Figure 8.10 shows a positive logic switch interfaced to PD0 and a positive logic LED interfaced to PD3. Program 8.3 initializes PD0 as an input and PD3 as an output. The main loop performs the desired function: if the switch is not pressed the LED is on.

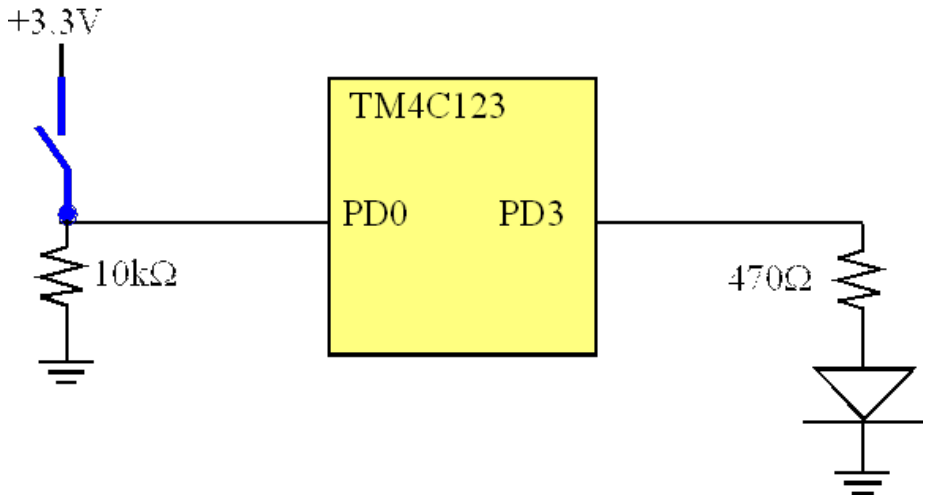

Figure 8.10. The microcontroller circuit with the switch and LED interfaced to the microcontroller.

Notice that Program 8.3 does not provide for abstraction like the previous programs 8.1 and 8.2 did. Program 8.3 although simpler and easier to understand will be harder to modify.

```
unsigned long in,out;
int main(void){ unsigned long volatile delay;
  SYSCTL_RCGC2_R |= 0x08;           // Port D clock
  delay = SYSCTL_RCGC2_R;           // wait 3-5 bus cycles
  GPIO_PORTD_DIR_R |= 0x08;         // PD3 output
  GPIO_PORTD_DIR_R &= ~0x01;        // PD0 input
  GPIO_PORTD_AFSEL_R &= ~0x09;      // not alternative
  GPIO_PORTD_AMSEL_R &= ~0x09;      // no analog
  GPIO_PORTD_PCTL_R &= ~0x0000F00F; // bits for PD3, PD0
  GPIO_PORTD_DEN_R |= 0x09;         // enable PD3, PD0
  while(1){
    in = (GPIO_PORTD_DATA_R&0x01); // in 0 if not pressed, 1 if pressed
    out = (in^0x01)<<3;   // out 8 if not pressed, 0 if switch pressed
```

```
    GPIO_PORTD_DATA_R = out;
  }
}
```

Program 8.3. Software system that turns on the LED if the switch is not pressed (C8_SwitchLED).

VIDEO 8.6. SOFTWARE SOLUTION IN SIMULATION

PROFESSOR JONATHAN VALVANO: An effective design process

is to first build the solution in the simulator.

Each of the projects in this class will have a starter file,

and this is the starter file for this module.

We'll begin by making sure we're in the simulator.

So we hit Options, Debug, and we see that

Help

0:00 / 6:51 1.0x

This solution used an **unfriendly** approach to accessing Port D. Think about how it could be changed to make it friendly.