

Driver GE Ethernet SRTP

Nome do Arquivo	GEETH.dll
Fabricante	FANUC Corporation
Equipamentos	90-30, 90-70 e Versamax
Protocolo	GE SRTP
Versão	2.0.9
Última Atualização	03/09/2025
Plataforma	Win32
Dependências	IOKit versão 2.0 ou superior
Leitura de Superblocos	Sim
Nível	0

Introdução

O Driver GE Ethernet SRTP comunica-se com os CLPs GE 90-30, 90-70 e Versamax usando o protocolo GE SRTP via Ethernet.

Preparação do Equipamento

Esta seção contém informações sobre a configuração da conexão de um equipamento com este Driver. Este equipamento deve ser configurado para suportar conexões em modo **Ethernet TCP/IP**.

Conexão de um Computador com um CLP

Os equipamentos utilizados para testes deste Driver são os seguintes:

- BASE 10-SLOT (IC693CHS391D)
- SERIES 90-30 POWER SUPPLY (IC693PWR321R)
- CPU 351 (IC693CPU351-DF)
- ETHERNET CONTROLLER (IC693CMM321-BA)
- INPUT SIMULATOR MODULE (IC693ACC300D)
- OUTPUT 12/24 0.5A 32PT POS (IC693MDL753D)
- Cabo STAT. MGR to PC (IC693CBL316A)
- TRANSCEIVER

Os passos para a montagem destes equipamentos estão descritos a seguir.

1. Encaixe o **POWER SUPPLY**, **CPU 351**, **ETHERNET CONTROLLER**, **INPUT** e **OUTPUT** no *rack* (**BASE 10-SLOT**) conforme explicado no documento *Series 90-30 Programmable Controller Installation Manual*.
2. Conecte um lado do cabo **STAT** na *port 1* da **CPU** e o outro lado na porta serial do computador. Este passo só é necessário se o endereço IP do CLP ainda não foi configurado.

3. Conecte uma ponta do **TRANSCEIVER** à rede Ethernet corporativa e a outra ponta ao **ETHERNET CONTROLLER** (conexão lateral).
4. Ligue o cabo de alimentação do **POWER SUPPLY** na corrente elétrica. O *led* frontal **PWR** deve estar ligado.
5. Utilize o software **LM90** para programar o CLP e modificar o endereço IP de acordo com a rede utilizada. Para mais informações sobre o endereço IP a ser utilizado, consulte o *Appendix D - Assigning IP Addresses of Host Communications Toolkit for C/C++ Application User's Manual (GFK - 0870B)*. Se o **LM90** acusar erro de *time-out*, há erros na instalação do equipamento. Verifique a porta serial utilizada pelo **LM90** para comunicar-se.

Detectando se um CLP Possui um Endereço IP Válido

Para detectar se um CLP possui um endereço IP válido, deve-se utilizar o comando **ping** do Windows, conforme o exemplo a seguir.

```
C:\WINDOWS>ping 128.0.0.3
Pinging 128.0.0.3 with 32 bytes of data:
Reply from 128.0.0.3: bytes=32 time<10ms TTL=32
Reply from 128.0.0.3: bytes=32 time<10ms TTL=32
Reply from 128.0.0.3: bytes=32 time<10ms TTL=32
Reply from 128.0.0.3: bytes=32 time<10ms TTL=32
```

Se este comando **ping** obtém uma resposta, a configuração está correta. Senão, a resposta deste comando **ping** deve ser como no exemplo a seguir.

```
C:\WINDOWS>ping 234.2.3.18
Pinging 234.2.3.18 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
```

Neste caso, volte ao **LM90** e configure um endereço IP compatível com a rede.

Configuração do Driver

Este Driver não utiliza os parâmetros **[P]** de configuração. Todas as configurações da biblioteca **IOKit**, e as específicas deste Driver, devem ser efetuadas na caixa de diálogo de propriedades deste Driver ou na opção **Extra Settings** no **Elipse SCADA**.

Nesta caixa de diálogo de propriedades, a aba **GEETH** contém as configurações específicas deste Driver. As demais abas se referem às configurações de comunicação da biblioteca **IOKit** da **Elipse Software**.

Para mais informações sobre as configurações da biblioteca **IOKit**, consulte o tópico **Documentação das Interfaces de Comunicação**.

Configuração das Propriedades

Este tópico contém informações sobre as propriedades disponíveis na aba **GEETH**, já incluindo o valor das **Strings** de propriedades *offline*, que podem ser programadas pelo usuário ao iniciar uma aplicação em modo **Offline**.

Para aplicações **Elipse E3**, **Elipse Power** ou **Elipse Water**, o valor destas configurações também podem ser definido em tempo de execução. Para isto, inicialize este Driver em modo **Offline** selecionando a opção **Start driver OFFLINE** na aba **Setup** da caixa de diálogo de propriedades.

As opções de configuração deste Driver estão listadas na tabela a seguir.

Opções de configuração do Driver GE Ethernet SRTP

ABA	PARÂMETRO	STRING OFFLINE	TIPO DE DADOS	DESCRIÇÃO
Setup	Physical Layer	IO.Type	Texto	Por padrão, utilize a opção Ethernet
	Timeout	IO.TimeoutMs	Número	Limite de tempo, em milissegundos, para recepção dos dados na resposta do 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	Valor da porta TCP/IP de conexão Ethernet
GEETH	Source ID (Master)	GEETH.SourceID1	Número	Primeiro valor, da esquerda para a direita, referente à identificação da origem da mensagem
		GEETH.SourceID2	Número	Segundo valor, da esquerda para a direita, referente à identificação da origem da mensagem
		GEETH.SourceID3	Número	Terceiro valor, da esquerda para a direita, referente à identificação da origem da mensagem
		GEETH.SourceID4	Número	Quarto valor, da esquerda para a direita, referente à identificação da origem da mensagem
	Destination ID (Slave)	GEETH.DestinationID1	Número	Primeiro valor, da esquerda para a direita, referente à identificação do destino da mensagem
		GEETH.DestinationID2	Número	Segundo valor, da esquerda para a direita, referente à identificação do destino da mensagem

ABA	PARÂMETRO	STRING OFFLINE	TIPO DE DADOS	DESCRIÇÃO
		GEETH.DestinationID3	Número	Terceiro valor, da esquerda para a direita, referente à identificação do destino da mensagem
		GEETH.DestinationID4	Número	Quarto valor, da esquerda para a direita, referente à identificação do destino da mensagem

Todas as propriedades *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

Todos os Tags deste Driver são configurados com valores numéricos através dos parâmetros **N** e **B**.

Parâmetros [N] e [B] de Tags PLC ou Bloco

Utilize um Tag PLC ou Bloco para ler ou escrever dados na memória de um CLP.

N1/B1	Não utilizado
N2/B2	Tipo de dados. Para mais informações, consulte o tópico Tipos de Dados
N3/B3	Endereço de um ponto, geralmente começando em 1 (um)
N4/B4	Número máximo de caracteres, em caso de tipo de dados Texto , ou seja, o parâmetro <i>N2</i> igual a 27 ou 28

NOTA

Como a estrutura do protocolo GE SRTP aloca no máximo 200 bytes para a área de dados, consulte o tópico **Quantidade Máxima de Elementos por Tag Bloco** para mais informações sobre a quantidade máxima de Elementos que um Tag Bloco pode suportar, de acordo com o tipo de dados a ser processado.

Leitura do PLC Status Word

Função 999

Utilizado para consultar o último valor do **PLC Status Word** retornado por um equipamento, via script de evento **OnRead** da leitura de um Tag Bloco, conforme os parâmetros descritos na tabela a seguir.

B1	Não utilizado
B2	999
B3	Não utilizado
B4	Não utilizado

Cada evento **OnRead** retorna um Bloco com a estrutura de Elementos da tabela a seguir para interpretação do último **PLC Status Word** recebido.

Elementos disponíveis no Tag PLC Status Word

ELEMENTO	DESCRIÇÃO	VALOR
1	Sinalizador Oversweep . Significativo somente quando o modo de varredura constante está ativado	Os valores possíveis são 0 : Não existe nenhuma condição de <i>oversweep</i> ou 1 : Valor de varredura constante excedido
2	Modo de varredura constante	Os valores possíveis são 0 : Modo de varredura constante desativado ou 1 : Modo de varredura constante ativado
3	CLP Fault Entry desde a última leitura	Os valores possíveis são 0 : Tabela de falhas do CLP inalterada desde a última leitura ou 1 : Tabela de falhas do CLP foi alterada desde a última leitura deste dispositivo
4	CLP I/O Fault Entry desde a última leitura	Os valores possíveis são 0 : Tabela de falhas de I/O do CLP inalterada desde a última leitura ou 1 : Tabela de falhas de I/O do CLP foi alterada desde a última leitura deste dispositivo
5	CLP Fault Entry presente	Os valores possíveis são 0 : Tabela de falhas do CLP está vazia ou 1 : Um ou mais eventos de falha na tabela de falhas do CLP
6	CLP I/O Fault Entry presente	Os valores possíveis são 0 : Tabela de falhas de I/O do CLP está vazia ou 1 : Um ou mais eventos de falha na tabela de falhas de I/O do CLP
7	Indicador de anexo do programador	Os valores possíveis são 0 : Nenhum anexo do programador encontrado ou 1 : Anexo do programador encontrado
8	Habilita ou desabilita a configuração do interruptor frontal do painel	Os valores possíveis são 0 : Saídas habilitadas ou 1 : Saídas desabilitadas
9	Inicia ou para a configuração do interruptor frontal do painel	Os valores possíveis são 0 : Stop ou 1 : Run
10	OEM protegido	Os valores possíveis são 0 : Nenhuma proteção de OEM ou 1 : Proteção de OEM em vigor

ELEMENTO	DESCRIÇÃO	VALOR
11	Não utilizado	0 (zero)
12	Não utilizado	0 (zero)
13	Status do CLP	Os valores possíveis são 0 : Run I/O habilitado, 1 : Run I/O desabilitado, 2 : Stop I/O desabilitado, 3 : CPU parada por falha, 4 : CPU parada, 5 : CPU suspensa ou 6 : Stop I/O habilitado

Coleta do Histórico de Eventos

Função 200

Utilizado para coletar vários registros de histórico do SER PLC, via script de evento **OnRead** da leitura de um Tag Bloco, conforme os parâmetros descritos na tabela a seguir.

B1	Não utilizado
B2	200
B3	Não utilizado
B4	Não utilizado

Cada evento **OnRead** retorna um Bloco com a seguinte estrutura de Elementos, caso haja algum evento disponível:

- **Elemento 0**: Número do circuito
- **Elemento 1**: Estado de Transição. Os valores possíveis são **1**: Normal ou **0**: Alarme

NOTA

A propriedade **TimeStamp** deste Tag Bloco retorna o **horário do dia** da coleta dos dados.

Leitura do Status do SER PLC

Utilize um Tag PLC para consultar o estado atual do SER PLC, acessando a posição de memória **%Q2016**, conforme a tabela a seguir.

B1	Não utilizado
B2	2 (dois)
B3	2016
B4	Não utilizado

Os valores possíveis para este Tag PLC são **0**: Offline ou **1**: Online.

Leitura do Status do Histórico de Eventos do SER PLC

Utilize um Tag PLC para consultar a presença de eventos no histórico de eventos do SER PLC acessando a posição de memória **%Q2011**, conforme a tabela a seguir.

B1	Não utilizado
B2	2 (dois)
B3	2011
B4	Não utilizado

Os valores possíveis para este Tag PLC são **0**: Sem eventos ou **1**: Com eventos.

Coleta de Eventos Horner SER300/CDC300

Função 300

Utilizado para coletar os eventos de um equipamento Horner SER300/CDC300 através da leitura de um Tag Bloco, conforme os parâmetros descritos na tabela a seguir.

B1	Não utilizado
B2	300
B3	Registro inicial do bloco de eventos gerado por um equipamento CDC300
B4	Não utilizado

O evento é retornado em um Bloco com a seguinte estrutura de Elementos:

- **Elemento 0:** New Event Input Data (entre zero e 15 bits)
- **Elemento 1:** New Event Input Data (entre 16 e 31 bits)
- **Elemento 2:** New Event Input Data (entre 32 e 47 bits)
- **Elemento 3:** New Event Input Data (entre 48 e 63 bits)
- **Elemento 4:** New Event Input Data (entre 64 e 79 bits)
- **Elemento 5:** New Event Input Data (entre 80 e 95 bits)
- **Elemento 6:** New Event Input Data (entre 96 e 111 bits)
- **Elemento 7:** New Event Input Data (entre 112 e 127 bits)
- **Elemento 8:** New Event Input Data (entre 128 e 143 bits)
- **Elemento 9:** New Event Input Data (entre 144 e 159 bits)
- **Elemento 10:** New Event Input Data (entre 160 e 175 bits)

- **Elemento 11:** New Event Input Data (entre 176 e 191 bits)
- **Elemento 12:** New Event Input Data (entre 192 e 207 bits)
- **Elemento 13:** New Event Input Data (entre 208 e 223 bits)
- **Elemento 14:** New Event Input Data (entre 224 e 239 bits)
- **Elemento 15:** New Event Input Data (entre 240 e 255 bits)
- **Elemento 16:** Current Time (formato **Data e Hora**)
- **Elemento 17:** Current Input Data (entre zero e 15 bits)
- **Elemento 18:** Current Input Data (entre 16 e 31 bits)
- **Elemento 19:** Current Input Data (entre 32 e 47 bits)
- **Elemento 20:** Current Input Data (entre 48 e 63 bits)
- **Elemento 21:** Current Input Data (entre 64 e 79 bits)
- **Elemento 22:** Current Input Data (entre 80 e 95 bits)
- **Elemento 23:** Current Input Data (entre 96 e 111 bits)
- **Elemento 24:** Current Input Data (entre 112 e 127 bits)
- **Elemento 25:** Current Input Data (entre 128 e 143 bits)
- **Elemento 26:** Current Input Data (entre 144 e 159 bits)
- **Elemento 27:** Current Input Data (entre 160 e 175 bits)
- **Elemento 28:** Current Input Data (entre 176 e 191 bits)
- **Elemento 29:** Current Input Data (entre 192 e 207 bits)
- **Elemento 30:** Current Input Data (entre 208 ... 223 bits)
- **Elemento 31:** Current Input Data (entre 224 e 239 bits)
- **Elemento 32:** Current Input Data (entre 240 e 255 bits)
- **Elemento 33:** SER300 Module Command Register
- **Elemento 34:** SER300 Module Response Register
- **Elemento 35:** SER300 Module Status Register
- **Elemento 36:** SER300 Module Firmware Version Number
- **Elemento 37:** New Output Data (entre zero e 15 bits)
- **Elemento 38:** New Output Data (entre 16 e 31 bits)
- **Elemento 39:** New Output Data (entre 32 e 47 bits)
- **Elemento 40:** New Output Data (entre 48 e 63 bits)
- **Elemento 41:** Input Mask Word Index
- **Elemento 42:** Input Mask

NOTA

A propriedade **TimeStamp** deste Tag Bloco retorna o **New Event Time Stamp** da coleta dos dados.

Coleta de Eventos PAC System RX3i

Função 400

Utilizado para coletar vários registros do histórico de eventos de um equipamento RX3i via script de evento **OnRead** da leitura de um Tag Bloco, conforme os parâmetros descritos na tabela a seguir.

B1	Não utilizado
B2	400
B3	Não utilizado
B4	Igual a 1 (um) para limpeza automática do histórico de eventos de um equipamento RX3i (opcional)

Cada evento **OnRead** retorna um Bloco com a seguinte estrutura de Elementos, caso haja algum evento disponível:

- **Elemento 0:** Estado de um evento
- **Elemento 1:** Número de um Canal
- **Elemento 2:** Contador de Eventos de um Canal
- **Elemento 3:** Contador de eventos de todos os Canais
- **Elemento 4:** Estado corrente

NOTA

A propriedade **TimeStamp** deste Tag Bloco retorna o **horário do dia** da coleta dos dados.

Leitura do Status do Histórico de Eventos do PAC System RX3i

Utilize um Tag PLC para consultar a presença de eventos no histórico de eventos de um equipamento RX3i, acessando a posição de memória **%W00001**, conforme a tabela a seguir.

B1	Não utilizado
B2	40
B3	1 (um)
B4	Não utilizado

Os valores possíveis para este Tag PLC são **0**: Sem eventos ou **1**: Com eventos.

Limpeza do Histórico de Eventos do PAC System RX3i

Utilize uma escrita em um Tag PLC para liberar todo o conteúdo do histórico de eventos de um equipamento RX3i, escrevendo o valor 0 (zero) na posição de memória **%W00001**, conforme a tabela a seguir.

B1	Não utilizado
B2	40
B3	1 (um)
B4	Não utilizado

O valor de escrita para este Tag PLC é 0 (zero, libera eventos).

NOTA

O parâmetro *B2* igual a 400 pode realizar a limpeza do histórico de eventos automaticamente, se configurado em conjunto com o parâmetro *B4* igual a 1 (um).

Tipos de Dados

Tipos de dados disponíveis

TIPO DE TAG	N2 OU B2	DESCRIÇÃO	OPERAÇÃO
PLC ou Bloco	0 (zero)	Discrete Inputs (%I), Bit	Leitura e Escrita
PLC ou Bloco	1 (um)	Discrete Inputs (%I), Byte	Leitura e Escrita
PLC ou Bloco	2 (dois)	Discrete Outputs (%Q), Bit	Leitura e Escrita
PLC ou Bloco	3 (três)	Discrete Outputs (%Q), Byte	Leitura e Escrita
PLC ou Bloco	4 (quatro)	Discrete Temporaries (%T), Bit	Leitura e Escrita
PLC ou Bloco	5 (cinco)	Discrete Temporaries (%T), Byte	Leitura e Escrita
PLC ou Bloco	6 (seis)	Discrete Internals (%M), Bit	Leitura e Escrita
PLC ou Bloco	7 (sete)	Discrete Internals (%M), Byte	Leitura e Escrita
PLC ou Bloco	8 (oito)	Special Contacts A (%SA), Bit	Somente Leitura
PLC ou Bloco	9 (nove)	Special Contacts A (%SA), Byte	Somente Leitura
PLC ou Bloco	10	Special Contacts B (%SB), Bit	Somente Leitura
PLC ou Bloco	11	Special Contacts B (%SB), Byte	Somente Leitura
PLC ou Bloco	12	Special Contacts C (%SC), Bit	Somente Leitura
PLC ou Bloco	13	Special Contacts C (%SC), Byte	Somente Leitura
PLC ou Bloco	14	System Fault (%S), Bit	Leitura e Escrita

TIPO DE TAG	N2 OU B2	DESCRIÇÃO	OPERAÇÃO
PLC ou Bloco	15	System Fault (%S), Byte	Leitura e Escrita
PLC ou Bloco	16	Genius Global Data (%G), Bit	Leitura e Escrita
PLC ou Bloco	17	Genius Global Data (%G), Byte	Leitura e Escrita
PLC ou Bloco	18	Analog Input table (%AI), Word de 16 bits sem sinal	Leitura e Escrita
PLC ou Bloco	19	Analog Input table (%AI), Word de 16 bits com sinal	Leitura e Escrita
PLC ou Bloco	20	Analog Output table (%AQ), Word de 16 bits sem sinal	Leitura e Escrita
PLC ou Bloco	21	Analog Output table (%AQ), Word de 16 bits com sinal	Leitura e Escrita
PLC ou Bloco	22	Register table (%R), Word de 16 bits sem sinal	Leitura e Escrita
PLC ou Bloco	23	Register table (%R), Word de 16 bits com sinal	Leitura e Escrita
PLC ou Bloco	24	Register table (%R), DWord de 32 bits com sinal	Leitura e Escrita
PLC ou Bloco	25	Register table (%R), DWord de 32 bits sem sinal	Leitura e Escrita
PLC ou Bloco	26	Register table (%R), Float de 32 bits IEEE-754 Floating point	Leitura e Escrita
PLC ou Bloco	27	Register table (%R), String	Leitura e Escrita
PLC ou Bloco	28	Register table (%R), String reverso	Leitura e Escrita
PLC ou Bloco	29	Register table (%R), Double de 64 bits IEEE-754 Floating point	Leitura e Escrita
PLC ou Bloco	30	Local Memory (%L), Word de 16 bits sem sinal	Leitura e Escrita
PLC ou Bloco	31	Local Memory (%L), Word de 16 bits com sinal	Leitura e Escrita
PLC ou Bloco	32	Local Memory (%L), DWord de 32 bits com sinal	Leitura e Escrita
PLC ou Bloco	33	Local Memory (%L), DWord de 32 bits sem sinal	Leitura e Escrita
PLC ou Bloco	34	Local Memory (%L), Float de 32 bits IEEE-754 Floating point	Leitura e Escrita
PLC ou Bloco	35	Local Memory (%L), Double de 64 bits IEEE-754 Floating point	Leitura e Escrita
PLC ou Bloco	40	Word Memory (%W), Word de 16 bits sem sinal	Leitura e Escrita

TIPO DE TAG	N2 OU B2	DESCRIÇÃO	OPERAÇÃO
PLC ou Bloco	41	Word Memory (%W), Word de 16 bits com sinal	Leitura e Escrita
PLC ou Bloco	42	Word Memory (%W), DWord de 32 bits com sinal	Leitura e Escrita
PLC ou Bloco	43	Word Memory (%W), DWord de 32 bits sem sinal	Leitura e Escrita
PLC ou Bloco	44	Word Memory (%W), Float de 32 bits IEEE-754 Floating point	Leitura e Escrita
IOTag/IOBlock	45	Word Memory (%W), Double de 64 bits IEEE-754 Floating point	Leitura e Escrita
PLC ou Bloco	50	UDT Word sem sinal	Leitura e Escrita
PLC ou Bloco	51	UDT Short Int (com sinal)	Leitura e Escrita
PLC ou Bloco	52	UDT I32 (com sinal)	Leitura e Escrita
PLC ou Bloco	53	UDT U32 (sem sinal)	Leitura e Escrita
PLC ou Bloco	54	UDT R32 Float	Leitura e Escrita
PLC ou Bloco	55	UDT R64 Double	Leitura e Escrita
PLC ou Bloco	56	UDT Boolean (byte)	Leitura e Escrita
PLC ou Bloco	57	UDT String	Leitura e Escrita
PLC	100	Relógio do CLP	Leitura e Escrita
PLC	500	Read UDT	Leitura

Quantidade Máxima de Elementos por Tag Bloco

Quantidade máxima de Elementos por Tag Bloco

TIPO DE DADOS	LIMITE
Bit	Número de Elementos menor ou igual a 200
Byte	Número de Elementos menor ou igual a 200
Word	Número de Elementos menor ou igual a 100
DWord ou Float	Número de Elementos menor ou igual a 50
Double	Número de Elementos menor ou igual a 25
Texto	Número de Elementos multiplicado pelo valor do parâmetro <i>B4</i> é menor ou igual a 200

User Data Types

Este Driver permite a leitura de Tags com endereçamento por **Strings**, relativos a objetos ou tipos definidos pelo usuário, conhecidos como **UDT** (*User Data Types*).

Um Tag do tipo **UDT** é endereçado através do parâmetro *Item*, informando o caminho de um objeto, e também através do parâmetro *N2*, em que é definido o tipo de dados. Para mais informações, consulte o tópico **Tipos de Dados**.

Para que os Tags UDT sejam lidos, é necessário criar um Tag PLC com o parâmetro *N2* igual a 500 e com a propriedade **AdviseType** igual a **AlwaysInAdvise**. A cada *scan* deste Tag, todos os Tags PLC e Tags Bloco do tipo **UDT** são atualizados e lidos em grupos de 40 no máximo a cada solicitação.

Caso seja informado no parâmetro *Item* um objeto que represente um vetor, ou *array*, de tipos básicos, é possível criar um Tag Bloco e cada Elemento de Bloco corresponder a um elemento deste vetor.

Exemplos de Endereçamento

- Para ler o bit **%SB80** em modo **Bit**:
 - **N2**: 10 (igual a **%SB** em modo **Bit**)
 - **N3**: 80 (lê o bit 80)
 - **N4**: 0 (zero)
- Para ler o registrador **%R594** em modo **Word**:
 - **N2**: 20 (igual a **%R** sempre em modo **Word**)
 - **N3**: 594 (número do registrador a ser lido)
 - **N4**: 0 (zero)
- Para ler 100 bytes em modo **Texto**, partindo do endereço **%R594**:
 - **N2**: 27 (igual a **%R** em modo **Texto**)
 - **N3**: 594 (começa em **%R594**)
 - **N4**: 100 (número de caracteres a serem lidos)

No nível do protocolo, este Tag realiza uma leitura de 50 registros (**Words**), partindo de **%R594** até **%R644**, para receber todos os 100 bytes que compõem o texto (caracteres em formato **ASCII**).

- Para ler as variáveis **%G10** a **%G50**, inclusive, em modo **Byte** do CLP com endereço IP igual a 4.0.0.4 utilizando o arquivo *gef_cfg.ini* como exemplo:
 - **B2**: 17 (igual a **%G** em modo **Byte**)
 - **B3**: 10 (começa em **%G10**)
 - **B4**: 0 (zero)
- Para ler 60 bits do CLP com endereço IP igual a 3.0.0.1 em modo **Bit**, começando do endereço **%SA5**:
 - **B2**: 8 (igual a **%SA** em modo **Bit**)
 - **B3**: 5 (começa em **%SA5**)
 - **B4**: 0 (zero)

Documentação das Interfaces de Comunicação

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

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

Aba Ethernet

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

Ethernet

Transport: TCP/IP ▾

PING before connecting
 Timeout: 4000 ms
 Retries: 1

Listen for connections on port: 0

Share listen port with other processes

Interface: (All Interfaces) ▾

Use IPv6 Use SSL SSL Settings

Enable 'ECHO' suppression

IP Filter:

Connect to

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

Aba Ethernet

Opções disponíveis na aba Ethernet

OPÇÃO	DESCRIÇÃO
Transport	Selecione o valor TCP/IP para um <i>socket</i> TCP (<i>stream</i>) ou selecione o valor UDP/IP para utilizar um <i>socket</i> UDP (<i>connectionless datagram</i>)
Listen for connections on port	Utilize esta opção para aguardar por novas conexões em uma porta IP específica, comum em Drivers Escravos. Caso esta opção permaneça desmarcada, um Driver se conecta ao endereço e porta especificados no grupo Connect to
Share listen port with other processes	Selecione esta opção para compartilhar a porta de escuta com outros Drivers e processos
Interface	Selecione a interface de rede local, identificada pelo endereço IP, que um Driver utiliza para efetuar e receber conexões, ou selecione o valor (All Interfaces) para permitir conexões em qualquer interface de rede
Use IPv6	Selecione esta opção para forçar um Driver a utilizar endereços no formato IPv6 em todas as conexões Ethernet. Deixe esta opção desmarcada para utilizar o formato IPv4
Enable 'ECHO' suppression	Habilite esta opção para eliminar o <i>eco</i> dos dados recebidos. O <i>eco</i> é uma cópia dos dados enviados, que pode ser retornada antes da mensagem de resposta
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
	<p>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:</p> <ul style="list-style-type: none"> • Timeout: Especifique o número de milissegundos de espera por uma resposta de um comando ping. Deve-se usar um comando ping para verificar o tempo normal de resposta, configurando esta opção para um valor acima desta média. Normalmente pode-se configurar um valor entre 1000 e 4000 milissegundos, ou seja entre 1 (um) e 4 (quatro) segundos • Retries: Número de retentativas de um comando ping, sem contar a tentativa inicial. Se todas as tentativas falharem, então a conexão com o <i>socket</i> é abortada

Opções disponíveis no grupo Connect to

OPÇÃO	DESCRIÇÃO
Main IP	<p>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"</p>
Port	<p>Digite a porta IP de um dispositivo remoto, entre 0 (zero) e 65535</p>
Local port	<p>Selecione esta opção para utilizar uma porta IP local fixa ao conectar a um dispositivo remoto</p>
Backup IP 1, 2 e 3	<p>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</p>

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]

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

IO.Ethernet.BackupLocalPort[2,3]

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

IO.Ethernet.BackupLocalPortEnable[2,3]

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

IO.Ethernet.BackupPort[2,3]

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

IO.Ethernet.IPFilter

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

- **192.168.0.24:** Aceita apenas conexões do endereço IPv4 192.168.0.24
- **192.168.0.41-50:** Aceita conexões dos endereços IPv4 no intervalo entre 192.168.0.41 e 192.168.0.50
- **192.168.0.*:** Aceita conexões dos endereços IPv4 no intervalo entre 192.168.0.0 e 192.168.0.255

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

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

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

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

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

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

IO.Ethernet.ListenIP

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

IO.Ethernet.ListenPort

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

IO.Ethernet.MainIP

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

IO.Ethernet.MainLocalPort

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

IO.Ethernet.MainLocalPortEnable

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

IO.Ethernet.MainPort

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

IO.Ethernet.PingEnable

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

IO.Ethernet.PingTimeoutMs

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

IO.Ethernet.PingTries

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

IO.Ethernet.ShareListenPort

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

IO.Ethernet.SupressEcho

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

IO.Ethernet.Transport

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
2.0.9	03/09/2025	M. Ludwig	<ul style="list-style-type: none"> Driver atualizado para a biblioteca IOKit versão 3.0 e Visual Studio 2022 (<i>Case 37974</i>).
2.0.8	07/10/2024	M. Salvador	<ul style="list-style-type: none"> Adicionado suporte a UDT (<i>User Data Types</i>).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
2.0.7	12/07/2021	C. Mello	<ul style="list-style-type: none"> • Ajustes de compatibilidade para leitura de superblocos via propriedade EnableReadGrouping do Elipse E3, Elipse Power ou Elipse Water (Case 31113).
2.0.6	04/11/2020	C. Mello	<ul style="list-style-type: none"> • Adicionado suporte para o tipo de dados Double 64 bits IEEE-754 Floating point para os registradores %R, %L e %W (Case 29410). • Alterada a configuração do tipo de dados Word 16 bits sem sinal do registrador %W dos parâmetros N2 ou B2 igual a 29 para N2 ou B2 igual a 40.
2.0.5	01/08/2019	M. Ludwig	<ul style="list-style-type: none"> • Driver portado para o Visual Studio 2017 (Case 27187).
2.0.4	31/10/2017	C. Mello	<ul style="list-style-type: none"> • Corrigidas inconsistências internas relacionadas com a atualização dos Tags via Superblocos (Case 23555).
2.0.3	25/07/2017	M. Ludwig	<ul style="list-style-type: none"> • Corrigido um acesso inválido eventual ao realizar uma leitura agrupada com parâmetro N3 maior que zero (Case 23079).
2.0.2	06/02/2017	C. Mello	<ul style="list-style-type: none"> • Adicionada uma opção para configurar os campos Source ID e Destination ID na caixa de diálogo de propriedades deste Driver (Case 18507).
2.0.1	16/12/2016	C. Mello	<ul style="list-style-type: none"> • Adicionado suporte para leitura com Superblocos mais <i>callbacks</i> para os registradores %L (Case 21895). • Ajustes de sincronização com valores modificados pelo modo de varredura constante de alguns controladores (Case 18018). • Ajustes nas escritas de Tags Bloco para o tipo de dados String (Case 17902). • Ajustes para leitura e escrita com Tags Bloco envolvendo

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			<p>os tipos de dados Bit e Byte (Case 17415).</p> <ul style="list-style-type: none"> • Adicionado um comando para consulta do último PLC Status Word informado por um equipamento (Case 17380). • Adicionadas novas interpretações de valores para os registradores de memória %W (Case 16545). • Adicionadas novas interpretações de valores para a região Program Block Memory (registradores %L) dos equipamentos da série 90-70 (Case 15904). • Adicionado suporte para leitura de Tags por <i>callbacks</i> e serviço de Superblocos transferido para o controle e processamento interno deste Driver (Case 14555). • Driver portado para o novo padrão da biblioteca IOKit versão 2.0 (Case 14582).
1.10.1	19/01/2012	C. Mello	<ul style="list-style-type: none"> • Adicionado suporte para interpretação de códigos de erro previstos pelo protocolo (Case 12677).
1.9.1	27/06/2011	C. Mello	<ul style="list-style-type: none"> • Adicionado suporte para o tipo de dados Local Memory (%L) (parâmetro <i>B2</i> igual a 30) (Case 11886).
1.8.1	17/06/2010	C. Mello	<ul style="list-style-type: none"> • Adicionado suporte para o tipo de dados Word Memory (%W) (parâmetro <i>B2</i> igual a 29) e também suporte para a coleta de eventos PAC System RX3i, Função 400 (Case 10854). • Adicionadas informações sobre a consulta do status da coleta de eventos do SER PLC (Case 11232).
1.7.1	16/03/2009	C. Mello	<ul style="list-style-type: none"> • Ajustes para prevenir quebras no controle de transações das mensagens (Case 10187).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
1.6.1	06/06/2008	C. Mello	<ul style="list-style-type: none"> • Ajustes para prevenir um erro após a perda física da conexão Ethernet (<i>Case 9571</i>).
1.5.1	18/05/2007	F. Englert	<ul style="list-style-type: none"> • Ajustes para inicialização do protocolo de comunicação sempre que há uma perda da conexão física (<i>Case 8274</i>).
1.4.1	24/11/2006	C. Mello	<ul style="list-style-type: none"> • Adicionada a Coleta de Eventos Horner SER300/CDC300 - Função 300 (<i>Case 6776</i>). • Adicionada a Coleta de Histórico de Eventos do PLC SER - Função 200 (<i>Case 6306</i>). • Ajustes para troca do endereço IP ativo (<i>Case 7255</i>). • Ajustes no contador TNS para se tornar sensível ao contexto de reconexão automática (<i>Case 6796</i>).
		F. Englert	<ul style="list-style-type: none"> • Suporte a Superblocos atualizado para o padrão do Elipse E3 versão 2.0 (<i>Case 6792</i>). • Corrigido um erro na leitura e escrita em Elementos de Bloco com o parâmetro <i>B2</i> igual a 24, 25 ou 26 que utilize um <i>offset</i> incorreto (<i>Case 6793</i>).
1.3.1	27/10/2005	M. Salvador	<ul style="list-style-type: none"> • Adicionado suporte a Superblocos, compatível apenas com o Elipse E3 versão 1.23, não oficial. • Implementado o controle de reconexão automática em caso de falha constante na comunicação.
1.2.1	25/04/2005	C. Mello	<ul style="list-style-type: none"> • Revisão geral de todo o código fonte.
		M. Salvador	<ul style="list-style-type: none"> • Desabilitadas as escritas nos tipos de dados %SA, %SB e %SC, pois não funcionam em alguns modelos de CLP.

VERSÃO	DATA	AUTOR	COMENTÁRIOS
1.1.1	24/04/2004	L. Silva	<ul style="list-style-type: none">• Adicionada uma nova opção para a função 200 deste Driver para controlar estampas de tempo e milissegundos (<i>Case 3524</i>).
1.0.1	28/01/2005	M. Salvador	<ul style="list-style-type: none">• Versão inicial deste Driver.

Matriz

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

90430-000 — Porto Alegre — RS

Fone: (+55 51) 3346-4699

Fax: (+55 51) 3222-6226

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

Filial no Paraná

Av. Sete de Setembro, 4698/1708

80240-000 — Curitiba — PR

Fone: (+55 41) 4062-5824

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

Filial no Rio de Janeiro

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

22250-044 — Rio de Janeiro — RJ

Fone: (+55 21) 2430-5912

Suporte Técnico: (+55 21) 2430-5963

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

Filial em São Paulo

Rua dos Pinheiros, 870/Cj. 141 e 142

05422-001 — São Paulo — SP

Fone: (+55 11) 3061-2828

Fax: (+55 11) 3086-2338

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

Filial em Minas Gerais

Rua Antônio de Albuquerque, 156/705

30112-010 — Belo Horizonte — MG

Fone: (+55 31) 4062-5824

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

Filial em Taiwan

9F., No.12, Beiping 2nd St., Sanmin Dist.

807 — Kaohsiung City — Taiwan

Fone: (+886 7) 323-8468

Fax: (+886 7) 323-9656

E-mail: evan@elipse.com.br

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

www.elipse.com.br

kb.elipse.com.br

forum.elipse.com.br

www.youtube.com/elipsesoftware

elipse@elipse.com.br



Gartner, Cool Vendors in Brazil 2014, April 2014.

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

Microsoft Partner

Gold Independent Software Vendor (ISV)