

Driver Atos APR03

Nome do Arquivo	APR03.dll
Fabricante	Atos Automação Industrial Ltda.
Equipamentos	MPC1200, MPC1600, MPC2000, MPC2002, MPC4004, EXPERT e módulos 4004.72 e 4004.72R
Protocolo	APR03
Versão	3.0.2
Última Atualização	01/09/2025
Plataforma	Win32
Dependências	IOKit versão 2.0 ou superior
Leitura com Superblocos	Sim
Nível	0

Introdução

Este é o Driver Atos APR03 para a comunicação entre sistemas da **Eclipse Software** e controladores da Atos Automação Industrial que implementem o protocolo APR03.

Preparando o Equipamento

Esta seção contém informações sobre os parâmetros **[P]** e configurações extras deste Driver.

Parâmetros **[P]** de Configuração do Driver

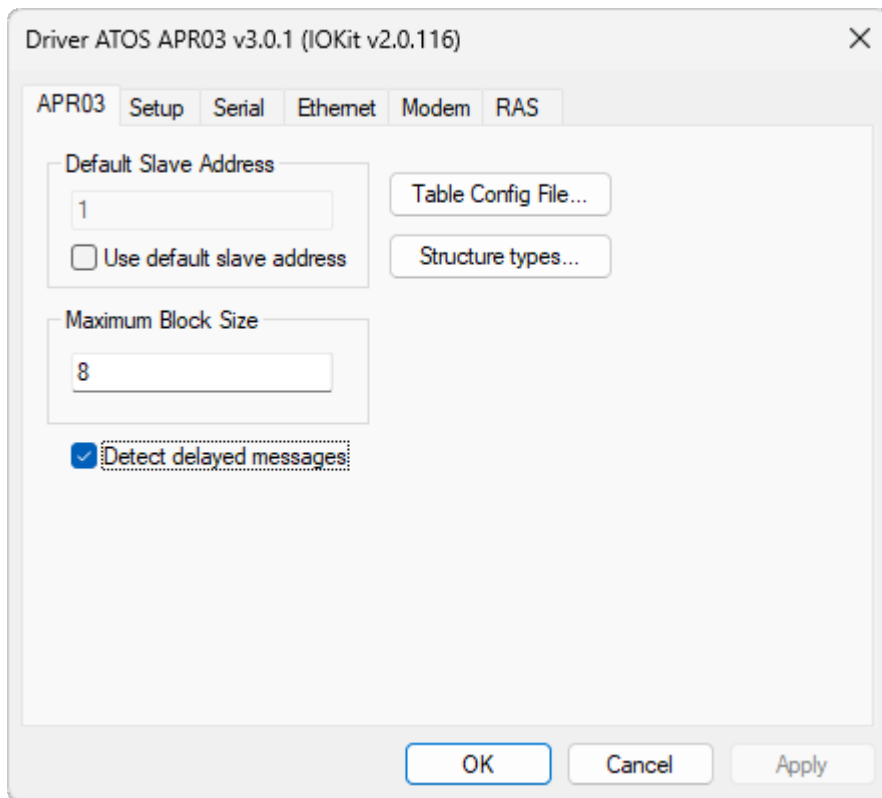
P1	Não utilizado
P2	Não utilizado
P3	Não utilizado
P4	Não utilizado

Este Driver não utiliza os parâmetros de configuração **[P]**. Todas as configurações deste Driver devem ser realizadas na janela de configuração.

A aba **APR03** da janela de configuração contém configurações específicas para este Driver. As outras abas permitem configurar os parâmetros de comunicação da biblioteca **IOKit**. Para mais informações sobre a configuração da biblioteca **IOKit**, consulte o tópico **Documentação das Interfaces de Comunicação**.

Aba APR03

Para configurar este Driver, utilize a aba **APR03**, mostrada na figura a seguir.



Aba APR03

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

Opções disponíveis na aba APR03

OPÇÃO	DESCRIÇÃO
Default Slave Address	Indica o valor do endereço escravo padrão. Esta opção é habilitada somente se a opção Use default slave address está selecionada
Use default slave address	Habilita o uso do valor definido na opção Default Slave Address como endereço de destino de um comando para todos os Tags, independente do valor configurado nos parâmetros <i>N1</i> ou <i>B1</i>
Maximum Block Size	Esta opção é usada apenas no Elipse E3 , Elipse Power ou Elipse Water se a propriedade EnableReadGrouping de um Tag (superblocos) está habilitada. Informa o número máximo de bytes de dados em cada bloco de leitura. Os valores adequados para cada modelo de equipamento estão descritos na tabela Limites de cada Bloco de Comunicação
Detect delayed messages	Esta opção permite que este Driver não retorne erro ao detectar uma resposta em atraso. Consulte a nota a seguir para mais informações sobre esta opção
Table Config File	Abre um arquivo de configuração de tabelas para leitura de dados por evento
Structure types	Abre um arquivo de configuração para tipos de dados Struct

Limites de cada Bloco de Comunicação

MODELO	LIMITE
Equipamentos da linha MPC1200, MPC1600, MPC2002, MPC4004.01, MPC4004.02, MPC4004.04, MPC4004.09, MPC4004.11, MPC4004.12, MPC4004.11L, MPC4004.12L, EXPERT e todos os controladores da linha MPC2200	8 (oito) bytes
Equipamentos MPC4004.04B, MPC4004.05B, MPC4004.06B, MPC4004.09B, MPC4004.05E, MPC4004.06E, MPC4004.09E, MPC4004.05R, MPC4004.06R, MPC4004.09R, MPC4004.05T, MPC4004.06T, MPC4004.09T e módulo 4004.72	16 bytes
Módulo 4004.72R	32 bytes

NOTA

A opção **Detect delayed messages** é útil quando o protocolo TCP/IP (Ethernet) é usado como camada de comunicação física (*Physical Layer*) da biblioteca **IOKit**. Neste caso, é comum ocorrerem atrasos não previstos na resposta de um equipamento, principalmente se a ligação física com um equipamento incluir *links* de rádio. Assim, é possível que este Driver envie uma primeira requisição e não receba a resposta no tempo esperado, gerando erro de *time-out*. Ao enviar uma nova requisição, este Driver pode receber a resposta atrasada da requisição anterior. Se esta opção não está habilitada, é possível que ocorra uma sincronização a partir daí, e assim todas as requisições subsequentes obtêm como resposta o retorno atrasado da requisição anterior. Quando esta opção está habilitada, este Driver não retorna erro imediatamente após detectar uma resposta em atraso, e sim executa uma nova tentativa de obter a resposta correta. Se uma nova mensagem é recebida antes de ocorrer um *time-out*, e a resposta é válida, esta é retornada à aplicação. Caso contrário, este Driver passa a descartar todos os bytes recebidos até a ocorrência de *time-out*, limpando assim o canal de comunicação e prevenindo a propagação do erro.

Arquivo de Configuração para Leitura de Dados por Evento

Para editar um arquivo de configuração para leitura de dados por evento, clique em **Table Config File** na aba **APR03**. Neste arquivo é possível definir os endereços das variáveis e ponteiros para o controle da leitura das tabelas de dados, bem como o formato de cada tabela. Os campos que devem ser definidos neste arquivo estão descritos na tabela a seguir.

Campos de um arquivo de configuração

CAMPO	DESCRIÇÃO
RECORD POINTER ADDRESS	Valor do endereço do ponteiro de gravação, em decimal
TABLE STATUS ADDRESS	Valor do endereço do status da tabela, em decimal
ACQUISITION STATUS ADDRESS	Valor do endereço do status da aquisição, em decimal
TABLE	Número do formato de tabela a ser definido, entre 0 (zero) e 65535. Não é necessário usar dois pontos entre a palavra-chave e o número
REGISTERS	Número máximo de registros ou estruturas que a tabela deve suportar no <i>buffer</i> . Este valor define o tamanho do <i>buffer</i> circular reservado para a tabela
START ADDRESS	Endereço inicial da tabela, em decimal, em que o <i>buffer</i> circular reservado para a tabela inicia

CAMPO	DESCRIÇÃO
PACKET SIZE	Tamanho máximo do bloco de dados que pode ser transferido a cada comunicação com um equipamento, em bytes. Este valor depende do equipamento utilizado. Quanto maior, menos comunicações são necessárias para a leitura de todos os dados
STRUCT	Define a estrutura de cada registro da tabela. Os membros devem ser definidos por valores inteiros equivalendo aos tipos de dados interpretados por este Driver

Pode-se adicionar comentários de linha a um arquivo usando os caracteres //, conforme o padrão utilizado para a linguagem C. O tópico **Leitura de Dados por Evento** analisa este recurso em detalhes.

Arquivo de Configuração para Tipos de Dados Struct

Para editar tipos de dados **Struct**, ou seja, os parâmetros *N2* ou *B2* igual a 20, clique em **Structure types** na aba **APR03**. A declaração das estruturas deve respeitar a sintaxe a seguir.

```
STRUCT TYPE
{
  Elemento1,
  Elemento2,
  Elemento3,
  [...]
}
```

Os campos que devem ser definidos em um arquivo estão descritos na tabela a seguir.

Campos de um arquivo de configuração

CAMPO	DESCRIÇÃO
TYPE	Define o número do tipo de dados Struct . Nos Tags de Comunicação, este valor é usado nos parâmetros <i>N4</i> ou <i>B4</i>
Elemento	Os elementos devem aparecer separados por vírgulas e entre chaves, após a definição do campo TYPE , e definem a estrutura de cada registro da tabela. Os membros devem ser definidos por valores inteiros equivalentes aos tipos de dados interpretados por este Driver . Os tipos de dados String , ou seja, com o parâmetro <i>N2</i> igual a 19, não são aceitos como membros de estruturas

A leitura dos tipos de dados **Struct** deve ser realizada por um Tag Bloco em que cada Elemento de Bloco corresponde a um membro da estrutura.

Além da declaração de estruturas, este arquivo pode conter também comentários, precedidos pelos caracteres //, conforme o padrão utilizado para a linguagem C. O código a seguir contém um exemplo de estrutura.

```
// Estrutura acessada por Tags com os parâmetros
// N2/B2 igual a 20 e N4/B4 igual a 1 (um, type)
STRUCT 1
{
  1, // WORDBCDLH
  0, // WORDBCDHL
  3, // WORDHEXAHL
}
```

Referência de Tags

Esta seção contém informações sobre os **parâmetros [N/B] de endereçamento para Tags PLC e Bloco**, bem como sobre a **leitura de dados por evento**.

Parâmetros [N/B] de Endereçamento Para Tags PLC e Bloco

N1/B1	Endereço de um equipamento
N2/B2	Tipo de dados da variável
N3/B3	Endereço absoluto, em decimal
N4/B4	Não utilizado

Através dos Tags deste Driver é possível ler e escrever em praticamente todos os tipos de variáveis dos equipamentos Atos.

Tipos de Dados Interpretados por este Driver

Tipos de dados

N2/B2	TIPO DE DADOS	FAIXA DE VALORES	BYTES
0	Word BCD (Hi - Lo)	Entre 0 (zero) e 9999	2 (dois)
1	Word BCD (Lo - Hi)	Entre 0 (zero) e 9999	2 (dois)
2	Word Hexa (Lo - Hi)	Entre 0 (zero) e 65535	2 (dois)
3	Word Hexa (Hi - Lo)	Entre 0 (zero) e 65535	2 (dois)
4	Byte	Entre 0 (zero) e 255	1 (um)
5	Word Modificado (Hi × 100 + Lo)	Entre 0 (zero) e 25755	2 (dois)
6	Byte Digital , ou seja, o byte PLC XXXXXXX0 igual a 1 (um) ou o byte PLC XXXXXXX1 igual a 0 (zero)	0 (zero) ou 1 (um)	1 (um)
7	Word Negado , ou seja, cada bit 0 (zero) é igual a 1 (um) e cada bit 1 (um) é igual a 0 (zero)	Entre 0 (zero) e 65535	2 (dois)
8	Byte Digital com escrita do último bit, ou seja, 1 (um) é igual ao byte PLC XXXXXXX0 ou 0 (zero) é igual ao byte PLC XXXXXXX1	0 (zero) ou 1(um)	1 (um)

N2/B2	TIPO DE DADOS	FAIXA DE VALORES	BYTES
9	DWord BCD , WHi (Hi - Lo) - WLo (Hi - Lo)	Entre 0 (zero) e 99999999	4 (quatro)
10	DWord BCD , WLo (Lo - Hi) - WHi (Lo - Hi)	Entre 0 (zero) e 99999999	4 (quatro)
11	DWord Hexa , WLo (Lo - Hi) - WHi (Lo - Hi)	Entre 0 (zero) e 4294967295	4 (quatro)
12	DWord Hexa , WHi (Hi - Lo) - WLo (Hi - Lo)	Entre 0 (zero) e 4294967295	4 (quatro)
13	Ponto Flutuante ANSI/IEEE 754 - 1985	-	4 (quatro)
14	Data e Hora no formato UTC (<i>Coordinated Universal Time</i>) de 32 bits, ou seja, o número de segundos desde 01/01/1970	-	4 (quatro)
15	INT16 Hexa (Lo - Hi)	Entre -32768 e 32767	2 (dois)
16	INT16 Hexa (Hi - Lo)	Entre -32768 e 32767	2 (dois)
17	INT32 Hexa , WLo (Lo - Hi) - WHi (Lo - Hi)	Entre -2147483648 e 2147483647	4 (quatro)
18	INT32 Hexa , WHi (Hi - Lo) - WLo (Hi - Lo)	Entre -2147483648 e 2147483647	4 (quatro)
19	String terminada em 0 (zero)	Até 256 caracteres, incluindo o caractere nulo	-
20	Struct	-	-

NOTA

Hi: Byte mais significativo, **Lo:** Byte menos significativo, **WHi:** **Word** mais significativo, **WLo:** **Word** menos significativo.

Word BCD

Leitura e Escrita

N1/B1	Endereço de um equipamento
N2/B2	0 (zero)
N3/B3	Endereço absoluto
N4/B4	Não utilizado

Este Tag lê ou escreve uma palavra de 16 bits no formato **BCD**, em que o primeiro byte é o mais significativo.

- Em equipamentos da linha MPC1200, MPC1600, MPC2002, MPC4004.01, MPC4004.02, MPC4004.04, MPC4004.09, MPC4004.11, MPC4004.12, MPC4004.11L, MPC4004.12L e EXPERT e em todos os controladores da linha MPC2200, o número máximo de Elementos deste Tag é 4 (quatro)

- Em equipamentos MPC4004.04B, MPC4004.05B, MPC4004.06B, MPC4004.09B, MPC4004.05E, MPC4004.06E, MPC4004.09E, MPC4004.05R, MPC4004.06R, MPC4004.09R, MPC4004.05T, MPC4004.06T e MPC4004.09T e no módulo 4004.72, o número máximo de Elementos deste Tag é 8 (oito)
- Em módulos 4004.72R, o número máximo de Elementos deste Tag é 16
- A faixa de valores varia entre 0 (zero) e 9999

Word BCD Inverso

Leitura e Escrita

N1/B1	Endereço de um equipamento
N2/B2	1 (um)
N3/B3	Endereço absoluto
N4/B4	Não utilizado

Este Tag lê ou escreve uma palavra de 16 bits inversa no formato **BCD**, em que o primeiro byte é o menos significativo.

- Em equipamentos da linha MPC1200, MPC1600, MPC2002, MPC4004.01, MPC4004.02, MPC4004.04, MPC4004.09, MPC4004.11, MPC4004.12, MPC4004.11L, MPC4004.12L e EXPERT e em todos os controladores da linha MPC2200, o número máximo de Elementos deste Tag é 4 (quatro)
- Em equipamentos MPC4004.04B, MPC4004.05B, MPC4004.06B, MPC4004.09B, MPC4004.05E, MPC4004.06E, MPC4004.09E, MPC4004.05R, MPC4004.06R, MPC4004.09R, MPC4004.05T, MPC4004.06T e MPC4004.09T e no módulo 4004.72, o número máximo de Elementos deste Tag é 8 (oito)
- Em módulos 4004.72R, o número máximo de Elementos deste Tag é 16
- A faixa de valores varia entre 0 (zero) e 9999

Word Hexadecimal Inverso

Leitura e Escrita

N1/B1	Endereço de um equipamento
N2/B2	2 (dois)
N3/B3	Endereço absoluto
N4/B4	Não utilizado

Este Tag lê ou escreve uma palavra de 16 bits inversa, em que o primeiro byte é o menos significativo.

- Em equipamentos da linha MPC1200, MPC1600, MPC2002, MPC4004.01, MPC4004.02, MPC4004.04, MPC4004.09, MPC4004.11, MPC4004.12, MPC4004.11L, MPC4004.12L e EXPERT e em todos os controladores da linha MPC2200, o número máximo de Elementos deste Tag é 4 (quatro)

- Em equipamentos MPC4004.04B, MPC4004.05B, MPC4004.06B, MPC4004.09B, MPC4004.05E, MPC4004.06E, MPC4004.09E, MPC4004.05R, MPC4004.06R, MPC4004.09R, MPC4004.05T, MPC4004.06T e MPC4004.09T e no módulo 4004.72, o número máximo de Elementos deste Tag é 8 (oito)
- Em módulos 4004.72R, o número máximo de Elementos deste Tag é 16
- A faixa de valores varia entre 0 (zero) e 65535

Word Hexadecimal

Leitura e Escrita

N1/B1	Endereço de um equipamento
N2/B2	3 (três)
N3/B3	Endereço absoluto
N4/B4	Não utilizado

Este Tag lê ou escreve uma palavra de 16 bits, em que o primeiro byte é o mais significativo.

- Em equipamentos da linha MPC1200, MPC1600, MPC2002, MPC4004.01, MPC4004.02, MPC4004.04, MPC4004.09, MPC4004.11, MPC4004.12, MPC4004.11L, MPC4004.12L e EXPERT e em todos os controladores da linha MPC2200, o número máximo de Elementos deste Tag é 4 (quatro)
- Em equipamentos MPC4004.04B, MPC4004.05B, MPC4004.06B, MPC4004.09B, MPC4004.05E, MPC4004.06E, MPC4004.09E, MPC4004.05R, MPC4004.06R, MPC4004.09R, MPC4004.05T, MPC4004.06T e MPC4004.09T e no módulo 4004.72, o número máximo de Elementos deste Tag é 8 (oito)
- Em módulos 4004.72R, o número máximo de Elementos deste Tag é 16
- A faixa de valores varia entre 0 (zero) e 65535

Byte

Leitura e Escrita

N1/B1	Endereço de um equipamento
N2/B2	4 (quatro)
N3/B3	Endereço absoluto
N4/B4	Não utilizado

Este Tag lê ou escreve uma palavra de 8 (oito) bits.

- Em equipamentos da linha MPC1200, MPC1600, MPC2002, MPC4004.01, MPC4004.02, MPC4004.04, MPC4004.09, MPC4004.11, MPC4004.12, MPC4004.11L, MPC4004.12L e EXPERT e em todos os controladores da linha MPC2200, o número máximo de Elementos deste Tag é 8 (oito)

- Em equipamentos MPC4004.04B, MPC4004.05B, MPC4004.06B, MPC4004.09B, MPC4004.05E, MPC4004.06E, MPC4004.09E, MPC4004.05R, MPC4004.06R, MPC4004.09R, MPC4004.05T, MPC4004.06T e MPC4004.09T e no módulo 4004.72, o número máximo de Elementos deste Tag é 16
- Em módulos 4004.72R, o número máximo de Elementos deste Tag é 32
- A faixa de valores varia entre 0 (zero) e 255

Word Hexadecimal Modificado

Leitura e Escrita

N1/B1	Endereço de um equipamento
N2/B2	5 (cinco)
N3/B3	Endereço absoluto
N4/B4	Não utilizado

Este Tag lê ou escreve uma palavra de 16 bits, em que o primeiro byte mais significativo é multiplicado por 100 e somado ao segundo byte.

- Em equipamentos da linha MPC1200, MPC1600, MPC2002, MPC4004.01, MPC4004.02, MPC4004.04, MPC4004.09, MPC4004.11, MPC4004.12, MPC4004.11L, MPC4004.12L e EXPERT e em todos os controladores da linha MPC2200, o número máximo de Elementos deste Tag é 4 (quatro)
- Em equipamentos MPC4004.04B, MPC4004.05B, MPC4004.06B, MPC4004.09B, MPC4004.05E, MPC4004.06E, MPC4004.09E, MPC4004.05R, MPC4004.06R, MPC4004.09R, MPC4004.05T, MPC4004.06T e MPC4004.09T e no módulo 4004.72, o número máximo de Elementos deste Tag é 8 (oito)
- Em módulos 4004.72R, o número máximo de Elementos deste Tag é 16
- A faixa de valores varia entre 0 (zero) e 25755

Byte Digital

Leitura e Escrita

N1/B1	Endereço de um equipamento
N2/B2	6 (seis)
N3/B3	Endereço absoluto
N4/B4	Não utilizado

Este Tag lê ou escreve um valor digital, ou seja, 0 (zero) ou 1 (um). Na leitura, se o bit 0 (zero) do byte em um equipamento é igual a 1 (um), o valor devolvido é 0 (zero). Se o bit 0 (zero) do byte em um equipamento é igual a 0 (zero), o valor devolvido é 1 (um). Já na escrita, se o valor escrito é 0 (zero), o valor escrito é FFh. Se o valor escrito é diferente de 0 (zero), o valor escrito é FEh.

- Em equipamentos da linha MPC1200, MPC1600, MPC2002, MPC4004.01, MPC4004.02, MPC4004.04, MPC4004.09, MPC4004.11, MPC4004.12, MPC4004.11L, MPC4004.12L e EXPERT e em todos os controladores da linha MPC2200, o número máximo de Elementos deste Tag é 8 (oito)
- Em equipamentos MPC4004.04B, MPC4004.05B, MPC4004.06B, MPC4004.09B, MPC4004.05E, MPC4004.06E, MPC4004.09E, MPC4004.05R, MPC4004.06R, MPC4004.09R, MPC4004.05T, MPC4004.06T e MPC4004.09T e no módulo 4004.72, o número máximo de Elementos deste Tag é 16
- Em módulos 4004.72R, o número máximo de Elementos deste Tag é 32
- Possíveis valores são 0 (zero) ou 1 (um)

Word Hexadecimal Negado

Leitura e Escrita

N1/B1	Endereço de um equipamento
N2/B2	7 (sete)
N3/B3	Endereço absoluto
N4/B4	Não utilizado

Este Tag lê ou escreve uma palavra de 16 bits com bits negados, em que o primeiro byte é o mais significativo.

- Em equipamentos da linha MPC1200, MPC1600, MPC2002, MPC4004.01, MPC4004.02, MPC4004.04, MPC4004.09, MPC4004.11, MPC4004.12, MPC4004.11L, MPC4004.12L e EXPERT e em todos os controladores da linha MPC2200, o número máximo de Elementos deste Tag é 4 (quatro)
- Em equipamentos MPC4004.04B, MPC4004.05B, MPC4004.06B, MPC4004.09B, MPC4004.05E, MPC4004.06E, MPC4004.09E, MPC4004.05R, MPC4004.06R, MPC4004.09R, MPC4004.05T, MPC4004.06T e MPC4004.09T e no módulo 4004.72, o número máximo de Elementos deste Tag é 8 (oito)
- Em módulos 4004.72R, o número máximo de Elementos deste Tag é 16
- A faixa de valores varia entre 0 (zero) e 65535

Byte Digital com Escrita no Último Bit

Leitura e Escrita

N1/B1	Endereço de um equipamento
N2/B2	8 (oito)
N3/B3	Endereço absoluto
N4/B4	Não utilizado

Este Tag lê ou escreve um valor digital, ou seja, 0 (zero) ou 1 (um). Este Tag funciona quase da mesma forma que o Tag **Byte Digital**, com a diferença que, na escrita, este Tag altera apenas o último bit dentro do byte. Para isto, este Driver

realiza uma leitura antes da operação de escrita, mascarando o último bit com o valor adequado e reescrevendo o byte mascarado.

Na leitura, se o bit 0 (zero) do byte em um equipamento é igual a 1 (um), o valor devolvido é 0 (zero). Se o bit 0 (zero) do byte em um equipamento é igual a 0 (zero), o valor devolvido é 1 (um). Já na escrita, se o valor escrito é 0 (zero), o bit menos significativo do byte em um equipamento é configurado. Se o valor escrito é diferente de 0 (zero), o bit menos significativo do byte em um equipamento é zerado.

- Em equipamentos da linha MPC1200, MPC1600, MPC2002, MPC4004.01, MPC4004.02, MPC4004.04, MPC4004.09, MPC4004.11, MPC4004.12, MPC4004.11L, MPC4004.12L e EXPERT e em todos os controladores da linha MPC2200, o número máximo de Elementos deste Tag é 8 (oito)
- Em equipamentos MPC4004.04B, MPC4004.05B, MPC4004.06B, MPC4004.09B, MPC4004.05E, MPC4004.06E, MPC4004.09E, MPC4004.05R, MPC4004.06R, MPC4004.09R, MPC4004.05T, MPC4004.06T e MPC4004.09T e no módulo 4004.72, o número máximo de Elementos deste Tag é 16
- Em módulos 4004.72R, o número máximo de Elementos deste Tag é 32
- Possíveis valores são 0 (zero) ou 1 (um)

DWord BCD

Leitura e Escrita

N1/B1	Endereço de um equipamento
N2/B2	9 (nove)
N3/B3	Endereço absoluto
N4/B4	Não utilizado

Este Tag lê ou escreve uma palavra de 32 bits no formato **BCD**, em que o primeiro byte é o mais significativo.

- Em equipamentos da linha MPC1200, MPC1600, MPC2002, MPC4004.01, MPC4004.02, MPC4004.04, MPC4004.09, MPC4004.11, MPC4004.12, MPC4004.11L, MPC4004.12L e EXPERT e em todos os controladores da linha MPC2200, o número máximo de Elementos deste Tag é 2 (dois)
- Em equipamentos MPC4004.04B, MPC4004.05B, MPC4004.06B, MPC4004.09B, MPC4004.05E, MPC4004.06E, MPC4004.09E, MPC4004.05R, MPC4004.06R, MPC4004.09R, MPC4004.05T, MPC4004.06T e MPC4004.09T e no módulo 4004.72, o número máximo de Elementos deste Tag é 4 (quatro)
- Em módulos 4004.72R, o número máximo de Elementos deste Tag é 8 (oito)
- A faixa de valores varia entre 0 (zero) e 99999999

DWord BCD Inverso

Leitura e Escrita

N1/B1	Endereço de um equipamento
N2/B2	10
N3/B3	Endereço absoluto
N4/B4	Não utilizado

Este Tag lê ou escreve uma palavra de 32 bits no formato **BCD**, em que o primeiro byte é o menos significativo.

- Em equipamentos da linha MPC1200, MPC1600, MPC2002, MPC4004.01, MPC4004.02, MPC4004.04, MPC4004.09, MPC4004.11, MPC4004.12, MPC4004.11L, MPC4004.12L e EXPERT e em todos os controladores da linha MPC2200, o número máximo de Elementos deste Tag é 2 (dois)
- Em equipamentos MPC4004.04B, MPC4004.05B, MPC4004.06B, MPC4004.09B, MPC4004.05E, MPC4004.06E, MPC4004.09E, MPC4004.05R, MPC4004.06R, MPC4004.09R, MPC4004.05T, MPC4004.06T e MPC4004.09T e no módulo 4004.72, o número máximo de Elementos deste Tag é 4 (quatro)
- Em módulos 4004.72R, o número máximo de Elementos deste Tag é 8 (oito)
- A faixa de valores varia entre 0 (zero) e 99999999

DWord Hexadecimal Inverso

Leitura e Escrita

N1/B1	Endereço de um equipamento
N2/B2	11
N3/B3	Endereço absoluto
N4/B4	Não utilizado

Este Tag lê ou escreve uma palavra de 32 bits, em que o primeiro byte é o menos significativo.

- Em equipamentos da linha MPC1200, MPC1600, MPC2002, MPC4004.01, MPC4004.02, MPC4004.04, MPC4004.09, MPC4004.11, MPC4004.12, MPC4004.11L, MPC4004.12L e EXPERT e em todos os controladores da linha MPC2200, o número máximo de Elementos deste Tag é 2 (dois)
- Em equipamentos MPC4004.04B, MPC4004.05B, MPC4004.06B, MPC4004.09B, MPC4004.05E, MPC4004.06E, MPC4004.09E, MPC4004.05R, MPC4004.06R, MPC4004.09R, MPC4004.05T, MPC4004.06T e MPC4004.09T e no módulo 4004.72, o número máximo de Elementos deste Tag é 4 (quatro)
- Em módulos 4004.72R, o número máximo de Elementos deste Tag é 8 (oito)
- A faixa de valores varia entre 0 (zero) e 4294967295

DWord Hexadecimal

Leitura e Escrita

N1/B1	Endereço de um equipamento
N2/B2	12
N3/B3	Endereço absoluto
N4/B4	Não utilizado

Este Tag lê ou escreve uma palavra de 32 bits, em que o primeiro byte é o mais significativo.

- Em equipamentos da linha MPC1200, MPC1600, MPC2002, MPC4004.01, MPC4004.02, MPC4004.04, MPC4004.09, MPC4004.11, MPC4004.12, MPC4004.11L, MPC4004.12L e EXPERT e em todos os controladores da linha MPC2200, o número máximo de Elementos deste Tag é 2 (dois)
- Em equipamentos MPC4004.04B, MPC4004.05B, MPC4004.06B, MPC4004.09B, MPC4004.05E, MPC4004.06E, MPC4004.09E, MPC4004.05R, MPC4004.06R, MPC4004.09R, MPC4004.05T, MPC4004.06T e MPC4004.09T e no módulo 4004.72, o número máximo de Elementos deste Tag é 4 (quatro)
- Em módulos 4004.72R, o número máximo de Elementos deste Tag é 8 (oito)
- A faixa de valores varia entre 0 (zero) e 4294967295

Ponto Flutuante ANSI/IEEE 754 - 1985

Leitura e Escrita

N1/B1	Endereço de um equipamento
N2/B2	13
N3/B3	Endereço absoluto
N4/B4	Não utilizado

Este Tag lê ou escreve um valor em ponto flutuante de 32 bits, segundo a norma **ANSI/IEEE 754 - 1985**.

- Em equipamentos da linha MPC1200, MPC1600, MPC2002, MPC4004.01, MPC4004.02, MPC4004.04, MPC4004.09, MPC4004.11, MPC4004.12, MPC4004.11L, MPC4004.12L e EXPERT e em todos os controladores da linha MPC2200, o número máximo de Elementos deste Tag é 2 (dois)
- Em equipamentos MPC4004.04B, MPC4004.05B, MPC4004.06B, MPC4004.09B, MPC4004.05E, MPC4004.06E, MPC4004.09E, MPC4004.05R, MPC4004.06R, MPC4004.09R, MPC4004.05T, MPC4004.06T e MPC4004.09T e no módulo 4004.72, o número máximo de Elementos deste Tag é 4 (quatro)
- Em módulos 4004.72R, o número máximo de Elementos deste Tag é 8 (oito)

Data e Hora

Leitura e Escrita

N1/B1	Endereço de um equipamento
N2/B2	14
N3/B3	Endereço absoluto
N4/B4	Não utilizado

Este Tag lê ou escreve um valor de data e hora de um equipamento no formato **UTC** (*Coordinated Universal Time*) de 32 bits, ou seja, o número de segundos desde primeiro de janeiro de 1970.

- Em equipamentos da linha MPC1200, MPC1600, MPC2002, MPC4004.01, MPC4004.02, MPC4004.04, MPC4004.09, MPC4004.11, MPC4004.12, MPC4004.11L, MPC4004.12L e EXPERT e em todos os controladores da linha MPC2200, o número máximo de Elementos deste Tag é 2 (dois)
- Em equipamentos MPC4004.04B, MPC4004.05B, MPC4004.06B, MPC4004.09B, MPC4004.05E, MPC4004.06E, MPC4004.09E, MPC4004.05R, MPC4004.06R, MPC4004.09R, MPC4004.05T, MPC4004.06T e MPC4004.09T e no módulo 4004.72, o número máximo de Elementos deste Tag é 4 (quatro)
- Em módulos 4004.72R, o número máximo de Elementos deste Tag é 8 (oito)

INT16 Hexadecimal Inverso

Leitura e Escrita

N1/B1	Endereço de um equipamento
N2/B2	15
N3/B3	Endereço absoluto
N4/B4	Não utilizado

Este Tag lê ou escreve uma palavra de 16 bits com sinal, em que o primeiro byte é o menos significativo.

- Em equipamentos da linha MPC1200, MPC1600, MPC2002, MPC4004.01, MPC4004.02, MPC4004.04, MPC4004.09, MPC4004.11, MPC4004.12, MPC4004.11L, MPC4004.12L e EXPERT e em todos os controladores da linha MPC2200, o número máximo de Elementos deste Tag é 4 (quatro)
- Em equipamentos MPC4004.04B, MPC4004.05B, MPC4004.06B, MPC4004.09B, MPC4004.05E, MPC4004.06E, MPC4004.09E, MPC4004.05R, MPC4004.06R, MPC4004.09R, MPC4004.05T, MPC4004.06T e MPC4004.09T e no módulo 4004.72, o número máximo de Elementos deste Tag é 8 (oito)
- Em módulos 4004.72R, o número máximo de Elementos deste Tag é 16
- A faixa de valores varia entre -32768 e 32767

INT16 Hexadecimal

Leitura e Escrita

N1/B1	Endereço de um equipamento
N2/B2	16
N3/B3	Endereço absoluto
N4/B4	Não utilizado

Este Tag lê ou escreve uma palavra de 16 bits com sinal, em que o primeiro byte é o mais significativo.

- Em equipamentos da linha MPC1200, MPC1600, MPC2002, MPC4004.01, MPC4004.02, MPC4004.04, MPC4004.09, MPC4004.11, MPC4004.12, MPC4004.11L, MPC4004.12L e EXPERT e em todos os controladores da linha MPC2200, o número máximo de Elementos deste Tag é 4 (quatro)
- Em equipamentos MPC4004.04B, MPC4004.05B, MPC4004.06B, MPC4004.09B, MPC4004.05E, MPC4004.06E, MPC4004.09E, MPC4004.05R, MPC4004.06R, MPC4004.09R, MPC4004.05T, MPC4004.06T e MPC4004.09T e no módulo 4004.72, o número máximo de Elementos deste Tag é 8 (oito)
- Em módulos 4004.72R, o número máximo de Elementos deste Tag é 16
- A faixa de valores varia entre -32768 e 32767

INT32 Hexadecimal Inverso

Leitura e Escrita

N1/B1	Endereço de um equipamento
N2/B2	17
N3/B3	Endereço absoluto
N4/B4	Não utilizado

Este Tag lê ou escreve uma palavra de 32 bits com sinal, em que o primeiro byte é o menos significativo.

- Em equipamentos da linha MPC1200, MPC1600, MPC2002, MPC4004.01, MPC4004.02, MPC4004.04, MPC4004.09, MPC4004.11, MPC4004.12, MPC4004.11L, MPC4004.12L e EXPERT e em todos os controladores da linha MPC2200, o número máximo de Elementos deste Tag é 2 (dois)
- Em equipamentos MPC4004.04B, MPC4004.05B, MPC4004.06B, MPC4004.09B, MPC4004.05E, MPC4004.06E, MPC4004.09E, MPC4004.05R, MPC4004.06R, MPC4004.09R, MPC4004.05T, MPC4004.06T e MPC4004.09T e no módulo 4004.72, o número máximo de Elementos deste Tag é 4 (quatro)
- Em módulos 4004.72R, o número máximo de Elementos deste Tag é 8 (oito)
- A faixa de valores varia entre -2147483648 até 2147483647

INT32 Hexadecimal

Leitura e Escrita

N1/B1	Endereço de um equipamento
N2/B2	18
N3/B3	Endereço absoluto
N4/B4	Não utilizado

Este Tag lê ou escreve uma palavra de 32 bits com sinal, em que o primeiro byte é o mais significativo.

- Em equipamentos da linha MPC1200, MPC1600, MPC2002, MPC4004.01, MPC4004.02, MPC4004.04, MPC4004.09, MPC4004.11, MPC4004.12, MPC4004.11L, MPC4004.12L e EXPERT e em todos os controladores da linha MPC2200, o número máximo de Elementos deste Tag é 2 (dois)
- Em equipamentos MPC4004.04B, MPC4004.05B, MPC4004.06B, MPC4004.09B, MPC4004.05E, MPC4004.06E, MPC4004.09E, MPC4004.05R, MPC4004.06R, MPC4004.09R, MPC4004.05T, MPC4004.06T e MPC4004.09T e no módulo 4004.72, o número máximo de Elementos deste Tag é 4 (quatro)
- Em módulos 4004.72R, o número máximo de Elementos deste Tag é 8 (oito)
- A faixa de valores varia entre -2147483648 até 2147483647

String

Leitura e Escrita

N1/B1	Endereço de um equipamento
N2/B2	19
N3/B3	Endereço absoluto
N4/B4	Tamanho da String

Este Tag lê ou escreve valores do tipo de dados **String** terminados em nulo no endereço especificado. Deve ser definido como um Tag PLC ou um Tag Bloco com apenas um Elemento.

Este Tag não é linear, ou seja, não é agrupável pelo recurso de Superblocos do **Eclipse E3**, **Eclipse Power** ou **Eclipse Water**.

O parâmetro *N4* ou *B4* contém o tamanho em caracteres da **String** a ser lida ou escrita, incluindo o terminador nulo. O valor máximo é de 256 caracteres.

Struct

Leitura e Escrita

N1/B1	Endereço de um equipamento
N2/B2	20
N3/B3	Endereço absoluto
N4/B4	Tipo da estrutura

Este Tag lê ou escreve valores de elementos de estrutura, conforme configurado na aba **APR03**. Os Elementos deste Tag Bloco podem ter **valores de tipos de dados diferentes**.

A leitura deste Tag só pode ser realizada em bloco e só é aceita a escrita de Elementos de Bloco. Este Tag não é linear, ou seja, não é agrupável pelo recurso de Superblocos do **Elipse E3**, **Elipse Power** ou **Elipse Water**.

O parâmetro *N4* ou *B4* indica o tipo de estrutura. Este valor referencia uma estrutura específica definida na janela de configuração, conforme descrito no tópico **Arquivo de Configuração para Tipos de Dados Struct**.

Leitura de Dados por Evento

Somente Leitura

N1/B1	Endereço de um equipamento
N2/B2	100
N3/B3	Número da tabela
N4/B4	Não utilizado

Este Tag é reportado por eventos e permite ler uma determinada tabela da memória de massa de um equipamento.

A cada *scan*, ou leitura por varredura, este Tag verifica no registrador **TABLE STATUS ADDRESS** se existem novos registros a serem lidos na tabela em uso, definida no parâmetro *N3*. Caso não existam, este Tag retorna uma lista de eventos vazia, ou seja, não retorna erro nem sucesso, e as respectivas propriedades permanecem inalteradas.

Caso existam registros novos, estes são retornados imediatamente como uma sequência de eventos. Para cada evento ou registro retornado, este Driver preenche as propriedades e valores deste Tag com os dados do evento e gera um evento **OnRead**.

A partir da versão **2.5** deste Driver, o *download* de eventos é realizado em uma *thread* adicional em segundo plano. Com isto, este Tag retorna sempre rapidamente, pois não aguarda mais a realização do *download* completo. Desta forma, este Tag pode ter um valor de varredura bem baixo, sem causar impactos negativos em termos de desempenho. A operação de leitura deste Tag apenas verifica a existência de registros previamente coletados pela *thread* de *download*. Caso estes registros existam, são retornados imediatamente. Caso não existam, este Tag verifica a existência de novos registros em um controlador. Se há registros a ler em um equipamento, uma nova requisição de *download* é agendada e este Tag retorna sem valor, ou seja, uma lista vazia. A *thread* única de *download* processa as requisições uma a uma, na ordem em que são agendadas. Para mais informações, consulte o tópico **Leitura de Dados por Evento**.

Cada Elemento de Bloco deve corresponder a um determinado elemento da estrutura da tabela associada, conforme declarado no arquivo de configuração.

A cada evento **OnRead** deste Tag, a propriedade **Timestamp** mostra a data e hora do registro atual lido de um equipamento.

Leitura de Dados por Evento

Para a leitura de dados por evento de um equipamento, é necessária a implementação e a manutenção de registradores e de uma tabela em memória, que devem ser gerenciadas e atualizadas pela aplicação do controlador. Esta aplicação deve criar um *buffer* circular, onde uma tabela de dados deve ser mantida.

A tabela de dados é formada por endereços de controle e registros contendo uma estampa de tempo e uma quantidade fixa de dados, conforme a tabela a seguir.

Formato da tabela de dados

REGISTRO	TIMESTAMP	DADO 1	DADO 2	...	DADO N
1					
2					
3					
4					
...					
N					

Além desta tabela, os registradores de controle **RECORD POINTER**, **TABLE STATUS** e **ACQUISITION STATUS** devem ser mantidos pela aplicação no equipamento. Consulte os tópicos **Parâmetros de Configuração da Tabela** e **Procedimento de Leitura de Registros da Tabela de um Equipamento** para mais informações sobre estes registradores.

Parâmetros de Configuração da Tabela

RECORD POINTER (Ponteiro de Gravação)

Endereço contendo a posição do último registro gravado na tabela. Trata-se de um índice iniciando em 1 (um), em que o valor 1 (um) indica que o último registro gravado está no início da tabela, ou seja, **START ADDRESS**.

NOTA

A gravação de dados deve ocorrer no sentido crescente.

TABLE STATUS (Status da Tabela)

Número de registros armazenados na tabela desde a última atualização.

NOTA

A aplicação deve ler a quantidade de registros apresentada neste parâmetro. A quantidade de registros lidos da tabela por uma aplicação deve ser informada ao equipamento através do endereço de controle **ACQUISITION STATUS** e, quando este número é igual ao **TABLE STATUS**, ambos são zerados pelo equipamento, indicando a conclusão da atualização.

ACQUISITION STATUS (Status da Aquisição)

Endereço informando ao controlador o número de registros lidos da tabela no ciclo atual.

NOTA

Quando o valor de **ACQUISITION STATUS** é igual ao **TABLE STATUS**, isto indica a conclusão da atualização.

REGISTERS

Varia entre 0 (zero) e o número máximo de registros.

Tipo de Dados

Deve ser selecionado para cada elemento da estrutura da tabela, correspondendo aos **Tipos de Dados Interpretados por este Driver**.

START ADDRESS (Endereço Inicial)

Endereço de memória do primeiro dado do primeiro registro.

PACKET SIZE (Número Máximo de Bytes ou Pacote)

Varia conforme a CPU utilizada.

As informações anteriores são configuradas em um arquivo na janela de configuração deste Driver. O código a seguir demonstra a estrutura de configuração a ser definida nesta janela.

```
TABLE X
RECORD POINTER ADDRESS: XXXX
TABLE STATUS ADDRESS: XXXX
ACQUISITION STATUS ADDRESS: XXXX
REGISTERS: X
START ADDRESS: X
PACKET SIZE: X
STRUCT
{
  1, // WORDBCDLH
  0, // WORDBCDHL
  3, // WORDHEX AHL
  ...
}
```

NOTA

Os valores dos endereços de registradores devem ser inseridos no formato decimal ou no formato hexadecimal com o sufixo H, como por exemplo 2C00H, que corresponde a 11264.

Procedimento de Leitura de Registros da Tabela de um Equipamento

Este Driver lê os registros em ordem decrescente, isto é, do mais recente para o mais antigo, começando pelo endereço indicado por **RECORD POINTER** e movendo-se em direção ao **START ADDRESS**. O número de registros a serem lidos sequencialmente, a partir do endereço do **RECORD POINTER**, é definido pelo valor de **TABLE STATUS**. Este Driver deve ler sequencialmente este número de registros e então verificar novamente se este número foi incrementado, lendo os registros adicionais a partir do novo **RECORD POINTER**, se necessário.

Caso o valor de **TABLE STATUS** seja superior ao tamanho máximo do pacote de dados de comunicação, **PACKET SIZE**, a leitura deve ser realizada em partes. Após cada operação de leitura de blocos de registros de um equipamento, este Driver deve atualizar o valor de **ACQUISITION STATUS** com o valor atual de registros já lidos no ciclo atual.

A tabela implementada em um equipamento é do tipo circular, ou seja, se o número de registros a serem lidos é superior à diferença entre **RECORD POINTER** e **START ADDRESS**, este Driver deve ler os registros no sentido decrescente até atingir o **START ADDRESS**, continuando a leitura no topo da pilha, no endereço resultante da soma do **START ADDRESS** mais o valor de **PACKET SIZE**.

Cada vez que o valor de **ACQUISITION STATUS** atingir o valor de **TABLE STATUS**, a aplicação do equipamento zera o conteúdo de **TABLE STATUS**, o que interrompe o ciclo de leitura atual. Este Driver deve prosseguir a varredura normal de **TABLE STATUS** a cada operação de leitura do Tag associado, aguardando a leitura de um valor não nulo para dar início a um novo ciclo de leituras.

Documentação das Interfaces de Comunicação

Esta seção contém a documentação das Interfaces de Comunicação referentes ao Driver **APR03**.

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

Setup

Physical Layer: Start driver OFFLINE

Timeout: ms Communication check time: ms

Connection management

Mode:

Retry failed connection every seconds

Give up after failed retries

Disconnect if non-responsive for seconds

Logging Options

Log to File:

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

Opções para o grupo Logging Options

OPÇÃO	DESCRIÇÃO
Log to File	<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**.

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

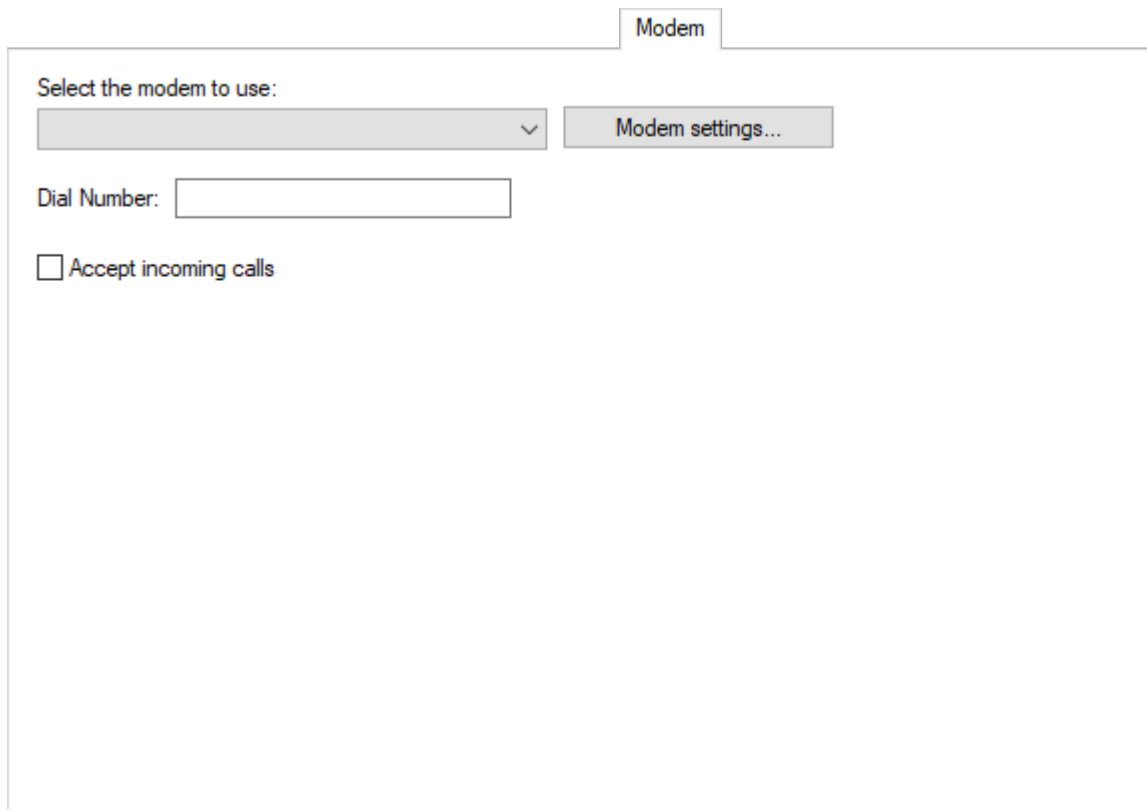
OPÇÃO	DESCRIÇÃO
IP Filter	Lista de endereços IP restringidos ou permitidos de onde um Driver aceita conexões (<i>Firewall</i>). Consulte a propriedade IO.Ethernet.IPFilter para mais informações
PING before connecting	Habilite esta opção para executar um comando ping , ou seja, para verificar se um dispositivo 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
3.0.2	01/09/2025	M. Ludwig	<ul style="list-style-type: none"> • Driver atualizado para a biblioteca IOKit versão 3.0 e Visual Studio 2022 (<i>Case 37940</i>).
3.0.1	16/03/2020	C. Mello	<ul style="list-style-type: none"> • Driver portado para a biblioteca IOKit versão 2.0 (<i>Case 27446</i>).
2.6.1	30/05/2011	A. Quites	<ul style="list-style-type: none"> • Resolvida uma vulnerabilidade, que agora rejeita <i>frames</i> inválidos mas com o byte verificador de erro correto, indicando ter o número esperado de bytes de dados mas contendo menos bytes (<i>Case 12264</i>).
2.5.1	02/08/2010	A. Quites	<ul style="list-style-type: none"> • Corrigida a forma de declaração dos tipos de dados Struct na documentação (<i>Case 10774</i>). • Separada a área de dados para otimizar e permitir a implementação de <i>threads</i> de coleta de dados com maior facilidade (<i>Case 10626</i>). • Corrigido um erro gerado pela compilação de arquivos de tabela de aplicações nas versões 2.2

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			<p>e 2.3, quando atualizadas para a versão 2.4 (Case 10690).</p> <ul style="list-style-type: none"> • Adicionada uma <i>thread</i> para permitir o <i>download</i> de tabelas de eventos em segundo plano (Case 10625).
2.4.1	08/06/2009	A. Quites	<ul style="list-style-type: none"> • Driver modificado para permitir o <i>download</i> de múltiplas tabelas de eventos (Case 10450).
2.3.1	09/10/2008	A. Quites	<ul style="list-style-type: none"> • Corrigido um erro na escrita de tipos de dados Data e Hora (Case 9847). • Implementada a detecção de respostas em atraso (Case 9843).
2.2.1	25/07/2008	A. Quites	<ul style="list-style-type: none"> • Implementado o tipo de dados Struct (Case 9257). • Implementados os recursos de Estampa de Tempo e Leitura de Memória de Massa (Case 8266). • Implementado o suporte a tipos de dados String (Case 8766). • Corrigido um erro no formato de data e hora no formato UTC no tipo de dados 14 (Case 9070). • Corrigido um erro que impedia a utilização do Tag Physical Layer Status da biblioteca IOKit, que não era reconhecido por este Driver após a implementação dos Superblocos (Case 8816).
2.1.1	17/01/2007	A. Quites	<ul style="list-style-type: none"> • Implementado o recurso de Superblocos (Case 7663).
2.0.1	11/08/2006	A. Quites C. Fehlauer	<ul style="list-style-type: none"> • Driver portado para a biblioteca IOKit (Case 3882). • Implementados os tipos de dados de 32 bits e ponto flutuante (Case 3943). • Implementado o tipo de dados Data e Hora e os tipos de dados de 16 e 32 bits com sinal (Case 6457).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
1.0.1		R. Haetinger	<ul style="list-style-type: none">Todas as publicações anteriores ao controle de revisões, sem a biblioteca IOKit.

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)