Lab 3: Pulse width modulation

Goals:

The main goal of this lab is to become familiar with pulse width modulation (PWM), use the on-chip timer to generate a PWM outputs, and to control a simple PWM device.

Pulse width modulation

This is one of the most common ways to digitally control the intensity of certain types of devices (e.g. lights, sound volume, motor speed.)

A servo motor controller

In this lab, you will extend a servo controller example from class to control a servo motor using a pair of push buttons. One button will cause the servo motor angle to be increased by a fixed amount each time the button is pressed, the other will cause it to be decreased. The total number of steps to go from one extreme angle to the other should be about 50. Use timer/counter1, as in the example.

You may use either polling or PCI interrupts to sample the button inputs, but there should only be one step for each distinct button press.

The keys can be input through `PORTC`, and the output will be from `OC1A` (pin PD5).

The servo motor has three wires - a red wire (power) a black wire (ground) and a white wire (control). The power and ground can be obtained from the STK-500 board.

Optional:

Describe how a second servo could be implemented efficiently.

Suppose five servos were to be controlled. Briefly describe one way this could be done.

A LED brightness control

This is a *design* problem, where you are given a required output but are free to implement it as you like. In your lab writeup, explain why you made the various decisions necessary for the design.

The LEDs on the STK-500 have an output circuit which provides nearly constant current through the device over a range of input voltages.

In order to control their brightness, we can use a PCM, switching on the power for a variable amount of time, but at a high frequency (at least 100 times per second).

Using timer/counter0 in PWM mode (you choose the particular mode, frequency, etc.) control one of the LEDs using a pair of buttons (one for up, one for down, as in the previous example) over a range of at least 32 different brightness levels.

Explain why you used any particular parameters in your design (refresh frequency, PWM step size, etc.) and the particular mode of operation for your counter. If it helps your explanation, sketch the waveforms you desire for the output.

Could you implement an independent controller for a second LED using the same timer? Briefly explain how, if possible.

Optional:

Implement this second controller, if possible, and demonstrate the independent operation to the instructor. Describe the changes required to accomplish this.