

Driver IEC870-5-101

Nome do Arquivo	IEC870-101.dll
Fabricante	International Electrotechnical Commission
Equipamentos	PLCs e Estações Remotas compatíveis com a norma IEC 870-5-101
Protocolo	IEC 870-5-101 Mestre Não Balanceado, IEC 870-5-101 Escravo Não Balanceado, IEC 870-5-101 Mestre Balanceado e IEC 870-5-101 Escravo Balanceado
Versão	4.0.4
Última Atualização	10/08/2026
Plataforma	Win32
Dependências	Nenhuma
Leitura com Superblocos	Não
Nível	31201

Introdução

O Driver IEC870-5-101 implementa o **Mestre** e o **Escravo** do protocolo IEC 870-5-101 **Balanceado** e **Não Balanceado**.

Configuração do Driver

Esta seção contém informações sobre a configuração deste Driver.

Parâmetros [P]

O Driver IEC870-5-101 não utiliza os parâmetros **[P]**. Recomenda-se, por questões de compatibilidade futura, que o valor destes parâmetros seja deixado em 0 (zero).

Janela de Propriedades

A janela de propriedades deste Driver é composta de diversas abas. As abas **IEC870** e **Properties** contém configurações específicas para este Driver. As abas **Setup**, **Serial**, **Ethernet**, **Modem** e **RAS** configuram os logs e o meio de comunicação deste Driver. Para mais informações sobre estas abas, consulte o tópico Documentação das Interfaces de Comunicação.

Aba IEC870

A aba **IEC870** contém as seções descritas a seguir.

Protocol

Caixa de seleção que permite selecionar o protocolo a ser utilizado por este Driver. As opções disponíveis são **IEC 870-5-101 MASTER (balanced)**, **IEC 870-5-101 MASTER (unbalanced)**, **IEC 870-5-101 SLAVE (balanced)** ou **IEC 870-5-101 SLAVE (unbalanced)**. De acordo com o protocolo selecionado são mostradas as aba General, Master ou Slave ou Properties, descritas mais adiante.

Use the Following List of Pre-Defined Stations

A lista de Estações permite pré-configurar as Estações que este Driver utiliza. Nos protocolos em que este Driver está no modo **Mestre**, esta lista contém as Estações do modo **Escravo** com as quais este Driver se comunica. Nos protocolos em que este Driver está no modo **Escravo**, esta lista contém as Estações servidas por este Driver para as Estações do modo **Mestre** na rede. Recomenda-se sempre preencher a Lista de Estações em Drivers no modo **Escravo**.

Opções disponíveis para a Lista de Estações

OPÇÃO	DESCRIÇÃO
Use the following list of pre-defined stations	Se esta opção está habilitada, este Driver suporta apenas as Estações definidas em tempo de configuração. Se esta opção está desabilitada, este Driver cria Estações de forma dinâmica, à medida que Tags com diferentes parâmetros <i>N1</i> ou <i>B1</i> são utilizados
Add	Adiciona uma nova Estação nesta lista. Esta opção mostra uma janela para a configuração das opções Station ID (N1/B1) : Define o parâmetro <i>N1</i> ou <i>B1</i> que endereça esta Estação, Name : Define o parâmetro do tipo String (ParamDevice) que endereça esta Estação, Data Link Address : Define o endereço físico de uma Estação e Common Address : Define o endereço lógico de uma Estação (<i>Common Address of ASDU</i>)
Delete	Remove a Estação selecionada desta lista
Properties	Abre uma janela para editar a configuração da Estação selecionada

Aba General

A aba **General** contém configurações gerais deste Driver. As opções disponíveis nesta aba estão descritas na tabela a seguir.

Opções disponíveis na aba General

OPÇÃO	DESCRIÇÃO
Field sizes (in octets)	Configura o tamanho em octetos, ou bytes, de diversos campos do pacote de dados utilizados por este Driver. Estas configurações devem estar de acordo com a configuração dos equipamentos com os quais este Driver está comunicando. As opções disponíveis são Slave Address : Configura o número de octetos do endereço de uma Estação no pacote FT 1.2 . Os valores permitidos são 1 (um) ou 2 (dois), Common Address of ASDU : Configura o número de octetos do endereço comum da ASDU (endereço de um dispositivo lógico dentro de um equipamento físico). Por padrão esta propriedade segue o número de octetos da opção Slave Address . Os valores permitidos são 1 (um) ou 2 (dois), Cause of Transmission : Configura o número de octetos da causa de transmissão da ASDU. Os valores permitidos são 1 (um) ou 2 (dois). O segundo octeto, quando habilitado, contém o valor do <i>originator address</i> e Information Object Address : Configura o número de octetos do endereço de um objeto utilizado nas ASDUs. Os valores válidos estão no intervalo entre 1 (um) e 3 (três)
Default slave address	Configura o endereço padrão de uma Estação utilizado por este Driver quando o parâmetro <i>N1</i> ou <i>B1</i> de um Tag é igual a 0 (zero)
Force Common Address	Por padrão, quando a lista de Estações não está pré-configurada, este Driver utiliza o mesmo valor da opção Slave Address na opção Common Address of ASDU . Caso seja necessário configurar um valor diferente, esta opção deve ser selecionada e preenchida com o valor adequado
Driver starts in listen mode	Selecione esta opção para que este Driver inicie em modo de Escuta quando executado
Convert the following objects to Boolean values (0 / 1)	Permite reconfigurar alguns tipos de dados para que utilizem valores Booleanos ao invés do valor nativo, entre 0 (zero) e 3 (três). As opções disponíveis são DIQ (used in M_DP_NA_1, M_DP_TA_1 and M_DP_TB_1) : Selecione esta opção para utilizar Pontos Digitais Duplos com valor Booleano, SEP (used in M_EP_TA_1 and M_EP_TD_1) : Selecione esta opção para utilizar eventos de proteção de equipamento com valor Booleano, DCO (used in C_DC_NA_1 and C_DC_TA_1) : Selecione esta opção para utilizar Comandos de Ponto Digital Duplo com valor Booleano ou RCO (used in C_RC_NA_1, and C_RC_TA_1) : Selecione esta opção para utilizar Comandos de Regulação de Passo com valor Booleano

Aba Master

Esta aba está disponível apenas para os protocolos em que este Driver está no modo **Mestre**. As opções disponíveis nesta aba estão descritas na tabela a seguir.

Opções disponíveis na aba Master

OPÇÃO	DESCRIÇÃO
General Interrogation	Configura a forma como um Driver no modo Mestre envia e processa um comando de Interrogação Geral. As opções disponíveis são Auto-run general interrogation : Habilita o envio de um comando de Interrogação Geral sempre que este Driver estabelece uma conexão com um Escravo , Repeat every (sec) : Habilita a repetição do envio de um comando de Interrogação Geral no intervalo especificado, em segundos, Wait ACTTERM : Selecione esta opção para forçar este Driver a aguardar o recebimento de uma sinalização de término de um comando de Interrogação Geral. Alguns Escravos não enviam esta sinalização de término, portanto nestes casos esta opção deve permanecer desmarcada ou Process data only at GI end : Selecione esta opção para processar dados apenas ao final de um comando de Interrogação Geral
Clock Synchronization	Configura o envio automático de um comando de Sincronização de Relógio. As opções disponíveis são Auto-run clock synchronization : Habilita o envio de um comando de Sincronização de Relógio sempre que este Driver estabelece uma conexão com um Escravo ou Repeat every (sec) : Habilita a repetição do envio de um comando de Sincronização de Relógio no intervalo especificado, em segundos
App Layer Timeout (sec)	Define o tempo máximo, em segundos, que este Driver aguarda pelo término do envio de qualquer comando. É importante definir um valor grande o suficiente para permitir que comandos potencialmente demorados, tais como Interrogação Geral, Seleção ou Execução, possam ser executados com sucesso
Link Layer	Define parâmetros de funcionamento da camada de enlace. As opções disponíveis são Link layer retries : Define o número de tentativas de envio de mensagens em caso de erro, Slave polling rate (ms) : Define o tempo, em milissegundos, entre cada varredura enviada para um Escravo , apenas para o protocolo IEC 870-5-101 Não-Balanceado , Try to reactivate failed slaves every (sec) : Define o tempo, em segundos, entre tentativas de reativação de Escravos em falha ou Wait M_EI_NA on activation : Selecione esta opção para forçar este Driver a aguardar o recebimento da ASDU M_EI_NA para que um Escravo seja considerado conectado
Test Command	Configura o envio automático de um comando de Teste. A opção disponível é Send when App Layer is idle (sec) : Habilita o envio periódico de um comando de Teste sempre que este Driver permanecer um determinado tempo, em segundos, sem efetuar comunicações no nível de aplicação

Aba Slave

Esta aba está disponível apenas para os protocolos onde este Driver está no modo **Escravo**. As opções disponíveis nesta aba estão descritas na tabela a seguir.

Opções disponíveis na aba Slave

OPÇÃO	DESCRIÇÃO
Clock synchronization command	Configura a ação que um Driver no modo Escravo executa quando recebe um comando de Sincronização de Relógio
Select/Execute command	Configura a ação que um Driver no modo Escravo executa quando recebe um comando de Seleção ou Execução
Test command	Configura a ação que um Driver no modo Escravo executa quando recebe um comando de Teste
Event buffering while not connected	Permite habilitar o armazenamento de eventos na memória deste Driver enquanto um Mestre não está conectado. Por padrão este Driver mantém apenas o valor atual de cada Tag em memória. Ao habilitar o uso de <i>buffers</i> , este Driver passa a manter também uma fila dos últimos <i>n</i> eventos de cada classe e envia estes eventos quando um Mestre se conecta. As opções disponíveis são 'class1' buffer (events) : Habilita o armazenamento em memória de eventos de Classe 1 (um, prioridade alta, tipicamente valores digitais) ou 'class2' buffer (events) : Habilita o armazenamento em memória de eventos de Classe 2 (dois, prioridade baixa, tipicamente valores analógicos)
General Interrogation	Configura o conjunto de dados que é retornado para um Mestre em resposta a um comando de Interrogação Geral. As opções disponíveis são Return a pre-defined data list : Habilita a edição da lista pré-definida de dados a serem retornados, Check : Valida a lista de dados pré-definidos ou Remove timestamp from GI data : Selecione esta opção para remover a estampa de tempo de todos os dados enviados em um comando de Interrogação Geral
Enable ACK using CC1	Habilita o uso do caractere especial CC1 (0xE5) como resposta positiva no protocolo IEC 870-5-101 Balanceado

Aba Properties

A aba **Properties** permite visualizar e configurar todas as propriedades de configuração deste Driver, agrupadas por categoria. Algumas propriedades podem ser editadas apenas na aba **Properties**, outras possuem um campo correspondente para edição na aba IEC870 e nas abas General, Master e Slave. Esta lista mostra em negrito as propriedades que estão com valor diferente do padrão. Para retornar o valor de uma propriedade para o padrão, utilize a tecla DEL.

Estações

Nos protocolos onde este Driver está no modo **Mestre**, uma Estação representa um **Escravo** com o qual este Driver se comunica. Nos protocolos onde este Driver está no modo **Escravo**, uma Estação representa um dispositivo lógico servido por este Driver com o qual um **Mestre** externo se comunica.

As Estações devem ser declaradas na aba IEC870 da janela de propriedades. Selecione a opção **Use the following list of pre-defined stations** para declarar explicitamente uma lista de Estações que este Driver utiliza. Se esta opção está desmarcada, este Driver opera de forma automática criando Estações com parâmetros padrão à medida que sejam necessárias. O modo automático é desaconselhado para Drivers no modo **Escravo**, pois Estações incorretas podem ser criadas de acordo com as requisições realizadas por um **Mestre**.

Lista Pré-Definida de Estações

Para configurar uma lista de Estações, selecione a aba **IEC870** na janela de propriedades deste Driver e selecione a opção **Use the following list of pre-defined stations**. Se este Driver está no modo **Escravo**, apenas uma Estação deve ser criada. Se este Driver está no modo **Mestre**, mais de uma Estação pode ser criada.

Para criar uma Estação, clique em **Add** e configure a janela Station Properties da seguinte forma:

- **Station ID (N1/B1):** Informe o identificador de uma Estação, utilizado pelos parâmetros *N1* ou *B1* dos Tags. Este valor é interno deste Driver e deve ser um valor entre 0 (zero) e 65535
- **Name (ParamDevice):** Informe o nome de uma Estação, utilizado na propriedade **ParamDevice** dos Tags, caso se utilize a configuração de Tags por **Strings**
- **Data Link Address:** Utilizado como endereço físico de uma Estação
- **Common Address:** Para uso em todos os protocolos, corresponde à opção **Common Address of ASDU**, que identifica um dispositivo lógico de origem ou destino da ASDU

Para remover uma Estação desta lista, selecione a Estação e clique em **Delete**. Para editar a configuração de uma Estação, selecione a Estação e clique em **Properties**.

NOTA

Este Driver atualmente suporta a comunicação com apenas um dispositivo lógico para cada dispositivo físico, isto é, o valor da opção **Data Link Address** de cada Estação declarada deve ser único.

Criação Automática de Estações

Em alguns cenários pode ser necessário ou conveniente deixar este Driver criar as Estações automaticamente em tempo de execução. Para utilizar a criação automática de Estações, desmarque a opção **Use the following list of pre-defined stations**. na aba **IEC870** da janela de configurações.

Quando este Driver está ativo, as Estações são criadas automaticamente sempre de acordo com o parâmetro *N1* ou *B1* de cada Tag. O parâmetro *N1* ou *B1* corresponde à opção **Data Link Address** e também à opção **Common Address** de uma Estação. Se a opção **Force Common Address** está selecionada, então a opção **Common Address** de todas as Estações criadas automaticamente é igual ao valor configurado para o valor forçado.

Nos protocolos no modo **Escravo**, uma nova Estação também é criada de acordo com os pedidos recebidos de um **Mestre**, que podem não ter relação com os Tags de uma aplicação. Por esta razão, não é recomendado utilizar a criação automática de Estações com os protocolos no modo **Escravo**.

Tags de Estatística e Diagnóstico de uma Estação

Os Tags desta seção permitem acessar diagnósticos e estatísticas mantidos por este Driver para cada Estação.

NOTAS

- Os Tags desta seção estão disponíveis para todos os tipos de protocolo.
- Nenhum dos Tags desta seção pode ser configurado usando **Strings**.
- Nenhuma ASDU (*Application Service Data Unit*) está disponível para os Tags desta seção.

Tag de Estado de Conexão de uma Estação

Somente Leitura

N1	Número de uma Estação
N2	992
N3	0 (zero)
N4	0 (zero)

Retorna o estado de conexão de uma Estação. Os valores possíveis são **0**: Estação não conectada ou não respondendo à camada de enlace ou **1**: Estação conectada.

Tag Bloco de Leitura do Log de Eventos

Somente Leitura

B1	Número de uma Estação
B2	99
B3	0 (zero)
B4	0 (zero)

Retorna o valor, a qualidade e a estampa de tempo dos dados recebidos de uma Estação. Use o evento **OnRead** deste Tag para analisar todas as ASDUs recebidas por uma Estação. Os Elementos deste Tag Bloco são os seguintes:

- **Elemento 0 (Word):** Tipo de dados ou ASDU (*type*)
- **Elemento 1 (Word):** Endereço de um objeto
- **Elemento 2 (DWord):** *Flag* 1 (um)
- **Elemento 3 (DWord):** *Flag* 2 (dois)
- **Elemento 4 (Double):** Valor de um objeto
- **Elemento 5 (Double):** Estampa de tempo
- **Elemento 6 (String):** Nome da ASDU
- **Elemento 7 (Word):** Causa da transmissão (COT ou *Cause of Transmission*)

Tag de Estatísticas do Contador de Pacotes Enviados

Somente Leitura

N1	Número de uma Estação
N2	998
N3	0 (zero)
N4	1 (um)

Retorna o número de pacotes de comunicação já enviados por este Driver para uma Estação.

Tag de Estatísticas do Contador de No-Reply

Somente Leitura

N1	Número de uma Estação
N2	998
N3	0 (zero)
N4	2 (dois)

Retorna o número de pacotes de comunicação enviados por este Driver sem resposta por parte de uma Estação.

Tag de Estatísticas do Contador de Erros de Frame Recebidos

Somente Leitura

N1	Número de uma Estação
N2	998
N3	0 (zero)
N4	3 (três)

Retorna o número de pacotes recebidos com erro de uma Estação.

Tag de Estatísticas do Contador de Erros de Envio de Frames

Somente Leitura

N1	Número de uma Estação
N2	998
N3	0 (zero)
N4	4 (quatro)

Retorna o número de falhas de envio de pacotes para uma Estação.

Tag de Estatísticas do Contador de Retentativas de Envio de Pacotes

Somente Leitura

N1	Número de uma Estação
N2	998
N3	0 (zero)
N4	5 (cinco)

Retorna o número de retentativas de envio de pacotes para uma Estação.

Tag Bloco de Estatísticas Avançadas

Somente Leitura

B1	Número de uma Estação
B2	999
B3	0 (zero)
B4	0 (zero)

Retorna um Tag Bloco com o valor atual de diversas estatísticas. Os Elementos deste Tag Bloco são os seguintes:

- **Elemento 0 (Long):** Número de ASDUs criados
- **Elemento 1 (Long):** Número de ASDUs removidos
- **Elemento 2 (Long):** Número de ASDUs inválidos
- **Elemento 3 (Long):** Número de ASDUs na memória *cache* interna de eventos de um **Escravo**
- **Elemento 4 (Long):** Número de ASDUs na memória *cache* interna de dados de um **Escravo**
- **Elemento 5 (Long):** Número de ASDUs na memória *cache* de logs de um **Escravo**
- **Elemento 6 (Long):** Número de HVALUES alocados
- **Elemento 7 (Long):** Número de referências para HVALUES
- **Elemento 8 (Long):** Número de HTIMERs alocados
- **Elemento 9 (Long):** Número de categorias de *timers*
- **Elemento 10 (Long):** Número de HVALUES criados
- **Elemento 11 (Long):** Número de bytes usados no *heap* de um processo
- **Elemento 12 (Long):** Número de blocos de *heap* em um processo
- **Elemento 13 (Long):** Número de bytes disponíveis no *heap* de um processo
- **Elemento 14 (Long):** Número de blocos de *heap* disponíveis em um processo
- **Elemento 15 (Long):** Número de processos no sistema
- **Elemento 16 (Long):** Número de *threads* no sistema

Protocolo IEC 870-5-101

Este Driver suporta o protocolo **IEC 870-5-101** nos modos **Balanceado** e **Não Balanceado**, tanto no modo **Mestre** quanto no modo **Escravo**.

Tipos de Dados de Monitoramento

Este Driver suporta os tipos de dados de monitoramento, ou leitura, descritos na tabela a seguir.

Tipos de dados de monitoramento suportados

TIPO DE DADOS	MNEMÔNICO	ASDU SEM TIMESTAMP	ASDU COM TIMESTAMP DE 24 BITS	ASDU COM TIMESTAMP DE 56 BITS
SP (Ponto Digital Simples)	SP	M_SP_NA_1 (1)	M_SP_TA_1 (2)	M_SP_TB_1 (30)
DP (Ponto Digital Duplo)	DP	M_DP_NA_1 (3)	M_DP_TA_1 (4)	M_DP_TB_1 (31)
ST (Posição de Passo)	ST	M_ST_NA_1 (5)	M_ST_TA_1 (6)	M_ST_TB_1 (32)
BO (Bitstring)	BO	M_BO_NA_1 (7)	M_BO_TA_1 (8)	M_BO_TB_1 (33)
ME (Valor de Medida Analógica)	MEn MEs MEf	M_ME_NA_1 (9) M_ME_NB_1 (11) M_ME_NC_1 (13)	M_ME_TA_1 (10) M_ME_TB_1 (12) M_ME_TC_1 (14)	M_ME_TD_1 (34) M_ME_TE_1 (35) M_ME_TF_1 (36)
IT (Total Integrado)	IT	M_IT_NA_1 (15)	M_IT_TA_1 (16)	M_IT_TB_1 (37)

Os dados de monitoramento são enviados de um **Escravo** para um **Mestre**. Em um Driver no modo **Escravo** é necessário que uma aplicação alimente este Driver com os valores a enviar para um **Mestre**, e isto é realizado através de escritas. Em um Driver no modo **Mestre**, os valores recebidos são fornecidos para uma aplicação através de leituras.

SP (Ponto Digital Simples)

Um **Ponto Digital Simples** é um objeto que armazena um valor **ON** (1, um) ou **OFF** (0, zero) e corresponde às ASDUs **1: M_SP_NA_1** sem estampa de tempo, **2: M_SP_TA_1** com estampa de tempo de 24 bits e **30: M_SP_TB_1** com estampa de tempo de 56 bits.

Os valores permitidos para este tipo de dados são **0**: OFF ou **1**: ON (na escrita qualquer valor diferente de zero é enviado como ON).

Este objeto suporta os bits de qualidade **IV**, **NT**, **SB** e **BL**.

DP (Ponto Digital Duplo)

Um **Ponto Digital Duplo** é um objeto que armazena um valor **ON** ou **OFF**, cada estado corresponde a um bit individual e corresponde às ASDUs **3: M_DP_NA_1** sem estampa de tempo, **4: M_DP_TA_1** com estampa de tempo de 24 bits e **31: M_DP_TB_1** com estampa de tempo de 56 bits.

No protocolo, o valor trafegado nestas ASDUs é um DIQ (*Double Point Information with Quality Descriptor*) que contém 4 (quatro) bits de qualidade (**IV**, **NT**, **SB** e **BL**) e 2 (dois) bits de valor, chamados de DPI (*Double Point Information*):

- **0:** Estado intermediário (inválido)
- **1:** OFF
- **2:** ON
- **3:** Estado intermediário (inválido)

O valor dos Tags de Ponto Digital Duplo neste Driver depende da configuração da opção **DIQ (used in M_DP_NA_1, M_DP_TA_1 and M_DP_TB_1)** do grupo Convert the following objects to Boolean values (0 / 1) na aba **General** da aba **IEC870**. Se esta opção está desmarcada, o valor deste Tag nas leituras é o próprio valor do DPI, tanto nas leituras quanto nas escritas. As escritas neste Driver falham se o valor não é um dos seguintes valores:

- **0:** Estado intermediário (inválido)
- **1:** OFF
- **2:** ON
- **3:** Estado intermediário (inválido)

Se a opção Convert the following objects to Boolean values (0 / 1) está selecionada, o valor dos Tags de Ponto Digital Duplo nas leituras é o seguinte:

- **DPI igual a 0:** Retorna **Null** e qualidade **Incerta**
- **DPI igual a 1:** Retorna 0 (zero)
- **DPI igual a 2:** Retorna 1 (zero)
- **DPI igual a 3:** Retorna **Null** e qualidade **Ruim**

Nas escritas, qualquer valor diferente de 0 (zero) envia um DPI igual a 2 (dois) e o valor 0 (zero) neste Tag envia um DPI igual a 1 (um).

ST (Posição de Passo)

Uma **Posição de Passo** é um objeto que armazena um valor inteiro no intervalo entre -64 e 63. Corresponde às ASDUs **5: M_ST_NA_1** sem estampa de tempo, **6: M_ST_TA_1** com estampa de tempo de 24 bits e **32: M_ST_TB_1** com estampa de tempo de 56 bits. Este objeto suporta os bits de qualidade **IV**, **NT**, **SB**, **BL** e **OV**.

NOTA

O *flag T* (transiente) não é suportado por este Driver, é ignorado na leitura e configurado para 0 (zero) na escrita.

BO (Bitstring)

Um **Bitstring** é um objeto que armazena uma sequência de 32 bits. Corresponde às ASDUs **7**: **M_BO_NA_1** sem estampa de tempo, **8**: **M_BO_TA_1** com estampa de tempo de 24 bits e **33**: **M_BO_TB_1** com estampa de tempo de 56 bits. Este Driver trata os valores de **Bitstring** como inteiros de 32 bits. Este objeto suporta os bits de qualidade **IV**, **NT**, **SB**, **BL** e **OV**.

ME (Valor de Medida Analógica)

As **Medidas Analógicas** podem ser representadas no formatos Normalizado, Com Escala ou Ponto Flutuante.

MEn (Medida Analógica Normalizada)

Uma **Medida Analógica Normalizada** é um objeto que armazena um valor em ponto flutuante no intervalo entre -1.0 e 1.0. Corresponde às ASDUs **9**: **M_ME_NA_1** sem estampa de tempo, **10**: **M_ME_TA_1** com estampa de tempo de 24 bits e **34**: **M_ME_TD_1** com estampa de tempo de 56 bits.

O intervalo de valores válidos neste Tag vai de -1.0 a 1.0 e de 2 (dois) a 15. A precisão deste valor é de 16 bits, portanto nem todos os valores dentro do intervalo podem ser representados com precisão, portanto podem ocorrer aproximações nos valores enviados e recebidos. Este objeto suporta os bits de qualidade **IV**, **NT**, **SB**, **BL** e **OV**.

MEs (Medida Analógica com Escala)

Uma **Medida Analógica com Escala** é um objeto que armazena um valor de 16 bits com sinal com um ponto decimal fixo. Corresponde às ASDUs **11**: **M_ME_NB_1** sem estampa de tempo, **12**: **M_ME_TB_1** com estampa de tempo de 24 bits e **35**: **M_ME_TE_1** com estampa de tempo de 56 bits.

Neste Driver o ponto decimal de todas as Medidas Analógicas com escala é fixo em uma casa decimal. Desta forma, o intervalo de valores válidos para os Tags deste tipo de dados varia entre -3276,8 a 3276,7. Se o ponto decimal de um equipamento é diferente de uma casa decimal, que é fixo neste Driver, então uma escala apropriada tem que ser configurada neste Tag. Este objeto suporta os bits de qualidade **IV**, **NT**, **SB**, **BL** e **OV**.

MEf (Medida Analógica em Ponto Flutuante)

Uma **Medida Analógica em Ponto Flutuante** é um objeto que armazena um valor em ponto flutuante de 32 bits no formato **IEEE STD 754** (*Short Floating Point Number*). Corresponde às ASDUs **13**: **M_ME_NC_1** sem estampa de tempo, **14**: **M_ME_TC_1** com estampa de tempo de 24 bits e **36**: **M_ME_TF_1** (com estampa de tempo de 56 bits. Este objeto suporta os bits de qualidade **IV**, **NT**, **SB**, **BL** e **OV**.

IT (Total Integrado)

Um **Total Integrado** é um objeto que armazena o valor de um contador de 32 bits com sinal. Corresponde às ASDUs **15**: **M_IT_NA_1** sem estampa de tempo, **16**: **M_IT_TA_1** com estampa de tempo de 24 bits e **37**: **M_IT_TB_1** com estampa de tempo de 56 bits.

Este Driver trata os valores de Total Integrado como inteiros de 32 bits com sinal e suporta o bit de qualidade **IV**.

Os campos **SQ** (número de sequência), **CY** (*overflow* do contador) e **CA** (contador ajustado) são ignorados por este Driver na recepção e enviados com o valor 0 (zero) na transmissão.

Configuração de Tags por String

Para utilizar a configuração por **String**, os parâmetros *N* de um Tag devem estar zerados. Deve-se colocar o endereço de um equipamento na propriedade **ParamDevice** e o endereço do objeto de monitoramento na propriedade **ParamItem**, de acordo com a sintaxe a seguir.

```
<tipo>:<endereçoObjeto>[<flagPrioridade>]
```

Em que:

- **<tipo>** é um mnemônico do tipo de dados, ou seja, **SP**, **DP**, **ST**, **BO**, **MEn**, **MEs** ou **MEf**. Os dois pontos que separam um mnemônico do endereço de um objeto são opcionais. A tabela a seguir contém os tipos de dados de monitoramento e os respectivos mnemônicos

Tipos de dados e mnemônicos

TIPO DE DADOS	ASDU	MNEMÔNICO	EXEMPLO
Ponto Digital Simples	M_SP_NA_1 (1) M_SP_TA_1 (2) M_SP_TB_1 (30)	SP	SP:1
Ponto Digital Duplo	M_DP_NA_1 (3) M_DP_TA_1 (4) M_DP_TB_1 (31)	DP	DP:328
Posição de Passo	M_ST_NA_1 (5) M_ST_TA_1 (6) M_ST_TB_1 (32)	ST	ST:10.434
Bitstring	M_BO_NA_1 (7) M_BO_TA_1 (8) M_BO_TB_1 (33)	BO	BO:99
Medida Analógica Normalizada	M_ME_NA_1 (9) M_ME_TA_1 (10) M_ME_ND_1 (21) M_ME_TD_1 (34)	MEn	MEn:10938
Medida Analógica com Escala	M_ME_NB_1 (11) M_ME_TB_1 (12) M_ME_TE_1 (35)	MEs	MEs:9833

TIPO DE DADOS	ASDU	MNEMÔNICO	EXEMPLO
Medida Analógica em Ponto Flutuante	M_ME_NC_1 (13) M_ME_TC_1 (14) M_ME_TF_1 (36)	MEf	MEf:10.35.22

- **<endereçoObjeto>** é o endereço de um objeto de monitoramento, conforme a tabela a seguir.

Formatos dos endereços de objetos

INFORMATION OBJECT ADDRESS	FORMATOS SUPORTADOS	EXEMPLO	OCTETOS GERADOS
Um octeto	<0-255>	SP:255	FF
Dois octetos	<0-65535>	SP:65535	FF FF
	<0-255>.<0-255>	SP:1.2	01 02
Três octetos	<0-16777215>	SP:16777215	FF FF FF
	<0-255>.<0-65535>	SP:1.65535	01 FF FF
	<0-255>.<0-255>.<0-255>	SP:26.5.3	1A 05 03

- **<flagPrioridade>** é um caractere opcional que permite definir a prioridade com que um dado é enviado, somente no caso de protocolos no modo **Escravo**, conforme a tabela a seguir.

Flags de prioridade

FLAG	DESCRIÇÃO	EXEMPLO
Omitido	Um dado é enviado com a prioridade padrão, ou seja, dados digitais com prioridade alta e dados analógicos com prioridade baixa. A prioridade padrão pode ser redefinida na aba Properties das configurações deste Driver, no grupo Send Priority	SP:255
!	Quando o caractere é um ponto de exclamação, um dado é enviado com prioridade Alta (Classe 1)	SP:255!
<	Quando o caractere é um sinal de menor que, um dado é enviado apenas na próxima Interrogação Geral	SP:255<
=	Quando o caractere é um sinal de igual, um dado é enviado com prioridade Baixa (Classe 2)	SP:255=

Tipos de Dados de Comando

Este Driver suporta os tipos de dados em comandos de Seleção ou Execução descritos na tabela a seguir.

Tipos de dados em comandos de Seleção ou Execução

TIPO DE DADOS	MNEMÔNICO	ASDU SEM TIMESTAMP	ASDU COM TIMESTAMP DE 56 BITS
SC (Ponto Digital Simples)	SC	C_SC_NA_1 (45)	C_SC_TA_1 (58)
DC (Ponto Digital Duplo)	DC	C_DC_NA_1 (46)	C_DC_TA_1 (59)
RC (Regulação de Passo)	RC	C_RC_NA_1 (47)	C_RC_TA_1 (60)
SE (Setpoint)	SEn SEs SEf	C_SE_NA_1 (48) C_SE_NB_1 (49) C_SE_NC_1 (50)	C_SE_TA_1 (61) C_SE_TB_1 (62) C_SE_TC_1 (63)
BO (Bitstring)	BO	C_BO_NA_1 (51)	C_BO_TA_1 (64)

Todos os comandos possuem em comum os campos a seguir, que são enviados junto com o valor de um comando:

- **SE:** Determina se é um comando de **Seleção** (1, um) ou **Execução** (0, zero)
- **QU:** Determina o qualificador do comando:
- **0:** Sem definição
- **1:** Pulso de duração curta (*circuit breaker*)
- **2:** Pulso de longa duração
- **3:** Saída persistente
- **4 a 8:** Reservado para novas definições do padrão IEC
- **9 a 15:** Reservado para outras funções pré-definidas
- **16 a 31:** Reservado para uso especial (intervalo privado)

Um Driver no modo **Mestre** suporta apenas o envio de comandos com **QU** igual a 0 (zero), 1 (um), 2 (dois) ou 3 (três).

SC (Comando de Ponto Digital Simples)

Um **Comando de Ponto Digital Simples** utiliza as ASDUs **45: C_SC_NA_1** sem estampa de tempo e **58: C_SC_TA_1** com estampa de tempo de 56 bits. Os valores permitidos para este tipo de dados são **0:** OFF ou **1:** ON.

DC (Comando de Ponto Digital Duplo)

Um **Comando de Ponto Digital Duplo** corresponde às ASDUs **46: C_DC_NA_1** sem estampa de tempo e **59: C_DC_TA_1** com estampa de tempo de 56 bits.

No protocolo, o valor trafegado nestas ASDUs é um DCO (*Double Command*) que contém os campos **SE** e **QU**, além de 2 (dois) bits de valor DCS (*Double Command State*):

- **0**: Estado intermediário (inválido)
- **1**: OFF
- **2**: ON
- **3**: Estado intermediário (inválido)

O valor de um Comando de Ponto Digital depende da configuração realizado na opção **DCO (used in C_DC_NA_1 and C_DC_TA_1)** do grupo Convert the following objects to Boolean values (0 / 1) na aba **General** da aba **IEC870**. Se esta opção está desmarcada, o valor escrito é o próprio valor do DCS:

- **0**: Estado intermediário (inválido)
- **1**: OFF
- **2**: ON
- **3**: Estado intermediário (inválido)

Se a opção de tratar como um Booleano está selecionada, o valor do comando é **0**: OFF (DPI igual a um) ou **1**: ON (DPI igual a dois).

RC (Comando de Regulação de Passo)

Um **Comando de Regulação de Passo** permite incrementar ou decrementar uma Posição de Passo. Corresponde às ASDUs **47: C_RC_NA_1** sem estampa de tempo e **60: C_RC_TA_1** com estampa de tempo de 56 bits.

No protocolo, o valor trafegado nestas ASDUs é um RCO (*Regulating Step Command*) que contém 1 (um) bit **SE**, que indica se é um comando **Select** ou **Execute**, 5 (cinco) bits **QU**, que indicam um qualificador de comando, como por exemplo pulso curto ou pulso longo, e 2 (dois) bits de valor RCS (*Regulating Step Command State*):

- **0**: Não permitido
- **1**: Próximo passo abaixo (decrementa)
- **2**: Próximo passo acima (incrementa)
- **3**: Não permitido

O valor de um Comando de Regulação de Passo depende da configuração realizada na opção **RCO (used in C_RC_NA_1 and C_RC_TA_1)** do grupo Convert the following objects to Boolean values (0 / 1) na aba **General** da aba **IEC870**. Se esta opção está desmarcada, o valor escrito é o próprio valor do RCS:

- **0**: Não permitido
- **1**: Próximo passo abaixo (decrementa)
- **2**: Próximo passo acima (incrementa)

- 3: Não permitido

Se esta opção de tratar como um Booleano está selecionada, o valor deste comando é **0**: Próximo passo abaixo (decrementa, RCS igual a um) ou **1**: Próximo passo acima (incrementa, RCS igual a dois).

SE (Comando de Setpoint ou Analógico)

Os **Comandos de Setpoint**, ou escrita em Medidas Analógicas, podem ser efetuado usando os valores Normalizado, Com Escala ou Ponto Flutuante.

SEn (Comando de Setpoint, Valor Normalizado)

Um **Comando de Setpoint** permite o envio de um valor normalizado no intervalo entre -1.0 e 1.0 para um **Escravo**. Corresponde às ASDUs **48**: **C_SE_NA_1** sem estampa de tempo e **61**: **C_SE_TA_1** com estampa de tempo de 56 bits.

Valores informados acima e abaixo dos limites permitidos são colados no limite mais próximo. A precisão deste valor é de 16 bits, portanto nem todos os valores dentro deste intervalo podem ser representados com precisão, portanto podem ocorrer aproximações nos valores enviados e recebidos.

SEs (Comando de Setpoint, Valor com Escala)

Um **Comando de Setpoint** permite o envio de um valor com escala no intervalo entre -3276,8 e 3276,7 para um **Escravo**. Corresponde às ASDUs **49**: **C_SE_NB_1** sem estampa de tempo e **62**: **C_SE_TB_1** com estampa de tempo de 56 bits.

O valor que realmente trafega no protocolo corresponde ao valor escrito multiplicado por 10, pois assume-se uma casa decimal fixa, resultando no envio de um valor inteiro na faixa entre -32768 e 32767.

SEf (Comando de Setpoint, Valor em Ponto Flutuante)

Um **Comando de Setpoint** permite o envio de um valor em ponto flutuante de 32 bits no formato **IEEE STD 754** (*Short Floating Point Number*). Corresponde às ASDUs **50**: **C_SE_NC_1** sem estampa de tempo e **63**: **C_SE_TC_1** com estampa de tempo de 56 bits.

BO (Comando de Bitstring)

Um **Comando de Bitstring** permite o envio de um valor inteiro de 32 bits. Corresponde às ASDUs **51**: **C_BO_NA_1** sem estampa de tempo e **64**: **C_BO_TA_1** com estampa de tempo de 56 bits.

Configuração de Comandos por String

Para usar a configuração de comandos por **Strings**, somente para protocolos no modo **Mestre**, os parâmetros *N* deste Tag devem estar zerados. Deve-se colocar o endereço de um equipamento na propriedade **ParamDevice** e a descrição de um comando na propriedade **ParamItem** com a sintaxe a seguir.

```
<comando>:<endereçoObjeto>.<ação>[(<qualificador>)]
```

Em que:

- **<comando>** é um mnemônico do tipo de comando, ou seja, **CBO**, **CBOt**, **DC**, **DCt**, **RC**, **RCt**, **SC**, **SCt**, **SEn**, **SEnt**, **SEs**, **SEst**, **SEf** ou **SEft**. Os dois pontos que separam o mnemônico do endereço de um objeto são opcionais. A tabela a seguir mostra os tipos de dados de comando e os respectivos mnemônicos

Tipos de dados e mnemônicos

TIPO DE COMANDO	ASDU	MNEMÔNICO	EXEMPLO
Comando de Ponto Digital Simples	C_SC_NA_1 (45) C_SC_TA_1 (58)	SC SCt	SC:1.Select SCt:1.Select
Comando de Ponto Digital Duplo	C_DC_NA_1 (46) C_DC_TA_1 (59)	DC DCt	DC:328.Execute DCt:328.Execute
Comando de Regulação de Passo	C_RC_NA_1 (47) C_RC_TA_1 (60)	RC RCt	RC:10.434.Execute RCt:10.434.Execute
Comando de Set-Point, valor Normalizado	C_SE_NA_1 (48) C_SE_TA_1 (61)	SEn SEnt	SEn:10938.Execute(ShortPulse) SEnt:10938.Execute(ShortPulse)
Comando de Set-Point, valor com Escala	C_SE_NB_1 (49) C_SE_TB_1 (62)	SEs SEst	SEs:9833.Execute SEst:9833.Execute
Comando de Set-Point, valor em Ponto Flutuante	C_SE_NC_1 (50) C_SE_TC_1 (63)	SEf SEft	SEf:10.35.22.Execute SEft:10.35.22.Execute
Comando de Bitstring	C_BO_NA_1 (51) C_BO_TA_1 (64)	CBO CBOt	CBO:99.Select CBOt:99.Select

NOTA

Os mnemônicos terminados em "t" correspondem a comandos com estampa de tempo. Os demais mnemônicos não possuem estampa de tempo.

- **<endereçoObjeto>** é o endereço de um objeto de comando, que pode estar nos formatos descritos na tabela a seguir

Formatos dos endereços de objetos

INFORMATION OBJECT ADDRESS	FORMATOS SUPORTADOS	EXEMPLO	OCTETOS GERADOS
Um octeto	<0-255>	SC:255.Select	FF
Dois octetos	<0-65535>	SC:65535.Select	FF FF
	<0-255>.<0-255>	SC:1.2.Select	01 02
Três octetos	<0-16777215>	SC:16777215.Select	FF FF FF
	<0-255>.<0-65535>	SC:1.65535.Select	01 FF FF
	<0-255>.<0-255>.<0-255>	SC:26.5.3.Select	1A 05 03

- **<ação>** define um tipo de comando, que pode ser **Select**: Envia um Comando de Seleção, **Execute**: Envia um Comando de Execução ou **Deactivate**: Envia um Comando de Cancelamento de Seleção
- **<qualificador>** é um parâmetro opcional e deve ser informado entre parênteses. Os valores possíveis são **(LongPulse)**: Envia um comando de pulso longo, **(ShortPulse)**: Envia um comando de pulso curto, **(PersistentOutput)**: Envia um comando de saída persistente ou **Omitido**: Envia um comando normal

Mestre 101

Esta seção cobre o funcionamento deste Driver utilizando os protocolos **IEC 870-5-101 Mestre Balanceado** e **IEC 870-5-101 Mestre Não Balanceado**. Nestes protocolos, este Driver assume o papel de **Mestre** na comunicação, suportando as funções a seguir:

- Leitura de Dados de Monitoramento
- Sincronização de Relógio
- Interrogação Geral
- Interrogação de Contadores
- Comando de Seleção ou Execução
- Comando de Reinício de Processo

- Comando de Teste

Leitura de Dados de Monitoramento

Os protocolos **IEC 870-5-101 Mestre Balanceado** e **IEC 870-5-101 Mestre Não Balanceado** permitem o recebimento de dados de monitoramento enviados por um equipamento de forma espontânea, ou como resposta a algum comando enviado por este Driver, como por exemplo uma Interrogação Geral ou uma Interrogação de Contadores. Os dados enviados por um equipamento são armazenados em uma memória interna deste Driver, chamada de Cache, e permanecem à disposição até que sejam lidos por um Tag de leitura de memória *cache*.

NOTA

A memória *cache* armazena no máximo 100 valores não lidos para cada Tag. Se o número de eventos pendentes de leitura para um determinado Tag exceder este limite, os eventos mais antigos são descartados para dar lugar aos eventos mais novos.

As formas de ler os dados da memória *cache* são Leitura de Eventos: Retorna todos os eventos recebidos para um Tag. Se nenhum evento está disponível, retorna uma leitura vazia ou Leitura de Valor Corrente: Retorna uma cópia do último valor recebido para um Tag. Se nenhum valor está disponível, retorna um erro.

Tag de Leitura de Eventos da Memória Cache

Somente Leitura

N1	Número de uma Estação
N2	4 (quatro), (cinco) ou 6 (seis) para leitura normal ou 7 (sete) para leitura com <i>reset</i> automático
N3	ASDU
N4	Endereço de um objeto

Retorna o valor, a qualidade e a estampa de tempo de eventos armazenados na memória *cache* interna deste Driver. Depois que um evento é retornado, este é removido da memória *cache*. Se esta memória está vazia, nenhum valor é retornado para este Tag, ou seja, a leitura não afeta o valor armazenado neste Tag. O evento **OnRead** deste Tag é ativado sempre que um valor novo é recebido, e pode ser usado para armazenar os eventos em um arquivo histórico, por exemplo. O parâmetro *N2* tem os seguintes significados:

- **4, 5 ou 6:** Leitura de eventos da memória *cache*, apenas retorna os valores que este Driver recebeu de um **Escravo**
- **7:** Leitura de eventos da memória *cache* com *reset* automático. Para os tipos de dados digitais, ou seja, Ponto Simples e Ponto Duplo, para cada valor **ON** recebido um evento **OFF** é gerado um milissegundo depois deste evento **ON**

Os valores possíveis para o parâmetro *N3* são os seguintes:

- **Para o protocolo 101:** **M_SP_NA_1** (1), **M_SP_TA_1** (2), **M_DP_NA_1** (3), **M_DP_TA_1** (4), **M_ST_NA_1** (5), **M_ST_TA_1** (6), **M_BO_NA_1** (7), **M_BO_TA_1** (8), **M_ME_NA_1** (9), **M_ME_TA_1** (10), **M_ME_NB_1** (11), **M_ME_TB_1** (12), **M_ME_NC_1** (13), **M_ME_TC_1** (14), **M_IT_NA_1** (15), **M_IT_TA_1** (16), **M_PS_NA_1** (20), **M_ME_ND_1** (21) e **M_EI_NA_1**

(70)

- Para o protocolo 104: **M_SP_TB_1** (30), **M_DP_TB_1** (31), **M_ST_TB_1** (32), **M_BO_TB_1** (33), **M_ME_TD_1** (34), **M_ME_TE_1** (35), **M_ME_TF_1** (36) e **M_IT_TB_1** (37)

A leitura deste Tag não causa nenhum tipo de comunicação, sempre é uma leitura imediata. Este Tag não suporta leituras em bloco.

NOTAS

- Recomenda-se que a Interrogação Geral seja habilitada para forçar uma atualização periódica de todos os valores enviados por um **Escravo**.
- Este Tag não pode ser configurado por **Strings**.

Tag de Leitura de Valor Corrente da Memória Cache

Somente Leitura

N1	Número de uma Estação
N2	1 (um)
N3	ASDU
N4	Endereço de um objeto

Lê da memória *cache* interna deste Driver o valor corrente, a qualidade e a estampa de tempo do ponto informado. A diferença deste Tag para o Tag de Leitura de Eventos da Memória Cache é que, se vários eventos são recebidos para um mesmo ponto, apenas o mais recente é retornado. Por exemplo, se o *scan* deste Tag está configurado para 5 (cinco) segundos e este Driver recebe dois eventos para este Tag, ou seja, muda para o valor 1 (um) e depois para 0 (zero), este Driver reporta apenas o evento cujo valor mudou para 0 (zero), pois apenas o valor atual é reportado. As alterações que ocorrem entre leituras deste Tag são descartadas.

Use este Tag apenas se o ponto a monitorar é analógico e se apenas o valor atual é importante. Nos demais casos, recomenda-se sempre utilizar o Tag de Leitura de Eventos da Memória Cache. Este Tag suporta leitura em bloco, isto é, os parâmetros definem o endereço do primeiro objeto a ler.

Os valores possíveis para o parâmetro *N3* são os seguintes:

- Para o protocolo 101: **M_SP_NA_1** (1), **M_SP_TA_1** (2), **M_DP_NA_1** (3), **M_DP_TA_1** (4), **M_ST_NA_1** (5), **M_ST_TA_1** (6), **M_BO_NA_1** (7), **M_BO_TA_1** (8), **M_ME_NA_1** (9), **M_ME_TA_1** (10), **M_ME_NB_1** (11), **M_ME_TB_1** (12), **M_ME_NC_1** (13), **M_ME_TC_1** (14), **M_IT_NA_1** (15), **M_IT_TA_1** (16), **M_PS_NA_1** (20), **M_ME_ND_1** (21) e **M_EI_NA_1** (70)
- Para o protocolo 104: **M_SP_TB_1** (30), **M_DP_TB_1** (31), **M_ST_TB_1** (32), **M_BO_TB_1** (33), **M_ME_TD_1** (34), **M_ME_TE_1** (35), **M_ME_TF_1** (36) e **M_IT_TB_1** (37)

NOTAS

- Este Tag retorna erro se o dado informado ainda não recebeu nenhuma informação.
- Este Tag não pode ser configurado por **Strings**.

Sincronização de Relógio

A **Sincronização de Relógio de um Escravo** (ASDU **C_CS_NA_1** do protocolo 103) pode ser efetuada de forma automática ou manual.

Sincronização Automática

Para configurar a sincronização automática de relógio, selecione a aba **Master** na aba **IEC870** da janela de propriedades deste Driver e habilite a opção Auto-run clock synchronization. Desta forma, este Driver sincroniza o relógio de um **Escravo** sempre que a comunicação é estabelecida. Adicionalmente pode-se habilitar a repetição da sincronização a intervalos fixos selecionando a opção Repeat every (sec) e preenchendo o valor em segundos para esta repetição.

Sincronização Manual

Para efetuar uma sincronização manual de relógio, escreva um valor qualquer no Tag de Envio de um Comando de Sincronização de Relógio.

Tag de Envio de um Comando de Sincronização de Relógio

Somente Escrita

N1	Número de uma Estação
N2	2 (dois)
N3	0 (zero)
N4	0 (zero)

Escreva qualquer valor neste Tag para enviar um comando de Sincronização de Relógio para um **Escravo**. O relógio deste **Escravo** é sincronizado com o horário atual do **Mestre**. Se este comando é bem-sucedido, este Tag retorna o valor 1 (um). Se este comando falha por qualquer motivo, este Tag retorna o valor 0 (zero).

NOTAS

- Se a Sincronização de Relógio e a Interrogação Geral estão habilitadas, a Sincronização de Relógio é sempre executada em primeiro lugar.
- Este Tag não permite a configuração por **Strings**.

Interrogação Geral

Uma **Interrogação Geral** é um procedimento utilizado para requisitar que um **Escravo** envie o estado atual de todos os objetos, que normalmente são enviados sem a estampa de tempo.

Para habilitar o envio de comandos de Interrogação Geral para um **Escravo**, selecione a aba **Master** na aba **IEC870** da janela de configuração e configure as opções a seguir no grupo General Interrogation:

- **Auto-run general interrogation:** Selecione esta opção para habilitar o envio de uma Interrogação Geral sempre ao estabelecer a comunicação com um **Escravo**
- **Repeat every (sec):** Selecione esta opção e configure um intervalo, em segundos, para repetir uma Interrogação Geral
- **Wait ACTTERM:** Selecione esta opção para forçar este Driver a aguardar o envio da notificação de término de uma Interrogação Geral (**ACTTERM**). Recomenda-se que esta opção só seja desmarcada em caso de detecção de que um **Escravo** não envia esta notificação, pois o envio é opcional
- **Process data only at GI end:** Selecione esta opção para processar os dados recebidos apenas no final de uma Interrogação Geral

Processar Dados ao Final de uma Interrogação Geral

Um **Escravo** tem a liberdade de enviar dados espontâneos durante uma Interrogação Geral. Entretanto, é preciso sempre garantir que o último valor enviado é correto e o mais atual.

Por exemplo, quando um **Mestre** inicia uma Interrogação Geral, um **Escravo** cria uma fotografia de todos os dados que são enviados e inicia o processo de envio, que pode durar alguns segundos. Durante este processo de envio, se algum dos dados sofre uma alteração de estado, esta alteração pode ser enviada de forma espontânea:

- Caso um **Escravo** já tenha enviado dados durante uma Interrogação Geral, este **Escravo** pode enviar a alteração espontânea sem problemas
- Caso um **Escravo** não tenha ainda enviado dados, este **Escravo** pode:
- Aguardar para enviar a alteração espontânea ao final de uma Interrogação Geral
- Enviar a alteração espontânea imediatamente e depois enviar a mesma informação, já com o valor atualizado, como parte de uma Interrogação Geral

Alguns **Escravos** implementam esta sequência de forma incorreta, enviando alterações espontâneas durante uma Interrogação Geral, mas sem garantir que os dados de interrogação enviados após o valor espontâneo estão atualizados. Por exemplo, no início de uma Interrogação Geral uma Medida Analógica está com o valor 20 mas, durante uma interrogação geral, ou seja, antes que esta Medida pudesse ser enviada, o valor desta Medida mudou para 30 e este valor foi enviado de forma espontânea. Mais adiante nesta Interrogação Geral, este **Escravo** envia o valor 20, que era o valor desta Medida no início da Interrogação Geral, permitindo que este Driver mantenha este último valor (20) como valor atual desta Medida.

Para contornar este problema em um **Escravo**, pode-se habilitar a opção *Process data only at GI end*. Quando esta opção está selecionada, os dados de interrogação, com causa de transmissão igual a 20, recebidos durante uma Interrogação Geral são empilhados por este Driver até o final do processo de interrogação. Neste momento, este Driver exclui dos dados empilhados de quaisquer pontos que tenham recebido valores espontâneos durante a interrogação. Se esta opção está desmarcada, os dados de interrogação recebidos durante este processo são processados imediatamente. Como esta opção não causa problemas colaterais, fora uma demora a mais para processar os dados de interrogação, pode ser mantida sempre habilitada.

NOTA

Os dados espontâneos recebidos durante uma Interrogação Geral são sempre processados imediatamente por este Driver.

Interrogação de Contadores

Um **Contador**, chamado de *Integrated Total* na documentação da IEC, é um acumulador de valores. O funcionamento de contadores pode ser:

- **Incremental**: Um **Escravo** memoriza, ou congela, os contadores de tempos em tempos e transmite estes contadores para um **Mestre**. Os contadores continuam acumulando valores sem serem afetados por esta operação de congelamento. Um **Mestre**, neste caso, é responsável por calcular a diferença entre os valores retornados por um **Escravo** para descobrir o quanto foi incrementado em cada intervalo
- **Com Reinício**: Um **Escravo** memoriza, ou congela, os contadores de tempos em tempos, reinicia o valor destes contadores para 0 (zero) e este valor congelado é então enviado para um **Mestre**

Normalmente o momento de congelamento é controlado por um **Escravo**. Opcionalmente um **Mestre** pode assumir este papel, enviando comandos de Interrogação de Contadores, ou seja, a ASDU 101 em um Driver no modo **Mestre** é um Tag com o parâmetro *N2* igual a 3 (três).

Cada equipamento pode ter diversos grupos de *Integration Totals*, o que permite ter diferentes períodos de integração para cada grupo de contadores. Esta configuração, ou seja, quais contadores pertencem a quais grupos, está totalmente a cargo do equipamento.

Tag de Envio de um Comando de Interrogação de Contadores

Somente Escrita

N1	Número de uma Estação
N2	3 (três)
N3	Grupo de contadores
N4	Operação

Escreva qualquer valor neste Tag para enviar um comando de Interrogação de Contadores para um **Escravo**. Se este comando é bem-sucedido, esta escrita retorna o valor 1 (um). Se este comando falha por qualquer motivo, esta escrita retorna o valor 0 (zero).

O parâmetro *N3* define o número do grupo de contadores a ser afetado. Configure este parâmetro com o valor 0 (zero) para não especificar nenhum grupo.

NOTA

O valor 0 (zero) no parâmetro *N3* pode ter um significado especial dependendo do equipamento.

O parâmetro *N4* define uma operação a ser efetuada. Os valores possíveis são **0**: Somente interrogação, sem congelamento ou reinício, **1**: Congelamento de contadores sem reinício, **2**: Congelamento de contadores com reinício ou **3**: Reinício de contadores.

Em resposta a um comando de Interrogação, um **Escravo** envia os dados memorizados dos contadores do grupo especificado no comando. As ASDUs dos dados dos contadores são as seguintes:

- **M_IT_NA_1** (ASDU 15, totais integrados)
- **M_IT_TA_1** (ASDU 16, totais integrados com estampa de tempo curta)
- **M_IT_TB_1** (ASDU 37, totais integrados com estampa de tempo completa)

Os dados retornados por uma interrogação podem ser lidos por um **Mestre** utilizando Tags de Leitura de Eventos da Memória Cache, configurando no parâmetro *N3* o número de qualquer uma das ASDUs (15, 16 ou 37) e configurando o parâmetro *N4* com o endereço de um contador. Ao utilizar a configuração por **Strings**, configure a propriedade **ParamItem** com o valor "ITxxx", em que xxx é o endereço de um contador.

Comando de Seleção ou Execução

Os **Comandos de Seleção ou Execução** permitem a um **Mestre** enviar informações a um **Escravo**. Normalmente um **Escravo** exige que seja enviado primeiro um Comando de Seleção, ou vários comandos de Seleção para objetos diferentes, momento em que diversos intertravamentos são verificados por este **Escravo**.

Se os comandos de Seleção são bem-sucedidos, então um **Mestre** pode enviar os Comandos de Execução quando este comando é efetivado. Um **Mestre** também pode enviar comandos de Cancelamento de Seleção para cancelar comandos de Seleção enviados anteriormente.

Há **Escravos** que permitem também o envio direto de Comandos de Execução sem que haja uma seleção anterior.

Tag de Envio de um Comando de Seleção

Somente Escrita

N1	Número de uma Estação
N2	Tipo de comando. Os valores possíveis são 10 : Normal, 20 : Pulso curto, 30 : Pulso longo ou 40 : Saída persistente
N3	Tipo de objeto, entre 45 e 51 e entre 58 e 64
N4	Endereço de um Comando

Escreva um valor neste Tag para enviar um comando de Seleção para um **Escravo**. Se este comando é bem-sucedido, posteriormente é possível enviar um comando de Execução ou um comando de Cancelamento de Seleção.

O parâmetro *N3* define o tipo de objeto que está sendo comandado. Os tipos de objetos a seguir são enviados sem estampa de tempo, conforme definido no protocolo 101:

- **45**: Comando de Ponto Digital Simples
- **46**: Comando de Ponto Digital Duplo
- **47**: Comando de Regulação de Passo
- **48**: Comando de Setpoint (Medida Analógica Normalizada)
- **49**: Comando de Setpoint (Medida Analógica com Escala)
- **50**: Comando de Setpoint (Medida Analógica em Ponto Flutuante)
- **51**: Comando de Bitstring

Os tipos de objetos a seguir são enviados com estampa de tempo completa, conforme definido no protocolo 104:

- **58**: Comando de Ponto Digital Simples com estampa de tempo
- **59**: Comando de Ponto Digital Duplo com estampa de tempo
- **60**: Comando de Regulação de Passo com estampa de tempo
- **61**: Comando de *set-point* (Medida Analógica Normalizada com estampa de tempo)
- **62**: Comando de *set-point* (Medida Analógica com Escala com estampa de tempo)
- **63**: Comando de *set-point* (Medida Analógica em Ponto Flutuante com estampa de tempo)
- **64**: Comando de Bitstring com estampa de tempo

O valor enviado neste comando deve estar de acordo com o tipo de objeto indicado no parâmetro *N3*. Pode-se enviar, no protocolo 101, os comandos definidos no protocolo 104 e vice-versa, desde que o **Escravo** suporte estes comandos. Se este comando foi enviado com sucesso, esta escrita retorna o valor 1 (um) e, se houve qualquer falha no envio, esta escrita retorna o valor 0 (zero).

NOTAS

- Os comandos **51** e **64** (**Bitstring**) não suportam o envio de qualificadores de pulso curto, pulso longo e saída persistente. Nestes casos, estes comandos são enviados para um **Escravo** sem um qualificador. Se este **Escravo** é outro Driver, este recebe o comando com o qualificador normal.
- Os comandos **51** e **64** (**Bitstring**) não possuem informação de **Select** ou **Execute**. Estes comandos são aceitos por este Driver, mas o comando enviado para um **Escravo** não indica se é **Select** ou **Execute**. Se este Escravo é outro Driver, este recebe o comando sempre como **Execute**.
- Para a configuração deste Tag usando **Strings**, consulte o tópico Configuração de Comandos por String.

Tag de Envio de um Comando de Execução

Somente Escrita

N1	Número de uma Estação
N2	Tipo de comando. Os valores possíveis são 11 : Normal, 21 : Pulso curto, 31 : Pulso longo ou 41 : Saída persistente
N3	Tipo de objeto, entre 45 e 51 e entre 58 e 64
N4	Endereço de um Comando

Escreva um valor neste Tag para enviar um comando de Execução para um **Escravo**. Se este comando é bem-sucedido, esta escrita retorna o valor 1 (um) e, se este comando falha, retorna o valor 0 (zero). Muitos equipamentos exigem que um comando de Execução seja precedido de um comando de Seleção contendo o mesmo objeto e valor.

O parâmetro *N3* define o tipo de objeto que está sendo comandado. Os tipos de objetos a seguir são enviados sem estampa de tempo, conforme definido no protocolo 101:

- **45**: Comando de Ponto Digital Simples
- **46**: Comando de Ponto Digital Duplo
- **47**: Comando de Regulação de Passo
- **48**: Comando de Setpoint (Medida Analógica Normalizada)
- **49**: Comando de Setpoint (Medida Analógica com Escala)
- **50**: Comando de Setpoint (Medida Analógica em Ponto Flutuante)
- **51**: Comando de Bitstring

Os tipos de objetos a seguir são enviados com estampa de tempo completa, conforme definido no protocolo 104:

- **58**: Comando de Ponto Digital Simples com estampa de tempo
- **59**: Comando de Ponto Digital Duplo com estampa de tempo
- **60**: Comando de Regulação de Passo com estampa de tempo
- **61**: Comando de Setpoint (Medida Analógica Normalizada com estampa de tempo)
- **62**: Comando de Setpoint (Medida Analógica com Escala com estampa de tempo)
- **63**: Comando de Setpoint (Medida Analógica em Ponto Flutuante com estampa de tempo)
- **64**: Comando de Bitstring com estampa de tempo

O valor enviado neste comando tem que estar de acordo com o tipo de objeto indicado no parâmetro *N3*. Pode-se enviar, no protocolo 101, os comandos definidos no protocolo 104 e vice-versa, desde que o **Escravo** suporte estes comandos.

NOTA

Para a configuração deste Tag usando **Strings**, consulte o tópico Configuração de Comandos por String.

Tag de Envio de um Comando de Cancelamento de Seleção

Somente Escrita

N1	Número de uma Estação
N2	Tipo de comando. Os valores possíveis são 12 : Normal, 22 : Pulso curto, 32 : Pulso longo ou 42 : Saída persistente
N3	Tipo de objeto, entre 45 e 51 e entre 58 e 64
N4	Endereço de um Comando

Escreva um valor neste Tag para enviar um comando de Cancelamento de uma Seleção enviada anteriormente para um **Escravo**. Se este comando é bem-sucedido, esta escrita retorna o valor 1 (um) e, se este comando falha por qualquer motivo, esta escrita retorna o valor 0 (zero).

O parâmetro **N3** define o tipo de objeto cujo comando de Seleção está sendo cancelado. Os tipos de objetos a seguir são enviados sem estampa de tempo, conforme definido no protocolo 101:

- **45**: Comando de Ponto Digital Simples
- **46**: Comando de Ponto Digital Duplo
- **47**: Comando de Regulação de Passo
- **48**: Comando de Setpoint (Medida Analógica Normalizada)
- **49**: Comando de Setpoint (Medida Analógica com Escala)
- **50**: Comando de Setpoint (Medida Analógica em Ponto Flutuante)
- **51**: Comando de Bitstring

Os tipos de objetos a seguir são enviados com estampa de tempo completa, conforme definido no protocolo 104:

- **58**: Comando de Ponto Digital Simples com estampa de tempo
- **59**: Comando de Ponto Digital Duplo com estampa de tempo
- **60**: Comando de Regulação de Passo com estampa de tempo
- **61**: Comando de Setpoint (Medida Analógica Normalizada com estampa de tempo)
- **62**: Comando de Setpoint (Medida Analógica com Escala com estampa de tempo)
- **63**: Comando de Setpoint (Medida Analógica em Ponto Flutuante com estampa de tempo)
- **64**: Comando de Bitstring com estampa de tempo

O valor enviado neste comando tem que estar de acordo com o tipo de objeto indicado no parâmetro *N3*. Pode-se enviar, no protocolo 101, os comandos definidos no protocolo 104 e vice-versa, desde que o **Escravo** suporte estes comandos.

NOTA

Para a configuração deste Tag usando **Strings**, consulte o tópico Configuração de Comandos por String.

Comando de Reinício de Processo

Um **Mestre** pode solicitar o reinício do processo de um **Escravo** enviando um Comando de Reinício de Processo.

Tag de Envio de um Comando de Reinício de Processo

Somente Escrita

N1	Número de uma Estação
N2	1105
N3	0 (zero)
N4	0 (zero)

O valor escrito neste Tag é um qualificador de Comando de Reinício de Processo, conforme os valores a seguir:

- **0:** Não usado
- **1:** Reinício geral de um processo
- **2:** Remove as informações com estampa de tempo pendentes no *buffer* de eventos
- **3 a 127:** Reservado para futuras definições da norma IEC
- **128 a 255:** Reservado para uso específico de cada fabricante

Se este comando é bem-sucedido, esta escrita retorna o valor 1 (um) e, se este comando falha por qualquer motivo, esta escrita retorna o valor 0 (zero).

Comando de Teste

Um **Comando de Teste** permite validar o funcionamento da camada de aplicação deste protocolo.

Um Driver no modo **Mestre** pode ser configurado para enviar um comando de Teste de forma automática, sempre que o nível de aplicação permanecer um determinado tempo sem ser utilizado.

Para habilitar o envio de um comando de Teste, selecione a aba **Master** na aba **IEC870** da janela de configurações, habilite a opção Send when App Layer is idle (sec) no grupo **Test Command** e preencha um valor em segundos.

No protocolo 101, a ASDU enviada é **C_TS_NA_1** (104). Se um comando de Teste falha, a conexão é fechada e reaberta.

Escravo 101

Esta seção cobre o funcionamento deste Driver utilizando os protocolos **IEC 870-5-101 Escravo Balanceado** e **IEC 870-5-101 Escravo Não Balanceado**. Nestes protocolos, este Driver assume o papel de **Escravo** na comunicação, suportando as funções a seguir:

- Envio de Dados de Monitoramento
- Sincronização de Relógio
- Interrogação Geral
- Comando de Seleção ou Execução
- Comando de Reinício de Processo
- Comando de Teste

Envio de Dados de Monitoramento

Um **Envio de Dados de Monitoramento**, como por exemplo dados digitais ou analógicos, para um **Mestre** pode ocorrer por Interrogação Geral, por envio espontâneo com prioridade alta ou por envio espontâneo com prioridade baixa.

Uma Interrogação Geral é um comando iniciado por um **Mestre** de tempos em tempos. Em resposta a um comando de Interrogação Geral, um **Escravo** envia o valor atual de um conjunto de pontos pré-configurados.

O envio espontâneo serve para que um **Escravo** informe a um **Mestre** a respeito das alterações de valor dos objetos, sem que este **Mestre** envie um comando para receber estas alterações. No protocolo **101 Balanceado** um **Escravo** pode enviar dados diretamente a um **Mestre**. No protocolo **101 Não Balanceado** é necessário aguardar por um *polling* de um **Mestre**, que pergunta para este **Escravo** se há dados de Classe 2 (dois) ou de Classe 1 (um) a serem enviados. Se há dados da Classe, ou prioridade, indicada, este **Escravo** pode enviá-los diretamente na resposta, caso contrário responde com uma mensagem de **NACK**.

Uma aplicação é responsável por alimentar os valores deste Driver no modo **Escravo** através de escritas em Tags. É comum que durante a inicialização de uma aplicação sejam realizadas várias escritas neste Driver no modo **Escravo** para alimentar o estado inicial de todos os Tags. Durante este período pode ser necessário desligar o nível de enlace deste Driver, de forma a evitar respostas incorretas ou incompletas para um **Mestre**.

Tag de Escrita de Eventos na Memória Cache

Leitura ou Escrita

N1	Número de uma Estação
N2	4 (quatro), 5 (cinco), 6 (seis) ou 7 (sete)
N3	ASDU
N4	Endereço de um objeto

Valores escritos neste Tag são armazenados na memória *cache* interna deste Driver e reportados para um **Mestre** da seguinte forma:

- **Quando o parâmetro N2 é igual a 4 (quatro) ou 7 (sete):** Os valores escritos são reportados apenas em uma Interrogação Geral
- **Quando o parâmetro N2 é igual a 5 (cinco):** Os valores escritos são enviados imediatamente para um **Mestre** através de um envio espontâneo de Classe 2, ou prioridade baixa, e também podem ser reportados em uma Interrogação Geral
- **Quando o parâmetro N2 é igual a 6 (seis):** Os valores escritos são enviados imediatamente para um **Mestre** através de um envio espontâneo de Classe 1, ou prioridade alta, e também podem ser reportados em uma Interrogação Geral

Depois que um evento é retornado, este é removido da memória *cache*. Se a memória *cache* está vazia, nenhum valor é retornado para este Tag, ou seja, a leitura não afeta o valor armazenado neste Tag. O evento **OnRead** deste Tag é ativado sempre que um valor novo é recebido e pode ser usado para armazenar os eventos em um arquivo histórico, por exemplo. A leitura e a escrita deste Tag não causam nenhum tipo de comunicação e este Tag não suporta leitura ou escrita em bloco.

No envio espontâneo de dados de Classe 1 (um) ou 2 (dois), a ASDU utilizada para envio destes dados é a especificada no parâmetro *N3*. Por exemplo, se o parâmetro *N3* é igual a 4 (quatro), uma ASDU **M_DP_TA_1** (ponto duplo com estampa de tempo de 24 bits) é enviada para um **Escravo**.

A ASDU utilizada para envio dos dados em uma Interrogação Geral pode ser diferente da configurada no parâmetro *N3* se a opção Remove timestamp from GI data está habilitada. Neste caso, dados carregados neste Driver com o parâmetro *N3* igual a 4 (quatro) podem ser enviado em uma Interrogação Geral com a ASDU **M_DP_NA_1** (ponto duplo sem estampa de tempo).

Os valores possíveis para o parâmetro *N3* são os seguintes:

- **Definidas no protocolo 101:** **M_SP_NA_1** (1), **M_SP_TA_1** (2), **M_DP_NA_1** (3), **M_DP_TA_1** (4), **M_ST_NA_1** (5), **M_ST_TA_1** (6), **M_BO_NA_1** (7), **M_BO_TA_1** (8), **M_ME_NA_1** (9), **M_ME_TA_1** (10), **M_ME_NB_1** (11), **M_ME_TB_1** (12), **M_ME_NC_1** (13), **M_ME_TC_1** (14), **M_IT_NA_1** (15), **M_IT_TA_1** (16), **M_PS_NA_1** (20), **M_ME_ND_1** (21) e **M_EI_NA_1** (70)
- **Definidas no protocolo 104:** **M_SP_TB_1** (30), **M_DP_TB_1** (31), **M_ST_TB_1** (32), **M_BO_TB_1** (33), **M_ME_TD_1** (34), **M_ME_TE_1** (35), **M_ME_TF_1** (36) e **M_IT_TB_1** (37)

NOTA

Para mais informações sobre como configurar este Tag por **Strings**, consulte o tópico Configuração de Tags por String.

Iniciando este Driver com o Nível de Enlace Desabilitado

Os valores iniciais servidos por este Driver no modo **Escravo** precisam ser escritos por uma aplicação durante a inicialização. Para evitar que um **Mestre** conecte-se a este Driver antes que o valor inicial de todos os Tags tenha sido carregado, inicie este Driver com o nível de enlace desabilitado, conforme os passos a seguir:

1. Na janela de propriedades deste Driver, selecione a aba Properties e, no grupo **Link Layer**, configure a opção **Link Layer Starts** com o valor **Disabled**.
2. Ao iniciar a aplicação, depois que o valor de todos os Tags é carregado neste Driver, através de escritas, escreva o valor 1 (um) no Tag de Habilitação do Nível de Enlace.

Tag de Habilitação do Nível de Enlace

Leitura ou Escrita

N1	0 (zero)
N2	996
N3	0 (zero)
N4	0 (zero)

Na escrita este Tag permite habilitar (valor um) ou desabilitar (valor zero) o nível de enlace deste Driver. Com este nível de enlace desabilitado, este Driver aceita conexões de um **Mestre** mas fecha a conexão assim que recebe qualquer mensagem.

Na leitura este Tag retorna o valor 1 (um) se o nível de enlace está habilitado ou o valor 0 (zero) se o nível de enlace está desabilitado.

NOTA

Este Tag não pode ser configurado por **Strings**.

Envio Cíclico de Dados Analógicos

Este Driver no modo **Escravo** pode ser programado para enviar 2 (dois) conjuntos de dados em intervalos cíclicos, como por exemplo a cada 5 (cinco) segundos.

Para configura o envio de dados cíclicos, selecione a aba Properties na janela de propriedades deste Driver e, no grupo **Slave - Cyclic**, configure as seguintes propriedades:

- **Cyclic Data Transmission (1): Send Period (ms):** Define o intervalo entre envios do conjunto 1 (um) de dados cíclicos, em milissegundos
- **Cyclic Data Transmission (1): Object Range:** Define os dados que são enviados de forma cíclica no conjunto 1 (um). Esta propriedade deve seguir a mesma sintaxe do intervalo de dados de uma Interrogação Geral. Deixe esta propriedade em branco para não enviar o conjunto 1 (um). Por exemplo, o valor "M_ME_TC_1:1-10; M_ME_NA_1:15-20, 24" envia os pontos de 1 (um) a 10 do tipo **M_ME_TC_1** (Medida Analógica em Ponto Flutuante com estampa de tempo de 24 bits) e os pontos de 15 a 20 mais o ponto 24 do tipo **M_ME_NA_1** (Medida Analógica Normalizada sem estampa de tempo)
- **Cyclic Data Transmission (2): Send Period (ms):** Define o intervalo entre envios do conjunto 2 (dois) de dados cíclicos, em milissegundos
- **Cyclic Data Transmission (2): Object Range:** Define os dados que são enviados de forma cíclica no conjunto 2 (dois). Esta propriedade deve seguir a mesma sintaxe do intervalo de dados de uma Interrogação Geral

Os dados cíclicos são enviados com causa de transmissão 1 (um) e prioridade baixa. Se um Tag é configurado para ser enviado de forma cíclica, automaticamente este Driver no modo **Escravo** captura todas as escritas neste Tag, mesmo as escritas que pedem para que este Tag seja enviado imediatamente com prioridade alta. Os dados cíclicos são sempre enviados no intervalo

programado, mesmo quando o valor não é alterado.

Forçamento Local

Uma aplicação, a qualquer momento, pode forçar um valor específico para cada Tag. Independente de outros valores escritos posteriormente em um mesmo Tag, este Driver reporta para um **Mestre** sempre o valor forçado, até o momento em que uma aplicação limpa o forçamento.

Ativação de Forçamento

Para ativar o forçamento local em um Tag, escreva um *array* de 2 (dois) elementos no Tag de Escrita de Eventos na Memória Cache. O primeiro elemento deste *array* deve conter o texto "FORCE" e o segundo elemento deve conter o valor a ser forçado, conforme o exemplo a seguir.

```
'Força o valor 1 (um) no tag001  
tag001.WriteEx Array("FORCE", 1)
```

O valor forçado é enviado para um **Mestre** utilizando o mesmo método dos valores normais de um Tag, ou seja, apenas em uma Interrogação Geral com prioridade alta ou prioridade baixa, com o *flag SB* ligado e com a estampa de tempo atual do computador.

Enquanto um Tag está com o valor forçado, este Tag continua aceitando escritas de outros valores não forçados. Neste caso, este Driver armazena o último valor enviado para este Tag e, quando o forçamento é cancelado, este último valor é enviado para um **Mestre**.

Pode-se sobrescrever um valor forçado com outro valor forçado.

Desativação de Forçamento

Para limpar um valor forçado, escreva um *array* com 2 (dois) elementos contendo o texto "CLEAR" no primeiro elemento. O segundo elemento deste *array* é ignorado e pode ter qualquer valor, conforme o exemplo a seguir.

```
'Limpa o valor forçado no tag001  
tag001.WriteEx Array("CLEAR", 0)
```

Quando um forçamento é cancelado, o último valor não forçado escrito neste Driver é reenviado para um **Mestre**. Se nenhum valor está disponível, então é enviado um valor inválido com a estampa de tempo atual do computador.

Sincronização de Relógio

Este Driver suporta a alteração no relógio do computador de acordo com comandos de sincronização de relógio enviados por um **Mestre** (ASDU 103, **C_CS_NA_1**).

Para configurar a forma como este Driver trata a sincronização de relógio, selecione a aba **Slave** na aba **IEC870** da janela de propriedades deste Driver e configure a opção Clock synchronization command da seguinte forma:

- **disabled**: Este Driver rejeita os comandos de sincronização de relógio recebidos, respondendo estes comandos com uma causa de transmissão 71 (confirmação negativa de ativação)
- **pass to the application**: Este Driver responde para um **Mestre** que este comando teve sucesso, ou seja, envia uma causa de transmissão 7 (sete, confirmação de ativação) e deixa a sincronização de relógio a cargo de uma aplicação

- **handle automatically:** Este Driver ajusta o horário local do computador de acordo com o horário recebido de um **Mestre**. Se este ajusta falha, retorna um erro para um **Mestre** com uma causa de transmissão 71 (confirmação negativa de ativação). Caso contrário, responde com uma causa de transmissão 7 (sete, confirmação de ativação) indicando o sucesso desta operação

Este Driver trata internamente os comandos de aquisição de atraso de transmissão (*Delay Acquisition Command*) enviados por um **Mestre** através da ASDU 106 (**C_CD_NA_1**). O valor retornado por este Driver neste comando é o mesmo enviado por um **Mestre**, isto é, são desconsiderados atrasos internos no processamento deste Driver.

Tag de Tratamento de um Comando de Sincronização de Relógio

Somente Leitura

N1	Número de uma Estação
N2	4 (quatro, leitura de dados da memória <i>cache</i>)
N3	103
N4	0 (zero)

Quando este Driver aceita um comando de sincronização de relógio enviado por um **Mestre**, o novo horário é indicado na leitura deste Tag.

Se este Driver está configurado para passar para uma aplicação os comandos de Sincronização de Relógio, ou seja, a opção **Clock synchronization command** está configurada com o valor *pass to the application*, então o evento **OnRead** deste Tag deve atualizar o relógio do computador.

NOTA

Este Tag não pode ser configurado por **Strings**.

Interrogação Geral

Este Driver no modo **Escravo** responde automaticamente aos comandos de Interrogação Geral enviados por um **Mestre**. O conjunto de dados que é retornado na resposta deste Driver pode ser configurado no grupo General Interrogation da aba **Slave** na aba **IEC870**: da janela de configurações deste Driver conforme as opções a seguir:

- **Return a pre-defined data list:** Selecione esta opção para forçar este Driver a sempre retornar uma lista de dados pré-definida em resposta a uma Interrogação Geral. Se esta opção está desmarcada, este Driver retorna todos os valores armazenados na memória *cache* interna
- **Remove timestamp from GI data:** Se esta opção está selecionada, os dados retornados por uma Interrogação Geral são convertidos para uma ASDU sem estampa de tempo, como por exemplo dados das ASDUs 2 (dois) e 30 são enviados com a ASDU 1 (um). Se esta opção está desmarcada, os dados são retornados utilizando a mesma ASDU com a qual foram carregados neste Driver. Para mais informações, consulte o Tag de Escrita de Eventos na Memória Cache

- **Check:** Verifica se sintaxe da lista de dados está correta

A lista de pontos é composta de várias linhas com a sintaxe a seguir.

```
<tipo>: <listaDeEndereços>
```

Em que:

- **<tipo>:** O nome de uma das seguintes ASDUs:
 - **M_SP_NA_1, M_SP_TA_1, M_SP_TB_1:** Ponto Digital Simples
 - **M_DP_NA_1, M_DP_TA_1, M_DP_TB_1:** Ponto Digital Duplo
 - **M_ST_NA_1, M_ST_TA_1, M_ST_TB_1:** Posição de Passo
 - **M_BO_NA_1, M_BO_TA_1, M_BO_TB_1:** Bitstring
 - **M_ME_NA_1, M_ME_TA_1, M_ME_TD_1, M_ME_NB_1, M_ME_TB_1, M_ME_TE_1, M_ME_NC_1, M_ME_TC_1, M_ME_TF_1, M_ME_ND_1:** Medida Analógica
 - **M_IT_NA_1, M_IT_TA_1, M_IT_TB_1:** Total Integrado
 - **M_PS_NA_1**
- **<listaDeEndereços>:** Uma lista de endereços de objetos, separados por vírgulas. Também permite intervalos de endereços contíguos separados por hífen, como por exemplo "1,2,3,10-15", ou seja, endereços 1 (um), 2 (dois) e 3 (três) e de 10 a 15

Por exemplo, para enviar os pontos simples com endereços de 100 a 119 e os pontos duplos de endereços 200, 205 e de 210 a 219, use a sintaxe a seguir.

```
M_SP_NA_1: 100-119
M_DP_NA_1: 200, 205, 210-219
```

Seleção ou Execução

Ao receber um **Comando de Seleção ou Execução**, este Driver pode responder das seguintes formas:

- **Negar este comando (disabled):** Este Driver responde para um **Mestre** com a causa de transmissão 135 (confirmação de ativação negativa)
- **Notificar a aplicação e aguardar que esta indique a resposta deste comando (pass to application):** Este Driver notifica uma aplicação através de uma leitura do Tag Bloco de Tratamento do Comando de Seleção ou Execução, delegando para esta aplicação a responsabilidade por gerar as respostas para um **Mestre**
- **Tratar este comando automaticamente e notificar a aplicação (handle automatically):** Este Driver responde para um **Mestre** como se este comando tivesse sido executado com sucesso e notifica uma aplicação através do Tag Bloco de Tratamento do Comando de Seleção ou Execução, mas neste caso esta aplicação não precisa gerar as respostas para um **Mestre**

Esta configuração é realizada na opção Select/Execute command da aba **Slave** na aba **IEC870** da janela de configurações deste Driver.

Tag Bloco de Tratamento de um Comando de Seleção ou Execução

Leitura ou Escrita

B1	Número de uma Estação
B2	4 (quatro, leitura de dados da memória <i>cache</i>)
B3	ASDU
B4	Endereço de um objeto

O parâmetro *B3* pode ter os seguintes valores:

- **45 (C_SC_NA_1):** *Single command*
- **46 (C_DC_NA_1):** *Double command*
- **47 (C_RC_NA_1):** *Regulating step command*
- **48 (C_SE_NA_1):** *Setpoint command, normalized value*
- **49 (C_SE_NB_1):** *Setpoint command, scaled value*
- **50 (C_SE_NC_1):** *Setpoint command, short floating point number*
- **51 (C_BO_NA_1):** *Bitstream with 32 bits*
- **58 (C_SC_TA_1):** *Single command with 7-octet timestamp*
- **59 (C_DC_TA_1):** *Double command with 7-octet timestamp*
- **60 (C_RC_TA_1):** *Regulating step command with 7-octet timestamp*
- **61 (C_SE_TA_1):** *Setpoint command, normalized value with 7-octet timestamp*
- **62 (C_SE_TB_1):** *Setpoint command, scaled value with 7-octet timestamp*
- **63 (C_SE_TC_1):** *Setpoint command, short floating point number with 7-octet timestamp*
- **64 (C_BO_TA_1):** *Bitstream with 32 bits with 7-octet timestamp*

Quando este Driver recebe um dos comandos descritos anteriormente, este Tag Bloco recebe um bloco com informações sobre este comando. No evento **OnRead** deste Tag Bloco, a aplicação deve tratar esta informação recebida, alterando os Elementos de Bloco necessários e escrevendo este bloco de volta para este Driver. Desta forma, este Driver pode enviar uma resposta para um **Mestre**. Os Elementos deste Tag Bloco são os seguintes:

- **0:** Valor de um comando, que depende do tipo de comando
- **1:** *Cause Of Transmission* (**6:** *Activate* ou **8:** *Deactivate*)
- **2:** *Select or Execute* (**0:** *Execute* ou **1:** *Select*)
- **3:** *Quality Of Command* (**0:** *Undefined*, **1:** *Short pulse duration*, **2:** *Long duration pulse* ou **3:** *Persistent output*)

Quando uma indicação de comando de ativação é recebida, ou seja, o segundo Elemento igual a 6 (seis), uma aplicação deve responder imediatamente com um dos seguintes valores:

- Um **ACTCON** positivo (*activation confirmation*, Elemento 1 igual a 7) se este comando pode prosseguir
- Um **ACTCON** negativo (Elemento 1 igual a 135, 128 + 7) se este comando não pode prosseguir

Se um **ACTCON** positivo é enviado, esta aplicação deve então verificar o terceiro Elemento (*Select or Execute*) e executar a ação apropriada. Se um comando **Execute** é requisitado, esta aplicação deve executar este comando e depois enviar um **ACTTERM** (*activation termination*, Elemento 1 igual a 10) para indicar que este comando foi executado.

O exemplo a seguir mostra um evento **OnRead** para este Tag Bloco no **E3**, **Power** ou **Water**.

```
' CMDIndication é um Tag Bloco com 4 Elementos. Os parâmetros deste Tag são:
' AdviseType      = 0 (AlwaysInAdvise)
' AllowRead       = True
' AllowWrite      = False
' B1              = <endereço da estação>
' B2              = 4
' B3              = 45 a 51, 58 a 64
' B4              = <endereço do ponto de comando>
' EnableDeadband  = False
' Size            = 4

Sub CMDIndication_OnRead()
    Dim CMD, SE, COT, QU
    CMD = Item(1).Value ' Valor do comando (depende do comando)
    COT = Item(2).Value ' COT (Cause of Transmission)
    SE  = Item(3).Value ' 1: Select ou 0: Execute
    QU  = Item(4).Value ' Quality of command

    Select Case COT
        Case 6
            ' ACT(6) deve ser respondido com ACTCON(7)
            WriteEx Array(CMD, 7, SE, QU), TimeStamp

            If SE = 0 Then
                ' TODO: Execute aqui o comando, que provavelmente
                ' é uma escrita em outro Driver

                ' Envia um ACTTERM(10)
                WriteEx Array(CMD, 10, SE, QU), TimeStamp 'ACTTERM
            End If
        Case 8
            ' DEACT(8) deve ser respondido com DEACTCON(9)
            WriteEx Array(CMD, 9, SE, QU), TimeStamp
    End Select
End Sub
```

NOTAS

- As escritas neste Tag Bloco falham se a opção Select/Execute command na aba **Slave** não está configurada com o valor **pass to the application**.
- os comandos **51** e **64 (Bitstring)** não suportam o envio dos qualificadores de pulso curto, pulso longo e saída persistente. Nestes casos, o quarto Elemento deste Tag Bloco (QU) sempre tem o valor 0 (zero), que é o qualificador de um comando normal.
- os comandos **51** e **64 (Bitstring)** não possuem informação de **Select** ou **Execute**. Nestes casos, o terceiro Elemento deste Tag Bloco (SE) recebe sempre o valor 0 (zero, **Execute**).
- Este Tag não permite a configuração por **Strings**.

Reinício de Processo

Este Driver no modo **Escravo** aceita e responde automaticamente com um **ACK** aos comandos de Reinício de Processo enviados por um **Mestre** (ASDU **C_RP_NA_1**) e depois envia uma indicação para que uma aplicação possa, opcionalmente, tratar este comando.

Tag de Tratamento de um Comando de Reinício de Processo

Somente Leitura

N1	Número de uma Estação
N2	4 (quatro, leitura de dados da memória <i>cache</i>)
N3	105
N4	0 (zero)

Quando este Driver aceita um comando de Reinício de Processo enviado por um **Mestre**, o qualificador enviado por este **Mestre** neste comando é retornado em uma leitura deste Tag. Uma aplicação pode, ao receber este valor, forçar por script o reinício deste Driver.

Para uma lista de qualificadores de reinício de processo definidos no protocolo IEC, consulte o Tag de Envio de um Comando de Reinício de Processo.

NOTA

Este Tag não permite a configuração por **Strings**.

Teste

Os **Comandos de Teste** enviados por um **Mestre** são tratados de forma automática por este Driver no modo **Escravo**, que suporta as ASDUs **C_TS_NA_1** (104) e **C_TS_TA_1** (107).

Artigos Técnicos

Esta seção descreve características técnicas diversas do Driver IEC870-5-101.

Modo Escuta

O Driver IEC870-5-101 suporta o funcionamento no modo **Escuta**. Quando este Driver está neste modo não envia nenhum pacote, apenas extrai informações das mensagens que recebe. O modo **Escuta** pode ser habilitado das seguintes formas:

- Na janela de propriedades deste Driver, selecione a opção **Driver starts in listen mode** da aba **General** na aba **IEC870**.
- Na janela de propriedades do Driver, configure a propriedade **Start in Listen Mode** com o valor True (Verdadeiro) no grupo **Master** da aba **Properties**.
- Com este Driver em modo **Offline**, escreva um valor diferente de 0 (zero) na propriedade IEC.ListenMode.
- Com este Driver em execução, escreva o valor 1 (um) no Tag de Controle do Modo de Escuta.

NOTA

Apesar de este Driver suportar o funcionamento em modo **Escuta** com qualquer protocolo, recomenda-se selecionar um dos protocolos do modo **Mestre**.

Tag de Controle do Modo Escuta

Leitura ou Escrita

N1	0 (zero)
N2	997
N3	0 (zero)
N4	0 (zero)

Escreva o valor 0 (zero) neste Tag para desabilitar o modo **Escuta**. Escrever qualquer outro valor diferente de 0 (zero) habilita o modo **Escuta**. Este Tag retorna o valor 1 (um) se este Driver está no modo **Escuta** ou o valor 0 (zero) se este Driver não está no modo **Escuta**.

NOTA

- Este Tag não permite a configuração por **Strings**.
- Este Tag está disponível para todos os tipos de protocolo.

Memória Cache Interna

A **Memória Cache Interna** é uma memória neste Driver que armazena tanto os dados recebidos de um equipamento quanto os dados enviados para um equipamento.

Memória Cache de Dados Recebidos

As ASDUs de dados recebidas de um equipamento são armazenadas na memória *cache* interna, agrupadas por tipo e endereço. Para ler os dados armazenados nesta memória *cache*, declare um Tag de Leitura da Memória Cache e informando o tipo de ASDU e o endereço de um objeto.

Memória Cache de Envio de Dados

Para tratar os dados a serem enviados, este Driver no modo **Escravo** mantém as seguintes áreas de memória:

- **Cache:** Mantém o valor atual de um objeto
- **Fila de eventos de prioridade alta:** Contém uma lista de eventos a serem enviados a um **Mestre** com prioridade alta
- **Fila de eventos de prioridade baixa:** Contém uma lista de eventos a serem enviados a um **Mestre** com prioridade baixa
- **Fila de eventos de respostas de comandos:** Contém uma lista de resultados de comandos a serem enviados a um **Mestre**

Quando a conexão com um **Mestre** é perdida, um **Escravo** limpa as filas de eventos e mantém apenas a memória *cache* com o valor atual de cada objeto.

Este Driver no modo **Escravo** pode ser configurado para manter em memória as filas de eventos de prioridade baixa e alta, mesmo em caso de desconexão. Para habilitar e configurar o tamanho destas filas, configure as opções do grupo **Event buffering while not connected** da aba **Slave** na aba **IEC870** da janela de propriedades deste Driver com os seguintes valores:

- **class 1 buffer (events):** Habilita manter n eventos de prioridade alta, ou de Classe 1 (um), enquanto um **Escravo** está desconectado de um **Mestre**
- **class 2 buffer (events):** Habilita manter n eventos de prioridade baixa, ou de Classe 2 (dois), enquanto um **Escravo** está desconectado de um **Mestre**

Estampas de Tempo

O Driver IEC870-5-101, qualquer que seja o protocolo selecionado, suporta o envio e recebimento de dados com ou sem estampa de tempo nos seguintes formatos:

- **Sem estampa de tempo:** Neste caso, a estampa de tempo é atribuída por este Driver de acordo com a hora atual do sistema
- **Estampa de tempo de 24 bits:** Este formato inclui apenas as informações de hora, minuto, segundo e milissegundo, e a data é preenchida por este Driver usando a data atual do sistema

- **Estampa de tempo de 56 bits:** Este formato é definido no protocolo 104, mas pode ser usado também no protocolo 101, e inclui a estampa de tempo completa com ano, mês, dia, hora, minuto, segundo e milissegundo

Nos dados recebidos sem estampa de tempo ou com estampa de tempo inválida, com o *flag IV* igual a 1 (um), a estampa de tempo é substituída pelo horário local do computador e a qualidade do valor retornado para uma aplicação é ajustada para indicar que esta estampa de tempo é local.

As estampas de tempo recebidas e enviadas por este Driver podem sofrer ajuste de horário de verão, dependendo da configuração deste Driver e do *flag SU* da estampa de tempo.

Horário de Verão

Este Driver pode ser configurado para tratar ou para ignorar o *flag SU* das estampas de tempo. Este *flag*, quando configurado com o valor 1 (um), indica que a estampa de tempo está no horário de verão.

Para configurar o tratamento de horário de verão deste Driver, configure a opção **Timestamp SU FFlag (Summertime adjustment)** no grupo **Timestamp** da aba Properties na janela de configurações deste Driver com os seguintes valores:

- **Ignore (valor padrão):** Ignora o *flag SU* das estampas de tempo recebidas e enviadas e ignora a configuração de horário de verão do computador. São geradas mensagens de aviso no log sempre que este Driver detecta estampas de tempo com o *flag SU* não consistente com o horário de verão do computador e as mensagens de log são as seguintes:
- **Ao receber uma estampa de tempo com incoerência:** "Warning: driver is ignoring mismatching summertime configurations on the PLC (<PlcBias> minutes) and on the server (<LocalBias> minutes). The resulting timestamp may be incorrect!"
- **Ao enviar uma estampa de tempo com incoerência:** "Warning: the driver is configured to ignore the summertime flag, but the system currently is operating on summertime (<LocalBias> minutes). The outgoing timestamp may be incorrect!"
- **Automatic:** Trata o *flag SU* das estampas de tempo recebidas e enviadas e leva em consideração o horário de verão do computador da seguinte forma:
 - Se a estampa de tempo é recebida com o *flag SU* igual a 1 (um) e o computador não está no horário de verão, atrasa a estampa de tempo em 1 (uma) hora
 - Se a estampa de tempo é recebida com o *flag SU* igual a 1 (um) e o computador está no horário de verão, não altera a estampa de tempo
 - Se a estampa de tempo é recebida com o *flag SU* igual a 0 (zero) e o computador não está no horário de verão, não altera a estampa de tempo
 - Se a estampa de tempo é recebida com o *flag SU* igual a 0 (zero) e o computador está no horário de verão, adianta a estampa de tempo em 1 (uma) hora
- Ao enviar uma estampa de tempo, se o computador está no horário de verão, o *flag SU* é igual a 1 (um)
- Ao enviar uma estampa de tempo, se o computador não está no horário de verão, o *flag SU* é igual a 0 (zero)
- Nos casos em que o computador está no horário de verão mas a configuração do horário de verão não é a padrão, ou seja, o *bias* é diferente de 60 minutos:
 - Se a estampa de tempo é recebida com o *flag SU* igual a 1 (um), atrasa a estampa de tempo em 1 (uma) hora, assumindo que no equipamento o *bias* do horário de verão é sempre de 60 minutos, e soma o *bias* do horário de verão do computador
- Ao enviar uma estampa de tempo, atrasa a estampa de tempo de acordo com o *bias* do horário de verão do computador e envia a estampa de tempo com o *flag SU* igual a 0 (zero)
- Nos casos em que a estampa de tempo é ajustada, gera uma mensagem no log informando:

- **Quando uma estampa de tempo recebida é ajustada:** "Timestamp adjusted: From PLC (<PlcTimestamp> [<PlcBias> minutes]) to Driver (<LocalTimestamp> [<LocalBias> minutes])"
- **Quando uma estampa de tempo é ajustada para envio:** "Timestamp adjusted: From Driver (<LocalTimestamp> [<LocalBias> minutes]) marked as *summertime*" ou "Timestamp adjusted: From Driver (<LocalTimestamp> [<LocalBias> minutes]) to PLC (<PlcTimestamp> [<PlcBias> minutes])"

Fuso Horário

O Driver IEC870-5-101 assume que o fuso horário das Estações com as quais se comunica é o mesmo do computador onde este Driver está executando.

As estampas de tempo enviadas por este Driver estão sempre expressas no horário local deste Driver, e não no formato **UTC** (*Coordinated Universal Time*). Este Driver assume que as estampas de tempo recebidas estão todas expressas no horário local das Estações e não no formato **UTC**.

Qualidade

A **Qualidade OPC** é composta de 8 (oito) bits, divididos nos campos **Status**, **Sub-Status** e **Limite**, descritos na tabela a seguir.

Campos da qualidade OPC

STATUS	SUB-STATUS	LIMITE
0: Ruim	0: Não específico, 1: Erro de configuração, 2: Não conectado, 3: Falha no dispositivo, 4: Falha no sensor, 5: Último valor conhecido, 6: Falha de comunicação ou 7: Fora de serviço	0: Livre, 1: Baixo, 2: Alto ou 3: Constante
1: Incerto	0: Não específico, 1: Último valor utilizável, 4: Sensor impreciso, 5: Unidades de engenharia excedidas ou 6: Sub-normal	
3: Bom	0: Não específico, 1: Sobrescrita local, 6: Estampa de tempo local (definida por este Driver) ou 7: Sobrescrita e estampa de tempo local (definida por este Driver)	

Os bits de qualidade definidos no protocolo IEC são mapeados na propriedade **Quality** de um Tag conforme a descrição a seguir. Os bits mais acima na lista têm maior precedência, então se por exemplo os bits **IV** e **NT** estão ligados, retorna o bit **IV**.

- **IV (inválido):** Se este bit está ligado, então a qualidade é igual a 4 (quatro, **Status** igual a **Ruim**, **Sub-Status** igual a **Erro de Configuração** e **Limite** igual a **Livre**)
- **NT (não tópico):** Se este bit está ligado e o bit **IV** não está ligado, então a qualidade é igual a 8 (oito, **Status** igual a **Ruim**, **Sub-Status** igual a **Não Conectado** e **Limite** igual a **Livre**)
- **SB (substituído):** Se este bit está ligado e a qualidade é boa (**Status** igual a **Bom**), então troca o **Sub-Status** para **Sobrescrita Local**, ou seja, soma 4 (quatro) à qualidade
- **BL (bloqueado):** Se este bit está ligado, adiciona 3 (três, **Limite** igual a **Constante**) à qualidade sem modificar o **Status** e o **Sub-Status**

- **OV (overflow)**: Se este bit está ligado e o bit **BL** não está ligado, adiciona 2 (dois, **Limite** igual a **Alto**) à qualidade sem modificar o **Status** e o **Sub-Status**
- **Timestamp IV (a ASDU veio sem estampa de tempo, ou se a estampa de tempo está com o flag IV (inválido) ligado)**: Se a qualidade é boa (**Status** igual a **Bom**), então troca o **Sub-Status** de 0 (zero, **Não Específico**) para 6 (seis, **Estampa de tempo local**) ou de 1 (um, **Sobrescrita Local**) para 7 (sete, **Sobrescrita e estampa de tempo local**)

A tabela a seguir ilustra as combinações de *flags* recebidos por este Driver e a qualidade resultante nos dados lidos.

Combinações de flags recebidos por este Driver

IV	NT	SB	BL	OV	TIMESTAMP IV	RESULTADO
1 (um)	x	x	0 (zero)	0 (zero)	x	4: Ruim / Erro de Configuração / Livre (00 0001 00)
			0 (zero)	1 (um)		6: Ruim / Erro de Configuração / Alto (00 0001 10)
			1 (um)	x		7: Ruim / Erro de Configuração / Constante (00 0001 11)
0 (zero)	1 (um)	x	0 (zero)	0 (zero)	x	8: Ruim / Não conectado / Livre (00 0010 00)
			0 (zero)	1 (um)		10: Ruim / Não conectado / Alto (00 0010 10)
			1 (um)	x		11: Ruim / Não conectado / Constante (00 0010 11)
0 (zero)	0 (zero)	0 (zero)	0 (zero)	0 (zero)	0 (zero)	192: Boa / Não Específico / Livre (11 0000 00)
			0 (zero)	1 (um)		194: Boa / Não Específico / Alto (11 0000 10)
			1 (um)	x		195: Boa / Não Específico / Constante (11 0000 11)

IV	NT	SB	BL	OV	TIMESTAMP IV	RESULTADO
0 (zero)	0 (zero)	1 (um)	0 (zero)	0 (zero)	0 (zero)	196: Boa / Sobrescrita Local / Livre (11 0001 00)
			0 (zero)	1 (um)		198: Boa / Sobrescrita Local / Alto (11 0001 10)
			1 (um)	x		199: Boa / Sobrescrita Local / Constante (11 0001 11)
0 (zero)	0 (zero)	0 (zero)	0 (zero)	0 (zero)	1 (um)	216: Boa / Timestamp Local / Livre (11 0110 00)
			0 (zero)	1 (um)		218: Boa / Timestamp Local / Alto (11 0110 10)
			1 (um)	x		219: Boa / Timestamp Local / Constante (11 0110 11)
0 (zero)	0 (zero)	1 (um)	0 (zero)	0 (zero)	1 (um)	220: Boa / Sobrescrita e Timestamp Local / Livre (11 0111 00)
			0 (zero)	1 (um)		222: Boa / Sobrescrita e Timestamp Local / Alto (11 0111 10)
			1 (um)	x		223: Boa / Sobrescrita e Timestamp Local / Constante (11 0111 11)

De forma inversa, quando uma aplicação efetua escritas neste Driver, a qualidade proveniente de um Tag é transformada nos bits correspondentes do protocolo de acordo com a tabela anterior, apenas substituindo o valor x desta tabela pelo valor 0 (zero). Por exemplo, a qualidade 4 (quatro) resulta em **IV** igual a 1 (um), **NT** igual a 0 (zero), **SB** igual a 0 (zero), **BL** igual a 0 (zero), **OV** igual a 0 (zero) e **TimestampIV** igual a 0 (zero). As seguintes exceções devem ser observadas na qualidade das escritas:

- Qualquer qualidade **Boa** não listada na tabela anterior é mapeada com todos os *flags* em 0 (zero, com exceção dos *flags* **BL** e **OV**, que sempre podem ser extraídos de uma qualidade)
- Qualquer qualidade **Ruim** ou **Incerta** não listada na tabela anterior é mapeada com o *flag* **IV** igual a 1 (um) e os demais *flags* em 0 (zero, também com exceção dos *flags* **BL** e **OV**)
- O envio de estampas de tempo inválidas por este Driver não é suportado, a estampa de tempo enviada é sempre válida, ou seja, o *flag* **IV** é igual a 0 (zero). Desta forma, os Sub-Status **Timestamp Local** (qualidades 216, 218 e 219) e **Sobrescrita e Timestamp Local** (qualidades 220, 222 e 223) são remapeados para o Sub-Status **Não Específico** (qualidades 192, 194 e 195)

Configuração Offline

Nas situações em que é necessário configurar este Driver em tempo de execução, este Driver pode ser colocado em modo **Offline** e os parâmetros alterados das seguintes formas:

- **Individualmente:** Declare um Tag com a propriedade **ParamItem** contendo o nome da propriedade a alterar, sem aspas, e escreva o novo valor da propriedade neste Tag
- **Em conjunto:** Utilize o Tag **Set Configuration Parameters** da biblioteca **IOKit**, ou seja, o parâmetro *B1* igual a -1 (menos um), o parâmetro *B2* igual a 0 (zero), o parâmetro *B3* igual a 0 (zero) e o parâmetro *B4* igual a 3 (três), e escreva neste Tag Bloco um *array* em que cada 2 (dois) Elementos contêm o nome de uma propriedade e o respectivo valor

Causas de Transmissão

A **Causa de Transmissão** é uma informação que faz parte de uma ASDU enviada ou recebida no protocolo IEC 870 e indica o motivo pelo qual aquela ASDU está sendo enviada ou requisitada. Os valores mais comuns de causa de transmissão no protocolo 101 estão listados na tabela a seguir.

Valores de causa de transmissão no protocolo 101

SIGNIFICADO	CAUSA DE TRANSMISSÃO	CAUSA DE TRANSMISSÃO NEGATIVA (+64)	CAUSA DE TRANSMISSÃO DE TESTE (+128)
Envio cíclico (cyc)	1 (um)	65	129
Envio em background (back)	2 (dois)	66	130
Envio espontâneo (spont)	3 (três)	67	131
Inicialização (init)	4 (quatro)	68	132
Requisitado (req)	5 (cinco)	69	133
Ativação (act)	6 (seis)	70	134
Confirmação de ativação (actcon)	7 (sete)	71	135
Desativação (deact)	8 (oito)	72	136
Confirmação de desativação (deactcon)	9 (nove)	73	137
Término de ativação (actterm)	10	74	138

SIGNIFICADO	CAUSA DE TRANSMISSÃO	CAUSA DE TRANSMISSÃO NEGATIVA (+64)	CAUSA DE TRANSMISSÃO DE TESTE (+128)
Interrogado por interrogação geral (genint)	20	84	148
Interrogado por interrogação do grupo 1 a 6 (inro1 a inro16)	21 a 36	85 a 100	149 a 164
Requisitado por requisição geral de contadores (reqcogen)	37	101	165
Requisitado por requisição do grupo 1 a 4 de contadores (reqco1 a reqco4)	38 a 41	102 a 105	166 a 169
Identificação de tipo (ASDU) desconhecida (unktype)	44	108	172
Causa de transmissão desconhecida (unkcot)	45	109	173
Endereço comum de ASDU desconhecido (unkcommonaddr)	46	110	174
Endereço de objeto de informação desconhecido (unkobjaddr)	47	111	175

Referência de Propriedades deste Driver

A tabela a seguir lista todas as propriedades do Driver IEC870-5-101. Estas propriedades estão disponíveis para configuração na aba Properties. Também é possível configurar estas propriedades em tempo de execução. Para mais informações, consulte o tópico Configuração Offline.

Propriedades do Driver IEC870-5-101

PROPRIEDADE	VALOR PADRÃO	COMENTÁRIOS
Protocol	101	Configura o protocolo a ser utilizado por este Driver. Gravada como "IEC.Protocol". Os valores possíveis são 101 : IEC 870-5-101 Mestre Não Balanceado, 1002 : IEC 870-5-101

PROPRIEDADE	VALOR PADRÃO	COMENTÁRIOS
		Escravo Não Balanceado, 1003 : IEC 870-5-101 Mestre Balanceado ou 1004 : IEC 870-5-101 Escravo Balanceado
Link Layer - Default Slave Address	0 (zero)	Determina o identificador de Estação a ser utilizado quando o parâmetro <i>N1</i> ou <i>B1</i> é igual a 0 (zero). Gravada como "IEC.DefaultSlaveAddress"
Link Layer - Force Common Address	0 (zero)	Determina se as Estações criadas dinamicamente têm um <i>Common Address of ASDU</i> igual ao endereço físico ou se utilizam o valor configurado na propriedade IEC.CommonAddress. Gravada como "IEC.EnableCommonAddress". Os valores possíveis são 0 : Utiliza o mesmo valor do endereço físico da Estação ou 1 ou qualquer outro valor diferente de 0 : Utiliza o valor configurado na propriedade IEC.CommonAddress
Link Layer - Common Address	0 (zero)	Valor do <i>Common Address of ASDU</i> de Estações criadas dinamicamente. Gravada como "IEC.CommonAddress"
Link Layer - Originator Address for outgoing ASDUs	0 (zero)	Valor a ser colocado na opção Originator Address em todas as ASDUs enviadas por este Driver. Este valor só é enviado se o número de octetos da causa de transmissão é igual a 2 (dois). Gravada como "IEC.OriginatorAddress"
Link Layer - Slave Polling Rate (ms)	100	[Mestre 101 Não Balanceado] Intervalo de tempo, em milissegundos, entre cada pedido de dados realizado para os Escravos , entre 10 e 60000 milissegundos. Gravada como "IEC.SlavePollTime"
Link Layer - Slave Reactivate Period (sec)	30	Tempo, em segundos entre cada tentativa de reativar Estações que estão em estado de falha, entre 1 (um) e 3600 segundos. Gravada como "IEC.SlaveReactivate"
Link Layer - Link Layer Retries	0 (zero)	Define o número de retentativas no nível de enlace, quando ocorre erro na recepção de uma resposta, entre 0 (zero) e 10. Gravada como "IEC.LinkRetries"
Link Layer - DIR (Physical Transmission Direction)	0 (zero)	[Mestre ou Escravo 101 Balanceado e Mestre 102] Em uma comunicação balanceada, em que ambos os lados podem enviar mensagens simultaneamente, permite diferenciar entre as mensagens enviadas de um lado e de

PROPRIEDADE	VALOR PADRÃO	COMENTÁRIOS
		outro, em que um lado é o A e o outro lado é o B na comunicação. Gravada como "IEC.InvertDIR". Os valores possíveis são 0 : Este Driver quando Mestre é a Estação A e quando Escravo é a Estação B ou 1 ou qualquer outro valor diferente de 0 : Este Driver quando Mestre é a Estação B e quando Escravo é a Estação A
Link Layer - Link Layer Starts	0 (zero)	Determina se o nível de enlace deste Driver inicia habilitado ou desabilitado. Gravada como "IEC.LinkLayerStartsDisabled". Os valores possíveis são 0 : Nível de enlace inicia habilitado ou 1 ou qualquer outro valor diferente de 0 : Nível de enlace inicia desabilitado
Octets - Octets (Cause Of Transmission)	1 (um)	Número de octetos (bytes) do campo Cause of Transmission do pacote FT 1.2. Gravada como "IEC.Octets.COT". Os valores possíveis são 1 : Um octeto ou 2 : Dois octetos, um para Cause of Transmission e um para Originator Address
Octets - Octets (Information Object Address)	2 (dois)	Número de octetos (bytes) do campo Information Object Address do pacote FT 1.2. Gravada como "IEC.Octets.INFOOBJ". Os valores possíveis são 1 : Um octeto (endereços de objeto entre 0 e 255), 2 : Dois octetos (endereços de objeto entre 0 e 65535) ou 3 : Três octetos (endereços de objeto entre 0 e 16777215)
Octets - Octets (Station Address)	1 (um)	Número de octetos (bytes) do campo Slave Address no pacote FT 1.2. Gravada como "IEC.Octets.SlaveAddress". Os valores possíveis são 1 : Um octeto (endereços entre 0 e 255) ou 2 : Dois octetos (endereços entre 0 e 65535)
Octets - Octets (Common Address of ASDU)	-1 (menos um)	Número de octetos (bytes) do campo Common Address of ASDU no pacote FT 1.2. Gravada como "IEC.Octets.CommonAddress". Os valores possíveis são -1 : Utiliza a mesma configuração da propriedade IEC.Octets.SlaveAddress. Os valores possíveis são 1 : Um octeto (endereços entre 0 e 255) ou 2 : Dois octetos (endereços entre 0 e 65535)
Octets - Maximum Packet Size (default = 261 bytes)	261	Configura o tamanho máximo de cada pacote, em bytes. O valor padrão desta propriedade é 261, que é o tamanho

PROPRIEDADE	VALOR PADRÃO	COMENTÁRIOS
		máximo de um pacote FT1.2 utilizado pelo protocolos 101. Gravada como "IEC.MaxPacketSize"
Master - Enable Callbacks	0 (zero)	Determina se os valores armazenados na memória <i>cache</i> deste Driver são retornados para uma aplicação via <i>polling</i> ou via <i>callbacks</i> . O Elipse SCADA e versões antigas do E3 só suportam <i>polling</i> . Gravada como "IEC.EnableCallbacks". Os valores possíveis são 0 : Retorna os dados via <i>polling</i> ou 1 ou qualquer outro valor diferente de 0 : Retorna os dados via <i>callbacks</i>
Master - Application Layer Timeout (sec)	4 (quatro)	[Mestre] Define o <i>time-out</i> de comandos de aplicação, em segundos. Este é o tempo que este Driver aguarda pela confirmação de término de diversos comandos, como por exemplo Interrogação Geral, Selecionar ou Executar ou Sincronização de Relógio. Gravada como "IEC.AppTimeout"
Master - Wait EI (End Of Initialization)	0 (zero)	[Mestre] Indica se este Driver deve aguardar o recebimento de uma ASDU 70 (M_EI_NA_1 , Fim de inicialização) para considerar um Escravo como conectado. Gravada como "IEC.WaitEI". Os valores possíveis são 0 : Não aguarda ou 1 ou qualquer outro valor diferente de 0 : Aguarda. Se a ASDU não é recebida dentro do tempo configurado no <i>time-out</i> de aplicação, o Escravo é desconectado
Master - Don't wait ACTTERM for Execute commands	0 (zero)	[Mestre] Define se este Driver aguarda a sinalização de término dos comandos Execute (<i>ACTivation TErmination</i>). Gravada como "IEC.CommandExecute.DontWaitACTTERM". Os valores possíveis são 0 : Aguarda ou 1 ou qualquer outro valor diferente de 0 : Não aguarda
Master - Start in Listen Mode	0 (zero)	Habilita o modo Escuta. Gravada como "IEC.ListenMode". Os valores possíveis são 0 : Modo Escuta desabilitado ou 1 ou qualquer outro valor diferente de 0 : Modo Escuta habilitado
Master - General Interrogation - Interrogation Auto-run	0 (zero)	[Mestre] Habilita o comando automático de Interrogação Geral. Gravada como "IEC.Interrogate.AutoRun". Os valores possíveis são 0 : Desabilita ou 1 ou qualquer outro valor diferente de 0 : Habilita

PROPRIEDADE	VALOR PADRÃO	COMENTÁRIOS
Master - General Interrogation - Interrogation Repeat	0 (zero)	[Mestre] Habilita a repetição de uma Interrogação Geral. Gravada como "IEC.Interrogate.Repeat". Os valores possíveis são 0 : Desabilita e efetua uma Interrogação Geral apenas uma vez quando se conectar a um Escravo ou 1 ou qualquer outro valor diferente de 0 : Habilita a repetição de uma Interrogação Geral
Master - General Interrogation - Interrogation Repeat Period (sec)	60	[Mestre] Define o intervalo de tempo, em segundos, entre cada repetição de uma Interrogação Geral. Gravada como "IEC.Interrogate.PollTime"
Master - General Interrogation - Interrogation Wait ACTTERM	1 (um)	[Mestre] Determina se este Driver aguarda ou não pelo recebimento da notificação de término de uma Interrogação Geral. Gravada como "IEC.Interrogate.WaitACTTERM". Os valores possíveis são 0 : Não aguarda pela notificação de término (<i>ACTivation TERMination</i>) ou 1 ou qualquer outro valor diferente de 0 : Aguarda o recebimento da notificação de término de uma Interrogação Geral
Master - General Interrogation - Interrogation Process At End	0 (zero)	[Mestre] Define o momento em que os dados de uma Interrogação Geral são processados. Gravada como "IEC.Interrogate.ProcessAtEnd". Os valores possíveis são 0 : Processa durante uma Interrogação Geral ou 1 ou qualquer outro valor diferente de 0 : Processa apenas no final de uma Interrogação Geral
Master - General Interrogation - Disconnect if General Interrogation returns empty results	0 (zero)	[Mestre] Determina se este Driver aceita ou não uma resposta vazia de uma Interrogação Geral. Gravada como "IEC.Interrogate.DisconnectIfEmpty". Os valores possíveis são 0 : Aceita uma resposta de Interrogação Geral vazia ou 1 ou qualquer outro valor diferente de 0 : Se a resposta de uma Interrogação Geral é vazia força uma desconexão, permitindo que este Driver conecte a outro equipamento em uma conexão de <i>backup</i>
Master - Test Command - Test Command Enabled	0 (zero)	[Mestre] Habilita o uso do envio automático de um Comando de Teste. Gravada como "IEC.TestCommand.Enabled". Os valores possíveis são 0 : Desabilita ou 1 ou qualquer outro valor diferente de 0 : Habilita

PROPRIEDADE	VALOR PADRÃO	COMENTÁRIOS
Master - Test Command - Test Command Idle Time (sec)	15	[Mestre] Define o período de tempo, em segundos, que este Driver tem que permanecer sem enviar comandos do nível de aplicação para que ative o envio automático de um Comando de Teste. Gravada como "IEC.TestCommand.IdleTimeSec"
Master - Clock Sync - Clock Sync Auto-run	0 (zero)	[Mestre] Habilita o envio de um Comando de Sincronização de Relógio. Gravada como "IEC.ClockSync.AutoRun". Os valores possíveis são 0 : Desabilita ou 1 ou qualquer outro valor diferente de 0 : Habilita
Master - Clock Sync - Clock Sync Repeat	0 (zero)	[Mestre] Habilita a repetição do envio de um Comando de Sincronização de Relógio. Gravada como "IEC.ClockSync.Repeat". Os valores possíveis são 0 : Desabilita ou 1 ou qualquer outro valor diferente de 0 : Habilita
Master - Clock Sync - Clock Sync Repeat Period (sec)	60	[Mestre] Define o intervalo, em segundos para repetir o envio de um Comando de Sincronização de Relógio. Gravada como "IEC.ClockSync.PollTime"
Slave - Send EI (End Of Initialization) When Connected	0 (zero)	[Escravo] Determina se um Escravo envia ou não uma ASDU M_EI_NA_1 (70) indicando que uma conexão foi estabelecida com um Mestre . Gravada como "IEC.DontSendEI". Os valores possíveis são 0 : Envia a ASDU M_EI_NA_1 sempre que uma conexão é estabelecida ou 1 ou qualquer outro valor diferente de 0 : Não envia a ASDU M_EI_NA_1
Slave - Ack using CC1	0 (zero)	[Mestre ou Escravo 101 Balanceado] Indica se este Driver envia ACKs utilizando o caractere especial CC1 (0xE5) do protocolo. Gravada como "IEC.EnableCC1Ack". Os valores possíveis são 0 : Não envia o ACK com o caractere CC1 ou 1 ou qualquer outro valor diferente de 0 : Envia o ACK utilizando o caractere CC1 quando possível
Slave - Analog Values Send Interval (ms)	0 (zero)	[Escravo] Intervalo entre o envio cíclico de dados analógicos, em milissegundos. Gravada como "IEC.AnalogSendIntervalMs"

PROPRIEDADE	VALOR PADRÃO	COMENTÁRIOS
Slave - Analog Deadband (% of previous value)	0 (zero)	[Escravo] Banda morta para envio cíclico de dados analógicos, em uma porcentagem de variação do valor anterior. Gravada como "IEC.AnalogDeadband"
Slave - Treat First Write As	0 (zero, <i>New Event</i>)	[Escravo] Determina se a primeira escrita efetuada nos Tags de um Escravo é considerada um novo evento ou o valor inicial de um Tag. Esta propriedade funciona até que a primeira Interrogação Geral seja efetuada. Após a primeira Interrogação Geral, todas as escritas em um Tag Escravo são retornadas para um Mestre como novos eventos. Gravada como "IEC.FirstWritesCurrentValue". Os valores possíveis são 0 (New Event) : Trata a primeira escrita de cada Tag como um evento ou 1 ou qualquer outro valor diferente de 0 (Current Value) : Trata a primeira escrita de cada Tag como o valor inicial, ou seja, envia o valor por Interrogação Geral mas não envia a notificação espontânea
Slave - Enable Callbacks	0 (zero)	Determina se os valores armazenados na memória <i>cache</i> deste Driver são retornados para uma aplicação via <i>polling</i> ou via <i>callbacks</i> . O Elipse SCADA e versões antigas do E3 só suportam <i>polling</i> . Gravada como "IEC.EnableCallbacks". Os valores possíveis são 0 : Retorna os dados via <i>polling</i> ou 1 ou qualquer outro valor diferente de 0 : Retorna os dados via <i>callbacks</i> . NOTA : É importante habilitar as <i>callbacks</i> para que os comandos recebidos por um Escravo sejam tratados por uma aplicação na mesma ordem em que foram recebidos
Slave - Poll for Class1 may respond Class2 data	1 (um)	[Escravo 101 Não Balanceado] Habilita o envio de dados de Classe 2 (dois, prioridade baixa) quando este Driver no modo Escravo recebe um pedido de envio de Classe 1 (um, prioridade alta) e não há dados de Classe 1 (um) a enviar. Gravada como "IEC.Class1MayRespondClass2". Os valores possíveis são 0 : Desabilita ou 1 ou qualquer outro valor diferente de 0 : Habilita
Slave - Poll for Class2 may respond Class1 data	1 (um)	(Escravo 101 Não-Balanceado) Habilita o envio de dados de Classe 1 (um, prioridade alta) quando este Driver no modo Escravo recebe um pedido de envio de Classe 2 (dois, prioridade baixa) e não há dados de Classe 2 (dois) a enviar. Gravada como "IEC.Class2MayRespondClass1". Os valores

PROPRIEDADE	VALOR PADRÃO	COMENTÁRIOS
		possíveis são 0 : Desabilita ou 1 ou qualquer outro valor diferente de 0 : Habilita
Slave - General Interrogation - Interrogation Enable Range	0 (zero)	[Escravo] Habilita o retorno de uma lista fixa de objetos em uma Interrogação Geral. Gravada como "IEC.EnableInterrogationRange". Os valores possíveis são 0 : Retorna todos os objetos da memória <i>cache</i> deste Driver ou 1 ou qualquer outro valor diferente de 0 : Retorna apenas os objetos declarados na lista de uma Interrogação Geral
Slave - General Interrogation - List of data objects returned by General Interrogation	String vazia	[Escravo] Define a lista de objetos que são retornados em uma Interrogação Geral. Gravada como "IEC.InterrogationRange"
Slave - General Interrogation - Remove timestamp from data sent on General Interrogation	0 (zero)	[Escravo] Determina se os dados de uma Interrogação Geral são enviados sem a estampa de tempo. Gravada como "IEC.InterrogationRemoveTimestamp". Os valores possíveis são 0 : Envia os dados utilizando a ASDU original, que pode ou não ter estampa de tempo, ou 1 ou qualquer outro valor diferente de 0 : Remove a estampa de tempo dos dados enviados em uma Interrogação Geral, convertendo-os para uma ASDU sem estampa de tempo
Slave - Command Handling - Handle Mode for Clock Sync commands	0 (zero)	[Escravo] Define o tratamento para Comandos de Sincronização de Relógio recebidos por este Driver. Gravada como "IEC.CommandHandleMode.ClockSync". Os valores possíveis são 0 : Padrão (desabilitado), 1 : Desabilitado, 2 : Passa para a aplicação ou 3 : Trata automaticamente
Slave - Command Handling - Handle Mode for Select/Execute commands	0 (zero)	[Escravo] Define o tratamento para Comandos de Seleção ou Execução recebidos por este Driver. Gravada como "IEC.CommandHandleMode.SelectExecute". Os valores possíveis são 0 : Padrão (passa para a aplicação), 1 : Desabilitado, 2 : Passa para a aplicação ou 3 : Trata automaticamente
Slave - Command Handling - Handle Mode for Test commands	0 (zero)	[Escravo] Define o tratamento para Comandos de Teste recebidos por este Driver. Gravada como "IEC.CommandHandleMode.Test". Os valores possíveis são 0 : Padrão (trata automaticamente), 1 : Desabilitado, 2 : Passa para a aplicação ou 3 : Trata automaticamente

PROPRIEDADE	VALOR PADRÃO	COMENTÁRIOS
Slave - Cyclic - Cyclic Data Transmission (1): Send Period (ms)	10000	[Escravo] Intervalo, em milissegundos, para o envio dos dados cíclicos declarados na propriedade IEC.CyclicRange. Gravada como "IEC.CyclidPeriodMs". NOTA: Repare que o nome de gravação desta variável possui um erro de digitação, <i>Cyclid</i> ao invés de <i>Cyclic</i> , mantido para não quebrar a compatibilidade com versões antigas deste Driver
Slave - Cyclic - Cyclic Data Transmission (1): Object Range	String vazia	[Escravo] Define os dados que devem ser enviados de forma cíclica. Gravada como "IEC.CyclicRange"
Slave - Cyclic - Cyclic Data Transmission (2): Send Period (ms)	5000	[Escravo] Intervalo, em milissegundos, para o envio dos dados cíclicos declarados na propriedade IEC.CyclicRange2. Gravada como "IEC.CyclicPeriodMs2"
Slave - Cyclic - Cyclic Data Transmission (2): Object Range	String vazia	[Escravo] Define um segundo conjunto de dados que devem ser enviados de forma cíclica. Gravada como "IEC.CyclicRange2"
Slave - Offline Buffers - Offline Class1 buffering	0 (zero)	[Escravo] Habilita manter eventos de Classe 1 (um, alta prioridade) em memória quando um Mestre está desconectado. Gravada como "IEC.OfflineClass1.EnableBuffering". Os valores possíveis são 0 : Desabilita ou 1 ou qualquer outro valor diferente de 0 : Habilita
Slave - Offline Buffers - Offline Class1 buffer size	100	[Escravo] Número de eventos de Classe 1 (um, alta prioridade) a manter em memória quando um Mestre está desconectado. Gravada como "IEC.OfflineClass1.BufferSize"
Slave - Offline Buffers - Offline Class2 buffering	0 (zero)	[Escravo] Habilita manter eventos de Classe 2 (dois, baixa prioridade) em memória quando um Mestre está desconectado. Gravada como "IEC.OfflineClass2.EnableBuffering". Os valores possíveis são 0 : Desabilita ou 1 ou qualquer outro valor diferente de 0 : Habilita
Slave - Offline Buffers - Offline Class2 buffer size	100	[Escravo] Número de eventos de Classe 2 (dois, baixa prioridade) a manter em memória quando um Mestre está desconectado. Gravada como "IEC.OfflineClass2.BufferSize"

PROPRIEDADE	VALOR PADRÃO	COMENTÁRIOS
Timestamp - Timestamp SU Flag (Summertime adjustment)	0 (zero)	Determina o tratamento do horário de verão nas estampas de tempo. Gravada como "IEC.TimestampSuFlag". Os valores possíveis são 0 : Ignora o flag SU nas estampas de tempo ou 1 ou qualquer outro valor diferente de 0 : Utiliza o flag SU nas estampas de tempo
Timestamp - Enable Station TimeZone	0 (zero)	Habilita o uso do parâmetro <i>Station TimeZone</i> , que permite indicar um fuso horário para todas as Estações diferente do fuso horário do computador. Gravada como "IEC.TimeZoneEnable"
Timestamp - Station TimeZone (in minutes)	0 (zero)	Define a diferença, em minutos, entre o fuso horário das Estações e o fuso horário de Greenwich. Por exemplo, utilize o valor -180 para o fuso horário de Brasília (GMT-3). Gravada como "IEC.TimeZoneMinutes"
Timestamp - SP Timestamp Format	2 (dois)	[Escravo] Define o formato da estampa de tempo dos dados de Ponto Simples enviados para um Mestre fora de uma Interrogação Geral, apenas para Tags configurados por String . Gravada como "IEC.TimestampFormat.SP". Os valores possíveis são 0 : Sem estampa de tempo (utiliza a ASDU M_SP_NA_1 , 1), 1 : Com estampa de tempo de 24 bits (utiliza a ASDU M_SP_TA_1 , 2) ou 2 : Com estampa de tempo de 56 bits (utiliza a ASDU M_SP_TB_1 , 30)
Timestamp - DP Timestamp Format	2 (dois)	[Escravo] Define o formato da estampa de tempo dos dados de Ponto Duplo enviados para um Mestre fora de uma Interrogação Geral, apenas para Tags configurados por String . Gravada como "IEC.TimestampFormat.DP". Os valores possíveis são 0 : Sem estampa de tempo (utiliza a ASDU M_DP_NA_1 , 3), 1 : Com estampa de tempo de 24 bits (utiliza a ASDU M_DP_TA_1 , 4) ou 2 : Com estampa de tempo de 56 bits (utiliza a ASDU M_DP_TB_1 , 31)

PROPRIEDADE	VALOR PADRÃO	COMENTÁRIOS
Timestamp - BO Timestamp Format	2 (dois)	[Escravo] Define o formato da estampa de tempo dos dados de Bitstring enviados para um Mestre fora de uma Interrogação Geral, apenas para Tags configurados por String . Gravada como "IEC.TimestampFormat.BO". Os valores possíveis são 0 : Sem estampa de tempo (utiliza a ASDU M_BO_NA_1 , 7), 1 : Com estampa de tempo de 24 bits (utiliza a ASDU M_BO_TA_1 , 8) ou 2 : Com estampa de tempo de 56 bits (utiliza a ASDU M_BO_TB_1 , 33)
Timestamp - ST Timestamp Format	2 (dois)	[Escravo] Define o formato da estampa de tempo dos dados de Posição de Passo enviados para um Mestre fora de uma Interrogação Geral, apenas para Tags configurados por String . Gravada como "IEC.TimestampFormat.ST". Os valores possíveis são 0 : Sem estampa de tempo (utiliza a ASDU M_ST_NA_1 , 5), 1 : Com estampa de tempo de 24 bits (utiliza a ASDU M_ST_TA_1 , 6) ou 2 : Com estampa de tempo de 56 bits (utiliza a ASDU M_ST_TB_1 , 32)
Timestamp - MEn Timestamp Format	2 (dois)	[Escravo] Define o formato da estampa de tempo dos dados de Medidas Analógicas Normalizadas enviados para um Mestre fora de uma Interrogação Geral, apenas para Tags configurados por String . Gravada como "IEC.TimestampFormat.MEn". Os valores possíveis são 0 : Sem estampa de tempo (utiliza a ASDU M_ME_NA_1 , 9), 1 : Com estampa de tempo de 24 bits (utiliza a ASDU M_ME_TA_1 , 10) ou 2 : Com estampa de tempo de 56 bits (utiliza a ASDU M_ME_TD_1 , 34)
Timestamp - MEs Timestamp Format	2 (dois)	[Escravo] Define o formato da estampa de tempo dos dados de Medidas Analógicas com Escala enviados para um Mestre fora de uma Interrogação Geral, apenas para Tags configurados por String . Gravada como "IEC.TimestampFormat.MEs". Os valores possíveis são 0 : Sem estampa de tempo (utiliza a ASDU M_ME_NB_1 , 11), 1 : Com estampa de tempo de 24 bits (utiliza a ASDU M_ME_TB_1 , 12) ou 2 : Com estampa de tempo de 56 bits (utiliza a ASDU M_ME_TE_1 , 35)
Timestamp - MEf Timestamp Format	2 (dois)	[Escravo] Define o formato da estampa de tempo dos dados de Medidas Analógicas em Ponto Flutuante enviados para um Mestre fora de uma Interrogação Geral, apenas para Tags configurados por String . Gravada como

PROPRIEDADE	VALOR PADRÃO	COMENTÁRIOS
		"IEC.TimestampFormat.MEF". Os valores possíveis são 0 : Sem estampa de tempo (utiliza a ASDU M_ME_NC_1 , 13), 1 : Com estampa de tempo de 24 bits (utiliza a ASDU M_ME_TC_1 , 14) ou 2 : Com estampa de tempo de 56 bits (utiliza a ASDU M_ME_TF_1 , 36)
Send Priority - SP Send Priority	1 (um)	[Escravo] Define a forma e prioridade de envio para um Mestre dos dados de Ponto Digital Simples, apenas para Tags configurados por String . Gravada como "IEC.SendFlags.SP". Os valores possíveis são 1 : Envia com alta prioridade, 2 : Envia com baixa prioridade ou 3 : Envia apenas em uma Interrogação Geral
Send Priority - DP Send Priority	1 (um)	[Escravo] Define a forma e prioridade de envio para um Mestre dos dados de Ponto Digital Duplo, apenas para Tags configurados por String . Gravada como "IEC.SendFlags.DP". Os valores possíveis são 1 : Envia com alta prioridade, 2 : Envia com baixa prioridade ou 3 : Envia apenas em uma Interrogação Geral
Send Priority - BO Send Priority	2 (dois)	[Escravo] Define a forma e prioridade de envio para um Mestre dos dados de Bitstring, apenas para Tags configurados por String . Gravada como "IEC.SendFlags.BO". Os valores possíveis são 1 : Envia com alta prioridade, 2 : Envia com baixa prioridade ou 3 : Envia apenas em uma Interrogação Geral
Send Priority - ST Send Priority	2 (dois)	[Escravo] Define a forma e prioridade de envio para um Mestre dos dados de Posição de Passo, apenas para Tags configurados por String . Gravada como "IEC.SendFlags.ST". Os valores possíveis são 1 : Envia com alta prioridade, 2 : Envia com baixa prioridade ou 3 : Envia apenas em uma Interrogação Geral
Send Priority - MEn Send Priority	2 (dois)	[Escravo] Define a forma e prioridade de envio para um Mestre das Medidas Analógicas Normalizadas, apenas para Tags configurados por String . Gravada como "IEC.SendFlags.MEn". Os valores possíveis são 1 : Envia com alta prioridade, 2 : Envia com baixa prioridade ou 3 : Envia apenas em uma Interrogação Geral

PROPRIEDADE	VALOR PADRÃO	COMENTÁRIOS
Send Priority - MEs Send Priority	2 (dois)	[Escravo] Define a forma e prioridade de envio para um Mestre das Medidas Analógicas com Escala, apenas para Tags configurados por String . Gravada como "IEC.SendFlags.MEs". Os valores possíveis são 1 : Envia com alta prioridade, 2 : Envia com baixa prioridade ou 3 : Envia apenas em uma Interrogação Geral
Send Priority - MEf Send Priority	2 (dois)	[Escravo] Define a forma e prioridade de envio para um Mestre das Medidas Analógicas em Ponto Flutuante, apenas para Tags configurados por String . Gravada como "IEC.SendFlags.MEf". Os valores possíveis são 1 : Envia com alta prioridade, 2 : Envia com baixa prioridade ou 3 : Envia apenas em uma Interrogação Geral
Boolean Values - Convert DCO to Boolean	0 (zero)	Define o formato dos valores de um Comando de Ponto Digital Duplo. Gravada como "IEC.UseBoolean.DCO". Os valores possíveis são 0 : Usa como um DCO (0 : Indeterminado, 1 : OFF, 2 : ON ou 3 : Indeterminado) ou 1 ou qualquer outro valor diferente de 0 : Usa como um valor Booleano (0 : OFF ou 1 : ON)
Boolean Values - Convert DIQ to Boolean	0 (zero)	Habilita o uso de Pontos Digitais Duplos como valores Booleanos. Gravada como "IEC.UseBoolean.DIQ". Os valores possíveis são 0 : Usa como um DIQ (0 : Indeterminado, 1 : OFF, 2 : ON ou 3 : Indeterminado) ou 1 ou qualquer outro valor diferente de 0 : Usa como um valor Booleano (0 : OFF ou 1 : ON)
Boolean Values - Convert RCO to Boolean	0 (zero)	Define o formato dos valores de um Comando de Regulação de Passo. Gravada como "IEC.UseBoolean.RCO". Os valores possíveis são 0 : Usa como um RCS (0 : Não permitido, 1 : Decrementa, 2 : Incrementa ou 3 : Não permitido) ou 1 ou qualquer outro valor diferente de 0 : Usa como um valor Booleano (0 : Decrementa ou 1 : Incrementa)

PROPRIEDADE	VALOR PADRÃO	COMENTÁRIOS
Boolean Values - Convert SEP to Boolean	0 (zero)	Define o formato dos valores de um Evento Simples de um Equipamento de Proteção. Gravada como "IEC.UseBoolean.SEP". Os valores possíveis são 0 : Usa como um SEP (0 : Indeterminado, 1 : OFF, 2 : ON ou 3 : Indeterminado) ou 1 ou qualquer outro valor diferente de 0 : Usa como um valor Booleano (0 : OFF ou 1 : ON)
Stations - Use Fixed Station List	0 (zero)	Habilita o uso de uma lista pré-definida de Estações. Gravada como "IEC.Stations.UseFixedList". Os valores possíveis são 0 : Desabilita ou 1 ou qualquer outro valor diferente de 0 : Habilita
Stations - Fixed Station Count	0 (zero)	Determina o número de Estações declaradas em uma lista pré-definida de Estações. Gravada como "IEC.Stations.Count"
Stations - Station <n> - Station N1/B1	0 (zero)	Identificador de uma Estação de índice <i>n</i> , baseado em 0 (zero). Gravada como "IEC.Stations(<n>).N1B1"
Stations - Station <n> - Station Name	String vazia	Nome de uma Estação de índice <i>n</i> , baseado em 0 (zero). Gravada como "IEC.Stations(<n>).Name"
Stations - Station <n> - Station Link Address	0 (zero)	Endereço de uma Estação de índice <i>n</i> , baseado em 0 (zero). Gravada como "IEC.Stations(<n>).DataLinkAddress"
Stations - Station <n> - Station Common Address	0 (zero)	<i>Common address</i> de uma Estação de índice <i>n</i> , baseado em 0 (zero). Gravada como "IEC.Stations(<n>).CommonAddress"

Referência de Erros no Log

A tabela a seguir lista alguns possíveis erros que podem aparecer no log deste Driver, com as respectivas dicas de solução.

Erros no log deste Driver

ERRO	DESCRIÇÃO OU SOLUÇÃO
ERROR: Both balanced stations are using the same DIR	[Mestre e Escravo 101 Balanceado] Por padrão, quando este Driver está no modo Mestre , atua como <i>Station A</i> , enviando pacotes com DIR igual a 1 (um) e, quando está no modo

ERRO	DESCRIÇÃO OU SOLUÇÃO
	Escravo , atua como <i>Station B</i> , enviando pacotes com DIR igual a 0 (zero). CAUSA : Se esta mensagem aparece, é sinal que ambas as Estações, este Driver e a outra Estação com a qual este Driver se comunica, estão enviando pacotes utilizando o mesmo valor de DIR . SOLUÇÃO : Considere reconfigurar a propriedade DIR (Physical Transmission Direction) deste Driver
ERROR: Failed to send command 'XXX', application layer is locked	[Mestre] Esta mensagem ocorre quando este Driver tenta enviar um comando em nível de aplicação enquanto outro comando já está em execução. Por exemplo, tentar enviar um comando de Seleção ou Execução enquanto uma Interrogação Geral está em andamento. Este Driver só gera esta mensagem e desiste do envio de um comando depois de aguardar o tempo especificado no <i>time-out</i> de uma aplicação. CAUSA : Camada de Aplicação deste Driver trancada por muito tempo. SOLUÇÃO : Aumentar o <i>time-out</i> de uma aplicação
GeneralInterrogation: received actterm with no GI active, retrying last GI	[Mestre 101] Esta mensagem ocorre quando este Driver recebe uma confirmação de término de uma Interrogação Geral e não há um processo de Interrogação Geral ativo. CAUSAS : Uma Interrogação Geral iniciada por este Driver anteriormente foi interrompida devido a um <i>time-out</i> da camada de aplicação deste Driver, uma Interrogação Geral iniciada por este Driver anteriormente foi interrompida devido a uma queda na conexão com um equipamento ou este Driver está configurado para não aguardar a notificação de término de uma Interrogação Geral e um Escravo está enviando esta notificação de término. SOLUÇÕES : Aumentar o <i>time-out</i> da aplicação deste Driver, Investigar a causa da interrupção na conexão ou configurar este Driver para aguardar o término de uma Interrogação Geral e, possivelmente, também considerar aumentar o <i>time-out</i> da aplicação deste Driver

Documentação das Interfaces de Comunicação

Esta seção contém a documentação das Interfaces de Comunicação referentes ao Driver **IEC870-101**.

Esta documentação é compartilhada e mantida no manual do IOKit:

- Configurações de um Driver
- Configurações Gerais
- Configuração de Estatísticas
- Configuração da Interface Ethernet

- Configuração da Interface Modem
- Configuração da Interface RAS
- Configuração da Interface Serial

Histórico de Revisões do Driver

VERSÃO	DATA	AUTOR	COMENTÁRIOS
4.0.4	10/08/2026	M. Ludwig	• Driver atualizado para a biblioteca IOKit versão 3 e Visual Studio 2022 (<i>Case 37985</i>).
4.0.3	10/12/2024	F. Englert	• Implementado um Tag com o parâmetro <i>N2</i> igual a 993, que retorna o estado do <i>link layer</i> para a Estação indicada no parâmetro <i>N1</i> (<i>Case 36974</i>).
4.0.2	27/03/2024	F. Englert	• Implementada uma resposta ao comando Reset Process (C_RP_NA_1) neste Driver em modo Escravo (<i>Case 35741</i>). • Adicionada a este Driver no modo Escravo a opção Respond Commands With High Priority para aumentar a prioridade de resposta dos comandos em relação ao envio de dados espontâneos de classe 1 (um) ou 2 (dois) (<i>Case 11637</i>).
4.0.1	26/12/2023	F. Englert	• Alterado o nome deste Driver para IEC870-101.dll. Removido o protocolo 104 (Mestre e Escravo), agora tratado por um Driver específico (<i>Case 35356</i>).
3.0.7	25/9/2023	F. Englert	• Corrigido um eventual GPF (<i>General Protection Failure</i>) que poderia ocorrer ao tratar um dado recebido próximo à finalização de um comando de aplicação (<i>Case 34921</i>).
3.0.6	1/2/2021	F. Englert	• Dados recebidos em uma Interrogação Geral sem estampa de tempo não geram mais uma qualidade 216 em Tags que não têm o valor alterado (<i>Case 28623</i>). • Este Driver agora envia corretamente os dados de Totais Integrados (ASDUs 15, 16 e 37) (<i>Case 30398</i>).
3.0.4	27/8/2019	M. Ludwig	• Driver portado para o Visual Studio 2017 (<i>Case 27164</i>).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
3.0.1	20/6/2017	F. Englert	- Removidos os protocolos 102, ZIV e Procome, agora tratados por Drivers específicos (<i>Case 22924</i>). - Alterado o nome deste Driver para IEC870-101-104.dll, que agora utiliza o ID de proteção de Drivers do Power e a licença é intercambiável com outros Drivers deste mesmo grupo (<i>Case 13800</i>).
2.22.80	22/5/2017	F. Englert	Implementada a opção Disconnect if General Interrogation returns empty results (<i>Case 22256</i>).
2.22.77	11/05/2017	F. Englert	- Implementada a opção Enable Callbacks também para os protocolos Escravos 101 Balanceado, 101 Não Balanceado e 104 (<i>Case 22652</i>). - Os comandos analógicos SEn, SEs e SEf de Select ou Execute com qualificadores ShortPulse, LongPulse e PersistentOutput agora são enviados corretamente (<i>Case 22662</i>).
2.22.76	12/04/2017	F. Englert	Este Driver agora suporta a opção IP Filter, que permite indicar de quais endereços IP este Driver aceita ou rejeita conexões (<i>Case 22540</i>).
2.22.75	17/10/2016	F. Englert	Adicionadas propriedades neste Driver para configurar o fuso horário, em minutos, das Estações Mestre e Escravo com as quais este Driver se comunica. Este Driver converte as estampas de tempo locais do computador para o fuso horário da Estação e vice-versa (<i>Case 21621</i>).
2.22.74	30/09/2016	F. Englert	O bloco do Tag de recebimento de comandos SD, DC e RC deste Driver no modo Escravo 101 e 104 agora retorna um <i>array</i> de Variants ao invés de um <i>array</i> de inteiros, permitindo a manipulação em scripts de uma aplicação (<i>Case 21588</i>).
2.22.72	01/09/2016	F. Englert	Atualizada a documentação com as implementações da versão 2.22, somente em português (<i>Case 21167</i>).
2.22.70	18/08/2016	F. Englert	A janela de propriedades foi remodelada. A aba Properties agora mostra todas as propriedades de configuração. A aba Stations foi movida para dentro da aba IEC870 (<i>Case 21385</i>).
2.22.63	30/03/2016	F. Englert	Adicionada uma propriedade para configurar o <i>Originator Address</i> que vai junto com todas as ASDUs enviadas por este Driver, caso o número de octetos da Causa de Transmissão seja configurado com o valor 2 (dois) (<i>Case 21126</i>).
2.22.61	17/03/2016	F. Englert	Ao receber um pacote com o bit FCB repetido, este Driver no modo Escravo agora reenvia corretamente pacotes de ACK ou NACK (<i>Case</i>

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			20685).
2.22.60	11/02/2016	F. Englert	• Este Driver agora marca corretamente como LinkActive as Estações do protocolo 101 Unbalanced Master, após a correta inicialização (Case 20531). • Este Driver no modo 101 Balanceado Mestre agora está respondendo de forma imediata aos comandos de RESET_LINK enviados por um Escravo, tornando a inicialização mais rápida (Case 20533).
2.22.59	04/02/2016	F. Englert	• Removida a opção Respond using do protocolo Escravo 101 Não Balanceado (Case 20515).
2.22.58	03/02/2016	F. Englert	• Implementado o protocolo Procome (Case 19889).
2.22.55	04/12/2015	F. Englert	• Adicionada uma opção para controlar o uso do <i>flag SU (Summertime)</i>, Timestamp SU Flag (Summertime adjustment) (Case 20163).
2.22.54	21/11/2014	F. Englert	• Este Driver agora registra no log, ao lado do número de uma causa de transmissão, o significado (Case 17694). • Este Driver no modo Escravo agora suporta 2 (dois) intervalos de dados para envio cíclico (Case 17708).
2.22.53	13/11/2014	F. Englert	• Este Driver no modo Mestre não ocupa mais 100% de CPU caso a opção Slave Polling Rate seja configurada para 0 (zero) (Case 15377).
2.22.52	29/10/2014	F. Englert	• Este Driver nos modos Escravo 101 e 104 agora descarta os dados de uma Interrogação Geral pendente (Case 17626). • Melhoradas as respostas deste Driver no modo Escravo a pedidos de Interrogação Geral com COT (Cause of Transmission) ou IOA (Information Object Address) inválidos ou não suportados (Case 17629).
2.22.51	28/10/2014	F. Englert	• Resolvido um GPF que ocorria neste Driver em modo Escravo 104 ou Escravo 101 Balanceado logo após este responder a uma Interrogação Geral (Case 17643).
2.22.50	23/10/2014	F. Englert	• Criados novos parâmetros de configuração para definir se este Driver pode enviar dados de classe 1 (um) em requisições de classe 2 (dois), caso não tenha dados de classe 2 (dois) para enviar e vice-versa (Case 17608).
2.22.46	23/05/2013	F. Englert	• Este Driver no modo Escravo agora suporta o envio de dados cíclicos nas opções Cyclic Data Transmission: Object Range e Cyclic Data Transmission: Send Period (ms) da aba Properties (Case 13952).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			Implementado suporte a <i>callbacks</i> neste Driver em modo Mestre na opção Enable Callbacks da aba Properties (<i>Case 9178</i>).
2.22.45	03/04/2013	F. Englert	Corrigido um problema no tratamento de pacotes deste Driver no modo Escravo 101 Balanceado, que podia descartar incorretamente uma resposta de um Mestre caso este enviasse um novo pedido antes de enviar a resposta (<i>Case 13942</i>).
2.22.44	22/03/2013	F. Englert	Corrigido o empacotamento dos ASDUs 5, 7, 13, 15 e 21, que poderiam ser enviados utilizando uma sequência de objetos, opção não suportada pelo padrão do protocolo (<i>Case 13898</i>).
2.22.43	26/02/2013	F. Englert	Corrigido um problema no tratamento de pacotes, que poderia reutilizar pacotes antigos recebidos antes da última desconexão, o que deixava este Driver continuamente desconectando e reconectando. Também foi acelerado o processo de desconexão ou reconexão forçado ao detectar algum erro de sequência nos pacotes (<i>Case 13808</i>).
2.22.41	07/11/2012	F. Englert	Corrigido um vazamento de memória que ocorria sempre que este Driver recebia um dado e o colocava na memória <i>cache</i> interna (<i>Case 13454</i>).
2.22.37	15/02/2012	F. Englert	Adicionada uma nova opção Send EI (End Of Initialization) When Connected na aba Properties da janela de propriedades deste Driver no modo Escravo, que permite configurar se envia um comando EI (<i>end of initialization</i>) quando um Mestre se conecta (<i>Case 12761</i>).
2.22.36	18/11/2011	F. Englert	- Este Driver no modo Escravo 104 agora envia a ASDU 70 (M_EI_NA_1) quando a comunicação é iniciada (<i>Case 12575</i>). - Os bits BL e OV agora são mapeados no campo Limit da qualidade, ou seja, os 2 (dois) bits menos significativos (<i>Case 12569</i>).
2.22.35	13/05/2011	F. Englert	Criada uma nova opção na aba Properties dos protocolos no modo Mestre, Don't wait ACTTERM for Execute commands (<i>Case 12229</i>).
2.22.34	18/04/2011	F. Englert	- Corrigido o empacotamento de alguns tipos de dados que não estavam permitindo o envio de dados em bloco (<i>Case 12161</i>). - Este Driver no modo Escravo agora permite que um usuário configure uma banda morta para envio de valores analógicos na opção Analog Deadband (% of previous value) da aba Properties (<i>Case 12109</i>).
2.22.33	11/02/2011	F. Englert	Este Driver no modo 101 Mestre Balanceado respondia o bit DIR com o valor 0 (zero) na mensagem RESPOND_STATUS, quando deveria

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			responder o bit DIR com o valor 1 (um). Adicionada a opção DIR (Physical Transmission Direction), que controla o comportamento do bit DIR no pacote do protocolo no modo 101 Balanceado, tanto Mestre quanto Escravo (Case 10174). - Corrigida a causa de transmissão utilizada por este Driver no modo Escravo 101 ao retornar dados de comandos, como por exemplo uma resposta de interrogação geral por classe 1 (um) (Case 12057). - Os protocolos 101 Escravo Balanceado e Não Balanceado agora suportam o comando Delay Acquisition Command (C_CD_NA_1, ASDU 106) (Case 12058). - Este Driver no modo Mestre agora permite habilitar o envio de Test Command sempre que a camada de aplicação permanecer <i>n</i> segundos sem receber uma mensagem de um Escravo. Também foi removido o Tag com o parâmetro <i>N2</i> igual a 1104, que enviava um comando de teste sempre que este Tag era lido (Case 12042).
2.22.32	08/02/2011	F. Englert	- Este Driver, tanto no modo Mestre quanto no modo Escravo, agora força uma desconexão caso não receba um TESTDTcon como resposta a um TESTDT act (Case 12043). - Adicionada uma opção Treat First Write As na aba Properties dos protocolos Escravos (Case 12045).
2.22.31	18/01/2011	F. Englert	Adicionada uma nova opção Link Layer Starts na aba Properties da janela de configuração deste Driver, que permite definir o estado inicial do <i>link layer</i>, habilitado ou desabilitado (Case 12023).
2.22.30	09/12/2010	F. Englert	Este Driver, quando no modo Mestre, agora força uma reconexão caso falhe o recebimento da confirmação, positiva ou negativa, de uma Interrogação Geral (Case 11931).
2.22.29	08/12/2010	F. Englert	- Este Driver agora trata graciosamente estampas de tempo acima do ano 2035. Ao detectar uma estampa de tempo inválida, converte-a para a data e hora local e habilita o <i>flag IV</i> da qualidade da estampa de tempo (Case 11855). - Este Driver agora agrupa os valores com formato de estampa de tempo diferente mas com mesmo tipo de dados, eliminando a necessidade de um cliente declarar explicitamente o formato de estampa de tempo de um Tag. A opção Remove timestamp dos dados recebidos foi removida, pois esta operação é realizada automaticamente. - Implementada a opção de <i>local override</i> nos Tags deste Driver no modo Escravo (Case 11662). - Adicionada a opção Analog Values Send Interval (ms) para controlar a frequência de envio de valores analógicos por este Driver no modo Escravo

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			(Case 11732). Adicionada uma nova propriedade a este Driver, IEC.MaxPacketSize, que permite configurar o tamanho máximo dos pacotes montados, tanto no modo Mestre quanto no modo Escravo (Case 11731).
2.22.27	13/08/2010	F. Englert	- Este Driver no modo Escravo agora trata corretamente os comandos de Sincronização de Relógio enviados por um Mestre, de acordo com o perfil de tratamento configurado nas propriedades (Case 11660). - Este Driver agora reporta o valor do <i>flag SB (substituted)</i> na qualidade OPC dos Tags (Case 11661).
2.22.26	05/08/2010	F. Englert	O Tag Read Slave Statistics não está atualizando os valores em nenhuma das opções disponíveis para configuração, com este Driver configurado no modo Escravo Não Balanceado (Case 11623).
2.22.25	04/08/2010	F. Englert	Adicionado uma nova opção Common Address of ASDU no grupo Field sizes (in octets) da janela de propriedades deste Driver, que permite configurar um número de octetos da opção Common Address of ASDU diferente da opção Slave Address (Case 11634).
2.22.23	18/08/2009	F. Englert	- Quando o número de octetos da Causa de Transmissão era 2 (dois), este Driver estava considerando o segundo octeto como parte da causa de transmissão, mas na verdade este octeto é o Originator Address. Agora este Driver descarta o Originator Address nos pacotes recebidos. Na transmissão este Driver sempre envia o valor 0 (zero) no Originator Address (Case 10723). - Agora a configuração deste Driver é realizada em apenas uma aba, IEC870. Nesta aba, conforme o protocolo selecionado, são mostradas outras abas específicas, General, Master ou Slave, Properties e Stations (Case 10643).
2.22.22	07/08/2009	F. Englert	Este Driver no modo Escravo agora aproveita melhor a largura de banda, agrupando em um mesmo pacote vários eventos não solicitados para enviá-los a um Mestre (Case 10702).
2.22.21	03/08/2009	F. Englert	Agora é possível informar o endereço de objetos com 3 (três) octetos utilizando os parâmetros no formato String na propriedade ParamItem (Case 9396).
2.22.20	29/06/2009	F. Englert	- Corrigido o tratamento de reconexões no protocolo 104 Mestre nos casos em que ocorre uma desconexão e reconexão muito rápida (Case 8039). - Adicionada uma nova aba Properties a este Driver, que permite editar, em formato de lista, as propriedades de cada protocolo, anteriormente

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			disponíveis na aba IEC , bem como as propriedades T0 , T1 , T2 , T3 , K e W do protocolo 104 (Case 10608).
2.22.19	19/05/2009	F. Englert	• Este Driver agora realiza uma desconexão do meio físico caso um Escravo pare de responder (Case 10126).
2.22.18	14/10/2008	F. Englert	• Este Driver agora força uma desconexão do meio físico quando recebe um pacote mal-formado ou fora de ordem (Case 10040).
2.22.16	12/03/2008	F. Englert	• Este Driver no modo Mestre agora retenta automaticamente uma Interrogação Geral caso receba uma indicação de final de Interrogação Geral (cot igual a 10) e não haja uma Interrogação Geral ativa e a última Interrogação Geral tenha falhado (Case 9260).
2.22.15	10/03/2008	F. Englert	• Criada uma nova função (<i>N2</i>) para ler dados da memória <i>cache</i> deste Driver (Case 9166).
2.22.14	19/02/2008	F. Englert	• Adicionada uma nova opção na aba Commands deste Driver, disponível apenas para os protocolos Mestres, Process data only at GI end (Case 9165).
2.22.13	08/11/2007	F. Englert	• Corrigido o funcionamento do comando Read Interrogation Group neste Driver em modo Mestre (Case 8402). • Este Driver no modo Escravo agora permite configurar <i>buffers</i> para armazenagem de eventos de classe 1 (um) e 2 (dois) enquanto um Mestre está desconectado (Case 7658). • Implementada uma nova aba Stations na janela de configuração deste Driver. Esta aba permite que um usuário configure se este Driver vai criar as Estações automaticamente, o modo antigo, ou se vai restringir a comunicação às Estações declaradas pelo usuário, o modo novo (Case 7687). • Agora o protocolo 104 Mestre utiliza a opção Wait M_EI_NA on activation para controlar o comportamento da espera pelo STARTDT.CON (Case 8963).
2.22.12	19/06/2007	F. Englert	• Corrigido o tratamento do bit FCB, que estava sendo incrementado antes de saber se o pacote foi enviado com sucesso ou não (Case 8405). • Corrigido o tratamento de comandos Read Interrogation Group neste Driver em modo Escravo (Case 8406). • Implementado um descarte das mensagens inválidas. Se um RESPOND_NACK é recebido em resposta a um SEND/CONFIRM, este é descartado e este Driver continua aguardando pelo CONFIRM_ACK ou CONFIRM_NACK (Case 8151).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			Adicionado uma nova caixa de seleção na aba Commands para permitir esperar ou não pelo ACTTEM de uma Interrogação Geral, Wait ACTTERM (<i>Case 8401</i>).
2.22.11	13/04/2007	F. Englert	Agora as escritas em um Tag com o parâmetro <i>N2</i> igual a 1102 só retornam erro ao falhar a comunicação. Se um equipamento retornar um código de erro, a escrita é considerada bem-sucedida e o código de erro retorna no status da escrita (<i>Case 8018</i>).
2.22.10	13/04/2007	F. Englert	- Corrigido o comportamento deste Driver no modo Escravo, que estava ignorando a qualidade dos dados recebidos de uma aplicação e enviando para um Mestre sempre com qualidade Boa (<i>Case 8111</i>). - Adicionado um novo Tag com o parâmetro <i>N2</i> igual a 6 (seis) neste Driver no modo Escravo, que permite enviar dados espontaneamente com classe 1 (um) e também em uma Interrogação Geral, se configurado (<i>Case 7653</i>).
2.22.8	23/03/2007	F. Englert	Corrigido o comportamento do <i>timer T2</i> no protocolo 104, que poderia fazer com que este Driver excedesse o tempo máximo para envio de um pacote de ACK, ou seja, 10 segundos (<i>Case 8038</i>).
2.22.7	06/03/2007	F. Englert	- Corrigido o formato do parâmetro <i>N2</i> igual a 99, que estava retornando uma data no formato Double do Elipse SCADA (<i>Case 7954</i>). - Adicionada à aba Commands uma lista de seleção que permite selecionar o modo como este Driver no modo Escravo trata comandos de Select ou Execute, default (pass to application): disabled, pass to the application ou handle automatically (<i>Case 5963</i>).
2.22.6	16/02/2007	F. Englert	- Corrigido o Common Slave Address dos pacotes enviados por este Driver no modo Escravo durante uma Interrogação Geral, caso o usuário tenha configurado uma lista pré-definida de pontos a retornar (<i>Case 7940</i>). - Adicionado um novo grupo de controles na aba Commands para configurar o envio automático de comandos de sincronização de relógio, Auto-run clock synchronization e Repeat every (sec) (<i>Case 7509</i>). - Implementado o suporte ao ASDU 105 (C_RP_NA_1, Reset process command) tanto no Mestre quanto no Escravo 101 ou 104 (<i>Case 7836</i>).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
2.22.3	10/01/2007	F. Englert	• Corrigido o tamanho máximo de pacotes do protocolo 104, que permitia gerar pacotes com até 4 (quatro) bytes a mais do que o permitido, ocasionando problemas na comunicação (Case 7798). • Este Driver agora é protegido pelo ID 31290 (Case 7797). • Agora a qualidade é alterada para 216 quando se gera uma estampa de tempo. Isto só ocorre se a qualidade antes era Boa (192). Se a qualidade tinha qualquer valor diferente de 192, não é alterada (Case 6569).
2.22.1	28/09/2006	F. Englert	• Corrigido o tratamento de pontos duplos recebidos com valor inválido (00b ou 11b), que eram descartados caso estivessem configurados para tratar pontos duplos como um valor Booleano (Case 7423). • Alterado o tratamento de pontos duplos com valores inválidos (Case 6571).
2.21.1	18/08/2006	F. Englert	• Corrigido o bit FCB enviado nos pacotes dos protocolos Mestre 101 e 102, que era enviado invertido (Case 7286).
2.20.1	11/08/2006	F. Englert	• Corrigido o script de exemplo do E3, Power ou Water para tratamento de comandos recebidos por um Escravo (Case 5452). • Agora registra-se no log os eventos que são descartados quando a memória <i>cache</i> interna enche, ou seja, mais de 100 eventos para um único objeto (Case 6563). • Corrigida a estampa de tempo incorreta (zero) enviada quando são utilizados os tipos de dados do protocolo 870-5-104 no modo Escravo (Case 6564). • Corrigido o parâmetro Slave polling rate, que não tem efeito na comunicação com um Escravo (Case 6570). • Corrigido o recebimento de um dado com estampa de tempo marcada como inválida (IV igual a um), em que o próprio dado era marcado como inválido e deveria apenas ser descartada a estampa de tempo (Case 6666). • Os protocolos Mestre e Escravo foram extensivamente testados, e foram realizadas pequenas melhorias (Case 6798). • Corrigida a utilização de um <i>time-out</i> fixo de um segundo, ao invés de utilizar o tempo configurado na opção App Layer Timeout, quando precisa enviar um comando e o nível de aplicação está ocupado, por exemplo ao utilizar uma Interrogação Geral, levando um comando a falhar (Case 6848). • Corrigidas as estampas de tempo enviadas ou recebidas, que poderiam estar incorretas com uma diferença de um milissegundo quando recebidas ou enviadas por uma aplicação. Esta correção só funciona em conjunto com o E3 versão 2.5.131 e a biblioteca IOLite versão 1.13 (Case 7256). • Adicionado suporte às ASDUs de comando 48 (C_SE_NA_1), 49 (C_SE_NB_1), 50 (C_SE_NC_1) e 51 (C_BO_NA_1) (Case 5361). • Adicionado suporte ao protocolo 870-5-104 Escravo. Adicionados os comandos das ASDUs 58 (C_SC_TA_1), 59 (C_DC_TA_1), 60

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			(C_RC_TA_1), 61 (C_SE_TA_1), 62 (C_SE_TB_1), 63 (C_SE_TC_1) e 64 (C_BO_TA_1). Também adicionado suporte ao comando de teste (<i>Test Command</i>) utilizando a ASDU 107 (C_TS_TA_1) (<i>Case 5506</i>). • Implementadas atualizações no protocolo ZIV (<i>Case 5846</i>). • Modificada a leitura de ajustes de VTCD para se ajustar ao novo formato definido no protocolo ZIV (<i>Case 6561</i>). • Adicionada a opção Modify the timestamp on received values na aba Interrogation (<i>Case 6562</i>). • Adicionado suporte ao commando de sincronização de relógio (<i>Clock Synchronization Command</i>, ASDU 103 ou C_CS_NA_1) nos protocolos Escravos (101 e 104). Também foi adicionada a aba Commands (<i>Case 6826</i>). • Adicionado suporte ao protocolo 870-5-104 Mestre (<i>Case 7259</i>).
2.19.1	17/01/2005	F. Englert	• Adicionada uma opção para usar o Common Address diferente do Link Address (<i>Case 4364</i>). • Corrigida uma situação em que não se suportava endereços de Escravos maiores que 127 quando configurado para utilizar um octeto para o endereço deste Escravo, que deveria permitir endereços entre 0 (zero) e 255 (<i>Case 4766</i>). • Corrigida uma situação em que não se reconhecia pacotes CC1 recebidos como sendo pacotes de ACK (<i>Case 5210</i>). • Corrigida uma situação em que os tipos de dados do protocolo 104 estavam retornando a estampa de tempo como o valor do Tag (<i>Case 5211</i>). • Corrigido o bloco do protocolo ZIV, Read Configuration Block 2, com o parâmetro <i>B2</i> igual a 1141 (<i>Case 5212</i>). • Agora utiliza-se a biblioteca IOKit versão 1.8, com suporte a ligar e desligar o log em tempo de execução.
2.18.1	28/07/2004	F. Englert	• Corrigida uma situação em que não se enviava os dados de uma Interrogação Geral em blocos no modo Mestre Não Balanceado (<i>Case 4238</i>). • Corrigida uma situação em que, apenas no modo Escravo, não se reenviava o último pacote quando um Mestre enviava um pacote com o mesmo FCB anterior (<i>Case 4276</i>). • Adicionada uma opção para selecionar o tipo de resposta que se deve enviar quando se recebe um pedido de um Mestre, <i>Respond with</i> (<i>Case 3741</i>). • Os tipos de dados definidos no protocolo 870-5-104 agora estão disponíveis na lista de pontos pré-definidos retornados em uma Interrogação Geral (<i>Case 4035</i>).
2.17.1	09/06/2004	F. Englert	• Adicionadas retentativas no <i>link layer</i> (<i>Case 3937</i>).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
2.16.1	04/05/2004	F. Englert	• Modificado o valor padrão de <i>time-out</i> de aplicação, de 4000 para 4 (quatro) segundos. • Adicionada a opção Add timestamp to interrogation data (Case 3608). • Adicionada uma opção para permitir tratar os dados DIQ, DCO, RCO e SEP como valores Booleanos (Case 3609).
2.15.1	01/04/2004	F. Englert	• Adicionada a opção Remove timestamp from interrogation data (Case 3506). • Adicionado suporte aos tipos de dados do protocolo IEC 870-5-104 encapsulados no protocolo IEC 870-5-101. O protocolo 104 NÃO É suportado (Case 3419). • Traduzida a documentação para o português (Case 3568).
2.14.1	20/02/2004	F. Englert	• As estampas de tempo enviadas estão incorretas, ou seja, usa-se sempre a hora atual do computador (Case 3385).
2.13.1	29/01/2004	F. Englert	• Comandos de ponto duplo (DCO) e Eventos Simples de Equipamento de Proteção (SEP) agora usam os estados possíveis, Intermediário ou 0 (zero), OFF ou 1 (um), ON ou 2 (dois) e Indeterminado ou 3 (três), ao invés de um valor Booleano indicando o estado Ligado ou Desligado (Case 3214). • Este Driver não está respondendo aos comandos de Test Link, a função 2 (dois) do protocolo, recebidos no modo Escravo ou Balanceado (Case 3231). • Adicionada a opção Interrogation, que permite configurar a lista de objetos retornados em uma Interrogação Geral apenas no modo Escravo (Case 3284). • Adicionado o Tag TEST LINK apenas no modo Escravo ou Balanceado. Este Tag envia um comando de teste do <i>link</i> para um Mestre e retorna o valor 1 (um) se este Mestre responder (Case 3285). • Corrigido um problema em que este Driver em modo Escravo ou Balanceado às vezes trancava ao receber uma mensagem de um Mestre enquanto tentava enviar uma outra mensagem para o mesmo Mestre (Case 3286).
2.12.1	08/01/2004	F. Englert	• Dados de ponto duplo (M_DP_NA_1 e M_DP_TA_1) agora retornam os estados possíveis, Intermediário ou 0 (zero), OFF ou 1 (um), ON ou 2 (dois) e Indeterminado ou 3 (três), ao invés de um valor Booleano indicando o estado Ligado ou Desligado (Case 3214). • Adicionada uma opção para habilitar ou desabilitar o uso do <i>frame</i> CC1 ao invés do <i>frame</i> de ACK no envio de mensagens de quitação (<i>acknowledgement</i>) do nível de enlace em modo Balanceado (Case 3215).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
2.11.1	19/12/2003	F. Englert	• Adicionado o tratamento de comandos enviados por um Mestre no modo Escravo (Case 3211).
2.10.1	18/12/2003	F. Englert	• Implementado o protocolo 870-5-101 Escravo no modo Balanceado (Case 195).
2.9.1	02/12/2003	F. Englert	• Os Escravos não são mais criados quando em modo Offline (Case 3039).
2.8.1	27/11/2003	F. Englert	• Implementado o protocolo 870-5-101 Escravo (Case 195).
2.7.1	18/11/2003	F. Englert	• O modo Escuta agora funciona corretamente, pois os valores recebidos neste modo não estavam sendo retornados (Case 2960).
2.6.1	13/11/2003	F. Englert	• Erros na qualidade do Tag agora são reportados se a conexão é perdida enquanto está em modo Escuta (Case 2935).
2.5.1	07/11/2003	F. Englert	• Adicionado um campo de qualidade à leitura de memória de massa, com o parâmetro B2 igual a 1189 ou 1190 (Case 2893).
2.4.1	05/11/2003	F. Englert	• Adicionado o Tag Read MAE ID , com o parâmetro N2 igual a 1212 para Escravos no protocolo ZIV .
2.3.1	29/09/2003	F. Englert	• Melhorado o log de mensagens no nível de aplicação.
2.2.1	11/08/2003	F. Englert	• Adicionada a documentação para extensões ZIV .
2.1.1	29/07/2002	F. Englert	• Driver portado para a biblioteca IOKit . • Implementados os protocolos 870-5-102 e 870-5-102 ZIV extensions .