BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

EEE402: CONTROL SYSTEM LABORATORY PROJECT

"AUTOMATIC CARGO HANDLER BOT"

Under The Supervision of:

Zabir Ahmed	Md. Shafiqul Islam
Lecturer	Lecturer
Dept. of EEE	Dept. of EEE
BUET	BUET

Task Accomplished by:

Mohaiminul Al Nahian (1106037)

Shaikhul Islam (1106046)

A.N.M Shahriyar Hossain(1106048)

Md. Tauhiduzzaman Khan(1106064)

Md. Habibullah(1106065)

Abstract:

In this project, we have developed a prototype of automated cargo handler robot. This robot can identify a cargo by its color and shape by continuously taking pictures in front and processing the image to identify the desired object. Not only identifying the cargo but also it approaches towards it, picks it up when it's in range and find the drop zone completely by image processing techniques. The total process leads to an efficient automated cargo handling system that signifies the artificial intelligence and machineries to replace the human resource in a tiring process and it was the main motivation behind the whole project.

Here, image filtering and image position techniques were applied. Also the total structure of the robot was customized as per the project demand. The wise use of buck modules and IR sensor used in the robot helped the total process to make it robust.

Introduction:

Designing an automatic system with artificial intelligence has been the prior interest point for the engineers in the recent times. This can simultaneously perform the tasks efficiently and systematically.

Transportation of materials largely handled in thousands of ports across the world that holds the most powerful economic factor. The goods contained in millions of cargo unloaded in the ship yard are arranged as per the orders completely manually. This takes a toll of time and human effort at a large extent.

Our project is the prototype to bring a revolution in the process by making the process automated completely. In order to do that, the identification of the cargo and arrangement of orders can be done by utilizing the image processing technique and carrying it using the control system techniques. This project gave an on hand experience of control system by dealing with the combination of image processing signal data and corresponding task performed by the machineries.

Thus, using color bands in the cargo can reduce the human effort in a very good way to process the cargo arrangement with precision and leave the human resource for better development.

1. Working Principle:

1.1 Video Acquisition Through Cellphone camera:

The automatic cargo handler robot has the vision with the help of camera. In order to make the project less costly, thoroughly used cellphone camera has been used as its eye. This camera takes instantaneous video with a resolution of 320×240 for low data consumption which is good enough for this robust robot to work on.

1.2 Transmitting video stream through Wi-Fi:

The video taken by the camera is to be transferred to the PC for image processing. In order to do that, we used the freeware software IP CAM in the android mobile. It enables to transmit the video stream over the Wi-Fi router. By using its own hotspot mode, we secured our video stream so that none can interfere it without the authority.

1.3 Capturing the video in PC Browser

The IP (Internet Protocol) address generated holds the video stream in the specific hotspot. In order to acquire the video, the desired pc is connected to the Wi-Fi using the secret password. Then, the pc can access the IP address and capture the video for further use.

1.4 Image Processing by Matlab:

This is the main part of the robot. In Matlab, the whole image processing has been done. The task can be segmented into several portions:

1.4.1 Initial Filtering:

The acquired video is imported and it is segmented into snapshots to make the process faster. The Image contains a lot of grains and partial colors. Using a threshold value of size, the image is filtered to remove the noise. Then we can get a better picture of the specific snap.

1.4.2 Image processing and color identification:

Using the filtered image data which is a 3 dimension matrix, we can find the desired colors values from here. By fixing the threshold pixel size of the cargo prototype with specific color, we can detect it here and locate the position of it in the specific image.

1.4.3 Location finding and decision making:

By using the position found using image data, we can locate the position of object in real. For example, if the cargo is in the left-bottom portion of the image, we can say that it is at a distance and left to the alignment of the robot. From this, we can take decision that the robot should move forward with a left alignment. These decisions are taken by interpreting the image and converted into coded data to be interpreted in the robot.

1.4.4 Serial Data Sending:

The decisions are turned into byte data and send over the serial link to the Arduino Uno by using Matlab serial communication process. To do this, Matlab opens a serial communication to the port of Arduino Uno using fopen() command and writes serial data to it using fwrite();

1.5 Sending Command from MATLAB to the BOT Using RF module

In order to send the command from the MATLAB to the robot, RF module(nRF24L01) was used. It is a pair of module of transceiver with radio frequency. One is connected to the Arduino Uno, which is connected to the PC and receives the Matlab serial data to it.

Another one is connected to the Arduino MEGA 2560 in the robot that receives the serial data over this radio communication from the previous one. This radio data receives the byte data and has a problem of buffering signals. This is avoided by using radio.available()>8 command.

1.6 The Bot executes the command:

The robot receives the data via RF module and interprets the byte data in it. In these data lines, it gets different types of command to move front, back, stop, left, right and so on. The robot executes the task accordingly for about 100ms. Then it takes another command and executes it. When the command says that there is no cargo in front, it rotates to find the object. Thus, the bot identify and approach toward the cargo.

Whenever it gets close to the cargo, it gets the pickup command by itself and picks it up using robotic hand made by servo motors and pulls it up. After that, the robot sends the command that it need a drop zone identification to the PC.

Then the robot executes the same process to find the drop zone and approaches to the drop zone with the cargo pulled up. When it reaches the drop zone, sensed by Sharp IR proximity sensor, it drops the cargo and gets back to search for another cargo.

2. Work Flow diagram:

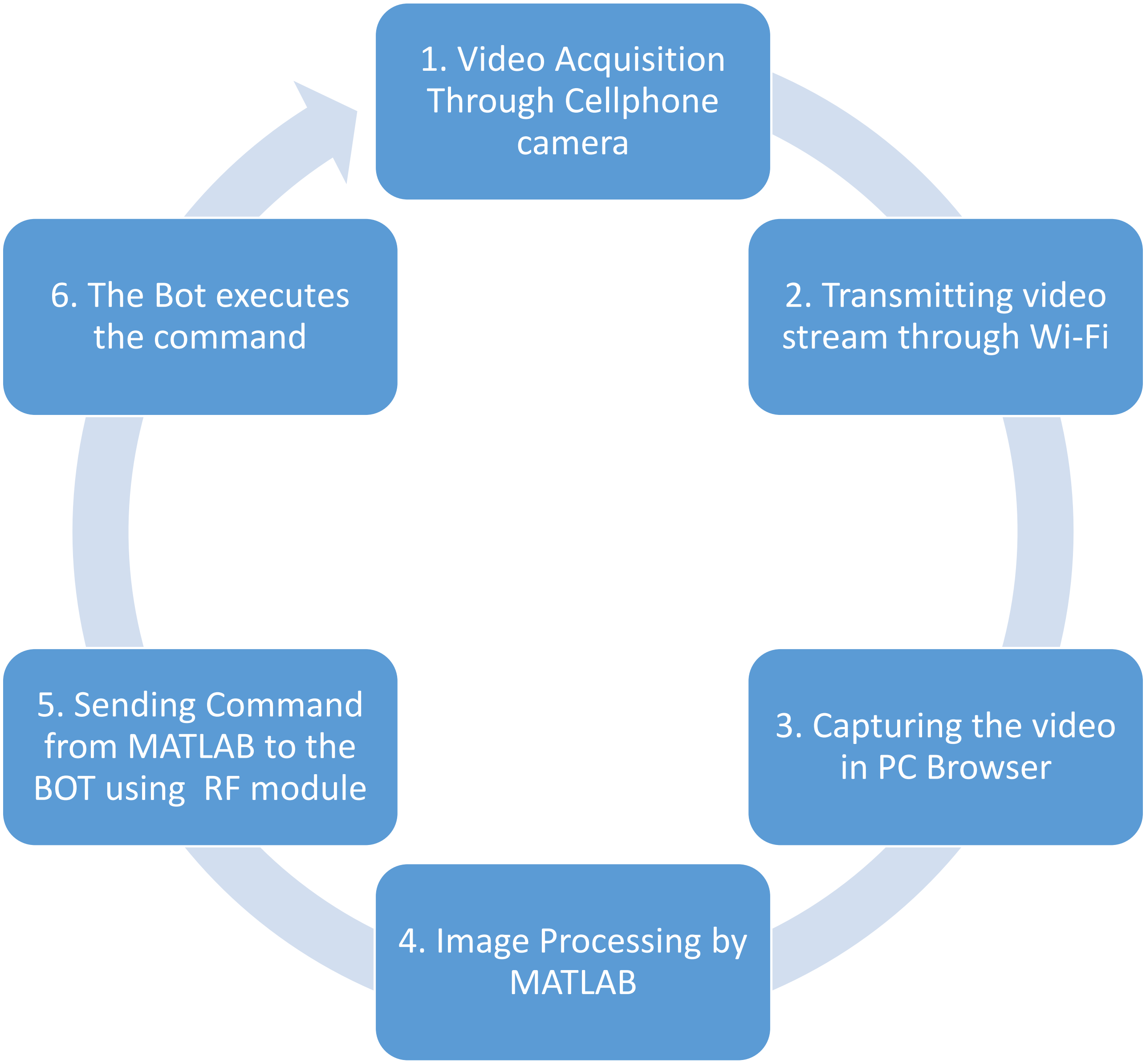
1. Video Acquisition Through Cellphone camera
2. Transmitting video stream through Wi-Fi
3. Capturing the video in PC Browser
4. Image Processing by MATLAB
5. Sending Command from MATLAB to the BOT using RF module
6. The Bot executes the command

Figure 1: Work Flow Diagram

3. Electrical Components:

1. **Arduino MEGA 2560:**
   This board acts as the Central Processing Unit (CPU) of the automation. The reason of selecting this for this specific robot is it has 16 ADC pin for better development and freedom of having pins.

Technical specs	
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by boot loader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
Length	101.52 mm
Width	53.3 mm
Weight	37 g

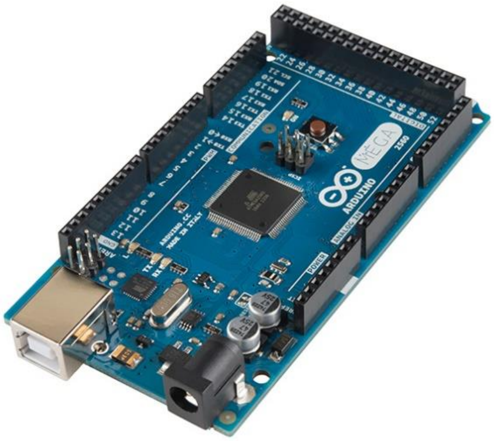

Figure 2: Arduino MEGA 2560

2. **Motor Driver Shield:**
   L293N motor driver shield was used to run the wheels of the robot. It comes with a built in heat-sink for smooth performance.

Figure 3: Motor Driver Shield L293N

3. **RF Module:**
nRF24L01, The RF module pair with a range about 50-200 feet to transmit the image processing serial data from computer to the Arduino board via radio communication.

Figure 5: NRF24L01

Figure 4: Servo Motor

4. **Servo Motor:**

Two Servo Motor was used in the robot to precisely control the hand in vertical and horizontal motion.

Speed 0.14sec/60degrees (4.8V) 0.12sec/60degrees (6.0V)
Torque 2.5kgf-cm (4.8V) 3.0kgf-cm (6.0V)
Voltage 4.8V-6.0V
Connector type JR type

5. **Power Supply:**
The main supply was split into two parts. One is for the CPU, communication and motors, where another was for the robotic hand. For that purpose, 2 pair of 8V Li-Ion battery was used which was made by cascading two 4V batteries.

Figure 6: 8V Li-Ion Battery

Figure 7: Buck Conveter

6. **Buck Converter Module:**
To supply the wheels with constant voltage to ensure the constant PWM (Pulse Width Modulation) output, LM2596 DC-DC Buck converter was used onboard. We generated 7v constant output supply by it to the motor driver.

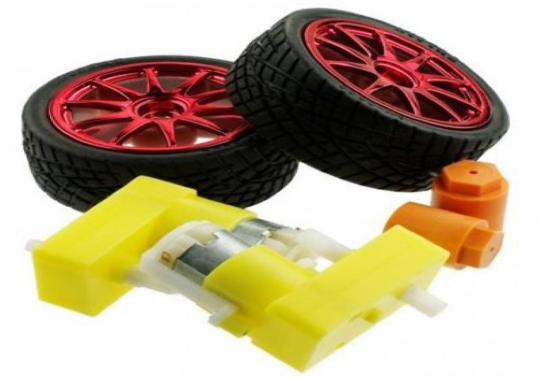

Figure 8 : DC Motor

7. DC Motor

We have used two DC motor for the wheels of the robot. The motors were rated at 6-12 volt and required good constant voltage and current. That was done by the buck and motor driver module. As per the requirement, we used high torque motor here and the supply voltage of 7V in the motor to work well with PWM.

8. **Arduino Uno**:

We used a Arduino Uno connected to the PC to transmit the serial data from the Matlab to the Bot using RF module.

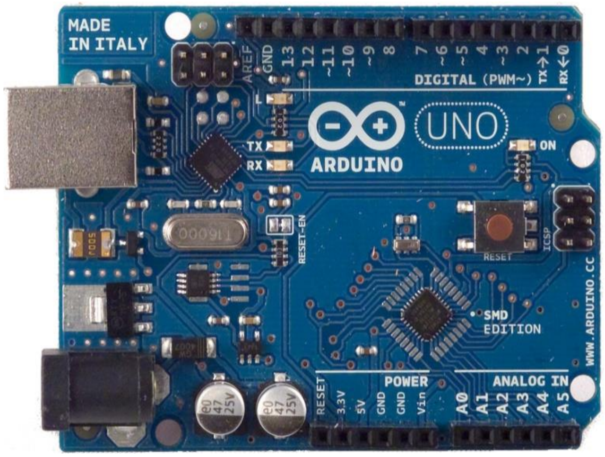

Figure 10: Arduino Uno

Technical specs	
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB
SRAM	2 KB
EEPROM	1 KB

9. **Sharp GP2Y0A21YK0F IR Distance Sensor :**

To find the distance from the cargo and the drop zone barrier, we used Sharp IR proximity sensor which gives a better approximation for low distance objects.

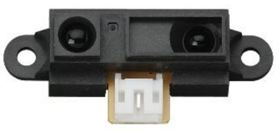

Figure 9: Sharp IR

Conclusion:

This project has been a prototype for large scale cargo handling system. Preliminarily, we started with color detection to find the cargo. But, it can be developed as shape detection with it too. Also, it can be a line follower one like the roads in the shipyard that will reduce the upwelling of the robot and make the task easier. We could have developed it that far, but the limitation of time of submission compelled us to limit it up to this point.

We faced a huge barrier to send serial data of image processing decision from Matlab to the robot. This is because of the processing time of Matlab and radio data receiving speed of Arduino. For further development, we thought of using Raspberry Pi onboard to reduce the complexity of using PC and video streaming as well as the serial communication speed. That will make the robot receive signal in real time and run on smooth.

Finally, this project has endured us roll deep into control system techniques with image processing to cope up with this technological globe. And obviously, it boosted our confidence to take up the challenging tasks and think alive.

Appendix:

MATLAB CODE for Image Processing:

```
clc;
clear all;
close all;
s=serial('COM4','Baudrate',9600);
fopen(s);


% Capture the video frames using the videoinput function
% You have to replace the resolution & your installed adaptor name.

%%%%vid = videoinput(camera_name, camera_id, format);
vid=ipcam('http://192.168.0.100:8080/video');




%start the video aquisition here
%%%%start(vid)
i=1;
% Set a loop that stop after 100 frames of aquisition
%%%%while(vid.FramesAcquired<=200)
while(1)
    tic;
    red=0;

    blue=0;

    % Get the snapshot of the current frame
    %%%%    data = getsnapshot(vid);
    data=snapshot(vid);
    [a b c]=size(data);
    y=a;
    x=b;

    diff_im = imsubtract(data(:,:,1), rgb2gray(data));
    diff_im = medfilt2(diff_im, [3 3]);
    diff_im = im2bw(diff_im,0.2);
    diff_im = bwareaopen(diff_im,300);


    %se = strel('disk',4);
    %diff_im = imclose(diff_im,se);%figure;
    %diff_im = imfill(diff_im,'holes');

    % Label all the connected components in the image.
    bw = bwlabel(diff_im, 8);
    [B,L] = bwboundaries(bw,'noholes');
    stats = regionprops(bw,'Area', 'BoundingBox', 'Centroid');

    imshow(data)

    hold on


    for g = 1:length(stats)

        bb = stats(g).BoundingBox;
        centroid = stats(g).Centroid;
        red=1;
        rectangle('Position',bb,'EdgeColor','r','LineWidth',2)
        plot(centroid(1),centroid(2), '-m+')
        a=text(centroid(1)+15,centroid(2), strcat('X: ',
num2str(round(centroid(1))), '    Y: ', num2str(round(centroid(2)))));
        set(a, 'FontName', 'Arial', 'FontWeight', 'bold', 'FontSize', 12,
'Color', 'yellow');

        hold on;


    end
    if red==0
        fprintf(s,'N');

    elseif red==1

        cenx=centroid(1);
        ceny=centroid(2);
        X1=x/2+30; % center region of the frame
        Y1=y/2+30;
        X2=x/2-30;
        Y2=y/2-30;


        if (cenx>X2 && cenx<X1)
            fprintf(s,'F');
        elseif (cenx>X1)                              % 1st Quadrant
            fprintf(s,'R');
        elseif (cenx<X2)                          % 2nd Quadrant
            fprintf(s,'L');
        end

    end



    %Blue zone
    diff_im2 = imsubtract(data(:,:,3), rgb2gray(data));
    diff_im2 = medfilt2(diff_im2, [3 3]);
    diff_im2 = im2bw(diff_im2,0.2);
    diff_im2 = bwareaopen(diff_im2,300);
    %se = strel('disk',4);
    %diff_im = imclose(diff_im,se);%figure;
    %diff_im = imfill(diff_im,'holes');

    % Label all the connected components in the image.
    bw2 = bwlabel(diff_im2, 8);

    [B,L] = bwboundaries(bw2,'noholes');
    stats2 = regionprops(bw2,'Area', 'BoundingBox', 'Centroid');

    imshow(data)

    hold on


    for g2 = 1:length(stats2)

        bb2 = stats2(g2).BoundingBox;
        centroid2 = stats2(g2).Centroid2;
        blue=1;
        rectangle('Position',bb2,'EdgeColor','r','LineWidth',2)
        plot(centroid2(1),centroid2(2), '-m+')
        a2=text(centroid2(1)+15,centroid2(2), strcat('X: ',
num2str(round(centroid2(1))), '    Y: ', num2str(round(centroid2(2)))));
        set(a2, 'FontName', 'Arial', 'FontWeight', 'bold', 'FontSize', 12,
'Color', 'yellow');




        toc;
    end
    if blue==0
        fprintf(s,'N');

    elseif blue==1

        cenx2=centroid2(1);
        ceny2=centroid2(2);
        X1=x/2+30; % center region of the frame
        Y1=y/2+30;
        X2=x/2-30;
        Y2=y/2-30;



        if (cenx2>X2 && cenx2<X1)
            fprintf(s,'f');
        elseif (cenx2>X1)                             % 1st Quadrant
            fprintf(s,'r');
        elseif (cenx2<X2)                         % 2nd Quadrant
            fprintf(s,'l');
        end
    end
    hold off
end
```

Arduino CODE for Robot:

```
#include <SPI.h>
#include <Servo.h>
#include <SharpIR.h>
#include "nRF24L01.h"
#include "RF24.h"

#define ir1 A0
#define model 1080
//RF module
int msg[2];
RF24 radio(9,53);
const uint64_t pipe = 0xE8E8F0F0E1LL;

volatile int finish=0;

Servo myservo1;
Servo myservo2;
volatile int dis1 = 120;
volatile int picked = 0;
SharpIR sharp1(ir1, 100, 93, model);

int right_front = 2;
int right_back = 3;
int right_en = 4;
int left_front = 6;
int left_back = 7;
int left_en = 8;

int ls=13;
int fs=11;
int rs=12;
int rn=10;


volatile char a = 0;

void setup(void) {

  delay(2000);

   Serial.begin(9600);


  myservo1.attach(45);
  myservo2.attach(44);

  radio.begin();
  radio.setDataRate(RF24_250KBPS); // Transmission speed
radio.setChannel (100); // Channel number from 0 to 127
radio.setRetries (15,15); //
radio.openReadingPipe(1,pipe);
radio.startListening();
  //remember to use for loop here finally to calculate angle of servo
  for (int i = 20; i >= 5; i = i - 1)
  { myservo1.write(i);
```

```
    delay(50);
  }

  myservo2.write(20);
  delay(1000);



  pinMode(left_front, OUTPUT);
   pinMode(rn, OUTPUT);


  pinMode(left_back, OUTPUT);
  pinMode(right_front, OUTPUT);
  pinMode(right_back, OUTPUT);
  pinMode(left_en, OUTPUT);
  pinMode(right_en, OUTPUT);
  pinMode(ls, OUTPUT);
  pinMode(fs, OUTPUT);
  pinMode(rs, OUTPUT);
  pinMode(32, OUTPUT);

  pinMode(ir1, INPUT);
}



void loop()
{ if(finish==0){
  dis1 = sharp1.distance();

if(picked==0){
```

```
if (dis1 >= 13 && picked == 0) {

  if (radio.available()) {
    radio.read(msg,2);

    switch (msg[0]) {
      case 'F' ://front
        front();

        break;

      case 'R' ://right front
        rightf();
        break;

      case 'L' ://left front
        leftf();
        break;

      case 'N':
       randomrun();
        break;
        case 'f':
        randomrun();
        break;

        case 'r':
        randomrun();
        break;

        case 'l':
          randomrun();
          break;


      }


    }//if serial available

  }//if dis>12
  else if (picked == 0 && dis1 < 10)
  {
    back();
  }
  else if (picked == 0 && dis1 <= 13 && dis1 >= 10)
  { //stop when less than 12
   pickobject();
  }
}

//for picked ==1
else if(picked==1){

  if (dis1 >= 13 && picked ==1) {

    if (radio.available()) {



       radio.read(msg,2);

      switch (msg[0]) {
```

```
      case 'f' ://front
        front();

        break;

      case 'r' ://right front
        rightf();
        break;

      case 'l' ://left front
        leftf();
        break;

      case 'N':
        randomrun();
        break;


    }


  }//if serial available

}//if dis>12
else if (picked == 1 && dis1 < 10)
{
  back();
}
else if (picked == 1 && dis1 <= 13 && dis1 >= 10)
{ //stop when less than 12
 dropobject();
}
```

```
}// if picked==1


}// finish=0
}//loop


void front() {
//
digitalWrite(fs,1);
digitalWrite(ls,0);
digitalWrite(rs,0);
digitalWrite(rn,0);

  //delay(20);
  digitalWrite(left_back, 0);
  digitalWrite(left_front, 1);
  digitalWrite(right_front, 1);
  digitalWrite(right_back, 0);
  analogWrite(left_en, 195);
  analogWrite(right_en, 220);
  delay(100);

  //delay(500);
  //delay(20);

}

void back() {

  //delay(20);
  digitalWrite(left_back, 1);
  digitalWrite(left_front, 0);
  digitalWrite(right_front, 0);
  digitalWrite(right_back, 1);
  analogWrite(left_en, 160);
  analogWrite(right_en, 160);
  delay(300);

}



void rightf() {



digitalWrite(fs,0);
digitalWrite(ls,0);
digitalWrite(rs,1);
digitalWrite(rn,0);
  digitalWrite(left_back, 0);
  digitalWrite(left_front, 1);
  digitalWrite(right_front, 1);
  digitalWrite(right_back, 0);
  analogWrite(left_en, 220);
  analogWrite(right_en, 155);
  delay(100);

  //delay(50);
}



void leftf() {

digitalWrite(fs,0);
digitalWrite(ls,1);
digitalWrite(rs,0);
digitalWrite(rn,0);
  //delay(20);
  digitalWrite(left_back, 0);
  digitalWrite(left_front, 1);
  digitalWrite(right_front, 1);
  digitalWrite(right_back, 0);
  analogWrite(left_en, 150);
  analogWrite(right_en, 230);
  //delay(180);

  delay(100);
}



void randomrun() {
  //delay(20);
  digitalWrite(fs,0);
digitalWrite(ls,0);
digitalWrite(rs,0);
digitalWrite(rn,1);
  digitalWrite(left_back, 1);
  digitalWrite(left_front, 0);
  digitalWrite(right_front, 1);
  digitalWrite(right_back, 0);
  analogWrite(left_en, 130);
  analogWrite(right_en, 130);
  delay(100);

//  digitalWrite(left_back, 0);
//  digitalWrite(left_front, 0);
//  digitalWrite(right_front, 0);
//  digitalWrite(right_back, 0);
//  analogWrite(left_en, 0);
//  analogWrite(right_en, 0);
//  delay(10);


}

void STOP() {

  digitalWrite(left_back, 0);
  digitalWrite(left_front, 0);
  digitalWrite(right_front, 0);
  digitalWrite(right_back, 0);
  analogWrite(left_en, 0);
  analogWrite(right_en, 0);
  delay(50);

}

void pickobject()
{
   digitalWrite(left_back, 0);
    digitalWrite(left_front, 0);
    digitalWrite(right_front, 0);
    digitalWrite(right_back, 0);
    analogWrite(left_en, 0);
    analogWrite(right_en, 0);
    delay(50);

    for (int j = 20; j < 170; j=j+2) {
```

```
    myservo2.write(j);
    delay(10);
  }
  delay(500);


  for (int k = 5; k < 179; k=k+2) {
    myservo1.write(k);
    delay(10);
  }

  delay(500);

  for (int l = 169; l > 20; l--) {
    myservo2.write(l);
    delay(20);
  }
  delay(500);

  for (int m = 178; m >= 5; m--) {
    myservo1.write(m);
    delay(10);
  }
  delay(500);


  //up


  picked = 1;
  back();
}
```

```
void dropobject(){

  STOP();

  for (int j = 5; j <=155; j++) {
    myservo1.write(j);
    delay(10);
  }
  delay(500);

  for (int k = 20; k < 169; k++) {
    myservo2.write(k);
    delay(10);
  }

  delay(500);

  for (int l = 155; l >= 5; l--) {
    myservo1.write(l);
    delay(20);
  }
  delay(500);

  for (int m = 169; m >= 20; m--) {
    myservo2.write(m);
    delay(10);
  }
  delay(500);

  picked = 0;
  dropback();
  STOP();

  finish=1;
  digitalWrite(32,1);
  delay(1000);
  digitalWrite(32,0);
}

void dropback()
{
  digitalWrite(left_back, 1);
  digitalWrite(left_front, 0);
  digitalWrite(right_front, 0);
  digitalWrite(right_back, 1);
  analogWrite(left_en,200);
  analogWrite(right_en,200);
  delay(680);

  digitalWrite(left_back, 0);
  digitalWrite(left_front, 0);
  digitalWrite(right_front, 0);
  digitalWrite(right_back, 0);
  analogWrite(left_en,0);
  analogWrite(right_en,0);
  delay(20);
  }
```