Driver Ictel CT850

Nome do Arquivo	CT850.dll
Fabricante	Ictel Instrumentação Controles e Telemetria LTDA.
Equipamentos	Central de Telemetria
Protocolo	Modbus
Versão	1.0.5
Última Atualização	22/05/2026
Plataforma	Win32
Dependências	IOKit versão 2.0 ou superior
Leitura com Superblocos	Sim
Nível	0

Introdução

Este Driver implementa o protocolo Modbus, permitindo a uma aplicação desenvolvida pela **Elipse Software** comunicar com equipamentos do tipo Central de Telemetria da Ictel Instrumentação Controles e Telemetria LTDA.

Preparação de um Equipamento

Este Driver tem como objetivo efetuar todos os procedimentos para executar uma rotina de telecomando com as Centrais de Telemetria Ictel **CT850** e Centrais de Telemetria **AUTTOM** (**Radio 1** e **Radio 2**), de acordo com o esquema proprietário de cada Central.

Central CT850

As Estações Remotas da Central **CT850** podem receber telecomandos via aplicação externa por meio das seguintes áreas de memória:

- **0x9800**: Endereço fixo de memória Modbus para efetuar requisições de telecomandos
- **0xA000**: Endereço fixo de memória Modbus para obter a resposta de sucesso ou falha do telecomando requisitado pelo endereço **0x9800**

A Central **CT850** só pode executar um telecomando por vez.

Central AUTTOM (Radio1 e Radio 2)

As Estações Remotas da Central **AUTTOM** podem receber telecomandos via aplicação externa por meio das seguintes áreas de memória **Radio 1** (para as áreas de memória **Radio 2**, incremente as áreas de memória **Radio 1** em 100, com exceção da memória **303**, que permanece fixa):

- **301**: Endereço fixo de memória Modbus para indicar o endereço de uma Estação Remota
- **302**: Endereço fixo de memória Modbus para indicar o tipo de telecomando
- **303**: Endereço fixo de memória Modbus para indicar o número do controle ou o tipo de Estação Remota, dependendo do telecomando utilizado
- **300**: Endereço fixo de memória Modbus para disparo de um comando de *trigger*
- **320**: Endereço fixo de memória Modbus para obter a resposta de sucesso ou falha
- **321**: Endereço fixo de memória Modbus para obter o código de retorno (48 para sucesso ou outros códigos para erros)
- **304-311 e 324-331**: Regiões fixas de memória Modbus para armazenamento dos Setpoints de controle
- **312-319 e 332-339**: Regiões fixas de memória Modbus para armazenamento dos Setpoints de bloqueio
- **340**: Endereço fixo de memória Modbus para indicar o número de um periférico
- **341**: Endereço fixo de memória Modbus para indicar a memória inicial interna de um periférico
- **342**: Endereço fixo de memória Modbus para indicar a quantidade de memórias
- **343**: Endereço fixo de memória Modbus para atribuir um valor de referência a um periférico

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 usando a opção **Extra Settings** no **Elipse SCADA**.

Nesta caixa de diálogo de propriedades, a aba **CT850** 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 **CT850**, 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 definidos 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 Ictel CT850

ABA	PARÂMETRO	STRING OFFLINE	TIPO DE DADOS	DESCRIÇÃO
Setup	Physical Layer	IO.Type	Texto	Configura a interface física. As opções disponíveis são **Serial**, **Ethernet**, **Modem** ou **RAS**
	Timeout	IO.TimeoutMs	Número	Limite de tempo, em milissegundos, para recepção dos dados na resposta de um equipamento. Por exemplo, o valor 1000 define um limite de 1 (um) segundo
Ethernet	Transport	IO.Ethernet.Transport	Texto	Configura a camada de transporte. As opções disponíveis são **TCP/IP** ou **UDP/IP**
	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	Porta TCP/IP de um equipamento. Os valores possíveis variam entre 0 (zero) e 65535

ABA	PARÂMETRO	STRING OFFLINE	TIPO DE DADOS	DESCRIÇÃO
CT850	Telecommand Time-out (s)	CT850.TeleCommandTimeout	Número	Limite de tempo, em segundos, para a Central de Telemetria responder um código de retorno referente ao processamento do telecomando que foi requisitado

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

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

Telecomandos

Esta seção contém Tags para manipulação de telecomandos.

Comandos

Utilize Tags PLC para manipular telecomandos específicos das Centrais de Telemetria, usando os parâmetros indicados na tabela a seguir.

Parâmetros para Centrais de Telemetria

Device	Parâmetros de um equipamento, no formato **ID:XXXXX:Rn:Tn**
Item	Parâmetros de um telecomando. Para mais informações, consulte a tabela **Tags de telecomandos**
N1 ou B1	Não usado, deixe em 0 (zero)
N2 ou B2	Não usado, deixe em 0 (zero)
N3 ou B3	Não usado, deixe em 0 (zero)
N4 ou B4	Não usado, deixe em 0 (zero)

No parâmetro **Device** destes Tags, informe os dados de identificação das Centrais de Telemetria utilizando o formato **ID:XXXXX:Rn:Tn**, conforme descrito a seguir:

- **ID**: Identificador de um escravo Modbus
- **XXXXX**: Tipo de Central, conforme os valores a seguir:
  - **CT850**: Central Ictel CT850
  - **AUTTOM_R1**: Central Auttom Radio 1
  - **AUTTOM_R2**: Central Auttom Radio 2
- **Rn**: Número de uma Estação Remota
- **Tn**: Tipo de Estação Remota

A tabela a seguir contém os telecomandos que podem ser utilizados para interagir com Centrais de Telemetria, por meio do parâmetro **Item**.

Tags de telecomandos

ITEM	DESCRIÇÃO	OPERAÇÃO
STATUS	Retorna informações sobre o estado atual deste Driver. Os valores possíveis são **0**: Livre (número) ou **1**: Processando telecomando (número)	Somente leitura

ITEM	DESCRIÇÃO	OPERAÇÃO
Vn:COMMAND	Envia um telecomando para o número do controlador indicado em **Vn**. Os valores de escrita deste Tag usados para selecionar o tipo de controle a ser enviado são **67**: Executa o acionamento de controle indicado em **Vn**, **68**: Executa o desligamento de controle indicado em **Vn**, **69**: Seleciona um controle para manual indicado em **Vn** ou **70**: Seleciona um controle para automático indicado em **Vn**. Utilize um Tag **Vn:RESPONSE** para verificar se o processamento do telecomando foi concluído com sucesso ou com falha	Somente escrita
Vn:RESPONSE	Tag de leitura usada para monitorar o resultado do último telecomando enviado para o número do controlador indicado em **Vn**. Em caso de sucesso, este Tag retorna um inteiro de 32 bits, com a parte alta zerada e a parte baixa contendo o valor que foi escrito pelo Tag **Vn:COMMAND**. Em caso de falha, este Tag retorna um inteiro de 32 bits, com a parte alta contendo o erro da Central e a parte baixa contendo o valor que foi escrito pelo Tag **Vn:COMMAND**	Somente leitura

Exemplos de Uso

Escrita de um telecomando para a Central **CT850** com identificador igual a 1 (um, CT850), Estação Remota igual a 3 (três, R3), Tipo de Estação Remota igual a 2 (dois, T2) e para acionamento do controle 5 (cinco, V5):

- **Tag.ParamDevice**: 1:CT850:R3:T2
- **Tag.ParamItem**: V5:COMMAND
- **Valor de escrita deste Tag**: 67 (acionamento de controle)

Retorno deste telecomando:

- **Tag.ParamDevice**: 1:CT850:R3:T2
- **Tag.ParamItem**: V5:RESPONSE
- **Valor de leitura deste Tag em caso de sucesso**: 67, ou seja, um inteiro de 32 bits 0x00000043, com a parte alta igual a 0 (zero) e a parte baixa igual a 67
- **Valor de leitura deste Tag em caso de falha**: 3407939, ou seja, um inteiro de 32 bits 0x00340043, com a parte alta igual a 52 e a parte baixa igual a 67, em que a parte alta recebe o código de erro 0x0034 da Central

Setpoints

Utilize Tags PLC para manipular telecomandos específicos das Centrais de Telemetria, usando os parâmetros indicados na tabela a seguir.

Parâmetros para Centrais de Telemetria

Device	Parâmetros de um equipamento, no formato **ID:XXXXX:Rn:Tn**
Item	Parâmetros de um telecomando. Para mais informações, consulte as tabelas **Tags de telecomandos para Setpoints nas Centrais CT850** e **Tags de telecomandos para Setpoints nas Centrais AUTTOM**
N1 ou B1	Não usado, deixe em 0 (zero)
N2 ou B2	Não usado, deixe em 0 (zero)
N3 ou B3	Não usado, deixe em 0 (zero)
N4 ou B4	Não usado, deixe em 0 (zero)

No parâmetro **Device** destes Tags, informe os dados de identificação das Centrais de Telemetria utilizando o formato **ID:XXXXX:Rn:Tn**, conforme descrito a seguir:

- **ID**: Identificador de um escravo Modbus
- **XXXXX**: Tipo de Central, conforme os valores a seguir:
  - **CT850**: Central Ictel CT850
  - **AUTTOM_R1**: Central Auttom Radio 1
  - **AUTTOM_R2**: Central Auttom Radio 2
- **Rn**: Número de uma Estação Remota
- **Tn**: Tipo de Estação Remota

As tabelas a seguir contêm os telecomandos que podem ser utilizados para interagir com Centrais de Telemetria, por meio do parâmetro **Item**.

Tags de telecomandos para Setpoints nas Centrais CT850

TELECOMANDO	ITEM	DESCRIÇÃO	OPERAÇÃO
Requisição de escrita de um Setpoint de comando	Vn:SP_COMMAND	Envia um telecomando de Setpoint para a Estação Remota indicada por **Vn**, no formato **Vn = SetPointCount * 1000 + SetPointIndex**. Por exemplo, se a Central possui 5 (cinco) Setpoints e é preciso modificar o terceiro Setpoint, este telecomando é igual a **Vn = 5 * 1000 + 3 -> Vn = 5003**. Para as requisições **Vn:SP_COMMAND** e **Vn:SP_BLOQ**, o valor de escrita deste Tag representa o novo valor de Setpoint a ser enviado para a Central. Para as requisições de consulta dos Setpoints **Vn:SP_QUERY_COMMAND** e **Vn:SP_QUERY_BLOQ**, a escrita deste Tag pode ter qualquer valor. Utilize o respectivo Tag **Vn:SP_RESPONSE** ou **Vn:SP_RESP_BLOQ** para verificar se o processamento deste telecomando foi concluído com sucesso ou com falha	Somente escrita
Requisição de consulta de um Setpoint de comando	Vn:SP_QUERY_COMMAND		
Requisição de escrita de um Setpoint de bloqueio	Vn:SP_BLOQ		
Requisição de consulta de um Setpoint de bloqueio	Vn:SP_QUERY_BLOQ		

TELECOMANDO	ITEM	DESCRIÇÃO	OPERAÇÃO
Leitura de resposta da requisição de escrita ou consulta de um Setpoint de comando	Vn:SP_RESPONSE	Tag de leitura usado para monitorar o resultado do último telecomando de Setpoint enviado para a Estação Remota indicada por **Vn** no formato **Vn = SetPointCount * 1000 + SetPointIndex**. Por exemplo, se a Central possui 5 (cinco) Setpoints e é preciso modificar o terceiro Setpoint, este telecomando é igual a **Vn = 5 * 1000 + 3 -> Vn = 5003**. Em caso de sucesso, este Tag retorna um inteiro de 32 bits, com a parte alta zerada e a parte baixa contendo o valor escrito pelo Tag **Vn:SP_COMMAND**. Em caso de falha, este Tag retorna um inteiro de 32 bits, com a parte alta contendo o erro da Central e a parte baixa contendo o valor escrito pelo Tag **Vn:SP_COMMAND**	Somente leitura
Leitura de resposta da requisição de escrita ou consulta de um Setpoint de bloqueio	Vn:SP_RESP_BLOQ		
Requisição de escrita de um Setpoint Analógico	Vn:SP_ANALOG	Envia um telecomando de Setpoint Analógico para a Estação Remota indicada por **Vn**, no formato **Vn = SetPointCount * 1000 + 2**. Este Setpoint Analógico tem uma posição fixa em "SetPoint 2". Por exemplo, se a Central possui 5 (cinco) Setpoints, este telecomando é igual a **Vn = 5 * 1000 + 2 -> Vn = 5002**. Utilize o Tag **Vn:SP_RESP_ANALOG** para verificar se o processamento deste telecomando foi concluído com sucesso ou com falha	Somente escrita

TELECOMANDO	ITEM	DESCRIÇÃO	OPERAÇÃO
Leitura de resposta da requisição de escrita de um Setpoint Analógico	Vn:SP_RESP_ANALOG	Tag de leitura usado para monitorar o resultado do último telecomando de Setpoint Analógico enviado para a Estação Remota indicada por **Vn** no formato **Vn = SetPointCount * 1000 + 2**. Este Setpoint Analógico tem uma posição fixa em "SetPoint 2". Por exemplo, se a Central possui 5 (cinco) Setpoints, este telecomando é igual a **Vn = 5 * 1000 + 2 -> Vn = 5002**. Em caso de sucesso, este Tag retorna um inteiro de 32 bits, com a parte alta zerada e a parte baixa contendo o valor escrito pelo Tag **Vn:SP_ANALOG**. Em caso de falha, este Tag retorna um inteiro de 32 bits, com a parte alta contendo o erro da Central e a parte baixa contendo o valor escrito pelo Tag **Vn:SP_ANALOG**	Somente leitura

Exemplo de Uso

Escrita de um telecomando de Setpoint para a Central **CT850** com identificador igual a 1 (um, CT850), Estação Remota igual a 3 (três, R3), Tipo de Estação Remota igual a 2 (dois, T2), com um grupo de 5 (cinco) Setpoints, para modificação do Setpoint 3 (três, V5003):

- **Tag.ParamDevice**: 1:CT850:R3:T2
- **Tag.ParamItem**: V5003:SP_COMMAND
- **Valor de escrita deste Tag**: 1234 (valor do Setpoint)

Retorno deste telecomando:

- **Tag.ParamDevice**: 1:CT850:R3:T2
- **Tag.ParamItem**: V5003:SP_RESPONSE
- **Valor de leitura deste Tag em caso de sucesso**: 1234, ou seja, um inteiro de 32 bits 0x000004D2, com a parte alta igual a 0 (zero) e a parte baixa igual a 1234
- **Valor de leitura deste Tag em caso de falha**: 4850898, ou seja, um inteiro de 32 bits 0x004A04D2, com a parte alta igual a 74 e a parte baixa igual a 1234, em que a parte alta recebe o código de erro 0x004A da Central

Tags de telecomandos para Setpoints nas Centrais AUTTOM

TELECOMANDO	ITEM	DESCRIÇÃO	OPERAÇÃO
Requisição de escrita de um Setpoint de comando	Vn:SP_COMMAND:Mn	Envia um telecomando de Setpoint para a Estação Remota indicada por **Vn**, no formato **Vn = SetPointCount * 1000 + SetPointIndex**. Por exemplo, se a Central possui 5 (cinco) Setpoints e é preciso modificar o terceiro Setpoint, o telecomando é **Vn = 5 * 1000 + 3 -> Vn = 5003**. Indique em **Mn** a posição inicial da memória Modbus vinculada ao grupo de Setpoints deste telecomando. Para as requisições **Vn:SP_COMMAND:Mn** e **Vn:SP_BLOQ:Mn**, o valor de escrita deste Tag representa o novo valor de Setpoint a ser enviado para a Central. Para as requisições de consulta dos Setpoints **Vn:SP_QUERY_COMMAND:Mn** e **Vn:SP_QUERY_BLOQ:Mn**, a escrita deste Tag pode ter qualquer valor. Utilize o respectivo Tag **Vn:SP_RESPONSE:Mn** ou **Vn:SP_RESP_BLOQ:Mn** para verificar se o processamento deste telecomando foi concluído com sucesso ou com falha	Somente escrita
Requisição de consulta de um Setpoint de comando	Vn:SP_QUERY_COMMAND:Mn		
Requisição de escrita de um Setpoint de bloqueio	Vn:SP_BLOQ:Mn		
Requisição de consulta de um Setpoint de bloqueio	Vn:SP_QUERY_BLOQ:Mn		

TELECOMANDO	ITEM	DESCRIÇÃO	OPERAÇÃO
Leitura de resposta da requisição de escrita ou consulta de um Setpoint de controle	Vn:SP_RESPONSE:Mn	Tag de leitura usado para monitorar o resultado do último telecomando de Setpoint enviado para a Estação Remota indicada por **Vn**, no formato **Vn = SetPointCount * 1000 + SetPointIndex**. Por exemplo, se a Central possui 5 (cinco) Setpoints e é preciso modificar o terceiro Setpoint, o telecomando é **Vn = 5 * 1000 + 3 -> Vn = 5003**. Indique em **Mn** a posição inicial da memória Modbus vinculada ao grupo de Setpoints deste telecomando. Em caso de sucesso, este Tag retorna um inteiro de 32 bits, com a parte alta zerada e a parte baixa contendo o valor escrito pelo Tag **Vn:SP_COMMAND:Mn**. Em caso de falha, este Tag retorna um inteiro de 32 bits, com a parte alta contendo o erro da Central e a parte baixa contendo o valor escrito pelo Tag **Vn:SP_COMMAND:Mn**	Somente leitura
Leitura de resposta da requisição de escrita ou consulta de um Setpoint de bloqueio	Vn:SP_RESP_BLOQ:Mn		

Exemplo de Uso

Escrita de um telecomando de Setpoint para a Central **AUTTOM** com identificador igual a 1 do Radio 1 (um, AUTTOM_R1), Estação Remota igual a 2 (dois, R2), Tipo de Estação Remota igual a 2 (dois, T2), com um grupo de 4 (quatro) Setpoints, para modificação do Setpoint 2 (dois, V4002), vinculado ao endereço Modbus 40035 (M40035):

- **Tag.ParamDevice**: 1:AUTTOM_R1:R2:T2
- **Tag.ParamItem**: V4002:SP_COMMAND:M40035
- **Valor de escrita deste Tag**: 1234 (valor do Setpoint)

Retorno deste telecomando:

- **Tag.ParamDevice**: 1:AUTTOM_R1:R2:T2
- **Tag.ParamItem**: V4002:SP_RESPONSE:M40035
- **Valor de leitura deste Tag em caso de sucesso**: 1234, ou seja, um inteiro de 32 bits 0x000004D2, com a parte alta igual a 0 (zero) e a parte baixa igual a 1234
- **Valor de leitura deste Tag em caso de falha**: 4850898, ou seja, um inteiro de 32 bits 0x004A04D2, com a parte alta igual a 74 e a parte baixa igual a 1234, em que a parte alta recebe o código de erro 0x004A da Central

No caso da Central **AUTTOM**, o valor de escrita de um Setpoint sempre é armazenado no endereço de memória Modbus vinculado a este Setpoint. No exemplo anterior, em caso de sucesso, o valor 1234 é armazenado na posição de memória Modbus 40035 (M40035).

Os exemplos a seguir mostram como uma Central **AUTTOM** se relaciona com registradores Modbus de retorno de dados.

- **AUTTOM**: Setpoint de comando com retorno Modbus invertido
  - Write Setpoint 1 (**V4001**:SP_COMMAND:M40001) | AUTTON Ctrl01_SPH **#304** ===> Read Setpoint 1 (**V4001**:SP_RESPONSE:M40001) | Modbus **#40002**
  - Write Setpoint 2 (**V4002**:SP_COMMAND:M40001) | AUTTON Ctrl01_SPL **#305** ===> Read Setpoint 2 (**V4002**:SP_RESPONSE:M40001) | Modbus **#40001**
  - Write Setpoint 3 (**V4003**:SP_COMMAND:M40001) | AUTTON Ctrl02_SPH **#306** ===> Read Setpoint 3 (**V4003**:SP_RESPONSE:M40001) | Modbus **#40004**
  - Write Setpoint 4 (**V4004**:SP_COMMAND:M40001) | AUTTON Ctrl02_SPL **#307** ===> Read Setpoint 4 (**V4004**:P_RESPONSE:M40001) | Modbus **#40003**
- **AUTTOM**: Setpoint de bloqueio com retorno Modbus direto
  - Write Setpoint 1 (**V4001**:SP_BLOQ:M40001) | AUTTON Ctrl01_Blq **#312** ===> Read Setpoint 1 (**V4001**:SP_RESP_BLOQ:M40001) | Modbus **#40001**
  - Write Setpoint 2 (**V4002**:SP_BLOQ:M40001) | AUTTON Ctrl01_Lib **#313** ===> Read Setpoint 2 (**V4002**:SP_RESP_BLOQ:M40001) | Modbus **#40002**
  - Write Setpoint 3 (**V4003**:SP_BLOQ:M40001) | AUTTON Ctrl02_Blq **#314** ===> Read Setpoint 3 (**V4003**:SP_RESP_BLOQ:M40001) | Modbus **#40003**
  - Write Setpoint 4 (**V4004**:SP_BLOQ:M40001) | AUTTON Ctrl02_Lib **#315** ===> Read Setpoint 4 (**V4004**:SP_RESP_BLOQ:M40001) | Modbus **#40004**

Periféricos

Utilize Tags PLC para manipular telecomandos específicos das Centrais de Telemetria, usando os parâmetros indicados na tabela a seguir.

Parâmetros para Centrais de Telemetria

Device	Parâmetros de um equipamento, no formato **ID:XXXXX:Rn:Tn**
Item	Parâmetros de um telecomando. Para mais informações, consulte a tabela **Tags de telecomandos para periféricos**
N1 ou B1	Não usado, deixe em 0 (zero)
N2 ou B2	Não usado, deixe em 0 (zero)
N3 ou B3	Não usado, deixe em 0 (zero)
N4 ou B4	Não usado, deixe em 0 (zero)

No parâmetro **Device** destes Tags, informe os dados de identificação das Centrais de Telemetria utilizando o formato **ID:XXXXX:Rn:Tn**, conforme descrito a seguir:

- **ID**: Identificador de um escravo Modbus
- **XXXXX**: Tipo de Central, conforme os valores a seguir:
  - **CT850**: Central Ictel CT850
  - **AUTTOM_R1**: Central Auttom Radio 1
  - **AUTTOM_R2**: Central Auttom Radio 2
- **Rn**: Número de uma Estação Remota
- **Tn**: Tipo de Estação Remota

A tabela a seguir contém os telecomandos que podem ser utilizados para interagir com Centrais de Telemetria, por meio do parâmetro **Item**.

Tags de telecomandos para periféricos

TELECOMANDO	ITEM	DESCRIÇÃO	OPERAÇÃO
Requisição de escrita em periféricos	Vn:PERIPHERAL:Mn[.DataType]	Envia um telecomando de dados para um periférico vinculado à Central, em que **Vn** corresponde ao índice (identificador) deste periférico, **Mn** corresponde à posição de memória deste periférico e **DataType** corresponde ao tipo de dados **.u32** ou **.i32** para enviar inteiros de 32 bits. Este parâmetro é opcional e, se não é especificado, são enviados valores inteiros de 16 bits. Para as requisições **Vn:PERIPHERAL:Mn[.DataType]**, o valor de escrita deste Tag representa o novo valor a ser enviado para um periférico. Para as requisições de consulta **Vn:QUERY_PERIPHERAL:Mn[.DataType]**, a escrita deste Tag pode ter qualquer valor. Utilize o respectivo Tag **Vn:RESP_PERIPHERAL:Mn[.DataType]** para verificar se o processamento deste telecomando foi concluído com sucesso ou com falha	Somente escrita
Requisição de consulta em periféricos	Vn:QUERY_PERIPHERAL:Mn[.DataType]		
Leitura de resposta da requisição de escrita ou consulta em periféricos	Vn:RESP_PERIPHERAL:Mn[.DataType]	Tag de leitura usado para monitorar o resultado do último telecomando de dados enviado para o respectivo periférico indicado por **Vn** na posição de memória **Mn** e, por fim, o tipo de dados **[.DataType]**, se utilizado. Em caso de sucesso, este Tag retorna um inteiro de 32 bits, com a parte alta zerada e a parte baixa contendo o valor escrito pelo Tag **Vn:PERIPHERAL:Mn[.DataType]**. Em caso de falha, este Tag retorna um inteiro de 32 bits, com a parte alta contendo o erro da Central e a parte baixa contendo o valor escrito pelo Tag **Vn:PERIPHERAL:Mn[.DataType]**	Somente leitura

Exemplo de Uso

Escrita de um telecomando para a Central **CT850** com identificador igual a 1 (um, CT850), Estação Remota igual a 3 (três, R3), Tipo de Estação Remota igual a 2 (dois, T2), para envio do valor 1234 ao periférico 1 (um, V1), na posição de memória 6 (seis, M6):

- **Tag.ParamDevice**: 1:CT850:R3:T2
- **Tag.ParamItem**: V1:PERIPHERAL:M6
- **Valor de escrita deste Tag**: 1234 (valor enviado para a memória seis do periférico um)

Retorno deste telecomando:

- **Tag.ParamDevice**: 1:CT850:R3:T2
- **Tag.ParamItem**: V1:RESP_PERIPHERAL:M6
- **Valor de leitura deste Tag em caso de sucesso**: 1234, ou seja, um inteiro de 32 bits 0x000004D2, com a parte alta igual a 0 (zero) e a parte baixa igual a 1234
- **Valor de leitura deste Tag em caso de falha**: 7603410, ou seja, inteiro de 32 bits 0x007404D2, com a parte alta igual a 116 e a parte baixa igual a 1234, em que a parte alta recebe o código de erro 0x0074 da Central

Modbus

Utilize Tags PLC ou Tags Bloco para manipular registradores Modbus, usando os parâmetros indicados na tabela a seguir.

Parâmetros para Tags Modbus

Device	Identificador de um escravo Modbus
Item	Parâmetros Modbus, no formato **XXmmmmm[.tt][.ss]**
N1 ou B1	Não usado, deixe em 0 (zero)
N2 ou B2	Não usado, deixe em 0 (zero)
N3 ou B3	Não usado, deixe em 0 (zero)
N4 ou B4	Não usado, deixe em 0 (zero)

No parâmetro **Item** destes Tags, informe os parâmetros deste Tag Modbus utilizando o formato **XXmmmmm[.tt][.ss]**, conforme descrito a seguir:

- O parâmetro **XX** representa o tipo de comando Modbus (*FCode*):
  - **cl**: Coils | Bits | Read 01 / Write 15
  - **di**: Discrete Input (read only) | Bits | Read 02
  - **hr**: Holding Register | Words | Read 03 / Write 16
  - **ir**: Input Register (read only) | Words | Read 04
  - **scl**: Single Coil | Bit | Read 01 / Write 05
  - **shr**: Single Holding Register | Word | Read 03 / Write 06
- O parâmetro **mmmmm** indica um endereço de memória Modbus
- O parâmetro **.tt** é opcional e indica o tipo de interpretação aplicada aos dados. Se não é informado, os dados são interpretados como um tipo de dados **Word**:
  - **.u16**: Word
  - **.i16**: Inteiro de 16 bits
  - **.u32**: DWord
  - **.i32**: Inteiro de 32 bits
  - **.f**: Ponto flutuante de 32 bits
  - **.d**: Double de 64 bits
- O parâmetro **.ss** é opcional e indica a ordem de bytes a ser aplicada nos dados. Se não é informado, a ordem de bytes aplicada é **.msb** (by7 by6 by5 by4 by3 by2 by1 by0):
  - **.sb**: Swap Byte (by6 by7 by4 by5 by2 by3 by0 by1)
  - **.sw**: Swap Word (by5 by4 by7 by6 by1 by0 by3 by2)
  - **.swsb**: Swap Word + Byte (by4 by5 by6 by7 by0 by1 by2 by3)

- **.sdw**: Swap DWord (by3 by2 by1 by0 by7 by6 by5 by4)
- **.sdwsb**: Swap DWord + Byte (by2 by3 by0 by1 by6 by7 by4 by5)
- **.sdwsw**: Swap DWord + Word (by1 by0 by3 by2 by5 by4 by7 by6)
- **.sdwswsb**: Swap DWord + Word + Byte (by0 by1 by2 by3 by4 by5 by6 by7)

Exemplos de Uso

Leitura de um *Holding Register* na posição de memória 120:

- **Tag.ParamDevice**: 1 (um)
- **Tag.ParamItem**: hr120

Leitura de um *Holding Register* na posição de memória 120 com bytes invertidos (*Swap Byte*):

- **Tag.ParamDevice**: 1 (um)
- **Tag.ParamItem**: hr120.sb

Leitura de um *Holding Register* na posição de memória 120 com bytes invertidos (*Swap Byte*) e um inteiro com sinal (*int16*):

- **Tag.ParamDevice**: 1 (um)
- **Tag.ParamItem**: hr120.i16.sb

Documentação das Interfaces de Comunicação

Esta seção contém a documentação das Interfaces de Comunicação referentes ao Driver **CT850**.

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.

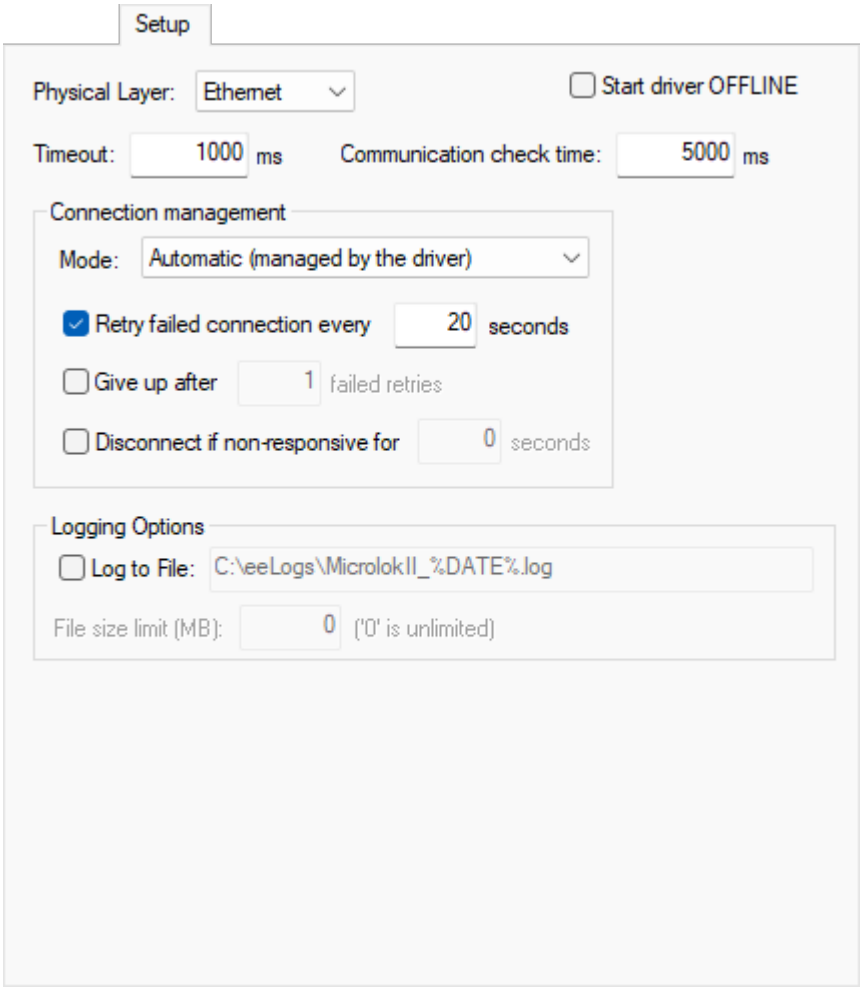
Caixa de Diálogo de Configuração

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

OPÇÃO	DESCRIÇÃO
Communication check time	Configure o tempo, em milissegundos, para definir o intervalo em que a comunicação é considerada em estado inativo. Enquanto um Driver de Comunicação receber dados válidos, o estado de comunicação é considerado ativo. Porém, se durante o funcionamento um Driver de Comunicação não receber dados válidos neste período de tempo, o estado é considerado inativo. O estado de comunicação é mostrado no Tag **IO.CommunicationStatus**
Start driver OFFLINE	Selecione esta opção para que um Driver inicie em modo **Offline** ou parado. Isto significa que a interface de I/O não é criada até que se configure um Driver em modo **Online** utilizando-se um Tag em uma aplicação. Este modo possibilita a configuração dinâmica da interface de I/O em tempo de execução

Opções para o grupo Connection management

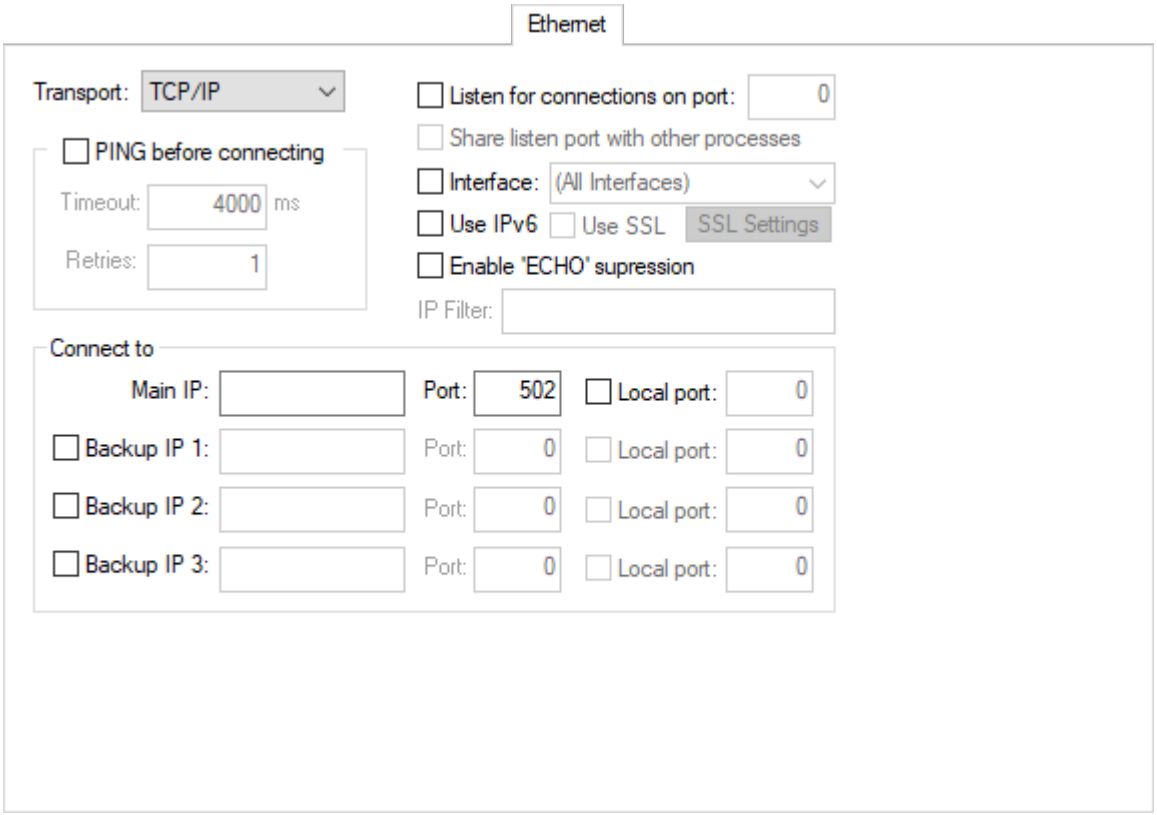
OPÇÃO	DESCRIÇÃO
Mode	Seleciona o modo de gerenciamento de conexão. Selecionar a opção **Automatic** permite que um Driver gerencie a conexão automaticamente, como especificado nas opções seguintes. Selecionar a opção **Manual** permite que uma aplicação gerencie a conexão completamente
Retry failed connection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

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

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

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

IO.Ethernet.MainIP

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
1.0.5	22/05/2026	C. Mello	• Ajustes para prevenir a inversão de **SetPoints de Bloqueio** nas Centrais **AUTTOM** (*Case 39905*).
1.0.4	26/01/2026	C. Mello	• Adicionado o telecomando **SP_ANALOG** para efetuar escritas com Setpoints analógicos nas Centrais **CT850** (*Case 39332*).
1.0.3	30/12/2025	C. Mello	• Ajustes nas Centrais **AUTTOM** para seguir redirecionamento efetuado por um sistema legado ao enviar telecomandos pelo Radio 2 (*Case 39277*).
1.0.2	30/09/2025	M. Ludwig	• Driver atualizado para a biblioteca **IOKit** versão **3.0** e Visual Studio 2022 (*Case 37978*).
		A. Fetzner	• Corrigido o retorno de *feedback* de um comando nas Centrais **AUTTOM**, que estava com a posição invertida (*Case 38143*).
		C. Mello	• Adicionadas falhas físicas de comunicação aos códigos de retorno de uma Central (*Case 38615*). • Adicionada uma *bufferização* das escritas individuais dos Setpoints (*Case 37716*).
1.0.1	11/04/2025	C. Mello	• Versão inicial deste Driver (*Case 37522*).

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 © 2025-2026 Elipse Software Ltda. Todos os direitos reservados.