

Driver EXXOTEST DIIMuxIso

Nome do Arquivo	DIIMuxIso.dll
Fabricante	Annecy Electronique
Equipamentos	Interfaces USB/PCI Exxotest
Protocolo	ISO 14230
Versão	2.0.3
Última Atualização	10/08/2026
Plataforma	Win32
Dependências	IOKit versão 2.0, Muxdll.dll e USB Driver Kit
Leitura com Superblocos	Não
Nível	0

Introdução

Este é o Driver EXXOTEST DIIMuxIso para a comunicação entre aplicações da **Elipse Software** e interfaces USB/PCI EXXOtest. Este Driver estabelece comunicação com a biblioteca Muxdll do fabricante das interfaces.

Preparando o Equipamento

Um Driver de dispositivo USB Driver Kit, fornecido pelo fabricante, deve ser instalado no sistema operacional Windows. O arquivo Muxdll.dll, de biblioteca dinâmica fornecida pelo fabricante, deve estar localizado na mesma pasta do arquivo da biblioteca do Driver EXXOTEST DIIMuxIso, ou em um caminho conhecido (*Path*) do sistema operacional.

A interface EXXOTest deve ser fisicamente instalada em um computador compatível com o formato, USB ou PCI, e seguindo as instruções do fabricante.

Em testes realizados em laboratório, o Driver de dispositivo da interface EXXOTest apresentou compatibilidade com os sistemas operacionais Windows Vista e Windows 7, versões 32 e 64 bits, comunicando corretamente com o conjunto Elipse-interface-controlador.

Configuração do Driver

Esta seção contém informações sobre a configuração dos parâmetros **[P]** e dos parâmetros da aba DIIMuxIso na janela de propriedades deste Driver.

Este Driver não utiliza as interfaces de comunicação da biblioteca **IOKit**, portanto, ao incluir um objeto de Driver EXXOTEST DIIMuxIso em um projeto, este objeto já está configurado para não operar sobre nenhuma interface. Para mais informações sobre a aba **Setup**, consulte o tópico Documentação das Interfaces de Comunicação.

Parâmetros [P] de Configuração do Driver

P1	Número da placa ou interface EXXOtest a ser acessada
P2	Endereço local, ou seja, um endereço a ser atribuído como identificador local
P3	Não usado
P4	Não usado

Aba DIIMuxIso

A aba **DIIMuxIso** contém configurações específicas para este Driver. As opções disponíveis estão descritas na tabela a seguir.

Opções disponíveis na aba DIIMuxIso

PARÂMETRO	DESCRIÇÃO
Initialization type	Tipo de inicialização da linha de barramento ISO 14230. As opções disponíveis são 5 Baud Initialization ou Fast Initialization
W0	Tempo de ociosidade do barramento antes de enviar o <i>frame</i> de inicialização
W1	Tempo entre o fim do byte de endereçamento e o início do padrão de sincronização
W2	Tempo entre o fim do padrão de sincronização e o início do byte chave 1 (um)
W3	Tempo entre o byte chave 1 (um) e o byte chave 2 (dois)
W4	Tempo entre o byte chave 2 (dois) e o inverso, e entre este e o byte de endereçamento invertido
W5	Tempo entre duas inicializações
P0	Tempo entre a recepção do byte de endereçamento invertido e o início da emissão de um comando StartCommunication

PARÂMETRO	DESCRIÇÃO
TInitL	Tempo em nível baixo depois do tempo de ociosidade
TWup	Se não há recepção ou há erro durante a recepção de um comando StartCommunication , a ECU deve retornar em espera por um padrão de WakeUp antes do fim da temporização de Tidle
Tidle	Tempo mínimo esperado de inatividade na linha antes de enviar o primeiro padrão de WakeUp ou de repetir um novo padrão de WakeUp quando a ECU está fora de diagnóstico
P1	Espera entre bytes na resposta da ECU
P2	<i>Time-out</i> entre uma requisição e a resposta da ECU
P3	Espera entre a resposta da ECU e uma nova requisição deste Driver
P4	Espera entre bytes na requisição deste Driver

Todos os valores numéricos de tempo da tabela anterior estão em milissegundos. Os parâmetros **W[0-5]** e **P0** só são relevantes para a inicialização do tipo **5 Baud Initialization**, e os parâmetros **TInitL**, **TWup** e **Tidle** só são relevantes para a inicialização do tipo **Fast Initialization**.

Driver EXXOTEST DIIMuxIso v2.0.1 (IOKit v2.0.25)

DIIMuxIso
Setup
Serial
Ethernet
Modem
RAS

Initialization type:
5 Baud Initialization
Set to default...

Iso Bus Configuration (ms)

W0:	W4:	TInitL:	P1:
300	40	25	20
W1:	W5:	TWup:	P2:
100	300	50	40
W2:	P0:	TIdle:	P3:
10	100	300	55
W3:			P4:
10			5

OK
Cancel
Apply

Configuração para a inicialização 5 Baud

Driver EXXOTEST DIIMuxIso v2.0.1 (IOKit v2.0.25)

DIIMuxIso Setup Serial Ethernet Modem RAS

Initialization type:
Fast Initialization Set to default...

Iso Bus Configuration (ms)

W0: 300	W4: 40	TinitL: 25	P1: 20
W1: 100	W5: 300	TWup: 50	P2: 40
W2: 10	P0: 100	Tidle: 300	P3: 55
W3: 10			P4: 5

OK Cancel Apply

Configuração para a inicialização Fast

Clique em **Set to default** para retornar todos os valores ao padrão de temporização.

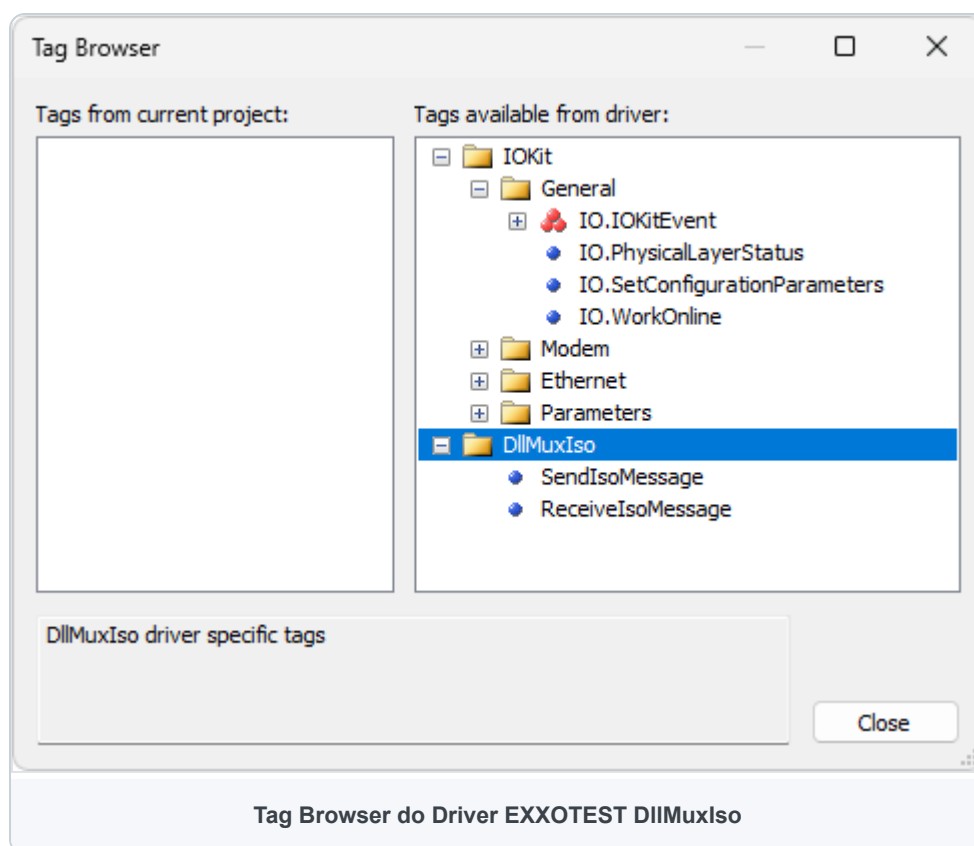
Referência de Tags

Esta seção contém informações sobre a configuração dos Tags deste Driver. Os Tags são referenciados a comandos através da configuração por código numérico do parâmetro *N1*. Os demais parâmetros *N* são irrelevantes na configuração dos Tags. Recomenda-se que estes sejam deixados em 0 (zero). Todos os Tags são do tipo **PLC**.

Os Tags deste Driver também podem ser referenciados por nomes sintáticos, ou seja, parâmetros do tipo **String**. O campo **Item** deve ser preenchido com um nome que se refere ao nome de um Tag, de acordo com a função. Os valores possíveis são DIIMuxIso.SendIsoMessage e DIIMuxIso.ReceiveIsoMessage.

Utilizando o Tag Browser

O **E3** 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 from current project** (*Tags do projeto corrente*) mostra os Tags e pastas que existem no projeto corrente. A lista **Tags available from driver** (*Tags disponibilizados pelo driver*) contém uma árvore com os Tags disponíveis no Driver EXXOTEST DIIMuxIso. Para criar um Tag em uma aplicação, arraste um Tag definido neste Driver para alguma pasta no projeto corrente. Os nós disponíveis para o Driver EXXOTEST DIIMuxIso nesta árvore são os seguintes:

- 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 **DIIMuxIso** lista os Tags definidos neste Driver. Arraste qualquer um destes Tags para a lista **Tags from current project**, ou a pasta para criar todos os Tags ao mesmo tempo, pois os Tags já estão devidamente configurados

Tag SendIsoMessage

Parâmetro N1 Igual a 1 (um, Somente Escrita) ou Parâmetro Sintático Item Preenchido com o Valor "DIIMuxIso.SendIsoMessage"

Utilize este Tag para enviar um quadro de dados (*frame*) à interface EXXOtest. A propriedade **Value** deste Tag é do tipo **String**, e devem ser colocados nesta propriedade os valores de bytes do *frame* a serem enviados. O formato destes bytes é hexadecimal.

Não é necessário escrever o cabeçalho e o código de verificação (*checksum*) do protocolo ISO 14230, pois estes valores são inseridos automaticamente. A seguir um exemplo de *frames* que podem ser enviados:

- **Valor do campo 81:** O valor hexadecimal 81h é escrito neste Tag, formando um conjunto de dados de apenas um byte. O *frame* formado automaticamente a partir deste byte é **81 10 F1 81 3**, em que os bytes colocados automaticamente estão em negrito
- **Valor do campo 17FF00:** O valor hexadecimal 17FF00h é escrito neste Tag, formando um conjunto de dados de três bytes. O *frame* formado automaticamente a partir destes três bytes é **83 10 F1 17 FF 00 9A**, em que os bytes colocados automaticamente estão em negrito

Tag ReceivelsoMessage

Parâmetro N1 Igual a 0 (zero, Somente Leitura) ou Parâmetro Sintático Item Preenchido com o Valor "DIIMuxIso.ReceivelsoMessage"

Utilize este Tag para verificar o recebimento de dados provenientes de uma interface. Caso haja dados, estes são representados na propriedade **Value** deste Tag no formato **String**.

Se não há dados, este Tag é deixado vazio, mantendo o último valor recebido. A leitura deste Tag pode ser executada repetidamente, com valores de varredura relativamente baixos, como por exemplo valores menores que 1000 ms, pois não depende da chegada de dados, nem possui entraves que consumam muito tempo na chamada.

Diferente do que ocorre no envio de dados, em que o cabeçalho e o código de verificação (*checksum*) não são necessários, o recebimento de dados conta com a apresentação de um *frame* completo, isto é, o valor do campo conta com o cabeçalho e o código de verificação (*checksum*) do protocolo ISO 14230. A interpretação dos bytes recebidos deve ser executada em um script na aplicação. A seguir um exemplo de um *frame* que pode ser recebido:

- **Valor do campo 83F110C1D08FA4:** Frame de sete bytes recebidos **83 F1 10 C1 D0 8F A4**, com valores em hexadecimal, em que os bytes que compõem o cabeçalho e o código de verificação (*checksum*) estão em negrito

NOTA

A propriedade **Value** sempre contém um número par de caracteres que representam bytes inteiros, mesmo que a metade mais significativa seja igual a 0 (zero) e mesmo que o conteúdo seja formado por zeros e apenas zeros.

Parâmetros de Configuração

Existe a possibilidade de se alterar os parâmetros configurados na aba **DIIMuxIso** em tempo de execução, com este Driver em modo **Offline**. Todas as configurações neste modo **Offline** utilizam a escrita de um Tag Bloco com 2 (dois) Elementos e os respectivos parâmetros conforme a tabela a seguir.

Somente Escrita

B1	-1 (menos um)
B2	0 (zero)
B3	0 (zero)

B4

3 (três)

Os itens a seguir mostram cada um dos parâmetros configuráveis desta forma. Para informações adicionais sobre o uso deste Tag, consulte o tópico Documentação das Interfaces de Comunicação.

Parâmetro DIIMuxIso.Initialization

Utilize os parâmetros a seguir para alterar o tipo de inicialização da interface:

- **Elemento 0:** DIIMuxIso.Initialization
- **Elemento 1:** Valor 0 (zero) para **Fast Initialization** e 1 (um) para **5 Baud Initialization**

Outros Parâmetros de Temporização DIIMuxIso

- **Elemento 0:** Sintaxe do parâmetro tempo, descrito a seguir
- **Elemento 1:** Valor de tempo, em milissegundos

Os parâmetros de temporização possíveis são DIIMuxIso.TInitL, DIIMuxIso.TWup, DIIMuxIso.TIdle, DIIMuxIso.P0, DIIMuxIso.P1, DIIMuxIso.P2, DIIMuxIso.P3, DIIMuxIso.P4, DIIMuxIso.W0, DIIMuxIso.W1, DIIMuxIso.W2, DIIMuxIso.W3, DIIMuxIso.W4 ou DIIMuxIso.W5.

Documentação das Interfaces de Comunicação

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

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
2.0.3	10/08/2026	M. Ludwig	• Driver atualizado para a biblioteca IOKit versão 3 e Visual Studio 2022 (Case 38054).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
2.0.1	26/07/2013	M. Ludwig	• Driver atualizado para a versão mais recente da biblioteca do fabricante (<i>Case 13725</i>). • Driver portado para a biblioteca IOKit versão 2 (<i>Case 14380</i>).
1.0.1	28/01/2009	M. Ludwig	• Primeira versão deste Driver (<i>Case 10097</i>).