

Driver Altus TCP/IP

Nome do Arquivo	AltusTCPIP.dll
Fabricante	Altus Sistemas de Automação S.A.
Equipamentos	UCPs Altus conectadas às interfaces AL-3405, PO7091 e WebGates PO9900 ou PO9901
Protocolo	Alnet II via TCP/IP
Versão	2.0.2
Última Atualização	11/09/2026
Plataforma	Win32
Dependências	IOMKit versão 3.0 ou superior
Leitura com Superblocos	Não
Nível	0

Introdução

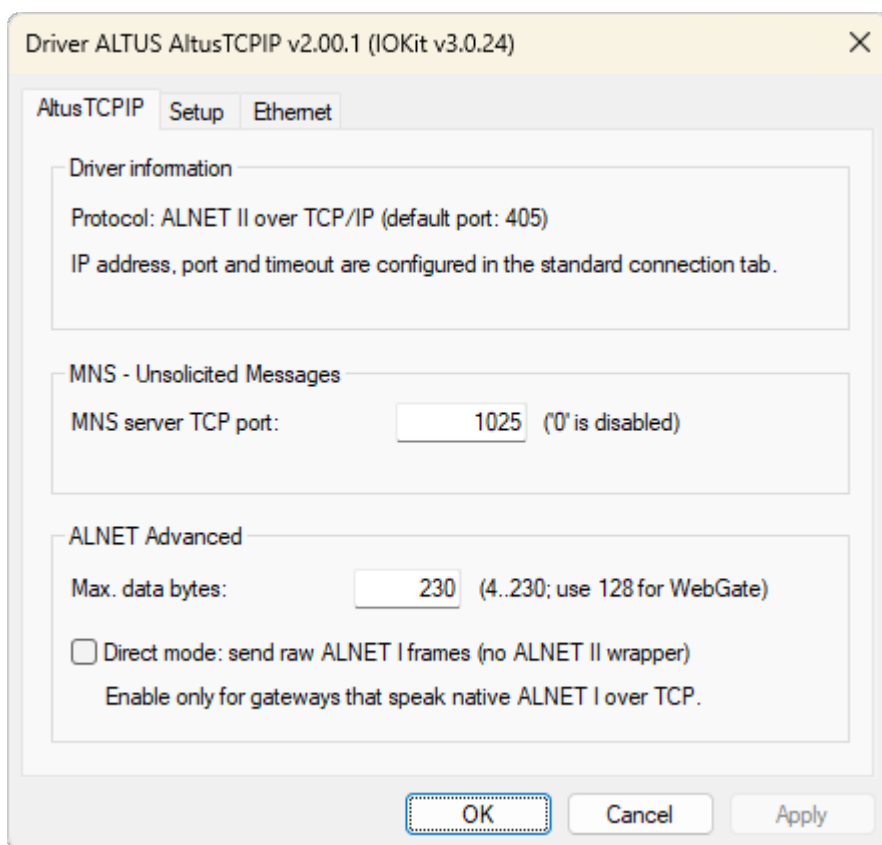
O Driver Altus TCP/IP efetua a comunicação entre sistemas da **Eclipse Software** e qualquer UCP da Altus Sistemas de Automação S.A. conectada à uma rede Ethernet através das interfaces AL-3405, PO7091 ou WebGates PO9900 ou PO9901. Este Driver também permite receber e tratar mensagens não solicitadas (MNS), isto é, decodificar comandos enviados através de instruções ECR ou ECH por uma aplicação que executa em um controlador programável.

Configuração do Driver

Os parâmetros [P] de configuração deste Driver não são utilizados. Todas as configurações devem ser realizadas na caixa de diálogo de propriedades do Driver. Para mais informações sobre as abas de configurações, consulte o tópico **Documentação das Interfaces de Comunicação**.

Configuração das Propriedades

Os parâmetros de configuração para efetuar a comunicação com um equipamento estão disponíveis na janela de propriedades deste Driver, mostrada na figura a seguir.



Aba AltusTCPIP

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

Opções disponíveis na aba AltusTCPIP

OPÇÃO	DESCRIÇÃO
MNS server TCP port	Porta TCP para conexão com um servidor MNS. Utilize o valor 0 (zero) para desabilitar mensagens não solicitadas
Max. data bytes	Tamanho máximo de dados, ou <i>payload</i> , por <i>frame</i> do protocolo ALNET II. O valor padrão desta opção é 230. Use o valor 128 para compatibilidade com equipamentos WebGate. Os limites válidos variam entre 4 (quatro, mínimo útil) e 230 (máximo do protocolo)
Direct mode	Quando esta opção está habilitada, este Driver envia frames ALNET I brutos sobre TCP/IP, sem o envoltório ALNET II, para dispositivos ou <i>gateways</i> que transmitem <i>frames</i> ALNET I nativos. O valor padrão desta opção é desabilitada

Além desta janela de propriedades, estas configurações também podem ser definidas em tempo de execução em aplicações **Elipse E3**, **Elipse Power** ou **Elipse Water**. Para isto, inicialize este Driver em modo **Offline**, ou seja, execute uma aplicação com a opção **Start driver OFFLINE** habilitada, configurável na aba **Setup** da janela de propriedades. As opções de configuração deste Driver estão descritas na tabela a seguir.

Opções de configuração do Driver Altus TCP/IP

ABA	PARÂMETRO	STRING OFFLINE	TIPO DE DADOS	DESCRIÇÃO
Setup	Physical Layer	IO.Type	Texto	Por padrão, utilize a opção Ethernet

ABA	PARÂMETRO	STRING OFFLINE	TIPO DE DADOS	DESCRIÇÃO
	Timeout	IO.TimeoutMs	Número	Limite de tempo, em milissegundos, para recepção de dados na resposta de um equipamento. Por exemplo, o valor 1000 define um limite de 1 (um) segundo
Ethernet	Transport	IO.Ethernet.Transport	Texto	Por padrão, utilize a opção TCP
	Main IP	IO.Ethernet.MainIP	Texto	Endereço IP de um equipamento, no formato [0-255].[0-255].[0-255].[0-255]
	Port	IO.Ethernet.MainPort	Número	Por padrão, utilize o valor 405
AltusTCPIP	MNS server TCP port	AltusTCPIP.MnsPort	Número	Por padrão, utilize o valor 0 (zero) para desabilitar mensagens não solicitadas
	Max. data bytes	AltusTCPIP.MaxData	Número	Por padrão, utilize o valor 230. Use o valor 128 para compatibilidade com equipamentos WebGate. Os limites válidos variam entre 4 (quatro, mínimo útil) e 230 (máximo do protocolo)
	Direct mode	AltusTCPIP.Direct	Número	Utilize o valor 0 (zero) para desabilitar e um valor diferente de 0 (zero) para habilitar. Quando esta opção está habilitada, este Driver envia <i>frames</i> ALNET I brutos sobre TCP/IP, sem o envoltório ALNET II, para dispositivos ou <i>gateways</i> que transmitem <i>frames</i> ALNET I nativos

Todas as propriedades no modo **Offline** devem ser configuradas via Tags PLC no formato **String** usando o parâmetro *N1* igual a -1 (menos um), o parâmetro *N2* igual a 0 (zero), o parâmetro *N3* igual a 0 (zero) e o parâmetro *N4* igual a 3 (três). Para mais detalhes e exemplos, consulte o tópico **Documentação das Interfaces de Comunicação**.

Referência de Tags

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

Parâmetros [N]

Utilize um Tag PLC para execução de comandos comuns, utilizando os parâmetros descritos na tabela a seguir.

N1	Deixe em 0 (zero) para mensagens solicitadas ou em 1000 para mensagens não solicitadas (MNS)
N2	Tipo de dados. Para mais informações, consulte a tabela a seguir
N3	Endereço. Para mais informações, consulte a tabela a seguir
N4	Bit ou posição de tabela. Para mais informações, consulte a tabela a seguir

O parâmetro *N1* contém o número da conexão na lista de controladores programáveis, começando em 0 (zero). Para mensagens não solicitadas (MNS), use o valor 1000 + IPPLC. A tabela a seguir contém a lista de comandos comuns suportados por este Driver.

Tipos de Tags PLC de dados

TIPO DE TAG	N1	N2	N3	N4	MODO	COMENTÁRIOS
Memória	Deixe em 0 (zero) para mensagens solicitadas ou em 1000 para mensagens não solicitadas	0 (zero)	Endereço	(*)	Leitura ou escrita	
Tabela de memória		1	Endereço	Posição	Leitura ou escrita	
Bit de memória		2	Endereço	n	Leitura ou escrita	Em desuso
Decimal		3	Endereço	(*)	Leitura ou escrita	
Tabela de decimais		4	Endereço	Posição	Leitura ou escrita	
Auxiliar		5	Endereço	(*)	Leitura ou escrita	
Entradas e saídas		6	Endereço	(*)	Leitura ou escrita	
Inteiro		7	Endereço	(*)	Leitura ou escrita	
Bit em octeto (entrada e saída)		8	Endereço	Bit, entre 0 (zero) e 7 (sete)	Somente escrita	Em desuso
Bit em auxiliar		9	Endereço	Bit, entre 0 (zero) e 7 (sete)	Somente escrita	Em desuso

TIPO DE TAG	N1	N2	N3	N4	MODO	COMENTÁRIOS
Real (float)		10	Endereço	(*)	Leitura ou escrita	
Tabela de reais (float)		11	Endereço	Posição	Leitura ou escrita	
Tabela de inteiros		12	Endereço	Posição	Leitura ou escrita	
Captura de eventos		13	Endereço TM	Aux × 10 + bit	Somente leitura	
Evento		14	Módulo, entre 0 (zero) e 15	Ponto, entre 0 (zero) e 31	Somente leitura	Requer um Tag com o parâmetro N2 igual a 13
Erro de módulo		15	0 (zero)	0 (zero)	Somente leitura	Requer um Tag com o parâmetro N2 igual a 13
Overflow de memória		16	0 (zero)	0 (zero)	Somente leitura	Requer um Tag com o parâmetro N2 igual a 13
Mudança de data e hora		17	0 (zero)	0 (zero)	Somente leitura	Requer um Tag com o parâmetro N2 igual a 13

NOTAS

- Adicione o valor 10000 ao parâmetro N2 para tratar os valores lidos como inteiros de 16 bits sem sinal, portanto os valores variam entre 0 (zero) e 65535.
- Os parâmetros N4 marcados com asterisco (*) na tabela anterior indicam que, se este parâmetro é igual a 0 (zero), todo o operando é lido. Se este parâmetro é maior que 0 (zero), um comando é executado sobre o bit especificado neste parâmetro. Os valores variam entre 1 (um) e 32, dependendo do tamanho do operando, em que o bit menos significativo é indicado pelo valor 1 (um).
- Quando o valor do parâmetro N1 está configurado para mensagens não solicitadas, ou seja, igual a 1000, os valores são inicializados e permanecem em 0 (zero) até que um controlador programável atualize os operandos correspondentes.

Parâmetros [B]

Utilize um Tag Bloco para execução de comandos comuns, utilizando os parâmetros descritos na tabela a seguir.

B1	Deixe em 0 (zero) para mensagens solicitadas ou em 1000 para mensagens não solicitadas (MNS)
B2	Tipo de dados. Para mais informações, consulte a tabela a seguir
B3	Endereço. Para mais informações, consulte a tabela a seguir
B4	Não utilizado, deixe em 0 (zero)

A tabela a seguir contém a lista de comandos comuns suportados por este Driver.

Tipos de Tags Bloco de dados

TIPO DE TAG	B1	B2	B3	MODO
Tabela de memória	Deixe em 0 (zero)	Número na tabela	Posição inicial na tabela	Leitura ou escrita
Tabela de decimais	Deixe em 0 (zero)	1000 + número na tabela	Posição inicial na tabela	Leitura ou escrita
Tabela de reais (float)	Deixe em 0 (zero)	4000 + número na tabela	Posição inicial na tabela	Leitura ou escrita
Tabela de inteiros	Deixe em 0 (zero)	6000 + número na tabela	Posição inicial na tabela	Leitura ou escrita
Tabela de memória de MNS	1000	Número na tabela	Posição inicial na tabela	Leitura ou escrita
Tabela de decimais de MNS	1000	1000 + número na tabela	Posição inicial na tabela	Leitura ou escrita
Tabela de reais (float) de MNS	1000	4000 + número na tabela	Posição inicial na tabela	Leitura ou escrita
Status barramentos de entrada e saída	Deixe em 0 (zero)	2000	Número do barramento	Leitura ou escrita
Status comunicação	Deixe em 0 (zero)	2001	Reset de variáveis (*)	Leitura ou escrita
Status equipamento	Deixe em 0 (zero)	2002	-	Leitura ou escrita
Bloco de memória	Deixe em 0 (zero)	3000	Endereço inicial	Leitura ou escrita
Bloco de real (float)	Deixe em 0 (zero)	5000	Endereço inicial	Leitura ou escrita
Bloco de entradas	Deixe em 0 (zero)	5001	Endereço inicial	Leitura ou escrita
Bloco de saídas	Deixe em 0 (zero)	5001	Endereço inicial	Somente escrita
Bloco de auxiliares	Deixe em 0 (zero)	5002	Endereço inicial	Leitura ou escrita
Bloco de inteiros	Deixe em 0 (zero)	5003	Endereço inicial	Leitura ou escrita
Bloco de decimais	Deixe em 0 (zero)	5004	Endereço inicial	Leitura ou escrita

NOTAS

- Adicione o valor 10000 ao parâmetro *B2* para tratar os valores lidos como inteiros de 16 bits sem sinal, portanto os valores variam entre 0 (zero) e 65535.
- Quando o valor do parâmetro *B1* está configurado para mensagens não solicitadas, ou seja, igual a 1000, os valores são inicializados e permanecem em 0 (zero) até que um controlador programável atualize os operandos correspondentes.
- Ao reiniciar as variáveis, os valores válidos são **0**: Os valores dos dados sobre recepção e transmissão não são alterados ou **1**: Os valores dos dados sobre recepção e transmissão são zerados.

Leitura de Eventos (SOE)

Este Driver segue a nota de aplicação NAE-040 da Altus Sistemas de Automação S.A., seção 2.3.2, no que se refere ao tratamento de eventos. Para isto são necessários os seguintes procedimentos:

1. Crie um Tag de captura de eventos, ou seja, com o parâmetro *N2* igual a 13, com um ciclo de varredura recomendado de 1000 ms. O parâmetro *N3* define qual tabela de memória de 63 posições possui os eventos, pois o *ladder* deve gravar os eventos nesta tabela. O parâmetro *N4* especifica o número do Auxiliar e o bit que é reiniciado após a leitura dos eventos. Da mesma forma, o *ladder* deve configurar este bit auxiliar para 1 (um) quando há eventos. É importante configurar corretamente os parâmetros *N3* e *N4* para um correto funcionamento da captura dos eventos. A utilização deste Tag é obrigatório para a leitura de eventos. A propriedade **Value** deste Tag contém o número de eventos encontrados na tabela de memória durante a última leitura.
2. Após a criação e leitura do Tag criado anteriormente, este Driver começa a guardar os eventos internamente. Para a leitura destes eventos, é necessário criar um Tag especificando o número da conexão, o parâmetro *N1*, o número do módulo ou placa, o parâmetro *N3*, e o número do ponto, o parâmetro *N4*. Os valores válidos são **0**: Estado 0 (zero) da entrada ou **1**: Estado 1 (um) da entrada.

Deve-se, portanto, criar um Tag para cada bit de cada módulo do qual se deseja ler os eventos. Cada evento traz na propriedade **TimeStamp** a data e a hora que um evento ocorreu em um controlador programável. Desta forma, é desejável que um controlador programável esteja com a data e hora ajustada com a data e hora do computador.

Mensagens Não Solicitadas (MNS)

O uso de mensagens não solicitadas permite melhorar significativamente a performance de uma aplicação, pois não há necessidade de realizar uma operação de *polling* nas variáveis de um controlador programável constantemente, mesmo quando estas variáveis não sofrem alteração. Ao contrário, usando mensagens não solicitadas, sempre que uma variável sofre alteração, o próprio controlador programável envia uma mensagem avisando este Driver que, por sua vez, executa a alteração na base de dados.

Uma aplicação, por sua vez, deve configurar os Tags PLC e Bloco de forma muito semelhante à configuração de Tags normais, por *polling*, mudando apenas o parâmetro *N1* ou *B1* para o valor 1000. Este número indica a este Driver que os valores destes Tags devem ser lidos não de um controlador programável, mas diretamente da base de dados deste Driver, atualizada pelas mensagens não solicitadas. Os Tags de uma aplicação, portanto, continuam funcionando por *polling*, exceto que a leitura é realizada diretamente de uma memória imagem contida neste Driver, que não gera comunicação com um controlador programável. Como esta operação é extremamente rápida, sugere-se usar um período de varredura (*scan*) muito pequeno, da ordem de milissegundos, o que permite uma atualização imediata dos Tags em caso de alteração de valor por mensagens não solicitadas. A tabela a seguir contém um exemplo de parâmetros para mensagens não solicitadas. O valor de varredura (*scan*) deve ser configurado em 10 milissegundos.

N1	1000, leitura da memória imagem
N2	0 (zero, memória)
N3	1 (um, memória M1)
N4	Não usado, deixe em 0 (zero)

Documentação das Interfaces de Comunicação

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

Configurações de um Driver

A configuração das Interfaces de Comunicação é realizada na caixa de diálogo de configuração de um Driver. Para acessar a configuração da caixa de diálogo no **Elipse E3** na versão 1.0, siga estes passos:

1. Clique com o botão direito do mouse em um objeto Driver (IODriver).
2. Selecione o item **Propriedades** no menu contextual.
3. Selecione a aba **Driver**.
4. Clique em **Outros parâmetros**.

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

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

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

Caixa de Diálogo de Configuração

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

Aba Setup

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

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

Setup

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

Opções para o grupo Logging Options

OPÇÃO	DESCRIÇÃO
Log to File	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 Eclipse 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 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

☐ 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
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

OPÇÃO	DESCRIÇÃO
	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

Configurações Gerais

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

Tags de Comunicação

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

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

IO.CommunicationStatus

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

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

IO.IOKitEvent

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

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

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

NOTA

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

IO.PhysicalLayerStatus

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

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

- **0:** Camada física parada, ou seja, um Driver está em modo **Offline**, a camada física falhou ao inicializar ou excedeu o número máximo de tentativas de reconexão
- **1:** Camada física iniciada mas não conectada, ou seja, um Driver está em modo **Online**, mas a camada física não está conectada. Se a opção **Connection management** está configurada com o valor **Automatic**, a camada física pode estar conectando, desconectando ou esperando por uma tentativa de reconexão. Se a opção **Connection management** está configurada com o valor **Manual**, então a camada física permanece neste estado até ser forçada a conectar
- **2:** Camada física conectada, ou seja, a camada física está pronta para ser usada. Isto **NÃO** significa que um equipamento esteja conectado, apenas que a camada de acesso está funcionando

IO.SetConfigurationParameters

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

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

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

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

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

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

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

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

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

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

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

IO.WorkOnline

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

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

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

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

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

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

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

IMPORTANTE

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

Propriedades

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

IO.ConnectionMode

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.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).

Histórico de Revisões do Driver

VERSÃO	DATA	AUTOR	COMENTÁRIOS
2.0.2	11/09/2026	M. Ludwig	Driver atualizado para a biblioteca IOKit versão 3.0 e Visual Studio 2022 (<i>Case 17154</i>).
1.13.1	26/01/2006	A. Corrêa A. Quites	Implementado o compartilhamento para mensagens não solicitadas (<i>Case 6126</i>).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			• Implementada a leitura de eventos (<i>Case 6143</i>). • Corrigido um problema na leitura de operandos com o parâmetro <i>N4</i> diferente de zero (<i>Case 6127</i>). • Corrigido um problema na comunicação com controladores programáveis antigos (<i>Case 6613</i>).
1.12.1	05/07/2005	A. Quites	• Implementada um lógica completa de redundância, ou <i>Hot-Standby</i> (<i>Case 4344</i>). • Melhorada a rotina de importação de configurações de versões antigas, que agora contempla todas as versões anteriores deste Driver (<i>Case 5477</i>). • Criadas opções de configuração para o processo de reconexão automática deste Driver (<i>Case 5759</i>). • Melhorado o desempenho da escrita de bits para os operandos Auxiliar, Octeto e Memória, que antes usavam mascaramento, para usar o recurso de subdivisão do protocolo (<i>Case 5833</i>). • Substituído o comando 129 (<i>Força Operando Simples</i>) pelo comando 133 (<i>Escreve Operando Simples</i>) na escrita de operandos simples, conforme orientação da Altus (<i>Case 5832</i>).
1.11.1	11/04/2005	L. Silva	• Adicionado suporte a otimização de leituras, ou agrupamento automático de Tags em Blocos (<i>Case 5521</i>). • Adicionado suporte a leituras síncronas reais mesmo quando usando <i>threads</i> de <i>read-ahead</i> (<i>Case 4977</i>).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
1.10.1	31/03/2005	A. Quites	Corrigida a importação das configurações extras de versões anteriores deste Driver. A importação continua não funcionando para Drivers publicados antes de junho de 2002 (Case 5477).
1.9.1	07/03/2005	L. Silva	- Melhorada a lógica de <i>read-head</i> para Tags com tempo de varredura (<i>scan</i>) alto, ou seja, maior que 10 segundos (Case 4969). - Corrigida a lógica da troca de endereço IP, manual ou automática, quando são usados os canais de <i>read-head</i> (Case 5089). - Criada uma opção para desabilitar a troca de endereço IP automática quando é usada redundância de controladores programáveis (Case 5089).
1.8.1	08/11/2004	A. Corrêa A. Quites	- Pequenas modificações apenas na documentação, sem alteração neste Driver. - Adicionada uma opção que desabilita a conversão para o protocolo ALNET II, tornando este Driver um Driver ALNET I comum, portanto desabilitando o recurso de mensagens não solicitadas (Case 3190). - Este Driver agora não retorna mais erro no procedimento StartCom caso ocorra um erro na abertura do arquivo de log (Case 3323). - Adicionadas leituras de bloco para os operandos Auxiliares, Decimais, Entradas e Saídas (Cases 4455 e 3932). - Implementado o operando I (inteiro) (Case 4456). - Adicionado suporte ao Elipse Drivers OPC (Case 4457).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			• Removida a opção Log Date & Time. • Adicionado o número da conexão na janela de edição (<i>Case 4458</i>). • Este Driver agora permite escrever blocos de operandos de qualquer tamanho, mesmo maiores que o tamanho máximo permitido pelo protocolo, situação em que este Driver particiona a escrita em mais de um bloco (<i>Case 4754</i>). • A escrita de blocos de mais de 64 operandos agora funciona devido a uma alteração no código de conversão ALNET I para ALNET II, que antes não estava convertendo para a instrução ALNET II mais adequada (<i>Case 4751</i>). • Implementada a redundância de conexões (<i>Case 4344</i>).
1.7.1	09/01/2004	F. Englert	• Corrigido um GPF (<i>General Protection Failure</i>) que ocorria na segunda vez que uma aplicação executava se as mensagens não solicitadas estavam habilitadas (<i>Case 3193</i>). • O padrão deste Driver agora é não gerar logs (<i>Case 3193</i>).
1.5.1	31/10/2003	F. Englert	• Habilitado o suporte a mensagens não solicitadas (MNS) (<i>Case 2805</i>). • Resolvido um problema na memória <i>cache</i> de pré-leitura, em que este Driver não descartava valores antigos de um Tag Bloco se algum Elemento era escrito (<i>Case 2838</i>).
1.4.1	29/09/2003	F. Englert	• Melhorado o log de mensagens NACK recebidas pelo protocolo.
1.3.1	26/09/2003	F. Englert	• Este Driver agora converte Strings para números nas escritas. Ao escrever um

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			valor "1234", o número 1234 é enviado. Se a String contém algum caractere não numérico ou um número mal-formatado, a escrita retorna erro.
1.2.1	05/09/2003	F. Englert	• Criada uma rotina para minimizar descarte de dados <i>read-ahead</i>. Um Tag só é lido quando falta menos de 8 (oito) segundos para a próxima varredura (<i>scan</i>).
1.1.1	02/09/2003	F. Englert	• Criadas <i>threads</i> de <i>read-ahead</i> e retentativas de leitura e escrita.
1.0.1		R. Haetinger	• Todas as publicações anteriores ao controle de revisões.

Matriz

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

90430-000 — Porto Alegre — RS

Fone: (+55 51) 3346-4699

Fax: (+55 51) 3222-6226

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

Filial no Paraná

Av. Sete de Setembro, 4698/1708

80240-000 — Curitiba — PR

Fone: (+55 41) 4062-5824

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

Filial 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 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 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)