

Driver Modicon ModbusSlave

Nome do Arquivo	ModbusSlave.dll
Fabricante	Modicon Schneider Electric
Equipamentos	Qualquer equipamento compatível com o protocolo Modbus RTU/ASCII/TCP em modo Escravo
Protocolo	Modbus RTU/ASCII/TCP em modo Escravo
Versão	3.1.7
Última Atualização	10/08/2026
Plataforma	Win32
Dependências	IOKit versão 2.0 ou superior
Leitura de Superblocos	Não
Nível	0

Introdução

Este Driver permite a comunicação entre aplicações da **Elipse Software** e dispositivos externos que suportam o protocolo padrão Modicon Modbus RTU/ASCII/TCP. Este Driver foi projetado para operar no modo **Escravo**, em que uma aplicação da **Elipse Software** atua como **Escravo** e um dispositivo externo atua como **Mestre** nas transações de comunicação e suporta as seguintes funções:

- **Leitura de Bits de Status:** Código de função 1 (um), referências 0x
- **Leitura de Bits de Entrada:** Código de função 2 (dois), referências 1x
- **Leitura de Múltiplos Registros:** Código de função 3 (três), referências 4x
- **Leitura de Múltiplos Registros de Entradas:** Código de função 4 (quatro), referências 3x
- **Escrita em um Único Bit:** Código de função 5 (cinco), referências 0x
- **Escrita em Múltiplos Bits:** Código de função 15, referências 0x
- **Escrita em um Único Registro:** Código de função 6 (seis), referências 4x
- **Escrita em Múltiplos Registros:** Código de função 16, referências 4x
- **Leitura de Diagnóstico:** Código de função 8 (oito)

Estas funções são universalmente implementadas e completamente consistentes com sistemas em conformidade com o protocolo Modbus RTU/ASCII/TCP.

NOTA

A função **Leitura de Diagnóstico** (código de função oito) foi adicionada apenas para fins de compatibilidade, ou seja, é realizado apenas um eco da mensagem enviada por um **Mestre**, sem contabilizar qualquer informação relacionada a contadores de diagnósticos.

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 realizadas na janela de configurações ou na janela **Extra Settings** no **Elipse SCADA**. Nesta janela, a aba **ModbusSlave** contém as configurações específicas deste Driver, enquanto as demais abas apresentam as 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 ModbusSlave, informando inclusive o valor das **Strings** de propriedades *offline*, que podem ser programadas pelo usuário ao iniciar uma aplicação em modo **Offline**. Estas **Strings** podem estar listadas em uma tabela, conforme o exemplo a seguir.

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

Opções disponíveis na aba ModbusSlave

OPÇÃO	STRING	TIPO DE DADOS	DESCRIÇÃO
Modbus Protocol Settings	ModbusSlave.MB_Protocol	Número	Define o formato do protocolo. Os valores <i>offline</i> válidos são 0 : Modbus RTU ou 1 : Modbus ASCII
Use Modbus TCP in Ethernet	ModbusSlave.UseModbusTCP	Booleano	Se esta opção está habilitada, utiliza o protocolo Modbus TCP em modo Ethernet TCP/IP . Se esta opção está desabilitada, utiliza o protocolo serial RTU/ASCII em modo Ethernet TCP/IP . Os valores <i>offline</i> válidos são 0 : Desabilitado ou 1 : Habilitado
Stores data from Master with base 1 memory address	ModbusSlave.UseMemBase1	Booleano	A memória interna dos equipamentos Modbus Escravos possui endereçamento base 0 (zero, nível de hardware), ou seja, inicia pelo

OPÇÃO	STRING	TIPO DE DADOS	DESCRIÇÃO
			endereço 0x0000. De modo geral, os equipamentos Modbus Mestres oferecem endereçamento base 1 (um, nível de usuário), ou seja, iniciam pelo endereço 0x0001 mas, quando envia algum dado a um Escravo, o protocolo já se encarrega de endereçar na base 0 (zero), decrementando -1 (menos um) no endereço informado pelo usuário. Se esta opção está habilitada, os dados enviados por um Mestre são gravados com base 1 (um), incrementando +1 (mais um) ao endereço destino da memória interna de um equipamento Modbus Escravo, tornando-a equivalente ao endereçamento de um equipamento Modbus Mestre no nível do usuário. Os valores <i>offline</i> válidos são 0: Desabilitado ou 1: Habilitado
Use Slave IDs (1, 2, 3, ..., 247, or empty for all)	ModbusSlave.UseSlaveID	Texto	Define uma lista específica de identificadores de equipamentos Modbus Escravos, separados por vírgulas, para comunicação com um equipamento Modbus Mestre. Deixe este campo em branco para que qualquer identificador Escravo responda a um Mestre. O formato válido para este parâmetro é 1, 2, 3, ... n (Endereço do Escravo um, Endereço do Escravo dois, Endereço do Escravo 3, ..., Endereço do Escravo n)
Use Buffer Memory	ModbusSlave.UseBuffMemory	Booleano	Se esta opção está habilitada, permite armazenar mais de um dado no mesmo registro de memória. Os valores <i>offline</i> válidos são 0: Desabilitado ou 1: Habilitado

OPÇÃO	STRING	TIPO DE DADOS	DESCRIÇÃO
Amount of data per register	ModbusSlave.DataPerRegister	Número	Define a quantidade de dados que podem ser empilhados em um mesmo registro de memória
Hold latest data (prevents 'zero' value when the buffer is empty)	ModbusSlave.HoldLatestData	Booleano	Se esta opção está habilitada, preserva a presença do último dado armazenado no registro de memória. Os valores <i>offline</i> válidos são 0 : Desabilitado ou 1 : Habilitado

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 de configuração das outras abas, consulte o tópico Documentação das Interfaces de Comunicação.

NOTAS

- Para comunicações em modo **Ethernet**, a porta TCP/IP padrão usada pelo protocolo Modbus TCP é a porta 502.
- Com a opção **Use Buffer Memory** habilitada, a lista de dados existente em cada registro de memória deve ser coletada usando um evento **OnRead** de um Tag PLC ou Bloco.

Configurando os Tags de uma Aplicação da Elipse Software

Um desenvolvedor de aplicações da **Elipse Software** deve criar Tags PLC ou Bloco para permitir acesso de usuários aos registros locais de um equipamento Modbus, alocados dinamicamente na memória interna deste Driver. Estes registros locais de um equipamento Modbus são exclusivos para cada identificador **Escravo**. O tamanho destes Tags Bloco, ou seja, o número de Elementos, depende do número de registros Modbus lidos ou escritos e do tipo de dados utilizado em um Tag Bloco.

Para aplicações **E3**, **Power** ou **Water**, existe a possibilidade de efetuar leitura de Tags por *callback*, ou seja, obter o retorno dos dados imediatamente, sem aguardar pela varredura, ou *scan*, de um Tag, assim que os dados são atualizados por um equipamento Modbus **Mestre** ou via escrita de Tags pelo usuário.

Limites para Tags Bloco

TIPO DE DADOS	QUANTIDADE DE ELEMENTOS
Registros de coils (registrador de um bit)	Máximo de 2000 Elementos
Registros de dois bytes (registrador de 16 bits)	Máximo de 125 Elementos
Registros de quatro bytes (2 × registrador de 16 bits)	Máximo de 62 Elementos

TIPO DE DADOS	QUANTIDADE DE ELEMENTOS
Registros de oito bytes (4 × registrador de 16 bits)	Máximo de 31 Elementos

Parâmetros [N/B] para Tags PLC ou Bloco

Leitura ou Escrita

Utilize os parâmetros [N/B] para configurar Tags numericamente, conforme a tabela a seguir.

Dispositivo	Não utilizado
Item	Use o valor CB para definir uma leitura por callbacks, somente no E3 , Power ou Water
N1 ou B1	Identificador de um Escravo , entre 1 (um) e 247
N2 ou B2	Use o valor Ordem de Bytes + Tipo de Dados . Para mais informações, consulte as tabelas Tipos de dados e Ordem dos bytes
N3 ou B3	Tamanho de uma String , caso o parâmetro <i>N2</i> ou <i>B2</i> seja igual a 7 (sete) ou 8 (oito)
N4 ou B4	Endereço de uma variável dos registros locais de um equipamento Modbus, entre 0 (zero) e 65535

Tipos de dados

CÓDIGO	TIPO DE DADOS	TAMANHO	REGISTRO MODBUS
1	Bit (registro de <i>Coils</i>)	1 (um) byte	Ocupa 1 (um) bit do registro de <i>Coils</i> de um equipamento Modbus
2	Word (inteiro de 16 bits sem sinal)	2 (dois) bytes	Ocupa 1 (um) registro em um equipamento Modbus
3	Short (inteiro de 16 bits com sinal)	2 (dois) bytes	Ocupa 1 (um) registro em um equipamento Modbus
4	Double Word (inteiro de 32 bits sem sinal)	4 (quatro) bytes	Ocupa 2 (dois) registros em um equipamento Modbus

CÓDIGO	TIPO DE DADOS	TAMANHO	REGISTRO MODBUS
5	Int (inteiro de 32 bits com sinal)	4 (quatro) bytes	Ocupa 2 (dois) registros em um equipamento Modbus
6	Float (ponto flutuante de 32 bits, EXP F0 F1 F2)	4 (quatro) bytes	Ocupa 2 (dois) registros em um equipamento Modbus
7	String (texto no formato ASCII)	<i>N</i> bytes	A cada 2 (dois) caracteres, ocupa 1 (um) registro em um equipamento Modbus. Por exemplo, a String "12" é igual a 0x3132
8	String Hexadecimal (texto no formato ASCII com notação hexadecimal)	<i>N</i> bytes	A cada 4 (quatro) caracteres, ocupa 1 (um) registro em um equipamento Modbus. Por exemplo, a String hexadecimal "04D2" é igual a 0x04D2
9	Double (ponto flutuante de 64 bits, EXP F0 F1 F2 F3 F4 F5 F6)	8 (oito) bytes	Ocupa 4 (quatro) registros em um equipamento Modbus

Ordem dos bytes

CÓDIGO	ORDEM	DESCRIÇÃO
100	Swap Byte	Word 0x2211 resulta em 0x1122 (bytes invertidos)
200	Swap Word	DWord 0x44332211 resulta em 0x22114433 (Word invertido)
300	Swap Word + Swap Byte	DWord 0x44332211 resulta em 0x11223344 (Word invertido + bytes invertidos)
400	Swap DWord	QWord 0x8877665544332211 resulta em 0x4433221188776655 (DWord invertido)
500	Swap DWord + Swap Byte	QWord 0x8877665544332211 resulta em 0x3344112277885566 (DWord invertido + bytes invertidos)
600	Swap DWord + Swap Word	QWord 0x8877665544332211 resulta em 0x2211443366558877 (DWord invertido + Word invertido)
700	Swap DWord + Swap Word + Swap Byte	QWord 0x8877665544332211 resulta em 0x1122334455667788 (DWord invertido + Word invertido + bytes invertidos)

NOTAS

- Para o tipo de dados **String**, o tamanho máximo permitido no parâmetro *N3* é de 250 caracteres para um Tag PLC. Já para um Tag Bloco, o tamanho máximo de 250 caracteres deve ser dividido pelo número de Elementos deste Tag Bloco. Por exemplo, para um Tag Bloco de 10 Elementos, o tamanho máximo permitido no parâmetro *B3* é de 25 caracteres.
- Para o tipo de dados **String Hexadecimal**, o tamanho máximo permitido no parâmetro *N3* é de 500 caracteres para um Tag PLC. Já para um Tag Bloco, o tamanho máximo de 500 caracteres deve ser dividido pelo número de Elementos deste Tag Bloco. Por exemplo, para um Tag Bloco de 10 Elementos, o tamanho máximo permitido no parâmetro *B3* é de 50 caracteres.

Parâmetros [Dispositivo/Item] para Tags PLC ou Bloco

Leitura ou Escrita

Para aplicações **E3**, **Power** ou **Water**, pode-se deixar os parâmetros **[N/B]** zerados e configurar Tags por **Strings** usando os campos **Dispositivo** e **Item**, conforme a tabela a seguir.

Dispositivo	Identificador de um Escravo , entre 1 (um) e 247
Item	Parâmetros no formato [CB:]Rn[.tipo-de-dado][.ordem-dos-bytes][.tamanho-da-string]
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 campo **Dispositivo**, informe o identificador de um **Escravo** para ler ou escrever dados nos registradores de memória via Tags de uma aplicação.

No campo **Item**, informe uma **String** de configuração no formato **[CB:]Rn[.tipo-de-dados][.ordem-dos-bytes][.tamanho-da-string]**, conforme descrito a seguir.

- CB**: Identificador para definir uma leitura por *callbacks* neste Tag. Se este valor não é informado, a leitura deste Tag é realizada por varredura, ou *scan*, padrão
- Rn**: Posição inicial de memória, *Coil* ou *Register*. Por exemplo, um valor **R99** manipula um *Register* 99 e um valor **R99.bit** manipula um *Coil* 99
- .tipo-de-dados**: Define o tipo de interpretação aplicada aos dados. Se este valor não é informado, assume o tipo de dados **.u16**. Os valores possíveis são os seguintes:
 - .bit**: Indica dados com valor 1 (um) ou 0 (zero) para registradores *Coil*
 - .u16**: Indica um valor inteiro de 16 bits sem sinal, ou 2 (dois) bytes ocupando 1 (um) registrador Modbus
 - .i16**: Indica um valor inteiro de 16 bits com sinal, ou 2 (dois) bytes ocupando 1 (um) registrador Modbus

- **.u32:** Indica um valor inteiro de 32 bits sem sinal, ou 4 (quatro) bytes ocupando 2 (dois) registradores Modbus
- **.i32:** Indica um valor inteiro de 32 bits com sinal, ou 4 (quatro) bytes ocupando 2 (dois) registradores Modbus
- **.f:** Indica um valor de ponto flutuante de 32 bits, EXP F0 F1 F2, ou 4 (quatro) bytes ocupando 2 (dois) registradores Modbus
- **.d:** Indica um valor de ponto flutuante de 64 bits, EXP F0 F1 F2 F3 F4 F5 F6, ou 8 (oito) bytes ocupando 4 (quatro) registradores Modbus
- **.str:** Indica dados como **Strings**, por exemplo o valor "12" é igual a 0x3132, ou *n* bytes ocupando 1 (um) registrador Modbus a cada 2 (dois) caracteres
- **.strhex:** Indica dados como **Strings** em notação hexadecimal, por exemplo o valor "04D2" é igual a 0x04D2, ou *n* bytes ocupando 1 (um) registrador Modbus a cada 4 (quatro) caracteres
- **.ordem-dos-bytes:** Define a ordem dos bytes aplicada aos dados em memória. Se este valor não é informado, assume a ordem *Big-Endian*, como por exemplo o valor 1234 nesta ordem é igual a 0x04D2. Os valores possíveis são os seguintes:
- **.sb: Swap Byte.** Por exemplo, um valor **Word** 0x2211 resulta em 0x1122 (bytes invertidos)
- **.sw: Swap Word.** Por exemplo, um valor **DWord** 0x44332211 resulta em 0x22114433 (**Word** invertido)
- **.swsb: Swap Word + Swap Byte.** Por exemplo, um valor **DWord** 0x44332211 resulta em 0x11223344 (**Word** invertido + bytes invertidos)
- **.sdw: Swap DWord.** Por exemplo, um valor **QWord** 0x8877665544332211 resulta em 0x4433221188776655 (**DWord** invertido)
- **.sdwsb: Swap DWord + Swap Byte.** Por exemplo, um valor **QWord** 0x8877665544332211 resulta em 0x3344112277885566 (**DWord** invertido + bytes invertidos)
- **.sdwsw: Swap DWord + Swap Word.** Por exemplo, um valor **QWord** 0x8877665544332211 resulta em 0x2211443366558877 (**DWord** invertido + **Word** invertido)
- **.sdwsbsb: Swap DWord + Swap Word + Swap Byte.** Por exemplo, um valor **QWord** 0x8877665544332211 resulta em 0x1122334455667788 (**DWord** invertido + **Word** invertido + bytes invertidos)
- **.tamanho-da-string:** Indica o tamanho, em caracteres, para os tipos de dados **.str** ou **.strhex**. Por exemplo, um valor **R99.str.12** indica uma leitura ou escrita de 12 bytes no formato **String** a partir de um registrador de memória 99

NOTAS

- Para o tipo de dados **.str**, o tamanho máximo permitido é de 250 caracteres para um Tag PLC. Já para um Tag Bloco, o tamanho máximo de 250 caracteres deve ser dividido pelo número de Elementos deste Tag Bloco. Por exemplo, para um Tag Bloco de 10 Elementos, o tamanho máximo permitido é de 25 caracteres.
- Para o tipo de dados **.strhex**, o tamanho máximo permitido é de 500 caracteres para um Tag PLC. Já para um Tag Bloco, o tamanho máximo de 500 caracteres deve ser dividido pelo número de Elementos deste Tag Bloco. Por exemplo, para um Tag Bloco de 10 Elementos, o tamanho máximo permitido é de 50 caracteres.

Documentação das Interfaces de Comunicação

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

Esta documentação é compartilhada e mantida no manual do IOKit:

- Configurações de um Driver
- Configurações Gerais
- Configuração de Estatísticas
- Configuração da Interface Ethernet
- Configuração da Interface Modem
- Configuração da Interface RAS
- Configuração da Interface Serial

Histórico de Revisões do Driver

VERSÃO	DATA	AUTOR	COMENTÁRIOS
3.1.7	10/08/2026	C. Mello	• Adicionado suporte para leituras por <i>callbacks</i> e configuração de Tags por Strings (Case 36886). • Adicionado suporte para o formato String Hexadecimal dos tipos de dados (Case 38656). • Adicionada uma configuração para definir a ordem dos bytes aplicada aos dados (Case 37837).
		M. Ludwig	• Driver atualizado para a biblioteca IOKit versão 3 e Visual Studio 2022 (Case 37996).
3.1.6	10/12/2024	M. Ludwig	• Adicionada, via biblioteca IOKit, uma configuração para o tamanho máximo de arquivos de log na aba Setup da janela de propriedades deste Driver (Case 36834).
3.1.5	07/05/2021	C. Mello	• Adicionado suporte para dados no formato de ponto flutuante de 64 bits (Case 30348).
3.1.4	06/08/2019	M. Ludwig	• Driver portado para o Visual Studio 2017 (Case 27191).
3.1.2	15/12/2017	C. Mello	• Corrigida a quantidade máxima de bits por requisição de <i>Coils</i> (Case 23788).
3.1.1	26/11/2014	C. Mello	• Adicionada uma opção para armazenar dados de um Mestre no endereço de memória equivalente ao do nível de usuário (Case 16474). • Adicionado suporte para o código de função 8 do protocolo Modbus (Case 17714).
3.0.1	03/09/2014	C. Mello	• Ajustes de migração para a biblioteca IOKit versão 2 (Case 16472).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
2.5.1	14/05/2010	C. Mello	• Adicionado suporte para acesso a registros do tipo Bit com o parâmetro <i>N2</i> igual a 1 (um) (<i>Case 10993</i>). • Adicionada uma opção para empilhar dados na mesma posição de memória (<i>Case 11166</i>).
2.4.1	19/01/2009	C. Mello	• Ajustes para permitir o endereçamento de memória na posição 65535 (<i>Case 9935</i>).
2.3.1	28/01/2008	C. Mello	• Adicionado um novo tipo de dados Reverse Float com o parâmetro <i>N2</i> igual a 8 (oito) (<i>Case 9108</i>). • Ajustes para receber uma mensagem completa de um Mestre antes de descartar esta mensagem por não corresponder a um identificador de um Escravo associado a este Driver (<i>Case 9109</i>).
2.2.1	16/02/2007	C. Mello	• Adicionada uma opção de leitura ou escrita no formato String com o parâmetro <i>N2</i> igual a 7 (sete) (<i>Case 7126</i>). • Implementada a compatibilidade com o Windows CE (<i>Case 7173</i>).
2.1.1	22/12/2005	C. Mello	• Adicionada uma opção de criar uma lista de Escravos considerados válidos para a comunicação.
2.0.1	13/06/2005	C. Mello	• Versão implementada com a biblioteca IOKit.
1.1.1	25/06/2004	C. Mello	• Versão inicial sem a biblioteca IOKit.