

# USB Digital PIO Card

## DEVELOPMENT KIT

*A low cost solution for controlling 24 digital lines*

This project can be used to control 24 digital lines. The well-known IC 8255 is used as the PIO Controller.

The Card can be used for a variety of purposes. For instance, it can help you speed up your software development time, and it can assist you in controlling your own projects by using the latest USB technology.

The USB PIO Card may also be used to control modules such as:

- ADC converters
- DAC converters
- DTMF transceivers
- PLL generators
- DDS synthesizers
- Relay Cards
- Stepper Motor Cards
- Switch Cards
- LED Indicator Cards
- And much more...

Softmark supplies its universal USB\_OCX component that allows you to easily control all these modules.

The project features three ports: A, B and C. These ports can be set as either inputs or outputs. Port C can be set both as an input and output. The USB\_OCX component will be used to write to and read from the ports. With this component you can fully control the USB PIO Card and all modules from your application.

### **Features:**

- 24 digital input/output lines
- 3 bi-directional 8-bit ports programmable by software
- digital output capable of driving relays
- 2,500V isolation between your computer and the controlled hardware
- USB\_OCX tool to control the USB PIO Card from your own application (only in full version of the kit) with free data acquisition examples

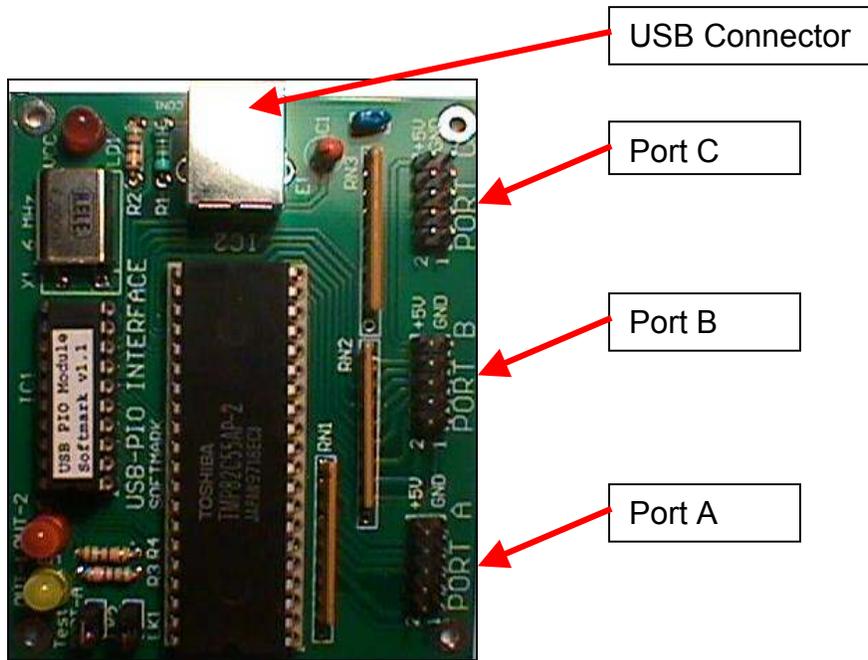


Figure 1. 24-line input/output interface controlled via the USB bus.

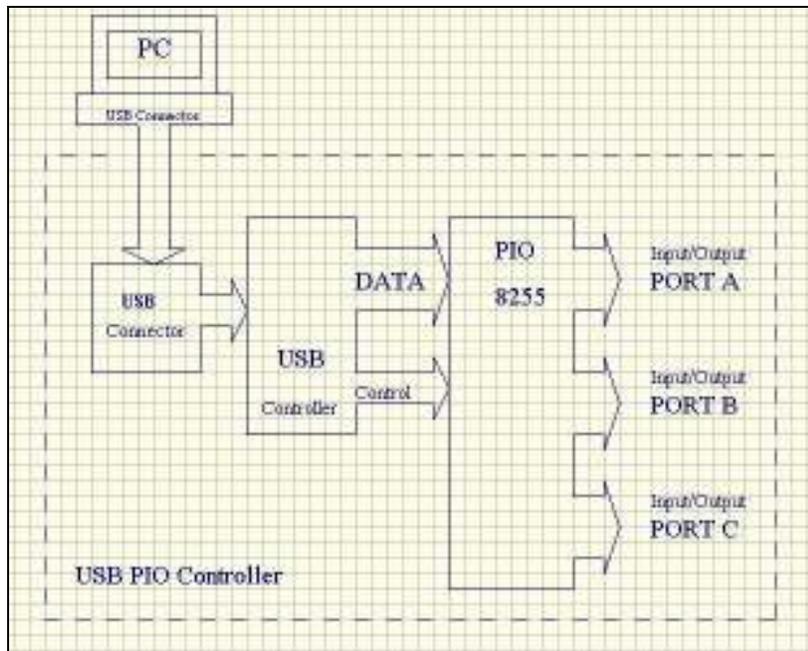
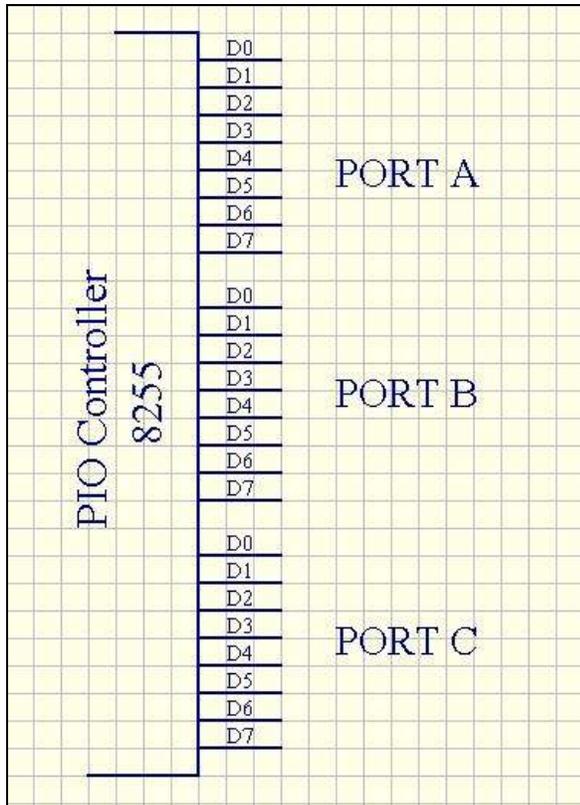


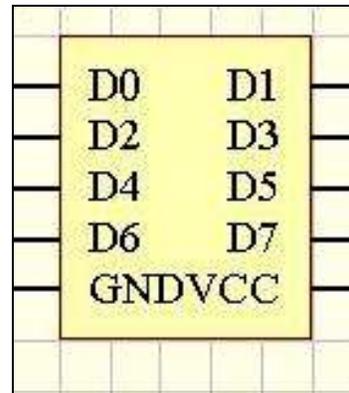
Figure 2. Block diagram of the USB PIO Card.

The USB PIO Card is connected to the PC via cable (type A to B). Ports A, B and C are bi-directional. They can be set as inputs or outputs via control lines from the USB Controller. Figure 3 below gives a more detailed picture of the ports.

All ports are 8-bit TTL lines. The USB PIO Card is powered by the USB bus line. The maximum current from this line is 100mA. The current taken by the Card is about 10mA.



← **Figure 3.** 8 digital line ports. These lines can be easily set (for high or low level) from your application using Softmark's USB\_OCX component.



**Figure 4.** Port connector (pinout) for ports A, B and C – PCB top view.

## How to write to a particular port

The simplest way to write to a port is to connect an LED to a particular port line to test the port. Softmark has developed an LED Interface to read the status of a particular port. Figure 9 shows how the LEDs are reading the status of particular port lines. The LED Interface is connected to the USB PIO Card with a ribbon cable as shown in figure 7.

Note: ports have to be set to output line mode by software. Data can be transferred from your application via the USB\_OCX to the hardware by executing just a few lines of code:

```

' Configure port A as output
Wr_PortP1(conf) ' if conf=128 then all ports are set as outputs
' Set all data on port A to High
Wr_PortP0(255)
' Set all data on port A to Low
Wr_PortP0(0)
' Set Line D0 to High, the rest to zero
Wr_PortP0(1)

```

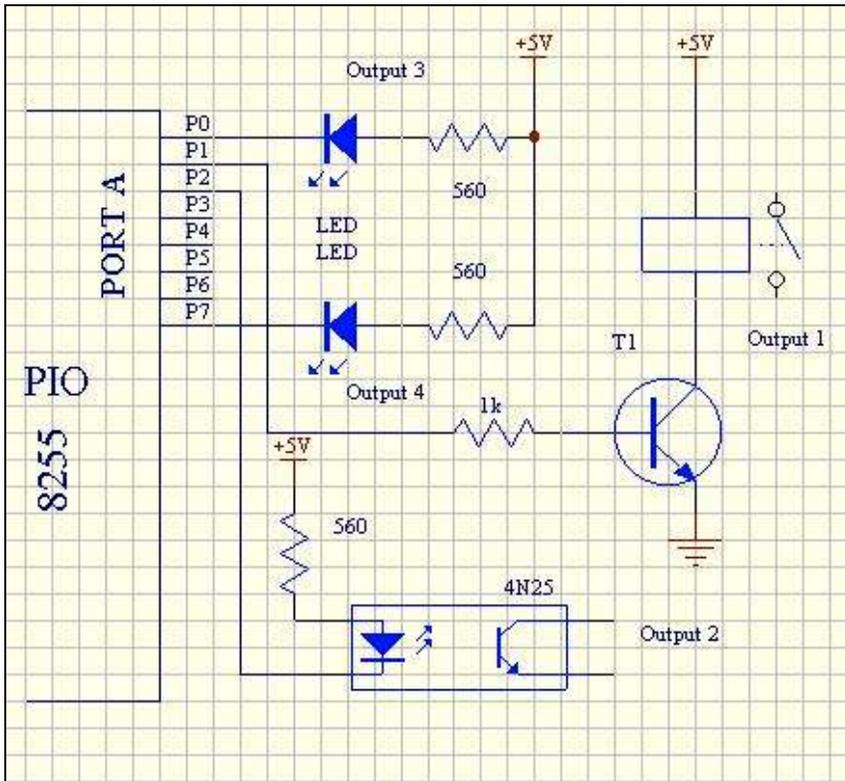


Figure 5. Output indicators connected to port A.

Softmark supplies four different I/O interfaces for use with this USB PIO Card:

- PIO Output Interface (with 8 LEDs) (“LED Interface”)
- PIO 8-Relay/Stepper Motor Interface, based on the 8-bit ULN 2803 driver
- PIO Input Interface with switches (“Switch Interface”)
- PIO Input Interface (with opto isolation)

(These interfaces are assembled only in the full version of the kit. Normally, only bare PCBs are attached to the USB PIO Card kit.)

The Switch Interface (with 8 switches) can be used to set the input lines of the USB PIO Card. This interface can be replaced or modified by your own switch interface, for example, a switch to detect if a door was opened.

## LED Interface

The LED indicators display the status on the output lines.

The 8 LED Interface can control the status on any port (A, B or C) if these ports are configured as outputs. This interface will be helpful if you are testing or debugging your software or it can just be used as an interface to display the status of a particular port. The LED Interface can also be replaced by your own device, such as a buzzer.

See Figures 7 and 8 below which show the LED Interface connected to ports A and B.

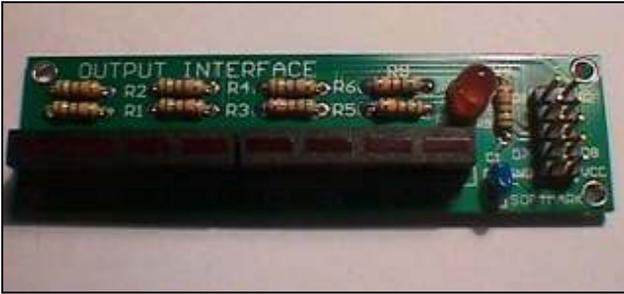


Figure 6. LED Interface.

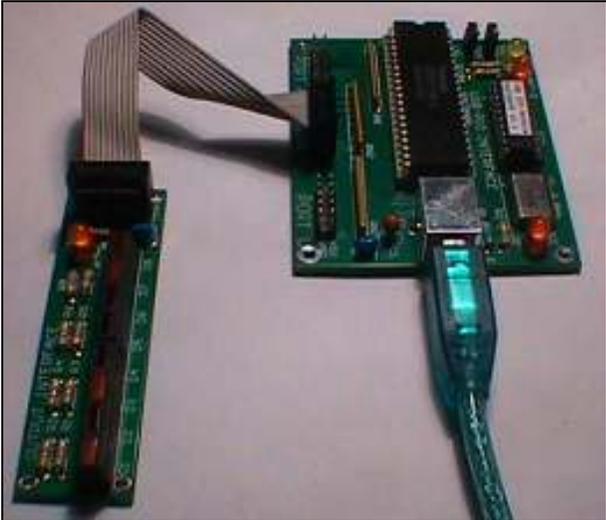


Figure 7. LED Interface connected to port B (via ribbon cable).



Figure 8. LED Interface connected to port A (direct connection to the USB PIO Card).

With the LED Interface, any changes on the connected port will be easily visible. You can connect up to 3 LED Interfaces to each port. Figure 9 below shows a diagram for an 8-bit LED Interface.

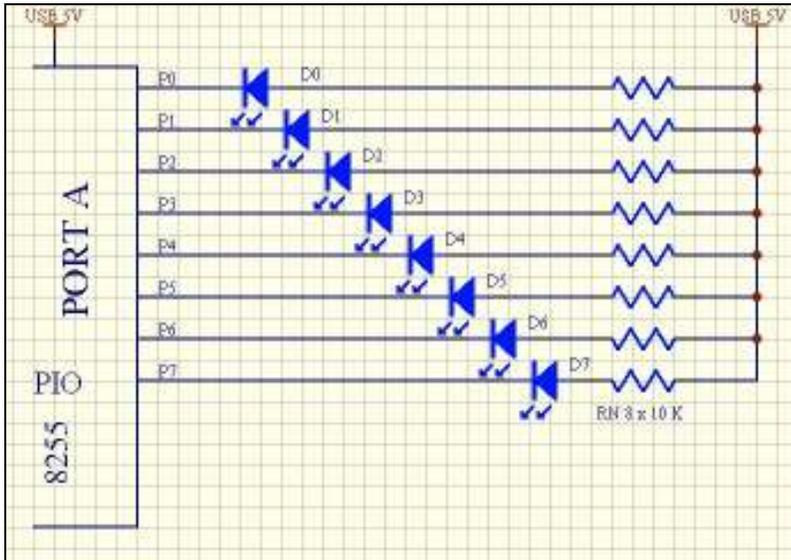


Figure 9. LED Interface (with indicators) connected to port A.

## 8-Relay/Stepper Motor Interface based on the 8-bit ULN 2803 driver

The output from a particular port can be used to drive, for example, a relay or a stepper motor. This interface can be used to drive devices which use no more than 50V and their output current is maximum 1A from one line. A comprehensive diagram for this Interface is supplied with the Interface itself.

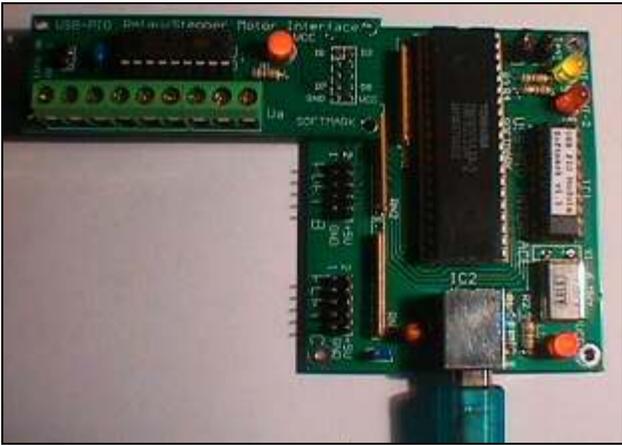
As driving relays or stepper motors will require the use of current drivers, the USB bus doesn't give you enough current. For that reason, you should have your own power supply.



Figure 10. A Relay/Stepper Motor Card Interface.



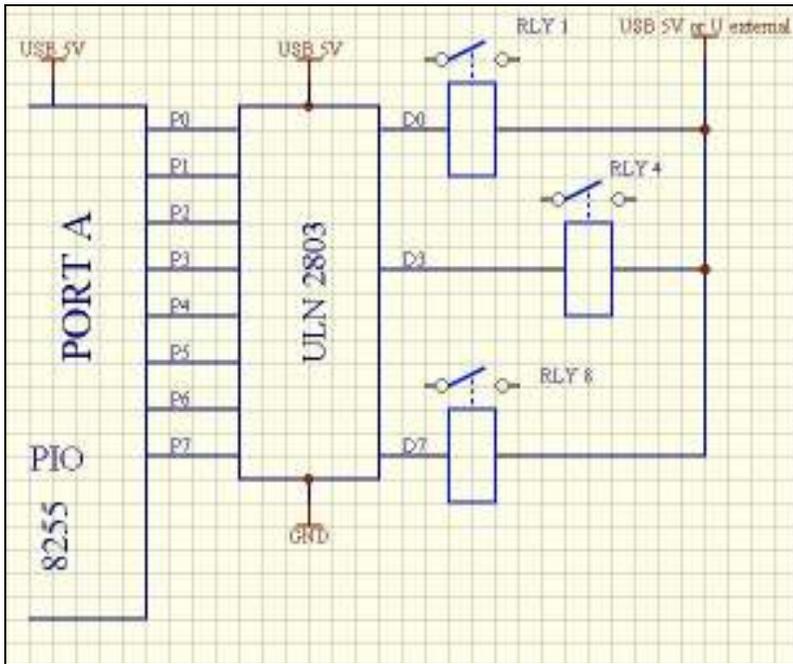
Figure 11. A Relay/Stepper Motor Interface connected to the USB PIO Card (via ribbon cable).



**Figure 12.** A Relay/Stepper Motor Interface connected to the USB PIO Card (direct connection).

The output from a particular port will drive the current driver based on the ULN 2803. Open collector outputs from the driver can be used to directly drive relays or stepper motor coils.

Note: the maximum current is 1A and maximum voltage which you can supply is 50V. If you will only use a 5V relay, the voltage can be directly taken from the board. No extra power supply is required. This voltage is supplied by the USB bus and the maximum current cannot exceed 100mA. If more current is required, you have to supply external voltage to the Relay/Stepper Motor Interface. Figure 13 below shows more details for these modules.



**Figure 13.** The ULN 2803 driver connected to the USB PIO Card.

Note: The port has to be set to output mode by software. Data can be transferred from the USB\_OCX to the hardware by executing the following:

```

' Configure port A as output
Wr_PortP1(conf) ' if conf=128 then all ports are set as outputs
' Set RLY 1 to on
Wr_PortP0(1)
' Set RLY 1 to off
Wr_PortP0(0)
' Set RLY 1 and RLY 8 to on and RLY 4 to off
Wr_PortP0(129)

```

You can add more relays and they can be set individually. You can use up to 24 relays in total connected to ports A, B and C.

### How a Relay/Stepper Motor Interface controls stepper motors

A stepper motor can be controlled via the USB PIO Card. You can control two stepper motors from one port. Only four lines from a port have to be used to drive one stepper motor. The maximum voltage supplied to a card can be 50V and maximum current per coil is 1A. To calculate the current taken by a stepper motor, use the following equation:

$$I_{max} = \text{supplied voltage} / \text{resistance of stepper motor coil}$$

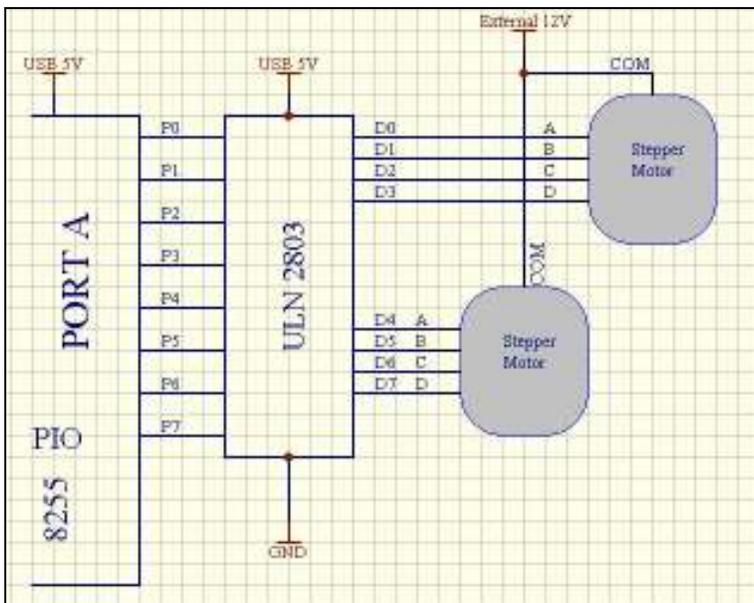
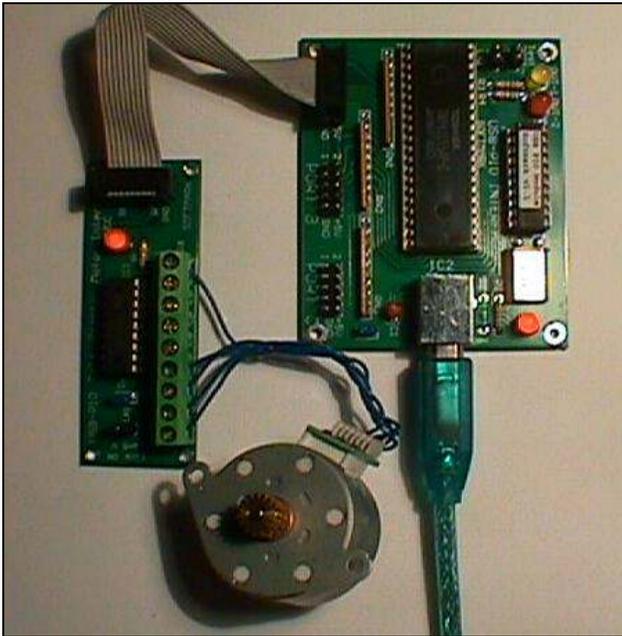


Figure 14 Two stepper motors connected to and controlled by the USB PIO Card.

Port C can be configured as a 4-bit output and a 4-bit input. This setup can be very useful if, for example, a stepper motor uses 4 outputs and 4 inputs (which can be used to control the status, switches and eventually switch the stepper motor off).

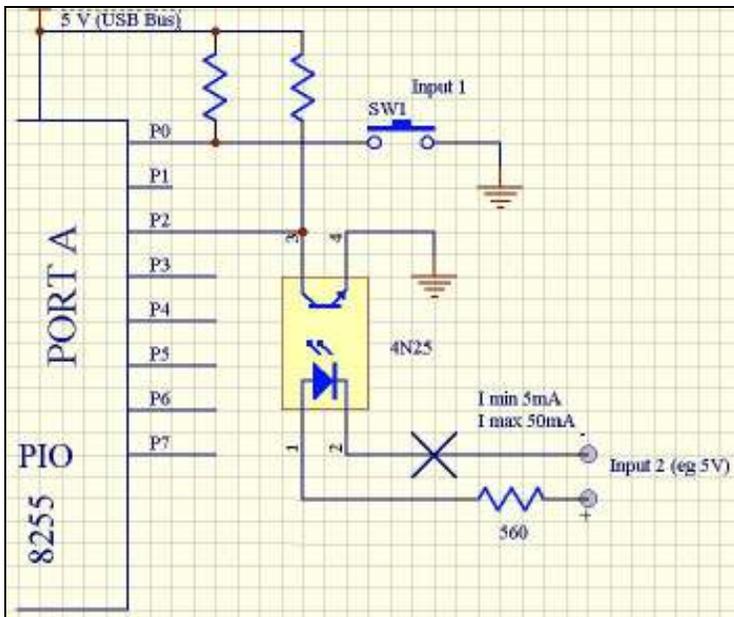


**Figure 15.** A stepper motor connected to the USB PIO Card via Relay/Stepper Motor Interface.

## How to read status from a particular port

If one of the USB PIO Card's ports is set as an input, this port will acquire data and transfer it to your application. This can be used to inform your application that this line was, for example, set to high.

Figure 16 below shows an input interface used to set lines of the port to high or low. Softmark has developed two input interfaces: the Switch Interface and the PIO Input Interface (with opto isolation). These interfaces can be used effectively during the writing of software, the testing or debugging of your software, or even to control industrial processes.



**Figure 16.** Input controlled by different interfaces.

We have a PIO Input Interface (with opto isolation) which gives you 2,500V of isolation between the USB PIO Card and your hardware.

Just pressing a switch on the Switch Interface can set a particular line. There are 8 switches. The USB PIO Card can control 24 switches at the same time. Figure 19 below shows the Switch Interface connected to the USB PIO Card via a ribbon cable.

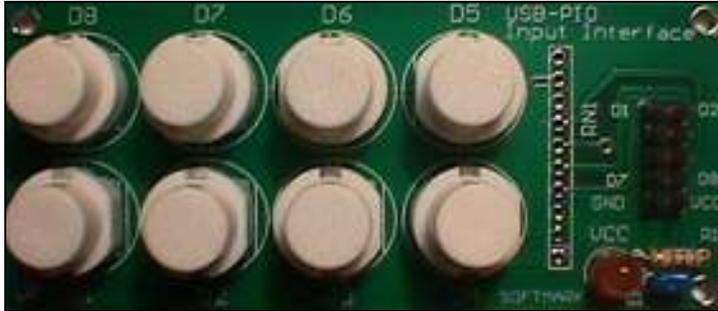


Figure 17. A Switch Interface.

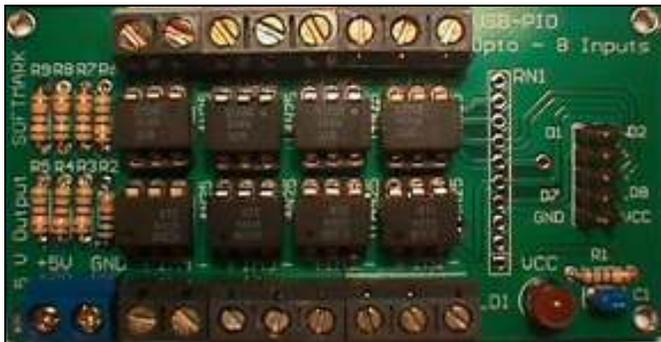


Figure 18. PIO Input Interface (with opto isolation).

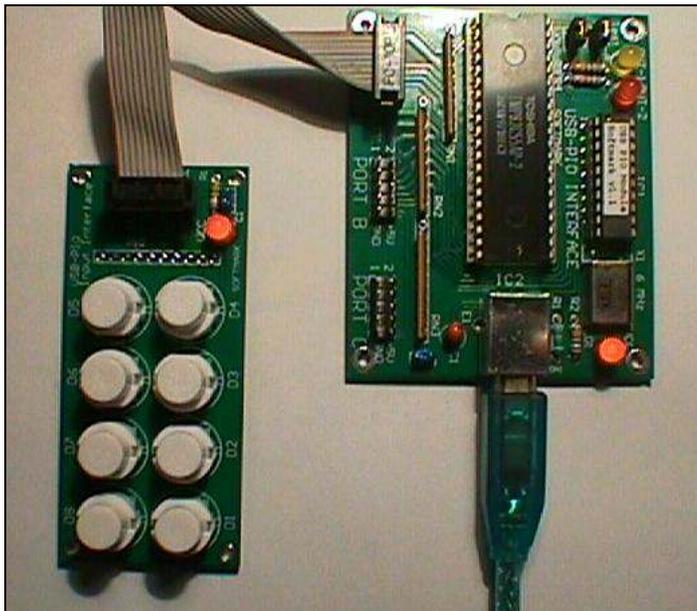
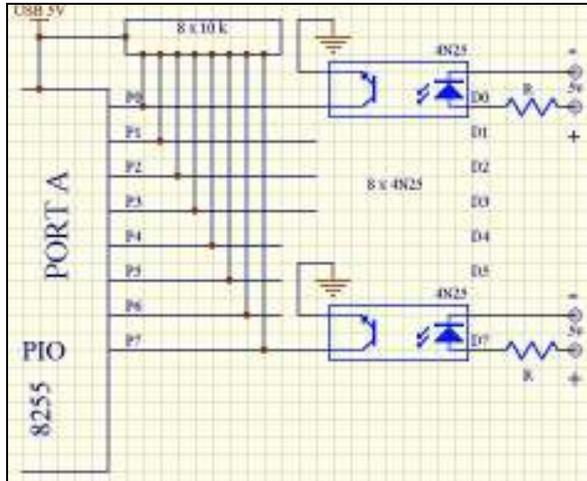


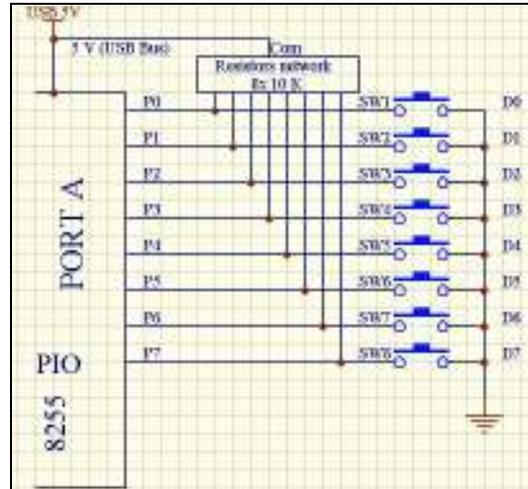
Figure 19. Pressing a button on the Switch Interface will set a line.

A diagram for the Switch Interface is shown below in Figure 21 and a diagram for the PIO Input Interface (with opto isolation) is shown in Figure 20.

To set a level (high or low) on a particular port, +5V should be used. The maximum current which can flow through the opto element is 50mA. If the voltage used is more than +5V, it will be necessary to adjust resistor **R2** to avoid exceeding the 50mA current in the opto element.



**Figure 20.** Inputs are isolated from the USB PIO Card.



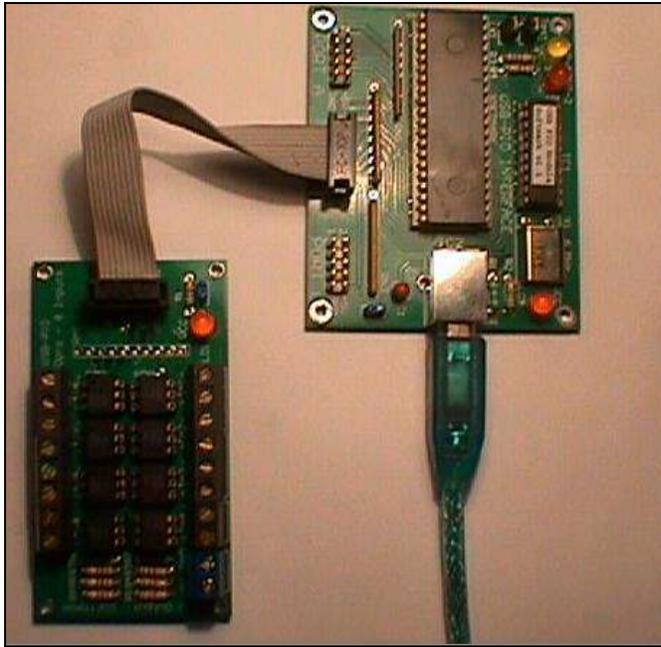
**Figure 21.** Setting inputs on port A using switches.

Figure 21 above shows the PIO Input Interface (with opto isolation) connected to the USB PIO Card. You can control this board via the USB\_OCX component. For instance, you can configure the board to read data from port A.

If you use the command:

```
a = RD_PO()
```

a port value will be returned. Don't forget to write the configuration word to the 8255 and set port A as an input. More information about the configuration of 8255 can be found on the data sheet for this particular IC.



**Figure 22.** The PIO Input Interface (with opto isolation) connected to the USB PIO Card. The Card is ready to acquire data to port B. The isolation level is 2,500 V.

## Kit details

This kit is assembled and tested. It consists of the main USB PIO Card and the four I/O interfaces described in this document. USB\_OCX is included in the full version. This component will allow you to control the module from your own application. You can also control this module from DOS level. The DOS\_USB file is free and can be downloaded from our web site [www.ar.com.au/~softmark](http://www.ar.com.au/~softmark). A comprehensive guide (with schematic diagram) including information on how to install the driver on your PC is also included. Figure 23 shows the contents of the full version of this development package.

We can also supply an assembled USB PIO Card with four bare PCBs for you to assemble. This option comes without the USB\_OCX component as you may have already purchased this component with one of our other USB kits previously.



Figure 23. Contents of the package (full version).

## Important

The USB PIO Card does not need a power supply as voltage is supplied from the USB bus in your PC.

---

Copyright © 2006 Softmark

All rights reserved. The information presented in this document is believed to be accurate and reliable and may be changed without notice. No liability will be accepted for any consequences of its use.