

Driver Unitronics Vision 280

| | |
|--------------------------------|--|
| Nome do Arquivo | Vision280.dll |
| Fabricante | Unitronics - Industrial Automation Systems |
| Equipamentos | M90/M91/Vision OPLC Series |
| Protocolo | Proprietário sobre Serial ou TCP/IP |
| Versão | 2.0.1 |
| Última Atualização | 08/01/2025 |
| Plataforma | Win32 |
| Dependências | ActiveX UnCmDrv1.dll da Unitronics, versão 4.0 ou superior |
| Leitura com Superblocos | Não |
| Nível | 0 |

Introdução

O Driver Vision280 implementa a comunicação Mestre-Escravo com equipamentos M90/M91/Vision OPLC Series compatíveis com o ActiveX UnCmDrv1.dll, por meio da interface de uma aplicação da **Elipse Software**.

O arquivo UnCmDrv1.dll precisa ter uma chave registrada no Windows, que é realizada durante a instalação do software do PLC.

Preparação do Equipamento

Todos os PLCs devem efetuar comunicação via Serial RS-485 como Escravos, definindo um *Slave ID* único para o endereçamento na rede Serial RS-485.

Se a comunicação com os PLCs é via Ethernet TCP/IP, deve-se ter atenção com os seguintes detalhes:

- PLCs do tipo **M90**, **M91** e **Vision 120** não têm suporte para comunicação em modo **Ethernet**
- PLCs Vision do tipo **230**, **260** e **280** requerem a instalação de uma porta Ethernet para comunicação via TCP/IP

Este Driver efetua a comunicação por meio do ActiveX UnCmDrv1.dll, fornecido pelo próprio fabricante dos PLCs. No entanto, este ActiveX possui diversas dependências, que devem estar pré-instaladas e registradas no sistema operacional. Os arquivos a seguir são necessários:

- ssa3d30.ocx
- ComDlg32.ocx
- MSComCtl.ocx
- AmiCommPro.ocx
- MSComm32.ocx
- MSWinSck.ocx
- UnCmDrv1.dll

Por padrão, a instalação de qualquer software da Unitronics deve colocar todas estas dependências em ordem. No entanto, caso algum dos arquivos não esteja registrado, deve ser utilizado o comando **regsvr32** em um Prompt de Comando com privilégios de **Administrador**, conforme o exemplo a seguir.

```
regsvr32 [nome do arquivo com extensão]
```

Configuração do Driver

Este Driver utiliza os parâmetros **[P]** para definir a configuração da comunicação em modo **Serial**. Para configurar outros modos de comunicação, abra a janela de propriedades deste Driver no **Elipse E3**, **Elipse Power** ou **Elipse Water** ou selecione a opção **Extra Settings** no **Elipse SCADA**.

Comunicação Serial

Utilize os parâmetros **[P]** para configurar a comunicação deste Driver em modo **Serial**, conforme a tabela a seguir.

| | |
|-----------|---|
| P1 | Define a porta utilizada para comunicação Serial RS232. Os valores possíveis para este parâmetro são 1 : COM1 (padrão), 2 : COM2, 3 : COM3, 4 : COM4 ou n : COMn |
| P2 | Define a taxa de transmissão, em bits por segundo. Os valores possíveis para este parâmetro são 1 : 110, 2 : 300, 3 : 600, 4 : 1200, 5 : 2400, 6 : 4800, 7 : 9600 (padrão), 8 : 19200, 9 : 38400 ou 10 : 57600 |
| P3 | Define o tamanho de caractere, a paridade e os bits de parada, conforme a tabela a seguir. Para configurar este parâmetro, é preciso somar todos os valores de tamanho de caractere, paridade e bits de parada. Por exemplo, para configurar um caractere de 8 (oito) bits, paridade ímpar e 1 (um) bit de parada, o valor deste parâmetro deve ser 811 (800 + 10 + 1) |
| P4 | Não utilizado. Consulte a nota a seguir para mais informações |

NOTA

A partir da versão **2.0** deste Driver, o parâmetro **P4** não é mais utilizado. Portanto, o valor de *time-out* deve ser definido na aba **Setup** da janela de propriedades deste Driver.

Valores possíveis para o parâmetro P3

| VALOR | DESCRIÇÃO |
|------------|----------------------------|
| 700 | Caractere de 7 (sete) bits |
| 800 | Caractere de 8 (oito) bits |
| 000 | Sem paridade |
| 010 | Paridade ímpar |
| 020 | Paridade par |
| 001 | 1 (um) bit de parada |
| 002 | 2 (dois) bits de parada |

Comunicação Ethernet

Para habilitar a comunicação no modo **Ethernet TCP/IP**, abra a caixa de diálogo de propriedades deste Driver no **Elipse E3**, **Elipse Power** ou **Elipse Water** ou selecione a opção **Extra Settings** no **Elipse SCADA**.

Habilite a opção **Use Ethernet** para selecionar a comunicação no modo **Ethernet TCP/IP** e informe o endereço IP e a porta TCP/IP utilizados para efetuar a conexão com um equipamento.

NOTA

A partir da versão **2.0** deste Driver, o parâmetro *P4* não é mais utilizado. Portanto, o valor de *time-out* deve ser definido na aba **Setup** da caixa de diálogo de propriedades deste Driver.

Referência de Tags

Os Tags são configurados através dos parâmetros **N/B**. Os parâmetros não mencionados são irrelevantes na configuração dos Tags. Recomenda-se que estes parâmetros sejam deixados em 0 (zero).

Tag Valor

Tag Valor (Leitura e Escrita)

| | |
|--------------|--------------------------|
| N1/B1 | Identificador da Unidade |
| N2/B2 | Tipo de dados |
| N3/B3 | Endereço IP inicial |
| N4/B4 | Não utilizado |

Use um Tag PLC ou Bloco para consultar uma variável. O tipo de dados desta variável deve ser referenciado no parâmetro *N2/B2*, conforme o tópico **Tipos de Dados**. Consulte este tópico para verificar quais os tipos de operações suportadas para cada tipo de dados, se somente leitura, somente escrita ou ambos. As tabelas a seguir mostram exemplos de configuração deste Tag.

- Esta configuração lê ou escreve um valor inteiro no endereço de memória 10 de um equipamento

| PARÂMETRO N | DESCRIÇÃO |
|-------------|----------------------------------|
| N1 | 1 (um, Identificador da Unidade) |
| N2 | 22 (MEM INTEGERS) |
| N3 | 10 (endereço inicial) |
| N4 | 0 (zero) |

- Esta configuração lê ou escreve 5 (cinco) valores inteiros nos endereços de memória de 10 a 14 de um equipamento

| B | DESCRIÇÃO |
|----|----------------------------------|
| B1 | 1 (um, Identificador da Unidade) |
| B2 | 22 (MEM INTEGERS) |
| B3 | 10 (endereço inicial) |
| B4 | 0 (zero) |

Tipos de Dados

Códigos disponíveis para tipos de dados

| CÓDIGO | TIPO DE DADOS | PROCESSO |
|--------|------------------|-------------------|
| 01 | INPUTS BITS | Somente leitura |
| 02 | OUTPUTS BITS | Somente escrita |
| 11 | SYS BITS | Leitura e escrita |
| 12 | SYS INTEGERS | Leitura e escrita |
| 13 | SYS LONGS | Leitura e escrita |
| 14 | SYS DOUBLE WORDS | Leitura e escrita |
| 21 | MEM BITS | Leitura e escrita |
| 22 | MEM INTEGERS | Leitura e escrita |
| 23 | MEM LONGS | Leitura e escrita |
| 24 | MEM DOUBLE WORDS | Leitura e escrita |
| 25 | MEM FLOATS | Leitura e escrita |
| 92 | M90 DB INTEGERS | Leitura e escrita |

Tags Especiais com Serviços Ethernet

Alguns Tags especiais dedicam-se a alguns serviços da interface **Ethernet**, que podem ser utilizados em tempo de execução por este Driver.

Tag IP e Port

Tag IP e Port (Leitura e Escrita)

| | |
|----|---------------|
| B1 | 999 |
| B2 | 1 (um) |
| B3 | Não utilizado |
| B4 | Não utilizado |

Altera a configuração do endereço IP e da porta TCP/IP para conexão com um equipamento. Estes novos valores passam a atuar apenas quando há uma desconexão e uma nova conexão. O Tag Bloco deve conter 5 (cinco) Elementos, descritos a seguir.

- **Elemento 1:** Endereço IP 0 (zero, menos significativo)
- **Elemento 2:** Endereço IP 1 (um)
- **Elemento 3:** Endereço IP 2 (dois)
- **Elemento 4:** Endereço IP 3 (três, mais significativo)
- **Elemento 5:** Porta TCP/IP

Exemplo de configuração de um Tag Bloco para o endereço IP 127.0.0.1 e porta TCP/IP 8080:

- **Elemento 1:** 1 (um)
- **Elemento 2:** 0 (zero)
- **Elemento 3:** 0 (zero)
- **Elemento 4:** 127
- **Elemento 5:** 8080

A escrita neste Tag Bloco não altera os valores definidos nos parâmetros da janela de propriedades deste Driver.

NOTA

A partir da versão **2.0** deste Driver, não é mais utilizada a configuração dos parâmetros *B1* e *B2* iguais a -1 (menos um), para evitar conflito de configuração com os Tags de I/O padrão do **IOKit**.

Tag Reconnect

Tag Reconnect (Somente Escrita)

| | |
|-----------|---------------|
| N1 | 999 |
| N2 | 2 (dois) |
| N3 | Não utilizado |
| N4 | Não utilizado |

A escrita neste Tag solicita uma reconexão.

NOTA

A partir da versão **2.0** deste Driver, não é mais utilizada a configuração do parâmetro *N1* igual a -1 (menos um) e do parâmetro *N2* igual a -2 (menos dois), para evitar conflito de configuração com os Tags de I/O padrão do **IOKit**.

Tag Disconnect

Tag Disconnect (Somente Escrita)

| | |
|-----------|---------------|
| N1 | 999 |
| N2 | 3 (três) |
| N3 | Não utilizado |
| N4 | Não utilizado |

A escrita neste Tag desfaz a conexão TCP/IP ativa no momento.

NOTA

A partir da versão **2.0** deste Driver, não é mais utilizada a configuração do parâmetro *N1* igual a -1 (menos um) e do parâmetro *N2* igual a -3 (menos três), para evitar conflito de configuração com os Tags de I/O padrão do **IOKit**.

Documentação das Interfaces de Comunicação

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 **Eclipse 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 **Eclipse E3** versão 2.0 ou posterior, clique em **Configurar o driver**  na barra de ferramentas de um Driver. No **Eclipse 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 <i>time-out</i> , em milissegundos, para a camada física. Esta é a medida de tempo que a interface de I/O aguarda para a recepção de um byte qualquer do <i>buffer</i> de recepção |
| Communication check time | Configure o tempo, em milissegundos, para definir o intervalo em que a comunicação é considerada em estado |

| OPÇÃO | DESCRIÇÃO |
|-----------------------------|--|
| | <p>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</p> |
| Start driver OFFLINE | <p>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</p> |

Opções para o grupo Connection management

| OPÇÃO | DESCRIÇÃO |
|---|--|
| Mode | <p>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</p> |
| Retry failed connection every ... seconds | <p>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</p> |
| Give up after ... failed retries | <p>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</p> |
| Disconnect if non-responsive for ... seconds | <p>Habilite esta opção para forçar um Driver a se desconectar se nenhum byte chegou à interface de I/O no <i>time-out</i> especificado, em segundos. Este <i>time-out</i> deve ser maior que o <i>time-out</i> configurado na opção Timeout</p> |

Opções para o grupo Logging Options

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

Configurações Gerais

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

Tags de Comunicação

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

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

IO.CommunicationStatus

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

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

IO.IOKitEvent

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

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

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

NOTA

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

IO.PhysicalLayerStatus

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

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

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

IO.SetConfigurationParameters

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

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

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

No **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

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

IO.RecoverEnable

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

IO.RecoverPeriodSec

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

NOTA

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

IO.StartOffline

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

NOTA

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

IO.TimeoutMs

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

IO.Type

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

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

Configuração de Estatísticas

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

Tags de Comunicação

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

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

IO.Stats.Partial.BytesRecv

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

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

IO.Stats.Partial.BytesSent

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

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

IO.Stats.Partial.TimeConnectedSeconds

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

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

IO.Stats.Partial.TimeDisconnectedSeconds

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

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

IO.Stats.Total.BytesRecv

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

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

IO.Stats.Total.BytesSent

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

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

IO.Stats.Total.ConnectionCount

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

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

IO.Stats.Total.TimeConnectedSeconds

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

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

IO.Stats.Total.TimeDisconnectedSeconds

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

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

Propriedades

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

Histórico de Revisões do Driver

| VERSÃO | DATA | AUTOR | COMENTÁRIOS |
|--------|------------|-----------|---|
| 2.0.1 | 08/01/2025 | C. Mello | <ul style="list-style-type: none">• Ajustes de migração para o novo IOKit v2.00 (<i>Case 36743</i>). |
| 1.2.1 | 13/07/2010 | M. Ludwig | <ul style="list-style-type: none">• Configuração do endereço IP e porta TCP/IP em tempo de execução (<i>Case 11338</i>).• Implementada a comunicação no modo Ethernet (<i>Case 10586</i>). |
| 1.1.1 | 21/09/2004 | C. Mello | <ul style="list-style-type: none">• Implementado o parâmetro Delay between address exchange na caixa de diálogo Extras deste Driver, para permitir um ajuste de atraso (<i>delay</i>) entre trocas de unidades remotas.• Substituição da opção Log Date & Time por Log Timestamp na aba Log da caixa de diálogo Extras. A opção Log Date & Time é utilizada por padrão por este Driver.• Revisão geral de todo o código fonte. |
| 1.0.1 | 11/09/2003 | C. Mello | <ul style="list-style-type: none">• 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, 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)