



Device: PLT-1003

This document Version: 2.0

Matches hardware Version: 4

Date: 12 May 2012

Description: USB to Serial interface and USB development platform

## Table of Contents

Introduction.....	3
Features.....	3
Hackability.....	3
Firmware.....	3
Construction.....	3
Connections.....	4
Power.....	4
Buttons.....	4
Indicators.....	4
Schematic.....	5
Microcontroller.....	5
Connections.....	6
PCB.....	8
Installation.....	8
Changing the COM port number.....	14
Versions.....	16

## Introduction

---

The PLT-1003 is a Microchip PIC18F14K50 based platform that works out of the box as a USB to TTL serial converter. It can be used for learning about how USB works and creating other USB interfaces.

## Features

---

The PLT-1003 features PIC18F14K50 clocked at 48Mhz (PLL enabled up to 12mips).

With 16k of flash and 768 bytes of RAM, the 18F14K50 has enough space to get some interesting jobs done while being price competitive with dedicated USB to serial chips. All pins are broken out for easy access and an ICSP programming socket is provided (although all modules ship with Boostbloder and USB to TTL serial firmware installed).

## Hackability

---

The PLT-1003 is 100% hackable.

At Embedded Adventures, we believe you have the most fun when you have the most control over your hardware. For the PLT-1003 we provide a datasheet, complete schematic and complete source code. After that, it's all up to you. We'd love to hear about the projects you're using it for – send us information and photos to [myproject@embeddedadventures.com](mailto:myproject@embeddedadventures.com)

## Firmware

---

Firmware has been written using the SourceBoost compiler. It is available in the PicPack library under Projects\ea\_plt1003v4\_usb2serial along with a number of other firmware alternatives for the PLT-1003 (for example, you can connect a real time clock module and a 7 segment display module and make a clock).

## Construction

---

It's pre-built! All you need do is solder male or female 0.1" headers to suit your project, or solder wires directly.

## Connections

---

The PLT-1003 has four connection ports.

SERIAL	TTL level serial port
ICSP	Programming port for PIC
EXPANSION1	Breakout of pins RC0 to RC7
EXPANSION2	Breakout of pins RA0 to RA3, GND, RB4 to RB7, VCC and GND

## Power

---

The PLT-1003 happily slurps power from the USB port and can provide either 3.3v or 5v depending on the switch setting.

## Buttons

---

The PLT-1003 has two buttons.

B1 is connected to RC1 and B2 is connected to RC0. Both include a capacitor to reduce de-bounce noise, and a pull up resistor. To check for a button press, set the appropriate pin as an input and look for it to go to 0 (press) and then back to 1 (release). RC0 and RC1 pins are also available as interrupt pins.

## Indicators

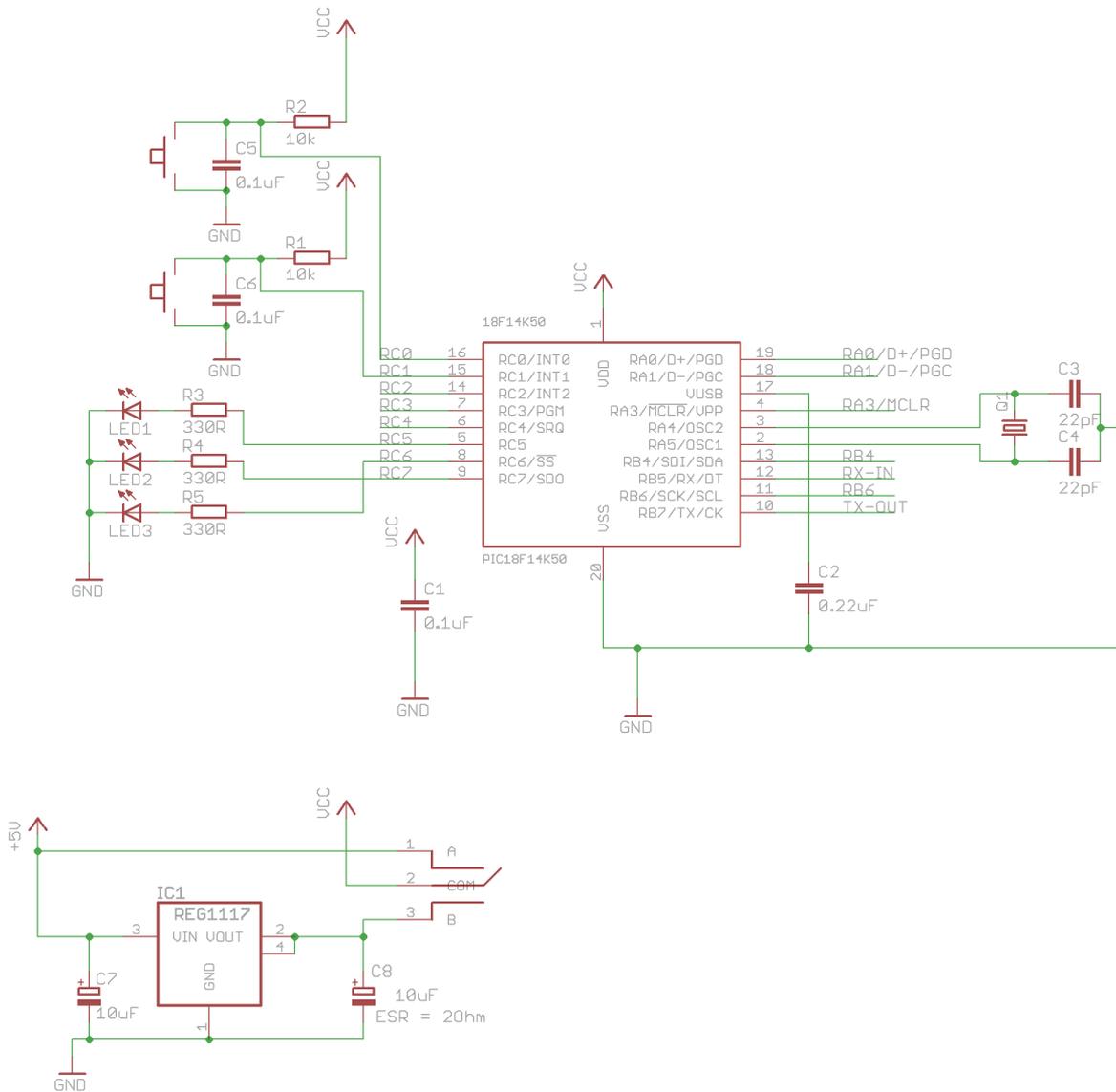
---

The PLT-1002 has three LEDs connected to RC5, RC6 and RC7 (LED1, LED2 and LED3 respectively). LED 1 (red) will light briefly at boot up during Boostbloder initialisation.

You can use these for your own nefarious purposes, although predictably the default firmware uses LED2 (green) and LED3 (yellow) for Rx and Tx.

## Schematic

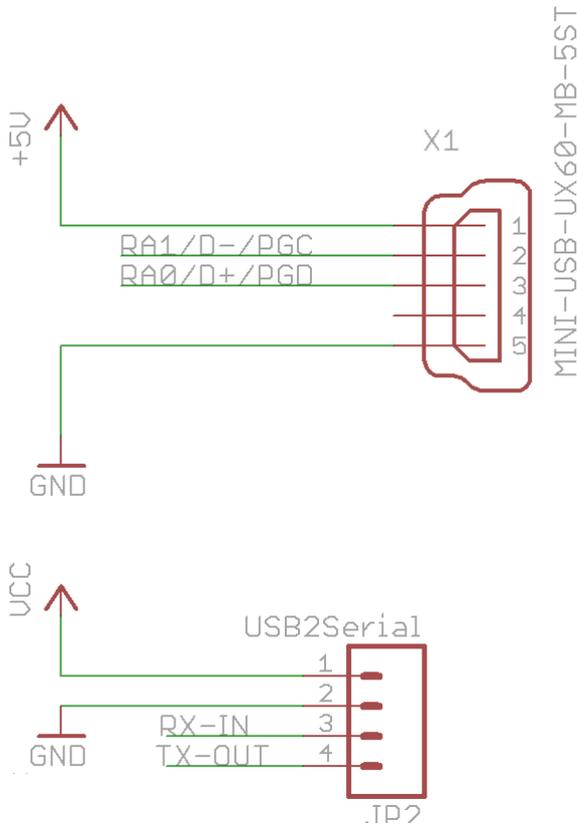
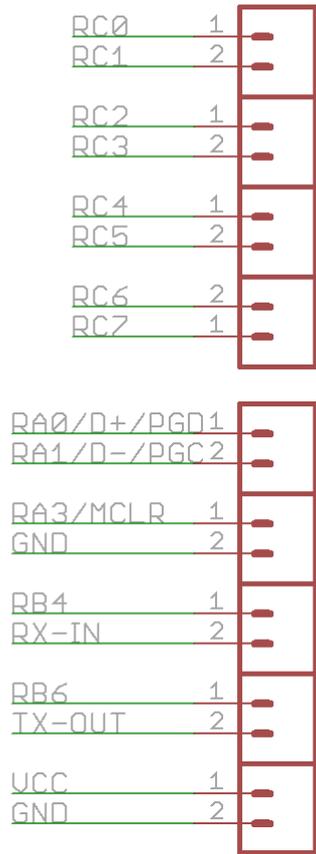
### Microcontroller



C3, C4 and 12Mhz crystal allow the PIC to run at 48Mhz (via the PLL), or 12mips.

C1 is a power supply bypass for the PIC.

C2 is required for USB operation (USB data actually runs at 3.3v, despite its 5v power provision).



## Connections

The Expansion ports break all pins from the microcontroller except for the pins used for the crystal. These ports allow communication with external devices such as humidity or pressure sensors to expand the capability of the board, or to provide other serial pin outs such as RTS/DTS and DCD/DTR.

You have access to:

### RC0 – RC7

RC0 and RC1 are also connected to the push buttons

RC5, RC6 and RC7 are also connected to the LEDs

### RA0 – RA3

RA0 and RA1 are normally connected to the USB port

### RB4 – RB7

RB5 is also RX for the TTL serial port

RB7 is also TX for the TTL serial port

### VCC, GND

Power and several ground connections are also made available.

### USB

USB is provided as standard mini-USB.

## Serial Port

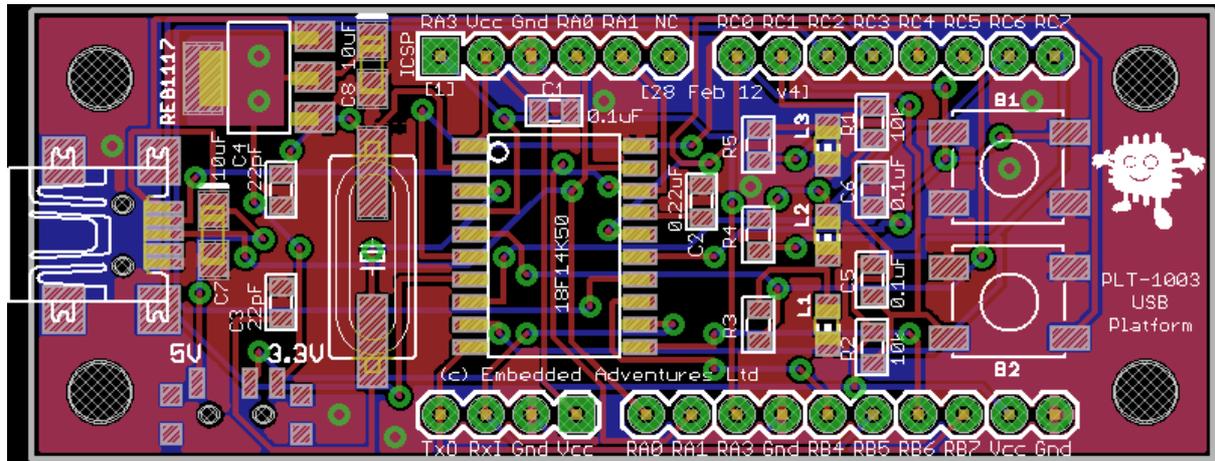
The Serial port provides TX-O (out from the board) and RX-I (in to the board), along with ground and VCC connections if required. Note that if you are connecting a

serial connection to the board, you will typically connect TX on the PLT-1003 to RX on your external serial connection, as well as TX on the PLT-1003 to RX externally. That is, you swap TX and RX between the PLT-1003 and another device. That other device may even be another PLT-1003 (yes, you can reprogram a PLT-1003 using another one!)

The ICSP port allows access to PGC (RA1), PGD (RA0) and MCLR (RA3). If you're not actually using the USB connection, these pins are available for other uses. Due to the fact these pins are *also* shared with USB D+/D- it's important to unplug your ICSP programmer, if you use one, while using USB.

ICSP allows you to program the 18F14K50 using the Pickit or ICD compilers from Microchip. Typically, this is only necessary in order to get a bootloader on the microcontroller, and the Boostloader is supplied preloaded on all modules.

## PCB



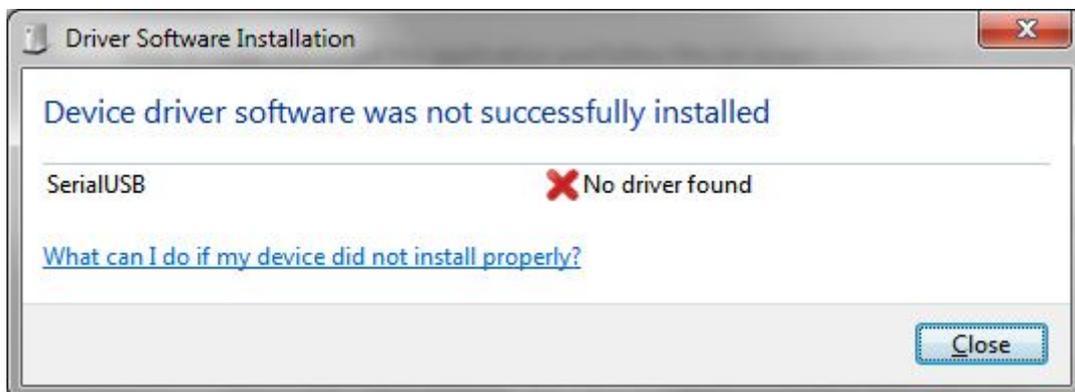
The board is provided pre-assembled, and programmed with boot loader and USB 2 TTL Serial functionality installed.

## Installation

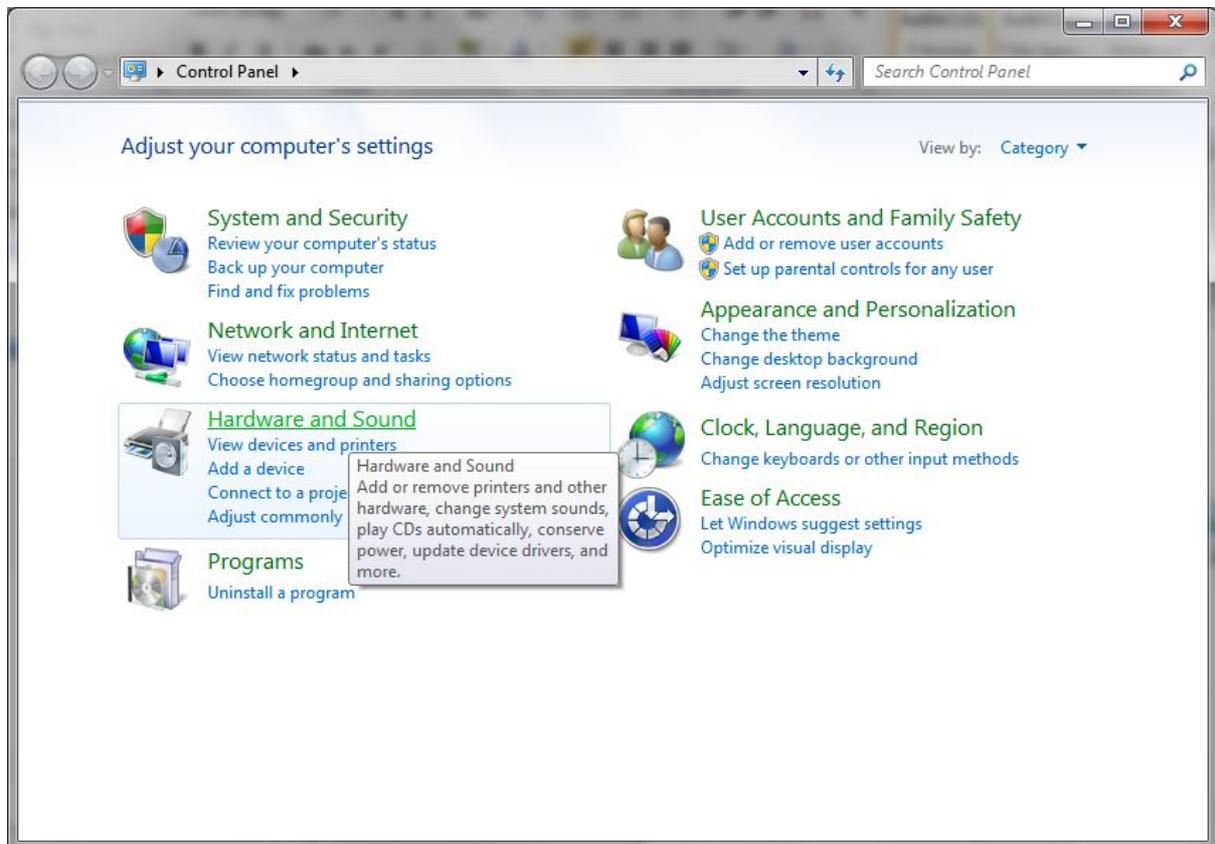
These instructions will get you up and running in about 30 seconds with Windows 7 or Vista.

The first time you install your USB2Serial board, it will flash the three LEDs in a hypnotic pattern, giving a clear indication of the happiness the board feels at being powered up and ready to do its thing.

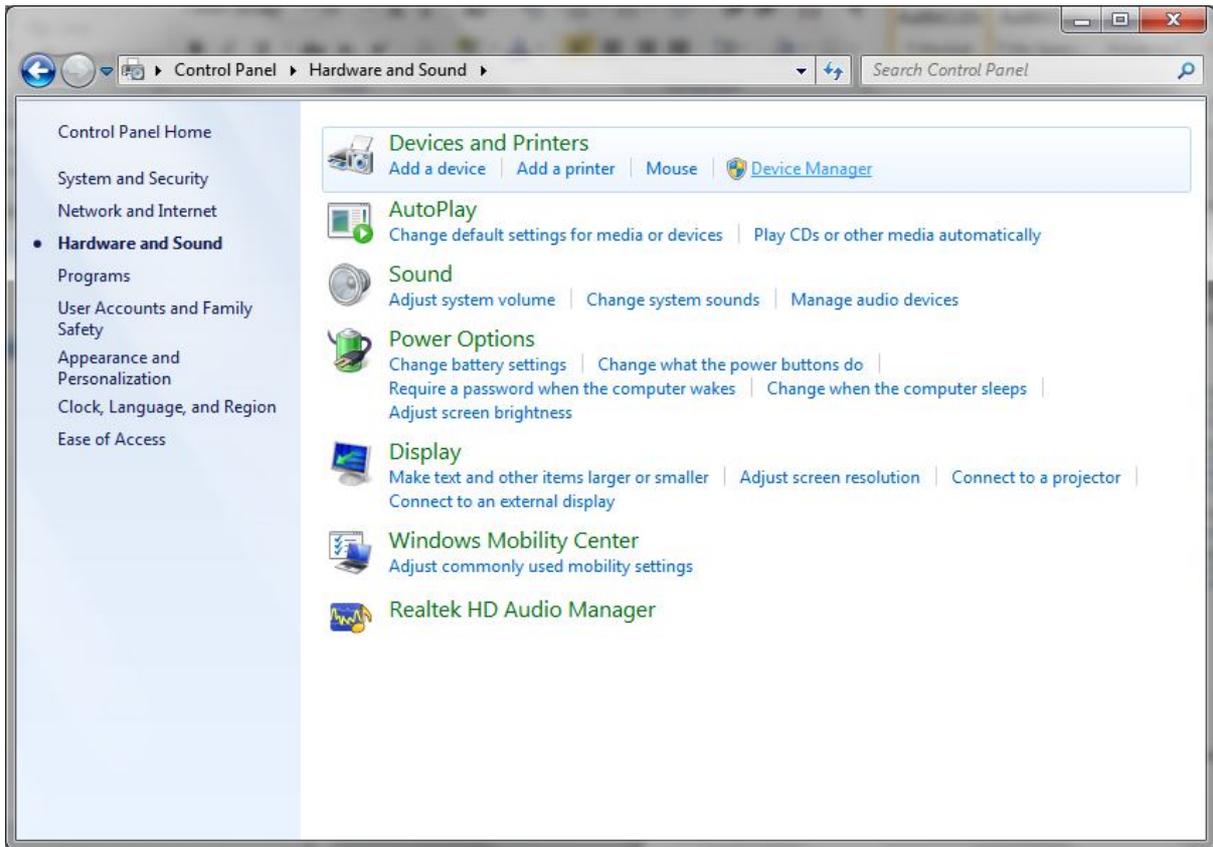
Windows will grind away for a while and either ask you for the installation files (in which case, you can simply point it at the USB2Serial.inf file that you can download from the Embedded Adventures product page) or decide it doesn't know what to do and present you with this:



No problem! Just go to control panel and click on **Hardware and Sound**:

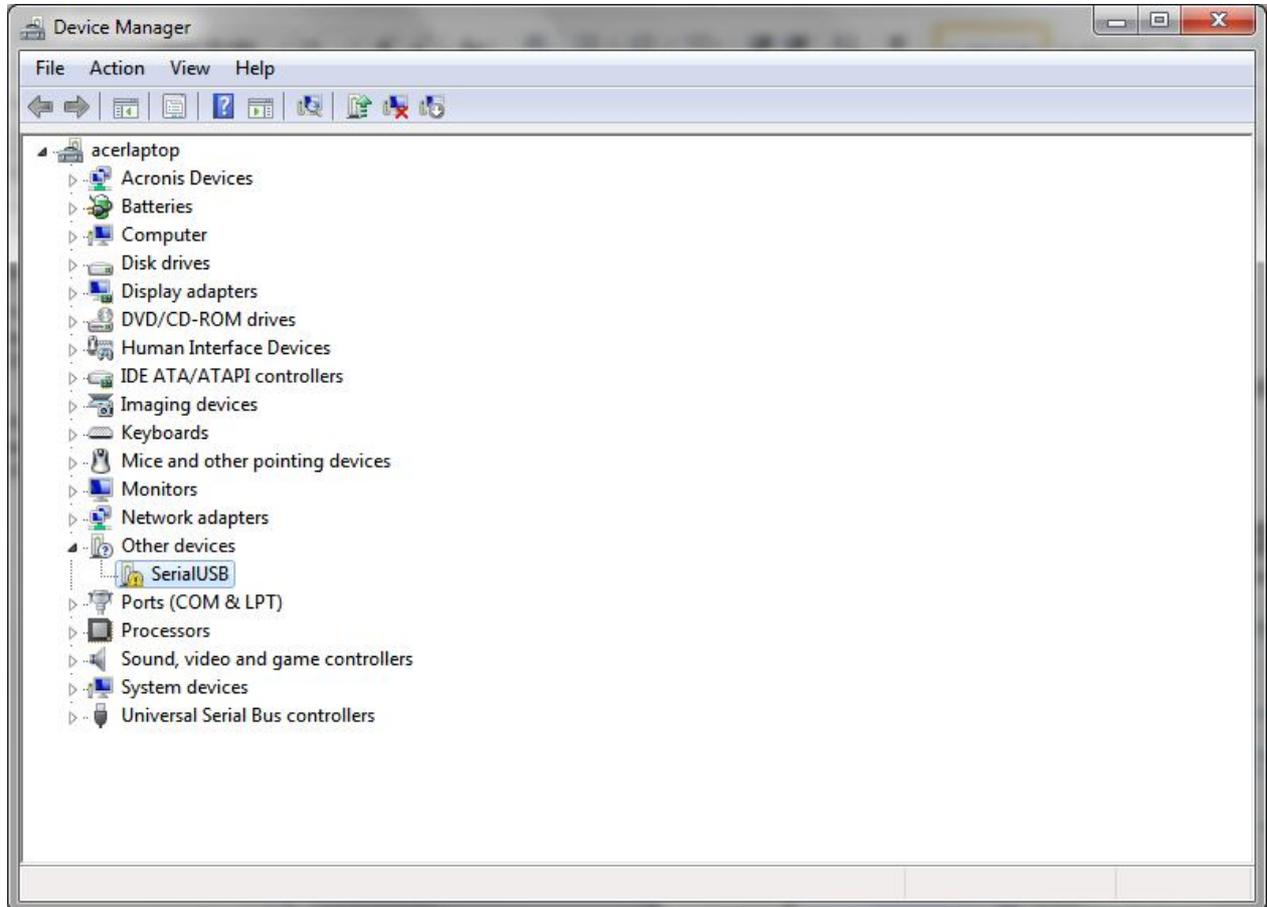


You'll now see the following screen. Click on **Device Manager**.

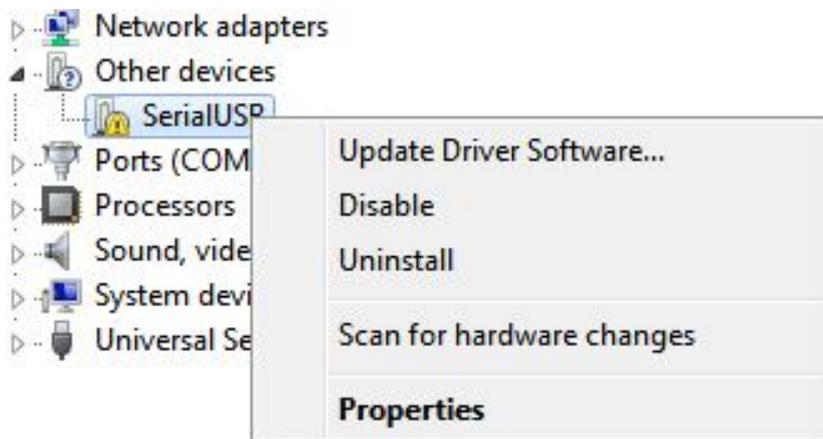


Windows will ask you if you want to allow this program to make changes to your computer. Say yes, and then you should see this list of devices.

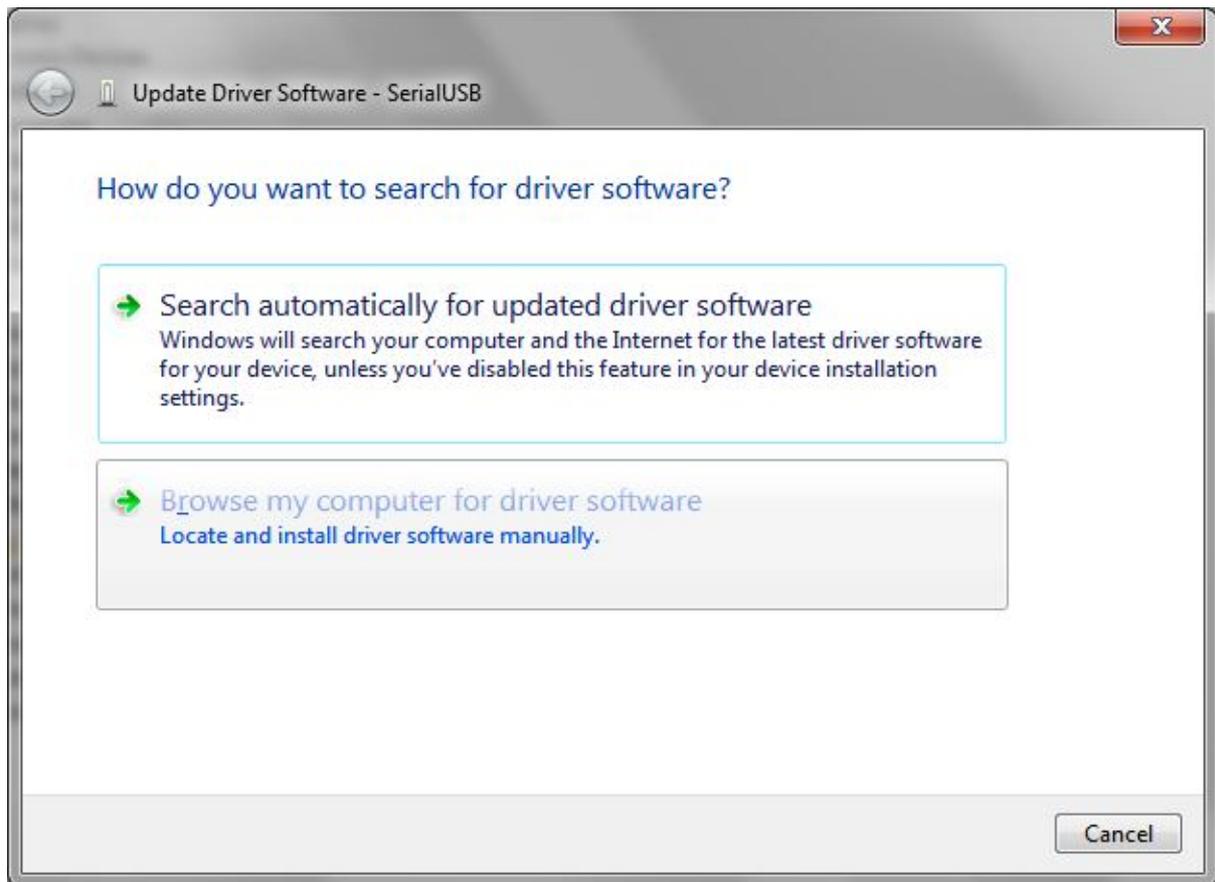
The "Serial2USB" device will be listed there under **Other Devices** for the moment, since windows doesn't know what it is yet. It might be called something similar, like "PLT-1003 USB2Serial module".



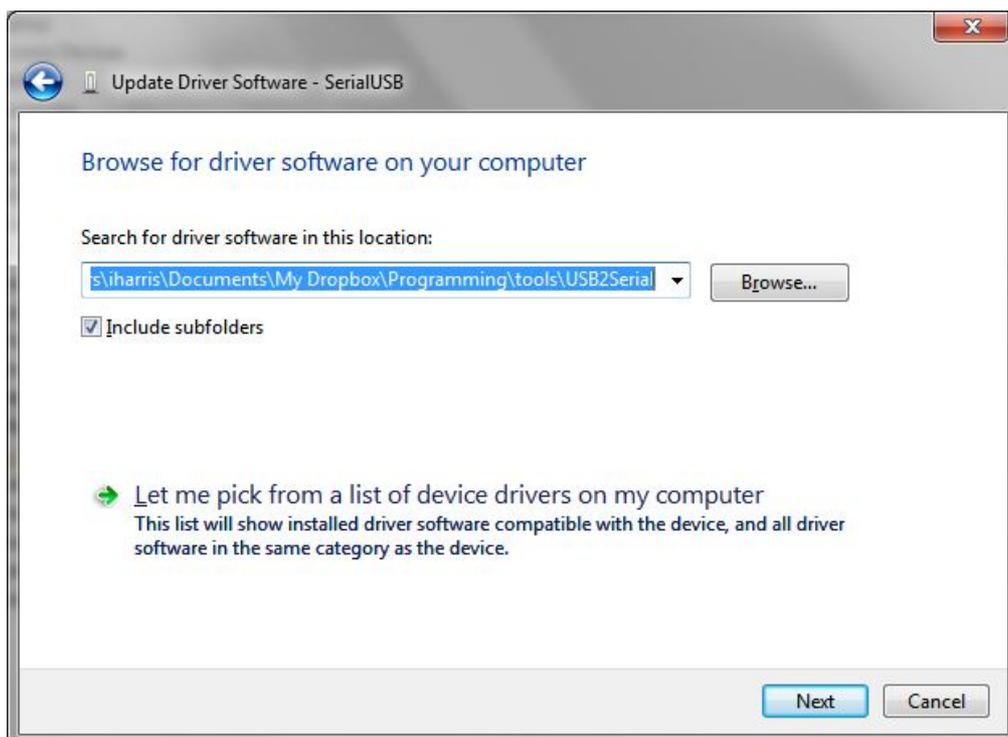
Just right click on Serial2USB and choose **Update Driver Software**



You want to **Browse my computer for driver software**



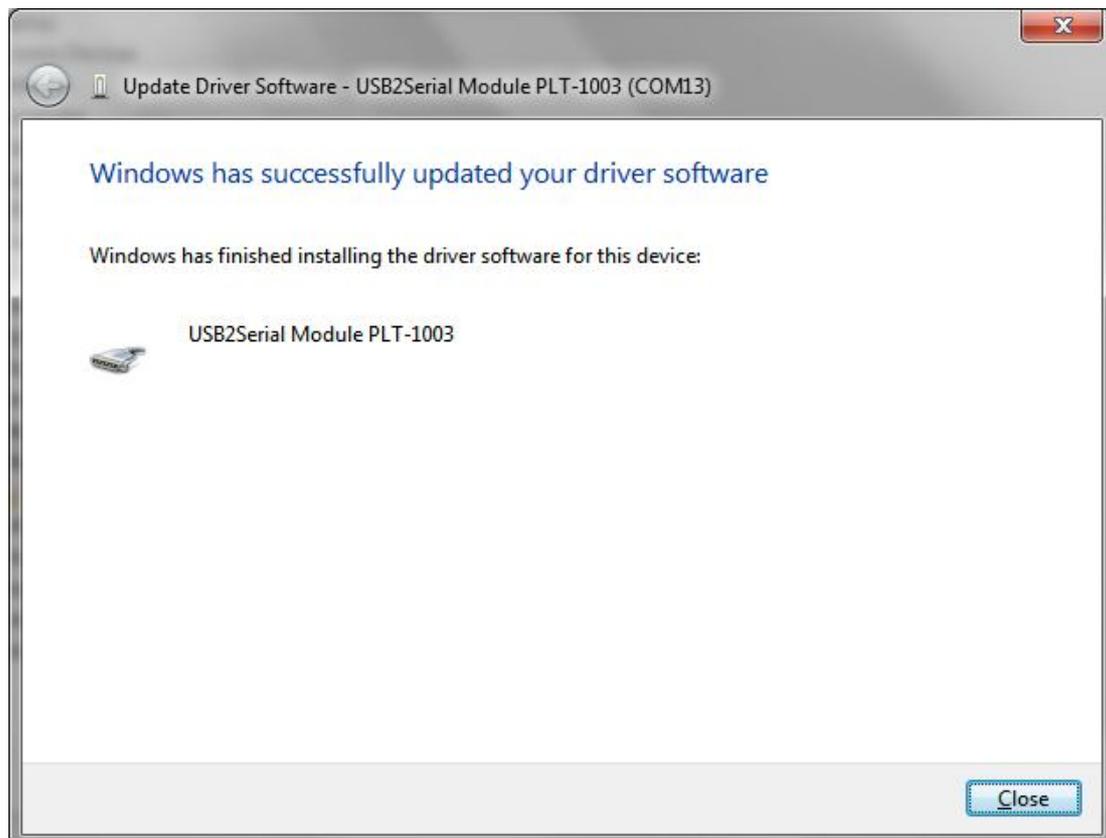
Now you can enter directory of wherever you've put the USB2Serial.inf file (or press browse to find it).



Windows will want to check if you really want to do this (the irony here is that we're using the Microsoft USB serial driver, so it really can verify the publisher of the driver software!) Click **Install this driver software anyway**:



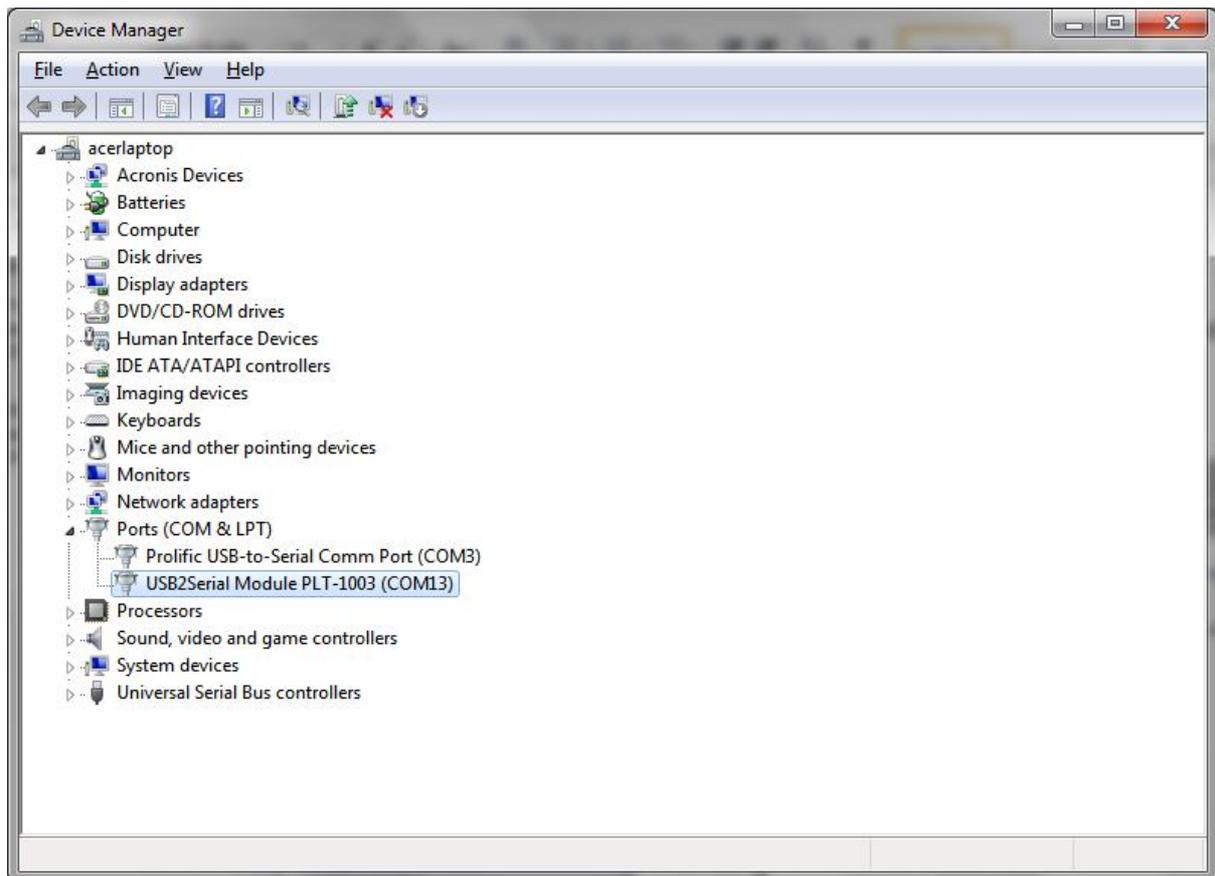
A little gurgling later, you'll come up with this:



Whoohoo! We're done. It wasn't that hard, was it. Please note that once you've installed it, the flashing will stop and unless you've connected it to something, the green "RX" LED will be continuously lit.

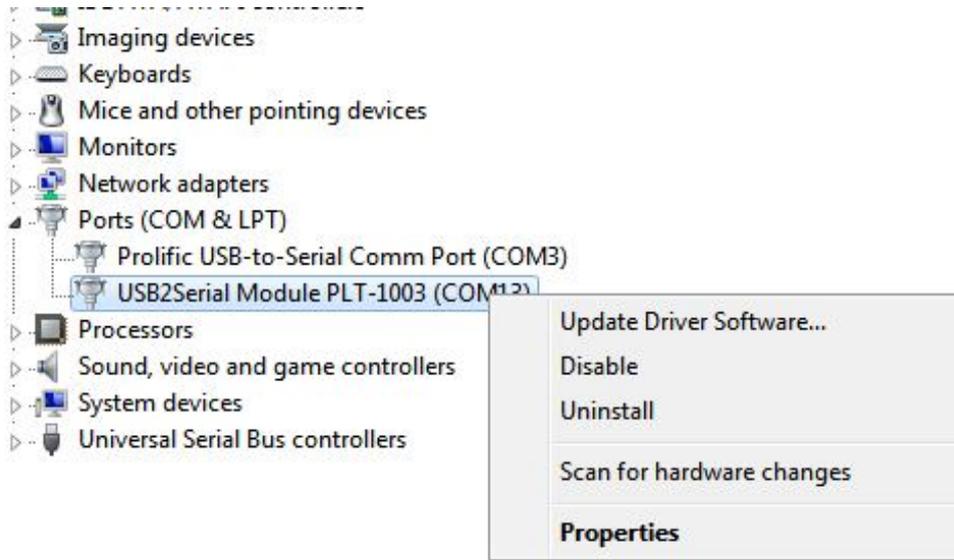
### Changing the COM port number

If you want to change the COM port number used by the USB2Serial board, simply go back into Device manager:

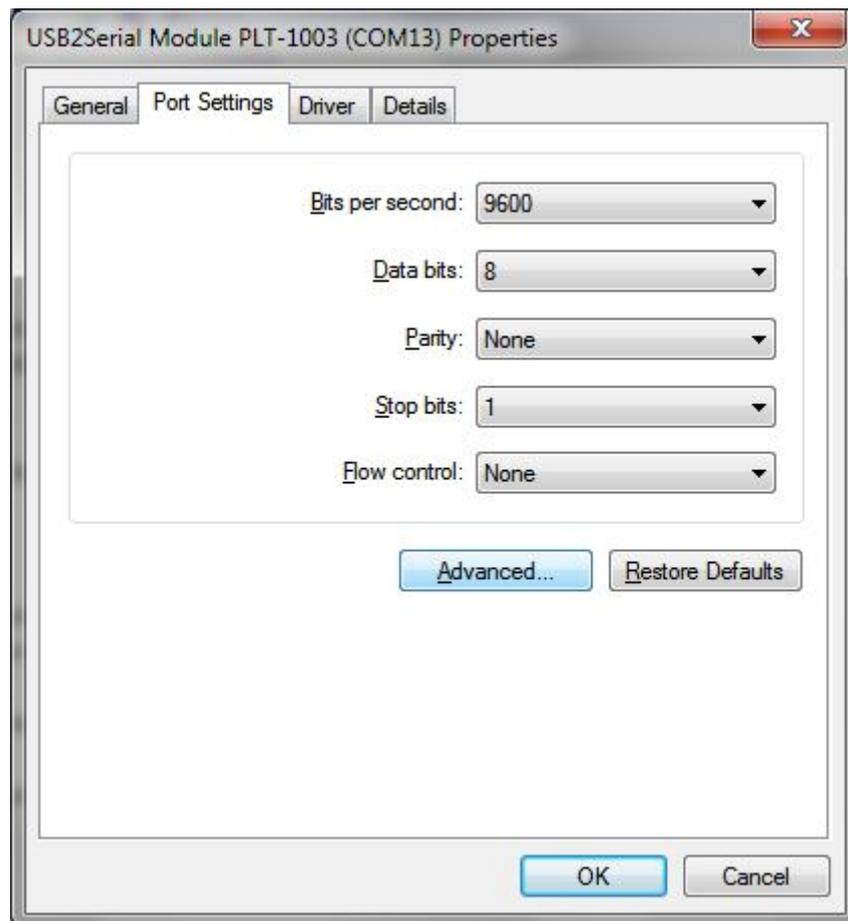


(You can see here that two COM ports are available, one is a boring Prolific USB 2 serial port and the other is the PLT-1003).

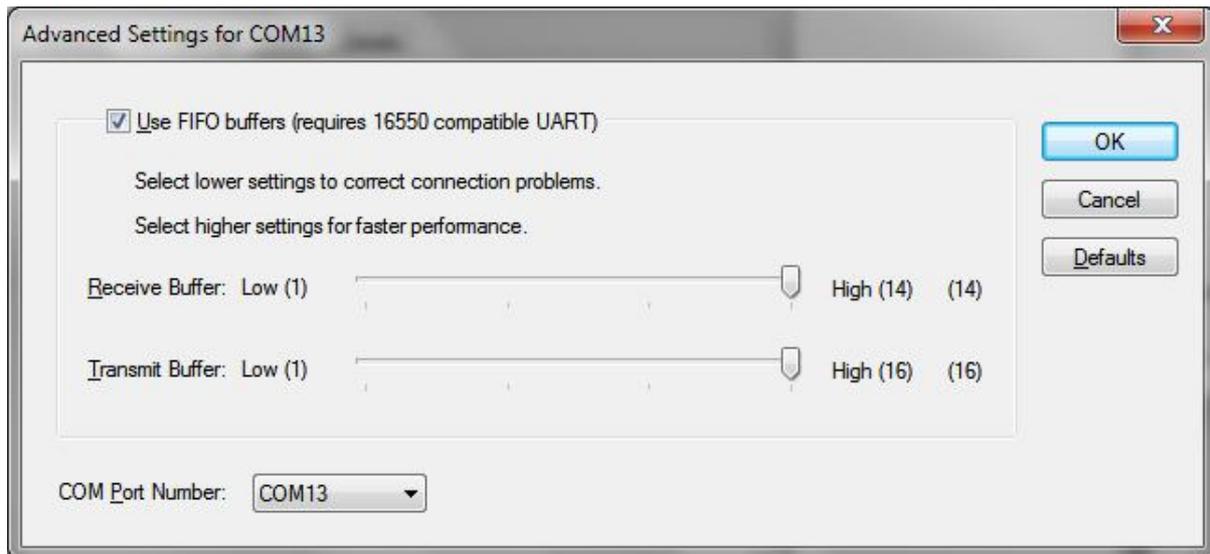
Both are under **Ports (COM & LPT)**. Right click on USB2Serial and choose **Properties**:



Click on the **Advanced...** button.



And now you can change the **COM Port Number** from the pull down list.



Don't worry if it says **(in use)** after the COM port number. That just means that some other device, at some time, has used that COM port number. You can duplicate them, quite happily. If you actually do have (for example) a COM 1 already on your system, then obviously you can't open both of them. Often Bluetooth drivers etc create bunches of COM ports.

## Versions

Document Version	Matches hardware	Date	Comments
Version 1.0	Version 1.1	10 Oct 2010	Initial Version for board v1.1AB
Version 2.0	Version 4 and 5	12 May 2012	Updated for new surface mount board. V5 board is identical except a silkscreen error has been corrected. V4 has this error corrected with an overlay.