Embedded Systems Design

EECE 4038C, Embedded System Design
Ranga Vemuri

Laboratory Assignment – 8

Preparation:

Go through Lessons 9 and 10 in Chapter 3 of the "44-Pin Demo Board User's Guide" and execute the corresponding programs on the demo board with PIC 16F887.

Assignment:

Using the knowledge you have gained from these lessons, develop an assembly program for the following embedded system:

Bear Mobile

A microcontroller installed in a car performs two tasks: (1) Controlling the wiper assembly and (2) Controlling a decorative teddy bear on the dashboard of the car.

Your program should incorporate the use of TMR0 overflow interrupt to switch from Task 1 to Task 2. Upon servicing the timeout interrupt (which will progress Task 2) the processor should return to Task 1.

Task 1: The wiper assembly has 128 speed settings one of which is selected by the driver by rotating a "smooth" rotary control. The speed settings range from 0 to 127. The rotary control is internally discretized into 128 evenly spaced control zones, one for each setting. Assume that the rotary control is modeled by the potentiometer. Assume that the wiper assembly is driven by RD1 through RD7 to indicate one of the 128 speed settings.

Task 2: Controlled by the embedded microcontroller, a toy teddy bear blinks continuously, keeping its eyes open for one second, closed for the next second, open for the following second and so on. Its eyes are controlled by an actuator as follows: when the input to the actuator is high, it opens the eyes and when the input to the actuator is low, it closes the eyes. Assume that the actuator is driven by the RD0 port of the PIC16F887 from your demo board.

Exercise:

If you were not required to use TMR0 overflow interrupt, what would have been your method to simultaneously perform both the tasks on the PIC16F887 in the demo board setup. Describe the alternative method using a flow chart and discuss its merits over using the TMR0 overflow interrupt method

In addition note that,

1. Your report must include a flow chart for your solution.
2. Your code must be well documented and must correspond to your flow chart.
3. You must use macros and subroutines wherever appropriate to improve modularity and maintainability of the code.
4. You must use a good template design for your program, following the coding practices you have noticed in your reading assignments.
5. You must discuss the algorithmic and programming choices you have made while developing this solution.