

Microprocessor-Based Systems (E155)

Harris

Lab 7: PWM Controller

Requirement

Write a C program for the PIC using interrupts to pulse-width-modulate (PWM) a bit on an output port. The duty cycle will vary from 0 to 100 percent based on the analog voltage read from Port A bit 1. Use a 3.3-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 3.3 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 0.1-3.3 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.

Programs can use either interrupts or polling to check for an event. Polling is simpler, but it wastes the processors time continually checking to see if an event has occurred. Interrupts let the program go about its usual business without explicitly checking for the event; then the interrupt service routine is notified when the event occurs. For example, a robot might be receiving data from a sensor and performing a FFT to analyze the data at the same time it is driving its motors at a desired speed. By using interrupts, the robot can go about its signal processing without having to frequently check if it is time to toggle the motor control output. In this lab, use interrupts so that you learn how they work. Be sure that you do not poll. Your main function should have an endless loop that does nothing (the signal processing might happen here in a real application), and the loops should be interrupted to generate the PWM signal.

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 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 writeup should include:

- Your design approach
- A listing of your C program.
- 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.