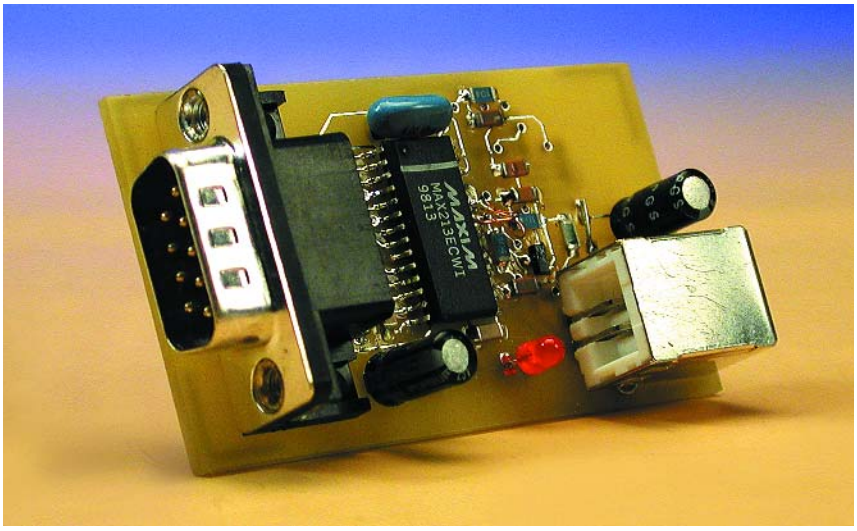
USB-RS232 Interface

A compact solution for missing ports

Design by L. Lemmens

Thanks to a special integrated circuit from FTDI Chip, computer peripherals with an RS232 interface are easily connected to a USB port. This simple solution is ideal if a peripheral does not have a USB port, your notebook PC has no free RS232 port available, or none at all!



After a slow and faltering start, the USB port has become commonplace on PCs, to the extent that the latest GHz machines have just one RS232 port left, or none at all. The compact USB-RS232 interface described in this article allows your good old RS232 peripherals (printer, programmer system, etc.) to be hooked up to a USB port. The free driver programs for Win98/ME/2000/XP, Linux and Apple Macintosh make the interface virtually transparent, enabling the USB port to behave like a regular COM interface. The driver and the conversion chip from Glasgow-based FTDI Chip allow a full serial data link to be set up on a 9-way RS232 connector, including all handshaking signals. The UK representative for FTDI Chip is Alpha Micro Components Ltd. in Basingstoke (www.alphamicro.net). International distributors for FTDI Chip products may be found on the 'Sales Network' page which may be accessed via www.ftdichip.com. FTDI Chip has representatives in most parts of the world and hosts a very useful website.

Function and architecture

Although it is not necessary to know all the ins and outs of the converter chip if you just want to use the circuit we're about to describe, it is still useful to have an idea of what is going on inside the black box.

The simplified block diagram of the FT232AM is shown in **Figure 1** and the pin assignment, in **Figure 2**. Like its family member the FT245AM, the FTDI Chip FT232AM is essentially a serial USB FIFO (first in, first out register), which is controlled by the computer by way of a virtual COM port. The difference between these two components is that the FT232AM contains a UART which in turn comprises an RS232 interface (using TTL levels). The FT245AM, on the other hand, features a 8-bit interface with handshake lines that allow direct access to the on-chip FIFO. This makes the '245 IC particularly interesting if an existing microcontroller system is to be upgraded with a serial interface. Note, however, that you will need to provide the necessary software 'glue' yourself!

At the side of the USB, the two chips are identical and not surprisingly they contain the same drivers in the relevant section. At the input side, an USB Transceiver forms the link with the USB cable and its two signal wires called D+ and D– for the USB 1.1 Full-Speed mode. The 3.3-V reference voltage needed for the USB Transceiver is supplied by an internal low-drop voltage regulator whose output is available at pin 6 of the chip. This is done not only to enable the reference voltage to be applied to an external decoupling capacitor but also to allow the USB Full-Speed mode to be defined. With reference to the circuit diagram shown in **Figure 3**, that is achieved using resistor R6 which pulls the D+

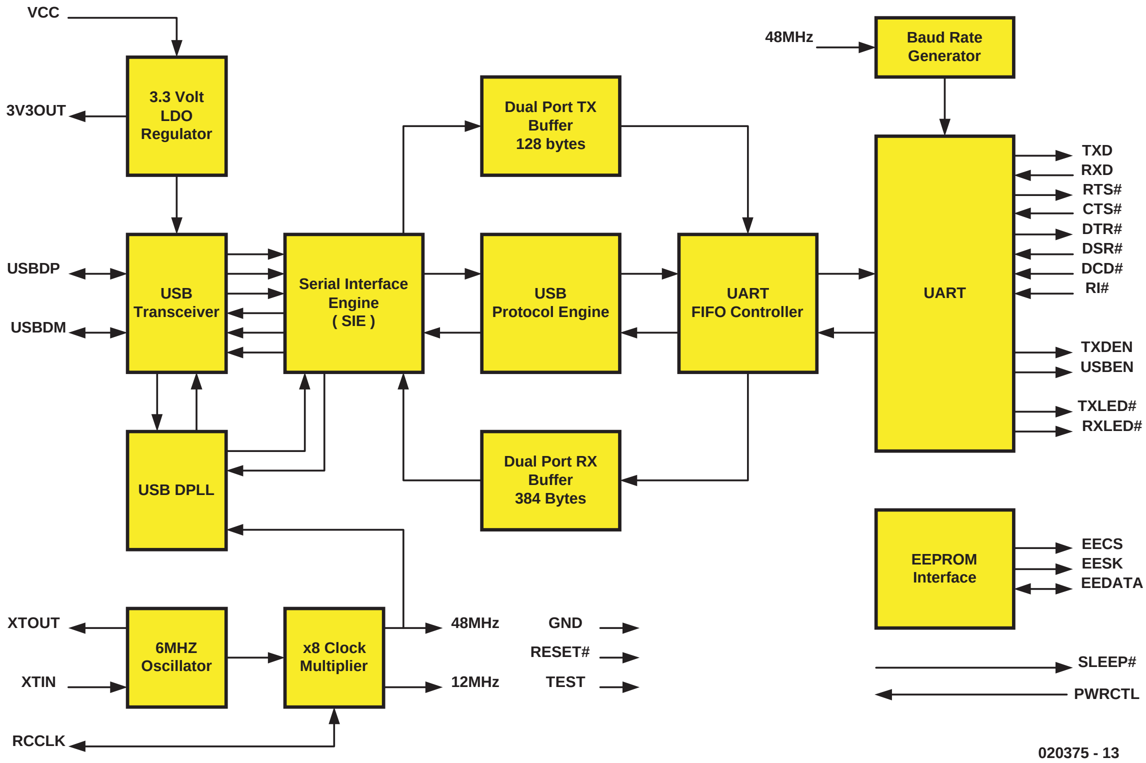


Figure 1. Simplified block diagram of the USB/RS232 converter type FT232AM (courtesy FTDI Chip).

line to +3.3 V. This level causes the USB Host (i.e., the USB controller in the PC) to recognise our interface as a Full-Speed device and arrange for the appropriate addressing. In the case of a Low-Speed device, the D– line is held at +3.3 V with the aid of a resistor. Behind the USB Transceiver we find a functional block identified as 'Serial Interface Engine' which handles the parallel-to-serial and serial-to-parallel conversion of USB data. Next, the 'USB Control Engine' processes the USB control information and looks after the communication with the USB Host Controller (in accordance with the USB Low Level protocol), as well as the commands that define the UART's functional parameters.

Buffers for 'receive' and 'transmit' ('Dual-Port TX Buffer' with 128 bytes capacity and a 384-byte 'Dual-Port RX Buffer') arrange for the exchange of data in both directions (between Serial Interface Engine and the UART registers). The block identified as 'UART FIFO Controller' is responsible for the exchange process between the two buffers and the transmit and receive registers of the UART.

Functionally, the 'UART' proper is not unlike the one found in the PC. Its task is to supply all relevant signals to the RS232 interface and in addition, for RS422 and RS485.

The 'Baud Rate Generator' allows the serial data speed to be set between 300 bits/s and 2 Mbits/s (actually, up to 920 kbits/s for RS232 and 2 Mbits/s for RS422/485).

The 'EEPROM Interface' on board the FT232AM chip is intended for the connection of an external 93C46 EEPROM chip. Although the FT232AM will work happily without the addition of a non-volatile data memory, the interface will then 'report' as a standard serial device. The small EEPROM allows specific data like manufacturer and product identifier

FTDI Chip contact details

Future Technology Devices International Ltd.,
St. George's Studios, 93-97 St George's Road, Glasgow G3 6JA, United Kingdom.
Tel.: (+44) (0)141 353 2565, Fax: (+44) (0)141 353 2656.
URL: www.ftdichip.com

UK distributor:
Alpha Micro Components Ltd, Springfield House, Cranes Road,
Sherbourne St. John, Basingstoke, Hants RG24 9LJ, United Kingdom.
Tel : (+44) (0)1256 851 770, Fax: (+44) (0)1256 851 771.
URL: www.alphamicro.net

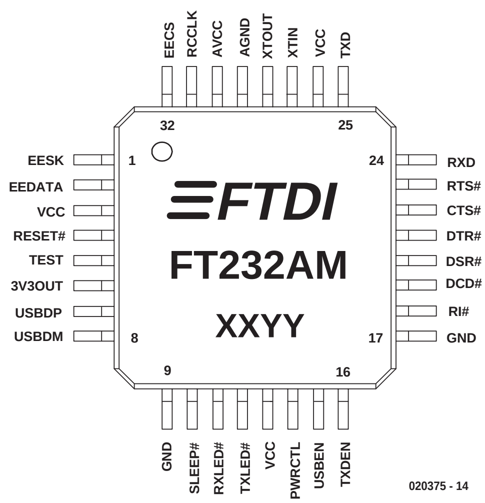


Figure 2. Pin connections of the FT232AM in QFP case (7 x 7 mm) (courtesy FTDI Chip).

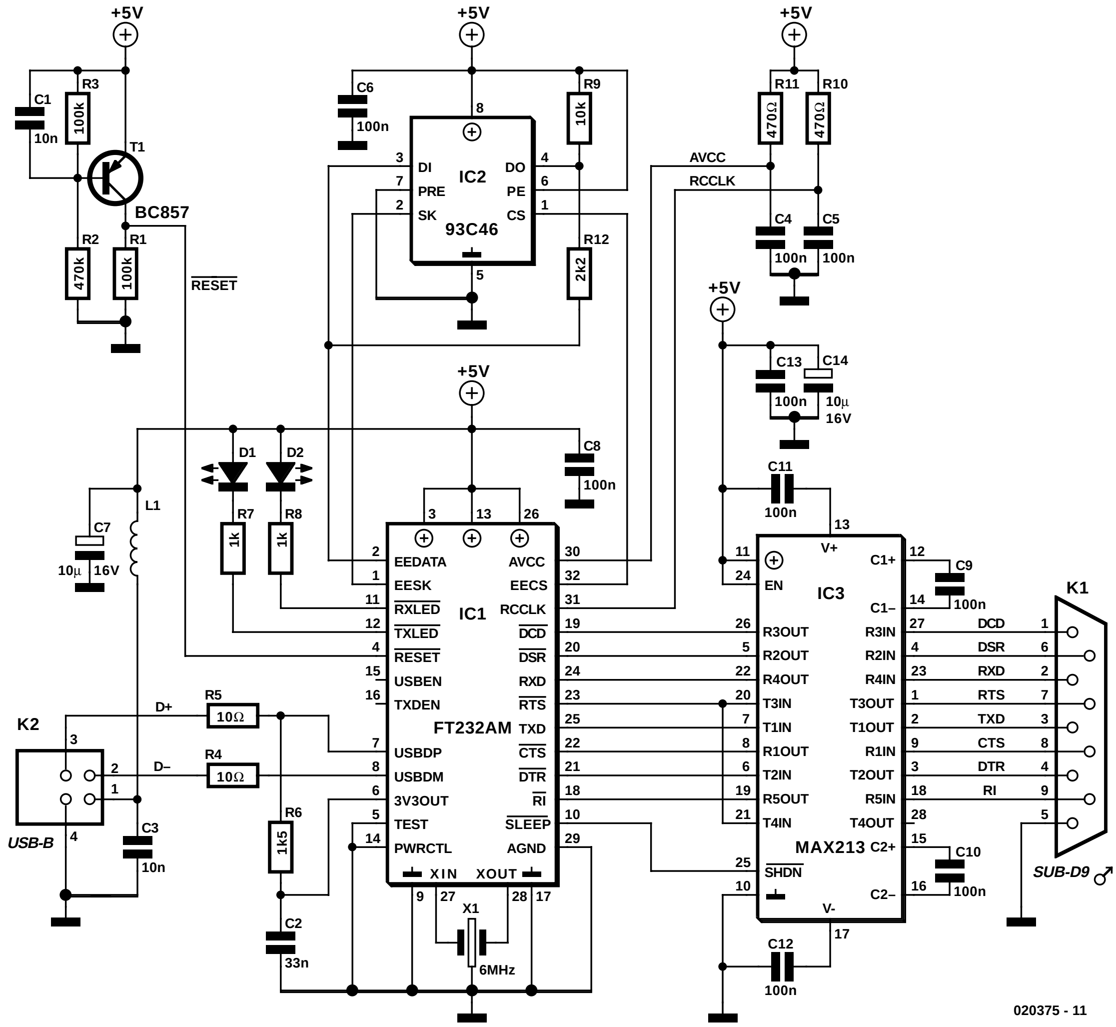


Figure 3. Circuit diagram of the USB/RS232 Interface.

codes (VID, vendor ID; PID, product ID), serial numbers, etc., to be permanently stored and made known to the operating system. Note that the EEPROM is obligatory when several USB/RS232 converters using FTDI chips are connected to a PC. This is because the drivers will only install virtual COM ports for converters with unique serial numbers. Without the serial number (i.e., without the EEPROM) only one virtual COM port can be installed.

Circuit diagram

The circuit diagram shown in Figure 3 looks rather uncomplicated. In the top left-hand corner we find the power-up reset circuit comprising T1 and the customary R-C network. Right beside it is the (optional) EEPROM. In the next lower 'row' we find, from the left to the right, the USB-B connection, the FT232AM, a MAX213 and finally the RS232 header.

The circuit receives its +5 V supply voltage from the PC via pin 1 of the USB connector (K2). A sufficient amount of supply noise decoupling is afforded by a small choke (L1) and capacitor (C7). In addition all integrated circuits have their own supply decoupling.

As already mentioned, resistor R6 pulls the USB D+ line to +3.3 V, in order to tell the USB host that the interface is a 'Full-Speed' device. The same resistor also triggers the recognition of an USB device when the interface cable is plugged into the USB port on the PC or on a hub.

Very conveniently, the FT232AM features two LED driver outputs that allow active data transmission (D1) and reception (D2) to be visualised.

Although the two R-C networks R11-C4 and R10-C5 are identical in value, their functions are quite different. The combination R10-C5 at the RCCLK pin is a timing network to ensure clock stability when the FT232AM wakes up from Sleep mode while booting. The other R-C combination, R11-C4, only decouples the voltage at the AVCC (analogue supply) pin which powers the internal 8x clock multiplier.

The MAX213 and its external charge pump capacitors only serve to convert the 5-V signals at the RS232 side of the FT232AM into true RS232 signals (i.e., having a polarized swing). Normally, that would mean approximately ±12 V, but in practice only ±8 V is achieved, with a maximum of up to ±10 V.

Circuit board

Even if the circuit diagram is uncluttered and fairly straightforward, that does not necessarily apply to the printed circuit board (**Figure 4**). The main reason for the discrepancy is found in the use of SMD parts. Also, the PCB is double-sided and

COMPONENTS LIST

All resistors and capacitors SMD shape 1206

Resistors:
R1,R3,R10 = 100kΩ
R2 = 470kΩ
R4,R5 = 10Ω
R6 = 1kΩ5
R7,R8 = 1kΩ
R9 = 10kΩ
R11 = 470Ω
R12 = 2kΩ2

Capacitors:
C1,C3 = 10nF
C2 = 33nF
C4,C5,C6,C8-C13 = 100nF
C7,C14 = 10µF 16V radial

Inductors:
L1 = BLM31A601S (Murata) (e.g., Farnell # 581-094)

Semiconductors:
D1,D2 = LED, 3mm dia.
T1 = BC857
IC1 = FT232AM or FT232BM (FTDI Chip order code FT8U232AM)
IC2 = 93C46 (optional)
IC3 = MAX213ECWI

Miscellaneous:
K1 = 9-way sub-D plug (male), PCB mount, angled pins
K2 = USB connector, type B, PCB mount
X1 = 6MHz ceramic resonator

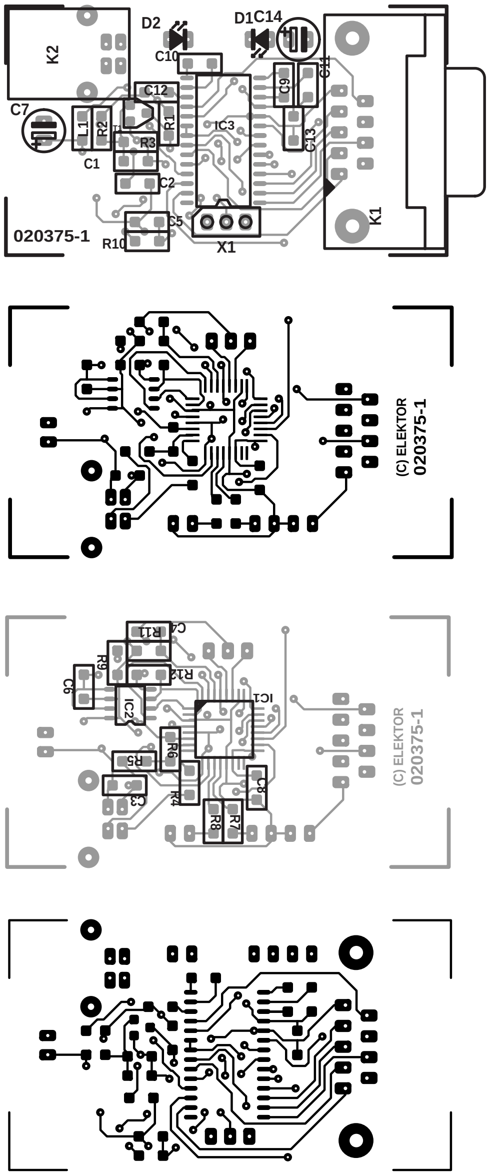


Figure 4. Copper track layout and component overlay of the PCB designed for the interface.

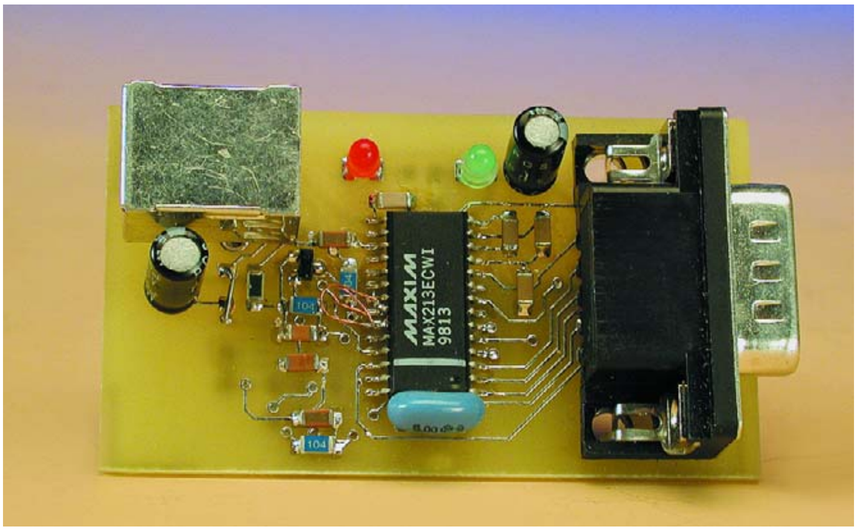


Figure 5. Our finished and fully working prototype board seen from above…

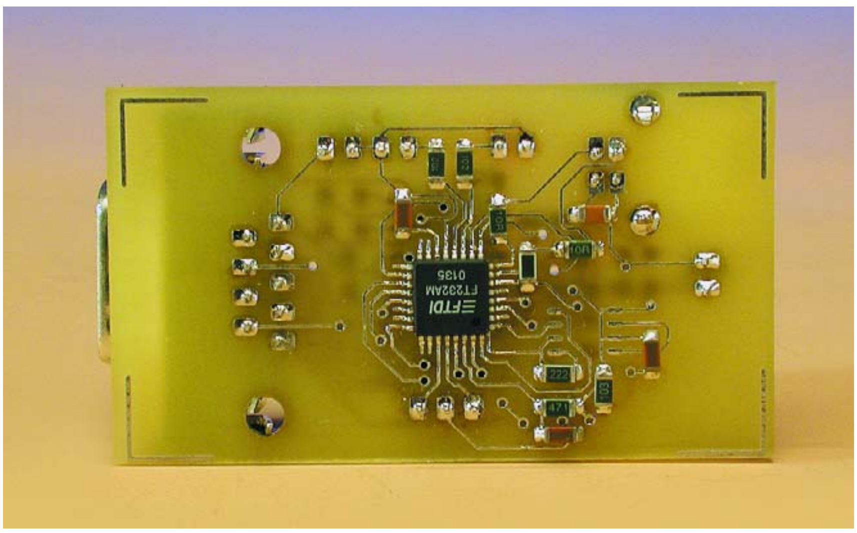


Figure 6. … and from below.

through-plated, and has components fitted at both sides! Those of you who have experience in dealing with SMDs will not be deterred by the extremely compact board layout. Fortunately, for those with less know-how in this area, *Elektor Electronics* recently published a number of useful articles on the subject of SMD soldering (see ‘References’ at the end of this article). Beginners are advised to prepare themselves by first reading the articles and then acquire practical experience from some (defective) boards and SMD parts to prevent disappointment with the present project.

The bare board is first populated with the SMDs and then with the ‘leaded’ components and sockets. One of the sockets is a USB Type ‘B’, for which the pinout is shown in **Figure**

7. The other variant of the socket, called USB 'A', is always at the PC or hub side, while type 'B' should always be inside the equipment reporting to the PC. Via the USB cable, the type 'A' socket supplies current to the 'B' socket at the equipment side. In our case, this current is used to power the USB-RS232 converter board. USB cables always contain 1-to-1 wire connections.

Ready-made modules

Those of you who dislike DIY constructing on PC boards, and SMDs in particular, will be glad to know that FTDI chip also supply ready-made USB-RS232 interface modules under part number DLP-USB232M, see their website for further details. These modules have all connections brought out to pins of a standard 0.6-inch wide DIP plug.

Software

Before connecting your circuit to the PC's USB port, give your soldering work a last, thorough, visual inspection, for which a magnifying glass will prove very useful.

Next, download the necessary drivers from the FTDI Chip website. Drivers are available for all popular Windows versions, as well as for the Macintosh and Linux platforms. Regarding the Windows drivers, versions are available with and without PNP (Plug & Play) support. The latter are called 'non-PNP'. The difference is small but essential. The drivers with PNP support should only be used when the peripheral connected to the PC via the USB/RS232 interface also installed its drivers using Windows PNP. In other words, in case of doubt, resort to the non-PNP drivers first to prevent problems. Typical problems you may encounter with the PNP supporting drivers include slow booting and erroneous identification of an USB/RS232 converter as a 'pointer device' which will result in failure of the mouse to operate as expected! For Windows XP, there's an additional XPNPNP tool that allows the Plug & Play function to be switched off for serial FTDI interfaces.

With the right driver securely stored in a subdirectory , you may start the installation process by connecting the USB/RS232 interface to a USB port on the PC. After a short delay, Windows will report that a USB device has been found. If there is no reference to a USB device, then there is a problem on our little board. In some cases, all you have to do is unplug the USB connector, wait a few seconds and insert it again, so try this first. When everything goes well, all you have to do is browse the system and click on FTDIBUS.INF to make the rest of the installation proceed.

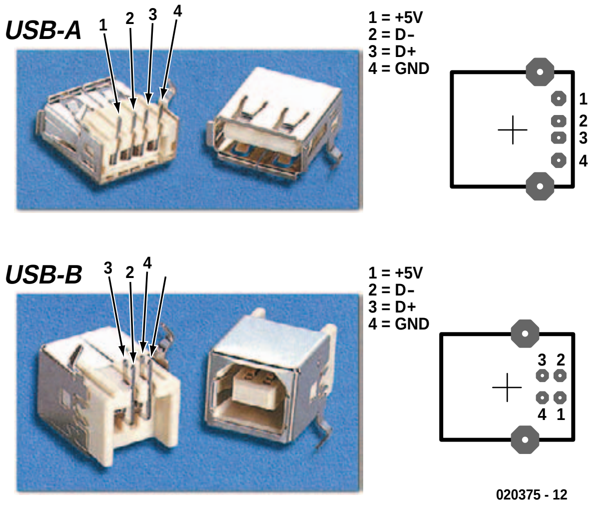


Figure 7. USB-A and USB-B socket pinouts.

Once finished, you will have a new serial port on your system, just check it using Start → Settings → Control Panel → System → Device Manager. If you open the 'Ports (COM & LPT)' folder you should see a new 'USB Serial Port (COMx)' as illustrated in **Figure 8a**.

During the installation, two drivers are actually installed that are linked with one another. One of these provides the virtual COM port you just found as the new device in the Device Manager. The other driver

Installation

As is well known, USB devices may be 'hot plugged' to the PC. The operating system will recognise the interface and request the associated driver. This may be downloaded from the Drivers and Utilities page on the FTDI Chip website. The so-called Virtual COM Port (VCP) drivers arrange for the interface to behave like an ordinary serial port. There are drivers for Windows, MacOS and Linux. Here, we will assume the Windows drivers are used.

Having installed the driver, the simulated COM port may be addressed by applications just like any regular serial port on the system. Higher programming languages like Delphi and C++ allow 'components' like Tcomport to be employed for the communication with the serial interface. If you do your own programming anyway, we'd recommend using the '**D2XX 'Direct' Drivers**' for Windows instead of the VCP ones. A Direct Driver **must** be employed to be able to program the external EEPROM!

The Windows VCP drivers copied form the FTDI website come in two flavours: with and without PNP. This has nothing to do with the interface proper (which is always recognised automatically by Windows). But only concerns the hardware connected to the interface (see the subheading 'Software').
The VCP driver download comes as a zip file containing drivers for Windows98, ME, 2000 and XP. The zip archive file is unpacked on the hard disk. The FTDI Chip website supplies extensive documentation, all the latest on the software and the installation process.

Windows will automatically launch its New Hardware Wizard when the interface is connected to the USB port on the PC. Next it will prompt you to point to the location where the drivers may be found. Browse and navigate to the folder containing the unpacked files. After a short delay, the Wizard will find FTDIBUS.INF and install the drivers and associated software for the interface.

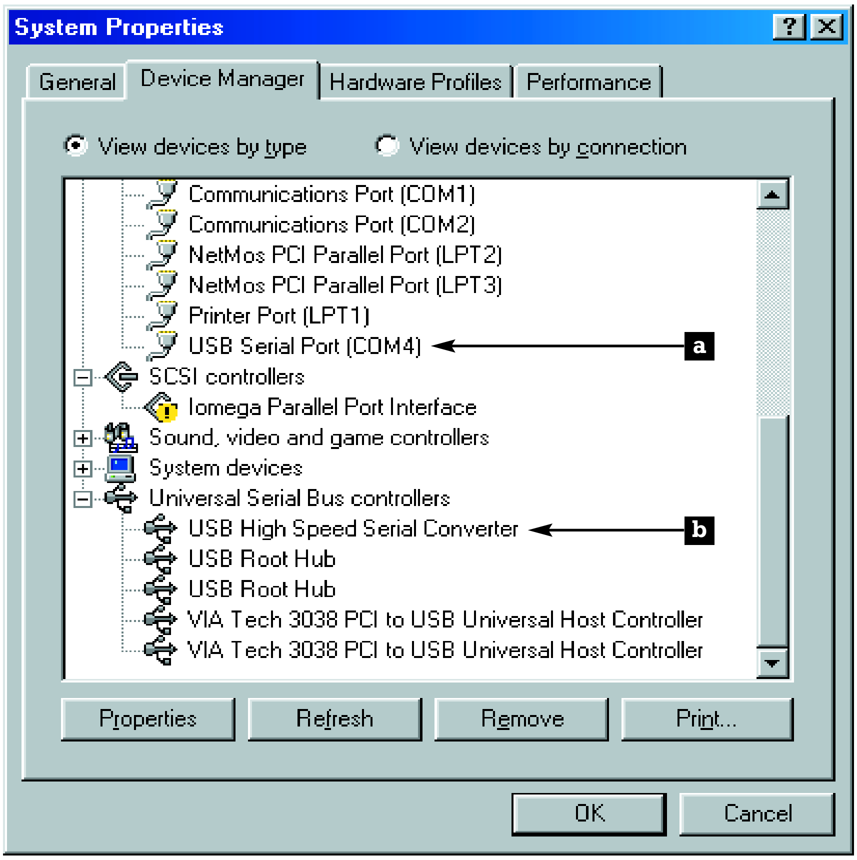


Figure 8. After the installation you should find a new 'USB Serial COM Port' under 'Ports' in the Device Manager (8a) and a new 'USB High Speed Serial Converter' under 'Universal Serial Bus Controller' (8b).

Note: *since writing this article we have been advised by FTDI Chip that the FT232AM chip has been superseded by the FT232BM.*

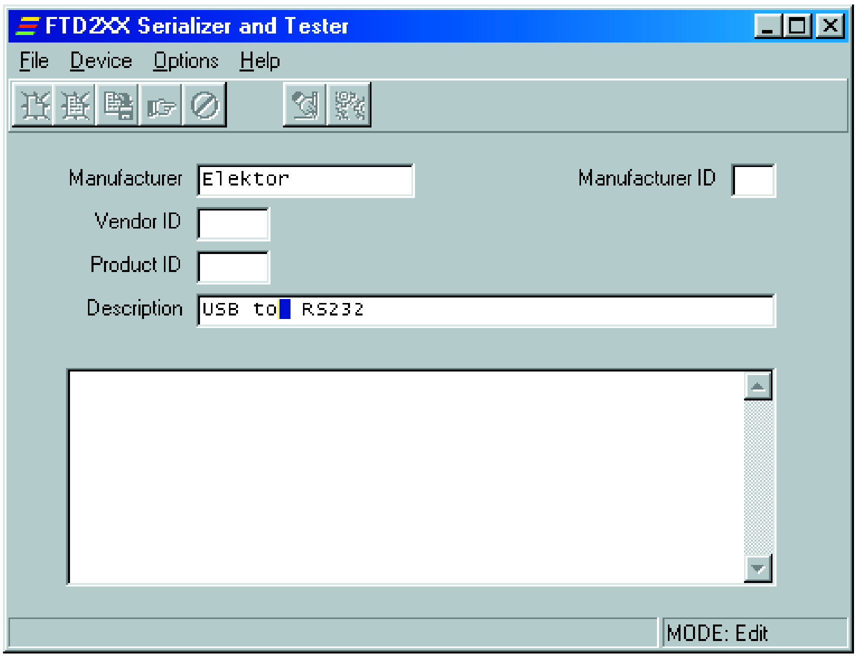


Figure 9. Utility program for (optional) programming of PID and VID information into the 93C46 EEPROM.

ensures that the USB side of the FT232AM appears as a USB device in the sub-folder 'Universal Serial Bus Controllers' (see **Figure 8b**).

Figure 9 shows the programming window of the tool available for programming the EEPROM connected to the FT232AM. This little tool may also be downloaded from the FTDI website. OEMs (original equipment manufacturers) may want to use this tool to program their own VID or PID. If you do not have a VID or PID, you may either omit them or omit the entire EEPROM. Yet another possibility is to simply resort to the PIDs and VIDs reserved by FTDI Chip. For the FT232AM, VID = 0403 and PID = 6001. A more extensive description of the actual uses of this option may be found in the programming instructions supplied by FTDI in the form of yet another free download.

Finally, we should mention that there are, of course, limits to the performance of a converter acting as a simulated RS232 port. Control of the data flow is essential to ensure trouble-free conversion between RS232 and USB, in order to prevent overruns occurring on the two buffers inside the FT232AM (128 and 384 bytes). Such situations would cause bytes to be lost in the conversion process and are most likely to occur at high data speeds when no handshaking is used.

(020375-1)

Literature:

– USB Driver Programming, Elektor Electronics October and November 2002.
– USB UART, Elektor Electronics December 2001 and January 2002.
– USB Interface, Elektor Electronics September 2000.
– SMD's? Don't Panic!, Elektor Electronics January and February 2003.

Free Downloads

– PCB layout (pdf file), on www.elektor-electronics.co.uk/dl/dl.htm. Select file number **020375-11.zip**, month of publication.
– FT232AM datasheets, drivers, info on ready-made modules, etc., www.ftdichip.com