SOIL TESTING FOR AGRICULTURE USING AGRO-BOTS

A Project (part of 8th Semester course Curriculum) submitted in partial fulfilment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

(INTERNET OF THINGS & CYBER SECURITY INCLUDING BLOCKCHAIN TECHNOLOGY)

By

Soumya Naskar (34230921011)

&

Pratham Biswas (34230921003)

Future Institute of Technology

Department of Computer Science & Engineering

(Internet of Things & Cyber Security including Blockchain Technology)

Future Institute of Technology

2021-2025

Future EDUCATION

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

(INTERNET OF THINGS & CYBER SECURITY INCLUDING BLOCKCHAIN TECHNOLOGY)

FUTURE INSTITUTE OF TECHNOLOGY

KOLKATA – 700154

2025

CERTIFICATE

We do hereby declaring that the work which is being presented in the Project Report entitled **"Soil Testing For Agriculture Using Agro Bots",** in partial fulfilment of the requirements for the award of the **Bachelor of Technology** in **CSE(IOT-CYS-BCT)** and submitted to the **Department of CSE(IOT-CYS-BCT)** of **Future Institute of Technology, Kolkata,** is an authentic record of our own work carried out during the period from July 2024 to May 2025, under the supervision of **Prof. Tuli Bakshi**, HoD, Department of Computer Science & Engineering (IoT-CYS-BCT), and Prof. Indrani Rana, Assistant Professor, Department of Computer Science & Engineering (IoT-CYS-BCT).

The matter presented in this thesis has not been submitted by us for the award of any other degree elsewhere.

Full Signature of the Students

a)

b)

This is to certify that the above statement made by the students, is correct to the best of my knowledge.

Date:

Prof. Tuli Bakshi,
Head of Department,
Department of Computer Science & Engineering,
(Internet of Things & Cyber Security including Blockchain Technology),
Future Institute of Technology, Kolkata, WB.

Prof. Indrani Rana,
Assistant Professor,
Department of Computer Science & Engineering,
(Internet of Things & Cyber Security including Blockchain Technology),
Future Institute of Technology, Kolkata, WB.

ACKNOWLEDGEMENT

We, the students of Department of Computer Science & Engineering (Internet of Things & Cyber Security including Blockchain Technology) 8th Semester, are submitting the Project work titled "**SOIL TESTING FOR AGRICULTURE USING AGRO BOTS**" as a part of 8th semester course curriculum which was carried out by us under the guidance and supervision of the Department of Computer Science and Engineering (IOT-CYS-BCT) and express our sincere thanks to all the faculty and staff members of the department for their kind operation.

We would also like to thank our guide, Prof. Tuli Bakshi, HoD, Department of Computer Science and Engineering (IOT-CYS-BCT), and Co-Guide, Prof. Indrani Rana, Assistant Professor, Department of Computer Science and Engineering (IOT-CYS-BCT) for extending a helping hand towards preparation and submission of the Project work.

Date:

Place:

Soumya Naskar
Roll No. – 34230921011

Pratham Biswas
Roll No. – 34230921003

CONTENT

Abstract

In modern agriculture, efficient soil management is essential for maximizing crop yield and ensuring sustainable farming practices. Traditional soil testing methods are often time-consuming, labour-intensive, and lack real-time feedback. This project presents an innovative solution through the development of an **Agro Bot** — a manually controlled ground surveillance robot equipped with a suite of environmental sensors to measure **temperature**, **humidity**, **soil moisture**, and **gas concentration**.

The robot navigates farmland manually and collects data using on-board sensors including the **DHT11 sensor** for temperature and humidity, a **soil moisture sensor**, and a **gas sensor** for detecting harmful gases. The collected data is transmitted in real-time to the **ThingSpeak** cloud platform via Wi-Fi, allowing farmers and agricultural experts to monitor and analyse soil conditions remotely.

This system promotes **precision agriculture** by enabling data-driven decision-making, optimizing irrigation, improving crop health monitoring, and reducing environmental impact. The Agro Bot serves as a cost-effective and scalable solution for integrating **IoT and robotics** into everyday agricultural practices.

INTRODUCTION

Agriculture is one of the most ancient and essential human activities, dating back to the dawn of civilization over 10,000 years ago. In its earliest forms, it involved the domestication of plants and animals, allowing human societies to shift from a nomadic lifestyle to a more settled and structured way of life. Early agricultural practices were simple and manual, relying on basic tools made from stone, wood, and bone. Farmers depended heavily on natural patterns, such as the seasons and weather, to determine planting and harvesting times. Over time, with the growth of civilizations like Mesopotamia, the Indus Valley, and Ancient Egypt, farming techniques evolved. Innovations such as irrigation systems, the plow, and crop rotation began to emerge, slowly increasing productivity and efficiency.

The Industrial Revolution brought about a significant transformation in agriculture with the advent of machinery. Tools like mechanical plows, seed drills, and threshing machines drastically reduced human labour and increased output. Later, the Green Revolution of the mid-20th century introduced high-yield crop varieties, synthetic fertilizers, and chemical pesticides, leading to a dramatic boost in food production, particularly in developing countries. However, these advances also had their downsides. The overuse of chemical inputs led to soil degradation, water contamination, and a growing concern for the long-term sustainability of these practices. Additionally, rising global populations, climate change, and resource scarcity have placed immense pressure on the agricultural sector, demanding more food with fewer resources and greater environmental sensitivity.

In response to these challenges, agriculture has entered a new era marked by the integration of advanced digital technologies, with the Internet of Things (IoT) playing a pivotal role. IoT refers to a network of interconnected physical devices embedded with sensors, software, and connectivity tools that collect and share data in real time. In the context of agriculture, this means equipping fields, equipment, and livestock with sensors that monitor various parameters such as soil moisture, temperature, humidity, gas levels, pH, and crop health. These sensors gather and transmit real-time data to centralized cloud platforms, where it is analysed and used to make informed decisions. With this technology, farmers no longer need to rely solely on traditional knowledge or instinct. Instead, they can make data-driven decisions that enhance productivity, conserve resources, and respond promptly to emerging issues.

The introduction of IoT into agriculture has revolutionized farming in numerous ways. It enables precision farming, where resources such as water, fertilizers, and pesticides are applied only when and where needed, and minimizing waste and environmental impact. Automated irrigation systems, for example, can detect when the soil is dry and activate water pumps only in those areas that require moisture. Similarly, environmental sensors can alert farmers to sudden temperature changes or pest infestations, allowing them to act quickly and prevent crop loss. Livestock monitoring devices help track the location and health of animals, ensuring their well-being and reducing losses. All this data can be accessed remotely through smartphones or computers, allowing farmers to manage their operations even from a distance.

Furthermore, cloud-based IoT platforms like ThingSpeak and Blynk allow for real-time visualization and analysis of collected data. This capability empowers farmers to monitor long-term trends, improve planning, and automate repetitive tasks, all while minimizing labour costs. The scalability of IoT systems makes them suitable not only for large-scale industrial farms but also for smallholder farmers in rural areas, helping bridge the digital divide in agriculture. By optimizing resource usage, improving yield, and enhancing sustainability, IoT contributes significantly to food security and environmental protection.

Key Applications of IoT in Agriculture:-

1. **Precision Agriculture:**

   Precision agriculture uses IoT sensors to deliver data-driven insights for managing crops. For example, sensors can detect soil moisture levels in different parts of a field and activate irrigation systems only where needed, conserving water and improving plant health.

2. **Smart Irrigation System:**

   IoT-based irrigation systems automate water distribution based on real-time soil moisture and weather conditions. These systems eliminate overwatering and reduce water consumption by up to 50%.

3. **Social Health Monitoring:**

   Devices equipped with sensors can continuously monitor soil parameters such as **temperature**, **moisture**, **nutrients**, and **gas levels**. This allows farmers to apply fertilizers and composts more precisely, ensuring optimal soil fertility and avoiding nutrient run-off.

4. **Climate and Weather Tracking:**

   By integrating local weather stations or remote weather APIs, farmers can use IoT platforms to anticipate rainfall, frost, or extreme heat events, and plan their activities accordingly — from sowing and harvesting to pesticide application.

5. **Livestock Monitoring:**

   Smart collars or RFID tags help monitor animal health, movement, and breeding cycles. Alerts can be sent to the farmer if an animal becomes sick, inactive, or strays from the herd.

6. **Crop Disease Detection:**

   Advanced IoT systems use drones and image processing to detect signs of crop diseases or pest infestations before they spread, allowing for timely treatment and reduced crop loss.

OBJECTIVE

The primary objective of this project is to design and implement a semi-autonomous ground-based agricultural robot—referred to as the **Agro Bot**—that can perform real-time soil testing and environmental monitoring to aid in precision agriculture. This system is intended to empower farmers and agricultural professionals with accurate, real-time data about key soil and atmospheric parameters including **soil moisture**, **soil temperature**, **ambient humidity**, and the **presence of harmful or volatile gases**. The collected data is transmitted to a **cloud-based platform (ThingSpeak)** where it is stored, visualized, and analysed, allowing for better-informed decisions regarding irrigation, fertilization, crop selection, and timing of cultivation activities.

The Agro Bot is equipped with a suite of sensors, including the **DHT11 sensor** for temperature and humidity measurements, a **soil moisture sensor** to detect water content levels in the soil, and a **gas sensor** to identify the presence of gases that may affect crop health or signal soil contamination. The robot is manually controlled, allowing operators to navigate it through fields and collect data from various zones. This manual control allows for targeted, flexible testing in diverse farm environments, especially beneficial for heterogeneous fields.

By integrating **Internet of Things (IoT)** capabilities into the system, the Agro Bot ensures seamless transmission of the acquired data to the cloud. This real-time data accessibility allows stakeholders to monitor and evaluate field conditions remotely from any internet-enabled device. Furthermore, the historical data stored in the cloud can be used for trend analysis, predictive modelling, and optimizing agricultural practices over time.

The broader goal of this project is to **improve resource efficiency**, **reduce labour intensity**, and **enhance the sustainability and productivity** of agricultural practices. It addresses critical issues faced by modern farmers, such as water scarcity, soil degradation, and lack of timely soil diagnostics, by offering an affordable and scalable solution using commonly available electronic components and open-source platforms. Ultimately, the Agro Bot aims to serve as a practical prototype for precision farming, encouraging the adoption of **smart farming technologies** in both smallholder and commercial agricultural settings.

Specific Technical Objectives:

1. **Development of a Ground Surveillance Robotic Platform:**

   To construct a mobile robotic platform (Agro Bot) using affordable and easily accessible hardware components such as Arduino Uno, ESP32, motor drivers, and a manually controlled navigation system to traverse farmland for data collection.

2. **Integration of Sensor Module for Real-Time Soil and Environmental Monitoring:**

   To integrate and calibrate a range of sensors including:
   - DHT11 sensor for measuring ambient temperature and humidity
   - Soil moisture sensor for determining the water content of the soil

- Gas sensor (e.g., MQ series) to detect the presence of harmful or volatile gases like ammonia, methane, or CO2 that may indicate pollution or soil degradation

3. **Bidirectional Communication between Arduino Uno and ESP32:**

   To establish a reliable serial communication protocol between Arduino Uno (sensor hub) and ESP32 (IoT interface) to efficiently transmit sensor data without loss or error.

4. **Implementation of IoT Cloud Integration using ThingSpeak:**

   To connect the ESP32 microcontroller to Wi-Fi and configuration it to upload real-time sensor data to ThingSpeak, enabling remote access, live visualization and historical trend analysis of field conditions.

5. **Design of a Cloud-Based Dashboard for Data Visualization:**

   To design and implement user-friendly dashboards on ThingSpeak that visually represent temperature, humidity, soil moisture, and gas concentrations through charts, gauges, and logs for easier interpretation by end users (e.g., farmers or agricultural analysts).

6. **Manual Control and Navigation of the Agro Bot:**

   Tor design a simple control interface (wired or wireless) that allows the operator to manoeuvre the robot manually across different parts of the field, ensuring flexible and site specific data collection.

7. **Power Management and Portability:**

   To ensure the Argo Bot is fully portable, powered by a rechargeable battery source, with efficient energy consumption for continuous field operation without frequent recharging.

Strategic and Impact-Oriented Objectives:

8. **Facilitate Site-Specific Soil Health Monitoring:**

   To enable farmers to gather granular-level data from different zones of their farmland, helping them identify underperforming areas or zones with specific deficiencies or issues.

9. **Enable Data-Driven Decision Making in Agriculture:**

   To promote precision agriculture by providing timely and accurate data that supports decisions related to irrigation scheduling, fertilizer application, crop selection, and disease prevention.

10. **Minimize Resource Wastage and Improve Sustainability:**

    To reduce unnecessary usage if water, fertilizers and chemicals through accurate soil monitoring, thus lowering operational costs and reducing the environmental footprint of farming practices.

11. **Promote Technological Adoption in Rural Agriculture:**

    To offer a low-cost, scalable solution that can be easily adopted by small and marginal farmers, encouraging wider use of IoT-based smart farming practices in rural and undeserved regions.

12. **Support Educational and Research Purposes:**

    To provide a working prototype that can be used by students, researches and institutions for educational demonstration, agricultural innovation and further enhancements in agritech.

13. **Contribute to the Long-Term Goal of Food Security and Sustainable Agriculture:**

    To support broader sustainable development goals (SDGs), particularly those focused on zero hunger, climate action and responsible consumption by increasing agricultural productivity and resilience through technology.

ADVANTAGES

1. Real – Time Monitoring of Soil and Environmental Conditions:

One of the most important advantages of this project is the ability to monitor soil and atmospheric parameters in real time. The use of sensors like DHT11 (for temperature and humidity), a soil moisture sensor, and a gas sensor enables the Agro Bot to collect accurate and live data from the field. This ensures that farmers receive timely information, helping them make immediate and informed decisions regarding irrigation, fertilization, and crop care.

2. Promotes Precision Agriculture:

By enabling location–specific soil testing, this project supports precision agriculture, where inputs like water, fertilizers and pesticides are applied only where necessary. This targeted approach minimizes resource wastage and maximizes crop yield. Farmers can identify which sections of a field are dry or nutrient-deficient and treat them accordingly, avoiding overuse of water and chemicals.

3. Integration with Cloud-Based Platforms for Remote Access:

The integration of the IoT-enabled ESP32 with ThingSpeak cloud services allows for wireless data transmission and remote access. This means farmers or agricultural scientists can monitor conditions from anywhere, using a smartphone or computer. The ability to view historical data, real-time graphs, and alerts online automation and intelligence to agricultural monitoring.

4. Reduces Manual Labour and Human Error:

Traditional soil testing methods are labour-intensive and often require samples to be physically collected and sent to a lab. This project automates the process by collecting and transmitting data directly from field. It reduces human error and the time required for manual sampling, analysis and documentation.

5. Early Detection of Harmful Gases and Soil Contaminants:

The inclusion of a gas sensor (such as MQ-series sensors) adds another layer of environmental intelligence. The Agro Bot can detect gases like ammonia, methane, or

CO_2 that might indicate contamination or anaerobic soil conditions. Early detection of such harmful elements helps prevent crop damage, supports soil health management, and ensures better crop safety.

6. Low-Cost, Scalable, and DIY Friendly:

This project uses affordable, widely available components such as Arduino Uno, ESP32, and basic sensors. As a result, the system is cost-effective and highly accessible for small and medium-scale farmers, agricultural students, and researchers. It can also serve as a prototype for more advanced systems, and its modular design makes it easy to scale or modify for future needs.

7. Encourages Data-Driven Agriculture Practices:

The project introduces farmers to **data-driven farming**, which is a major leap from traditional, intuition-based decision-making. Having access to environmental trends over time allows for long-term planning, predictive maintenance of crops, and risk mitigation (e.g., predicting drought stress or irrigation needs).

8. Portability and Field Flexibility:

The Agro Bot is designed to be mobile and manually controlled, making it capable of navigating through different parts of a farm. This portability ensures that data can be collected from various zones, even those that are remote or unevenly fertilized, leading to better understanding and management of field variability.

9. Educational and Research Value:

This project serves as an excellent educational tool for students studying agriculture, electronics, robotics, or environmental science. It demonstrates how multidisciplinary technologies can be combined to solve real-world problems. It also provides a research platform to test new sensors, data transmission methods, or AI-based predictive models in agriculture.

10. Environmental Sustainability:

By optimizing water use and minimizing chemical applications through precise data, the project contributes to **sustainable farming**. It prevents over-irrigation and nutrient runoff, conserves groundwater, and promotes healthy soil ecosystems, aligning with global goals of eco-friendly agriculture and climate-resilient practices.

11. Empowering Small and Marginal Farmers:

Often, small-scale farmers lack access to high-end agri-tech solutions due to their cost and complexity. This project, due to its simplicity and affordability, empowers such farmers to adopt smart farming tools, improving their yield and income with minimal investment.

12. Foundation for Future Automation:

While this prototype is manually controlled, it lays the groundwork for future enhancements such as autonomous navigation, AI-based analytics, or drone integration. This flexibility ensures that the system can evolve with advancing technology without needing a complete redesign.

COMPONENTS USED

1. HARDWARE COMPONENTS:

1.1. ARDUINO UNO:

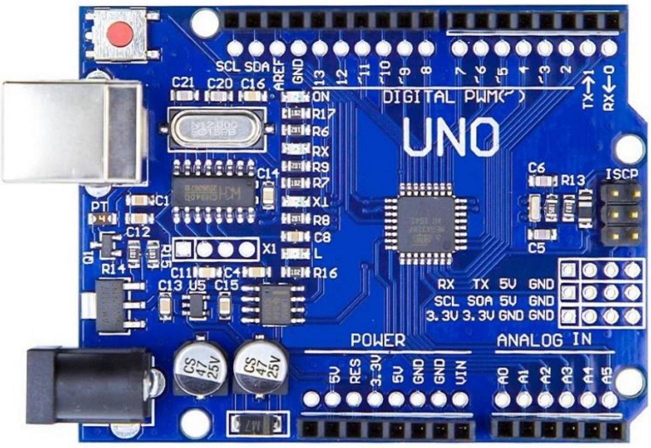


Fig. 1.0 – Arduino Uno

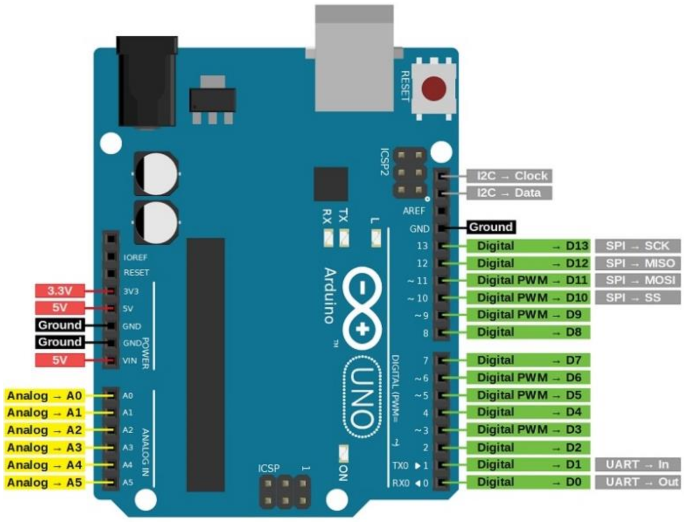


Fig. 1.1 – Arduino Uno Pin Diagram

The **Arduino Uno** is one of the most popular and widely used microcontroller boards in the Arduino ecosystem. Developed by the Arduino.cc team, it serves as an open-source electronics platform based on easy-to-use hardware and software. Due to its simplicity, affordability, and versatility, the Arduino Uno has become a go-to solution for hobbyists, students, educators, and professionals working on embedded systems, automation, robotics, and IoT projects.

At the heart of the Arduino Uno lies the **ATmega328P** microcontroller, an 8-bit AVR chip from Microchip Technology. This microcontroller features:

- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Digital I/O Pins: 14 (of which 6 can provide PWM output)
- Analog Input Pins: 6
- Clock Speed: 16 MHz
- Flash Memory: 32 KB (0.5 KB used by bootloader)
- SRAM: 2 KB
- EEPROM: 1 KB

These features make it powerful enough for most embedded applications while still being easy to program and deploy.

The Arduino Uno is programmed using the **Arduino IDE (Integrated Development Environment)**, which supports simplified C/C++ programming. The IDE features a straightforward interface, a wide range of built-in libraries, and a large community of contributors. Programs written in the IDE are referred to as “sketches” and are uploaded to the board via the USB cable.

Additionally, the Arduino Uno supports **serial communication**, allowing it to interact with PCs or other serial devices. This is particularly useful for debugging, data logging, and IoT integration, such as sending sensor data to an ESP32 or cloud service.

The Arduino Uno can be powered via:

- USB connection (providing 5V directly)
- Barrel Jack using an external adapter (7-12V recommended)
- VIN pin, where an external voltage source can be supplied. The on-board voltage regulator ensures that sensitive components receive a steady 5V or 3.3V as required.

Thanks to its flexibility, the Arduino Uno is ideal for a wide range of applications, including:

- Sensor Interfacing (e.g., soil moisture, temperature, humidity)
- Home Automation and smart farming
- Robotics and motor control
- IoT prototyping when connected with modules like ESP32. Wi-Fi, GSM, or Bluetooth
- Educational purposes for learning programming, electronics, and embedded systems

The Arduino Uno stands out as a versatile, robust, and beginner-friendly microcontroller board that bridges the gap between theoretical learning and practical implementation. In the context of projects like the Agro Bot for soil testing, it plays a vital role in collecting sensor data, performing basic signal processing, and communicating with other devices such as the ESP32 for cloud integration. Its simplicity and reliability make it a cornerstone in the development of smart, connected systems in modern agriculture and beyond.

1.2. ESP32:

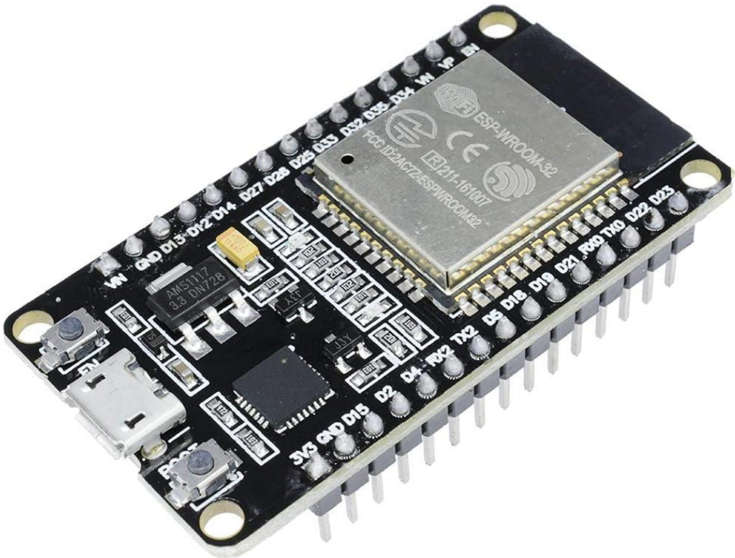


Fig. 2.0 – ESP32 Microcontroller

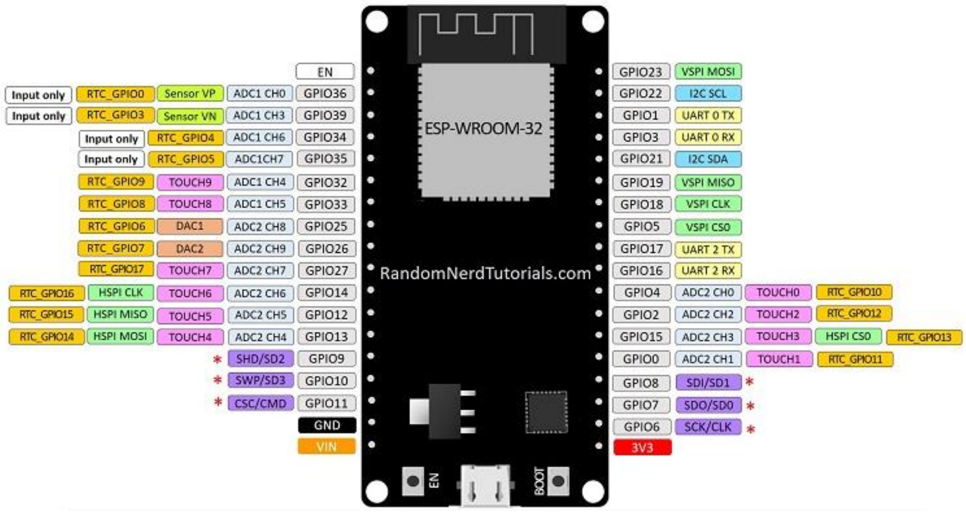


Fig. 2.1 – ESP32 Pin Diagram

The **ESP32** is a highly integrated and powerful microcontroller developed by **Espressif Systems**. It is an upgrade to the earlier ESP8266, offering a significant boost in performance, wireless communication capabilities, and peripheral interfaces. The ESP32 is widely used in **Internet of Things (IoT)** applications due to its built-in Wi-Fi, Bluetooth, and a wide range of GPIO pins and features — making it ideal for smart devices, remote monitoring systems, robotics, automation, and wireless sensor networks.

At its core, the ESP32 is based on a **dual-core Tensilica Xtensa LX6 processor**, which can be configured as either single-core or dual-core depending on power and performance needs. It includes:

- Processor: Dual-core Xtensa 32-bit LX6 microprocessor (up to 240 MHz)
- RAM: 250 KB SRAM

- Flash Memory: Varies by module (typically 4MB)
- Wireless Connectivity:
    - Wi-Fi (802.11 b/g/n)
    - Bluetooth v4.2 (Classic and BLE)
- Operating Voltage: 3.0V to 3.3V
- GPIO Pins: 34 programmable pins (not all available in every model)
- Analog Inputs: 18 channels of 12-bit ADC (Analog-to-Digital Converter)
- DAC Outputs: 2 channels of 8-bit DAC
- PWM Channels: Multiple PWM outputs
- Communication Interfaces:
    - UART, SPI, I2C, CAN, and IR
- Timers and Watchdogs: Multiple general-purpose timers, watchdogs, and RTC (real-time clock)

These features make the ESP32 extremely versatile for any real-time, networked application. One of the biggest advantages of the ESP32 is that it comes with integrated Wi-Fi and Bluetooth radios, eliminating the need for external modules:

- Wi-Fi: Supports full TCP/IP stack and multiple encryption protocols (WPA/WPA2)
- Bluetooth: Supports both Classic Bluetooth and Bluetooth Low Energy (BLE) for connecting to wireless peripherals or smartphones

In our Agro Bot project, this is used to upload sensor data wirelessly to the cloud (ThingSpeak), making ESP32 the communication bridge between hardware and the Internet.

ESP32 supports multiple development environments, including:

- Arduino IDE (most common for beginners)
- PlatformIO
- Espressif’s own ESP-IDF (IoT Development Framework)
- MicroPython
- Lia (NodeMCU firmware)

The ESP32 comes in several hardware variants:

- ESP32-WROOM-32: The most common and affordable module
- ESP32-WROVER: Includes additional PSRAM and advanced features
- Dev Boards: Such as NodeMCU ESP32 and DOIT ESP32, which include USB, voltage regulation, and pin headers for easy development

The **ESP32** is a powerful and flexible microcontroller ideal for wireless and IoT applications. With integrated Wi-Fi and Bluetooth, a fast processor, and rich peripheral support, it serves as an efficient bridge between embedded systems and the internet. In the context of the Agro Bot project, it plays a pivotal role in real-time data communication, making traditional soil testing intelligent, automated, and remotely accessible — thus bringing the power of smart agriculture to the hands of farmers and researchers.

1.3. Soil Moisture Sensor:

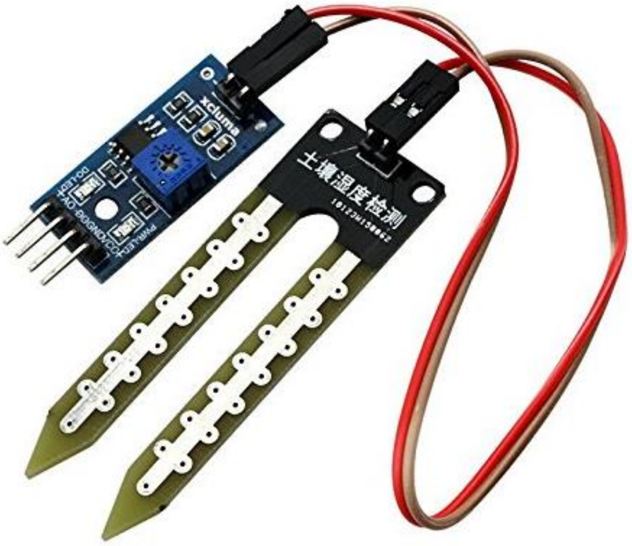


Fig. 3.0 – Soil Moisture Sensor

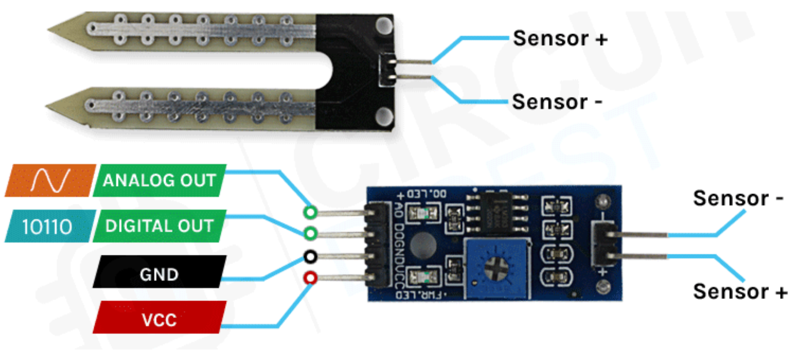


Fig. 3.1 – Soil Moisture Sensor Pin Diagram

The **Soil Moisture Sensor** is an electronic sensing device used to measure the water content (moisture level) in soil. In agriculture, horticulture, environmental monitoring, and automated irrigation systems, soil moisture sensors play a critical role in determining when and how much to irrigate. This helps optimize water use, prevent overwatering, and improve crop yield and soil health.

In IoT-based agricultural projects such as the **Agro Bot**, the soil moisture sensor acts as one of the key input devices for assessing ground conditions in real-time and facilitating smart decision-making.

The most commonly used soil moisture sensor in educational and prototype-level projects is the resistive soil moisture sensor. It operates based on the principle of **electrical resistance** between two probes inserted into the soil.

- When the soil is **wet**, water conducts electricity better, reducing resistance and allowing more current to pass.
- When the soil is **dry**, resistance increases, reducing current flow.

This variation in resistance is translated into an analog voltage signal, which is read by microcontrollers like Arduino Uno or ESP32.

A typical soil moisture sensor module includes:

- **Sensor Probes (Electrodes):** Two conductive metal rods (usually made of copper or corrosion-resistant material) that are inserted into the soil to detect moisture levels.
- **Signal Conditioning Board/Module (Optional):** Many soil moisture sensors come with an attached module that features:
    - Analog and Digital outputs
    - On-board Comparator (usually an LM393)
    - Potentiometer to set a threshold for digital output
    - Power indicator LED and digital output indicator LED
- **Output Pins:**
    - AO (Analog Output): Sends a continuous analog voltage based on the moisture level.
    - DO (Digital Output): Outputs HIGH or LOW based on a user-defined moisture threshold.
    - VCC and GND: For power supply (usually 3.3V-5V).

The sensor is typically connected to an **analog input pin** on a microcontroller like Arduino Uno, which reads the voltage and maps it to a moisture percentage or threshold value. For example:

```
int sensorValue = analogRead(A0);
int moisturePercent = map(sensorValue, 0, 1023, 100, 0);  // Inverted scale
```

The **Soil Moisture Sensor** is an essential tool in modern precision agriculture and smart farming systems. It enables real-time detection of soil water levels, allowing for informed and automated irrigation decisions. In the context of our **"Soil Testing Agro Bot"**, the sensor provides critical data that contributes to sustainable water use, better crop health, and an overall smarter agricultural ecosystem. Its affordability, ease of use, and adaptability make it a valuable component in both academic projects and real-world agricultural deployments.

1.4. GAS SENSOR (MQ Series):

Fig. 4.0 – Gas Sensor (MQ Series)

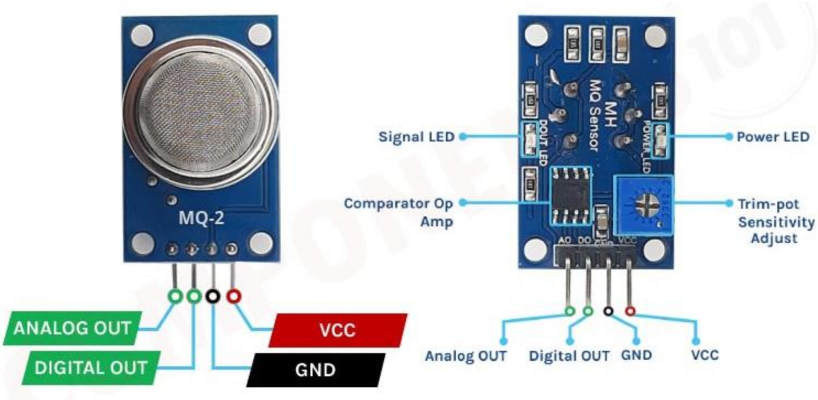


Fig. 4.1 – Gas Sensor Pin Diagram

The **MQ Series Gas Sensors** are a family of low-cost, versatile gas-detecting modules used in a variety of applications, including environmental monitoring, industrial safety, and agricultural diagnostics. These sensors are widely adopted in embedded systems and Internet of Things (IoT) projects due to their affordability, ease of interfacing, and support for detecting different gases.In our Agro Bot soil testing system, an MQ sensor helps detect the presence of gases like methane (CH_4), ammonia (NH_3), carbon monoxide (CO), or others that may be released from decomposing organic material, fertilizer residue, or soil pollution.

The MQ sensors work on the **chemiresistive sensing principle**. Inside the sensor, there is a heating element and a semiconducting metal oxide layer (typically SnO_2).

- When the sensor is powered, the heating element heats the sensing material.
- In clean air, the resistance of the sensing material is stable.
- When a target gas is present, it reacts with the metal oxide surface and **changes the sensor's resistance**.

- The variation in resistance is then converted to a voltage, which can be read as an **analog signal** by a microcontroller.

Each sensor in the MQ series is calibrated to detect specific gases or groups of gases with different sensitivity levels.

A typical MQ sensor module includes:

- **Sensing element:** housed in a stainless steel mesh (filters particulates)
- **Heating coil:** inside the sensor for warming the element
- **Pins for connections**: typically VCC, GND, Analog Output (AO), and Digital Output (DO)
- **Signal conditioning circuitry**: like op-amps or comparators
- **Potentiometer**: for adjusting sensitivity threshold on digital output
- **Onboard LEDs**: indicating power and digital output status

The MQ sensors are easily interfaced with platforms like **Arduino Uno** or **ESP32**:

- **Analog Output (AO):** Connects to analog pins (e.g., A0) of Arduino to read a variable voltage based on gas concentration.
- **Digital Output (DO):** Sends a HIGH or LOW signal based on a threshold set by the onboard potentiometer.

Example Arduino code snippet:

```
int gasSensor = analogRead(A0);
Serial.print("Gas Level: ");
Serial.println(gasSensor);
```

Some popular MQ series Models and the gases detected by them are:

Sensor	Target Gases
MQ-2	LPG, propane, methane, hydrogen, smoke
MQ-3	Alcohol, ethanol, smoke
MQ-4	Methane, natural gas
MQ-5	LPG, natural gas, town gas
MQ-6	LPG, isobutane
MQ-7	Carbon monoxide (CO)
MQ-8	Hydrogen (H_2)
MQ-135	Ammonia (NH_3), alcohol, benzene, smoke, CO_2, etc.

The **MQ Series Gas Sensors** are practical tools for implementing low-cost, effective gas monitoring in smart agriculture and environmental systems. In our Agro Bot soil testing system, integrating an MQ sensor allows detection of soil-borne or atmospheric gases, giving valuable insights into soil health, fertilizer effects, or potential pollution. When paired with cloud platforms and microcontrollers like Arduino and ESP32, MQ sensors extend their utility to real-time remote monitoring, making them ideal for sustainable and intelligent farming.

1.5. TEMPERATURE SENSOR (DHT11):

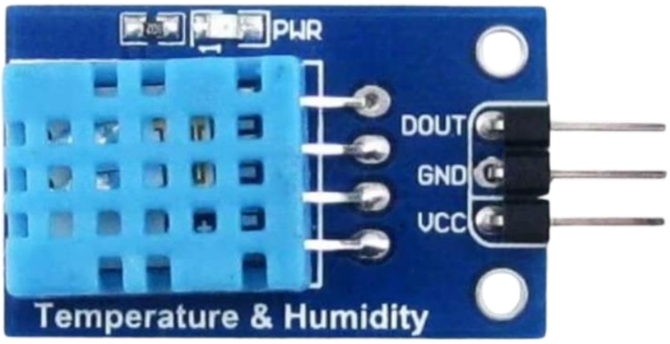


Fig. 5.0 – DHT11 (Temperature Sensor)

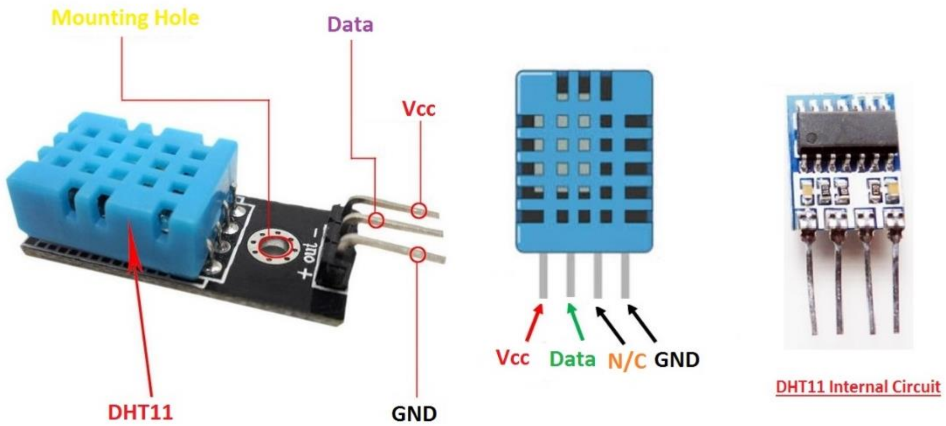


Fig. 5.1 – DHT11 Pin Diagram

The **DHT11** is a basic, low-cost digital temperature and humidity sensor used extensively in weather monitoring, indoor environmental systems, agricultural automation, and IoT applications. It provides a simple and reliable method to collect two essential atmospheric parameters—temperature and relative humidity—using a single module. In our Soil Testing Agro Bot, the DHT11 sensor helps assess the environmental conditions around the soil, which directly influences plant growth, moisture retention, and overall crop productivity.

The DHT11 operates on a **capacitive humidity sensing and thermistor-based temperature sensing** principle:

- **Humidity Sensing:** The sensor contains a **moisture-holding substrate** (usually a polymer or capacitive material) between two electrodes. When the surrounding humidity changes, the dielectric constant of this substrate changes, affecting the capacitance between the electrodes. This change is measured and converted into a humidity value.
- **Temperature Sensing:** A **thermistor** (temperature-sensitive resistor) inside the DHT11 changes its resistance with temperature. An internal analog-to-digital converter (ADC) converts the analog signal to a digital value.

Both the temperature and humidity values are **processed internally** by the sensor's microcontroller and **output digitally** through a single-wire protocol.

The DHT11 sensor module typically includes:

- **Sensing element housing:** A blue plastic casing with slits, allowing air to flow over the sensing elements.
- **PCB board:** Houses internal microcontroller, signal processing unit, and output logic.
- **3 or 4 pins:**
    - **VCC**: Power supply (3.3V–5V)
    - **GND**: Ground
    - **DATA**: Serial digital output pin
    - (Sometimes an **NC** pin - not connected)

The module may also include a **pull-up resistor** on the data line and an optional decoupling capacitor for stable readings.

DHT11 is very easy to interface with popular microcontrollers like **Arduino Uno** or **ESP32**. Libraries like `DHT.h` for Arduino provide simple functions to retrieve temperature and humidity readings.

Sample Arduino Code:

```
#include "DHT.h"
#define DHTPIN 2      // Digital pin connected to DHT11
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  dht.begin();
}

void loop() {
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  Serial.print("Humidity: ");
  Serial.print(h);
  Serial.print(" %, Temperature: ");
  Serial.print(t);
  Serial.println(" °C");
  delay(2000);
}
```

The **DHT11 Temperature and Humidity Sensor** is a simple yet powerful component in the toolbox of modern precision agriculture. In our Agro Bot, it serves as a crucial environmental monitoring device, providing real-time data that helps correlate soil conditions with ambient atmosphere. Though limited in range and precision, its affordability, ease of use, and dual functionality make it a reliable choice for beginners, educational projects, and low-cost automated farming systems.

2. ROBOT COMPONENTS:

2.1. ROBOT CHASSIS:

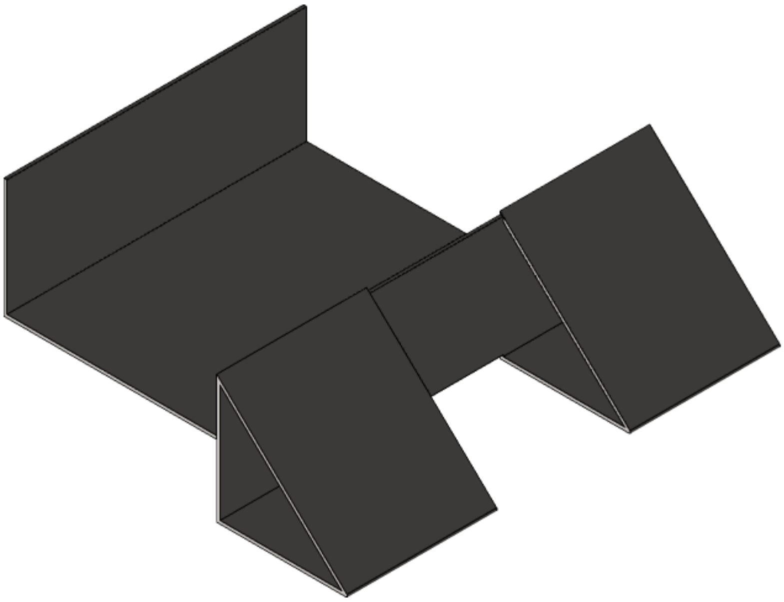


Fig. 6.0 – Proposed Chassis for the Robot

The **robot chassis** is the **mechanical frame** that forms the **structural base of a ground surveillance robot**. In the context of our soil testing agro bot, the chassis not only supports all electronic modules and sensors but also enables smooth navigation across farmland or garden soil surfaces. It is an essential component that brings together mechanical mobility, electrical integration, and real-world deployment.

The robot chassis used in this project is typically a **two-wheel or four-wheel mobile platform** made from durable materials such as metals, aluminum alloy, acrylic, or high-density plastic. It consists of several key elements:

- **Base Plate:** The main body of the chassis, often pre-drilled with multiple mounting holes for attaching motors, wheels, batteries, and microcontrollers.
- **Wheels and Motors:** Usually 2 or 4 DC gear motors with rubber wheels or all-terrain tires to allow mobility on rough and uneven surfaces like soil.
- **Mounting Slots:** Dedicated sections or holes for mounting sensor modules, including gas sensors, temperature/humidity sensors (DHT11), and soil moisture probes.
- **Battery Holder:** A compartment or bracket to hold Li-ion or Li-Po batteries for powering the whole system.
- **Enclosure or Upper Deck:** A secondary platform for organizing controllers like Arduino Uno, ESP32, jumper wires, and power regulators.

The robot chassis serves as the **platform to house and align** all the major electronics required in the Agro Bot project. Here’s how it integrates:

- **Gas Sensor (MQ Series):** Mounted on the front or top surface to ensure exposure to surrounding air.
- **DHT11 Sensor:** Placed in an open area away from heat-generating components to ensure accurate ambient temperature and humidity readings.
- **Soil Moisture Sensor:** Mounted with an extendable probe or manually inserted into the soil near the robot's path to measure moisture levels.
- **Arduino Uno and ESP32:** Placed on upper decks or screwed into mounts, often enclosed in plastic boxes to protect from dust or water.
- **Power Bank or Battery Module:** Secured at the base with Velcro straps or clips to maintain center of gravity.
- **Cables and Wiring:** Managed using zip ties or spiral tubes to prevent entanglement during movement.

In agriculture, a robust robot chassis allows **mobile soil sampling**, **data collection**, and **environmental monitoring** across different plots without human intervention. Its benefits include:

- **Accessibility:** Reaches narrow or muddy pathways between crops.
- **Automation Ready:** Easily upgradable to autonomous movement using GPS or camera vision.
- **Real-time testing:** As the robot moves, it can test multiple patches of soil—temperature, gas, moisture—without manual relocation.

This kind of chassis is ideal for **small to medium-sized farms**, **greenhouses**, **nurseries**, or **precision agriculture** research setups.

The **robot chassis** is the physical backbone of your Agro Bot project. It enables the seamless integration of various sensors and controllers while ensuring mobility, adaptability, and usability in outdoor farming environments. A well-designed chassis not only supports accurate and stable sensor readings but also expands the scope of automation in agriculture. As the base for smart farming robots, it plays a critical role in enhancing productivity, reducing labour, and enabling real-time soil and environmental diagnostics.

2.2. DC MOTORS:

Fig. 7.0 – DC Motor (Johnson, 300 RPM)

The **DC Johnson Motor (300 RPM)** is a high-torque, low-speed DC motor known for its **robust performance**, **durability**, and **reliability** in automation and robotic applications. The "Johnson" designation typically refers to the **style and manufacturer-inspired design**, which includes an integrated gearhead and sturdy shaft mounting, making it suitable for rugged environments like agricultural fields.

In the context of our project, this motor is ideal for powering the wheels of the robot chassis, allowing it to move over uneven soil terrain while carrying sensors and microcontrollers such as the Arduino Uno and ESP32.

The Johnson motor consists of two primary parts:

1. **Motor Unit (DC Motor Core):** This is the actual electrical motor, housed in a cylindrical metal casing, operating typically on 6–12V DC input.
2. **Gearbox Assembly:** Attached to the front of the motor, the gear assembly is made of **hardened metal gears** that reduce the speed and **increase the torque** output. The gear ratio determines the final RPM. In this motor, the gear ratio is optimized to deliver approximately **300 RPM** at the output shaft.
3. **Output Shaft:** The output shaft is usually a **D-shaped metal shaft** designed for securely mounting wheels or pulleys using grub screws or shaft couplers. Shaft diameters typically range from 6mm to 8mm.

This motor operates on the **Lorentz Force principle**, like any DC motor. When DC voltage is applied:

- Current flows through the motor windings, creating a magnetic field.
- This interacts with the permanent magnets in the stator, causing the armature to rotate.

- The gearbox reduces the high-speed rotation of the motor to a more useful, high-torque, low-speed rotation suitable for driving heavy loads like a robot on uneven soil.

The gear ratio inside the gearbox is crucial—it balances speed and torque. A 300 RPM motor provides moderate speed with good torque, ideal for agricultural robots that need to move steadily over soft terrain without slipping or stalling.

The motor typically has two terminals or wires—one positive and one negative:

- Reversing polarity changes the motor direction.
- It is usually controlled via a motor driver (e.g., L298N, L293D, or BTS7960) connected to a microcontroller like Arduino or ESP32.

PWM (Pulse Width Modulation) is used for speed control, while digital logic HIGH/LOW controls direction.

The **DC Johnson Motor – 300 RPM** is a rugged and reliable power source for small to medium robotic platforms. Its high torque, combined with moderate speed and durable construction, make it exceptionally suitable for agricultural robots that must perform in variable field conditions. Whether you're building a mobile soil tester or an autonomous crop monitor, this motor ensures dependable mobility and mechanical stability.

2.3. MOTOR DRIVER

Fig. 8.0 – Custom Designed Motor Driver

In robotic and electromechanical systems, a **motor controller** serves as the vital intermediary between the control unit (such as a microcontroller) and the motors that perform physical tasks. While there are many commercially available motor driver modules (like L298N, BTS7960, etc.), these off-the-shelf solutions may not always meet the exact voltage, current, performance, or customization needs of specific projects. This limitation often necessitates the design and development of a **custom-made motor controller**, tailored to the functional and operational requirements of a particular system.

A custom motor controller typically comprises power transistors (MOSFETs or BJTs)**,** current sensing circuits**,** flyback diodes, and pulse-width modulation (PWM) control logic. The controller interprets low-power signals from the microcontroller and modulates high-power output to the motors, enabling both speed and direction control. With such a setup, motor behavior can be finely tuned based on feedback from sensors like encoders, current sensors, or load detection, making it ideal for applications requiring adaptive motion—such as navigating through varied field conditions in agriculture.

Specs:

- **Input Voltage Range**: 11-17V (3S-4S)
- **Output Current (Continuous)**: 20 amps/channel
- **Output Current (Peak)**: 45 amps/channel
- **Dimensions**: 62 x 40 x 16 mm (excluding wires)
- **Weight**: 50g (excluding wires)
- **Input**: Standard Servo Input
- **Signal Range**: 1100 – 1900 us
- **Throttle Centre**: 1460-1540 us
- **Dead Band (From Centre)**: 40 us
- **BEC**: Yes (5V 100mA max)
- **Brake**: Yes
- **Signal Loss Protection**: Yes
- **Input Wire Thickness**: 14 AWG Silicone Wire (Red & Black)
- **Output Wire Thickness**: 16 AWG Silicone Wire (Yellow & Blue)

2.4. POWER SUPPLY (Li-Po Battery):

Fig. 9.0 – Li-PO Battery

A **3S Li-Po (Lithium Polymer) Battery** with a **3300mAh (milliampere-hour)** capacity is a **high-performance, rechargeable power source** that is widely used in drones, RC vehicles, and robotics due to its **lightweight construction**, **high energy density**, and **ability to deliver high current outputs**. The term **"3S"** indicates that the battery contains **three cells connected in series**, each typically rated at **3.7V nominal voltage**, yielding a combined nominal voltage of **11.1V** (3.7V × 3 cells).

This makes it an ideal energy solution for mobile platforms like agricultural bots, which require a **stable and portable power supply** capable of handling motors, microcontrollers, sensors, and communication modules (e.g., ESP32, Arduino Uno, gas sensors, etc.).

Specification	**Description**
Cell Configuration	3S (3 Cells in Series)
Nominal Voltage	11.1V (3 × 3.7V)
Fully Charged Voltage	12.6V (4.2V per cell × 3)
Capacity	3300mAh (3.3 Ah)
Discharge Rate (C-Rating)	Typically 20C–35C (can deliver 66A–115A peak)
Connector Type	XT60 / Deans / JST / Custom (varies by model)
Charging Connector	JST-XH (for balanced charging)
Weight	~200–250 grams (depends on brand)
Dimensions	Approx. 130mm × 45mm × 25mm (varies)
Recharge Cycle Life	~200–300 cycles (with proper usage and care)

A **3S 3300mAh Li-Po battery** provides the ideal balance between power, portability, and runtime for mobile embedded systems like our Agro Bot. Its ability to deliver high currents while maintaining a relatively compact form makes it well-suited for rugged environments where stable, off-grid energy is essential. When managed properly with appropriate chargers and protection circuits, this battery type significantly enhances the efficiency and autonomy of field robotics in agriculture and beyond.

2.5. ROBOT CONTROLLER:

Fig. 10.0 – Flysky FS-i6s Controller

The **FlySky FS-i6S** is a **modern, 10-channel 2.4GHz digital radio transmitter**, widely used in RC (Radio-Controlled) drones, planes, cars, and increasingly in custom robotics projects due to its **versatility, affordability, and user-friendly interface**. It is an advanced version of the FS-i6 and is known for its **sleek touchscreen display**, **bidirectional communication**, and **compact ergonomic design**.

In the context of an agriculture-focused robotic platform, the FS-i6S can serve as the **manual control interface**, allowing a user to remotely operate the robot's motion, steer it across fields, or even switch between modes such as soil sampling or data transmission. Its wide channel support and modular firmware make it well-suited for tasks that require multiple control inputs.

Here are the technical specifications:

Parameter	**Specification**
Operating Frequency	2.4GHz (AFHDS 2A)
Number of Channels	6–10 (expandable via firmware)
Modulation Mode	GFSK (Gaussian Frequency Shift Keying)
Output Power	< 20 dBm
Transmission Distance	~500–1000 meters (line of sight)
Power Supply	3.7V–8.4V (2S Li-ion or USB charging)
Current Consumption	~80 mA
Display	3.5" Backlit Capacitive Touchscreen
Binding Method	1-key binding with iA6B/iA10B/iA6C receivers
Firmware Upgradable	Yes (via USB)
Weight	~410 grams

The FS-i6S communicates wirelessly with a **FlySky-compatible receiver** (e.g., FS-iA6B, FS-iA10B, or FS-X6B), which is connected to the robot's microcontroller or motor controller.

- Each channel outputs a **PWM signal** or **PPM/iBUS/SBUS signal**, depending on receiver and firmware.
- These signals can be interpreted by an **Arduino Uno**, **ESP32**, or a custom motor controller to drive motors, trigger sensors, or change modes.
- In your Agro Bot, you can assign:
    - **Two channels for motion control** (forward/backward, left/right)
    - **Other channels for triggering sensors, adjusting sampling arms, or switching operation modes**

The **FlySky FS-i6S controller** is a **powerful and intuitive RC transmitter** designed for both hobbyists and professionals working in robotics, drones, and automation. With its **touchscreen interface, telemetry support, long range, and multi-channel capability**, it becomes an excellent choice for **field-ready agricultural robots** that require **manual control, override features, or direct intervention** during sensitive tasks like soil testing and data logging. Its integration with modern microcontrollers like the ESP32 further expands its utility, enabling the creation of hybrid systems that blend manual and autonomous operation effectively.

3. SOFTWARE SPECIFICATION:

3.1. ARDUINO IDE:

The **Arduino Integrated Development Environment (IDE)** is the **official software platform** used for writing, compiling, and uploading code to Arduino boards such as the **Arduino Uno**, **Mega**, **Nano**, and other compatible microcontrollers. Developed and maintained by the Arduino.cc team, the IDE is designed to be **simple, open-source, and beginner-friendly**, yet powerful enough for advanced embedded development.

At its core, the Arduino IDE allows users to program microcontrollers using a **simplified version of C/C++**, known as **Arduino language**. This environment significantly lowers the barrier to entry for students, hobbyists, and engineers alike by abstracting the complexities of microcontroller programming through a highly accessible interface.

Here are some key features of Arduino IDE:

Feature	**Description**
Sketch-Based Programming	Code written in the IDE is organized into "sketches," which consist of `setup()` and `loop()` functions.
Board Manager	Supports a wide range of official and third-party boards, including ESP32, STM32, and ATtiny.
Library Manager	Easily integrate external libraries for sensors, communication protocols, displays, etc.
Serial Monitor & Plotter	Provides real-time debugging and visualization of data from the microcontroller.
One-Click Uploading	Compile and upload code to connected devices via USB with a single click.
Syntax Highlighting	Visual aids like color-coded syntax and auto-indentation improve code readability.
Extensible	Custom boards, cores, and libraries can be added to support non-Arduino hardware.

In your **Soil Testing Agro Bot**, the Arduino IDE is used to:

- Write code to **interface sensors** like the DHT11 (for temperature and humidity), MQ gas sensors, and soil moisture sensors.
- Control **motors and actuators** via motor drivers or custom motor controller circuits.
- Establish **communication protocols** such as UART (for ESP32–Arduino data transfer).
- Upload data to the **cloud via ESP32**, using libraries like `WiFi.h` and `HTTPClient.h`.
- Monitor live sensor data during testing using the **Serial Monitor** or **Serial Plotter**.

The Arduino IDE supports thousands of open-source libraries, allowing easy integration of:

- **Sensor Libraries** (`DHT.h`, `Adafruit_Sensor.h`, `MQUnifiedsensor.h`)
- **Communication Libraries** (`WiFi.h`, `SoftwareSerial.h`, `HTTPClient.h`)

- **Cloud Libraries** (e.g., `ThingSpeak.h` for sending data to the ThingSpeak cloud)
- **Motor Control** (`Servo.h`, `AFMotor.h`, `AccelStepper.h`)

While the IDE is built for Arduino boards, it can also program other microcontrollers like:

- **ESP32 and ESP8266** (by installing respective board packages via Board Manager)
- **STM32, ATtiny, and SAMD-based boards**
- **Custom hardware** via third-party cores and configuration files

This flexibility allows projects like yours to seamlessly integrate **Arduino Uno and ESP32 in the same development environment**.

The **Arduino IDE** remains the **foundation for accessible and efficient embedded development**, especially in educational, prototyping, and research contexts. Its use in our agricultural robot project not only facilitates **sensor interfacing, motor control, and cloud communication**, but also ensures a **streamlined development cycle** through its plug-and-play nature. Whether used by beginners exploring electronics or by advanced developers integrating multi-sensor automation, the Arduino IDE continues to be a **cornerstone tool** in the expanding world of **IoT, robotics, and automation**.

3.2. THINGSPEAK:

ThingSpeak is a cloud-based Internet of Things (IoT) platform that allows users to collect, store, analyze, and visualize sensor data in real-time. Developed and maintained by **MathWorks**, the creators of MATLAB, ThingSpeak offers a robust environment where developers and researchers can effortlessly connect embedded devices like **Arduino**, **ESP32**, **Raspberry Pi**, and others to the internet and **stream live data**.

ThingSpeak plays a pivotal role in **remote sensing and monitoring applications**—particularly in smart agriculture—by enabling automated data logging from environmental sensors, and providing tools for **data analysis**, **triggering alerts**, and **visual representation** through graphs and dashboards.

In our **Soil Testing for Agriculture using Agro Bot**, ThingSpeak acts as the **central cloud server** to which data from the robot—such as **temperature, humidity, soil moisture**, and **gas levels**—is uploaded through the ESP32 microcontroller. This allows users (like farmers or researchers) to:

- Monitor real-time environmental data remotely
- Visualize the soil conditions through easy-to-read graphs
- Make informed decisions on **irrigation**, **fertilizer use**, or **air quality**
- Track historical trends in field parameters over time

Some core features of the software are:

Feature	Description
Real-Time Data Logging	Continuously receive and store data from sensors and IoT devices.
Visualization	Line charts, bar graphs, area plots, gauges, and more, displayed on customizable dashboards.
MATLAB Analytics	Built-in support for MATLAB code for advanced analysis and predictions.
Alerts and Actions	Automatically trigger alerts, tweets, or HTTP requests based on data thresholds.
Data Export	Data can be downloaded in CSV, JSON, or XML formats for external use.
API Integration	Provides RESTful APIs for writing and reading data from any internet-enabled device.

Here are the steps in which ThingSpeak works:

- **Channel Creation**: A ThingSpeak user creates a **channel**, which acts as a container for storing data. Each channel supports up to **8 data fields**, a **status field**, and **location metadata** (latitude, longitude, elevation).
- **API Key Generation**: ThingSpeak provides a **Write API Key** (for uploading data) and **Read API Key** (for retrieving data) to ensure secure access to the channel.
- **Data Upload**: The ESP32 sends data using HTTP requests like: `https://api.thingspeak.com/update?api_key=XXXX&field1=23.5&field2=70&field3=wet&field4=lowCO`
- **Real-Time Visualization**: The uploaded data is displayed on **graphs** or **gauges**, which update automatically.
- **Analytics (Optional)**: MATLAB code can be embedded into the channel for processing raw data—for example, calculating daily averages or predicting irrigation needs.

ThingSpeak is a powerful, flexible, and user-friendly cloud IoT platform that bridges the gap between **hardware-level data collection** and **web-based visualization and analysis**. Its use in agricultural robotics, such as your **Soil Testing Agro Bot**, enables real-time monitoring, data-driven decision-making, and long-term environmental tracking—all essential elements of **smart farming** in the modern age. By integrating it with an ESP32, you gain a **cost-effective and scalable solution** for turning raw sensor data into actionable insights, making your project not just functional, but impactful.

CIRCUIT DIAGRAM

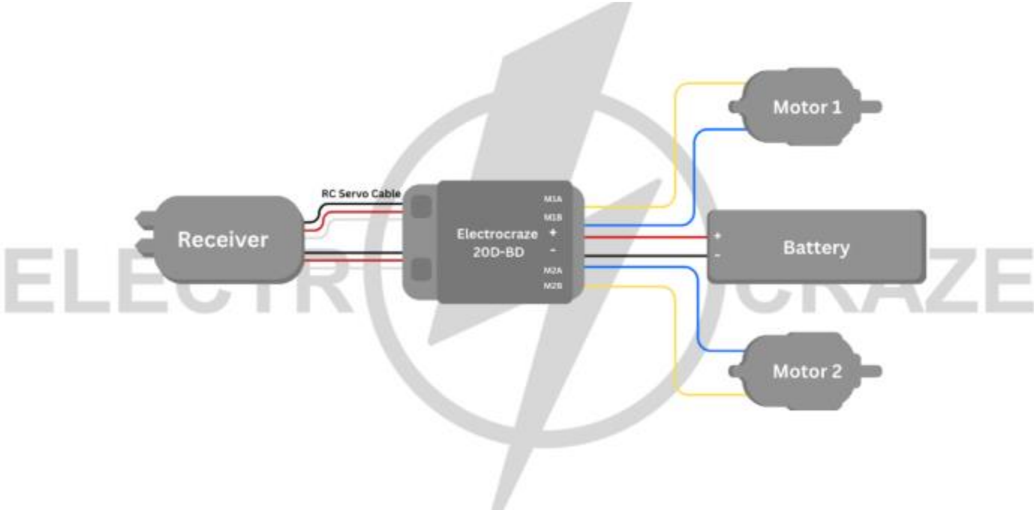
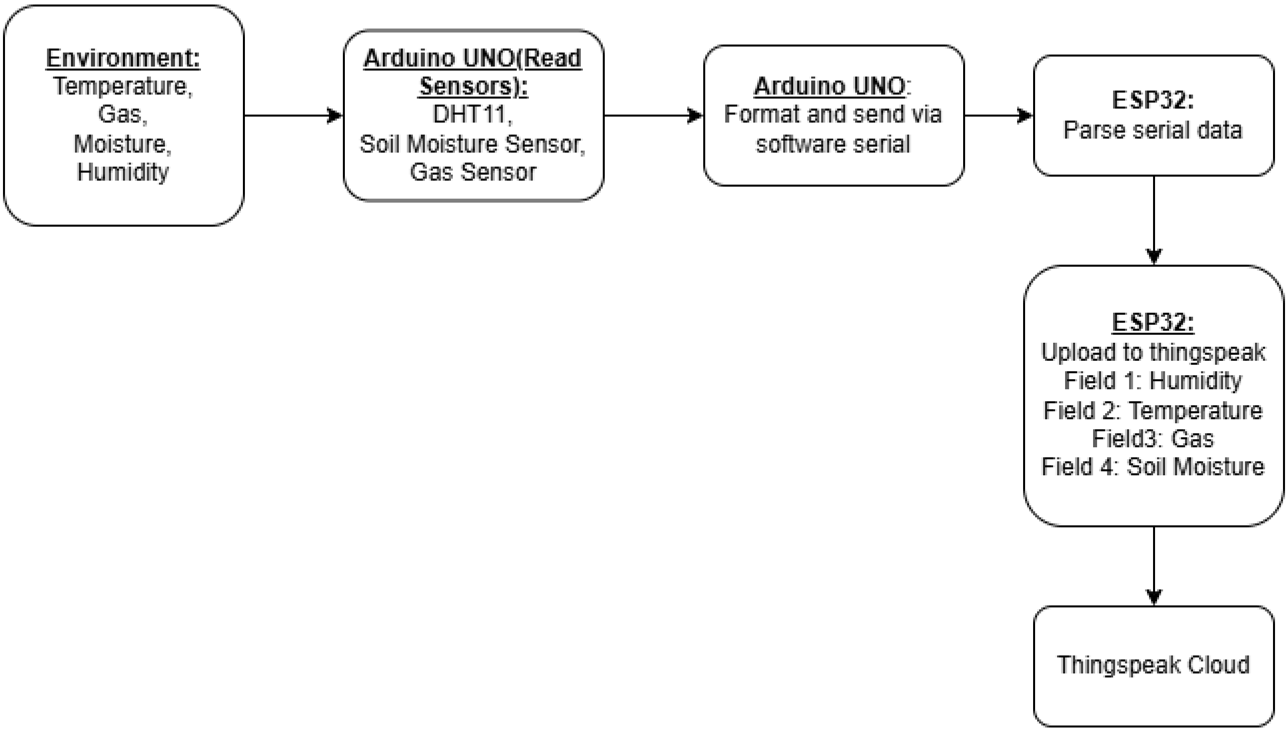


Fig. 11.0 – Circuit Diagram of connection in the Robot

Fig. 11.1 – Circuit Diagram of connecting sensors
to Arduino UNO and ESP32

DATAFLOW DIAGRAM



WORKING PRINCIPLE

The **Soil Testing Agro Bot** operates as a ground-based, sensor-equipped mobile robot designed to assess key soil and environmental parameters in real-time. It combines the functionalities of **embedded electronics, wireless communication, remote control**, and **IoT-based cloud integration** to deliver actionable insights for smart farming.

The working principle is described in several interconnected stages:

1. System Initialization:

Upon powering on the Agro Bot using a **3S Li-Po battery (11.1V, 3300 mAh)**:

- The **Arduino Uno** initializes all connected sensors including the **DHT11 (temperature & humidity)**, **soil moisture sensor**, and **MQ series gas sensor**.
- Simultaneously, the **ESP32** microcontroller boots up and attempts to connect to a predefined **Wi-Fi network**.
- The **FSi6s receiver**, linked to the user's handheld controller, activates to receive movement commands.

2. Manual Navigation:

The bot is **manually navigated** by the user using the **FSi6s controller**:

- The **motor controller circuit**, interfaced with the Arduino, interprets the controller's signals and drives the **DC Johnson motors** accordingly.
- The robot moves across the agricultural field, stopping at selected points for soil analysis.

This phase gives users full control over where data is collected—ensuring coverage across critical field zones.

3. Sensor Data Acquisition:

As the robot traverses the field:

- The **DHT11 sensor** captures **temperature and humidity** from the surrounding air. These values are critical for understanding the microclimatic condition affecting crop growth.
- The **soil moisture sensor**, inserted slightly into the ground through a probe or mounted arm, measures the **volumetric water content** of the soil. This helps determine if irrigation is necessary.
- The **MQ gas sensor** detects the presence and concentration of gases such as **methane, ammonia, or carbon monoxide** in the soil. Abnormal gas levels may indicate chemical imbalance or contamination.

Each sensor outputs either analog or digital signals, which are read and interpreted by the **Arduino Uno**.

4. Data Processing and Serial Communication:

- The Arduino processes the raw sensor data and converts them into meaningful units (e.g., °C for temperature, % for humidity/moisture, ppm for gas).
- The processed data is then sent to the **ESP32** via **serial communication (UART)** using TX-RX lines.
- At this point, the Arduino acts as a **data aggregator**, while the ESP32 functions as the **IoT communication gateway**.

5. Wi-Fi Connectivity and Cloud Upload:

Once the ESP32 receives the data:

- It connects to the internet using the **Wi-Fi module** built into the board.
- The ESP32 formats the data into an HTTP GET request compatible with **ThingSpeak's REST API**.
- The data is uploaded to a **ThingSpeak channel** using a valid **Write API Key**.

For example: https://api.thingspeak.com/update?api_key=XYZ&field1=25.6&field2=65&field3=wet&field4=10ppm

This request logs temperature, humidity, soil moisture status, and gas level into four separate fields in the cloud.

6. Real-Time Monitoring and Visualization:

After successful data transmission:

- **ThingSpeak** stores the uploaded values in a time-series format.
- Users can **view real-time graphs** and **data visualizations** on a web dashboard or mobile browser.
- These visualizations help in:
    - Monitoring current field conditions
    - Detecting dry patches in the soil
    - Spotting temperature or gas anomalies
    - Making irrigation or fertilization decisions

The dashboard updates with every new data packet received, usually every 15–20 seconds depending on the update frequency.

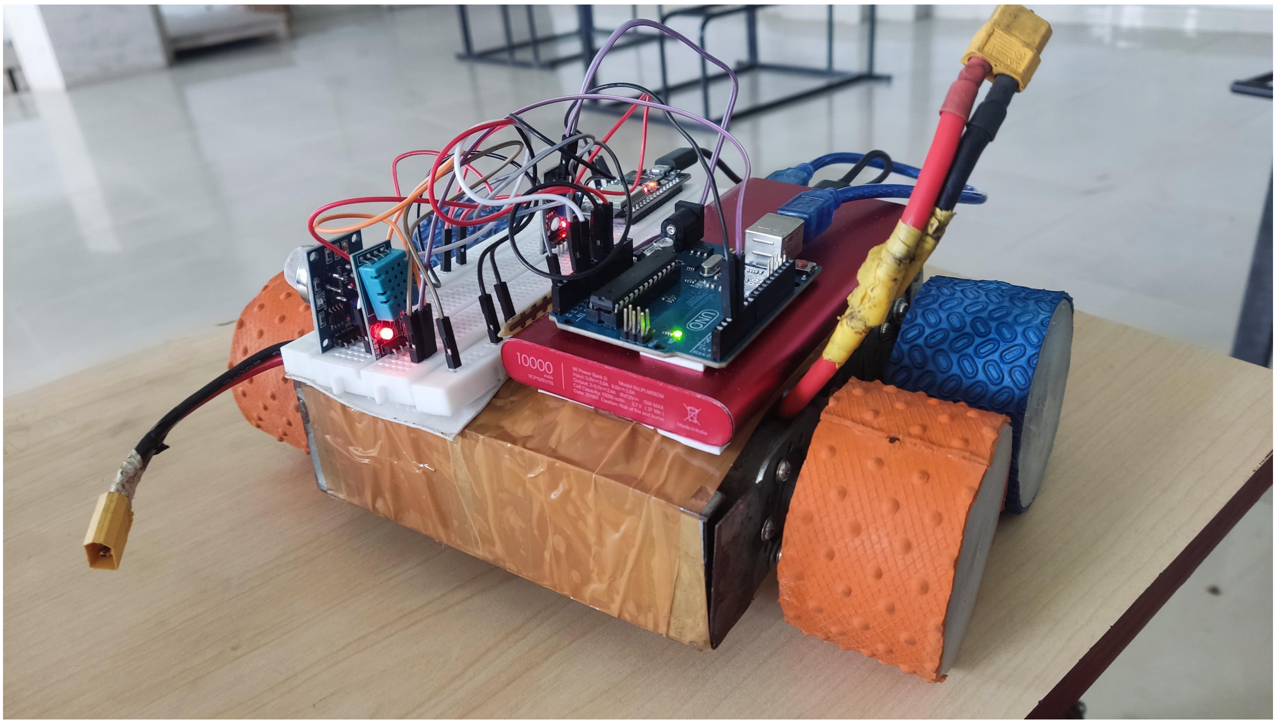
7. Loop and Continuous Operation:

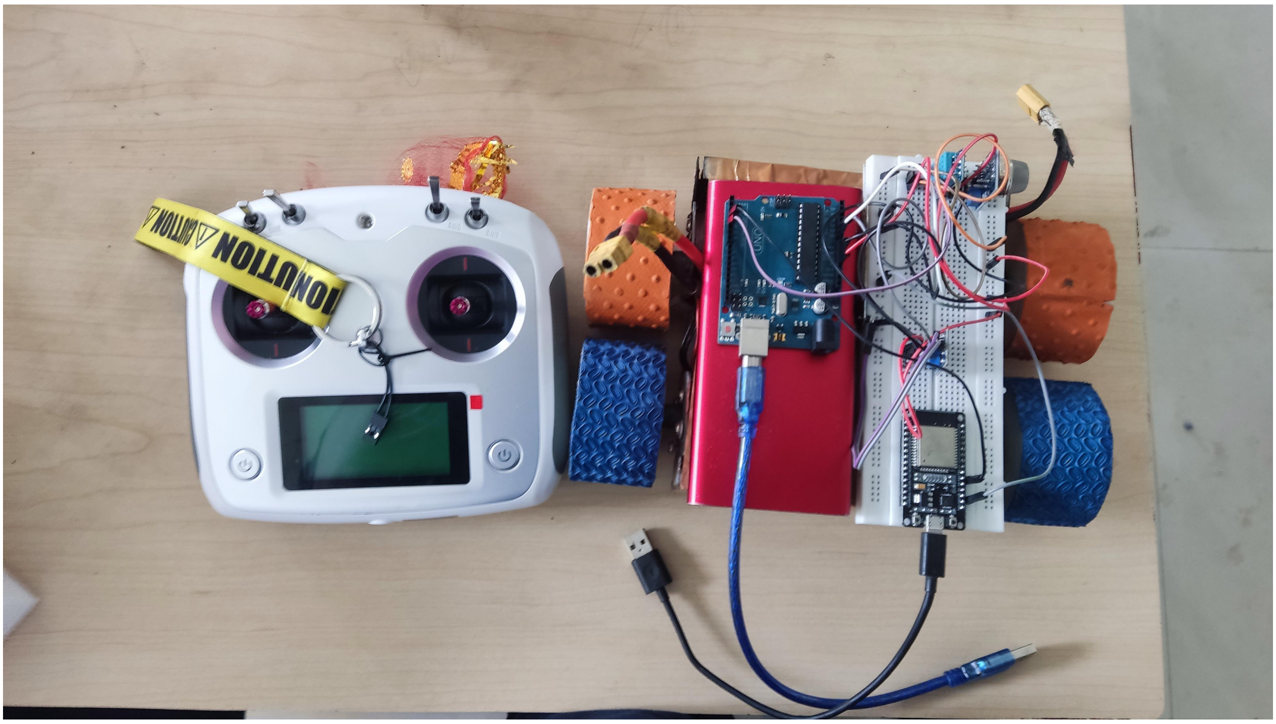
The process repeats in a continuous loop:

- The robot moves to a new location in the field.

- Sensors acquire fresh data.
- Arduino sends data to ESP32.
- ESP32 transmits data to ThingSpeak.
- Users monitor live updates.

This real-time loop ensures **continuous environmental surveillance** with minimal user intervention beyond movement control.


10000



FUTURE SCOPE

As agriculture faces growing challenges such as climate change, soil degradation, unpredictable weather, and increasing food demands, integrating technology into farming practices has become more crucial than ever. The current Agro Bot system, while already effective for real-time soil analysis, has vast potential for future development and scalability. The following aspects outline the promising avenues in which this project can evolve:

1. Autonomous Navigation and AI Integration

Currently, the Agro Bot is manually controlled via an FSi6s transmitter. In the future, this robot can be upgraded to function autonomously using technologies such as:

- **GPS-based waypoint navigation** to allow the bot to move automatically across large agricultural fields.
- **AI-based decision-making systems** to identify patterns in soil data and determine optimal paths and sample points.
- **Obstacle detection and avoidance** using ultrasonic, LiDAR, or vision-based sensors to improve field mobility and safety.
  This will minimize human intervention and allow 24/7 data collection, even in remote or hazardous conditions.

2. Enhanced Sensor Suite for Advanced Soil Analysis

While the current model measures temperature, humidity, soil moisture, and gas levels, future versions can incorporate a more diverse range of sensors for comprehensive soil profiling:

- **Soil pH sensors** to monitor acidity/alkalinity affecting crop growth.
- **Electrical conductivity sensors** to measure soil salinity.
- **Nutrient sensors** to assess the presence of essential elements like nitrogen, phosphorus, and potassium (NPK).
- **Sunlight/UV sensors** to analyze light conditions. Adding these sensors will enable more accurate crop and fertilizer planning, improving productivity and reducing environmental harm.

3. Integration with Irrigation and Fertilization Systems

One of the most impactful upgrades would be connecting the Agro Bot to **automated irrigation and fertilization systems**:

- Based on real-time soil moisture data, the bot can trigger **drip irrigation systems** to supply precise water quantities.
- Data from nutrient or pH sensors can activate **variable-rate fertilization**, delivering targeted soil amendments only where needed. This would reduce resource waste, save water, and ensure optimal plant health.

4. Cloud Analytics and Predictive Modeling

Using platforms like **ThingSpeak**, **MATLAB**, or **Google Cloud**, the uploaded data can be analyzed using:

- **Machine learning models** to predict soil trends or anticipate irrigation needs.
- **Time-series forecasting** to track seasonal variations and optimize planting schedules.
- **Dashboards and mobile apps** for farmers to receive actionable insights, alerts, and recommendations.

Over time, a central database can be built to support **region-specific farming guidelines** and help other farmers in similar geographies.

5. Scalability for Larger and Commercial Farms

The current model is suited for small or experimental plots, but it can be scaled for larger farms by:

- Deploying **multiple coordinated robots** working in a grid pattern across hectares.
- Using **cloud-based fleet management** systems to control and monitor several bots simultaneously.
- Enhancing battery life or adopting **solar charging stations** in the field for longer autonomous operation.

This would bring automation to commercial farming, reducing labour and increasing yield predictability.

6. Integration with Weather and Satellite Data

To provide context to the soil data, the bot can be linked to:

- **Weather APIs** for real-time temperature, rainfall, and wind predictions.
- **Satellite imagery** for NDVI (Normalized Difference Vegetation Index) and remote crop monitoring.
  Combining ground data with aerial and environmental data can help create a **complete precision agriculture ecosystem**.

7. Educational and Research Applications

The Agro Bot has excellent potential in:

- **Agricultural universities** for research on crop-soil interaction and sustainable farming.
- **Training tools** for teaching IoT, robotics, and data science in agriculture.
- **Pilot projects** in rural or resource-limited regions to educate farmers and promote adoption of smart farming techniques.

8. Mobile App Development

A future mobile application can allow:

- **Real-time data access**
- **Bot control interface**
- **Push notifications for thresholds**
- **Data logging and export in Excel or PDF**

Such integration would offer farmers or researchers mobility, convenience, and better user experience.

9. Green Energy Adaptation

To ensure sustainability, future versions of the Agro Bot can be powered using:

- **Onboard solar panels** for energy harvesting.
- **Energy-efficient components** and sleep modes for non-operational hours. This aligns the project with global goals for eco-friendly and low-carbon agricultural practices.

CONCLUSION

The project **“Soil Testing for Agriculture using Agro Bot”** represents a meaningful advancement in the application of modern technology to address traditional agricultural challenges. By integrating core concepts of the **Internet of Things (IoT)**, **embedded systems**, **robotics**, and **wireless communication**, this project successfully demonstrates how real-time soil and environmental monitoring can be automated, digitized, and made accessible remotely.

The robot is designed as a ground-based mobile platform fitted with essential sensors such as the **DHT11 temperature and humidity sensor**, **soil moisture sensor**, and **MQ series gas sensor**. These components are managed by an **Arduino Uno**, which collects, processes, and transmits data to an **ESP32** module. The ESP32, with built-in Wi-Fi capability, sends this sensor data to a **ThingSpeak cloud platform** where it is visualized in real time. This allows users, such as farmers or agricultural experts, to monitor critical soil and atmospheric parameters from any location through a web interface or mobile device.

Manual control via an **FSi6s transmitter and receiver system** ensures that the robot can be moved across different parts of the field to collect localized soil data. This not only allows for targeted analysis but also supports decision-making for site-specific irrigation, fertilization, and crop management. The use of a **custom motor driver**, **DC Johnson motors**, and a **3S Li-Po battery** provides a robust and reliable hardware setup suitable for field conditions.

Through this approach, the project significantly reduces human effort, enhances accuracy, and minimizes guesswork in soil analysis. It addresses key agricultural concerns such as **water scarcity**, **soil degradation**, and **crop productivity** by empowering farmers with real-time insights into their land's health. The integration with ThingSpeak adds a layer of intelligence, enabling data logging, trend analysis, and potentially even predictive alerts when thresholds are breached.

Furthermore, the project opens up avenues for future innovation. By incorporating additional sensors (e.g., pH, NPK, electrical conductivity), AI-based data analytics, autonomous navigation, and integration with automated irrigation systems, this solution can evolve into a fully autonomous precision agriculture platform. It also holds promise in research, education, and government-supported rural development initiatives aimed at digital farming transformation.

In conclusion, the **Soil Testing Agro Bot** is more than a prototype—it is a step toward **sustainable, efficient, and data-driven agriculture**. It demonstrates how even small-scale IoT solutions can bring about significant improvements in farming practices, helping to meet the rising global demand for food while preserving natural resources for future generations.