

**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**



**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

**Prof. Ananda M
Professor**

Signature of the HOD

**Dr. Subhash Kulkarni
HOD, ECE**

Signature of the Principal

**Dr. Subhash Kulkarni
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 and Ultrasonic sensors. This is done with the help of a Raspberry Pi and Wi-Fi module.

Keywords: TCP-IP Sockets, Raspberry Pi, Surveillance, Open-CV, Python 3, Website, Web development, Flask, Server, Robotics, Robot, Camera, Ultrasonic sensor.

Contents

1	Introduction	1
1.1	Objective	1
1.2	Problem statement	1
1.3	Problem Solution	2
2	Literature Survey	3
2.1	Overview	3
2.2	Web Application Development	3
2.3	Remote Control of Robot	4
3	Hardware and Software Requirements Specification	6
3.1	Hardware Specification	6
3.1.1	Raspberry Pi 4	6
3.1.2	Raspberry Pi Camera	8
3.1.3	BO DC Motor	8
3.1.4	L298N Motor Driver	9
3.1.5	Ultrasonic Sensor (HC-SR04)	10
3.1.6	LiPo Battery	10
3.1.7	Router(Gateway)	11
3.1.8	Solar Panel	11
3.2	Software Requirements	12
3.2.1	Raspbian	12
3.2.2	Python	12
3.2.3	Flask	12
4	System Design	13
4.1	Overview	13
4.2	Designing the Web Application	14
4.3	Designing the remote controlled robot movement	15
4.4	TCP/IP Communication	16

4.5	Sensor Data API	18
4.6	Ultrasonic Sensor Circuit	18
4.7	Solar Charging	19
5	Implementation	20
5.1	Implementation of Website	20
5.2	Implementation of Robot Movement	21
6	Result Analysis	22
6.1	Robot Movement	22
6.2	Video Streaming	22
6.3	Sensor Data Acquisition	23
6.4	Solar Charging	23
7	Conclusion and Future Scope	24
7.1	Conclusion	24
7.2	Future Scope	25
References		25
A	Review Process	28
A.1	Phases of Review	28
B	Project Planning	30
C	Work Division	31
C.1	Work Done in 7th Semester	31
C.2	Work Done in 8th Semester	32
D	Links to Project files and Videos	33

List of Figures

3.1	Raspberry Pi 4	7
3.2	Raspberry Pi Pin-out	7
3.3	Raspberry Pi Camera v2.1	8
3.4	BO DC Motor	8
3.5	L298N Motor Driver	9
3.6	Ultrasonic Sensor (HC-SR04)	10
3.7	LiPo Battery	10
3.8	Router	11
3.9	Solar Panel	11
4.1	System Design Flowchart	13
4.2	Block Diagram of Web Application	14
4.3	Block Diagram of Robot	16
4.4	Working of API(Local)	18
4.5	Ultrasonic Circuit	19
4.6	Block Diagram of Solar Charging Circuit	19
5.1	Ultrasonic Distance Graph	20
5.2	Motor Circuit Connections	21
6.1	Complete Robot Setup	22
6.2	Video Streaming Webpage (Screenshot)	23
6.3	Obstacle detected in Left and Rear Sensors	23
C.1	Prototype - 1	31
C.2	Prototype - 2	32

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 measures to avoid occurrence of any such incidents. It is not just government bodies or organizations that are choosing to take preventive actions, but citizens at a personal level as well, are worried about individual, home and organizational security. Increasing concerns about business and private property surveillance has given manufacturers and vendors of security equipment a chance to sell these systems by leveraging this fear. Market value of security systems are increasing exponentially as a result of people believing that these systems can add a layer of security and safeguard their assets in case of any incidents. 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 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 its 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 its 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

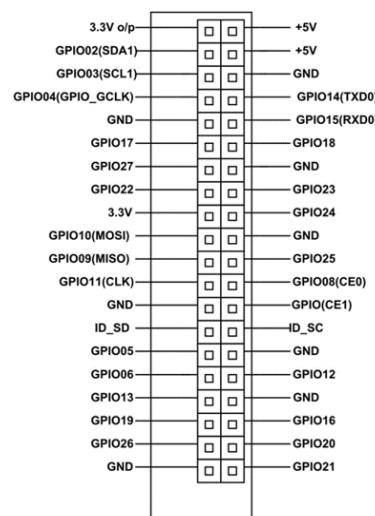


Figure 3.2: Raspberry Pi Pin-out

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

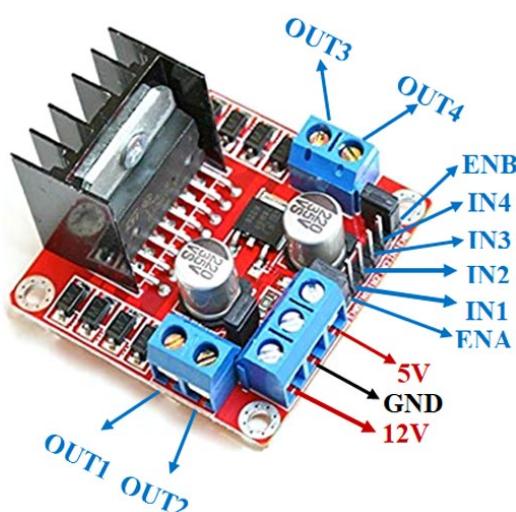


Figure 3.5: L298N Motor Driver

3.1.5 Ultrasonic Sensor (HC-SR04)

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal.



Figure 3.6: Ultrasonic Sensor (HC-SR04)

3.1.6 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 3.7: LiPo Battery

3.1.7 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 3.8: Router

3.1.8 Solar Panel

A solar panel is an assembly of photo-voltaic cells mounted in a framework for installation. It uses sunlight as a source of energy to generate direct current electricity.

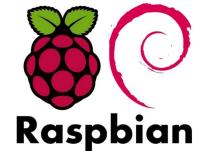


Figure 3.9: Solar Panel

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.

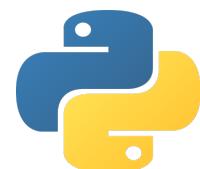


Previous Pi OS versions have been 32bit and based on Raspbian core, taking the name Raspbian. Since recent 64bit versions no longer use the Raspbian core, the name has been changed to Raspberry Pi OS for both 64bit and 32bit versions.

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.

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.

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.



Chapter 4

System Design

4.1 Overview

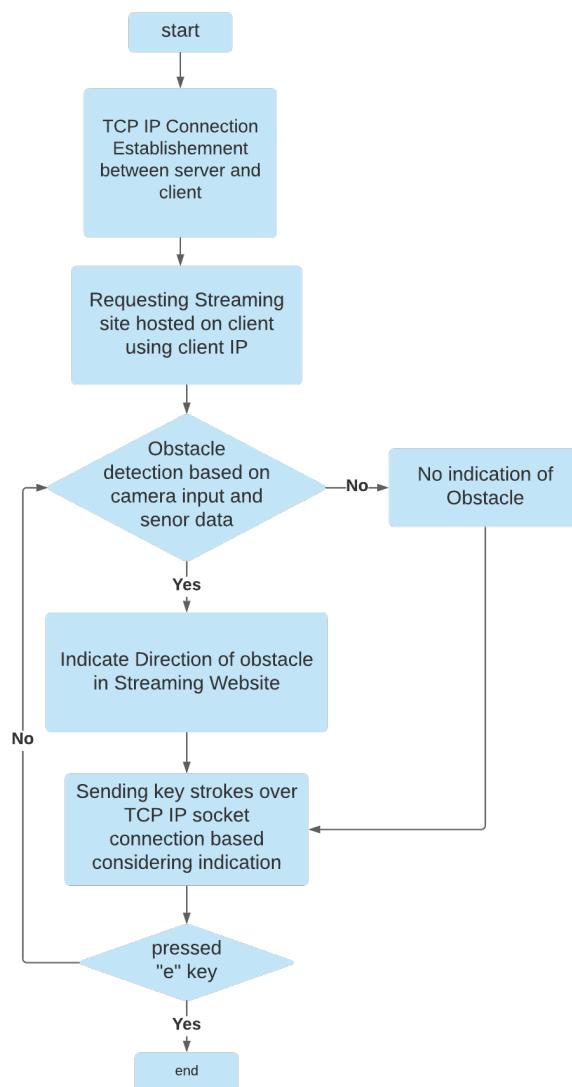


Figure 4.1: System Design Flowchart

The above flowchart 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 be 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.

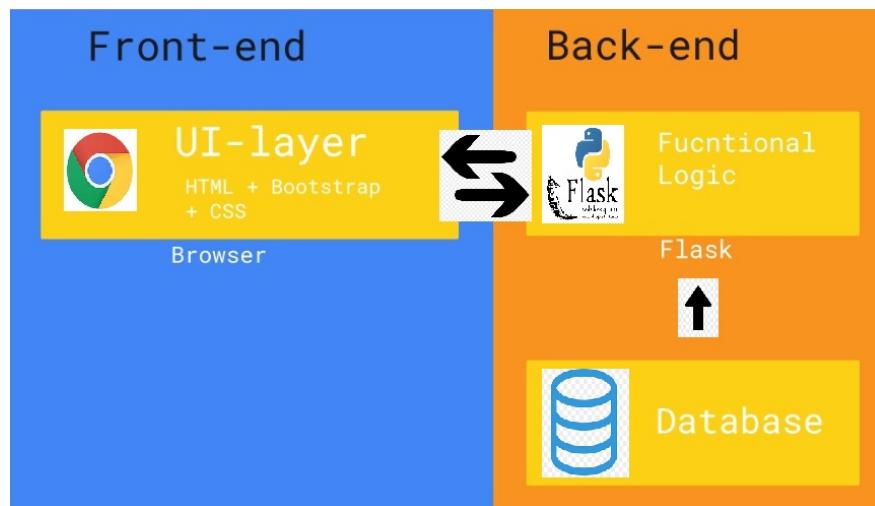


Figure 4.2: Block Diagram of Web Application

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.

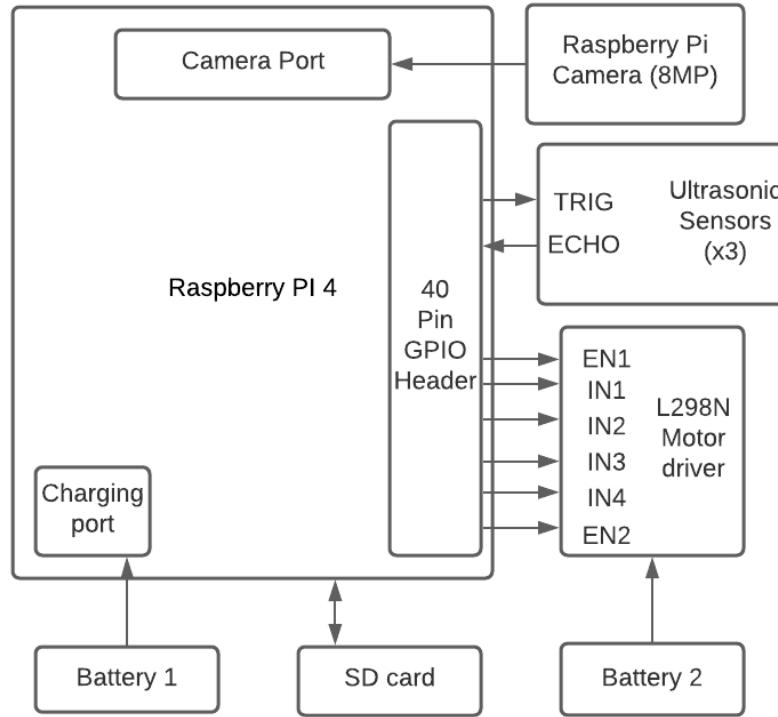


Figure 4.3: Block Diagram of Robot

4.4 TCP/IP Communication

A TCP-IP Socket is a combination of a computer's IP address and port number. The IP address is used to identify a computer in the network while the port number distinguishes between the various applications running on the computer.

Port numbers range from 0-65535 and are divided into the following ranges:

- 0-1023: Allocated to server services. Example SMTP uses 25 as the port number
- 1024-49151: This can be registered for services but cannot run user defined programs
- 49151-65535: These can be used freely for client programs

TCP-IP sockets provides an easy way to link two computers for data transfer. It is easy and quickly configurable. It provides bidirectional data transfer and ensures that the sequence of data being transferred is maintained thereby ensuring real timeliness. The two devices which we have used in our project are a Raspberry Pi and any PC/Laptop of any kind. The PC/Laptop is the server while the Raspberry Pi is the client and is the heart of the robot. The key strokes are recorded using a python

library called keyboard and this information is sent to the Raspberry Pi over sockets. The following events occur during the data transfer in the server and the client:

1. Server side:

- a. Socket object declaration: In order to access the socket functionality firstly a Socket object of the class Socket from the Python3 Socket library is declared.
- b. Socket binding: Socket binding is the process of assigning an IP address a port number.
- c. Listening for client connection requests: The server now begins to listen to active clients for connection requests. The number of connections to be accepted can be specified
- d. Accepting client connections: The server begins to accept client connections and for every client connection accepted the server receives a connection object and a tuple. The tuple consists of the clients IP address and Port number.
- e. Data Transfer: The server now enters an infinite while loop where it waits for keys of the PC/Laptop to be pressed. The information to be sent is firstly encoded from string to byte string and is then sent over socket to the client. Once a key is pressed the keystroke is recorded as a String, converted to byte string and is then sent. If no key is pressed then a default value is sent in the same way as mentioned above to prevent timeout.

2. Client Side:

- a. Socket object declaration: In order to access the socket functionality firstly a Socket object of the class Socket from the Python3 Socket library is declared.
- b. Connection to the server: The client uses the connect function of the socket to send a connection request to the server using the servers IP address and Port number
- c. Data Transfer: Once the server accepts the client's connection request the client begins to receive data sent by the server in the form of chunks. The chunk size can be specified. The received chunk of data is decoded back into string and based on the final string the appropriate action is performed by the robot using an if-else-if ladder.

4.5 Sensor Data API

In this part of the project we made an API which interacts with an ultrasonic distance sensor using the raspberry pis GPIO header. We use three sensors,one for the left, one for the right and one for the back of the car. The api acquires the sensor data and it is returned in the JSON format. Using JavaScript and jQuery we request the data from the API and change the content on the screen using DOM manipulation.

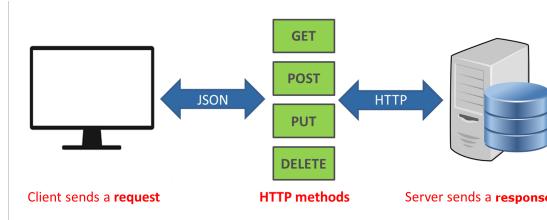


Figure 4.4: Working of API(Local)

RESTful API

REST stands for representational state transfer and the APIs which follow this architecture are called RESTful APIs. A REST API is a software architecture style which uses a subset of the HTTP protocol. An API maybe viewed as a bridge between two entities, an information provider and an information user. The 5 methods used in a RESTful API are GET, POST, PUT, DELETE, PATCH. In our project we make use only of the GET method.

GET: Used to retrieve data from a server. Upon retrieving the data successfully the error code given is 200(OK). If the resource is not found the error code is 404 (NOT FOUND) and if the request sent is not properly formatted then the error code is 400(BAD REQUEST).A response maybe in the form of a HTML code, JSON or XML.

4.6 Ultrasonic Sensor Circuit

The output of the Ultrasonic sensor (HCSR-04) is 5V while the maximum input voltage acceptable by the Raspberry Pi is 3.3V. If the output of the ECHO pin is not stepped down to 3.3V then the Raspberry Pi can get damaged. Thus we need to step down the voltage using a voltage divider resistor network. Circuit is show above and the calculation for it as follows:

$$V_{out} = V_{in} * \frac{R_2}{R_1 + R_2}$$

$$3.3 = 5 * \frac{R_2}{680 + R_2}$$

$$R_2 = 1320\Omega$$

$$R_2 \approx 1560\Omega$$

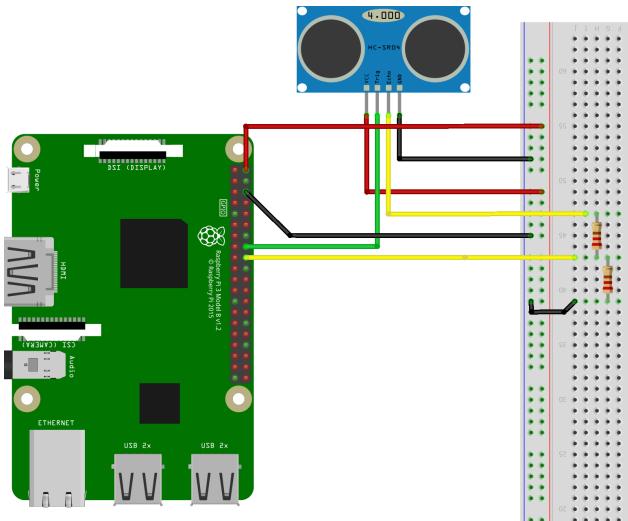


Figure 4.5: Ultrasonic Circuit

4.7 Solar Charging



Figure 4.6: Block Diagram of Solar Charging Circuit

The above Fig. 6 represents the block diagram of various components used in the Solar Charging circuit.

- a. **Solar panel:** It is the input source for the circuit. It provides voltage between 0-26V.
- b. **DC Voltage Control Circuit:** It helps regulate the fluctuating output voltage of the solar panel varying from 0-26V to Constant Voltage(CV) of 12V(For 12V Li-Po Battery).
- c. **Battery Management System:** This circuit prevents Short Circuit, Over Charging and also helps in faster charging.

Chapter 5

Implementation

5.1 Implementation of Website

To host the live streaming server we make use of a python script in the back-end, communicating with the rendered front-end on the client side. In the script we make use of Open-CV Python Library for streaming and control of the stream parameters. The website contains two main elements which are, Video streaming and Alerts from ultrasonic sensors on the Left, Right and Rear Ends of the Robot.

Video Streaming

Video frames are captured by Raspberry Pi Camera which are then converted to JPEG images. These JPEG images are displayed continuously on the website at a high speed. This results in the images appearing to be a Real Time Video.

Ultrasonic Sensor Alerting

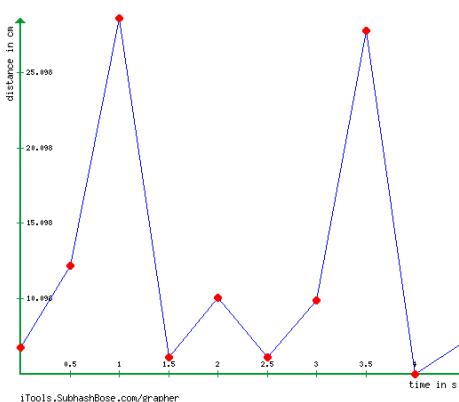


Figure 5.1: Ultrasonic Distance Graph

In Ultrasonic Sensor Alerting, we host an API i.e., host the sensor data acquisition code on a flask server. We make use of JavaScript to request sensor data from

the API. Once the Ultrasonic Sensor readings are received, the data is then extracted and is in the JSON format. The data here being the distance. We then make use of DOM manipulation to trigger alerts on the website application if the distance is less than 10cm. The above figure represents the distance recorded in Real Time.

5.2 Implementation of Robot Movement

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

- Pin 18 IN1(Right)
- Pin 16 IN2(Right)
- Pin 15 IN3(Left)
- Pin 13 IN4(Left)
- Pin 12 ENA(PWM - Right)
- Pin 33 ENB(PWM - Left)

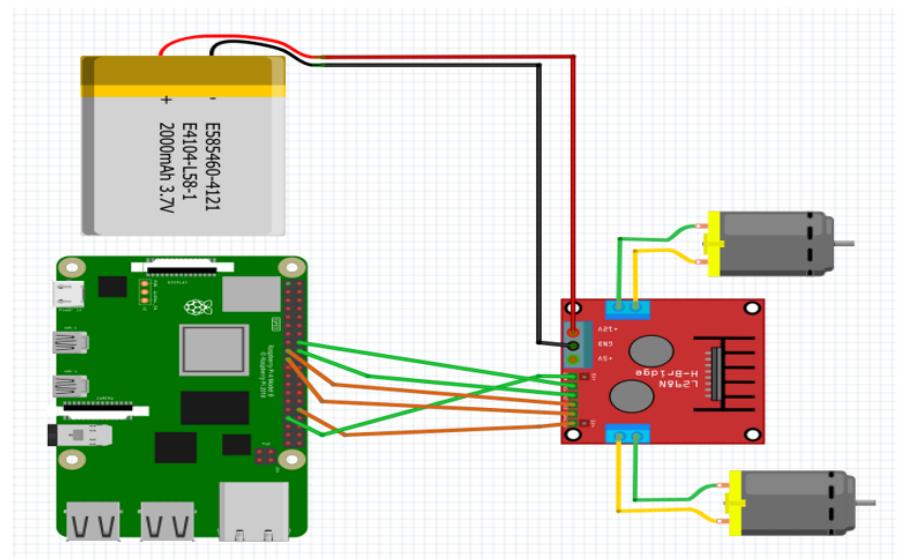


Figure 5.2: Motor Circuit Connections

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 Wi-Fi and a delay less TCP-IP socket transfer protocol. All the functionality of the TCP-IP socket protocol were successfully implemented without any errors. Upon pressing of the keys the respective robot movements in different directions were observed.

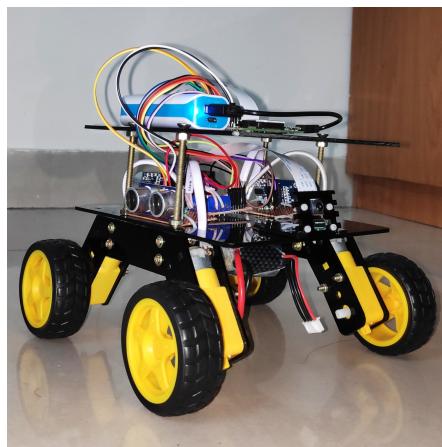


Figure 6.1: Complete Robot Setup

6.2 Video Streaming

We also successfully captured video frames using a Raspberry Pi Camera and displayed it on a flask website.

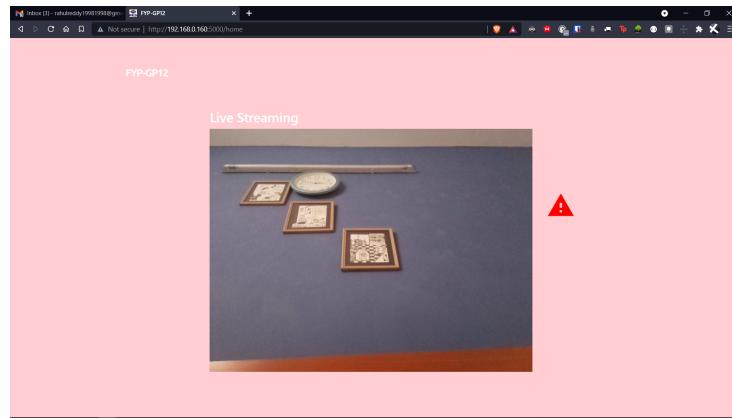


Figure 6.2: Video Streaming Webpage (Screenshot)

6.3 Sensor Data Acquisition

We also hosted an API which helps in the acquisition of Ultrasonic sensor data. We then request sensor data from this API to provide alerts for the left, right and rear ends of the robot on the web-page. The API worked according to the intended functionality. The Video Stream and the API together helped us in remotely controlling the robot.

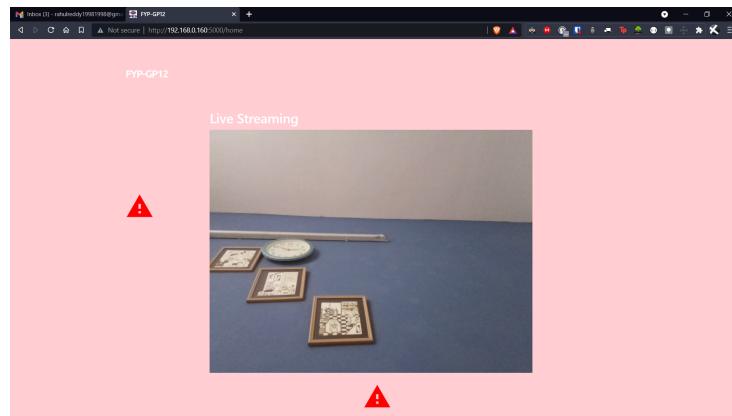


Figure 6.3: Obstacle detected in Left and Rear Sensors

6.4 Solar Charging

In addition we also built a solar charging circuit in-order to charge the batteries. This made our project more environment friendly and helped conserve Non-Renewable Energy resources.

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] J. Bae, D. Lee and K. Cho, "Speed and Direction Control of Two In-wheel BLDC Motors for the Self-Driving Surveillance Robot," 2020 6th International Conference on Mechatronics and Robotics Engineering (ICMRE), 2020, pp. 17-21, doi: 10.1109/ICMRE49073.2020.9065011.
- [2] M. Hamza, M. Atique-ur-Rehman, H. Shafqat and S. B. Khalid, "CGI SCRIPT AND MJPG VIDEO STREAMER BASED SURVEILLANCE ROBOT USING RASPBERRY PI," 2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST), 2019, pp. 947-951, doi: 10.1109/IBCAST.2019.8667217.
- [3] A. U. Bokade and V. R. Ratnaparkhe, "Video surveillance robot control using smartphone and Raspberry Pi," 2016 International Conference on Communication and Signal Processing (ICCSP), 2016, pp. 2094-2097, doi: 10.1109/ICCSP.2016.7754547.
- [4] H. R. and M. H. Safwat Hussain, "Surveillance Robot Using Raspberry Pi and IoT," 2018 International Conference on Design Innovations for 3Cs Compute Communicate Control (ICDI3C), 2018, pp. 46-51, doi: 10.1109/ICDI3C.2018.00018.
- [5] M. F. Alvi, F. Junejo, M. W. Baloch and A. J. Rajput, "RoboCop; A Robust Surveillance Robot," 2018 1st International Conference on Advanced Research in Engineering Sciences (ARES), 2018, pp. 1-8, doi: 10.1109/ARESX.2018.8723270.
- [6] S. Meghana, T. V. Nikhil, R. Murali, S. Sanjana, R. Vidhya and K. J. Mohammed, "Design and implementation of surveillance robot for outdoor security," 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2017, pp. 1679-1682, doi: 10.1109/RTEICT.2017.8256885.

- [7] N. K. Raghavendra and K. Padmavathi, "Solar Charge Controller for Lithium-Ion Battery," 2018 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES), 2018, pp. 1-5, doi: 10.1109/PEDES.2018.8707743.
- [8] Raspberry Pi L298N Interface Tutorial — Control a DC Motor
<https://www.electronicshub.org/raspberry-pi-l298n-interface-tutorial-control-dc-motor-l298n-raspberry-pi/>
- [9] Flask For Python3
https://youtu.be/ZIRJmh_OqeA
- [10] Solar Battery Charger (LiPo/Li-Ion)
<https://youtu.be/kEttqWJrdww>
- [11] Web Basics - Accessing JSON data from URL
<https://youtu.be/kIRp6HOQzP8>
- [12] Run a python script forever on Raspberry Pi
<https://gist.github.com/skynet-05/f5445b8ad046a08b9469884c12a1a443>
- [13] Raj Kamal, Internet of Things-Architecture and design principles, McGraw Hill Education.
- [14] Holger Karl & Andreas Willig, "Protocols And Architectures for Wireless Sensor Networks" , John Wiley, 2005.
- [15] Feng Zhao & Leonidas J. Guibas, Wireless Sensor Networks- An Information Processing Approach", Elsevier, 2007.

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 Analysis Review (27/10/2020)

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

1st Review : Literature survey and Project plan Review (9/11/2020)

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 (7/12/2020)

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 Project implementation.

3rd Review : Project Progress Review (27/5/2021)

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

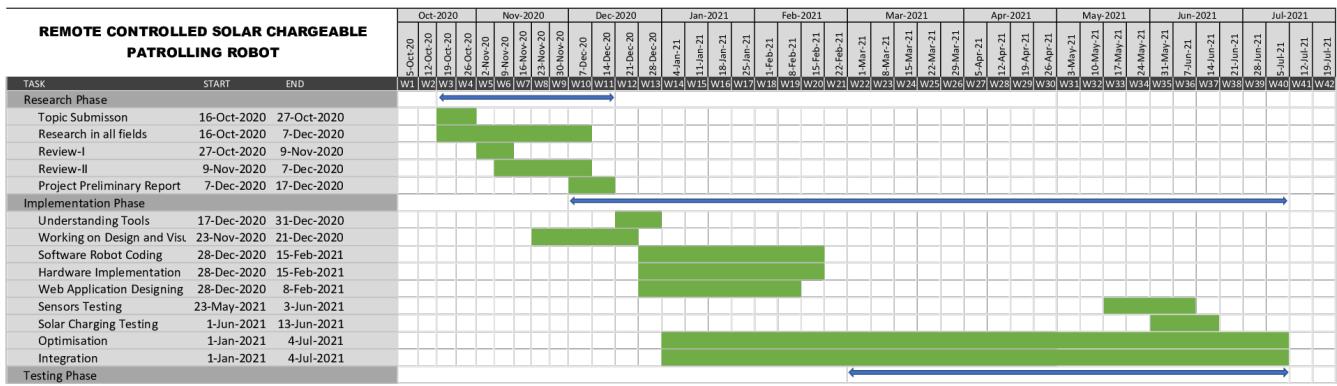
4th Review : Final Demonstration Review (17/7/2021)

1. 100% Module implementation and Integration.
2. Quality of packaged full demo presentation.
3. Quality of documentation made across project.
4. Group presentation / communication skill.

Appendix B

Project Planning

Gantt Chart



Appendix C

Work Division

C.1 Work Done in 7th Semester

1. Building Prototype of the Robot with remote control movement.

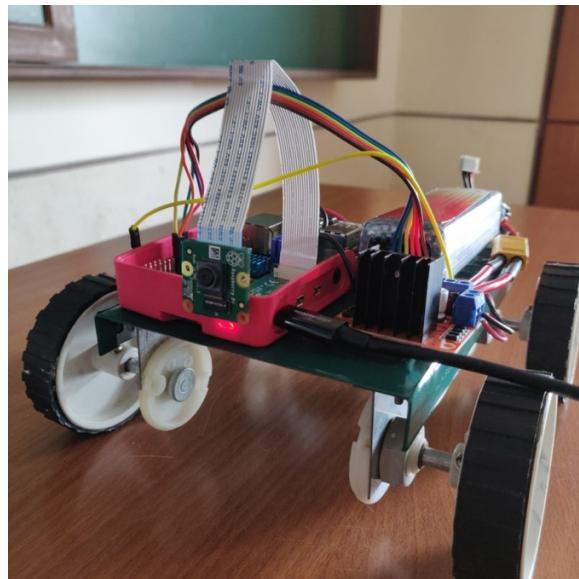


Figure C.1: Prototype - 1

2. Developed Web Application for video streaming and remote control of the Robot.

C.2 Work Done in 8th Semester

1. Re-built the Prototype with better Frame and Parts.

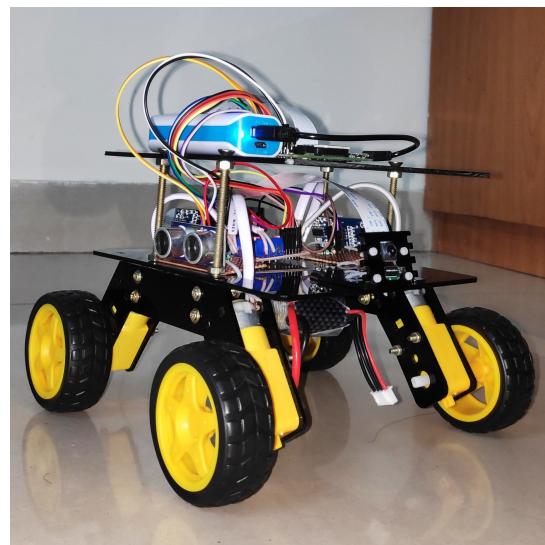


Figure C.2: Prototype - 2

2. Redesigned the Web Application by adding Ultrasonic Sensor Alerts.
3. Added Solar Charging to the Robot.

Appendix D

Links to Project files and Videos

1. GitHub Link: <https://bit.ly/pesfyp12>
2. Phase 1 Video: <https://youtu.be/-SnVCEUNwMg>
3. Phase 2 Video: