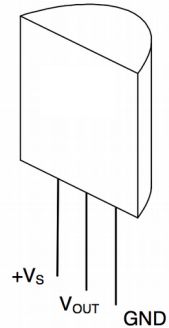


## Assignment #2 – Digital Thermometer

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

The MCP9701 is the temperature sensor; its pinout is shown to the right. Connect the MCP9701's +Vs pin to the mbed's VOUT pin, its Vout pin to the mbed's p19 pin, and its GND pin to the mbed's GND pin. Do not confuse the MCP9701's Vout pin with the mbed's VOUT pin. The voltage on the MCP9701's Vout pin is related to the temperature by the following equation ( $V$  in volts and  $T$  in degrees Celsius):

$$V = 0.400 + 0.0195T$$



Attach a wire to the mbed's p21 pin that can connect to either the mbed's VOUT pin or the mbed's GND pin (this will select the temperature units). Connect the 7-segment LED display to the mbed as shown in Figure 3.10 (1<sup>st</sup> ed.) or Figure 3.12 (2<sup>nd</sup> 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 MCP9701, 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 is 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 p19 from the MCP9701 and connect p19 instead to the mbed's GND; this simulates a temperature of around -20.5 C or -4.9 F).

The pin p21 select the temperature mode. When p21 is connected to GND, Celsius mode is selected. When p21 is connected to VOUT, Fahrenheit mode is selected. You may either read the state of the pin when the program starts and require the mbed to be reset if the mode is changed, or dynamically update the mode each time the temperature is displayed, whichever is more convenient to you. You may also choose the default temperature mode used if p21 is left disconnected using the internal PullUp or PullDown setting.

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 assignment dropbox on Canvas by the end of March 10th.