# Home Security
# (Pratical Group Assignment)

SWINBURNE UNIVERSITY OF TECHNOLOGY

## UNIT DETAILS

| | | | | | | |
|---|---|---|---|---|---|---|
| Unit name | IoT Programming | | | Class day/time | Thu 01-05pm | Office use only |
| Unit code | SWE30011 | Assignment no. | 03 | Due date | 4th August | |
| Name of lecturer/teacher | Khoa Nguyen Dang | | | | | |
| Tutor/marker's name | Khoa Nguyen Dang | | | | | Faculty or school date stamp |

## STUDENT(S)

| | Family Name(s) | Given Name(s) | Student ID Number(s) |
|---|---|---|---|
| (1) | Ta | Quang Tung | 104222196 |
| (2) | Doan | Minh Hieu | 104168106 |
| (3) | | | |
| (4) | | | |
| (5) | | | |
| (6) | | | |

## DECLARATION AND STATEMENT OF AUTHORSHIP

1. I/we have not impersonated, or allowed myself/ourselves to be impersonated by any person for the purposes of this assessment.
2. This assessment is my/our original work and no part of it has been copied from any other source except where due acknowledgement is made.
3. No part of this assessment has been written for me/us by any other person except where such collaboration has been authorised by the lecturer/teacher concerned.
4. I/we have not previously submitted this work for this or any other course/unit.
5. I/we give permission for my/our assessment response to be reproduced, communicated, compared and archived for plagiarism detection, benchmarking or educational purposes.

I/we understand that:

6. Plagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offence that may lead to exclusion from the University. Plagiarised material can be drawn from, and presented in, written, graphic and visual form, including electronic data and oral presentations. Plagiarism occurs when the origin of the material used is not appropriately cited.

**Student signature/s**

I/we declare that I/we have read and understood the declaration and statement of authorship.

| | | | |
|---|---|---|---|
| (1) | | (4) | |
| (2) | | (5) | |
| (3) | | (6) | |

## Table Of Contents

### Contents

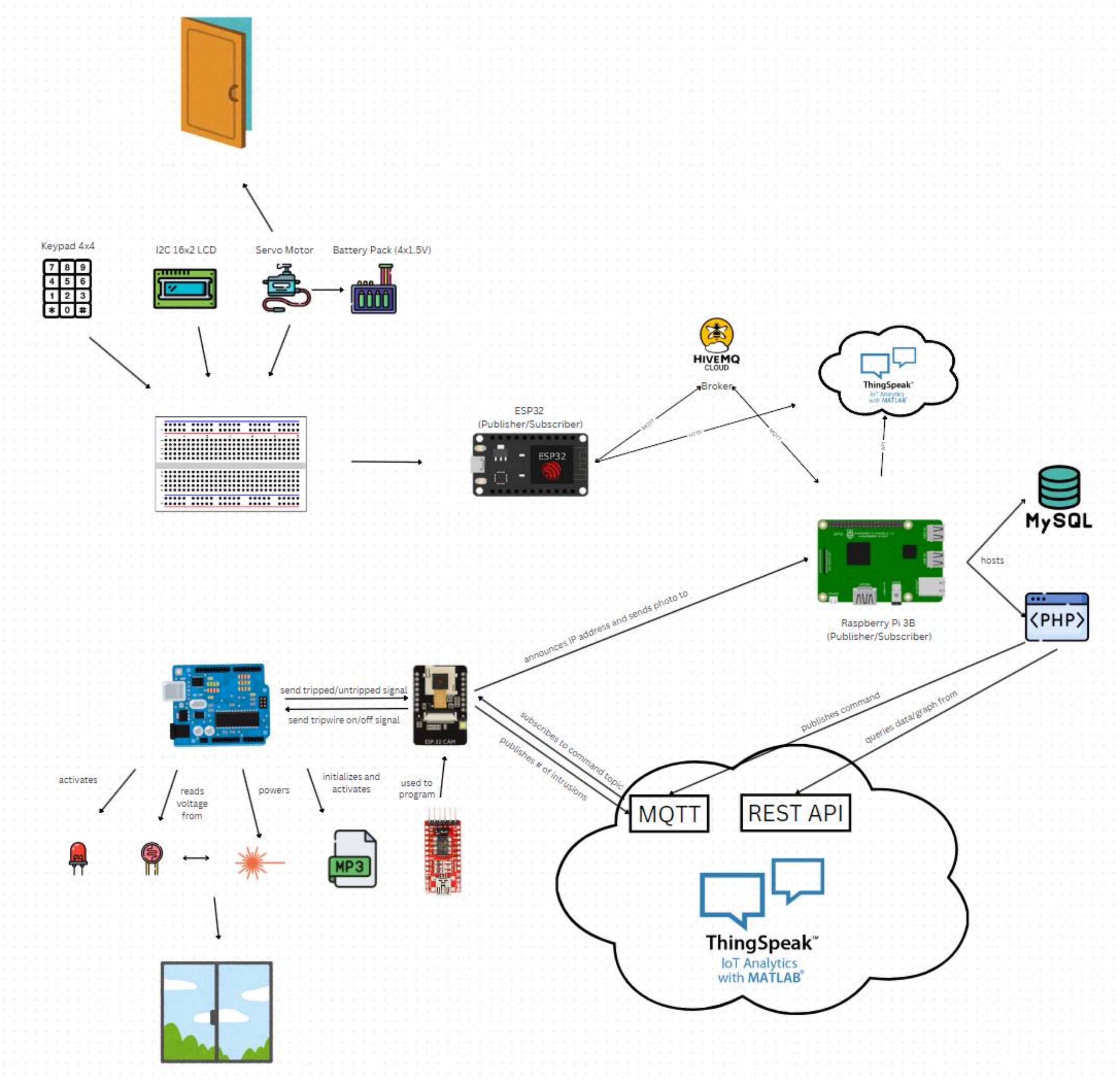
## Introduction

To ensure the protection of both property and occupants, home security is an essential component of modern living. Our project, which focuses on a smart door and a secure window, intends to improve home security through the development of smart systems.

To regulate entry, the smart door system combines an ESP32 microprocessor, servo motor, LCD display, and keypad. Passwords entered through the keypad are verified, and feedback is displayed on the LCD. A Raspberry Pi that maintains the password database and records access attempts is in communication with the system. It also makes use of ThingSpeak for visualizing failed password attempts and MQTT for communication.

On the other hand, the secure window utilizes a laser tripwire to detect when someone breaks into the house through the window. When this happens, the tripwire activates a red LED, plays a dog barking sound with the MP3 module, and triggers an ESP32 camera. The camera will send the photos taken to a web server hosted on the Raspberry Pi for future viewing and publish the intrusion status to an MQTT broker hosted on the cloud.

## Conceptual Design Of The Project



Keypad 4x4
I2C 16x2 LCD
Servo Motor
Battery Pack (4x1.5V)
ESP32 (Publisher/Subscriber)
ESP32
HIVEMQ CLOUD
Broker
MQTT
HTTP
MQTT
API
ThingSpeak IoT Analytics with MATLAB
MySQL
hosts
<PHP>
Raspberry Pi 3B (Publisher/Subscriber)
announces IP address and sends photo to
send tripped/untripped signal
send tripwire on/off signal
ESP-32 CAM
subscribes to command topic
publishes # of intrusions
publishes command
queries data/graph from
activates
reads voltage from
powers
initializes and activates
used to program
MP3
MQTT
REST API
ThingSpeak IoT Analytics with MATLAB

## Tasks Breakdown

| Task | Responsible Team Member |
|---|---|
| *Wiring the Smart Door circuit (Keypad Integration, LCD Display Setup, Servo Motor Control, ESP32 Programming)* | Doan Minh Hieu |
| *Wiring the Secure Window circuit (LED, Laser Tripwire, MP3 Module, ESP32 Camera, Arduino)* | Ta Quang Tung |
| *Setting up Raspberry Pi for web server deployment and script execution* | Both |
| *PHP Web Server Development* | Both |
| *HiveMQ Configuration and Integration* | Doan Minh Hieu |
| *ThingSpeak Configuration and Integration* | Both |
| *Testing and Debugging* | Both |
| *Documentation and Reporting* | Both |

## Implementation

### A. *Sensing System and Actuators:*

1. Smart Door:

The system comprises various components and their interactions to ensure secure access control of the smart door. Its sensing system consists of an LCD display (I2C 16x2) and a keypad (4x4). Users can input a password into the keypad, and the ESP32 will handle it from there. The password is input and shown on the LCD in a secret format (****), giving the user immediate response.

2. Secure Window:

The secure window's sensing system consists of a laser emitter on one side and a photoresistor on the other side. The emitter will fire a continuous laser beam directly on the photoresistor's detection surface. The two components are installed such that the laser is parallel to the window. In its untripped state, the photoresistor's resistance will be very low. However, when an object obstructs the laser beam (either the window is opened or someone enters through the window), the beam will break, causing the photoresistor's resistance to increase. The photoresistor is connected to an Arduino, which constantly reads the voltage running through it. When the tripwire is tripped, this voltage will drop drastically and the Arduino will activate the actuators.

The secure window features 3 actuators: a red LED, an MP3 module which has been configured to play only a dog barking sound, and an ESP32 camera. When the tripwire is tripped, the Arduino will turn on the LED, MP3 module, and send a HIGH voltage to the ESP32, informing it to take a photo of the window. The Arduino then goes on a 15-second rest period to prevent accidentally triggering the other components too frequently.

### B. *Edge Servers*

1. Smart Door:

The central controller for this smart door system is the ESP32 Dev Kit V1. It accepts keypad input from the user, manages the LCD display and servo motor, and uses MQTT to connect with the Raspberry Pi. The Raspberry Pi sends the ESP32 the current password and servo position, and the ESP32 processes user inputs accordingly. Additionally, access logs are sent back to the Raspberry Pi for processing and archiving.

The Raspberry Pi 3B in this system hosts the MySQL database and PHP web server. It manages MQTT communication, publishes the current password and servo position to the ESP32, receives access logs, and updates the database accordingly. Data processing and storage are made efficient thanks to the Raspberry Pi's role as a bridge connecting the ESP32 and cloud services.

2. Secure Window:

The secure window includes 2 edge devices: an ESP32 AI-Thinker Camera and a Raspberry Pi 3B. The ESP32 camera is responsible for taking photos of the window and sending them to the

Raspberry Pi. It also publishes the intrusion status (number of intrusions per minute) to an MQTT broker on ThingSpeak and subscribes to a command topic from the same broker. In addition, the ESP32 camera hosts a small local web server which can be accessed to configure the camera settings.

The Raspberry Pi hosts a local web server which receives upload requests from the camera and allows the user to view and control the secure window's settings. Through this web server, the user can turn the tripwire on/off, view graphs related to the window, and see all photos the camera has taken.

### *C. Communication Protocols*

1. Smart Door:

The ESP32 and Raspberry Pi communicate with each other via MQTT in the smart door system. As the MQTT broker, HiveMQ makes communication easier. The Raspberry Pi provides the ESP32 with updates on the current password and servo position, while the ESP32 publishes access logs. This permits effective control of the door lock system and guarantees real-time synchronization between the devices.

In addition, the ESP32 transmits data to ThingSpeak for logging and visualization via HTTP protocols. The ESP32 allows cloud-based data monitoring and analysis by updating ThingSpeak with information about failed password attempts using HTTP POST requests.

2. Secure Window:

The secure window involves 2 communication protocols: HTTP and MQTT.
HTTP communication takes place between the ESP32 camera and the Raspberry Pi when the camera sends photos to the edge server. HTTP is also used when the web server requests API data from ThingSpeak.

MQTT communication takes place between the ESP32 camera and the ThingSpeak MQTT broker. The camera publishes the intrusion status to the broker every minute and subscribes to the command topic on the broker. MQTT also takes place between the web server and the ThingSpeak broker: the web server can publish commands (turn on/off) to the broker to control the tripwire state.

### *D. API or Website Details*

1. Smart Door:

A web interface for controlling the servo position, monitoring access logs, and managing the door password is provided via the PHP web server running on the Raspberry Pi. This interface allows users to change the servo location and password. In order to retrieve and update the stored values, the web server communicates with the MySQL database. It then uses MQTT to send the updated values to the ESP32.

2. Secure Window:

Another local web server running on the ESP32 camera is built into the camera itself and can be accessed by anyone on the local network to configure the camera settings such as photo resolution or quality.

A local web server hosted on the Raspberry Pi is the main user interface where the user can view the window's information and control the tripwire. It has a link to the ESP32 camera's configuration server, a graph showing recent intrusion status, a list of recent commands, buttons to turn the tripwire on or off, and a photo gallery.

### *E. Cloud Computing*

1. Smart Door:

ThingSpeak, a cloud platform, is used for logging and visualizing incorrect password attempts. The ESP32 sends data on incorrect password attempts to ThingSpeak, which provides a visualization graph for the web interface. Users are able to track unauthorized access attempts and spot possible security problems as a result. A graph of these attempts is displayed via the PHP web interface, which improves the system's security monitoring capabilities and offers insights into access patterns.

2. Secure Window:

ThingSpeak has also been used to store and visualize data on the intrusion status per minute and issued tripwire commands. It features a REST API and an MQTT broker, both of which have been used in this project.

## User Manual

The smart door system offers several user interactions. To enter the password, users need to use the keypad to input their password and press the "*" key to submit it. If the password is correct, the servo motor will unlock the door. If the password is incorrect, the door will remain closed, and the attempt will be logged and visualized on ThingSpeak. Both successful and unsuccessful attempts will also be recorded in the database.

Furthermore, users can use the Raspberry Pi's PHP web interface or any other browser that has its IP address to change the password. After typing a new 4-digit password, send the form. The ESP32 will get the new password and it will be stored in the database.

Another feature for the users is using the slider on the PHP web interface to establish the desired servo position, which may be adjusted from 0 to 180 degrees. The new position will be transmitted to the ESP32 and stored in the database.

There is also a feature to view access history where users check the table displaying the latest 20 access attempts. This table includes the timestamp, door state, and the entered password, providing a detailed log of recent activity. Additionally, the PHP web interface provides a graph from ThingSpeak that displays the frequency and timing of incorrect password submissions, providing information about potentially unwanted access attempts.

Regarding the secure window, to start up this system, power the Arduino, ESP32 camera, and Raspberry Pi by connecting them to a power source. After all devices are powered, wait a few seconds for the ESP32 camera and Raspberry Pi to connect to WiFi and the Arduino to initialize the MP3 module. After the laser lights up, the system is ready to go. It is ready to function without any further configuration.

When the laser tripwire is tripped, you will see the LED light up and hear a dog barking sound. The ESP32 camera will automatically take a photo of the window and send it to the edge server. You can access the web page by going to the link http://<rpi_ip_address>/window.php on your local device, provided that it is on the same local network as the ESP32 camera and the Raspberry Pi.

On the web page, you will see a variety of information. First, you will see the IP address of the camera, which you can visit to configure the camera settings. Second, you will see a graph and list with information related to the window and tripwire status. Below this, you will see two buttons that you can use to turn the tripwire on or off. A tripwire that is turned off effectively disables the system. At the bottom of the page, you will see a link to a gallery page containing every photo the camera has taken, newest photos first.

## Limitations

For the smart door, there are a few points we need to take into consideration:

- Network Dependency: In order for the ESP32, Raspberry Pi, and ThingSpeak to communicate with one another, the system needs a steady WiFi connection.
- Security: Passwords are sent via MQTT, which could not be completely secure in the absence of extra encryption.

The secure window has a number of limitations that can be addressed in the future:

- The edge servers (ESP32 camera and Raspberry Pi) must be connected to the same local network and they are inaccessible outside of it. This is very inconvenient if the user wants to control the window remotely.
- Currently, the ESP32 camera is statically programmed with the SSID and password of the local WiFi. This means that every time the system is moved to a new network, the ESP32 camera has to be reprogrammed again. A better solution is to have an input mechanism that allows the user to type the new SSID and password into the camera.
- This design requires the laser emitter and photoresistor to be perfectly lined up. This can be quite tricky to get right because these components are very small and a slight disturbance can knock them out of place, breaking the setup.

## Appendix

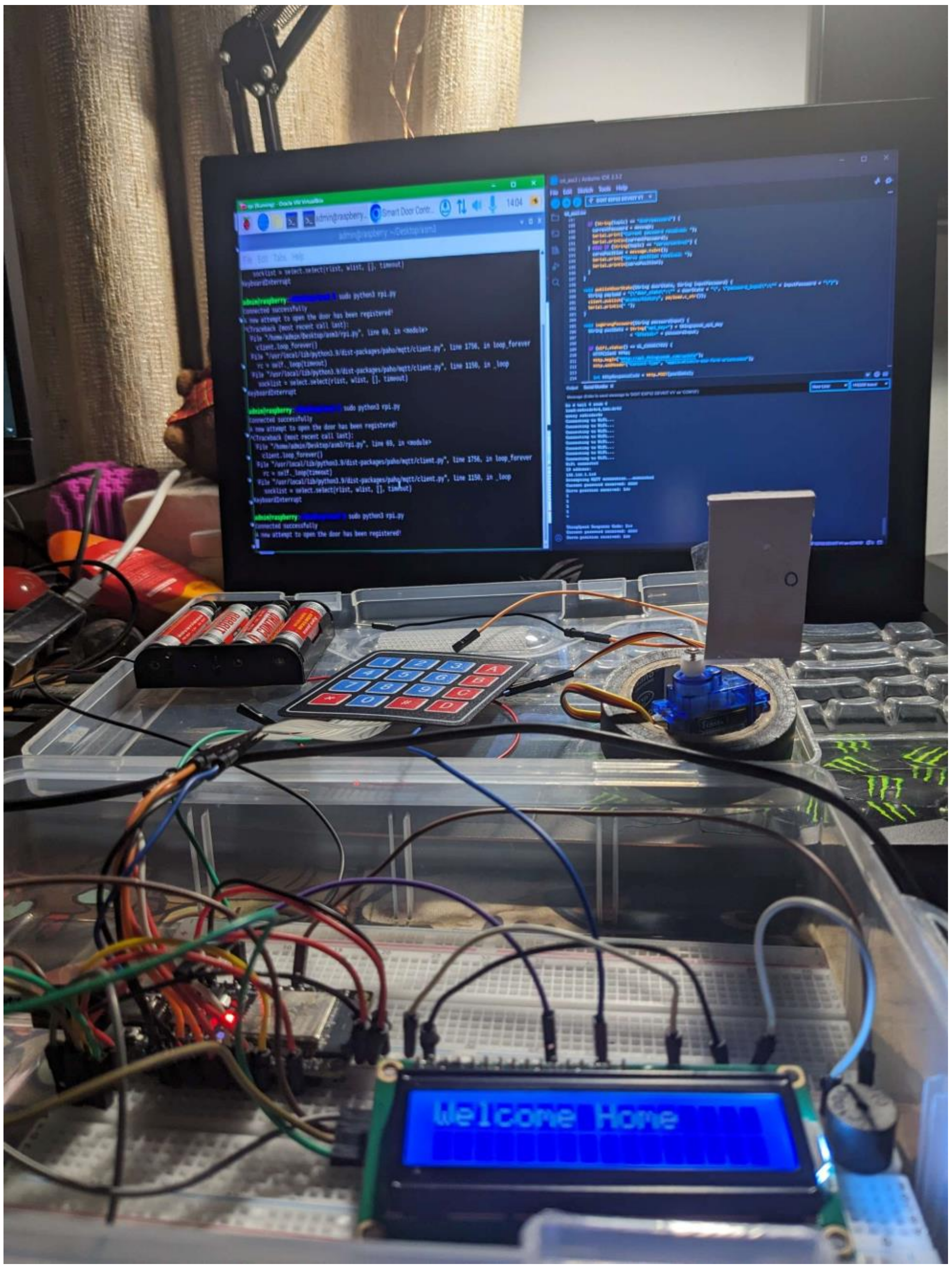
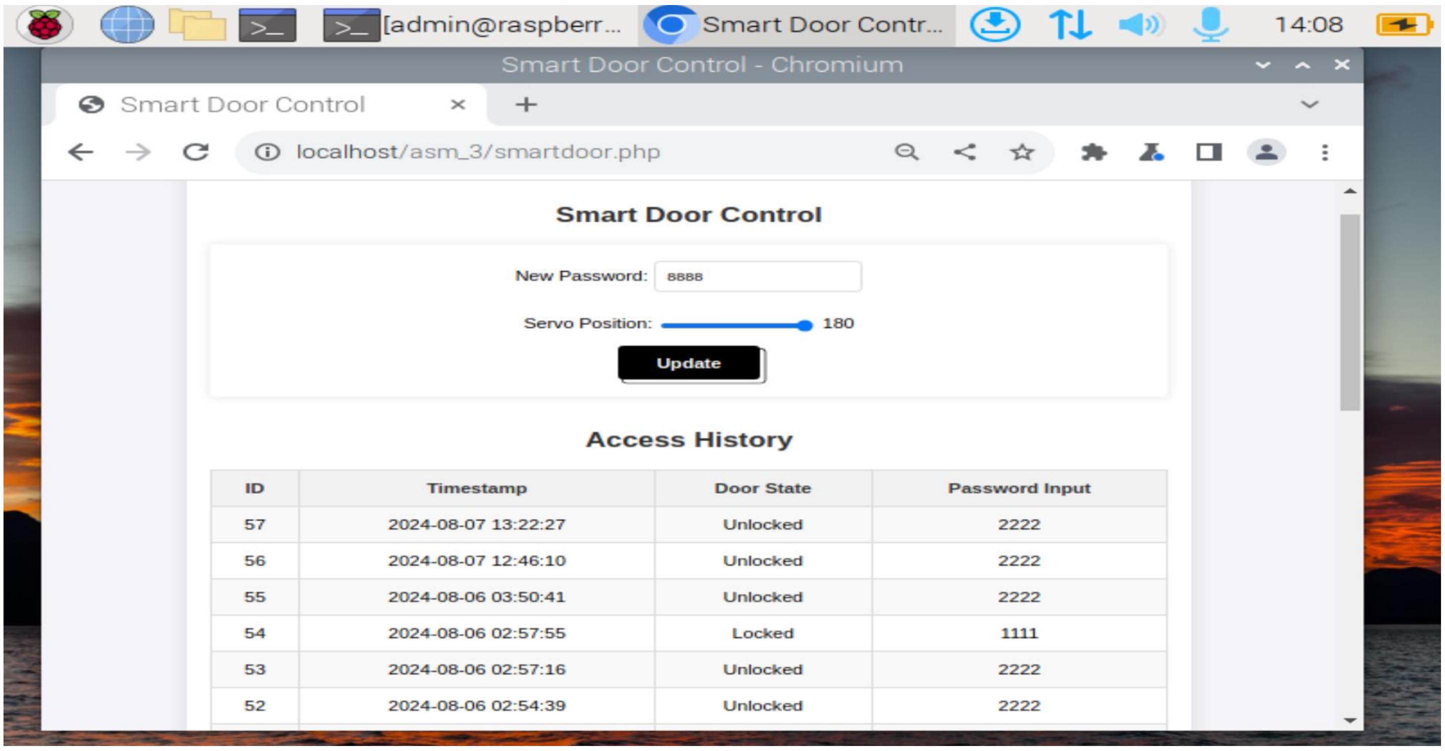


*Fig. 1. Smart Door Circuit*



Smart Door Control

New Password: 8888

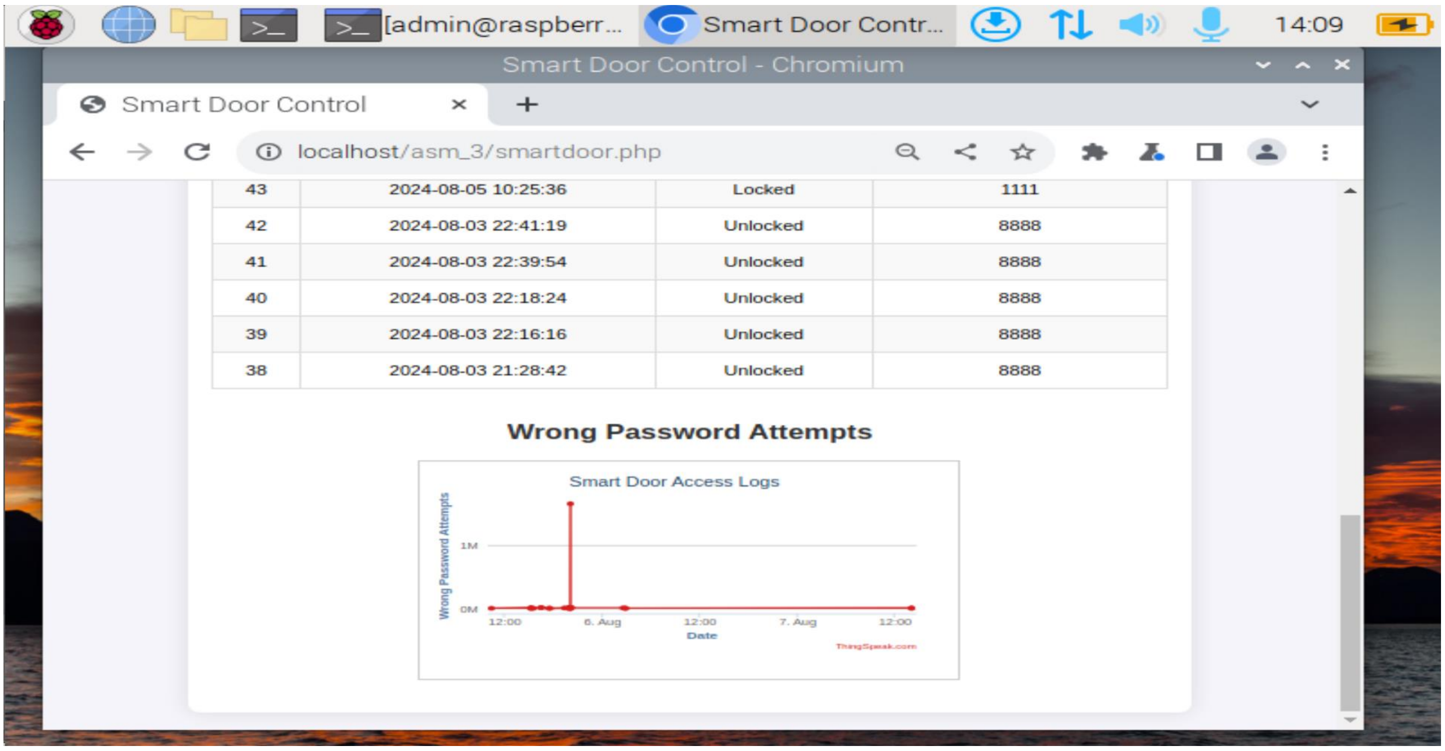
Servo Position: 180

Update

Access History

| ID | Timestamp | Door State | Password Input |
|---|---|---|---|
| 57 | 2024-08-07 13:22:27 | Unlocked | 2222 |
| 56 | 2024-08-07 12:46:10 | Unlocked | 2222 |
| 55 | 2024-08-06 03:50:41 | Unlocked | 2222 |
| 54 | 2024-08-06 02:57:55 | Locked | 1111 |
| 53 | 2024-08-06 02:57:16 | Unlocked | 2222 |
| 52 | 2024-08-06 02:54:39 | Unlocked | 2222 |

*Fig. 2. Smart Door Control Page*



| | | | |
|---|---|---|---|
| 43 | 2024-08-05 10:25:36 | Locked | 1111 |
| 42 | 2024-08-03 22:41:19 | Unlocked | 8888 |
| 41 | 2024-08-03 22:39:54 | Unlocked | 8888 |
| 40 | 2024-08-03 22:18:24 | Unlocked | 8888 |
| 39 | 2024-08-03 22:16:16 | Unlocked | 8888 |
| 38 | 2024-08-03 21:28:42 | Unlocked | 8888 |

Wrong Password Attempts

*Fig. 3. ThingSpeak graph on the webpage above*

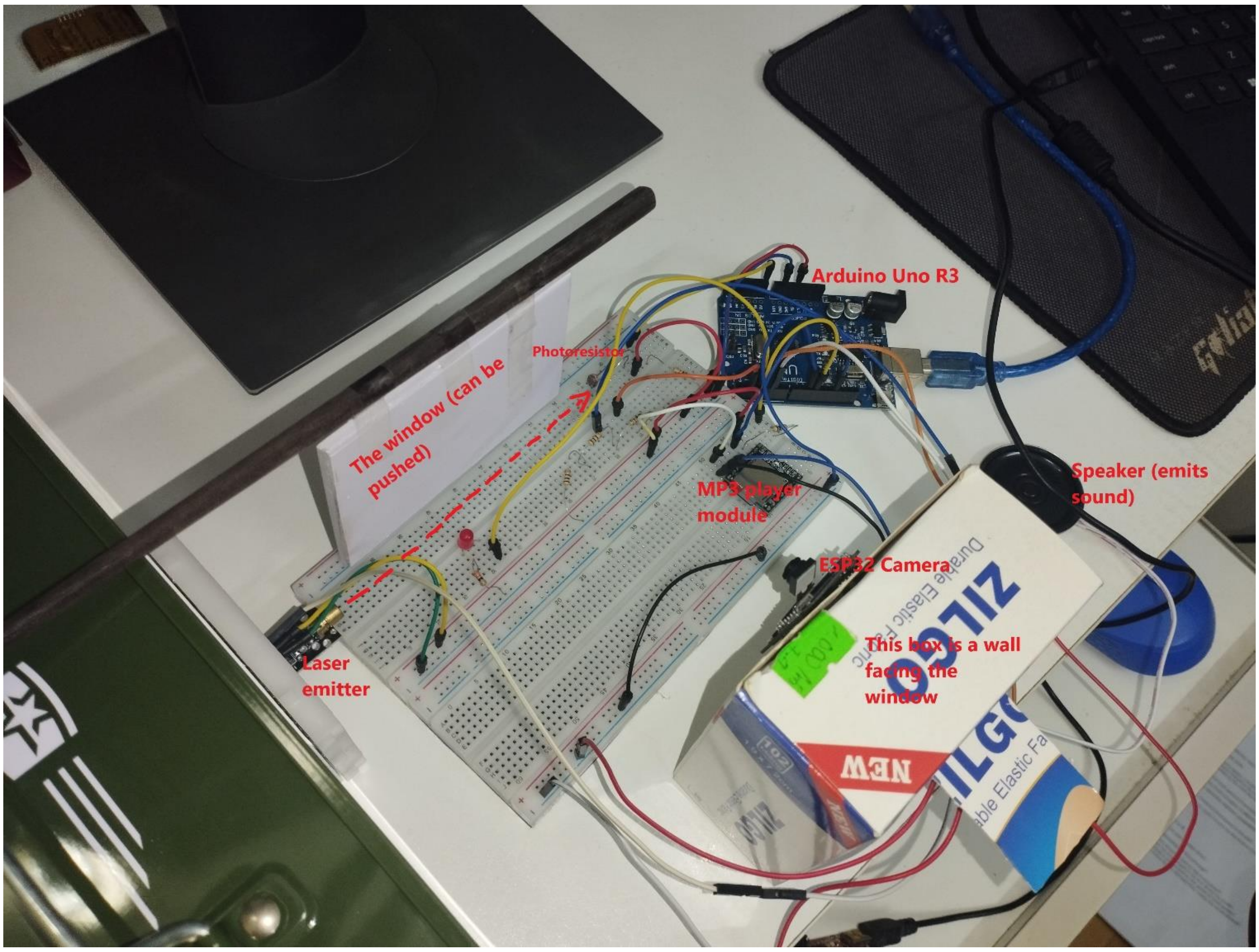
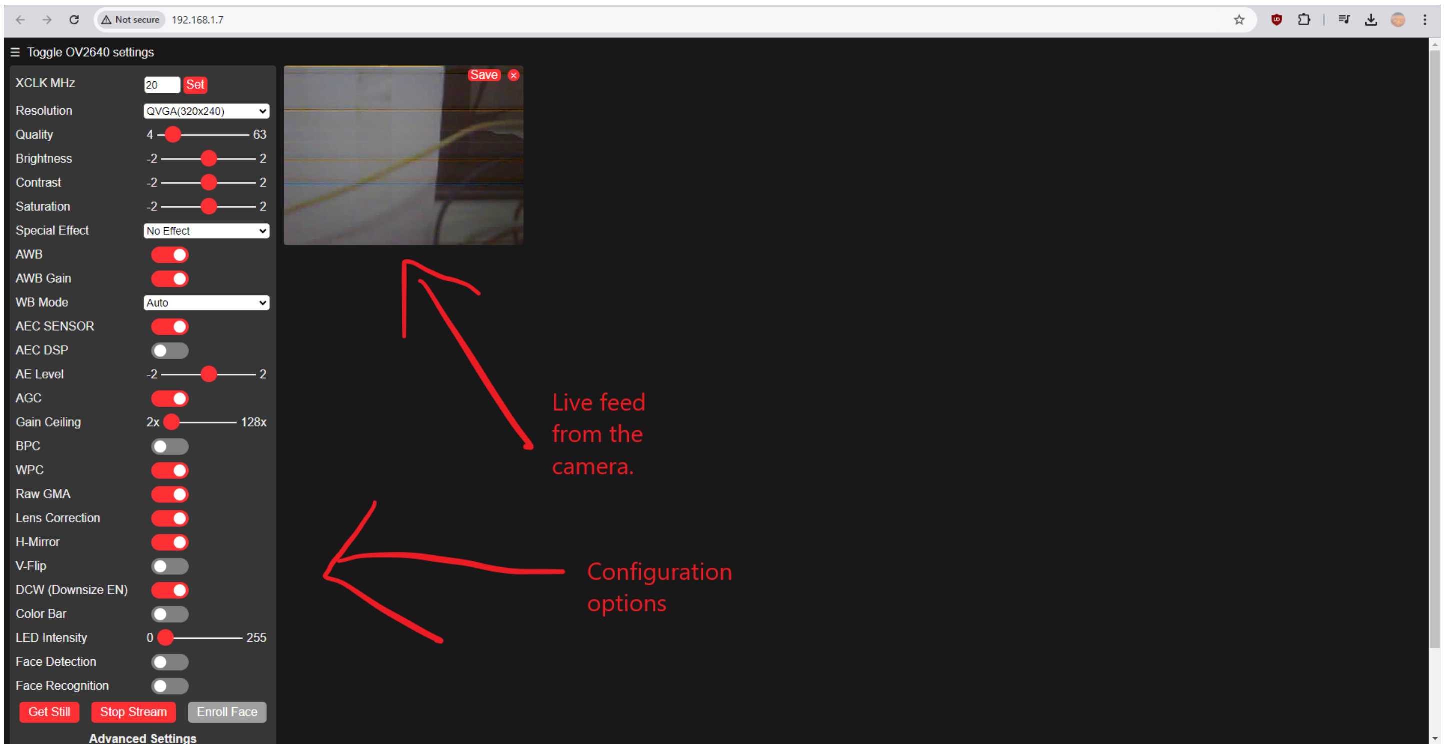


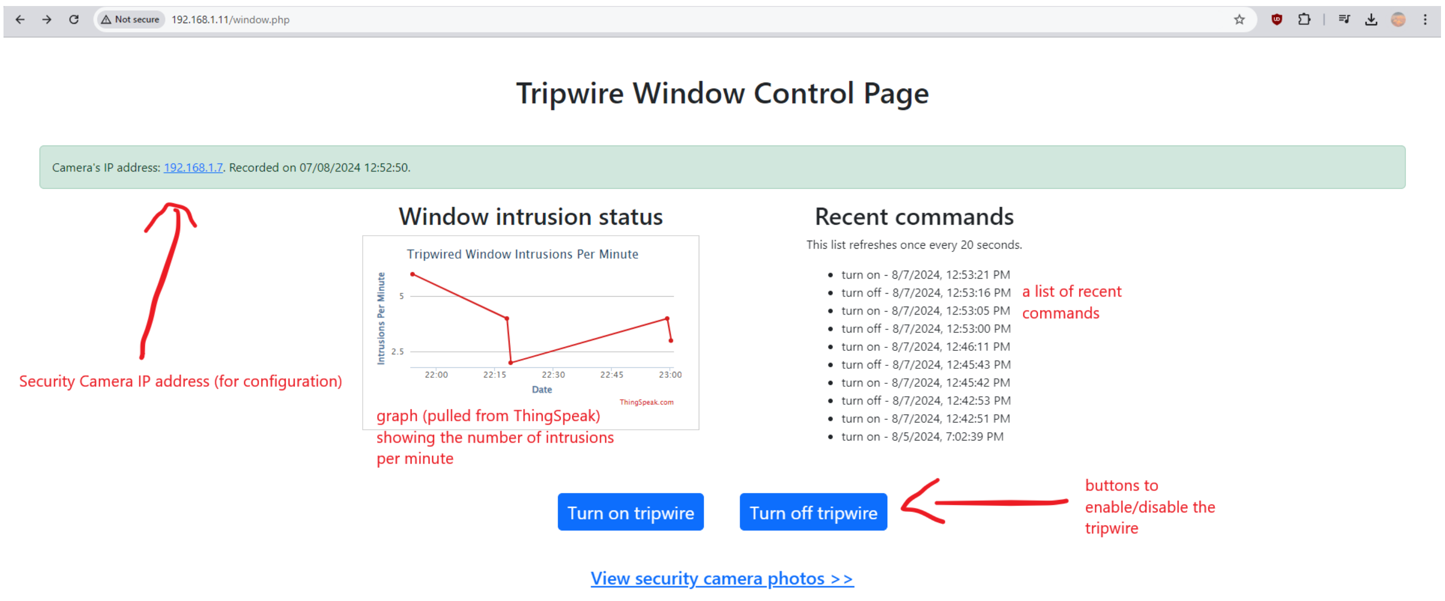
Fig. 4. Secure Window Circuit



☰ Toggle OV2640 settings

| | |
|---|---|
| XCLK MHz | 20 Set |
| Resolution | QVGA(320x240) |
| Quality | 4 — 63 |
| Brightness | -2 — 2 |
| Contrast | -2 — 2 |
| Saturation | -2 — 2 |
| Special Effect | No Effect |
| AWB | ✓ |
| AWB Gain | ✓ |
| WB Mode | Auto |
| AEC SENSOR | ✓ |
| AEC DSP | ○ |
| AE Level | -2 — 2 |
| AGC | ✓ |
| Gain Ceiling | 2x — 128x |
| BPC | ○ |
| WPC | ✓ |
| Raw GMA | ✓ |
| Lens Correction | ✓ |
| H-Mirror | ✓ |
| V-Flip | ○ |
| DCW (Downsize EN) | ✓ |
| Color Bar | ○ |
| LED Intensity | 0 — 255 |
| Face Detection | ○ |
| Face Recognition | ○ |

Fig. 5. Camera Configuration Page (Hosted on the ESP32 Camera)



Not secure 192.168.1.11/window.php

Tripwire Window Control Page

Camera's IP address: 192.168.1.7. Recorded on 07/08/2024 12:52:50.

Security Camera IP address (for configuration)

Window intrusion status

graph (pulled from ThingSpeak) showing the number of intrusions per minute

Recent commands

This list refreshes once every 20 seconds.

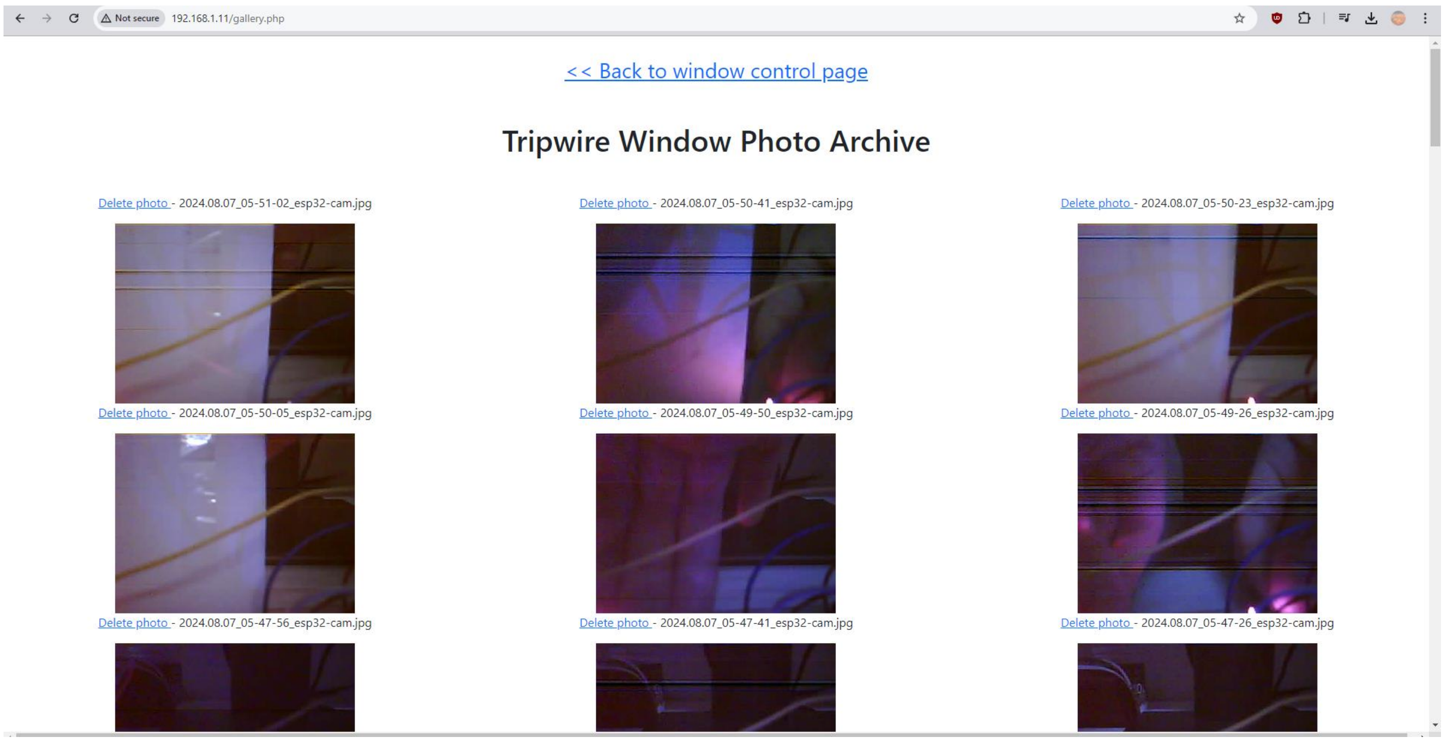
- turn on - 8/7/2024, 12:53:21 PM
- turn off - 8/7/2024, 12:53:16 PM
- turn on - 8/7/2024, 12:53:05 PM
- turn off - 8/7/2024, 12:53:00 PM
- turn on - 8/7/2024, 12:46:11 PM
- turn off - 8/7/2024, 12:45:43 PM
- turn on - 8/7/2024, 12:45:42 PM
- turn off - 8/7/2024, 12:42:53 PM
- turn on - 8/7/2024, 12:42:51 PM
- turn on - 8/5/2024, 7:02:39 PM

a list of recent commands

buttons to enable/disable the tripwire

View security camera photos > >

*Fig. 6. Secure Window Control Page*



Not secure 192.168.1.11/gallery.php

< < Back to window control page

Tripwire Window Photo Archive

Delete photo - 2024.08.07_05-51-02_esp32-cam.jpg
Delete photo - 2024.08.07_05-50-41_esp32-cam.jpg
Delete photo - 2024.08.07_05-50-23_esp32-cam.jpg
Delete photo - 2024.08.07_05-50-05_esp32-cam.jpg
Delete photo - 2024.08.07_05-49-50_esp32-cam.jpg
Delete photo - 2024.08.07_05-49-26_esp32-cam.jpg
Delete photo - 2024.08.07_05-47-56_esp32-cam.jpg
Delete photo - 2024.08.07_05-47-41_esp32-cam.jpg
Delete photo - 2024.08.07_05-47-26_esp32-cam.jpg

*Fig. 7. Webpage showing all photos that the camera has taken*

## Demonstration Video

Our Home Security project’s demo video link: https://youtu.be/82U4zV2EHf8

## Resources

MQTT with HiveMQ: https://www.hivemq.com/blog/how-to-get-started-with-mqtt/

ThingSpeak API Documentation: https://www.mathworks.com/help/thingspeak/

PHP Form Validation: https://www.w3schools.com/php/php_form_validation.asp

ESP32 Dev Kit V1 Pinout Diagram: https://www.circuitstate.com/pinouts/doit-esp32-devkit-v1-wifi-development-board-pinout-diagram-and-reference/

Certification Authorities: https://letsencrypt.org/certificates/

https://community.hivemq.com/t/issues-connecting-esp32-to-mqtt-broker/2822

Servo Motor Control with ESP32: https://esp32io.com/tutorials/esp32-servo-motor

Keypad and LCD interfacing with ESP32: https://esp32io.com/tutorials/esp32-keypad-lcd

HTTP POST Request Method: https://randomnerdtutorials.com/esp32-http-post-ifttt-thingspeak-arduino/

ESP32-CAM Video Streaming and Face Recognition with Arduino IDE: https://randomnerdtutorials.com/esp32-cam-video-streaming-face-recognition-arduino-ide/

ESP32-CAM AI-Thinker Pinout Guide: GPIOs Usage Explained: https://randomnerdtutorials.com/esp32-cam-ai-thinker-pinout/

ESP32-CAM Post Images to Local or Cloud Server using PHP (Photo Manager): https://randomnerdtutorials.com/esp32-cam-post-image-photo-server/

ESP32 MQTT – Publish and Subscribe with Arduino IDE: https://randomnerdtutorials.com/esp32-mqtt-publish-subscribe-arduino-ide/

Introduction to the DFPlayer Mini MP3 Player module: https://wiki.dfrobot.com/DFPlayer_Mini_SKU_DFR0299

ThingSpeak MQTT Basics: https://www.mathworks.com/help/thingspeak/mqtt-basics.html

MQTT.js library: https://github.com/mqttjs/MQTT.js