ECE-218 EMBEDDED MICROCONTROLLER PROJECTS:
PROJECT 1: TEMPERATURE ALARM

Video and Code – due Sunday 1/20 midnight, Report – due Monday 1/21 midnight

PROJECT OBJECTIVE

The goal of this project is to develop an embedded microcontroller system with the Microchip Microstick II board to monitor and display the ambient temperature, and to sound an alarm if that temperature reaches 80 degrees Fahrenheit. You will use an analog temperature sensor, the ADC on the PIC24, an active buzzer, and a serial connection to a monitor for display. Some specifications follow:

- After system reset, the terminal will display **DISARMED**, followed by the temperature (with 3 significant figures) along with the units (°F). Sample: `DISARMED  72.1 degrees F`
- The system will be "armed" when a pushbutton is pressed and released, and **ARMED** will be displayed on the display, again followed by the current temperature.
- If the temperature reaches 80 degrees or above, the terminal will display **ALARM! Press Button to Stop!** and the buzzer will sound until a pushbutton is pressed again.
- Once the buzzer is turned off by the pushbutton being pressed and then released, the system is disarmed and DISARMED, followed by the temperature is again displayed until the system is armed again.

PROJECT HINTS

Since this problem has time ordered behavior, it will be helpful to model it as a state machine.

It is good practice in developing embedded systems to break the problem into smaller problems and get these working before combining them together. Here is a suggested list of sub-problems:

- Display temperature correctly on monitor.
    - Read analog value, convert to millivolts (floating point), then convert to temperature in Celsius, and then to Fahrenheit.
    - Convert floating point temperature to an array of ASCII characters
    - Print array to monitor.
- Implement the basic state machine that sequences to the next state based on the button and the analog value read from the temperature sensor. This can be done independently of getting the actual temperature displayed.

To convert the analog voltage value that you read from the temperature sensor to a temperature, you can use the linear approximation given in equation (3) of the temperature sensor datasheet (page 10). This approximation is accurate for a small range of temperatures, so use V1,T1 and V2,T2 from Table 1 that corresponds to the highest and lowest temperatures that you would expect

to be displayed. Note that the temperatures in the table are in degrees Celsius. Use print statements to print the voltage value as well as the temperature value when debugging.

You may find it helpful to define some functions to keep the main part of your program simple.

When using natural numbers in math operations in C, you will need to use "float" (floating point) variables to hold the result. Review the slides on Nexus on C data types and operations.

A major challenge in this assignment is to convert the floating point number to an ASCII string. You can use an algorithm similar to the one provided in Exercise 2. Note that the temperature value will be something like 71.5 rather than the voltage value of 1.56, so you will need to modify the algorithm for the different number format.

PROJECT DOCUMENTATION

You will document your project in a video demonstration, and a project report. Both will be uploaded to Nexus.

DEMONSTRATION VIDEO

The purpose of the video is evaluate both the functionality and your understanding of your final system. Assume that you are preparing the demo for someone who has no idea about this project. Both partners should participate equally, and you should practice it a few times before filming. The video should be no longer than 4 minutes and there should be three parts:

1. Intro: A segment that shows you (both) and your system, where you explain the project goals in enough detail that the demo will be clear.
2. Hardware: A segment that zooms in on the system, where you point out the system components and explain the purpose of each one in the system function.
3. Demo: A segment where you demonstrate that the system is working correctly. Here you should remember all of the goals of the project and point out as each is accomplished.

REPORT

The project report is where you document the details of how you solved the problem. Follow the design report format guidelines posted on Nexus.