

## 9.6 Procedure

- (1) Use an ASCII editor (e.g., Windows Notepad or MS Word - Text Only) to create the program "blink.bas" listed in Section 9.3. Save the file in a folder in your network file space.
- (2) Follow the procedure in Section 9.4 to compile the "blink.bas" program into hexadecimal machine code ("blink.hex") and to load this code onto a PIC.
- (3) Assemble and test the circuit shown in Figure 9.1. When power is applied, the LED should immediately begin to blink on and off, cycling once each second.
- (4) Repeat steps (1) through (2) for the "onint.bas" program listed in Section 9.5. **Be sure to create a new project and follow the entire procedure.** Before constructing the circuit for onint.bas, identify the additional components required in Figure 9.2. Indicate the necessary changes in the figure and check with your Teaching Assistant to verify that your changes are appropriate. The program should turn on an LED attached to PORTB.7. An interrupt on PORTB.0 should cause the LED to turn off for half a second, and then turn back on again. This should be signaled by a single pole, single throw (SPST) switch or a normally open (NO) button. You must make sure to wire the switch or button such that the ON state applies 5 Vdc to the pin, and the OFF state grounds the pin. The input should not be allowed to "float" in the OFF state (i.e., it must be grounded through a 1k $\Omega$  resistor). When you are sure you have all components wired properly, apply power to the circuit and test it to determine if it is working properly.

**NOTE - When removing ICs from a breadboard, always use a "chip puller" tool to lift both ends together. Alternatively, use a small flat-head screwdriver to pry each end up a little at a time to release the IC without causing damage (e.g., bent or broken pins).**

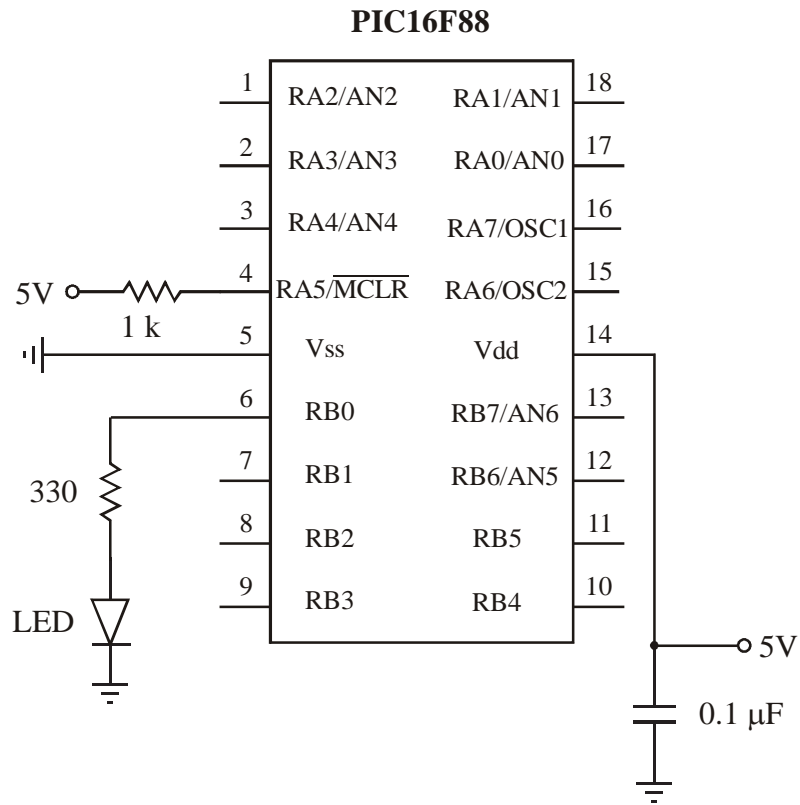


Figure 9.2 Circuit to be modified to run onint.bas

**LAB 9 QUESTIONS**

Group: \_\_\_\_\_ Names: \_\_\_\_\_  
\_\_\_\_\_

- (1) Explain all differences between PORTA and PORTB if using the pins for inputs. Refer to Section 7.8 in the textbook for more information.
  
  
  
  
  
  
  
  
  
  
  
- (2) For the onint.bas interrupt example, if the button is held down for more than 0.5 second and then released, is it possible that the LED would blink off again? If so, explain why. (Hint: consider switch bounce.)
  
  
  
  
  
  
  
  
  
  
  
- (3) Show two different ways to simply and properly interface an LED to a PIC output pin. One circuit should light the LED only when the pin is high (this is called positive logic) and the other circuit should light the LED only when the pin is low (this is called negative logic).

- (4) Explain what you would observe if power were applied to a PIC loaded with the following code if an LED is connected to RB0 as shown in Figure 9.1. Note - *Goto* is being used, not *Gosub*.

```
before: High PORTB.0
        Pause 500
        Low PORTB.0
        Goto during
        Pause 100
        High PORTB.0
        Goto after
during: Low PORTB.0
        Pause 300
        High PORTB.0
        Pause 400
after:  Pause 200
        Low PORTB.0
        End
```