Driver IEC 870-5-104 Mestre/Escravo

Nome do Arquivo	IEC870-104.dll
Fabricante	International Electrotechnical Commission
Equipamentos	PLCs e estações remotas compatíveis com o protocolo IEC 870-5-104
Protocolo	IEC 870-5-104 Mestre e IEC 870-5-104 Escravo
Versão	4.0.25
Última Atualização	21/09/2026
Plataforma	Win32
Dependências	Nenhuma
Leitura com Superblocos	Não
Nível	31201

Introdução

O Driver IEC 870-5-104 Mestre/Escravo implementa os modos **Mestre** e **Escravo** do protocolo **IEC 870-5-104**.

Configuração do Driver

Esta seção contém informações sobre a configuração deste Driver.

Parâmetros [P]

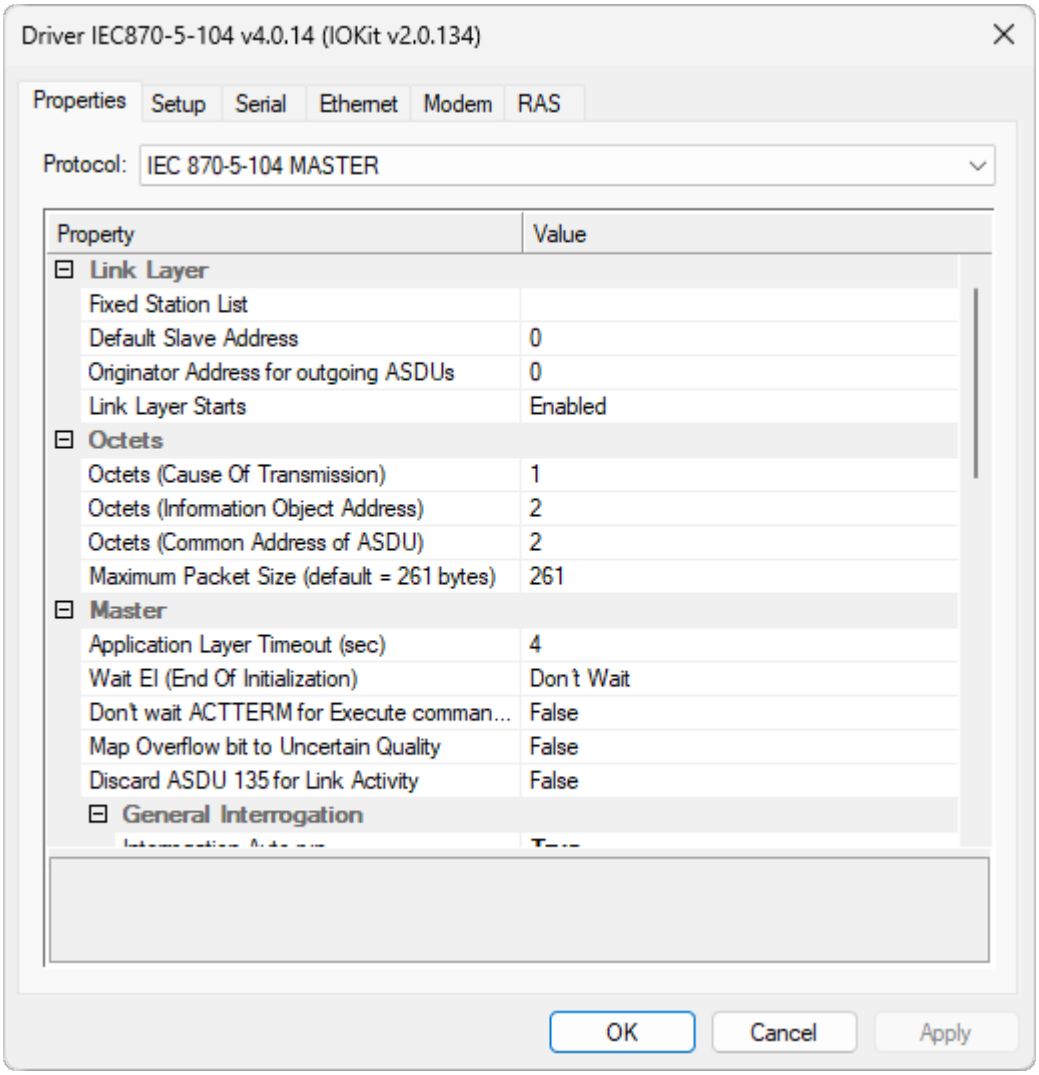
O Driver IEC 870-5-104 Mestre/Escravo não utiliza os parâmetros **[P]**. Recomenda-se, por questões de compatibilidade futura, que o valor destes parâmetros seja configurado em 0 (zero).

Janela de Propriedades

A Janela de Propriedades deste Driver é composta de diversas abas. A aba que configura o protocolo em si é a aba **Properties**. 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 Properties

A aba **Properties** permite visualizar e configurar **todas as propriedades** de configuração deste Driver, mostrada na figura a seguir.



Aba Properties

As propriedades deste Driver estão agrupadas por categorias e descritas a seguir.

Opção Protocol

Opção que permite selecionar o protocolo a ser utilizado por este Driver. As opções disponíveis são **IEC 870-5-104 MASTER** ou **IEC 870-5-104 SLAVE**. A tabela a seguir contém todas as propriedades disponíveis para o protocolo selecionado. As propriedades em negrito indicam propriedades com um valor diferente do padrão. Para retornar o valor de uma propriedade para o padrão, use a tecla DEL.

Grupo Link Layer

Opções gerais para a camada de dados.

Opções disponíveis para o grupo Link Layer

OPÇÃO	DESCRIÇÃO
Fixed Station List	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. Através desta opção é possível

OPÇÃO	DESCRIÇÃO
	declarar uma lista de Estações que este Driver utiliza. Se esta opção não está preenchida, este Driver opera de forma automática, criando Estações à medida que sejam necessárias. O modo automático é desaconselhado para protocolos no modo **Escravo**, pois Estações incorretas podem ser criadas de acordo com as requisições realizadas por um **Mestre**. Se este Driver está no modo **Escravo**, apenas uma Estação deve ser criada, informando apenas o endereço deste **Escravo**. Se este Driver está no modo **Mestre**, mais de uma Estação pode ser criada, informando a sequência de endereços (*Common Address of ASDU*) separados por vírgulas
Default Slave Address	O endereço da Estação de cada Tag pode ser informado no parâmetro **Device** ou no parâmetro *N1*. Esta opção configura o endereço padrão de Estação utilizado por este Driver quando o parâmetro *N1* ou *B1* de um Tag é igual a 0 (zero)
Originator Address for Outgoing ASDUs	Um *Originator Address* é um segundo campo de endereço no protocolo **IEC 870-5-104**. Este endereço é transmitido no segundo octeto da opção **Octets (Cause of Transmission)**. Caso seja necessário enviar o *Originator Address*, configure a opção **Octets (Cause of Transmission)** com o valor 2 (dois)
Link Layer Starts	Informa se este Driver já inicia comunicando ou se aguarda a inicialização através da escrita no Tag especial com o parâmetro *N2* igual a 996

Grupo Octets

Configura o tamanho, em octetos (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.

Opções disponíveis para o grupo Octets

OPÇÃO	DESCRIÇÃO
Octets (Cause of Transmission)	Configura o número de octetos da Causa de Transmissão de uma ASDU. Os valores permitidos são 1 (um) ou 2 (dois). O segundo octeto, quando habilitado, contém o valor da opção **Originator Address**
Octets (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)
Octets (Common Address of ASDU)	Configura o número de octetos do endereço comum de uma ASDU, ou o endereço de um dispositivo lógico dentro de um equipamento físico. Por padrão esta propriedade segue o número de octetos da opção **Default Slave Address**. Os valores permitidos são 1 (um) ou 2 (dois)
Maximum Packet Size	Informa o tamanho máximo de um pacote de dados. O valor padrão desta opção é 261

Grupo Master

Opções gerais para quando este Driver está no modo **Mestre**.

Opções disponíveis para o grupo Master

OPÇÃO	DESCRIÇÃO
Application 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, como por exemplo **Interrogação Geral** ou **Seleção ou Execução**, possam ser executados com sucesso
Wait EI (End of Initialization)	Selecione esta opção para forçar este Driver a aguardar o recebimento da ASDU **M_EI_NA** para que um **Escravo** seja considerado conectado
Don't Wait ACTTERM for Execute Commands	Desmarque esta opção para forçar este Driver a aguardar o recebimento de uma sinalização de término do Comando de Execução. Alguns **Escravos** não enviam a sinalização de término, portanto nestes casos esta opção deve permanecer selecionada
Map Overflow bit to Uncertain Quality	No processamento dos *flags* de ASDUs analógicas, informa se o bit de *overflow* (**OV**) deve ser mapeado para uma qualidade incerta (84, **EU_EXCEEDED** do padrão OPC)
Discard ASDU 135 for Link Activity	Descarta a ASDU 135 (privada) para considerar um *link* como ativo. Logo, se somente esta mensagem é recebida em um canal, um *link* é considerado inativo
General Interrogation	Configura a forma como este Driver no modo **Mestre** envia e processa um **Comando de Interrogação Geral**. As opções disponíveis são **Interrogation Auto-run**: Habilita o envio de um Comando de Interrogação Geral sempre que este Driver estabelece uma conexão com um **Escravo**, **Interrogation Repeat**: Habilita a repetição do envio de um Comando de Interrogação Geral, **Interrogation Repeat Period (sec)**: Informa o intervalo de repetição de um Comando de Interrogação Geral em segundos, **Interrogation Wait ACTTERM**: Força este Driver a aguardar o recebimento de uma sinalização de término de um Comando de Interrogação Geral. Alguns **Escravos** não enviam a sinalização de término, portanto nestes casos esta opção deve permanecer desmarcada, **Interrogation Process At End**: Selecione esta opção para **processar os dados apenas ao final de um Comando de Interrogação Geral**, **Disconnect if General Interrogation returns empty results**: Força uma desconexão deste Driver caso o resultado de um Comando de Interrogação Geral retorne uma lista vazia e **Use Broadcast**: Informa se um Comando de Interrogação Geral deve utilizar um endereço de *broadcast*
Test Command	Configura o envio automático de um **Comando de Teste**. As opções disponíveis são **Test Command Enabled**: Habilita o envio periódico de um Comando de Teste

OPÇÃO	DESCRIÇÃO
	sempre que este Driver permanecer um determinado tempo, em segundos, sem efetuar comunicações no nível de aplicação e **Test Command Idle Time (sec)**: Informa o período de um Comando de Teste
Clock Sync	Configura o envio automático de um **Comando de Sincronização de Relógio**. As opções disponíveis são **Clock Sync Auto-run**: Habilita o envio de um Comando de Sincronização de Relógio sempre que este Driver estabelece uma conexão com um **Escravo**, **Clock Sync Repeat**: Habilita o envio de um Comando de Sincronização de Relógio a intervalos fixos e **Clock Sync Repeat Period (sec)**: Informa o período da repetição da sincronização, em segundos

Grupo Slave

Opções gerais para quando este Driver está no modo **Escravo**.

Opções disponíveis para o grupo Slave

OPÇÃO	DESCRIÇÃO
Send EI (End of Initialization) When Connected	Configura o envio de um comando **EI** (*End of Initialization*) quando conectado, informando que um **Escravo** está pronto para a comunicação. As opções disponíveis são **Send EI** ou **Don't Send EI**
Analog Values Send Interval (ms)	Intervalo de envio das mudanças dos valores analógicos, em milissegundos
Analog Deadband (% of previous value)	Banda morta padrão para Tags analógicos. Pode-se também informar uma banda morta individual por Tag configurando o parâmetro **Item**
Treat First Write As	Informa se a primeira escrita de um Tag é considerada um novo evento (opção **New Event**) ou apenas armazena como o valor corrente (opção **Current Value**). Neste último caso, somente a partir da segunda escrita é possível gerar um evento, caso tenha ocorrido mudança de qualidade ou violação da banda morta
Use Cache File	Indica que é usado um arquivo de *cache* para armazenar e recuperar os últimos valores deste Driver na inicialização. Neste caso, a opção **Treat First Write As** força o modo **New Event**, visto que o primeiro valor é obtido deste arquivo de *cache*. Este arquivo é binário e contém duas seções, uma com os dados correntes e outra com eventos. A cada 1 (uma) hora de execução (valor fixo), este arquivo é re-escrito somente com os dados correntes, consolidando ou removendo todos os eventos. Conforme os eventos ocorrem nesse intervalo, são adicionados ao final deste arquivo, com o objetivo de obter a melhor performance possível na gravação
Cache File Path	Diretório onde um arquivo de *cache* é gravado ou lido. Pode ser um caminho absoluto ou relativo, como por

OPÇÃO	DESCRIÇÃO
	exemplo "\folder\file". A extensão é livre e sugere-se os valores ".cache" ou ".104cache"
Cache Startup/Shutdown Mode Time (s)	Tempo, em segundos, que este Driver aguarda para reportar valores nulos. Na inicialização este tempo é contado automaticamente. No término, ou *shutdown*, uma aplicação deve avisar este Driver que a origem dos dados, como por exemplo um Driver no modo **Mestre**, foi parado usando uma escrita no Tag **Shutdown**., como por exemplo usando o código "Application.GetObject(':IEC104_LAN_Slave.ShutDown').WriteEx 1" em um evento **BeforeStop**
General Interrogation	Configura o conjunto de dados que é retornado a um **Mestre** em resposta a um Comando de Interrogação Geral. Para mais informações, consulte o tópico **Interrogação Geral**
Command Handling	As opções disponíveis são **Handle Mode for Clock Sync commands**: Configura a ação que este **Escravo** executa quando recebe um **Comando de Sincronização de Relógio**, **Handle Mode for Select/Execute commands**: Configura a ação que este **Escravo** executa quando recebe um Comando de Seleção ou Execução e **Handle Mode for Test commands**: Configura a ação que este **Escravo** executa quando recebe um Comando de Teste
Cyclic	As opções disponíveis são **Cyclic Data Transmission (1): Send Period (ms)**: Período para envio de dados de Classe 1 (um). O valor padrão é 5000, **Cyclic Data Transmission (1): Object Range**: Configura a faixa de objetos a serem enviados na Classe 1 (um). A faixa de objetos deve ser informada da mesma forma que na lista de objetos para Comandos de Interrogação Geral, **Cyclic Data Transmission (2): Send Period (ms)**: Período para envio de dados de Classe 1 (um). O valor padrão é 10000 e **Cyclic Data Transmission (2): Object Range**: Configura a faixa de objetos a serem enviados na Classe 1 (um). A faixa de objetos deve ser informada da mesma forma que na lista de objetos para Comandos de Interrogação Geral
Offline Buffers	As opções disponíveis são **Offline Class 1 buffering**: Habilita o armazenamento em memória de eventos de Classe 1 (um, prioridade alta, tipicamente valores digitais), **Offline Class 1 buffer size**: Informa a quantidade máxima de eventos de cada Tag que acumulam em memória para a Classe 1 (um), **Offline Class 2 buffering**: Habilita o armazenamento em memória de eventos de Classe 2 (dois, prioridade baixa, tipicamente valores analógicos) e **Offline Class 2 buffer size**: Informa a quantidade máxima de eventos de cada Tag que acumulam em memória para a Classe 2 (dois)

Grupo 104

Opções de temporização e tratamento de mensagens no protocolo **IEC 870-5-104**, tanto no modo **Mestre** quanto no modo **Escravo**.

Opções disponíveis para o grupo 104

OPÇÃO	DESCRIÇÃO
T0 (sec)	*Time-out* para o estabelecimento de uma conexão
T1 (sec)	*Time-out* para as APDUs de envio ou teste
T2 (sec)	*Time-out* para reconhecimento em caso de não haver mensagens de dados t2 < t1
T3 (sec)	*Time-out* para envio de *frames* de teste em caso de um longo período de inatividade
K	Máxima diferença de mensagens para enviar novas mudanças. O valor padrão desta opção é 12 APDUs
W	Último reconhecimento depois de receber APDUs de informação (**I**). O valor padrão desta opção é 8 (oito) APDUs

Grupo Timestamp

Opções gerais de estampa de tempo.

Opções disponíveis no grupo Timestamp

OPÇÃO	DESCRIÇÃO
Timestamp SU Flag (Summertime adjustment)	Informa o comportamento em relação ao bit **SU** (horário de verão) dos *flags* da estampa de tempo. Os valores possíveis para esta opção são **Ignore** ou **Automatic**
Enable Station TimeZone	Habilita um *offset* de fuso horário para as Estações
Station TimeZone (in minutes)	Informa o fuso horário das Estações, em minutos

Grupo Timestamp (Slave)

Opções de estampa de tempo adicionais quando este Driver está no modo **Escravo**.

Opções disponíveis no grupo Timestamp para Escravos

OPÇÃO	DESCRIÇÃO
SP Timestamp Format	Define o formato da estampa de tempo para tipos de dados **SP**. Os valores possíveis para esta opção são **No timestamp**, **Short timestamp (3 octets)** ou **Full timestamp (7 octets)**
DP Timestamp Format	Define o formato da estampa de tempo para tipos de dados **DP**. Os valores possíveis para esta opção são **No timestamp**, **Short timestamp (3 octets)** ou **Full timestamp (7 octets)**
BO Timestamp Format	Define o formato da estampa de tempo para tipos de dados **BO**. Os valores possíveis para esta opção são **No timestamp**, **Short timestamp (3 octets)** ou **Full timestamp (7 octets)**

OPÇÃO	DESCRIÇÃO
ST Timestamp Format	Define o formato da estampa de tempo para tipos de dados **ST**. Os valores possíveis para esta opção são **No timestamp**, **Short timestamp (3 octets)** ou **Full timestamp (7 octets)**
MEn Timestamp Format	Define o formato da estampa de tempo para tipos de dados **MEn**. Os valores possíveis para esta opção são **No timestamp**, **Short timestamp (3 octets)** ou **Full timestamp (7 octets)**
MEs Timestamp Format	Define o formato da estampa de tempo para tipos de dados **MEs**. Os valores possíveis para esta opção são **No timestamp**, **Short timestamp (3 octets)** ou **Full timestamp (7 octets)**
MEf Timestamp Format	Define o formato da estampa de tempo para tipos de dados **MEf**. Os valores possíveis para esta opção são **No timestamp**, **Short timestamp (3 octets)** ou **Full timestamp (7 octets)**

Grupo Send Priority (Slave)

Prioridade de envio para cada tipo de objeto quando este Driver está no modo **Escravo**.

Opções disponíveis no grupo Send Priority para Escravos

OPÇÃO	DESCRIÇÃO
SP Send Priority	Informa como são enviados os dados para tipos de dados **SP**. Os valores possíveis para esta opção são **Send with HIGH Priority**, **Send with LOW Priority** ou **Send on General Interrogation only**
DP Send Priority	Informa como são enviados os dados para tipos de dados **DP**. Os valores possíveis para esta opção são **Send with HIGH Priority**, **Send with LOW Priority** ou **Send on General Interrogation only**
BO Send Priority	Informa como são enviados os dados para tipos de dados **BO**. Os valores possíveis para esta opção são **Send with HIGH Priority**, **Send with LOW Priority** ou **Send on General Interrogation only**
ST Send Priority	Informa como são enviados os dados para tipos de dados **ST**. Os valores possíveis para esta opção são **Send with HIGH Priority**, **Send with LOW Priority** ou **Send on General Interrogation only**
MEn Send Priority	Informa como são enviados os dados para tipos de dados **MEn**. Os valores possíveis para esta opção são **Send with HIGH Priority**, **Send with LOW Priority** ou **Send on General Interrogation only**
MEs Send Priority	Informa como são enviados os dados para tipos de dados **MEs**. Os valores possíveis para esta opção são **Send with HIGH Priority**, **Send with LOW Priority** ou **Send on General Interrogation only**

OPÇÃO	DESCRIÇÃO
MEf Send Priority	Informa como são enviados os dados para tipos de dados **MEf**. Os valores possíveis para esta opção são **Send with HIGH Priority**, **Send with LOW Priority** ou **Send on General Interrogation only**

Grupo Boolean Values

Opções de conversão de valores nativos para valores Booleanos.

Opções disponíveis no grupo Boolean Values

OPÇÃO	DESCRIÇÃO
Convert DCO to Boolean	Esta opção permite utilizar **Comandos de Ponto Digital Duplo (DC)** com um valor Booleano
Convert DIQ to Boolean	Esta opção permite utilizar **Comandos de Ponto Digital Duplo (DP)** com um valor Booleano
Convert RCO to Boolean	Esta opção permite utilizar **Comandos de Regulação de Passo (RC)** com um valor Booleano
Convert SEP to Boolean	Esta opção permite utilizar Eventos de Proteção de Equipamento com um valor Booleano

Estações

Nos protocolos em que este Driver está no modo **Mestre**, uma Estação representa um **Escravo** com o qual este Driver se comunica. Nos protocolos em que 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 janela de propriedades deste Driver, na propriedade **Fixed Station List** da aba **Properties**.

Se esta opção não está definida, este Driver opera de forma automática, criando Estações com parâmetros padrão à medida em que são necessárias. O modo automático é desaconselhado para protocolos no modo **Escravo**, pois Estações incorretas podem ser criadas de acordo com as requisições realizadas por um **Mestre**.

Tags de Estatísticas e Diagnóstico de uma Estação

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

Estado do Canal de Comunicação

Protocolos	Os valores possíveis são **IEC 870-5-104 MASTER** ou **IEC 870-5-104 SLAVE**
Parâmetros Numéricos	**N1**: Número de uma Estação, **N2**: 993, **N3**: 0 (zero) e **N4**: 0 (zero)
Parâmetros por String	Não disponível
Escrita	Não disponível
Leitura	Retorna o estado do canal de comunicação. Os valores possíveis são **0**: O Canal não está ativo ou **1**: O Canal está ativo
ASDUs	Não disponível

Estado de Conexão de uma Estação

Protocolos	Os valores possíveis são **IEC 870-5-104 MASTER** ou **IEC 870-5-104 SLAVE**
Parâmetros Numéricos	**N1**: Número de uma Estação, **N2**: 992, **N3**: 0 (zero) e **N4**: 0 (zero)
Parâmetros por String	Não disponível
Escrita	Não disponível
Leitura	Retorna o estado de conexão de uma Estação. Os valores possíveis são **0**: A Estação não está conectada ou não está respondendo à camada de enlace ou **1**: A Estação está conectada
ASDUs	Não disponível

Leitura do Log de Eventos

Protocolos	Os valores possíveis são **IEC 870-5-104 MASTER** ou **IEC 870-5-104 SLAVE**
Parâmetros Numéricos	**B1**: Número de uma Estação, **B2**: 99, **B3**: 0 (zero) e **B4**: 0 (zero)
Parâmetros por String	Não disponível
Tamanho do Tag Bloco	8 (oito) Elementos
Escrita	Não disponível
Leitura	Retorna o valor, a qualidade e a estampa de tempo dos dados recebidos
ASDUs	Não disponível

Este Tag Bloco retorna um log de todas as ASDUs recebidas por 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 1 (Word)**: Tipo de dados da ASDU (*type*)

- **Elemento 2 (Word)**: Endereço do objeto
- **Elemento 3 (DWord)**: *Flags* 1 (um)
- **Elemento 4 (DWord)**: *Flags* 2 (dois)
- **Elemento 5 (Double)**: Valor do objeto
- **Elemento 6 (Double)**: Estampa de tempo
- **Elemento 7 (String)**: Nome da ASDU
- **Elemento 8 (Word)**: Causa da Transmissão (COT, *Cause of Transmission*)

Contador de Pacotes Enviados

Protocolos	Os valores possíveis são **IEC 870-5-104 MASTER** ou **IEC 870-5-104 SLAVE**
Parâmetros Numéricos	**N1**: Número de uma Estação, **N2**: 998, **N3**: 0 (zero) e **N4**: 1 (um)
Parâmetros por String	Não disponível
Escrita	Não disponível
Leitura	Retorna o número de pacotes de comunicação já enviados por este Driver para uma Estação
ASDUs	Não disponível

Contador de No-Replies

Protocolos	Os valores possíveis são **IEC 870-5-104 MASTER** ou **IEC 870-5-104 SLAVE**
Parâmetros Numéricos	**N1**: Número de uma Estação, **N2**: 998, **N3**: 0 (zero) e **N4**: 2 (dois)
Parâmetros por String	Não disponível
Escrita	Não disponível
Leitura	Retorna o número de pacotes de comunicação enviados por este Driver sem resposta por parte de uma Estação
ASDUs	Não disponível

Contador de Erros de Frame Recebidos

Protocolos	Os valores possíveis são **IEC 870-5-104 MASTER** ou **IEC 870-5-104 SLAVE**
Parâmetros Numéricos	**N1**: Número de uma Estação, **N2**: 998, **N3**: 0 (zero) e **N4**: 3 (três)
Parâmetros por String	Não disponível
Escrita	Não disponível
Leitura	Contador do número de pacotes recebidos com erro de uma Estação
ASDUs	Não disponível

Contador de Erros de Envio de Frames

Protocolos	Os valores possíveis são **IEC 870-5-104 MASTER** ou **IEC 870-5-104 SLAVE**
Parâmetros Numéricos	**N1**: Número de uma Estação, **N2**: 998, **N3**: 0 (zero) e **N4**: 4 (quatro)
Parâmetros por String	Não disponível
Escrita	Não disponível
Leitura	Retorna o número de falhas de envio de pacotes para uma Estação
ASDUs	Não disponível

Contador de Retentativas de Envio de Pacotes

Protocolos	Os valores possíveis são **IEC 870-5-104 MASTER** ou **IEC 870-5-104 SLAVE**
Parâmetros Numéricos	**N1**: Número de uma Estação, **N2**: 998, **N3**: 0 (zero) e **N4**: 5 (cinco)
Parâmetros por String	Não disponível
Escrita	Não disponível
Leitura	Retorna o número de retentativas de envio de pacotes para uma Estação
ASDUs	Não disponível

Estatísticas Avançadas

Protocolos	Os valores possíveis são **IEC 870-5-104 MASTER** ou **IEC 870-5-104 SLAVE**
Parâmetros Numéricos	**B1**: Número de uma Estação, **B2**: 999, **B3**: 0 (zero) e **B4**: 0 (zero)
Parâmetros por String	Não disponível
Tamanho do Bloco	17 Elementos
Escrita	Não disponível
Leitura	Retorna um Tag Bloco com o valor atual de diversas estatísticas
ASDUs	Não disponível

Os Elementos deste Tag Bloco são os seguintes:

- **Elemento 1 (Long)**: Número de ASDUs criadas
- **Elemento 2 (Long)**: Número de ASDUs removidas
- **Elemento 3 (Long)**: Número de ASDUs inválidas
- **Elemento 4 (Long)**: Número de ASDUs na memória *cache* interna de eventos de um **Escravo**
- **Elemento 5 (Long)**: Número de ASDUs na memória *cache* interna de dados de um **Escravo**
- **Elemento 6 (Long)**: Número de ASDUs na memória *cache* de logs de um **Escravo**
- **Elemento 7 (Long)**: Número de **HVALUEs** alocados
- **Elemento 8 (Long)**: Número de referências para **HVALUEs**
- **Elemento 9 (Long)**: Número de **HTIMERs** alocados
- **Elemento 10 (Long)**: Número de categorias de *timers*
- **Elemento 11 (Long)**: Número de **HVALUEs** criados
- **Elemento 12 (Long)**: Número de bytes usados no *heap* de um processo
- **Elemento 13 (Long)**: Número de blocos de *heap* em um processo
- **Elemento 14 (Long)**: Número de bytes disponíveis no *heap* de um processo
- **Elemento 15 (Long)**: Número de blocos de *heap* disponíveis em um processo
- **Elemento 16 (Long)**: Número de processos no sistema
- **Elemento 17 (Long)**: Número de *threads* no sistema

Tags e Tipos de Dados

Esta seção contém os tipos de dados que podem ser lidos e escritos por este Driver.

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	DESCRIÇÃO	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** ou **MEf**	M_ME_NA_1 (9), M_ME_NB_1 (11) e M_ME_NC_1 (13)	M_ME_TA_1 (10), M_ME_TB_1 (12) e M_ME_TC_1 (14)	M_ME_TD_1 (34), M_ME_TE_1 (35) e 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**. Neste Driver em modo **Escravo**, uma aplicação deve alimentar este Driver com os valores a enviar para um **Mestre**, na forma de escritas. Neste Driver em modo **Mestre**, os valores recebidos são fornecidos para uma aplicação na forma de leituras.

Ponto Digital Simples (SP)

Um **Ponto Digital Simples** é um objeto que armazena um valor **ON** (1, um) ou **OFF** (0, zero). 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**.

Ponto Digital Duplo (DP)

Um **Ponto Digital Duplo** é um objeto que armazena um valor **ON** ou **OFF**, e cada estado corresponde a um bit individual. 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 **IEC 870-5-104**, o valor trafegado nestas ASDUs é um DIQ (*Double Point Information with Quality Descriptor*), que contém quatro bits de **qualidade** (**IV**, **NT**, **SB** e **BL**) e dois bits de valor (chamados de DPI, *Double Point Information*):

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

O valor deste Tag depende da configuração realizada na janela de propriedades deste Driver, na opção **Convert DIQ to Boolean** do grupo **Boolean Values** na aba **Properties**. Se esta opção está desmarcada, o valor de um Tag nas leituras é o próprio valor do DPI, tanto nas leituras quanto nas escritas. As escritas neste Driver falham se este valor não é um dos valores descritos anteriormente. Se a opção **Convert DIQ to Boolean** está selecionada, o valor dos Tags de Ponto Digital Duplo nas leituras é o seguinte:

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

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

Posição de Passo (ST)

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, ou seja, é ignorado na leitura e configurado para 0 (zero) na escrita.

Bitstring (BO)

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**.

Valor de Medida Analógica (ME)

As **Medidas Analógicas** podem ser representadas nos formatos descritos nos tópicos a seguir.

Medida Analógica Normalizada (MEn)

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 em um 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 deste intervalo podem ser representados com precisão, ou seja, podem ocorrer aproximações nos valores enviados e recebidos. Este objeto suporta os bits de **qualidade IV**, **NT**, **SB**, **BL** e **OV**.

Medida Analógica com Escala (MEs)

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 vai de -3276.8 a 3276.7. Se o ponto decimal de um equipamento é diferente de uma casa decimal, então uma escala apropriada tem que ser configurada em um Tag. Este objeto suporta os bits de **qualidade IV**, **NT**, **SB**, **BL** e **OV**.

Medida Analógica em Ponto Flutuante (MEf)

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**.

Total Integrado (IT)

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. Este objeto 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>][<DB:BandaMorta>]
```

Em que:

- **<tipo>** é o mnemônico do tipo de dados, ou seja, **SP**, **DP**, **ST**, **BO**, **MEn**, **MEs** ou **MEf**. Os dois pontos que separam o 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) e **M_SP_TB_1** (30)	SP	SP:1
Ponto Digital Duplo	**M_DP_NA_1** (3), **M_DP_TA_1** (4) e **M_DP_TB_1** (31)	DP	DP:328
Posição de Passo	**M_ST_NA_1** (5), **M_ST_TA_1** (6) e **M_ST_TB_1** (32)	ST	ST:10.434
Bitstring	**M_BO_NA_1** (7), **M_BO_TA_1** (8) e **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) e **M_ME_TD_1** (34)	MEn	MEn:10938
Medida Analógica com Escala	**M_ME_NB_1** (11), **M_ME_TB_1** (12) e **M_ME_TE_1** (35)	MEs	MEs:9833
Medida Analógica em Ponto Flutuante	**M_ME_NC_1** (13), **M_ME_TC_1** (14) e **M_ME_TF_1** (36)	MEf	MEf:10.35.22

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

Formatos dos endereços de objetos

ENDEREÇO DO OBJETO	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.

Opções disponíveis para 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 no grupo **Send Priority** da aba **Properties** das configurações deste Driver	SP:255
!	Quando este caractere é um ponto de exclamação, um dado é enviado com prioridade **Alta** (Classe 1)	SP:255!
<	Quando este caractere é um sinal de menor que, um dado é enviado apenas no próximo Comando de Interrogação Geral	SP:255<
=	Quando este caractere é um sinal de igual, um dado é enviado com prioridade **Baixa** (Classe 2)	SP:255=

- **<DB:BandaMorta>** é uma informação opcional por Tag de uma banda morta em porcentagem, em relação ao valor corrente, usada para definir se houve uma mudança de valor significativa para envio de uma informação, quando este Driver está configurado no modo **Escravo**. Se não é informada, utiliza-se como banda morta a definição geral da opção **Analog Deadband (% of previous value)**. Por exemplo, o valor "MEs:9833DB:0.7" indica o uso de uma banda morta de 0.7%.

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	DESCRIÇÃO	MNEMÔNICO	ASDU SEM TIMESTAMP	ASDU COM TIMESTAMP DE 56 BITS
SC	**Comando de Ponto Digital Simples**	SC	**C_SC_NA_1** (45)	**C_SC_TA_1** (58)
DC	**Comando de Ponto Digital Duplo**	DC	**C_DC_NA_1** (46)	**C_DC_TA_1** (59)
RC	**Comando de Regulação de Passo**	RC	**C_RC_NA_1** (47)	**C_RC_TA_1** (60)
SE	**Comando de Setpoint**	**SEn**, **SEs** ou **SEf**	**C_SE_NA_1** (48), **C_SE_NB_1** (49) e **C_SE_NC_1** (50)	**C_SE_TA_1** (61), **C_SE_TB_1** (62) e **C_SE_TC_1** (63)
BO	**Comando de 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 de um Comando:
  - **0**: Sem definição
  - **1**: Pulso de Curta Duração (*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)

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

Ponto Digital Simples (SC)

O **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.

Ponto Digital Duplo (DC)

O **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 **IEC 870-5-104**, o valor trafegado nestas ASDUs é um DCO (*Double Command*) que contém os campos **SE** e **QU**, além de 2 (dois) bits de valor, chamados de DCS (*Double Command State*):

- **0**: Estado intermediário (inválido)
- **1**: OFF

- **2**: ON
- **3**: Estado intermediário (inválido)

O valor deste Comando depende da configuração realizado na janela de propriedades deste Driver, na opção **Convert DCO to Boolean** do grupo **Boolean Values** na aba **Properties**. Se esta opção está desmarcada, o valor escrito é o próprio valor do DCS, descrito anteriormente.Se a opção de tratar como Booleano está selecionada, o valor deste Comando é **0**: OFF (DPI igual a um) ou **1**: ON (DPI igual a dois).

Regulação de Passo (RC)

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 **IEC 870-5-104**, o valor trafegado nestas ASDUs é um RCO (*Regulating Step Command*), que contém 1 (um) bit **SE**, indicando se é um Comando de Seleção ou de Execução, 5 (cinco) bits **QU**, indicando o qualificador de um Comando, como por exemplo Pulso de Curta Duração, Pulso de Longa Duração ou Saída Persistente, e 2 (dois) bits de valor, chamados de RCS (*Regulating Step Command State*):

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

O valor deste Comando depende da configuração realizada na janela de propriedades deste Driver, na opção **Convert RCO to Boolean** do grupo **Boolean Values** na aba **Properties**. Se esta opção está desmarcada, o valor escrito é o próprio valor do RCS, descrito anteriormente. Se a opção de tratar como 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).

Setpoint ou Analógico (SE)

Os **Comandos de Setpoint**, ou escritas em Medidas Analógicas, podem ser efetuados usando os valores descritos nos tópicos a seguir.

Setpoint com Valor Normalizado (SEn)

Um **Comando de Setpoint** permite o envio de um valor normalizado no intervalo entre -1.0 e 1.0 para este Driver no modo **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 ajustados no limite mais próximo. A precisão do valor é de 16 bits, portanto nem todos os valores dentro deste intervalo podem ser representados com precisão, ou seja, podem ocorrer aproximações nos valores enviados e recebidos.

Setpoint com Valor com Escala (SEs)

Um **Comando de Setpoint** permite o envio de um valor com escala, no intervalo entre -3276.8 e 3276.7 para este Driver no modo **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.

Setpoint com Valor em Ponto Flutuante (SEf)

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**.

Bitstring (BO)

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** apenas no modo **Mestre**, os parâmetros **[N]** de um Tag devem estar zerados. Deve-se colocar o endereço de um equipamento na propriedade **ParamDevice** e a descrição do Comando na propriedade **ParamItem**, de acordo com a sintaxe a seguir.

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

Em que:

- **<comando>** é o 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
Ponto Digital Simples	**C_SC_NA_1** (45) e **C_SC_TA_1** (58)	SC ou SCt	SC:1.Select ou SCt:1.Select
Ponto Digital Duplo	**C_DC_NA_1** (46) e **C_DC_TA_1** (59)	DC ou DCt	DC:328.Execute ou DCt:328.Execute
Regulação de Passo	**C_RC_NA_1** (47) e **C_RC_TA_1** (60)	RC ou RCt	RC:10.434.Execute ou RCt:10.434.Execute
Setpoint com Valor Normalizado	**C_SE_NA_1** (48) e **C_SE_TA_1** (61)	SEn ou SEnt	SEn:10938.Execute(ShortPulse) ou SEnt:10938.Execute(ShortPulse)
Setpoint com Valor com Escala	**C_SE_NB_1** (49) e **C_SE_TB_1** (62)	SEs ou SEst	SEs:9833.Execute ou SEst:9833.Execute
Setpoint com Valor em Ponto Flutuante	**C_SE_NC_1** (50) e **C_SE_TC_1** (63)	SEf ou SEft	SEf:10.35.22.Execute ou SEft:10.35.22.Execute
Bitstring	**C_BO_NA_1** (51) e **C_BO_TA_1** (64)	CBO ou CBOt	CBO:99.Select ou 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 do 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 o 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:
  - **(LongPulse)**: Envia um Comando de Pulso de Longa Duração
  - **(ShortPulse)**: Envia um Comando de Pulso de Curta Duração
  - **(PersistentOutput)**: Envia um Comando de Saída Persistente
  - **omitido**: Envia um Comando normal

Protocolo IEC 870-5-104 em Modo Mestre

No modo **Mestre**, este Driver suporta as funções descritas nos tópicos a seguir.

NOTA

A porta TCP/IP padrão para o protocolo **IEC 870-5-104** é 2404.

Leitura de Dados de Monitoramento

O protocolo **IEC 870-5-104** no modo **Mestre** permite o recebimento de dados de monitoramento enviados por um **Escravo** de forma espontânea ou como resposta a algum Comando enviado por este **Mestre**, como por exemplo um Comando de Interrogação Geral ou de Interrogação de Contadores. Os dados enviados por um equipamento são armazenados em uma memória interna deste Driver, chamada de **Cache**, e estão 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 a eventos mais novos.

A forma de ler dados da memória *cache* está descrita no tópico a seguir.

Leitura de Eventos da Memória Cache

Protocolos	**IEC 870-5-104 MASTER**
Parâmetros Numéricos	**N1**: Número de uma Estação, **N2**: 4, 5, 6 (quatro, cinco, seis, leitura normal) ou 7 (sete, leitura com *reset* automático), **N3**: ASDU e **N4**: Endereço de um Objeto
Parâmetros por String	Consulte o tópico **Configuração de Tags por String**
Escrita	Não disponível
Leitura	Retorna o valor, a qualidade e a estampa de tempo dos dados recebidos
ASDUs	• **Definidas no protocolo IEC 870-5-101**: **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 IEC 870-5-104**: **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)

Este Tag retorna 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:

- **N2 igual a 4, 5 ou 6**: Leitura de eventos da memória *cache*, apenas retorna os valores que este Driver recebeu de um **Escravo**
- **N2 igual a 7**: Leitura de eventos da memória *cache* com *reset* automático. Para os tipos de dados de Ponto Digital Simples e Ponto Digital Duplo, para cada valor **ON** recebido, um evento **OFF** é gerado um milissegundo depois do evento **ON**

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

NOTA

Recomenda-se que o Comando de Interrogação Geral seja habilitado para forçar uma atualização periódica de todos os valores enviados por um **Escravo**.

Sincronização de Relógio

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

Sincronização Automática

Para configurar a Sincronização Automática de Relógio, configure a propriedade **Clock Sync Auto-run** no grupo **Master - Clock Sync** da aba **Properties** da janela de configuração deste Driver. 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 pelas opções **Clock Sync Repeat** e **Clock Sync Repeat Period (sec)** no mesmo grupo.

Sincronização Manual

Para efetuar uma Sincronização Manual de Relógio, escreva um valor qualquer no **Tag de Envio de Sincronização de Relógio**.

Envio de Sincronização de Relógio

Protocolos	**IEC 870-5-104 MASTER**
Parâmetros Numéricos	**N1**: Número de uma Estação, **N2**: 2 (dois), **N3**: 0 (zero) e **N4**: 0 (zero)
Parâmetros por String	Não disponível
Escrita	Escreva qualquer valor neste Tag para efetuar a Sincronização de Relógio de um **Escravo**
Leitura	Não suportado
ASDUs	**C_CS_NA_1** (103)

Qualquer valor escrito neste Tag envia um Comando de Sincronização de Relógio para um **Escravo**. O relógio deste **Escravo** é sincronizado com o horário atual de um **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).

NOTA

Se tanto o Comando de Sincronização de Relógio quanto o Comando de Interrogação Geral estão habilitados, o Comando de Sincronização de Relógio é sempre executado em primeiro lugar.

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 a um **Escravo**, configure as opções a seguir no grupo **Master - General Interrogation** na aba **Properties** da janela de configurações deste Driver:

- **Interrogation Auto-run**: Selecione esta opção para habilitar o envio de um Comando de Interrogação Geral sempre que se estabelece a comunicação com um **Escravo**
- **Interrogation Repeat e Interrogation Repeat Period (sec)**: Selecione esta opção e configure o intervalo, em segundos, para repetir um Comando de Interrogação Geral
- **Interrogation Wait ACTTERM**: Selecione esta opção para forçar este Driver a aguardar o envio da notificação de término de um Comando de Interrogação Geral (**ACTTERM**). Recomenda-se que esta opção só seja desmarcada caso se detecte que um **Escravo** não envia esta notificação, pois este envio é opcional
- **Interrogation Process At end**: Selecione esta opção para **processar os dados recebidos apenas no final de um Comando de Interrogação Geral**

Envio de uma Interrogação Geral

Protocolos	**IEC 870-5-104 MASTER**
Parâmetros Numéricos	**N1**: Número de uma Estação, **N2**: 100, **N3**: 0 (zero) e **N4**: 0 (zero)
Parâmetros por String	Não disponível
Escrita	Escreva qualquer valor neste Tag para enviar a um **Escravo** um Comando de Interrogação do grupo de contadores
Leitura	Não suportado
ASDUs	**C_IC_NA_1** (100)

Este Driver no modo **Mestre** já realiza um Comando de Interrogação Geral (**GI**) toda vez que uma Estação no modo **Escravo** ativa a comunicação, ou quando o intervalo de **GI** cíclico (*Interrogation Repeat*) expira. Para solicitar um Comando **GI** a qualquer momento, escreva um valor qualquer neste Tag. Qualquer valor escrito neste Tag envia um Comando de Interrogação Geral a um **Escravo**. Se este Comando é bem-sucedido, a escrita retorna o valor 1 (um). Se este Comando falha por qualquer motivo, a escrita retorna o valor 0 (zero). A leitura deste Tag não é suportada.

Envio de uma Interrogação de Grupos

Protocolos	**IEC 870-5-104 MASTER**
Parâmetros Numéricos	**N1**: Número de uma Estação, **N2**: 101 a 116, **N3**: 0 (zero) e **N4**: 0 (zero)
Parâmetros por String	Não disponível
Escrita	Não suportado
Leitura	Os valores possíveis são **1**: Indica que um Comando de Interrogação teve sucesso ou **0**: Houve falha
ASDUs	**C_IC_NA_1** (100)

Habilite a leitura deste Tag para solicitar um Comando de Interrogação de Grupo, indicado pelo parâmetro *N2* igual a 100 mais o índice do grupo, de 1 (um) a 16. 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).

Processando Dados ao Final de uma Interrogação Geral

Um **Escravo** tem a liberdade de enviar dados espontâneos durante um Comando de Interrogação Geral. Entretanto, este **Escravo** tem que garantir sempre que o último valor enviado é o correto, ou mais atual. Por exemplo, quando um **Mestre** inicia um Comando de Interrogação Geral, um **Escravo** cria um instantâneo de todos os dados que são enviados e inicia o processo de envio, que pode durar alguns segundos. Durante o processo de envio, se algum dos dados sofrer uma alteração de estado, esta alteração pode ser enviada de forma espontânea:

- Caso um **Escravo** já tenha enviado os dados em um Comando de Interrogação Geral, pode enviar a alteração espontânea sem problemas
- Caso não tenha ainda enviado os dados, pode:
  - Aguardar para enviar a alteração espontânea no final de um Comando de Interrogação Geral

- Enviar a alteração espontânea imediatamente e depois enviar a mesma informação, já com o valor atualizado, como parte de um Comando de Interrogação Geral

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

Para contornar este problema em um **Escravo**, pode-se habilitar a opção **Interrogation Process At End**. Quando esta opção está selecionada, os dados de interrogação, ou seja, causa de transmissão igual a 20, recebidos durante um Comando de Interrogação Geral são empilhados por este Driver até o final do processo de interrogação. Neste momento, este Driver exclui dos dados empilhados quaisquer pontos que tenham recebido valores espontâneos durante esta interrogação. Se esta opção está desmarcada, os dados de interrogação recebidos durante o processo são processados imediatamente. Como esta opção não causa problemas colaterais, exceto uma demora a mais para processar os dados de interrogação, pode-se mantê-la sempre habilitada.

NOTA

Os dados espontâneos recebidos durante um Comando de 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. A forma de funcionamento dos Contadores é **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 ou **Com Reinício**: Um **Escravo** memoriza, ou congela, os Contadores de tempos em tempos e reinicia o valor destes Contadores para 0 (zero). O valor congelado é então enviado para um **Mestre**.

Normalmente o momento do congelamento é controlado por um **Escravo**. Opcionalmente, um **Mestre** pode assumir este papel, enviando **Comandos de Interrogação de Contadores** ou **ASDU 101**, que 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 diferentes períodos de integração para cada grupo de Contadores. Esta configuração, ou seja, quais Contadores pertencem a quais grupos, é responsabilidade dos equipamentos.

Envio de Interrogação de Contadores

Protocolos	**IEC 870-5-104 MASTER**
Parâmetros Numéricos	**N1**: Número de uma Estação, **N2**: 3 (três), **N3**: Grupo de Contadores e **N4**: Operação
Parâmetros por String	Não disponível
Escrita	Escreva qualquer valor neste Tag para enviar a um **Escravo** um Comando de Interrogação do Grupo de Contadores
Leitura	Não suportado
ASDUs	**C_CI_NA_1** (101)

Qualquer valor escrito neste Tag envia um Comando de Interrogação de Contadores a um **Escravo**. Se este Comando é bem-sucedido, a escrita retorna o valor 1 (um). Se este Comando falha por qualquer motivo, a escrita retorna o valor 0 (zero). O parâmetro *N3* define o número do Grupo de Contadores afetado. Configure este parâmetro com o valor 0 (zero) para não especificar nenhum grupo, o que pode ter significado especial, dependendo de cada equipamento. O parâmetro *N4* define a operação a efetuar:

- **0**: Interrogação apenas, sem congelamento ou reinício
- **1**: Congelamento de Contadores sem reinício
- **2**: Congelamento de Contadores com reinício
- **3**: Reinício de Contadores

Em resposta a um Comando de Interrogação, um **Escravo** envia os dados memorizados dos Contadores do grupo especificado neste 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 um Comando de Interrogação podem ser lidos por uma aplicação **Mestre** utilizando Tags de leitura da memória *cache*, ou seja, o parâmetro *N2* igual a 5 (cinco), o parâmetro *N3* com o número de qualquer uma das ASDUs, ou seja, 15, 16 ou 37, e o parâmetro *N4* com o endereço de um Contador. Ao utilizar a configuração por **Strings**, configure na propriedade **ParamItem** o valor "ITxxx", em que *xxx* é o endereço de um Contador.

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 têm êxito, então um **Mestre** pode enviar os **Comandos de Execução** quando o 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.

Envio de Seleção

Protocolos	**IEC 870-5-104 MASTER**
Parâmetros Numéricos	**N1**: Número de uma Estação, **N2**: 10 (Normal), 20 (Pulso de Curta Duração), 30 (Pulso de Longa Duração) ou 40 (Saída Persistente), **N3**: Tipo de objeto (45-51, 58-64) e **N4**: Endereço de um Comando
Parâmetros por String	Consulte o tópico **Configuração de Comandos por String**
Escrita	Escreva um valor neste Tag para enviar um Comando de Seleção (*Select*) a um **Escravo**
Leitura	Não suportado
ASDUs	• **Definidos no protocolo IEC 870-5-101**: **C_SC_NA_1** (45), **C_DC_NA_1** (46), **C_RC_NA_1** (47), **C_SE_NA_1** (48), **C_SE_NB_1** (49), **C_SE_NC_1** (50) e **C_BO_NA_1** (51) • **Definidos no protocolo IEC 870-5-104**: **C_SC_TA_1** (58), **C_DC_TA_1** (59), **C_RC_TA_1** (60), **C_SE_TA_1** (61), **C_SE_TB_1** (62), **C_SE_TC_1** (63) e **C_BO_TA_1** (64)

Escreva um valor neste Tag para enviar um Comando de Seleção a um **Escravo**. Se este Comando é bem-sucedido, posteriormente é possível enviar um Comando de Execução ou então um Comando de Desativação de Seleção. O parâmetro *N2* define o tipo de Comando a ser enviado:

- **10**: Comando Normal
- **20**: Pulso de Curta Duração
- **30**: Pulso de Longa Duração
- **40**: Saída Persistente

O parâmetro *N3* define o tipo de objeto que está sendo comandado. Os tipos de objetos a seguir são enviados sem estampa de tempo, definidos no protocolo **IEC 870-5-101**:

- **45**: Comando de **Ponto Digital Simples**
- **46**: Comando de **Ponto Digital Duplo**
- **47**: Comando de **Posição 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, definidos no protocolo **IEC 870-5-104**:

- **58**: Comando de **Ponto Digital Simples com Estampa de Tempo**
- **59**: Comando de **Ponto Digital Duplo com Estampa de Tempo**

- **60**: Comando de **Posição 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*. Se este Comando foi enviado com sucesso, a escrita retorna 1 (um). Se há qualquer falha no envio deste Comando, a escrita retorna 0 (zero).

NOTAS

- Os Comandos de **Bitstring**, 51 e 64, não suportam o envio dos qualificadores de Pulso Curto, Pulso Longo e Saída Persistente. Estes Comandos, nestes casos, são enviados para um **Escravo** sem a especificação de um qualificador. Se um **Escravo** é de um outro Driver IEC870, este recebe estes Comandos com um qualificador de Comando Normal.
- Os Comandos de **Bitstring**, 51 e 64, não possuem informação de Seleção ou Execução. Ambos os Comandos são aceitos por este Driver, mas o Comando enviado para um **Escravo** não indica se é um Comando de Seleção ou de Execução. Se um **Escravo** é de um outro Driver IEC870, este recebe estes Comandos sempre como um Comando de Execução.

Envio de Execução

Protocolos	**IEC 870-5-104 MASTER**
Parâmetros Numéricos	**N1**: Número de uma Estação, **N2**: 11 (Normal), 21 (Pulso de Curta Duração), 31 (Pulso de Longa Duração) ou 41 (Saída Persistente), **N3**: Tipo de objeto (45-51, 58-64) e **N4**: Endereço de um Comando
Parâmetros por String	Consulte o tópico **Configuração de Comandos por String**
Escrita	Escreva um valor neste Tag para enviar um Comando de Execução (*Execute*) a um **Escravo**
Leitura	Não suportado
ASDUs	• **Definidos no protocolo IEC 870-5-101**: **C_SC_NA_1** (45), **C_DC_NA_1** (46), **C_RC_NA_1** (47), **C_SE_NA_1** (48), **C_SE_NB_1** (49), **C_SE_NC_1** (50) e **C_BO_NA_1** (51) • **Definidos no protocolo IEC 870-5-104**: **C_SC_TA_1** (58), **C_DC_TA_1** (59), **C_RC_TA_1** (60), **C_SE_TA_1** (61), **C_SE_TB_1** (62), **C_SE_TC_1** (63) e **C_BO_TA_1** (64)

Escreva um valor neste Tag para enviar um Comando de Execução a um **Escravo**. Se este Comando é bem-sucedido, a escrita retorna 1 (um) e no caso de falha retorna 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 *N2* define o tipo de Comando a ser enviado:

- **11**: Comando Normal
- **21**: Pulso de Curta Duração

- **31**: Pulso de Longa Duração
- **41**: Saída Persistente

O parâmetro *N3* define o tipo de objeto que está sendo comandado. Os tipos de objetos a seguir são enviados sem estampa de tempo, definidos no protocolo **IEC 870-5-101**:

- **45**: Comando de **Ponto Digital Simples**
- **46**: Comando de **Ponto Digital Duplo**
- **47**: Comando de **Posição 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, definidos no protocolo **IEC 870-5-104**:

- **58**: Comando de **Ponto Digital Simples com Estampa de Tempo**
- **59**: Comando de **Ponto Digital Duplo com Estampa de Tempo**
- **60**: Comando de **Posição 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*. É permitido enviar no protocolo **IEC 870-5-101** os Comandos definidos no protocolo **IEC 870-5-104** e vice-versa, desde que um **Escravo** suporte estes Comandos.

Envio de Cancelamento de Seleção

Protocolos	**IEC 870-5-104 MASTER**
Parâmetros Numéricos	**N1**: Número de uma Estação, **N2**: 12 (Normal), 22 (Pulso de Curta Duração), 32 (Pulso de Longa Duração) ou 42 (Saída Persistente), **N3**: Tipo de objeto (45-51, 58-64) e **N4**: Endereço de um Comando
Parâmetros por String	Consulte o tópico **Configuração de Comandos por String**
Escrita	Escreva um valor neste Tag para cancelar um Comando de Seleção (*Select*) enviado anteriormente a um **Escravo**
Leitura	Não suportado
ASDUs	• **Definidos no protocolo IEC 870-5-101**: **C_SC_NA_1** (45), **C_DC_NA_1** (46), **C_RC_NA_1** (47), **C_SE_NA_1** (48), **C_SE_NB_1** (49), **C_SE_NC_1** (50) e **C_BO_NA_1** (51) • **Definidos no protocolo IEC 870-5-104**: **C_SC_TA_1** (58), **C_DC_TA_1** (59), **C_RC_TA_1** (60), **C_SE_TA_1** (61), **C_SE_TB_1** (62), **C_SE_TC_1** (63) e **C_BO_TA_1** (64)

Escreva um valor neste Tag para enviar um Comando de Cancelamento de uma Seleção enviado anteriormente a um **Escravo**. Se este Comando é bem-sucedido, a escrita retorna 1 (um) e se falha por qualquer motivo, a escrita retorna 0 (zero). O parâmetro *N2* define o tipo de Comando a ser cancelado:

- **12**: Comando Normal
- **22**: Pulso de Curta Duração
- **32**: Pulso de Longa Duração
- **42**: Saída Persistente

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, definidos no protocolo **IEC 870-5-101**:

- **45**: Comando de **Ponto Digital Simples**
- **46**: Comando de **Ponto Digital Duplo**
- **47**: Comando de **Posição 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, definidos no protocolo **IEC 870-5-104**:

- **58**: Comando de **Ponto Digital Simples com Estampa de Tempo**
- **59**: Comando de **Ponto Digital Duplo com Estampa de Tempo**

- **60**: Comando de **Posição 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*. É permitido enviar no protocolo **IEC 870-5-101** os Comandos definidos no protocolo **IEC 870-5-104** e vice-versa, desde que um **Escravo** suporte estes Comandos.

Reinício de Processo

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

Envio de Reinício de Processo

Protocolos	**IEC 870-5-104 MASTER**
Parâmetros Numéricos	**N1**: Número de uma Estação, **N2**: 1105, **N3**: 0 (zero) e **N4**: 0 (zero)
Parâmetros por String	Não disponível
Escrita	Escreva um valor neste Tag para solicitar o reinício do processo de um **Escravo**
Leitura	Não suportado
ASDUs	**C_RP_NA_1** (105)

O valor escrito neste Tag é o **Qualificador de Comando de Reinício de Processo**, um inteiro entre 0 (zero) e 255:

- **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, a escrita retorna 1 (um) e se falha por qualquer motivo, retorna 0 (zero).

Teste

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

Este 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, habilite as opções do grupo **Test Command** na opção **Master** da aba **Properties** na janela de propriedades deste Driver.

No protocolo **IEC 870-5-104**, a ASDU enviada é a **C_TS_TA_1** (107). Se este Comando falha, a conexão é fechada e reaberta.

Protocolo IEC 870-5-104 em Modo Escravo

No modo **Escravo**, este Driver suporta as funções descritas nos tópicos a seguir.

NOTA

A porta TCP/IP padrão para o protocolo **IEC 870-5-104** é 2404.

Shutdown

Protocolos	Somente para **IEC 870-5-104 SLAVE**
Parâmetros Numéricos	**N1**: 0, **N2**: 990, **N3**: 0 (zero) e **N4**: 0 (zero)
Parâmetros por String	Não disponível
Escrita	Ao escrever o valor 1 (um), isto indica que uma aplicação que tem a origem dos dados que estão sendo publicados está sendo desligada, iniciando o modo de *shutdown* do canal **Escravo** 104, em que as escritas recebidas nos Tags com valor **NULL** são temporariamente bloqueadas durante o intervalo especificado na propriedade **Cache Startup/Shutdown Mode Time (s)**. Se após este intervalo este Driver ainda está em execução, os Tags que continuam com o último valor em **NULL** são notificados com qualidade ruim para um Driver **Mestre**
Leitura	Não disponível
ASDUs	Não disponível

Envio de Dados de Monitoramento

O **Envio de Dados de Monitoramento**, digitais e analógicos, entre outros, para um **Mestre** pode ocorrer como um Comando de Interrogação Geral, como um Envio Espontâneo com prioridade alta ou como um Envio Espontâneo com prioridade baixa.

O **Comando de Interrogação Geral** é um Comando iniciado por um **Mestre** de tempos em tempos. Em resposta a este 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 a necessidade de um **Mestre** enviar um Comando para isto.

Uma aplicação é responsável por alimentar os valores de um **Escravo** através de escritas em Tags. É comum que, durante a inicialização de uma aplicação, sejam realizadas várias escritas em um **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**.

Escrita de Eventos na Memória Cache

Protocolos	**IEC 870-5-104 SLAVE**
Parâmetros Numéricos	**N1**: Número de uma Estação, **N2**: 4 (quatro), 5 (cinco), 6 (seis) ou 7 (sete), **N3**: ASDU e **N4**: Endereço de um Objeto
Parâmetros por String	Consulte o tópico **Configuração de Tags por String**
Escrita	Carrega na memória *cache* interna deste Driver, ou seja, nas filas de envio, o valor de um ponto
Leitura	Retorna o valor atual da memória *cache* interna deste Driver
ASDUs	• **Definidas no protocolo IEC 870-5-101**: **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 IEC 870-5-104**: **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)

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

- **N2 igual a 4 (quatro) ou N2 igual a 7 (sete)**: Os valores escritos são reportados apenas no Comando de Interrogação Geral
- **N2 igual a 5 (cinco)**: Os valores escritos são enviados imediatamente para um **Mestre** através de um Envio Espontâneo de Classe 2 (dois, prioridade baixa) e também podem ser reportados no Comando de Interrogação Geral
- **N2 igual a 6 (seis)**: Os valores escritos são enviados imediatamente para um **Mestre** através de um Envio Espontâneo de Classe 1 (um, prioridade alta) e também podem ser reportados no Comando de Interrogação Geral

Este Tag retorna eventos armazenados na memória *cache* interna deste Driver. 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 causa nenhum tipo de comunicação. Este Tag não suporta leitura ou escrita em bloco.

No Envio Espontâneo de dados, de Classe 1 (um) ou Classe 2 (dois), a ASDU utilizada para envio dos dados é exatamente a especificada no parâmetro *N3*. Por exemplo, ao utilizar o parâmetro *N3* igual a 4 (quatro), é enviada para um **Escravo** uma ASDU **M_DP_TA_1** (Ponto Duplo com Estampa de Tempo de 24 bits).

A ASDU utilizada para envio dos dados no Comando de Interrogação Geral pode ser diferente da configurada no parâmetro *N3* caso a opção **Remove timestamp from GI data** esteja habilitada. Neste caso, um dado carregado neste Driver com o parâmetro *N3* igual a 4 (quatro) pode ser enviado no Comando de Interrogação Geral com a ASDU **M_DP_NA_1** (Ponto Duplo sem Estampa de Tempo).

Iniciando este Driver com o Nível de Enlace Desabilitado

Os valores iniciais servidos por um **Escravo** precisam ser escritos por uma aplicação durante a inicialização. Para evitar que um **Mestre** conecte a este **Escravo** antes que o valor inicial de todos os Tags tenha sido carregado, inicie este Driver com o nível de enlace desabilitado:

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 esta 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**.

Habilitação do Nível de Enlace

Protocolos	**IEC 870-5-104 SLAVE**
Parâmetros Numéricos	**N1**: 0 (zero), **N2**: 996, **N3**: 0 (zero) e **N4**: 0 (zero)
Parâmetros por String	Não disponível
Escrita	O valor 0 (zero) desabilita o Nível de Enlace, evitando que este Driver estabeleça conexões com um **Mestre**. O valor 1 (um) habilita o Nível de Enlace
Leitura	Retorna 1 (um) se o Nível de Enlace está habilitado e 0 (zero) se o Nível de Enlace está desabilitado
ASDUs	Nenhuma

Este Tag de leitura e escrita permite ligar e desligar o Nível de Enlace deste Driver. Com o Nível de Enlace desabilitado, este Driver aceita conexões de um **Mestre** mas fecha a conexão assim que recebe qualquer mensagem.

Envio Cíclico de Dados Analógicos

Este Driver no modo **Escravo** pode ser programado para enviar 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 **Cyclic**, configure as seguintes opções:

- **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). Deve seguir a mesma sintaxe do intervalo de dados do Comando de Interrogação Geral. Deixe este campo vazio para não enviar o conjunto 1 (um). Por exemplo, o intervalo "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). Deve seguir a mesma sintaxe do intervalo de dados do Comando de 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 um Tag seja enviado imediatamente com prioridade alta. Os dados cíclicos são sempre enviados no intervalo programado, mesmo quando um valor não é alterado.

Forçação Local

Uma aplicação pode, a qualquer momento, forçar um valor específico para cada Tag. Independente de outros valores escritos posteriormente no mesmo Tag, este Driver no modo **Escravo** reporta para um **Mestre** sempre o valor forçado, até o momento em que esta aplicação limpa a forçação.

Ativação da Forçação

Para ativar a forçação local em um Tag, escreva um *array* de 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 no tag001
tag001.WriteEx Array("FORCE", 1)
```

Este valor forçado é enviado para um **Mestre** utilizando o mesmo método dos valores normais de um Tag, ou seja, apenas no Comando de Interrogação Geral, com prioridade alta ou com prioridade baixa, com o *flag* **SB** ligado e com a estampa de tempo atual do computador.

Enquanto um Tag está com um valor forçado, este continua aceitando escritas de outros valores não forçados. Neste caso, este Driver armazena o último valor enviado para um Tag e, quando a forçação é cancelada, este último valor é enviado para um **Mestre**. Pode-se sobrescrever um valor forçado com outro valor forçado.

Desativação da Forçação

Para limpar um valor forçado, escreva um *array* com 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 que está forçado no tag001
tag001.WriteEx Array("CLEAR", 0)
```

Quando a forçação é cancelada, 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 alterar o 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, configure a opção **Handle mode for Clock sync command** no grupo **Slave** da aba **Properties** na janela de configurações deste Driver da seguinte forma:

- **disabled**: Este Driver rejeita os Comandos de Sincronização de Relógio recebidos, respondendo-os 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 da aplicação
- **handle automatically**: Este Driver ajusta o horário local do computador de acordo com o horário recebido de um **Mestre**. Se este ajuste falha, retorna um erro para este **Mestre** (Causa de Transmissão 71, confirmação negativa de ativação). Caso contrário, responde com uma Causa de Transmissão 7 (sete, confirmação de ativação) indicando o sucesso da 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.

Tratamento da Sincronização de Relógio

Protocolos	**IEC 870-5-104 SLAVE**
Parâmetros Numéricos	**N1**: Número de uma Estação, **N2**: 4 (quatro, leitura de dados da memória *cache*), **N3**: 103 e **N4**: 0 (zero)
Parâmetros por String	Não suportado
Escrita	Não suportado
Leitura	O valor deste Tag contém o novo horário enviado por um **Mestre**
ASDUs	**C_CS_NA_1** (103)

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, configurado no item **pass to the application** da opção **Clock sync command**, os Comandos de Sincronização de Relógio, então o evento **OnRead** deste Tag deve atualizar o relógio do computador.

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 na janela de propriedades deste Driver, na opção **General Interrogation** do grupo **Slave** da aba **Properties**:

- **Interrogation Enable Range**: Habilita o envio de uma lista pré-definida de dados
- **List of Data Objects returned by General Interrogation**: Selecione esta opção para forçar este Driver a sempre retornar uma lista de dados pré-definida em resposta a um Comando de Interrogação Geral. Se esta opção está desmarcada, este Driver retorna todos os valores armazenados na memória *cache* interna

A lista de pontos é composta de um ou mais objetos com a sintaxe a seguir.

```
<tipo>:<listaDeEndereços>;<tipo>:<listaDeEndereços>;<tipo>:<listaDeEndereços>
```

Em que:

- **<tipo>**: Corresponde ao nome de uma ASDU
  - **SP (M_SP_NA_1, M_SP_TA_1, M_SP_TB_1)**: **Ponto Digital Simples**
  - **DP (M_DP_NA_1, M_DP_TA_1, M_DP_TB_1)**: **Ponto Digital Duplo**
  - **ST (M_ST_NA_1, M_ST_TA_1, M_ST_TB_1)**: **Posição de Passo**
  - **BO (M_BO_NA_1, M_BO_TA_1, M_BO_TB_1)**: **Bitstring**
  - **MEn (M_ME_NA_1, M_ME_TA_1, M_ME_TD_1)**: **Medida Analógica**
  - **MEs (M_ME_NB_1, M_ME_TB_1, M_ME_TE_1)**: **Medida Analógica**

- **MEf (M_ME_NC_1, M_ME_TC_1, M_ME_TF_1)**: **Medida Analógica**
- **IT (M_IT_NA_1, M_IT_TA_1, M_IT_TB_1)**: **Total Integrado**
- **EPa (M_EP_TA_1, M_EP_TD_1)**: *Event of Protection Equipment*
- **EPb (M_EP_TB_1, M_EP_TE_1)**: *Packed Start Events of Protection Equipment*
- **EPc (M_EP_TC_1, M_EP_TF_1)**: *Packed Output Circuit Information of Protection Equipment*

- **<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 de endereço 100 a 119 e os pontos duplos de endereços 200, 205 e 210 a 219, use a sintaxe a seguir.

```
SP:100-119;DP:200,205,210-219
```

- **Remove timestamp from data sent on General Interrogation**: Se esta opção está selecionada, os dados retornados por um Comando de 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 tópico **Tag de Escrita de Eventos na Memória Cache**

Seleção ou Execução

Ao receber um **Comando de Seleção ou Execução**, este Driver pode ter os seguintes comportamentos:

1. **Negar o Comando (disabled)**: Este Driver responde a um **Mestre** com a Causa de Transmissão 135 (Confirmação de ativação negativa).
2. **Notificar a aplicação e aguardar que esta indique a resposta do 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 à esta aplicação a responsabilidade por gerar as respostas para um **Mestre**.
3. **Tratar o Comando automaticamente e notificar a aplicação (handle automatically)**: Este Driver responde a um **Mestre** como se um 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 janela de propriedades deste Driver, na opção **Command Handling** do grupo **Slave** na aba **Properties**.

Tratamento da Seleção ou Execução

Protocolos	**IEC 870-5-104 SLAVE**
Parâmetros Numéricos	**B1**: Número de uma Estação, **B2**: 4 (quatro, leitura de dados da memória *cache*), **B3**: ASDU (45 a 51, 58 a 64) e **B4**: Endereço de um Objeto
Parâmetros por String	Não disponível
Tamanho do Bloco	Quatro Elementos
Escrita	Envia para um **Mestre** a resposta do Comando de Seleção ou Execução
Leitura	Recebe os Comandos de Seleção ou Execução enviados por um **Mestre**
ASDUs	**C_SC_NA_1** (45), **C_DC_NA_1** (46), **C_RC_NA_1** (47), **C_SE_NA_1** (48), **C_SE_NB_1** (49), **C_SE_NC_1** (50), **C_BO_NA_1** (15), **C_SC_TA_1** (58), **C_DC_TA_1** (59), **C_RC_TA_1** (60), **C_SE_TA_1** (61), **C_SE_TB_1** (62), **C_SE_TC_1** (63) e **C_BO_TA_1** (64)

O parâmetro *B3* pode ser um dos 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)**: *Set point command, normalized value*
- **49 (C_SE_NB_1)**: *Set point command, scaled value*
- **50 (C_SE_NC_1)**: *Set point command, short floating point number*
- **51 (C_BO_NA_1)**: *Bitstream of 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)**: *Set point command, normalized value (with 7-octet timestamp)*
- **62 (C_SE_TB_1)**: *Set point command, scaled value (with 7-octet timestamp)*
- **63 (C_SE_TC_1)**: *Set point command, short floating point number (with 7-octet timestamp)*
- **64 (C_BO_TA_1)**: *Bitstream of 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, uma aplicação deve tratar a informação recebida, alterando os Elementos de Bloco necessários e escrevendo este Tag Bloco de volta para este Driver, de forma que este Driver possa enviar uma resposta a um **Mestre**. Os Elementos deste Tag Bloco são os seguintes:

- **0**: Valor de um Comando, dependendo do tipo de Comando

- **1**: Causa da Transmissão (**6**: Ativar ou **8**: Desativar)
- **2**: Seleção ou Execução (**0**: Execução ou **1**: Seleção)
- **3**: Qualidade de um Comando (**0**: Indefinido, **1**: Pulso de Curta Duração, **2**: Pulso de Longa Duração ou **3**: Saída Persistente)

Quando uma indicação de Comando de Ativação, ou seja, o Elemento 1 (um) igual a 6 (seis), é recebida, uma aplicação deve responder imediatamente com um dos seguintes valores:

- Uma Causa de Transmissão **ACTCON** positiva se este Comando pode prosseguir, ou seja, o Elemento 1 (um) igual a 7 (sete)
- Uma Causa de Transmissão **ACTCON** negativa se este Comando não pode prosseguir, ou seja, o Elemento 1 (um) igual a 135 (128 + 7)

Se uma Causa de Transmissão **ACTCON** positiva é enviada, uma aplicação deve então verificar o Elemento 2 (dois, Seleção ou Execução) e executar a ação apropriada.

Se um Comando de Execução é requisitado, uma aplicação deve executar este Comando e depois enviar uma Causa de Transmissão **ACTTERM**, ou seja, o Elemento 1 (um) igual a 10 (*activation termination*), para indicar que este Comando foi executado. O exemplo a seguir mostra um evento **OnRead** para este Tag Bloco no **Elipse E3**, no **Elipse Power** ou no **Elipse Water**.

```
' CMDIndication é um Tag Bloco de tamanho 4. 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, 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, provavelmente vai
        ' ser 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 um DEACTCON(9)
      WriteEx Array(CMD, 9, SE, QU), TimeStamp
  End Select
End Sub
```

NOTAS

- As escritas neste Tag Bloco falham se o modo de tratamento de Comandos de Seleção ou Execução não está configurado como **Pass to the Application**.
- Os Comandos de **Bitstring** (51 e 64) não suportam o envio dos qualificadores de Pulso de Curta Duração, Pulso de Longa Duração ou Saída Persistente. Nestes casos, o quarto Elemento deste Bloco (**QU**) tem sempre o valor 0 (zero), ou seja, o qualificador de um Comando Normal.
- Os Comandos de **Bitstring** (51 e 64) não possuem informação de Seleção ou de Execução. Nestes casos, o terceiro Elemento deste Bloco (**SE**) recebe sempre o valor 0 (zero, Execução).

Reinício de Processo

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

Tratamento de Reinício de Processo

Protocolos	**IEC 870-5-104 SLAVE**
Parâmetros Numéricos	**N1**: Número de uma Estação, **N2**: 4 (quatro, leitura de dados da memória *cache*), **N3**: 105 e **N4**: 0 (zero)
Parâmetros por String	Não suportado
Escrita	Não suportado
Leitura	O valor deste Tag contém o qualificador de reinício de processo enviado por um **Mestre**, ou seja, um inteiro entre 0 (zero) e 255
ASDUs	**C_RP_NA_1** (105)

Quando este Driver aceita um Comando de Reinício de Processo enviado por um **Mestre**, o qualificador enviado por este **Mestre** neste Comando é sinalizado em uma leitura deste Tag. Uma aplicação pode, ao receber um valor de leitura deste Tag, forçar via script um reinício deste Driver. Para consultar a lista de qualificadores de reinício de processo definidos no protocolo **IEC 870-5-104**, consulte o **Tag de Envio de Comando de Reinício de Processo**.

Teste

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

Artigos Técnicos

Esta seção descreve características técnicas diversas do Driver IEC 870-5-104 Mestre/Escravo.

Memória Cache Interna

A **Memória Cache Interna** é uma memória deste Driver que armazena tanto os dados recebidos de um equipamento quanto os dados que são 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**, 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 das filas, configure a opção **Offline Buffers** do grupo **Slave** na aba **Properties** da janela de configurações deste Driver conforme os valores a seguir:

- **Offline Class 1 buffering e Offline Class 1 Buffer size (events)**: Habilita manter *n* eventos de prioridade alta (Classe 1) enquanto um **Escravo** está desconectado de um **Mestre**
- **Offline Class 2 buffering e Offline Class 2 Buffer size (events)**: Habilita manter *n* eventos de prioridade baixa (Classe 2) enquanto um **Escravo** está desconectado de um **Mestre**

Estampas de Tempo

Este Driver, 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 operacional
- **Estampa de tempo de 24 bits**: Este formato é definido no protocolo **IEC 870-5-101**, mas pode ser usado tanto no protocolo **IEC 870-5-101** quanto no protocolo **IEC 870-5-104**. Inclui apenas as informações de hora, minuto, segundo e milissegundo. A data é preenchida por este Driver usando a data atual do sistema operacional
- **Estampa de tempo de 56 bits**: Este formato é definido no protocolo **IEC 870-5-104**, mas pode ser usado tanto no protocolo **IEC 870-5-104** quanto no protocolo **IEC 870-5-101**. 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, ou seja, com o *flag* **IV** igual a 1 (um), esta estampa de tempo é substituída pelo horário local do computador e a **qualidade do valor** retornada para uma aplicação é ajustada para indicar que a 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, ou seja, igual a 1 (um), indica que uma estampa de tempo está no horário de verão.

Para configurar o tratamento de horário de verão deste Driver, configure a opção **Timestamp SU Flag (Summertime adjustment)** do grupo **Timestamp** na aba **Properties** da 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 detectar estampas de tempo com o *flag* **SU** não consistente com o horário de verão do computador:
  - **Ao receber uma estampa de tempo com incoerência**: A mensagem é "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**: A mensagem é "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:
  - Ao receber uma estampa de tempo com o *flag* **SU** igual a 1 (um) e se o computador não está em horário de verão, atrasa a estampa de tempo em uma hora
  - Ao receber uma estampa de tempo com o *flag* **SU** igual a 1 (um) e se o computador está em horário de verão, não altera a estampa de tempo
  - Ao receber uma estampa de tempo com o *flag* **SU** igual a 0 (zero) e se o computador não está em horário de verão, não altera a estampa de tempo
  - Ao receber uma estampa de tempo com o *flag* **SU** igual a 0 (zero) e se o computador está em horário de verão, adianta a estampa de tempo em uma hora
  - Ao enviar uma estampa de tempo, se o computador está em horário de verão, o *flag* **SU** é igual a 1 (um)
  - Ao enviar uma estampa de tempo, se o computador não está em horário de verão, o *flag* **SU** é igual a 0 (zero)
  - Nos casos em que o computador está em horário de verão mas a configuração do horário de verão não é padrão, ou seja, o *bias* é diferente de 60 minutos:
    - Ao receber uma estampa de tempo com o *flag* **SU** igual a 1 (um), atrasa a estampa de tempo em uma hora, assumindo que em um equipamento o *bias* do horário de verão é sempre 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**: A mensagem é "Timestamp adjusted: From PLC (<PlcTimestamp> [<PlcBias> minutes]) to Driver (<LocalTimestamp> [<LocalBias> minutes])"
    - **Quando uma estampa de tempo é ajustada para envio**: A mensagem é "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

Este Driver 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 em 3 (três) campos, descritos na tabela a seguir.

Campos da qualidade OPC

STATUS		SUB-STATUS				LIMITE	
B7	**B6**	**B5**	**B4**	**B3**	**B2**	**B1**	**B0**
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 870-5-104** são mapeados na propriedade **Quality** de um Tag. Os bits mais acima na lista têm maior precedência. Por exemplo, se os bits **IV** e **NT** estão ligados, retorna o bit **IV**. O mapeamento dos bits é o seguinte:

- **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. Se a opção **Map Overflow bit to Uncertain Quality** está configurada, altera o **Status** para Incerto e o **Sub-Status** para Unidades de Engenharia excedidas
- **Timestamp IV (a ASDU veio sem timestamp, ou se o timestamp estava 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	QUALIDADE
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)
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)

IV	NT	SB	BL	OV	TIMESTAMP IV	QUALIDADE
			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 / Estampa de Tempo Local / Livre (11 0110 00)
			0 (zero)	1 (um)		**218**: Boa / Estampa de Tempo Local / Alto (11 0110 10)
			1 (um)	x		**219**: Boa / Estampa de Tempo Local / Constante (11 0110 11)
0 (zero)	0 (zero)	1 (um)	0 (zero)	0 (zero)	1 (um)	**220**: Boa / Sobrescrita e Estampa de Tempo Local / Livre (11 0111 00)
			0 (zero)	1 (um)		**222**: Boa / Sobrescrita e Estampa de Tempo Local / Alto (11 0111 10)
			1 (um)	x		**223**: Boa / Sobrescrita e Estampa de Tempo 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 *x* da tabela pelo valor 0 (zero). Por exemplo, a qualidade 4 (quatro) resulta no bit **IV** igual a 1 (um), no bit **NT** igual a 0 (zero), no bit **SB** igual a 0 (zero), no bit **BL** igual a 0 (zero), no bit **OV** igual a 0 (zero) e no bit **Timestamp IV** 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 as 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 contém o nome de uma propriedade e o respectivo valor

Causas de Transmissão

Uma **Causa de Transmissão** é uma informação que faz parte de uma ASDU enviada ou recebida no protocolo **IEC 870-5-104**, e indica o motivo pelo qual aquela ASDU está sendo enviada ou requisitada. Os valores mais comuns de Causas de Transmissão estão listados na tabela a seguir.

Valores de Causas de Transmissão no protocolo IEC 870-5-104

SIGNIFICADO	CAUSA DE TRANSMISSÃO	TRANSMISSÃO NEGATIVA (SOMA 64)	TRANSMISSÃO DE TESTE (SOMA 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
Informação de retorno causada por um comando remoto (RETREM)	11	75	139
Informação de retorno causada por um comando local (RETLOC)	12	76	140
Transferência de arquivo (FILE)	13	77	141

SIGNIFICADO	CAUSA DE TRANSMISSÃO	TRANSMISSÃO NEGATIVA (SOMA 64)	TRANSMISSÃO DE TESTE (SOMA 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 do Driver

A tabela a seguir lista todas as propriedades do Driver IEC 870-5-104 Mestre/Escravo. 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 IEC 870-5-104 Mestre/Escravo

PROPRIEDADE	VALOR PADRÃO	COMENTÁRIOS
Protocol	104	Configura o protocolo a ser utilizado por este Driver. Esta propriedade é gravada como "IEC.Protocol". Os valores possíveis são **104**: Protocolo **IEC 870-5-104 Mestre** ou **1005**: Protocolo **IEC 870-5-104 Escravo**
Link Layer - Fixed Station List	**String** vazia	Lista fixa de Estações. Gravada como "IEC.FixedStationList"
Link Layer - Default Slave Address	0 (zero)	Determina o identificador de Estação a ser utilizado quando o parâmetro *N1* ou *B1* é igual a 0 (zero). Gravada como "IEC.DefaultSlaveAddress"
Link Layer - Originator Address for outgoing ASDUs	0 (zero)	Valor do campo **Originator Address** em todas as ASDUs enviadas por este Driver. Este campo só é enviado se o

PROPRIEDADE	VALOR PADRÃO	COMENTÁRIOS
		número de octetos da Causa de Transmissão é igual a 2 (dois). Gravada como "IEC.OriginatorAddress"
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 zero e 255), **2**: Dois octetos (endereços de objeto entre zero e 65535) ou **3**: Três octetos (endereços de objeto entre zero e 16777215)
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 de "IEC.Octets.SlaveAddress", **1**: Um octeto (endereços entre zero e 255) ou **2**: Dois octetos (endereços entre zero 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. Gravada como "IEC.MaxPacketSize"
Master - Application Layer Timeout (sec)	4 (quatro)	Define o *time-out* de Comandos de aplicação, em segundos. É o tempo que este Driver no modo **Mestre** aguarda pela confirmação de término de diversos Comandos, como por exemplo Interrogação Geral, Selecionar ou Executar ou Sincronização de Relógio. Gravada como "IEC.AppTimeout"

PROPRIEDADE	VALOR PADRÃO	COMENTÁRIOS
Master - Wait EI (End Of Initialization)	0 (zero)	Indica se este Driver no modo **Mestre** deve aguardar o recebimento de uma ASDU 70 (**M_EI_NA_1**, Fim de inicialização) para considerar um **Escravo** 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 *time-out* de aplicação, o **Escravo** é desconectado)
Master - Don't wait ACTTERM for Execute commands	0 (zero)	Define se este Driver no modo **Mestre** aguarda a sinalização de término dos **Comandos de Execução** (*ACTivation TERMination*). Gravada como "IEC.CommandExecute.DontWaitACTTERM". Os valores possíveis são **0**: Aguarda ou **1 ou qualquer outro valor diferente de 0**: Não aguarda
*** Master - Map Overflow bit to Uncertain Quality**	0 (zero)	Define se a recepção do bit **OV** (*Overflow*) deste Driver no modo **Mestre** é mapeada para qualidade Incerta. Gravada como "IEC.MapOvAsUncertain". Os valores possíveis são **0**: Não mapeia o bit **OV** para Incerto ou **1**: Mapeia o bit **OV** para Incerto, Sub-Status **Eu_Exceeded**
Master - Discard ASDU 135 for Link Activity	0 (zero)	Descarta a ASDU 135 (privada) para considerar um *link* como ativo. Gravada como "IEC.DiscardAsdu135ForLinkActivity". Os valores possíveis são **0**: Não descarta ou **1**: Descarta
Master - General Interrogation - Interrogation Auto-run	0 (zero)	Habilita o Comando Automático de **Interrogação Geral** neste Driver no modo **Mestre**. Gravada como "IEC.Interrogate.AutoRun". Os valores possíveis são **0**: Desabilita ou **1 ou qualquer outro valor diferente de 0**: Habilita
Master - General Interrogation - Interrogation Repeat	0 (zero)	Habilita a repetição do Comando de **Interrogação Geral** neste Driver no modo **Mestre**. Gravada como "IEC.Interrogate.Repeat". Os valores possíveis são **0**: Desabilita e efetua o Comando de Interrogação Geral apenas uma vez ao conectar a um **Escravo** ou **1 ou qualquer outro valor diferente de 0**: Habilita a repetição de um Comando de Interrogação Geral
Master - General Interrogation - Interrogation Repeat Period (sec)	60	Define o intervalo de tempo, em segundos, entre cada repetição de um Comando de **Interrogação Geral**

PROPRIEDADE	VALOR PADRÃO	COMENTÁRIOS
		neste Driver no modo **Mestre**. Gravada como "IEC.Interrogate.PollTime"
Master - General Interrogation - Interrogation Wait ACTTERM	1 (um)	Determina se este Driver no modo **Mestre** aguarda ou não pelo recebimento da notificação de término de um Comando de **Interrogação Geral**. Gravada como "IEC.Interrogate.WaitACTTERM". Os valores possíveis são **0**: Não aguarda pela notificação de término (*ACTivation TERMination*) ou **1 ou qualquer outro valor diferente de 0**: Aguarda o recebimento da notificação de término de um Comando de Interrogação Geral
Master - General Interrogation - Interrogation Process At End	0 (zero)	Define o **momento em que os dados de um Comando de Interrogação Geral são processados** neste Driver no modo **Mestre**. Gravada como "IEC.Interrogate.ProcessAtEnd". Os valores possíveis são **0**: Processa durante um Comando de Interrogação Geral ou **1 ou qualquer outro valor diferente de 0**: Processa apenas no final de um Comando de Interrogação Geral
Master - General Interrogation - Disconnect if General Interrogation returns empty results	0 (zero)	Determina se este Driver no modo **Mestre** aceita ou não uma resposta vazia de um Comando de Interrogação Geral. Gravada como "IEC.Interrogate.DisconnectIfEmpty". Os valores possíveis são **0**: Aceita uma resposta vazia de um Comando de Interrogação Geral ou **1 ou qualquer outro valor diferente de 0**: Se a resposta de um Comando de Interrogação Geral vem vazia força-se uma desconexão, permitindo que este Driver conecte-se a outro equipamento em uma conexão de *backup*
Master - General Interrogation - Use Broadcast	0 (zero)	Define se este Driver no modo **Mestre**, ao enviar um Comando de Interrogação Geral, usa o endereço de *broadcast*. Gravada como "IEC.UseBroadcast". Os valores possíveis são **0**: Não usa o endereço de *broadcast* ou **1 ou qualquer outro valor diferente de 0**: Usa o endereço de *broadcast*
Master - Test Command - Test Command Enabled	0 (zero)	Habilita o uso do envio automático de um **Comando de Teste** neste Driver

PROPRIEDADE	VALOR PADRÃO	COMENTÁRIOS
		no modo **Mestre**. Gravada como "IEC.TestCommand.Enabled". Os valores possíveis são **0**: Desabilita ou **1 ou qualquer outro valor diferente de 0**: Habilita
Master - Test Command - Test Command Idle Time (sec)	15	Define o período de tempo, em segundos, que este Driver no modo **Mestre** tem que permanecer sem enviar Comandos do nível de aplicação para que se ative o envio automático de um **Comando de Teste**. Gravada como "IEC.TestCommand.IdleTimeSec"
Master - Clock Sync - Clock Sync Auto-run	0 (zero)	Habilita o envio de um **Comando de Sincronização de Relógio** neste Driver no modo **Mestre**. 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)	Habilita a repetição do envio de um **Comando de Sincronização de Relógio** neste Driver no modo **Mestre**. 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	Define o intervalo, em segundos para repetir o envio de um **Comando de Sincronização de Relógio** neste Driver no modo **Mestre**. Gravada como "IEC.ClockSync.PollTime"
Slave - Send EI (End Of Initialization) When Connected	0 (zero)	Determina se este Driver no modo **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 uma ASDU **M_EI_NA_1** sempre que uma conexão é estabelecida ou **1 ou qualquer outro valor diferente de 0**: Não envia uma ASDU **M_EI_NA_1**
Slave - Analog Values Send Interval (ms)	0 (zero)	Intervalo entre o **envio cíclico de dados analógicos**, em milissegundos, neste Driver no modo **Escravo**. Gravada como "IEC.AnalogSendIntervalMs"
Slave - Analog Deadband (% of previous value)	0 (zero)	Banda morta para **envio cíclico de dados analógicos** neste Driver no modo **Escravo**, em uma porcentagem

PROPRIEDADE	VALOR PADRÃO	COMENTÁRIOS
		de variação do valor anterior. Gravada como "IEC.AnalogDeadband"
Slave - Treat First Write As	0 (zero)	Determina se a primeira escrita efetuada nos Tags deste Driver no modo **Escravo** é considerada um novo evento ou o valor inicial de um Tag. Esta propriedade funciona até que o primeiro Comando de Interrogação Geral seja efetuado. Após o primeiro Comando de Interrogação Geral, todas as escritas em um Tag no modo **Escravo** são retornadas para um **Mestre** como novos eventos. Gravada como "IEC.FirstWriteIsCurrentValue". 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 Comando de Interrogação Geral mas não envia uma notificação espontânea
Slave - General Interrogation - Interrogation Enable Range	0 (zero)	Habilita o retorno de uma lista fixa de objetos em um Comando de **Interrogação Geral** neste Driver no modo **Escravo**. Gravada como "IEC.EnableInterrogationRange". Os valores possíveis são **0**: Retorna todos os objetos da memória *cache* deste Driver ou **1 ou qualquer outro valor diferente de 0**: Retorna apenas os objetos declarados na lista de Interrogação Geral
Slave - General Interrogation - List of data objects returned by General Interrogation	**String** vazia	Define a **lista de objetos que são retornados em um Comando de Interrogação Geral** neste Driver no modo **Escravo**. Gravada como "IEC.InterrogationRange"
Slave - General Interrogation - Remove timestamp from data sent on General Interrogation	0 (zero)	Determina se os dados de um Comando de **Interrogação Geral** deste Driver no modo **Escravo** são enviados sem a estampa de tempo. Gravada como "IEC.InterrogationRemoveTimestamp". Os valores possíveis são **0**: Envia os dados utilizando a ASDU original ou seja, 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 um Comando de Interrogação Geral, convertendo-os para uma ASDU sem estampa de tempo

PROPRIEDADE	VALOR PADRÃO	COMENTÁRIOS
Slave - Command Handling - Handle Mode for Clock Sync commands	0 (zero)	Define o tratamento para **Comandos de Sincronização de Relógio** recebidos por este Driver no modo **Escravo**. Gravada como "IEC.CommandHandleMode.ClockSync". Os valores possíveis são **0**: Padrão (desabilitado), **1**: Desabilitado, **2**: **Passa para uma aplicação** ou **3**: Trata automaticamente
Slave - Command Handling - Handle Mode for Select/Execute commands	0 (zero)	Define o tratamento para **Comandos de Seleção ou Execução** recebidos por este Driver no modo **Escravo**. Gravada como "IEC.CommandHandleMode.SelectExecute". Os valores possíveis são **0**: Padrão (passa para uma aplicação), **1**: Desabilitado, **2**: **Passa para uma aplicação** ou **3**: Trata automaticamente
Slave - Command Handling - Handle Mode for Test commands	0 (zero)	Define o tratamento para **Comandos de Teste** recebidos por este Driver no modo **Escravo**. Gravada como "IEC.CommandHandleMode.Test". Os valores possíveis são **0**: Padrão (trata automaticamente), **1**: Desabilitado, **2**: Passa para uma aplicação ou **3**: Trata automaticamente
Slave - Cyclic - Cyclic Data Transmission (1): Send Period (ms)	10000	Intervalo, em milissegundos, para o **envio dos dados cíclicos** declarados na propriedade "IEC.CyclicRange" deste Driver no modo **Escravo**. Gravada como "IEC.CyclidPeriodMs". **NOTA**: O nome de gravação desta variável possui um erro de digitação (*Cyclid* ao invés de *Cyclic*), mantido por questão de compatibilidade com versões antigas deste Driver
Slave - Cyclic - Cyclic Data Transmission (1): Object Range	**String** vazia	Define os dados que devem ser **enviados de forma cíclica** por este Driver no modo **Escravo**. Gravada como "IEC.CyclicRange"
Slave - Cyclic - Cyclic Data Transmission (2): Send Period (ms)	5000	Intervalo, em milissegundos, para o **envio dos dados cíclicos** declarados na propriedade "IEC.CyclicRange2" deste Driver no modo **Escravo**. Gravada como "IEC.CyclicPeriodMs2"
Slave - Cyclic - Cyclic Data Transmission (2): Object Range	**String** vazia	Define um segundo conjunto de dados que devem ser **enviados de forma cíclica** por este Driver no modo **Escravo**. Gravada como "IEC.CyclicRange2"
Slave - Offline Buffers - Offline Class1 buffering	0 (zero)	Habilita a manutenção de eventos de Classe 1 (um, alta prioridade) **em**

PROPRIEDADE	VALOR PADRÃO	COMENTÁRIOS
		memória neste Driver em modo **Escravo** 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	Número de eventos de Classe 1 (um, alta prioridade) a **manter em memória** por este Driver no modo **Escravo** quando um **Mestre** está desconectado. Gravada como "IEC.OfflineClass1.BufferSize"
Slave - Offline Buffers - Offline Class2 buffering	0 (zero)	Habilita manter eventos de Classe 2 (dois, baixa prioridade) **em memória** por este Driver no modo **Escravo** 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	Número de eventos de Classe 2 (dois, baixa prioridade) a **manter em memória** por este Driver no modo **Escravo** quando um **Mestre** está desconectado. Gravada como "IEC.OfflineClass2.BufferSize"
104 - T0 (sec)	30	Configura o parâmetro **T0**, em segundos, do protocolo **IEC 870-5-104**. Os valores possíveis variam entre 1 (um) e 255 segundos. Gravada como "IEC.IEC104.T0"
104 - T1 (sec)	15	Configura o parâmetro **T1**, em segundos, do protocolo **IEC 870-5-104**. Os valores possíveis variam entre 1 (um) e 255 segundos. Gravada como "IEC.IEC104.T1"
104 - T2 (sec)	10	Configura o parâmetro **T2**, em segundos, do protocolo **IEC 870-5-104**. Os valores possíveis variam entre 1 (um) e 255 segundos. Gravada como "IEC.IEC104.T2"
104 - T3 (sec)	20	Configura o parâmetro **T3**, em segundos, do protocolo **IEC 870-5-104**. Os valores possíveis variam entre 1 (um) e 255 segundos. Gravada como "IEC.IEC104.T3"
104 - K	12	Configura o parâmetro **K**, ou número de pacotes, do protocolo **IEC 870-5-104**. Os valores possíveis variam entre

PROPRIEDADE	VALOR PADRÃO	COMENTÁRIOS
		1 (um) e 32767 pacotes. Gravada como "IEC.IEC104.K"
104 - W	8 (oito)	Configura o parâmetro **W**, ou número de pacotes, do protocolo **IEC 870-5-104**. Os valores possíveis variam entre 1 (um) e 32767 pacotes. Gravada como "IEC.IEC104.W"
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 "Station TimeZone", 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 utilizar o fuso horário de Brasília (GMT-3). Gravada como "IEC.TimeZoneMinutes"
Timestamp - SP Timestamp Format	2 (dois)	Define o formato da estampa de tempo dos dados de **Ponto Simples** enviados por este Driver no modo **Escravo** para um **Mestre** fora de um Comando de 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)	Define o formato da estampa de tempo dos dados de **Ponto Duplo** enviados por este Driver no modo **Escravo** para um **Mestre** fora de um Comando de 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**:

PROPRIEDADE	VALOR PADRÃO	COMENTÁRIOS
		Com estampa de tempo de 56 bits (utiliza a ASDU **M_DP_TB_1**, 31)
Timestamp - BO Timestamp Format	2 (dois)	Define o formato da estampa de tempo dos dados de **Bitstring** enviados por este Driver no modo **Escravo** para um **Mestre** fora de um Comando de 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)	Define o formato da estampa de tempo dos dados de **Posição de Passo** enviados por este Driver no modo **Escravo** para um **Mestre** fora de um Comando de 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)	Define o formato da estampa de tempo dos dados de **Medidas Analógicas Normalizadas** enviados por este Driver no modo **Escravo** para um **Mestre** fora de um Comando de 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)	Define o formato da estampa de tempo dos dados de **Medidas Analógicas com Escala** enviados por este Driver no modo **Escravo** para um **Mestre** fora de um Comando de 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**,

PROPRIEDADE	VALOR PADRÃO	COMENTÁRIOS
		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)	Define o formato da estampa de tempo dos dados de **Medidas Analógicas em Ponto Flutuante** enviados por este Driver no modo **Escravo** para um **Mestre** fora de um Comando de Interrogação Geral, apenas para Tags configurados por **String**. Gravada como "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)	Define a forma e prioridade de envio dos dados de **Ponto Digital Simples** deste Driver no modo **Escravo** para um **Mestre**, 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 um Comando de Interrogação Geral
Send Priority - DP Send Priority	1 (um)	Define a forma e prioridade de envio dos dados de **Ponto Digital Duplo** deste Driver no modo **Escravo** para um **Mestre**, 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 um Comando de Interrogação Geral
Send Priority - BO Send Priority	2 (dois)	Define a forma e prioridade de envio dos dados de **Bitstring** deste Driver no modo **Escravo** para um **Mestre**, 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 um Comando de Interrogação Geral
Send Priority - ST Send Priority	2 (dois)	Define a forma e prioridade de envio dos dados de **Posição de Passo** deste Driver no modo **Escravo** para um Mestre, apenas para Tags

PROPRIEDADE	VALOR PADRÃO	COMENTÁRIOS
		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 no Comando de Interrogação Geral
Send Priority - MEn Send Priority	2 (dois)	Define a forma e prioridade de envio das **Medidas Analógicas Normalizadas** deste Driver no modo **Escravo** para um **Mestre**, 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 um Comando de Interrogação Geral
Send Priority - MEs Send Priority	2 (dois)	Define a forma e prioridade de envio das **Medidas Analógicas com Escala** deste Driver no modo **Escravo** para um **Mestre**, 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 um Comando de Interrogação Geral
Send Priority - MEf Send Priority	2 (dois)	Define a forma e prioridade de envio das **Medidas Analógicas em Ponto Flutuante** deste Driver no modo **Escravo** para um **Mestre**, 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 um Comando de Interrogação Geral
Boolean Values - Convert DCO to Boolean	0 (zero)	Define o formato dos valores do **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 dos **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**:

PROPRIEDADE	VALOR PADRÃO	COMENTÁRIOS
		Usa como um valor Booleano (**0**: OFF ou **1**: ON)
Boolean Values - Convert RCO to Boolean	0 (zero)	Define o formato dos valores do **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)
Boolean Values - Convert SEP to Boolean	0 (zero)	Define o formato dos valores de um Evento Simples de 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)

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
LinkLayer104: Wrong frame sequence (received #XXX, expected #YYY), disconnecting...	Modos **Mestre** e **Escravo**. Esta mensagem reporta o recebimento de uma mensagem fora de sequência no protocolo **IEC 870-5-104**. Quando este Driver detecta esta situação, força uma reconexão. A causa é provavelmente uma mensagem perdida pela rede. Para solucionar, investigue a conexão física ou a rede entre este Driver e um equipamento
ERROR: Failed to send command 'XXX', application layer is locked	Somente no modo **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 um Comando de 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 *time-out* de uma aplicação. A causa é que a Camada de Aplicação deste Driver está trancada por muito tempo. Para solucionar, aumente o *time-out* desta aplicação
GeneralInterrogation: received actterm with no GI active, retrying last GI	Somente no modo **Mestre**. Esta mensagem ocorre quando este Driver recebe uma confirmação de término de um Comando de Interrogação Geral e não há processo de interrogação geral ativo. As causas podem ser um Comando de Interrogação Geral iniciado por este Driver

ERRO	DESCRIÇÃO OU SOLUÇÃO
	anteriormente que foi interrompido devido a um *time-out* da camada de aplicação deste Driver, um Comando de Interrogação Geral iniciado por este Driver anteriormente que foi interrompido devido a uma queda na conexão com um equipamento ou este Driver está configurado para não aguardar a notificação de término de um Comando de Interrogação Geral e um **Escravo** está enviando esta notificação de término. As soluções possíveis são aumentar o *time-out* na aplicação deste Driver, investigar a causa da interrupção na conexão ou configurar este Driver para aguardar o término de um Comando de Interrogação Geral, e possivelmente também considerar aumentar o *time-out* na aplicação deste Driver

Documentação das Interfaces de Comunicação

Esta seção contém a documentação das Interfaces de Comunicação referentes ao Driver **IEC870-104**.

Configurações de um Driver

A configuração das Interfaces de Comunicação é realizada na caixa de diálogo de configuração de um Driver. Para acessar a configuração da caixa de diálogo no **Elipse E3** na versão 1.0, siga estes passos:

1. Clique com o botão direito do mouse em um objeto Driver (IODriver).
2. Selecione o item **Propriedades** no menu contextual.
3. Selecione a aba **Driver**.
4. Clique em **Outros parâmetros**.

No **Elipse E3** versão 2.0 ou posterior, clique em **Configurar o driver** na barra de ferramentas de um Driver. No **Elipse SCADA**, siga estes passos:

1. Abra o Organizer.
2. Selecione um Driver na árvore do Organizer.
3. Clique em **Extras** na aba **Driver**.

Atualmente, as Interfaces de Comunicação permitem que apenas uma conexão seja aberta para cada Driver. Isto significa que, no caso de acesso a duas portas seriais, é preciso adicionar dois Drivers em um aplicação e configurar cada um destes Drivers para cada porta serial.

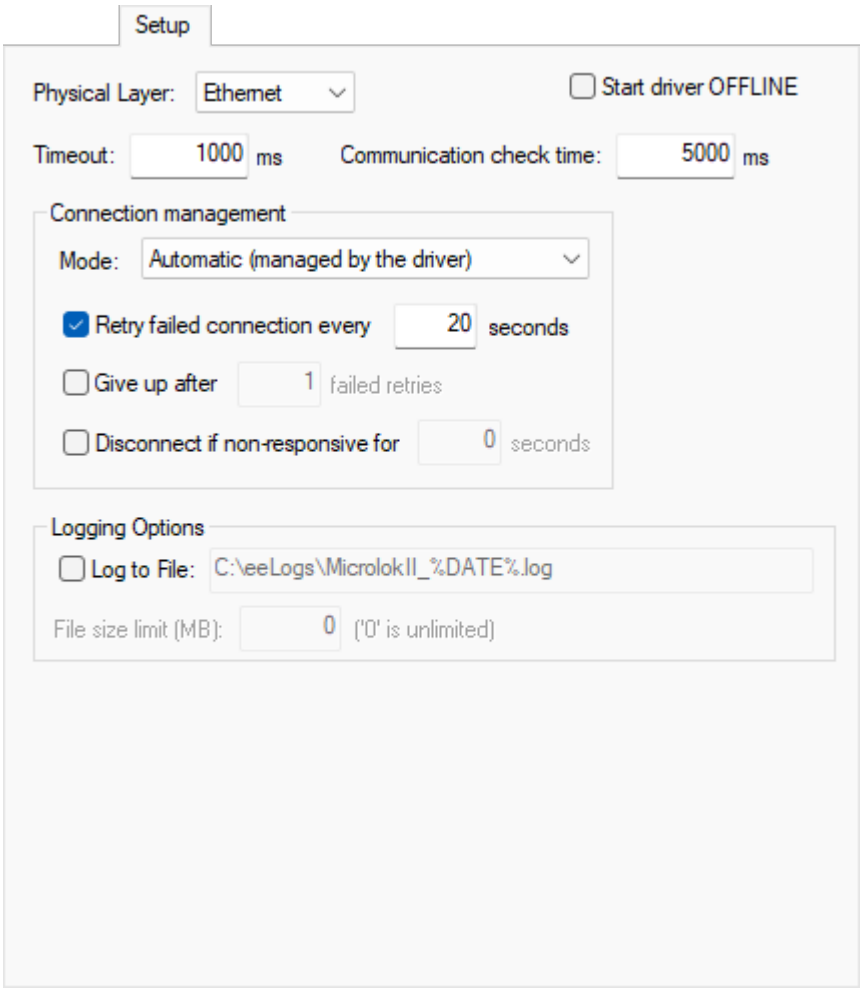
Caixa de Diálogo de Configuração

A caixa de diálogo das Interfaces de Configuração permite configurar a conexão de I/O que é utilizada por um Driver. Esta caixa de diálogo contém as abas **Setup**, **Serial**, **Ethernet**, **Modem** e **RAS** descritas nos tópicos a seguir. Se um Driver não implementa uma conexão de I/O específica, a respectiva aba não está disponível para configuração. Alguns Drivers podem conter abas adicionais, específicas para aquele Driver, na caixa de diálogo de configuração.

Aba Setup

A aba **Setup** contém a configuração geral de um Driver. Esta aba é dividida nos seguintes grupos:

- **Configurações gerais**: Configurações da camada física de um Driver, *time-out* e modo de inicialização
- **Connection management**: Configurações de como a Interface de Comunicação mantém a conexão e qual a política de recuperação em caso de falha
- **Logging options**: Controla a geração dos arquivos de log



Aba Setup

Opções gerais da aba Setup

OPÇÃO	DESCRIÇÃO
Physical Layer	Selecione a interface física em uma lista. As opções disponíveis são **Serial**, **Ethernet**, **Modem** e **RAS**. A interface selecionada deve ser configurada na aba específica
Timeout	Configure o *time-out*, em milissegundos, para a camada física. Esta é a medida de tempo que a interface de I/O aguarda para a recepção de um byte qualquer do *buffer* de recepção
Communication check time	Configure o tempo, em milissegundos, para definir o intervalo em que a comunicação é considerada em estado inativo. Enquanto um Driver de Comunicação receber dados válidos, o estado de comunicação é considerado ativo. Porém, se durante o funcionamento um Driver de Comunicação não receber dados válidos neste período de tempo, o estado é considerado inativo. O estado de comunicação é mostrado no Tag **IO.CommunicationStatus**

OPÇÃO	DESCRIÇÃO
Start driver OFFLINE	Selecione esta opção para que um Driver inicie em modo **Offline** ou parado. Isto significa que a interface de I/O não é criada até que se configure um Driver em modo **Online** utilizando-se um Tag em uma aplicação. Este modo possibilita a configuração dinâmica da interface de I/O em tempo de execução

Opções para o grupo Connection management

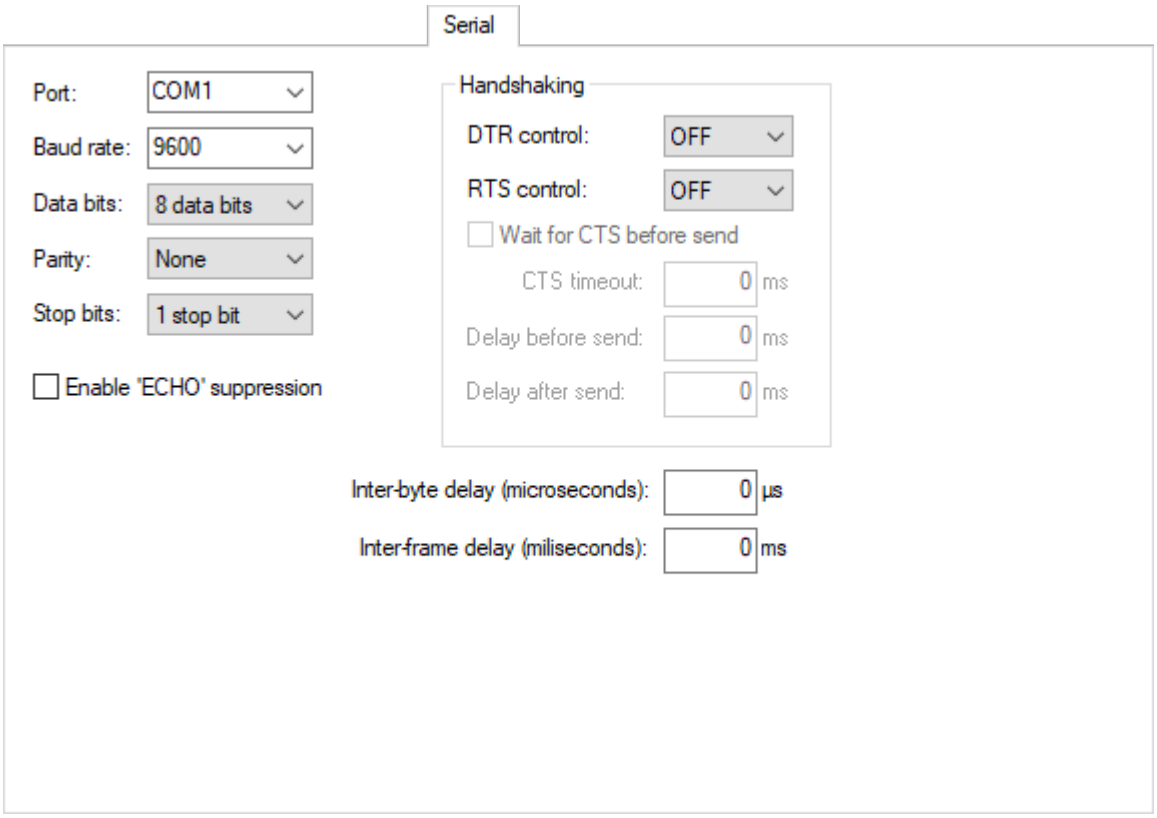
OPÇÃO	DESCRIÇÃO
Mode	Seleciona o modo de gerenciamento de conexão. Selecionar a opção **Automatic** permite que um Driver gerencie a conexão automaticamente, como especificado nas opções seguintes. Selecionar a opção **Manual** permite que uma aplicação gerencie a conexão completamente
Retry failed connection every ... seconds	Selecione esta opção para habilitar a retentativa de conexão de um Driver em um determinado intervalo, em segundos. Se a opção **Give up after failed retries** não está selecionada, este Driver continua retentando até que a conexão seja efetuada, ou que a aplicação seja parada
Give up after ... failed retries	Habilite esta opção para definir um número máximo de retentativas de conexão. Quando o número especificado de tentativas consecutivas de reconexão é atingido, um Driver vai para o modo **Offline**, assumindo que um problema de hardware foi detectado. Se um Driver estabelece uma conexão com sucesso, o número de retentativas sem sucesso é zerado. Se esta nova conexão é perdida, então o contador de retentativas inicia do zero
Disconnect if non-responsive for ... seconds	Habilite esta opção para forçar um Driver a se desconectar se nenhum byte chegou à interface de I/O no *time-out* especificado, em segundos. Este *time-out* deve ser maior que o *time-out* configurado na opção **Timeout**

Opções para o grupo Logging Options

OPÇÃO	DESCRIÇÃO
Log to File	Habilite esta opção e configure o nome do arquivo onde o log é escrito. Arquivos de log podem ser bem extensos, portanto utilize esta opção por curtos períodos de tempo, apenas para o propósito de testes e depurações. Caso se utilize a macro **%PROCESS%** no nome do arquivo de log, esta é substituída pelo identificador do processo atual. Esta opção é particularmente útil ao se utilizar várias instâncias de um mesmo Driver no **Elipse E3**, permitindo assim que cada instância gere um arquivo separado de log. Por exemplo, ao configurar esta opção com o valor "c:\e3logs\drivers\sim_%PROCESS%.log", gera-se um arquivo c:\e3logs\drivers\sim_00000FDA.log para o processo **0FDAh**. Pode-se também utilizar a macro **%DATE%** no nome do arquivo. Neste caso é gerado um arquivo de log por dia, no formato **aaaa_mm_dd**. Por exemplo, ao configurar esta opção com o valor "c:\e3logs\drivers\sim_%DATE%.log", gera-se o arquivo c:\e3logs\drivers\sim_2005_12_31.log em 31/12/2005 e o arquivo c:\e3logs\drivers\sim_2006_01_01.log em 01/01/2006. De forma semelhante, a macro **%DATE_HOUR%** gera um arquivo de log por hora, no formato **aaaa_mm_dd_hh**
File size limit (MB)	Configure o limite de tamanho do arquivo de log, em megabytes. Um valor igual a 0 (zero) significa que não há limite de tamanho para o arquivo de log

Aba Serial

Utilize esta aba para configurar os parâmetros da Interface **Serial**.



Aba Serial

Opções gerais da aba Serial

OPÇÃO	DESCRIÇÃO
Port	Selecione uma porta serial a partir da lista, de **COM1** até **COM4**, ou digite o nome de uma porta serial no formato **COM*n***, como por exemplo "COM15". Ao digitar o nome de uma porta serial manualmente, a caixa de diálogo aceita apenas nomes de portas seriais começando com a expressão "COM"
Baud rate	Selecione um *baud rate* a partir da lista (**1200**, **2400**, **4800**, **9600**, **19200**, **38400**, **57600** ou **115200**) ou digite um *baud rate*, como por exemplo 600
Data bits	Selecione 7 (sete) ou 8 (oito) bits de dados a partir da lista
Parity	Selecione uma paridade a partir da lista. As opções disponíveis são **None**, **Even**, **Odd**, **Mark** ou **List**
Stop bits	Selecione o número de stop bits a partir da lista. As opções disponíveis são **1**, **1.5** ou **2** stop bits
Enable 'ECHO' supression	Habilite esta opção para remover o eco recebido após a Interface de Comunicação enviar dados por uma porta serial. Se o eco não é igual aos bytes recém enviados, a Interface de Comunicação aborta a comunicação
Inter-byte delay (microseconds)	Defina uma espera entre cada byte transmitido pela Interface de Comunicação, em milionésimos de segundo, ou seja, 1000000 é igual a um segundo. Esta opção deve ser utilizada com esperas pequenas de menos de um milissegundo
Inter-frame delay (milliseconds)	Defina uma espera entre pacotes enviados ou recebidos pela Interface de Comunicação, em milésimos de segundo,

OPÇÃO	DESCRIÇÃO
	ou seja, 1000 é igual a um segundo. Esta espera é aplicada caso a Interface de Comunicação envie dois pacotes consecutivos, ou entre um pacote recebido e o próximo envio

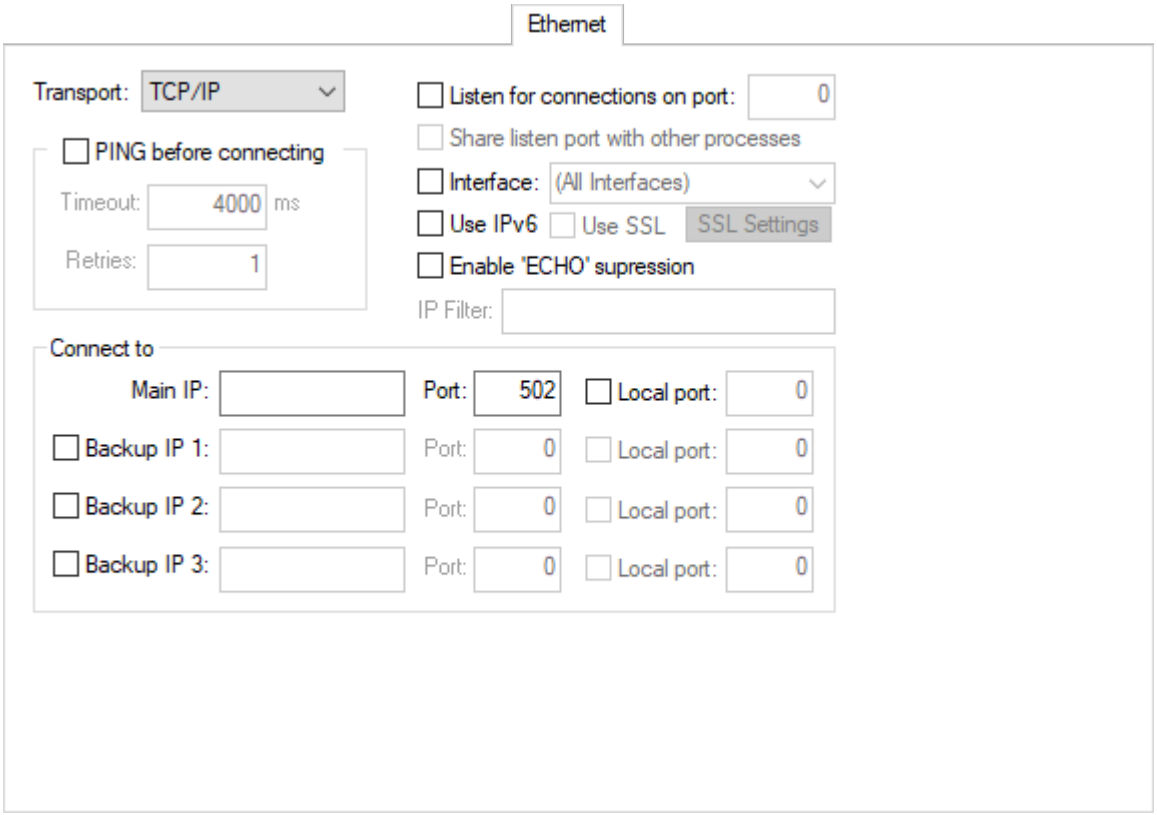
O grupo **Handshaking** configura o uso dos sinais **RTS**, **CTS** e **DTR** no processo de *handshaking* ou seja, controla quando um dado pode ser enviado ou recebido através de uma linha serial. Na maioria das vezes, configurar a opção **DTR control** para **ON** e a opção **RTS control** para **Toggle** funciona tanto com linhas seriais do tipo **RS232** quanto com linhas seriais do tipo **RS485**.

Opções disponíveis no grupo Handshaking

OPÇÃO	DESCRIÇÃO
DTR control	Selecione o valor **ON** para deixar o sinal **DTR** sempre ligado enquanto a porta serial está aberta. Selecione o valor **OFF** para desligar o sinal **DTR** enquanto a porta serial está aberta. Alguns equipamentos exigem que o sinal **DTR** esteja ligado para permitir a comunicação
RTS control	Selecione o valor **ON** para deixar o sinal **RTS** sempre ligado enquanto a porta serial está aberta. Selecione o valor **OFF** para desligar o sinal **RTS** enquanto a porta serial está aberta. Selecione o valor **Toggle** para ligar o sinal **RTS** enquanto se envia os bytes através da porta serial, e desligá-lo quando não se está enviando bytes e, portanto, habilitando a recepção
Wait for CTS before send	Disponível apenas quando a opção **RTS control** está configurada com o valor **Toggle**. Utilize esta opção para forçar um Driver a verificar o sinal **CTS** antes de enviar os bytes através da porta serial, após ligar o sinal de **RTS**. Neste modo o sinal **CTS** é tratado como um *flag* de permissão para envio
CTS timeout	Determina o tempo máximo, em milissegundos, que um Driver aguarda pelo sinal de **CTS** depois de ligar o sinal de **RTS**. Se o sinal de **CTS** não é levantado dentro deste *time-out*, este Driver falha a comunicação atual e retorna erro
Delay before send	Alguns equipamentos de porta serial demoram a habilitar o circuito de envio de dados depois que o sinal **RTS** é ligado. Configure esta opção para aguardar uma determinada quantidade de milissegundos depois de ligar o sinal **RTS** e antes de enviar o primeiro byte. **IMPORTANTE**: Esta espera deve ser utilizada com muito cuidado, pois consome 100% dos recursos de CPU enquanto aguarda. A performance geral do sistema se degrada conforme este valor aumenta
Delay after send	Tem o mesmo efeito que a opção **Delay before send**, mas neste caso a espera é efetuada depois que o último byte é enviado, antes de desligar o sinal **RTS**

Aba Ethernet

Utilize esta aba para configurar os parâmetros da Interface **Ethernet**. Estes parâmetros, exceto as configurações de porta, devem ser também configurados para uso na Interface **RAS**.



Aba Ethernet

Opções disponíveis na aba Ethernet

OPÇÃO	DESCRIÇÃO
Transport	Selecione o valor **TCP/IP** para um *socket* TCP (*stream*) ou selecione o valor **UDP/IP** para utilizar um *socket* UDP (*connectionless datagram*)
Listen for connections on port	Utilize esta opção para aguardar por novas conexões em uma porta IP específica, comum em Drivers Escravos. Caso esta opção permaneça desmarcada, um Driver se conecta ao endereço e porta especificados no grupo **Connect to**
Share listen port with other processes	Selecione esta opção para compartilhar a porta de escuta com outros Drivers e processos
Interface	Selecione a interface de rede local, identificada pelo endereço IP, que um Driver utiliza para efetuar e receber conexões, ou selecione o valor **(All Interfaces)** para permitir conexões em qualquer interface de rede
Use IPv6	Selecione esta opção para forçar um Driver a utilizar endereços no formato **IPv6** em todas as conexões Ethernet. Deixe esta opção desmarcada para utilizar o formato **IPv4**
Enable 'ECHO' suppression	Habilite esta opção para eliminar o *eco* dos dados recebidos. O *eco* é uma cópia dos dados enviados, que pode ser retornada antes da mensagem de resposta

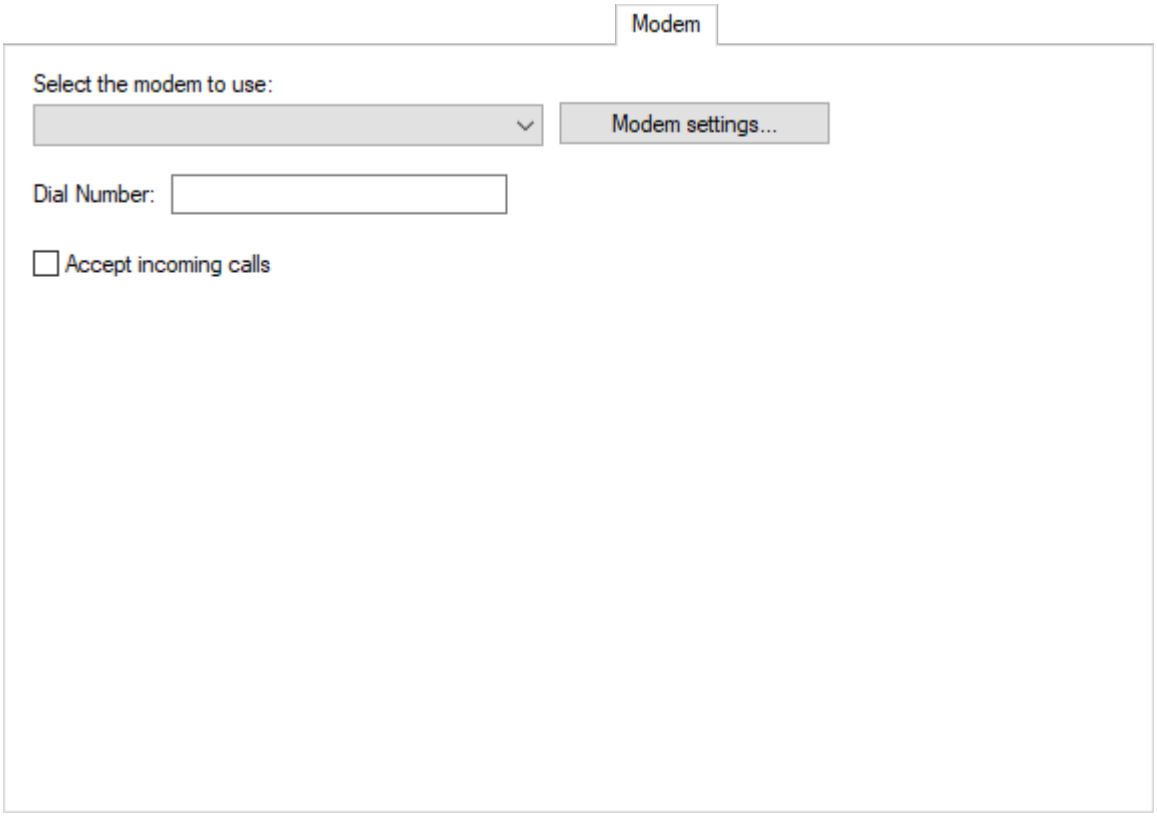
OPÇÃO	DESCRIÇÃO
IP Filter	Lista de endereços IP restringidos ou permitidos de onde um Driver aceita conexões (*Firewall*). Consulte a propriedade **IO.Ethernet.IPFilter** para mais informações
PING before connecting	Habilite esta opção para executar um comando **ping**, ou seja, para verificar se um dispositivo pode ser encontrado na rede, em um dispositivo antes de tentar uma conexão com o *socket*. Esta é uma maneira rápida de determinar uma conexão bem sucedida antes de tentar abrir um *socket* com um dispositivo. O *time-out* de uma conexão com um *socket* pode ser bem alto. As opções disponíveis são: • **Timeout**: Especifique o número de milissegundos de espera por uma resposta de um comando **ping**. Deve-se usar um comando **ping** para verificar o tempo normal de resposta, configurando esta opção para um valor acima desta média. Normalmente pode-se configurar um valor entre 1000 e 4000 milissegundos, ou seja entre 1 (um) e 4 (quatro) segundos • **Retries**: Número de retentativas de um comando **ping**, sem contar a tentativa inicial. Se todas as tentativas falharem, então a conexão com o *socket* é abortada

Opções disponíveis no grupo Connect to

OPÇÃO	DESCRIÇÃO
Main IP	Digite o endereço IP de um dispositivo remoto. Pode-se usar tanto o endereço IP separado por pontos quanto uma URL. No caso de uma URL, um Driver usa o serviço de DNS disponível para mapear a URL para um endereço IP, como por exemplo "192.168.0.13" ou "Server1"
Port	Digite a porta IP de um dispositivo remoto, entre 0 (zero) e 65535
Local port	Selecione esta opção para utilizar uma porta IP local fixa ao conectar a um dispositivo remoto
Backup IP 1, 2 e 3	Indique o endereço IP, a porta IP e a porta IP local fixa de até 3 (três) endereços de *backup* de um dispositivo remoto

Aba Modem

Utilize esta aba para configurar os parâmetros da Interface **Modem**. Algumas opções da aba **Serial** afetam a configuração de um modem, portanto é interessante não esquecer de configurar a Interface **Serial**.



Aba Modem

A Interface **Modem** utiliza os modems TAPI instalados no computador.

Opções disponíveis na aba Modem

OPÇÃO	DESCRIÇÃO
Select the modem to use	Selecione um modem a partir da lista de modems disponíveis no computador. Caso selecione-se o valor **Default modem**, então o primeiro modem disponível é utilizado. Selecionar este valor é recomendado especialmente quando uma aplicação é utilizada em outro computador
Modem settings	Clique para abrir a janela de configuração do modem selecionado
Dial Number	Digite o número padrão para discagem. Este valor pode ser modificado em tempo de execução. Pode-se utilizar o caractere **w** para representar uma pausa ou espera pelo tom de discagem. Por exemplo, "0w33313456" disca o número 0 (zero), espera e então disca o número "33313456"
Accept incoming calls	Habilite esta opção para que um Driver atenda o telefone quando receber uma chamada externa. Para utilizar esta opção é necessário configurar a opção **Connection management** na aba **Setup** para o valor **Manual**

Aba RAS

Use esta aba para configurar os parâmetros da Interface **RAS**. É necessário também configurar a aba **Ethernet**.

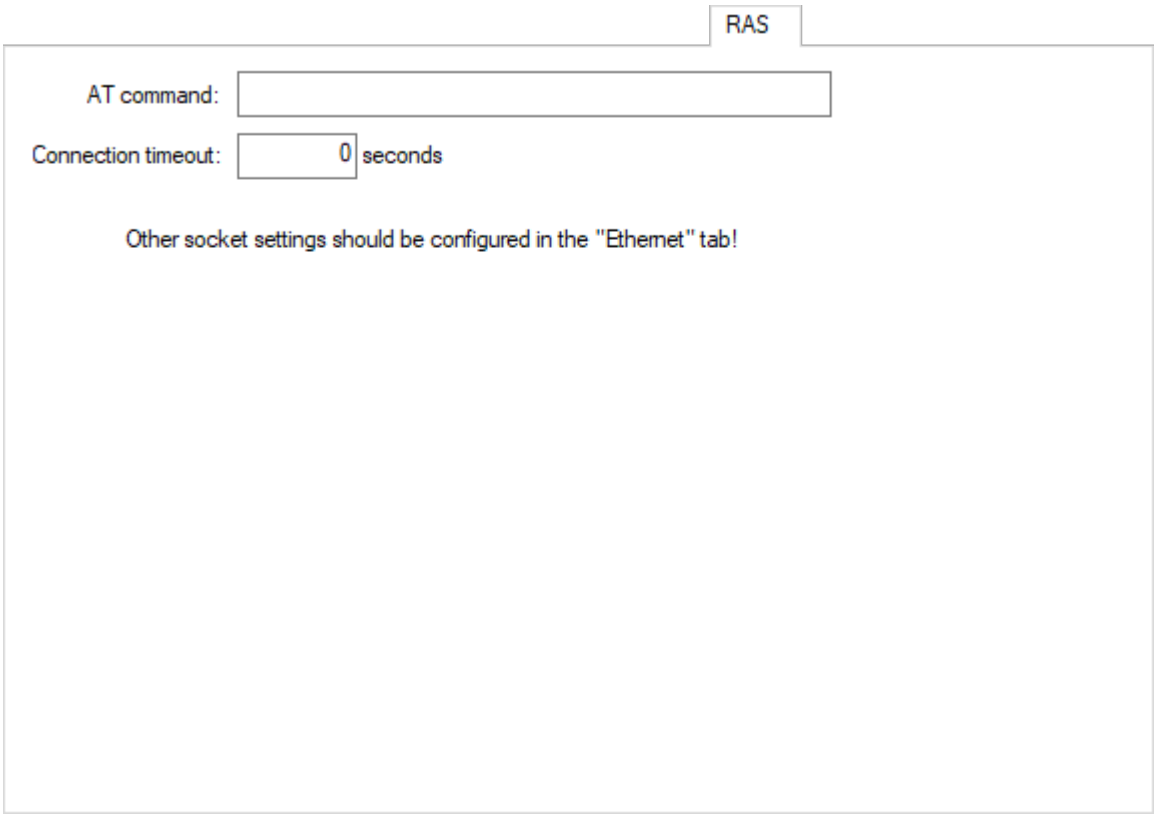
A Interface **RAS** abre uma conexão *socket* com um dispositivo RAS. Um dispositivo RAS é um servidor de modems acessível através de TCP/IP, aguardando por conexões *socket* em uma porta IP. Para cada conexão aceita nesta porta tem-se acesso a um modem.

Ao conectar-se a um dispositivo RAS, primeiramente a Interface de Comunicação conecta ao *socket* no endereço IP e na porta configurados na aba **Ethernet**. Depois que o *socket* é aberto, os passos de inicialização ou de conexão a seguir são efetuados:

1. Limpeza do *socket*, ou seja, remove qualquer mensagem de saudação **TELNET** recebida de um dispositivo RAS.
2. Envio de um comando de discagem **AT**, no formato **ASCII**, no *socket*.
3. Aguarda pela recepção de uma resposta **CONNECT**.
4. Caso o *time-out* expire, a conexão é abortada.
5. Se a resposta **CONNECT** é recebida dentro do *time-out*, o *socket* está disponível para comunicação com um dispositivo, ou seja, a conexão foi estabelecida.

Se o passo 5 (cinco) é efetuado com sucesso, então o *socket* comporta-se como um *socket* normal, com o dispositivo RAS funcionando como um roteador entre um Driver e o dispositivo. Os bytes enviados por um Driver são recebidos pelo dispositivo RAS e enviados para o dispositivo de destino utilizando um modem. Os bytes recebidos pelo dispositivo RAS do modem são enviados de volta a um Driver utilizando o mesmo *socket*.

Depois que a conexão é estabelecida, a Interface **RAS** monitora os dados recebidos por um Driver. Caso uma **String** "NO CARRIER" seja encontrada, o *socket* é fechado. Se o dispositivo RAS não envia o sinal **NO CARRIER**, a Interface **RAS** não consegue detectar quando a conexão modem entre o dispositivo RAS e o dispositivo final de I/O falha. Para recuperação de tal falha é fortemente recomendado que seja habilitada a opção **Disconnect if non-responsive** na aba **Setup**.



Aba RAS

Opções disponíveis na aba RAS

OPÇÃO	DESCRIÇÃO
AT command	Uma **String** com o comando **AT** completo usado para discar para um dispositivo de destino. Por exemplo, "ATDT33313456" disca por tom para o número "33313456"
Connection timeout	Número de segundos a aguardar por uma resposta **CONNECT** do modem, após o envio de um comando **AT**

Configurações Gerais

Esta seção contém informações sobre a configuração dos **Tags de Comunicação** e das **Propriedades** gerais das Interfaces de Comunicação.

Tags de Comunicação

Tags Gerais das Interfaces de Comunicação (N2/B2 = 0)

Os Tags descritos a seguir são fornecidos para todas as Interfaces de I/O suportadas.

IO.CommunicationStatus

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Leitura
Parâmetro N1	-1 (menos um)
Parâmetro N2	0 (zero)
Parâmetro N3	0 (zero)
Parâmetro N4	6 (seis)
Configuração por String	IO.CommunicationStatus

Este Tag informa o estado da comunicação de um Driver. Indica o funcionamento da comunicação em função do recebimento de dados válidos dentro de um período de tempo arbitrado na configuração. Para mais informações, consulte o tópico **Aba Setup**. Os valores possíveis são **0 - Comunicação inativa**: O Driver não recebeu dados válidos ou deixou de receber dados depois de *n* milissegundos, conforme configurado na janela de propriedades, ou **1 - Comunicação ativa**: O Driver está recebendo dados válidos.

IO.IOKitEvent

Tipo de Tag	Tag Bloco
Tipo de Acesso	Somente Leitura
Parâmetro B1	-1 (menos um)
Parâmetro B2	0 (zero)
Parâmetro B3	0 (zero)
Parâmetro B4	1 (um)
Propriedade Size	4 (quatro)
Propriedade ParamItem	IO.IOKitEvent

Este Bloco retorna eventos de Driver gerados por várias fontes nas Interfaces de Comunicação. A propriedade **TimeStamp** de um Bloco representa o momento em que um evento ocorre. Os Elementos de Bloco são os seguintes:

- **Elemento 0**: Tipo de evento. Os valores possíveis são **0**: Informação, **1**: Advertência ou **2**: Erro
- **Elemento 1**: Fonte de um evento. Os valores possíveis são **0**: Driver (específico de um Driver), **-1**: IOKit (eventos genéricos da Interface de Comunicação), **-2**: Interface **Serial**, **-3**: Interface **Modem**, **-4**: Interface **Ethernet** ou **-5**: Interface **RAS**
- **Elemento 2**: Número do erro, específico de cada fonte de evento
- **Elemento 3**: Mensagem de um evento, uma **String** específica de cada evento

NOTA

Um Driver mantém um número máximo de 100 eventos internamente. Se eventos adicionais são reportados, os eventos mais antigos são descartados.

IO.PhysicalLayerStatus

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Somente Leitura
Parâmetro N1	-1 (menos um)
Parâmetro N2	0 (zero)
Parâmetro N3	0 (zero)
Parâmetro N4	2 (dois)
Configuração por String	IO.PhysicalLayerStatus

Este Tag indica o estado da camada física. Os valores possíveis são os seguintes:

- **0**: Camada física parada, ou seja, um Driver está em modo **Offline**, a camada física falhou ao inicializar ou excedeu o número máximo de tentativas de reconexão
- **1**: Camada física iniciada mas não conectada, ou seja, um Driver está em modo **Online**, mas a camada física não está conectada. Se a opção **Connection management** está configurada com o valor **Automatic**, a camada física pode

estar conectando, desconectando ou esperando por uma tentativa de reconexão. Se a opção **Connection management** está configurada com o valor **Manual**, então a camada física permanece neste estado até ser forçada a conectar

- **2**: Camada física conectada, ou seja, a camada física está pronta para ser usada. Isto **NÃO** significa que um equipamento esteja conectado, apenas que a camada de acesso está funcionando

IO.SetConfigurationParameters

Tipo de Tag	Tag Bloco
Tipo de Acesso	Somente Leitura
Parâmetro B1	-1 (menos um)
Parâmetro B2	0 (zero)
Parâmetro B3	0 (zero)
Parâmetro B4	3 (três)
Propriedade Size	2 (dois)
Propriedade ParamItem	IO.SetConfigurationParameters

Use este Tag para modificar qualquer propriedade da caixa de diálogo de configuração de um Driver em tempo de execução.

Este Tag funciona somente enquanto um Driver está em modo **Offline**. Para iniciar um Driver em modo **Offline**, selecione a opção **Start driver OFFLINE** na caixa de diálogo de configuração deste Driver. Pode-se tanto escrever em um Tag PLC ou em um Tag Bloco contendo os parâmetros a serem modificados. As escritas de Elementos de Bloco individuais não são suportadas, um Bloco inteiro precisa ser escrito de uma vez só.

No **Elipse SCADA** é necessário usar um Tag Bloco. Cada parâmetro a ser configurado utiliza dois Elementos de Bloco. Por exemplo, caso seja necessário configurar 3 (três) parâmetros, então o tamanho do Bloco deve ser 6 (seis, 3 × 2). O primeiro Elemento é o nome da propriedade, como uma **String**, e o segundo Elemento é o valor desta propriedade, conforme o exemplo a seguir.

```
// 'Block' deve ser um Tag Bloco com leitura automática,
// leitura por varredura e escrita automática desabilitadas.
// Configura os parâmetros
Block.element001 = "IO.Type" // Parâmetro 1
Block.element002 = "Serial"
Block.element003 = "IO.Serial.Port" // Parâmetro 2
Block.element004 = 1
Block.element005 = "IO.Serial.BaudRate" // Parâmetro 3
Block.element006 = 19200
// Escreve o Bloco inteiro
Block.Write()
```

Ao usar o **Elipse E3**, a habilidade de criar *arrays* em tempo de execução permite o uso tanto de um Tag de Comunicação quanto de um Tag Bloco. Pode-se utilizar o método **Write** de um Driver para enviar os parâmetros diretamente para este Driver, sem a necessidade de criar um Tag, conforme o exemplo a seguir.

```
Dim arr(6)
' Configura os elementos do array
arr(1) = "IO.Type"
arr(2) = "Serial"
arr(3) = "IO.Serial.Port"
arr(4) = 1
arr(5) = "IO.Serial.BaudRate"
arr(6) = 19200
' Há dois métodos de enviar os parâmetros
' Método 1: Usando um Tag de Comunicação
tag.WriteEx arr
' Método 2: Sem utilizar um Tag
Driver.Write -1, 0, 0, 3, arr
```

Uma variação do exemplo anterior usa um *array* bidimensional.

```
Dim arr(10)
' Configura os elementos do array. Note que o array foi redimensionado
' para 10 elementos. Elementos vazios são ignorados pelo Driver
arr(1) = Array("IO.Type", "Serial")
arr(2) = Array("IO.Serial.Port", 1)
arr(3) = Array("IO.Serial.BaudRate", 19200)
Driver.Write -1, 0, 0, 3, arr
```

Um Driver não valida nomes de parâmetros ou valores passados, por isto tenha cuidado ao escrever parâmetros e valores. O método **Write** falha se o *array* de configuração é criado incorretamente. Pode-se consultar o log de um Driver ou usar o parâmetro *writeStatus* do método **WriteEx** para descobrir a causa exata de um erro.

```
Dim arr(10), strError
arr(1) = Array("IO.Type", "Serial")
arr(2) = Array("IO.Serial.Port", 1)
arr(3) = Array("IO.Serial.BaudRate", 19200)
If Not Driver.WriteEx -1, 0, 0, 3, arr, , , strError Then
  MsgBox "Falha ao configurar os parâmetros do Driver: " + strError
End If
```

IO.WorkOnline

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Leitura ou Escrita
Parâmetro N1	-1 (menos um)
Parâmetro N2	0 (zero)
Parâmetro N3	0 (zero)
Parâmetro N4	4 (quatro)
Configuração por String	IO.WorkOnline

Este Tag informa o estado atual de um Driver e permite iniciar ou parar a camada física. Os valores possíveis são os seguintes:

- **0 - Driver Offline**: A camada física está fechada ou parada. Este modo permite uma configuração dinâmica dos parâmetros de um Driver através do Tag **IO.SetConfigurationParameters**
- **1 - Driver Online**: A camada física está aberta ou em execução. Enquanto está em modo **Online**, a camada física pode ser conectada ou desconectada e o estado atual pode ser conferido no Tag **IO.PhysicalLayerStatus**

No exemplo a seguir, utilizando o **Elipse E3**, um Driver é colocado em modo **Offline**, a porta COM é modificada e então é colocado em modo **Online** novamente.

```
'Configura o Driver em modo Offline
Driver.Write -1, 0, 0, 4, 0
'Muda a porta para COM2
Driver.Write -1, 0, 0, 3, Array("IO.Serial.Port", 2)
'Configura o Driver em modo Online
Driver.Write -1, 0, 0, 4, 1
```

O método **Write** pode falhar ao configurar um Driver em modo **Online**, ou seja, escrevendo o valor 1 (um). Neste caso, este Driver permanece em modo **Offline**. A causa da falha pode ser:

- Tipo de camada física configurada incorretamente, provavelmente um valor inválido foi configurado para a propriedade **IO.Type**
- Este Driver pode ter ficado sem memória
- A camada física pode ter deixado de criar a *thread* de trabalho. Procure no arquivo de log pela mensagem "Failed to create physical layer thread!"
- A camada física não conseguiu inicializar. A causa da falha depende do tipo de camada física. Pode ser um número de porta serial inválida, falha ao inicializar o Windows Sockets ou falha ao inicializar o TAPI (modem), entre outras. A causa é gravada no arquivo de log

IMPORTANTE

Mesmo que a configuração de um Driver para o modo **Online** seja bem-sucedida, isto não significa necessariamente que a camada física esteja pronta para uso, ou seja, pronta para executar operações de entrada e saída com um equipamento externo. O Tag **IO.PhysicalLayerStatus** deve ser verificado para assegurar que a camada física esteja conectada e preparada para a comunicação.

Propriedades

Estas são as propriedades gerais de todas as Interfaces de I/O suportadas.

IO.ConnectionMode

Controla o modo de gerenciamento da Conexão. Os valores possíveis são **0**: Modo automático, em que um Driver gerencia a conexão ou **1**: Modo manual, em que uma aplicação gerencia a conexão.

IO.GiveUpEnable

Quando configurada para Verdadeiro, define um número máximo de tentativas de reconexão. Se todas as reconexões falharem, um Driver entra em modo **Offline**. Se configurada para Falso, um Driver tenta até que uma reconexão seja bem-sucedida.

IO.GiveUpTries

Número de tentativas de reconexão antes que esta seja abortada. Por exemplo, se o valor desta propriedade é igual a 1 (um), um Driver tenta apenas uma reconexão quando a conexão é perdida. Se esta falhar, este Driver entra em modo **Offline**.

IO.InactivityEnable

Configure em Verdadeiro para habilitar e em Falso para desabilitar a detecção de inatividade. A camada física é desconectada se está inativa por um certo período de tempo. A camada física é considerada inativa apenas se é capaz de enviar dados mas não de recebê-los de volta.

IO.InactivityPeriodSec

Número de segundos para a verificação de inatividade. Se a camada física está inativa por este período de tempo, então é desconectada.

IO.RecoverEnable

Configure em Verdadeiro para habilitar um Driver a recuperar conexões perdidas e em Falso para deixar um Driver em modo **Offline** quando uma conexão é perdida.

IO.RecoverPeriodSec

Tempo de espera entre duas tentativas de conexão, em segundos.

NOTA

A primeira reconexão é executada imediatamente após a conexão ser perdida.

IO.StartOffline

Configure em Verdadeiro para iniciar um Driver em modo **Offline** e em Falso para iniciar um Driver em modo **Online**.

NOTA

Não faz sentido modificar esta propriedade em tempo de execução, já que esta só pode ser modificada quando um Driver já está em modo **Offline**. Para configurar um Driver em modo **Online** em tempo de execução, escreva o valor 1 (um) no Tag **IO.WorkOnline**.

IO.TimeoutMs

Define o *time-out* da camada física, em milissegundos. Um segundo equivale a 1000 milissegundos.

IO.Type

Define o tipo de interface física utilizada por um Driver. Os valores possíveis são os seguintes:

- **N ou None**: Não utiliza uma interface física, ou seja, um Driver deve fornecer uma interface personalizada
- **S ou Serial**: Utiliza uma porta serial local (COM*n*)
- **M ou Modem**: Utiliza um modem local, interno ou externo, acessado via TAPI (*Telephony Application Programming Interface*)
- **E ou Ethernet**: Utiliza um *socket* TCP/IP ou UDP/IP
- **R ou RAS**: Utiliza uma Interface **RAS** (*Remote Access Server*). Um Driver conecta-se a um equipamento RAS através da Interface **Ethernet** e então emite um comando **AT** (*dial*)

Configuração de Estatísticas

Esta seção contém informações sobre a configuração dos **Tags de Comunicação** e das **Propriedades** das estatísticas das Interfaces de Comunicação.

Tags de Comunicação

Tags de Estatísticas das Interfaces de Comunicação (N2/B2 = 0)

Os Tags descritos a seguir mostram estatísticas para todas as Interfaces de Comunicação.

IO.Stats.Partial.BytesRecv

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Somente Leitura
Parâmetro N1	-1 (menos um)
Parâmetro N2	0 (zero)
Parâmetro N3	0 (zero)
Parâmetro N4	1101
Configuração por String	IO.Stats.Partial.BytesRecv

Este Tag retorna a quantidade de bytes recebidos na conexão atual.

IO.Stats.Partial.BytesSent

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Somente Leitura
Parâmetro N1	-1 (menos um)
Parâmetro N2	0 (zero)
Parâmetro N3	0 (zero)
Parâmetro N4	1100
Configuração por String	IO.Stats.Partial.BytesSent

Este Tag retorna a quantidade de bytes enviados na conexão atual.

IO.Stats.Partial.TimeConnectedSeconds

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Somente Leitura
Parâmetro N1	-1 (menos um)
Parâmetro N2	0 (zero)
Parâmetro N3	0 (zero)
Parâmetro N4	1102
Configuração por String	IO.Stats.Partial.TimeConnectedSeconds

Este Tag retorna o número de segundos que um Driver está conectado na conexão atual ou 0 (zero) se um Driver está desconectado.

IO.Stats.Partial.TimeDisconnectedSeconds

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Somente Leitura
Parâmetro N1	-1 (menos um)
Parâmetro N2	0 (zero)
Parâmetro N3	0 (zero)
Parâmetro N4	1103
Configuração por String	IO.Stats.Partial.TimeDisconnectedSeconds

Este Tag retorna o número de segundos que um Driver está desconectado desde o término da última conexão ou 0 (zero) se um Driver está conectado.

IO.Stats.Total.BytesRecv

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Somente Leitura
Parâmetro N1	-1 (menos um)
Parâmetro N2	0 (zero)
Parâmetro N3	0 (zero)
Parâmetro N4	1001
Configuração por String	IO.Stats.Total.BytesRecv

Este Tag retorna a quantidade de bytes recebidos desde que um Driver foi carregado.

IO.Stats.Total.BytesSent

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Somente Leitura
Parâmetro N1	-1 (menos um)
Parâmetro N2	0 (zero)
Parâmetro N3	0 (zero)
Parâmetro N4	1000
Configuração por String	IO.Stats.Total.BytesSent

Este Tag retorna a quantidade de bytes enviados desde que um Driver foi carregado.

IO.Stats.Total.ConnectionCount

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Somente Leitura
Parâmetro N1	-1 (menos um)
Parâmetro N2	0 (zero)
Parâmetro N3	0 (zero)
Parâmetro N4	1004
Configuração por String	IO.Stats.Total.ConnectionCount

Este Tag retorna a quantidade de conexões que um Driver já estabeleceu, com sucesso, desde que foi carregado.

IO.Stats.Total.TimeConnectedSeconds

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Somente Leitura
Parâmetro N1	-1 (menos um)
Parâmetro N2	0 (zero)
Parâmetro N3	0 (zero)
Parâmetro N4	1002
Configuração por String	IO.Stats.Total.TimeConnectedSeconds

Este Tag retorna o número de segundos que um Driver permaneceu conectado desde que foi carregado.

IO.Stats.Total.TimeDisconnectedSeconds

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Somente Leitura
Parâmetro N1	-1 (menos um)
Parâmetro N2	0 (zero)
Parâmetro N3	0 (zero)
Parâmetro N4	1003
Configuração por String	IO.Stats.Total.TimeDisconnectedSeconds

Este Tag retorna o número de segundos que um Driver permaneceu desconectado desde que foi carregado.

Propriedades

Atualmente, não existem propriedades definidas especificamente para mostrar as estatísticas das Interfaces de Comunicação em tempo de execução.

Configuração da Interface Ethernet

Esta seção contém informações sobre a configuração dos **Tags de Comunicação** e das **Propriedades** da Interface **Ethernet**.

Tags de Comunicação

Tags da Interface Ethernet (N2/B2 = 4)

Os Tags descritos a seguir permitem controlar e identificar a Interface **Ethernet** em tempo de execução e também são válidos quando a Interface **RAS** está selecionada.

IMPORTANTE
Estes Tags estão disponíveis **SOMENTE** enquanto um Driver está em modo **Online**.

IO.Ethernet.IPSelect

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Leitura ou Escrita
Parâmetro N1	-1 (menos um)
Parâmetro N2	0 (zero)
Parâmetro N3	4 (quatro)
Parâmetro N4	0 (zero)
Configuração por String	IO.Ethernet.IPSelect

Indica o endereço IP ativo. Os valores possíveis são **0**: O endereço principal de IP está selecionado, **1**: O primeiro endereço IP alternativo ou de *backup* está selecionado, **2**: O segundo endereço IP alternativo ou de *backup* está selecionado ou **3**: O terceiro endereço IP alternativo ou de *backup* está selecionado.

Se a Interface **Ethernet** ou **RAS** está conectada, este Tag indica qual dos quatro endereços IP configurados está em uso. Se a Interface está desconectada, este Tag indica qual endereço IP é usado primeiro na próxima tentativa de conexão.

Durante o processo de conexão, se o endereço IP ativo não está disponível, a Interface de Comunicação tenta conectar-se usando o outro endereço IP. Se a conexão com o endereço IP alternativo funcionar, este é configurado como o endereço IP ativo (*switchover* automático).

Para forçar um *switchover* manual, escreva valores de 0 (zero) a três (3) neste Tag. Isto força a reconexão com o endereço IP especificado (**0**: Endereço principal, **1, 2, 3**: Endereços alternativos) se um Driver está atualmente conectado. Se um Driver está desconectado, este Tag configura o endereço IP ativo para a próxima tentativa de conexão.

IO.Ethernet.IPSwitch

Tipo de Tag	Tag de Comunicação

Tipo de Acesso	Somente Escrita
Parâmetro N1	-1 (menos um)
Parâmetro N2	0 (zero)
Parâmetro N3	4 (quatro)
Parâmetro N4	1 (um)
Configuração por String	IO.Ethernet.IPSwitch

Qualquer valor escrito neste Tag força um *switchover* manual. Se o endereço principal de IP está ativo, então o primeiro endereço IP alternativo ou de *backup* é ativado, e assim por diante passando por todos os endereços IP alternativos e voltando para o endereço principal até estabelecer uma conexão.

Se um Driver está desconectado, este Tag configura o endereço IP ativo para a próxima tentativa de conexão.

IO.Ethernet.SocketState

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Somente Leitura
Parâmetro N1	-1 (menos um)
Parâmetro N2	0 (zero)
Parâmetro N3	4 (quatro)
Parâmetro N4	2 (dois)
Configuração por String	IO.Ethernet.SocketState

A propriedade **Value** deste Tag corresponde a estados do *socket* em um mapa de bits:

- **Bit 0**: 0 (zero, não está em escuta) ou 1 (um, em escuta)
- **Bit 1**: 0 (zero, desconectado) ou 1 (um, conectado)

Propriedades

Estas propriedades controlam a configuração da Interface **Ethernet**.

NOTA

A Interface **Ethernet** também é usada pela Interface **RAS**.

IO.Ethernet.AcceptConnection

◩ Configure em Falso se um Driver não deve aceitar conexões externas, ou seja, se um Driver se comporta como mestre, ou configure em Verdadeiro para habilitar a recepção de conexões, ou seja, se um Driver se comporta como escravo.

IO.Ethernet.BackupEnable[2,3]

◩ Configure em Verdadeiro para habilitar o endereço IP alternativo ou de *backup*. Se a tentativa de reconectar com o endereço IP principal falhar, um Driver tenta utilizar um endereço IP alternativo ou de *backup*. Configure em Falso para

desabilitar a utilização.

IO.Ethernet.BackupIP[2,3]

Endereço IP alternativo ou de *backup* de um equipamento remoto. Pode-se utilizar tanto o endereço numérico como o nome de *host* de um equipamento, como por exemplo "192.168.0.7" ou "SERVER2".

IO.Ethernet.BackupLocalPort[2,3]

Número da porta local a ser utilizada na conexão ao endereço IP alternativo ou de *backup* de um equipamento remoto. Usado apenas se a propriedade **IO.Ethernet.BackupLocalPortEnable** está configurada para Verdadeiro.

IO.Ethernet.BackupLocalPortEnable[2,3]

Configure em Verdadeiro para forçar o uso de uma porta local específica ao conectar ao endereço IP alternativo ou de *backup* ou configure em Falso para utilizar qualquer porta local disponível.

IO.Ethernet.BackupPort[2,3]

Número da porta do endereço IP alternativo ou de *backup* de um equipamento remoto, usado juntamente com a propriedade **IO.Ethernet.BackupIP**.

IO.Ethernet.IPFilter

Lista de endereços IPv4 ou IPv6 separados por vírgula, que define de quais endereços um Driver aceita ou bloqueia conexões. Pode-se utilizar asteriscos, como por exemplo "192.168.*.*", ou intervalos, como por exemplo "192.168.0.41-50", em qualquer parte dos endereços IP. Para bloquear um endereço IP ou um intervalos de endereços IP, use o caractere til ("~") no início do endereço, conforme os exemplos a seguir:

- **192.168.0.24**: Aceita apenas conexões do endereço IPv4 192.168.0.24
- **192.168.0.41-50**: Aceita conexões dos endereços IPv4 no intervalo entre 192.168.0.41 e 192.168.0.50
- **192.168.0.***: Aceita conexões dos endereços IPv4 no intervalo entre 192.168.0.0 e 192.168.0.255
- **fe80:3bf:877::*:* (expande para fe80:03bf:0877:0000:0000:0000:*:*)**: Aceita conexões de endereços IPv6 no intervalo entre fe80:03bf:0877:0000:0000:0000:0000:0000 e fe80:03bf:0877:0000:0000:0000:ffff:ffff
- **192.168.0.10, 192.168.0.15, 192.168.0.20**: Aceita conexões dos endereços IPv4 192.168.0.10, 192.168.0.15 e 192.168.0.20
- **~192.168.0.95, 192.168.0.***: Aceita conexões dos endereços IPv4 no intervalo entre 192.168.0.0 e 192.168.0.255, exceto o endereço IPv4 192.168.0.95

Quando um Driver recebe uma tentativa de conexão, a lista de filtros é percorrida sequencialmente da esquerda para a direita, procurando por uma autorização ou bloqueio específico para o endereço IP de onde veio a conexão. Se nenhum elemento da lista corresponde ao endereço IP, a autorização ou bloqueio são ditados pelo último elemento da lista:

- Se o último elemento da lista é uma autorização, como por exemplo "192.168.0.24", então todos os endereços IP não encontrados na lista são bloqueados
- Se o último elemento da lista é um bloqueio, como por exemplo "~192.168.0.24", então todos os endereços IP não encontrados na lista são autorizados

Se um endereço IP aparece em mais de um filtro da lista, o filtro mais à esquerda tem precedência. Por exemplo, no caso de "~192.168.0.95, 192.168.0.*", o endereço IP 192.168.0.95 se encaixa nas duas regras, mas a regra que vale é a mais à esquerda, "~192.168.0.95", e portanto o endereço IP é bloqueado.

Quando o **IOKit** bloqueia uma conexão, a mensagem "Blocked incoming socket connection from {IP}!" é logada.

No caso de conexões UDP em modo escuta em *broadcast*, em que um Driver pode receber pacotes de diferentes endereços IP, o bloqueio ou permissão é realizado a cada pacote recebido. Se um pacote é recebido de um endereço IP bloqueado, a mensagem "Blocked incoming packet from {IP} (discarding {N} bytes)!" é logada.

IO.Ethernet.ListenIP

A Endereço IP da interface local de rede por onde um Driver efetua e aceita conexões. Deixe esta propriedade vazia para efetuar e aceitar conexões por qualquer interface local de rede.

IO.Ethernet.ListenPort

9 Número da porta IP utilizada por um Driver para escutar conexões.

IO.Ethernet.MainIP

A Endereço IP de um equipamento remoto. Pode-se utilizar tanto o endereço numérico como o nome de *host* de um equipamento, como por exemplo "192.168.0.7" ou "SERVER2".

IO.Ethernet.MainLocalPort

9 Número da porta local a ser utilizada na conexão ao endereço IP principal de um equipamento remoto. Este valor é usado apenas se a propriedade **IO.Ethernet.MainLocalPortEnable** é igual a Verdadeiro.

IO.Ethernet.MainLocalPortEnable

◩ Configure em Verdadeiro para forçar o uso de uma porta local específica ao conectar ao endereço IP principal ou configure em Falso para utilizar qualquer porta local disponível.

IO.Ethernet.MainPort

9 Número da porta IP em um equipamento remoto, usado em conjunto com a propriedade **IO.Ethernet.MainIP**.

IO.Ethernet.PingEnable

◩ Configure em Verdadeiro para habilitar o envio de um comando **ping** para o endereço IP de um equipamento remoto, antes de tentar conectar-se ao *socket*. O *time-out* de conexão do *socket* não pode ser controlado, por isto o envio de um comando **ping** antes de conectar-se é uma maneira rápida de detectar se a conexão vai falhar. Configure em Falso para desabilitar o comando **ping**.

IO.Ethernet.PingTimeoutMs

9 Tempo de espera por uma resposta de um comando **ping**, em milissegundos.

IO.Ethernet.PingTries

9 Número máximo de tentativas de comandos **ping**. O valor mínimo é 1 (um), incluindo o primeiro comando **ping**.

IO.Ethernet.ShareListenPort

Configure em Verdadeiro para compartilhar a porta de escuta com outros Drivers e processos ou Falso para abrir a porta de escuta em modo exclusivo. Para compartilhar uma porta de escuta com sucesso, todos os Drivers e processos envolvidos devem abrir esta porta em modo compartilhado. Quando uma porta de escuta é compartilhada, cada nova conexão é distribuída para um dos processos que estão escutando. Desta forma, se um Driver Escravo só suporta uma conexão por vez, pode-se utilizar várias instâncias deste Driver escutando na mesma porta, portanto simulando um Driver com suporte a múltiplas conexões.

IO.Ethernet.SupressEcho

Configure em Verdadeiro para eliminar o eco presente em uma comunicação. O eco é a recepção indesejada de uma cópia exata de todos os pacotes de dados que um Driver enviou para um equipamento.

IO.Ethernet.Transport

Define o protocolo de transporte. Os valores possíveis são **T ou TCP**: Utiliza o protocolo TCP/IP ou **U ou UDP**: Utiliza o protocolo UDP/IP.

IO.Ethernet.UseIPv6

Configure em Verdadeiro para utilizar endereços IPv6 em todas as conexões Ethernet ou configure em Falso para utilizar endereços IPv4 (padrão).

Configuração da Interface Modem

Esta seção contém informações sobre a configuração dos **Tags de Comunicação** e das **Propriedades** da Interface **Modem** (TAPI).

Tags de Comunicação

Tags da Interface Modem (N2/B2 = 3)

Os Tags descritos a seguir permitem controlar e diagnosticar a Interface **Modem** (TAPI) em tempo de execução.

IMPORTANTE
Estes Tags estão disponíveis **SOMENTE** enquanto um Driver está em modo **Online**.

IO.TAPI.ConnectionBaudRate

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Somente Leitura
Parâmetro N1	-1 (menos um)
Parâmetro N2	0 (zero)
Parâmetro N3	3 (três)
Parâmetro N4	5 (cinco)
Configuração por String	IO.TAPI.ConnectionBaudRate

Indica o valor de *baud rate* da conexão atual. Se o modem não está conectado, retorna o valor 0 (zero).

IO.TAPI.Dial

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Somente Escrita
Parâmetro N1	-1 (menos um)
Parâmetro N2	0 (zero)
Parâmetro N3	3 (três)
Parâmetro N4	1 (um)
Configuração por String	IO.TAPI.Dial

Escreva qualquer valor neste Tag para forçar a Interface **Modem** a iniciar uma chamada. Este comando é assíncrono, apenas iniciando o processo de chamada. Pode-se monitorar o Tag **IO.TAPI.IsModemConnected** para detectar quando uma chamada é estabelecida.

IO.TAPI.HangUp

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Somente Escrita
Parâmetro N1	-1 (menos um)
Parâmetro N2	0 (zero)
Parâmetro N3	3 (três)
Parâmetro N4	4 (quatro)
Configuração por String	IO.TAPI.HangUp

Qualquer valor escrito neste Tag desliga a chamada atual.

NOTA

Use este comando apenas quando gerenciar a camada física manualmente ou ao explicitamente tentar forçar um Driver a reiniciar a comunicação. Se a camada física está configurada para reconexão automática, um Driver imediatamente tenta restabelecer a conexão.

IO.TAPI.IsModemConnected

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Somente Leitura
Parâmetro N1	-1 (menos um)
Parâmetro N2	0 (zero)
Parâmetro N3	3 (três)
Parâmetro N4	3 (três)
Configuração por String	IO.TAPI.IsModemConnected

Este Tag indica o estado da conexão do modem. Os valores possíveis são **0**: O modem não está conectado, mas pode estar realizando ou recebendo uma chamada externa ou **1**: O modem está conectado e um Driver completou ou recebeu uma chamada externa com sucesso. Enquanto está neste estado, a camada física consegue enviar ou receber dados.

IO.TAPI.IsModemConnecting

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Somente Leitura
Parâmetro N1	-1 (menos um)
Parâmetro N2	0 (zero)
Parâmetro N3	3 (três)
Parâmetro N4	6 (seis)
Configuração por String	IO.TAPI.IsModemConnecting

Este Tag indica o estado de conexão do modem, com mais detalhes do que o Tag **IO.TAPI.IsModemConnected**. Os valores possíveis são **0**: O modem não está conectado, **1**: O modem está conectando, ou seja, realizando ou recebendo uma chamada externa, **2**: O modem está conectado. Enquanto está neste estado, a camada física consegue enviar ou receber dados ou **3**: O modem está desconectando a chamada atual.

IO.TAPI.ModemStatus

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Somente Leitura
Parâmetro N1	-1 (menos um)
Parâmetro N2	0 (zero)
Parâmetro N3	3 (três)
Parâmetro N4	2 (dois)
Configuração por String	IO.TAPI.ModemStatus

Retorna uma **String** com o estado atual do modem. Os valores possíveis são os seguintes:

- **"No status!"**: A Interface **Modem** ainda não foi aberta ou já foi fechada
- **"Modem initialized OK!"**: A Interface **Modem** foi inicializada com sucesso
- **"Modem error at initialization!"**: Um Driver não conseguiu inicializar a linha do modem. Confira o arquivo de log deste Driver para mais detalhes
- **"Modem error at dial!"**: Um Driver não conseguiu começar ou aceitar uma chamada
- **"Connecting..."**: Um Driver iniciou uma chamada com sucesso, e está atualmente processando esta chamada
- **"Ringing..."**: Indica que o modem está recebendo uma chamada externa, mas ainda não a aceitou
- **"Connected!"**: Um Driver conectou-se com sucesso, ou seja, completou ou aceitou uma chamada externa
- **"Disconnecting..."**: Um Driver está desligando a chamada atual
- **"Disconnected OK!"**: Um Driver desligou a chamada atual
- **"Error: no dial tone!"**: Um Driver abortou a chamada porque o sinal de linha disponível não foi detectado
- **"Error: busy!"**: Um Driver abortou a ligação porque a linha estava ocupada
- **"Error: no answer!"**: Um Driver abortou a chamada porque não recebeu resposta do outro modem
- **"Error: unknown!"**: A chamada atual foi abortada por um erro desconhecido

IO.TAPI.PhoneNumber

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Leitura ou Escrita
Parâmetro N1	-1 (menos um)
Parâmetro N2	0 (zero)
Parâmetro N3	3 (três)
Parâmetro N4	0 (zero)
Configuração por String	IO.TAPI.PhoneNumber

Este Tag é uma **String** que lê ou modifica o número do telefone utilizado pelo Tag **IO.TAPI.Dial**. Ao modificar este Tag, o novo valor é usado apenas no próximo comando **Dial**.

Propriedades

Estas propriedades controlam a configuração da Interface **Modem** (TAPI).

IO.TAPI.AcceptIncoming

Configure em Falso se o modem não pode aceitar chamadas externas, ou seja, se um Driver se comporta como mestre, e configure em Verdadeiro para habilitar a recepção de chamadas, ou seja, se um Driver se comporta como escravo.

IO.TAPI.ModemID

9 É o número de identificação do modem. Este ID é criado pelo Windows e é usado internamente para identificar o modem dentro de uma lista de equipamentos instalados no computador. Este ID pode não permanecer válido caso o modem seja reinstalado ou a aplicação seja executada em outro computador.

NOTA

Recomenda-se que esta propriedade seja configurada em 0 (zero), indicando que um Driver deve utilizar o primeiro modem disponível.

IO.TAPI.PhoneNumber

A O número de telefone utilizado em comandos **Dial**, como por exemplo "0w01234566", em que o caractere "w" força o modem a esperar por um sinal de chamada.

Configuração da Interface RAS

Esta seção contém informações sobre a configuração dos **Tags de Comunicação** e das **Propriedades** da Interface **RAS**.

Tags de Comunicação

Tags da Interface RAS (N2/B2 = 5)

Atualmente, não existem Tags definidos especificamente para gerenciar a Interface **RAS** em tempo de execução.

Propriedades

Estas propriedades controlam a configuração da Interface **RAS**.

NOTA

A Interface **RAS** utiliza a Interface **Ethernet**, que por este motivo também deve ser configurada.

IO.RAS.ATCommand

A Comando **AT** a ser enviado através do *socket* para forçar um equipamento RAS a realizar uma ligação usando o canal RAS atual, como por exemplo "ATDT6265545".

IO.RAS.CommandTimeoutSec

9 Tempo de espera pela mensagem **CONNECT** em resposta a um comando **AT**, em segundos.

Configuração da Interface Serial

Esta seção contém informações sobre a configuração dos **Tags de Comunicação** e das **Propriedades** da Interface **Serial**.

Tags de Comunicação

Tags da Interface Serial (N2/B2 = 2)

Atualmente, não existem Tags definidos especificamente para gerenciar a Interface **Serial** em tempo de execução.

Propriedades

Estas propriedades controlam a configuração da Interface **Serial**.

IO.Serial.Baudrate

Especifica a taxa de *bauds* da porta serial, como por exemplo 9600.

IO.Serial.CTSTimeoutMs

Tempo de espera pelo sinal **CTS**, em milissegundos. Após o sinal **RTS** ser ligado (**ON**), um temporizador é iniciado para esperar pelo sinal **CTS**. Se este temporizador expira, um Driver aborta o envio de bytes através da porta serial. Disponível apenas quando a propriedade **IO.Serial.RTS** está configurada com o valor **Toggle** e a propriedade **IO.Serial.WaitCTS** está configurada em Verdadeiro.

IO.Serial.DataBits

Especifica o número de bits de dados para a configuração da porta serial. Os valores possíveis são **5**: Cinco bits de dados, **6**: Seis bits de dados, **7**: Sete bits de dados ou **8**: Oito bits de dados.

IO.Serial.DelayAfterMs

Número de milissegundos de atraso após o último byte ter sido enviado através da porta serial, mas antes de desligar (**OFF**) o sinal **RTS**. Disponível apenas quando a propriedade **IO.Serial.RTS** está configurada com o valor **Toggle** e a propriedade **IO.Serial.WaitCTS** está configurada em Falso.

IO.Serial.DelayBeforeMs

Número de milissegundos de atraso após o sinal **RTS** ter sido ligado (**ON**), mas antes dos dados serem enviados. Disponível apenas quando a propriedade **IO.Serial.RTS** está configurada com o valor **Toggle** e a propriedade **IO.Serial.WaitCTS** está configurada em Falso.

IO.Serial.DTR

Indica o modo como um Driver lida com o sinal **DTR**. Os valores possíveis são **OFF**: Sinal **DTR** sempre desligado ou **ON**: Sinal **DTR** sempre ligado.

IO.Serial.InterbyteDelayUs

Tempo de espera, em milissegundos (1/1000000 de um segundo), para cada dois bytes enviados pela Interface **Serial**.

IO.Serial.InterframeDelayMs

Tempo de espera, em milissegundos, antes de enviar um pacote após o último pacote enviado ou recebido.

IO.Serial.Parity

Especifica a paridade para a configuração da porta serial. Os valores possíveis são **E ou Even**: Paridade par, **N ou None**: Sem paridade, **O ou Odd**: Paridade ímpar, **M ou Mark**: Paridade de marca ou **S ou Space**: Paridade de espaço.

IO.Serial.Port

Número da porta serial local. Os valores possíveis são **1**: Utiliza a porta COM1, **2**: Utiliza a porta COM2, **3**: Utiliza a porta COM3 ou **n**: Utiliza a porta COM*n*.

IO.Serial.RTS

Indica como um Driver lida com o sinal **RTS**. Os valores possíveis são **OFF**: Sinal **RTS** sempre desligado, **ON**: Sinal **RTS** sempre ligado ou **Toggle**: Liga (**ON**) o sinal **RTS** quando está transmitindo dados e desliga (**OFF**) o sinal **RTS** quando não está transmitindo dados.

IO.Serial.StopBits

Especifica o número de bits de parada para a configuração da porta serial. Os valores possíveis são **1**: Um bit de parada, **2**: Um bit e meio de parada ou **3**: Dois bits de parada.

IO.Serial.SupressEcho

Utilize um valor diferente de 0 (zero) para habilitar a supressão de eco ou 0 (zero) para desabilitá-la.

IO.Serial.WaitCTS

Configure em Verdadeiro para forçar um Driver a esperar pelo sinal **CTS** antes de enviar bytes quando o sinal **RTS** está ligado (**ON**). Disponível apenas quando a propriedade **IO.Serial.RTS** está configurada com o valor **Toggle**.

Histórico de Revisões do Driver

VERSÃO	DATA	AUTOR	COMENTÁRIOS
4.0.25	21/09/2026	M. Ludwig	• Adicionada criptografia TLS por certificado apenas a algoritmos fortes (TLS 1.3 e TLS 1.2 com ECDHE + AEAD) (*Case 39683*).
4.0.23	27/08/2026	M. Ludwig	• Driver atualizado para a biblioteca **IOKit** versão **3.0** e Visual Studio 2022 (*Case 37983*).
4.0.22	28/03/2025	M. Salvador	• Agora é possível declarar a lista de Estações através de números ou faixas de números, separados por vírgula, como por exemplo "3,5-9". Neste caso são criados os endereços 3, 5, 6, 7, 8 e 9 (*Case 37272*).
4.0.19	08/04/2024	F. Englert	• Este Driver agora valida dados do tipo **MEs** recebidos de uma aplicação, ajustando para o limite e atribuindo o *flag* **OV** igual a 1 (um) caso estejam fora do limite válido (*Case 35807*).
4.0.18	22/12/2023	M. Salvador	• Corrigido um eventual GPF (*General Protection Failure*) do protocolo **IEC 870-5-104** no modo **Mestre** ao perder

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			a comunicação ou finalizar este Driver (*Case 35352*).
4.0.16	16/11/2023	M. Salvador	• Corrigido um eventual GPF (*General Protection Failure*) ao terminar um comando de aplicação (*Case 35170*).
4.0.15	08/05/2023	M. Salvador	• Criada a escrita de um Tag com o parâmetro *N2* igual a 100, que força uma Interrogação Geral para a Estação informada no parâmetro *N1* (*Case 34153*).
4.0.14	24/03/2023	M. Salvador	• Criada uma opção para descartar a ASDU135 para reiniciar a contagem de tempo de inatividade de um canal (*Case 33908*).
4.0.13	22/03/2023	F. Englert	• Corrigido um problema no modo **Mestre**, que podia não enviar uma Interrogação Geral (**GI**) em caso de reconexão rápida (*Case 34029*).
4.0.12	21/03/2023	F. Englert	• Corrigido um problema que reduzia a performance de envio de dados drasticamente quando há muitos dados a enviar (*Case 34018*).
4.0.11	12/01/2023	M. Salvador	• Corrigido o envio do **TESTDT** quando inativo no modo **Mestre** (*Case 33714*). • Reimplementados os Tags **InterrogationGroup**, com o parâmetro *N2* entre 100 e 115, que existiam na versão **3.0** e permitiam o controle granular do grupo e a frequência de interrogação (Case 33702).
4.0.9	23/11/2022	F. Englert	• Este Driver no modo **Mestre** agora envia comandos **GI** corretamente quando configurado para não esperar o **M_EI_NA** ou se não está configurado com uma lista fixa de Estações (*Case 33414*).
4.0.8	23/11/2022	F. Englert	• Implementado um Tag com o parâmetro *N2* igual a 993, que retorna o estado do

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			link layer para a Estação indicada no parâmetro *N1* (*Case 33508*).
4.0.6	20/10/2022	M. Salvador	• Valores inválidos em objetos dos tipos **DIQ** e **SEP** agora não invalidam um comando **GI** (*Case 33329*).
4.0.5	02/08/2022	M. Salvador	• Corrigida a leitura do Tag **Reset Process** no modo **Escravo** (Case 33036). • Criada a opção **Map Overflow bit to Uncertain Quality** no modo **Mestre** (*Case 32653*).
4.0.4	01/02/2021	F. Englert	• Este Driver, no modo **Escravo**, agora consegue enviar dados de totais integrados (ASDUS **15**, **16** e **37**) (*Case 30396*).
4.0.2	13/07/2020	F. Englert	• Corrigido um problema neste Driver, no modo **Escravo**, que às vezes enviava a um **Mestre** cópias de pacotes recebidos (*Case 29139*). • Este Driver agora suporta múltiplas Estações nos modos **Mestre** e **Escravo** (*Case 28195*).
4.0.1	03/01/2020	F. Englert	• Criada uma versão independente para o protocolo **IEC 870-5-104** nos modos **Mestre** e **Escravo**. Este Driver do protocolo **IEC 870-5-101** ainda continua suportado na versão **3.0** (101/104) (*Case 28186*).
3.0.4	27/08/2019	M. Ludwig	• Driver portado para o Visual Studio 2017 (*Case 27164*).
3.0.1	20/06/2017	F. Englert	• Removidos os protocolos **102**, **ZIV** e **Procome**, agora tratados por Drivers específicos (*Case 22924*). • Alterado o nome do arquivo deste Driver para IEC870-101-104.dll, que agora utiliza o identificador de proteção de Drivers do

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			Elipse Power e a licença é intercambiável com outros Drivers deste mesmo grupo (*Case 13800*).
2.22.80	22/05/2017	F. Englert	• Implementada a opção **Disconnect if General Interrogation returns empty results** (*Case 22256*).
2.22.77	11/05/2017	F. Englert	• Implementada a opção **Enable Callbacks** também para os protocolos **IEC 870-5-101** no modo **Escravo Balanceado**, **IEC 870-5-101** no modo **Não-Balanceado** e **IEC 870-5-104** (*Case 22652*). • Os Comandos analógicos **SEn**, **SEs** e **SEf** de **Select** ou **Execute** com qualificadores **ShortPulse**, **LongPulse** e **PersistentOutput** agora são enviados corretamente (*Case 22662*).
2.22.76	12/04/2017	F. Englert	• Este Driver agora suporta a opção **IP Filter**, que permite indicar de quais endereços TCP/IP este Driver aceita ou rejeita conexões (*Case 22540*).
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 comunica. Este Driver converte as estampas de tempo locais de um computador para o fuso horário de uma Estação e vice-versa (*Case 21621*).
2.22.74	30/09/2016	F. Englert	• O Tag Bloco de recebimento de Comandos **SD**, **DC** e **RC** deste Driver dos protocolos **IEC 870-5-101** e **IEC 870-5-104** no modo **Escravo** agora retorna um *array* de **Variants** ao invés de um *array* de Inteiros, permitindo a manipulação em scripts de uma aplicação (*Case 21588*).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
2.22.72	01/09/2016	F. Englert	• Atualizada a documentação com as implementações da versão **2.22**, somente em português (*Case 21167*).
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** (*Case 21385*).
2.22.63	30/03/2016	F. Englert	• Adicionada uma propriedade para configurar o **Originator Address** 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 dois (*Case 21126*).
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** (*Case 20685*).
2.22.60	11/02/2016	F. Englert	• Este Driver agora marca corretamente como *LinkActive* as Estações do protocolo **IEC 870-5-101** no modo **Mestre Não-Balanceado** após a correta inicialização (*Case 20531*). • Este Driver do protocolo **IEC 870-5-101** no modo **Mestre Balanceado** agora responde de forma imediata aos Comandos **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 **IEC 870-5-101** no modo **Escravo Não-Balanceado** (*Case 20515*).
2.22.58	03/02/2016	F. Englert	• Implementado o protocolo **Procome** (*Case 19889*).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
2.22.55	04/12/2015	F. Englert	• Adicionada uma opção para controlar o uso do flag **SU** (horário de verão), **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 dois intervalos de dados para envio cíclico (*Case 17708*).
2.22.53	13/11/2014	F. Englert	• Este Driver no modo **Mestre** não mais ocupa 100% de CPU caso a opção **Slave Polling Rate** seja configurada para zero (*Case 15377*).
2.22.52	29/10/2014	F. Englert	• Este Driver dos protocolos **IEC 870-5-101** e **IEC 870-5-104** no modo **Escravo** agora descartam os dados de Interrogação Geral pendentes (*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 dos protocolos **IEC 870-5-104** ou **IEC 870-5-101** no modo **Escravo Balanceado** logo após 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*).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
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*). • Implementado suporte a *callbacks* neste Driver no modo **Mestre** na opção **Enable Callbacks** da aba **Properties** (*Case 9178*).
2.22.45	03/04/2013	F. Englert	• Corrigido um problema no tratamento de pacotes deste Driver do protocolo **IEC 870-5-101** no modo **Escravo Balanceado**, que podia descartar incorretamente uma resposta de um **Mestre** caso este enviasse um novo pedido antes de enviar a resposta (*Case 13942*).
2.22.44	22/03/2013	F. Englert	• Corrigido o empacotamento dos ASDUs 5 (cinco), 7 (sete), 13, 15 e 21, que poderiam ser enviados utilizando uma sequência de objetos, opção não suportada pelo padrão do protocolo (*Case 13898*).
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 quando é detectado algum erro de sequência nos pacotes (*Case 13808*).
2.22.41	07/11/2012	F. Englert	• Corrigido um vazamento de memória que ocorria sempre que este Driver recebia dados e os

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			colocava na memória cache interna (*Case 13454*).
2.22.37	15/02/2012	F. Englert	• Adicionada uma 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** (*End of Initialization*) quando um **Mestre** conecta (*Case 12761*).
2.22.36	18/11/2011	F. Englert	• Este Driver do protocolo **IEC 870-5-104** no modo **Escravo** agora envia a ASDU **70** (**M_EI_NA_1**) quando a comunicação é iniciada (*Case 12575*). • Os bits **BL** e **OV** agora são mapeados no campo **Limit** da qualidade, ou seja, os dois bits menos significativos (*Case 12569*).
2.22.35	13/05/2011	F. Englert	• Criada uma opção na aba **Properties** dos protocolos **Mestres**, **Don't wait ACTTERM for Execute commands** (*Case 12229*).
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 (*Case 12161*). • Este Driver no modo **Escravo** agora permite que o usuário configure uma banda morta para envio de valores analógicos na opção **Analog Deadband (% of previous value)** da aba **Properties** (*Case 12109*).
2.22.33	11/02/2011	F. Englert	• Este Driver do protocolo **IEC 870-5-101** no modo **Mestre Balanceado** respondia o bit **DIR** com o valor 0 (zero) na mensagem **RESPOND_STATUS**, quando deveria responder o bit **DIR** com o valor 1 (um).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			Adicionada a opção **DIR (Physical Transmission Direction)**, que controla o comportamento do bit **DIR** no pacote do protocolo **IEC 870-5-101** no modo **Balanceado**, tanto no modo **Mestre** quanto no modo **Escravo** (*Case 10174*). • Corrigida a Causa de Transmissão utilizada por este Driver do protocolo **IEC 870-5-101** no modo **Escravo** ao retornar dados de Comandos, como por exemplo uma resposta de Interrogação Geral, por Classe um (*Case 12057*). • O protocolo **IEC 870-5-101** nos modos **Escravo Balanceado** e **Não-Balanceado** deste Driver 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 um **Test Command** sempre que a camada de aplicação permanecer *n* segundos sem receber uma mensagem de um **Escravo**. Também foi removido o Tag com o parâmetro *N2* igual a 1104, que enviava um Comando de Teste sempre que um 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** da janela de configuração deste Driver no modo **Escravo** (*Case 12045*).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
2.22.31	18/01/2011	F. Englert	• Adicionada uma opção **Link Layer Starts** na aba **Properties** da janela de configuração deste Driver, que permite definir o estado inicial do *link layer*, 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 flag **IV** 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 **Remover timestamp dos dados recebidos** foi removida, pois esta operação é realizada automaticamente. • Implementada a opção de *local override* 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** (*Case 11732*). • Adicionada uma propriedade a este Driver,

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			IEC.MaxPacketSize, que permite configurar o tamanho máximo dos pacotes montados por este Driver, 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 flag **SB** (*substituted*) 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	• Adicionada uma 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 do *Slave Address* (*Case 11634*).
2.22.23	18/08/2009	F. Englert	• Quando o número de octetos da Causa de Transmissão é igual a 2 (dois), este Driver considerava 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 0 (zero) no

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			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 sub-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 neste Driver do protocolo **IEC 870-5-104** no modo **Mestre** nos casos em que ocorre uma desconexão e reconexão muito rápida (*Case 8039*). • Adicionada uma aba **Properties** a este Driver. Esta aba permite editar, em formato de lista, as propriedades de cada protocolo, anteriormente disponíveis na aba **IEC**, bem como as propriedades **T0**, **T1**, **T2**, **T3**, **K** e **W** do protocolo **IEC 870-5-104** (*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*).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
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, ou seja, **cot** igual a 10, 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 função para ler dados da memória cache deste Driver (*Case 9166*).
2.22.14	19/02/2008	F. Englert	• Adicionada uma opção na aba **Commands** deste Driver, disponível apenas para o modo **Mestre**, **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 no modo **Mestre** (*Case 8402*). • Este Driver no modo **Escravo** agora permite configurar *buffers* para armazenagem de eventos de classe 1 (um) e classe 2 (dois) enquanto um **Mestre** está desconectado (*Case 7658*). • Implementada uma aba **Stations** na janela de configuração deste Driver. Esta aba permite que o usuário configure se este Driver cria as Estações automaticamente, o modo antigo, ou se restringe a comunicação às Estações declaradas pelo usuário, o modo novo (*Case 7687*). • Agora este Driver do protocolo **IEC 870-5-104** no modo **Mestre** utiliza a opção **Wait M_EI_NA on activation** para controlar o comportamento da espera pelo **STARTDT.CON** (*Case 8963*).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
2.22.12	19/06/2007	F. Englert	• Corrigido o tratamento do bit **FCB**, que era incrementado antes de saber se um pacote foi enviado com sucesso ou não (*Case 8405*). • Corrigido o tratamento de Comandos **Read Interrogation Group** neste Driver no 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*). • Adicionado uma opção na aba **Commands** para permitir esperar ou não pelo **ACTTEM** da Interrogação Geral, **Wait ACTTERM** (*Case 8401*).
2.22.11	13/04/2007	F. Englert	• Agora as escritas em um Tag com o parâmetro *N2* igual a 1102 só retornam erro ao falhar a comunicação. Se um equipamento retorna um código de erro, a escrita é considerada bem-sucedida e o código de erro retorna no status desta escrita (*Case 8018*).
2.22.10	13/04/2007	F. Englert	• Corrigido o comportamento deste Driver no modo **Escravo**, que ignorava a qualidade dos dados recebidos de uma aplicação e enviava para um **Mestre** sempre com qualidade Boa (*Case 8111*). • Adicionado um Tag com o parâmetro *N2* igual a 6 (seis) neste Driver no modo **Escravo**, que permite enviar dados espontaneamente com classe 1 (um) e também na Interrogação

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			Geral, se configurado (*Case 7653*).
2.22.8	23/03/2007	F. Englert	• Corrigido o comportamento do *timer* **T2** do protocolo **IEC 870-5-104**, que poderia induzir este Driver a exceder o tempo máximo para envio de um pacote de **ACK**, ou seja, 10 segundos (*Case 8038*).
2.22.7	06/03/2007	F. Englert	• Corrigido o formato do parâmetro *N2* igual a 99, que retornava uma data no formato **Double** do **Elipse SCADA** (*Case 7954*). • Adicionada à aba **Commands** uma lista de seleção que permite selecionar o modo como este Driver no modo **Escravo** trata Comandos de Seleção ou Execução, **default (pass to application)** (*Case 5963*).
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 (*Case 7940*). • 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)** (*Case 7509*). • Implementado o suporte ao ASDU 105 (**C_RP_NA_1**, *Reset process command*) dos protocolos **IEC 870-5-101** e **IEC 870-5-104** nos modos **Mestre** e **Escravo** (*Case 7836*).
2.22.3	10/01/2007	F. Englert	• Corrigido o tamanho máximo de pacotes do protocolo **IEC 870-5-104**, que permitia gerar pacotes com até 4 (quatro) bytes a

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			mais que o permitido, ocasionando problemas na comunicação (*Case 7798*). • Este Driver agora é protegido pelo identificador 31290 (*Case 7797*). • Agora a qualidade é alterada para 216 ao gerar a 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 se este Driver estava configurado 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 deste Driver dos protocolos **IEC 870-5-101** e **IEC 870-5-102** no modo **Mestre**, que era enviado invertido (Case 7286).
2.20.1	11/08/2006	F. Englert	• Corrigido o script de exemplo para tratamento de Comandos recebidos por um **Escravo**, que não funcionava (*Case 5452*). • Agora registra-se no log os eventos que são descartados quando a memória *cache* 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 **IEC 870-5-104** no modo **Escravo** (*Case 6564*).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			• Corrigido o parâmetro **Slave polling rate**, que não tem efeito na comunicação com um **Escravo** (*Case 6570*). • Corrigido o recebimento de dados com estampa de tempo marcada como inválida (**IV** igual a um), em que os próprios dados eram marcados como inválidos e deveria apenas ser descartada a estampa de tempo (*Case 6666*). • Os modos **Mestre** e **Escravo** foram extensivamente testados e foram realizadas pequenas melhorias (*Case 6798*). • Corrigida a utilização de um *time-out* fixo de um segundo, ao invés de utilizar o tempo configurado na opção **App Layer Timeout**, quando é preciso enviar um Comando e o nível de aplicação está ocupado, como 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 **Elipse E3** versão **2.5.131** e a biblioteca **IOKit** 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 **IEC 870-5-104** no modo **Escravo**. Adicionados

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			os Comandos das ASDUs 58 (**C_SC_TA_1**), 59 (**C_DC_TA_1**), 60 (**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 foi adicionado suporte ao Comando de Teste (*Test Command*) utilizando a ASDU 107 (**C_TS_TA_1**) (*Case 5506*). • Implementadas atualizações no protocolo **ZIV** (*Case 5846*). • Modificada a leitura de ajustes de **VTCD** para se ajustar ao novo formato definido no protocolo **ZIV** (*Case 6561*). • Adicionada a opção **Modify the timestamp on received values** na aba **Interrogation** (*Case 6562*). • Adicionado suporte ao Comando de Sincronização de Relógio (*Clock Synchronization Command*, ASDU 103 ou **C_CS_NA_1**) dos protocolos **IEC 870-5-101** e **IEC 870-5-104** no modo **Escravo**. Também foi adicionada a aba **Commands** (*Case 6826*). • Adicionado suporte ao protocolo **IEC 870-5-104** no modo **Mestre** (*Case 7259*).
2.19.1	17/01/2005	F. Englert	• Adicionada uma opção para usar o **Common Address** diferente do **Link Address** (*Case 4364*). • Corrigida uma situação em que não se suportava endereços de **Escravo** maiores que 127 quando configurado para utilizar um octeto para o endereço do **Escravo**, e deveria permitir endereços entre 0 (zero) e 255 (*Case 4766*). • Corrigida uma situação em que não se reconhecia pacotes **CC1** recebidos

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			como sendo pacotes de **ACK** (*Case 5210*). • Corrigida uma situação em que os tipos de dados do protocolo **IEC 870-5-104** estavam retornando a estampa de tempo como o valor de um Tag (*Case 5211*). • Corrigido o bloco do protocolo **ZIV**, **Read Configuration Block 2**, com o parâmetro *B2* igual a 1141, que não estava funcionando (*Case 5212*). • 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 Interrogação Geral em Blocos no modo **Mestre Não-Balanceado** (*Case 4238*). • Corrigida uma situação em que, no modo **Escravo** apenas, não se reenviava o último pacote quando um **Mestre** enviava um pacote com o mesmo **FCB** anterior (*Case 4276*). • Adicionada uma opção para selecionar o tipo de resposta que se deve enviar quando se recebe um pedido de um **Mestre**, **Respond with** (*Case 3741*). • Os tipos de dados definidos no protocolo **IEC 870-5-104** agora estão disponíveis na lista de pontos pré-definidos retornados em uma Interrogação Geral (*Case 4035*).
2.17.1	09/06/2004	F. Englert	• Adicionadas retentativas no *link layer* (*Case 3937*).
2.16.1	04/05/2004	F. Englert	• Modificado o valor padrão do *time-out* de aplicação, de 4000 para 4 (quatro) segundos.

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			• 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 **IEC 870-5-104 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 **IEC**, recebidos nos modos **Escravo** ou **Balanceado** (*Case 3231*). • Adicionada a opção **Interrogation**, que permite configurar a lista de objetos retornados na Interrogação Geral, apenas no modo **Escravo** (*Case 3284*).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			• Adicionado um Tag **TEST LINK** apenas no modo **Escravo** ou **Balanceado**. Este Tag envia um Comando de Teste do *link* para um **Mestre**, retornando 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 Booleano indicando o estado **Ligado** ou **Desligado** (*Case 3214*). • Adicionada uma opção para habilitar ou desabilitar o uso do *frame* **CC1** ao invés do *frame* **ACK** no envio de Mensagens de Quitação (*acknowledgment*) do nível de enlace em modo **Balanceado** (*Case 3215*).
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 **IEC 870-5-101** no modo **Escravo Balanceado** (*Case 195*).
2.9.1	02/12/2003	F. Englert	• Este Driver no modo **Escravo** não é mais criado quando em modo **Offline** (*Case 3039*).
2.8.1	27/11/2003	F. Englert	• Implementado o protocolo **IEC 870-5-101** no modo **Escravo** (*Case 195*).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
2.7.1	18/11/2003	F. Englert	• O modo de escuta agora funciona corretamente, pois os valores recebidos em modo de escuta não eram retornados (*Case 2960*).
2.6.1	13/11/2003	F. Englert	• Erros na qualidade de um Tag agora são reportados se a conexão é perdida enquanto está em modo de escuta (*Case 2935*).
2.5.1	07/11/2003	F. Englert	• Adicionado um campo de qualidade à leitura de memória de massa, no parâmetro *B2* igual a 1189 ou 1190 (*Case 2893*).
2.4.1	05/11/2003	F. Englert	• Adicionado um Tag **Read MAE ID**, com o parâmetro *N2* igual a 1212 para este Driver no modo **Escravo** do 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 do protocolo **ZIV**.
2.1.1	29/07/2002	F. Englert	• Driver portado para a biblioteca **IOKit**. • Implementados os protocolos **IEC 870-5-102** e **IEC 870-5-102 ZIV Extensions**.

Matriz
Rua Mostardeiro, 322/Cj. 902, 1001 e 1002
90430-000 — Porto Alegre — RS
Fone: (+55 51) 3346-4699
Fax: (+55 51) 3222-6226
E-mail: elipse-rs@elipse.com.br

Filial no Paraná
Av. Sete de Setembro, 4698/1708
80240-000 — Curitiba — PR
Fone: (+55 41) 4062-5824
E-mail: elipse-pr@elipse.com.br

Filial no Rio de Janeiro
Av. José Silva de A. Neto, 200/Bl. 4/Sl. 109B
22250-044 — Rio de Janeiro — RJ
Fone: (+55 21) 2430-5912
Suporte Técnico: (+55 21) 2430-5963
E-mail: elipse-rj@elipse.com.br

Filial em São Paulo
Rua dos Pinheiros, 870/Cj. 141 e 142
05422-001 — São Paulo — SP
Fone: (+55 11) 3061-2828
Fax:(+55 11) 3086-2338
E-mail: elipse-sp@elipse.com.br

Filial em Minas Gerais
Rua Antônio de Albuquerque, 156/705
30112-010 — Belo Horizonte — MG
Fone: (+55 31) 4062-5824
E-mail: elipse-mg@elipse.com.br

Filial em Taiwan
9F., No.12, Beiping 2nd St., Sanmin Dist. 807 — Kaohsiung City — Taiwan
Fone: (+886 7) 323-8468
Fax: (+886 7) 323-9656
E-mail: evan@elipse.com.br

Consulte nosso website para informações sobre o representante do seu estado.

www.elipse.com.br
kb.elipse.com.br
forum.elipse.com.br
www.youtube.com/elipsesoftware
elipse@elipse.com.br

Gartner 2014 Cool Vendor

OPC FOUNDATION MEMBER

Gartner, Cool Vendors in Brazil 2014, April 2014.
Gartner does not endorse any vendor, product or service depicted in its research publications, and does not advise technology users to select only those vendors with the highest ratings. Gartner research publications consist of the opinions of Gartner's research organization and should not be construed as statements of fact. Gartner disclaims all warranties, expressed or implied, with respect to this research, including any warranties of merchantability of fitness for a particular purpose.

Microsoft Partner
Gold Independent Software Vendor (ISV)

Copyright © 2002-2026 Elipse Software Ltda. Todos os direitos reservados.