UNIVERSITY OF ALASKA FAIRBANKS

COLLEGE OF ENGINEERING AND MINES

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

COURSE CODE	EE F102 F01 (CRN: 32862)
COURSE NAME	INTRODUCTION TO ELECTRICAL AND COMPUTER ENGINEERING
SEMESTER	SPRING
YEAR	2023
LABORATORY LOCATION	JUB 331 (ELECTRONICS LAB)
LAB SESSION DATE AND TIME	MONDAY 06 MAR 2023
TYPE OF SUBMISSION	LABORATORY REPORT
NUMBER	7
TITLE OF SUBMISSION	ATMOSPHERIC TEMPERATURE DATA LOGGER
METHOD OF SUBMISSION	ONLINE VIA CANVAS
DUE DATE OF SUBMISSION	MONDAY 20 MAR 2023
DUE TIME OF SUBMISSION	23:59

STUDENT NAME	Jewel Maldonado

MAKE THIS FORM A "COVER PAGE" FOR YOUR REPORT SUBMISSION.

FOR THE TA USE ONLY

REMARKS:

ATMOSPHERIC TEMPERATURE DATA LOGGER

Objective

In this laboratory, we will put together all the pieces of our Atmospheric Temperature Data Logger and take data over the span of 24 hours. This lab is the culmination of the previous laboratory work.

System Performance and Observation

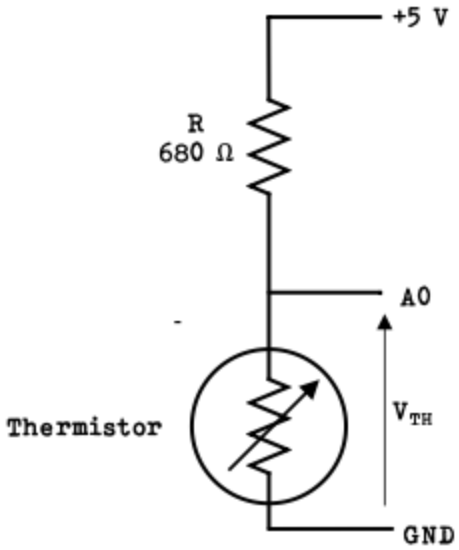

Figure 1. Circuit Diagram

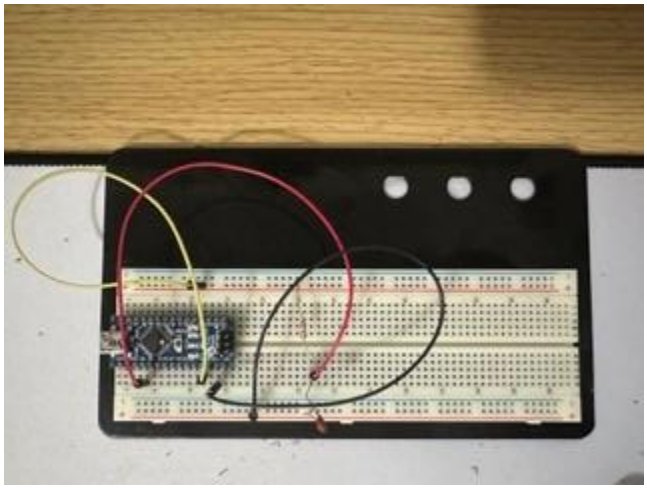

Figure 2. Built Circuit

In Figure 1 provided an understanding of how the circuit would be supplied power and receive input to the Arduino and how the thermistor would be connected to the Arduino in Figure 2.

$$V_{TH} = 5V\frac{R_{TH}}{R_{TH}+R} \quad (1)$$

Equation (1) shows how we measure the voltage received from the value measured from the voltage across pin **A0** and the ground point **GND** and this measured value is represented by V_{TH}. With V_{TH} we can go on to find R_{TH} with the equation (2) shown below.

$$R_{TH} = \frac{V_{TH} R}{5 - V_{TH}} \quad (2)$$

Equation (3) will be used to determine the thermistor temperature which is represented as T_{TH} . This will result in T_{TH} measured in Kelvin (**K**).

$$T_{TH} = \frac{1}{\frac{1}{B} ln(\frac{R_{TH}}{R_2}) + \frac{1}{T_2}} \quad (3)$$

Since we will be gathering the data in Fahrenheit we must convert from Kelvin to Fahrenheit first we will use equation (3) to calculate the temperature in Kelvin (**K**) and then use that value to convert to Fahrenheit in equation (4) below.

$$ºF = (9 \times \frac{T_{TH} - 273.15}{5}) + 32 \quad (4)$$

Each of the equations shown is implemented in the Arduino code as seen in *Sketch 1* in the Appendix on Lines 42 - 46.

Measuring the Data

In table 1 from the Appendix the data was measured in my home office. At some period of time, the heat had increased in my home and this was evidently recorded by the thermostat. The data was collected over 1 day, 10 hours, 56 minutes, and 24 seconds (125,784 seconds). The Arduino Program shown in Sketch 1 in the Appendix was programmed to loop every 10 minutes.

See Figure 3 below for the temperature over time.

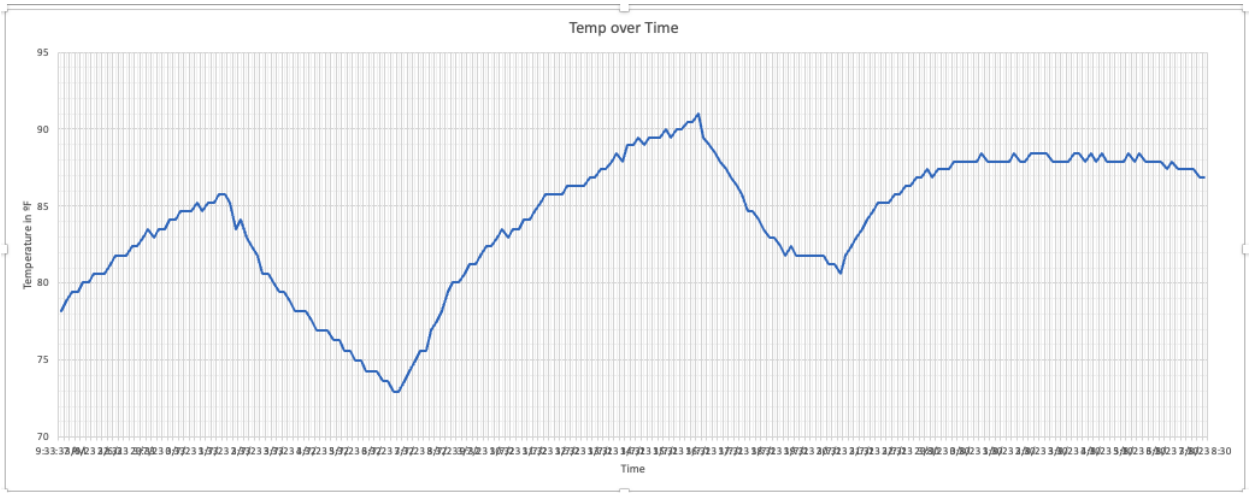

Figure 3. Plot of Temperature vs Time

Conclusion

Overall this lab gave me confidence in how I can measure data over a period of time using the Processing IDE program. Not only did I learn how to utilize new software through this lab but I also learned how something I use almost every day functions. If I could improve on this lab it would be how much time I spend learning how exactly the inputs from the thermistor are directly read to the Arduino. I enjoyed especially seeing how the code was implemented through the hardware.

Appendix

Sketch 1: UART_DATALOGGER.ino

```
/*
 * Created by : Thimira Thilakarathna
 * For : EE102
 * Edited By Jewel Maldonado for LAB 07 DATA_LOGGER of atmospheric temperature
 */

// Controls for the data logging system
#define  LOG_INTERVAL 600000 // milli seconds between entries
// 10 minute intervals = 600000 ms

// Time keeper
uint32_t timeStamp = 0; // The time stamp used when recording data points

//INCLUDE WHATEVER CONSTANTS YOU NEED HERE
int B = 4100;
int R1 = 680;
int R2 = 10000;
float T2 = 298.15;

//Include global variables here
int sensorValue=0;
float tempVol=0.0;
float tempRes=0.0;
float tempK=0.0;
float tempF=0.0;
float tempC=0.0;

char tempCString[7];

// This variable will hold the data from a read cycle
String dataString = " ";

void setup(void) {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop(void) {
  // put your main code here, to run repeatedly:

  //code from lab4
  int tempVal=analogRead(A0);
  float tempVol=tempVal*5.0/1023.0;
  float tempRes=R1*tempVol/(5.0-tempVol);
  float tempK=1.0/((1.0/B)*log(tempRes/R2)+(1.0/T2));
  float tempF=(9.0*(tempK-273.15)/5.0)+32.0;
  Serial.println(tempF);
  // Delay your time interval
  delay(LOG_INTERVAL);

}
```

Table 1. LogFile of Data Measured

Date[yyyy/mm/dd] time	Temp in F
2023/3/6 21:33:37	78.18
2023/3/6 21:43:36	78.8
2023/3/6 21:53:35	79.42
2023/3/6 22:03:34	79.42
2023/3/6 22:13:33	80.03
2023/3/6 22:23:32	80.03
2023/3/6 22:33:31	80.63
2023/3/6 22:43:30	80.63
2023/3/6 22:53:28	80.63
2023/3/6 23:03:27	81.22
2023/3/6 23:13:26	81.81
2023/3/6 23:23:25	81.81
2023/3/6 23:33:24	81.81
2023/3/6 23:43:23	82.4
2023/3/6 23:53:22	82.4
2023/3/7 0:03:21	82.97
2023/3/7 0:13:20	83.54
2023/3/7 0:23:19	82.97
2023/3/7 0:33:18	83.54
2023/3/7 0:43:17	83.54
2023/3/7 0:53:16	84.11
2023/3/7 1:03:15	84.11
2023/3/7 1:13:14	84.67
2023/3/7 1:23:13	84.67
2023/3/7 1:33:12	84.67
2023/3/7 1:43:11	85.22
2023/3/7 1:53:10	84.67
2023/3/7 2:03:09	85.22
2023/3/7 2:13:08	85.22
2023/3/7 2:23:07	85.77
2023/3/7 2:33:06	85.77
2023/3/7 2:43:05	85.22
2023/3/7 2:53:04	83.54
2023/3/7 3:03:03	84.11
2023/3/7 3:13:02	82.97
2023/3/7 3:23:01	82.4

2023/3/7 3:33:00	81.81
2023/3/7 3:42:59	80.63
2023/3/7 3:52:58	80.63
2023/3/7 4:02:57	80.03
2023/3/7 4:12:56	79.42
2023/3/7 4:22:54	79.42
2023/3/7 4:32:53	78.8
2023/3/7 4:42:52	78.18
2023/3/7 4:52:51	78.18
2023/3/7 5:02:50	78.18
2023/3/7 5:12:49	77.55
2023/3/7 5:22:48	76.91
2023/3/7 5:32:47	76.91
2023/3/7 5:42:46	76.91
2023/3/7 5:52:45	76.27
2023/3/7 6:02:44	76.27
2023/3/7 6:12:43	75.62
2023/3/7 6:22:42	75.62
2023/3/7 6:32:41	74.96
2023/3/7 6:42:40	74.96
2023/3/7 6:52:39	74.29
2023/3/7 7:02:37	74.29
2023/3/7 7:12:36	74.29
2023/3/7 7:22:35	73.61
2023/3/7 7:32:34	73.61
2023/3/7 7:42:33	72.92
2023/3/7 7:52:32	72.92
2023/3/7 8:02:31	73.61
2023/3/7 8:12:30	74.29
2023/3/7 8:22:29	74.96
2023/3/7 8:32:28	75.62
2023/3/7 8:42:27	75.62
2023/3/7 8:52:26	76.91
2023/3/7 9:02:25	77.55
2023/3/7 9:12:24	78.18
2023/3/7 9:22:23	79.42
2023/3/7 9:32:22	80.03
2023/3/7 9:42:20	80.03
2023/3/7 9:52:19	80.63

2023/3/7 10:02:18	81.22
2023/3/7 10:12:17	81.22
2023/3/7 10:22:16	81.81
2023/3/7 10:32:15	82.4
2023/3/7 10:42:14	82.4
2023/3/7 10:52:13	82.97
2023/3/7 11:02:12	83.54
2023/3/7 11:12:11	82.97
2023/3/7 11:22:10	83.54
2023/3/7 11:32:09	83.54
2023/3/7 11:42:08	84.11
2023/3/7 11:52:07	84.11
2023/3/7 12:02:06	84.67
2023/3/7 12:12:05	85.22
2023/3/7 12:22:04	85.77
2023/3/7 12:32:03	85.77
2023/3/7 12:42:02	85.77
2023/3/7 12:52:01	85.77
2023/3/7 13:02:00	86.32
2023/3/7 13:11:59	86.32
2023/3/7 13:21:58	86.32
2023/3/7 13:31:57	86.32
2023/3/7 13:41:56	86.86
2023/3/7 13:51:55	86.86
2023/3/7 14:01:54	87.39
2023/3/7 14:11:53	87.39
2023/3/7 14:21:52	87.92
2023/3/7 14:31:51	88.45
2023/3/7 14:41:50	87.92
2023/3/7 14:51:49	88.97
2023/3/7 15:01:48	88.97
2023/3/7 15:11:47	89.48
2023/3/7 15:21:46	88.97
2023/3/7 15:31:45	89.48
2023/3/7 15:41:44	89.48
2023/3/7 15:51:43	89.48
2023/3/7 16:01:42	89.99
2023/3/7 16:11:41	89.48
2023/3/7 16:21:40	89.99

2023/3/7 16:31:39	89.99
2023/3/7 16:41:38	90.5
2023/3/7 16:51:37	90.5
2023/3/7 17:01:36	91
2023/3/7 17:11:35	89.48
2023/3/7 17:21:34	88.97
2023/3/7 17:31:32	88.45
2023/3/7 17:41:31	87.92
2023/3/7 17:51:30	87.39
2023/3/7 18:01:29	86.86
2023/3/7 18:11:28	86.32
2023/3/7 18:21:27	85.77
2023/3/7 18:31:26	84.67
2023/3/7 18:41:25	84.67
2023/3/7 18:51:24	84.11
2023/3/7 19:01:23	83.54
2023/3/7 19:11:22	82.97
2023/3/7 19:21:21	82.97
2023/3/7 19:31:20	82.4
2023/3/7 19:41:19	81.81
2023/3/7 19:51:18	82.4
2023/3/7 20:01:17	81.81
2023/3/7 20:11:16	81.81
2023/3/7 20:21:15	81.81
2023/3/7 20:31:14	81.81
2023/3/7 20:41:13	81.81
2023/3/7 20:51:12	81.81
2023/3/7 21:01:11	81.22
2023/3/7 21:11:10	81.22
2023/3/7 21:21:09	80.63
2023/3/7 21:31:08	81.81
2023/3/7 21:41:07	82.4
2023/3/7 21:51:06	82.97
2023/3/7 22:01:05	83.54
2023/3/7 22:11:04	84.11
2023/3/7 22:21:03	84.67
2023/3/7 22:31:02	85.22
2023/3/7 22:41:01	85.22
2023/3/7 22:51:00	85.22

2023/3/7 23:00:59	85.77
2023/3/7 23:10:58	85.77
2023/3/7 23:20:56	86.32
2023/3/7 23:30:55	86.32
2023/3/7 23:40:54	86.86
2023/3/7 23:50:53	86.86
2023/3/8 0:00:52	87.39
2023/3/8 0:10:51	86.86
2023/3/8 0:20:50	87.39
2023/3/8 0:30:49	87.39
2023/3/8 0:40:48	87.39
2023/3/8 0:50:47	87.92
2023/3/8 1:00:46	87.92
2023/3/8 1:10:45	87.92
2023/3/8 1:20:44	87.92
2023/3/8 1:30:43	87.92
2023/3/8 1:40:42	88.45
2023/3/8 1:50:41	87.92
2023/3/8 2:00:40	87.92
2023/3/8 2:10:39	87.92
2023/3/8 2:20:38	87.92
2023/3/8 2:30:37	87.92
2023/3/8 2:40:36	88.45
2023/3/8 2:50:35	87.92
2023/3/8 3:00:34	87.92
2023/3/8 3:10:33	88.45
2023/3/8 3:20:32	88.45
2023/3/8 3:30:31	88.45
2023/3/8 3:40:30	88.45
2023/3/8 3:50:29	87.92
2023/3/8 4:00:28	87.92
2023/3/8 4:10:27	87.92
2023/3/8 4:20:26	87.92
2023/3/8 4:30:25	88.45
2023/3/8 4:40:24	88.45
2023/3/8 4:50:23	87.92
2023/3/8 5:00:22	88.45
2023/3/8 5:10:21	87.92
2023/3/8 5:20:20	88.45

2023/3/8 5:30:19	87.92
2023/3/8 5:40:18	87.92
2023/3/8 5:50:17	87.92
2023/3/8 6:00:16	87.92
2023/3/8 6:10:15	88.45
2023/3/8 6:20:14	87.92
2023/3/8 6:30:13	88.45
2023/3/8 6:40:12	87.92
2023/3/8 6:50:11	87.92
2023/3/8 7:00:10	87.92
2023/3/8 7:10:09	87.92
2023/3/8 7:20:08	87.39
2023/3/8 7:30:07	87.92
2023/3/8 7:40:06	87.39
2023/3/8 7:50:05	87.39
2023/3/8 8:00:04	87.39
2023/3/8 8:10:03	87.39
2023/3/8 8:20:02	86.86
2023/3/8 8:30:01	86.86