



MicrolokII

v1.0.10 · Português

Conteúdo

Driver ANSALDOSTS MicrolokII

Configuração do Driver

Configuração das Propriedades

Referência de Tags

Utilizando o Tag Browser

Leitura de Seções StationData

MessageInfo

HeaderField.UtcClockInitialization

HeaderField.UtcTimestampData

DataField.Data.ValueDataCounts

DataField.Data.ValueOfBooleanOutputs

Leitura e Escrita MyData

DataField.MyData.Read.ValueOfBooleanOutputs

DataField.MyData.Write.ValueOfBooleanOutputs

ConfirmNewValues

Documentação das Interfaces de Comunicação

Histórico de Revisões do Driver

Driver ANSALDOSTS MicrolokII

Nome do Arquivo	MicrolokII.dll
Fabricante	Ansaldo STS
Equipamentos	Microlok II
Protocolo	Peer
Versão	1.0.10
Última Atualização	05/09/2025
Plataforma	Win32
Dependências	N/A
Leitura com Superblocos	Não
Nível	31315

Introdução

Este Driver implementa o protocolo Peer, permitindo a uma aplicação desenvolvida pela **Elipse Software** comunicar com os equipamentos Microlok II da Ansaldo STS.

Configuração do Driver

Os parâmetros **[P]** de configuração deste Driver não são utilizados. Todas as configurações devem ser realizadas na caixa de diálogo de propriedades deste Driver. Para mais informações sobre as abas de configuração, consulte o tópico Documentação das Interfaces de Comunicação.

Configuração das Propriedades

A aba **MicrolokII** contém configurações específicas para este Driver.

Driver ANSALDOSTS MicrolokII v1.0.3 (IKit v2.0.136)

MicrolokII
Setup
Serial
Ethernet
Modem
RAS

Property	Value
☒ Peer protocol station parameters	
Session initiative	Wait for partner to initialize
Station type	Vital station
Acknowledgement timeout (ms)	24000
Heartbeat interval (ms)	24000
Indication update cycle	10
Stale data timeout (ms)	60000

Destination address:
Boolean output count:
Boolean input count:

Add

Source address:

Update
Remove

Destination address	Source address	Boolean output count	Boolea

OK
Cancel
Apply

Aba MicrolokII

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

Opções disponíveis na aba MicrolokII

OPÇÃO	DESCRIÇÃO
Session initiative	Seleciona o modo de inicialização que este Driver tem na rede. Os valores possíveis para esta opção são Wait for partner to initialize : Permanece à espera da inicialização de sessão por outro equipamento ou Initialize proactively : Este Driver inicializa a sessão com outro equipamento. Selecionar uma destas opções é o comportamento a ser adotado por este Driver com todos os equipamentos da rede
Station type	Tipo de Estação que este Driver se apresenta para a rede. Os valores possíveis para esta opção são Vital station ou Non-vital station
Acknowledgment timeout (ms)	Esta opção define o tempo, em milissegundos, que uma Estação <i>peer</i> emissora aguarda depois de enviar uma mensagem que requer confirmação. O valor ideal para esta opção depende das latências médias e piores latências do <i>link</i> de comunicação física e do volume de tráfego de mensagens que o <i>link</i> físico transporta. Os valores suportados para esta opção variam entre 50 e 60000 milissegundos. Em muitos casos, pode ser necessário definir esta opção por tentativa e erro. Esta opção deve ser definida para o valor mais baixo, em que os tempos limite de reconhecimento são minimizados e não completamente eliminados. O valor padrão desta opção é o valor definido na opção Heartbeat interval (ms)
Heartbeat interval (ms)	Esta opção define a taxa em que uma Estação <i>peer</i> tenta se comunicar com um par quando nenhuma alteração está ocorrendo nos bits da saída serial. O intervalo de <i>heartbeat</i> deve ser definido como um valor que não seja maior que metade do valor da opção Stale data timeout (ms) . Normalmente, esta opção deve ser definida como um valor que não seja inferior à opção Acknowledgment timeout (ms) . Esta opção, para uma Estação <i>peer</i> , pode ser definida para valores entre 100 e 600000 milissegundos. O valor padrão desta opção é definido como 40% do valor da opção Stale data timeout (ms)
Indication update cycle	Esta opção é usada em conjunto com a opção Heartbeat interval (ms) para controlar a taxa em que uma Estação <i>peer</i> envia mensagens de atualização de indicação para um par quando não há alterações nas saídas seriais para esta Estação. Um ciclo de atualização com o valor 1 (um) permite que uma mensagem de dados de indicação completa seja enviada a cada intervalo de <i>heartbeat</i> . Um ciclo de atualização com o valor 2 (dois) permite que uma mensagem de dados de indicação completa seja enviada em todos os outros intervalos de <i>heartbeat</i> , e assim por diante. Nos intervalos de <i>heartbeat</i> para os quais nenhuma mensagem de dados de indicação é enviada, uma mensagem de status da Estação que não contém dados é enviada. O uso periódico da mensagem de status ao invés da mensagem de dados de indicação reduz o carregamento em um canal de comunicação muito carregado, reduzindo o número de bytes transmitidos por este canal. Esta opção pode ser definida para valores entre 1 (um) e 100 e o valor padrão é 10

OPÇÃO	DESCRIÇÃO
Stale data timeout (ms)	Esta opção define o intervalo de tempo em que uma falha de comunicação é declarada quando uma Estação <i>peer</i> não recebe dados de indicação ou mensagens de status do par. Este intervalo de tempo, para uma Estação <i>peer</i> , é reinicializado para o valor especificado por esta opção cada vez que um dado de indicação ou mensagem de status é recebido da Estação <i>peer</i> . Se o temporizador de dados obsoleto de uma Estação <i>peer</i> de mesmo nível expirar, o status da Estação é definido para <i>falhado</i> e a Estação é definida em estado de inicialização. Uma vez declarado como <i>falhado</i> , o <i>link</i> de comunicação da Estação não é novamente declarado <i>ativo</i> até que a sequência de inicialização da comunicação tenha sido concluída e uma mensagem de dados válida tenha sido recebida. O tempo limite de dados obsoletos para uma Estação <i>peer</i> de mesmo nível pode ser definido para valores entre 500 e 600000 milissegundos. Para Estações não vitais, o valor padrão é 60000 ms

Separadamente, há parâmetros para o gerenciamento de conexão na camada de Sessão caso seja utilizada a camada física Ethernet.

Parâmetros para gerenciamento de conexão na camada de Sessão

OPÇÃO	DESCRIÇÃO
Performs an IP switch in case of timeout	Um valor True seleciona o comportamento de realizar uma troca pelo endereço IP de <i>backup</i> , ou vice-versa, após <i>n</i> expirações de tempo. O tempo de espera utilizado é igual ao valor da opção Acknowledgment timeout para o caso de este Driver realizar o papel de inicializador da Sessão e é igual ao valor da opção Stale data timeout quando este Driver está configurado com a opção Wait for partner to initialize . Um valor False não realiza troca pelo endereço IP de <i>backup</i> , ou vice-versa
IP switch after 'n' timeout(s)	Número de <i>time-outs</i> que devem ocorrer para que este Driver realize a troca pelo endereço IP de <i>backup</i> , ou vice-versa. O valor mínimo desta opção é 1 (um)
Wait time for new IP switch (ms)	Tempo de espera, em milissegundos, para permitir uma nova requisição de troca de endereço IP. Este tempo de espera impede que várias requisições vindas de múltiplas Sessões sejam chamadas em intervalos de tempo muito curtos. Os valores suportados para esta opção variam entre 0 e 60000 milissegundos. Esta configuração não tem impacto sobre os comandos dos Tags IO.Ethernet.IPSelect e IO.Ethernet.IPSwitch

Também é necessário configurar um cadastro para cada Estação de destino que se deseja comunicar. O cadastro de Estações é realizado em uma lista com os parâmetros informados na tabela a seguir.

Parâmetros para cadastro de Estações

PARÂMETRO	DESCRIÇÃO
Destination address	Endereço da Estação <i>peer</i> de destino
Source address	Endereço da Estação <i>peer</i> que este Driver usa para se identificar com um destinatário
Boolean output count	Número de saídas Booleanas. Esta opção refere-se à informação que este Driver comunica sobre si próprio na inicialização e deve coincidir com o valor esperado pelas Estações <i>peer</i>
Boolean input count	Número de entradas Booleanas. Esta opção refere-se à informação que a respectiva Estação <i>peer</i> comunica sobre si própria na inicialização e deve coincidir com este valor

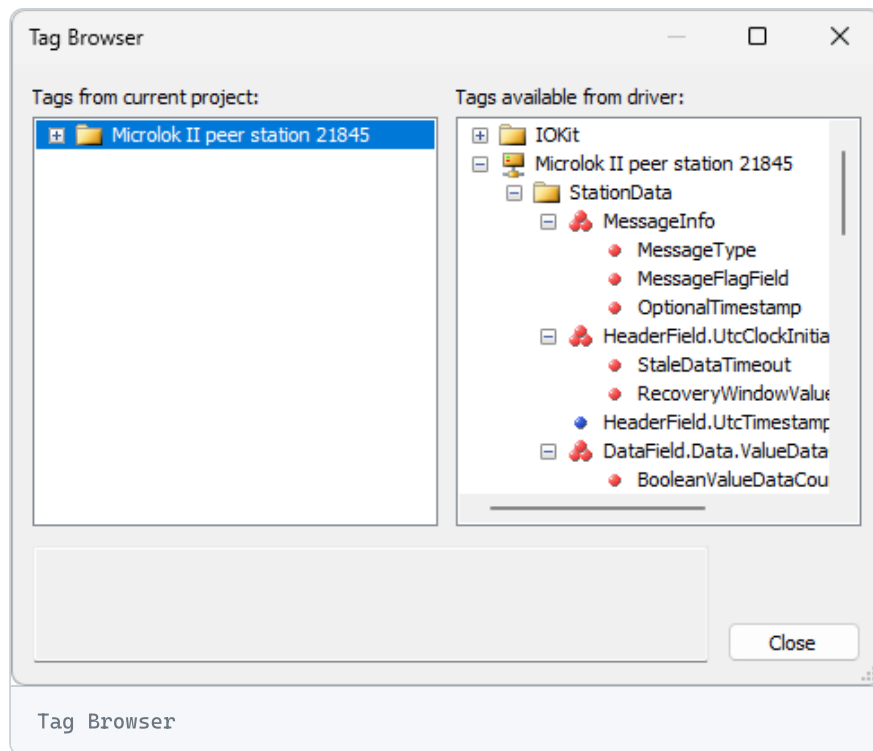
Utilize as opções **Add**, **Update** e **Remove** para adicionar parâmetros, alterar parâmetros já listados ou remover totalmente uma linha de parâmetros, respectivamente.

Referência de Tags

Esta seção contém informações sobre a configuração dos Tags deste Driver.

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 um Driver auxilie o usuário na criação e configuração de Tags.



A listagem **Tags do projeto corrente** (*Tags from current project*) mostra os Tags e pastas que existem no projeto corrente. A listagem **Tags disponibilizados pelo driver** (*Tags available from driver*) mostra uma árvore com os Tags disponíveis em um Driver. Para criar um Tag novo em uma aplicação, arraste um dos Tags definidos pelo Driver para a pasta desejada no diretório corrente. O Driver ANSALDOSTS MicrolokII contém os seguintes nós nesta árvore:

- O nó **IOKit** mostra os Tags disponíveis na 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 **Microlok II peer station #** possui as pastas **StationData**, **MyData** e **Control**, com os respectivos Tags. Em cada uma destas pastas há Tags Bloco ou PLC. Há um nó destes para cada Estação *peer* cadastrada na janela de propriedades deste Driver. Para mais informações, consulte o tópico Configuração das Propriedades

O Tag Browser para o Driver ANSALDOSTS MicrolokII visa automatizar a configuração dos Tags, auxiliando a carga e configuração de Tags disponíveis para um equipamento, e podem ser carregados tantos equipamentos quantos sejam necessários a um projeto, de forma que cada um destes equipamentos seja configurado pelo respectivo valor de endereço de destino.

Leitura de Seções StationData

Somente Leitura

Dispositivo	Endereço de destino, no formato <Endereço>:
Item	Nome da Seção, no formato <Nome da seção>

A pasta **StationData** possui um conjunto de Tags cuja origem das informações são as mensagens de dados recebidas da Estação *peer* endereçada. Por isto, estas mensagens são divididas em Seções para cada um dos Tags se encarregar de exibir a respectiva parte das informações. Todas as mensagens de dados vitais ou não vitais, assim que recebidas, são imediatamente disponibilizadas nestes Tags. As Seções `HeaderField.UtcClockInitialization` e `HeaderField.UtcTimestampData` somente existem em mensagens que possuem este campo de cabeçalho. Caso estas Seções não existam nas mensagens, são exibidas vazias.

NOTA

Dada a característica deste Driver de mostrar os valores assim que são recebidos, os tempos de varredura em todos estes Tags são desconsiderados. Desta forma, recomenda-se configurar a propriedade **AdviseType** de cada um dos Tags da pasta **StationData** com o valor 0 (zero, **AlwaysInAdvise**).

Se preferir omitir o campo **Dispositivo**, a sintaxe a seguir é a esperada para o campo **Item**:

<Endereço>:<Nome da Seção>

Driver Listen 43690-21845

Name	Device	Item	P1/...	P2/...	P3/...	P4/...	Size/Index
Driver Listen 43690-21845			0	0	0	0	
Microlok II peer station 21845	21845:						
StationData	21845:						
MessageInfo	21845:	MessageInfo	0	0	0	0	3
HeaderField.UtcClockInitialization	21845:	HeaderField.UtcClockInitialization	0	0	0	0	2
HeaderField.UtcTimestampData	21845:	HeaderField.UtcTimestampData	0	0	0	0	
DataField.Data.ValueDataCounts	21845:	DataField.Data.ValueDataCounts	0	0	0	0	2
DataField.Data.ValueOfBooleanOutputs	21845:	DataField.Data.ValueOfBooleanOutputs	0	0	0	0	16
MyData	21845:						
DataField.MyData.Read.ValueOfBooleanOutputs	21845:	DataField.MyData.Read.ValueOfBooleanOutputs	0	0	0	0	16
DataField.MyData.Write.ValueOfBooleanOutputs	21845:	DataField.MyData.Write.ValueOfBooleanOutputs	0	0	0	0	16
Control	21845:						
ConfirmNewValues	21845:	ConfirmNewValues	0	0	0	0	

Design

Scripts

Exemplo de Seções

As Seções disponíveis estão descritas nos tópicos a seguir.

MessageInfo

Elementos do Tag Bloco MessageInfo

ELEMENTO	NOME	SIGNIFICADO
1	MessageType	Tipo de mensagem. Os valores possíveis para este Elemento são 1 : Inicialização, 2 : Dados vitais ou 3 : Dados não vitais
2	MessageFlagField	Campo de <i>flags</i> no formato Word . Os valores possíveis para este Elemento são Bit 0 : Mensagem <i>Ack</i> , Bit 1 : Mensagem <i>NAck</i> , Bit 2 : <i>Ack</i> requerido, Bit 3 : <i>Link</i> estabelecido, Bit 4 : <i>Broadcast</i> , Bit 5 : Mensagem com estampa de tempo, Bit 6 : <i>Reset</i> de estampa de tempo, Bit 7 : Requerimento de inicialização e Bit 8 : Informação de cabeçalho opcional
3	OptionalTimestamp	Estampa de tempo da mensagem, no formato Date

NOTA

Utilize as propriedades de bits de 0 (zero) a 8 (oito) no Elemento **MessageFlagField** para acessar individualmente cada *flag*.

HeaderField.UtcClockInitialization

Elementos do Tag Bloco HeaderField.UtcClockInitialization

ELEMENTO	NOME	SIGNIFICADO
1	StaleDataTimeout	DWord com o valor de tempo limite de dados obsoletos
2	RecoveryWindowValue	DWord com o valor de janela de recuperação

HeaderField.UtcTimestampData

Contém um valor com o tipo de dados **DWord** contendo a estampa de tempo no formato **UTC** (*Coordinated Universal Time*).

DataField.Data.ValueDataCounts

Elementos do Tag Bloco DataField.Data.ValueDataCounts

ELEMENTO	NOME	SIGNIFICADO
1	BooleanValueDataCount	Número de dados de valores Booleanos
2	NumericValueDataCount	Número de dados de valores numéricos

NOTA

O valor do Elemento **BooleanValueDataCount** dividido por 8 (oito) determina o número de Elementos do Tag Bloco DataField.Data.ValueOfBooleanOutputs.

DataField.Data.ValueOfBooleanOutputs

Elementos do Tag Bloco DataField.Data.ValueOfBooleanOutputs

ELEMENTO	NOME	SIGNIFICADO
1	BooleanOutputValue0	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 0 (zero) até 7 (sete)
2	BooleanOutputValue1	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 8 (oito) até 15
3	BooleanOutputValue2	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 16 até 23
4	BooleanOutputValue3	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 24 até 31
5	BooleanOutputValue4	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 32 até 39
6	BooleanOutputValue5	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 40 até 47
7	BooleanOutputValue6	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 48 até 55
8	BooleanOutputValue7	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 56 até 63
9	BooleanOutputValue8	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 64 até 71
10	BooleanOutputValue9	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 72 até 79
11	BooleanOutputValue10	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 80 até 87
12	BooleanOutputValue11	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 88 até 95
13	BooleanOutputValue12	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 96 até 103
14	BooleanOutputValue13	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 104 até 111
15	BooleanOutputValue14	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 112 até 119

ELEMENTO	NOME	SIGNIFICADO
16	BooleanOutputValue15	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 120 até 127

NOTAS

- Utilize as propriedades de bits de 0 (zero) a 7 (sete) para acessar individualmente cada saída Booleana, no respectivo Elemento deste Tag Bloco.
- O número de saídas Booleanas pode variar de 0 (zero) a 128, assim este Tag Bloco pode variar de tamanho útil de 1 (um) a 16 Elementos, em que cada Elemento comporta de 8 (oito) a 128 saídas. Portanto, pode-se alterar o tamanho do Tag Bloco e remover Elementos para adequar-se ao tamanho verdadeiro.
- Se o número de saídas é igual a 0 (zero), este Tag Bloco é vazio.

Leitura e Escrita MyData

A pasta **MyData** contém os Tags Bloco a seguir, dedicados às saídas Booleanas que este Driver deve informar como Estação *peer*.

- `DataField.MyData.Read.ValueOfBooleanOutputs`: para a leitura dos valores que estão sendo enviados atualmente
- `DataField.MyData.Write.ValueOfBooleanOutputs`: para a escrita dos valores que são atualizados no envio de dados vitais ou não vitais após a confirmação

Enquanto não há escrita de valores pelo Tag Bloco `DataField.MyData.Write.ValueOfBooleanOutputs` e após a confirmação pela escrita do Tag `ConfirmNewValues`, este Driver envia dados vitais ou não vitais para a respectiva Estação *peer* endereçada com todos os valores em 0 (zero).

A escrita do Tag Bloco `DataField.MyData.Write.ValueOfBooleanOutputs` pode ser realizada por Bloco inteiro, por cada Elemento de Bloco separadamente, ou por cada bit de Elemento de Bloco separadamente, ou seja, com a propriedade **UseBitFields** configurada em Verdadeiro.

Desta forma, a sequência de ações deve ser a seguinte:

1. Escrever valores no Tag Bloco `DataField.MyData.Write.ValueOfBooleanOutputs` com uma ou mais operações de escrita.

2. Confirmar os valores com a escrita do Tag ConfirmNewValues.
3. Este Driver passa a considerar os valores confirmados para serem enviados ciclicamente à Estação *peer* endereçada.
4. Se necessário, os valores podem ser consultados via Tag Bloco `DataField.MyData.Read.ValueOfBooleanOutputs`.

DataField.MyData.Read.ValueOfBooleanOutputs

Somente Leitura

Dispositivo	Endereço de destino, no formato <Endereço>:
Item	DataField.MyData.Read.ValueOfBooleanOutputs

Se preferir omitir o campo **Dispositivo**, a sintaxe a seguir é a esperada para o campo **Item**:

```
<Endereço>:DataField.MyData.Read.ValueOfBooleanOutputs
```

Este Tag Bloco exibe todos os valores de saídas Booleanas enviados à Estação *peer* correspondente. Os valores iniciais de todos os Elementos são iguais a 0 (zero).

Elementos do Tag Bloco `DataField.MyData.Read.ValueOfBooleanOutputs`

ELEMENTO	NOME	SIGNIFICADO
1	BooleanOutputValue0	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 0 (zero) até 7 (sete)
2	BooleanOutputValue1	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 8 (oito) até 15
3	BooleanOutputValue2	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 16 até 23
4	BooleanOutputValue3	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 24 até 31
5	BooleanOutputValue4	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 32 até 39
6	BooleanOutputValue5	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 40 até 47
7	BooleanOutputValue6	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 48 até 55
8	BooleanOutputValue7	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 56 até 63
9	BooleanOutputValue8	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 64 até 71
10	BooleanOutputValue9	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 72 até 79
11	BooleanOutputValue10	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 80 até 87
12	BooleanOutputValue11	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 88 até 95
13	BooleanOutputValue12	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 96 até 103
14	BooleanOutputValue13	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 104 até 111
15	BooleanOutputValue14	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 112 até 119

ELEMENTO	NOME	SIGNIFICADO
16	BooleanOutputValue15	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 120 até 127

NOTAS

- Utilize as propriedades de bits de 0 (zero) a 7 (sete) para acessar individualmente cada saída Booleana, no respectivo Elemento deste Tag Bloco.
- O número de saídas Booleanas pode variar de 0 (zero) a 128, assim este Tag Bloco pode variar de tamanho útil de 1 (um) a 16 Elementos, em que cada Elemento comporta de 8 (oito) a 128 saídas. Portanto, pode-se alterar o tamanho do Tag Bloco e remover Elementos para adequar-se ao tamanho verdadeiro.
- Se o número de saídas é igual a 0 (zero), este Tag Bloco é vazio.

DataField.MyData.Write.ValueOfBooleanOutputs

Somente Escrita

Dispositivo	Endereço de destino, no formato <Endereço>:
Item	DataField.MyData.Write.ValueOfBooleanOutputs

Se preferir omitir o campo **Dispositivo**, a sintaxe a seguir é a esperada para o campo **Item**:

```
<Endereço>:DataField.MyData.Write.ValueOfBooleanOutputs
```

Inicialmente este Driver envia dados, vitais ou não, para a Estação *peer* correspondente endereçada com todos os valores zerados. Novos valores passam a ser enviados após a confirmação do Tag ConfirmNewValues com a escrita prévia de valores pelo Tag Bloco **DataField.MyData.Write.ValueOfBooleanOutputs**.

Elementos do Tag Bloco DataField.MyData.Write.ValueOfBooleanOutputs

ELEMENTO	NOME	SIGNIFICADO
1	BooleanOutputValue0	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 0 (zero) até 7 (sete)
2	BooleanOutputValue1	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 8 (oito) até 15
3	BooleanOutputValue2	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 16 até 23
4	BooleanOutputValue3	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 24 até 31
5	BooleanOutputValue4	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 32 até 39
6	BooleanOutputValue5	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 40 até 47
7	BooleanOutputValue6	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 48 até 55
8	BooleanOutputValue7	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 56 até 63
9	BooleanOutputValue8	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 64 até 71
10	BooleanOutputValue9	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 72 até 79
11	BooleanOutputValue10	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 80 até 87
12	BooleanOutputValue11	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 88 até 95
13	BooleanOutputValue12	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 96 até 103
14	BooleanOutputValue13	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 104 até 111
15	BooleanOutputValue14	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 112 até 119

ELEMENTO	NOME	SIGNIFICADO
16	BooleanOutputValue15	Elemento de tipo de dados Byte contendo saídas Booleanas, da posição 120 até 127

NOTAS

- É possível utilizar as propriedades de bits de 0 (zero) a 7 (sete) para escrever individualmente cada saída Booleana, no respectivo Elemento deste Tag Bloco.
- O número de saídas Booleanas pode variar de 0 (zero) a 128, assim este Tag Bloco pode variar de tamanho útil de 1 (um) a 16 Elementos, em que cada Elemento comporta de 8 (oito) a 128 saídas. Portanto, pode-se alterar o tamanho do Tag Bloco e remover Elementos para adequar-se ao tamanho verdadeiro.
- Se o número de saídas é igual a 0 (zero), este Tag Bloco é vazio.

ConfirmNewValues

Somente Escrita

Dispositivo	Endereço de destino, no formato <Endereço> :
Item	ConfirmNewValues

Se preferir omitir o campo **Dispositivo**, a sintaxe a seguir é a esperada para o campo **Item**:

```
<Endereço>:ConfirmNewValues
```

Utilize a escrita de qualquer valor neste Tag PLC para confirmar os novos valores escritos em `DataField.MyData.Write.ValueOfBooleanOutputs`. Ao escrever neste Tag PLC, os valores passam a ser utilizados para enviar à Estação *peer* correspondente, e assim o conteúdo do Tag Bloco `DataField.MyData.Read.ValueOfBooleanOutputs` passa a exibir os valores escritos. Esta escrita também provoca o envio imediato dos novos dados para o dispositivo de destino, antecipando o próximo *heartbeat*.

Documentação das Interfaces de Comunicação

Esta seção contém a documentação das Interfaces de Comunicação referente ao Driver **MicrolokII**.

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
1.0.10	05/09/2025	M. Ludwig	• Driver atualizado para a biblioteca IOKit versão 3 e Visual Studio 2022 (<i>Case 37937</i>).
1.0.8	09/09/2024	M. Ludwig	• Implementado um mecanismo de <i>debouncing</i> com tempo de espera após a primeira solicitação de troca de endereço IP, a fim de evitar filas de requisições de todas as Sessões ativas (<i>Case 36577</i>).
1.0.7	18/04/2024	M. Ludwig	• Adicionado o gerenciamento de conexão na camada Session (<i>Case 35834</i>).
1.0.6	08/02/2024	M. Ludwig	• Implementada a atualização de Tags para um estado de falha em caso de Sessão <i>offline</i> (<i>Case 35510</i>).
1.0.5	05/12/2023	M. Ludwig	• Corrigido o envio de dados no primeiro intervalo de <i>heartbeat</i> (<i>Case 35036</i>). • Corrigido o incremento de SNDMSN e RCVMSN (<i>Case 35036</i>). • Implementado o tratamento de colisões em esperas de <i>Ack</i> (<i>Case 35036</i>). • Corrigida a validação do código CRC em mensagens vitais (<i>Case 35036</i>). • Corrigida a leitura de Tags da biblioteca IOKit (<i>Case 35036</i>). • Corrigida a operação de escrita de bits de Elementos de Bloco (<i>Case 35036</i>).
1.0.4	11/09/2023	M. Ludwig	• Corrigida uma função deste Driver que poderia causar instabilidade durante a recepção de dados (<i>Case 34827</i>).
1.0.3	29/08/2023	M. Ludwig	• Realizadas melhorias de desempenho e otimizações no código-fonte (<i>Case 34743</i>).
1.0.2	24/07/2023	M. Ludwig	• Implementadas configurações para cada Estação de destino (<i>Case 34303</i>). • Implementado o envio de dados vitais e não vitais (<i>Case 34329</i>). • Implementada a leitura de Tags por <i>callbacks</i> (<i>Case 34449</i>).
1.0.1	24/04/2023	M. Ludwig	• Versão inicial deste Driver.