IoT Home Security System

ECE 442: Internet of Things and Cyber Physical Systems

Part 4: Final Report

Instructor: Prof Jafar Saniie and Prof. Won-Jae Yi

Due Date: 06/23/2022

By: Alan Palayil in collaboration with Nikhil Aditya Chaganti and Hamad Abdelrahim

Table of Contents

Our Team

Alan Palayil

B.S. Computer and Cybersecurity Engineering

A Computer and Cybersecurity Engineering major who is focused on the software integration and design architecture for the Home Security System project. In the past, I have integrated IOT hubs and built a couple of smart home proposals. The smart home included cloud connectivity and security monitoring. I am currently enrolled in the coterminal program for M.S. in Cybersecurity Engineering with the focus in Network and Application security.

Nikhil Chaganti

B.S. Computer and Cybersecurity Engineering

A Computer and Cybersecurity Engineering major who is focused on facial recognition using various computer vision software for this project like OpenCV and Deepstack.

I am a 4th-year in B.Sc. in Computer and Cybersecurity Engineering. I have experience in recommending and implementing security standards and best practices in various projects.

Hamad Abdelrahim

B.S. Computer Engineering

A Computer Engineering major who is focused on the integration of a network based and GPS based presence system and assisted with the automation of the integrated components.

I am a 4th-year in B.Sc. in Computer Engineering. I have worked on facilitating first person view on drones for a project and worked on coding projects using experience gained from courses.

Abstract

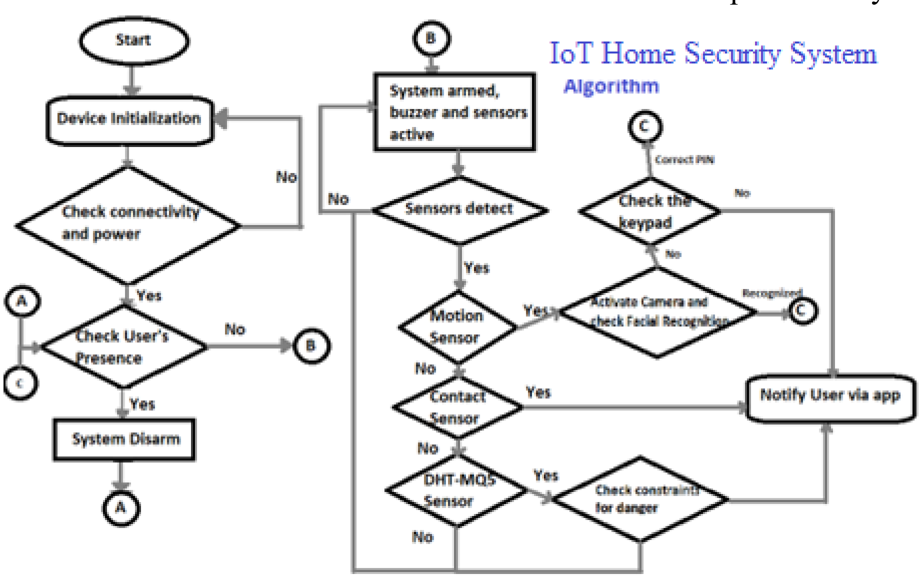
Home Alarm Systems are a key element for monitoring home security and is a developing field in the IoT. The current systems are often limited to the coverage of the system, false alarms, and power usage along with few automations. The IoT Smart Security System is not only an alarm system that will be used on a daily basis for arming the user's home security system. Rather than manually arming with limited functionality, our system offers a smart approach to IoT security systems. The system comprises various nodes and sensors to improve the functionality of monitoring and receiving real-time information using Home Assistant platform. The system communicates using WiFi through ESP protocols between the nodes and microcontroller hub. The sensor detects motion using PIR sensor, temperature and gas conditions using DHT and MQ-5 sensors, and door/window contacts using Reed switch to detect entry. They are connected via jumper cables to Serial WiFi modules or directly to the microcontroller. There is a physical keypad to enter a pin in case the facial recognition/ presence detection doesn't follow through. As for the user interface, the UI is based on the Home Assistant platform which provides a webpage and mobile application which can be customized depending on various users with a possible integration with smart IoT home automations.

Introduction

The goal of our project is to develop a smart versatile alarm system. It has the ability to disarm using facial recognition and presence detection along with a keypad. The project is also to develop a smart security system which can arm and disarm without a lot of user input. The system is divided into 2 groups: Pi group and NodeMCU group. The NodeMCU group consists of the sensors which are connected to the main microcontroller using ESP Serial WiFi modules to increase the range of the system. The Pi group consists of the single-board computer, keypad, and camera which are components for user interaction. The system can automatically be armed when the user is away from home using GPS, keypad, or the mobile application. The system should arm itself by detecting if the user is away and send notifications depending on the detections and presence in the home. When the system is armed, if the sensors detect any exterior motion, it will notify the user depending on the threat level. If there is a forced entry or gas leak, the user is notified with quick links to the respected authorities and any motion is detected is captured and listed in the Home Assistant. The camera is triggered using motion detection in order to decrease the power consumption. The components will be working in tandem to provide home safety. With fire and gas leak monitoring as an addition with the hopes to create an open system which can be used for home automation and monitoring with the help of the webpage and mobile application.

Project Design

The whole system is designed to be in sleep mode with wake intervals or when sensors are activated. The system architecture includes contact sensors for doors/windows, motion sensors which are used to activate camera and be placed anywhere, and DHT-MQ5 sensor to monitor gas leak and fire. The system can monitor and through automation learn user habits. To disarm the system, the individual can come up to the module for the facial recognition, use custom geotag/smartphone for presence detection, or use the keypad to input the PIN. The system will use presence detection to notify users of alerts to the Home Assistant platform. The system can be accessed using the Home Assistant to check the activity report, the images/videos captured, and monitor the user's routines which can later be used to improve the system's automation.



Flowchart for the program architecture for IoT Home Security System

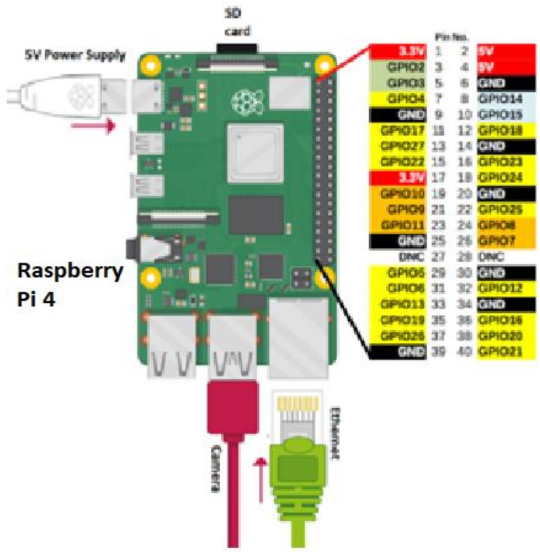
Hardware

The following are the list of hardware components used for this project. The schematics of the component in the system is given below.

Raspberry Pi

A Raspberry Pi 4 4GB was selected single-board computer for this project and it will act as the local server. The Pi was booted with the Home Assistant OS which is an open-source platform to develop various IoT home related projects. The local server in our project collects and stores the data from the microcontroller in the SD card, it later processes the data and uploads it on the Home Assistant Platform. The Pi is connected to the internet via the ethernet cable and is coded to act as a private hotspot to host the necessary network for the system to communicate.

Schematic



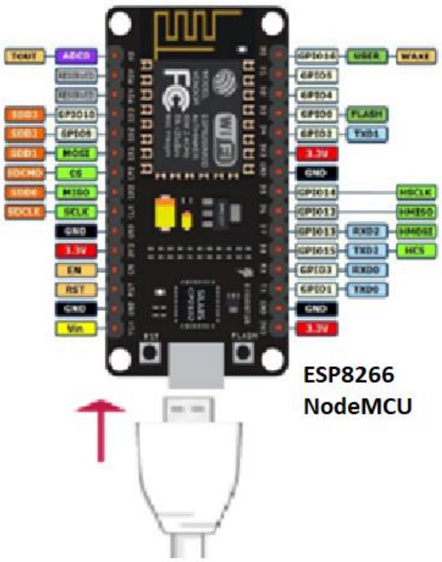
Datasheet

https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf

ESP8266 NodeMCU

An ESP8266 NodeMCU is the microcontroller which is being used for this project and acts as a hub to collect and send data to the sensors and nodes. It is a development board based on Lua firmware. It was selected for its low power consumption and built-in WiFi SoC chip. The NodeMCU is connected to the sensor and nodes either via ESP8266 Serial modules or jumper cables.

Schematic



Datasheet
https://components101.com/sites/default/files/component_datasheet/ESP8266-NodeMCU-Datasheet.pdf

Battery Module

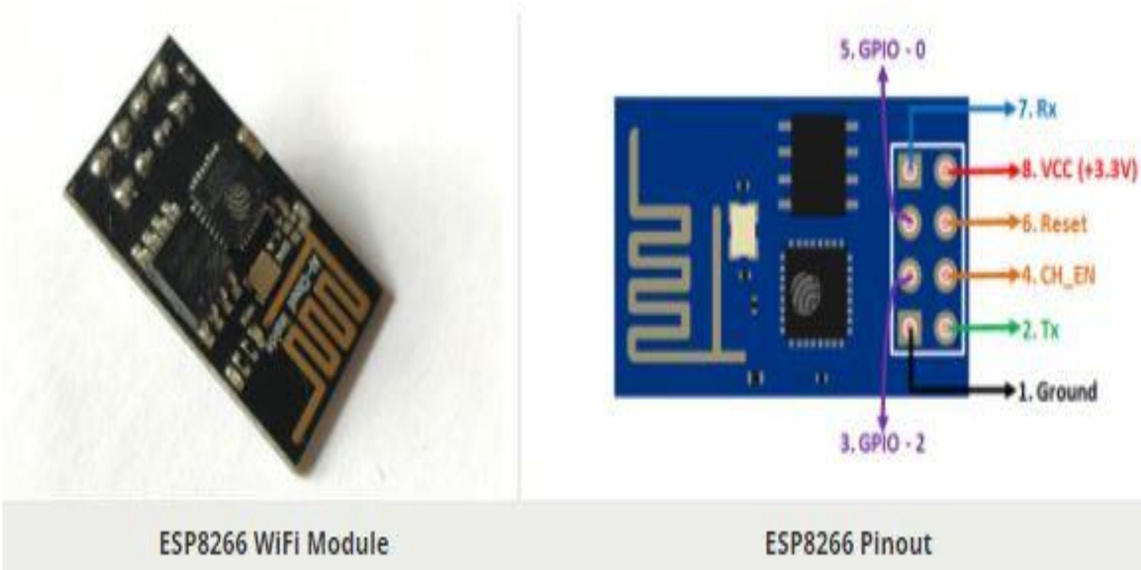
The battery module is used for the system to continue working if there is a power shortage. The module can recharge the 2 Li-ion cells and relay power to the connected devices. The module is connected to the Raspberry Pi, NodeMCU, ESP8266 Serial modules to provide back-up power for our project.

Figure

ESP8266 Serial Wi-Fi Wireless Module

The ESP-01 Serial WiFi Transceiver is a SoC with integrated IP protocol that gives sensors and microcontrollers access to WiFi. The module can host an application or offloading all WiFi networking functions from other processors like NodeMCU. The module is used to connect the sensor and nodes to the NodeMCU with the ability transmit data in an increased range.

Schematic



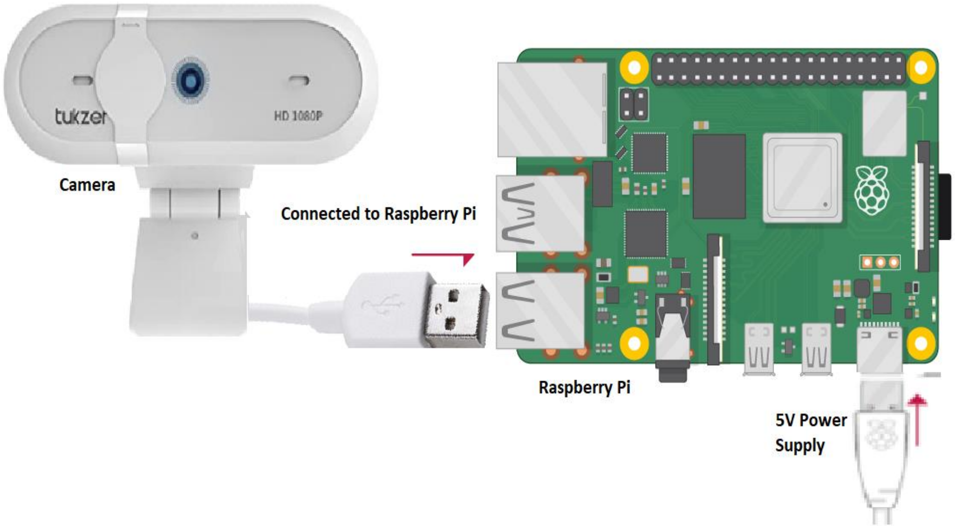
Datasheet

https://components101.com/sites/default/files/component_datasheet/ESP8266%20Datasheet.pdf

1080p USB Camera

A 1080p USB camera is connected to the Raspberry Pi 4 and runs OpenCV software. The camera is placed at the entrance of the building. The camera will be used to run facial recognition to disarm the system and can be used to monitor the entrance using motion sensor.

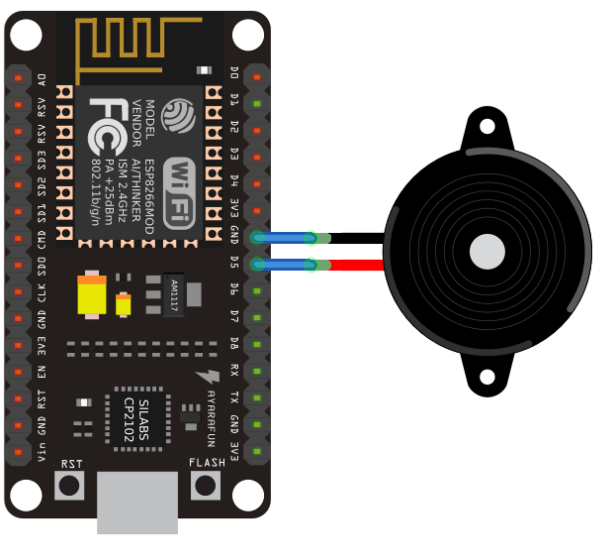
Figure



Passive Buzzer Alarm

The buzzer is used to sound an alarm which is connected to the NodeMCU to hear sounds if any of the conditions are met like forced entry and notifications.

Schematic



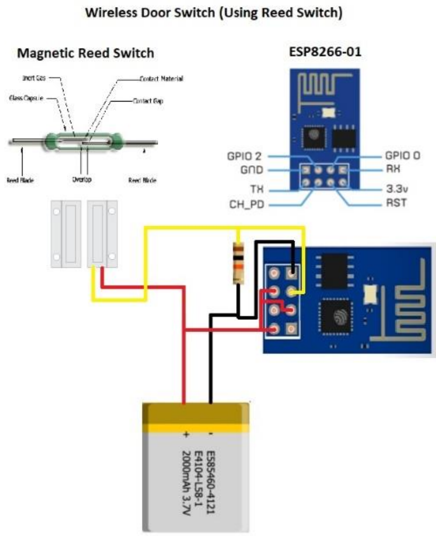
Datasheet

https://components101.com/sites/default/files/component_datasheet/Buzzer%20Datasheet.pdf

Magnetic Recessed Reed Switch

The MC-38 Wired Door and Window sensor can be used to add security to the entrance points. It produces the signal when the magnets are moved away from each other that is transmitted to the NodeMCU using the Wireless Serial module. When the magnet is closed by, the circuit is closed or open if the magnet is far from the sensor. The trigger shows the status of ON/OFF which can be viewed in the Home Assistant platform.

Schematics



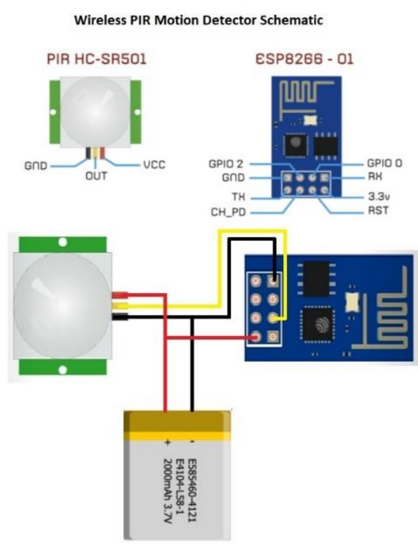
Datasheet

https://components101.com/sites/default/files/component_datasheet/MC38-Datasheet.pdf

PIR Sensor

The PIR sensor is used to sense motion and can almost always detect whether a human has moved in or out of the sensors range. The sensor detects the motion of a human body using Passive Infrared sensor. It is used in this project to activate the camera on motion and capture images/videos.

Schematics



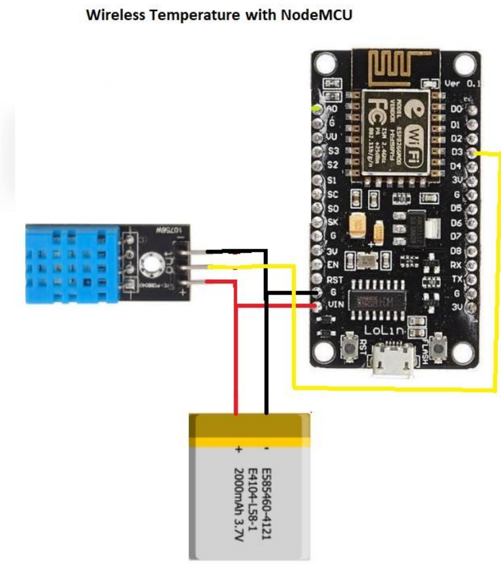
Datasheet

https://components101.com/sites/default/files/component_datasheet/HC%20SR501%20PIR%20Sensor%20Datasheet.pdf

DHT11 Sensor

The DHT11 sensor is a pre-calibrated resistive sensor that is coupled with NTC thermistor to precisely read the relative humidity and surrounding temperature. The sensor sends a digital data and is a low-cost and power module.

Schematics



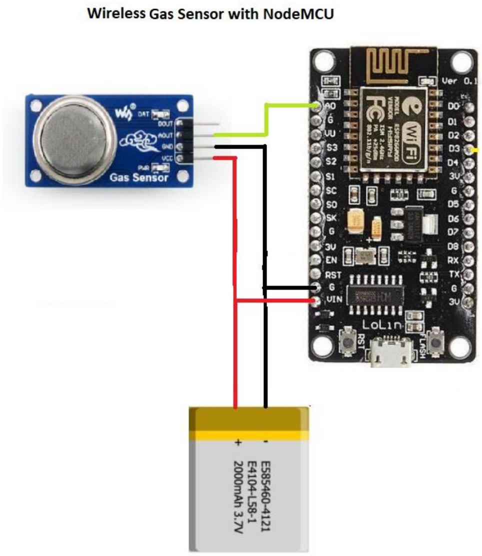
Datasheet

https://components101.com/sites/default/files/component_datasheet/DHT11-Temperature-Sensor.pdf

MQ-5 Gas Sensor

The MQ-5 Gas Sensor is used for Methane LPG module which senses particulate like LPG, natural gas, town gas and avoids alcohol and cooking fumes and cigarette smoke. The module targets combustible gas and measures its concentration. This module is added to improve the safety of the security system to detect gas leak.

Schematics



Datasheet

https://datasheetspdf.com/pdf-file/904638/HANWEIELETRONICS/MQ-5/1

Communication

The system is divided into 2 groups. The Pi group comprises of the Raspberry Pi, camera, and keypad. The keypad is connected to the Pi via the GPIO pins, and the camera is connected to the USB 3.0 port of the Raspberry Pi. The Pi itself is connected to the internet via the ethernet which provides a physical internet connection. The Pi acts as a local server and connects to the Home Assistant platform via the internet. The Pi collects the data from the nodes, it stores it in the SD card, processes the data and uploads it to the cloud. The data can be accessed either by through the Pi or the Home Assistant platform remotely via the internet for greater accessibility.

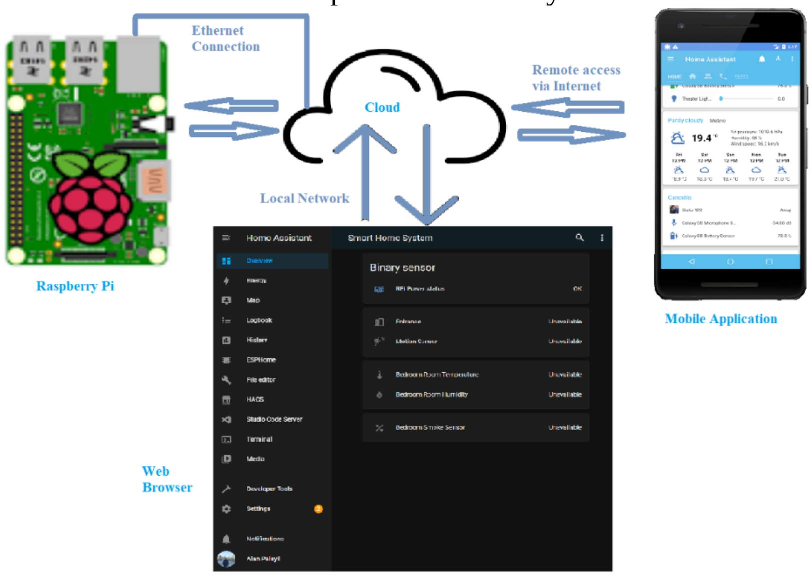


Figure to show the communication between the Pi and Home Assistant

The NodeMCU group consists of the ESP8266 NodeMCU along with various sensors. The NodeMCU acts as the microcontroller hub through which the data from the sensors is transmitted to the local server (Raspberry Pi). The sensors are connected with back-up battery module and ESP-01 Serial WiFi module to transmit the data to the NodeMCU. The WiFi modules communicate using the ESP protocol and transmits the data to the local server through the MQTT protocol via local network.

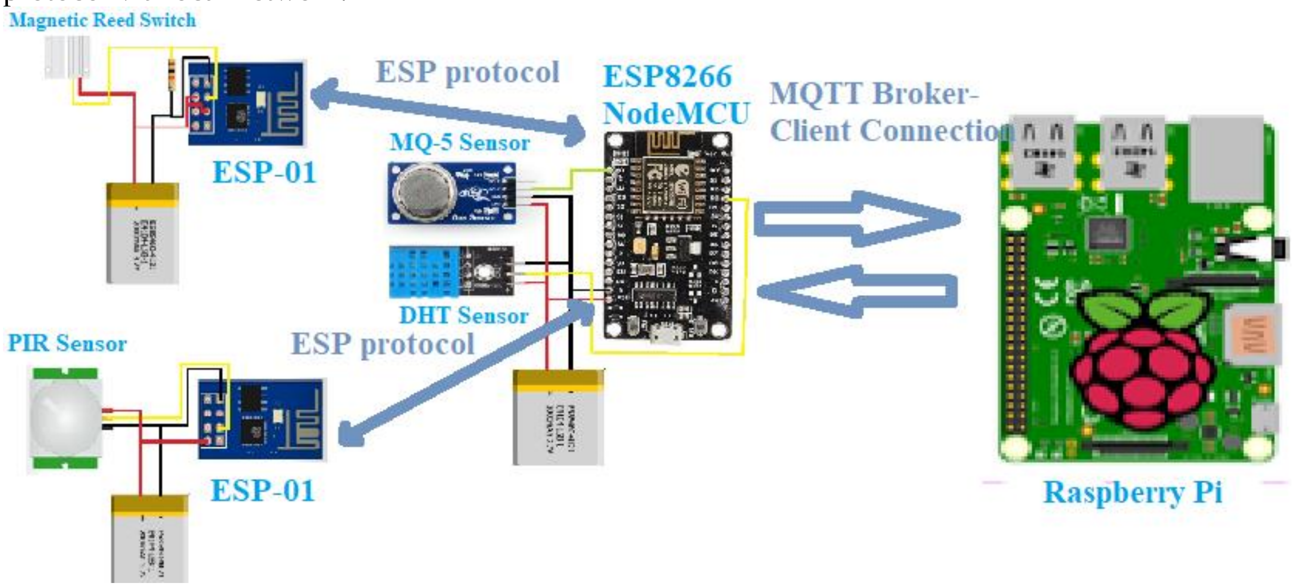
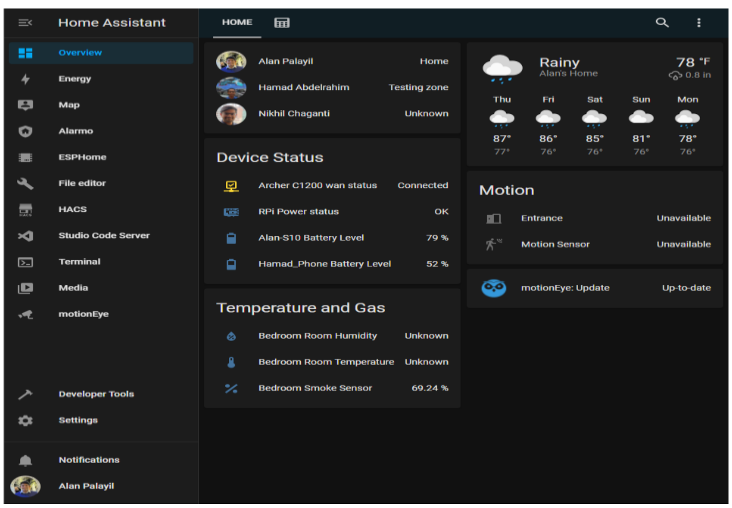


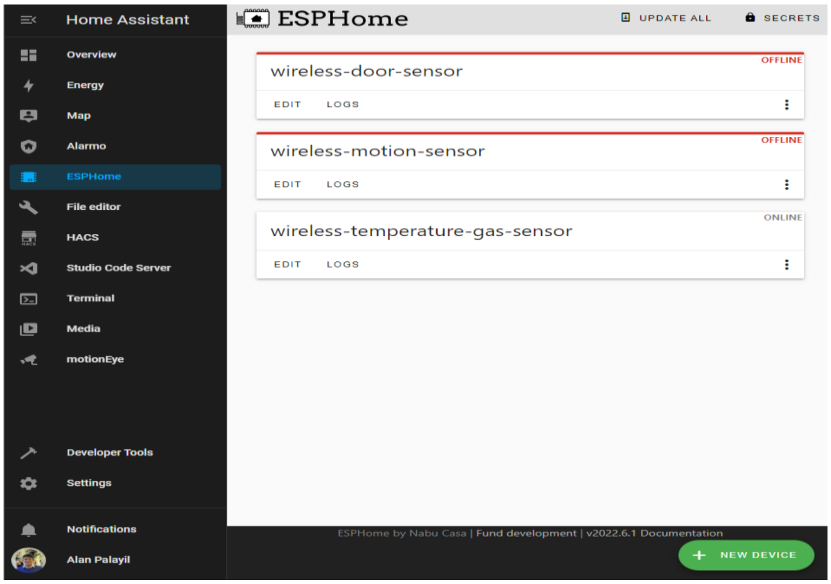
Figure to show how the sensors communicate with the NodeMCU and Local Server

Software

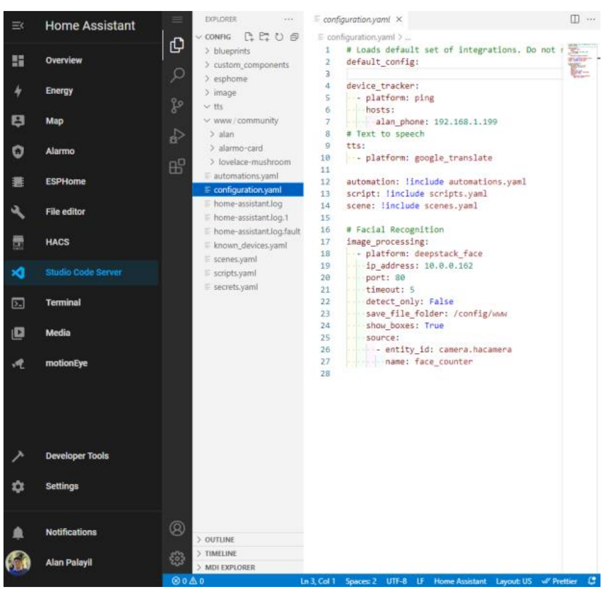
Home Assistant: This is the Operating System of our project and the platform onto which our project is built on. The Home Assistant is an open-sourced home automation that puts local control and privacy first. It is used as the main platform to connect the sensors and actuators to the cloud. Using in-platform services like ESPHome and MQTT broker the nodes are defined and installed to the system. Home Assistant keeps your data local using SD card and periodically updates the cloud with the data logs.



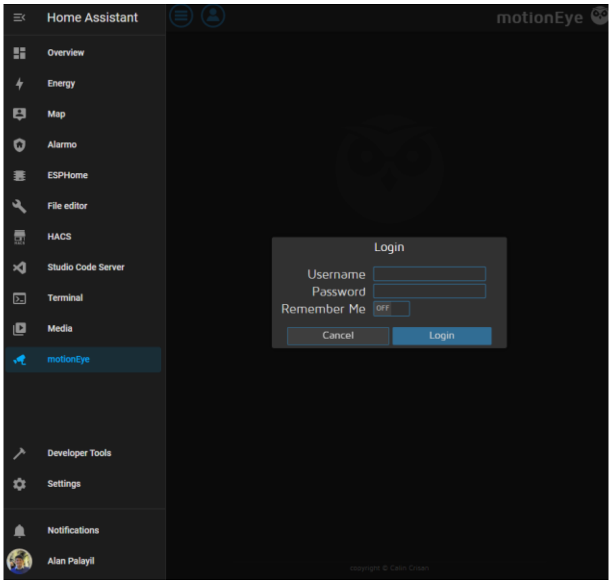
ESPHome: This is the integration platform to the ESP modules and is used to set up scripts for the sensors and actuators connected to the ESP module. It helps in creating the configuration files with the Home Assistant and controls them remotely through the OTA support. The integration includes various ESP modules along with different sensor and actuator components.



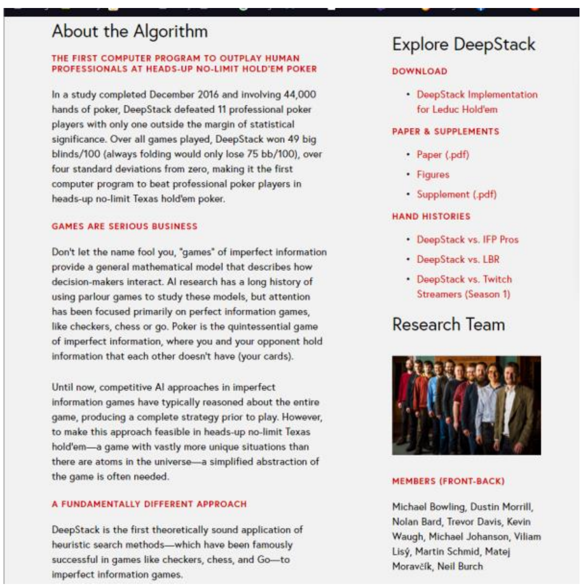
Studio Code Server: This is the integration used to edit and program configuration codes. It allows to open any folder in Home Assistant with the ability to develop, edit, build, or debug on target deployment.



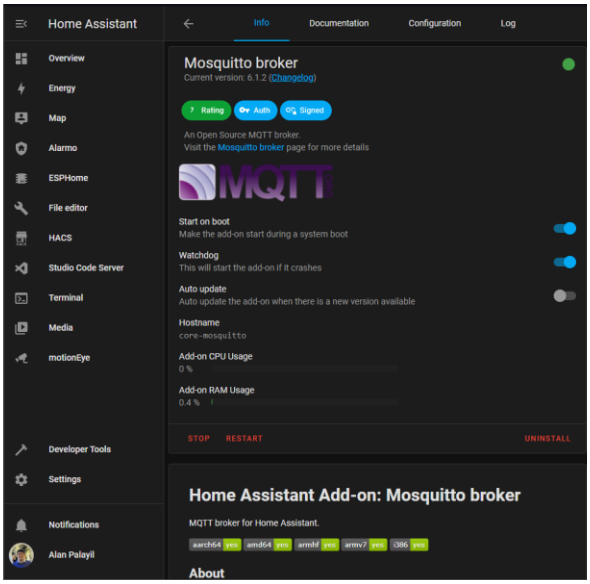
MotionEye: This is the visual aspect of our camera. It is used to setup a motion-triggered security camera which captures pictures or videos and displays them on the Home Assistant dashboard.



DeepStack and OpenCV: The software to run facial recognition. Using Deepstack integration on Home Assistant, a camera can detect motion and run facial recognition scans. Deepstack is manually taught new faces to improve recognition accuracy.



Mosquitto Broker: The MQTT broker to establish communication between nodes and the Raspberry Pi which implements the MQTT protocol version. This makes it suitable method to carry out messaging using publish/subscribe model. The Mosquitto project also provides a C library for implementing MQTT clients, and the very popular mosquitto_pub and mosquitto_sub command line MQTT clients.



Programming Language

For the programming languages are used for our Home Security System:

Raspberry Pi: The Pi is initially set up using Linux and later when installing the Home Assistant operating system, we used Python codes and scripts to run automations and events on the Home Assistant platform. The ESPHome integration on Home Assistant uses C to communicate with the microcontroller and sensors.

Microcontroller: The ESP8266 NodeMCU and ESP-01 WiFi module both are coded using C and to upload the scripts we use the ESPHome to upload the scripts over the air (OTA).

Sensor: The sensors are connected to the microcontroller or ESP-01 module and are initialized using C in ESPHome.

Facial Recognition: We are using two software for running facial recognition on our software. The OpenCV platform was used to understand the programming functions of real-time computer vision libraries and the system is set up and programmed using C++ and C. The Deepstack software also uses C and C++ to build custom models to improve facial recognition software.

Constraints

Power: The ESP SoC chip and the sensors requires a constant 3.3V DC input. We have implemented a battery module which connects a battery back to the nodes.

Price: The system needs to be in the similar price range as that of the systems available in the market today.

Range: Since the system communicates wirelessly, the range of the system depends on WiFi range of the ESP SoC chips and the home network.

Speed: Since the system works through WiFi, the communication between the sensor nodes is fast as it takes place using the ESP protocol.

Storage: The local server's storage depends on the size of the SD card and to overcome this the data logs are updated periodically to the cloud server.

Security

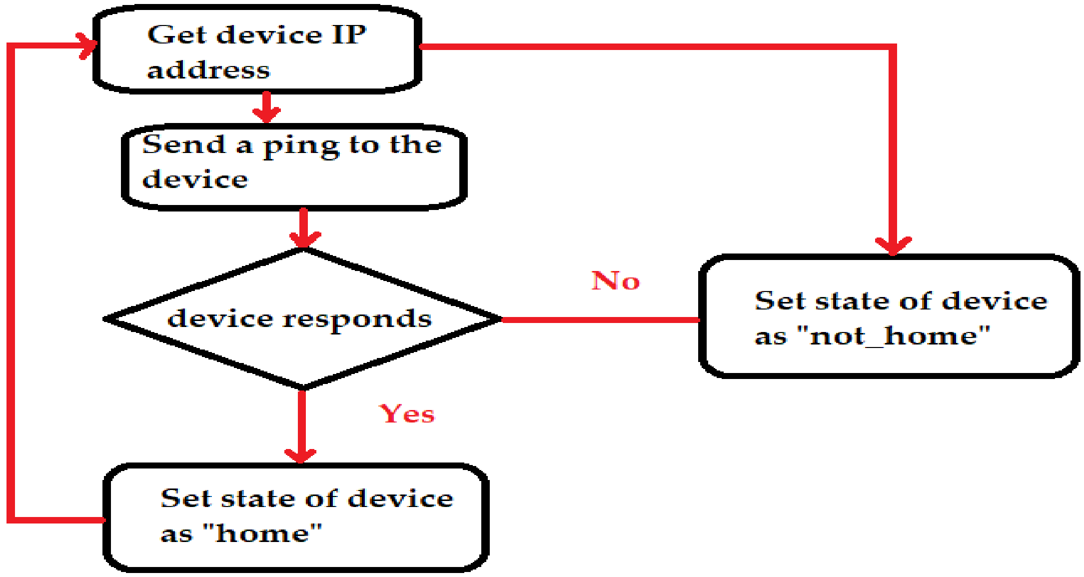
Network Security: Since our system is heavily designed over network communication, we need to have an encryption protocol. Using ESP for the sensors to communicate with the microcontroller and the microcontroller communicating to the local server using the MQTT protocol decreases the change of intruders or hackers to gain access to the sensor nodes.

Sensor Security: Our project needs to have fault tolerances to ensure the integrity of the system. This includes the power loss and tampering of the system nodes. The system needs to be notified if any of its node has been tampered with. This includes moving of the nodes or cutting of the power supply.

Originality

The system is built over existing software platform of Home Assistant with additions of personal scripts to run the system according to our goals.

Presence detection system: It uses two-factor authentication with Network based detection and GPS based detection. For the Network detection, the user provides their phones or device's IP address to the Home Assistant. The Home Assistant periodically sends a single data packet (ping) through the home network to the device and if the device responds the status of the device is set to "home" else its set to "not_home". The GPS detection uses the mobile application to track the user's location and sends it to Home Assistant. The location is processed into a map and can be accessed in the sidebar of the platform. Using the GPS based detection, we are able to create zones which can be used as a trigger for events and automations. Using both of the detection system helps us create an accurate presence system.



Flowchart for Basic Presence Detection System

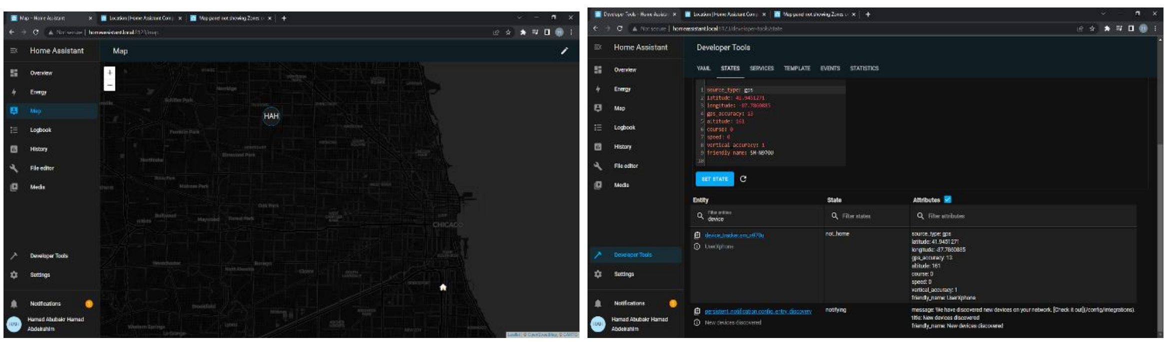


Figure to show Network based presence detection

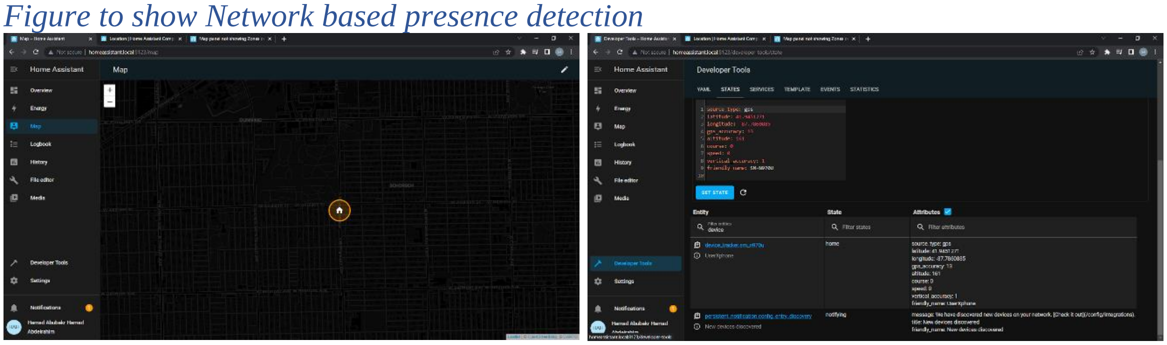
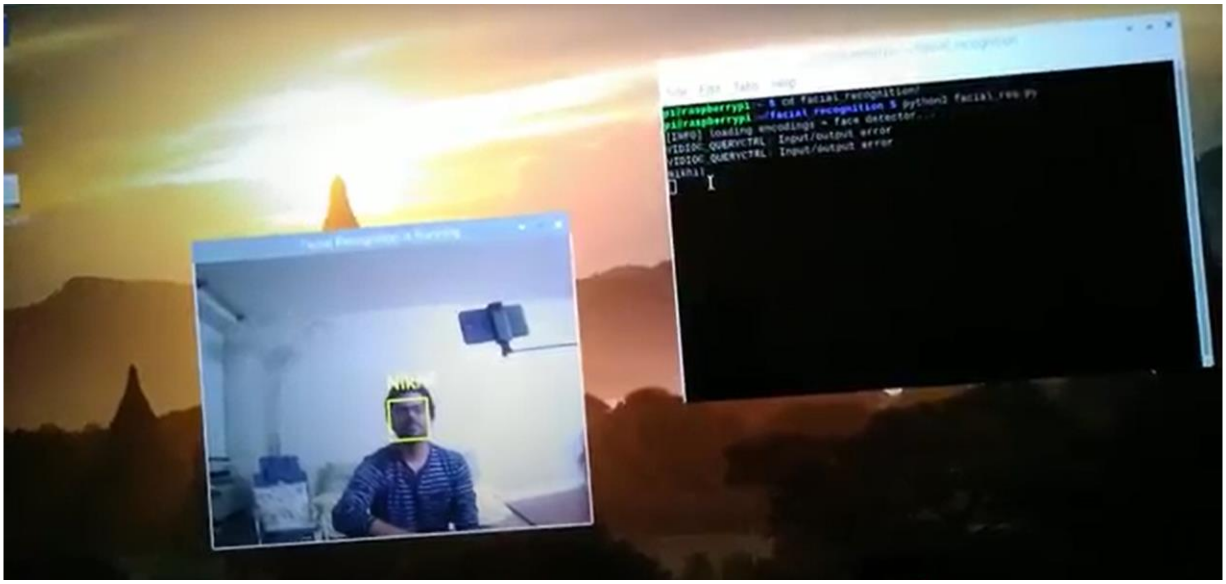


Figure to show GPS based presence detection

Facial Recognition: The system comprises the facial recognition at the entrance of the building which gets activated during motion detection. The software used for this feature is Deepstack which runs in the Home Assistant as an integration and is similar to OpenCV. The Deepstack is an AI server for computer vision applications like object and face detection and recognition. The service can be integrated to Home Assistant and can be taught multiple faces for high-confidence recognition results.

With the inclusion of motion detection, the system has low power consumption to improve the battery life of the sensor nodes. This ensures that the camera is not recording reductant videos to increase memory storage.



Related Work

There are many projects similar to our project. Each project included a part of the system, or the components used on the projects were different. The projects all included sensor nodes and microcontrollers which were connected using wires that limit the range and modularity of the alarm system. With the addition of Zigbee or other transceivers to connect the sensor nodes to the local server.

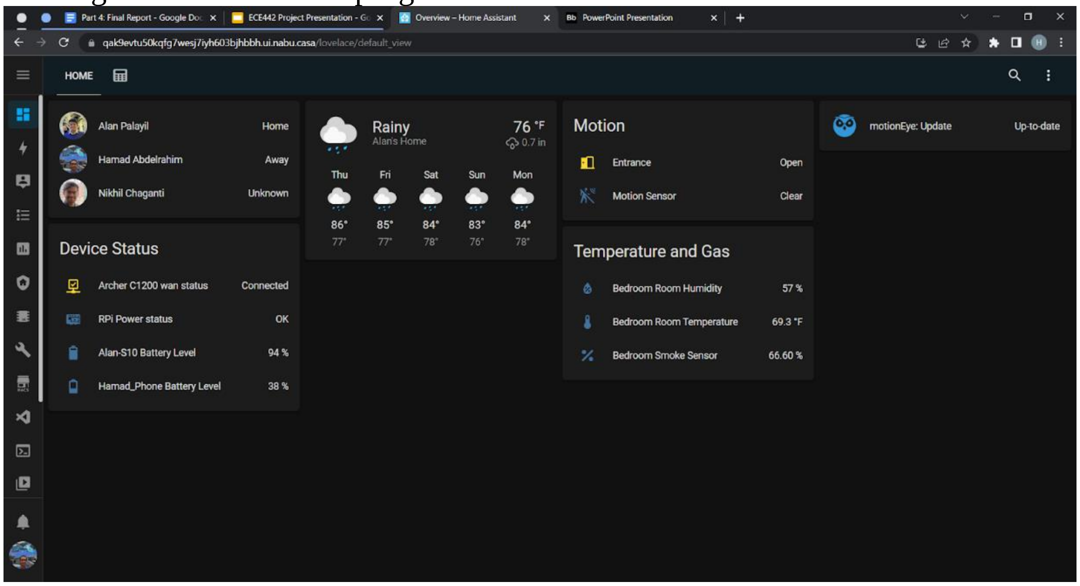
Our project is divided into two groups to ease the design of the system. With the microcontroller acting as a hub and communicating with the sensor using the ESP modules which establishes a wireless communication protocol between the nodes which can increase the range of the system and by using a separate communication between the local server and hub, it does not increase the traffic between the nodes and server. Our project also looks over the various faults like network and power loss.

User Manual

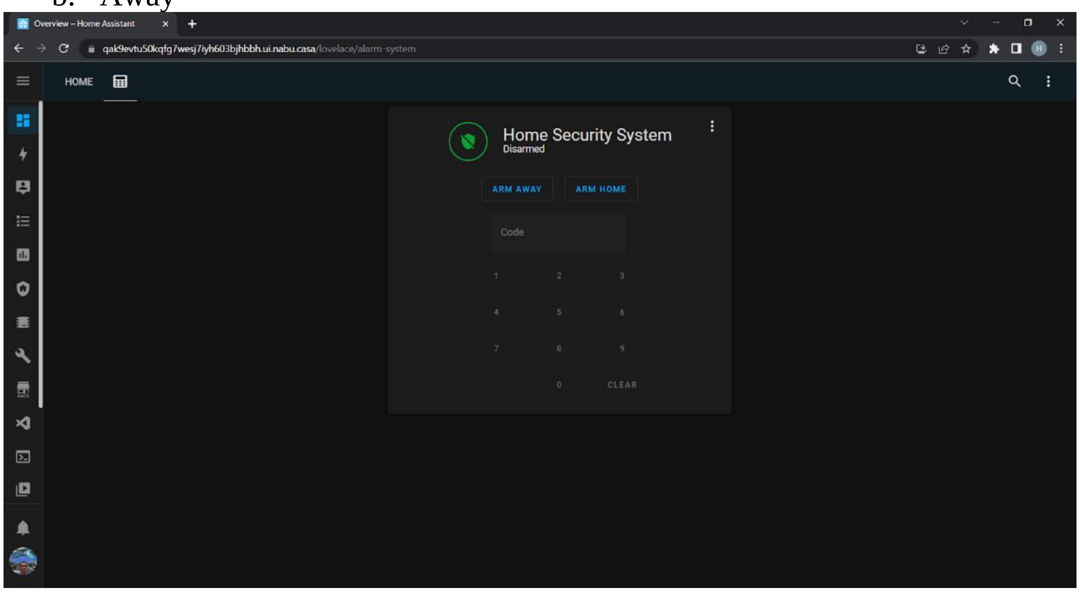
Navigation through Home Assistant:

1. The Dashboard is divided into two sections:
   a. Home
   b. Alarm System
2. Under the Home, the user can view and manage the following:
   a. Other users' presence to the home.
   b. Weather Information
   c. Sensor Status
   d. Temperature and Humidity
   e. Gas Concentration
   f. Device Status

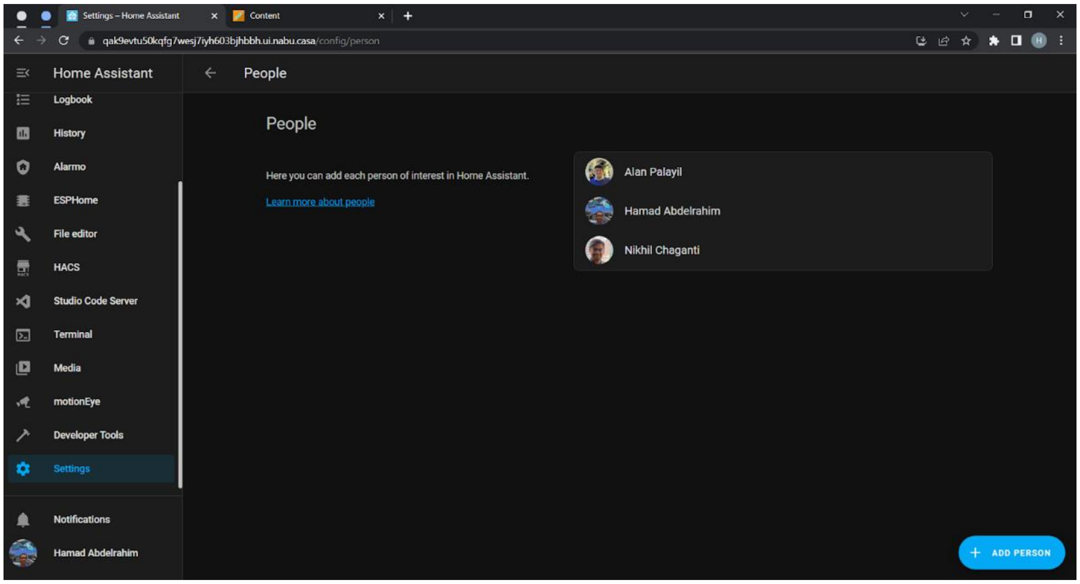
3. If the user wishes to add more information or devices, they can modify the dashboard by clicking the 3-dots in the top right corner.



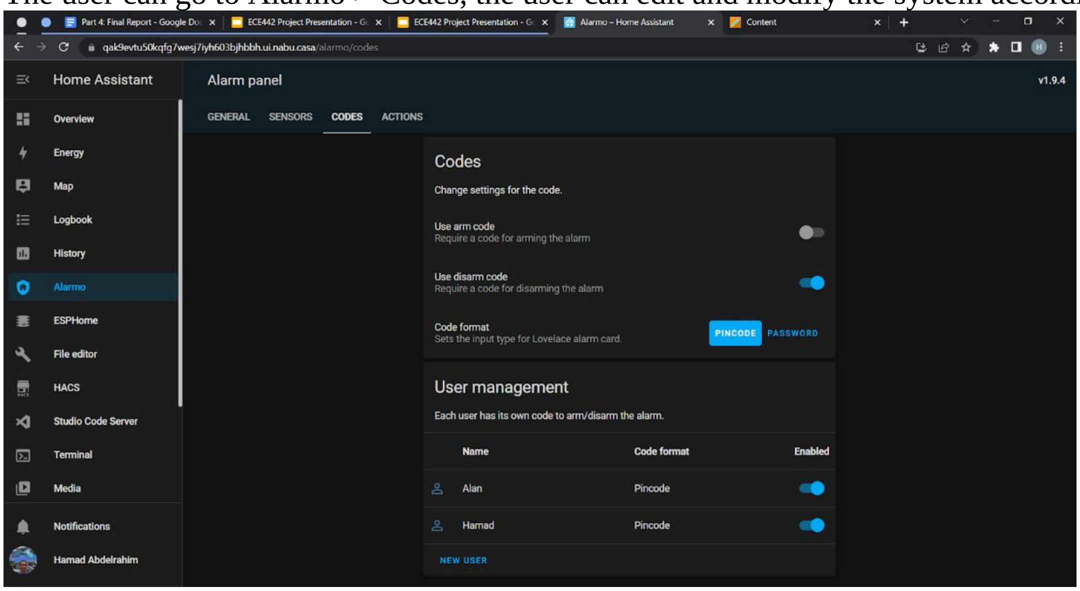
4. Under the Alarm System, the user has access to a keypad to arm/disarm system.
5. The system includes two modes:
    a. Home
    b. Away



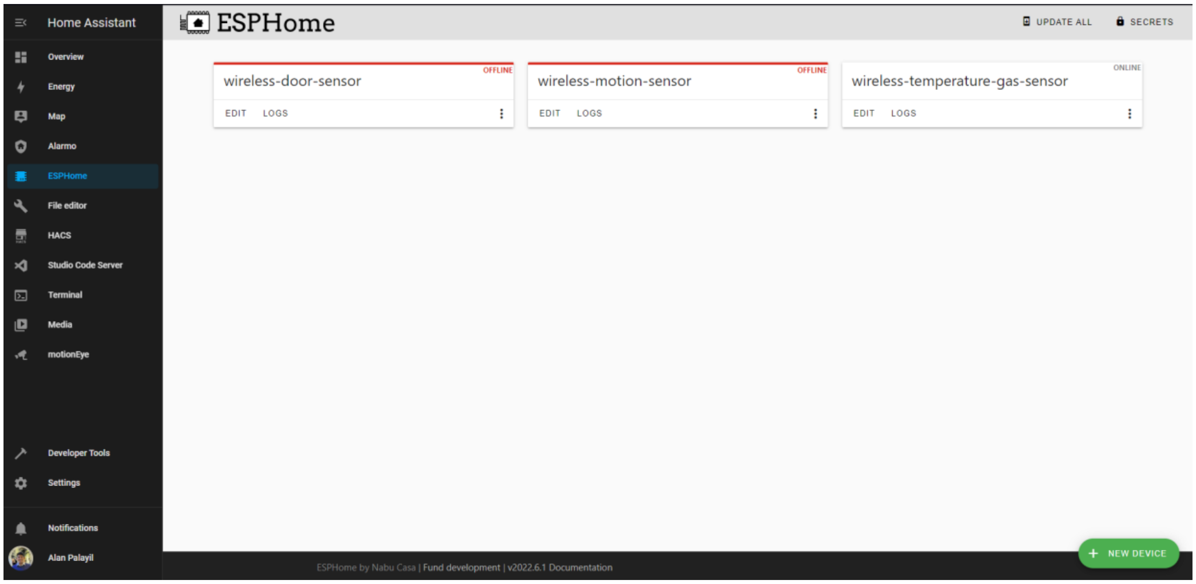
6. The system is open to the addition of more users which can be navigated by going to Settings > People.



7. The user can go to Alarmo > Codes, the user can edit and modify the system accordingly.



8. The users can add sensors or update sensor scripts using ESPHome integration via the sidebar.



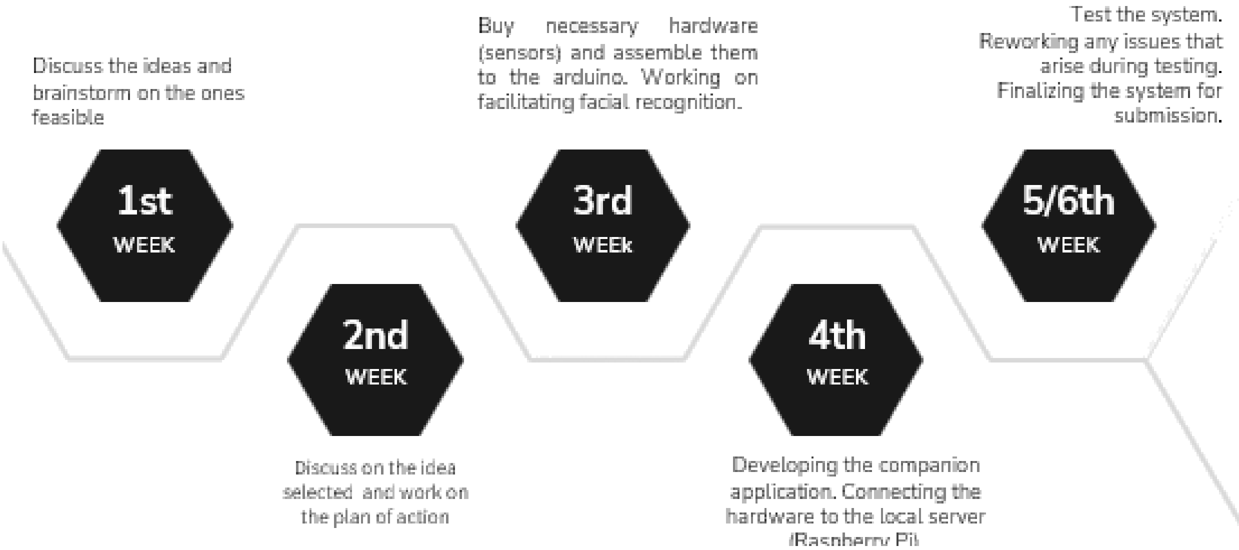
9. The system can be updated by going to the Settings where the update will be displayed.

Milestones

The system is currently implementing the facial recognition separately along with the base system to implement presence detection.

1. We have installed the Home Assistant operating system on Raspberry Pi with the initial system setup to build the system on.
2. Research over the various scripts and libraries that can be used in Home Assistant to set up NodeMCU modules with WiFi using the ESP protocols.
3. Researched on the possible features that can be implemented on the home alarm system via Home Assistant to make it smart and user friendly.
4. Researched on various presence detection software and devices to implement automated features using custom automation.
5. Researched into the use of IFTTT to automate features of the system with integration of Google Services and Amazon Alexa services.
6. Implementing facial recognition via OpenCV through Deepstack and Double Take separately and then pairing it with the system using Facebox in Home Assistant.

Timeline



Work Contribution

Member	Contributions
Alan Palayil	1. Design an existing system using ESP as the modes of communication between the sensor nodes and microcontroller ESP8266 NodeMCU. 2. Research on integration of the system and design for multi-node set-up in Home Assistant platform.
Nikhil Chaganti	1. Research and look over the implementation of facial recognition software on Raspberry Pi. 2. Work on the integration of software within OpenCV like Deepstack and Double Take with the Home Assistant platform.
Hamad Abdelrahim	1. Research on the working and implementation of various sensors and microcontrollers. 2. Look over different automations and presence detection protocols to implement in our system.

Future Work

Although our current systems seem like other home security systems that can be purchased online, there are always improvements that can be done with both the assembly and software of our project. First of all, the assembly of the system can be housed in various sealed plastic covers to ensure the system is durable and the internal components cannot sustain any damage. The sensor nodes can be housed in small forms with the battery module to ensure versatility and stay on par with the competition.

With future development the whole system can be built from scratch which can be programmed to have easy integration with other software and services. The system has been developed with the thought to expand the system by connecting it to the Smart Home to complete the smart environment. This complete environment can include the integration of Google Home and Amazon Alexa to increase automations and

like the inclusion of home automation with light control, network monitoring, and power consumption. A key feature we want to add is a buzzer system that notifies the user with a quick link to inform the officials if the system is set to arm and the sensors detect something.
Mass-producing the system may seem like a challenge and expensive to do, however, we believe the device is cheaper than those on the market. With bulk purchase the price of the system can be reduced further.

Technical Challenges

Power-Loss: We considered the scenarios where the system doesn't receive power which can ultimately decrease the security aspect. To overcome this, we have attached rechargeable battery modules and programmed the hardware to either be on low-power mode or sleep-mode which activates the device only if the sensor detects anything.

Facial Recognition: With the core mechanism of this system being facial recognition, accuracy is vital and depends on the image resolution of the camera. The camera set-up took the majority of time due to which the programming of the facial recognition software wasn't completed to its fullest.

Fault Tolerance: We considered scenarios in which parts of the system might fail or get bypassed by the intruder such as if the intruder manages to force the front door open, a contact sensor that is placed at the door will trigger and cause the system to send a notification to the user if they are home or call the authorities immediately if they are not.

Remote Connection for Presence: While working on improving the presence detection software, there were setbacks on the implementation of the DNS server with the router to accept remote access. We decided to change our approach and look over the existing cloud connection for the demonstration of the project.

Broken Hardware: While connecting the sensor and testing them. The PIR sensor seemed to have a low range of 10cm in comparison to the 1m range. The default parameters of the sensor were changed and had to be manually updated.

Along with this the setting up process took longer than expected and there were few troubleshoots with the component integration and testing.

Conclusion

The project went on at a good pace with all the team members working and researching on their respective tasks. There were slight delays along the way, but the Home security system can be incorporated with other IoT devices and systems to build a smart home ecosystem that learns from the user's daily routine and helps in securing their home. It not only offers valuable real-time information and monitors the user's routine, but it gives the user full control over with smart automation and notifications. It serves as a tool that makes arming your system easier and more efficient with self-arming and disarming protocols by user presence detection, facial recognition, and keypad. The system comprises a couple of fault tolerances which include manual keypad, emergency hotspot, and back-up power. The system communicates completely via WiFi using the ESP protocol between the microcontroller hub (ESP8266 NodeMCU) and local server (Raspberry Pi) and the local server uploads the data to an online cloud (Home Assistant). Although there we troubleshoot while designing and setting up the project, viable solutions were chosen to ensure the system is working as intended.

References

The following are the references used for the research, designing, and set-up of our project:

1. https://components101.com

To get detailed description of components along with their datasheets.

2. https://www.home-assistant.io

It was used to install and refer the various functionalities of the Home Assistant software.

3. https://esphome.io

It was used to add the required libraries of the modules for our project.

4. https://everythingsmarthome.co.uk/face-recognition-just-got-easier-home-assistant-double-take-guide/

A blog on the facial recognition software integration on Home Assistant.

5. https://espresense.com

It was used to understand the various presence detection systems.

6. https://github.com/smarthomejunkie/Home-Assistant-Tutorials

To learn about the various functionalities of Home Assistant automations

7. https://www.room-assistant.io

To add zones and create triggers for presence.

8. https://companion.home-assistant.io/docs/getting_started/

The documentation for mobile application.

9. https://github.com/nielsfaber/alarmo

The alarm system example which we referred.

10. https://github.com/thejeffreystone

To add custom information cards on the Home Assistant platform.

11. https://everythingsmarthome.co.uk/building-a-temperature-sensors-for-home-assistant-wemos-d1-mini-with-ds18b20-build-guide/

A blog to understand the parameters on create automations related to fire and gas.

12. https://github.com/sensorsiot

Repository for sample automations on various sensor modules.

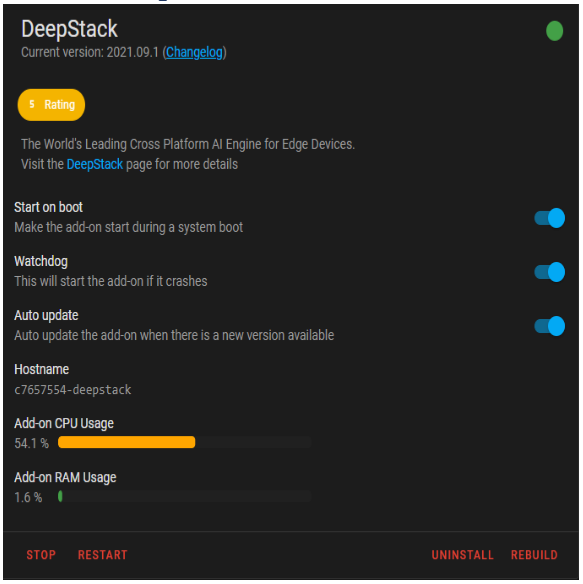
Appendix

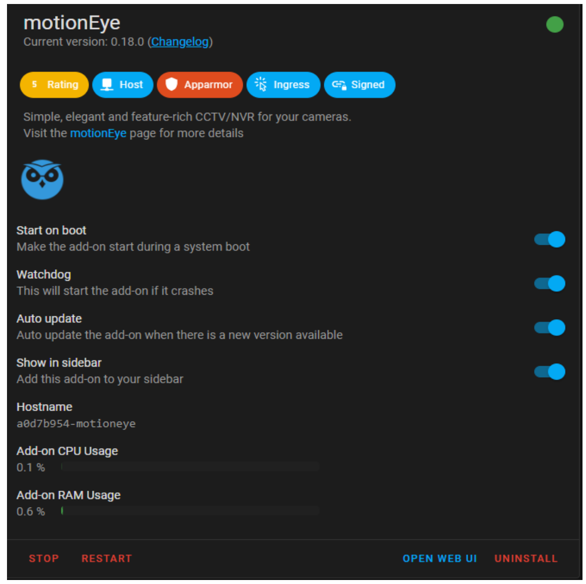
Codes

Home Assistant and Raspberry Pi

https://www.home-assistant.io/installation/raspberrypi

Facial Recognition





NodeMCU and Sensors

Presence Detection system; configuration.yaml file

```
device_tracker:
platform: ping
hosts:
  user_phone: Phone’s IP address

```

Magnetic Door Sensor .yaml file

```
esphome:
  name: wireless-door-sensor

esp8266:
  board: esp01

# Enable logging
logger:

# Enable Home Assistant API
api:
  encryption:
    key: "OQaH43dnB3VC3QT3QWrDS4Z1/RtDgXiAdg33t2zKZt4="

ota:
  password: "b3a6a81f8e01b436ba0952229b958480"

wifi:
  ssid: !secret wifi_ssid
  password: !secret wifi_password

  # Enable fallback hotspot (captive portal) in case wifi connection fails
  ap:
    ssid: "Wireless-Door-Sensor"
    password: "EV2MdWtejOyK"
binary_sensor:
```

```
  - platform: gpio
    pin: GPIO02
    name: "Entrance"
    device_class: door
captive_portal:

```

Motion Sensor .yaml file

```
esphome:
  name: wireless-motion-sensor
esp8266:
  board: esp01

# Enable logging
logger:

# Enable Home Assistant API
api:
  encryption:
    key: "w8jhXudaRBhebfJS1peSgEXSuttjdcMAAJjC6V+Qtkk="

ota:
  password: "0a1bcf48974e0367a00c0716e8693734"

wifi:
  ssid: !secret wifi_ssid
  password: !secret wifi_password

  # Enable fallback hotspot (captive portal) in case wifi connection fails
  ap:
    ssid: "Wireless-Motion-Sensor"
    password: "OaR3LIXJQ3dx"
binary_sensor:
  - platform: gpio
    pin: GPIO02
    name: "Motion Sensor"
    device_class: motion
captive_portal:

```

DHT and MQ-5 Sensor .yaml file

```
esphome:
  name: wireless-temperature-gas-sensor

esp8266:
  board: nodemcuv2

# Enable logging
logger:

# Enable Home Assistant API
api:
  encryption:
    key: "+Yp9s8SQBG7tMgkz91ezpfW4seQ8kItwEwK4O1565HY="

ota:
  password: "be0260f7a82a6f8f413f6528995e6e55"

wifi:
```

```yaml
  ssid: !secret wifi_ssid
  password: !secret wifi_password

  # Enable fallback hotspot (captive portal) in case wifi connection fails
  ap:
    ssid: "Wireless-Temperature-Gas-Sensor"
    password: "dI5JrppZheGL"
sensor:
  - platform: dht
    pin: D2
    temperature:
      name: "Bedroom Room Temperature"
    humidity:
      name: "Bedroom Room Humidity"
    update_interval: 5s
  - platform: adc
    pin: A0
    name: "Bedroom Smoke Sensor"
    update_interval: 5s
    filters:
      - multiply: 100
    unit_of_measurement: "%"
    icon: "mdi:percent"
captive_portal:

```

Cost of Components				
Name	Price ($)	Link	Quantity	Total Amount
Wired Door Window Sensor Magnetic Recessed Reed Switch	6.80	https://www.amazon.in/gp/product/B07MDH1976/ref=ppx_yo_dt_b_asin_title_o02_s00?ie=UTF8&psc=1	1	6.80
HC-SR501 PIR Sensor	1.47	https://www.amazon.in/Generic-HC-SR501-Sensor-Pyroelectric-Infrared/dp/B00VNWWZM0/ref=sr_1_8?keywords=pir+sensor&qid=1655228888&sprefix=pir+s%2Caps%2C211&sr=8-8	1	1.47
DHT11 Humidity & Temperature Sensor	2.42	https://www.amazon.in/xcluma-Digital-Relative-Humidity-Temperature/dp/B072FJBF9T/ref=sr_1_3?crid=1PRCBI5OIKXXB&keywords=dht+sensor&qid=1655228949&sprefix=dht+sensor%2Caps%2C208&sr=8-3	1	2.42
MQ-5 Gas Sensor	3.58	https://www.amazon.in/REES52-liquefied-Methane-Sensor-Module/dp/B01L0FXZLE/ref=sr_1_1?crid=28J7O8RAJJ0IK&keywords=mq5+sensor&qid=1655229028&sprefix=mq5+sensor%2Caps%2C219&sr=8-1	1	3.58

Esp8266 Serial Wi-Fi Wireless Module	3.85	https://www.amazon.in/gp/product/B01EM11EU2/ref=ppx_yo_dt_b_asin_title_o00_s00?ie=UTF8&psc=1	2	7.70
ESP8266 Nodemcu	4.49	https://www.amazon.in/gp/product/B07262H53W/ref=ppx_yo_dt_b_asin_title_o01_s00?ie=UTF8&psc=1	1	4.49
Micro to USB 2.0 Data cable	1.28	https://www.amazon.in/gp/product/B08HXMD4K1/ref=ppx_yo_dt_b_asin_title_o03_s00?ie=UTF8&psc=1	1	1.28
USB to ESP8266 Serial Wireless WiFi Module Development Board	3.01	https://www.amazon.in/gp/product/B06X19QLVP/ref=ppx_yo_dt_b_asin_title_o03_s00?ie=UTF8&psc=1	1	3.01
Raspberry Pi	40.00	https://www.raspberrypi.com/products/raspberry-pi-4-model-b/	1	40.00
1080p USB Camera	12.98	https://www.amazon.in/Tukzer-Microphone-Auto-Focus-Rotatable-Plug-n-Play/dp/B09FFCWRNK/ref=sr_1_8?crid=O1349X4G5EHT&keywords=usb+camera&qid=1655229335&sprefix=usb+camera%2Caps%2C212&sr=8-8	1	12.98
Breadboard Jumper Wires	2.08	https://www.amazon.in/Tukzer-Microphone-Auto-Focus-Rotatable-Plug-n-Play/dp/B09FFCWRNK/ref=sr_1_8?crid=O1349X4G5EHT&keywords=usb+camera&qid=1655229335&sprefix=usb+camera%2Caps%2C212&sr=8-8	1	2.08
Breadboard	1.67	https://www.amazon.in/Generic-Elementz-Solderless-Piecesb-Circuit/dp/B00MC1CCZQ/ref=sr_1_5?crid=2PJ3NZC3UHH36&keywords=breadboard&qid=1655228634&sprefix=breadboard%2Caps%2C213&sr=8-5	2	3.34
Battery Charging Module	8.93	https://www.amazon.in/CentIoT®-Lithium-Rechargeable-Charging-Raspberry/dp/B07RSQNDTH/ref=sr_1_1?crid=2QHEXMALA62JI&keywords=3.3v+battery+esp&qid=1655228735&sprefix=3.3v+battery+esp%2Caps%2C194&sr=8-1	4	35.72

3.3-5V Passive Buzzer Alarm Module	2.55	https://www.amazon.in/REES52-3-3-5V-Passive-Buzzer-Arduino/dp/B01I1NHCIU/ref=sr_1_2?crid=2CJKZ34094LQR&keywords=buzzer+module&qid=1655230210&sprefix=buzzer+modul%2Caps%2C206&sr=8-2	1	2.55
Total			19	120.62

Power Consumption			
Devices	Quantity	Power Consumed (mAh)	Total Consumption (mAh)
Wired Door Window Sensor Magnetic Recessed Reed Switch	1	0	0
HC-SR501 PIR Sensor	1	1	1
DHT11 Humidity & Temperature Sensor	1	3	3
MQ-5 Gas Sensor	1	1	1
Esp8266 Serial Wi-Fi Wireless Module	2	8	16
ESP8266 Nodemcu	1	5	5
Micro to USB 2.0 Data cable	1	0	0
USB to ESP8266 Serial Wireless WiFi Module Development Board	1	0	0
Raspberry Pi	1	540	540
1080p USB Camera	1	150	150
Breadboard Jumper Wires	1	0	0
Breadboard	2	0	0
Battery Charging Module	4		0
3.3-5V Passive Buzzer Alarm Module	1	5	5
Total	19	713	721