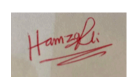
Department of Electrical
& Computer Engineering
Faculty of Engineering & Architectural Science

Course Title:	Engineering Design Project
Course Number:	ELE 70B
Semester/Year (e.g. F2017)	W2023

FLC	Dr. Farah Mohammadi

Assignment/Lab Title:	Theory and Design

Submission Date:	May 2, 2023
Due Date:	May 2, 2023

Student LAST Name	**Student FIRST Name**	**Section**	**Signature***
Ali	Syed Hamza	n/a	
Fessahaye	Naod	n/a	N.F
Al-Kwiliy	Fayez	n/a	F.A.
Bukhari	Daniel	n/a	D.B.

Index:

SMART POWER MANAGEMENT SYSTEM [FM01]
Theory/Design Project Report - Winter Semester

By: Naod Fessahaye, Daniel Bukhari, Syed Hamza Ali, Fayez Al-Kwiliy

Abstract:

For this project, a smart home system was designed with multiple features such as face recognition and Infrared sensor. Implementation of these features was completed via modules connected to Arduino uno. Infrared sensors are implemented to check for people and access the home system accordingly. An esp32 camera is also implemented for face recognition. The purpose of the face recognition is to check users face and compare with the already existing data in the database and if recognized, allow access to home applications such as tv. The database chosen to be implemented is MySQL using HTTP response.

Introduction:

Technology has become a major part of modern society ever since the invention of the internet. It has aided in the connection of various things with one another as well as between various items and people. Everything has gotten so incredibly convenient, whether it's for evaluating, updating, or even creating. Power management is important in modern homes with lots of technological equipment. Every device's turning on and off needs to be automated given our hectic lifestyles. Whether in a little room or a large hall, the lights should turn on and off based on whether a person is present or not. This will not only automate electricity management and reduce power consumption but also make daily tasks easier. The television should also be turned off in a vacant space. It can also be activated by identifying the person's face. This can be implemented using various sensors to constantly check on the vicinity of the house. Additionally, face recognition can be implemented for further management of home applications.

Objectives:

The main objective is to create a smart power management system which can keep control of specific devices in terms of turning them off when a person is not in a room, to save energy. The product will have two features. The features are an IR sensor to check for people and turn the home light accordingly on or off and an ESP32 CAM for face recognition to turn specific home applications on and off. Another objective is to provide a cost effective option for smart power management systems. By making this product, we can also learn how electronics communicate through different signals and mediums.

Theory:

Iris Recognition Theory:

Iris recognition is an emerging field of study that will increase in use in the future. Currently, face recognition is used for highly secured applications such as government and corporation security access. The main purpose of face recognition is to identify a person with high efficiency and accuracy. Iris recognition works by taking a high contrast photograph of the user's face using near-infrared light. This is then run to check for unique patterns in face. This is very similar to how face recognition works.

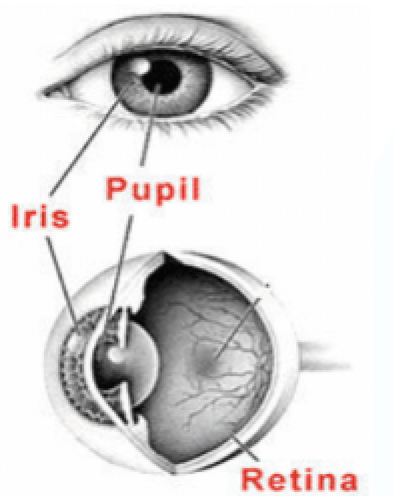


Figure 1. Human Eye Configuration

Figure 2. Eye

To take the image and complete face recognition various computer algorithms must be used. For example, the hough transform can be used to determine the parameters of simple geometric objects, such as lines and circles, present in an image. This algorithm can be used to transform the face image and find the coordinates of the pupil and face regions.The circle region (face and pupil) is taking and is defined by $X^2_c + Y^2_c - r^2 = 0$, where r is the radius and X_C and Y_c are the coordinate of the center.

Following this, the region captured is normalized to a fixed dimension to allow a constant compression between images captured. For example; the product should be able to detect that two images of the same face that are taken in two different environments are the same. This is done using the Daugman's Rubber model which unwraps the face and converts it into its polar equivalent. The center of the pupil is used as a reference point. To convert to polar scale from cartesian scale we use the remapping formula.

After this, the next stage is to extract the face patterns. To extract spatial patterns, the phase information of the patterns is used. There are various feature extraction Algorithms, amongst them are Wavelets Transform, Gabor Filter, Laplacian of Gaussian filter, Key Local Variations Hilbert Transform and Discrete Cosine Transform. The most commonly used algorithm in this stage is wavelet transform and Gabor filter. Wavelet transform works by dividing the face into different components with different resolutions and encodes and generating a biometric template. Similarly the Gabor filter also extracts patterns using quadrant 2D Gabor wavelets. The quadrants are separated into four planes and each is replaced by two bits. Pixels from each quadrant are then extracted into two bits code templates. Ofcourse, this data obtained is run through an algorithm to check for matches.

Face Recognition Theory:

Face recognition is an emerging field of study that will increase in use in the future. Currently, Face recognition is used for highly secured applications such as government and corporation security access. The main purpose of face recognition is to identify a person using faces with high efficiency and accuracy. A popular facial recognition approach is feature-based face recognition, which analyzes local aspects of faces to distinguish distinct persons. The key local characteristics taken from the collected image are the nose, lips, eyes, chins, and head.

Figure 3: Face recognition

Binary classification is a feature-based method. Binary classification compares the image to the composite data set of the feature's image; it would scan over every area on the face until it located the location where it believes it has matched the feature, such as the eye. For numerous reasons, this strategy is inefficient: You must search the entire image beginning with index 0; this would require an inordinate amount of calculation time and processor power. You'd run into the local optima problem, in which the algorithm chose the optimum maxima point without considering the remainder of the image. For example, the algorithm could focus on the mustache and determine that it is the brow.

Elastic Bunch Graph Matching (EBGM) is a more efficient Feature-Based Face Recognition system that employs nodes to form a topological graph. The nodes define the basic form of the face, and additional nodes develop in crucial identifying locations such as the eyes and nose. EBGM recognises faces by comparing the probe set input to the gallery database and comparing it to other probe sets. The nodes are linked to form a web-like structure.

The minimum distance between the input face graph and the model face graph must be reached in order to correctly identify a face for the input face graph. Face recognition with EBGM can be improved by weighting Gabor kernels. In comparison to utilizing the lowest weighted kernels, adopting the highest weighted kernels would result in a better depiction of faces and higher match rates.

Passive Infrared Sensor Theory:

The functionality of the PIR sensor outlined its ability to sense infrared light from a warm body person or animal through the on board fresnel lens. The fresnel lens has two sections which one section has the responsibility to intercept the infrared from the passing warm body. This occurrence allows there to be positive change in detection of infrared between the respective two sections. The signal from the PIR sensor would be a pulse in the positive direction. The opposite reaction occurs when the warm body that emits the respective infrared leaves the range of the PIR sensor. When this occurs the fresnel lens second section detects that the warm body is leaving range thus the PIR sensor will generate a negative pulse.

The fresnel lens has plentiful concentric grooves that act as individual surfaces that refract incoming wavelengths; this increases the probability of the sensor to detect infrared . Also depending on the construction of the infrared sensor module the infrared LED would be more powerful due to the input voltage as well as the lens at use. This increases the distance in which the infrared sensor can detect an individual is present. From the perspective of the physical interaction of the infrared light, it is considered a common electromagnetic radiation which lies in the light wavelength index in between the visible light and microwave part of the spectrum, specifically the wavelength is described approximately the 0.76 um up to 1000 um. Now the benefit of the infrared light can be seen as it is less absorbed and scattered into the atmosphere when in place in comparison to the visible light spectrum. The industry utilizes and prefers the Infrared light as the NASA telescopes which interpret space geography utilizes the infrared light imaging systems.

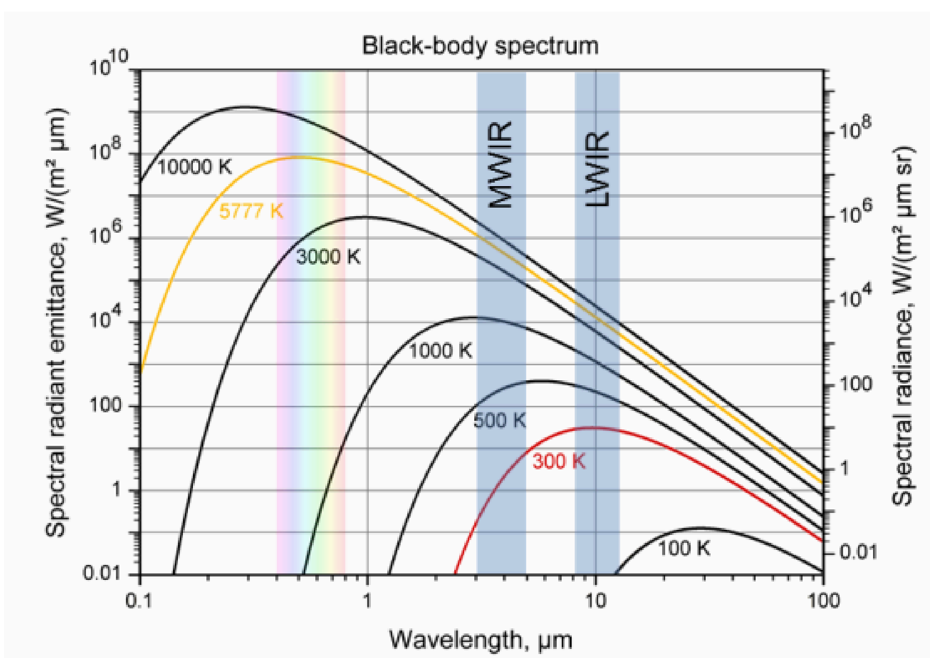


Figure 4: Black-Body Spectrum

Block Diagram Design

Preliminary Design:

Figure 5: Product Design

In Figure above, the block diagram of the product design can be seen. The blocks that are double outlined represent the interactive features in the product. A complete breakdown of each module can be seen below.

Intermediate Block Diagram for the PIR Sensor Circuit and Central Module:

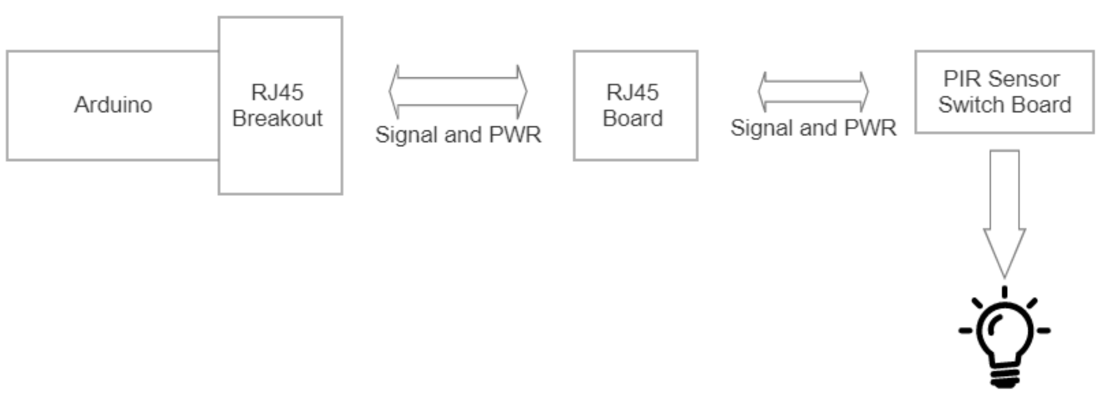


Figure 6: Intermediate Design for PIR Sensor Circuit

The above diagram that includes the PIR sensor circuit describes the interfacing the PIR sensor switch board will have with the Arduino. The RJ45 breakout board serves the purpose of passing the power signal through the same wire. The RJ45 block as shown above will consist of the RJ45 adapter port as well as its own breakout board as the individual wires will have to routed their respective ports on the PIR Sensor Switch board. The PIR Sensor Switch board consists of the PIR sensor having its own wires interconnected to appropriate locations. Furthermore, the TIP120 TU NPN darlington transistor pairing as a BJT. The design choice for the TIP120TU was due to its NPN darlington configuration which allows a smaller base current to be used to have the over transistor turn on which in return allows the Arduino to turn on the transistor which will have the load connected to the test source.

There will be ethernet cables to connect one RJ45 breakout board to the other and this ethernet wire will carry the Signal Out from the PIR sensor, 5V to power the PIR sensor and Signal In from the Arduino board. The Signal Out from the PIR sensor will allow the central module board to keep track of how many times the lights have been turned on. The Signal Out is a signal coming from the Arduino which will occur subsequent to the Signal In from the PIR sensor. This will ensure that the Arduino has been able to track the light PIR sensor being triggered and its own GPIO Pin will send a SIGNAL OUT to the base of NPN BJT.

Motion detection was a vital focus point of the smart home power management module and to have this capability the PIR sensor module was selected. The PIR sensor is a sensor which detects the change in infrared emission through its lens. The motion of humans displaces the infrared thus it is detected through the PIR sensor. Furthermore, the PIR sensor stands for passive infrared sensor and the chosen PIR sensor has utilized the fresnel lens PIR sensor. The fresnel lens was a decisive point for the sensor, as it allows more refraction of incoming infrared light, therefore, a larger chance of detection of incoming infrared light. The end goal of the product is modularization and adaptability to user function. Thus the additional ESP32 module was placed in the center module as it enables the Arduino to send its tabulated data. The tabulated data consists of how many times a person has turned on the light and it is sent via the onboard wifi module that resides on the ESP32, enabling the feature of the website to be updated.

Final Block Diagram for the PIR Sensor Circuit and Center Module:

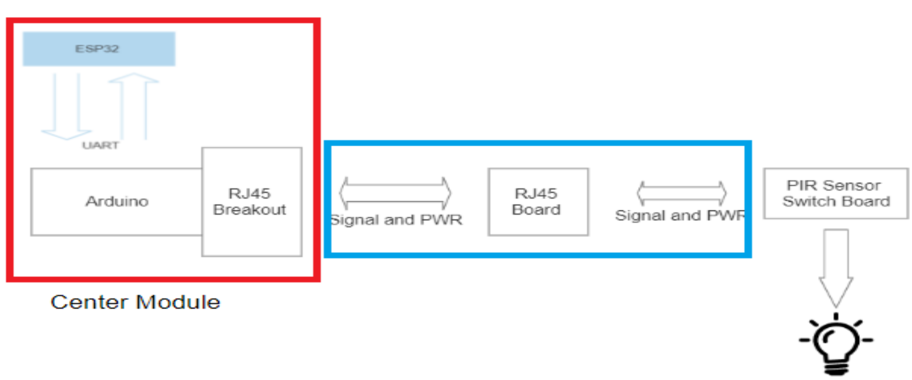


Figure 7: Final Block Diagram of the PIR Sensor Circuit

The figure above describes the final implementation of the PIR circuit along with the center module is in red. The center module consists of the voltage divider network between the TX pin of the Arduino and RX pin for the ESP32 as the 5V logical signal can not be interpreted by the ESP32. Thus, the voltage divider network makes the 5V logical signal coming from the Arduino to the ESP32 as 3.3V. Furthermore, this is beneficial as the tabulated data is being sent via UART from the Arduino to the ESP32. The center module consists of the ESP32 and Arduino. The interfacing method for the center module to the PIR sensor circuit is shown in the figure above with the blue box representing the RJ45 breakout board and ethernet cable used.

UART communication protocol is Universal Asynchronous Receiver/Transmitter and it was utilized as it communicates the data in a serial communication manner offering an implementation that streamlined the process. This was due to UART having the ability to use the baud rate from the Arduino and have that match with the ESP32 also the TX and RX from both respective microcontrollers were only utilized. The final implementation as depicted in the above block diagram has indicated the ESP32 and Arduino would be utilized in the center module. The center module would be interfacing the PIR circuit with the RJ45 breakout board as well as the RJ45 CAT6 ethernet cable which would be carrying the 5V from the Arduino to power the PIR sensor, Signal IN from the PIR sensor when detection of motion has been made. GND and the Signal OUT which would be a logical 5 V signal to trigger the BJT base to allow the source to power the LED/light load.

Features:

Final Housing Design for the PIR Sensor

Figure 8: Final Housing Design for PIR Sensor Circuit

Infrared Light Sensor Block

The PIR switch board along with the PIR sensor will be placed in a light switch housing and the ethernet will run out of the housing to the housing for the central Arduino. This creates an organized system with clear modules. With the perspective of the Passive Infrared Light sensor block which serves the purpose of detecting Infrared Light from the fresnel lens to detect whether a presence of an individual is there or not. There is a maximum distance in which the warm body can be detected and that would be 6 m. The angle of view of fresnel lens 120 degrees, this angle of view is proportional to distance in which the lens can detect infrared. The detection will be done by the PIR sensor as it outputs a positive pulse when the initial detection of infrared waves are from a warm body and then a negative pulse will be sent when the warm body leaves the field of view. These two signals are handled by the Arduino respectively through conditional execution to record the amount of times the PIR sensor detected a warm body. The dynamic array size would do so in the Arduino code.

Camera

The camera block that is depicted above can be described for its ability to capture images of the user's face. The face is unique from individual to individual and as the ESP32 camera module captures images of the user's face it will map the unique curvature and patterns of the user's face. The purpose of this block is to serve as a checkpoint before any user can access specific functions in the home.

Alternate Design Analysis and Approach

For the perspective of the Infrared Light Sensor Module required the development of testing methodologies to test the functionality of the module and occurred with the development of test code. With the comparison of two infrared light sensors which were readily available from trusted vendors. The TCRT-5000 VMA326 module is first looked upon module:

Figure 9. TCRT5000

The operational voltage is 3.3 V to 5V which is in the range of the voltage which can be supplied to the microcontroller which is important with consideration of designing a printed circuit board where a voltage regulator would be providing the same voltage to all the elements. The board is small considering the 42 x 10.5 mm PCB size. The detection is 40cm. The price is 7.32 CAD

```
/*DanielBukhari Test Code IR SENSOR*/

// Arduino IR Sensor Code
int IRSensor = 9; // connect ir sensor module to Arduino pin 9
int LED = 13; // conect LED to Arduino pin 13
void setup()
{

  Serial.begin(125200); // Init Serila at 115200 Baud
  Serial.println("Serial Working"); // Test to check if serial is working or not
  pinMode(IRSensor, INPUT); // IR Sensor pin INPUT
  pinMode(LED, OUTPUT); // LED Pin Output
}

void loop(){


  int sensorStatus = digitalRead(IRSensor); // Set the GPIO as Input

  if (sensorStatus == 1) // Check if the pin high or not
  {// if the pin is high turn off the onboard Led
    digitalWrite(LED, LOW); // LED LOW
    Serial.println("Motion Ended!"); // print Motion Detected! on the serial monitor window
  }

  else{//else turn on the onboard LED
    digitalWrite(LED, HIGH); // LED High
    Serial.println("Motion Detected!"); // print Motion Ended! on the serial monitor window
    }
}
```

Figure 10. TCRT5000 Test Code

The above code implementation was a test code which provided an understanding that the usage of the library from the Arduino is sufficient.

The comparison was made to the Sharp GP2Y0A21YK0F IR Distance Sensor which provided the working voltages of 4.5 V to 5.5 V which falls under the operating range of our specification. The price of the product is 14.33 CAD and detection is 80 cm.

```
/*SHARP GP2Y0A21YK0F IR distance sensor Test */
/*Daniel Bukhari*/

// Include the library:
#include <SharpIR.h>

// Define model and input pin:
#define IRPin_1 A1
#define model 1080

// Create variable to store the distance in cm:
int distance;

/* Model :
  GP2Y0A02YK0F --> 20150
  GP2Y0A21YK0F --> 1080
  GP2Y0A710K0F --> 100500
  GP2YA41SK0F --> 430
*/

// Create a new instance of the SharpIR:
SharpIR mySensor = SharpIR(IRPin_1, model);

void setup() {
  // Begin serial communication at a baudrate of 9600:
  Serial.begin(9600);
}

void loop() {
  // Get a distance measurement and store it as distance_cm:
  distance_cm = mySensor.distance();

  // Print the measured distance to the serial monitor:
  Serial.print("Mean distance: ");
  Serial.print(distance_cm);
  Serial.println(" cm");

  delay(1000);
}
```

Figure 11. Sharp Infrared Sensor Test Code

The tested product was the SHARP IR sensor due to its ability to detect the presence of the individual within 80 cm which is greater than the TCRT 5000 Infrared Sensor. Due to the SHARP Infrared Sensor, its long-distance reading capability and implementation constraints that stayed within the Arduino's capabilities, the Sharp GP2Y0A21YK0F IR Distance Sensor module was chosen. Before proceeding any further, the purpose of the developed test script was to evaluate the sensor's fundamental functionality.The purpose of the test was to determine the distance from an object that was in the way of the IR sensor's line of sight. In this case, the object in question was my hand.The SharpIR.h was incorporated into the code and had its strategy functionalities through the Sharp IR library.The test script's methodology involved including the Sharp IR sensor library via #include SharpIR.h.Since the Sharp IR sensor has an analog output, it was physically connected to the Arduino Uno's ADC pins.

Moreover, the distance_cm variable was instated as whole number as it would be a comparing perusing from the sensor.The Arduino was used with the baud rate set to 9600 because that was the rate that worked in previous projects.The IR_sensor.distance() method was utilized to correctly relate the ADC value to a readable value that could be printed on the serial monitor and read the ADC value.The sensor used an average of 25 readings every 53 milliseconds to calculate the distance value.The SharpIR(IR_Pin_1,model) object was assigned to the variable IR_Sensor.The model of the sensor that was being used was reflected in the model variable.In addition, the readings contained noise and jitter because the values were abnormally large in comparison to the actual distance that my hand was from the sensor.As a result, a 100uF tantalum capacitor was chosen to filter the power lines, and the sensor values were as expected.

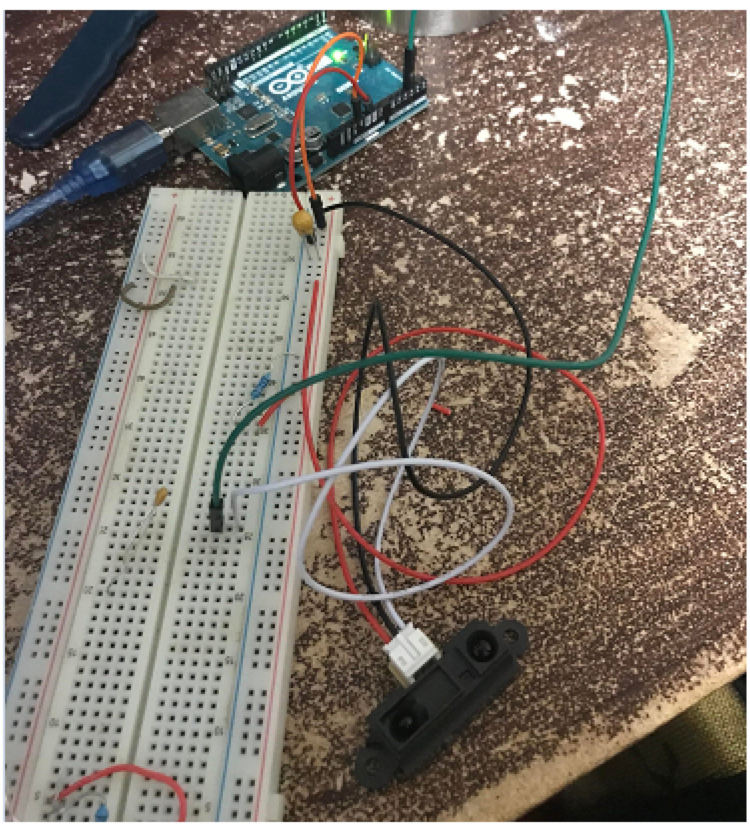


Figure 12. Sharp Infrared Sensor Test Circuit

Final Design Analysis and Approach

With further analysis and the scale in which the product will take on, the Sharp IR sensor detection distance will not be beneficial as it is only 80 cm. The new scale of the project will be for a real life sized home thus the 6 m detection was the distance chosen. The fresnel lens offers further concentration of an infrared wave from a larger distance as the lens itself magnifies the waves.

The PIR Sensor Switch Board will consist of the following components: barrel jack adaptor, screw terminals, TIP120TU NPN BJT Darlington transistor, 6 Pin female socket, 3 pin socket. The purpose of this board will be for the demonstration as the load and the supply will be interfaced with it. The 12V load Light bulb will be interfaced to the 12 V barrel jack adaptor. The power supply will be attached to the screw terminal.

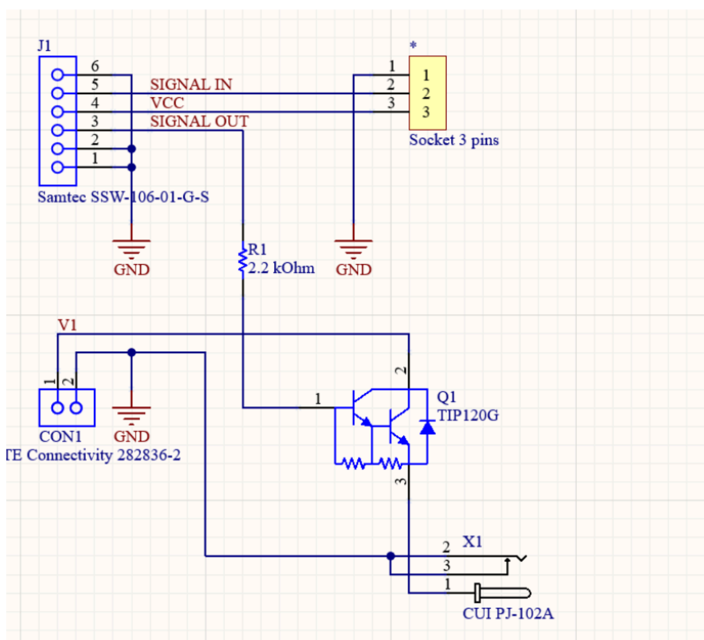


Figure 13. Schematic Circuit Design of PIR Switch PCB

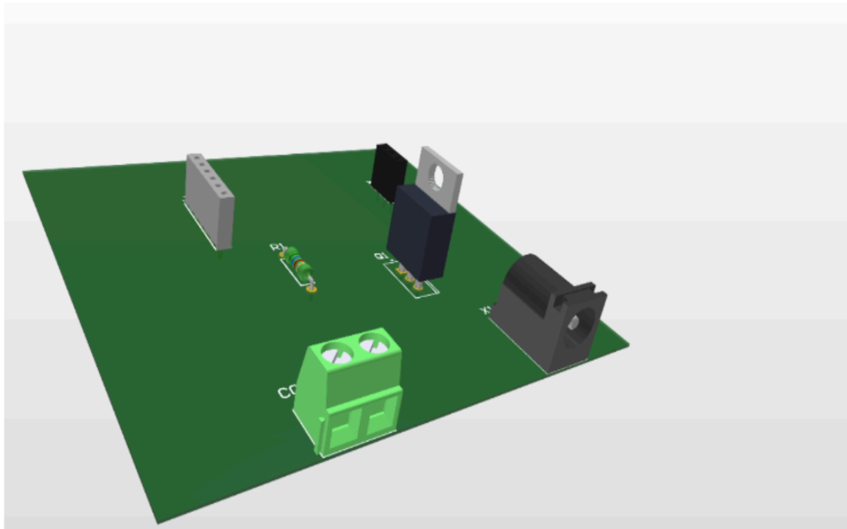


Figure 14. Schematic Circuit Design of PIR Switch PCB

The PIR Switch Sensor PCB board design was done through Altium Circuitmaker where the signal net for GND was connected commonly amongst all the vias. The board was designed to be a two layer board as the routing of the PCB traces can be done on the top layer and bottom layer. Furthermore, the ground plane was made common across the GND vias as the polygon pour function was utilized to make a GND plane. The purpose of this was to minimize the distance which is required for the signal/PWR line to return back to GND. As the longer the return path the more possibility of the line to induce noise in the circuit and this can negatively impact the signal out from the PIR sensor.

The design approach taken for the ESP32 CAM was the following the test connection between the computer and the cam module which was established. For initial testing purposes, a simple Wificam program is initialized from the esp32 cam library acquired from the last reporting period. For this program, the wifi ssid and password are entered into the main file of the program and built while the esp32-cam is connected. Upon successful completion the program generated a http server for live camera surveillance. Although this was not successful due to the connection issues with the cam, once the connection was solved, the server functioned properly.

The Arduino microcontroller was chosen in a cost breakdown between the Raspberry PI and the Arduino which costs $ 50 CAD whereas the Raspberry PI is $150 CAD. The widely available and applicable libraries of the Arduino to sensors have made our team decide on the Arduino as the microcontroller.

The hardware connections for the esp32-cam MB component, which is a pin connection component that can be connected to a usb-c cable for testing, was included with the cam module.

A usb-c cable was used to make a connection to the computer using this component for the initial testing.The cam displays a red light that indicates power, but the device manager on the computer was unable to recognize the module.After conducting additional troubleshooting, it was determined that the solution was to switch cables and install a USB2.0-ser driver package. This was a wire connection test which was validated through the prompt of the application on the desktop.

Through the preliminary design approach there were necessary steps taken to evaluate the dependencies and implement the required testing for the different components. The test planning was sufficient proof to conclude that the separate components were functional and the test code passed successfully. This leads the team to transition into more complex testing of conditional testing with the sensors and the implementation of on board data storage.

The overall design of Smart Home Management consists of three separate modules in which data will be stored or manipulated to benefit the user usage of the house. The data stored can showcase the number of times the user utilized a particular portion of the home. Additionally, the user will be able to have additional security within the home as the door access will be elaborately secured through the user's image.

Final Design Approach for the PIR Circuit and Center Module

Functional Testing and Performance Testing of PIR Circuit and Center Module

Functional Testing and Performance Testing

- Test functionality of BJT Darlignton transistor with the Arduino, and successfully witnessed the ability to have it be triggered by the Arduino
- The RJ45 cables and along with the adapters are included in the functional test to ensure signal integrity is met through the long wire and further filtering capacitors are added
- The Performance testing of the PIR sensor was done from 2m away as the PIR sensor was connected to the circuit which had the Arduino triggering the BJT upon PIR sensor detection of movement. Another Arduino was utilized to model the final Power supply that would be utilized to turn on the light bulb in this case the LED.
- The Performance test has passed as the PIR sensor was able to detect motion up to 2 meters away.

Circuit Set Up for Functional and Performance Testing

- Arduino to Trigger base of BJT upon PIR Sensor detection of movement - Arduino Number 2 would provide the power through the collector and emitter for the BJT to light up the LED subsequently after the First Arduino's triggering of the base of BJT - RJ45 Cable utilized to interface the Arduino Number 1 to the PIR sensor and this RJ45 cable is carrying the signal out from the PIR sensor (movement detection), GND, 5V VCC and signal out from the Arduino to trigger the BJT

- Below two figures show the PIR sensor circuit and the main arduino circuit

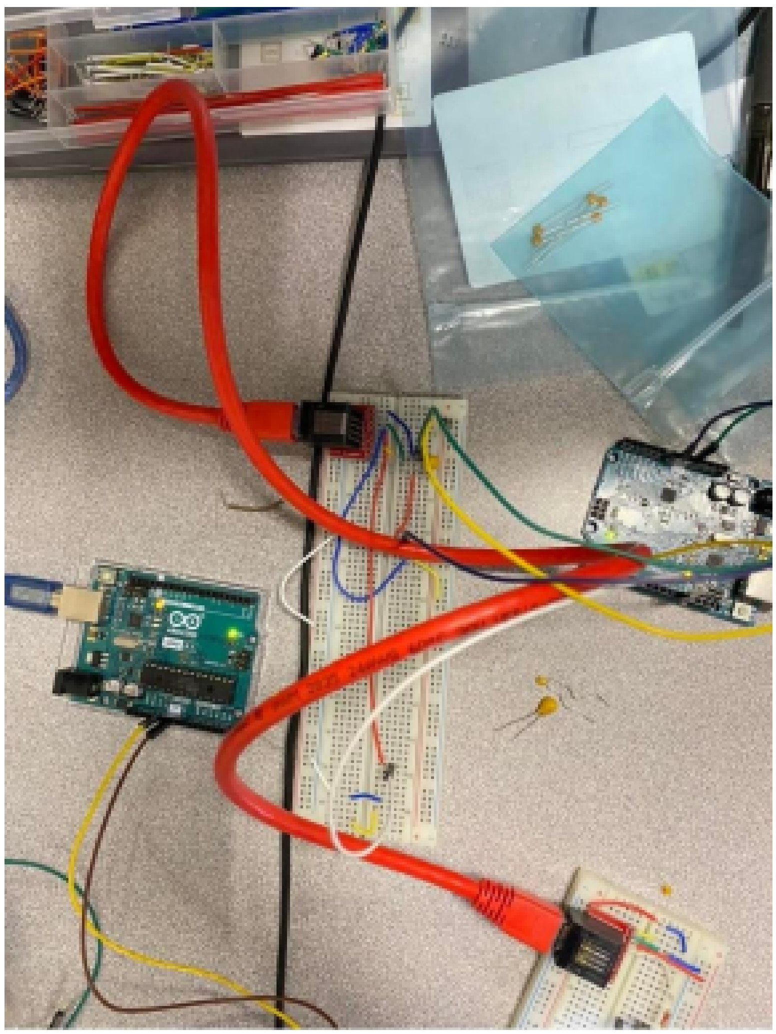


Figure 15: PIR Circuit and Central Board

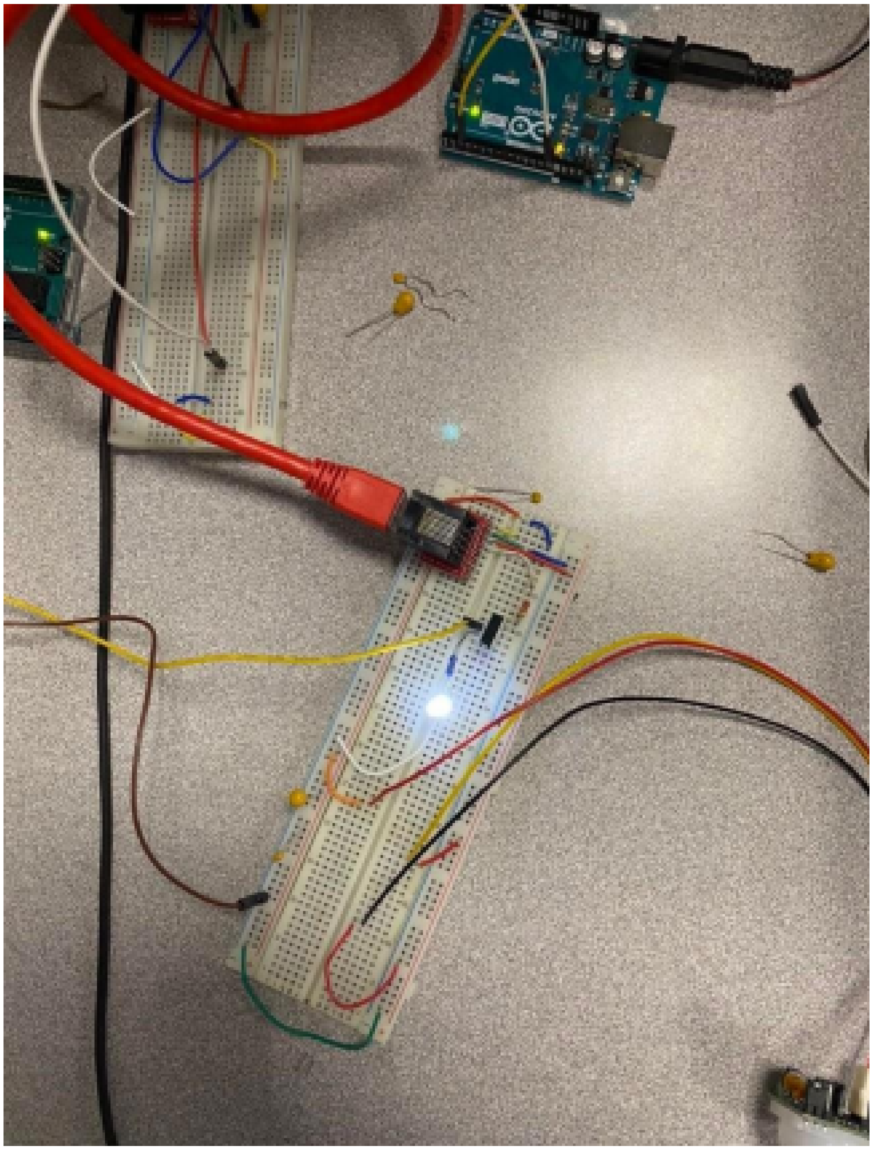


Figure 16: PIR Circuit Test

Functional Test to ensure 5V is distributed through the circuit and that the filtering capacitors are working as intended. Performance Testing of PIR detection from various angles of movement and indeed the movement was still detected. Limitation was when the test subject is 180 degrees to the PIR sensor.

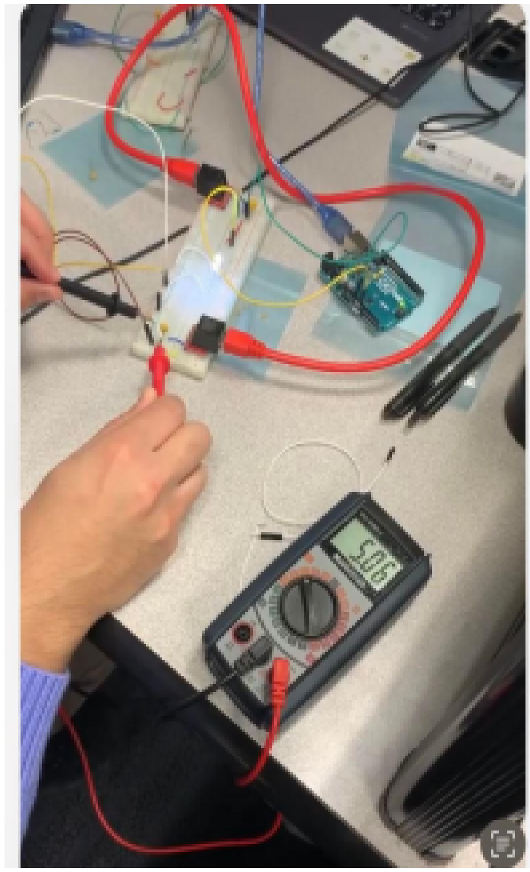


Figure 17: Test PIR Circuit

Furthermore, additional voltage measurements were taken with the volt-meter that indeed the 5V was being distributed through the circuit. This was an essential validation step as it ensures that the components such as the PIR sensor were receiving the desired 5V, and if there were large fluctuations that would have a negative effect on the module's operation. This is due to how the operating voltage of the PIR sensor allows the sensor to reach a certain distance of detection and to reach the 5V desired voltage capacitors have been placed along the VCC lines of the circuit.

The capacitors that have been used are the 100 uF tantalum capacitor as well as the ceramic capacitor 22 pF. The intention of the 100 uF tantalum capacitor with its polarized ends results to the positive end of the tantalum capacitor to be larger. This characteristic is beneficial as the tantalum capacitor has been utilized as a power capacitor to maintain the VCC voltage through circuit to be 5V. Additionally, the ceramic capacitor has been used as the decoupling capacitor to filter any AC noise that may be in the power supply.

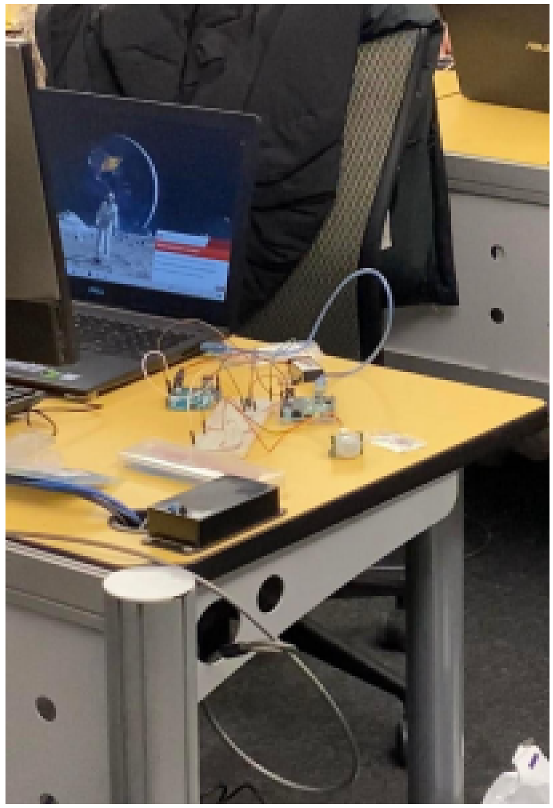


Figure 18: Test PIR Circuit

Performance Testing of the Test Subject being 2 meters away from the PIR sensor was done and indeed the person was detected. The performance testing was conducted separately from the functional testing since the desired distance of the test subject was intended to be found. The functional testing was done by having the test subject stand 20 cm away from the PIR sensor and it was able to detect.

Design of Center Module and PIR Circuit

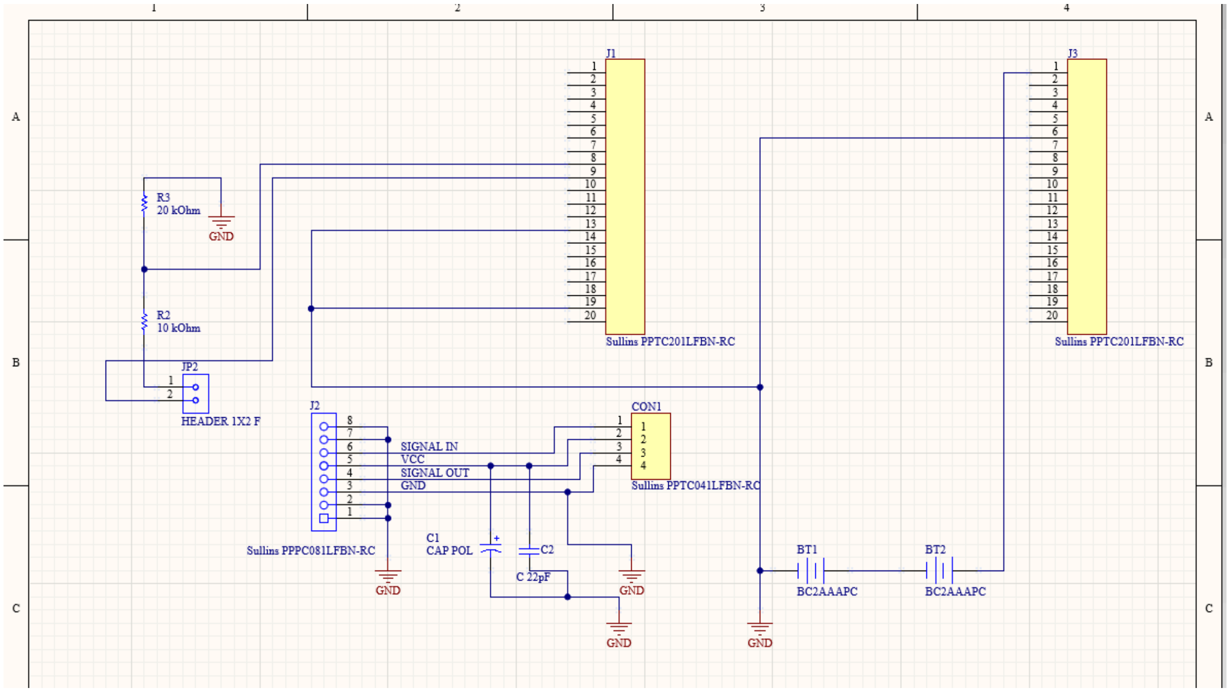


Figure 19:center module

The central hub circuit that is above has been designed to include sockets which interface the Arduino and ESP32, where the Arduino serves the purpose of interpreting the motion detected signal from the PIR sensor, and subsequently after that the Arduino would send a 5 V logic high signal to trigger the BJT to turn on the light. Additionally, the UART communication protocol has been enabled between the Arduino and ESP32 through their TX and RX pins and created a voltage divider circuit consisting of the 10kOhm and 20kOhm resistors to change the Arduino's 5V to 3.3 V. The J2 socket interfaces the RJ45 breakout socket which holds the ethernet cable and its cluster of signals. Within the ethernet cable it holds the SIGNAL IN, VCC, SIGNAL OUT, and GND, and respectively the signals are tied to the CON1 socket which would interface wires that connect to the Arduino. The J1 and J3 sockets interface the ESP32 its pins for the TX, RX as well as GND.

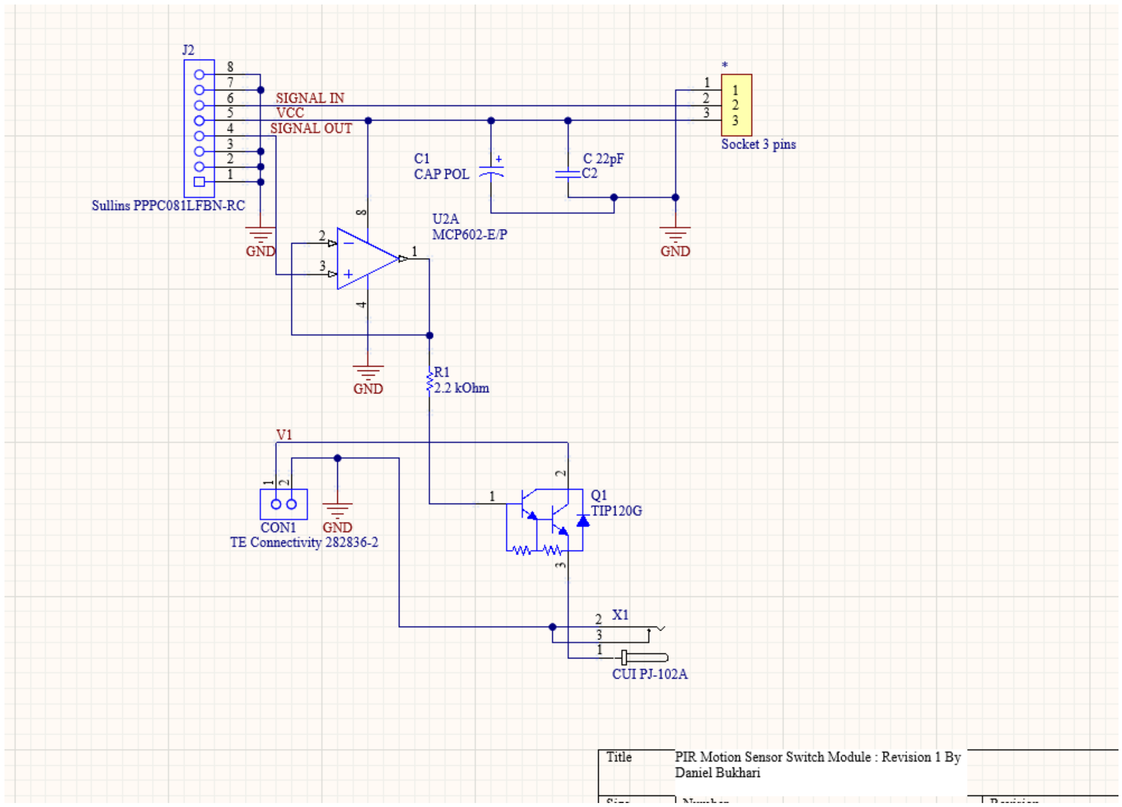


Figure 20:PIR circuit

The above figure depicts the design of the PIR circuit as it would include the addition of the PIR sensor in the PIR circuit and the appropriate BJT TIP120TU was included, a darlington transistor whose base current requirement matched the Arduino output current. The BJT has been chosen in particular as it allows the Arduino to trigger the base of the BJT to turn on the light. The BJT would have its base port connected to signal wire from the Arduino, collector port connected to a power supply and emitter side to the light. The resistor of 2.2 kOhm was utilized as the current limiting resistor in this case to prevent the base of BJT to draw large amounts of current as it is nonlinear which would damage the Arduino port. X1 in the schematic above represents the barrel jack socket that would interface the light load. The V1 is the screw terminal which would interface the power supply. The J2 socket will interface the RJ45 cable and it would consist of the PIR sensor SIGNAL IN, SIGNAL OUT from the Arduino to trigger BJT and GND.

PCB Development for Center Module and PIR Circuit

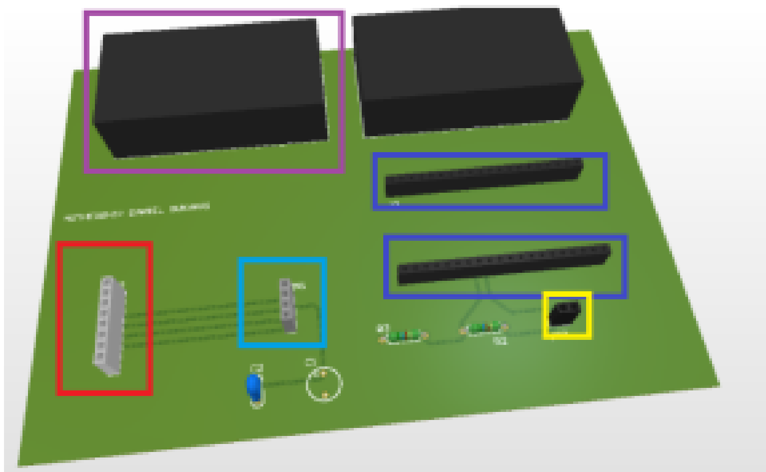


Figure 21:PCB center module

The purple box below represents the voltage source of the esp32 which consists of the four 1.5 V AAA battery which in total is 6V. The blue boxes below represent the sockets which would connect to the esp32. The yellow box represents the socket which would connect to the TX and RX ports of the Arduino to successfully have UART communication between the Arduino and ESP32. The light blue box is the sockets which would connect to the Arduino digital ports to have the PIR sensor signal out (motion detected), signal out which would trigger the BJT on the next PCB, 5V and GND. The red box would have the RJ45 cable connected to it allowing the complete distribution of signal from this central hub PCB to the PIR switch PCB board.

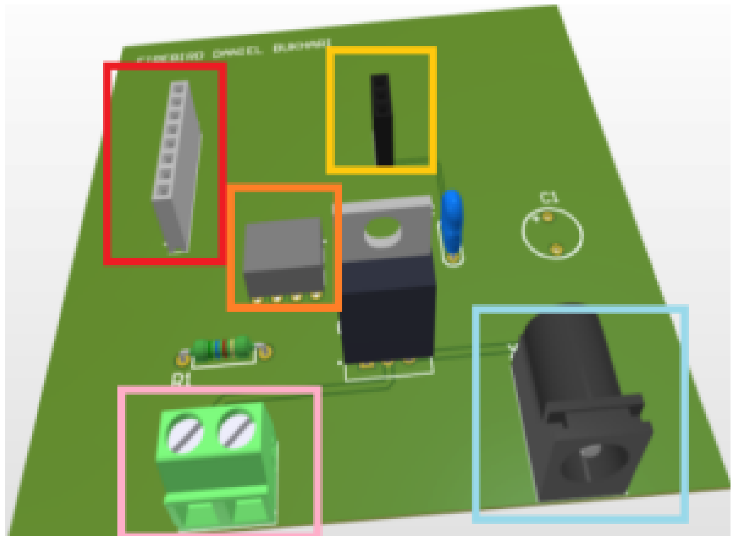


Figure 22: PCB PIR Switch

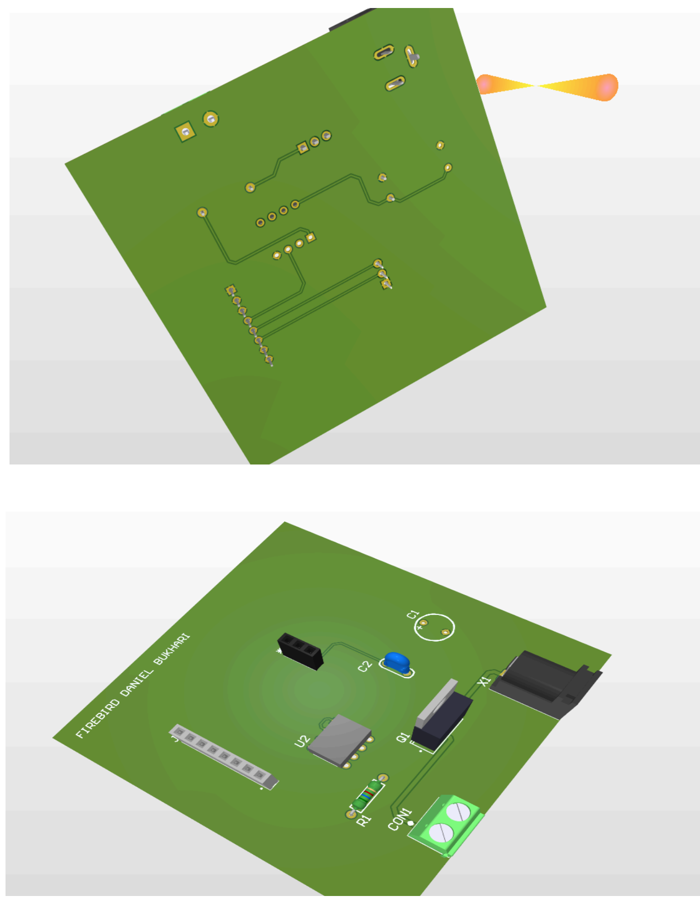


Figure 23: Top View PIR Switch

PIR Switch PCB board

The red box below is the sockets which interface the RJ45 cables carrying the 5V, GND, Signal out from PIR sensor, Signal out from the Arduino to trigger the BJT. The yellow box is the socket which interfaces the PIR sensor. The pink box is the screw terminal which would interface the power supply 12V and the light blue box is the barrel jack which connects to the light bulb. The signal net of ground was made connected commonly amongst all vias. The board is designed to be a two layer board. With my prior design experience in PCB design and Altium Circuitmaker. The two layer board was considered as the ground plane will act as a shielding of electromagnetic noise. The ground plane was done through polygon pour as it reduces the current loop and this reduces the return path of these signals which will be short as a ground plane has been poured.

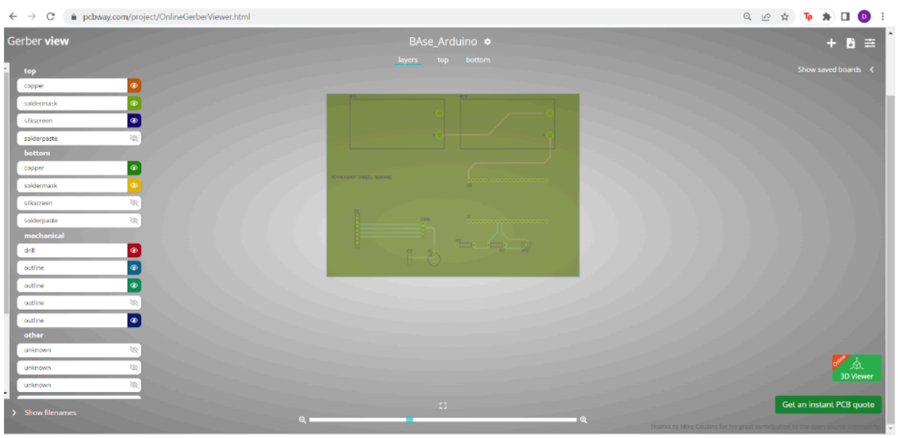


Figure 24: Gerber File View Central PCB

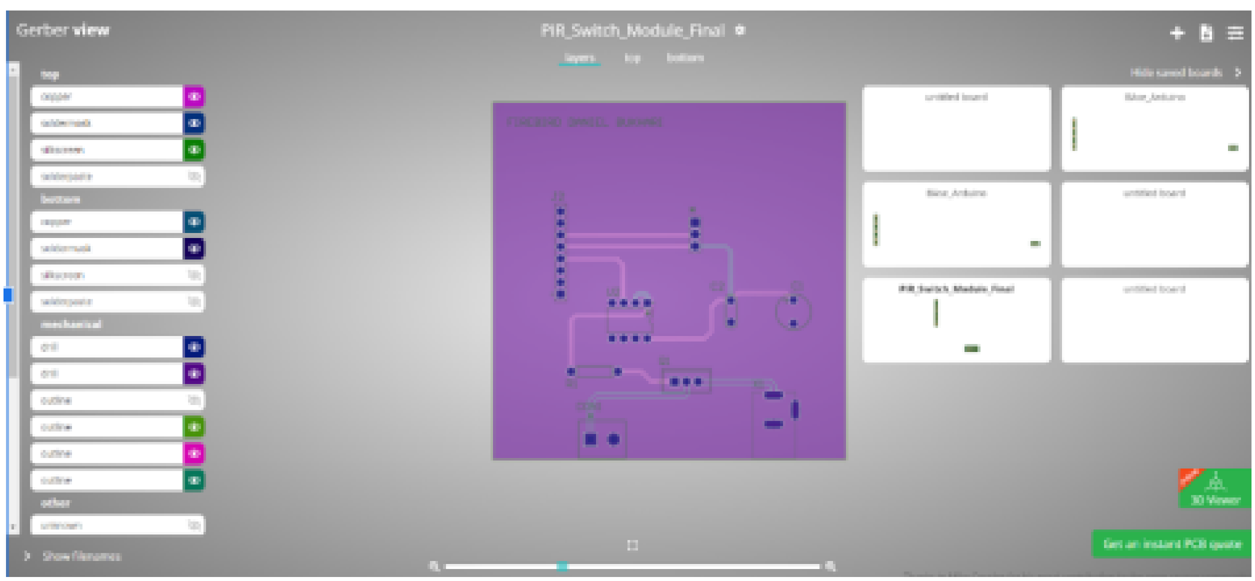
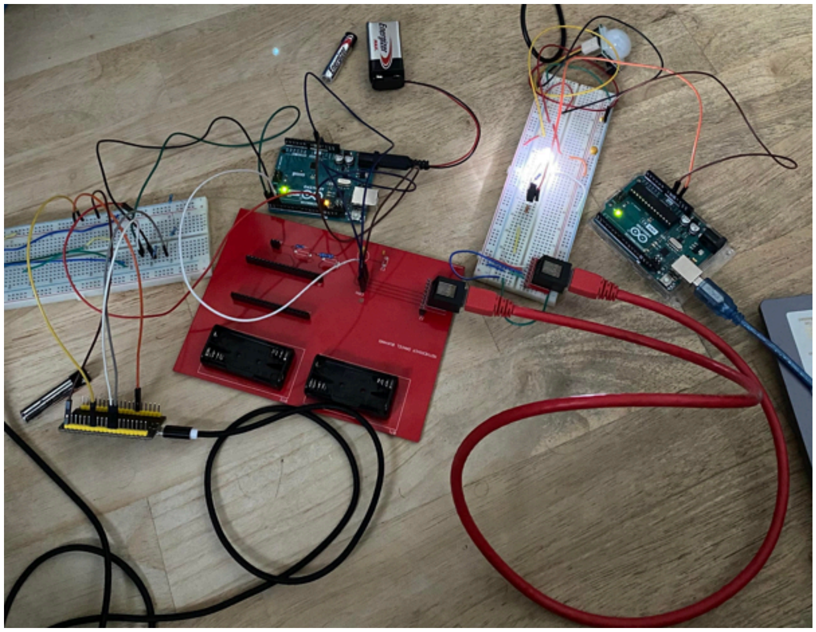


Figure 25: Gerber File View for PIR Switch

Gerber File contained the different layers of the PCB top solder paste, top overlay and etc. The NC Drill file indicated where the connections will be drilled in.Validation of design was done and ordered through PCBWay.

Final PCB Design and Circuit:



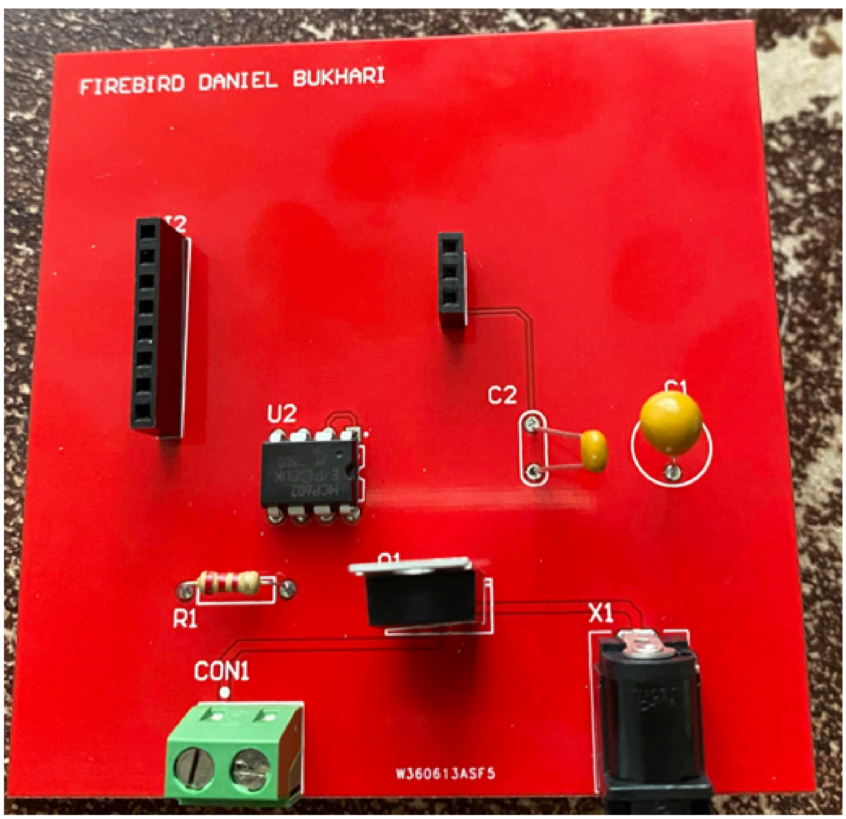


Figure 26: Final Layout of Central PCB and PIR Switch Board

Human Detection:

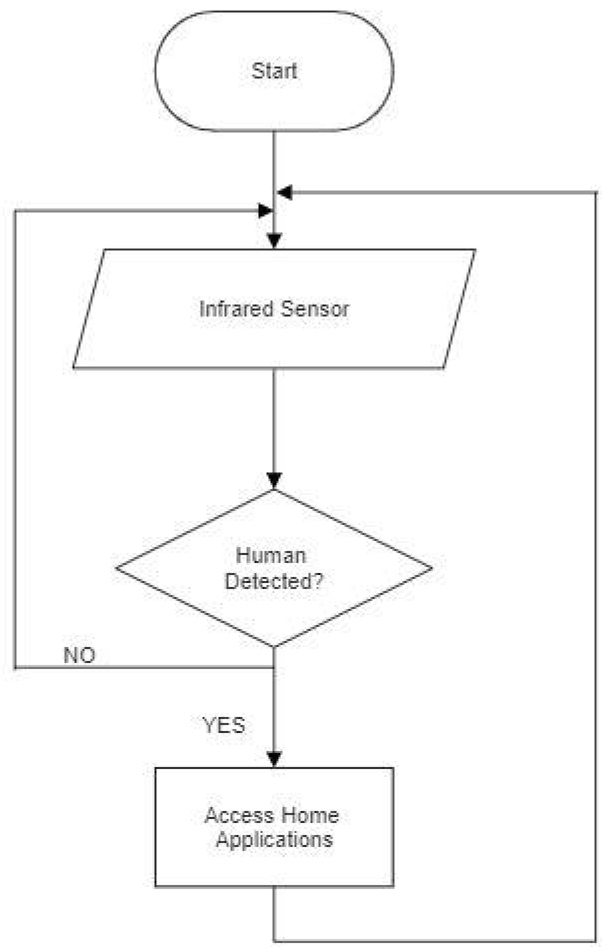


Figure 27. Flowchart of Infrared Sensor Human Detection

The infrared sensor will be used to detect if there are any humans within the range of the sensor, in this case 6 m. If humans are detected, home applications are accessed i.e. home lights turn on.

Face Recognition:

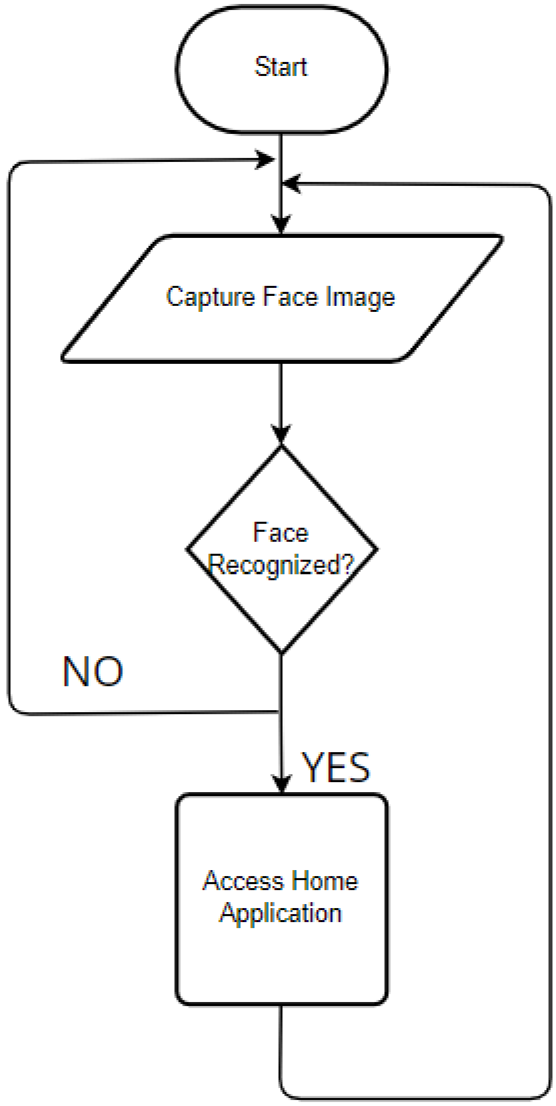


Figure 28. Flowchart of Face Recognition

The flowchart of the face recognition is quite similar to that of the IR sensor, and can be seen in the above figure. The face detection will be realized by an ESP32 cam component which includes an OVA260 camera along with capabilities such as wifi and bluetooth which can be used for web and phone applications for user interaction. This particular block serves the purpose of creating a comparator action between the data which resides in the on module data storage and the data that has been captured by the ESP32 Cam Module. The user's face as mentioned is captured and sent to the data storage which resides on the board. Now, after this initialization process with the user information. The ESP32 CAM module will capture the next image along the microcontroller which would control data flow and the comparison process of the data storage. If the current face reading maps to the authorized user's face that has been saved to data storage, the specific control of the functions in the home can occur.

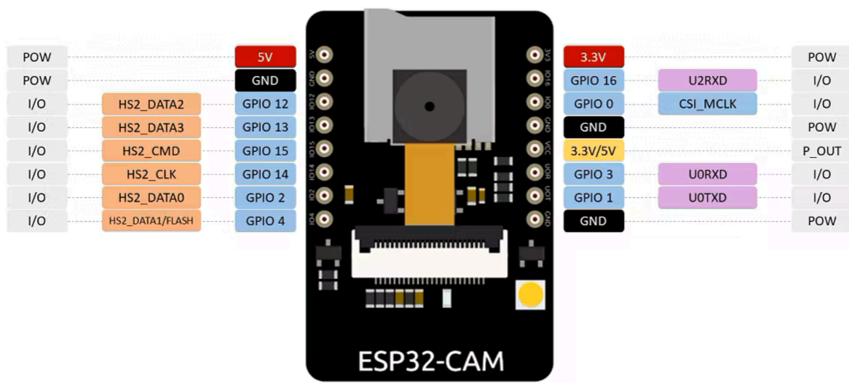


Figure 29. ESP32-cam pin layout.

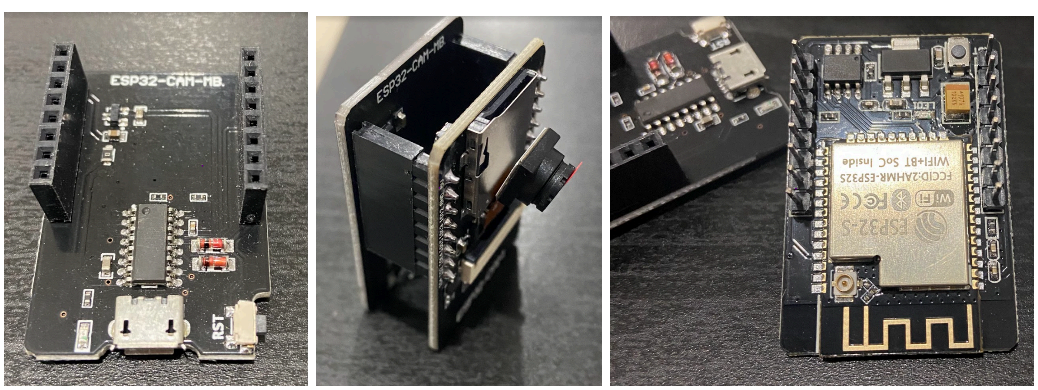


Figure 30. Depicts the ESP32-cam MB component (left), the cam module pins (right), and a combined unit (middle) that were purchased for the project.

The esp32 camera will capture a photograph of the user's face, and a series of computations will take place to compare the image to images found in the database. If the face matches with another in the database, the home applications are accessed for example, tv turns on. This functionality has already been tested with a few arduino libraries that are readily available for the cam module. One of the successful tests shown below is a web server that tracks the features of an individual's face and can save them for authorization later when they appear on the surveillance once again.

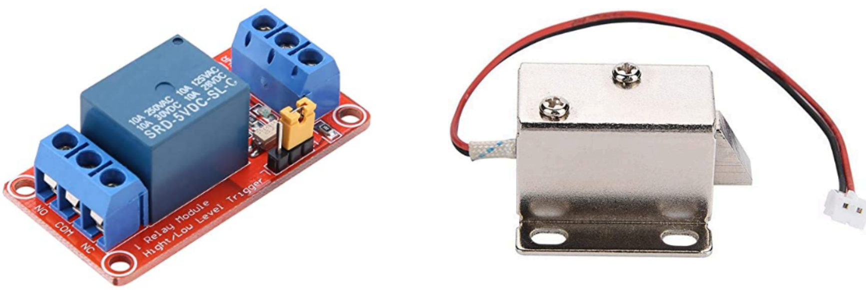


Figure 31. A 5V Relay module (left) and a 12V solenoid electronic lock (right).

For the construction of an automatic door lock, firstly a relay circuit is developed while being paired with an external 12 V battery and a 12V solenoid lock. A 5V relay module is paired in between the lock and the battery power supply with the connections made on Normally Open (NO) Input. With this connection the door lock circuit remains open until a signal is provided through the esp32 cam which closes the circuit and energizes the lock. This will last for 5 seconds until the door locks again. The timer can be modified in the Arduino IDE code uploaded into the esp32 cam.

The web streaming library code which was demonstrated in the last semester for the testing of the Esp32 cam is modified with a ne wer library which allows the detection and saving of faces. Once a saved face is recognized, the esp32 cam will output a message that the door is unlocked in the serial monitor and send a signal to the relay module to energize the lock.

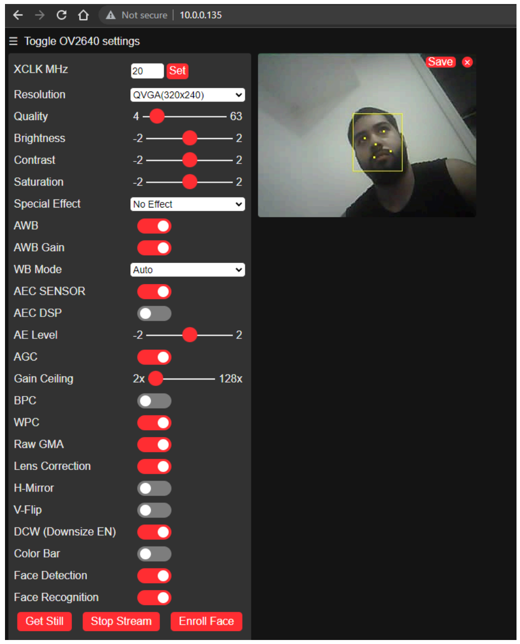


Figure 32. Simple Camera web server with face detection/recognition capabilities (Fall term).

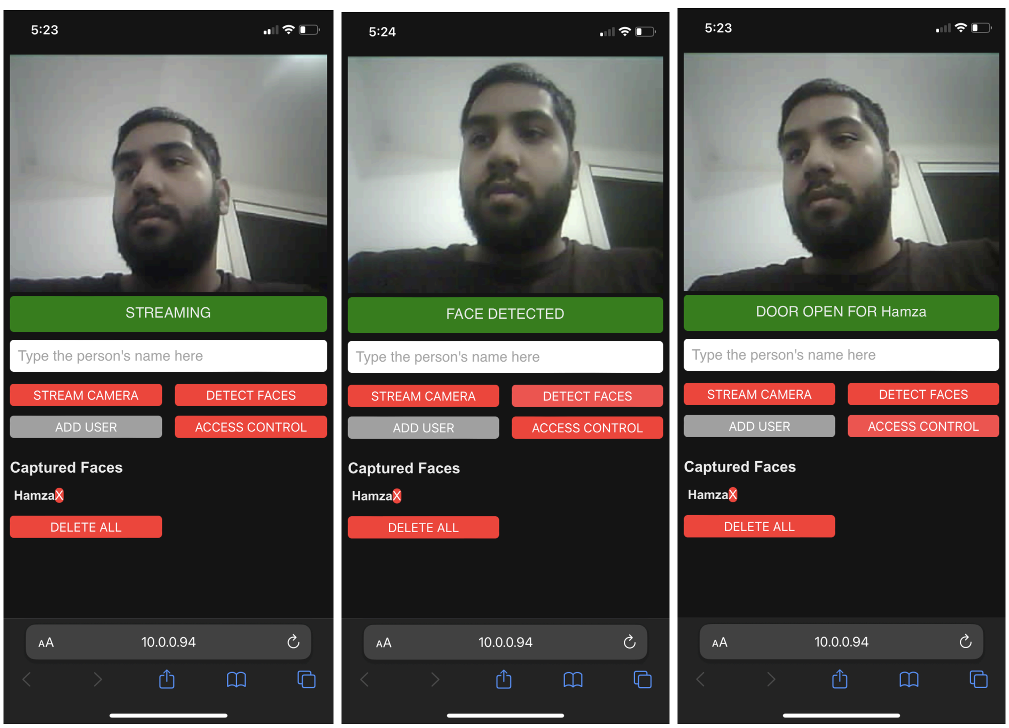


Figure 33.Modified library web stream with streaming/detection.

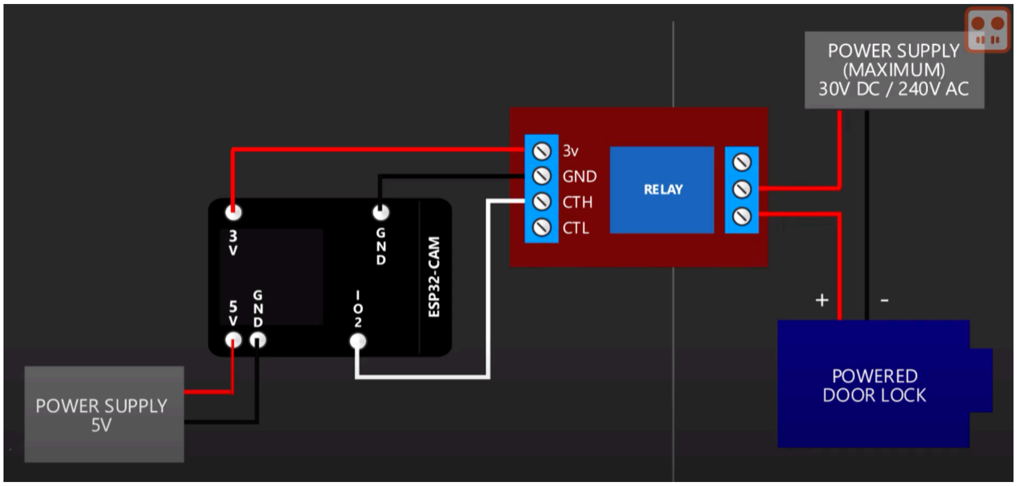


Figure 34. Prototype circuit for esp32-cam with door lock system

Figure 13 depicts the Esp-32 cam connections with the door lock and relay module. A 5v relay module is being tested and will later be developed to a 3V relay module for a simpler circuit which is depicted above. The 5V power supply connections for the Esp-32 cam will be supplied by the Arduino Uno board in the IR sensor circuit above.

Final Face Recognition Door Lock Circuit

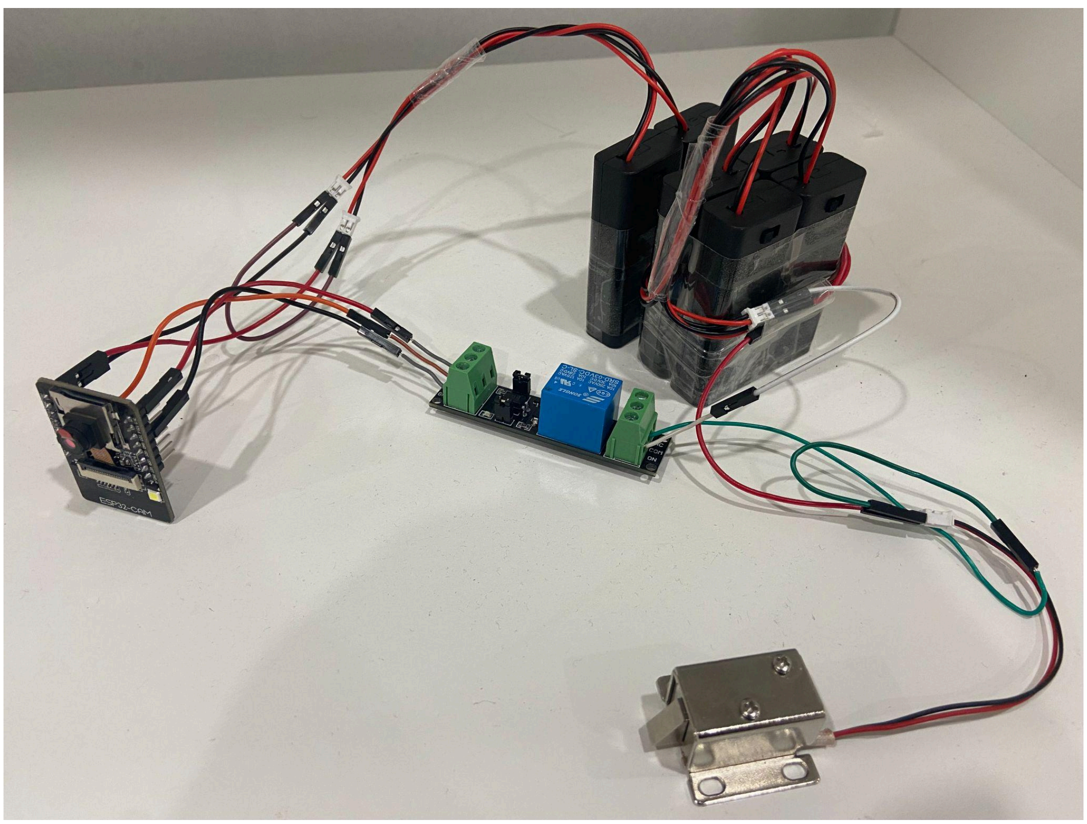


Figure 35. Final circuit for esp32-cam with door lock system

Final Web Server Development

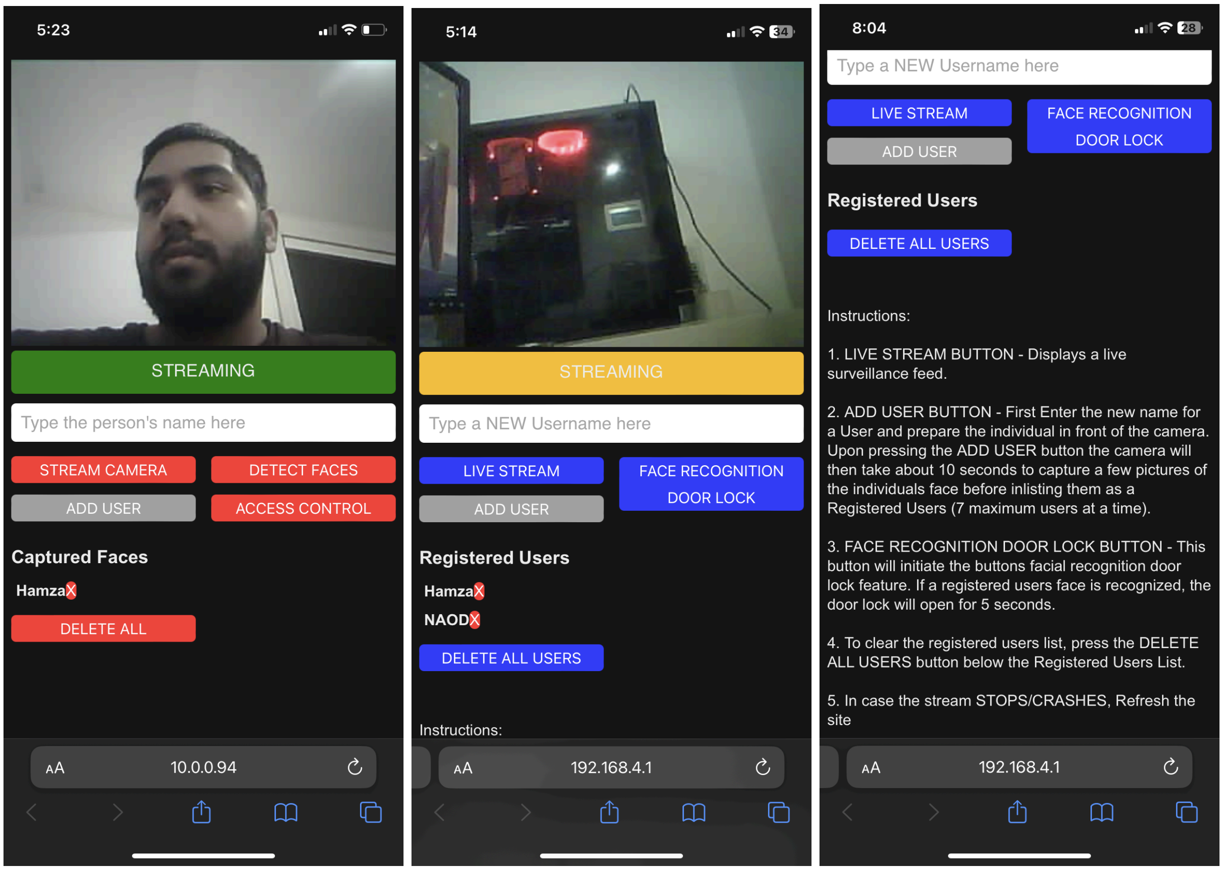


Figure 36. Final web server (two on the right) with original server for comparison(left).

```
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  app_httpserver_init();
  app_facenet_main();
  socket_server.listen(82);

  Serial.print("Camera Ready! Use 'http://");
  Serial.print(WiFi.localIP());
  Serial.println("' to connect");
}
```

Figure 37. Original Web Server private connection to router SSID and password.

```
WiFi.softAP(ssid, password);

app_httpserver_init();
app_facenet_main();
socket_server.listen(82);
```

Figure 38. Final Web Server local/public connection to network through Access Point connection using arduino softAP method.

The final design for the door lock facial recognition circuit consists of the three main components discussed along with switch batteries and minimal wiring. The esp32 cam board is connected to the 3V relay module through the IO2 and 3V pins. The relay module is constructed in series with the electronic door lock which is Normally Open (NO). The Cam board and door lock are both powered by DC switch batteries that can be manually switched on or off. The final circuit can be viewed in figure 16. The camera web server is modified to utilize the softAP method which turns the unit into an access point connection utilizing the network of the device to which it connects. The code change can be seen in figures 18 and 19. The final HTML of the web server is modified to include detailed information on the buttons and how to operate the server. The colors of the website are also altered to show TMU colors and the javascript is modified to remove a redundant button which can be seen in figure 17.

Data Storage And Website:

To this project, a MySQL database and website were designed to store and display the necessary data. After evaluating various options, including MongoDB and Oracle, it was decided that the product would use MySQL as its database to store data. Being an open-source relational database management system (RDBMS) supported by Oracle, MySQL was selected due to its availability, cost-effectiveness, vast online resources, scalability, and robust security features. MySQL implements multiple security layers, such as conditional access, database auditing, and encryption, to ensure the protection of sensitive data.

Data can be protected through encryption by obscuring the container it is stored in or by encrypting the data itself. This will help secure our data and access to it. To connect an Arduino with MySQL involves connecting the Arduino indirectly to the MySQL server through HTTP/HTTPS. As the arduino itself is not capable of using http request, we implemented ESP32 to utilize its http request functionality. The ESP32 accesses the data gained from the PIR sensor continuously and uploads it to out database.

The process of connecting an Arduino to MySQL via HTTP/HTTPS is as follows:

- The Arduino sends an HTTP request to the web server.
- The web server runs a PHP script.
- The PHP script receives the HTTP request and interacts with the MySQL server.
- The PHP script retrieves the data from the server and returns it to the Arduino through an HTTP response.

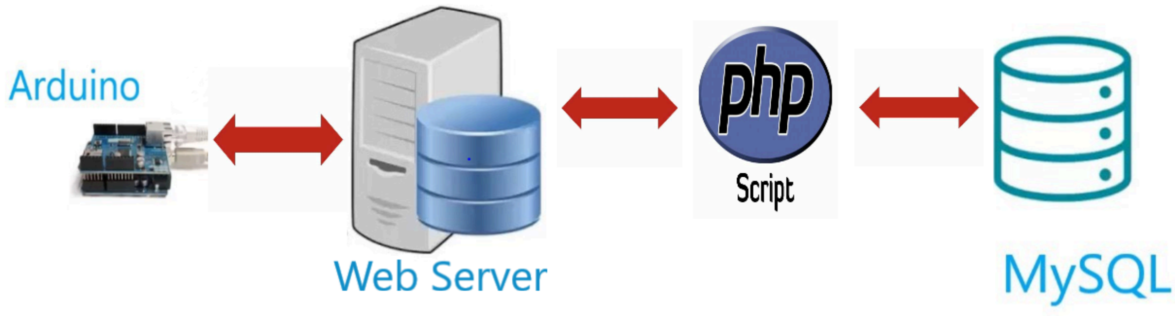


Figure 39. Arduino indirect connection to MySQL via HTTP/HTTPS

Using Mysql, the data from the infrared sensor will be stored. The infrared sensor will store every time it is activated when a person is detected within the distance of the sensor. The arduino code also checks for how long the light application has been on and calculates the power used. This is then stored in Mysql along with the light state and a timestamp to show when the sensor was activated. An example of our MySQL database design that stores both the state, time, and powerused can be seen below.

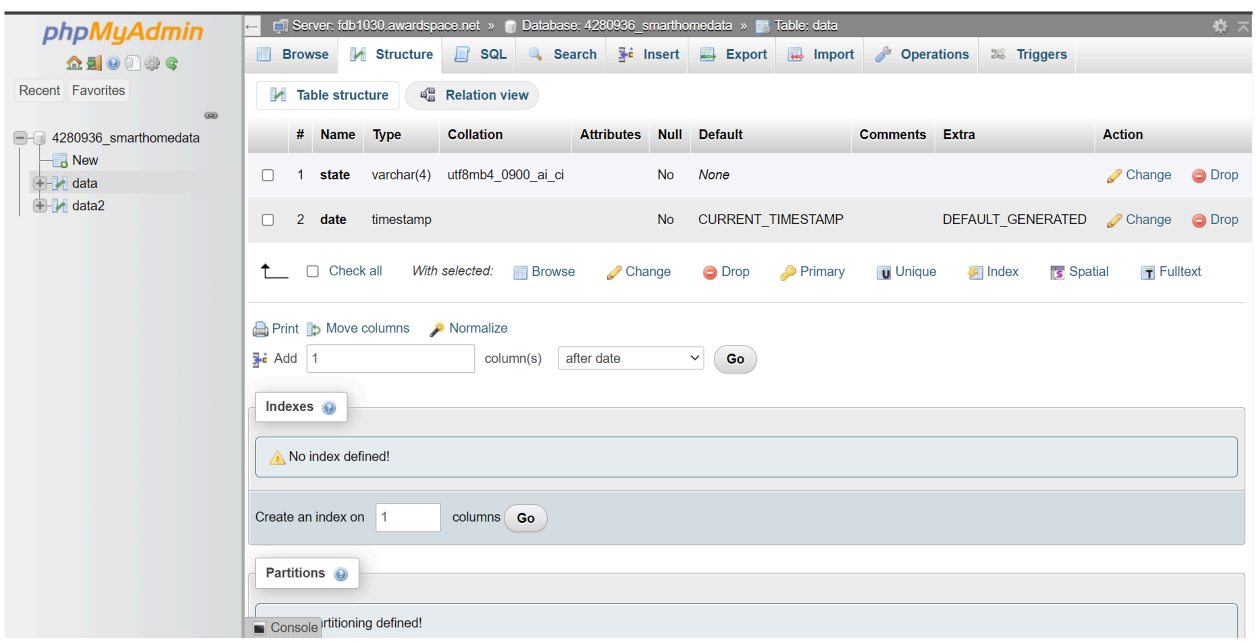


Figure 40. MySQL database table to store state of the Infrared sensor

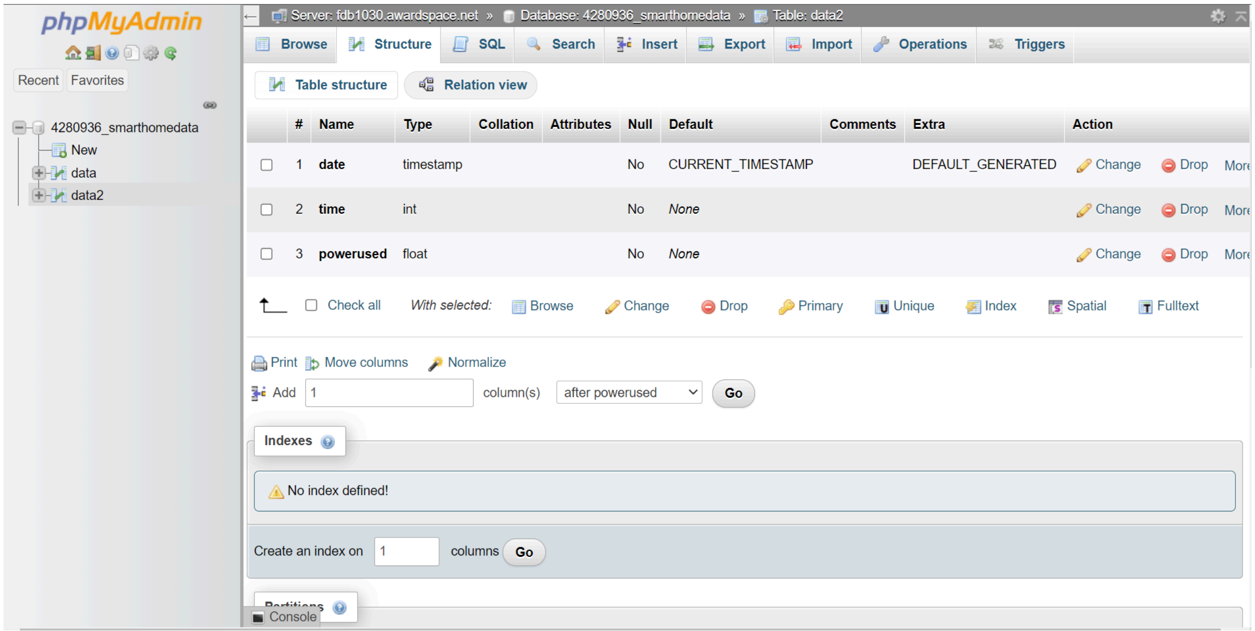


Figure 41. MySQL database table to store time on and power used of the Infrared sensor

```php
<?php

function OpenCon()

{

        $con=mysqli_connect("fdb1030.awardspace.net",
"4280936_smarthomedata", "Hiwet#2000", "4280936_smarthomedata",
"3306");

        return $con;

 }

function CloseCon($con)

{

        $con -> close();

}

?>
```

Figure 42. PHP script for connecting MySQL, Arduino (awardspace hosting)

The PHP script above defines two functions, OpenCon() and CloseCon(), for opening and closing a connection to a MySQL database.

The OpenCon() function uses the mysqli_connect() function to create a connection to a MySQL database. The function takes four arguments:

- The hostname of the database server (in this case, "fdb1030.awardspace.net").
- The username to use for the connection ("4280936_smarthomedata").
- The password to use for the connection ("Hiwet#2000").
- The name of the database to use ("4280936_smarthomedata").

It also includes the port number for the MySQL database as an optional 5th argument.

The CloseCon() function takes a single argument, which is the database connection object, and uses the close() method to close the connection

```
<?php

include 'db_connection.php';

$con= OpenCon();

$k = $_GET["key"];

$state = $_GET["state"];

if ($k =="smarthomesystemproject")

{

        $sql ="INSERT INTO data (state) VALUES ('$state')";

        mysqli_query($con, $sql);

        echo "<br>Done!";

}

CloseCon($con);

?>
```

Figure 43. PHP script for MySQL with Arduino storing state send

```php
<?php

include 'db_connection.php';

$con= OpenCon();

$k = $_GET["key"];

$time = $_GET["time"];

$powerused = $_GET["powerused"];

if ($k =="smarthomesystemproject")

{

	$sql ="INSERT INTO data2 (time,powerused) VALUES ('$time','$powerused')";

	mysqli_query($con, $sql);

	echo "<br>Done!";

}

CloseCon($con);

?>
```

Figure 44. PHP script for MySQL with Arduino storing time on and power used

The two separate PHP scripts above, each of which inserts data into a MySQL database table using values passed through an HTTP GET request.

The first script expects two parameters to be passed through the HTTP GET request: "key" and "state". If the value of the "key" parameter is equal to "smarthomesystemproject", then the script inserts the value of the "state" parameter into the "state" column of the "data" table. The script then outputs the message "Done!".

The second script also expects three parameters to be passed through the HTTP GET request: "key", "time", and "powerused". If the value of the "key" parameter is equal to "smarthomesystemproject", then the script inserts the values of the "time" and "powerused" parameters into the "time" and "powerused" columns of the "data2" table. The script then outputs the message "Done!".

Both scripts include the db_connection.php file, which defines the OpenCon() and CloseCon() functions that create and close the database connection. Then they call the OpenCon() function to create the database connection.

These scripts can be called from an ESP32 microcontroller using an HTTP GET request to send data to the MySQL database. For example, the ESP32 can send a request like:

http://example.com/insert_data.php?key=smarthomesystemproject&state=on to insert the value "on" into the "state" column of the "data" table, and a request like:

http://example.com/insert_data2.php?key=smarthomesystemproject&time=12:00&powerused=500 to insert the values "12:00" and "500" into the "time" and "powerused" columns of the "data2" table

```
#include <WiFi.h>

#include <HTTPClient.h>

#define RXp2 16

#define TXp2 17

#define WIFI_SSID "Dan iPhone"

#define WIFI_PASSWORD "hello123"


String prestate;

String state="OFF";

int var1=1;

int count=10;

unsigned long counttime=0;

float powerused=0;

unsigned long StartTime;


void setup() {
```

```
  Serial.begin(115200);
  Serial2.begin(9600, SERIAL_8N1, RXp2, TXp2);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.println("starting");
}
bool isConnected = false;
void loop() {
  if (WiFi.status() == WL_CONNECTED && !isConnected) {
    Serial.println("Connected");
    isConnected = true;
  }
  if (WiFi.status() != WL_CONNECTED) {
    Serial.println(".");
    isConnected = false;
  }
  Serial.println("Message Received: ");
  var1= Serial2.parseInt();
  count= Serial2.parseInt();
  Serial.println("state:"+String(var1));
  Serial.println("time:"+String(count));
  powerused = (count/60)*5;
   if(var1 == 0){
     state = "OFF";
```

```
    }
  if(var1==1){
    state = "ON";
    StartTime = millis();
    counttime += (millis()-StartTime);
    Serial.println("counter"+String(counttime));
    }
 if(state != prestate){
      sendFirstRequest();
      sendSecondRequest();
 }
 else{
      sendSecondRequest();
  }
  delay(60000);
  prestate = state;
}
void sendFirstRequest() {
String  url       = "http://smarthomesystemproject.atwebpages.com/insert_data.php?key=smarthomesystemproject&state=ON";
 HTTPClient http;
 Serial.print("Sending first request to ");
 Serial.println(url);
```

```
  // Send the HTTP GET request

  http.begin(url);

  int httpResponseCode = http.GET();

  // Check for errors

  if (httpResponseCode > 0) {

    Serial.print("First request success, response code: ");

    Serial.println(httpResponseCode);

  } else {

    Serial.print("First request failed, error code: ");

    Serial.println(httpResponseCode);

  }

  // Free resources

  http.end();

}

void sendSecondRequest() {

String  url      =
"http://smarthomesystemproject.atwebpages.com/insert_data2.php?key=smarthomesystemproject&time="+String(count)+"&powerused="+String(powerused);

  HTTPClient http;

  Serial.print("Sending second request to ");

  Serial.println(url);

  // Send the HTTP GET request

  http.begin(url);

  int httpResponseCode = http.GET();
```

```
// Check for errors
if (httpResponseCode > 0) {
  Serial.print("Second request success, response code: ");
  Serial.println(httpResponseCode);
} else {
  Serial.print("Second request failed, error code: ");
  Serial.println(httpResponseCode);
}
// Free resources
http.end();
}
```

Figure 45. Arduino ESP32 script for data http request to MySQL

The above code is an Arduino sketch that communicates with a web server using HTTP requests. It uses the WiFi and HTTPClient libraries to establish a connection to a web server and send data to two separate PHP scripts hosted on the server.

The sketch reads input data from a sensor connected to a serial port and uses it to determine the state (on/off) and calculate the power usage of a smart home device. If the state of the device changes, the sketch sends an HTTP request to the server to insert the state data into a MySQL database using the insert_data.php script.

The sketch also sends an HTTP request to insert the time and power usage data into another table of the same database using the insert_data2.php script. This data is also calculated from the sensor input.

The sketch includes functions to send HTTP GET requests to the server using the HTTPClient library. These functions take a URL string as a parameter and use it to send GET requests to the server.

Following this, the website accesses the database in MySQL and displays the state, time on, and powerused. This website html code, along with all the necessary php code was hosted on Awardspace.com. AwardSpace is a web hosting company that provides free and paid web hosting services to individuals and businesses.

```
<!DOCTYPE html>
<html>
<head>
    <title>Smart Home System</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f7f7f7;
            display: flex;
            flex-direction: column;
            align-items: center;
            justify-content: center;
            height: 100vh;
            margin: 0;
            margin-left: 400px;
        }
        h1 {
            background-color: #0077b6;
            color: #fff;
            padding: 20px;
            margin: 0;
            text-align: center;
```

```
        width: 100%;
        position: absolute;
        top: 0;
        left: 0;
        border-bottom: 1px solid #004365;
}
h2 {
        color: #004365;
        margin-top: 30px;
        margin-right: 400px;
        font-size: 24px;
        font-weight: bold;
        text-align: center;

}
.container {
        display: flex;
        flex-direction: column;
        align-items: center;
        margin-top: 30px;
        background-color: #fff;
        padding: 10px;
        box-shadow: 0px 0px 0px 0px rgba(0,0,0,0.2);
```

```
        border-radius: 5px;
}
table {
        border-collapse: collapse;
        width: 100%;
        margin-top: 20px;
        margin-left:100px;
}
th, td {
        padding: 12px;
        text-align: left;
        border-bottom: 1px solid #ddd;
}
tr:hover {
        background-color: #f5f5f5;
}
table.powerused {
        border-collapse: collapse;
        margin-top: 20px;
        margin-right: 100px;
}
table.powerused th, table.powerused td {
        padding: 12px;
```

```
                text-align: left;
                border-bottom: 1px solid #ddd;
            }
            table.powerused tr:hover {
                background-color: #f5f5f5;
            }
            button {
                background-color: #004365;
                color: #fff;
                padding: 12px 20px;
                border: none;
                border-radius: 5px;
                font-size: 16px;
                cursor: pointer;
                position: absolute;
                top: 5%;
                left: 10%;
                transform: translate(-50%, -50%);
            }
        </style>
    </head>
    <body>
        <h1>Smart Home System Data</h1>
```

```
<button onclick="location.reload();">Refresh</button>

<img src="image.jpg" alt="Smart Home System Image" style="display: block; margin: 550px 400px 0px 0px;">

<?php

    // Establish a connection to the MySQL database

    $servername = "fdb1030.awardspace.net";

    $username = "4280936_smarthomedata";

    $password = "Hiwet#2000";

    $dbname = "4280936_smarthomedata";

    $conn = new mysqli($servername, $username, $password, $dbname);

    if ($conn->connect_error) {

      die("Connection failed: " . $conn->connect_error);

    }

    // Retrieve data from the MySQL database

    $sql = "SELECT * FROM data ORDER BY date DESC LIMIT 10";

    $result = $conn->query($sql);


    // Display the data in an HTML table

    if ($result->num_rows > 0) {

        echo "<h2>Light State</h2>";

        echo "<table>";

        echo "<tr><th>state</th><th>date</th></tr>";

      while($row = $result->fetch_assoc()) {
```

```
            echo "<tr><td>" . $row["state"] . "</td><td>" .
$row["date"] . "</td></tr>";
         }
         echo "</table>";
      } else {
         echo "No data found.";
      }
      // Retrieve power usage data from the MySQL database
      $sql = "SELECT * FROM data2 ORDER BY date DESC
LIMIT 1";
      $result = $conn->query($sql);
      // Display the power usage data in an HTML table
      if ($result->num_rows > 0) {
            echo "<h2>Power Usage and time ON</h2>";
            echo "<table class ='powerused'>";
            echo "<tr><th>time ON(Min)</th><th>Power
Used(W)</th></tr>";
            while($row = $result->fetch_assoc()) {
                  echo "<tr><td>" . $row["time"] .
"</td><td>" . $row["powerused"] . "</td></tr>";
            }
            echo "</table>";
      } else {
            echo "No power usage data found.";
      }
```

```
$conn->close();
?>
</body>
</html>
```

Figure 46. HTMLwebsite script

This is an HTML file that displays data from a MySQL database about a Smart Home System. It shows two tables, one with the light state and the date it was recorded, and the other with power usage and time the device was on. It also includes a button to refresh the page and an image of the Smart Home System. The PHP code establishes a connection to the database, retrieves the data, and displays it in the HTML tables.

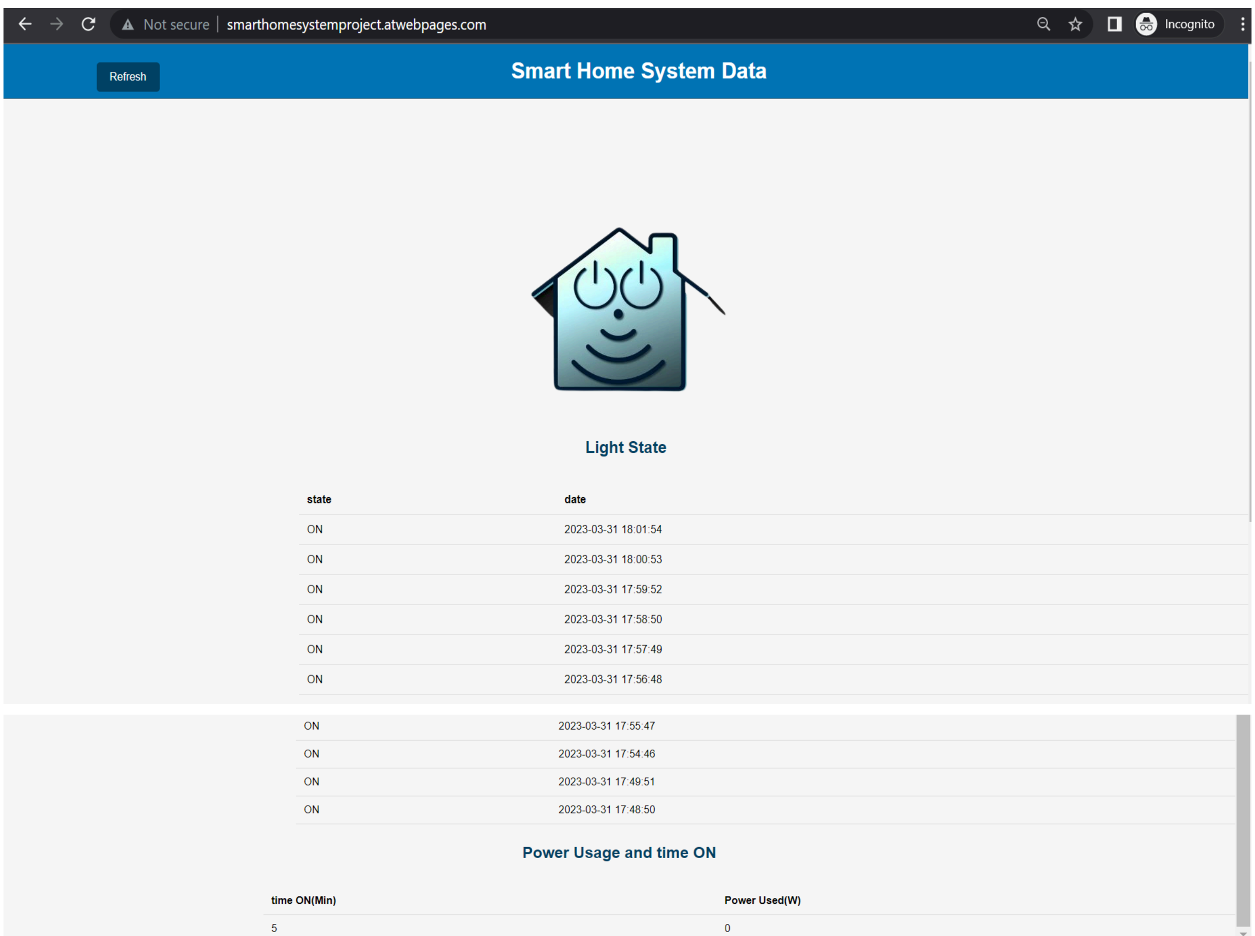


Figure 47. Website display

Design Analysis and Approach

In conclusion, the project has been completed successfully as all the required functions, including the implementation of an infrared sensor, face recognition, and website, have been accomplished. Our team utilized the abundance of library programs available with the esp32 cam and conducted extensive research to ensure the project's success. Despite the complexity of combining the different modules, we were able to complete the project within the allocated time and resources.

Reference:

1. "Arduino - MySQL | Arduino Tutorial." https://arduinogetstarted.com/tutorials/arduino-mysql (accessed: Dec. 02, 2022).
2. Phillips, P., Grother, P., Micheals, R., Blackburn, D., Tabassi, E. & Bone, M. (2003b), Face Recognition Vendor Test 2002 - Overview and Summary, Technical Report NISTIR 6965, DoD Counterdrug Technology Development Program Office, Virginia, United States.
3. "Iris recognition algorithms," *Encyclopedia of Biometrics*, pp. 819–819, 2009.
4. "Iris recognition immigration system (IRIS)," *Encyclopedia of Biometrics*, pp. 825–825, 2009.
5. "Figure 2: Iris recognition system." doi: 10.7717/peerj-cs.919/fig-2.
6. Bonsor, K., Johnson, R. "How Facial Recognition Systems Work." *How Stuff Works*. Retrieved from https://electronics.howstuffworks.com/gadgets/high-tech-gadgets/facial-recognition.htm, Nov. 23, 2019.
7. A. Karim1 and J. Y. Andersson1, "IOPscience," *IOP Conference Series: Materials Science and Engineering*, 16-Dec-2013. [Online]. Available: https://iopscience.iop.org/article/10.1088/1757-899X/51/1/012001. [Accessed: 02-Dec-2022].
8. "Advantages of fresnel lenses: Edmund Optics," *Edmund Optics Worldwide*. [Online]. Available:

https://www.edmundoptics.ca/knowledge-center/application-notes/optics/advantages-of-fresnel-lenses. [Accessed: 10-Feb-2023].

9. E. Notes, “Transistor Darlington Pair,” *Electronics Notes*. [Online]. Available: https://www.electronics-notes.com/articles/analogue_circuits/transistor/darlington-pair.php#:~:text=The%20Darlington%20Pair%20offers%20a,compared%20to%20a%20single%20transistor. [Accessed: 10-Feb-2023].
10. E. Pena and M. G. Legaspi, “UART: A hardware communication protocol understanding universal asynchronous receiver/transmitter,” *UART: A Hardware Communication Protocol Understanding Universal Asynchronous Receiver/Transmitter | Analog Devices*. [Online]. Available: https://www.analog.com/en/analog-dialogue/articles/uart-a-hardware-communication-protocol.html. [Accessed: 01-Apr-2023].