Lab 5 – Application Exam 2

Background Story

After years of faithfully teaching students at Baylor, I (the infamous Professor Potter) have decided that I've had enough of you people and that I'm moving to California to be a beach bum. I'm going to spend my days surfing, making bonfires on the beach, and bagging rays to get a savage tan. I have a problem though…my Baylor salary did not leave me with much money, so the only place I can afford to rent is a basement apartment with no windows. I have no idea what the weather is like and what activity I should do that day. That's where you come in. I need a weather station outside of the house to tell me what I should do that day, depending on the sunlight and the temperature. I can't bag rays if there are no rays….and I don't want to make a bonfire and roast mallows when it's already toasty outside. And I need to know what activity I'm doing before I leave because the little old lady who owns the house doesn't like me going in and out of the house all the time and letting the flies in. So the weather station should remotely report the day's activity to a base station in the house so I can choose the most rad activity for the day. Rock on, dude!

Ground Rules

You may only utilize the following resources while doing this problem. You may not use the internet for any other purposes or communicate with anyone else.

1. Your previous labs and files stored on the RPi
2. The Python Tutorial used in class
3. The MPC3008 datasheet
4. The TMP35/36/37 datasheet
5. The RPi lab manual on Canvas

Hardware

To complete this extremely important job, you will need the following circuits.

1. Photoresistor/ADC circuit from Lab 2.
2. MOSFET LED circuit from Lab 2.
3. Potentiometer circuit. There are 3 pins on the potentiometer, and you will use either outer pin and the inner pin. The circuit should be: 3.3V -> 47k resistor -> Potentiometer outer pin. Then connect the Potentiometer inner pin to ground. You can connect the node between the 47k resistor and Potentiometer outer pin to Channel 3 on the ADC.
4. TMP36 temperature sensor circuit. This circuit is very simple. Pin 1 should connect to the 3.3V power rail. Pin 3 should connect to GND. Pin 2 should connect to Channel 2 on the ADC. When looking at the flat side of the sensor, pin 1 is on the left and pin 3 is on the right.

Part 1 – Design (10 points)

We will start with design. The design does not include a diagram. It is simply figuring out the calculations to get a temperature from this temperature sensor. Please answer the following questions in a document of your choice and submit the document with this exam.

1. Write an equation to convert the voltage output of the TMP36 sensor to a temperature in Celsius. Pin 2 of the TMP36 will produce a voltage that corresponds to the temperature. To calculate the temperature, you will need to analyze the TMP36 datasheet linked above.
    a. IMPORTANT: The datasheet is for TMP35, TMP36, and TMP37. They all work differently, so make sure to focus on the calculations for TMP36. You do not need to look beyond the first page on this datasheet.
    b. As stated in the datasheet, the relationship between voltage and temperature is linear. To help you out, here are the two equations of a line:
        i. Slope-intercept Form: y = mx + b
        ii. Point Slope Form: (y - y1) = m(x - x1)
2. What Celsius temperatures would be represented by the following voltage values?
    a. 0 V
    b. 0.5 V
    c. 1 V
    d. 1.5V

**** IMPORTANT NOTE: If you cannot figure out the equation to get the temperature, I don't want you to be stuck at this part and not progress. You can ask me for the equation and I will provide it to you, but there will be a 10 point deduction on your grade. If you want the equation, come to my instructor station with something that I can write on.

Part 2 – Testing the Temperature Sensor

1. With the TMP36 connected as described in the 'Hardware' section, use your multimeter to measure the voltage at Pin 2. Use your equation from your design to calculate the temperature.
    a. The room will likely be in the 21-25 C range and 68-75 F range
    b. TempF = (9/5)*TempC + 32

Part 3 – Adjusting the Temperature (25 points)

One challenge with a temperature sensor is that it is difficult to substantially change the temperature without bringing in a hair dryer or something cold. To allow us the change the temperature without this, we will utilize the potentiometer circuit.

1. Build and test the potentiometer circuit as described in the 'Hardware' section. The range of voltages at the node with the 47k resistor and outer pin of the potentiometer should be between approximately 0V and 0.7V.
2. Connect this node to Channel 3 of the ADC.
3. Channel 2 of the ADC is connected to the TMP36 and Channel 3 is connected to the potentiometer circuit node. We are going to use Differential mode on the MPC3008 ADC module to determine the difference between Channel 2 and Channel 3 (Channel 2 Voltage – Channel 3 Voltage). Here's the idea:
    a. When the potentiometer node voltage is 0V, the Differential reading is just the output voltage of the temperature sensor
    b. When you turn the potentiometer so that the potentiometer voltage increases, the potentiometer voltage will be subtracted from the TMP36 voltage, resulting in a lower temperature value.
4. Turn the potentiometer until the voltage at the potentiometer node is 0V.

5. In your code repository on the RPi, create a directory called application_exam. Create two files in this directory:
    a. weather_station.py
    b. home_base.py
6. Utilizing what you have done in previous labs and the MCP3008 datasheet, write software in weather_station.py to calculate the Differential ADC value between Channel 2 and Channel 3. Note that this should NOT be getting a single-ended voltage for Channels 2 and 3 and then subtracting them. It should utilize the Differential capability built into the ADC chip. In software, turn this ADC value into a voltage and use the voltage to calculate and print the temperature every 1 second. Since you have the potentiometer circuit tuned to 0V, this should just be producing the room temperature value.
7. Turn the potentiometer and observe the temperature dropping as you turn the knob. Turn the knob the other way and observe the temperature rising.

Part 4- Choosing Potter's Activity in weather_station.py (10 points)

Now you will use a combination of the temperature and the light level (from Lab 2) to determine what activity I should do. The photosensor circuit (connected to Channel 0 of the ADC) will determine whether it is dark (> 2V) or light (< 2V). Use the following table to determine the activity (corresponding numerical value in parentheses):

Temperature Range	Activity if dark	Activity if light
< 35 F	STAY_HOME (0)	STAY_HOME (0)
35 – 50F	BONFIRE (1)	BONFIRE (1)
50 – 65 F	TOO_DARK_TO_SURF (3)	SURFS_UP (2)
65+	TOO_DARK_TO_BAG_RAYS (5)	BAG_RAYS (4)

Print the following messages for each activity every 1 second:

Activity Numerical Value	Activity Name	Activity Message
0	STAY_HOME	Stay home, dude!
1	BONFIRE	Bonfire, dude!
2	SURFS_UP	Surf's up, dude!
3	TOO_DARK_TO_SURF	Too dark to surf, dude!
4	BAG_RAYS	Bag some rays, dude!
5	TOO_DARK_TO_BAG_RAYS	Too dark to bag rays, dude!

Part 5 – Communicating Potter's Activity to home_base.py (10 points)

Now it is time to communicate the activity to home_base.py so that I will know the plan for the day. To do this:

1. Update weather_station.py to publish the following topic:
    a. your_last_name/weather_station/weather_update
        i. Pass in your_last_name as a command line argument (using your real last name)

        1. There should only be one command line argument in this program
    b. The value published should be the 'Activity Numerical Value' from the table above (a number from 0-5)
    c. The value should ONLY be published when it changes, not on a cyclical basis.
2. Update home_base.py to subscribe to the topic being published by weather_station.py.
    a. When a message is received, the 'Activity Message' from the table above should be printed to the terminal.

Part 6 – LED Indicator for Potter's Activity (5 points)

1. Finally, use the activity received in the weather update topic to provide an LED indication of what activity I should do.
    a. If the activity is BAG_RAYS, home_base.py should turn the LED on to full brightness (I love bagging rays)
    b. If the activity is SURFS_UP home_base.py should turn the LED on to 50% brightness
    c. If the activity is BONFIRE, home_base.py should turn the LED on to 25% brightness
    d. Otherwise, home_base.py should turn the LED off.

Submitting the Lab

1. Please submit the following files:
    a. Design document
    b. weather_station.py
    c. home_base.py