PIXEL CMOS PROJECT

MIMOSA26 PROTOTYPE

Technical Documentation Version 0.2

JTAG SOFTWARE

COM INTERFACE

Ref Number: IPHC_CMOS_MIMOSA26_MULTI_DEV_JTAG_SW_COM_INF_1002 (100217)

Support:

Web address: http://www.iphc.cnrs.fr/-CMOS-ILC-.html

IPHC
CMOS Group
23 Rue du Loess
F-67037 Strasbourg Cedex

Written by: Kimmo JAASKELAINEN (kimmo.jaaskelainen@ires.in2p3.fr)

1. Important Information

Warranty:

The MIMOSA26 test board is warranted against defects in material and workmanship for a period of one year from the date of shipment, as evidence by receipts or other documentation. IPHC laboratory will, at its option, repair or replace equipment that proves to be defective during the warranty period. This warranty includes parts and labor.

IPHC laboratory believes that the information in this manual is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, IPHC reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult IPHC if errors are suspected. In no event shall IPHC be liable for any damages arising out of or related to this document or the information contained in it.

Expect as specified herein, IPHC laboratory makes no warranties, express or implied, and specifically disclaims any warranty of merchantability or fitness for a particular purpose. Customer's right to recover damages caused by fault or negligence on the part of IPHC laboratory shall be limited to amount therefore paid by the customer. IPHC laboratory will not be liable for damages resulting from loss of data, profits, use of products, or incidental, or consequential damages, even if advised of the possibility thereof. This limitation of the liability of IPHC laboratory will apply regardless of the form of action, whether in contract or tort, including action accrues. IPHC laboratory shall not be liable for any delay in performances due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures causes by owner's failure to follow the IPHC laboratory installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright:

Under the copyright laws, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, storing in an information.

2. Contents

3. About this manual

This is a short description of COM (Common Object Module) interface library for JTAG software of MIMOSA26 device. The MI26COM interface (MI26COMI) for the MIMOSA26 device is based on Microsoft's Component Object Model (COM). With MI26COMI you can control the MIMOSA26 JTAG software from a separate user program running on the same PC or on a remote PC.

4. Modifications Chronology

VERSION	MODIFICATIONS	CHAPTERS
0.1	Creation of the document.	All
0.2	Functions modified MasterConfReadBack and MasterConfReadBiasBack	All

5. Getting Started

This document is a short description for initiation of using COM interface library to interface to the JTAG software for a MIMOSA26 prototype test system. The system requirements are described in document MIMOSA26 JTAG SOFTWARE GETTING STARTED

6. Using the MIMOSA26 JTAG COM interface with Borland C++ Builder

1. Add “include” directive with a string “MI26LIB_TLB.h” to the start section of Application class header file (Application specific Form is inherited from TForm class.

```
ifndef MI26ClientUnitH
#define MI26ClientUnitH
//---------------------------------------------------------------------------
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <Dialogs.hpp>
#include "MI26LIB_TLB.h"
#include <Dialogs.hpp>
//---------------------------------------------------------------------------
class TFormMI26Client : public TForm
{
```

2. Add a reference “TCOMIMI26MasterConf MI26MasterConf “ to the private section of Application class described in the previous stage.

```
class TFormMI26Client : public TForm
{
__published:       // Composants gérés par l'EDI
      TEdit *Edit1;
      TButton *ButtonInfo;
....

private: // Déclarations de l'utilisateur
            TCOMIMI26MasterConf  MI26MasterConf;
public:              // Déclarations de l'utilisateur
      __fastcall TFormMI26Client(TComponent* Owner);
};
//---------------------------------------------------------------------------
extern PACKAGE TFormMI26Client *FormMI26Client;
//---------------------------------------------------------------------------
#endif
```

3. Initialize connection to the COM server with a method “Create” and the bounding can be tested with a command “get_info”. If the COM server application doesn’t show up, please re-register the MIMOSA26 COM server by executing the MI32.exe application on directory “C:\CCMOS_SCTRL\MIMOSA26_TAG”.

```
...
    if(!MI26MasterConf.IsBound()) // check if server is OK
    {
        OleCheck(CoMI26MasterConf::Create(MI26MasterConf));
    }
    if(MI26MasterConf.IsBound()) // perform request
    {
        WideString strValue;
        OleCheck(MI26MasterConf.get_Info(&strValue));
        Edit1->Text = strValue;
    }
...
```

4. The COM server request can be performing using a following request template. At the first stage server connection is checked. If the server connection is available the request to the server is performed.

```
    if(!MI26MasterConf.IsBound()) // check if server is OK
    {
        OleCheck(CoMI26MasterConf::Create(MI26MasterConf));
    }

    if(MI26MasterConf.IsBound()) // perform request
    {
        OleCheck(MI26MasterConf.<METHOD><ARGS>);
    }
```

In a following example is show how to use method “MasterConfUpdateAll” to update all the parameters to the device.

```
...
    if(!MI26MasterConf.IsBound()) // check if server is OK
    {
        OleCheck(CoMI26MasterConf::Create(MI26MasterConf));
    }

    if(MI26MasterConf.IsBound()) // perform request

    {
        WideString MsgStr;
        OleCheck(MI26MasterConf.MasterConfUpdateAll(&MsgStr));
        Memo1->Lines->Add(MsgStr);
    }
...
```

5. At the end of usage of the COM server application, the client application should release the server connection by using “Unbind” method.

```
void __fastcall TFormMi26MasterConfClient::FormClose(TObject *Sender,
    TCloseAction &Action)
{
   MI26MasterConf.Unbind();

}
```

7. Methods included to the MIMOSA26 JTAG COM object

7.1 IMI26MasterConf::Info

Description

This property allows client to retrieve a string with a name of server machine and a time information.

ODL syntax

[

propget,

id(0x00000001)

]

HRESULT _stdcall Info([out, retval] BSTR * Value);

Arguments

Value	Pointer to store value of info string.

HRESULT Return code

S_OK	The operation succeeded.

7.2 IMI26MasterConf::MasterConfLoadFile

Description

This method allows client to load a specific Master Configuration file.

ODL syntax

```
[
id(0x00000002)
]
HRESULT _stdcall MasterConfLoadFile([in] BSTR FileName, [out, retval] BSTR * Msg
);
```

Arguments

FileName	String with the full path to the Master Configuration.
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.3 IMI26MasterConf::MasterConfSaveFile

Description

This method allows client to save to the specific Master Configuration file.

ODL syntax

[

id(0x00000003)

]

HRESULT _stdcall MasterConfSaveFile([in] BSTR FileName, [out, retval] BSTR * Msg);

Arguments

FileName	String with the full path to the Master Configuration.
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.4 IMI26MasterConf::MasterConfUpdateAll

Description

This method allows client to update all the parameters to the device

ODL syntax

[

id(0x00000004)

]

HRESULT _stdcall MasterConfUpdateAll([out, retval] BSTR * Msg);

Arguments

Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.5 IMI26MasterConf::MasterConfUpdateBias

Description

This method allows client to update all the bias parameters to the device

ODL syntax

[

id(0x00000005)

]

HRESULT _stdcall MasterConfUpdateBias([out, retval] BSTR * Msg);

Arguments

Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.6 IMI26MasterConf::MasterConfUpdateLinePat

Description

This method allows client to update all the Line Pattern [0:1] parameters to the device

ODL syntax

[

id(0x00000014)

]

HRESULT _stdcall MasterConfUpdateLinePat([out, retval] BSTR * Msg);

Arguments

Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.7 IMI26MasterConf::MasterConfReadBack

Description

This method allows client to read back all the parameters to the device

ODL syntax

```
[
id(0x00000006)
]
HRESULT _stdcall MasterConfReadBack([out] long * Rb_Err, [out, retval] BSTR * Msg );
```

Arguments

Rb_Err	Pointer to store number of read back errors
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.8 IMI26MasterConf::MasterConfReadReset

Description

This method allows client to reset the target device.

ODL syntax

[

id(0x00000007)

]

HRESULT _stdcall MasterConfReset([out, retval] BSTR * Msg);

Arguments

Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.9 IMI26MasterConf::MasterConfReadStart

Description

This method allows client to initialize the device for running operation.

ODL syntax

```
[
id(0x00000008)
]
HRESULT _stdcall MasterConfStart([out, retval] BSTR * Msg );
```

Arguments

Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.10 IMI26MasterConf::MasterConfSetParallelPortAddr

Description

This method allows client to modify the base address of the parallel port device.

ODL syntax

[

id(0x00000009)

]

HRESULT _stdcall MasterConfSetParallelPortAddr([in] int Addr, [out] long * Rb_Addr, [out, retval] BSTR * Msg);

Arguments

Addr	Address of the parallel port device (e.g. 0x378)
Addr_Rb	Retrieved address of the parallel port device
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.11 IMI26MasterConf::MasterConfSetDevNum

Description

This method allows client to select a device number on multi device configuration.

ODL syntax

[

id(0x0000001C)

]

HRESULT _stdcall MasterConfSetDevNum([in] long DevNum, [out, retval] BSTR * Msg
);

Arguments

DevNum	Device number
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.12 IMI26MasterConf::MasterConfGetDevNum

Description

This method allows client to retrieve the current device number on multi device configuration.

ODL syntax

[

id(0x0000001D)

]

HRESULT _stdcall MasterConfGetDevNum([out] long * DevNum, [out, retval] BSTR * Msg);

Arguments

DevNum	Pointer to store the current device number
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.13 IMI26MasterConf::MIMOSA26ConfSetBias

Description

This method allows client to modify a value of Bias registers. Please see the MIMOSA26 User Manual Chapter 3.2.

ODL syntax

[

id(0x0000000A)

]

HRESULT _stdcall MIMOSA26ConfSetBias([in] long RegNum, [in] long RegValue, [out, retval] BSTR * Msg);

Arguments

RegNum	Bias Register number [0:19]
RegValue	Bias Register value (8-bit)
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.14 IMI26MasterConf::MIMOSA26ConfGetBias

Description

This method allows client to retrieve write and read values of Bias registers. Please see the MIMOSA26 User Manual Chapter 3.2.

ODL syntax

[

id(0x0000000B)

]

HRESULT _stdcall MIMOSA26ConfGetBias([in] long RegNum, [out] long * RegValue, [out] long * Rb_RegValue, [out, retval] BSTR * Msg);

Arguments

RegNum	Bias Register number [0:19]
RegValue	Bias Register value (write to device) (8-bit)
Rb_RegValue	Bias Register value (read from device) (8-bit)
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.15 IMI26MasterConf::MIMOSA26ConfSetRomode

Description

This method allows client to modify a value of RoMode registers. Please see the MIMOSA26 User Manual Chapter 2.3.13 and 2.3.14.

ODL syntax

[

id(0x0000000C)

]

HRESULT _stdcall MIMOSA26ConfSetRoMode([in] long RegNum, [in] long RegValue, [out, retval] BSTR * Msg);

Arguments

RegNum	Romode Register number [0:15] Register number 0 – 7: Romode Register 0 Register number 8 – 15: Romode Register 1
RegValue	Romode Register value (1-bit)
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.16 IMI26MasterConf::MIMOSA26ConfGetRoMode

Description

This method allows client to retrieve write/read values of RoMode register. Please see the MIMOSA26 User Manual Chapter 2.3.13 and 2.3.14.

ODL syntax

[

id(0x0000000D)

]

HRESULT _stdcall MIMOSA26ConfGetRoMode([in] long RegNum, [out] long * RegValue, [out, retval] BSTR * Msg);

Arguments

RegNum	Romode Register number [0:15]
	Register number 0 – 7: Romode Register 0
	Register number 8 – 15: Romode Register 1
RegValue	RoMode Register value (read/write to device) (1-bit)
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.17 IMI26MasterConf::MIMOSA26ConfSetDiscri

Description

This method allows client to modify a value of Discriminator register.

ODL syntax

[

id(0x0000000E)

]

HRESULT _stdcall MIMOSA26ConfSetDiscri([in] long RegNum, [in] long RegValue, [out, retval] BSTR * Msg);

Arguments

RegNum	Discriminator Register number [0:35]
RegValue	Discriminator Register value (32-bit)
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.18 IMI26MasterConf::MIMOSA26ConfGetDiscri

Description

This method allows client to retrieve write and read values of Discriminator registers.

ODL syntax

[

id(0x0000000F)

]

HRESULT _stdcall MIMOSA26ConfGetDiscri([in] long RegNum, [out] long * RegValue, [out] long * Rb_RegValue, [out, retval] BSTR * Msg);

Arguments

RegNum	Discriminator Register number [0:35]
RegValue	Discriminator Register value (write to device) (32-bit)
Rb_RegValue	Discriminator Register value (read from device) (32-bit)
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.19 IMI26MasterConf::MIMOSA26ConfSetLinePat0

Description

This method allows client to modify a value of Line Pattern 0 registers.

ODL syntax

[

id(0x00000010)

]

HRESULT _stdcall MIMOSA26ConfSetLinePat0([in] long RegNum, [in] long RegValue, [out, retval] BSTR * Msg);

Arguments

RegNum	Line Pattern 0 register number [0:35]
RegValue	Line Pattern 0 register value (32-bit)
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.20 IMI26MasterConf::MIMOSA26ConfGetLinePat0

Description

This method allows client to retrieve write and read values of Line Pattern 0 registers.

ODL syntax

[

id(0x00000011)

]

HRESULT _stdcall MIMOSA26ConfGetLinePat0([in] long RegNum, [out] long * RegValue, [out] long * Rb_RegValue, [out, retval] BSTR * Msg);

Arguments

RegNum	Line Pattern 0 Register number [0:35]
RegValue	Line Pattern 0 Register value (write to device) (32-bit)
Rb_RegValue	Line Pattern 0 Register value (read from device) (32-bit)
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.21 IMI26MasterConf::MIMOSA26ConfSetLinePat1

Description

This method allows client to modify a value of Line Pattern 1 registers.

ODL syntax

[

id(0x00000012)

]

HRESULT _stdcall MIMOSA26ConfSetLinePat1([in] long RegNum, [in] long RegValue, [in] BSTR * Msg);

Arguments

RegNum	Line Pattern 1 register number [0:35]
RegValue	Line Pattern 1 register value (32-bit)
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.22 IMI26MasterConf::MIMOSA26ConfGetLinePat1

Description

This method allows client to retrieve write and read values of Line Pattern 1 registers.

ODL syntax

[

id(0x00000013)

]

HRESULT _stdcall MIMOSA26ConfGetLinePat1([in] long RegNum, [out] long * RegValue, [out] long * Rb_RegValue, [out, retval] BSTR * Msg);

Arguments

RegNum	Line Pattern 1 Register number
RegValue	Line Pattern 1 Register value (write to device) (32-bit)
Rb_RegValue	Line Pattern 1 Register value (read from device) (32-bit)
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.23 IMI26MasterConf::MIMOSA26ConfShowWindow

Description

This method allows client to show or hide the device configuration window.

ODL syntax

[

id(0x00000015)

]

HRESULT _stdcall MIMOSA26ConfShowWindow([in] long ShowWin, [out, retval] BSTR * Msg);

Arguments

ShowWin	To show window (=1) and to hide window (=0)
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.24 IMI26MasterConf::MIMOSA26ConfSetCtrl

Description

This method allows client to modify values of Row Markers on Control Pixel register. Please see the MIMOSA26 User Manual Chapter 2.3.8 CONTROL_PIX_REG register.

ODL syntax

[

id(0x00000016)

]

HRESULT _stdcall MIMOSA26ConfSetCtrl([in] long RegNum, [in] long RegValue, [out, retval] BSTR * Msg);

Arguments

RegNum	Control register number [0:2]
	0: RowMka (10-bit)
	1: RowMkd (10-bit)
	2 : RowMkLast (10-bit)
RegValue	Control register value
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.25 IMI26MasterConf::MIMOSA26ConfGetCtrl

Description

This method allows client to retrieve write and read values of Row Markers on Control Pixel register. Please see the MIMOSA26 User Manual Chapter 2.3.8 CONTROL_PIX_REG register.

ODL syntax

[

id(0x00000017)

]

HRESULT _stdcall MIMOSA26ConfGetCtrl([in] long RegNum, [out] long * RegValue, [out] long * Rb_RegValue, [out, retval] BSTR * Msg);

Arguments

RegNum	Control register number [0:2]
	0: RowMka (10-bit)
	1: RowMkd (10-bit)
	2 : RowMkLast (10-bit)
RegValue	Control Register value (write to device)
Rb_RegValue	Control Register value (read from device)
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.26 IMI26MasterConf::MIMOSA26ConfSetTest1Pad

Description

This method allows client to modify a value of SelPad1 section on Control Pixel register. Please see the MIMOSA26 User Manual Chapter 2.3.8 CONTROL_PIX_REG register.

ODL syntax

[

id(0x00000018)

]

HRESULT _stdcall MIMOSA26ConfSetTest1Pad([in] long RegValue, [out, retval] BSTR * Msg);

Arguments

RegValue	SelPad1 selection value (3-bit)
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.27 IMI26MasterConf::MIMOSA26ConfGetTest1Pad

Description

This method allows client to retrieve write/read values of SelPad1 section Control Pixel registers. Please see the MIMOSA26 User Manual Chapter 2.3.8 CONTROL_PIX_REG register.

ODL syntax

[

id(0x00000019)

]

HRESULT _stdcall MIMOSA26ConfGetTest1Pad([out] long * RegValue, [out, retval] BSTR * Msg);

Arguments

RegValue	SelPad1 selection value (3-bit)
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.28 IMI26MasterConf::MIMOSA26ConfSetTest2Pad

Description

This method allows client to modify a value of SelPad2 section on Control Pixel register. Please see the MIMOSA26 User Manual Chapter 2.3.8 CONTROL_PIX_REG register.

ODL syntax

[

id(0x0000001A)

]

HRESULT _stdcall MIMOSA26ConfSetTest2Pad([in] long RegValue, [out, retval] BSTR * Msg);

Arguments

RegValue	SelPad2 selection value (3-bit)
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.29 IMI26MasterConf::MIMOSA26ConfGetTest2Pad

Description

This method allows client to retrieve write/read values of SelPad2 section on Control Pixel register. Please see the MIMOSA26 User Manual Chapter 2.3.8 CONTROL_PIX_REG register.

ODL syntax

[

id(0x0000001B)

]

HRESULT _stdcall MIMOSA26ConfGetTest2Pad([out] long * RegValue, [out, retval] BSTR * Msg);

Arguments

RegValue	SelPad2 value (3-bit)
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.30 IMI26MasterConf::Mimosa26ConfSetHeaderTrailer

Description

This method allows client to modify a value of Header Trailer register. Please see the MIMOSA26 User Manual Chapter 2.3.11 HEADER_REG register.

ODL syntax

[

id(0x0000001E)

]

HRESULT _stdcall Mimosa26ConfSetHeaderTrailer([in] long RegNum, [in] long RegValue, [out, retval] BSTR * Msg);

Arguments

RegNum	Header Trailer Register number [0:3]
	0: Trailer1
	1: Trailer0
	2: Header1
	3: Header0
RegValue	Header Trailer Register value (32-bit)
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.31 IMI26MasterConf:: Mimosa26ConfGetHeaderTrailer

Description

This method allows client to retrieve write and read values of Header Trailer register. Please see the MIMOSA26 User Manual Chapter 2.3.11 HEADER_REG register.

ODL syntax

[

id(0x0000001F)

]

HRESULT _stdcall Mimosa26ConfGetHeaderTrailer([in] long RegNum, [out] long * RegValue, [out] long * Rb_RegValue, [out, retval] BSTR * Msg);

Arguments

RegNum	Header Trailer Register number [0:3]
	0: Trailer1
	1: Trailer0
	2: Header1
	3: Header0
RegValue	Header Trailer Register value (write to device) (32-bit)
Rb_RegValue	Header Trailer Register value (read from device) (32-bit)
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.32 IMI26MasterConf::Mimosa26ConfSetSeqPix

Description

This method allows client to modify a value of Sequencer Pixel registers. Please see the MIMOSA26 User Manual Chapter 2.3.7 SEQUENCER_PIX_REG register.

ODL syntax

[

id(0x00000020)

]

HRESULT _stdcall Mimosa26ConfSetSeqPix([in] long RegNum, [in] long RegValue, [out, retval] BSTR * Msg);

Arguments

RegNum	Sequencer Pixel Register number [0:7]	
	0:	DataPwrOn0 (DataPwrOn[15:0])
	1:	DataPwrOn1 (DataPwrOn[31:16])
	2:	DataRdDsc
	3:	DataCalib
	4:	DataClp
	6:	DataRst1
	7:	DataRdPix
RegValue	Sequencer Pixel Register value (16-bit)	
Msg	Pointer to store server message	

HRESULT Return code

S_OK	The operation succeeded.

7.33 IMI26MasterConf::Mimosa26ConfGetSeqPix

Description

This method allows client to retrieve write and read values of Sequencer Pixel registers. Please see the MIMOSA26 User Manual Chapter 2.3.7 SEQUENCER_PIX_REG register.

ODL syntax

[

id(0x00000021)

]

HRESULT _stdcall Mimosa26ConfGetSeqPix([in] long RegNum, [out] long * RegValue, [out] long * Rb_RegValue, [out, retval] BSTR * Msg);

Arguments

RegNum	Sequencer Pixel Register number [0:7]
	0: DataPwrOn0 (DataPwrOn[15:0])
	1: DataPwrOn1 (DataPwrOn[31:16])
	2: DataRdDsc
	3: DataCalib
	4: DataClp
	6: DataRst1
	7: DataRdPix
RegValue	Sequencer Pixel Register value (write to device) (16-bit)
Rb_RegValue	Sequencer Pixel Register value (read from device) (16-bit)
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.34 IMI26MasterConf::Mimosa26ConfSetSeqSuze

Description

This method allows client to modify a value of Sequencer SUZE registers. Please see the MIMOSA26 User Manual Chapter 2.3.10 SEQUENCER_SUZE_REG register.

ODL syntax

[

id(0x00000022)

]

HRESULT _stdcall Mimosa26ConfSetSeqSuze([in] long RegNum, [in] long RegValue, [out, retval] BSTR * Msg);

Arguments

RegNum	Sequencer SUZE Register number [0:9]
RegValue	Sequencer SUZE Register value (16-bit)
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.35 IMI26MasterConf::Mimosa26ConfGetSeqSuze

Description

This method allows client to retrieve write and read values of Sequencer SUZE registers. Please see the MIMOSA26 User Manual Chapter 2.3.10 SEQUENCER_SUZE_REG register.

ODL syntax

[

id(0x00000023)

]

HRESULT _stdcall Mimosa26ConfGetSeqSuze([in] long RegNum, [out] long * RegValue, [out] long * Rb_RegValue, [out, retval] BSTR * Msg);

Arguments

RegNum	Sequencer SUZE Register number [0:9]
RegValue	Sequencer SUZE Register value (write to device) (16-bit)
Rb_RegValue	Sequencer SUZE Register value (read from device) (16-bit)
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.36 IMI26MasterConf::Mimosa26ConfSetCtrlSuze

Description

This method allows client to modify a value of Control SUZE registers. Please see the MIMOSA26 User Manual Chapter 2.3.12 CONTROL_SUZE_REG register.

ODL syntax

[

id(0x00000024)

]

HRESULT _stdcall Mimosa26ConfSetCtrlSuze([in] long RegNum, [in] long RegValue, [out, retval] BSTR * Msg);

Arguments

RegNum	Control SUZE Register number [0:15] 0: CFGADR (3-bit) 1: CFGCS (3-bit) 2: CFGDATA (3-bit) 3: CFGWR (3-bit) 4: DISCKGMODGATE (1-bit) 5: JSUPINITMEM (1-bit) 6: CLKRATEOUT (1-bit) 7: DUALCHANNELOUT (1-bit) 8: SCANLINETST (10-bit) 9: ROWLASTSUZE (10-bit) 10: ENTESTDATADISC (1-bit) 11: TESTAFTERMUX (1-bit) 12: ENSCAN (1-bit) 13: ENAUTOSCANDISCRI (1-bit) 14: SELPAD3 (3-bit) 15: SELPAD4 (3-bit)
RegValue	Control SUZE Register value
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.37 IMI26MasterConf:: Mimosa26ConfGetCtrlSuze

Description

This method allows client to retrieve write and read values of Control SUZE register. Please see the MIMOSA26 User Manual Chapter 2.3.12 CONTROL_SUZE_REG register.

ODL syntax

[

id(0x00000025)

]

HRESULT _stdcall Mimosa26ConfGetCtrlSuze([in] long RegNum, [out] long * RegValue, [out] long * Rb_RegValue, [out, retval] BSTR * Msg);

Arguments

RegNum — Control SUZE Register number [0:15]

0: CFGADR (3-bit)
1: CFGCS (3-bit)
2: CFGDATA (3-bit)
3: CFGWR (3-bit)
4: DISCKGMODGATE (1-bit)
5: JSUPINITMEM (1-bit)
6: CLKRATEOUT (1-bit)
7: DUALCHANNELOUT (1-bit)
8: SCANLINETST (10-bit)
9: ROWLASTSUZE (10-bit)
10: ENTESTDATADISC (1-bit)
11: TESTAFTERMUX (1-bit)
12: ENSCAN (1-bit)
13: ENAUTOSCANDISCRI (1-bit)
14: SELPAD3 (3-bit)
15: SELPAD4 (3-bit)

RegValue — Control SUZE Register value (write to device)

Rb_RegValue — Control SUZE Register value (read from device)

Msg — Pointer to store server message

HRESULT Return code

S_OK — The operation succeeded.

7.38 IMI26MasterConf:: Mimosa26ConfSet8B10Reg0

Description

This method allows client to modify a value of Control 8B10B Reg0 registers.

ODL syntax

[

id(0x00000026)

]

HRESULT _stdcall Mimosa26ConfSet8B10Reg0([in] long RegNum, [in] long RegValue, [out, retval] BSTR * Msg);

Arguments

RegNum	Control 8B10B Reg0 Register number [0:15]
RegValue	Control 8B10B Reg0 Register value (9-bit)
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.39 IMI26MasterConf::Mimosa26ConfGet8B10Reg0

Description

This method allows client to retrieve write and read values of Control 8B10B Reg0 registers.

ODL syntax

[

id(0x00000027)

]

HRESULT _stdcall Mimosa26ConfGet8B10Reg0([in] long RegNum, [out] long * RegValue, [out] long * Rb_RegValue, [out, retval] BSTR * Msg);

Arguments

RegNum	Control 8B10B Reg0 Register number [0:7]
RegValue	Control 8B10B Reg0 Register value (write to device) (9-bit)
Rb_RegValue	Control 8B10B Reg0 Register value (read from device) (9-bit)
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.40 IMI26MasterConf:: Mimosa26ConfGet8B10BReg1RawData

Description

This method allows client to retrieve read values of Control 8B10B Reg1 (Raw Data Field) registers.

ODL syntax

[

id(0x00000028)

]

HRESULT _stdcall Mimosa26ConfGet8B10BReg1RawData([in] long RegNum, [out] long * Rb_RegValue, [out, retval] BSTR * Msg);

Arguments

RegNum	Control 8B10B Reg1 (Raw Data) Register number [0:15]
Rb_RegValue	Control 8B10B Reg1 (Raw Data) Register value (read from device)
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.41 IMI26MasterConf::Mimosa26ConfGet8B10BReg1EncodedData

Description

This method allows client to retrieve read values of Control 8B10B Reg1 (Encoded Data Field) registers.

ODL syntax

[

id(0x00000029)

]

HRESULT _stdcall Mimosa26ConfGet8B10BReg1EncodedData([in] long RegNum, [out] long * Rb_RegValue, [out, retval] BSTR * Msg);

Arguments

RegNum	Control 8B10B Reg1 (Encoded Data) Register number [0:15]
Rb_RegValue	Control 8B10B Reg1 (Encoded Data) Register value (read from device)
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.42 IMI26MasterConf::Mimosa26ConfGet8B10BReg1Flags

Description

This method allows client to retrieve read values of Control 8B10B Reg1 (Flags Field) registers.

ODL syntax

```
[
  id(0x0000002A)
]
```

HRESULT _stdcall Mimosa26ConfGet8B10BReg1Flags([in] long RegNum, [out] long * Rb_RegValue, [out, retval] BSTR * Msg);

Arguments

RegNum	Control 8B10B Reg1 (Flags) Register number [0:4]
	0: Start, 1: DataRdy, 2: RstB, 3: RawDataRdy, 4: EncodedDataRdy
Rb_RegValue	Control 8B10B Reg1 (Flags) Register value (read from device)
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.43 IMI26MasterConf::MasterConfSaveAllFile

Description

This method allows client to save the parameters to the specific Master Configuration file and to the “current.mcf” file. Also all the Device Configuration files are saved.

ODL syntax

```
[
  id(0x0000002B)
  ]
HRESULT _stdcall MasterConfSaveAllFile([out, retval] BSTR * Msg );
```

Arguments

Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.

7.44 IMI26MasterConf::MasterConfReadBiasBack

Description

This method allows client to read back all the bias parameters from the device.

ODL syntax

[

id(0x0000002C)

]

HRESULT _stdcall MasterConfReadBiasBack([out] long * Rb_Err, [out, retval] BSTR * Msg);

Arguments

Rb_Err	Pointer to store number of read back errors
Msg	Pointer to store server message

HRESULT Return code

S_OK	The operation succeeded.