Final Project Report

EE 249 Fall 2014

The Marauder Map

LongXiang Cui, Ying Ou, Jordan Makansi, Chenyang Xu
Dec 19th 2014

Introduction

Certain building operations are necessary for an inhabited building, to but many of these operations persist even when the building is uninhabited. Knowledge of building occupancy can reduce these unnecessary operations. Two such examples are energy conservation (HVAC/lighting), and crowdedness; if a room is known to be empty, then the lights and AC can turn off, and likewise if a room hosting a public service (such as a library) is too crowded, people will go elsewhere for this service. To sense building occupancy, The Marauder's Map implemented a system that tracks the flow of people in and out of a doorway.

The implementation of the system was divided into two main components. The first component is the actual sensing of the physical system, and modeling of the physical dynamics. This involved decisions regarding sensors to use, the number of sensors, and placement of sensors. The second component transmits and displays the data in a useful manner, so that it may be used to impact observers. This involved decisions about what piece of hardware to use for communication, and how to display the data. Both components when through many design iterations, as will be discussed below. The objective of these iterations was guided by the expected functionality: to accurately sense building occupancy, and display it in understandable way.

Design

The hardware decisions were driven by not only the functionality of the system, but also the tradeoff between cost and reliability. Figure 1 shows a complete diagram of the system implementation. Since it was estimated that only a few sensors would be needed, it was determined that an investment in the ultrasonic ranging module, HC SR-04 was more fitting than the cheaper alternative - infrared sensors.

Four sensors were used in total - three on the ground for detecting people, and one for detecting the opening and closing of the door. For the door sensor, a threshold of 60 cm was used for determining the opening and closing of the door. The combined information of these four sensors is sufficient to determine if a person has left or entered the room via the doorway.

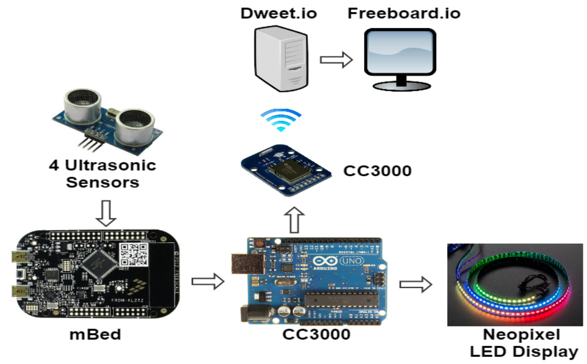


Figure 1. The block diagram of the system and data flow

Ultrasonic sensors

An ultrasonic sensor works by computing a distance based on the time that it takes for the sent signal to return, and the width of this "echo pulse" as seen in the figure below, is proportional to the distance to the object detected, as shown by the formula in the figure.

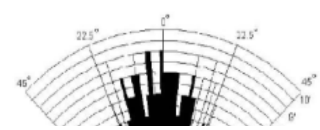


Figure 2a. Angle range of Ultrasonic sensor

The nature of the HC SR-04 is such that it has a detection angle range of 30 degrees in which it is accurate (HC SR-04 User's Manual), also seen in Figure 2. The ultrasonic sensor works by a pulse width, which is proportional to the distance that it senses. These qualities of the HC SR-04 affected the logical analysis used to determine the flow of people in and out of the room.

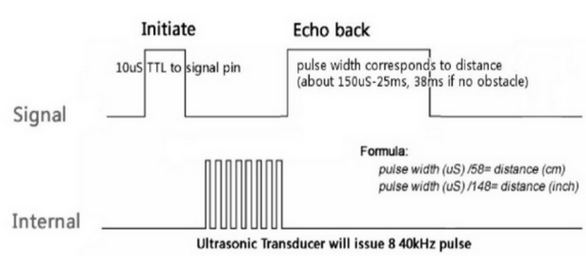


Figure 2. The detection angle (top) and detection mechanism of ultrasonic sensor (bottom)

The sensors are connecting to an mbed board, which does the logical analysis of the information (*Description of the*

Algorithm for details). After connecting the four sensors to the mbed board, there were not enough pins to complete the functionality of a wifi shield. So, an Arduino board was connected to mbed, which concurrently transmits the change in occupancy via a CC 3000 wifi shield to an online server (dweet.io) and lights up Neopixel LEDs. The mbed board is constantly sensing - Arduino only sends data when there is a change in the count.

Two types of feedback are given to the user from this system. The first is the lighting of neopixels, which occurs when a person enters and/or leaves through the doorway. The neopixel turns a different color according to the number of people entering: each distinctly different color corresponds to 5 people being present within the room. For more remotely accessible information, the count of people is displayed on the online platform called freeboard.io. The counted data is first pushed into the online server dweet.io with its speical label via the wifi sheild. Afterwards, by using the socket provided by freeboard.io, the data can be retrieved from dweet.io and the real time data can be accessed on any web terminal.

State Description:

Check door sensor: check whether the door sensor is triggered or not.
Check door flag: check whether the door_flag is high or not, prevent counting the number of open of the door more than it should be.
Check front sensor: check whether front sensor is triggered or not. Look at the condition where front and door sensor is triggered at the same instance.
Check crossing sensor: check whether crossing sensor is triggered or not. If it is on, we would start to check the value previously stored in the buffer.
Check previous buffer and set flag: check the sensor values stored in the buffer for the previous 200 instances. Setup flag based on the value triggered.
Check door sensor again: check whether door is open or not.
Start Tracking: track the value of sensor of incoming 200 time instances. Change the count based on the triggered sensor and according flag.

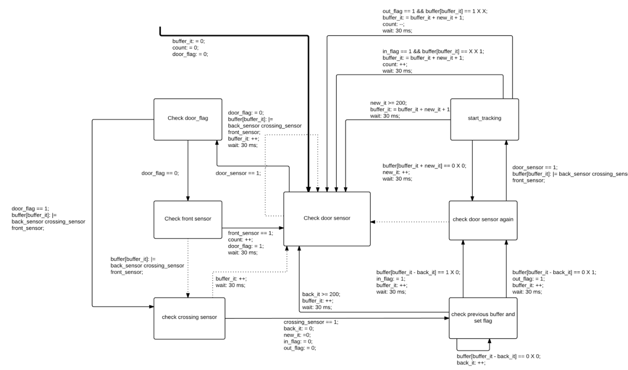
Detection without door

We started to develop our detection algorithm by modeling the doorway as a hole (ignoring the door), and aligning the three ultrasonic sensors. The Boolean value acquired by the sensor is received every 0.2 seconds, and is set to 1 when the sensor detects object within 70 cm, and 0 otherwise. The acquired Boolean

Description of the Finite State Machine and Detection algorithm

Inputs:
door_sensor, front_sensor, crossing_sensor, back_sensor {True, False}
Variables:

Buffer	{int[500]}
buffer_it, back_it, new_it	R
count	R
door_flag, in_flag, out_flag	{True, False}



values are stored in a buffer of size 500, in order to increase the stability of our system and save memory. For the same reason, the bit-encoded information from all three sensors was aggregated into a single digit integer. For example, if the first two sensors were triggered, then the

Figure 4. Finite State Machine Diagram

Boolean representation would be 011, and the integer stored would be 3.

The value is observed when the crossing sensor is triggered, at which point the previous 200 values stored in the buffer are examined. We want to find the most recent instance when the back or front sensor is triggered. Based on the configuration of our system, a flag is used to keep track of whether the front, or back sensor was triggered first. After it is determined which sensor was first triggered, the subsequent 200 values were examined. In those 200 values, if back or front sensor was triggered, and the corresponding flag was set to high, then the count would be added or subtracted appropriately. For instance, if the front sensor was triggered while the flag was up, we would add one to our count, and vice versa for back sensor. The portion of the finite state machine with this functionality is on the right hand side of Figure 4.

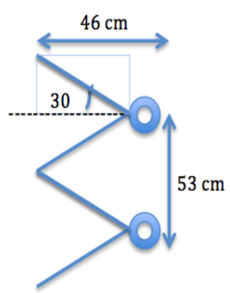
Detection with door

When the door is open, the door can be treated as a hole, like before, and we use the same detection algorithm previously mentioned. A closed door is a different situation, so another sensor was added to check whether the door is open or not. The closed door affects the algorithm when the door is closed and people are trying to come inside the door. For this purpose a front sensor was placed close to the door (10 cm away) to make sure that when the door sensor was triggered, the front sensor would be triggered at the same instance. This is an indication that a person has certainly entered the room. The portion of finite state machine below shows the flow to deal with this situation.

Design Consideration

Sensor Distance

Based on the datasheet of the HC SR-04, the detection angle is 30 degree. Therefore in order to detect people standing at the center of the 92 cm door as well as preventing sensor from interrupting with each other, they should be placed cm away based on the following calculation.



$$distance = 2 * 46 * \tan(30) = 53.11$$

Figure 6. Two sensor laying on the same side with the minimum distance (53 cm) that does not cause interference.

However, the two sensors should not been placed too far away from each other, because then the latency would increase significantly, because more buffers must be searched before a flag is found. This prevents the system from still being considered a real time counter.

Also, the front sensor was placed slightly towards the outside of the room, to ensure that it wouldn't accidentally detect people inside an open door. To compensate for this, center sensor was placed 60 cm away from the front sensor.

Sensor Locations

Two sensors

We started by using only two sensors, as shown below. The algorithm is very simple: a count is added the front sensor is triggered after the center sensor is triggered.

However, this method is not stable. For example, if two people are entering the door at once, then the system will incorrectly consider it as if one person is coming outside. This is illustrated in the table below; the buffer from 124 to 125 shows that the front sensor is triggered twice by the two people, but because the buffer reads the front sensor first, and then the back sensor, this case is considered as one person leaving.

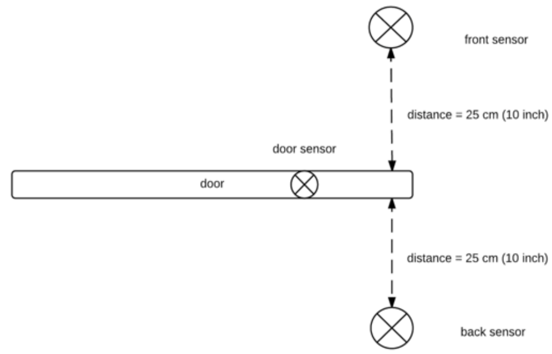


Figure 7. Sensor set up diagram for two sensors design

Table 1. Example of buffer value and program behavior for two people coming in with two sensor design

Buf #	...	121	122	23	124	125	126	127
Buf val	…	NA	ack	ont	front	back	front	…
behavior	…	NA	_flag	dd	Out_flag	subtract	Out_flag	…
count	…	0	0	1	1	0	0	…

Three sensors

To increase the accuracy of the system, another sensor was placed in between the front and back sensor, as shown below.

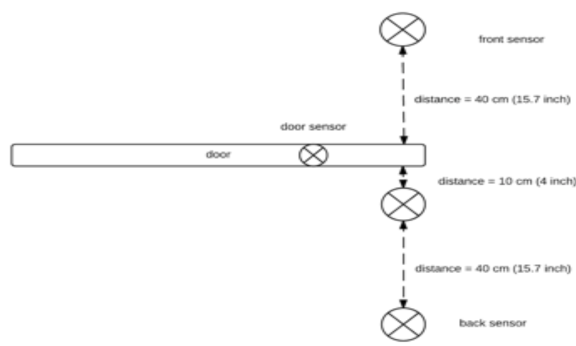


Figure 8. Sensor set up diagram for two sensors design

In this scenario, only the sensor reading directly in front of the center buffer matters. For example, for the scenario where two people are coming inside the door together, as described above, the "In_flag" will be set at buffer 127 based on sensor reading at buffer 126, and thus preventing the error previously encountered, as seen in the table below.

Table 2. Example of buffer value and program behavior for two people coming in with three sensors design

Buf #	...	121	122	123	124	125	126	127	128	...
Buf val	...	NA	back	crossing	front	front	back	crossing	front	...
behavior	...	NA	NA	In_flag	add	NA	NA	In_flag	add	...
count	...	0	0	0	1	1	1	1	2	...

Based on testing, this design accurately detects two people entering 50 cm apart from each other, when they entering the room one following another. However, when we testing the system in the lab, the front sensor was always interfered by the door that is opening towards inside. This is because the front sensor is still being used to detect people crossing this threshold.

Three sensors on one side

In order to allow our system to work in an environment with a door, the sensors were placed outside the door, so that none of the three human detection sensors detect the door anymore. The figure is shown below.

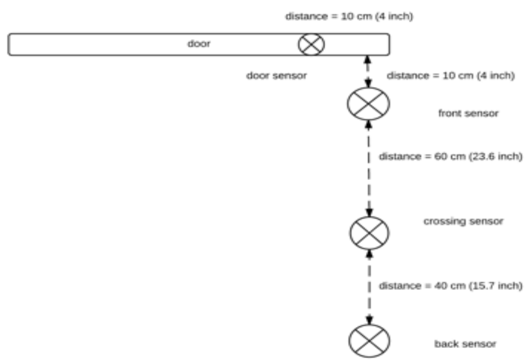


Figure 9. setup for three sensors on one side design

The distance between front sensor and the crossing sensor is increased, since the front sensor is pointing at 10 degrees away from parallel with door, in order to avoiding detect people inside the door when the door is first opened.

Discussion of the performance

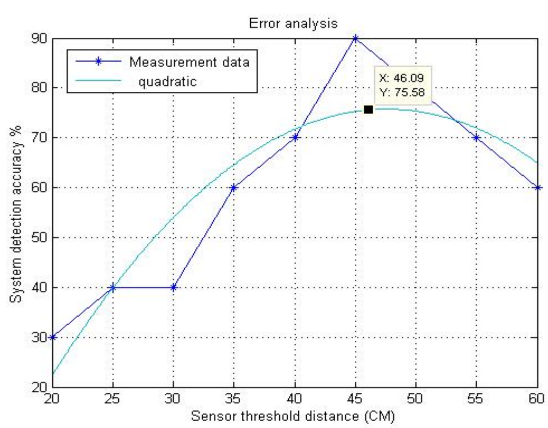


Figure 10. The system accuracy related to detection threshold

The figure above shows the correlation between the sensor detection threshold and the system performance. The test was running under same algorithm, same system set up, same walking test pattern and with 10 tests per case. For the ultrasonic sensors, thresholds were used to detect the presence of a person. Threshold distances that shorter than 40cm and larger than 60 cm both behave poorly. By observing the curve above, the threshold value of 45cm achieves the best results. The inaccuracy is caused by the overlapping detection angles of the ultrasonic sensors and the physical detection requirement such as the assumption that people will enter the doorway in the center (where the sensor is).

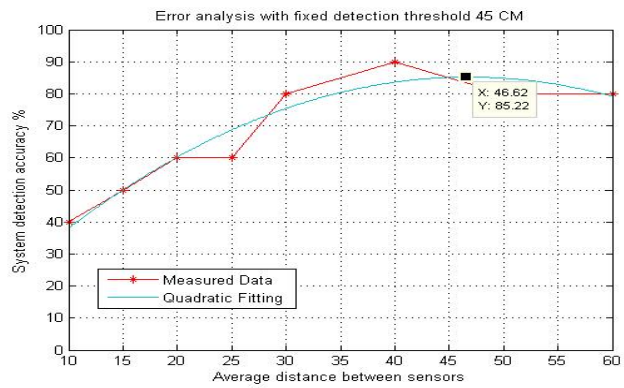


Figure 11. The system accuracy related to average distance between sensors (cm)

The picture above shows the correlation between the sensor placement separation and the system performance. If the separation distance is too small, multiple sensors will be triggered concurrently, causing false readings. If it is too large, then the buffer may clear the first sensor reading before the last sensor is triggered. Larger also makes implementation more difficult. Based on the test result, when placement distance is larger than 40cm, the system accuracy starts to converge to a stable state with relative good performance around 80% accuracy. Due to the limitation of the testing space and the observed performance after calibration of the sensor placement, the sensors were set apart by 40cm and 60cm, respectively to achieve best performance. People cannot follow each other too closely when they are crossing the doorway from the same direction. As shown below, when two people are coming inside a door, the system may incorrectly add 1 or 0 instead of 2.

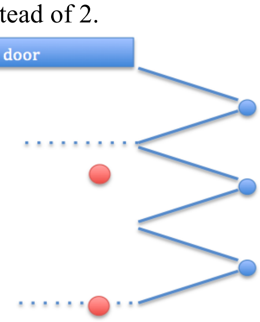


Figure 12. Two people following the sensor too close that cause the incorrect function of the system.

Table 3. Buffer value for incorrect counting when two people are too close

Buf #	...	121	122	123	124	125
Buf val	...	NA	back	crossing	back	front
behavior	...	NA	NA	In_flag	NA	add
count	...	0	0	0	0	1

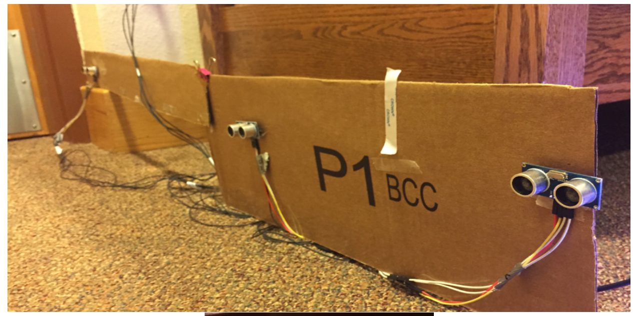
In this case, the second person triggers the back sensor before the first person triggers the front sensor, as shown in buffer #124 above. This has no effect on detecting the first person. However, after the first person triggers the front sensor, in buffer #125, this will be treated as the out_flag when the second person crosses the crossing sensor. Then the system will either count 1 instead of 2, if no one comes close, or will subtracting 1 when the back sensor is triggered again in 2 seconds - both of which are incorrect. To prevent this from happening, another sensor was used with even narrower detection angle, so that the sensors can be placed closer together. This will reduce the frequency of this error, since it would be less likely that the second person triggers the back sensor before the first person finishes crossing. This will also make our system even more compact and easier to implement.

Conclusion

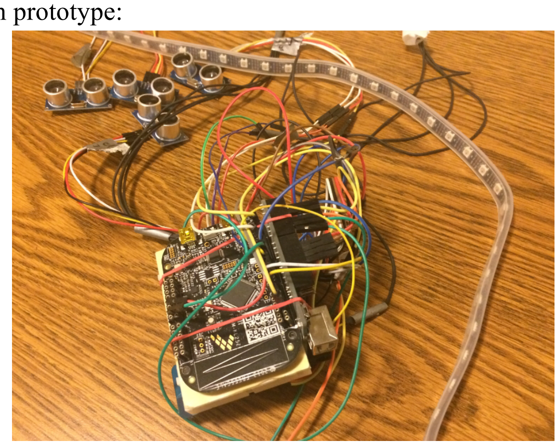
This design accomplishes the objective of detecting people and displaying it usefully. For the future, another advancement might be to improve the accuracy, to have a more accurate count of people in the room/building. Furthermore, the system should be easier to implement and setup, ideally, such that it can scale to the size of an entire building.

Appendix

Sensor placement:



System prototype:



References

1. Ultrasonic Sensor Documentation (HC SR-04 User's Manual) https://docs.google.com/a/berkeley.edu/document/d/1Y-yZnNhMYy7rwhAgyL_pfa39RsB-x2qR4vP8saG73rE/edit

2. CC3000 WIFI module and Freeboard Platform http://www.openhomeautomation.net/internet-of-things-dashboard/