A Mini project report

On

SMART SOIL ANALYSIS ROVER

Project Submitted to Jawaharlal Nehru Technological University Hyderabad In partial fulfillment of the Academic Requirement for the award of

Bachelor of Technology

In

Electronics & Communication Engineering

Submitted by

K. CHANDRA SHEKAR	**235U5A0409**
B. KRUTHIKA	**235U5A0402**
Y. HARISH REDDY	**225U1A04C6**
CH. POOJITHA	**235U5A0404**

Under the guidance of

Dr. G. DHANALAKSHMI

Head-R&D, Professor, Dept of E.C.E

Department of Electronics and Communication Engineering

AVN INSTITUTE OF ENGINEERING & TECHNOLOGY

(Approved by A.I.C.T.E. & Govt. of Telangana & Affiliated to JNTU Hyderabad)

Accredidated by NAAC & NBA

PATEL GUDA, IBRAHIMPATANAM (M), R.R-Dist-501510-TS

2025-26

AVN INSTITUTE OF ENGINEERING & TECHNOLOGY
Accredited by NAAC & NBA

CERTIFICATE

This is to certify that the dissertation entitled **"Smart Soil Analysis Rover"** is a bonafide work done and submitted by

K. CHANDRA SHEKAR	**235U5A0409**
B. KRUTHIKA	**235U5A0402**
Y. HARISH REDDY	**225U1A04C6**
CH. POOJITHA	**235U5A0404**

in partial fulfillment of requirement for the award of **Bachelor of Technology degree** in **Electronics and Communication Engineering** at AVN Institute of Engineering & Technology, Hyderabad.

Certified further that to the best of our knowledge the work in this dissertation has not been submitted to any other university or institute for the award of any degree.

COORDINATOR

Mrs. B. Mamatha,
Assistant Professor
Dept of E.C.E

COORDINATOR

Mrs. M. Arpana,
Assistant Professor
Dept of E.C.E

HEAD OF THE DEPARTMENT

Dr. N.Suresh,
Professor & H.O.D
Dept of E.C.E

INTERNAL GUIDE

Mrs. G. Dhanalakshmi,
Professor & Head – R&D
Dept of E.C.E

EXTERNAL EXAMINER

AVN INSTITUTE OF ENGINEERING & TECHNOLOGY
Accredited by NAAC & NBA

DECLARATION

We, hereby declare that the work which is being presented in this dissertation entitled **"SMART SOIL ANALYSIS ROVER",** being submitted by us for the award of degree of Bachelor of Technology in Department of Electronics and Communication Engineering at **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD** is a bonafide record of **AVN INSTITUTE OF ENGINEERING AND TECHNOLOGY** and has not been submitted to any other course or university for award of any degree.

SUBMITTED BY

K. CHANDRA SHEKAR	**235U5A0409**
B. KRUTHIKA	**235U5A0402**
Y. HARISH REDDY	**225U1A04C6**
CH. POOJITHA	**235U5A0404**

AVN INSTITUTE OF ENGINEERING & TECHNOLOGY
Accredited by NAAC & NBA

ACKNOWLEDGEMENT

We are greatly thankful to the following people for their contribution, advice and assistance towards the development of this project and also in the completion of the report entitled **"SMART SOIL ANALYSIS ROVER"**.

We express our sincere thanks to our co-coordinators **Mrs. B. Mamatha,** Assistant Professor & **Mrs. M. Arpana,** Assistant Professor for the guidance, they showed us and for their sincere efforts in building the project right direction.

We express our sincere thanks to our Guide **Dr. G. Dhanalakshmi,** Head – R&D, Professor for the guidance, she showed us and for her sincere efforts in building the project in right direction.

We are indebted to **Dr. N. Suresh,** Head of the Department of ECE for his valuable suggestion and for the motivation he provided. Our sincere thanks to him for sparing his valuable time and constant encouragement throughout the project.

We are highly indebted to Principal, **Dr. P. Nageswara Reddy,** for the support who has not only shown utmost patience, but fertile in suggestions, vigilant in directions of error and who have been infinitely helpful.

K. CHANDRA SHEKAR	**235U5A0409**
B. KRUTHIKA	**235U5A0402**
Y. HARISH REDDY	**225U1A04C6**
CH. POOJITHA	**235U5A0404**

ABSTRACT

The **Smart Soil Analysis Rover** is a mobile robotic system developed for efficient and real-time analysis of soil conditions in agricultural fields, construction zones, and environmental monitoring sites. It integrates key sensors to measure soil moisture, temperature, humidity, and detect metal presence. A six-wheel chassis ensures stable movement across uneven terrain, while a servo-powered jaw enables sample collection from hard-to-reach areas.

Sensor data is displayed locally and transmitted via Bluetooth and Wi-Fi to cloud platforms like **Think View** for remote visualization and analysis. The system is built using cost-effective and modular components, making it easily maintainable and upgradable. Future enhancements aim to include autonomous navigation and improved durability, positioning the rover as a valuable tool for precision agriculture, site analysis, and research applications.

The project demonstrates how embedded systems, IoT integration, and basic robotics can be combined to create a scalable and practical solution for field-based soil monitoring. Its real-time feedback and dual control modes make it suitable for both educational and industrial applications.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The project is aimed at evaluating the performance of an operating system on an embedded system. Before delving into its implementation, an introduction is needed to the parts involved in the project. The whole report is centered around the field of embedded systems and the use of Linux to run applications on them. Hence an introduction to Embedded Systems and using Linux as an OS in them is provided.

1.1 Embedded Systems

An embedded system is a special purpose computer system that is designed to perform very small sets of designated activities. Embedded systems date back as early as the late 1960s where they used to control electromechanical telephone switches. The first recognizable embedded system was the Apollo Guidance Computer developed by Charles Draper and his team. Later they found their way into the military, medical sciences and the aerospace and automobile industries.

Today they are widely used to serve various purposes like:

• Network equipment such as firewall, router, switch, and so on.

• Consumer equipment such as MP3 players, cell phones, PDAs, digital cameras, camcorders, home entertainment systems and so on.

• Household appliances such as microwaves, washing machines, televisions and so on.

• Mission-critical systems such as satellites and flight control.

The key factors that differentiate an embedded system from a desktop computer:

• They are cost sensitive.

• Most embedded systems have real time constraints.

• There are multitudes of CPU architectures such as ARM, MIPS, PowerPC that are used in embedded systems. Application-specific processors are employed in embedded systems.

• Embedded Systems have and require very few resources in terms of ROM or other I/O devices as compared to a desktop computer.

1.1.1 Types of Setup

Embedded systems generally have a setup that includes a host which is generally a personal computer, and a target that actually executes all the embedded applications. The various types of host/ desktop architectures that are used in embedded systems are:

Linked Setup:

In this setup, the target and the host are permanently linked together using a physical cable. This link is typically a serial cable or an Ethernet link. The main property of this setup is that no physical hardware storage device is being transferred between the target and the host. The host contains the cross-platform development environment while the target contains an appropriate bootloader, a functional kernel, and a minimal root filesystem.

Removable Storage Setup:

In the removable setup, there are no direct physical links between the host and the target. Instead, a storage device is written by the host, is then transferred into the target, and is used to boot the device. The host contains the cross-platform development environment. The target, however, contains only a minimal bootloader. The rest of the components are stored on a removable storage media, such as a CompactFlash IDE device, MMC Card, or any other type of removable storage device.

Standalone Setup:

The target is a self-contained development system and includes all the required software to boot, operate, and develop additional software. In essence, this setup is similar to an actual workstation, except the underlying hardware is not a conventional workstation but rather the embedded system itself. This one does not require any cross-platform development environment, since all development tools run in their native environments. Furthermore, it does not require any transfer between the target and the host, because all the required storage is local to the target.

1.2 Operating Systems

In an embedded system, when there is only a single task that is to be performed, then only a binary is to loaded into the target controller and is to be executed. However, when there are multiple tasks to be executed or multiple events to be handled, then there has to be a program that handles and prioritizes these events. This program is the Operating System (OS), which one is very familiar with, in desktop PCs.

Various Operating Systems:

Embedded Operating Systems are classified into two categories:

1. Real-time Operating Systems (RTOS) :

Real Time Operating Systems are those which guarantee responses to each event within a defined amount of time. This type of operating system is mainly used by time-critical applications such as measurement and control systems. Some commonly used RTOS for embedded systems are: VxWorks, OS-9, Symbian, RTLinux.

2. Non-Real-time Operating Systems:

Non-Real Time Operating Systems do not guarantee defined response times. These systems are mostly used if multiple applications are needed. Windows CE and PalmOS are examples for such embedded operating systems.

Why Linux?

There are a wide range of motivations for choosing Linux over a traditional embedded OS.

The following are the criteria due to which Linux is preferred:

1. Quality and Reliability of Code:

Quality and reliability are subjective measures of the level of confidence in the code that comprises software such as the kernel and the applications that are provided by distributions. Some properties that professional programmers expect from a “quality” code are modularity and structure, readability, extensibility and configurability. “Reliable” code should have features like predictability, error recovery and longevity. Most programmers agree that the Linux kernel and other projects used in a Linux system fit this description of quality and reliability. The reason is the open source development model, which invites many parties to contribute to projects, identify existing problems, debate possible solutions, and fix problems effectively.

2. Availability of Code:

Code availability relates to the fact that the Linux source code and all build tools are available without any access restrictions. The most important Linux components, including the kernel itself, are distributed under the GNU General Public License (GPL).Access to these components' source code is therefore compulsory (at least to those users who have purchased any system running GPL-based software, and they have the right to redistribute once they obtain the source in any case). Code availability has implications for standardization and commoditization of components, too. Since it is possible to build Linux systems based entirely upon software for which source is available, there is a lot to be gained from adopting standardized embedded software platforms.

3. Hardware Support:

Broad hardware support means that Linux supports different types of hardware platforms and devices. Although a number of vendors still do not provide Linux drivers, considerable progress has been made and more is expected. Because a large number of drivers are maintained by the Linux community itself, you can confidently use hardware components without fear that the vendor may one day discontinue driver support for that product line. Linux also provides support for dozens of hardware architectures. No other OS provides this level of portability.

Typical architecture of an Embedded Linux System

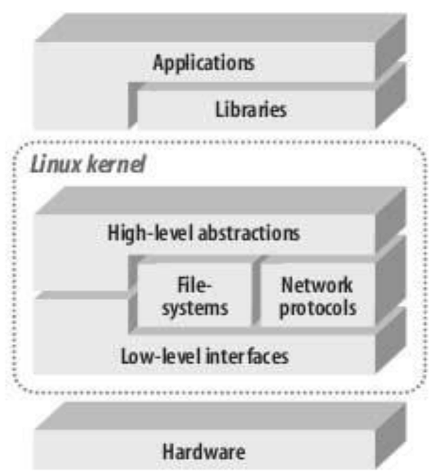


Figure 1.1 Architecture of an Embedded Linux System

1. Hardware

Linux normally requires at least a 32-bit CPU containing a memory management unit (MMU). A sufficient amount of RAM must be available to accommodate the system. Minimal I/O capabilities are required if any development is to be carried out on the target with reasonable debugging facilities. The kernel must be able to load a root filesystem through some form of permanent storage, or access it over a network.

2. Linux Kernel

Immediately above the hardware sits the kernel, the core component of the operating system. Its purpose is to manage the hardware in a coherent manner while providing familiar high-level abstractions to user-level software. It is expected that applications using the APIs provided by a kernel will be portable among the various architectures supported by this kernel with little or no changes. The low-level interfaces are specific to the hardware configuration on which the kernel runs and provide for the direct control of hardware resources using a hardware-independent API. Higher-level components provide the abstractions common to all UNIX systems, including processes, files, sockets, and signals. Since the low-level APIs provided by the kernel are common among different architectures, the code implementing the higher-level abstractions is almost constant, regardless of the underlying architecture. Between these two levels of abstraction, the kernel sometimes needs what could be called interpretation components to understand and interact with structured data coming from or going to certain devices. Filesystem types and networking protocols are prime examples of sources of structured data the kernel needs to understand and interact with in order to provide access to data going to and coming from these sources.

3. Applications and Libraries

Applications do not directly operate above the kernel, but rely on libraries and special system daemons to provide familiar APIs and abstract services that interact with the kernel on the application's behalf to obtain the desired functionality. The main library used by most Linux applications is the GNU C library, glibc. For Embedded Linux systems, substitutes to this library that are much less in size than glibc are preferred.

CHAPTER 2
LITERATURE SURVEY

2.1 Introduction to Literature Survey

Technological advancements in robotics and sensor systems have revolutionized the field of precision agriculture, enabling farmers and environmental researchers to obtain real-time data about soil conditions. Mobile robotic systems equipped with environmental sensors provide a dynamic approach to in-situ soil monitoring, which is crucial for crop management, construction site evaluation, and environmental conservation. Literature from recent years emphasizes the importance of integrating embedded hardware, wireless communication, and cloud-based data platforms for comprehensive soil analysis.

The review was completed by identifying current research in the field of study, extracting the necessary material, and synthesising the data for the final step of review elaboration. the selected papers were published in the time frame of 2015 to 2023. Regarding the selection of the 120 papers about robots in agriculture, the following keywords were used: “Soil testing”, “Agricultural rover”, “Automation”, and “Robotics”.

Jie Wei, et al. demonstrated the ruggedness of the MMRS (Mars Micro-beam Raman Spectrometer) in a 10- day 2013 LITA field campaign at the Atacama Desert. The automated soil sample analysis made by the MMRS unambiguously identified a variety of igneous minerals (quartz, feldspars, and TiO_2 polymorphs), carbonates, sulphates, and carbonaceous materials. The field-MMRS identifications were confirmed by laboratory Raman analysis, which showed that the MMRS has comparable performance as the laboratory instrument when its focusing condition was satisfied. Quantified distributions of major minerals and carbonaceous materials are extracted from the measurements, which can indicate regional geological evolution and potential bioactivities.

2.1.1 INTEGRATION OF GPS

Mohammed Z. Al-Faiz., and **Ghufran E. Mahameda**., [2] observed that Using data from the GPS and digital compass sensors, the mobile robot correctly determines the route between its starting point and its destination. It has been demonstrated that the mobile robot's five Sharp IR sensors allowed it to recognise and steer clear of challenging obstructions in its route. When a PID controller was used, the mobile robot moved in a straight line at a precise angle towards the goal rather than following a predetermined course and taking longer to get there.

Without one, the robot deviates from the chosen path and travels around the target. The disturbances are reduced to 50% and the mobile robot's speed is reduced to 33.3% by regulating the speed of each side of its motors in accordance with the digital compass readings. Despite this, the robot moves smoothly and arrives at the destination in roughly the same amount of time without the need for PID**.**

2.1.2 INTEGRATION OF SENSORS

Satish Kumar V, et al. created a clever, self-governing gardening rover using a rocker-bogie suspension setup. Features include the ability to identify different types of plants, measure environmental variables like temperature, humidity, and moisture, and provide the right amount of water and fertiliser to promote plant growth. Utilising the information gathered by the rover, data analysis was carried out, and the outcome was integrated into the Android app and website to offer comprehensive statistics about the garden. In order to identify plants, they used neural networks in conjunction with ORB and FLANN matcher. Real-time and dataset photos were compared to identify the plants. Using JavaScript, Java, and Python, they created a website and app that provides users with information about the current weather, including chances of rain, sunrise and sunset times, actual temperature, wind direction, moon phase, pressure, and humidity.

They also included data from rover sensors, such as temperature, humidity, and soil moisture, as well as controls and real-time information about the rover's location and activity. Francesco Visentin, et al. presented an idea for a mixed-autonomous robotic system for weed detection and removal from crops. The system consists of a fully autonomous robotic system, a gantry robot, and RGB-D cameras. The system uses artificial intelligence and computer vision to accurately detect and eradicate weeds. The robot can be enhanced through control system, autonomous navigation, and identification system enhancement. Other sensors like lidar, sonar, and GPS RTK can be added for autonomous cart movement. The recognition system has seen significant software advancements, with the first division focusing on green vegetables. To address color issues, a neural network could be developed to identify and localize various plants in a single image. This prototype demonstrates the viability of a robotic weeding system that can identify and eradicate individual weeds, removing them from the field. This invention lays the foundation for further research and development to improve the effectiveness, safety, and financial viability of agricultural operations. It highlights the importance of innovation in this field and has the potential to advance sustainable agriculture.

2.1.3 ROBOTIC SYSTEM FOR IN-FIELD

Goran Kitić, et al. introduces **Agrobot Lala**, an autonomous robotic system for in-field, real-time nitrate analysis and soil sampling. The suggested approach aids in fertiliser optimisation, which promotes more productive and sustainable output. The system consists of a robotic system for collecting and analysing soil samples in the field, a smartphone application for task monitoring and customization, and a cloud-based application for task management and plotting sample points based on proprietary artificial intelligence (AI) algorithms.

A farmer can design tasks for soil collection and analysis using the system, which uses an AI algorithm and a cloud-based platform. The robotic system completes the work and provides the measurement findings instantly. A fertilisation prescription that exceeded 7.5% of KAN fertilizer savings and demonstrated a 1.76% yield enhancement in the initial test is produced as a result of the analysis.

Five soil samples were collected and examined in the 1 ha experimental area in order to confirm that the system performed as advertised.

2.2 Soil Monitoring Technologies

Soil quality and composition are vital indicators for agricultural productivity. Researchers have extensively implemented soil moisture sensors, temperature and humidity sensors, and metal detectors to assess various parameters. Moisture sensors, such as capacitive types, are preferred for their low power consumption and reliable readings. Temperature and humidity are commonly measured using sensors like DHT11 and DHT22, which offer satisfactory precision for field applications.

Metal detection technology is also gaining traction, especially in construction and post-industrial land monitoring. Metal sensors using inductive principles have proven effective in identifying subsurface metallic objects. These technologies assist in ensuring soil is free from hazardous materials and suitable for specific applications.

2.3 Robotic and Wireless Systems

Mobile platforms are increasingly used to navigate large areas and collect sensor data efficiently. Research indicates that six-wheel and track-based robotic rovers offer better terrain handling compared to traditional four-wheel designs. These systems are often powered by microcontrollers such as Arduino and Raspberry Pi, and are capable of interfacing with multiple sensor modules.

Wireless communication plays a vital role in transmitting sensor data. Bluetooth modules like HC-05 support short-range communication with smartphones, while Wi-Fi-enabled microcontrollers like ESP8266 (D1 Mini) allow long-range and cloud-integrated communication.

2.4 Existing System

Several soil monitoring systems have been developed using basic sensor modules mounted on stationary or mobile platforms. These systems primarily collect data on one or two soil parameters such as moisture or temperature. Most use Bluetooth-based transmission for local access to data via smartphones. However, these systems have significant limitations.

The common drawbacks of existing systems include:

- Limited sensor integration (usually only one or two parameters)
- Poor terrain adaptability due to fixed or four-wheel chassis
- Lack of real-time cloud connectivity
- Absence of physical soil sample collection mechanisms
- Limited operational range due to Bluetooth-only communication

While these existing methods provide basic monitoring capabilities, they do not support robust field deployment, particularly in uneven terrains and remote regions.

2.5 Proposed Method

The proposed system, **Smart Soil Analysis Rover**, is developed to overcome the limitations of conventional soil monitoring and sampling methods by combining advanced hardware integration with smart data transmission and control features. This rover is designed to operate in outdoor, rugged environments, particularly agricultural lands and research sites, providing real-time environmental and soil data while also allowing physical sample collection.

At the heart of the design is a **six-wheeled, terrain-adaptive chassis**, which ensures enhanced mobility and stability across various surface conditions. This configuration allows the rover to navigate over rough agricultural fields, loose soil, and mildly uneven ground without compromising balance or functionality.

The system is equipped with a **suite of sensors** for environmental and subsurface analysis:

- A **soil moisture sensor** measures the volumetric water content of the soil.
- A **DHT11 sensor** records temperature and humidity, aiding in climate-aware soil interpretation.
- A **metal detector module** identifies metallic objects beneath the surface, useful for assessing contamination or buried infrastructure.

To complement the sensor system, a **servo-powered robotic jaw mechanism** is installed on the rover's front end. This feature allows the rover to collect soil samples autonomously or via remote command, which can later be analyzed in laboratories for more detailed physical or chemical properties.

For communication, the rover employs both **Bluetooth (via HC-05)** for short-range manual control and **Wi-Fi (via ESP8266-based D1 Mini module)** for cloud connectivity. This dual-mode setup enables flexible usage in both field and lab environments. Collected data is not only displayed locally on an **LCD module** but also transmitted to the **ThinkSpeak IoT platform**, where users can monitor, analyze, and archive environmental trends remotely through web or mobile access.

This multi-functional system enables:
• Real-time monitoring of critical soil parameters through onboard and cloud displays
• Adaptability to rugged terrains using a stable six-wheel drive mechanism
• Seamless mobile and remote access to sensor data for effective decision-making and analysis

Overall, the **Smart Soil Analysis Rover** presents a scalable, robust, and economically feasible solution tailored for **precision agriculture**, **environmental monitoring**, and **civil engineering site assessments**. Its modular design allows for easy upgrades, such as adding GPS for location tagging, camera modules for visual inspection, or more advanced analytical sensors in future iterations.

CHAPTER 3

DESIGN AND IMPLEMETATION

3.1 BLOCK DIAGRAM

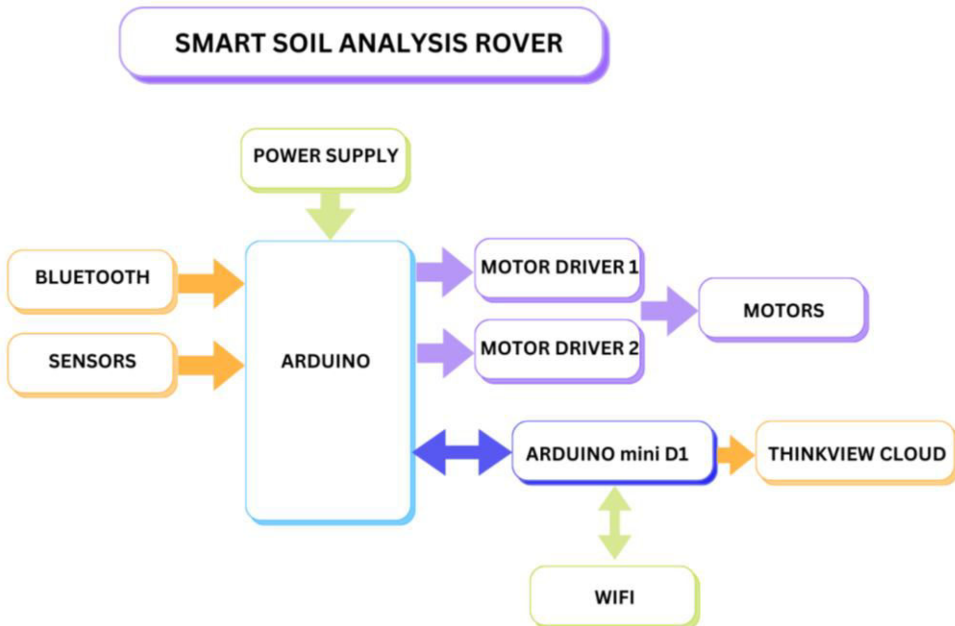


Figure 3.1 Block diagram of the Smart Soil Analysis Rover

The above block diagram represents the architecture of the **Smart Soil Analysis Rover**, which is designed to analyze soil parameters and transmit data wirelessly. The central controller, **Arduino UNO**, interfaces with various sensors and modules.

- **Sensors** like the Soil Moisture Sensor, Metal Detector, and DHT11 (for humidity and temperature) collect environmental and soil-related data.
- This data is processed by the Arduino UNO and then displayed on the **LCD** in real-time.
- Simultaneously, the processed data is transmitted via **Bluetooth Module (HC-05)** and **Wi-Fi Module (D1 Mini)** for local and cloud-based monitoring.

- **Motor Drivers** control the **DC Motors** that move the rover chassis across the terrain.
- A **Buzzer** is included to provide alert signals when specific conditions are detected (e.g., metal presence or sensor limits).

This setup enables both **manual control and remote monitoring**, supporting applications in precision agriculture, environmental research, and site analysis.

3.2 Components Description

3.2.1 Arduino UNO

The Arduino UNO is a widely-used open-source microcontroller board based on the ATmega328P. It features 14 digital input/output pins, 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, and a reset button. It is the brain of the rover and is responsible for interfacing with all the sensors, actuators, and communication modules. It executes the embedded C code uploaded via the Arduino IDE and controls the logic for data collection, transmission, and mobility.

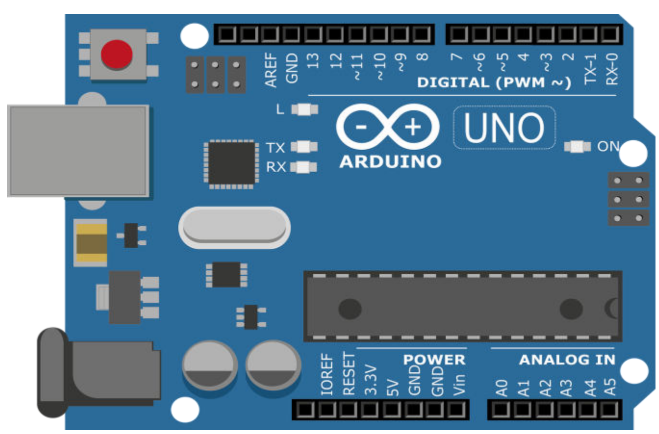


Figure 3.2 Arduino UNO

Key Features:

- Operating Voltage: 5V
- Input Voltage: 7-12V
- Digital I/O Pins: 14
- Analog Input Pins: 6
- Clock Speed: 16 MHz

3.2.2 D1 Mini (Wi-Fi Module)

The D1 Mini is a compact Wi-Fi-enabled microcontroller board based on the ESP8266 chipset. It enables wireless communication with cloud platforms like ThinkView, allowing real-time soil data to be uploaded and visualized remotely. Its small size and low power consumption make it ideal for portable embedded applications.

Figure 3.3 D1 Mini (Wi-Fi Module)

Key Features:

- Microcontroller: ESP8266
- Operating Voltage: 3.3V
- Wi-Fi: 802.11 b/g/n
- USB to Serial: Micro USB port
- Flash Memory: 4 MB

3.2.3 Bluetooth Module (HC-05)

The HC-05 is a serial Bluetooth module designed for wireless communication. In this project, it facilitates short-range communication between the rover and a smartphone or computer. It operates via the Serial (UART) communication protocol and can both send and receive data.

Key Features:

- Operating Voltage: 3.3V to 5V
- Communication: UART (baud rate 9600 default)
- Range: Up to 10 meters
- Role: Master/Slave switchable

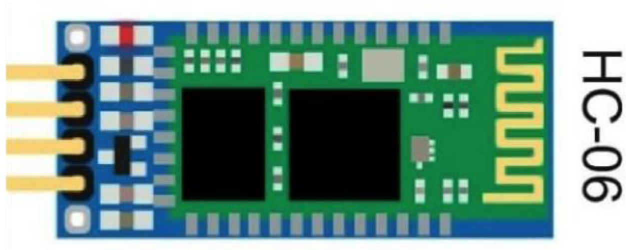


Figure 3.4 Bluetooth Module (HC-05)

3.2.4 Soil Moisture Sensor

This sensor detects the volumetric water content in the soil. It has two probes that act as variable resistors based on the moisture present. More water decreases resistance and increases conductivity. The analog output is read by the Arduino to determine moisture levels.

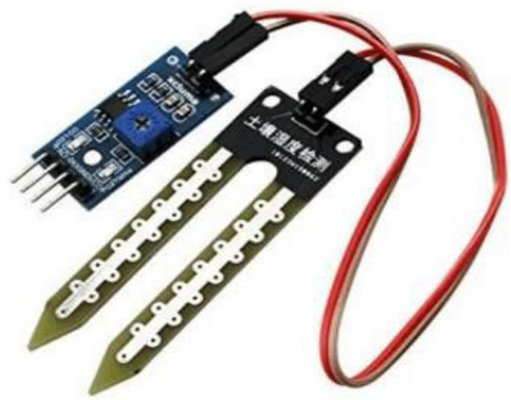


Figure 3.5 Soil Moisture Sensor

Key Features:

- Operating Voltage: 3.3V–5V
- Output: Analog (moisture level)
- Applications: Irrigation systems, soil monitoring

3.2.5 DHT11 Humidity and Temperature Sensor

The DHT11 is a basic digital sensor used for sensing temperature and relative humidity. It uses a thermistor and a capacitive humidity sensor to measure surrounding air conditions and outputs calibrated digital signals.

Figure 3.6 DHT11 Humidity Sensor

Key Features:

- Temperature Range: 0–50°C (±2°C accuracy)
- Humidity Range: 20–90% RH (±5% accuracy)
- Operating Voltage: 3V–5V
- Interface: Single-wire digital signal

3.2.6 Metal Detector

The metal detector in the rover is used to sense the presence of metallic objects below or within the soil surface. It works on the principle of electromagnetic induction. When a metallic object comes near the detector coil, the magnetic field is disturbed, and this change is detected and indicated.

Figure 3.7 Metal Detector Sensor

Key Features:

- Detection Principle: Electromagnetic Induction
- Output: Analog/Digital signal to Arduino
- Application: Construction, mining, archaeology

3.2.7 Motor Drivers

Motor drivers are circuits used to control the speed and direction of DC motors. The L298N motor driver module is likely used, which allows the Arduino to control two motors independently by amplifying the control signals.

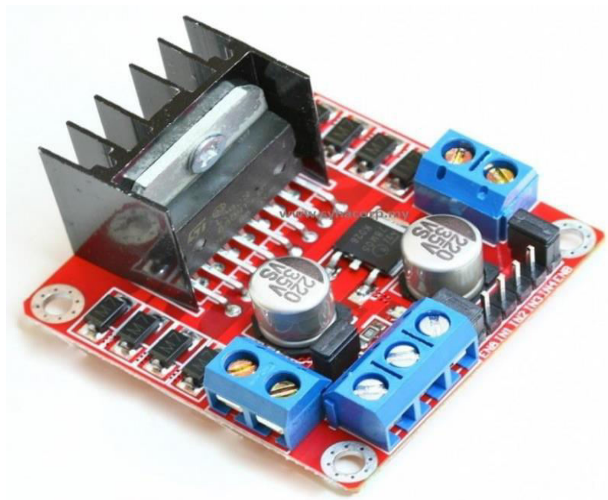


Figure 3.8 Motor Driver

Key Features:

- Operating Voltage: 5V–12V
- Output Current: 2A per channel
- Dual H-Bridge motor driver

3.2.8 Motors

DC gear motors are used to provide mobility to the rover. These motors are selected based on torque and speed requirements, ensuring the rover can navigate rough terrain. Each motor is connected to the motor driver for directional control.

Figure 3.9 Motor

Key Features:

- Voltage: 6V–12V
- Torque: Depends on gear ratio
- Type: Brushed DC geared motors
- Mounted on: Six-wheel chassis

3.2.9 LCD Display

The 16x2 LCD display is used to show real-time sensor values such as soil moisture, temperature, and humidity. It communicates with the Arduino using 4-bit or 8-bit parallel communication.

Figure 3.10 16x2 LCD Display

Key Features:

- Display: 2 lines × 16 characters
- Operating Voltage: 5V
- Interface: Parallel
- Backlight: LED

3.2.10 Buzzer

The buzzer is an audio signaling device that emits a tone when certain sensor thresholds are met, such as metal detection or extremely low moisture levels. It acts as a warning or status notifier.

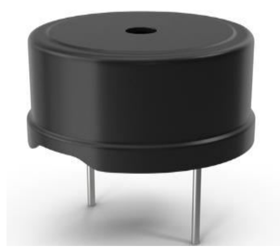


Figure 3.11 Buzzer

Key Features:

- Operating Voltage: 3V–5V
- Type: Piezoelectric or magnetic

CHAPTER 4

SOFTWARE TOOLS AND PLATFORM

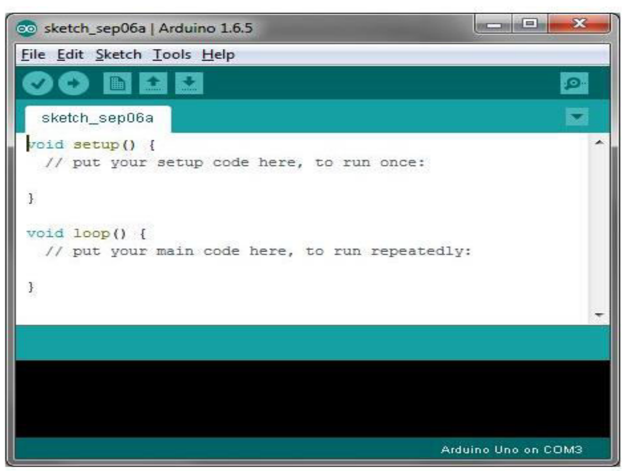
The software components of the **Smart Soil Analysis Rover** are essential for programming the microcontroller, simulating data communication, visualizing real-time results, and facilitating user interaction. The tools used in this project support embedded system development, Bluetooth communication, and cloud-based monitoring.

4.1 Arduino IDE

The Arduino Integrated Development Environment (IDE) is used to write, compile, and upload programs to the Arduino UNO microcontroller. It supports C/C++ programming and provides a simple interface to interact with connected hardware components.

Features:

- Syntax highlighting, serial monitor, and library support.
- Cross-platform compatibility (Windows, macOS, Linux).
- Open-source and lightweight.



Arduino IDE Default Window

Figure 4.1 Arduino IDE Default Window.

Use in Project:

- Writing embedded code to read sensor data.
- Controlling motors and actuators.
- Sending data via Bluetooth and Wi-Fi.

4.2 Arduino Bluetooth Controller App

This is an Android-based application used for wireless communication between a smartphone and the Arduino via the HC-05 Bluetooth module. It allows users to control rover movement and monitor sensor readings remotely.

Features:

- Button and joystick control.
- Custom command transmission.
- User-friendly interface.

Use in Project:

- Sending movement commands to the rover.
- Displaying sensor data transmitted from Arduino.

4.3 ThinkView (IoT Visualization Platform)

ThinkView is a cloud-based data visualization platform used to receive, analyze, and display real-time sensor values over the internet. It is compatible with the ESP8266-based D1 Mini module used for Wi-Fi communication.

Features:

- Real-time dashboard creation.
- Remote access to sensor data.
- Cloud data storage and graphical visualization.

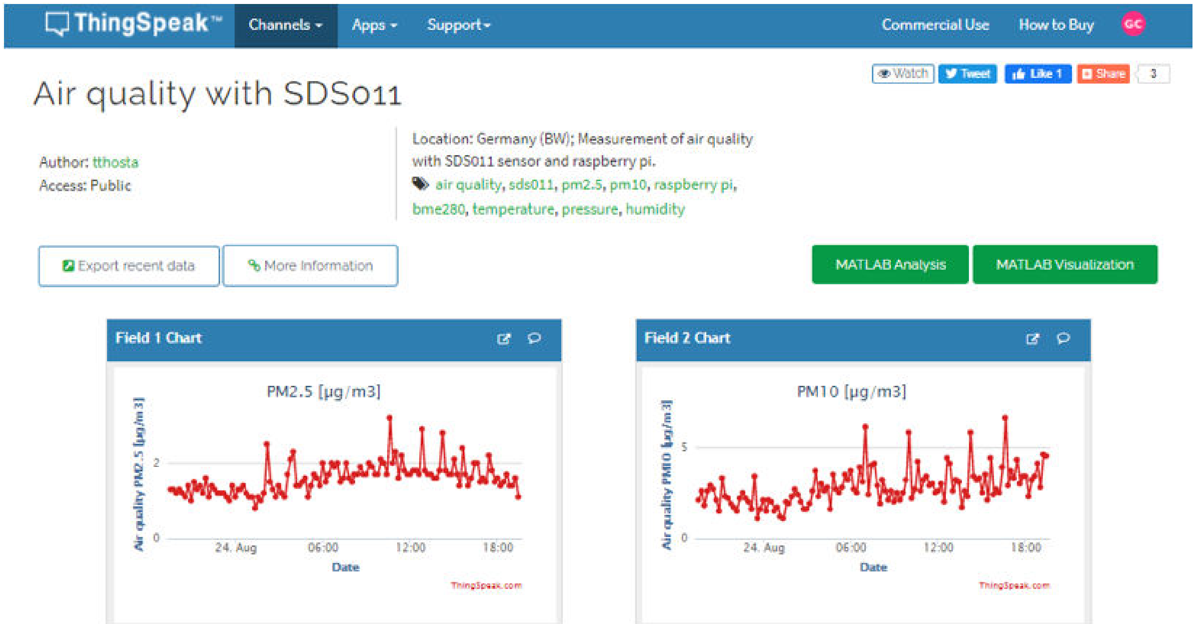


Figure 4.1 ThinkView Visualization Platform

Use in Project:

- Uploading sensor values via D1 Mini.
- Real-time environmental monitoring from any location.

4.4 Embedded C Programming

Embedded C is the programming language used to develop logic for the Arduino. It is a hardware-oriented language optimized for controlling embedded systems and allows efficient use of resources such as memory and processor cycles.

Features:

- Direct access to hardware ports and registers.
- Supports timer, interrupt, and PWM control.
- Fast execution and efficient code.

CHAPTER 5

FUNCTIONAL DESCRIPTION

5.1 SCHEMATIC DIAGRAM

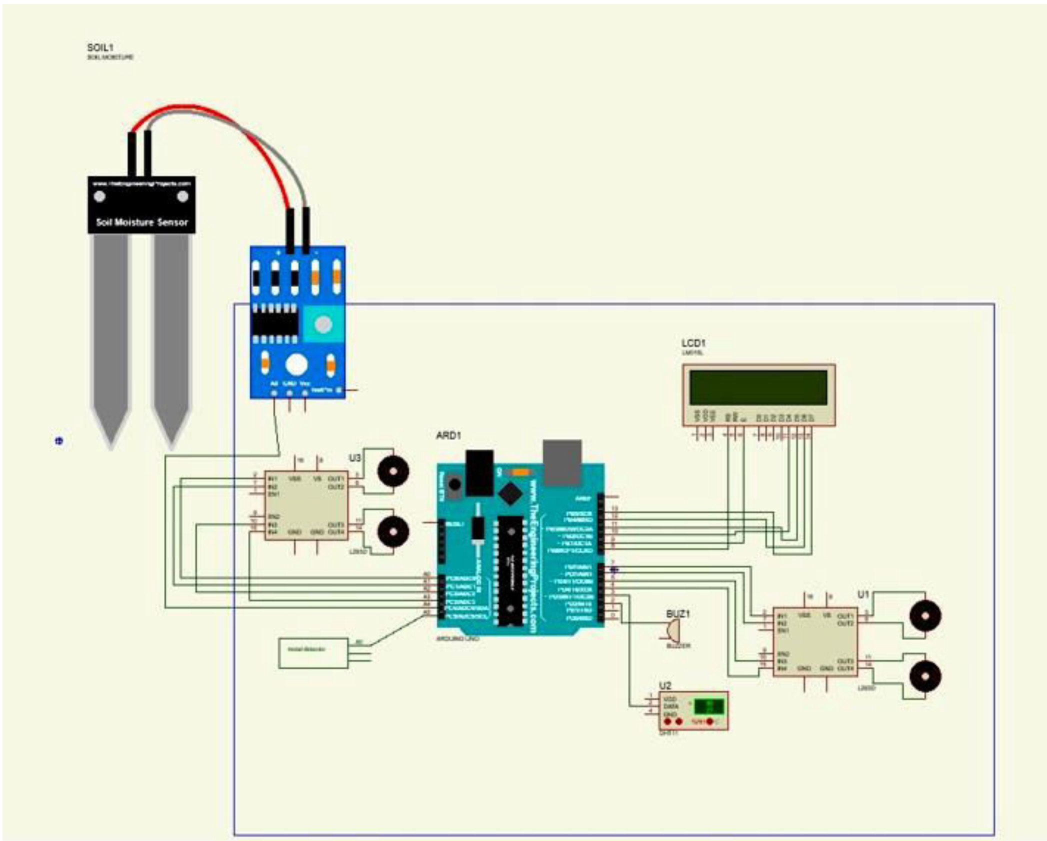


Figure 5.1 Circuit Diagram

5.2 WORKING

The Smart Soil Analysis Rover is a multi-functional six-wheeled robotic vehicle designed for autonomous soil analysis, environmental monitoring, object detection, and sample collection. It is built around an Arduino UNO microcontroller, which controls various sensors, actuators, and modules connected through a central circuit system.

The overall operation of the rover is executed in a looped sequence based on embedded Arduino logic. The Arduino continuously reads input data, processes commands, and coordinates the modules based on real-time feedback. The major components and their functions are as follows:

1. **Initialization**
   Upon powering up, the Arduino initializes all connected modules, including the soil sensor, DHT11, metal detector, LCD, motor drivers, and communication modules (Bluetooth and Wi-Fi). The system checks for module readiness before proceeding to active monitoring.
2. **Sensor Data Acquisition**
   The Soil Moisture Sensor, embedded beneath the rover, detects the moisture content in the soil and sends analog values to the Arduino.
   The DHT11 Sensor collects ambient temperature and humidity data from the surroundings.
   A Metal Detector Module scans the ground beneath the rover to detect hidden metallic objects.
   These readings are processed and displayed on the 16x2 LCD screen.
   Simultaneously, environmental readings from the DHT11 sensor are transmitted wirelessly to the ThinkSpeak IoT platform using the ESP8266 Wi-Fi module for remote data logging.
3. **Metal Detection Logic**
   The system checks for metal detection after sensor data is read.
   If metal is detected, the rover triggers a buzzer alarm and displays an alert on the LCD.
   If no metal is found, the rover continues to receive movement instructions from the user via the Bluetooth module (HC-05).
4. **Motor Control and Navigation**
   The rover features 6 DC motors, managed by two L298N motor driver modules, enabling full-direction control (forward, reverse, left, right, and stop).
   Commands received over Bluetooth are interpreted and executed to move the rover accordingly.

5. **Sample Collection Mechanism**

   A servo-operated jaw mechanism is mounted at the front. It is activated via mobile commands to pick up and release soil or small objects. This functionality allows collection from targeted or hazardous zones, expanding the rover's application in real-world field analysis.

6. **Relay Module Control**

   A relay module manages the switching of high-power components such as motors and actuators. This ensures electrical isolation, prevents circuit damage, and improves system reliability.

7. **Display and Monitoring**

   The LCD Display provides real-time feedback to the user on local conditions such as moisture level, temperature, humidity, and metal presence.

   Remote users can monitor data live via the ThinkSpeak platform, enabling decisions based on graphical and historical insights.

8. **Continuous Operation**

   This entire operation runs in a loop controlled by Arduino logic:

- Initialize system
- Read all sensor data
- Display data on LCD and send to cloud
- Check for metal detection
- If metal found → alert; if not → allow movement
- Accept navigation commands and drive motors
- Monitor for sample pickup conditions
- Repeat until stop command is issued

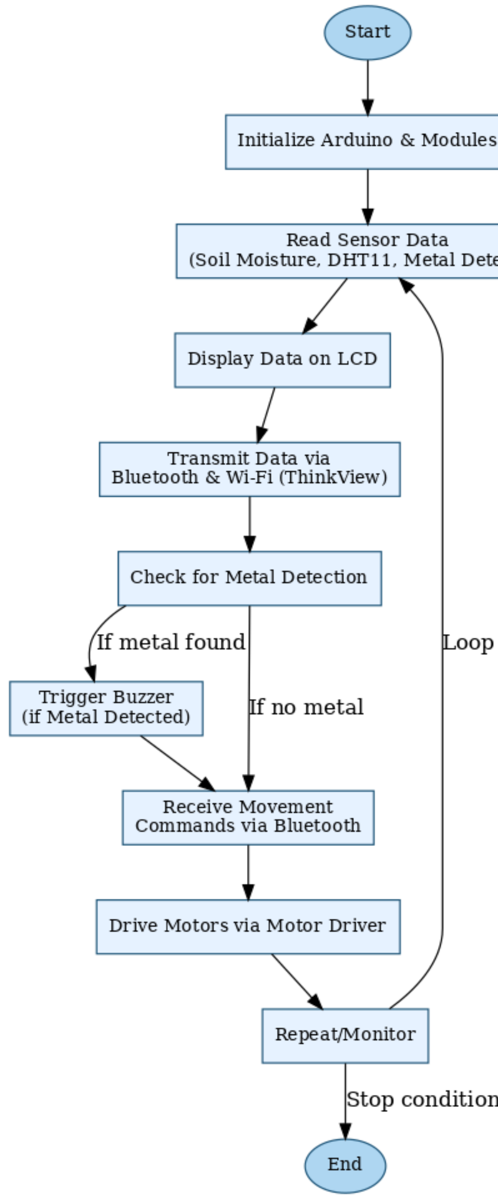


Figure 5.2 Flow Diagram

The rover is powered by a rechargeable battery pack, with carefully designed wiring to minimize power loss and electromagnetic interference. Its modular approach ensures flexibility, making it suitable for use in agriculture, mining, and environmental surveying.

CHAPTER 6

RESULTS

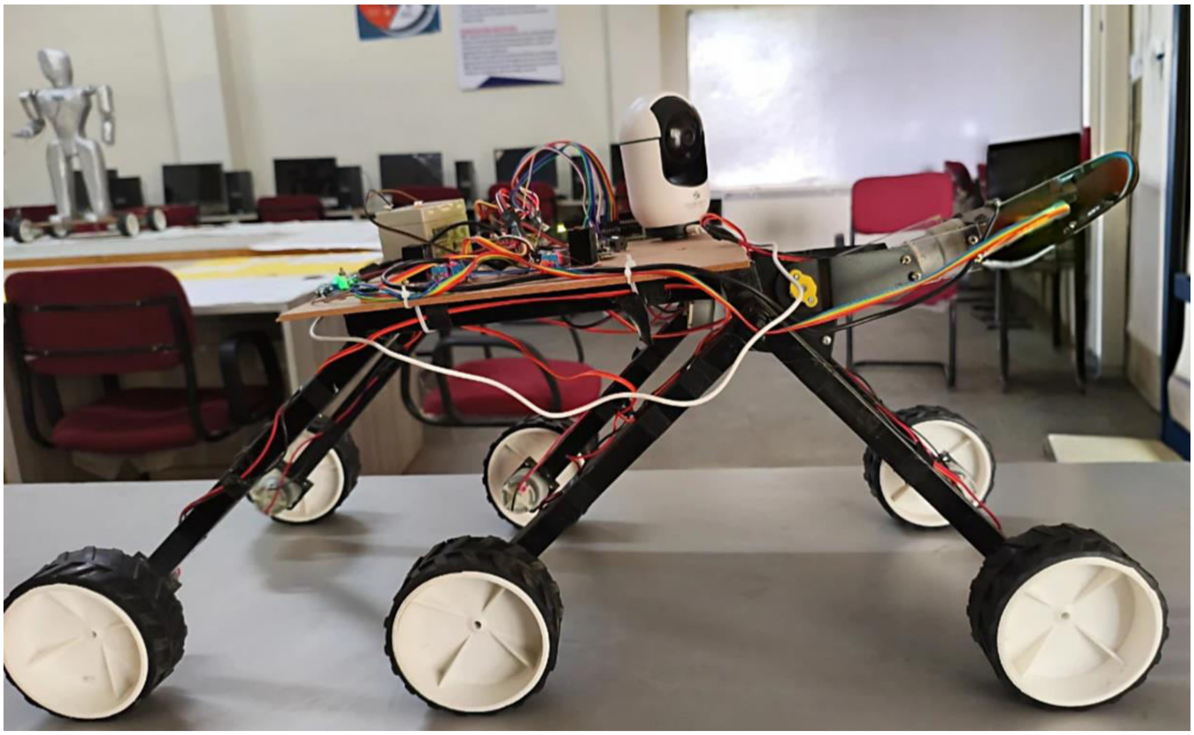


Figure 6.1 Smart Soil Analysis Rover

The Smart Soil Analysis Rover was successfully developed to perform real-time soil and environmental monitoring. It was tested in varied terrain conditions and showed effective operation. Key outcomes include:

- **Mobility and Terrain Handling**
  The six-wheeled design ensured stable navigation on uneven agricultural fields with smooth control during forward, reverse, and turning movements.
- **Sensor Performance and Data Access**
  The soil moisture sensor, DHT11 (temperature and humidity), and metal detector worked reliably. Data was displayed on the LCD and transmitted to the ThinkSpeak

IoT platform. Manual control and monitoring via Bluetooth through a mobile app was seamless.

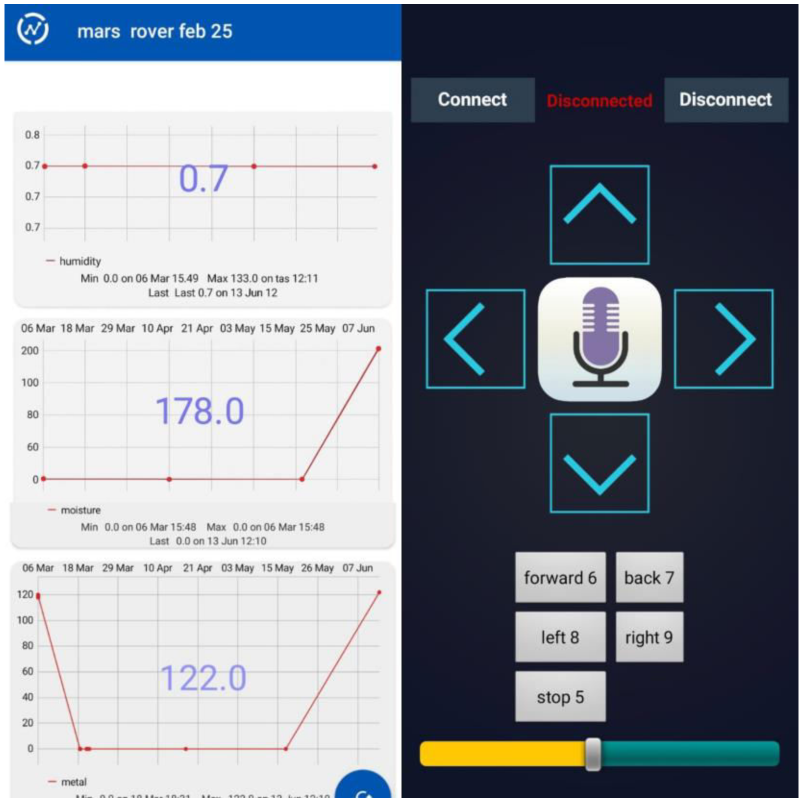


Figure 6.2 ThinkSpeak IoT and Bluetooth App Interface

- **Remote Visualization**
  Sensor values were logged and visualized remotely on ThinkSpeak. The mobile app supported real-time rover control and feedback.
- **Sample Collection**
  The servo-based gripper mechanism successfully picked up soil samples from uneven and constrained areas, aiding safe and remote sample handling.
- **Modular and Economical Build**
  Built using affordable and easily available components like Arduino UNO and ESP8266, the system remains low-cost, modular, and upgradable.

CHAPTER 7

ADVANTAGES & APPLICATIONS

7.1 ADVANTAGES

- Monitors soil moisture, humidity, temperature, and metal presence in real time
- Sensor data is accessible via Bluetooth and ThinkSpeak IoT cloud
- Gripper allows collection from hard-to-reach or hazardous areas
- Components are easily replaceable and extendable
- Uses affordable, widely available electronic components

7.2 APPLICATIONS

- **Smart Agriculture**
  Used for analyzing soil moisture and environmental conditions to assist in precision farming.
- **Disaster Zone Exploration**
  Deployed in areas affected by floods, landslides, or chemical spills to collect soil and air data without human risk.
- **Geological and Environmental Surveys**
  Helpful for soil testing, humidity monitoring, and metal detection in research and field surveys.
- **Educational and Research Tool**
  Acts as a prototype for robotics, IoT integration, and environmental sensing in academic institutions.
- **Military and Surveillance Use**
  Can be adapted for terrain mapping, obstacle detection, and object retrieval in reconnaissance missions.

CHAPTER 8

CONCLUSION & FUTURE SCOPE

CONCLUSION

The project **“Smart Soil Analysis Rover”** has been successfully developed and tested. It integrates multiple sensors and modules to monitor soil and environmental conditions in real time. The rover also demonstrates efficient wireless communication through Bluetooth and IoT-based data logging using ThinkSpeak.

The system performed reliably in collecting data, navigating terrain, and operating the sample collection jaw. It reflects a strong understanding of embedded systems, robotics, and IoT integration. All components were carefully selected and implemented to ensure smooth functionality, making this project a successful prototype for practical and educational use.

FUTURE SCOPE

- Add GPS and compass modules for autonomous navigation and location tracking
- Incorporate solar panels for sustainable and energy-efficient power management
- Implement advanced machine learning for object detection, terrain classification, and intelligent decision-making
- Integrate AI-based obstacle avoidance and path planning systems to reduce manual control
- Upgrade to long-range wireless communication (e.g., LoRa or satellite uplink simulation) for broader field usability
- Enable real-time collaboration between multiple rovers using a central cloud platform
- Expand support for additional scientific instruments like gas sensors, UV sensors, and spectrometers to enhance research capability

APPENDIX

SOURCE CODE

```
#include <LiquidCrystal.h>
#include <Servo.h>
const int rs = 8, en = 9, d4 = 10, d5 = 11, d6 = 12, d7 = 13;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
#include "DHT.h"
#define DHTPIN 3
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
int buz=2;
int m1=6;
int m2=7 ;
int m3=4;
int m4=5;
int m5=A0;
int m6=A1;
int m7=A2;
int m8=A3;
int mos=A4;
int ms=A5;
int x;
int cnt=0;
void setup() {
pinMode(m1,OUTPUT);
pinMode(m2,OUTPUT);
pinMode(m3,OUTPUT);
pinMode(m4,OUTPUT);
pinMode(m5,OUTPUT);
pinMode(m6,OUTPUT);
pinMode(m7,OUTPUT);
pinMode(m8,OUTPUT);
pinMode(mos,INPUT);
pinMode(ms,INPUT);
pinMode(buz,OUTPUT);
Serial.begin(9600);
digitalWrite(m1,0);
digitalWrite(m2,0);
digitalWrite(m3,0);
digitalWrite(m4,0);
digitalWrite(m5,0);
```

```
digitalWrite(m6,0);
digitalWrite(m7,0);
digitalWrite(m8,0);
  lcd.begin(16, 2);
  lcd.print("smart rover");
  delay(1500);
  lcd.clear();
  dht.begin();
}
void loop() {
     int t = dht.readTemperature();
    int h = dht.readHumidity();
    int mval=1023-analogRead(mos);
    int mtval=1023-analogRead(ms);
    if(mtval>500)
   {
    digitalWrite(buz,1);
   }
   else
   {
    digitalWrite(buz,0);
   }
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("T: "+String(t)+" H:"+String(h));
      lcd.setCursor(0, 1);
    lcd.print("MOS: "+String(mval)+" MT:"+String(mtval));
    delay(500);

    cnt++;
    if (cnt > 10) {
      Serial.print("2838060,P6Z07T35NVBAGTEV,0,0,project1,12345678,"
        + String(t) + "," + String(h) + ","
        + String(mval) + "," + String(mtval) + "\n");
      cnt = 0;
    }
 x=Serial.read();
 if(x==1)
{
 digitalWrite(m1,1);
 digitalWrite(m2,0);

}
  if(x==2)
```

```
{
 digitalWrite(m1,0);
 digitalWrite(m2,1);

}
  if(x==3)
{
 digitalWrite(m3,1);
 digitalWrite(m4,0);

}
 if(x==4)
{
 digitalWrite(m3,0);
 digitalWrite(m4,1);

}
 if(x==5)
{
 digitalWrite(m5,0);
 digitalWrite(m6,0);
 digitalWrite(m7,0);
 digitalWrite(m8,0);
  digitalWrite(m1,0);
 digitalWrite(m2,0);
 digitalWrite(m3,0);
 digitalWrite(m4,0);
} if(x==6)
{
 digitalWrite(m5,1);
 digitalWrite(m6,0);
 digitalWrite(m7,1);
 digitalWrite(m8,0);
}
 if(x==7)
{
 digitalWrite(m5,0);
 digitalWrite(m6,1);
 digitalWrite(m7,0);
 digitalWrite(m8,1);
} if(x==8)
{

 digitalWrite(m5,0);
```

```
 digitalWrite(m6,1);
 digitalWrite(m7,1);
 digitalWrite(m8,0);
} if(x==9)
{
 digitalWrite(m5,1);
 digitalWrite(m6,0);
 digitalWrite(m7,0);
 digitalWrite(m8,1);
}

}
```

REFERENCES

[1] **G. Deshpande, P. Kulkarni, and R. Patel**, “IoT-Based Low-Cost Soil Moisture and Soil Temperature Monitoring System,” *2022 International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN)*, Chennai, India, May 2022.
[Online]. Available: https://arxiv.org/abs/2206.07488

[2] **S. Sadowski and P. Spachos**, “Wireless Technologies for Smart Agricultural Monitoring Using Internet of Things Devices with Energy Harvesting Capabilities,” *IEEE Access*, vol. 8, pp. 52504–52520, 2020.
[Online]. Available: https://arxiv.org/abs/2005.02477

[3] **M. Campbell et al.**, “A Portable Agricultural Robot for Continuous Apparent Soil Electrical Conductivity Measurements,” *2021 IEEE International Conference on Robotics and Automation (ICRA)*, Xi'an, China, July 2021.
[Online]. Available: https://arxiv.org/abs/2107.09219

[4] **A. Milella, G. Reina, and M. Nielsen**, “A Multi-Sensor Robotic Platform for Ground Mapping and Estimation Beyond the Visible Spectrum,” *2021 IEEE Sensors Journal*, vol. 21, no. 12, pp. 13459–13468, April 2021.
[Online]. Available: https://arxiv.org/abs/2104.05259