Assignment #3 – Temperature Monitor

In this assignment, you will use the InterruptIn, AnalogIn, and BusOut interfaces on the mbed microcontroller board to implement a simple temperature monitor.

The MCP9700A is the temperature sensor; its pinout is shown to the right. Connect the MCP9700A's +Vs pin to the mbed's VOUT pin, its Vout pin to the mbed's p20 pin, and its GND pin to the mbed's GND pin. Do not confuse the MCP9700A's Vout pin with the mbed's VOUT pin. The voltage on the MCP9700A's Vout pin is related to the temperature by the following equation (*V* in volts and *T* in degrees Celsius):

$$V = 0.500 + 0.010T$$
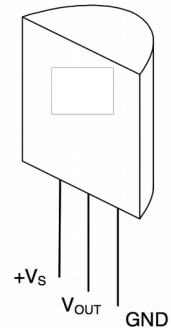
Connect a pushbutton switch between the mbed's VOUT and p23. Connect a second pushbutton switch between the mbed's GND and p21. Connect the 7-segment LED display to the mbed as shown in Figure 3.10 (1st ed.) or Figure 3.12 (2nd ed.) in the textbook, but insert 220 ohm resistors in series with each LED segment to reduce the current drawn.

Write a program for the mbed that will periodically read and display the temperature from the MCP9700A, either in Celsius or Fahrenheit depending on the selected mode. Your program should use the mean average voltage from the temperature sensor (computed over at least 1000 samples) so that the displayed value does not fluctuate due to noise when the temperature is stable.

The temperatures should be displayed to the nearest degree of whichever unit was last selected. Since the display can only display one digit, you must display the temperature one digit at a time. Display each digit for 0.5 – 1 seconds, then blank the display for 0.1 seconds (this way the user can distinguish a digit shown once from two digits with the same value shown consecutively). After displaying the temperature, the program should display either a "C" if the temperature was in Celsius or an "F" if the temperature was in Fahrenheit. Note that the temperature might be less than 0, in which case you will need to display the leading negative sign (if you want to test this without a freezer, disconnect p20 from the MCP9700A and connect p20 instead to the mbed's GND; this simulates a temperature of around −50 C).

The two switches select the temperature mode. Pressing the switch connected to p23 should select Celsius mode. Pressing the switch connected to p21 selects Fahrenheit mode. Use InterruptIn to trigger interrupts when these switches are pressed so that the mode can be changed at any time. Although the mode can change at any time, be sure to completely display the temperature in consistent units (for example, don't display the first digit from the Celsius temperature followed by the second digit from the Fahrenheit temperature). When your program starts, it can default to whichever mode you prefer.

Hints: For debugging purposes, it might be useful to use the Serial object to display intermediate results on your computer, although this is not a requirement for the assignment. Sections 6.2 and 6.4 of the textbook may also be useful. Also, since the voltages being measured from the temperature sensor are very small, even slight resistive voltage drops can skew the results; you should minimize this effect by sampling the temperature only when the LED display is blank.

Submit your "main.cpp" to the appropriate dropbox on Canvas by the end of March 7th.