Serious about science: Serious about timing

The Black Box ToolKit

Application Note

Using the BBTK USB TTL Module on a Mac or Linux System

20171011

Installing VCP Drivers

The USB TTL Module uses an industry standard FTDI chip combined with our own custom SOC and firmware to convert serial signals to TTL and vice-versa.

Before you can use the BBTK TTL Module on your system you will need to install a set of drivers from FTDI for your operating system from the link below:

http://www.ftdichip.com/Drivers/VCP.htm

This will provide a Virtual COM Port or VCP via which you can communicate with the module.

For installation instructions consult the documentation supplied by FTDI.

Serious about science: Serious about timing

The Black Box ToolKit

Checking Operation of the new VCP

The following instructions are for Mac OSX but will be similar on Linux.

Plug in TTL Module

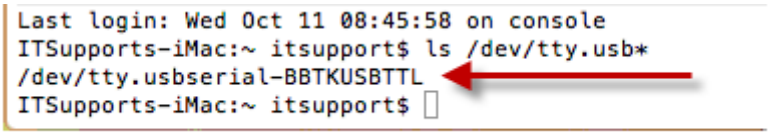
a. Open a terminal window (BASH). Check which serial ports are available after installing the correct FTDI Virtual Com Port (VCP) driver for your Mac.

Type:

ls /dev/tty.usb*

Press return and press enter.

The name of the VCP should then be displayed as shown below:



In this case, tty.usbserial-BBTKUSBTTL.

b. Next you will need to set the correct serial speed and name your terminal

screen /dev/tty.usbserial-BBTKUSBTTL 115200

Note: The terminal screen will go blank and appear unresponsive

c. Type RR to reset the module - this won't be echoed back. DO NOT PRESS ENTER after typing commands. It is your responsibility to ensure the Module remains in sync. If commands do not appear to work correctly then you should send RR to the Module to reset it.

NOTE: No serial commands you type will be echoed back to the terminal display.

d. Type 01 to test the module - this should make TTL 01 high and stay on. The red LED on the rear of the module will illuminate to confirm that there is an output on pin 1.

e. Type 00 to clear all TTL lines, i.e. pin 1 in this example.

f. To get the firmware version of the module type VV. The firmware version should be echoed back as two bytes, e.g. 01

Serious about science: Serious about timing

The Black Box ToolKit

Checking Latency Timing Using Your own Software on Your own Mac

The BBTK USB Module has an inbuilt timer to check how quickly your software/Mac can communicate with it. To make use of this inbuilt timing audit you need to send two serial commands one after another in your experimental software. The Module will then return a raw round trip time from which you can work out the number of elapsed milliseconds. On a PC this should be under 1mS.

a. Send RR to reset the Module

b. Set your software to send the following two characters to start the round trip timer, >>

c. Immediately after sending >> set your software to send, <<. Note that these two sets of commands should be adjacent with no other processing being done other than sending of the two pairs of characters or bytes.

d. After sending << the Module will respond with a raw elapsed time, e.g. 23500. After sending << to the module you should read the raw elapsed time into your software.

e. To turn this raw elapsed time into milliseconds you will need to apply the following scaling factor, 13.832 as follows where 1000 is a constant:

(23500 * 13.832) / 1000

= 325.052mS

in this example.

NOTE: We cannot provide support for any software you may running and ensuring that the BBTK USB TTL Module is performing accurately and reliably is your sole responsibility.

If the round trip time is consistently greater than 1mS then we would suggest you do not use the BBTK USB Module for timing critical applications. We cannot be held responsible for any mistimings howsoever caused.

If the round trip time above is greater than 1mS you are advised to consult the FTDI documentation for how to speed up serial port access and rerun the test (see below).

Serious about science: Serious about timing

The Black Box ToolKit

How to set FTDI Driver Latency to be as low as Possible

To use the USB TTL Module in timing critical applications you will need to ensure you run the latency tests above. On this PC this should be done using our configuration app detailed in the printed manual.

On other systems you will need to manually set the driver latency manually. Details of how to do this are provided below for reference. Note The Black Box ToolKit Ltd cannot be held responsible for any mistimings howsoever caused and this information is provided as is and not warranted as being fit for purpose.

OS X

OS X does things differently. The driver bundle contains a file,

/System/Library/Extensions/FTDIUSBSerialDriver.kext/Contents/Info.plist

This XML plist file describes different profiles for the serial port, including different LatencyTimer values, depending on how the FTDI identifies itself on the USB bus. FTDI's own Technical Note on the subject explains how to edit that value to change the latency.

http://www.ftdichip.com/Support/Documents/TechnicalNotes/TN_105%20Adding%20Support%20for%20New%20FTDI%20Devices%20to%20Mac%20Driver.pdf

Search "Technical Note TN_105".
On OS X the latency timer defaults to 2ms for any FTDI FT232 that uses the default vendor & device USB IDs (0403:6001), i.e. the BBTK USB TTL Module.

Linux

In proper Linux style, the kernel's FTDI driver exposes a nice sysfs interface that lets you get and set the latency timer. For example, if your serial port is ttyUSB0:

```
# cat /sys/bus/usb-serial/devices/ttyUSB0/latency_timer
16
# echo 1 > /sys/bus/usb-serial/devices/ttyUSB0/latency_timer
# cat /sys/bus/usb-serial/devices/ttyUSB0/latency_timer
1
```

… that will lower the timer from 16ms to 1ms (the minimum), to reduce latency.

In my experience, the timer value won't change immediately on an open serial port. If an application is using it then you'll need to close and reopen it before the new value takes effect.

If you're writing code, there is also a Linux-specific serial flag ASYNC_LOW_LATENCY that programmatically sets the latency timer down to 1ms.

Serious about science: Serious about timing

The Black Box ToolKit

Windows

FTDI's own driver for Windows has a combo box in the Port Settings dialog that lets you choose the latency timer value. Follow our printed manual for details on how to set this and also check latency using our configuration app.