

University of Ljubljana
Faculty of *Electrical Engineering*
Laboratory of Photovoltaics and Optoelectronics



USB GPIB Interface v2 User's manual

Rev. 4.3, October 2010

Introduction

The interface communicates with PC through USB, which is also used to power it. On the other side GPIB (IEEE-488) connector is provided to connect to the bus. This enables control of instruments and other GPIB devices. Maximum number of GPIB devices connected is 14 plus this interface. Current version of firmware enables this device to be system controller only. Multi-controller and device operations are not supported at present.

Front matter

The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license under patent- or other industrial or intellectual property rights.

Connection to the PC

Universal Serial Bus (USB) is used for communication between the interface and the PC and to provide power. USB version is 2.0 with Full speed supported. The interface enumerates as a serial interface and drivers are requested. There are two types of drivers; Virtual Com Port (VCP) and D2XX drivers. Both can be downloaded from FTDI's homepage <http://www.ftdichip.com>. With D2XX you have access to the interface using DLL calls. The new drivers for LabVIEW require D2XX. You are advised to use these and the new version of LabVIEW drivers.

If you install VCP drivers another serial (COM) port is available. There is no need to adjust speed and flow control settings to communicate to the interface. The old drivers for LabVIEW require COM port therefore VCP drivers must be used in this case.

Note that only one type of drivers can be installed at a time.

If you have 64-bit Windows, you can only install CDM drivers. These are combined D2XX and VCP drivers.

Connection to the instruments

Up to 15 GPIB devices can be connected to a GPIB bus. This number includes the controller. Devices can be connected in any configuration (star, bus or combined). For correct operation, at least one third of devices must be turned on. Each device connected must have its own address on the bus.

LED indicator

The interface has one green LED indicator at the side with USB connector. This LED lights when the interface is initialised and ready for communication over GPIB bus.

D2XX or CDM drivers installation on Windows

- Before connecting the interface to the PC, check if you have USB drivers accessible (e.g. unzipped on your hard drive). You can download them from our website.
- Connect the interface to the PC.
- **Found New Hardware Wizard** will appear on your screen). Select: **No, not at this time** and click **Next** to continue.
- In the second window select **Install the software from a list or specific (Advanced)** and click **Next** to continue.
- Then select **Search for the best drivers in these locations**. If you are installing from a removable media (e.g. CD-ROM), tick **Search removable media (floppy, CD-ROM...)**. If you are installing from a hard drive, tick **Include this location in the search** and browse to the folder with drivers. Click **Next** to continue.
- Information window should now appear. Wait for the installation to complete.
- After the installation window will appear saying "**The wizard has finished installing the software for: FTDI FT8U2XX device**". Click **Finish** to close the window.

Interface commands

If you will be using LabVIEW drivers, there is no need to read this chapter. The commands described here are implemented in the driver.

Communication with the interface can be established from any program that provides serial communication or a programming language using DLL calls, depending on installed USB drivers. Commands for the interface begin with IB, which is followed by a command code, parameter and a carriage return (CR). Commands are case sensitive. Data transmitting is an exception and does not require CR. The interface has an on board FIFO and there is no need to wait for a command to end before sending next one. Commands for GPIB communication will power on the interface and perform interface clear if powered off.

Commands

IB<CR>

Powers on the interface and performs interface clear. The green LED indicator will turn on.

Returns: <ACK>.

IBC*x*<CR> and IB*c**x*<CR>

Both send byte *x* as a bus command with and without releasing ATN line respectively. The later can be used to send multi-byte bus commands. Byte timeout applies for this command.

Returns: <ACK>, 0x01, 0x02 or 0x08.

IB*data*

Sends binary *data* in BSC format. Data should begin with <DLE><STX> and end with <DLE><ETX>. If <DLE> is to be transferred, it is followed by another <DLE>. EOI will be asserted with the last byte transferred if enabled. If timeout occurs during operation, error code will be immediately sent, but the command should still be concluded with <DLE><ETX>. If <DLE><ACK> is encountered in a stream, the interface will respond with <ACK>. This can be used to know how much data was already sent (i.e. not waiting in buffer).

Total timeout starts when the command is initiated. If this timeout occurs the transfer of current byte is finished and error returned (normally 0x01 (Not Ready)). First byte must be transferred before First byte timeout occurs. After the first byte, time between byte is limited with Byte timeout.

Returns: <ACK>, 0x01, 0x02 or 0x08.

IB?<CR>

Starts receiving binary data in BSC format. Control bytes will be sent to the PC even if there is no data available. Reception is finished when EOI is detected, a timeout occurs or interrupted with <ESC>. On a timeout the interface normally returns 0x09 (No Data). Timeout operation is the same as for IB*data*.

The Byte timeout is not enabled if the transmit buffer is full. Transfer can be interrupted by sending <ESC> to the interface. This character can only be detected when the interface is waiting for data (i.e. not at full speed). The feature is useful to implement End Of String (EOS) character recognition.

Returns binary data in BSC and: <ACK>, 0x07, 0x03 or 0x09.

IBB<CR>

Starts reception of one byte. Received byte is sent to the PC alone. Byte timeout applies for this command. This is useful for reading serial poll status byte. If byte is not received, the interface returns 0x00 instead of byte read.

Returns byte and: <ACK>, 0x03 or 0x09.

IBI<type><CR>

Sends identification string to the PC. All strings ends with <CR><LF>.

- 0 Type of the device (the interface).
- 1 Manufacturer of the device.
- 2 Version number.

IBZ<CR>

Performs interface clear by pulling IFC low for 100 μ s. REN will be set according to setting.
Returns: <ACK>.

IBO<CR>

Powers off the interface and returns Write Mode, timeout and REN setting to default state.
Returns: No result code is returned.

IBe<Write Mode><CR>

Sets Write Mode (decimal value). The mode is reset to default when IBO command is issued and on power-on.

0-3 - EOI will be asserted with the last byte send.

4-7 - EOI will not be asserted with the last byte send.

Default: 0.

Note: CR or LF is not added to the end of the stream, even if Write Mode implies so.

Returns: <ACK> or <NAK> if the number is invalid.

IBt<TimeOutByte><CR>

Sets byte timeout (decimal value) in increments of 32.768 ms. This is the maximum time between two characters. The timeout is also used for commands that transfer only one byte (IBC, IBc and IBB). The timeout is accurate to one digit. The timeout is reset to default when IBO command is issued and on power-on. If set to 0, the timeout is disabled.

Valid range is 2-65535 (32-2,147 s). Default: 0 (disabled)

Returns: <ACK> or <NAK> if there is no number.

Note: Valid range for firmware version 2.5.4 and lower is 2-255 (32-8323 ms).

IBT<TimeOutTot><CR>

Sets total timeout for write and read (decimal value) in increments of 32.768 ms. The timeout is accurate to one digit. The timeout is reset to default when IBO command is issued and on power-on. If set to 0, the timeout is disabled.

Valid range is 2-65535 (32-2,147 s). Default: 61 (2 sec)

Returns: <ACK> or <NAK> if there is no number.

IBf<TimeoutFirst><CR>

Sets timeout to transfer first byte on read and write in increments of 32.768 ms. The timeout is accurate to one digit. If set to 0, the timeout is disabled. The timeout is reset to default when IBO command is issued and on power-on. If set to 0, the timeout is disabled.

Valid range is 2-65535 (32-2,147 s). Default: 61 (2 sec)

Returns: <ACK> or <NAK> if there is no number.

IBN<BytesToRead><CR>

Sets number of bytes that IB? command transfers. Setting holds for all subsequent reads, not just for the first one. If EOI is not asserted with last byte transferred IB? will return 0x07 instead of <ACK>. Maximum transfer rate will be about 30 % lower when this option is used.

Returns: <ACK>.

Note: This command is only implemented on firmware version 2.6 and above.

IBm<ren><CR>

Specifies if REN is asserted with IFC, ren is decimal value. Setting is reset to default when IBO command is issued and on power-on.

1 asserts REN, 0 leaves REN unasserted.

Valid values of ren: 0, 1. Default: 1 (asserted).

Returns: <ACK> or <NAK> if the number is invalid.

IBS<CR>

Returns state of GPIB control lines. The state is returned as is (active zero) as one byte. No <ACK> is returned. The bits are as follows:

7 (128)	6 (64)	5 (32)	4 (16)	3 (8)	2 (4)	1 (2)	0 (1)
SRQ	ATN	EOI	DAV	NRFD	NDAC	IFC	REN

IBQ<En><CR>

Enables ($En=1$) or disables ($En=0$) interrupt on SRQ line. SRQ line is monitored in idle mode. If the interrupt is enabled and SRQ goes active, <ENQ> character will be sent to the PC. <ENQ> is also sent for subsequent inactive-to-active transitions. If SRQ line is already active, and the interrupt enable command is issued, ENQ is also sent. Default value is disabled.

Returns: <ACK> or <NAK> if the number is invalid.

Return codes

One byte result code is sent after every command. The exception is when USB goes to suspend state and when the interface is powered off. In this case nothing is sent to the PC.

<ACK> (0x06)

Command completed normally.

0x07 – EOI Not Asserted

EOI was not asserted with the last byte transferred on read. More bytes might be at the device' output buffer.

Note: This response is only possible on firmware version 2.6 and above.

<NAK> (0x15)

Command is not recognised. It can also be sent if the command gets too long.

<ENQ> (0x05) - SRQ interrupt

SRQ line transitioned from inactive-to-active state. This is only possible during idle and if enabled.

0x01 – Not Ready

Time-out occurred before NRFD became false (during talking).

0x02 – Not Accepted

Time-out occurred before byte was accepted (NDAC became false) (during talking).

0x03 – Not DAV Released

Time-out occurred before DAV was released (during listening).

0x08 – No Listeners

Time-out occurred before NRFD or NDAC became true.

0x09 – No Data

Time-out occurred before reception (DAV true) of a byte.

USB suspend operation

When USB goes to suspend state current operation is aborted without result code and the interface is powered off. It remains in that state till first received byte from the PC.

LabVIEW drivers for D2XX

LabVIEW drivers are designed as a collection of VIs: USB GPIB Initialization, USB GPIB Write, USB GPIB Write List, USB GPIB Read, USB GPIB Clear, USB GPIB Read STB, USB GPIB Request Wait, USB GPIB Wait SRQ, USB GPIB Set timeout, USB GPIB Trigger, USB GPIB Trigger List and USB GPIB Close. In order to use the interface in LabVIEW, all functions that communicate with instruments should be replaced with ours. Our functions will pass control to LabVIEW VISA functions if VISA resource name does not begin with USB-GPIB::. This enables creation of only one program for both types of GPIB interfaces.

All functions except Initialization, Close and Wait SRQ perform addressing and un-addressing after completion if an address is given and re-addressing enabled at initialisation. Execution is divided into steps and if an error occurs at one step execution is stopped and proper error code returned. Each step is composed of sending interface command and waiting for a response. All functions send one command at a time and wait for acknowledge.

LabVIEW drivers are designed in LabVIEW 6.0. For LabVIEW 5.0 drivers contact us. Drivers for older versions of LabVIEW are not available.

To use these functions, virtual D2XX drivers should be installed for the interface.

Common parameters and returns to all functions

error in describes error conditions that occur before a VI runs. The default input of this cluster is no error. If there is an error wired to a VI, VI will be skipped and error passed to the output.

timeout [ms] specifies the maximum time a VI will use to send the data. A VI will use the IBT command to set timeout if different from current setting. Note that actual data transfer time will be limited to approx. 100 ms lower value in order to compensate for addressing time. This parameter applies to total timeout and is local to the VI. To set Byte Timeout and First byte timeout use USB GPIB Set timeout function.

VISA resource name contains the address of the GPIB device with which a function communicates and the interface used for the communication. To use our interface, instrument's address should be prefixed with USB-GPIB::. This prefixed should be used for all functions that communicate with our interface. Instrument's address is a decimal string. If you do not specify an address, a function does not perform addressing before they attempt to read and write the string or commands.

duplicate VISA resource name is the same as **VISA resource name** input.

error out contains error information. If error in indicates an error, **error out** contains the same error information. Otherwise, it describes the error status that this VI produces.

Code numbers means the following:

- 2 ENOL Write detected no Listeners.
Timeout when sending bus commands or data. This is returned if error 0x08 – No Listeners is returned from the interface.
- 6 EABO I/O operation aborted.
Serial port operation Error
PC-to-interface protocol error
A talker has no data to send (0x09 – No Data) on read
A listener is not ready for data (0x01 – Not Ready)
- 7 ENEB Non-existent board.
There is an error opening the serial port.
- 14 EBUS GPIB bus error.
Time-out occurred before NRFD became false (during talking) (0x01).
Time-out occurred before byte was accepted (NDAC became false) (during talking) (0x02).
Time-out occurred before reception (DAV true) of a byte.
This code is applicable to both bus commands and data.

String returned in **error out** has the following format: <name of VI>:<description of error>.

Description is added to extend error codes:

- I/O I/O error occurred when communicating with serial port.
- Ctrl Returned if there is a PC-to-interface protocol error.
- Not Rdy Returned if 0x01 – Not Ready response received from the interface.
- Not Acc Returned if 0x02 – Not Accepted response received from the interface.
- Not DAV Rel Returned if 0x03 – Not DAV Released response received from the interface.
- No Lstn Returned if 0x08 – No Listeners response received from the interface.
- No data Returned if 0x09 – No Data response received from the interface.

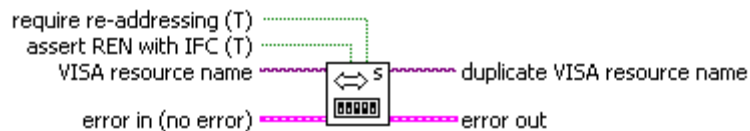
VIs

USB GPIB Initialization

Opens serial port, sets REN state, powers on the interface and performs interface clear using command IB<CR>. It also sets global variables.

If **require re-addressing** is TRUE, the function addresses the device before every read or write. If FALSE, the device must be able to retain addressing from one read or write to the next. This can be disabled in order to speed-up communication. Address is not retained when multiple addresses are used (Trigger List and Write List). The default is TRUE.

If **assert REN with IFC** is TRUE, the function asserts the Remote Enable line. The default is TRUE. Possible error codes returned: 6, 7.



USB GPIB Initialization.vi

USB GPIB Write

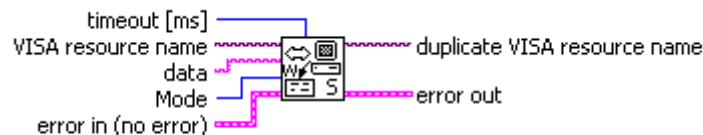
Writes data to the GPIB device identified by VISA resource name. The VI performs addressing if an address is given and re-addressing is enabled or the address is different from current.

data is the data the function writes to the GPIB device.

Mode indicates how to terminate the GPIB Write. The VI will use the IBc command to set EOI mode if different from current setting. Default is 0.

- 0 Send EOI with the last character of the string.
- 1 Append CR to the string and send EOI with CR.
- 2 Append LF to the string and send EOI with LF.
- 3 Append CR LF to the string and send EOI with LF.
- 4 Append CR to the string but do not send EOI.
- 5 Append LF to the string but do not send EOI.
- 6 Append CR LF to the string but do not send EOI.
- 7 Do not send EOI.

Possible error codes returned: 2, 6, and 14.



USB GPIB Write.vi

USB GPIB Read

Reads data from the GPIB device at the VISA resource name. If you do not specify an address, the function does not perform addressing it attempt to read data. The VI stops reading when EOI is detected, EOS character is received or **byte count** byte are received. Note that with EOS option, the interface can read more characters, but the VI will return only specified, the other characters might get lost. If firmware version is below 2.6 the same holds for byte count option.

byte count specifies the maximum number of bytes the function reads from the GPIB device. If positive value is specified and firmware version is 2.6 or above, maximum transfer rate will be about 30 % lower.

mode specifies conditions other than reaching byte count for terminating the read.

0 No EOS character. The EOS termination mode is disabled.

1 EOS character is CR. Read terminated on EOI, byte count, or CR.

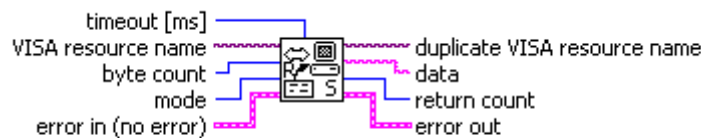
2 EOS character is LF. Read terminated on EOI, byte count, or LF.

x Any other mode indicates the number (decimal) of the desired EOS character.

data contains the string read without the EOS character.

return count is the length of **data**. If the read operation is terminated with EOS character or byte count, the number of bytes actually read from the instrument might be higher. These byte are lost.

Possible error codes returned: 2, 6 and 14.

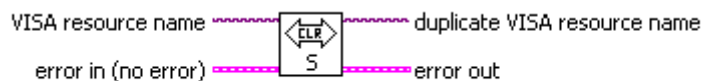


USB GPIB Read.vi

USB GPIB Clear

Sends either SDC (Selected Device Clear) or DCL (Device Clear). If you specify an address in address string, the function sends SDC to that address. If address string is unwired, the function sends DCL.

Possible error codes returned: 2, 6, and 14.



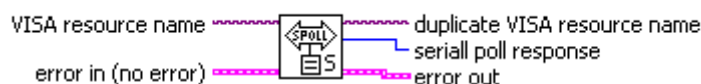
USB GPIB Clear.vi

USB GPIB Read STB

Serial polls a single device to get its status byte.

serial poll response is the status byte of the device at **address string**. You can learn how to interpret this byte from the documentation supplied with your device. -1 is returned if the byte can not be read.

Possible error codes returned: 2, 6, and 14.



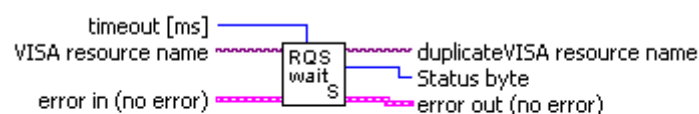
USB GPIB Read STB.vi

USB GPIB Request Wait

Waits for Service Request using USB GPIB Read STB and monitoring bit 6 of the status byte.

Status byte is **serial poll response** from USB GPIB Read STB.

Possible error code returned: 6.



USB GPIB Request Wait.vi

Waits for Service Request using serial poll.

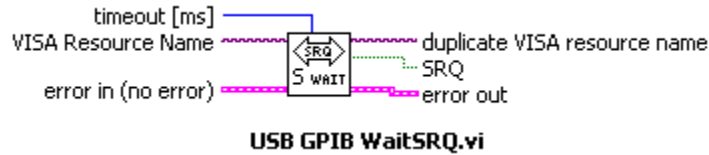
USB GPIB Wait SRQ

Waits for SRQ line to be asserted using interrupt. If the line is active when the VI is called it immediately returns.

Note: The function is passive to the bus and does not clear the SRQ flag.

SRQ is true if SRQ occurred within the timeout.

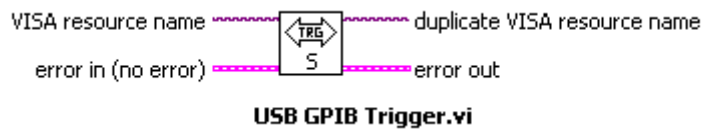
Possible error codes returned: 6 and 14.



USB GPIB Trigger

Triggers a single device using GET bus command. If you specify an address, the function sends trigger to that address. If you do not specify an address, the function does not perform addressing before it sends trigger. It assumes you have sent this commands another way.

Possible error codes returned: 2, 6, and 14.



USB GPIB Set timeout

Sets timeouts of the GPIB interface. The timeouts are also written to global variables.

Total timeout is maximum duration of a GPIB operation. If set to -1, the timeout will not change. Use 0 to disable the timeout. The defaults value is 10,000 ms.

First byte timeout is maximum duration between the start of a transfer command and the first byte transferred. If set to -1, the timeout will not change. Use 0 to disable the timeout. The defaults value is 10,000 ms.

Byte timeout is maximum duration between successive two bytes during write or read operation. This timeout is also used for commands that transfer only one byte (IBC, IBC and IBB). If set to -1, the timeout will not change. Use 0 to disable timeout. The defaults value is 1,000 ms.

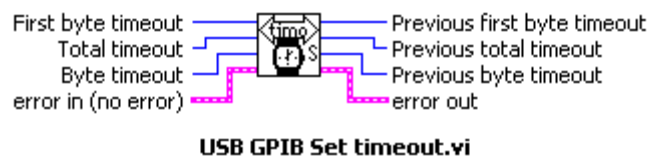
Previous total timeout is the previous total timeout period for all GPIB Functions.

Previous first byte timeout is the previous first byte timeout period for all GPIB Functions.

Previous byte timeout is the previous byte timeout period for all GPIB Functions.

Note: There should be at least one valid timeout always enabled, otherwise the interface might hang up.

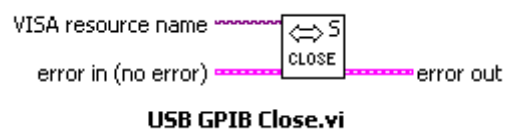
Possible error codes returned: 14 (interface error).



USB GPIB Close

Turns off GPIB interface and closes serial port.

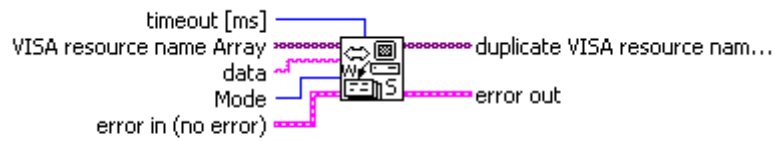
Possible error codes returned: 2, 6, and 14.



USB GPIB Write List

Writes data to multiple GPIB devices identified by VISA resource name array with a single transfer. All devices will receive data at exactly the same time. Other parameters and returns are the same as for USB GPIB Write.

Note: This VI is included in version 3.2 and above.

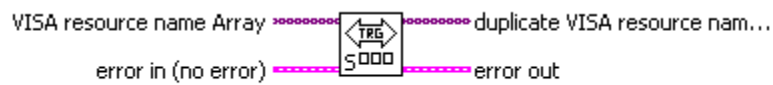


USB GPIB Write List.vi

USB GPIB Trigger List

Triggers multiple single device using GET bus command. Other parameters and returns are the same as for USB GPIB Write.

Note: This VI is included in version 3.2 and above.



USB GPIB Trigger List.vi

LabVIEW drivers for VCP

These drivers communicate with the interface through Virtual Serial Port (VCP). To use this functions virtual COM port should be installed as a FTDI driver. Functions access serial port through VISA library.

These VIs are very similar to LabVIEW drivers for D2XX with following exceptions:

- **address string** is used instead of **VISA resource name**. The string only contains decimal address of the instrument.
- **USB GPIB VISA** used for USB GPIB Initialization and USB GPIB Close is a VISA reference to serial port the interface is connected to. This parameter is global for all functions.

Note: These drivers are not maintained. Uses are encouraged to use D2XX drivers.

Firmware upgrades

Firmware of the microcontroller can be upgraded to support other GPIB features and options. Check our web page for firmware updates. If there is no need you should not update the firmware as an error during the upgrade (such as power failure) could cause the interface to malfunction and force you to send the interface to service.

Firmware revision history

Rev. V2.6, Apr. 2010

- Added command IBN - end read by byte count.
- If read operation is terminated because of byte count (IBN) (not EOI), 0x07 will be returned instead of <ACK> (0x06).

Rev. V2.5.4, Oct. 2009

- Byte timeout couldn't be disabled. This is now fixed.
- Maximum value for First byte timeout was limited to only 255.
- All timeouts (including) byte timeout are now 16 bits (maximum 65535).

Rev. V2.5.3, Feb. 2008

- Waiting for DLE ETX after error updated for 16 MHz clock (correct FTDI cycles).
- Sending NAK immediately after NotDAVrel error omitted.

Rev. V2.5.2, Oct. 2007

- T1 delay is now corrected (1100 ns).

Rev. V2.5.1, Nov. 2006

- T1 delay updated to standard.

Rev. V2.5.0, March 2006

- nop instruction added when reading FTDI to correctly support 16 MHz clock.
Access time for FTDI is now 1.5 cycle - 100 ns at 16 MHz
- Added debug commands to check lines as device.
- Corrected bug in converting received 0x0A to 0x0D in conjunction with debug commands.

Rev. V2.4.0, October 2005

- Each timeout can now be disabled.
- IBf, IBQ and IBS commands added.
- Handshake timeout not used any more for write while waiting for listener.
- Reading of one byte (IBB) corrected to return null in case of timeout.

Rev. V2.3.0, March 2005

- Interruption of data received corrected.
- Corrected bug with releasing REN on timeout.
- Added total timeout.
- Added option for Write Mode (disabling EOI).
- Added ACK during write.
- Compacted code in SendBinData function.
- Debug commands are now standard repertory.
- Operation with variable flags corrected.

Specifications for v2

USB

Power	5V/ 320 mA, Bus powered
Standard	2.0 (1.1 compatible)
Speed	Full speed (12 Mbps)
Connector	B Receptacle

GPIB

Standard	IEEE-488
Connector	24-pin centronics socket

Transfer rate

Talk	max. 420 KBps
Listen	max. 440 KBps

General

Dimensions (LxWxH)	66x66x28 mm ³
Weight	85 g

Specifications for v1

Transfer rate

Talk	max. 210 kBps
Listen	max. 220 kBps

General

Dimensions (LxWxH)	123x70x30 mm ³
Weight	106 g

Other specifications are the same as the one for v2.

Designed by

University of Ljubljana
Faculty of Electrical Engineering
Tržaška cesta 25, 1000 Ljubljana, Slovenia

Laboratory of Photovoltaics and Optoelectronics

<http://lpvo.fe.uni-lj.si>

<http://lpvo.fe.uni-lj.si/gpib/>

Dr. Marko Topič

Head of LPVO

E-mail: marko.topic@fe.uni-lj.si

Phone: +386 1 4768 470

Dr. Boštjan Glažar

E-mail: bostjan.glazar@fe.uni-lj.si

Phone: +386 1 4768 723

Last update: May 2010

Disclaimers

FEE reserves the right to make changes to its products and/or software at any time to improve design and/or performance.

This product is provided as an assembled KIT; that is without any warranty or guaranteed support. FEE is not liable and does not provide free of charge repair or refund in case of malfunction.

To the extent not prohibited by law, in no event will FEE be liable for any lost data, revenue or profit, or for special, indirect, consequential, incidental or punitive damages, regardless of the theory of liability (including negligence), arising out of or related to the use of or inability to use the product (including any software), even if FEE has been advised of the possibility of such damages. In no event will FEE' liability exceed the amount paid by you for the product.

This product is not designed for use in life support appliances or systems where malfunction of this product can reasonably be expected to result in personal injury. FEE' customers using or selling this product for use in such applications do so at their own risk and agree to fully indemnify FEE for any damages resulting from such application.

Document revision history

Rev 4.3

Date: October 2010

CDM drivers added.

Rev 4.2

Date: May 2010

New controller command described: IBN.

New LabVIEW VIs described: USB GPIB Write List.vi, USB GPIB Trigger List.vi.

Rev 4.1

Date: September 2009

Laboratory was renamed from Laboratory of Semiconductor Devices to Laboratory of Photovoltaics and Optoelectronics.

Rev 4.0

Date: March 2006

D2XX drivers installation added.

D2XX LabVIEW drivers added.

Firmware upgraded with boot-loader.

Specifications updated for version 2.0 of the interface.

Some spelling errors corrected.

Rev 3.1

Date: December 2005

USB GPIB Trigger was added.

Rev 3.0

Date: October 2005

IBdata, IB?<CR>, IBB<CR> and timeout interface commands complemented for new timeout features.

Description of new interface commands (IBS, IBQ and IBf) added.

ENQ return code added to the list.

LabVIEW drivers description updated (VISA)

Descriptions of USB GPIB Read and USB GPIB Read STB complemented.

New LabVIEW functions added (USB GPIB Wait SRQ and USB GPIB Set timeout).

Firmware revision V2.4.0 added.

Rev 2.0

Date: April 2005

Link to FTDI updated.

Description of IBdata and IB? commands corrected according to new firmware (timeouts).

Descriptions of other commands revised.

New commands, support by firmware V2.3 added: IBe, IBt, IBT, IBm.

More detailed description of error codes returned added.

Some grammatical errors corrected.