Driver Altus Alnetlv2

Nome do Arquivo	Alnetlv2.dll
Fabricante	Altus Sistemas de Automação S.A.
Equipamentos	AL-2000, AL-3000, AL-600, QUARK, PICCOLO e PONTO
Protocolo	Alnet I versão 2.0
Versão	2.4.10
Última Atualização	01/09/2025
Plataforma	Win32
Dependências	IOKit versão 1.6 ou superior
Superblocos	Sim
Nível	0

Introdução

Este é o Driver Altus Alnetlv2 para a comunicação entre sistemas da **Elipse Software** e os controladores programáveis Altus das séries AL-2000, AL-3000, AL-600, QUARK, PICCOLO e PONTO usando o protocolo Alnet I versão 2.0.

Este Driver é o resultado da unificação dos Drivers **AL2000**, **AL2000MNS**, **AL2000-Modem** e **AL2000 Timestamp**. Houve um esforço para tentar manter a compatibilidade com estas versões no que se refere às configurações dos Tags. Com exceção do Tag com o parâmetro *B2* igual a 6000, **Leitura de Bloco de Dados com Estampa de Tempo**, que no Driver original era identificado pelo parâmetro *B2* igual a 2000, todos os demais Tags seguem os mesmos padrões dos Drivers anteriores e podem funcionar sem alterações com aplicações já existentes para estas versões.

Embora este Driver tenha sido especificado para permitir comunicação em redes Alnet-I, também permite a comunicação em redes Alnet-II através de uma biblioteca do fabricante, que realiza a conversão entre estes dois protocolos.

NOTA

Este Driver suporta o recurso de Superblocos ou agrupamento de Tags, ou seja, a propriedade **EnableReadGrouping**.

Preparando um Equipamento

A comunicação pode se dar diretamente via cabo RS232 ou através de AL1413, utilizando um cabo RS485, para maiores distâncias. A configuração de um equipamento deve ser realizada antes usando o programa MasterTool do fabricante.

Este Driver pode também se comunicar com CLPs em rede Alnet II, se é usado um Gateway AL2400/S. Neste caso, é preciso especificar não apenas o endereço de um CLP nos Tags de Comunicação, mas também o endereço da sub-rede, multiplicado por 100 ao parâmetro *N1* ou *B1* de cada Tag. Para mais informações, consulte os tópicos **Parâmetros [B] de Endereçamento para Tags Bloco** e **Parâmetros [N] de Endereçamento para Tags PLC**. A configuração padrão de comunicação serial dos CLPs, configurável em alguns equipamentos, e deste Driver é a seguinte:

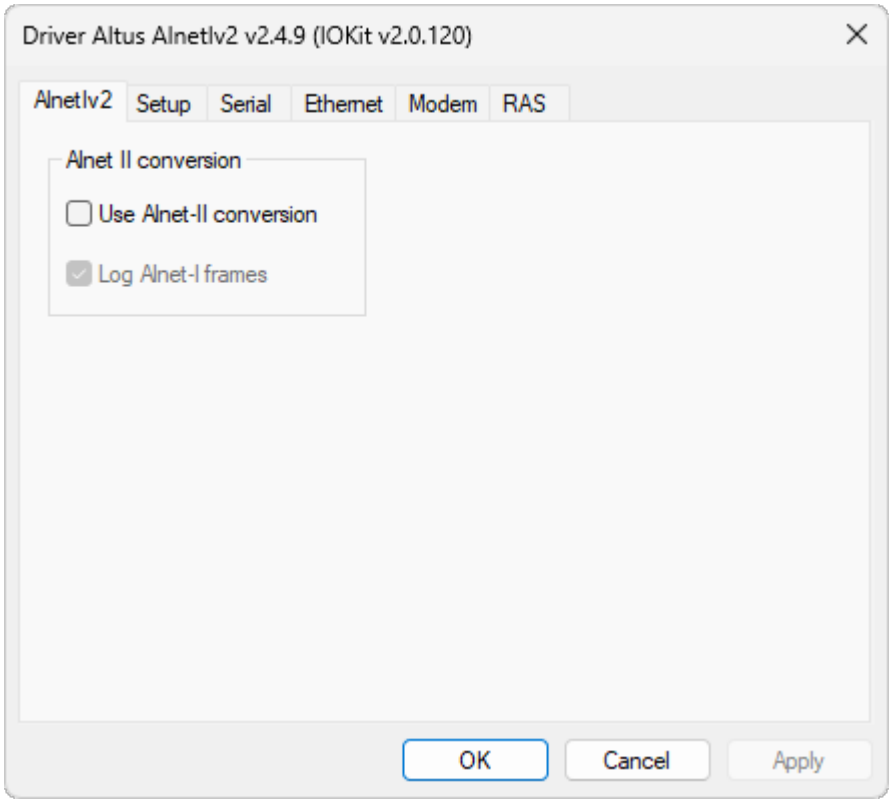
- **Velocidade de comunicação**: 9600 bps
- **Número de bits de dados por caractere**: 8 (oito)
- **Paridade**: par
- **Número de start bits**: 1 (um)

- **Número de stop bits**: 1 (um)

Configuração do Driver

Este Driver não utiliza os parâmetros **[P]**. Todas as configurações referentes à comunicação devem ser realizadas na janela de configurações deste Driver. Nesta janela pode-se encontrar uma aba específica, **AlnetIv2**, e as abas **Setup**, **Serial**, **Ethernet**, **Modem** e **RAS**, específicas da biblioteca **IOKit** da **Elipse Software**, utilizada por este Driver. Para mais informações sobre esta biblioteca, consulte o tópico **Documentação das Interfaces de Comunicação**.

Configuração das Propriedades



Aba AlnetIv2

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

Opções disponíveis na aba AlnetIv2

OPÇÃO	DESCRIÇÃO
Use Alnet-II conversion	Converte os pacotes de rede Alnet-I para pacotes de rede Alnet-II, permitindo a comunicação nestas redes
Log Alnet-I frames	Caso a opção **Use Alnet-II conversion** esteja habilitada, permite que os *frames* de pacotes de rede Alnet-I sejam visualizados nos logs antes de passarem pela camada de conversão de pacotes de rede Alnet-I para pacotes de rede Alnet-II

Referência de Tags

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

Introdução

Através dos Tags deste Driver é possível ler e escrever em operandos e tabelas de operandos de um CLP. Pode-se também acessar operandos da memória interna deste Driver, modificados por mensagens não solicitadas recebidas dos equipamentos, atribuindo-se 9999 ao parâmetro *N1* ou *B1* nos Tags que suportam mensagens não solicitadas. Este Driver, seguindo o padrão antes implementado no antigo Driver **AL2000 Timestamp**, permite acessar por mensagens não solicitadas apenas os operandos **Memória**, **Tabela de Memória**, **Octeto** e **Auxiliar**. Não é permitida a escrita de valores de mensagens não solicitadas através dos Tags. Estes valores podem apenas ser modificados por mensagens dos CLPs. Na leitura de valores de mensagens não solicitadas, a propriedade **Timestamp** de um Tag indica o momento da última modificação de um valor ou do recebimento de uma mensagem. Para mais informações sobre o recebimento de mensagens não solicitadas de um CLP, consulte o tópico **Mensagens Não Solicitadas (MNS)**.

Os Tags de leitura e escrita de operandos podem ser configurados usando parâmetros *N* e *B* como nas versões anteriores, ou alternativamente, se este Driver é usado com o **Elipse E3**, **Elipse Power** ou **Elipse Water**, através de **Strings**. A configuração por **Strings** segue o mesmo padrão usado nos Drivers OPC, através das propriedades **Dispositivo** e **Item** dos Tags do **Elipse E3**, **Elipse Power** ou **Elipse Water**. Para mais informações, consulte o tópico **Configuração por Strings**.

Parâmetros [N] de Endereçamento para Tags PLC

N1	Número da rede × 100 + número de um CLP. Consulte a **nota** a seguir para mais informações
N2	Tipo de dados. Consulte a tabela **Tipos de dados suportados para Tags PLC**. Some 10000 para inteiros sem sinal, ou seja, os tipos de dados **Memória** e **Tabela de Memória**
N3	Endereço. Consulte a **nota** a seguir para mais informações
N4	Bit ou posição de tabela

NOTA

Alguns endereços estão expressos em octal em um CLP. Para a leitura correta destes valores nas aplicações da **Elipse Software**, os endereços devem ser convertidos para decimal.

A tabela a seguir contém uma síntese dos tipos de dados que podem ser lidos com estes Tags e a respectiva configuração, que são descritos nos tópicos a seguir.

Tipos de dados suportados para Tags PLC

TIPO DE DADOS	MNEMÔNICO	N2	N3	N4	MODO
Memória	M<end>	0 (zero)	Endereço	**0**: Valor, **1-8**: Bit	Leitura ou escrita
Tabela de Memória	TM<end>:<pos>	1 (um)	Endereço	Posição	Leitura ou escrita
Bit de Memória	bit <n> de M<end>	2 (dois)	Endereço	*n*	Leitura ou escrita
Decimal	D<end>	3 (três)	Endereço	**0**: Valor, **1-8**: Bit	Leitura ou escrita
Tabela de Decimais	TD<end>:<pos>	4 (quatro)	Endereço	Posição	Leitura ou escrita
Auxiliar	A<end>	5 (cinco)	Endereço	**0**: Valor, **1-8**: Bit	Leitura ou escrita

TIPO DE DADOS	MNEMÔNICO	N2	N3	N4	MODO
Octeto de Entrada e Saída	E<end> e S<end>	6 (seis)	Endereço	**0**: Valor, **1-8**: Bit	Leitura ou escrita
Inteiro de 32 Bits	I<end>	7 (sete)	Endereço	**0**: Valor, **1-8**: Bit	Leitura ou escrita
Bit em Octeto	E<end> e S<end>	8 (oito)	Endereço	Bit (0-7)	Escrita
Bit em Auxiliar	A<end>	9 (nove)	Endereço	Bit (0-7)	Escrita
Real ou Float	F<end>	10	Endereço	**0**: Valor, **1-8**: Bit	Leitura ou escrita
Tabela de Real ou Float	TF<end>:<pos>	11	Endereço	Posição	Leitura ou escrita
Tabela de Inteiro de 32 Bits	TI<end>:<pos>	12	Endereço	Posição	Leitura ou escrita
Leitura de Dados com Estampa de Tempo		13	Endereço da tabela decimal	Endereço auxiliar	Leitura
Libera Todos os Forçamentos	-	200	-	-	Escrita

Memória

Leitura e Escrita

N1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP ou 9999 para MNS
N2	0 (zero)
N3	Endereço
N4	Bit. Os valores possíveis são **0**: Lê o valor de um operando inteiro ou maior que 0 (zero) para ler apenas o bit especificado. O bit menos significativo é o de número 1 (um)

Este Tag lê e escreve operandos de tipo **Memória**.

Tabela de Memória

Leitura e Escrita

N1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP ou 9999 para MNS
N2	1 (um)
N3	Tabela
N4	Posição

Este Tag lê e escreve operandos de tipo **Tabela de Memória**.

Bit de Memória

Leitura e Escrita

N1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP ou 9999 para MNS
N2	2 (dois)
N3	Endereço
N4	Bit a ser acessado. O bit menos significativo é o de número 0 (zero)

Este Tag lê e escreve em bits de operandos do tipo **Bit de Memória**.

NOTA

Este Tag foi mantido por compatibilidade e deve-se evitar usá-lo em aplicações recentes. As aplicações devem utilizar o Tag de acesso a operandos de tipo **Memória** com o parâmetro *N4* maior que 0 (zero).

Decimal

Leitura e Escrita

N1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP
N2	3 (três)
N3	Endereço
N4	Bit. Os valores possíveis são **0**: Lê o valor de um operando inteiro ou maior que 0 (zero) para ler apenas o bit especificado. O bit menos significativo é o de número 1 (um)

Este Tag lê e escreve operandos de tipo **Decimal**.

Tabela de Decimais

Leitura e Escrita

N1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP
N2	4 (quatro)
N3	Tabela
N4	Posição

Este Tag lê e escreve em tabelas de decimais.

Auxiliar

Leitura e Escrita

N1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP ou 9999 para MNS
N2	5 (cinco)
N3	Endereço
N4	Bit. Os valores possíveis são **0**: Lê o valor de um operando inteiro ou maior que 0 (zero) para ler apenas o bit especificado. O bit menos significativo é o de número 1 (um)

Este Tag lê e escreve operandos de tipo **Auxiliar**.

Octeto de Entrada e Saída

Leitura e Escrita

N1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP ou 9999 para MNS
N2	6 (seis)
N3	Endereço
N4	Bit. Os valores possíveis são **0**: Lê o valor de um operando inteiro ou maior que 0 (zero) para ler apenas o bit especificado. O bit menos significativo é o de número 1 (um)

Este Tag respectivamente lê e escreve em octetos de entradas e saídas digitais.

Inteiro de 32 Bits

Leitura e Escrita

N1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP
N2	7 (sete)
N3	Endereço
N4	Bit. Os valores possíveis são **0**: Lê o valor de um operando inteiro ou maior que 0 (zero) para ler apenas o bit especificado. O bit menos significativo é o de número 1 (um)

Este Tag respectivamente lê e escreve em operandos de tipo **Inteiro de 32 bits**.

Bit em Octeto

Somente Escrita

N1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP
N2	8 (oito)
N3	Endereço
N4	Bit a ser acessado. O bit menos significativo é o de número 0 (zero)

Este Tag escreve em bits de octetos.

NOTA

Este Tag foi mantido por compatibilidade e deve-se evitar usá-lo em aplicações recentes. As aplicações devem utilizar o Tag de acesso a operandos de **Entrada e Saída** com o parâmetro *N4* maior que 0 (zero).

Bit em Auxiliar

Somente Escrita

N1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP
N2	9 (nove)
N3	Endereço
N4	Bit a ser acessado. O bit menos significativo é o de número 0 (zero)

Este Tag respectivamente lê e escreve em bits de operandos de tipo **Auxiliar**.

NOTA

Este Tag foi mantido por compatibilidade e deve-se evitar usá-lo em aplicações recentes. As aplicações devem utilizar o Tag de acesso a operandos de tipo **Auxiliar** com o parâmetro *N4* maior que 0 (zero).

Real ou Float

Leitura e Escrita

N1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP
N2	10
N3	Endereço
N4	Bit. Os valores possíveis são **0**: Lê o valor de um operando inteiro ou maior que 0 (zero) para ler apenas o bit especificado. O bit menos significativo é o de número 1 (um)

Este Tag respectivamente lê e escreve em operandos de tipo **Real** ou **Float**.

NOTA

Para valores no formato **Float** com números não normalizados, assume-se o valor 0 (zero) para os casos de *underflow* e assume-se o valor **null** para os casos de *overflow*.

Tabela de Real ou Float

Leitura e Escrita

N1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP
N2	11
N3	Tabela
N4	Posição

Este Tag lê e escreve em tabelas de **Real** ou **Float**.

NOTA

Para valores no formato **Float** com números não normalizados, assume-se o valor 0 (zero) para os casos de *underflow* e assume-se o valor **null** para os casos de *overflow*.

Tabela de Inteiro de 32 Bits

Leitura e Escrita

N1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP
N2	12
N3	Tabela
N4	Posição

Este Tag lê e escreve em tabelas de **Inteiros de 32 bits**.

Leitura de Dados com Estampa de Tempo

Leitura

N1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP
N2	13
N3	Endereço de uma tabela de leitura decimal
N4	Endereço auxiliar

Este Tag lê variáveis de tabelas de decimais com estampa de tempo, com intervalo constante entre variáveis. Para mais informações sobre a utilização deste recurso, consulte o tópico **Leitura de Dados com Estampa de Tempo**.

Parâmetros [B] de Endereçamento para Tags Bloco

A tabela a seguir contém os tipos de dados que podem ser lidos com estes Tags e a respectiva configuração, que são descritos nos tópicos a seguir.

Tipos de dados suportados para Tags Bloco

TIPO DE DADOS	B2	B3	B4	MODO
Tabela de Memória	Número da tabela	Posição inicial da tabela	-	Leitura ou escrita
Tabela de Decimal	1000 + número da tabela	Posição inicial da tabela	-	Leitura ou escrita
Status de Barramentos de Entrada e Saída	2000	Número do barramento	-	Leitura
Status de Comunicação	2001	*Reset* de variáveis	-	Leitura
Status do Equipamento	2002	-	-	Leitura
Bloco de Memória	3000	Endereço inicial	-	Leitura ou escrita
Tabela de Real ou Float	4000 + número da tabela	Posição inicial da tabela	-	Leitura ou escrita
Tabela de Inteiro de 32 Bits	6000 + número da tabela	Posição inicial da tabela	-	Leitura ou escrita
Bloco de Real ou Float	5000	Endereço inicial	-	Leitura ou escrita
Bloco de Inteiro de 32 Bits	5003	Endereço inicial	-	Leitura ou escrita
Bloco de Dados com Estampa de Tempo	5005	Posição inicial da tabela decimal	Endereço auxiliar	Leitura

Tabela de Memória

Leitura e Escrita

B1	Endereço de um CLP. Os valores possíveis são Número da rede × 100 + Número de um CLP ou 9999 para MNS
B2	Número da tabela
B3	Posição
B4	Não usado

Este Tag lê e escreve em tabelas de memória.

Tabela de Decimal

Leitura e Escrita

B1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP
B2	1000 + Número da tabela
B3	Posição
B4	Não usado

Este Tag lê e escreve em tabelas de decimal.

Status de Barramentos de Entrada e Saída

Somente Leitura

B1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP
B2	2000
B3	Número do barramento
B4	Não usado

Este Tag lê o status de um barramento de entrada e saída e dos módulos que o compõem. Os significados dos valores dos Elementos de Bloco retornados estão descritos na tabela a seguir.

Elementos de Bloco de um barramento de entrada e saída

ELEMENTO	DESCRIÇÃO	COMENTÁRIO
1	TOT_BAR	Total de barramentos de entrada e saída possíveis em um CLP
2	TIPO_BAR	Tipo de um barramento. Os valores possíveis são **0**: Barramento AL-1000, **1**: Barramento AL-3000 ou **2**: Barramento PONTO
3	ST_BAR	Estado de um barramento
4	NUM_MOD	Número de módulos de entrada e saída possíveis no barramento solicitado
5	TIPO_MOD_0	Tipo do módulo 0 (zero)
6	ST_MOD_0	Estado do módulo 0 (zero)
7	TIPO_MOD_1	Tipo do módulo 1 (um)

ELEMENTO	DESCRIÇÃO	COMENTÁRIO
8	ST_MOD_1	Estado do módulo 1 (um)
...	...	...
	TIPO_MOD_N	Tipo do módulo *n*
	ST_MOD_N	Estado do módulo *n*

Status de Comunicação

Somente Leitura

B1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP
B2	2001
B3	*Reset* de variáveis. Os valores possíveis são **0**: Os valores dos dados sobre recepção e transmissão não são alterados ou **1**: Os valores dos dados sobre recepção e transmissão são zerados
B4	Não usado

Este Tag lê informações sobre o estado da comunicação ALNET II. Os significados dos valores dos Elementos de Bloco estão descritos na tabela a seguir.

Elementos de Bloco do status da comunicação

ELEMENTO	DESCRIÇÃO	COMENTÁRIO
1	DADOS_TX	Número de transmissões sem erros
2		Número de transmissões com erro de colisão
3		Número de transmissões com erro de *underrun*
4		Número de transmissões sem recepção de ACK de hardware
5		Número de transmissões canceladas por esgotamento de retentativas
6		Número de *time-outs* de serviço
7		Número de faltas de *buffers* de transmissão
8		Reservado
9		Reservado
10	DADOS_RX	Número de recepções sem erros
11		Número de recepções com erro de colisão

ELEMENTO	DESCRIÇÃO	COMENTÁRIO
12		Número de recepções com erro de *overrun*
13		Número de recepções com erro de CRC
14		Número de recepções com erro de alinhamento
15		Número de recepções de pacotes com erro de tamanho
16		Número de *time-outs* de pacote
17		Número de faltas de *buffers* de recepção
18		Reservado
19	DADOS_COM	Velocidade de comunicação
20		Endereço de um nó
21		Endereço da sub-rede local
22		Grupos de *multicast* dos quais um nó é parte
23		Reservado
24		Reservado
25		*Time-out* intrabarramento
26		*Time-out* interbarramento
27		*Time-out* de pacote
28		Número máximo de retentativas de transmissão
29		Identificador de um nó na rede, com 20 bytes
30	DADOS_CNX	Tipo de conexão física. Os valores possíveis são **0**: Elétrica padrão ou **1**: Óptica
31		Redundância de conexões físicas. Os valores possíveis são **0**: Sem redundância ou **1**: Com redundância
32		Período para teste da redundância, em segundos
33		Tempo para a comutação da conexão física, em segundos
34		Reservado
35		Reservado
36		Conexão física selecionada. Os valores possíveis são 1 (um) ou 2 (dois)
37		Estado da conexão 1 (um). Os valores possíveis são **0**: OK ou **1**: Falha

ELEMENTO	DESCRIÇÃO	COMENTÁRIO
38		Estado da conexão 2 (dois). Os valores possíveis são **0**: OK ou **1**: Falha
39		Reservado
40		Conexão física forçada. Os valores possíveis são 0 (sem forçamento), 1 (um) ou 2 (dois)
41		Estado forçado da conexão 1 (um). Os valores possíveis são **0**: OK ou **1**: Falha
42		Estado forçado da conexão 2 (dois). Os valores possíveis são **0**: OK ou **1**: Falha

Status de um Equipamento

Somente Leitura

B1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP
B2	2002
B3	Não usado
B4	Não usado

Este Tag lê informações de estado de um equipamento. Os significados dos valores de cada Elemento de Bloco estão descritos na tabela a seguir.

Elementos de Bloco do status de um equipamento

ELEMENTO	MNEMÔNICO	DESCRIÇÃO
1	MODELO_UCP	Modelo da UCP
2	VERS_EXE_H	Versão do executável
3	VERS_EXE_L	
4	MODO_OP_1	Modo de operação 1 (um)
5	COD_MESG_1	Código de mensagem 1 (um)
6	RAM_LIV_2H	Espaço livre da RAM da aplicação banco 2 (dois)
7	RAM_LIV_2H	
8	RAM_LIV_1H	Banco 1 (um)
9	RAM_LIV_1L	
10	STAT_RAM_1	Estado da RAM da aplicação 1 (um)
11	TC_INST_H	Tempo de ciclo instantâneo
12	TC_INST_L	

ELEMENTO	MNEMÔNICO	DESCRIÇÃO
13	TC_MED_H	Tempo de ciclo médio
14	TC_MED_L	
15	TC_MAX_H	Tempo de ciclo máximo
16	TC_MAX_L	
17	TC_MIN_H	Tempo de ciclo mínimo
18	TC_MIN_L	
19	PER_E018	Período de chamada do módulo E018
20	PER_E019	Período de chamada do módulo E019
21	-	-
22	TEMPO_PROG	Tempo máximo de execução de uma aplicação
23	STAT_RAM_2	Estado da RAM da aplicação 2 (dois)
24	MODO_OP_2	Modo de operação 2 (dois)
25	RAM_LIV_8H	Espaço livre da RAM da aplicação banco 8 (oito)
26	RAM_LIV_8L	
27	RAM_LIV_7H	Banco 7 (sete)
28	RAM_LIV_7L	
29	RAM_LIV_6H	Banco 6 (seis)
30	DA RAM_LIV_6L	
31	RAM_LIV_5H	Banco 5 (cinco)
32	RAM_LIV_5L	
33	RAM_LIV_4H	Banco 4 (quatro)
34	RAM_LIV_4L	
35	RAM_LIV_3H	Banco 3 (três)
36	RAM_LIV_3L	
37	STATUS_EPR	Estado da EPROM de uma aplicação
38	EPR_LIV_8H	Espaço livre da EPROM da aplicação banco 8 (oito)
39	EPR_LIV_8L	
40	EPR_LIV_7H	Banco 7 (sete)
41	EPR_LIV_7L	
42	EPR_LIV_6H	Banco 6 (seis)
43	EPR_LIV_6L	
44	EPR_LIV_5H	Banco 5 (cinco)
45	EPR_LIV_5L	
46	EPR_LIV_4H	Banco 4 (quatro)
47	EPR_LIV_4L	

ELEMENTO	MNEMÔNICO	DESCRIÇÃO
48	EPR_LIV_3H	Banco 3 (três)
49	EPR_LIV_3L	
50	EPR_LIV_2H	Banco 2 (dois)
51	EPR_LIV_2L	
52	EPR_LIV_1H	Banco 1 (um)
53	EPR_LIV_1L	
54	COD_MESG_2	Código de mensagem 2 (dois)
55	COD_MESG_3	Código de mensagem 3 (três)
56	COD_MESG_4	Código de mensagem 4 (quatro)
57	IDENT_0	Identificação secundária de produto (ASCII)
58	IDENT_1	
59	IDENT_2	
60	IDENT_3	
61	IDENT_4	
62	IDENT_5	
63	IDENT_6	
64	IDENT_7	
65	EPR_LIV_16H	Banco 16
66	EPR_LIV_16L	
67	EPR_LIV_15H	Banco 15
68	EPR_LIV_15L	
69	EPR_LIV_14H	Banco 14
70	EPR_LIV_14L	
71	EPR_LIV_13H	Banco 13
72	EPR_LIV_13L	
73	EPR_LIV_12H	Banco 12
74	EPR_LIV_12L	
75	EPR_LIV_11H	Banco 11
76	EPR_LIV_11L	
77	EPR_LIV_10H	Banco 10
78	EPR_LIV_10L	
79	EPR_LIV_9H	Banco 9 (nove)
80	EPR_LIV_9L	

Bloco de Memória

Leitura e Escrita

B1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP
B2	3000
B3	Endereço inicial
B4	Não usado

Este Tag lê e escreve em blocos de operandos do tipo **Memória**.

Tabela de Real ou Float

Leitura e Escrita

B1	Endereço de um CLP
B2	4000 + Número da tabela
B3	Posição
B4	Não usado

Este Tag lê e escreve em tabelas de operandos de tipo **Real** ou **Float**.

NOTA

Para valores no formato **Float** com números não normalizados, assume-se o valor 0 (zero) para os casos de *underflow* e assume-se o valor **null** para os casos de *overflow*.

Tabela de Inteiro de 32 Bits

Leitura e Escrita

B1	Endereço de um CLP
B2	6000 + Número da tabela
B3	Posição
B4	Não usado

Este Tag lê e escreve em tabelas de operandos de tipo **Inteiro de 32 bits**.

Bloco de Real ou Float

Leitura e Escrita

B1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP
B2	5000
B3	Endereço inicial
B4	Não usado

Este Tag lê e escreve em blocos de operandos do tipo **Real** ou **Float**.

NOTA

Para valores no formato **Float** com números não normalizados, assume-se o valor 0 (zero) para os casos de *underflow* e assume-se o valor **null** para os casos de *overflow*.

Bloco de Inteiro de 32 Bits

Leitura e Escrita

B1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP
B2	5003
B3	Endereço inicial
B4	Não usado

Este Tag lê e escreve em blocos de operandos do tipo **Inteiro de 32 bits**.

Bloco de Dados com Estampa de Tempo

Leitura e Escrita

B1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP
B2	5005
B3	Posição inicial da tabela decimal
B4	Endereço auxiliar

Este Tag permite a realização da leitura de variáveis de tabelas decimais com estampa de tempo. Para mais informações sobre a utilização deste recurso, consulte o tópico **Leitura de Dados com Estampa de Tempo**.

Configuração por Strings

Este Driver permite a configuração de Tags através de **Strings**, utilizando o mesmo formato usado no Driver OPC. A configuração por **Strings** é realizada através das propriedades **Dispositivo** e **Item** dos Tags do **Elipse E3**, **Elipse Power** ou **Elipse Water**.

NOTA

A configuração por **Strings** ainda não estava disponível para o **Elipse SCADA** até a publicação desta versão.

A propriedade **Dispositivo** pode ser utilizada para definir o endereço de um CLP, usando o formato **<net>.plc:**, onde **net** é um parâmetro opcional identificando o número da rede, e **plc** é o endereço de um CLP. Se a propriedade **Dispositivo** é deixada em branco, o endereço deve ser informado na propriedade **Item** utilizando o mesmo formato, seguido por dois pontos. Se o endereço de rede não está presente, considera-se como 0 (zero). Se o endereço é definido na propriedade **Dispositivo**, os dois pontos finais podem ser dispensados. A propriedade **Item** define os dados do operando a ser acessado. A tabela a seguir contém os mnemônicos que podem ser utilizados com este Driver.

Mnemônicos suportados

MNEMÔNICO	DESCRIÇÃO	MODO
<<net>.plc:>Ax	Auxiliar	Leitura e escrita
<<net>.plc:>Ax.b	Bit de auxiliar	Leitura e escrita
<<net>.plc:>Dx	Decimal	Leitura e escrita
<<net>.plc:>Dx.b	Bit de decimal	Leitura e escrita
<<net>.plc:>Ex	Entrada	Leitura
<<net>.plc:>Ex.b	Bit de entrada	Leitura
<<net>.plc:>Fx	Real (*Float*)	Leitura e escrita
<<net>.plc:>Ix	Inteiro (*32-bits*)	Leitura e escrita
<<net>.plc:>Mx	Memória	Leitura e escrita
<<net>.plc:>Mx.b	Bit de memória	Leitura e escrita
<<net>.plc:>Sx	Saída	Escrita
<<net>.plc:>Sx.b	Bit de saída	Escrita
<<net>.plc:>TDx.y	Tabela de decimais	Leitura e escrita
<<net>.plc:>TFx.y	Tabela de Reais (*Float*)	Leitura e escrita
<<net>.plc:>TIx.y	Tabela de Inteiros (*32-bits*)	Leitura e escrita
<<net>.plc:>TMx.y	Tabela de memórias	Leitura e escrita
<<net>.plc:>BSx	Status do barramento de entrada e saída	Leitura
<<net>.plc:>CSx	Status da comunicação	Leitura
<<net>.plc:>ES	Status do equipamento	Leitura

Em que:

- Os caracteres em negrito representam mnemônicos fixos que identificam os comandos, enquanto os caracteres que não estão em negrito representam os valores que devem ser substituídos
- O caractere *y* representa a posição inicial na tabela
- O caractere *b* representa o número de um bit, variando de 0 (zero, bit menos significativo) até 31 (bit mais significativo), porém o número máximo depende do número de bits presentes no tipo de dados

A seguir alguns exemplos de uso dos mnemônicos e a respectiva conversão em parâmetros *N* ou *B*, conforme realizada internamente por este Driver.

Exemplos de uso de mnemônicos com Tags do tipo PLC

DISPOSITIVO	ITEM	N1 OU B1	N2 OU B2	N3 OU B3	N4 OU B4
	1:D8	1 (um)	3 (três)	8 (oito)	0 (zero)
	2.5:F50	205	10	50	0 (zero)
8.2:	E7	802	6 (seis)	7 (sete)	0 (zero)
	120:S5	120	6 (seis)	5 (cinco)	0 (zero)
5.4:	A120	504	5 (cinco)	120	0 (zero)
	2:M8	2 (dois)	0 (zero)	8 (oito)	0 (zero)
	6.5:TD6.123	605	4 (quatro)	6 (seis)	123
2.3	TF84.5	203	11	84	5 (cinco)
6	TM5.64	6 (seis)	1 (um)	5 (cinco)	64
7.1:	M125.1	701	0 (zero)	125	2 (dois)

NOTA

Os mnemônicos **E** e **S** são ambos mapeados para Tags PLC com o parâmetro *N2* igual a 6 (seis). A diferença está apenas na operação em si, pois somente são permitidas leituras em operandos **E** e escritas em operandos **S**.

Leitura de Dados com Estampa de Tempo

Este Driver permite a leitura de dados com estampa de tempo vinda de um CLP indicando o momento da aquisição.

Os modos de aquisição são a **leitura de blocos de dados adquiridos simultaneamente**, ou seja, com a mesma estampa de tempo, e a **leitura de blocos de dados adquiridos em intervalos constantes**. Em ambos os casos, o programa dos controladores precisa ser capaz de atender a determinados requisitos. A seguir são detalhados estes modos de trabalho.

Modo de Blocos de Dados com Aquisição Simultânea

Neste modo, pode-se realizar a leitura de variáveis de tabelas decimais com estampa de tempo quando se tem um conjunto de variáveis armazenadas simultaneamente. Logo, cada conjunto de valores com a informação de tempo associada formam uma estrutura, que é repetida diversas vezes, formando assim uma tabela. Cada estrutura deve estar no formato descrito a seguir.

```
Dado 1 - HHMMSS  (Exemplo: 19:02 e 34 segundos são colocados como 190234)
Dado 2 - DDMMAA  (Exemplo: 18 de fevereiro de 1998 é colocado como 180298)
Dado 3 - valor 1
Dado 4 - valor 2
(...)
Dado N - valor N - 2
```

O tamanho máximo permitido para uma tabela é de 255 operandos inteiros, ou 1020 bytes ao todo.

Por exemplo, em uma tabela de 255 posições, deseja-se armazenar várias leituras de 5 (cinco) variáveis analógicas com a estampa de tempo. Logo, a tabela após preenchida tem a configuração da tabela a seguir, supondo que a data atual é 18/02/1998, 9:00:00, com aquisições de minuto em minuto.

Exemplo de estrutura de dados com estampa de tempo

POSIÇÃO NA TABELA	VALOR	POSIÇÃO NA TABELA	VALOR	POSIÇÃO NA TABELA	VALOR
0	090000	**7**	090100	**14**	090200
1	180298	**8**	180298	**15**	180298
2	Dado	**9**	Dado	**16**	Dado
3	Dado	**10**	Dado	**17**	Dado
4	Dado	**11**	Dado	**18**	Dado
5	Dado	**12**	Dado	**19**	Dado
6	Dado	**13**	Dado	**20**	Dado

E assim por diante, até preencher a tabela.

NOTA

Como tem-se apenas 2 (dois) dígitos para o ano, este Driver está preparado para funcionamento até o ano de 2098.

Para o controle de acesso à tabela, existem 4 (quatro) bits de uma memória auxiliar em um CLP:

- **BIT 0 (pede_permissao_de_leitura)**: Executa uma requisição de acesso à tabela para um CLP
- **BIT 1 (leitura_permitida)**: Verifica se a requisição foi aceita
- **BIT 2 (limpa)**: Este Driver escreve o valor 1 (um) neste bit e o CLP se encarrega de zerar a tabela e voltar este bit para 0 (zero)
- **BIT 3 (tabela_zerada)**: Este bit informa se a tabela está zerada

Para a leitura dos dados armazenados, deve-se utilizar um Tag Bloco com o parâmetro *B2* igual a 6000, em que o parâmetro *B3* indica a posição inicial da tabela e o parâmetro *B4* indica o endereço do byte contendo os bits auxiliares descritos anteriormente.

Na leitura, caso exista mais de um conjunto de valores ou estruturas na tabela, este Driver devolve uma lista de valores para uma aplicação da **Elipse Software**, como é o caso do exemplo anterior. A leitura de estruturas prossegue até que este Driver encontre valores nulos nos primeiros 2 (dois) campos de data e hora, indicando o fim da tabela, ou até que ultrapasse o limite de 255 operandos.

A utilização dos recursos do exemplo anterior está prevista somente para a utilização com blocos. Portanto, no caso de apenas uma variável, crie um bloco de apenas uma variável.

Sequência de Passos Executados por este Driver

1. Verifica o bit 3 (três, tabela zerada). Se está configurado em 1 (um), sai ou prossegue em caso contrário.
2. Escreve 1 (um) no bit 0 (zero, pede permissão).
3. Verifica o bit 1 (um). Se está configurado em 1 (um), prossegue ou verifica mais uma vez em caso contrário. Persistindo a não autorização, sai. Neste caso, o CLP deve tratar a requisição pendente, voltando o bit 0 (zero, pede permissão) para 0 (zero).
4. Lê as primeiras *n* + 2 (duas) variáveis, em que *n* é o tamanho do bloco.
5. Caso os 2 (dois) primeiros valores contenham uma data válida, estes valores são copiados.
6. Repete o passo anterior até chegar ao fim da tabela ou quando existirem valores nulos nos 2 (dois) primeiros campos.
7. Zera a tabela, caso tenha sido lida pelo menos 1 (uma) estrutura, escrevendo 1 (um) no bit 2 (dois, limpa).
8. Este Driver devolve os valores para uma aplicação da **Elipse Software**.

Modo de Blocos de Dados com Aquisição a Intervalos Constantes

Deve-se utilizar este modo quando o intervalo de tempo entre os dados em uma tabela é fixo. Assim pode-se economizar espaço em memória dentro de um CLP, armazenando-se somente o instante inicial da coleta. Este modo só pode ser usado quando há vários dados para 1 (uma) só variável. Para isto, deve-se realizar o procedimento a seguir.

Em um CLP, deve existir um programa que armazene os dados em uma tabela de decimais. O formato da tabela tem a estrutura a seguir.

```
Dado 1 - HHMMSS
Dado 2 - DDMMAA
Dado 3 - intervalo de tempo entre cada coleta, em milissegundos
Dado 4 - valor 1
(...)
Dado N - valor N - 3
```

Por exemplo, em uma tabela de 255 posições, deseja-se armazenar várias leituras de 1 (uma) variável analógica com intervalos de tempo constantes. Logo, a tabela após preenchida tem com a seguinte configuração, supondo que a data atual é 18/02/1998, 9:00:00, com aquisições de segundo em segundo.

Exemplo de dados com intervalo de estampa de tempo constante

POSIÇÃO NA TABELA	VALOR	POSIÇÃO NA TABELA	VALOR	POSIÇÃO NA TABELA	VALOR
0	090000	**7**	Dado	**14**	Dado
1	180298	**8**	Dado	**15**	Dado
2	1000	**9**	Dado	**16**	Dado
3	dado	**10**	Dado	**17**	Dado
4	dado	**11**	Dado	**18**	Dado
5	dado	**12**	Dado	**19**	Dado
6	dado	**13**	Dado	**20**	Dado

E assim por diante, até preencher a tabela. Para o controle de acesso à tabela, existem 4 (quatro) bits de uma memória auxiliar em um CLP. Este auxiliar é informado na leitura de um Tag.

- **BIT 0 (pede_permissao_de_leitura)**: Executa uma requisição de acesso à tabela para um CLP
- **BIT 1 (leitura_permitida)**: Verifica se a requisição foi aceita
- **BIT 2 (limpa)**: Este Driver escreve o valor 1 (um) neste bit e o CLP se encarrega de zerar a tabela e voltar este bit para 0 (zero)
- **BIT 3 (tabela_zerada)**: Este bit informa se a tabela está zerada

A leitura dos dados só pode ser realizada através de um Tag PLC com o parâmetro *N2* igual a 13, em que o parâmetro *N3* indica o endereço da tabela decimal e o parâmetro *N4* indica o endereço do auxiliar que contém os bits de controle. Consulte as tabelas **Tipos de dados suportados para Tags PLC** e **Tipos de dados suportados para Tags Bloco** para mais informações. Caso o valor de um dado lido seja diferente de 0 (zero), este Driver devolve uma lista de valores de eventos para uma aplicação da **Elipse Software**, cada evento com uma estampa de tempo igual à estampa de tempo da tabela + (*N* - 3) × intervalo de tempo, em que *N* é a posição da variável dentro da tabela.

Sequência de Passos Executados por este Driver

Ao executar a leitura da tabela com a configuração anterior, este Driver executa os passos a seguir.

1. Verifica o bit 3 (três, tabela zerada). Se está configurado em 1 (um) sai ou prossegue em caso contrário.
2. Escreve 1 (um) no bit 0 (zero, pede permissão).
3. Verifica o bit 1 (um) e, se está configurado em 1 (um), prossegue. Caso contrário verifica mais uma vez. Persistindo a não autorização, sai. Neste caso o CLP deve tratar a requisição pendente, voltando o bit 0 (zero, pede permissão) para 0 (zero).
4. Lê as primeiras 3 (três) variáveis com a estampa de tempo e o intervalo.
5. Caso os 3 (três) primeiros valores contenham uma data válida, esta é armazenada.
6. Lê as próximas posições da tabela, de 12 em 12 unidades, armazenando o valor juntamente com a estampa de tempo adicionada de cada intervalo sucessivamente. Caso o primeiro valor lido de cada leitura seja igual a 0 (zero), devolve os últimos valores lidos e zera a tabela, ou seja, só transfere valores de um bloco de 12 variáveis se o primeiro valor é diferente de 0 (zero).
7. Zera a tabela, caso tenha sido lida pelo menos uma estrutura, e escreve 1 (um) no bit 2 (dois, limpa).
8. Este Driver devolve os valores para uma aplicação da **Elipse Software**.

Mensagens Não Solicitadas (MNS)

O uso das mensagens não solicitadas permite melhorar significativamente a performance de uma aplicação, pois não há necessidade de executar *polling* nas variáveis de um PLC o tempo todo, mesmo quando estas variáveis não sofrem alteração. Ao contrário, usando-se mensagens não solicitadas, toda vez que uma variável sofre alteração, o próprio PLC envia uma mensagem avisando este Driver sobre o novo valor que, por sua vez, altera a base de dados.

Uma aplicação, por sua vez, deve configurar os Tags PLC e Bloco de maneira muito semelhante àquela usada para Tags normais, por *polling*, mudando-se apenas o parâmetro *N1* ou *B1* para 9999. Este número indica a este Driver que os valores destes Tags devem ser lidos não de um PLC, mas diretamente da base de dados deste Driver, atualizada pelas MNS. Os Tags de uma aplicação, portanto, continuam funcionando por *polling*, exceto que a leitura é direta de uma memória imagem contida neste Driver, que não gera comunicação com um PLC. Como esta operação é extremamente

rápida, sugere-se usar um período de *scan* muito curto, da ordem de milissegundos, o que permite uma atualização imediata dos Tags em caso de alteração do valor por mensagens não solicitadas. A leitura de mensagens não solicitadas só está habilitada neste Driver para os operandos do tipo **Memória**, **Tabela de Memória**, **Auxiliar** e **Octeto**.

Exemplo de Tag PLC para Mensagens Não Solicitadas

- **N1**: 101 (Rede igual a um e PLC igual a um)
- **N2**: 0 (zero, Memória)
- **N3**: 1 (um, Memória M1)
- **N4**: 0 (zero, não usado neste caso)
- **Scan**: 500 (500 milissegundos)

Para passar este Tag PLC para leitura diretamente da memória imagem deste Driver, altere o parâmetro *N1* para 9999. Uma vez que a operação neste caso é bem mais rápida do que por *polling*, pode-se alterar também o período de *scan* para apenas alguns milissegundos. A nova configuração é a seguinte:

- **N1**: 9999 (Leitura da Memória Imagem)
- **N2**: 0 (zero, Memória)
- **N3**: 1 (um, Memória M1)
- **N4**: 0 (zero, não usado neste caso)
- **Scan**: 10 (10 milissegundos)

Documentação das Interfaces de Comunicação

Esta seção contém a documentação das Interfaces de Comunicação referentes ao Driver **AlnetIv2**.

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.

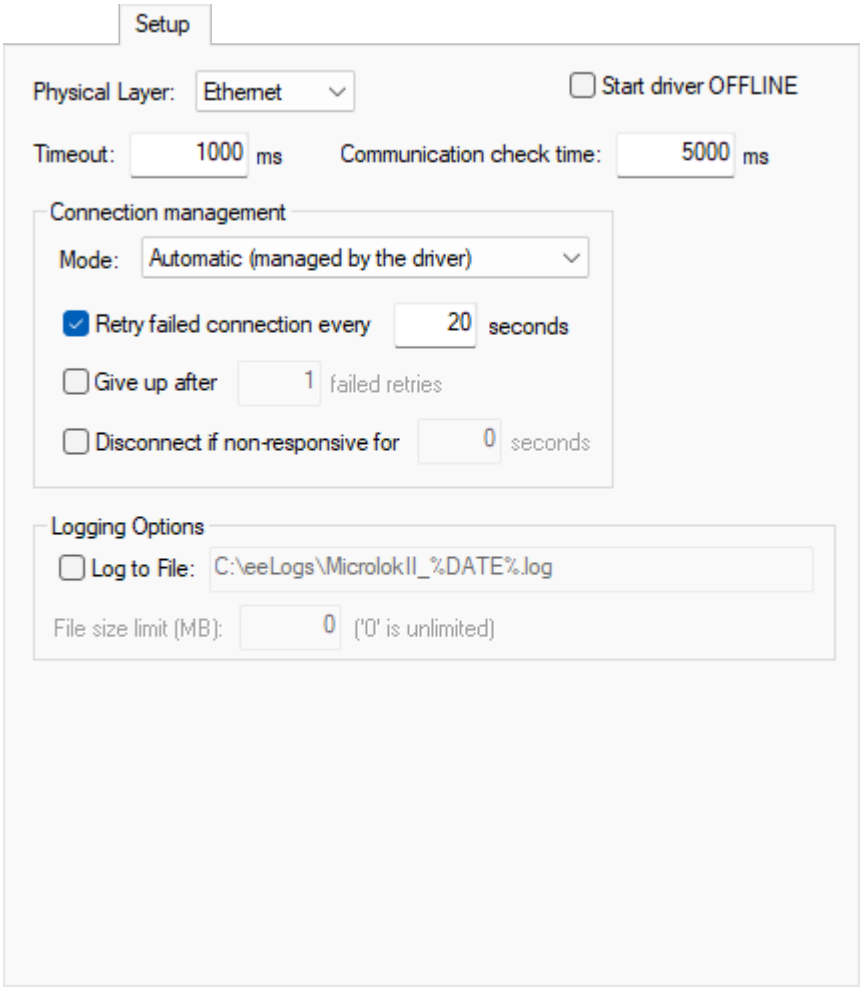
Caixa de Diálogo de Configuração

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



Aba Setup

Opções gerais da aba Setup

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

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

Opções para o grupo Connection management

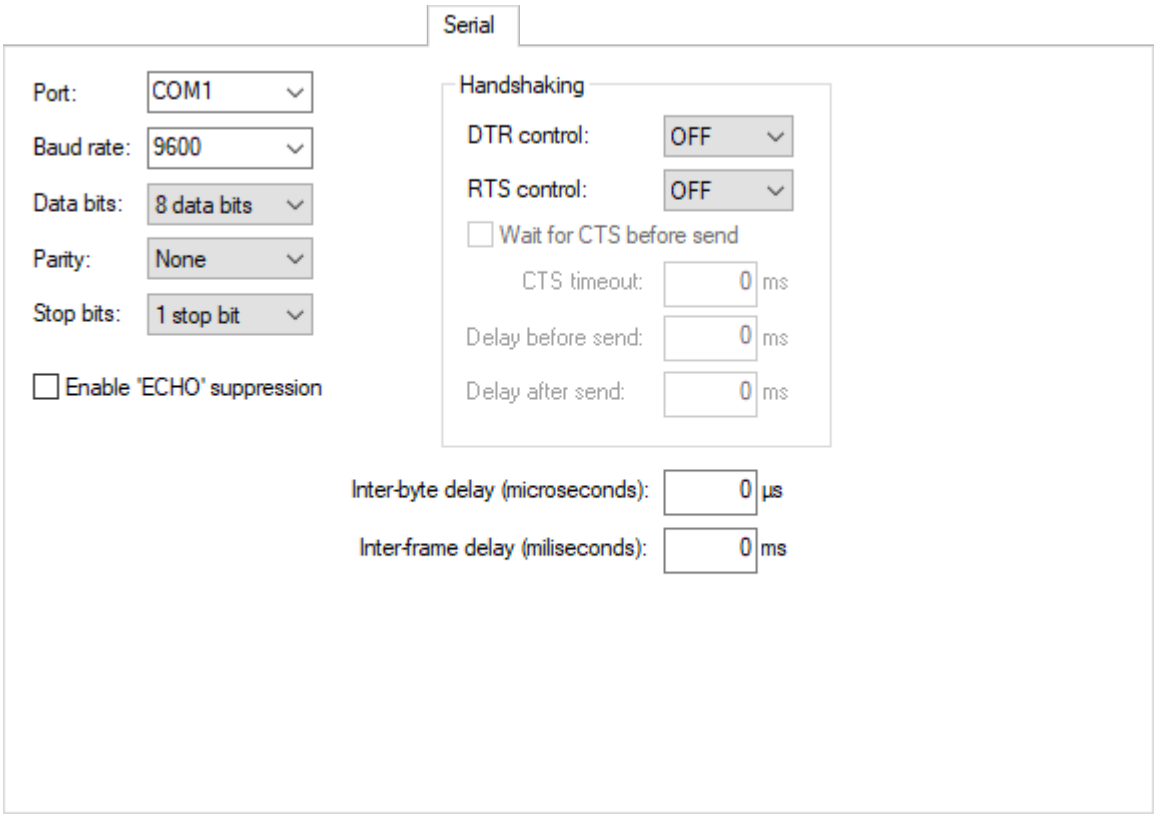
OPÇÃO	DESCRIÇÃO
Mode	Seleciona o modo de gerenciamento de conexão. Selecionar a opção **Automatic** permite que um Driver gerencie a conexão automaticamente, como especificado nas opções seguintes. Selecionar a opção **Manual** permite que uma aplicação gerencie a conexão completamente
Retry failed connection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

Opções para o grupo Logging Options

OPÇÃO	DESCRIÇÃO
Log to File	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**
File size limit (MB)	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

Aba Serial

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



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 **COM*n***, 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 *baud rate* a partir da lista (**1200**, **2400**, **4800**, **9600**, **19200**, **38400**, **57600** ou **115200**) ou digite um *baud rate*, 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' supression	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

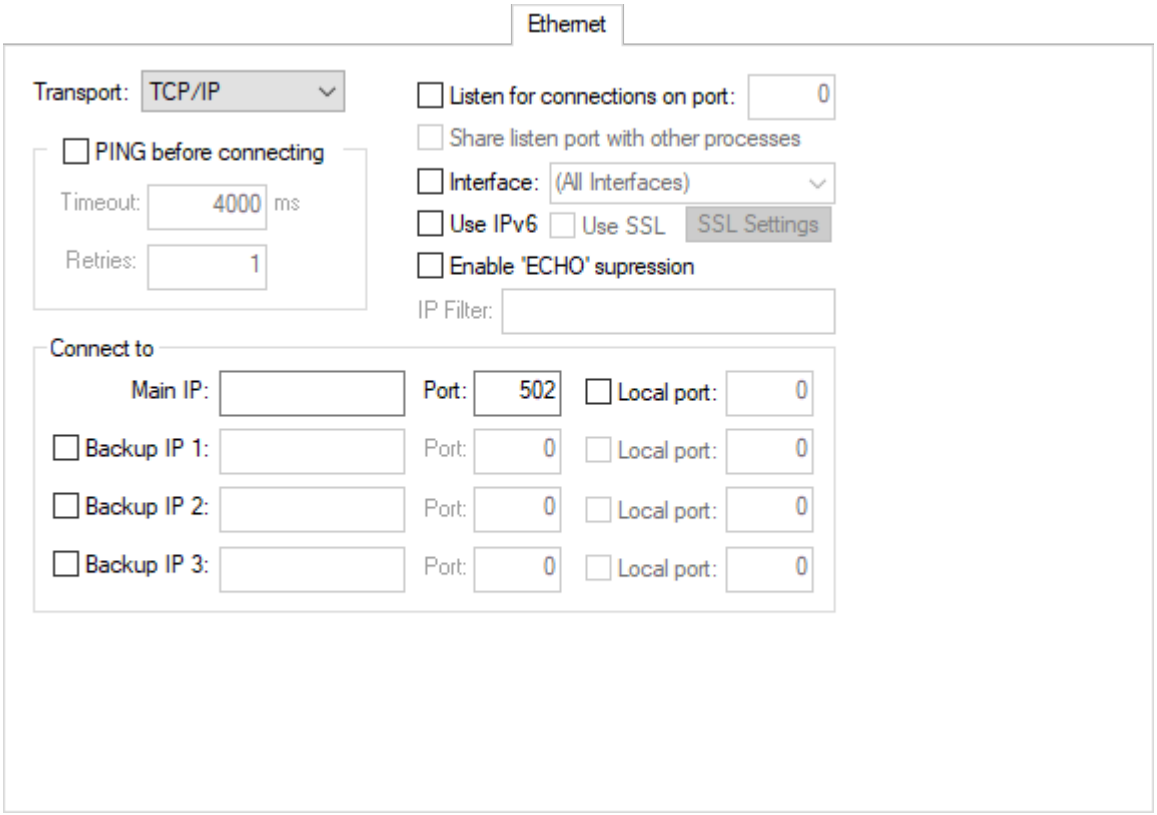
O grupo **Handshaking** configura o uso dos sinais **RTS**, **CTS** e **DTR** no processo de *handshaking* ou seja, controla quando um dado pode ser enviado ou recebido através de uma linha serial. Na maioria das vezes, configurar a opção **DTR control** para **ON** e a opção **RTS control** para **Toggle** 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 *flag* 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 *time-out*, 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 *socket* TCP (*stream*) ou selecione o valor **UDP/IP** para utilizar um *socket* UDP (*connectionless datagram*)
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 *eco* dos dados recebidos. O *eco* é uma cópia dos dados enviados, que pode ser retornada antes da mensagem de resposta

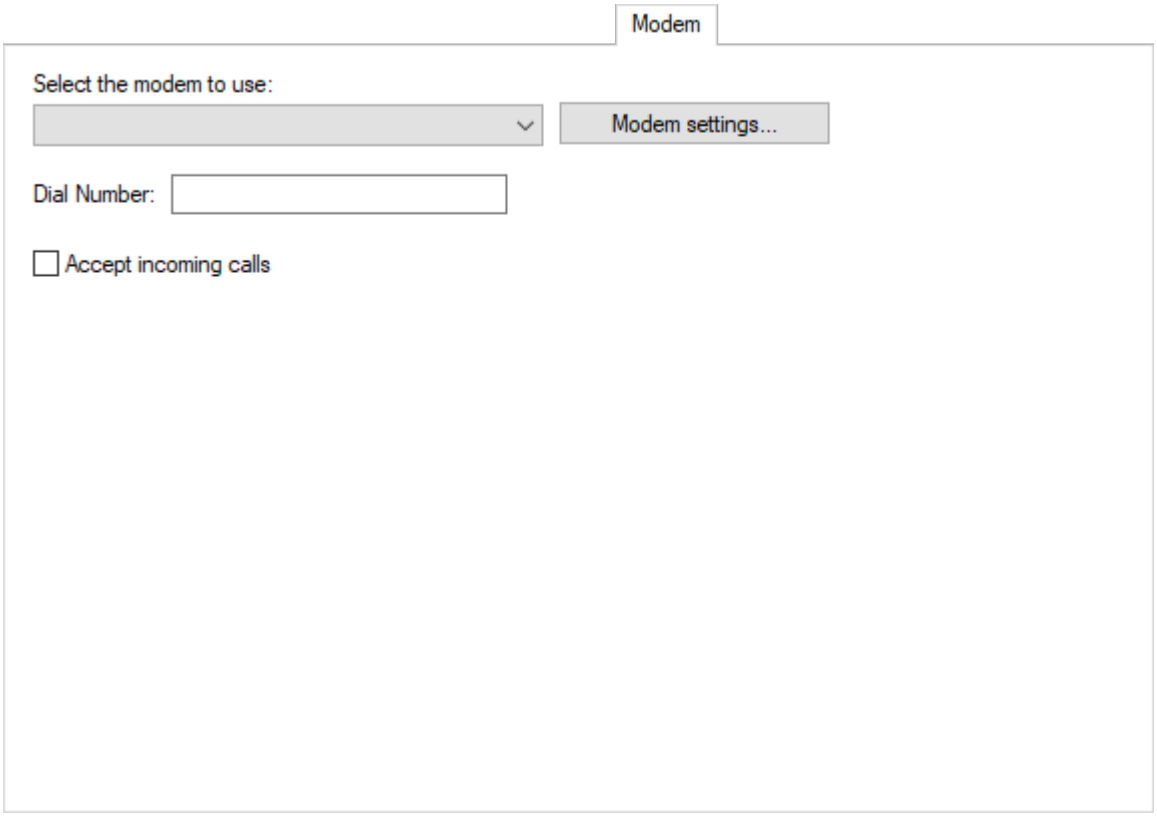
OPÇÃO	DESCRIÇÃO
IP Filter	Lista de endereços IP restringidos ou permitidos de onde um Driver aceita conexões (*Firewall*). Consulte a propriedade **IO.Ethernet.IPFilter** para mais informações
PING before connecting	Habilite esta opção para executar um comando **ping**, ou seja, para verificar se um dispositivo pode ser encontrado na rede, em um dispositivo antes de tentar uma conexão com o *socket*. Esta é uma maneira rápida de determinar uma conexão bem sucedida antes de tentar abrir um *socket* com um dispositivo. O *time-out* de uma conexão com um *socket* pode ser bem alto. As opções disponíveis são: • **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 *socket* 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 *backup* 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**.

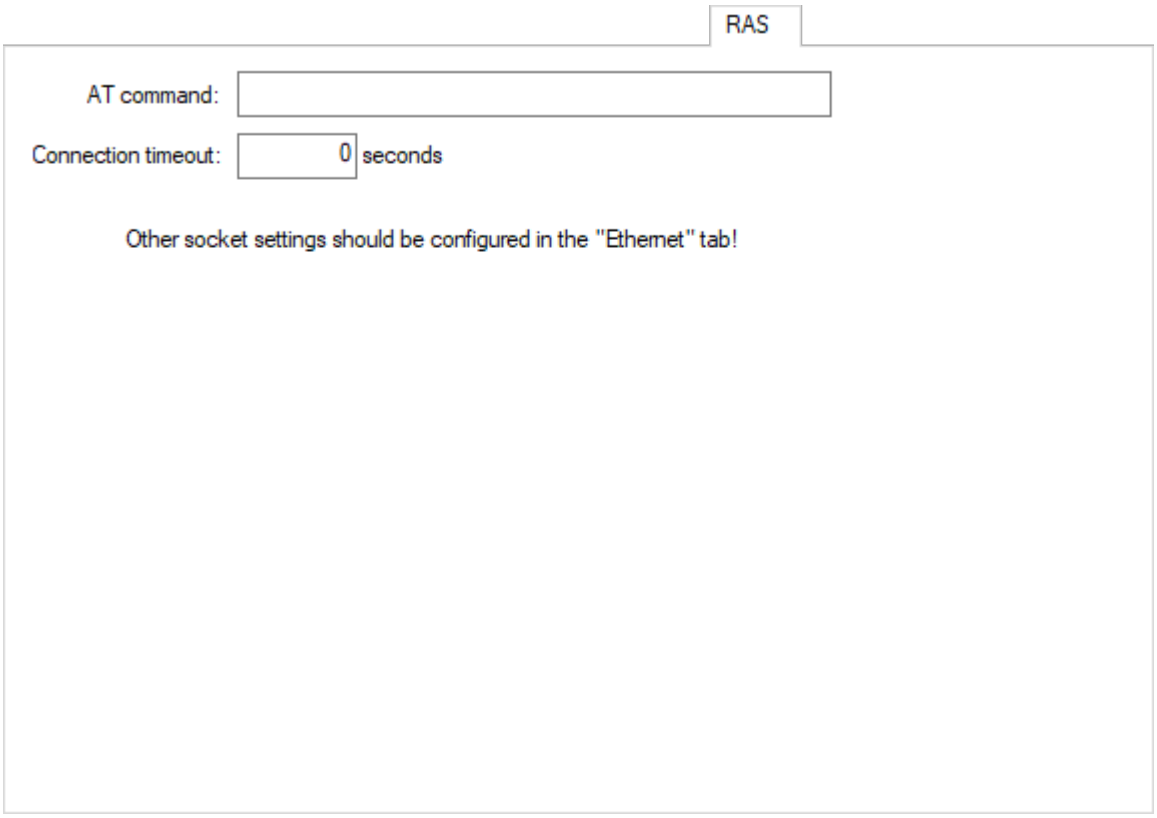
A Interface **RAS** abre uma conexão *socket* com um dispositivo RAS. Um dispositivo RAS é um servidor de modems acessível através de TCP/IP, aguardando por conexões *socket* em uma porta IP. Para cada conexão aceita nesta porta tem-se acesso a um modem.

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**.



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 **Elipse 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 **Elipse 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

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

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

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

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

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

IO.Type

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]

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]

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]

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

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:*:*)**: 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

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

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

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

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

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 COM*n*.

IO.Serial.RTS

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

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.SupressEcho

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 das Revisões do Driver

VERSÃO	DATA	AUTOR	COMENTÁRIOS
2.4.10	01/09/2025	M. Ludwig	• Driver atualizado para a biblioteca **IOKit** versão **3.0** e Visual Studio 2022 (*Case 37935*).
2.4.9	02/10/2020	C. Mello	• Adicionado suporte para operandos do tipo **Inteiro de 32 Bits** (*Case 29471*). • Alterado o parâmetro *B2* do Tag Bloco de Dados com Estampa de Tempo de 6000 para 5005.
2.4.8	27/09/2019	C. Mello	• Driver portado para o Visual Studio 2017 (*Case 27437*).
2.4.7	27/09/2013	A. Quites	• Ajustes na validação de parâmetros dos Tags (*Case 14706*). • Corrigido um erro na leitura de Tags internos da biblioteca **IOKit** (*Case 13585*). • Driver portado para a biblioteca **IOKit** versão **2.0** (*Case 13654*).
2.3.1	07/10/2008	A. Quites	• Adicionado o tipo de dados inteiro sem sinal (*Case 9555*).
2.2.1	31/10/2007	A. Quites	• Corrigido um erro na escrita de bits de

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			operandos **Auxiliar** e **Octeto** com o parâmetro *N4* maior que 0 (zero) (*Case 8887*).
2.1.1	21/03/2007	A. Quites	• Corrigido um erro na leitura de blocos de dados com estampa de tempo, adquiridos no modo de aquisição simultânea, em que este Driver não enviava um comando para limpar a tabela após a leitura, de forma que os dados eram lidos mais de uma vez (*Case 7374*).
2.0.1	28/07/2006	A. Quites	• Versão original para a biblioteca **IOKit**, unificando os Drivers **AL2000**, **AL2000 Timestamp**, **AL2000MNS** e **AL2000-Modem**.
1.0.1		R. Haetinger	• Todas as publicações de Drivers para Altus AL2000 anteriores ao controle de revisões.

Matriz
Rua Mostardeiro, 322/Cj. 902, 1001 e 1002
90430-000 — Porto Alegre — RS
Fone: (+55 51) 3346-4699
Fax: (+55 51) 3222-6226
E-mail: elipse-rs@elipse.com.br

Filial no Paraná
Av. Sete de Setembro, 4698/1708
80240-000 — Curitiba — PR
Fone: (+55 41) 4062-5824
E-mail: elipse-pr@elipse.com.br

Filial no Rio de Janeiro
Av. José Silva de A. Neto, 200/Bl. 4/Sl. 109B
22250-044 — Rio de Janeiro — RJ
Fone: (+55 21) 2430-5912
Suporte Técnico: (+55 21) 2430-5963
E-mail: elipse-rj@elipse.com.br

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

Filial em Minas Gerais
Rua Antônio de Albuquerque, 156/705
30112-010 — Belo Horizonte — MG
Fone: (+55 31) 4062-5824
E-mail: elipse-mg@elipse.com.br

Filial em Taiwan
9F., No.12, Beiping 2nd St., Sanmin Dist. 807 — Kaohsiung City — Taiwan
Fone: (+886 7) 323-8468
Fax: (+886 7) 323-9656
E-mail: evan@elipse.com.br

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

www.elipse.com.br
kb.elipse.com.br
forum.elipse.com.br
www.youtube.com/elipsesoftware
elipse@elipse.com.br

Gartner 2014 Cool Vendor

OPC FOUNDATION MEMBER

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)

Copyright © 2013-2025 Elipse Software Ltda. Todos os direitos reservados.