

## USER MANUAL AND SPECIFICATIONS

# MT1001

## Space Wire Adapter Module

### Overview

---

The MT1001 is Space Wire Adapter Module designed to work in conjunction with your NI FlexRIO™ FPGA module.

Space Wire NI FlexRIO Adapter Module is a communication network based in part on the IEEE 1355 standard of communications. It covers two (physical and data-link) of the seven layers of the OSI model for communications.

Space Wire data communication protocols enable the reliable sending of data at high-speed (between 5Mbps and 200Mbps) from one unit to another. Space Wire links are full-duplex point-to-point and serial data communication links.

The MT1001 features the following connectors and chips:

- Multi-channel programmable adapter 1892ХД1Я for packet transferring of data
- Space Wire MT1001 connector
- The following front panel connectors
  - CH0
  - CH1
  - CH2
  - CH3

This document contains input and output signal information of the MT1001 Space Wire Adapter and its technical specifications, matching/complying with technical characteristics of NI FlexRIO™ FPGA modules and MT1001 device.



**Note** The MT1001 refers to the combination of your MT1001 adapter module and your NI FlexRIO FPGA module. The MT1001 refers to your MT1001 adapter module only.



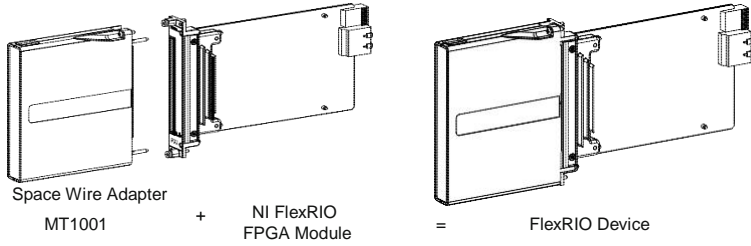
**Note** Before programming the MT1001 adapter, the appropriate software and hardware must be installed. Installation Guides are encompassed in the NI FlexRIO FPGA Module Installation Guide and Specifications.



**Note** For electromagnetic compatibility (EMC) compliance, operate this device according to the documentation.

The following figure shows an example of a properly connected NI FlexRIO device.

**Figure 1. NI FlexRIO Device**



## Related Information

[MT1001 Specifications](#) on page 20

# Contents

Overview .....	1
Electromagnetic Compatibility Guidelines.....	4
Connecting Cables.....	4
How to Use Your NI FlexRIO Documentation Set .....	4
Key Features.....	5
Front Panel and Connector Pinouts .....	5
Space Wire Connector .....	6
Block Diagram .....	8
Register Features.....	9
MBA Registers.....	9
Data interchange with the 1892ХД1Я chip via the MBA adapter .....	12
Memory Card .....	15
MT1001 Component-Level Intellectual Property (CLIP). ...	17
Using Your MT1001 with a LabVIEW FPGA Example VI .....	18
MT1001 Configuration Design Library.....	20
MT1001 Specifications .....	20
Environment .....	22
Operating Environment.....	22
Storage Environment .....	22
Installing PXI EMC Filler Panels .....	23
Where to Go for Support .....	24

# Electromagnetic Compatibility Guidelines

---

This product was tested and complies with the regulatory requirements and limits for electromagnetic compatibility (EMC) stated in the product specifications. These requirements and limits are designed to provide reasonable protection against harmful interference when the product is operated in the intended operational electromagnetic environment.

This product is intended for use in industrial locations. However, harmful interference may occur in some installations, when the product is connected to a peripheral device or test object, or if the product is used in residential or commercial areas. To minimize interference with radio and television reception and prevent unacceptable performance degradation, install and use this product in strict accordance with the instructions in the product documentation.

Furthermore, any modifications to the product not expressly approved by Mush Technologies could void your authority to operate it under your local regulatory rules.



**Caution** To ensure the specified EMC performance, operate this product only with shielded cables and accessories.



**Caution** To ensure the specified EMC performance, the length of all I/O cables must be no longer than 3 m (10 ft).



**Caution** To ensure the specified EMC performance, you must install PXI EMC Filler Panels (National Instruments part number 778700-1) in adjacent chassis slots.

## Related Information

[Installing PXI EMC Filler Panels](#) on page 23

# Connecting Cables

---

1. Use the J1ЦKB.685663.002 cable to connect Space Wire Channels connectors CH0..CH3

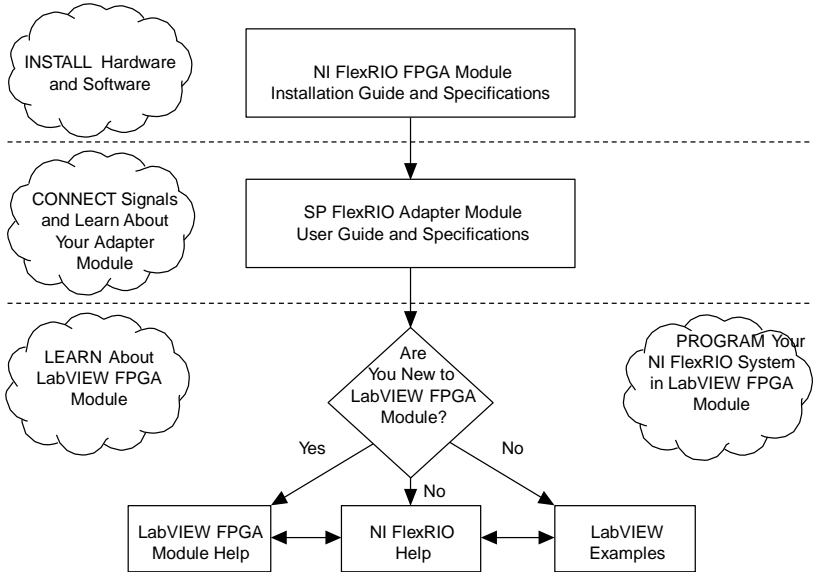
## Related Information

[MT1001 Front Panel and Connectors Pinouts](#) on page 5

# How to Use Your NI FlexRIO Documentation Set

Refer to Figure 2 and Table 1 to learn how to use your FlexRIO documentation set.

**Figure 2.** How to Use Your NI FlexRIO Documentation Set.



**Table 1.** NI FlexRIO Documentation Locations and Descriptions

Document	Location	Description
<i>NI FlexRIO FPGA Module Installation Guide and Specifications</i>	Available from the Start menu and at <a href="http://ni.com/manuals">ni.com/manuals</a> .	Contains installation instructions for your NI FlexRIO system and specifications for your FPGA module.
<i>MT1001 User Manual and Specifications</i> (this document)	Available from the Start menu and at <a href="http://www.mush-tech.com/manuals">www.mush-tech.com/manuals</a> .	Contains signal information, examples, CLIP details, and specifications for your adapter module.

**Table 1. NI FlexRIO Documentation Locations and Descriptions (Continued)**

<b>Document</b>	<b>Location</b>	<b>Description</b>
<i>LabVIEW FPGA Module Help</i>	Embedded in <i>LabVIEW Help</i> and at <a href="http://ni.com/manuals">ni.com/manuals</a> .	Contains information about the basic functionality of the LabVIEW FPGA Module.
<i>NI FlexRIO Help</i>	Available from the Start menu and at <a href="http://ni.com/manuals">ni.com/manuals</a> .	Contains FPGA Module, adapter module, and CLIP configuration information.
LabVIEW Examples	Available in NI Example Finder.	Contains examples of how to run FPGA VIs and Host VIs on your device.
IPNet	<a href="http://ni.com/ipnet">ni.com/ipnet</a>	Contains LabVIEW FPGA functions and intellectual property to share.
NI FlexRIO product page	<a href="http://ni.com/flexrio">ni.com/flexrio</a>	Contains product information and data sheets for NI FlexRIO devices.

## Key Features

---

The MT1001 includes the following key features:

- Sending/receiving data with high noise stability up to 200Mbps
- Full compatibility with the Space Wire standard (ECSS-E50-12A/C)
- 4 external Space Wire ports and internal router

## Front Panel and Connector Pinouts

---

Table 2 shows the front panel connector and signal descriptions for the MT1001.

Detail descriptions connector pinouts and signals presented in the figure 2 are presented in the section of technical specifications.




**Caution** To avoid permanent damage to the MT1001, disconnect all signals connected to the MT1001 before powering down the module, and connect signals only after the adapter has been powered on by the NI FlexRIO FPGA module.



**Caution** Connections that exceed any of the maximum ratings of any connector on the MT1001 can damage the device and the chassis. NI and SP is not liable for any damage resulting from such connections.

**Table 2. MT1001 Front Panel Connectors**

Device Front Panel	Connector	Signal Description
	CH 0	Space Wire Channels <0..3>
	CH 1	
	CH 2	
	CH 3	

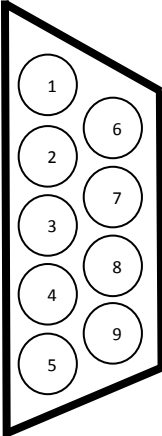
**Related Information**

[MT1001 Specifications](#) on page 20

# Space Wire Connector

The table 3 shows the pin-connection diagram of contacts (pinouts) of Space Wire MT1001 connector.

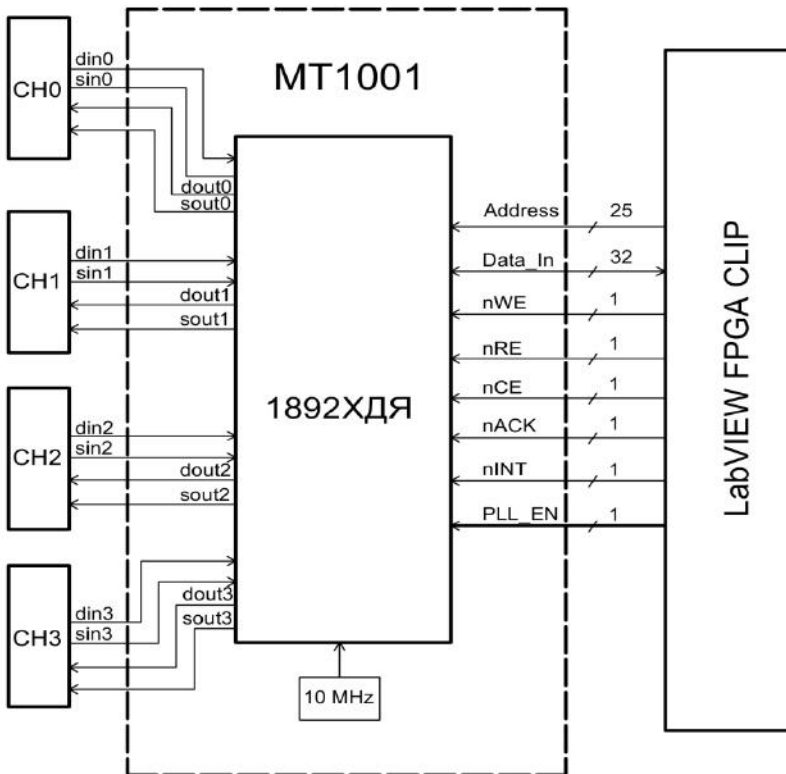
**Table 3.** MT1001 Connector Pinouts

Space Wire Connector	Pin	Signal	Signal Description
	1	$D_{in} +$	Data In
	2	$S_{in} +$	Strobe In
	3	Inner Shield	Ground
	4	$S_{out} -$	Strobe Out
	5	$D_{out} -$	Data Out
	6	$D_{in} -$	Data In
	7	$S_{in} -$	Strobe In
	8	$S_{out} +$	Strobe Out
	9	$D_{out} +$	Data Out

# Block Diagram

The following figure shows the MT1001 structural diagram with ways of transmitting signals and IP-cores at the component level (CLIP) into the adapter and the device MT1001.

Figure 3. MT1001 Structural Diagram





# Registers Features

The pairing interface is designed to exchange data with the registers of the SWIC, PMSC and DPRAM memory. The data is exchanged by performing of 32-bit words. When writing data in the DPRAM memory by using the nWE signal, from 1 to 4 bytes of 32-bit words can be written.

## MBA Registers

### 1. Interrupt Request Register (Register QSTR)

The following table shows the format of the QSTR Register.

**Table 4.** MT1001 QSTR Register

Bit Number	Conventional Designation (Symbol)	Purpose
<b>Interruptions from the PMSC</b>		
31	MASTER_DONE	The end of transmission of a data unit by the PCI bus in the Master mode
30	MASTER_ERROR	Detection of an error by transmission of a data unit by the PCI bus in the Master mode
29	MASTER_WMARK	The prior notice of transmission by the PCI bus in the Master mode of a number of words of the TMR_PCI register set by the Water Mark field
28	INT_MBR	Data record sign in the MBR_PCI mailbox register of the PMSC controller from the PCI bus. This bit is nullified when reading contents of the MBR_PCI register
27	SWIC3_TX_DATA	End of transmission of a data unit or a chain of data units by the DMA SWIC3_TX_DATA channel
26	SWIC3_TX_DESC	End of transmission of a data unit or a chain of data units by the DMA SWIC3_TX_DESC channel
25	SWIC3_RX_DATA	End of transmission of a data unit or a chain of data units by the DMA SWIC3_RX_DATA channel
24	SWIC3_RX_DESC	End of transmission of a data unit or a chain of data units by the DMA SWIC3_RX_DESC channel
<b>Interruptions from the DMA SWIC2</b>		
23	SWIC2_TX_DATA	End of transmission of a data unit or a chain of data units by the DMA SWIC2_TX_DATA channel

22	SWIC2_TX_DESC	End of transmission of a data unit or a chain of data units by the DMA SWIC2_TX_DESC channel
21	SWIC2_RX_DATA	End of transmission of a data unit or a chain of data units by the DMA SWIC2_RX_DATA channel
20	SWIC2_RX_DESC	End of transmission of a data unit or a chain of data units by the DMA SWIC2_RX_DESC channel
Interruptions from the DMA SWIC1		
19	SWIC1_TX_DATA	End of transmission of a data unit or a chain of data units by the DMA SWIC1_TX_DATA channel
18	SWIC1_TX_DESC	End of transmission of a data unit or a chain of data units by the DMA SWIC1_TX_DESC channel
17	SWIC1_RX_DATA	End of transmission of a data unit or a chain of data units by the DMA SWIC1_RX_DATA channel
16	SWIC1_RX_DESC	End of transmission of a data unit or a chain of data units by the DMA SWIC1_RX_DESC channel
Interruptions from the DMA SWIC0		
15	SWIC0_TX_DATA	End of transmission of a data unit or a chain of data units by the DMA SWIC0_TX_DATA channel
14	SWIC0_TX_DESC	End of transmission of a data unit or a chain of data units by the DMA SWIC0_TX_DESC channel
13	SWIC0_RX_DATA	End of transmission of a data unit or a chain of data units by the DMA SWIC0_RX_DATA channel
12	SWIC0_RX_DESC	End of transmission of a data unit or a chain of data units by the DMA SWIC0_RX_DESC channel
Interruptions from the SWIC3		
11	SWIC3_TIME	Received control code
10	SWIC3_ERR	Receiving channel error
9	SWIC3_LINK	The connection is established/Received the package
Interruptions from the SWIC2		
8	SWIC2_TIME	The control code is received

7	SWIC2_ERR	Error in the channel of reception
6	SWIC2_LINK	Connection is set/The packet is received
Interruptions from the SWIC1		
5	SWIC1_TIME	The control code is received
4	SWIC1_ERR	Error in the channel of reception
3	SWIC1_LINK	Connection is set/The packet is received
Interruptions from the SWIC0		
2	SWIC0_TIME	The control code is received
1	SWIC0_ERR	Error in the channel of reception
0	SWIC0_LINK	Connection is set/The packet is received

All signals of internal interruptions arrive on inputs of the QSTR register, available only on reading. The initial status of QSTR – all zero (there are no interruption requests). All unmasked interruptions combined by "or" and arrive on the nINT external output.

In the course of service of the interruption it is necessary to analyze the status of the device for determination of reasons of its origin. The interruption reset is performed at the time of exclusion of the causes of interruption. For example, the interruption from SWIC is reset in case of recording of units in the appropriate discharges of the SWIC\_STATUS register.

The data interchange with the registers is carried out by 32-bit words. If discharges of the register aren't used, then zero are read out from them. In case of record in these discharges it is necessary to specify zero.

## 2. MASKR, Interrupt Requests Mask Register.

Each 1892XD1Я chip internal interrupt is masked by means of an available on recording and reading MASKR mask register, the format of which is similar to the format of the QSTR register. In the initial status MASKR=0 (all interrupt traps are masked).

## 3. BDR, Address and Data Buffering Register.

The 32-bit register BDR is intended for execution of the procedure of data reading from the SWIC, PMSC registers and an address window of the PCI bus. The BDR register is available on recording and reading with FlexRio from the data interchange bus. Discharges of the BDR register correspond to discharges of the D [31:0] data bus. The initial contents of the BDR register – all zero.

## 4. Register BUSY.

The register BUSY is intended for synchronization of a data interchange with the SWIC, PMSC, registers and also with the PCI bus in the Master mode through an address window. The bit 31 of this register defines the polarity of the nACK signal. The format of the BUSY register is given in Table 5.

**Table 5. Format of the BUSY register**

Bit Number	Conventional Designation (Symbol)	Purpose	Access
31	ACK	Polarity of the nACK signal: 1 – high active level; 0 – low active level	RW
30:1	-	Isn't used	R
0	BUSY	Sign of the MBA employment during data interchange with the SWIC, PMSC registers, or with the PCI bus in Master mode: 0 – The MBA isn't busy; 1 – The MBA is busy. It is set in 1: <ul style="list-style-type: none"> <li>in case of initialization of the procedure of data recording in the SWIC, PMSC registers or an address window of the PCI bus;</li> <li>in case of the address recording in the BDR register for initialization of the data reading procedure from the SWIC, PMSC registers or an address window of the PCI bus.</li> </ul> It is set in 0: <ul style="list-style-type: none"> <li>in case of completion of the data recording in the SWIC, PMSC registers or an address window of the PCI bus;</li> <li>in case of completion of the data reading from the SWIC, PMSC registers, or address window of the PCI bus and recording of these data in the BDR register.</li> </ul>	R

The BUSY register is available on reading and recording of the bit 31. The initial status – 0.

## Data interchange with the 1892ХД1Я chip via the MBA Adapter

### 1. General provisions

Via the MBA adapter the SWIC, PMSC registers, DPRAM memory, and also the bus PCI are available in the Master mode through an address window.

The data interchange with the SWIC, PMSC registers can be executed by two methods:

- with use of the BUSY sign of employment,
- with use of the nACK signal.

## 2. Data interchange with use of the employment BUSY sign.

The data interchange with the MBA registers and DPRAM memory is executed for one writing or reading operation.

Data recording in the SWIC, PMSC registers and PCI bus address window is executed for 2 operations:

- data reading operation: BUSY inquiry (waiting BUSY=0),
- data writing operation. The operation address is defined according to the Table 6.

Data reading from the SWIC, PMSC registers and address window of the PCI bus is executed for 4 operations:

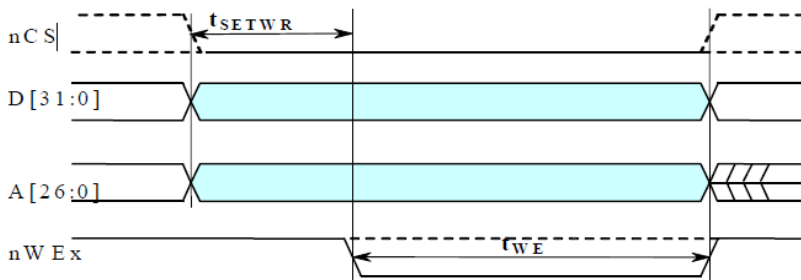
- reading operation: inquiry BUSY=0 (waiting BUSY=0);
- address register writing operation (or PCI bus windows) in the BDR register.

After execution of this operation the procedure of reading contents of the register (a PCI bus window) is initiated. The result of execution of the procedure is written in the BDR register;

- reading operation: inquiry BUSY=0 (waiting BUSY=0);
- reading operation of contents of the BDR register.

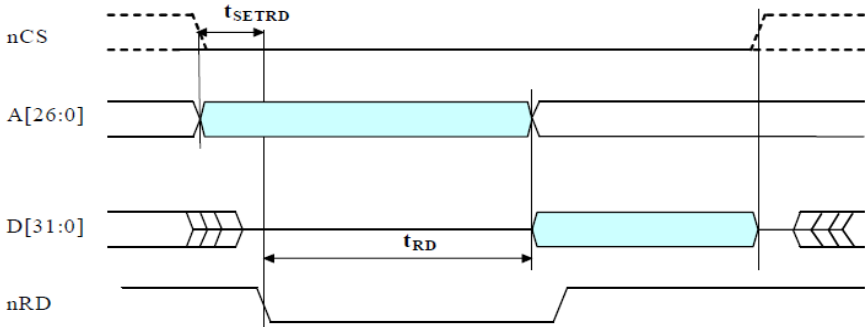
The figure 4 shows the time diagram of data writing operation in the 1892XД1Я chip through the MBA.

**Figure 4. Time diagram of data recording in the 1892XД1Я**



The figure 5 shows the time diagram of data reading operation from 1892XД1Я chip through the MBA.

**Figure 5.** Time diagram of data reading from the 1892ХД1Я

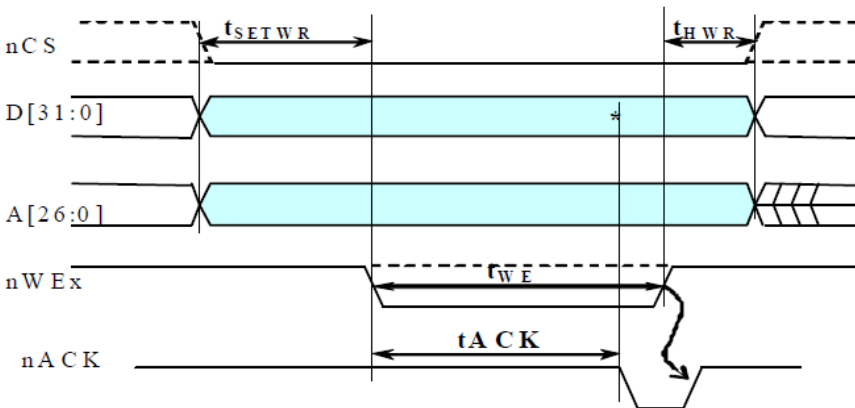


### 3. Data interchange with use of the nACK signal.

In this case all data interchanges with the MBA is executed by one writing or reading operation.

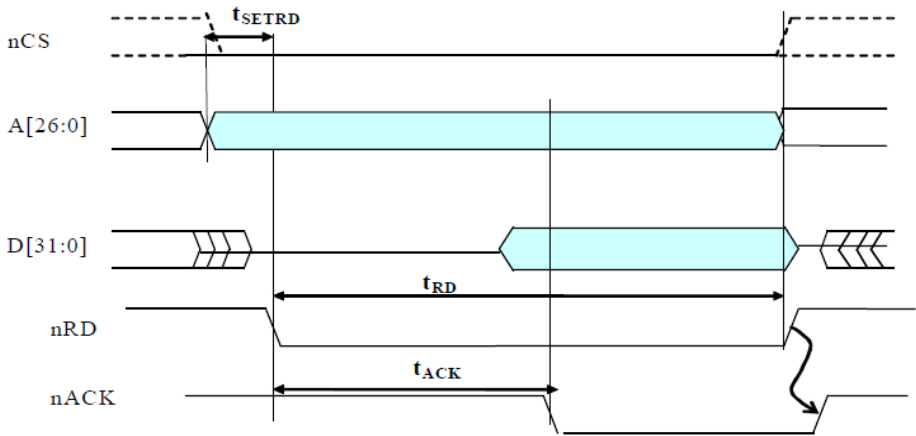
The figure 6 shows the time diagram of data writing operation into the 1892ХД1Я through the MBA.

**Figure 6.** Data recording time diagram in the 1892ХД1Я



The figure 7 shows the time diagram of data reading diagram from the 1892ХД1Я chip through the MBA.

**Figure 7.** Data reading time diagram from the 1892XД1Я through the MBA



## Memory Card

The table 6 shows the memory stick of the 1892XД1Я chip in case of a data interchange via the MBA adapter.

The address of the 32-bit word is transferred through the external A [24:0] bus.

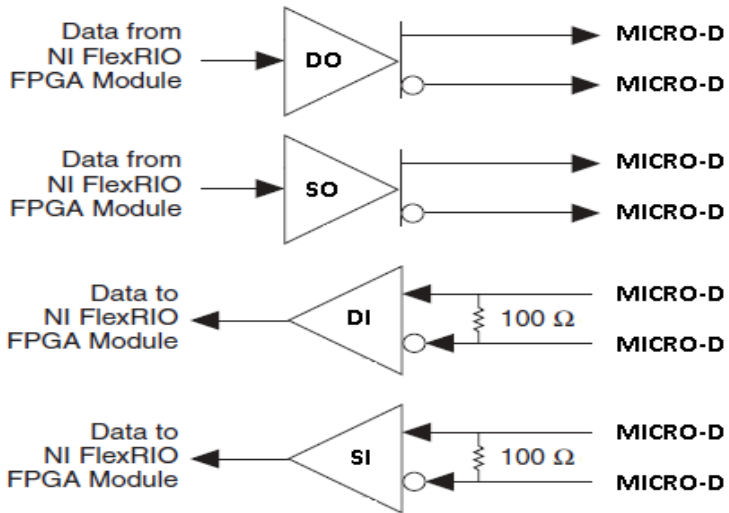
**Table 6.** Memory Stick of the 1892HD1Y Chip in case of a data interchange via the MBA adapter.

Address Range	Area Name	Area Volume, MB
1FF_FFFC-1E0_0000	Reserve	2
1DF_FFFC-1C0_0000	MBA Registers	2
1BF_FFFC-1B0_0000	DMA_SWIC3 Registers	1
1AF_FFFC-1A0_0000	SWIC3 Registers	1
19F_FFFC-190_0000	DMA_SWIC2 Registers	1
18F_FFFC-180_0000	SWIC2 Registers	1
17F_FFFC-170_0000	DMA_SWIC1 Registers	1
16F_FFFC-160_0000	SWIC1 Registers	1
15F_FFFC-150_0000	DMA_SWIC0 Registers	1
14F_FFFC-140_0000	SWIC0 Registers	1
13F_FFFC-120_0000	PMSC Registers	2
11F_FFFC-104_0000	Reserve	.
103_FFFC-100_0000	DPRAM Memory	0.256
0FF_FFFC-000_0000	Reserve	16

For more detailed operation see the chip specification.

The figure 8 shows the structure of the MT1001 duplex channel.

**Figure 8.** Block/skeleton diagram.





# MT1001 Component-Level Intellectual Property (CLIP)

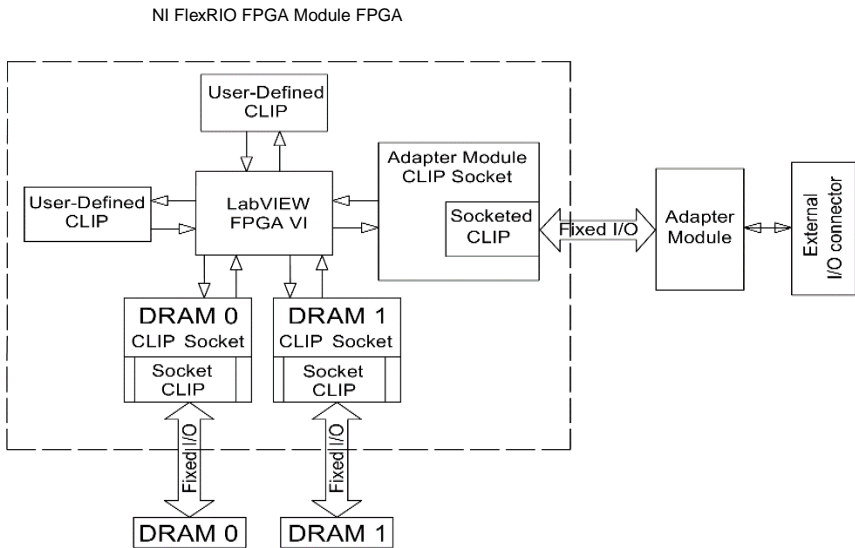
The LabVIEW FPGA Module includes component-level intellectual property (CLIP) for HDL IP integration.

NI FlexRIO devices support two types of CLIP: user-defined and socketed.

- *User-defined CLIP* allows you to insert HDL IP into an FPGA target, enabling VHDL code to communicate directly with an FPGA VI.
- *Socketed CLIP* provides the same IP integration of the user-defined CLIP, but also allows the CLIP to communicate directly with circuitry external to the FPGA. Adapter module socketed CLIP allows your IP to communicate directly with both the FPGA VI and the external adapter module connector interface.

The following figure shows the relationship between an FPGA VI and the CLIP.

**Figure 9.** CLIP and FPGA VI Relationship



The MT1001 ships with socketed CLIP items that add module I/O to the LabVIEW project.

# Using Your MT1001 with a LabVIEW FPGA Example VI

---



**Note** You must install the software before running this example. Refer to the *NI FlexRIO FPGA Installation Guide and Specifications* for more information about installing your software.

The NI FlexRIO Adapter Module Support software includes an example project to help you get started creating your LabVIEW FPGA application. This section explains how to use an existing LabVIEW FPGA example project to acquire samples with the MT1001



**Note** The examples available for your device are dependent on the version of the software and driver you are using. For more information about which software versions are compatible with your device, visit [ni.com/info](http://ni.com/info), enter `rdsoftwareversion` in the text field, and click the NI FlexRIO link in the results.

The MT1001example project includes the following components:

- A LabVIEW FPGA VI that can be compiled and run on the FPGA embedded in the hardware
- At least one VI that runs on Windows and interacts with the LabVIEW FPGA VI



**Note** In the LabVIEW FPGA Module software, NI FlexRIO adapter modules are referred to as *IO Modules*.

## Creating a LabVIEW Project

This section explains how to set up your target and create an FPGA VI and host VI for data communication. This section focuses on proper project configuration, proper CLIP configuration, and how to access the MT1001 I/O nodes.

### Creating a Project

1. Launch LabVIEW, or if LabVIEW is already running, select **File»Create Project**.
2. In the **Create Project** dialog box, select **LabVIEW FPGA Project** and click **Finish**.
3. Select **FlexRIO on My Computer** and click **Next**.
4. Either discover a LabVIEW FPGA target in your system or create a new system and specify an FPGA target for which to construct a project.
5. Click **Finish** in the **Project Preview** dialog box.
6. Click **File»Save** and specify a name for the project.

## Creating an FPGA Target VI

1. In the **Project Explorer** window, expand **FPGA Target**.
2. Right-click **IO Module** in the **Project Explorer** window and select **Properties**.
3. Select the **Enable IO Module**.
4. Select the MT1001 from the **IO Module** list. The available CLIP for the MT1001 is displayed in the Component Level IP pane.
5. Select the MT1001 Sample CLIP in the Name list of the Component Level IP pane.
6. Click **OK**.
7. Select the destination folder for the new file, specify a file name, and click **OK**. Use this FPGA VI with the MT1001 Configuration Design Library.
8. In the Project Explorer window, expand IO Module Tree View. Use any element under
9. IO Module (MT1001: MT1001) in the block diagram of the FPGA VI.
10. Add any FPGA code, controls, and indicators that you need. Refer to
11. Space Wire.lvproj for example FPGA code, controls, and indicators.
12. Click the Run button. LabVIEW creates a default build specification and begins compiling the VI. The Generating Intermediate Files window displays the code generation process. The Compilation Status window displays the progress of the compilation. The compilation takes several minutes.
13. Click **Close** in the **Compilation Status** window.
14. Save and close the VI.
15. Save the project.

## Creating a Host VI

1. In the **Project Explorer** window, right-click **My Computer** and select **New»VI** to open a blank VI.
2. Select **Window»Show Block Diagram** to open the VI block diagram.
3. Add the Open FPGA VI Reference function from the FPGA Interface palette to the block diagram.
4. Right-click the Open FPGA VI Reference function and select **Configure Open FPGA VI Reference**.
5. In the **Configure Open FPGA VI Reference** dialog box, select **VI** in the **Open** section.
6. In the **Select VI** dialog box, select your project under your device and click **OK**.
7. Click **OK** in the **Configure Open FPGA VI Reference** dialog box. The target name appears under the Open FPGA VI Reference function in the block diagram.
8. Open the FPGA Interface palette.
9. Add any Read/Write Control or Invoke Method nodes necessary to configure and communicate with your FPGA VI.
10. Add the Close FPGA VI Reference function to your block diagram.

11. Wire the FPGA VI Reference function to the Close FPGA VI Reference function.
12. Save and close the VI.
13. Save the project.

## Run the Host VI

1. Open the front panel of your host VI.
2. Click the **Run** button to run the VI.

# MT1001 Configuration Design Library

The MT1001 Configuration Design Library consists of host and FPGA VIs that provide an interface to configure the hardware on the MT1001.

The library allows you to perform the following actions:

- Configure Space Wire channels.
- Configure Space Wire channel modes.
- Read from and write to the Space Wire channel.
- Read the channel status.
- Reinitialize the CLIP.
- Query for CLIP errors.

The MT1001 Configuration Design Library relies on the Register Bus Design Library. The Register Bus provides a packet-based configuration interface which exposes all of the address spaces of the configurable chips and subsystems of the adapter module, without requiring hundreds of controls and indicators on your FPGA VI front panel.

The MT1001 Configuration Design Library host VIs all require a register bus object for the device you want to configure. Create the register bus object using `Open Session.vi`, or use `Space Wire KIT Open.vi`.

For more information about how to use the MT1001 Configuration Design Library, refer to the example located at `<labview>\examples\instr\Space Wire_KIT\Space Wire.lvproj`.

# MT1001 Specifications

The MT1001 specifications are warranted by design and under the following conditions unless otherwise noted:

- Chassis fan speed is set to High. In addition, NI recommends using slot blockers and EMC filler panels in empty module slots to minimize temperature drift.
- The MT1001 uses NI LabVIEW and LabVIEW FPGA software.

*Specifications* describe the warranted product performance over ambient temperature ranges

of 0 °C to 55 °C, unless otherwise noted.

*Typical* values describe useful product performance beyond specifications that are not covered by warranty and do not include guard bands for measurement uncertainty or drift. Typical values may not be verified on all units shipped from the factory. Unless otherwise noted, typical values cover the expected performance of units over ambient temperature ranges of 23 °C ± 5 °C with a 90% confidence level, based on measurements taken during development or production.

*Nominal* values (or supplemental information) describe additional information about the product that may be useful, including expected performance that is not covered under *Specifications* or *Typical* values. Nominal values are not covered by warranty.

## The MT1001 Main characteristics

The MT1001 Main characteristics are the followings:

- Connector Type: MICRO-DSUB 9POS RCPT,
- Number of Channels: 4,
- Communication Type: Duplex,
- Exchange Rate: 5 to 200 MBIT\S,
- Standard: ECSS-E-50-12A/C.

## Supply

Power consumption from NI FlexRIO FPGA module:

+2.5 V.....	350 mA, 0.625 W, max
+3.3 V.....	300 mA, 0.99 W, max

## Dimensions and Weight

Dimensions.....	12.9 × 2.0 × 12.1 cm (5.1 × 0.8 × 4.7 in)
Weight.....	413 g (14.6 oz)

# Environment

---

Maximum altitude .....2,000 m (at 25 °C ambient temperature)

Pollution Degree ..... 2

Indoor use only.

## Operating Environment

Ambient temperature range .....0 °C to 55 °

Relative humidity range .....10% to 90%

## Storage Environment

Ambient temperature range.....-40 °C to 70 °C

Relative humidity range.....5% to 95%

## Required hardware and software

NI PXI platform

NI PXI-795xR, NI PXIe-796xR

Space Wire adapter module

LabVIEW Run-Time Engine

Space Wire soft front panel

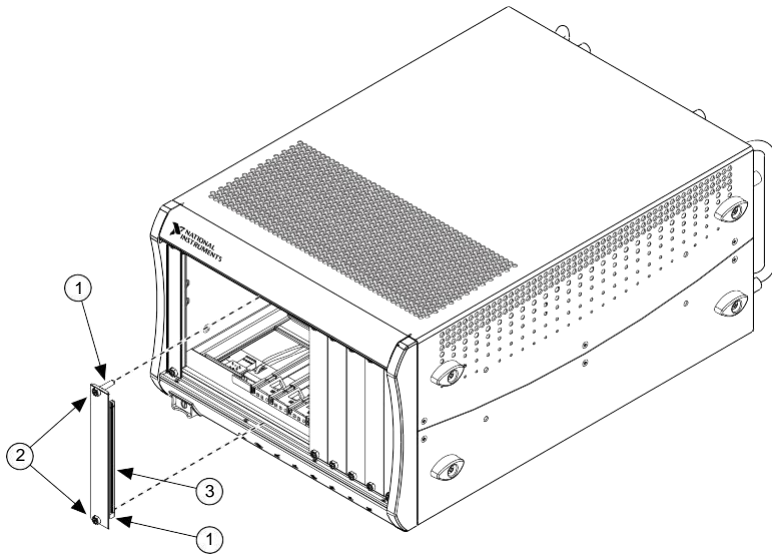
Space Wire user guide

# Installing PXI EMC Filler Panels

To ensure specified EMC performance, PXI EMC filler panels must be properly installed in your FlexRIO system. The PXI EMC filler panels (National Instruments part number 778700-01) must be purchased separately. For more installation information, refer to the *NI FlexRIO FPGA Module Installation Guide and Specifications*.

1. Remove the captive screw covers.
2. Install the PXI EMC filler panels by securing the captive mounting screws to the chassis, as shown in the figure below. Make sure that the EMC gasket is on the right side of the PXI EMC filler panel.

**Figure 10.** PXI EMC Filler Panels and Chassis



1. Captive Screw Covers
2. Captive Mounting Screws
3. EMC Gasket



**Note** You must populate all slots with a module or a PXI EMC filler panel to ensure proper module cooling. Do not over tighten screws (2.5 lb · in maximum). For additional information about the use of PXI EMC filler panels in your PXI system, visit [ni.com/info](http://ni.com/info) and enter `emcpanels`.

## Where to Go for Support

The Mush Technologies Web site is your complete resource for technical support. At <http://www.mush-tech.com/> you have access to everything from troubleshooting and application development self-help resources to email and phone assistance from MT Application Engineers.

Mush Technologies corporate headquarters is located at 123 Hovsep Emin Street, EIF Entrance, Yerevan, 0051, Armenia, Phone: +37410 21 97 26.