Driver Alstom BAC

Nome do Arquivo	BAC.dll
Fabricante	Alstom
Equipamentos	Controladores Alstom
Protocolo	BAC
Versão	2.0.12
Última Atualização	14/09/2026
Plataforma	Win32
Dependências	IOKit versão 2.0 ou superior
Leitura com Superblocos	Não
Nível	31304

Introdução

O Driver Alstom BAC suporta comunicação com os Controladores Alstom. Este Driver é utilizado nos modos a seguir para comunicação entre o CCO (*Centro de Controle Operacional*), SCL/SAM (*Estações*) e UTRs (*Unidades Terminais Remotas*) do metrô de São Paulo:

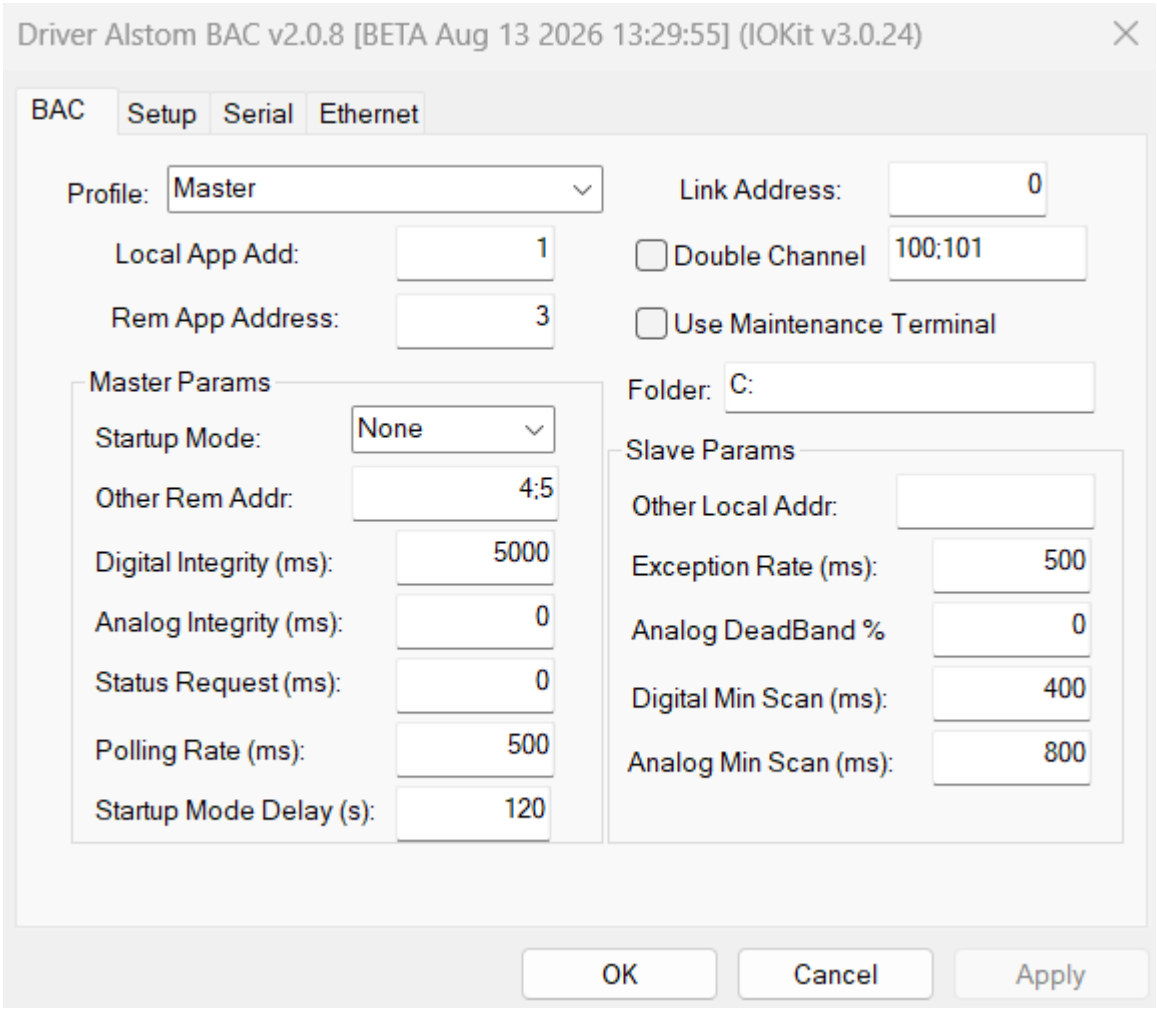
- **Master** (CCO ou Estação)
- **Slave** (UTR)
- **MasterSlave** (modo escuta)
- **Master by Files**
- **Slave Ebilock**

O modo **Escuta** é utilizado para monitorar a comunicação entre o CCO e uma UTR. O modo **Master by Files** é usado para interpretar as comunicações a partir de arquivos armazenados com o tráfego **TX/RX**.

Configuração do Driver

A configuração deste Driver é realizada na **janela de configuração** e no **endereçamento dos Tags**.

Janela de Configuração



Aba BAC

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

Opções disponíveis na aba BAC

OPÇÃO	DESCRIÇÃO
Profile	Modo de operação deste Driver. Os valores possíveis são: **Master:** Driver será o mestre da comunicação com um escravo BAC **Master by Files:** Driver se comportará como um mestre, porém os dados serão obtidos através de arquivos de log gerados pelo sistema Prodix (Metro/SP) **Slave**: Driver será o escravo da comunicação com um mestre BAC (ex: CCO do Metrô) **Slave Ebilock**: Driver será o escravo da comunicação com um mestre BAC no perfil Ebilock **Master&Slave Listen Mode:** O driver se comportará como um Mestre, porém em modo escuta (sem enviar dados), somente processando os canais TX e RX entre outro mestre e outro escravo. **Slave Listen Mode**: O driver se comportará como um Mestre, porém em modo escuta (sem enviar dados), somente processando o canal RX da comunicação entre outro mestre e outro escravo.
Link Address	Endereço da camada de enlace de dados
Local App Add	Endereço local da estação ou UTR, de acordo com o modo de operação utilizado. Deve estar no formato decimal.

OPÇÃO	DESCRIÇÃO
Rem App Address	Endereço da estação ou UTR, de acordo com o modo de operação utilizado. Deve estar no formato decimal.
Double Channel	Habilita o suporte para comunicação em duas portas de comunicação, **Serial** ou **Ethernet**. Esta opção é utilizada nos casos que exijam redundância do canal. No campo ao lado é possível informar até 2 endereços (um para cada canal) separados por ponto e vírgula, representando endereços de destino diferentes do Rem App Address nas respostas. Ex: "100;200" (sem as aspas). No caso do driver estar no perfil Mestre, o endereço IP e porta de destino do primeiro canal é informado na aba Ethernet, campo Main IP e Main Port, e do segundo canal (caso esta propriedade esteja habilitada) é informado nos campos Backup IP 1 e Backup Port 1.
Use Maintenance Terminal	Indica o uso do terminal de dados como origem, que insere 2 (dois) bytes a mais na comunicação
Folder	No modo de operação **Master by Files**, indica o diretório onde os arquivos com os logs de comunicação são buscados
Master Params - Startup Mode	Define se ao iniciar uma nova conexão, o driver deve enviar automaticamente um comando de início de modo, com as opções: **None** (nenhum comando é enviado), **Central** (C0DCh), **Local** (C0DDh) ou **Manutenção** (C0DEh).
Master Params - Other Rem Addr:	Permite indicar uma lista de endereços separados por ponto e vírgula, além do que for definido no "Rem App Address". Exemplo, o driver (como mestre) precisa conversar com as UTR´s de endereço 2,5,6 e 7. Neste caso o Rem App Address = "2", e este campo ficaria como "5;6;7" (sem as aspas).
Master Params - Digital Integrity (ms)	Intervalo de solicitação de dados digitais no modo **Mestre**, em milissegundos. O driver faz uma solicitação inicial de integridade digital sempre que houver uma partida do driver ou reconexão.
Master Params - Analog Integrity (ms)	Intervalo de solicitação de dados analógicos no modo **Mestre**, em milissegundos. O driver faz uma solicitação inicial de integridade analógica sempre que houver uma partida do driver ou reconexão.
Master Params - Status Request (ms)	Intervalo de solicitação de status no modo **Mestre**, em milissegundos. O driver faz uma solicitação inicial de status sempre que houver uma partida do driver ou reconexão.
Master Params - Polling Rate (ms)	Intervalo de *polling*, ou busca por eventos ou mudanças, no modo **Mestre**, em milissegundos. É a taxa de envio de ACKs para cada um dos escravos configurados.
Master Params - Startup Mode Delay (s)	Numero de segundos para aguardar, após a inicialização, para enviar o comando de modo (Startup Mode).
Slave Params - Other Local Addr	Permite indicar uma lista de endereços separados por ponto e vírgula, além do que for definido no "Local App Address". Exemplo, o driver (como slave) vai representar as UTR´s de endereço 2,5,6 e 7. Neste caso o Local App Address = "2", e este campo ficaria como "5;6;7" (sem as aspas).
Slave Params - Exception Rate (ms)	Intervalo mínimo para envio de exceções no modo **Escravo**, em milissegundos. Se houverem mais mudanças dentro deste intervalo, elas são acumuladas em memória até que este tempo seja ultrapassado e que um ACK tenha sido recebido do mestre.
Slave Params - Analog DeadBand %	Banda morta em porcentagem do ultimo valor enviado para considerar uma mudança analógica no modo **Escravo**
Slave Params - Digital Min Scan (ms)	Intervalo mínimo de verificação de mudanças digitais, para evitar sobrecarga do canal, no modo **Escravo**, em milissegundos. No modo Ebilock, caso o numero de mudanças ultrapasse 25% do total de pontos, uma integridade é enviada.

OPÇÃO	DESCRIÇÃO
Slave Params - Analog Min Scan (ms)	Intervalo mínimo de verificação de mudanças analógicas, para evitar sobrecarga do canal, no modo **Escravo**, em milissegundos. Na pratica, ao colocar um intervalo maior neste caso, diminuimos a quantidade de mudanças analogicas reportadas.

Referencia de Tags

Abaixo segue a configuração dos tags para cada um dos perfis.

Perfil Master

Perfil Master

Configuração Geral do Driver

P1	Se > 0, indica o intervalo em segundos para o envio espontâneo de data/hora, conforme a configuração do parâmetro P2 abaixo (50D0h para PLC e C0D8h para UTR).
P2	Se > 0, indica que o driver está comunicando com um PLC. Se = 0, é uma UTR. Esta configuração serve para indicar se a resposta a um comando 51D0h (solicitação de hora) é respondido com 50D0h (PLC) ou C0D8h (UTR).
P3	Se > 0, define o Timeout de Aplicação em milisegundos (ex: tempo de espera da execução de um comando). Se = 0, será utilizado o Timeout padrão de 5000 ms. **Importante: O parâmetro Timeout definido na Aba Setup, define o Keep Alive do canal, na camada de datalink.**
P4	Não usado

Configuração Geral de Tags

N1	Endereço de Aplicação BAC (UTR ou PLC). Ao deixar este parâmetro em 0 (zero), é usado o valor default (Rem App Address).
N2	Tipo de dado - veja tabela abaixo
N3	Endereço
N4	Bit (para comandos e estados digitais)

TAG	LEITURA OU ESCRITA	N2	N3 / COMENTÁRIOS	N4
Analógico	Somente leitura	0	Endereço/Porta, começando em 0.	Não usado (deixar em 0)
Estado digital	Somente leitura	1	Endereço/Porta, começando em 0.	Bit (0 a 7)

TAG	LEITURA OU ESCRITA	N2	N3 / COMENTÁRIOS	N4
Comando Monoestável Síncrono	Somente escrita	2	Endereço/Porta, começando em 0. Neste comando o valor do tag não é usado. A Largura do pulso pode opcionalmente ser definida no parametro Item do tag, na seguinte forma: W:NNN onde NNN é a largura em milisegundos. Ex: W:50 (50 milisegundos). Ao escrever neste tag, nenhuma outra operação é permitida, até que seja recebida uma resposta do equipamento (C1D0h) ou que ocorra timeout.	Bit (0 a 7)
Comando Biestável Síncrono	Somente escrita	3	Endereço/Porta, começando em 0. Neste comando, o valor do tag é usado, sendo 1 para ativar e 0 para desativar. Ao escrever neste tag, nenhuma outra operação é permitida, até que seja recebida uma resposta do equipamento (C1D0h) ou que ocorra timeout.	Bit (0 a 7)
Diagnósticos	Somente leitura	5	Índice do byte de diagnóstico, começando em 0 (zero) e máximo até 20 (total de 21 bytes)	Não usado
Envio de Início de Modo de Operação	Somente escrita	6	Altera o modo de operação da UTR: Valor do Tag: 1: Central 2: Local 3: Manutenção	Não usado
Envio de Termino de Modo de Operação	Somente escrita	7	Finaliza o modo de operação da UTR: Valor do Tag: 1: Central 2: Local 3: Manutenção	Não usado
Chaveamento Processador Operante/Reserva	Somente escrita	8	Envia o comando C0DBh para troca entre processador operante e reserva.	Não usado
Alarmes do CLP	Somente leitura	9	Índice do byte de diagnóstico	Não usado
Dados de Consumo de Energia	Somente escrita	10	**Reservado, porém não implementado**	
RDTD - Registrador Digital para Tarifação Diferenciada	Somente leitura	11	Endereço do Ponto (começando em 0). **Observação:** Deve ser um bloco com 4 elementos, na seguinte forma: Elemento 0 (TIME): horário relativo à leitura Elemento 1 (DELTA): Tempo de amostragem em segundos Elemento 2 (ACTIVE): Consumo Ativo	Não usado

TAG	LEITURA OU ESCRITA	N2	N3 / COMENTÁRIOS	N4
			Elemento 3 (REACTIVE): Consumo Reativo	
Comando Monoestável Assíncrono	Somente escrita	12	Endereço/Porta, começando em 0. Neste comando o valor do tag não é usado. A Largura do pulso pode opcionalmente ser definida no parametro Item do tag, na seguinte forma: W:NNN onde NNN é a largura em milisegundos. Ex: W:50 (50 milisegundos). Ao escrever neste tag, outras operações são permitidas, e quando for recebida uma resposta do equipamento (C1D0h) ou ocorrer timeout, uma leitura do tag de retorno (N2=14) será disparada.	Bit (0 a 7)
Comando Biestável Assíncrono	Somente escrita	13	Endereço/Porta, começando em 0. Neste comando, o valor do tag é usado, sendo 1 para ativar e 0 para desativar. Ao escrever neste tag, outras operações são permitidas, e quando for recebida uma resposta do equipamento (C1D0h) ou ocorrer timeout, uma leitura do tag de retorno (N2=15) será disparada.	Bit (0 a 7)

Retorno do comando Monoestável Assíncrono	somente leitura	14	Endereço/Porta, começando em 0.
			Observação: Ao iniciar um novo comando monoestável assíncrono este tag assume o valor `BAC_CMD_RESULT_PENDING = 0`
			Ao finalizar o comando, este tag assumirá um dos valores abaixo:
			`BAC_CMD_RESULT_OK 1`
			`BAC_CMD_RESULT_NOK 2`
			`BAC_CMD_RESULT_CARTAO_INEXISTENTE 3`
			`BAC_CMD_RESULT_MODO_MONITORACAO 4`
			`BAC_CMD_RESULT_MODO_CENTRAL 5`
			`BAC_CMD_RESULT_MODO_LOCAL 6`
			`BAC_CMD_RESULT_MODO_MANUTENCAO 7`
			`BAC_CMD_RESULT_TIMEOUT 99`
				Bit (0 a 7)

Retorno do comando Biestável Assíncrono	somente leitura	15	Endereço/Porta, começando em 0.
			Observação: Ao iniciar um novo comando biestável assíncrono este tag assume o valor `BAC_CMD_RESULT_PENDING = 0`
			Ao finalizar o comando, este tag assumirá um dos valores abaixo:
			`BAC_CMD_RESULT_OK 1`
			`BAC_CMD_RESULT_NOK 2`
			`BAC_CMD_RESULT_CARTAO_INEXISTENTE 3`
			`BAC_CMD_RESULT_MODO_MONITORACAO 4`
			`BAC_CMD_RESULT_MODO_CENTRAL 5`
			`BAC_CMD_RESULT_MODO_LOCAL 6`
			`BAC_CMD_RESULT_MODO_MANUTENCAO 7`
			`BAC_CMD_RESULT_TIMEOUT 99`
				Bit (0 a 7)

TAG	LEITURA OU ESCRITA	N2	N3 / COMENTÁRIOS	N4
ChannelStatus	somente leitura	17	0 = Canal Primário 1 = Canal Secundário O valor do tag é composto por 2 bits, na seguinte forma: Bit 0: Indica se o canal está ativo (trocando ACK's) Bit 1: Indica se o canal está trocando mensagens de dados (Em Uso)	Não usado

Perfil Ebilock Slave

Configuração Geral: o Parâmetro P3 do driver, se > 0, define o Timeout de Aplicação em milisegundos (ex: tempo para execução de um comando). Se = 0, será utilizado o Timeout padrão de 5000 ms. **Importante: O parâmetro Timeout definido na Aba Setup, define o Keep Alive do canal, na camada de datalink.**

O endereçamento dos Tags é realizado através da configuração dos parâmetros **[N]**.

Perfil Slave Ebilock

N1	0 (zero)
N2	Tipo de dados
N3	Endereço
N4	Endereço extra

Opções disponíveis para os parâmetros N2, N3 e N4

TAG	LEITURA OU ESCRITA	N2	N3	N4
Estado digital por Word	Somente escrita	0 (zero)	Endereço do valor no formato **Word**	0 (zero)
Estado digital por Byte	Somente escrita	1 (um)	Endereço do valor no formato **Word**	Os valores possíveis são **0**: LSB ou **1**: MSB
Comando de rota	Somente leitura	2 (dois)	Para inserir valores no formato **Hexadecimal** no **Studio**, digite "&hXXXX", como por exemplo "&hF5E1". O valor deste Tag é um **Word** com **MSB**: Param1 e **LSB**: Param2, se houver. Os valores possíveis são **Req_RT**: Requisição de rota (&hF5E1, LSB), **Can_RT**: Cancelamento de rota (&hF5E2), **Cancela_RT**: Cancela rota microlok (&hF5E3), **Req_TP**: Requisição de rota em traçado permanente (&hF5E4),	

TAG	LEITURA OU ESCRITA	N2	N3	N4
			Autoriz_RT: Autoriza rota microlok (&hF5E5), **Imp_TP**: Impõe traçado permanente (&hF5E7), **Rem_TP**: Remove traçado permanente (&hF5E8) ou **Req_CHM**: Requisição de rota de chamada (&hF5EB)	
Comando de sinaleiro	Somente leitura	3 (três)	Os valores possíveis são **Req_ENP**: Proibição de entrada de sinal (&hFCE1), **Can_ENP**: Cancela uma proibição de entrada (&hFCE2), **Rem_ENP**: Remove uma proibição de entrada (&hFCE4), **Req_EXP**: Comando de proibição de saída (&hFCE7), **Can_EXP**: Cancela uma proibição de saída (&hFCE8) ou **Rem_EXP**: Remove uma proibição de saída (&hFCEB)	
Comando de máquina chave	Somente leitura	4 (quatro)	Os valores possíveis são **Req_NWC**: Comando de posição normal (&hF3E1), **Req_RWC**: Comando de posição reversa (&hF3E2), **Auto**: Restaura o modo automático (&hF3E4), **Req_IW**: Interdita uma máquina (&hF3E7), **Can_IW**: Libera uma máquina interditada (&hF3E8) ou **Rem_IW**: Remove uma interdição de máquina (&hF3EB)	
Comando CDV	Somente leitura	5 (cinco)	Os valores possíveis são **TOCC_SIM**: Ocupação simulada em CDV (&hF9E1), **Imp_ID**: Impõe interdição de CDV (&hF9E2), **Can_ID**: Cancela interdição de CDV (&hF9E4) ou **Rem_ID**: Remove a interdição de CDV (&hF9E7)	
Comandos diversos	Somente leitura	6 (seis)	Os valores possíveis são **ITR_CEN**: Ativa ou desativa o modo **ITR** central (&hF6E1), **ITR_LOC**: Ativa ou desativa o modo **ITR** local (&hF6E2), **DISP_CEN**: Ativa ou desativa o modo despacho local (&hF6E4), **DISP_LOC**: Ativa ou desativa o modo despacho local (&hF6E7), **TM_DISP**: Despacho em TM (&hF6E8), **CAN_DISP**: Cancela o despacho em TM (&hF6EB), **TM_CMD**: Ativa ou desativa o comando TM (&hF6ED) ou **Partida_Fria**: Cancela as restrições na partida (&hF6EE)	
Comando de restrição de velocidade	Somente leitura	7 (sete)	Os valores possíveis são **SR_ON**: Impõe restrição de velocidade (&hF4E1), **SR_OFF**: Retira a restrição de velocidade (&hF4E2) ou **REM_SR**: Remove a restrição de velocidade (&hF4E3)	
Comando de inversão	Somente leitura	8 (oito)	O valor possível é **INV_SEN**: Inverte o sentido de tráfego (&hF7E1)	
Comando dummy	Somente leitura	9 (nove)	Os valores possíveis são **DS_ON**: Ativa o sinal dummy (&hF8E1) ou **DS_OFF**: Desativa o sinal dummy (&hF8E2)	
Diagnóstico de hardware	Somente escrita	10	Contém 32 bytes de diagnóstico. Informe o byte de 0 (zero) a 31	
Comunicação ativa	Somente leitura	11	Indica se o canal está trocando mensagens de dados na camada de aplicação. Os valores possíveis são **0**: Inativo ou **1**: Ativo	

Demais Perfis

Perfil Master by Files: O driver consome todos os arquivos com extensão ".DAT" que forem colocados em uma pasta, informada na propriedade "Folder". Antes do processamento, os arquivos são ordenados por data, e após o processamento são deletados. Dentro de cada arquivo, esperamos encontrar em cada linha a data/hora e os bytes da mensagem em hexadecimal. Exemplo: `29/09/2011 00:00:00 35 19 10 09 c1 db 00 0f`

Perfil Slave: Assim como no perfil Slave Ebilock, os dados de estados e analógicos devem ser escritos nos respectivos tags, e tags de comandos são notificados quando algum comando é recebido. Usar o parametro N1 para acessar endereços diferentes do Local Address (N1 = 0), declarados no campo *Other Local Addr*.

Master&Slave Listen Mode: O driver se comportará como um Mestre, porém em modo escuta (sem enviar dados), somente processando os canais TX e RX entre outro mestre e outro escravo. O driver se conectará na porta informada no Main IP/Port e espera receber nessa porta tanto as mensagens enviadas pelo mestre (TX) como as recebidas (RX), no mesmo canal. Espera-se que a propriedade Double Channel esteja desmarcada.

Slave Listen Mode: O driver se comportará como um Mestre, porém em modo escuta (sem enviar dados), somente processando o canal RX da comunicação entre outro mestre e outro escravo. O driver se conectará na porta informada no Main IP/Port e espera receber nessa porta somente as mensagens recebidas pelo Mestre (RX). Espera-se que a propriedade Double Channel esteja desmarcada.

Configuração Geral do Driver: o Parâmetro P3 do driver, se > 0, define o Timeout de Aplicação em milisegundos (ex: tempo para execução de um comando). Se = 0, será utilizado o Timeout padrão de 5000 ms. **Importante: O parâmetro Timeout definido na Aba Setup, define o Keep Alive do canal, na camada de datalink.**

Configuração geral de Tags

Leitura

N1	Endereço BAC. Se este Driver está no modo **Master**, este endereço deve ser o da UTR. Caso contrário, o endereço deve ser do CCO. Ao deixar este parâmetro em 0 (zero), são usados os valores configurados na **janela de configuração** deste Driver
N2	Tipo de dados. Os valores possíveis são **0**: Valores analógicos, **1**: Valores digitais, **2**: Comando mono-estável, **3**: Comando bi-estável ou pulso, **4**: Status ou **5**: Diagnósticos
N3	Endereço da porta, começando em 0 (zero). Para os diagnósticos, ou seja, o parâmetro *N2* igual a 5 (cinco), não é utilizado e deve ser deixado em 0 (zero)
N4	Endereço do bit da porta, com valores entre 0 (zero) e 7 (sete). Para os diagnósticos, ou seja, o parâmetro *N2* igual a 5 (cinco), deve ser informado o índice do byte de diagnóstico, começando em 0 (zero)

Escrita

N1	Endereço BAC. Se este Driver está no modo **Master**, este endereço deve ser o da UTR. Caso contrário, o endereço deve ser do CCO. Ao deixar este parâmetro em 0 (zero), são usados os valores configurados na **janela de configuração** deste Driver
N2	Tipo de dados. Os valores possíveis são **0**: Valores analógicos, **1**: Valores digitais, **2**: Comando mono-estável, **3**: Comando bi-estável, **4**: Status ou **5**: Diagnósticos
N3	Endereço da porta, começando em 0 (zero)
N4	Endereço do bit da porta, com valores entre 0 (zero) e 7 (sete)

Documentação das Interfaces de Comunicação

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

Configurações de um Driver

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

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

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

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

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

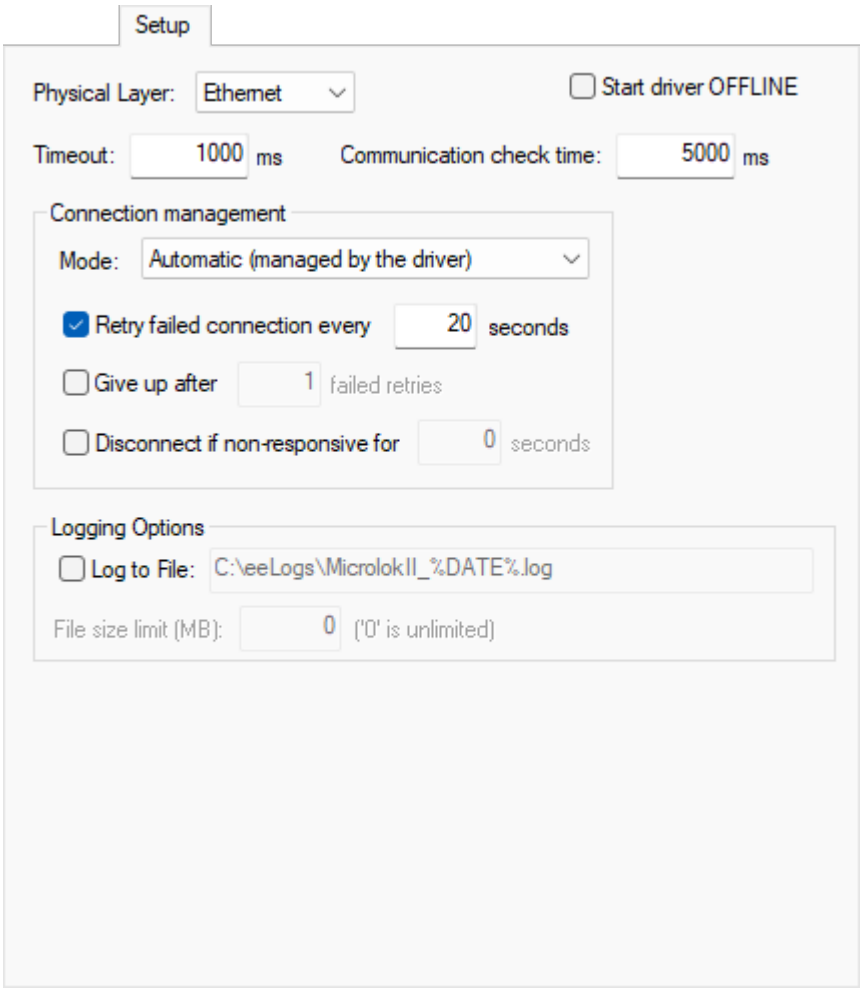
Caixa de Diálogo de Configuração

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

Aba Setup

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

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



Aba Setup

Opções gerais da aba Setup

OPÇÃO	DESCRIÇÃO
Physical Layer	Selecione a interface física em uma lista. As opções disponíveis são **Serial**, **Ethernet**, **Modem** e **RAS**. A interface selecionada deve ser configurada na aba específica
Timeout	Configure o *time-out*, em milissegundos, para a camada física. Esta é a medida de tempo que a interface de I/O aguarda para a recepção de um byte qualquer do *buffer* de recepção
Communication check time	Configure o tempo, em milissegundos, para definir o intervalo em que a comunicação é considerada em estado inativo. Enquanto um Driver de Comunicação receber dados válidos, o estado de comunicação é considerado ativo. Porém, se durante o funcionamento um Driver de Comunicação não receber dados válidos neste período de tempo, o estado é considerado inativo. O estado de comunicação é mostrado no Tag **IO.CommunicationStatus**

OPÇÃO	DESCRIÇÃO
Start driver OFFLINE	Selecione esta opção para que um Driver inicie em modo **Offline** ou parado. Isto significa que a interface de I/O não é criada até que se configure um Driver em modo **Online** utilizando-se um Tag em uma aplicação. Este modo possibilita a configuração dinâmica da interface de I/O em tempo de execução

Opções para o grupo Connection management

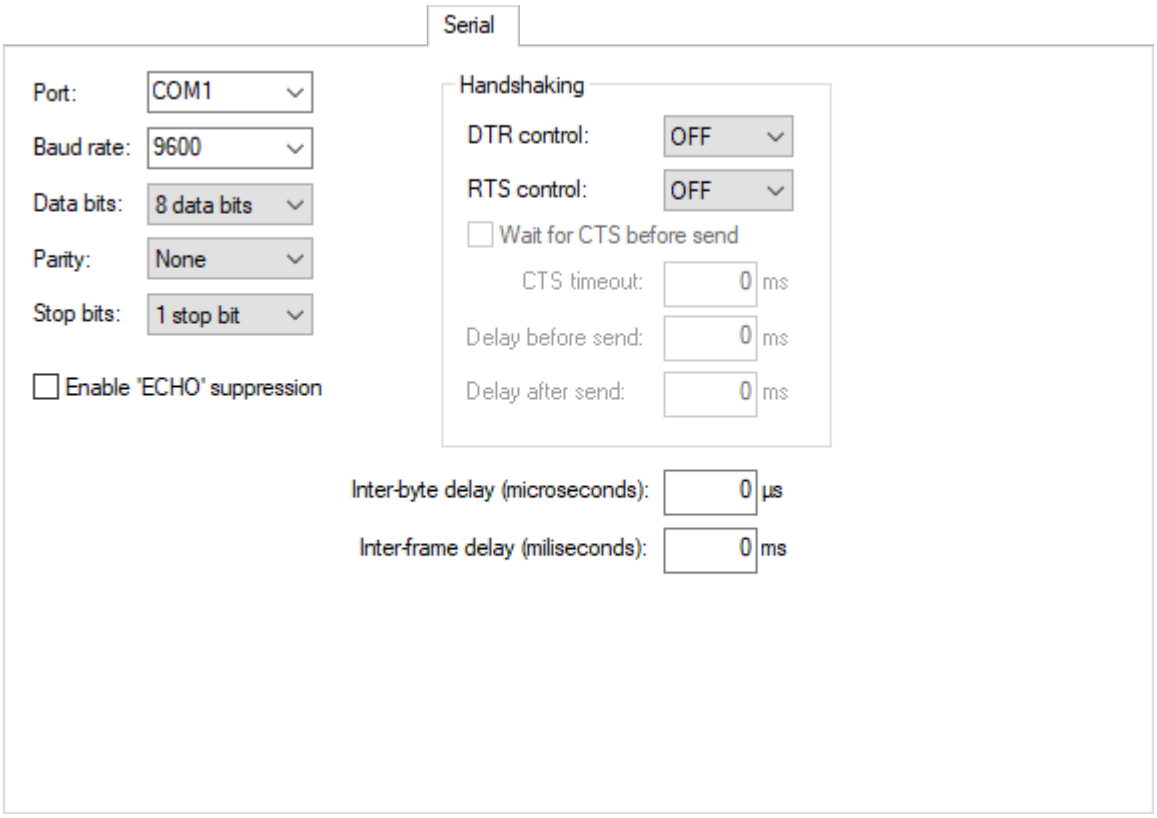
OPÇÃO	DESCRIÇÃO
Mode	Seleciona o modo de gerenciamento de conexão. Selecionar a opção **Automatic** permite que um Driver gerencie a conexão automaticamente, como especificado nas opções seguintes. Selecionar a opção **Manual** permite que uma aplicação gerencie a conexão completamente
Retry failed connection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

Opções para o grupo Logging Options

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

Aba Serial

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



Aba Serial

Opções gerais da aba Serial

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

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

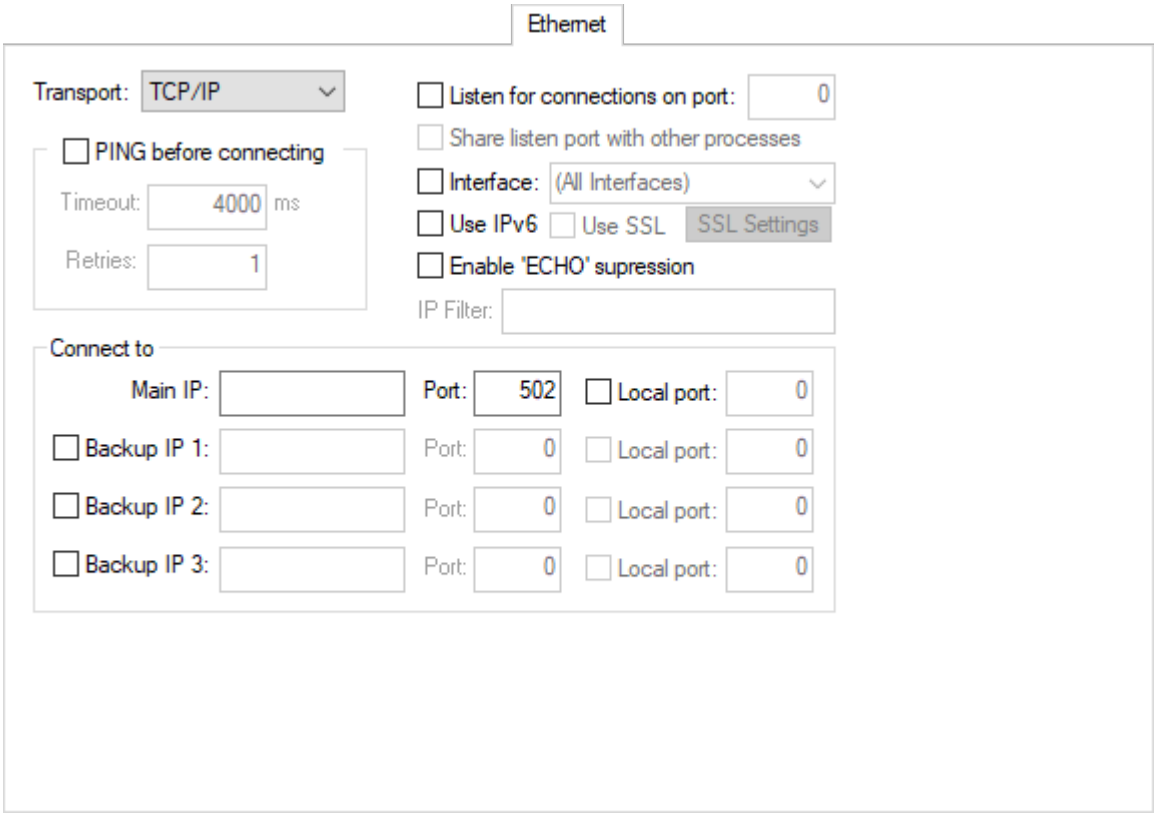
O grupo **Handshaking** configura o uso dos sinais **RTS**, **CTS** e **DTR** no processo de *handshaking* ou seja, controla quando um dado pode ser enviado ou recebido através de uma linha serial. Na maioria das vezes, configurar a opção **DTR control** para **ON** e a opção **RTS control** para **Toggle**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

Opções disponíveis no grupo Handshaking

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

Aba Ethernet

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



Aba Ethernet

Opções disponíveis na aba Ethernet

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

OPÇÃO	DESCRIÇÃO
IP Filter	Lista de endereços IP restringidos ou permitidos de onde um Driver aceita conexões (*Firewall*). Consulte a propriedade **IO.Ethernet.IPFilter** para mais informações
PING before connecting	Habilite esta opção para executar um comando **ping**, ou seja, para verificar se um dispositivo pode ser encontrado na rede, em um dispositivo antes de tentar uma conexão com o *socket*. Esta é uma maneira rápida de determinar uma conexão bem sucedida antes de tentar abrir um *socket* com um dispositivo. O *time-out* de uma conexão com um *socket* pode ser bem alto. As opções disponíveis são: • **Timeout**: Especifique o número de milissegundos de espera por uma resposta de um comando **ping**. Deve-se usar um comando **ping** para verificar o tempo normal de resposta, configurando esta opção para um valor acima desta média. Normalmente pode-se configurar um valor entre 1000 e 4000 milissegundos, ou seja entre 1 (um) e 4 (quatro) segundos • **Retries**: Número de retentativas de um comando **ping**, sem contar a tentativa inicial. Se todas as tentativas falharem, então a conexão com o *socket* é abortada

Opções disponíveis no grupo Connect to

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

Configurações Gerais

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

Tags de Comunicação

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

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

IO.CommunicationStatus

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

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

IO.IOKitEvent

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

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

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

NOTA

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

IO.PhysicalLayerStatus

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

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

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

IO.SetConfigurationParameters

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

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

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

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

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

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

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

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

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

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

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

IO.WorkOnline

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

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

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

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

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

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

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

IMPORTANTE

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

Propriedades

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

IO.ConnectionMode

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

IO.GiveUpEnable

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

IO.GiveUpTries

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

IO.InactivityEnable

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

IO.InactivityPeriodSec

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

IO.RecoverEnable

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

IO.RecoverPeriodSec

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

NOTA

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

IO.StartOffline

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

NOTA

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

IO.TimeoutMs

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

IO.Type

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

- **N ou None**: Não utiliza uma interface física, ou seja, um Driver deve fornecer uma interface personalizada
- **S ou Serial**: Utiliza uma porta serial local (COM*n*)
- **M ou Modem**: Utiliza um modem local, interno ou externo, acessado via TAPI (*Telephony Application Programming Interface*)

- **E ou Ethernet**: Utiliza um *socket* TCP/IP ou UDP/IP
- **R ou RAS**: Utiliza uma Interface **RAS** (*Remote Access Server*). Um Driver conecta-se a um equipamento RAS através da Interface **Ethernet** e então emite um comando **AT** (*dial*)

Configuração de Estatísticas

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

Tags de Comunicação

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

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

IO.Stats.Partial.BytesRecv

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

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

IO.Stats.Partial.BytesSent

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

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

IO.Stats.Partial.TimeConnectedSeconds

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

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

IO.Stats.Partial.TimeDisconnectedSeconds

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

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

IO.Stats.Total.BytesRecv

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

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

IO.Stats.Total.BytesSent

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

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

IO.Stats.Total.ConnectionCount

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

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

IO.Stats.Total.TimeConnectedSeconds

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

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

IO.Stats.Total.TimeDisconnectedSeconds

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

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

Propriedades

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

Configuração da Interface Ethernet

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

Tags de Comunicação

Tags da Interface Ethernet (N2/B2 = 4)

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

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

IO.Ethernet.IPSelect

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

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

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

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

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

IO.Ethernet.IPSwitch

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

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

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

IO.Ethernet.SocketState

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

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

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

- **Bit 1**: 0 (zero, desconectado) ou 1 (um, conectado)

Propriedades

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

NOTA

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

IO.Ethernet.AcceptConnection

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

IO.Ethernet.BackupEnable[2,3]

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

IO.Ethernet.BackupIP[2,3]

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:*:*)**: Aceita conexões de endereços IPv6 no intervalo entre fe80:03bf:0877:0000:0000:0000:0000:0000 e fe80:03bf:0877:0000:0000:0000:ffff:ffff
- **192.168.0.10, 192.168.0.15, 192.168.0.20**: Aceita conexões dos endereços IPv4 192.168.0.10, 192.168.0.15 e 192.168.0.20
- **~192.168.0.95, 192.168.0.***: Aceita conexões dos endereços IPv4 no intervalo entre 192.168.0.0 e 192.168.0.255, exceto o endereço IPv4 192.168.0.95

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

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

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

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

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

IO.Ethernet.ListenIP

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

IO.Ethernet.ListenPort

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

IO.Ethernet.MainIP

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

IO.Ethernet.MainLocalPort

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

IO.Ethernet.MainLocalPortEnable

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

IO.Ethernet.MainPort

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

IO.Ethernet.PingEnable

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

IO.Ethernet.PingTimeoutMs

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

IO.Ethernet.PingTries

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

IO.Ethernet.ShareListenPort

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

IO.Ethernet.SupressEcho

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

IO.Ethernet.Transport

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 Serial

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

Tags de Comunicação

Tags da Interface Serial (N2/B2 = 2)

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

Propriedades

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

IO.Serial.Baudrate

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

IO.Serial.CTSTimeoutMs

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

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

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

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

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

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

IO.Serial.InterframeDelayMs

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

IO.Serial.Parity

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

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

IO.Serial.RTS

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

IO.Serial.StopBits

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

IO.Serial.SupressEcho

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

IO.Serial.WaitCTS

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

Histórico de Revisões

VERSÃO	DATA	AUTOR	COMENTÁRIOS
2.0.12	14/09/2026	M. Salvador	• Revisão do driver para o projeto do CCO (Metro SP)
2.0.5	25/08/2025	M. Ludwig	• Driver atualizado para a biblioteca **IOKit** versão **3.0** e Visual Studio 2022 (*Case 37932*).
2.0.3	08/02/2024	M. Salvador	• Adicionado suporte ao protocolo **BAC Slave Ebilock** (*Case 35280*).
1.0.1	31/01/2013	M. Salvador	• Versão inicial deste Driver (*Case 12328*).

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

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

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

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

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

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

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

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

Gartner 2014 Cool Vendor

OPC FOUNDATION MEMBER

Gartner, Cool Vendors in Brazil 2014, April 2014.
Gartner does not endorse any vendor, product or service depicted in its research publications, and does not advise technology users to select only those vendors with the highest ratings. Gartner research publications consist of the opinions of Gartner's research organization and should not be construed as statements of fact. Gartner disclaims all warranties, expressed or implied, with respect to this research, including any warranties of merchantability of fitness for a particular purpose.

Microsoft Partner
Gold Independent Software Vendor (ISV)

Copyright © 2013-2026 Elipse Software Ltda. Todos os direitos reservados.