Lab 6: Temperature Statistics

In this lab you will be using the Arduino temperature sensor to record the current temperature. Using this device you will be practice using arrays and computing some statistics on the input temperature data.

To do this lab you will need to:

1. Do the Tutorial on using the temperature sensor
2. Do the lab at the end of this document

## Tutorial: The Temperature Sensor

This tutorial is on how to drive the temperature sensor. The temperature sensor measures the environment’s temperature. For this module you should connect I to VCC and S to GND, we will be connecting the data transfer pin (the middle pin) to digital pin 2 on the Arduino. In order to run this tutorial you will need access to the OneWire module library as well as the DallasTemperature library. The code for global space and the setup function is shown below:

#include <OneWire.h>

#include <DallasTemperature.h>

//connect I to VCC

//connect S to GND

//connect the middle pin to digital 2

//setup onewire

OneWire oneWire(2);

//pass the onewire struct/object to

//the Dallas Temperature initializer

DallasTemperature sensor(&oneWire);

void setup(void)

{

//setup serial output

Serial.begin(9600);

// Start up the sensor

sensor.begin();

}

First we are including OneWire and DallasTemperature. We then create a OneWire object for digital communication between our sensor and the Arduino. Additionally we create an object called sensor of type DallasTemperature in global space and pass in the OneWire object. The DallasTemperature object is designed to communicate specifically to the temperature module through the serial digital communication module: OneWire.

In our setup function we are setting up the Serial object for logging our temperature to the Arduino console, we also setup the sensor object using the begin() function for our DallasTemperature object.

Finally the code below is for the Arduino loop function where we request the temperature and print it out. The requestTemperatures() function is called on sensor and the getTempCByIndex() function retrieves the temperature in Celsius. Here we are passing in 0 because we want to index the correct sensor (in this case there is only one sensor, but if there were more we could pass in 1,2,3..n for as many sensors as required. We use Serial to print out the temperature then wait for 100 milliseconds before loop is called again.

## Lab 6: Temperature Stats

In this lab we will be retrieving temperature data and computing various statistics on it. This is a common operation in computer science. Using a variable N (maximum value: 100) and an array of 100 floating point numbers named temps (float temps[100]), insert items into the array by means of a circular buffer (where there are a maximum of N items in the buffer). Periodically add the latest temperature into the array, replacing the oldest temperature currently stored. Then compute several statistics about the list. The statistics to be computed include: the mean (the average), the mode (the most frequently occurring temperature) and the median (if the list of temperatures were sorted, the median would be the number at the centre of the list).

Note: In order to implement a circular array, we need a counter: C with an initial value of 0 (int C = 0;). We can add a new item to our array using the modulo function with C:

temps[C++ % N] = getNewTemperature();

Then we can compute various statistics on the array, eg.

float mean = 0.0f;

for(int i = 0; i < N; i++)

mean += temps[i] / (float) N;

//output mean

Serial.println(mean);

We leave the median and mode for you to write.

Advanced: Early on in the program, there will not be N number of temperatures recorded, but currently our algorithm takes the initial values in temps as filler. Extend the application by only taking into account the temperatures and number of temperatures which have been recorded by the module. You will need to add another variable used in conjunction with C, this variable will represent the length of the array (which will eventually grow to N).