



IOKit

v3.00 · Português

Conteúdo

Introdução

Configurações de um Driver

Caixa de Diálogo de Configuração

Aba Setup

Aba Serial

Aba Ethernet

Aba Modem

Aba RAS

Configurações Gerais

Tags de Comunicação

IO.CommunicationStatus

IO.IOKitEvent

IO.PhysicalLayerStatus

IO.SetConfigurationParameters

IO.WorkOnline

Propriedades

IO.ConnectionMode

IO.GiveUpEnable

IO.GiveUpTries

IO.InactivityEnable

IO.InactivityPeriodSec

IO.RecoverEnable

IO.RecoverPeriodSec

IO.StartOffline

IO.TimeoutMs

IO.Type

Configuração de Estatísticas

Tags de Comunicação

IO.Stats.Partial.BytesRecv

IO.Stats.Partial.BytesSent

IO.Stats.Partial.TimeConnectedSeconds

IO.Stats.Partial.TimeDisconnectedSeconds

IO.Stats.Total.BytesRecv

IO.Stats.Total.BytesSent

IO.Stats.Total.ConnectionCount

IO.Stats.Total.TimeConnectedSeconds

IO.Stats.Total.TimeDisconnectedSeconds

Propriedades

Configuração da Interface Serial

Tags de Comunicação

Propriedades

IO.Serial.Baudrate

IO.Serial.CTSTimeoutMs

IO.Serial.DataBits

IO.Serial.DelayAfterMs

IO.Serial.DelayBeforeMs

IO.Serial.DTR

IO.Serial.InterbyteDelayUs

IO.Serial.InterframeDelayMs

IO.Serial.Parity

IO.Serial.Port

IO.Serial.RTS

IO.Serial.StopBits

IO.Serial.SuppressEcho

IO.Serial.WaitCTS

Configuração da Interface Ethernet

Tags de Comunicação

IO.Ethernet.IPSelect

IO.Ethernet.IPSwitch

IO.Ethernet.SocketState

Propriedades

IO.Ethernet.AcceptConnection

IO.Ethernet.BackupEnable[2,3]

IO.Ethernet.BackupIP[2,3]

IO.Ethernet.BackupLocalPort[2,3]

IO.Ethernet.BackupLocalPortEnable[2,3]

IO.Ethernet.BackupPort[2,3]

IO.Ethernet.IPFilter

IO.Ethernet.ListenIP

IO.Ethernet.ListenPort

IO.Ethernet.MainIP

IO.Ethernet.MainLocalPort

IO.Ethernet.MainLocalPortEnable

IO.Ethernet.MainPort

- IO.Ethernet.PingEnable
- IO.Ethernet.PingTimeoutMs
- IO.Ethernet.PingTries
- IO.Ethernet.ShareListenPort
- IO.Ethernet.SuppressEcho
- IO.Ethernet.Transport
- IO.Ethernet.UseIPv6

Configuração da Interface Modem

Tags de Comunicação

- IO.TAPI.ConnectionBaudRate
- IO.TAPI.Dial
- IO.TAPI.HangUp
- IO.TAPI.IsModemConnected
- IO.TAPI.IsModemConnecting
- IO.TAPI.ModemStatus
- IO.TAPI.PhoneNumber

Propriedades

- IO.TAPI.AcceptIncoming
- IO.TAPI.ModemID
- IO.TAPI.PhoneNumber

Configuração da Interface RAS

Tags de Comunicação

Propriedades

- IO.RAS.ATCommand
- IO.RAS.CommandTimeoutSec

Eventos do IOKit

Tópicos Avançados

Estados de um Driver

Trabalhando em Modo Offline

Gerenciamento da Conexão

- Modo Automático

- Modo Manual

- Modo Listen

- Detecção de Inatividade

Threads de um Driver

- Threads do Elipse SCADA

- Threads do Elipse E3

Knowledge Base

Enumerando a Lista de Modems

Histórico de Revisões

Introdução

Este Manual é uma referência para os usuários do **IOKit** da **Elipse Software**. Descreve como utilizar e configurar o **IOKit** tanto no **Elipse SCADA** quanto no **Elipse E3**, **Elipse Power**, **Elipse Plant Manager** e **Elipse Water**, fornecendo informações técnicas do uso prático deste componente.

O **IOKit** é um componente compartilhado utilizado pelos Drivers de I/O da **Elipse Software**, implementando o acesso padrão do nível físico e proporcionando interfaces para:

- Portas seriais
- Modem, através de TAPI
- Ethernet, utilizando WinSockets via TCP/IP ou UDP/IP
- RAS (*Remote Access Server*)

Todos os Drivers escritos utilizando o **IOKit** utilizam os recursos descritos neste Manual, como por exemplo:

- Independência de nível físico
- Geração de logs
- Configuração *offline*
- Gerenciamento de conexões

O **IOKit** é implementado como uma DLL (*Dynamic Link Library*) carregada com um Driver.

IMPORTANTE

Para evitar conflitos entre Drivers, o arquivo IOKit.dll deve estar presente **APENAS** na pasta **Windows\System32**. Cópias desta biblioteca em outras pastas podem causar comportamentos incorretos de um Driver ou do **IOKit**.

Configurações de um Driver

A configuração é realizada na caixa de diálogo de configuração de um Driver. Para acessar a configuração da caixa de diálogo no **Eclipse 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, 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 configurar a conexão de I/O que é utilizada por um Driver. Esta caixa de diálogo contém as abas 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. Consulte a documentação do Driver para mais informações sobre cada aba específica.

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 mantém a conexão e qual a política de recuperação em caso de falha
- **Logging options:** Controla a geração dos arquivos de log

The screenshot shows the 'Setup' tab of a configuration window. It is divided into four main sections:

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

At the bottom of the window, the text 'Aba Setup' is displayed.

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 <code>IO.CommunicationStatus</code>
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. Consulte o tópico Trabalhando em Modo Offline para mais informações

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

Opções para o grupo Logging Options

OPÇÃO	DESCRIÇÃO
Log to File	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**.

Serial

Port: COM1

Baud rate: 9600

Data bits: 8 data bits

Parity: None

Stop bits: 1 stop bit

☐ Enable 'ECHO' suppression

Handshaking

DTR control: OFF

RTS control: OFF

☐ Wait for CTS before send

CTS timeout: 0 ms

Delay before send: 0 ms

Delay after send: 0 ms

Inter-byte delay (microseconds): 0 μ s

Inter-frame delay (milliseconds): 0 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' supression	Habilite esta opção para remover o eco recebido após enviar dados por uma porta serial. Se o eco não é igual aos bytes recém enviados, aborta a comunicação
Inter-byte delay (microseconds)	Defina uma espera entre cada byte transmitido , 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 , em milésimos de segundo, ou seja, 1000 é igual a um segundo. Esta espera é aplicada caso 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 .

Ethernet

Transport: TCP/IP

☐ Listen for connections on port: 0

☐ PING before connecting

☐ Share listen port with other processes

Timeout: 4000 ms

☐ Interface: (All Interfaces)

Retries: 1

☐ Use IPv6 ☐ Use SSL **SSL Settings**

☐ Enable 'ECHO' suppression

IP Filter:

Connect to

Main IP: Port: 502 ☐ Local port: 0

☐ Backup IP 1: Port: 0 ☐ Local port: 0

☐ Backup IP 2: Port: 0 ☐ Local port: 0

☐ Backup IP 3: Port: 0 ☐ 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
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

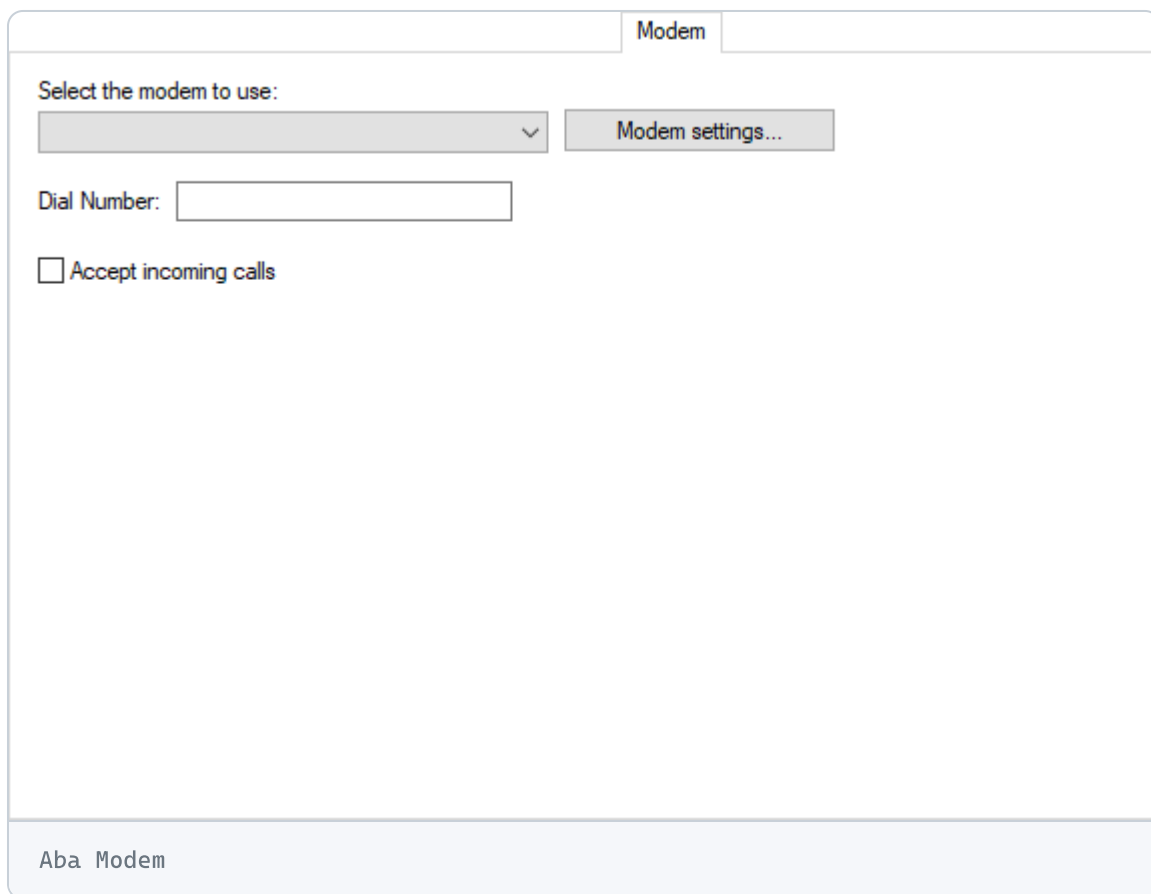
OPÇÃO	DESCRIÇÃO
PING before connecting	Habilite esta opção para executar um comando ping, ou seja, para verificar se um dispositivo pode ser encontrado na rede, em um dispositivo antes de tentar uma conexão com o <i>socket</i>. Esta é uma maneira rápida de determinar uma conexão bem sucedida antes de 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: • 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 afetam a configuração de um modem, portanto é interessante não esquecer de configurar a Interface **Serial**.



The image shows a software window titled 'Modem' at the top. Inside the window, there is a section labeled 'Select the modem to use:' followed by a dropdown menu and a button labeled 'Modem settings...'. Below this, there is a label 'Dial Number:' next to an empty text input field. Further down, there is a checkbox labeled 'Accept incoming calls'. At the bottom of the window, there is a light blue footer bar with the text '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

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.

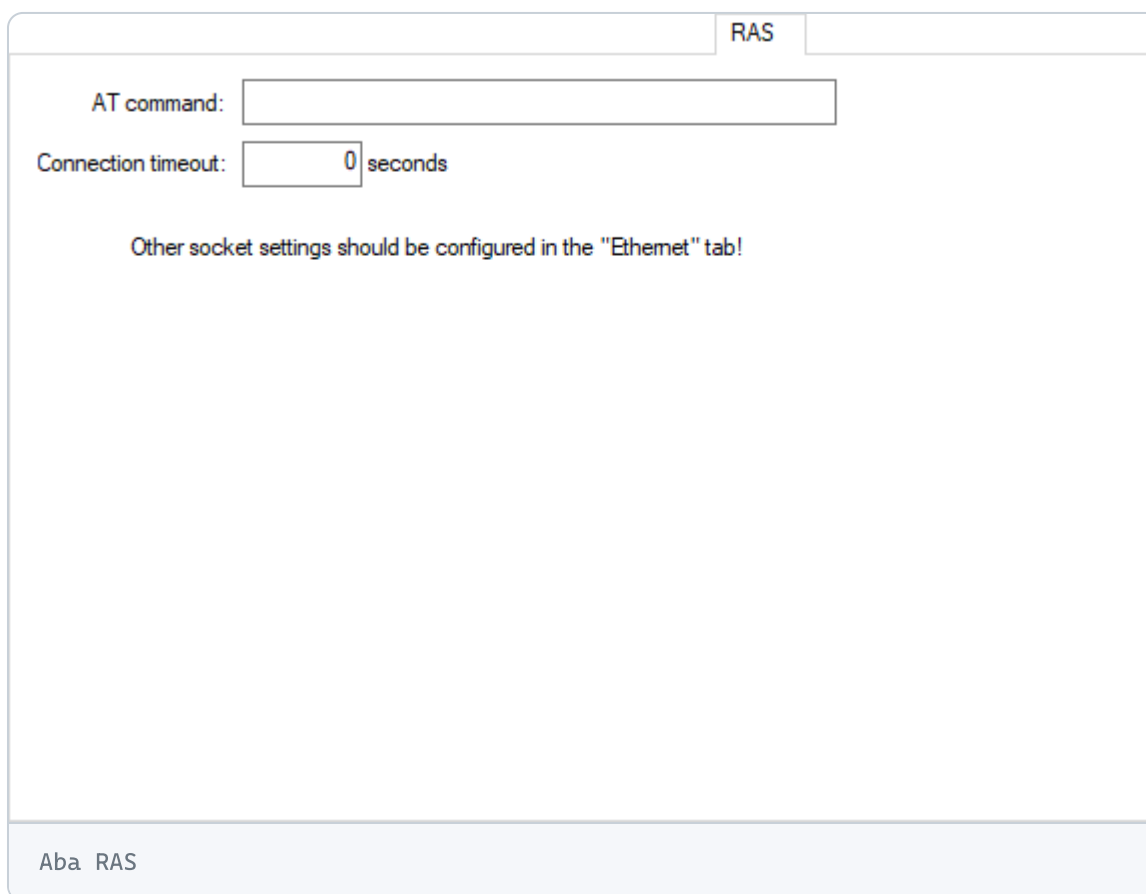
Ao conectar-se a um dispositivo RAS, primeiramente conecta ao *socket* no endereço IP e na porta configurados na aba . Depois que o *socket* é aberto, os passos de inicialização ou de conexão a seguir são efetuados:

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 software window titled "RAS" with a tab labeled "RAS". Inside the window, there are two input fields: "AT command:" followed by a text box, and "Connection timeout:" followed by a numeric box containing "0" and the text "seconds". Below these fields, a message states: "Other socket settings should be configured in the 'Ethernet' tab!". At the bottom of the window, there is a label "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 .

Tags de Comunicação

Tags Gerais (N2/B2 = 0)

Os Tags descritos a seguir são fornecidos para todas as Interfaces de I/O suportadas. Os Tags disponíveis nesta seção são os seguintes:

- IO.CommunicationStatus
- IO.IOKitEvent
- IO.PhysicalLayerStatus
- IO.SetConfigurationParameters
- IO.WorkOnline

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 . 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), **-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 **Eclipse 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. As propriedades disponíveis nesta seção são as seguintes:

- `IO.ConnectionMode`
- `IO.GiveUpEnable`
- `IO.GiveUpTries`
- `IO.InactivityEnable`
- `IO.InactivityPeriodSec`
- `IO.RecoverEnable`
- `IO.RecoverPeriodSec`
- `IO.StartOffline`
- `IO.TimeoutMs`
- `IO.Type`

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 (COMn)
 - **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 .

Tags de Comunicação

Tags de Estatísticas (N2/B2 = 0)

Os Tags descritos a seguir mostram estatísticas para . Os Tags disponíveis nesta seção são os seguintes:

- IO.Stats.Partial.BytesRecv
- IO.Stats.Partial.BytesSent
- IO.Stats.Partial.TimeConnectedSeconds
- IO.Stats.Partial.TimeDisconnectedSeconds
- IO.Stats.Total.BytesRecv
- IO.Stats.Total.BytesSent
- IO.Stats.Total.ConnectionCount
- IO.Stats.Total.TimeConnectedSeconds

- IO.Stats.Total.TimeDisconnectedSeconds

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 em tempo de execução.

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**. As propriedades disponíveis nesta seção são as seguintes:

- IO.Serial.Baudrate
- IO.Serial.CTSTimeoutMs
- IO.Serial.DataBits
- IO.Serial.DelayAfterMs
- IO.Serial.DelayBeforeMs
- IO.Serial.DTR
- IO.Serial.InterbyteDelayUs
- IO.Serial.InterframeDelayMs
- IO.Serial.Parity
- IO.Serial.Port
- IO.Serial.RTS
- IO.Serial.StopBits
- IO.Serial.SuppressEcho
- IO.Serial.WaitCTS

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

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**.

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**.

Os Tags disponíveis nesta seção são os seguintes:

- IO.Ethernet.IPSelect
- IO.Ethernet.IPSwitch

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, 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**.

As propriedades disponíveis nesta seção são as seguintes:

- IO.Ethernet.AcceptConnection
- IO.Ethernet.BackupEnable
- IO.Ethernet.BackupIP
- IO.Ethernet.BackupLocalPort
- IO.Ethernet.BackupLocalPortEnable
- IO.Ethernet.BackupPort
- IO.Ethernet.IPFILTER
- IO.Ethernet.ListenIP
- IO.Ethernet.ListenPort
- IO.Ethernet.MainIP
- IO.Ethernet.MainLocalPort
- IO.Ethernet.MainLocalPortEnable
- IO.Ethernet.MainPort
- IO.Ethernet.PingEnable
- IO.Ethernet.PingTimeoutMs
- IO.Ethernet.PingTries
- IO.Ethernet.ShareListenPort
- IO.Ethernet.SuppressEcho
- IO.Ethernet.Transport
- IO.Ethernet.UseIPv6

IO.Ethernet.AcceptConnection

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

IO.Ethernet.BackupEnable[2,3]

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

IO.Ethernet.BackupIP[2,3]

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

IO.Ethernet.BackupLocalPort[2,3]

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

IO.Ethernet.BackupLocalPortEnable[2,3]

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

IO.Ethernet.BackupPort[2,3]

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

IO.Ethernet.IPFilter

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

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

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

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

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

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

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

IO.Ethernet.ListenIP

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

IO.Ethernet.ListenPort

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

IO.Ethernet.MainIP

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

IO.Ethernet.MainLocalPort

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

IO.Ethernet.MainLocalPortEnable

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

IO.Ethernet.MainPort

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

IO.Ethernet.PingEnable

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

IO.Ethernet.PingTimeoutMs

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

IO.Ethernet.PingTries

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

IO.Ethernet.ShareListenPort

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

IO.Ethernet.SuppressEcho

● 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

A 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**.

Os Tags disponíveis nesta seção são os seguintes:

- IO.TAPI.ConnectionBaudRate

- IO.TAPI.Dial
- IO.TAPI.HangUp
- IO.TAPI.IsModemConnected
- IO.TAPI.IsModemConnecting
- IO.TAPI.ModemStatus
- IO.TAPI.PhoneNumber

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). As propriedades disponíveis nesta seção são as seguintes:

- IO.TAPI.AcceptIncoming
- IO.TAPI.ModemID
- IO.TAPI.PhoneNumber

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.

As propriedades disponíveis nesta seção são as seguintes:

- IO.RAS.ATCommand
 - IO.RAS.CommandTimeoutSec
-

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.

Eventos do IOKit

Esta seção lista os eventos atualmente reportados pelo **IOKit**. Para acessar estes eventos, deve-se declarar um Tag Bloco IO.IOKitEvent ($B1 = -1$, $B2 = 0$, $B3 = 0$, $B4 = -1$ e $Size = 4$).

Eventos reportados pelo IOKit

TIPO	FONTE	CÓDIGO	MENSAGEM
Error (2)	Serial (-2)	0 (zero)	Error opening serial port: %s • Tried to open port COM%u twice • Failed to open port %s (Windows error %u) • Failed to configure port %s (Windows error %u)
Error (2)	Modem (-3)	0 (zero)	Error opening modem line: %s • Tried to open line twice • Failed to initialize line (TAPI error %s) • Failed to negotiate API version (TAPI error %s) • Failed to open line (TAPI error %s) • Failed to configure status messages (TAPI error %s) • Failed to get line address status (TAPI error %s) • Out of memory getting line address status • Failed to get device capabilities (TAPI error %s) • Device ID = %u not found • Out of memory getting device capabilities
Info (0)	Modem (-3)	2 (dois)	Beginning of the list of available modems
Info (0)	Modem (-3)	3 (três)	%u:%s • %u is the modem ID (decimal) • %s is the modem description
Info (0)	Modem (-3)	4 (quatro)	End of the list of available modems
Error (2)	Ethernet (-4)	0 (zero)	Error connecting to %s on port %u: %s Error connecting to %s on port %u (backup IP): %s • Tried to connect socket twice • Null address • Socket error %s(%d) calling %s()
Error (2)	RAS (-5)	1 (um)	Time-out waiting for CONNECT
Info (0)	RAS (-5)	3 (três)	RAS response: '%s'
Error (2)	RAS (-5)	4 (quatro)	RAS connection error: NO CARRIER
Info (0)	IOKit (-1)	1 (um)	Initializing physical layer...
Error (2)	IOKit (-1)	2 (dois)	Physical layer initialization failed!

TIPO	FONTE	CÓDIGO	MENSAGEM
Info (0)	IOKit (-1)	3 (três)	Physical layer initialized!
Info (0)	IOKit (-1)	4 (quatro)	Connecting physical layer...
Error (2)	IOKit (-1)	5 (cinco)	Failed to connect physical layer!
Info (0)	IOKit (-1)	6 (seis)	Physical layer connected!
Error (2)	IOKit (-1)	7 (sete)	Physical layer connection lost!
Info (0)	IOKit (-1)	8 (oito)	Reconnecting physical layer...
Error (2)	IOKit (-1)	9 (nove)	Failed to reconnect physical layer!
Info (0)	IOKit (-1)	10	Physical layer reconnected!
Info (0)	IOKit (-1)	11	Physical layer connected automatically!
Info (0)	IOKit (-1)	12	Reconnecting physical layer (retry #%u)...
Error (2)	IOKit (-1)	13	Failed to reconnect physical layer (retry #%u)!
Info (0)	IOKit (-1)	14	Physical layer reconnected (on retry #%u)!
Info (0)	IOKit (-1)	15	Terminating physical layer...
Info (0)	IOKit (-1)	16	Physical layer terminated! (%u bytes sent, %u bytes received)
Info (0)	IOKit (-1)	17	Disconnecting physical layer...
Error (2)	IOKit (-1)	18	Physical layer aborted (exhausted all retries)!
Error (2)	IOKit (-1)	19	Physical layer non-responsive for %u seconds, disconnecting...

Tópicos Avançados

Este capítulo possui informações mais aprofundadas sobre o funcionamento interno do **IOKit**.

Estados de um Driver

Existem quatro estados distintos de um Driver:

- **Parado**
- **Offline** (iniciado)
- **Desconectado** (iniciado, *online*)
- **Conectado** (iniciado, *online*)

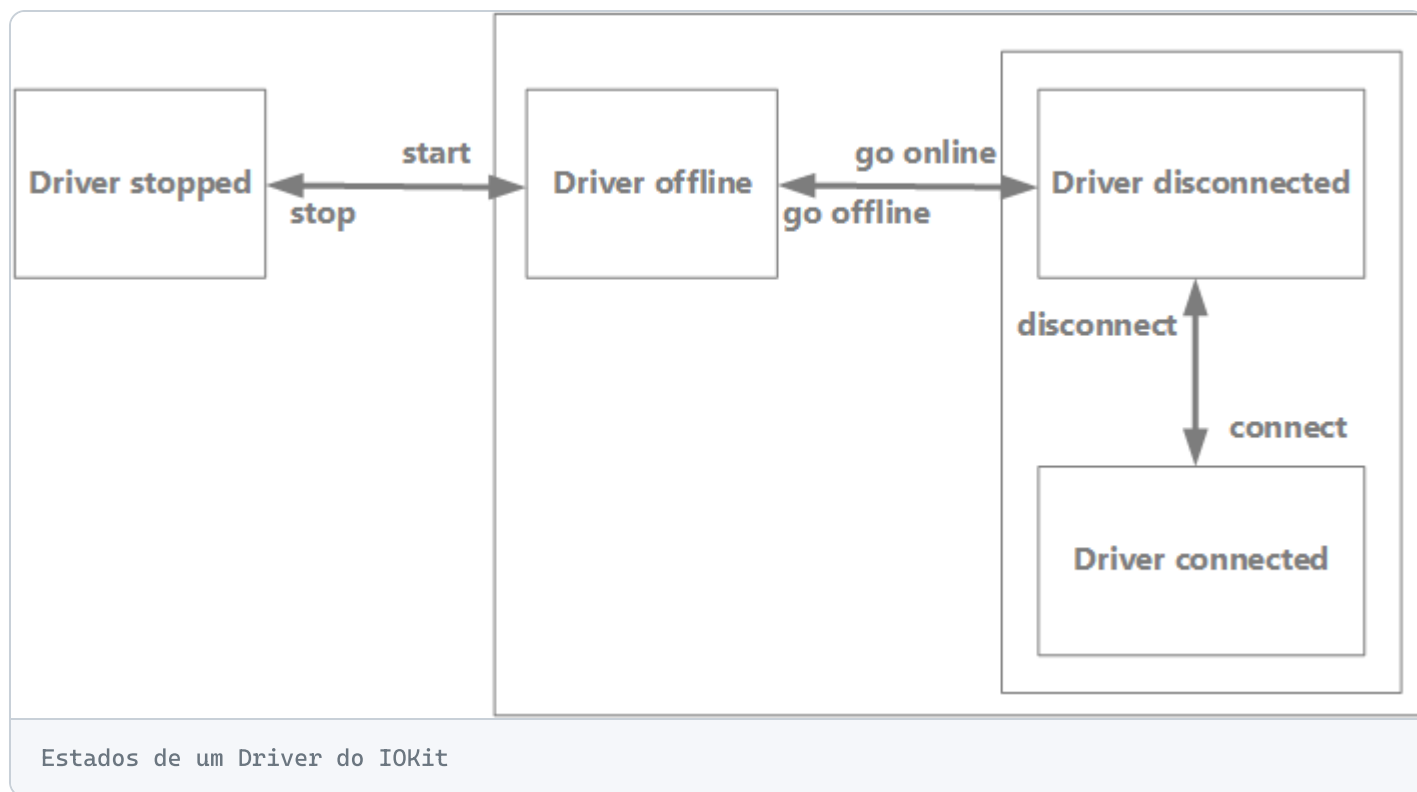
O estado **Parado** acontece quando um Driver não está em execução ou carregado, normalmente durante a configuração de uma aplicação.

Assim que uma aplicação é executada, este Driver entra no estado **Offline**. Neste estado, são detectadas as condições mínimas para a execução, ou seja, basicamente a DLL deste Driver foi carregada e este Driver espera por mais configurações de parâmetros ou por um comando para entrar em modo **Online**.

O estado **Online** pode se apresentar de duas formas diferentes, **Conectado** ou **Desconectado**. Um Driver está **Desconectado** quando a camada física ainda não é capaz de transmitir ou receber dados. Um Driver está **Conectado** quando a camada física está pronta para ser usada.

NOTA

Os estados definidos neste tópico se aplicam somente à camada física. Quando a camada física está conectada, um Driver consegue acessar um equipamento de comunicação (PLC), mas este pode ainda assim não responder. Conectar a camada física é um bom ponto de partida, mas procure sempre verificar os Tags específicos de um Driver que reportam o estado de um equipamento.



Trabalhando em Modo Offline

O modo **Offline** foi projetado especialmente para aplicações com Drivers que precisam ser configurados em tempo de execução. Dê preferência a este modo quando o tipo de conexão é ignorado, ou quando nem todos os parâmetros de conexão são conhecidos antes de uma aplicação ser executada.

Enquanto está em modo **Offline**, todos os Tags de Comunicação de um Driver falham, em todas as leituras e escritas. Os únicos Tags permitidos são os Tags Gerais do IOKit (**N1/B1 = -1** e **N2/B2 = 0**).

Embora seja possível mudar os estados **Online** e **Offline** a qualquer momento, geralmente são seguidos estes procedimentos:

1. Configure a opção Start driver OFFLINE na caixa de diálogo Extras deste Driver. Esta opção permite que este Driver comece em modo **Offline**.
2. Configure os parâmetros deste Driver no script da aplicação. O script pode ser executado automaticamente ou em resposta a um comando da interface do usuário.
3. Configure este Driver para o modo **Online**.

Gerenciamento da Conexão

Quando um Driver está configurado em modo **Online**, este entra em um estado em que tenta conectar com a camada física. Esta conexão pode ser realizada de três modos diferentes:

- Modo Automático: Um Driver gerencia a conexão de acordo com as configurações da caixa de diálogo deste Driver
- Modo Manual: Um Driver permanece desconectado e depende de comandos de uma aplicação, como por exemplo um comando *dial*, para conectar-se à camada física
- Modo Listen: Um Driver permanece desconectado e se comporta como escravo, aceitando conexões requisitadas por outros equipamentos na linha física

Esta seção também cobre a Detecção de Inatividade, que automaticamente desconecta a camada física se esta não receber nenhum byte dentro de um período especificado de tempo.

Modo Automático

O modo **Automático** pode ser selecionado de duas maneiras:

- Na caixa de diálogo de configuração de um Driver, selecionando a opção Automatic (managed by the driver) como o modo de conexão na aba **Setup**
- Configurando a propriedade IO.ConnectionMode com o valor 0 (zero, automático)

O algoritmo de conexão automática começa quando um Driver está configurado em modo **Online**. Neste momento, este Driver estabelece uma conexão inicial. A partir deste momento, este Driver passa a se comportar de acordo com as opções de gerenciamento de conexão.

Se a recuperação de conexão está habilitada na opção Retry failed connection every ... seconds na aba **Setup**, este Driver automaticamente tenta se reconectar à camada física caso a conexão seja perdida. Se a recuperação da conexão está desabilitada e a conexão é perdida, este Driver retorna ao modo **Offline**.

Pode-se também definir o número máximo de tentativas de conexão na opção Give up after ... failed connections na aba **Setup**. Isto limita o número de reconexões com a camada física. Se o número máximo de tentativas é atingido, este Driver é parado, voltando ao modo **Offline** novamente. O contador de conexões é reiniciado, ou seja, configurado para zero, quando a conexão é bem-sucedida.

Modo Manual

Quando um Driver é configurado para trabalhar no modo **Manual**, este permanece no estado desconectado após ter sido configurado em modo **Online**. Após esta configuração, pode-se forçá-lo a conectar com a camada física através dos Tags de gerenciamento de conexão.

Por exemplo, ao usar a camada física **Modem**, há uma Tela onde pode-se editar o número de telefone através de um SetPoint e um botão para iniciar a ligação. Se a ligação é completada, este Driver passa automaticamente para o estado **Conectado**. Pode-se desconectar este Driver novamente através do Tag IO.TAPI.HangUp.

Modo Listen

O modo **Listen** é quase idêntico ao modo Manual. A diferença é que a camada física é programada para aceitar conexões externas. Para a camada **Ethernet**, verifique a opção Listen for connections on port. Para a camada **Modem**, habilite a opção Accept incoming calls.

Para que o modo **Listen** funcione, é preciso configurar um Driver para trabalhar em modo **Manual**. Este permanece no estado desconectado até que a camada física indique que uma conexão está disponível. Pode-se verificar quando a conexão é estabelecida através do monitoramento do Tag IO.PhysicalLayerStatus.

Detecção de Inatividade

Um Driver pode ser programado para se desconectar automaticamente da camada física se esta permanecer inativa por um certo período de tempo. Um Driver considera que a camada física está inativa se está constantemente solicitando dados, ou seja, tentando receber caracteres sem que consiga obter nenhum dado. Se qualquer caractere é recebido, mesmo que não seja um caractere de protocolo válido, o tempo de inatividade é reiniciado.

O temporizador de inatividade é iniciado sempre que o tempo de recebimento de caracteres da camada física expira. O temporizador é reiniciado ou parado quando qualquer byte é recebido pela camada física.

O *time-out* de inatividade deve ser maior que o *time-out* da camada física, do contrário a camada física pode ser considerada inativa durante a operação normal de um Driver. Recomenda-se configurar este valor para pelo menos 10 segundos.

Threads de um Driver

Tanto o **Elipse E3** quanto o **Elipse SCADA** beneficiam-se do *multithreading* nos esquemas de aquisição de dados. Com o *multithreading*, vários Drivers podem comunicar ao mesmo tempo, sem que um bloqueie o outro. Nesta seção há uma explicação sobre os *threads* de comunicação do **Elipse E3** e do **Elipse SCADA**.

Threads do Elipse SCADA

Ao usar o **Elipse SCADA** (Elipse32.exe), os seguintes *threads* de comunicação são apresentados. Somente os *threads* relativos à operações de I/O são listados aqui:

- **Thread Principal (Main):** Este *thread* executa a interface do usuário e todos os scripts de uma aplicação. Todas as solicitações de um Driver, como leituras ou escritas disparadas por scripts ou pela interface do usuário, além de escritas assíncronas, são geradas neste *thread*. O **Thread**

Principal (Main) bloqueia estas solicitações síncronas enquanto estão sendo manipuladas por um Driver

- **Thread de um Driver:** Este *thread* é onde um Driver é executado. Existe um *thread* de Driver para cada Driver declarado em uma aplicação. Este *thread* lida com as leituras em *background* (*polling*) dos Tags, bem como as solicitações assíncronas geradas pelo **Thread Principal**. Se um Driver está configurado para o modo **16-bit Compatibility**, então este *thread* não existe e todas as solicitações síncronas e leituras em *background* são manipuladas pelo **Thread Principal (Main)**
- **Thread da Camada Física:** Este *thread* é iniciado quando um Driver entra em modo **Online**. Este *thread* lida com as solicitações de comunicação enviadas pelo **Thread de um Driver**, e também é responsável por gerenciar a conexão, como por exemplo conectar, desconectar ou tentar uma reconexão, entre outros. Solicitações do **Thread de um Driver** só podem ser manipuladas pelo **Thread da Camada Física** quando um Driver está conectado

Threads do Elipse E3

Ao usar o **Elipse E3**, cada Driver tem o próprio processo (IOServer.exe). Todos os processos de Drivers são gerenciados pelo **E3Run** (E3Run.exe). São cobertos nesta seção apenas os *threads* do **IOServer**:

- **Thread Principal (Main):** Este *thread* recebe solicitações do **E3Run**. Estas solicitações incluem iniciar ou parar um Driver, escrever comandos e iniciar ou parar a varredura de Tags
- **Thread Callback:** Este *thread* é responsável por enviar valores de volta ao **E3Run**
- **Thread de um Driver:** Este *thread*, assim como no Thread de um Driver do Elipse SCADA, é onde um Driver é efetivamente executado
- **Thread da Camada Física:** Este *thread*, assim como o Thread da Camada Física do Elipse SCADA, lida com as solicitações de comunicação enviadas pelo **Thread de um Driver**, bem como gerencia a conexão física. Este *thread* é iniciado apenas quando um Driver está em modo **Online**

Knowledge Base

Esta seção contém artigos com dicas sobre o uso do **IOKit**.

Enumerando a Lista de Modems

A propriedade `IO.TAPI.ModemID` permite ao usuário selecionar um novo modem em tempo de execução. O problema é que este ID varia de um computador para outro, e não há uma maneira simples de determinar o ID válido para um computador.

A partir da versão 1.14, o **IOKit** gera **eventos** sempre que não consegue conectar-se a um modem. Estes eventos listam o valores válidos para o modem ID do computador atual. Os procedimentos a seguir devem ser usados para enumerar a lista de modems.

Iniciar um Driver com um Modem ID Inválido

1. Configure o Driver para o modo **Offline**, selecionando a opção Start Driver OFFLINE na aba **Setup**, ou escreva o valor 0 (zero) no Tag `IO.WorkOnline` enquanto este Driver está em execução.
2. Escreva um modem ID inválido, como por exemplo -1 (menos um), na propriedade `IO.TAPI.ModemID`, usando o Tag `IO.SetConfigurationParameters`.
3. Agora o Driver entra em modo **Offline** e com um modem ID inválido configurado.
4. Configure o Driver para o modo **Online** novamente, escrevendo o valor 1 (um) no Tag `IO.WorkOnline`. Isto ativa a interface TAPI com um modem ID inválido, o que gera eventos enumerando os IDs válidos.

NOTA

Estes eventos são gerados assincronamente. Portanto, ao realizar estes passos em um script, termine-o neste ponto.

Capturar os Eventos de Enumeração dos Modem IDs Válidos

1. Para capturar estes eventos, é necessário declarar um Tag `IO.IOKitEvent` em uma aplicação. No evento **OnRead** deste Tag, deve-se verificar os eventos que reportam os modem IDs disponíveis. Deve-se procurar por eventos com tipo igual a 0 (zero, Informação) e fonte igual a -3 (menos

três, Modem). A enumeração sempre inicia com um evento com código igual a 2 (dois), seguido de um evento com código igual a 3 (três) para cada modem ID e termina com um evento com código igual a 4 (quatro).

2. A tabela a seguir lista os eventos retornados quando dois modems fictícios são encontrados, com os IDs 318 e 99383.

Eventos de enumeração

TIPO	FONTE	CÓDIGO	MENSAGEM
0 (Informação)	-3 (Modem)	2	Beginning of the list of available modems
0 (Informação)	-3 (Modem)	3	318: Modem on serial 3
0 (Informação)	-3 (Modem)	4	99383: Built-in modem
0 (Informação)	-3 (Modem)	5	End of the list of available modems

3. O evento **OnRead** pode extrair os modem IDs da mensagem para eventos com código igual a 3 (três) e armazená-los para uso posterior.

Reconfigurar o IOKit com um Modem ID Válido

1. Depois de enumerar os modem IDs válidos, pode-se continuar a configurar o Driver.
2. Primeiro, configure o Driver novamente em modo **Offline**, escrevendo o valor 0 (zero) no Tag `IO.WorkOnline`.
3. Escreva um dos modem IDs válidos enumerados na propriedade `IO.TAPI.ModemID` usando o Tag `IO.SetConfigurationParameters`.
4. Finalmente, configure o Driver novamente em modo **Online**, escrevendo o valor 1 (um) no Tag `IO.WorkOnline`.

Histórico de Revisões

VERSÃO	DATA	AUTOR	COMENTÁRIOS
1.15.4	14/08/2009	F. Englert	• Adicionado suporte para leituras por <i>callback</i> (<i>Case 8222</i>). • O IOKit agora registra uma mensagem quando o nome do arquivo de logs é modificado, como por exemplo "Log redirected to newlog.log" (<i>Case 10175</i>). • O IOKit agora registra no log a configuração de um Driver e o número da versão deste Driver e IOKit nas seguintes situações: Quando a camada física está conectada, quando o Driver é iniciado no modo Offline e quando o log é ativado ou quando o nome do arquivo de log é modificado em tempo de execução (<i>Case 10175</i>). • Agora é possível modificar o nome do arquivo de log em tempo de execução, sem precisar configurar um Driver para o modo Offline, escrevendo nos Tags Enable e Filename (<i>Case 10175</i>).
1.15.2	17/07/2008	F. Englert	• Adicionada a opção Specify local port na Interface Ethernet (<i>Case 9311</i>). • Corrigido um problema em que o IOKit pode travar se um Driver é parado enquanto está comunicando com outro <i>thread</i> (<i>Case 9423</i>).
1.15.1	28/04/2008	F. Englert	• Adicionada a opção ECHO Suppression nas Interfaces Ethernet e RAS (<i>Case 8383</i>). • A opção Retry failed connection every ... seconds agora é habilitada por padrão (<i>Case 9140</i>). • Corrigido um problema com a leitura de Tags de configuração, como por exemplo ParamItem = "IO.Ethernet.MainIP", que causava um vazamento de memória (<i>Case 8949</i>). • Corrigido um problema em que o Tag IP Switch não funcionava quando disparado logo após uma reconexão bem-sucedida (<i>Case 8012</i>). • Corrigido o texto da opção Give up after ... failed retries na aba Setup do IOKit (<i>Case 9289</i>).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
1.14.1	15/01/2007	F. Englert	• Corrigido um problema em que o IOKit pode travar quando está finalizando um Driver serial (<i>Case 5313</i>). • Melhorado o tempo de resposta para Drivers seriais escravos (<i>Case 7280</i>). • O IOKit agora enumera os modems disponíveis, com os respectivos IDs, quando um Driver é ativado com um modem ID inválido (<i>Case 7474</i>). • Corrigido um problema em que as primeiras operações de comunicação, leituras ou escritas, podiam falhar enquanto o IOKit ainda estava se conectando ao equipamento alvo (<i>Case 7614</i>). • Corrigido um problema em que o IOKit podia falhar quando um valor Float ou Double muito grande era escrito ou lido, tal como -45858873669441620000000000000.0 (<i>Case 7806</i>).
1.13.1	16/08/2006	F. Englert	• Corrigidas as estampas de tempo de valores lidos ou escritos, que podem estar incorretas em um milissegundo (<i>Case 7261</i>).
1.12.1	05/07/2006	F. Englert	• Corrigido um vazamento de memória quando um Driver retornava um HVALUE dimensionado como um <i>array</i> de HVALUES (<i>Case 7092</i>). • Corrigido um problema em que um Driver com <i>thread</i> de comunicação separado podia travar ou fechar quando o IOKit é configurado em modo Offline (<i>Case 7099</i>).
1.11.2	12/06/2006	F. Englert	• Agora é possível verificar a versão do IOKitLib com a qual um Driver foi compilado. Uma função cujo nome seja IOKitLib_v1.11 é exportada da DLL de um Driver, permitindo que a versão do IOKitLib seja facilmente identificada (<i>Case 6968</i>). • Corrigido um problema em que o <i>time-out</i> não é cumprido quando os dados são enviados através da Interface Ethernet. Este problema pode travar um Driver se o equipamento pareado não está lendo dados de um <i>socket</i> conectado (<i>Case 6935</i>). • Corrigido um problema em que a opção Disconnect if inactive não funciona se um Driver está no modo Listen (<i>Case 6933</i>). • Corrigido um problema em que não havia retentativas de reconexão se a Interface Serial não pudesse ser aberta (<i>Case 6865</i>).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
1.10.1	21/12/2005	F. Englert	• O IOKit agora gera um arquivo de log por dia se o nome do arquivo contém a macro %DATE% (<i>Case 2047</i>). • Corrigido o log do IOKit, que misturava bytes enviados ou recebidos de <i>threads</i> diferentes (<i>Case 6530</i>).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
1.9.3	20/12/2005	F. Englert	• A Interface Modem não lista ou abre um modem se o sistema tem outros modems inválidos (<i>Case 3613</i>). • Uma caixa de mensagem agora é mostrada quando o usuário habilita a opção Start driver OFFLINE, perguntando se o usuário confirma esta opção (<i>Case 4584</i>). • Os parâmetros de configuração do IOKit agora estão listados no arquivo de logs quando um Driver é iniciado (<i>Case 4584</i>). • Corrigido um problema em que os atrasos nas configurações do sinal RTS não funcionam no Windows NT (<i>Case 5525</i>). • Adicionada a opção Inter-frame delay (milliseconds) na Interface Serial, também disponível para a Interface Modem (<i>Case 5525</i>). • A mensagem registrada quando o parceiro remoto fecha o <i>socket</i>, "recv3() returned error (unknown) (0)", mudou para "Socket gracefully closed by the remote partner" (<i>Case 5599</i>). • A Interface Ethernet agora usa o WinSock v2.2 se disponível, enquanto as versões anteriores usavam o WinSock v1.1. Alguns PLCs requerem esta versão (<i>Case 5617</i>). • Adicionada a opção Enable 'ECHO' suppression para a Interface Serial, também disponível para a Interface Modem (<i>Case 5647</i>). • Adicionada a opção Modem settings à Interface Modem, permitindo ao usuário configurar um modem (<i>Case 5656</i>). • Adicionado suporte ao <i>browsing</i> dos Tags internos do IOKit, utilizado pela caixa de diálogo do Tag Browser no Eclipse E3 (<i>Case 6016</i>). • Adicionadas as opções 38700, 57600 e 115200 à lista de seleção de <i>baud rate</i> da Interface Serial (<i>Case 6076</i>). • O IOKit agora pode receber e identificar a fonte de transmissões UDP. Este recurso agora está disponível para Drivers que precisam receber transmissões UDP (<i>Case 6085</i>). • O IOKit não tenta se reconectar imediatamente se a conexão recentemente realizada é perdida (<i>Case 6294</i>). • Adicionada a opção None à lista de interfaces disponíveis, permitindo ao usuário desabilitar a camada física. Esta configuração é útil para Drivers que implementam camadas físicas personalizadas (<i>Case 6316</i>).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
1.8.2	10/01/2005	F. Englert	• Adicionado o estado Ringling e o Tag Is Modem Connecting à Interface Modem (Case 4368). • Implementadas as opções Backup IP e Ping na Interface Ethernet (Cases 3014 e 3015). • Adicionada proteção contra <i>buffer</i> excedido na geração de logs (Case 4365). • Adicionada a opção Interbyte delay nas Interfaces Serial e Modem (Case 4343). • IOKit portado para Linux e Windows CE (Case 4280). • Implementados os serviços de ativação ou desativação de logs utilizados pela caixa de diálogo Driver Manager no E3Run (Case 4513). • Logs extras foram adicionados para ajudar na depuração e diagnóstico de Drivers: comunicação desconectada indicada em sputs e sgets (Case 4851). • A versão do IOKit agora é mostrada na barra de título da janela de propriedades de um Driver ("Driver nonono (IOKit v1.08)") (Case 4778). • O estado anterior da porta serial agora é restaurado quando a porta está fechada. Isto evita problemas em outros programas que contam com a porta serial como configurada no Painel de Controle do Windows (Case 4484).
1.8.1	10/01/2005	F. Englert	• Implementada a opção UDP na interface Ethernet (Case 3017).
1.7.1	26/01/2004	F. Englert	• A Interface Serial às vezes demora um segundo antes de lidar com bytes de entrada. Isto afeta apenas Drivers escravos (Case 3279).
1.6.1	18/12/2003	F. Englert	• O IOKit agora verifica a versão de um Driver para assegurar-se que este Driver foi compilado com uma versão compatível do IOKit (Case 3018).
1.5.1	27/11/2003	F. Englert	• Implementada a opção Listen for connections on port na Interface Ethernet. Drivers escravos via Ethernet agora são suportados (Case 3018).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
1.4.1	27/10/2003	F. Englert	• A Interface RAS agora desconecta corretamente o <i>socket</i> se receber uma String "NO CARRIER" após a conexão ter sido estabelecida (<i>Case 2643</i>).
1.3.1	09/10/2003	F. Englert	• A aba Ethernet não permite que os parâmetros de porta IP sejam editados quando a Interface RAS está selecionada (<i>Case 2675</i>). • Permite que a macro %PROCESS% seja inserida no nome do arquivo de logs, permitindo que o ID do processo seja inserido no nome do arquivo. Por exemplo, c:\driver_%PROCESS%.log cria o arquivo de logs c:\driver_0000.log, onde 0 é o ID do processo (<i>Case 2676</i>). • As propriedades Enable e Filename não funcionam, e devem ser imediatamente aplicadas quando modificadas através do Tag Set Configuration Parameters (<i>Case 2678</i>).
1.2.1	07/10/2003	F. Englert	• A Interface RAS agora conecta-se com uma única porta IP, a porta especificada na aba Ethernet. O equipamento RAS verifica se há um modem disponível e atribui o modem ao <i>socket</i> ou então nega a conexão ao <i>socket</i> (<i>Case 2656</i>).
1.1.1	06/10/2003	F. Englert	• A Interface RAS agora desconecta o <i>socket</i> se receber uma String "NO CARRIER" após a conexão ter sido estabelecida (<i>Case 2643</i>). NOTA: Este recurso não funciona corretamente, utilize a versão 1.4.1.
1.0.1	11/08/2003	F. Englert	• Primeira versão. As opções Ping, Backup address, UDP, Use new socket e Accept connection de Ethernet ainda não foram implementadas.