

# Driver ABB SPABus32

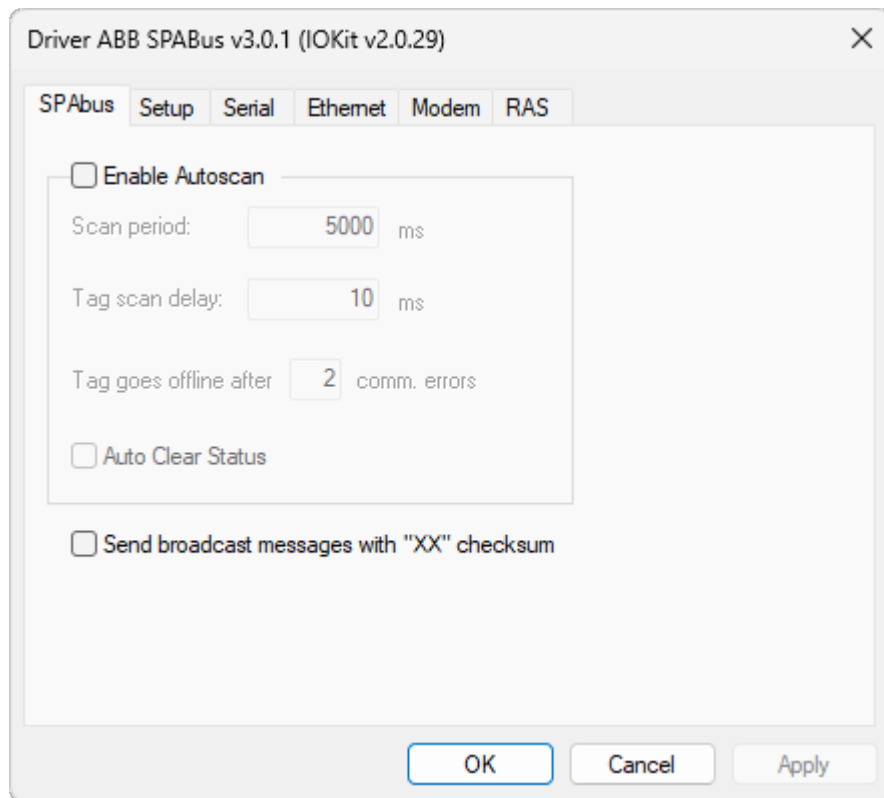
<b>Nome do Arquivo</b>	SPABus32.dll
<b>Fabricante</b>	Asea Brown Bowery Ltd.
<b>Equipamentos</b>	REF/RET/REM 541/543/545 Feeder Terminals, REX521 Relay e REJ/REU 521/525/527 Relays
<b>Protocolo</b>	SPABus Communication Protocol versão 2.5
<b>Versão</b>	3.0.2
<b>Última Atualização</b>	22/08/2025
<b>Plataforma</b>	Win32
<b>Dependências</b>	Nenhuma
<b>Leitura Superblocos</b>	Não
<b>Nível</b>	0

## Introdução

Este é o Driver ABB SPABus32, que permite a comunicação de sistemas da **Eclipse Software** com a série de produtos ABB compatíveis com o protocolo **SPABus**.

## Configuração do Driver

A partir da versão **2.0**, este Driver não utiliza mais os parâmetros **[P]** de configuração. Todas as configurações devem ser realizadas na janela de configurações deste Driver. A configuração da comunicação deve ser realizada através das abas de configuração da biblioteca **IOKit**. Para mais informações, consulte o tópico **Documentação das Interfaces de Comunicação**. A aba **SPABus** é mostrada na figura a seguir.



**Aba SPABus**

As opções disponíveis nesta aba estão descritas na tabela a seguir:

**Opções disponíveis na aba SPABus**

OPÇÃO	DESCRIÇÃO
<b>Enable Autoscan</b>	Habilita a varredura automática por eventos de categorias <b>L</b> ou <b>B</b> . Para mais informações, consulte o tópico <b>Leitura Automática de Eventos</b>
<b>Scan period</b>	Define o período de varredura, em milissegundos, para a leitura automática de eventos. Para mais informações, consulte o tópico <b>Leitura Automática de Eventos</b>
<b>Tag scan delay</b>	Define um intervalo de tempo, em milissegundos, entre a leitura de cada Tag registrado para a leitura automática, durante a varredura interna deste Driver, a fim de não sobrecarregar uma aplicação
<b>Tag goes offline after <i>n</i> comm. errors</b>	Define o número de erros de comunicação após os quais um Tag de leitura automática de eventos passa a ser considerado em estado <b>Offline</b> . Este Tag volta para o estado <b>Online</b> assim que alguma comunicação seja bem-sucedida
<b>Auto Clear Status</b>	Se esta opção está habilitada durante a captura automática de eventos, caso este Driver receba eventos de código <b>E50</b> ou <b>E51</b> , a <i>thread</i> interna de varredura envia automaticamente um comando para limpar o status de um equipamento, ou seja, escrever o valor 0 (zero) na categoria <b>C</b> , retirando um escravo da condição de erro. Os eventos <b>E50</b> e <b>E51</b> coletados são retornados normalmente para uma aplicação

OPÇÃO	DESCRIÇÃO
Send broadcast messages with "XX" checksum	Esta opção habilita o envio e recebimento de <i>frames</i> com os caracteres "XX" no lugar da soma de verificação ( <i>checksum</i> ). Esta opção pode ser habilitada para testes, mas recomenda-se deixá-la desabilitada, que é o valor padrão, em produção, pois sem a soma de verificação é possível receber <i>frames</i> inválidos, provenientes de ruídos, o que pode ocasionar a leitura de valores inválidos

## Parâmetros [N] de Endereçamento de Tags PLC

N1	Número de um escravo. Use o valor 900 para o endereço de <i>broadcast</i>
N2	Número de um canal. O valor padrão é 0 (zero)
N3	Tipo de dados. Para mais informações, consulte a tabela a seguir
N4	Número de um dado por categoria de um escravo (I, O, S, V ou M)

Para comunicação com os relés REJ-525, adicione o valor 5000 ao parâmetro *N1*. Por exemplo, para endereçar o escravo REJ-525 de número 3 (três), configure o parâmetro *N1* com o valor 5003.

O parâmetro *N3* deve ter um dos valores referentes ao tipo de função ou dado a ser executado, lido ou escrito, como descrito na tabela a seguir. Para os tipos de dados **17** e **19**, pode-se configurar o parâmetro *N4* com um tempo adicional, em milissegundos, para compensar um atraso na rede. O intervalo de sincronização é dado pelo tempo de varredura (*scan*) de um Tag.

### Tipos de dados obtidos por Tags PLC

VALOR	DESCRIÇÃO
0	Entrada de dados (I)
1	Saída de dados (O)
2	Configuração de valores (S)
3	Variáveis internas (V)
4	Dados de memória (M)*
5	Status de um escravo (C)
6	Identificação de um escravo (F)
7	Hora (D)*
8	Últimos eventos (L) no formato <b>Texto</b>
9	Data e hora (T)
10	Últimos eventos do <i>buffer</i> de <i>backup</i> (B) no formato <b>Texto</b>
11	Alarme válido (A)*
17	Tag de sincronização de relógio (data completa, somente leitura)

VALOR	DESCRIÇÃO
19	Tag de sincronização de relógio (somente os segundos, somente leitura)
101	Indica se existem dados disponíveis no <i>buffer</i> interno de eventos deste Driver
102	Indica se o respectivo Tag de leitura automática de eventos está no modo <b>Online</b>

## Parâmetros [B] de Endereçamento de Tags Bloco

Este Driver permite ler Blocos ou Elementos de Blocos de dados e escrever em Elementos de Bloco. Não é possível escrever em Blocos inteiros. A escrita deve ser sempre realizada por Elementos.

B1	Número de um escravo. Use o valor 900 para o endereço de <i>broadcast</i>
B2	Número de um canal. O valor padrão é 0 (zero)
B3	Categoria de um dado ou tipo de função. Para mais informações, consulte a tabela a seguir
B4	Número do índice do primeiro Elemento de um Tag Bloco

Para comunicação com os relés REJ-525, adicione o valor 5000 ao parâmetro B1. Por exemplo, para endereçar o escravo REJ-525 de número 3 (três), configure o parâmetro B1 com o valor 5003.

### Categorias de dados obtidos por Tags Bloco

VALOR	DESCRIÇÃO
0	Entrada de dados (I)
1	Saída de dados (O)
2	Configuração de valores (S)
3	Variáveis internas (V)
4	Dados de memória (M)
8	Últimos eventos (L). Esta opção utiliza um Bloco com os Elementos <b>Data e Hora</b> , <b>Número de um Canal</b> e <b>Código de Causa de uma Ocorrência</b>
10	Últimos eventos do <i>buffer</i> de <i>backup</i> (B). Esta opção utiliza um Bloco com os Elementos <b>Data e Hora</b> , <b>Número de um Canal</b> e <b>Código de Causa de uma Ocorrência</b>
13	Lê condições dos eventos em variáveis internas (V), com o primeiro Elemento transformado em data e hora
100	Tag de leitura automática de eventos

## Referência de Tags PLC

Esta seção contém informações sobre a configuração dos parâmetros [N] deste Driver.

## Entrada de Dados

### Leitura e Escrita

N1	Endereço de um escravo. Use o valor 900 para o endereço de <i>broadcast</i>
N2	Canal
N3	0 (zero)
N4	Número de um dado

Corresponde à categoria **I**. Permite escrever ou ler dados analógicos de um escravo, bem como estados digitais de entrada. O tipo de dados da propriedade **Value** é um ponto flutuante (*floating point*).

## Saída de Dados

### Leitura e Escrita

N1	Endereço de um escravo. Use o valor 900 para o endereço de <i>broadcast</i>
N2	Canal
N3	1 (um)
N4	Número de um dado

Corresponde à categoria **O**. Permite escrever ou ler dados analógicos ou estados digitais de saída. O tipo de dados da propriedade **Value** é um ponto flutuante (*floating point*).

## Configuração de Valores

### Leitura e Escrita

N1	Endereço de um escravo. Use o valor 900 para o endereço de <i>broadcast</i>
N2	Canal
N3	2 (dois)
N4	Número de um dado

Corresponde à categoria **S**. Permite escrever ou ler parâmetros de um escravo. O tipo de dados da propriedade **Value** é um ponto flutuante (*floating point*).

## Variáveis Internas

### Leitura e Escrita

N1	Endereço de um escravo. Use o valor 900 para o endereço de <i>broadcast</i>
N2	Canal
N3	3 (três)
N4	Número de um dado

Corresponde à categoria **V**. Permite escrever ou ler em variáveis internas de um equipamento. O tipo de dados da propriedade **Value** é um ponto flutuante (*floating point*). Os valores das variáveis internas fornecem acesso a dados ou funções, como por exemplo:

- Dados complementares sobre um processo supervisionado e os respectivos eventos
- Dados complementares relativos a um escravo ou ao funcionamento
- Controle das funções de um escravo em operação
- Programação de funções gerais em um escravo

## Dados de Memória

### Leitura e Escrita

N1	Endereço de um escravo. Use o valor 900 para o endereço de <i>broadcast</i>
N2	Canal
N3	4 (quatro)
N4	Número de um dado

Corresponde à categoria **M**. Permite escrever ou ler dados da memória do escravo. Pode incluir dados de medições ou de status armazenados na memória do equipamento. O tipo de dados da propriedade **Value** é uma **String**.

## Status de um Escravo

### Leitura e Escrita

<b>N1</b>	Endereço de um escravo. Use o valor 900 para o endereço de <i>broadcast</i>
<b>N2</b>	0 (zero)
<b>N3</b>	5 (cinco)
<b>N4</b>	0 (zero)

Corresponde à categoria **C**. O status de um escravo é representado usando-se 2 (dois) bits, em que a mensagem de resposta pode conter os valores 0 (zero), 1 (um), 2 (dois) ou 3 (três). O tipo de dados da propriedade **Value** é um inteiro sem sinal. O significado de cada bit é o seguinte:

- **Bit 0:** *Reset* de um escravo ou outra situação passível de gerar perda de dados de eventos. Quando este bit está ligado, ou seja, C é igual a 1 (um) ou 3 (três), um escravo envia como resposta às leituras de eventos o evento **E50** até que este bit seja limpo, escrevendo o valor 0 (zero) neste Tag, ou seja, C igual a 0 (zero)
- **Bit 1:** Indica *overflow* no *buffer* de eventos de um escravo. Quando este bit está ativo, ou seja, C é igual a 2 (dois) ou 3 (três), um escravo sempre envia o evento **E51** como resposta às leituras de evento, até que este bit seja limpo, escrevendo o valor 0 (zero) neste Tag, ou seja, C igual a 0 (zero). Durante este *overflow*, um escravo geralmente para de inserir novos eventos no *buffer*, e o status deve ser limpo para que se reinicie o armazenamento de eventos

## Identificação de um Escravo

### Leitura e Escrita

<b>N1</b>	Endereço de um escravo. Use o valor 900 para o endereço de <i>broadcast</i>
<b>N2</b>	0 (zero)
<b>N3</b>	6 (seis)
<b>N4</b>	0 (zero)

Corresponde à categoria **F**. Permite acesso ao código de identificação do tipo de um escravo. Este código pode ser, por exemplo, o código de produto de um escravo, ou *product code*. O tipo de dados da propriedade **Value** é uma **String**.

## Hora

### Leitura e Escrita

<b>N1</b>	Endereço de um escravo. Use o valor 900 para o endereço de <i>broadcast</i>
<b>N2</b>	0 (zero)
<b>N3</b>	7 (sete)
<b>N4</b>	0 (zero)

Corresponde à categoria **D**. Permite ler e escrever a data e a hora. O tipo de dados da propriedade **Value** é **Data e Hora**.

## Últimos Eventos

### Leitura e Escrita

<b>N1</b>	Endereço de um escravo. Use o valor 900 para o endereço de <i>broadcast</i>
<b>N2</b>	0 (zero)
<b>N3</b>	8 (oito)
<b>N4</b>	0 (zero)

Corresponde à categoria **L**. Permite ler os eventos mais recentes do *buffer* de um escravo. Quando os eventos são lidos como um Tag PLC, estes são retornados como uma **String** no formato **<timestamp em segundos> <canal>E<código do evento>**. Em caso de erro, para retentar a leitura de um evento, deve-se ler o *buffer* de *backup*. Para mais informações, consulte o Tag PLC **Último Evento do Buffer de Backup** e o Tag Bloco **Últimos Eventos do Buffer de Backup**. O tipo de dados da propriedade **Value** é uma **String**. A propriedade **Timestamp** deste Tag contém a estampa de tempo retornada por um equipamento, que corresponde ao primeiro campo da **String** da propriedade **Value**.

## Data e Hora

### Leitura e Escrita

<b>N1</b>	Endereço de um escravo. Use o valor 900 para o endereço de <i>broadcast</i>
<b>N2</b>	0 (zero)
<b>N3</b>	9 (nove)
<b>N4</b>	0 (zero)

Corresponde à categoria **T**. Permite ler e escrever valores de tempo em segundos. O tipo de dados da propriedade **Value** é um ponto flutuante (*floating point*).

## Último Evento do Buffer de Backup

### Leitura e Escrita

<b>N1</b>	Endereço de um escravo. Use o valor 900 para o endereço de <i>broadcast</i>
<b>N2</b>	0 (zero)
<b>N3</b>	10
<b>N4</b>	0 (zero)

Corresponde à categoria **B**. Permite reler um evento perdido após um erro de comunicação na leitura da categoria **L**. Para mais informações, consulte o Tag **Últimos Eventos**. Quando um evento é lido como um Tag PLC, a **String** de identificação deste evento é retornada como uma **String** no formato **<timestamp em segundos> <canal>E<código do evento>**. O tipo de dados da propriedade **Value** é uma **String**. A propriedade **Timestamp** deste Tag contém a estampa de tempo retornada por um equipamento, que corresponde ao primeiro campo da **String** da propriedade **Value**.

## Alarme Válido

### Leitura e Escrita

<b>N1</b>	Endereço de um escravo. Use o valor 900 para o endereço de <i>broadcast</i>
<b>N2</b>	0 (zero)
<b>N3</b>	11
<b>N4</b>	0 (zero)

Corresponde à categoria **A**. Durante a leitura deste Tag, um escravo responde com os eventos de alarmes válidos sem informação de tempo. O tipo de dados da propriedade **Value** é uma **String**.

## Sincronização de Relógio: Data e Hora

### Somente Leitura

<b>N1</b>	Endereço de um escravo. Use o valor 900 para o endereço de <i>broadcast</i>
<b>N2</b>	0 (zero)
<b>N3</b>	17
<b>N4</b>	Tempo adicional, em milissegundos

Cada operação de leitura neste Tag gera o envio de um comando de sincronização de relógio a um escravo, sincronizando o relógio do escravo com o relógio do computador. A categoria usada é a **D**. Para compensar um eventual atraso da rede no envio de um comando, pode-se definir um tempo adicional a ser acrescentado à hora do computador no parâmetro *N4*. A propriedade **Value** não é usada e retorna 0 (zero) em modo **Broadcast**.

# Sincronização de Relógio: Somente Segundos

## Somente Leitura

<b>N1</b>	Endereço de um escravo. Use o valor 900 para o endereço de <i>broadcast</i>
<b>N2</b>	0 (zero)
<b>N3</b>	19
<b>N4</b>	Tempo adicional, em milissegundos

Cada operação de leitura neste Tag gera o envio de um comando de sincronização de relógio a um escravo, sincronizando o relógio do escravo com o relógio do computador. A categoria usada é a **T**. O que diferencia este Tag do Tag de **Sincronização de Relógio: Data e Hora** é a categoria de dados, **T** ao invés de **D**, e o formato dos dados enviados a um equipamento, somente em segundos. O princípio de funcionamento é o mesmo. Para compensar um eventual atraso da rede no envio de um comando, pode-se definir um tempo adicional a ser acrescentado à hora do computador no parâmetro *N4*. A propriedade **Value** não é usada e retorna 0 (zero) em modo **Broadcast**.

## Eventos no Buffer

### Somente Leitura

<b>N1</b>	0 (zero)
<b>N2</b>	0 (zero)
<b>N3</b>	101
<b>N4</b>	0 (zero)

Este Tag indica se ainda existem eventos no *buffer* ainda não lidos por uma aplicação, coletados através do *scan* interno deste Driver. Este Tag pode ser usado para evitar a perda de eventos ao finalizar uma aplicação. Colocando-se este Driver em modo **Offline**, o *scan* interno é suspenso, permitindo que os Tags de leitura automática de eventos retirem dados do *buffer* interno, até que este Tag assuma o valor 0 (zero). Após o esvaziamento do *buffer* interno deste Driver, o objeto Driver de Comunicação pode ser destruído ou parado, sem perda de eventos. Para mais informações, consulte o tópico **Leitura Automática de Eventos**. Os valores possíveis para a propriedade **Value** são **0**: Sem dados no *buffer* ou **1**: Ainda existem dados no *buffer* aguardando leitura por uma aplicação.

## Evento Online

### Somente Leitura

<b>N1</b>	Endereço de um escravo. Use o valor 900 para o endereço de <i>broadcast</i>
<b>N2</b>	0 (zero)
<b>N3</b>	102
<b>N4</b>	0 (zero)

Este Tag está relacionado ao Tag Bloco **Leitura Automática de Eventos** e indica se aquele Tag Bloco está em modo **Online** ou **Offline**, permitindo diagnosticar a conexão com um escravo. Aquele Tag Bloco vai para o modo **Online** sempre que a comunicação com um equipamento ocorre com sucesso, e vai para o modo **Offline** após um determinado número de erros de comunicação. Este valor pode ser configurado na opção **Tag goes offline after n comm. errors**. Para mais informações, consulte o tópico **Leitura Automática de Eventos**. Os valores possíveis para a propriedade **Value** são **0**: Tag Bloco **Leitura Automática de Eventos** em modo **Offline** ou **1**: Tag Bloco **Leitura Automática de Eventos** em modo **Online**.

## Referência de Tags Bloco

Esta seção contém informações sobre a configuração dos parâmetros **[B]** deste Driver.

### Entrada de Dados

#### Leitura e Escrita

<b>B1</b>	Endereço de um escravo. Use o valor 900 para o endereço de <i>broadcast</i>
<b>B2</b>	Canal
<b>B3</b>	0 (zero)
<b>B4</b>	Número de um dado

Corresponde à categoria **I**. Permite escrever ou ler dados analógicos de um escravo, bem como os estados digitais de entrada. O tipo de dados da propriedade **Value** dos Elementos de Bloco é um ponto flutuante (*floating point*).

### Saída de Dados

#### Leitura e Escrita

<b>B1</b>	Endereço de um escravo. Use o valor 900 para o endereço de <i>broadcast</i>
<b>B2</b>	Canal
<b>B3</b>	1 (um)
<b>B4</b>	Número de um dado

Corresponde à categoria **O**. Permite escrever ou ler dados analógicos ou estados digitais de saída. O tipo de dados da propriedade **Value** dos Elementos de Bloco é um ponto flutuante (*floating point*).

## Configuração de Valores

### Leitura e Escrita

<b>B1</b>	Endereço de um escravo. Use o valor 900 para o endereço de <i>broadcast</i>
<b>B2</b>	Canal
<b>B3</b>	2 (dois)
<b>B4</b>	Número de um dado

Corresponde à categoria **S**. Permite escrever ou ler parâmetros de um escravo. O tipo de dados da propriedade **Value** dos Elementos de Bloco é um ponto flutuante (*floating point*).

## Variáveis Internas

### Leitura e Escrita

<b>B1</b>	Endereço de um escravo. Use o valor 900 para o endereço de <i>broadcast</i>
<b>B2</b>	Canal
<b>B3</b>	3 (três)
<b>B4</b>	Número de um dado

Corresponde à categoria **V**. Permite escrever ou ler em variáveis internas de um equipamento. O tipo de dados da propriedade **Value** dos Elementos de Bloco é um ponto flutuante (*floating point*). Os valores das variáveis internas fornecem acesso a dados ou funções, como por exemplo:

- Dados complementares sobre um processo supervisionado e os respectivos eventos
- Dados complementares relativos a um escravo ou ao funcionamento
- Controle das funções de um escravo em operação
- Programação de funções gerais em um escravo

## Dados de Memória

### Leitura e Escrita

<b>B1</b>	Endereço de um escravo. Use o valor 900 para o endereço de <i>broadcast</i>
<b>B2</b>	Canal
<b>B3</b>	4 (quatro)
<b>B4</b>	Número de um dado

Corresponde à categoria **M**. Permite escrever ou ler dados da memória de um escravo. Pode incluir dados de medições ou de status armazenados na memória de um equipamento. O tipo de dados da propriedade **Value** dos Elementos de Bloco é uma **String**.

## Últimos Eventos

### Leitura e Escrita

<b>B1</b>	Endereço de um escravo. Use o valor 900 para o endereço de <i>broadcast</i>
<b>B2</b>	0 (zero)
<b>B3</b>	8 (oito)
<b>B4</b>	0 (zero)

Corresponde à categoria **L**. Permite ler os eventos mais recentes do *buffer* de um escravo. Quando um evento é lido como um Tag PLC, este evento é retornado como uma **String**. Para mais informações, consulte o tópico **Referência de Tags PLC**. Na leitura em um Tag Bloco, os eventos são retornados com os seguintes Elementos de Bloco:

- **Elemento 0:** Estampa de tempo
- **Elemento 1:** Canal
- **Elemento 2:** Código de um evento

A propriedade **Timestamp** retorna o mesmo valor do Elemento de Bloco 0 (zero).

#### NOTA

Eventos analógicos ainda não são suportados para leitura como Tags Bloco. Note que é possível configurar um escravo através de uma variável interna e portanto não enviar eventos analógicos. Para mais informações, consulte a documentação do fabricante.

## Últimos Eventos do Buffer de Backup

### Leitura e Escrita

<b>B1</b>	Endereço de um escravo. Use o valor 900 para o endereço de <i>broadcast</i>
<b>B2</b>	0 (zero)
<b>B3</b>	10
<b>B4</b>	0 (zero)

Corresponde à categoria **B**. Permite ler os eventos mais recentes do *buffer* de um escravo. Quando um evento é lido como um Tag PLC, este evento é retornado como uma **String**. Para mais informações, consulte o tópico **Referência de Tags PLC**. Na leitura como um Tag Bloco, os eventos são retornados com os seguintes Elementos de Bloco:

- **Elemento 0:** Estampa de tempo
- **Elemento 1:** Canal
- **Elemento 2:** Código de um evento

A propriedade **Timestamp** retorna o mesmo valor do Elemento de Bloco 0 (zero).

## Condições de Eventos

### Somente Leitura

<b>B1</b>	Endereço de um escravo. Use o valor 900 para o endereço de <i>broadcast</i>
<b>B2</b>	Canal
<b>B3</b>	13
<b>B4</b>	0 (zero)

Corresponde à categoria **V**. Lê condições de eventos das variáveis internas de um escravo em um Tag Bloco. O primeiro Elemento de Bloco é retornado no formato **Data e Hora**. Os demais Elementos de Bloco são retornados no formato de ponto flutuante (*floating point*) de 32 bits.

## Leitura Automática de Eventos

### Leitura e Escrita

<b>B1</b>	Endereço de um escravo. Use o valor 900 para o endereço de <i>broadcast</i>
<b>B2</b>	0 (zero)
<b>B3</b>	100
<b>B4</b>	0 (zero)

Este Tag Bloco retorna eventos armazenados no *buffer* interno deste Driver, previamente coletados através do *scan* interno de leitura automática de eventos. Para mais informações, consulte o tópico **Leitura Automática de Eventos**. Os Elementos de Bloco deste Tag Bloco são os seguintes:

- **Elemento 0:** Estampa de tempo
- **Elemento 1:** Canal
- **Elemento 2:** Código de um evento

A propriedade **Timestamp** retorna o mesmo valor do Elemento de Bloco 0 (zero)

## Leitura Automática de Eventos

A partir da versão **2.0**, este Driver passa a contar com a leitura automática de eventos.

O mecanismo de leitura automática utiliza uma *thread* interna deste Driver para realizar a varredura, ou *scan*, de eventos de um equipamento. Os eventos coletados de um equipamento são armazenados em um *buffer* interno deste Driver e retornados para uma aplicação através do Tag Bloco **Leitura Automática de Eventos**.

Os eventos lidos por este Tag Bloco são eliminados do *buffer* interno deste Driver e reportados por evento, ou seja, cada operação de leitura deste Tag Bloco dispara uma série de eventos **OnRead**. Para mais informações, consulte o tópico *Tags de Comunicação Reportados por Eventos* no **Manual do Usuário do Elipse E3**.

A leitura automática de eventos equivale à leitura dos Tags Bloco **Últimos Eventos** e **Últimos Eventos do Buffer de Backup**, ou seja, as categorias **L** e **B**. As leituras são sempre realizadas em Tags Bloco com 3 (três) Elementos de Bloco. Este Driver inicia a varredura interna lendo a categoria **L**. Em caso de erro de comunicação em um determinado canal e escravo, a leitura é automaticamente comutada para a categoria **B** (*backup*) para evitar a perda de eventos. Após a primeira leitura com sucesso de eventos da categoria **B**, este Driver volta a buscar eventos da categoria **L**.

Assim como para a leitura dos Tags Bloco **Últimos Eventos** e **Últimos Eventos do Buffer de Backup**, não são suportados eventos analógicos.

## Configuração da Leitura Automática

Para usar a leitura automática de eventos, é preciso habilitar a varredura automática na janela de configurações deste Driver. Deve-se também configurar o período de varredura, ou *scan period*, e o intervalo entre Tags, ou *Tag scan delay*. A cada período de varredura, este Driver inicia o *scan* de todos os Tags de leitura automática de eventos registrados, com um intervalo entre cada leitura, ou *Tag scan delay*. O intervalo entre leituras de Tags permite distribuir a carga da varredura interna, evitando sobrecarregar uma aplicação e congestionar o canal de comunicação a cada *scan*.

A primeira operação de leitura de qualquer Tag Bloco **Leitura Automática de Eventos** não retorna valores, ou seja, retorna uma lista vazia, servindo apenas para que este Driver registre internamente aquele Tag Bloco no mecanismo de varredura automática. Nenhuma propriedade daquele Tag Bloco é alterada. A partir deste momento, aquele Tag Bloco passa a ser incluído em cada operação de varredura automática. As leituras seguintes passam então a retornar os valores coletados, sempre que estão disponíveis no *buffer* interno deste Driver. Os tempos de varredura dos Tags de leitura automática em uma aplicação podem ser baixos, como por exemplo 100 milissegundos. Uma vez que estes Tags não geram comunicação com um equipamento, o consumo de CPU é baixo.

## Auto Clear Status

Além da captura de eventos, é possível configurar a *thread* de varredura interna para enviar uma escrita do valor 0 (zero) para a categoria **C**, ou seja, C igual a 0 (zero), de forma automática, sempre que são recebidos eventos com códigos **E50** ou **E51**. Isto remove um equipamento da condição de erro ou *overflow*, reportadas por estes eventos. Os eventos **E50** e **E51** coletados são armazenados no *buffer* interno deste Driver e retornados normalmente para uma aplicação. Para usar este recurso, habilite a opção **Auto Clear Status** na janela de configurações.

## Evitando a Perda de Eventos aos Encerrar este Driver

O Tag PLC **Eventos no Buffer** permite monitorar a presença de dados no *buffer* interno deste Driver. Este Tag PLC pode ser usado para evitar a perda de eventos ao finalizar uma aplicação. Se é importante prevenir a perda de eventos que permaneçam no *buffer* interno no momento da finalização de uma aplicação, deve-se suspender a varredura interna, colocando este Driver em modo **Offline** antes de terminá-lo, deixando-se as varreduras dos Tags de leitura automática habilitadas. Desta forma, não são adicionados novos eventos ao *buffer* enquanto os Tags de leitura automática consomem os eventos restantes. Aguarde o Tag PLC **Eventos no Buffer** retornar a 0 (zero) e então finalize este Driver com segurança, sem perda de eventos.

## Monitoramento do Status da Comunicação com cada Escravo

A fim de monitorar o status da varredura dos Tags de leitura automática, bem como o status da comunicação com cada escravo, foi criado o Tag PLC **Evento Online**. Este Tag PLC referencia Tags de leitura automática através dos parâmetros *N1* e *N2*, retornando 0 (zero, modo **Offline**) ou 1 (um, modo **Online**) para indicar o status da comunicação para cada Tag. Os Tags de leitura automática são considerados em modo **Online** sempre que uma leitura automática é bem-sucedida, e passam a ser considerados em modo **Offline** após um determinado número de erros de comunicação consecutivos. O número máximo de erros de comunicação permitidos sem que um Tag seja considerado em modo **Offline** pode ser configurado na opção **Tag goes offline after n comm. errors** da janela de configuração.

## Documentação das Interfaces de Comunicação

Esta seção contém a documentação das Interfaces de Comunicação referentes ao Driver **SPABus32**.

## 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
<b>Physical Layer</b>	Selecione a interface física em uma lista. As opções disponíveis são <b>Serial</b> , <b>Ethernet</b> , <b>Modem</b> e <b>RAS</b> . A interface selecionada deve ser configurada na aba específica
<b>Timeout</b>	Configure o <i>time-out</i> , 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 <i>buffer</i> de recepção
<b>Communication check time</b>	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

OPÇÃO	DESCRIÇÃO
	Comunicação não receber dados válidos neste período de tempo, o estado é considerado inativo. O estado de comunicação é mostrado no Tag <b>IO.CommunicationStatus</b>
<b>Start driver OFFLINE</b>	Selecione esta opção para que um Driver inicie em modo <b>Offline</b> ou parado. Isto significa que a interface de I/O não é criada até que se configure um Driver em modo <b>Online</b> 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
<b>Mode</b>	Seleciona o modo de gerenciamento de conexão. Selecionar a opção <b>Automatic</b> permite que um Driver gerencie a conexão automaticamente, como especificado nas opções seguintes. Selecionar a opção <b>Manual</b> permite que uma aplicação gerencie a conexão completamente
<b>Retry failed connection every ... seconds</b>	Selecione esta opção para habilitar a retentativa de conexão de um Driver em um determinado intervalo, em segundos. Se a opção <b>Give up after failed retries</b> não está selecionada, este Driver continua retentando até que a conexão seja efetuada, ou que a aplicação seja parada
<b>Give up after ... failed retries</b>	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 <b>Offline</b> , 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
<b>Disconnect if non-responsive for ... seconds</b>	Habilite esta opção para forçar um Driver a se desconectar se nenhum byte chegou à interface de I/O no <i>time-out</i> especificado, em segundos. Este <i>time-out</i> deve ser maior que o <i>time-out</i> configurado na opção <b>Timeout</b>

Opções para o grupo Logging Options

OPÇÃO	DESCRIÇÃO
<b>Log to File</b>	<p>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 <b>%PROCESS%</b> 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 <b>Elipse E3</b>, 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 <b>0FDAh</b>. Pode-se também utilizar a macro <b>%DATE%</b> no nome do arquivo. Neste caso é gerado um arquivo de log por dia, no formato <b>aaaa_mm_dd</b>. 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 <b>%DATE_HOUR%</b> gera um arquivo de log por hora, no formato <b>aaaa_mm_dd_hh</b></p>
<b>File size limit (MB)</b>	<p>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</p>

## Aba Serial

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

Serial

Port:

Baud rate:

Data bits:

Parity:

Stop bits:

Enable 'ECHO' suppression

Handshaking

DTR control:

RTS control:

Wait for CTS before send

CTS timeout:  ms

Delay before send:  ms

Delay after send:  ms

Inter-byte delay (microseconds):   $\mu$ s

Inter-frame delay (milliseconds):  ms

Aba Serial

## Opções gerais da aba Serial

OPÇÃO	DESCRIÇÃO
<b>Port</b>	Selecione uma porta serial a partir da lista, de <b>COM1</b> até <b>COM4</b> , ou digite o nome de uma porta serial no formato <b>COMn</b> , 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"
<b>Baud rate</b>	Selecione um <i>baud rate</i> a partir da lista ( <b>1200, 2400, 4800, 9600, 19200, 38400, 57600</b> ou <b>115200</b> ) ou digite um <i>baud rate</i> , como por exemplo 600
<b>Data bits</b>	Selecione 7 (sete) ou 8 (oito) bits de dados a partir da lista
<b>Parity</b>	Selecione uma paridade a partir da lista. As opções disponíveis são <b>None, Even, Odd, Mark</b> ou <b>List</b>
<b>Stop bits</b>	Selecione o número de stop bits a partir da lista. As opções disponíveis são <b>1, 1.5</b> ou <b>2</b> stop bits
<b>Enable 'ECHO' suppression</b>	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
<b>Inter-byte delay (microseconds)</b>	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
<b>Inter-frame delay (milliseconds)</b>	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
<b>DTR control</b>	Selecione o valor <b>ON</b> para deixar o sinal <b>DTR</b> sempre ligado enquanto a porta serial está aberta. Selecione o valor <b>OFF</b> para desligar o sinal <b>DTR</b> enquanto a porta serial está aberta. Alguns equipamentos exigem que o sinal <b>DTR</b> esteja ligado para permitir a comunicação
<b>RTS control</b>	Selecione o valor <b>ON</b> para deixar o sinal <b>RTS</b> sempre ligado enquanto a porta serial está aberta. Selecione o valor <b>OFF</b> para desligar o sinal <b>RTS</b> enquanto a porta serial está aberta. Selecione o valor <b>Toggle</b> para ligar o sinal <b>RTS</b> enquanto se envia os bytes através da porta serial, e desligá-lo quando não se está enviando bytes e, portanto, habilitando a recepção
<b>Wait for CTS before send</b>	Disponível apenas quando a opção <b>RTS control</b> está configurada com o valor <b>Toggle</b> . Utilize esta opção para forçar um Driver a verificar o sinal <b>CTS</b> antes de enviar os bytes através da porta serial, após ligar o sinal de <b>RTS</b> . Neste modo o sinal <b>CTS</b> é tratado como um <i>flag</i> de permissão para envio
<b>CTS timeout</b>	Determina o tempo máximo, em milissegundos, que um Driver aguarda pelo sinal de <b>CTS</b> depois de ligar o sinal de <b>RTS</b> . Se o sinal de <b>CTS</b> não é levantado dentro deste <i>time-out</i> , este Driver falha a comunicação atual e retorna erro
<b>Delay before send</b>	Alguns equipamentos de porta serial demoram a habilitar o circuito de envio de dados depois que o sinal <b>RTS</b> é ligado. Configure esta opção para aguardar uma determinada quantidade de milissegundos depois de ligar o sinal <b>RTS</b> e antes de enviar o primeiro byte. <b>IMPORTANTE:</b> 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
<b>Delay after send</b>	Tem o mesmo efeito que a opção <b>Delay before send</b> , mas neste caso a espera é efetuada depois que o último byte é enviado, antes de desligar o sinal <b>RTS</b>

## 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**.

Ethernet

Transport: TCP/IP ▼

PING before connecting

Timeout: 4000 ms

Retries: 1

Listen for connections on port: 0

Share listen port with other processes

Interface: (All Interfaces) ▼

Use IPv6  Use SSL SSL Settings

Enable 'ECHO' suppression

IP Filter:

Connect to

<input type="checkbox"/> Main IP: <span style="border: 1px solid gray; padding: 2px;"> </span>	Port: <span style="border: 1px solid gray; padding: 2px;">502</span>	<input type="checkbox"/> Local port: <span style="border: 1px solid gray; padding: 2px;">0</span>
<input type="checkbox"/> Backup IP 1: <span style="border: 1px solid gray; padding: 2px;"> </span>	Port: <span style="border: 1px solid gray; padding: 2px;">0</span>	<input type="checkbox"/> Local port: <span style="border: 1px solid gray; padding: 2px;">0</span>
<input type="checkbox"/> Backup IP 2: <span style="border: 1px solid gray; padding: 2px;"> </span>	Port: <span style="border: 1px solid gray; padding: 2px;">0</span>	<input type="checkbox"/> Local port: <span style="border: 1px solid gray; padding: 2px;">0</span>
<input type="checkbox"/> Backup IP 3: <span style="border: 1px solid gray; padding: 2px;"> </span>	Port: <span style="border: 1px solid gray; padding: 2px;">0</span>	<input type="checkbox"/> Local port: <span style="border: 1px solid gray; padding: 2px;">0</span>

**Aba Ethernet**

### Opções disponíveis na aba Ethernet

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

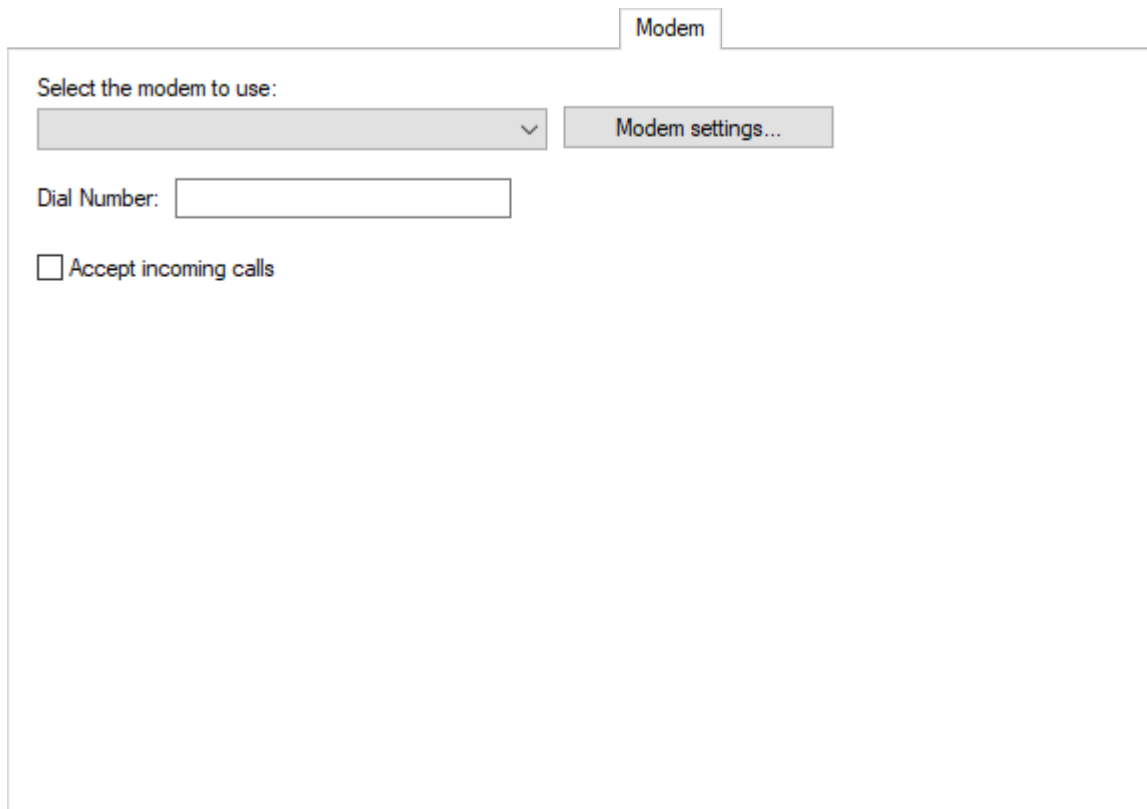
OPÇÃO	DESCRIÇÃO
<b>IP Filter</b>	Lista de endereços IP restringidos ou permitidos de onde um Driver aceita conexões ( <i>Firewall</i> ). Consulte a propriedade <b>IO.Ethernet.IPFilter</b> para mais informações
<b>PING before connecting</b>	Habilite esta opção para executar um comando <b>ping</b> , ou seja, para verificar se um dispositivo pode ser encontrado na rede, em um dispositivo antes de tentar uma conexão com o <i>socket</i> . Esta é uma maneira rápida de determinar uma conexão bem sucedida antes de tentar abrir um <i>socket</i> com um dispositivo. O <i>time-out</i> de uma conexão com um <i>socket</i> pode ser bem alto. As opções disponíveis são: <ul style="list-style-type: none"> <li>• <b>Timeout:</b> Especifique o número de milissegundos de espera por uma resposta de um comando <b>ping</b>. Deve-se usar um comando <b>ping</b> 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</li> <li>• <b>Retries:</b> Número de retentativas de um comando <b>ping</b>, sem contar a tentativa inicial. Se todas as tentativas falharem, então a conexão com o <i>socket</i> é abortada</li> </ul>

#### Opções disponíveis no grupo Connect to

OPÇÃO	DESCRIÇÃO
<b>Main IP</b>	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"
<b>Port</b>	Digite a porta IP de um dispositivo remoto, entre 0 (zero) e 65535
<b>Local port</b>	Selecione esta opção para utilizar uma porta IP local fixa ao conectar a um dispositivo remoto
<b>Backup IP 1, 2 e 3</b>	Indique o endereço IP, a porta IP e a porta IP local fixa de até 3 (três) endereços de <i>backup</i> 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
<b>Select the modem to use</b>	Selecione um modem a partir da lista de modems disponíveis no computador. Caso selecione-se o valor <b>Default modem</b> , então o primeiro modem disponível é utilizado. Selecionar este valor é recomendado especialmente quando uma aplicação é utilizada em outro computador
<b>Modem settings</b>	Clique para abrir a janela de configuração do modem selecionado
<b>Dial Number</b>	Digite o número padrão para discagem. Este valor pode ser modificado em tempo de execução. Pode-se utilizar o caractere <b>w</b> 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"
<b>Accept incoming calls</b>	Habilite esta opção para que um Driver atenda o telefone quando receber uma chamada externa. Para utilizar esta opção é necessário configurar a opção <b>Connection management</b> na aba <b>Setup</b> para o valor <b>Manual</b>

**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**.

The image shows a screenshot of a software interface with a tab labeled "RAS". Inside the tab, there are two input fields: "AT command:" followed by an empty text box, and "Connection timeout:" followed by a text box containing the number "0" and the word "seconds". Below these fields, there is a text message: "Other socket settings should be configured in the 'Ethernet' tab!".

Aba RAS

### Opções disponíveis na aba RAS

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

## 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

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Leitura
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	0 (zero)
<b>Parâmetro N4</b>	6 (seis)
<b>Configuração por String</b>	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

<b>Tipo de Tag</b>	Tag Bloco
<b>Tipo de Acesso</b>	Somente Leitura
<b>Parâmetro B1</b>	-1 (menos um)
<b>Parâmetro B2</b>	0 (zero)
<b>Parâmetro B3</b>	0 (zero)
<b>Parâmetro B4</b>	1 (um)
<b>Propriedade Size</b>	4 (quatro)
<b>Propriedade ParamItem</b>	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

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Somente Leitura
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	0 (zero)
<b>Parâmetro N4</b>	2 (dois)
<b>Configuração por String</b>	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

<b>Tipo de Tag</b>	Tag Bloco
<b>Tipo de Acesso</b>	Somente Leitura
<b>Parâmetro B1</b>	-1 (menos um)
<b>Parâmetro B2</b>	0 (zero)
<b>Parâmetro B3</b>	0 (zero)
<b>Parâmetro B4</b>	3 (três)
<b>Propriedade Size</b>	2 (dois)
<b>Propriedade ParamItem</b>	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 **Eclipse 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,  $3 \times 2$ ). 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 **Eclipse 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

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Leitura ou Escrita
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	0 (zero)
<b>Parâmetro N4</b>	4 (quatro)
<b>Configuração por String</b>	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

**9** 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

**9** 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

9 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

9 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

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

## IO.Type

A 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

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

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

#### IO.Stats.Partial.BytesSent

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

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

#### IO.Stats.Partial.TimeConnectedSeconds

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Somente Leitura
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	0 (zero)
<b>Parâmetro N4</b>	1102
<b>Configuração por String</b>	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

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Somente Leitura
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	0 (zero)
<b>Parâmetro N4</b>	1103
<b>Configuração por String</b>	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

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

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

## IO.Stats.Total.BytesSent

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

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

## IO.Stats.Total.ConnectionCount

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Somente Leitura
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	0 (zero)
<b>Parâmetro N4</b>	1004
<b>Configuração por String</b>	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

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

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

## IO.Stats.Total.TimeDisconnectedSeconds

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Somente Leitura
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	0 (zero)
<b>Parâmetro N4</b>	1003
<b>Configuração por String</b>	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

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Leitura ou Escrita
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	4 (quatro)
<b>Parâmetro N4</b>	0 (zero)
<b>Configuração por String</b>	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

<b>Tipo de Tag</b>	Tag de Comunicação
--------------------	--------------------

<b>Tipo de Acesso</b>	Somente Escrita
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	4 (quatro)
<b>Parâmetro N4</b>	1 (um)
<b>Configuração por String</b>	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

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Somente Leitura
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	4 (quatro)
<b>Parâmetro N4</b>	2 (dois)
<b>Configuração por String</b>	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]

**A** 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]

**9** 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]

**9** 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

**A** 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:0877:0000:0000:0000:0000:0000:\***): Aceita conexões de endereços IPv6 no intervalo entre fe80:03bf:0877:0000:0000:0000:0000:0000 e fe80:03bf:0877:0000:0000:0000:0000: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

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Somente Escrita
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	3 (três)
<b>Parâmetro N4</b>	1 (um)
<b>Configuração por String</b>	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

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Somente Escrita
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	3 (três)
<b>Parâmetro N4</b>	4 (quatro)
<b>Configuração por String</b>	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

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Somente Leitura
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	3 (três)
<b>Parâmetro N4</b>	3 (três)
<b>Configuração por String</b>	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

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Somente Leitura
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	3 (três)
<b>Parâmetro N4</b>	6 (seis)
<b>Configuração por String</b>	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

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Somente Leitura
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	3 (três)
<b>Parâmetro N4</b>	2 (dois)
<b>Configuração por String</b>	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

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Leitura ou Escrita
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	3 (três)
<b>Parâmetro N4</b>	0 (zero)
<b>Configuração por String</b>	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

**9** 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

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

### IO.Serial.CTSTimeoutMs

9 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

9 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

9 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

9 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

A 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

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

### IO.Serial.InterframeDelayMs

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

### IO.Serial.Parity

A 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

9 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 COMn.

## IO.Serial.RTS

**A** 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

**9** 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.SuppressEcho

**9** 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
3.0.2	22/08/2025	M. Ludwig	<ul style="list-style-type: none"> <li>Driver atualizado para a biblioteca <b>IOKit</b> versão <b>3.0</b> e Visual Studio 2022 (<i>Case 38034</i>).</li> </ul>
3.0.1	30/09/2013	A. Quites C. Mello	<ul style="list-style-type: none"> <li>Driver portado para a biblioteca <b>IOKit</b> versão <b>2.0</b> (<i>Case 14683</i>).</li> <li>Resolvido um problema na escrita do Tag <b>Sincronização de Relógio: Data e Hora</b>, parâmetro <i>N3</i> igual a 17, em modo <b>Broadcast</b>, parâmetro <i>N1</i> igual a 900, ou ao escrever Tags de categoria <b>D</b>, em que a data era enviada com o ano com 4 (quatro) dígitos, ao contrário da especificação do protocolo, que define este campo com 2 (dois) dígitos, de forma que os equipamentos não reconheciam os comandos (<i>Case 14666</i>).</li> <li>Implementada uma opção de enviar os caracteres 'XX' no lugar da soma de verificação (<i>checksum</i>) nos <i>frames</i> de comunicação. Esta opção pode ser útil em testes com HyperTerminal</li> </ul>

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			<p>ou simuladores, quando não é necessário calcular este <i>checksum</i> manualmente, ou para retrocompatibilidade com sistemas legados (Case 14684).</p> <ul style="list-style-type: none"> <li>• Driver modificado para retornar o valor 0 (zero) nas leituras dos Tags de sincronização de relógio em modo <b>Broadcast</b> nas categorias <b>T</b> e <b>D</b> (Case 14930).</li> <li>• Corrigido um vazamento de memória na finalização deste Driver quando utilizado no <b>Elipse SCADA</b> com a opção <b>autoscan</b> habilitada. Este erro não ocorre quando este Driver é usado com o <b>Elipse E3</b>, <b>Elipse Power</b> ou <b>Elipse Water</b> (Case 14905).</li> </ul>
2.3.1	15/10/2012	C. Mello	<ul style="list-style-type: none"> <li>• Adicionado suporte para comunicação com os relés REJ-525 (Case 12967).</li> </ul>
2.2.1	27/04/2011	A. Quites	<ul style="list-style-type: none"> <li>• Resolvido um erro de <i>rollover</i> na virada dos minutos na estampa de tempo de eventos (Case 12047).</li> </ul>
2.1.1	01/09/2010	A. Quites	<ul style="list-style-type: none"> <li>• Otimizada a leitura de status de Tags de Leitura Automática de Eventos (Case 11705).</li> </ul>
2.0.1	06/04/2010	A. Quites	<ul style="list-style-type: none"> <li>• Driver portado para a biblioteca <b>IOKit</b> (Case 11192).</li> <li>• Criada a captura automática de eventos em uma <i>thread</i> em segundo plano (Case 11208).</li> <li>• Inclusão de informação no manual do usuário sobre a limpeza dos eventos <b>E50</b> e <b>E51</b> através da escrita do valor 0 (zero) na categoria <b>C</b> (Case 11219).</li> </ul>
1.1.1	04/09/2006	C. Mello	<ul style="list-style-type: none"> <li>• Ajustes para prevenir um erro interno neste Driver</li> </ul>

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			durante a coleta de eventos ( <i>Case 5141</i> ). <ul style="list-style-type: none"><li>• Revisão geral de todo o código fonte.</li></ul>
		A. Quites	<ul style="list-style-type: none"><li>• Ajustes para comandos em modo <b>Broadcast</b>, ou seja, os parâmetros <i>N1</i> ou <i>B1</i> igual a 900 (<i>Case 7329</i>).</li><li>• Adicionado um novo Elemento de Bloco para retornar o índice da lista de eventos <b>L</b> e <b>B</b> (<i>Case 7333</i>).</li></ul>
<b>1.0.1</b>	11/12/2003	C. Kohlmann	<ul style="list-style-type: none"><li>• Todas as publicações anteriores ao controle de revisões.</li></ul>

**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](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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](mailto:evan@elipse.com.br)

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

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

[kb.elipse.com.br](http://kb.elipse.com.br)

[forum.elipse.com.br](http://forum.elipse.com.br)

[www.youtube.com/elipsesoftware](http://www.youtube.com/elipsesoftware)

[elipse@elipse.com.br](mailto:elipse@elipse.com.br)



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)