Bachelor Thesis in
Electrical Engineering
06 2020

IoT Based Home Monitoring System

Tabu Sravani Guduru

Suryanarayana Murthy Tatavarthy

Dept. of Mathematics and Nature Sciences
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden

Contact Information:
Authors:

Tabu Sravani Guduru
E-mail: tagu19@student.bth.se

Suryanarayana Murthy Tatavarthy
E-mail: suta19@student.bth.se

University Supervisor:
Prof. Wlodek J. Kulesza

Dept. of Mathematics and Nature Sciences	Website:	www.bth.se/didd
Blekinge Institute of Technology	Phone:	+46 455 38 50 00
SE–371 79 Karlskrona, Sweden	Fax:	+46 455 38 50 57

Thesis Examiner:
Irina Gertsovich

Dept. of Mathematics and Nature Sciences
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden

Abstract

Home surveillance is a major concern in this day and age as with the rapid increase in the technology around us. There is a need to get updated with new possibilities to make our lives better and easy. Some cases and situations exhibit the need for home monitoring. So, we set out to discover a solution to this problem of home monitoring.

Generally, we have issues with pets and kids alike such as approaching dangerous places like electric switches, stairs, and hot things. Not only a problem with kids and pets but also a chance of burglary and stranger's unusual activities. To overcome this problem we are designing and prototyping a system to keep an eye on kids, pets, and older people. This system is used in other applications like theft monitoring. The device can monitor the field all the time. In this way, this system helps in-home monitoring.

The system consists of Arduino, which is the brain of the system, the PIR sensor, ESP32-CAM, and buzzer. The PIR sensor detects motion then gives input to the Arduino. Arduino gives output to the ESP32-CAM and buzzer. The ESP32-CAM can be activated and sends information to the user through web Interface and the application. The user can see the video streaming on the PC screen or any other display. The buzzer can emit a high volume signal indicating that "there is an alert at the home".

Keywords: Arduino, Application, ESP32-CAM, PIR sensor, Web Interface.

Acknowledgments

We are really grateful because we managed to complete our Home Monitoring System project within the given time. We sincerely thank our lecturer Wlodek J. Kulesza for the guidance and encouragement in finishing this project. Last but not least, we would like to express our gratitude to our friends for the support and willingness to spend some time with us to fill in the questionnaires.

Sincerely,

Tabu Sravani Guduru
Suryanarayana Murthy Tatavarthy

Contents

List of Figures

List of Tables

Acronyms

clk clock

EEPROM Electrically Erasable Programmable Read-Only Memory

GSM Global System for Mobile communication

HCI Human to Computer Interface

HMI Human to Machine Interface

IOT Internet of Things

IP Internet Protocol

LCD Liquid Crystal Display

LED Light Emitting Diode

MMI Machine Interface

PC Personal Computer

PIR Passive Infrared Sensor

RAM Random Access Memory

SRAM Static Random Access Memory

HSM Home Security Monitoring

UDD User Driven Design

Chapter 1

Introduction

Home surveillance is a major concern in this day and age as with the rapid increase in the technology around us there is a need to get updated with it. There are many cases in which home monitoring is needed and we set out to discover a solution to the problem that we researched. For example, kids often play and not aware of the dangers around them. There is a chance of approaching dangerous places like electric switches, stairs, and hot things. It is not only a danger for kids but also for pets, and also there will be chances for theft activities. There will be a loss of valuable items and documents in case of any theft activities. Pets also will eat unhealthy foods like plastic or rubber, which will make them sick. All these problems can be solved by home monitorization.

There are many solutions to home surveillance systems in the market and what they do is simply either record and store or else record and stream it to a remote server. There is nothing much you can do about with this type of system rather than just sitting and watching about what is happening in your home. There are other instances, which require interactive surveillance such as kids between the ages of 2 years and 5 years where the parents are preoccupied with some work and require to monitor them using the surveillance system. There are cases when older people are alone in the house and need surveillance just like kids. However, there is a need for interactive surveillance for all these scenarios and there is none product available in the market, which perfectly fits into this description.

Our approach to the problem is quite simple as it lets the user monitor their homes 24/7 from a remote location and in case of unusual activities, the user is alerted through the web interface. Then the user can monitor all events from the web camera. If necessary, the users can interact with the help of ambulance or police through his mobile. We have developed a system using Arduino as the main micro-controller and a PIR sensor, which detects the movement in the house. An ESP32 Camera module is installed, which can show the user what is going on in the house as soon as any movement is detected. A buzzer is also installed, which can be activated by the user to scare off any intruders or pets if there are any unwanted activities taking place.

In Chapter 2, we have written about the survey of related works. From this chapter, we understood the pros and cons of all the components and we tried to pick the good components for our project and the technologies that are used and existing are mentioned. The problem statement, objectives, and the main contributions of our project are stated and discussed in Chapter 3. Our team members' contributions are presented. In Chapter 4, we show system modeling and we presented how our system can fulfill all the required functionalities and the flowchart. The description of the components involved in the project is mentioned in Chapter 5. The web streaming is also described. In Chapter 6, we have discussed the various tests that we made on the components to check their compatibility and their accuracy. The results of the tests are briefly described in the same chapter. Finally, in Chapter 7, we concluded the project work and ended our thesis with the future scope, which is presently suitable. Finally, in the appendix, we have put down the coding.

Chapter 2

Survey of Related Works

The related works are divided into sub-subjects of *Home security*, *Existing solutions* and *Used technologies.*

2.1 Home security issues

The survey shows that there are many problems faced due to the lack of a home security system. Some of the major problems are:

- *Chances of burglaries*: The main problem due to lack of security system is burglary. "According to research, there are 4 burglaries per minute, 240 an hour and 6000 a day" [1]. These cases are most seen in the places where there is no home security system [2].
- *Lack of monitoring*: In homes, there will be elder people, small children, and pets, they need to be monitored regularly. There are so many accidents in our country that took place, Due to lack of monitoring. For example, many little children are dying due to falling into manholes [3].
- *Lack of medical assistance*: Many house owners, especially when they are elder may have difficulties with their health conditions and need to be monitored. If there is any home security system in the home, then it will alert the user's relatives or can be able to call for a medical assistance [4].

2.2 Existing solutions

Some of the existing solutions available for home security system are presented in this section.

One of the proposal developed for the monitoring of houses and to provide security is "Arduino Home security system" [5]. Some of the components involved in this project are: *PIR sensor*, *LEDs*, *camera*, *relay* for on and off of camera and *keypad* to operate the system by entering the pin. The report consists of a built cardboard house for an example of how it works, and the prototype is placed on that cardboard house. This project provides security for the home by triggering an alarm when any door is opened or if any movement is detected in the room. Arduino Mega 2560 is the main part of the project. This solution helps to detect theft activities or unusual activities taking place at the home. This project is limited to small areas. The prototype of this project is given in Figure 2.1.

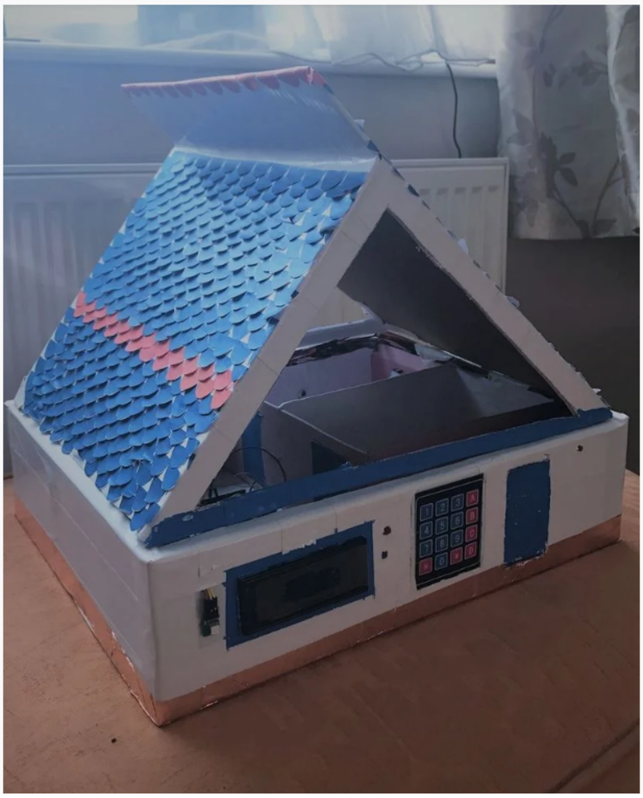


Figure 2.1: Model of the *Arduino home security system* [5]

Another project, which is used to monitor the home and provide security is the "DIY Arduino Home Security System" [6]. This project is similar to the above one but also, it will send email notifications to the user through Cayenne. They have also used ROHM's Temperature Sensor and Barometric Pressure Sensor to detect heat in the room. Some of the components involved in this project are Arduino Mega 2560, buzzer, Grove LEDs, and PIR sensor. The notification is given to the user on the following basis: If Blue LED blinks, it indicates motion detected. If Green LED blinks, it indicated a triggered alarm. If Red LED blinks it indicates both the detected motion and a triggered alarm. The model of the system is shown below in Figure 2.2.

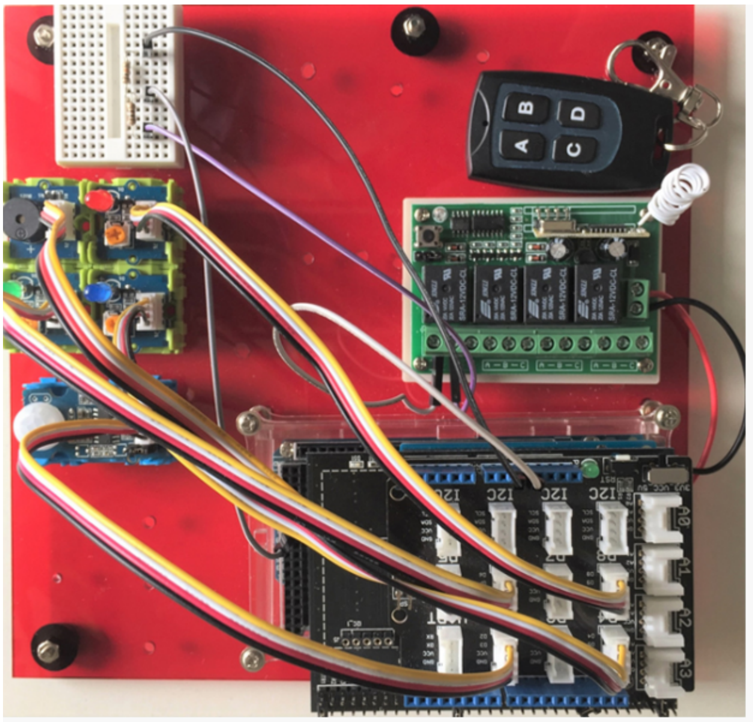


Figure 2.2: Picture of the prototype [6]

Another proposal, which is related to monitor the home and to provide security is low cost multi level home security system [7]. The system consists of Arduino UNO, fingerprint sensor, PIR sensor, buzzer, keypad, GSM module for communication with the user, LCD display and webcam, which is connected in home 24/7. The system is designed in such a way that if any of the user wants to go inside, they have to give their password/fingerprint. If any intruder breaks in to the house, then the user will get information through the GSM module connected to Arduino. If the PIR sensor detects any unusual activity, then the lights will be turned on and alarm triggers. The block diagram of the system is shown in Figure 2.3.

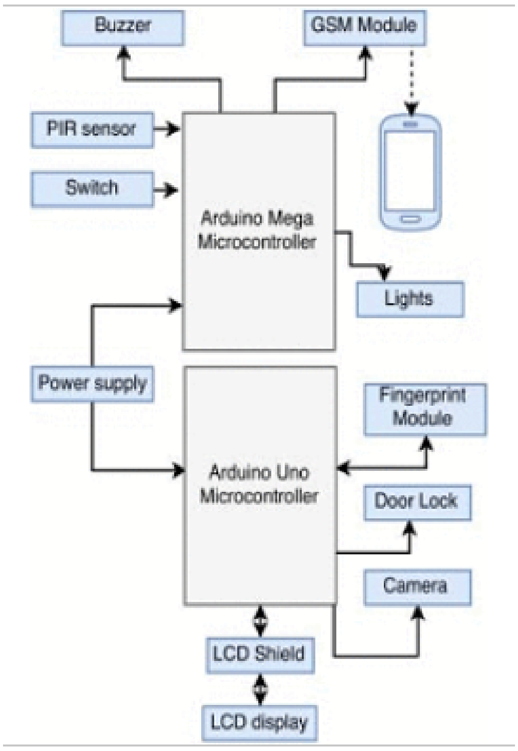


Figure 2.3: Block diagram of the design [7]

In the security system proposed in [8], cameras were installed in every room of the house/apartment. Through these cameras, the users used to monitor the whole house/apartment. The users can detect motion,video, and face recognition. The USB camera is used to face recognition, network camera is used for motion detection, and CCD camera is used for video capturing. Users can use a mobile phone/PC to monitor the home security status in real-time from a far distance.

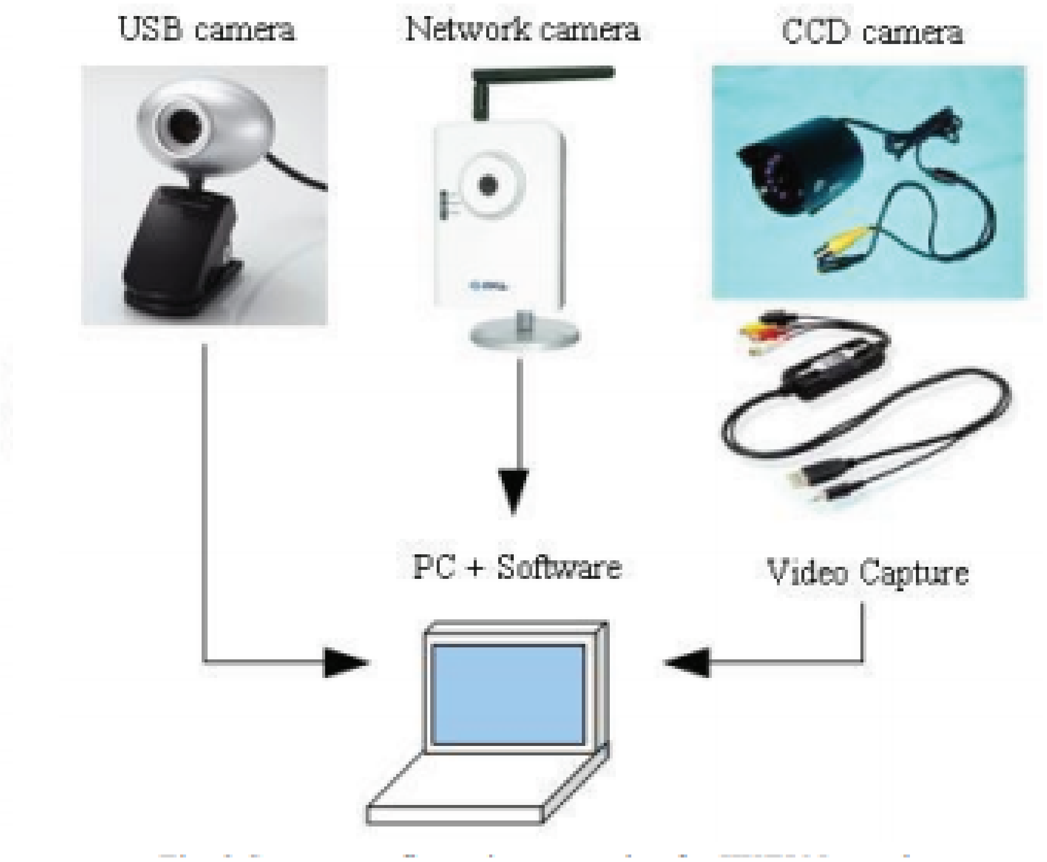


Figure 2.4: Simple design of the security system [8]

2.3 Available technologies

Technologies of this project interest can be divided into three categories: *sensors*, *communication methods* and *controllers*.

2.3.1 Sensors

Sensors are sensing devices that perform input function in a system as they *sense* the changes in a quantity. Sensors convert the physical event into the electrical signal [9]. In this section some types of sensors are presented.

Gas sensors are used to detect the presence of gas in the surrounding atmosphere. There are many operation mechanisms used in gas sensing techniques and each technique has a different type of mechanism involved in the each type like [10]:

- Electrochemical,
- Catalytic bead (pellistor),

- Photo-ionization,
- Infrared point,
- Infrared imaging,
- Metal oxide semiconductor,
- Ultrasonic,
- Holographic.

To understand how the *gas sensors* work let us consider the MQ-6 gas sensor as an example. The ability of a gas sensor to detect gases depends on the gas sensing layer to conduct current. Here, the gas sensing layer contains SnO_2. Normally, the atmosphere will contain more oxygen than combustible gases. The oxygen particles attract the free electrons present in SnO_2. But, when there is toxic or combustive gas present, the reducing gas reacts with oxygen resulting in free electrons resulting in the flow of current. The amount of concentration of gas is proportional to the flow of current [11].

PIR (Passive Infra-Red) sensor measures infrared light radiating from objects. PIR sensor gives information about the general movement but it does not give any information about the activity that caused it [12].

The flame sensor is used to detect the presence of flame. It alarms, if the fire present in less than 3-meter proximity to the sensor. This sensor/detector can be built with an electronic circuit using a receiver like electromagnetic radiation. A flame emits infrared radiations on their ignition. With the help of these radiations, the sensor indicates the presence of a flame. This sensor uses the infrared flame flash method, which allows the sensor to work through a coating of oil, dust, water vapor, otherwise ice [13].

The light sensor is also called a photo resistor or Light Dependent Resistor (LDR). This sensor helps to detect the brightness level in the surrounding. It works in such a way that when the light is incident on the resistor the resistance is decreased [14].

2.3.2 Communication technologies

In this section, we briefly present the most common communication standards.

Wi-Fi: Wi-Fi is a wireless networking technology that uses radio waves to provide wireless high-speed internet and network connections. The 802.11 standard provides different radio frequency ranges for the use in the Wi-Fi communication: 900 MHz, 2.4 GHz, 5 GHz, 5.9 GHz, 60 GHz. Where the 2.4 GHz frequency band is generally used.

Type	Power	Max power level	Designed operating range	Sample devices
Class 1	High	100 mW (20 dBm)	Upto 100 m (328 feet)	USB adapters, access points
Class 1.5 (low energy)	Medium high	10 mW (10 dBm)	Upto 30 m (100 feet), but typically 5 m (16 feet)	Beacons, wearable sensors
Class 2	Medium	2.5 mW (4 dBm)	Upto 10 m (33 feet)	Mobile devices, Bluetooth adapters, smart card readers
Class 3	Low	1 mW (0 dBm)	Upto 1 m (3 feet)	Bluetooth adapters

Table 2.1: Bluetooth ranges table [17]

From Wi-Fi, higher standards MIMO was introduced. MIMO means Multiple Input Multiple Output. This allows the devices to have multiple antennas on the transmitter and receiver. This helps the Wi-Fi technology in giving much faster speed and greater range. Nowadays Wi-Fi embedded chips are used in almost every electronic equipment like drones, computers, laptops, phones, and in many accessories [15].

Bluetooth: Bluetooth frequency range is between 2.402 GHz and 2.480 GHz. Bluetooth uses a radio technology called *frequency-hopping spread spectrum.* Bluetooth divides the data and sends it through packets [16]. The range of Bluetooth is divided by classes. This is shown in Table 2.1. Generally, class 2 is used in mobile phones and its range is about 10 meters. In industries class 1 is used and its range is about 100 meters. Class 3 has a range of 1 meter.

Chapter 3
Methodology

In this chapter we discuss three main areas. First we present the problems that we are facing due to burglaries and the problems with the old technologies. Then, we introduce the objectives that we defined to overcome the problems. Finally, we summarise the main contributions and team members' individual contribution to the project.

3.1 Problem statement

Apart from the problems that we mentioned in Chapter 2 such as burglaries, medical assistance, safety of kids and pets and stranger unusual activities, the problem that we consider here is to provide an application-based IoT system for the users. By installing the application, the users can have access to his/her home. They should be able to watch video streaming. The used technologies do not apply a proper user-interface while the technologies that provide a user interface such as app and web interface are not economical. Therefore, there is a need to find a economically reasonable solution for home monitoring with a remote access.

3.2 Objectives of the proposal

The objective of our project is to develop the safety monitoring system to keep an eye on kids, pets and stranger activities in the house. The system should be able to detect the unusual movement. The users must be able to watch the surveillance through user-interface (application, web-interface). We assume that the system should consists of following elements such as Arduino, buzzer, ESP 32 web cameras and web interface, as illustrated in Figure 3.1.

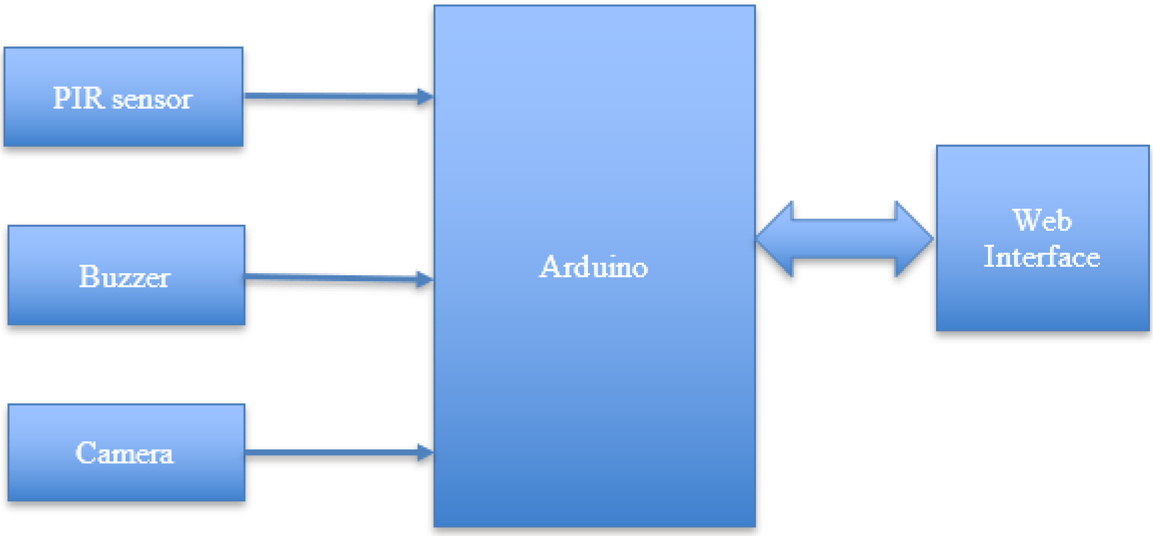


Figure 3.1: Block diagram of the design

PIR sensor and camera provide information to the Arduino. Through the web interface, the Arduino communicates with the users, which can monitor their homes 24/7.

We adopt the following principles to our system [18]:

- It monitors a field for a movement.
- If there is any unusual movement, it activates a camera and alarms the user.
- Web interface facilitates the communication between the users and the system.
- The system monitors the house 24/7.

3.3 Contribution of the project

The main contributions of our project could be summarised as follow:

- To design a system in such a way that the householders can monitor there house when they are away from home.
- To implement the system on Arduino based platform.
- To develop an application, through which the user gets the notifications about activities at home.

We divided project tasks between us depending upon authors' skills in developing hardware and software along with writing documentation. The software part (coding, interface) was assigned to Tabu Sravani Guduru. The design and implementation of the hardware part was assigned to Suryanarayana Murthy Tatavarthy. For documentation, both the authors played an equal role.

Chapter 4

System Design

This chapter is divided into two parts. The first part concerns User Driven Design and the second part deals with modelling of the system.

4.1 User driven design (UDD)

The User-Driven Design guides a process of product development and facilitates communication among future users, stakeholders, and designers. It is a constant design process that encompasses validation from the user every step of the way. It could be divided into two parts: *problem formulation* and *product development*.

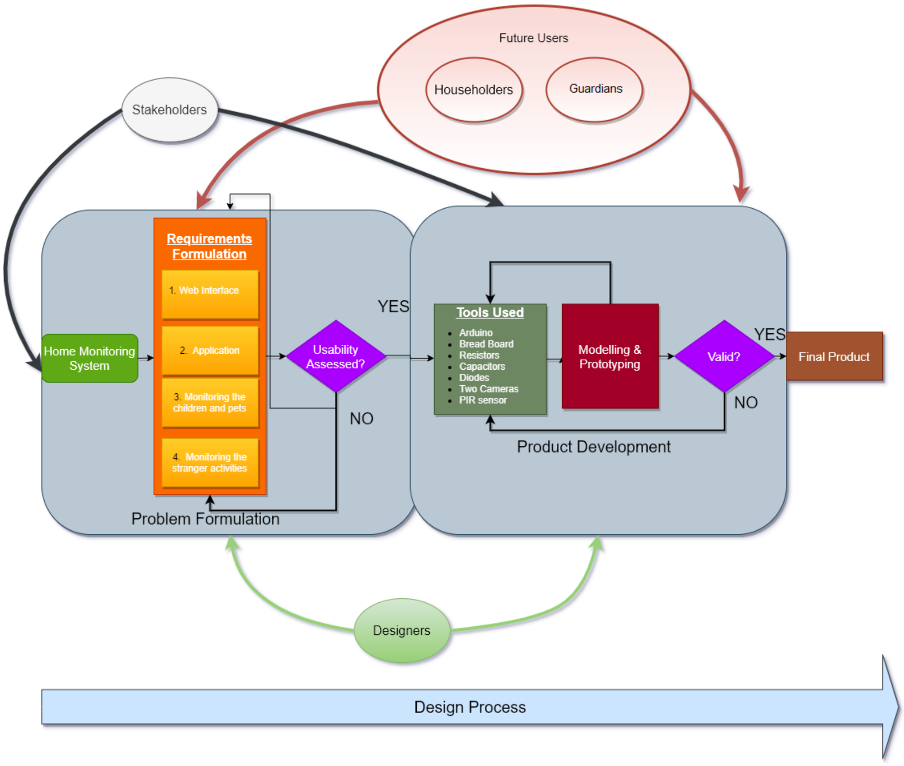


Figure 4.1: User Driven Design schema

Problem formulation. The users of our system would be householders and guardians. By implementing the system, the users must be able to monitor the kids, pets, and stranger's unusual activities. They must get an alert notification in case of any unusual activity. They must be able to watch the video streaming in the web interface or application. The requirements of the system are user-interface (e.g. application or web browser), detection of motion, and monitoring the kids' activity, pets, and stranger's unusual activities, see Figure 4.1.

Product development. The tools required to achieve the requirements discussed in problem formulation are Arduino, buzzer, ESP32 camera, PIR sensor. Now, the block diagram, schematic diagram, and flow chart for building the prototype should be developed. After building the prototype, the system must pass the tests. After all the tests, we can say that the system is useful to provide safety by monitoring the kids, pets, and stranger's unusual activities. It can alert the user in case of any dangerous conditions through the user-interface and through the buzzer [19]. In this way, the user-driven design can be characterized as a multi-stage problem-solving process.

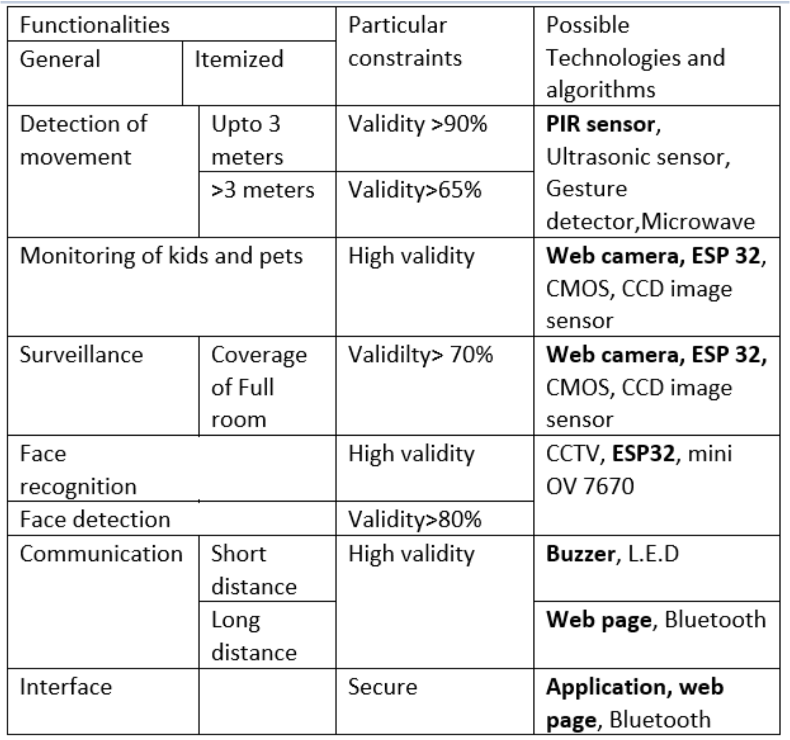
The functionalities of our system are shown in Table 4.1. It shows all functionalities both the general and itemized and particular constraints along with the related technologies and algorithms.

- *Detection of movement*: The PIR sensor present in the system is responsible for the movement detection. With the help of the infrared mechanism, the pyro-electric sensor detects the movement. The proximity of the PIR sensor is 3 meters. There are similar sensors such as ultrasonic sensors, gesture detectors. They can also be used for the detection of motion.

- *Monitoring of pets and kids*, *Surveillance* and, *Face detection and recognition.* The functionalities mentioned here come under the same section. The ESP32 camera is used to monitor the home (kids, pets, and stranger's unusual activities). It is responsible for video streaming and face recognition and detection. If the PIR sensor detects any motion, then the ESP32 camera will be triggered from a sleep mode and will start video streaming. The video can be watched in the user-interface.

- *Communication.* The system must be able to warn the user in a case of any alert conditions. The system should be able to communicate the users, irrespective of their location distance from the home (long or short). Hence, for the short-distance, the buzzer is used. The buzzer will indicate that "there is an alert". User-interface is used for long-distance communication. The user will get an alert notification through the user-interface which is an application.

- *Interface.* The user interface is a medium between the user and the system. It allows the users to access the data from anywhere and at anytime. The application interface and web interface are developed. The users can either log in to the application or web browser to get access to the data or to watch video streaming.



Functionalities		Particular constraints	Possible Technologies and algorithms
General	Itemized		
Detection of movement	Upto 3 meters	Validity >90%	**PIR sensor**, Ultrasonic sensor, Gesture detector,Microwave
	>3 meters	Validity>65%	
Monitoring of kids and pets		High validity	**Web camera, ESP 32**, CMOS, CCD image sensor
Surveillance	Coverage of Full room	Validilty> 70%	**Web camera, ESP 32,** CMOS, CCD image sensor
Face recognition		High validity	CCTV, **ESP32**, mini OV 7670
Face detection		Validity>80%	
Communication	Short distance	High validity	**Buzzer**, L.E.D
	Long distance		**Web page**, Bluetooth
Interface		Secure	**Application, web page**, Bluetooth

Table 4.1: Functionalities table of the system

4.2 System block diagram

Figure 3.1 represents the block diagram of the system. The left part of the diagram represents the sensors connected to the Arduino and right part represents the user-interface. The Arduino is the the central component of the system to which the three sensors: PIR sensor, buzzer, camera are connected. The Arduino is responsible for the input and output flow in the system. PIR sensor is used to detect the motion. The ESP32-CAM is used for video streaming and buzzer is either used to alert the users or to scare the intruders in case of any unusual activity.

If the PIR sensor senses any motion, then the ESP32-Camera will start video streaming and buzzer will be triggered to alert the user that there is an unusual activity in the home. Now, the Arduino will process data to web interface/application regarding the alert. The user can see the video streaming through web interface/application to observe the unusual activity.

4.3 Flowchart

Flowchart describes the working of our system. The flowchart starts with the terminator. When the power supply is given to the system, the Arduino and PIR sensor are activated. The PIR sensor is ready to detect the motion and the information will be processed to the Arduino. Figure 4.2 represents the flowchart of the system. The process performs in two ways based on motion detection by the PIR sensor:

- $PIR\ sensor = 1$, if the PIR sensor is active, it means that a motion is detected. The ESP32 camera will be enabled from a sleep mode and start working. Then the Arduino will alert the buzzer. It will be high for 2 minutes to alert the surroundings and an alert will be sent to the users through the application. The alert also can be seen in the web interface. The users have access to the video streaming record of the ESP32 camera.

- $PIR\ sensor = 0$, if PIR sensor is passive, it means there is no motion. In such cases, the safe conditions can be seen in the web browser and also in the application. Even if there are safe conditions, if the users wants to watch the video streaming, they can do it through the user-interface.

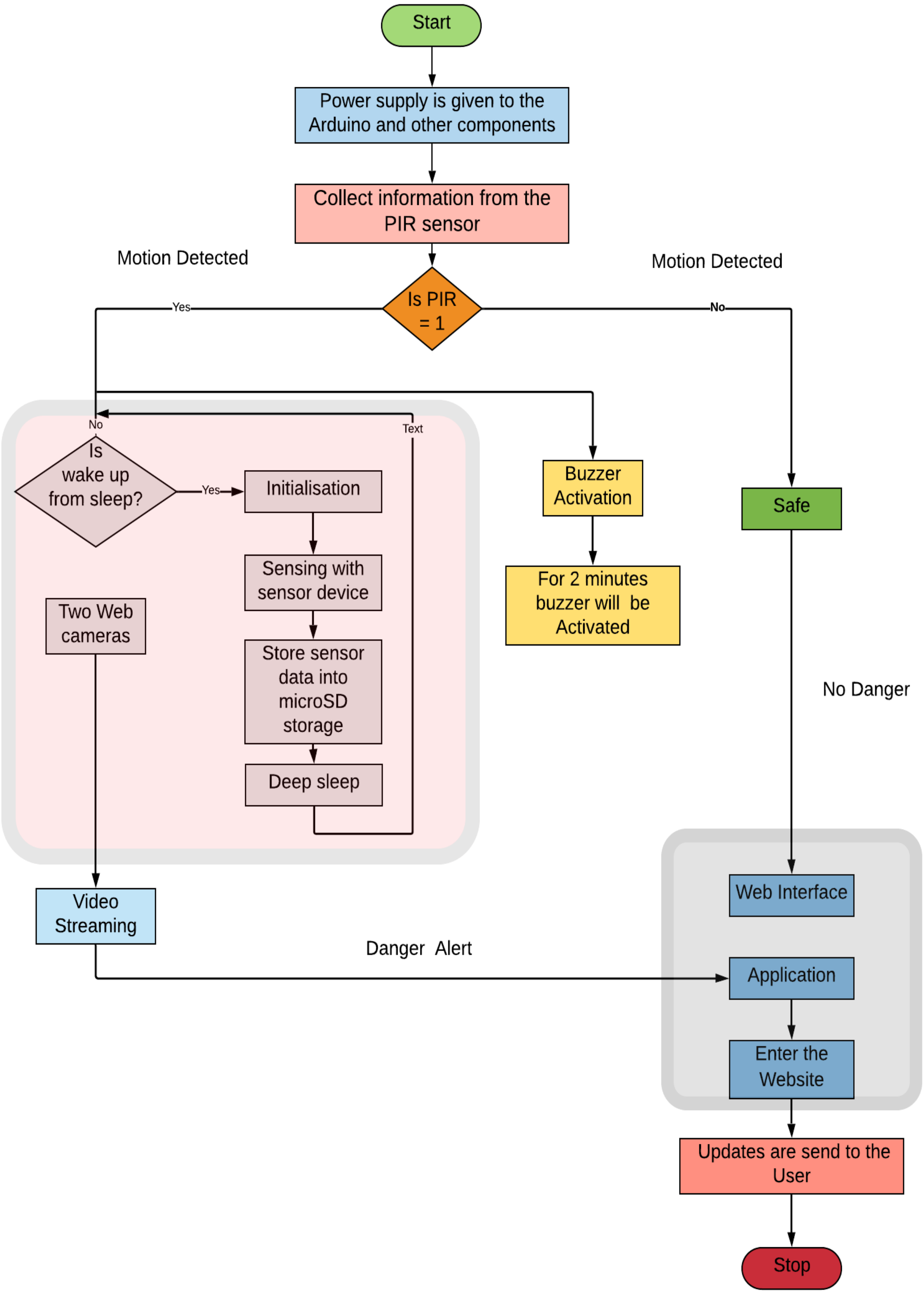

Start
Power supply is given to the Arduino and other components
Collect information from the PIR sensor
Motion Detected
Motion Detected
Is PIR = 1
Yes
No
No
Text
Is wake up from sleep?
Yes
Initialisation
Sensing with sensor device
Store sensor data into microSD storage
Deep sleep
Two Web cameras
Buzzer Activation
For 2 minutes buzzer will be Activated
Safe
No Danger
Video Streaming
Danger Alert
Web Interface
Application
Enter the Website
Updates are send to the User
Stop

Figure 4.2: Flowchart of the system

Chapter 5

System Implementation

This chapter is divided into two parts such as hardware description and the software description. Hardware description concerns implementation and validation of the system. In the implementation, we discuss the types of components that were used in this system, and whereas in validation, we verify system functionalities. The software description consists of two parts: code description and application description.

5.1 Implementation of the system

This section consists of a schematic diagram and descriptions of components that are used and finally, the prototype of the system is presented.

5.1.1 Schematic diagram

The schematic diagram shown in Figure 5.1 explains the connections between the Arduino and the other components such as camera module, PIR sensor, and the buzzer, where referred symbols are as follow:

- SG1 refers to buzzer,
- GND refers to ground,
- *Vin* refers to input voltage,
- *Tx* refers to transmitter pin,
- *Rx* refers to receiver pin.

The connection of the components to the Arduino are:

Connection of Arduino with the computer The computer is connected to the Arduino using the USB cable type A/B and the reset pin of the Arduino is connected to the ground pin of the Arduino.

Connection pins of ESP32 Camera with the Arduino:

- An 5V pin of Arduino connected to Vcc pin of ESP32 camera and ground pin of Arduino to ground pin of ESP32 camera.
- An D01 pin of ESP 32 camera connected to the ground pin of Arduino.
- A receiver pin (Rx) of ESP32 camera connected to a receiver pin of Arduino and the transmitter pin (Tx) of ESP32 camera connected to the transmitter pin of Arduino.

Connection pins of buzzer with the Arduino.

- A positive pin of buzzer connected to the digital 8th pin of the Arduino.
- A ground pin of buzzer connected to the ground pin of the Arduino.

Connection pins of PIR sensor with the Arduino.

- An Vcc pin of PIR sensor connected to the 5V pin of the Arduino.
- A ground pin of PIR sensor connected to the ground pin of the Arduino.
- The Dout (digital out) pin of PIR sensor connected to the digital 2nd pin of the Arduino.

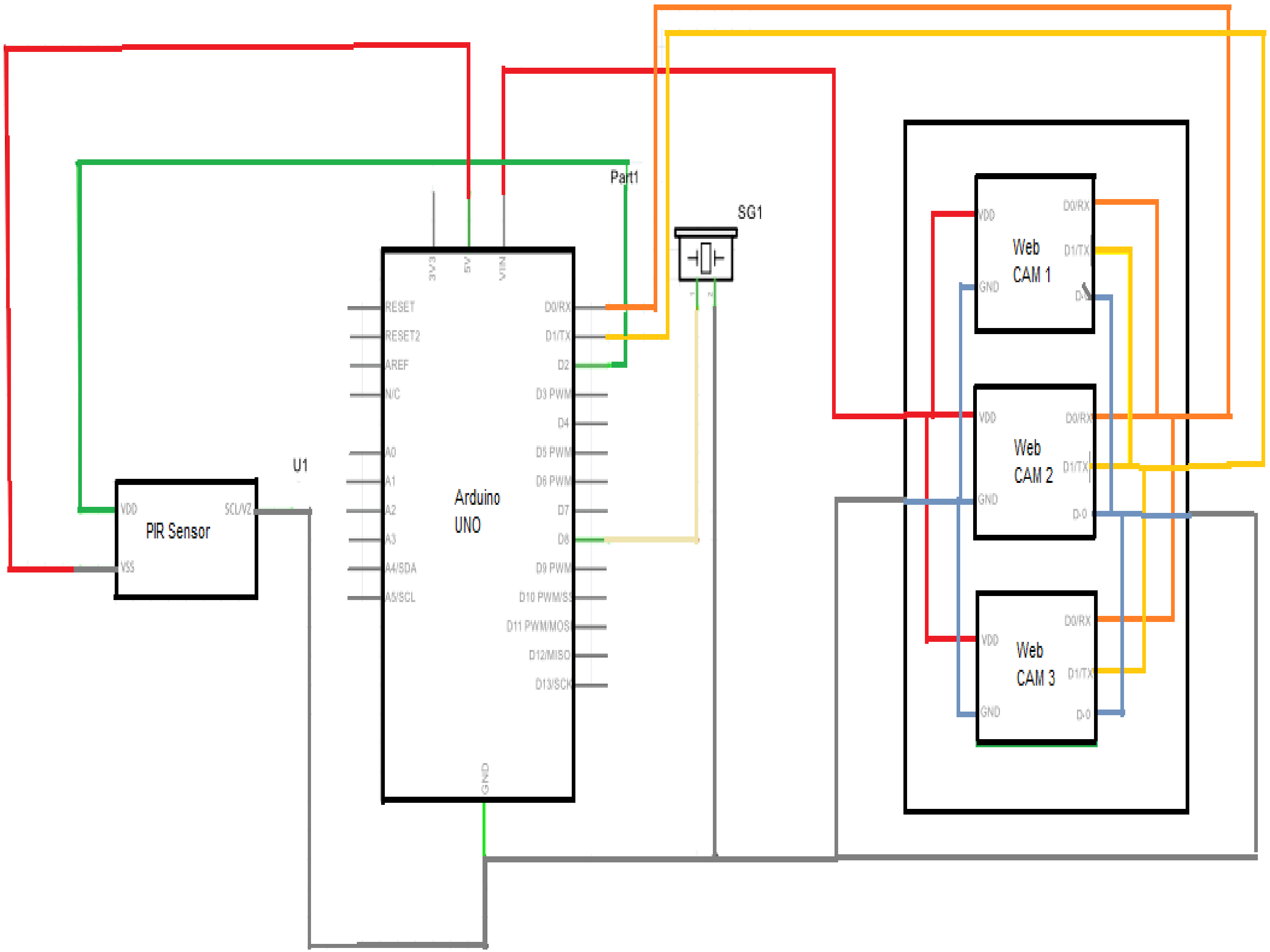


Figure 5.1: Schematic diagram of the system

5.1.2 Arduino uno

Arduino uno is a micro-controller based on the ATmega328. It has 32 kB ISP flash memory, 1 kB EEPROM, 2 kB SRAM, 23 general purpose I/O lines. It operates between 1.8 V to 5.5 V. Arduino Uno has 14 digital input/output pins, and 6 of them can be used as pulse width modulation (PWM) output pins and 6 as analog input.

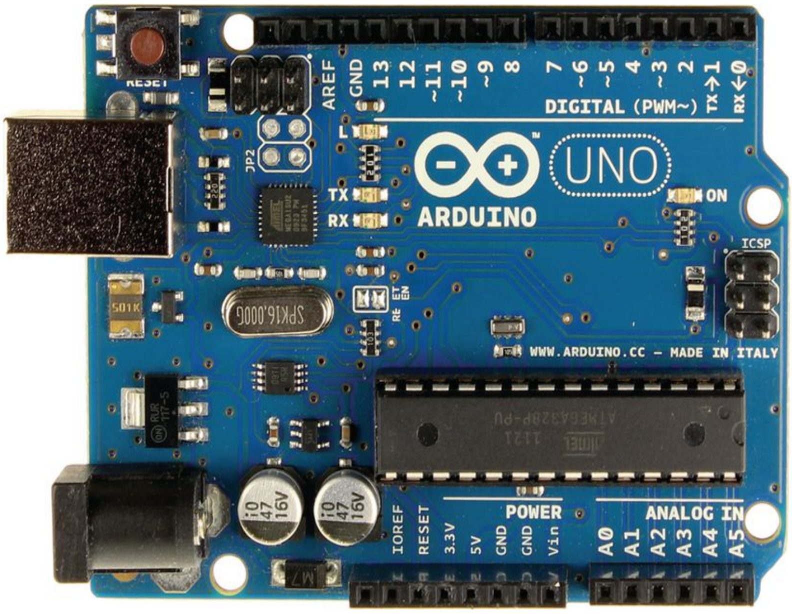


Figure 5.2: Arduino UNO [13]

- **Pins 0 (Rx) and 1 (Tx):** Receiver (Rx) pin and Transmitter (Tx) pin are used to receive and transmit Transistor-Transistor Logic (TTL) serial data.
- **Pins 2 and 3:** These are called as external interrupt pins. These pins are configured to change in the value when they are triggered.
- **Pins 3,5,6,9 and 11:** These pins provide an 8-bit PWM (Pulse Width Modulation) output.
- **Pins 10(SS), 11 (MOSI), 12 (MISO) and 13 (SCK):** These pins are used for SPI communication.
- **Pin 13:** This pin has built-in LED. If pin 13 is receiving any power then the LED will be switched 'ON', otherwise it is 'OFF'.
- **Reset pin:** It is used to reset the micro-controller.

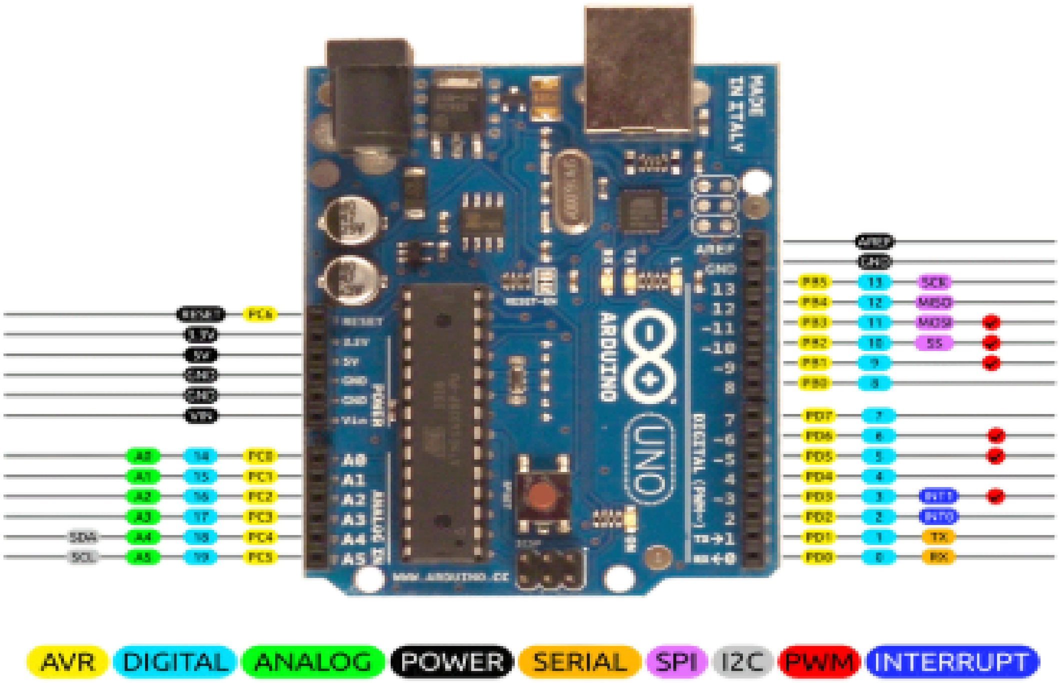


Figure 5.3: Ports description of Arduino UNO [13]

It also has a USB slot, a power jack, an ICSP header and a reset button. It is programmed using Integrated Development Environment (IDE) software [13].

5.1.3 ESP32-CAM

The ESP32-CAM is a small camera with module including a micro SD-card slot. The images shot by the ESP32-CAM can be saved in a micro SD-card.

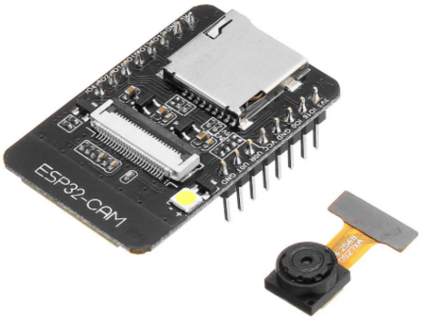


Figure 5.4: ESP32-CAM [20]

It is a 2 megapixel camera of dimension of 40.5 mm x 27 mm x 4.5 mm. The ESP32 web camera internal memory includes:

- 448 kB of ROM for booting and core functions.
- 520 kB of on-chip SRAM for data and instructions.
- Embedded flash.

There are a total of 39 digital pins on the ESP32 where 4 out of 39 are input-only. It can support 18-channels for 12-bit ADC, 2-channel for 8-bit DAC. It has 16 channels for PWM signal generation. The ports of the ESP32 is shown in Figure 5.5.

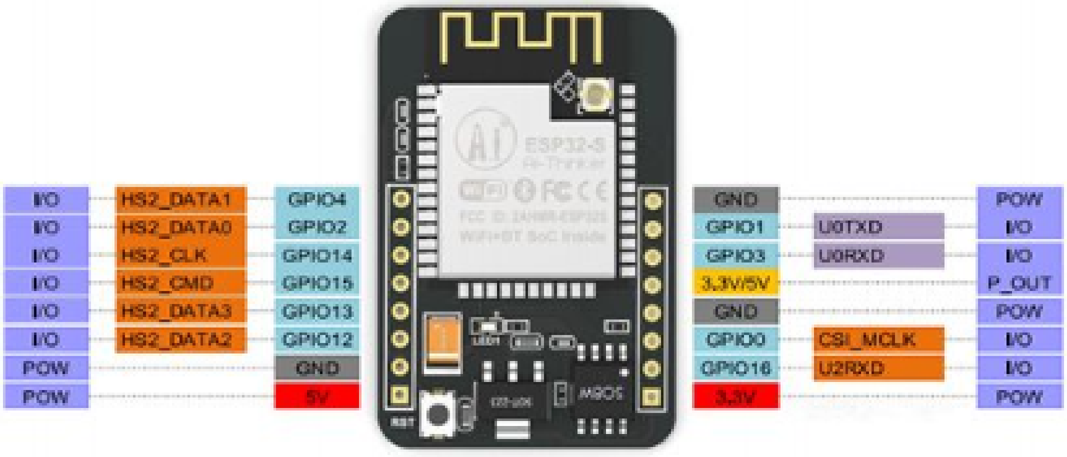
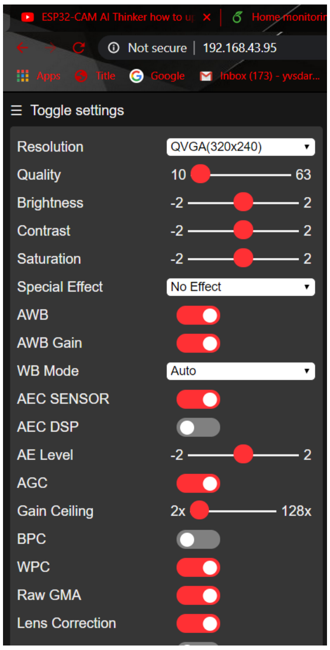


Figure 5.5: ESP32-CAM ports description [20]

ESP32 comes with the in-built Wi-Fi and Bluetooth module. This Wi-Fi module is used to assign the IP address. The web streaming settings when this camera module connected with the Arduino is represented in Figure 5.6 [20].



☰ Toggle settings

Resolution QVGA(320x240)
Quality 10 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 ☑

Figure 5.6: Camera settings shown in web browser

5.1.4 PIR sensor

PIR means Passive Infra-Red sensor. The PIR sensor module is shown in Figure 5.7.

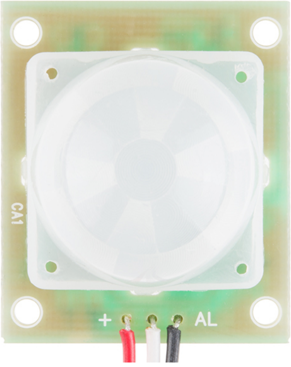


Figure 5.7: PIR Sensor [21]

This PIR sensor module consists of a pyro-electric sensor shown in Figure 5.8. It generates energy when it is exposed to heat. When the human or animal body comes in the range of this sensor it will detect a movement of infrared radiation.

The supply voltage of the PIR sensor is 5 V and the output voltage is 3.3 V TTL. This PIR sensor has high sensitivity. It can distinguish between object

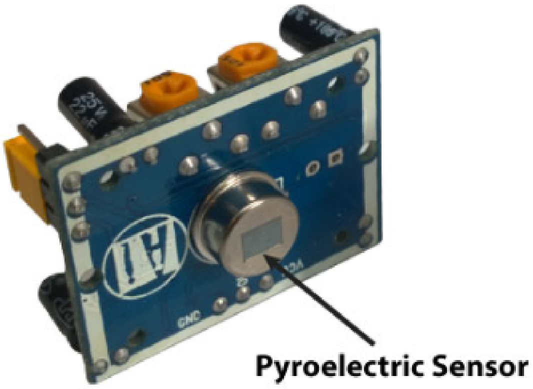


Figure 5.8: Pyro-electric sensor when the cover is removed [21]

movement and human movement. It has a cover distance of about 7 m. The power consumption of this sensor is low, just about 65 mA. Operating temperature of this sensor is from -20°C to +80°C.

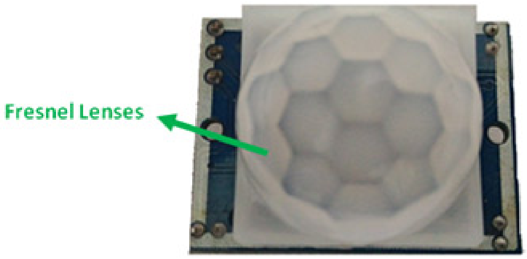


Figure 5.9: Fresnel Lens [21]

This module consists of a Fresnel lenses, which cover the pyro-electric sensor. The Fresnel lenses help to focus the infrared signals to the pyro-electric sensor [21].

5.1.5 Buzzer

The buzzer is an electrical device with an integrated structure and produces buzzing sound. They are of three types: mechanical, electro-mechanical and piezo-electric. These types are classified based upon the type of tones they produce. Figure 5.10 shows a piezo-electric buzzer, which is used in our project. Its operating voltage is 3 V - 16 V DC. The frequency range of piezo-electric buzzer is 31 Hz - 65535 Hz and can differ from type to type [22].

Figure 5.10: Buzzer [21]

5.2 Prototype

Figure 5.11 represents the prototype of our system. It shows the design of the system, the connection between Arduino and other components (e.g: buzzer, PIR sensor, ESP32-camera). It can be easily installed in houses, in any room (living, dining) depending upon the user's necessity.

The necessary conditions for installing the prototype are:

- The user must have access to the Internet.
- The prototype must be given power supply.

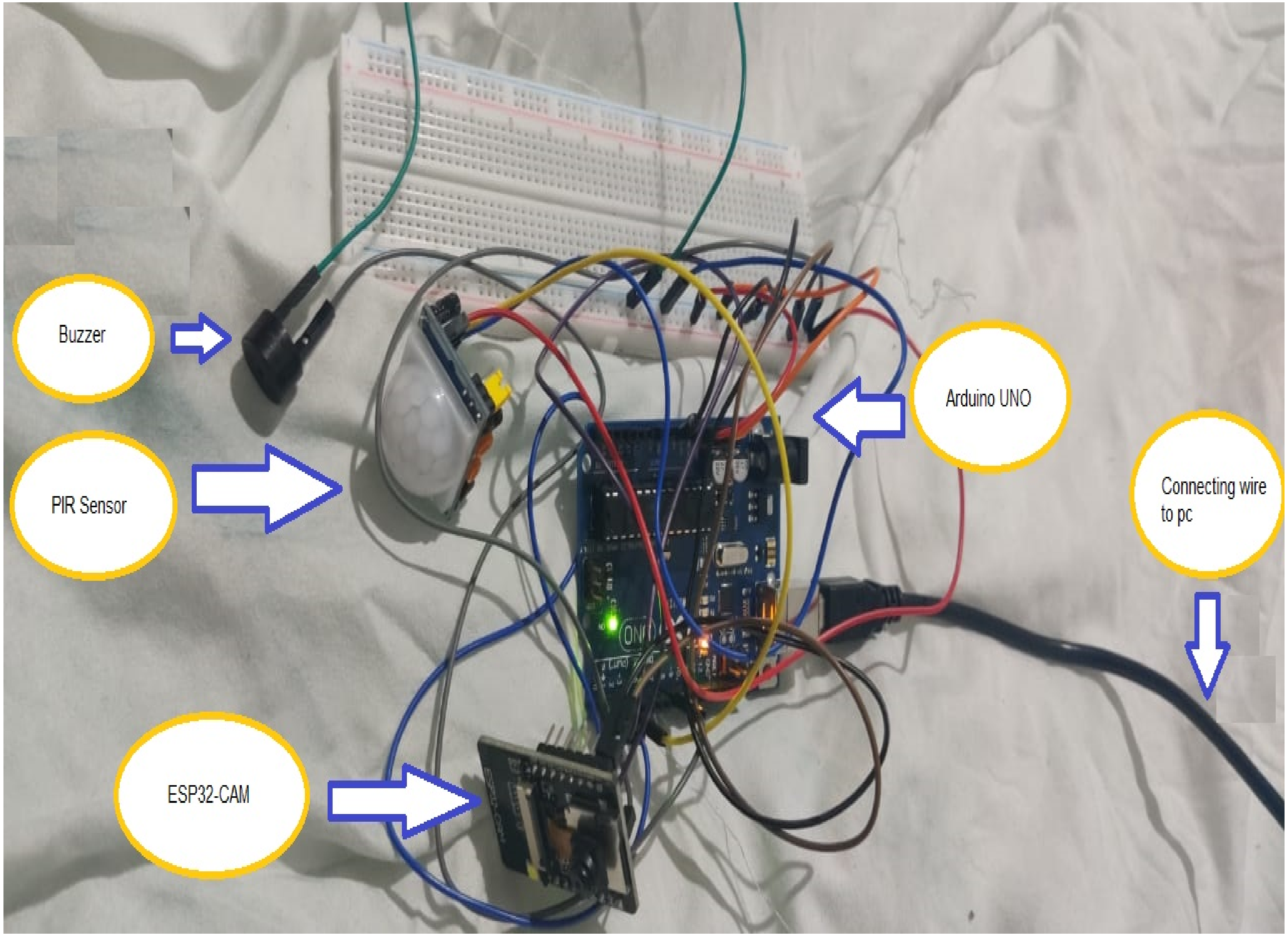


Figure 5.11: Prototype

The design and functioning of the system can be seen in the operational mock-up represented in Figure 5.12. Through this picture, the user can easily understand the advantages of having the safety monitoring system installed in his/her home. The picture briefly explains the involved components in the home and the interlink between the user and the system through the web interface.

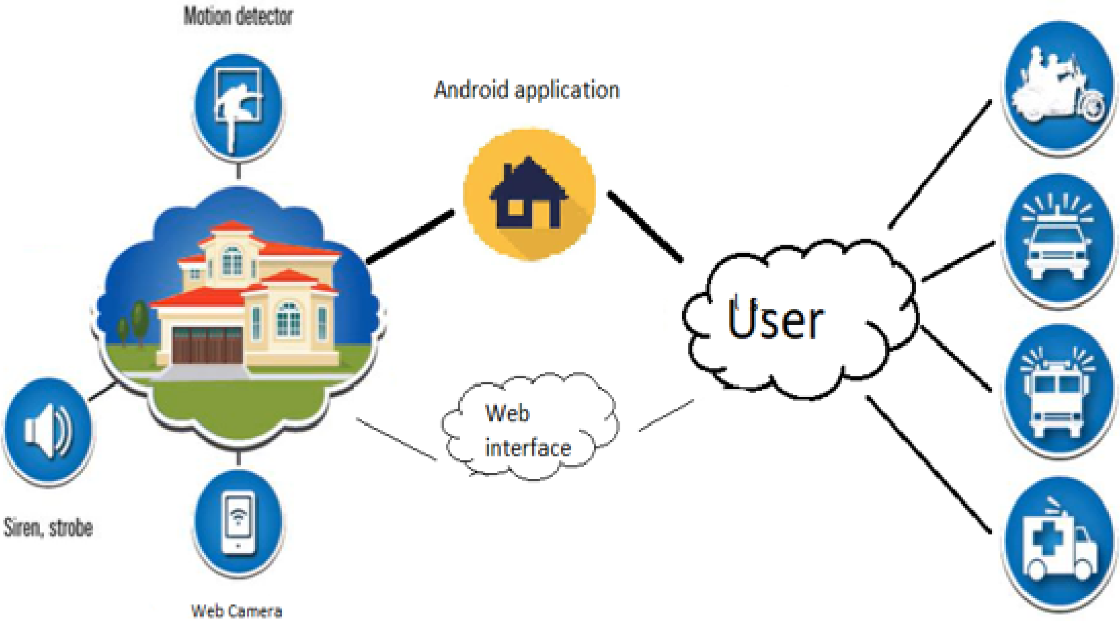
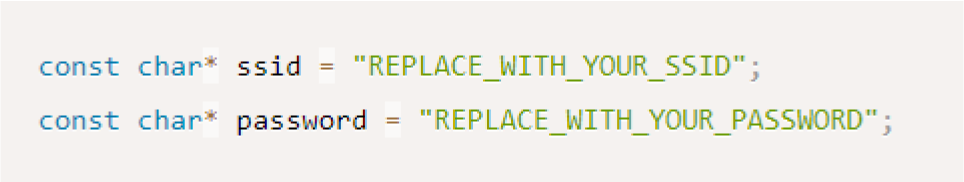


Figure 5.12: System operational mock-up

5.3 Code description

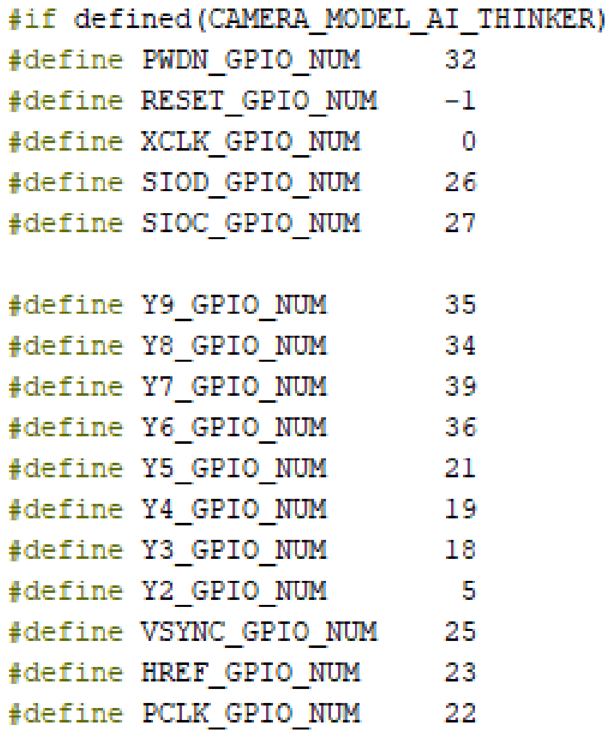
Coding is the crucial part of this project. The first part is a code for ESP32-CAM. This code requires to insert network credentials to make the connection between the system and the network. Creating an IP address is necessarily for the program to run, see Figure 5.13.



```
const char* ssid = "REPLACE_WITH_YOUR_SSID";
const char* password = "REPLACE_WITH_YOUR_PASSWORD";
```

Figure 5.13: Coding for connection between system and web network

The following part of code defines the Arduino's pins like *power*, *clock*, $y2$ to $y9$ etc. The pins must be predefined to get access for the usage of them, see in Figure 5.14.



```
#if defined(CAMERA_MODEL_AI_THINKER)
#define PWDN_GPIO_NUM     32
#define RESET_GPIO_NUM    -1
#define XCLK_GPIO_NUM      0
#define SIOD_GPIO_NUM     26
#define SIOC_GPIO_NUM     27

#define Y9_GPIO_NUM       35
#define Y8_GPIO_NUM       34
#define Y7_GPIO_NUM       39
#define Y6_GPIO_NUM       36
#define Y5_GPIO_NUM       21
#define Y4_GPIO_NUM       19
#define Y3_GPIO_NUM       18
#define Y2_GPIO_NUM        5
#define VSYNC_GPIO_NUM    25
#define HREF_GPIO_NUM     23
#define PCLK_GPIO_NUM     22
```

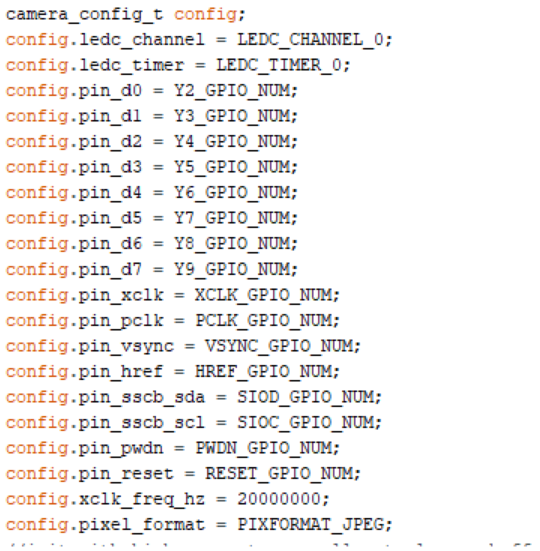
Figure 5.14: Coding for pins define

Generally, pins are configured to assign the pin mode output or input. An output pin will have low impedance state. The properties of the pin vary depending upon the mode they are configured to such as input, input pull up and output. Through this configuration, the connection is made between Arduino and other components. We can see the configuration in Figure 5.15.

After entering the network credentials, the network is connected with the system and then the whole process starts, and we will get an IP address on the screen. We need to run the code to see this whole process as shown in Figure 5.16.

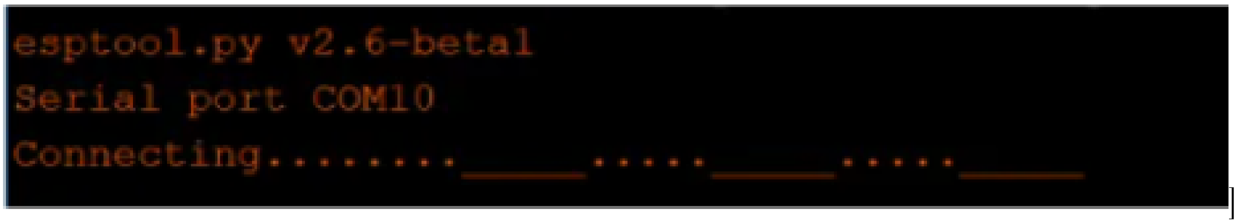
After connecting, when the setup is finished we can see *Ready to stream* dialog in the output console.

In serial monitor, if the PIR sensor does not detect any motion then the serial monitor shows continuous dots. If the PIR sensor detects any motion then in the serial monitor we can see *Motion detected.* After that IP address is generated and will appear in the serial monitor (output of the Arduino IDE). The image of the serial monitor is shown in Figure 5.17.



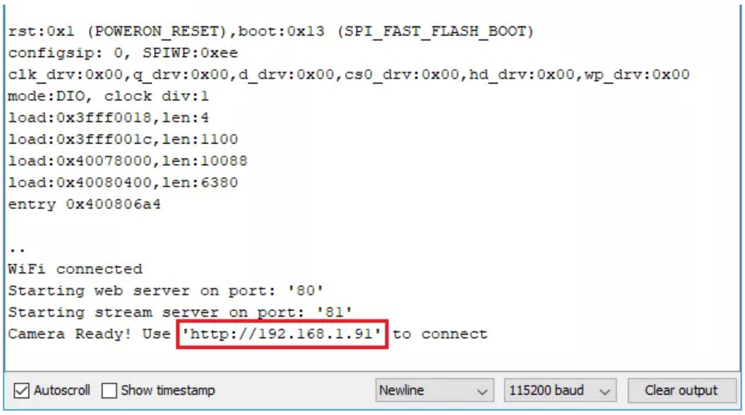
```
camera_config_t config;
config.ledc_channel = LEDC_CHANNEL_0;
config.ledc_timer = LEDC_TIMER_0;
config.pin_d0 = Y2_GPIO_NUM;
config.pin_d1 = Y3_GPIO_NUM;
config.pin_d2 = Y4_GPIO_NUM;
config.pin_d3 = Y5_GPIO_NUM;
config.pin_d4 = Y6_GPIO_NUM;
config.pin_d5 = Y7_GPIO_NUM;
config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM;
config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;
```

Figure 5.15: Coding for pins configuration



```
esptool.py v2.6-beta1
Serial port COM10
Connecting........_____....._____....._____
```

Figure 5.16: Web camera results



```
rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1100
load:0x40078000,len:10088
load:0x40080400,len:6380
entry 0x400806a4

..
WiFi connected
Starting web server on port: '80'
Starting stream server on port: '81'
Camera Ready! Use 'http://192.168.1.91' to connect
```

☑ Autoscroll □ Show timestamp Newline 115200 baud Clear output

Figure 5.17: Results

After entering the IP address (the IP address which is generated by control panel of the router) in the web browser , we can see the video streaming in the browser as shown in Figure 6.1.

5.4 Web interface and application

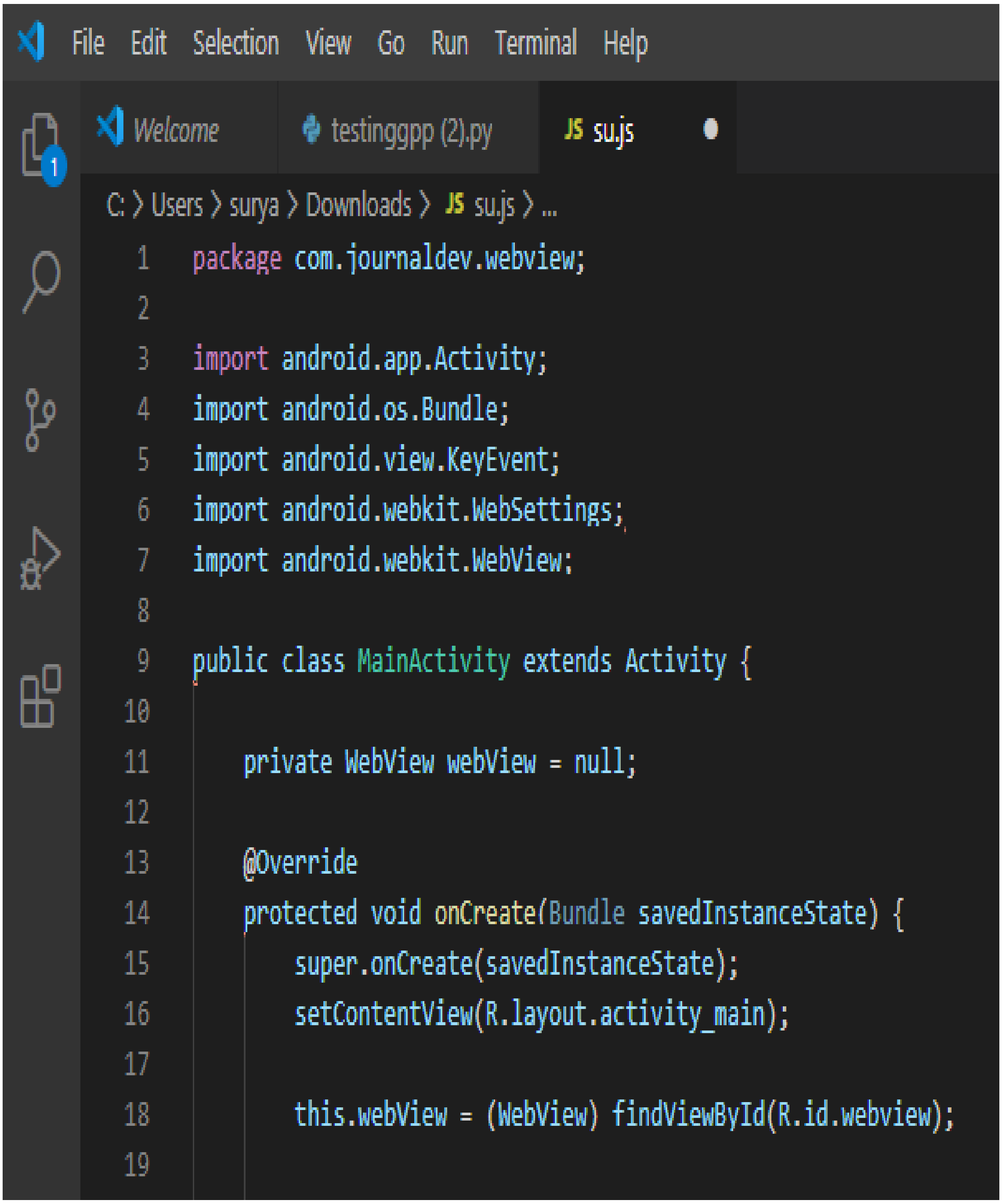
An interface is a kind of medium between the connectors, e.g. medium between the users and the system. Through the *web interface*, users get the information to anywhere at any time. The web interface helps the user to make or command or create or change the system. The users can make their desired result with their hand tips at home or from a far distance. There are many types of interfaces such as Human to Machine Interface (HMI), Man to Machine Interface (MMI), Human to Computer Interface (HCI), which can be still further classified. The main purpose of the web interface / user interface is to connect the user and the system even from far distance. Routers such as gateways, access points consist a control panels, which provide access to interface through entering the IP address to the web browser.

In our system, we used the web interface to connect the user and the project. Our interface falls into the type of Human to Computer Interface (HCI). Through this, the users are able to monitor or control their home through the mobile phone. There are many types of interfaces standards such as Bluetooth, IP address connection, and also the Application Program Interface, API. Here, in our system, we have developed both the IP address connection and the application program interface. Through the interfaces, the data can be seen by the user through his mobile phone or PC. In the web interface, the user will get access to the data by entering the IP address in the web browser. The web interface is less secure.

The application that we implemented in our project is named HMS (Home monitoring system). The users can install the application on their devices (mobile phones or tablets). They need to login by entering the required credentials for eg (IP address generated by Arduino IDE). It has an easy installation process and users can get hands-on experience with the usage of the application easily. Users get access to the data from the moment they logged in to application. Compared to the web browser, the application interface is secure. The users can watch video streaming if there are any alert situations. Even, if there are no dangerous conditions, users can start the streaming manually by the start streaming option.

Through this application, the users will get access to the system and he can be able to monitor his/her home all the time. The coding required to built *the application interface* is shown in Figures 5.18, Figure 5.19.

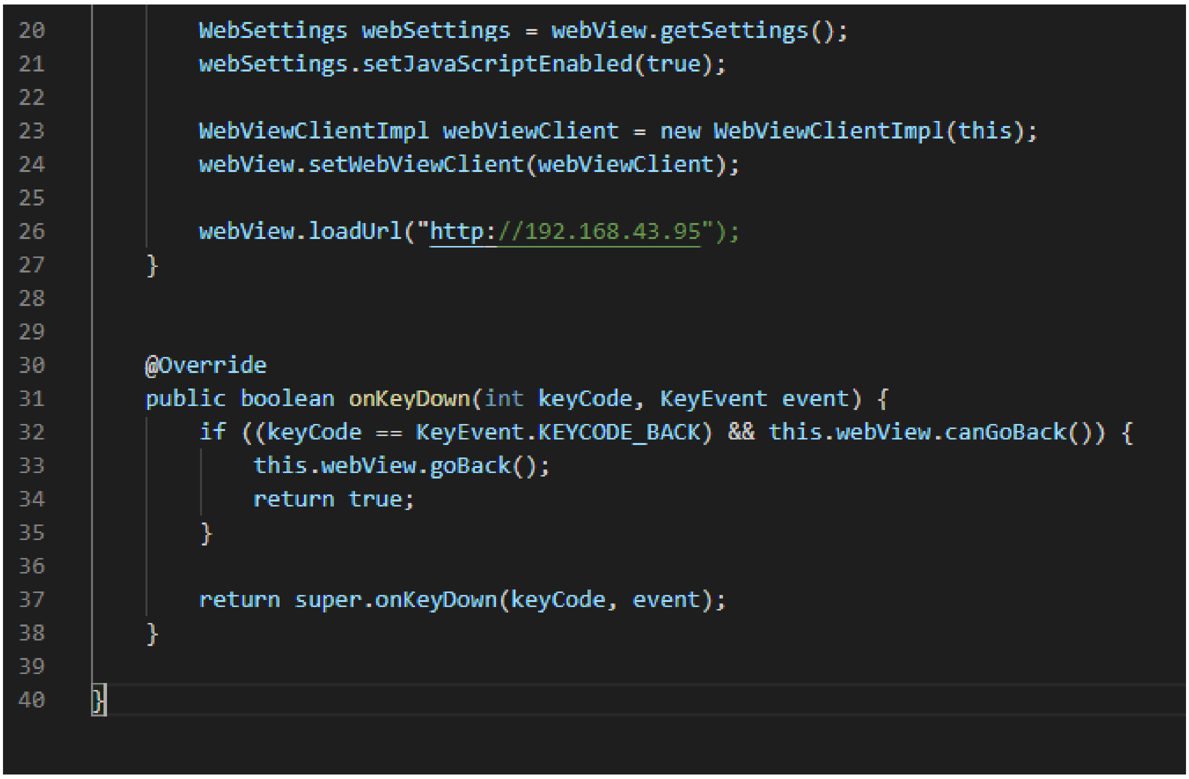
The coding for the application is written in "Java script". The first step in the coding is importing libraries such as (android app activity, os bundle, view key event, web kit web settings, web kit web view) to get access for all the functions



```
package com.journaldev.webview;

import android.app.Activity;
import android.os.Bundle;
import android.view.KeyEvent;
import android.webkit.WebSettings;
import android.webkit.WebView;

public class MainActivity extends Activity {

    private WebView webView = null;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        this.webView = (WebView) findViewById(R.id.webview);

```

Figure 5.18: Importing libraries for application coding

associated with them. The next step is to built a class that contains the functions such as *onCreate*, *onKeydown* to create the application.



```
        WebSettings webSettings = webView.getSettings();
        webSettings.setJavaScriptEnabled(true);

        WebViewClientImpl webViewClient = new WebViewClientImpl(this);
        webView.setWebViewClient(webViewClient);

        webView.loadUrl("http://192.168.43.95");
    }


    @Override
    public boolean onKeyDown(int keyCode, KeyEvent event) {
        if ((keyCode == KeyEvent.KEYCODE_BACK) && this.webView.canGoBack()) {
            this.webView.goBack();
            return true;
        }

        return super.onKeyDown(keyCode, event);
    }

}
```

Figure 5.19: Application coding

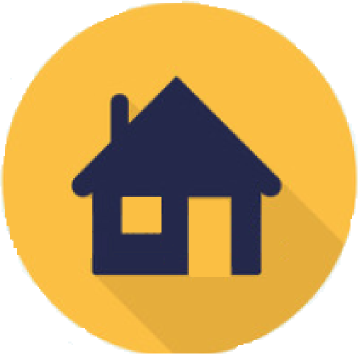


Figure 5.20: Application logo

Application uses less amount of storage (11.10 Mb) space. The logo of application is shown in Figure 5.20. The main page of application is shown in Figure 5.21.

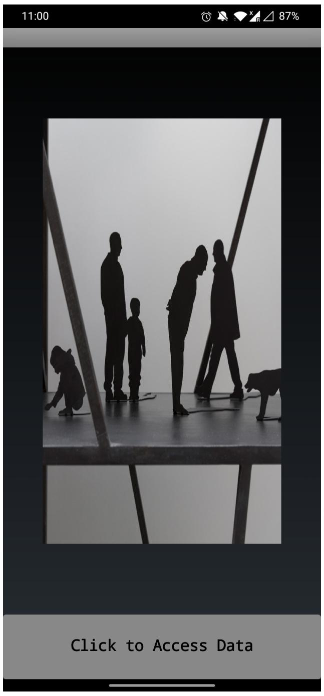


Figure 5.21: Application main page

Chapter 6

System Validation

This section describes various tests that were performed to validate and verify the system and the results that were observed from the various tests are analysed.

6.1 Test1: Validation of vision functionalities

Aim: To check the functioning of vision part of the system.

Components: Arduino Uno, Web camera and Web interface.

Objectives: The objectives that should be tested/achieved using this Web camera are stated below:

- To test the video quality.
- To check the coverage of Web camera.
- To check the live video streaming accuracy in the user-interface.
- To check the possibility for black/white streaming.

Procedure: The procedure is explained in the following steps:

- Connect the ESP32 camera to the Arduino. The connections are: An 5V pin of Arduino connected to Vcc pin of ESP32 camera and ground pin of Arduino to ground pin of ESP32 camera. An D01 pin of ESP 32 is connected to the ground pin of Arduino. A receiver pin (Rx) of ESP32 camera connected to a receiver pin of Arduino and the transmitter pin (Tx) of ESP32 camera connected to the transmitter pin of Arduino.
- When the connections between ESP32 camera and Arduino is made, then the network credentials (such as Wi-Fi ssid (network name you are logged in) and password of the network) for the connection between system and the network are entered.

- Estimating the area of the coverage of the web camera (the range up to which the camera can cover in the streaming).
- Checking the possibility of black and white video streaming in the user-interface.
- Checking the live video streaming accuracy in the user-interface (application, web-interface).

The object is kept in a distance of 1.5 meter from the camera while conducting the test.
Technical specifications:

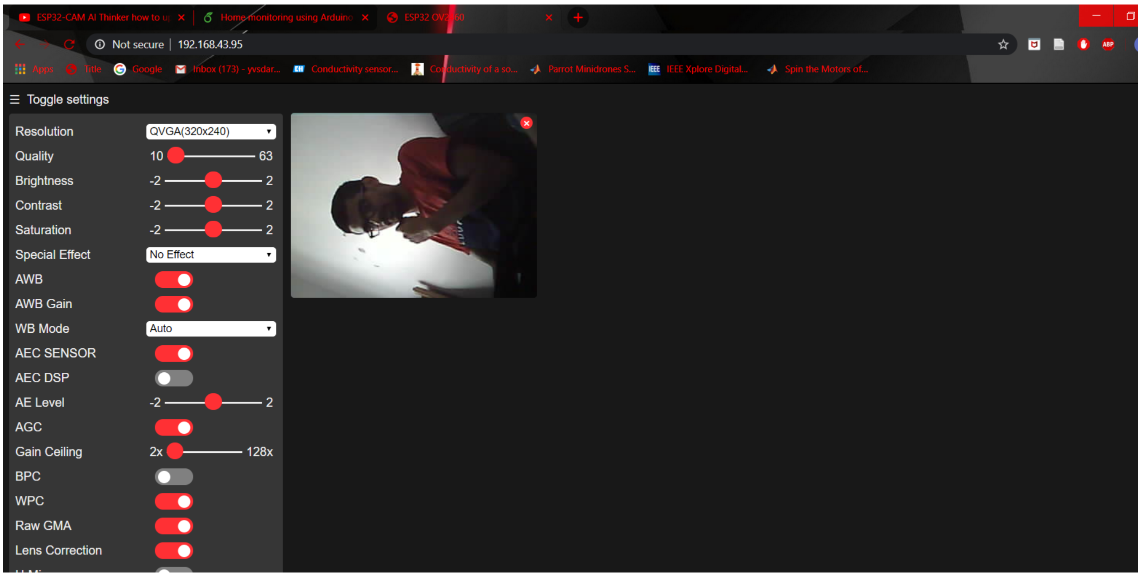
Arduino UNO:

- Operating voltage = 5 V.
- No of digital input/output pins = 14.
- No of analog input pins = 6.
- Flash memory = 32 kB.
- SRAM = 2 kB.
- EEPROM = 1 kB.
- CLK Speed = 16 MHz.

Web Camera:

- Clk speed = 160 MHz.
- Input Voltage = 5 V.

Observations: The image from the web camera is shown in Figure 6.1. From the picture, we can assess the quality of the video that is been streaming in the browser (the quality of the video is 240 HD). The connection of the camera with the Arduino and web interface is proven. Black and white, negative and grey scale video streaming can be adjusted through the setting options available in the user-interface. Black and white streaming is possible in web interface and not possible in application.



Resolution QVGA(320x240)
Quality 10 ● 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 ●

Figure 6.1: Web camera results

Results: The quality of the video streaming is average in the terms of HD it is 240 HD. The black and white video streaming is possible by adjusting in the setting options in web interface. The live video streaming is very accurate in the user-interface.The coverage of the web camera is 3 meters. Hence, we conclude that the vision functionalities are validated.

6.2 Test2: Validation of object detection functionality

Aim: To check whether the PIR sensor is detecting the object correctly.

Components: PIR sensor, Arduino Uno.

Objectives: The PIR motion sensor should be able to detect the motion in 3-meter proximity to the sensor. It should be high (PIR = 1), when there is any motion detection and it should be low (PIR = 0), when there is no object detection. To check the delay time of the sensor. The delay time of the sensor means how long the PIR sensor is high after detecting the motion.

Procedure: The procedure is explained in detail in the following points:

- Connect the PIR sensor to the Arduino. PIR sensor positive and negative to the Arduino Vcc and ground, the sensor pin is connected to the 2nd pin of the Arduino.

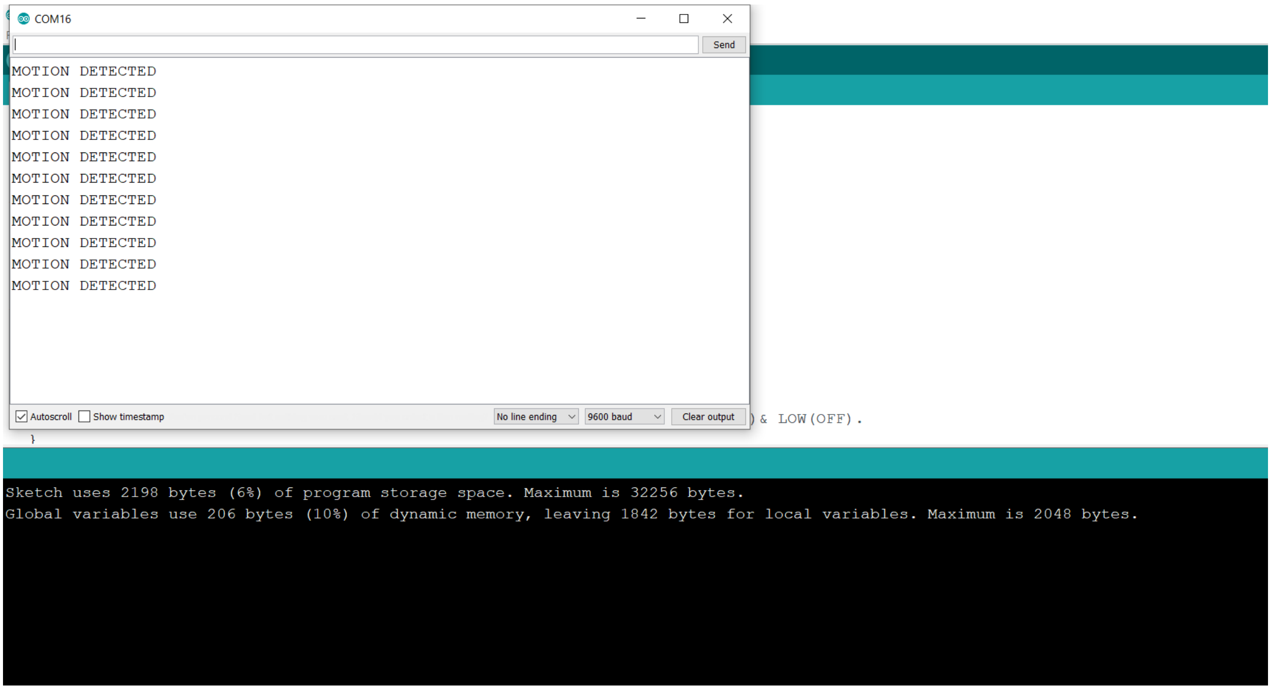
- Give power supply to the Arduino and the PIR sensor will start working. Dump the code inside the Arduino.
- Now the detection of movement is explained in two ways.
  - Case 1: When the object is placed is placed within 3 meters proximity to the sensor.
  - Case 2: When the object is placed above the distance of 3 meters proximity to the sensor.
- The status of the PIR sensor (high, low) can be seen in the output of the Arduino IDE.
- Calculate the delay time means how long the PIR sensor is high after detecting the motion (2 s, 5 s).

Technical specifications:

PIR sensor:

- Cover angle = 120°.
- Cover distance = 7 m.
- Power consumption = 65 mA.

Observations: The image corresponding to the motion detection is shown in Figure 6.2. The pyro-electric sensor is able to detect the motion in the range of 3 meter proximity to the sensor and the validity of detection of movement above the range of 3 meters is less than 60% . In our experiments, we observed the delay time of the sensor is 20 seconds.



```
MOTION DETECTED
MOTION DETECTED
MOTION DETECTED
MOTION DETECTED
MOTION DETECTED
MOTION DETECTED
MOTION DETECTED
MOTION DETECTED
MOTION DETECTED
MOTION DETECTED
MOTION DETECTED
```

```
Sketch uses 2198 bytes (6%) of program storage space. Maximum is 32256 bytes.
Global variables use 206 bytes (10%) of dynamic memory, leaving 1842 bytes for local variables. Maximum is 2048 bytes.
```

Figure 6.2: PIR results

Results: By conducting this test, we understood the sensor proximity (3 meters), delay time of the sensor is 20 seconds. The PIR sensor is working properly. The PIR sensor's cover angle is 120°.

6.3 Test3: Validation of buzzer notification functionalities

Aim: To check the working of buzzer corresponding to the PIR sensor and the frequency of the buzzer when it is high.

Components: Arduino Uno, buzzer, PIR sensor.

Objectives: The buzzer should be activated when there is a motion detected by the PIR sensor.

Procedure **:** This test is conducted to test the buzzer. The procedure is explained in detail in the following steps.

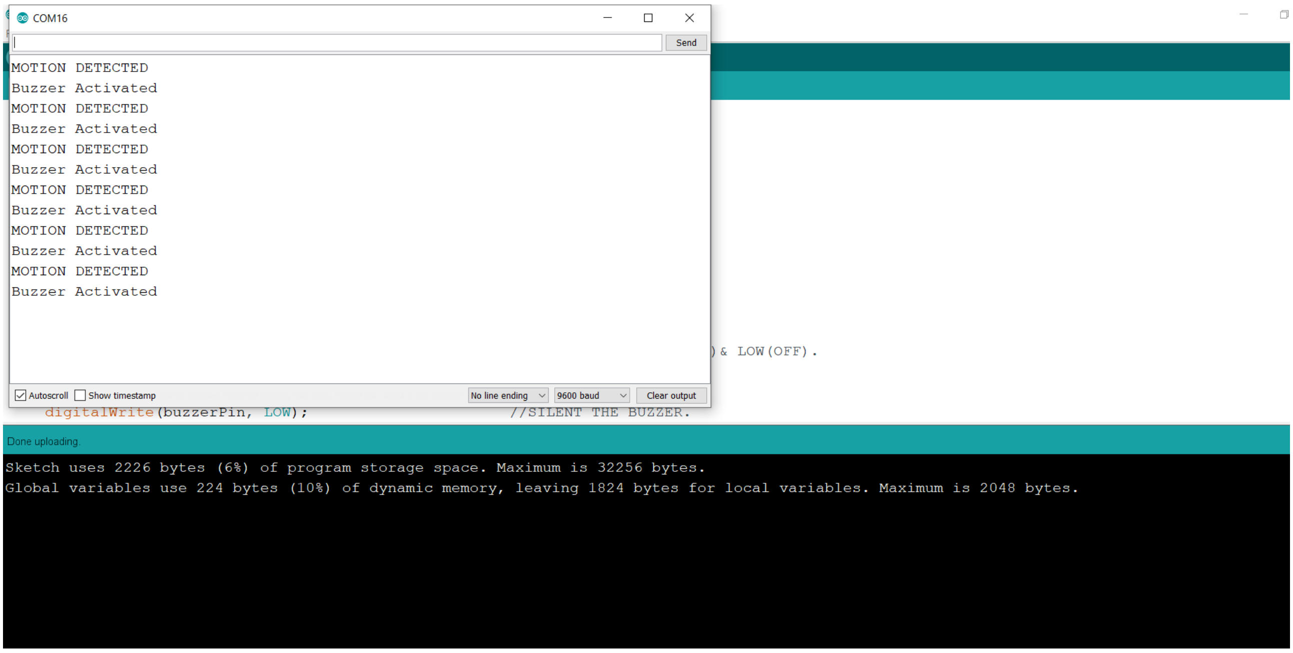
- Connect the PIR sensor positive and negative to the Arduino 5V and ground and the sensor pin is connected to the 2nd pin and buzzer negative is connected to ground, positive pin to 8th pin.
- Make the connections as above mentioned and give power supply to the Arduino. Now, the PIR sensor will start working.
- Case 1: If PIR sensor detects any motion, then note down the state of the buzzer (low or high) and it's frequency.
- Case 2: If there is no motion detection, then note down the state of the buzzer and it's frequency.

Technical specifications:

Buzzer:

- Rated voltage = 6 V DC.
- Operating Voltage = 4-8 V DC.
- Sound Type = Continuous Beep.
- Resonant Frequency = 2300 Hz.

Observations: Figure 6.3 represents the working of buzzer. We could not show the sounds but, through the picture we can clearly understand the buzzer activation with the motion detection. Whenever the PIR sensor senses the motion, the buzzer is high alerting the user that there is an unusual activity.



```
MOTION DETECTED
Buzzer Activated
MOTION DETECTED
Buzzer Activated
MOTION DETECTED
Buzzer Activated
MOTION DETECTED
Buzzer Activated
MOTION DETECTED
Buzzer Activated
MOTION DETECTED
Buzzer Activated
```

```
) & LOW(OFF).
    digitalWrite(buzzerPin, LOW);                    //SILENT THE BUZZER.
```

```
Sketch uses 2226 bytes (6%) of program storage space. Maximum is 32256 bytes.
Global variables use 224 bytes (10%) of dynamic memory, leaving 1824 bytes for local variables. Maximum is 2048 bytes.
```

Figure 6.3: State of the buzzer depending upon the PIR sensor

Results: Hence, by conducting this test, we can observe the working of the buzzer in relative to the PIR sensor. It's on and off conditions. We can state that the buzzer is on when there is a motion detection and off when there is no motion detected. The buzzer will help by scaring the intruders in case of any unusual activities. Hence, the working of buzzer is tested through this experiment. The buzzer can alert the user who is inside the home in case of any alert conditions.

6.4 Test4: Validation of web interface

Aim: To check the interface of the system and also to prove all the system functionalities when all the components are connected.

Components: Buzzer, PIR sensor, Arduino UNO, Web camera.

Objectives: The objectives of the test are:

- User-interface (application, web-interface) must be able to give access to the data for the users.
- The users must get the alert notifications in case of any alert situations.

Procedure: The procedure is explained detail in following steps.

- An 5V pin of Arduino connected to Vcc pin of ESP32 camera and ground pin of Arduino to ground pin of ESP32 camera. An D01 pin of ESP 32 is connected to the ground pin of Arduino. A receiver pin (Rx) of ESP32 camera connected to a receiver pin of Arduino and the transmitter pin (Tx) of ESP32 camera connected to the transmitter pin of Arduino.
- A positive pin of buzzer connected to the digital 8th pin of the Arduino. A ground pin of buzzer connected to the ground pin of the Arduino.
- An *Vcc* pin of PIR sensor connected to the 5V pin of the Arduino. A ground pin of PIR sensor connected to the ground pin of the Arduino. *Dout* pin of PIR sensor connected to the digital 2nd pin of the Arduino.
- Compiling the Arduino code to give the required network credentials for the connection between Arduino and the network.
- When once the system is connected to the network, the system functioning is checked in two ways:
  - *Method 1:* If the the pyro-electric sensor does not detect any object, then the (PIR = 0) and the output of the Arduino IDE will be "no object detected". The user-interface (application, web browser) will notify the user that the conditions are safe in home.
  - *Method 2:* If PIR sensor is high, then the buzzer and the ESP32 camera will be activated through the Arduino. The application and the web browser will notify the user regarding the alert. The user can watch the video streaming through the web interface or the application. The live video streaming will be processed to the user-interface.
- If the user wants to see the live streaming without any alert from the PIR sensor. He can watch it through the start streaming option by logging to the user-interface (application, web interface).

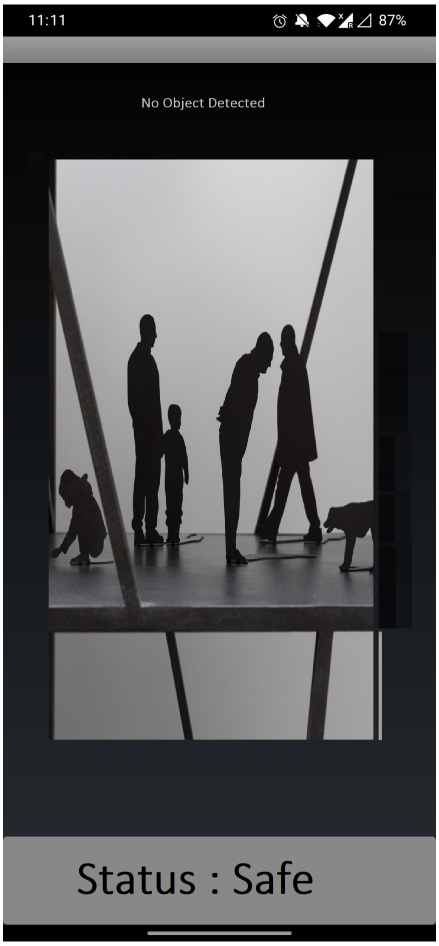
Technical specifications:

ESP32-CAM:

- Bluetooth 4.2 with BLE
- 802.11b/g/n Wi-Fi.
- Supports Wi-Fi Image Upload
- SRAM = 520 kB.

- PSRAM = 4 MB.
- CLK Speed = 160 MHz.

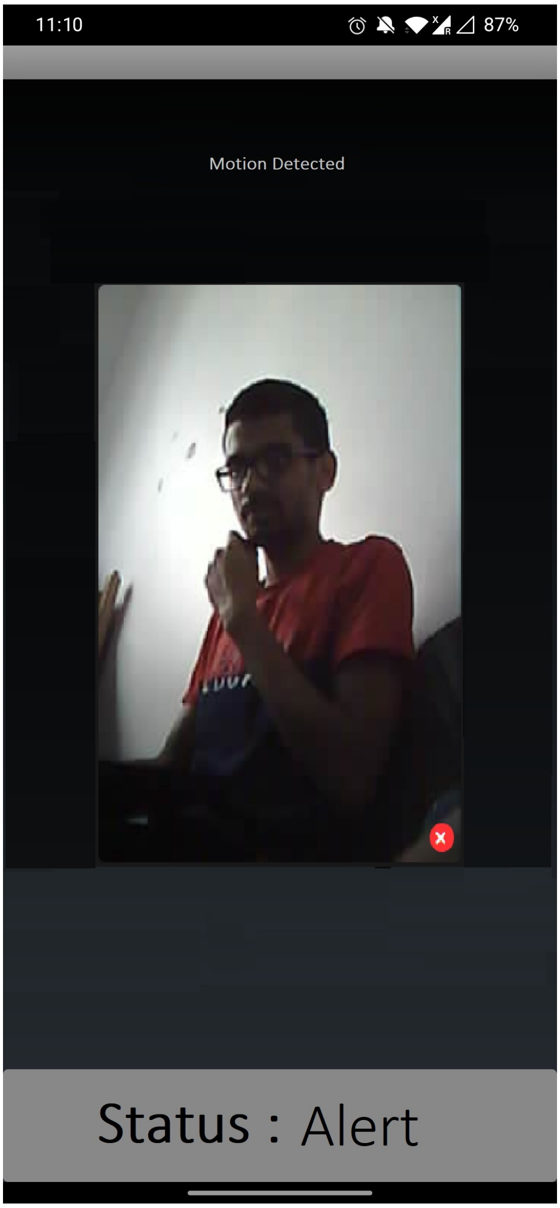
Observations The images corresponding to the alert and safe notification in the user-interface are shown below in Figure 6.4, Figure 6.5. The user will get an alert notification if there is any alert condition. The users can either start or stop streaming through the 'start streaming option' and 'stop streaming option' available in the user-interface the application and web-interface.



No Object Detected

Status : Safe

Figure 6.4: Safe condition in the application



Motion Detected

Status : Alert

Figure 6.5: Alert condition in the application

Results: The system is working flawless when all the components are connected together. The user-interface both the application and web interface are able to provide access to the user. Application program interface is more secure than the web interface. If there is an object detection, then the users will receive the alert notification in the user-interface. If there is no detection, then the system updates the user with safe conditions.

6.5 Test results summary

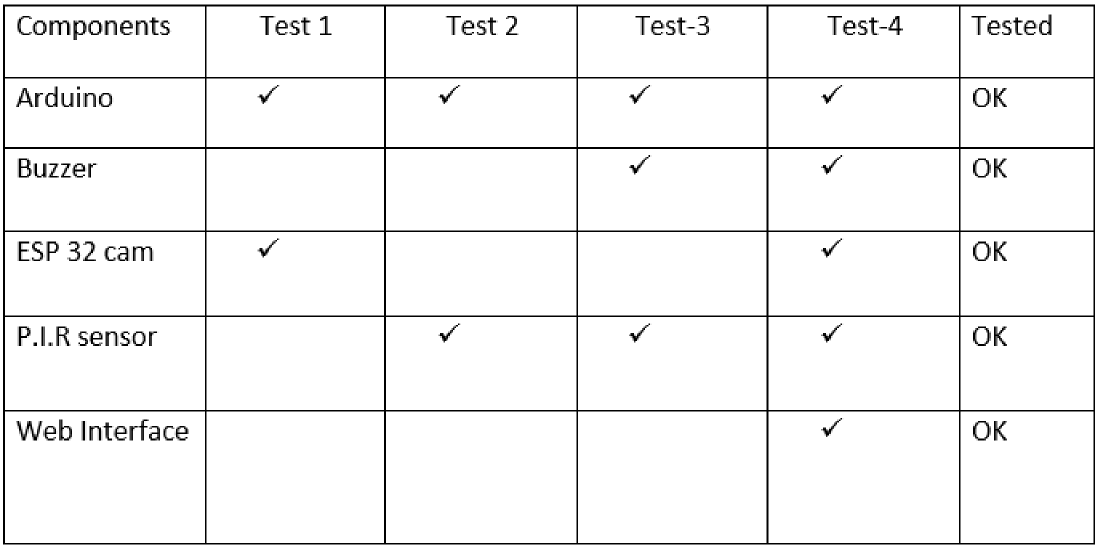
Tests are made upon different components and the results are obtained for each test. For each test, different results are carried and observed. Some discussions can be made using the results.

In the first test result, we can know the functioning of the web camera and the specialties of the web camera. We can know the video quality and the different methods that we can implement using the web camera such as (black and white video streaming).

In the second test result, which was carried on the PIR sensor, we can know its ability and the proximity until which it can detect the motion. It shows the accuracy of the device. The delay time of the sensor is validated.

In the third test. It was carried on the buzzer to check its tone and response with the P.I.R sensor. Through this test, we can know the frequency of the buzzer.

In the fourth test, which is made upon the web browser. We can understand what is happening in the home. We can know whether the application is responding to the system or not and the accuracy of the live streaming.



Components	Test 1	Test 2	Test-3	Test-4	Tested
Arduino	✓	✓	✓	✓	OK
Buzzer			✓	✓	OK
ESP 32 cam	✓			✓	OK
P.I.R sensor		✓	✓	✓	OK
Web Interface				✓	OK

Table 6.1: Table of the test results

These are the discussions that we can make using the results that came from the tests. The components used in different tests and the results of the tests(OK/not OK) are tabulated in the Table 6.1.

Chapter 7

Conclusions and Future Work

Hereby, we conclude that our system can achieve all the objectives that we mentioned earlier in Section 3.2 and it can overcome all the difficulties. The home is a much more safe place because of the system. By adjusting the system manually (start streaming option and stop streaming option), we can be able to see and stop the streaming whenever we wish to see even if there are no dangerous conditions at home. The objectives and the points that we achieved due to the developed system are shown below:

- *Monitoring* the field all the time. This device can monitor the house 24/7 through the sensors used in this project. The user also can be able to monitor through the web interface or application.

- *Web camera* is enabled, if there is any motion. If the users want to watch the streaming, then the web camera will be enabled.

- *User-interface* is developed. The application program interface and web-interface both are developed. The users can get access to the data through the user-interface.

- *Application* is developed through which the user can have access all the time for his/her home. Through this application, users can keep an eye on kids, pets, and stranger activities.

- *Human Intervention* in the field is reduced due to the continuous presence of active sensors in the field.

- *Pets* are monitored all the time from this device. The user can be able to observe the pets whenever it is required through the user-interface. The users can monitor the pets even when the user is away from the house all the time.

- *Children* are monitored 24/7 from this device. The user can be able to observe the children whenever it is required through the user-interface. So, the user can take the necessary action required even if the user is away from the house.

- *Medical Assistance:* The user can see the movements in the house through streaming. If there is any unusual movement in the house the user can take the immediate response require situation. That is he can come home or he can call the helpline number in case of emergencies is any for old people can be provided.

The future scope of our project is discussed below briefly as shown below briefly in the bullet description.

- To make this extended for larger areas as now it is limited to smaller areas and to make the application much more effective.
- We can also make face recognition and detection to know the strangers who entered the house so that we can change the alert conditions by recognition of the faces.
- We can advance the project to the door (lock and unlock activity) so that the chance of burglaries or the stranger activities will be reduced.
- Now, our project has to be set manually for start and stop of the streaming process, in further we can try to skip this manual process.

Bibliography

[1] (May 1, 2020). Do i really need a security system? SafeWise. Library Catalog: www.safewise.com, [Online]. Available: `https://www.safewise.com/home-security-faq/do-i-need-a-security-system/` (visited on 05/04/2020).

[2] Do i need a home security system? Security.org. Library Catalog: www.security.org, [Online]. Available: `https://www.security.org/guide/do-i-need-a-home-security-system/` (visited on 05/04/2020).

[3] M. Al-khafajiy, T. Baker, C. Chalmers, M. Asim, H. Kolivand, M. Fahim, and A. Waraich, "Remote health monitoring of elderly through wearable sensors," *Multimedia Tools and Applications*, vol. 78, no. 17, pp. 24 681–24 706, Sep. 1, 2019, ISSN: 1573-7721. DOI: `10.1007/s11042-018-7134-7`. [Online]. Available: `https://doi.org/10.1007/s11042-018-7134-7` (visited on 05/04/2020).

[4] H. Huo, Y. Xu, H. Yan, S. Mubeen, and H. Zhang, "An elderly health care system using wireless sensor networks at home," in *2009 Third International Conference on Sensor Technologies and Applications*, Jun. 2009, pp. 158–163. DOI: `10.1109/SENSORCOMM.2009.32`.

[5] barzillialloydFollow. (). Arduino home security system, Instructables. Library Catalog: www.instructables.com, [Online]. Available: `https://www.instructables.com/id/Arduino-Home-Security-System/` (visited on 04/27/2020).

[6] P. Nuhalim. (Feb. 1, 2017). DIY arduino home security system - part 1, Device Plus. Library Catalog: www.deviceplus.com Section: Arduino, [Online]. Available: `https://www.deviceplus.com/arduino/diy-arduino-home-security-system-using-rohm-sensor-kit-part-1-mechanics/` (visited on 05/04/2020).

[7] H. U. Zaman, T. E. Tabassum, T. Islam, and N. Mohammad, "Low cost multi-level home security system for developing countries," in *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, Jun. 2017, pp. 549–554. DOI: `10.1109/ICCONS.2017.8250522`.

[8] T. Yokote, Y. Fujii, K. Maru, N. Yoshiura, N. Ohta, and H. Ueda, "Intro-duction of security camera system with privacy protection into a residential area," *Procedia - Social and Behavioral Sciences*, vol. 2, no. 1, pp. 105–110, 2010, ISSN: 18770428. DOI: 10.1016/j.sbspro.2010.01.022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1877042810000236 (visited on 05/11/2020).

[9] (Feb. 19, 2019). Introduction to sensors and transducers, differences, classi-fication, Electronics Hub. Library Catalog: www.electronicshub.org, [Online]. Available: https://www.electronicshub.org/sensors-and-transducers-introduction/ (visited on 05/18/2020).

[10] *Gas detector*, in *Wikipedia*, Page Version ID: 957247030, May 17, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Gas_detector&oldid=957247030 (visited on 05/18/2020).

[11] MQ-6 gas sensor pinout, specs, equivalent, circuit & datasheet. Library Catalog: components101.com, [Online]. Available: https://components101.com/sensors/mq-6-gas-sensor-pinout-equivalent-datasheet (visited on 05/18/2020).

[12] PIR sensor arduino alarm | make: [Online]. Available: https://makezine.com/projects/pir-sensor-arduino-alarm/ (visited on 04/03/2020).

[13] Arduino uno pin diagram, specifications, pin configuration & program-ming. Library Catalog: components101.com, [Online]. Available: https://components101.com/microcontrollers/arduino-uno (visited on 07/02/2020).

[14] *Photodetector*, in *Wikipedia*, Page Version ID: 950937136, Apr. 14, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Photodetector&oldid=950937136 (visited on 05/18/2020).

[15] Introduction to wi-fi technology. Library Catalog: smallbusiness.chron.com, [Online]. Available: https://smallbusiness.chron.com/introduction-wifi-technology-62018.html (visited on 05/18/2020).

[16] The history of bluetooth, [Online]. Available: https://www.androidauthority.com/history-bluetooth-explained-846345/ (visited on 05/25/2020).

[17] Table 2 -1. bluetooth device classes of power management, ResearchGate. Library Catalog: www.researchgate.net, [Online]. Available: https://www.researchgate.net/figure/1-Bluetooth-Device-Classes-of-Power-Management_tbl1_329973291 (visited on 06/29/2020).

[18] P. V. Vimal and K. S. Shivaprakasha, "IOT based greenhouse environment monitoring and controlling system using arduino platform," in *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, ISSN: null, Jul. 2017, pp. 1514–1519. DOI: 10.1109/ICICICT1.2017.8342795.

[19] User-centered design: Definition, examples, and tips | inside design blog. Library Catalog: www.invisionapp.com, [Online]. Available: `https://www.invisionapp.com/inside-design/user-centered-design-definition-examples-and-tips/` (visited on 07/12/2020).

[20] ESP32-CAM video streaming and face recognition with arduino IDE | random nerd tutorials, [Online]. Available: `https://randomnerdtutorials.com/esp32-cam-video-streaming-face-recognition-arduino-ide/` (visited on 05/31/2020).

[21] HC-SR501 PIR sensor working, pinout & datasheet. Library Catalog: components101.com, [Online]. Available: `https://components101.com/hc-sr501-pir-sensor` (visited on 07/02/2020).

[22] Buzzer: Pinout, working, specifications & datasheet. Library Catalog: components101.com, [Online]. Available: `https://components101.com/buzzer-pinout-working-datasheet` (visited on 06/01/2020).

Chapter 8

Appendix

Appendix consists of the prototype coding written in C language for the components(ESP32-CAM, buzzer, PIR sensor and Arduino). It also consists of application coding written in Java language (user-interface). Through this code, we can fulfill the objectives of the system that are mentioned in Chapter 3.

```
#include "esp_camera.h" //library for esp 32 camera
#include <WiFi.h> // library for Wi-Fi
#include "soc/soc.h"//libraries for flawless code running
#include "soc/rtc_cntl_reg.h"

int sensor = 2; // the pin that the sensor is
attached to
int val = 0;    // variable to store the sensor
status (value)
int buzzer = 8; // pin that buzzer connected to


//Make sure that you have either selected ESP32 Wrover Module,


// Select camera model
#define CAMERA_MODEL_AI_THINKER

const char* ssid = "tabu"; // for the connection
const char* password = "tabu@123";// network credentials


#if defined(CAMERA_MODEL_AI_THINKER)
#define PWDN_GPIO_NUM     32 //defining the pins of ESP32
camera module
#define RESET_GPIO_NUM    -1
#define XCLK_GPIO_NUM      0
#define SIOD_GPIO_NUM     26
```

```
#define SIOC_GPIO_NUM     27

#define Y9_GPIO_NUM       35
#define Y8_GPIO_NUM       34
#define Y7_GPIO_NUM       39
#define Y6_GPIO_NUM       36
#define Y5_GPIO_NUM       21
#define Y4_GPIO_NUM       19
#define Y3_GPIO_NUM       18
#define Y2_GPIO_NUM        5
#define VSYNC_GPIO_NUM    25
#define HREF_GPIO_NUM     23
#define PCLK_GPIO_NUM     22

#else
#error "Camera␣model␣not␣selected"//if other model is selected
#endif
// web server is about to run
void startCameraServer();

void setup() {
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);//disables gpio 0
  pinMode(buzzer, OUTPUT);  // initialize buzzer as an output
  pinMode(sensor, INPUT);     // initialize sensor as an input
  Serial.begin(115200);//no of bytes us 115200
  WiFi.begin(ssid, password);// connection is made
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
    }
  Serial.println("");
  Serial.println("Wi-Fi␣connected");
  Serial.setDebugOutput(true);
  Serial.println();

  camera_config_t config;//configuring the pins
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
  config.xclk_freq_hz = 20000000;
  config.pixel_format = PIXFORMAT_JPEG;
  //init with high specs to pre-allocate larger buffers
  if(psramFound()){
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
  } else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
  }

  // camera init
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Camera␣init␣failed␣with␣error␣0x%x", err);
    return;
  }

  //drop down frame size for higher initial frame rate
  sensor_t * s = esp_camera_sensor_get();
  s->set_framesize(s, FRAMESIZE_QVGA);


}


void loop(){
  val = digitalRead(sensor);    // read sensor value
  if (val == HIGH) {            // check if the sensor is HIGH
    digitalWrite(buzzer,HIGH);
    startCameraServer();

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

    Serial.println("Motion␣detected!");
  }else {
    digitalWrite(buzzer, LOW); // turn buzzer OFF
    Serial.println("Motion␣ended!");

    }

    delay(5000);                // delay 100 milliseconds


  Serial.println(val);
}


Application code:

The application code is written in java script.

package com.journaldev.webview;

import android.app.Activity;
import android.os.Bundle;
import android.view.KeyEvent;
import android.webkit.WebSettings;
import android.webkit.WebView;

public class MainActivity extends Activity {

    private WebView webView = null;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        this.webView = (WebView) findViewById(R.id.webview);
        WebSettings webSettings = webView.getSettings();
        webSettings.setJavaScriptEnabled(true);

        WebViewClientImpl webViewClient = new WebViewClientImpl(this);
        webView.setWebViewClient(webViewClient);

        webView.loadUrl("http://192.168.43.95");
    }


    @Override
    public boolean onKeyDown(int keyCode, KeyEvent event) {
        if ((keyCode == KeyEvent.KEYCODE_BACK) && this.webView.canGoBack
            this.webView.goBack();
            return true;
        }

        return super.onKeyDown(keyCode, event);
    }

}
```