COMP6043 Physical Computing Lab 2

Note: if a task asks you to demonstrate your work to your lecturer, you must demonstrate and get your work signed off. Otherwise, no marks will be awarded for your work.

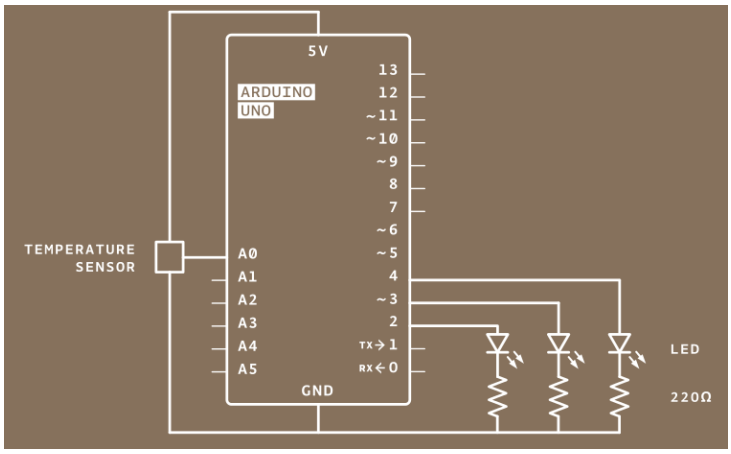
Once completed, upload your report to Canvas.

Pay particular attention when connecting the PCB to the Arduino power pins. It is very important that you connect with the correct polarity, i.e. 5 V on the Arduino to 5 V on the PCB, and GND on the Arduino to GND on the PCB. If you wire up with the incorrect polarity, you will burn out the on-board temperature sensor and damage the board.

This lab is based on the "Love-O-Meter" project in the Arduino Projects Book (p.42).

Task 1 [60%]

Wire up your circuit as follows, with GRN2 to pin 2 of the Arduino, AMB to pin 3 and RED2 to pin 4:



Open the Arduino IDE and enter the following code:

```
// named constant for the pin the sensor is connected to
const int SENSOR_PIN = A0;
// room temperature in Celcius
const float BASELINE_TEMP = 20.0;

void setup(){
  // open a serial connection to display values
```

```
  Serial.begin(9600);
  // set the LED pins as outputs
  // the for() loop saves some extra coding
  for(int pinNumber = 2; pinNumber<5; pinNumber++){
    pinMode(pinNumber,OUTPUT);
    digitalWrite(pinNumber, LOW);
  }
}

void loop(){
  // read the value on AnalogIn pin 0 and store it in a variable
  int sensorVal = analogRead(SENSOR_PIN);

  // send the 10-bit sensor value out the serial port
  Serial.print("sensor Value: ");
  Serial.print(sensorVal);

  // convert the ADC reading to voltage
  float voltage = (sensorVal/1023.0) * 5.0;

  // Send the voltage level out the Serial port
  Serial.print(", Volts: ");
  Serial.print(voltage);

  // convert the voltage to temperature in degrees C
  // the sensor changes 10 mV per degree
  // the datasheet says there's a 500 mV offset
  // ((voltage - 500mV) times 100)
  Serial.print(", degrees C: ");
  float temperature = (voltage - 0.5) * 100;
  Serial.println(temperature);

  // if the current temperature is less than 2 degrees above the baseline
  // turn off all LEDs
  if(temperature < BASELINE_TEMP + 2){
    digitalWrite(2, LOW);
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);
  } // if the temperature rises 2-4 degrees, turn an LED on
  else if(temperature < BASELINE_TEMP + 4){
    digitalWrite(2, HIGH);
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);
  } // if the temperature rises 4-6 degrees, turn a second LED on
  else if(temperature < BASELINE_TEMP + 6){
    digitalWrite(2, HIGH);
```

```
    digitalWrite(3, HIGH);
    digitalWrite(4, LOW);
  } // if the temperature rises more than 6 degrees, turn all LEDs on
  else{
    digitalWrite(2, HIGH);
    digitalWrite(3, HIGH);
    digitalWrite(4, HIGH);
  }
  delay(1);
}
```

Upload your code to the Arduino and verify that your project works correctly (see comments in code for description of how the project should work).

Demonstrate your project to your lecturer once completed.

Task 2 [20%]

Modify your code to also display (i.e. print to the serial monitor) how far above or below the baseline temperature the reading is, e.g.

```
Sensor: 140, Volts: .684, degrees C: 18.426, 1.574 degrees below baseline
Sensor: 150, Volts: .733, degrees C: 23.314, 3.314 degrees above baseline
```

Hint:
If the temperature is less than the baseline, print: baseline – temperature.
Otherwise, print: temperature – baseline.

Note that you should also print "above baseline" or "below baseline" as appropriate.

Copy and paste the changes made to your code into your report.

Task 3 [10%]

Modify your code such that, instead of hard-coding the baseline temperature, the Arduino takes an initial reading and uses this as the baseline temperature. This initial reading should be made in `setup()`, i.e. the baseline temperature is assigned in `setup()`.

Note that if you want to use the variables `sensorVal`, `voltage` and `temperature` in the initial reading, you might wish to create them as global variables rather than local, i.e. the variables will now be declared outside of the loop function. In this case, your code will look something like:

```
const int SENSOR_PIN = A0;
int sensorVal;
float voltage, temperature, baselineTemp;
// note: baselineTemp not a constant any more

void setup(){
  Serial.begin(9600);
  for(int pinNumber = 2; pinNumber < 5; pinNumber++){
    pinMode(pinNumber, OUTPUT);
    digitalWrite(pinNumber, LOW);
  }
  // read temperature here and assign to baselineTemp
}

void loop(){
  sensorVal = analogRead(SENSOR_PIN);
  // note: drop int since variable is already declared
  // etc.
}
```

Copy and paste your code into your report.

Task 4 [10%]

Add another LED to your circuit and modify your project as follows:

- Temperature less than two degrees above baseline: all LEDs off.
- 2 – 4 degrees above baseline: green LED on, other LEDs off.
- 4 – 6 degrees above baseline: green and amber LEDs on, other LEDs off.
- 6 – 8 degrees above baseline: green, amber and red LEDs on, other LED off.
- $>$ 8 degrees above baseline: green, amber and red LEDs on, other LED flashing.

Copy and paste your code into your report.

Demonstrate your project to your lecturer once completed.