EK210 Final Project Report

Room Occupancy Monitor

James Conlon, Phyliss Darko, Allen Fraiman, Hayden Robinson

Section A6, Group H8 - Professor Roscoe Giles

Spring 2024

Executive Summary

The purpose of this project is to design a device for administrators to monitor and control classroom occupancy, preventing overcrowding for health, safety, and compliance with fire and disease spread regulations. When a room is at the set maximum occupancy, the device deploys a mechanical indicator that discourages people from entering. A 3D printed stop sign attached to a servo motor is used to accomplish the actuation mechanism. When the room reaches maximum capacity, the motor rotates the sign mechanically indicating that the room is not safe to enter. Along with the stop sign, LED's and a buzzer are employed to signify that the room is full. The device is mounted onto an adjustable tripod which offers accessibility features such as location and height control. Attached to the housing are the ultrasonic motion sensors, UI interface, and multiple powering options such as an AC adapter and a switch for 9V battery power. The cost of the final product is $104.22, within the budget of $200. After testing the product, the sensors had a 13% total misread error rate across 3 different walking speeds. A"total" misread defines when a person enters or exits and the sensor does nothing, only creating an error of one. If a person enters and the occupancy decreases, or a person exits and the occupancy increases, (i.e. does the opposite of what it should), this increases the error by two. This is a "fatal" misread and is limited as much as possible.

Introduction

This report provides an overview of the room occupancy monitor system focusing on the major design elements, product description, performance evaluation, and a discussion of key insights. The purpose of this project was to design a device for administrators to monitor and control classroom occupancy, preventing overcrowding for health, safety, and compliance with fire and disease spread regulations. In the starting stages of the design process the team met with the client and outlined a list of key objectives along with their respective metrics, as seen in Table 1. The glass box diagram in Figure 1 focuses on system I/O. Throughout the project, the team kept in mind the priority of the objectives as ranked in Table 2. A morph chart, found in Table 3, was used to explore design options. The goal of this report is to outline how and why the final product fulfilled the set objectives.

Design Elements

To accomplish the product's objectives, several electronic and mechanical components were innovatively employed in the design process. For the mechanical actuation process, a 28BYJ-48 Stepper Motor module was used to rotate a rod to indicate that a room was at full occupancy. Testing was conducted on this motor's step angle, as seen in Figure 2, to ensure a consistent 90 degree turn for the mechanical actuation process. Not only did this motor provide consistent rotations and a long enough shaft to mount a rod onto, it didn't require activation at all times which helps save power when the device is operated solely via battery. See Figure 3 for the complete circuit diagram.

For the sensing process, two HC-SR04 Ultrasonic sensors were mounted in a case which separates them at a 115 degree angle from each other to detect motion. This was done to differentiate entry and exit. The Arduino code works by continuously sensing through both sensors for a “break” of less than 70cm (value can be changed by the client)—see Figure 4 for code snippet and Figure 5 for code flow chart. After a “break,” the code then waits for 2.5 seconds for the alternate sensor to break. This determines directionality of entry/exit. Additionally, multiple timers are implemented to debounce and prevent quick triggers in a short span (limited to about 2.5 seconds between successive entries).

The housing itself is mounted onto an adjustable tripod, fixed to a mounting bracket via adhesive glue. This design option offers portability allowing for the position and height of the monitor to be adjusted depending on the setting, including an indoor or outdoor setting. The design offers two power options to take into account the environment in which the device is being used in. A user has the option to power the device with an AC adapter compatible with an Arduino Uno (7-12V) or a replaceable 9V battery which offers approximately 9 hours of power, as seen in equation (1), if a nearby outlet is unavailable. Lastly, the design offers a user intuitive interface which displays the current occupancy in the room via an LCD display along with two LEDs which indicate if the room can be entered or not.

Final Product Description

The final product consists of a 3D printed housing with a sliding lid mechanism, see Figures 6 and 7, which offers easy access to replace the battery when needed.The housing holds electronic components such as an Arduino Uno, 9V battery, buzzer, and two ultrasonic motion sensors. (Figure 8) Components attached to the housing include a stepper motor and a deployable arm, two LEDs, an LCD screen, a rocker switch, and a tripod. (Figure 9) The product offers various options regarding its power supply, portability, and settings. To change the occupancy limit, the user can connect to the Arduino via computer and change the variable to their desired maximum occupancy. To power the device, the user can choose between using an AC adapter connected to any wall outlet or a replaceable 9V battery.

Lastly, the device can be adjusted to various environments by altering the height of the tripod and positioning the device wherever the user wishes. The code is designed with a realistic default (70cm) for the doorway, but can be modified up to 400 cm. See equation (2) for doorway distance calculation. The tripod setup and code logic means it is equally usable in an outdoor setting with no walls and just a constrained entryway. The user interface includes a live occupancy count on the LCD screen along with two LED’s and a stop sign indicating if the room is enterable. If the room is at maximum occupancy, the device deploys a stop sign along with a red LED and a temporary buzzing sound indicating that the room is not enterable. The cost of the entire product is $104.22, as seen in Table 5. The CAD assembly of the final product can be seen in Figure 10 along with the full list of specifications in Table 4.

Performance Evaluation

The team ran 30 trials at 3 different paces: “Walking, Late to Class, and Boston Marathon.” The trials were run in enter, exit, enter, exit order, so after 30 trials, the cumulative sum should be 0. See Table 6 for the results, plotted in Figure 11. A 13% misread rate was found, but despite this, the data did not deviate too far from the 0 mean that is expected. This is close to the metric of 10% that can be found in Table 1, and is acceptable for normal operation. The code has been rewritten from the prototype to greatly reduce the amount of “fatal” errors. The current algorithm makes this case far less common, instead doing “nothing” on a misread, which is a more acceptable error. This code also works for different paces in a more reasonable way, which has greatly improved the error rate from the prototype. (40% error rate in prototype (Table 7), has been reduced to 13%).

Table 8 reflects the sense operation when multiple people enter at once. It is not realistic to expect more than 2 people to be side-by-side in a doorway, so for groups larger than 2, the 3rd and 4th person are tailing close behind to simulate quick subsequent entries. Some additional notes are included with the table, but the general conclusion is that the debouncing system prevents misreads from sensor noise or rapid triggers, but can mean that rapid people entries are not counted. A 1.5s DEBOUNCE_TIME is in the code for all tests, which limits the error to a reasonable level at normal walking speeds with multiple people groups.

Discussion/Lessons Learned

Trialing different electronic components and designs was crucial during the conceptual design process to achieve the metrics listed in Table 1. This is where the morphological chart in Table 3 came to use. Initially, the team needed to decide on what sensing equipment would work best for the product. After testing infrared break beam sensors and ultrasonic sensors, the team decided that two ultrasonic sensors placed at a certain angle from each other would work best in determining the direction and quantity of traffic flow. The choice of ultrasonic sensors led to many design interactions with the code to get them to work reliably for this purpose. The code was nearly completely rewritten from the prototype after getting feedback about realistic sensing approaches. Figuring out where and how to place the sensors to count occupants accurately was another process of iterative testing. The final configuration was to place the sensors at a 115 degree angle apart in a window of the housing. As for mechanical actuation, the team needed to decide on what motor would work best for rotating the stop sign consistently while being power efficient and easy to mount. After testing various servo and stepper motors, the team decided that a stepper motor, mentioned in the design elements section, fulfilled the necessary requirements for the mechanism. Throughout the project, the team split up in half to focus respectively on mechanical and sensing aspects of the project. Through testing and active communication between the two sub-teams, the final product successfully met the metrics. Through rigorous trialing and iteration, we learned the critical importance of precise component selection and adaptive design processes. Choosing ultrasonic sensors and a stepper motor based on their performance in realistic testing scenarios ensured our device met operational demands efficiently. Our approach of dividing labor between sub-teams focused on mechanical and sensing tasks enhanced collaboration and problem-solving efficacy. This project also highlighted the necessity of integrating feedback and continuously refining our design to achieve a reliable and functional final product.

Appendix

A.1 Figures

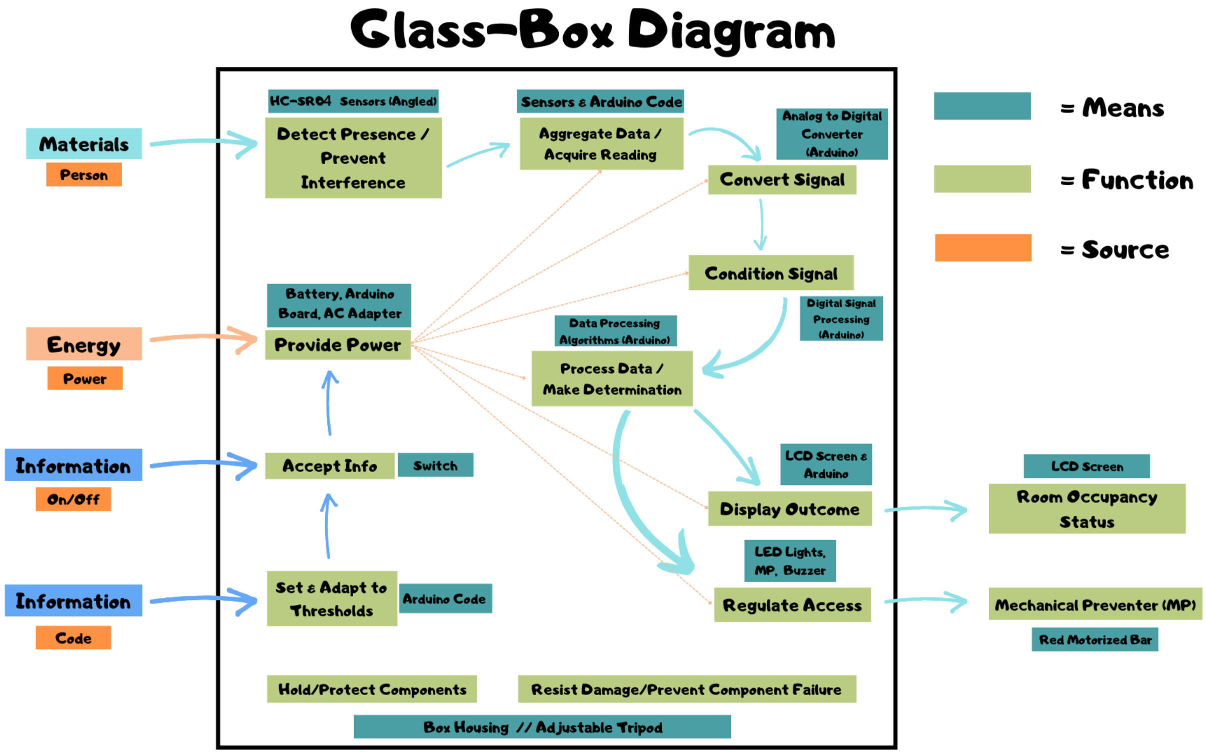
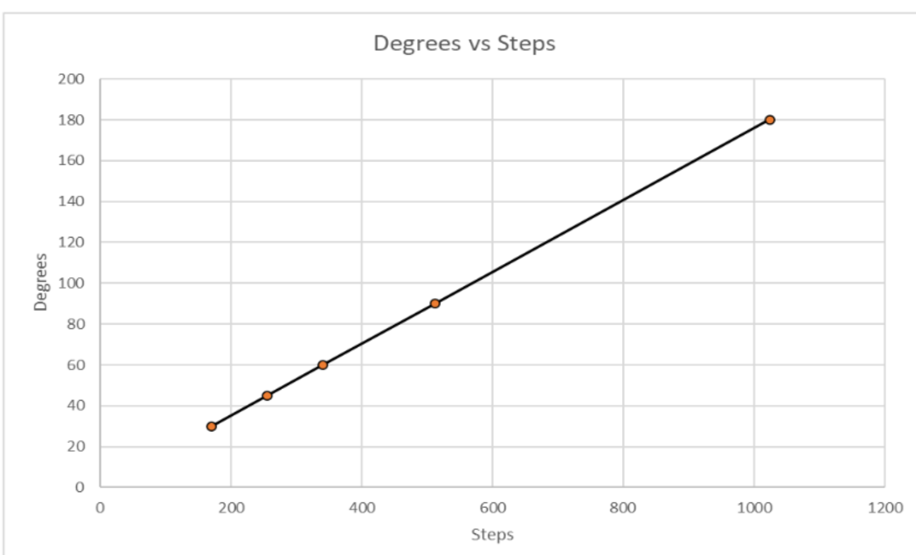


Figure 1: Glass box diagram for the room occupancy monitor. Focuses on inputs into the system, functions, and outputs.



- Step Angle: 11.25°
- Gear Reduction: 64:1
- 360° Turn: 2048 Steps
- 90° Turn: 512 Steps

Figure 2: Testing results of the 28BYJ-48 Stepper Motor, linear relationship between steps and degrees. 512 steps are necessary for a consistent 90 degree rotation.

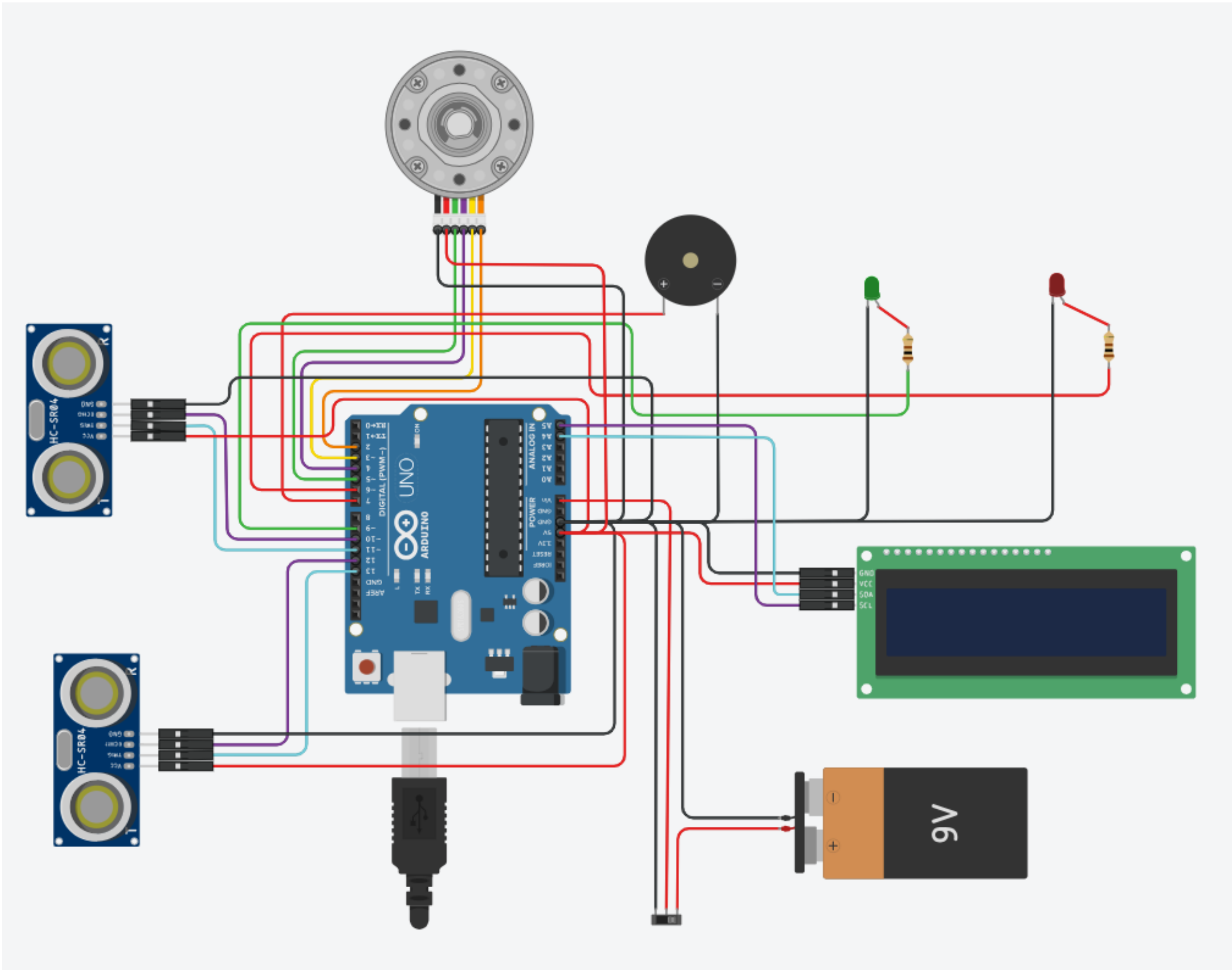


Figure 3: Circuit diagram, components include stepper motor, two ultrasonic sensors, Arduino Uno, 9V battery, rocker switch, LCD Screen, Piezo Buzzer, and two LED's.

```
// ******* CHANGE MAX OCCUPANCY AS NEEDED ******** //
const int MAX_OCCUPANCY = 3;
// ****** CHANGE ENTRY_THRESHOLD AS NEEDED ******* //
long ENTRY_THRESHOLD = 70;        // if person is less than 70cm they are present and the beam is "broken"
// ********************************************** //

// debouncers and delays to change (not recommended)
const int DEBOUNCE_TIME = 1500; // A new entry or exit cannot occur less than 1500 ms apart
const int SENSOR_DELAY = 2250;  // duration to wait for a second sensor to trigger after the first sensor is broken
const int SENSE_LOOP = 40;        // time between sensors (should be a small number to ensure continous sensing)

// code that checks if the distances currently being sensed count as an entry or exit or nothing
void checkEntryOrExit() {
  distance1 = measureDistance(trigPin1, echoPin1);
  distance2 = measureDistance(trigPin2, echoPin2);

  Serial.print("Occ: "); Serial.print(currentOccupancy);
  Serial.print(" Sensor 1 Distance: "); Serial.print(distance1);
  Serial.print(", Sensor 2 Distance: "); Serial.println(distance2);

  if (!isDebouncing) {
        // if sensor 1 is broken and no sensor is active
        if (distance1 < ENTRY_THRESHOLD && !firstSensorActive && !secondSensorActive) {
                firstSensorActive = true;
                firstSensorTime = millis();
                Serial.println("First sensor triggered");
        // if sensor 2 is broken and no other sensor is active
        } else if (distance2 < ENTRY_THRESHOLD && !secondSensorActive && !firstSensorActive) {
                secondSensorActive = true;
                secondSensorTime = millis();
                Serial.println("Second sensor triggered");
        }
  }
  handleSensorEvents();
}

// handles the cases where both sensors have broken OR if nothing happened after one sensor was broken
void handleSensorEvents() {
  if (firstSensorActive && millis() - firstSensorTime < SENSOR_DELAY && distance2 < ENTRY_THRESHOLD) {
        entryDetected();
        firstSensorActive = false;
  }
  if (secondSensorActive && millis() - secondSensorTime < SENSOR_DELAY && distance1 < ENTRY_THRESHOLD) {
        exitDetected();
        secondSensorActive = false;
  }
  if (millis() - firstSensorTime >= SENSOR_DELAY) {
        firstSensorActive = false;
  }
  if (millis() - secondSensorTime >= SENSOR_DELAY) {
        secondSensorActive = false;
  }
}
```

Figure 4: Selected code snippet. This shows both the modifiable constants for the client, some constants that *can* be modified by the client but are not recommended, and a section of the primary sensing algorithm with directionality. Full code: https://github.com/proconlon/RoomOccupancyMonitor

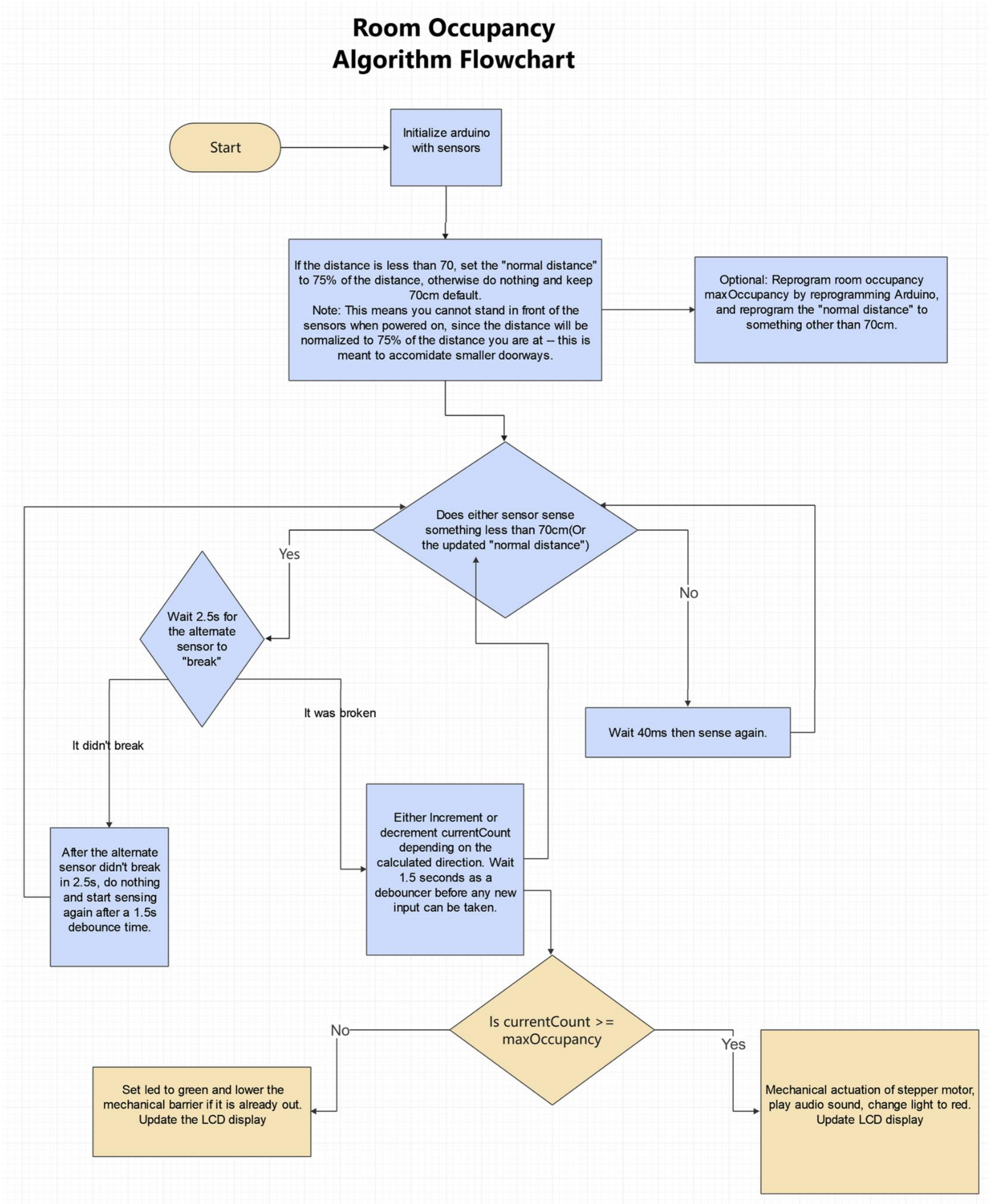


Figure 5: Code flow chart.

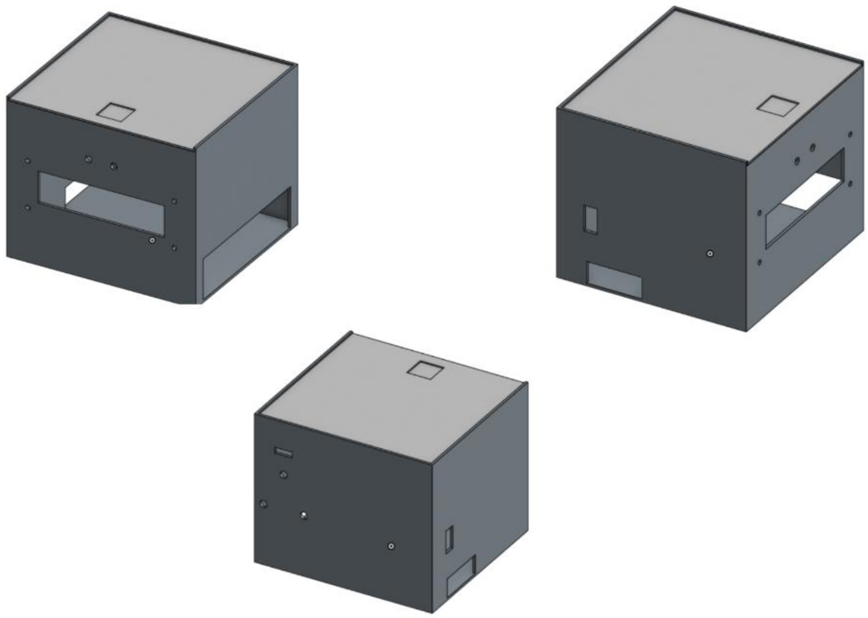


Figure 6: Three isometric views of the housing, holes can be seen for LCD Screen, Sensors, Stepper Motor attachment, rocker switch, and a cutout for AC adapter.

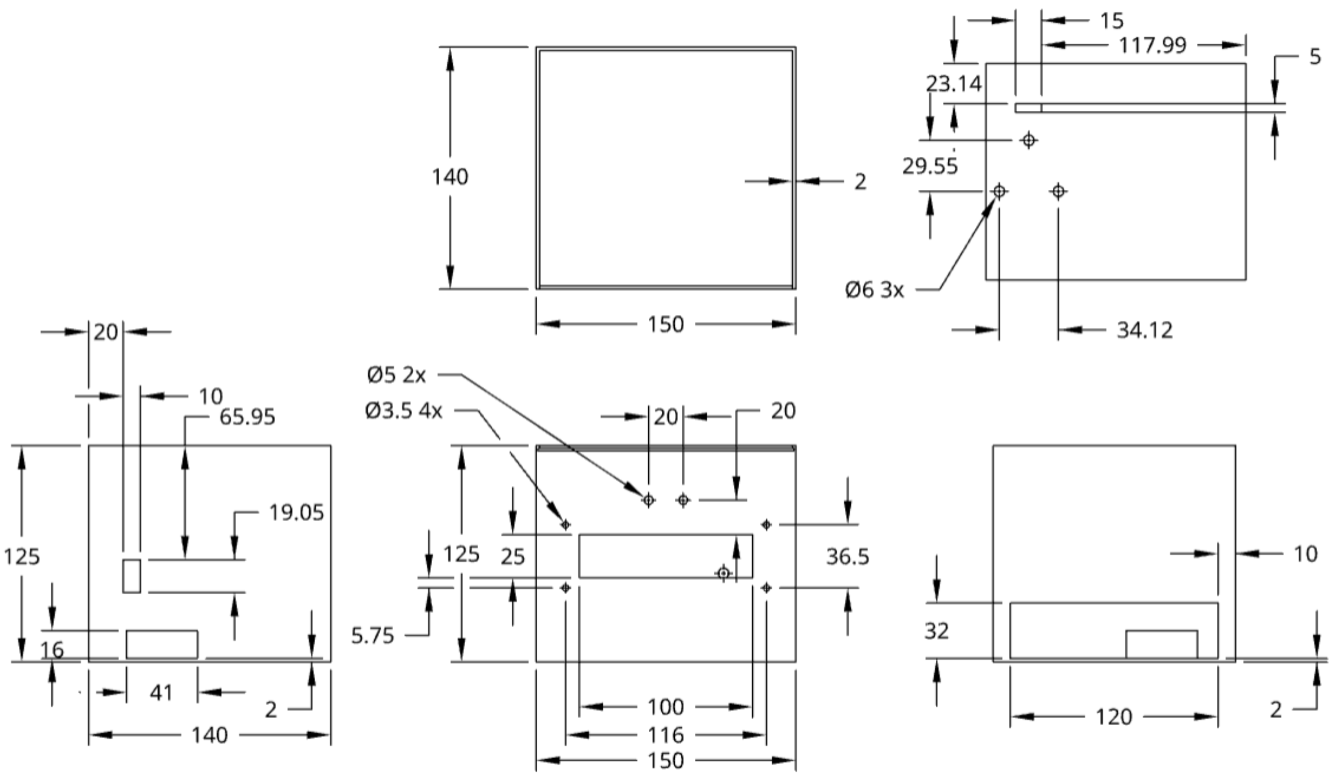


Figure 7: Engineering drawing for the housing, all dimensions are in millimeters. Left is left side, right is right side(sensor side), middle bottom is front, middle top is top view, and top right is back view(motor side).

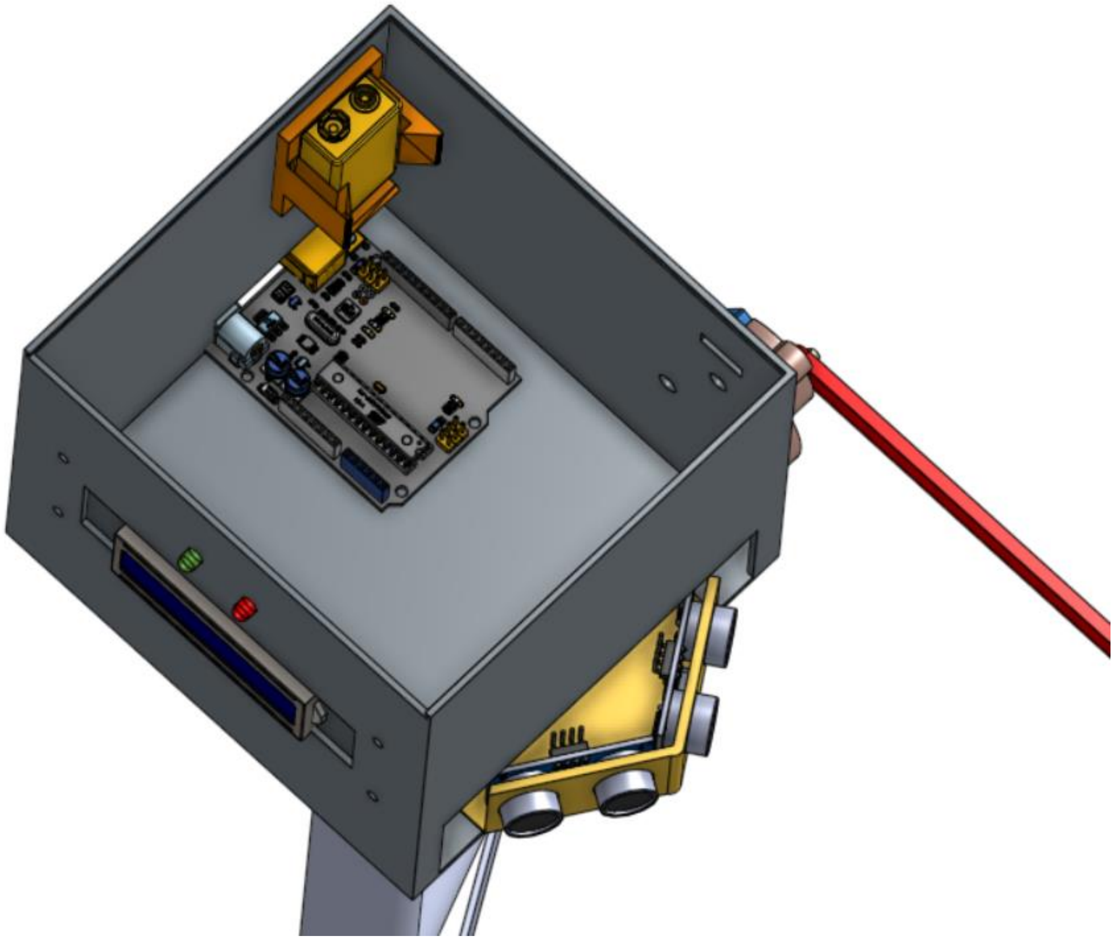


Figure 8: 3D View of components inside the housing. Arduino is attached to the base of the housing via hot glue, the 9V battery can be seen in its case which is hot glued to the housing wall. The two ultrasonic sensors can be seen connected to their case and a plate that gives them support as they sit outside the housing.

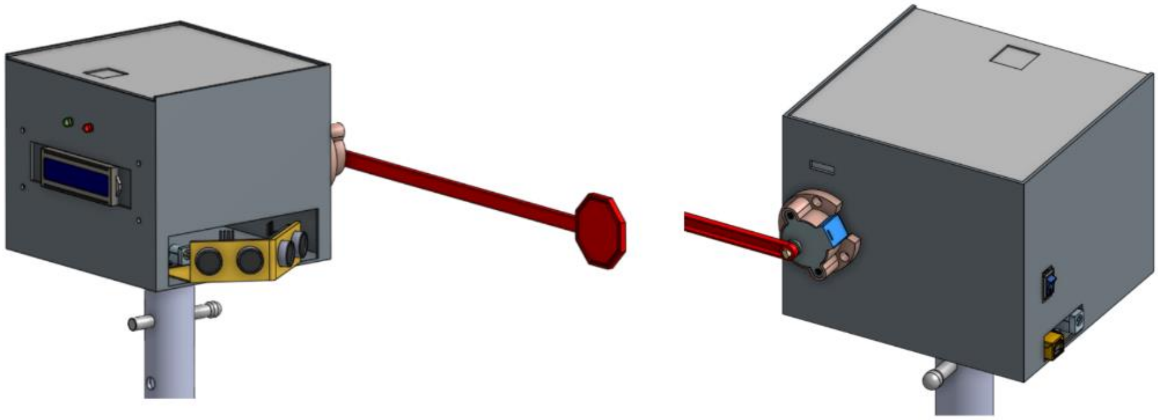


Figure 9: Different views showing how various components are connected to the housing. All holes seen in the housing have fasteners in the real product.

Figure 10: Assembly of the product, housing mounted onto an adjustable tripod.

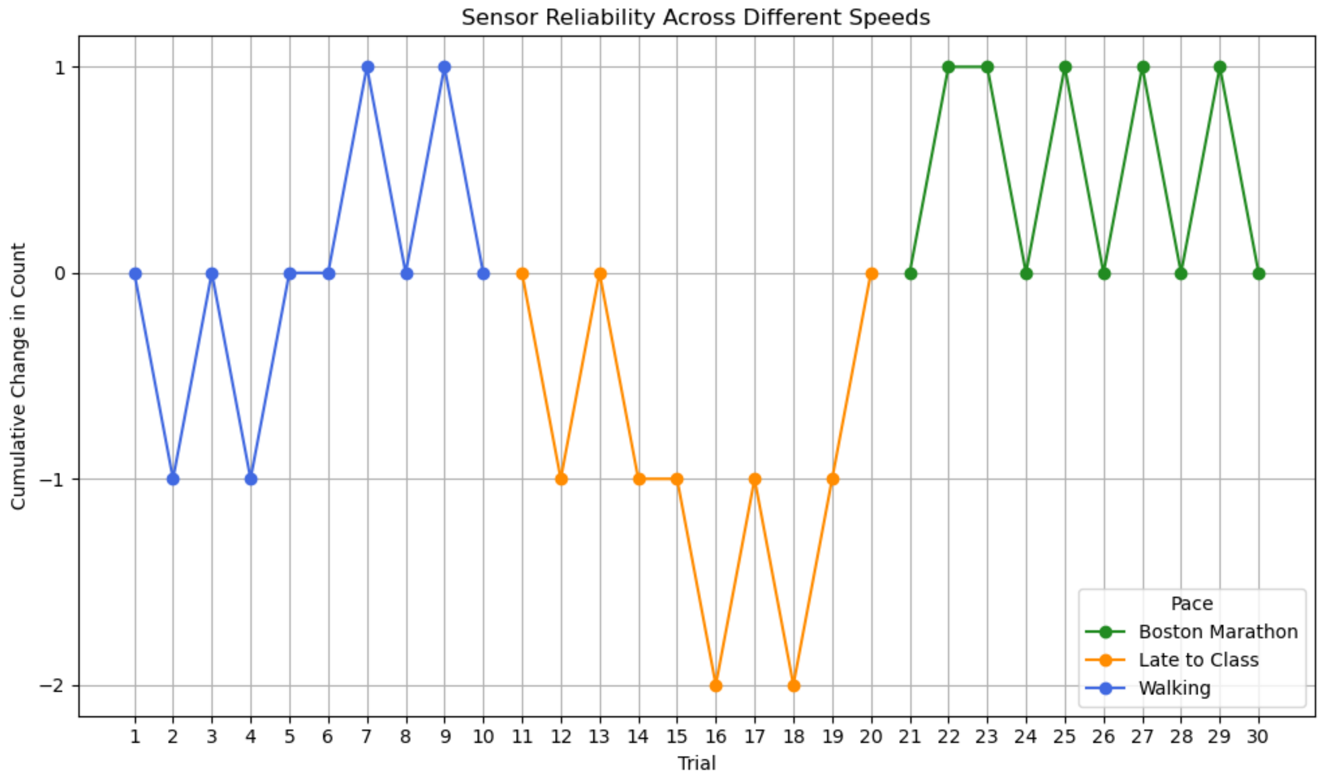


Figure 11: Test data of reliability at different speeds. The data should follow the pattern +1 then -1, +1, -1… in a "zig zag" pattern since our trials were run in Entry, Exit, Entry, Exit order… Thus, the cumulative change in count for each pace should be 0 after 10 trials. Overall, the data shows the zig zag is well maintained, which proves the reliability of the system, with only 13% error rate in our tests.

A.2 Tables

Table 1. Key Objectives vs Metrics/Specifications for room occupancy monitor.

Objectives	**Metrics/Specification**	**Met?**
Count occupants entering & exiting without interference.	- Accuracy rate >90%, less than 1% undercount or overcount.	Close
	- Operational range of 0-1m.	Y
	- Able to process over 50 individuals per day.	Y
Measure occupants precisely	- Counting precision with max deviation of ±1 person.	Y
Complete count quickly	- Time from switch press to system occupancy indication should be less than 5 sec.	Y
	- Should process counts within 1 sec of passing.	Y
Determine occupancy status	- Real time occupancy update with lag time less than 5 sec.	Y
	- Unmistakable indication of occupancy value/count.	Y
Prevents overcrowding (mechanical prevention)	- Mechanical bar deploys within 2 sec after reaching maximum occupancy.	Y
Easy to use/portable	- System setup/takedown time under 5mins. Total weight under 5kg.	Y
Does not obstruct entry/exit	- Door closes and opens freely (even when arm is deployed).	Y
	- No reduction in entry/exit flow rate under normal operation.	Y

Table 2: PCC Chart for ranking objectives. Priority determined by score in the right most column.

	Allows variable occupancy	Doesn’t obstruct exit/entry	No permanent damage	Accurate and reliable	Mechanical prevention	Minimal maintenance/calibration	Score
Allows variable occupancy		1	1	1	1	1	5
Doesn’t obstruct exit/entry	0		1	0	0	1	2
No permanent damage	0	0		0	0	1	1
Accurate and reliable	0	1	1		0	1	3
Mechanical prevention	0	1	1	1		1	4
Minimal maintenance/calibration	0	0	0	0	0		0

Table 3: Morph chart discussing different means for various functions of the device.

	Means	**Means**	**Means**	**Means**
Function	1	2	3	4
Mount Device	Suction Cups	Adhesive Tape	Tripod	Anchor hooks
Power Device	9V Battery	AC Adapter	Solar	—
Sense Traffic	Infrared Sensors	Ultrasonic Sensors	Vibration Sensors	PIR Sensors
Mechanically actuate preventative device	Stepper Motor	Servo Motor	Pulley System	Magnetic System
Alert user	Motorized Stop Sign	LED	Buzzer	LCD Screen Message

Table 4: Specification sheet

Name:	Room Occupancy Monitor
Color	Matte Black
Enclosure dimensions:	150 mm x 125 mm x 140 mm
Tripod height:	0.58 m - 1.9 m
Mechanical bar length:	284 mm
Battery life:	Unlimited with AC power supply, approximately 9 hours with a 9V battery.
Power cable length:	1.83 m
Maximum sensing range:	- 4 m (theoretical) - Tested max doorway width up to 1.33 m across - Default range is 0.93 m across (used in test cases)
Pace:	Irrelevant (for normal walking speeds), not useful in race/high speed environment
Timing:	Cannot count people faster than 1.5s apart. (Default, is easily modifiable in code)

Table 5: List of parts with respective quantity and unit price. Total cost listed at the bottom.

Part	Quantity	Unit Price
28BYJ-48 Stepper Motor with ULN2003 driver	1	$2.96
HC-SR04 Ultrasonic Sensor	2	$2.00
5mm LED	2	$0.60
16x2 I2C LCD Display	1	$8.86
Rocker Switch	1	$0.67
Arduino Uno	1	$27.60
Adjustable Tripod	1	$31.99
Piezo Buzzer	1	$0.55
Black PLA 1kg Spool	1	$24.99
9V Battery	1	$2.00
Total Cost: $104.22		

Table 6: Final product test cases. Total failure rate = 13% – fatal error rate 3%. Plotted in Fig 11.

Trial	Direction	Result	Pace	Result
1	Entry	Increment	Walking	Success
2	Exit	Decrement	Walking	Success
3	Entry	Increment	Walking	Success
4	Exit	Decrement	Walking	Success
5	Entry	Increment	Walking	Success
6	Exit	Nothing	Walking	Fail
7	Entry	Increment	Walking	Success
8	Exit	Decrement	Walking	Success
9	Entry	Increment	Walking	Success
10	Exit	Decrement	Walking	Success
11	Entry	Increment	Late to Class	Success
12	Exit	Decrement	Late to Class	Success
13	Entry	Increment	Late to Class	Success
14	Exit	Decrement	Late to Class	Success
15	Entry	Increment	Late to Class	Success
16	Exit	Decrement	Late to Class	Success
17	Entry	Increment	Late to Class	Success
18	Exit	Decrement	Late to Class	Success
19	Entry	Increment	Late to Class	Success
20	Exit	Increment	Late to Class	Fail
21	Entry	Increment	Boston Marathon	Success
22	Exit	Increment	Boston Marathon	Fail
23	Entry	Nothing	Boston Marathon	Fail
24	Exit	Decrement	Boston Marathon	Success
25	Entry	Increment	Boston Marathon	Success
26	Exit	Decrement	Boston Marathon	Success
27	Entry	Increment	Boston Marathon	Success
28	Exit	Decrement	Boston Marathon	Success
29	Entry	Increment	Boston Marathon	Success
30	Exit	Decrement	Boston Marathon	Success

Table 7: Protype Test cases. With 40% total error rate – 13% fatal error rate

Trial	Direction	Result	Pace
1	Entry	Increment	Walking
2	Exit	Decrement	Walking
3	Entry	Increment	Walking
4	Exit	Decrement	Walking
5	Entry	Decrement	Walking
6	Exit	Increment	Walking
7	Entry	Nothing	Walking
8	Exit	Decrement	Walking
9	Entry	Increment	Walking
10	Exit	Decrement	Walking
11	Entry	Increment	Late to Class
12	Exit	Nothing	Late to Class
13	Entry	Nothing	Late to Class
14	Exit	Decrement	Late to Class
15	Entry	Increment	Late to Class
16	Exit	Increment	Late to Class
17	Entry	Increment	Late to Class
18	Exit	Decrement	Late to Class
19	Entry	Increment	Late to Class
20	Exit	Decrement	Late to Class
21	Entry	Nothing	Boston Marathon
22	Exit	Decrement	Boston Marathon
23	Entry	Nothing	Boston Marathon
24	Exit	Decrement	Boston Marathon
25	Entry	Increment	Boston Marathon
26	Exit	Decrement	Boston Marathon
27	Entry	Nothing	Boston Marathon
28	Exit	Nothing	Boston Marathon
29	Entry	Decrement	Boston Marathon
30	Exit	Nothing	Boston Marathon

Table 8: Test cases for multiple people entering at once. Tested for 2, 3, and 4 people, where the first two people walked in side-by-side as a pair. Aimed to be within ±1 for the error with groups of people. Our assumption is that no more than two people can enter side-by-side at once, and under that assumption, the data is within the error.

Number of people coming in	Actual recorded count	Notes
2	1	Two people entering at once only records one person. This is a limitation of the sensors and code, so it is working as expected.
2	1	
2	1	
3	1	For a third person entering right behind two people, they may be counted depending on how far behind they are. If they are too quick, then they will not be counted. The code is designed so that the count is unlikely to increment when it should decrement, and vice versa. In ambiguous cases like this, it should do nothing.
3	3	
3	3	
4	4	Same concept as groups of 3 applies here.
4	4	
4	3	

A.3 Equations

$$Battery\ Life = \frac{Battery\ Capacity(mAh)}{Load\ Current(mA)} = \frac{400\ mAh}{43.4\ mA} \approx 9\ hours \qquad (1)$$

$$\text{ENTRY\_THRESHOLD} = \text{doorway size} * 0.75 \qquad (2)$$

(2) Calculation for doorway width, modifiable by client. The default ENTRY_THRESHOLD is 70cm, which corresponds to the average doorway width of 93 cm = 3 ft doorway. The theoretical limit of the sensor is 400cm, however, this will likely cause extra interference. Our code is only tested up to 100 cm in an outdoor space without any walls - which corresponds to a “doorway” about 4.3 ft across.

References

ChatGPT Used for higher level code debugging.

Conlon J., RoomOccupancyMonitor, GitHub. (n.d.). https://github.com/proconlon/RoomOccupancyMonitor.

Dejan, Ultrasonic sensor HC-SR04 and Arduino - Complete Guide, How To Mechatronics. (2022). https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/.

HC-SR04 Ultrasonic Sensor Module User Guide. (n.d.). https://www.handsontec.com/dataspecs/HC-SR04-Ultrasonic.pdf.

In-depth: Control 28BYJ-48 stepper motor with ULN2003 Driver & Arduino, Last Minute Engineers. (2023). https://lastminuteengineers.com/28byj48-stepper-motor-arduino-tutorial/.