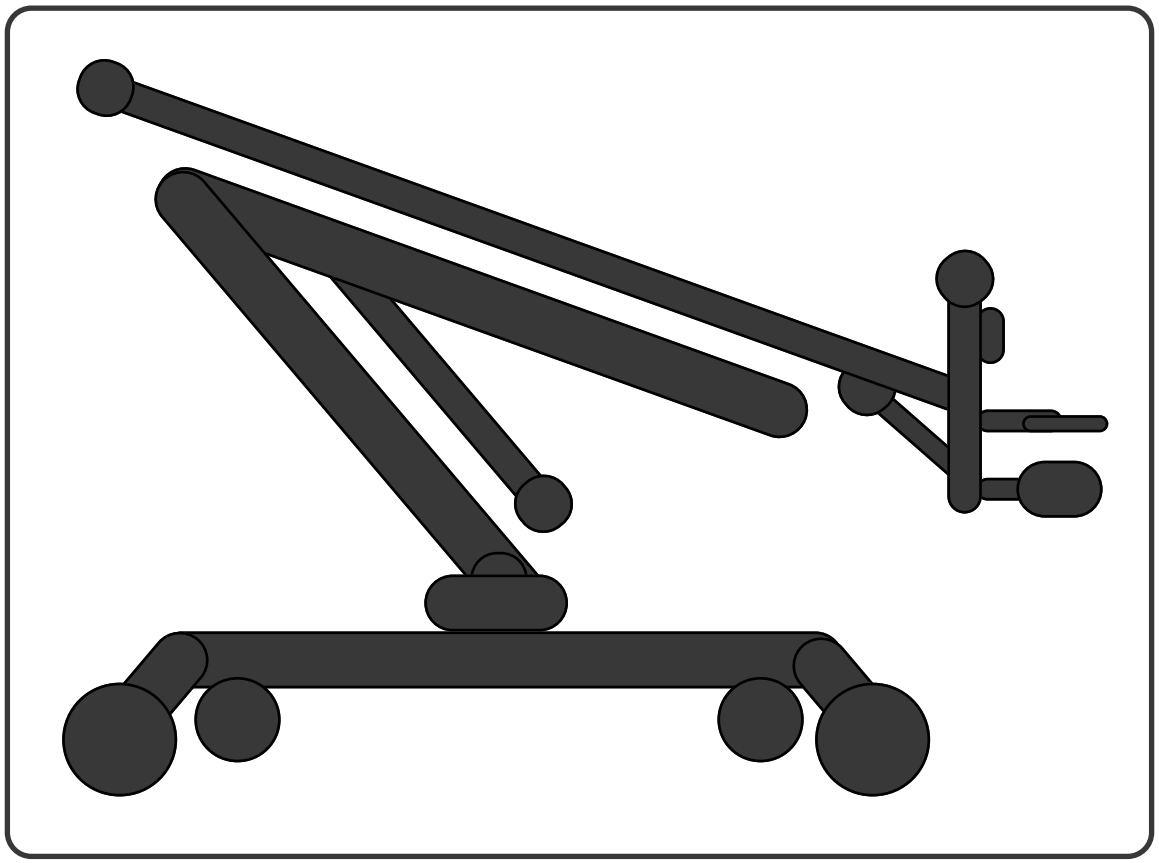
Green Buddy Robot

Agricultural Robot designed to harvest crops such as fruits and vegetables



This file will describe

Functionality
Hardware
Electronics
Design
Software
mobile Application

Hardware

PVC Pipe

The main material chosen to create the robot's body is PVC plastic pipes, which are considered to be strong enough to handle the required weight and movement.

Pipe Hanger

The metal Pipe Hanger was chosen to join the robot's parts together since it was strong and compatible with the pipes used.

pipe caps

To improve the design and hide the wires, each tube's side holes were closed with a pipe cap.

pipe t shape

A T-shaped pipe was used to design a part of the robot's base, making it more balanced and stable.

Office chair legs

Office chair legs were used to build the moveable basis for the robot, allowing it to rotate 180 degrees.

Motorcycle gears

Motorcycle gears were used to connect the moving base to the motor generating the movement.

Motorcycle chain

A motorcycle chain was used to connect the gears that move the base.

Hardware

Drawer slides

Drawer slides were used in specific parts of the robot to facilitate part movement and make it more reliable.

Satellite Dish Motor

The Satellite Dish Motor was used to control the main movements of the robot due to their strong performance.

Tension spring

Tension springs were used to balance the force and weight of the robot loads caused by carrying the robot head.

Scissor

The scissor was modified with a motor to cut the neck of the fruit after it's been grabbed by the robot's hand.

Food scoop scissors

Food scoop scissors were used after modifying and installing a motor to work as a fruit catcher.

Bike phone holder

The bike phone holder was used to place the phone in the appropriate place, allowing us to use the phone's camera correctly.

Gear Motor

A gear motor is used to mechanically control the scissor and gripper parts.

Electronics

Arduino Mega

The Arduino Mega will be used as the brain controller for this project.

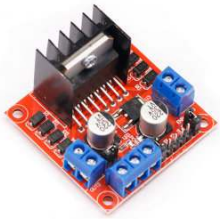
Hc-05 Bluetooth module

The HC-05 Bluetooth module will be used as the link between the application and the microcontroller.

8 Channel Relay

An 8-channel relay will be used to convert the required voltage between the motors.

L298 Motor Driver Module



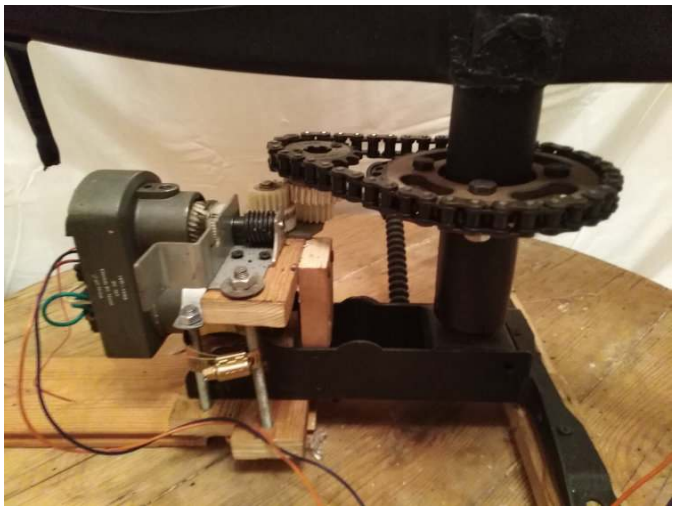
The L298 Motor Driver Module will be used for controlling motors that require less than 12 volts.

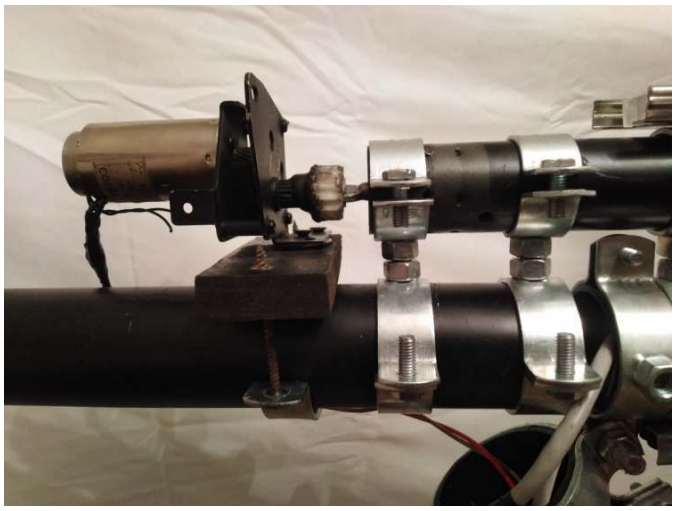
RJ45 Network Cable

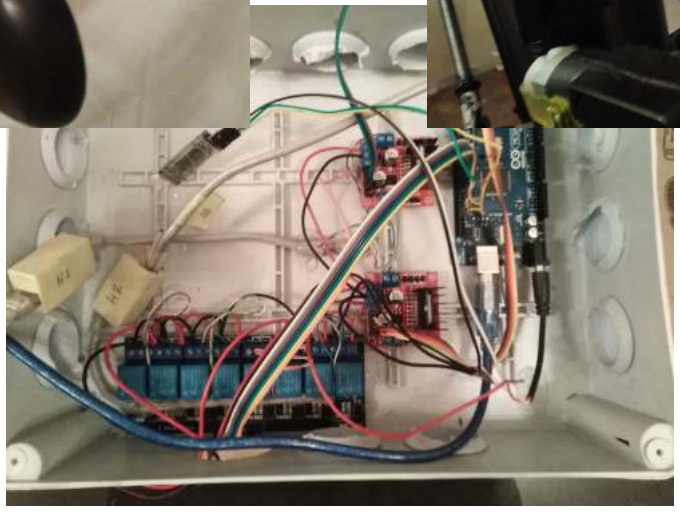
The RJ45 network cable will be used to make basic connections between the motors and the microcontroller.

Design

Design







Software

The Microcontroller Software

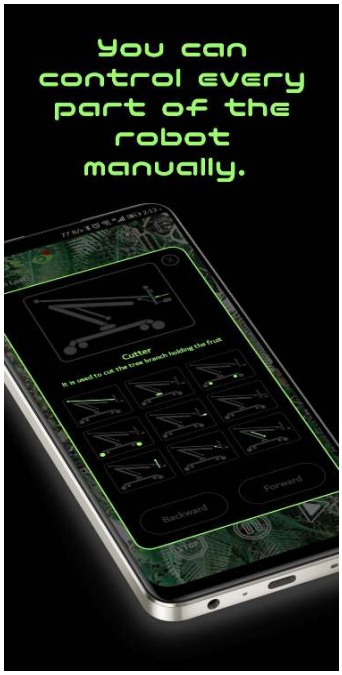
Of course, the C language will be used to programme the Arduino Mega microcontroller. The simple code will be as follows:

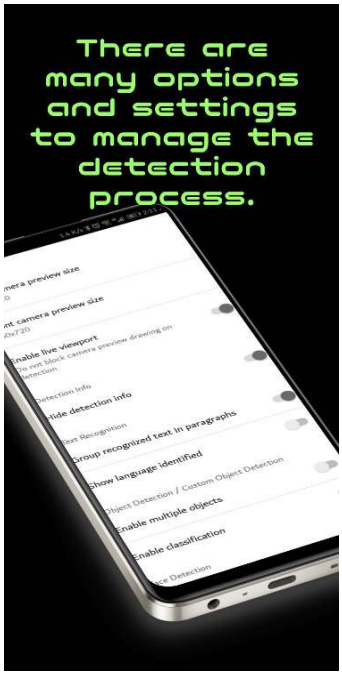
```
#include <Wire.h>
#include<SoftwareSerial.h>

////////////////////////////
//Driver one
int d1_in1 =22;
int d1_in2 =23;
int d1_in3 =24;
int d1_in4 =25;

//Driver two
int d2_in1 =26;
int d2_in2 =27;
int d2_in3 =28;
int d2_in4 =29;

//Relay Module
int in1 =2;
int in2 =3;
int in3 =4;
int in4 =5;
int in5 =6;
int in6 =7;

//Bluetooth serial
const int txpin = 12;
const int rxpin = 13;

SoftwareSerial blue(txpin, rxpin);
char Command = 'o'  ;

void setup() {

  blue.begin(9600);
  // Serial.begin(115200);
  blue.setTimeout(10);

    pinMode(d1_in1, OUTPUT);
    pinMode(d1_in2, OUTPUT);
    pinMode(d1_in3, OUTPUT);
    pinMode(d1_in4, OUTPUT);

    pinMode(d2_in1, OUTPUT);
    pinMode(d2_in2, OUTPUT);
    pinMode(d2_in3, OUTPUT);
    pinMode(d2_in4, OUTPUT);

    pinMode(in1, OUTPUT);
    pinMode(in2, OUTPUT);
    pinMode(in3, OUTPUT);
    pinMode(in4, OUTPUT);
    pinMode(in5, OUTPUT);
    pinMode(in6, OUTPUT);

    digitalWrite(in1, HIGH);
    digitalWrite(in2, HIGH);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, HIGH);
    digitalWrite(in5, HIGH);
    digitalWrite(in6, HIGH);

}
void loop() {

   if (blue.available()) {
     while (blue.available()) {
       Command = blue.read();

      // MOTOR_HEAD_UD
      if(Command == 'E')
{ControlMotors(d1_in1,HIGH,d1_in2,LOW);}
      if(Command == '4')
{ControlMotors(d1_in1,LOW,d1_in2,LOW);}
      if(Command == 'e')
{ControlMotors(d1_in1,LOW,d1_in2,HIGH);}

      //MOTOR_HEAD_LR
      if(Command == 'D')
{ControlMotors(d1_in3,HIGH,d1_in4,LOW);}
      if(Command == '3')
{ControlMotors(d1_in3,LOW,d1_in4,LOW);}
      if(Command == 'd')
{ControlMotors(d1_in3,LOW,d1_in4,HIGH);}

      // MOTOR_CUT
      if(Command == 'F')
{ControlMotors(d2_in1,HIGH,d2_in2,LOW);}
      if(Command == '5')
{ControlMotors(d2_in1,LOW,d2_in2,LOW);}
      if(Command == 'f')
{ControlMotors(d2_in1,LOW,d2_in2,HIGH);}

      //MOTOR_CATCH
      if(Command == 'G')
{ControlMotors(d2_in3,HIGH,d2_in4,LOW);}
      if(Command == '6')
{ControlMotors(d2_in3,LOW,d2_in4,LOW);}
      if(Command == 'g')
{ControlMotors(d2_in3,LOW,d2_in4,HIGH);}

       //MOTOR_BASE
      if(Command == 'A')
{ControlMotors(in1,HIGH,in2,LOW);}
      if(Command == 'O')
{ControlMotors(in1,HIGH,in2,HIGH);}
      if(Command == 'a')
{ControlMotors(in1,LOW,in2,HIGH);}

      //MOTOR_UP_DOWN
      if(Command == 'T')
{ControlMotors(in3,HIGH,in4,LOW);}
      if(Command == '1'){
ControlMotors(in3,HIGH,in4,HIGH);}
      if(Command == 't')
{ControlMotors(in3,LOW,in4,HIGH);}

      //MOTOR_ARM
      if(Command == 'C')
{ControlMotors(in5,HIGH,in6,LOW);}
      if(Command == '2')
{ControlMotors(in5,HIGH,in6,HIGH);}
      if(Command == 'c')
{ControlMotors(in5,LOW,in6,HIGH);}

   }}}

void ControlMotors(int pin1,int val1,int pin2,int
val2){
        digitalWrite(pin1, val1);
        digitalWrite(pin2, val2);
  }
```

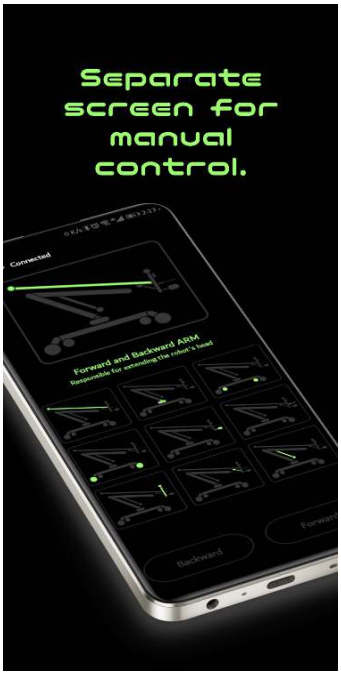
mobile Application

Green Buddy application

This Green Buddy application functions as both the robot's eye and brain.







Functionality

The Green Buddy project is a huge and difficult project that mixes mechanics, microcontroller programming, and mobile application programming, and we use machine learning techniques everywhere.

The concept is to develop a robot arm that can move freely in all directions and search for fruits, determining their location and size using a phone camera mounted in a precise area on the robot's head. After that, it can approach the fruit, hold it, and pick it.

The Arduino microcontroller is programmed in C, and the simple code's task is to receive commands sent via phone application and translate them to move a specific motor for a set amount of time.

The phone application is considered the robot's eyes and brain because it performs the majority of the work. Its mission is to search and analyse images captured by the phone camera and translated using machine learning libraries. When it finds the fruit, it sends the necessary commands and coordinates to the robot mechanics, allowing them to move, catch, and pick. The fruit