An-Najah National University

Faculty of Engineering and Information Technology

Computer Engineering Department

Graduation Project

Smart agricultural robot

Student's teams:

Supervisor:

This project is submitted in partial fulfillment of the requirements for a Bachelor's degree in Computer Engineering.

2025 /2026

Table of Contents

List of Figures:

List of Tables:

Acknowledgment

In the name of Allah, the Most Gracious and Most Merciful, who granted us the strength, knowledge, and perseverance to complete this project. We extend our heartfelt gratitude to those who have inspired and supported us throughout this journey: our families, friends, and our supervisor.

Our heartfelt gratitude goes to our supervisor, whose continuous guidance, thoughtful feedback, and constant encouragement were instrumental in overcoming challenges and reaching our goals.

We are deeply thankful to our families for their unwavering support, patience, and faith in us, which have been a source of strength throughout this work.

We also thank our friends and peers for their collaboration,

motivation, and assistance.

Finally, we extend our appreciation to all who contributed to the success

of this project in any capacity. Your efforts, whether big or small,

truly made a difference.

Abstract

The smart agricultural robot aims to assist farmers in tilling their land. We designed and implemented this robot to increase agricultural production, improve crop quality, reduce human intervention, and lower operational costs. Our project integrates electronic and mechanical technologies with programming to automatically control tilling, planting, and irrigation processes.

The robot tills the soil using a track mounted at its rear and plants seeds in a systematic manner, promoting plant growth. Additionally the robot measures soil moisture using humidity sensors and automatically activates water pumps when the moisture level falls below the required threshold. This helps maintain optimal conditions for crop growth and prevents drought.

One of the most important components of the agricultural robot is a local web server, which allows us to control the robot's movement (forward, backward, right, left). The server also controls the plow's raising and lowering, as well as the water pump, which can be operated automatically (based on the humidity sensor) or manually. Furthermore, the server allows us to control the planting of seeds by opening and closing the servo.

Chapter 1: Introduction

1.1 Preface

Our project aims to design and develop an intelligent agricultural robot capable of increasing agricultural production and reducing operational costs by improving crop quality and increasing productivity. The robot utilizes advanced technologies to perform several tasks in the field including plowing the land using a rail system mounted on its rear. It also plants seeds systematically ensuring optimal seed distribution.

Furthermore the robot monitors soil moisture using a sensor that measures the moisture content. When the moisture level drops below the required threshold the system activates water pumps to maintain high soil moisture preventing drought and ensuring ideal conditions for plant growth.

In addition, the robot can be controlled via a Local Web Server operating on the ESP32 controller. A local web server network is created on the controller and an IP address is assigned that can be accessed from any nearby phone or computer. By entering this address into the browser an interactive web interface is opened, allowing control of the robot's movement and operation of tillage and farming systems. This mechanism provides a reliable and easy way to control the robot in the field without the need for an internet connection or additional equipment which enhances operational flexibility in different agricultural environments.

1.2 Problem Statement

Farmers worldwide face increasing challenges related to declining labor rising global food demand and environmental pressures that threaten the sustainability of traditional agriculture. The concept of farming systems was once limited to massive mechanical machines like tractors and traditional harvesters. However over time these systems have evolved from mere machines to sensors and data analysis. This is where artificial intelligence has entered the agricultural landscape transforming it from a purely machine based system to one driven by precise data.

Therefore there is a pressing need to develop an intelligent system capable of performing essential agricultural tasks accurately and efficiently while monitoring and reacting to farm environmental conditions in real time. This system aims to address these challenges by designing an agricultural robot capable of plowing planting seeds monitoring soil moisture and automatically operating water pumps. It also provides easy control via a local web server running on an ESP32 with a dedicated IP address. This system seeks to reduce human effort and resource waste and improve the quality of agricultural production.

1.3 Project Aims and Objectives:

The importance of agricultural robots lies in increasing productivity reducing costs improving crop quality and enhancing sustainability by automating precise agricultural tasks such as planting irrigation and harvesting. They also help address labor shortages and conserve resources like water and fertilizer more effectively.

The agricultural robot achieves several objectives including:

1. Developing an agricultural robot capable of moving within farmland and plowing it using a suitable track.
2. The ability to plant seeds automatically and precisely in the soil.
3. Using sensors to continuously measure soil moisture to ensure the soil does not dry out.
4. Implementing an irrigation system for crops where pumps activate when moisture falls below a predetermined level.
5. Improving the quality and efficiency of agricultural production by reducing labor and increasing the accuracy of task execution.
6. Developing a control interface based on a local web server using ESP32 allowing the user to control the robot's movement and operate various systems via an IP address.

1.4 Project Requirements

1.4.1 Functional Requirements

The system must meet the following functional requirements to ensure the agricultural robot performs as required:

1. The robot must plant seeds regularly and accurately at predefined spacing's.
2. The robot must measure soil moisture using its dedicated moisture sensor.
3. The water pumps must activate when soil moisture falls below the required level to prevent the soil from drying out.
4. The pumps should stop operating when the moisture level reaches the appropriate level.
5. The user must be able to control the robot via the ESP32 controller's IP address.
6. The robot must stop moving in the event of an error or malfunction.
7. The robot must automatically stop plowing and lift the plowshare off the ground upon completion of the plowing process.

1.4.2 Non-Functional Requirements

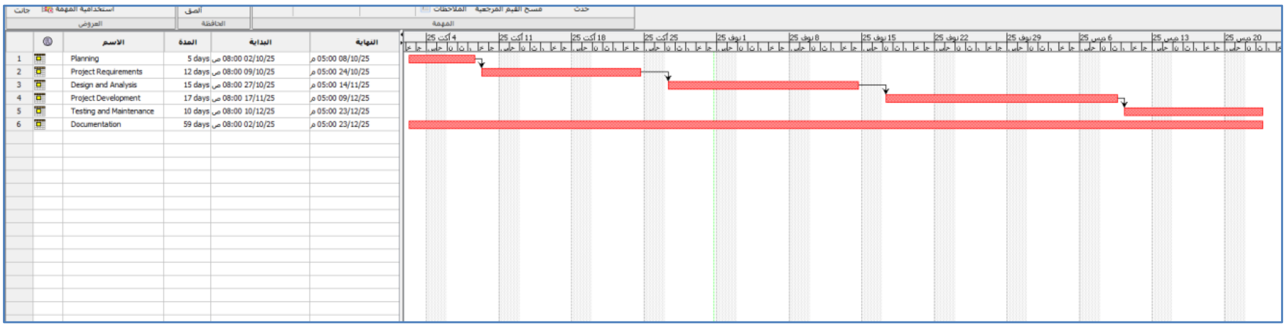
1. Performance: The agricultural robot must operate correctly and accurately without any performance errors.
2. Scalability: The agricultural robot must be able to be used on a wide range of farmland without any performance issues.
3. Security: All data must be encrypted during transmission and storage to ensure privacy.
4. Reliability: The system must operate consistently and accurately without errors under various environmental conditions.
5. Ease of Maintenance: Frequent updates, additions, and modifications must be allowed.
6. Ease of Use: The local web server must be simple and easy to use for both educated and uneducated users.

1.5 Project Schedule

Table 1: Project Schedule

العروض		الحافظة			
		الاسم	المدة	البداية	النهاية
1		Planning	5 days	ص 08:00 02/10/25	م 05:00 08/10/25
2		Project Requirements	12 days	ص 08:00 09/10/25	م 05:00 24/10/25
3		Design and Analysis	15 days	ص 08:00 27/10/25	م 05:00 14/11/25
4		Project Development	17 days	ص 08:00 17/11/25	م 05:00 09/12/25
5		Testing and Maintenance	10 days	ص 08:00 10/12/25	م 05:00 23/12/25
6		Documentation	59 days	ص 08:00 02/10/25	م 05:00 23/12/25

Table 2: Project Gannet



CHAPTER 2: Literature Review

2. 1 Literature Review

2.1.1 Agricultural robots—system analysis and economic feasibility

This paper focuses on the economic feasibility of applying autonomous robotic vehicles compared to conventional systems in three different applications: robotic weeding in high value crops (particularly sugar beet), crop scouting in cereals and grass cutting on golf courses. The comparison was based on a systems analysis and an individual economic feasibility study for each of the three applications. The results showed that in all three scenarios, the robotic applications are more economically feasible than the conventional systems. The high cost of real time kinematics Global Positioning System (RTK-GPS) and the small capacity of the vehicles are the main parameters that increase the cost of the robotic systems.[1]

2.1.2 Agricultural Robotics for Field Operations

Modern agriculture is related to a revolution that occurred in a large group of technologies (e.g., informatics, sensors, navigation) within the last decades. In crop production systems, there are field operations that are quite labour-intensive either due to their complexity or because of the fact that they are connected to sensitive plants/edible product interaction, or because of the repetitiveness they require throughout a crop production cycle. These are the key factors for the development of agricultural robots. In this paper, a systematic review of the literature has been conducted on research and commercial agricultural robotics used in crop field operations. This study underlined that the most explored robotic systems were related to harvesting and weeding, while the less studied were the disease detection and seeding robots. The optimization and further development of agricultural robotics are vital, and should be evolved by producing faster processing algorithms, better communication between the robotic platforms and the implements, and advanced sensing systems.[2]

2.1.3 Development status and trend of agricultural robot technology

In the face of the contradiction between the increasing demand for agricultural products and the sharp reduction of agricultural resources and labor force, agricultural robot technology is developing explosively on the basis of decades of technical and industrial exploration. In view of the complexity and particularity of the development of agricultural robot technology, it is of great value to summarize its development characteristics and make reasonable judgments on its development trend. In this paper, the type of agricultural robot systems was first discussed. From the classification of agricultural robot systems, the development of main types of monitoring robots, non-selective and selective working robots for crop farming, livestock and poultry farming and aquaculture were introduced in detail. Then the scientific research, core technology, and commercialization of different types of agricultural robots were summarized. It is believed that navigation in complex agricultural environments, damage-free robot-crop interaction, and agronomy-robot fusion has high scientific value and significance to promote the revolutionary advances in agricultural robot technology. The characteristics of inter-discipline between agricultural robot technology and new materials, artificial intelligence, bionics, agronomy are research focus. The fast damage-free operation, autonomous navigation for complex environments, target detection for complex backgrounds, and special design for agricultural robots are considered to be the key technology of agricultural robot development, and the development path is given. Finally, robot-crop interaction simulation, big data support, and artificial intelligence are regarded as paths to realize the breakthrough of key agricultural robot technologies.[3]

CHAPTER 3: Methodology

3.1 Hardware System Components

This section describes all hardware used in our project, it presents a figure for each one with short description about its work principle and why it is used in the system.

✓ ESP32 microcontroller

ESP32 is a series of low-cost, low-power system on a chip microcontrollers with integrated Wi-Fi and dual-mode Bluetooth. The ESP32 series employs either a Tensilica Xtensa LX6 microprocessor in both dual-core and single-core variations, Xtensa LX7 dual-core microprocessor or a single-core RISC-V microprocessor and includes built-in antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power-management modules..It is a successor to the ESP32 microcontroller.

Features of the ESP32 include the following:

1. CPU: Xtensa dual-core (or single-core) 32-bit LX6 microprocessor operating at 160 or 240 MHz and performing at up to 600 DMIPS.
2. Ultra low power (ULP) co-processor.
3. Memory: 320 KiB RAM, 448 KiB ROM.
4. Wireless connectivity.
5. Wi-Fi: 802.11 b/g/n.

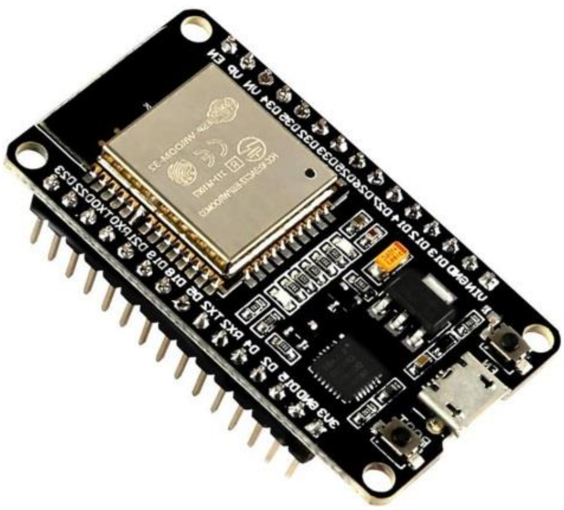


Figure 1:ESP32 Microcontroller[4]

✓ **Relay 4 channel**

The Relay Board can be used to turn lights, fans and other devices on/off while keeping them isolated from your microcontroller. Relay's coil voltage is 5V, so you can use directly with any 5V regulator. (We suggest separate regulators between your Arduino and Relay board) .This Relay Board allows you to control high-power devices (up to 10 A) via the on-board relay. Control of the relay is provided via a 1 x 3 header – friendly to servo cables and convenient to connect to many development boards.

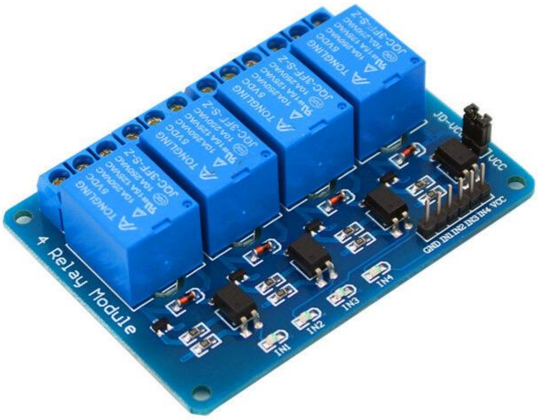


Figure 2:Relay 4 cchannel[5]

✓ **12V Battery pack 7000mAh**

These battery packs consist of two 11.1V lithium-ion cells connected in series, delivering a total voltage range of 11.1V~12.6V. With their compact size and high energy density, these battery packs are suitable for a wide range of applications, including RC vehicles, drones, portable electronics, and more. We offer a variety of capacity options to cater to different power requirements. Experience long-lasting performance and seamless power delivery with our 3S lithium-ion battery packs for all your electronic needs.

Figure 3: 12V Battery pack 7000mAh[6]

✓ MG996R 360 Servo Motor

A servo motor is an electric motor used in many applications where precise control of a machine's position or rotation angle is required. It is commonly used in motion control systems for robots, drones, remote-controlled cars, 3D printers, and CNC machines. The main advantage of a servo motor is its ability to provide precise position control. Using a feedback loop, a servo motor can adjust its position in real time to accurately follow a command signal. Servo motors are also known for their high energy efficiency, using power only when position adjustment is needed.

The MG996R servo motor is a type of digital servo motor and is considered one of the best servo motors available on the market. This motor boasts several features, including:

1. 360-degree rotation capability.
2. High torque capacity up to 10 kg·cm.
3. High positioning accuracy.
4. High rotation speed, ranging from 0.17 seconds per 60-degree rotation. 5. It can be controlled via PWM (Pulse Width Modulation) signal, which offers precise and fast control.

MG996R servo motors are used in many applications requiring full-angle rotation, such as robotics, small vehicles, control devices, hobby projects, and educational applications.

Figure 4: MG996R servo motor[7]

✓ Wheel DC Motor

New wheels, hollow, middle muster, shock effect, pattern beautiful, deep engraving effect.

A DC motors with gear, single axis, with DC 3–6V operating voltage and an RPM of 100R / minute. This gear box is ideal for robotic car or line-tracing robot. With plastic construction and colored in bright yellow, the DC gear motor measures approx. 6.5cm long, 2.5cm wide and 1.8cm thick.

Motor Detailed parameters:

1. No load speed (6V): 100RPM
2. No load current (6V): 60mA
3. Motor speed: 100 r/min
4. Motor Size: approx. 6.5x2.2x1.8cm
5. Voltage: 6VA

Figure 5: Wheel DC Motor[8]

✓ **Converter Board PCB XY-3606**

The XY-3606 Converter Board is a widely used DC-DC buck converter designed to efficiently step down higher input voltages to a stable low-voltage output, typically 5 V. This module supports an input range from 9 V to 36 V and provides a regulated output capable of supplying currents up to 6 A, making it suitable for powering microcontrollers, sensors, USB-powered devices, and various electronic circuits. The board operates using synchronous rectification technology, which significantly increases efficiency and reduces heat generation compared to traditional linear regulators.

It also includes both USB and terminal outputs, allowing for easy integration into different electronic systems. Due to its compact size, high efficiency, and stable performance, the XY-3606 is commonly used in automotive applications, portable power systems, robotics, renewable-energy projects, and DIY electronics. These features make it an essential component for systems requiring flexible, reliable, and efficient voltage conversion.

Figure 6: Converter Board PCB XY-3606[9]

✓ Soil Moisture Hygrometer Detection Humidity Sensor

The Soil Moisture Hygrometer Detection Humidity Sensor is an electronic module used to measure the water content within soil, making it essential for agricultural automation, smart irrigation systems, and environmental monitoring. The sensor typically consists of two conductive probes that are inserted into the soil; these probes measure the soil's conductivity, which varies based on moisture levels. When the soil is wet, conductivity increases, resulting in a lower output resistance, while dry soil produces higher resistance.

The sensor module often includes an analog output for precise moisture level readings and a digital output that triggers at a user-defined threshold, enabling automated watering systems. Due to its simplicity, low cost, and effectiveness, the soil moisture sensor is widely used in plant monitoring projects, greenhouse systems, and IoT-based smart farming applications to help optimize water usage and maintain ideal growing conditions.

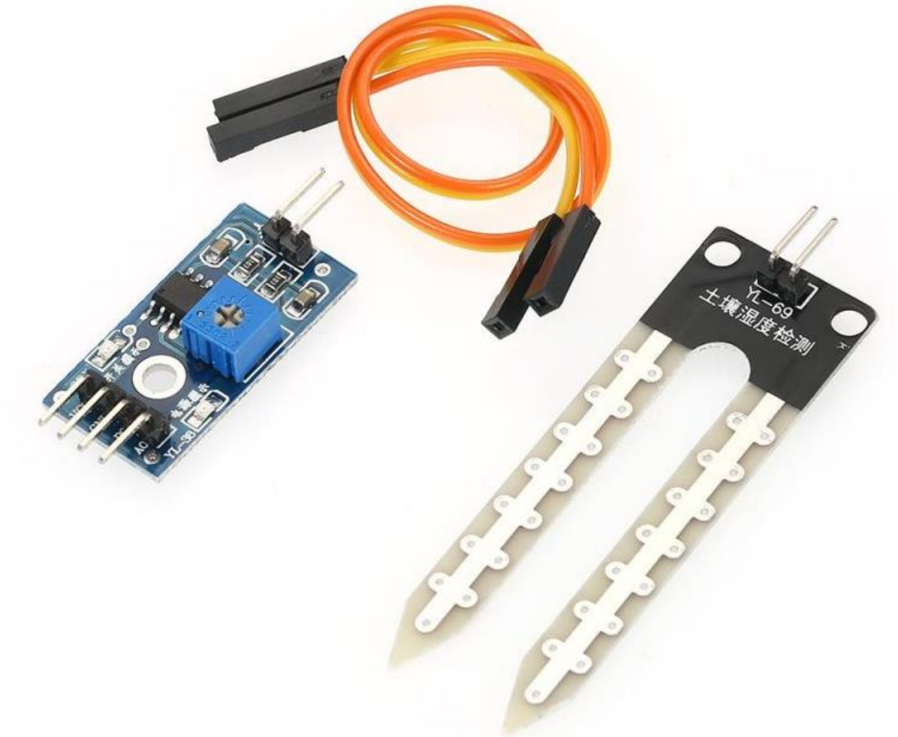


Figure 7: Soil Moisture Hygrometer[10]

✓ Motor Dosing Liquid Water Squeeze Pump

The Motor Dosing Liquid Water Squeeze Pump is a compact, low-power air pump designed to provide continuous aeration for small aquariums, hydroponic systems, and water-based experiments. Its primary function is to pump air into the water, increasing oxygen levels and improving the environment for fish, plants, and microorganisms. The pump typically uses a small diaphragm or motor-driven mechanism to generate airflow while operating quietly and efficiently.

Figure 8: Motor Dosing Liquid Water Squeeze[11]

3.2 System Software Component

This section will provide some information about the main programs used in our project.

3.2.1 Arduino IDE

Arduino code is written in C+ with an addition of special methods and functions. C++ is a human-readable programming language. The Arduino Integrated Development Environment (IDE) is the main text editing program used for Arduino programming. It is where you'll be typing up your code before uploading it to the board you want to program. Arduino code is referred to as sketches, It is processed and compiled to machine language, and Arduino sketch code is used to program wemos microcontrollers.

Figure 9:Arduino IDE logo[12]

3.2.2 Fritizing

It is an open source initiative to support designers, artists, researchers and hobbyists to take the step from physical prototypes to actual product. We make this software in the spirit of processing and Arduino, developing a tool that allows users to document their Arduino and other electronic models, and create a PCB layout for manufacturing. The complementary website helps users share and discuss drafts and experiences as well as reduce manufacturing costs. Fritizing is essentially electronic design automation software with a low barrier to entry, suitable for the needs of designers and artists. It uses the metaphor of a breadboard, so that it is easy to transfer your hardware drawing to the software.

Figure 10: Fritzing program logo[13]

3.2.3 Visual studio code and PlatformIO IDE

Visual Studio Code (VS Code) is a powerful, lightweight, and open-source code editor developed by Microsoft, widely used for software development across multiple programming languages. It provides intelligent code completion, integrated debugging tools, version control support through Git, and a rich ecosystem of extensions that enhance productivity. One of the most important extensions for embedded systems development is PlatformIO IDE.

PlatformIO is an advanced; cross-platform development environment designed specifically for IoT and embedded applications. It supports hundreds of microcontroller platforms, including Arduino, ESP32, STM32, and Raspberry Pi Pico, and offers features such as automated library management, project configuration, firmware uploading, and serial monitoring. When integrated with VS Code, PlatformIO provides a unified and highly efficient workspace that allows developers to write, compile, and deploy firmware seamlessly. This combination enables a professional workflow for building modern embedded systems and significantly improves development speed, reliability, and scalability.

Figure 11: Visual Studio Code logo[14]

CHAPTER 4: SYSTEM DESIGN

4.1 Preface

This chapter will present the system design for an agricultural robot showcasing various design options. We will provide a conceptual description of the system discuss the algorithms and methodologies used and present schematic diagrams illustrating the system's components and connections and this chapter serves as a comprehensive guide to the project design phase.

4.2 Conceptual system description (block diagram)

The following figure shows the block diagram of the agricultural robot, where the system relies on an ESP32 controller unit that is connected to a number of sensors and devices to collect and manage data to control the robot. The diagram includes a soil sensor that measures soil moisture and sends readings to the controller to give commands to operate the water pump, in addition to relays that control the movement of the wheels. Commands are also sent to the servo to control the track that plows the land. The robot and the commands for movement and stopping are controlled from a local web server.

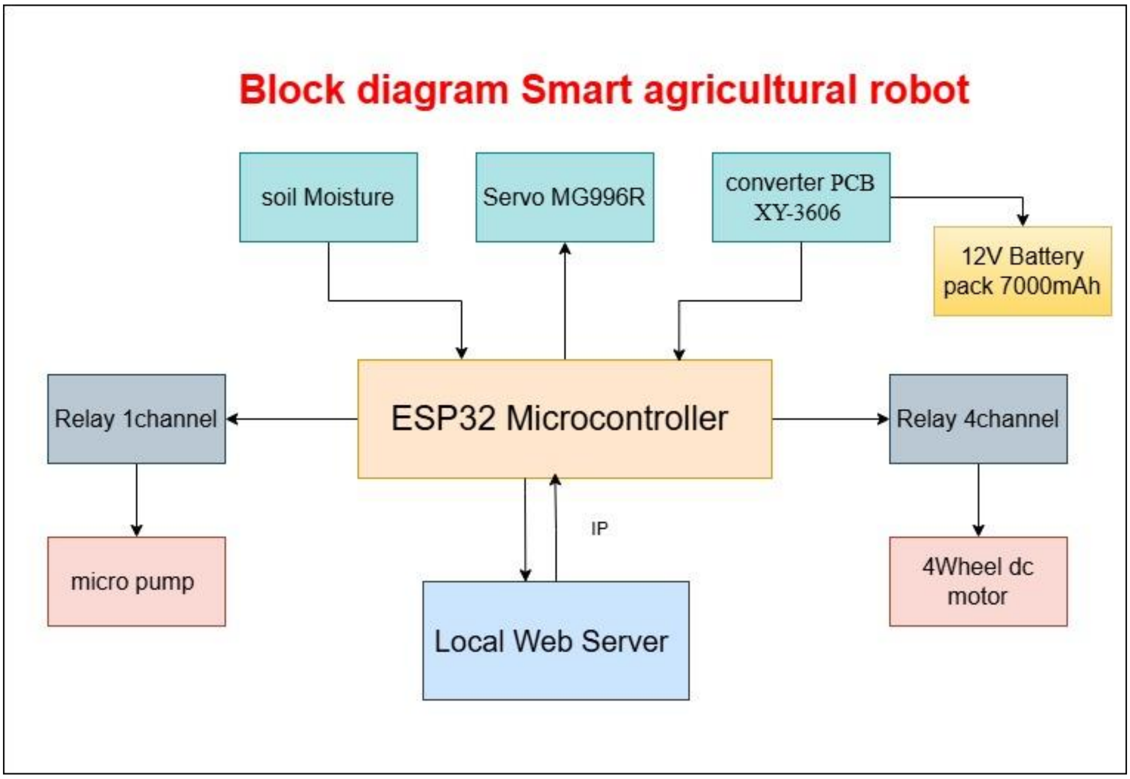


Figure 12: Block diagram

4.3 Agricultural Robot Prototype

The following figure illustrates a scaled-down prototype of the smart agricultural robot proposed in this project. It shows the robot's overall structure mounted on a wooden platform equipped with wheels for movement, as well as the electronic units mounted on top, such as the ESP32 microcontroller, relay modules, and power supply. The integrated irrigation system, including the water pump and connecting pipes, is also shown, along with the tillage and planting mechanism mounted on the rear of the robot.

Figure 13: Agricultural Robot Prototype

4.4 Wiring Diagrams

In this section some component is shown with its connection the of the ESP32 microcontroller the pins of each equipment will be connected physically as shown in the following figures.

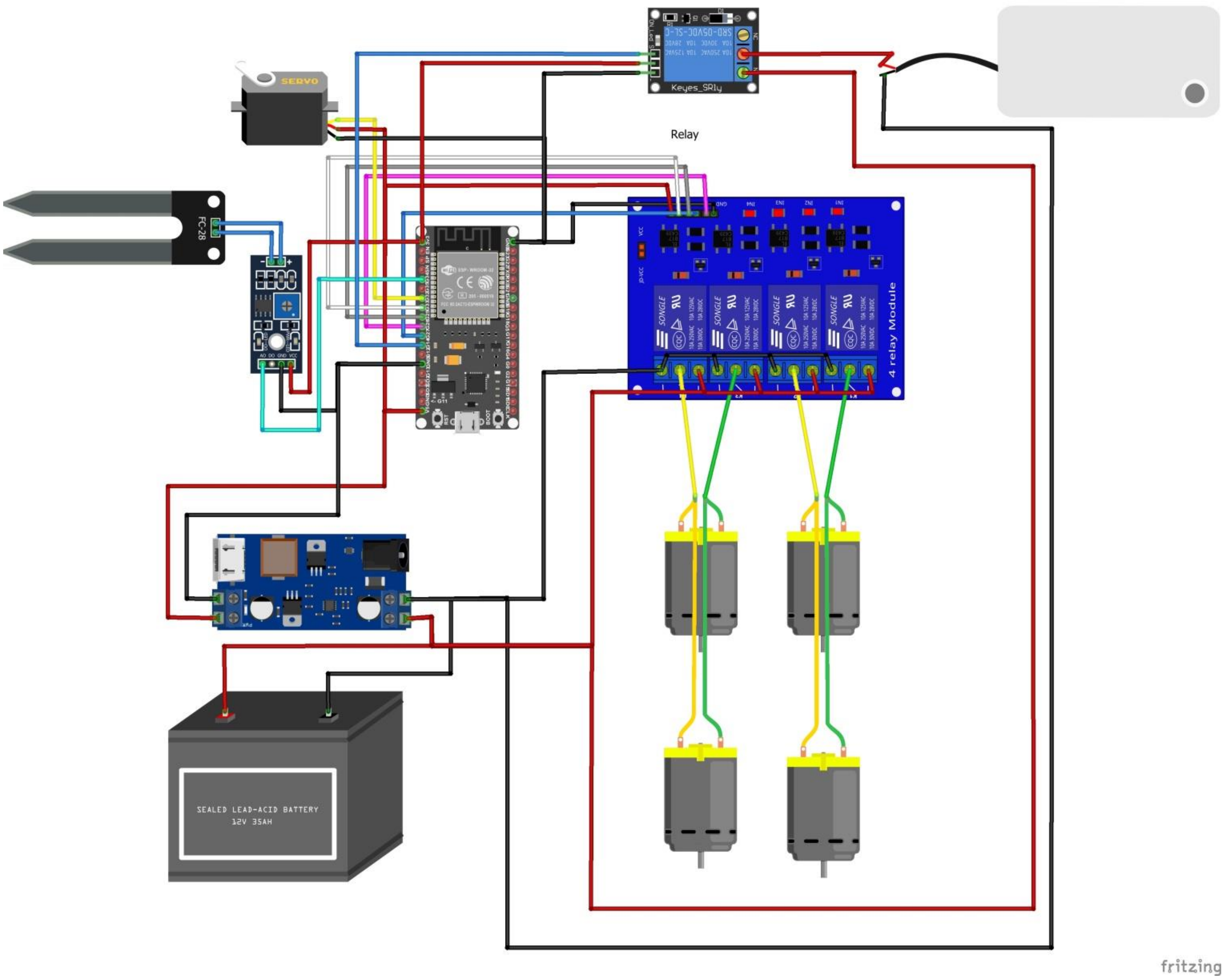


Figure 14: Wiring Diagrams

4.5 Schematic Diagrams

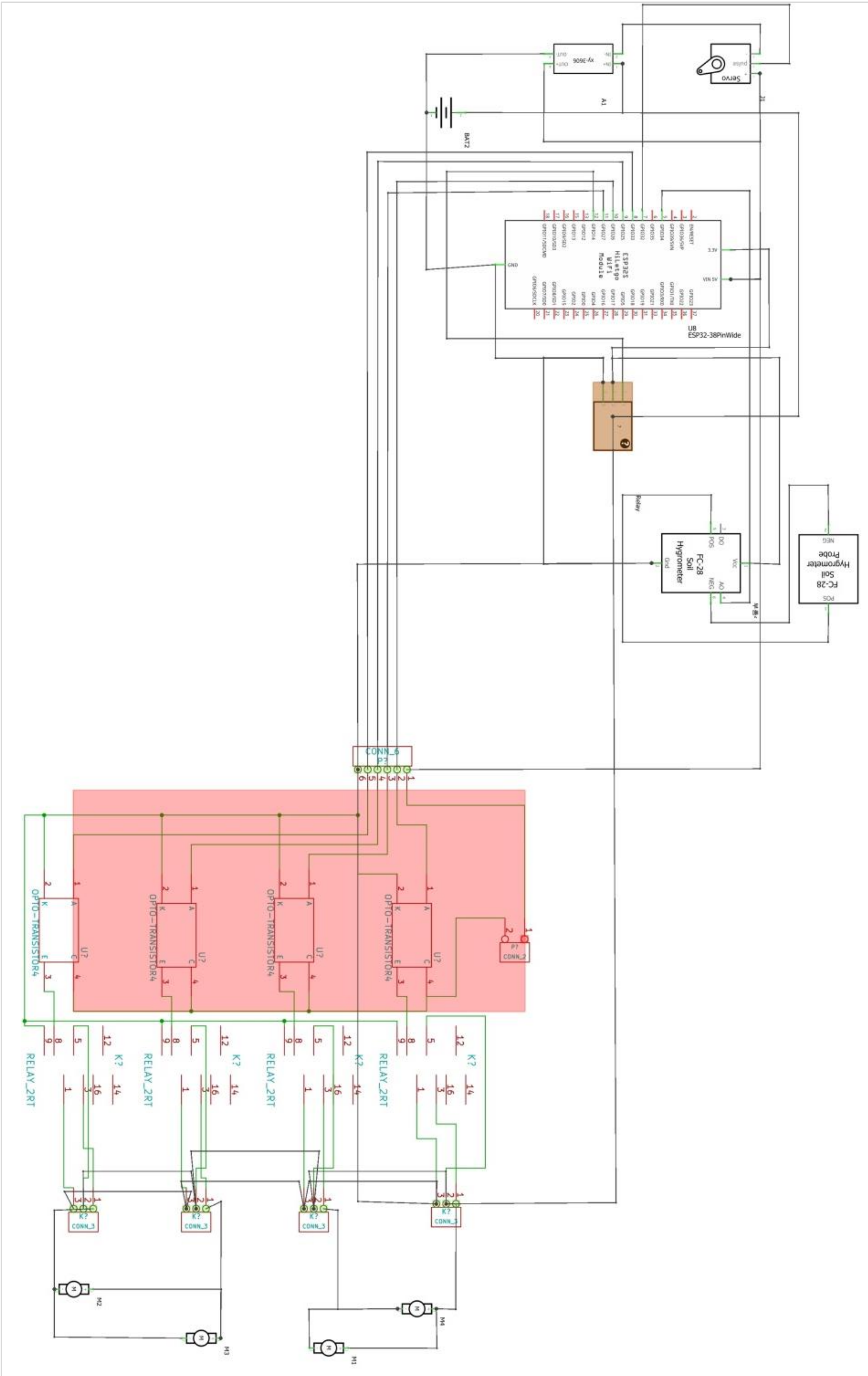


Figure 15: Schematic Diagrams

CHAPTER 5: Results and Discussion

5.1 Hardware Result

The agricultural robot successfully achieved all our objectives. We integrated all hardware and software components into a single system. The ESP32 controller reads sensor values and controls the robot's movement via the local web server. The robot can move in four directions, and the plowing rail can be easily raised and lowered. Testing the agricultural robot revealed that its movement across the field during plowing was stable, as demonstrated by the rear of the robot. This proved the efficiency and accuracy of the agricultural robot's operation. The following figure 16 shows the agricultural robot and the sensors mounted on it.

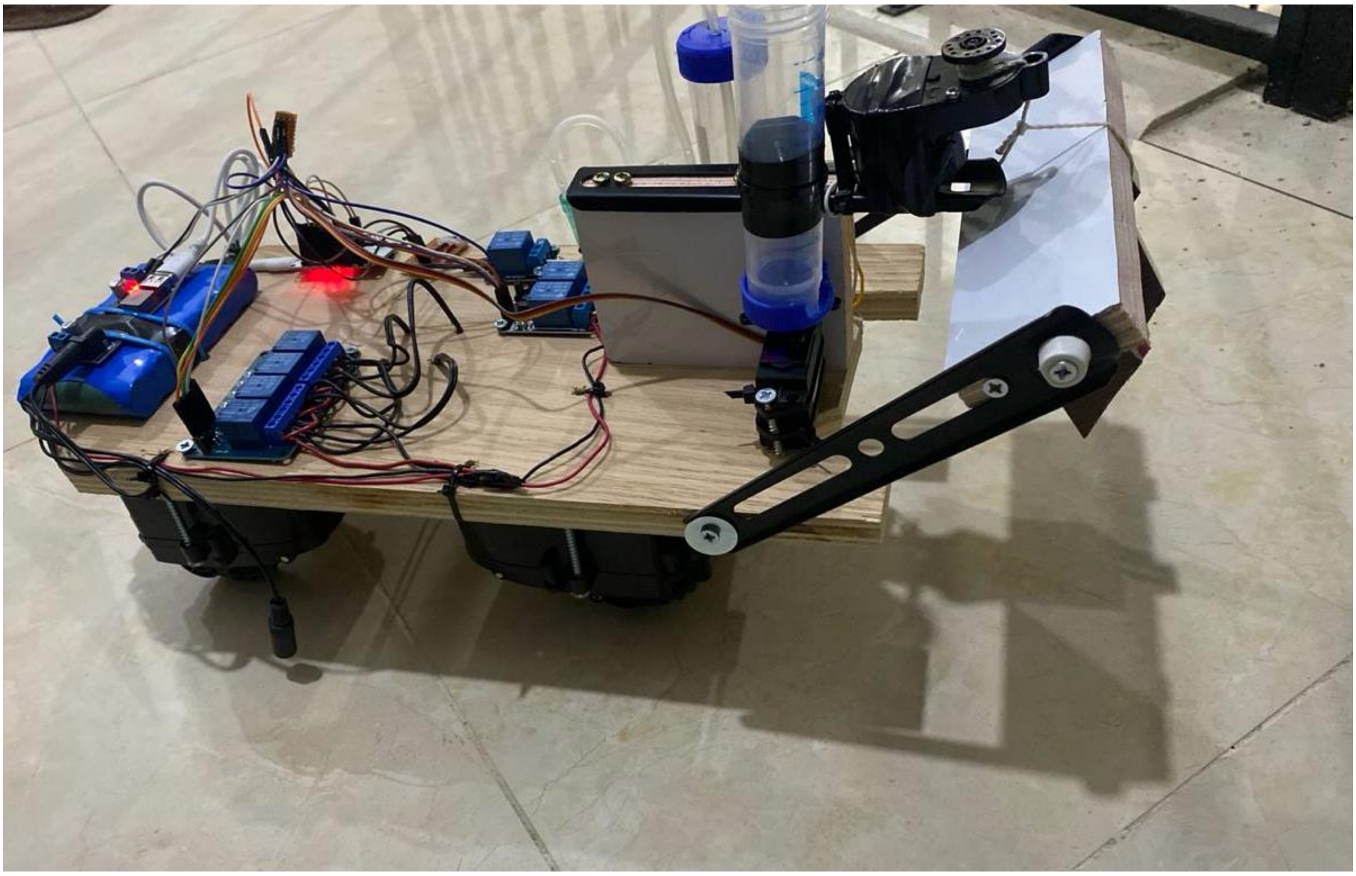


Figure 16: Agricultural robot

In addition we achieved the objective of seed planting. The seeds were distributed evenly, and the planting process was controlled via the local web server. Pressing the start button initiated seed distribution during plowing. Water pumps were also activated to irrigate the soil during both plowing and planting, as illustrated in the following figure 17.

Figure 17: Water pumps on agricultural robot

5.2 Software Result

Regarding Local Web Server Control, by establishing a local network and assigning an IP address, users could access the robot's control interface via a standard web browser on a smartphone or computer. This interface allows for immediate control of the robot's movement, tillage operations, and irrigation systems, as illustrated in the following figure18.

Figure 18: Local Web Server Control

The following figure 19 interface displays the status of system components, showing soil moisture and its current value. This helps determine if the soil needs irrigation. The screen also displays the pump status (on/off), as well as the status of the tiller, seed drill, and wheel motors—whether they are running or off.

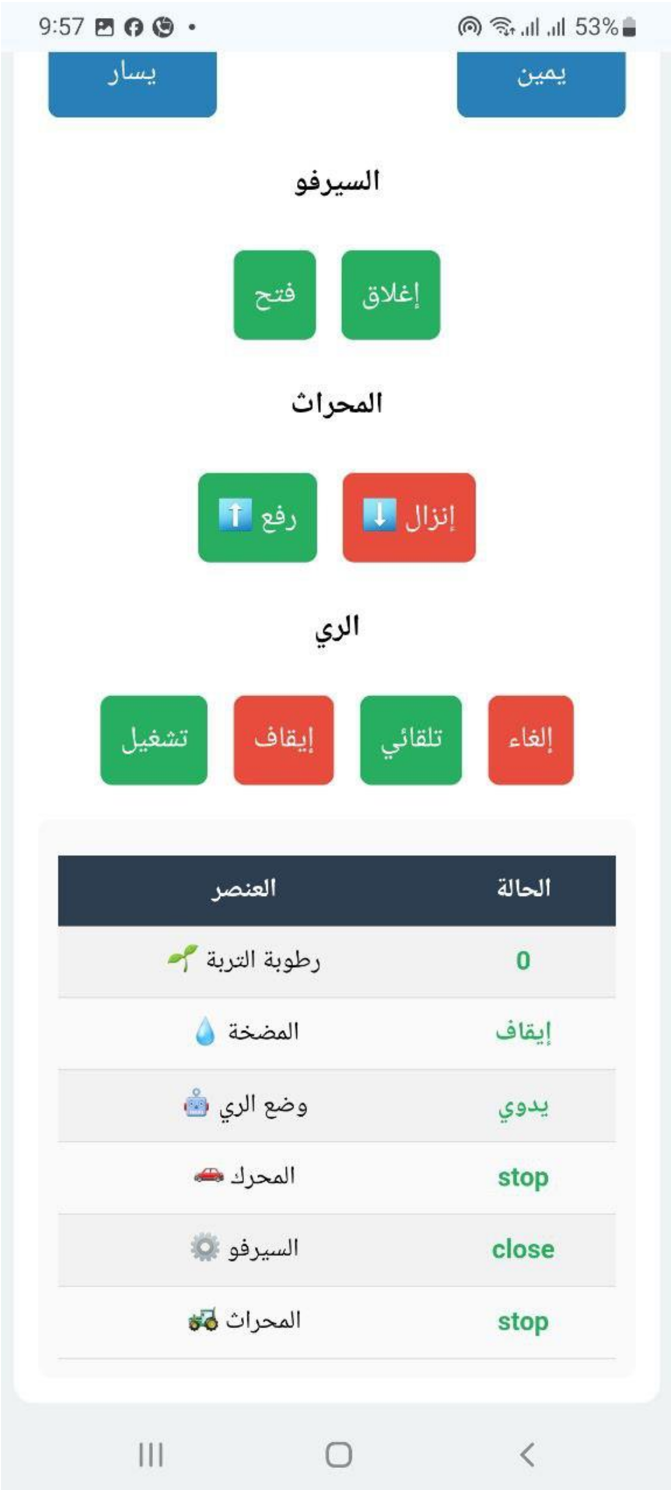


Figure 19: Interface displays the status of system

CHAPTER 6: Conclusion and Recommendation

6.1 Conclusion

In conclusion, we successfully designed and developed an intelligent agricultural robot that integrates a controller, sensors and online control. Our project demonstrated its ability to perform essential agricultural tasks such as plowing and seed planting automatically when operated from a local web server. It also monitored soil moisture and performed automated irrigation based on readings from the sensors installed on the robot.

Furthermore the use of the ESP32 controller and a local web server enabled precise and reliable control of the robot's movement. This will allow farmers to operate and monitor the robot via a simple web interface without requiring an internet connection. This feature enhances the system's flexibility and ease of use particularly in remote agricultural areas. Overall our project reduces manual labor improves water use and provides a cost effective and scalable solution for smart agriculture supporting the transition to more efficient and sustainable farming systems.

6.2 Recommendations

Based on the results achieved in our project we can propose several recommendations and future improvements to enhance the performance of the intelligent agricultural robot. The robot's mechanical design can be improved to suit various environmental conditions such as uneven ground or different soil types. This will increase its stability and operational reliability in real world agricultural fields.

In addition, new sensors can be added, such as temperature, humidity and light intensity sensors to analyze the surrounding environmental conditions. This will help in making smarter decisions and improving crop management. Furthermore an AI-powered image processing camera can be added to monitor plant health detect weeds, and diagnose diseases early.

CHAPTER 7: Limitation

Sensor accuracy:

Despite the success of our project in developing an intelligent agricultural robot, we identified several limitations during the development and testing phases. A major constraint was the robot's reliance on sensor accuracy, particularly the soil moisture sensor. Any changes or variations, such as in soil type or temperature, could affect measurement accuracy, leading to incorrect irrigation instructions and negatively impacting crop quality and yield.

Overall robot structure:

Additionally, some limitations were related to the robot's mechanical structure. We designed it to be suitable for relatively flat and homogeneous soils, as its performance may be reduced in uneven or rocky fields.

Local web server:

Furthermore, the system relies on control via a local web server. While efficient and reliable, its communication range is limited to the local network. This means the robot cannot be remotely monitored or controlled over long distances without additional communication technologies, which was another limitation.

Battery life:

As for the battery life of the agricultural robot, we had to constantly charge it to ensure its operation. This was another system limitation, requiring periodic testing and performance verification.

CHAPTER 8: System Coding

- The following figure shows the code for the controller profile and the library download call.

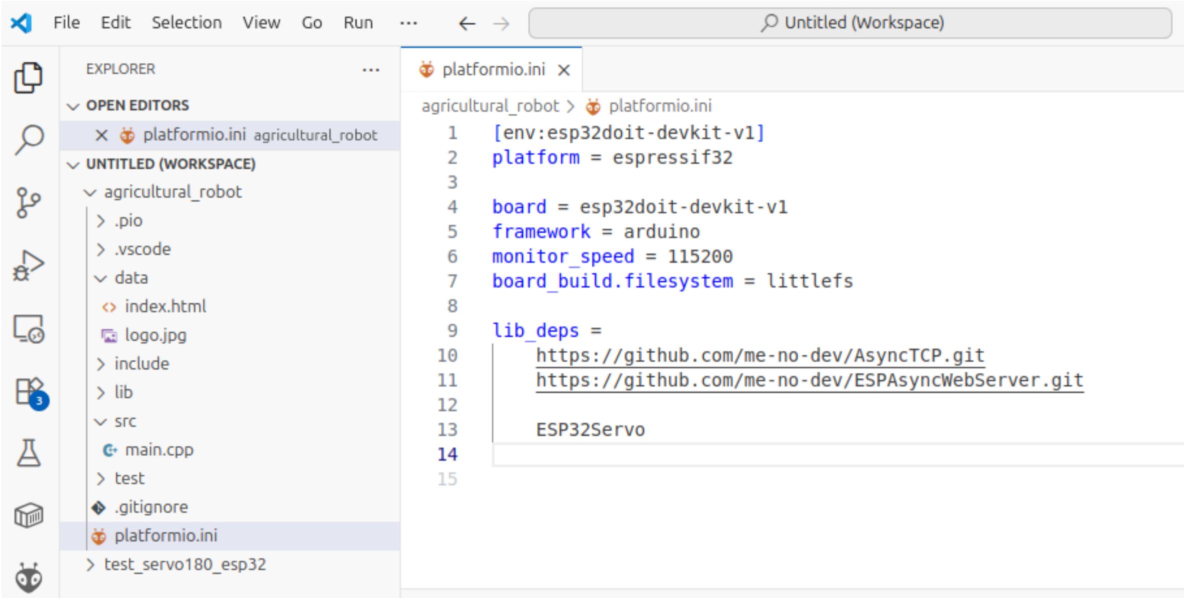


Figure 20: Code the controller profile and the library

- The following figure shows the HTML file code, which is located inside the data folder.

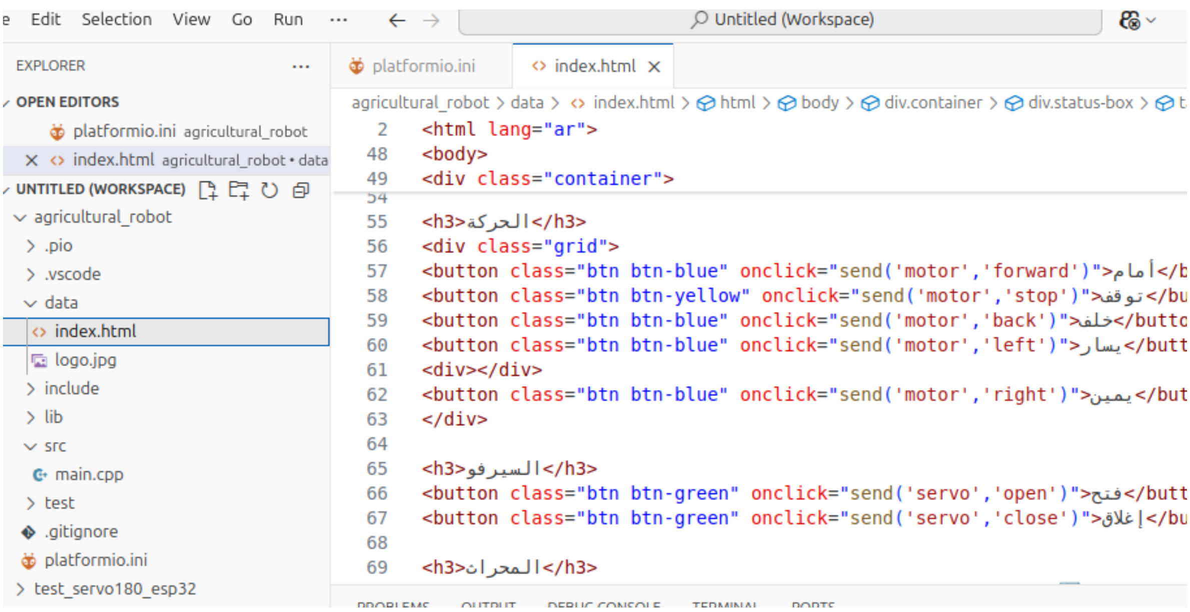


Figure 21: Code HTML file

▶ The following diagram illustrates the code that is downloaded onto the microcontroller, and we will explain several important parts of it.

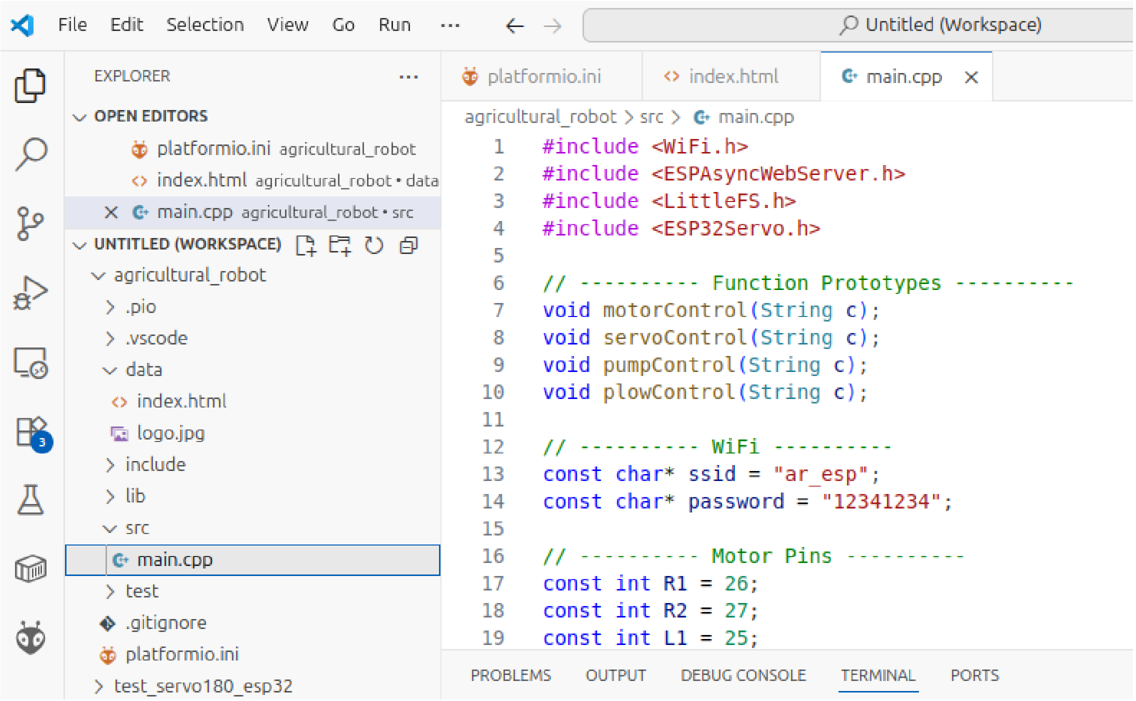


Figure 22: Code downloaded onto the microcontroller

▶ The following figure shows the code used to define the network name, password, ports, and libraries used within the code.

platformio.ini | index.html | main.cpp

agricultural_robot > src > main.cpp > setup()

```cpp
#include <WiFi.h>
#include <ESPAsyncWebServer.h>
#include <LittleFS.h>
#include <ESP32Servo.h>

// ---------- Function Prototypes ----------
void motorControl(String c);
void servoControl(String c);
void pumpControl(String c);
void plowControl(String c);

// ---------- WiFi ----------
const char* ssid = "ar_esp";
const char* password = "12341234";

// ---------- Motor Pins ----------
const int R1 = 26;
const int R2 = 27;
const int L1 = 25;
const int L2 = 33;

// ---------- Actuators ----------
const int pump_pin = 14;
const int moist_pin = 34;
const int servo_pin = 32;

// ---------- Plow Relays ----------
const int PLOW_R1 = 18;
const int PLOW_R2 = 19;

```

Figure 23: Code define the network name, password, ports

- The following figure shows the code for setting the port type and establishing a Wi-Fi connection from within the setup function.

```
platformio.ini   index.html   main.cpp ×
agricultural_robot > src > main.cpp > setup()

void setup() {
  Serial.begin(115200);

  // ---------- LittleFS ----------
  if (!LittleFS.begin(true)) {
    Serial.println("LittleFS Failed");
    return;
  }

  // ---------- Pins ----------
  pinMode(R1, OUTPUT); pinMode(R2, OUTPUT);
  pinMode(L1, OUTPUT); pinMode(L2, OUTPUT);
  pinMode(pump_pin, OUTPUT);
  pinMode(PLOW_R1, OUTPUT);
  pinMode(PLOW_R2, OUTPUT);

  digitalWrite(R1, HIGH); digitalWrite(R2, HIGH);
  digitalWrite(L1, HIGH); digitalWrite(L2, HIGH);
  digitalWrite(pump_pin, LOW);
  digitalWrite(PLOW_R1, HIGH); digitalWrite(PLOW_R2, HIGH);

  // ---------- Servo ----------
  servoMotor.attach(servo_pin);
  servoMotor.write(90);

  // ---------- WiFi ----------
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nConnected: " + WiFi.localIP().toString());
```

Figure 24: Code setting the port type and establishing a Wi-Fi

- The following figure shows the server code on the ESP32 that calls the index.php file and receives and sends data to the client through the web page.

```
platformio.ini   index.html   main.cpp ×
agricultural_robot > src > main.cpp > setup()
void setup() {

  // ---------- Web ----------
  server.on("/", HTTP_GET, [](AsyncWebServerRequest *req){
    req->send(LittleFS, "/index.html", "text/html");
  });
  server.serveStatic("/logo.jpg", LittleFS, "/logo.jpg");

  server.on("/motor", HTTP_GET, [](AsyncWebServerRequest *req){
    motorControl(req->getParam("action")->value());
    req->send(200);
  });
  server.on("/servo", HTTP_GET, [](AsyncWebServerRequest *req){
    servoControl(req->getParam("action")->value());
    req->send(200);
  });
  server.on("/pump", HTTP_GET, [](AsyncWebServerRequest *req){
    pumpControl(req->getParam("action")->value());
    req->send(200);
  });
  server.on("/plow", HTTP_GET, [](AsyncWebServerRequest *req){
    plowControl(req->getParam("action")->value());
    req->send(200);
  });

  // ---------- STATUS ----------
  server.on("/status", HTTP_GET, [](AsyncWebServerRequest *req){
    int m = analogRead(moist_pin);
    m = map(m, 4000, 0, 0, 100);
    m = constrain(m, 0, 100);
    String js = "{";
    js += "\"moisture\":\"" + String(m) + "%\",";
    js += "\"pump\":" + String(pump_state ? 1 : 0) + ",";
    js += "\"auto\":" + String(auto_mode ? 1 : 0) + ",";
    js += "\"motor\":\"" + motor_state + "\",";
    js += "\"servo\":\"" + servo_state + "\",";
    js += "\"plow\":\"" + plow_state + "\"}";
    req->send(200, "application/json", js);
  });

```

Figure 25: Code ESP32 that calls the index.php

- The following figure illustrates the control code for the motors and pump.

```cpp
// ---------- FUNCTIONS ----------

void motorControl(String c) {
  digitalWrite(R1, HIGH); digitalWrite(R2, HIGH);
  digitalWrite(L1, HIGH); digitalWrite(L2, HIGH);

  motor_state = c;

  if (c == "forward") digitalWrite(R1, LOW), digitalWrite(L1, LOW);
  else if (c == "back") digitalWrite(R2, LOW), digitalWrite(L2, LOW);
  else if (c == "left") digitalWrite(R1, LOW), digitalWrite(L2, LOW);
  else if (c == "right") digitalWrite(R2, LOW), digitalWrite(L1, LOW);
  else motor_state = "stop";
}

void servoControl(String c) {
  servo_state = c;
  if (c == "close") servoMotor.write(0);
  else if (c == "open") servoMotor.write(90);
  else servoMotor.write(90);
}

void pumpControl(String c) {
  if (c == "on") {
    auto_mode = false;
    digitalWrite(pump_pin, HIGH);
    pump_state = true;
  }
  else if (c == "off") {
    auto_mode = false;
    digitalWrite(pump_pin, LOW);
    pump_state = false;
  }
  else if (c == "auto") auto_mode = true;
  else if (c == "cancel") {
    auto_mode = false;
    digitalWrite(pump_pin, LOW);
    pump_state = false;
  }
}
```

Figure 26: Code the motors and pump

CHAPTER 9: References

[1] SM Pedersen, S Fountas, H Have, BS Blackmore, “Agricultural robots—system analysis and economic feasibility”. [Online]. Available: https://link.springer.com/article/10.1007/s11119-006-9014-9.

[2] Nikos Mylonas ,Ioannis Malounas ,Efthymios Rodias ,Christoph , “Agricultural Robotics for Field Operations” . [Online]. Available: https://www.mdpi.com/1424-8220/20/9/2672 .

[3] Yucheng Jin, Jizhan Liu, Zhujie Xu1“Development status and trend of agricultural robot technology”. [Online]. Available: http://www.ijabe.net/en/article/pdf/preview/10.25165/j.ijabe.20211404.6821.pdf

[4] Espressif "ESP32 NODMCU microcontroller" [Online]. Available: https://www.make-it.ca/nodemcu-details-specifications/ .

[5] 4-Channel Relay Module – 5 V DC, 10 A per channel [Online]. Available: https://colgroad.com/product/relay-module-4-channel/ .

[6] “12.Lithium-ion rechargeable cell 12V - 7000 mAh”. [Online]. Available: https://www.ram-e-shop.com/ar/shop/battery-li-12v-7000mah-battery-pack-12v-7000mah-btry-qbl-llshhn-lythywm-6729?srsltid=AfmBOoqEHKH1-zRvYfcDPSHxrsV0TVDvPp3klOsLaBWYNkVQWrtQOyyY.

[7] TowerPro MG996R 360° High Torque Continuous Rotation Servo Motor [Online]. Available: https://rees52.com/products/tower-pro-mg996r-digital-metal-gear-high-torque-servo-motor-360 .

[8] Wheel DC Motor" Mounting a Wheel to the DC Motor"[Online]. Available: https://forum.arduino.cc/t/mounting-a-wheel-to-the-dc-motor/183566 .

[9] Converter Board PCB XY-3606 " XY-3606 24V/12V to 5V 5A DC-DC Step Down Module" [Online]. Available: https://electropeak.com/xy-3606-24v-12v-to-5v-5a-dc-dc-step-down-module.

[10] Soil Moisture Hygrometer Detection Humidity Sensor[Online]. Available: https://www.devobox.com/en/enviromental/541-soil-hygrometer-moisture-detection-module-for-arduino.html .

[11] Arduino Integrated Development Environment (Arduino IDE) – Official Software Download and Documentation [Online]. Available: https://www.arduino.cc/en/software.

[12] Fritzing – Open-Source Electronics Design Automation (EDA) Tool [Online]. Available: https://fritzing.org .

[13] Visual Studio Code – Official Code Editor by Microsoft [Online]. Available: https://code.visualstudio.com/ .