

IoT HOME SECURITY SYSTEM

ECE 442: Internet of Things and Cyber Physical Systems

Part 2: Progress Report

Due Date: 06/09/2022

BY:

Nikhil A. Chaganti, Alan Palayil, Hamad Abdelrahim

Problem Statement

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.

Summary

The goal of our project is to develop a smart alarm system which will disarm using facial recognition triggered using motion detection to decrease power consumption. The system should arm itself by detecting if the user is away and send notifications depending on the detections and presence in the building. 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. The system will comprise of various sensor which will be used to monitor and report the users via mobile-app.

Design Overview

The system is designed to be in sleep mode until motion is detected within a certain range of the entrance. The individual comes up to the module and facial recognition/ pin input is giving to disarm the system. The system has presence detection to notify users for alerts. The system can monitor activity and has the potential to track and create automations based on user's routines.

Hardware

- Raspberry Pi
 - It will be used as the local server with Home Assistant installed. The Pi will be used to collect, store and upload data to the Home Assistant cloud for remote accessibility. The Pi will be designed to host a private network for all the nodes to communicate in case of internet shortage.
- ESP8266 NodeMCU
 - It will be the hub for all sensors and actuators to be connected to and will transfer the data from the nodes to the Raspberry Pi using the WiFi. It will communicate with the sensors via ESP communications to increase the range of the system.
- USB Camera
 - It will be used at the entrance of the building for the facial detection of the individuals trying to enter. It will be connected to the Raspberry Pi for the OpenCV software to run facial recognition.
- Motion Sensor/Ultrasonic Sensor
 - It will be used to activate the camera for motion detection to decrease the overall power consumption of the system. It will be connected to the Raspberry Pi.
- DHT and MQ-5 Sensor
 - It will be used to monitor the temperature and CO concentration in the building to detect fire or gas leak. It will be connected to the NodeMCU to transfer the data.
- Magnetic Reed Switch
 - It will be attached to the entrance door and used to detect forced entry to the building. It will be connected to the NodeMCU to send the status of the system.

Software

- Home Assistant
 - It's an open-sourced platform used to put local control and privacy for home automation projects. It was selected for its versatility with various modules and mobile app. We are using it to connect the sensors and actuators to the cloud using services like ESPHome and MQTT Broker for monitoring and control.
- OpenCV
 - It's an open-sourced platform for computer vision with cross-platform compatibility. We will be working on repositories like Deepstack and Double Take to add facial recognition to the Home Assistant.
- IFTTT and Amazon Alexa/ Google Home
 - It will be implemented in the back end to the Home Assistant to enable cross-platform integration with various automations and services like Google services.

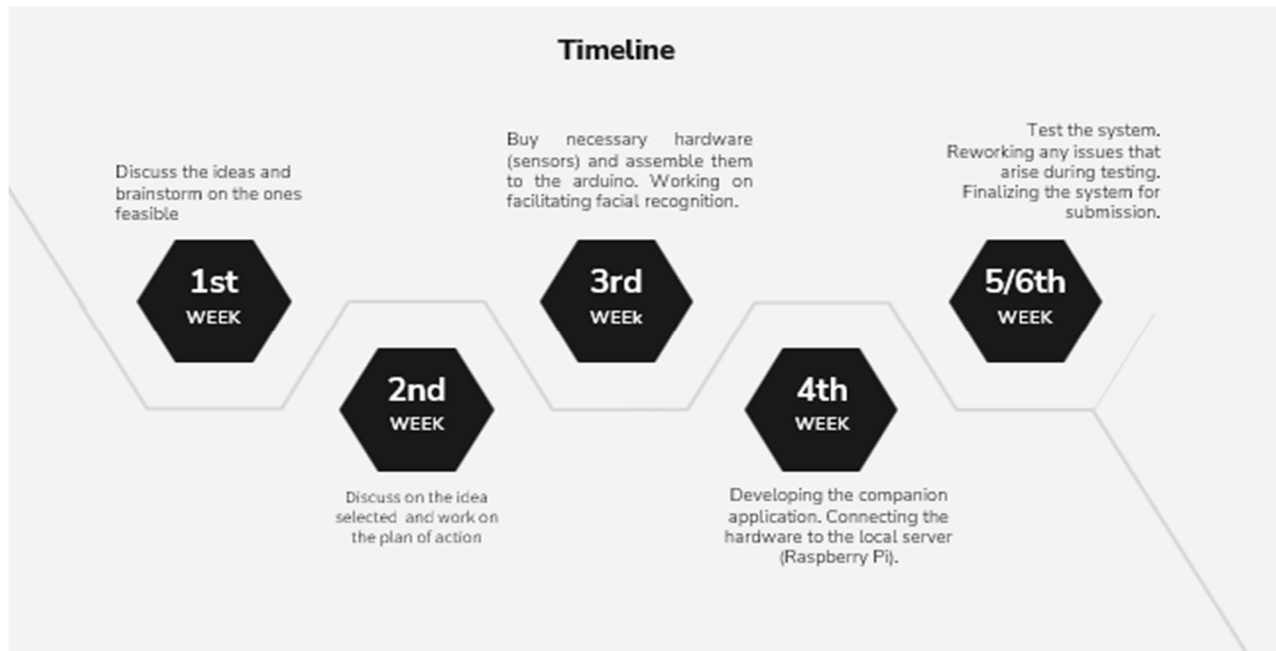
Communication

Currently the Raspberry Pi is connected to the ethernet to provide a physical internet connection between the Pi which acts as the server and the cloud for remote accessibility. The sensors are being connected to the NodeMCU module currently through wires and later using ESP32 modules for the device to communicate with the NodeMCU via the Wi-Fi. The NodeMCU is connected to the Raspberry Pi using the MQTT protocol via local network. The Raspberry Pi is set up to publish a hotspot during internet shortage and the sensors can still send the data to the server which can upload the data once the internet connection is back online.

Progress and 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 set-up 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 algorithm.
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.



Work Distribution and Tasks

Table with work distribution

Team Member	Task	Due Date
Hamad Abdelrahim	Research and implementation of various sensors. Research over different automation and presence detection protocols.	06/12/2022
Alan Palayil	Design an existing system using the ESP8266 NodeMCU Research on integration of system for multi node setup	06/12/2022
Nikhil Aditya Chaganti	Research implementation of facial recognition software on Raspberry Pi. Integration of software within OpenCV.	06/12/2022

Future Work

With time, the system can be built from scratch and have easy integration with other software and services. The system can include fire/ gas leak monitoring using various sensors. The system to be integrated with Google Home and Amazon Alexa to increase automations 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.

Technical Challenges

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.

Facial Recognition: With the core mechanism of this system being facial recognition, accuracy is vital and depends on the image resolution of the camera.

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

Conclusion

The project is going at a good pace with all team members working on their respective tasks with a slight delay during the lab submission week. The system is designed to send custom notification to users present in-home and while the user leaves automatically detect and arms the system. With the system architecture to disarm the system using facial detection and during faults have manual keypad for inputs. With the system currently being implemented to communicate via WiFi using the ESP protocol with a local server-client set-up using Raspberry Pi.

References

- Home Assistant
 - <https://www.home-assistant.io/getting-started/>
 - https://companion.home-assistant.io/docs/getting_started/
 - <https://esphome.io/index.html>
 - <https://github.com/smarthomejunkie/Home-Assistant-Tutorials>
- Home Assistant automations
 - <https://slack-labs.com/2020/04/10/how-i-secured-my-home-using-home-assistant-part-one/>
 - <https://community.home-assistant.io/t/i-can-t-send-notification-to-pushover/86639/5>
- OpenCV
 - <https://everythingsmarthome.co.uk/face-recognition-just-got-easier-home-assistant-double-take-guide/>
 - <https://www.home-assistant.io/integrations/facebox/>
 - <https://github.com/robmarkcole/HASS-Deepstack-face>
- Presence Detection
 - <https://github.com/ESPresense/ESPresense/releases>
 - <http://mqtt-explorer.com>
- IFTTT automations
 - <https://leonardosmarthomemakers.com/how-to-create-a-diy-alarm-system-with-home-assistant/>