Microprocessor-Based Systems (E155)

Harris **Fall 2004**

Lab 7: PWM Controller Due: Week of Nov 1

Requirement

Write a program using interrupts to pulse-width-modulate (PWM) Port D bit 1. The duty cycle will vary from 0 to 100 percent based on the analog voltage read from Port A bit 1. Use a 5-volt source for a potentiometer and read the analog voltage at PA1. Vary the duty cycle from 0 to 100% for an input voltage of 0 to 5 V. You are free to choose the period of your PWM output, but it should not exceed 10 ms. Measure the input voltage and the average output voltage level with a DC voltmeter. The two voltages should be equal to within 50 mV over an input voltage range of 0.1-4.9 V.

Discussion

Pulse width modulation is often used to drive motors. In this lab, you will use the output compare feature of one of the internal timers on the PIC to generate a pulse-width-modulated output.

Read Chapters 8, 14, and 17 of the PIC data sheet for information on the output compare and A/D converter. Be sure to use interrupts in your design; do not use the PIC's built-in PWM feature.

Hints

The PIC has two types of interrupts: high priority and low priority. It does not matter which one you use, but your code will depend on your choice.

When an interrupt occurs, the program counter is pushed onto the stack and loaded with an interrupt vector. On the PIC, this vector is 0x0008 for high priority interrupts and 0x0018 for low priority interrupts. You cannot change this vector, so you need to put a branch at 0x00x8 instead. For example:

```
org   0x0008
bra   ccpisr      ; branch to the CCP interrupt service routine
```

You also need to make sure that your main code routines aren't sitting in the interrupt vector, so use the ORG directive to start your main code at 0x0020 or a similar address. The program will default to 0x0000 when you reset the PIC, so put a branch statement at this address to jump to your main program.

The fancy digital multimeters have an autoranging feature that sometimes measures different values than you may have intended. For example, when the duty cycle is low, the voltmeter may measure the value of the signal when it is low instead of measuring the average value. Pressing the ^ key on the front panel adjusts the range of the meter, putting it in manual mode. This may help obtain an average reading.

What to Turn In

Demonstrate your PWM system. Be prepared to answer fault-tolerance questions about your software and output. Understand why you are getting any discrepancy between input voltage and average output voltage. Your lab notebook should include:

- Your design approach
- A listing of your assembly language code generated in your .lst file
- Breadboard schematics
- Data about the output vs. input voltage at the top, bottom, and middle of the range
- How many hours did you spend on the lab? This will not count toward your grade.