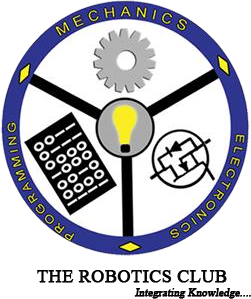
# Project Report on

# CHAUSER (SMART SHOE RACK)

***Submission to the THE ROBOTICS CLUB as a part of INDUCTION' 20***

# TEAM *12*



**SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY (AUTONOMOUS)**

**(Affiliated to JNT University, Hyderabad)**

## Yamnampet, Ghatkesar, Hyderabad – 501 301.

2020

**CERTIFICATE**

This is the project work titled ***CHAUSER (SMART SHOE RACK)*** by

1.Akhilesh

2.Gowtham Karthikeya

3Abdul Rahman

4.Ganesh

5.Vamshi

6.Yashwanth

7.Premsingh

8.Vinay kumar

is a record of the project work carried out by them during the year 2020 as part of **INDUCTION’20 .**

# M M SAI PRAKASH G DATTA LOHITH

**Technical Head Technical Head**

**T.V.HARI KRISHNA**

**The President of THE ROBOTICS CLUB**

**Dr. A. Purushotham Technical Advisor Mechanical Department**

**Declaration**

This project work reported in the present thesis titled ***CHAUSER (SMART SHOE RACK)***is a record work done by Team *12* in **THE ROBOTICS CLUB** as a part of  **INDUCTION-20.**No part of the thesis is copied from books/ journals/ Internet and wherever the portion is taken, the same has been duly referred in the text. The report is based on the project work done entirely by TEAM *12* and not copied from any other source.

**ACKNOWLEDGMENT**

This project report is the outcome of the efforts of many people, who have driven our passion to explore into implementation of ***CHAUSER (SMART SHOE RACK)*.** We have received great guidance, encouragement and support from them and have learned a lot because of their willingness to share their knowledge and experience.

We thank our Technical Heads **M M Sai Prakash , G Datta Lohith** for being with us till the end of the project completion.

We thank the **Executive Body** and the **Technical Advisory Board** of **The Robotics Club** for helping us in crucial parts of the project. We are deeply indebted to **T.V. Hari Krishna ,** The President of **THE ROBOTICS CLUB**, **Mr. S. Sai Krishna ,** General Secretary of **THE ROBOTICS CLUB** and **Mr. Bhanu teja ,** Vice-President of **THE ROBOTICS CLUB** who spared their most valuable time without any hesitation whenever we wanted.

We also thank our Technical advisor **Dr. A. Purushotham**, Professor, Mechanical Department, who encouraged us during this project by rendering their help when needed.

CONTENTS

[List of Figures 6](#_TOC_250010)

[Abstract 7](#_TOC_250009)

[Chapter 1 Introduction 9](#_TOC_250008)

* 1. [Aim of the project 9](#_TOC_250007)
  2. Introduction of the project 9

Chapter 2 Architecture 10

* 1. [Components used 10](#_TOC_250006)

2.1.(a)ESP32 10

2.1.(b)2 channel relay 11

2.1.(c)Rack and Pinion 12

2.1.(d)L298N 12

2.1.(e) Sideshaft motors 13

2.1.(f) Ultrasonic sensors13

2.1.(g) IR sensors 14

* 1. Software 16

16

2.2. (a)Visual studio code

2.2. (b)Platformio

16

[Chapter 3 Implementation and working](#_TOC_250005)

* 1. [Block diagram 17](#_TOC_250004)
  2. Mechanical part 18
  3. [Algorithm 19](#_TOC_250002)

Chapter 4 Experimental Results and Conclusions 20

* 1. [Results 20](#_TOC_250001)
  2. [Future enhancements 20](#_TOC_250000)

4.3 Conclusions . 20

# List of Figures

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Description** | **Page No.** |
| Fig 1 | block diagram | 7 |
| Fig 2 | ESP32 | 11 |
| Fig 3 | 2 Channel Relay | 11 |
| Fig 4 | Rack and Pinion | 12 |
| Fig 5 | L298N | 12 |
| Fig 6 | Side Shaft motor | 13 |
| Fig 7 | Ultrasonic Sensor | 14 |
| Fig 8 | IR Sensors | 15 |
| Fig 9  Fig 10  Fig 11 | Mechanical Body  360 wheel  Final bot | 17  18  20 |

# ABSTRACT

TEAM-12

THE ROBOTICS CLUB –INDUCTION’20

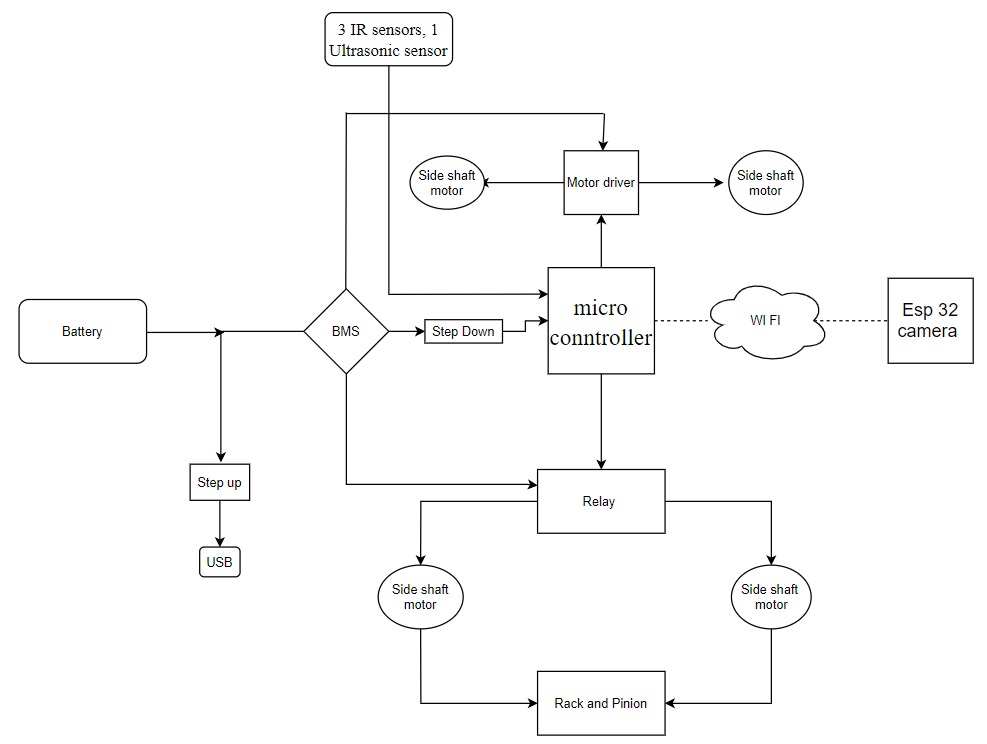
The Problem:

People feel uncomfortable to remove their shoes and get back their shoes after the completion of their work at different places like religious places or apartments and offices. Also people don’t have security for their footwear at public places, so we have come up with a solution which is a smart shoe rack.

The team’s approach to solve the problem:

Smart shoe rack bot features of a camera which is used to detect the face and it stores the data in the database. After recognizing the data of the person then it picks up the shoes from the person and then it stores the shoes of the person in the respective shoe rack which is vacant, where we use a line follower bot mechanism to move the bot from person to the rack. When the person returns back for collecting the shoes it checks for the data in the database and then collects the shoes from the rack and gives it to the person.

Block Diagram:



Fig

Title of this project:

CHAUSER

What inspired you to select the problem?

Removing shoes in public places like temples and mosques is a mess and many people lose their shoes at those places. And also, in countries like Japan and China, it is their culture to remove shoes at their work place. To make this task simple and efficient, we have come up with this idea.

What do you feel is the most innovative part of the problem?

The bot gets back the shoes whenever the person returns after completion of the work based on face detection.

**CHAPTER 1:**

**INTRODUCTION**

**1.1 AIM OF THE PROJECT:**

To make a smart shoe rack and make people feel comfortable and convenient at different locations like temples, hospitals….etc

# 1.2 INTRODUCTION TO PROJECT:

Smart shoe rack is a robot. The foremost idea of the bot is to store the shoes of the person who visits the public places and to return back the shoes whenever the person returns after completion of the work.

An ESP32 camera is used which is used to recognize the image of the person visiting the public places and security is also provided to the shoes until he collects back his shoes from the smart shoe rack .

# 

# CHAPTER-2

# Architecture

# 

# 2.Components used:

**2.1 HARDWARE:**

2.1. (a)ESP32

2.1. (b)2 channel relay

2.1. (c)Rack and Pinion

2.1. (d)L298N-1

2.1. (e) Sideshaft motors

2.1. (f) Ultrasonic sensors

2.1. (g) IR sensors

**2.2 SOFTWARE:**

2.2. (a) Visual studio code

2.2. (b) Platformio

**2 COMPONENTS DESCRIPTION:**

**2.1 HARDWARE:**

**2.1. (a)ESP32:**

The **ESP32**-**CAM** is a very small **camera** module with the **ESP32**-S chip , Besides the OV2640 **camera**, and several GPIOs to connect peripherals, it also features a microSD card slot that can be useful to store images taken with the **camera** or to store files to serve to clients.

****

Fig2

**2.1. (b)2 channel relay:**

A relay is an electrically operated device. It has a control system and (also called input circuit or input contactor) and controlled system (also called output circuit or output cont actor). It is frequently used in automatic control circuit. To put it simply, it is an automatic switch to controlling a high-current circuit with a low-current signal.

****

Fig3

**2.1. (c)Rack and Pinion:**

A rack and pinion is a type of linear actuator that comprises a circular gear (the pinion) engaging a linear gear (the rack), which operate to translate rotational motion into linear motion. Driving the pinion into rotation causes the rack to be driven linearly. Driving the rack linearly will cause the pinion to be driven into a rotation.

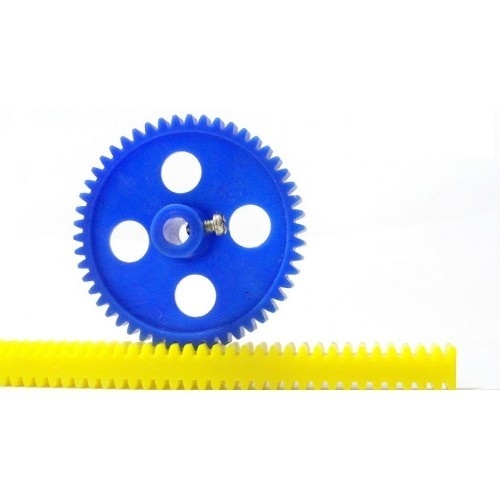
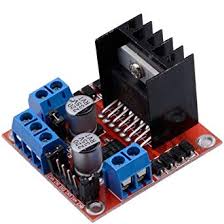


Fig4

**2.1. (d)L298N:**

The L298N is an integrated monolithic circuit in a 15- lead Multi watt and PowerSO20 packages. It is a high voltage , high current dual full-bridge driver designed to accept standard TTL logic level sand drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the in-put signals .The emitters of the lower transistors of each bridge are connected together rand the corresponding external terminal can be used for the connection of an external sensing resistor. An additional Supply input is provided so that the logic works at a lower voltage

**** Fig5

**2.1.(e) Sideshaft motors:**

200 RPM Side Shaft Heavy Duty DC Gear Motor is suitable for large robots / automation systems. It has sturdy construction with gear box built to handle stall torque produced by the motor. Drive shaft is supported from both sides with metal bushes. Motor runs smoothly from 4V to 12V and gives 200 RPM at 12V. Motor has 8mm diameter, 17.5mm length drive shaft with D shape for excellent coupling.

Fig6

**2.1.(f) Ultrasonic sensors:**

As the name indicates, ultrasonic sensors measure distance by using ultrasonic waves.

The sensor head emits an ultrasonic wave and receives the wave reflected back from the target. Ultrasonic Sensors measure the distance to the target by measuring the time between the emission and reception.

Outline and detection principle

An optical sensor has a transmitter and receiver, whereas an ultrasonic sensor uses a single ultrasonic element for both emission and reception. In a reflective model ultrasonic sensor, a single oscillator emits and receives ultrasonic waves alternately. This enables miniaturization of the sensor head.

Fig7

**2.1. (g) IR sensors:**

An infrared sensor is an electronic device, that emits in order to sense some aspects of the surroundings. An IR sensor can measure the heat of an object as well as detects the motion. These types of sensors measure only infrared radiation, rather than emitting it that is called a passive IR sensor. Usually, in the infrared spectrum, all the objects radiate some form of thermal radiation. These types of radiations are invisible to our eyes, that can be detected by an infrared sensor. The emitter is simply an IR LED (Light Emitting Diode) and the detector is simply an IR photodiode that is sensitive to IR light of the same wavelength as that emitted by the IR LED. When IR light falls on the photodiode, the resistances and the output voltages will change in proportion to the magnitude of the IR light received.



Fig8

**2.2 SOFTWARE:**

**2.2. (a)Visual studio code:**

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity).

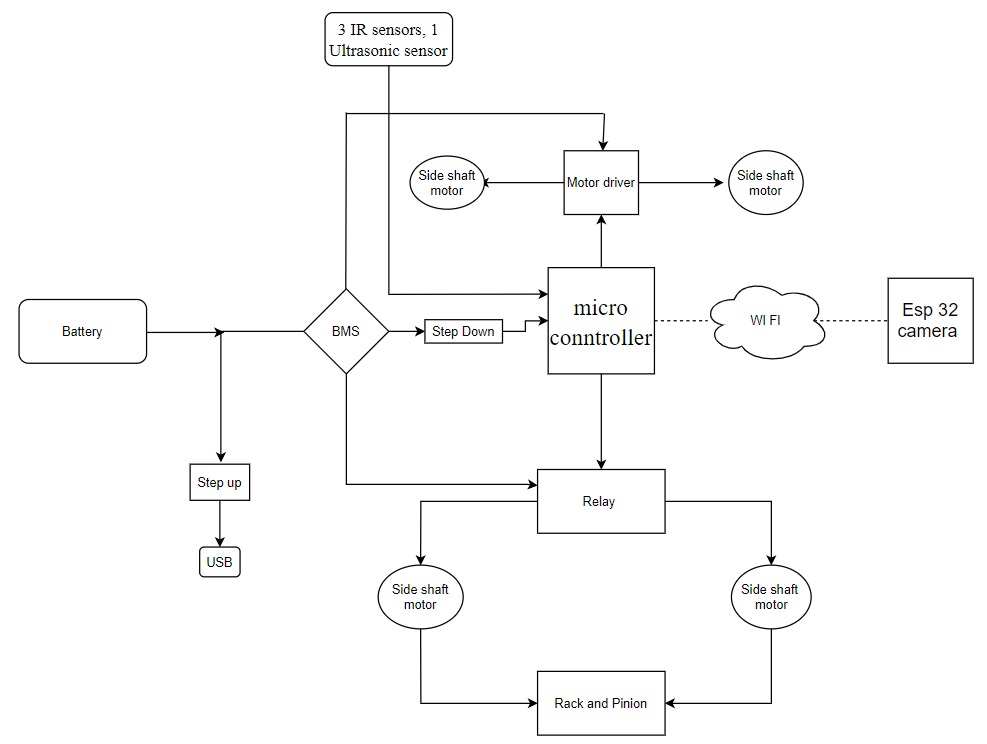
**2.2. (b)Platformio:**

This is cross platform code builder and library manager with platforms like Arduino or MBED support. They took care of toolchains, debuggers, frameworks that work on most popular platforms like Windows, Mac and Linux.

# Chapter 3

# Implementation and working

# Block diagram:



* 1. **Mechanical parts :**

**3.2.1: Body:**

The body is made of light weight wood and cardboard material





Fig 9

# 3.2.2: 360 degree wheel:

The 360 degree wheel offers free motion to the bot to move in all the directions.



Fig 12

# 3.3Algorithm:

Ultrasonic sensor is used to detect the presence of the shoes on the bot and if the shoes are present then it starts moving towards the shoe rack based on line follower (Using Ir sensors) after detection of the image of the person and stores the shoes in the shoe rack based on the vacancy of the shoes. When the person returns backs to collect the shoes back from the bot it

Detects the image of the person and gives back the shoes stored on that image detected by the ESP32 camera.

**Chapter-4**

**Experimental Results and Conclusions**

# Results:

We have successfully come up with a reliable solution to solve the problem of storing the shoes, which can be used in both public and private places like temples, parks, offices, house, etc….

**4.2Future enhancements:**

We can make the bot even more intelligent by making it complete autonomous like

to detect whether the give items are shoes or any other object.

# 4.3Conclusion:

# we have made a bot which is useful to all the people in both public and the private

# places and which reduces the human effort and also saves the time of the person.

**CODE:**

import cv2

import numpy as np

import pickle

import os

import time

from copy import copy

from PIL import Image

import webbrowser

counter = 0

frame\_counter=0

currentName = ''

name = "no"

rack = {1:'',2:'',3:''}

pname = 1

face\_cas = cv2.CascadeClassifier('C:\my files\Langages\python\python atom\haascascade\haarcascades\haarcascade\_frontalface\_default.xml')

recognizer = cv2.face.LBPHFaceRecognizer\_create()

i=1

folder = "images/id"

recognizer.read("trainner.yml")

lables = {}

with open("lables.pickle", 'rb') as f:

lables = pickle.load(f)

lables = {v:k for k,v in lables.items()}

print(lables.items())

cap = cv2.VideoCapture(0)

while(1):

if frame\_counter > 200:

frame\_counter = 0

currentName=''

ret,frame = cap.read()

frame\_copy = copy(frame)

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

faces = face\_cas.detectMultiScale(gray,1.3,5)

for (x,y,w,h) in faces:

roi\_gray = gray[y:y+h, x:x+w]

roi\_color = frame[y:y+h, x:x+w]

cv2.rectangle(frame, (x,y), (x+w,y+h), (255,255,0),2)

id\_,conf = recognizer.predict(roi\_gray)

if(conf>=45 and conf<=85):

#print(lables[id\_])

font = cv2.FONT\_HERSHEY\_DUPLEX

name = lables[id\_]

cv2.putText(frame, name,(x,y), font, 1, (0,0,255), 2 )

counter = 1

if currentName!= name:

currentName = name

for x,y in rack.items():

if name == y:

if(x==1):

webbrowser.open('192.168.4.1/one?')

rack[1]=""

print(1)

elif(x==2):

webbrowser.open('192.168.4.1/two?')

print(2)

rack[2]=""

elif(x==3):

webbrowser.open('192.168.4.1/three?')

print(3)

rack[3]=""

else:

if rack[1]=="":

webbrowser.open('192.168.4.1/one?')

print(1)

rack[1]=currentName

elif rack[2]=="":

webbrowser.open('192.168.4.1/two?')

print(2)

rack[2]=currentName

elif rack[3]=="":

webbrowser.open('192.168.4.1/three?')

print(3)

rack[3]=currentName

print(rack)

else:

print(counter)

counter+=1

try:

if counter>=20:

#time.sleep(2)

loc = folder + str(i)

os.mkdir(loc)

imgloc = loc+"/s.jpg"

cv2.imwrite(imgloc, frame\_copy)

i+=1

base\_dir = os.path.dirname(os.path.abspath(\_file\_))

image\_dir = os.path.join(base\_dir, "images")

tface\_cas = cv2.CascadeClassifier('haarcascadeFiles\haarcascade\_frontalface\_default.xml')

trecognizer = cv2.face.LBPHFaceRecognizer\_create()

x\_train = []

y\_lables = []

current\_id = 0

lable\_ids = {}

for root, dirs, files in os.walk(image\_dir):

for file in files:

if file.endswith("png") or file.endswith("jpg"):

path = os.path.join(root, file)

lable = os.path.basename(root).replace(" ", "-").lower()

if not lable in lable\_ids:

lable\_ids[lable]= current\_id

current\_id +=1

id\_ = lable\_ids[lable]

#print(lable\_ids)

pil\_image = Image.open(path).convert("L")

image\_array = np.array(pil\_image, "uint8")

#print(image\_array)

faces = tface\_cas.detectMultiScale(image\_array,1.5,5)

for (x,y,w,h) in faces:

roi = image\_array[y:y+h, x:x+w]

x\_train.append(roi)

y\_lables.append(id\_)

with open("lables.pickle", 'wb') as f:

pickle.dump(lable\_ids, f)

trecognizer.train(x\_train, np.array(y\_lables))

trecognizer.save("trainner.yml")

#time.sleep(2)

counter = 1

except(Exception):

pass

recognizer.read("trainner.yml")

lables = {}

with open("lables.pickle", 'rb') as f:

lables = pickle.load(f)

lables = {v:k for k,v in lables.items()}

print(lables.items())

cv2.imshow('vdo', frame)

frame\_counter+=1

print(frame\_counter)

print(currentName)

if(cv2.waitKey(1) == ord('q')):

break

cap.release()

cv2.destroyAllWindows()

#include <Arduino.h>

#define ir1 1

#define ir2 2

#define ir3 3

#define lf 4

#define lb 5

#define rf 6

#define rb 7

#define trig 8

#define echo 9

#define up 10

#define down 11

void getData();

void goHome();

void goTo(int);

boolean shoes();

void lower();

void rise();

void forward();

void backword();

void right();

void left();

void stop();

void setup() {

pinMode(ir1,INPUT);

pinMode(ir2,INPUT);

pinMode(ir3,INPUT);

pinMode(lf,OUTPUT);

pinMode(lb,OUTPUT);

pinMode(rf,OUTPUT);

pinMode(rb,OUTPUT);

pinMode(trig,OUTPUT);

pinMode(echo,INPUT);

pinMode(up,OUTPUT);

pinMode(down,OUTPUT);

Serial.begin(115200);

}

void loop()

{

Serial.println("loop");

getData();

}

void getData()

{

goTo(1);

}

void goHome()

{

backword();

delay(200);

while(digitalRead(1) >0 && digitalRead(2)<0 && digitalRead(3)>0)

{

if(digitalRead(1) <0 && digitalRead(2)>0 && digitalRead(3)<0)

{

forward();

}

if(digitalRead(1) >0 && digitalRead(2)<0 && digitalRead(3)<0)

{

left();

}

if(digitalRead(1) <0 && digitalRead(2)<0 && digitalRead(3)>0)

{

right();

}

if( digitalRead(2)>0 && digitalRead(3)>0)

{

right();

}

}

lower();

stop();

delay(500);

}

void goTo(int x)

{

int c=1;

if(shoes())

rise();

while(digitalRead(1) >0 && digitalRead(2)<0 && digitalRead(3)>0)

{

if(digitalRead(1) <0 && digitalRead(2)>0 && digitalRead(3)<0)

{

forward();

}

if(digitalRead(1) >0 && digitalRead(2)<0 && digitalRead(3)<0)

{

left();

}

if(digitalRead(1) <0 && digitalRead(2)<0 && digitalRead(3)>0)

{

right();

}

if(digitalRead(1) <0 && digitalRead(2)>0 && digitalRead(3)>0)

{

if(c==x)

{

right();

}

else

{

c++;

forward();

}

}

}

stop();

delay(500);

if(shoes())

lower();

else

rise();

goHome();

}

boolean shoes()

{

digitalWrite(trig,HIGH);

delay(100);

digitalWrite(trig,LOW);

delay(20);

int t = pulseIn(echo,HIGH);

int dist = t\*0.034/2;

Serial.println(dist);

if(t<10)

return true;

else

return false;

}

void lower()

{

Serial.println("lower");

digitalWrite(down,HIGH);

delay(2000);

digitalWrite(down,LOW);

}

void rise()

{

Serial.println("rise");

digitalWrite(up,HIGH);

delay(2000);

digitalWrite(up,LOW);

}

void forward()

{

Serial.println("forward");

digitalWrite(rb,LOW);

digitalWrite(lb,LOW);

digitalWrite(rf,HIGH);

digitalWrite(lf,HIGH);

}

void backword()

{

Serial.println("backword");

digitalWrite(rf,LOW);

digitalWrite(lf,LOW);

digitalWrite(rb,HIGH);

digitalWrite(lb,HIGH);

}

void right()

{

Serial.println("right");

digitalWrite(rb,LOW);

digitalWrite(lb,LOW);

digitalWrite(rf,LOW);

digitalWrite(lf,HIGH);

}

void left()

{

Serial.println("left");

digitalWrite(rb,LOW);

digitalWrite(lb,LOW);

digitalWrite(rf,HIGH);

digitalWrite(lf,LOW);

}

void stop()

{

Serial.println("stop");

digitalWrite(rb,LOW);

digitalWrite(lb,LOW);

digitalWrite(rf,LOW);

digitalWrite(lf,LOW);

}