Peter Norberg Consulting, Inc.

Professional Solutions to Professional Problems

P.O. Box 10987 Ferguson, MO 63135-0987 (314) 521-8808

Information and Instruction Manual for the **USBToTTL Series of** USB to TTL-Serial Conversion Boards

By
Peter Norberg Consulting, Inc.

Table of Contents

Table of Contents 2
Disclaimer and Revision History 3
Product Warnings 3
    LIFE SUPPORT POLICY 3
Introduction and Product Summary 4
Connecting the USBToTTL 5
Software Installation Under Windows® 7
    Base Driver Installation Under Windows® 7
    Adjusting Default COM port properties for best operation 8
Operation As A Com Port 9
    Quick test by use of SimpleSerial 9
        Configuring our SimpleSerial application in Windows® 10
        Setting up some other emulator 11
        Testing without being connected to any other board 12
        Testing when connected to any of our motor controllers 12
        The most common problems in testing the USBToTTL board with Hyperterminal 12

Disclaimer and Revision History

All of our products are constantly undergoing upgrades and enhancements. Therefore, while this manual is accurate to the best of our knowledge as of its date of publication, it cannot be construed as a commitment that future releases will operate identically to this description. Errors may appear in the documentation; we will correct any mistakes as soon as they are discovered, and we will post the corrections on the web site in a timely manner. Please refer to the specific manual for the version of the hardware and firmware that you have for the most accurate information for your product.

This manual describes the USBToTTL USB to TTL-Serial conversion board, artwork versions 4, 6 and 7.

Product Warnings

Note that the product is not fully protected against static electricity. Its components can be damaged simply by touching the board when you have a "static charge" built up on your body. Such damage is not covered under either the satisfaction guarantee or the product warranty. Please be certain to safely "discharge" yourself before handling any of the boards or components.

LIFE SUPPORT POLICY

Due to the components used in the products (such as those from National Semiconductor Corporation and others), Peter Norberg Consulting, Inc.'s products are not authorized for use in life support devices or systems or in devices that can cause any form of personal injury if a failure occurs.

Note that National Semiconductor states "Life support devices or systems are devices which (a) are intended for surgical implant within the body, or (b) support or sustain life, and in whose failure to perform when properly used in accordance with instructions or use provided in the labeling, can be reasonably expected to result in a significant injury to the user". For a more detailed set of such policies, please contact National Semiconductor Corporation.

Introduction and Product Summary

The Peter Norberg Consulting, Inc. USBToTTL series of serial communications adapter board are tiny (1" x 1.5") boards that allow TTL-Serial devices (such as any of our stepper motor controller boards) to communicate via a standard USB port on a computer. The products is based on the FTDI (Future Technology Device International) FT232 series of chips and uses their USB drivers and COM emulators to allow operation as if the connection were via a standard RS232 serial COM port.

The USBToTTL adapters are USB-bus powered – no additional power supply is needed in order for it to operate. This means that as soon as the adapter is plugged into the USB system (either on your computer or via a hub), it will notify the computer of its presence, and will accept communications with the computer. Note, therefore, that the USBToTTL product may be "up and running" while the final board (one of our motor controllers, a Parallax Basic Stamp, etc.) has not yet been powered.

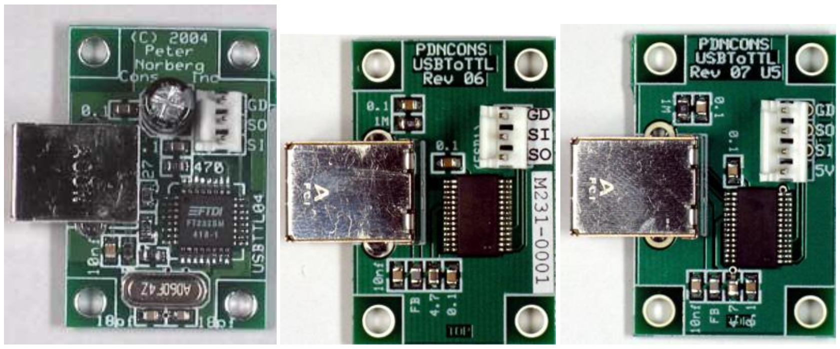
The USBToTTLEU version of the adapter adds ESD protection and a 5 volt tap on the USB power. The 5 volts is present only while the computer is "awake" – when it is turned off or in some form of sleep mode, the 5 volt output from the board is turned off. Your product may draw up to 400 mA, assuming that your USB port supports that much current (some non-powered ports can be limited to just 100 mA, which means that your device might be limited to 80 mA). Please note that the ESD protection is only active while the board is attached to a properly grounded USB cable (i.e., a computer that is correctly grounded), and only protects against spikes coming in at the connectors. It does not protect against spikes cause by directly touching components or traces on the board itself.

The boards support communication rates of up to 3 megabaud; however, when used with our stepper motor controllers, you must operate at whatever baud rate was specified for that particular product (usually, this is 9600 baud, due to the design specifications of those products). If the board is connected to some other TTL-Serial compatible device, its full communication rate may be available, depending on the limitations of the device to which it is being connected.

From a connections point of view, there are exactly 2 connectors on the board. One is a USB-B connector, to be used to connect to the USB system. The other is a 3 or 4-wire connector (either a screw-terminal or an MTA-100 SIP header), which is to be connected to your target board. The 3 signals transferred are Serial Input (to the USBToTTL board), Serial Output (from the USBToTTL board), and ground (so that the systems are correctly referenced to each other). No other standard serial signals are provided (such as CTS, RTS, DSR, etc.); this reduces the cost and complexity of the connections. You should refer to the following manual section ("Connecting the USBToTTL") for more information about the connections.

Connecting the USBToTTL

The USBToTTL board appears as follows:



The image on the left shows the revision 4 board, the one in the center shows the revision 6 board, and the one on the right shows the revision 7 (with USB 5 volt output). All show the MTA-100 connector option.

The large square connector (shown on the left in the photos) is for the USB-B side of the USB cable that you provide. As soon as this is plugged into the cable, and the cable is plugged into the computer, then the board is powered and ready for action. Therefore, ***you need to mount the board before you connect it, as it will otherwise be "live" (powered), and you could damage your computer (or the board) by handling it while activated. For example, laying the board on top of a metal plate (or screwdriver) will probably destroy the board (and possibly your computer), if it is powered. This type of failure is not warranted!***

The board as shown (not including the overhang from the USB connector) is 1 inch wide by 1.5 inches tall. The mounting holes are 0.125" in diameter, positioned in 0.125" from each corner. This size hole can accept up to a number 5 screw; this means that the standard #4 mounting posts work quite well. The screw hole positions are therefore:

0.125, 0.125	0.875, 0.125
0.125, 1.375	0.875, 1.375

The signals on the remaining connector are to be attached to your motor controller board (or your Parallax Basic or Javalin Stamp, etc.). The signals are defined as:

Signal	Description
GD	Ground reference. This must be connected to the ground on your controller board; otherwise, the TTL signals may not have a correct reference for operation.
SO	Serial Output FROM the USBToTTL board TO your controller (usually, this is attached to the SI line for our boards)
SI	Serial Input TO the USBToTTL board FROM your controller (usually, this is attached to the SO line for our boards)
5V	Present only on the USBToTTLEU version of the board, this is a 400 mA 5 volt tap into the USB bus power (assuming your USB port can supply it), for your use on your project. It is automatically switched off when the computer goes into suspend, to conserve power.

As noted in the above table, you need to have a good ground reference for the system to work correctly. The "GD" signal on the USBToTTL board is connected to the ground wire of the USB bus and also ***must*** be attached to the ground reference for the board to which it is attached (such as one of our motor controller boards). Failure to connect this line can result in inability to communicate and also can cause failure of the USBToTTL board.

The SO line is the serial output from the USBToTTL board to the motor controller (or other board). It is "logically connected" to the serial output line from the computer; thus, it needs to be connected to the Serial Input ("SI") line of the motor controller in order for the controller to see the serial data.

Similarly, the SI line is the serial input to the USBToTTL board from the motor controller. It is "logically connected" to the serial input line to the computer; therefore, it needs to be connected to the Serial Output ("SO") line of the motor controller in order for the computer to see the data from said controller.

Please also note that the **positions of the SI and SO pins are reversed** on the revision 6 artwork relative to the other two artworks.

The signal levels to and from the USBToTTL board are standard TTL logic levels. This means that the board's worst-case response is:

High signal: Minimum 3.2 volts, maximum 4.9 volts

Low signal: Minimum 0.3 volts, maximum 0.6 volts

Input switching threshold: 1.3 to 1.9 volts, with about 50 mV of hysteresis

This is fully compatible with all of our board level products and should be compatible with most TTL-serial devices.

When connecting the USBToTTL board to any of our products which have a "**JS**" jumper (which is most of them), the **JS** jumper MUST be removed in order for the USBToTTL board to be able to "talk" to the motor controller board. Some of our latest products use an automatic sensing technique which removes the need for this jumper – those models therefore no longer have that jumper present on the artwork.

Simlarly, for our (much) older boards which have a socketed MAX232 serial chip (such as the BiStepA04 or the SimStepA04), the serial chip must be removed (same effect as removing the JS jumper on the later boards). Otherwise, your connection to the motor controller is:

USBToTTL Signal	**Motor Controller Signal**	**Description**
GD	GND	Common Ground
SO	SI	Serial Data from USBToTTL to Motor Controller
SI	SO	Serial Data from Motor Controller to USBToTTL

Observe that the "SO" signal from each board is connected to the "SI" of the other board. This is because the signal names are always relative to the given board; therefore, "Serial Input" to one board must be connected to the "Serial Output" of the other board.

Software Installation Under Windows®

FTDI (http://www.ftdichip.com) provides drivers for operation under Windows®, Linux, and Mac/OS. Our installation disk includes snapshot copies of their Windows® drivers; however, you instead may prefer to go to their web site to obtain the most up-to-date revisions of their code. All Linux and Mac/OS customers must download their drivers directly from the http://www.ftdichip.com site, as we have no support capability for those platforms. Look for the drivers and documentation that relate to their FT232 series of devices.

A short summary of the installation of the drivers under Windows follows. For installation under Linux or on the Mac, please refer to the FTDI documentation available from their web site.

Base Driver Installation Under Windows®

Installation of the drivers under *Microsoft® Windows®* is fairly straightforward (and is well documented by FTDI). You should review the FTDI documentation (specifically, their "Windows Driver and Installation Guide" manual) for a more comprehensive description of the installation process. We install a snapshot version of this manual as part of our installation process, but you may wish to obtain a more up-to-date version directly from their web site.

A short summary of the procedure follows, along with a description of the adjustments that should be made to the COM emulator port settings after installation has been completed.

1. Thanks to the "magic" of "Plug-N-Play", simply connect the USBToTTL board to your computer (use a normal USB A-B cable of the appropriate length, connecting the 'A' side to your computer USB slot, and the 'B' side to our board). ***Make certain that the USBToTTL board is NOT on any sort of conductive surface (for example, metal, your hand, a carpet) when you do this, since you could damage the board (or your computer!) if any of the signals on the board get shorted.*** Note that you do NOT need to have the board connected to any external product (such as one of our motor controllers) to install the drivers: just the USBToTTL board and a USB A-B cable are needed, in addition to your computer with its USB 1.1 or 2.0 connection.
2. This will cause Windows to bring up their "Found New Hardware" wizard, which will guide you through the installation process. Note that under *Microsoft® Windows®* Vista®, you may not be given the option to do this installation. If this happens to you, then you will have to go to the next section ("Adjusting Default COM port properties") to verify that your port has been correctly configured.
3. Place our installation CD into your CD drive.
4. If our setup application starts up, cancel out of it
5. Tell the wizard to "search for a suitable driver", and then tell it to "specify a location".
6. It will then ask for where to search: tell it to look in the "FtdiStepperBoard" directory on our support CD.
7. Then tell it to install the driver. If you are installing from the 'FtdiStepperBoard' version of the drivers, then Windows will complain that the drivers are not 'Windows Certified'. You may ignore the error; all that is different between the 'ftdi' certified installation and the 'FtdiStepperBoard' non-certified installation is that the installation script sets the communication defaults to our recommended values (below).
8. The installation may then go through the same process in order to install the virtual COM drivers (if you have never installed an FTDI USB-based product before). Use the same subdirectory and process to install those drivers as were used under step 7, above.
9. Once that process completes, the code will automatically add a new "COM" serial port which is "attached" to the board when it is plugged into the ***same*** USB port on your computer. *The system will automatically add a new COM port each time you attach the USBToTTL board to a different USB port on your computer or hub.*

Adjusting Default COM port properties for best operation

Once the system has created the COM port for the board, you may need to change the system defaults to match the requirements of our motor controllers. If you installed from the 'FtdiStepperBoard' subdirectory, then these changes will normally have been done for you. Otherwise, you will need to perform the following procedure:

1. Under Windows 2000 or XP, go to your "system properties" page. Do this by
    a. Right-click on your "System" icon
    b. Select "Properties"
    c. Select "Hardware devices" (it might just be called "Hardware")
    d. Select "Device Manager"
2. Under Windows Vista, log in as an administrator, and then get to your "device manager" page by:
    a. Go to your 'Start' menu, and click on the 'Computer' button
    b. On the ribbon that appears at the top of the resulting window, click on "System Properties"
    c. On the task pane on the left of the new window, click on "Device Manager"
    d. The system will ask for your permission to continue. Press the "Continue" button.
3. Look under "Ports (COM and LPT)", and select the COM port which you just added (it will normally be the highest-numbered port on the system, such as "COM6"), and edit its properties.
4. Reset the default communication rate to:
    a. 9600 Baud,
    b. No Parity,
    c. 1 Stop Bit,
    d. 8 Data Bits,
    e. and No Handshake
5. Select the "Advanced Properties" page, and set the:
    a. Read and Write buffer sizes to 64 (from their default of 4096).
    b. Latency Timer to 1 millisecond
    c. Minimum Read Timeout to 0
    d. Minimum Write Timeout to 0
    e. Serial Enumerator to checked
    f. Serial Printer to unchecked
    g. Cancel If Power Off to unchecked
    h. Event On Surprise Removal to unchecked
    i. Set RTS On Close to unchecked
    j. (that is to say, only the Serial Enumerator is checked in the set of check boxes on the display)

Operation As A Com Port

Once you have performed the appropriate installation (above), then you can communicate with our motor controller board (or whatever board you have connected to the USBToTTL product) via use of the Windows serial system, as a "COM" port. From the point of view of your application, the target board (our motor controller) will appear to be available on a COM port, operating at whatever rate is supported by the target board.

The USBToTTL board supports up to 3 megabaud as a communication rate; however, our standard motor controllers normally all operate at 9600 baud – therefore, you should "set up" your communications program to operate at 9600 baud, no parity, 8 data bits, one stop bit, and no handshake (as was done as part of the driver installation, above).

Note that if you are "talking" to some other TTL-Serial product with a higher communication rate than our standard motor controllers, you will want to adjust the "properties" of the com port to match the requirements of your particular product. You may also need to change the buffer size back to the default of 4096, if you are communicating at a very high rate (in order to avoid data loss).

Quick test by use of SimpleSerial

The easiest way to test the connection to one of our boards is through use of our SimpleSerial application (or some equivalent terminal emulator if you are not using the *Microsoft*® *Windows*® operating system). Load SimpleSerial, and tell it to connect to the COM port that was just added.

If you are not using *Microsoft*® *Windows*®, use whatever serial-communication-capable terminal emulator available on your system that you prefer as your interactive test tool. (This document cannot provide direct assistance for other operating systems.)

Configuring our SimpleSerial application in Windows®

Configure SimpleSerial as follows:

a. Load "SimpleSerial". On most installations of our system, you will find this under "Start", "Programs", "Stepperboard", "SimpleSerial".

b. When Simple first loads (the first time on a new installation), it will automatically search for a "StepperBoard" motor controller product. If it does not find one, then it will automatically search for the lowest-numbered COM port, and will attach to that at 9600 Baud.

c. If it has not selected the correct port for the USBToTTL converter, then manually select the correct port using the the "File->Open Specific COM Port" menu option (press the 'File' menu text, and then the "Open Specific COM port" menu text).

d. It will next ask for a COM port and a baud rate.

    i. Select the COM port connected to our controller board. If you have multiple COM ports, you may have to read your computer's manual to determine which is connected to the controller; if this is the first time that you have connected the USBToTTL board to your computer, then it will normally be assigned the highest COM port number which you can find.

    ii. For a self-test, select a Baud rate of 9600 baud.

    iii. If you are using some other terminal application, the full settings for a self-test would be:

        - 9600 baud
        - 8 data bits
        - no parity ("none")
        - 1 stop bit
        - flow control off ("none")
        - Local echo on (so you can see what you are typing)

e. After you have completed the port settings page, SimpleSerial will go to its emulation page (an empty window waiting for serial input data or for text which you type). Anything that you type will be sent to the COM port, and any data received will be displayed in the edit window.

Setting up some other emulator

Set your terminal emulator to:

- 9600 Baud
- No parity
- 8 data bits
- 1 stop bit
- No flow control (i.e., no automatic hardware or software handshake is used to control transmission or reception of serial data)
- Terminal emulation should be TTY (i.e., characters are purely displayed; there are no motion keys)
- The emulator should be set to echo typed characters locally; otherwise, you will not see what you type.

For information on how to adjust these settings, refer to your terminal emulator's documentation.

Testing without being connected to any other board

You can easily test the USBToTTL board by simply jumpering its SO line back to its SI line, and then sending serial data to it. Everything that you send will be echoed back.

Under SimpleSerial, this method will make all of the characters that you type appear twice on the screen: for example, typing "hello" would cause the screen in SimpleSerial to show "hheelloo", assuming that the USBToTTL board is operating.

Testing when connected to any of our motor controllers

Having set up the communications parameters, turn on power to our motor controller board (which has already been connected to the USBToTTL board as described previously under Connecting the USBToTTL). Our copyright message should appear on the SimpleSerial screen if everything is connected correctly.

All of our motor controllers will also respond to the command "-12?" by displaying another copy of the copyright message.

The most common problems in testing the USBToTTL board with Hyperterminal

The single most common problem when testing the USBToTTL board is forgetting to disable flow control (i.e., no automatic hardware or software handshake is used to control transmission or reception of serial data). If the handshake is left on, then Hyperterminal will actually never sent a character – it will be waiting for the USBToTTL board to report that the CTS signal is "ready", and it never will attain that status.

Additionally, if you forget to turn on the "echo characters typed locally" check box, then you will only see what is sent to you by the board. You will not see what you sent, which can be confusing.