronizados. A fila de eventos também é sincronizada neste procedimento da UCP, mesmo que a lista não esteja habilitada pelo usuário ela será sincronizada entre as UCPs mesmo não sendo usada pelo usuário. Para mais informações consultar capítulo Sincronização de Dados Redundantes
Projeto único para UCPA e UCPB	Existe um projeto único comum para a UCPA e UCPB, gerado pelo MasterTool. O projeto é composto do projeto de aplicação (código executável) e do <i>project archive</i> (código fonte).
Identificação de uma UCP	As UCPs são identificadas como UCPA ou UCPB automaticamente, dependendo da posição no qual estão inseridas. Ao inserir uma UCP na posição 2, ela recebe a designação de UCPA, e na posição 3, UCPB. Esta identificação não faz parte do projeto aplicativo gerado pelo MasterTool Xtorm. A identificação de cada UCP viabiliza a característica de ter um projeto único para UCPA e UCPB.
Sincronização automática do projeto	Se o projeto da UCP Ativa se tornar diferente do projeto da UCP Não-Ativa, o mesmo é copiado da UCP Ativa para a UCP Não-Ativa. Esta sincronização pode durar diversos ciclos da MainTask. Deve-se lembrar que o projeto é composto do projeto aplicativo (código executável) e do código fonte (Project archive), sendo ambos sincronizados.
Endereços IP privados para UCPA e UCPB	Pode-se conectar a uma UCP específica (UCPA ou UCPB) utilizando um endereço IP privado do mesmo, para, por exemplo, obter diagnósticos específicos dessa UCP, usando a interface Web ou a ferramenta de programação Mastertool. Para consulta a diagnósticos associados a protocolos de comunicação só será possível fazê-lo na UCP ativa.
IP Ativo	Nome de uma estratégia que permite que clientes Ethernet se conectem a um servidor na UCP redundante, utilizando sempre o mesmo endereço de IP. Isto evita a necessidade de complexos scripts para mudar o endereço IP quando ocorrem switchovers devido à redundância. O Endereço de IP Ativo estará sempre associado à interface NET (i) da UCP Ativa.

Características Gerais de uma UCP redundante	
NIC Teaming	Nome de uma estratégia que permite que duas interfaces Ethernet de uma UCP formem um par redundante compartilhando um mesmo endereço IP. Desta forma, podem-se construir redes Ethernet redundantes com maior facilidade, sem que os clientes conectados a um par NIC Teaming tenham de implementar complexos scripts para chavear endereços IP.
Tarefa de usuário cíclica	Somente uma tarefa é permitida. A tarefa é cíclica e tem o nome pré-definido: MainTask. A MainTask se destina a ser a tarefa principal, com a maior parte da lógica de usuário.
POUs principais	<p>Na criação de um projeto redundante, o MasterTool gera automaticamente program POU's vazias que devem ser preenchidas pelo usuário.</p> <p>POUs associadas à MainTask:</p> <ul style="list-style-type: none"> - NonSkippedPrg: esta POU é executada nas duas UCPs (UCPA e UCPB), independente do estado de redundância (Ativo ou Não-Ativo). Destina-se ao gerenciamento de diagnósticos e comandos especiais. - UserPrg: esta program POU é executada somente na UCP Ativa, e destina-se ao controle do processo do usuário final. - StartPrg: esta program POU é executada nas duas UCPs (UCPA e UCPB), somente no primeiro ciclo da MainTask. - MainPrg: POU que faz a chamada de outras assim como administra estas POU's. Não pode ser editada. - AlarmPrg: Esta POU estará associada aos alarmes utilizados na aplicação do usuário. Não pode ser editada. - EngineeringPrg: Assim como a AlarmPrg, é criada automaticamente e trata os processos de engenharia do projeto. Não pode ser editada.

Tabela 217: Características Gerais de uma UCP Redundante

6.3. Princípios 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 que serão utilizados nos próximos capítulos.

6.3.1. 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.

6.3.2. Estrutura do Projeto Redundante

6.3.2.1. 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.

6.3.2.2. Tarefas Cíclicas: MainTask e ProfTask

O projeto de uma UCP redundante pode possuir duas tarefas: MainTask e ProfTask (criação opcional no Wizard), que são cíclicas. O usuário pode ajustar o intervalo de cada tarefa.

6.3.2.3. Programa MainPrg

A tarefa MainTask está associada a uma única POU do tipo programa, denominada MainPrg. O programa MainPrg é criado automaticamente.

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

```
(*Main POU associated with MainTask that calls StartPrg,
UserPrg, NonSkippedPrg and the system POUs AlarmPrg and EngineeringPrg.
This POU is blocked to edit.*)

PROGRAM MainPrg
VAR
  isFirstCycle : BOOL := TRUE;
END_VAR

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 tem por objetivo ser executada uma única vez no primeiro ciclo de cada uma das UCPs, ou seja, 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.

6.3.2.4. 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. Para maiores informações, consultar a seção [Configurações da MainTask](#).

6.3.2.5. 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.

Para maiores informações, consultar a seção [Programa NonSkippedPrg](#).

6.3.2.6. 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.

6.3.2.7. Programa UserProtPrg

O principal objetivo deste programa, que é executado somente na UCP Ativa, é controlar processos com prioridade superior a *UserPrg*. Ou seja, nesta POU o usuário deve inserir suas lógicas de processos que necessitam de uma rápida atuação da UCP.

Este é chamado pelo programa *ProtPrg* que, por sua vez, está associado a tarefa *ProtTask*.

Para maiores informações sobre a tarefa *ProtTask*, consultar a seção [Configurações da ProtTask](#).

6.3.2.8. 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.

Para maiores informações, consultar a seção [Programa NonSkippedProtPrg](#).

6.3.2.9. 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.

6.3.2.10. 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.

6.3.2.11. 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.

6.3.2.12. 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.

ATENÇÃO

Não se deve confundir variáveis simbólicas com variáveis simbólicas endereçadas através da diretiva AT. Variáveis simbólicas que utilizam a diretiva AT são nomes simbólicos atribuídos 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:

- Quando declaradas em POU's do tipo "programa" criadas na aplicação do usuário, com exceção dos programas *NonSkippedPrg* e *NonSkippedProtPrg*.
- Quando declaradas em GVL's criadas na aplicação do usuário e estas GVL's marcadas como redundantes.

Variáveis simbólicas são não-redundantes nos seguintes casos:

- Quando declaradas nos programas *NonSkippedPrg* e *NonSkippedProtPrg*, já descritos anteriormente.
- Quando declaradas em POU's do tipo "função". Observar que tais tipos de POU's 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 POU's do tipo "função", isto é considerado uma má prática de programação. Tais variáveis estáticas, caso sejam criadas, serão consideradas não redundantes.
- Quando declaradas em POU's 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 POU's do tipo programa ou dentro de GVL's, 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.

6.3.3. 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.

Onde:

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.

6.3.4. 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

Na seção [Manutenção](#), as seguintes subseções fornecerão maiores detalhes sobre as [Estrutura de Diagnósticos da Redundância](#):

- [Diagnósticos da Redundância](#)
- [Comandos da Redundância](#)
- [Informações do Usuário Trocados entre UCPA e UCPB](#)

6.3.5. 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 tem 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:

- Primeiro, executa-se o serviço Troca de Diagnósticos e Comandos.
- 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.

6.3.5.1. 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 hajam duas UCPs no barramento (nos slots corretos) e desde que a aplicação presente nas duas UCPs seja redundante.

6.3.5.2. 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.

ATENÇÃO

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.

ATENÇÃO

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.

6.3.6. 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.

6.3.6.1. 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 é 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.

ATENÇÃO

Uma sincronização de projeto terá os mesmos efeitos de um download na UCP Não-Ativa.

6.3.7. 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.

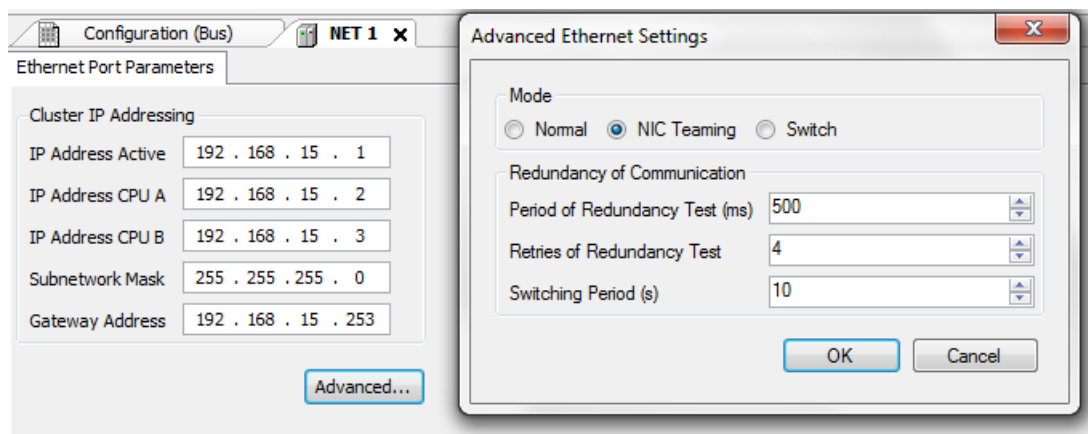


Figura 215: Exemplo de Rede Ethernet Redundante com NIC Teaming

Neste exemplo a UCP HX3040 conecta-se à rede de supervisão (SCADAs). 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.

ATENÇÃO

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.

Maiores detalhes sobre configuração e diagnósticos de portas NIC Teaming são fornecidos na seção [Configurações das Portas Ethernet da UCP HX3040 \(NET 1 até NET 6\)](#).

6.3.8. 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 da mesma, 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.

6.3.8.1. 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.

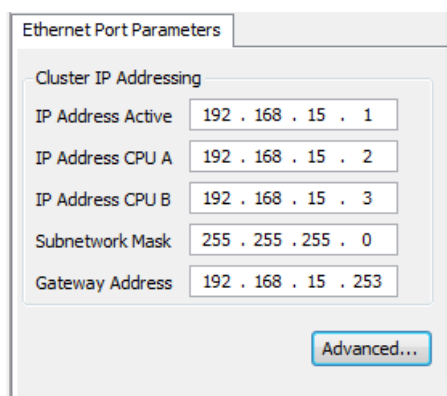


Figura 216: Método IP Ativo - HX3040 Redundante

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.

6.3.9. 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.

6.3.10. 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

ATENÇÃO

Frequentemente este manual utilizará a designação “*Não-Ativo*” para qualquer estado diferente de Ativo, ou seja, para designar qualquer um dos outros quatro estados: *Reserva*, *Inativo*, *Não-Configurado* e *Inicializando*.

A seguir, estes cinco estados são descritos brevemente. Maiores detalhes sobre os estados de uma UCP redundante serão abordados na seção [Transições entre Estados de Redundância](#), ao descrever a máquina de estados e as causas das transições entre os mesmos.

6.3.10.1. 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 (ver seção [Serviços de Sincronização Cíclicos através de Canais de Sincronismo de Redundância](#) desde que as condições para sua execução estejam presentes;
- Os serviços de sincronização esporádicos também podem ser executados (ver seção [Serviços de Sincronização Esporádicos através dos Canais de Sincronismo de Redundância](#)).

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

6.3.10.2. 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 UCPC, de acordo com o onde a UCP está inserido);
- Verifica se há problemas nos parâmetros de configuração extraídos do projeto MasterTool;
- Os serviços de sincronização cíclicos são executados (ver seção [Serviços de Sincronização Cíclicos através de Canais de Sincronismo de Redundância](#) desde que as condições para sua execução estejam presentes;
- Verifica compatibilidade de versões de firmware entre os dois UCPCs;
- Verifica igualdade de projetos entre os dois UCPCs;
- Caso a outra UCP esteja em estado Ativo, verifica a possibilidade de estabelecer comunicação passiva com o mesmo, 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 o mesmo, 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.

6.3.10.3. 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).

6.3.10.4. 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.

6.3.10.5. 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.

6.3.11. 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 consequentemente 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”).

ATENÇÃO

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](#). Uma descrição mais detalhada destes comandos pode ser encontrada na seção [Comandos da Redundância](#).

6.3.12. Transições entre Estados de Redundância

A figura seguinte mostra a máquina de estados da redundância, ilustrando todas as possíveis transições entre estados de redundância.

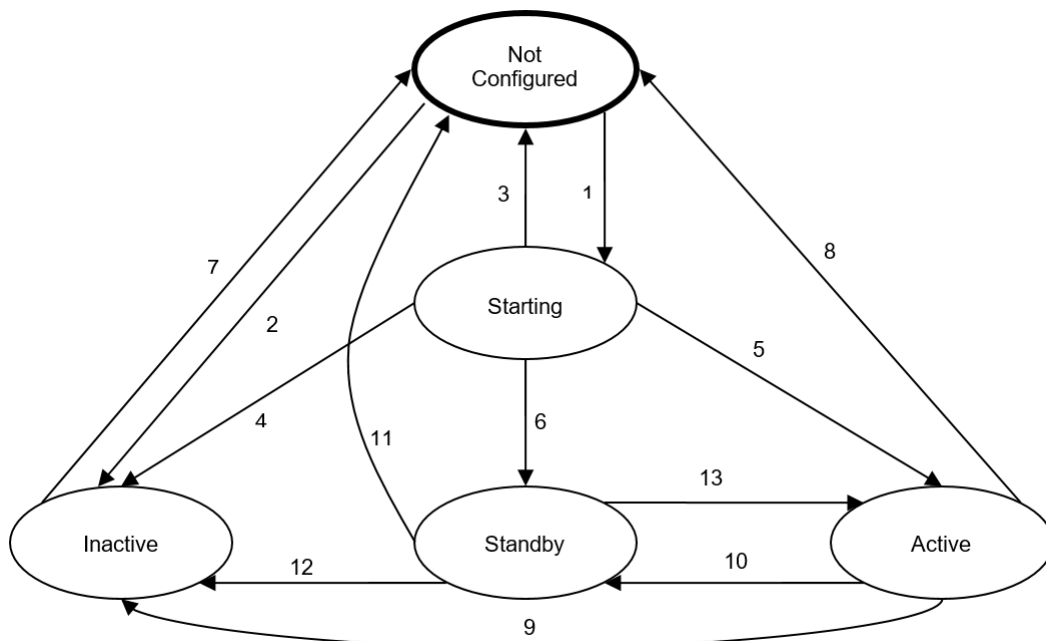


Figura 217: Máquina de Estados da Redundância

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). Tais comandos estão melhores detalhados na seção mencionada preliminarmente [Estruturas de Dados de Diagnósticos, Comandos e Usuário](#). 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.

6.3.12.1. 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 também em outras situações, descritas nas próximas subseções.
- 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.

6.3.12.2. Transição 2 – Não-Configurado para Inativo

- Ocorreu alguma exceção de software nesta UCP (cão-de-guarda, access violation, illegal instruction).

6.3.12.3. Transição 3 - Inicializando para Não-Configurado

- 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.

6.3.12.4. Transição 4 - Inicializando para Inativo

- 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.

6.3.12.5. 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.
- O estado da outra UCP não pode ser descoberto via link interno de redundância, e além disso esta 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.

6.3.12.6. 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.

6.3.12.7. Transição 7 - Inativo para Não-Configurado

- Esta UCP foi desligada ou reinicializada (Reset a Quente, Reset a Frio ou Reset Origem).
- 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.

6.3.12.8. Transição 8 - Ativo para Não-Configurado

- Esta UCP foi desligada ou reinicializada (Reset a Quente, Reset a Frio ou Reset Origem).

6.3.12.9. Transição 9 - Ativo para Inativo

- Ocorreu alguma exceção de software nesta UCP (cão-de-guarda, access violation, illegal instruction).
- 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.

6.3.12.10. 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 UCPC 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.
- 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.

6.3.12.11. Transição 11 – Reserva para Não-Configurado

- 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 onde 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.

6.3.12.12. Transição 12 – Reserva para Inativo

- Ocorreu alguma exceção de software nesta UCP (cão-de-guarda, access violation, illegal instruction).
- O comando para transicionar 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.

6.3.12.13. Transição 13 – Reserva para Ativo

- 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.

6.3.13. 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:

- 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.
- 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](#).

6.3.14. Falhas mais Comuns Causadoras de Switchovers Automáticos 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](#).

- Falha no circuito interno de alimentação na UCP Ativa.
- Falha no acesso ao barramento do bastidor pela UCP Ativa.
- Falha 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.).

6.3.15. Falhas Associadas à Switchovers entre UCPs Gerenciados pelo Usuário

Entre as transições examinadas na seção [Transições entre Estados de Redundância](#), algumas possibilitam que o usuário gere 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. Maiores detalhes sobre estas estruturas de dados são fornecidos nas seções:

- [Diagnósticos da Redundância](#)
- [Comandos da Redundância](#)
- [Informações do Usuário Trocados entre UCPA e UCPB](#)

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 também 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:

- 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.
- 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.

6.3.16. 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.

6.3.16.1. 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, tratam-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 a mesma não se constitui em uma indisponibilidade total do sistema. Ela se constitui em uma indisponibilidade parcial, somente das malhas de controle que utilizam este módulo de E/S.

Embora não haja previsão de redundância de módulos de E/S, a aplicação do usuário pode gerenciá-la em casos especiais. Por exemplo, o usuário pode inserir três módulos de entradas analógicas 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 as mesmas. Tais soluções, em geral, também implicam na redundância de transdutores e atuadores no campo.

6.3.16.2. Falhas Simples sem Indisponibilidade Causando um Switchover

UCPs HX3040 redundantes toleram falhas simples sem causar indisponibilidade, mas causam switchover.

6.3.16.3. Falhas Simples sem Indisponibilidade

Fontes de alimentação HX8300 e HX8320 redundantes toleram falhas simples, sem causar indisponibilidade e sem gerar switchover.

6.3.17. 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 100Kb de dados. Para o cálculo efetivo do tempo máximo de transferência, é estimado um tempo mínimo de 7ms (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 $7\text{ms} + 6,4\text{ms/Kbyte} \cdot 117\text{Kbytes} = 15\text{ms}$. Este tempo é mostrado pelo Mastertool como “*overhead máximo de redundância*”.

ATENÇÃO

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 15ms, o tempo de ciclo deverá ser de pelo menos 20ms. Contudo, em aplicações redundantes, o tempo mínimo permitido é de 50ms, devido ao tempo necessário para processamento da fila de eventos no switchover.

ATENÇÃO

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 POU's principais (NonSkippedPrg e UserPrg). Este tempo normalmente é medido após o desenvolvimento do projeto (descontando o tempo adicional para redundância);
- Tempo necessário para detecção e geração de eventos de pontos internos (por exemplo, a ocorrência de 1000 eventos de pontos analógicos com banda morta no mesmo ciclo pode levar até 15ms);
- 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 POU's de proteção (NonSkippedProtPrg e UserProtPrg), quando utilizadas. Como a prioridade da tarefa ProtTask, associada a estas POU's, 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 800us se a mesma 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. Para maiores informações, consultar a seção [Tarefas Cíclicas: MainTask e ProtTask](#).

ATENÇÃO

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.

6.4. Programação de uma UCP Redundante

6.4.1. Wizard para Criação de um Novo Projeto Redundante

Para criar um novo projeto redundante, deve-se utilizar o comando *Arquivo -> Novo Projeto*, e selecionar o *Projeto MasterTool Padrão*.

Inicialmente, o usuário deve informar o nome que deseja dar ao projeto e o diretório onde deseja armazená-lo, conforme mostra a figura abaixo.

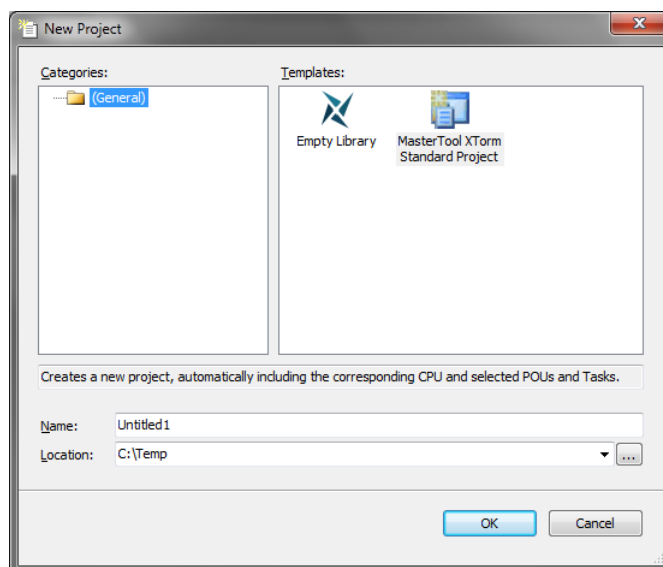


Figura 218: Classificação do Projeto

A seguir, o Wizard que gera o projeto de redundância realiza alguns questionamentos ao usuário quanto à configuração desejada, os quais devem ser respondidos sucessivamente.

O primeiro ponto a ser definido é a configuração inicial de hardware da UTR:

- Selecionar o modelo da UCP: como a redundância está implementada somente na HX3040, a mesma deve ser selecionada pelo usuário;
- Selecionar o modelo do bastidor: existem duas opções de bastidores disponíveis e a escolha do mesmo dependerá da quantidade de módulos utilizados;
- Selecionar o modelo da fonte de alimentação;
- Selecionar a configuração da redundância de UCP;
- 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).

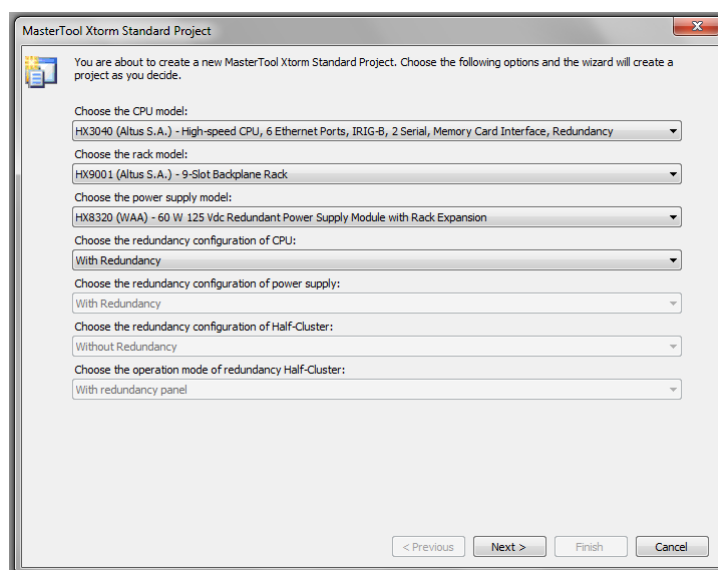


Figura 219: Configuração Inicial de Hardware

Após, o usuário deve definir a quantidade e os tipos de E/S utilizadas na aplicação:

- Selecionar o número de pontos de entradas digitais;

- Selecionar o número de pontos de saídas digitais;
- Selecionar o número de pontos de entradas analógicas V/I;
- Selecionar o número de pontos de entradas analógicas RTD.

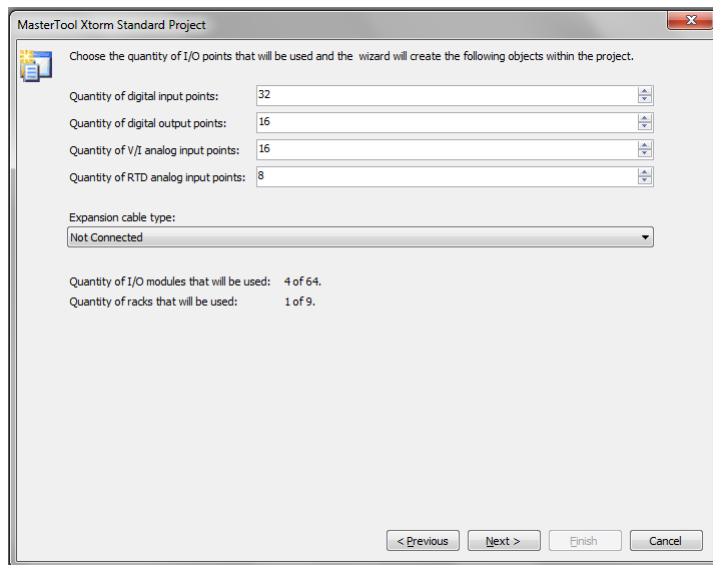


Figura 220: Configuração dos Pontos de E/S

Então, deve ser selecionado o perfil de projeto e a linguagem padrão para a criação dos programas:

- 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;
- 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.

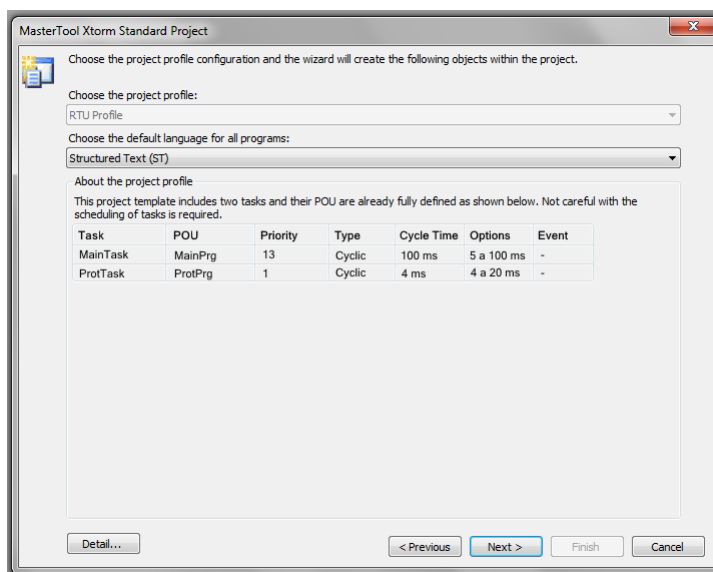


Figura 221: Perfil de Projeto e Linguagem Padrão

Para finalizar, o usuário deve selecionar a linguagem de programas comuns e associados à redundância:

- Programas associados à MainTask (MainPrg);
- Programas associados a tarefas cíclicas: deverá ser, obrigatoriamente, em linguagem ST, sendo que o MasterTool desabilita as outras opções;
- Programas associados às tarefas principais da redundância.

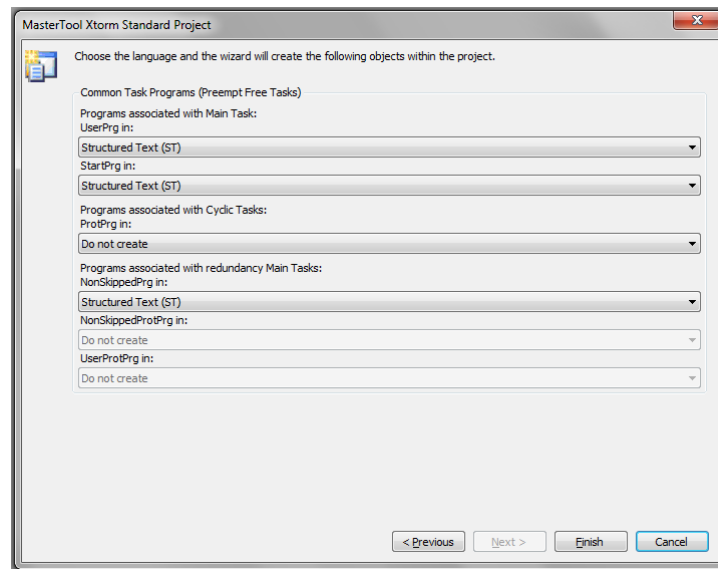


Figura 222: Linguagem dos Programas Específicos

ATENÇÃO

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. Para mais informações, ver a seção [Limitações na Programação de uma UCP Redundante](#).

ATENÇÃO

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 inicial 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)

6.4.2. 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. Os capítulos 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.

6.4.2.1. 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.

6.4.3. Configurações das Portas Ethernet da UCP HX3040 (NET 1 até NET 6)

6.4.3.1. Configuração do Endereço de IP

A figura abaixo 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.

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 223: 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 está descrito na seção [IP Ativo](#).

ATENÇÃO

Os três endereços IPs (Ativo, da UCPA e da UCPB), bem como o endereço do Gateway, devem pertencer à mesma subrede.

ATENÇÃO

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 também selecionar o modo Switch.

6.4.4. 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.

6.4.5. 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:

- Intervalo da MainTask
- Tempo de cão-de-guarda

Diversas considerações devem ser feitas para ajustar adequadamente o intervalo da MainTask:

- O intervalo deve ser suficientemente baixo para controlar o processo efetivamente, observando os tempos de resposta de todos os laços de controle.
- O intervalo deve ser suficientemente alto para acomodar, no mínimo, a soma dos dois seguintes tempos:
 - O tempo máximo de execução das POUs NonSkippedPrg e UserPrg, em conjunto
 - O tempo necessário para gerenciar a redundância (overhead da redundância)
- Além disso, 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.

Por 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.

6.4.5.1. 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.

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

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 POUs 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 POUs criadas pelo usuário serão, inicialmente, redundantes.

ATENÇÃO

Variáveis do tipo VAR_TEMP não devem ser utilizadas no programa redundante.

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

ATENÇÃO

Quando chamar POUs adicionais do tipo programa dentro de NonSkippedPrg, desmarque as mesmas na janela Redundancy Configuration do MasterTool. Por padrão as variáveis simbólicas declaradas dentro destas POUs serão redundantes e, dentro da NonSkippedPrg, normalmente desejam-se variáveis não redundantes. Normalmente o código de NonSkippedPrg é pequeno, dispensando a chamada de POUs adicionais do tipo programa para sua estruturação, mas, caso a sua estruturação seja necessária, o mais recomendável é utilizar bloco funcional ou funções.

ATENÇÃO

Não se recomenda utilizar os blocos funcionais TOF_RET, TON_RET, TOF e TON no programa NonSkippedPrg. Ver seção [Limitações na Programação de uma UCP Redundante](#).

6.4.6. Configurações da ProfTask

A *ProfTask* 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 *ProfTask* é uma tarefa do tipo cíclica e terá sua prioridade fixada como 1.

6.4.6.1. 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*".

6.4.6.2. 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.

6.4.7. 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.

ATENÇÃO

Por boa prática, 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.

6.4.8. POUs do Tipo Programa com Variáveis Simbólicas Redundantes

O usuário pode declarar variáveis simbólicas redundantes em POUs do tipo programa, com exceção da POU *NonSkippedPrg* e da *NonSkippedProtPrg* onde as variáveis simbólicas declaradas são consideradas não-redundantes.

O usuário pode criar outras POU, 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 criadas pelo usuário serão, inicialmente, redundantes.

ATENÇÃO

Por boa prática, é recomendado evitar a utilização da diretiva AT em POU que sejam redundantes, a fim de evitar o mapeamentos de variáveis em áreas não redundantes.

6.4.9. 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 a mesma não respeite esta condição.

6.4.10. 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.

6.4.10.1. Limitações em GVLs e POU 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.

6.4.10.2. Limitações no Programa Não-Redundante (StartPrg, NonSkippedPrg e NonSkippedProtPrg)

Em uma POU do tipo programa que não seja redundante, no caso, as POU *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 os mesmos 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*. Ver seção [Timer Não-Redundante](#).
- Não podem ser utilizadas POU 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.

6.4.11. 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 estão descritos na tabela abaixo.

eRedState	
NOT_CONFIGURED	0
STARTING	2
STANDBY	3
ACTIVE	4
INACTIVE	5
NOT_AVAILABLE	6

Tabela 218: Estados de Redundância

6.4.12. 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. Veja mais em [Informações do Usuário Trocados entre UCPA e UCPB](#).

6.5. Carga de Programas em uma UCP Redundante

A seção [Programação de uma UCP Redundante](#) tratou de aspectos relativos ao desenvolvimento de um projeto para uma UCP redundante com UCP HX3040.

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.

6.5.1. 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.

ATENÇÃO

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, esta automaticamente irá se sincronizar com a UCP já em funcionamento, caso a mesma esteja em estado Ativo.

6.5.1.1. 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 em [Menu Informativo e de Configuração da UCP](#). 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.

6.5.1.2. 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.

6.5.1.3. 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 também 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 (ver seção [Configurações das Portas Ethernet da UCP HX3040 \(NET 1 até NET 6\)](#)).

6.5.1.4. 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 CPs 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.

6.5.1.5. 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 \(NET 1 até NET 6\)](#)

Obviamente, também é possível fazer um projeto redundante completo e só depois carregá-lo na UCPA e UCPB, por exemplo, caso o hardware destes CPs 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.

Nota:

Ao fim do envio do projeto para uma das UCPs, a outra UCP fará download automaticamente do projeto por canais de sincronismo internos.

ATENÇÃO

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) – ver seção [Configurações das Portas Ethernet da UCP HX3040 \(NET 1 até NET 6\)](#).

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.

Para maiores detalhes sobre como reconectar o MasterTool, ver seção [Conexão do MasterTool com uma UCP HX3040 de uma UCP Redundante](#).

6.5.2. 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:

1. Endereço de IP da UCPA: endereço de NET 1 exclusivo para a UCPA.
2. 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 os CPs 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:

1. *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.
2. *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.

ATENÇÃO

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.

6.5.3. 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.

ATENÇÃO

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.

6.5.4. 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. Veja mais em [Carga Off-Line de Modificações com Interrupção do Controle do Processo](#).

6.5.4.1. Modificações que Demandam Carga Off-Line com Interrupção do Controle do Processo

As seguintes modificações em um projeto farão com que o mesmo se torne impossível de ser carregado num sistema redundante sem a interrupção do controle do processo:

- Adicionar ou remover dispositivos na árvore de dispositivos, por exemplo:
 - Módulos de E/S;
 - 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;
 - Parâmetros de protocolos instanciados, seus mapeamentos, etc.
- Modificações nas configurações das tarefas.

6.5.4.2. Modificações que Permitem Carga On-Line

Em princípio, as modificações não citadas na seção [Modificações que Demandam Carga Off-Line com Interrupção do Controle do Processo](#), permitem carga on-line.

Mesmo assim, 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, POUs e GVLs, 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 deve ser consultado MU299048.

6.5.5. Carga On-Line de Modificações

Na seção [Carga de Modificações Off-Line e On-Line](#), foram descritas modificações que demandam carga off-line e aquelas que permitem carga on-line.

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 da 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.

6.5.6. 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, esta UCP passa para o estado Inicializando e depois voltará para o estado Reserva.

6.6. Manutenção de UTRs Redundantes

6.6.1. 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.

6.6.1.1. 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”.

6.6.1.2. 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.

6.6.2. 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.

6.6.2.1. Estado de Redundância da UCP

O estado de redundância da UCP, descritos em [Estados de uma UCP Redundante](#), é visto nos três caracteres iniciais da segunda linha da tela principal, conforme mostra a seção [Visor Gráfico](#). 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).

6.6.2.2. Telas Abaixo do Menu Redundância

Existe um menu denominado *REDUNDANCIA*, 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](#).

6.6.3. 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:
 - *RedDgnLoc*: contém diagnósticos de redundância da UCP local (onde o MasterTool está conectado), como por exemplo, o estado da redundância da UCP. Essa seção é descrita em [Diagnósticos da Redundância](#).
 - *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. Essa seção é descrita em [Diagnósticos da Redundância](#).
 - *RedCmdLoc*: contém comandos de redundância gerados nesta UCP (local), por exemplo, através de escritas a partir de um sistema SCADA, ou gerados em POUs desta UCP (ex: *UserPrg* ou *NonSkippedPrg*). Essa seção é descrita em [Comandos da Redundância](#).
 - *RedCmdRem*: trata-se de uma cópia de *RedCmdLoc* da outra UCP (remota), recebida via canais de sincronismo. Essa seção é descrita em [Comandos da Redundância](#).
 - *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. Essa seção é descrita em [Informações do Usuário Trocados entre UCPA e UCPB](#).
 - *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. Essa seção é descrita em [Informações do Usuário Trocados 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*.

6.6.3.1. Diagnósticos da Redundância

Os diagnósticos da Redundância podem ter várias utilidades. Entre elas:

- Podem ser consultados para verificar a existência de algum problema que precisa ser sanado.
- Podem ser utilizados para consultar o status do sistema, bem como a tomada de ações em casa 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.

ATENÇÃO

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:

- *RedundancyDiagnostics*: diagnósticos gerais da redundância.
- *SyncLinkDiags[1]*: diagnósticos do canal de sincronismo A.
- *SyncLinkDiags[2]*: diagnósticos do canal de sincronismo B.
- *SyncLinkStatistics*: estatísticas comuns para os canais de sincronismo A e B, para contagem de sucessos e falhas dos serviços de sincronização.

A subestrutura “*RedundancyDiagnostics*” possui os seguintes campos para diagnósticos gerais da redundância:

.RedundancyDiagnostics.*	Tipo	Descrição
bConfigDone	BOOL	TRUE – O processo de configuração, executado no estado Não-Configurado, terminou. FALSE – O processo de configuração, executado no estado Não-Configurado, ainda não terminou ou não foi executado.
bConfigError	BOOL	TRUE – O processo de configuração, executado no estado Não-Configurado, terminou com erros. Trata-se de um erro de sistema, normalmente não esperado. Entre em contato com o suporte da Altus para reportá-lo. Informe também o valor do diagnóstico ConfigErrorCode para o suporte da Altus. FALSE – O processo de configuração ocorreu com sucesso ou não foi realizado.
bTooManyRedAreas	BOOL	TRUE – O número de áreas redundantes excedeu o número máximo permitido. Trata-se de um erro de sistema, normalmente não esperado. Entre em contato com o suporte da Altus para reportá-lo. FALSE – número de áreas redundantes está dentro do esperado.
bTemporaryBufferTooSmall	BOOL	TRUE – Estrutura de dados intermediária com tamanho insuficiente. Trata-se de um erro de sistema, normalmente não esperado. Entre em contato com o suporte da Altus para reportá-lo. FALSE – O tamanho da estrutura de dados está dentro do esperado.
bIncompatibleApplication	BOOL	TRUE – A aplicação não é compatível entre os dois CPs. Foi efetuado o download de uma nova aplicação para um dos UCP com uma das seguintes alterações: Modificação da área de dados redundantes; Modificação de variáveis simbólicas redundantes. Enquanto este diagnóstico estiver ligado, um dos CPs ficará em estado Inativo até que a mesma aplicação esteja presente nos dois CPs. Isto implica em recarregar a aplicação antiga nos dois CPs ou atualizar os dois CPs com a nova aplicação. Para mais informações sobre como proceder, consultar capítulo Carga de Programas em uma UCP Redundante . FALSE – A aplicação corrente em ambos UCPs é compatível, ou seja, a mesma.
bIncompatibleFirmware	BOOL	TRUE – Esta UCP foi para o estado Não-Configurado pois sua versão de firmware é incompatível com a versão de firmware da UCP Ativa. FALSE – versão de firmware da UCP Ativa é compatível com a versão de firmware da UCP Não-Ativa.

.RedundancyDiagnostics.*	Tipo	Descrição
bExchangeSync	BOOL	TRUE – O serviço de sincronização Troca de Diagnósticos e Comandos foi executado com sucesso neste ciclo da MainTask. FALSE – A estrutura RedDgnRem tem valores obsoletos ou inválidos, pois não foi lida da outra UCP (remota) neste ciclo da MainTask.
bRedDataSync	BOOL	TRUE – O serviço Sincronização de Dados Redundantes foi executado com sucesso neste ciclo da MainTask. FALSE – O serviço Sincronização de Dados Redundantes não foi executado com sucesso neste ciclo da MainTask.
bApplicationProjectDiff	BOOL	TRUE – O projeto aplicativo desta UCP é diferente do presente na outra UCP. UCP Ativa. FALSE – O projeto aplicativo deste UCP é igual ao da outra UCP.
bProjectArchiveDiff	BOOL	TRUE – O Project archive desta UCP é diferente do presente na outra UCP. FALSE – O Project archive desta UCP é igual ao da outra UCP.
bOnlineChangeApply	BOOL	TRUE – Foi realizada alguma alteração online na aplicação e esta ainda não foi sincronizada com a UCP reserva. FALSE – Não foram realizadas alterações online na aplicação ou estas já foram sincronizadas com a UCP reserva.
bBusError	BOOL	TRUE – Detectada alguma falha no acesso ao barramento. FALSE – Não há falha de acesso ao barramento.
bBusIdle	BOOL	TRUE – Não há atividade de barramento detectada pela UCP (trabalhando em modo passivo) FALSE – Há atividade de barramento detectada pela UCP.
eCPU_ID	ENUM (BYTE)	Este diagnóstico informa a identificação deste UCP: - 0 = não redundante - 2 = UCPA - 3 = UCPB Trata-se de uma cópia da identificação da UCP, conforme descrito na seção Identificação de uma UCP HX3040 .
eRedState	ENUM (BYTE)	Informa o estado de redundância desta UCP: - Não-Configurado = 0 - Inicializando = 2 - Reserva = 3 - Ativo = 4 - Inativo = 5
ePreviousRedState	ENUM (BYTE)	Valor que o diagnóstico RedState possuía antes da última transição de estados.
eStateChangeReason	ENUM (BYTE)	Código numérico que indica a razão pela qual o houve a troca de estado da redundância, essas informações estão disponíveis na tabela 222 .

.RedundancyDiagnostics.*	Tipo	Descrição
eAppState	ENUM (BYTE)	Modo da aplicação: - Desconhecido = 0 - Run = 1 - Stop = 2 - Breakpoint = 3
wRedStateDuration	WORD	Mede há quanto tempo (milissegundos) o estado de redundância atual foi assumido. Este tempo para de incrementar quando atinge 65535 ms.
dwApplicationCRC	DWORD	CRC de 32 bits do <i>Projeto Aplicativo</i> , utilizado para detectar diferenças entre os <i>Projeto Aplicativo</i> das duas UCPs.
dwArchiveCRC	DWORD	CRC de 32 bits do <i>Project Archive</i> , utilizado para detectar diferenças entre os <i>Project Archive</i> das duas UCPs.
dwFirmwareVersion	DWORD	Versão de firmware desta UCP, utilizada para verificar compatibilidade entre firmware das duas UCPs.
dwIECTimer	dwIECTimer	A sincronização do <i>IEC Timer</i> é necessária para operação <i>bump-less</i> de alguns blocos de função como TON e TOF. Através deste diagnóstico o <i>IEC Timer</i> da UCP Ativa é recebido e atualizado na UCP Não-Ativa, desde que o serviço Troca de Diagnósticos e Comandos tenha sido executado com sucesso. Sua contagem inicia em 0 e incrementa até 4294967295. Após estouro de contagem, reinicia com valor 0.
wCycleCounter	WORD	Contador de 16 bits utilizado como informação auxiliar de sequência nos <i>Logs de Eventos da Redundância</i> . Na UCP Ativa, é incrementado a cada ciclo da <i>MainTask</i> . Na UCP Não-Ativa, recebe uma cópia do valor existente na UCP Ativa, desde que o serviço Troca de Diagnósticos e Comandos tenha sido executado com sucesso. Sua contagem inicia em 0 e incrementa até 65535. Após estouro de contagem, reinicia com valor 0.

Tabela 219: Diagnósticos Gerais da Redundância

Notas:

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.

A subestrutura “*SyncLinkDiags*” é um array de dois elementos e possui os seguintes campos para diagnósticos dos dois canais de sincronismo (array 1 e 2):

.SyncLinkDiags[1..2].*	Tipo	Descrição
bGeneralFailure	BOOL	TRUE – O canal de sincronismo possui algum tipo de falha. Os 3 próximos diagnósticos indicarão a falha específica. FALSE – O canal de sincronismo está em correto funcionamento.

.SyncLinkDiags[1..2].*	Tipo	Descrição
bInternalFailure	BOOL	TRUE – A falha detectada tem sua causa localizada dentro desta UCP. Tais falhas são tratadas de forma diferenciada. FALSE – Nenhuma falha interna.Nenhuma falha interna.
bLinkDownFailure	BOOL	TRUE – Não há link entre as UCPs. FALSE – O link está operacional.
bTimeoutFailure	BOOL	TRUE – Esta falha é reportada caso um serviço de sincronização não tenha terminado com sucesso até um time-out especificado, e não tenham sido encontradas falhas do tipo bInternalFailure ou bLinkDownFailure que justificassem isso. FALSE – Não houve time-out.

Tabela 220: Diagnósticos Específicos dos Links de Sincronismo

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 Remoto
DG_HX3040_01.RedCmdLoc.bResetNETStatisticsRemote := TRUE;
```

.SyncLinkStatistics.*	Tipo	Descrição
wSuccessExchDgCmdSync	WORD	Contagem de sucessos do serviço Troca de Diagnósticos e Comandos. Sua contagem inicia em com o valor 0, incrementa até 65535, e então reinicia novamente com o valor 0 quando seu valor limite é ultrapassado.
wFailedExchDgCmdSync	WORD	Contagem de falhas do serviço Troca de Diagnósticos e Comandos. Sua contagem inicia em com o valor 0, incrementa até 65535, e então reinicia novamente com o valor 0 quando seu valor limite é ultrapassado.
wSuccessRedDataSync	WORD	Contagem de sucessos do serviço Sincronização de Dados Redundantes. Sua contagem inicia em com o valor 0, incrementa até 65535, e então reinicia novamente com o valor 0 quando seu valor limite é ultrapassado.
wFailedRedDataSync	WORD	Contagem de falhas do serviço Sincronização de Dados Redundantes. Sua contagem inicia em com o valor 0, incrementa até 65535, e então reinicia novamente com o valor 0 quando seu valor limite é ultrapassado.

Tabela 221: Diagnósticos Específicos da Interface

A tabela abaixo mostra todas as mensagens possíveis de serem 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. Para mais informações sobre Log, consulte os [Logs de Eventos da Redundância](#), para mais informações sobre os estados da redundância ver seção [Transições entre Estados de Redundância](#).

String do Estado	Valor do Estado	Mensagem apresentada via Log	Descrição
INIT_APP	0	Redundancy initialization started.	Processo de redundância foi iniciado.
INIT_DONE	1	Redundancy initialization done.	Processo de redundância concluído.
INIT_REDUNDANT_DATA_OVER_MAXIMUM	2	Error: Redundant data is too big.	Área destinada à redundância excede o limite máximo.
INIT_CLUCONF_ERROR	3	CPU is not inserted in the right position for a redundant setup.	A UCP não esta inserida na posição correta para uma configuração redundante.
NCF_LOCAL_CPU_USER_CMD	100	User state change requested via local diagnostic command bit.	Não Configurado para outros estados, comando de troca de estado via bit local de diagnostico.
NCF_REMOTE_CPU_USER_CMD	101	User state change requested via remote diagnostic command bit.	Não Configurado para outros estados, comando de troca de estado via bit remoto de diagnostico.
NCF_LCD_MENU_USER_CMD	102	User state change requested via LCD menu.	Não Configurado para outros estados, comando de troca de estado via menu do LCD.
NCF_AUTO_CONFIGURATION	103	Auto configuration.	Não Configurado para outros estados, auto configuração da redundância.
STR_STANDBY	200	Other CPU is controlling the system.	Inicializando para Reserva, a outra UCP está controlando o sistema.
STR_ACTIVE	201	CPU started controlling the system.	Inicializando para Ativo, UCP iniciou controlando o sistema em Ativo.
STR_CONFIG_ERROR	202	The Application has a configuration error.	Inicializando para Não Configurado, a aplicação possui erro de configuração.
STR_LINK_WITH_INTERNAL_FAILURE	203	Redundant link with an internal failure.	Inicializando para Inativo, canal de redundância interna com falha.
STR_NO_COMM_OTHER_CPU_AND_BUS_IDLE	204	This CPU is unable to communicate with the other CPU via redundant link and the BUS is idle.	Inicializando para Ativo, esta UCP não consegue se comunicar com a outra UCP por canal de redundância e o barramento esta parado.
STR_NO_COMM_OTHER_CPU_AND_BUS_BUSY	205	This CPU is unable to communicate with the other CPU via redundant link and the BUS is busy.	Inicializando para Inativo, esta UCP não consegue se comunicar com a outra UCP por canal de redundância e o barramento esta em uso.
STR_INCOMPATIBLE_FIRMWARE	206	The firmware version of this CPU is not compatible with the other one.	Inicializando para Inativo, a versão de firmware desta UCP não é compatível com a da outra UCP.
STR_APPLICATION_DIFF	207	The Application between the CPUs is different.	Inicializando para Não Configurado, a aplicação entre as UCPs está diferente.
STR_ARCHIVE_DIFF	208	The Project Archive between the CPUs is different.	Inicializando para Não Configurado, o Project Archive entre as UCPs está diferente.
STR_ONLINE_CHANGE_APPLY	209	There is an online change in the Application of the Active CPU.	Inicializando para Não configurado, uma carga online de aplicação foi feita na UCP em Ativo.
STR_DATA_EXCHANGE_FAILED	210	Redundant data synchronization failed.	Inicializando para Inativo, sincronismo de dados redundantes falhou.

String do Estado	Valor do Estado	Mensagem apresentada via Log	Descrição
STR_APP_STATE_DIFF_WITH_OTHER_ACT	211	The application state of this CPU is different from the other Active CPU.	Inicializando para Não Configurado, o estado da aplicação desta UCP é diferente da outra UCP em ativo.
STR_APP_STATE_DIFF_WITH_OTHER_NON_ACT	212	The application state of this CPU is different from the other Non-Active CPU.	Inicializando para Ativo, o estado da aplicação desta UCP é diferente da outra UCP não ativa.
STR_BUS_ERROR	213	Bus error detected.	Inicializando para Inativo, erro de barramento detectado.
STR_OTHER_CPU_ACT_WITH_BRKP	214	Other CPU is Active and with Application stopped in a Breakpoint.	Inicializando para Inativo, A outra UCP está em Ativo e com a aplicação parada em Breakpoint.
SBY_LINK_WITH_INTERNAL_FAILURE	301	Redundant link with an internal failure.	Reserva para Ativo, canal de redundância com falha interma.
SBY_APPLICATION_DIFF	302	The Application between the CPUs is different.	Reserva para Não Configurado, a aplicação entre as UCPs está diferente.
SBY_ARCHIVE_DIFF	303	The Project Archive between the CPUs is different.	Reserva para Não Configurado, o Project Archive entre as UCPs está diferente.
SBY_ONLINE_CHANGE_APPLY	304	There is an online change in the Application of the Active CPU.	Reserva para Não Configurado uma carga online de aplicação foi feita na UCP em Ativo.
SBY_DATA_EXCHANGE_FAILED	305	Redundant data synchronization failed.	Reserva para Inativo, sincronismo de dados redundantes falhou.
SBY_VITAL_FAILURE	306	The other CPU wasn't Active	Reserva para Inativo, a outra UCP não estava Ativa.
SBY_OTHER_PLC_NON_ACTIVE	307		Reserva para Ativo, a outra UCP não estava Ativa.
SBY_DIAGNOSTICS_FAILED	308	Exchange of Redundancy diagnostics failed	Reserva para Inativo, a outra UCP não estava Ativa.
SBY_BUS_ERROR	309	Bus error detected	Reserva para Inativo, erro de barramento detectado a outra UCP está ativa.
SBY_BUS_IDLE	310	BUS in idle state	Reserva para Ativo, a outra UCP não estava Ativa.
SBY_APP_STATE_DIFF_WITH_OTHER_ACT	311	The application state of this CPU is different from the other Active CPU	Reserva para Inativo, o estado da aplicação desta UCP é diferente da outra UCP em ativo.
SBY_LOCAL_CPU_USER_CMD	312	User state change requested via local diagnostic command bit	Outros estados para Reserva, Comando de troca de estado via bit local de diagnóstico.
SBY_REMOTE_CPU_USER_CMD	313	User state change requested via remote diagnostic command bit	Outros estados para Reserva, Comando de troca de estado via bit remoto de diagnóstico.
SBY_LCD_MENU_USER_CMD	314	User state change requested via LCD menu	Outros estados para Reserva, Comando de troca de estado via menu do LCD.
SBY_OTHER_CPU_ACT_WITH_BRKP	315	Other CPU is Active and with Application stopped in a Breakpoint	Reserva, A outra UCP está em Ativo e com a aplicação parada em Breakpoint.
INA_LOCAL_CPU_USER_CMD	401	User state change requested via local diagnostic command bit	Outros estados para Inativo, comando de troca de estado via bit local de diagnóstico.

String do Estado	Valor do Estado	Mensagem apresentada via Log	Descrição
INA_REMOTE_CPU_USER_CMD	402	User state change requested via remote diagnostic command bit	Outros estados para Inativo, comando de troca de estado via bit remoto de diagnóstico.
INA_LCD_MENU_USER_CMD	403	User state change requested via LCD menu	Outros estados para Inativo, Comando de troca de estado via menu do LCD.
ACT_LINK_WITH_INTERNAL_FAILURE	501	Redundancy internal link failure for one link	Houve erro de um canal interno de redundância.
ACT_BOTH_LINKS_WITH_INTERNAL_FAILURE	502	Redundancy internal link failure for both links	Houve erro em ambos os canais internos de redundância.
ACT_CONFLICT_ACTIVE	503	Both CPUs are in active redundancy state	Ativo para Reserva, ambas as UCPs estão em ativo.
ACT_VITAL_FAILURE	504	Both CPUs are in active redundancy state and Application state is equal	Ativo para Inativo, ambas as UCPs estão em ativo e com a aplicação no mesmo estado.
ACT_CONFLICT_ACTIVE_APP_STATE_EQUAL	505		Ativo para Reserva, ambas as UCPs estão em ativo e com a aplicação no mesmo estado.
ACT_CONFLICT_ACTIVE_APP_STATE_DIFF	506		
ACT_BUS_ERROR	507	Bus error detected	Ativo para Inativo, erro de barramento detectado a outra UCP está em Reserva.
ACT_LOCAL_CPU_USER_CMD	508	User state change requested via local diagnostic command bit	Outros estados para Ativo, comando de troca de estado via bit local de diagnóstico.
ACT_REMOTE_CPU_USER_CMD	509	User state change requested via remote diagnostic command bit	Outros estados para Ativo, comando de troca de estado via bit remoto de diagnóstico.
ACT_LCD_MENU_USER_CMD	510	User state change requested via LCD menu	Outros estados para Ativo, Comando de troca de estado via menu do LCD.
ACT_APP_DELETE	511	Application of this CPU was deleted	A aplicação desta UCP foi apagada.
ACT_APP_RESET	512	Application of this CPU was reset	A Aplicação desta UCP sofreu reset.
ACT_APP_EXCEPTION	513	Application of this CPU entered in exception state	A aplicação desta UCP entrou em estado de exceção.
NON_ACT_OTHER_CPU_APP_DELETED_BY_USER	600	Application of the other CPU was deleted	A aplicação da outra UCP foi apagada.
NON_ACT_OTHER_CPU_APP_RESET_BY_USER	601	Application of the other CPU was reset	A Aplicação da outra UCP sofreu reset.
NON_ACT_OTHER_CPU_APP_BRKP_BY_USER	602	Application of the other CPU is stopped by breakpoint	Aplicação da outra UCP está em breakpoint.

Tabela 222: Razões de Troca de Estados da Redundância

6.6.3.2. 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:

- 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.
- 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.

ATENÇÃO

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.

A seguir, são definidos os campos das estruturas *RedCmdLoc* e *RedCmdRem*:

.RedCmdLoc.* .RedCmdRem.*	Tipo	Descrição
bAutoConfigLocal	BOOL	TRUE – Este diagnóstico informa uma solicitação de configuração automática, necessária para deixar o estado Não-Configurado em algumas situações. FALSE – Solicitação de configuração automática desabilitada.
bStandbyLocal	BOOL	TRUE – Este comando produz uma ação equivalente à solicitação de “Trocar para Reserva” do menu Redundancia, disponível no visor da UCP. FALSE – Não há solicitação pendente.
bMenuStandbyLocal	BOOL	TRUE – Liga por um ciclo da <i>MainTask</i> , indicando que o usuário solicitou um comando de “Trocar para Reserva” através do Menu do Visor da UCP. FALSE – Não há solicitação pendente.
bInactiveLocal	BOOL	TRUE – Este comando produz uma ação equivalente à solicitação de “Trocar para Inativo” do menu Redundancia, disponível no visor da UCP. FALSE – Não há solicitação pendente.

.RedCmdLoc.* .RedCmdRem.*	Tipo	Descrição
bMenuInactiveLocal	BOOL	TRUE – Liga por um ciclo da MainTask, indicando que o usuário solicitou um comando de “Trocar para Inativo” através do Menu do Visor da UCP. FALSE – Não há solicitação pendente.
bStandbyRemote	BOOL	TRUE – Este comando produz uma ação equivalente à solicitação de “Trocar para Reserva” do menu Redundância, disponível no visor da UCP Remota. FALSE – Não há solicitação pendente.
bInactiveRemote	BOOL	TRUE – Este comando produz uma ação equivalente à solicitação de “Trocar para Inativo” do menu Redundância, disponível no visor da UCP remota. FALSE – Não há solicitação pendente.
bResetNETStatisticsLocal	BOOL	TRUE – Este comando reseta as estatísticas dos canais de sincronismo de redundância A/B (veja subestrutura SyncLinkStatistics em RedDgnLoc e RedDgnRem). Tais estatísticas são contadores de falhas e sucessos em serviços de sincronização. FALSE – O comando de reset das estatísticas dos canais de sincronismo A/B na UCP local não foi acionado.
bResetNETStatisticsRemote	BOOL	TRUE – Produz uma ação equivalente ao comando ResetNETStatisticsLocal, porém desta vez na UCP remota. FALSE – O comando de reset das estatísticas dos canais de sincronismo A/B na UCP remota não foi acionado.

Tabela 223: Comandos da Redundância

6.6.3.3. 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), já discutidos na seção [Estrutura de Diagnósticos da Redundância](#).
- Comandos de Redundância (RedCmdLoc e RedCmdRem), já discutidos na seção [Comandos da Redundância](#).
- Informações do Usuário Trocados entre UCPA e UCPB (RedUsrLoc e RedUsrRem), que serão discutidos nesta seção.

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*.

6.6.4. 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, registram 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
- 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 “*Device(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.

ATENÇÃO

Alguns diagnósticos podem apontar possíveis falhas durante a inicialização do sistema redundante e nos primeiros ciclos das tarefas. Mas, em um correto funcionamento do sistema, estes diagnósticos voltam a indicar a ausência de erros logo após a inicialização do sistema.

7. Manutenção

ATENÇÃO

Para uma manutenção adequada e com segurança, deve-se consultar primeiramente a seção [Segurança Elétrica](#).

7.1. 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 proporcionam 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. A seção de Manutenção deve ser consultada pelo usuário sempre que necessário.

7.1.1. 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 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, descrito na seção [Menu Informativo e de Configuração da UCP](#).

A figura abaixo mostra a localização da tecla na UCP:

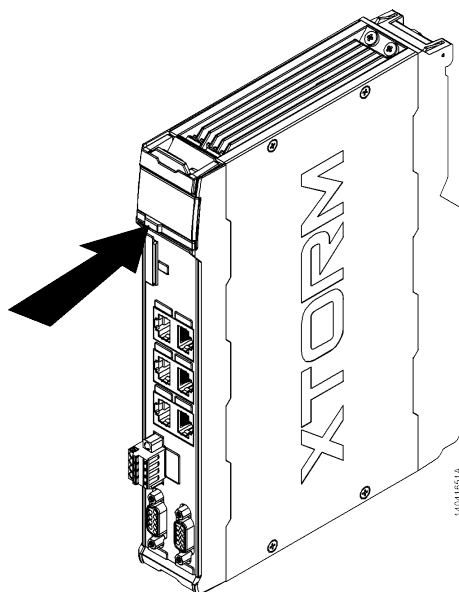


Figura 224: Botão de Diagnóstico

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 dos mesmos, no mesmo modelo da apresentação dos diagnósticos da UCP.

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

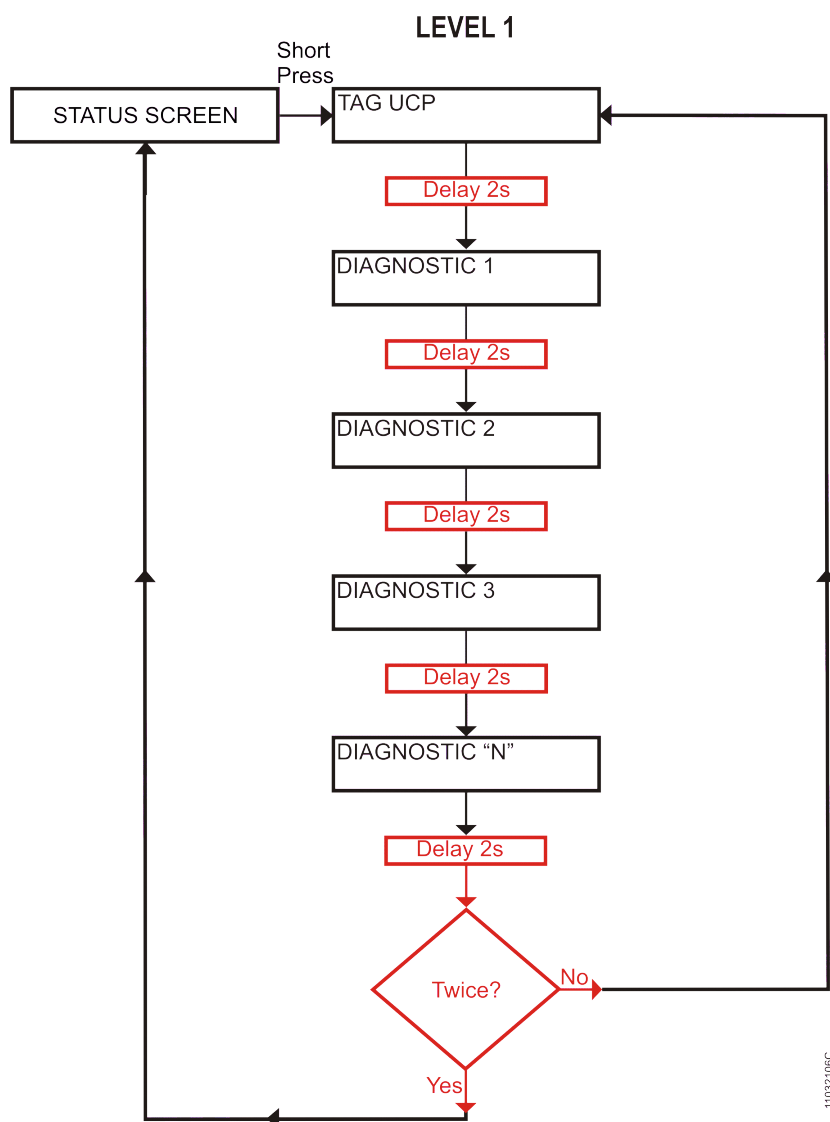


Figura 225: Visualização dos Diagnósticos da UCP

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 [Menu Informativo e de Configuração da UCP](#).

A tabela abaixo mostra a diferença entre os tempos do pressionamento curto, pressionamento longo e botão preso.

Tipo de Pressionamento	Tempo Mínimo	Tempo Máximo	Condição para Indicação
Sem pressionamento	-	59,99 ms	-
Pressionamento curto	60 ms	0,99 s	Pressionamento e liberação do botão dentro do período definido
Pressionamento longo	1 s	20 s	Pressionamento e liberação do botão dentro do período definido
Botão preso	20,01 s	(∞)	Pressionamento do botão por mais de 20 segundos

Tabela 224: Tempos de Pressionamento Botão de Diagnóstico

As mensagens exibidas no visor gráfico da UCP Hadron Xtorm, correspondentes aos diagnósticos, estão descritas na seção [Diagnósticos via Variáveis](#), na tabela 226 e nas tabelas de Diagnósticos Detalhados logo abaixo.

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.

ATENÇÃO

One Touch Diag (OTD): Essa opção somente estará disponível ao usuário quando o módulo estiver em modo operacional.

7.1.2. Diagnósticos via LED

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

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

Tabela 225: Descrição dos Estados do LED de Diagnóstico

Nota:

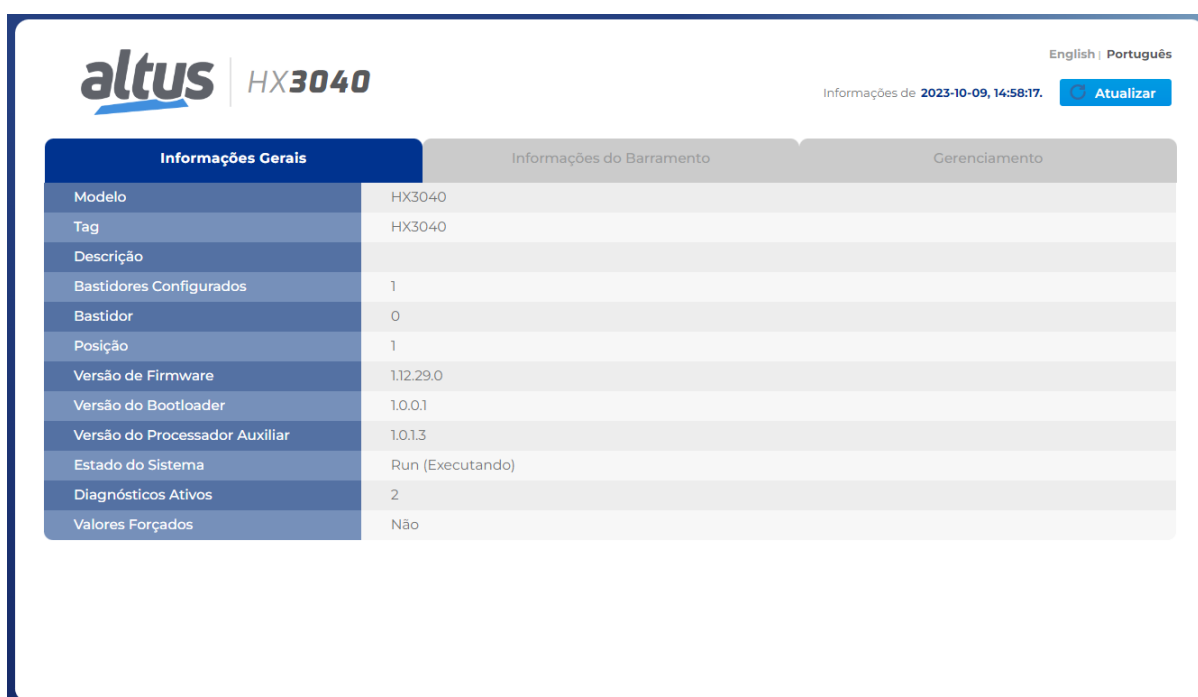
Cão-de-guarda de Software: Para remover a indicação de cão-de-guarda, deve-se efetuar um reset da aplicação ou desligar e ligar novamente a UCP. Esse cão-de-guarda ocorre quando o tempo de execução da aplicação de usuário for maior que o tempo de cão-de-guarda configurado.

7.1.3. 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 abaixo:



The screenshot shows the Altus HX3040 web interface. At the top left is the Altus logo and 'HX3040'. At the top right, there are language options 'English | Português' and a timestamp 'Informações de 2023-10-09, 14:58:17.' with an 'Atualizar' button. Below this is a table with three tabs: 'Informações Gerais' (selected), 'Informações do Barramento', and 'Gerenciamento'. The 'Informações Gerais' tab contains the following data:

Informações Gerais	
Modelo	HX3040
Tag	HX3040
Descrição	
Bastidores Configurados	1
Bastidor	0
Posição	1
Versão de Firmware	1.12.29.0
Versão do Bootloader	1.0.0.1
Versão do Processador Auxiliar	1.0.1.3
Estado do Sistema	Run (Executando)
Diagnósticos Ativos	2
Valores Forçados	Não

Figura 226: Tela Inicial

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



Figura 227: 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.

ATENÇÃO

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

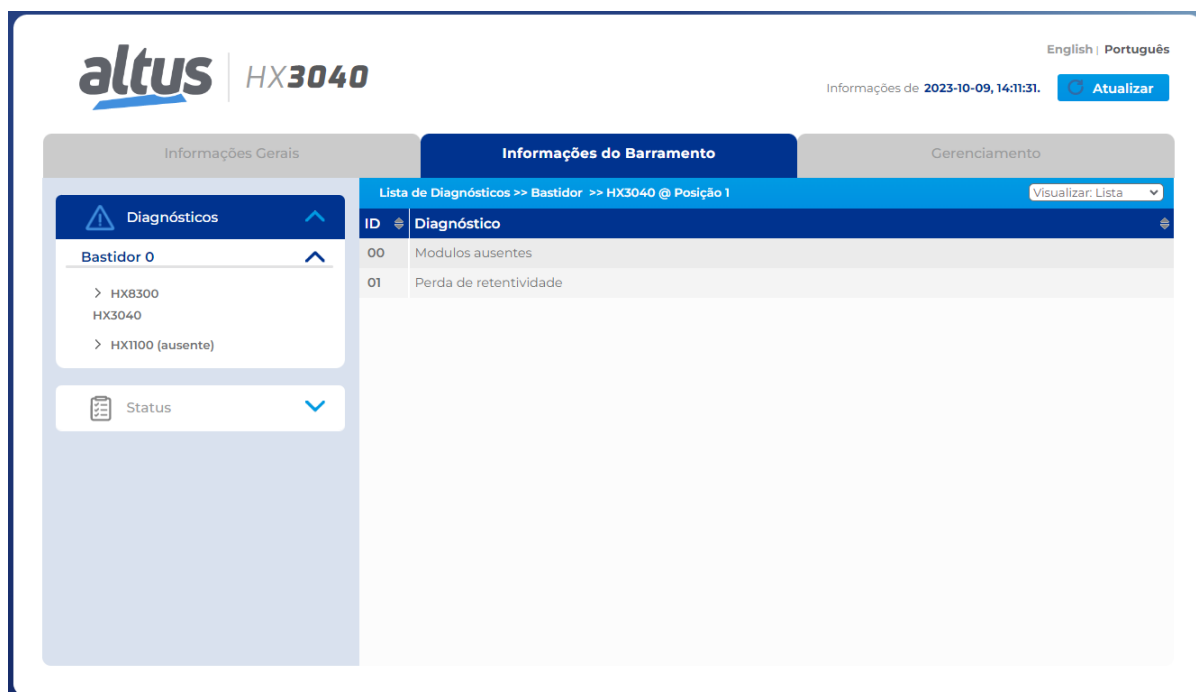


Figura 228: Diagnósticos do Sistema

Caso a aba *Status* seja selecionada, o estado de todos os diagnósticos detalhados é exibido na tela, conforme mostra a figura abaixo:

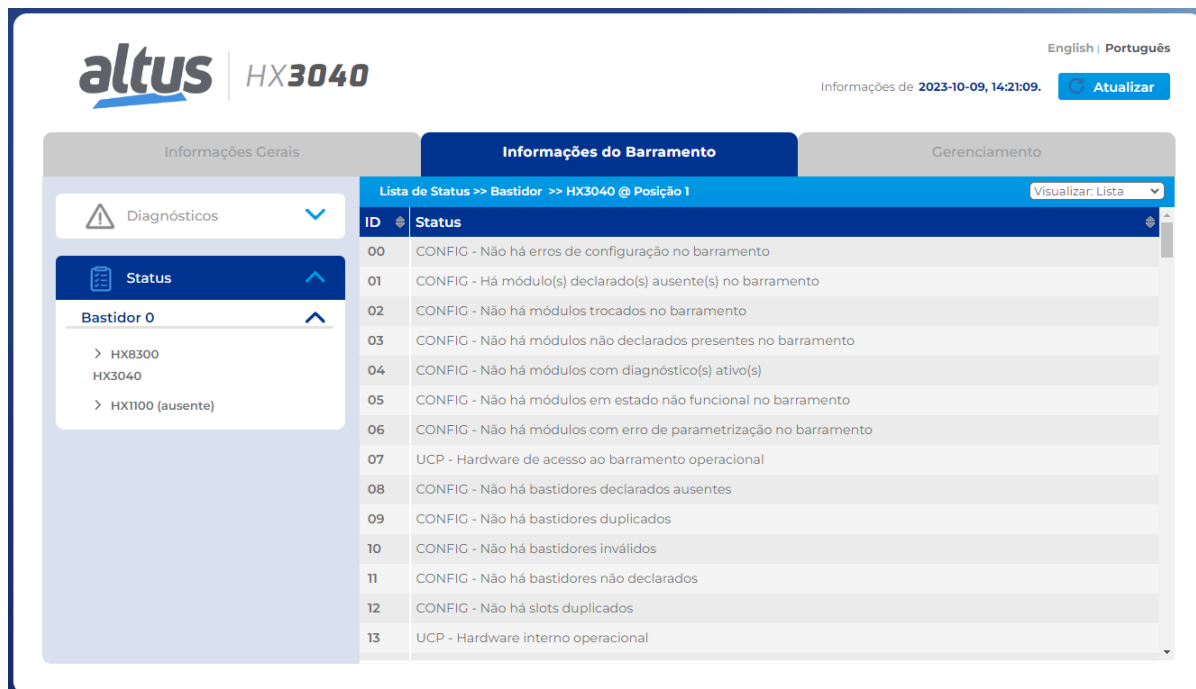


Figura 229: Status do Sistema

O usuário pode optar por duas opções de idioma: Português e Inglês. Basta alterar na parte superior direita da tela para o idioma desejado.

7.1.4. Diagnósticos via Variáveis

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.

7.1.4.1. Diagnósticos Resumidos

A tabela abaixo mostra o significado de cada das variáveis de diagnósticos resumidos da UCP:

Mensagem de Diagnóstico	DG_modulename .tSummarized.*	Tipo	Descrição
SEM DIAG	-	BOOL	Não existe diagnóstico ativo.
CONFIG. INCOMPATIVEL	bConfigMismatch	BOOL	TRUE – Existe algum problema de configuração no barramento, como módulo em posição incorreta. FALSE – O barramento está configurado corretamente.
MODULOS AUSENTES	bAbsentModules	BOOL	TRUE – Um ou mais módulos declarados estão ausentes. FALSE – Todos os módulos estão presentes no barramento.
MODULOS TROCADOS	bSwappedModules	BOOL	TRUE – Dois módulos estão trocados entre si no barramento. FALSE – Não há módulos trocados no barramento.
MODULOS NÃO DECLARADOS	bNonDeclaredModules	BOOL	TRUE – Um ou mais módulos presentes no barramento não estão declarados. FALSE – Todos os módulos presentes no barramento estão declarados.
MODULOS C/ DIAGNOSTICO	bModulesWithDiagnostic	BOOL	TRUE – Um ou mais módulos do barramento estão com diagnóstico ativo. FALSE – Não existem diagnósticos ativos nos módulos do barramento.
MODULOS C/ ERRO FATAL	bModuleFatalError	BOOL	TRUE – Um ou mais módulos presentes no barramento estão em estado não funcional. FALSE – Todos os módulos presentes no barramento estão em estado funcional.
MODULOS C/ ERRO PARAM.	bModuleParameterError	BOOL	TRUE – Um ou mais módulos do barramento estão com erro de parametrização. FALSE – Todos os módulos estão parametrizados.
ERRO NO BARRAMENTO	bWHSBBusError	BOOL	TRUE – Indicação do mestre que existe falha no barramento WHSB. FALSE – O barramento WHSB está em correto funcionamento.
BASTIDOR AUSENTE	bAbsentRacks	BOOL	TRUE – Um ou mais bastidores declarados estão ausentes. FALSE – Todos os bastidores declarados estão presentes.

Mensagem de Diagnóstico	DG_modulename .tSummarized.*	Tipo	Descrição
BASTIDOR DUPLICADO	bDuplicatedRacks	BOOL	TRUE – Há algum bastidor com o número de identificação duplicado. FALSE – Não existem bastidores com o número de identificação duplicado.
BASTIDOR INVALIDO	bInvalidRacks	BOOL	TRUE – Há algum bastidor com o número de identificação inválido. FALSE – Não existem bastidores com o número de identificação inválido.
BASTIDOR NÃO DECLARADO	bNonDeclaredRacks	BOOL	TRUE – Há algum bastidor com o número de identificação não declarado. FALSE – Não existem bastidores com o número de identificação não declarado.
SLOT DUPLICADO	bDuplicatedSlots	BOOL	TRUE – Há algum endereço de posição duplicado. FALSE – Não existem endereços de posição duplicados.
-	bReserved_13..15	BOOL	Reservado.
FALHA DE HARDWARE	bHardwareFailure	BOOL	TRUE – Falha no hardware da UCP. FALSE – O hardware está em correto funcionamento.
EXCECAO NO SOFTWARE	bSoftwareException	BOOL	TRUE – Uma ou mais exceções geradas pelo software. FALSE – Não foram geradas exceções no software.
-	bReserved_18	BOOL	Reservado.
ERRO CARTAO DE MEMORIA	bMemoryCardError	BOOL	TRUE – O cartão de memória está inserido na UCP, porém não está funcionando corretamente. FALSE – O cartão de memória está funcionando corretamente.
-	bReserved_20..23	BOOL	Reservado.
ERRO CONFIG. COM 1	bCOM1ConfigError	BOOL	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface serial COM 1. FALSE – A configuração da interface serial COM 1 está correta.
ERRO CONFIG. COM 2	bCOM2ConfigError	BOOL	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface serial COM 2. FALSE – A configuração da interface serial COM 2 está correta.
ERRO CONFIG. NET 1	bNET1ConfigError	BOOL	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface Ethernet NET 1. FALSE – A configuração da interface Ethernet NET 1 está correta.
ERRO CONFIG. NET 2	bNET2ConfigError	BOOL	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface Ethernet NET 2. FALSE – A configuração da interface Ethernet NET 2 está correta.
ERRO CONFIG. NET 3	bNET3ConfigError	BOOL	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface Ethernet NET 3. FALSE – A configuração da interface Ethernet NET 3 está correta.

Mensagem de Diagnóstico	DG_modulename .tSummarized.*	Tipo	Descrição
ERRO CONFIG. NET 4	bNET4ConfigError	BOOL	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface Ethernet NET 4. FALSE – A configuração da interface Ethernet NET 4 está correta.
ERRO CONFIG. NET 5	bNET5ConfigError	BOOL	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface Ethernet NET 5. FALSE – A configuração da interface Ethernet NET 5 está correta.
ERRO CONFIG. NET 6	bNET6ConfigError	BOOL	TRUE – Ocorreu algum erro durante, ou após, a configuração da interface Ethernet NET 6. FALSE – A configuração da interface Ethernet NET 6 está correta.
DATA/HORA INVALIDA	bInvalidDateTime	BOOL	TRUE – A data ou a hora são inválidas. FALSE – A data e a hora estão corretas.
RUNTIME RESET	bRuntimeReset	BOOL	TRUE – O RTS (Runtime System) foi reiniciado pelo menos uma vez. Esse diagnóstico somente é limpo na reinicialização do sistema. FALSE – O RTS (Runtime System) está operando normalmente.
ERRO TECLA OTD	bOTDSwitchError	BOOL	TRUE – A tecla ficou travada por mais de 20 s pelo menos uma vez enquanto a UCP estava energizada. Esse diagnóstico somente é limpo na reinicialização do sistema. FALSE –
PERDA DE RETENTIVIDADE	bRetentivityLost	BOOL	TRUE – Dados inválidos encontrados na memória retentiva durante a inicialização.
UCP POSICAO INCORRETA	bWrongCPUSlot	BOOL	TRUE – UCP posicionada em posição diferente da configurada via programador FALSE – UCP posicionada na posição correta, conforme configurada via programador.
-	bReserved_37..39	BOOL	Reservado.

Tabela 226: Tabela dos Diagnósticos Resumidos da UCP

Notas:

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 o mesmo 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 a mesma apresenta problemas no RTC, processador auxiliar, ou outros recursos de hardware.

Exceção no software: Caso o diagnóstico de exceção no software seja verdadeiro, o usuário deverá verificar a sua aplicação para garantir que a mesma não esteja acessando indevidamente a memória. Se o problema persistir, o setor de Suporte da Altus deverá ser consultado. Os códigos de exceção no software estão descritos após a tabela de diagnósticos detalhados da UCP.

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.

7.1.4.2. Diagnósticos Detalhados

As tabelas a seguir mostram os diagnósticos detalhados da CPU Hadron Xtorm, para sua consulta é importante verificar as observações abaixo:

- **Visualização das Estruturas de Diagnóstico:** As Estruturas de Diagnóstico adicionadas ao projeto podem ser visualizadas acessando o item “*Library Manager*” na árvore de dispositivos da janela do MasterTool Xtorm. Com isso, é possível visualizar todos os tipos de dados definidos na estrutura.
- **Contadores:** Todos os contadores dos diagnósticos da UCP retornam à zero quando o seu valor limite é ultrapassado.

Variável DG_HX3040.tDetailed.*	Tipo	Descrição
Target. dwCPUModel	DWORD	HX3040 = 0x3040.
Target. abyCPUVersion	BYTE ARRAY(4)	Versão do firmware.
Target. abyBootloaderVersion	BYTE ARRAY(4)	Versão do bootloader.
Target. abyAuxprocVersion	BYTE ARRAY(4)	Versão do processador auxiliar.

Tabela 227: Diagnósticos Detalhados Grupo Target

Variável DG_HX3040.tDetailed.*	Tipo	Descrição
Hardware. bAuxprocFailure	BIT	Falha na comunicação entre o processador auxiliar e o processador principal.
Hardware. bRTCFailure	BIT	O processador principal não está habilitado para comunicar com o RTC (relógio da UCP).
Hardware. bThermometerFailure	BIT	Falha na comunicação entre o termômetro e o processador principal.
Hardware. bLCDFailure	BIT	Falha na comunicação entre o visor LCD e o processador principal.

Tabela 228: Diagnósticos Detalhados Grupo Hardware

Variável DG_HX3040.tDetailed.*	Tipo	Descrição
Exception. wExceptionCode	WORD	Código de Exceção gerado pelo RTS, o significado dos códigos pode ser conferido na tabela 230.
Exception. byProcessorLoad	BYTE	Nível, em percentagem (%), de carga no processador.

Tabela 229: Diagnósticos Detalhados Grupo Exception

Código	Descrição	Código	Descrição
0x0000	Não há código de exceção.	0x0051	Violação de acesso.
0x0010	Tempo de Cão-de-Guarda da tarefa IEC expirado (Cão-de-Guarda de Software).	0x0052	Instrução privilegiada.
0x0012	Erro de configuração de E / S.	0x0053	Falha na página.
0x0013	Erros de check-up após o download do programa.	0x0054	Estouro de pilha.
0x0014	Erro de rede de campo.	0x0055	Disposição inválida.
0x0015	Erro de atualização de E / S.	0x0056	Manobra inválida.
0x0016	Tempo de ciclo (execução) excedido.	0x0057	Página protegida.
0x0017	Atualização da aplicação Online é muito grande.	0x0058	Falha dupla.
0x0018	Referências externas não resolvidas.	0x0059	OpCode inválido.
0x0019	Download rejeitado.	0x0100	Desalinhamento do tipo de dado.
0x001A	Projeto não-carregado, pois as variáveis retentivas não podem ser realocadas.	0x0101	Limite de arrays excedido.
0x001B	Projeto não-carregado e excluído.	0x0102	Divisão por zero.
0x001C	Fora da pilha de memória.	0x0103	Overflow.
0x001D	Memória retentiva corrompida ou não pode ser mapeada.	0x0104	Não continuável.

Código	Descrição	Código	Descrição
0x001E	Projeto pode ser carregado, mas causa uma queda posteriormente.	0x0105	Cão-de-guarda na carga do processador de todas as tarefas IEC detectadas.
0x0021	Target da aplicação de inicialização não corresponde ao target atual.	0x0150	FPU: Erro não especificado.
0x0022	Erro nas tarefas escalonadas. Check-up do arquivo baixado não corresponde.	0x0151	FPU: Operando não normal.
0x0024	Identidade retentiva não corresponde com a atual identidade da aplicação de inicialização.	0x0153	FPU: Resultado inexato.
0x0025	Falha na configuração da tarefa IEC.	0x0154	FPU: Operação inválida.
0x0026	Aplicação está executando com target errado.	0x0155	FPU: Overflow.
0x0050	Instrução ilegal.	0x0156	FPU: Verificação da pilha.
		0x0157	FPU: Underflow.

Tabela 230: Códigos de Exceção RTS

Variável DG_HX3040.tDetailed.*	Tipo	Descrição
RetainInfo. byCPUInitStatus	BYTE	Status da Inicialização da UCP: 01: Hot start 02: Warm Start 03: Cold Start Obs.: Essas variáveis são reinicializadas em todas as energizações.
RetainInfo. wCPUColdStartCounter	WORD	Contador de Inicializações à frio: Incrementa caso o CP perca retentividade durante a inicialização. (0 a 65535).
RetainInfo. wCPUWarmStartCounter	WORD	Contador de Inicializações à quente: Incrementa caso o CP inicialize com dados retentivos válidos. (0 a 65535).
RetainInfo. wCPUHotStartCounter	WORD	Contador de distúrbios menores do que o tempo de suporte a falhas na alimentação da UCP. (0 a 65535)
RetainInfo. wRTSResetCounter	WORD	Contador de reset efetuados pelo RTS (Runtime System). (0 a 65535)

Tabela 231: Diagnósticos Detalhados Grupo RetainInfo

Variável DG_HX3040.tDetailed.*	Tipo	Descrição
Reset. bBrownOutReset	BIT	A UCP foi reiniciada devido a uma falha na alimentação na última inicialização.
Reset. bWatchdogReset	BIT	UCP foi reiniciada devido ao cão-de-guarda ativo na última inicialização.

Tabela 232: Diagnósticos Detalhados Grupo Reset

Nota:

Reset por Brownout: O diagnóstico de reset por brownout somente será verdadeiro quando a alimentação da fonte exceder o limite mínimo exigido nas características técnicas da mesma, mantendo-se com tensão baixa, ou seja, sem sofrer uma interrupção. A UCP irá identificar a queda da alimentação e indicará o diagnóstico de falha na alimentação. Quando a tensão for restabelecida, a UCP será reinicializada automaticamente e indicará o diagnóstico de reset por brownout.

Variável DG_HX3040.tDetailed.*	Tipo	Descrição
Thermometer. bOverTemperatureAlarm	BIT	Alarme gerado devido à temperatura interna estar em 85°C ou acima de 85°C.
Thermometer. bUnderTemperatureAlarm	BIT	Alarme gerado devido à temperatura interna estar em 0°C ou abaixo de 0°C.
Thermometer. diTemperature	DINT	Temperatura lida no sensor interno da UCP, medida em graus Celsius, tendo sua resolução dada em 1°C.

Tabela 233: Diagnósticos Detalhados Grupo Thermometer

Variável DG_HX3040.tDetailed.*	Tipo	Descrição
Serial.COM[1-2]. byProtocol	ENUM (BYTE)	Protocolo selecionado na COM[1-2]: NO_PROTOCOL (0): Sem protocolo; MODBUS_RTU_MASTER (1): MODBUS RTU Mestre; MODBUS_RTU_SLAVE (2): MODBUS RTU Escravo; OTHER_PROTOCOL (3): Outro protocolo;
Serial.COM[1-2]. dwRXBytes	DWORD	Contador de caracteres recebidos através da COM[1-2]. (0 a 4294967295)
Serial.COM[1-2]. dwTXBytes	DWORD	Contador de caracteres transmitidos através da COM[1-2]. (0 a 4294967295)
Serial.COM[1-2]. wRXPendingBytes	WORD	Número de caracteres pendentes no buffer de leitura na COM[1-2]. (0 a 65535)
Serial.COM[1-2]. wTXPendingBytes	WORD	Número de caracteres pendentes no buffer de transmissão na COM[1-2]. (0 a 65535)
Serial.COM[1-2]. wBreakErrorCounter	WORD	Esses contadores serão reiniciados nas seguintes condições: Energização; Configuração da porta serial COM[1-2]; Remoção das filas RX e TX;
Serial.COM[1-2]. wParityErrorCounter	WORD	
Serial.COM[1-2]. wFrameErrorCounter	WORD	
Serial.COM[1-2]. wRXOverrunCounter	WORD	

Tabela 234: Diagnósticos Detalhados Grupo Serial COM 1 e COM 2

Nota:

Contador de erro de paridade: Quando a UCP estiver configurada com paridade *Sem Paridade*, o contador de erros de paridade não é incrementado caso receba uma paridade diferente. Nesse caso, será indicado erro de frame. O valor máximo de cada contador é 65535.

Variável DG_HX3040.tDetailed.*	Tipo	Descrição
Ethernet.NET[1-6]. bLinkDown	BIT	Indica o estado do link na interface.
Ethernet.NET[1-6]. byOperatingMode	ENUM(BYTE)	Modo de operação da interface: STANDALONE (0): Modo normal. NIC_TEAMING (1): Modo NIC Teaming com outra interface. SWITCH (2): Modo switch.
Ethernet.NET[1-6]. byOperatingState	ENUM(BYTE)	Estado da interface: ERROR_UNCONFIGURED (0): Inter- face não configurada. DISABLED (1): Interface desabilitada. CONFIGURED_AND_RUNNING (2): Interface configurada e em funciona- mento.
Ethernet.NET[1-6]. wProtocol	WORD	Protocolo selecionado na NET[1-6]: 00: Sem protocolo.
Ethernet.NET[1-6]. wProtocol. bMODBUS_RTU_ETH_Client	BIT	MODBUS RTU via TCP Cliente.
Ethernet.NET[1-6]. wProtocol. bMODBUS_ETH_Client	BIT	MODBUS TCP Cliente.
Ethernet.NET[1-6]. wProtocol. bMODBUS_RTU_ETH_Server	BIT	MODBUS RTU via TCP Servidor.
Ethernet.NET[1-6]. wProtocol. bMODBUS_ETH_Server	BIT	MODBUS TCP Servidor.
Ethernet.NET[1-6]. wProtocol. bDNP3_Client	BIT	DNP3 Cliente.
Ethernet.NET[1-6]. wProtocol. bDNP3_Server	BIT	DNP3 Servidor.
Ethernet.NET[1-6]. wProtocol. bIEC61850_Server	BIT	IEC 61850 Servidor.
Ethernet.NET[1-6]. szIP	STRING(15)	Endereço IP da porta.
Ethernet.NET[1-6]. szMask	STRING(15)	Máscara de Subrede da porta.
Ethernet.NET[1-6]. szGateway	STRING(15)	Endereço Gateway da porta.
Ethernet.NET[1-6]. szMAC	STRING(17)	Endereço MAC da porta.
Ethernet.NET[1-6]. abyIP	BYTE ARRAY(4)	Endereço IP da porta.
Ethernet.NET[1-6]. abyMask	BYTE ARRAY(4)	Máscara de Subrede da porta

Variável DG_HX3040.tDetailed.*	Tipo	Descrição
Ethernet.NET[1-6]. abyGateway	BYTE ARRAY(4)	Endereço Gateway da porta.
Ethernet.NET[1-6]. abyMAC	BYTE ARRAY(6)	Endereço MAC da porta.
Ethernet.NET[1-6]. NICteaming. bStandbyState	BIT	Interface está em estado de espera.
Ethernet.NET[1-6]. NICteaming. bActiveState	BIT	Interface está em estado ativo.
Ethernet.NET[1-6]. NICteaming. bLinkDown	BIT	Interface sem link.
Ethernet.NET[1-6]. NICteaming. bInterMsgTimeout	BIT	Time-out na espera por pacotes do par NIC Teaming.
Ethernet.NET[1-6]. NICteaming. bGeneralRxMsgTimeout	BIT	Time-out na espera por pacotes da red.
Ethernet.NET[1-6]. dwTransmittedBytes	DWORD	Contador de bytes enviados através da interface. (0 a 4294967295)
Ethernet.NET[1-6]. dwTransmittedPackets	DWORD	Contador de pacotes enviados através da interface. (0 a 4294967295)
Ethernet.NET[1-6]. dwTransmittedDropErrors	DWORD	Contador de perdas de conexão na transmissão através da interface. (0 a 4294967295)
Ethernet.NET[1-6]. dwTransmittedCollisionErrors	DWORD	Contador de erros de colisão na transmissão através da interface. (0 a 4294967295)
Ethernet.NET[1-6]. dwReceivedBytes	DWORD	Contador de bytes recebidos através da porta. (0 a 4294967295)
Ethernet.NET[1-6]. dwReceivedPackets	DWORD	Contador de pacotes recebidos através da porta. (0 a 4294967295)
Ethernet.NET[1-6]. dwReceivedDropErrors	DWORD	Contador de perdas de conexão na recepção através da interface. (0 a 4294967295)
Ethernet.NET[1-6]. dwReceivedFrameErrors	DWORD	Contador de erros de frame na recepção através da interface. (0 a 4294967295)

Tabela 235: Diagnósticos Detalhados Grupo Ethernet NET 1 .. NET 6

Variável DG_HX3040.tDetailed.*	Tipo	Descrição
UserFiles. byMounted	BYTE	Indica se a memória utilizada para gravar arquivos de usuário está apta para receber os dados.
UserFiles. dwFreeSpacekB	DWORD	Espaço livre da memória de arquivos de usuário em Kbytes.
UserFiles. dwTotalSizekB	DWORD	Capacidade de armazenamento da memória de arquivos de usuário em Kbytes.

Tabela 236: Diagnósticos Detalhados Grupo UserFiles

Variável DG_HX3040.tDetailed.*	Tipo	Descrição
UserLogs. byMounted	BYTE	Status da memória em que são inseridos os logs de usuário.
UserLogs. wFreeSpacekB	WORD	Espaço livre na memória de logs de usuário em Kbytes.
UserLogs. wTotalSizekB	WORD	Capacidade de armazenamento da memória de logs de usuário em Kbytes.

Tabela 237: Diagnósticos Detalhados Grupo UserLogs

Variável DG_HX3040.tDetailed.*	Tipo	Descrição
MemoryCard. byMounted	ENUM(BYTE)	Status do Cartão de Memória: NOT_MOUNTED (0): Cartão de memória não montado MOUNTED (1): Cartão de memória inserido e montado.
MemoryCard. bMemcardtoCPUEnabled	BIT	Nível de Proteção do Cartão de Memória: Leitura de dados do cartão de memória pela UCP autorizada.
MemoryCard. bCPUtoMemcardEnabled	BIT	Escrita de dados no Cartão de Memória pela UCP autorizada.
MemoryCard. dwFreeSpacekB	DWORD	Espaço livre no Cartão de Memória em Kbytes.
MemoryCard. dwTotalSizekB	DWORD	Capacidade de armazenamento do Cartão de Memória em Kbytes.

Tabela 238: Diagnósticos Detalhados Grupo MemoryCard

Variável DG_HX3040.tDetailed.*	Tipo	Descrição
WHSB. byHotSwapAndStartupStatus	ENUM(BYTE)	Informa a situação anormal no barramento que ocasionou a parada da aplicação, para cada <i>Modo de troca a quente</i> . Consulte a tabela 240 para detalhes das possibilidades.
WHSB. adwRackIOErrorStatus	DWORD ARRAY(32)	Identificação de erros em módulos de E/S, individualmente. Para mais informações sobre este diagnóstico consultar a nota abaixo.
WHSB. adwModulePresenceStatus	DWORD ARRAY(32)	Status de presença de módulos declarados de E/S em barramentos, individualmente. Para mais informações sobre este diagnóstico consultar a nota abaixo.
WHSB. byWHSBBusErrors	BYTE	Contador de falhas no barramento WHSB. Esse contador é reinicializado na energização (0 a255).

Tabela 239: Diagnósticos Detalhados Grupo WHSB

Notas:

Diagnóstico de erro dos módulos do barramento: Cada DWORD do Array deste diagnóstico representa um bastidor, cujas posições são representadas pelos bits destas DWORDs. Logo, o Bit-0 da DWORD-0 equivale a posição zero do bastidor com endereço zero. Cada um desses Bits é o resultado de uma operação lógica OU entre os diagnósticos de configuração incompatível (*bConfigMismatch*), módulos ausentes (*bAbsentModules*), módulos trocados (*bSwappedModules*), módulos com erro fatal (*bModuleFatalError*) e o estado operacional do módulo de uma determinada posição.

Status de presença de módulos: Cada DWORD do Array deste diagnóstico representa um bastidor, cujas posições são representadas pelos bits destas DWORDs. Logo, o Bit-0 da DWORD-0 equivale a posição zero do bastidor com endereço zero. Então, se o módulo estiver presente, este bit será verdadeiro. É importante ressaltar que esse diagnóstico é válido para todos os módulos, exceto fontes de alimentação e UCPs, ou seja, não apresentam a presença no barramento em suas respectivas posições (bit permanece em falso).

Situações que ocasionam parada da aplicação: Os códigos das possíveis situações que ocasionam parada da aplicação podem ser consultados abaixo:

Código	Enumerável	Descrição
00	INITIALIZING	Esse estado é apresentado enquanto os demais estados não estiverem prontos.
01	RESET_WATCHDOG	Aplicação em Modo Stop devido ao reset por cão-de-guarda de hardware ou por uma reinicialização do Runtime, quando desabilitada a configuração "Iniciar Aplicação de Usuário após Reset por Cão-de-Guarda".
02	ABSENT_MODULES_HOT_SWAP_DISABLED	Aplicação em Modo Stop devido à ativação do diagnóstico Módulos Ausentes, quando configurada troca a quente desabilitada ou troca a quente desabilitada apenas para módulos declarados.
03	CFG_MISMATCH_HOT_SWAP_DISABLED	Aplicação em Modo Stop devido à ativação do diagnóstico Configuração Incompatível, quando configurada troca a quente desabilitada ou troca a quente desabilitada apenas para módulos declarados.
04	ABSENT_MODULES_HOT_SWAP_STARTUP_CONSISTENCY	Aplicação em Modo Stop devido à ativação do diagnóstico Módulos Ausentes, quando configurada troca a quente com consistência na partida ou troca a quente com consistência na partida somente para os módulos declarados.

Código	Enumerável	Descrição
05	CFG_MISMATCH_HOT_SWAP_STARTUP_CONSISTENCY	Aplicação em Modo Stop devido à ativação do diagnóstico Configuração Incompatível, quando configurada troca a quente com consistência na partida ou troca a quente com consistência na partida somente para os módulos declarados.
06	APPL_STOP_ALLOWED_TO_RUN	Aplicação em Modo Stop e todas as consistências realizadas com sucesso. Aplicação pode ser colocada no Modo Run.
07	APPL_STOP_MODULES_NOT_READY	Aplicação em Modo Stop e todas as consistências realizadas com sucesso, mas os módulos de E/S não estão aptos para a partida do sistema. Não é possível colocar a Aplicação em Modo Run.
08	APPL_STOP_MODULES_GETTING_READY_TO_RUN	Aplicação em Modo Stop e todas as consistências realizadas com sucesso. Os módulos de E/S estão sendo preparados para a partida do sistema. Não é possível colocar a Aplicação em Modo Run.
09	NORMAL_OPERATING_STATE	Aplicação em Modo Run.
10	MODULE_CONSISTENCY_OK	Uso interno.
11	APPL_STOP_DUE_TO_EXCEPTION	Aplicação em modo Stop pois uma exceção ocorreu na UCP.
12	DUPLICATED_SLOT_HOT_SWAP_DISABLED	Aplicação em Modo Stop devido à ativação do diagnóstico Slots Duplicados, quando configurada troca a quente desabilitada ou troca a quente desabilitada apenas para módulos declarados.
13	DUPLICATED_SLOT_HOT_SWAP_STARTUP_CONSISTENCY	Aplicação em Modo Stop devido à ativação do diagnóstico Slots Duplicados, quando configurada troca a quente com consistência na partida ou troca a quente com consistência na partida somente para os módulos declarados.
14	DUPLICATED_SLOT_HOT_SWAP_ENABLED	Aplicação em Modo Stop devido à ativação do diagnóstico Slots Duplicados, quando configurada troca a quente habilitada sem consistência na partida.
15	NON_DECLARED_MODULE_HOT_SWAP_STARTUP_CONSISTENCY	Aplicação em Modo Stop devido à ativação do diagnóstico Módulos Não Declarados, quando configurada troca a quente com consistência na partida.
16	NON_DECLARED_MODULE_HOT_SWAP_DISABLED	Aplicação em Modo Stop devido à ativação do diagnóstico Módulos Não Declarados, quando configurada troca a quente desabilitada.

Tabela 240: Códigos de Situações que Ocasionam Parada da Aplicação

Variável DG_HX3040.tDetailed.*	Tipo	Descrição
Application. byCPUState	ENUM(BYTE)	Informa o estado de operação da UCP: RUN (1): A aplicação está em execução (Modo Run). STOP (3): A aplicação está parada (Modo Stop).
Application. bForcedIOs	BIT	Existem um ou mais pontos de E/S forçados.

Tabela 241: Diagnósticos Detalhados Grupo Application

Variável DG_HX3040.tDetailed.*	Tipo	Descrição
TimeManager. eActiveTimeSource	ENUM(BYTE)	Indica qual é a fonte de sincronismo ativa: NONE (0): Nenhuma fonte de sincronismo está ativa. IRIGB (1): Está ativa a sincronização de tempo via IRIGB. SNTP (2): Está ativa a sincronização de tempo via SNTP. DNP (3): Está ativa a sincronização de tempo via DNP3. PTP (4): Está ativa a sincronização de tempo via PTP.
TimeManager.SNTP. bServiceEnabled	BIT	Serviço SNTP habilitado.
TimeManager.SNTP. byActiveTimeServer	ENUM(BYTE)	Indica qual servidor está ativo: NO_TIME_SERVER (0): Nenhum servidor ativo. PRIMARY_TIME_SERVER (1): Servidor primário ativo. SECONDARY_TIME_SERVER (2): Servidor secundário ativo.
TimeManager.SNTP. wPrimaryServerDownCount	WORD	Contador de vezes que o servidor primário esteve indisponível. (0 a 65535)
TimeManager.SNTP. wSecondaryServerDownCount	WORD	Contador de vezes que o servidor secundário esteve indisponível. (0 a 65535)
TimeManager.SNTP. dwRTCTimeUpdatedCount	DWORD	Contador de vezes que o RTC foi atualizado pelo serviço SNTP. (0 a 4294967295)
TimeManager.SNTP. byLastUpdateSuccessful	ENUM(BYTE)	Indica status da última atualização: NOT_UPDATED (0): Não foi atualizado. FAILED (1): Última atualização falhou. UPDATE_SUCCESSFUL (2): Última atualização teve sucesso.
TimeManager.SNTP. byLastUpdateTimeServer	ENUM(BYTE)	Indica qual servidor foi utilizado na última atualização: NO_TIME_SERVER (0): Nenhuma atualização. PRIMARY_TIME_SERVER (1): Servidor primário. SECONDARY_TIME_SERVER (2): Servidor secundário.
TimeManager.SNTP. sLastUpdateTime	EXTENDED_DATE_ AND_TIME	Data e horário da última sincronização de tempo via SNTP.
TimeManager.SNTP. sLastUpdateTime. byDayOfMonth	BYTE	
TimeManager.SNTP. sLastUpdateTime. byMonth	BYTE	
TimeManager.SNTP. sLastUpdateTime. wYear	WORD	

Variável DG_HX3040.tDetailed.*	Tipo	Descrição
TimeManager.SNTP. sLastUpdateTime. byHours	BYTE	
TimeManager.SNTP. sLastUpdateTime. byMinutes	BYTE	
TimeManager.SNTP. sLastUpdateTime. bySeconds	BYTE	
TimeManager.SNTP. sLastUpdateTime. byMilliseconds	WORD	
TimeManager.IRIGB. bServiceEnabled	BIT	Serviço IRIGB habilitado.
TimeManager.IRIGB. dwTimeUpdatedCount	DWORD	Contador de vezes que o horário foi atualizado pelo serviço IRIGB. (0 a 4294967295)
TimeManager.IRIGB. byLastUpdateSuccessful	ENUM(BYTE)	Indica qual servidor foi utilizado na última atualização: NO_TIME_SERVER (0): Nenhuma atualização. PRIMARY_TIME_SERVER (1): Servidor primário. SECONDARY_TIME_SERVER (2): Servidor secundário.
TimeManager.IRIGB. sLastUpdateTime	EXTENDED_DATE_ AND_TIME	Data e horário da última sincronização de tempo via IRIGB.
TimeManager.IRIGB. sLastUpdateTime. byDayOfMonth	BYTE	
TimeManager.IRIGB. sLastUpdateTime. byMonth	BYTE	
TimeManager.IRIGB. sLastUpdateTime. wYear	WORD	
TimeManager.IRIGB. sLastUpdateTime. byHours	BYTE	
TimeManager.IRIGB. sLastUpdateTime. byMinutes	BYTE	
TimeManager.IRIGB. sLastUpdateTime. bySeconds	BYTE	
TimeManager.IRIGB. sLastUpdateTime. byMilliseconds	WORD	

Tabela 242: Diagnósticos Detalhados Grupo TimeManager

Nota:

Fonte de sincronismo ativa: Este diagnóstico pode levar alguns segundos para ser atualizado após a ocorrência de troca da fonte de sincronismo ativa do sistema (o que também ocorre na partida do sistema).

Variável DG_HX3040.tDetailed.*	Tipo	Descrição
Events. wUsage	WORD	Amostragem do uso da fila de eventos.
Events. bEventQueueOverflow	BIT	Bit que indica que ocorreu Overflow na fila de Eventos, veja o capítulo ClearRtu-Diagnostic .
Events. bEventQueueCleared	BIT	Bit que indica que a fila de eventos foi limpa, veja o capítulo ClearEventQueue .
Events. byReserved_00	BYTE	Reservado.
Events. wReserved_01	WORD	Reservado.
Events. dwReserved_02	DWORD	Reservado.

Tabela 243: Diagnósticos Detalhados Grupo Events

Variável DG_HX3040.tDetailed.*	Tipo	Descrição
Rack. dwAbsentRacks	DWORD	Cada bit representa um número de identificação de um bastidor, se algum bit for TRUE, isso significa que o bastidor, com o referido número de identificação, está ausente.
Rack. dwDuplicatedRacks	DWORD	Cada bit representa um número de identificação de um bastidor, se algum bit for TRUE, isso significa que mais de um bastidor está configurado com o mesmo número de identificação.
Rack. dwNonDeclaredRacks	DWORD	Cada bit representa um número de identificação de um bastidor, se algum bit for TRUE, significa que há algum bastidor configurado com um número de identificação que não está declarado no projeto.

Tabela 244: Diagnósticos Detalhados Grupo Rack

7.1.5. Diagnósticos via Blocos Funcionais

Os blocos funcionais proporcionam a visualização de alguns parâmetros que não podem ser acessados de outra maneira. As três funções sobre diagnósticos avançados estão localizadas na biblioteca *LibPleStandard* e estão descritas abaixo.

7.1.5.1. GetTaskInfo

Essa função retorna informações sobre uma tarefa de uma determinada aplicação.

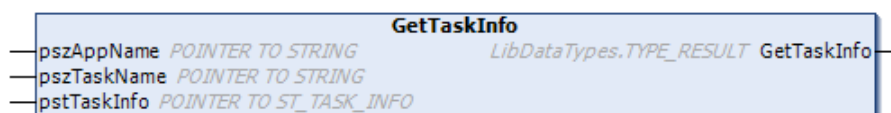


Figura 230: Função GetTaskInfo

Abaixo, são descritos os parâmetros que devem ser repassados à função para que ela retorne as informações da aplicação.

Parâmetros de entrada	Tipo	Descrição
pszAppName	POINTER TO STRING	Nome da aplicação.
pszTaskName	POINTER TO STRING	Nome da tarefa.
pstTaskInfo	POINTER TO ST_TASK_INFO	Ponteiro para receber informações da tarefa.

Tabela 245: Parâmetros de Entrada GetTaskInfo

Os dados que a função retorna, através do ponteiro informado nos parâmetros de entrada, são os descritos na tabela abaixo.

Parâmetros de saída	Tipo	Descrição
dwCurScanTime	DWORD	Tempo de ciclo (execução) da tarefa com 1 μ s de resolução.
dwMinScanTime	DWORD	Tempo mínimo de ciclo da tarefa com 1 μ s de resolução.
dwMaxScanTime	DWORD	Tempo máximo de ciclo da tarefa com 1 μ s de resolução.
dwAvgScanTime	DWORD	Tempo médio de ciclo da tarefa com 1 μ s de resolução.
dwLimitMaxScan	DWORD	Tempo máximo de ciclo da tarefa antes de ocorrer o cão-de-guarda.
dwIECCycleCount	DWORD	Contador de ciclos IEC decorridos.

Tabela 246: Parâmetros de Saída GetTaskInfo

Possíveis TYPE_RESULT:

- OK_SUCCESS: Execução com sucesso.
- ERROR_FAILED: A tarefa desejada não existe.

Exemplo de utilização em Linguagem ST:

```

PROGRAM UserPrg
VAR
  sAppName : STRING;
  pszAppName : POINTER TO STRING;
  sTaskName : STRING;
  pszTaskName : 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.
pszAppName := ADR(sAppName); //Ponteiro com o nome da aplicação.

```

```

sTaskName := 'MainTask'; //Variável recebe o nome da tarefa.
psTaskName := ADR(sTaskName); //Ponteiro com o nome da tarefa.
pstTaskInfo := ADR(TaskInfo); //Ponteiro que irá receber as informações da
tarefa.
//FUNÇÃO:
//Chamada da função.
Info := GetTaskInfo (psAppName, psTaskName, pstTaskInfo);
//Variável 'Info' recebe possíveis erros da função.

```

7.1.6. 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, consultar a seção [One Touch Diag](#).

Na figura abaixo, é possível visualizar todos os caracteres disponíveis no visor gráfico da UCP Hadron Xtorm e, a seguir, os seus respectivos significados.

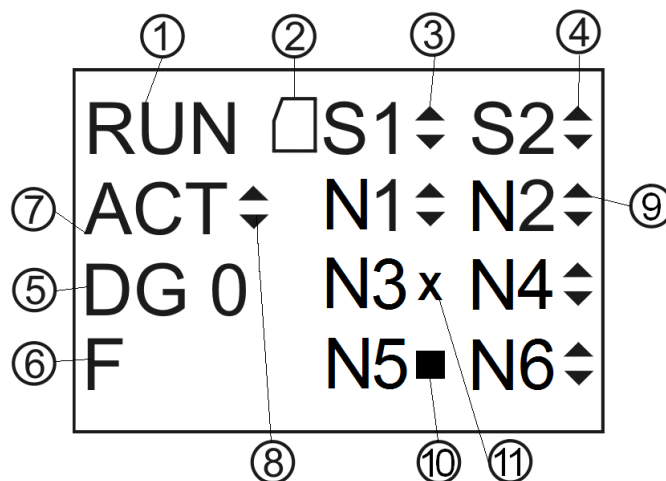


Figura 231: Tela de Status da UCP HX3040

Legenda:

- 1. Indicação do estado de operação da UCP. Caso a aplicação da UCP esteja em execução, o estado será Run. Caso a aplicação da UCP esteja parada, o estado será STOP e quando estiver parado em marcas de depuração da aplicação o estado será BRKP. Para maiores detalhes, consultar a seção [Estados de Operação da UCP](#).
- 2. Indicação da presença do Cartão de Memória. Para maiores detalhes sobre a instalação do cartão, consultar a seção [Instalação do Cartão de Memória](#).
- 3. Indicação de Tráfego na COM 1. A seta para cima (▲) indica transmissão de dados e a seta para baixo (▼) indica recepção de dados. Para maiores informações sobre a interface COM 1, consultar seção [Interfaces Seriais](#).
- 4. Indicação de Tráfego na COM 2. A seta para cima (▲) indica transmissão de dados e a seta para baixo (▼) indica recepção de dados. Para maiores informações sobre a interface COM 2, consultar seção [Interfaces Seriais](#).
- 5. Indicação da quantidade de diagnósticos ativos na UCP. Caso o número mostrado seja diferente de 0 (zero), existem diagnósticos ativos na UCP. Para maiores detalhes sobre a visualização dos mesmos no visor gráfico da UCP, através da tecla de diagnósticos, consultar seção [One Touch Diag](#).
- 6. Indicação de variáveis forçadas na UCP. Caso o caractere F esteja exibido no visor gráfico, alguma variável está sendo forçada pelo usuário, seja ela a variável simbólica ou variável de representação direta mapeada em uma variável simbólica. Para maiores detalhes sobre forçamento de variáveis, consultar seção [Modo Run](#).

- 7. 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).
- 8. Indicação de que está sendo executada a sincronização de projeto. A seta para cima (▲) indica transmissão de dados do projeto e a seta para baixo (▼) indica recepção de dados de um projeto. Para maiores informações sobre a sincronização de projetos, consultar seção [Sincronização de Projetos](#).
- 9. 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 a seção [Configuração das Interfaces Ethernet](#).
- 10. 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 a seção [Configuração das Interfaces Ethernet](#).
- 11. 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 a seção [Configuração das Interfaces Ethernet](#).

Além dos caracteres descritos acima, as UCPs Nexto poderão apresentar algumas mensagens no visor gráfico, correspondentes a algum processo que está sendo executado no momento.

A tabela abaixo mostra as mensagens e suas respectivas descrições:



Mensagem	Descrição
FORMATANDO...	Indica que a UCP está formatando o cartão de memória.
ERRO NA FORMATACAO	Indica que ocorreu um erro durante a formatação do cartão de memória pela UCP.
FORMATO INCORR.	Indica que o formato do cartão de memória está incorreto.
SENHA INCORRETA	Indica que a senha digitada não confere com a senha configurada.
TRANSFERINDO...	Indica que o projeto está sendo transferido.
ERRO NA TRANSFER.	Indica que ocorreu um erro na transferência do projeto, ocasionado por algum problema no cartão de memória ou a remoção do mesmo durante a transferência.
TRANSFER. COMPLETA	Indica que a transferência foi completada com êxito.
TIMEOUT TRANSFER.	Indica que ocorreu um time-out (expirou tempo de comunicação) durante a transferência do projeto.
MODELO UCP INVALIDO	Indica que o modelo da UCP é diferente do configurado no projeto que está no cartão de memória.
VERSAO UCP INVALIDA	Indica que a versão da UCP é diferente do configurado no projeto que está no cartão de memória.
APLICACAO CORROMPIDA	Indica que a aplicação que está no cartão de memória está corrompida.
APLICACAO INEXISTENTE	Indica que não existe aplicação dentro do cartão de memória para ser transferida à UCP.
CRC INEXISTENTE	Indica que o CRC da aplicação não existe.
MCF INEXISTENTE	Indica que não existe o arquivo MCF dentro do cartão de memória.
SEM TAG	Não existe Tag configurada para a UCP no MasterTool IEC XE.
SEM DESC	Não existe Descrição configurada para a UCP no MasterTool IEC XE.
ERRO MSG.	Indica que há erro(s) na(s) mensagem(s) de diagnóstico(s) do(s) módulo(s) requisitado(s).
SEM ASSINATURA	Indica que o produto apresentou um problema inesperado. Entre em contato com o setor de Suporte da Altus.
ERRO NA APL. REINICIANDO	Indica que ocorreu um erro na aplicação e o Runtime está reiniciando a aplicação.

Mensagem	Descrição
APL. NAO CARREGADA	Indica que o Runtime não irá carregar a aplicação.
CARREGANDO APL.	Indica que o Runtime irá carregar a aplicação.
POSICAO INCORRETA	Indica que a UCP está em uma posição incorreta no bastidor.
ERRO FATAL	Indica que há problemas graves na inicialização da UCP como por exemplo, as partições da UCP não foram devidamente montadas. Entre em contato com o setor de suporte da Altus.
HW-SW INCOMPATIVEL	Indica que o software e hardware da UCP não são compatíveis pois o produto apresentou algum problema inesperado. Entre em contato com o setor de suporte da Altus.
ATUALIZANDO FIRMWARE	Indica que o firmware está sendo atualizado na UCP.
RECEBENDO FIRMWARE	Indica que o arquivo de atualização está sendo transferido para a UCP.
ATUALIZADO	Exibe a versão de firmware atualizada na UCP.
ERRO NA ATUALIZACAO	Indica que houve algum erro durante a atualização de firmware da UCP, ocasionado por falha na comunicação ou problemas de configuração.
REINICIANDO SISTEMA...	Indica que a UCP está sendo reiniciada para que as atualizações tenham efeito.

Tabela 247: Outras Mensagens do Visor Gráfico

7.2. 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 na 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*.

ATENÇÃO

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

Nota:

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.

7.3. Diagnósticos dos Módulos de E/S

As funções de acesso do 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:

1. Visor gráfico da UCP
2. Visor do módulo a ser acessado
3. Botão de diagnóstico do módulo a ser acessado

7.3.1. 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 abaixo.

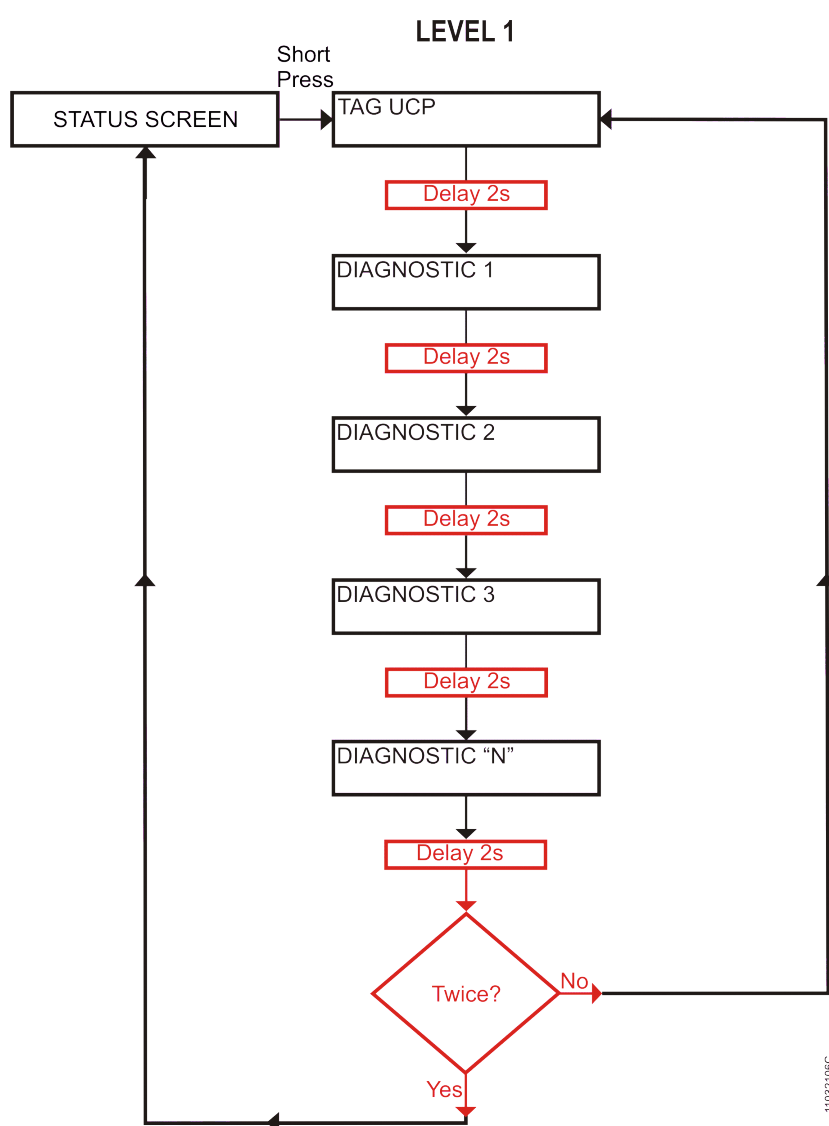


Figura 232: Visualização dos Diagnósticos da UCP

Conforme mostrado na figura acima, 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 sair 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, indicados na figura abaixo, do visor do módulo estiverem piscando. A quantidade de segmentos depende da quantidade de pontos de cada um dos módulos de E/S.

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

Figura 233: Visor do Módulo

7.3.2. 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 do mesmo da mesma forma como indicada no fluxograma da figura 232.

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.

7.3.3. 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.

ATENÇÃO

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 caractér ponto “.”.

A utilização de qualquer outro caractér diferente dos acima citados não é recomendada.

ATENÇÃO

Quando utilizar a função *ETD – Electronic Tag on Display* 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.

ATENÇÃO

Quando utilizar a função *ETD – Electronic Tag on Display* 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.

ATENÇÃO

Quando utilizar a função *ETD – Electronic Tag on Display* 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.

7.3.4. Pressionamento curto e pressionamento longo

A tabela abaixo indica os tempos.

Tipo de Pressionamento	Tempo Mínimo	Tempo Máximo	Condição para Indicação
Sem pressionamento	-	59,99 ms	-
Pressionamento curto	60 ms	0,99 s	Pressionamento e liberação do botão dentro do período definido
Pressionamento longo	1 s	20 s	Pressionamento e liberação do botão dentro do período definido
Botão preso	20,01 s	(∞)	Pressionamento do botão por mais de 20 segundos

Tabela 248: Tempos de Pressionamento Botão de Diagnóstico

7.4. Não Carregar a Aplicação na Inicialização

Caso seja necessário, o usuário pode optar por não carregar uma aplicação já existente na UCP durante sua inicialização. Para isto, basta energizar a UCP com 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.

7.5. 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).

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.

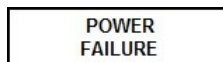


Figura 234: Mensagem de Falha na Alimentação

7.6. Problemas mais Comuns

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

- A temperatura ambiente está dentro da faixa suportada pelos equipamentos?
- A fonte de alimentação do bastidor está sendo alimentada com a tensão correta?
- A fonte de alimentação é o módulo mais a 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 a esquerda no bastidor será a fonte, a sua direita é seguido 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.

7.7. Solução de Problemas

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

Sintoma	Possível Causa	Solução
Não liga	Falta de alimentação ou alimentado incorretamente.	Verificar se a UCP está conectada corretamente no bastidor. Desenergizar e retirar todos os módulos do barramento, menos a fonte de alimentação e a UCP. Energizar o barramento e verificar o funcionamento da fonte de alimentação, tanto a externa como a fonte conectada ao barramento. Verificar se a tensão de alimentação chega ao borne da fonte de alimentação Hadron Xtorm, e também se a polarização está correta.
Não comunica	Mau 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.

Tabela 249: Solução de Problemas

7.8. Manutenção Preventiva

- Deve-se verificar, a cada ano, se os cabos de interligação estão com as conexões firmes, sem depósitos de poeira, principalmente os dispositivos de proteção.
- Em ambientes sujeitos a contaminação excessiva, deve-se limpar periodicamente o equipamento, retirando resíduos, poeira, etc.
- Os diodos TVS utilizados para a proteção contra transientes causados por descargas atmosféricas devem ser inspecionados periodicamente, pois podem estar danificados ou destruídos caso a energia absorvida esteja acima do limite. Em muitos casos, a falha pode não ser evidente ou facilmente visível. Em aplicações críticas, é recomendável a substituição periódica dos diodos TVS, mesmo os que não apresentarem sinais visíveis de falha.
- Aperto e limpeza do barramento a cada 6 meses.

8. Projeto do Painel Elétrico

8.1. Projeto Mecânico

8.1.1. Dimensões

A dimensão de cada módulo da Série Hadron Xtorm pode ser encontrado no documento de Características Técnicas de cada módulo, cujos documentos estão listados na tabela 1.

As dimensões dos módulos principais são mostradas a seguir, em mm.

8.1.1.1. Bastidores

8.1.1.1.1. Bastidor de 9 Posições

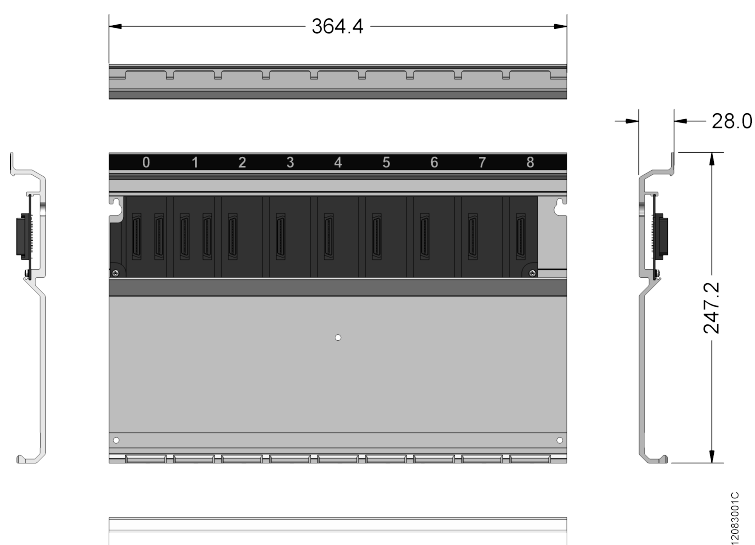


Figura 235: Bastidor de 9 posições

8.1.1.1.2. Bastidor de 18 Posições

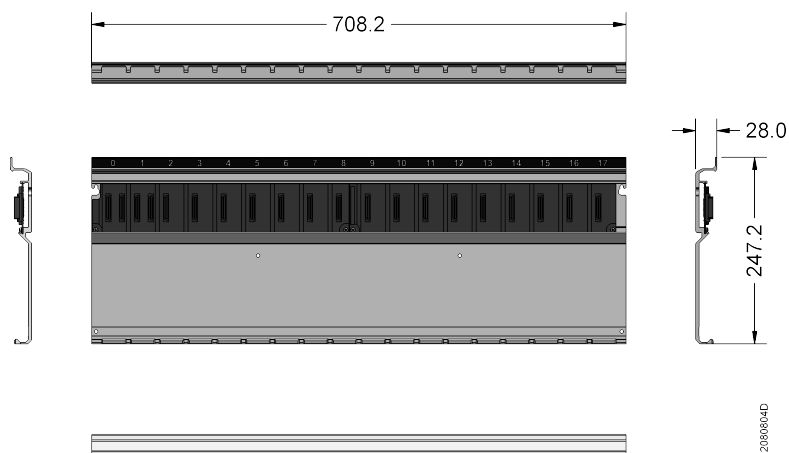


Figura 236: Bastidor de 18 posições

8.1.1.2. Módulos

Este tamanho de módulo é usado por todos os demais módulos da Série Hadron Xtorm. A figura ilustra a UCP HX3040.

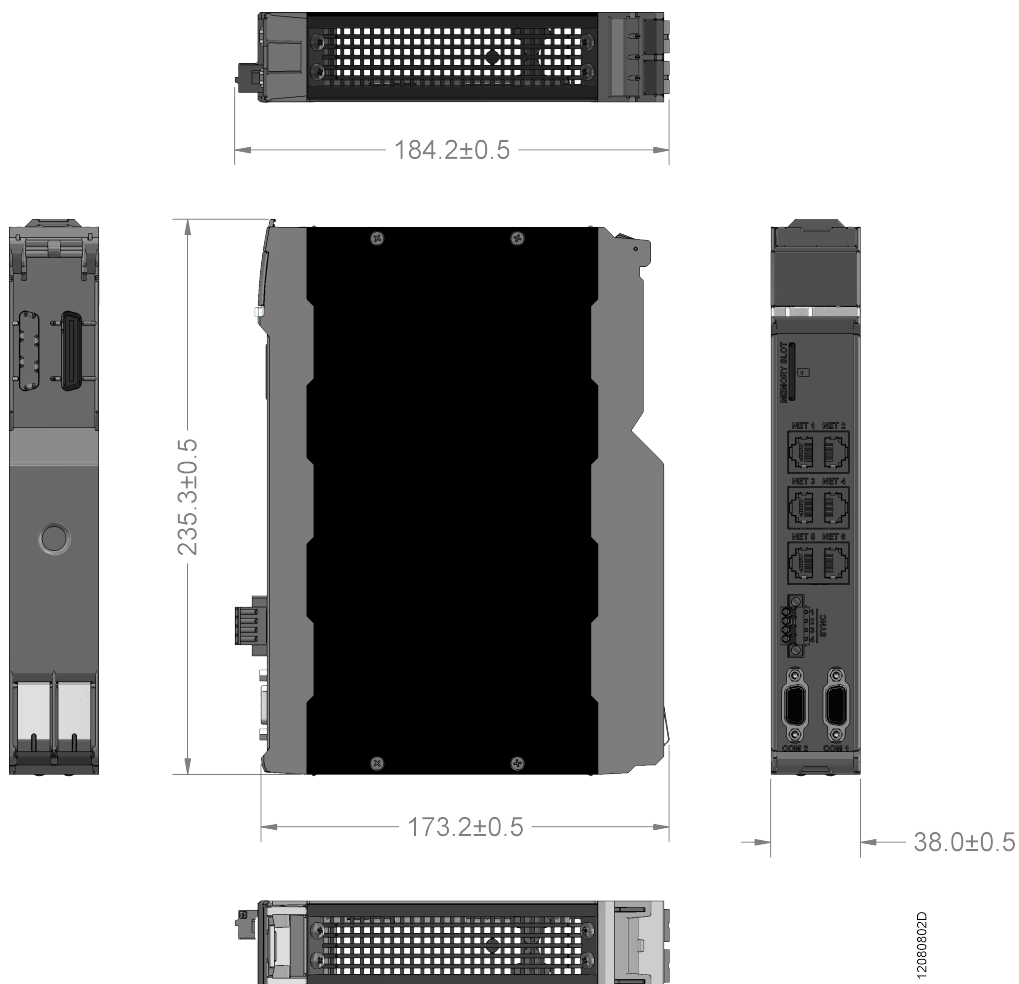


Figura 237: Módulo

8.1.2. Profundidade do Módulo Montado no Bastidor

A profundidade do conjunto módulo da Série Hadron Xtorm e bastidor da Série Hadron Xtorm pode ser encontrada ao somar 28 mm à profundidade do módulo. No exemplo da figura abaixo foi utilizado um módulo padrão com profundidade de 184,2 mm. Ao considerar o bastidor, a profundidade do conjunto passa para 212,2 mm podendo variar 0,5mm.

Dimensão em mm.

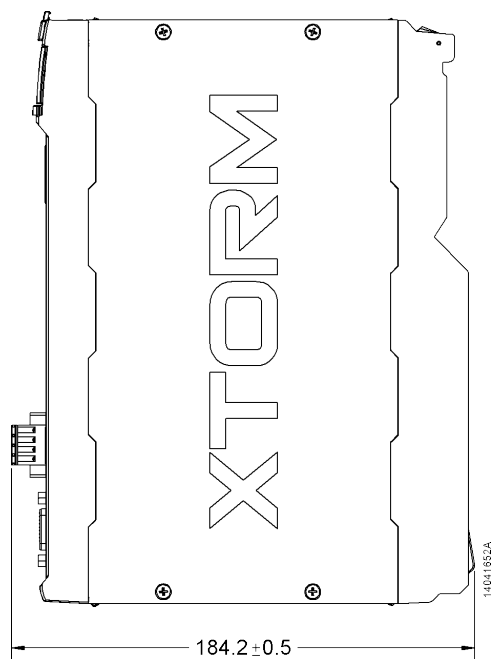


Figura 238: Profundidade do Módulo Montado no Bastidor

8.1.3. Espaçamento entre módulos e outros equipamentos do painel

A UTR necessita de espaços livres em torno do mesmo. Isto é necessário para permitir o correto manuseio do equipamento. Além disto, tal espaçamento deve ser respeitado de forma a permitir a passagem de ar através da UCP, na forma de convecção, responsável por manter a temperatura do equipamento.

A figura abaixo e a tabela a seguir indicam o espaçamento necessário para os módulos da Série Hadron Xtorm.

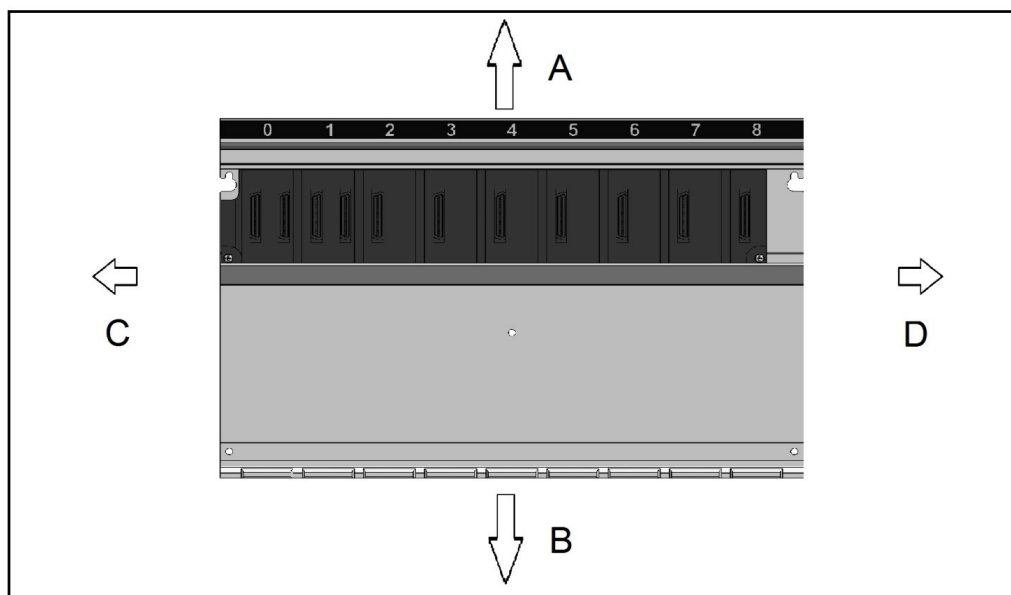


Figura 239: Espaço Livre em Torno da UTR

Dimensão A	Dimensão B	Dimensão C	Dimensão D
10 cm	10 cm	4 cm	4 cm

Tabela 250: Dimensões do Espaço Livre em Torno da UTR

8.1.4. Dimensionamento da Calha

Para o dimensionamento da calha, além da área ocupada pelos fios, observar o aquecimento interno na calha, provocado pelo calor dissipado pelos fios, que pode levar a uma redução na área de ocupação da calha.

Utilizar a seguinte regra: área da calha \geq somatório da área dos fios / 0,4

Área dos fios = $(3,14 * \text{raio}^2)$

Considera-se como área dos fios a área total, incluindo a isolamento.

8.1.5. Montagem Horizontal/ Vertical

A Série Hadron Xtorm permite a utilização da UTR na orientação horizontal. Não é permitida a montagem vertical do bastidor.

8.2. Projeto Térmico

Os equipamentos Altus são projetados para trabalhar a uma temperatura ambiente de até 60°C (exceto quando especificado). Portanto, esta deve ser a temperatura interna máxima do armário. Os seguintes cuidados devem ser observados no projeto do painel:

- Dimensionar armários com volume interno suficiente para uma boa circulação de ar.
- Prever ventilação forçada ou trocadores de ar com o meio externo, caso necessário, para que não haja elevação da temperatura além do limite. Em casos críticos, recomenda-se o uso de equipamentos de refrigeração, para manter o equipamento operando dentro dos níveis de temperatura de operação.
- Distribuir de forma homogênea fontes de calor dentro do armário.
- Considerar a dissipação nos cabos que conduzem correntes mais elevadas para evitar superaquecimento interno às calhas.

ATENÇÃO

Para obter a máxima dissipação de cada módulo da Série Hadron Xtorm, consulte o documento Características Técnicas do módulo em questão.

A seguir, é explicado um método para calcular a temperatura interna do painel em função da sua dissipação e potência.

8.2.1. Dissipação de calor em um painel elétrico

Cada painel elétrico dissipa, através de sua superfície, uma quantidade definida de calor para uma dada diferença de temperatura interna e externa. Para o cálculo da dissipação de calor em situações em que a diferença de temperatura entre o interior e o exterior do painel chegue até 50 °C, as seguintes grandezas devem ser consideradas:

- Superfície de dissipação efetiva do painel; calculada segundo a norma DIN-VED 0660 capítulo 500, conforme indicado pelo tipo de instalação.
- A constante de dissipação para a chapa de aço pintada em $W/m^2 \text{ } ^\circ C$.
- As condições de ventilação do painel (local de instalação).
- Grau de ocupação interna do painel (impedância à circulação do ar no interior)

Dos valores citados anteriormente, apenas o valor da superfície do painel pode ser calculada exatamente.

Cálculo da superfície efetiva de dissipação A (m^2) de um painel:

O cálculo da superfície "A" é feito conforme indicado pela norma DIN-VDE, segundo o tipo de instalação do painel:

Tipo de instalação conforme a norma DIN-VDE 0660/500	Fórmula para o cálculo de A (m ²)
Painel livre de todos os lados	$A = 1,8 * H * (L + P) + 1,4 * L * P$
Painel com a superfície traseira obstruída	$A = 1,4 * L * (H + P) + 1,8 * P * H$
Painel com uma superfície lateral obstruída	$A = 1,4 * L * (H + L) + 1,8 * L * H$
Painel com uma lateral e a superfície traseira obstruídas	$A = 1,4 * H * (L + P) + 1,4 * L * P$
Painel com as duas laterais obstruídas	$A = 1,8 * L * H + 1,4 * L * P + P * H$
Painel com as duas laterais e a superfície traseira obstruídas	$A = 1,4 * L * (H + P) + P * H$
Painel com as duas laterais e a superfície traseira e superior obstruídas	$A = 1,4 * L * H + 0,7 * L * P + P * H$

Tabela 251: Cálculo da Superfície Efetiva de Dissipação

L = Largura (m), H = Altura (m), P = Profundidade (m)

Na aplicação em painéis construídos com chapa de aço pintada, para o ar parado ao seu redor, a constante de dissipação de calor pode ser considerada 5,5 W/m² °C.

A potência dissipada por um painel pode então ser calculada por meio da equação $Q_s = k * A * (\text{temperatura interna} - \text{temperatura externa})$, ou obtida a partir da figura abaixo.

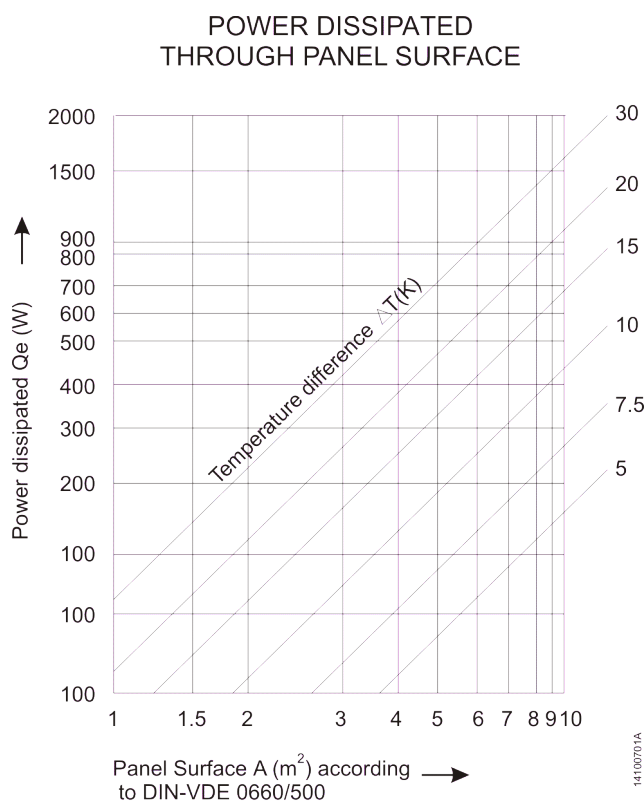


Figura 240: Potência Dissipada x Superfície x Dif. de Temperatura

Este valor poderá, no entanto, ser triplicado se for provocada circulação de ar no exterior do painel.

A circulação de ar em um painel é obstruída pela instalação dos equipamentos em seu interior, conduzindo à formação de focos de aquecimento localizados. Nesta situação, pode-se obter auxílio para a circulação desejada do ar por meio da instalação de ventiladores internos ao painel, aumentando o fluxo do ar em seu interior.

A circulação forçada através de ventiladores no interior do painel traz também uma melhora na convecção própria e uma tendência a se igualarem as temperaturas ao longo do painel. Sem a circulação forçada do ar tem-se um foco de calor no alto do painel, em virtude da convecção.

Exemplos:

Para um painel livre de todos os lados, com área efetiva de 3,96 m², potência instalada de 350 W e temperatura ambiente externa de 30 °C, calcule a temperatura média interna.

$$Q_s = k * A * (T_i - T_e)$$

$$350 = 5,5 * 3,96 * (T_i - 30)$$

$$T_i = 46 \text{ °C}$$

Para o mesmo painel, calcule a temperatura interna para uma potência instalada de 1000 W.

$$Q_s = k * A * (T_i - T_e)$$

$$1000 = 5,5 * 3,96 * (T_i - 30)$$

$$T_i = 76 \text{ °C}$$

Neste caso, a temperatura excedeu o limite de operação dos equipamentos (60 °C), e deve ser providenciada uma outra maneira para retirada do calor excedente. O limite da potência instalada para a temperatura interna de 60 graus é:

$$Q_s = k * A * (T_i - T_e)$$

$$Q_s = 5,5 * 3,96 * (60 - 30)$$

$Q_s = 653 \text{ W}$, sendo o limite 653 W, os 347 W restantes (1000 W – 653 W) devem ser retirados, por exemplo, através de um equipamento de ar-condicionado.

ATENÇÃO

Nos cálculos anteriores, observar que a temperatura interna, é sempre uma temperatura média, e que caso não haja circulação forçada de ar no interior do painel, a temperatura no topo do painel será maior que na base, e poderão existir focos quentes localizados. A devida margem de segurança deve ser dada em cada caso.

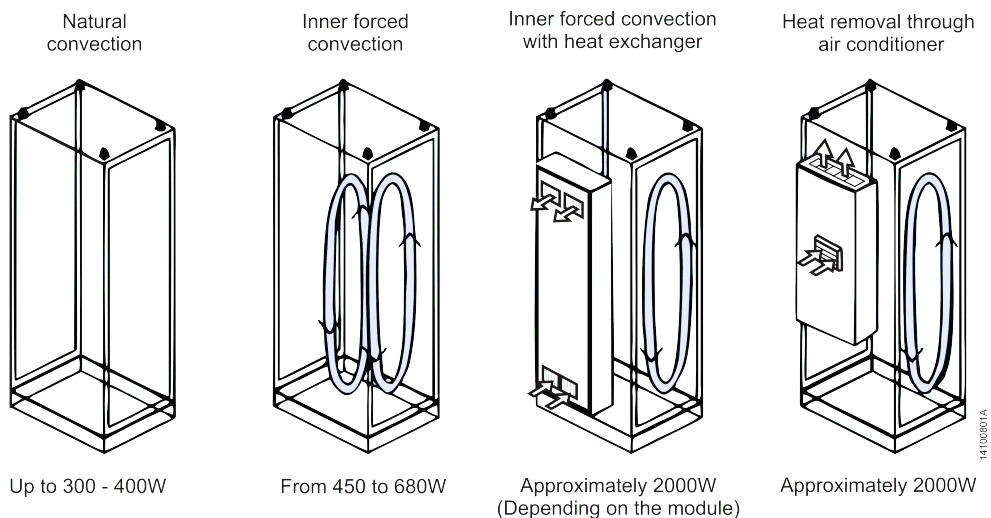


Figura 241: Exemplo de Movimentação do Calor - Instalação Fechada

Uma dissipação bem maior de calor, comparando-se com a obtida anteriormente, pode ser alcançada se for permitida a troca de ar com o exterior. A ventilação é normalmente realizada introduzindo-se venezianas de ventilação nas laterais, na porta ou na tampa traseira. Isto irá evidentemente reduzir o grau de proteção (IP) do painel.

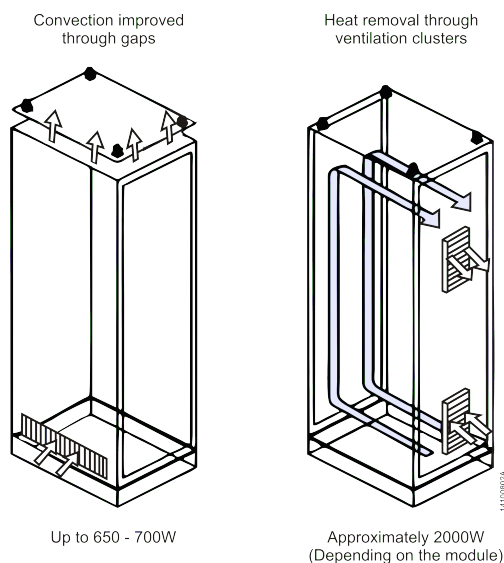


Figura 242: Exemplo de Movimentação do Calor - Instalação Aberta

8.3. Projeto Elétrico

8.3.1. Informações Gerais

As Unidades Terminais Remotas são fabricadas atendendo normas mundiais, que estabelecem os níveis aceitáveis de condições ambientais e de ruído normalmente encontrados em aplicações de controle de sistemas de geração, transmissão e distribuição de energia elétrica, como usinas hidrelétricas (UHEs), e subestações de energia.

É fundamental também que a instalação destes produtos siga regras de projeto convenientes, estabelecidas por normas de instalação. Problemas causados por interferências eletromagnéticas (EMI), tais como falhas de comunicação, falhas de execução de programa, ruído em variáveis analógicas, e até mesmo perda de programa, podem ser causadas por um projeto elétrico ou instalação deficientes.

O projeto elétrico das UTR's Altus deve respeitar a norma IEEE 518/1977, "Guide for Installation of Electrical Equipment to Minimize Electrical Noise Inputs to Controllers External Sources". A seguir, os pontos mais importantes são abordados.

8.3.2. Alimentação do Armário

A alimentação do sistema de controle deve possuir chave geral. Recomenda-se o uso de bornes para alimentação geral do painel de montagem com fusíveis integrados, bem como a previsão de uma tomada fornecendo 127 ou 220 Vac, para uso do terminal de programação. É importante que esta tomada possua pino de aterramento, pois o terminal de programação deverá, obrigatoriamente, possuir conexão com o terra do sistema. Todas as tomadas do armário devem possuir indicação clara de suas tensões.

8.3.3. Distribuição dos Cabos no Armário

A correta distribuição dos cabos no armário e o correto aterramento das partes garantem a compatibilidade eletromagnética (EMC) da instalação.

É importante que as alimentações do painel elétrico sejam corretamente distribuídas, através de barras de distribuição ou bornes de ligação.

A partir destes pontos de distribuição geral, leva-se um cabo próprio a cada ponto específico a ser alimentado. Deve-se evitar ramificações locais nas alimentações dos módulos, diminuindo-se assim os percursos dos cabos conduzindo alta corrente.

Para o melhor desempenho do equipamento, é necessário separar os circuitos quanto ao seu tipo, para reduzir interferências eletromagnéticas, como segue:

1. Circuitos de alimentação AC e acionamentos de cargas AC e DC
2. Circuitos de entrada e saídas digitais de baixa corrente (menor ou igual a 1A)
3. Circuitos analógicos e de comunicação

Estes circuitos devem ser distribuídos preferencialmente em calhas separadas ou evitando-se que se disponham paralelamente uns aos outros. A distância mínima de 150 mm é recomendada entre todos os sinais de E/S e alimentações maiores que 500 V.

8.3.4. Iluminação do Armário

É fundamental que se coloque iluminação interna no armário, acionada por interruptor, para facilitar a sua operação.

Recomenda-se que a iluminação seja com lâmpadas incandescentes, pois lâmpadas fluorescentes podem gerar interferências indesejáveis. Se estas forem utilizadas, as seguintes precauções devem ser tomadas de modo a reduzir a interferência:

1. Colocar tela metálica aterrada entre a lâmpada e o armário, para reduzir a emissão de ruídos;
2. Colocar blindagem nos cabos de alimentação da lâmpada;
3. Proteger o interruptor em caixa metálica e colocar filtro na rede de alimentação preferencialmente junto à lâmpada.

8.3.5. Aterramento

É necessária uma borneira geral ou uma barra de terra no armário, onde serão realizados todos os aterramentos de fontes e módulos. Esta barra deve estar ligada a um terra com baixa resistência.

8.3.6. Interferência Eletromagnética

A interferência eletromagnética (EMI) é responsável pela grande maioria dos problemas encontrados em equipamentos instalados.

Pode-se reduzir significativamente estes problemas se as seguintes precauções forem tomadas na fase de projeto do painel:

1. Distribuir e arranjar os cabos nas calhas, evitando misturar cabos de alimentação com cabos de sinais;
2. Partes metálicas inativas devem ser aterradas no armário;
3. Caso existam elementos que causem emissão de ruídos, recomenda-se a utilização de blindagens;
4. Filtrar a entrada de alimentação do painel.

8.3.7. Blindagem

Fortes fontes geradoras de interferência eletromagnética (transformadores, motores, cabos com alta corrente ou tensão) situadas dentro do armário, devem ser cobertas por chapas metálicas aterradas, quando situados a menos de 50 cm das partes eletrônicas da UTR. Cabos que ultrapassam as partes blindadas devem ser blindados ou filtrados.

Os cabos blindados dentro do armário devem ser aterrados conforme as instruções de cada equipamento.

8.3.8. Supressores de Ruído

É extremamente importante a conexão de supressores de ruído de porte adequado diretamente em todas as cargas indutivas (relés, contactoras, solenóides, etc.) acionadas ou não pela UTR. O acionamento de cargas indutivas gera fortes ruídos elétricos que podem ultrapassar os limites estabelecidos pelas normas. Os ruídos, se não atenuados em sua origem, podem atingir a UTR, afetando seu funcionamento.

Os circuitos de proteção devem ser montados próximos da carga, como regra, não devem estar afastados mais que 0,5 metros. No caso de cargas resistivas (lâmpadas incandescentes, LEDs de sinalização, resistores de aquecimento, etc.), não é necessário o uso destes dispositivos.

8.3.8.1. Circuito RC

O circuito de proteção RC (Resistor em série com um Capacitor) pode ser montado em paralelo com o contato ou em paralelo com a carga. A montagem em paralelo com os contatos é recomendada para cargas alimentadas em tensão contínua. A montagem em paralelo com a carga é recomendada para cargas alimentadas com tensões contínuas ou alternadas. Os circuitos RC são mais eficazes quando utilizados em tensões acima de 100 V.

Para selecionar os valores de R e C, recomenda-se que o resistor tenha de 0,5 a 1 Ohm para cada 1 V de tensão, e o capacitor tenha 0,5 a 1 μF para cada 1 A de corrente. Por exemplo, em uma carga de 220 V/1 A pode-se utilizar um resistor de 220 Ohms e um capacitor de 1 μF (o modelo do capacitor deve estar adequado ao tipo e valor da tensão da carga).

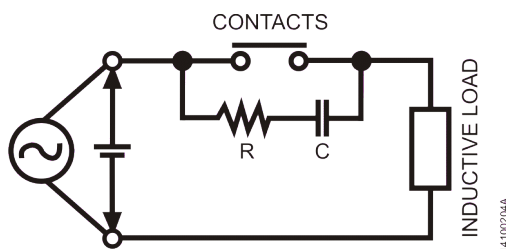


Figura 243: CKT RC Paralelo com os Contatos

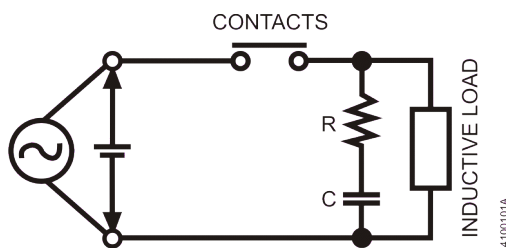


Figura 244: CKT RC Paralelo com a Carga

ATENÇÃO

Para tensões contínuas de 110/125 Vdc, deve ser usado o circuito de proteção RC em paralelo com a carga, sob pena de desgaste acentuado dos contatos. Sugere-se os seguintes valores de R e C: 33 Ohms e 470 μ F, respectivamente.

8.3.8.2. Circuito com Diodo

Esta é a forma mais eficiente para limitar a tensão do circuito indutivo no momento do desarme. Porém, pode trazer problemas pois aumenta o tempo de desarme caso a carga seja, por exemplo, uma contactora ou solenóide.

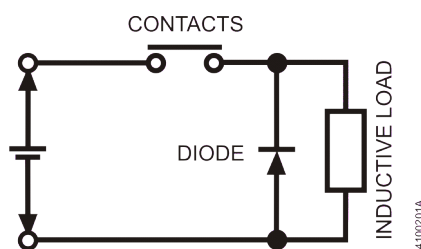


Figura 245: CKT com Diodo

O circuito pode ser utilizado somente para tensões contínuas, sua tensão reversa recomendada deve ser 10 vezes maior que a da fonte e a corrente no mínimo igual a da carga.

8.3.8.3. Circuito com Diodo e Zener

O circuito com diodo e zener é adequado quando o tempo de desarme do circuito com diodo é excessivo. Assim como o circuito de proteção com diodo, ele só deve ser utilizado em tensões contínuas. A tensão do zener deve ser um pouco superior da tensão de pico da carga.

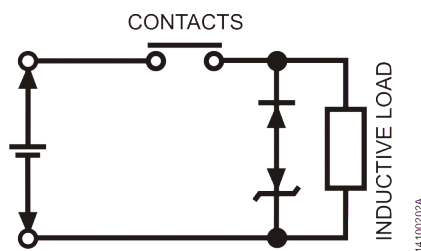


Figura 246: CKT Diodo Zener

8.3.8.4. Circuito com Varistor

O circuito com varistor limita a tensão do circuito indutivo de forma semelhante a um diodo zener. Sua tensão de condução é em geral maior que um diodo zener e é bidirecional, possibilitando seu uso em circuitos DC ou AC, onde é mais utilizado. Deve ser selecionado conforme tensão máxima da fonte, energia armazenada na carga e vida útil desejada.

ATENÇÃO

É imprescindível a leitura completa das características técnicas dos produtos utilizados antes da instalação ou utilização dos mesmos. Para a escolha do tipo de supressor a ser utilizado, deve-se verificar o tipo de carga (DC ou AC) e os níveis de tensão suportado pelo módulo escolhido para o projeto.

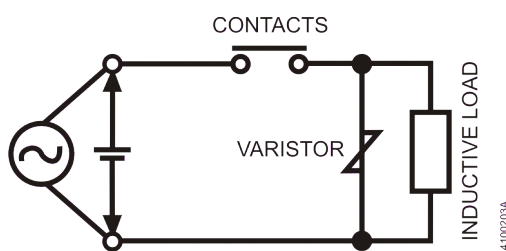


Figura 247: CKT Varistor

8.3.8.5. Circuito com Capacitor

O circuito com capacitor é altamente eficaz para suprimir arcos gerados durante a abertura dos contatos, mas pode causar desgaste do contato em função da corrente para carga e descarga do capacitor. Para a seleção do capacitor usar a mesma regra do circuito RC.

ATENÇÃO

Este circuito é desaconselhado na maioria das aplicações. Somente deve ser escolhido quando os circuitos anteriores se mostrarem inadequados.

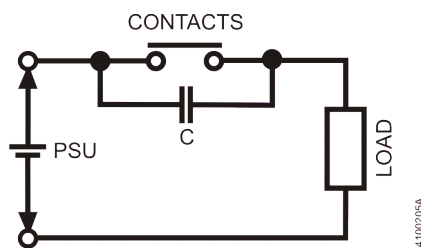


Figura 248: CKT com Capacitor em Paralelo com Contatos

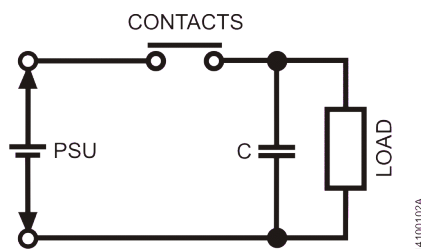


Figura 249: CKT com Capacitor em Paralelo com a Carga

8.3.9. Distribuição das Alimentações fora do Armário

Em aplicações onde o armário está distante da máquina ou do sistema a ser controlado, embora esteja no mesmo prédio, recomendam-se os seguintes procedimentos:

1. A condução dos cabos do armário à máquina deve ser feita em condutores metálicos.
2. O aterramento destes dutos deve ser feito a cada 20 metros.
3. Separar os cabos em dois grupos para distribuição nos dutos:
 - Cabos de sinais digitais até 60 V, cabos blindados conduzindo sinais analógicos e cabos blindados com alimentações até 230 V;
 - Cabos com tensão superior a 230 V.

8.3.10. Proteção contra Raios

Em aplicações externas, ou seja, em que os cabos ou linhas de comunicação do UCP com os sinais de campo saiam para fora da instalação ou percorram caminhos a céu aberto, deve-se considerar os possíveis danos causados por raios.

Recomenda-se o uso de varistores ou arrestores (com gases inertes) nestes cabos, para proteção do sistema contra sobre tensões decorrentes da queda de raios nestas linhas. Algumas blindagens também são necessárias, conforme mostra a figura abaixo.

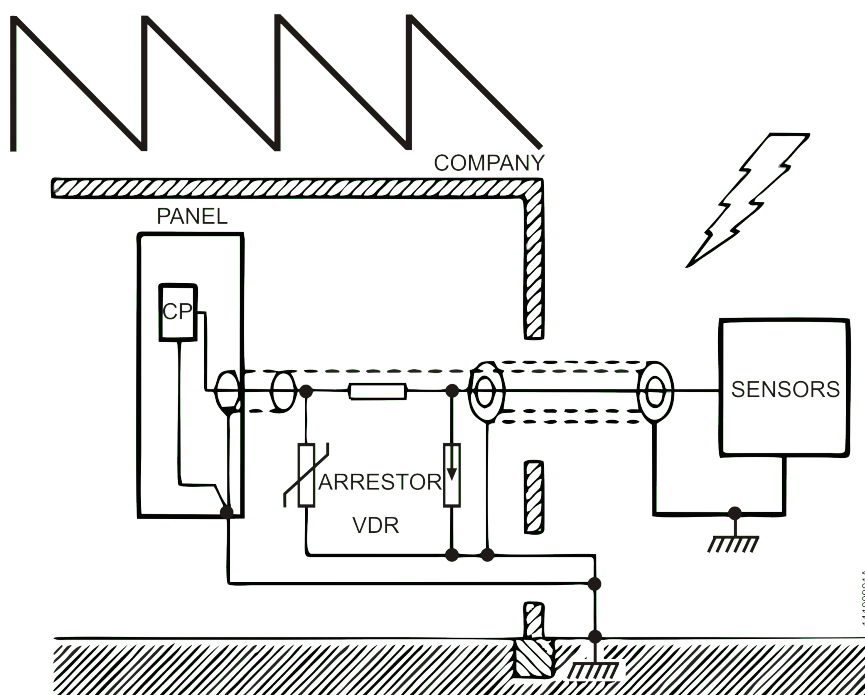


Figura 250: Proteção Contra Raios

É recomendável que se instalem estes dispositivos de proteção junto à entrada da indústria ou mesmo junto ao armário.

8. PROJETO DO PAINEL ELÉTRICO

A figura acima mostra a forma correta de instalação de proteção contra raios para um sistema genérico. Cada sistema possui detalhes próprios de instalação, portanto recomenda-se que se estudem cada caso individualmente para definição da melhor forma de proteção.

Em casos considerados críticos, consulte diretamente o serviço de suporte da Altus.