1. Introduction

Agriculture, a cornerstone of human civilization, is at a crossroads where traditional practices must evolve to meet modern challenges. Factors such as labor shortages, resource inefficiencies, and climate variability necessitate the adoption of technology-driven solutions. IoT and robotics are transforming farming practices, enabling automation and data-driven decision-making.

This project proposes a smart Agribot that automates critical farming tasks, leveraging block-based programming for rapid development. The Agribot integrates sensors for monitoring environmental conditions and actuators for executing tasks like ploughing, sowing and watering. It’s controlled through a user-friendly mobile app, providing real-time control capabilities. This innovation aims to bridge the gap between small-scale farmers and advanced agricultural technologies.

Key Features:

Automation of ploughing and seed sowing.

Real-time soil health monitoring (Nitrogen, Potassium, Phosphorous)

Cloud integration for data storage and analysis

User-friendly control via mobile application (Bluetooth)

2. Literature Survey

1. "IoT Prototyping Using Block-based Programming: An use of smart Agriculture," 3rd International conference on Artificial Intelligence and Signal Processing (AISP), 2023 by Meenaxi M Raikar

   This paper proposes how block-based programming simplifies IoT application development. It demonstrates the potential of tools like MIT App Inventor in building user-friendly IoT prototypes for smart agriculture.

2. "Arduino-based Automated Smart Agriculture Robot," International Journal of Advanced Research in Electronics and Communication Engineering, 2021 by B. Sujatha et al.

   This paper focuses on the hardware assembly and control logic using Arduino for farming robot. Arduino-controlled system integrating sensors (e.g., soil moisture and ultrasonic) to automate irrigation and seeding processes. It enhances efficiency, reduces labour dependency, and optimizes resource usage for sustainable farming.

3. Problem Definition

The Challenges in Traditional farming practices are labor-intensive and prone to inefficiencies. Farmers face difficulties in maintaining consistent crop yields due to unpredictable environmental factors and resource mismanagement. Automation is crucial to overcoming these barriers, yet existing solutions are either too complex or too expensive for small-scale farmers.

The Solution:

The project aims to design and implement a multifunctional Agribot capable of:

- Automating ploughing, sowing, and environmental monitoring.

- Storing real-time data on the cloud for future analysis.

- Providing an intiative user interface via a mobile application for control and monitoring.

4. Objectives

1. To develop a low-cost, multifunctional Agribot for smart agriculture.
2. To simplify IoT prototype development using block-based programming.
3. To enable real-time monitoring and control of farming operations through a mobile application.

5. Methodology

The agriculture robot project is an advanced solution designed to automate key farming tasks such as soil testing, seed dispensing, nutrient addition, and ploughing. At the heart of the system is an Arduino microcontroller, which serves as the brain of the robot, controlling all its functions based on inputs from sensors and commands received via a Bluetooth-enabled mobile app. The robot uses NPK and moisture sensors to assess soil quality and nutrient content, with the data being transmitted to the app for real-time monitoring by the user.

The robot is powered by a robust power supply unit that ensures reliable operation of all components, including servo motors, DC motors, and sensors. Servo motors are employed for precise actions such as dispensing seeds and nutrients (Nitrogen, Phosphorus, Potassium) and dipping a soil-testing probe into the ground. The DC motors, controlled by a motor driver, handle the robot's movement, enabling it to navigate forward, backward, and turn left or right, as well as drive the ploughing mechanism.

The working of the robot begins with initialization, during which it establishes a Bluetooth connection with the user’s mobile device. Soil testing is conducted first, where the robot dips the soil probe and collects NPK and moisture data, which is displayed on the mobile app. Based on these readings, the user can command the robot to dispense seeds and nutrients in precise quantities. Movement commands allow the robot to navigate to the next area for processing. The ploughing mechanism can also be activated and deactivated as needed to prepare the soil for planting.

This project is designed for efficiency and automating repetitive farming tasks to reduce human effort while improving productivity. With its modular design, the robot can easily integrate additional functionalities such as irrigation systems or more advanced soil analysis tools. It is a scalable and user-friendly solution that leverages automation and IoT technology to address modern agricultural challenges.

The fig5.1 block diagram represents an automated agriculture robot designed for efficient farming tasks. At its core is an Arduino Uno microcontroller, which serves as the brain of the system, controlling all operations and ensuring seamless coordination between various components. The robot is powered by a solar panel and a battery, with the solar panel providing sustainable energy while the battery ensures consistent power supply. To navigate and perform its tasks, the robot uses four DC motors connected to a motor driver L298N, which translates the Arduino’s control signals into high-current outputs to drive the motors. These motors enable the robot to move forward, backward, or turn left and right, providing complete mobility.

The system includes several sensors for monitoring soil conditions. A moisture sensor measures the soil's water content, while an NPK sensor analyses the levels of nitrogen, phosphorus, and potassium, providing critical data on soil fertility. This information is processed by the Arduino and displayed on a mobile app created using MIT App Inventor. The Bluetooth module (HC-05) facilitates wireless communication between the robot and the app, allowing the user to send commands and monitor real-time data remotely.

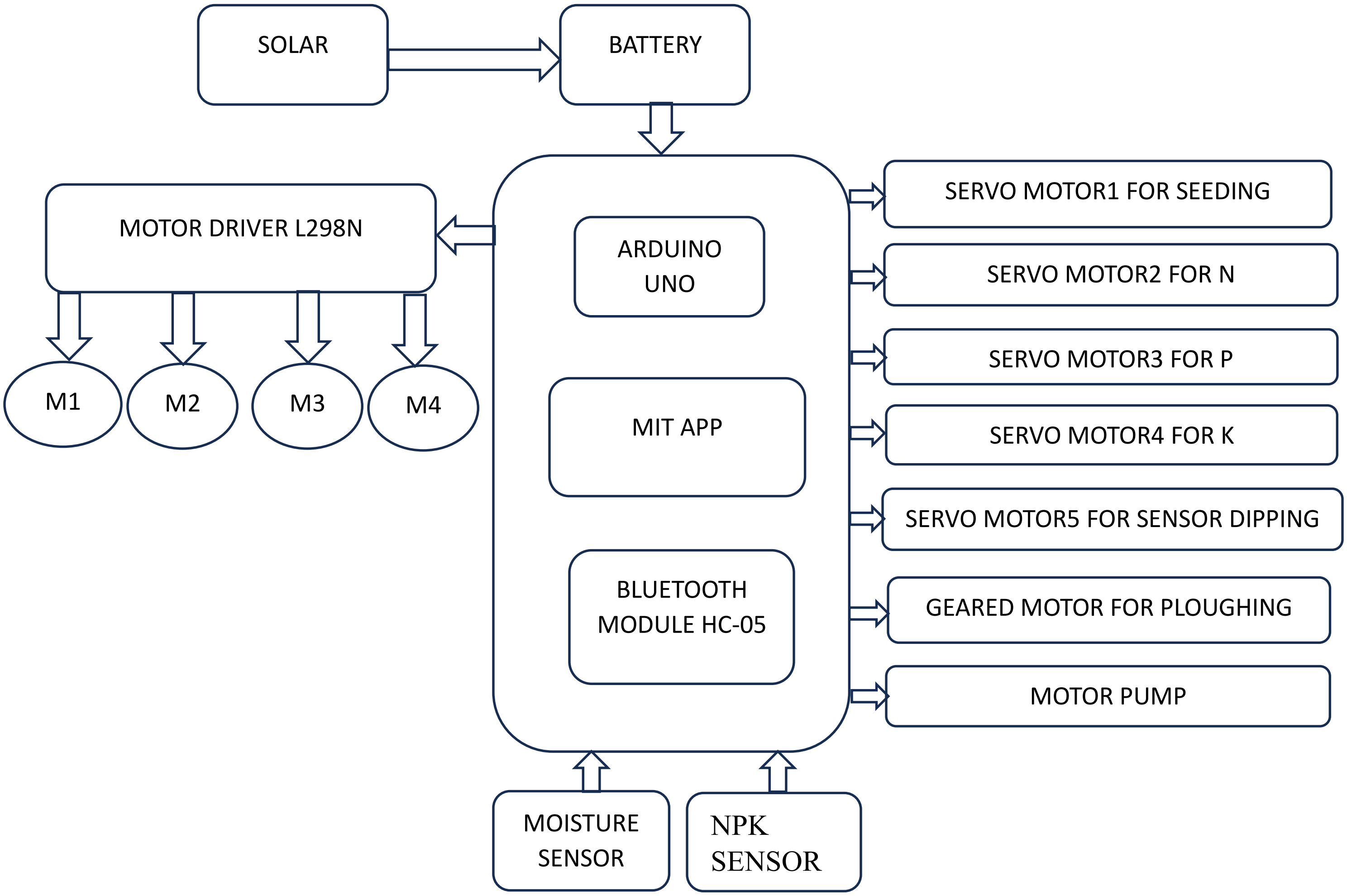


Fig 5.1: BLOCK DAIGRAM

The robot is equipped with servo motors for precise mechanical operations. One servo motor dispenses seeds into the soil, while three additional servo motors dispense nutrients (nitrogen, phosphorus, and potassium) based on the NPK sensor readings. Another servo motor handles the dipping mechanism for soil testing, lowering and raising the soil probe as needed. Additionally, a geared motor powers the ploughing mechanism to prepare the soil, while a motor pump irrigates the field when moisture levels are insufficient.

The entire system is controlled via a mobile app, which acts as a user-friendly interface. The app allows the user to send movement commands, activate ploughing, dispense seeds and nutrients, and control irrigation. This integration of renewable energy, sensor technology, and automated mechanisms makes the robot a versatile and efficient tool for modern agriculture, reducing labour and improving productivity.

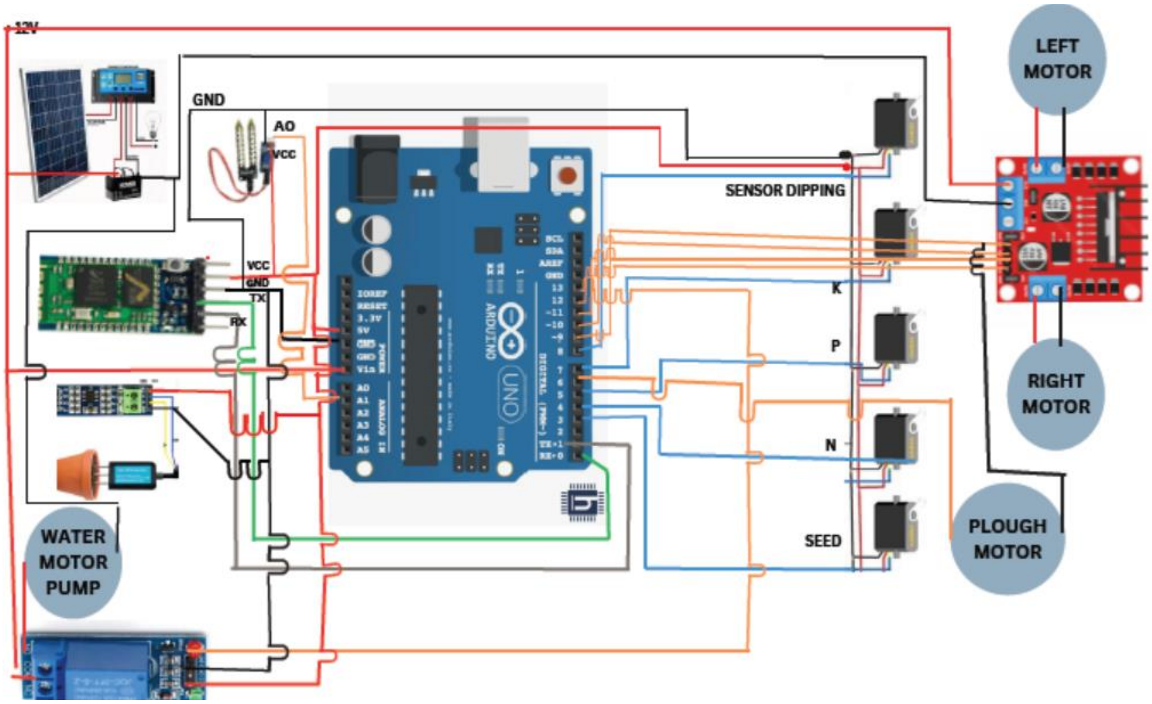


Fig 5.2: Circuit diagram

The Fig 5.2: Circuit diagram shows the integration of the Arduino Uno, sensors, motors, and other components. The Arduino Uno acts as the central processing unit, connected to various inputs and outputs. Power is supplied by a combination of a battery and solar panel, ensuring consistent operation. The motor driver L298N controls the movement of the robot through its left and right motors, as well as the water pump motor and plough motor. The connections for the motor driver include inputs from the Arduino's digital pins, which send control signals to enable forward, backward, and directional movements.

The servo motors, responsible for seeding, nutrient dispensing (N, P, K), and sensor dipping, are connected to specific PWM pins of the Arduino, allowing precise angular control. The sensors, including the moisture and NPK sensors, are connected to the analog pins of the Arduino for data acquisition. The Bluetooth module facilitates communication with the MIT App, enabling the user to send commands wirelessly and monitor system status. All components are grounded to ensure a common electrical reference.

6. Hardware and Software Design

Hardware Components

Here’s a detailed explanation of the components used in your project, suitable for a project report:

1. **Bluetooth Module**

Fig 6.1: Bluetooth module hc-05

- **Purpose:** Enables wireless communication between the robot and a controlling device (e.g., smartphone or computer).
- **Function:** Receives commands via Bluetooth for operating the robot's motors, pump, or sensors.

2. **Motor Driver L298N**

Fig 6.2: Motor Driver L298N

- **Purpose:** Drives the DC motors, allowing them to move forward or backward.
- **Function:** Acts as an interface between the Arduino and the motors, providing sufficient current and voltage to the motors.

3. Moisture Sensor

Fig 6.3: Moisture Sensor

- **Purpose:** Detects soil moisture levels to determine the need for irrigation.
- **Function:** Sends data to the Arduino, which uses it to trigger the water pump if soil moisture is low.

4. NPK Sensor

Fig 6.4: NPK Sensor

- **Purpose:** Measures soil nutrient levels (Nitrogen, Phosphorus, and Potassium).
- **Function:** Provides data to optimize fertilizer distribution.

5. Battery (12V)

Fig 6.5: lithium battery 12v

- **Purpose:** Powers the entire system, including motors, sensors, and the Arduino.
- **Function:** Supplies stable DC voltage to components, ensuring reliable operation.

6. **Johnson DC Motors (4)**

Fig 6.6: Johnson DC Motor

- **Purpose:** Drives the wheels to move the robot.
- **Function:** Converts electrical energy from the battery into mechanical energy for locomotion.

7. **Geared Motor**

Fig 6.7: Geared Motor

- **Purpose:** Provides precise and high-torque movement for specific mechanisms, such as seed or fertilizer dispensing.
- **Function:** Converts the rotational movement to the desired speed and torque.

8. Wheels (4)

Fig 6.8: Wheels

- **Purpose:** Provides mobility to the robot.
- **Function:** Facilitates smooth movement across agricultural terrain.

9. Arduino Uno

Fig 6.9: Arduino uno

- **Purpose:** Acts as the central control unit for the robot.
- **Function:** Processes sensor data, sends control signals to actuators, and manages communication with the Bluetooth module.

10. Relay

Fig:6.10: Relay

- **Purpose:** Controls high-current devices like the water pump.
- **Function:** Acts as an electronic switch controlled by the Arduino to turn the pump on or off.

11. Servo Motors

Fig:6.11: Servo Motor

- **Purpose:** Used for precise angular movement, such as positioning arms or dispensing systems.
- **Function:** Converts electrical signals from the Arduino into controlled rotational movement.

12. Breadboard and Jumpers

Fig 6.12: Breadboard and Jumpers

- **Purpose:** Facilitates prototyping and connections without soldering.

- **Function:** Connects components to the Arduino for testing and debugging.

13. Water Pump

Fig 6.13: Water pump

- **Purpose:** Pumps water for irrigation.
- **Function:** Operates based on commands from the Arduino when the moisture sensor detects low soil moisture.

14. Soldering and Glue Gun

Fig 6.14: Soldering and Glue Gun

- **Purpose:** Used for assembling and securing components.
- **Function:** Soldering ensures strong electrical connections; the glue gun provides mechanical stability.

15. Acrylic Seat

Fig 6.15: Acrylic seat

- **Purpose:** Acts as a sturdy base for mounting components.
- **Function:** Supports sensors, containers, and the pump securely.

16. Chassis (33 x 39 cm)

Fig 6.16: Chassis

- **Purpose:** Serves as the structural foundation of the robot.
- **Function:** Holds all components in place while ensuring stability and balance.

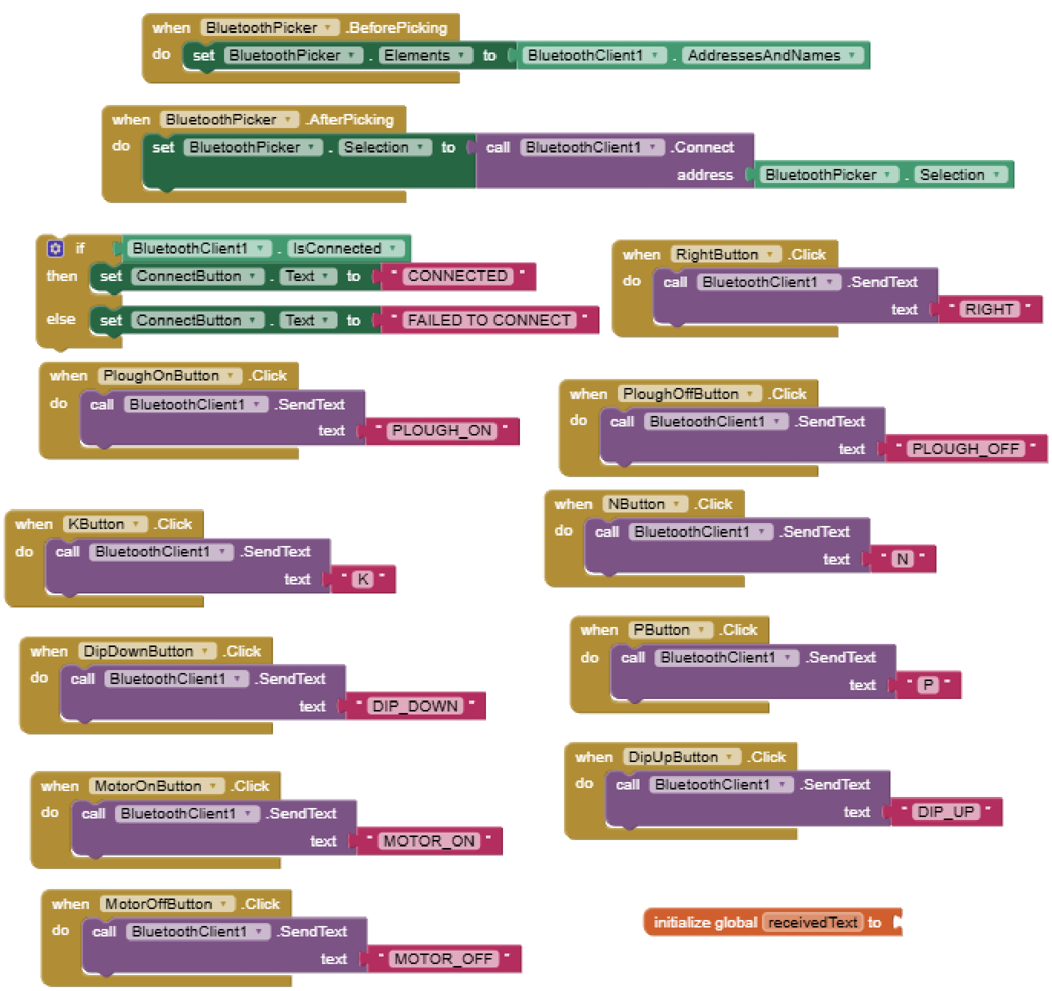
These components combine to create a multifunctional agricultural robot capable of irrigation, seeding, and fertilizing, while being controllable wirelessly via Bluetooth.

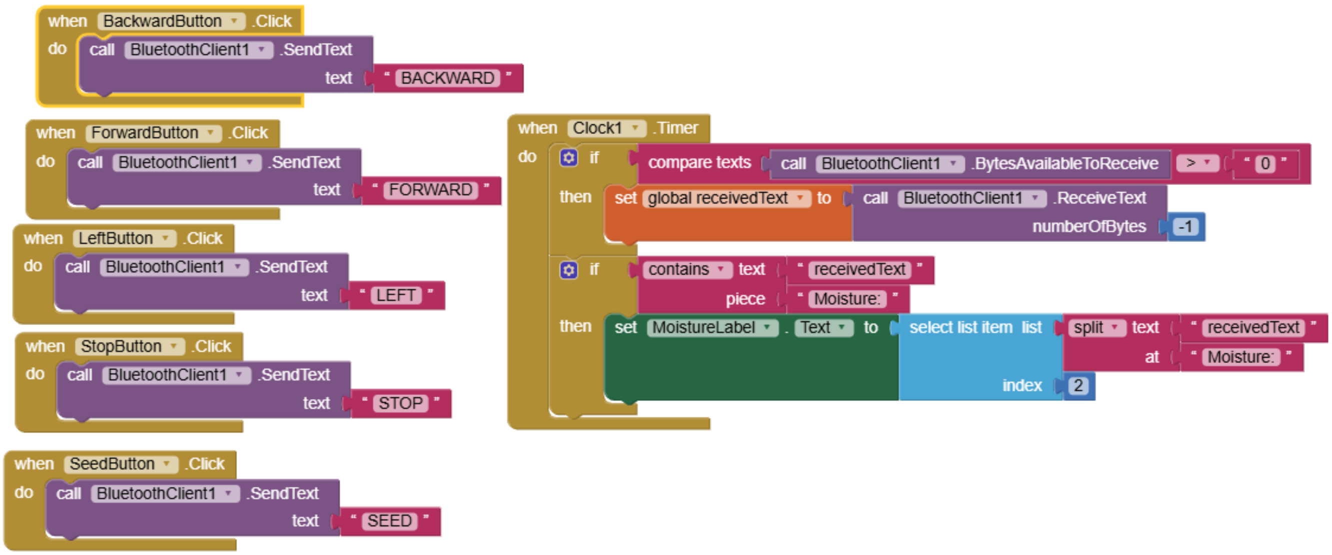
Software Tools

Here's an explanation of the **Software Tools** used in your project:

1. MIT App Inventor

- **Purpose:** To create a custom mobile application for controlling the robot wirelessly via Bluetooth.
- **Features Utilized:**
  - **User Interface Design:** Allows the creation of an intuitive interface for users to send commands (e.g., move forward, backward, turn, or start irrigation).
  - **Bluetooth Integration:** Provides blocks to connect to the Bluetooth module for real-time communication with the robot.
  - **Programming Interface:** Drag-and-drop block programming enables simple and visual implementation of logic. The block codes are:





- **Function in Project:**
    - Acts as the remote control for the robot.
    - Sends instructions to the Arduino through the Bluetooth module based on user input.
- **Advantages:**
    - No need for extensive programming knowledge.
    - Provides a platform-independent application that works on Android devices.

2. Arduino IDE

- **Purpose:** To write, debug, and upload code to the Arduino Uno microcontroller.
- **Features Utilized:**
    - **C++ Programming Environment:** Allows you to write and compile sketches (programs) for the robot. The program is :

```
#include <Servo.h>
// Servo motors
Servo seedServo, npkServos[3], dipServo;

// Pins for sensors and actuators
#define NPK_SENSOR_PIN A0
#define MOISTURE_SENSOR_PIN A1
#define PLOUGH_MOTOR_PIN 6
#define MOTOR_L1 9
#define MOTOR_L2 10
#define MOTOR_R1 11
#define MOTOR_R2 12
#define MAIN_MOTOR_PIN 13
 // New pin for main motor control
void setup() {
  Serial.begin(9600); // Bluetooth Communication
// Attach servo motors
  seedServo.attach(3);
  npkServos[0].attach(4);  // N
  npkServos[1].attach(5);  // P
  npkServos[2].attach(7);  // K
  dipServo.attach(8);      // Sensor dipping
 // Setup pins for DC motors
  pinMode(PLOUGH_MOTOR_PIN, OUTPUT);
  pinMode(MOTOR_L1, OUTPUT);
  pinMode(MOTOR_L2, OUTPUT);
  pinMode(MOTOR_R1, OUTPUT);
  pinMode(MOTOR_R2, OUTPUT);
```

```
  pinMode(MAIN_MOTOR_PIN, OUTPUT); // New motor control pin}
void loop() {
  // Read sensor data
  int npkValue = analogRead(NPK_SENSOR_PIN);
  int moistureValue = analogRead(MOISTURE_SENSOR_PIN);
// Send data to mobile app
  Serial.print("NPK: ");
  Serial.println(npkValue);
  Serial.print("Moisture: ");
  Serial.println(moistureValue);
// Check for commands
  if (Serial.available()) {
    String command = Serial.readString();
    if (command == "SEED") dispenseSeed();
    else if (command == "N") dispenseNPK(0);
    else if (command == "P") dispenseNPK(1);
    else if (command == "K") dispenseNPK(2);
    else if (command == "DIP_DOWN") dipSensors(true);
    else if (command == "DIP_UP") dipSensors(false);
    else if (command == "PLOUGH_ON") digitalWrite(PLOUGH_MOTOR_PIN, HIGH);
    else if (command == "PLOUGH_OFF") digitalWrite(PLOUGH_MOTOR_PIN, LOW);
    else if (command == "FORWARD") moveForward();
    else if (command == "BACKWARD") moveBackward();
    else if (command == "LEFT") turnLeft();
    else if (command == "RIGHT") turnRight();
    else if (command == "STOP") stopMovement();
    else if (command == "MOTOR_ON") digitalWrite(MAIN_MOTOR_PIN, HIGH);
 // New command
```

```
    else if (command == "MOTOR_OFF") digitalWrite(MAIN_MOTOR_PIN, LOW);
// New command
}
 delay(500);
}
void dispenseSeed() {
  seedServo.write(90); delay(1000); seedServo.write(0);
}
void dispenseNPK(int container) {
  npkServos[container].write(90); delay(1000); npkServos[container].write(0);
}
void dipSensors(bool down) {
 if (down) dipServo.write(90);
  else dipServo.write(0);
}
void moveForward() {
 digitalWrite(MOTOR_L1, HIGH); digitalWrite(MOTOR_L2, LOW);
  digitalWrite(MOTOR_R1, HIGH); digitalWrite(MOTOR_R2, LOW);
}
void moveBackward() {
  digitalWrite(MOTOR_L1, LOW); digitalWrite(MOTOR_L2, HIGH);
  digitalWrite(MOTOR_R1, LOW); digitalWrite(MOTOR_R2, HIGH);
}
void turnLeft() {
  digitalWrite(MOTOR_L1, LOW); digitalWrite(MOTOR_L2, HIGH);
  digitalWrite(MOTOR_R1, HIGH); digitalWrite(MOTOR_R2, LOW);
}
void turnRight() {
  digitalWrite(MOTOR_L1, HIGH); digitalWrite(MOTOR_L2, LOW);
```

```
  digitalWrite(MOTOR_R1, LOW); digitalWrite(MOTOR_R2, HIGH);
}
void stopMovement() {
  digitalWrite(MOTOR_L1, LOW); digitalWrite(MOTOR_L2, LOW);
  digitalWrite(MOTOR_R1, LOW); digitalWrite(MOTOR_R2, LOW);
    }
```

  - **Serial Monitor:** Used for testing and debugging by monitoring sensor data and system responses in real time.
  - **Libraries:** Supports various libraries (e.g., for Bluetooth, sensors, and motors) to simplify coding.
- **Function in Project:**
  - Acts as the development platform to program the Arduino for controlling sensors, actuators, and communication modules.
  - Enables the integration of sensor data (e.g., from the moisture or NPK sensor) and actuator commands (e.g., for motors, pump, or servo motors).
- **Advantages:**
  - Open-source and beginner-friendly.
  - Extensive online community support for troubleshooting and library usage.

How They Work Together:

- **MIT App Inventor:** Creates the mobile application to send commands via Bluetooth.
- **Arduino IDE:** Programs the microcontroller to interpret these commands and execute appropriate actions, such as moving the robot or starting the water pump.

7. Advantages, Disadvantages, and Applications

Advantages:

1. Cost-effective and scalable solution for small-scale farmers:

The robot's design and functionality are tailored to meet the needs of small-scale farmers without incurring high costs. Its modular and scalable nature allows farmers to start small and expand the system as their needs grow.

2. Real-time data acquisition enhances decision-making:

With built-in sensors and data collection capabilities, the robot provides farmers with real-time information about soil, crops, and environmental conditions. This data helps optimize farming practices and improves productivity by enabling precise decision-making.

3. Simplified development using block-based programming:

The use of block-based programming simplifies the process of customizing and operating the robot. Even individuals with minimal programming experience can design tasks and workflows, making it accessible to a wider audience.

Disadvantages:

1. Limited scalability for large-scale agricultural applications:

The robot is primarily designed for small-scale operations, which may limit its efficiency in handling larger agricultural areas or more complex farming needs.

2. Battery maintenance requirements:

The robot relies on battery power, which requires regular charging and maintenance. This dependency might be a challenge in regions with inconsistent power supplies or for users unfamiliar with battery upkeep.

Applications:

1. Small-scale precision farming:

The robot is ideal for tasks that require precise control over planting, fertilization, and pest management in small-scale farming operations.

2. Automated seeding and irrigation:

It can automate repetitive farming tasks such as planting seeds and delivering water to crops, saving time and reducing manual labour.

3. Environmental monitoring in greenhouses:

Equipped with sensors, the robot can monitor temperature, humidity, and soil conditions in greenhouses, helping maintain optimal growth environments for plants.

8. Results and Conclusion

The Agribot prototype successfully automates essential farming tasks. Testing in simulated environments demonstrated its effectiveness in reducing manual labour and improving resource efficiency. Real-time monitoring via a mobile app ensures farmers can make informed decisions, promoting sustainable practices.

9. Future Scope

1. Adding features for pest control and crop health monitoring.
2. Expanding cloud integration for predictive analytics.
3. Enhancing scalability for large-scale farming operations.

10. References

1. Meenaxi M Raikar et al., "IoT Prototyping Using Block-based Programming: A Use Case of Smart Agriculture," IEEE Conference, 2023.

2. MIT App Inventor, "Online IoT Development Platform," Accessed December 2022.

3. N. Perumalsamy et al., "Design of Faulty Switching Detection and Alert System," IEEE Access, 2022.

4. M. Ayaz et al., "Internet-of-Things (IoT)-Based Smart Agriculture," IEEE Access, 2019.

Photo

Fig: App interface

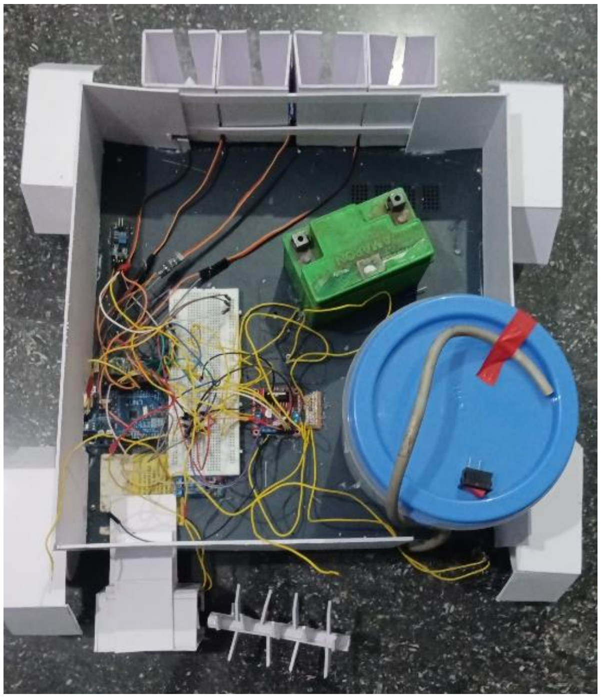


Fig: Prototyope

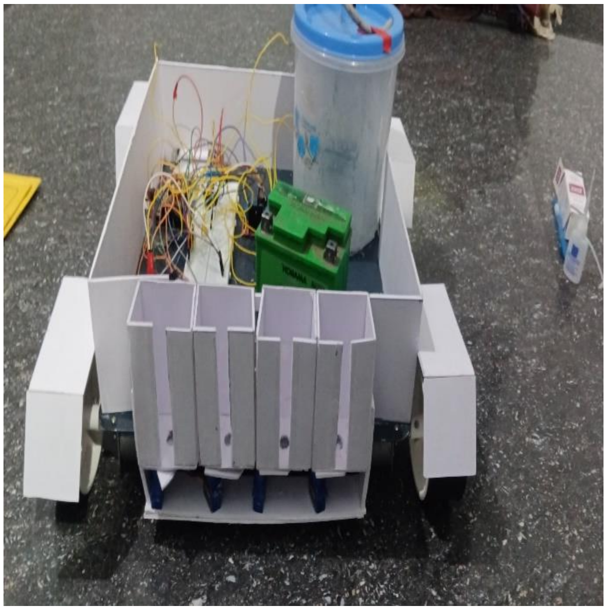


Fig: Prototype from back side

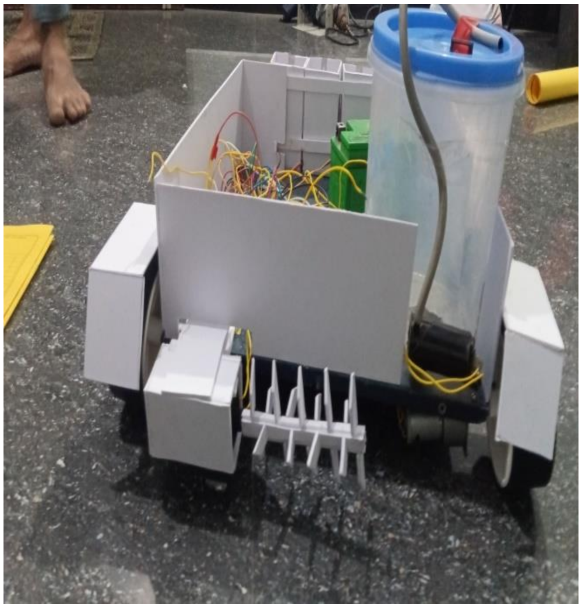


Fig: Prototype from front side

About Guide

Prof. SUJATA G. HUDDAR

Contact No: 9742066852

Received B.E degree in Electrical and Electronics Engineering from Visvesvaraya Technological University, HIT in 2011. Received M.Tech in Power System Engineering from AIT College of Engineering, Chickamagalur in 2014. Currently working as an Assistant Professor in the Electrical and Electronics Engineering Department at Hirasugar Institute of Technology, Nidasoshi, Karnataka. Has 4 years of teaching experience.

Project Associates

Mr. Suraj N Satwar

Address: Mullanki Road Satwar colony Examba

Pin No - 591244

Tal: Chikodi Dist: Belagavi

Contact No: 9380732154

Email: surajsatwar@gmail.com

Mr. Shivam R Metri

Address: K E B Road Desai Plot Raibag

Pin No - 591317

Tal: Raibag Dist: Belagavi

Contact No: 7204521378

Email: metrishivam14@gmail.com

Mr. Satyagouda K Patil

Address: Umarani

Pin No -591222

Tal: Chikodi Dist: Belagavi

Contact No: 6361137467

Email: patilsatish1303@gmail.com

Mr. Arun B Ghatage

Address: Chandur

Pin No - 591213

Tal: Chikodi Dist: Belagavi

Contact No: 7337641952

Email: arunghatage06@gmail.com