

Introdução

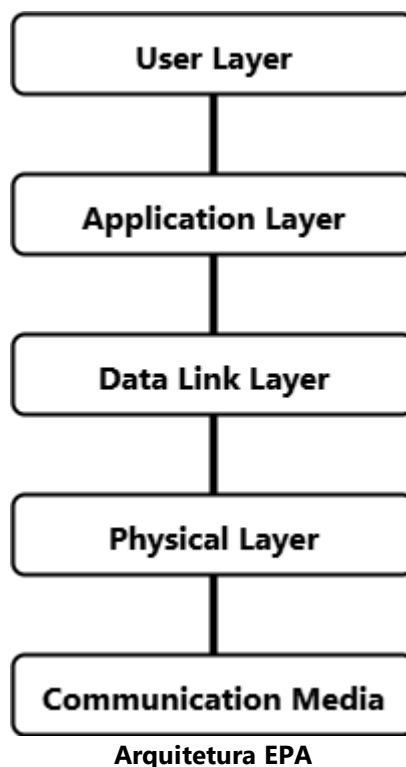
Nome do Arquivo	DNPMaster.dll
Fabricante	DNP (Distributed Network Protocol)
Equipamentos	
Protocolo	DNP 3.0
Versão	4.0.58
Última atualização	02/09/2025
Plataforma	Win32
Dependências	IOKit versão 2.0 ou superior
Leitura com Superblocos	Não
Nível	31201

Introdução

Este Driver implementa o protocolo DNP 3.0 no modo **Mestre**, de acordo com os níveis 2 (dois) e algumas das funcionalidades de nível 3 (três).

O DNP (*Distributed Network Protocol*) é um protocolo de comunicação aberto e não proprietário, baseado nas especificações da IEC (*International Electrotechnical Commission*), adaptado para ser utilizado 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 somente nas camadas física, *data link* e aplicação. Este modelo é chamado de EPA (*Enhanced Performance Architecture*). A figura a seguir mostra a estrutura EPA e o respectivo sistema de comunicação.



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

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

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

O protocolo DNP pode ser configurado para trocas de mensagens via *polling* (comunicação constante), ou via integridade ou mudanças (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. Recomenda-se a utilização das seguintes configurações:

- **Habilitação da Classe 0 (zero) na inicialização e a intervalos regulares:** Desta forma todos os Tags têm um valor ao iniciar a aplicação
- **Utilização de Mensagens não solicitadas ou Scan de eventos a intervalos regulares:** A atualização dos dados, conforme ocorrem mudanças, pode ser enviada de forma não solicitada pelo lado escravo ou através de solicitações de eventos das Classes 1 (um), 2 (dois) e 3 (três) automaticamente por este Driver
- **Configuração dos Tags utilizando objetos de eventos ao invés de objetos estáticos:** Tags configurados como objetos estáticos geram comunicação por *polling* (troca constante de mensagens), o que gera tráfego desnecessário. Tags configurados como eventos não realizam comunicação e são atualizados automaticamente conforme chegam mensagens de integridade ou de mudanças, conforme exposto anteriormente

Códigos de Função

Um código de função identifica o propósito de uma mensagem. Existe um grupo de funções para os pedidos e outro para as 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 (controles, SetPoints ou saídas analógicas) e responde com o status da operação. A função Operate deve ser usada para ativar estas saídas
4	Operate	Configura ou produz ações nas saídas ou pontos previamente selecionados com a função Select
5	Direct Operate	Seleciona e opera as saídas especificadas e responde com o status

CÓDIGO	FUNÇÃO	DESCRIÇÃO
		dos pontos de controle
6	Direct Operate NO ACK	Seleciona e opera as saídas especificadas mas não envia resposta, porém leva menos tempo
7	Immediate Freeze	Copia os objetos especificados para um <i>buffer</i> de congelamento
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, respondendo 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 a uma classe
23	Delay Measurement	Permite à 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 da mensagem, se necessária, chamada de ASDU. Cada ASDU consiste em um ou mais identificadores de dados (DUI), cabeçalhos de objetos, informações de objetos ou campos de dados.

Cabeçalhos de Objetos

O cabeçalho de objeto de uma mensagem identifica os objetos de dados que estão contidos nesta mensagem ou que são utilizados na resposta para esta mensagem. É composto basicamente dos elementos mostrados na figura a seguir.



Cabeçalho de objeto

Objetos

Os dispositivos inteligentes que usam a camada de aplicação do protocolo DNP são capazes de monitorar, controlar ou 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. Os tipos de categorias de objetos de dados existentes são os 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 por um objeto diferente, conforme descrito na tabela a seguir.

Categorias de objetos

OBJETO	DESCRIÇÃO
Entradas Digitais	O grupo de entradas digitais contém todos os objetos que representam entradas binárias (status ou atributos booleanos). Varia entre 1 (um) e 9 (nove)
Saídas Digitais	O grupo de saídas digitais contém todos os objetos que representam saídas binárias ou informação de controle de relés. Varia entre 10 e 19
Contadores	Este grupo contém todos os objetos contadores. Varia entre 20 e 29
Entradas Analógicas	Contém todas as entradas analógicas. Varia entre 30 e 39
Saídas Analógicas	Contém todas as saídas analógicas. Varia entre 40 e 49
Tempo	Contém todos os objetos que representam tempo em forma absoluta ou relativa. Varia entre 50 e 59
Classes	Este grupo contém todos os objetos que representam classes de dados ou prioridade de dados. Varia entre 60 e 69
Arquivos	Arquivos ou sistema de arquivos. Varia entre 70 e 79
Dispositivos	Varia entre 80 e 89
Aplicações	Objetos que representam aplicações de software ou processos do sistema operacional. Varia entre 90 e 99
Numéricos alternativos	Representações numéricas customizadas. Varia 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 exemplo, uma entrada digital 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 do 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 informação de tempo
- **Objeto 02 - Variação 02:** Representa uma mudança de entrada digital com 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*), descrito a seguir.

Faixa

Indica a quantidade de objetos, índices iniciais e finais ou identificadores para os objetos em questão.

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 a seguir:

- **Classe 0:** Significa todos os objetos, ou seja, na inicialização este 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

Esta seção contém informações sobre a configuração do Driver DNP 3.0 Master.

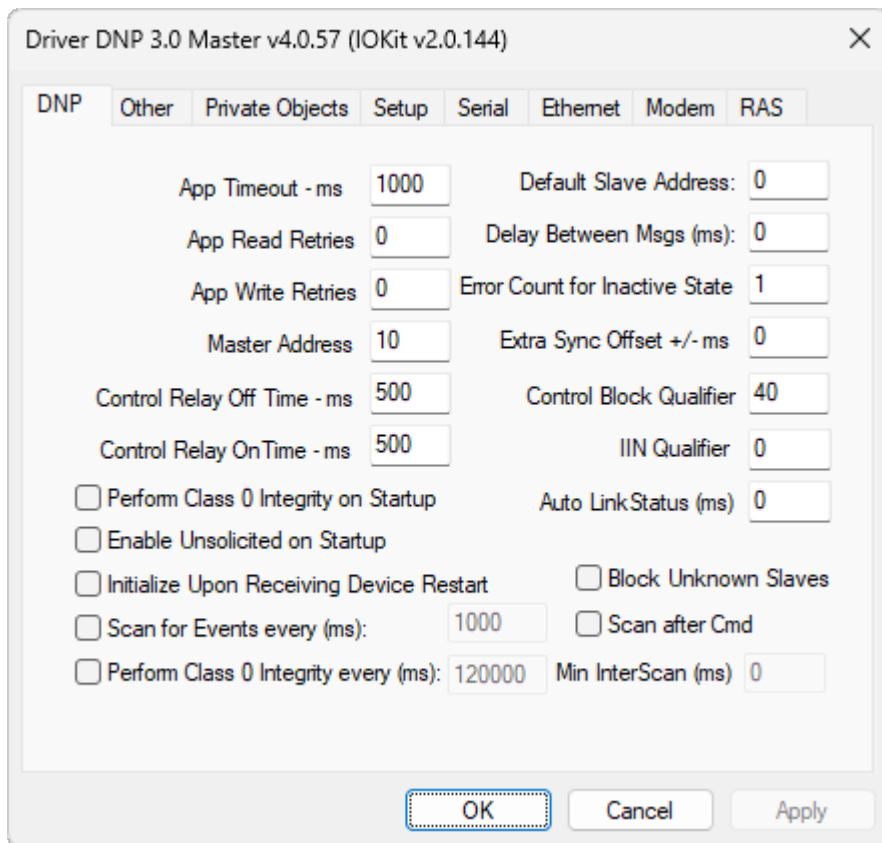
Parâmetros P de Configuração

P1	Não usado, mantenha o valor em 0 (zero)
P2	Não usado, mantenha o valor em 0 (zero)
P3	Não usado, mantenha o valor em 0 (zero)
P4	Mantenha o valor em 0 (zero). Caso esteja utilizando um equipamento Toshiba Regulator, configure em 1 (um)

Configurações Extras

Esta seção contém as configurações deste Driver contidas nas abas **DNP**, **Other** e **Private Objects**, descritas nos tópicos a seguir.

Aba DNP



Aba DNP

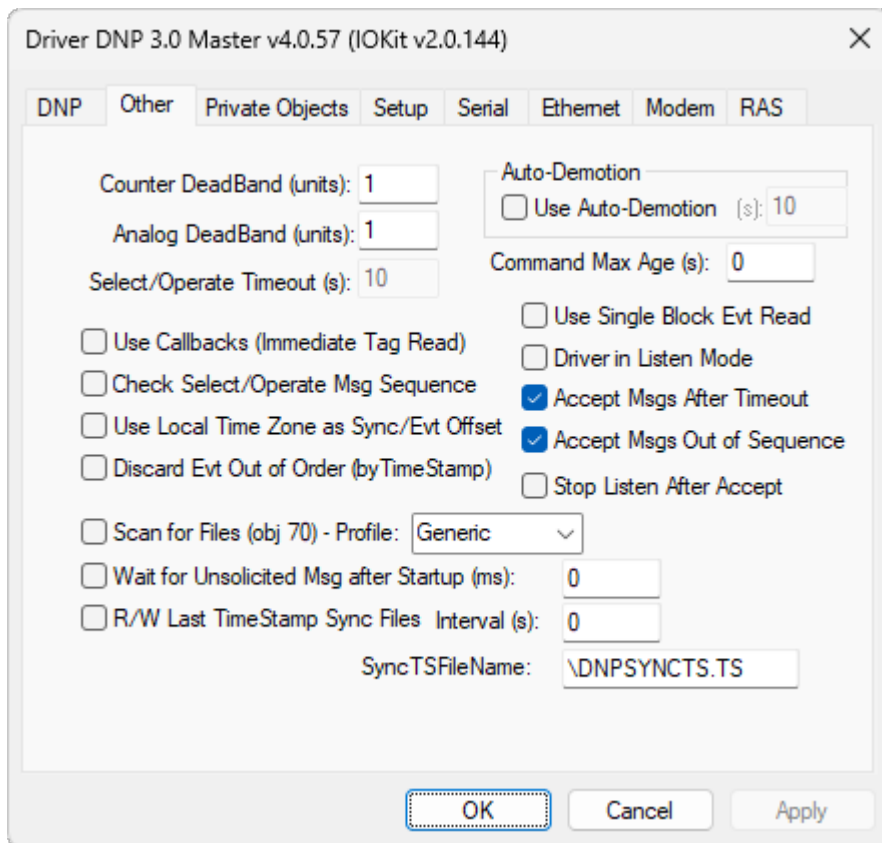
Opções disponíveis na aba DNP

OPÇÃO	DESCRIÇÃO
App Timeout (ms)	Tempo máximo que a camada de aplicação espera por uma resposta completa da camada <i>Data Link</i> . Se o recebimento do 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, mas recomenda-se que este valor seja igual ou superior ao valor de <i>time-out</i> da biblioteca IOKit , na aba Setup . Este valor representa o <i>time-out</i> byte a byte da mensagem, enquanto a opção App Timeout (ms) representa uma ou mais mensagens completas de <i>Data Link</i>
App Read Retries	Número de tentativas de comunicação realizadas pela camada de aplicação no caso de erro de leitura. O valor padrão desta opção é 0 (zero)
App Write Retries	Número de tentativas de comunicação realizadas pela camada de aplicação no caso de erro de escrita. O valor padrão desta opção é 0 (zero)
Master Address	Este é o endereço da estação Mestre (PC)
Control Relay Off-Time	Quando são enviados comandos de relés (<i>Control Relay Commands</i>) com Tags individuais, este campo indica o tempo normal de desligado ou <i>off-time</i> para os comandos Pulso On/Off ou Latch On/Off

OPÇÃO	DESCRIÇÃO
Control Relay On-Time	Quando são enviados comandos de relés (<i>Control Relay Commands</i>) com Tags individuais, este campo indica o tempo normal de ligado ou <i>on-time</i> para os comandos Pulso On/Off ou Latch On/Off
Default Slave Address	Endereço DNP padrão do dispositivo, a ser utilizado quando o parâmetro <i>N1</i> de cada Tag é configurado com o valor 0 (zero)
Delay Between Messages	Tempo de atraso a ser aplicado entre cada mensagem enviada por este Driver, em milissegundos
Error Count for Inactive State	Indica quantos erros consecutivos este Driver deve considerar para colocar um dispositivo em estado Inativo . Este Driver tenta comunicar novamente com este dispositivo no próximo <i>scan</i> de qualquer Tag do mesmo endereço ou somente no tempo informado na propriedade Demotion Time , caso este recurso seja utilizado
Extra Sync Offset	Tempo adicional, positivo ou negativo, a ser adicionado aos comandos de sincronismo
Control Block Qualifier	Ao realizar um comando de escrita de bloco de controle, como por exemplo um comando de saída digital como Direct Operate , Select ou Operate , é necessário o envio de um qualificador, que pode variar de equipamento para equipamento. Os valores possíveis neste campo são 39 (27h) para o campo Range com um byte e objeto prefixado com dois bytes, ou 40 (28h) para o campo Range com dois bytes e objeto prefixado com dois bytes. Consulte o <i>device profile</i> do equipamento para saber o qualificador correto
IIN Qualifier	Quando o dispositivo Escravo informa uma reinicialização, este Driver realiza uma escrita para o objeto de indicação interna (<i>Internal Indications Object</i>) informando o reconhecimento do evento. Use esta opção para informar o qualificador (presente no documento <i>Slave Device Profile</i> do dispositivo Escravo). Os valores mais comuns são 0 (zero, um byte de índice de início e fim) e 1 (um, dois bytes de índice de início e fim)
Auto LinkStatus (ms)	Intervalo para envio da mensagem <i>LinkStatus Request</i> em milissegundos, que deve receber como resposta uma mensagem <i>LinkStatus Response</i> . A falta da recepção coloca o endereço em estado Inativo . Para desabilitar esta opção, deixe o valor em 0 (zero)
Perform Class 0 Integrity On Startup	Realiza um pedido de Classe 0 (zero) ao iniciar este Driver, que obtém toda a base de dados do equipamento, permitindo que todos os Tags sejam atualizados. Os Tags de eventos, neste caso, têm a propriedade TimeStamp como sendo o horário da recepção dos dados e a propriedade Quality igual a 216
Enable Unsolicited on Startup	Indica se este Driver deve enviar o comando de habilitação de mensagens não solicitadas ao iniciar a comunicação
Initialize Upon Receiving Device Restart	Indica se este Driver deve enviar os comandos de inicialização do dispositivo (<i>Reset Link</i> , <i>Classe 0</i> e <i>Enable</i>

OPÇÃO	DESCRIÇÃO
	<i>Unsolicited</i> , caso habilitados) quando receber uma mensagem de que o dispositivo foi reiniciado
Scan for Events Every X ms	Envia um comando de leitura de Classes 1 (um), 2 (dois) ou 3 (três) no intuito de verificar se existem eventos no equipamento. Indique o intervalo de tempo que o comando deve ser repetido
Min InterScan ms	Se após alguma comunicação, ou mesmo após um <i>scan</i> de eventos, ainda existirem eventos pendentes no Escravo, o que é verificado através dos bits IIN, este Driver solicita novos eventos da classe ou classes pendentes. Para limitar esta comunicação de forma a não ocorrer continuamente, é possível definir um intervalo mínimo de <i>Interscans</i> através desta opção. O valor padrão é 0 (zero), de forma que se ainda houver eventos pendentes, estes são solicitados imediatamente, mantendo o comportamento das versões anteriores
Perform Class 0 Integrity every X ms	Realiza um pedido de Classe 0 (zero) ciclicamente, com o objetivo apenas de verificar a base de dados deste Driver, para saber se a base de dados do equipamento está idêntica. O tempo padrão é de 10 ou 15 minutos
Block Unknown Slaves	Bloqueia a criação de Escravos ao receber mensagens não solicitadas de endereços não declarados em nenhum Tag da aplicação
Scan after Cmd	Se a opção Scan for Events Every X ms está habilitada, através desta opção é possível solicitar um novo <i>scan</i> imediatamente após o envio de um comando, a fim de obter a mudança de estado mais rapidamente

Aba Other



Aba Other

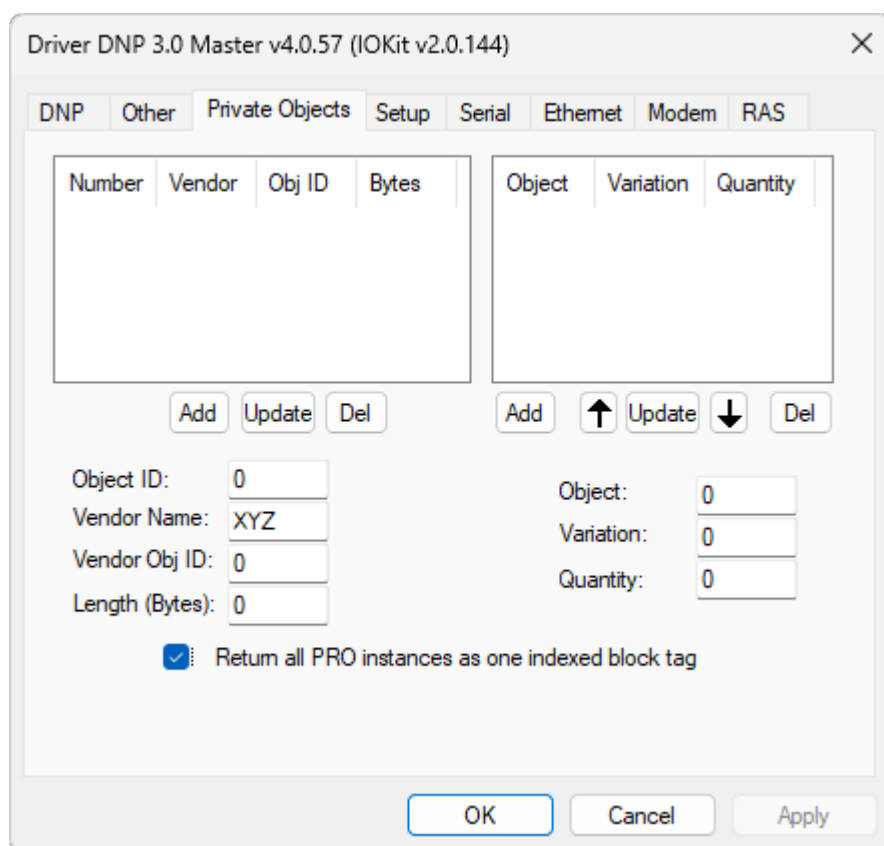
Opções disponíveis na aba Other

OPÇÃO	DESCRIÇÃO
Counter DeadBand (units)	Banda morta, em unidades, a serem cheçadas na verificação de novos eventos para os contadores
Analog DeadBand (units)	Banda morta, em unidades, a serem cheçadas na verificação de novos eventos para os pontos analógicos
Select/Operate Timeout	Tempo máximo para o bloqueio de outras mensagens, exceto um comando Operate logo após um comando Select
Use Auto Demotion	Habilita o sistema de <i>auto-demotion</i> , que retira e insere automaticamente da comunicação os Escravos que estiverem no estado Inativo , ou seja, com erro de comunicação. Este procedimento é utilizado quando dois ou mais Escravos utilizam o mesmo canal, evitando assim que um Escravo neste estado monopolize a utilização do canal
Demotion Time	Tempo, em segundos, que este Driver tenta comunicar com cada Escravo no estado Inativo , a fim de verificar se a comunicação está correta, colocando-o de volta no estado Ativo
Command Max Age (s)	Verifica se a estampa de tempo do comando solicitado tem no máximo a estampa de tempo local menos a quantidade de segundos definida nesta opção, de forma a evitar o tratamento de comandos antigos que, por algum motivo, permaneceram pendentes

OPÇÃO	DESCRIÇÃO
Use Callbacks	Este Driver pode ser configurado para utilizar <i>callbacks</i> ou chamadas assíncronas para o Elipse E3 , Elipse Power ou Elipse Water , permitindo que este último receba imediatamente os valores dos objetos que chegam na comunicação, e neste caso a taxa de <i>scan</i> não é utilizada. Caso esta opção esteja desabilitada, os Tags levam mais tempo para serem atualizados, em média mais do que o tempo de <i>scan</i> definido
Check Select/Operate Message Sequence	Permite a utilização de comandos Operate controlando o número de sequência da mensagem, para que seja imediatamente subsequente ao comando Select . Ao selecionar esta opção, este Driver não envia nenhum comando, exceto o comando Operate , até que o tempo máximo para a operação seja excedido
Use Local Time Zone as Sync/Event Time Offset	Instrui este Driver a considerar a diferença de horário oficial do Windows em relação ao horário UTC/GMT para todos os eventos e solicitações de sincronia de horário. Esta função é útil quando o lado Servidor tem o horário UTC mas deseja-se informar o horário relativo local para a aplicação
Discard Events out of Order (by TimeStamp Info)	Para os objetos que possuem informação de estampa de tempo, esta opção permite que eventos cuja estampa de tempo seja mais antiga do que o último evento processado sejam descartados
Scan for Files (obj 70)	Solicita a leitura de arquivos através do objeto 70 do DNP, que está descontinuado pela norma mas ainda é implementado em alguns equipamentos. Os perfis disponíveis são Pextron e Generic. Para mais informações, consulte o tópico Coleta e Gravação de Arquivos no Padrão COMTRADE
Wait for Unsolicited Messages after Startup (ms)	Indica quanto tempo este Driver deve esperar por mensagens não solicitadas após se conectar, com o objetivo de obter eventos recentes antes de realizar o pedido de integridade, ou Classe 0 (zero). Desta forma, os pontos com eventos recebem a qualidade 192 e estampa de tempo correta, visto que na resposta de integridade os objetos não possuem estampa de tempo
R/W Last TimeStamp Sync Files	Com esta opção, a última estampa de tempo de cada Tag é salva em um arquivo a intervalos pré-definidos, contados a partir de um novo evento recebido. Ao iniciar, este Driver tenta ler o arquivo para inicializar as estampas de tempo e descartar eventos mais antigos. O arquivo gerado pode ser copiado entre estações redundantes. O nome do arquivo e extensão pode ser alterado pelo usuário, e seu conteúdo está no formato ASCII
Use Single Block Evt Read	Através desta opção, todos os eventos enviados por um dispositivo são reportados em um único Tag Bloco. Esta opção é útil para concentrar todos os eventos a fim de gerar logs específicos de eventos, por exemplo. NOTA: Deve ser criado um Tag Bloco conforme descrito no tópico Bloco Único de Eventos para remover os eventos recebidos

OPÇÃO	DESCRIÇÃO
Driver in Listen Mode	Neste modo, este Driver não transmite nada, mas pode interpretar respostas, recebidas de um <i>sniffer</i> da comunicação entre outros Mestres e um Escravo
Accept Msgs After Timeout	Indica se este Driver deve aceitar uma resposta (<i>RESPONSE</i>) a uma mensagem solicitada após ultrapassado o tempo de <i>time-out</i> definido na opção App Timeout (ms) . O valor padrão desta opção é Falso para manter compatibilidade com versões anteriores
Accept Msgs Out of Sequence	Indica se este Driver deve aceitar uma resposta (<i>RESPONSE</i>) a uma mensagem solicitada se o número de sequência de aplicação é diferente do esperado

Aba Private Objects



Aba Private Objects

Os objetos proprietários são declarações que podem ser criadas por cada fabricante, e que refletem estruturas de dados compostas de tipos básicos de dados do protocolo DNP.

A aba **Private Objects** apresenta uma lista à esquerda onde são declarados os objetos, compostos de um índice (**Number**), um código de até quatro caracteres para o fabricante (**Vendor**), um número de identificação do objeto para o fabricante (**Object ID**) e o tamanho total em bytes do objeto (**Length**). Esta lista pode ser editada utilizando as opções **Add**, **Update** e **Del**.

Clique em um objeto para inserir os objetos DNP que são parte de cada um dos PROs (*Private Registration Objects*) na lista à direita. Deve ser informado o código do objeto (**Object**), a variação (**Variation**) e a quantidade (**Quantity**). A ordem na qual os objetos aparecem também é importante. Isto pode ser modificado utilizando as teclas de direção ou pelas opções **Add**, **Update** e **Del**.

Ao utilizar a declaração de objetos de tempo, ou seja, do tipo **50**, esta declaração é usada como estampa de tempo do objeto PRO.

A opção **Return all PRO instances as one indexed block tag** permite que todos os objetos PRO sejam enviados ao mesmo Tag Bloco, desde que possua o mesmo endereço DNP. Neste caso, o Tag Bloco deve ser criado com um Elemento a mais do que a declaração do objeto PRO, pois o primeiro Elemento contém o índice do objeto, e em seguida vêm os outros elementos, cada um com uma das variáveis declaradas no objeto PRO.

Parâmetros N de Endereçamento de Tags PLC

N1	Endereço do escravo ou 0 (zero) para usar o Default Slave Address
N2	Código de função a realizar. Para mais informações, consulte a tabela Códigos de Funções Suportadas
N3	Código do objeto e variação. Para mais informações, consulte a tabela Objetos Suportados
N4	Endereço da variável ou número

NOTAS

- O parâmetro *N3* deve ser informado como uma fórmula, *Object Code* × 100 + *Variation*. *Object Code* é o tipo de objeto, como por exemplo *Binary Inputs*, e *Variation* é um sub-tipo. Consulte a tabela **Objetos Suportados** para informações sobre objetos e variações suportados.
- O parâmetro *N4* é o endereço da variável ou número, não importando se é um ponto físico ou lógico.

Códigos de Funções

Códigos de funções suportadas

N2	LEITURA OU ESCRITA	OPERAÇÃO
-34	Escrita	Dados para canais analógicos do arquivo COMTRADE (perfil Pextron). O parâmetro <i>N3</i> indica o canal, de 0 (zero) a 7 (sete). Deve ser usado um Bloco com até oito Elementos. Os Elementos devem ser 0 : Nome do Canal, 1 : Constante A, 2 : Constante B, 3 : Valor Máximo do Canal, 4 : Valor Mínimo do Canal, 5 : Constante RPV, 6 : Constante RSV e 7 : Circuito Primário ou Secundário (pode ser p , P , s ou S)
-33	Leitura e escrita	Dados para a transferência do perfil Pextron. Os valores do parâmetro <i>N3</i> podem ser 0 : Indica que um arquivo acabou de ser gravado (pulso em um), 1 : Número de arquivos que faltam para transmitir, 2 : Solicita a coleta do arquivo Pextron (instantâneo) ou 3 : Solicita o início de contagem randômica para coleta do arquivo Pextron (conta o tempo para solicitar a coleta)
-32	Leitura e escrita	Nome base do arquivo COMTRADE

N2	LEITURA OU ESCRITA	OPERAÇÃO
-31	Leitura e escrita	Nome do circuito para arquivos COMTRADE
-30	Leitura e escrita	Diretório para armazenamento de arquivos
-22	Leitura	Retorna um Word contendo Internal Indications, IIN1 no byte Low e IIN2 no byte High . Os valores possíveis para os bits são 0 : All station messages received, 1 : Class 1 data available, 2 : Class 2 data available, 3 : Class 3 data available, 4 : Time sync required, 5 : Points in local status, 6 : Device trouble, 7 : Device restart, 8 : Not implemented, 9 : Requested objects unknown, 10 : Parameters not valid, 11 : Buffer overflow, 12 : Operation already executing, 13 : Configuration corrupted, 14 : Not used e 15 : Not used
-21	Leitura	Endereço indicado no parâmetro <i>N1</i> está ativo (um) ou inativo (zero)
-20	Leitura e escrita	Endereço indicado no parâmetro <i>N1</i> está ligado (um) ou desligado (zero)
-15	Escrita	Solicita um envio do comando de Link Status . O resultado é exibido no Tag com o parâmetro <i>N2</i> igual a -10
-14	Escrita	Solicita reiniciar a inicialização (Reset Link, Classe 0, etc.)
-13	Leitura e escrita	Estatísticas de Comunicação. Os valores possíveis para o parâmetro <i>N4</i> são 0 : Zera estatísticas (somente leitura), 1 : Frames enviados, 2 : Frames sem resposta, 3 : Erros de CRC no formato da resposta, 4 : Erros de envio, 5 : Retentativas de transmissão, 6 : Frames recebidos, 7 : Bytes enviados ou 8 : Bytes recebidos
-11	Leitura e escrita	Solicita Classe 0 (Integridade)
-10	Leitura	Link Status
-7	Escrita	Sincronia de tempo (procedimento Ethernet)
-6	Escrita	Sincronia de tempo. Escreve o horário do computador na estação remota
0	Leitura	Read from Cache . Remove todos os eventos que estiverem disponíveis para o Tag, deixando o último evento como o valor atual para o Tag. Para os objetos estáticos, realiza uma comunicação de leitura antes (função Read)

N2	LEITURA OU ESCRITA	OPERAÇÃO
1	Leitura	Read Events. Retorna todos os eventos que estiverem disponíveis para o Tag. Para os objetos estáticos, realiza uma comunicação de leitura antes (função Read)
2	Escrita	Write
3	Escrita	Select
4	Escrita	Operate
5	Escrita	Direct Operate
6	Escrita	Direct Operate No ACK
7	Escrita	Immediate Freeze
9	Escrita	Freeze and Clear
10	Escrita	Freeze and Clear No ACK
11	Escrita	Freeze with Time
13	Escrita	Cold Restart
14	Escrita	Warm Restart
20	Escrita	Enable Unsolicited Messages
21	Escrita	Disable Unsolicited Messages
22	Escrita	Assign Classes
23	Leitura	Delay Measurement
50	Leitura	Single Block Event
51	Leitura	Auto Zero Bad Read. Função usada quando o Tag recebe um evento de mudança apenas de subida (de zero para um) e nunca de descida (de um para zero). Neste caso, este Driver insere um evento de valor 0 (zero) antes de retornar cada evento em 1 (um)
101	Leitura	Read (Integridade). Executa uma operação de integridade, solicitando todos os pontos com o tipo de objeto $\times 100 + \text{Variação}$ declarados no parâmetro <i>N3</i> . Pode ser também usada a variação 0 (zero), correspondendo à qualquer objeto informado, não importando a variação. Esta função realiza uma comunicação (integridade) cada vez que o Tag é lido

Objetos Suportados

Objetos, variações, qualificadores ou funções suportadas

OBJETO	VARIAÇÃO	NOME DO OBJETO	CÓDIGO DE FUNÇÃO (N2)
1 (Estático)	1	Binary Input without Status	0, 1
1 (Estático)	2	Binary Input with Status	0, 1
2 (Evento)	1	Binary Input Change without Time	1
2 (Evento)	2	Binary Input Change with Time	1
2 (Evento)	3	Binary Input Change with Relative Time	1
3 (Estático)	1	Double bit binary Input without Status	0, 1
3 (Estático)	2	Double bit binary Input with Status	0, 1
4 (Evento)	1	Double bit binary Input Change without Time	1
4 (Evento)	2	Double bit binary Input Change with Time	1
10 (Estático)	1	Binary Output	0, 1
10 (Estático)	2	Binary Output Status	0, 1
11 (Evento)	1	Binary Output Event without Time	1
11 (Evento)	2	Binary Output Event with Time	1
12	1	Control Relay Output Block	3, 4, 5, 6
20 (Estático)	1	32-bit Counter	1, 7, 9, 10, 11
20 (Estático)	2	16-bit Binary Counter	1, 7, 9, 10, 11
20 (Estático)	3	32-bit Delta Counter	1, 7, 9, 10, 11
20 (Estático)	4	16-bit Delta Counter	1, 7, 9, 10, 11
20 (Estático)	5	32-bit Counter without Flag	1, 7, 9, 10, 11
20 (Estático)	6	16-bit Counter without Flag	1, 7, 9, 10, 11
20 (Estático)	7	32-bit Delta Counter without Flag	1, 7, 9, 10, 11
20 (Estático)	8	16-bit Delta Counter without Flag	1, 7, 9, 10, 11
21 (Estático)	1	32-bit Frozen Counter	1, 7, 9, 10, 11
21 (Estático)	2	16-bit Frozen Counter	1, 7, 9, 10, 11
21 (Estático)	3	32-bit Frozen Delta Counter	1, 7, 9, 10, 11
21 (Estático)	4	16-bit Frozen Delta Counter	1, 7, 9, 10, 11
21 (Estático)	5	32-bit Frozen Counter with Time Of Freeze	1, 7, 9, 10, 11
21 (Estático)	6	16-bit Frozen Counter with Time Of Freeze	1, 7, 9, 10, 11

OBJETO	VARIAÇÃO	NOME DO OBJETO	CÓDIGO DE FUNÇÃO (N2)
21 (Estático)	7	32-bit Frozen Delta Counter with Time Of Freeze	1, 7, 9, 10, 11
21 (Estático)	8	16-bit Frozen Delta Counter with Time Of Freeze	1, 7, 9, 10, 11
21 (Estático)	9	32-bit Frozen Counter without Flag	1, 7, 9, 10, 11
21 (Estático)	10	16-bit Frozen Counter without Flag	1, 7, 9, 10, 11
21 (Estático)	11	32-bit Frozen Delta Counter without Flag	1, 7, 9, 10, 11
21 (Estático)	12	16-bit Frozen Delta Counter without Flag	1, 7, 9, 10, 11
22 (Evento)	1	32-bit Counter Change Event without Time	1
22 (Evento)	2	16-bit Counter Change Event without Time	1
22 (Evento)	3	32-bit Delta Counter Change Event without Time	1
22 (Evento)	4	16-bit Delta Counter Change Event without Time	1
22 (Evento)	5	32-bit Counter Change Event with Time	1
22 (Evento)	6	16-bit Counter Change Event with Time	1
22 (Evento)	7	32-bit Delta Counter Change Event with Time	1
22 (Evento)	8	16-bit Delta Counter Change Event with Time	1
23 (Evento)	1	32-bit Counter Change Event without Time	1
23 (Evento)	2	16-bit Frozen Counter Event without Time	1
23 (Evento)	3	32-bit Frozen Delta Counter Event without Time	1
23 (Evento)	4	16-bit Frozen Delta Counter without Time	1
23 (Evento)	5	32-bit Frozen Counter Event with Time	1
23 (Evento)	6	16-bit Frozen Counter Event with Time	1
23 (Evento)	7	32-bit Frozen Delta Counter Event with Time	1
23 (Evento)	8	16-bit Frozen Delta Counter Event with Time	1

OBJETO	VARIAÇÃO	NOME DO OBJETO	CÓDIGO DE FUNÇÃO (N2)
30 (Estático)	1	32-bit Analog Input	0, 1
30 (Estático)	2	16-bit Analog Input	0, 1
30 (Estático)	3	32-bit Analog Input without Flag	0, 1
30 (Estático)	4	16-bit Analog Input without Flag	0, 1
30 (Estático)	5	32-bit Analog Input Floating Point	0, 1
31 (Estático)	1	32-bit Frozen Analog Input	0, 1
31 (Estático)	2	16-bit Frozen Analog Input	0, 1
31 (Estático)	3	32-bit Frozen Analog Input with Time Of Freeze	0, 1
31 (Estático)	4	16-bit Frozen Analog Input with Time Of Freeze	0, 1
31 (Estático)	5	32-bit Frozen Analog Input without Flag	0, 1
31 (Estático)	6	16-bit Frozen Analog Input without Flag	0, 1
31 (Estático)	7	32-bit Frozen Analog Input Floating Point	0, 1
32 (Evento)	1	32-bit Change Event without Time	1
32 (Evento)	2	16-bit Change Event without Time	1
32 (Evento)	3	32-bit Analog Change with Time	1
32 (Evento)	4	16-bit Analog Change Event with Time	1
32 (Evento)	5	32-bit Analog Change Floating Point without Time	1
32 (Evento)	7	32-bit Analog Change Floating Point with Time	1
33 (Evento)	1	32-bit Frozen Analog Event without Time	1
33 (Evento)	2	16-bit Frozen Analog Event without Time	1
33 (Evento)	3	32-bit Frozen Analog Event with Time	1
33 (Evento)	4	16-bit Frozen Analog Event with Time	1
33 (Evento)	5	32-bit Frozen Analog Floating Point without Time	1
33 (Evento)	7	32-bit Frozen Analog Floating Point with Time	1

OBJETO	VARIAÇÃO	NOME DO OBJETO	CÓDIGO DE FUNÇÃO (N2)
34 (Estático)	1	16-bit Analog Input Deadband	1
34 (Estático)	2	32-bit Analog Input Deadband	1
34 (Estático)	3	32-bit Analog Input Floating Point DeadBand	1
40 (Estático)	1	32-bit Analog Output Status	1
40 (Estático)	2	16-bit Analog Output Status	1
40 (Estático)	3	32-bit Analog Output Status Floating Point	1
41	1	32-bit Analog Output Block	2, 3, 4, 5, 6
41	2	16-bit Analog Output Block	2, 3, 4, 5, 6
41	3	32-bit Floating Point Analog Output Block	2, 3, 4, 5, 6
42 (Evento)	1	32-bit Analog Output Event without Time	1
42 (Evento)	2	16-bit Analog Output Event without Time	1
42 (Evento)	3	32-bit Analog Output Event with Time	1
42 (Evento)	4	16-bit Analog Output Event with Time	1
42 (Evento)	5	32-bit Floating Point Analog Output Event without Time	1
42 (Evento)	7	32-bit Floating Point Analog Output Event with Time	1
50 (Estático)	1	Time and Date	1, 2
51 (Estático)	1	Time and Date CTO (Common Time of Occurrence)	1
51 (Estático)	2	Unsynchronized Time and CTO (Common Time of Occurrence)	1
52 (Estático)	1	Time Delay Coarse	1
52 (Estático)	2	Time Delay Fine	1
60 (Estático)	1	Class 0 Data	1
60 (Evento)	2	Class 1 Data	1
60 (Evento)	3	Class 2 Data	1
60 (Evento)	4	Class 3 Data	1
80	1	Internal Indications	1, 2
83 (Evento)	1	Pro	1
110 (Estático)	X	Octet String	1

OBJETO	VARIAÇÃO	NOME DO OBJETO	CÓDIGO DE FUNÇÃO (N2)
111 (Evento)	X	Octet String Event	1

Qualificadores Suportados

Qualificadores suportados

OBJETO OU VARIAÇÃO	QUALIFICADOR DE ENVIO	QUALIFICADOR DE RECEPÇÃO
Objetos estáticos em geral (Leitura)	1	0, 1, 7, 8, 17, 18, 27, 28, 47, 58
Objeto 34 (Escrita)	1	0, 1, 7, 8, 17, 18, 27, 28, 47, 58
Objeto 60 (Classes) ou Pedidos Gerais de Integridade (Leitura)	6	0, 1, 7, 8, 17, 18, 27, 28, 47, 58
Objeto 50 (Escrita)	7	0, 1, 7, 8, 17, 18, 27, 28, 47, 58
Objeto 80 (Escrita)	Definido pelo usuário	0, 1, 7, 8, 17, 18, 27, 28, 47, 58
Objeto 12, Objeto 41 (SELECT e OPERATE, entre outros)	Definido pelo usuário	0, 1, 7, 8, 17, 18, 27, 28, 47, 58

Parâmetros B de Endereçamento de Tags Bloco

B1	Endereço do escravo
B2	Código de função a realizar. Para mais informações, consulte a tabela Códigos de Funções
B3	Código do objeto e variação (Objeto × 100 + Variação)
B4	Endereço inicial da variável ou número

A utilização de Blocos para leitura de eventos não é permitida, pois todos os Elementos do Bloco compartilham uma única estampa de tempo, o que não pode ser aceito pois cada evento tem seu próprio instante de ocorrência. Portanto, para leitura o parâmetro B2 deve ser igual a 0 (zero, **Read from cache**). A exceção é o uso de um **Bloco Único de Eventos** pois, neste caso, a estampa de tempo de cada evento é reportada em um Elemento fixo do Tag Bloco.

Exemplos de Configuração de Tags

Supondo-se um equipamento configurado com o endereço DNP igual a 3 (três), seguem exemplos dos parâmetros N1, N2, N3 e N4, no formato **N1.N2.N3.N4**.

```
Entrada Digital 100, Objeto 1 Variação 2: 3.1.102.100 (configurado como objeto estático)
Entrada Digital 100, Objeto 2 Variação 2: 3.1.202.100 (configurado como evento)
```

No exemplo anterior, ambos os Tags referenciam a mesma variável (Entrada digital 100), porém o primeiro Tag permanece continuamente perguntando o valor atual (*polling*), enquanto o segundo Tag apenas recebe notificações quando o ponto muda de valor. Além disto, na forma de **Evento** a estampa de tempo de mudança originada no equipamento é mantida. Já no modo **Estático**, como não existe estampa de tempo, esta é gerada com o horário do computador na chegada do evento.

Comentários Gerais

Esta seção fornece informações adicionais sobre o Driver DNP 3.0 Master.

Integridade de Dados e Eventos

O comando de integridade realiza um pedido de todos os dados configurados no Escravo, para todos os objetos de Classe 0 (zero) ou para objetos específicos.

Devido à particularidade do protocolo DNP enviar na integridade o valor atual das variáveis, portanto como objetos estáticos e não como eventos, é necessário que este Driver processe estas informações de forma a unificar o valor estático com o valor de evento, para que a aplicação possua apenas um único Tag para o ponto.

Desta forma, se a aplicação utiliza os Tags como eventos, conforme recomendado neste manual, então durante a inicialização, ao receber uma integridade de Classe 0 (zero) com o valor estático de um ponto, é gerado um evento provisório com a propriedade **Quality** igual a 216, indicando que a estampa de tempo utilizada não foi gerada pelo equipamento, mas sim pelo horário local do computador, devido à ausência desta informação na integridade.

Assim que houver a recepção de um evento para o Tag em questão, então a qualidade passa a 192 e a propriedade **Timestamp** reflete a estampa de tempo recebida no evento, caso o objeto contenha uma estampa de tempo.

NOTA

Os valores de qualidade expostos anteriormente pressupõem que o status dos objetos esteja informando uma situação normal, caso contrário os valores de qualidade refletem o status ruim do ponto, conforme exposto no tópico **Informação de Qualidade**.

Leitura de Eventos

A recepção de eventos por este Driver é fundamental para que os Tags configurados como eventos recebam seus valores. Este Driver só descobre que existem dados disponíveis para uma das classes de eventos (um, dois ou três) se a aplicação utilizar pelo menos uma das opções a seguir:

- A opção **Scan for Events Every X ms** na configuração deste Driver está habilitada. Isto instrui este Driver a solicitar as Classes 1 (um), 2 (dois) ou 3 (três) a intervalos fixos, recebendo a indicação de eventos
- O equipamento envia eventos espontaneamente através de mensagens não solicitadas
- A aplicação possui ao menos um Tag configurado para a leitura de uma variável estática qualquer, como por exemplo entradas ou saídas digitais ou analógica e contadores, entre outros

Comandos

Ao utilizar um objeto **Control Relay Output Block** (Objeto 12, Variação 1), uma série de dados é enviada por este Driver, como descrito a seguir.

Byte 0: Control Code

O valor do Tag atribuído na aplicação é copiado para este byte, que especifica detalhes sobre a operação do comando. Este campo é subdividido da maneira indicada a seguir.

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

- **Trip ou 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 **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, operação esta considerada ilegal

- **Clear**: Se o comando possui este campo em 1 (um, ligado), todas as operações de controle são removidas da fila, incluindo o comando que estiver sendo executado, e esta operação de controle é realizada
- **Queue**: Indica a colocação do comando em uma fila de comandos no 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 estiver sendo executado se o campo **Clear** estiver ligado. Quando a função de controle é executada e completada, esta é removida da fila. Se o comando em questão tem o atributo **Queue** ligado, então a operação é colocada na fila novamente (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	Os pontos são ligados pelo tempo especificado em On-Time , desligados pelo tempo especificado em Off-Time e deixados no estado OFF
0010	PULSE OFF	Os pontos são desligados pelo tempo especificado em Off-Time , ligados pelo tempo especificado em On-Time e deixados no estado ON
0011	LATCH ON	Mantém os pontos no estado ON
0100	LATCH OFF	Mantém os pontos no estado OFF

Outros valores fora desta tabela não são definidos.

Byte 1: Count

Este byte indica quantas vezes a 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 e fixo para todos os comandos.

Bytes 6 a 9: Off-Time

Tempo **Off-time**, em milissegundos, definido na janela de configurações extras deste Driver e 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 disponíveis são os seguintes:

- **0:** Comando executado corretamente, incluindo as operações **Select** e **Operate**
- **1:** Comando **Operate** enviado após o tempo máximo de **Select** definido no Escravo
- **2:** O comando **Operate** foi 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

O valor do campo **Status** pode ser obtido em uma aplicação **Elipse E3**, **Elipse Power** ou **Elipse Water** através do parâmetro *WStatus* do método **WriteEx** dos Tags ou Blocos, conforme o exemplo a seguir.

Para envio dos comandos, pode ser utilizado tanto um Tag PLC (IOtag) quanto um Tag Bloco. Ao usar um Tag PLC, deve ser atribuído ao Tag um número entre 0 (zero) e 255, que corresponde ao *Control Code* (byte zero da mensagem). Os bytes restantes são obtidos das configurações padrão deste Driver, definidas na janela de configurações extras.

Para um Tag Bloco deve ser usado um script que execute o método **Write** deste Tag Bloco. Para isto, o Tag Bloco deve possuir apenas quatro Elementos, que devem possuir a propriedade de escrita individual desabilitada:

- **Elemento 0:** *Control Code*
- **Elemento 1:** *Count*
- **Elemento 2:** *Relay On-Time*
- **Elemento 3:** *Relay Off-Time*

Este recurso pode ser utilizado caso sejam necessárias programações individuais de comandos para cada ponto, desprezando as configurações padrão de temporização **On-time** e **Off-time**.

Exemplo de script no Elipse SCADA usando um Bloco com quatro Elementos

```
//Configurações: B2=5 (DIRECT OPERATE), B3 = 1201
Block1.Element0 = 65 // Operation Code
Block1.Element1 = 1 // Count
Block1.Element2 = 500 // On-Time
Block1.Element3 = 500 // Off-Time
Block1.Write()
```

Exemplo de script no Elipse E3, Elipse Power ou Elipse Water usando um Tag de comando simples

```

Valor = 65 // Código da Operação
msg = MsgBox("Deseja enviar um comando?", 292, "Comando")
If msg = 7 Then
    MsgBox "Comando abortado", 48, "Desistência"
ElseIf msg = 6 Then
    If MsgBox("Confirma o envio do comando?", 292, "Atenção") = 6 Then
        Set Tag = Application.GetObject("DriverDNP.NomedoTag")
        If Tag.WriteEx(Valor, , , WStatus) = False Then
            MsgBox "Erro no envio do comando", 48, "Erro"
        Else
            If WStatus <> 0 Then
                MsgTrack = "Comando não executado,"
                EndText = ""
                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
                MsgBox MsgTrack & EndText, 48, "Erro"
            End If
        End If
    End If
End If

```

Bloco Único de Eventos

Caso a opção **Use Single Block Read for All Events** esteja habilitada, pode-se utilizar um Tag Bloco para receber todos os eventos de forma concentrada. Este Tag Bloco deve ter a seguinte configuração:

- **B1:** Endereço DNP
- **B2:** 50
- **B3 e B4:** Não utilizados
- **Tamanho:** Pode ter até sete Elementos:
 - **Elemento 0:** Objeto
 - **Elemento 1:** Variação
 - **Elemento 2:** Índice
 - **Elemento 3:** Status
 - **Elemento 4:** Valor
 - **Elemento 5:** Estampa de tempo
 - **Elemento 6:** Qualidade

NOTA

O Bloco Único de Eventos não pode ser usado em conjunto com outros Tags de evento em uma aplicação.

Informação de Qualidade

Para os objetos que possuem indicação de status (a grande maioria), este Driver realiza um mapeamento do status para o padrão OPC que é utilizado no **Elipse E3**, no **Elipse Power** ou no **Elipse Water**, exposto a seguir.

Para os objetos 1, 2 e 10

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

- **OL (Online):** Em zero, Q = 12 (*Bad, device failure*)
- **CL (Comm lost):** Em um, Q = 12 (*Bad, device failure*)
- **RS (Restart):** Em um, Q = 68 (*Uncertain*)
- **CF (Chatter filter):** Em um, Q = 80 (*Uncertain*)
- **RF (Remote forced):** Em um, Q = 216 (*Good, local override*)
- **LF (Local forced):** Em um, Q = 216 (*Good, local override*)

Para o objeto 40

Bit	7	6	5	4	3	2	1	0
Significado	0	RE	OR	LF	RF	CL	RS	OL

- **OL (Online):** Em zero, Q = 12 (*Bad, device failure*)
- **CL (Comm lost):** Em um, Q = 12 (*Bad, device failure*)
- **RS (Restart):** Em um, Q = 68 (*Uncertain*)
- **RE (Reference error):** Em um, Q = 80 (*Uncertain*)
- **OR (Over range):** Em um, Q = 66 (*Uncertain*)
- **RF (Remote forced):** Em um, Q = 216 (*Good, local override*)
- **LF (Local forced):** Em um, Q = 216 (*Good, local override*)

Demais objetos

Bit	7	6	5	4	3	2	1	0
Significado	0	XX	XX	LF	RF	CL	RS	OL

- **OL (Online):** Em zero, Q = 12 (*Bad, device failure*)
- **CL (Comm lost):** Em um, Q = 12 (*Bad, device failure*)

- **RS (Restart):** Em um, Q = 68 (*Uncertain*)
- **Bits 5 ou 6:** Em um, Q = 80 (*Uncertain*)
- **RF (Remote forced):** Em um, Q = 216 (*Good, local override*)
- **LF (Local forced):** Em um, Q = 216 (*Good, local override*)

NOTA

De acordo com o padrão OPC, uma qualidade maior ou igual a 192 é considerada **Boa** (*Good*).

Coleta e Gravação de Arquivos no Padrão COMTRADE

Caso a coleta de arquivos no perfil **Pextron** esteja habilitada, este Driver executa os seguintes passos:

1. Ao iniciar, conta-se um tempo randômico entre 0 (zero) e 13 minutos e então executa uma tentativa de coleta. Caso a coleta tenha sucesso, o arquivo é salvo e sua existência é indicada pelo Tag **DNPADD.-33.0.X**. O número de arquivos a transferir é indicado pelo Tag **DNPADD.-33.1.X**.
2. A aplicação pode escrever no Tag **DNPADD.-33.2.X** para solicitar uma nova coleta imediata ou no Tag **DNPADD.-33.3.X** para solicitar uma nova coleta após o tempo randômico. Em ambos os casos a coleta é executada independentemente se o Tag **DNPADD.-33.1.X** indicar 0 (zero) ou mais coletas a serem transferidas.
3. Caso nenhum Tag seja escrito, este Driver não executa novas coletas.

A configuração dos canais, diretório e nome do arquivo devem ser definidas antes do início da coleta.

Documentação das Interfaces de Comunicação

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

Configurações de um Driver

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

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

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

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

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

Caixa de Diálogo de Configuração

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

Aba Setup

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

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

The screenshot shows the 'Setup' tab of a configuration dialog. It includes the following elements:

- Physical Layer:** A dropdown menu currently set to 'Ethernet'.
- Start driver OFFLINE:** An unchecked checkbox.
- Timeout:** A text box containing '1000' followed by 'ms'.
- Communication check time:** A text box containing '5000' followed by 'ms'.
- Connection management:** A sub-section containing:
 - Mode:** A dropdown menu set to 'Automatic (managed by the driver)'.
 - Retry failed connection every:** A checked checkbox followed by a text box with '20' and 'seconds'.
 - Give up after:** An unchecked checkbox followed by a text box with '1' and 'failed retries'.
 - Disconnect if non-responsive for:** An unchecked checkbox followed by a text box with '0' and 'seconds'.
- Logging Options:** A sub-section containing:
 - Log to File:** An unchecked checkbox followed by a text box with the path 'C:\eeLogs\MicrolokII_%DATE%.log'.
 - File size limit (MB):** A text box with '0' and the note '(0 is unlimited)'.

Aba Setup

Opções gerais da aba Setup

OPÇÃO	DESCRIÇÃO
Physical Layer	Selecione a interface física em uma lista. As opções disponíveis são Serial , Ethernet , Modem e RAS . A

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

Opções para o grupo Connection management

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

Opções para o grupo Logging Options

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

Inter-frame delay (milliseconds): ms

Aba Serial

Opções gerais da aba Serial

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

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

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

Opções disponíveis no grupo Handshaking

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

Aba Ethernet

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

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: 	Port: 502	<input type="checkbox"/> Local port: 0
<input type="checkbox"/> Backup IP 1: 	Port: 0	<input type="checkbox"/> Local port: 0
<input type="checkbox"/> Backup IP 2: 	Port: 0	<input type="checkbox"/> Local port: 0
<input type="checkbox"/> Backup IP 3: 	Port: 0	<input type="checkbox"/> Local port: 0

Aba Ethernet

Opções disponíveis na aba Ethernet

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

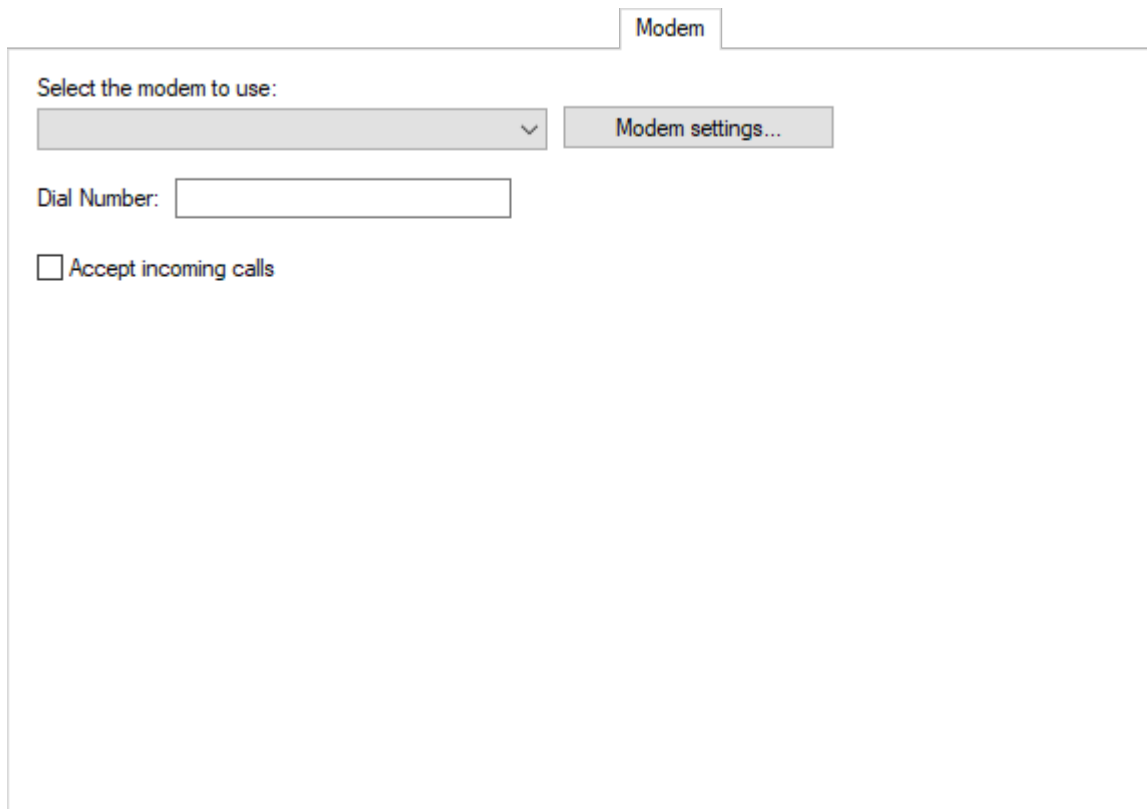
OPÇÃO	DESCRIÇÃO
IP Filter	Lista de endereços IP restringidos ou permitidos de onde um Driver aceita conexões (<i>Firewall</i>). Consulte a propriedade IO.Ethernet.IPFilter para mais informações
PING before connecting	Habilite esta opção para executar um comando ping , ou seja, para verificar se um dispositivo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

Opções disponíveis no grupo Connect to

OPÇÃO	DESCRIÇÃO
Main IP	Digite o endereço IP de um dispositivo remoto. Pode-se usar tanto o endereço IP separado por pontos quanto uma URL. No caso de uma URL, um Driver usa o serviço de DNS disponível para mapear a URL para um endereço IP, como por exemplo "192.168.0.13" ou "Server1"
Port	Digite a porta IP de um dispositivo remoto, entre 0 (zero) e 65535
Local port	Selecione esta opção para utilizar uma porta IP local fixa ao conectar a um dispositivo remoto
Backup IP 1, 2 e 3	Indique o endereço IP, a porta IP e a porta IP local fixa de até 3 (três) endereços de <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
Select the modem to use	Selecione um modem a partir da lista de modems disponíveis no computador. Caso selecione-se o valor Default modem , então o primeiro modem disponível é utilizado. Selecionar este valor é recomendado especialmente quando uma aplicação é utilizada em outro computador
Modem settings	Clique para abrir a janela de configuração do modem selecionado
Dial Number	Digite o número padrão para discagem. Este valor pode ser modificado em tempo de execução. Pode-se utilizar o caractere w para representar uma pausa ou espera pelo tom de discagem. Por exemplo, "0w33313456" disca o número 0 (zero), espera e então disca o número "33313456"
Accept incoming calls	Habilite esta opção para que um Driver atenda o telefone quando receber uma chamada externa. Para utilizar esta opção é necessário configurar a opção Connection management na aba Setup para o valor Manual

Aba RAS

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

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

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

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

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

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

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

Configurações Gerais

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

Tags de Comunicação

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

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

IO.CommunicationStatus

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

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

IO.IOKitEvent

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

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

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

NOTA

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

IO.PhysicalLayerStatus

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

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

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

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

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

IO.SetConfigurationParameters

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

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

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

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

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

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

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

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

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

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

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

IO.WorkOnline

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

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

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

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

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

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

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

IMPORTANTE

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

Propriedades

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

IO.ConnectionMode

9 Controla o modo de gerenciamento da Conexão. Os valores possíveis são **0**: Modo automático, em que um Driver gerencia a conexão ou **1**: Modo manual, em que uma aplicação gerencia a conexão.

IO.GiveUpEnable

Quando configurada para Verdadeiro, define um número máximo de tentativas de reconexão. Se todas as reconexões falharem, um Driver entra em modo **Offline**. Se configurada para Falso, um Driver tenta até que uma reconexão seja bem-sucedida.

IO.GiveUpTries

9 Número de tentativas de reconexão antes que esta seja abortada. Por exemplo, se o valor desta propriedade é igual a 1 (um), um Driver tenta apenas uma reconexão quando a conexão é perdida. Se esta falhar, este Driver entra em modo **Offline**.

IO.InactivityEnable

Configure em Verdadeiro para habilitar e em Falso para desabilitar a detecção de inatividade. A camada física é desconectada se está inativa por um certo período de tempo. A camada física é considerada inativa apenas se é capaz de enviar dados mas não de recebê-los de volta.

IO.InactivityPeriodSec

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

IO.RecoverEnable

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

IO.RecoverPeriodSec

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

NOTA

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

IO.StartOffline

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

NOTA

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

IO.TimeoutMs

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

IO.Type

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

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

Configuração de Estatísticas

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

Tags de Comunicação

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

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

IO.Stats.Partial.BytesRecv

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

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

IO.Stats.Partial.BytesSent

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

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

IO.Stats.Partial.TimeConnectedSeconds

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

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

IO.Stats.Partial.TimeDisconnectedSeconds

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

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

IO.Stats.Total.BytesRecv

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

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

IO.Stats.Total.BytesSent

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

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

IO.Stats.Total.ConnectionCount

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

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

IO.Stats.Total.TimeConnectedSeconds

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

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

IO.Stats.Total.TimeDisconnectedSeconds

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

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

Propriedades

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

Configuração da Interface Ethernet

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

Tags de Comunicação

Tags da Interface Ethernet (N2/B2 = 4)

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

IMPORTANTE

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

IO.Ethernet.IPSelect

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

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

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

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

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

IO.Ethernet.IPSwitch

Tipo de Tag	Tag de Comunicação
--------------------	--------------------

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

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

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

IO.Ethernet.SocketState

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

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

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

Propriedades

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

NOTA

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

IO.Ethernet.AcceptConnection

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

IO.Ethernet.BackupEnable[2,3]

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

desabilitar a utilização.

IO.Ethernet.BackupIP[2,3]

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: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

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

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

IO.TAPI.HangUp

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

Qualquer valor escrito neste Tag desliga a chamada atual.

NOTA

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

IO.TAPI.IsModemConnected

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

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

IO.TAPI.IsModemConnecting

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

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

IO.TAPI.ModemStatus

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

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

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

IO.TAPI.PhoneNumber

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

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

Propriedades

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

IO.TAPI.AcceptIncoming

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
4.0.58	02/09/2025	M. Ludwig	<ul style="list-style-type: none"> Driver atualizado para a biblioteca IOKit versão 3.0 e Visual Studio 2022 (<i>Case 37954</i>).
4.0.57	10/12/2024	M. Salvador	<ul style="list-style-type: none"> Criada uma opção Stop Listen After Accept, que é usada caso a opção Listen for connections on port esteja habilitada na aba Ethernet. Desta forma, este Driver não permanece no modo Listen após receber uma conexão (<i>Case 36661</i>). Este Driver agora interrompe a espera de respostas caso esteja sendo parado (<i>Case 36477</i>).
4.0.53	20/03/2024	M. Salvador	<ul style="list-style-type: none"> Leituras de Strings (objetos 110 e 111) agora funcionam corretamente (<i>Case 34740</i>). A Classe 0 não estava sendo realizada esporadicamente após um <i>device restart</i> (<i>Case 33970</i>). Adicionada uma proteção caso este Driver receba uma mensagem de tamanho menor que o esperado (<i>Case 33840</i>).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			<ul style="list-style-type: none"> • Resolvido um GPF (<i>General Protection Failure</i>) que poderia ocorrer quando se adiciona e remove eventos muito rapidamente (<i>Case 35489</i>).
4.0.45	29/03/2022	M. Salvador	<ul style="list-style-type: none"> • Implementadas variações ou objetos 3006, 3206 e 3208 (64 bits) (<i>Case 32404</i>). • Agora quando um Tag de <i>polling</i> ou solicitação de eventos (classe 1, 2 e 3) não consegue acesso ao canal de comunicação por estar ocupado com outra comunicação, no caso de uma não solicitada, isto não é considerado como erro de comunicação. Além disto, agora este Driver espera o término da recepção dos fragmentos para aceitar enviar novas mensagens, sempre priorizando o tratamento e confirmação das não solicitadas, o que já era realizado antes, apenas foi adicionada uma nova proteção contra colisões (<i>Case 31112</i>). • A solicitação de Link Status agora pode ser corretamente desabilitada (<i>Case 30482</i>). • Implementado suporte ao objeto 5201 (<i>Time Delay Coarse</i>) (<i>Case 30148</i>). • Correção de textos sobrepostos na caixa de diálogo de configuração (<i>Case 29115</i>).
4.0.40	27/11/2019	M. Salvador	<ul style="list-style-type: none"> • Corrigido o tratamento e as retentativas de limpeza do bit Device Restart - IIN1 Bit 7 (<i>Case 27925</i>).
		C. Mello	<ul style="list-style-type: none"> • Driver portado para o Visual Studio 2017 (<i>Case 27493</i>).
4.0.39	23/04/2019	M. Salvador	<ul style="list-style-type: none"> • Adicionadas as opções Accept Msgs After Timeout e Accept Msgs

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			<p>Out of Order na aba Other (Case 26541).</p> <ul style="list-style-type: none"> • Ajustes no valor da qualidade após o recebimento da integridade (Case 26525). • Ajustes no valor da qualidade após a troca de endereço IP de um Escravo redundante (Case 25830).
4.0.36	08/08/2018	M. Salvador	<ul style="list-style-type: none"> • Adicionada a propriedade Command Max Age (s) na aba Other (Case 24231). • Ajustes na sequência das mensagens Select e Operate (Case 24230). • Corrigido o tratamento de mensagens de Data Link iniciadas por um Escravo (Case 22797). • Adicionada a opção Block Unknown Slaves para prevenir envio de mensagens a equipamentos não declarados (Case 22581). • Adicionado suporte para solicitação de integridade e mudanças em ordem alfabética de múltiplos Escravos (Case 22401). • Adicionada a opção Scan After Cmd para efetuar um <i>scan</i> de eventos após o envio de comandos (Case 22219). • Melhorada a precisão do intervalo mínimo dos Tags -6 e -7 de sincronização (Case 22105). • Adicionada a opção Min Delay between Messages para comunicação (Case 21534). • Melhorias no tratamento de mensagens vindas de um Escravo como primárias (Case 21813). • Ajustes no tratamento da Classe 0 (zero) para prevenir conflitos com

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			<p>mensagens não solicitadas (Case 21627).</p> <ul style="list-style-type: none"> • Adicionado suporte para o objeto 41 variação 4 (Case 21184). • Corrigido o funcionamento do objeto 100 variação 1 (Case 20963). • Melhorias no processo de reconexão física da comunicação (Case 20962). • Melhorias no processamento de eventos (Case 20883). • Ajustes nas atualizações dos Tags de status para o modo Offline (Case 19395). • Adicionado um comando de escrita via Tag com parâmetro N2 igual a -15 para solicitação assíncrona da mensagem de Status Link aos Escravos (Case 19155). • Melhorado o controle de intervalo de <i>time-out</i> e solicitação de integridade da Classe 0 (Case 18956). • Adicionado um parâmetro de tempo mínimo entre <i>scans</i> de eventos pendentes (Case 18070). • Ajustes no Tag Link Status, parâmetro N2 igual a -10, para solicitação assíncrona da mensagem de Status Link aos Escravos (Case 17926).
4.0.21	27/10/2014	M. Salvador	<ul style="list-style-type: none"> • Adicionada uma opção para sincronização via Ethernet no Tag com o parâmetro N2 igual a -7 (Case 17569). • Ajustes no processamento dos Tags com múltiplos Escravos (Case 17518). • Adicionado suporte a objetos do tipo String 110 e 111 (Case 16920). • Corrigidos problemas com endereçamentos de

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			<p>Escravos para Tags de eventos (<i>Case 16919</i>).</p> <ul style="list-style-type: none"> • Correção de vulnerabilidade da notificação ICS-CERT VU-028282, conforme testes com o emulador de erros de formatação em mensagens DNP Aegis Fuzzer (<i>Case 16590</i>).
4.0.17	29/05/2014	M. Salvador	<ul style="list-style-type: none"> • Adicionado suporte aos objetos 4003 e 4103 (<i>Case 15232</i>). • Migração para o Visual Studio 2013 (<i>Case 16392</i>). • Migração para a biblioteca IOKit versão 2.0 (<i>Case 13516</i>). • Permissão de valores negativos na opção Extra Sync Offset da aba DNP. • Correção de <i>timers</i> de verificação de eventos e Classe 0 (zero) quando o horário do computador volta no tempo, ou seja, não usa mais o método etGetTimeDouble. • Geração de arquivos de sincronização da última estampa de tempo (<i>Case 13528</i>). • Melhorias de performance quando usados milhares de Drivers simultâneos (<i>Case 13117</i>). • Corrigido o comportamento quando este Driver só possui Tags de eventos com a opção de <i>callbacks</i> habilitada (<i>Cases 14040 e 16088</i>). • Bypass Reset of Remote Link na primeira comunicação não solicitada vinda do Escravo (<i>Case 15233</i>).
3.2.1	31/05/2012	M. Salvador	<ul style="list-style-type: none"> • Beta 1: Correções no Bloco Único de Eventos. • Beta 2: Adicionado suporte a leitura do objeto 50 variação 01. Correção de

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			<p>qualificadores 7 (sete) e 8 (oito). Ao conectar, a solicitação de eventos é realizada antes da Classe 0 (zero).</p> <ul style="list-style-type: none"> • Beta 3: Correção na verificação de eventos fora de ordem. • Beta 4: Tempo de espera por mensagens não solicitadas, em milissegundos. • Beta 5: Verificação de eventos fora de ordem é individual por Tag. • Beta 6: Correção de mensagem de log quando há erro de CRC. • Beta 7: Correção na informação de qualidade do Bloco Único de Eventos. • Betas 8 e 9: Correção no número de sequência esperado na camada de aplicação. • Beta 10: Correção de pontos duplos nos objetos 3 (três) e 4 (quatro). • Beta 11: Tag indicando número de <i>frames</i> recebidos. • Beta 12: Comando de escrita no objeto 10 variação 1. • Beta 13: Diminuição no tempo de espera de transmissão quando não conectado. • Beta 14: Tag com o parâmetro <i>N2</i> igual a -21 só informa Ativo após ter recebido a resposta de um Reset of Remote Link do Escravo. • Beta 15: Tratamento de 05 duplicado no início do frame, como por exemplo 05 05 64. • Beta 16: Separados os processos de envio, recepção e verificação.

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			<ul style="list-style-type: none"> • Beta 17: Processo de envio agora é síncrono com a camada de aplicação. • Beta 18: Opção de aplicar o <i>Local Time Zone</i> a eventos e sincronização. • Beta 19: Opção de descartar eventos fora de ordem estava sendo sempre aplicada. • Beta 20: Estatísticas de bytes enviados e recebidos. • Beta 21: Melhorias de desempenho com utilização de muitos Tags com <i>callbacks</i>. • Beta 22: Correção no tratamento do objeto 10 variação 1. • Beta 23: Liberação de confirmação de mensagens de aplicação mesmo sem um Reset of Remote Link, para permitir o processamento de mensagens não solicitadas recebidas imediatamente após a conexão. • Beta 24: Adição do Tag de indicação de IIN (-22).
3.1.1	14/12/2009	M. Salvador	<ul style="list-style-type: none"> • Correção do case 11020, em que a conexão era considerada inativa assim que entrava em estado ativo, devido a um erro de digitação interno. • Mudança de nome da biblioteca para DNPMaster.dll.
3.0.1	30/11/2009	M. Salvador	<ul style="list-style-type: none"> • Implementadas as opções Read from Cache e Read Events, tornando o comportamento similar ao do Driver IEC 60870-101/104. • Revisão da documentação.
2.29.1	30/11/2008	M. Salvador	<ul style="list-style-type: none"> • Correção de sincronização para mais de um Escravo no mesmo <i>link</i>.

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			<ul style="list-style-type: none"> • Correção de revalidação dos pontos após queda de conexão. • Correção de leitura de Tags Bloco com tipo de dados Float. • Correção na mensagem de verificação de eventos, em que a mesma classe era solicitada mais de uma vez na mesma mensagem. • Correção da competição entre processos quando em Select. • Opção de inicialização no <i>Device Restart</i>. • Coleta de arquivos.
2.28.1	29/10/2006	M. Salvador	<ul style="list-style-type: none"> • Retentativas de leitura e escrita independentes. • Bloco Único de Eventos. • Habilitação de mensagens não solicitadas ao iniciar.
2.20.1	20/12/2005	M. Salvador	<ul style="list-style-type: none"> • Controle On ou Off para cada endereço. • Controle de Ativo ou Inativo para cada endereço. • Controle automático de <i>Demotion</i>.
2.19.1	11/10/2005	M. Salvador	<ul style="list-style-type: none"> • Melhorias no processo de atualização da base de dados. • Leitura de eventos baseada em solicitação de classes ao invés do comando de leitura de relógio.
2.18.1	30/09/2005	M. Salvador	<ul style="list-style-type: none"> • Sistema de integridade e <i>polling</i> internos. • Correção de congelamento dos valores.
2.17.1	24/06/2005	M. Salvador	<ul style="list-style-type: none"> • Adicionado suporte ao equipamento Toshiba Regulator (<i>Case 5768</i>).
1.0.1	03/08/2004	M. Salvador	<ul style="list-style-type: none"> • Todas as publicações anteriores ao controle de revisões.

Matriz

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

90430-000 — Porto Alegre — RS

Fone: (+55 51) 3346-4699

Fax: (+55 51) 3222-6226

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

Filial no Paraná

Av. Sete de Setembro, 4698/1708

80240-000 — Curitiba — PR

Fone: (+55 41) 4062-5824

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

Filial no Rio de Janeiro

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

22250-044 — Rio de Janeiro — RJ

Fone: (+55 21) 2430-5912

Suporte Técnico: (+55 21) 2430-5963

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

Filial em São Paulo

Rua dos Pinheiros, 870/Cj. 141 e 142

05422-001 — São Paulo — SP

Fone: (+55 11) 3061-2828

Fax: (+55 11) 3086-2338

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

Filial em Minas Gerais

Rua Antônio de Albuquerque, 156/705

30112-010 — Belo Horizonte — MG

Fone: (+55 31) 4062-5824

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

Filial em Taiwan

9F., No.12, Beiping 2nd St., Sanmin Dist.

807 — Kaohsiung City — Taiwan

Fone: (+886 7) 323-8468

Fax: (+886 7) 323-9656

E-mail: evan@elipse.com.br

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

www.elipse.com.br

kb.elipse.com.br

forum.elipse.com.br

www.youtube.com/elipsesoftware

elipse@elipse.com.br



Gartner, Cool Vendors in Brazil 2014, April 2014.

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

Microsoft Partner

Gold Independent Software Vendor (ISV)