

# ONE BUTTON, ONE POT, AND ONE BLINKY LIGHT

## 2.1 Objectives

In this lab, you will construct another simple device from a PIC18F2455 microcontroller. In the process, you should become familiar with the process of interfacing a potentiometer to the PIC through one of its built-in analog-to-digital converter (ADC) channels. You will also learn about the concept of the *duty cycle* of a *pulse-width-modulated* (PWM) signal. You will also learn about interfacing a hobby servomotor to a PIC.

## 2.2 Building the Base Device

Obtain the `lab2.zip` source code archive from the course website and unzip it into a new folder. This archive contains two files, `lab2.c` and `18F2455.1kr`. The `lab2.c` file is the source code for the base device for this lab. The `18F2455.1kr` is identical to the one distributed in `lab1.zip`. The base firmware configures PORTA so that pin RA0/AN0 is an analog input pin, pin RA1 is a digital output pin, and pin RA2 is a digital input pin. The analog voltage on pin RA0/AN0 is used to control the *duty cycle* of the 1 Hz square wave that the PIC produces on pin RA1, which drives an LED so that you can observe the signal. The duty cycle of a square wave, typically expressed as a percent, is the fraction of the period over which the signal is high. For example, a square wave that is high for one quarter of its period and low for the remaining three quarters would have a 25% duty cycle. Such a signal in which a time series of information is encoded in the widths of a succession of pulses is called a *pulse-width-modulated* (PWM) signal. The base firmware is written so that the duty cycle is only updated when the button connected to pin RA2 is pushed.

Figure 2.1 shows the circuitry required to implement the firmware contained in the `lab2.c` source file. Note that this circuitry is nearly identical to that which you used in lab 1. The only difference is that the 470- $\Omega$  resistor in series with an LED connected to pin RA0 has been replaced by a 5-k $\Omega$  trim potentiometer. A potentiometer, or pot, is typically used either as a variable resistor whose value can be adjusted between 0 and the value specified on the pot or as a variable voltage divider to create an analog voltage whose value can be adjusted continuously from one power supply voltage to the other. In the former mode, one end is typically shorted to the center wiper, creating a two-terminal resistor. In the latter mode, the ends are connected to the power supplies and the center wiper is the variable voltage output. Program your PIC with the base lab 2 firmware. With a bit of luck, you should see the LED blinking once per second with a duty cycle controlled by the potentiometer setting while the button is pushed.

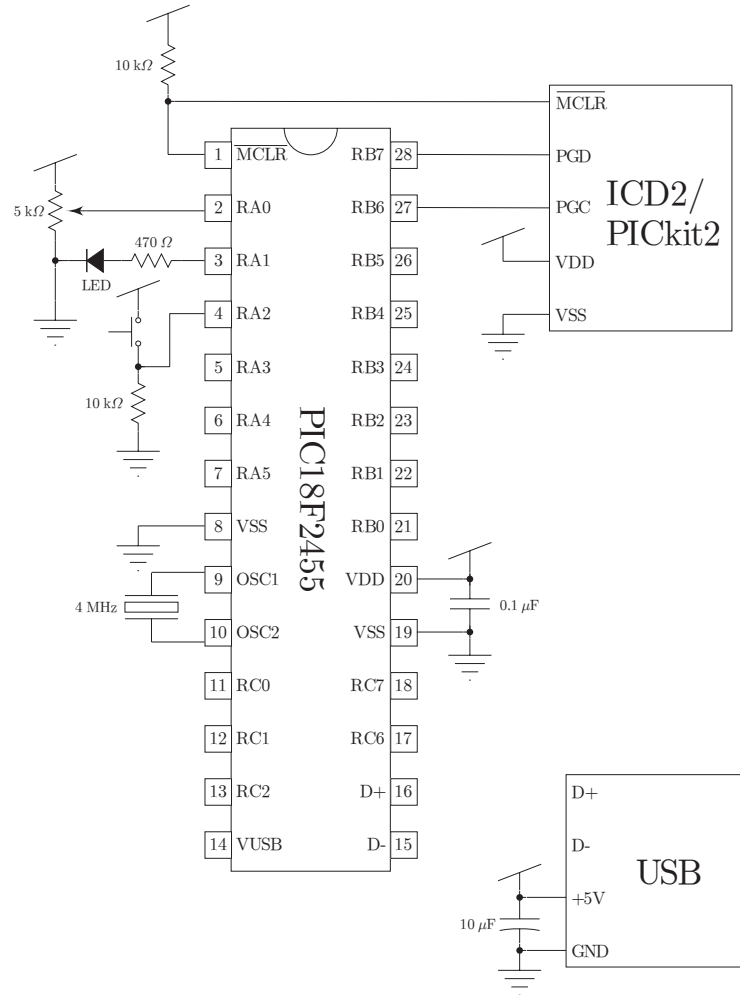


Figure 2.1: PIC18F2455 lab 2 circuitry.

## 2.3 Modifying the Device

Now, you will need to modify the firmware so that the potentiometer value changes the angular position of a hobby servomotor instead of the duty cycle of a blinking LED. The position of a hobby servo is controlled by the duration of a pulse that is repeatedly sent to the servo. Typically, the pulse duration can vary between 0.9 ms and 2.1 ms with the neutral position at 1.5 ms. If the repetition rate of these pulses is fixed, this pulse-width encoding corresponds to the notion of duty cycle. However, most hobby servos do not require a constant pulse repetition rate. Typically, the servo position is updated between 50 times per second (i.e., once every 20 ms) and 100 times per second (i.e., once every 10 ms). Modify the base firmware so that the position of the potentiometer updates the PWM signal sent to a servo when the button is pushed. When the button is not being pushed, the servo should maintain its angular position just as the LED maintained the duty cycle of its blinking.