VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI - 590018

Project Report
on

"REMOTE CONTROLLED SOLAR CHARGEABLE PATROLLING ROBOT"

Submitted in partial fulfillment of the requirements for the VIII Semester

Bachelor of Engineering
in
ELECTRONICS AND COMMUNICATION ENGINEERING
For the Academic Year
2020-2021
BY

Mayank Saini	1PE17EC075
Rahul Reddy V	1PE17EC098
Rohit Prabhu M	1PE17EC105
Shrishti Upadhyay	1PE17EC133

UNDER THE GUIDANCE OF
Prof. Ananda M
Professor, Dept. of ECE, PESITBSC

Department of Electronics and Communication Engineering
PESIT - BANGALORE SOUTH CAMPUS
Hosur Road, Bengaluru - 560100

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI - 590018

Project Report

on

"REMOTE CONTROLLED SOLAR CHARGEABLE PATROLLING ROBOT"

Submitted in partial fulfillment of the requirements for the VIII Semester

Bachelor of Engineering

in

ELECTRONICS AND COMMUNICATION ENGINEERING

For the Academic Year

2020-2021

BY

Mayank Saini	1PE17EC075
Rahul Reddy V	1PE17EC098
Rohit Prabhu M	1PE17EC105
Shrishti Upadhyay	1PE17EC133

UNDER THE GUIDANCE OF

Prof. Ananda M

Professor, Dept. of ECE, PESIT-BSC

PES INSTITUTIONS

Department of Electronics and Communication Engineering

PESIT - BANGALORE SOUTH CAMPUS

Hosur Road, Bengaluru - 560100

PESIT - BANGALORE SOUTH CAMPUS

HOSUR ROAD, BENGALURU - 560100
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the project work entitled "**Remote Controlled Solar Chargeable Patrolling Robot**" carried out by "**Mayank Saini, Rahul Reddy V, Rohit Prabhu M, Shrishti Upadhyay** bearing **1PE17EC075, 1PE17EC098, 1PE17EC105, 1PE17EC133**" respectively in partial fulfillment for the award of Degree of Bachelors (Bachelors of Engineering) in Electronics and Communication Engineering of Visvesvaraya Technological University, Belagavi during the year 2020-2021. It is certified that all suggestions indicated for internal assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said Degree.

Signature of the Guide	Signature of the HOD	Signature of the Principal
Prof. Ananda M	**Dr. Subhash Kulkarni**	**Dr. Subhash Kulkarni**
Professor	**HOD, ECE**	**Principal, PESIT-BSC**

External Viva

Name of the Examiners **Signature with Date**

1.

2.

Acknowledgements

We would like to express our sincere gratitude to all the lecturers and staff of the Electronics and Communication department for extending their help and guidance towards our project.

We would like to thank the college management and express our sincere gratitude to Dr. Subhash Kulkarni, Principal/Head of Department, Electronics and Communication Engineering for having given us the opportunity for the completion of this project.

We would also like to express our deepest appreciation and gratitude to our project guide Mr. Ananda M, Assistant Professor, for providing us valuable insights throughout our journey. We would like to thank him for all the encouragement and support that was necessary for the progress of this project.

We would also like to thank our friends who volunteered and helped us with enthusiasm in learning various elements that we used in our project.

ABSTRACT

Increasing numbers of crimes around the globe have created a demand for high level of safety and security. At the same time with the fast paced world there has been a requirement to reduce human effort. Many people are increasingly approaching solutions offered by technology to tackle this.

Our project aims at introducing a product in the market that not only ensures security of a bounded area , but also focuses on safety and right to privacy by maintaining a highly secure data storage system. Designed and realized in this project, the robot is equipped with a camera, microphone and ultrasonic sensors that provide remote visuals and audio streams. This is done with the help of a Raspberry Pi and Wi-Fi module. Solar charging of the robot allows conservation of non renewable resources, thus helping us reduce carbon footprint.

Keywords: Robot, Camera, Microphone, Ultrasonic sensors, Raspberry Pi, Wi-Fi, Solar Panels.

Contents

List of Figures

Chapter 1

Introduction

1.1 Objective

To reduce human effort whilst maintaining the integrity of a system and at the same time improving the efficiency in applications requiring surveillance or security by providing a movable camera/microphone on a Remote controlled robot supported by Wi-Fi.

1.2 Problem statement

Crimes like Robbery, breaking and entering, threats to ones safety have made all organizations aware and have forced them to take preventive actions to preclude occurrence of such incidents. It is not just governments or organizations that are choosing to wear a defensive posture, but citizens at an individual level as well, are worried about personal, home and organizational security.

Increasing concerns about business and private property surveillance has given manufacturers and vendors of security equipments a chance to sell these systems by leveraging this fear. Sales figures of security systems are skyrocketing as people tend to believe that these systems can add a layer of security and safeguard their possessions in case of any incident.

The major problem with the development of these devices are to make them compact and light weight. With respect to the project some major hurdles include dealing with huge amounts of data storage that has to be highly secured to ensure the privacy of the organizational content. Another problem that we face is to establish a method to make controlling of the robot very handy,i.e obtaining a wireless control between the robot and the keyboard.

1.3 Problem Solution

Our project aims at introducing a similar product in the market that not only ensures security of a bounded area , but also focuses on safety and right to privacy by maintaining a highly secure data storage system.

The primary objective of our product is to provide remote surveillance to enhance security in an organization. It can be used to monitor inaccessible areas henceforth reducing the risk to human lives.

The robot can be used to increase efficiency of observation and data accumulation in-case of repetitive and tedious tasks. In short, the robot provides remotely accessible and monitored visuals and audio streams to increase the efficiency of security and surveillance while reducing human effort and maintaining safe cloud storage of data.

The solar charging allows conservation of non renewable resources, thus helping us reduce carbon footprint. This product is environment friendly and is a step towards sustainable development.

Chapter 2

Literature Survey

2.1 Overview

The literature survey of a project forms it's backbone, it defines the path and approach to a project. This formed the base of our project that we referred to whenever needed and looked upon it for guidance.

We started our literature survey by watching many videos on YouTube, videos of remote controlled robot, various types of robot structuring. After watching tons of these videos we studied about various microprocessors that would form the core of our project. We tried to build upon an communication algorithm between our robot and other components.

We referred to various IEEE papers for similar projects and took inspiration.After long hours of thorough research and study we decided on major segments of our projects. The two segments that we have completed working on are the web application for live feed and the remote control of the robot using keyboard.

2.2 Web Application Development

In this study via an IEEE paper, a web application was presented to be interface for all the functionality we intended to provide, live video stream monitoring, motor control of the robots, authentication of clients, download option for videos stream etc.

Since our motor control code was already written in Python, we decided to make use of python for development of our web application as well by using FLASK. We came across the fact that tens of thousands of web applications are written in

Flask, a Python-based web framework, and that this is a standard practice for these kind of projects.

There is a rich ecosystem of extensions available in flask. In another IEEE paper, we studied about how to use Flask python module for web application and OpenCV for streaming the video feed from the robot camera.

We already were capable to design and code HTML and CSS files for web application and code python script for motor control and video streaming.Finally we learnt Flask framework by watching a YouTube video tutorial.

2.3 Remote Control of Robot

Obtaining wireless control between the robot and the keyboard was one of the main obstacles of the project. Since we are using the ‘a’, ’w’, ‘s’ and ‘d’ keys to control the movement of the robot we had to make sure that the delay between the moments when the key was pressed and the data being received by the robot was less in order to make it a real time embedded system. Hence, we had to choose an appropriate data transfer protocol to meet these requirements. Upon doing research we have decided to use a local area network (LAN) and have also decided to choose TCP/IP sockets as the mode of communication due to its fast nature of data transfer and also due to its peer-to-peer connection.

The heart of the robot is the raspberry pi 4 (2 GB RAM). After a lot of study we have chosen to use Raspberry Pi over the conventional micro-controllers like Arduino or ESP8266 and the reasons are as follows:

a. The Raspberry Pi being a microprocessor has a real time OS whose kernel is responsible for handling applications and can thus be used to run multiple applications at the same time and hence multithreading need not be introduced in the application code.

b. The Raspberry Pi has inbuilt Wi-Fi and Bluetooth modules and thus doesn’t need separate modules for the same. No additional code is required to connect to the Wi-Fi.

c. The Raspberry Pi has 4 USB ports and can thus support USB devices like microphone and cameras.

d. The Raspberry Pi also has an inbuilt camera slot where we can attach the Pi-Camera for streaming live feed.

e. The Raspberry pi has 40 GPIO headers out of which 4 are available as PWM pins and can thus be used to control the speed of the motors.

f. The Raspberry Pi GPIOs can be accessed and controlled using python code.

The OS run on the raspberry pi is Raspbian OS. The reasons why we have used the OS are as follows:

a. Raspbian is version of Debian used for the Raspberry Pi. Hence it is very secure.

b. VNC server is inbuilt and freely available on Raspbian and thus there is no need of a separate monitor nor is there any need to install it again.

c. Package installations in Raspbian are simple as installations can be done by running Linux commands in the terminal.

Choosing a camera for viewing live feed was also another important task. After surfing the web for various cameras we finally decided to use the raspberry pi camera v2.1. The reason why we used it is because it's a camera specially meant to be used on the Raspberry Pi and thus has good amount of support to start live streams and capture videos and pictures.

Chapter 3

Hardware and Software Requirements Specification

3.1 Hardware Specification

3.1.1 Raspberry Pi 4

Raspberry Pi microprocessor is most preferred when developing advanced applications. Raspberry Pi is an open source platform where one can get a lot of related information so you can customize the system depending on the need.

Features of Raspberry Pi 4

1. Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
2. 2GB, 4GB or 8GB LPDDR4-3200 SDRAM (depending on model)
3. 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
4. Gigabit Ethernet
5. 2 USB 3.0 ports; 2 USB 2.0 ports.
6. Raspberry Pi standard 40 pin GPIO header (fully backwards compatible with previous boards)
7. 2 × micro-HDMI ports (up to 4kp60 supported)
8. 2-lane MIPI DSI display port
9. 2-lane MIPI CSI camera port
10. 4-pole stereo audio and composite video port
11. H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode)

12. OpenGL ES 3.0 graphics
13. Micro-SD card slot for loading operating system and data storage
14. 5V DC via USB-C connector (minimum 3A*)
15. 5V DC via GPIO header (minimum 3A*)
16. Power over Ethernet (PoE) enabled (requires separate PoE HAT)
17. Operating temperature: 0°C – 50°C ambient

* A good quality 2.5A power supply can be used if downstream USB peripherals consume less than 500mA in total.

Figure 3.1: Raspberry Pi 4

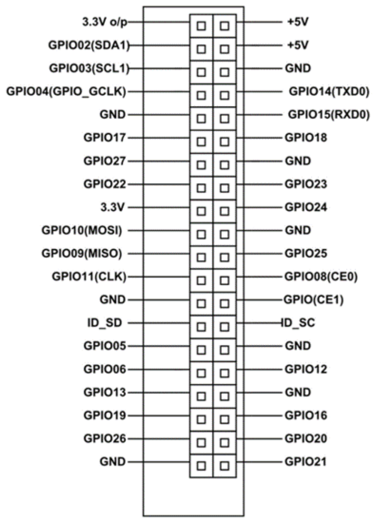

Figure 3.2: Raspberry Pi Pin-out

3.1.2 Raspberry Pi Camera

The Raspberry Pi Camera Modules are official products from the Raspberry Pi Foundation. The original 5-megapixel model was released in 2013, and an 8-megapixel Camera Module v2 was released in 2016. For both iterations, there are visible light and infrared versions. A 12-megapixel High Quality Camera was released in 2020.

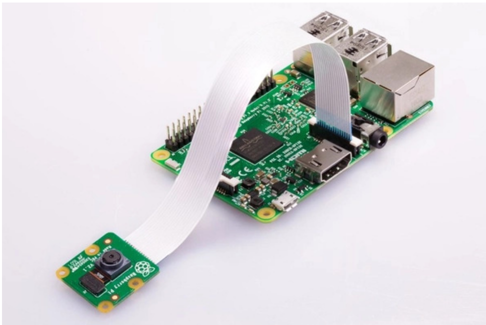

Figure 3.3: Raspberry Pi Camera v2.1

3.1.3 BO DC Motor

BO (Battery Operated) light weight DC geared motor which gives good torque and rpm at lower voltages. This motor can run at approximately 200 rpm when driven by a single Li-Ion cell. Great for battery operated light weight robots. It can do reverse and forward directions.

BO Motor Specifications:

1. Working Voltage 3-12V
2. No Load Speed: 200 rpm /- 10rpm
3. No Load Current: 125mA (max.170mA)
4. Torque: 500gf.cm min 40gm weight

Figure 3.4: BO DC Motor

3.1.4 L298N Motor Driver

L298N Motor Driver Module is a high power motor driver module for driving DC and Stepper Motors. This module consists of an L298 motor driver IC and a 78M05 5V regulator. L298N Module can control up to 4 DC motors, or 2 DC motors with directional and speed control.

L298N Module Features & Specifications:

1. Driver Model: L298N 2A
2. Driver Chip: Double H Bridge L298N
3. Motor Supply Voltage (Maximum): 46V
4. Motor Supply Current (Maximum): 2A
5. Logic Voltage: 5V
6. Driver Voltage: 5-35V
7. Driver Current:2A
8. Logical Current:0-36mA
9. Maximum Power (W): 25W
10. Heatsink for better performance

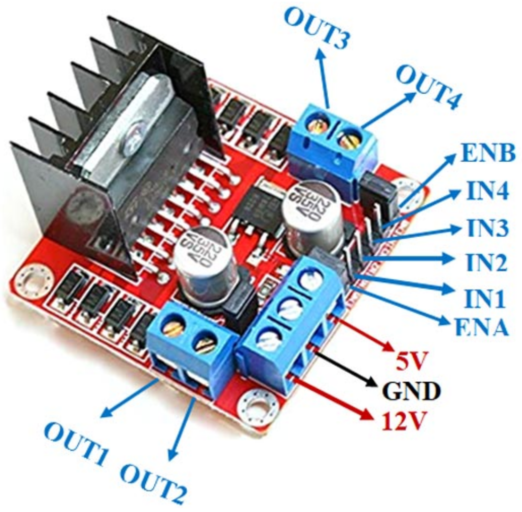

Figure 3.5: L298N Motor Driver

3.1.5 LiPo Battery

A lithium polymer battery, or more correctly lithium-ion polymer battery, is a rechargeable battery of lithium-ion technology using a polymer electrolyte instead of a liquid electrolyte. High conductivity semisolid (gel) polymers form this
electrolyte. These batteries provide higher specific energy than other lithium battery types and are used in applications where weight is a critical feature, such as mobile devices, radio-controlled aircraft and some electric vehicles

Figure 3.6: LiPo Battery

3.1.6 Router(Gateway)

The router is a physical or virtual inter-networking device that is designed to receive, analyze, and forward data packets between computer networks. A router examines a destination IP address of a given data packet, and it uses the headers and forwarding tables to decide the best way to transfer the packets. Some important points of routers are given below:

A router is used in LAN (Local Area Network) and WAN (Wide Area Network) environments. For example, it is used in offices for connectivity, and you can also establish the connection between distant networks.

It shares information with other routers in networking.

It uses the routing protocol to transfer the data across a network.

Figure 3.7: Router

3.2 Software Requirements

3.2.1 Raspbian

Raspberry Pi OS (formerly Raspbian) is a Debian-based operating system for Raspberry Pi. Since 2015 it has been officially provided by the Raspberry Pi Foundation as the primary operating system for the Raspberry Pi family of compact single-board computers.

Previous Pi OS versions have been 32bit and based on Raspbian core, taking the name Raspbian. Since recent 64bit versions no longer use the Raspbian core, the name has been changed to Raspberry Pi OS for both 64bit and 32bit versions.

Raspberry Pi OS is highly optimized for the Raspberry Pi line of compact single-board computers with ARM CPUs. Raspberry Pi OS uses a modified LXDE as its desktop environment with the Openbox stacking window manager.

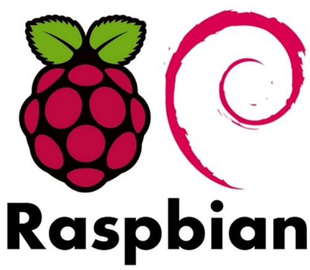

Figure 3.8: Rasbian

3.2.2 Python

Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a ”batteries included” language due to its comprehensive standard library.

Figure 3.9: Python

3.2.3 Flask

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

Figure 3.10: Flask

Chapter 4

System Design

4.1 Overview

This chapter describes the overall design of the wireless communicated robot movement and the web application working for the live feed from camera mounted on the robot. As we already know the hardware and software components used we will further go ahead and discuss the system design that we developed.

4.2 Designing the Web Application

Web technologies we are using for the project are HTML, CSS, Bootstrap, Python and Flask. These technologies are needed to configured in a proper manner for the web application to work properly. In the directory where the project folder for the web application is established all the constituent files and folders have to be kept in a specific format as supported by Flask.

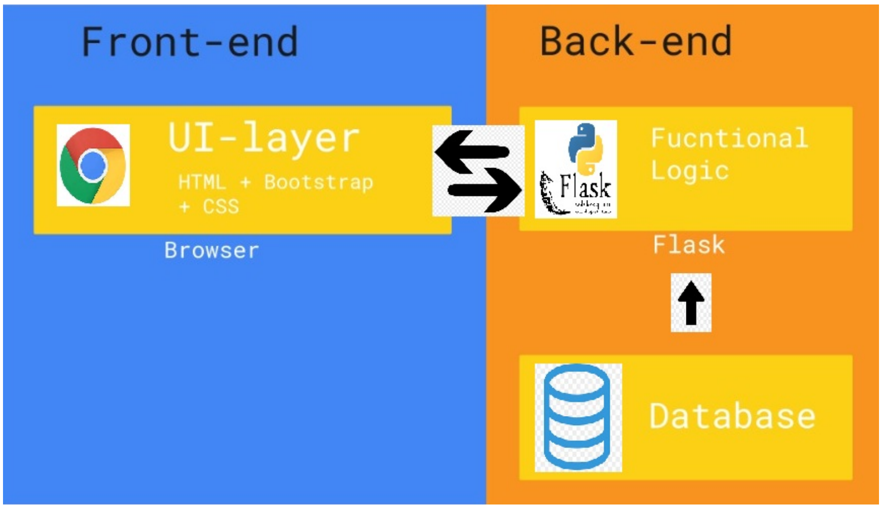

Figure 4.1: Block Diagram of Web Application

Here is the basic file structure for flask:

```
yourapp/
    .py
    static/
        .js
        .css
        Images
    templates/
        .html
```

Description of files:

.py: contains the actual python code that will import the app and start the development server.
static/ : contains static files i.e. CSS (including bootstrap), JavaScript, images.
templates/ : This is where you store your HTML templates i.e. index.html, layout.html
Python modules used in app.py (main web app Script/Program):

1. Flask - Web application framework
2. CV2 - Library to help the drawing process with Open-CV
3. Sockets - provides access to the BSD socket interface
4. Keyboard - provides various keys-related functions
5. Time - provides various time-related functions
6. Rpi.GPIO - This package provides a class to control the GPIO on a Raspberry Pi.

4.3 Designing the remote controlled robot movement

The heart of the robot is a raspberry pi 4(2GB RAM) which is a credit card sized computer used in a wide range of applications varying from a simple personal computer to advanced robotics. The raspberry pi has an inbuilt Wi-Fi module which can connect to any network. The OS of the raspberry pi resides within an SD card of size 32 GB. The motors of the robot are controlled using the PWM pins available on the 40 pin GPIOS of the raspberry pi. The raspberry pi GPIO pins output a current insufficient to drive the motors and hence we have used the L298N motor driver which amplifies the current so that it drives the motors easily. In order to get information about the surroundings of the area in which the robot is patrolling we have used three sensors. They are ultrasonic sensors, Picamera v2.1 and a USB mic. The ultrasonic sensor is used to measure distances on all sides of the robot in order to ensure that the robot doesn't collide with any obstacles in its vicinity. The Picamera acts as the eyes of the robot. The robot is steered using the 'a', 'w', 's' and 'd' keys of the robot based on the camera feed received. The USB mic acts like the ears of the robot. Using the mic, the robot can capture audio within its vicinity.

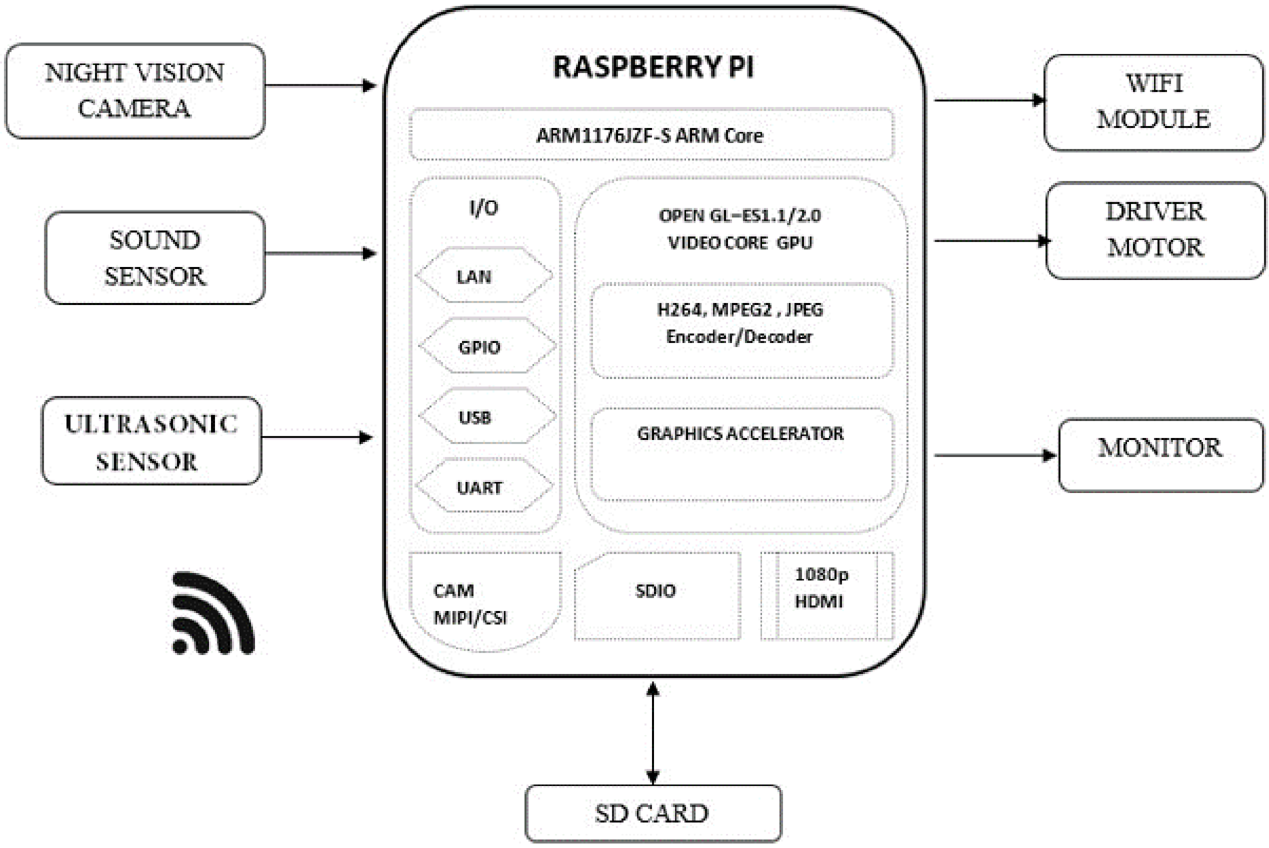

Figure 4.2: Block Diagram of Robot

4.3.1 TCP/IP Communication

The communication protocol which we have opted to use in our project is the TCP/IP socket protocol. The TCP/IP socket protocol establishes a peer-to-peer connection between the server and the client followed by which data is transferred in the form of packets of fixed length. In our case each packet is 1024 bytes in length. A socket is one end of a 2-way communication. A socket consists of an IPV4 address and a port number. The connection between the server and the client happens with a series of events occurring on both the server and the client sides.

The following occurs on the server side:

1. Declaration of the socket object: first a socket object is created which is an instance of the socket class.
2. Socket binding: The process of assigning a port number to a socket is called socket binding.
3. Listening for connections: Once the socket has been binded it soon starts listening for connection requests from active clients.
4. Accepting Client connections: The server accepts all the client connections who are willing to connect to the server. Upon connection the server receives connection objects using which the server sends data to the client.
5. Data transfer: The server uses the connection objects to send data to the client.

The following occurs on the client side:

1. Declaration of the socket object: first a socket object is created which is an instance of the socket class.
2. Connection: The client sends a connection request to the server using the ip and port number of the server.
3. Data transfer: Once the server accepts the connection of the client, the client starts receiving the data sent by the server.

4.3.2 GIX algorithm

The GIX algorithm is an acronym for "Gone in exchange" Algorithm . The GIX algorithm ensures that data is transferred securely over the internet by using encryption and decryption algorithms on the sender and the receiver sides respectively.

The following occurs on the sender side:

1. First the pure control message is received. Which in our case is the key pressed.
2. A random key is chosen and the control message is encrypted using the key.

3. The random key is hidden within the encrypted control message and the final encrypted message is returned.
4. The encrypted message is sent over the internet.

The following occurs on the receiver side:

1. The encrypted control message is received.
2. The hidden key is found within the encrypted message.
3. Using the key, the message is decrypted to get the pure control message.
4. Appropriate tasks are performed based on the control message received.

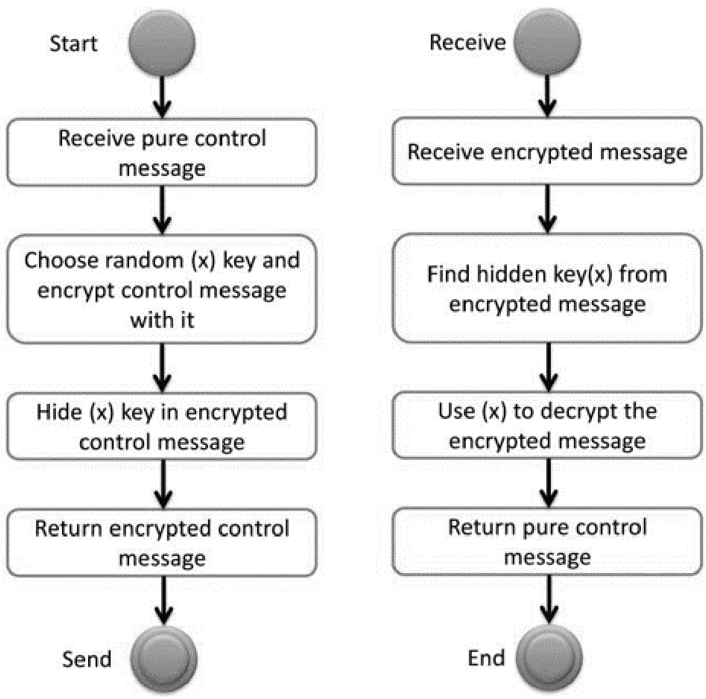

Figure 4.3: GIX Algorithm

Chapter 5

Implementation

5.1 Implementation of Web Application

5.1.1 Software Implementation

To host the live streaming server we are making use of the python script in the back-end, communicating with the rendered front-end on the client side.

For the script we are using open-cv modules for support streaming and control of the stream parameters.

5.1.2 Hardware Implementation

Raspberry Pi camera is used directly with the camera port on the Raspberry Pi.

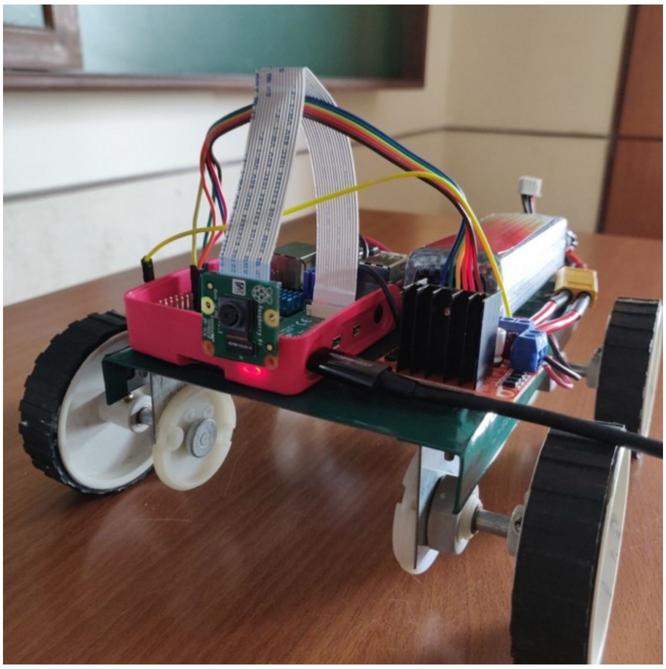

Figure 5.1: Live feed capture using Raspberry Pi Camera

5.2 Implementation of Robot Movement

5.2.1 Software Implementation

First the keystrokes are recorded using keyboard module in the form of a character, the recorded characters are then sent from server to client using TCP/IP sockets.

Upon receiving the character on the receiver side the appropriate action is implemented using an if-else ladder which is reflected by the motion of the robot.

5.2.2 Hardware Implementation

Raspberry Pi GPIO used in Board mode and is connected to the motor driver in the format given below:

Pin 13 – IN1(Left)

Pin 15 – IN2(Left)

Pin 16 – IN3(Right)

Pin 18 – IN4(Right)

Pin 32 – ENA(PWM - Left)

Pin 33 – ENB(PWM -Right)

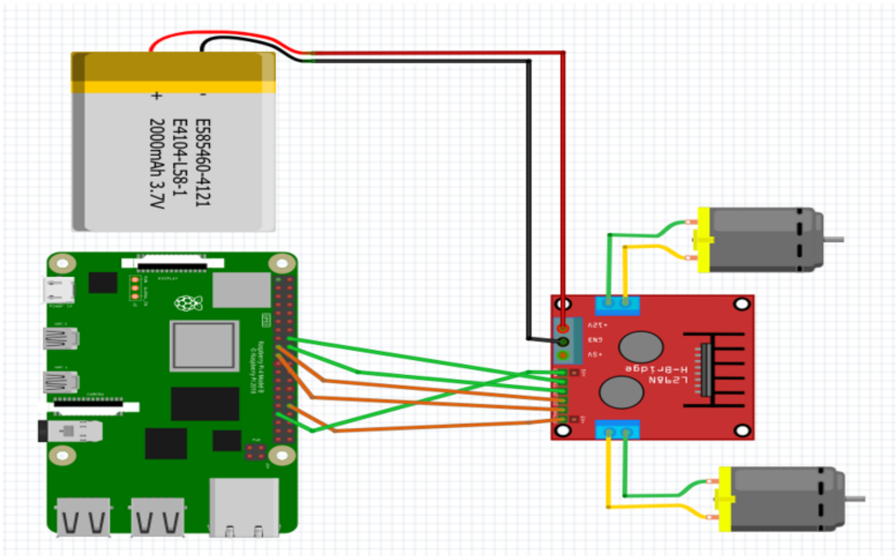

Figure 5.2: Circuit Connections

Chapter 6

Result Analysis

6.1 Robot Movement

The robot movement in various directions was performed by pressing the a, w, s, d keys on the keyboard. The robot reacted spontaneously. This was accomplished with the help of wifi and delay less transfer of data. Upon pressing of the keys the respective robot movement in different directions was observed. In addition to this we demonstrated the working of the motors by the means of LED bulb s and a table to showed the multidirectional movement.

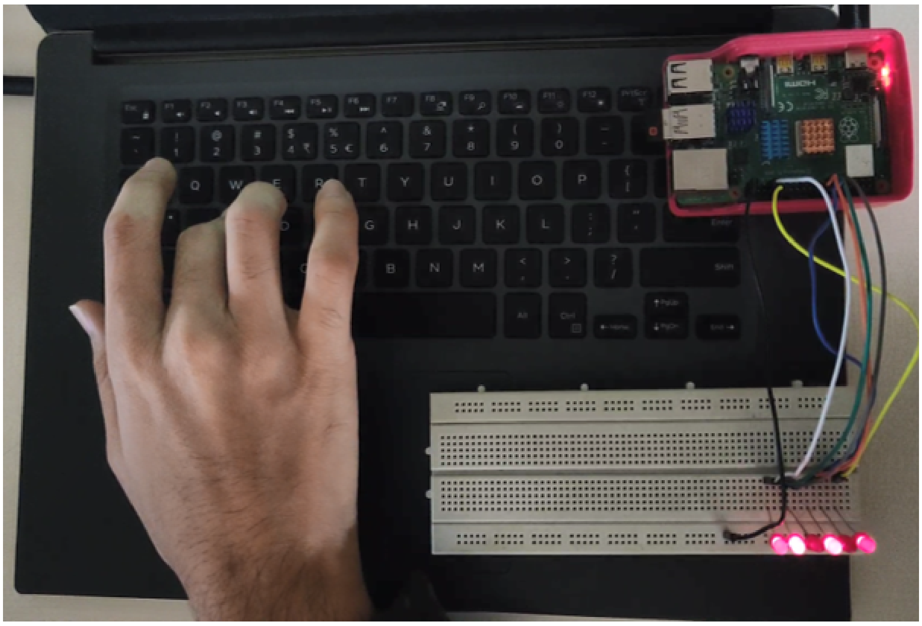

Figure 6.1: Working of motor using LEDs

6.2 Live Feed Capture

The camera connected to the raspberry pi gave instant camera feed without any delay. This was reflected on the websites that we developed .It was observed that the camera quality was good and hence clear images were obtained.

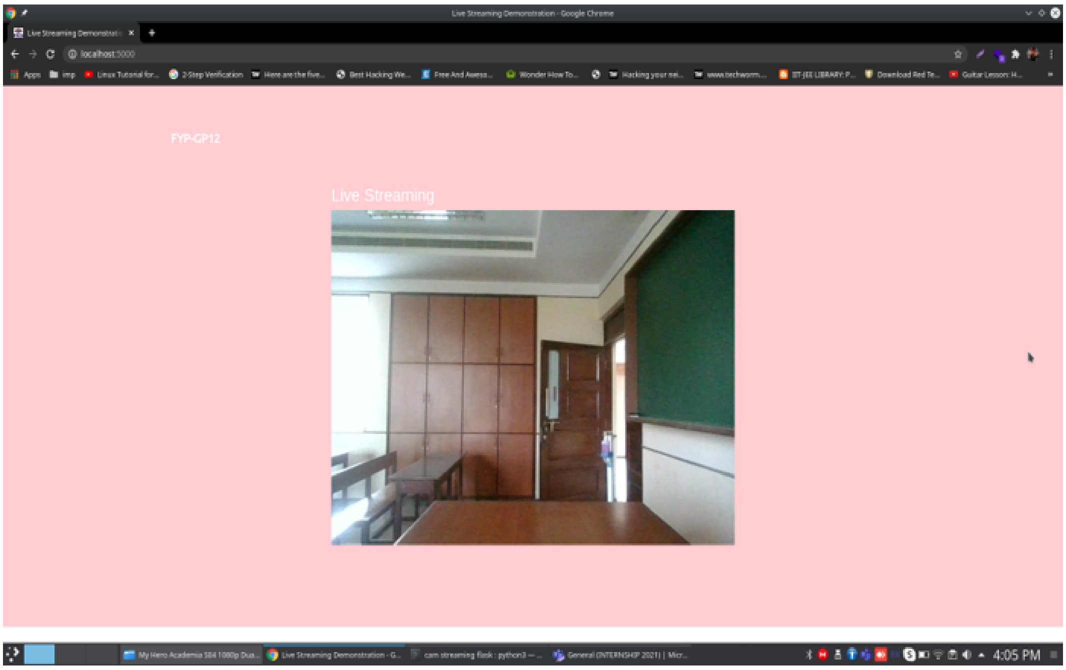

Figure 6.2: Local Live feed on Webpage

Chapter 7

Conclusion and Future Scope

7.1 Conclusion

After successfully testing the various segments of our project so far we can conclude that this product can be a very ideal device for surveillance in bounded areas. the robot responds to all the keys in ideal time frame and moves accordingly in different directions. The accuracy of these movements make it a very idealistic equipment.

The live feed capturing from the camera resent on the robot is also very accurate and has minimal or no delay at all. The clarity of the same is astonishing.

After testing and analysing the project so far these are the advantages that we can conclude upon:

1. Low cost
2. Wireless system makes it compact
3. Easily portable
4. System operates at near real time
5. Makes remote controlling very easy
6. Huge add-ons like obstacle avoidance is possible
7. Eliminates human error
8. Nature friendly as it uses solar charging
9. Ensures data privacy of the user

7.2 Future Scope

The project that we have implemented has immense future prospective and an abysmal scope for future enhancements. There are limitless possibilities of its usage in the industry. However our main aim was to implement a low cost compact handy remote controlled robot due to which the choices we had for implementation were limited and narrowed down only to a few.

Future applications of the product are as follows:

1. **Security management systems**: The robot can be used for patrolling in campuses of various organizations spread over an area. The camera and microphone present on it can provide access to live feed of any corner of the campus at any given point of time. A server setup in the security room can store data in a safe and secure manner.
2. **Wildlife photography and videography**: The robot can be used by a wildlife photography team in order to observe the flora and fauna for endless hours . As the robot can manage obstacles, it is easier for it to move around in rough terrains . This helps the organization to study the wildlife without disturbing it with human presence. Example mine caves.
3. **Survey of underground tunnels**: the product can be applicable for uses in data logging and observational surveys that need to be conducted in inaccessible places. For example a survey of underground metro lines or sewage tunnels.
4. **Information gathering**: The device can be excellent for active reconnaissance. Military observation of a region to locate an enemy or ascertain strategic features. the safe and secure data storage assures national security.

References

[1] *Design and Implementation of Security Patrol Robot using Android Application*
https://ieeexplore.ieee.org/abstract/document/8424310

[2] *A Model for Remote Controlled Mobile Robot over Wi-Fi Network Using Arduino Technology*
https://ieeexplore.ieee.org/document/7141255

[3] *Software System for Remote Robot Control and Monitoring based on Android Operating System and Wireless Communication*
https://ieeexplore.ieee.org/document/8249435

[4] *DESIGN AND IMPLEMENTATION OF SURVEILLANCE ROBOT FOR OUTDOOR SECURITY*
https://ieeexplore.ieee.org/document/8256885

[5] *Robocop; A Robust Surveillance Robot*
https://ieeexplore.ieee.org/document/8256885

[6] *Video Surveillance Robot Control using Smartphone and Raspberry Pi*
https://ieeexplore.ieee.org/abstract/document/7754547

[7] *Surveillance Robot using Raspberry Pi and IoT*
https://ieeexplore.ieee.org/document/8437086

[8] *Surveillance Robot with Human Detection*
https://ieeexplore.ieee.org/document/8473137

[9] *Intro to IP addresses*
https://youtu.be/1pWxI7iVWoQ

[10] *Intro to ports*
https://youtu.be/qsZ8Qcm6_8k

[11] *Intro to sockets*
https://youtu.be/_FVvlJDQTxk

[12] *Reverse +direct connections*
https://youtu.be/IVMVRPkLjqs

[13] *Creating socket*
https://youtu.be/TMbHg43as_E

[14] *Socket binding and connections*
https://youtu.be/WitopEiN3IQ

[15] *Accepting connections*
https://youtu.be/BE7ocj5wKBk

[16] *Sending commands*
https://youtu.be/JT0WaoHR0TI

[17] *Python socket client server*
https://youtu.be/lT6s0T8lPyU

[18] *Python socket client side*
https://youtu.be/bbkuljM3SW0

[19] *Locally testing*
https://youtu.be/G1oUCZZGgHQ

[20] *Flask for Python*
https://www.youtube.com/watch?v=Z1RJmh_OqeA

[21] *Learnt to use RPI.GPIO library*
https://learn.sparkfun.com/tutorials/raspberry-gpio/python-rpigpio-api

Appendix A

Review Process

A.1 Phases of Review

Project was evaluated at phases with major checkpoints as follows :

Pre-Review : ***Defining Problem Statement and Feasibility analysis Review***

1. Understanding of the problem statement.
2. Technical understanding of domain.
3. Identification of differentiating features.
4. Feasibility of conversion to product or paper.

1st Review : ***Literature survey and Project plan Review***

1. Clarity in understanding of the problem/project.
2. Completion of Literature Survey.
3. Identification of sub-blocks and their interaction.
4. Timeline for completion of project using Gantt chart.

2nd Review : ***Module level design and Test Plan Review***

1. Detail design of each module.
2. Integration and module test plan.
3. Availability status of required Hardware and Software components.
4. 20% Completion of Block/Sub module implementation.

3rd Review : *Project Progress Review*

1. Adherence to project plan.
2. Completion of module interaction interface design.
3. 50% Module level implementation.
4. Demonstration of completed modules using primitive interfaces

4th Review : *Project Progress and Finishing plan Review*

1. 70% Module level implementation
2. Presentation of completed test data as per test plan.
3. Packaging plan.
4. Final demonstration plan.

5th Review : *Module completion and Integration Review*

1. 100% Module implementation.
2. Completion of Integration.
3. Presentation of test data as per test plan.
4. Adherences to project plan.

6th Review : *Demonstration Review*

1. Quality of packaged full demo presentation.
2. Integration test data as per test plan.
3. Quality of documentation made across project.
4. Group presentation / communication skill.

Appendix B

Project Planning

Gantt Chart