University of Notre Dame

Electrical Engineering: Senior Design

Home Security System I - Final Report

Group:
- Adéniyi Emiabata
- Amaris Vanegas
- Ana Rivera
- Lucas Carrit Delgado Pinheiro
- Thomas Antonio Silva Gonzaga

Professor:
Dr. Michael Schafer

May 8, 2023

Contents

1 Introduction

In 2020, more than 7 million property crimes were committed in the US, with burglary being the second most common crime. According to a study conducted by the Insurance Information Institute, the average cost of property damage caused by burglaries in the US is $2,661. Surprisingly, research shows that poor neighborhoods are more likely to be targets of burglary compared to richer neighborhoods, despite the belief that thieves would target rich households due to the likelihood of high-value goods being present. The reasons for this trend are numerous. Economic segregation plays a significant role in this trend as wealthier neighborhoods are often located at a physical distance from less affluent areas, which are where burglars tend to originate from due to poverty being linked to criminal behavior. Affluent neighborhoods also have more strong and assertive community members who are more willing to intervene if they notice suspicious behavior, such as having neighborhood watch systems and Home Owners Associations, causing burglars to avoid them. Poor neighborhoods show reduced community responsibility, which makes individual homes more vulnerable since neighbors are less likely to address a possible criminal situation. Socially, there is a perceived vulnerability of disadvantaged neighborhoods that is supported by statistics, as home security systems (HSS) are almost three times more present in households with incomes above $100K than among lower-income households, making lower-income households easier targets for break-ins and other crimes.

In addition to the high prevalence of burglaries in poor neighborhoods, house fires pose another significant threat to disadvantaged households. According to the National Fire Protection Association (NFPA), structural fires occur in approximately 358,000 homes in the United States every year, and poverty and fire risk are closely intertwined. Crowded living conditions, outdated hazardous appliances, and older housing structures are just a few examples of the conditions that exacerbate the likelihood of house fires in poor neighborhoods. While poverty does not directly cause fires, it is evident that specific conditions create added difficulties that contribute to the incidence of fires. Furthermore, low-income households are less likely to invest in fire protection insurance, given its high cost. These two issues combined leave impoverished communities with limited resources to recover from the consequences of burglary or house fires, as they cannot afford costly insurances, home security systems, or safer electronic devices.

Our proposed home security system is an ESP32-based product that aims to provide homeowners with affordable protection against various dangers,

such as contact, motion, smoke, flame, and vibration. By using inexpensive yet durable sensors, we want to ensure that even low-income households can afford to protect themselves and their property. By providing a cost-effective and reliable security solution, residents of disadvantaged neighborhoods could gain greater control over their safety and security. This, in turn, could help to reduce the perception of these neighborhoods as easy targets for burglars and other criminals.

The system uses an RF tree network with ESP-NOW technology to transmit data from the sensors to a central base called a home pod. This data is then sent to a database that can be accessed by users through a website or mobile app. Our user-friendly interface allows users to easily monitor alerts and view potential safety threats to their home. Compared to current HSS products, which tend to prioritize aesthetics over durability and are priced above $300, our system is much cheaper, with the potential to cost around $100-$150 with a streamlined manufacturing process. By making affordable and reliable home security accessible to lower-income households, our solution could significantly reduce the risks of burglaries and house fires in disadvantaged communities.

Ultimately, the design was successful. As discussed in more detail in the subsequent sections of the report, all the sensors met their respective requirements. Within the RF network subsystem, each node was able to interpret the data collected by the respective sensors and communicate immediately with the home pod using ESP-NOW protocol. In terms of the communications subsystem, the main requirements were also met. The home pod continuously sent data to the Back4App database, while the website retrieved the data and displayed it to the user, sending warning emails when appropriate. Further, the ESP32 can connect to any WiFi network, allowing for the implementation of the product in any household. There were a few limitations in both subsystems in designing this project. In the communications subsystem, it would be better if, instead of sending emails to the user, the design sent an automated phone call or notified the appropriate authorities regarding a breach and relayed the address of the place where it occurred. Since the project was made for demonstration purposes, it would not be feasible to attain a phone number exclusively for testing or to report incidents to the police or fire department. Further, a mobile app would be a better GUI since the user would have access to his home information with just their cellphone. However, that was not possible since no member of the group had previous experience in developing mobile apps.

2 Detailed System Requirements

2.1 Basic System Requirements

1. The home security system shall consist of four sensors to ensure a variety of physical inputs are detected. The four sensors being a vibration sensor, contact sensor, gas sensor, and motion sensor.

2. The four sensors shall be capable of direct communication with a main device (home pod).

3. The home security system shall include a user interface to communicate any concerning activity detected by the sensors to the user through a website.

4. The user interface shall include an email system capable of sending real-time email messages to users in the event of a breach detected by the sensors.

2.2 Basic Sensor Requirements

1. All sensors shall be compatible with the ESP32 microcontroller in use to ensure uniformity and communication with the home pod.

2. The gas sensor shall detect multiple gases to account for various types of hazards.

3. The gas sensor shall provide real-time values in parts-per-million (ppm) to keep users updated of impending danger.

4. The gas sensor shall detect gases of at least 2000 ppm as this is the value at which sustained exposure (up to 1 hour) to smoke becomes a direct death threat.

5. The vibration sensor shall provide instantaneous acceleration values in the event of a break-in.

6. The vibration sensor shall withstand major force from potential intruders and maintain its position on the window.

7. The motion sensor shall provide instantaneous values based on proximity.

8. The motion sensor shall detect motion up to at least 5 metres away with at least a 90 degree spread to inform users of potential break-ins before intruders make contact with the doors/windows.

9. The contact sensor shall provide real-time data based on opening or closing of doors to alert users of breaches.

10. The contact sensor shall withstand major force from potential intruders and maintain its position on the door

2.3 RF Subsystem Requirements

1. All sensor devices shall send real-time data to the home pod immediately upon detection to enable swift notifications to the user.

2. The home pod shall receive data from all sensors simultaneously to prevent delays in notifications to the user.

3. The home pod shall use unique identifiers to distinguish between messages from the various in order to ensure the correct data is displayed on the website.

2.4 Communications Subsystem Requirements

1. The communication subsystem should receive messages from the RF subsystem in real time.

2. The subsystem should interpret the disturbances picked up by the RF subsystem correctly.

3. The website must be fully functional with customizable house setup, sign-up, log-in, and log-out options, in addition to safety features.

4. The website must be able to update its display based off the information collected from the Back4App database using data from the receiver ESP32

5. The users must receive a warning email whenever a sensor gets triggered

6. The ESP32 must be able to connect to any WiFi network and not just local ones

7. The website must be able to render conditionally, displaying different information to the user based on how many/which sensors are installed in the house

2.5 Printed Circuit Board Requirements

1. Versatile PCB design that allows for the integration of different sensors
2. Different sensor circuits can be tested depending on surface mount devices soldered onto the board
3. Multiple power schemes for each sensor, ensuring appropriate voltages were fed to each one
4. Header pins included for UART interface programming to facilitate programming
5. Microcontroller antenna accounted for to ensure no signal is blocked by ground plain
6. Holes included for mounting device onto plastic chassis

2.6 Chassis Requirements

1. Must be sturdy and compact enough to allow easy installation and placement.
2. The material from which the chassis is made from must be lightweight but strong enough so that it is resistant to bending or cracking but easy to move.
3. Casing should have the battery holder and board securely set in place.
4. The chassis must protect electronic components from environmental factors.
5. Screw holes must hold top and bottom portions of the cases together.

3 Detailed Project Description

3.1 System Theory of Operation

The home security system consists of a radio frequency (RF) network subsystem and communications subsystem. It is through these subsystems

that sensor readings are collected and transmitted to the website which notifies users of any breaches detected. The RF network subsystem consists of sensors, their respective microcontrollers, and the central microcontroller belonging to the home pod. The sensors are transmitter devices as they send data collected to the central microcontroller. The central microcontroller is a transmission device that sends the data received to a backend database.

Information collected by the sensors is sent to the microcontrollers at all times. To prevent any interception and unauthorized access of data being sent to the transmitter device all data is encrypted in the network using ESP-NOW software. To ensure the main board receives data from specific ESP-32 devices in the network, the devices must be paired to one another. The sensors are able to send data to the central microcontroller by knowing its MAC address. The central microcontroller differentiates data from the various sensors by interpreting their unique identifiers (Figures 3.3.2-3.3.4). Finally, the main board sends the data collected to the website using Wi-Fi.

The communications subsystem relays information collected by the sensors to the user through a graphical user interface which, for this home security system, is a website. The website receives the information collected from a database that stores information in structured classes. The structured classes allows the user to know when any suspicious activity is picked up by the sensors, which sensor is picking up the activity and when the issue is resolved. This information is displayed on the website in real time. In addition, users receive emails when breaches are detected. Conditional rendering was implemented in the subsystem to allow users to add additional sensors to their home security system. Users must create password-protected accounts to access the website displaying their home security system.

3.2 System Block diagram

The home security system is composed of two subsystems which are the RF network and communications subsystems. The system block diagram of the home security system is shown in Figure 3.2.1.

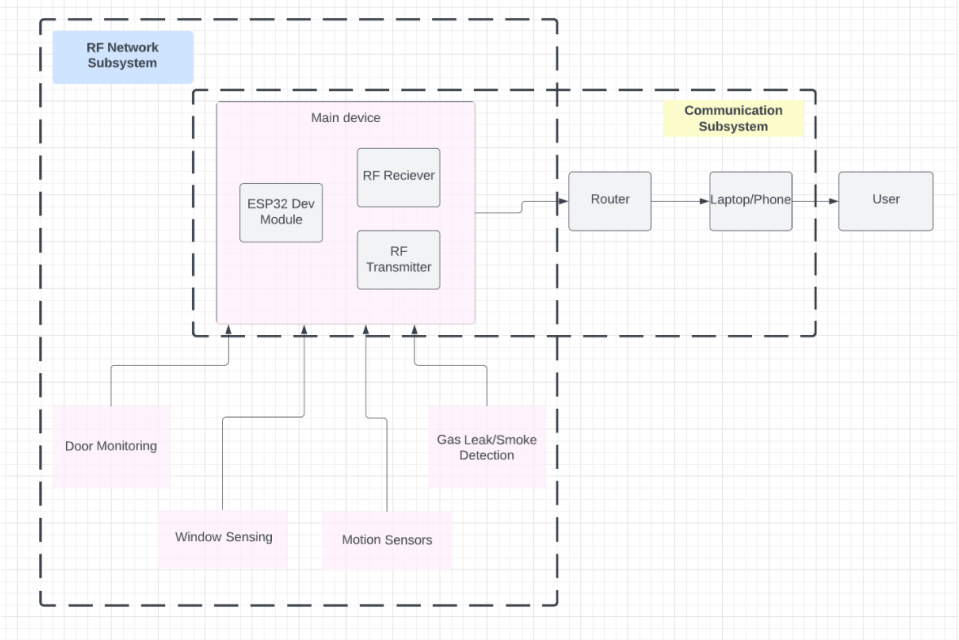

Figure 3.2.1: System Block Diagram

As seen in figure 3.2.1, the primary subsystem in the home security system is the RF network. All sensor readings are collected and sent through the RF network to a backend database which makes up the communications subsystem. The communications subsystem interprets the readings collected and displays them on a website for client easy access. Each electronic component in the home security system is housed by a chassis.

3.3 Detailed Operation of the RF Subsystem

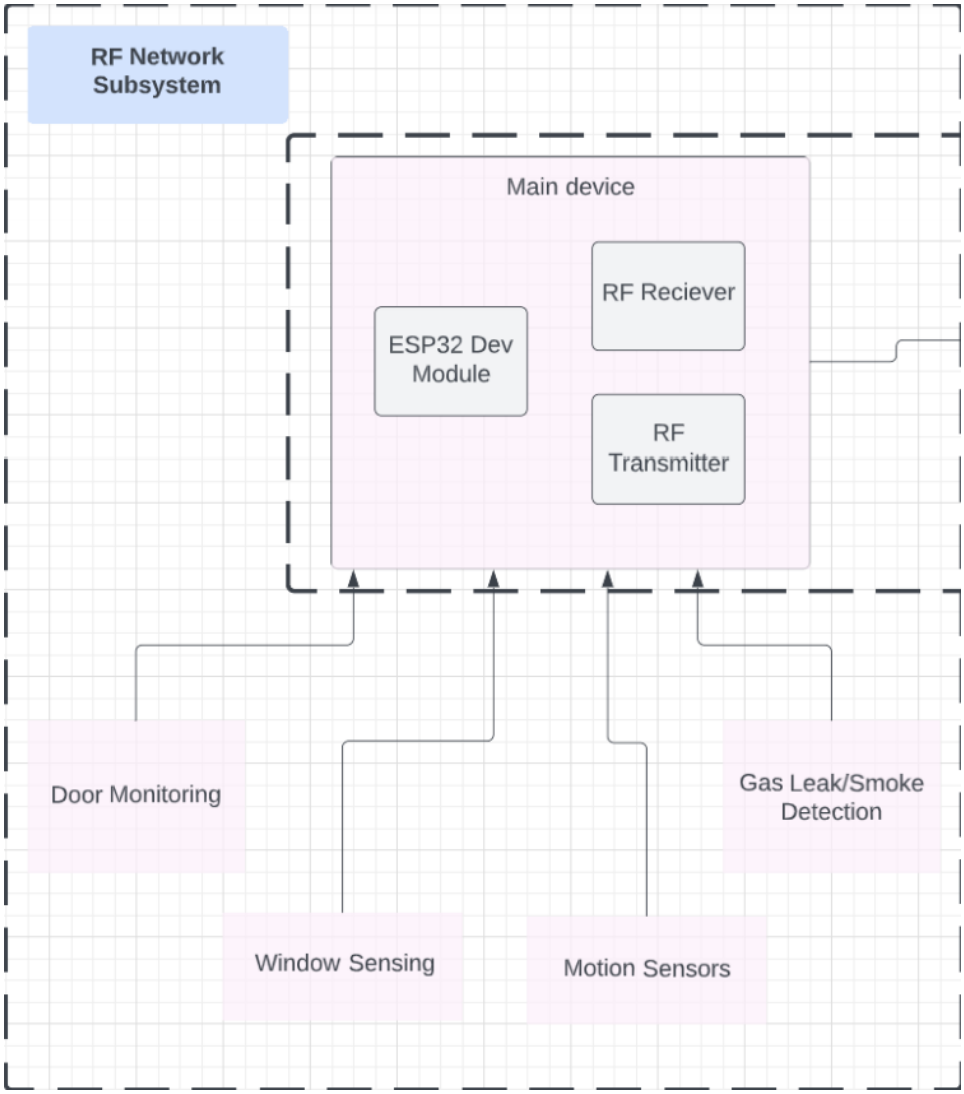

Figure 3.3.1: RF Communication Subsystem

The purpose of the RF Network Subsystem is to facilitate the transfer of data from the sensors to the main device (home pod) before it is then transferred to the database for display on the website. The major components of

this subsystem were the contact sensor, vibration sensor, gas sensor, motion sensor, the main device (home pod), and the ESP-NOW communication protocol.

3.3.1 Contact Sensor

The contact sensor used was the SM351LT magnetoresistive sensor. The SM351LT sensor offers high sensitivity (7 - 11 Gauss) and omnipolarity. Most importantly, it costs 2.25 and has nanopower requirements, falling in line with the cost-saving goal of this project. Per its magnetoresistive mode of operation, when the sensor is either on the same plane or perpendicular to a magnetic field (Fig. 3.3.2) it allows electricity to flow. The contact sensors are to be placed on doors that open up to the exterior of the house, in order to notify users of break-ins.

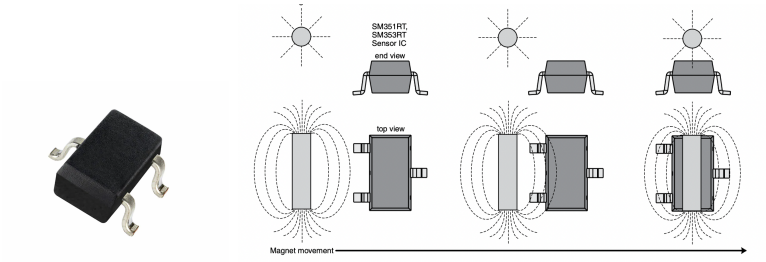

Figure 3.3.2: SM351LT magnetoresistive sensor

The contact sensor produces two values, 0 and 4095. 0 is the output when the door is closed, meaning the sensor is in contact with the magnet placed on the door. 4095 is the output when this magnetic contact no longer exists, signifying that the door has bee opened. The ESP32 is programmed to receive this analog data and forward the information for the Home pod which will notify the user if a door is opened via email. The code excerpt on Figure 3.3.3 shows how the GPIO pin that is wired to the output pin of the SM341LT is defined, then data is read from the pin and loaded into a data structure that will be sent to the home pod device receiver.

```
// Define the pin that the sensor output is connected to
#define sensorPin 32

//In setup
pinMode(sensorPin, INPUT);
//In loop
// Read the sensor data from the analog input pin
int sensorValue = analogRead(sensorPin);
// Set values to send
myData.id = 3;
myData.x = sensorValue;
myData.y = 48; // User identifier
```

Figure 3.3.3: Analog contact sensor data flow

3.3.2 Gas Sensor

The gas sensor used was the MQ2 Gas sensor. This Gas sensor is capable of detecting multiple types of gases. Specifically, it can detect LPG, smoke, methane, butane, and propane. This sensor has a fast response time, a sensitivity that can be adjusted with a potentiometer and only costs 7.6, making it an ideal sensor for gas detection in any home. The MQ2 gas sensor has a wide range of gas detection capabilities. It can detect 200-500 ppm of LPG, 200-10000 ppm of smoke, 500-20000 ppm of methane, 3000-5000 ppm of butane and 200-5000 ppm of propane.

Figure 3.3.4: Gas Sensor

In this specific home security system the gas sensor is used for smoke detection purposes only. This gas sensor produces values ranging from 200-10000 ppm when any smoke is detected. The higher the number produced the closer the smoke is to the sensor. When a value above 1000ppm is output the client is immediately notified of the danger on the website and through email notification. Any values below 1000ppm indicate that there is no imminent danger.

```
#define GAS_ANALOG 32
#define GAS_DIGITAL 2
int Threshold = 2300;

//In setup
pinMode(GAS_DIGITAL, INPUT);

//In loop
int gassensorAnalog = analogRead(GAS_ANALOG);

 // Set values to end
myData.id = 1; // sensor identifier
myData.x = gassensorAnalog;
myData.y = 23;
```

Figure 3.3.5: Gas sensor data flow

As shown in the code excerpt (Fig. 3.3.5), the microcontroller is programmed to take in analog information from the sensor using using its GPIO

pin 32 and uses a threshold variable to determine when to start the notification protocol.

3.3.3 Vibration Sensor

The vibration sensor used was the LIS3DHTR accelerometer. The LIS3DH is a low-power three-axis accelerometer with I^2C/SPI serial interface standard output. This accelerometer provides outputs at rates ranging from 1 Hz to 5.3 kHz while also possessing the ability to select scales of ±2g/±4g/±8g/±16g. Most importantly, the price is 2.25 which allows us to maintain our goal of providing low-cost options. The accelerometer possesses a wide range of functions, from free-fall detection to acting as a pedometer.

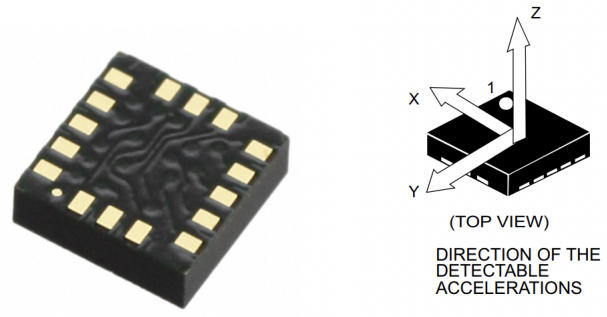

Figure 3.3.6: Accelerometer

For the purpose of this home security system, the accelerometer is placed on a window and used to determine the acceleration of the window based on how much it is being moved/shook. The accelerometer produces a three-axis (x,y,z) output (Fig. 3.3.6) of acceleration values which are then processed into a single value:

$$\sqrt{x^2 + y^2 + z^2}$$

This value is then compared to $10m/s^2$. $10m/s^2$ was chosen as the threshold based on experiments to determine when vibration became potentially dangerous. For the website, the output for the user is either a 0 or 1 (See Fig. 3.3.8). If the window's acceleration is greater than $10m/s^2$, the output is a 1, and the user is then notified of a potential breach via email. On the other hand, if the window's acceleration is less than $10m/s^2$, the output is a 0, meaning there is no imminent danger.

```
// I2C Initialized
Adafruit_LIS3DH lis;
#define I2C_SDA 21
#define I2C_SCL 22

//In setup -> data is gatheredd and stored in lis

//In loop
  lis.read();
  sensors_event_t event;
  lis.getEvent(&event);

// xyz data processed into one value -> holder

  myData.x = holder;  // Holder has value 0 or 1 to show vibration
```

Figure 3.3.7: Accelerometer data flow

Figure 3.3.7 shows how the data is interpreted by the ESP32 using the I2C interface and after processing the three dimensional data (Fig. 3.3.8). This flow is possible the use of an object 'lis' that belongs to a custom class that provides ways to interface with the LIS3DH sensor to read its data and configure its settings. Using 'lis', the sensor's range and data rate can be setup (Figure 3.3.9). The G in the range setup refers to gravity and by setting the range to 16G the accelerometer is set to collect acceleration data up to 16 times the acceleration due to gravity. The rate setup had several options ranging from 1 Hz to 400 Hz, but for the purposes of window vibration sensing a rate of 50 Hz was appropriate.

```
if ( (sqrt(pow(event.acceleration.x, 2) + pow(event.acceleration.y, 2)
+ pow(event.acceleration.z, 2))) > 10 ){
  holder = 1;
}
else{
  holder = 0;
}
```

Figure 3.3.8: Three-axis data processing

```
// Sensor initialized
  bool status = lis.begin(0x18);
  if (!status) {
    Serial.println("Not working");
    while(1);
  }
//Calibration
  lis.setRange(LIS3DH_RANGE_16_G);   // 2, 4, 8 or 16 G!
  Serial.print("Range = "); Serial.print(2 << lis.getRange());
  Serial.println("G");
  lis.setDataRate(LIS3DH_DATARATE_50_HZ);
  Serial.print("Data rate set to: ");
  switch (lis.getDataRate()) {
    case LIS3DH_DATARATE_1_HZ: Serial.println("1 Hz"); break;
    case LIS3DH_DATARATE_10_HZ: Serial.println("10 Hz"); break;
    case LIS3DH_DATARATE_25_HZ: Serial.println("25 Hz"); break;
    case LIS3DH_DATARATE_50_HZ: Serial.println("50 Hz"); break;
    case LIS3DH_DATARATE_100_HZ: Serial.println("100 Hz"); break;
    case LIS3DH_DATARATE_200_HZ: Serial.println("200 Hz"); break;
    case LIS3DH_DATARATE_400_HZ: Serial.println("400 Hz"); break;
    case LIS3DH_DATARATE_POWERDOWN: Serial.println("Powered Down"); break;
    case LIS3DH_DATARATE_LOWPOWER_5KHZ: Serial.println("5 Khz Low Power"); break;
    case LIS3DH_DATARATE_LOWPOWER_1K6HZ: Serial.println("16 Khz Low Power"); break;
  }
```

Figure 3.3.9: Accelerometer setup

3.3.4 Motion Sensor

The motion sensor used was the BS6123 mini Passive InfraRed (PIR) sensor. It was determined that the this PIR sensor would be the best to use in the home security system since it has the ability to detect motion up to 8 meters away, has a 120 degree spread, is much more compact than other

PIR sensors and runs on 3.3 V of voltage. In addition, this sensor only costs 1.95 so it is relatively cheap compared to other PIR sensors. It is also very quick to trigger and has a wide range of motion detection.

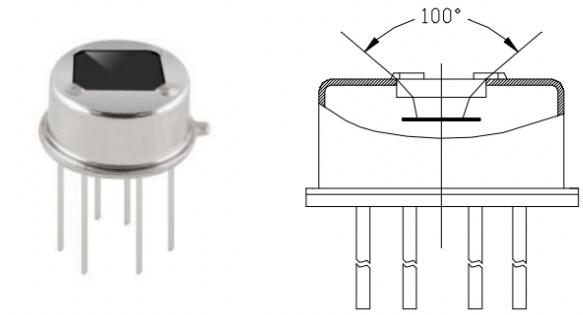

Figure 3.3.10: PIR Sensor

This sensor produces either the value of 0 or 1 when activated. When motion is detected the sensor produces a 1 in real-time otherwise the 0 is produced. When a value of 1 is output the client is immediately notified of the unusual activity detected on the website and through an email notification.

```
#define PIR_SENSOR_PIN 27
//In Setup
pinMode(PIR_SENSOR_PIN, INPUT);

//In loop
int sensorValue = digitalRead(PIR_SENSOR_PIN);
// Set values to send
myData.id = 2;
myData.x = sensorValue;
myData.y = 56; // unique identifier
```

Figure 3.3.11: PIR data flow

3.3.5 Communication Protocol

The main communication protocol used in this subsystem is ESP-NOW. ESP-NOW is a protocol commonly used for fast, low-power, and secure communication between ESP32 devices. ESP-NOW offers a sizable payload of 250MB, which is more than enough for sending sensor data. The communication is a peer-to-peer model, requiring a sender and receiver. The MAC address of the receiver ESP32 is also used as a unique identifier to ensure the data is being sent to the right ESP32 device. Additionally, ESP-NOW also offers delivery feedback on data being sent with messages, such as "Delivery Fail" or "Delivery Success" There are three modes available: one-to-one, one-to-many, and many-to-one. For this system, the many-to-one configuration was most suitable. This is based on the fact that the four microcontrollers used on the sensors (vibration, contact, gas, motion) all need to simultaneously communicate with the principal node which was a single ESP32 board.

```
// REPLACE WITH YOUR RECEIVER MAC Address
uint8_t broadcastAddress[] = {0x8C, 0x4B, 0x14, 0x9E, 0xF4, 0x24};
```

Figure 3.3.12: ESP-NOW MAC Address Example Call

Figure 3.3.12 shows the first step in using ESP-NOW, which is to insert the MAC address of the receiver ESP32. This was done in the code for all four sensors.

```
typedef struct struct_message {
    int id; //sensor type identifier ex. PIR id = 2
    int x;  // will contain sensor value
    int y;  //device unique identifier
} struct_message;
```

Figure 3.3.13: Message Structure

Figure 3.3.13 displays another step in implementing ESP-NOW, which involves creating a structure for the data to be sent. Our selection was to use

simple integer variables to reduce payload and allow for easy manipulation when being transferred to the user database before going to the website. id - an identifier for each sensor (for example, 4 - accelerometer), x - the value being sent, and y - an identifier for each system. All sensors in a house will have the same y value.

```
// Set values to send
myData.id = 2;
myData.x = sensorValue;
myData.y = 23; // unique identifier
```

Figure 3.3.14: Message Structure Example

Above in Figure 3.3.14 is an example of the message structure for the PIR Motion Sensor.

```
// callback when data is sent
void OnDataSent(const uint8_t *mac_addr, esp_now_send_status_t status) {
Serial.print("\r\nLast Packet Send Status:\t");
Serial.println(status == ESP_NOW_SEND_SUCCESS ? "Delivery Success" : "Delivery Fail");
}
```

Figure 3.3.15: ESP-NOW Delivery Status Message

Figure 3.3.15 shows a callback function executed whenever data was sent to provide a status update. When sensor data was output, it was accompanied with either “Delivery Fail” or “Delivery Success” to show whether the message was received by the receiver ESP32. “Delivery Success” allowed for the message to then proceed into the communications subsystem. “Delivery Fail” could either mean the wrong MAC address was used or that the main device (with the receiver ESP32) was turned off.

3.4 Detailed Description of Chassis Design

To ensure the sensors would be protected from environmental elements and could be placed in any household multiple chassis where designed. When designing the chassis it was determined the dimensions of each case would

be the same or similar for a cohesive and clean look. The vibration, motion, gas and contact sensor cases all had the same dimensions of 170 mm in length and 100 mm in width. Similarly the casing for the magnet used had the same dimensions. The home pod on the other hand was designed to be slightly larger in length compared to the rest of the cases since it's board is the only one not powered by batteries but instead is powered by a power outlet. The home pod uses a USB cable instead which is why it was designed to be longer. Its dimensions are 185 mm in length and 100 mm in width. The bottom case design is shown in the figure below.

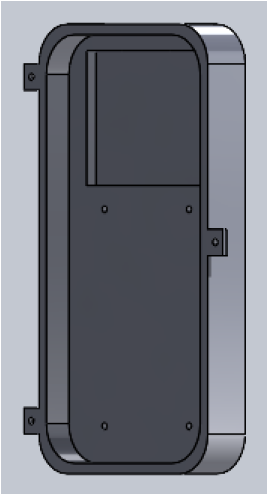

Figure 3.4.1: Design For Bottom Board Case

Figure 3.4.1 shows primary features for all board holding bottom cases. One of the first things to note is that the case has an indent for battery holder placement, 4 screw holes to screw the board in place and 3 lips to screw the top and bottom cases together. The outer shell of the case was 3 mm thick for all. This was to ensure the cases wouldn't crack or break of they where to be dropped, stepped on, etc... The only case which didn't contain screw holes for its board was the one for the gas sensor. This case instead had holes for spacers to elevate the gas sensor and its board to prevent gas sensor damage.

The extrusion of the bottom portion of each case was the same, 30 mm, for all sensors except the contact sensor. The extrusion for this board was made slightly smaller due to the limitations set by the sensor. Since the contact sensor has to pick up magnetic waves from a magnet the extrusion

was only 15 mm. This allowed for the sensor and its board to be close enough to the case housing the magnet for easy interaction. The second case, the one housing the magnet, was made much thicker and shaped differently which can be seen in the figure shown below.

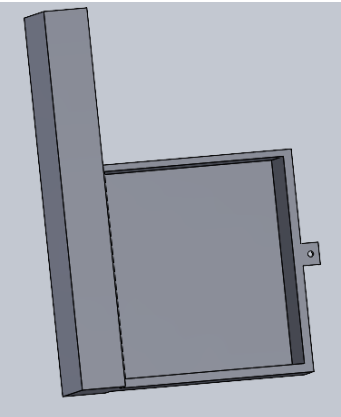

Figure 3.4.2: Design For Magnetic Case

Figure 3.4.2 shows the case used for housing the magnets. This case was made thick to ensure there would be enough room for different types and different amounts of magnets to fit in the extruded pocket in the chance the magnets used need to be replaced. The extrusion for this portion of the magnetic case was 25 mm. Only the bottom third of this case has the extrusion since the magnets need to be placed near the magnetic sensor so both the top and bottom cases lie flush against each other when the door is closed and so the magnetic waves are sensed at that point by the sensor. This was also done so the top half of the bottom case holding the batteries wouldn't crash with the top case. This was done to ensure nothing would prevent both cases from touching.

The top portion of each case was extruded similarly for all sensors and home pod. The vibration, motion and home pod where all extruded by 15 mm. This was done so all sensors would fit snugly inside each case and for proper protection of electronic components. The basic design of the top cases is shown below.

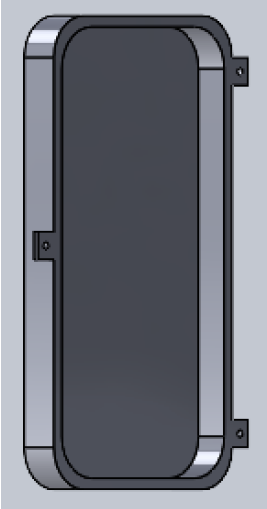

Figure 3.4.3: Design For Top Board Case

Figure 3.4.3 shows how each top case contained lips for easy closing and connection with the bottom cases. Each top case had a unique element in its design so each sensor would work efficiently. The gas sensor casing had tiny holes throughout its bottom half so any smoke near by will be detected quickly. The motion sensor top case had a small opening that allowed the sensor to stick out. This allowed the sensors 120 degree spread to be used completely. The vibration sensor top case had a very slim thickness of 1.5 mm across its top. This was so that there would be no issues with the sensor picking up vibrations on the glass it is located on. The top case of the magnetic sensor was even thinner. It measured to be about 1.5 mm in thickness to allow easy electromagnetic wave readings.

The top portion of the gas sensor case was extruded slightly higher than that of the rest because the gas sensor was the largest one used. This case was extruded by 20 mm. The casing component of the contact sensor differed the most in dimensions. The second component, the portion housing the magnet, was extruded by 35 mm. This was done so that there would be enough room for the first component of the case, the portion holding the sensor, and the top component to have a proper interface.

Lastly, the home pod and gas bottom cases differed slightly from the rest of the cases. The bottom case for the gas sensor has holes along its side for easy smoke detection. The bottom home pod case has a hole on its side from which the USB cable used to power its board sticks out of. This is for easy connection to a power outlet. All chassis fit the sensors, battery

holders and circuit boards properly.

3.5 Detailed Operation of the Communications Subsystem

The main purpose of the communications subsystem is to deliver the data to the user. The main component of this subsystem is a website hosted on the domain http://www.safehousehss.com/ through Amazon Lightsail. The website was designed using the React frame on JavaScript combined with CSS files for the styling of the pages. The home page of the website contains a logo and buttons for users to log in and sign up, as shown on Figure 3.5.1.

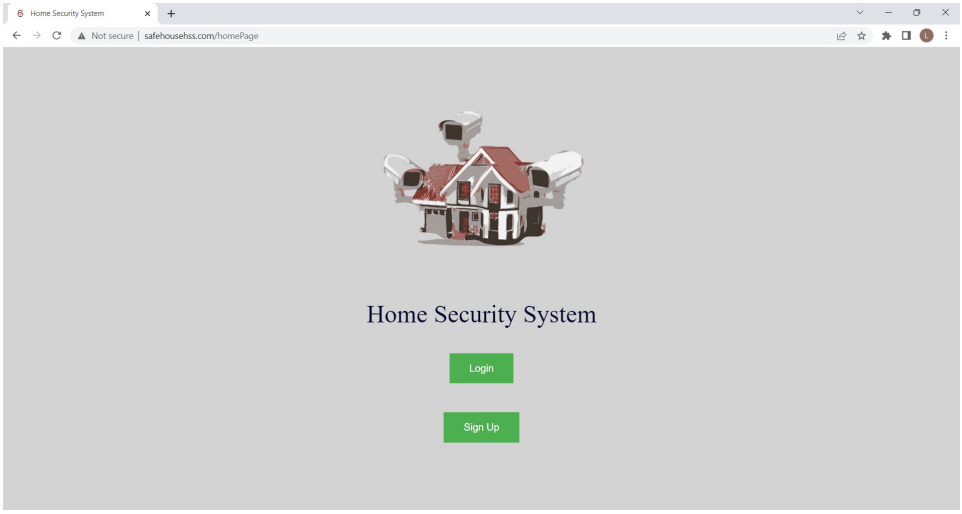

Figure 3.5.1: Home page of the website

The sign up form asks the user to enter the first name, the last name, email, password, and device code, which is a unique identifier for each home security system. The user would be sent this code upon purchasing the product. The email and the password are used to log in to the website. It is important to highlight that the log in and sign up system has protected routes in React to ensure the safety of the users' data. Without this protection, it would be possible to access private data simply by typing the URL of the restricted area on a browser. However, the protected routes redirect the website users to the log in page if they attempt to access restricted pages using this method.

After logging in, users are redirected to a restricted area with the readings for each of the sensors. This page has a welcome message, a log out button, and a table with three columns. The first column of the table corresponds to the type of sensor, the second one indicates whether there is a security breach on that sensor, and the third column displays the numeric reading for the sensor. The website also displays an additional warning if at least one of the sensors indicate a security breach. The columns of the table are rendered conditionally depending on which sensors the user has. Figure 3.5.2 shows the layout of the website for a user that has one sensor of each type.

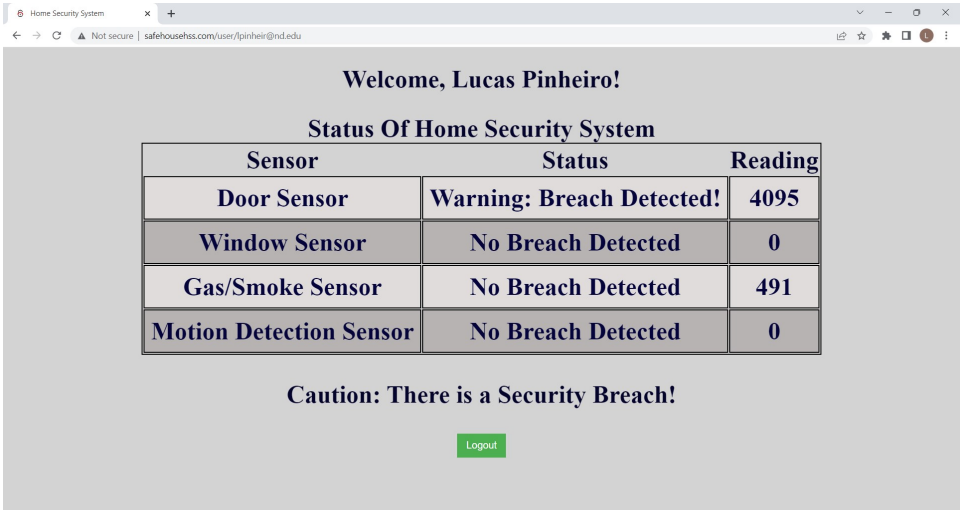

Figure 3.5.2: Restricted area of the website

This page is periodically refreshed to update the live readings from the sensors.

Another important aspect of the communications subsystem is the email alerts system. The purpose of this system is to notify the user about any security breaches as quickly as possible. Differently from the website, the email alerts system is supposed to be run locally by the home security company. Since this system has access to private user data and even has the functionality of sending automatic emails to the users, it is safer to run it in a local server. The alert system reads data from the database and emails the users alerting about any security breaches in any of the sensors. In order to prevent an excessive amount of emails to be sent in case of a continu-

ous breach, the system will only send an alert at least 15 minutes after the previous email was sent. Figure 3.5.3 shows the automatic message sent to alert users about a security breach.

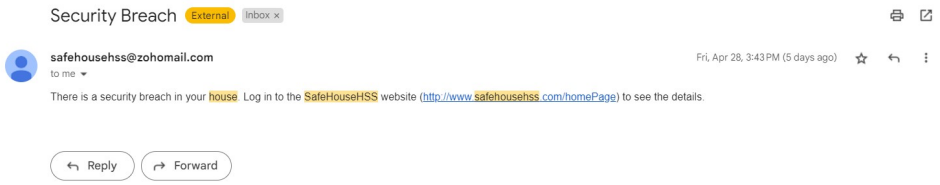

Figure 3.5.3: Email that alerts users for security breaches

3.6 Interface

The two subsystems interface through a database. All the data from the sensors is sent to the receiver, which uploads it to the database. The website retrieves the data and displays it in the restricted area for users. The database is stored on the backend Back4App and consists of two main classes: the "User" class and the "Readings class". The structure of the database is summarized on Figure 3.6.1.

Figure 3.6.1: Diagram of the database

When the user creates an account on the website, a unique entry is created in the "User" class. This entry contains some personal data such as the first name, last name, email, and the unique identifier for the user's system. Entries on the "Readings" class are created when a receiver first sends data to the database and contains fields to identify the type of device, its reading, and the identifier for the system. After the receiver sends data for the first time, it will start replacing previous entries whenever it sends new data. It is important to notice that both classes have the "deviceCode" field, which is what connects the classes.

The receiver interacts exclusively with the "Readings" class by periodically sending data from each sensor. The website, on the other hand, reads data from both classes. When a user accesses the restricted area, the website compares the value of the "deviceCode" field for the entry corresponding to the logged in user with the values in the entries in the "Readings" class and displays all readings for entries that have a matching device code.

4 System Integration Testing

4.1 Testing Procedures

In order to verify that the system was functioning properly, all sensors were assigned the same device code, and an account was created on the website using the same code. It is important to highlight that each code corresponds to a specific house and it is hard-coded in the program uploaded to each sensors, so assigning the same code to each sensor means that the variables that contained the device code in the program uploaded to each sensor was set to exactly the same string, which was then also used in the sign up form to create an account on the website.

After the basic setup was complete, the first step of the testing was to verify that the website displayed readings for each of the sensors, which was indeed the case. All sensors were then individually submitted to the trigger conditions. The readings and warning messages displayed on the restricted area of the website were always in accordance with the expected values. Since the website displayed the expected readings for all sensors, both subsystems, as well as the integration between them, were functioning properly.

The last step in the system testing was to verify if the alert mechanism was working. In order to do so, the program that sends the email alerts, which is separate from the main website due to security reasons, was ran on a local server. After the program was running, each sensor was triggered, and the alert emails were indeed sent to the user.

Another important part of the alert mechanism testing was to verify if the user would only get an email at least 15 minutes after the last alert was sent. In order to test this, the sensors were turned off, and the readings in the database were manually set to values that would trigger the alert system. After the server with the alert mechanism ran for a prolonged period, it was possible to verify that the email notifications were only sent every 15 minutes.

4.2 Results

The proposition of assigning each sensor to a unique device code was ultimately successful, as it allowed the design to be implemented in different households simultaneously. As an experiment, different accounts were created, assigned a specific device code, and then also linked to different sensors. For example, it is possible to set up an “Account A” having 23 as

its device code and associated with four different sensors, and at the same time an "Account B" with a device code of 5 linked with two sensors. Because they have different device codes, they will only update the database for the appropriate homeowner. In addition to the device code, a second identification number is assigned within the sensor code, allowing for the implementation of several sensors within a household.

Next, the logic implemented to determine whether a breach has been detected also worked properly. Whenever the reading for a given sensor went above or below the established threshold, a warning message was displayed on the website as seen in Figure 3.5.2. Because the website is automatically refreshed at every five seconds, the user is always able to visualize the most updated information regarding the sensors installed in the house.

Finally, the emails were sent to the correct user whenever a sensor was triggered. The 15-minute restriction was a positive addition to avoid saturating the inbox of the user whenever a breach is detected. For example, the gas sensor is triggered whenever the ppm value surpasses 1000 ppm. However, the gas sensor could read that value continuously for multiple minutes. This would result in a substantial amount of emails being sent in a short amount of time regarding the same breach in case the correct logic had not been implemented.

5 Installation manual

5.1 How to Install the Product

All sensors come in pods that can be attached to doors, walls, and windows using Velcro strips, which are provided with the product. The sensors are not water proof so they cannot come in contact with any liquids. The specific instructions for each of the sensors are as follows:

- **Receiver**: The receiver is the only circuit component in the home security system not powered by batteries. Since it needs to be plugged into a power outlet, the only requirement for the location of this pod is to be near an outlet inside the home.

- **Gas sensor**: This sensor could be placed anywhere in the house. Ideally it would be best to place it in the kitchen which would be the most likely place in the house for a fire or a gas leak to happen.

- **Window sensor**: This sensor must be placed on a window. Specifically on the side that faces the interior of the house.

- **Contact sensor**: The contact sensor is made up of two components. The component containing the sensor itself must be placed on the exterior wall right next to a door. The second component, which consists of the magnet, must be placed on the edge of the same door in a way that it is right next to the pod when the door is closed.

- **Motion sensor**: this sensor could be located anywhere inside or outside the house for motion detection.

5.2 How to Setup the Product

5.2.1 Hardware

The first step to set up the system is to power it. All sensors are powered by three AA batteries each. The battery holder is inside the pod, so it is necessary to unscrew it to place the batteries.

The receiver pod is the only one that is powered in a different way. Instead of using batteries, the receiver must be plugged to a power outlet. There is a USB cord sticking out of the receiver pod. This cord must be attached to a charger and placed on a power outlet.

After the system is powered, the user needs to set the WiFi connection of the receiver. If the receiver does not identify any known WiFi networks, it will set an access point named "ESP32_AP", which will be displayed along with the WiFi networks on a laptop or a mobile device. The user must connect to this access point. Mobile devices will normally display an option to automatically redirect the user to the access point page on a browser. However, it is also possible to simply paste the URL "http://192.168.4.1" on a browser to configure the receiver's access point. There will be a "Connect" button on the page. After pressing this button, the user simply has to enter the name of the WiFi network of the house and its password, which allows the receiver to connect to it. After the WiFi credentials are sent to the receiver for the first time, there is no need to do this again at any point, unless the WiFi password changes or the user starts using a different WiFi network.

Figure 5.2.1 shows the main page of the access point generated by the receiver, and Figure 5.2.2 lists the WiFi networks found by the receiver.

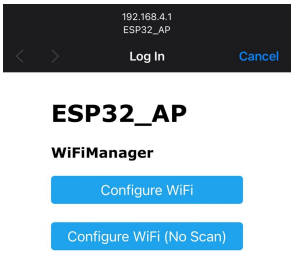

Figure 5.2.1: Main page of the access point generated by the receiver

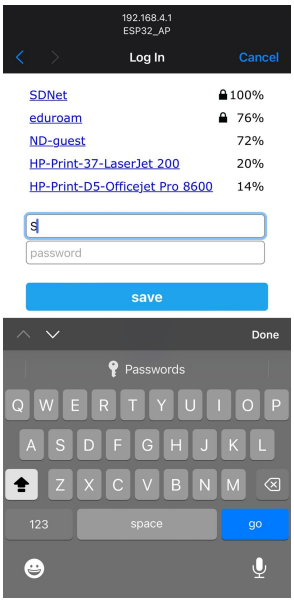

Figure 5.2.2: List of WiFi networks found by the receiver

5.2.2 Software

Besides setting up the sensors in the house, the user also needs to create an account on the website (http://www.safehousehss.com/) by clicking the "Sign Up" button. The user will be prompted to fill a form with the first name, last name, email, password and device code. It is important to remember that upon purchasing the product, each user would receive a unique code that identifies the system. This is the code that must be used when creating an account.

5.3 How the User Can Tell if the Product is Working

In order to tell if the product is working, users can log in to the website and check if the main page displays each of the sensors in the system and their respective readings. If the system is functioning correctly, the data displayed on the website should periodically update to display the current readings of the sensors.

5.4 How the User Can Troubleshoot the Product

In order to troubleshoot the product, users can log in to their personal account on the website and check the readings from the sensors. If the data is being periodically updated on the website and the displayed readings seem to be reasonable, the system is working properly. It is important to highlight that with the exception of the gas sensors, all other sensors will only display two possible values, one of them indicating a security breach and the other representing no breaches. Therefore, it should not be hard for users to identify if the data displayed is reasonable. For the gas sensor, the users can check if there is a warning message displayed for the reading shown on the website. There are a few potential issues that users might encounter if there are any hardware malfunctions in the system:

- One of the sensors is not displayed on the website or the data for the sensor never updates. This probably means that the circuitry for this sensor is malfunctioning and needs to be replaced.
- There is no data displayed on the website or the data in the website is never updated. This most likely means that the main unit of the system (the receiver) is not working properly and must be replaced. There is also a less likely possibility that all sensors are malfunctioning.

The user might also want to troubleshoot the email alert system. In order to do so, the best alternative would be to trigger each sensor to see and see if the alerts are being sent. The most likely cause of failures in the alert system would be that the email provider is labeling the notifications as spams. The solution would be to check the spam box and add the sender of the email to the contacts.

6 To-Market Design Changes

6.1 RF Network Subsystem

The RF Network subsystem proved to be reliable. Sensor readings where being documented in real time and picked up by their respective microcontroller properly. The microcontrollers had an established connection to the main board and readings collected from the sensors where being transmitted to the board with accuracy. To improve this subsystem it would be best for a different microcontroller to be used. The ESP32 used ended up consuming a lot of power and ideally this should be reduced as much as possible.

6.2 Chassis Design

The chassis used in this home security system functioned as expected. Each chassis proved to be reliable and sturdy enough for proper use and application in any household. In addition, the chassis's did not interfere with readings from any of the sensors. Even though the chassis worked well improvements to their design could be made. The material from which the chassis's where made from was limited to the material available from the EIH. ABS 3D printer filament was used. It proved to be rigid and sturdy enough to hold the sensors inside or outside of a house. It would have been neat to have had the opportunity to experiment with other materials to see if different case thicknesses and dimensions could be used to improve sensor casing.

The chassis where also designed with the intentions of placing the battery holder and circuit board next to each other. In a future iteration it would be nice to have the battery holder placed behind the circuit board and have a tab that can be opened or closed for easy battery change. This would allow the chassis built to be much more compact and would give them a much more subtle look. It would also be nice in future generations for the top and bottom parts of each case to connect to each other with hinges for

easier opening and closing rather than the current lips you the client has to screw together.

6.3 Communications Subsystem

For the communications subsystem, the most important improvement would be to have an additional user interface through a mobile application. Although the website conveys all information form the sensors to the users and the alert mechanism sends emails, it is likely that many users would prefer to use a mobile application. Due to time constraints, it was not possible to include a mobile application in addition to the website.

An advantage of a mobile application is that it could have an integrated alerts system through app notifications, which would eliminate the need to have a separate server to send emails to the users.

The general idea of the mobile application would be very similar to the website. It would still get the readings from the sensors through API requests to the same database used for the website, so there would be no need to make any changes to the RF subsystem or to the integration between the subsystems in order to add a mobile application. The technologies used to create this application would also be extremely similar to the ones used on the website. The code for it would still involve a combination of JavaScript, CSS, and HTML. The only slight difference is that the React Native framework for JavaScript would have to be used instead of the React framework, which is its counterpart for websites.

Another change to the communications subsystem to make the product commercially viable would be to modify the website to make it more visually appealing, add more details about the company and the home security products, and most importantly, implement a mechanism to allow users to purchase the product online. This purchasing feature was out of the scope of the project but would be essential for a company trying to sell the product.

7 Conclusions

We successfully developed and built a home security system that met our initial goals and requirements. All hardware parts purchased came at a low cost without compromising quality and efficiency, ensuring that we met our foundational aim of making an affordable product. The home security system successfully detected breaches and notified users instantly via email messages, prompting them to view the website which had more detailed information on the breaches. We were able to integrate various areas

of learning from engineering, such as embedded system design and web development. Although there remains room for improvement, the idea of a low-cost home security system is realisable and scalable from the current design we have.

8 Appendices

8.1 Schematic and Board

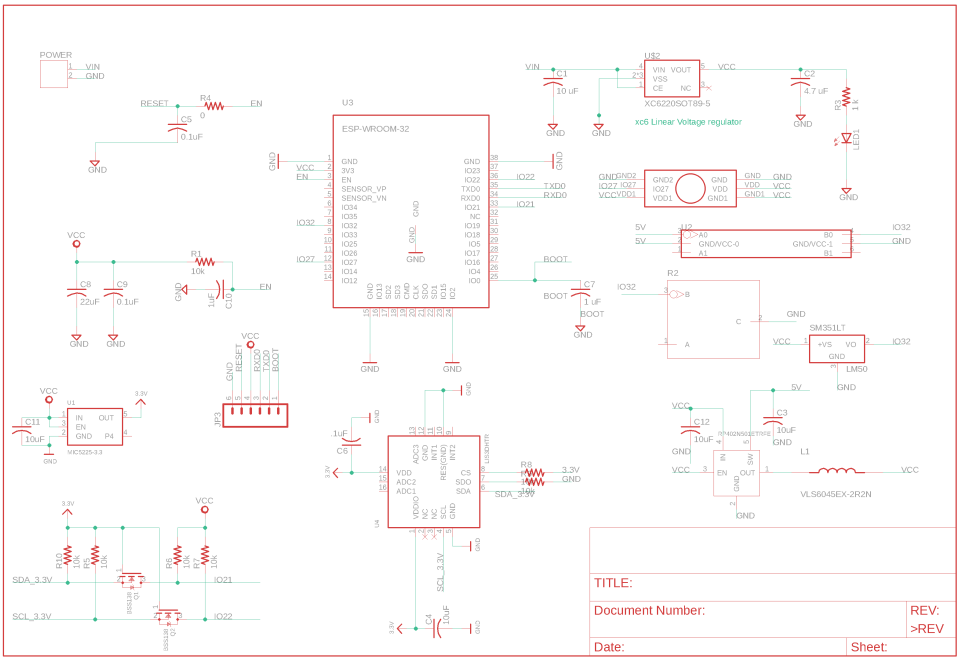

Figure 8.1.1: Device Schematic

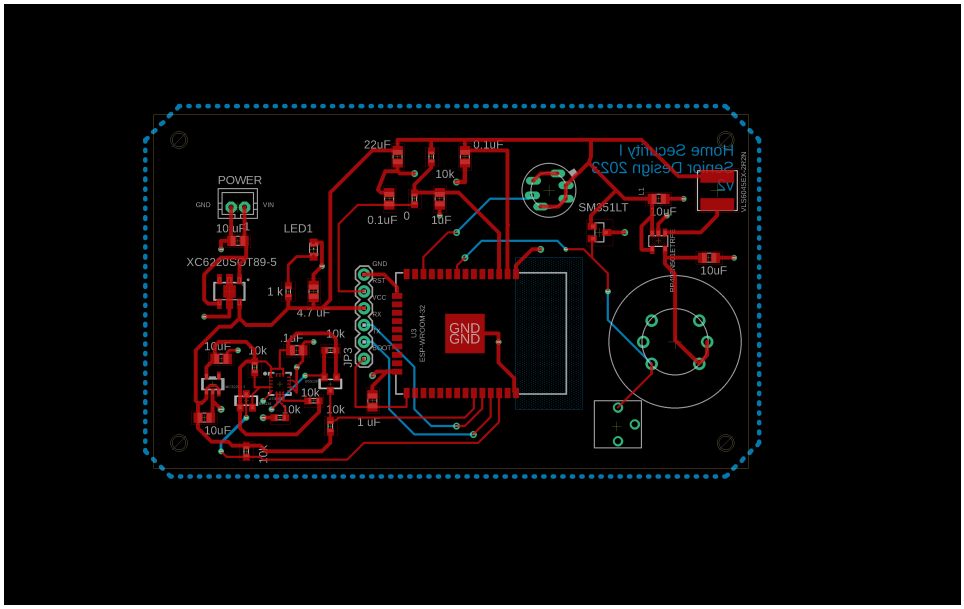

Figure 8.1.2: Board Design

8.2 Software Listings

The complete code listing is available on our website:
http://seniordesign.ee.nd.edu/2023/DesignTeams/homesecurity1/top_page.html

8.3 Datasheets

SM351LT Contact Sensor buy at: `www.digikey.com/short/wcnd2h0n`

MQ-2 Gas Sensor buy at: `http://sfe.io/p17049`

BS612 PIR Sensor buy at: `https://www.adafruit.com/product/5578`

LIS3DHTR Accelerometer buy at: `https://www.digikey.com/short/7nhn3mjv`

ESP32-WROOM-32E buy at: `https://www.digikey.com/short/783dzpjv`