Memo

To: Professor Pisano

From: OccuSense ECE Senior Design Team

Team: OccuSense: Team 11

Date: 11/15/16

Subject: First Deliverable Test Report

1.0 Project Objective

The overall objective of this project is to create a real-time, accurate and reliable sensor system that determines the occupancy of a room. With this information, air circulation in HVAC (Heating Ventilation & Air Circulation) systems can be adjusted according to the number of people in the room. This will not only save energy but provide useful feedback regarding current and past room usage.

2.0 Test Objective and Significance

2.1 The goal of this test is to set up the thermal sensor, receive data and extract meaningful information from the data. This is an important step in the project as it is will allow for the creation of heat-maps of the doors' entrances and exits. Using these heat-maps and thermal data, it can be determined if a person is entering or exiting a room. This deliverable is essential because it will verify the proper detection of movement by the sensor which can be graphically represented as a heatmap image.

2.2 The test setup is displayed in Figure 1. This test utilizes an MLX90621 thermal sensor, an Arduino Mega 2560 and a computer.

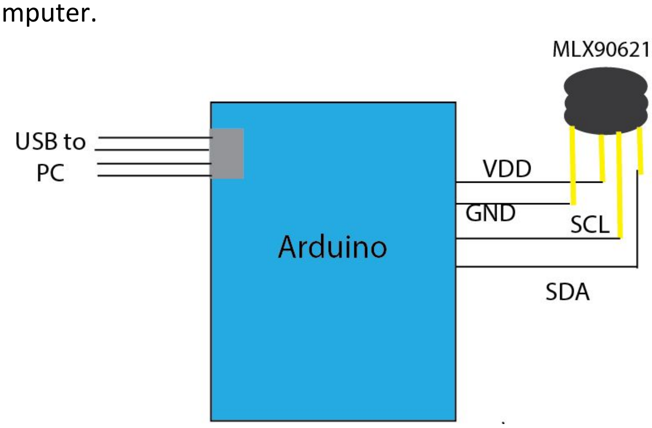


Figure 1: Diagram of the set-up between the thermal sensor the arduino and the computer

2.3 MLX90621 Thermal Sensor

The MLX90621 contains 64 IR pixels (as a 16x4 array) that detects objects moving in front of the sensor and a PTAT (Proportional to Absolute Temperature) sensor to measure the ambient temperature of the chip. The outputs of these sensors are stored in an internal RAM and can be accessed via I^2C. This process begins by reading the EEPROM calibration data that is used to prep the thermal sensor data points to be processed into ambient temperature (Ta) readings. The measurement data (PTAT and IR readings) stored in the internal ram of the sensor is used to calculate the Ta values of the thermopile array.

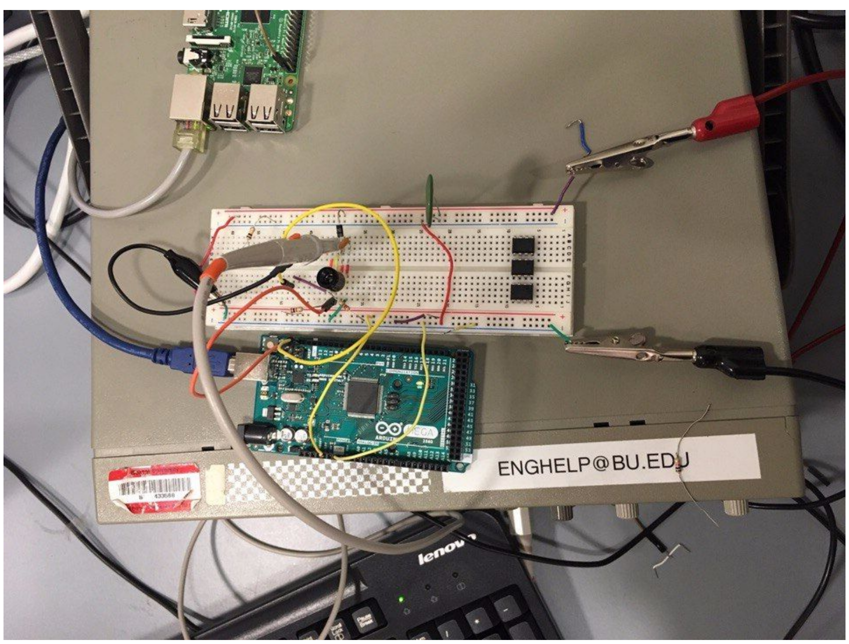
2.4 Arduino Mega 2560

The arduino mega 2560, connected to the thermal sensor and computer, is used to process the data signals from the sensor using I^2C and transmit the thermal data to the computer. The arduino is a microcontroller with 54 digital I/O pins, 16 analog inputs, 4 UARTs, a 16 MHz crystal oscillator and a USB connection.

3.0 Equipment and Setup

3.1 Overall Setup

The Arduino is powered by 5.0V from the USB connection. The SCL and SDL pins of the thermal sensor, as can be seen in Figure 3, correspond to the analog CLK and DATA pins on the Arduino. The MLX90621 thermal sensor is powered by 2.6V coming from a diode through Pin 3 (VDD) and Pin 4 (GND) is wired to ground. The pullup resistor values that we used for the circuit ar 10 kohms. The design of the system can be observed in Figure 4.



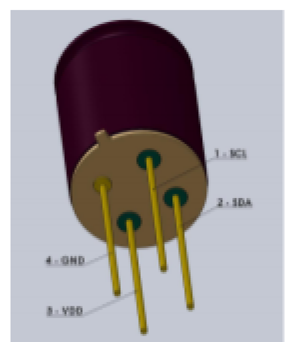


Figure 3: 3D Render of the MLX90621 Thermal Sensor

Figure 4: Picture of circuit board and Arduino Mega 2560/MLX90621 integration

3.2 **MLX90621 Thermal Sensor**

To test the functionality of the MLX906021 a python code will be ran to process the raw signal data collected by the sensor. Once the signals are processed into ambient temperature data (Celsius) the readings can be observed by passing objects with variable temperatures across the sensor's field of view. The python code will then produce the 16X4 temperature array corresponding to the object. These measurements can be compared to the pre measured temperatures of these objects in order to validate the accuracy and functionality of the sensor.

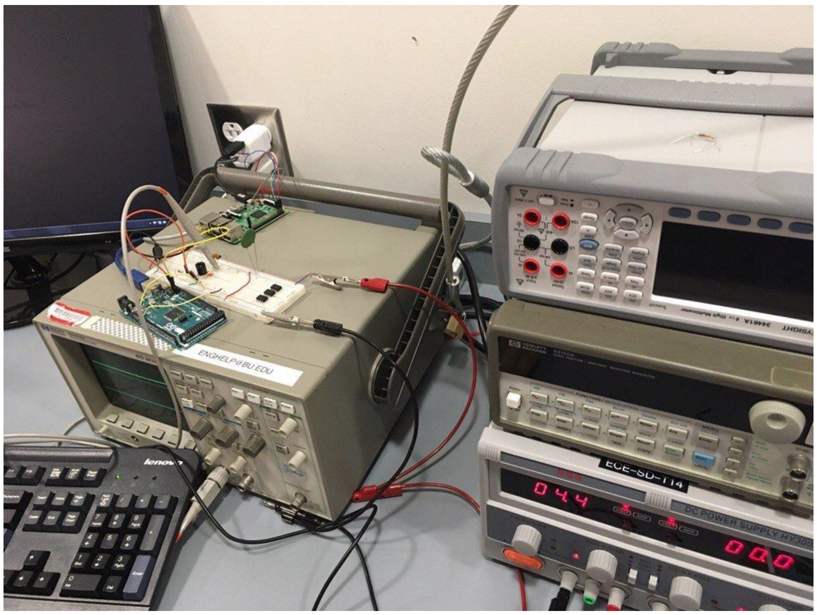


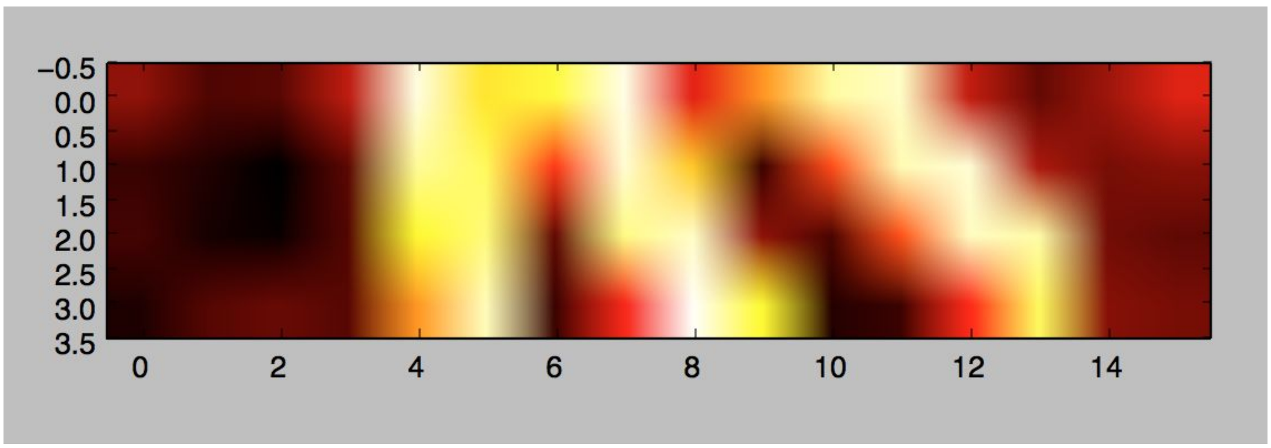
Figure 5: Additional picture of system setup including power source and oscilloscope

4.0 Measurements and Data

4.1 **16x4 Thermal Array** (°Celsius)

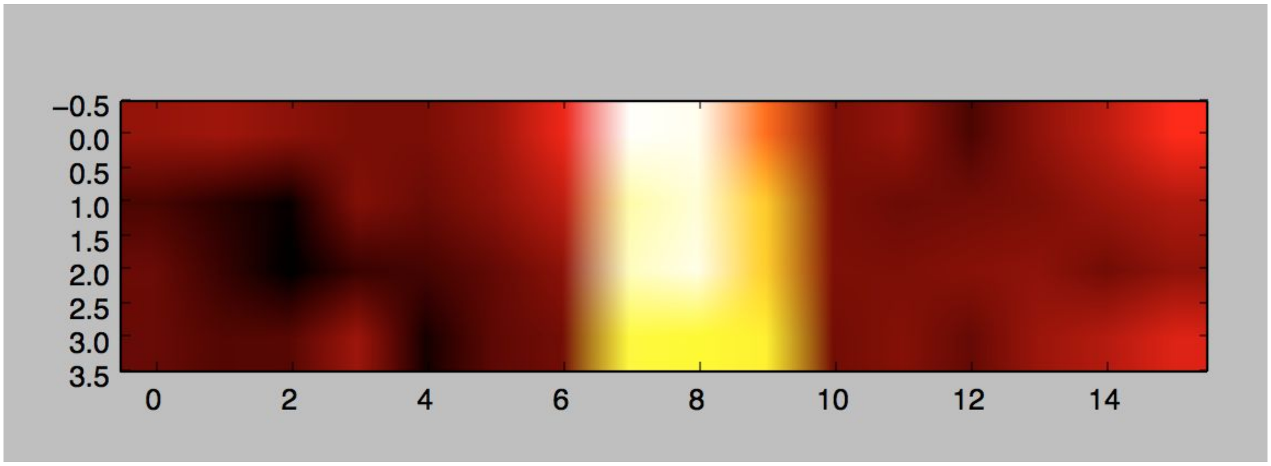
For testing the 16X4 Thermal Array produced by the python signal processing script, data was collected by observing the heat map under three different cases. The three situations being no heated objects near the thermal sensor, hand placed 5 inches from the sensor, and a finger 2 inches on top of the sensor. From these three cases the 16X4 thermal array and array of ambient temperatures (Ta) are collected. We have been able to also take a snapshot of the heatmap from the array as well. Here are the following heatmaps and their corresponding arrays.

Case 1: Hand about 4 inches above the sensor



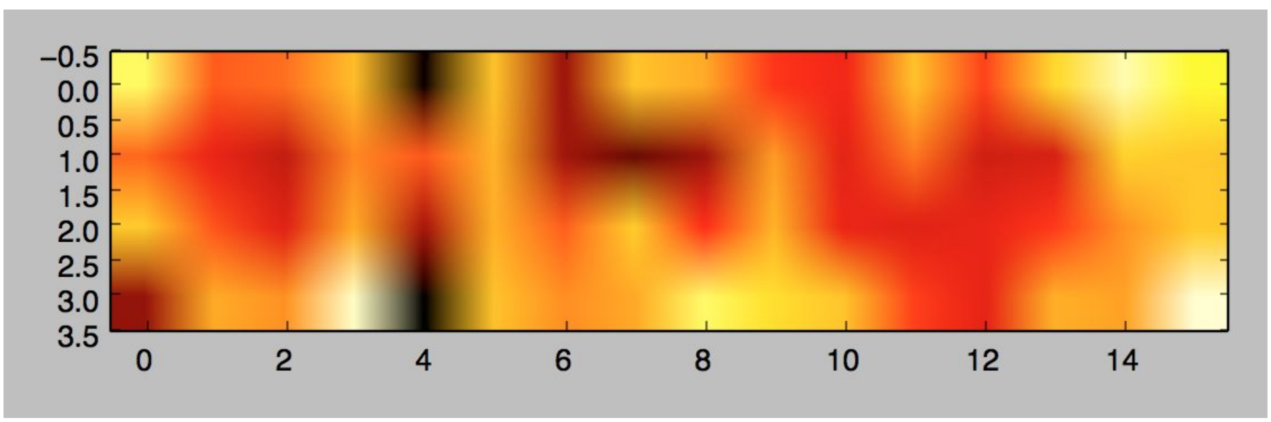
[23.13, 22.21, 22.32, 23.85, 31.74, 28.3, 29.04, 31.84, 24.53, 26.87, 30.65, 31.28, 23.94, 22.47, 23.33, 24.45, 21.92, 21.66, 21.31, 22.36, 30.51, 29.53, 25.38, 31.51, 27.76, 21.92, 25.69, 31.0, 31.56, 23.55, 22.78, 23.02, 22.06, 21.54, 21.44, 22.26, 32.41, 30.02, 22.43, 30.18, 31.3, 23.09, 22.14, 25.77, 31.21, 30.77, 22.68, 22.42, 21.63, 22.31, 22.55, 22.3, 26.88, 31.11, 21.92, 24.87, 32.23, 28.73, 21.71, 21.95, 25.04, 29.57, 22.97, 22.76]

Case 2: Finger 3 inches above the sensor



[21.59, 21.72, 21.45, 21.2, 21.21, 21.67, 22.9, 29.66, 29.45, 24.29, 21.22, 21.57, 20.63, 21.48, 22.17, 23.19, 20.62, 20.28, 19.99, 21.32, 21.0, 21.43, 22.13, 28.36, 29.07, 25.68, 21.24, 21.01, 21.1, 21.22, 21.62, 21.94, 21.02, 20.43, 19.88, 20.43, 20.55, 20.91, 21.43, 28.71, 29.34, 25.74, 21.25, 21.24, 21.37, 21.47, 21.11, 21.49, 20.99, 20.74, 20.78, 21.7, 20.09, 20.86, 21.11, 26.77,29.8,26.32,21.12,21.36,20.95,21.66, 22.02, 22.65]

Case 3: Nothing on top of the sensor



[22.56, 21.72, 21.79, 22.09, 20.69, 22.11, 21.12, 22.12, 22.02, 21.58, 21.46, 22.11, 21.63, 22.2, 22.86, 23.2, 21.76, 21.42, 21.26, 21.88, 21.71, 22.05, 21.14, 20.94, 21.12, 21.96, 21.4, 21.82, 21.31, 21.34, 22.18, 22.13, 22.15, 21.69, 21.38, 22.02, 21.17, 22.04, 21.73, 22.16, 21.54, 22.05, 21.43, 21.38, 21.42, 21.59, 21.92, 22.14, 21.09, 22.02, 21.92, 22.94, 20.66, 22.11, 21.91, 22.01, 22.59, 22.23, 22.12, 21.62, 21.41, 22.04, 21.98, 22.99]

5.0 Conclusions

5.1 **16x4 Thermal Array** (°Celsius)

The criteria for success for the 16X4 Thermal Array are that it produces a real-time visual representation of the ambient heat temperatures (celsius) being recorded by the MLX90621 thermal sensor. Although we are currently not able to produce a real-time heat array of the data, we are able to measure the 16X4 array of the raw ambient temperature data points and plot static pictures at given times. The next step will be to produce the real-time thermal array form these data points and demonstrate this for the next design review.

5.2 **Summary and Future Plans**

From the measurements collected from our testing, it can be concluded that we are successfully able to collect ambient temperature data from the MLX90621 thermal sensor using an Arduino Mega 2560 to process the raw signal data. We are also able to render a 16X4 thermopile heat array for any given moment during data collection. The next step will be to develop a python script that will produce a 16X4 real-time heat map based off data being constantly collected by our MLX90621/Arduino system. We will use this heat map to draw conclusions on whether or not a 16X4 array is sufficient for the needs of the project. If not, we will retest the same design with a Panasonic thermal sensor that produces an 8X8 array of ambient temperatures. We are also planning on implementing a Raspberry Pi, rather than the Arduino Mega, in a future design of our system once we get our initial design with the Arduino fully functional. The real-time heat map and comparison between the 16X4 and 8X8 thermal sensor data will be demonstrated during the next design review.