



SPABus32

v3.0.2 · Português

Conteúdo

Driver ABB SPABus

Configuração do Driver

Parâmetros [N] de Endereçamento de Tags PLC

Parâmetros [B] de Endereçamento de Tags Bloco

Referência de Tags PLC

Entrada de Dados

Saída de Dados

Configuração de Valores

Variáveis Internas

Dados de Memória

Status de um Escravo

Identificação de um Escravo

Hora

Últimos Eventos

Data e Hora

Último Evento do Buffer de Backup

Alarme Válido

Sincronização de Relógio: Data e Hora

Sincronização de Relógio: Somente Segundos

Eventos no Buffer

Evento Online

Referência de Tags Bloco

Entrada de Dados

Saída de Dados

Configuração de Valores

Variáveis Internas

Dados de Memória

Últimos Eventos

Últimos Eventos do Buffer de Backup

Condições de Eventos

Leitura Automática de Eventos

Leitura Automática de Eventos

Configuração da Leitura Automática

Auto Clear Status

Evitando a Perda de Eventos aos Encerrar este Driver

Monitoramento do Status da Comunicação com cada Escravo

Documentação das Interfaces de Comunicação

Histórico de Revisões do Driver

Driver ABB SPABus

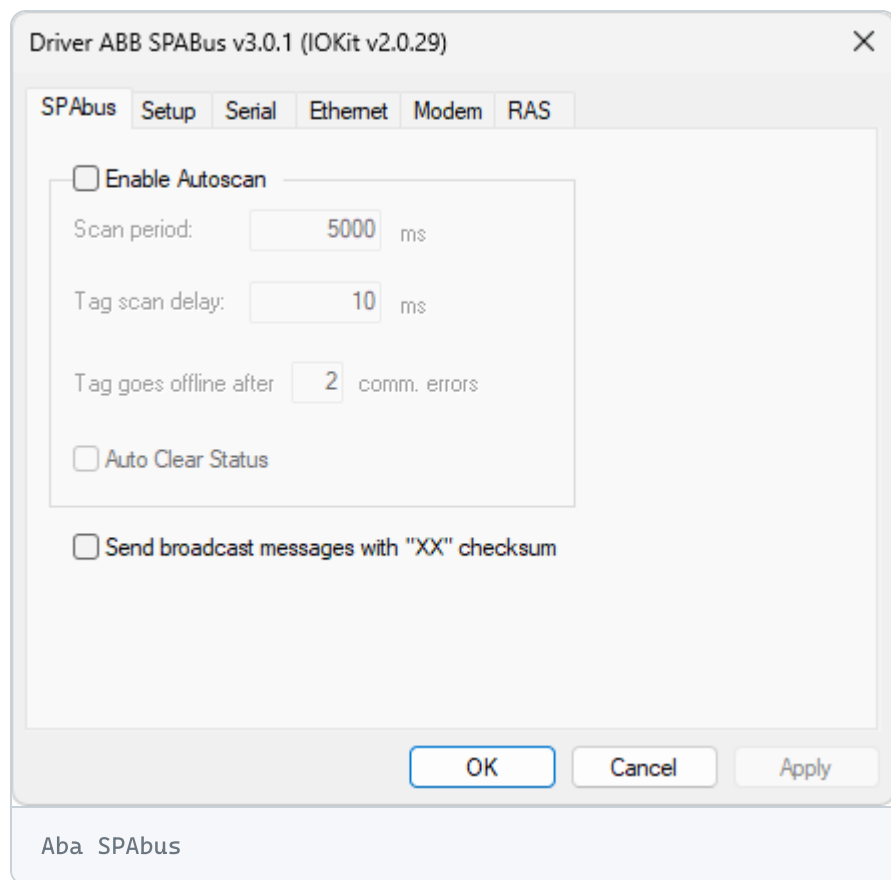
Nome do Arquivo	SPABus32.dll
Fabricante	Asea Brown Bowery Ltd.
Equipamentos	REF/RET/REM 541/543/545 Feeder Terminals, REX521 Relay e REJ/REU 521/525/527 Relays
Protocolo	SPABus Communication Protocol versão 2.5
Versão	3.0.2
Última Atualização	22/08/2025
Plataforma	Win32
Dependências	Nenhuma
Leitura Superblocos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

Introdução

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

Configuração do Driver

A partir da versão **2**, 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.



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

Opções disponíveis na aba SPABus

OPÇÃO	DESCRIÇÃO
Enable Autoscan	Habilita a varredura automática por eventos de categorias L ou B . Para mais informações, consulte o tópico Leitura Automática de Eventos
Scan period	Define o período de varredura, em milissegundos, para a leitura automática de eventos. Para mais informações, consulte o tópico Leitura Automática de Eventos
Tag scan delay	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
Tag goes offline after n comm. errors	Define o número de erros de comunicação após os quais um Tag de leitura automática de eventos passa a ser considerado em estado Offline . Este Tag volta para o estado Online assim que alguma comunicação seja bem-sucedida
Auto Clear Status	Se esta opção está habilitada durante a captura automática de eventos, caso este Driver receba eventos de código NaN ou NaN , 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 C , retirando um escravo da condição de erro. Os eventos NaN e NaN coletados são retornados normalmente para uma aplicação
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 Texto
9	Data e hora (T)
10	Últimos eventos do <i>buffer de backup</i> (B) no formato Texto
11	Alarme válido (A)*
17	Tag de sincronização de relógio (data completa, somente leitura)
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 Online

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 Data e Hora , Número de um Canal e Código de Causa de uma Ocorrência
10	Últimos eventos do <i>buffer de backup</i> (B). Esta opção utiliza um Bloco com os Elementos Data e Hora , Número de um Canal e Código de Causa de uma Ocorrência
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

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

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

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

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

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

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

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

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

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

N1	0 (zero)
N2	0 (zero)
N3	101
N4	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

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

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

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

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

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

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

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

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

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

B1	Endereço de um escravo. Use o valor 900 para o endereço de <i>broadcast</i>
B2	0 (zero)
B3	100
B4	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**, 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 **NaN** ou **NaN**. Isto remove um equipamento da condição de erro ou *overflow*, reportadas por estes eventos. Os eventos **NaN** e **NaN** 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**.

Esta documentação é compartilhada e mantida no manual do IOKit:

- Configurações de um Driver
- Configurações Gerais
- Configuração de Estatísticas
- Configuração da Interface Ethernet
- Configuração da Interface Modem
- Configuração da Interface RAS
- Configuração da Interface Serial

Histórico de Revisões do Driver

VERSÃO	DATA	AUTOR	COMENTÁRIOS
3.0.2	22/08/2025	M. Ludwig	• Driver atualizado para a biblioteca IOKit versão 3 e Visual Studio 2022 (<i>Case 38034</i>).
3.0.1	30/09/2013	A. Quites C. Mello	• Driver portado para a biblioteca IOKit versão 2 (<i>Case 14683</i>). • Resolvido um problema na escrita do Tag Sincronização de Relógio: Data e Hora, parâmetro <i>N3</i> igual a 17, em modo Broadcast, parâmetro <i>N1</i> igual a 900, ou ao escrever Tags de categoria D, 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>). • 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 ou simuladores, quando não é necessário calcular este <i>checksum</i> manualmente, ou para retrocompatibilidade com sistemas legados (<i>Case 14684</i>). • Driver modificado para retornar o valor 0 (zero) nas leituras dos Tags de sincronização de relógio em modo Broadcast nas categorias T e D (<i>Case 14930</i>). • Corrigido um vazamento de memória na finalização deste Driver quando utilizado no Elipse SCADA com a opção autoscan habilitada. Este erro não ocorre quando este Driver é usado com o E3, Power ou Water (<i>Case 14905</i>).
2.3.1	15/10/2012	C. Mello	• Adicionado suporte para comunicação com os relés REJ-525 (<i>Case 12967</i>).
2.2.1	27/04/2011	A. Quites	• Resolvido um erro de <i>rollover</i> na virada dos minutos na estampa de tempo de eventos (<i>Case 12047</i>).
2.1.1	01/09/2010	A. Quites	• Otimizada a leitura de status de Tags de Leitura Automática de Eventos (<i>Case 11705</i>).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
2.0.1	06/04/2010	A. Quites	• Driver portado para a biblioteca IOKit (Case 11192). • Criada a captura automática de eventos em uma <i>thread</i> em segundo plano (Case 11208). • Inclusão de informação no manual do usuário sobre a limpeza dos eventos NaN e NaN através da escrita do valor 0 (zero) na categoria C (Case 11219).
1.1.1	04/09/2006	C. Mello	• Ajustes para prevenir um erro interno neste Driver durante a coleta de eventos (Case 5141). • Revisão geral de todo o código fonte.
		A. Quites	• Ajustes para comandos em modo Broadcast, ou seja, os parâmetros <i>N1</i> ou <i>B1</i> igual a 900 (Case 7329). • Adicionado um novo Elemento de Bloco para retornar o índice da lista de eventos L e B (Case 7333).
1.0.1	11/12/2003	C. Kohlmann	• Todas as publicações anteriores ao controle de revisões.