

Driver Siemens Prodave

Nome do Arquivo	ProdaveDll.dll
Fabricante	Siemens AG
Equipamentos	S7-200, S7-300 e S7-400
Protocolo	MPI e Industrial Ethernet
Versão	2.0.5
Última Atualização	15/05/2026
Plataforma	Win32
Dependências	Prodave MPI/IE versão 6.x
Leitura com Superblocos	Não
Nível	0

Introdução

Este Driver implementa uma interface com a biblioteca Prodave MPI/IE versão 6.x da Siemens AG para sistemas operacionais **Windows 7** ou superiores, que permite aos produtos da **Elipse Software** se comunicarem com os equipamentos S7-200, S7-300 e S7-400.

O Driver Siemens Prodave foi desenvolvido em conjunto com a biblioteca **IOKit** da **Elipse Software**. Para mais informações, consulte o tópico **Documentação das Interfaces de Comunicação**.

NOTA

Para utilização do produto **Siemens Prodave MPI versão 5.6**, a versão **1.1** deste Driver é a versão compatível com aquele produto e a que deve ser utilizada.

Requerimentos de Funcionamento do Driver

Para que o Driver Siemens Prodave funcione corretamente, é necessário antes instalar o Prodave MPI/IE versão 6.x. Após a instalação bem-sucedida, o arquivo de biblioteca dinâmica Prodave6.dll deve estar localizado na pasta de sistema do Windows e acessível. O Driver Siemens Prodave conecta-se diretamente a este arquivo de biblioteca.

Também é necessário definir a interface de comunicação no Painel de Controle do Windows, no item **Setting the PG/PC Interface**. Neste local deve existir um *Access Point* com o nome "S7ONLINE". Para mais informações a respeito da instalação e configuração desta interface, consulte o manual correspondente ao produto da Siemens AG.

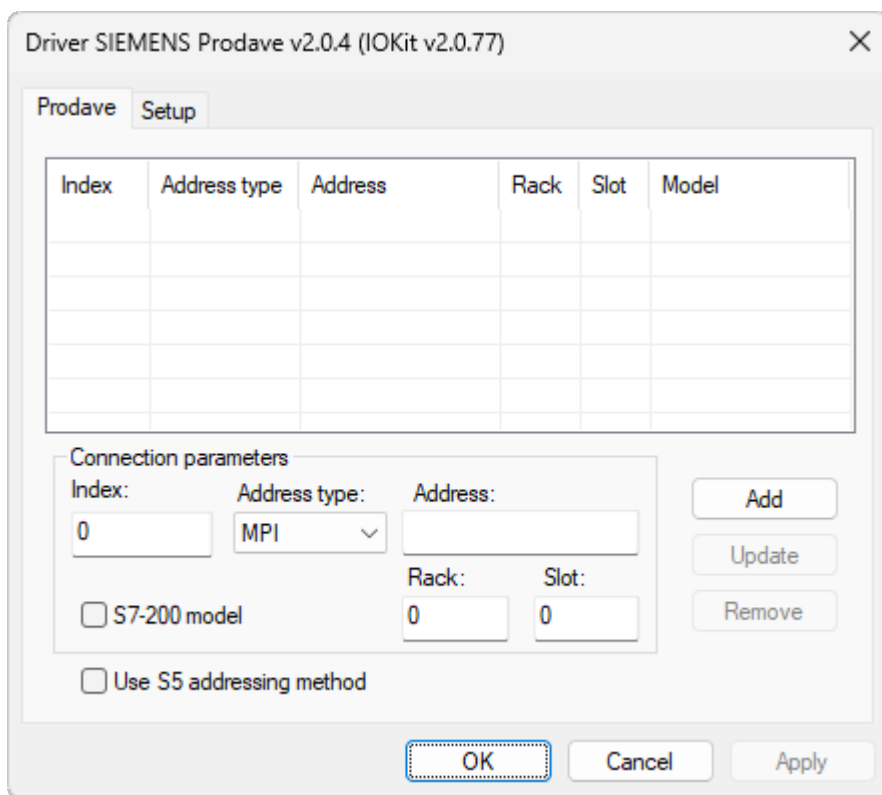
Configuração do Driver

Os parâmetros **[P]** de configuração deste Driver não são utilizados. Todas as configurações são executadas na janela de propriedades deste Driver. Nesta janela, os parâmetros mais relevantes da aba **Setup** são:

- **Physical Layer**: Sempre selecione o valor **None**, pois a interface é fornecida pela biblioteca Prodave
- **Logging options - Log to File**: Nome de um arquivo de log gerado para depuração de problemas na comunicação

Para mais informações sobre a aba **Setup**, consulte o tópico **Documentação das Interfaces de Comunicação**.

A aba **Prodave** contém configurações específicas para este Driver. Quase todas estas configurações são apenas relevantes quando se configura Tags por parâmetros **[N/B]**. Estas configurações dizem respeito à lista de conexões a declarar. Desta forma, ao configurar Tags por endereçamento simbólico, a lista de conexões pode ser deixada em branco. A exceção é a opção **Use S5 addressing method**, que deve ser considerada independente do método de endereçamento selecionado.



Aba Prodave

Todos os equipamentos com os quais haja comunicação devem ser configurados. Para criar uma conexão com um equipamento, preencha os parâmetros de conexão e clique em **Add**. Os parâmetros de conexão estão descritos na tabela a seguir.

Parâmetros de configuração de conexões

PARÂMETRO	DESCRIÇÃO
Index	Índice de uma conexão a ser utilizada no parâmetro <i>N1</i> ou <i>B1</i>
Address type	Tipo de endereço. Os valores possíveis são MPI , IP ou MAC
Address	Endereço de um equipamento
S7-200 model	Selecione esta opção para indicar equipamentos do modelo S7-200
Rack	Valor do <i>rack</i> de um equipamento
Slot	Valor do <i>slot</i> de um equipamento

A opção **Update** reconfigura os parâmetros da conexão selecionada e a opção **Remove** apaga a conexão selecionada. A opção **Use S5 addressing method** define o modo de endereçamento dos dados. Se esta opção está selecionada, o modo de endereçamento é **S5** e endereça palavras sequencialmente, como por exemplo DW0, DW1, DW2, DW3 e assim por diante. Se esta opção não está selecionada, o modo de endereçamento é **S7** e endereça palavras pelo endereço do primeiro byte, como por exemplo DW0, DW2, DW4, DW8 e assim por diante.

Referência de Tags

Esta seção contém informações sobre a configuração dos Tags deste Driver. Tags podem ser configurados usando **Endereçamento por Parâmetros [N/B]** ou por **Endereçamento Simbólico**.

Endereçamento por Parâmetros [N/B]

Leitura e Escrita

Use a sintaxe padrão a seguir para todos os Tags PLC e Bloco.

Campos de Endereçamento por Parâmetros [N/B]

- **N1 ou B1:** Índice da conexão na lista de conexões definida na aba **Prodave**
- **N2 ou B2:** Área e tipo de dados. Para mais informações, consulte a tabela **Parâmetros de configuração**
- **N3 ou B3:** Número do DB. Se a área selecionada não é **DB**, deixe em 0 (zero)
- **N4 ou B4:** Endereço do DB (*offset*) ou endereço absoluto da área

Parâmetros de configuração

N2 OU B2	ÁREAS E TIPOS DE DADOS	S7-200	S7-300	S7-400
0	DB tipo SINT (8 bits com sinal)		•	•
1	DB tipo INT (16 bits com sinal)		•	•
2	DB tipo DINT (32 bits com sinal)		•	•
3	DB tipo BYTE (8 bits sem sinal)		•	•
4	DB tipo WORD (16 bits sem sinal)		•	•
5	DB tipo DWORD (32 bits sem sinal)		•	•
6	DB tipo REAL (32 bits ponto flutuante IEEE 754)		•	•
7	<i>Flag ou Memory</i> tipo SINT	•	•	•
8	<i>Flag ou Memory</i> tipo INT	•	•	•
9	<i>Flag ou Memory</i> tipo DINT	•	•	•
10	<i>Flag ou Memory</i> tipo BYTE	•	•	•

N2 OU B2	ÁREAS E TIPOS DE DADOS	S7-200	S7-300	S7-400
11	Flag ou Memory tipo WORD	•	•	•
12	Flag ou Memory tipo DWORD	•	•	•
13	Flag ou Memory tipo REAL	•	•	•
14	Saída (<i>output</i>) tipo BYTE	•	•	•
15	Entrada (<i>input</i>) tipo BYTE	•	•	•
16	Temporizador (<i>timer</i>) tipo WORD	•	•	•
17	Contador (<i>counter</i>) tipo WORD	•	•	•
18	Special Flag ou Memory tipo SINT	•		
19	Special Flag ou Memory tipo INT	•		
20	Special Flag ou Memory tipo DINT	•		
21	Special Flag ou Memory tipo BYTE	•		
22	Special Flag ou Memory tipo WORD	•		
23	Special Flag ou Memory tipo DWORD	•		
24	Special Flag ou Memory tipo REAL	•		
25	Variable Memory ou Storage tipo SINT	•		
26	Variable Memory ou Storage tipo INT	•		
27	Variable Memory ou Storage tipo DINT	•		
28	Variable Memory ou Storage tipo BYTE	•		
29	Variable Memory ou Storage tipo WORD	•		
30	Variable Memory ou Storage tipo DWORD	•		
31	Variable Memory ou Storage tipo REAL	•		

NOTA

Ao utilizar Tags Bloco, cada Elemento de Bloco corresponde a uma posição de área múltipla ao tipo de dados, iniciando a contagem a partir do endereço configurado.

Endereçamento Simbólico

Leitura e Escrita

O **Elipse E3** a partir da versão **2.0**, o **Elipse Power** e o **Elipse Water** permitem a configuração de Tags por endereçamento simbólico. Para isto, Utilize a sintaxe a seguir:

- O campo **Dispositivo** deve obedecer à sintaxe [**<Modelo>**]**<Endereço>**[**<Rack>**, **<Slot>**]:
 - **<Modelo>** é opcional e, caso seja utilizado, deve ser utilizada uma **String** literal entre parênteses, como por exemplo "S7-200", "S7-300" ou "S7-400". A omissão deste campo considera como um modelo qualquer S7-300 ou S7-400
 - **<Endereço>** deve conter um endereço no formato **MPI, IP** ou **MAC**
 - **<Rack>** e **<Slot>** são opcionais, devem estar entre parênteses e separados por uma vírgula. Cada campo é o respectivo valor de um equipamento. A omissão destes itens considera estes campos com um valor igual a 0 (zero)

Exemplos

1. Endereço **MPI** de valor 2 (dois) com um modelo de equipamento S7-200: Campo **Dispositivo** preenchido com o valor "(S7-200)2".
 2. Endereço IP 192.168.20.70 com um modelo de equipamento S7-400, *Rack* com um valor 0 (zero) e *Slot* com um valor 2 (dois): Campo **Dispositivo** preenchido com o valor "192.168.0.1(0,2)".
- O campo **Item**, quando é preciso endereçar DB, deve obedecer à sintaxe **<DB com tipo><Número do DB>:<Endereço>**:
 - **<DB com tipo>** deve conter alguma das **Strings** da tabela **Nomes de Áreas com Tipos de Dados**
 - **<Número do DB>** deve conter o valor de número do DB
 - **<Endereço>** deve conter o endereço, ou *offset*, no DB
 - O campo **Item**, quando é preciso endereçar outras áreas que não DB, deve obedecer à sintaxe **<Nome da área com tipo><Endereço>**:
 - **<Nome da área com tipo>** deve conter alguma das **Strings** da tabela **Nome de Áreas com Tipos de Dados**
 - **<Endereço>** deve conter o endereço na respectiva área

Nomes de Áreas com Tipos de Dados

ÁREA	ÁREA E TIPO DE DADOS	S7-200	S7-300	S7-400
DBSI	DB tipo SINT (8 bits com sinal)		.	.

ÁREA	ÁREA E TIPO DE DADOS	S7-200	S7-300	S7-400
DBI	DB tipo INT (16 bits com sinal)		•	•
DBDI	DB tipo DINT (32 bits com sinal)		•	•
DBB	DB tipo BYTE (8 bits sem sinal)		•	•
DBW	DB tipo WORD (16 bits sem sinal)		•	•
DBD	DB tipo DWORD (32 bits sem sinal)		•	•
DBR	DB tipo REAL (32 bits ponto flutuante IEEE 754)		•	•
MSI	<i>Flag ou Memory</i> tipo SINT	•	•	•
MI	<i>Flag ou Memory</i> tipo INT	•	•	•
MDI	<i>Flag ou Memory</i> tipo DINT	•	•	•
MB	<i>Flag ou Memory</i> tipo BYTE	•	•	•
MW	<i>Flag ou Memory</i> tipo WORD	•	•	•
MD	<i>Flag ou Memory</i> tipo DWORD	•	•	•
MR	<i>Flag ou Memory</i> tipo REAL	•	•	•
A ou Q	Saída (<i>output</i>) tipo BYTE	•	•	•
E ou I	Entrada (<i>input</i>) tipo BYTE	•	•	•
T	Temporizador (<i>timer</i>) tipo WORD	•	•	•
Z ou C	Contador (<i>counter</i>) tipo WORD	•	•	•
SSI	<i>Special Flag ou Memory</i> tipo SINT	•		
SI	<i>Special Flag ou Memory</i> tipo INT	•		
SDI	<i>Special Flag ou Memory</i> tipo DINT	•		
SB	<i>Special Flag ou Memory</i> tipo BYTE	•		

ÁREA	ÁREA E TIPO DE DADOS	S7-200	S7-300	S7-400
SW	Special Flag ou Memory tipo WORD	•		
SD	Special Flag ou Memory tipo DWORD	•		
SR	Special Flag ou Memory tipo REAL	•		
VSI	Variable Memory ou Storage tipo SINT	•		
VI	Variable Memory ou Storage tipo INT	•		
VDI	Variable Memory ou Storage tipo DINT	•		
VB	Variable Memory ou Storage tipo BYTE	•		
VW	Variable Memory ou Storage tipo WORD	•		
VD	Variable Memory ou Storage tipo DWORD	•		
VR	Variable Memory ou Storage tipo REAL	•		

Exemplos

1. Leitura ou escrita de DB de número 10, com um tipo de dados **WORD** e *offset* 4 (quatro): Campo **Item** preenchido com o valor "DBW10:4".
2. Leitura ou escrita de *Output* de endereço 10: Campo **Item** preenchido com o valor "Q10".
3. Leitura ou escrita de *Special Memory* com um tipo de dados **REAL** e endereço 16: Campo **Item** preenchido com o valor "SR16".

NOTA

Ao utilizar Tags Bloco, cada Elemento de Bloco corresponde a uma posição de área múltipla ao tipo de dados, iniciando ao contagem a partir do endereço configurado.

Documentação das Interfaces de Comunicação

Enter topic text here. Esta seção contém a documentação das Interfaces de Comunicação referentes ao Driver **Prodave**.

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

Setup

Physical Layer: Ethernet Start driver OFFLINE

Timeout: 1000 ms Communication check time: 5000 ms

Connection management

Mode: Automatic (managed by the driver)

Retry failed connection every 20 seconds

Give up after 1 failed retries

Disconnect if non-responsive for 0 seconds

Logging Options

Log to File: C:\eeLogs\MicrolokII_%DATE%.log

File size limit (MB): 0 ('0' is unlimited)

Aba Setup

Opções gerais da aba Setup

OPÇÃO	DESCRIÇÃO
Physical Layer	Selecione a interface física em uma lista. As opções disponíveis são Serial , Ethernet , Modem e RAS . A interface selecionada deve ser configurada na aba específica
Timeout	Configure o <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
Communication check time	Configure o tempo, em milissegundos, para definir o intervalo em que a comunicação é considerada em estado inativo. Enquanto um Driver de Comunicação receber dados válidos, o estado de comunicação é considerado ativo. Porém, se durante o funcionamento um Driver de Comunicação não receber dados válidos neste período de tempo, o estado é considerado inativo. O estado de comunicação é mostrado no Tag IO.CommunicationStatus
Start driver OFFLINE	Selecione esta opção para que um Driver inicie em modo Offline ou parado. Isto significa que a interface de I/O não é criada até que se configure um Driver em modo Online utilizando-se um Tag em uma aplicação. Este modo possibilita a configuração dinâmica da interface de I/O em tempo de execução

Opções para o grupo Connection management

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

Opções para o grupo Logging Options

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

Configurações Gerais

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

Tags de Comunicação

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

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

IO.CommunicationStatus

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

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

IO.IOKitEvent

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

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

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

NOTA

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

IO.PhysicalLayerStatus

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

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

- **0**: Camada física parada, ou seja, um Driver está em modo **Offline**, a camada física falhou ao inicializar ou excedeu o número máximo de tentativas de reconexão
- **1**: Camada física iniciada mas não conectada, ou seja, um Driver está em modo **Online**, mas a camada física não está conectada. Se a opção **Connection management** está configurada com o valor **Automatic**, a camada física pode estar conectando, desconectando ou esperando por uma tentativa de reconexão. Se a opção **Connection management** está configurada com o valor **Manual**, então a camada física permanece neste estado até ser forçada a conectar
- **2**: Camada física conectada, ou seja, a camada física está pronta para ser usada. Isto **NÃO** significa que um equipamento esteja conectado, apenas que a camada de acesso está funcionando

IO.SetConfigurationParameters

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

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

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

No **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, 3×2). 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

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

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

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

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

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

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

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

IMPORTANTE

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

Propriedades

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

IO.ConnectionMode

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

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

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

IO.Stats.Partial.BytesSent

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

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

IO.Stats.Partial.TimeConnectedSeconds

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

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

IO.Stats.Partial.TimeDisconnectedSeconds

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

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

IO.Stats.Total.BytesRecv

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

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

IO.Stats.Total.BytesSent

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

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

IO.Stats.Total.ConnectionCount

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

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

IO.Stats.Total.TimeConnectedSeconds

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

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

IO.Stats.Total.TimeDisconnectedSeconds

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

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

Propriedades

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

Histórico de Revisões do Driver

VERSÃO	DATA	AUTOR	COMENTÁRIOS
2.0.5	15/05/2026	M. Ludwig	<ul style="list-style-type: none">• Driver atualizado para a biblioteca IOKit versão 3.0 e Visual Studio 2022 (<i>Case 38068</i>).
2.0.4	20/03/2017	M. Ludwig	<ul style="list-style-type: none">• Corrigida uma desordem na leitura ou escrita nas áreas M, S e V para tipos de dados de 16 bits ou maiores (<i>Case 22198</i>).
2.0.1	20/01/2015	M. Ludwig	<ul style="list-style-type: none">• Driver atualizado para comunicar com a biblioteca Prodrive MPI/IE versão 6.x (<i>Case 15813</i>).
1.1.1	20/07/2006	M. Ludwig	<ul style="list-style-type: none">• Atualização de documentação (<i>Case 5534</i>).
1.0.1	-	-	<ul style="list-style-type: none">• Todas as publicações anteriores ao controle de revisões.

Matriz

Rua Mostardeiro, 322/Cj. 902, 1001 e
1002

90430-000 — Porto Alegre — RS

Fone: (+55 51) 3346-4699

Fax: (+55 51) 3222-6226

E-mail: elipse-rs@elipse.com.br

Filial no Paraná

Av. Sete de Setembro, 4698/1708

80240-000 — Curitiba — PR

Fone: (+55 41) 4062-5824

E-mail: elipse-pr@elipse.com.br

Filial no Rio de Janeiro

Av. José Silva de A. Neto, 200/Bl. 4/Sl.
109B

22250-044 — Rio de Janeiro — RJ

Fone: (+55 21) 2430-5912

Suporte Técnico: (+55 21) 2430-5963

E-mail: elipse-rj@elipse.com.br

Filial em São Paulo

Rua dos Pinheiros, 870/Cj. 141 e 142

05422-001 — São Paulo — SP

Fone: (+55 11) 3061-2828

Fax: (+55 11) 3086-2338

E-mail: elipse-sp@elipse.com.br

Filial em Minas Gerais

Rua Antônio de Albuquerque, 156/705

30112-010 — Belo Horizonte — MG

Fone: (+55 31) 4062-5824

E-mail: elipse-mg@elipse.com.br

Filial em Taiwan

9F., No.12, Beiping 2nd St., Sanmin Dist.
807 — Kaohsiung City — Taiwan

Fone: (+886 7) 323-8468

Fax: (+886 7) 323-9656

E-mail: evan@elipse.com.br

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

www.elipse.com.br

kb.elipse.com.br

forum.elipse.com.br

www.youtube.com/elipsesoftware

elipse@elipse.com.br



Gartner, 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)