Occusense Sensor System Full Connectivity

To: Professor Pisano

From: Occusense ECE Senior Design Team

Team: Occusense: Team 11

Date: 2/15/17

Subject: Second Deliverable Test Plan

Goal & Description:

The goal of this test is to demonstrate that we have successfully connected all parts of the Occusense sensor system. These parts include the MLX90621 thermal sensor, Arduino Mega 2560, Raspberry Pi, PIR reset sensor and Firebase web server. In order to validate that we are able to do this we will have someone walk under our sensor and track the data from start to finish and have a mock up test for the PIR reset system. The data is first processed on the Arduino Mega 2560 as a 16X4 thermopile array then sent to the Raspberry Pi where it will be processed into a +1 depending on the activity under the sensor. Over ethernet connection, this value is pushed to the Firebase web server which houses a real-time database. This database will hold the time, date, and value of the data. This data will be pushed to the web application also hosted on the Firebase web server. Finally, the real-time occupancy of the room will be updated on the web user interface. On another test, we will first prove that the PIR sensor is working by showing real time data and we will also send a mock reset request to the server so that the count stored resets to zero. We will track the process at every stage to prove that we have a fully connected and functional system. This will be very important to show that we can meet the project objectives in our final deliverable. With our system working well together we can focus on getting the main algorithm as efficient and accurate as possible and on getting a well designed and presented product with everything fully integrated.

Procedure:

In preparation for running this test the MLX90621 thermal sensor, Arduino Mega 2560, and Raspberry Pi will all be placed above the door to the senior design lab. The C++ script used to collect and process the raw thermal data picked up by the MLX90621 sensor will then be run. At the same time, we will run the Python-based client server program (which will host the main algorithm) that will take in the 16X4 thermopile array, process it and convert it into a +1 based on the data. This program will then push the time, date, and value data to the Firebase web server. Once the data is sent to our real-time database hosted on the Firebase web server we will be able to track the incoming data via the online console connected to the database. From here the data will be pushed in real time to the web application, also hosted on the Firebase web server, where the occupancy of the room will be updated as data is pushed by the Raspberry Pi. We can then compare the activity of the database console and web application to confirm that all of the parts of the Occusense sensor system are working in unison and has a fully functional datapath. To test the PIR we will first show that we can successfully receive data from it and we will after send a mock reset to the Firebase to reset the count to zero. Below is a diagram of all the parts of the system with arrows to show the flow of data.

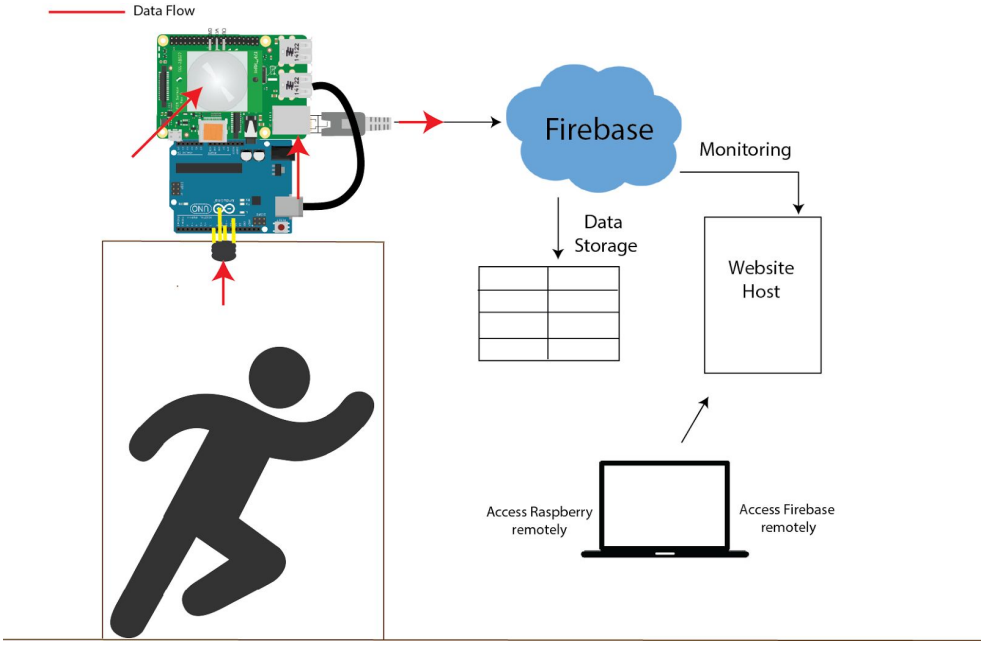

Figure 1: System configuration for the test plan with data flow illustration

Verifiable Goals:

By the end of this test we should be able to send data gathered from the MLX90621 and have it successfully run through the datapath of the sensing system. At the conclusion of this test we should be able to sense somebody walking under the MLX90621. If we can we hope to be able to also detect direction of a single person walking in or out and thus be able to send a +1, it is not however the main aspect to test in this test plan. The main test is to use data from the sensor to update the real-occupancy that is being observed on our web application and stored in our real time database. We should also be able to collect data from the PIR and successfully push a reset command to the firebase to update its count. These two goals demonstrate the ability of the system to work synchronously with all the parts, leaving the algorithm as the only real task left to accomplish.