Project Report On

The Smart Agriculture Line Following Bot

SUBMITTED BY :

POPURI HEMANTH KUMAR

M.Eesha

S. Ashish

A.Ramya

Shruthi

Charan

Abrar

Akshaya

Umair

Contents

THE PROBLEM

Agricultural fields face constant threats from fluctuating environmental conditions. Soil humidity, pollution levels, and unpredictable weather patterns significantly impact crop health and yield. Variations in soil moisture can hinder nutrient absorption, while pollutants can stunt growth and reduce crop quality. Extreme weather, such as droughts or excessive rainfall, can devastate entire harvests. These environmental pressures create a challenging landscape for farmers.Traditional manual monitoring methods often prove inadequate. Physically inspecting vast fields is time-consuming, expensive, and labor-intensive. This approach frequently leads to delayed problem detection, hindering timely intervention. Farmers may struggle to make informed decisions regarding irrigation, fertilization, and pest control due to a lack of real-time, comprehensive data. Consequently, crop yields may suffer, and potential profits are diminished. The need for efficient and accurate field monitoring is crucial for sustainable and profitable agriculture.

TEAM'S APPROACH TO SOLVE THE PROBLEM:

This project introduces an autonomous surveillance bot designed to revolutionize agricultural practices. Addressing the challenges farmers face in monitoring vast fields, the bot employs a grid-based navigation system for systematic coverage. Equipped with advanced sensors, it gathers crucial real-time data, including soil humidity, pollution concentrations, and prevailing weather conditions. This information empowers farmers with actionable insights for optimized resource allocation. Complementing the sensor data, a 360-degree camera system provides comprehensive visual surveillance. Farmers can remotely access live video feeds, enabling prompt detection of anomalies like pest infestations, crop diseases, or irrigation issues. This proactive approach minimizes potential losses and maximizes crop yields, contributing to more sustainable and efficient farming. The bot's autonomous nature reduces the need for manual labor, saving time and resources. Ultimately, this technology aims to bridge the gap between traditional farming methods and modern technological advancements, fostering a more resilient and productive agricultural sector.

ABSTRACT

Agricultural fields face numerous environmental challenges, including fluctuating soil humidity, varying pollution levels, and unpredictable weather conditions. These factors significantly impact crop yield and quality, making efficient monitoring crucial. Traditional manual methods are often inadequate, proving time-consuming, costly, and inefficient. This often leads to delayed issue detection and suboptimal decision-making for farmers, hindering productivity and profitability.

To address these challenges, our project introduces an autonomous surveillance bot designed to revolutionize agricultural monitoring. This bot is equipped with advanced sensors and cutting-edge technology, enabling it to systematically monitor fields using a grid-based approach. It captures real-time data on critical parameters, including soil humidity levels, pollution concentrations, and prevailing weather conditions. This precise data empowers farmers with actionable insights into the specific needs of their crops.

Furthermore, the bot features a 360-degree camera system, providing comprehensive visual surveillance of the entire field. Farmers can remotely access live video feeds, enabling them to promptly detect potential threats or anomalies, such as pest infestations, crop diseases, or irrigation issues. This proactive approach allows for timely intervention, minimizing potential losses and maximizing crop yields. By providing farmers with timely and accurate information, our surveillance bot facilitates data-driven decision-making. This empowers them to optimize resource allocation, including irrigation, fertilization, and pest control, leading to enhanced agricultural productivity and sustainable practices. Ultimately, this technology aims to bridge the gap between traditional farming methods and modern technological advancements, fostering a more resilient and efficient agricultural sector.

Chapter 1

INTRODUCTION

Smart agriculture is revolutionizing traditional farming through technological integration, boosting productivity and promoting sustainability. A prime example is the development of smart agriculture line-following bots. These autonomous robots navigate fields precisely, adhering to pre-defined paths, and executing tasks like planting, weeding, fertilizing, and crop health monitoring.

These bots utilize a combination of sensors, machine learning algorithms, and GPS technology to maintain accurate course along designated lines. This automation significantly reduces manual labor, minimizes human error, and increases field operation efficiency. By consistently and accurately performing tasks, these bots empower farmers to optimize resource utilization, reduce operational costs, and ultimately enhance crop yields.

Integrating line-following bots into agricultural practices streamlines operations and supports sustainable farming. Precise input application minimizes waste and promotes environmentally friendly practices. This targeted approach reduces the overuse of fertilizers and pesticides, benefiting both the environment and the farmer's bottom line. As the agricultural sector continues to embrace technological advancements, line-following bots represent a significant step towards more efficient, sustainable, and profitable farming practices. They signify the shift towards data-driven agriculture, where technology plays a crucial role in optimizing every stage of the farming process.

1.1 EXISTING SYSTEM

Smart agriculture line-following bots are autonomous robots revolutionizing field operations. Designed to execute diverse agricultural tasks, these robots navigate pre-defined paths or lines within the field, enhancing efficiency and precision. They represent a significant advancement in smart farming technology.These bots integrate several key components. A robust navigation system, often utilizing GPS and computer vision, enables accurate path following. Sensors, such as cameras and proximity detectors, perceive the environment and ensure obstacle avoidance. A control system processes sensor data and directs the bot's movement, maintaining it on the designated line. Actuators, like motors, drive the wheels or other locomotion mechanisms. Finally, a task-specific payload, which could be planting, weeding, or fertilizing equipment, allows the bot to perform its designated function. Existing systems often incorporate machine learning algorithms to improve path following, adapt to changing field conditions, and even identify potential issues like crop disease. The integration of these technologies allows for optimized resource use, reduced labor costs, and increased crop yields, making these line-following bots a crucial tool for modern, sustainable agriculture.

1.2 PROPOSED SYSTEM

This proposed smart agriculture line-following bot offers a promising approach to automating key farming tasks. Designed for autonomous navigation, it can plant, water, fertilize, and monitor crops, increasing efficiency and reducing labor. The core of its operation lies in the line-following mechanism. Infrared (IR) or optical sensors detect pre-laid lines on the ground, guiding the bot's movement. A microcontroller processes this sensor data, making real-time adjustments to the bot's path to ensure accurate line following.The bot begins at a designated field point and follows the lines, continuously monitoring both soil moisture and environmental conditions. This constant monitoring is crucial for optimized resource allocation. When soil moisture falls below a pre-defined threshold, the bot's integrated watering system activates, providing targeted irrigation. All collected data, including moisture levels and environmental readings, are logged and transmitted to the cloud, enabling real-time monitoring and valuable historical analysis for farmers.Obstacle detection is also incorporated. If the bot encounters an obstacle, it autonomously navigates around it, ensuring uninterrupted operation.

Chapter 2

ARCHITECTURE

2.1 HARDWARE COMPONENTS

2.1.1 Esp 32

The ESP32 is a low-cost, low-power system-on-a-chip (SoC) with Wi-Fi and Bluetooth capabilities. It features a dual-core processor, making it ideal for various applications, from IoT devices to complex control systems. Popular in the maker community, it's easily programmable using Arduino IDE and other platforms. Its versatility and affordability make it a popular choice for connected projects. The ESP32 boasts a rich set of peripherals, including GPIO pins, analog-to-digital converters, and various communication interfaces. This makes it highly adaptable for interfacing with sensors, actuators, and other components in embedded systems. Its widespread use is supported by extensive documentation and community resources.

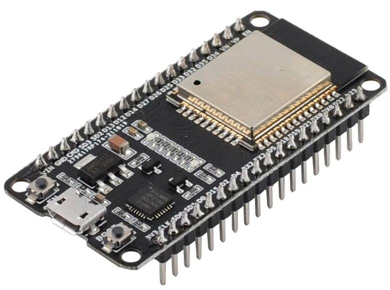

Figure 2.1: Esp 32

2.1.2 Esp 32 cam

The ESP32-CAM is a tiny camera module based on the ESP32 microcontroller. It integrates a 2MP OV2640 camera, making it ideal for image and video capturing in IoT projects. Besides the camera, it retains the ESP32's Wi-Fi and Bluetooth connectivity, enabling wireless image transmission and control. Its small size and low cost make it perfect for applications like image recognition, surveillance, and even streaming. It's easily programmable using the Arduino IDE, simplifying development.

Figure 2.2: Esp 32 cam

2.1.3 Li ion battery

Lithium-ion (Li-ion) batteries are rechargeable batteries widely used in portable electronics, electric vehicles, and power tools. They offer high energy density, meaning they can store a lot of energy in a small package, making them ideal for devices where size and weight are critical. Li-ion batteries have a relatively long lifespan, but their performance can degrade over time with repeated charging and discharging.

Figure 2.3: Li ion battery

2.1.4 Breadboard

A breadboard is a solderless prototyping platform used for building and testing electronic circuits. It features rows of interconnected holes, allowing components to be easily connected without soldering. This makes it ideal for experimentation and quick circuit development. Components are simply pushed into the holes, creating temporary connections. Breadboards are reusable, making them a cost-effective tool for learning and prototyping.

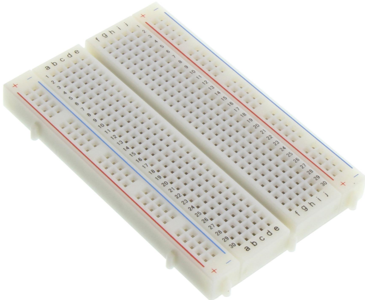

Figure 2.4: Breadboard

2.1.5 Buck Booster XL 6009

The XL6009 is a versatile DC-DC buck-boost converter module. It can both step down (buck) and step up (boost) input voltage, making it useful when the input voltage is either higher or lower than the desired output. It has an adjustable output voltage, allowing for flexibility in powering different devices. The XL6009 is commonly used in projects requiring a stable voltage supply, such as battery-powered systems, solar panel chargers, and LED drivers.

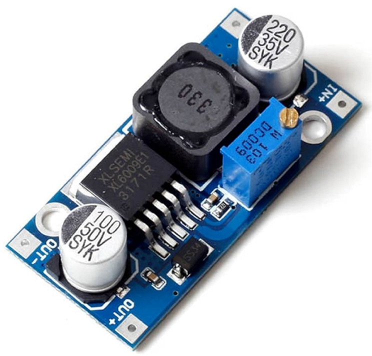

Figure 2.5: Buck Booster XL 6009

2.1.6 DTH 11

The DHT11 is a popular, low-cost digital sensor used for measuring both temperature and humidity. Its ability to provide both readings in a digital format makes it easy to interface with microcontrollers like Arduino, making it a favorite among hobbyists and professionals. The sensor requires only a single data pin for communication, further simplifying its integration into projects. Its affordability also contributes to its widespread use. While the DHT11 offers a convenient solution for environmental monitoring, it's important to be aware of its limitations.

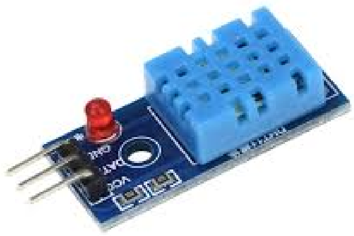

Figure 2.6: DTH 11

2.1.7 LDR

A Light Dependent Resistor (LDR), also known as a photoresistor, is a component whose resistance changes based on the amount of light it's exposed to. In bright light, its resistance decreases, allowing more current to flow. Conversely, in darkness, its resistance increases, restricting current flow. This characteristic makes LDRs useful for light-sensitive applications. They are commonly used in light-activated switches, automatic streetlights, and light-sensing circuits.

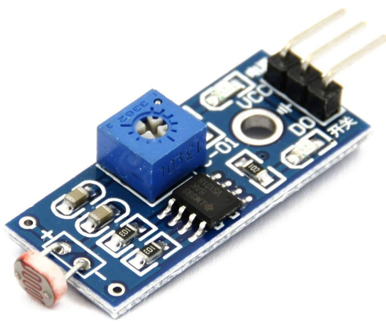

Figure 2.7: LDR

2.1.8 MQ-4 Gas Sensor

The MQ-4 gas sensor is designed to detect combustible gases, primarily methane (CH4), but it's also sensitive to other gases like propane, butane, and LPG. It's commonly used in gas leak detection systems and portable gas detectors. The sensor's output is an analog signal that varies with the gas concentration. A higher gas concentration results in a higher output voltage. This analog signal is then processed by a microcontroller or other circuitry to determine the gas level. While relatively inexpensive and easy to use, the MQ-4 isn't highly selective; it can react to other gases, leading to false positives. Calibration is also important for accurate measurements.

Figure 2.8: MQ-4 Gas Sensor

2.1.9 Soil Moisture Sensor

A soil moisture sensor measures the water content in soil. These sensors are crucial in agriculture, gardening, and environmental monitoring for efficient irrigation and preventing overwatering. There are two main types: resistive and capacitive. Resistive sensors measure the electrical resistance of the soil, which varies with moisture content. Capacitive sensors measure the soil's dielectric permittivity, also related to moisture levels.

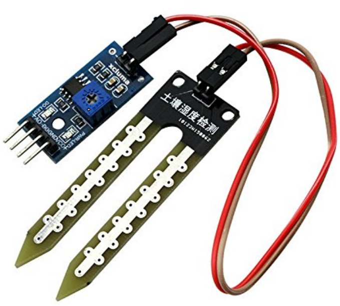

Figure 2.9: Soil Moisture Sensor

2.1.10 Motor Driver - L298N

The L298N is a popular dual H-bridge motor driver IC. It allows you to control the speed and direction of two DC motors simultaneously. It can also drive a single stepper motor. The L298N can handle relatively high currents and voltages, making it suitable for driving small to medium-sized motors. It requires a separate power supply for the motors and a control signal from a microcontroller.

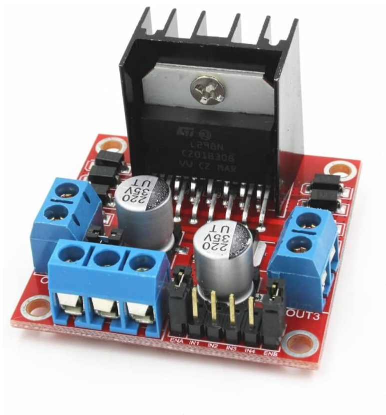

Figure 2.10: Motor driver l298N

2.1.11 Wheels

Wheels are fundamental components in countless applications, enabling movement and supporting loads. They typically consist of a circular rim and a tire, though variations exist. Different wheel types cater to specific needs, from small caster wheels for furniture to large pneumatic tires for vehicles. Wheel materials range from rubber and plastic for lighter applications to metal and specialized alloys for heavy-duty uses. Factors like wheel diameter, width, and tread pattern influence traction, load capacity, and rolling resistance.

Figure 2.11: Wheels

2.1.12 IR Sensor

An infrared (IR) sensor detects infrared light. These sensors are used in various applications, including remote controls, motion detectors, and line-following robots. IR sensors typically consist of an IR LED, which emits infrared light, and an IR receiver, which detects that light. When an object interrupts the IR beam or reflects it back to the receiver, the sensor detects a change. This change is then used to trigger an action, such as turning on a light or stopping a robot.

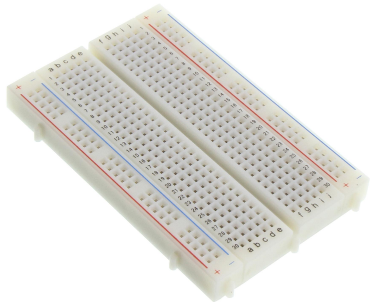

Figure 2.12: IR sensor

2.2 Software

2.2.1 Arduino IDE

The Arduino IDE is a cross-platform software application that simplifies programming Arduino boards. Its code editor features syntax highlighting and auto-completion, making code writing easier. The built-in compiler translates code into a format the Arduino understands, and the uploader sends it to the board via USB. The serial monitor facilitates communication between the Arduino and the computer, aiding debugging.

Figure 2.13: Arduino ide

2.2.2 Fusion 360

Fusion 360 is a cloud-based CAD/CAM software platform that combines 3D design, modeling, simulation, and manufacturing tools into a single integrated package. It's popular among hobbyists, makers, and professional engineers due to its powerful features and accessibility. Fusion 360 allows users to create complex 3D models, simulate their performance, and generate toolpaths for CNC machining or 3D printing.

Figure 2.14: Fusion 360

2.2.3 Fritzing

Fritzing is an open-source software tool that simplifies the creation of electronic circuit diagrams, especially for beginners and hobbyists. It provides a user-friendly interface with drag-and-drop components, allowing users to visually build circuits on a virtual breadboard, schematic, or PCB layout. Fritzing's focus is on visual representation, making it easier to understand and document circuits.

Figure 2.15: Fritzing

2.2.4 Blynk IOT

Blynk is a platform that allows you to easily control and monitor your hardware projects remotely using a smartphone app. It provides a simple way to create a graphical interface for your devices without needing extensive coding knowledge. Blynk supports various hardware platforms like Arduino, ESP32, and Raspberry Pi. You can create custom dashboards with buttons, sliders, graphs, and other widgets to interact with your devices. Data from your devices can be visualized and stored within the Blynk platform.

Figure 2.16: Blynk IOT

Chapter 3

Implementation and working

3.1 Circuit diagram

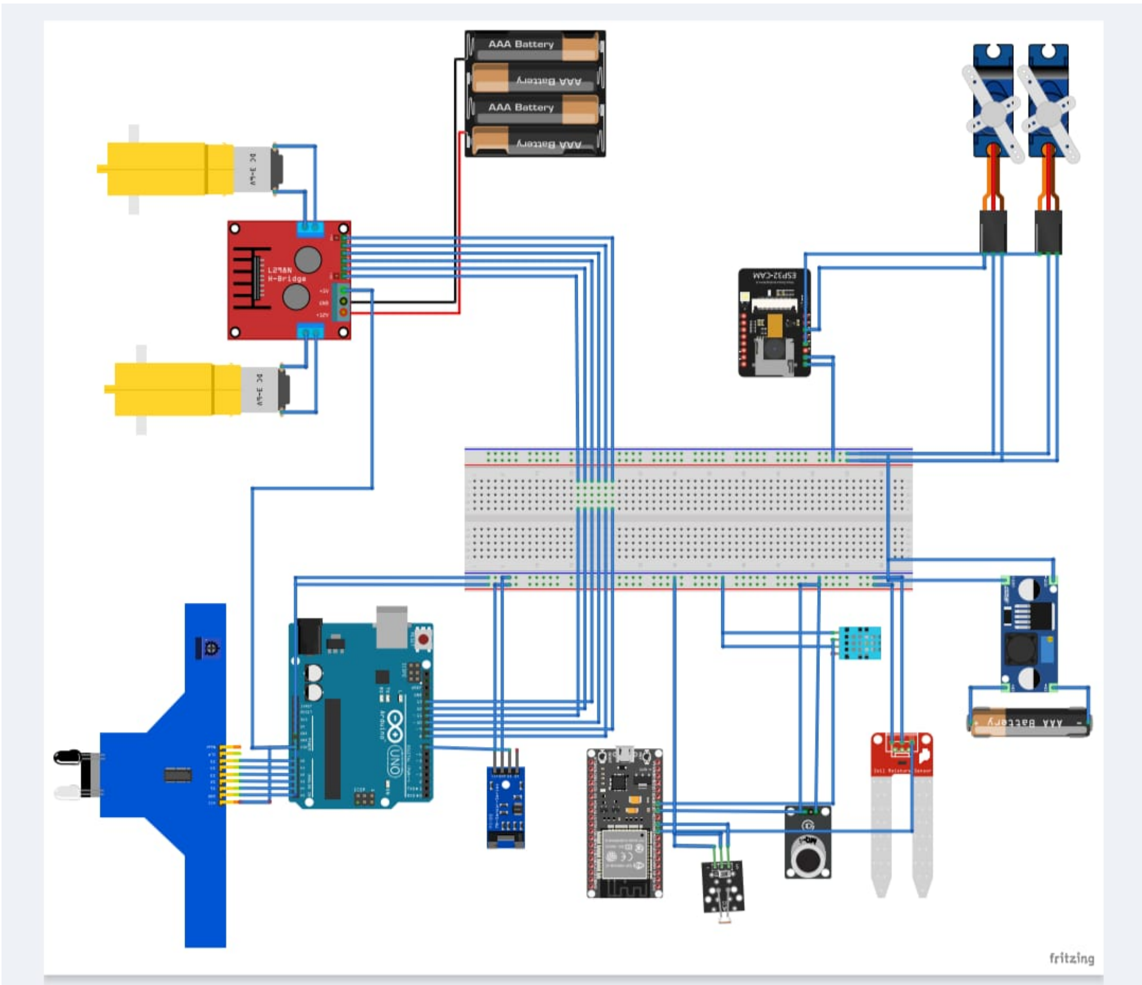

Figure 3.1: Circuit Diagram

3.2 Block diagram

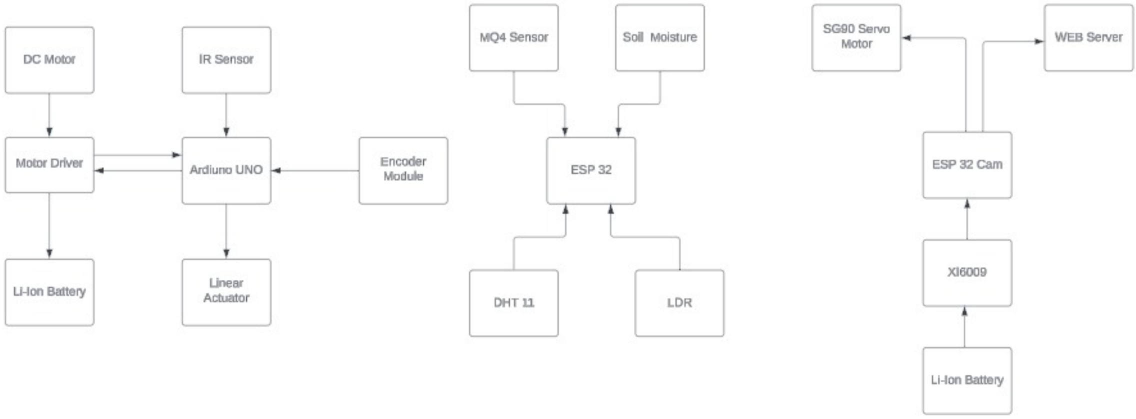

Figure 3.2: Block diagram

3.3 Working Mechanism

This smart agriculture bot automates several crucial farming tasks, reducing manual labor and improving efficiency. Built as a line-following robot, it navigates fields autonomously, guided by pre-defined paths. Equipped with a suite of sensors, the bot gathers critical data about the environment and crop conditions. A moisture sensor, likely a capacitive type, measures soil water content, informing irrigation decisions. A gas sensor, such as the MQ-4, detects gases emitted by fruit, allowing it to assess ripeness, crucial for harvest timing. An IR sensor enables the bot to follow the designated lines and detect obstacles in its path. Upon encountering an obstacle, a linear actuator, controlled by the Arduino board, can be used for tasks like adjusting the height of the moisture sensor for optimal readings at various soil depths. An Arduino board serves as the central processing unit, integrating data from all sensors and controlling the bot's movements and actions. By automating tasks like soil moisture monitoring and fruit ripeness assessment, this bot offers a significant advantage over traditional farming methods, potentially replacing the need for numerous human workers and promoting more efficient and data-driven agricultural practices.

Chapter 4

Experimental Results and Conclusions

4.1 Results

The smart agriculture line-following bot offers a significant reduction in manual labor across various agricultural tasks. This automated approach provides farmers with valuable insights into their crops and environment. The bot gathers crucial data, including soil moisture and humidity levels at specific locations within the field. This localized information allows for targeted irrigation, optimizing water usage and promoting healthy plant growth. An integrated ESP32-CAM provides a wide-angle view, capturing a 180-degree perspective of the plants. This visual data can be used for remote monitoring and potentially for identifying plant health issues. The Blynk app further enhances accessibility by transmitting real-time data from the bot directly to the farmer's smartphone. This allows for remote monitoring of soil conditions, plant health, and other crucial parameters, enabling timely intervention when necessary. Finally, the line-following capability, combined with the bot's ability to detect black objects, ensures efficient navigation along designated paths, automating routine tasks and freeing up valuable time for farmers to focus on other critical aspects of farm management.

4.2 Future Enhancement

Enhancing a smart agriculture line-following bot involves integrating advanced technologies to boost efficiency and adaptability. A key area for improvement is AI and machine learning integration. Adaptive learning allows the bot to adjust to varying environments and crop types. Machine learning algorithms can analyze real-time data from sensors and cameras to optimize the bot's path, task execution, and resource allocation, ensuring efficient field coverage and targeted actions. Predictive analytics takes this a step further. By leveraging AI, the bot can analyze historical data, weather patterns, and sensor readings to predict potential crop health issues, changing soil conditions, and the impact of weather events. This predictive capability enables farmers to take proactive measures, such as adjusting irrigation schedules, applying targeted treatments, or even preparing for adverse weather, ultimately leading to improved crop yields and reduced losses. These AI-driven enhancements transform the bot from a simple automation tool into a sophisticated agricultural assistant.

4.3 Conclusion

In conclusion, the future of smart agriculture line-following bots holds tremendous potential to revolutionize modern farming practices. By integrating advanced technologies such as AI, machine learning, multi-spectral imaging, and IOT connectivity, these bots can significantly enhance precision, efficiency, and sustainability in agriculture. Future enhancements focusing on adaptive learning, autonomous navigation, all-terrain capabilities, and collaborative robotics will further empower farmers to optimize crop management, reduce resource usage, and increase productivity. As these innovations continue to evolve, smart agriculture line following bots will play a pivotal role in addressing the global challenges of food security, environmental sustainability, and agricultural efficiency, ushering in a new era of intelligent farming.

4.4 References

1] International Research journal of Engineering and technology(IRJET),Robot to The Smart Agriculture Line Fol- lowing Bot.

2]IEEE Papers The Smart Agriculture Line Following Bot.

4.5 Cad Model

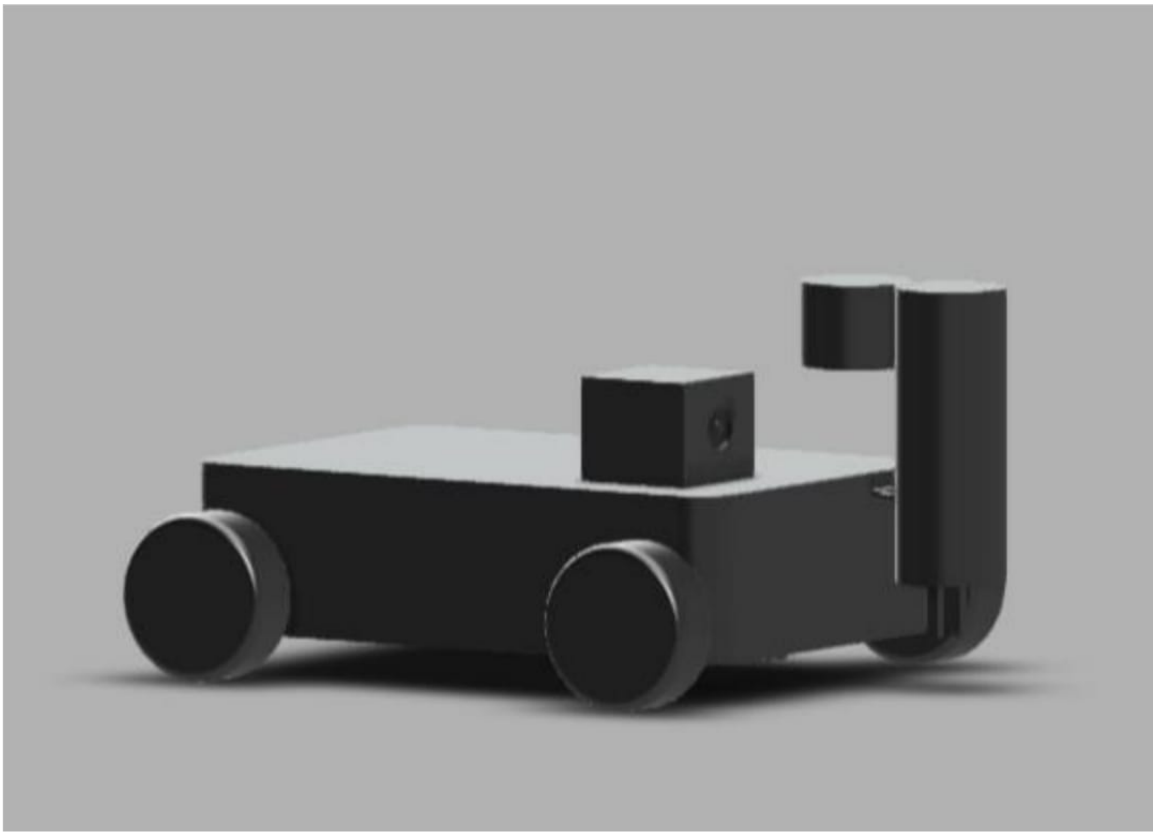

Figure 4.1: Cad Model

Chapter 5

Source Code

```
    #include<DHT.h>
DHT dht11(DHT11_PIN, DHT11);
#define AOUT_PIN 13
#define DHT11_PIN  12
#define LDRSensor  14
const int gasSensorPin = 27;
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
char auth[] = "YourAuthToken";
char ssid[] = "YourSSID";
char pass[] = "YourWiFiPassword";
void setup() {
  Blynk.begin(auth, ssid, pass);
  Serial.begin(115200);
  dht11.begin();
  pinMode (LDRSensor, INPUT);
}
void loop() {
  Blynk.run();
  int value = analogRead(AOUT_PIN);
```

```
Serial.print("Moisture value: ");
Serial.println(value);
Blynk.virtualWrite(V0, value);
delay(5000);
float humi  = dht11.readHumidity();
float tempC = dht11.readTemperature();
if ( isnan(tempC) || isnan(humi)) {
  Serial.println("Failed to read from DHT11 sensor!");
}
else {
  Serial.print("Humidity: ");
  Serial.print(humi);
  Serial.print("%");
  Serial.print("  |  ");
  Serial.print("Temperature: ");
  Serial.print(tempC);
  Serial.print("°C  ~  ");
}
Blynk.virtualWrite(V1, humi);
Blynk.virtualWrite(V2, tempC);
delay(5000);
int sensorValue = analogRead(gasSensorPin);
Serial.print("Gas Sensor Value: ");
Serial.println(sensorValue);
Blynk.virtualWrite(V3, sensorValue);
delay(5000);
int Sensordata = digitalRead (LDRSensor);
Serial.print("Sensor value:");
Serial.println(Sensordata);
Blynk.virtualWrite(V4, Sensordata);
```

```
  delay(5000);


}
#define m1 4
#define m2 5
#define m3 2
#define m4 3
#define e1 9
#define e2 10
#define ir1 A0
#define ir2 A1
#define ir3 A2
#define ir4 A3
#define ir5 A4
void setup() {
  pinMode(m1, OUTPUT);
  pinMode(m2, OUTPUT);
  pinMode(m3, OUTPUT);
  pinMode(m4, OUTPUT);
  pinMode(e1, OUTPUT);
  pinMode(e2, OUTPUT);
  pinMode(ir1, INPUT);
  pinMode(ir2, INPUT);
  pinMode(ir3, INPUT);
  pinMode(ir4, INPUT);
  pinMode(ir5, INPUT);
}
void loop() {
  //Reading Sensor Values
  int s1 = digitalRead(ir1);
```

```
int s2 = digitalRead(ir2);
int s3 = digitalRead(ir3);
int s4 = digitalRead(ir4);
int s5 = digitalRead(ir5);
if((s1 == 1) && (s2 == 1) && (s3 == 0) && (s4 == 1) && (s5 == 1))
{
  analogWrite(e1, 255);
  analogWrite(e2, 255);
  digitalWrite(m1, HIGH);
  digitalWrite(m2, LOW);
  digitalWrite(m3, HIGH);
  digitalWrite(m4, LOW);
}
if((s1 == 1) && (s2 == 0) && (s3 == 1) && (s4 == 1) && (s5 == 1))
{
  analogWrite(e1, 255);
  analogWrite(e2, 255);
  digitalWrite(m1, HIGH);
  digitalWrite(m2, LOW);
  digitalWrite(m3, LOW);
  digitalWrite(m4, LOW);
}
if((s1 == 0) && (s2 == 1) && (s3 == 1) && (s4 == 1) && (s5 == 1))
{
  //going right with full speed
  analogWrite(e1, 255);
  analogWrite(e2, 255);
  digitalWrite(m1, HIGH);
  digitalWrite(m2, LOW);
  digitalWrite(m3, LOW);
```

```
  digitalWrite(m4, HIGH);
}
if((s1 == 1) && (s2 == 1) && (s3 == 1) && (s4 == 0) && (s5 == 1))
{
  analogWrite(e1, 255);
  analogWrite(e2, 255);
  digitalWrite(m1, LOW);
  digitalWrite(m2, LOW);
  digitalWrite(m3, HIGH);
  digitalWrite(m4, LOW);
}
if((s1 == 1) && (s2 == 1) && (s3 == 1) && (s4 == 1) && (s5 == 0))
{
  analogWrite(e1, 255);
  analogWrite(e2, 255);
  digitalWrite(m1, LOW);
  digitalWrite(m2, HIGH);
  digitalWrite(m3, HIGH);
  digitalWrite(m4, LOW);
}
if((s1 == 1) && (s2 == 1) && (s3 == 0) && (s4 == 0) && (s5 == 1))
{
  analogWrite(e1, 255);
  analogWrite(e2, 255);
  digitalWrite(m1, LOW);
  digitalWrite(m2, LOW);
  digitalWrite(m3, HIGH);
  digitalWrite(m4, LOW);
}
if((s1 == 1) && (s2 == 0) && (s3 == 0) && (s4 == 1) && (s5 == 1))
```

```
{
  analogWrite(e1, 255);
  analogWrite(e2, 255);
  digitalWrite(m1, HIGH);
  digitalWrite(m2, LOW);
  digitalWrite(m3, LOW);
  digitalWrite(m4, LOW);
}
if((s1 == 0) && (s2 == 0) && (s3 == 0) && (s4 == 1) && (s5 == 1))
{
  analogWrite(e1, 255);
  analogWrite(e2, 255);
  digitalWrite(m1, HIGH);
  digitalWrite(m2, LOW);
  digitalWrite(m3, LOW);
  digitalWrite(m4, LOW);
}
if((s1 == 1) && (s2 == 1) && (s3 == 0) && (s4 == 0) && (s5 == 0))
{
  //going left with full speed
  analogWrite(e1, 255);
  analogWrite(e2, 255);
  digitalWrite(m1, LOW);
  digitalWrite(m2, LOW);
  digitalWrite(m3, HIGH);
  digitalWrite(m4, LOW);
}
if((s1 == 0) && (s2 == 0) && (s3 == 0) && (s4 == 0) && (s5 == 0))
{
  digitalWrite(m1, LOW);
```

```
    digitalWrite(m2, LOW);
    digitalWrite(m3, LOW);
    digitalWrite(m4, LOW);
  }
}
```