

# Driver DNP 3.0 Slave

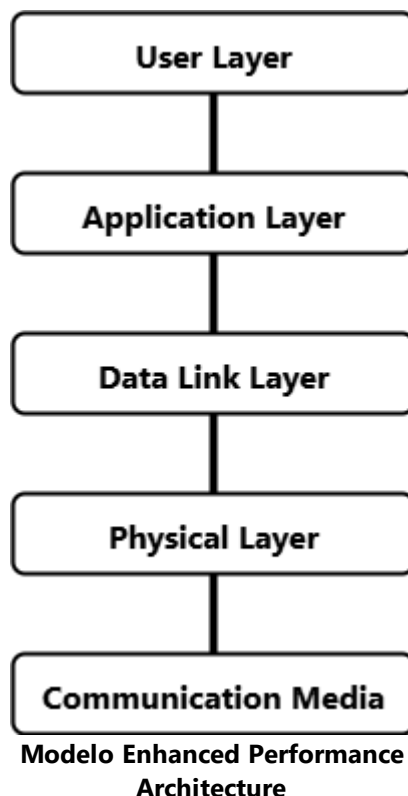
<b>Nome do Arquivo</b>	DNPSlave.dll
<b>Fabricante</b>	DNP (Distributed Network Protocol)
<b>Equipamentos</b>	
<b>Protocolo</b>	DNP 3.0
<b>Versão</b>	4.0.75
<b>Última Atualização</b>	02/09/2025
<b>Plataforma</b>	Win32
<b>Dependências</b>	IOKit versão 1.15 ou superior
<b>Leitura de Superblocos</b>	Não
<b>Nível</b>	31201

## Introdução

Este Driver implementa o protocolo DNP 3.0 no modo **Escravo** (*Outstation*), de acordo com os níveis 2 (dois) e 3 (três).

O DNP (*Distributed Network Protocol*) é um protocolo de comunicações aberto e não proprietário, baseado nas especificações da IEC (*International Electrotechnical Commission*), adaptado para uso em aplicações altamente seguras, à velocidade e quantidade de dados moderada. É extremamente flexível e pode ser utilizado em qualquer plataforma de hardware.

O modelo especificado pela ISO - OSI (*International Standards Organization - Open System Interconnection*) estabelece 7 (sete) camadas para um protocolo de rede. Já a IEC especifica um modelo simplificado, que consiste nas camadas física, *data link* e aplicação somente. Tal modelo é chamado **EPA** (*Enhanced Performance Architecture*). A figura a seguir mostra a estrutura do modelo EPA e o respectivo sistema de comunicação.



A **Camada do Usuário** (*User Layer*) pode ser definida como o local em que o usuário manipula os dados depois de todas as comunicações. Nas aplicações da **Elipse Software**, esta camada é representada pela aplicação de usuário. A camada do Usuário usa a camada de Aplicação deste Driver para enviar ou receber mensagens completas para ou de uma estação.

A **Camada de Aplicação** (*Application Layer*) é responsável por especificar em detalhes os pedidos da camada do Usuário, e de volta à esta quando a mensagem vem da camada *Data Link*. Em outras palavras, esta camada junta as mensagens da camada do Usuário, os *fragmentos*, em uma mensagem de múltiplos fragmentos com informação completa para ser processada e enviada para uma estação pela camada *Data Link*.

A **Camada Data Link** (*Data Link Layer*) é usada para passar mensagens entre as estações primária, ou originadora, e secundária, ou recebedora. Esta camada também empacota os dados, verifica-os em relação aos erros de transmissão e envia para a rede TCP/IP.

O protocolo DNP pode ser configurado para trocas de mensagens via *polling* ou comunicação constante, ou via integridade ou mudanças, que é mais eficiente. O envio das mudanças, também conhecido como RBE (*Report by Exception*), pode ocorrer de forma espontânea ou não solicitada ou não espontânea, através da solicitação explícita do lado **Mestre** pelas mudanças.

## Códigos de Função

Um código de função identifica o propósito de uma mensagem. Existem grupos de funções para pedidos e para respostas. Há vários tipos de funções para pedidos, como mostrado na tabela a seguir. As funções de resposta são usadas internamente e não são abertas ao usuário.

**Códigos de função**

CÓDIGO	FUNÇÃO	DESCRIÇÃO
1	Read	Solicita os Objetos especificados da estação remota e responde com os Objetos pedidos que estão disponíveis
2	Write	Armazena os Objetos especificados na estação remota e responde com o status da operação
3	Select	Seleciona ou arma pontos de saída mas não configura ou produz qualquer ação, tais como controles, <i>setpoints</i> ou saídas analógicas, e responde com o status da operação. A função <b>Operate</b> deve ser usada para ativar estas saídas
4	Operate	Configura ou produz ações nas saídas ou pontos previamente selecionados com a função <b>Select</b>
5	Direct Operate	Seleciona e opera as saídas especificadas e responde com o status dos pontos de controle
6	Direct Operate NO ACK	Seleciona e opera as saídas especificadas e não envia resposta, porém leva menos tempo
7	Immediate Freeze	Copia os Objetos especificados para um <i>buffer</i> de congelamento

CÓDIGO	FUNÇÃO	DESCRIÇÃO
9	Freeze and Clear	Copia os Objetos especificados para um <i>buffer</i> de congelamento e depois zera os Objetos
10	Freeze and Clear No ACK	Copia os Objetos especificados para um <i>buffer</i> de congelamento e depois zera os Objetos, mas não envia resposta de confirmação
11	Freeze with Time	Copia os Objetos especificados para um <i>buffer</i> de congelamento no instante especificado e responde com o status da operação
13	Cold Restart	Realiza uma sequência de <i>reset</i>
14	Warm Restart	Realiza uma sequência parcial de <i>reset</i>
20	Enable Unsolicited Messages	Habilita a notificação espontânea dos Objetos especificados
21	Disable Unsolicited Messages	Desabilita a notificação espontânea dos Objetos especificados
22	Assign Classes	Associa os Objetos especificados com uma Classe
23	Delay Measurement	Permite à uma aplicação calcular o atraso ou tempo de propagação para uma estação em particular

Após a especificação do propósito da mensagem (APCI), segue a segunda parte, se necessária, chamada de ASDU. Cada ASDU consiste em um ou mais identificadores de dados (DUI), cabeçalhos de Objetos e informações de Objetos ou campos de dados.

## Cabeçalho de Objetos

O cabeçalho de Objeto de uma mensagem identifica os Objetos de dados que estão contidos naquela mensagem ou que são utilizados na resposta para aquela mensagem. É composto basicamente de um **Objeto**, uma **Variação**, um **Qualificador** e uma **Faixa**.

## Objetos

Os dispositivos inteligentes que usam a camada de Aplicação do protocolo DNP são capazes de monitorar, controlar e produzir um grande número de dados. Estes dados, chamados de **Elementos de Informação**, são processados e armazenados como Objetos de informação, que são padronizados de forma a atingir meios de descrevê-los e representá-los de forma única. As categorias de Objetos de dados existentes são as seguintes:

- **Objetos Estáticos (Static Objects):** São os Objetos que refletem o valor atual de uma variável de campo ou interna
- **Objetos de Evento (Event Objects):** São os Objetos gerados como resultado de uma mudança de valor ou outro estimulante. São Objetos históricos, ou seja, refletem o valor de um dado em algum instante no passado
- **Objetos Estáticos Congelados (Frozen Static Objects):** Refletem o valor congelado atual de uma variável de campo ou interna. Dados são congelados como resultado de um pedido de congelamento de dados

- **Objetos de Evento Congelados (*Frozen Event Objects*):** São os Objetos resultantes da mudança de um valor congelado ou outro estímulo. São Objetos históricos, ou seja, refletem o valor de um dado em algum instante no passado

Cada categoria é representada com um Objeto diferente, conforme descrito na tabela a seguir.

### Objetos de dados

OBJETO	DESCRIÇÃO
<b>Entradas Digitais</b>	O grupo de entradas digitais contém todos os Objetos que representam entradas binárias, ou seja, status ou atributos Booleanos. Varia entre 1 (um) e 9 (nove)
<b>Saídas Digitais</b>	O grupo de saídas digitais contém todos os Objetos que representam saídas binárias ou informação de controle de relés. Os valores possíveis variam entre 10 e 19
<b>Contadores</b>	Este grupo contém todos os Objetos contadores. Os valores possíveis variam entre 20 e 29
<b>Entradas Analógicas</b>	Contém todas as entradas analógicas. Os valores possíveis variam entre 30 e 39
<b>Saídas Analógicas</b>	Contém todas as saídas analógicas. Os valores possíveis variam entre 40 e 49
<b>Tempo</b>	Contém todos os Objetos que representam tempo em forma absoluta ou relativa. Os valores possíveis variam entre 50 e 59
<b>Classes</b>	Este grupo contém todos os Objetos que representam Classes ou prioridades de dados. Os valores possíveis variam entre 60 e 69
<b>Arquivos</b>	Arquivos ou sistema de arquivos. Os valores possíveis variam entre 70 e 79
<b>Dispositivos</b>	Os valores possíveis variam entre 80 e 89
<b>Aplicações</b>	Objetos que representam aplicações ou processos do sistema operacional. Os valores possíveis variam entre 90 e 99
<b>Numéricos alternativos</b>	Representações numéricas personalizadas. Os valores possíveis variam entre 100 e 109

É importante observar que um Objeto estático, quando sofre variação, é capaz de gerar um Objeto de evento indicando esta mudança. Entretanto, ambos representam o mesmo Objeto.

## Variação

São modificações ou sub-tipos que podem ocorrer nos Objetos, como por exemplo uma entrada digital, que pode ser representada apenas por um único bit (zero ou um), por uma palavra de status (um byte) ou ainda conter ou não a informação de tempo (*timestamp*). Sendo assim, a combinação de um Objeto mais a Variação descreve completamente uma informação, conforme os exemplos a seguir.

- **Objeto 01:** Variação 01 representa uma entrada digital sem status (apenas um bit)
- **Objeto 01:** Variação 02 representa uma entrada digital com status (um byte)

- **Objeto 02:** Variação 01 representa uma mudança de entrada digital sem a informação de tempo
- **Objeto 02:** Variação 02 representa uma mudança de entrada digital com a informação de tempo

Apesar de ambos Objetos estarem relacionados à mesma entrada digital, esta informação pode ser simbolizada de maneiras diferentes.

## Qualificador

Especifica o significado do campo **Faixa** (*Range*).

## Faixa

Indica a quantidade de Objetos, os índices iniciais e finais ou identificadores para os Objetos.

## Classes

Os Objetos declarados em um sistema ou equipamento que implementa o protocolo DNP no modo **Escravo** podem ser agrupados em Classes. O protocolo DNP define as Classes padronizadas a seguir.

- **Classe 0:** Significa todos os Objetos, ou seja, na inicialização deste Driver no lado **Mestre** pode realizar um pedido de Classe 0 (zero) e na resposta o lado **Escravo** envia o valor corrente de todos os Objetos declarados
- **Classes 1 a 3:** São entidades que armazenam temporariamente listas de eventos ou mudanças nos Objetos. Cada Objeto precisa estar configurado no lado **Escravo** para gerar eventos quando há mudanças, e geralmente há um padrão entre os usuários de DNP para reservar a Classe 1 (um) para eventos digitais, a Classe 2 (dois) para eventos analógicos e a Classe 3 (três) para contadores

## Configuração do Driver

Este tópico contém informações sobre a configuração do Driver DNP 3.0 Slave.

## Configurações Extras

Este tópico contém informações sobre a Janela de Propriedades deste Driver, conforme os tópicos a seguir.

Além da Janela de Propriedades, as configurações deste Driver também podem ser definidas em tempo de execução nas aplicações **Elipse E3**, **Elipse Power** ou **Elipse Water**. Para isto, inicialize este Driver em modo **Offline**, ou seja, execute a aplicação com a opção **Start driver OFFLINE** habilitada, configurável na aba **Setup** da Janela de Propriedades. As opções de configuração deste Driver estão descritas na tabela a seguir.

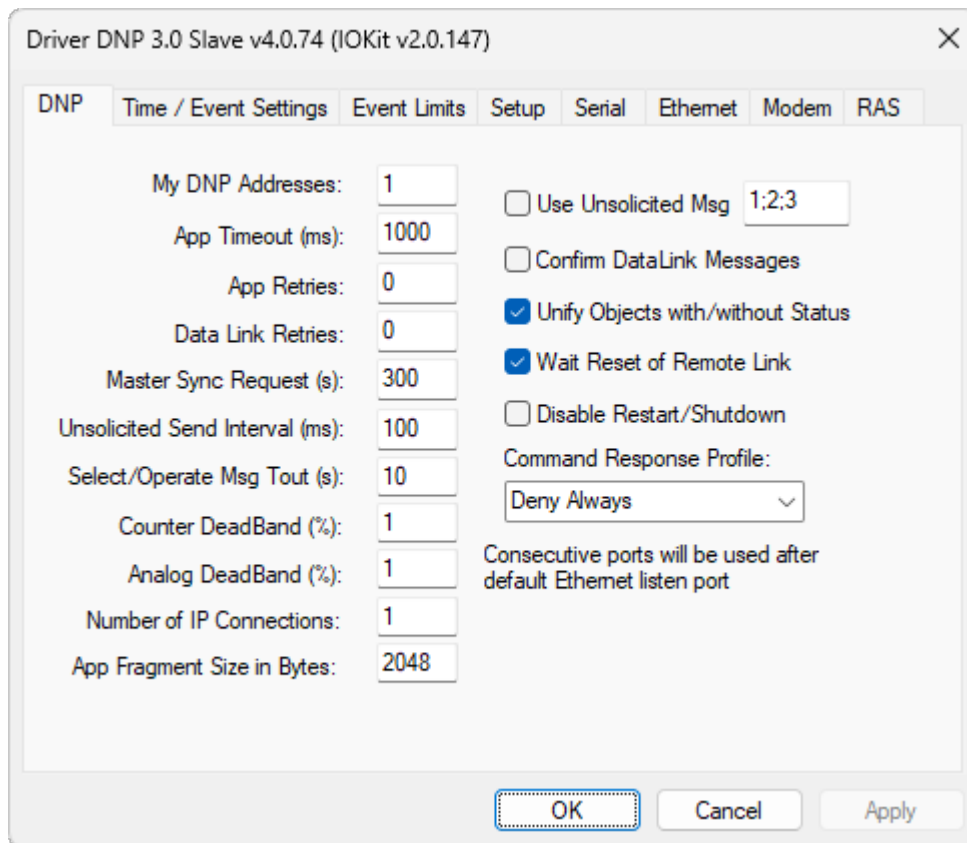
**Opções de configuração do Driver DNP 3.0 Slave**

PARÂMETRO	STRING OFFLINE	TIPO DE DADOS
<b>My DNP Addresses</b>	DNP.myaddress	Word
<b>App Retries</b>	DNP.App_retries	Word
<b>Data Link Retries</b>	DNP.DII_retries	Word
<b>Select / Operate Msg Tout (s)</b>	DNP.select_operate_tout	Word
<b>Counter DeadBand (%)</b>	DNP.counter_deadband	Float
<b>Analog DeadBand (%)</b>	DNP.analog_deadband	Float
<b>App Timeout (ms)</b>	DNP.App_timeout	Word
<b>Use Unsolicited Messages</b>	DNP.UseUnsol	Booleano

PARÂMETRO	STRING OFFLINE	TIPO DE DADOS
<b>Unsolicited Classes</b>	DNP.UnsolicitedClasses	String
<b>Unify Objects With/Without Status</b>	DNP.UnifyObjects	Booleano
<b>Confirm DataLink Messages</b>	DNP.ConfirmDLink	Booleano
<b>Master Sync Request (s)</b>	DNP.SyncTime	Word
<b>Unsolicited Send Interval (ms)</b>	DNP.SendInterval	Word
<b>Enable Time Sync</b>	DNP.EnableSync	Booleano
<b>Sync Events with Stby Driver</b>	DNP.SyncEvents	Booleano
<b>Command Response Profile</b>	DNP.CmdProfile	Word
<b>App Fragment Size in Bytes</b>	DNP.MaxAppSize	DWord
<b>Stop Analog Events if not connected</b>	DNP.StopAnalogEventsWhenDisconnected	Booleano
<b>Wait Reset of Remote Link</b>	DNP.WaitResetOfRemotelink	Booleano
<b>Sort Analog events by Index</b>	DNP.SortAnalogByIndex	Booleano
<b>Ignore Assign Class Requests</b>	DNP.IgnoreAssignClass	Booleano
<b>Limit Event Class Msg Size - Single Datalink fragment</b>	DNP.LimitEventMsg	Booleano
<b>Report Events in GMT Time</b>	DNP.ReportEventsInGMT	Booleano
<b>Enable Time Zone Offset on sync</b>	DNP.SyncTZOffset	Booleano
<b>One Analog Event per Index</b>	DNP.OneAnalogEvent	Booleano
<b>Use Callbacks for commands</b>	DNP.UseCallbacks	Booleano
<b>Class 1 max events</b>	DNP.MaxEventsClass1	DWord
<b>Class 2 max events</b>	DNP.MaxEventsClass2	DWord
<b>Class 3 max events</b>	DNP.MaxEventsClass3	DWord
<b>Disconnect after unconfirmed unsolicited sequence count</b>	DNP.MaxUnconfirmedUnsolicitedCount	Word
<b>Generate event on first write</b>	DNP.GenerateEventOnFirstWrite	Booleano
<b>Number of IP Connections</b>	DNP.IPConnections	Word
<b>Disable Restart / Shutdown</b>	DNP.DisableRestartShutdown	Booleano
<b>Add new timestamped events on first connection</b>	DNP.ReportEventsFirstConnection	Booleano
<b>Add new timestamped events on reconnection</b>	DNP.RepostEventsReconnection	Booleano
<b>Allow multi app fragment Class 2,3 - No Confirm</b>	DNP.MultiFragAppClass2_3_No_Confirm	Booleano

Estas propriedades podem ser escritas através de um Tag com o parâmetro *N1* igual a -1 (menos um), o parâmetro *N2* igual a 0 (zero), o parâmetro *N3* igual a 0 (zero) e o parâmetro *N4* igual a 3 (três).

## Aba DNP



Aba DNP

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

### Opções disponíveis na aba DNP

OPÇÃO	DESCRIÇÃO
<b>My DNP Addresses</b>	Informa o endereço DNP deste Driver. É possível informar mais de um endereço DNP através do uso de uma lista numérica separada por ponto e vírgula, como por exemplo "1;3;5". O primeiro endereço da lista é considerado o endereço DNP principal ou padrão
<b>App Timeout (ms)</b>	Tempo máximo que a camada de Aplicação espera por uma resposta completa da camada <i>Data Link</i> . Se o recebimento de um pedido está em andamento pela camada <i>Data Link</i> , este tempo é estendido automaticamente até o término da recepção pela camada <i>Data Link</i> . O valor padrão desta opção depende da taxa de comunicação utilizada. O <i>time-out</i> de <i>Data Link</i> , ou byte a byte, é definido pelo <i>time-out</i> da biblioteca <b>IOKit</b> , configurado na aba <b>Setup</b> . O valor desta opção deve ser maior ou igual ao <i>time-out</i> de <i>Data Link</i>
<b>App Retries</b>	Número de retentativas de comunicação realizadas pela camada de Aplicação em caso de erro de transação. O valor padrão desta opção é 0 (zero)
<b>Data Link Retries</b>	Número de retentativas de comunicação realizadas pela camada <i>Data Link</i> em caso de erro de transação. O valor padrão desta opção é 0 (zero)

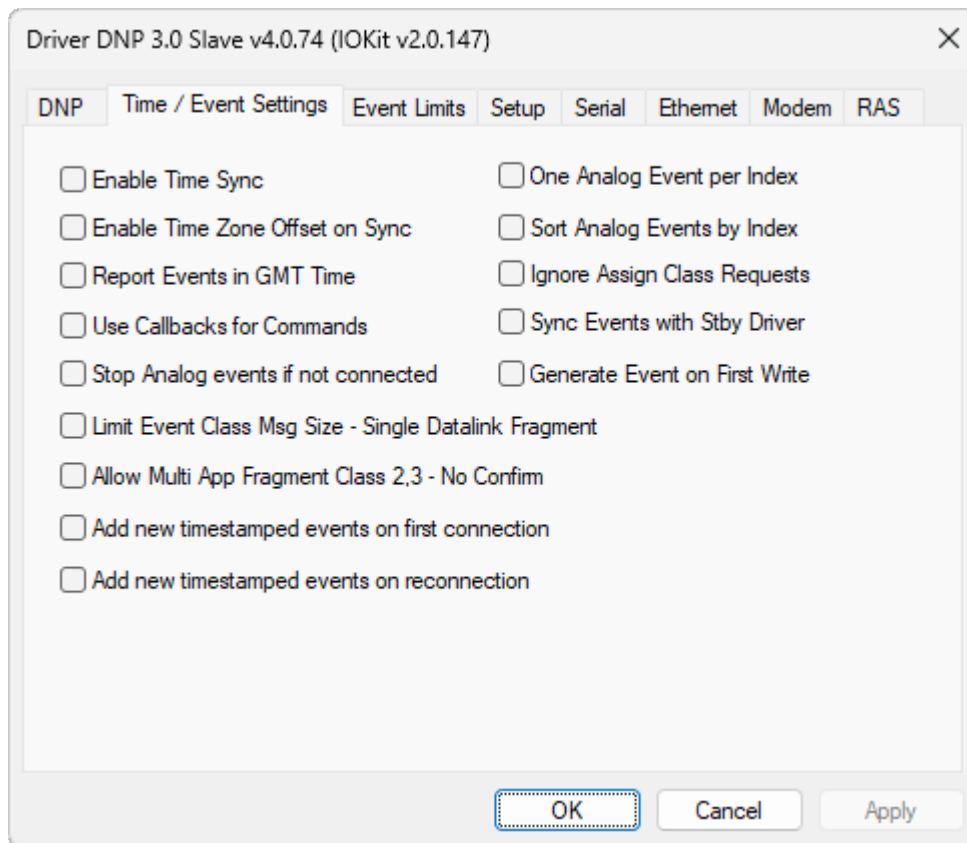
OPÇÃO	DESCRIÇÃO
<b>Master Sync Request (s)</b>	Intervalo de tempo em que este Driver solicita ao <b>Mestre</b> uma sincronização de relógio, que pode ou não ser aceita dependendo da configuração da opção <b>Enable Time Sync</b> na aba <b>Time / Event Settings</b> . Para desabilitar o envio, configure esta opção com o valor 0 (zero)
<b>Unsolicited Send Interval (ms)</b>	Define o intervalo de tempo em que este Driver verifica a existência de eventos das Classes 1 (um), 2 (dois) ou 3 (três) disponíveis para serem enviados através de mensagens não solicitadas, caso a opção <b>Use Unsolicited Messages</b> esteja habilitada. O lado <b>Mestre</b> também pode habilitar ou desabilitar o envio de mensagens não solicitadas por parte do <b>Escravo</b> através do envio das funções <b>Enable Unsolicited Messages (20)</b> e <b>Disable Unsolicited Messages (21)</b>
<b>Select / Operate Msg Tout (s)</b>	Tempo máximo, em segundos, entre um comando <b>Select</b> e um comando <b>Operate</b> . Ultrapassado este tempo, o comando <b>Operate</b> não é mais aceito por este Driver
<b>Counter DeadBand (%)</b>	Informa a banda morta, em porcentagem, para efeitos de notificação de eventos para os contadores
<b>Analog DeadBand (%)</b>	Informa a banda morta, em porcentagem, para efeitos de notificação de eventos para os pontos analógicos. Para especificar bandas mortas individuais, verifique as configurações de banda morta por Tag no tópico <b>Endereçamento de Tags</b>
<b>Number of IP Connections</b>	Define o número de conexões simultâneas, TCP ou UDP, que são aceitas a partir da porta TCP/IP definida na aba <b>Ethernet</b> . Por exemplo, ao definir 3 (três) conexões e a porta TCP/IP 20000, então este Driver está apto a receber 3 (três) conexões nas portas TCP/IP 20000, 20001 e 20002. Pode ser definido um valor entre 1 (uma) e 5 (cinco) conexões
<b>App Fragment Size in Bytes</b>	Informa o tamanho máximo de um fragmento de uma aplicação. O valor padrão para a maioria dos sistemas é 2048 bytes
<b>Use Unsolicited Messages</b>	Informa se este Driver envia mensagens não solicitadas
<b>Confirm DataLink Messages</b>	Informa se este Driver solicita mensagens de confirmação na camada de <i>Data Link</i>
<b>Unify Objects With/Without Status</b>	Se esta opção está selecionada, então as diferentes Variações de um Objeto são unificadas em uma única Variação com status. Por exemplo, se uma aplicação possui Tags digitais do tipo Objeto 1 Variação 1 e também Variação 2, todos os índices são unificados no mesmo Objeto com Variação 2. Se esta opção está desmarcada, então as diferentes Variações de um Objeto são mantidas como Objetos independentes. Por exemplo, se uma aplicação possui um Tag do tipo Objeto 1 Variação 1 e Índice 25, e outro Tag do tipo Objeto 1 Variação 2 e Índice 26, estes são reportados na integridade como Objetos diferentes. Não é recomendável que Objetos de Variações diferentes possuam os mesmos Índices, pois ambos podem gerar eventos, o que pode

OPÇÃO	DESCRIÇÃO
	causar confusão na interpretação. Esta opção atualmente vale para os Objetos 101, 301, 1001, 1101 e 3004
<b>Wait Reset of Remote Link</b>	Instrui este Driver a só responder comunicações após ter recebido o comando <b>Reset of Remote Link</b> vindo de um <b>Mestre</b> . O valor padrão desta opção é selecionada
<b>Disable Restart / Shutdown</b>	Desabilita o comando de desligar o computador, que pode ser enviado pelo <b>Mestre</b> a partir da função <b>13</b> do DNP ( <i>Cold Restart</i> )
<b>Command Response Profile</b>	<p>Informa o tipo de tratamento que é dado aos comandos solicitados a este Driver (Objeto 12 Variação 1 e Objeto 41 Variações 1, 2, 3 e 4). As opções disponíveis são <b>Deny Always</b>: Todos os comandos são respondidos negativa e instantaneamente, com o campo <b>Status</b> do comando reportando o código 4 (quatro, comando não suportado), <b>Accept Always</b>: Todos os comandos são respondidos de forma positiva, com o campo <b>Status</b> do comando reportando o código 0 (zero, comando OK), após uma aplicação ler o Tag de comando respectivo. Isto serve para indicar que o comando solicitado foi entendido e vai ser processado pela aplicação e <b>Wait for Application Response</b>: Os comandos só são respondidos depois que uma aplicação ler e escrever o Tag respectivo. A leitura serve para que uma aplicação entenda que houve uma solicitação por parte do <b>Mestre</b>, que é repassado para outro Driver ou saída do sistema. Após este processamento, uma aplicação deve escrever de volta no Tag o mesmo valor lido para indicar que o processamento foi bem-sucedido, ou um valor diferente caso não tenha obtido sucesso. Para mais informações, consulte o tópico <b>Tratamento de Comandos</b></p>

**NOTA**

É necessário um script para enviar a mensagem de um Driver para outro. Para mais informações, consulte o tópico **Sincronização de Eventos com Redundância**.

## Aba Time / Event Settings



**Aba Time / Event Settings**

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

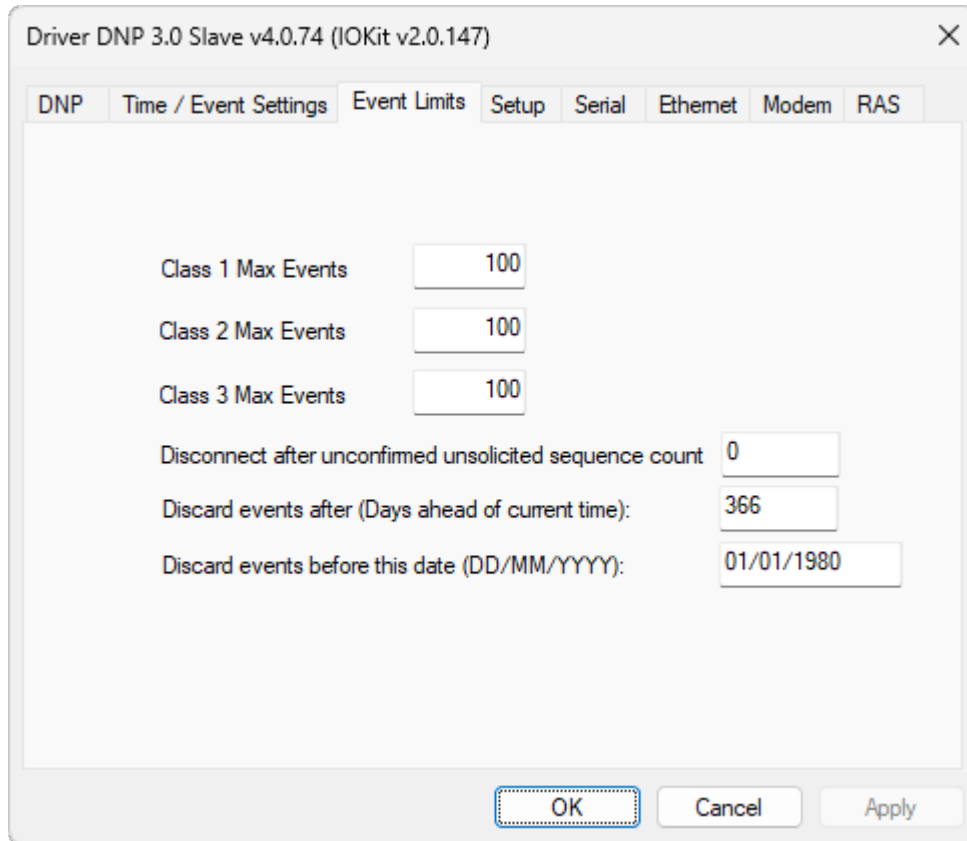
### Opções disponíveis na aba Time / Event Settings

OPÇÃO	DESCRIÇÃO
<b>Enable Time Sync</b>	Informa se este Driver aceita pedidos de sincronização de relógio, que por sua vez sincronizam o relógio do Windows
<b>Enable Time Zone Offset on Sync</b>	Aplica o fuso horário local no horário a ser sincronizado
<b>Report Events in GMT Time</b>	Modifica a estampa de tempo dos Tags, transformando-os para o horário GMT ( <i>Greenwich Mean Time</i> ) com base nas configurações regionais e horário de verão do Windows
<b>Use Callbacks for Commands</b>	Notifica os Tags de uma aplicação imediatamente quando comandos são recebidos de um <b>Mestre</b> . Caso esta opção não esteja selecionada, a notificação é baseada no <i>scan</i> de cada Tag de comando
<b>Generate Event on First Write</b>	Indica se a primeira escrita de um Tag deve gerar ou não um evento. Se esta opção não está selecionada, a primeira escrita apenas cria o ponto na base de dados interna deste Driver, e o valor é usado para comparação com a segunda escrita em diante, verificando se houve ou não um evento ou mudança de valor ou qualidade. Se esta opção está selecionada, a primeira escrita cria o ponto na base de dados interna deste Driver e gera um evento

OPÇÃO	DESCRIÇÃO
<b>Limit Event Class Msg Size - Single Datalink Fragment</b>	Limita as mensagens de eventos a um fragmento de <i>Data Link</i> por vez
<b>Sync Events with Stby Driver</b>	Na existência de um segundo <b>Escravo</b> na mesma aplicação que esteja operando de forma redundante, ou seja, o lado <b>Mestre</b> seleciona por qual canal é realizada a comunicação, esta opção instrui este Driver para que a cada mensagem de confirmação de eventos de Classes recebida, gere uma mensagem correspondente para ser enviada ao Driver redundante
<b>Stop Analog events if not connected</b>	Para de acumular eventos analógicos quando a conexão com um <b>Mestre</b> é interrompida. O valor corrente de cada Tag analógico continua sendo armazenado. Aplica-se aos Objetos <b>32</b> e <b>33</b>
<b>One Analog Event per Index</b>	Envia apenas um evento por índice a cada comunicação, de forma a tornar compatível a comunicação com implementações de <b>Mestres</b> que suportam apenas um evento por vez
<b>Sort Analog Events by Index</b>	Instrui este Driver a ordenar eventos analógicos (Objetos 3X com qualquer Variação) pelo Índice e não pela estampa de tempo, conforme é realizado por padrão para os demais eventos. Isto serve para permitir que eventos do tipo <b>Sag/Swell</b> de um ponto sejam reportados em conjunto, tornando possível para o lado <b>Mestre</b> compreender que se trata de um evento deste tipo
<b>Ignore Assign Class Requests</b>	Instrui este Driver a ignorar os pedidos de <i>Assign Class</i> recebidos. Um pedido de <i>Assign Class</i> é uma função do DNP que permite associar determinados Objetos a uma Classe de eventos 1 (um), 2 (dois) ou 3 (três) em tempo de execução
<b>Allow Multi App Fragment Class 2,3 - No Confirm</b>	Informa este Driver para enviar eventos pendentes de Classes 2 (dois) ou 3 (três) em múltiplos fragmentos de uma aplicação sem confirmação, em resposta a um pedido de leitura de Classes 2 (dois) ou 3 (três), caso o <i>buffer</i> de resposta não contenha nenhum evento de Classe 1 (um). Se contém, então a resposta é limitada a apenas 1 (um) fragmento de aplicação com confirmação
<b>Add new timestamped event on First Connection</b>	Informa que, quando ocorrer a primeira conexão de um <b>Mestre</b> , se este Driver <b>Escravo</b> deve gerar novos eventos com estampa de tempo a partir do valor corrente dos pontos. O objetivo é que o <b>Mestre</b> tenha a chance de receber os valores com estampa de tempo antes de realizar a Classe 0 (zero), que por definição do protocolo DNP, ocorre com valores estáticos, ou correntes, sem estampa de tempo. Desta forma, ao se conectar, se o <b>Mestre</b> solicitar eventos antes da Classe 0 (zero), ou se as mensagens não solicitadas estão habilitadas, então o <b>Mestre</b> recebe os valores correntes com estampa de tempo, caso estejam configurados desta forma. Esta opção só é utilizada para os Tags que estão configurados com <b>SOETYPE</b> com estampa de tempo, ou seja, o parâmetro <i>N1</i>

OPÇÃO	DESCRIÇÃO
<b>Add new timestamped event on reconnection</b>	O mesmo que a opção anterior, porém quando há reconexões

## Aba Event Limits



**Aba Event Limits**

Configure nesta aba o número máximo de eventos que este Driver deve manter em memória para as Classes de eventos 1 (um), 2 (dois) e 3 (três) nas opções **Class 1 Max Events**, **Class 2 Max Events** e **Class 3 Max Events**, respectivamente. Caso o limite de eventos seja atingido, os eventos mais antigos são removidos da lista.

Deve ser informada também a quantidade máxima de mensagens não solicitadas que podem ficar sem confirmação. Após este limite, este Driver fecha a conexão e aguarda uma nova conexão do **Mestre**.

## Endereçamento de Tags

<b>N1</b>	SOETYPE × 10 + Classe
<b>N2</b>	Código de Função. Para mais informações, consulte o tópico <b>Códigos de Funções Suportadas</b>
<b>N3</b>	Código de Objeto e Variação. Para mais informações, consulte o tópico <b>Objetos Suportados</b>
<b>N4</b>	Endereço da variável ou número
<b>Device</b>	Parâmetro opcional que informa o endereço lógico DNP, informado no parâmetro <b>My DNP Addresses</b> . Se não é informado, um Tag é endereçado para o primeiro endereço da lista, ou endereço DNP principal
<b>Item</b>	Parâmetro opcional que informa a banda morta individual de um Tag

O parâmetro *N1* contém a configuração de Classes e Eventos, de acordo com a fórmula **SOETYPE × 10 + Classe**. **SOETYPE** pode ser **0** (zero, sem SOE), **1** (um, COS ou *Change of State* sem estampa de tempo) ou **2** (dois, SOE ou *Sequence of Events* com estampa de tempo). **Classe** pode ser **1** (um), **2** (dois) ou **3** (três). Os casos especiais para o valor do parâmetro *N1* são os seguintes:

- Se o parâmetro *N1* é igual a 999, este Tag representa um Tag **On/Off** do protocolo, cujo valor padrão é 1 (um). Ao escrever o valor 1 (um), desliga toda e qualquer comunicação, retornando na escrita o valor 1 (um)
- Se o parâmetro *N2* é menor do que 0 (zero, Funções especiais), o parâmetro *N1* é usado para identificar a conexão, variando entre 1 (um) e 5 (cinco)

O parâmetro *N3* deve ser informado de acordo com a fórmula **ObjectCode × 100 + Variation**. **ObjectCode** é o tipo de Objeto, como por exemplo Entradas Binárias, e **Variation** é o sub-tipo.

O parâmetro *N4* é o endereço da variável ou número, independente de ser um ponto físico ou lógico.

Caso seja necessário informar uma banda morta individual por Tag, é possível usar o parâmetro *Item* para informar uma banda morta, com as seguintes opções:

- **Banda morta em porcentagem:** Este parâmetro deve estar no formato **DB:X%**, em que *X* é um valor decimal. Por exemplo, o valor **DB:3%** corresponde a 3 (três) por cento e o valor **DB:0,6%** corresponde a 0,6 por cento. Alternativamente, é possível especificar uma banda morta relativa, utilizando no parâmetro *Item* o formato **DBR:X%**, em que *X* é um valor decimal. Desta forma, as solicitações de valor instantâneo, como por exemplo a Classe 0 (zero), reportam sempre um valor atualizado, enquanto os eventos só são gerados se há uma variação acima da banda morta com relação ao último evento reportado
- **Banda morta em valores absolutos:** Este parâmetro deve estar no formato **DB:XA**, em que *X* é um valor decimal. Por exemplo, o valor **DB:3A** corresponde a 3 (três) unidades na escala da própria variável. Da mesma forma que na opção anterior, é possível especificar uma banda morta relativa, utilizando no parâmetro *Item* o formato **DBR:XA**, em que *X* é um valor decimal
- **Banda morta BitString:** Também é possível especificar uma banda morta especial para Tags analógicos inteiros que representam um conjunto de bits, como por exemplo **Word** ou **DWord**, em que cada bit é um estado digital. Para isto, configure o parâmetro *Item* no formato **DB:BSTR**. Desta forma, caso ocorra uma mudança em qualquer bit deste Tag, um evento é gerado, independente das opções **One Analog Event per Index** e **Stop Analog if not Connected**, permitindo que alterações neste Tag não sejam perdidas, independente da configuração dos demais Tags analógicos

## Códigos de Funções Suportadas

### Códigos de funções suportadas

N2	LEITURA OU ESCRITA	OPERAÇÃO
-13	Leitura	Estatísticas de comunicação por canal. Os valores possíveis para o parâmetro <i>N4</i> são <b>1</b> : Frames enviados, <b>2</b> : Frames sem resposta, <b>3</b> : Erros de CRC ( <i>Cyclic Redundancy Check</i> ) no formato da resposta, <b>4</b> : Erros de envio ou <b>5</b> : Retentativas de transmissão
-20	Leitura e escrita	Comunicação ligada ou desligada por canal. Os valores possíveis são <b>0</b> (zero, desligada) ou <b>1</b> (um, ligada)

N2	LEITURA OU ESCRITA	OPERAÇÃO
-21	Leitura	Comunicação inativa por canal. Os valores possíveis são <b>1</b> (um, inativa) ou <b>0</b> (zero, ativa)
-22	Leitura	Leitura de eventos quitados para serem enviados a um Driver redundante, por canal
-23	Escrita	Escrita de eventos para serem quitados neste Driver, por canal
-42	Escrita	Força a geração de eventos baseado no valor corrente de todos os Objetos do mesmo tipo declarado no parâmetro N3, $Object \times 100 + variation$ . O evento considera a configuração definida no parâmetro N1, $SOETYPE \times 10 + Class$
1	Escrita	Read
2	Leitura e escrita	Write
3	Leitura e escrita	Select
4	Leitura e escrita	Operate
5	Leitura e escrita	Direct Operate
13	-	Cold Restart
14	-	Warm Restart
20	-	Enable Unsolicited Messages
21	-	Disable Unsolicited Messages
22	-	Assign Classes
23	-	Delay Measurement
24	-	Record Current Time

## Objetos Suportados

### Objetos suportados

OBJETO	TIPO	VARIAÇÃO	NOME	CÓDIGO DE FUNÇÃO
1†	Estático	0 (zero)	Retorna Objeto 1 (um) Variação 2 (dois)	6 (seis)
1†	Estático	2 (dois)	Binary Input with Status	1 (um)
2‡	Evento	1 (um)	Binary Input Change without Time	1 (um)
2‡	Evento	2 (dois)	Binary Input Change with Time	1 (um)
2‡	Evento	3 (três)	Binary Input Change with Relative Time	1 (um)

OBJETO	TIPO	VARIAÇÃO	NOME	CÓDIGO DE FUNÇÃO
3†	Estático	2 (dois)	Double Bit Binary Input with Status	1 (um)
4#	Evento	1 (um)	Double Bit Binary Input Change without Time	1 (um)
4#	Evento	2 (dois)	Double Bit Binary Input Change with Time	1 (um)
10†	Estático	1 (um)	Binary Output	1 (um)
10†	Estático	2 (dois)	Binary Output Status	1 (um)
12	--	1 (um)	Control Relay Output Block	3 (três), 4 (quatro), 5 (cinco), 6 (seis)
20†	Estático	2 (dois)	Retorna Objeto 20 Variações 1 (um) e 2 (dois)	6 (seis)
20†	Estático	1 (um)	32-bit Counter	1 (um), 7 (sete), 9 (nove), 10, 11
20†	Estático	2 (dois)	16-bit Binary Counter	1 (um), 7 (sete), 9 (nove), 10, 11
20†	Estático	3 (três)	32-bit Delta Counter	1 (um), 7 (sete), 9 (nove), 10, 11
20†	Estático	4 (quatro)	16-bit Delta Counter	1 (um), 7 (sete), 9 (nove), 10, 11
20†	Estático	5 (cinco)	32-bit Counter without Flag	1 (um), 7 (sete), 9 (nove), 10, 11
20†	Estático	6 (seis)	16-bit Counter without Flag	1 (um), 7 (sete), 9 (nove), 10, 11
20†	Estático	7 (sete)	32-bit Delta Counter without Flag	1 (um), 7 (sete), 9 (nove), 10, 11
20†	Estático	8 (oito)	16-bit Delta Counter without Flag	1 (um), 7 (sete), 9 (nove), 10, 11
21†	Estático	0 (zero)	Retorna Objeto 21 Variações 1 (um) e 2 (dois)	6 (seis)
21†	Estático	1 (um)	32-bit Frozen Counter	1 (um), 7 (sete), 9 (nove), 10, 11
21†	Estático	2 (dois)	16-bit Frozen Counter	1 (um), 7 (sete), 9 (nove), 10, 11
21†	Estático	3 (três)	32-bit Frozen Delta Counter	1 (um), 7 (sete), 9 (nove), 10, 11
21†	Estático	4 (quatro)	16-bit Frozen Delta Counter	1 (um), 7 (sete), 9 (nove), 10, 11
21†	Estático	5 (cinco)	32-bit Frozen Counter with Time Of Freeze	1 (um), 7 (sete), 9 (nove), 10, 11
21†	Estático	6 (seis)	16-bit Frozen Counter with Time Of Freeze	1 (um), 7 (sete), 9 (nove), 10, 11

OBJETO	TIPO	VARIAÇÃO	NOME	CÓDIGO DE FUNÇÃO
21†	Estático	7 (sete)	32-bit Frozen Delta Counter with Time Of Freeze	1 (um), 7 (sete), 9 (nove), 10, 11
21†	Estático	8 (oito)	16-bit Frozen Delta Counter with Time Of Freeze	1 (um), 7 (sete), 9 (nove), 10, 11
21†	Estático	9 (nove)	32-bit Frozen Counter without Flag	1 (um), 7 (sete), 9 (nove), 10, 11
21†	Estático	10	16-bit Frozen Counter without Flag	1 (um), 7 (sete), 9 (nove), 10, 11
21†	Estático	11	32-bit Frozen Delta Counter without Flag	1 (um), 7 (sete), 9 (nove), 10, 11
21†	Estático	12	16-bit Frozen Delta Counter without Flag	1 (um), 7 (sete), 9 (nove), 10, 11
22‡	Evento	1 (um)	32-bit Counter Change Event without Time	1 (um)
22‡	Evento	2 (dois)	16-bit Counter Change Event without Time	1 (um)
22‡	Evento	3 (três)	32-bit Delta Counter Change Event without Time	1 (um)
22‡	Evento	4 (quatro)	16-bit Delta Counter Change Event without Time	1 (um)
22‡	Evento	5 (cinco)	32-bit Counter Change Event with Time	1 (um)
22‡	Evento	6 (seis)	16-bit Counter Change Event with Time	1 (um)
22‡	Evento	7 (sete)	32-bit Delta Counter Change Event with Time	1 (um)
22‡	Evento	8 (oito)	16-bit Delta Counter Change Event with Time	1 (um)
23‡	Evento	1 (um)	32-bit Counter Change Event without Time	1 (um)
23‡	Evento	2 (dois)	16-bit Frozen Counter Event without Time	1 (um)
23‡	Evento	3 (três)	32-bit Frozen Delta Counter Event without Time	1 (um)
23‡	Evento	4 (quatro)	16-bit Frozen Delta Counter without Time	1 (um)
23‡	Evento	5 (cinco)	32-bit Frozen Counter Event with Time	1 (um)

OBJETO	TIPO	VARIAÇÃO	NOME	CÓDIGO DE FUNÇÃO
23#	Evento	6 (seis)	16-bit Frozen Counter Event with Time	1 (um)
23#	Evento	7 (sete)	32-bit Frozen Delta Counter Event with Time	1 (um)
23#	Evento	8 (oito)	16-bit Frozen Delta Counter Event with Time	1 (um)
30+	Estático	0 (zero)	Retorna Objeto 30 Variações 1 (um), 2 (dois) e 5 (cinco)	6 (seis)
30+	Estático	1 (um)	32-bit Analog Input	1 (um)
30+	Estático	2 (dois)	16-bit Analog Input	1 (um)
30+	Estático	3 (três)	32-bit Analog Input without Flag	1 (um)
30+	Estático	4 (quatro)	16-bit Analog Input without Flag	1 (um)
30+	Estático	5 (cinco)	32-bit Analog Input Floating Point	1 (um)
30+	Estático	6 (seis)	64-bit Analog Input Double Floating Point	1 (um)
31+	Estático	0 (zero)	Retorna Objeto 31 Variações 1 (um), 2 (dois) e 5 (cinco)	6 (seis)
31+	Estático	1 (um)	32-bit Frozen Analog Input	1 (um)
31+	Estático	2 (dois)	16-bit Frozen Analog Input	1 (um)
31+	Estático	3 (três)	32-bit Frozen Analog Input with Time Of Freeze	1 (um)
31+	Estático	4 (quatro)	16-bit Frozen Analog Input with Time Of Freeze	1 (um)
31+	Estático	5 (cinco)	32-bit Frozen Analog Input without Flag	1 (um)
31+	Estático	6 (seis)	16-bit Frozen Analog Input without Flag	1 (um)
31+	Estático	7 (sete)	32-bit Frozen Analog Input Floating Point	1 (um)
32#	Evento	1 (um)	32-bit Change Event without Time	1 (um)
32#	Evento	2 (dois)	16-bit Change Event without Time	1 (um)

OBJETO	TIPO	VARIAÇÃO	NOME	CÓDIGO DE FUNÇÃO
32#	Evento	3 (três)	32-bit Analog Change with Time	1 (um)
32#	Evento	4 (quatro)	16-bit Analog Change Event with Time	1 (um)
32#	Evento	5 (cinco)	32-bit Analog Change Floating Point without Time	1 (um)
32#	Evento	6 (seis)	64-bit Analog Change Double Floating Point without Time	1 (um)
32#	Evento	7 (sete)	32-bit Analog Change Floating Point with Time	1 (um)
33#	Evento	1 (um)	32-bit Frozen Analog Event without Time	1 (um)
33#	Evento	2 (dois)	16-bit Frozen Analog Event without Time	1 (um)
33#	Evento	3 (três)	32-bit Frozen Analog Event with Time	1 (um)
33#	Evento	4 (quatro)	16-bit Frozen Analog Event with Time	1 (um)
33#	Evento	5 (cinco)	32-bit Frozen Analog Floating Point without Time	1 (um)
33#	Evento	7 (sete)	32-bit Frozen Analog Floating Point with Time	1 (um)
34+	Estático	1 (um)	16-bit Analog Input Dead Band	1 (um)
34+	Estático	2 (dois)	32-bit Analog Input Dead Band	1 (um)
34+	Estático	3 (três)	32-bit Analog Input Floating Point Dead Band	1 (um)
40+	Estático	1 (um)	32-bit Analog Output Status	1 (um)
40+	Estático	2 (dois)	16-bit Analog Output Status	1 (um)
40+	Estático	3 (três)	32-bit Analog Output Status Floating Point	1 (um)
41	--	1 (um)	32-bit Analog Output Block	2 (dois), 3 (três), 4 (quatro), 5 (cinco), 6 (seis)
41	--	2 (dois)	16-bit Analog Output Block	2 (dois), 3 (três), 4 (quatro), 5 (cinco), 6 (seis)

OBJETO	TIPO	VARIAÇÃO	NOME	CÓDIGO DE FUNÇÃO
41	--	3 (três)	32-bit Floating Point Analog Output Block	2 (dois), 3 (três), 4 (quatro), 5 (cinco), 6 (seis)
41	--	4 (quatro)	64-bit Double Analog Output Block	2 (dois), 3 (três), 4 (quatro), 5 (cinco), 6 (seis)
50†	Estático	1 (um)	Time and Date	1 (um), 2 (dois)
50	--	3 (três)	Last Recorded Time	2 (dois)
51†	Estático	1 (um)	Time and Date CTO ( <i>Common Time of Occurrence</i> )	1 (um)
51†	Estático	2 (dois)	Unsynchronized Time and CTO	1 (um)
52†	Estático	2 (dois)	Time Delay Fine	1 (um)
60‡	Evento	1 (um)	Class 0 Data	1 (um)
60‡	Evento	2 (dois)	Class 1 (um) Data	1 (um)
60‡	Evento	3 (três)	Class 2 (dois) Data	1 (um)
60‡	Evento	4 (quatro)	Class 3 (três) Data	1 (um)
80	--	1 (um)	Internal Indications	1 (um), 2 (dois)

## Qualificadores Suportados

### Qualificadores suportados

OBJETO OU VARIAÇÃO	QUALIFICADOR DO PEDIDO	QUALIFICADOR DA RESPOSTA
<b>Objetos Estáticos em Geral (Leitura)</b>	0 (zero), 1 (um), 7 (sete), 8 (oito), 17, 18, 27, 28, 47, 58	Segue a pergunta
<b>Objeto 34 (Escrita)</b>	0 (zero), 1 (um), 7 (sete), 8 (oito), 17, 18, 27, 28, 47, 58	Segue a pergunta
<b>Objeto 60 (Classes) ou Pedidos Gerais de Integridade (Leitura)</b>	6 (seis), 7 (sete), 8 (oito)	17, 18, 27, 28, 47, 58
<b>Objeto 50 (Escrita)</b>	7 (sete), 2 (dois)	7 (sete), 2 (dois)
<b>Objeto 80 (Escrita)</b>	1 (um), 7 (sete)	7 (sete)
<b>Objetos 12 e 41 (Select e Operate, entre outros)</b>	0 (zero), 1 (um), 7 (sete), 8 (oito), 17, 18, 27, 28, 47, 58	Segue a pergunta

## Exemplos de Configuração de Tags

Supondo que um Driver DNP **Escravo** deseje reportar para um outro sistema, que implementa um Driver DNP **Mestre**, variáveis recebidas de um protocolo qualquer, os exemplos a seguir mostram os parâmetros *N1*, *N2*, *N3* e *N4* no formato **N1.N2.N3.N4**:

- Entrada Digital X gerando eventos com estampa de tempo na Classe 1 (um): 21.1.102.X
- Entrada Digital Y gerando eventos sem estampa de tempo na Classe 1 (um): 11.1.102.Y
- Entrada Analógica Z de 16 bits gerando eventos sem estampa de tempo na Classe 3 (três): 13.1.3002.Z

# Comentários Gerais

Este tópico fornece informações adicionais sobre o Driver DNP 3.0 Slave.

## Funcionamento

Este Driver deve ser usado para operar em modo **Escravo**, enviando os dados para algum Driver DNP em modo **Mestre** via Serial, Modem ou Ethernet TCP/IP.

Os dados que são enviados ao **Mestre** através de pedidos de leitura, geralmente comandos de leitura de variáveis estáticas ou de eventos, devem ser constantemente informados a este Driver através da escrita em Tags, em que cada Tag deve estar apontando para um ponto na base de dados.

Para efeitos de comunicação pelo protocolo, este Driver só reconhece a existência de um Objeto DNP quando o primeiro valor é escrito para um Objeto. A partir deste momento, o lado **Mestre** pode realizar solicitações para este Driver, o lado **Escravo**, que responde conforme os Objetos solicitados.

### NOTA

Os Objetos que geralmente são lidos pelo lado **Mestre**, através da Função 1 (um) de leitura, devem ter os Tags correspondentes cadastrados neste Driver **Escravo** com o parâmetro *N2* igual a 1 (um), ou seja, a escrita deste Tag para o Driver **Escravo** informa o valor atual para um Objeto. A partir desta escrita, este Driver verifica se o novo valor informado quando comparado ao valor anterior corresponde à geração de um evento, de acordo ainda com as configurações de cada Tag, o que é informado no parâmetro *N1*.

Por exemplo, o **Mestre** pergunta o valor do Objeto 1 (um), Variação 2 (dois) e Índice 100. Este ponto digital deve estar cadastrado na base de dados do **Escravo** através de um Tag com o parâmetro *N1* igual a 21, supondo que este ponto vai gerar um SOE e está associado à Classe 1 (um), o parâmetro *N2* igual a 1 (um), o parâmetro *N3* igual a 102 e o parâmetro *N4* igual a 100. Ao iniciar uma aplicação, este ponto não tem nenhum valor. Através da escrita de um valor, 0 (zero) ou 1 (um), neste Tag, o ponto é efetivamente criado como um ponto *online* na base de dados deste Driver.

### NOTA

Conclui-se que uma aplicação deve sempre receber os dados de algum lugar e atribuí-los aos Tags deste Driver para que os valores sejam atualizados.

Para os Tags declarados como pertencentes às Classes 1 (um), 2 (dois) ou 3 (três), com eventos com ou sem estampa de tempo, ao realizar a atribuição é automaticamente verificado se deve ser gerado um novo evento, de acordo com a banda morta no caso de contadores ou analógicos, colocando-se o novo evento disponível para a Classe.

Mesmo quando a propriedade **DeadBand** é declarada com o valor 0 (zero), é necessária uma mínima mudança de valor ou qualidade para que um novo evento seja gerado. Para que a banda morta não seja verificada, o valor 100 pode ser adicionado ao parâmetro *N1*, conforme os exemplos a seguir.

- Tag com o parâmetro *N1* igual a 22 (SOE, Classe 2), parâmetro *N2* igual a 1 (um), parâmetro *N3* igual a 3002 e parâmetro *N4* igual a 10 (16-bit Analog point 10, com SOE)
- Valor atual igual a 100, qualidade atual igual a 192 e estampa de tempo igual a 1/1/2012 12:00:00

Uma escrita neste Tag com o mesmo valor e qualidade mas com a estampa de tempo diferente, na configuração com o parâmetro *N1* igual a 22, não gera um novo evento. Mas caso o parâmetro *N1* seja modificado para 122, mesmo que o valor e qualidade sejam os mesmos, um novo evento é gerado na escrita. Isto é útil para representar eventos de *trip* sem retorno ou eventos de **Sag/Swell**.

## Bloco Único de Eventos

Pode-se optar por criar um único Tag Bloco para inserir todos os eventos neste Driver, evitando a criação de vários Tags individuais para cada informação. Este Bloco tem a seguinte configuração:

- **N1:** 50 (valor fixo)
- **N2, N3, N4:** Não utilizados
- **Tamanho:** Deve possuir 7 (sete) Elementos:
  - **Elemento 0:** Parâmetro *N1* contém a informação  $SOETYPE \times 10 + Classe$ . Para mais informações, consulte o tópico **Endereçamento de Tags**
  - **Elemento 1:** Parâmetro *N2* contém o código de uma função
  - **Elemento 2:** Parâmetro *N3* contém a informação  $Objeto \times 100 + Variação$ . Para mais informações, consulte o tópico **Endereçamento de Tags**
  - **Elemento 3:** Parâmetro *N4* contém o índice de um ponto
  - **Elemento 4:** Valor de um ponto
  - **Elemento 5:** Estampa de tempo no formato de tempo nativo do sistema
  - **Elemento 6:** Qualidade

### NOTA

Um Bloco Único de Eventos não pode ser usado em conjunto com outros Tags de evento em uma aplicação, ocorrendo neste caso a duplicação de eventos.

## Comandos

O Objeto *Control Relay Output* (Objeto 12, Variação 1) define os campos de uma mensagem, descritos na tabela a seguir.

### Byte 0 (zero, Control Code)

Bit	7	6	5	4	3	2	1	0
Significado	Trip/Close		Clear	Queue	Code			

- **Trip/Close:** Este campo determina qual relé de controle é ativado em um sistema onde um par de relés *trip* e *close* é usado para energizar e desenergizar os pontos no campo. Os valor possíveis são, em formato binário, **00:** NUL, **01:** Close e **10:** Trip. O valor **NUL** pode ser usado para ativar o relé de seleção sem ativar os relés de *trip* ou *close*. Em um sistema sem relés de seleção, o valor **NUL** não realiza nenhuma operação. Já em um sistema sem relés *trip* ou *close*, este campo deve ser sempre igual a **NUL** para indicar uma operação normal de controle digital, onde o ponto exato de controle é implícito ou totalmente conhecido. Este campo não suporta os comandos de *trip* e *close* simultaneamente
- **Clear:** Se o comando possui este campo em 1 (um, ligado), todas as operações de controle são removidas da fila, incluindo o comando que está sendo executado, e esta operação de controle é realizada

- **Queue (Fila):** Indica a colocação do comando em uma fila de comandos em um equipamento. Se este campo é 0 (zero, **NUL**), então nenhuma operação é colocada na fila e a fila é limpa de todos os controles, incluindo o comando que está sendo executado se o campo **Clear** está ligado. Quando a função de controle é executada e completada, esta é removida da fila. Se o comando tem o atributo *Queue* ligado, então a operação é enfileirada novamente, ou seja, posicionada no fim da fila para aquele ponto
- **Code:** Este campo especifica o tipo de operação propriamente dito. Este comando pode ser usado com equipamentos que suportam enfileiramento de comandos (*queuing*), ponto a ponto ou outros mecanismos de controle. No primeiro tipo, qualquer comando de controle deve ser enfileirado para o ponto em questão. No segundo tipo, cada controle é realizado até ser completado antes do próximo comando ser aceito para aquele ponto

#### Valores possíveis para Code, em formato binário

VALOR	OPERAÇÃO	DESCRIÇÃO
0000	NUL	Nenhuma operação é realizada
0001	PULSE ON	O ponto, ou pontos, é ligado pelo tempo especificado em <b>On Time</b> , desligado pelo tempo especificado em <b>Off Time</b> e deixado no estado <b>OFF</b>
0010	PULSE OFF	O ponto, ou pontos, é desligado pelo tempo especificado em <b>Off Time</b> , ligado pelo tempo especificado em <b>On Time</b> e deixado no estado <b>ON</b>
0011	LATCH ON	Mantém o ponto ou pontos no estado <b>ON</b>
0100	LATCH OFF	Mantém o ponto ou pontos no estado <b>OFF</b>

#### NOTA

Valores fora da tabela anterior não são definidos.

#### Byte 1 (um, Count)

Este byte indica quantas vezes uma operação é executada. Este valor é mantido fixo em 1 (um) por este Driver.

#### Bytes 2 a 5 (On Time)

Tempo *on-time*, em milissegundos, definido na janela de configurações extras deste Driver. Este valor é fixo para todos os comandos.

#### Bytes 6 a 9 (Off Time)

Tempo *off-time*, em milissegundos, definido na janela de configurações extras deste Driver. Este valor é fixo para todos os comandos.

#### Byte 10 (Status)

Status da operação retornado por este Driver se a operação foi concluída com sucesso. O status só é interpretado na resposta e pode ser usado por uma aplicação para verificar se o comando foi executado com sucesso. Os códigos são os seguintes:

- **0:** Comando executado corretamente, incluindo operações de **Select** e **Operate**
- **1:** Comando **Operate** enviado após o tempo máximo do comando **Select** definido no **Escravo**
- **2:** Comando **Operate** enviado sem um comando **Select** prévio
- **3:** Erros de formatação na mensagem
- **4:** Operação não suportada para o ponto em questão
- **5:** Fila cheia ou ponto já está ativo
- **6:** Problemas de hardware
- **Outros:** Códigos de erro não padronizados

## Tratamento de Comandos

Este Driver suporta o recebimento de comandos **Select**, **Operate** ou **Direct Operate** para os Objetos **12 Variação 1 (um)** e **41 Variações 1 (um), 2 (dois), 3 (três) ou 4 (quatro)**.

Caso a opção **Command Response Profile** na aba **DNP** esteja definida com o valor **Wait for Application Response**, a sequência completa para tratar um comando é a seguinte:

1. O comando **Select**, **Operate** ou **Direct Operate** é recebido.
2. Um Tag PLC ou Tag Bloco com o parâmetro *N2* igual a 3 (três), 4 (quatro) ou 5 (cinco) deve ser lido e tratado por uma aplicação, como por exemplo repassar para outro Driver que vai reenviá-lo para um equipamento.
3. Ao receber a resposta deste comando ao equipamento, deve-se escrever de volta no mesmo Tag cujo comando foi lido o status. Se a resposta do comando ao equipamento foi positiva, deve escrever o mesmo valor de volta. Caso contrário, deve escrever um valor diferente.
4. Se nenhuma escrita é realizada no Tag no intervalo de 10 segundos, este Driver envia uma resposta com status igual a 6 (seis, *Request Not Accepted. Hardware Problems*).

Para receber uma solicitação de comando destes Objetos, pode-se utilizar um Tag Individual ou um Tag Bloco.

### Objeto 12 Variação 1 (um)

Para um Tag individual, o valor do Tag quando lido contém o valor do campo **Control Code**. Para o uso das operações **Select** e **Operate** devem ser criados dois Tags, e cada um recebe a leitura no momento que a operação correspondente, **Select** ou **Operate**, ocorrer. Para o uso da operação **Direct Operate** (cinco) só é necessário um único Tag com os parâmetros a seguir:

- **N1:** Não usado
- **N2:** 3 (três), 4 (quatro) ou 5 (cinco)
- **N3:** 1201 ou 41XX
- **N4:** Índice do ponto

Para um Tag Bloco, valem as mesmas configurações anteriores, porém este Tag Bloco pode ter até 6 (seis) Elementos. A diferença é que não importa a operação, **Select**, **Operate** ou **Direct Operate**, só é necessário um único Tag Bloco, pois a operação é retornada no Elemento 5 (cinco):

- **Elemento 0:** Control Code
- **Elemento 1:** Count
- **Elemento 2:** On Time
- **Elemento 3:** Off Time
- **Elemento 4:** Status
- **Elemento 5:** Operação (**3:** Select, **4:** Operate ou **5:** Direct Operate)
- **Elemento 6:** Endereço do **Mestre**

### Objeto 41 Variações 1 (um), 2 (dois), 3 (três) ou 4 (quatro)

Para um Tag individual, o valor do Tag quando lido contém o valor do campo **Control Code**. Para o uso das operações **Select** e **Operate** devem ser criados dois Tags, e cada um recebe a leitura no momento que a operação correspondente, **Select** ou **Operate**, ocorrer. Para o uso da operação **Direct Operate** (cinco) só é necessário um único Tag com os parâmetros a seguir:

- **N1:** Não usado
- **N2:** 5 (cinco)
- **N3:** 1201 ou 41XX
- **N4:** Índice do ponto

Para um Tag Bloco valem as mesmas configurações anteriores, porém este Tag Bloco pode ter até 3 (três) Elementos:

- **Elemento 0:** Valor
- **Elemento 1:** Status
- **Elemento 2:** Operação

Exemplo de script para tratar a recepção de comandos em um Objeto 12 Variação 1 (um).

```

// O Bloco AbreFecha é formado por 5 (cinco) Elementos
// O exemplo pressupõe o envio do comando recebido
// para um outro Driver DNP Mestre

Sub AbreFecha_OnRead()
  Set Digitais = Parent.Parent.Item("LeituraDigitais")
  ControlCode = Item("ControlCode").Value

  Trip = 1
  Close = 0
  CmdOk = 1

  Select Case ControlCode
    Case 65 'Pulse On Close
      VComando = Close
    Case 66 'Pulse Off Close
      VComando = Close
    Case 67 'Latch On Close
      VComando = Close
    Case 68 'Latch Off Close
      VComando = Close
    Case 129 'Pulse On Trip
      VComando = Trip
    Case 130 'Pulse Off Trip
      VComando = Trip
    Case 131 'Latch On Trip
      VComando = Trip
    Case 132 'Latch Off Trip
      VComando = Trip
    Case Else 'Comando inválido ou não formatado
      CmdOk = 0
  End Select

  If Not(CmdOk) Then
    WStatus = 7
  Else
    If Digitais.Item("ESTADODL01").WriteEx(VComando, , , WStatus) Then
      WStatus = 0 'Força o valor 0 (zero) ou sucesso na escrita
      'Status esperados pelo Mestre em caso de erro:
      Select Case WStatus
        Case 1
          Endtext = " Operate recebido depois do time-out de seleção"
        Case 2
          Endtext = " Sem mensagem de seleção anterior"
        Case 3
          Endtext = " Erro de formatação no comando"
        Case 4
          Endtext = " Operação não suportada para este ponto"
        Case 5
          Endtext = " Fila está cheia ou ponto já está ativo"
        Case 6
          Endtext = " Problemas de hardware"
        Case Else
          Endtext = " Problema não definido"
      End Select
    Else
      WStatus = 7
    End If
  End If
  Item("Status").Value = WStatus
  Write(EWriteSyncMode)
End Sub

```

## Informação de Qualidade

Para todos os Tags é utilizada a qualidade do Tag informada na escrita para efeito da propagação do status do ponto original.

Para isto é realizado um mapeamento simples. Se um Tag tem qualidade igual ou superior a 192, o status DNP do ponto correspondente tem o bit *Online* igual a 1 (um). Caso contrário, o bit *Online* tem o valor igual a 0 (zero), conforme o exemplo na tabela a seguir para os Objetos 1 (um), 2 (dois) e 10.

<b>Bit</b>	7	6	5	4	3	2	1	0
<b>Significado</b>	XX	XX	CF	LF	RF	CL	RS	OL

## Sincronização de Eventos com Redundância

Caso a opção **Sync Events with Stby Driver** na aba **Time / Event Settings** esteja habilitada, é necessário que cada Driver **Escravo** redundante possua 2 (dois) Tags Bloco para receber e enviar os eventos a serem quitados.

Através do Tag Bloco com o parâmetro *B2* igual a -22, é possível ler os eventos quitados em um Driver e com o Tag Bloco com o parâmetro *B2* igual a -23 é possível escrever os eventos a serem quitados em outro Driver.

Em ambos os casos o Tag Bloco deve possuir 6 (seis) Elementos, descritos a seguir:

- **Elemento 0:** Objeto × 100 + Variação
- **Elemento 1:** Índice
- **Elemento 2:** Classe (um, dois ou três)
- **Elemento 3:** Valor
- **Elemento 4:** Estampa de tempo válida (1: Válida ou 0: Inválida)
- **Elemento 5:** Estampa de tempo no formato de ponto flutuante de 64 bits

Exemplo de código para o **Elipse E3, Elipse Power** ou **Elipse Water**, supondo a existência de um **DriverSlave1** quitando eventos em **DriverSlave2** e vice-versa.

```
Sub SyncEvtsRead_OnRead() //Script em DriverSlave1
  Set EvtTag = Application.GetObject("DriverSlave2.SyncEvtsWrite")
  EvtTag.Item("Elemento1").Value = Item("Elemento1").Value
  EvtTag.Item("Elemento2").Value = Item("Elemento2").Value
  EvtTag.Item("Elemento3").Value = Item("Elemento3").Value
  EvtTag.Item("Elemento4").Value = Item("Elemento4").Value
  EvtTag.Item("Elemento5").Value = Item("Elemento5").Value
  EvtTag.Item("Elemento6").Value = Item("Elemento6").Value
  EvtTag.Write()
End Sub

Sub SyncEvtsRead_OnRead() //Script em DriverSlave2
  Set EvtTag = Application.GetObject("DriverSlave1.SyncEvtsWrite")
  EvtTag.Item("Elemento1").Value = Item("Elemento1").Value
  EvtTag.Item("Elemento2").Value = Item("Elemento2").Value
  EvtTag.Item("Elemento3").Value = Item("Elemento3").Value
  EvtTag.Item("Elemento4").Value = Item("Elemento4").Value
  EvtTag.Item("Elemento5").Value = Item("Elemento5").Value
  EvtTag.Item("Elemento6").Value = Item("Elemento6").Value
  EvtTag.Write()
End Sub
```

## Cold e Warm Restart

Na recepção do comando **Cold Restart**, após o envio da mensagem de resposta, este Driver reinicia o sistema operacional.

Na recepção do comando **Warm Restart**, após o envio da mensagem de resposta, este Driver limpa a fila de eventos aguardando para serem enviados ou quitados e se coloca em estado de inicialização, aguardando uma mensagem de **Reset of Remote Link**. Os valores correntes dos Tags na base de dados deste Driver permanecem em memória.

## Documentação das Interfaces de Comunicação

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

### 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

Setup

Physical Layer: Ethernet ▾
 Start driver OFFLINE

Timeout: 1000 ms
Communication check time: 5000 ms

Connection management

Mode: Automatic (managed by the driver) ▾

Retry failed connection every 20 seconds

Give up after 1 failed retries

Disconnect if non-responsive for 0 seconds

Logging Options

Log to File: C:\eeLogs\MicrolokII\_%DATE%.log

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

**Aba Setup**

**Opções gerais da aba Setup**

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

### Opções para o grupo Connection management

OPÇÃO	DESCRIÇÃO
<b>Mode</b>	Selecione o modo de gerenciamento de conexão. Selecionar a opção <b>Automatic</b> permite que um Driver gerencie a conexão automaticamente, como especificado nas opções seguintes. Selecionar a opção <b>Manual</b> permite que uma aplicação gerencie a conexão completamente
<b>Retry failed connection every ... seconds</b>	Selecione esta opção para habilitar a retentativa de conexão de um Driver em um determinado intervalo, em segundos. Se a opção <b>Give up after failed retries</b> não está selecionada, este Driver continua retentando até que a conexão seja efetuada, ou que a aplicação seja parada
<b>Give up after ... failed retries</b>	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 <b>Offline</b> , 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
<b>Disconnect if non-responsive for ... seconds</b>	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 <b>Timeout</b>

## Opções para o grupo Logging Options

OPÇÃO	DESCRIÇÃO
<b>Log to File</b>	<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. Caso se utilize a macro <b>%PROCESS%</b> 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 <b>Elipse E3</b>, 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 <b>0FDAh</b>. Pode-se também utilizar a macro <b>%DATE%</b> no nome do arquivo. Neste caso é gerado um arquivo de log por dia, no formato <b>aaaa_mm_dd</b>. 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 <b>%DATE_HOUR%</b> gera um arquivo de log por hora, no formato <b>aaaa_mm_dd_hh</b></p>
<b>File size limit (MB)</b>	<p>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</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):   $\mu$ s

Inter-frame delay (milliseconds):  ms

Aba Serial

## Opções gerais da aba Serial

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

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

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

#### Opções disponíveis no grupo Handshaking

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

## Aba Ethernet

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

Ethernet

Transport: TCP/IP ▾

PING before connecting

Timeout: 4000 ms

Retries: 1

Listen for connections on port: 0

Share listen port with other processes

Interface: (All Interfaces) ▾

Use IPv6  Use SSL SSL Settings

Enable 'ECHO' suppression

IP Filter:

Connect to

<input type="checkbox"/> Main IP: <span style="border: 1px solid gray; padding: 2px;"> </span>	Port: <span style="border: 1px solid gray; padding: 2px;">502</span>	<input type="checkbox"/> Local port: <span style="border: 1px solid gray; padding: 2px;">0</span>
<input type="checkbox"/> Backup IP 1: <span style="border: 1px solid gray; padding: 2px;"> </span>	Port: <span style="border: 1px solid gray; padding: 2px;">0</span>	<input type="checkbox"/> Local port: <span style="border: 1px solid gray; padding: 2px;">0</span>
<input type="checkbox"/> Backup IP 2: <span style="border: 1px solid gray; padding: 2px;"> </span>	Port: <span style="border: 1px solid gray; padding: 2px;">0</span>	<input type="checkbox"/> Local port: <span style="border: 1px solid gray; padding: 2px;">0</span>
<input type="checkbox"/> Backup IP 3: <span style="border: 1px solid gray; padding: 2px;"> </span>	Port: <span style="border: 1px solid gray; padding: 2px;">0</span>	<input type="checkbox"/> Local port: <span style="border: 1px solid gray; padding: 2px;">0</span>

**Aba Ethernet**

### Opções disponíveis na aba Ethernet

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

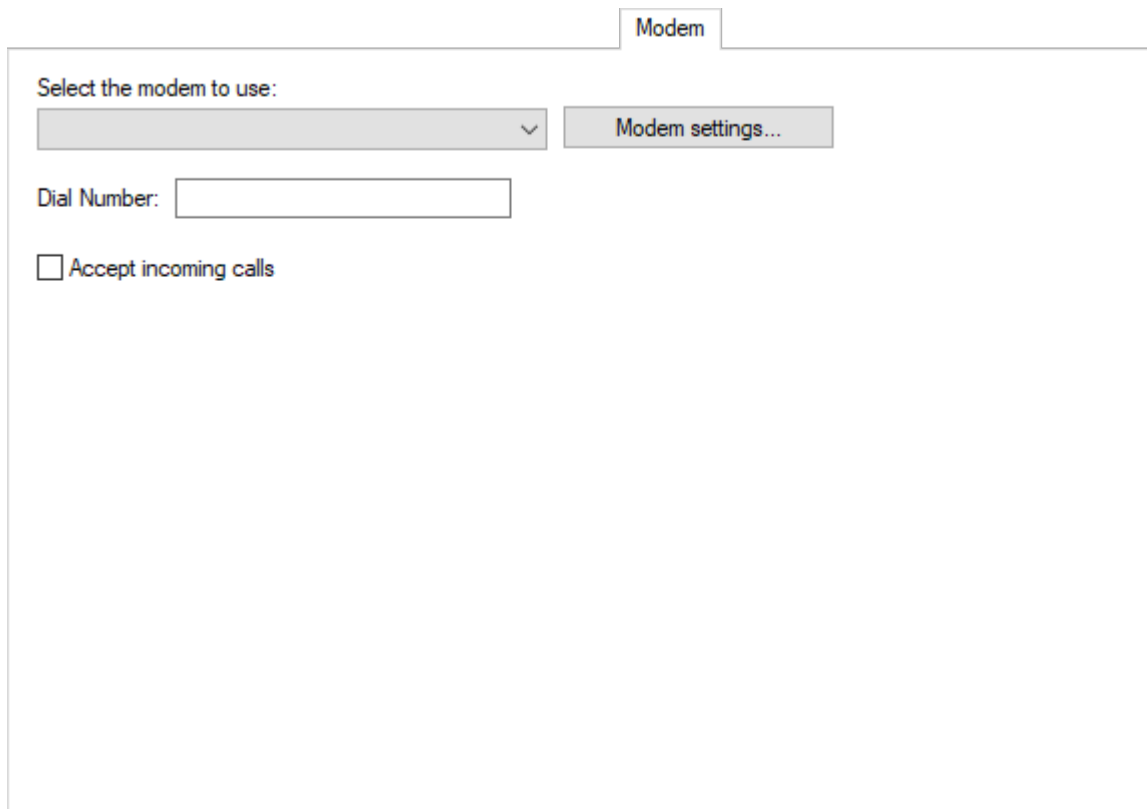
OPÇÃO	DESCRIÇÃO
<b>IP Filter</b>	Lista de endereços IP restringidos ou permitidos de onde um Driver aceita conexões ( <i>Firewall</i> ). Consulte a propriedade <b>IO.Ethernet.IPFilter</b> para mais informações
<b>PING before connecting</b>	Habilite esta opção para executar um comando <b>ping</b> , ou seja, para verificar se um dispositivo pode ser encontrado na rede, em um 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 um dispositivo. O <i>time-out</i> de uma conexão com um <i>socket</i> pode ser bem alto. As opções disponíveis são: <ul style="list-style-type: none"> <li>• <b>Timeout:</b> Especifique o número de milissegundos de espera por uma resposta de um comando <b>ping</b>. Deve-se usar um comando <b>ping</b> para verificar o tempo normal de resposta, configurando esta opção para um valor acima desta média. Normalmente pode-se configurar um valor entre 1000 e 4000 milissegundos, ou seja entre 1 (um) e 4 (quatro) segundos</li> <li>• <b>Retries:</b> Número de retentativas de um comando <b>ping</b>, sem contar a tentativa inicial. Se todas as tentativas falharem, então a conexão com o <i>socket</i> é abortada</li> </ul>

#### Opções disponíveis no grupo Connect to

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

## Aba Modem

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



**Aba Modem**

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

**Opções disponíveis na aba Modem**

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

**Aba RAS**

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

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

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

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

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

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

The image shows a screenshot of a software configuration 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, there is a text instruction: "Other socket settings should be configured in the 'Ethernet' tab!".

Aba RAS

## Opções disponíveis na aba RAS

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

## 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

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Leitura
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	0 (zero)
<b>Parâmetro N4</b>	6 (seis)
<b>Configuração por String</b>	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

<b>Tipo de Tag</b>	Tag Bloco
<b>Tipo de Acesso</b>	Somente Leitura
<b>Parâmetro B1</b>	-1 (menos um)
<b>Parâmetro B2</b>	0 (zero)
<b>Parâmetro B3</b>	0 (zero)
<b>Parâmetro B4</b>	1 (um)
<b>Propriedade Size</b>	4 (quatro)
<b>Propriedade ParamItem</b>	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

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Somente Leitura
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	0 (zero)
<b>Parâmetro N4</b>	2 (dois)
<b>Configuração por String</b>	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

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

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

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

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

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

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

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

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

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

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

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

## IO.WorkOnline

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Leitura ou Escrita
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	0 (zero)
<b>Parâmetro N4</b>	4 (quatro)
<b>Configuração por String</b>	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

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

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

#### IO.Stats.Partial.BytesSent

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

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

#### IO.Stats.Partial.TimeConnectedSeconds

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Somente Leitura
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	0 (zero)
<b>Parâmetro N4</b>	1102
<b>Configuração por String</b>	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

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Somente Leitura
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	0 (zero)
<b>Parâmetro N4</b>	1103
<b>Configuração por String</b>	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

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

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

## IO.Stats.Total.BytesSent

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

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

## IO.Stats.Total.ConnectionCount

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Somente Leitura
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	0 (zero)
<b>Parâmetro N4</b>	1004
<b>Configuração por String</b>	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

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

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

## IO.Stats.Total.TimeDisconnectedSeconds

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

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

## Propriedades

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

## Configuração da Interface Ethernet

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

### Tags de Comunicação

#### Tags da Interface Ethernet (N2/B2 = 4)

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

#### IMPORTANTE

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

### IO.Ethernet.IPSelect

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

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

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

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

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

### IO.Ethernet.IPSwitch

<b>Tipo de Tag</b>	Tag de Comunicação
--------------------	--------------------

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

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

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

## IO.Ethernet.SocketState

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

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

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

## Propriedades

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

### NOTA

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

## IO.Ethernet.AcceptConnection

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

## IO.Ethernet.BackupEnable[2,3]

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

desabilitar a utilização.

## IO.Ethernet.BackupIP[2,3]

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

## IO.Ethernet.BackupLocalPort[2,3]

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

## IO.Ethernet.BackupLocalPortEnable[2,3]

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

## IO.Ethernet.BackupPort[2,3]

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

## IO.Ethernet.IPFilter

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

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

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

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

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

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

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

## IO.Ethernet.ListenIP

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

## IO.Ethernet.ListenPort

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

## IO.Ethernet.MainIP

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

## IO.Ethernet.MainLocalPort

**9** Número da porta local a ser utilizada na conexão ao endereço IP principal de um equipamento remoto. Este valor é usado apenas se a propriedade **IO.Ethernet.MainLocalPortEnable** é igual a Verdadeiro.

## IO.Ethernet.MainLocalPortEnable

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

## IO.Ethernet.MainPort

**9** Número da porta IP em um equipamento remoto, usado em conjunto com a propriedade **IO.Ethernet.MainIP**.

## IO.Ethernet.PingEnable

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

## IO.Ethernet.PingTimeoutMs

**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.SupressEcho

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

## IO.Ethernet.Transport

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

## IO.Ethernet.UseIPv6

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

# Configuração da Interface Modem

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

## Tags de Comunicação

### Tags da Interface Modem (N2/B2 = 3)

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

#### IMPORTANTE

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

## IO.TAPI.ConnectionBaudRate

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

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

## IO.TAPI.Dial

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

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

## IO.TAPI.HangUp

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

Qualquer valor escrito neste Tag desliga a chamada atual.

### NOTA

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

## IO.TAPI.IsModemConnected

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

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

## IO.TAPI.IsModemConnecting

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

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

## IO.TAPI.ModemStatus

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

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

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

## IO.TAPI.PhoneNumber

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

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

## Propriedades

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

## IO.TAPI.AcceptIncoming

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

## IO.TAPI.ModemID

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 um Driver deve utilizar o primeiro modem disponível.

## IO.TAPI.PhoneNumber

A O número de telefone utilizado em comandos **Dial**, como por exemplo "0w01234566", em que o caractere "w" força o modem a esperar por um sinal de chamada.

## Configuração da Interface RAS

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

### Tags de Comunicação

#### Tags da Interface RAS (N2/B2 = 5)

Atualmente, não existem Tags definidos especificamente para gerenciar a Interface **RAS** em tempo de execução.

### Propriedades

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

### NOTA

A Interface **RAS** utiliza a Interface **Ethernet**, que por este motivo também deve ser configurada.

## IO.RAS.ATCommand

A Comando **AT** a ser enviado através do *socket* para forçar um equipamento RAS a realizar uma ligação usando o canal RAS atual, como por exemplo "ATDT6265545".

## IO.RAS.CommandTimeoutSec

9 Tempo de espera pela mensagem **CONNECT** em resposta a um comando **AT**, em segundos.

## Configuração da Interface Serial

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

### Tags de Comunicação

#### Tags da Interface Serial (N2/B2 = 2)

Atualmente, não existem Tags definidos especificamente para gerenciar a Interface **Serial** em tempo de execução.

## Propriedades

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

### IO.Serial.Baudrate

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

### IO.Serial.CTSTimeoutMs

9 Tempo de espera pelo sinal **CTS**, em milissegundos. Após o sinal **RTS** ser ligado (**ON**), um temporizador é iniciado para esperar pelo sinal **CTS**. Se este temporizador expira, um Driver aborta o envio de bytes através da porta serial. Disponível apenas quando a propriedade **IO.Serial.RTS** está configurada com o valor **Toggle** e a propriedade **IO.Serial.WaitCTS** está configurada em Verdadeiro.

### IO.Serial.DataBits

9 Especifica o número de bits de dados para a configuração da porta serial. Os valores possíveis são **5**: Cinco bits de dados, **6**: Seis bits de dados, **7**: Sete bits de dados ou **8**: Oito bits de dados.

### IO.Serial.DelayAfterMs

9 Número de milissegundos de atraso após o último byte ter sido enviado através da porta serial, mas antes de desligar (**OFF**) o sinal **RTS**. Disponível apenas quando a propriedade **IO.Serial.RTS** está configurada com o valor **Toggle** e a propriedade **IO.Serial.WaitCTS** está configurada em Falso.

### IO.Serial.DelayBeforeMs

9 Número de milissegundos de atraso após o sinal **RTS** ter sido ligado (**ON**), mas antes dos dados serem enviados. Disponível apenas quando a propriedade **IO.Serial.RTS** está configurada com o valor **Toggle** e a propriedade **IO.Serial.WaitCTS** está configurada em Falso.

### IO.Serial.DTR

A Indica o modo como um Driver lida com o sinal **DTR**. Os valores possíveis são **OFF**: Sinal **DTR** sempre desligado ou **ON**: Sinal **DTR** sempre ligado.

### IO.Serial.InterbyteDelayUs

9 Tempo de espera, em milissegundos (1/1000000 de um segundo), para cada dois bytes enviados pela Interface **Serial**.

### IO.Serial.InterframeDelayMs

9 Tempo de espera, em milissegundos, antes de enviar um pacote após o último pacote enviado ou recebido.

### IO.Serial.Parity

A Especifica a paridade para a configuração da porta serial. Os valores possíveis são **E ou Even**: Paridade par, **N ou None**: Sem paridade, **O ou Odd**: Paridade ímpar, **M ou Mark**: Paridade de marca ou **S ou Space**: Paridade de espaço.

### IO.Serial.Port

9 Número da porta serial local. Os valores possíveis são **1**: Utiliza a porta COM1, **2**: Utiliza a porta COM2, **3**: Utiliza a porta COM3 ou **n**: Utiliza a porta COMn.

## IO.Serial.RTS

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

## IO.Serial.StopBits

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

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

## Histórico de Revisões do Driver

VERSÃO	DATA	AUTOR	COMENTÁRIOS
4.0.75	02/09/2025	M. Ludwig	<ul style="list-style-type: none"> <li>• Driver atualizado para a biblioteca <b>IOKit</b> versão <b>3.0</b> e Visual Studio 2022 (<i>Case 37955</i>).</li> </ul>
4.0.74	13/05/2025	M. Salvador	<ul style="list-style-type: none"> <li>• Comandos analógicos de 16 e 32 bits agora são interpretados como <b>int16</b> e <b>int32</b>, respectivamente (<i>Case 37456</i>).</li> <li>• Corrigido o comportamento da recepção de dados ao usar a opção <b>Unify Objects with/without status</b> (<i>Case 37535</i>).</li> <li>• Criada a opção <b>Discard Events Before this date (DD/MM/YY)</b> (<i>Case 37650</i>).</li> </ul>
4.0.70	10/12/2024	M. Salvador	<ul style="list-style-type: none"> <li>• Criada uma opção para enviar eventos de Classes 2 (dois) e 3 (três) mais rapidamente, sem solicitar confirmação de uma aplicação, supondo que eventos de SOE são reportados somente na Classe 1 (um) (<i>Case 32987</i>).</li> <li>• Criada uma opção de banda morta individual absoluta, além da forma já</li> </ul>

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			<p>existente de individual por porcentagem (Case 33326).</p> <ul style="list-style-type: none"> <li>• Agora com a opção <b>Generate Events on First Write</b> configurada, um evento não é gerado na primeira escrita de um Tag caso a qualidade seja ruim (Case 33375).</li> <li>• Criada uma proteção para evitar a entrada de valores do tipo <b>Float</b> inválidos (Case 34437).</li> <li>• Adicionado suporte a pedidos de leituras com Variação 0 (zero) e Qualificador diferente de 6 (seis) (Case 36673).</li> <li>• Criada uma opção de número máximo de dias à frente do horário atual para que escritas sejam aceitas. O limite inferior de data é fixo em 01/01/1980 e corrigido o tratamento de escritas de valores <b>NULL</b> (Case 33162).</li> <li>• Agora este Driver responde a pedidos de leitura de <b>Strings</b> com Qualificador 6 (seis) e índice não especificado corretamente (integridade de objetos estáticos) e responde com erro a pedidos com outros Qualificadores (Case 34739).</li> <li>• Agora este Driver pode enviar objetos do tipo <b>String</b> (110 ou 111) com índice maior que 255 (Case 34774).</li> <li>• Melhorada a performance para tratar mensagens vindas de um Mestre (Case 35103).</li> <li>• Comunicação em UDP revisada. Agora este Driver responde para o endereço IP e porta TCP/IP utilizado por um Mestre em cada conexão (Case 35375).</li> <li>• Revisado o de tamanho máximo de mensagem de</li> </ul>

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			<p>aplicação, que agora está realmente limitado ao máximo declarado pelo usuário na propriedade <b>Max App Size</b> (Case 35646).</p> <ul style="list-style-type: none"> <li>• Corrigido um problema ao reportar eventos com um <i>offset</i> que acumula a cada conexão adicional (Case 36655).</li> </ul>
4.0.50	22/06/2022	M. Salvador	<ul style="list-style-type: none"> <li>• Agora é permitido informar mais de um endereço DNP na opção <b>My DNP Addresses</b>, separados por ponto e vírgula, compartilhando a mesma conexão (Case 30900).</li> <li>• Agora o Objeto 2005 pode ser independente do Objeto 2001 usando a opção <b>Unify objects with and without Status</b> desmarcada (Case 31397).</li> <li>• Implementados os Objetos 3006 e 3206 (<i>Double Floating Point</i>) (Case 31588).</li> <li>• Adicionado suporte ao Objeto 100 variação 1 (um) (Case 31780).</li> <li>• Criado um Tag com o parâmetro <i>N2</i> igual a -42 (<i>force_events</i>), que permite forçar a geração de eventos de um tipo específico definido no parâmetro <i>N3</i>. O parâmetro <i>N4</i> define como é tratada a estampa de tempo (Case 30899).</li> <li>• Agora é possível informar quais Classes este Driver começa enviando, informando na opção <b>Use Unsolicited Msg</b> uma sequência de números 1 (um), 2 (dois) ou 3 (três) separados por ponto e vírgula, se necessário. Esta configuração pode ser alterada pelo Mestre ao enviar os comandos de <b>Enable</b> ou <b>Disable unsolicited</b> (Case 32331).</li> </ul>

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			<ul style="list-style-type: none"> <li>• Implementados os Objetos de Variações 3006, 3206 e 3208 de 64 bits (<i>Case 32401</i>).</li> <li>• Criada a opção <b>Disconnect after unconfirmed unsolicited sequence count</b>, que permite informar o número máximo de mensagens não solicitadas consecutivas sem resposta para que um Escravo realize uma desconexão forçada (<i>Case 32403</i>).</li> <li>• Agora os pedidos <i>deferred</i> são respondidos corretamente após o recebimento de uma confirmação ou <i>time-out</i> (<i>Case 32177</i>).</li> <li>• Eventos analógicos sem estampa de tempo agora são enviados na ordem em que foram gerados. Além disto, se a opção <b>One Analog Event per Index</b> está selecionada, então na ocorrência de um novo evento e um anterior está pendente de envio, então o valor é atualizado mas a ordem original é mantida. Demais eventos continuam sendo ordenados por estampa de tempo, se usada, ou por índice (<i>Case 32223</i>).</li> <li>• Agora este Driver espera corretamente o <i>time-out</i> ou a chegada de confirmação para enviar a próxima mensagem não solicitada (<i>Case 32346</i>).</li> <li>• Melhorias nas regras para envio de mensagens não solicitadas, de forma a aumentar a velocidade de envio (<i>Case 32838</i>).</li> </ul>
4.0.38	18/05/2021	M. Salvador	<ul style="list-style-type: none"> <li>• Adicionado suporte ao Qualificador 8 (oito) nos pedidos de eventos das Classes 1 (um), 2 (dois) e 3 (três) (<i>Case 29111</i>).</li> </ul>

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			<ul style="list-style-type: none"> <li>• Implementado suporte à função <i>Assign Classes</i> (22) (<i>Case 29138</i>).</li> <li>• Implementada a resposta a pedidos de leitura do Objeto 80 Variação 1 (um) (<i>Internal Indications</i>, índices de zero a 15) (<i>Case 29192</i>).</li> <li>• Criada uma opção para ignorar a função <i>Assign Class</i> e definido um <i>SoeType</i> padrão sem estampa de tempo quando recebe uma função <i>Assign Class</i> e não há <i>SoeType</i> no Tag (parâmetro <i>N1</i>) (<i>Case 29246</i>).</li> <li>• Criada a opção <b>Unify objects with and without status</b>, que vem selecionada por padrão para manter compatibilidade com versões anteriores (<i>Case 29516</i>).</li> <li>• Criada uma nova opção de SOE igual a 3 (três), que gera 2 (dois) eventos, com e sem estampa de tempo (<i>Case 29769</i>).</li> <li>• Implementada uma opção de Analógica do tipo <b>BitString</b> (<i>Case 30602</i>).</li> <li>• Corrigidas as funções 20 (<i>Enable Unsolicited</i>), 21 (<i>Disable Unsolicited</i>) e 24 (<i>Record Current Time</i>) (<i>Case 28900</i>).</li> <li>• <i>Warm</i> e <i>Cold restart</i> agora iniciam imediatamente após a resposta de um pedido (<i>Case 29206</i>).</li> <li>• Corrigido o uso de banda morta individual para os Tags dos tipos <b>R32</b> e <b>R64</b> (<i>Case 29223</i>).</li> <li>• Adicionado suporte à Variação 0 (zero) com outros Qualificadores além do 6 (seis, não especificado) (<i>Case 29515</i>).</li> <li>• A resposta de Classe 0 (zero) agora pode ser</li> </ul>

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			<p>respondida no mesmo fragmento de aplicação com respostas de outros pedidos na mesma mensagem (<i>Case 29380</i>).</p> <ul style="list-style-type: none"> <li>• Adicionado suporte a pedidos de Objetos de eventos diretamente, com Variação 0 (zero) ou Variação específica (<i>Case 29615</i>).</li> <li>• Corrigido um problema durante uma Classe 0 (zero) caso o Escravo não tenha nenhum Objeto criado (<i>Case 29768</i>).</li> <li>• Criadas novas opções para publicação de eventos na primeira conexão e nas conexões subsequentes, com o objetivo do Mestre receber antes da Classe 0 (zero) os valores correntes dos Tags com estampa de tempo relativa à última alteração (<i>Case 30111</i>).</li> </ul>
4.0.18	10/08/2019	M. Salvador	<ul style="list-style-type: none"> <li>• Adicionado suporte para operações em UDP/IP (<i>Case 27693</i>).</li> </ul>
		C. Mello	<ul style="list-style-type: none"> <li>• Driver portado para o Visual Studio 2017 (<i>Case 27509</i>).</li> </ul>
4.0.17	14/02/2019	M. Salvador	<ul style="list-style-type: none"> <li>• Implementado suporte ao comando <b>Direct Operate No Ack</b> no Tag com o parâmetro <i>N2</i> igual a 6 (seis) (<i>Case 26212</i>).</li> </ul>
4.0.16	16/10/2018	M. Salvador	<ul style="list-style-type: none"> <li>• Definido um valor padrão 1 (um) para o número de conexões DNP (<i>Case 25520</i>).</li> </ul>
4.0.15	08/10/2018	M. Salvador	<ul style="list-style-type: none"> <li>• Implementado o parâmetro <b>Item</b> "DBR:valor" para especificar uma banda morta relativa (<i>Case 24431</i>).</li> <li>• Corrigida a resposta de requisições com um qualificador 27h, ou 39 em decimal (<i>Case 24291</i>).</li> <li>• Adicionado um novo Elemento no bloco de leitura de comandos (<i>Control Block</i>) para</li> </ul>

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			<p>identificar o Mestre que enviou um comando (<i>Case 22800</i>).</p> <ul style="list-style-type: none"> <li>• Adicionada uma opção para desabilitar a reinicialização via comandos <b>Cold Restart</b> e <b>Warm Restart</b> (<i>Case 22328</i>).</li> <li>• Corrigido um problema com mensagens não solicitadas após uma desconexão (<i>Case 21757</i>).</li> <li>• Adicionado suporte a múltiplos canais (<i>Case 21616</i>).</li> <li>• Corrigido um erro de arredondamento no Objeto 41 Variação 03 (<i>Case 21162</i>).</li> <li>• Implementadas melhorias nos logs da camada de <i>Data Link</i> (<i>Case 20168</i>).</li> <li>• Adicionada uma opção para definir a banda morta individualmente por Tag (<i>Case 20161</i>).</li> <li>• Modificada a banda morta de pontos analógicos e contadores para utilizar casas decimais (<i>Case 19337</i>).</li> <li>• Implementado um evento para escrita de valores nulos (<i>Case 19117</i>).</li> <li>• Melhorias no sistema de controle e tratamento de sequenciamento de mensagens (<i>Case 18783</i>).</li> <li>• Implementadas funções <i>callback</i> para Tags Bloco (<i>Case 18782</i>).</li> </ul>
4.0.1	20/02/2015	M. Salvador	<ul style="list-style-type: none"> <li>• Removidos os Objetos de comando da resposta de Classe 0 (zero) (<i>Case 18274</i>).</li> <li>• Adicionada uma opção para gerar eventos na primeira escrita de cada ponto (<i>Case 18273</i>).</li> <li>• Melhorias nos logs de comandos (<i>Case 18176</i>).</li> </ul>

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			<ul style="list-style-type: none"> <li>• Implementado um Tag com o parâmetro <i>N2</i> igual a -30 para invalidar todos os Tags existentes na base de dados (<i>Case 18175</i>).</li> <li>• Adicionadas opções para definir a pilha de eventos das Classes 1 (um), 2 (dois) e 3 (três) (<i>Case 18085</i>).</li> <li>• Melhorias no processamento de mensagens não solicitadas (<i>Case 17836</i>).</li> <li>• Implementado suporte para funções de <i>callback</i> (<i>Case 17835</i>).</li> <li>• Implementado o tipo de dados <b>String</b> (<i>Case 16921</i>).</li> <li>• Driver portado para a biblioteca <b>IOKit</b> versão <b>2.0</b> (<i>Case 15635</i>).</li> </ul>
3.2.1	22/02/2013	M.Salvador	<ul style="list-style-type: none"> <li>• Mudança do identificador de proteção.</li> </ul>
3.1.1	31/05/2012	M. Salvador	<ul style="list-style-type: none"> <li>• <b>Beta 1, 2:</b> Modificações no Bloco Único de Eventos.</li> <li>• <b>Beta 3:</b> Correção no teste de banda morta.</li> <li>• <b>Beta 4:</b> Suporte à leitura do Objeto 50 Variação 1 (um) e correção dos Qualificadores 7 (sete) e 8 (oito).</li> <li>• <b>Beta 6:</b> Implementação do Objeto 1 (um) Variação 1 (um).</li> <li>• <b>Beta 11:</b> Confirmação de aplicação não pode ser desmarcada, suporte a mensagens de Classe com qualificador 7 (sete), suporte aos Objetos 1 (um) Variação 1 (um) Qualificador 6 (seis) e Objeto 1 (um) Variação 0 (zero).</li> <li>• <b>Beta 12, 13:</b> Correções na banda morta.</li> <li>• <b>Beta 14, 15:</b> A função de ordenação na resposta estava invertendo a ordem de pontos sem estampa de tempo.</li> </ul>

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			<ul style="list-style-type: none"> <li>• <b>Beta 16:</b> Correção do Tag <b>OnOff</b>.</li> <li>• <b>Beta 17:</b> Limitação do tamanho máximo da camada de Aplicação em 2048 bytes.</li> <li>• <b>Beta 18:</b> Correção no suporte a pontos duplos nos Objetos 3 (três) e 4 (quatro).</li> <li>• <b>Beta 20:</b> Implementado o comando <b>Cancel</b> e verificação de comando <b>Select sobre Select</b>.</li> <li>• <b>Beta 22:</b> Correção de escrita do Objeto 10 Variação 1 (um).</li> <li>• <b>Beta 23:</b> Correção na propagação de qualidade na escrita de valores para os eventos.</li> <li>• <b>Beta 24:</b> Opção de tratamento de pontos analógicos se desconectado.</li> <li>• <b>Beta 25:</b> Opção de espera pelo comando <b>Reset of Remote Link</b>, que anteriormente sempre esperava.</li> <li>• <b>Beta 26:</b> Opção <b>Sort Analog by Index</b>.</li> <li>• <b>Beta 27:</b> Opção <b>Respond Link Status</b> estava utilizando código incorreto.</li> <li>• <b>Beta 28:</b> Mensagens de <i>Data Link</i> de <i>unacknowledged</i> não estavam sendo tratadas. Eventos de Objeto e Variação diferentes na mesma Classe não eram transmitidos ao mesmo tempo.</li> <li>• <b>Beta 29:</b> Opção de limitar as mensagens de Classe a um fragmento de <i>Data Link</i>.</li> <li>• <b>Beta 30:</b> Suporte a pedidos de eventos com Variação 0 (zero) e Qualificador 6 (seis).</li> </ul>

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			<ul style="list-style-type: none"><li>• <b>Beta 31:</b> Suporte a estampas de tempo convertidas para o formato <b>GMT</b> (<i>Greenwich Mean Time</i>).</li></ul>
<b>3.0.1</b>	20/01/2010	M. Salvador	<ul style="list-style-type: none"><li>• Reimplementada a camada de Aplicação (<i>Case 8270</i>).</li></ul>
<b>1.1.1</b>	31/08/2004	M. Salvador	<ul style="list-style-type: none"><li>• Todas as publicações anteriores ao controle de revisões.</li></ul>

**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](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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](mailto:evan@elipse.com.br)

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

[www.elipse.com.br](http://www.elipse.com.br)

[kb.elipse.com.br](http://kb.elipse.com.br)

[forum.elipse.com.br](http://forum.elipse.com.br)

[www.youtube.com/elipsesoftware](http://www.youtube.com/elipsesoftware)

[elipse@elipse.com.br](mailto: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)