Isolated USB UART User Guide

Release 4

Partially Stapled

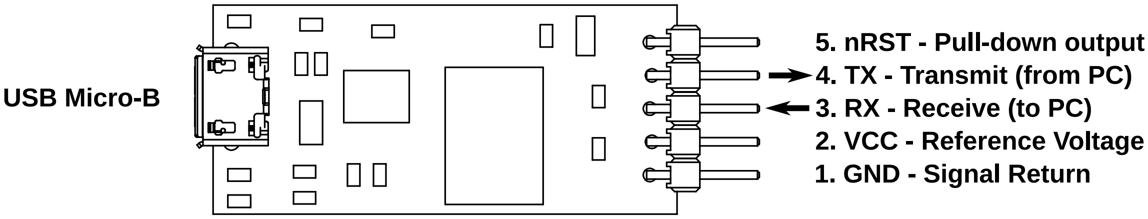
September 29, 2014

CONTENTS

CHAPTER

ONE

SPECIFICATIONS


USB Micro-B
5. nRST - Pull-down output
4. TX - Transmit (from PC)
3. RX - Receive (to PC)
2. VCC - Reference Voltage
1. GND - Signal Return

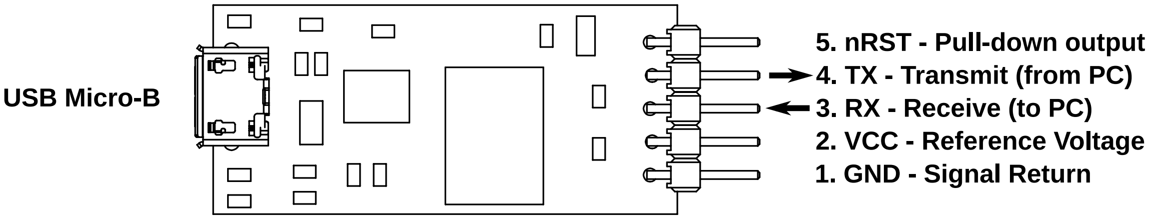
The Isolated USB UART is a super useful, super safe way to talk logic-level serial to all sorts of projects. It is compatible with most logic standards (see below for details), and is galvanically isolated to keep your PC safe and to avoid accidentally creating ground loops.

This device is Open Source Hardware[1]. Design files and user documentation are provided under the Creative Commons Attribution 3.0 License[2].

- 1MBit data rate UART with transmit and receive
- USB 2.0 compatible, with Micro-B type connector
- 1000 volts RMS isolation
- Logic voltage of 2.5-5.5 volts
- Bonus pull-down transistor, useful for e.g. resetting a microcontroller to initiate a bootloader
- LEDs indicate USB connection, and transmit and receive activity
- Windows, Linux, and Mac comptaible – uses FTDI's FT230X chip with drivers readily available
- Configurable - Disable LEDs or make them toggle or blink, invert UART polarity, and customize VID/PID and device strings
- RoHS-compatible and lead-free
- Open Source Hardware - design documents provided under the Creative Commons Attribution 3.0 License[3], plus gerbers and schematic PDF

[1] http://www.oshwa.org/definition/
[2] https://creativecommons.org/licenses/by/3.0/
[3] https://creativecommons.org/licenses/by/3.0/

USING THE UART


USB Micro-B
5. nRST - Pull-down output
4. TX - Transmit (from PC)
3. RX - Receive (to PC)
2. VCC - Reference Voltage
1. GND - Signal Return

2.1 Quick Start

1. Connect your project's logic-level serial pins to the UART's header pins using jumper wire, ribbon cable, socket, or other means.
2. You must at minimum connect the GND and VCC pins. Connect GND to your circuit's ground, and connect VCC to the supply rail that matches the logic level you wish to use. For example, if your project's microcontroller runs at 5V then connect the UART's VCC pin to the 5V supply. Supply voltages of 2.5 - 5.5V are supported.
3. Connect the TX and/or RX pins to their counterpart on your project. Depending on how the target circuit is labeled, you may need to either wire TX to RX and vice versa, or you might need to wire TX to TX and RX to RX. The labels on the UART are from the perspective of the PC – TX is an output, and RX is an input.
4. Finally, connect the UART to your PC using a USB cable with a Micro-B USB cable.
5. You may need to install the FTDI Virtual COM Port drivers before you can use the UART. See *Resources* for download links. If you did install drivers, unplug and re-plug the USB cable afterwards to ensure it is detected. No additional drivers are required on Linux.
6. The UART is now available as a regular COM port. See *below* for some tips on terminal emulator software for communicating with serial devices.

2.2 Using the nRST Output

The nRST is an "open drain" output that can be used to pull down a reset line. It can pull its output low or float but cannot drive it high, which means it will not interfere with other reset sources on the target board.

This is ideal for use with a bootloader. For example, if your microcontroller has a serial bootloader that runs on power-up, you can use the reset output to programmatically trigger a reset then load new software into the target.

The reset function is implemented with a N-channel MOSFET connected to the FTDI's "CBUS0" pin. Setting CBUS0 to logic high will pull nRST low, and setting CBUS0 to logic low will let nRST float. The CBUS0 function is not accessible through the Virtual COM Port interface. In order to control it, you will need to use a FTDI-specific library or tool. For more information about how to access FTDI functions from your application, see FTDI's code examples[1].

2.3 Reconfiguring

FTDI supplies a tool called FT_Prog that can be used to reprogram various aspects of the UART's behavior from a Windows PC. For example, you can change how the LEDs react to activity or disable them, invert the serial lines, or set a custom product description. See *Resources* for a download link.

2.4 Terminal Emulators

A typical use of the UART is to connect to a command-line terminal interface. To do this you will need terminal emulator software. On Windows, PuTTY is recommended. On Linux and BSD "screen" works well.

2.4.1 Windows - using PuTTY

Download PuTTY[2], either the standalone executable `putty.exe` (no install required), or the full installer `putty-installer.exe`. After starting PuTTY, a window titled "PuTTY Configuration" will appear. In the right-side pane, click "Serial". In the text box under "Serial line", type the name of the COM port. You can find a list of COM ports in the Windows Device Manager, where the UART will be named "USB Serial Port" with the COM port name following it in parentheses. In the "Speed" text box type the baud rate. Click "Open". See below for instructions on how to use the command-line. When you are finished, close the window.

2.4.2 Linux/BSD - using screen

screen can be used to interact with a serial port with minimal fuss. To start a screen session with a serial port, or to open a new window from within an existing screen session, type:

```
screen /dev/ttyUSB0 115200
```

Where `/dev/ttyUSB0` is the device that appears when you connected the UART, and 115200 is an example baud rate (actual baud rate will depend on your target device). Consult `dmesg` or `/var/log/messages` or `journalctl` for logging that indicates which serial port was assigned. When you are finished, type `Ctrl-a`, then `k` (for "kill"), then `y` to confirm.

[1] http://www.ftdichip.com/Support/SoftwareExamples/CodeExamples.htm

[2] http://www.chiark.greenend.org.uk/ sgtatham/putty/download.html

CHAPTER

THREE

RESOURCES

3.1 Buy Isolated USB UART

You can buy Isolated USB UART at Tindie[1]

3.2 Device Drivers

FTDI Virtual COM Port Driver
http://www.ftdichip.com/Drivers/VCP.htm
It is recommended to download the setup executable, not the ZIP bundle, linked on the right side of the drivers page.

FTDI Installation Guides
http://www.ftdichip.com/Support/Documents/InstallGuides.htm

FT_Prog utility for reconfiguring
http://www.ftdichip.com/Support/Utilities.htm

3.3 Source code and Design files

Hardware design files and source code for the documentation are maintained in a Mercurial repository:
http://hg.partiallystapled.com/circuits/usbjumper

If you just want a ZIP file to download, you can find them here:
http://hg.partiallystapled.com/circuits/usbjumper/archive/tip.zip

[1] https://www.tindie.com/products/gxti/isolated-usb-uart/