VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belgaum – 590 018

A project report on

"TWO-DIMENSIONAL MAPPING USING AN AUTONOMOUS ROBOT"

Submitted in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

INFORMATION SCIENCE & ENGINEERING

by

Deepak P (1CR16IS028)
Dhanish V (1CR16IS030)
Deepak T (1CR16IS029)
Kumareshan Subramaniyan (1CR16IS048)

Under the guidance of

Dr. Anand R
Associate Professor
Dept. of ISE, CMRIT, Bengaluru

CMR INSTITUTE OF TECHNOLOGY
DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING
#132, AECS Layout, IT Park Road, Bengaluru-560037

2019-20

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belgaum – 590 018

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

Certificate

This is to certify that the project entitled as "**Two Dimensional Mapping Using an Autonomous Robot**", is a bonafide work carried out by **Kumareshan Subramaniyan(1CR16IS048), Deepak P (1CR16IS028), Dhanish V (1CR16IS030), and Deepak T (1CR16IS029)** in partial fulfillment of the award of the degree of Bachelor of Engineering in Information Science & Engineering of Visvesvaraya Technological University, Belgaum, during the year 2019-20. It is certified that all corrections/suggestions indicated during reviews have been incorporated in the report. The project report satisfies the academic requirements in respect of the Phase I project work prescribed for the said Degree.

…………………………………..	……………………………………
Name & Signature of Guide	**Name & Signature of HOD**
Dr. Anand R	**Dr. Farida Begam**
Assoc. Professor	**HoD**
Dept. of ISE, CMRIT	**Dept. of ISE, CMRIT**

External Viva

Name of the examiner	**Signature with date**
1.	
2.	

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany a successful completion of any task would be incomplete without the mention of people who made it possible, success is the epitome of hard work and perseverance, but steadfast of all is encouraging guidance.

So with gratitude I acknowledge all those whose guidance and encouragement served as beacon of light and crowned our effort with success.

I would like to thank **Dr. Sanjay Jain,** Principal, CMRIT, Bangalore for providing excellent academic environment in the college and his never-ending support for the B.E program.

I would also like to thank **Dr. Farida Begam,** Assoc. Professor and HOD, Department of Information Science and Engineering, CMRIT, Bangalore who shared his opinions and experiences through which I received the required information crucial for the project.

I consider it a privilege and honour to express my sincere gratitude to my internal guide **Dr. Anand R**, Assoc. Professor, Information Science and Engineering, CMRIT, Bangalore for their valuable guidance throughout the tenure of this project work.

I would also like to thank all the faculty members who have always been very Co-operative and generous. Conclusively, I also thank all the non-teaching staff and all others who have done immense help directly or indirectly during my project

DEEPAK P(1CR16IS028)

DEEPAK T(1CR16IS029)

DHANISH V(1CR16IS030)

KUMARESHAN SUBRAMANIYAN(1CR16IS048)

ABSTRACT

An efficient and an innovative way to build a two-dimensional map of an indoor environment using an ultrasonic sensor has been described in this project. A mobile robot deployed with an ultrasonic sensor has been used and map generation algorithm has been implemented that is hosted on Heroku platform that can be remotely accessed by any device. This project presents a simple map building algorithm to build a map using static obstacles present in the indoor environment. The main motive is to build a low-cost mapping mobile robot to create the map of an unknown indoor environment. The experimental results have been obtained from the presented indoor environment with different sets of obstacles.

Keywords : Indoor environment, Mapping, Ultrasonic sensor, Heroku Platform

TABLE OF CONTENT

Title	Page No
Acknowledgement	**i**
Abstract	**ii**
List of Figures	**v**
Chapter 1	
PREAMBLE	**1-9**
1.1 Introduction	1
1.2 Existing System	3
1.2.1 Limitations of Existing System	3
1.3 Problem Statement	3
1.4 Objective of the project	3
1.5 Proposed System	4
1.5.1 Components of the proposed system	4
1.5.2 Interfacing ultrasonic sensor and raspberry pi	7
1.5.3 Working of the Proposed System	7
1.5.4 Advantages of the Proposed System	9
Chapter 2	
LITERATURE SURVEY	10-11
2.1 Related Work	10
Chapter 3	
SYSTEM REQUIREMENT SPECIFICATION	12-24
3.1 Functional Requirements	12
3.2 Non Functional Requirements	13
3.2.1 Product Requirements	13
3.2.2 User Requirements	13
3.2.3 Basic Operational Requirements	13
3.3 Hardware Requirements	13

3.4 Software Requirements………………………………………………………….. 24

Chapter 4

IMPLEMENTATION 25-27

4.1 Distance Sensing………………………………………………………… 25

4.2 Data transfer to server ………………………………………………….... 25

4.3 API function definition to get data……………………………………………. 25

4.4 API function to store data……………………………………………………….. 26

4.5 Mapping Function……………………………………………………………….. 26

Chapter 5

System Design 28-30

5.1 System Development methodology………………………………………….. 28

5.2 Design using UML………………………………………………………… 28

5.2.1 Dataflow Diagram………………………………………………………. 29

5.2.2 Usecase Diagram…………………………………………………………... 30

Chapter 6

SCREENSHOTS **31**

Chapter 7

CONCLUSION AND FUTURE SCOPE **32**

REFERENCES **33**

LIST OF FIGURES

Figure No.	Title	Page No.
1.5.1	Proposed system	4
1.5.2	Working of US sensor	5
1.5.3	Stepper motor	6
1.5.4	US sensor and raspberry pi interfacing	7
1.5.5	API architecture	8
3.1	Localization Architecture	12
3.2.1	User architecture	13
3.3.1	Raspberry Pi	13
3.3.2	Pin diagram of Raspberry Pi	14
3.3.3	US sensor	16
3.3.4	Pin diagram of US sensor	16
3.3.5	Arduino	17
3.3.6	Pin diagram of Arduino	18
3.3.7	Bluetooth module(HC-05)	20
3.3.8	Pin Diagram of HC-05	20
3.3.9	Robo chasis	21
3.3.10	L298D module	22
3.3.11	Pin diagram of L298D	22
3.3.12	Battery	23
3.4.1	IDE Arduino	24

3.4.2	Heroko	24
5.2.1	Dataflow diagram of obstacle detection	28
5.2.2	Use Case Diagram for 2D mapping using US sensor.	29
6.1	Output with mapping data	31

Chapter 1

PREAMBLEs

1.1 INTRODUCTION

Robotics is an emerging, revolutionary field and also a branch of engineering that includes electronics, mechanical engineering and computer science and so on. The above technologies will replace humans and their activities in coming years. Robotics has a high scope of new innovations and technologies.

Robotics and automation have several areas of application and hence covered almost every industrial field. Mobile Robotics is a subfield of robotics and information Engineering. Mobile robotics has its application in many areas including, navigation, medical, mapping, tracking, mobile transportation, unmanned vehicles, underwater exploration etc.

Robotic Mapping is a technique which is capable of constructing a potential map of an indoor environment and hence locating static obstacles in the given area and thus localize itself in a map. Mapping is used in an area which is unknown and an also cannot be matched using satellites. Sensor systems are used to gather data to identify the indoor environment and to build the corresponding map.

A mobile node uses a single ultrasonic sensor for obstacle detection and navigation of mobile robot in the unknown indoor environment. Ultrasonic sensor is one of the most commonly sensors used in mobile robots due to its low cost and simplicity of use.

In order to find the range of an obstacle in front of it, an ultrasonic sensor sends a small burst of ultrasonic waves and helps in measuring the time taken for the echo to be received.

The time-of-flight (TOF) of the echo pulse received from the obstacle within the ultrasonic range and speed of sound in air at standard temperature helps in measuring the distance from the obstacle. The range of ultrasonic sensor might vary from few centimetres to a several metres.

Most of the mobile robots are intended to be deployed with more than one sensor, but this technique proved to be inefficient in both cost wise and mainly cross talk arises due to interference of many ultrasonic sensors.

So, a mobile node is deployed with single ultrasonic sensor which can send information horizontally and also vertically and helps in creating sub maps in both directions.

The information gathered from the maps are merged to obtain to obtain a global map.

1.2 EXISTING SYSTEM

In the existing robots, very expensive sensors and interfaces are being used such as Kinetic cameras and controllers or laptops merged with lidar sensor which makes it an expensive ones more than others.

1.2.1 Limitations of Existing System

- In the existing system very, expensive sensors are being used hence cost increases.

1.3 PROBLEM STATEMENT

The major problem in existing system is that it is very expensive due to usage of costly sensors like rplidar. So automating devices using slightly lesser cost sensors will be an efficient way for the users.

1.4 OBJECTIVE OF THE PROJECT

As technology is advancing now-a-days, objectives of autonomous robot not only to move around in an unknown environment. It is extended to achieve many other features. Objectives are not constant. They are increasing day by day with the increase in advancement in technology.

In this project “2D mapping using Ultra Sonic sensor”, the main objectives to achieve are:

- Movement-

  The robot should be able to move on its own or with less human intervention.
- Mapping-

  A robot with ultra-sonic sensor should be able to map itself around the environment as the robot moves in an area.
- Display-

  The mapping process done by the robot should send the information or data points to the server which allows you to plot the graph accordingly.

1.5 PROPOSED SYSTEM

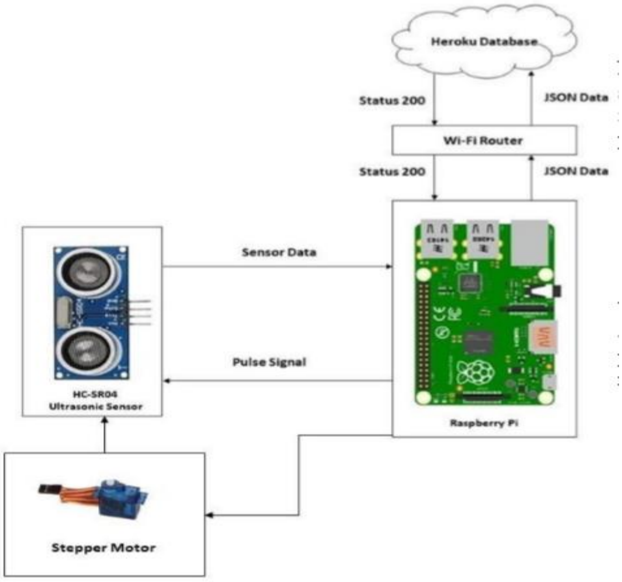

Fig 1.5.1: Proposed system

1.5.1 Components of the Proposed system

HC-SR04(ultrasonic sensor)

- **Working:**

The HC-SR04(ultrasonic sensor) is a 4pin module, whose pins are named as follows:

1. Vcc
2. Trigger
3. Echo
4. Ground

This sensor is popularly used in distance measuring or object sensing applications. This module has two eyes which is the ultrasonic transmitter and the ultrasonic receiver.

This sensor works with the formula:

Distance = speed * time.

The sensors transmitter transmits an ultrasonic wave, these waves travels in air until it hits an object. Once the waves hit the object it bounces back and the bounced back wave is caught by the ultrasonic receiver.

Fig 1.5.2 : Working of US sensor

In order to calculate distance using the above formula we need to know the speed of ultrasonic wave in room conditions which is 330m/s. The circuitry which is inbuilt for the on the module helps in calculating the time taken for the ultrasonic wave to come back and turns on the echo pin high for that same particular amount of time.

- **Using the ultrasonic sensor:**

The HC-SR04 sensor can be used with both microcontroller and microprocessor platforms like Arduino, Raspberry pie etc.

Powering of the sensor is done by providing +5V through Vcc and Ground pins of the sensor. The trigger and echo pins ate input pins and hence they can be connected to input pins of microcontroller. To start the process, the trigger pin has to be made high and then turned off. This will trigger an ultrasonic wave from the transmitter and the receiver will wait until it receives the wave. Once the wave is returned after it getting reflected by an object the echo pin will go high for some amount of time which is equal to the time taken for the wave to return back to the sensor.

- **Applications:**

1. This sensor is used to avoid obstacles and also for detecting obstacles.
2. Used to measure in the range of around 2cm to 400cm.
3. Can also be used to measure depth of certain places like well, pits etc since the waves can also penetrate through water.

Raspberry pi:

The Raspberry pi is a series of small single board computers developed in the UK by the Raspberry Pi foundation. It is now being used in many of the research projects because of its low cost and portability.

- **Working:**

The Raspberry Pi board is a Broadcom SOC(system on chip) board. It is equipped ARM1176JZF-S core CPU, 256 MB of SDRAM and 700 MHz. The 2.0 ports of the Raspberry Pi uses only external data connectivity options.
The board draws its power from a micro USB adapter with a min range of 2watts. The graphics, specialized chip is designed to speed up the operations of images calculations.

- **Applications:**

Used in many applications like Arcade machine, streamer, Tablet computer, Home automation, controlling robots and also in raspberry pi projects.

Stepper Motor:

A Stepper Motor or a step motor is a brushless, synchronous motor, which divides a full rotation into a number of equal steps.

Fig 1.5.3 : Stepper motor

- **Applications**

Gaming machines.

Small robotics.

3D printing equipment.

Textile machines.

1.5.2 Interfacing ultrasonic sensor with raspberry pi

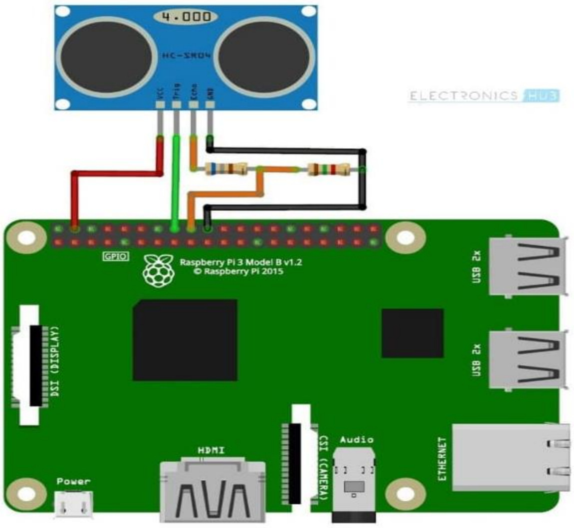

Fig 1.5.4 : US sensor and raspberry pi interfacing

The Raspberry Pi works at 3.3V logic while the ultrasonic sensor (HC-SR04) works at 5V.

The Raspberry Pi needs to read the Echo pin to calculate the time and hence accordingly the GPIO pin on the Raspberry Pi must be configured as Input so, before connecting the Echo pin to the Raspberry Pi, it must be given to the level converter.

Finally, provide the +5V and GND connections to the ultrasonic sensor from the Raspberry Pi pins.

1.5.3 Working of the proposed system:

- **Reading and Storing Architecture**

  The ultrasonic sensor has a 4 major pins :5V Supply (Vcc), Echo Pulse Output (ECHO), Trigger Pulse Output (TRIG), and ground (GND). Vcc is to power the module, GND is used by ground. The raspberry Pi sends an input signal to TRIG, which triggers the sensor to send an ultrasonic pulse. The pulse waves bounce off any nearby objects and some are reflected to the sensor. The returning waves are detected by the sensor and the time is measured between the trigger and returning pulse, which then sends a 5V signal to the ECHO pin. ECHO pin will be “low” until the sensor is triggered when it receives the echo pulse. Once triggered ECHO is set to “high” for the duration of the pulse.

  The raspberry pi is made wireless by getting the network information and configuring the wi-fi network settings. The controlling circuit calculates the difference in time using a python script embedded into the raspberry pi. The distance is calculated using the formula.

$$Speed = \frac{Distance}{Time}$$

- **RESTful API architecture:**

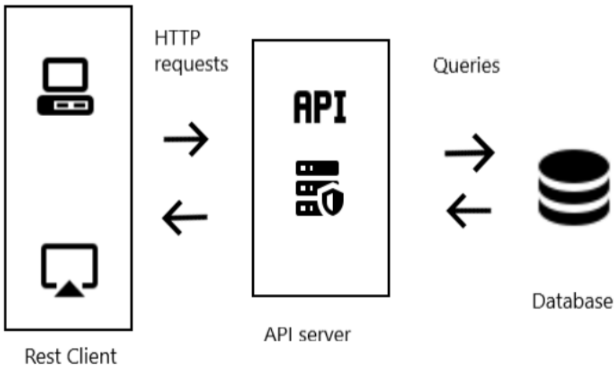

Fig 1.5.5: API architecture

The python script incorporates these calculations and sends the data from the wireless raspberry pi to the Heroku platform in JSON form. The JSON file also consists of the map number along with the degree of rotation at each interval until it covers a semi-circle rotation of 180˚. This data is stored in the Heroku Platform database where it can be accessed to plot the map.

Any time the data has to be stored or retrieved from the database, a HTTP call is made to the RESTful API which is also hosted in the Heroku platform, the idea behind a RESTful implementation for the API is that the data requests can originate from anywhere, any system and depending on the request the desired data can be stored or retrieved from the database provided the requests are valid. This gives immense flexibility to the robot as multiple sensors can send data to the database without having to interface each one of them individually, also the thought of future changes and upgrades to the robot becomes much simpler and more efficient.

- **Mapping and Plotting Architecture:**

The data will be mapped into a polar plot, which is a 2D plot where the radius r (in this case distance, d) is plotted against angles from 0° through 360°, but in our case here we only plot till 180°. Hence the data to plot the map will require two fields: the angle and the distance. The angles along with their respective distances are put through a loop to sequentially plot through the entire data.

The mapped data will be displayed on the web application that is also hosted in the Heroku platform, it makes a API call to receive the required map data, and it receives the data in the JSON format, which the function running app converts to a dictionary.

This plotting is done with the help of pythons Matplotlib library, which provides a function polar which takes two parameters, the angles and the distance. The angle first needs to be converted to radians before the mapping process, after which the two lists one for the angles and the other for the distances are provided to the polar function which then plots the polar plot for us.

This polar plot is then converted into an image before being displayed in the app so that the resulting output from the Matplotlib library is compatible with all the devices

1.5.4 Advantages of Proposed System

- Real time monitoring of robot at anytime
- Low cost
- By implementing this model, robot can be automated

Chapter 2

LITERATURE SURVEY

2.1 RELATED WORK

- **Two Dimensional Mapping by using Single Ultrasonic Sensor by Neelam Barak, Neha Gaba, Shipra Aggarwal.** With the advancements in robotics in the recent years, automation due to robotics has been in almost all industries and due to the high efficiency and reliability of robots it has reduced human intervention and load and thereby reducing human errors as well. Mapping is a technique which generates a map of an indoor environment that cannot be mapped using satellites. This paper proposes a cost-effective solution to automate the mapping process using a single ultrasonic sensor. This sensor is present on a mobile robot. The mobile robot has an IC motor along with ATMEGA8 microprocessor that is used for the main control. The mapping data from the ultrasonic sensor is sent to be processed through a Bluetooth module. The map is made using the MATLAB simulation software. A serial communication is created between the robot and the MATLAB software, it is controlled through a graphical user interface defined in MATLAB. The robot moves along the walls of the indoor environment in an L shape and the ultrasonic-sensor is mounted on the left of the robot hence will be facing the centre of the environment. When the robot moves along the first wall, it generates one map, now completing the L-shape when it moves along the second wall, it will generate another map. The sensor can detect obstacles within a range of one and half metres. These two maps are merged to create the final map of the indoor environment.

- **A SLAM algorithm for indoor mobile robot localization using an Extended Kalman Filter and asegment based environment mapping by Luigi D'Alfonso, Andrea Griffo, Pietro Muraca, Paolo Pugliese.** Simultaneous Localisation and Mapping is a problem that includes the concurrent estimation of the robot position and orientation and of the environment where the robot moves in. To solve this problem the robot needs to build a map of the environment and localise itself within it. In this paper the SLAM problem for a mobile robot in an unknown indoor environment has been faced. The environment has been modelled using a set of segments and segments' starting and ending points have been used as SLAM landmarks. A Segment based mapping algorithm has been proposed. The algorithm is divided into 5 main parts: landmark extraction, data association, state estimation, state update and landmark update. Using the sonar measurements and the actual robot pose estimation, the data association and landmark extraction processes are performed and yield to the actual environment mapping. Starting from this mapping and from the model inputs, the state estimation, state update and landmark update processes provide the robot pose and update the environment map. The SLAM algorithm consists of a modified Extended Kalman filter suitably adapted to face both the mobile robot localization problem and the surrounding environment mapping problem. Measurements taken from a set of on board ultrasonic sensors have been used to update the robot pose estimation and the environment mapping.

Chapter 3

SYSTEM REQUIREMENT SPECIFICATION

3.1 FUNCTIONAL REQUIREMENTS

AN Arduino board is used as the main control that acts as a microprocessor. A L293D motor driver module is used to interface the motors with the microprocessor. The autonomous robot uses HC-SR04 ultrasonic sensor to relay the distance of the obstacle.

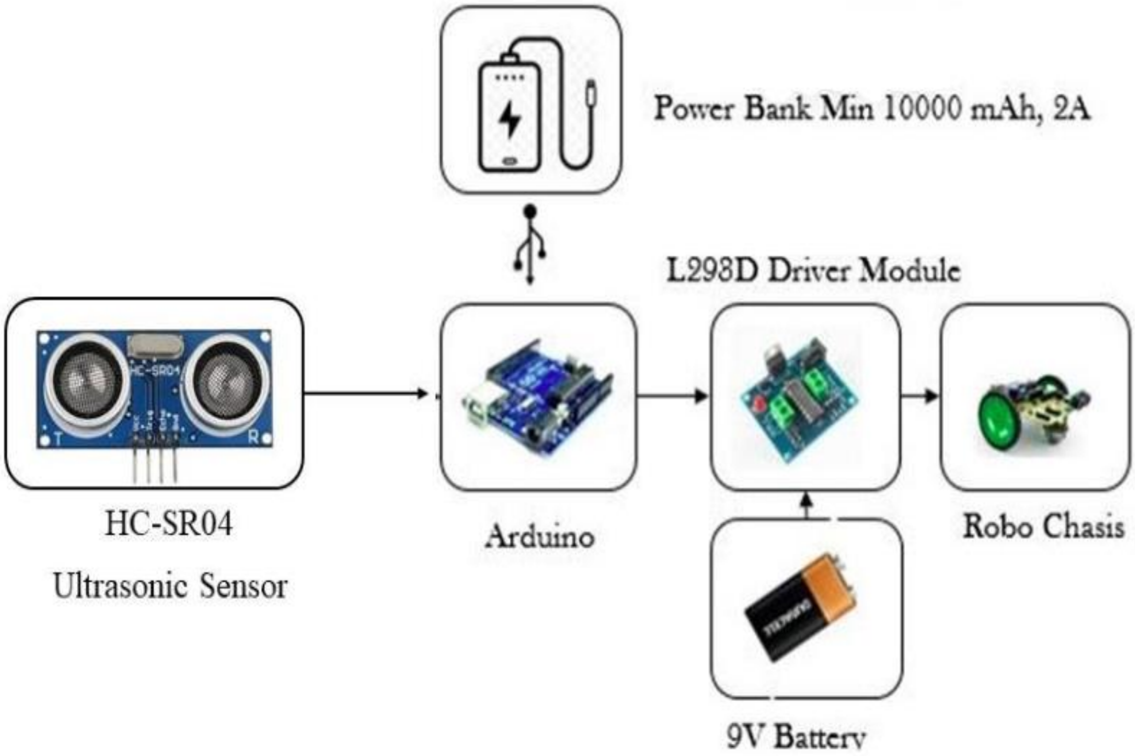

Fig 3.1 : Robot Localization Architecture

A robot chassis which contains the motor is interfaced with the Arduino through the Driver L293D motor driver and a 9V battery is connected to the driver circuit. A power bank of minimum 10000 mAh is used to provide continuous supply to the Arduino.

3.2 NON-FUNCTIONAL REQUIREMENTS

3.2.1 PRODUCT REQUIREMENTS

The overall concept of 2D mapping using autonomous robot system is to include additional functionalities, i.e., to show the user the mapping of an unknown environment using an ultrasonic sensor.

3.2.2 USER REQUIREMENTS

The user excepts the system to be friendly by providing the features like reading the mapping done by sensor and also if possible, the visual display of unknown environment, so that the user gets to know the exact location of the robot in the environment.

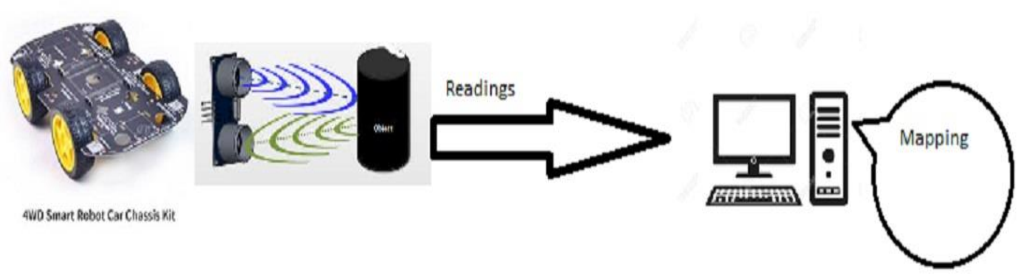

Fig 3.2.1 : User Architecture

3.2.3 BASIC OPERATIONAL REQUIREMENTS

The operational requirements describe the basic overall requirements for the system, how it interacts with the other systems and system performance goals.

A continuous power supply has to be provided in order to move the robot and get the mapping done accordingly.

3.3 HARDWARE REQUIREMENTS

1 Raspberry Pi:

Fig 3.3.1 : Raspberry Pi

The Raspberry Pi is a low cost and a small card sized computer that plugs into a computer monitor or even TV, and uses a standard keyboard and mouse.

Raspberry has the ability to interact with outside world and has been used widely in digital projects.

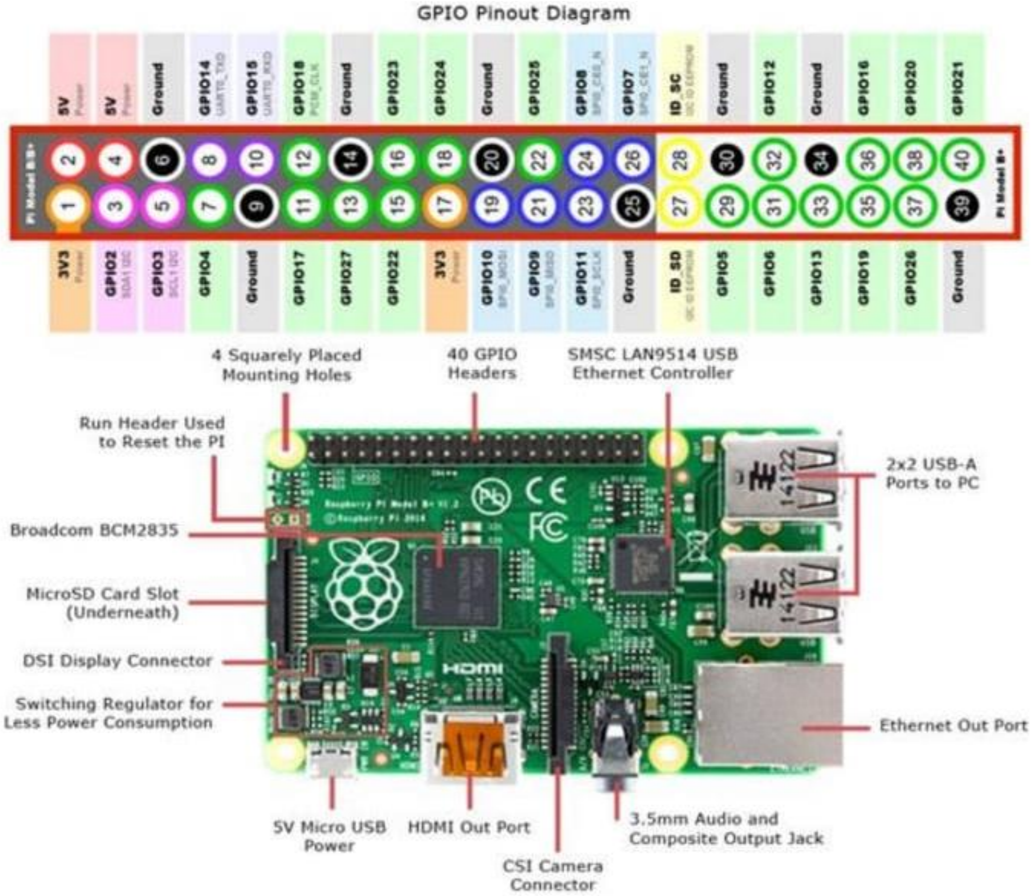

Fig 3.3.2 : Pin diagram of Raspberry Pi

- **Hardware specifications of Raspberry Pi**

1. **Memory**

   The raspberry pi model is designed with 256MB of SDRAM and model B is designed with 51MB.

   The RAM memory is available more than 256MB or 512MB.

2. **CPU(Central Processing Unit)**

   The CPU of the Raspberry Pi is the brain of the raspberry pi board and is responsible for carrying out the instructions of the computer through logical and mathematical operations.

3. GPU (Graphics Processing Unit)

The GPU is a specialized chip in the raspberry pi board and is designed to speed up the image operations. The board is designed with Broadcom video core IV and it supports OpenGL.

4. Ethernet Port

The ethernet port in the raspberry pi board is the main gateway for communicating with other devices. It can also be used to connect home router to access the internet.

5. GPIO pins

The raspberry pi GPIO pins are generally used to associate it with other electronic boards. These pins can accept input and output commands based on programming raspberry pi. For example, we can connect ultrasonic sensor to detect the distance.

6. XBee Socket

This is used in the raspberry pi board for the wireless connections.

7. Power Source Connector

The power source is a small switch which is placed at side of the shield. The main purpose is to enable external power source.

8. UART

The Universal Asynchronous Receiver/Transmitter is a serial input and output port. It can be used to transfer data serially in the form of text.

9. Display

The connection of the raspberry pi board is of two types such as HDMI and composite. Many LCD and HD TV monitors can be attached suing HDMI. Older TVs can be connected using composite video.

2 Ultrasonic sensor (HC-SR04):

Fig 3.3.3 : US sensor

It is a device that measures the distance between itself and the object. It comes with ultrasonic transmitter and receiver modules.

- **Pinout diagram of Ultrasonic sensor**

Fig 3.3.4 : Pin diagram of US sensor

Vcc : The Vcc pin powers the sensor, mostly with 5V.

Trigger : The trigger pin is an input pin and has to be kept high for 10us to initialize measurement by sending US wave.

Echo : The Echo pin is an output pin. This pin goes high for some amount of time which will be equal to time taken for the US wave to return back to the sensor.

Ground : This pin is to be connected to the Ground of the system.

- **Features of HC-SR04 sensor :**

  Operating voltage : +5V

  Measuring distance : 2cm to 450cm

  Accuracy : 3mm

  Operating current : <15mA

  Operating Frequency : 40Hz

  Measuring angle covered : <15º

3 **Arduino:**

Fig 3.3.5 : Arduino

Arduino uno is a microcontroller board based on *bit ATmega328P microcontroller. It also consists other components such as voltage regulator, serial communication etc. It is an open source platform used for building electronics projects. Arduino consists of both physical programmable circuit board and a piece of software, or IDE that runs on your computer.

Arduino uses a simplified version of C++, making it easier to learn program.

Finally, Arduino breaks out the functions of the microcontroller into a more accessible package.

- **Pinout diagram of Arduino**

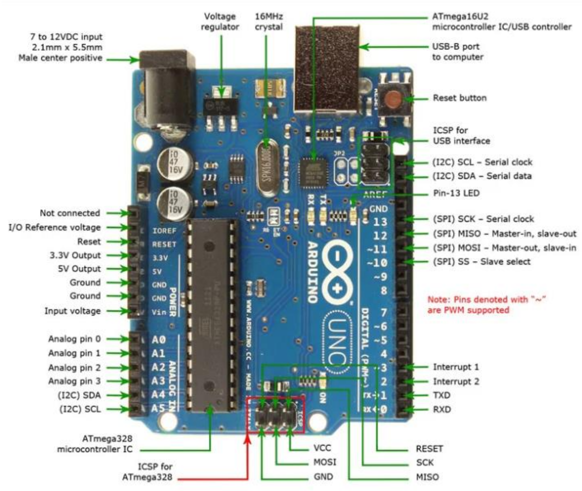

Fig 3.3.6 : Pin diagram of Arduino

Vin: input voltage to Arduino when using an external power source.

3.3V: generated by on board voltage regulator.

GND: ground pins.

RESET: resets the microcontroller.

Analog pins(A0-A5): used to provide the analog input in the range of 0-5V

Input/Output(Digital pins 0-13): can be used as input or output pins.

Serial(0(Rx), 1(Tx)): used to transmit and receive serial data.

External Interrupts(2,3): triggering an input.

PWM(3,5,6,9,11): provides 8-bit PWM output.

SPI(10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK)): used for SPI communication.

Inbuilt LED(13): to turn on the inbuilt LED.

TWI(A4 (SDA), A5 (SCA)): used for TWI communication.

AREF: To provide reference voltage for input voltage.

- **Arduino uno technical specifications:**

Microcontroller: ATmega328P

Operating voltage: 5V

Input voltage: 7-12V

Input Voltage limits: 6-20V

Analog input pins: 6(A0-A5)

Digital I/O pins: 14

DC current on I/O pins: 40mA

DC current on 3.3V pin: 50mA

Flash Memory: 32KB

SRAM: 2KB

EEPROM: 1KB

FREQUENCY: 16MHz

- **Working of Arduino**

The 14 digital input/output pins can be used as input or output pins by using pinMode(), digitalRead() and digitalWrite() functions in arduino programming. Each pin operate at 5V and can provide or receive a maximum of 40mA current, and has an internal pull-up resistor of 20-50 KOhms which are disconnected by default. Out of these 14 pins, some pins have specific functions as listed below:

Serial Pins 0 (Rx) and 1 (Tx): Rx and Tx pins are used to receive and transmit TTL serial data. They are connected with the corresponding ATmega328P USB to TTL serial chip.

External Interrupt Pins 2 and 3: These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.

WM Pins 3, 5, 6, 9 and 11: These pins provide an 8-bit PWM output by using analogWrite() function.

SPI Pins 10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK): These pins are used for SPI communication.

AREF: Used to provide reference voltage for analog inputs with analogReference() function.

Reset Pin: Making this pin LOW, resets the microcontroller

4 HC-05:

Fig 3.3.7 : Bluetooth module(HC-05)

It is a Bluetooth module which is designed for wireless communication. It can be used to communicate between two microcontrollers like Arduino or communicate with any device with Bluetooth functionality like phone or laptop.

- **Pinout Diagram of HC-05**

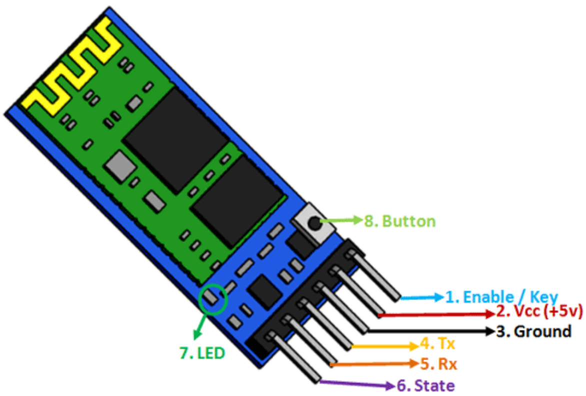

Fig 3.3.8 : Pin Diagram of HC-05

Enable/Key: This pin is used to toggle between Data Mode(set low) and AT command mode(set high).But by default it will be in Data mode.

Vcc: Powers the module. Connect with +5V.

Ground: Ground pin of module, connect to system ground.

Tx-Transmitter: transmits serial data.

Rx-Receiver: receives serial data.

State: The state pin is connected to on board LED, it can be used to check whether the Bluetooth is working properly.

- **Technical specifications**

  Operating Voltage: 4V to 6V

  Operating current: 30mA

  Range: <100m

  Works with serial communication and TTL compatible.

  Can be easily interfaced with laptops or mobile phones using Bluetooth.

- **Working**

  The Bluetooth module has two operating modes, one is the data mode in which it can send and receive data from other devices and other is AT command mode.

  The module can be easily paired with microcontrollers because it operates with serial port protocol (SPP). Just power the module with +5V and connect the Rx pin of the module to the Tx of MCU and Tx pin of module to Rx of MCU. During power up the key pin can be grounded to enter into command mode, if left free it will go to data mode by default. As soon as you power the module u will be able to see the Bluetooth device "HC-05" then connect it with the default password 1234 and start communicating with it.

- Applications

  Wireless communication between two microcontrollers.

  Communicate it with Laptop, Desktop and mobile phones.

  Data Logging application.

  Wireless robots.

  Home automation.

5 **Robo Chassis:**

Fig 3.3.9 : Robo chasis

It comprises the body of a robot. The chassis plate can be used with out custom length robot platform.

6 Driver Module((L298D):

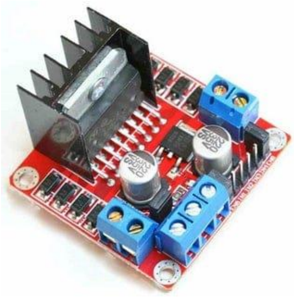

Fig 3.3.10 : L298D module

The L298N motor river module is a high-power motor driver module for driving Dc and stepper motors. This module can control up to 4 DC motors or 2 DC motors with directional and speed control.

- **Pin-Out Diagram**

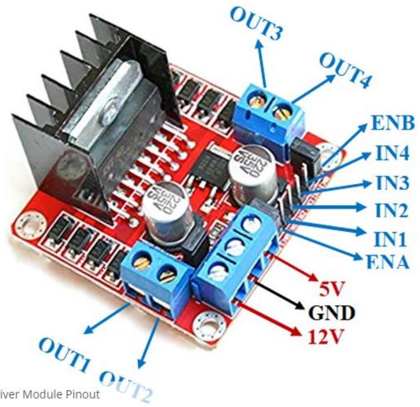

Fig 3.3.11 : Pin diagram of L298D

IN1 & IN2: Motor A input pins. Control spinning direction of Motor A.

IN3 & IN4: Motor B input pins. Control spinning direction of Motor B.

ENA: Enables PWM signal for Motor A.

ENB: enables PWM signal for Motor B.

OUT1 & OUT2: Motor A output pins.

OUT1 & OUT2: Motor B 22 output pins.

12V: DC power source.

5V: Supplies power for the switching logic circuitry inside L298N IC.

GND: Ground pin.

- **Technical Specifications**

  Supply Voltage: 5V.

  Supply Current: 2A.

  Driver Voltage: 5-35V.

  Driver Current: 2A.

  Maximum power: 25W

- **Working**

  The L298N Motor Driver module consists of an L298 Motor Driver IC, 78M05 Voltage Regulator, resistors, capacitor, Power LED, 5V jumper in an integrated circuit.

  When the power supply is less than or equal to 12V, then the internal circuitry will be powered by the voltage regulator and the 5V pin can be used as an output pin to power the microcontroller.

  ENA & ENB pins are speed control pins for Motor A and Motor B while IN1& IN2 and IN3 & IN4 are direction control pins for Motor A and Motor B

- **Applications**

  Drive DC motors.

  In robotics.

7 **Battery:**

Fig 3.3.12 : Battery

3.4 SOFTWARE REQUIREMENT

The software used to develop an application is Arduino IDE and Heruko Database. The **Arduino Integrated Development Environment** is a cross platform application that can be written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards.

Fig 3.4.1 : Arduino

Heroku is a cloud platform as a service (PaaS) supporting several programming languages. One of the first cloud platforms, Heroku has been in development since June 2007, when it supported only the Ruby programming language, but now supports Java, Node.js, Scala, Clojure, Python, PHP, and Go.

Fig 3.4.2 : Heroku

Chapter 4

SYSTEM DESIGN

4.1 System Development Methodology

A software development methodology or system development methodology is a framework that is used to structure, plan, and control the process of developing an information system. A wide variety of such frameworks have evolved over the years, each with its own recognized strengths and weakness. There are many development methodologies but different projects will involve different methodologies involved. The selection of methodologies can be based on various technical, organizational, project and team considerations.

4.2 Design Using UML

4.2.1 Dataflow Diagram

Dataflow diagrams are the graphical representation of the data “flow” through any information system that is built. It is usually used as a primary step to create a complete overview of the system without going into great detail, which in later stages are elaborated. It shows how the data in the system moves through the system and be stored. Unlike flowchart it doesn't show any details about process timings or sequence of operations but only concentrates on the control flow.

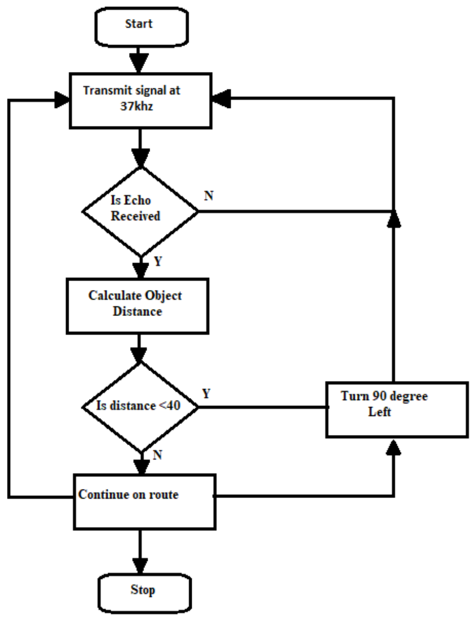

Fig 4.2.1. Dataflow diagram of obstacle detection

According to the above dataflow diagram, the system consists of an ultrasonic sensor, a microcontroller, and a stepper motor.

The ultrasonic sensor is implemented to detect obstacles on the robot's path by sending signals to an interfaced microcontroller. The microcontroller then redirects the robot to an alternate direction to avoid the detected obstacle. The performance evaluation of the system indicates an accuracy of 85% and 0.15 probability of failure respectively.

4.2.2 Use Case Diagram

Use Case Diagram is a diagram used to model dynamic behaviour i.e., behaviour of a system when it is running/operating of any system. As it involves dynamic interaction there should be external and internal factors for making interactions. These factors are known as actors. Use case diagrams consists of actors, use cases and their relationships. The purpose of use case diagram is to gather the requirements of a system it could be internal or external influences. Use case are drawn to capture functional requirements of a system.

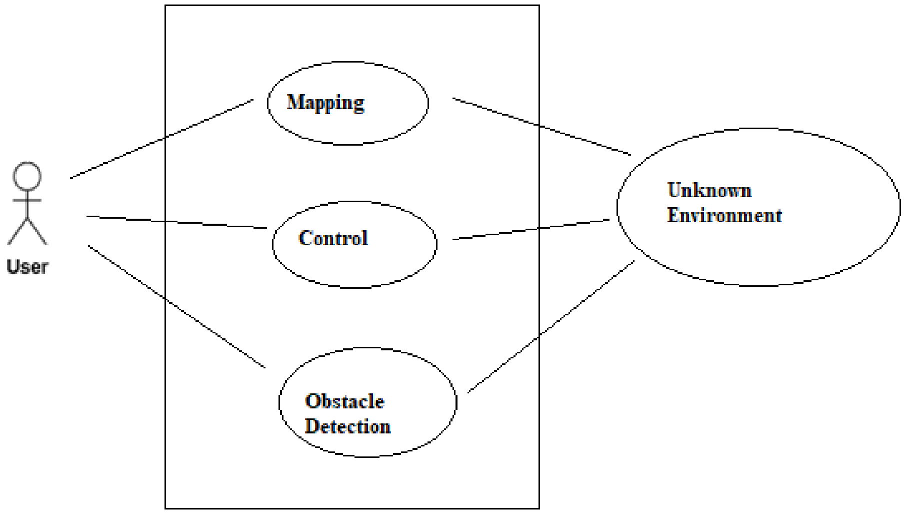

Fig 4.2.2 Use Case Diagram for 2D mapping using US sensor.

In the above Use Case diagram suitable actors are selected and it shows the relationship and dependences clearly in the diagram. The name of the Use Case is very important & It should be chosen such that It can identify the functionality performed. It does not include all types of relationships, as the main purpose of the diagram is to identify the requirements. The actors in the above Use Case Diagram are User and3 Unknown Environment. And the Use Cases are Mapping, Control, Obstacle Detection All of these Use Cases are related to the two actors it is shown using Arrows. These diagrams are used at very high-level designs. It specifies the events of the system and their flows. But it doesn't describe how are they implemented.

4.2.3 Sequence diagram for Ultrasonic sensor

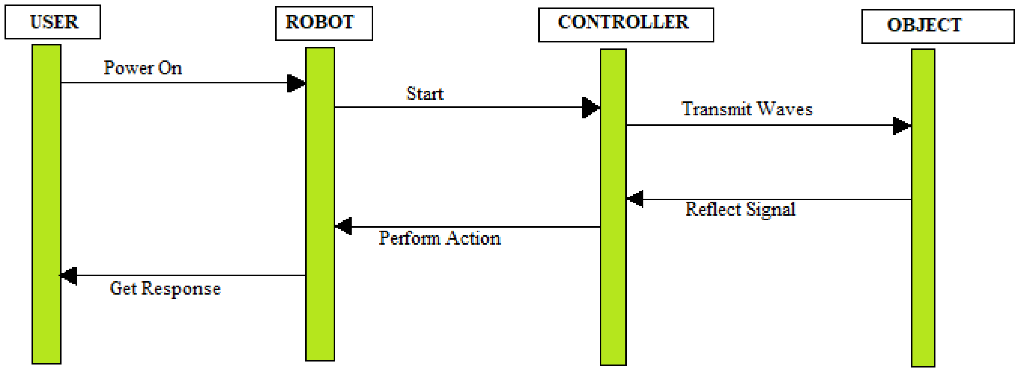

Fig 4.2.3 Sequence Diagram for US sensor.

When the user switches on the robot, the controller also starts working. The controller contains the Ultrasonic sensor which is capable of finding the distance between the robot and the random object. The ultrasonic sensor transmits the signals and when the signals hits the object it bounces back. The reflected signal is received by the ultrasonic sensor. The time difference between the transmission of the wave and received wave is taken into consider. And the distance between the robot and object is calculated using the following formula:

$$Speed = \frac{Distance}{Time}$$

4.2.4 Sequence diagram for Mapping

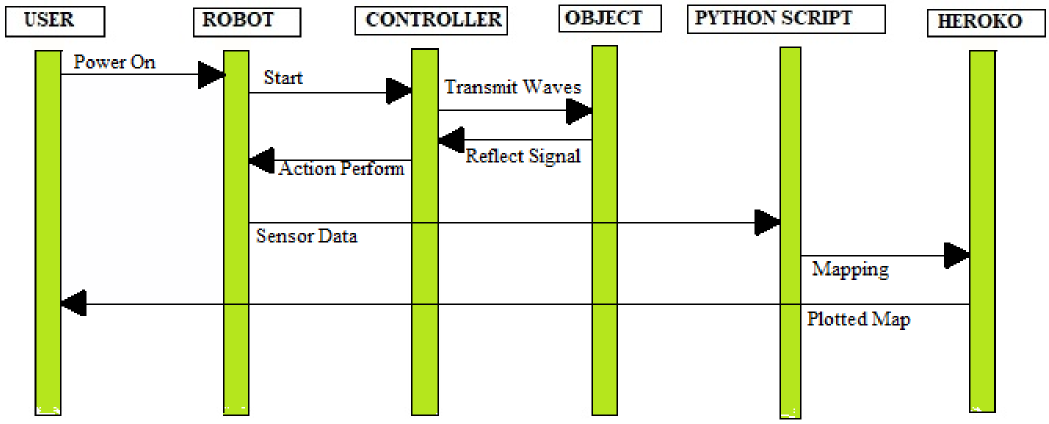

Fig 4.2.4 Sequence diagram for Mapping

When the user switches on the robot, the controller starts and the ultrasonic sensor transmits the signal and when it gets reflected from the object the sensor receives it. Then the distance between the robot and object is calculated. The sensed data is then sent as an input to the python script and the mapping is done by the Heroku and is thus displayed to the user.

Chapter 5

Implementation

5.1 Distance Sensing

```
distance = sensor.distance
    # But we want in centimetres
    distance1 = sensor.distance*100
    distance1 = round(distance1,2)
    data['coordinates'].append({
              'dist':distance1,
              'angle':x;
              'time_stamp':x,
              'map_no':j
    })
    x = x+3
```

5.2 Data transfer to Server

```
url = "http://mapper-api.herokuapp.com/api/storeData
              r = requests.post(url, json.dumps(data))
              print(str(r.content))
              file2 = open("file_no.txt","w")
               file2.write(str(j))
               file2.close()
```

5.3 API function definition to get data

```
@ app.route ('/api/getData', methods = ['GET'])
def getData ( ):
    print(request.data )
    content = json.loads(request.data)
    return Data =   dbModel.query.filter_ b y(mapNumber= content).all()
```

5.4 API function to store data

```
@app.route('/api/storeData', methods=['POST'])
def store():
    content = json.loads(request.data)
    for i in content['coordinates']:
        angl = i['angle']
        dist = i['dist']
         mapNum = i['map_no']
        timeSt = i['time_stamp']
         data = dbModel(angl, dist, mapNum, timeSt)
         db.session.add(data)
        db.session.commit()

return 'SUCCESS'
```

5.5 Mapping function

```
def createMap(data ):
d = { }
d['coordinates'] = data
 # reading coordinates from database
for i in d['coordinates']:
    phi=i['angle']
    rho =i['dist']
    x = np.deg2rad(phi)
    angle.append(x)
    dist.append(rho )
 # Plotting Graph
plt.polar(angle,dist)
fig = plt.figure(frameon = Fals e )
ax = fig.add_sub plot(111, polar= True )
c = ax.plot(angle, dist, color='r', linewidth = 3 )
ax.set_thetamin(0)
ax.set_thetamax(180)
```

```
ax.set_theta_offset(2 * np.pi)

# Convert plot to PNG image
pngImage = io.BytesIO( )
FigureCanvas(fig).print_ png(pngImage)
return pngImage
```

5.6 Stepper Motor 180-degree Rotation

```
GPIO.setmode(GPIO.BOARD)
GPIO.setup(22,GPIO.OUT)
servo1 = GPIO.PWM(22,50)
servo1.start(0)
print ("Waiting for 2 seconds")
time.sleep(1)
print ("Rotating 180 degrees in 60 steps")
angle = 180
# Loop for angle values from 0 to 180 (0 to 180 degrees)
while angle >=0:
   servo1.ChangeDutyCycle(2+(angle/18))
   time.sleep(0.3)
   servo1.ChangeDutyCycle(0)
   #time.sleep(1.0)
   angle = angle - 3
print ("Turning back to start")
duty=12
servo1.ChangeDutyCycle(2+(180/18))
# Wait a couple of seconds
time.sleep(2)
#servo1.ChangeDutyCycle(12)
#time.sleep(0.5)
servo1.stop()
GPIO.cleanup()
print ("Goodbye")
```

CHAPTER 6

SCREEN SHOTS

After deploying the rover in the environment and letting it gather sufficient data, we can view the map that has been built from the said data on our website.

The end result can be accessed from anywhere in the world using any device with seamless internet connection or Wi-Fi capabilities.

The output is generated on the webpage URL http://mapper-api.herokuapp.com/ which is dynamic and hence always shows the latest data .

By clicking on the search tab and entering the appropriate map number whose values are already sent and stored in the database it is possible to access and view the maps with their values which consists of the angle and the distance

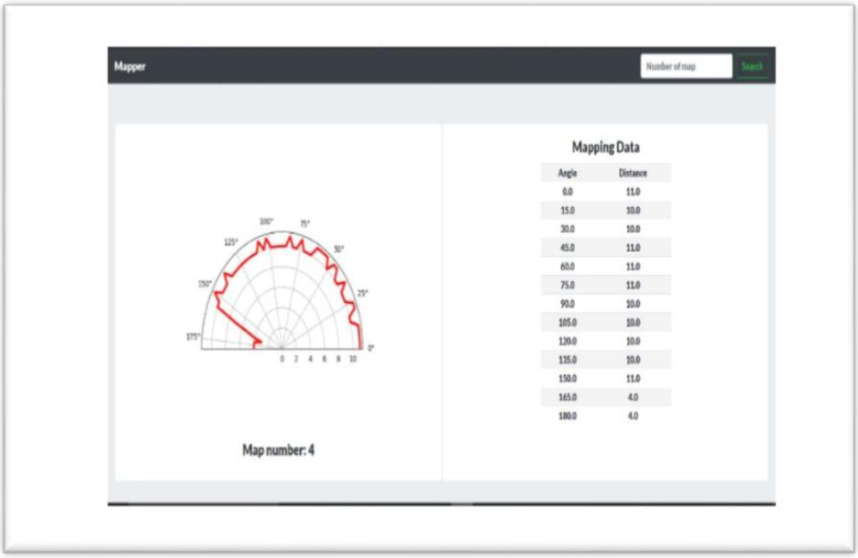

Fig 6.1 : Output with mapping data

Figure 6.2 is a snapshot that contains data from the stepper motor and printed on the console.

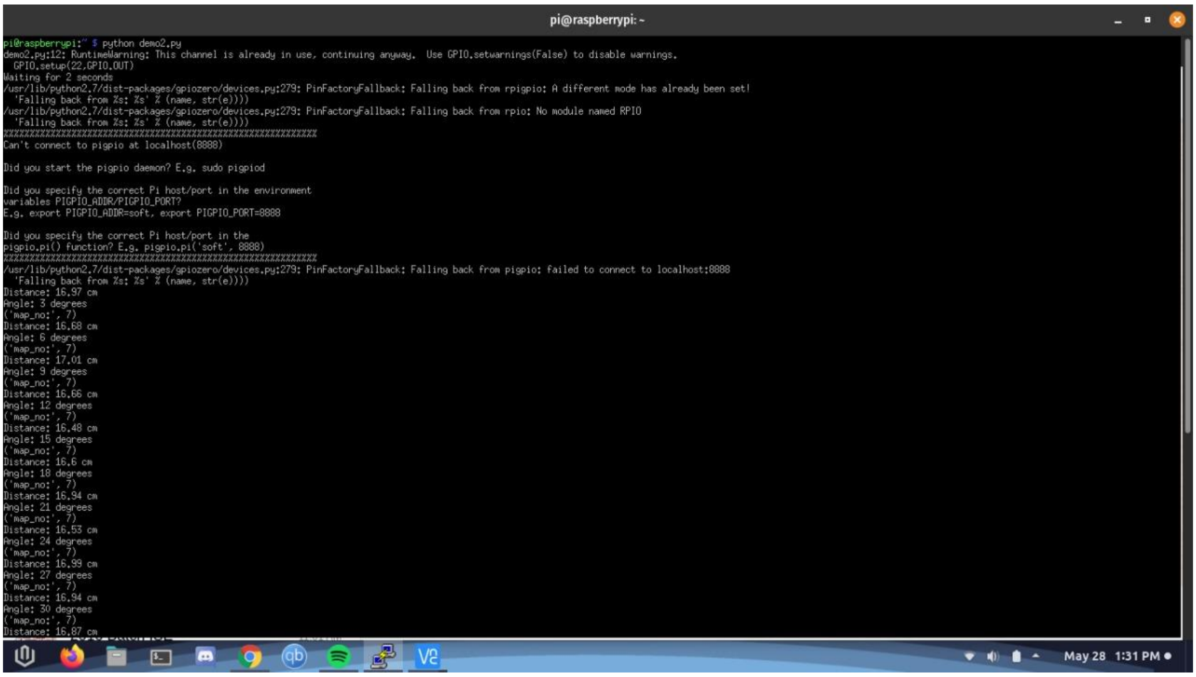

Fig 6.2 : Output from the stepper motor

Chapter 7

Conclusion and Future Scope

We first deployed the rover in the environment. It then moved around on its own simultaneously detecting obstacles and mapping its surroundings. The rover requires some basic internet connection that can be provided by way of a mobile hotspot. This generates multiple singular maps about the surrounding of the rover from a particular position of the rover in the environment.

By stitching all the resultant maps from a single run of the rover, we can generate a single coherent map of the entire surrounding, although this functionality has not been implemented yet due to a lack of computational resources at the server side and a lack of funds to upgrade on our current resources. The rover can run for upto 8 hours with the help of a 20000 mAh power bank as the power source. With this we proved that a cost-efficient method exists for such a system. We built it using many alternatives that would otherwise seem inappropriate to use.

The Proposed system can be further improved an altered in future depending on the need and requirements. Use of Raspberry pi enables many of these as it allows for multiple functionalities.

- It can further be improved by stitching the maps to create a coherent map of the surrounding, instead of multiple maps that have to be analyzed.
- The speed of the mapping and distance ranging can be improved by using a higher quality sensor.
- It can also be used to survey buildings and other such human constructs greatly reducing the need for human intervention in such scenarios.
- This can be used in land surveys where the rover can be deployed to create a surrounding map of the land. The Raspberry Pi can also be attached with multiple soil quality sensors that can be used for surveying the suitability of the soil in the area for multiple reasons.

References

- Anandabalan.S, Avinash U.A, J.Chiraj, Harish.S, Vivek Thiyagarajan .G, Adarsh. S. et al., "Indoor mapping using ultrasonic sensor," Proc. IRF International Conference, April 2014, pp.52-53 , doi:ISBN: 978-93-84209-07-0.
- Hyoung Jo Jeon and Byung Kook Kim, "Feature-based probabilistic map building using time and amplitude information of sonar in indoor environments," Robotica, vol. 19, July 2001, pp. 423-437, doi: 10.1017/S0263574700003180.
- S.Kodagoda, E.A.S.M. Hemachandra, P. G. Jayasekara, R. L. Peiris, A. C. De Silva, Rohan Munasinghe, "Obstacle Detection and Map Building with a Rotating Ultrasonic Range Sensor using Bayesian Combination," Proc. Of International Conference on Information and Automation, IEEE Press, Dec. 2006, pp. 98-103, doi: 10.1109/ICINFA.2006.374159.
- Sonali K.Kalmegh, Dharmesh H.Samra, M Rasegaonkar Nishant, "Obstacle avoidance for a mobile exploration robot using a single ultrasonic range sensor," Proc. of International Conference on Emerging Trends in Robotics and Communication Technologies (INTERACT), IEEE Press, Dec. 2010, pp. 8 – 11, doi:10.1109/INTERACT.2010.5706156.
- https://thepihut.com/blogs/raspberry-pi-tutorials/hc-sr04-ultrasonic-range-sensor-on-the-raspberry-pi
- https://flask-sqlalchemy.palletsprojects.com/en/2.x/
- https://jinja.palletsprojects.com/en/2.11.x/
- https://www.dexterindustries.com/howto/run-a-program-on-your-raspberry-pi-at-startup/
- https://devcenter.heroku.com/articles/getting-started-with-python /