

Driver Allen Bradley CIP

Nome do Arquivo	ABCIP.dll
Fabricante	Rockwell
Equipamentos	Controladores Rockwell das famílias ControlLogix, FlexLogix e CompactLogix usando um cartão Ethernet, porta Ethernet ou porta Serial
Protocolo	Ethernet/IP - CIP (Control Information Protocol)
Versão	3.0.23
Última Atualização	20/08/2020
Plataforma	Win32
Dependências	IOKit v1.15
Leitura com Superblocos	Não
Nível	0

Introdução

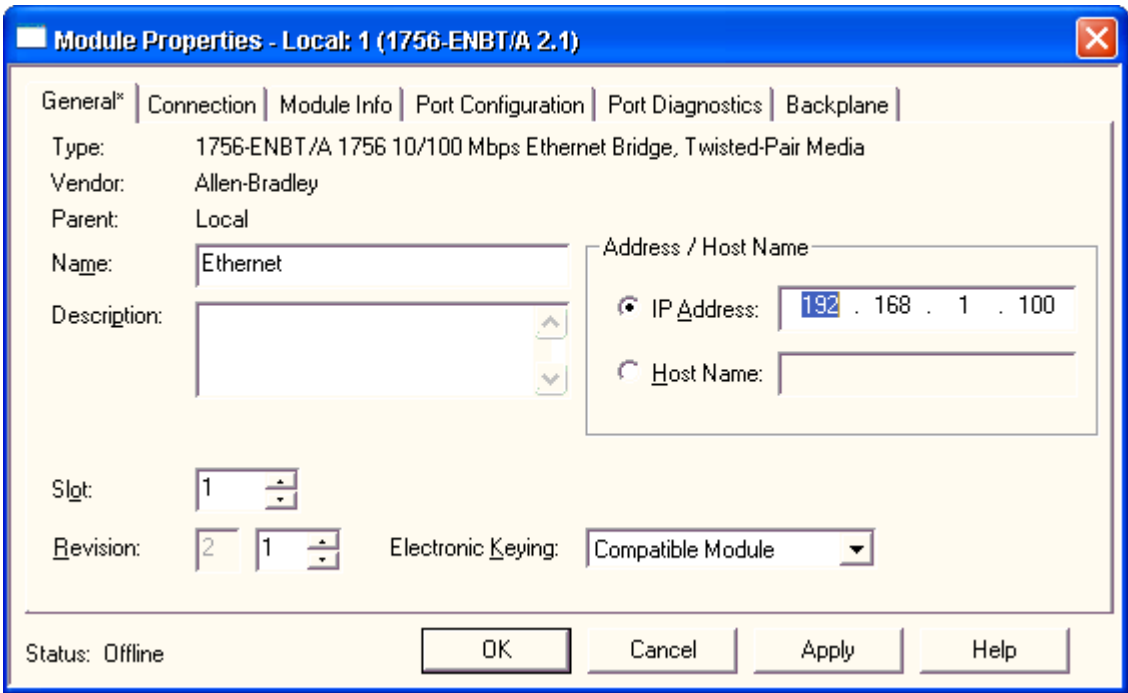
O Driver Allen Bradley CIP suporta comunicação com Controladores Rockwell das famílias ControlLogix, FlexLogix e CompactLogix usando um cartão Ethernet, porta Ethernet ou porta Serial.

Configuração do Driver

Esta seção fornece informações sobre a configuração do Driver Allen Bradley CIP.

Configuração do Equipamento

Para comunicação via Ethernet, é necessário informar o endereço IP do controlador e a porta 44818. O endereço IP é definido nas propriedades do controlador no aplicativo RSLogix 5000, conforme a figura a seguir.



Definição do endereço IP no aplicativo RSLogix 5000

Configuração dos Parâmetros [P]

P1	Não utilizado (manter o valor em zero)
P2	Não utilizado (manter o valor em zero)
P3	Não utilizado (manter o valor em zero)
P4	Não utilizado (manter o valor em zero)

Configurações Extras

Esta seção contém informações sobre as configurações extras do Driver Allen Bradley CIP executadas nas abas **CIP EthernetIP** e **CIP Addresses**.

Aba CIP Ethernet/IP

Aba CIP Ethernet/IP

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

Opções disponíveis na aba CIP Ethernet/IP

OPÇÃO	DESCRIÇÃO
Use Async Reads	A partir da versão 3.0 do E3 , operando com o IOKit v1.15 ou superior, este Driver pode operar no esquema assíncrono de leituras, que torna as leituras mais rápidas
Monitor Program Downloads	Quando se utiliza o esquema assíncrono da opção anterior, este Driver solicita as leituras a partir do endereço físico que as variáveis ocupam na memória do PLC. Se durante a comunicação ocorrer um <i>download</i> de programa para o PLC, os endereços físicos podem se modificar, fazendo com que os valores lidos fiquem incorretos. Selecione esta opção caso exista a possibilidade de ocorrer novos <i>downloads</i> enquanto o Driver estiver em operação, fazendo com que se obtenha novamente os endereços atualizados imediatamente após o término do <i>download</i> . NOTA: A utilização desta opção pode tornar a comunicação aproximadamente 2% mais lenta
Import each property as individual Tags	Ao utilizar o <i>Tag Browsing</i> do E3 , é possível selecionar se as estruturas de dados, <i>arrays</i> ou vetores são criados como Tags Bloco ou como Tags individuais. NOTA: Se for selecionada a criação de Tags individuais, recomenda-se o uso de leituras assíncronas

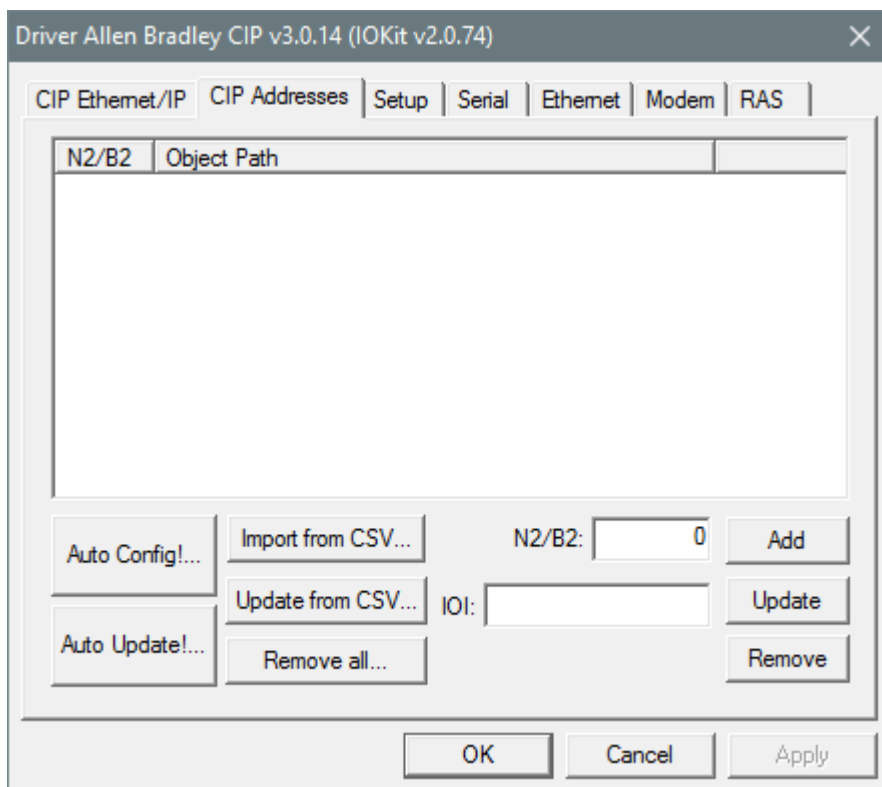
OPÇÃO	DESCRIÇÃO
Start Offline	Indica que este Driver deve partir sem iniciar a comunicação, para que seja possível a alteração, em tempo de execução, de alguns parâmetros antes da conexão. Após definir os parâmetros desejados, o Driver pode ser liberado para comunicação através da escrita do valor 0 (zero) no parâmetro CIP.StartOffline , através dos serviços de alteração de parâmetros do IOLite . Para mais informações, consulte o tópico Alteração de Parâmetros em Tempo de Execução
Port ID	Especifica uma saída do cartão Ethernet e deve ser igual a 1 (um, porta para o <i>backplane</i> do controlador)
CPU Slot Number	Posição de <i>slot</i> onde está o processador, entre 0 (zero) e 255
Operation Retries	Retentativas internas para cada operação de leitura e escrita antes de informar um erro
Async Read Mode	<p>Ao utilizar a opção de leituras assíncronas, informa o tipo de otimização de leitura que é adotado. O modo Blocked tenta encontrar os pais, isto é, os itens que representam o primeiro nível na estrutura de objetos definida pelo usuário no mapa de memória, lendo todos os filhos. Este modo é recomendado quando a maioria dos itens de cada estrutura deve ser realmente lida pela aplicação.</p> <p>Já o modo Non-Blocked identifica cada item filho separadamente, solicitando a leitura destes itens individualmente em um pedido agrupado (geralmente cada grupo contém aproximadamente 30 itens). Este modo é recomendado quando apenas alguns itens de cada estrutura de objetos definida pelo usuário devem ser realmente lidos pela aplicação</p>
CompactLogix Arrays Firmware <20 Compatibility	Selecione esta opção se estiver utilizando um controlador CompactLogix com versão de <i>firmware</i> menor ou igual a 20.XX, que apresenta um deslocamento nos tipos Array usando endereçamento físico (<i>Non-Blocked</i>)
TCP/IP Connections	Permite que sejam definidas entre uma e quatro conexões TCP/IP simultâneas com o controlador, com o objetivo de aumentar a performance da comunicação. Caso a comunicação seja serial, este parâmetro não é utilizado. NOTA: Recomenda-se o uso de leituras assíncronas (veja a opção Use Async Reads) para que as conexões sejam usadas com mais eficiência
Master Address (RS232 only)	Indica o endereço do Driver (Mestre) para comunicação via porta serial
Check Type (RS232 only)	Informe se o cálculo de verificação de erros ao comunicar pela porta serial é CRC ou BCC

NOTA

Este Driver **NÃO** utiliza o sistema padrão de Superblocos do **E3**. A opção **EnableReadGrouping** do Driver deve estar desabilitada.

Aba CIP Addresses

A aba **CIP Addresses** deve ser utilizada para mapeamento dos endereços simbólicos do controlador para os parâmetros numéricos *N* utilizados pelo **Elipse SCADA** versão 2.29 ou anterior.



Aba CIP Addresses

Para o endereçamento dos Tags é necessário definir uma associação entre cada nome de variável dentro do programa do ControlLogix, chamada IOI (*Internal Object Identifier*), e um índice a ser definido no parâmetro *N2* de cada Tag. Essas associações podem ser definidas de três formas:

- **Automaticamente pelo Driver com comunicação com o CLP:** Clique em **Auto Config** e forneça um nome de arquivo com extensão .csv onde a base de dados está armazenada. Cada Tag encontrado é associado com um número sequencial *N2*. A opção **Auto Update** realiza a mesma operação, exceto por incluir apenas as variáveis que não se repetem entre as que já existem na lista (recomendado para atualizações realizadas na memória do PLC).
- **Importar de um arquivo CSV:** O arquivo precisa ter uma coluna com o parâmetro *N2* ou *B2* e uma coluna com o nome IOI. Utilize a opção **Import from CSV** para criar uma nova lista a partir da importação de um arquivo CSV ou utilize a opção **Update from CSV** para atualizar a lista importando de um arquivo CSV apenas as variáveis que não se repetem na lista existente.
- **Digitando diretamente na janela o parâmetro N2 e o IOI correspondente:** O IOI deve representar **TODO** o caminho do item desejado, incluindo o nome do programa ou *task*.

Parâmetros de Endereçamento de Tags

Elipse E3

- **N1:** Se = 1, Indica que a leitura será feita em modo exclusivo (polling individual).
Se = 2, deve ser usado quando o item a ser lido ou escrito é um string de tamanho maior que o padrão (82 caracteres, do tipo ASCIISTRING82).
- **N2:** Não utilizado
- **N3:** Não utilizado
- **N4:** Não utilizado
- **Device:** Não utilizado
- **Item:** Caminho da variável no controlador

Elipse SCADA

- **N1:** Não utilizado
- **N2:** Índice do IOI do Tag (definido na aba **CIP Addresses**)
- **N3:** Não utilizado
- **N4:** Não utilizado

Tipos de Dados Suportados


O ControlLogix suporta vários tipos internos de dados, descritos a seguir:

- **Boolean, Bit, Float, Real, Signed, Unsigned, Int**, etc.
- *Arrays* de tipos padrão (uma, duas ou três dimensões)
- Estruturas pré-definidas do Logix: **Timer, Counter, Axis, SFC_Step**, etc.
- Estruturas definidas pelo usuário
- *Arrays* de estruturas
- Qualquer combinação dos tipos descritos anteriormente

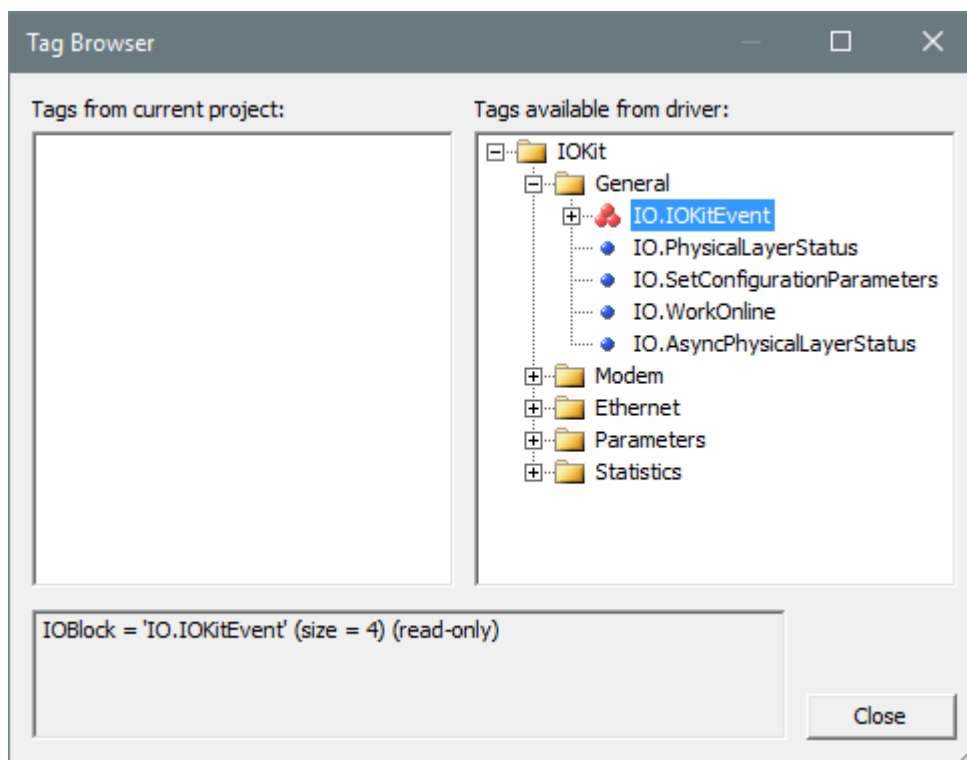
Caso o usuário tenha optado por desabilitar a opção **Import each property as individual tags** ou se está usando o **Elipse SCADA**, cada variável dentro de uma estrutura ou *array* é mapeada para um Elemento de Bloco. Variáveis dentro de *Tasks* são inseridas em pastas ou Blocos com o mesmo nome da *Task*. Os tipos de dados de cada variável são definidos no controlador e automaticamente adaptados, portanto não é necessário definir tipos de dados em cada Tag.

Caso a opção **Import each property as individual tags** esteja selecionada, todos os Tags são criados de forma independente sem utilizar Blocos, seguindo a estrutura de dados criada no controlador.

Janela Tag Browser

O **E3** permite utilizar o serviço de *Tag Browsing online*, que busca do controlador todas as variáveis existentes. Esta função está disponível na opção **Tag Browser**  da janela de edição dos Tags.

Antes de utilizar a opção **Tag Browser**, é necessário especificar corretamente a interface de comunicação (**Ethernet** ou **Serial**), bem como seus parâmetros. Para utilizar os Tags, arraste-os para a listagem **Tags do projeto corrente** (**Tags from current project**).



Janela Tag Browser

Tags IOKit

Quando usados em modo **Ethernet**, os Tags padrão do **IOKit** não funcionam neste Driver. Para realizar ações e obter status básicos do **IOKit**, dois Tags especiais foram criados, descritos na tabela a seguir.

Tags especiais para o IOKit

TAG	CONFIGURAÇÃO	DESCRIÇÃO
Status da Conexão	Device: Não utilizado Item: IO.AsyncPhysicalLayerStatus	Retorna 2 (dois) se todos os canais (entre um e quatro) estão conectados, 1 (um) se algum dos canais ainda está conectando e 0 (zero) se nenhum canal está conectado
IP Switch	Device: Não utilizado Item: IO.Ethernet.AsyncIPSwitch	Da mesma forma que o Tag similar do IOKit , quando este Tag é escrito o Driver alterna a conexão TCP/IP de todos os canais (entre um e quatro) entre os endereços principal e <i>backup</i>

Alteração de Parâmetros em Tempo de Execução

Os parâmetros da tabela a seguir podem ser ajustados em tempo de execução através da escrita no Tag **Set Configuration Parameters** do **IOKit**, definido pelo endereço -1, 0, 0, 3 (*N1.N2.N3.N4*).

Parâmetros em tempo de execução

PARÂMETRO	COMENTÁRIO	VALOR PADRÃO
CIP.StartOffline	Se estiver em 1 (um), permite ao Driver iniciar de forma inativa, para que outros parâmetros sejam configurados em tempo de execução. Após esta definição, esta variável deve ser colocada em 0 (zero), assim o Driver inicia a comunicação com os novos parâmetros	0 (zero)
CIP.CheckDownload		0 (zero)
CIP.Retries		0 (zero)
CIP.PortID		1 (um)
CIP.DefaultSlaveAddress		0 (zero)
CIP.SlotNumber		1 (um)

Exemplo de script de configuração em tempo de execução para o **E3**, supondo que o Driver foi salvo com a opção **StartOffline** selecionada.

```
Dim arr(2)
//Configure o nome e o caminho do Driver
Set Driver = Application.GetObject("DriverABCIP")
arr(1) = Array("CIP.Retries", 2)
arr(2) = Array("CIP.StartOffline", 0)
Driver.Write -1, 0, 0, 3, arr
```

Os parâmetros a seguir não podem ser alterados em tempo de execução:

- **CIP.CheckDownload**
- **CIP.TCPConn**
- **CIP.UseCache**
- **CIP.TotalTargets**
- **CIP.TotalDefs**
- **CIP.ReadType**

Documentação das Interfaces de Comunicação

Esta seção contém a documentação das Interfaces de Comunicação referentes ao **Driver Allen Bradley CIP**.

Configurações do Driver

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

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

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

1. Abra o Organizer.
2. Selecione o 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, se for necessário o acesso a duas portas seriais, é preciso adicionar dois Drivers na aplicação e configurar cada um destes Drivers para cada porta serial.

Caixa de Diálogo de Configuração

A caixa de diálogo das Interfaces de Configuração permitem configurar a conexão de I/O que é utilizada pelo 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 cada Driver) na caixa de diálogo de configuração.

Aba Setup

A aba **Setup** contém a configuração geral do Driver. A aba é dividida em três partes distintas:

- **Configurações gerais:** Configurações da camada física do 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

The screenshot shows the 'Setup' tab of a configuration window. It contains three main sections: 'Physical Layer' with a dropdown set to 'Serial' and a 'Start driver OFFLINE' checkbox; 'Timeout' with a text box set to '1000' and 'ms'; and 'Connection management' with a 'Mode' dropdown set to 'Automatic (managed by the driver)', a checked 'Retry failed connection every 20 seconds' option, and two unchecked options: 'Give up after 1 failed retries' and 'Disconnect if non-responsive for 0 seconds'. Below these is a 'Logging Options' section with an unchecked 'Log to File' checkbox and an empty text box.

Aba Setup

Opções gerais da aba Setup

OPÇÃO	DESCRIÇÃO
Physical Layer	Selecione a interface física em uma lista. As opções são Serial , Ethernet , Modem e RAS . A interface selecionada deve ser configurada na sua 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 byte do <i>buffer</i> de recepção).
Start driver OFFLINE	Selecione esta opção para que o Driver inicie em modo Offline (parado). Isto significa que a interface de I/O não é criada até que se configure o Driver em modo Online (utilizando-se um Tag na aplicação). Este modo possibilita a configuração dinâmica da interface de I/O em tempo de execução. Veja o tópico Trabalhando em Modo Offline para maiores detalhes.

Opções para o grupo Connection management

OPÇÃO	DESCRIÇÃO
Mode	Seleciona o modo de gerenciamento de conexão. Selecionar a opção Automatic permite que o Driver gerencie a conexão automaticamente, como especificado nas opções seguintes. Selecionar a opção Manual permite que a aplicação gerencie a conexão completamente. Veja o tópico Estados do Driver para maiores detalhes.
Retry failed connection every ... seconds	Selecione esta opção para habilitar a retentativa de conexão do Driver em um determinado intervalo, em segundos. Se a opção Give up after failed retries não estiver selecionada, o 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 for atingido, o Driver vai para o modo Offline , assumindo que um problema de hardware foi detectado. Se o 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 o Driver a se desconectar se nenhum byte chegou à interface de I/O no <i>time-out</i> especificado, em segundos. Este <i>time-out</i> deve ser maior que o <i>time-out</i> configurado na opção Timeout .

Opções para o grupo Logging Options

OPÇÃO	DESCRIÇÃO
Log to File	<p>Habilite esta opção e configure o nome do arquivo onde o log é escrito. Arquivos de log podem ser bem extensos, portanto utilize esta opção por curtos períodos de tempo, apenas para o propósito de testes e depurações.</p> <p>Caso se utilize a macro %PROCESS% no nome do arquivo de log, esta é substituída pelo ID do processo atual. Esta opção é particularmente útil ao se utilizar várias instâncias do mesmo Driver no E3, permitindo assim que cada instância gere um arquivo separado de log. Por exemplo, ao configurar esta opção como c:\e3logs\drivers\sim_%PROCESS%.log, gera-se o arquivo c:\e3logs\drivers\sim_00000FDA.log para o processo 0FDAh.</p> <p>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 como c:\e3logs\drivers\sim_%DATE%.log, gera-se o arquivo c:\e3logs\drivers\sim_2005_12_31.log no dia 31 de dezembro de 2005 e o arquivo c:\e3logs\drivers\sim_2006_01_01.log no dia primeiro de janeiro de 2006.</p>

Aba Serial

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

Serial

Port:
Baud rate:
Data bits:
Parity:
Stop bits:
☐ Enable 'ECHO' suppression

Handshaking

DTR control:
RTS control:
☐ Wait for CTS before send
CTS timeout: ms
Delay before send: ms
Delay after send: ms

Inter-byte delay (microseconds): μ s
Inter-frame delay (milliseconds): ms

Aba Serial

Opções gerais da aba Serial

OPÇÃO	DESCRIÇÃO
Port	Selecione a porta serial a partir da lista (de COM1 até COM4) ou digite o nome da porta serial no formato COMn (por exemplo, "COM15"). Ao digitar o nome da porta manualmente, a caixa de diálogo aceita apenas nomes de portas começando com a expressão "COM".
Baud rate	Selecione o <i>baud rate</i> a partir da lista (1200, 2400, 4800, 9600, 19200, 38400, 57600 ou 115200) ou digite o <i>baud rate</i> desejado (por exemplo, 600).
Data bits	Selecione 7 ou 8 bits de dados a partir da lista.
Parity	Selecione a paridade a partir da lista (None, Even, Odd, Mark ou List).
Stop bits	Selecione o número de stop bits a partir da lista (1, 1.5 ou 2 stop bits).
Enable 'ECHO' suppression	Habilite esta opção para remover o eco recebido após a Interface de Comunicação enviar dados pela porta serial. Se o eco não for igual aos bytes recém enviados, a Interface de Comunicação aborta a comunicação.

OPÇÃO	DESCRIÇÃO
Inter-byte delay (microseconds)	Define uma espera entre cada byte transmitido pela Interface de Comunicação, em milionésimos de segundo (1000000 igual a um segundo). Esta opção deve ser utilizada com esperas pequenas (menos de um milissegundo).
Inter-frame delay (milliseconds)	Define uma espera entre pacotes enviados ou recebidos pela Interface de Comunicação, em milésimos de segundo (1000 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* (controlar quando o dado pode ser enviado ou recebido através da 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 RS232 quanto com linhas seriais RS485.

Opções disponíveis no grupo Handshaking

OPÇÃO	DESCRIÇÃO
DTR control	Selecione ON para deixar o sinal DTR sempre ligado enquanto a porta serial está aberta. Selecione 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 ON para deixar o sinal RTS sempre ligado enquanto a porta serial está aberta. Selecione OFF para desligar o sinal RTS enquanto a porta serial está aberta. Selecione 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 para Toggle . Utilize esta opção para forçar o 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 <i>flag</i> de permissão para envio.
CTS timeout	Determina o tempo máximo, em milissegundos, que o Driver aguarda pelo sinal de CTS depois de ligar o sinal de RTS . Se o sinal de CTS não é levantado dentro deste <i>time-out</i> , o Driver falha a comunicação atual e retorna erro.
Delay before send	Alguns hardwares 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 da 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 (todos exceto as configurações da porta) devem ser também configurados para uso na Interface **RAS**.

Aba Ethernet

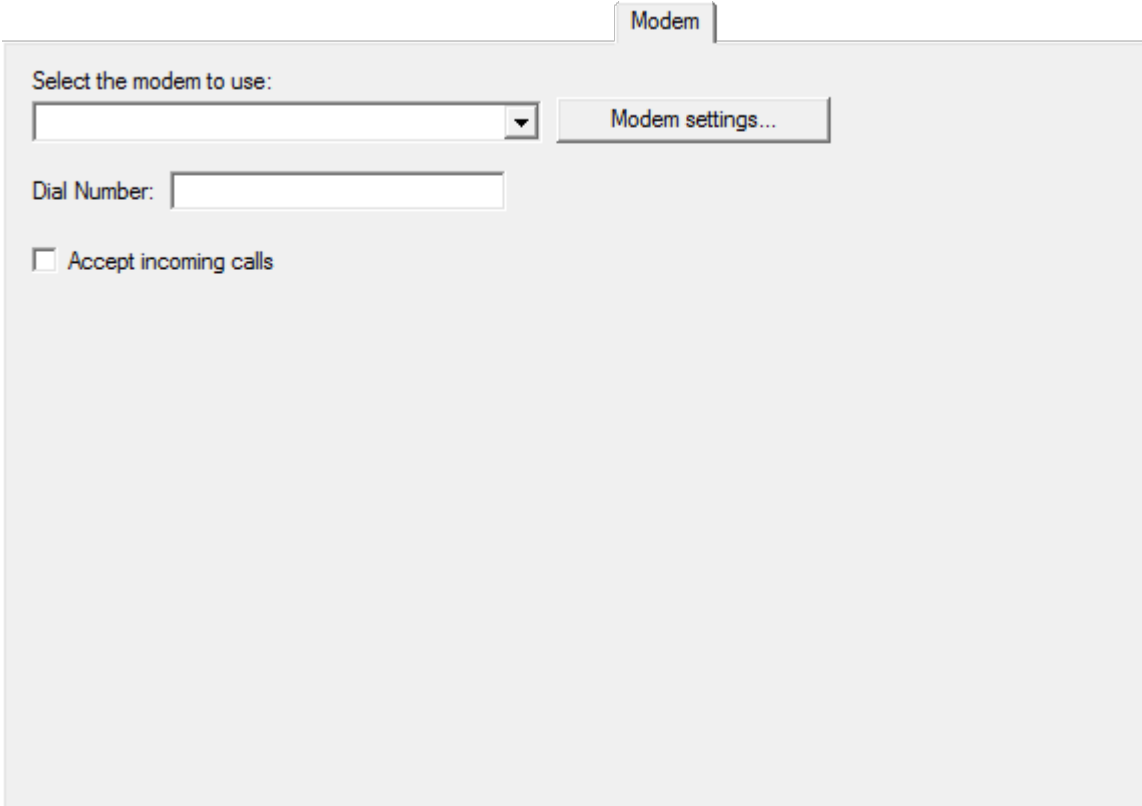
Opções disponíveis na aba Ethernet

OPÇÃO	DESCRIÇÃO
Transport	Selecione TCP/IP para um <i>socket</i> TCP (<i>stream</i>). Selecione UDP/IP para utilizar um <i>socket</i> UDP (<i>connectionless datagram</i>)
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, o Driver se conecta ao endereço e porta especificados na opção 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 por seu endereço IP) que o Driver utiliza para efetuar e receber conexões, ou selecione o item (All Interfaces) para permitir conexões em qualquer interface de rede
Use IPv6	Selecione esta opção para forçar o 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 <i>eco</i> dos dados recebidos. O <i>eco</i> é uma cópia dos dados que foram enviados, que pode ser retornada antes da mensagem de resposta

OPÇÃO	DESCRIÇÃO
IP Filter	Lista de endereços IP restringidos ou permitidos de onde o Driver aceita conexões (<i>Firewall</i>). Veja a propriedade IO.Ethernet.IPFilter para mais detalhes
Main IP Backup IP 1 Backup IP 2 Backup IP 3	<p>Estas opções permitem configurar até quatro endereços para o dispositivo remoto:</p> <ul style="list-style-type: none"> • IP: Digite o endereço IP do dispositivo remoto. Pode-se usar tanto o endereço IP separado por pontos quanto uma URL. No caso de uma URL, o Driver usa o serviço de DNS disponível para mapear a URL para um endereço IP. Por exemplo, "192.168.0.13" ou "Server1" • Port: Digite a porta IP do dispositivo remoto (de 0 até 65535) • Specify local port: Selecione esta opção para utilizar uma porta local fixa ao conectar ao dispositivo remoto
PING before connecting	<p>Habilite esta opção para executar um comando ping (verificar se o dispositivo pode ser encontrado na rede) no dispositivo antes de tentar uma conexão com o <i>socket</i>. Esta é uma maneira rápida de determinar uma conexão bem sucedida antes de tentar abrir um <i>socket</i> com o dispositivo (o <i>time-out</i> de uma conexão com um <i>socket</i> pode ser bem alto):</p> <ul style="list-style-type: none"> • Timeout: Especifique o número de milissegundos de espera por uma resposta do comando ping. Deve-se usar o 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 (entre um e quatro segundos) • Retries: Número de retentativas do comando ping (não conta a tentativa inicial). Se todas as tentativas falharem, então a conexão com o <i>socket</i> é abortada

Aba Modem

Utilize esta aba para configurar os parâmetros da Interface **Modem**. Algumas opções da aba **Serial** afetam a configuração do 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 a opção Default modem , então o primeiro modem disponível é utilizado. Selecionar esta opção é recomendado especialmente quando a 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 (espera pelo tom de discagem). Por exemplo, "0w33313456" (disca o número zero, espera e então disca o número "33313456").
Accept incoming calls	Habilite esta opção para que o Driver atenda o telefone quando receber uma chamada externa. Para utilizar esta opção é necessário configurar a opção Connection management na aba Setup para 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. O 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 seguintes passos de inicialização ou de conexão são efetuados:

1. Limpeza do *socket* (remove qualquer mensagem de saudação **TELNET** recebida do dispositivo RAS).
2. Envio do comando de discagem **AT** (em 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 o dispositivo (a conexão foi estabelecida).

Se o passo 5 é efetuado com sucesso, então o *socket* comporta-se como um *socket* normal, com o dispositivo RAS funcionando como um roteador entre o Driver e o dispositivo. Os bytes enviados pelo 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 ao Driver utilizando o mesmo *socket*.

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

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

Aba RAS

Opções disponíveis na aba RAS

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

Configurações Gerais

Esta seção contém informações sobre a configuração dos **Tags de Comunicação** e as **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.IOKitEvent

Tipo de Tag	Tag Bloco
Tipo de Acesso	Somente Leitura
Parâmetro B1	-1
Parâmetro B2	0
Parâmetro B3	0
Parâmetro B4	1
Propriedade Size	4
Propriedade ParamItem	IO.IOKitEvent

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

- **Elemento 0:** Tipo de evento
 - **0:** Informação
 - **1:** Advertência
 - **2:** Erro
- **Elemento 1:** Fonte do evento
 - **0:** Driver (específico do Driver)
 - **-1:** IOKit (eventos genéricos da Interface de Comunicação)
 - **-2:** Interface **Serial**

- -3: Interface **Modem**
- -4: Interface **Ethernet**
- -5: Interface **RAS**
- **Elemento 2:** Número do erro (específico de cada fonte de evento)
- **Elemento 3:** Mensagem do evento (**String**, específica de cada evento)

NOTA

O 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
Parâmetro N2	0
Parâmetro N3	0
Parâmetro N4	2
Configuração por String	IO.PhysicalLayerStatus

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

- **0:** Camada física parada (o 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 (o Driver está em modo **Online**, mas a camada física não está conectada. Se a opção **Connection management** estiver configurada para **Automatic**, a camada física pode estar conectando, desconectando ou esperando por uma tentativa de reconexão. Se a opção **Connection management** estiver configurada para **Manual**, então a camada física permanece neste estado até ser forçada a conectar)
- **2:** Camada física conectada (a camada física está pronta para ser usada). Isto **NÃO** significa que o 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
Parâmetro B2	0
Parâmetro B3	0
Parâmetro B4	3
Propriedade Size	2
Propriedade ParamItem	IO.SetConfigurationParameters

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

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

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

```
// 'Block' deve ser um Tag Block 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 Block inteiro
Block.Write()
```

Ao usar o **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** do Driver para enviar os parâmetros diretamente para o Driver, sem a necessidade de criar um Tag. Veja estes exemplos:

```
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 do array 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
```

O Driver não valida nomes de parâmetros ou valores passados, por isto tenha cuidado ao escrever parâmetros e valores. O comando **Write** falha se o *array* de configuração é criado incorretamente. Pode-se consultar o log do Driver ou usar o parâmetro *writeStatus* do método **WriteEx** para descobrir a causa exata do 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
Parâmetro N2	0
Parâmetro N3	0
Parâmetro N4	4
Configuração por String	IO.WorkOnline

Este Tag informa o estado atual do Driver e permite iniciar ou parar a camada física.

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

No exemplo a seguir (utilizando o **E3**), o Driver é colocado em modo **Offline**, sua 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 quando estiver configurando o Driver em modo **Online** (escrevendo o valor um). Neste caso, o 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**)
- O Driver pode ter ficado sem memória
- A camada física pode ter deixado de criar seu *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, falha ao inicializar o TAPI (modem), etc. A causa é gravada no arquivo de log

IMPORTANTE

Mesmo que a configuração do Driver para o modo **Online** seja bem sucedida, isto não significa necessariamente que a camada física esteja pronta para uso (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:

- 0: Modo automático (o Driver gerencia a conexão)
- 1: Modo manual (a aplicação gerencia a conexão)

IO.GiveUpEnable

■ Quando configurada para Verdadeiro, define um número máximo de tentativas de reconexão. Se todas as reconexões falharem, o Driver entra em modo **Offline**. Se configurada para Falso, o 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), o Driver tenta apenas uma reconexão quando a conexão é perdida. Se esta falhar, o 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 estiver 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 estiver inativa por este período de tempo, ela é desconectada.

IO.RecoverEnable

■ Configure em Verdadeiro para habilitar o Driver a recuperar conexões perdidas e em Falso para deixar o 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 o Driver em modo **Offline** e em Falso para iniciar o Driver em modo **Online**.

NOTA

Não faz sentido modificar esta propriedade em tempo de execução, já que ela só pode ser modificada quando o Driver já estiver em modo **Offline**. Para configurar o 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 é igual a 1000 milissegundos).

IO.Type

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

- **N ou None**: Não utiliza uma interface física (o Driver deve fornecer uma interface personalizada)
- **S ou Serial**: Utiliza uma porta serial local (COMn)
- **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*). O Driver conecta-se ao 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 as **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
Parâmetro N2	0
Parâmetro N3	0
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
Parâmetro N2	0
Parâmetro N3	0
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
Parâmetro N2	0
Parâmetro N3	0
Parâmetro N4	1102
Configuração por String	IO.Stats.Partial.TimeConnectedSeconds

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

IO.Stats.Partial.TimeDisconnectedSeconds

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

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

IO.Stats.Total.BytesRecv

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

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

IO.Stats.Total.BytesSent

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

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

IO.Stats.Total.ConnectionCount

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

Este Tag retorna a quantidade de conexões que o 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
Parâmetro N2	0
Parâmetro N3	0
Parâmetro N4	1002
Configuração por String	IO.Stats.Total.TimeConnectedSeconds

Este Tag retorna o número de segundos que o 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
Parâmetro N2	0
Parâmetro N3	0
Parâmetro N4	1003
Configuração por String	IO.Stats.Total.TimeDisconnectedSeconds

Este Tag retorna o número de segundos que o 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 as **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 (também são válidos quando a Interface **RAS** estiver selecionada).

IMPORTANTE

Estes Tags estão disponíveis **SOMENTE** enquanto o Driver estiver 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
Parâmetro N4	0 (zero)
Configuração por String	IO.Ethernet.IPSelect

Indica o endereço IP ativo. Os valores possíveis são os seguintes:

- **0:** O endereço principal de IP está selecionado
- **1:** O endereço alternativo (*backup*) de IP está selecionado
- **2:** O endereço alternativo (*backup*) 2 de IP está selecionado
- **3:** O endereço alternativo (*backup*) 3 de IP está selecionado

Se a Interface **Ethernet** (ou **RAS**) estiver conectada, este Tag indica qual dos quatro endereços IP configurados está em uso. Se a Interface estiver 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 estiver 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 o Driver estiver atualmente conectado. Se o Driver estiver 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
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 estiver ativo, então o primeiro endereço alternativo de IP (*backup*) é ativado, e assim por diante passando por todos os endereços alternativos de IP e voltando para o endereço principal até conseguir uma conexão.

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

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 o Driver não deve aceitar conexões externas (o Driver se comporta como mestre) ou configure em Verdadeiro para habilitar a recepção de conexões (o Driver se comporta como escravo).

IO.Ethernet.BackupEnable

■ Configure em Verdadeiro para habilitar o endereço IP de reserva (*backup*). Se a tentativa de reconectar com o endereço IP principal falhar, o Driver tenta utilizar o endereço IP de reserva. Configure em Falso para desabilitar sua utilização.

IO.Ethernet.BackupIP

📌 Endereço IP alternativo (*backup*) do equipamento de destino. Pode-se utilizar tanto o endereço numérico como o nome de *host* do equipamento, como por exemplo "192.168.0.7" ou "SERVER2".

IO.Ethernet.BackupLocalPort

9 Número da porta local a ser utilizada na conexão ao endereço IP alternativo do equipamento de destino. Usado apenas se **IO.Ethernet.BackupLocalPortEnable** for Verdadeiro.

IO.Ethernet.BackupLocalPortEnable

■ Configure em Verdadeiro para forçar o uso de uma porta local específica ao conectar ao endereço IP de reserva (*backup*). Configure em Falso para utilizar qualquer porta local disponível.

IO.Ethernet.BackupPort

9 Número da porta do endereço IP alternativo do equipamento de destino (usado juntamente com a propriedade **IO.Ethernet.BackupIP**).

IO.Ethernet.IPFilter

A Lista de endereços IPv4 ou IPv6 separados por vírgula, que define de quais endereços o 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. Exemplos:

- **192.168.0.24**: Aceita apenas conexões do endereço IPv4 192.168.0.24
- **192.168.0.41-50**: Aceita conexões dos endereços IPv4 no intervalo entre 192.168.0.41 e 192.168.0.50
- **192.168.0.***: Aceita conexões dos endereços IPv4 no intervalo entre 192.168.0.0 e 192.168.0.255
- **fe80:3bf:877::** (expande para fe80:03bf:0877:0000:0000:0000:0000:0000 e fe80:03bf:0877:0000:0000:0000:ffff:ffff)**: 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 o 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 corresponder 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 que não forem 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 que não forem encontrados na lista são autorizados

Se um endereço IP aparecer 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, é logada a mensagem "Blocked incoming socket connection from {IP}!".

No caso de conexões UDP em modo escuta em *broadcast*, onde o 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, é logada a mensagem "Blocked incoming packet from {IP} (discarding {N} bytes)!".

IO.Ethernet.ListenIP

A Endereço IP da interface local de rede por onde o Driver efetua e aceita conexões. Deixe esta propriedade vazia para efetuar e aceitar conexões por qualquer interface local de rede.

IO.Ethernet.ListenPort

9 Número da porta IP utilizada pelo Driver para escutar conexões.

IO.Ethernet.MainIP

A Endereço IP do equipamento de destino. Pode-se utilizar tanto o endereço numérico como o nome de *host* do equipamento, como por exemplo "192.168.0.7" ou "SERVER2".

IO.Ethernet.MainLocalPort

9 Número da porta local a ser utilizada na conexão ao endereço IP principal do equipamento de destino. 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. Configure em Falso para utilizar qualquer porta local disponível.

IO.Ethernet.MainPort

9 Número da porta IP no equipamento de destino (usado juntamente com a propriedade **IO.Ethernet.MainIP**).

IO.Ethernet.PingEnable

■ Configure em Verdadeiro para habilitar o envio de um comando **ping** para o endereço IP do equipamento de destino, antes de tentar conectar-se ao *socket*. O *time-out* de conexão do *socket* não pode ser controlado, por isto o envio de um comando **ping** antes de conectar-se é uma maneira rápida de detectar se a conexão vai falhar. Configure em Falso para desabilitar o comando **ping**.

IO.Ethernet.PingTimeoutMs

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

IO.Ethernet.PingTries

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

IO.Ethernet.ShareListenPort

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

IO.Ethernet.SuppressEcho

■ 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 o Driver enviou para o equipamento.

IO.Ethernet.Transport

A Define o protocolo de transporte. Os valores possíveis são os seguintes:

- **T ou TCP**: Utiliza o protocolo TCP/IP
- **U ou UDP**: Utiliza o protocolo UDP/IP

IO.Ethernet.UseIPv6

■ Configure em Verdadeiro para utilizar endereços IPv6 em todas as conexões Ethernet. Configure em Falso para utilizar endereços IPv4 (este é valor padrão).

Configuração da Interface Modem

Esta seção contém informações sobre a configuração dos **Tags de Comunicação** e as **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 o Driver estiver em modo **Online**.

IO.TAPI.ConnectionBaudRate

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Somente Leitura
Parâmetro N1	-1
Parâmetro N2	0
Parâmetro N3	3
Parâmetro N4	5
Configuração por String	IO.TAPI.ConnectionBaudRate

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

IO.TAPI.Dial

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Somente Escrita
Parâmetro N1	-1
Parâmetro N2	0
Parâmetro N3	3
Parâmetro N4	1
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 a chamada é estabelecida.

IO.TAPI.ModemStatus

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Somente Leitura
Parâmetro N1	-1
Parâmetro N2	0
Parâmetro N3	3
Parâmetro N4	2
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!"**: O Driver não conseguiu inicializar a linha do modem. Confira o arquivo de log do Driver para maiores detalhes
- **"Modem error at dial!"**: O Driver não conseguiu começar ou aceitar uma chamada
- **"Connecting..."**: O 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!"**: O Driver conectou-se com sucesso (completou ou aceitou uma chamada externa)
- **"Disconnecting..."**: O Driver está desligando a chamada atual
- **"Disconnected OK!"**: O Driver desligou a chamada atual
- **"Error: no dial tone!"**: O Driver abortou a chamada porque o sinal de linha disponível não foi detectado
- **"Error: busy!"**: O Driver abortou a ligação porque a linha estava ocupada
- **"Error: no answer!"**: O 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
Parâmetro N2	0
Parâmetro N3	3
Parâmetro N4	0
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**.

IO.TAPI.HangUp

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Somente Escrita
Parâmetro N1	-1
Parâmetro N2	0
Parâmetro N3	3
Parâmetro N4	4
Configuração por String	IO.TAPI.HangUp

Qualquer valor escrito neste Tag desliga a chamada atual.

NOTA

Use este comando apenas quando estiver gerenciando a camada física manualmente, ou se estiver explicitamente tentando forçar o Driver a reiniciar a comunicação. Se a camada física estiver configurada para reconexão automática, o Driver imediatamente tenta restabelecer a conexão.

IO.TAPI.IsModemConnected

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Somente Leitura
Parâmetro N1	-1
Parâmetro N2	0
Parâmetro N3	3
Parâmetro N4	3
Configuração por String	IO.TAPI.IsModemConnected

Este Tag indica o estado da conexão do modem. Os valores possíveis são os seguintes:

- **0:** O modem não está conectado, mas pode estar realizando ou recebendo uma chamada externa
- **1:** O modem está conectado e o Driver completou ou recebeu uma chamada externa com sucesso. Enquanto estiver 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
Parâmetro N2	0
Parâmetro N3	3
Parâmetro N4	6
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 os seguintes:

- **0**: O modem não está conectado
- **1**: O modem está conectando (realizando ou recebendo uma chamada externa)
- **2**: O modem está conectado. Enquanto estiver neste estado, a camada física consegue enviar ou receber dados
- **3**: O modem está desconectando a chamada atual

Propriedades

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

IO.TAPI.AcceptIncoming

9 Configure em Falso se o modem não pode aceitar chamadas externas (o Driver se comporta como mestre) e configure em Verdadeiro para habilitar a recepção de chamadas (o Driver se comporta como escravo).

IO.TAPI.ModemID

9 É o número de identificação do modem. Este ID é criado pelo Windows e é usado internamente para identificar o modem dentro de uma lista de equipamentos instalados no computador. Este ID pode não permanecer válido caso o modem seja reinstalado ou a aplicação seja executada em outro computador.

NOTA

Recomenda-se que esta propriedade seja configurada em 0 (zero), indicando que o Driver deve utilizar o primeiro modem disponível.

IO.TAPI.PhoneNumber

A O número de telefone utilizado em comandos **Dial**. Por exemplo, "0w01234566" (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 as **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 o equipamento RAS a realizar uma ligação usando o canal RAS atual. Exemplo: "ATDT6265545".

IO.RAS.CommandTimeoutSec

 Tempo de espera pela mensagem **CONNECT** em resposta ao comando **AT**, em segundos.

Configuração da Interface Serial

Esta seção contém informações sobre a configuração dos **Tags de Comunicação** e as **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

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

IO.Serial.CTSTimeoutMs

 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 expirar, o Driver aborta o envio de bytes através da porta serial. Disponível apenas quando a propriedade **IO.Serial.RTS** está configurada em **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 os seguintes:

- **5:** Cinco bits de dados
- **6:** Seis bits de dados
- **7:** Sete bits de dados
- **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 em **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 em **Toggle** e a propriedade **IO.Serial.WaitCTS** está configurada em Falso.

IO.Serial.DTR

A Indica o modo como o Driver lida com o sinal **DTR**:

- **OFF:** Sinal **DTR** sempre desligado
- **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 os seguintes:

- **E ou Even:** Paridade par
- **N ou None:** Sem paridade
- **O ou Odd:** Paridade ímpar
- **M ou Mark:** Paridade de marca
- **S ou Space:** Paridade de espaço

IO.Serial.Port

9 Número da porta serial local:

- **1:** Utiliza a porta COM1
- **2:** Utiliza a porta COM2
- **3:** Utiliza a porta COM3
- **n:** Utiliza a porta COMn

IO.Serial.RTS

A Indica como o Driver lida com o sinal **RTS**:

- **OFF:** Sinal **RTS** sempre desligado
- **ON:** Sinal **RTS** sempre ligado
- **Toggle:** Liga (**ON**) o sinal **RTS** quando estiver transmitindo dados e desliga (**OFF**) o sinal **RTS** quando não estiver transmitindo dados

IO.Serial.StopBits

9 Especifica o número de bits de parada para a configuração da porta serial. Os valores possíveis são os seguintes:

- **1:** Um bit de parada
- **2:** Um bit e meio de parada
- **3:** Dois bits de parada

IO.Serial.SuppressEcho

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

IO.Serial.WaitCTS

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

Histórico de Revisões do Driver

VERSÃO	DATA	AUTOR	COMENTÁRIOS
3.0.23	20/08/2020	M. Salvador	<ul style="list-style-type: none">• Leituras e escritas de strings grandes não funcionavam (Case 29327)
3.0.21	11/09/2019	M. Salvador	<ul style="list-style-type: none">• Correção de escritas causado na versão anterior (Case 27554)• Variáveis ficam sem atualização em casos específicos (Case 27549)

VERSÃO	DATA	AUTOR	COMENTÁRIOS
3.0.20	19/08/2019	C. Mello	<ul style="list-style-type: none"> Atualização de plataforma no código fonte do driver (Case 27381).
3.0.19	17/09/2018	M. Salvador	<ul style="list-style-type: none"> Ajustes no processo de conexão com os controladores (Case 25138).
3.0.18	05/09/2018	M. Salvador	<ul style="list-style-type: none"> Adicionado sistema de cache para as configurações do controlador (Case 24866). Melhorias no processo de reconexão com a rede (Case 24790).
3.0.14	06/12/2016	M. Salvador	<ul style="list-style-type: none"> Corrigida a leitura de tipos de dados inteiros de 64 bits (Case 20933).
3.0.12	26/02/2016	M. Salvador	<ul style="list-style-type: none"> Agora caso um ou mais serviços de um pedido Non-Blocked retorne erro, somente os Tags correspondentes a estes serviços retornam erro de comunicação (Case 20505).
3.0.11	28/01/2015	M. Salvador	<ul style="list-style-type: none"> Criado um mecanismo para atualização do endereço IP em tempo de execução, StartOffline (Case 14612). Removido o envio de mensagem de identificação do controlador em cada conexão (Case 17847).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
3.0.7	08/09/2014	M. Salvador	<ul style="list-style-type: none"> • Melhorias no processo de reconexão com o controlador após uma falha (<i>Case 17088</i>). • Corrigido o funcionamento da opção de monitoramento de <i>download</i> do programa (<i>Case 16353</i>). • Corrigido um problema de aumento de consumo de memória ao realizar muitas escritas continuamente (<i>Case 16369</i>). • Implementada uma maior tolerância à falhas de comunicação durante o processo de inicialização (<i>Case 17076</i>). • Criada uma nova opção de configuração que permite adicionar um <i>offset</i> de quatro bytes na comunicação de <i>arrays</i> em versões de <i>firmware</i> de controladores menor ou igual a 20, na opção Non-Blocked (<i>Case 17082</i>).
3.0.4	28/01/2014	M. Salvador	<ul style="list-style-type: none"> • Agora cada canal TCP/IP tem um controle <i>KeepAlive</i> que mantém a conexão ativa através de uma troca de mensagens de identificação a cada 10 segundos (<i>Case 15623</i>). • Driver portado para o IOKit v2.00 (<i>Case 14026</i>).
2.2.1	22/02/2013	M. Salvador	<ul style="list-style-type: none"> • Se há uma queda de comunicação durante o processo de inicialização, o Driver agora descarta as estruturas de dados parcialmente obtidas e reinicia o processo de inicialização. (<i>Case 13530</i>).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
2.1.1	18/04/2011	M. Salvador M. Ludwig	<ul style="list-style-type: none"> • Corrigido um possível valor errado dos bits (<i>Case 11951</i>). • Implementadas múltiplas conexões TCP/IP (<i>Case 11109</i>). • Implementada a verificação de <i>download</i> de programa. • Implementado suporte a tipos de dados String definidos pelo usuário e ainda uso de Strings em <i>arrays</i> ou <i>structs</i>. • Implementadas as leituras assíncronas (<i>Case 11109</i>). • Corrigido um problema com determinadas estruturas de dados que não estavam sendo corretamente mapeadas pelo Driver (<i>Case 10109</i>). • Implementada a opção Auto Update (<i>Case 10026</i>). • Melhorias na interface CIP Addresses (<i>Case 10026</i>). • Criados Tags IOKit especiais para quando o Driver usa comunicação TCP/IP (<i>Case 11387</i>).
1.6.1	10/10/2006	M. Salvador	<ul style="list-style-type: none"> • O Tag Browser agora pode ser aberto independente de a janela de configuração já ter sido aberta. • Corrigidos erros de leitura para <i>arrays</i> de estruturas. • Os Tags lidos pelo Tag Browser agora aparecem ordenados alfabeticamente. • Adicionada uma barra de progresso na janela Tag Browser para indicar o progresso de identificação dos Tags.
1.5.1	25/09/2006	M. Salvador	<ul style="list-style-type: none"> • Revisão de código.
1.0.1		M. Salvador	<ul style="list-style-type: none"> • Versão inicial do Driver.

Matriz

Rua 24 de Outubro, 353 - 10º andar
90510-002 Porto Alegre
Fone: (+55 51) 3346-4699
Fax: (+55 51) 3222-6226
E-mail: elipse-rs@elipse.com.br

Filial PR

Av. Sete de Setembro, 4698/1705
80240-000 Curitiba - PR
Fone: (+55 41) 4062-5824
E-mail: elipse-pr@elipse.com.br

Filial RJ

Praia de Botafogo, 300/525
22250-044 Rio de Janeiro - RJ
Fone: (+55 21) 2158-1015
Fax: (+55 21) 2158-1099
E-mail: elipse-rj@elipse.com.br

Filial SP

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

Filial MG

Rua Antônio de Albuquerque, 156
7º andar Sala 705
30112-010 Belo Horizonte - MG
Fone: (+55 31) 4062-5824
E-mail: elipse-mg@elipse.com.br

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)