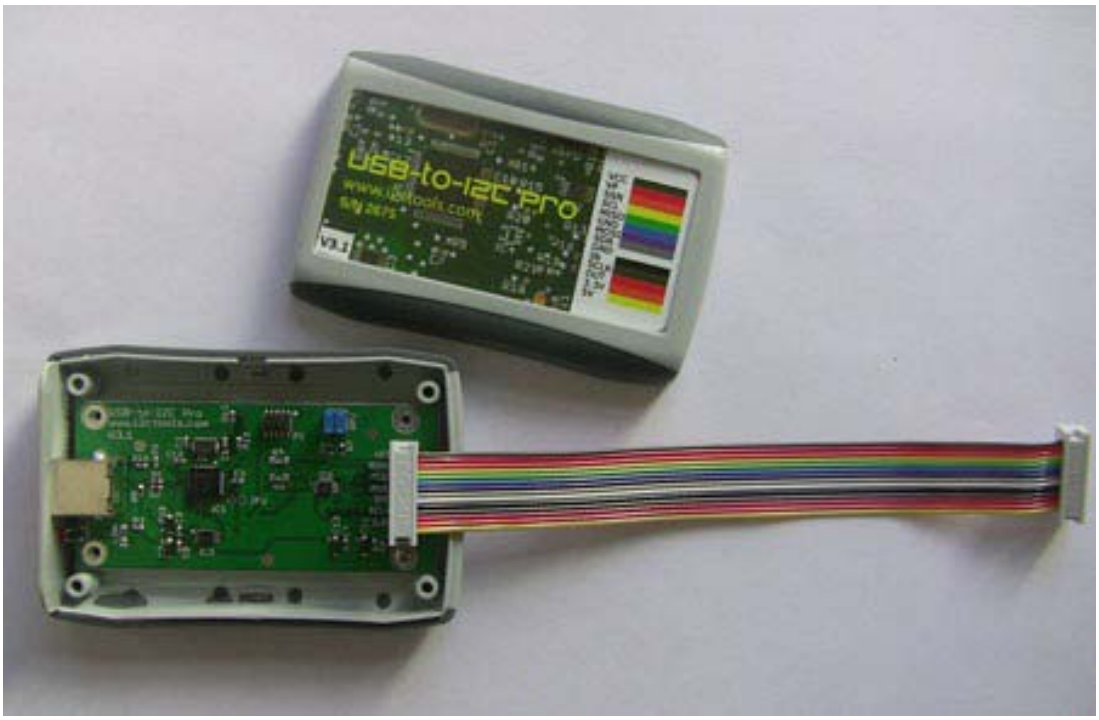
USB-to-I2C©

Hardware User's Manual



http://www.i2ctools.com/

Information provided in this document is solely for use with the USB-to-I2C product from SB Solutions, Inc. SB Solutions, Inc. reserves the right to make changes or improvements to this document at any time without notice. SB Solutions, Inc. assumes no liability whatsoever in the sale or use of this product, including infringement of any patent or copyright.
Reasonable efforts have been made to ensure the accuracy of the information presented. However, SB Solutions, Inc. assumes no responsibility for the accuracy of the information.
No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of
SB Solutions, Inc.
P.O. Box 4506
Cary, NC 27519-4506

Microsoft and Windows are registered trademarks of Microsoft Corporation.
Other brand names are trademarks or registered trademarks of their respective owners.

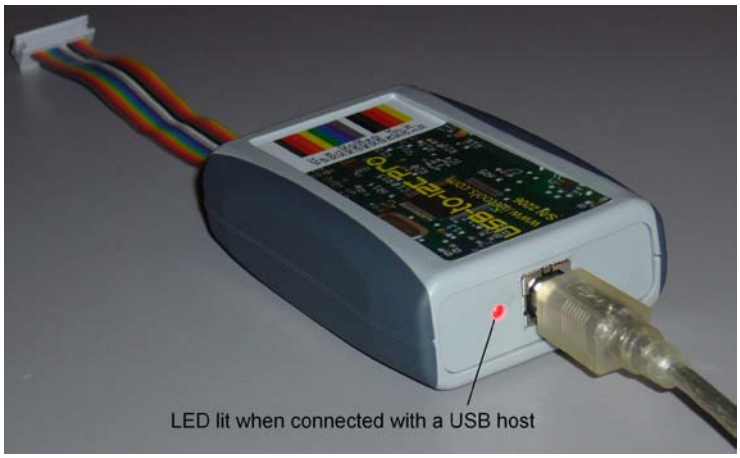
Copyright © 2004-2013 SB Solutions, Inc. All rights reserved.

Table of Contents

INTRODUCTION 1
USB-TO-I2C KIT CONTENTS 1
DISCLAIMER AND WARRANTY 2
GETTING STARTED 3
Assumptions 3
Static Handling Precautions 3
MINIMUM SYSYEM REQUIREMENTS 3
ADAPTER POWER REQUIREMENTS 3
INSTALLATION 3
USB-to-I2C Software 3
USB-to-I2C Hardware connection to computer USB port 3
HARDWARE DESCRIPTION 4
SPI Port 5
I²C Port 5
GPIO 5
USB-TO-I2C HARDWARE TO TARGET CONNECTION 7
Minimum hardware connection between USB-to-I2C and User Hardware 7
Typical hardware connection between USB-to-I2C and User Hardware 7
USING USB-TO-I2C TO POWER USER's SYSTEM 7
FIRMWARE UPDATES 8
SOFTWARE UPDATES 11
TECHNICAL SUPPORT 11
TROUBLESHOOTING 11

INTRODUCTION

The USB-to-I2C Hardware connects to a standard USB port found on most IBM-compatible PCs and provides bi-directional communication with I^2C devices using the I^2C protocol. The Hardware is powered directly from the PC's USB port.



The LED, located beside the USB connector, illuminates after it has successfully been enumerated by the USB host.

The USB-to-I2C software runs on Windows 98SE, ME, 2000, XP, Vista and is compatible with any PC hardware having a minimum of a Pentium processor and an USB port. The software allows the user to select a memory device (EEPROM, FRAM, or SRAM) from a menu and also provides several Universal modes to allow communication with most I^2C devices. The USB-to-I2C software is designed to operate only with the included USB-to-I2C hardware.

USB-TO-I2C KIT CONTENTS

- USB-to-I2C Hardware
- USB-to-I2C software installation CD-ROM (optional) which loads the following files:
  - Software license agreement
  - USB-to-I2C user Registration Form
  - USB-to-I2C Software User's Manual
  - USB-to-I2C Hardware User's Manual
  - USB-to-I2C Installation and User's Guide
  - USB-to-I2C application
  - USB-to-SPI application
  - I^2C and SPI DLL and examples

DISCLAIMER AND WARRANTY

Proper use of USB-to-I2C is the sole responsibility of the user. SB Solutions, Inc. is not responsible for any damage resulting from misuse or improper installation.

SB Solutions, Inc. will, at our option, repair or replace a defective USB-to-I2C hardware within thirty (30) days of the purchase date. Return shipping is the responsibility of the user.

GETTING STARTED

Assumptions

We are assuming the user of this product has experience with the I^2C Bus protocol. The I^2C Bus specification is a good source of detailed information about the I^2C Bus. The complete specification can be downloaded from the NXP Semiconductors website.

Static Handling Precautions

The USB-to-I2C hardware contains CMOS devices that can be damaged by ESD. It is recommended to use a ground strap or touching the PC case or other grounded source before unpacking or handling the USB-to-I2C hardware.

MINIMUM SYSYEM REQUIREMENTS

- ✓ PC Pentium 60 processor (or equivalent), 8 MB RAM, and 16 MB of hard drive space
- ✓ One USB port (either 2.0 or 1.1 compatible)
- ✓ 32-bit or 64-bit versions of XP, Vista, Windows 7, and Windows 8
- ✓ CD-ROM Drive for installation (note that we recommend using the latest installation package available for download from our website): http://www.i2ctools.com/downloads.html

ADAPTER POWER REQUIREMENTS

The USB-to-I2C hardware obtains its power from the computer's USB port. It should not be powered by an external power source.

INSTALLATION

USB-to-I2C Software

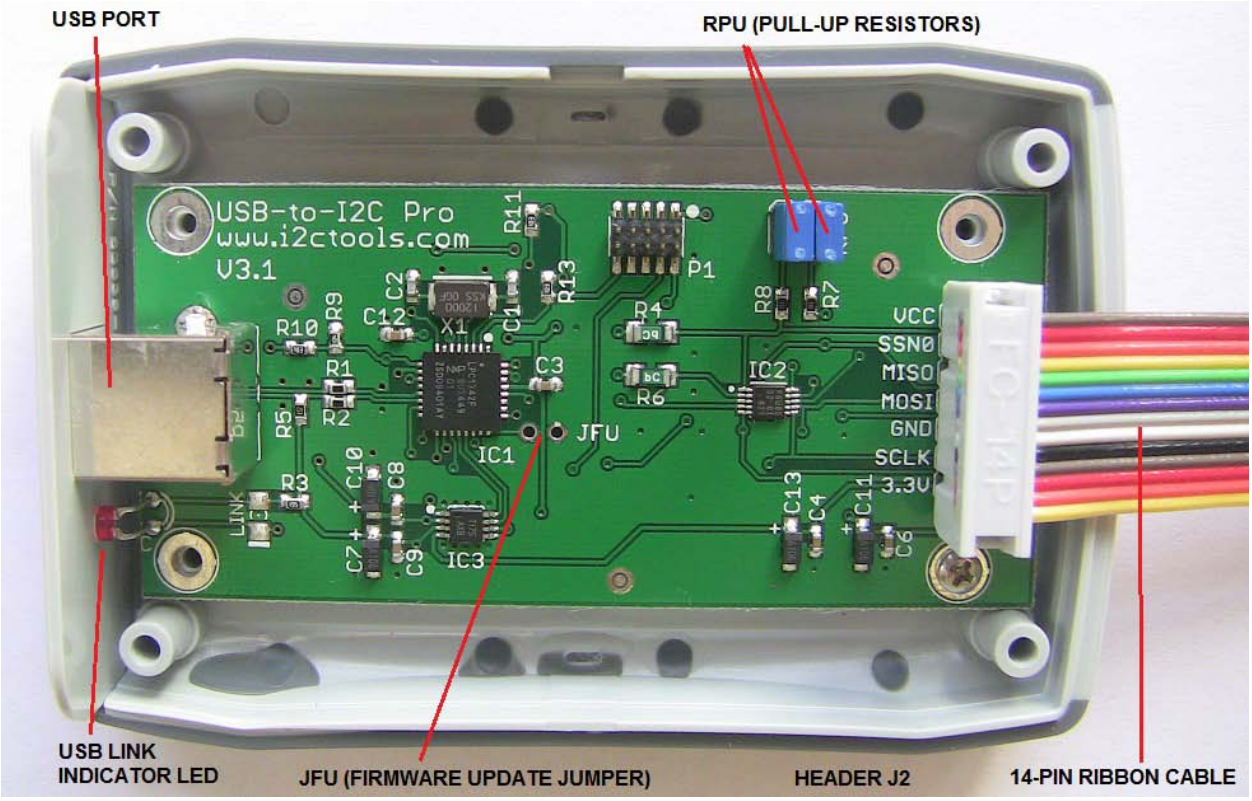
- You must have Administrator privileges to install the software
- Insert the USB-to-I2C installation CD-ROM into the appropriate drive.
- The installation software should automatically start
- If the installation software does not start automatically, double-click on the Setup.exe file on the CD-ROM
- Complete and send in the registration form via email

USB-to-I2C Hardware connection to computer USB port

- Neutralize any ESD (static charge) by touching the bare metal on the rear of your computer before removing the USB-to-I2C Hardware from the packaging
- Connect the USB-to-I2C Hardware to a USB Port using a standard USB cable

- The LED beside the USB port should illuminate

HARDWARE DESCRIPTION



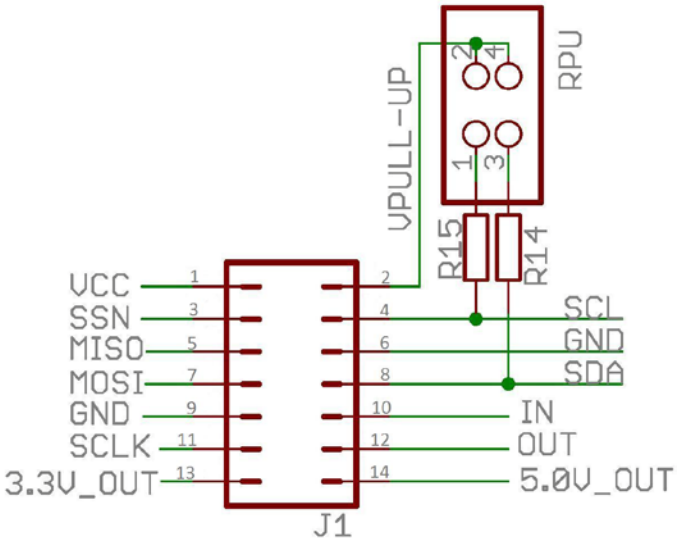
The figure above shows the items on the hardware which you should become familiar with. Here are the details:
USB Port – this is where you plug the usb cable into the board

USB LINK LED – the LED should be illuminated when the cable has been connected between the PC and the USB-to-I2C hardware, and the software has been installed correctly.

J2 Header
Header J1 contains the connections to a user target application.
A 14-pin ribbon cable is included with the following definitions:

Signal	Pin #	Ribbon cable color wire
VCC	1	Brown
VP	2	Red
SSN	3	Orange
SCL	4	Yellow
MISO	5	Green
GND	6	Blue
MOSI	7	Violet

SDA	8	Gray
GND	9	White
IN	10	Black
SCLK	11	Brown
OUT	12	Red
3.3V	13	Orange
5V	14	Yellow



VCC is the 3.3V power for the USB-to-I2C hardware. This supply is active whenever the hardware is plugged into a USB port. It is provided here to supply the pull-up resistors. It can also be used to power a user's target hardware.
VP – supplies the power to the I^2C pull-up resistors (see RPU). Note that the USB-to-I2C hardware by default has the jumpers connected at RPU and it is up to the user to ensure that a voltage source is connected to the VP pin. A pull-up voltage, up to 5.5V maximum, may be applied to the VP pin. If pull-up resistors are not required, please remove the two jumpers at connector RPU. It is not be necessary to apply a voltage at VP if the RPU jumper are not installed.
SPI Port – the SPI signals are connected to the target system from this port. Note that these signals drive a 3.3V output.
To communicate with a SPI slave, you require four signals and GND:
SSN – Slave Select (also known as CS or Chip Select)
MISO – Master In Slave Out is connected to the SPI slave output
MOSI – Master Out Slave In is connected to the SPI slave input
SCLK – **S**erial **Cl**ock outputs the clock to the slave device
The SPI pins may be used as IO pins controlled by the DLL, if not used for SPI functionality.
I^2C Port – connect these two pins (SDA and SCL) and GND to your target system. Details can be found in the section titled "USB-to-I2C Hardware To Target Connection" of this document.
SCL – is the serial clock generated by the USB-to-I2C hardware. Clock stretching is supported.
SDA – is the bidirectional I^2C data pin.
Supply a voltage at VP for the pull-up resistors or remove the jumpers at RPU before using the I2C port. It is advisable to remove the jumpers when connecting the USB-to-I2C hardware to a target system, assuming the target already has pull-up resistors. This will allow you to use the USB-to-I2C hardware at the voltage of your target system.
The I^2C pins may be used as open-drain IO pins controlled by the DLL, if not used for I2C functionality.
GPIO – the USB-to-I2C hardware has one general purpose input and one general purpose input.

IN pin operates as an input with the following characteristics:

INPUT PORT CHARACTERISTICS	
Vil low level input voltage	0.3*VDD = 0.99V
Vih high level input voltage	0.7*VDD = 2.31V

Note that the microcontroller is a 3.3V device but this input is 5V tolerant.
The **OUT** pin has the following characteristics:

OUTPUT PORT CHARACTERISTICS	
V_{oh}	**2.9V** (min @ 4 mA)
V_{ol}	**0.4V** (max @ 3.2mA)

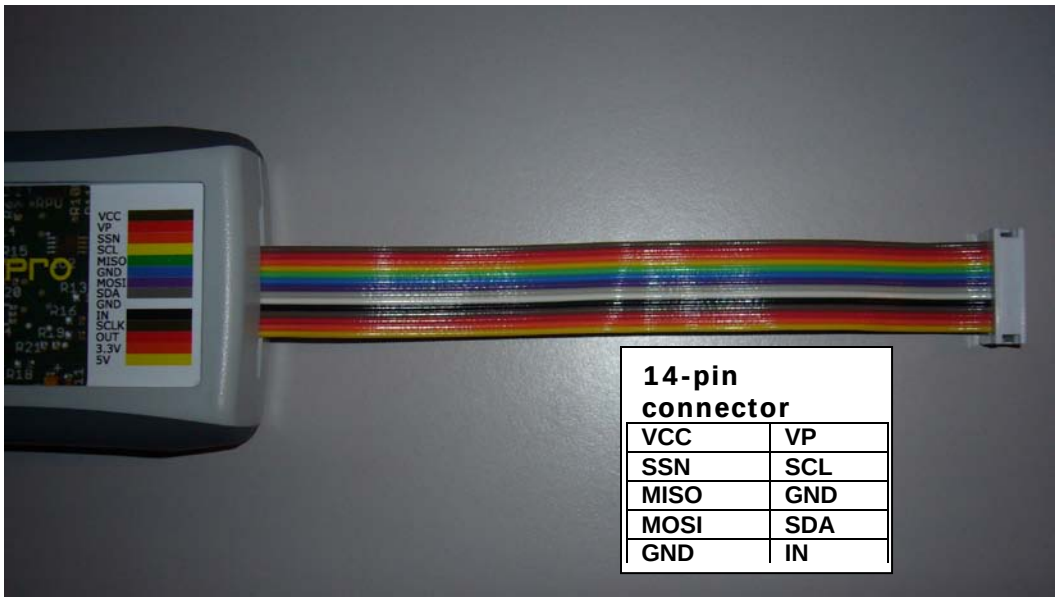
3.3V Switched Output – provides the user with the ability to power a low power target system at 3.3V. The USB-to-I2C software has the ability to switch this output on and off. The total current provided by the USB-to-I2C Hardware to the target system should not exceed 40mA. It is the user's responsibility to ensure that excessive current is not drawn from this supply pin.

5V Switched Output – provides the user with the ability to power a low power target system at 5V. The USB-to-I2C software has the ability to switch this output on and off. The total current provided by the USB-to-I2C Hardware to the target system should not exceed 40mA. It is the user's responsibility to ensure that excessive current is not drawn from this supply pin.

Jumper RPU allows the user to connect 3.3Kohm pull-up resistors, which are connected between the I^2C pins (SDA and SCL) and the VP pull-up voltage. The jumpers should be removed when connecting the USB-to-I2C hardware to a target system, if pull-up resistors are already available on the target system.

14-pin Ribbon Cable

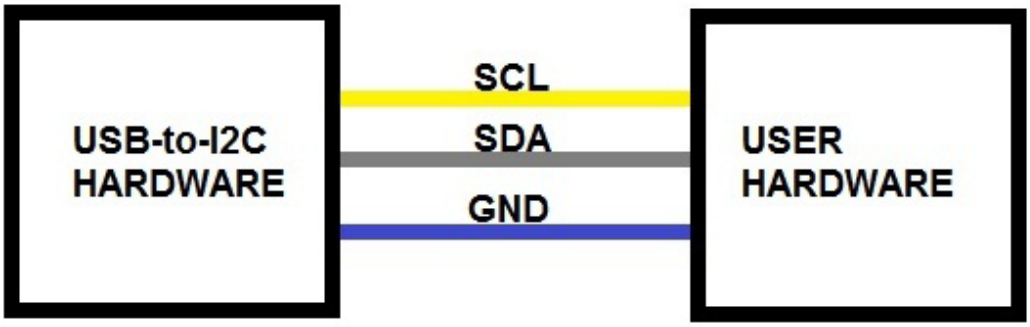
The USB-to-I2C hardware is shipped with a 14-pin ribbon cable with a 2x7 pin connector. The connector has a 0.1" (2.54mm) spacing. The pin configuration of the connector, when viewed from above, is shown here.



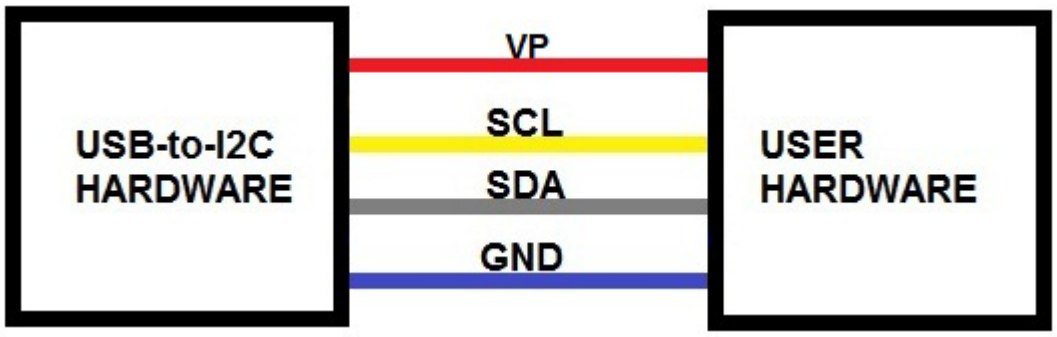
14-pin connector	
VCC	VP
SSN	SCL
MISO	GND
MOSI	SDA
GND	IN

USB-TO-I2C HARDWARE TO TARGET CONNECTION

When connecting the USB-to-I2C Hardware to an I^2C target system, a cable with a minimum of three (3) wires will be required. The SDA, SCL, and GND on the USB-to-I2C Hardware must be connected to their corresponding signals on the target.
The USB-to-I2C Hardware has 3.3K□pull-up resistors connected to pin VP on the 14-pin header. If the VP is not connected, you must remove the jumpers at RPU. If the jumpers are left in the circuit, ensure that VP has a voltage source applied, and note that the total I^2C sink current of 3mA is not exceeded when the target pull-ups and USB-to-I2C pull-ups are taken into consideration.
The SDA and SCL lines may be pulled up to a voltage range between 3.3V and 5.5V. It may work at voltages down to 2.5V; however, the noise margins will be lower.



Minimum hardware connection between USB-to-I2C and User Hardware (Jumpers at RPU removed)

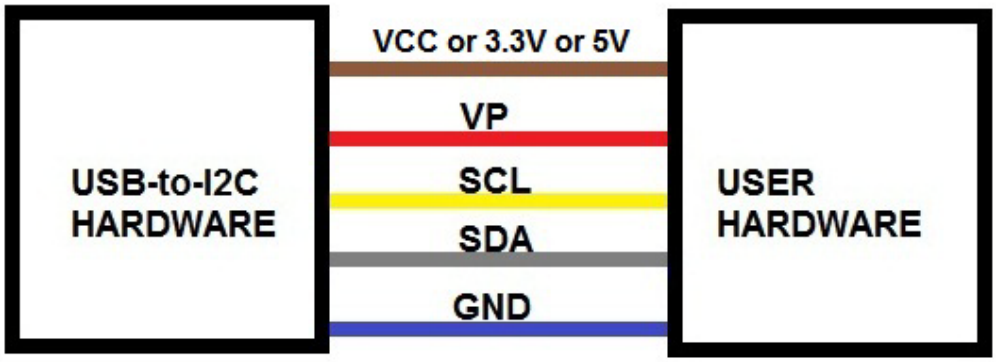


Typical hardware connection between USB-to-I2C and User Hardware (Internal pull-up resistors jumpers at RPU)

USING USB-TO-I2C TO POWER USER'S SYSTEM

When connecting the USB-to-I2C Hardware to the user's hardware, the target system can be self-powered or it may be powered from the VCC, 3.3V or 5V connectors on the USB-to-I2C hardware.
The 3.3V and 5V power outputs can be turned on and off using the USB-to-I2C software or DLL. To turn on the power, select the appropriate item under the Options menu. A check

mark will be shown in the menu beside an enabled power output. The status bar at the bottom of the USB-to-I2C software will also indicate if a power output is enabled. Note that the 3.3V and 5V power outputs are disabled upon hardware reset.
To use the USB-to-I2C hardware in a 3.3V environment, you will need to connect the SCL, SDA, GND and 3.3V (or the VCC pin) to your target system. The pull-up resistors may be left in the circuit or they may be removed by removing the jumpers at RPU. If using the on-board pull-up resistors, you should connect an appropriate voltage to VP.



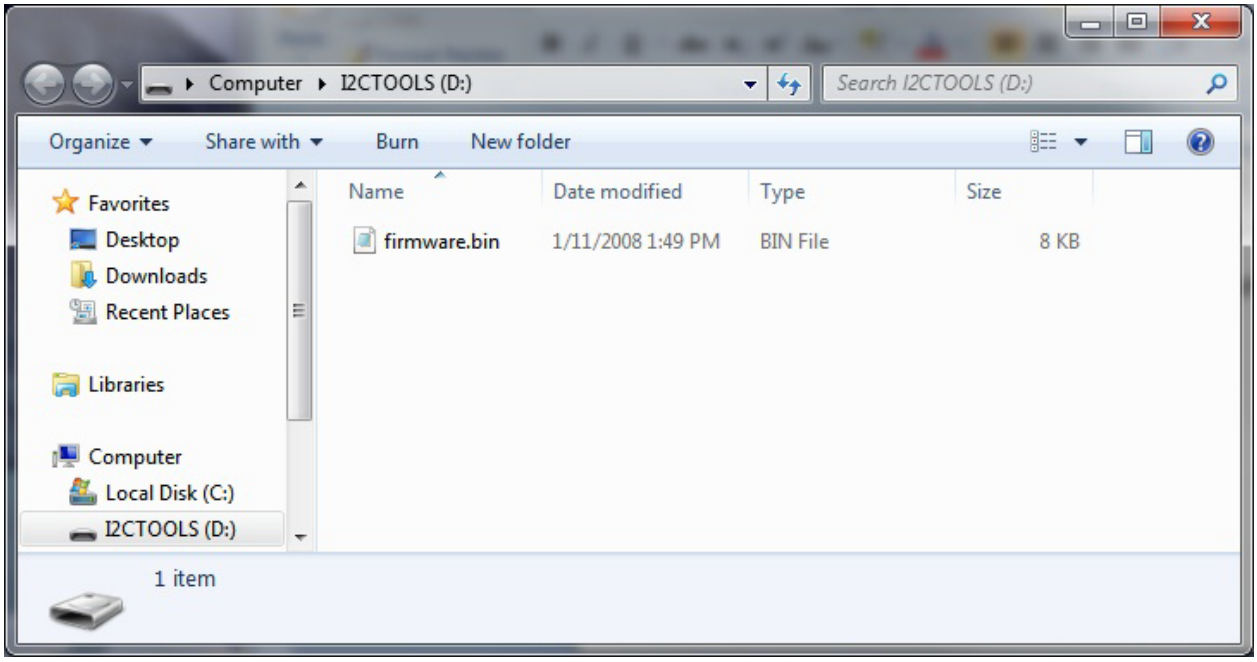
USB-to-I2C Hardware used to power a user application

Important note: the total current consumed by the USB-to-I2C hardware and the target system must be limited to 100mA by the user when using the power outputs. The USB-to-I2C hardware uses about 60mA so the target system can consume a maximum of 40mA.

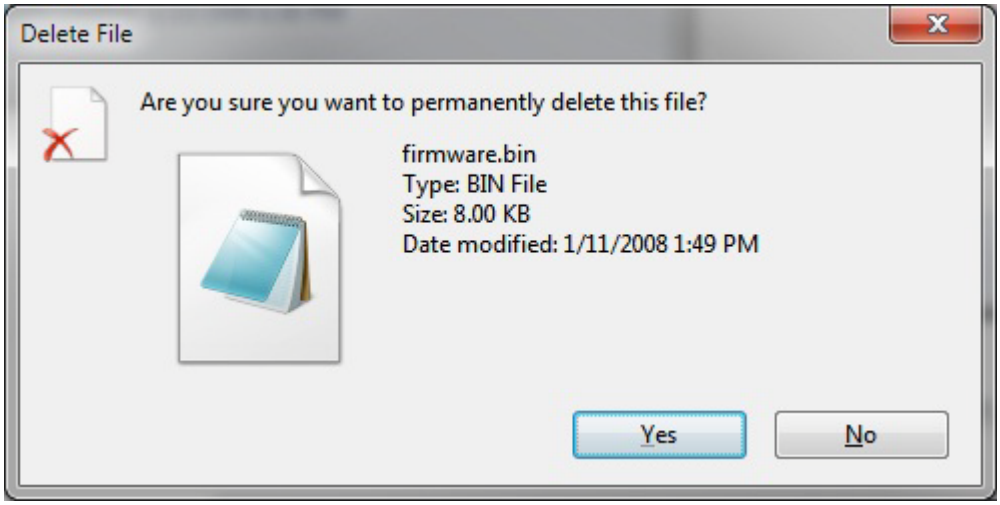
FIRMWARE UPDATES

The firmware in the USB-to-I2C hardware can be updated by the user. No special software is required for the update.
The firmware can be updated by using the following procedure.

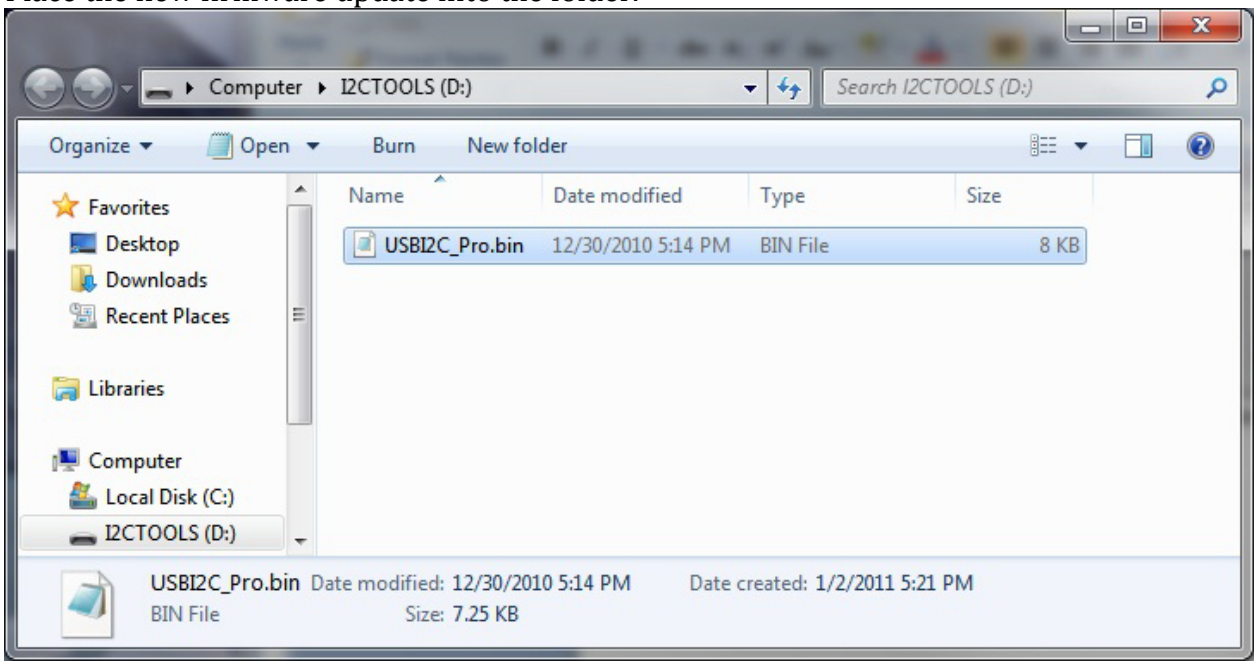
1. Disconnect the USB-to-I2C hardware from the USB port. It should also be disconnected from any user target hardware.
2. Place a jumper across the bottom two pins of jumper JFU (see figure in the Hardware Description section of this document).
3. Plug the USB-to-I2C hardware into the computer's USB port.
4. The USB-to-I2C hardware will be displayed as a mass storage device in Windows, as shown below.



5. Select the firmware.bin file and delete it. You will be asked if you would like to delete the file. Choose Yes.



6. Place the new firmware update into the folder.



7. Disconnect the USB-to-I2C hardware from the USB port.
8. Remove the jumper from JFU.

You can now use the USB-to-I2C hardware with updated firmware. You can check to see which firmware revision is loaded by clicking on About in the Help menu in the USB-to-I2C software.

SOFTWARE UPDATES

The USB-to-I2C software has a built-in update feature which allows you to check for updates whenever your PC has an internet connection. This feature can be found under the Options menu. USB-to-I2C updates can also be downloaded manually from the following website: http://www.i2ctools.com/updates.html
Check our website periodically for update announcements and information.

TECHNICAL SUPPORT

Technical Support for USB-to-I2C is available via an email to support@i2ctools.com.

TROUBLESHOOTING

- If you have problems with the software installation, ensure that you have Admin Privileges.

- If the installer application doesn't launch when the CD-ROM is inserted, then manually browse the CD-ROM and find the Setup.exe file in the root directory of the CD. Double-click on the Setup.exe file to start the installation process.

- The Adapter has 3.3k□ pull-up resistors connected to the I²C bus lines. Ensure that additional pull-up resistors in the connected target system do not bring the pull-up resistance below 1.6k□ for 5V systems or 1.1K□ for 3.3V systems. The on-board pull-up resistors can be removed from the system by removing the pull-up resistor jumpers RPU (see Hardware Description above).

- USB-to-I2C monitors the communications on the I²C bus for proper operation of connected peripherals; any errors on the bus are detected and reported by the software. Bus communication is stopped if errors are detected and can be resumed when the (hardware) problem is corrected and the transmission retried.

- Keep the original USB-to-I2C installation CD-ROM in the event the software needs to be re-installed. Future USB-to-I2C updates from our Website may require a previous installation from the original media.

- If you get a Timeout message when using the software, make sure that there is a pull-up connected to the SDA and SCL pins. The Timeout occurs when either, or both, of the I2C pins are held low for more than 0.5 seconds. If pull-ups are connected, remove the connection to the target system. Reset the hardware by removing the USB cable and reinserting it. Make sure that all three jumpers are connected then try

sending an I2C message. If you still get a Timeout, then microcontroller has probably been damaged. The unit will need to be returned for repair.

If all else fails, email a description of the problem you are having to us at support@i2ctools.com.
Note that all technical support requests must begin with an email to this email address.