

## Driver MITSUBISHI MitsubishiMC

|                         |                                                                                                                                                   |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| Nome do Arquivo         | MitsubishiMC.dll                                                                                                                                  |
| Fabricante              | Mitsubishi Electric Brasil                                                                                                                        |
| Equipamentos            | Melsec série FX3 com módulos FX3U-ENET ou FX3U-ENET-ADP, Melsec FX3GE com porta Ethernet embutida e Melsec série A e QnA com módulos Ethernet E71 |
| Protocolo               | Melsec communication binário em frame 1E compatível A sobre Ethernet                                                                              |
| Versão                  | 1.0.3                                                                                                                                             |
| Última Atualização      | 10/08/2026                                                                                                                                        |
| Plataforma              | Win32                                                                                                                                             |
| Dependências            | IOKit versão 2.0 ou superior                                                                                                                      |
| Leitura com Superblocos | Não                                                                                                                                               |
| Nível                   | 0                                                                                                                                                 |

## Introdução

Este Driver implementa o protocolo de comunicação Melsec communication binário em frame 1E compatível A sobre Ethernet da Mitsubishi Electric Brasil, que permite aos produtos da **Elipse Software** se comunicarem com os equipamentos Melsec série FX3 com módulos FX3U-ENET ou FX3U-ENET-ADP, Melsec FX3GE com porta Ethernet embutida e Melsec série A e QnA com módulos Ethernet E71.

O Driver MITSUBISHI MitsubishiMC foi desenvolvido em conjunto com a biblioteca **IOKit** da **Elipse Software**.

## Configuração do Driver

Todas as configurações são realizadas na janela de propriedades deste Driver. Ao incluir este Driver em uma aplicação, este é carregado com a configuração padrão de parâmetros extras para a interface **Ethernet**, ou seja, porta TCP/IP 5001 e *time-out* de 1000 milissegundos. É necessário abrir a janela de propriedades deste Driver e configurar um endereço IP principal válido referente ao dispositivo a ser acessado. Para mais informações sobre as abas **Setup** e **Ethernet**, consulte o tópico Documentação das Interfaces de Comunicação.

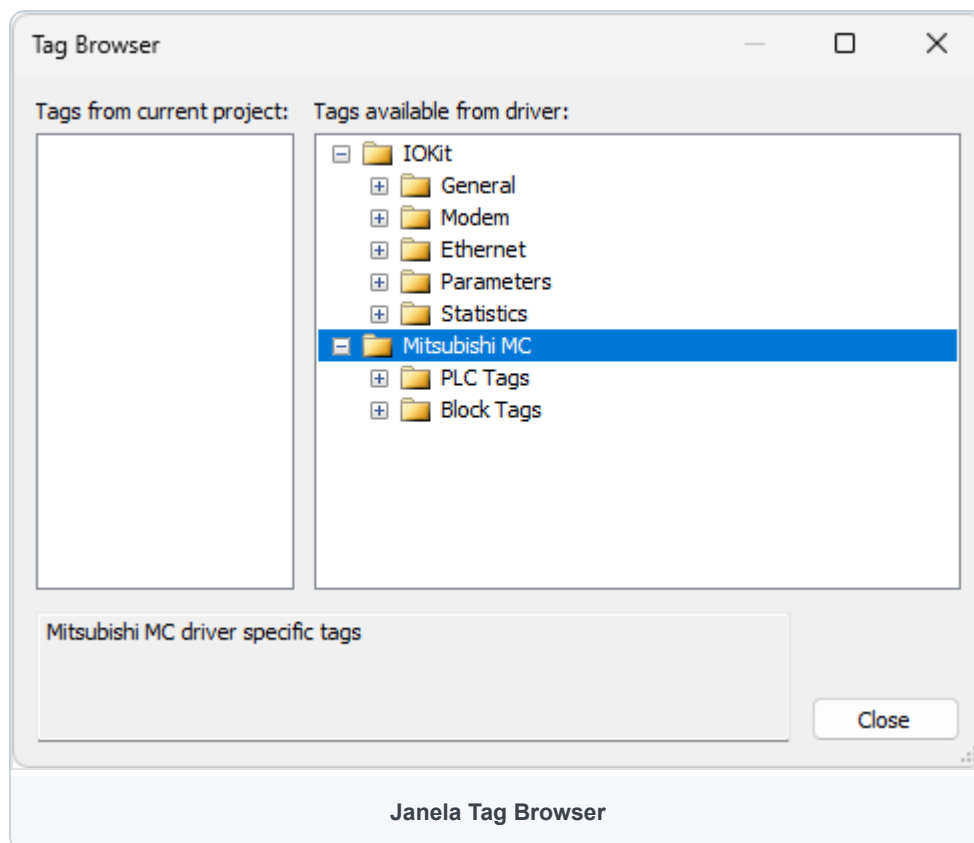
A aba **MitsubishiMC** contém somente a opção **Standard maximum string length**, que corresponde ao tamanho máximo padrão de uma variável do tipo **String**. O valor padrão desta opção é 128 e este valor pode variar entre 1 (um) e 128.

## Referência de Tags

Esta seção contém informações sobre a configuração dos Tags deste Driver. Existe a possibilidade de configurar estes Tags através de **Strings** no campo **Item** no **E3**, no **Power** ou no **Water** ou por parâmetros **[N/B]**.

## Utilizando o Tag Browser

O **E3** a partir da versão **2**, o **Power** e o **Water** possuem uma ferramenta chamada **Tag Browser**, que permite que este Driver auxilie o usuário na criação e configuração de Tags.



A lista **Tags do projeto corrente** (*Tags from current project*) mostra os Tags e pastas que existem no projeto corrente. A lista **Tags disponibilizados pelo driver** (*Tags available from driver*) mostra uma árvore com os Tags disponíveis neste Driver. Para criar um Tag em uma aplicação, arraste um dos Tags definidos por este Driver para uma pasta no diretório corrente. O Driver MITSUBISHI MitsubishiMC contém os seguintes nós nesta árvore:

- O nó **IOKit** mostra os Tags disponíveis na biblioteca **IOKit**, agrupados nas seguintes categorias:
- **General**: Tags de uso geral
- **Modem**: Tags de manipulação da comunicação via modem
- **Ethernet**: Tags de manipulação da comunicação via rede Ethernet
- **Parameters**: Tags para configuração dos parâmetros da biblioteca **IOKit**

- O nó **Mitsubishi MC** possui pastas classificando o tipo de Tags que contêm, **Tags PLC** (*PLC Tags*) ou **Tags Bloco** (*Block Tags*). Em cada uma destas pastas há outras pastas separando os Tags por Área, e estas por sua vez contêm os nós finais, que são os exemplos de Tags de cada área. Existe um exemplo para cada tipo de Tag previsto. Arraste um dos Tags para a listagem à esquerda, ou mesmo o nó de uma pasta inteira para mover todos os Tags ao mesmo tempo, e este Tag ou Tags já estão devidamente configurados para o funcionamento

O Tag Browser para o Driver MITSUBISHI MitsubishiMC visa automatizar a configuração dos Tags através de exemplos para cada área e cada tipo de Tag suportado. Evidentemente que na grande maioria dos casos é útil incluir apenas os Tags de áreas e tipos de Tags de interesse de um projeto, necessitando de adequações manuais posteriores. O Tag Browser sugere o valor "00000" (zero) para o Endereço de uma área, no campo **Item**, que muito provavelmente deve ser reconfigurado após a inclusão dos Tags. Ao selecionar um Tag Bloco, este já aparece com um tamanho sugerido de 10 Elementos, que também pode ser posteriormente modificado.

## Configuração por Parâmetros Sintáticos

### Leitura e Escrita

O **E3** a partir da versão **2**, o **Power** e o **Water** permitem a configuração de Tags por parâmetros sintáticos. Utilize a sintaxe a seguir para cada campo:

- O campo **Dispositivo** não é usado e deve ser mantido em branco
- O campo **Item** deve obedecer à sintaxe **<Nome da área><Endereço>[:<Tipo>]**, em que:
  - <Nome da área>** deve ser preenchido com alguma das **Strings** da tabela Nomes de Área
  - <Endereço>** deve ser preenchido com o valor do endereço na área ou *offset*
  - <Tipo>** é opcional e, caso seja utilizado, deve ser separado por dois pontos e preenchido com alguma das **Strings** da tabela Tipos de Dados. Se o tipo de dados é omitido, é exibido o padrão para a área

#### Nomes de Área

| ÁREA | SIGNIFICADO        | TIPOS DE DADOS SUPORTADOS                                                                                                                               |
|------|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| D    | Data register      | <b>Word</b> (padrão), <b>Short</b> , <b>BCD</b> , <b>DWord</b> , <b>Long</b> , <b>LBCD</b> , <b>Float</b> , <b>Double</b> , <b>Date</b> e <b>String</b> |
| R    | Extension register | <b>Word</b> (padrão), <b>Short</b> , <b>BCD</b> , <b>DWord</b> , <b>Long</b> , <b>LBCD</b> , <b>Float</b> , <b>Double</b> , <b>Date</b> e <b>String</b> |
| TN   | Timer value        | <b>Word</b> (padrão), <b>Short</b> e <b>BCD</b>                                                                                                         |
| TS   | Timer contact      | <b>Bit</b> (padrão), <b>Word</b> , <b>Short</b> , <b>BCD</b> , <b>DWord</b> , <b>Long</b> e <b>LBCD</b>                                                 |
| CN   | Counter value      | <b>Word</b> (padrão), <b>Short</b> , <b>BCD</b> , <b>DWord</b> , <b>Long</b> e <b>LBCD</b>                                                              |
| CS   | Counter contact    | <b>Bit</b> (padrão), <b>Word</b> , <b>Short</b> , <b>BCD</b> , <b>DWord</b> , <b>Long</b> e <b>LBCD</b>                                                 |

| ÁREA | SIGNIFICADO     | TIPOS DE DADOS SUPORTADOS                                                                               |
|------|-----------------|---------------------------------------------------------------------------------------------------------|
| X    | Input           | <b>Bit</b> (padrão), <b>Word</b> , <b>Short</b> , <b>BCD</b> , <b>DWord</b> , <b>Long</b> e <b>LBCD</b> |
| Y    | Output          | <b>Bit</b> (padrão), <b>Word</b> , <b>Short</b> , <b>BCD</b> , <b>DWord</b> , <b>Long</b> e <b>LBCD</b> |
| M    | Auxiliary relay | <b>Bit</b> (padrão), <b>Word</b> , <b>Short</b> , <b>BCD</b> , <b>DWord</b> , <b>Long</b> e <b>LBCD</b> |
| S    | State           | <b>Bit</b> (padrão), <b>Word</b> , <b>Short</b> , <b>BCD</b> , <b>DWord</b> , <b>Long</b> e <b>LBCD</b> |

### Tipos de Dados

| TIPO DE DADOS  | SIGNIFICADO                                                                                 |
|----------------|---------------------------------------------------------------------------------------------|
| W ou WORD      | Valor <b>Word</b> (16 bits sem sinal)                                                       |
| S, SH ou SHORT | Valor <b>Short</b> (16 bits com sinal)                                                      |
| DW ou DWORD    | Valor <b>Double Word</b> (32 bits sem sinal)                                                |
| L ou LONG      | Valor <b>Long</b> (32 bits com sinal)                                                       |
| F ou FLOAT     | Valor <b>Floating Point</b> (32 bits de ponto flutuante, IEEE 754)                          |
| D ou DOUBLE    | Valor <b>Double Floating Point</b> (64 bits de ponto flutuante de dupla precisão, IEEE 754) |
| BCD            | Valor <b>BCD</b> (16 bits)                                                                  |
| LBCD           | Valor <b>BCD Long</b> (32 bits)                                                             |
| DATE           | Valor de 32 bits                                                                            |
| STR ou STRING  | Sequência de caracteres <b>String</b>                                                       |

A sintaxe para o campo **Item**, caso seja selecionado o tipo de dados **String**, é a seguinte:

- **<Nome da área><Endereço>:STR[<Tamanho máximo>]**, em que:
- **<Nome da área>** e **<Endereço>** referem-se aos mesmos itens da sintaxe geral
- **<Tamanho máximo>** é opcional e informa o tamanho máximo declarado na **String**. Se não é informado, considera-se o tamanho máximo padrão da **String** como preenchido na janela de configuração deste Driver

**NOTA**

A qualquer área cujo tipo de dados padrão seja **Bit**, quando acessada com outro tipo de dados, deve ser endereçada com um valor múltiplo de 16.

## Exemplos

1. Leitura ou escrita de *Input* de endereço 1 (um) com tipo de dados básico **Bit**: Campo **Item** preenchido com "X1".
2. Leitura ou escrita de *Output* de endereço 100 com tipo de dados configurado como **Word**: Campo **Item** preenchido com "Y100:W". Os valores de 16 *Outputs* contíguos a partir do endereço 100 são reinterpretados como um valor **Word**.
3. Leitura ou escrita de *Counter Value* de endereço 50 com tipo de dados básico **Word**: Campo **Item** preenchido com "CN50". Preencher com "CN50:W" produz o mesmo resultado.
4. Leitura ou escrita de *Data Register* de endereço 1000 com tipo de dados configurado como **Double Word**: Campo **Item** preenchido com "D1000:DW". Os valores de dois *Data Registers* contíguos a partir do endereço 1000 são reinterpretados como um valor **Double Word**.
5. Leitura ou escrita de *Data Register* de endereço 100 com tipo de dados configurado como **String** de tamanho máximo 64: "D100:STR[64]".

## Configuração por Parâmetros Numéricos

### Leitura e Escrita

#### Campos de endereçamento por parâmetros N/B

- **N1 ou B1**: Não utilizado, deve ser deixado em 0 (zero)
- **N2 ou B2**: Tipo de dados e Área. O valor deve ser composto pela soma do tipo de dados mais a área (*N2 ou B2* igual a Valor de Tipo de Dados + Valor de Área). Se o valor de **Tipo** é igual a 0 (zero), assume o tipo de dados básico da área
- **N3 ou B3**: Valor do endereço na área ou *offset*
- **N4 ou B4**: Usado apenas se o tipo de dados é **String**. Informa o tamanho máximo declarado na **String**. Se não é informado, considera-se o tamanho máximo padrão da **String** como preenchido na janela de configuração deste Driver

#### Valores de área

| VALOR | SIGNIFICADO        | TIPOS DE DADOS SUPORTADOS                                                                                                                               |
|-------|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0     | Data register      | <b>Word</b> (padrão), <b>Short</b> , <b>BCD</b> , <b>DWord</b> , <b>Long</b> , <b>LBCD</b> , <b>Float</b> , <b>Double</b> , <b>Date</b> e <b>String</b> |
| 1     | Extension register | <b>Word</b> (padrão), <b>Short</b> , <b>BCD</b> , <b>DWord</b> , <b>Long</b> , <b>LBCD</b> , <b>Float</b> , <b>Double</b> , <b>Date</b> e <b>String</b> |
| 2     | Timer value        | <b>Word</b> (padrão), <b>Short</b> e <b>BCD</b>                                                                                                         |
| 3     | Timer contact      | <b>Bit</b> (padrão), <b>Word</b> , <b>Short</b> , <b>BCD</b> , <b>DWord</b> , <b>Long</b> e <b>LBCD</b>                                                 |

| VALOR | SIGNIFICADO     | TIPOS DE DADOS SUPORTADOS                                                                               |
|-------|-----------------|---------------------------------------------------------------------------------------------------------|
| 4     | Counter value   | <b>Word</b> (padrão), <b>Short</b> , <b>BCD</b> , <b>DWord</b> , <b>Long</b> e <b>LBCD</b>              |
| 5     | Counter contact | <b>Bit</b> (padrão), <b>Word</b> , <b>Short</b> , <b>BCD</b> , <b>DWord</b> , <b>Long</b> e <b>LBCD</b> |
| 6     | Input           | <b>Bit</b> (padrão), <b>Word</b> , <b>Short</b> , <b>BCD</b> , <b>DWord</b> , <b>Long</b> e <b>LBCD</b> |
| 7     | Output          | <b>Bit</b> (padrão), <b>Word</b> , <b>Short</b> , <b>BCD</b> , <b>DWord</b> , <b>Long</b> e <b>LBCD</b> |
| 8     | Auxiliary relay | <b>Bit</b> (padrão), <b>Word</b> , <b>Short</b> , <b>BCD</b> , <b>DWord</b> , <b>Long</b> e <b>LBCD</b> |
| 9     | State           | <b>Bit</b> (padrão), <b>Word</b> , <b>Short</b> , <b>BCD</b> , <b>DWord</b> , <b>Long</b> e <b>LBCD</b> |

#### Valores de tipos de dados

| VALOR | SIGNIFICADO                                                                                 |
|-------|---------------------------------------------------------------------------------------------|
| 0     | Tipo de dados básico de uma área ( <b>Bit</b> ou <b>Word</b> )                              |
| 100   | <b>Bit</b> (valor 1 ou 0)                                                                   |
| 200   | Valor <b>Word</b> (16 bits sem sinal)                                                       |
| 300   | Valor <b>Short</b> (16 bits com sinal)                                                      |
| 400   | Valor <b>Double Word</b> (32 bits sem sinal)                                                |
| 500   | Valor <b>Long</b> (32 bits com sinal)                                                       |
| 600   | Valor <b>Floating Point</b> (32 bits de ponto flutuante, IEEE 754)                          |
| 700   | Valor <b>Double Floating Point</b> (64 bits de ponto flutuante de dupla precisão, IEEE 754) |
| 800   | Valor <b>BCD</b> (16 bits)                                                                  |
| 900   | Valor <b>BCD Long</b> (32 bits)                                                             |
| 1000  | Valor de <b>Data</b> em 32 bits                                                             |
|       |                                                                                             |

| VALOR | SIGNIFICADO                           |
|-------|---------------------------------------|
| 1100  | Sequência de caracteres <b>String</b> |

**NOTA**

A qualquer área cujo tipo de dados padrão seja **Bit**, quando acessada com outro tipo de dados, deve ser endereçada com um valor múltiplo de 16.

## Exemplos

1. Leitura ou escrita de *Input* de endereço 1 (um), no tipo de dados padrão **Bit**:  $N1 = 0$ ,  $N2 = 0$ ,  $N3 = 1$  e  $N4 = 0$ . Se  $N2$  é preenchido com o valor "100", o resultado é o mesmo.
2. Leitura ou escrita de *Output* de endereço 100, com tipo de dados configurado como **Word**:  $N1 = 0$ ,  $N2 = 202$ ,  $N3 = 100$  e  $N4 = 0$ .
3. Leitura ou escrita de *Counter Value* de endereço 50, no tipo de dados padrão **Word**:  $N1 = 0$ ,  $N2 = 20$ ,  $N3 = 50$  e  $N4 = 0$ . Se  $N2$  é preenchido com o valor "220", o resultado é o mesmo.
4. Leitura ou escrita de *Data Register* de endereço 1000, com tipo de dados configurado como **Double Word**:  $N1 = 0$ ,  $N2 = 421$ ,  $N3 = 1000$  e  $N4 = 0$ . O valor de dois *Data Registers* contíguos a partir do endereço 1000 são reinterpretados como um **Double Word**.

## Operações com Blocos

Para um bom desempenho da comunicação de dados, é recomendável realizar leituras do maior número de pontos possíveis em uma mesma área. Quando os pontos são separados em Tags individuais, ocorre uma operação para cada Tag, enquanto que em um bloco ocorre uma única operação para todos os Tags.

Para tanto, é preciso levar em conta que o endereço indicado nos parâmetros é o endereço inicial e cada Elemento corresponde a uma posição de memória que é múltipla, conforme o tipo de dados básico relacionado ao tipo de dados reinterpretado.

Se o tipo de dados básico é igual a **Bit** e é configurado para ser mostrado como um **Bit**, cada Elemento representa um bit com incremento de posição de 1 (um) em 1 (um). Por exemplo, para *Y100* em um Bloco com 3 (três) Elementos, o primeiro Elemento é igual a *Y100*, o segundo Elemento é igual a *Y101* e o terceiro Elemento é igual a *Y102*.

No entanto, se o tipo de dados básico é igual a **Bit** e é configurado para ser mostrado como um **Word**, cada Elemento representa 16 bits com incrementos de posição de 16 em 16. Por exemplo, para *X0* em um Bloco com 3 (três) Elementos, o primeiro Elemento é igual a *X0*, o segundo Elemento é igual a *X16* e o terceiro Elemento é igual a *X32*.

## Documentação das Interfaces de Comunicação

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

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                                                                                                    |
|--------|------------|--------------|----------------------------------------------------------------------------------------------------------------|
| 1.0.3  | 10/08/2026 | M.<br>Ludwig | • Driver atualizado para a biblioteca <b>IOKit</b> versão <b>3</b> e Visual Studio 2022 ( <i>Case 37994</i> ). |
| 1.0.2  | 27/08/2019 | M.<br>Ludwig | • Driver portado para o Visual Studio 2017 ( <i>Case 27319</i> ).                                              |
| 1.0.1  | 09/04/2018 | M.<br>Ludwig | • Versão inicial deste Driver.                                                                                 |