

Driver Moeller Easy

Nome do Arquivo	Easy.dll
Fabricante	Moeller Eaton Industries
Equipamentos	Equipamentos Moeller Easy
Protocolo	Proprietário
Versão	3.0.1
Última Atualização	17/03/2026
Plataforma	Win32
Dependências	IOKit versão 1.15 ou superior
Superblocos	Não
Nível	0

Introdução

Este Driver permite a comunicação entre sistemas da **Eclipse Software** e equipamentos Easy da Moeller Eaton Industries, utilizando um protocolo de comunicação proprietário. A parte da comunicação utiliza um cabo RS-232 normal, fornecido pela Moeller Eaton Industries.

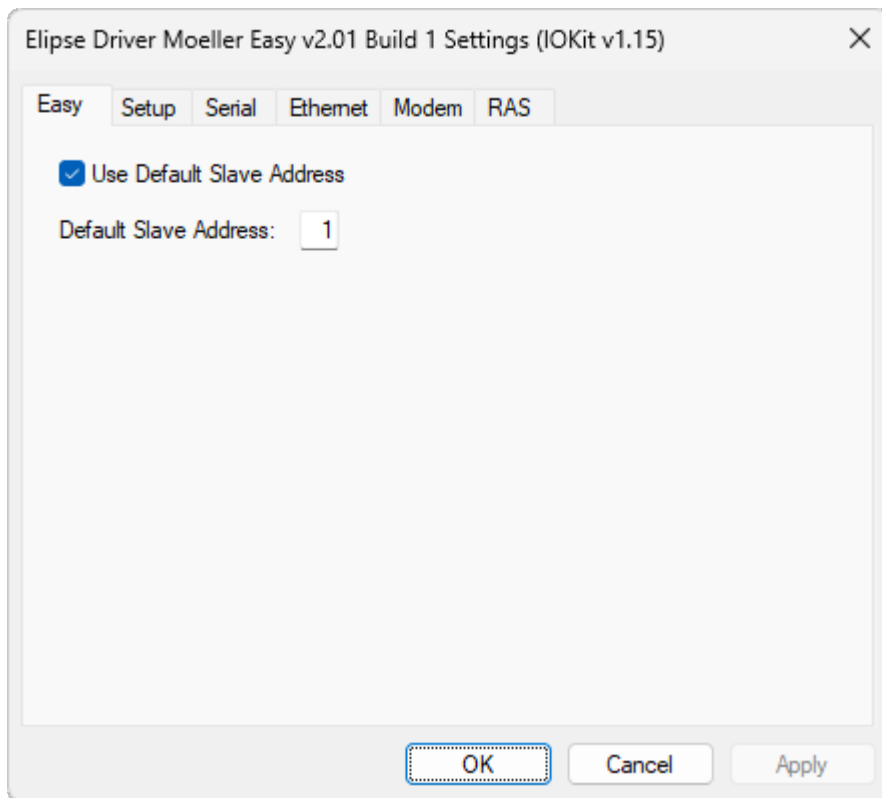
Configuração do Driver

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

Parâmetros [P]

P1	Não utilizado, deixe em 0 (zero)
P2	Não utilizado, deixe em 0 (zero)
P3	Não utilizado, deixe em 0 (zero)
P4	Não utilizado, deixe em 0 (zero)

Todas as configurações são realizadas na aba **Easy** da janela de propriedades deste Driver, mostrada na figura a seguir.



Aba Easy

A opção **Use Default Slave Address** permite configurar um valor padrão a ser usado como endereço de um equipamento Easy **Escravo**. Esta opção vem selecionada por padrão e o valor padrão da opção **Default Slave Address** é 1 (um). Se a opção **Use Default Address** está selecionada, qualquer valor usado nos parâmetros *N1* ou *B1* dos Tags deste Driver são desconsiderados.

NOTA

O número máximo de Elementos de um Tag Bloco é 8 (oito, o máximo suportado pelo protocolo). Caso seja criado um Tag Bloco que exceda este limite, os Elementos após este limite são ignorados por este Driver.

Referência de Tags

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

Parâmetros N e B de Endereçamento de Tags

N1 ou B1	Endereço de um equipamento Easy Escravo , entre 1 (um) e 15
N2 ou B2	Tipo de objeto para leitura ou escrita ou o número de um comando. Para mais informações, consulte as tabelas Comandos disponíveis e Leitura ou escrita
N3 ou B3	Endereço, usado somente para leitura ou escrita e não para comandos
N4 ou B4	Não usado, deixe em 0 (zero)

Comandos

Comandos disponíveis

N2 OU B2	N3 OU B3	COMANDO
C0h	-	Parar
C1h	-	Executar
C2h	-	Habilitar <i>Write Program</i> (Desabilitar Editor)
C3h	-	Desabilitar <i>Write Program</i> (Verificar EEPROM)
C4h	-	Reiniciar <i>System Test (Watchdog)</i>
C5h	-	Inverter <i>Remanenz On or Off</i>
C6h	-	Inverter <i>Anlauf Stop or Run</i>
C7h	-	Inverter LED
C8h	-	Ligar <i>P-Keys</i>
C9h	-	Desligar <i>P-Keys</i>
CAh	-	Ligar <i>I-Entprellung</i>
CBh	-	Desligar <i>I-Entprellung</i>
CCh	-	Ligar <i>Daylight Savings</i>
CDh	-	Desligar <i>Daylight Savings</i>
CEh	-	Ligar Modo Remoto
CFh	-	<i>Clear Display in Remote Mode</i>
D0h	-	Desligar Modo Remoto
D1h	-	<i>Hole vierstelligen Wert im Remotemodus</i>
D2h	-	Hablilitar <i>Auftragsbuch Sonder-Objekt</i>
D3h	-	Desabilitar <i>Auftragsbuch Sonder-Objekt</i>

Leitura ou Escrita

Leitura ou escrita

N2 OU B2	N3 OU B3	TIPO DE OBJETO
00h	Endereço	Programa
01h	Endereço	Dados Dinâmicos, Identificação e Status
02h	Endereço	Relógio interno
03h	Endereço	Display, apenas em Modo Remoto
04h	Endereço	Memória especial para Depuração e Monitoramento
05h	Endereço	<i>Auftragsbuch für Sonderspeicher</i>

NOTA

Para uma lista de objetos e respectivos endereços, consulte o tópico **Dicas Sucom Easy**.

Exemplos

Para acessar as saídas digitais de um equipamento Easy, os parâmetros de um Tag PLC devem ser os seguintes:

- **N1:** 1 (um, **Escravo** com identificador um)
- **N2:** 01h (dados dinâmicos, identificação e status)
- **N3:** 16h (para mais informações, consulte o tópico **Dicas Sucom Easy**)
- **N4:** 0 (zero, não utilizado)

Para acessar os *Merkers* de 0 (zero) a 7 (sete), os parâmetros de um Tag PLC devem ser os seguintes:

- **N1:** 1 (um, **Escravo** com identificador um)
- **N2:** 01h (dados dinâmicos, identificação e status)
- **N3:** 14h (para mais informações, consulte o tópico **Dicas Sucom Easy**)
- **N4:** 0 (zero, não utilizado)

Para ler e ajustar o relógio interno de um equipamento Easy, os parâmetros de um Tag Bloco devem ser os seguintes:

- **B1:** 1 (um, **Escravo** com identificador um)
- **B2:** 02h (relógio interno)
- **B3:** 00h (para mais informações, consulte o tópico **Dicas Sucom Easy**)
- **B4:** 0 (zero, não utilizado)
- **Size:** 3 (três, para mais informações, consulte o tópico **Dicas Sucom Easy**)

De acordo com o tópico **Dicas Sucom Easy**, os Elementos de um Tag Bloco para o relógio interno são os seguintes:

- Dia da semana, entre 0 (zero) e 6 (seis)
- Hora, entre 0 (zero) e 23
- Minuto, entre 0 (zero) e 59

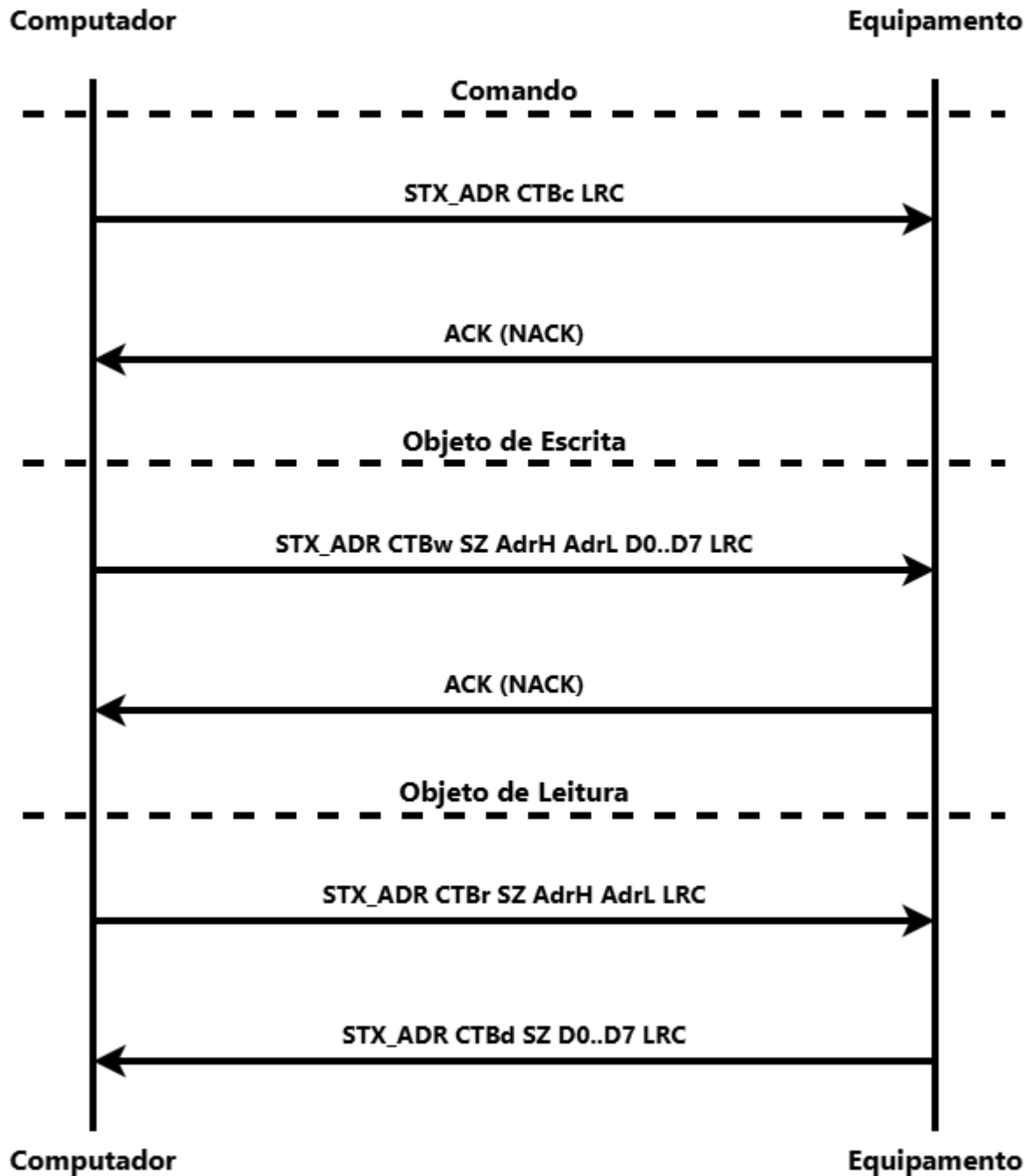
Pode-se ler estes Elementos separadamente mas, para configurar o relógio interno, um Tag Bloco deve ser escrito de uma vez só.

Dicas Sucom Easy

SUCOM-E

Transferência de dados para um computador

O protocolo aqui descrito permite a comunicação entre a porta serial RS-232 de um computador e a interface de um equipamento Easy com uma velocidade de 4800 bauds, 8 (oito) bits de dados e 1 (um) bit de parada, sem checagem de paridade.



Protocolo

```
STX_ADR = 1100aaaaB Byte de Controle Comando PC -> Easy
++++----- Endereçamento (16)
           00000 : Computador
           00001 : Easy
```

O byte **STX** sempre inicia a comunicação e contém o endereço de um equipamento Easy. Portanto, este protocolo suporta operações *multi-drop*, embora este recurso ainda não seja suportado pela porta serial RS-232.

A interface de programação Easy provê os serviços básicos descritos a seguir, codificados em um byte de controle, que é o primeiro byte a ser transferido.

Comandos em que o código de controle começa com o valor C0h (são permitidos até 32 comandos)

```

CTBc = 11000000B Byte de Controle Comando PC -> Easy
|||+++++--Código do Comando (32)
|||      00000: Parar
|||      00001: Iniciar
|||      00010: Habilitar programa de ESCRITA (-> Desabilita Editor)
|||      00011: Desabilitar programa de ESCRITA (Verifica->EEPROM)
|||      00100: Reiniciar teste do sistema (Watchdog)
|||      00101: Retenção ON/OFF
|||      00110: Execução STOP/RUN
|||      00111: Alternar estado do LED em 4xx (não implementado em 6xx)
|||      01000: Pushbuttons ON
|||      01001: Pushbuttons OFF
|||      01010: Debouncing ON
|||      01011: Debouncing OFF
|||      01100: Horário de Verão ON
|||      01101: Horário de Verão OFF
|||      01110: Ativar Modo Remoto
|||      01111: Limpar display no Modo Remoto
|||      10000: Desativar Modo Remoto
|||      10001: Buscar valor de 4 dígitos no Modo Remoto
|||      10010: Habilitar objeto especial de Order Book
|||      10011: Desabilitar objeto especial de Order Book
+++----- Comando
    
```

Um computador deve enviar um comando **EASY HALT** através da sequência a seguir.

```
STX_ADR (C1h), CTBc (C0h), LRC (C0h)
```

O byte **LRC** é igual a soma dos dados entre os bytes **STX_ADR** e **LRC**.

Objeto de memória contendo a leitura, um código de controle endereçado a partir de 80h, ou a escrita, um código de controle endereçado a partir de A0h (podem ser endereçados até 32 objetos)

```

CTBr = 10000000B Byte de Controle PC -> Easy "Leitura de objeto a partir de ENDEREÇO"
CTBw = 10100000B Byte de Controle PC -> Easy "Escrita de objeto a partir de ENDEREÇO"
|||+++++-- Código de Objetos (32 objetos)
|||      00000: Programa
|||      00001: Imagens (dados dinâmicos + ID, status)
|||      00010: Definir relógio
|||      00011: Display (somente no Modo Remoto)
|||      00100: Memória especial para debugging/visualização
|||      00101: Order book para armazenamento especial
+++----- 0 = Objeto de leitura
           1 = Objeto de escrita
    
```

A outra estrutura da mensagem consiste, primariamente, de 3 (três) bytes de controle. Em **SZ**, o número de bytes de dados é indicado e pode chegar a 8 (oito). Um ponteiro para um endereço de 16 bits é adicionado ao início de um objeto. Cada objeto pode ter um tamanho de até 64 Kb.

```

SZ = número de bytes de dados 1-8 (8 bytes)
AdrH = endereço inicial relativo ao início do objeto (High)
AdrL = endereço inicial relativo ao início do objeto (Low)
    
```

A isto se sucede uma mensagem de dados (*letter* para um equipamento Easy) de até 8 (oito) bytes de dados e um *check sum* (**LRC**), que contém a soma a partir de **AdrL**. Um equipamento Easy responde aos comandos com um byte de **ACK** (*Acknowledge*), **NACK** (*NoAcknowledge*) ou **REJ** (*Rejeição*).

```

REJ 01010000b ; REJECT + Adr.PC
NACK 00110000b ; NACK + Adr.PC
ACK 01100000b ; ACK + Adr.PC
    
```

Se um pacote de dados é solicitado, o protocolo responde no formato a seguir.

```
CTBd 00010100b ; 14h: Easy -> PC "Dados Recebidos";  
D0.D7 com até 8 (oito) bytes de dados; LRC
```

Composição de objetos e formatos

O cabeçalho de um objeto programa contém, entre outros dados, os valores de *set point* dos blocos de sistema.

Offset	Descrição
=====	
Cabeçalho do Programa	
0000h	2 Bytes Comprimento do programa (00 = Programa deletado) Low, High
0002h	1 Byte Comprimento dos parâmetros (094h = 148 bytes)
0003h	1 Byte Senha do programa
0004h	1 Byte Easy hardware mínimo requerido pelo programa
	00000000
	+-- Analógica: 1=True 0=False
	+--- Hora: 1=True 0=False
	+---- I16_Q8: 1=True 0=False
	+----- S_R_Data: 1=True 0=False
	+----- Pushbuttons usados no programa
	+----- Flag de Retenção
	+----- 0=Pushbuttons: 0=Desabilitados 1=Habilitados
	+----- 0=Debouncing: 0=Sim 1=Não
0005h	8 bytes Nome do programa
000Dh	1 byte Cabeçalho do Check sum (os primeiros 16 bytes 0-f ADDC)
000Eh	1 byte Check sum do programa
000Fh	1 byte Check sum do campo de parâmetros
Parâmetros (set point)	
0010h	1 byte Byte de controle do módulo de timer T0
	00000000
	+--- 000 = X Atraso de elevação (on delay)
	001 = ° Atraso de soltura (off delay)
	010 = ?X Aleatório - Atraso de elevação
	011 = ?° Aleatório - Atraso de soltura
	100 = _U_ Gerador de Clock
	+----- Base de tempo:
	00 = 10 ms
	01 = Segundos
	10 = Minutos
	11 = Não implementado
	+----- Não implementado
	+----- 0= - Remove do menu de parâmetros
	1= + Adiciona ao menu de parâmetros
	+----- 0= Parado
	1= Em progresso
0011h	2 bytes T0: Set point em 10 ms: S: (S*100+ms)LOW,HIGH
	T0: Set point em segundos: M:S (M*60+S) LOW,HIGH
	T0: Set point in minutos: H:M (H*60+M) LOW,HIGH
0013h	Parâmetros T2
0016h	Parâmetros T3
0019h	Parâmetros T4
001Ch	Parâmetros T5
001Fh	Parâmetros T6
0022h	Parâmetros T7
0025h	Parâmetros T8
0028h	1 byte Byte de controle do contador C0
	00000000
	+----- 0= - Remove do menu de parâmetros
	1= + Adiciona ao menu de parâmetros
	+----- 0= Parado
	1= Em progresso
0029h	2 Bytes C0: Set point: Low,High (0-9999)
002Bh	Parâmetros C2
002Eh	Parâmetros C3
0031h	Parâmetros C4
0034h	Parâmetros C5
0037h	Parâmetros C6
003Ah	Parâmetros C7
003Dh	Parâmetros C8
0040h	1 byte Byte de controle W1
	00000000
	+----- 0= Parado
	1= Em progresso
0041h	W1 1 byte Byte de controle do Canal 1, Dia da Semana Inicial-Final
	00000000
	+--- Data Inicial

```

|||||      000 = Canal: não editado
|||||      001 = Se
|||||      010 = Te
|||||      011 = Qu
|||||      100 = Qu
|||||      101 = Se
|||||      110 = Sa
|||||      111 = Do
|+---+---- Data Final
|      000 = Desabilitar
|      001 = Se
|      010 = Te
|      011 = Qu
|      100 = Qu
|      101 = Se
|      110 = Sa
|      111 = Do
+----- 1= Ton>Toff
|      0= Ton<=Toff ou Ton ou Toff
|      "Off" = (Bit.15=1)
+----- 0= - Remover do menu de parâmetros
|      1= + Adicionar ao menu de parâmetros
0042h      W1 2 bytes Canal 1, Hora/Minuto On (Low,High) Formato: Inteiro "HH*60+MM"
|      "Off" = Bit.15=1 e 10-Date=3
0044h      W1 2 bytes Canal 1, Hora/Minuto Off (Low,High.7=1:Off) Formato: Inteiro "HH*60+MM"
|      "Off" = Bit.15=1 e 10-Date=3
0046h      W1 1 byte Byte de controle do Canal 1, Dia da Semana Inicial-Final
0047h      W1 2 byte Canal 2, Hora/Minuto On
0049h      W1 2 byte Canal 2, Hora/Minuto Off
004Bh      W1 1 byte Byte de controle do Canal 2, Dia da Semana Inicial-Final
004Ch      W1 2 byte Canal 3, Hora/Minuto On
004Eh      W1 2 byte Canal 3, Hora/Minuto Off
0050h      W1 1 byte Byte de controle do Canal 3, Dia da Semana Inicial-Final
0051h      W1 2 byte Canal 4, Hora/Minuto On
0053h      W1 2 byte Canal 4, Hora/Minuto Off
0055h      Parâmetro W2
006Ah      Parâmetro W3
007Fh      Parâmetro W4
0094h      1 byte Byte de controle do comparador analógico A1
00000000
|||||+--- 0 = >=
|||||      1 = <=
|||||+--- Modo de Comparação
|||||      00 = IA1 : IA2
|||||      01 = IA1 : Const
|||||      10 = IA2 : Const
|||||      11 =
|+----- 0= - Remover do menu de parâmetros
|      1= + Adicionar ao menu de parâmetros
+----- 0= Parado
|      1= Em progresso
0095h      1 byte A1: Constante: (Formato: 0-99)
0096h      Parâmetro A2
0098h      Parâmetro A3
009Ah      Parâmetro A4
009Ch      Parâmetro A5
009Eh      Parâmetro A6
00A0h      Parâmetro A7
00A2h      Parâmetro A8
00A4h      4 bytes Compensação/Offset para o limite de paginação (16 Bytes) de EEPROMs seriais
Programa do usuário (início do diagrama)
00A8h      8 bytes por linha * 121 linhas (968 bytes)
0470h      e outras 2 linhas reservas
MMI Módulos de Texto
0480h      8 textos com 80 bytes de códigos de controle e texto (640 bytes)
06FFh      Final do objeto programa
Escrita nos endereços listados não é bloqueada.
Deste modo, podem ocorrer travamentos durante esta operação

```

Um objeto **imagem** contém os dados atuais de entradas, saídas e *flags*, entre outros, e também os valores de tempo real dos módulos de sistema.

```

Offset      Descrição
=====
Configuração Atual (determinada pelo auto-teste)
Recursos providos pelo equipamento
0000h      1 byte Configuração do Equipamento / identificador de tipo
           00000000
           |||||+-- Analógico: 1=True      0=False
           |||||+--- Relógio: 1=True     0=False
           |||||+---- I16_Q8: 1=True     0=False
           |||||+----- S_R_Data: 1=True  0=False
           |||+----- Início: 1=True    0=False
           ||+----- 1=1. Linguagem de consulta de tempo
           |+----- 1=DST ID
           +----- AC/DC: 1=AC        0=DC
           Tipo      Código Tipo      Código
           =====
           412-DC-R   01h    412-AC-R   80h
           412-DC-RC  03h    412-AC-RC  82h
           412-DC-TCX 03h    412-AC-RCX 82h
           620-DC-TC   0Fh    618-AC-RC   8Eh

EASY Status Equipamento
0001h      1 byte informação de status
           00000000
           |||||+-- Estado: 1=Executando 0=Parado
           |||||+--- Senha: 1=Senha ativa
           |||||+---- Programa: 1=Esquemático disponível
           |||||+----- Teste: 1=Teste de hardware ativado
           |||+----- Cartão: 1=Cartão de memória plugado
           ||+----- I2C: 1=I2C ocupado, acesso EEPROM
           |+----- Salvar Par. 1=Salvar parâmetros (EEPROM)
           +----- Salvar Pro. 1=Escrever programa (EEPROM) 620-DC-TCX 0Fh 618-AC-RCX 8Eh

EASY Erros/Configuração
0002h      1 byte Flags de Erros e parte 2 da configuração
           00000000
           |||||+-- EEPROM: Defeito na EEPROM
           |||||+--- EEAck: EEPROM não responde
           |||||+---- Relógio: Defeito no relógio
           |||||+----- LCD: Defeito no display ou display inexistente
           |||+----- ACLow: Nível AC abaixo do necessário
           ||+----- Rede: Desconectado/indisponível
           |+----- X-Type: 1=Tipo sem display (s. o.)
           +----- InitHW: 1=Inicialização do hardware

Configuração de Referência (requerida pelo programa)
Recursos requeridos pelo programa do equipamento
0003h      1 byte Flags de Configuração
           00000000
           |||||+-- Analógico: 1=Dispositivos analógicos em uso
           |||||+--- Relógio: 1=Relógio em uso
           |||||+---- I16Q8: 1=I>8 programa em uso
           |||||+----- SR-Operando: 1=Envio/Recebimento em uso
           |||+----- Push Buttons: 1=Push Buttons em uso
           ||+----- Retenção: 1=Retenção habilitada
           |+----- Push buttons: 1=Push Buttons habilitados
           +----- Debouncing: 1=Debouncing desabilitado

Versão / Linguagem
0004h      1 byte Versão do S0 (10h = V1.0)
0005h      1 byte Linguagem de exibição
           0: Inglês, 1: Alemão, 2: Francês, 3: Espanhol, 4: Italiano
           5: Sueco, 6: Holandês, 7: Polonês, 8: Português, 9; Turco

Dados de imagem
0006h      3 bytes Referência de relógio
           WD, Dia da semana 0-6
           HH, Hora 0-23
           MM, Minuto 0-59
0009h      2 bytes Referência das entradas analógicas I7,I8 (Formato: 0-99)
000Bh      1 byte Constant Analógica 0-k0 / N * k0 = FF (deve ser "0")
000Ch      2 bytes Imagem das entradas I0 até I16
000Eh      1 byte livre
000Fh      1 byte Push buttons
           00000000
           |||||

```

```

| | | | | | | | | | +----- Tesquerda
| | | | | | | | | | +----- Tcima
| | | | | | | | | | +----- Tbaixo
| | | | | | | | | | +----- Tdireita
| | | | | | | | | | +----- Tesc
| | | | | | | | | | +----- Tok
| | | | | | | | | | +----- Tdel
| | | | | | | | | | +----- Tpen
0010h      1 byte Imagem de saída do Timer
           OutputTimer: 0-7 (Bit 0 = T1 etc.)
0011h      1 byte Imagem de saída do Counter
           OutputCounter: 0-7
0012h      1 byte Imagem de saída do módulo de relógio W
           OutputClockWatch: 0-3
0013h      1 byte Imagem de saída do bloco analógico A
           OutputIAnalogComp: 0-7
Mais dados de imagem
0014h      Image_M0: Flag: 0-7
0015h      Image_M8: Flag: 8-F
0016h      Image_Q0: Saída Digital: 0-7
0017h      Image_S0: Enviar comunicação de saída: 0-7
0018h      Image_D0: Flag de texto: 0-7
0019h      Image_R0: Receber comunicação: 0-7
001Ah      Image_R8: Receber comunicação: 8-F
Dados de imagem para módulos de Timer
001Bh      T1: Flags de Status
           00000000 Byte de Controle
           | | | | | | | | | |
           | | | | | | | | | | +-+ 000 = X Atraso de Elevação.(ON delay)
           | | | | | | | | | |   001 = ° Atraso de soltura.(OFF delay)
           | | | | | | | | | |   010 = _|- Atraso de elevação aleatório
           | | | | | | | | | |   011 = _U_ Gerador de clock
           | | | | | | | | | |
           | | | | | | | | | | +-+ 0= MS
           | | | | | | | | | |   1= Segundos
           | | | | | | | | | | +-+ Reset-SET
           | | | | | | | | | | +-+ "SET"-Estado = SET-ALT
           | | | | | | | | | | +-+ "RES"-Estado
           | | | | | | | | | | +-+ "Tx"-Estado
           | | | | | | | | | | +----- 0= Parado
           | | | | | | | | | |   1= Incremento de INTERRUPT-TIMER
001Ch      2 bytes T1: Valor instantâneo do timer
001Eh      1 byte T1: Valor aleatório
001Fh      Timer 2 w.o.
0023h      Timer 3
0027h      Timer 4
002Bh      Timer 5
002Fh      Timer 6
0033h      Timer 7
0037h      Timer 8
Dados de imagem para módulos contadores
003Bh      1 byte C1: Flag de Status
           2 byte C1: Valor do Contador
003Eh      Contador 2 w.o.
0041h      Contador 3
0044h      Contador 4
0047h      Contador 5
004Ah      Contador 6
004Dh      Contador 7
0050h      Contador 8
Dados internos
0053h      1 byte FB Watch: Compara minutos
0054h      1 byte NetIdentCode: Identificação da extensão conectada
           0 Erro, 1 Expensão AC, 2 Expensão DC, 3 Easy200Easy, 4 Participantes da rede
0054h      Final do objeto imagem
           Escritas acima deste limite levam a um comportamento indefinido do computador
O objeto Uhr (relógio, em alemão) contém o relógio de tempo real
Offset      Descrição
=====
0000h      3 bytes Referência de relógio
           WD, Dia da semana 0-6

```

HH, Hora 0-23
MM, Minuto 0-59

A escrita deste objeto realiza a definição do horário
Com a seguinte sequência de comandos, definimos o horário atual para segunda, 12:00:
STX_ADR (C1h), CTBw Clock-Object (A2h), Number (03h), OffsetH (0), OffsetL (0),
WK (0=Monday), HH (12d), MM (0), LRC (B1h)
O Easy devolve um ACK (B0h) quando o comando é executado

Um objeto **memória especial** é utilizado para otimizar a transferência de dados para visualização.

Offset	Descrição
=====	
Array de bytes, onde cada byte representa uma área de dados de uma entrada correspondente no order book	
0000h	Tamanho do array
0001h	Dados relacionados ao primeiro array no order book
0002h	Dados para o segundo array, etc.
A memória é gerenciada dinamicamente	

O objeto **memória especial para order book** controla as entradas que um computador escreve no objeto **memória especial**.

Offset	Descrição
=====	
É manipulado usando um array de words	
0000h	Primeiro elemento de visualização especial de memória
0002h	Segundo... etc.
Format:	
aaaa aaaa-aaaa oooo	
+-----	Código de Objeto
+++++-----	12 bit Endereço Offset
A memória é gerenciada dinamicamente. O código 1Fh marca o final do objeto.	
Enviar o comando e-SUCOM "Habilita Objeto Especial Order Book (10010)" irá sobrescrever este código.	
Códigos (bytes) dos objetos de visualização no order book:	
00000:	Programa
00001:	Imagens (dados dinâmicos + ID, status)
00010:	Definir relógio
00011:	Display (somente no Modo Remoto)
00100:	Memória especial para debugging/visualização
00101:	Order book para armazenamento especial

NOTA

Formatos e conteúdos de dados inválidos enviados para um equipamento Easy podem resultar em travamento deste equipamento ou comportamentos indesejados, como por exemplo atuação de saídas. Isto pode causar acidentes e levar à morte.

Documentação das Interfaces de Comunicação

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

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

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

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

Caixa de Diálogo de Configuração

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

Aba Setup

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

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

Setup

Physical Layer: Ethernet Start driver OFFLINE

Timeout: 1000 ms Communication check time: 5000 ms

Connection management

Mode: Automatic (managed by the driver)

Retry failed connection every 20 seconds

Give up after 1 failed retries

Disconnect if non-responsive for 0 seconds

Logging Options

Log to File: C:\eeLogs\MicrolokII_%DATE%.log

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

Aba Setup

Opções gerais da aba Setup

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

Opções para o grupo Connection management

OPÇÃO	DESCRIÇÃO
Mode	Selecione 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 tentativa 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 tentativas 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 tentativas sem sucesso é zerado. Se esta nova conexão é perdida, então o contador de tentativas inicia do zero
Disconnect if non-responsive for ... seconds	Habilite esta opção para forçar um Driver a se desconectar se nenhum byte chegou à interface de I/O no <i>time-out</i> especificado, em segundos. Este <i>time-out</i> deve ser maior que o <i>time-out</i> configurado na opção Timeout

Opções para o grupo Logging Options

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

Aba Serial

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

Serial

Port:

Baud rate:

Data bits:

Parity:

Stop bits:

Enable 'ECHO' suppression

Handshaking

DTR control:

RTS control:

Wait for CTS before send

CTS timeout: ms

Delay before send: ms

Delay after send: ms

Inter-byte delay (microseconds): μ s

Inter-frame delay (milliseconds): ms

Aba Serial

Opções gerais da aba Serial

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

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

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

Opções disponíveis no grupo Handshaking

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

Aba Ethernet

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

Ethernet

Transport: TCP/IP ▼

PING before connecting

Timeout: 4000 ms

Retries: 1

Listen for connections on port: 0

Share listen port with other processes

Interface: (All Interfaces) ▼

Use IPv6 Use SSL SSL Settings

Enable 'ECHO' suppression

IP Filter:

Connect to

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

Aba Ethernet

Opções disponíveis na aba Ethernet

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

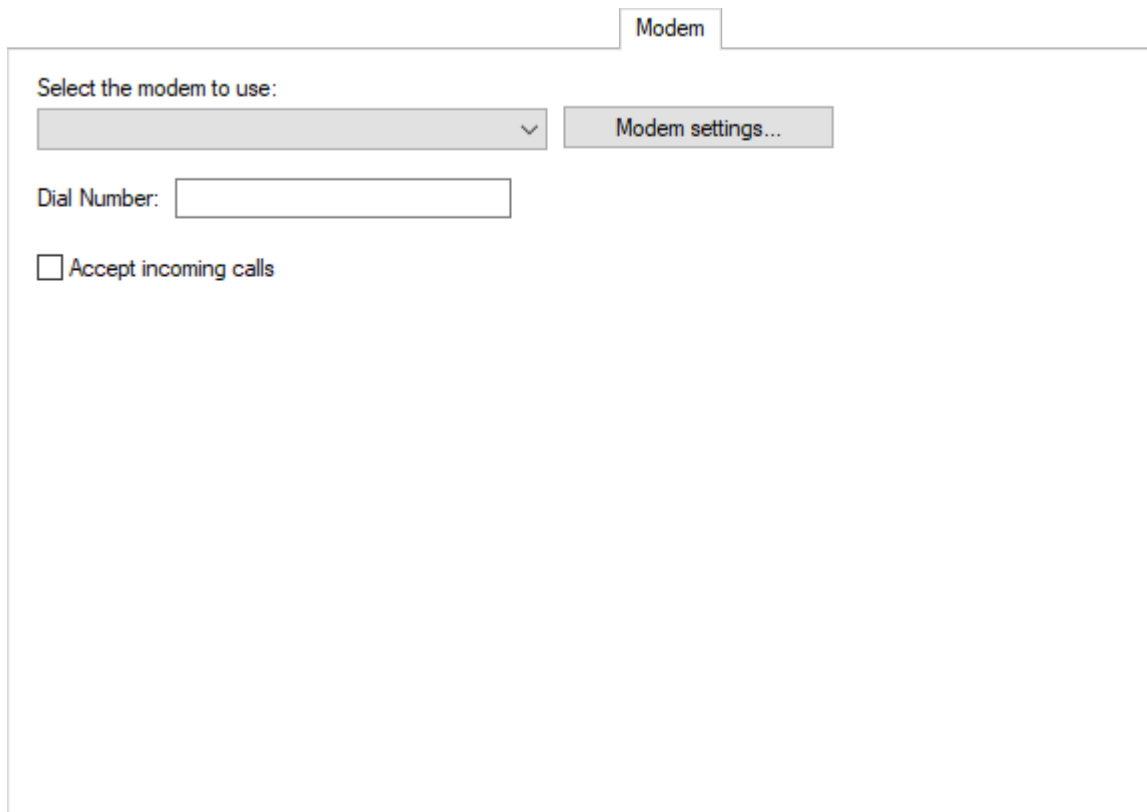
OPÇÃO	DESCRIÇÃO
IP Filter	Lista de endereços IP restringidos ou permitidos de onde um Driver aceita conexões (<i>Firewall</i>). Consulte a propriedade IO.Ethernet.IPFilter para mais informações
PING before connecting	Habilite esta opção para executar um comando ping , ou seja, para verificar se um dispositivo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

Opções disponíveis no grupo Connect to

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

Aba Modem

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



Aba Modem

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

Opções disponíveis na aba Modem

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

Aba RAS

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

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

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

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

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

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

The image shows a screenshot of a software interface with a tab labeled "RAS". Inside the tab, there are two input fields: "AT command:" followed by an empty text box, and "Connection timeout:" followed by a text box containing the number "0" and the word "seconds". Below these fields, there is a text instruction: "Other socket settings should be configured in the 'Ethernet' tab!".

Aba RAS

Opções disponíveis na aba RAS

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

Configurações Gerais

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

Tags de Comunicação

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

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

IO.CommunicationStatus

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

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

IO.IOKitEvent

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

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

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

NOTA

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

IO.PhysicalLayerStatus

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

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

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

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

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

IO.SetConfigurationParameters

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

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

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

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

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

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

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

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

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

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

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

IO.WorkOnline

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

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

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

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

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

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

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

IMPORTANTE

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

Propriedades

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

IO.ConnectionMode

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

IO.GiveUpEnable

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

IO.GiveUpTries

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

IO.InactivityEnable

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

IO.InactivityPeriodSec

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

IO.RecoverEnable

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

IO.RecoverPeriodSec

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

NOTA

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

IO.StartOffline

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

NOTA

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

IO.TimeoutMs

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

IO.Type

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

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

Configuração de Estatísticas

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

Tags de Comunicação

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

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

IO.Stats.Partial.BytesRecv

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

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

IO.Stats.Partial.BytesSent

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

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

IO.Stats.Partial.TimeConnectedSeconds

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

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

IO.Stats.Partial.TimeDisconnectedSeconds

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

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

IO.Stats.Total.BytesRecv

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

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

IO.Stats.Total.BytesSent

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

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

IO.Stats.Total.ConnectionCount

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

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

IO.Stats.Total.TimeConnectedSeconds

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

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

IO.Stats.Total.TimeDisconnectedSeconds

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

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

Propriedades

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

Configuração da Interface Ethernet

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

Tags de Comunicação

Tags da Interface Ethernet (N2/B2 = 4)

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

IMPORTANTE

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

IO.Ethernet.IPSelect

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

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

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

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

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

IO.Ethernet.IPSwitch

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

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

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

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

IO.Ethernet.SocketState

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

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

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

Propriedades

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

NOTA

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

IO.Ethernet.AcceptConnection

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

IO.Ethernet.BackupEnable[2,3]

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

desabilitar a utilização.

IO.Ethernet.BackupIP[2,3]

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

IO.Ethernet.BackupLocalPort[2,3]

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

IO.Ethernet.BackupLocalPortEnable[2,3]

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

IO.Ethernet.BackupPort[2,3]

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

IO.Ethernet.IPFilter

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

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

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

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

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

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

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

IO.Ethernet.ListenIP

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

IO.Ethernet.ListenPort

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

IO.Ethernet.MainIP

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

IO.Ethernet.MainLocalPort

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

IO.Ethernet.MainLocalPortEnable

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

IO.Ethernet.MainPort

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

IO.Ethernet.PingEnable

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

IO.Ethernet.PingTimeoutMs

9 Tempo de espera por uma resposta de um comando **ping**, em milissegundos.

IO.Ethernet.PingTries

9 Número máximo de tentativas de comandos **ping**. O valor mínimo é 1 (um), incluindo o primeiro comando **ping**.

IO.Ethernet.ShareListenPort

☑ Configure em Verdadeiro para compartilhar a porta de escuta com outros Drivers e processos ou Falso para abrir a porta de escuta em modo exclusivo. Para compartilhar uma porta de escuta com sucesso, todos os Drivers e processos envolvidos devem abrir esta porta em modo compartilhado. Quando uma porta de escuta é compartilhada, cada nova conexão é distribuída para um dos processos que estão escutando. Desta forma, se um Driver Escravo só suporta uma conexão por vez, pode-se utilizar várias instâncias deste Driver escutando na mesma porta, portanto simulando um Driver com suporte a múltiplas conexões.

IO.Ethernet.SupressEcho

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

IO.Ethernet.Transport

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

IO.Ethernet.UseIPv6

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

Configuração da Interface Modem

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

Tags de Comunicação

Tags da Interface Modem (N2/B2 = 3)

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

IMPORTANTE

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

IO.TAPI.ConnectionBaudRate

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

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

IO.TAPI.Dial

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

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

IO.TAPI.HangUp

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

Qualquer valor escrito neste Tag desliga a chamada atual.

NOTA

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

IO.TAPI.IsModemConnected

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

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

IO.TAPI.IsModemConnecting

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

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

IO.TAPI.ModemStatus

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

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

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

IO.TAPI.PhoneNumber

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

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

Propriedades

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

IO.TAPI.AcceptIncoming

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

IO.TAPI.ModemID

9 É o número de identificação do modem. Este ID é criado pelo Windows e é usado internamente para identificar o modem dentro de uma lista de equipamentos instalados no computador. Este ID pode não permanecer válido caso o modem seja reinstalado ou a aplicação seja executada em outro computador.

NOTA

Recomenda-se que esta propriedade seja configurada em 0 (zero), indicando que um Driver deve utilizar o primeiro modem disponível.

IO.TAPI.PhoneNumber

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

Configuração da Interface RAS

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

Tags de Comunicação

Tags da Interface RAS (N2/B2 = 5)

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

Propriedades

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

NOTA

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

IO.RAS.ATCommand

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

IO.RAS.CommandTimeoutSec

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

Configuração da Interface Serial

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

Tags de Comunicação

Tags da Interface Serial (N2/B2 = 2)

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

Propriedades

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

IO.Serial.Baudrate

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

IO.Serial.CTSTimeoutMs

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

IO.Serial.DataBits

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

IO.Serial.DelayAfterMs

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

IO.Serial.DelayBeforeMs

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

IO.Serial.DTR

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

IO.Serial.InterbyteDelayUs

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

IO.Serial.InterframeDelayMs

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

IO.Serial.Parity

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

IO.Serial.Port

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

IO.Serial.RTS

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

IO.Serial.StopBits

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

IO.Serial.SuppressEcho

9 Utilize um valor diferente de 0 (zero) para habilitar a supressão de eco ou 0 (zero) para desabilitá-la.

IO.Serial.WaitCTS

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

Histórico de Revisões do Driver

VERSÃO	DATA	AUTOR	COMENTÁRIOS
3.0.1	17/03/2026	M. Ludwig	<ul style="list-style-type: none"> • Driver atualizado para a biblioteca IOKit versão 3.0 e Visual Studio 2022 (<i>Case 38767</i>).
2.1.1	16/08/2013	G. Taschetto	<ul style="list-style-type: none"> • Traduzido o tópico Dicas Sucom Easy do alemão para o português (<i>Case 2845</i>).
2.0.1	23/04/2009	M. Zani M. Ludwig	<ul style="list-style-type: none"> • Driver portado para a biblioteca IOKit (<i>Case 6815</i>). • Driver portado para o Windows CE (<i>Case 10331</i>).
1.1.1	29/03/2004	C. Mello	<ul style="list-style-type: none"> • Corrigido um problema de instabilidade de comunicação, mantendo o sinal RTS sempre em nível alto.
1.0.1	15/08/2001	R. Haetinger	<ul style="list-style-type: none"> • Versão inicial deste Driver.

Matriz

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

90430-000 — Porto Alegre — RS

Fone: (+55 51) 3346-4699

Fax: (+55 51) 3222-6226

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

Filial no Paraná

Av. Sete de Setembro, 4698/1708

80240-000 — Curitiba — PR

Fone: (+55 41) 4062-5824

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

Filial no Rio de Janeiro

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

22250-044 — Rio de Janeiro — RJ

Fone: (+55 21) 2430-5912

Suporte Técnico: (+55 21) 2430-5963

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

Filial em São Paulo

Rua dos Pinheiros, 870/Cj. 141 e 142

05422-001 — São Paulo — SP

Fone: (+55 11) 3061-2828

Fax: (+55 11) 3086-2338

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

Filial em Minas Gerais

Rua Antônio de Albuquerque, 156/705

30112-010 — Belo Horizonte — MG

Fone: (+55 31) 4062-5824

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

Filial em Taiwan

9F., No.12, Beiping 2nd St., Sanmin Dist.
807 — Kaohsiung City — Taiwan

Fone: (+886 7) 323-8468

Fax: (+886 7) 323-9656

E-mail: evan@elipse.com.br

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

www.elipse.com.br

kb.elipse.com.br

forum.elipse.com.br

www.youtube.com/elipsesoftware

elipse@elipse.com.br



Gartner, Cool Vendors in Brazil 2014, April 2014.

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

Microsoft Partner

Gold Independent Software Vendor (ISV)