Serial Data Logger HF-10

Andy Talbot G4JNT *12 October 2011*

Covers PIC Firmware ***SerialLogger*** *Version 1.0*

Overview

The HF-10 logger is designed as a voltage monitor with serial output for a typical connection to a PC running serial communication software. It will measure the voltage on each of four analogue channels and send the corresponding measurements to the host in real time. A precision 4.096 Volt reference and a 10 bit D/A converter ensures accurate measurements to a resolution of 4mV.

The module communicates with the host either through a standard COM port (RS232) interface, or via USB. For the latter, a virtual COM port is established with a USB-serial converter chip on the PCB. In each case the module derives its power from whichever interface it is configured for , so no external power supply is required.

The interval between measurements can be set by link selection to be in the range 80ms to 5 minutes. The data sent to the PC appears as ASCII text containing an interval time stamp in seconds followed by the voltage on each of the four channels in millivolts. The format has been optimised so that it can be directly read into an Excel spreadsheet for curve plotting and data analysis.

Another user selectable link provides the option for a shorter output data string containing just a single value from analogue channel 1

Circuit Details

The circuit diagram is shown in Figure 1 and consists of a PIC 16F819 which has an internal 4 channel 10 bit A/D converter with a MAX6004 4.096V precision voltage reference supplying the reference for the A/D conversion process. The PIC generates an interrupt at regular intervals, measures the voltage present on each channel, converts the 10 bit data to a BCD value representing 0 to 4092mV and sends this as an ASCII text sentence on the serial link at 9600 baud. A continuous count of seconds since the logger was started, or reset is kept. Each output sentence is preceded by the current seconds count as a six digit number. The 24 bit count will roll over after approximately 194 days of continuous monitoring.

For COM port operation, power is derived from the positive voltage present on the RTS and / or the DTR lines of the RS232 interface (Pins 7 and 4 of the standard 9 Way D-Type connector). A zener diode clamps the supply to the PIC to a maximum of 5.6V. The output uses 'pseudo RS232' levels of 0 and 5V rather than true +/- 12V. No COM port ever encountered has ever failed to make use of this.

For USB interfacing, an FT232RL chip is installed on the PCB to supply the necessary COM port. Unless one already exists on the host PC, a driver for this will have to be downloaded from the FTDI chip website [1]. However, as many manufacturers of PC add-on hardware use FTDI Chip products for their interfacing, a suitable driver may already be present Details of driver installation are given at the end.

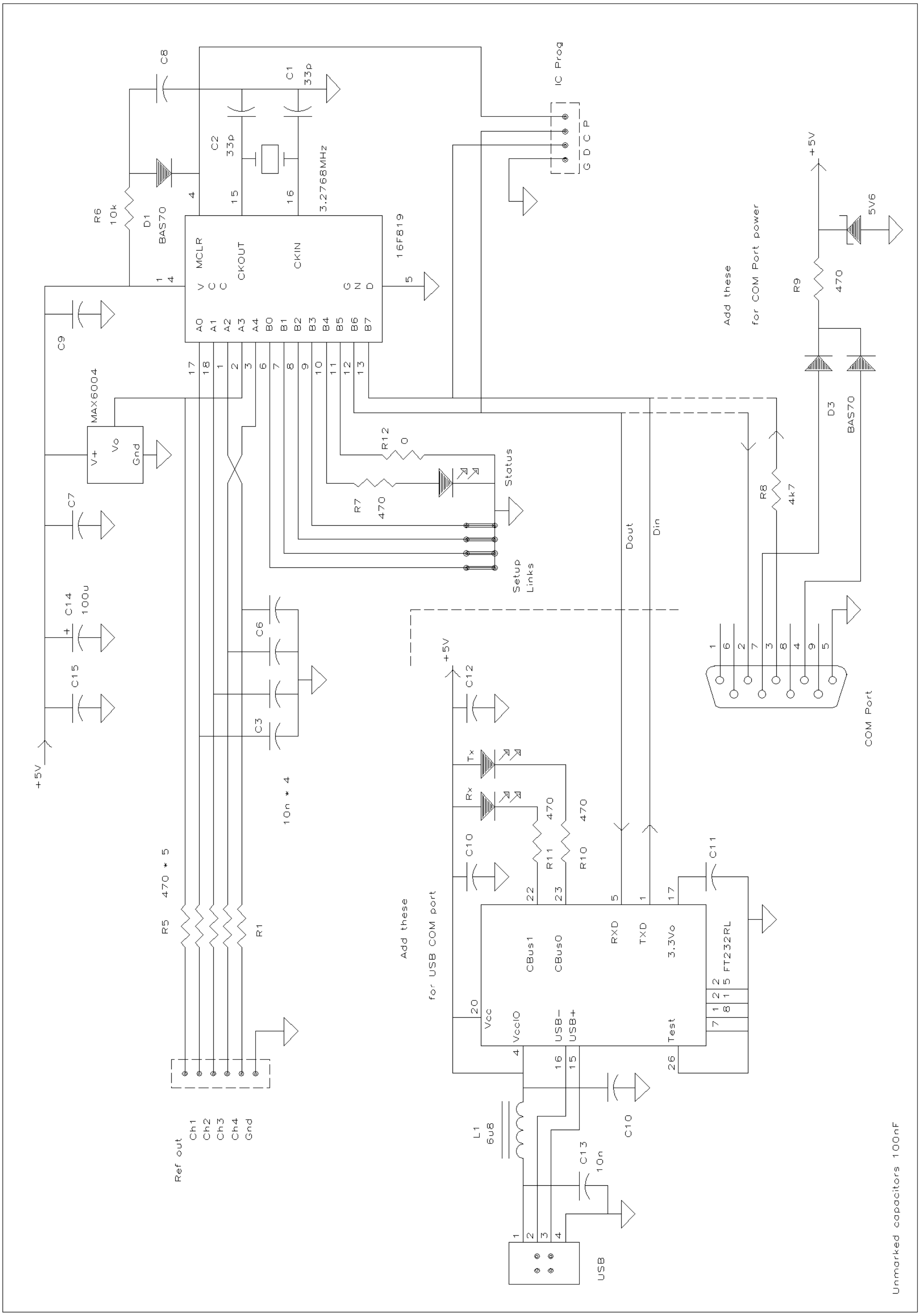


Figure 1 HF-10 Data Logger Circuit Diagram including both USB and COM port power / connections

As the serial data polarity is the opposite way round for direct RS232 connections and the FT232RL chip (and also likely for Bluetooth or fibre interface connections), it is not possible to use both RS232 or USB together on a single module. Different components are installed for each option, although it is permissible to install both sets of components together. In this event, dual mode operation is possible by adding or removing the single link, labelled R12 in the form of a zero-ohm resistor that defines the polarity in use.

The Hardware

The module is built onto a 40 x 60mm PCB and can have either a USB-B socket mounted on one end or a DB-9 female socket can be mounted directly on the edge of the board. Photo 1 shows a module configured for USB only, with the RS232 components not installed

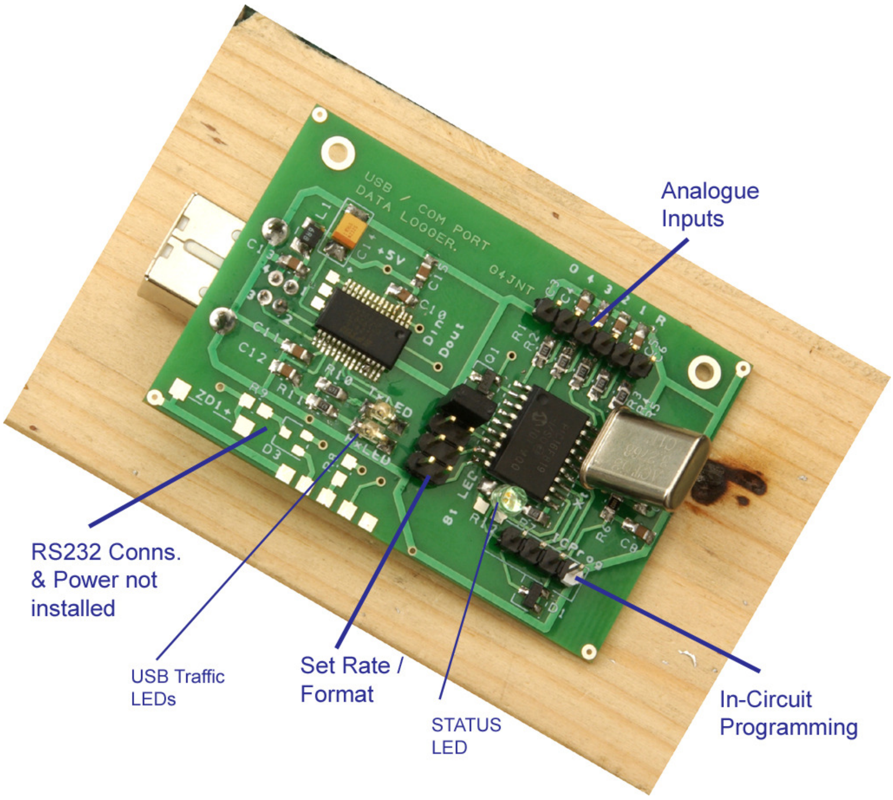


Photo 1

The four analogue channels '1' – '4' are brought out to a six-way header that also carries a ground connection 'G' and the reference voltage output 'R' The latter is provided to allow off-board analogue conditioning such as a DC offset to be incorporated. Series resistors are present on the board to provide some protection against overvoltages. The PIC contains clamp diodes to each supply rail and the current through these is specified as 20mA maximum. With the value of 470Ω shown, this means a maximum of around +15 / -10V before the specification is exceeded. Capacitors are included on the PCB for filtering, although the PCB pads for the resulting series-R / shunt-C allow users to instead install voltage dividers here by substituting the filter capacitors with shunt resistors, and changing the series value appropriately.

The four pin header is primarily for in-circuit programming of the PIC and in normal operation no connection need be made to this. It carries the reset line which can be used for resetting the seconds count at the start of a logging period. Future upgrades to the PIC firmware will be installed via this port.

The 2 x 4 way header takes the links that define timing interval and output format

The two LEDs around the FT232RL chip are not essential, they show RS232 traffic flowing to and from the chip. However, when initially setting up is helpful to be able to see when data is sent or received. Note that the LED labelled “Rx” actually shows data going OUT from the logger; the labelling is such that it refers to a COM port on a PC where Rx is data in, and Tx is Data out. The Rx LED should flash in synchronism with the Status LED

Link Settings

Four links select the measurement interval and sentence length as shown in the table below. ‘1’ means a link is installed, ‘0’ means there is no link in that position. In the photograph, Link 1 is shown installed

Table 2 Link Settings

Links 1,2,3	**Sampling interval**
000	80ms
100	1s
010	5s
110	10s
001	20s
101	1 minute
110	2 minutes
111	5 minutes

Link 4		**Output Data Format**
0 (no link)	4 Channel	000045 * 1234 4093 0000 3400
1 (link fitted)	1 channel	000074 * 4092

In Use.

Before using the USB interface, ensure the correct drivers from FTDI chip are installed on your Host / PC and you have identified the correct COM port the module appears at. See Annex 1 *for more details.*

Set up suitable software on your host machine to read the incoming serial data. ***Hyperterminal*** that comes (or at least used to) with Windows is a good one, although a web search for “serial interface software” will throw up a lot of alternatives for download. Select one that suits your own preferences.

The serial output format from the HF-10 is 9600 baud, 2 stop bits, No parity, No hardware handshaking which is often abbreviated to “9600-N-8-2” The logger sends a [carriage return] [line-feed] pair so you may also need to set up the serial interface software to not generate extra line-feeds (usually to be found in the CR-LF translation settings)

Install the links for your required sampling interval and connect to the interface. The logger will start running immediately, and the introductory message shown in the first line of Table 1 should appear. The Status LED will flash each time a measurement is made and a new line of data will appear. When the 80ms sampling rate is selected, the seconds count on the left hand side will show 12 or 13 each of the same values as there are 12.5 measurements made every second. If the link settings are changed while the logger is running, the new rate will not take effect until after the next reading. So beware of doing this when on the one or five minute settings. It may be easier to reset and start from zero. The logger can be reset by either disconnecting and reconnecting the interface, or by shorting the two end pins at each end of the four-pin In-Circuit programming interface.

And that's it ! The value appearing in each column to the right of the asterisk is the voltage in millivolts on the associated analogue input channel to the PIC. The 10-bit value is in increments of 4mV and ranges from 0 to 4.092 Volts corresponding to a digital value of 0 to 1023 with a 4.096V reference for the A/D converter.

Table 1 – Typical ASCII Data Output format

```
Serial Port Data Logger   G4JNT  Ver 1.0 Sept 2011
000000      *     0000  0000  0000  0000
000020      *     1848  0384  0024  000
000040      *     2580  1000  0152  0004
000060      *     2928  1500  0348  0024
000080      *     3136  1868  0576  0060
000100      *     3276  2144  0808  0112
000120      *     3380  2360  1028  0176
000140      *     3456  2532  1232  0252
```

The version 1.0 firmware does not make use of any serial input commands (typed in from a keyboard for example) although the hardware connections are in place to allow this. Future versions or upgrades may include the ability to alter logging parameters without changing the link settings.

Completely different software can also be installed.

Reading the data Into a Spreadsheet such as Excel.

First set up your serial interface software to save received data to a text file. In Hypertrm this is done by via the menu items "Transfer" then "Capture Text". Each line of data is then saved in the named file in exactly the format it appears on screen . A TAB character (ASCII 9) separates each column of data, including the asterisk.

To read into Excel, open the .TXT file with Excel ("open with" in Windows, or File-Open, type, .TXT from Within Excel) Then choose the 'character delimited' option ('constant width' will also work, but needs some fiddling with to get a decent imported result). Select the default TAB as the delimiter. Each column should then appear as a separate column of consecutive cells. Globally delete the whole column of asterisks to make a cleaner spreadsheet ready for analysis / plotting.

Kit / Hardware Construction

Purchasers of the part-kit should receive the following components:

Printed Circuit Board
Programmed 16F819 PIC
3.2768MHz crystal
FT232RL (USB version only)
USB-B Socket (USB version only)

You will need to provide capacitors - several of 100nF and two around 33 – 47pF are needed for the crystal oscillator. Resistors include several around 470Ω and one in the 10kΩ - 33kΩ region. None of the values are critical and you may have your own analogue input requirements anyway. Also you will need a BAS16 or similar SMT diode, MAX6004 voltage reference and header pins etc. COM port power will require a 5.6v Zener diode, a BAS70 or similar dual diode (the same type can be used in the D1 position). A 6.8uH choke is shown on the circuit for USB power line decoupling. This is not essential and may be replaced with a short circuit if conducted HF noise is not an issue The two USB traffic LEDs are not essential, although the module's functionality without them has not been tested.

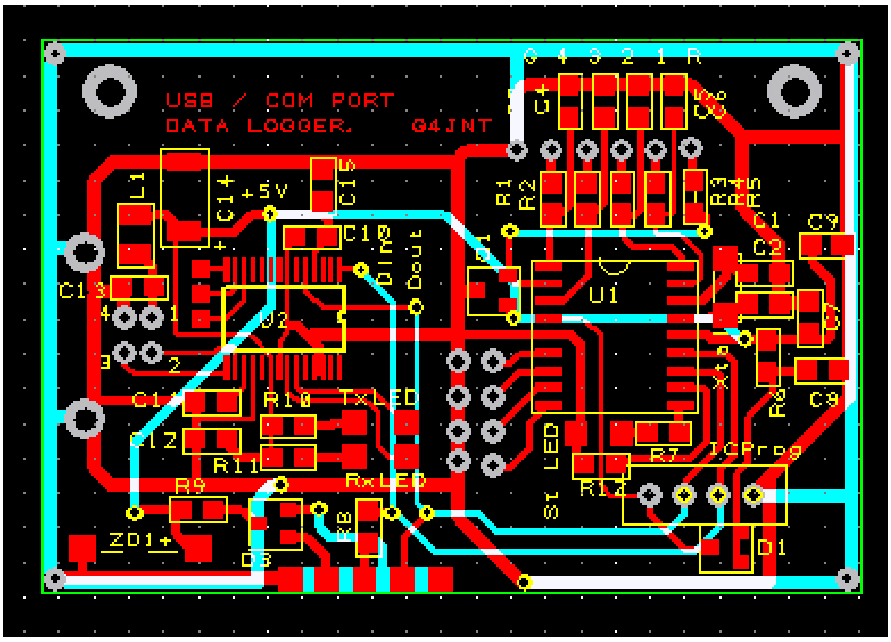


Figure 2 PCB Overview

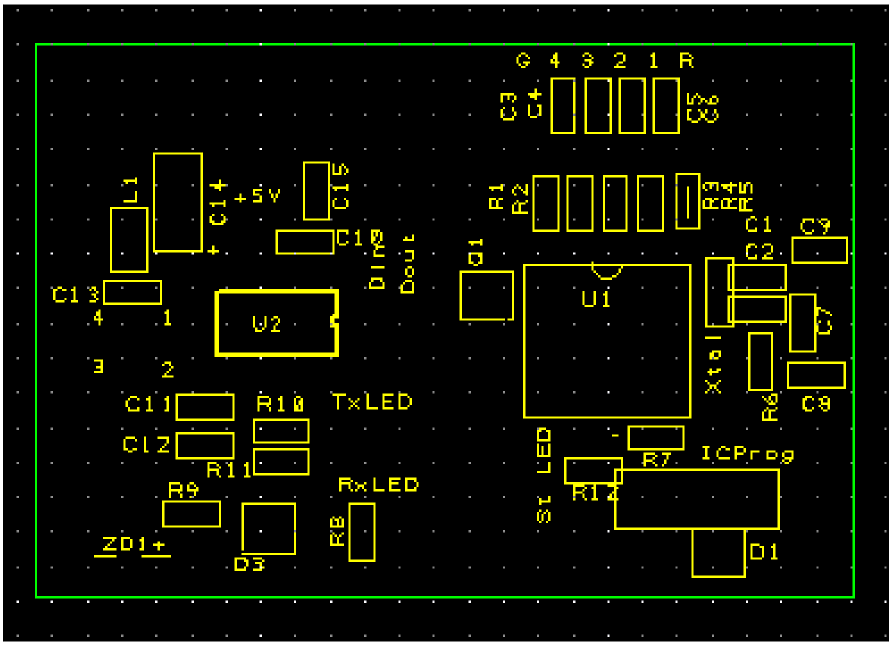


Figure 3 PCB Silk Screen

Install all components in their correct places on the PCB, referring to Figures 1 - 3 as needed. For the USB version, or any other application that needs 5V / logic level data, the zero-ohm link R12 is not installed. For RS232 connection direct to a PC, R12 or a wire link in its place is needed to instruct the PIC firmware to swap the data polarity.

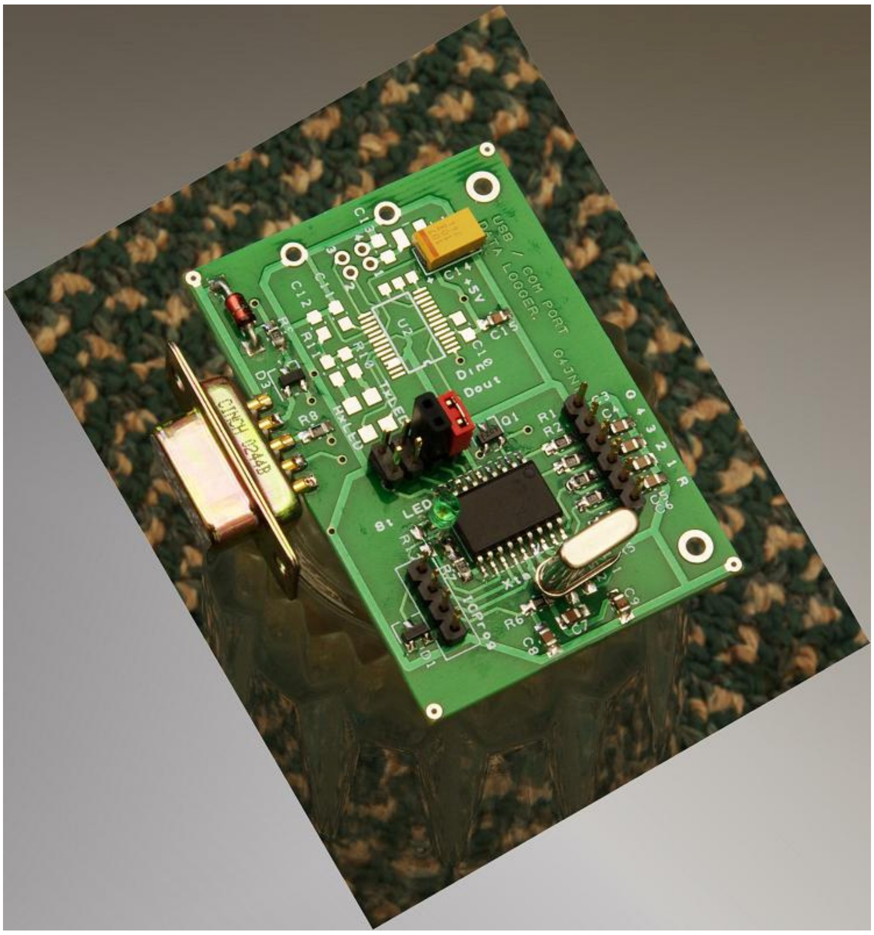
The close spaced FT232RL chip is best soldered using the flux-and-bulk-solder-wipe technique. Apply separate flux to the PCB and pins, carefully align the IC with the pads ensuring correct chip orientation, then gently wipe the soldering iron over all the pins / pads not bothering if adjacent pins are initially shorted together. Use a well-fluxed piece of braid or SolderWick to remove all excess solder, followed by a careful examination though a magnifier to double and triple-check there are no shorts and that all pads are connected.

Although the IC has 28 pins, only about half of these are actually needed in this application. The overall spacing for the FT232RL device pads is a little short. While quite adequate, it is best to align the chip longitudinally based on the middle pins on each side, and let those at each end overlap outside the pads at each end. The error is only about 0.1mm, but can be seen under decent magnification.

One pad for the 2x4 way header for the link selection is misaligned. Cut its associated pin short and solder to the surface of the pad.

Everything else should be straightforward enough using traditional soldering techniques with a small soldering iron bit.

COM Port Users



The Pads on the side of the board are designed to connect directly to a 9 Way D-Type female socket. It may be a bit tight, especially if a second-hand connector has been used with solder on the tags.

In your Serial Communications software, ensure RTS or DTR (and preferably both) are set. DTR usually is automatically, RTS sometimes wobbles with traffic flow.

Users of cheap (like several Ebay specials) RS232/USB interfaces may find RTS and / or DTR are not even implemented, or are not stiff enough, or of low voltage. In this case, you will have to use your own source of power for the 10mA or so needed.

Other Interfaces

Bluetooth Modules such as those available from Sparkfun http://www.sparkfun.com/products/582 or a custom fibre-optic or wireless interface can be used instead. The basic PIC logger circuitry without the FT232 or RS32 components can be built, and the serial data in/out connections connected to the respective external interface hardware. Ensure this is configured for 9600 baud. You will have to provide the PIC circuitry with a suitable 5V (nominal) supply which may shared with the one needed for the interface.

Annex 1 FTDI Driver Installation

Visit the FTDChip website http://www.ftdichip.com/Drivers/VCP.htm for their Virtual COM Port driver downloads. Take a look at the Installation Guides page and the Documents section of the site for selected operating systems.

Download and install the drivers applicable to your operating system, following those instructions.

Please note that drivers are backwards compatible with earlier devices. The FT232RL chip used here is one of these earlier devices so if you already have an old FTDI driver installed, no upgrade should be necessary.

Be aware, you may already have suitable drivers installed if other external hardware has been used in the past. Many manufacturers use FTDI products for interfacing, and it is not unknown for the drivers to be automatically and transparently installed as part of their installation process. If upgrading the driver, the old one has to be de-installed / removed first. The FTDI documentation includes instructions for doing this.

(Users of the SDR-IQ SDR Receiver from RF-Space will have the drivers already installed, as will owners of the Prologix IEEE488 Interface)

When the installation is complete, plug in the Logger and a "New hardware Found" message should appear. If it can't appear to find the driver, steer it to the location for your new installed driver . This will be saved for next time and subsequent connections will be automatic.

It may, or may not, come up with a message indicating which COM port has been allocated. Each new FTDI interface added generates a new COM port, and these can get to quite high values. Use Windows device manager, system, hardware, ports menu to identify the Com port if not indicated. There is also scope in the 'Advanced Settings' to reallocate the COM port number to one that was perhaps allocated previously and is not now in use – the old one will have to be deleted to make the new number available. Make a note of the COM Port it allocates. This will be remembered by the computer operating system..

If using HyperTerminal, or any other serial communications package that allows multiple individual profiles to be stored, save the one set up for this COM port and settings interface, under some meaningful name such as "*HF-10 Logger.ht*"