NUST College of
Electrical and Mechanical Engineering CEME

Design and Prototyping of a Serving Robot for a Hotel Environment.

Project Report
DE-40 (DEE)

Submitted By:

Muhammad Helal Abdullah khan Mayo
(217818)
DE-40 EE-B

Bachelors In
Electrical Engineering
Year
2022

Project Supervisor:

Asst Prof Dr. Ahmed Rauf Subhani

Project Co-Supervisor:

Asst Prof. Sobiya Haye

NUST COLLEGE OF ELECTRICAL AND MECHANICAL ENGINEERING PESHAWAR ROAD, RAWALPINDI

DECLARATION

I hereby declare that no portion of the work referred to in this Project Thesis has been submitted in support of an application for another degree or qualification of this of any other university or other institute of learning. If any act of plagiarism found, I am fully responsible for every disciplinary action taken against me depending upon the seriousness of the proven offence, even the cancellation of my degree.

COPYRIGHT STATEMENT

ACKNOWLEDGMENTS:

First of all, i would like to thank ALMIGHTY ALLAH who enlightened me with the knowledge and ability needed for the completion of the project. I would like to pay tribute to my directing supervisors Asst. Prof Dr. Ahmed Rauf Subhani and Asst Prof. Sobiya Haye for their kind supervision and guidance throughout the semester, otherwise without their directions and experience the undertaking of this project would not have been possible i are truly indebted to their guidance and support.

I highly appreciate the assistance by all my teachers and support staff especially the faculty of Department of Electrical Engineering who guided me in solutions of many problems especially to the Head of Department. I would also like to extend my gratitude to all those who encouraged and helped me in my endeavor. I would also like to pay my thanks and love to my parents who have been constantly praying for my success.

Abstract

This project implements the design of a serving robot for a hotel environment. Robot will deliver the food at the desired table. In this system a tray that is placed on the top of robot will carry the food to the table. In the restaurant when a customer enters, he/ she will place the order and pay the bill and sit at the table that is given to him by the person at reception. After the order is placed on the top of the robot, Robot will follow the line i.e. localize and then plan a path to its destination. Along with the line following, the Robot will also deal with any obstacle that comes in the way. A sensor will provide help in avoiding the obstacle and Robot will then deliver the order at the table.

Table of Contents

Chapter 1

Introduction

"Humans are limited in the attention, kindness and compassion that they can expend to others, but AI based compassionate robots can channel virtually unlimited resources into building compassionate relationships in the society."

Machines that are produced by the involvement of science, engineering and technology are called robots. Robots are the substitution of Human actions. Robots have always become the inspiring factor and fascinated the culture. A robot is made by taking out one capability of humans and digitalize it. Robots were once thought of as a monotonous machine that would do their job as required. As we know, Necessity is the mother of invention and invention is the first step to innovation so, this was the case in the field of robotics. Nowadays we live in a world in which robots have the tendency to assist humans based on their capacity of self-ability or we can say that how autonomous the machine is.

1.1 Motivation

Automation in field of Robotics is the future. With the progressing world and the never say never attitude humans tend to invent and innovate a lot nowadays. I got motivated by watching videos of the inventions in the field of Robotics, A good example of that is a robot in Multan that is currently serving food in a hotel there. The robots are now becoming the need of the modern world. Some companies are using these Robots to assemble the parts. The basic need of hour is to increase the efficiency of the system and creating a machine that can work tirelessly.

1.2 Problem statement

The problem arises when in the technological advanced world people still stick to the old ideas rather than innovating one. As the restaurant owner has to pay a huge amount of salary to several waiters which is the problem, and i want to reduce the expense of hotel by introducing the serving robot which is one time investment, and it could work more efficiently than waiters. So, i thought of a solution to change hotel environment by introducing serving robot which can work more tirelessly and efficiently than humans. Moreover, humans are the name of emotions and sometimes due to emotions the work gets affected but that won't be a case in robot as it is emotionless.

1.3 Background

The rise and implementation of a Robots is a decade long process which has evolved a lot in the past years. It started as a single project and now has reached many industries. Nowadays Robots are in many areas, i come to know about them continuously either through internet or any other news medium.

The basic mechanism used by any robot is "Sense, Plan and Act ". First thing that a robot does is sense the surroundings by the use of a sensor or a camera. Second step is to plan via the onboard computer in the robot by the information provided by the sensors. Third step is to take the action based on the information. This process is shown in the figure below:

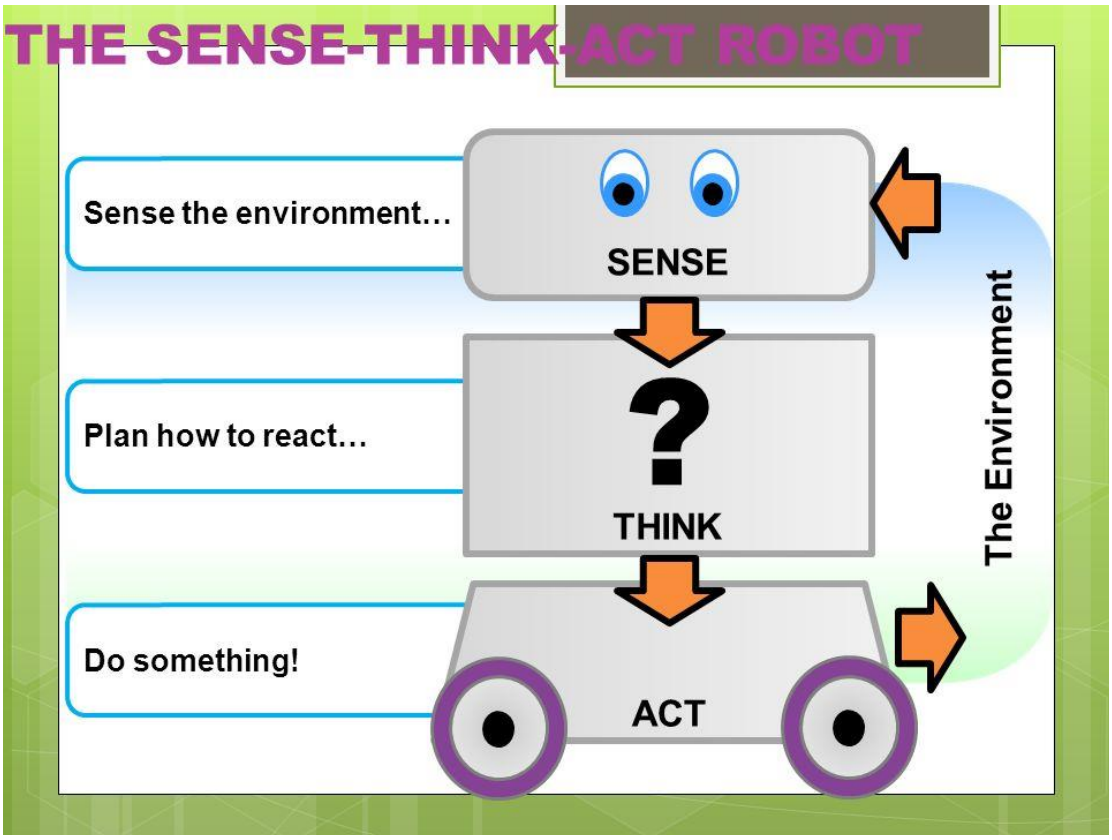


Figure 1.1 SENSE, PLAN, ACT

1.4 Scope

The project has been uniquely designed for serving purpose in restaurants. Its aim is to serve order from counter to the assigned table using line following mechanism. The aim is to deliver the food to the correct table tirelessly and efficiently without causing any damage. It is able to detect and avoid the obstacle that comes in its path. The robot is designed to follow the line on the plane surface to avoid any damage to cutlery and for easy operations.

Chapter 2

Literature Review

For our project, i reviewed several robots that are used for serving food and are already available in market. Names of some robots are given below.

1. Multan Pizzeria
2. LaLuchay Robotina
3. OriHime-D
4. Delivery RobotT2

2.1 Multan Pizzeria Robot

It is serving robot and it resembles slender woman wearing a long dress and apron. It is designed for following purpose:

- Designed to navigate to customer's table.
- It is used for serving purpose in restaurant.
- It detects obstacles in path and avoid them.

Figure 2. 1 Multan Pizzeria

2.1.1 Main Features

- Weight 25 kilograms
- Power autonomy: 12-13 hours
- Navigation System
- Obstacle detecting mechanism.
- Battery Type: Lithium-Ion Rechargeable Battery
- Battery lifespan: Up to 5-10 years
- Mechanical and Electrical components used in robot are completely made in Pakistan.
- Body Type: Fiberglass
- Raspberry Pi 3

2.2 LaLuchy Robotina

It is serving robot and it works as a medical staff in East Coast Railways Central Hospital in Odisha.

It is designed for following purpose.

- Designed to serve food and medicines to COVID-19 patients while avoiding physical contact.

- The robot senses the body temperature of the patient visiting hospital and displays it on smartphone.
- It is capable of raising alarm if the temperature of patient is abnormal to alert the medical staff to attend the patient.

Figure 2. 2 LaLuchy Robotina

2.2.1 Main Features

- Height: 1.4 meter tall (4.6 feet)
- Weight: 20kilograms
- Camera
- Tablet is used as Display Screen.
- Temperature Sensors
- Battery Type: Lithium-Ion Rechargeable Battery
- Battery lifespan: Up to 2-5 years
- Computer Vision
- Raspberry Pi 3

2.3 OriHime-D

It is avatar robot for disabled people. It can be controlled by internet.

It is designed for following purpose.

- Leading guests to a meeting room
- Responding to inquiries
- Customer services and Serving

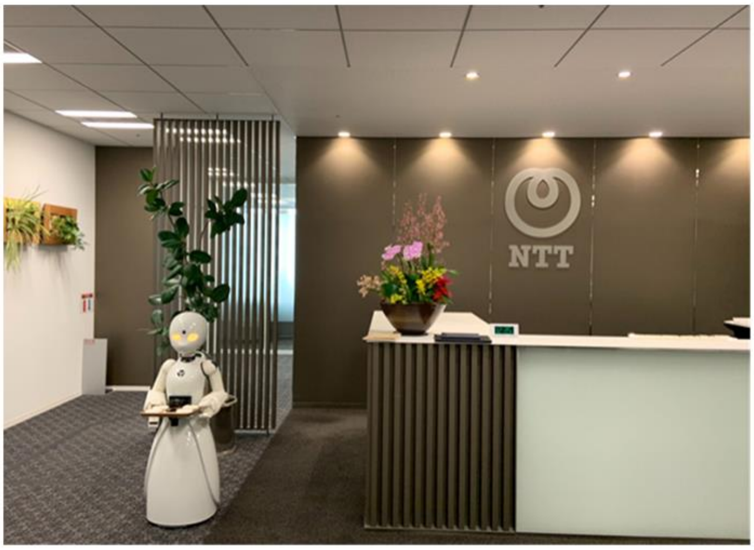


Figure 2. 3 OriHime-D

2.3.1 Main Features

- Height: 1.2 meter tall
- Weight: 25kilograms
- Camera
- Microphone
- Speakers
- Battery Type: Lithium-Ion Rechargeable Battery

- Battery lifespan: Up to 3-7 years
- Computer Vision
- 14 built-in motors
- Obstacle detectors
- V-Sido

2.4 Delivery RobotT2

Delivery robotT2 can be called as model of strength among all the delivery robots. It allows uninterrupted services and a better experience.
It is designed for following purpose.

- It is designed for large food capacity i.e. it can bear up to 50 kg food.
- Automatic cleaning of the inner cabin, it could achieve zero contact.
- Multi-layer adjustable oversized pallets can flexibly meet different distribution needs.
- Waterproof inner cabin.

Figure 2. 4 Delivery RobotT2

2.4.1 Main Features

- Size 590*470*1315mm
- Rated power 50W.
- Pallet size 570*420*253(Three Floors）
- Stand-by Current ＜0.5A
- Product Weight 75kg
- Battery life Continuous work more than 12-14h
- Load The bottom layer is 20kg, and the other single layer is 10kg.
- Standby time ＞48h
- Maximum walking speed 1m/s
- Operating life 20000h
- Maximum climbing angle ≤5°
- Network Interface Wi-Fi/4G/Bluetooth
- Battery Capacity DC 48V 12Ah
- Charging mode Automatic / manual

Chapter 3

Service Robot Applications

3.1 Advantages of Service Robots

Following are few main advantages of serving robots.

3.1.1 Safety

If i delegate a dangerous task to robot, i will have to just think of a repair bill rather than a medical bill or lawsuit. Employees who work on these dangerous tasks will be safe.

3.1.2 Speed

A robot is a machine so, it will not get tired and run slowly rather it can work all the time and speed up the production.

3.1.3 Consistency

A robot's attention cannot be diverted, and their work is not dependent on other people's work. They only do what they are supposed to do. Automation is more reliable than human labor.

3.1.4 Perfection

Robots are less likely to make mistakes that makes them perfect.

3.1.5 Productivity

Robots can not to everything. Some things can only be done by humans. If humans are not busy in doing the things that robots can easily do, humans can be more productive.

3.2 Applications

Following are the applications of service robots.

1. Restaurants.

2. Office.
3. Medical.
4. Industries.

3.2.1 Restaurants

Robot restaurants is the future of food industry. The trend of robot restaurants is getting common in the developed countries. Robots are performing multiple tasks in multiple restaurants like flipping burgers and pouring perfect cup of coffee.

Although robot restaurants are expensive, but it has its own perk and fascination. These restaurants have different kind of challenges, and they have to find the solution and work around for maximizing the benefits.

Figure 3. 1 Robot serving at Restaurant.

Some of the advanced robots are making food in the restaurants along with serving the food to the table. After the customer leaves, they collect empty dishes and take them back to the kitchen.

Figure 3. 2 Robot Making Food

3.2.2 Office

Similar to the restaurant industry the requirement of service robots in the field of official work can be made possible by giving it scenario-based tasks. Office bot can transport files from one table to another. Advantage of having robots in the workplace is that they are much cheaper as compared to humans and price of robots is further decreasing day by day. Robots are more accurate than humans because they do not shake or tremble like human hands. Robots have better performance and efficiency than humans. Moreover, robots can be designed according to the requirements such as shape, size, work needed etc. Robots can work even in hot and humid weather where humans cannot work properly. Robots can also serve as a tea-bot in the office environment.

Figure 3. 3 Robot at Office Environment

Figure 3. 4 Transporting files

3.2.3 Medical

With the growing world the use of AI has opened doors for many inventions and innovations. The use of robots in the field of medical is one of them. In the worst disease of the century known as COVID-19 robots served a great cause in the hospitals filling the gap between patients and healthcare staff. These robots are used to disinfect patient rooms and surgery suites. Since they were invented, their demand has increased vastly. Best example of this is UVD robots that came out in 2014. These robots were specialized to kill 99.99% viruses and bacteria on any surface. Demand of these robots has gone up by 1200% this year.

Figure 3. 5 Robot at hospital

Moreover, these robots can be designed keeping some tasks in mind. Robots have been designed as such to give medicines to the patients or to keep the patients in touch with the medical staff in the time of pandemic.

Figure 3. 6 Robot giving medicine.

Figure 3. 7 Communication via robots

3.2.4 Industries

Nowadays Warehouse robots are a major part of automotive industry, especially in the automotive spare parts section. These automotive companies invest in robots to easily handle and place the spare parts that are too heavy for a single person, and these robots also increase the speed of delivery and productivity of spare parts. Robots share the load of the workers by helping then out in the loading and unloading part including transportation of goods within the industry. Thus, they speed up the pace of work and do the work more efficiently.

Robots are more accurate, reduce order processing time, highly efficient and related order processing costs. That's why it is easy to say warehouse wants to automate technology which is very helpful for decreasing the load of human employees and helps them to perform task more accurately and efficiently. Automation in the field of warehouse is much more suitable and profitable. According to reports more than half of the budget, 65% to be precise, is given to the labors. So, this is the basic purpose of automation of warehouses.

Figure 3. 8 industrial Robots

Figure 3. 9 Robot transporting items in ware-house

Chapter 4

Robot Specifications & Components Selection

This delivery robot is designed for following purposes.

- Designed to navigate to customer's table.
- It is used for serving purpose in restaurant.
- It detects obstacles in path and avoid them.

Specifications

- Product Weight 8kg
- Dimensions 22×14×12 inches
- Battery life Continuous work more than 5-6h
- Load It can carry 3 kg
- Standby time ＞48h
- Maximum speed 1m/s
- Battery Operating life 3-5 years
- Maximum climbing angle ≤2°
- Network Interface Wi-Fi
- Battery Capacity DC 12V 7Ah
- Charging mode manual.

s

4.1 Micro Processor

The processor we bought is Raspberry pi 3b+. It is small in size normally equal to size of a credit card. It is a computer that can run Linux using its Arm processor. Our raspberry pi has 1gb SDRAM, Wi-Fi, Bluetooth, Ethernet port, HDMI output, audio output and USB ports etc. Raspberry pi requires a micro-SSD card which contains a n operating system in it. Some of its features are described below.

s

Figure 4. 1 Raspberry Pi module

4.1.1 Specifications

- Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- 1GB LPDDR2 SDRAM
- 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
- Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
- Extended 40-pin GPIO header
- Full-size HDMI
- 4 USB 2.0 ports
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display.
- 4-pole stereo output and composite video port
- Micro SD port for loading your operating system and storing data.
- 5V/2.5A DC power input
- Power-over-Ethernet (PoE) support (requires separate PoE HAT)

4.2 Camera

The camera i used is a Pi cam. As i have used a raspberry pi so, i used a Pi cam for easy connectivity of the camera. The pi cam has the ability to capture good photos and videos. It is light in weight and compact in size that makes it a better choice. I used a 8MP pi cam for our project.

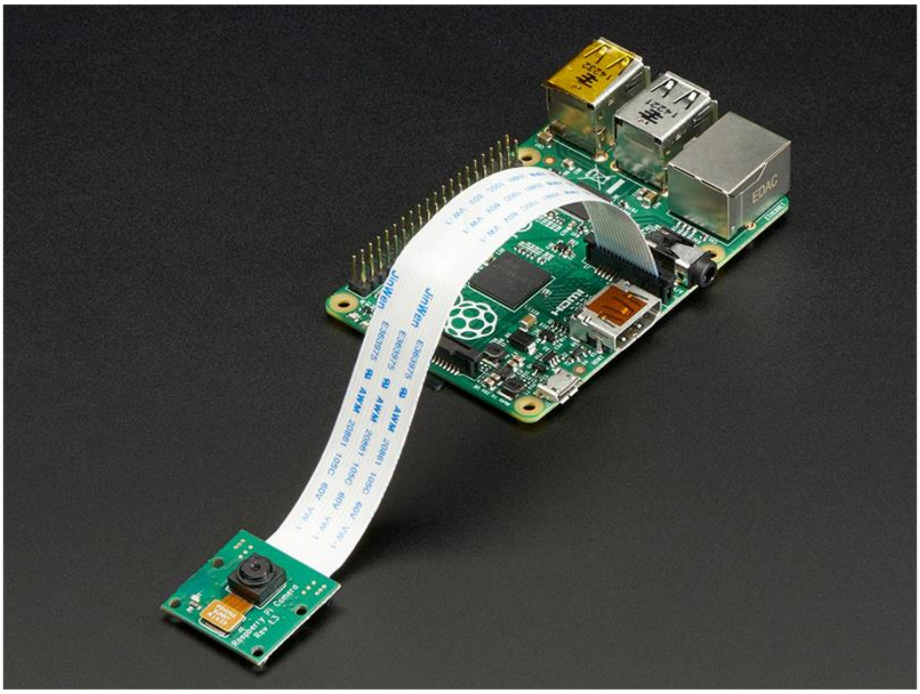


Figure 4. 2 Pi-cam

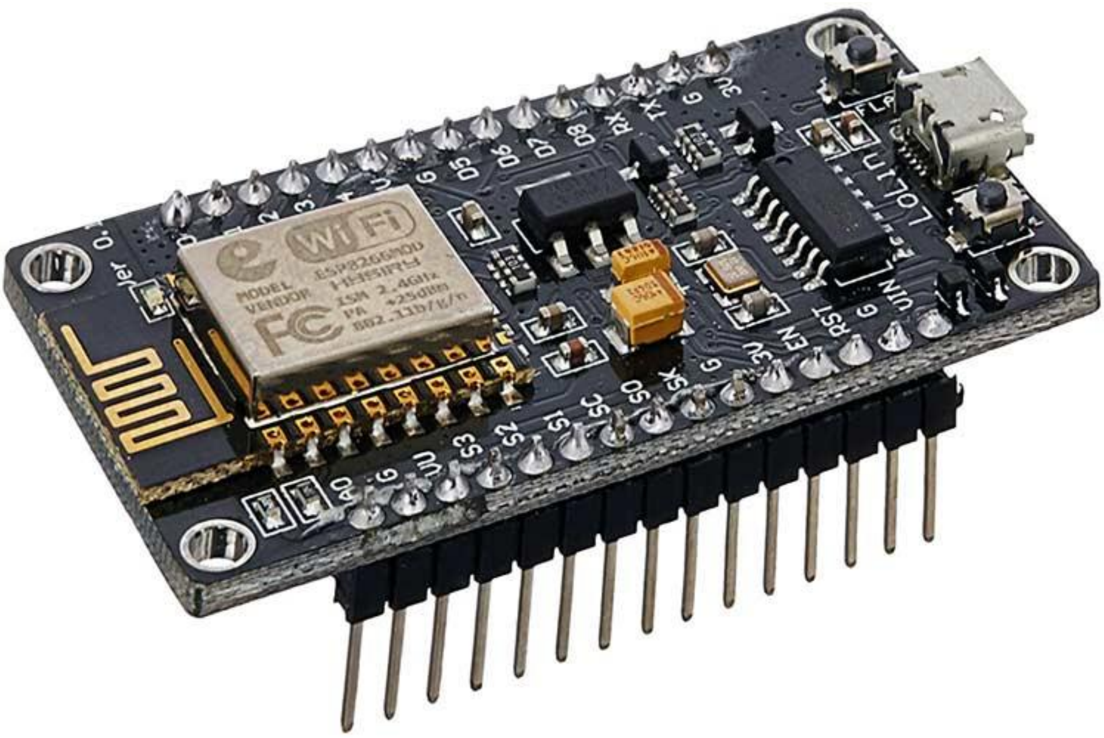
4.2.1 Specifications

- Camera module 8 Megapixel based around Sony IMX219 image sensor
- Photo Resolution 3280 x 2464Pixels
- Video 1080p30, 720p60 and 640x480p90
- Size 25mm x 23mm x 9mm
- Weight 3 grams
- Compatibility Raspberry Pi Model A, Model B and Model B + Raspberry Pi

4.3Micro Controller

The micro controller i choose is. **Node MCU ESP8266.** The ESP8266 is a low-cost Wi-Fi microchip, with built-in TCP/IP networking software, and microcontroller capability. This controls the motion of robot as it gets input from the IR sensors and gives command to the motor driver. It connects to firebase via Wi-Fi.

Figure 4. 3 . Node MCU ESP8266



4.3.1 Specifications

- Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
- Operating Voltage: 3.3V
- Input Voltage: 7-12V
- Digital I/O Pins (DIO): 16
- Analog Input Pins (ADC): 1
- UARTs: 1
- SPIs: 1
- I2Cs: 1
- Flash Memory: 4 MB
- SRAM: 64 KB
- Clock Speed: 80 MHz

- USB-TTL based on CP2102 is included onboard, Enabling Plug n Play
- PCB Antenna
- Small Sized module to fit smartly inside your IoT projects

4.4 Battery

There are basic two considerations one is led acid battery and the other is Lippo Battery. I chose Led acid battery due to cost constraints. The Led Acid battery is still the cheapest option available in the market and is beneficial for high capacity as well. Led acid batteries do not require any maintenance activity for years and can undergo a lot of charge and discharge cycles. These batteries can operate until their discharge capacity goes down by 30%. These batteries are widely available in low cost. And are easy to charge. These batteries are also known as workhouse batteries of our industries. They are used in a wide range of machinery, UPS, robotics and other systems which require a lot more power. They are available at a different range of voltages and the most commonly used among them are 2-24V.

Figure 4. 4 Lead Acid Battery

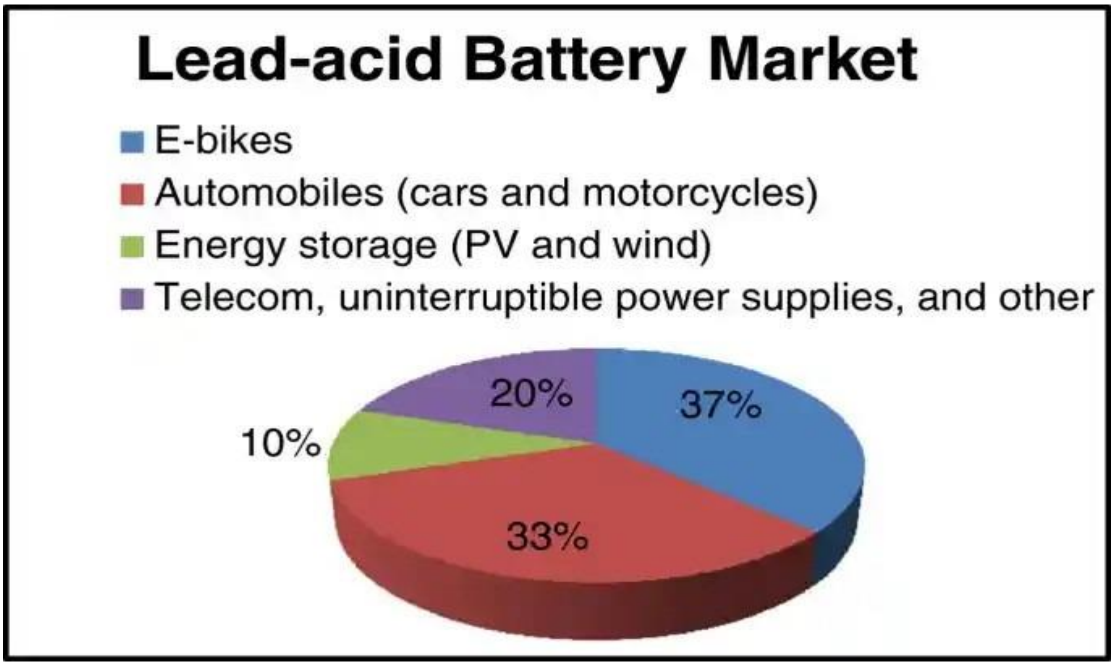


Figure 4. 5 Market of Lead Acid Battery

4.4.1 Specifications

- Dimensions: 151mm x 65 mm x 93 mm
- Voltage: 12V

- Built-in PCM to prevent overcharging.
- Electric charge capacity: 7000mAh
- Weight: 2.6kg
- Cost (subject to change): 2000 pkr/piece

4.5 Dc Motor

Selecting the drive motors is one of the most important part in making a robot. Before selecting the drive motors, you must know about the characteristics that you need for your robot. I selected the dc gear motors for this purpose. The specifications are given below which i got with the help of drive motor sizing tool.

4.5.1 Selection of motors

Certain parameters like mass of robot, Number of motors, desired operating time and radius of wheel etc. were given as input to the tool and i got the output for each motor. The calculations that are given by the drive motor sizing tool are shown in the figures below. The first picture shows the input parameters that were given to the tool.

The second picture shows the output that the tool gives according to the input parameter.

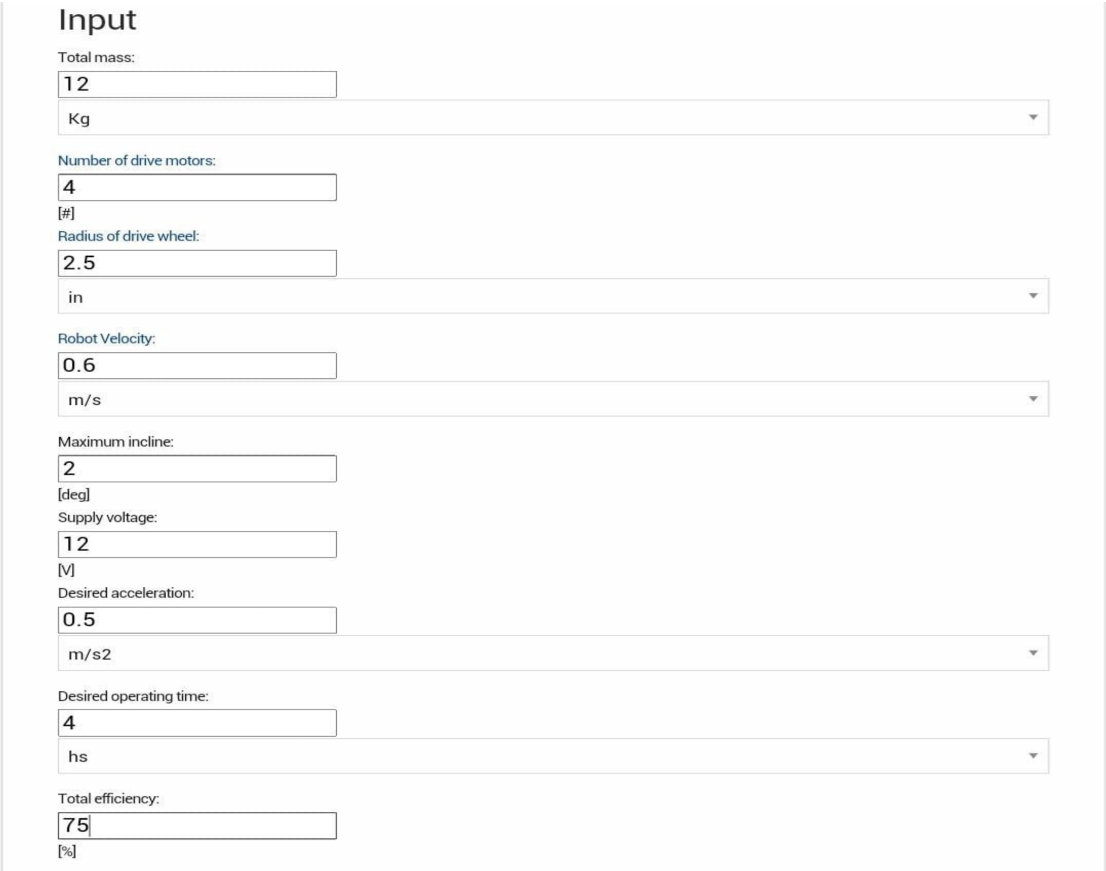


Figure 4. 6 input parameter for sizing tool

Based on the input parameters the tool gave us the required specifications for the selection of our motor that include torque, power, and current etc.

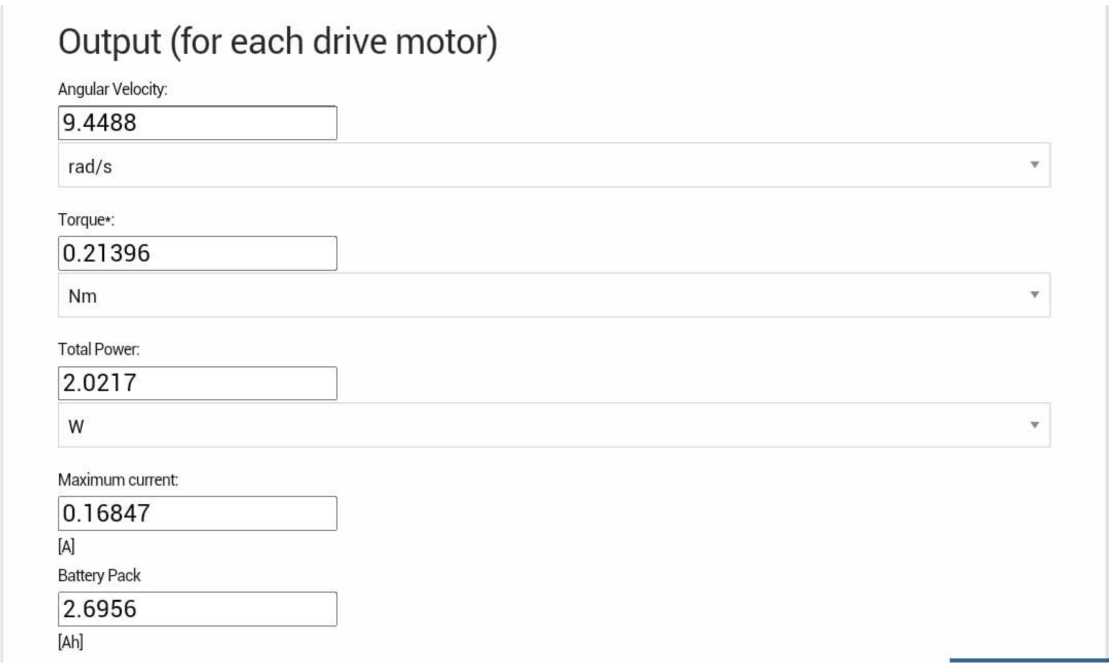


Figure 4. 7 output parameters for the sizing tool

I choose the following planetary gear brush motor and the specifications are given below.

Figure 4. 8 DC motor

4.5.2 Specifications

- 22mm diameter planetary gear brush motor
- Smooth, quiet operation
- Dual ball bearings
- Rated 142RPM @ 12Vdc, no-load. 113 RPM with load
- 6mm diameter x 14mm shaft with hole. 2mm diameter shaft at rear of motor
- Overall length of motor and gearbox, 60mm
- Solder lug terminals

4.6 Motor Driver

A moto driver is device that controls the motor. It may start or stop the motor manually or automatically. It controls its forward and reverse rotation. In addition to that it regulates the speed and limits the torque. It also protects the motors from overloading and electrical faults. I had two choices for the motor driver. One was L298N and the other was IBT2.

4.6.1 Comparison and Selection of Motor driver

The L298N motor driver is a high-power driver that is used to run DC and stepper motors. It consists of a L298 IC and a 7BMOS 5V regulator. It can control DC motors up to a no of 4, but it can control the speed or direction of 2 DC motors at the time.

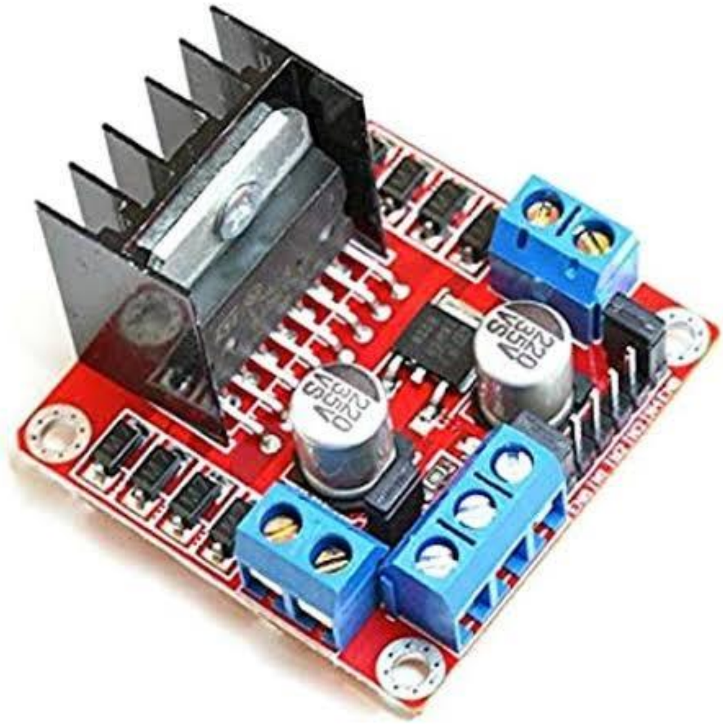


Figure 4. 9 L298N module

4.6.1.1 Specifications

- Input voltage   5-35V
- Output Current   2A per channel
- Control logic   standard 5V TTL
- Logical current   36 mA
- Power   25 W
- Size   55mm* 60mm*30mm
- Weight 33g
- Price 300 Pkr

The second motor driver is IBT2. It uses high power driver chip BTS7960 that is made up of a H bridge driver module. It has strong drive and break effects. It can bear up to 43A current. It has the capability to protect the motors in over temperature, over and under voltage, over current and short circuit.

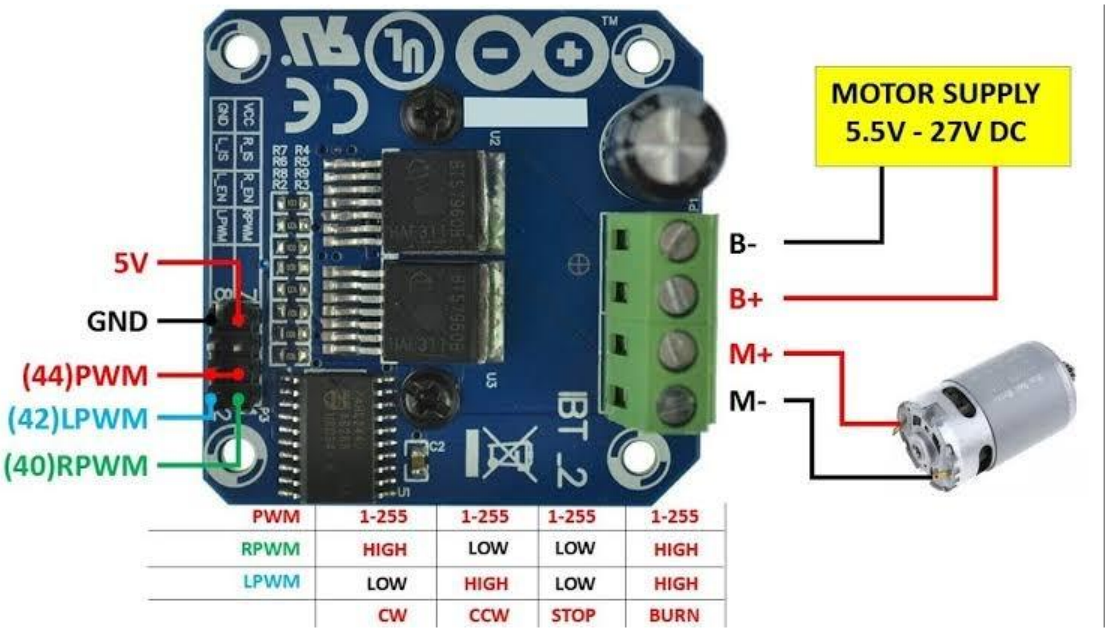


Figure 4. 10 IBT2 module

4.6.1.2 Specifications

- 100% brand new and high quality
- Quantity: 1pc
- Weight: 66 g
- Input voltage: 6V-27V
- Maximum Current: 43A
- Input level: 3.3V-5V
- Control mode: PWM or level
- Duty cycle: 0 to 100%
- Current conditioning output: yes
- Size:50mmx40mm/1.97"
- Price =1500 Pkr

I selected L298N because of our requirements. Our current rating was fulfilled by the L298N motor driver. Secondly the cost of IBT2 was very high so it was not the preferred option. L298N can control up to 2 motors so, 2 drivers were needed. On the other hand, i had to use four IBT2 drivers.

4.7 Sensors

This section is based on the different sensors that are used in the project.

4.7.1 Sonar Sensors

Nowadays distance measurement is most necessary in our daily outdoor projects. we can easily measure distance with small package of sonar sensor this is a small module which can easily measure distance 4-5 meters by ultrasound waves and surprisingly accurate. According to the study these sensors are installed at front end of the serving robot to avoid obstacles and to maintain the distance from the object. Ultrasonic sensor operates by controlling the time difference by this equation:

$$Distance\ from\ object = \frac{High\ level\ time * Speed\ of\ sound\ in\ air}{2}$$

As we can see in the figure below that a sonar sensor transmits an ultrasonic wave and after striking on the surface of object it retrieves at the receiver end. The time taken by the signal to travel from transmitter to the surface of the object is known as the time difference and it is shown in the figure below.

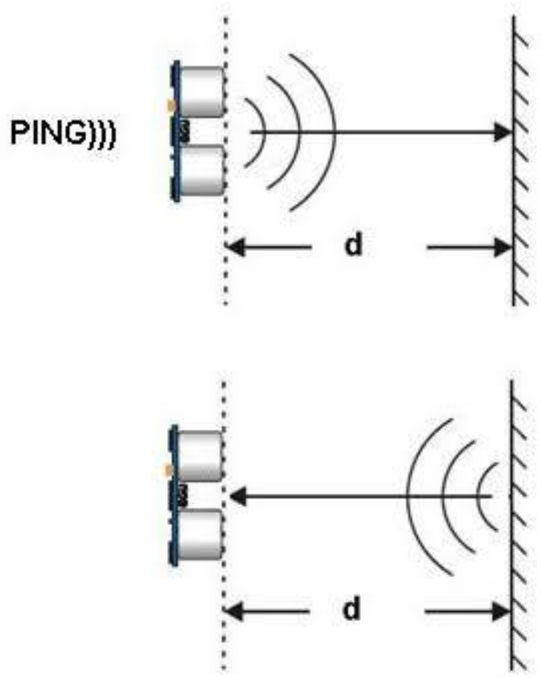


Figure 4. 11 Distance measurement

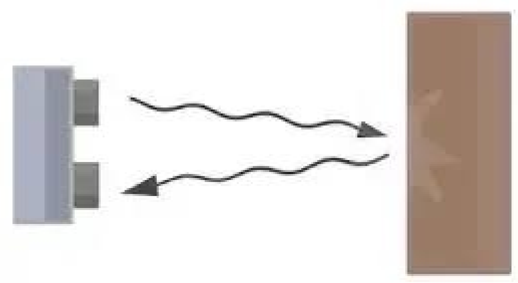


Figure 4. 12 Object detection

Object detection is done by the procedure shown in the figure above. Transmitted signal is returned after striking the surface of object.

4.7.2 IR Sensors

Infrared transmitter and receiver help us to detect an object. IR transmitter sends an infrared signal with a frequency that is compatible with IR receiver which performs the detection part. The signals that are transmitted by the IR transmitter are invisible to the naked eye. IR sensors help us to follow the line using the above-mentioned process. IR sensors detect the black line by emitting the IR light and detecting the returning light levels.

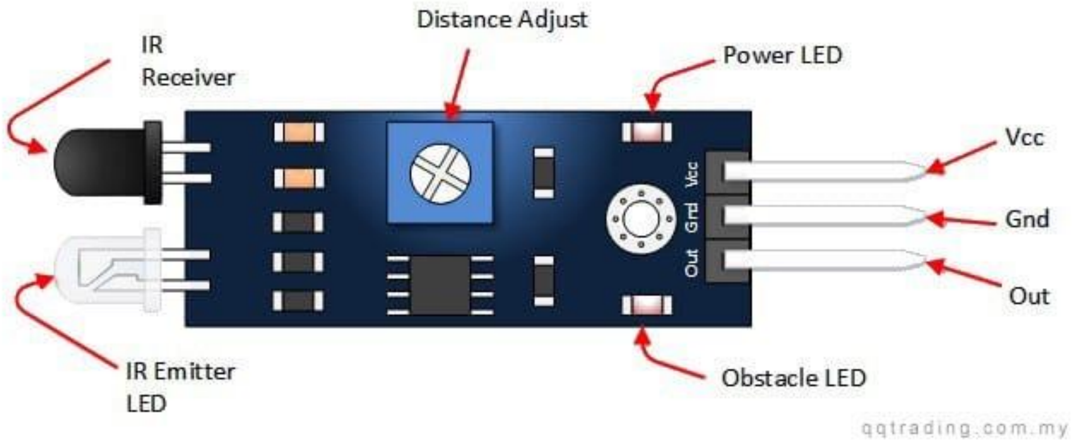


Figure 4. 13 IR sensor

Chapter 5

Hardware Architecture and working

5.1 Flow chart / Block diagram:

The block diagram of our project shows the electronic components and their placement or connection with the other components.

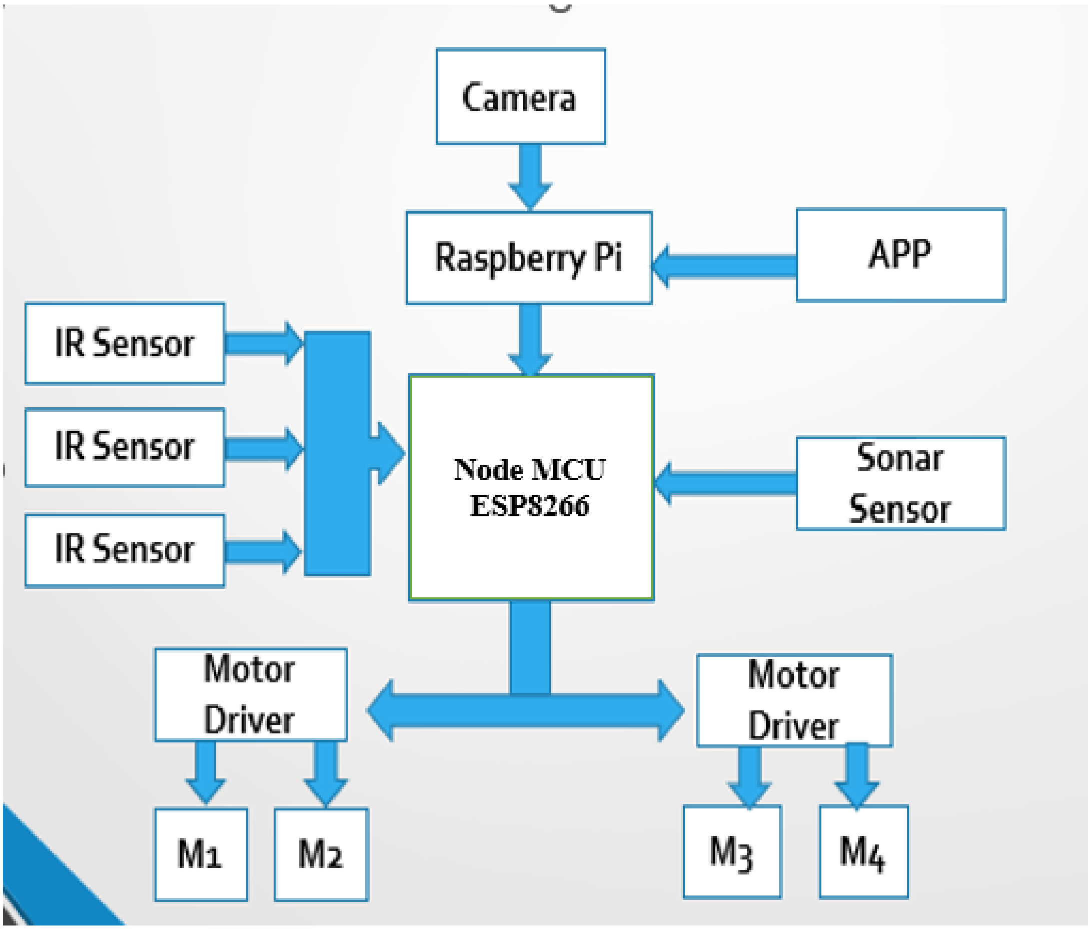


Figure 5. 1 Block diagram

5.1.1 Explanation of the block diagram

Block diagram of the project shows the connectivity between pi cam and the raspberry pi. Raspberry pi is also connected to the app via firebase. The app sends data about the order and the table number to raspberry pi. Then, IR sensors, Sonar sensor are connected with ESP8266 in Figure 5.1. First of all, IR sensor detects the line and makes the robot to follow the line. Simultaneously the Sonar sensor starts working and detects the obstacle in its path by using the obstacle detection method. Then we can see that Pi-cam is installed on Raspberry pi. Pi-cam helps to detect and scan the table number on which the food is to be delivered. Then we can see that Raspberry pi is connected with the ESP8266 which is further connected with the motor drivers these motor drivers control the motors. Raspberry pi sends the signal to the ESP8266 based on the information provided by the pi-cam. Now, the motor driver controls the motor speed and left right rotation according to the information that is provided to reach the destination.

5.1.2 Raspberry Pi to ESP8266 Communication for line tracking

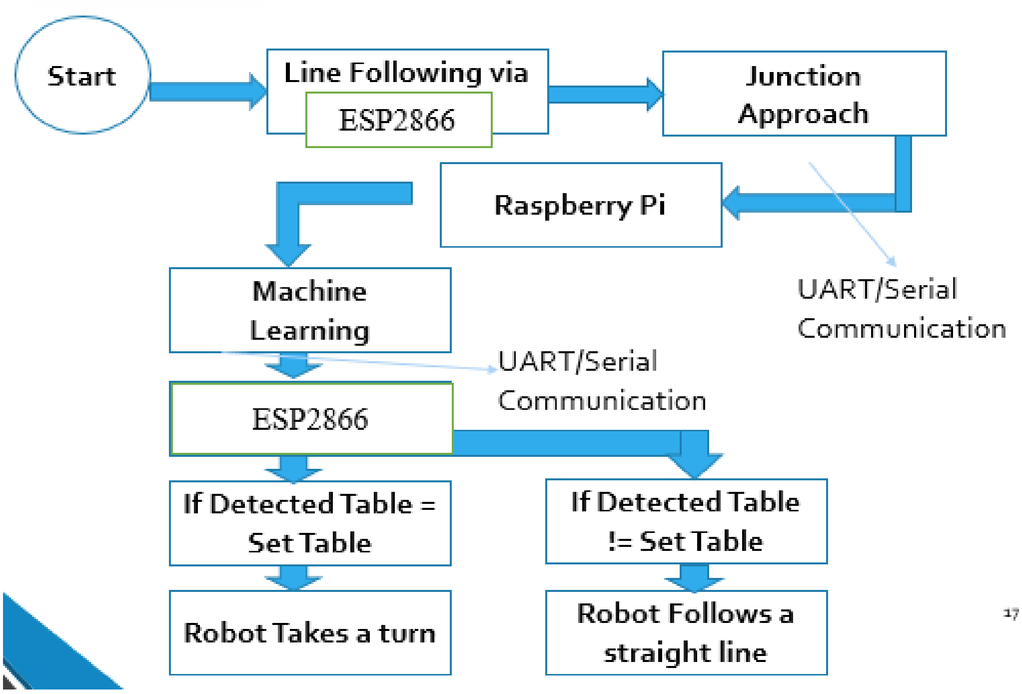


Figure 5. 2 Raspberry pi to ESP8266 communication

ESP8266 controls the line following feature of our robot and as the robot approaches junction then the using UART (serial Communication) method communicates with Raspberry Pi that the robot is near to the junction. Raspberry will then compare the captured picture with the train data and Raspberry pi will send this information to ESP8266 with serial communication. The ESP8266 then take decision if the detected table is the desired table, then it will take a turn move to the desired table else it will continuous on straight line.

5.2 Environment Setup

As we know the restaurants are crowded and kids are playing and running here and there and there is no such discipline, so it is really difficult for waiters to deliver food on time and sometimes food falls down due to collision with any of the customer walking in the way and everyone is in hurry. As far as we are concerned about the robot is that if it has to deliver food in this crowded environment it will have to face many obstacles in its path and customers will also face difficulty in walking which could result into collision and robot will have to change its path many times which will increase time of delivery and in other case if suddenly some comes in front of robot or fells on the robot, it will result in damage to cutlery and robot itself too. So, we setup specific environment for robot to travel in the restaurant without facing any interpretation. I will create the several junctions on floor along the line that it will be following as per number of tables in each restaurant. Each junction will be numbered along the line starting from 1 and so on. In case of our robot, i are considering 3 junctions along the line starting from 1 to 4 and robot will follow the line and detect it and deliver the order as per provided instructions.

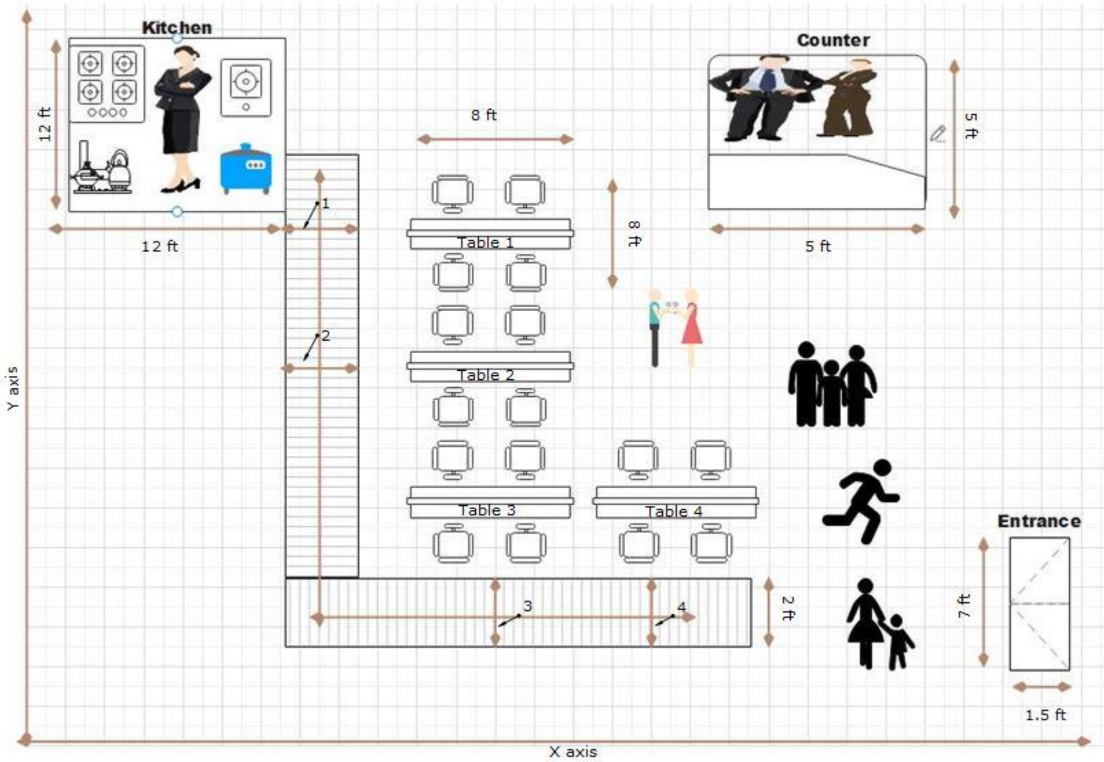


Figure 5. 3 Environment setup in a hotel environment

The above figure describes our setup that i have thought as a solution to our problem. There are 4 tables as represented by the 3 boxes along the main block. Each table has got a fixed number and the number allotted to that table is permanently written on the setup. As you can see that the setup starts with the kitchen. The order is placed at the top of the robot in the kitchen and allotted a table to deliver the order. The robot then delivers the order at the table and moves back to kitchen. If the charging of the robot is down the robot will then be taken to its charging area and put on charging. This is the complete environment setup that describes how this project works.

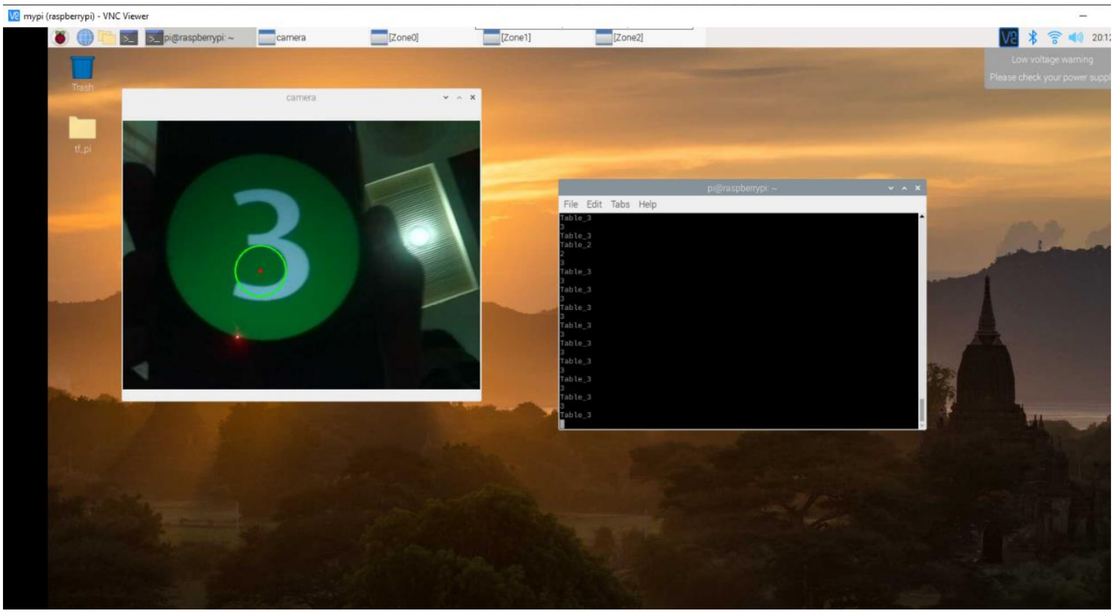


Figure 5. 4 Table no. 3 Machine leaning part

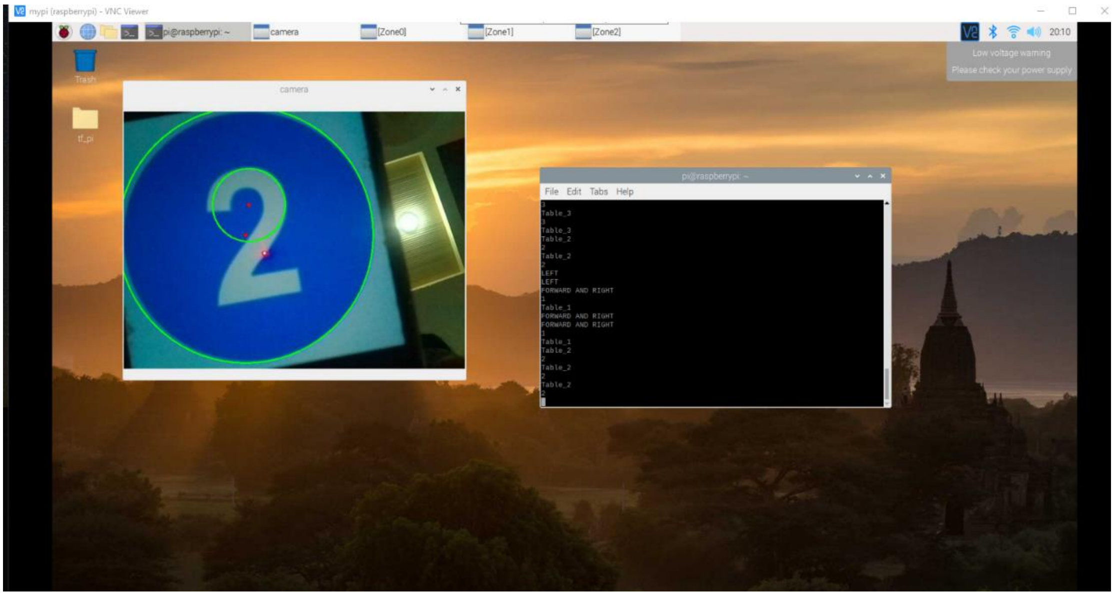


Figure 5. 5 Table no. 2 Machine learning part

5.3 Line Following

In the project the Line following is used for the movement of robot. Working of line tracking is pretty simple. Robot follows a black line that is on the floor. On its way the robot detects the black line on the ground and move along with that line. This method works by detecting the difference between the line you want to follow and the adjacent surface of ground. As it can be seen in the block diagram that we have used three IR sensors that will help us in detecting the line. These IR sensors consists of IR led and photodiode .IR sensors are attached at the front face of the robot. IR led emits a light signal. When the IR sensor comes into reflected surface the light emitted by IR led is detected by photodiode and sends the information to ESP8266. ESP8266 continuously monitors data from all 3 IR sensors and commands the robot how to move in left or right direction as per line detected by them.

Figure 5. 6 Line tracking

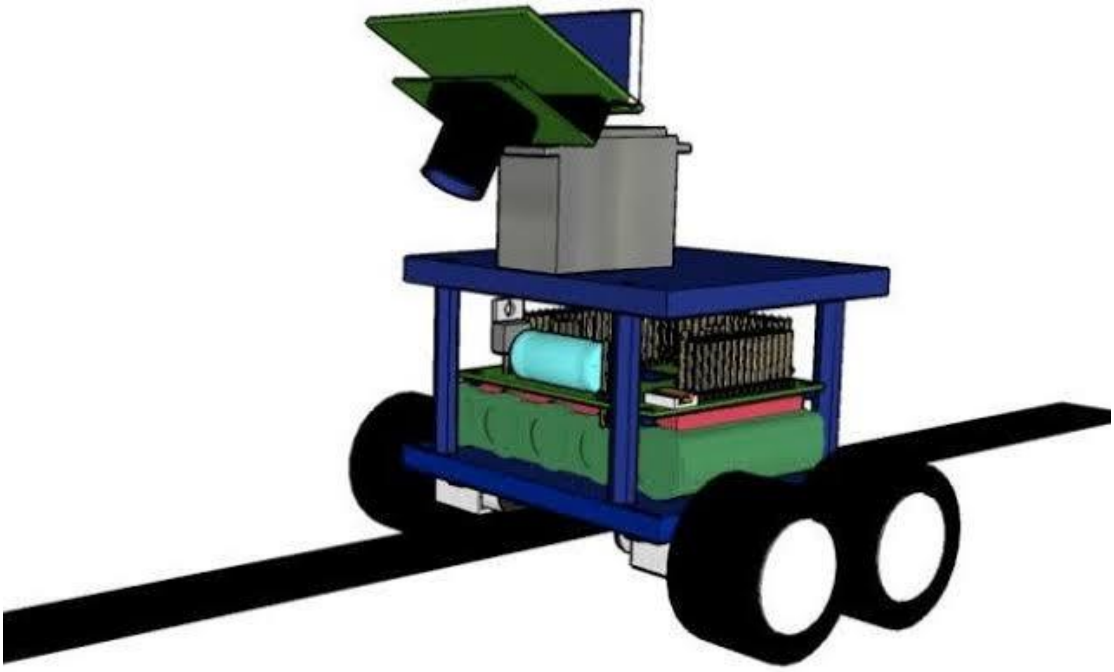


Figure 5. 7 Robot following line.

5.4 Mobile App:

5.4.1 Mobile App Block diagram:

The block diagram for mobile application is given below:

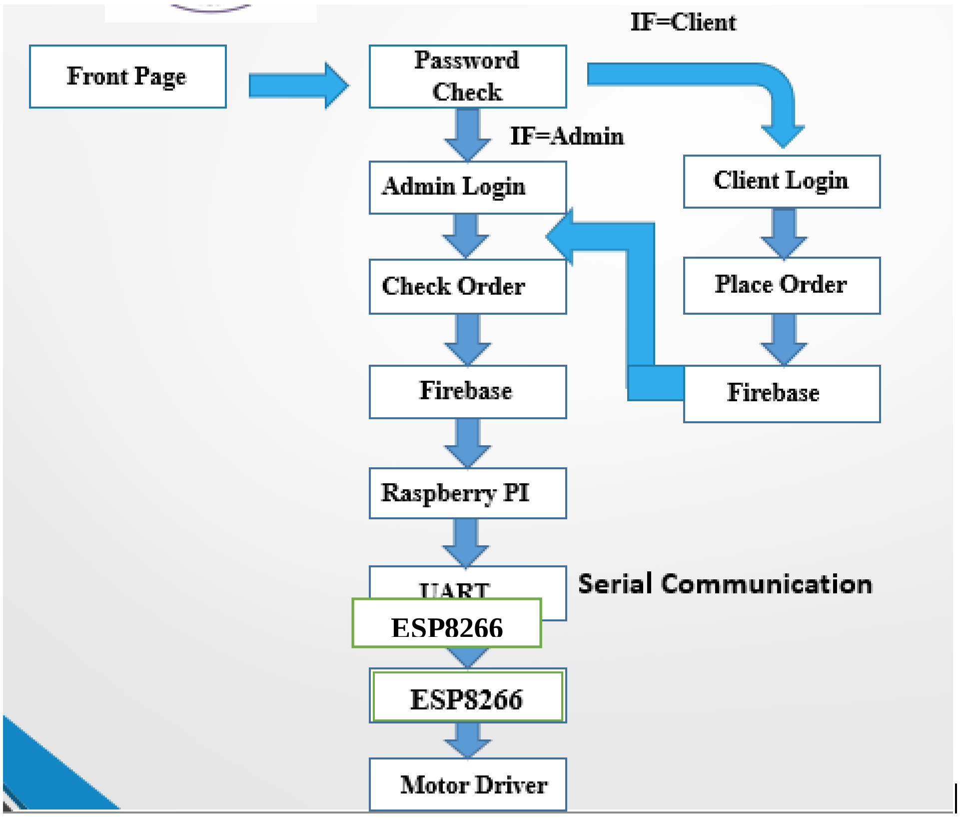


Figure 5. 8 App block Diagram

5.4.2 Mobile App Explanation:

The mobile application has been developed through MIT app Inventor. The mobile application is simple and easy to access; it has to different Interfaces one is for client and other is for admin. The mobile application is simple and easy to access; the user just need to connect to the Wi-Fi network and then app updated its self automatically from firebase with latest readings. Client can place the order while sitting at their place and admin can check the order.

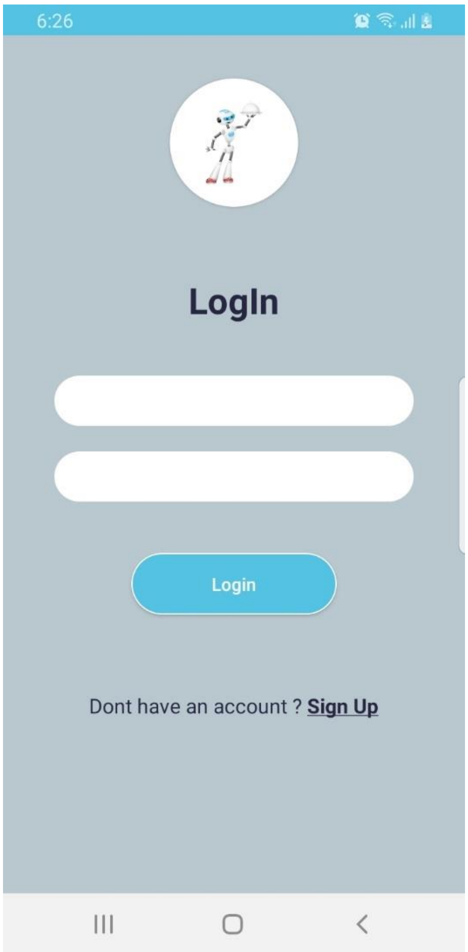


Figure 5. 9 Login interface

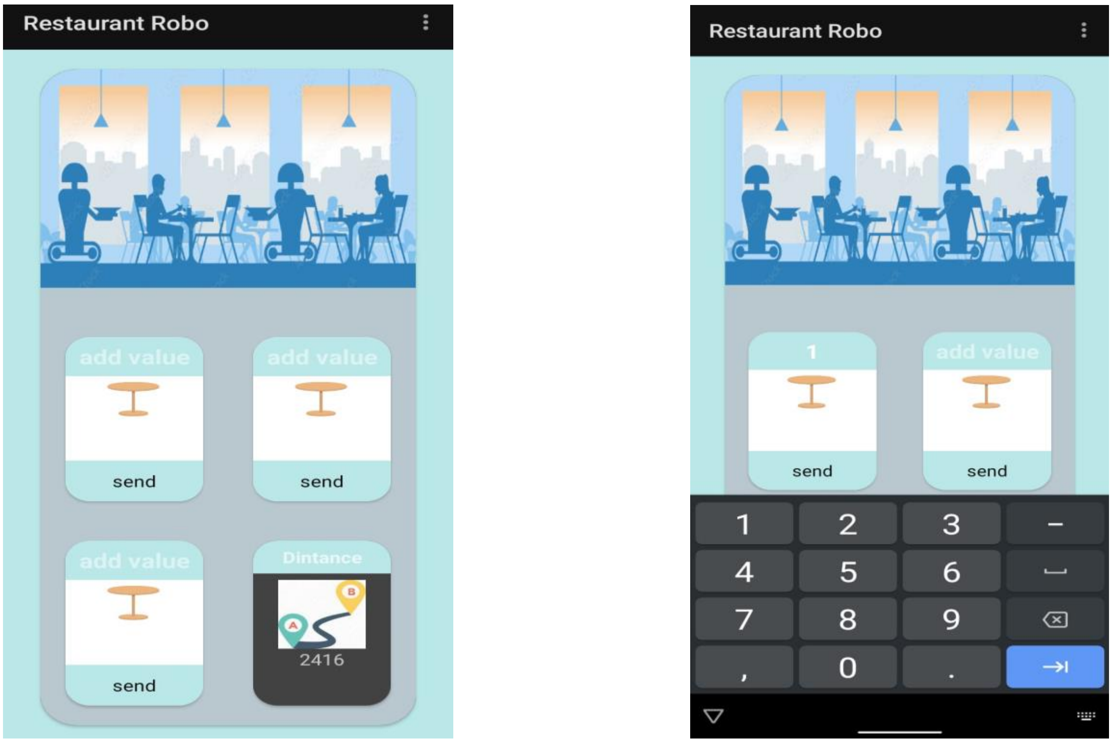


Figure 5. 10 Place order Screen

Chapter 6

6.1 Code

ESP 8266 code:

```
else if(digitalRead(IR1)==HIGH && digitalRead(IR2)==LOW)
{
  //Tilt robot towards right by stopping the right wheel and moving the left one
  analogWrite(D3,0);analogWrite(D4,150);


}
else if(digitalRead(IR1)==LOW && digitalRead(IR2)==LOW)  //IR not on black line
{
 digitalWrite(D3,LOW);
 digitalWrite(D4,LOW);
}

else
{
 digitalWrite(D3,LOW);
 digitalWrite(D4,LOW);
}

}



if(Table1==0)
{
 analogWrite(D3,0);
  analogWrite(D4,0);

}
 if(sonar==0)
{
 analogWrite(D3,0);
  analogWrite(D4,0);

}
}
```

```
#include "FirebaseESP8266.h"
#include <Wire.h>
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include "ESP8266WebServer.h"
int IR1=D1;      //Right sensor
int IR2=D2;    //left Sensor

 FirebaseData firebaseData;
 FirebaseData firebaseData1;

#define FIREBASE_HOST "restaurant-robo-web-default-rtdb.firebaseio.com"
#define FIREBASE_AUTH "5cQfMQ0mF9LqVoAtX1tMExQ5RFmtmHzmEMMCNqLO"

#define WIFI_SSID       "mywifi"
#define WIFI_PASSWORD   "123456789"

void setup()
{
 pinMode(IR1,INPUT);
  pinMode(IR2,INPUT);

  Serial.begin(115200);
  analogWrite(D3,0);
    analogWrite(D4,0);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
```

```
 }
 void loop()
 {
Firebase.get(firebaseData, "/Data/table1");
Firebase.get(firebaseData1, "/Data/sonar");

int Table1=firebaseData.intData();
int sonar=firebaseData1.intData();

if((Table1==1)&&(sonar==6))
{
  if(digitalRead(IR1)==HIGH && digitalRead(IR2)==HIGH) //IR will not glow on black line
 {
   analogWrite(D3,100);analogWrite(D4,100);

 }

 else if(digitalRead(IR1)==LOW && digitalRead(IR2)==LOW)  //IR not on black line
 {
  analogWrite(D3,0);analogWrite(D4,0);
 }

 else if(digitalRead(IR1)==LOW && digitalRead(IR2)==HIGH)
 {
  analogWrite(D3,150);analogWrite(D4,0);
}

 else if(digitalRead(IR1)==HIGH && digitalRead(IR2)==LOW)
 {
   //Tilt robot towards right by stopping the right wheel and moving the left one
   analogWrite(D3,0);analogWrite(D4,150);


 }
```

Pi cam (Raspberry Pi):

```
import cv2
import numpy as np
from scipy.stats import itemfreq
import pyrebase
config = {
  "apiKey": "AIzaSyBExQylwCIVyZAW57YGT_UMJWRU-neLFXg",
  "authDomain": "restaurant-robo-dc20c",
  "databaseURL": "https://restaurant-robo-dc20c-default-rtdb.firebaseio.com/",
  "storageBucket":"project-334589597167"
}
firebase = pyrebase.initialize_app(config)
database = firebase.database()
ProjectBucket = database.child("Data")
ProjectBucket.child("table_1").set(1)

def get_dominant_color(image, n_colors):
    pixels = np.float32(image).reshape((-1, 3))
    criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 200, .1)
    flags = cv2.KMEANS_RANDOM_CENTERS
    flags, labels, centroids = cv2.kmeans(
        pixels, n_colors, None, criteria, 10, flags)
    palette = np.uint8(centroids)
    return palette[np.argmax(itemfreq(labels)[:, -1])]


clicked = False
def onMouse(event, x, y, flags, param):
    global clicked
    if event == cv2.EVENT_LBUTTONUP:
        clicked = True
cameraCapture = cv2.VideoCapture(0)
cv2.namedWindow('camera')
cv2.setMouseCallback('camera', onMouse)
```

```python
cv2.setMouseCallback('camera', onMouse)
# Read and process frames in loop
success, frame = cameraCapture.read()
while success and not clicked:
    cv2.waitKey(1)
    success, frame = cameraCapture.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    img = cv2.medianBlur(gray, 37)
    circles = cv2.HoughCircles(img, cv2.HOUGH_GRADIENT,
                               1, 50, param1=120, param2=40)

    if not circles is None:
        circles = np.uint16(np.around(circles))
        max_r, max_i = 0, 0
        for i in range(len(circles[:, :, 2][0])):
            if circles[:, :, 2][0][i] > 50 and circles[:, :, 2][0][i] > max_r:
                max_i = i
                max_r = circles[:, :, 2][0][i]
        x, y, r = circles[:, :, :][0][max_i]
        if y > r and x > r:
            square = frame[y-r:y+r, x-r:x+r]
            dominant_color = get_dominant_color(square, 2)
            if dominant_color[2] > 100:
                print("1")
                database = firebase.database()
                ProjectBucket = database.child("Data")
                ProjectBucket.child("raspi").set(1)
                print("Table_1");

            elif dominant_color[0] > 80:
                zone_0 = square[square.shape[0]*3//8:square.shape[0]
                                * 5//8, square.shape[1]*1//8:square.shape[1]*3//8]
                cv2.imshow('Zone0', zone_0)
```

```python
            elif dominant_color[0] > 80:
                zone_0 = square[square.shape[0]*3//8:square.shape[0]
                                * 5//8, square.shape[1]*1//8:square.shape[1]*3//8]
                cv2.imshow('Zone0', zone_0)
                zone_0_color = get_dominant_color(zone_0, 1)

                zone_1 = square[square.shape[0]*1//8:square.shape[0]
                                * 3//8, square.shape[1]*3//8:square.shape[1]*5//8]
                cv2.imshow('Zone1', zone_1)
                zone_1_color = get_dominant_color(zone_1, 1)

                zone_2 = square[square.shape[0]*3//8:square.shape[0]
                                * 5//8, square.shape[1]*5//8:square.shape[1]*7//8]
                cv2.imshow('Zone2', zone_2)
                zone_2_color = get_dominant_color(zone_2, 1)

                if zone_1_color[2] < 60:
                    if sum(zone_0_color) > sum(zone_2_color):
                        print("LEFT")
                    else:
                        database = firebase.database()
                        ProjectBucket = database.child("Data")
                        ProjectBucket.child("raspi").set(2)
                        print("Table_2")
                        print("2")
                else:
                    if sum(zone_1_color) > sum(zone_0_color) and sum(zone_1_color) >
                        print("FORWARD")
                        print("2")
                    elif sum(zone_0_color) > sum(zone_2_color):
                        print("FORWARD AND LEFT")
                    else:
                        print("FORWARD AND RIGHT")
```

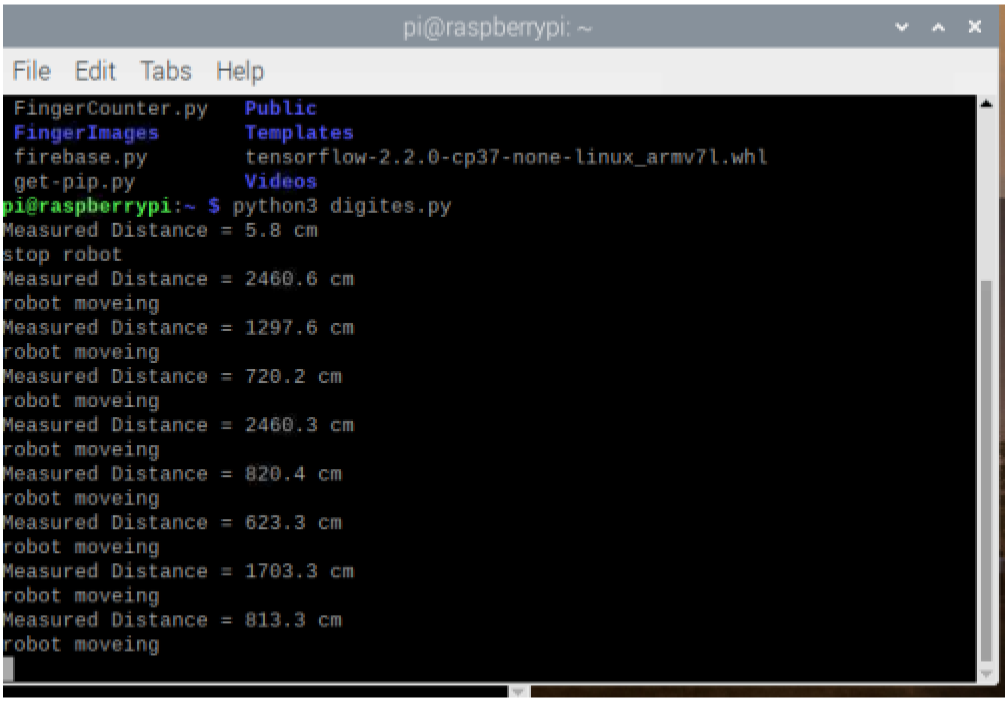
```
                        print("FORWARD AND RIGHT")
                else:
                    database = firebase.database()
                    ProjectBucket = database.child("Data")
                    ProjectBucket.child("raspi").set(3)
                    print("3")
                    print("Table_3");

            for i in circles[0, :]:
                cv2.circle(frame, (i[0], i[1]), i[2], (0, 255, 0), 2)
                cv2.circle(frame, (i[0], i[1]), 2, (0, 0, 255), 3)
        cv2.imshow('camera', frame)


cv2.destroyAllWindows()
cameraCapture.release()
```

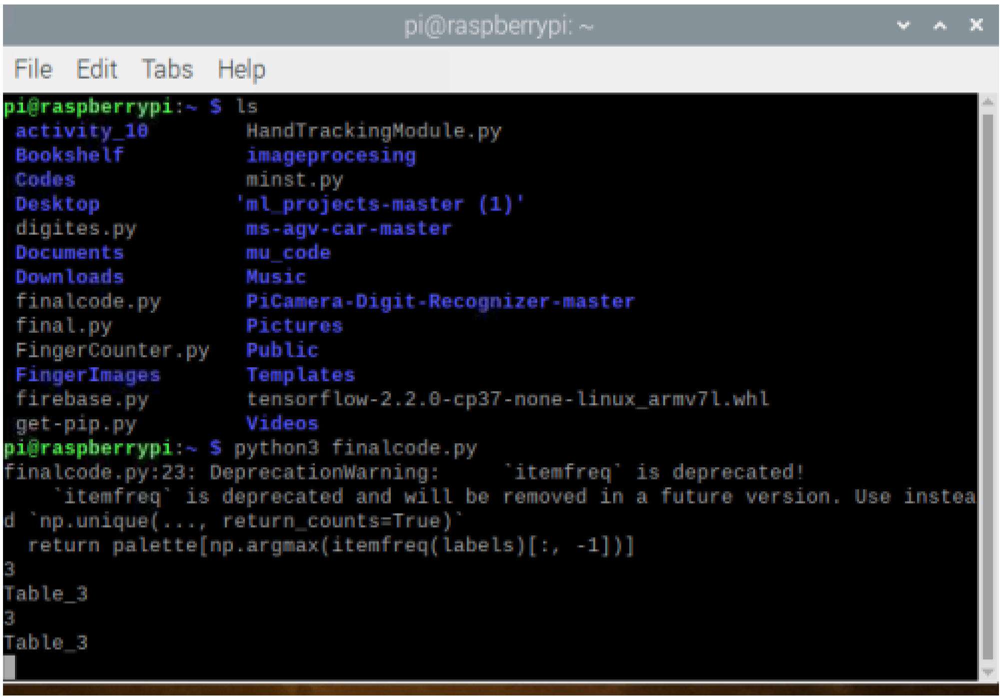
Sonar (Raspberry Pi):

```python
#Libraries
import RPi.GPIO as GPIO
import time
import pyrebase

#GPIO Mode (BOARD / BCM)
GPIO.setmode(GPIO.BCM)

#set GPIO Pins
GPIO_TRIGGER = 18
GPIO_ECHO = 24

#set GPIO direction (IN / OUT)
GPIO.setup(GPIO_TRIGGER, GPIO.OUT)
GPIO.setup(GPIO_ECHO, GPIO.IN)
config = {
  "apiKey": "AIzaSyBExQylwCIVyZAW57YGT_UMJWRU-neLFXg",
  "authDomain": "restaurant-robo-dc20c",
  "databaseURL": "https://restaurant-robo-dc20c-default-rtdb.firebaseio.com/",
  "storageBucket":"project-334589597167"
}
firebase = pyrebase.initialize_app(config)

def distance():
    # set Trigger to HIGH
    GPIO.output(GPIO_TRIGGER, True)

    # set Trigger after 0.01ms to LOW
    time.sleep(0.00001)
    GPIO.output(GPIO_TRIGGER, False)

```

```python

    StartTime = time.time()
    StopTime = time.time()

    # save StartTime
    while GPIO.input(GPIO_ECHO) == 0:
        StartTime = time.time()

    # save time of arrival
    while GPIO.input(GPIO_ECHO) == 1:
        StopTime = time.time()

    # time difference between start and arrival
    TimeElapsed = StopTime - StartTime
    # multiply with the sonic speed (34300 cm/s)
    # and divide by 2, because there and back
    distance = (TimeElapsed * 34300) / 2

    return distance

if __name__ == '__main__':
    try:
        database = firebase.database()
        ProjectBucket = database.child("Data")
        ProjectBucket.child("Table_1").set(10)
        ProjectBucket.child("Data").child("Table_2").set(10)
        ProjectBucket.child("Data").child("Table_3").set(10)
        ProjectBucket.child("Data").child("sonar").set(10)
        while True:
            dist = distance()
            print ("Measured Distance = %.1f cm" % dist)
            print ("Measured Distance = %.1f cm" % dist)
            time.sleep(1)
            if(dist>0 and dist<20):
                print("stop robot")
                ProjectBucket.child("Data").child("sonar").set(0)
                ProjectBucket.child("Data").child("Distance").set(dist)
            if(dist>30):
                print("robot moveing")
                ProjectBucket.child("Data").child("Distance").set(dist)
                ProjectBucket.child("Data").child("sonar").set(6)


# Reset by pressing CTRL + C
    except KeyboardInterrupt:
        print("Measurement stopped by User")
        GPIO.cleanup()
```

6.1 Training


pi@raspberrypi: ~
File Edit Tabs Help
FingerCounter.py Public
FingerImages Templates
firebase.py tensorflow-2.2.0-cp37-none-linux_armv7l.whl
get-pip.py Videos
pi@raspberrypi:~ $ python3 digites.py
Measured Distance = 5.8 cm
stop robot
Measured Distance = 2460.6 cm
robot moveing
Measured Distance = 1297.6 cm
robot moveing
Measured Distance = 720.2 cm
robot moveing
Measured Distance = 2460.3 cm
robot moveing
Measured Distance = 820.4 cm
robot moveing
Measured Distance = 623.3 cm
robot moveing
Measured Distance = 1703.3 cm
robot moveing
Measured Distance = 813.3 cm
robot moveing


pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ ls
activity_10 HandTrackingModule.py
Bookshelf imageprocesing
Codes minst.py
Desktop 'ml_projects-master (1)'
digites.py ms-agv-car-master
Documents mu_code
Downloads Music
finalcode.py PiCamera-Digit-Recognizer-master
final.py Pictures
FingerCounter.py Public
FingerImages Templates
firebase.py tensorflow-2.2.0-cp37-none-linux_armv7l.whl
get-pip.py Videos
pi@raspberrypi:~ $ python3 finalcode.py
finalcode.py:23: DeprecationWarning: `itemfreq` is deprecated!
`itemfreq` is deprecated and will be removed in a future version. Use instea
d `np.unique(..., return_counts=True)`
return palette[np.argmax(itemfreq(labels)[:, -1])]
3
Table_3
3
Table_3

Bibliography

Advani, V. (2021, Feb 11). *what is artificial intelligence* . Retrieved from mygreatlearning: http://mygreatlearning.com

ahmed, S. (2017, march 4). *Robot serves food at Multan pizzeria*. Retrieved from dawn news: https://www.dawn.com/news/1318249

Allen, D. W. (2018, april 24). *Brookings.edu*. Retrieved from Brooking: www.brookings.edu

carter, A. (2016, April). *The Raspberry Pi Camera Module v2 replaced the original Camera Module* . Retrieved from Camera Module V2: https://www.raspberrypi.org/products/camera-module-v2/

FRANCE-PRESSE, A. (2020, August 29). *Robot eases loneliness of Mexican virus patients*. Retrieved from rappler: https://www.rappler.com/technology/features/robot-eases-loneliness-of-mexican-virus-patients

Keennrobot. (2020). Retrieved from keenonrobot: https://www.keenonrobot.com

OryLaboratory. (2009). Retrieved from Orihime: http://ces15.orylab.com/

Ray, A. (2018). Compassionate Artificial Intelligence. *Amit Ray*, 1.

Robot, K. (2020). Retrieved from Keenon Delivery Robot T2: https://www.robotshop.com/en/keenon-delivery-robot-t2.html

Shiny.J.S. (2017). Automation of Restaurant. *Automation of Restaurant*, 8.

Wise, M. (2016). *Workshop on autonomous mobile service robots*. Retrieved from Fetch and freight: http://docs.fetchrobotics.com/