USB 25IO Digital/ADC/PWM - V1.1

Guide

The Mirrorbow USB 25IO Digital/ADC/PWM allows easy interfacing of custom devices from a PC without requiring any USB knowledge.

With the loss of serial and dedicated parallel ports from PCs it has got much harder to develop all those fun, challenging and wonderful projects that were accessible a few years ago. USB has been great for the average PC user but anyone wanting a simple interface to bridge their computer program with straightforward electronics has realised that in most cases, it is more complicated than the project itself, requiring in depth knowledge of windows DDK and substantial firmware! The Mirrorbow USB 25IO removes all of this complexity, allowing you to concentrate on your project rather than the interfaces.

The Mirrorbow USB IO board interfaces via USB, but pretends to be a simple com port. Hence the host PC software simply sends commands over the serial interface, which is easily programmed and understood.

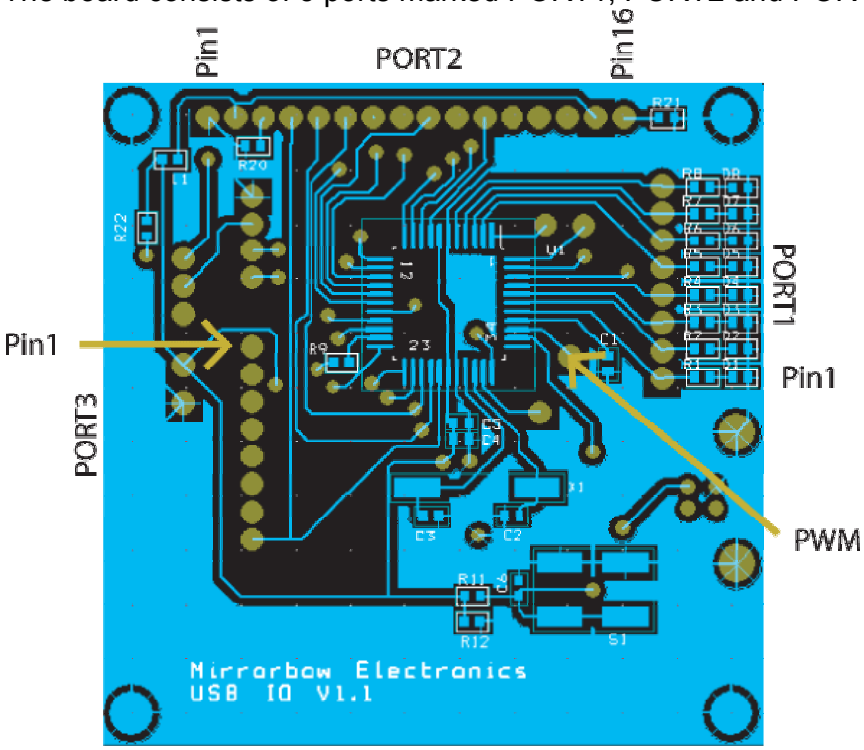
Warning: *As with any bare electronics It is wise to take static precautions when handling this board. Overloading IO outputs or exceeding 5V on inputs may cause permanent damage to the board. The maximum loading for any single output is 10mA, however total loading on all IO pins must not exceed 100mA. The 5V supply provided on the board is the USB supply, care must be taken not to exceed the available USB power which on a desktop PC is usually at least 400mA, but on a laptop may be much less.*

Compatible with Windows 2000 and Windows XP

Mirrorbow Electronics

The Board

The board consists of 3 ports marked PORT1, PORT2 and PORT3.



Ports 1 and 3 are simple 8 bit ports, however Port 2 is extended to include power, 3 data bits from Port 3 as well as all 8 data bits from Port 2. This is to make it easier to connect a daughter board with the maximum flexibility. Port2 can also be turned into a servo controller for up to 8 servos.

Port3 bits 0 and 1 can be configured to convert the analogue voltage to a digital number (ADCs)

	PORT1	**PORT2**	**PORT3**
Pin1	1D0	GND (0V)	3D0 /A0
Pin2	1D1	+5V	3D1 /A1
Pin3	1D2	Do not connect	3D2 /A2
Pin4	1D3	3D7 (PORT3 copy) /A6	3D3
Pin5	1D4	3D6 (PORT3 copy) /A5	3D4 /A3
Pin6	1D5	3D5 (PORT3 copy) /A4	3D5 /A4
Pin7	1D6	2D0 (also Servo0) /A7	3D6 /A5
Pin8	1D7	2D1 (also Servo1)	3D7 /A6
Pin9		2D2 (also Servo2)	
Pin10		2D3 (also Servo3)	
Pin11		2D4 (also Servo4) /A8	
Pin12		2D5 (also Servo5) /A9	
Pin13		2D6 (also Servo6) /A10	
Pin14		2D7 (also Servo7) /A11	
Pin15		Do not connect	
Pin16		Do not connect	

Installation

Put the CD supplied in your PC's drive, then plug in the Mirrorbow USB IO board using a standard USB cable (not supplied). Windows will recognise the new device and prompt for a driver. Select the Mirrorbow CD using browse, and then hit next. Windows will then install the required driver.

Simple Commands and Testing

After installation above, go to My Computer and View System Information (top left in winxp, or right click MyComputer in win2k). Select the Hardware tab, and the push the button marked "Device Manager". This will show a list of the devices on your system. Scroll down to "Ports (COM & LPT)" and click on the + to open the folder. You should now see the USB IO Port. Note the COM port allocated to the Mirrorbow USB IO board as you will need this later.

Command Set:

Command	Format	Returns	Description
DIR	DIR1 00	A *as an acknowledge*	Sets Port Direction: The port number (1,2 or 3) comes directly after the command, followed by a <space> and the hexadecimal value to write. A bit set to 0 is an output, to 1 and input. DIR3 00 sets all port 3 to output DIR3 0F sets the top of port 3 to output and the bottom 4 bits to input
OUT	OUT1 AA	A as an acknowledge	Outputs an ASCII hexadecimal value to a port: OUT2 55 puts the hexadecimal number 55 onto port 2
IN	IN2	Byte as two characters	Inputs the value of the port: IN1 The board responds with an ASCII hex value e.g. 0F IN3 INA returns either 2 ADC readings or 12 depending on the command below. E.g. when 2 AD inputs are active, 11112222 is returned, where 1111 is the ADC value on pin1 and 2222 the ADC from pin2. 0000 is 0V and 03FF is 5V.
AD	AD+ or AD- or AD*	A as an acknowledge	AD+ turns on the two ADC1 and ADC2 analogue to digital converter inputs, AD- turns off the ADC function. AD* turns on all 12 ADC inputs

Formatting commands: Note the commands always start with the command and port number without spaces, e.g. (OUT1,DIR2,IN1). When writing to the board with OUT or DIR commands there must be ONE space between the port number and the hexadecimal value to be written e.g. OUT1 AA
At any time an invalid input at a position causes the USB IO board to start looking for a fresh command, so you can use a carriage return or two spaces for example, to ensure you are at the start of a command. It is good practice when writing software to send a carriage return or two spaces to the board before sending a command, this ensures that the board is always ready to receive the command and cannot get out of synchronisation.

Simple test using Windows HyperTerminal: For this simple test open windows Hyperterminal (Windows, applications, communiations, HyperTerminal). Choose the COM port of the Mirrorbow USB IO Board, and select 115200,N,8,1. Then under ASCII options click "send LF" and "Echo". You are now ready for testing.
(A full explanation of the use of HyperTerminal is out of scope for this guide).

PORT1 is equipped with LEDs for simple testing.
Now type:
DIR1 00

If you are communicating with the board you should see an A appear after the command thus:
DIR1 00A
This is an *acknowledge*. DIR1 sets the direction of PORT1 to all outputs, 0 is output, and 1 input. You can also use DIR2 and DIR3 for the other ports.

Now you want to output a value. *Use the command:*

OUT1 AA
You will get an *acknowledge* so will see **OUT1 AAA**.
AA is the hexadecimal value which is then sent to the port, and should light alternate LEDs. Try OUT1 55, to see the opposite pattern.

To input data, use **IN1**. This will immediately give a response of the ASCII hexadecimal value on the port (no ack is given for input as the response is enough). Before you do this, set **DIR1 FF** to set all inputs on that port. You should get **IN100** for example. Where 00 is the value read. Try connecting pin1 on this port to 5V and type **IN101**, where 01 is the value read.

And for the ADCs:
AD+ turns the ADCs on for pins 1 and 2 of Port3. You will see **AD+A** with the acknowledge.
IN3 will now return the ADC values as well as the digital value of the port.
You will see **IN32003FF0000** in this case the 6th bit (bit 5) is set ADC0 is at 5V (3FF) and ADC0 at 0V.

PWM Operation:

Prescale "T"	valid values are: 04 = prescale of 1 05 = prescale of 4 07 = prescale of 16
Period "P"	00 to FF in hex
Mark/Space Ratio "M"	00 to FF in hex (must always be less than “P”)

For a 50/50 mark/space ratio P should be twice M + 1, see formula below. For 25% mark, P is 4xM +1

The output frequency is calculated with the following formula:

Fout=12000000/((P+1)*pre-scale)	e.g. for P=FF and prescale of 16 Fout=12000000/((256)*16)=2.93kHz

Inputting the values on the commands line:
To set the Period to FF hex (256) enter PFF, you will see PFFA with the Ack
To set the Mark to 50%, enter M80, you will see M80A with the Ack
To set the prescale to 1, enter T04, you will see T04A with the Ack (note valid values are only T04,T05 or T07, other values will cause unpredictable results.

Servo Mode: Turns Port2 into 8 servo outputs, with programmable pulse widths from under 1mS to greater than 2mS, designed for driving standard servos. Before using the servo port, ensure that DIR2 00 command is sent, to set the port to outputs and S+ is issued to turn on servo mode. Default duration for each servo is 00 which is off, logic 0 state (command S1 00 can be used to turn say servo 1 off after it was previously active). Example values are S0 C0 which places a 2mS pulse on servo output 0 (Port2 bit0), S7 05 places a 1mS pulse on servo output 7 (Port2 bit7). Each increment in the hex value represents an increase of 5.3uS. Total cycle time seen by each servo will range from around 16mS to 24mS (or 26mS if you set all the servo output beyond 1mS (FF)).

S	S+ or S- or S0 C0	A as an acknowledge	S+ turns Port2 into Servo Mode (see section under Servo Mode), while S- turns Port2 back to IO mode. S7 05 places a 1mS pulse on servo output 7 (Port2 bit7)

Host Software

An example of host software is provided on the accompanying disk. It is currently setup to use COM5 so must be changed to reflect the COM port the Mirrowbow USB 24IO appears in your system. The software is documented, please refer to the code for more information.