Driver Siemens MProt MPI, PPI e ISO/TCP

Nome do Arquivo	MProt.dll
Fabricante	Siemens AG, VIPA
Equipamentos	Modelos S7-200, S7-300, S7-400 e S7-1200 da Siemens, Speed7 da Vipa e demais equipamentos compatíveis com algum protocolo deste Driver
Protocolo	PPI e MPI (Serial), MPI encapsulado em Ethernet e ISO sobre TCP (RFC 1006 ou S7-TCP/IP em interface Ethernet)
Versão	4.0.40
Última Atualização	03/11/2026
Plataforma	Win32
Dependências	IOKit versão 2.0 ou superior
Leitura com Superblocos	Sim
Nível	0

Introdução

O Driver Siemens multiprotocolo (M-Prot) comunica com os equipamentos S7-200, S7-300, S7-400 e S7-1200 da Siemens e Speed7 da VIPA usando os protocolos Siemens PPI, MPI, ISO/TCP e MPI encapsulados em Ethernet (IBHLink).

O protocolo **PPI** deve ser usado somente para a linha S7-200 usando o cabo conversor RS232-PPI/MPI fornecido pela Siemens.

O protocolo **MPI** pode ser usado para as linhas S7-300 e S7-400 através de um cabo conversor RS232-PPI/MPI fornecido pela Siemens, ou também para a linha Speed7 da VIPA na porta MPI usando um cabo RS-232 comum.

O protocolo **ISO/TCP**, que também pode ser chamado por **ISO sobre TCP**, **RFC1006** ou **S7-TCP/IP** em diversos materiais dos fabricantes de hardware, pode ser usado para os modelos S7-300 ou S7-400 da Siemens através do uso de um cartão Ethernet CP-3XX, CP-433 ou CP-443. Para o modelo S7-1200, e também para a linha Speed7 da VIPA, diretamente na porta Ethernet da CPU. Para o modelo S7-200, há uma variação especial do protocolo **ISO/TCP** para ser usado em conjunto com a interface CP-243. Este protocolo é denominado **ISOTCP243**.

Para os equipamentos que não possuem porta Ethernet, uma alternativa pode ser o uso de um conversor Ethernet/MPI IBHLink, fornecido pelas empresas IBH Softec ou Hilscher, que atua no nível FDL. A vantagem deste conversor é que a velocidade nominal é mais alta, chegando a 187 kbps na rede MPI, ao passo que pelo conversor serial é de 38.4 kbps. O uso deste conversor é uma alternativa ao uso das placas CP5611 ou similares.

Outra alternativa semelhante é o cabo conversor NETLink PRO Eth da Softing, que converte de **ISO/TCP** para **MPI**.

Este Driver não suporta o uso de adaptadores Siemens PPI/MPI por interface USB.

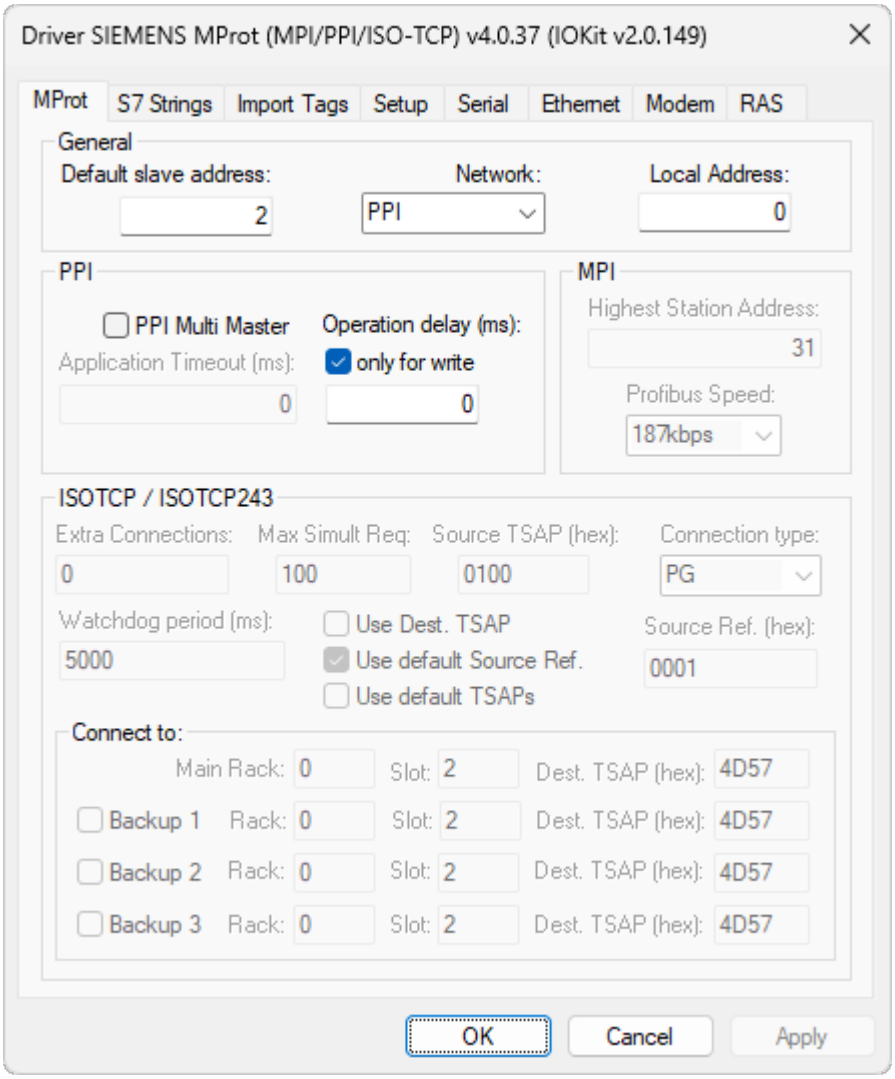
Este Driver não suporta o uso de interfaces CP5611 ou similares para o acesso à rede MPI. Para comunicar com estas placas deve ser usado o Driver S7Functions ou ainda o SIMATIC.NET da Siemens, através do servidor OPC já incluso.

NOTA

M-Prot é um nome criado pela **Elipse Software** com a finalidade de especificar um Driver que suporta múltiplos protocolos. Não existe relação alguma com nomes de equipamentos, protocolos ou padrões definidos pelos fabricantes supracitados.

Configuração do Driver

Os parâmetros **[P]** de configuração deste Driver não são utilizados. Todas as configurações são executadas na janela de configurações, mostrada na figura a seguir.



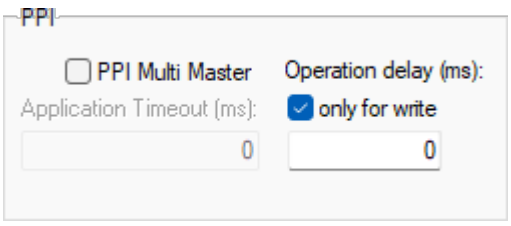
Aba MProt

As opções disponíveis para o grupo **General** estão descritas na tabela a seguir.

Opções disponíveis no grupo General

OPÇÃO	DESCRIÇÃO
Default Slave Address	Pode ser utilizado como o endereço padrão para qualquer Tag. Deixe o parâmetro *N1* em 0 (zero) para que seja substituído pelo endereço padrão
Network	Seleção do protocolo. As opções disponíveis são **PPI**, **MPI**, **ISOTCP**, **ISOTCP243** ou **IBHLinkMPI**. Os grupos **PPI**, **MPI** ou **ISOTCP/ISOTCP243** nesta aba são habilitados ou desabilitados conforme a seleção de um protocolo
Local Address	Endereço deste Driver na rede. Pode ser selecionado arbitrariamente

As opções disponíveis para o grupo **PPI** estão descritas na tabela a seguir.



Grupo PPI

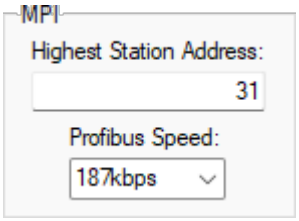
Opções disponíveis para o grupo PPI

OPÇÃO	DESCRIÇÃO
PPI Multi Master	Informa a este Driver que existem outros **Mestres** na rede
Application Timeout (ms)	Tempo máximo de comunicação para cada Tag, em milissegundos. Disponível apenas quando este Driver é Multi-Mestre
Operation delay (ms)	Tempo de parada para o intervalo entre operações de comunicação, em milissegundos. Selecione a opção **only for write** para indicar a aplicação deste intervalo apenas para operações de escrita. Para mais informações, consulte a nota a seguir

NOTA

A opção **Operation delay (ms)** adiciona um tempo mínimo de espera que deve haver entre o fim de uma operação de leitura ou escrita e o início de outra operação. Utilize um valor diferente de 0 (zero) nesta configuração apenas em caso de falhas de comunicação ocasionadas pela inércia de processamento de um equipamento. As operações de escrita são as mais prejudicadas, pois normalmente são aleatórias, e para estas operações existe a opção **only for write**. Se esta opção não está selecionada, a espera aplica-se à operações de leitura e de escrita. Se está selecionada, aplica-se apenas às operações de escrita, que é o recomendado. Note que o acréscimo de um tempo de espera pode diminuir o desempenho de uma aplicação.

As opções disponíveis para o grupo **MPI** estão descritas na tabela a seguir.

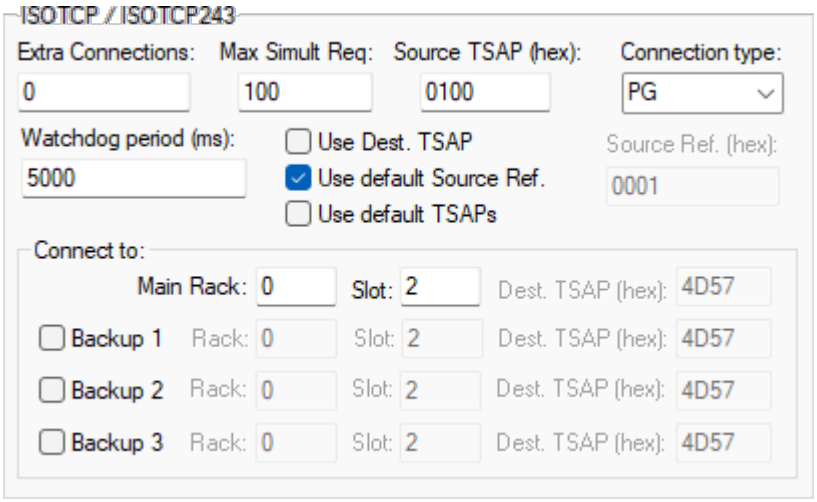


Grupo MPI

Opções disponíveis para o grupo MPI

OPÇÃO	DESCRIÇÃO
Highest Station Address	Indica o maior endereço disponível na rede, para que nos modos **PPI** e **MPI** este Driver encontre outros possíveis **Mestres** na rede. Devem ser inseridos apenas os valores **15**, **31** ou **63**
Profibus Speed	Velocidade nominal de uma rede Profibus

As opções disponíveis para o grupo **ISOTCP / ISOTCP243** estão descritas na tabela a seguir.



Grupo ISOTCP / ISOTCP243

Opções disponíveis para o grupo ISOTCP / ISOTCP243

OPÇÃO	DESCRIÇÃO
Extra Connections	Número de conexões TCP adicionais que podem ser criadas para melhorar o desempenho da comunicação
Max Simult Req	Número máximo de variáveis simultâneas em um único pedido. Esta configuração pode ser usada para melhorar significativamente o desempenho da comunicação, desde que se utilize um valor maior que 0 (zero) na opção **Extra Connections**
Source TSAP (hex)	Número formado por um **Word** em hexadecimal que identifica o TSAP local no protocolo
Connection type	Tipo da conexão. Os valores possíveis são **PG**, **OP** ou **PC**. Deve ser selecionada de acordo com a configuração na CPU
Watchdog period (ms)	Intervalo, em milissegundos, para expiração do **Tag de Watchdog**. Se este Tag existe e o valor não varia neste intervalo, este Driver realiza a troca de endereço IP
Use Dest. TSAP	Se esta opção está selecionada, habilita a entrada de valores de **Dest. TSAP**. Se esta opção não está selecionada, habilita a entrada de valores de **Rack**, **Slot** e **Connection type**
Use default TSAPs	Se esta opção está selecionada, utiliza valores TSAP padrão e assim desabilita a entrada de valores de configuração de **Source TSAP**, **Dest. TSAP**, **Connection Type**, **Rack** e **Slot**
Source Ref. (hex)	Número formado por um **Word** em hexadecimal que identifica a referência local (*source reference*) no protocolo. Somente é liberada quando a opção **Use default Source Ref** não está selecionada
Rack	Número do *rack* da CPU principal de destino

OPÇÃO	DESCRIÇÃO
Slot	Número do *slot* da CPU principal de destino
Dest. TSAP (hex)	Número formado por um **Word** em hexadecimal que identifica o TSAP de destino no protocolo
Backup 1 / Backup 2 / Backup 3	Habilita a entrada de valores de *rack*, *slot* ou TSAP de destino da(s) respectiva(s) CPU(s) de *backup*, para uso em sistemas com redundância que sejam de valores diferentes da CPU principal

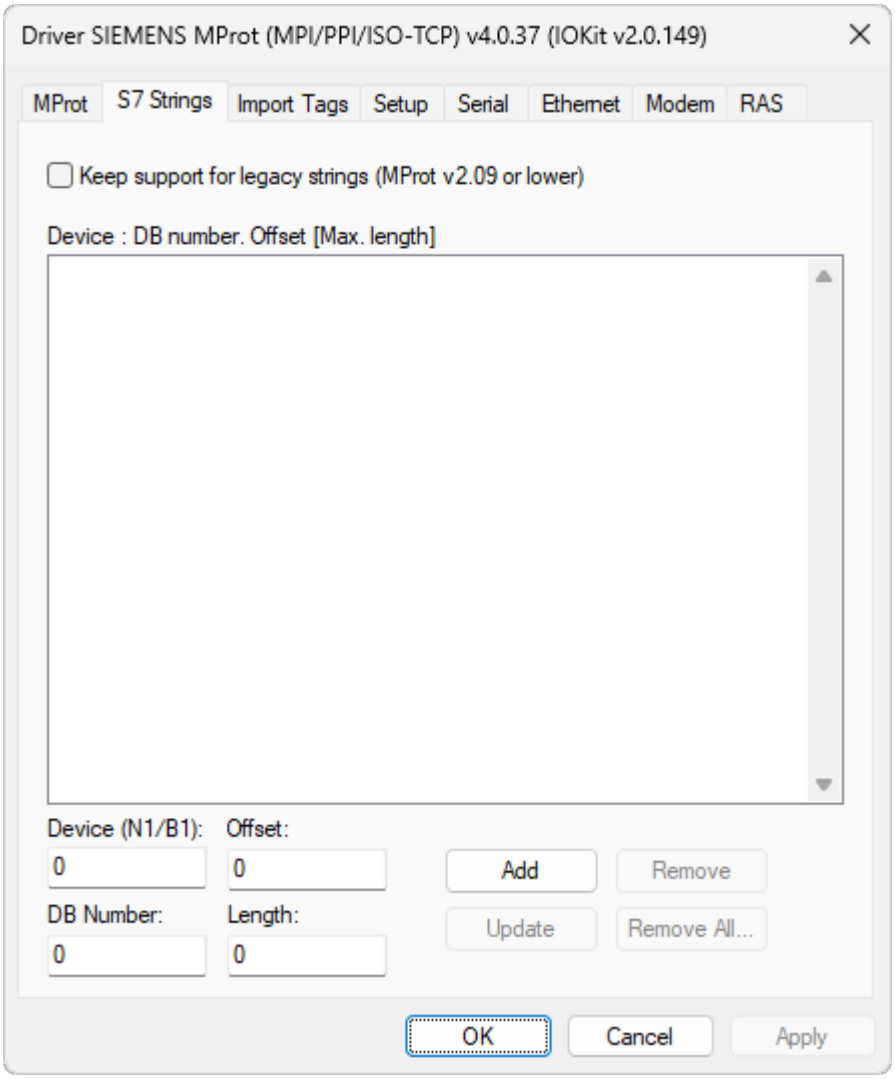
Para que a comunicação deste Driver funcione com o modelo de equipamento Siemens S7-1200, selecione a opção **ISOTCP**, desmarque a opção **Use default TSAPs**, configure a propriedade **Source TSAP (hex)** para o valor "0100" e defina as opções **Connection type** como **PG**, **Rack** com o valor 0 (zero) e **Slot** com o valor 1 (um).

NOTAS

- Ao selecionar os protocolos **ISOTCP** ou **ISOTCP243**, todos os Tags no objeto Driver de Comunicação devem ter o parâmetro *N1* ou *B1* configurado em 0 (zero) e o parâmetro **Default Slave Address** também configurado em 0 (zero).
- Os parâmetros **Source Ref** e **Source TSAP** devem ser considerados apenas em casos muito particulares. Devido à execuções bem sucedidas em uma enorme gama de topologias, recomenda-se manter a opção **Use default Source Ref** sempre selecionada e o valor de **Source TSAP** sempre em "0100".
- Quando a opção **Use Default TSAPs** está selecionada em conjunto com o protocolo **ISOTCP**, o valor de **Source TSAP** é "0100" e o valor de **Dest. TSAP** utilizado é "0202".
- TSAP vem do inglês *Transport Service Access Point*, e é uma terminologia utilizada no protocolo ISO.
- Quando se utiliza adaptadores seriais PC - PPI/MPI, tem sido bastante usual a necessidade de configurar o *handshaking* na aba **Serial** da janela de configurações deste Driver. Apenas o controle de **RTS** deve ser configurado em **ON**. Havendo insucesso de comunicação nos testes iniciais com este Driver, convém experimentar esta mudança, **RTS Control** configurado como **ON** e refazer o teste.

Configuração de Strings

Esta aba é útil caso haja necessidade de declarar **Strings** ou **WStrings** com um tamanho máximo definido.



Aba S7 Strings

A opção disponível na aba **S7 Strings** está descrita na tabela a seguir.

Opção disponível na aba S7 Strings

OPÇÃO	DESCRIÇÃO
Keep support for legacy strings (MProt v2.09 or lower)	Mantém o suporte para **Strings** antigas, anteriores à versão **2.10**. Esta opção selecionada mantém o formato antigo de **String** implementado em versões anteriores, evitando transtornos em uma migração de versão deste Driver. Recomenda-se selecionar esta opção apenas em caso de migração de um projeto cuja versão deste Driver seja igual a **2.9** ou anterior. Caso o projeto utilize **Strings** após uma migração de versão, os Tags de tipo **String** retornam erros de leitura de um equipamento. O formato de **Strings** legado possui uma reserva de 32 bytes de espaço a partir do *offset* configurado. Caso esteja trabalhando com um projeto novo, deixe esta opção desmarcada

Listagem de Tamanhos Máximos de Strings

Esta aba também mostra uma lista selecionável com **Strings** declaradas com tamanhos já determinados. A lista aparece em branco caso não existam **Strings** configuradas. As **Strings** podem ser declaradas na memória de um equipamento das seguintes formas:

- Sem especificar o tamanho máximo na declaração, conforme o exemplo a seguir.

```
STRING var;
```

A **String** é alocada automaticamente com a extensão máxima padrão de um equipamento.

- Especificando o tamanho máximo na declaração, conforme o exemplo a seguir.

```
STRING var[50];
```

No exemplo anterior, a **String** é alocada com tamanho máximo de "50". Devido a esta segunda forma é que a listagem de tamanho de **Strings** se mostra importante.

Para determinar o tamanho de uma **String** declarada, é preciso preencher corretamente todos os campos, conforme descrito na tabela a seguir.

Opções disponíveis para a configuração do tamanho máximo de Strings

OPÇÃO	DESCRIÇÃO
Device	Endereço de um equipamento. Preencha com o mesmo valor do parâmetro *N1* ou *B1* de um Tag. Para mais informações, consulte o tópico **Configuração por Parâmetros Numéricos [N/B]**
DB Number	Informe o valor do número do DB onde a **String** está localizada
Offset	Informe o valor do *offset* no DB onde a **String** está localizada
Length	Preencha com o valor máximo de tamanho da **String**, conforme declarado na programação de um equipamento

Caso já exista alguma **String** declarada na lista com mesmo valor de **Device**, **DB Number** e **Offset**, esta é automaticamente apontada na tabela e os valores são carregados em todos os campos de edição. As opções que se destinam à manipulação dos dados das **Strings** da lista são as seguintes:

- **Add**: Adiciona novos parâmetros
- **Update**: Altera parâmetros já listados
- **Remove**: Remove totalmente uma linha de parâmetros

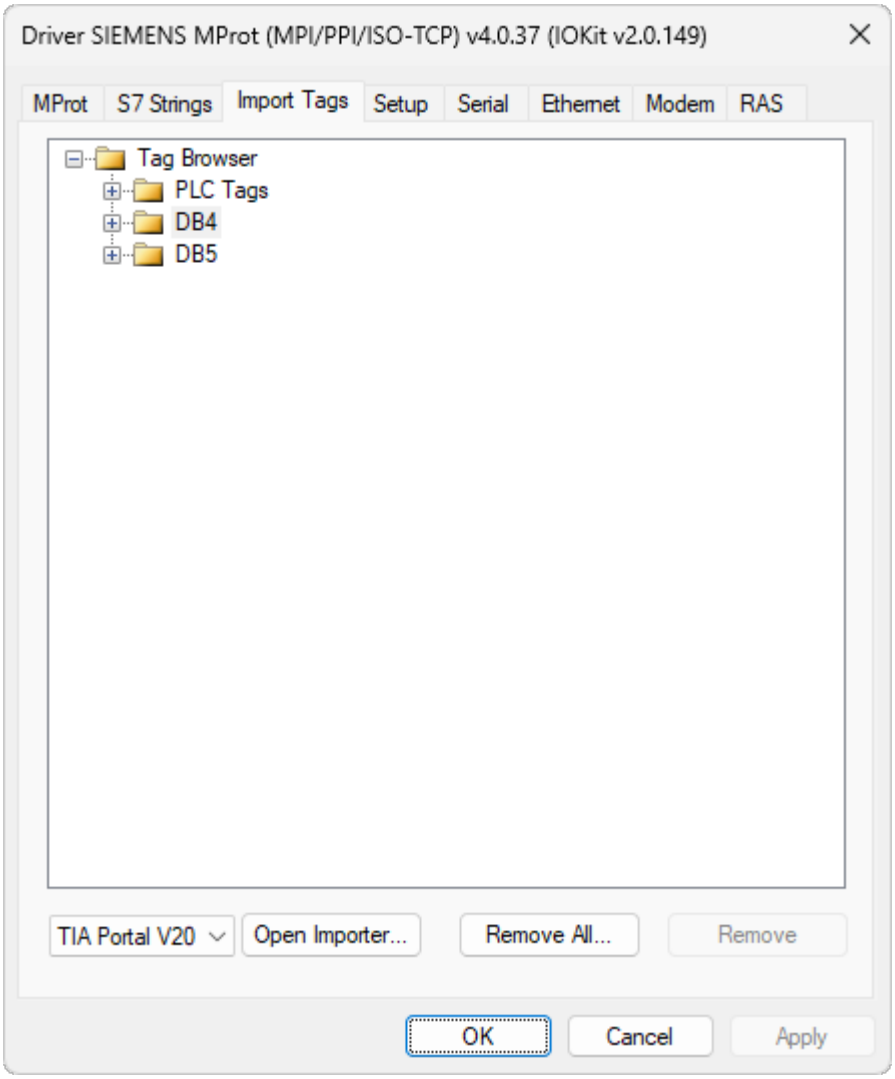
Clique em **OK** para confirmar todas as configurações listadas e fechar a janela. Clique em **Remove All** para remover todos os dados da lista.

NOTA

Ao declarar Tags com parâmetros sintáticos, não há necessidade de preencher a lista com declarações de **Strings**. O tamanho pode ser especificado no próprio parâmetro sintático disponível em um Tag.

Importação de Tags

A aba **Import Tags** traz algumas funcionalidades de importação de Tags a partir de um projeto de automação do **TIA Portal** (*Totally Integrated Automation Portal*).



Aba Import Tags

A aba **Import Tags** exibe uma estrutura de árvore dos Tags já importados e salvos no projeto deste Driver. Ao incluir um Driver em um projeto, o estado inicial é encontrar esta estrutura vazia, apenas com a pasta **Tag Browser**. Pode-se incluir Tags importados selecionando a versão do TIA Portal e usando a opção **Open Importer**, descrita no tópico **MProt TIA Portal Importer**.

Caso já existam Tags importados nesta estrutura de árvore, pode-se usar a opção **Remove** para remover um Tag selecionado e a opção **Remove All** para remover todos os Tags importados. Clique em **OK** para salvar todos os Tags e exibi-los no **Tag Browser**.

NOTAS

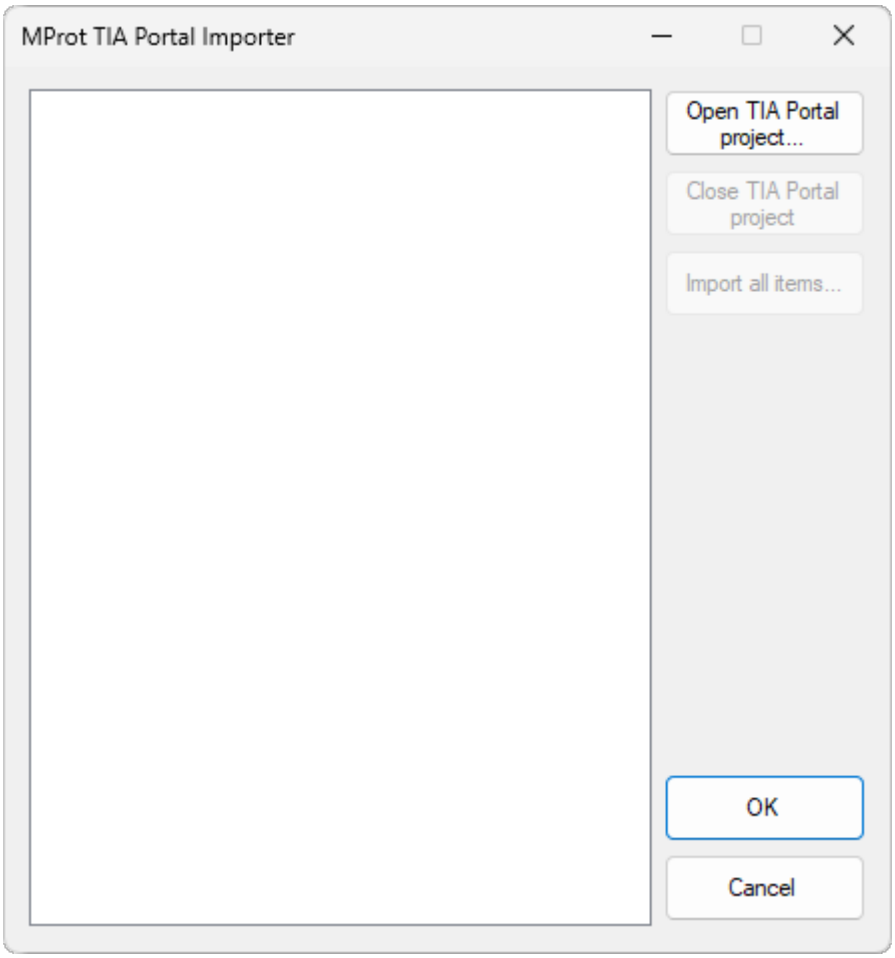
- Apenas o tipo de dados **LReal** não é suportado na importação de Tags. Caso haja variáveis deste tipo de dados em um projeto, estas são ignoradas pelo importador, que prossegue com a importação das variáveis dos demais tipos de dados.
- Blocos dos tipos de dados **ArrayDB** e **InstanceDB** não são suportados na importação de Tags.

MProt TIA Portal Importer

O **MProt TIA Portal Importer** é um arquivo executável distribuído junto com este Driver, destinado à importação de Tags de um projeto de automação criado pelo TIA Portal, da Siemens AG. Para o correto funcionamento, este executável deve atender aos seguintes pré-requisitos:

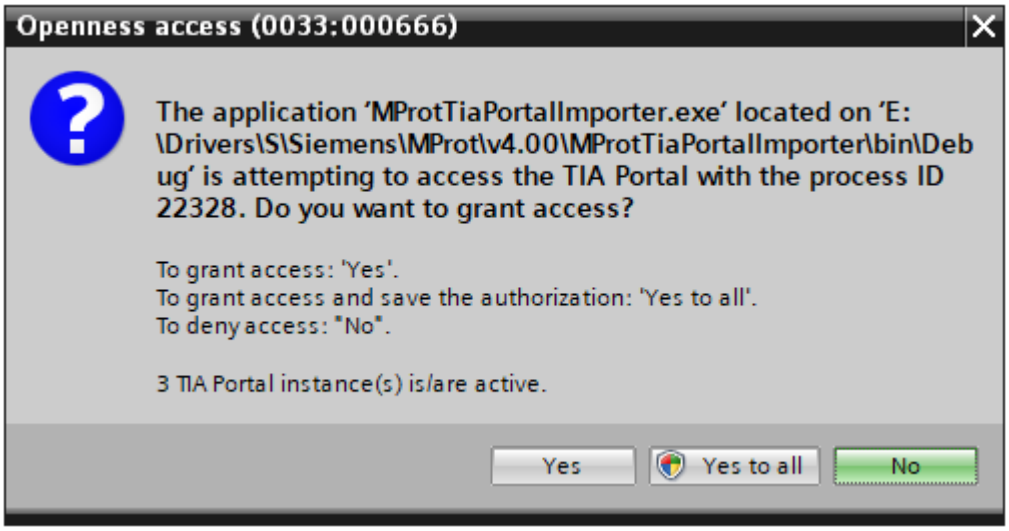
- Siemens TIA Portal versões 16, 17, 18, 19 ou 20 instalado
- Microsoft .NET Framework versão **4.8** ou superior instalado
- As pastas **TIAImporterV16**, **TIAImporterV17**, **TIAImporterV18**, **TIAImporterV19** e **TIAImporterV20**, fornecidas com este Driver, devem estar localizadas na mesma pasta deste Driver
- Os executáveis MProtTiaPortalImporter.exe e os arquivos Siemens.Engineering.dll, fornecidos com este Driver, devem estar localizados dentro das pastas supracitadas
- A conta de usuário que utiliza este Importador deve ser adicionada ao grupo local de usuários **Siemens TIA Openness**, localizado nas Ferramentas Administrativas do Windows

O MProt TIA Portal Importer é iniciado clicando em **Open Importer** na aba **Import Tags** da janela de configurações deste Driver para abrir a janela da figura a seguir.



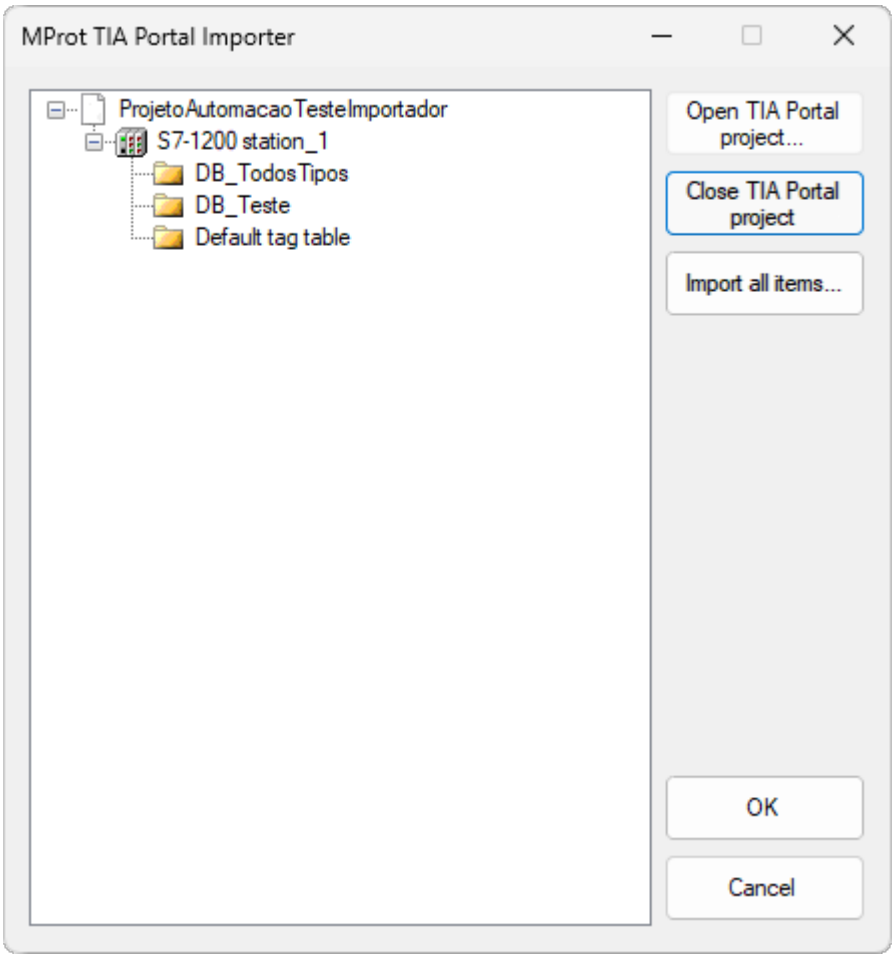
MProt TIA Portal Importer

Clique em **Open TIA Portal project** para selecionar um arquivo de projeto TIA Portal e clique em **OK** para abrir uma instância do TIA Portal em segundo plano. Caso o *firewall* do TIA Portal não encontre o **MProt TIA Portal Importer** na lista de autorizações, uma caixa de diálogo é aberta com as opções **Yes** para permitir o acesso, **Yes to all** para permitir o acesso e salvar a autorização de acesso na lista de autorizações, e **No** para recusar o acesso, conforme a figura a seguir.



Configuração de acesso

Após autorizar o **MProt TIA Portal Importer**, este é aberto exibindo a estrutura de árvore de um projeto com os itens de interesse para a importação.



Itens para importação

Para importar todos os Tags de um projeto, clique em **Import all items**. Após concluir a importação, clique em **OK** para confirmar a importação e fechar o Importador. Ao fechar o Importador, os Tags são automaticamente repassados para a janela de configurações deste Driver, onde podem ser salvos. Para mais informações, consulte o tópico **Importação de Tags**. A opção **Close TIA Portal project** permite fechar um projeto e a opção **Cancel** fecha o Importador sem realizar a importação.

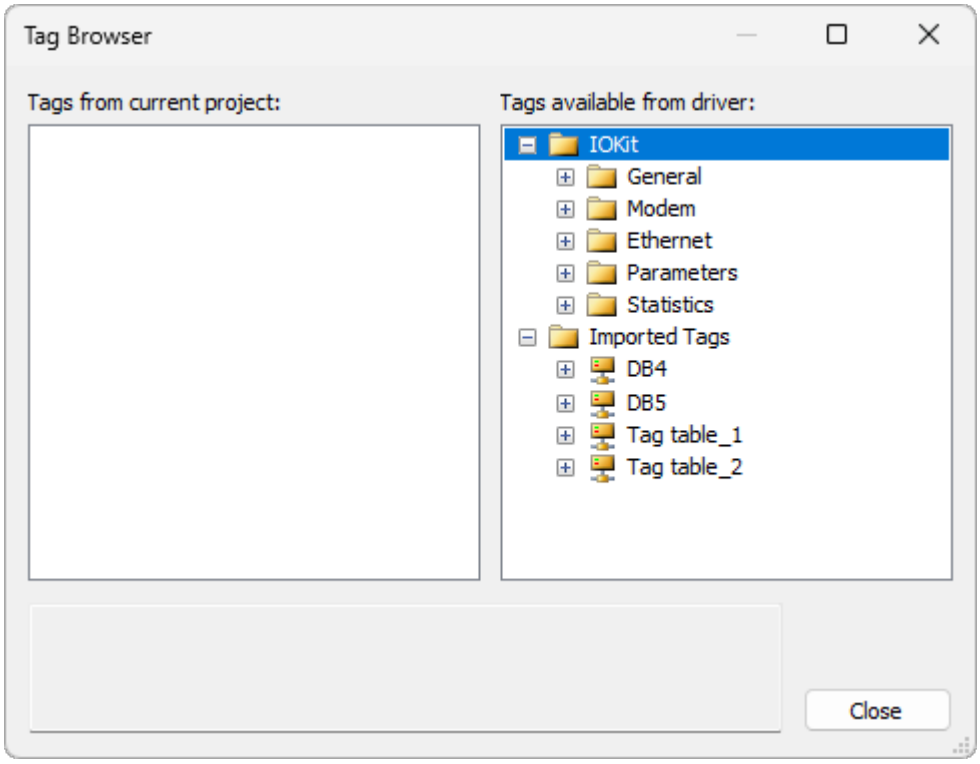
Referência de Tags

Esta seção contém informações de configuração de Tags por **Parâmetros Sintáticos** e por **Parâmetros Numéricos** (parâmetros **[N/B]**). Também contém referências aos **Tags da Interface em Conexões ISOTCP Extras**.

Utilizando o Tag Browser

O **Elipse E3** a partir da versão **2.0**, o **Elipse Power** e o **Elipse Water** possuem uma ferramenta chamada **Tag Browser**, que permite que um Driver auxilie o usuário na criação e configuração de Tags.

O Tag Browser visa automatizar a configuração dos Tags preenchendo automaticamente o campo **Item** e o tamanho em Elementos adequado, caso seja um Tag Bloco.



Janela Tag Browser

A listagem **Tags from current project** (*Tags do projeto corrente*) mostra os Tags e pastas que existem no projeto corrente. A listagem **Tags available from driver** (*Tags disponibilizados pelo driver*) mostra uma árvore com os Tags disponíveis em um Driver. Para criar um Tag em uma aplicação, arraste um dos Tags definidos por este Driver para a pasta desejada no diretório corrente. O Driver Siemens MProt MPI, PPI e ISO/TCP contém os seguintes nós nesta árvore:

- O nó **IOKit** mostra os Tags disponíveis na biblioteca **IOKit**, divididos nas seguintes categorias:
  - **General**: Tags de uso geral
  - **Modem**: Tags de manipulação da comunicação via modem
  - **Ethernet**: Tags de manipulação da comunicação via rede Ethernet
  - **Parameters**: Tags para configuração dos parâmetros da biblioteca **IOKit**
- O nó **Imported Tags** mostra todos os Tags importados e salvos na aba **Import Tags**. Este nó somente aparece se realmente há Tags importados. Neste caso o nó possui uma pasta para cada grupo separado de Tags importados. Se há um *Data Block* (DB), os Tags referentes às variáveis deste DB são agrupadas em uma pasta em referência a este DB. Os demais Tags são agrupados em pastas referentes à tabela a qual pertencem. Cada Tag é criado com a propriedade **ParamDevice** configurada com um valor padrão 0 (zero)

Configuração por Parâmetros Sintáticos

Utilize a sintaxe a seguir para cada campo presente no **Elipse E3**, **Elipse Power** ou **Elipse Water**:

- **Dispositivo**: Insira o endereço de um equipamento na rede. Se este endereço é igual a 0 (zero) e o protocolo é diferente de **ISOTCP** ou **ISOTCP243**, este valor é substituído pelo valor configurado na opção **Default Slave Address**. Se o protocolo é igual a **ISOTCP** ou **ISOTCP243**, este valor deve ser deixado em 0 (zero). O campo **Dispositivo** também pode permanecer vazio, desde que seja inserido no campo **Item** antes do caractere de dois pontos
- **Item**: Este campo deve obedecer alguma das sintaxes a seguir

Use a sintaxe geral a seguir se a área não é igual a **DB**. Valores entre colchetes são opcionais:

```
<[Dispositivo:]><Área><[Tipo]><Endereço>[.Bit]
```

Em que:

- **Dispositivo**: Endereço de um equipamento conforme exposto no item **Dispositivo**, caso não tenha sido informado naquele campo
- **Área**: Área de dados dentro de um equipamento. As seguintes opções podem ser utilizadas:
  - **S**
  - **SM**
  - **AI** (*Analog Input*)
  - **AQ** (*Analog Output*)
  - **C** (*Counter*)
  - **T** (*Timer*)
  - **I** (*Digital Input*)
  - **Q** (*Digital Output*)
  - **M** (*Memory*)
  - **HC** (*High Speed Counter*)
- **Tipo**: Tipo de dados a ler. A tabela a seguir mostra os possíveis símbolos para os tipos de dados

Opções disponíveis para tipos de dados

TIPO DE DADOS	SIGNIFICADO
X	Usado para ler ou escrever um bit de um byte
B	**Byte**
W	**Word**
D	**DWord**
I	**Int**
LI	**DInt**
F	**Float**

TIPO DE DADOS	SIGNIFICADO
S	**String**
WS	**WString**
S5T	**S5Time**
RB	**Byte** (LSB First)
RW	**Word** (LSB First)
RD	**DWord** (LSB First)
RI	**Int** (LSB First)
RLI	**DInt** (LSB First)
SW	**Word** (byte menos significativo primeiro)
SD	**DWord** (byte menos significativo primeiro)
SI	**Int** (byte menos significativo primeiro)
SLI	**DInt** (byte menos significativo primeiro)

NOTAS

- Tipos de dados **LSB First** consideram que a ordem dos bits é o menos significativo primeiro e a interpretação é o reverso dos demais tipos de dados.
- Tipos de dados com **byte menos significativo primeiro** consideram que a ordem dos bytes é o byte menos significativo primeiro e o mais significativo por último.

- **Endereço**: Endereço numérico a ler
- **Bit**: Opcional que informa o bit de uma palavra a ler ou a escrever, entre 0 (zero) e 31

Exemplo:

```
(Dispositivo 4, bit 1 da memória do endereço 10)
Device: Vazio - Item 4:M10.1
```

Se a área de dados é igual a **DB**, também conhecida como **V**, use a sintaxe a seguir. Valores entre colchetes são opcionais:

```
<[Dispositivo:]>DB<NumeroDB>:<Tipo><Endereço><[.Bit]>
```

- **Dispositivo**: Refere-se ao mesmo item opcional da sintaxe geral
- **NumeroDB**: Coloque o número do **DB**. Caso a memória contenha um bloco **DB** único ou não especificado, preencha com o valor 1 (um)
- **Endereço**: Endereço numérico (*offset*) a ler
- **Bit**: Valor opcional que informa o bit do tipo de dados a ler ou a escrever, entre 0 (zero) e 31

Opções disponíveis para tipos de dados de áreas de dados DB

TIPO DE DADOS	SIGNIFICADO
DBX	Usado para ler ou escrever um bit de um byte em um **DB**
DBB	Usado para ler ou escrever um byte em um **DB**
DBW ou DW	Usado para ler ou escrever um **Word** em um **DB**
DBD ou DD	Usado para ler ou escrever um **Double Word** em um **DB**
DBI ou DI	Usado para ler ou escrever um **Int** em um **DB**
DBLI ou DLI	Usado para ler ou escrever um **DInt** em um **DB**
DBF ou DF	Usado para ler ou escrever um **Floating Point** (real de 32 bits) em um **DB**
DBS ou DS	Usado para acessar uma **String** em um **DB**
DBWS ou DWS	Usado para acessar uma **WString** em um **DB**
DBS5T	Usado para acessar um *timer* do tipo de dados **S5Time** em um **DB**
RDBB	Usado para ler ou escrever um byte em um **DB** (LSB First)
RDBW ou RDW	Usado para ler ou escrever um **Word** em um **DB** (LSB First)
RDBD ou RDD	Usado para ler ou escrever um **Double Word** em um **DB** (LSB First)
RDBI ou RDI	Usado para ler ou escrever um **Int** em um **DB** (LSB First)
RDBLI ou RDLI	Usado para ler ou escrever um **DInt** em um **DB** (LSB First)
SDBW ou SDW	Usado para ler ou escrever um **Word** em um **DB** (byte menos significativo primeiro)
SDBD ou SDD	Usado para ler ou escrever um **Double Word** em um **DB** (byte menos significativo primeiro)
SDBI ou SDI	Usado para ler ou escrever um **Int** em um **DB** (byte menos significativo primeiro)
SDBLI ou SDLI	Usado para ler ou escrever um **DInt** em um **DB** (byte menos significativo primeiro)

Exemplos:

```
(Dispositivo 2, Word começando no endereço 20 do DB1)
Device: 2 - Item: DB1:DW20
(Mesmo do anterior, porém Device foi informado no campo Item)
Device: Vazio - Item: 2:DB1:DW20
(Dispositivo 7, DB 5, bit 2 do byte 7)
Device: Vazio - Item: 7:DB5:DBX7.2
```

A sintaxe para tipos de dados **String** ou **WString** em áreas de dados **DB** é a seguinte:

```
<[Dispositivo:]>DB<NumeroDB>:DBS<Endereço><[Tamanho máximo]>
```

Em que:

- **Device, NumeroDB e Endereço**: Referem-se aos mesmos itens da sintaxe geral

- **Tamanho máximo**: Opcional que informa o tamanho máximo declarado na **String** ou **WString**. Se não é informado, considera-se o tamanho máximo padrão da **String** (254 caracteres) ou da **WString** (16382 caracteres).

Exemplos de sintaxe para **Strings**:

```
(Dispositivo 2, String começando no endereço 16 do DB17,
usando o tamanho padrão máximo de um dispositivo)
Device: 2 - Item: DB17:DBS16
(Mesmo do anterior, porém Device foi informado no campo Item
e com tamanho máximo alocado de 25 caracteres)
Device: Vazio - Item: 2:DB17:DBS16[25]
(Dispositivo 4, String começando no endereço 100 do DB10,
com tamanho máximo alocado de 50 caracteres)
Device: Vazio - Item 4:DB10:DS100[50]
```

Configuração por Parâmetros Numéricos [N/B]

Use a sintaxe padrão descrita na tabela a seguir para todos os Tags.

Sintaxe padrão para Tags

PARÂMETRO	DESCRIÇÃO
N1 ou B1	Endereço de um equipamento. Se igual a 0 (zero) e protocolo diferente de **ISOTCP** ou **ISOTCP243**, é substituído pelo valor configurado na opção **Default Slave Address**. Se o protocolo é igual a **ISOTCP** ou **ISOTCP243**, este valor deve ser deixado em 0 (zero)
N2 ou B2	Tipo de dados e área de dados. Para mais informações, consulte as tabelas a seguir. O valor deve ser composto pelo tipo de dados multiplicado por 100 mais a área de dados (a fórmula é *N2 ou B2 = TipoDados × 100 + Área*)
N3 ou B3	Se a área de dados selecionada é **V** (**DB**), preencha com o número do bloco **DB**. Caso contrário, deixe em 0 (zero). Caso a memória contenha um bloco **DB** único ou não especificado, preencha com o valor 1 (um)
N4 ou B4	Endereço na área de dados ou *offset* do bloco **DB**. Para usar tipos de dados que ocupam mais de 1 (um) byte, devem ser colocados endereços múltiplos de 2 (dois) para tipos de dados de dois bytes (16 bits com e sem sinal) e múltiplos de 4 (quatro) para tipos de dados de quatro bytes (32 bits com e sem sinal e ponto flutuante de 32 bits)

Opções disponíveis para tipos de dados

TIPO DE DADOS	SIGNIFICADO
0	Padrão da área de dados
1	**BOOL** (Booleano)
2	**BYTE** (oito bits sem sinal)
3	**WORD** (16 bits sem sinal)
4	**INT** (16 bits com sinal)
5	**DWORD** (32 bits sem sinal)

TIPO DE DADOS	SIGNIFICADO
6	**DINT** (32 bits com sinal)
7	**REAL** (32 bits de ponto flutuante, IEEE 754)
8	**STRING**
12	**S5TIME** (tempo em segundos, 32 bits de ponto flutuante, IEEE 754, consulte a **nota** a seguir)
13	**WSTRING**
32	**BYTE** (oito bits sem sinal, LSB First)
33	**WORD** (16 bits sem sinal, LSB First)
34	**INT** (16 bits com sinal, LSB First)
35	**DWORD** (32 bits sem sinal, LSB First)
36	**DINT** (32 bits com sinal, LSB First)
43	**WORD** (16 bits sem sinal, byte menos significativo primeiro)
44	**INT** (16 bits com sinal, byte menos significativo primeiro)
45	**DWORD** (32 bits sem sinal, byte menos significativo primeiro)
46	**DINT** (32 bits com sinal, byte menos significativo primeiro)

Opções disponíveis para áreas de dados

ÁREA DE DADOS	SIGNIFICADO
0	**S**
1	**SM**
2	**AI** (*Analog Input*)
3	**AQ** (*Analog Output*)
4	**C** (*Counter*)
5	**T** (*Timer*)
6	**I** (*Digital Input*)
7	**Q** (*Digital Output*)
8	**M** (*Memory*)
9	**V** (*DB*)
10	**HC** (*High Speed Counter*)

NOTAS

- Tipos de dados **LSB First** consideram que a ordem dos bits é o menos significativo primeiro e a interpretação é o reverso dos demais tipos de dados.
- Tipos de dados com **byte menos significativo primeiro** consideram que a ordem dos bytes é o byte menos significativo primeiro e o mais significativo por último.
- Para tipos de dados **S5Time**, o valor a preencher é sempre em segundos, com ponto flutuante de 32 bits. A gama de valores diferentes de 0 (zero) está entre 0,01 e 9990,0 segundos. A base de tempo é preenchida ou interpretada automaticamente.
- No protocolo **PPI** há uma limitação em um Tag Bloco para dados em bytes. Para leitura, o máximo permitido são 224 bytes, e para escrita são 218 bytes. Isto significa, respectivamente, que para tipos de dados **Word** (16 bits), um Tag Bloco não pode ultrapassar 112 e 109 Elementos. Para tipos de dados **DWord** (32 bits), um Tag Bloco não pode ultrapassar 56 e 54 Elementos, e assim por diante.
- Caso desconheça a definição de **Rack** e **Slot** para endereçamento dos Tags em protocolo **ISOTCP**, consulte o artigo *KB-39019: Configurações de Rack e Slot* no **Elipse Knowledgebase**.

Tags da Interface em Conexões ISOTCP Extras

Ao optar pelo uso de conexões ISOTCP extras com o parâmetro **Extra Connections** na **janela de configurações deste Driver**, estas conexões podem ser controladas e monitoradas pelos Tags específicos de Interface **Physical Layer Status**, **IPSelect** e **IPSwitch**.

NOTA

Estes Tags não podem ser utilizados quando o parâmetro **Extra Connections** é igual a 0 (zero). Neste caso, utiliza-se os Tags da biblioteca **IOKit** correspondentes, de mesmo nome, cujo uso pode ser consultado no tópico **Documentação das Interfaces de Comunicação**.

Physical Layer Status (MProt)

Somente Leitura

Configuração por parâmetros numéricos

PARÂMETRO	VALOR
N1	-2 (menos dois)
N2	0 (zero)
N3	0 (zero)
N4	2 (dois)

Configuração por parâmetros sintáticos

PARÂMETRO	VALOR
Item	MProt.IO.PhysicalLayerStatus

Este Tag indica o estado da conexão na camada física. Os valores possíveis são **0**: Camada física desconectada ou **1**: Camada física conectada.

IPSelect (MProt)

Leitura ou Escrita

Configuração por parâmetros numéricos

PARÂMETRO	VALOR
N1	-2 (menos dois)
N2	0 (zero)
N3	4 (quatro)
N4	0 (zero)

Configuração por parâmetros sintáticos

PARÂMETRO	VALOR
Item	MProt.IO.Ethernet.IPSelect

Indica o endereço IP ativo. Os valores possíveis são **0**: O endereço IP principal está selecionado ou ativo ou **1**: O endereço IP alternativo ou de *backup* está selecionado ou ativo.

Se a interface Ethernet está conectada, este Tag indica qual dos dois 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 biblioteca **IOKit** tenta conectar com o outro endereço IP. Se a conexão com o endereço IP alternativo funcionar, este é configurado como o endereço IP ativo, ou seja, um *switchover* automático.

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

IPSwitch (MProt)

Somente Escrita

Configuração por parâmetros numéricos

PARÂMETRO	VALOR
N1	-2 (menos dois)
N2	0 (zero)
N3	4 (quatro)
N4	1 (um)

Configuração por parâmetros sintáticos

PARÂMETRO	VALOR
Item	MProt.IO.Ethernet.IPSwitch

Qualquer valor escrito neste Tag força um *switchover* manual. Se o endereço IP principal está ativo, então o endereço IP de *backup* é ativado, e vice-versa. Isto força a reconexão com o endereço IP especificado se este Driver está atualmente conectado. Se este Driver está desconectado, isto configura o endereço IP ativo para a próxima tentativa de conexão.

Watchdog

Esta funcionalidade é útil apenas para arquiteturas de redundância que contam com uma variável em um equipamento a ser eleita para indicar o momento de troca do endereço IP de destino. Na grande maioria dos casos, este tipo de Tag não é necessário. No caso de aplicar a um projeto, este Tag deve ser uma variável Booleana (**BIT**) que deve variar de valor antes que o tempo configurado expire, e logicamente deve ser lida com uma varredura de tempo razoavelmente inferior ao tempo de expiração, configurado na opção **Watchdog period (ms)** do grupo **ISOTCP / ISOTCP243** na aba **MProt** da janela de configuração. Sempre que esta variável não variar por este período, a troca de endereço IP é executada. Para configurar um Tag como **Watchdog**, adicione a **String** "/watchdog" ao final do parâmetro sintático **Item**, como por exemplo "DB2:DBX0.0/watchdog".

ReadOnce

Em arquiteturas de comunicação utilizando o protocolo **ISOTCP** com quantidades enormes de Tags de Comunicação, acima de mil Tags, pode ser interessante que uma parte dos Tags seja lida apenas uma vez ao entrar no estado **Em Advise** (consulte a **nota** a seguir), em oposição ao normal, em que são lidos repetidamente a cada ciclo de varredura. Para obter uma melhoria de desempenho, os Tags **ReadOnce** são retirados do bloco de Tags de leitura, diminuindo o volume de informações em futuras leituras.

Um Tag **ReadOnce** somente volta a ser lido quando sair do estado **Em Advise** e retornar, ou quando ocorrer alguma desconexão ou falha de comunicação.

Para configurar um Tag como **ReadOnce**, adicione a **String** "/readonce" ao final do parâmetro sintático **Item**, como por exemplo "DB2:DB0/readonce".

Esta funcionalidade é usada exclusivamente para configuração **ISOTCP** com a opção **Extra Connections** configurada com um valor igual ou superior a 1 (um).

NOTA

O estado **Em Advise** de um Tag é definido por este Tag manter-se em atualização constante, em intervalos de tempo definidos pela propriedade **Scan**, ou seja, períodos de varredura.

A forma de mudar o status de um Tag para **Em Advise** é configurando a propriedade **AllowRead** para Verdadeiro e por uma das seguintes condições:

- Um Tag está com a propriedade **AdviseType** igual a 0 (zero), isto é, igual a **AlwaysInAdvise**
- Um Tag está com a propriedade **AdviseType** igual a 1 (um), isto é, igual a **AdviseWhenLinked**, e está associado a algum objeto ativo em uma Tela aberta

Caso alguma destas condições não esteja presente, um Tag de Comunicação não tem o status **Em Advise**, ou seja, não é lido.

Leitura de um Valor com Estampa de Tempo

Este recurso lê um valor junto com uma estampa de tempo armazenada logo após este valor em um dispositivo, em uma única requisição, aplicando a estampa de tempo lida de um dispositivo nativamente em um Tag, ao invés de devolver

esta estampa de tempo como um segundo valor. Isto evita a necessidade de um script para combinar um valor e a respectiva estampa de tempo, e funciona somente sobre os protocolos **ISOTCP** ou **ISOTCP243**.

Preparando um Bloco de Dados

Dentro de uma variável **DB**, o valor e a estampa de tempo devem ser declarados juntos, com a estampa de tempo imediatamente após o valor, sem nenhuma folga entre os valores, já que este Driver lê os dois valores como um único bloco contíguo de bytes, conforme o exemplo a seguir.

```
Valor : Real // ex.: offset 100
Timestamp : DTL // offset 104, logo apos o Valor
```

O campo **Timestamp** deve ser uma variável **DTL** (*Date_And_Time_Long*) da Siemens, uma estrutura padrão de 12 bytes (**YEAR**, **MONTH**, **DAY**, **WEEKDAY**, **HOUR**, **MINUTE**, **SECOND** e **NANOSECOND**). Não é necessária nenhuma programação adicional em um dispositivo para montar esta estrutura. Uma simples atribuição a uma variável **DTL**, como por exemplo uma instrução **RD_LOC_T**, já produz o *layout* esperado.

NOTAS

- Uma variável **DB** precisa estar com a opção **Optimized block access** desligada no projeto do TIA Portal, caso contrário este Driver não consegue ler esta variável.
- O campo **WEEKDAY** da estrutura **DTL** não é utilizado por este recurso.

Tags para Leitura de um Valor com Estampa de Tempo

Um Tag para este recurso pode ser configurada tanto por **Parâmetros Numéricos [N/B]** quanto por sintaxe, conforme descrito a seguir. Em ambos os casos, este Tag deve apontar para o endereço do **valor**, não da estampa de tempo. A variável **DTL** é lida automaticamente logo em seguida.

Configuração por Parâmetros Numéricos [N/B]

Use os parâmetros *N1* e *N3* conforme descrito no tópico **Configuração por Parâmetros Numéricos [N/B]**. O parâmetro *N4* deve apontar para o *offset* do valor e, no parâmetro *N2*, use a fórmula *TipoDeDados* × *100* + *92* (área de dados **92**), usando os mesmos códigos para tipo de dados descritos naquele tópico. Por exemplo, para um valor **REAL** (tipo de dado **7**), *N2* é igual a **792**.

Configuração por Parâmetros Sintáticos

Use a sintaxe a seguir, em que valores entre colchetes são opcionais.

```
<[Dispositivo:]>DTL<NumeroDaDB>.<Tipo><Endereço>
```

Os valores **Dispositivo**, **NumeroDaDB** e **Endereço** referem-se aos mesmos itens descritos no tópico **Configuração por Parâmetros Sintáticos**, e **Tipo** é o tipo de dados de um valor, como por exemplo **DBF** para um valor **REAL**, da mesma tabela daquele tópico, conforme o exemplo a seguir.

```
(Dispositivo 0, valor REAL a partir do endereço 100 da DB1,
com a estampa de tempo DTL logo após)
Dispositivo: Vazio - Item: 0:DTL1.DBF100
```

Coleta SOE

Esta seção contém informações específicas para Coleta SOE de eventos.

Preparando a Coleta SOE

Antes de utilizar os Tags de Coleta SOE, é preciso preparar um equipamento com a construção de uma Tabela **DB** (área **V**) e o desenvolvimento de uma lógica programável compatível com os procedimentos de coleta SOE desenvolvida para este Driver.

Tabela de Eventos SOE

Esta tabela tem como objetivo dimensionar o tamanho do *buffer* de eventos e gerenciar a entrada e a saída destes em uma rotina de *buffer* circular. Esta tabela é constantemente atualizada, tanto por um equipamento quanto por este Driver.

A Tabela de Eventos SOE deve conter registros de controle e de armazenamento de eventos, com base na estrutura de dados descrita na tabela a seguir.

Estrutura de dados

ENDEREÇO	DESCRIÇÃO	TIPO DE DADOS
0.0		**STRUCT**
+0.0	Status da tabela	**WORD** (16 bits sem sinal)
+2.0	Ponteiro de gravação	**WORD** (16 bits sem sinal)
+4.0	Status da aquisição	**WORD** (16 bits sem sinal)
+6.0	Limite máximo de itens do *buffer* circular	**WORD** (16 bits sem sinal)
+8.0	*Buffer* circular	**ARRAY[1..n]** (limite de itens definido pelo usuário)
+0.0		**STRUCT**
+0.0	**TIMESTAMP_LOLO** (ano)	**WORD** (16 bits sem sinal)
+2.0	**TIMESTAMP_LOHI** (dia e mês)	**WORD** (16 bits sem sinal)
+4.0	**TIMESTAMP_HILO** (hora e minuto)	**WORD** (16 bits sem sinal)
+6.0	**TIMESTAMP_HIHI** (segundo e milissegundo)	**WORD** (16 bits sem sinal)
+8.0	Valor do tipo de evento 1 (um)	Tipo de dados de um evento, definido pelo usuário
+n.0	Valor do tipo de evento 2 (dois)	Repete o mesmo tipo de dados
+n.0	Valor do tipo de evento 3 (três)	Repete o mesmo tipo de dados
+n.0	Valor do tipo de evento n	Repete o mesmo tipo de dados
=n.0		**END_STRUCT**
=n.0		**END_STRUCT**

Descrição dos Registros de Controle dos Eventos

- **Status da Tabela**: Deve ser mantido exclusivamente por um equipamento, indicando o número de eventos disponíveis para a leitura no *buffer* circular. Deve ser atualizado por um equipamento sempre que novos eventos são adicionados ao *buffer* circular, ou após a conclusão da coleta de eventos por uma aplicação, o que pode ser detectado pela mudança no **Status da Aquisição**

- **Ponteiro de Gravação**: Deve ser mantido exclusivamente por um equipamento, indicando o índice, começando em 0 (zero), da posição onde deve ser inserido o próximo evento. O índice deve ser incrementado por um equipamento a cada nova inserção de eventos no *buffer* circular, voltando ao índice 0 (zero) após alcançar o limite máximo do *buffer* circular

- **Status da Aquisição**: Deve ser mantido por um equipamento em conjunto com este Driver, indicando o número de registros já lidos a cada transação. Após cada coleta, este Driver escreve neste registro o número de eventos que conseguiu ler. Ao detectar esta modificação, um equipamento deve imediatamente descontar este valor escrito por este Driver do **Status da Tabela**, e então zerar o **Status da Aquisição**

- **Limite Máximo de Itens do Buffer Circular**: Valor constante que especifica o limite máximo de eventos a armazenar no *buffer* circular antes do ponteiro voltar para o índice 0 (zero). Deve conter exatamente o valor limite do *Array* dimensionado para os eventos do *buffer* circular

Descrição dos Registros de Armazenagem dos Eventos

- **TIMESTAMP**: Horário da ocorrência de um evento

- **Valor do Evento**: Valor do evento ocorrido, que pode ser composto por 1 (um) ou *n* valores, todos com o mesmo tipo de dados, e que são agrupados para o mesmo **TIMESTAMP** gerado na ocorrência de um evento

Formato de um TIMESTAMP

Um **TIMESTAMP** é representado por quatro **Words**, conforme a estrutura de dados descrita na tabela a seguir.

Estrutura de dados

WORD	CONTEÚDO	INTERVALO
0	Ano	Entre 0 (zero) e 65535
1	Dia e mês	ddddddddmmmmmmmm
2	Hora e minuto	hhhhhhhhmmmmmmmm
3	Segundo e milissegundo	ssssssmmmmmmmmmm

- O primeiro **Word** contém um valor inteiro referente ao ano

- O segundo **Word** está dividido em parte alta para representar o dia e em parte baixa para representar o mês

- O terceiro **Word** está dividido em parte alta para representar as horas e em parte baixa para representar os minutos

- O quarto **Word** usa os seis bits mais altos para representar os segundos e os 10 bits mais baixos para representar os milissegundos

Procedimento de Aquisição

Um equipamento deve iniciar a inserção dos eventos no sentido crescente, a partir do endereço base da tabela, referente ao início do *buffer* circular. A cada novo evento inserido, o ponteiro de gravação deve ser incrementado, passando a apontar para o próximo endereço disponível do *buffer*.

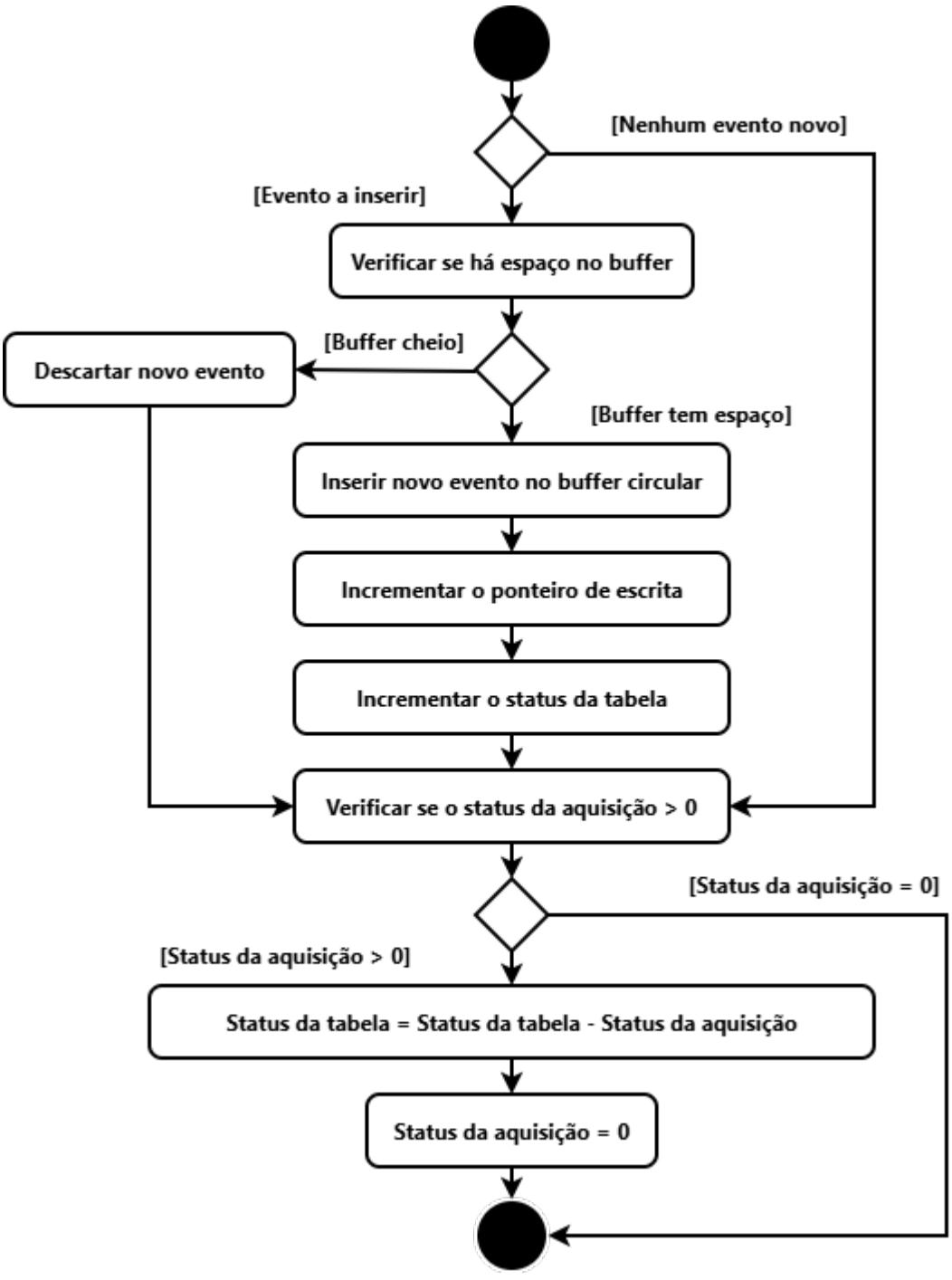
Este Driver executa a leitura do evento mais antigo para o mais recente. O endereço do início da leitura é calculado por este Driver através do valor do **Ponteiro de Gravação** e do **Status da Tabela**.

Se o número de eventos disponíveis é maior que o máximo permitido em um único *frame* de comunicação do protocolo, este Driver executa múltiplas leituras em bloco, atualizando o valor do **Status da Aquisição** no final do processo com o número total de eventos lidos.

Ao detectar que este Driver escreveu um valor maior que 0 (zero) no **Status da Aquisição**, um equipamento deve imediatamente subtrair o valor do **Status da Aquisição** do valor do **Status da Tabela** e zerar o **Status da Aquisição**.

Um equipamento pode inserir novos eventos na tabela durante o processo de aquisição por este equipamento, desde que não ocorra *overflow* do *buffer* circular, incrementando o **Status da Tabela**.

A figura a seguir apresenta um pequeno fluxograma, em formato de Diagrama de Atividades UML, com uma sugestão de implementação para a lógica de um equipamento.



Fluxograma SOE

Tags de Coleta SOE

A coleta SOE de eventos é executada com a utilização dos Tags descritos a seguir, por meio de uma comunicação ISOTCP com um equipamento.

Tag Bloco de Registro de Controle (Somente Leitura)

- **B1**: 0 (zero)
- **B2**: 391 (Tipo de dados = 3 × 100 + Área de dados = 91)
- **B3**: Número do bloco **DB**. Caso a memória contenha um bloco **DB** único ou não especificado, preencha com o valor 1 (um)
- **B4**: Não utilizado, deixe em 0 (zero)

Um Tag Bloco de Registro de Controle deve conter 4 (quatro) Elementos para retornar os seguintes valores:

- **Elemento 1**: Status da tabela
- **Elemento 2**: Ponteiro de gravação
- **Elemento 3**: Status da aquisição
- **Elemento 4**: Limite máximo de itens do *buffer* circular

Para uma descrição de cada um destes Registros de Controle, consulte o tópico **Preparando a Coleta SOE**.

Tag Bloco de Coleta de Dados (Somente Leitura)

- **B1**: 0 (zero)
- **B2**: Tipo de dados × 100 + Área de dados = 90
- **B3**: Número do bloco **DB**. Caso a memória contenha um bloco **DB** único ou não especificado, preencha com o valor 1 (um)
- **B4**: Não utilizado, deixe em 0 (zero)

Um Tag Bloco de Coleta de Dados deve conter um número de Elementos correspondente ao número de valores de tipo de evento *n* que compõem um único evento. Se um evento é composto por um único valor, dimensione este Tag com apenas 1 (um) Elemento. Caso um evento seja composto por 2 (dois) valores, este Tag deve ser dimensionado para 2 (dois) Elementos, e assim por diante. Utilize o parâmetro *B2* deste Tag para indicar o tipo de dados associado aos valores de um evento.

NOTAS

- Todos os valores que compõem um evento devem ser do mesmo tipo de dados, assim como cada tabela **DB** de um equipamento deve ser preenchida por um mesmo tipo de evento.
- As Áreas de Dados **90** e **91** existem para serem utilizadas somente no nível lógico de uma aplicação. Em nível de protocolo, ambos os Tags trabalham na Área de Dados **9** (**DA_V**) de um equipamento automaticamente.

Documentação das Interfaces de Comunicação

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

Configurações de um Driver

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

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

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

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

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

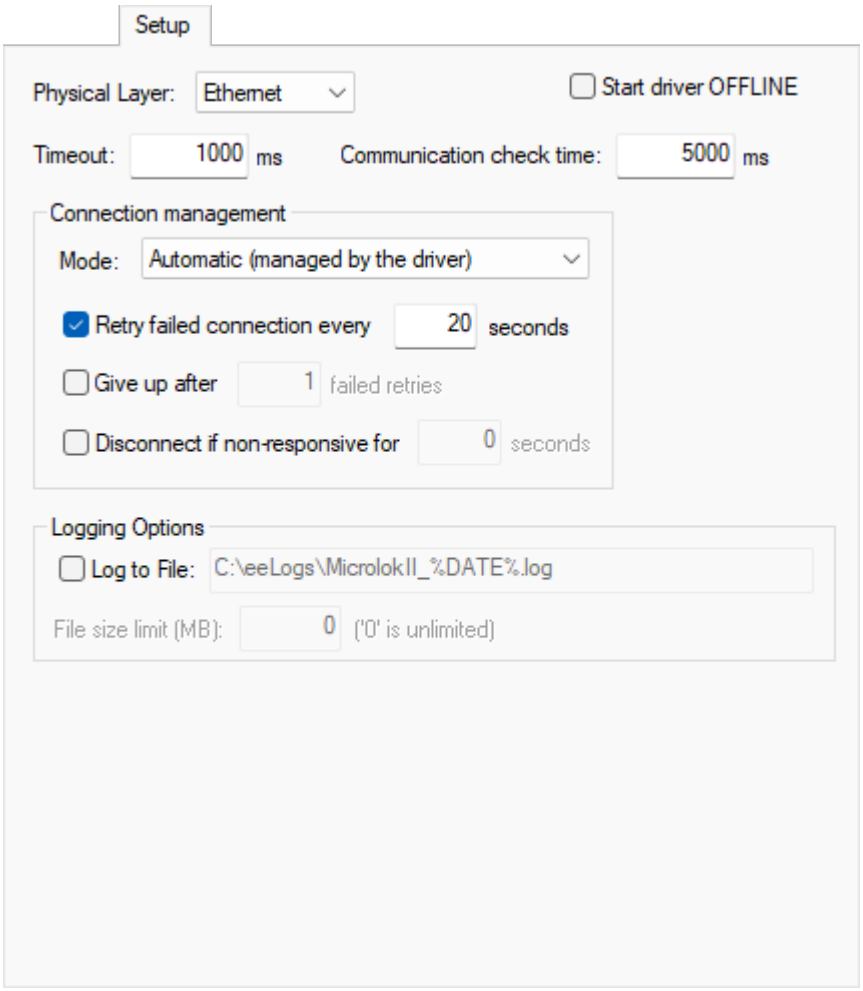
Caixa de Diálogo de Configuração

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

Aba Setup

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

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



Aba Setup

Opções gerais da aba Setup

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

Opções para o grupo Connection management

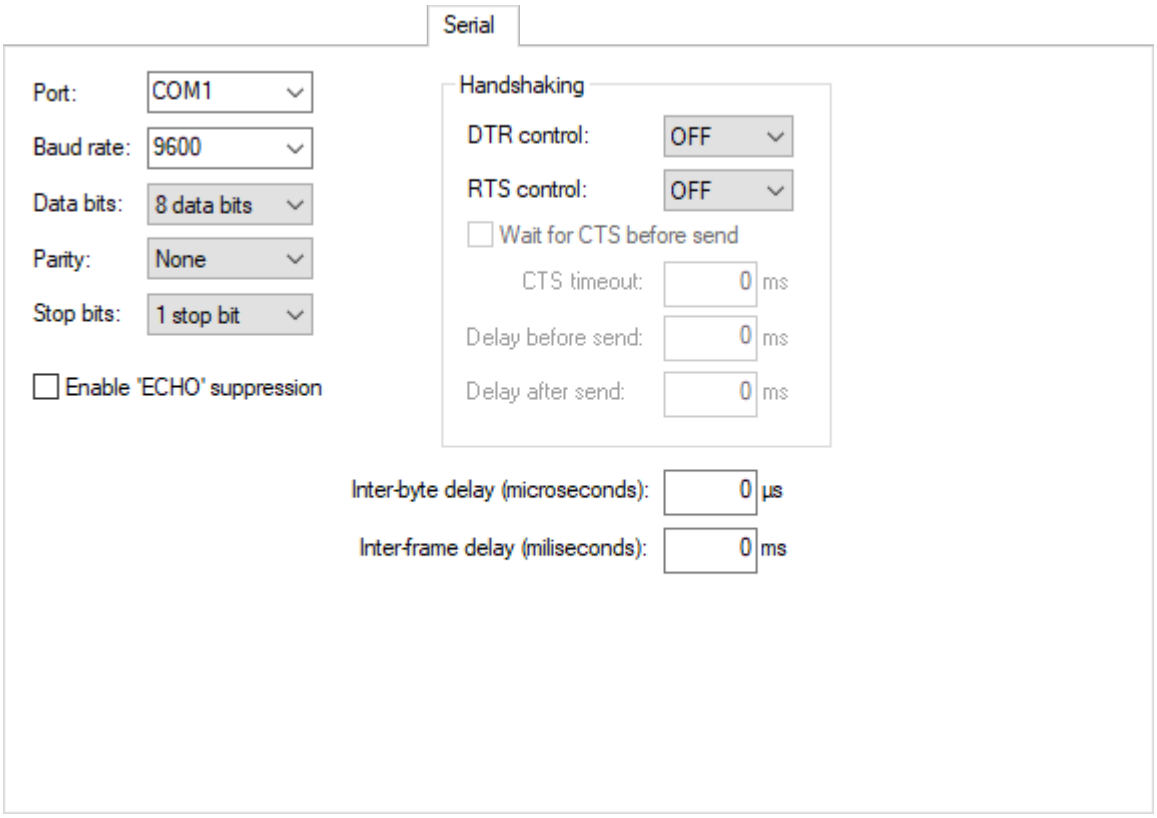
OPÇÃO	DESCRIÇÃO
Mode	Seleciona o modo de gerenciamento de conexão. Selecionar a opção **Automatic** permite que um Driver gerencie a conexão automaticamente, como especificado nas opções seguintes. Selecionar a opção **Manual** permite que uma aplicação gerencie a conexão completamente
Retry failed connection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

Opções para o grupo Logging Options

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

Aba Serial

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



Aba Serial

Opções gerais da aba Serial

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

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

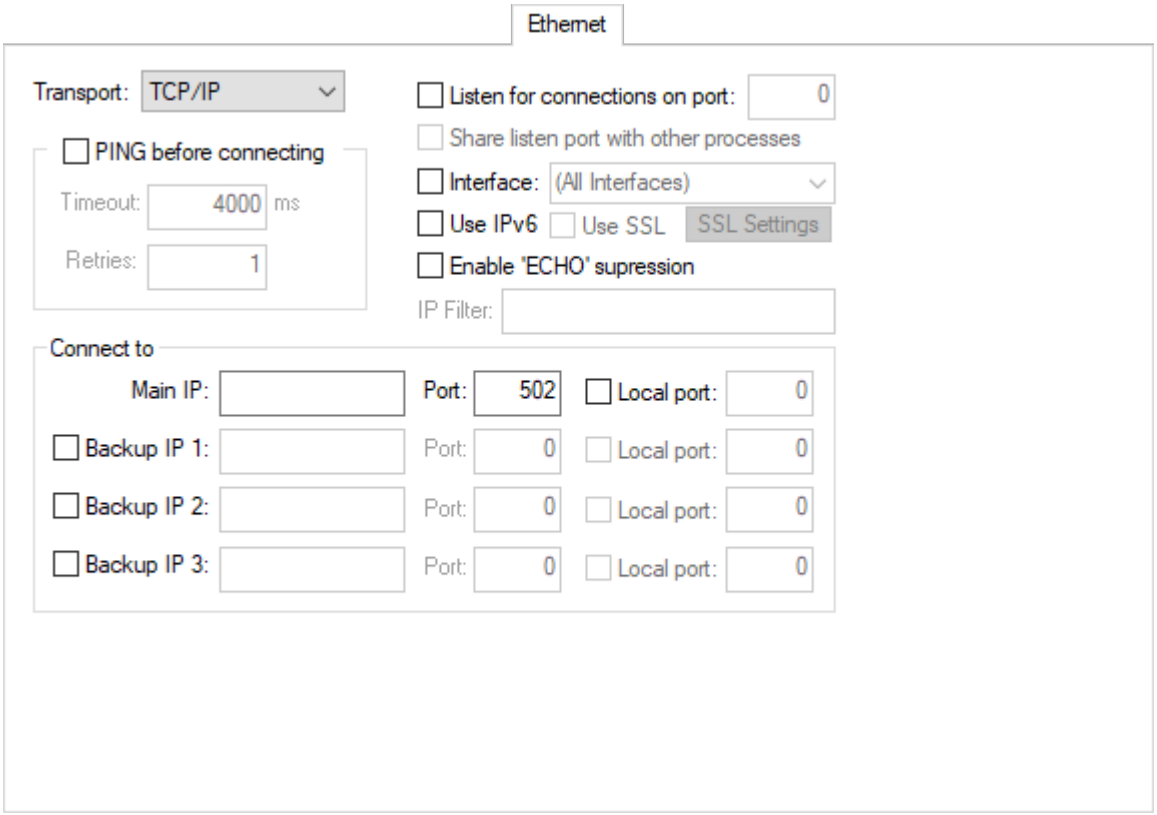
O grupo **Handshaking** configura o uso dos sinais **RTS**, **CTS** e **DTR** no processo de *handshaking* ou seja, controla quando um dado pode ser enviado ou recebido através de uma linha serial. Na maioria das vezes, configurar a opção **DTR control** para **ON** e a opção **RTS control** para **Toggle**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

Opções disponíveis no grupo Handshaking

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

Aba Ethernet

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



Aba Ethernet

Opções disponíveis na aba Ethernet

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

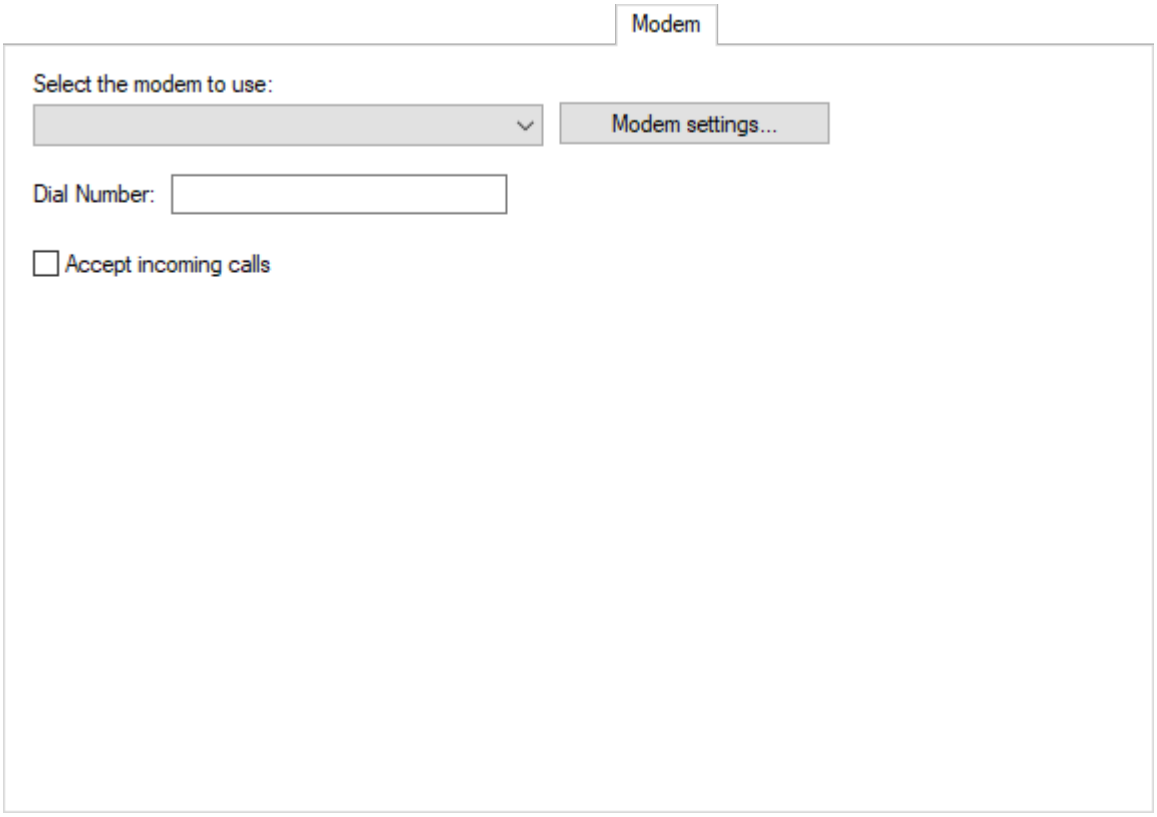
OPÇÃO	DESCRIÇÃO
IP Filter	Lista de endereços IP restringidos ou permitidos de onde um Driver aceita conexões (*Firewall*). Consulte a propriedade **IO.Ethernet.IPFilter** para mais informações
PING before connecting	Habilite esta opção para executar um comando **ping**, ou seja, para verificar se um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

Opções disponíveis no grupo Connect to

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

Aba Modem

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



Aba Modem

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

Opções disponíveis na aba Modem

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

Aba RAS

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

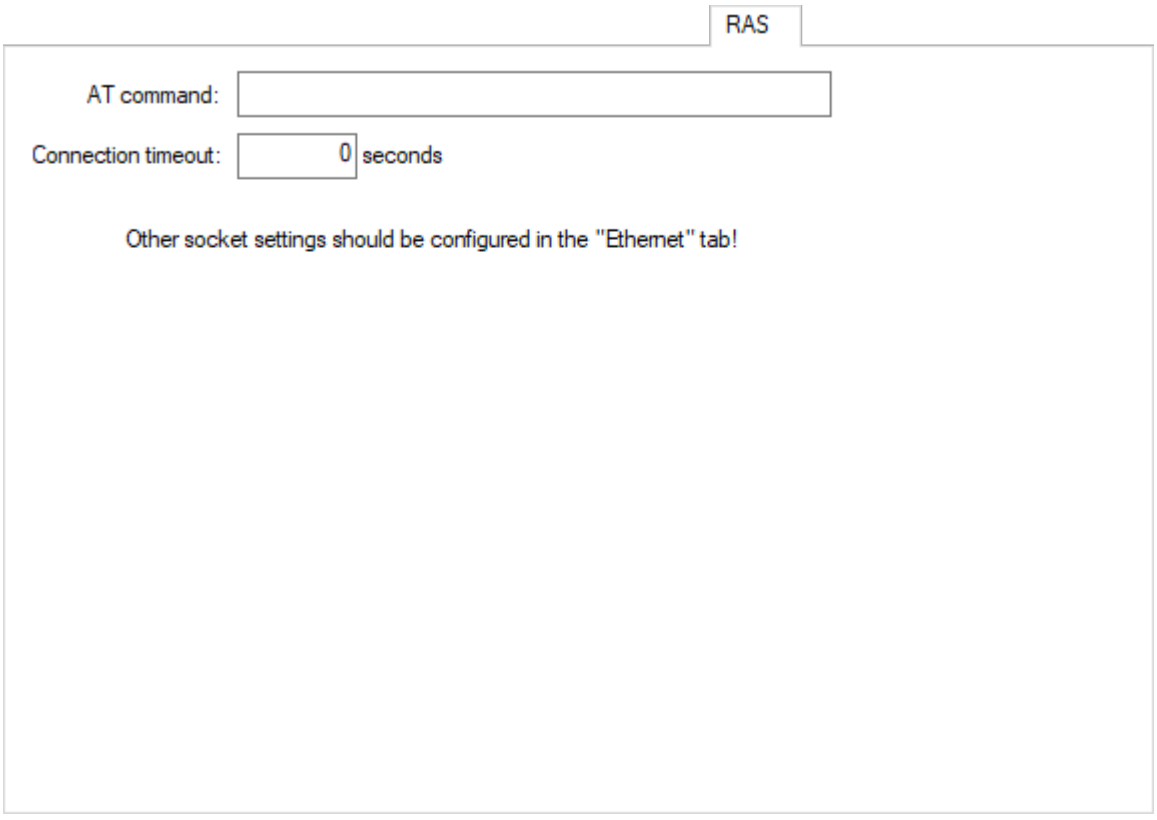
A Interface **RAS** abre uma conexão *socket* com um dispositivo RAS. Um dispositivo RAS é um servidor de modems acessível através de TCP/IP, aguardando por conexões *socket* em uma porta IP. Para cada conexão aceita nesta porta tem-se acesso a um modem.

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

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

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

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



Aba RAS

Opções disponíveis na aba RAS

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

Configurações Gerais

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

Tags de Comunicação

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

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

IO.CommunicationStatus

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

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

IO.IOKitEvent

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

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

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

NOTA

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

IO.PhysicalLayerStatus

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

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

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

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

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

IO.SetConfigurationParameters

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

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

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

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

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

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

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

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

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

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

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

IO.WorkOnline

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

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

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

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

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

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

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

IMPORTANTE

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

Propriedades

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

IO.ConnectionMode

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

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

IO.RecoverEnable

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

IO.RecoverPeriodSec

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

NOTA

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

IO.StartOffline

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

NOTA

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

IO.TimeoutMs

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

IO.Type

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

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

Configuração de Estatísticas

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

Tags de Comunicação

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

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

IO.Stats.Partial.BytesRecv

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

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

IO.Stats.Partial.BytesSent

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

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

IO.Stats.Partial.TimeConnectedSeconds

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

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

IO.Stats.Partial.TimeDisconnectedSeconds

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

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

IO.Stats.Total.BytesRecv

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

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

IO.Stats.Total.BytesSent

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

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

IO.Stats.Total.ConnectionCount

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

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

IO.Stats.Total.TimeConnectedSeconds

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

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

IO.Stats.Total.TimeDisconnectedSeconds

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

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

Propriedades

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

Configuração da Interface Ethernet

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

Tags de Comunicação

Tags da Interface Ethernet (N2/B2 = 4)

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

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

IO.Ethernet.IPSelect

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

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

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

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

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

IO.Ethernet.IPSwitch

Tipo de Tag	Tag de Comunicação

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

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

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

IO.Ethernet.SocketState

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

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

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

Propriedades

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

NOTA

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

IO.Ethernet.AcceptConnection

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

IO.Ethernet.BackupEnable[2,3]

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

desabilitar a utilização.

IO.Ethernet.BackupIP[2,3]

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

IO.Ethernet.BackupLocalPort[2,3]

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

IO.Ethernet.BackupLocalPortEnable[2,3]

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

IO.Ethernet.BackupPort[2,3]

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

IO.Ethernet.IPFilter

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

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

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

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

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

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

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

IO.Ethernet.ListenIP

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

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

IO.Ethernet.MainIP

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

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

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

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

IO.Ethernet.PingTries

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

IO.Ethernet.ShareListenPort

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

IO.Ethernet.SupressEcho

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

IO.Ethernet.Transport

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

IO.Ethernet.UseIPv6

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

Configuração da Interface Modem

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

Tags de Comunicação

Tags da Interface Modem (N2/B2 = 3)

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

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

IO.TAPI.ConnectionBaudRate

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

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

IO.TAPI.Dial

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

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

IO.TAPI.HangUp

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

Qualquer valor escrito neste Tag desliga a chamada atual.

NOTA

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

IO.TAPI.IsModemConnected

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

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

IO.TAPI.IsModemConnecting

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

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

IO.TAPI.ModemStatus

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

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

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

IO.TAPI.PhoneNumber

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

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

Propriedades

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

IO.TAPI.AcceptIncoming

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

IO.TAPI.ModemID

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

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

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

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 COM*n*.

IO.Serial.RTS

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

IO.Serial.StopBits

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

IO.Serial.SupressEcho

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

IO.Serial.WaitCTS

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

Histórico de Revisões do Driver

VERSÃO	DATA	AUTOR	COMENTÁRIOS
4.0.40	03/11/2026	M. Ludwig	• Corrigidas as quedas de comunicação no protocolo **ISOTCP** com múltiplas conexões, causadas por leituras agrupadas maiores que o tamanho de PDU negociado com um equipamento (regressão da versão **4.0.38**). Este Driver também volta a dividir a leitura quando um equipamento responde com erro de classe **81h** (*Case 41145*). • Adicionada a leitura de valores com estampa de tempo, em que um valor é lido juntamente com a estampa de tempo de um equipamento Siemens DTL adjacente em um único pedido, aplicado nativamente a um Tag (*Case 40697*).
4.0.39	26/08/2026	M. Ludwig	• Aplicada uma tolerância a múltiplos espaços entre tipos de dados **String**, **WString** ou **Array** e o caractere de colchete na importação de Tags do TIA Portal (*Case 40597*).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
4.0.38	27/04/2026	M. Ludwig	• Driver atualizado para a biblioteca **IOKit** versão **3.0** e Visual Studio 2022 (*Case 38017*). • Aprimoradas as informações em caixas de mensagens para erros de importação de Tags (*Case 38570*). • Adicionada a interpretação de classes de erro para o código 86h (*Case 39707*).
4.0.37	21/05/2025	M. Ludwig	• Adicionado suporte à importação de Tags provenientes do TIA Portal versão **20** (*Case 37856*).
4.0.35	14/04/2025	M. Ludwig	• Ajuste de documentação sobre leitura ou escrita de bits (*Case 35721*).
4.0.34	04/02/2025	M. Ludwig	• O campo **Dispositivo** agora é automaticamente adicionado nos Tags do **Tag Browser** (*Case 37297*).
4.0.33	11/12/2024	M. Ludwig	• Driver atualizado para a versão mais recente da biblioteca **IOKit** (*Case 37033*).
4.0.32	15/08/2024	A. Fetzner	• Adicionado suporte à importação de Tags provenientes do TIA Portal versão **19** (*Case 35940*).
4.0.31	5/10/2023	M. Ludwig A. Fetzner	• Corrigido um erro de cálculo na posição dos bits para tipos de dados com bytes reversos e bits invertidos (*Case 34863*).
		A. Fetzner	• Adicionado suporte a UDTs (*User-Defined Types*) e Arrays com limite inicial diferente de 0 (zero) e modificada a visualização de Tags PLC importados, dividindo-os em pastas para cada tabela (*Case 34783*).
4.0.30	29/08/2023	A. Fetzner	• Adicionado suporte à importação de Tags provenientes do TIA Portal versões **17** e **18** (*Case 31160*).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
4.0.29	25/08/2023	M. Ludwig	• Corrigida uma falha de comunicação quando este Driver está configurado com o protocolo **MPI** (*Case 32311*).
4.0.28	17/11/2021	M. Ludwig	• Adicionada uma opção de *inter-frame delay* para a camada Ethernet (*Case 31757*).
4.0.27	02/08/2021	M. Ludwig	• Implementado o tipo de dados **WString** (*Case 30670*).
4.0.26	05/05/2021	M. Ludwig	• Implementados os tipos de dados **Int**, **Word**, **DInt** e **DWord** com byte menos significativo primeiro (*Case 30617*).
4.0.24	22/10/2020	M. Ludwig	• Implementada uma lista recursiva de **DBs** e **TagTables** dentro de pastas (*Case 29822*).
4.0.23	14/10/2020	M. Ludwig	• Implementada a importação de Tags com o tipo de dados **Array de estruturas complexas** (*Case 29560*).
4.0.22	24/09/2020	M. Ludwig	• Corrigida a duplicidade de Tags importados ao abrir a janela de propriedades e selecionar a aba **Import Tags** pela primeira vez (*Case 29560*).
4.0.21	18/09/2020	M. Ludwig	• Criado o MProt TIA Portal Importer (*Case 29179*). • Implementada a importação de Tags (*Case 29185*).
4.0.20	26/12/2019	M. Ludwig	• Implementado o Tag **ReadOnce** (*Case 27374*). • Implementados os tipos de dados **LSB First** com bits reversos (*Case 27576*).
4.0.19	05/08/2019	M. Ludwig	• Corrigida a retenção de último valor em *cache* (*Case 26110*). • Driver portado para o Visual Studio 2017 (*Case 27092*).
4.0.18	28/09/2018	M. Ludwig	• Corrigido um congelamento em leituras usando conexões simultâneas após

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			um tempo indeterminado (*Case 25235*).
4.0.17	08/02/2018	M. Ludwig	• Implementados os tipos de dados **Int** e **DInt** em parâmetros sintáticos (*Case 23837*).
4.0.16	28/11/2017	M. Ludwig	• Implementadas as configurações de *rack*, *slot* e TSAP de destino para os endereços de *backup* 2 e 3 (*Case 23428*).
4.0.13	11/09/2017	M. Ludwig	• Corrigido um congelamento em leituras usando conexões simultâneas após um tempo indeterminado (*Case 23203*). • Implementado um mecanismo de *watchdog* para disparar a troca de endereços IP (*Case 23270*).
4.0.12	06/07/2017	M. Ludwig	• Corrigido um problema com a leitura de **Strings** em protocolos **ISOTCP** e **ISOTCP243** com conexões extras (*Case 22950*).
4.0.11	20/06/2017	M. Ludwig	• Desabilitada a otimização de leitura agrupada quando configurados os protocolos **ISOTCP** e **ISOTCP243** com otimização de Pedidos Simultâneos (*Case 22897*).
4.0.10	12/06/2017	C. Mello	• Ajustes para isolar a Coleta de SOE dos serviços de Superblocos e conexões simultâneas orientadas por *callbacks* (*Case 22785*).
4.0.9	29/05/2017	F. Englert	• Durante uma reconexão em uma CPU de *backup*, agora há uma verificação adicional se o endereço TSAP de destino corresponde ao novo endereço IP. Caso não seja o endereço IP esperado, este Driver não envia o pedido de conexão, e refaz o pedido imediatamente considerando o novo endereço IP e o respectivo TSAP (*Case 22020*).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
4.0.7	15/05/2017	M. Ludwig	• Adicionadas configurações de TSAP de destino em formato numérico hexadecimal (*Case 22432*).
4.0.4	10/04/2017	M. Ludwig	• Implementada a leitura e escrita de *database gateway* (*Case 22249*). • Corrigido um erro de leitura de **BOOL** e **BYTE** unitário (*Case 22451*).
4.0.1	16/03/2017	M. Salvador M. Ludwig	• Melhorias de performance no protocolo **ISOTCP** (*Case 22246*).
3.1.2	12/05/2015	M. Ludwig	• Corrigida a negação de funcionamento da opção de seleção do protocolo **ISOTCP243** (*Case 18675*).
3.1.1	19/09/2014	M. Ludwig	• Implementada a redundância de CPU, ou seleção automática de CPU de *backup*, *Rack* ou *Slot* alternativos, com a conexão ao endereço IP de *backup* (*Case 15782*). • Implementada a configuração de *Rack*, *Slot* e tipo de conexão na janela de propriedades deste Driver (*Case 15911*). • Adicionados os Tags de interface específicos para a opção de conexões extras (*Case 17221*).
3.0.1	20/12/2013	M. Salvador M. Ludwig	• Implementados os Superblocos internos em conexões TCP extras (*Case 14025*). • Driver portado para a biblioteca **IOKit** versão **2.0** (*Case 14019*).
2.13.1	21/08/2012	M. Ludwig	• Implementada a funcionalidade do campo **PDU REF** no protocolo **ISOTCP** (*Case 13299*).
2.12.1	30/05/2012	C. Mello	• Adicionado suporte para Coleta SOE de eventos em tabelas **DB** (*Case 12483*).
2.11.1	04/08/2011	M. Ludwig	• Incluída a consistência prevista no protocolo **MPI** e

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			melhorias de codificação (*Case 12392*). • Adicionadas informações sobre suporte a equipamentos Siemens do modelo S7-1200 (*Case 12292*).
2.10.1	25/03/2011	M. Ludwig	• Implementado o formato **S7 String** e a nova janela de propriedades para configuração de **Strings** (*Case 12005*).
2.9.1	25/08/2009	M. Ludwig	• Corrigida uma falha ao ler variáveis de tipo **Contador** (*Case 10701*). • Implementadas as configurações avançadas para protocolos **ISOTCP** e **ISOTCP243** (*Case 10717*).
2.8.1	19/06/2009	M. Ludwig	• Corrigida uma falha na desconexão endereçando múltiplos **Escravos** no protocolo **MPI** (*Case 10595*).
2.7.1	03/06/2009	M. Ludwig	• Implementado o tipo de dados **S5Time** (*Case 10413*).
2.6.1	07/01/2009	M. Ludwig	• Corrigida uma falha na conexão sob protocolo **ISOTCP** (*Case 10138*).
2.5.1	04/11/2008	M. Ludwig	• Melhorias na apresentação da janela de propriedades (*Case 9994*). • Implementado o *delay* de operação no protocolo **PPI** (*Case 9968*).
2.4.1	01/04/2008	M. Ludwig	• Corrigido um problema que ocorria ao endereçar entradas e saídas analógicas combinadas com a propriedade **EnableReadGrouping** em Verdadeiro (*Case 8927*). • Melhoria e consistência para evitar problema de desconexões de um equipamento, relatado no *Case 8968* (recebimento de valores aleatórios em variáveis de alarme no protocolo **ISOTCP**).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			• Corrigida a falta de tratamento de erro em recebimento de caracteres NAK no protocolo **MPI**, que ocasionava travamento em recepção de dados (*Case 8981*). • Melhoria de consistência em recepções do protocolo **MPI** (*Case 8981*). • Retirado um byte desnecessário em um *frame*, que ocasionava problemas em escritas de byte e bit sob o protocolo **ISOTCP** e equipamentos S7-400 (*Case 9021*). • Corrigida uma falha de reconexão automática em desconexão física no protocolo **ISOTCP** (*Case 9030*). • Corrigida a implementação de recepção de um *frame* longo de ACK no protocolo **PPI** (*Case 9118*). • Implementada a condição de dados indisponíveis no protocolo **PPI**. Quando a condição é encontrada, retorna uma lista vazia e **OK** ao invés de falha (*Case 9232*). • Corrigida a atribuição errada de *Service Access Point* no protocolo **MPI**, que ocasionava falhas de comunicação com adaptadores Tecnatron (*Case 9238*).
2.3.1	13/09/2007	M. Ludwig	• Corrigido um problema de reconexão com adaptador serial quando há desligamento de equipamento (*Case 8069*). • Implementado o endereçamento a múltiplos **Escravos** no protocolo **MPI** (*Case 8625*). • Porta Ethernet livremente configurável (*Case 8683*).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			• Driver compilado com a versão **1.14** da biblioteca **IOKit**, de forma a corrigir erros de leitura e escrita antes da primeira conexão (*Case 7614*). • Documentação atualizada com informações sobre tamanho de **Strings**, protocolos e equipamentos compatíveis (*Case 8206*).
2.2.1	28/03/2007	M. Ludwig	• Corrigida a falta de criação de *blob* que ocasionava erros em tempo de execução (*Case 8015*). • Corrigido um problema de troca de endereços IP em tempo de execução (*Case 8026*). • Adicionado suporte ao **Windows CE** (*Case 7504*). • Adicionado suporte a conversores IBHLink (*Case 7994*). • Corrigido um problema de escrita de **Strings** (*Case 7967*).
2.1.1	10/07/2006	M. Ludwig	• Correção no *parsing* de variáveis **DB** (*Case 7172*).
2.0.1	13/04/2006	M. Salvador M. Ludwig	• Corrigida uma falha do protocolo **PPI** Error: Single DLE in data field (*Case 6644*). • Retirada a verificação de endereços. Não importa o tipo de dados, é permitida a entrada de qualquer valor no parâmetro *N4* (*Case 6644*). • Corrigido um erro de interface de configuração, onde se misturavam configurações do conversor IBHLink com o protocolo **ISOTCP**. A porta TCP/IP 1099 era forçada ao invés da porta TCP/IP 102 (*Case 6644*). • Adicionado suporte a Superblocos e endereçamento simbólico (*Case 6644*).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
1.1.1	03/11/2005	M. Ludwig	• Otimização, padronização e revisão do código fonte.
1.0.1	01/05/2005	M. Salvador	• Versão inicial deste Driver.

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

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

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

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

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

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

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

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

Gartner 2014 Cool Vendor

OPC FOUNDATION MEMBER

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

Microsoft Partner
Gold Independent Software Vendor (ISV)

Copyright © 2005-2026 Elipse Software Ltda. Todos os direitos reservados.