Memo

To: Professor Pisano

From: Occusense ECE Senior Design Team

Team: Occusense: Team 11

Date: 3/30/2017

Subject: Functional Testing 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 Directionality Detection

The goal of this test is to verify that the Occusense sensor system can differentiate between someone walking into a room and out of a room. The updated version of the python people counting algorithm will send a +1 to the Firebase web server application when somebody walks in and a -1 value when somebody walks out of the room. The significance of this test is to show that the Occusense sensing system is fully capable of counting the number of people walking in and out of the room and thus can accurately keep track of the occupancy of an assigned room.

2.2 24-Hour Collection

The goal of this test is to verify that the Occusense sensor system can accurately collect data over a prolonged amount of time. We will test this by running the system for a prolonged amount of time prior to the time of testing. By then, we will have numerous time-stamped data entries in the Firebase web server. This data will also be displayed in the form of a Time vs. Occupancy graph on the Occusense web application. The significance of this test is to show that the Occusense web server and web application is able to keep up with the constant data collection of the sensor and display the data in real-time and historically.

3.0 Setup and Equipment

3.1 The test setup is displayed in Figure 1. This test utilizes an MLX90621 thermal sensor, Arduino Mega 2560, Raspberry Pi, PIR sensor and Firebase hosted database and web application.

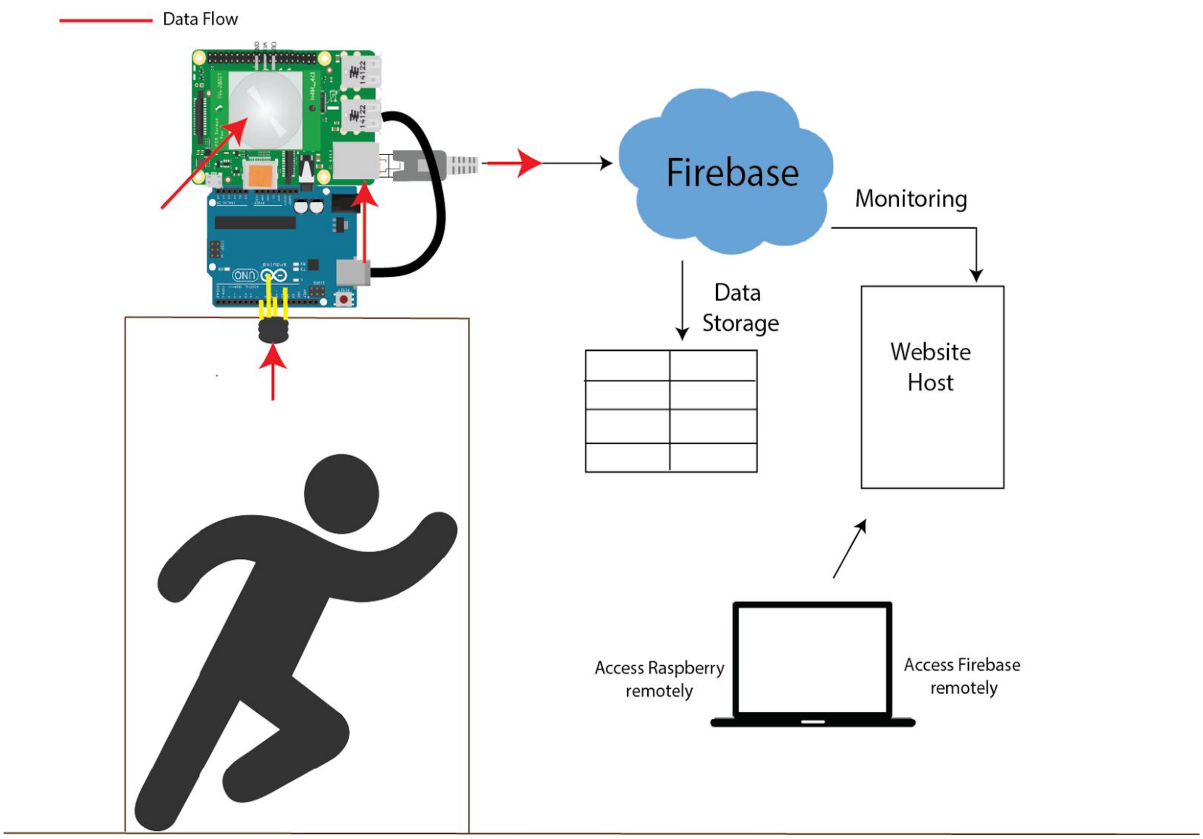

Figure 1: This diagram shows all the equipment setup and the red lines symbolize the flow of data of the integrated system. The input data comes from the thermal sensor and the PIR sensor which is then processed and displayed in the website host.

3.2 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. The MLX90621 will be mounted 6 inches above the center of the doorway and running at 16 frames per second.

3.3 Arduino Mega 2560

The Arduino Mega 2560, connected to the thermal sensor and raspberry pi, is used to process the data signals from the sensor using I^2C and transmit the thermal data to the Raspberry Pi. 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.

3.4 Raspberry Pi

For this test, The Raspberry Pi's role is to house the Python server client program that initiates the thermal data transfer from the Raspberry Pi to the Firebase web server. We utilize the Raspberry Pi's processing power and ability to connect to ethernet/wifi in order to act as the connection between the Arduino Mega that is actively processing the thermal data and the Firebase database that will store the real-time and historical data.

3.5 Firebase Web Server

The Firebase database is used to store the real-time and historical occupancy data for the specific room (Figure 2). So far the database stores three pieces of information per measurement: date, time, and value. The date and time of the measurement will be used to analyze the occupancy data historically. From this data we should be able to infer what days and times of days the room is most busy and adjust the power usage of our system accordingly. The value portion of the data is most directly used to update the real-time occupancy of the room. The value field will come in as a +1 or -1 and will keep a running total of the occupancy of the room for the client.

3.6 Web Application

The web application will house the web user interface for the Occusense sensing system. Here, the client will be able to login and access the real-time and historical occupancy data of a specified room. This web application is housed on the Firebase web server and will draw the real-time (Figure 2) and historical data (Figure 3) directly from the real-time database also housed on the Firebase web server.

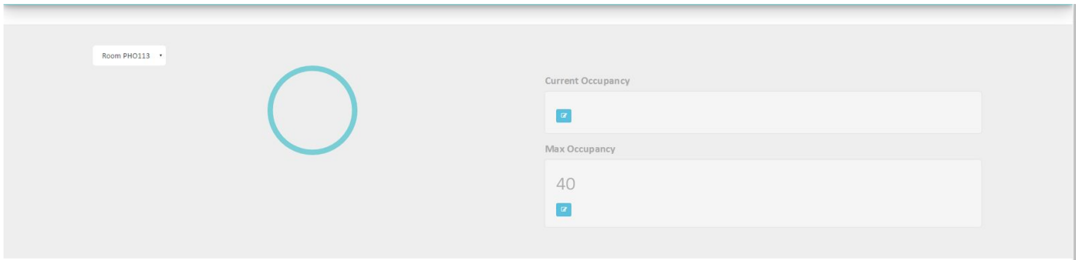

Figure 2: Real-time occupancy data displayed on website

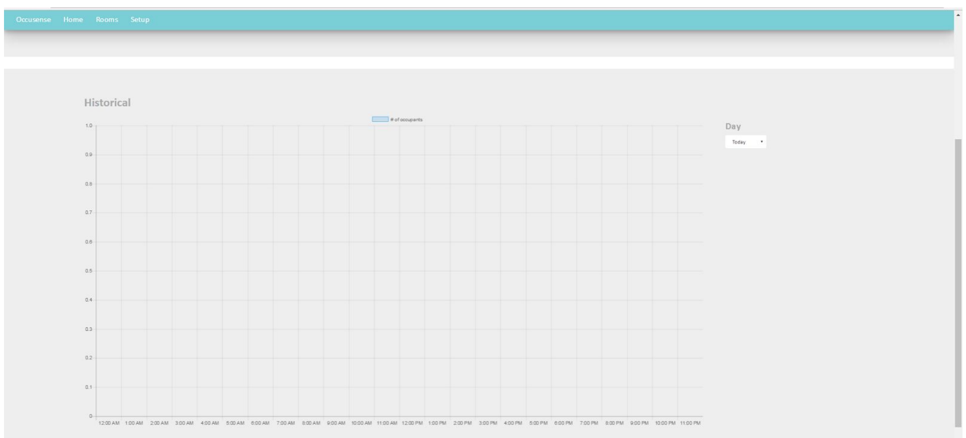

Figure 3: Historical occupancy data display

4.0 Measurement and Data

4.1 16 X 4 Thermopile Array

Figure 4: Thermal array when there is no one in the frame.

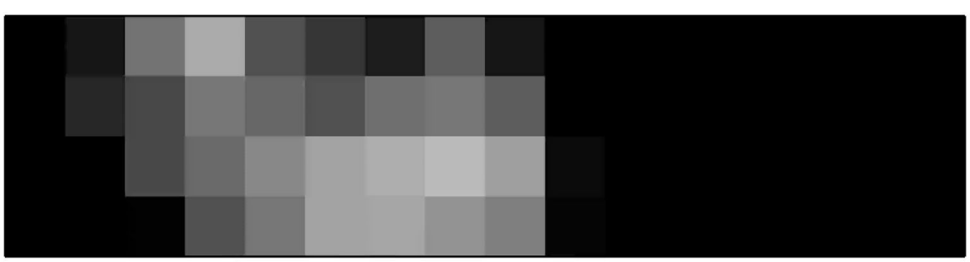

Figure 5: Person in the frame walking out of the room.

4.2 Average/Standard Variations of Temperatures Collected

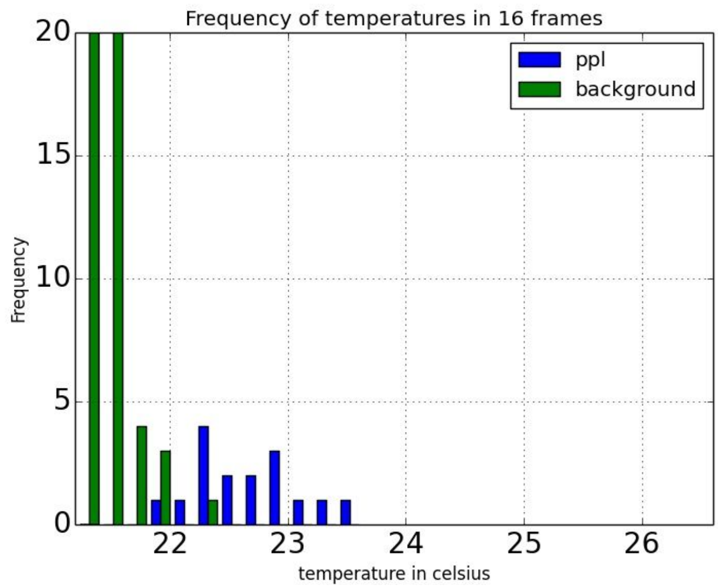

Figure 5: Average of temperatures collected in 16 frames

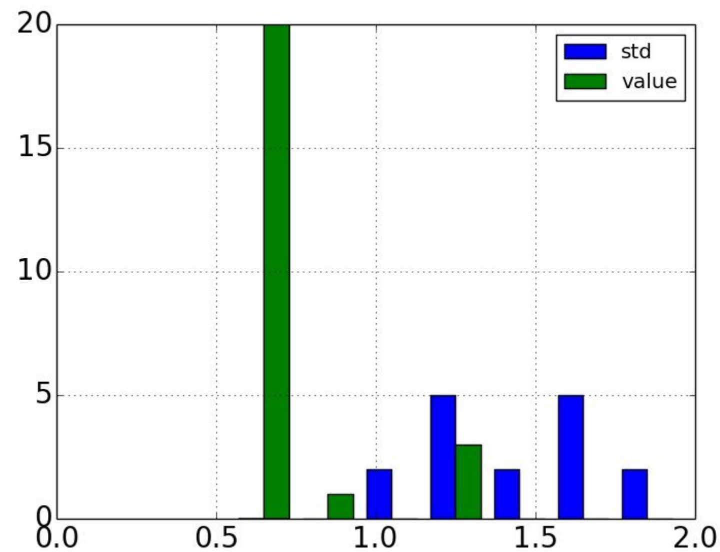

Figure 6: Frequency of standard deviations of all 16 frames

4.3 Detection and Direction Algorithm:

The algorithm running on the raspberry pi processes the temperature array frames from the arduino and detects when people walk in or out of the room and sends the result to the server. The detection is based on a threshold. Figure 5 and 6 show the two variables we are considering in determining what frames contain people. If the current frame has a variance bigger than 1.0 or an average temperature that is bigger than the running average by about 1 degree we know that someone is walking through the door.

When none is walking through the doorway the algorithm performs background subtraction, keeping an average of the background temperatures of each pixel. Then when the threshold is passed it subtracts that average to each pixel, getting a background subtracted frame that we append to an array until another background frame is reached. The direction function is then applied to the array with the background subtracted frames (frames of a person walking through) to determine if that person walked in or out.
The direction algorithm uses the last 5 frames of that array. It starts by computing a time series array of the hottest pixel in the first frame throughout the last frame. It then creates a cross-correlation array of that time-series to the time-series array of the remaining 63 pixels, getting a good idea of what direction the person is going. Another function then analyzes the sum and the count of negative values in each row of the cross-correlation array and determines the direction. Before pushing the value to the server it error-checks to make sure that the last frame makes sense with the computed direction (again by looking at the rows of the last frame).

The day of the testing we were in the lab for about 4-5 hours and achieved an accuracy rate of 80% detecting people going in and out. We have some ideas to simplify the algorithm and potentially doing some more sophisticated error checking to improve the accuracy even more. We are also setting up a camera to have a ground truth value and see how different tweaks to the algorithm improve the overall long-term accuracy.

5.0 Conclusions

5.1 We are able to demonstrate that we have a fully functional room occupancy sensing system. We are able to collect historical and real-time data of the occupancy of an assigned room by tracking the number of people entering and exiting the doorway. The updated version of the people counting algorithm loaded on the RPi takes into account directionality of the person.

5.2 Future Plans and Summary

Since we now have a fully functional Occusense room occupancy sensing system up and running we will attempt to develop a more robust machine learning algorithm that caters to more unique cases of people passing underneath a doorway. So far, we are able to detect one person leaving or entering a room with almost perfect precision. We are able to do this as long as there is ample separation between people. Cases that we intend on improving are when people are passing underneath the sensor in close proximity and when people hold the door open for others while underneath the sensor. Other future plans include setting up a second system that will work in addition to our original system. This way, if a room has two doorways both entrances can be monitored.