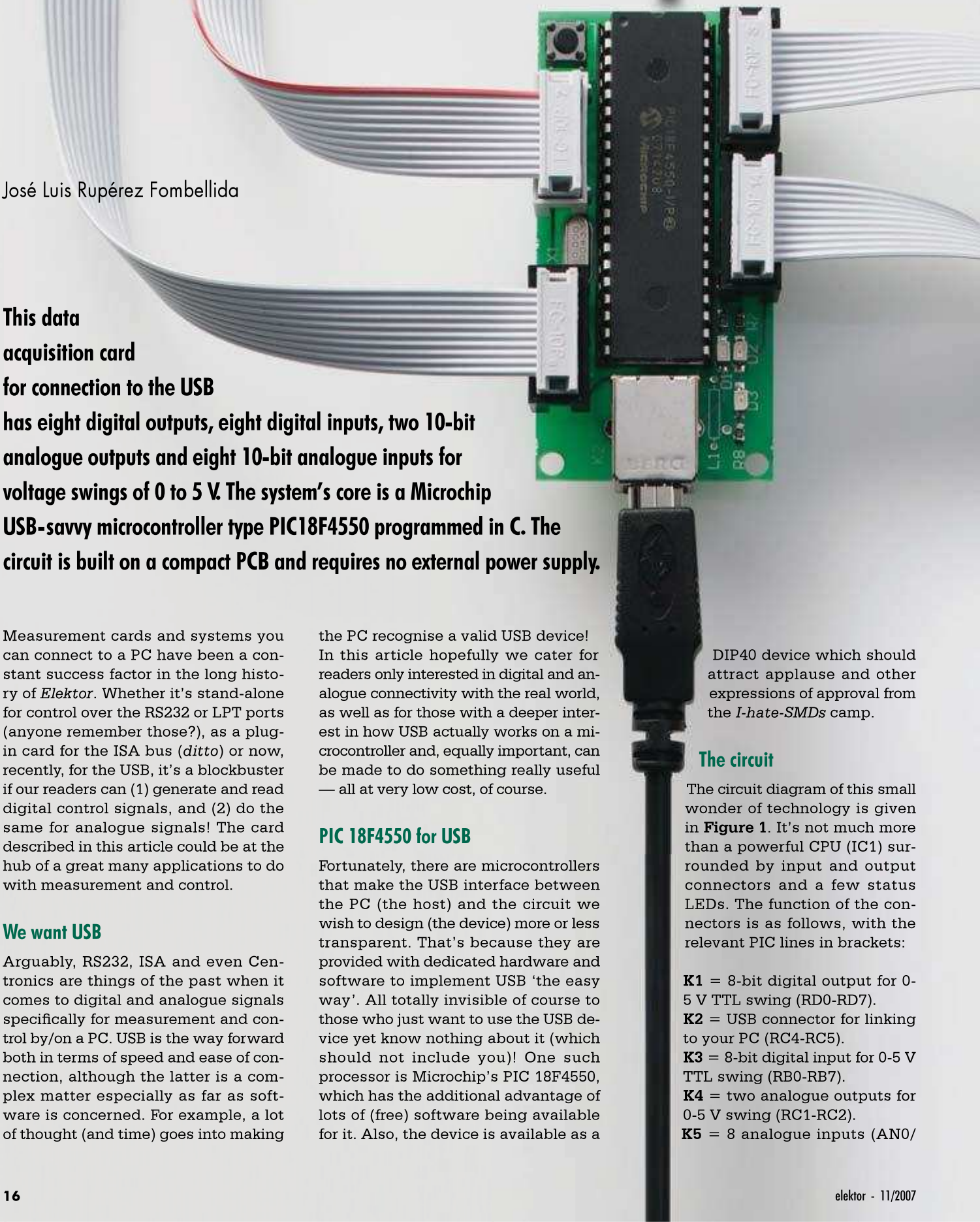
USB Data Acquisitio

José Luis Rupérez Fombellida



This data acquisition card for connection to the USB has eight digital outputs, eight digital inputs, two 10-bit analogue outputs and eight 10-bit analogue inputs for voltage swings of 0 to 5 V. The system's core is a Microchip USB-savvy microcontroller type PIC18F4550 programmed in C. The circuit is built on a compact PCB and requires no external power supply.

Measurement cards and systems you can connect to a PC have been a constant success factor in the long history of *Elektor*. Whether it's stand-alone for control over the RS232 or LPT ports (anyone remember those?), as a plug-in card for the ISA bus (*ditto*) or now, recently, for the USB, it's a blockbuster if our readers can (1) generate and read digital control signals, and (2) do the same for analogue signals! The card described in this article could be at the hub of a great many applications to do with measurement and control.

We want USB

Arguably, RS232, ISA and even Centronics are things of the past when it comes to digital and analogue signals specifically for measurement and control by/on a PC. USB is the way forward both in terms of speed and ease of connection, although the latter is a complex matter especially as far as software is concerned. For example, a lot of thought (and time) goes into making the PC recognise a valid USB device! In this article hopefully we cater for readers only interested in digital and analogue connectivity with the real world, as well as for those with a deeper interest in how USB actually works on a microcontroller and, equally important, can be made to do something really useful — all at very low cost, of course.

PIC 18F4550 for USB

Fortunately, there are microcontrollers that make the USB interface between the PC (the host) and the circuit we wish to design (the device) more or less transparent. That's because they are provided with dedicated hardware and software to implement USB 'the easy way'. All totally invisible of course to those who just want to use the USB device yet know nothing about it (which should not include you)! One such processor is Microchip's PIC 18F4550, which has the additional advantage of lots of (free) software being available for it. Also, the device is available as a DIP40 device which should attract applause and other expressions of approval from the *I-hate-SMDs* camp.

The circuit

The circuit diagram of this small wonder of technology is given in **Figure 1**. It's not much more than a powerful CPU (IC1) surrounded by input and output connectors and a few status LEDs. The function of the connectors is as follows, with the relevant PIC lines in brackets:

K1 = 8-bit digital output for 0-5 V TTL swing (RD0-RD7).
K2 = USB connector for linking to your PC (RC4-RC5).
K3 = 8-bit digital input for 0-5 V TTL swing (RB0-RB7).
K4 = two analogue outputs for 0-5 V swing (RC1-RC2).
K5 = 8 analogue inputs (AN0/

n Card

digital & analogue; input & output

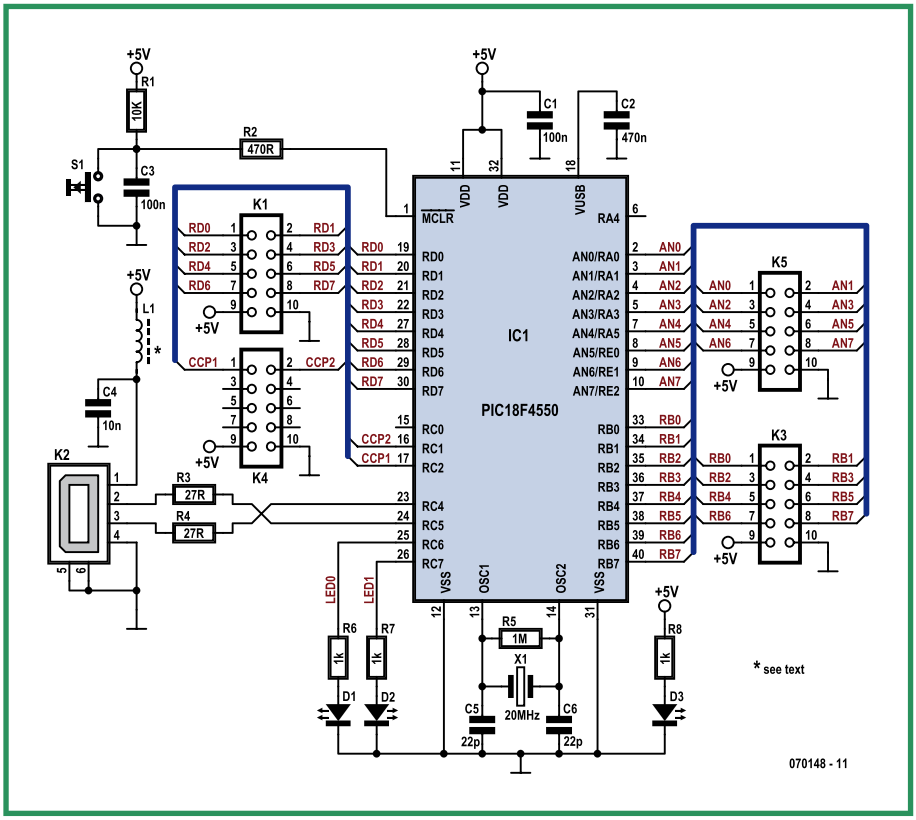


Figure 1. Circuit diagram of the data acquisition card with USB connectivity.

RA0-AN7/RE2) for 0-5 V swing.

Internal pull-up resistors are available on RB, the digital input port lines. The analogue outputs have a resolution of 10 bits each using PWM (pulsewidth modulation) at 2.9 kHz. If necessary these outputs can be filtered with a simple RC network. The DC output voltage V_o obtained after the filtering may be calculated from:

$V_o = 5D$ [volts]

where variable D is the duty cycle of the PWM, taking a value between 0 and 1.
The analogue inputs also have a resolution of 10 bits.

The oscillator in the PIC micro ticks at 20 MHz using quartz crystal X1 and the usual pair of small capacitors for the parallel loading, and a high-value resistor (R5) for the feedback.
Actually the microcontroller runs at 48 MHz, generated internally with the aid of a PLL and a frequency divisor from the 20 MHz supplied by the quartz crystal. The frequency of 48 MHz is an exact multiple of the USB bus speed (full speed, 12 Mbits/s).

Two status LEDs, D1 and D2, indicate the USB status. D3 is obviously the supply power indicator that lights when the card is connected to the USB port on your PC.
The circuit's supply voltage arrives via USB connector K2 and a small choke, L1, to suppress noise, with C4 assisting to that effect.
That effectively leaves components S1, R1, R2 and C3 at the $\overline{\text{MCLR}}$ input of the micro. Well it's just another wholly traditional Reset network.

PIC Firmware

Where there's simple hardware, there's a massive amount of software behind it all and usually lurking inside microcontrollers. The firmware (object code) the PIC is faithfully executing was created by the author using two free software tools from Microchip: IDE MPLAB V7.5 and C18 Student Edition V3.02. The Microchip website has instructions for installation and use of both programs.
The source code of the firmware is different from the Microchip original. All project software is available as a free download # **070148-11.zip** from our website at www.elektor.com. You will find at least three folders in the archive file: 'driver', 'firmware' and 'PC'. The content is an Aladdin's Cave for fans of C, PICs and USB (and that should cover a lot of our readers!). A piece of C code is shown in **Figure 2;** it's these PIC fuse settings you'll need to know if you're not buying the chip ready-programmed from Elektor.
The firmware file contains the whole project and the result of its compilation called TAD_v1.hex. The microcontroller must be programmed with this file.
Those of you interested in the deeper workings of USB should know that the connectivity implemented on the card described on this article is de-

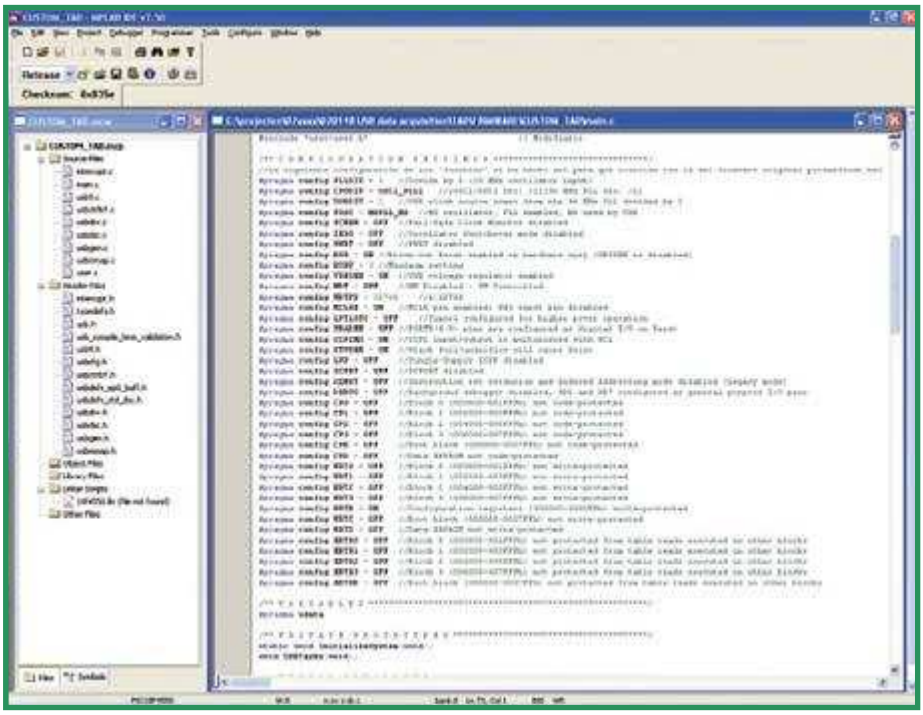


Figure 2. RU on MPLAB too? Lots of Elektor readers are. The C code for the project also contains useful information on the PIC fuse settings — an ongoing source of confusion to many microcontroller enthusiasts (and not just those on PIC).

fined by firmware in the PIC18F4550. The following building blocks are used: BUS POWER MODE; CUSTOM CLASS; FULL-SPEED (12 MBIT/S) and INTERRUPT TRANSFER.

Construction

The circuit is built on a compact double-sided printed circuit board of which the component placement at both sides is shown in **Figure 3**. Some empty space has been left at the short sides of the board be able to secure it with screws.

Although building up the board will be mostly plain sailing for experienced readers, some remarks may be in order for those just starting out in USB land with the present card.

The opening in the collar of each box-header is an orientation aid and should be at the edge of the board to enable an IDC connector on a flatcable to be plugged in.

SMD components are fitted **at both sides of the board**, so carefully study the two overlays to establish the correct position and of each part, as well as the orientation in the case of the SMD LEDs.

We recommend fitting the programmed PIC18F4550 in a good quality 40-way DIL socket. Watch the orientation of the large IC: pin 1 is near the reset switch S1.

L1, finally, is a ferrite bead with three or four holes through which a piece of enamelled or other stiff wire is pulled. A ferrite bead with one hole and three turns of wire through it should also work. The final inductance of the RF choke so made is uncritical.

To avoid possible damage to the PC, verify that there are no short circuits or other problems in the pins of USB connector K2.

First connection

Once the card is fully populated (and the microcontroller properly programmed), connect it to a PC by means of a standard USB cable. The power LED D3 lights and one of the LEDs D1 and D2 flashes while the other remains off. At the same time the PC will tell you that a new USB device has been connected and that a driver is required. Tell Windows where the driver is located (folder driver\mchpusb.inf). Once the driver is installed the USB status LEDs flash alternatively. The card is then ready for use.

VID/PID (Product ID/Vendor ID)

All USB devices have a unique combination consisting of two numbers so that no two equal devices exist. The first number, VID, identifies the manufacturer of the device and the second, PID, gives the product identifier.

The combination used in this project uses as VID the one of Microchip and

Data acquisition card trainer bench

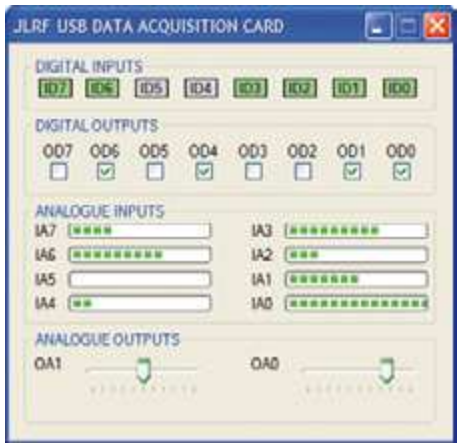
The author has developed four simple add-on cards for testing the data acquisition card for the following functionalities:

1. LED card: 8 LEDs to visualize the digital output.
2. Pushbutton and switch card: 4 pushbuttons and 4 switches to exercise the 8 digital inputs.
3. LED voltmeter card: two LEDs that change their brightness according to the two analogue outputs.
4. Potentiometer card: 8 potentiometers for testing correct operation of the 8 analogue inputs.

The data acquisition card and the four cards are shown in the picture. Although PCBs are shown and the author has the schematics and board designs in OrCAD format, these cards should be easy to build using Vero board.

To test the whole system, a program was developed in C++ CLR, for which the (free) Visual C++ 2005 Express compiler was used. This program is based on examples from Microchip.

A screendump of the program is shown here. This software is included in the archive file for the project.



as PID the one of a demo card of the PIC18F4550 Microchip. If the USB Data Acquisition Card is used for commercial purposes, it is essential to obtain a different VID/PID set of numbers—this can be done, for example, through www.usb.org or through Microchip.
This new combination should be included in the source code of firmware that would be compiled again in order to obtain an updated .hex file with which the microcontroller is programmed.
The PC software would also have to be modified in the same way, since firmware and software must have the same VID/PID. Finally, the driver 'mchpusb.inf' file would be modified.

Precautions

Some general precautions must be mentioned. All expansion connectors K1, K3, K4 and K5 include +5 V and ground to power any cards that can be connected to them. Great care must be taken to prevent short circuiting these terminals and not drawing more than 100 mA from any of them. Also, remember that these terminals are directly connected to the +5 V and ground of the USB port of your computer (and you do not want that to take damage — avoid using family or kiddies PCs in any case).
If you need more current for a certain application, consider the use of an external power supply, joining only application and USB Data Acquisition Card grounds.
Some precautions for the digital inputs

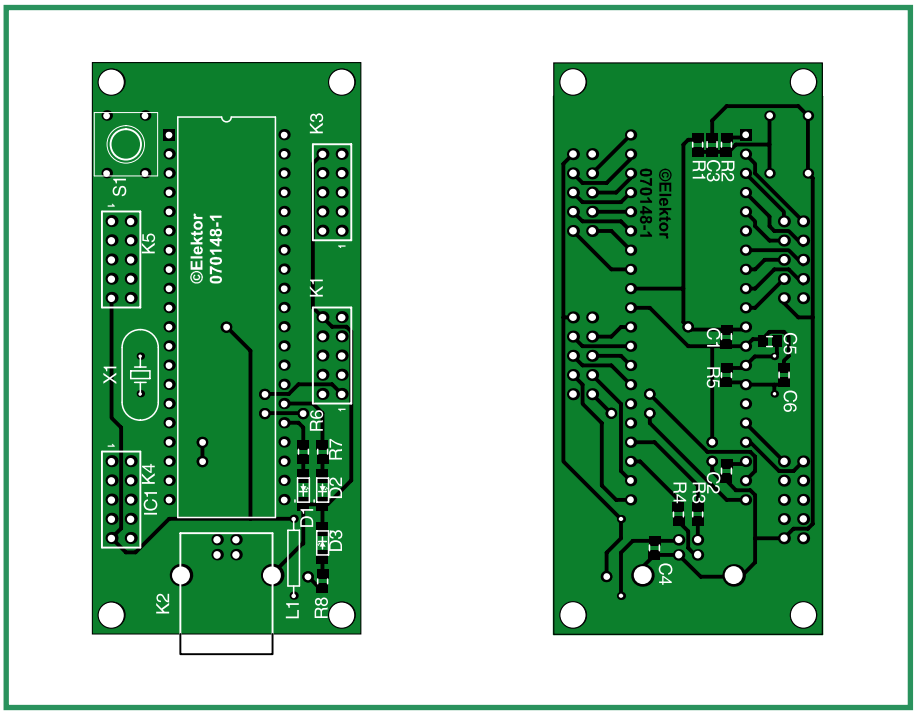


Figure 3. Top side and bottom side component placement for the PCB. The PCB artwork is a free pdf download from our website but bear in mind that the board is double-sided and through-plated.

COMPONENTS LIST

Resistors
(all SMD 0805 case)
R1 = 10kΩ
R2 = 470Ω
R3,R4 = 33Ω
R5 = 1MΩ
R6,R7,R8 = 1kΩ

Capacitors
(all SMD 0805 case)
C1,C3 = 100nF
C2 = 470nF
C4 = 10nF
C5,C6 = 22pF

Semiconductors
IC1 = PIC18F4550 I/P, programmed, Elektor Shop # **070148-41**
D1,D2,D3 = LED, SMD case 1206

Miscellaneous
K1,K3,K4,K5 = 10-way boxheader
K2 = type-B USB connector, PCB mount
X1 = 20MHz quartz crystal
L1 = VK200 or small ferrite bead with 2-4 turns thin enamelled copper wire
S1 = pushbutton, PCB mount, 6mm footprint
DIL40 socket for IC1
PCB (bare), Elektor Shop # **070148-1**
Project software, file # **070148-11**, free download from www.elektor.com.

Files & file locations

The complete project of the application for the PC is in PC\ TAD_V1_win\ folder and its name is TAD_V1_win.vcproj. The compiled program is in PC\TAD_V1_win\Release folder and its name is TAD_V1_win.exe (for the program to work, the dynamic link library mpusbapi.dll created by Microchip must be in that same folder). The executable one needs the .NET Framework. It is highly recommended to have the operating system updated by means of Windows Update.

If you wish to modify the project to adapt it to your requirements it is necessary to install the Visual compiler Microsoft C++ 2005 Express and update it with Service Pack 1: Visual C++ 2005 Express SP1. Later we will install Microsoft Platform SDK for Microsoft Visual C++ 2005 Express. This serves to develop WIN32 applications, necessary in this case to access the DLL mpusbapi.dll). All of it is free and can be downloaded from the Microsoft website. There you will also find instructions about installation and examples.

(K3): Do not apply voltages below zero or higher than 5 volts to avoid damage to the PIC microcontroller.
On the digital outputs (K1): each line can supply a maximum current of 25 mA for logic High or Low levels.
On the analogue inputs (K5); the same as with the digital inputs.
On the analogue outputs (K4): each line can supply a maximum current of 25 mA for logic High and Low levels of the PWM signal.
Finally, the sum of all currents of all the digital and analogue outputs must not exceed 200 mA.

Work in progress…

The USB data acquisition data card has a lot of potential and the author has developed, and is busy developing, the following application cards:

1. Triac card for the 8 digital outputs, to be able to control mains-powered loads in a comfortable manner. This card is isolated using optotriacs.
2. Resistor-to-voltage converter card supplying a voltage proportional to the input resistor. This voltage is applied to the digital input.
3. Voltmeter card with LED bar readout.
4. Speed control of a DC motor. This card controls the speed and direction of a motor through the analogue outputs.
5. Driver card for stepper motors, capable of microstepping through the digital outputs.
6. Distance sensors card using the analogue inputs.
7. Relay card under control of the digital outputs.

Elektor and the author welcome other applications you may have developed. Let us know!

(070148-I)

About the author

The author is a telecommunications technical engineer working as a teacher of electronics in a professional school in Madrid since 1984. He is a keen electronics enthusiast. He developed this card to enable his students to control small robots from the USB port in a PC by programmes written in C code.

Follow these steps

The PC software available for this project should be relatively easy to install use and/or adapt if you follow these steps.

1. Install Visual C++ 2005 Express:
http://msdn2.microsoft.com/en-us/express/aa975050.aspx)

2. Install Visual C++ 2005 Express SP1:
http://msdn2.microsoft.com/en-us/express/aa975050.aspx

3. Install PSDK: Microsoft Platform SDK for Microsoft Visual C++ 2005 Express:
http://msdn2.microsoft.com/en-us/express/aa975050.aspx

4. Update the operating system using Windows Update.

5. Tell Visual C++ to use PSDK.

The sequence to do so suggested by Microsoft is given below.

5.1 Update the Visual C++ directories in the Projects and Solutions section in the Options dialogue box.
Add the paths to the appropriate subsection:
Executable files: C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Bin;

Include files: C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Include;

Library files: C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Lib.

5.2. Update the corewin_express.vsprops file.
One more step is needed to make the Win32 template work in Visual C++ Express. You need to edit the corewin_express.vsprops file (found in C:\Program Files\Microsoft Visual Studio 8\VC\VCProjectDefaults) and change the string that reads:

AdditionalDependencies="kernel32.lib" to:

AdditionalDependencies="kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib".

5.3. Generate and build a Win32 application to test your paths.
In Visual C++ Express, the Win32 Windows Application type is disabled in the Win32 Application Wizard. To enable that type, you need to edit the file AppSettings.htm file located in the folder "%ProgramFiles%\Microsoft Visual Studio 8\VC\VCWizards\AppWiz\Generic\Application\html\1033\".

In a text editor, comment out lines 441 - 444 by putting a // (double slah forward) in front of them as shown here:

// WIN_APP.disabled = true;

// WIN_APP_LABEL.disabled = true;

// DLL_APP.disabled = true;

// DLL_APP_LABEL.disabled = true.

Save and close the file and open Visual C++ Express.