Sirius microSystems

172 Harvard Road
Waterloo, ON N2J 3V3
CANADA
www.siriusmicro.com

Beginner Project 2

Connecting LEDs to PICmicro I/O Pins

So, you've got a great idea for a project using a PICmicro microcontroller and you want to use at least one LED in your project as a power indicator. Or, maybe, you've set out to design a bar-graph voltmeter with eight LEDs, or a circuit using seven-segment LED digit displays, or a controller for a poster with blinking lights. Connecting LEDs to PICmicro I/O pins is a breeze as long as you know a few basic rules, and Ohm's law.

LED Characteristics

LED (Light Emitting Diode) lamps are electronic diodes that produce light. The light is produced only when current passes through the diode in the forward direction, propelled by a forward voltage charge (see figure 1). Before light is produced, however, the forward voltage across the diode must be higher than the internal barrier voltage of the diode. This point, labelled $+V_F$ (Voltage$_{\text{Forward}}$) on the graph in figure 1, is the point at which the diode begins to conduct and produce light.

It is important to notice that once the voltage across the LED reaches $+V_F$ the diode conducts current extremely well. This action is shown by the sharp rise in the forward current (+I) indicated by the near vertial line on the conduction graph. The LED attempts to clamp the voltage near $+V_F$ and can be easily destroyed by an excess of voltage. To protect the LED, a series current limiting resistor should be added, as shown in figure 2, below.

The value of the series limiting resistor must be calculated based on the maximum allowed LED current and the difference between the applied voltage and the LED's voltage drop, $+V_F$.

Like any other diode, LEDs pass current in the forward direction, but block current in the reverse direction (see figure 1). What this means is that the LED will only light up if connected with its cathode on the negative side of the circuit, and its anode on the positive side of the circuit. Too much reverse voltage will also destroy LEDs and diodes.

The cathode side of an LED is usually marked with a flat spot on the flange that rings the body of the diode. The cathode wire is also usually shorter that the anode wire of an LED. Figure 3, below, compares an LED with its schematic symbol.

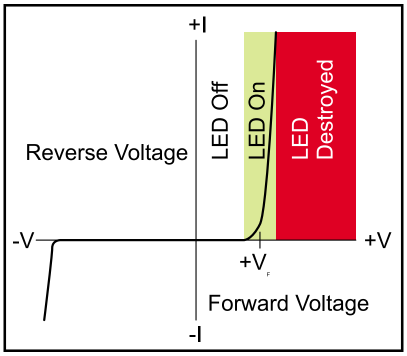


Figure 1. The LED conduction curve shows the safe operating area.

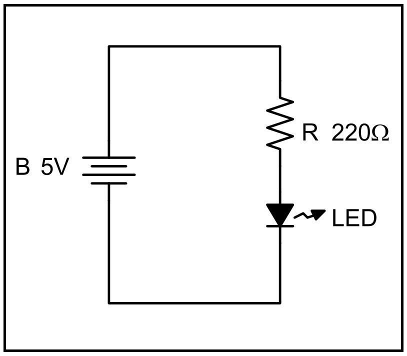


Figure 2. A typical LED circuit with a series current limiting resistor.

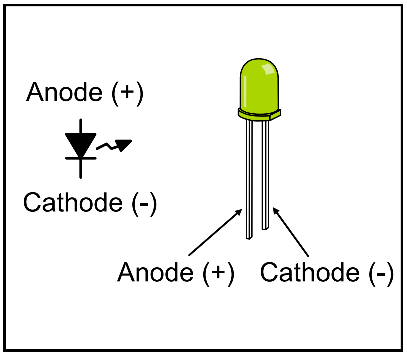


Figure 3. An LED schematic symbol compared with an LED.

Calculating the Series Current Limiting Resistor

We know from figure 1, and the discussion on the previous page, that a series current limiting resistor is required to prevent excessive current from destroying the LED. But, how do you know how much current is too much? A rule of thumb is that the maximum safe current for most LEDs is 20 mA.

Figure 2 shows a typical electrical schematic with a series current limiting resistor in circuit with the LED. This circuit is a voltage divider in which the diode voltage and the resistor voltage must add up to the total applied voltage. As long as the applied voltage exceeds the $+V_F$ of the LED the voltage across the LED remains fairly constant (see figure 1, again). A good digital multimeter with a diode test function will let you easily find the forward voltage of the LED. Once you know the voltage across the LED, you also know that the voltage across the resistor is the difference between the applied voltage and the LED forward voltage.

The calculations, below, demonstrate how to find the series resistor voltage. Once you know the resistor voltage and the maximum diode current, Ohm's Law lets you easily determine the size of the current limiting resistor. We'll assume that the LED forward voltage drop is 1.4 Volts.

$$V_R = V_B - V_{LED}$$
$$V_R = 5V - 1.4V$$
$$V_R = 3.6V$$

$$R = \frac{V_R}{I_{LED}}$$
$$R = \frac{3.6V}{0.02A}$$
$$R = 180\Omega$$

First, we determined that the resistor voltage would be 3.6 Volts. Next, Ohm's Law lets us calculate that a 180Ω resistor will limit the current to 20 mA.

If you can't easily find a 180Ω resistor, round up to the next commonly available size (figure 2 shows a 220Ω resistor). Rounding up errs on the side of caution, increasing the circuit resistance and decreasing the LED current. Although the LED may not be quite as bright as with the calculated resitance, you will not have to worry about damaging it with excess current.

Driving LEDs using a Microcontroller

TTL logic gates, like the 7400 family, are better at sinking current than sourcing current. What this means is that the output transistors of a TTL device work better when they connect a load to ground (0V) rather than to the +5V supply. The circuit in figure 4, below, shows a TTL-like drive configuration in which a PICmicro I/O pin must be cleared (logic 0, or 0V output) to turn on the LED.

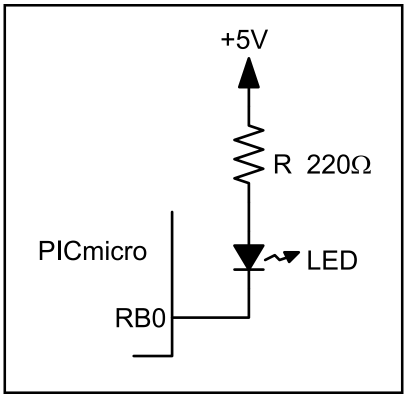


Figure 4. An active-low LED drive connection to a PICmicro.

PICmicro microcontrollers are constructed of CMOS transistors as opposed to TTL transistor. CMOS circuits work equally well when connecting an output device to the high voltage side (+5V) as they do when connecting to the low voltage side (see figure 5).

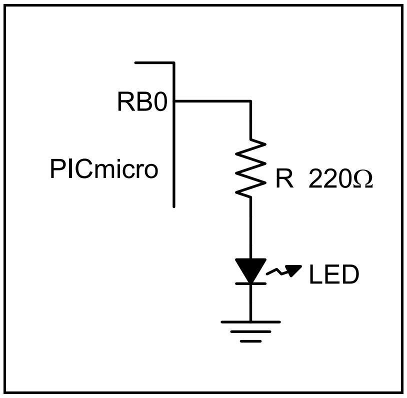


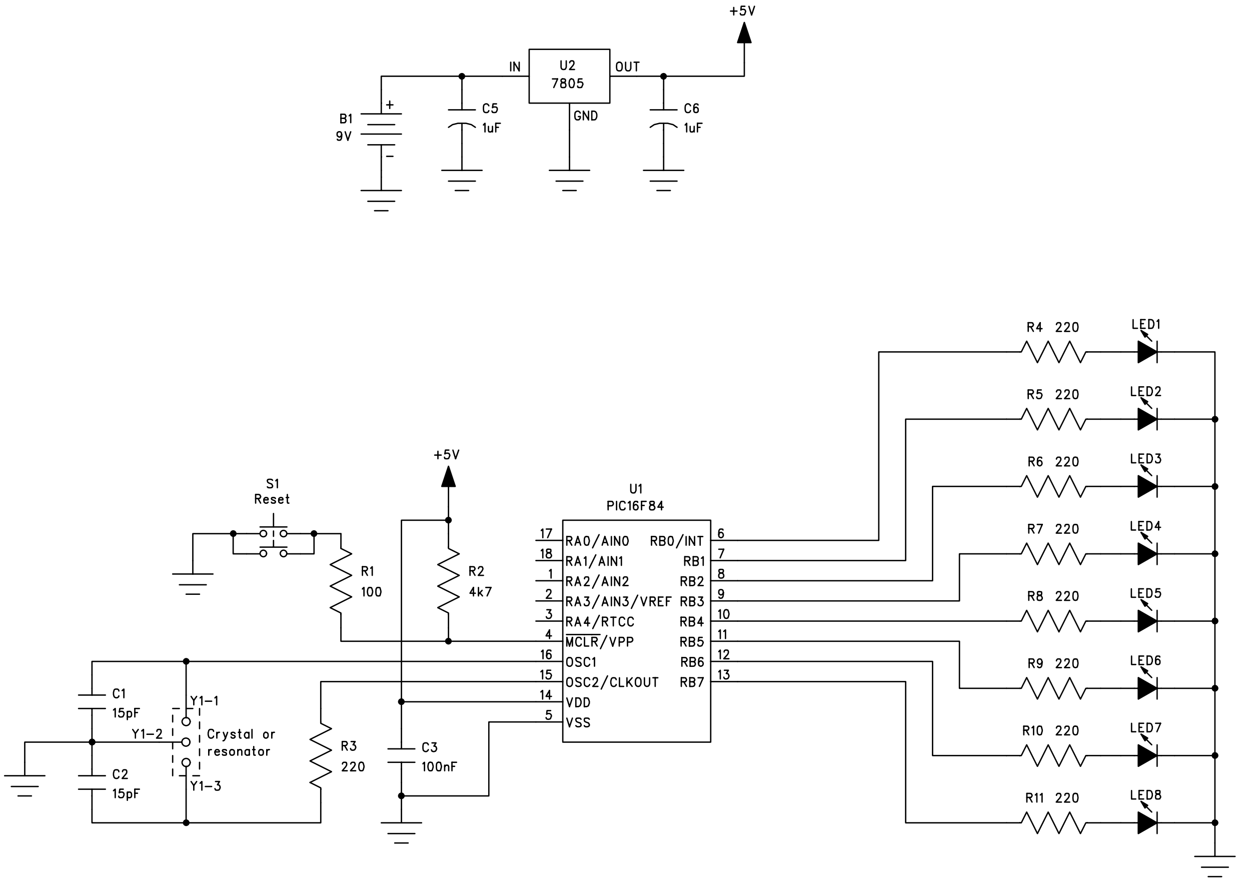
Figure 5. An active-high LED drive connection to a PICmicro.

The advantage of an active-high drive circuit like that the one in figure 5, above, is that it's easier for a programmer to follow the logic that lights the LED. Making the PICmicro output 1, or a high voltage, turns the LED on. Making the output 0, or a low voltage, turns the LED off.

The Example Circuit

The example circuit on the next page shows an active-high LED circuit, just like the one in figure 5, connected to each I/O pin on Port B of the PIC16F84 PICmicro. Resistors R1 and R2, along with switch S1, make up the reset circuit for the microcontroller. Capacitors C1, C2, crystal Y1, and resistor R3 form the clock oscillator circuit that runs the PICmicro. The power supply circuit, built around regulator U2 at the top of the schematic, produces the 5V output needed to power the PICmicro from a 9-12V input. A more detailed explanation of the function of the components in the reset, clock and power supply circuits can be found in Beginner Project 1.

Printed circuit board construction drawings and a parts list for a simple PICmicro LED output circuit can be found on page four of this project sheet.



Sirius microSystems Copyright © 1999

Title: Basic Project 2		Rev.: 0	Mod.: 0
Drawn by: JMR	Date: 99-09-17	Sheet 1 of 1	
Checked by: JMR	Date: 99-09-17	Filename: B2	

Circuit Ideas

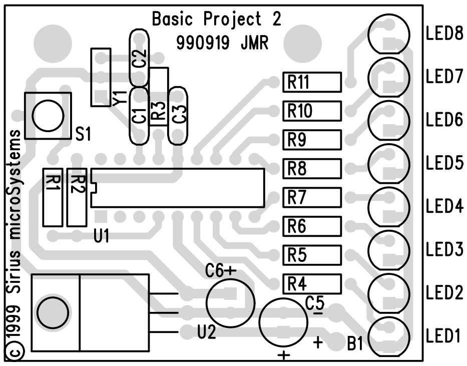
The simple PICmicro LED circuit shown in this project can be used in a variety of ways. Here are just a few possibilities:

- LED sequencer/chaser
- poster/display light controller
- analog bar graph level meter

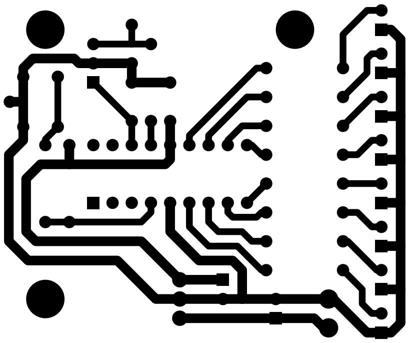
Circuit Programming

The PICmicro assembly code example shown below demonstrates the steps required to initialize Port B of the PIC16F84 for output and how to output a pattern on the LEDs connected to Port B.

This assembly code example was designed to be used in Microchip Technology's MPLAB development environment. Some changes may be needed in order for the assembly code to function with third-party tools.



Parts layout diagram, top view.



Printed circuit board wiring diagram, foil (bottom) side, top view.

B2 Parts List

C1,C2	15pF ceramic capacitor
C3	0.1µF ceramic capacitor
C5,C6	1µF, 25V electrolytic cap.
LED1-8	5mm (T1-¾) LED
R1	100Ω, ¼W resistor
R2	4.7kΩ, ¼W resistor
R3-R11	220Ω, ¼W resistor
S1	Omron B3F1100 or equiv.
U1	Microchip PIC16F84-04/P
U2	7805T 5V regulator
Y1	HC-49 4MHz crystal

Misc: Battery clip B1, or power supply wires; printed circuit board; regulator mounting hardware.

Note: Any momentary contact switch should work for S1. Y1 crystal should be "microprocessor" type.

```
;B2.ASM            V1.0    Last modified on November 5, 1999

;Sirius microSystems provides this software on an “as is” basis, without any
;warranty, either expressed or implied. All Sirius microSystems software is
;provided for educational use only, and Sirius microSystems does not assume
;any liability for damages, either incidental or consequential, arising out of
;the application, use, or misuse of any of its software or hardware products.
;Sirius microSystems reserves the right, without further notice, to make
;changes to any of its software or hardware referred to in this program or
;library program in order to improve its function, design or reliability.

;Description

;This program demonstrates port output. It initializes all port B pins
;to be outputs and then places a pattern on port B to illuminate LEDs.

            list    p=16f84
            include <p16f84.inc>

            org     00h             ;Reset Vector location

Initialize  bsf     STATUS,RP0      ;Select memory register page 1
            clrf    TRISB           ;Make Port B output by clearing TRISB
            bcf     STATUS,RP0      ;Go back to register page 0

Main        movlw   01010101b       ;Move 01010101b into W and
            movwf   PORTB           ;output pattern to LEDs on Port B

Done        sleep                   ;Stop executing the program

            END
```

MPLAB uses the list and include directives to incorporate support for specific microcontrollers.

TRISB is the port control register for Port B.

A sleep instruction puts a PICmicro to sleep until it is reset