

Driver Allen-Bradley ABDF1

Nome do Arquivo	ABDF1.dll
Fabricante	Allen-Bradley (Rockwell Automation)
Equipamentos	PLC5, SLC500, Micrologix e ControlLogix
Protocolo	DF1, DF1 em CSPv4, PCCC e Ethernet/IP
Versão	2.0.18
Última Atualização	22/08/2025
Plataforma	Win32
Dependências	IOKit versão 2.0 ou superior
Leitura com Superblocos	Sim
Nível	0

Introdução

O Driver Allen-Bradley ABDF1 permite a comunicação serial entre sistemas da **Elipse Software** e os controladores programáveis da Allen-Bradley (Rockwell Automation) compatíveis com o protocolo DF1 (Serial) e via Ethernet, diretamente no canal Ethernet para o modelo SLC5/05 e via módulo 1761-NET-ENI para os modelos PLC5, Micrologix e SLC5/0X.

Para os equipamentos ControlLogix deve ser utilizado preferencialmente o *Driver Allen-Bradley Ethernet/IP (ABCIP)*, mas também é possível utilizar este Driver com um mapeamento das variáveis internas dos equipamentos ControlLogix para o formato **DF1**.

Preparando o Equipamento

Esta seção contém informações sobre a configuração dos parâmetros **[P]** deste Driver.

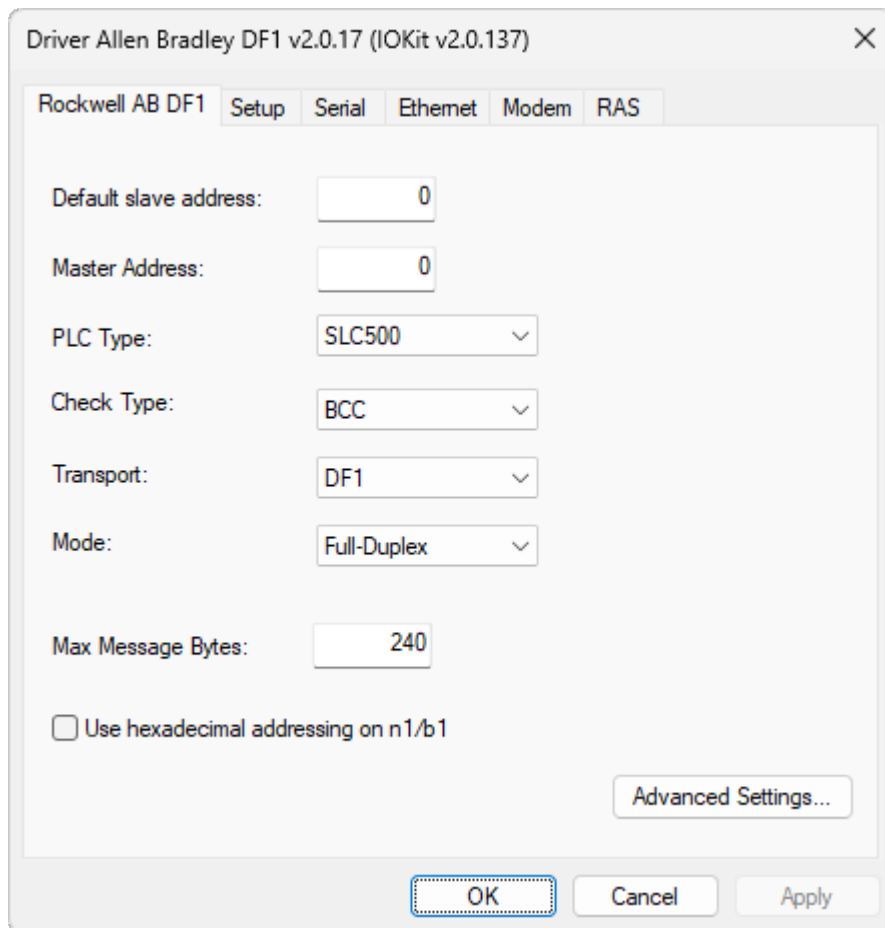
Parâmetros [P] de Configuração do Driver

P1	Não utilizado
P2	Não utilizado
P3	Não utilizado
P4	Não utilizado

O Driver Allen-Bradley ABDF1 utiliza a biblioteca **IOKit**, permitindo a comunicação via RS232 (DF1), Ethernet (DF1), Ethernet CSP, Ethernet/IP e Ethernet/IP com mensagens PCCC. Todos os parâmetros são definidos nas janelas de configuração da biblioteca **IOKit**. Para mais informações sobre esta biblioteca, consulte o tópico **Documentação das Interfaces de Comunicação**.

Janela de Configuração do Driver

Através da janela de configurações deste Driver, mostrada na figura a seguir, pode-se informar as configurações gerais.



Aba Rockwell AB DF1

Na aba **Rockwell AB DF1** encontram-se as configurações específicas deste Driver. Para informações sobre as demais abas, consulte o tópico **Documentação das Interfaces de Comunicação**. A tabela a seguir descreve as opções de configuração desta aba.

Opções disponíveis na aba Rockwell AB DF1

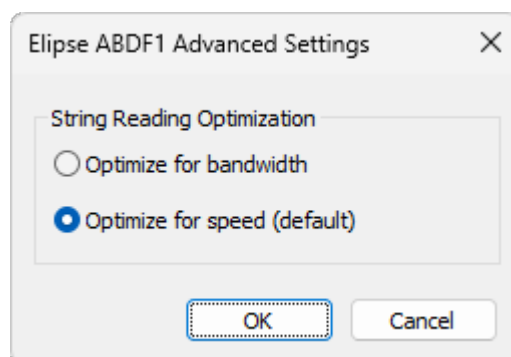
OPÇÃO	DESCRIÇÃO
Default slave address	Endereço padrão de um CLP. Utilizado sempre que no endereçamento dos Tags não é informado o endereço. Para a comunicação Ethernet CSP e Ethernet/IP (ENIP) , o endereço é sempre 0 (zero)
Master Address	Endereço padrão de um Mestre . Para comunicação Ethernet , o endereço é 0 (zero)
PLC Type	Tipo de CLP. As opções disponíveis são SLC500 , PLC5 , MicroLogix 1100 ou MicroLogix 1500 (porta Ethernet local) e MicroLogix ou ControlLogix (através de conversor Ethernet, emulando DF1)
Check Type	Tipo de verificação. As opções disponíveis são BCC ou CRC (somente para DF1)
Transport	Modo de transporte. As opções disponíveis são DF1 (geralmente se a comunicação é serial), CSPv4 (Ethernet) para SLC5/05 e ENIP se conectado através do conversor 1761 NET-ENI ou pela porta Ethernet do Micrologix 1100
Mode	Informa se é usado o modo Half-duplex ou o modo Full-duplex . Esta opção só é relevante se é usado o formato DF1 como camada de transporte

OPÇÃO	DESCRIÇÃO
Max Message Bytes	Este parâmetro define o número máximo de bytes da área de dados de cada <i>frame</i> de comunicação. Em caso de blocos com mais bytes, este Driver divide a comunicação em mais <i>frames</i> , cada um contendo no máximo o limite definido nesta opção. O valor padrão desta opção é 240. Verifique na documentação do equipamento se este valor pode ser aumentado. Em caso de dúvida, use o valor padrão
Use hexadecimal addressing on n1/b1	Se esta opção está habilitada, os parâmetros <i>N1</i> e <i>B1</i> dos Tags são alterados usando um endereçamento em formato hexadecimal. O byte mais significativo (MSB) indica o CLP de destino, enquanto o byte menos significativo (LSB) indica o número do arquivo. Com esta opção habilitada, é possível endereçar até 256 CLPs, o que não era possível com o endereçamento usado em versões anteriores, uma vez que o valor máximo dos parâmetros <i>N</i> e <i>B</i> é 65535 (16 bits). Com a opção habilitada, deve-se fornecer o valor dos parâmetros <i>N1</i> ou <i>B1</i> sempre em formato hexadecimal, adicionando o caractere "h" no Elipse SCADA ou o prefixo "&H" no Elipse E3 , no Elipse Power ou no Elipse Water . Para mais informações, verifique a <i>Referência da Linguagem VBScript</i> . Deixe esta opção desabilitada caso seja necessário manter a compatibilidade com versões anteriores deste Driver
Advanced Settings	Abre a Janela de Configurações Avançadas deste Driver, descrita no tópico a seguir

O restante dos parâmetros, como já mencionado, deve ser informado nas abas da biblioteca **IOKit**. Para a comunicação serial, o padrão é **19200 bps, 8 databits, 1 stopbit, no parity e check type igual a CRC**.

Janela de Configurações Avançadas

A janela de configurações avançadas deste Driver, mostrada na figura a seguir, é aberta clicando-se em **Advanced Settings** na aba **Rockwell AB DF1** da **Janela de Configurações** deste Driver.



Janela Elipse ABDF1 Advanced Settings

Esta janela reúne as configurações menos usadas, que em geral não requerem a atenção do usuário e podem ser mantidas nas opções padrão. Atualmente, as seguintes configurações avançadas estão disponíveis:

- **String Reading Optimization**

- **Optimize for bandwidth:** A cada leitura de **Strings**, este Driver executa 2 (duas) requisições a um equipamento. Na primeira requisição é lido apenas o valor numérico que indica o tamanho real da **String**, e na segunda requisição a **String** é lida com o tamanho detectado. Este procedimento tende a economizar o uso de banda, pois é lido apenas o tamanho exato da **String**. Este era o comportamento padrão deste Driver até a versão **2.0.3**
- **Optimize for speed (default):** Esta é a opção padrão para novas instâncias deste Driver, criadas com bibliotecas da versão **2.0.3** ou mais recentes. Nesta opção, este Driver executa uma única requisição para cada leitura de **Strings**, lendo sempre o tamanho máximo permitido, 82 caracteres, e descartando posteriormente os bytes não utilizados. Tende a deixar a leitura de **Strings** com aproximadamente o dobro da velocidade em comparação com a opção anterior, mas pode exigir a leitura de mais caracteres nesta requisição única, sempre 82 caracteres, do que seriam necessários

NOTA

Ao atualizar a biblioteca deste Driver em instâncias inicialmente criadas com versões anteriores à versão **2.0.3** em aplicações antigas, mantém-se o comportamento legado, ou seja, com a opção **Optimize for bandwidth** selecionada.

Referência de Tags

Esta seção contém informações sobre a configuração dos parâmetros **[N/B]** deste Driver.

Parâmetros **[N]** de Endereçamento para Tags do Tipo PLC

N1	Endereço de um CLP (<i>PC source</i>) × 1000 + número de um arquivo. Se um endereçamento hexadecimal é usado, MSB é igual ao endereço de um CLP e LSB é igual ao número de um arquivo. Para mais informações, consulte a tabela Números de Arquivos
N2	Tipo de variável. Para mais informações, consulte a tabela Tipos de Memória
N3	Número do elemento inicial de um arquivo
N4	Número do subelemento inicial de um arquivo. Para mais informações, consulte a nota a seguir

Se o endereço padrão de um CLP é configurado, o parâmetro *N1* é somente o número de um arquivo.

NOTA

Para manipulação de dados com arquivos do tipo **ASCII**, ou seja, com o parâmetro *N2* igual a 31, o parâmetro *N4* é utilizado para definir o tamanho dos dados (**String**), pois este tipo de arquivo não utiliza endereçamento com subelementos. Como a alocação interna dos dados ocupa um registrador de 16 bits, somente comprimentos de **String** com tamanho par de caracteres são permitidos.

Parâmetros [B] de Endereçamento para Tags do Tipo Bloco

B1	Endereço de um CLP (<i>PC source</i>) × 1000 + número de um arquivo. Se o endereçamento é em hexadecimal, MSB é igual ao endereço de um CLP e LSB é igual ao número de um arquivo. Para mais informações, consulte a tabela Números de Arquivos
B2	Tipo de variável. Para mais informações, consulte a tabela Tipos de Memórias
B3	Número do elemento inicial de um arquivo
B4	Número do subelemento inicial de um arquivo. Para mais informações, consulte a nota a seguir
Size	Número de Elementos de Bloco

Se o endereço padrão de um CLP é configurado, o parâmetro *B1* é somente o número de um arquivo.

NOTA

Para manipulação de dados com arquivos do tipo **ASCII**, ou seja, com o parâmetro *B2* igual a 31, o parâmetro *B4* é utilizado para definir o tamanho dos dados (**String**), pois este tipo de arquivo não utiliza endereçamento com subelementos. Como a alocação interna dos dados ocupa um registrador de 16 bits, somente comprimentos de **String** com tamanho par de caracteres são permitidos.

Números de Arquivos

Números de Arquivos

ARQUIVO	NÚMERO (N1 OU B1)	TIPO	DESCRIÇÃO
O0	0 (zero)	Output	Status dos terminais de saída
I1	1 (um)	Input	Status dos terminais de entrada
S2	2 (dois)	Status	Informações de operação de um CLP
B3	3 (três)	Bit	<i>Internal relay logic</i>
T4	4 (quatro)	Timer	Temporizador (<i>timer</i>), valores e bits de status pré-definidos
C5	5 (cinco)	Counter	Contador (<i>counter</i>), valores e bits de status pré-definidos
R6	6 (seis)	Control	Tamanho, posição do ponteiro e bits de status para instruções específicas como deslocamento (<i>shift</i>) de registradores
N7	7 (sete)	Integer	Armazena valores numéricos e informações de bit

ARQUIVO	NÚMERO (N1 OU B1)	TIPO	DESCRIÇÃO
F8	8 (oito)	Float	Armazena um número com um intervalo entre 1,1754944E-38 e 3,40282347E+38
F8	8 (oito)	Definido pelo usuário	Arquivos definidos pelo usuário para armazenamento de valores Bit, Timer, Counter, Control ou Integer
P9	9 (nove)	PID	Arquivos PID que contém diversas variáveis de controle
10-255	10 a 255	Definido pelo usuário	Arquivos definidos pelo usuário para armazenamento de valores Bit, Timer, Counter, Control ou Integer
10-255	10 a 255	Definido pelo usuário	Arquivos definidos pelo usuário para armazenamento de valores String (ST) ou ASCII (A) quaisquer

NOTA

Os arquivos podem variar de acordo com o modelo de CLP.

Tipos de Memória

Tipos de Memória (variável)

VARIÁVEL	ARQUIVO	TIPO (N2 OU B2)
Word	SLC_OUTPUT	0 (zero)
	SLC_INPUT	1 (um)
	SLC_STATUS	2 (dois)
	SLC_BIT	3 (três)
	SLC_TIMER	4 (quatro)
	SLC_COUNTER	5 (cinco)
	SLC_CONTROL	6 (seis)
	SLC_INTEGER	7 (sete)
	SLC_INTEGER, um inteiro de 16 bits com sinal	71. Para mais informações, consulte a nota a seguir
BCD	SLC_OUTPUT	8 (oito)
	SLC_INPUT	9 (nove)
	SLC_STATUS	10

VARIÁVEL	ARQUIVO	TIPO (N2 OU B2)
	SLC_BIT	11
	SLC_TIMER	12
	SLC_COUNTER	13
	SLC_CONTROL	14
	SLC_INTEGER	15
Bit	SLC_OUTPUT	16, apenas Tags individuais
	SLC_INPUT	17, apenas Tags individuais
	SLC_STATUS	18, apenas Tags individuais
	SLC_BIT	19, apenas Tags individuais
	SLC_TIMER	20, apenas Tags individuais
	SLC_COUNTER	21, apenas Tags individuais
	SLC_CONTROL	22, apenas Tags individuais
	SLC_INTEGER	23, apenas Tags individuais
Float	SLC_FLOAT	24
Long	SLC_LONG	25
String	SLC_STRING	26, apenas Tags individuais
Bit	SLC_LONG	28
Float	SLC_INTEGER	29
Word	SLC_PID. Para mais informações, consulte a nota a seguir	30
ASCII	SLC_ASCII	31

NOTAS

- Os tipos de memória 7 (sete) e 71 acessam o mesmo dado em um CLP. A diferença entre ambos é apenas a interpretação dos dados, que no primeiro caso são interpretados como inteiros de 16 bits sem sinal (**Word**) e no segundo caso como inteiros de 16 bits com sinal.
- Blocos para leitura de arquivos PID devem conter no máximo 23 Elementos, endereçados de 0 (zero) até 22.

Exemplos de Endereçamento

Exemplos de Endereçamento no Formato N1.N2.N3.N4

- Usando o endereçamento decimal:

```
CLP = 10, F8:40 (10008.24.40.0) (File 8, SLC_FLOAT)
CLP = 2, N25:100 (2025.7.100.0) (File 25, SLC_INTEGER)
CLP = 1, C5:42.ACC (1005.5.42.2) (File 5, SLC_COUNTER, Elemento 3)
CLP = 3, C5:42.PRE (3005.5.42.1) (File 5, SLC_COUNTER, Elemento 2)
```

- Usando o endereçamento hexadecimal:

```
CLP = 10, F8:40 (0A08h.24.40.0) (File 8, SLC_FLOAT)
CLP = 2, N25:100 (0219h.7.100.0) (File 25, SLC_INTEGER)
CLP = 1, C5:42.ACC (0105h.5.42.2) (File 5, SLC_COUNTER, Elemento 3)
CLP = 3, C5:42.PRE (0305h.5.42.1) (File 5, SLC_COUNTER, Elemento 2)
```

Exemplos de Endereçamento para Arquivos no Formato ASCII (N2 ou B2 igual a 31)

Arquivos no formato **ASCII** são endereçados por elementos de 16 bits e os dados são manipulados com base em um comprimento *n* de dados, todos em sequência, representando uma cadeia de caracteres. Por exemplo, para receber 20 caracteres a partir do endereço 10 de um arquivo no formato **ASCII**, configure um Tag PLC da forma indicada a seguir.

```
CLP = 1, A9:10 (1009.31.10.20) (File 9, SLC_ASCII, 20 caracteres a partir do Elemento 10 do arquivo)
```

No caso de Tags Bloco, o comprimento *n* de dados é processado para cada um dos Elementos. Por exemplo, para receber 20 caracteres em cada um dos três Elementos de um Tag Bloco, a partir do endereço 10 de um arquivo no formato **ASCII**, configure um Tag Bloco da forma indicada a seguir.

```
CLP = 1, A9:10 (1009.31.10.20) (File 9, SLC_ASCII, 20 caracteres a partir do Elemento 10 do arquivo,
para cada um dos Elementos do Tag Bloco)
Tag.Elemento1 = 20 caracteres a partir do Elemento 10 do arquivo
Tag.Elemento2 = 20 caracteres a partir do Elemento 20 do arquivo
Tag.Elemento3 = 20 caracteres a partir do Elemento 30 do arquivo
```

NOTAS

- Como a alocação interna dos dados ocupa um registrador de 16 bits, a configuração no parâmetro *N4* ou *B4* permite somente comprimentos de **String** com um tamanho par de caracteres.
- As leituras de **Strings** estão condicionadas ao comprimento configurado no parâmetro *N4* ou *B4*, mas pode retornar um tamanho inferior se um caractere **Null** (0x00), ou terminador de **String**, é recebido.
- As escritas de **Strings** com comprimento superior ao configurado no parâmetro *N4* ou *B4* têm os caracteres truncados até o limite deste parâmetro.
- As escritas de **Strings** com comprimento inferior ao configurado no parâmetro *N4* ou *B4* são preenchidas com caracteres **Null** (0x00) à direita até completar o comprimento definido neste parâmetro.

Configuração de Tags Através de Inteiros e Strings

Este Driver permite a configuração de Tags através de **Inteiros** e **Strings**. Para isto, devem ser usadas as propriedades **Dispositivo (ParamDevice)** e **Item (ParamItem)**, respectivamente.

A propriedade **Dispositivo** dos Tags tem o valor herdado deste Driver, pode ser sobrescrita em um Tag e define o endereço de um equipamento acessado por este Tag. Caso esta propriedade seja deixada vazia, automaticamente assume-se o valor 0 (zero).

Já a propriedade **Item** identifica um dado acessado por um Tag em um equipamento. Esta propriedade pode ter um dos formatos a seguir, em que os valores entre colchetes são opcionais.

```
[TIPODADO] AREA ARQUIVO [: ENDERECO] / [BIT]
```

Ou para *timers* e *counters*:

```
[TIPODADO] AREA ARQUIVO : ENDERECO. SUBELEMENTO
```

Ou para arquivos no formato **ASCII**:

```
[TIPODADO] AREA ARQUIVO : ENDERECO - TAMANHO DO DADO (STRING)
```

As tabelas a seguir definem os mnemônicos que podem ser usados nos valores descritos anteriormente.

Tipos de variáveis

TIPO DA VARIÁVEL	STRING	PARÂMETRO N2 OU B2
Word	W	Entre 0 (zero) e 7 (sete)
Long	DW, usado apenas com a área L	25
Signed Word	SW, usado somente com tipos de dados inteiros I	71
BCD	BCD	Entre 8 (oito) e 15
Bit	Sempre que o valor opcional Bit é definido	Entre 16 e 23
Float	Não definido. Para usar este tipo de variável, defina a área como F	24
String	Não definido. Para usar este tipo de variável, defina a área como ST	26
ASCII	Não definido. Para usar este tipo de variável, defina a área como A	31

Áreas

ÁREA	STRING
SLC_OUTPUT	O
SLC_INPUT	I
SLC_STATUS	S
SLC_BIT	B
SLC_TIMER	T
SLC_COUNTER	C
SLC_CONTROL	R
SLC_INTEGER	N
SLC_FLOAT	F
SLC_LONG	L, sempre com um tipo de variável DW
SLC_STRING	ST
SLC_ASCII	A
SLC_PID	PD, sempre com um tipo de variável W

A tabela a seguir contém exemplos de Tags configurados por **Strings**, relacionando-os com os parâmetros *N* ou *B* equivalentes.

Exemplos de Tags definidos por Strings

DISPOSITIVO	ITEM	N1 OU B1	N1 OU B1 (HEX)	N2 OU B2	N3 OU B3	N4 OU B4
2 (dois)	WO0:0/4	2000	0200h	16	0 (zero)	4 (quatro)

DISPOSITIVO	ITEM	N1 OU B1	N1 OU B1 (HEX)	N2 OU B2	N3 OU B3	N4 OU B4
8 (oito)	SWI1:86	8001	0801h	71	86	0 (zero)
8 (oito)	WS2:32	8002	0802h	2 (dois)	32	0 (zero)
0 (zero)	WI1:104	1 (um)	1 (um)	1 (um)	104	0 (zero)
11	BCDT4:38.9	11004	0B04h	12	38	9 (nove)
10	F8:3	10008	0A08h	24	3 (três)	0 (zero)
6 (seis)	ST4:30	6004	0604h	26	30	0 (zero)
13	L7:90	13007	0D07h	25	90	0 (zero)
13	DWL7:100	13007	0D07h	25	100	0 (zero)
11	PD4:5.15	11004	0B04h	30	5 (cinco)	15
3 (três)	A9:10-20	3009	0309h	31	10	20

Solução de Problemas Comuns

1. Ao tentar comunicar com um equipamento, deve-se ter certeza que este equipamento não está comunicando com outra aplicação. Normalmente ocorre de o usuário realizar algumas configurações usando o RS Linx e não desativar a comunicação deste com um CLP ao encerrar a comunicação.
2. No caso de tentativas de leitura ou escrita de um elemento inexistente, o sistema indica um erro de comunicação. Isto também acontece caso se tente ler um Bloco com mais **Words** do que o arquivo especificado possui.
3. Só é possível configurar um Tag para comunicar com um bit se o subelemento de um arquivo é igual a 0 (zero), pois ao configurar o parâmetro *N2* ou *B2* para acessar um bit, este Driver considera que o subelemento é igual a 0 (zero). Caso haja necessidade de buscar estes bits e o subelemento seja diferente de 0 (zero), aconselha-se configurar um Tag para buscar um dado completo e acessar estes bits com as funções existentes no **Elipse E3**, no **Elipse Power**, no **Elipse Water** ou no **Elipse SCADA**.
4. Ao tentar ler e escrever diretamente em entradas e saídas, ou seja, em arquivos **00** e **I1**, dependendo do modelo e versão de um equipamento, podem ocorrer problemas de formatação numérica no endereçamento, gerando erros de comunicação registrados no log com a mensagem "AB_command returned error in status = 10!". Sugere-se o uso da técnica de mapeamento das entradas e saídas para inteiros, descrita no artigo *Problemas de endereçamento ao acessar elementos de arquivos de entradas e saídas (00 e I1) com o Driver Allen-Bradley ABDF1* do **Elipse Knowledgebase**.
5. Ao tentar comunicar com um equipamento MicroLogix 1100 através da porta Ethernet local e ocorrerem falhas, mude a configuração **PLC Type** para **MicroLogix 1500**. Algumas versões de *firmware* deste equipamento requerem este ajuste neste Driver.

Leitura com Superblocos

Este Driver tem suporte a Superblocos desde a versão **1.7**. Este recurso permite ao **Elipse E3**, ao **Elipse Power** ou ao **Elipse Water** agrupar os Tags configurados em uma aplicação em blocos com o maior tamanho possível, a fim de otimizar a comunicação. Para habilitar este recurso no **Elipse E3**, no **Elipse Power** ou no **Elipse Water**, configure a propriedade **EnableReadGrouping** de um objeto Driver com o valor Verdadeiro.

Nem todos os tipos de dados possuem suporte a este recurso. Os tipos de dados **BCD** e as áreas **Counters** e **Timers** não são agrupáveis.

O suporte ao agrupamento de blocos de bits ainda não está implementado na versão atual deste Driver, mas deve ser implementado em futuras versões.

O tipo de dados **29**, que lê inteiros de 16 bits e os formata como **Floats** de 32 bits, também não é agrupável na versão atual.

Para os Tags com saídas no formato **String**, recomenda-se a utilização de tabelas **A**, ou **ASCII**, para agrupamento das leituras de **Strings** via Superblocos. Caso um projeto opte pela utilização de tabelas **ST**, ou **String**, não é possível aplicar a leitura em agrupamento para este tipo de tabela.

Modo Legado

Para manter a compatibilidade com aplicações antigas, na versão **1.15** deste Driver foi criado um modo de operação chamado de **Modo Legado**. Neste modo, este Driver se comporta da mesma forma que na versão **1.14**. Este modo pode ser habilitado ou desabilitado das seguintes formas:

- Ao incluir um novo Driver em uma aplicação e carregar o arquivo ABDF1.dll, este modo está desabilitado
- Se já existe uma versão anterior deste Driver em uma aplicação e o arquivo ABDF1.dll é substituído pela versão mais recente, este modo está habilitado

Além disto, existe a opção de habilitar ou desabilitar em tempo de execução através do Tag **Set Configuration Parameters** da biblioteca **IOKit**. O parâmetro a ser configurado é *ABDF1.LegacyMode*, e o valor deve ser 0 (zero) para desabilitar e 1 (um) para habilitar.

Para mais informações sobre a utilização do Tag **Set Configuration Parameters** da biblioteca **IOKit**, consulte o tópico **Documentação das Interfaces de Comunicação - Configurações Gerais - Tags de Comunicação**.

Para mais informações sobre as diferenças entre as versões deste Driver, consulte o **Histórico de Revisões do Driver**.

Documentação das Interfaces de Comunicação

Esta seção contém a documentação das Interfaces de Comunicação referentes ao Driver **ABDF1**.

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. At the top, there's a 'Physical Layer' dropdown menu set to 'Ethernet' and a checkbox for 'Start driver OFFLINE'. Below that are two input fields: 'Timeout' set to '1000 ms' and 'Communication check time' set to '5000 ms'. A section titled 'Connection management' contains a 'Mode' dropdown set to 'Automatic (managed by the driver)', a checked checkbox for 'Retry failed connection every' with a value of '20 seconds', an unchecked checkbox for 'Give up after' with a value of '1 failed retries', and another unchecked checkbox for 'Disconnect if non-responsive for' with a value of '0 seconds'. The 'Logging Options' section has an unchecked checkbox for 'Log to File' with the path 'C:\eeLogs\MicrolokII_%DATE%.log' and a 'File size limit (MB)' input set to '0' with a note that '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 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

OPÇÃO	DESCRIÇÃO
	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

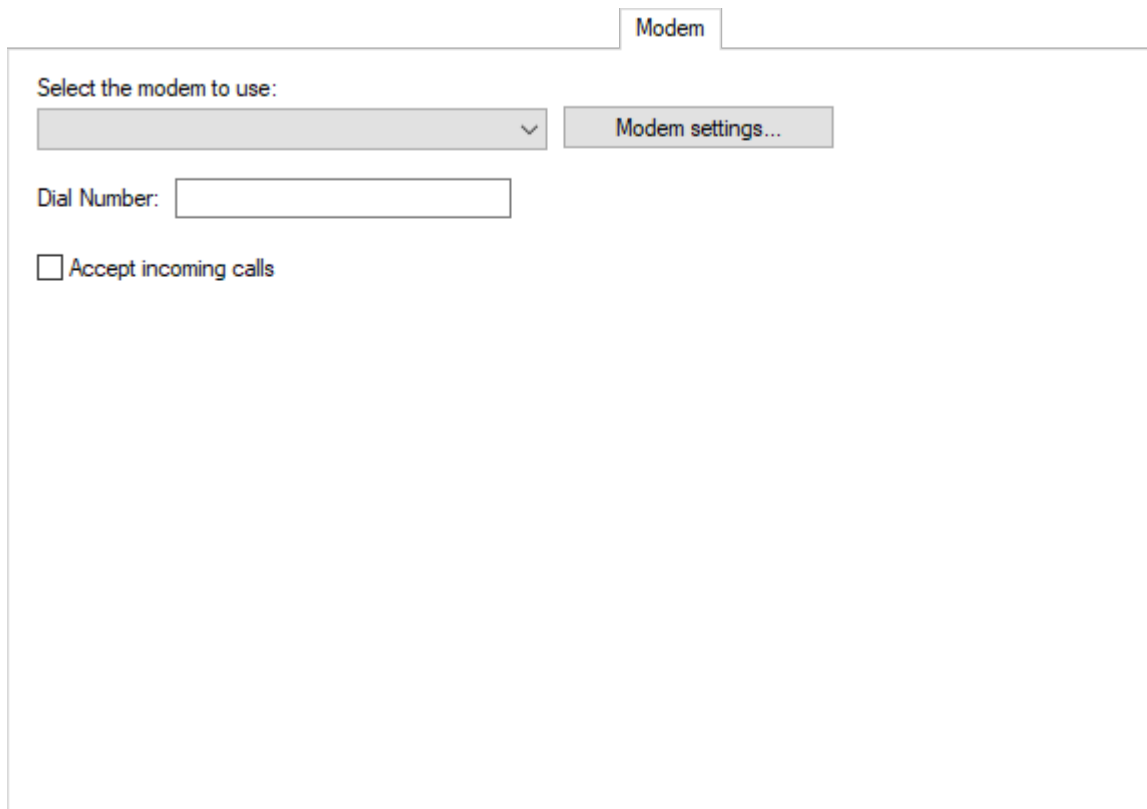
OPÇÃO	DESCRIÇÃO
IP Filter	Lista de endereços IP restringidos ou permitidos de onde um Driver aceita conexões (<i>Firewall</i>). Consulte a propriedade IO.Ethernet.IPFilter para mais informações
PING before connecting	Habilite esta opção para executar um comando ping , ou seja, para verificar se um dispositivo 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:0877:0000:0000:0000:0000: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
2.0.18	22/08/2025	M. Ludwig	<ul style="list-style-type: none"> Driver atualizado para a biblioteca IOKit versão 3.0 e Visual Studio 2022 (<i>Case 37930</i>).
2.0.17	08/12/2023	P. H. Santos	<ul style="list-style-type: none"> Implementada a validação de Tags e assumido o valor 0 (zero) para a propriedade Dispositivo se esta propriedade está vazia (<i>Case 35216</i>).
2.0.16	19/08/2019	C. Mello	<ul style="list-style-type: none"> Atualização de plataforma no código fonte deste Driver (<i>Case 27354</i>).
2.0.15	07/03/2019	C. Mello	<ul style="list-style-type: none"> Adicionado suporte para leitura em Superblocos para arquivos em formato ASCII (<i>Case 24440</i>).
2.0.14	28/01/2019	C. Mello	<ul style="list-style-type: none"> Adicionado suporte para uso de Tags Bloco em arquivos no formato ASCII (<i>Case 26033</i>).
2.0.13	17/09/2018	C. Mello	<ul style="list-style-type: none"> Adicionado suporte para manipulação de dados em arquivos no formato ASCII (<i>Case 25046</i>).
2.0.10	13/02/2017	A. Hertzog	<ul style="list-style-type: none"> Adicionado suporte a arquivos PID (<i>Case 21560</i>).
2.0.6	15/04/2016	M. Salvador	<ul style="list-style-type: none"> Implementado suporte a escritas e leituras diretas de

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			bits em áreas SLC_BIT (Case 20093).
2.0.5	10/09/2015	A. Quites	<ul style="list-style-type: none"> Adicionada uma opção avançada de otimização para a leitura de Strings (Case 18436).
2.0.1	26/09/2013	G. Taschetto	<ul style="list-style-type: none"> Driver portado para a biblioteca IOKit versão 2.0 (Case 14126).
1.15.1	21/11/2012	A. Quites G. Taschetto	<ul style="list-style-type: none"> Implementado um controle de transação do protocolo DF1 encapsulado em ENIP, permitindo a rejeição de quaisquer <i>frames</i> de resposta atrasados (Case 13090). O agrupamento por Superblocos foi desabilitado para Tags de acesso a arquivos de entrada e saída (parâmetro <i>N1</i> ou <i>B1</i> igual a zero ou igual a um), para evitar problemas com o eventual agrupamento de cartões não adjacentes (Case 13084). Adicionada uma sugestão de mapeamento de entradas e saídas digitais para variáveis inteiras no tópico Solução de Problemas Comuns (Case 12243). Corrigido um problema em que não era possível escrever em bits superiores ao bit 15 em cartões de I/O de 32 bits ou mais (Case 13274). Adicionada uma proteção na configuração de Tags por Strings contra o uso de bits no tipo de dados Float e de subelementos nos tipos de dados Integer e Float (Case 13339). Adicionado suporte a cartões de I/O de até 64 bits (Case 13330).
1.14.1	01/12/2010	A. Quites	<ul style="list-style-type: none"> Corrigido um erro na escrita de bits de variáveis

VERSÃO	DATA	AUTOR	COMENTÁRIOS
1.13.1	13/10/2010	A. Quites	<p>do tipo de dados Long (Case 11923).</p> <ul style="list-style-type: none"> • Corrigido um erro na escrita de bits do tipo de dados Word mais significativo de tipos de dados de 32 bits (Case 10768). • Adicionada uma verificação de transação (TNS) da camada DF1 quando encapsulada em CSPv4 (Case 11023). • Corrigido um erro na escrita de Elementos de Bloco do tipo de dados Float (Case 11024). • Driver portado para o Windows CE (Case 10913). • Corrigido um erro na leitura do tipo de dados 71 (I16) no Elipse E3, com Superblocos (propriedade EnableReadGrouping) desabilitado (Case 11653). • Corrigidos os exemplos de endereçamento do subelemento ACC (Case 10979).
1.12.1	22/04/2009	A. Quites	<ul style="list-style-type: none"> • Corrigido um erro na leitura de Tags internos da biblioteca IOKit (Case 10242). • Corrigido um erro de deslocamento (<i>offset</i>) ao escrever em Elementos de Bloco com tipos de dados de 32 bits (Case 10253). • Corrigido um erro ao escrever nos bits de 16 a 31 em tipos de dados de 32 bits (Case 10345). • Implementado um novo tipo de dados 29, que lê dois Words e os converte para Float (Case 10016). • Corrigido um erro de deslocamento (<i>offset</i>) ao ler Blocos de tipos de dados de 32 bits com tamanho total superior ao limite

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			<p>máximo configurado (<i>Case 10296</i>).</p> <ul style="list-style-type: none"> • Suporte a Superblocos de tipos de dados de 32 bits desabilitado, devido a um erro com o endereçamento de 16 bits (<i>Case 10297</i>).
1.11.1	07/10/2008	M. Salvador A. Qites	<ul style="list-style-type: none"> • Adicionado suporte a equipamentos Micrologix 1500 série B (<i>Case 9132</i>). • Adicionado suporte a escrita de bits em tipos de dados Long (<i>Case 9743</i>). • Adicionado um tipo de dados inteiro com sinal, correspondente ao parâmetro <i>N2</i> igual a 71 (<i>Case 9800</i>).
1.10.1	28/02/2007	A. Qites	<ul style="list-style-type: none"> • Resolvido um erro de TNS síncrono (<i>Case 7833</i>).
1.9.1	10/10/2006	M. Salvador	<ul style="list-style-type: none"> • Adicionado suporte a equipamentos Micrologix 1100 (<i>Case 7454</i>).
1.8.1	28/08/2006	A. Qites	<ul style="list-style-type: none"> • Corrigido um erro na recepção de pacotes em BCC (<i>Case 7289</i>).
1.7.1	26/07/2006	A. Qites M. Salvador	<ul style="list-style-type: none"> • Implementado o recurso de Superblocos e configuração de Tags por Strings (<i>Case 6742</i>). • Corrigido um erro na leitura e escrita de bits de entrada e saída (<i>Case 6415</i>). • Corrigido um problema na recepção nos modos Half-Duplex e Full-Duplex.
1.6.1	18/04/2004	M. Salvador	<ul style="list-style-type: none"> • Implementado um controle do número máximo de bytes na recepção. • Implementada a leitura de blocos maiores que o número máximo de bytes. • Implementada a escrita de bits.
1.5.1 Beta	15/04/2004	A. Qites M. Salvador	<ul style="list-style-type: none"> • Implementado o modo Half-Duplex na camada DF1 (<i>Case 5160</i>). • Criada a opção de configuração Use

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			hexadecimal addressing on N1/B1.
1.2.1	24/06/2004	M. Salvador	<ul style="list-style-type: none">• Versão anterior ao controle de revisões.
1.0.1	31/05/2004	M. Salvador	<ul style="list-style-type: none">• Versão original deste Driver (<i>Case 1020</i>).

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)