

Driver MITSUBISHI MelsecQ

Nome do Arquivo	MelsecQ.dll
Fabricante	Mitsubishi Electric Corporation
Equipamentos	Equipamentos Melsec das séries Q e L com Ethernet
Protocolo	Melsec communication binário em frame 3E compatível QnA sobre Ethernet
Versão	1.2.3
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 3E compatível QnA sobre Ethernet da Mitsubishi Electric Corporation, que permite aos produtos da **Elipse Software** se comunicarem com equipamentos Melsec das séries Q e L com Ethernet.

O Driver MITSUBISHI MelsecQ foi desenvolvido em conjunto com a biblioteca **IOKit** da **Elipse Software**. Esta biblioteca, após configurada, é responsável por implementar o acesso a um meio físico, ou seja, **Serial**, **Ethernet**, **Modem** ou **RAS**. Para mais informações, consulte o tópico Documentação das Interfaces de Comunicação.

Configuração do Driver

Todas as configurações são realizadas na janela de configurações deste Driver. O Driver MITSUBISHI MelsecQ, ao ser incluído em uma aplicação, 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 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 **MelsecQ**, mostrada na figura a seguir, contém configurações específicas para este Driver.

Driver MITSUBISHI MelsecQ v1.2.2 (IOKit v2.0.108)

MelsecQ Setup Serial Ethernet Modem RAS

Network Number:
0

PC Number:
255

Standard maximum string length:
128

OK Cancel Apply

Aba MelsecQ

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

Opções disponíveis na aba MelsecQ

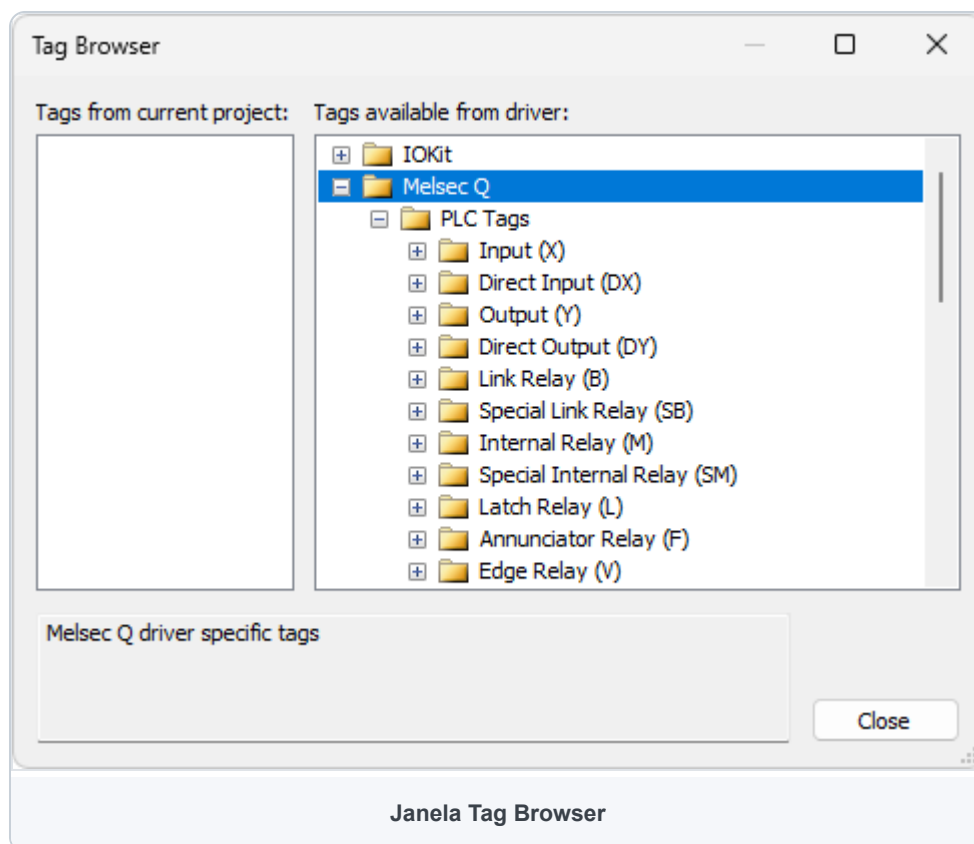
OPÇÃO	DESCRIÇÃO
Network Number	Número da rede de um dispositivo remoto, entre 0 (zero) e 255. Se a conexão é local, este valor deve ser igual a 0 (zero, padrão)
PC Number	Número de PC de um dispositivo remoto, entre 0 (zero) e 64, ou 255 para um número de PC local (padrão)
Standard maximum string length	Tamanho máximo padrão de uma variável do tipo String . O valor padrão é 128 e 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** na propriedade **Devicetem** no **E3**, no **Power** ou no **Water** ou usando parâmetros **[N/B]**.

Utilizando o Tag Browser

O **E3** a partir da versão 2.0 e o **Power** ou **Water** possuem uma ferramenta chamada **Tag Browser**, mostrada na figura a seguir, que permite que este Driver auxilie o usuário na criação e configuração de Tags.



Na listagem à esquerda desta janela, **Tags from current project** (*Tags do projeto corrente*), são mostrados os Tags e pastas que existem no projeto corrente. A listagem à direita, **Tags available from Driver** (*Tags disponíveis no 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 MelsecQ contém os nós indicados a seguir:

- O nó **IOKit** mostra os Tags disponibilizados pela biblioteca **IOKit**, divididos 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 **Melsec Q** possui pastas classificando o tipo de Tags que contém, **PLC Tags** (*Tags PLC*) e **Block Tags** (*Tags Bloco*). Em cada uma das pastas há outras várias pastas separando estes Tags por Área, e estas Áreas, por sua vez, contém os nodos finais, que são os exemplos de Tags de cada Área. Existe um exemplo para cada tipo previsto. Arraste um dos Tags para a lista à esquerda, ou mesmo o nó de uma pasta inteira para ter todos os Tags ao mesmo tempo, e este Tag ou Tags já estão devidamente configurados para o funcionamento

A ferramenta **Tag Browser** para o Driver MITSUBISHI MelsecQ visa automatizar a configuração dos Tags, através de exemplos para cada Área e cada tipo de dados suportado. Evidentemente que na grande maioria dos casos é útil incluir através desta ferramenta apenas os Tags de Áreas e tipos de dados que interessarem a uma aplicação, necessitando de adequações manuais posteriores. O **Tag Browser** sugere o valor "00000" (zero) para o Endereço em uma Área, no campo **Item**, que muito provavelmente deve ser reconfigurado após a inclusão dos Tags. Se um Tag Bloco é selecionado, 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 Áreas
 - **<Endereço>** deve ser preenchido com o valor do endereço em uma 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, o padrão para uma Área é exibido

Nomes de Áreas

ÁREA	SIGNIFICADO
X	Input
DX	Direct input
Y	Output
DY	Direct output
B	Link relay
SB	Special link relay
M	Internal relay
SM	Special internal relay
L	Latch relay

ÁREA	SIGNIFICADO
F	Annunciator relay
V	Edge relay
S	Step relay
TS	Timer contact
TC	Timer coil
SS	Integrating timer contact
SC	Integrating timer coil
CS	Counter contact
CC	Counter coil
TN	Timer value
SN	Integrating timer value
CN	Counter value
D	Data register
SD	Special data register
W	Link register
SW	Special link register
R	File register
ZR	File register
Z	Index register
LTS	Long Timer Contact

ÁREA	SIGNIFICADO
LTC	Long Timer Coil
LTN	Long Timer Current Value
LSTS	Long Retentive Timer Contact
LSTC	Long Retentive Timer Coil
LSTN	Long Retentive Timer Current Value
LCS	Long Counter Contact
LCC	Long Counter Coil
LCN	Long Counter Current Value

NOTAS

- O tipo de dados padrão das Áreas **TN**, **SN**, **CN**, **D**, **W**, **SW**, **R**, **ZR**, **Z** e **LSTN** é um **Word** (16 bits sem sinal) e não aceita ser configurado como um tipo de dados **Bit**.
- O tipo de dados padrão das Áreas **LTN** e **LCN** é um **Double Word** (32 bits sem sinal) e não aceita ser configurado como um tipo de dados **Bit**.
- O tipo de dados padrão das demais Áreas é um **Bit**.

Tipos de Dados

TIPO DE DADOS	SIGNIFICADO
W ou WORD	Valor Word (16 bits sem sinal)
S, SH ou SHORT	Valor Short (16 bits com sinal)
DW ou DWORD	Valor Double Word (32 bits sem sinal)
L ou LONG	Valor Long (32 bits com sinal)
F ou FLOAT	Valor Floating Point (32 bits de ponto flutuante, IEEE 754)

TIPO DE DADOS	SIGNIFICADO
D ou DOUBLE	Valor Double Floating Point (64 bits de ponto flutuante de dupla precisão, IEEE 754)
BCD	Valor BCD (16 bits)
LBCD	Valor BCD Long (32 bits)
DATE	Valor de 32 bits
STR ou STRING	Sequência de caracteres String

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

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

NOTA

Apenas Tags PLC configurados para as Áreas **Data Register**, **Link Register** ou **File Register** suportam o tipo de dados **String**. Tags Bloco ou Tags configurados com Áreas diferentes não suportam o tipo de dados **String**.

Exemplos

1. **Leitura ou escrita de Input de endereço 1 (um) e tipo de dados básico Bit**: Campo **Item** preenchido com o valor "X1".
2. **Leitura ou escrita de Output de endereço 100 e tipo de dados configurado como Word**: Campo **Item** preenchido com o valor "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 e tipo de dados básico Word**: Campo **Item** preenchido com o valor "CN50". Preencher com o valor "CN50:W" produz o mesmo resultado.
4. **Leitura ou escrita de Data Register de endereço 1000 e tipo de dados configurado como Double Word**: Campo **Item** preenchido com o valor "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 e tipo de dados configurado como String de tamanho máximo 64**: Campo **Item** preenchido com o valor "D100:STR[64]".

Configuração por Parâmetros Numéricos

Esta seção contém informações sobre a configuração deste Driver por parâmetros numéricos.

Parâmetros [N/B]

Leitura ou Escrita

N1 ou B1	Não utilizado, deixe em 0 (zero)
N2 ou B2	Tipo de dados e Área. Este valor deve ser composto pela soma de um tipo de dados mais a Área. Se o valor de um tipo de dados é igual a 0 (zero), assume-se o tipo de dados básico daquela Área
N3 ou B3	Parâmetro de uso especial. Deixe em 0 (zero) se o endereço na Área, ou <i>offset</i> , é menor do que 65535 e não é necessário informar o tamanho se um tipo de dados é igual a String . Caso contrário, consulte o tópico Usando o Parâmetro N3 ou B3
N4 ou B4	Valor do endereço em uma Área, ou <i>offset</i> , para valores até 65535. Para valores maiores, consulte o tópico Usando o Parâmetro N3 ou B3

Valores para Áreas

VALOR	SIGNIFICADO
0	Input
1	Direct input
2	Output
3	Direct output
4	Link relay
5	Special link relay
6	Internal relay
7	Special internal relay
8	Latch relay
9	Annunciator relay

VALOR	SIGNIFICADO
10	Edge relay
11	Step relay
12	Timer contact
13	Timer coil
14	Integrating timer contact
15	Integrating timer coil
16	Counter contact
17	Counter coil
18	Timer value
19	Integrating timer value
20	Counter value
21	Data register
22	Special data register
23	Link register
24	Special link register
25	File register
26	File register
27	Index register
28	Long Timer Contact
29	Long Timer Coil

VALOR	SIGNIFICADO
30	Long Timer Current Value
31	Long Retentive Timer Contact
32	Long Retentive Timer Coil
33	Long Retentive Timer Current Value
34	Long Counter Contact
35	Long Counter Coil
36	Long Counter Current Value

NOTAS

- O tipo de dados padrão das Áreas **18, 19, 20, 21, 23, 24, 25, 26, 27 e 33** é um **Word** (16 bits sem sinal) e não aceita ser configurado como um tipo de dados **Bit**.
- O tipo de dados padrão das Áreas **30 e 36** é um **Double Word** (32 bits sem sinal) e não aceita ser configurado como um tipo de dados **Bit**.
- O tipo de dados padrão das demais Áreas é um **Bit**.

Valores para Tipos de Dados

VALOR	SIGNIFICADO
0	Tipo de dados básico da área (Bit ou Word)
100	Bit (valor 1 ou 0)
200	Valor Word (16 bits sem sinal)
300	Valor Short (16 bits com sinal)
400	Valor Double Word (32 bits sem sinal)
500	Valor Long (32 bits com sinal)

VALOR	SIGNIFICADO
600	Valor Floating Point (32 bits de ponto flutuante, IEEE 754)
700	Valor Double Floating Point (64 bits de ponto flutuante de dupla precisão, IEEE 754)
800	Valor BCD (16 bits)
900	Valor BCD Long (32 bits)
1000	Valor de data em 32 bits
1100	Sequência de caracteres String

NOTA

Apenas Tags PLC configurados para as Áreas **Data Register**, **Link Register** ou **File Register** suportam o tipo de dados **String**. Tags Bloco ou Tags configurados com Áreas diferentes não suportam o tipo de dados **String**.

Exemplos

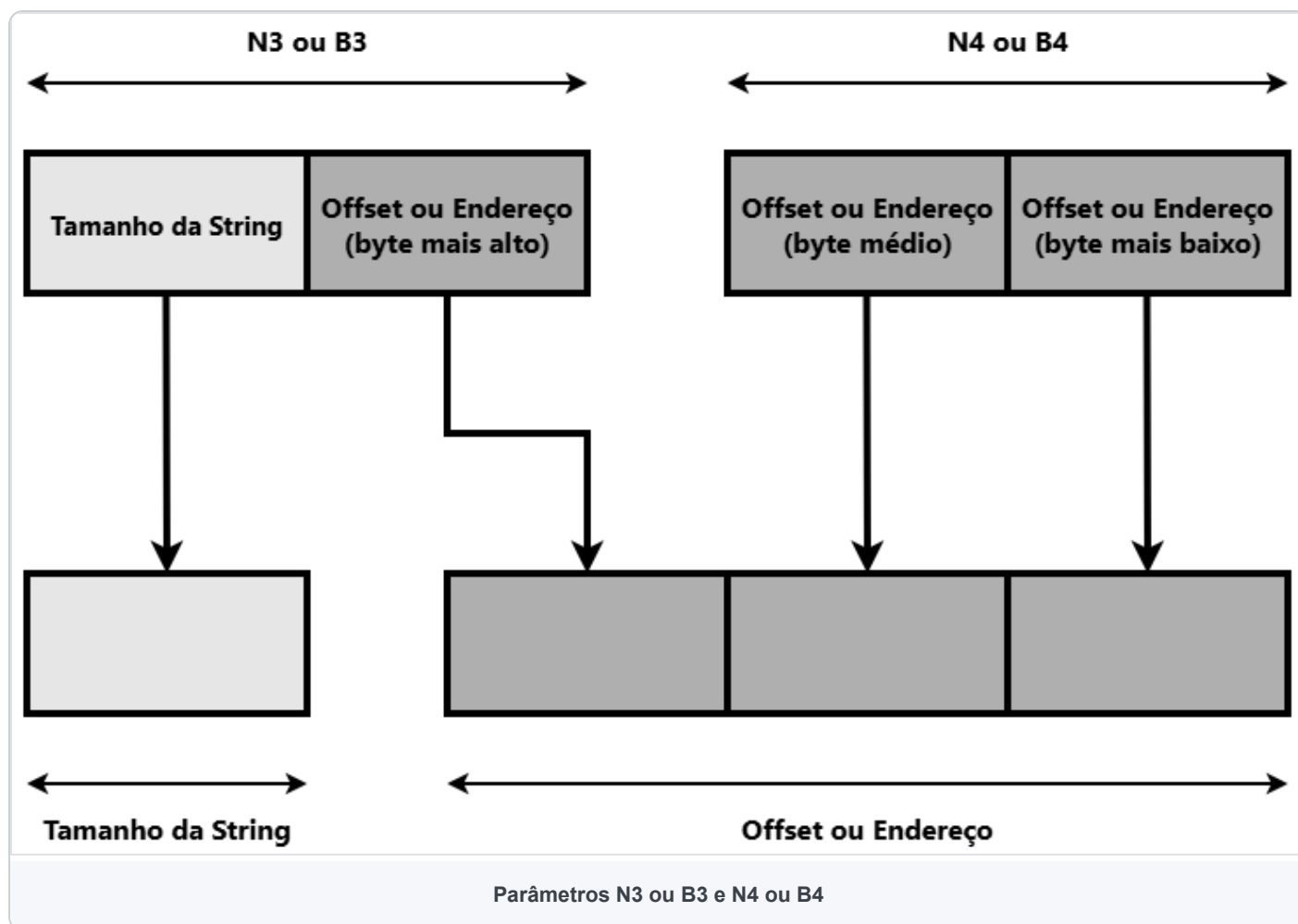
1. **Leitura ou escrita de Input de endereço 1 (um) e tipo de dados padrão Bit:** Parâmetro *N2* igual a 0 (zero), parâmetro *N3* igual a 0 (zero) e parâmetro *N4* igual a 1 (um). Se o parâmetro *N2* é igual a "100", produz o mesmo resultado.
2. **Leitura ou escrita de Output de endereço 100 e tipo de dados configurado como Word:** Parâmetro *N2* igual a 202, parâmetro *N3* igual a 0 (zero) e parâmetro *N4* igual a 100.
3. **Leitura ou escrita de Counter Value de endereço 50 e tipo de dados padrão Word:** Parâmetro *N2* igual a 20, parâmetro *N3* igual a 0 (zero) e parâmetro *N4* igual a 50. Se o parâmetro *N2* é igual a "220", produz o mesmo resultado.
4. **Leitura ou escrita de Data Register de endereço 1000 e tipo de dados configurado como Double Word:** Parâmetro *N2* igual a 421, parâmetro *N3* igual a 0 (zero) e parâmetro *N4* igual a 1000. O valor de 2 (dois) **Data Registers** contíguos a partir do endereço 1000 é reinterpretado como um **Double Word**.

Usando o Parâmetro N3 ou B3

Todos os parâmetros numéricos *N* e *B* têm capacidade de armazenar um número entre 0 (zero) e 65535. Quando o endereço em uma Área, ou *offset*, excede o valor máximo no parâmetro *N4* ou *B4*, o parâmetro *N3* ou *B3* deve ser utilizado para complementar a parcela excedente.

Da mesma forma, quando se utiliza um tipo de dados **String** e é necessário informar um tamanho máximo, o parâmetro *N3* ou *B3* deve ser utilizado para complementar a parcela excedente.

Assim, os parâmetros *N3* ou *B3* e *N4* ou *B4* devem ser imaginados como compostos por 2 (dois) bytes. Para este Driver, a parte mais alta do parâmetro *N3* ou *B3* é o **Tamanho da String**, e a parte mais baixa é o byte mais alto do valor de **Endereço**, ou *offset*, conforme a figura a seguir.



Portanto, o valor de **Endereço**, ou *offset*, é composto por um valor numérico de 24 bits, com o parâmetro **N4 ou B4** inteiro mais a metade mais baixa do parâmetro **N3 ou B3**. Se o valor de **Endereço** ou *offset* é inferior a 65535, a metade mais baixa do parâmetro **N3 ou B3** deve ser configurado com o valor 0 (zero). Do contrário, este valor deve ser decomposto em 3 (três) bytes, os 2 (dois) bytes mais baixos são o valor a ser preenchido no parâmetro **N4 ou B4** e o byte mais alto é o valor a ser preenchido na metade mais baixa do parâmetro **N3 ou B3**.

O valor de **Tamanho da String** é composto por um valor numérico de 8 (oito) bits, conforme explicado anteriormente, na parte mais alta do parâmetro **N3 ou B3**. Se o tipo de dados configurado não é **String** ou se a intenção é usar o tamanho máximo padrão de uma **String** como configurado na janela de configuração deste Driver, o preenchimento desta parte do parâmetro pode ser desconsiderada e deixada em 0 (zero).

Exemplos

1. **Endereço com valor 100000**: $100000 = 186A0h \gg 01\ 86\ A0 \gg N3\ ou\ B3 = 1; N4\ ou\ B4 = 86A0h = 34464$.
2. **String com tamanho 100 e Endereço inferior a 65535**: $100 = 64h \gg N3\ ou\ B3 = 6400h = 25600; N4\ ou\ B4 = \text{Endereço}$.
3. **String com tamanho 100 e Endereço 100000**: $100 = 64h; 100000 = 0186A0h \gg N3\ ou\ B3 = 6401h = 25601; N4\ ou\ B4 = 34464$.

Operações com Blocos

Para um bom desempenho da comunicação de dados, recomenda-se 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 Tag Bloco ocorre uma única operação para todos os Tags.

Para isto, considere que o endereço indicado nos parâmetros é o endereço inicial e cada Elemento de Bloco 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 **Bit** e é configurado para ser mostrado como **Bit**, cada Elemento de Bloco representa um bit com incremento de posição de um em um. Por exemplo, para **Y100** em um Tag Bloco com 3 (três) Elementos de Bloco, o primeiro Elemento de Bloco é **Y100**, o segundo Elemento de Bloco é **Y101** e o terceiro Elemento de Bloco é **Y102**.

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

Documentação das Interfaces de Comunicação

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

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
1.2.3	10/08/2026	M. Ludwig	• Driver atualizado para a biblioteca IOKit versão 3 e Visual Studio 2022 (<i>Case 37933</i>).
1.2.2	06/09/2019	M. Ludwig	• Driver portado para o Visual Studio 2017 (<i>Case 27086</i>).
1.2.1	23/01/2017	A. Hertzog	• Criados os parâmetros LTS , LTC , LTN , LSTS , LSTC , LSTN , LCS , LCC e LCN (<i>Case 21001</i>).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
1.1.4	14/03/2016	M. Ludwig	• Corrigido um falso erro ao utilizar um sufixo de tipo de dados correto nos parâmetros sintáticos (<i>Case 20613</i>).
1.1.3	08/07/2015	M. Ludwig	• Implementado suporte ao tipo de dados String (<i>Case 18884</i>).
1.1.2	14/11/2014	M. Ludwig	• Corrigido um GPF (<i>General Protection Failure</i>) na janela Tag Browser (<i>Case 17773</i>).
1.1.1	28/05/2014	M. Ludwig	• Corrigido o comportamento das opções Network Number e PC Number da janela de configurações, em que os valores salvos eram sempre os valores padrão (<i>Case 16150</i>).
1.0.1	19/07/2014	M. Ludwig	• Versão inicial deste Driver.