Development of Automated Sustainable Farm Management System

By

Ooi Jun Lin

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONOURS)

INFORMATION SYSTEMS ENGINEERING

Faculty of Information and Communication Technology

(Kampar Campus)

JUNE 2025

COPYRIGHT STATEMENT

© 2025 Ooi Jun Lin. All rights reserved.

This Final Year Project report is submitted in partial fulfillment of the requirements for the degree of Bachelor of Information Systems (Honours) Information Systems Engineering at Universiti Tunku Abdul Rahman (UTAR). This Final Year Project report represents the work of the author, except where due acknowledgment has been made in the text. No part of this Final Year Project report may be reproduced, stored, or transmitted in any form or by any means, whether electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the author or UTAR, in accordance with UTAR's Intellectual Property Policy.

Acknowledgement

I would like to express my sincere thanks and appreciation to my supervisor, Ms. Oh Zi Xin, who has given me the opportunity to work on this project titled *"Development of an Automated Sustainable Farm Management System."* A million thanks to her for her continuous guidance, valuable suggestions, motivation, encouragement, and support throughout the development of this project. It is worth mentioning that he consistently shared relevant materials and insights related to my project, which greatly helped me improve and stay on track.

Besides that, I would also like to thank my precious family and friends who have shown me unwavering support during this journey. Throughout the period of Final Year Project, I encountered many challenges, and I truly appreciate those who stood by me and supported me both physically and mentally. Their encouragement played a crucial role in helping me stay focused and complete this phase of the project.

ABSTRACT

This project presents the development of an automated sustainable farm management system aimed at improving agricultural efficiency through smart technology and eco-friendly practices. Farmers, especially small to medium-scale operators, often face challenges such as inefficient water usage, limited monitoring capabilities, and high dependency on manual operations. To address these issues, this project integrates Internet of Things (IoT) components including soil humidity sensors, automated water systems, and solar-powered operations to monitor and control key farming processes.

A mobile application is developed to enable real-time monitoring and remote control of watering schedules, lighting systems, and sensor data collection. Additionally, rainwater harvesting mechanisms and solar panels are incorporated to promote sustainability and reduce dependency on external resources. The system is developed using the Agile methodology, allowing for iterative testing and user feedback throughout the development cycle.

By automating repetitive tasks and promoting efficient resource utilization, the proposed system enhances farm productivity, reduces operational costs, and supports environmental sustainability. The outcome of this project provides a viable solution for modernizing traditional farming practices and contributes to the advancement of smart agriculture.

Area of Study: Internet of Things

Keywords: Data Collection in IoT, Monitoring Applications, Mobile Application, IoT, Farm Management System

Table of Contents

List of Figures

List of Tables

LIST OF ABBRIEVATION

IoT	Internet of Things
SDG	Sustainable Development Goals
FMS	Farm Management System
ERP	Enterprise Resource Planning
CRM	Customer Relationship Management
NGO	Non-Governmental Organization
KPI	Key Performance Indicator
LDR	Light Dependent Resistor
DHT11	Digital Humidity and Temperature Sensor *(model name)*
HC-SR04	Ultrasonic Sensor *(model name)*
TP4056	Lithium Battery Charger Module *(model name)*
MT3608	DC-DC Boost Converter Module *(model name)*
XL6009	DC-DC Boost Converter Module *(model name)*
FYP	Final Year Project
IDE	Integrated Development Environment
GPS	Global Positioning System
DC	Direct Current
LED	Light Emitting Diode

CHAPTER 1 Project Background

1.1 Introduction

In recent years, global agriculture has faced increasing challenges due to climate change, water scarcity, labor shortages, and the rising demand for food production. These challenges have emphasized the need for innovative approaches in managing farms more sustainably and efficiently. One promising approach is the integration of smart technologies, including the Internet of Things (IoT), automation systems, and renewable energy sources such as solar power. These technologies provide farmers with tools to optimize operations, reduce waste, and improve productivity [1], [2].

This project focuses on the development of an automated sustainable farm management system, designed specifically to support small and medium-sized agricultural operations. The proposed system integrates various technologies—soil humidity sensors, rainwater harvesting systems, automated irrigation and lighting modules, solar power generation, and mobile app control—to enable efficient and environmentally conscious farming practices [3].

Manual farming processes, particularly irrigation and monitoring, often consume unnecessary time and resources, leading to inefficiencies and excessive use of water and electricity. Furthermore, many farmers are not equipped with the tools to make data-driven decisions regarding soil conditions or optimal watering schedules. The system proposed in this project aims to address these issues by offering real-time monitoring, automated controls, and sustainability features.

This project leverages a mobile application to provide users with access to real-time environmental data and control mechanisms, ensuring farmers can monitor and manage their farms remotely and efficiently. It also supports long-term environmental goals by incorporating solar panels for energy supply and rainwater harvesting for irrigation. The entire system follows the Agile methodology to allow iterative development, user feedback, and continuous improvement.

The implementation of this system is anticipated to benefit farmers by saving time, reducing operational costs, promoting resource conservation, and encouraging the adoption of green technologies in agriculture. In the broader context, the system also aligns with the United Nations Sustainable Development Goals (SDGs), particularly SDG 2 (Zero Hunger) and SDG 13 (Climate Action), by fostering sustainable and smart farming practices [3],[34].

1.2 Problem Statement and Motivation

In the traditional farming sector, especially in rural and developing regions, there are several persistent issues that hinder productivity and sustainability:

- **Manual and inefficient irrigation practices** often lead to excessive water usage or under-irrigation, affecting crop health and increasing operational costs.
- **High dependency on non-renewable energy sources**, such as diesel-powered water pumps, contributes to environmental pollution and long-term costs.
- **Lack of real-time data and monitoring systems** prevents timely interventions, resulting in lower yields and wasted resources.
- **Labor-intensive processes** increase the workload on farmers and limit the scalability of their operations.

These issues not only reduce farm efficiency but also pose threats to environmental sustainability. With rising global concerns about food security and ecological preservation, it is imperative to introduce innovative technologies into the agricultural sector [3].

The motivation behind this project stems from both personal interest in sustainable technology and a broader goal of improving farming conditions through automation and renewable energy. By reducing the dependence on manual labor and unsustainable practices, farmers can adopt more efficient, cost-effective, and eco-friendly operations. Additionally, integrating user-friendly mobile applications makes technology more accessible, especially for those with limited technical backgrounds [1]. Addressing these challenges will not only increase productivity and efficiency but also support environmental sustainability and economic resilience for local farming communities. Addressing these challenges will not only increase productivity and efficiency but also support environmental sustainability and economic resilience for local farming communities.

1.3 Project Scope

This project is centered on the design and development of an **automated sustainable farm management system** that integrates various hardware and software components. The scope includes:

1. **Mobile Application Development**:
   - Monitor real-time soil moisture levels.
   - Remotely control irrigation and lighting systems.
   - Display farm data and analytics in a user-friendly interface.
2. **Hardware Automation**:
   - Soil humidity sensors for environmental monitoring.
   - Automated irrigation and lighting modules controlled by a microcontroller.
   - Water tank and piping system for efficient watering.
3. **Sustainability Features**:
   - Integration of a **solar-powered system** to supply energy to farm operations.
   - Implementation of a **rainwater harvesting mechanism** to reduce reliance on external water sources.
4. **System Integration**:
   - Seamless communication between sensors, microcontrollers, and mobile applications using IoT technologies [2].
   - Real-time data logging and feedback loop to enable automated decisions.

The prototype developed in this project will demonstrate these features on a small scale, simulating a real-world environment suitable for smallholders or household farms.

1.4 Objectives

The key objectives of the project are:

1. To design and develop a **mobile application** that enables farmers to monitor and control farming operations in real time.
2. To implement **automated irrigation and lighting systems** that are triggered based on soil humidity readings.
3. To integrate **renewable energy sources** such as solar power into the system to reduce dependency on traditional electricity.
4. To utilize **rainwater harvesting** as a sustainable method for water supply in irrigation.
5. To promote data-driven farming by **providing real-time monitoring and data visualization** through the mobile application interface.

These objectives are designed to directly address the issues outlined in the problem statement and serve as a guide throughout the development and testing phases.

1.5 Impact, Significance, and Contribution

The proposed system is expected to contribute significantly in several aspects:

- **Environmental Sustainability**: By leveraging solar energy and rainwater harvesting, the system helps reduce the carbon footprint and conserve water resources. This aligns with global sustainability initiatives such as the UN SDGs [3].
- **Increased Productivity**: Real-time monitoring and automated actions reduce delays and improve precision in irrigation and farm maintenance.
- **Cost Efficiency**: Farmers can reduce labor costs, electricity bills, and water consumption through automation.
- **Accessibility and Usability**: The system is designed with user-friendliness in mind, making advanced agricultural technology accessible to less tech-savvy users.
- **Academic Value**: The project serves as a practical demonstration of how IoT, mobile development, and renewable energy can be integrated to solve real-world problems [1], [2].

This system not only supports farmers but also contributes to broader research on smart farming and sustainable development. It offers a blueprint for future agricultural innovations in similar contexts.

1.6 Background Information

Smart farming, also known as precision agriculture, is a modern approach that uses technology to optimize agricultural processes. It often involves the use of sensors, mobile devices, automation, and data analytics to monitor and control conditions such as soil moisture, temperature, and crop health. With the advancement of IoT and wireless communication, farmers can now remotely access farm data and control equipment using smartphones or web platforms [2].

Rainwater harvesting is a practice that captures and stores rainwater for agricultural use. Combined with smart irrigation systems, it allows for efficient use of natural water resources, especially in areas prone to drought or water scarcity [3]. Solar energy systems provide a clean and renewable source of power for farms. When integrated with IoT-based control systems, they enable farms to operate independently from the electricity grid, reducing energy costs and environmental impact [1].

The convergence of these technologies creates an opportunity to revolutionize traditional farming. This project aims to take advantage of these trends by developing a system that brings together sustainability, efficiency, and ease of use in one integrated platform.

CHAPTER 2: Literature Reviews

2.1 Introduction and Technology Review

2.1.1 Introduction

Smart agriculture has emerged as a critical innovation in the global effort to improve food production efficiency while addressing environmental and labor-related challenges. The adoption of digital tools and automation in agriculture has enabled farmers to collect and analyze field data, monitor crops, predict yields, and manage resources more efficiently. Various commercial farm management systems (FMS) are available today that serve these purposes. This chapter reviews three popular smart farming systems: Agrivi, Granular, and Trimble Ag Software. Each system is evaluated in terms of its features, strengths, and limitations. This evaluation helps identify the technological gaps that the proposed Automated Sustainable Farm Management System seeks to address, particularly in terms of automation, affordability, and sustainability for small to medium-scale farms.

2.1.2 Arduino Uno R4 WiFi

The Arduino UNO R4 WiFi is an upgraded version of the widely used Arduino UNO series. It is equipped with a Renesas RA4M1 microcontroller running at 48 MHz and provides significant improvements in terms of memory, connectivity, and peripheral support compared to its predecessor [27]. One of the key features of the UNO R4 WiFi is its built-in ESP32-S3 module, which enables WiFi and Bluetooth connectivity. This makes the board suitable for Internet of Things (IoT) applications where wireless data communication is required [28].

The UNO R4 WiFi also retains compatibility with existing Arduino shields and libraries, which ensures smooth integration with previous Arduino-based projects. Its enhanced processing capability and additional GPIO support allow the system to handle multiple sensors and actuators simultaneously, making it an ideal choice for developing an automated farm management system. Furthermore, the availability of Arduino IDE support and extensive community resources enhances its usability for both beginners and advanced developers [27], [28].

2.1.3 Python Programming Language

Python is a high-level, interpreted programming language that emphasizes code readability and simplicity [29],[35]. It has become one of the most popular languages worldwide, widely used in fields such as web development, data science, artificial intelligence, and IoT. Python's straightforward syntax and extensive library support make it suitable for rapid application development and integration with IoT systems [30].

In the context of this project, Python can be used for tasks such as backend data processing, sensor data visualization, and integration with cloud services. Its versatility allows developers to create applications that can analyze and display collected farm data in real-time. Furthermore, Python's large ecosystem of third-party libraries, such as NumPy, Pandas, and Matplotlib, provides powerful tools for numerical computation and data visualization [31]. These features make Python an excellent complement to Arduino-based systems, where data can be transmitted from microcontrollers to Python-based applications for further processing.

2.1.4 Blynk IoT Platform

Blynk is a popular IoT platform that enables remote monitoring and control of hardware devices through mobile and web applications. It provides developers with an easy-to-use environment for connecting microcontrollers, such as Arduino and ESP32, to the cloud [32]. With Blynk, users can create custom dashboards that visualize sensor readings, send alerts, and control actuators using virtual pins.

One of the main advantages of Blynk is its seamless integration with WiFi-enabled boards like the Arduino UNO R4 WiFi. The platform handles device authentication, cloud synchronization, and mobile app interfaces, which reduces development complexity and accelerates prototyping [33]. For this project, Blynk is used to display real-time farm data, such as soil moisture, temperature, humidity, and water tank level, while also allowing users to manually control actuators such as the water pump and lighting system. Its cross-platform accessibility through mobile apps makes it a practical choice for smart farming solutions.

2.2 Review of the Existing Applications

2.2.1 Agrivi

Figures 2.2.1 Agrivi's Logo[17]

Agrivi, established in 2013, is a leading global ag-tech company with a mission to revolutionize food production and create a positive impact on over a billion lives. The platform provides knowledge-driven farm management solutions that cater to a wide range of agricultural stakeholders, including farmers of all scales, cooperatives, food sourcing companies, financial institutions, non-governmental organizations (NGOs), and government agencies [4]. Agrivi is designed to support sustainable and resource-efficient agricultural practices, and it is currently used by thousands of users across more than 150 countries to enhance crop production [5].

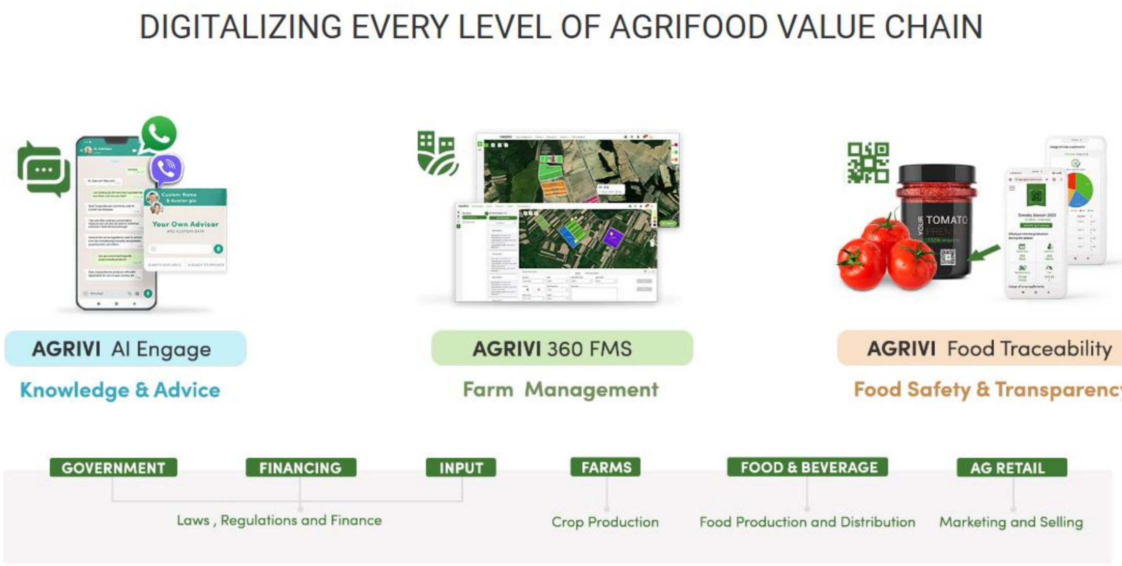
As digital agriculture continues to evolve, farm management systems (FMS) such as Agrivi are becoming increasingly essential. These systems can analyze and forecast optimal conditions for various farming activities, including planting, irrigation, harvesting, and pest management. According to Butu et al. [7], digital technologies like Agrivi allow farmers to operate with greater precision, reduce costs, and promote long-term sustainability. Agrivi helps optimize inputs like water, fertilizers, and pesticides, thus minimizing waste and environmental impact while enhancing productivity.

Agrivi offers a web-based platform complemented by a mobile application, enabling users to manage their farms from any location. While new users are encouraged to perform their initial farm setup through the web portal for complete data access, the mobile app allows quick field-level updates and insights, making it convenient for on-site operations [4]. This cross-platform accessibility enhances user engagement and efficiency, especially in dynamic farm environments.

The platform is divided into five key product categories, each tailored to a specific user group. The Farm Management module targets small and medium-scale farms and provides features such as crop lifecycle tracking, financial and sales monitoring, pest alarms, and reporting tools. Wine Management is designed for professional winemakers, offering functionalities to manage vineyard operations, monitor production traceability, and assess pest risks. For large-scale commercial operations, Enterprise Farm Management offers a customizable suite that integrates with external systems such as enterprise resource planning (ERP), customer relationship management (CRM), and farm machinery and sensors. Cooperative Management enables agricultural cooperatives to manage distributed farms under a unified system. Additionally, Traceability Reports are available across all product lines, providing QR code-based traceability that enhances transparency in the supply chain [5], [6].

Agrivi operates on a subscription model, beginning with a 30-day free trial. After the trial, users can select from monthly or annual plans, with pricing depending on the product tier—Standard, Professional, or Premium. Farm Management subscriptions start at €12.50 per month, while pricing for Wine Management and enterprise solutions is available upon request [4]. This flexible pricing strategy allows users to explore features before committing financially, though cost remains a concern for smaller farms.

Key features of Agrivi include comprehensive crop management tools that allow users to document activities such as planting, fertilization, and harvesting; financial tracking modules for recording input costs, revenues, and overall profitability; and inventory management, which helps in monitoring supplies like seeds, fertilizers, and pesticides. The platform also integrates real-time weather tracking, which delivers localized forecasts to help farmers make better operational decisions [6]. Additionally, the pest and disease management feature offers a knowledge base for diagnosing potential threats and applying recommended treatments. Agrivi also includes reporting and analytics tools to provide performance insights and help with regulatory compliance and operational transparency [6], [7].



Figures 2.2.2 Agrivi's System Features[18]

Agrivi's strengths lie in its centralized, cross-platform ecosystem, intuitive user interface, and knowledge-based decision support system. It promotes sustainable farming practices by optimizing resource use, reducing input waste, and offering environmental data analytics [8]. Users can generate simple yet comprehensive key performance indicators (KPIs) with a single click, providing quick insights into farm productivity and financial health [8].

However, despite its numerous strengths, Agrivi also has limitations. Its subscription-based pricing model may be a deterrent for smallholder farmers or those in developing regions with limited financial resources [9], [11]. Full functionality also requires a stable internet connection, which may not be consistently available in rural or remote agricultural zones. Furthermore, user feedback suggests that Agrivi has a steep learning curve, especially for users unfamiliar with digital platforms [10]. Additionally, reviews such as those from Software Connect highlight that while Agrivi's financial management module is useful for basic accounting, it lacks advanced bookkeeping features required by some users [9].

In conclusion, Agrivi provides a robust digital farm management platform with extensive features that support productivity, sustainability, and traceability. However, its reliance on internet connectivity, pricing structure, and lack of real-time automation through sensor integration make it less ideal for small-scale, resource-constrained farms. Nonetheless, its design philosophy and sustainability-focused tools serve as a valuable reference for the development of the proposed **Automated Sustainable Farm Management System**, particularly in integrating modules for crop monitoring, data analytics, and smart decision-making tools.

Figures 2.2.3 Agrivi's Mobile application[19]

2.2.2 Granular

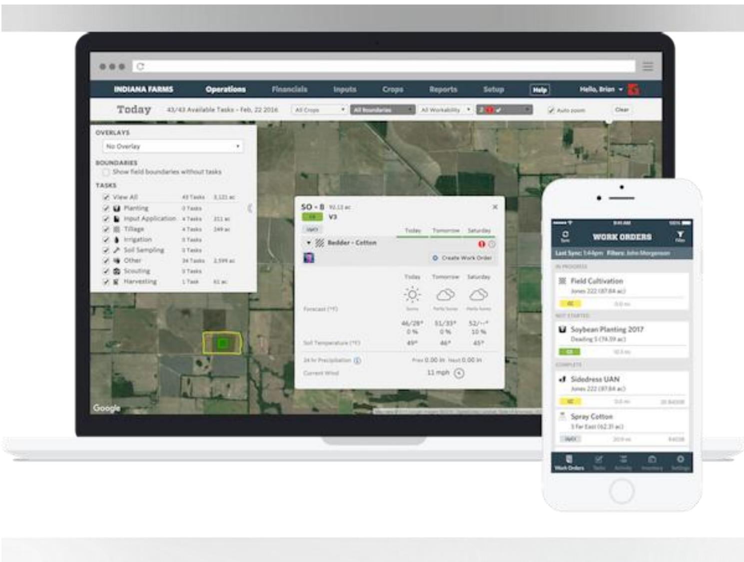
Figures 2.2.4 Granular's Logo[21]

Granular is a prominent farm management software that integrates business intelligence and operational efficiency for large-scale farms. Developed in the United States, Granular provides farmers with tools to analyze field-level profitability, manage labor and input costs, and make data-driven decisions. The system offers significant insights into economic aspects of farm management by breaking down profit margins by field, crop type, and input usage [12]. This allows farmers to optimize resource allocation and increase farm profitability.

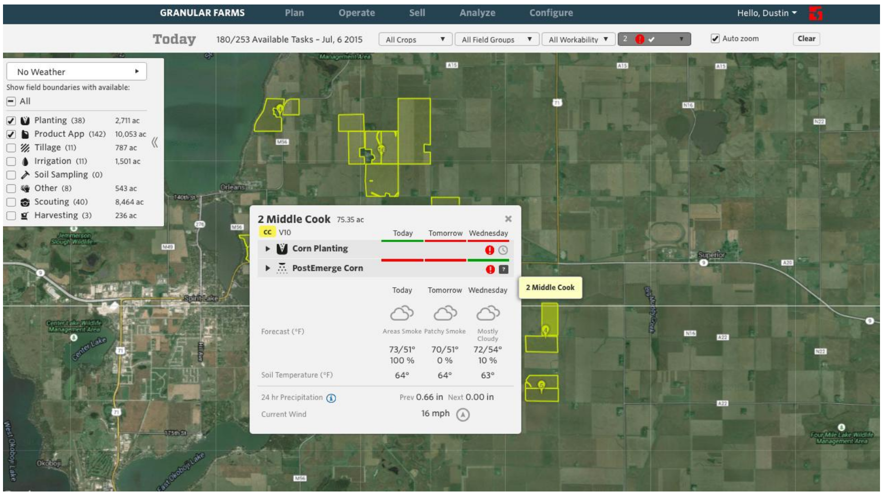
Key Features of Granular

Granular is recognized for its comprehensive set of features that cater to large-scale agricultural operations. The platform includes Field Mapping and Crop Planning tools, which enable users to create and refine operational plans based on real-time data [12]. These tools facilitate resource allocation and improve the efficiency of farming practices. Additionally, Real-Time Monitoring capabilities provide predictive models and satellite imagery that forecast yields and offer insight into crop health, helping farmers manage potential issues before they escalate [13].

Furthermore, Granular's Inventory and Financial Management tools allow for the tracking of farm inputs such as seeds, fertilizers, and pesticides, alongside labor costs and other operational expenses. These features help farmers ensure financial transparency and streamline budgeting processes, which are crucial for maintaining operational efficiency [13]. Mobile Access is also a standout feature, enabling farmers to manage tasks and schedules from remote locations [13]. Additionally, tools for Irrigation and Soil Analysis support sustainable farming practices by tracking water usage and soil health, although these features are more focused on data analysis rather than real-time environmental control [13].



Figures 2.2.5 Granular's Computer and Mobile Apps[22]



Figures 2.2.6 Granular's Field Mapping features[23]

Strengths of Granular

One of the core strengths of Granular is its ability to provide data-driven insights that integrate both agronomic and financial data. This integration allows farmers to make informed decisions throughout the farming cycle, from planting to harvest, and ultimately improve profitability [13]. Granular excels in collaboration features, which include shared tools for team members and advisors, streamlining workflows and enhancing communication within large farming operations [13].

Granular's advanced scouting features, such as satellite imagery and email alerts, enable farmers to monitor fields nearly in real-time, allowing them to identify issues early and act quickly [13]. Furthermore, the performance analysis tools offer field-specific insights, helping farmers optimize seed investments by identifying high-performing areas [13]. The system's user-friendly dashboards and seamless integration with platforms like John Deere simplify data analysis and make Granular an attractive option for farmers looking to optimize operations without extensive technical expertise [13].

Limitations and Weaknesses of Granular

Despite its numerous strengths, Granular does have limitations, particularly for smaller-scale farms or those in regions with less technological infrastructure. Cost is a significant barrier, especially for small-scale farmers, as advanced tools such as Granular Insights may be financially prohibitive [13]. The system's data-dependency also requires high-quality input data, which may not be readily available in underdeveloped or remote areas [13].

Another limitation is Granular's focus on analytics rather than real-time environmental automation. While the platform offers powerful data dashboards and field activity reports, it does not integrate IoT sensors for the automated control of farm equipment, such as irrigation and lighting [12]. As a result, farmers still need to manage these environmental factors manually or through other separate systems, which may reduce the overall automation benefits for large farms [12].

Granular's infrastructure is primarily designed for large-scale North American farms, and as such, it may not be easily scalable or financially viable for small-scale farmers in other regions [12]. The system's lack of features that support sustainable energy sources or water conservation also limits its appeal to environmentally conscious farmers or those working in resource-scarce areas [12].

Lastly, Granular's technical skill barrier could be challenging for farmers who are not familiar with advanced technology. Although the platform is designed to be user-friendly, certain features may require additional training [13]. Furthermore, Granular's reliance on internet connectivity may pose difficulties in remote areas with unreliable access, affecting the platform's overall usability [13].

Conclusion

Granular is a powerful farm management system that offers data-driven insights and operational efficiency for large-scale agricultural operations. While it excels in providing detailed financial and agronomic data, streamlining collaboration, and offering advanced scouting features, its high cost, data dependency, and lack of real-time environmental automation may limit its accessibility and scalability for smaller farms or those in resource-constrained regions. Despite these limitations, Granular remains a valuable tool for large-scale farmers seeking to optimize productivity and profitability through data analysis and field management.

2.2.3 Trimble

Figures 2.2.7 Trimble's Logo[24]

Trimble's Farm Management System is an advanced solution designed to optimize agricultural operations using data-driven insights and cutting-edge technologies. The system integrates planning, monitoring, and operational tools into one platform, making it versatile for various farm sizes and catering to both small and large-scale agricultural enterprises [14]. One of Trimble's key features is its field mapping and crop planning capabilities, which leverage GPS technology to create precise field maps. These maps include overlays of soil types, yield history, and topography, helping farmers make informed decisions for planting, spraying, and harvesting operations [15]. Additionally, the system includes crop health monitoring tools that use satellite imagery and drone integration to detect stressed areas in crops, allowing for early intervention to protect yield quality [15].

A critical component of Trimble's offering is its water management system. By integrating real-time data and soil moisture sensors, Trimble optimizes irrigation schedules, promoting water efficiency and aligning with sustainable farming practices [15]. Moreover, Trimble's farm financial tools track input costs, labor expenses, and revenue, while generating detailed profitability reports. These tools also assist in seasonal budgeting, allowing farmers to balance expenditures with expected returns [15]. The system emphasizes data integration and compatibility, seamlessly connecting with other Trimble solutions and third-party platforms. This level of integration ensures that farmers can manage different aspects of their operations cohesively [15]. Furthermore, the mobile app provides real-time updates, enabling users to manage their farms from any location [15].

Trimble's platform is scalable, making it suitable for a range of farming operations. It offers subscription options starting around $50 per month for basic features, with advanced plans for larger operations available upon request [14]. The system supports sustainable practices, promoting eco-friendly agriculture by tracking resource usage and optimizing inputs [15]. Despite these strengths, Trimble faces limitations. The initial setup and learning curve can be challenging for users unfamiliar with advanced technologies [16]. Additionally, subscription costs for full-feature packages may be prohibitive for smaller-scale farms [16].

Key Features of Trimble's Farm Management System

1. **Field Mapping & Crop Planning**
   - GPS-enabled field mapping to create precise field overlays with soil types, yield history, and topography.
   - Tools for planning planting, spraying, and harvesting operations based on accurate field data.
2. **Crop Health Monitoring**
   - Integration with satellite imagery and drones to detect crop stress and monitor overall field health.
   - Early intervention capabilities to protect yield quality and improve crop productivity.
3. **Water Management System**
   - Integration with soil moisture sensors for efficient irrigation scheduling.
   - Optimization of water use to align with sustainable farming practices.
4. **Farm Financial Tools**
   - Tracking of input costs, labor expenses, revenue, and profitability reports.
   - Seasonal budgeting features to help farmers balance expenditures with expected returns.
5. **Data Integration & Compatibility**
   - Seamless integration with other Trimble solutions and third-party platforms for unified farm management.
   - Real-time updates via a mobile app, enabling farmers to manage operations from anywhere.
6. **Scalability**
   - Flexible subscription plans starting at around $50 per month, with customized plans for larger operations.

Figures 2.2.8 Trimble's Computer and Mobile Apps[25]

Figures 2.2.9 Trimble's Data Integration and Compatibility Features[26]

Limitations of Trimble's Farm Management System

While Trimble offers numerous strengths, it does have limitations. The initial setup and learning curve may be difficult for those new to advanced farming technologies, and the subscription costs for full-feature packages may not be affordable for smaller-scale farms [16]. Moreover, real-time automation features like irrigation or environmental control are not integrated, and sustainability features, such as rainwater harvesting or solar power systems, are not prioritized [15]. These limitations make it less suitable for small-scale or resource-constrained farms.

2.3 Comparison Between Existing Systems and the Proposed System

Table 2.3.1 Comparison Systems

Feature / System	**Agrivi**	**Granular**	**Trimble**	**Proposed System**
Mobile Application	✓	✓	✓	✓
Real-time Automation (e.g., sensors)	✕	✕	✕	✓
Rainwater Harvesting Integration	✕	✕	✕	✓
Solar Power System Integration	✕	✕	✕	✓
IoT Sensor Support	✕	✕	✓	✓
Budget-Friendly for Small Farms	✕	✕	✕	✓
Task and Alert Automation	✓	✓	✓	✓

From the comparison above, it is evident that while Agrivi [4], [5], [6], Granular [12], [13], and Trimble [14], [15] are feature-rich and powerful platforms for commercial or large-scale farming, they do not address automation through IoT, nor do they integrate renewable energy or water sustainability features. The proposed system distinguishes itself by offering sensor-based irrigation, solar-powered operations, and rainwater harvesting, all controllable via a mobile application tailored for affordable and scalable implementation.

2.4 Summary

The literature review and analysis of existing smart farming systems reveal a significant gap in accessible, sustainable, and automated solutions tailored for small to medium-scale farms. Systems such as Agrivi [4], [5], [6], Granular [12], [13], and Trimble [14], [15] provide strong features for farm planning, record-keeping, and data analytics, but they fall short in supporting real-time sensor-based automation and environmental sustainability practices. These limitations highlight the need for a more integrated and cost-effective solution that can support farmers in resource-constrained environments.

The proposed Automated Sustainable Farm Management System seeks to address these shortcomings by combining real-time soil moisture monitoring, automated irrigation and lighting control, and the integration of renewable energy sources such as solar power and rainwater harvesting. In addition, the review of enabling technologies—Arduino UNO R4 WiFi, Python, and the Blynk IoT platform—demonstrates that the system is not only feasible but also scalable for future enhancements. Arduino provides reliable hardware interfacing with multiple sensors and actuators, Python offers powerful tools for data processing and visualization, and Blynk ensures real-time monitoring and remote control through mobile applications.

By integrating these technologies into a unified solution, the system aims to democratize smart farming by making it more accessible, affordable, and environmentally responsible. This establishes a solid foundation for the subsequent design, implementation, and evaluation of the project.

CHAPTER 3: Methodology

The development of the proposed automated sustainable farm management system follows the prototyping model, a user-focused and iterative software development approach. This methodology is particularly suitable for this project as it allows for early demonstration and testing of core functionalities, which can then be refined based on feedback and observations. The prototyping methodology used in this project consists of six main phases:

Phase 1: Requirement Gathering and Analysis

The first phase involved identifying the problems faced in small-scale farming, particularly in terms of water management, environmental monitoring, and sustainability. This was achieved through reviewing online resources and similar existing systems. The goal of this phase was to understand the needs of the users and define the basic requirements of the system, such as real-time soil moisture detection, water tank level monitoring, automation of irrigation, and sustainable power sourcing.

Phase 2: Initial Design

Based on the collected requirements, an initial system design was drafted. This included identifying the necessary hardware components such as the Arduino UNO R4 WiFi, soil moisture sensor, DHT11 sensor, LDR, ultrasonic sensor, relay module, water pump, and the solar charging circuit. The system flow was also planned out using a flowchart and block diagram to illustrate how the data would be processed and how each component would interact. The mobile app interface using Blynk was also considered during this phase to ensure it would support real-time monitoring and control.

Phase 3: Prototype Development

In this phase, the physical prototype of the system was assembled and programmed. All sensors were connected to the Arduino board and tested individually to ensure proper data collection. The logic for automatic control of the water pump and lighting was implemented using Arduino IDE. The Blynk mobile app was set up to receive sensor readings and control the actuators via

WiFi. At the same time, the solar-powered power system using TP4056 and a boost converter was partially developed and tested as a power source for the system.

Phase 4: Prototype Testing

The working prototype was tested in a controlled environment to evaluate its core functionalities such as moisture detection, water level monitoring, temperature and humidity reporting, and relay activation. The mobile application was also tested to verify live updates and user control.

Phase 5: Refinement and Iteration

Based on observations during testing, modifications were made to improve system accuracy and logic. Relay control logic was corrected for low-level triggering, and sensor reading intervals were optimized. The prototype wiring was adjusted to ensure reliability. Further improvements, such as enhancing power stability and expanding mobile control features, were noted for future implementation in the next development phase. The iterative nature of the prototype model allowed continuous enhancement without waiting for full system completion.

Phase 6: Final Review and Future Planning

Although the system is still in the prototyping stage, all major components have been integrated and validated through testing. In this phase, the current progress was reviewed, and plans for FYP2 were established. The final version will focus on full solar integration, improving reliability, optimizing the mobile interface, and implementing additional features such as push notifications or data logging. Regular maintenance and testing will also be planned to ensure the system remains efficient over time.

By adopting the prototyping methodology, this project allows early visibility of the system's functionality, enables iterative development, and ensures that feedback can be rapidly incorporated. This user-centric approach ensures the final system is practical, efficient, and aligns with the real needs of its intended users.

3.2 System Architecture Diagram

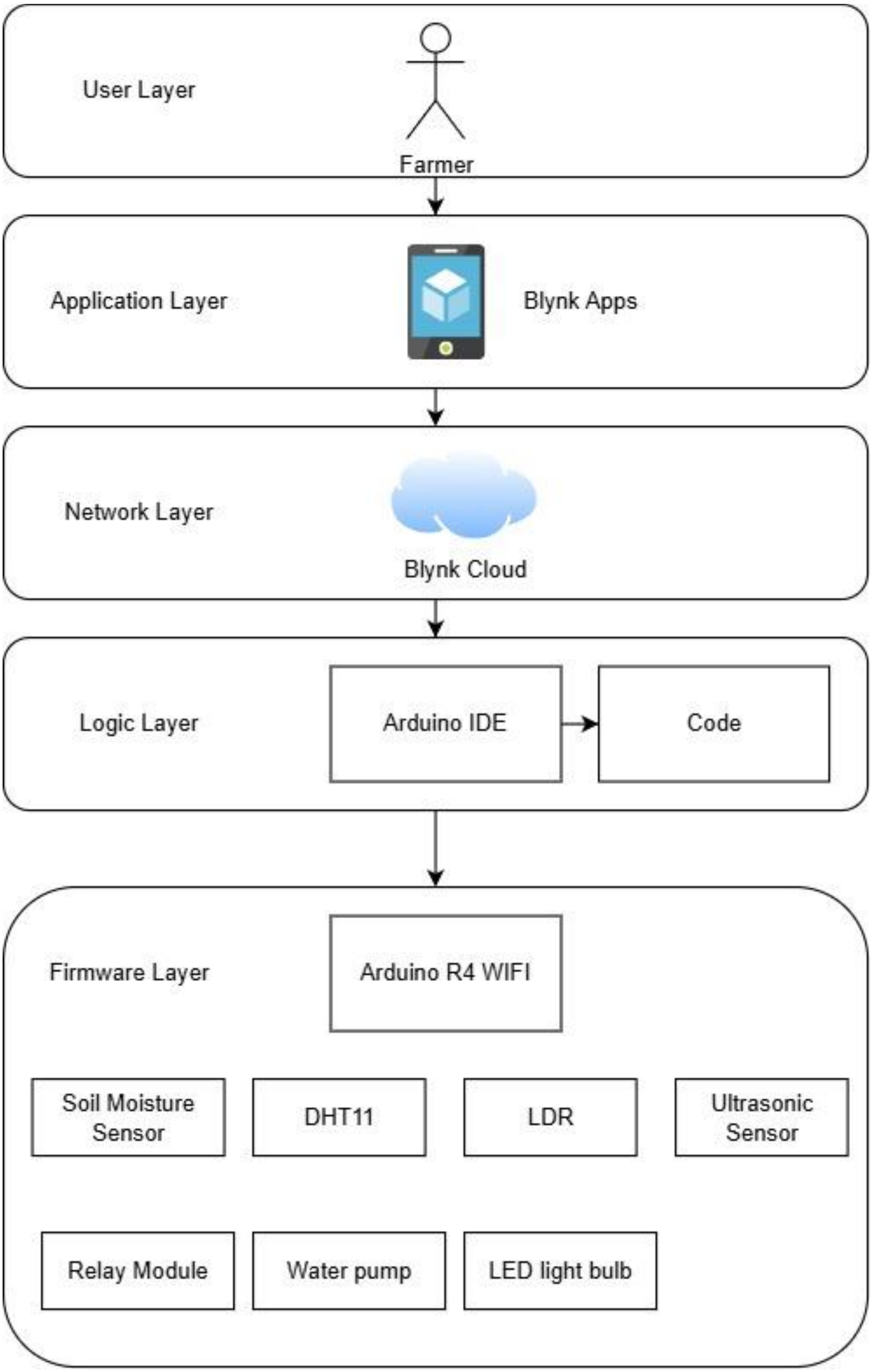


Figure 3.2.1 System Architecture Diagram

The Figure 3.2.1 system architecture diagram of the Automated Sustainable Farm Management System is organized into five interconnected layers that represent the flow of interaction from the user to the hardware components:

1. **User Layer (Farmer)**
   At the top of the diagram, the user layer represents the farmer who interacts with the system. The farmer initiates activities such as checking sensor readings, monitoring farm conditions, and sending control commands (e.g., turning on/off the water pump or light).

2. **Application Layer (Blynk Mobile App)**
   The farmer's actions are passed to the application layer, which is the Blynk mobile application. This layer provides the interface for real-time visualization of soil moisture, temperature, humidity, light levels, and water tank status. It also allows the farmer to send manual commands for irrigation or lighting.

3. **Network Layer (Blynk Cloud)**
   The application layer communicates with the network layer, represented by the Blynk Cloud. This layer serves as the intermediary that processes data requests and commands, ensuring secure communication between the mobile application and the hardware system.

4. **Logic Layer (Arduino IDE and Code)**
   The network layer then interacts with the logic layer, which consists of the Arduino IDE and the embedded program code. This layer processes incoming data from the cloud, interprets the farmer's commands, and executes control logic for automated decision-making (e.g., turning on the pump when soil moisture is low).

5. **Firmware Layer (Hardware Components)**
   At the final stage, the logic layer directs the firmware layer, which includes the physical components: the Arduino UNO R4 WiFi board, soil moisture sensor, LDR, DHT11 sensor, ultrasonic sensor, relay module, water pump, and light bulb. These hardware elements perform the actual actions, such as sensing environmental data or actuating the pump and light, with the results being sent back up through the layers for the farmer to view.

This layered flow ensures **end-to-end interaction**, where the farmer can monitor and control farm operations in real-time via the Blynk app, with automation handled by the Arduino logic and physical execution carried out by the connected hardware.

3.3 Use Case Diagram

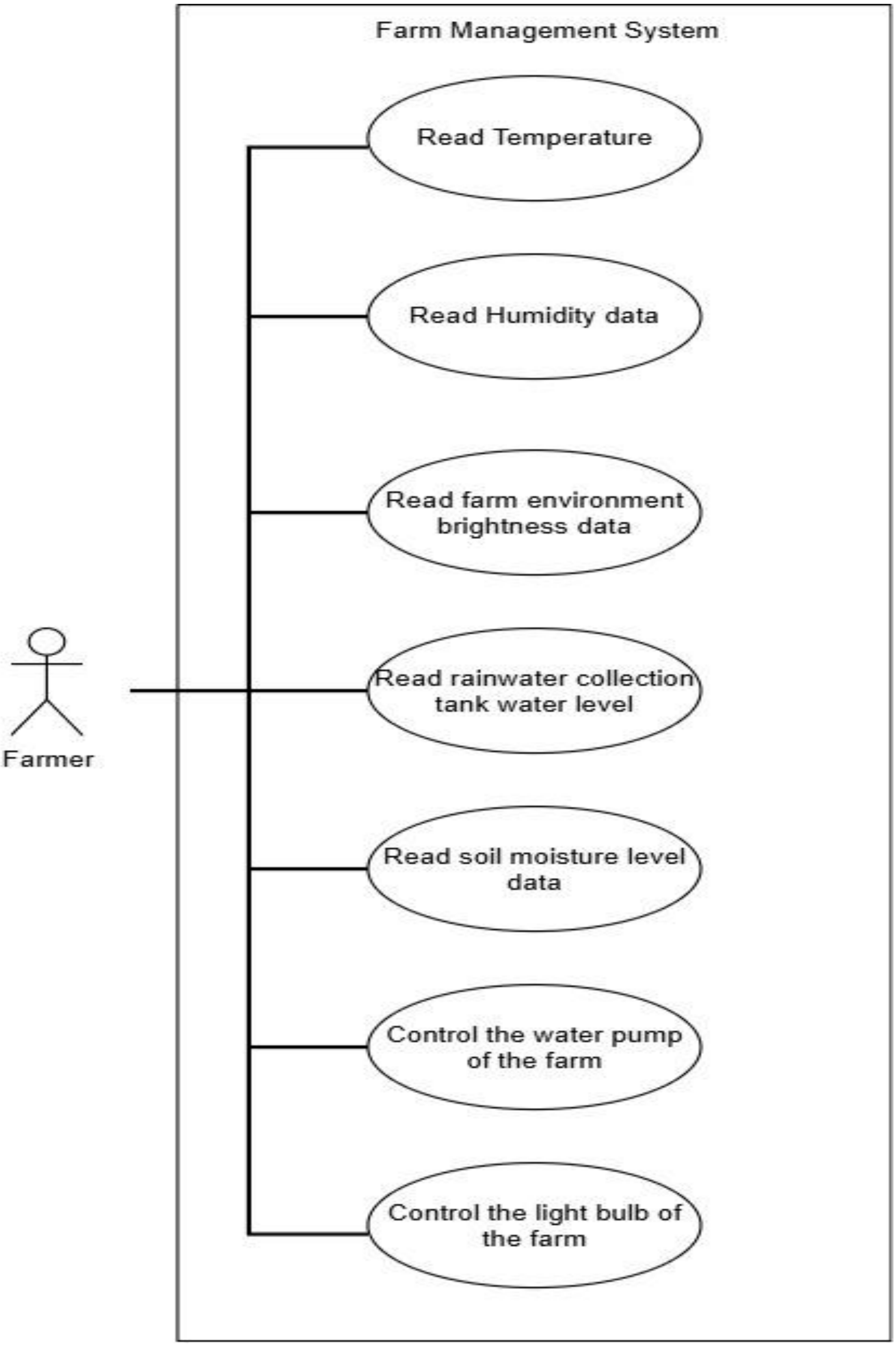


Figure 3.3.1 Use Case Diagram

The use case diagram for the Automated Sustainable Farm Management System illustrates the interactions between the primary user, the **farmer**, and the system. The farmer is the sole actor in this system and is responsible for both monitoring farm conditions and controlling specific actuators. The system provides several use cases that represent the farmer's capabilities:

1. **Read Temperature Data**
   The farmer can view real-time temperature readings collected by the DHT11 sensor. This helps monitor the farm's climate conditions.
2. **Read Humidity Data**
   The farmer can access humidity data from the DHT11 sensor to track moisture levels in the air, which is crucial for crop health.
3. **Read Farm Environment Brightness Data**
   The farmer can check light intensity through the LDR sensor. This information helps assess whether additional lighting is required for optimal plant growth.
4. **Read Rainwater Collection Tank Water Level**
   Using the ultrasonic sensor, the farmer can monitor the current water level in the rainwater collection tank. This ensures that sufficient water is available for irrigation.
5. **Read Soil Moisture Level Data**
   The farmer can read soil moisture levels from the soil moisture sensor. This data is vital for determining irrigation needs.
6. **Control the Water Pump**
   The farmer can turn the water pump on or off manually via the Blynk application. In addition, the system has automation logic to activate the pump when soil conditions are too dry.
7. **Control the Light Bulb**
   The farmer can switch the light bulb on or off manually to provide supplementary lighting when required. The system can also automatically control the light based on the brightness level detected by the LDR.

In summary, the use case diagram emphasizes the **dual functionality** of the system:

- **Monitoring**, where the farmer reads sensor data (temperature, humidity, brightness, water level, soil moisture).
- **Control**, where the farmer interacts with actuators (water pump and light bulb), either manually or with support from automation logic.

3.4 Activity Diagram

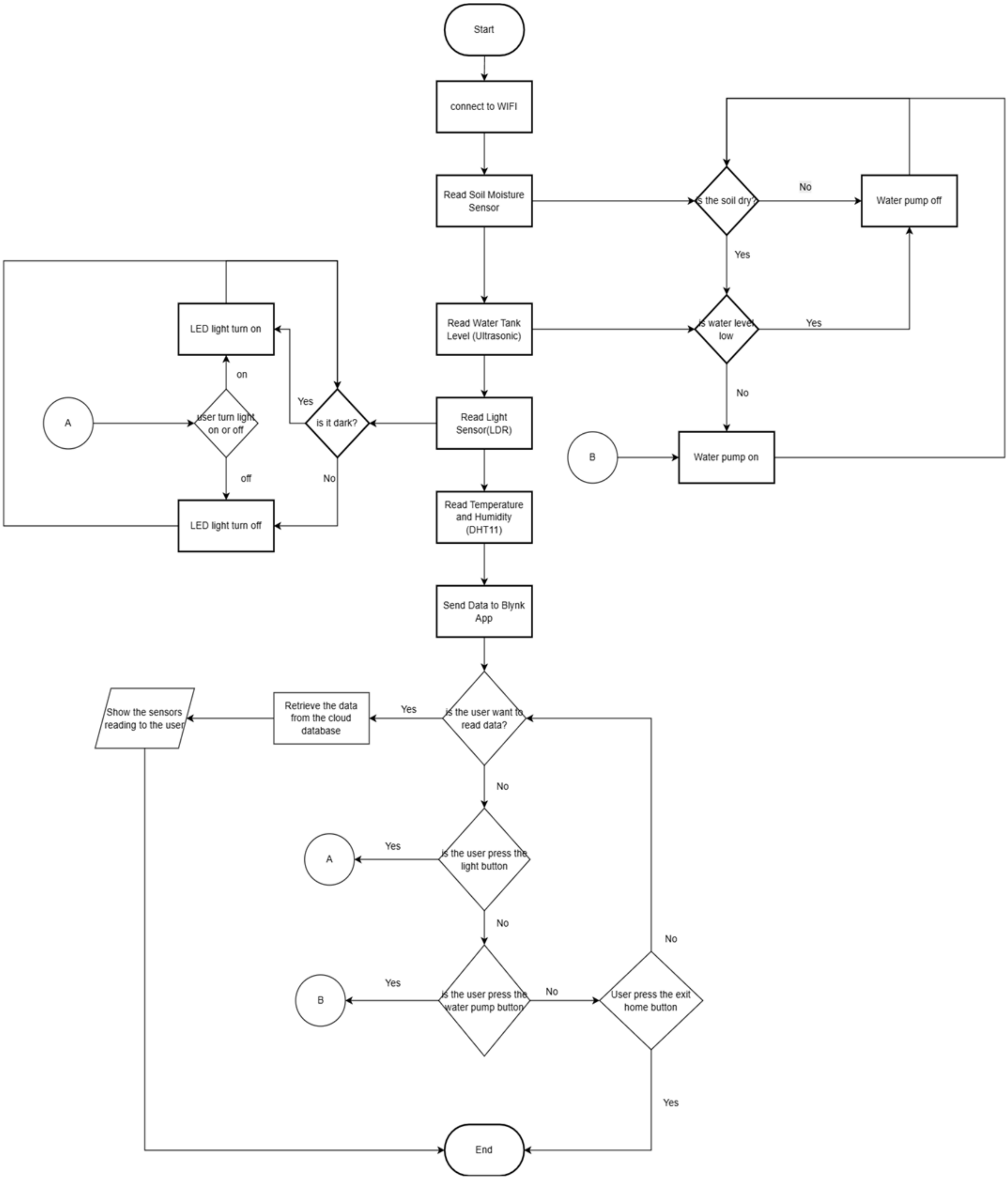


Figure 3.4.1 Activity Diagram of Automated Farm Management System

The diagram illustrates the operation of an automated smart farming system that monitors environmental conditions and controls irrigation and lighting based on sensor data. The system begins by establishing a Wi-Fi connection and reading data from several sensors: a soil moisture sensor, an ultrasonic sensor to check the water tank level, a light-dependent resistor (LDR) to detect light intensity, and a DHT11 sensor to capture temperature and humidity. If the soil is dry and the water tank level is sufficient, the water pump is activated to irrigate the plants. If the water level is too low, the pump remains off. The LDR sensor is used to detect whether it is dark; if so, the system checks if the user has turned on the light manually through the Blynk app, and then either turns on or off the LED lights accordingly.

All sensor data is transmitted to the Blynk app, enabling remote monitoring and manual control. Users can retrieve real-time data from the cloud database to view sensor readings. They are also given the option to manually control the system: turning on/off the LED lights or activating the water pump using buttons within the app. The flow concludes when the user exits the application via the "exit home" button. This system effectively integrates IoT components for automated and remote-controlled smart farming, enhancing convenience and optimizing resource usage.

3.5 Timeline

3.5.1 FYP1

Table 3.5.1 Project 1 Timeline – Gantt Chart

Activity	Week											
	1	2	3	4	5	6	7	8	9	10	11	12
Research and Planning												
Purchase & Gather Components												
Initial Hardware Assembly												
Test Individual Sensors												
Integrate Sensors with Arduino UNO R4 WiFi												
Connect Relay Module and Test Actuators												
Combine All Hardware for Initial Testing												
Soldering Components, Handle Wiring Challenges												
Fix Hardware Issues												
System Troubleshooting & Component Calibration												
FYP1 Report Writing												

3.5.2 FYP2

Table 3.5.2 Project 2 Timeline – Gantt Chart

Activity	Week											
	1	2	3	4	5	6	7	8	9	10	11	12
Set Up Software Environment												
Connect Arduino UNO to WiFi and Blynk Cloud												
Send Sensor Data												
Configure App Widgets to Display Real-Time Sensor Data												
Integrate Sensors with Arduino UNO R4 WiFi												
Implement Control to the system												
Implement Threshold Logic												
Debug and Refine Data Flow & Communication												
Optimize System												
Finalize FYP2 Report												

Chapter 4: System Design

4.1 System Block Diagram

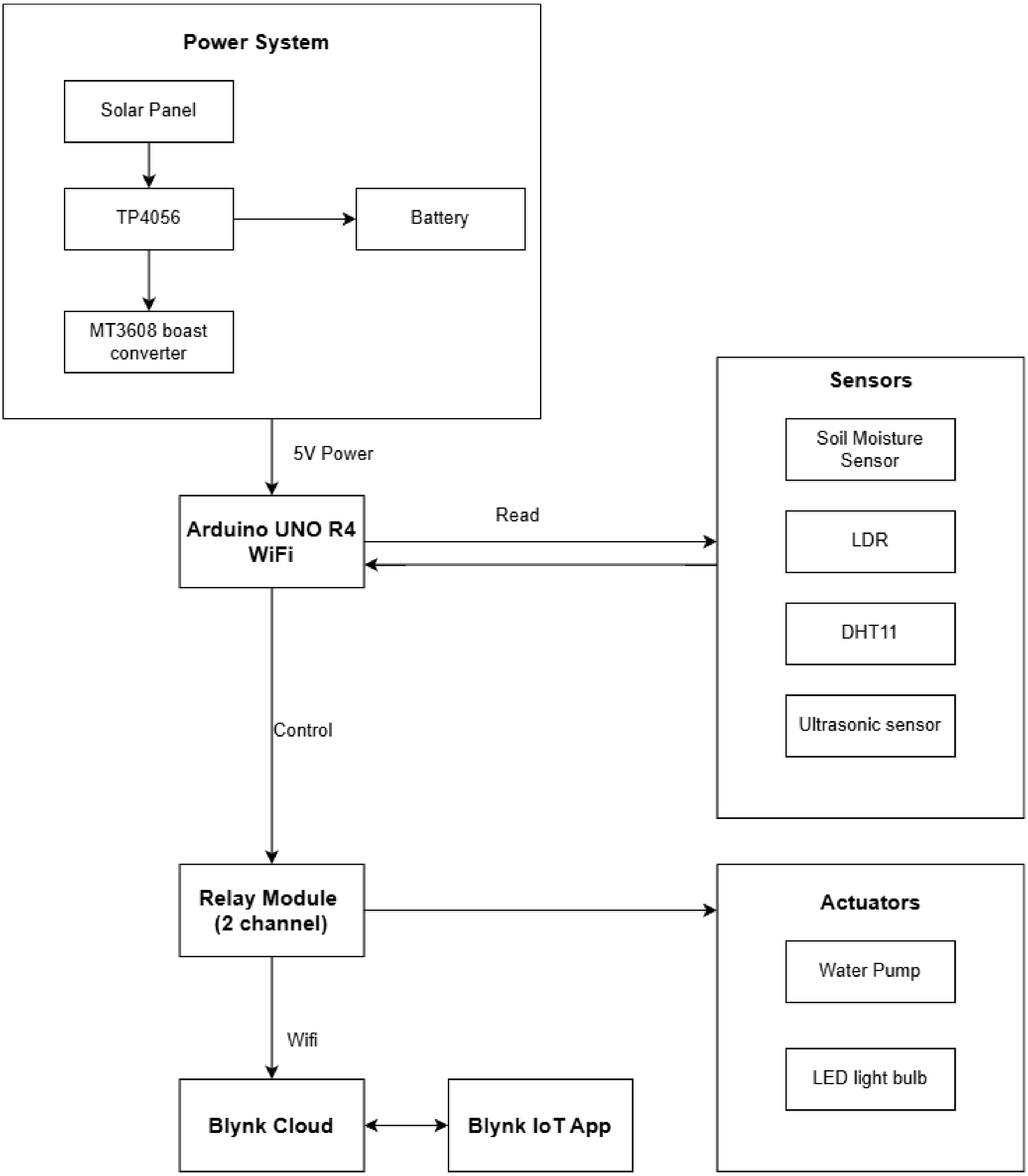


Figure 4.1.1 Block Diagram of the Farm Management System

The block diagram illustrates the hardware architecture of a solar-powered smart farming system based on Arduino and IoT integration. The system consists of five major components: the **power system**, **Arduino microcontroller**, **sensors**, **actuators**, and **Blynk IoT connectivity**.

The **power system** includes a solar panel that charges a battery through a TP4056 charging module. The MT3608 boost converter ensures that the power supplied to the system is a stable 5V, which is used to power the Arduino UNO R4 WiFi board. This setup makes the system energy-efficient and suitable for remote or off-grid agricultural use.

The **Arduino UNO R4 WiFi** acts as the central controller. It reads environmental data from various **sensors**, including the soil moisture sensor, LDR (light sensor), DHT11 (temperature and humidity sensor), and an ultrasonic sensor for water tank level detection. Based on sensor readings and user commands, the Arduino sends control signals to a **2-channel relay module**, which operates the **actuators**: a water pump for irrigation and an LED light bulb for artificial lighting.

The system also connects to the **Blynk IoT platform** via Wi-Fi. Sensor data is sent to the **Blynk Cloud**, allowing users to monitor real-time conditions through the **Blynk IoT App**. Users can also send commands from the app to manually activate the water pump or LED light. This remote control and monitoring capability make the system user-friendly and highly adaptable to modern farming needs.

In summary, this block diagram outlines a sustainable and automated IoT-based farming system that leverages renewable energy, sensor integration, and cloud-based control to support efficient agricultural operations.

4.2 System Specification

4.2.1 Hardware

Table 4.2.1 Hardware required

No.	Name	Specification	Description
1	**Arduino UNO R4 WiFi**	Microcontroller with built-in WiFi (Renesas chip)	Main microcontroller board to control all sensors, relays, and logic. Can also expand to WiFi monitoring later.
2	**Soil Moisture Sensor**	Analog output, 3.3V–5V operating voltage	Analog sensor to detect soil wetness or dryness to automatically control watering.
3	**LDR (Light Dependent Resistor) Module**	Digital output, 3.3V–5V operating voltage	Light sensor to detect day or night (brightness or darkness) and turn on/off the light bulb.
4	**DHT11 Temperature and Humidity Sensor**	Digital output, adjustable sensitivity	Digital sensor to measure ambient temperature (°C) and humidity (%), providing environmental data.
5	**Ultrasonic Sensor (HC-SR04)**	5V operating voltage, 2cm–400cm range	Distance measuring sensor used to monitor the water level in the tank (measure height by sound waves).
6	**2-Channel Relay Module**	5V relay control, supports high-power loads	Electronic switch module to control high-power devices like water pump and light bulb safely from Arduino.
7	**5V Mini Water Pump**	Submersible DC pump	Small DC pump to water the plants automatically when soil is dry (controlled via relay).
8	**Mini Light Bulb (5V LED Bulb)**	Low-power lighting	Small lighting element turned on during dark/night to simulate farm light conditions (controlled via relay).

9	**Solar Panel (6W, USB-C Output)**	6W output under full sunlight	Captures sunlight and converts it to electricity to charge the battery via TP4056.
10	**TP4056 Lithium Battery Charger Module (with Type-C Port)**	Lithium battery charging with protection	Module to safely charge 3.7V lithium battery using solar panel input. Includes overcharge and short circuit protection.
11	**3.7V Lithium-Ion Battery (e.g., 18650)**	Rechargeable battery, 2000mAh–3000mAh capacity	Rechargeable battery to store solar energy and provide continuous power to the system at night or when sunlight is low.
12	**MT3608 Boost Converter Module**	Step-up from 3.7V to 5.0V, adjustable output	Step-up converter to increase battery voltage (3.7–4.2V) to stable 5V output for Arduino and relays.
13	**Breadboard and Jumper Wires**	Standard 2.54mm pitch	Used for prototyping and connecting components without soldering initially. Jumper wires link modules together.
14	**USB Cable (for Arduino programming)**	Type-A to Type-C	To upload code from computer to Arduino. Also used for debugging Serial Monitor via Arduino IDE.
15	**Electrical Tape**	Insulation material	To safely insulate soldered joints and exposed wires, preventing short circuits.
16	**Solder Iron**	Soldering tools	To solder wire to connect to power system.
17	**Multimeter (for testing)**	Digital multimeter with DC voltage measurement	To measure voltages, check connections, and troubleshoot power issues.
18	**Water bucket**		To collect rainwater

4.2.2 Software

Table 4.2.2 Software required

Software	**Purpose**
Arduino IDE	To write and upload Arduino code to UNO R4 WiFi
Blynk Arduino Library	Library that allows Arduino to communicate with Blynk Cloud
WiFiS3 Library	WiFi support library for Arduino UNO R4 WiFi
Blynk IoT App	Real-time monitoring and remote control
Blynk Cloud	The server where Arduino sends sensor data and receives commands

4.3 Circuit Components and Design

The circuit design of the Automated Sustainable Farm Management System integrates multiple sensors, actuators, and a renewable energy power system with the Arduino UNO R4 WiFi as the central controller. This section explains how the individual components are interconnected and how the circuit operates as a whole.

4.3.1 Sensor Connections

- **Soil Moisture Sensor**: The analog output (AO) pin is connected to the Arduino analog input pin (A0). This allows the Arduino to measure soil moisture on a scale of 0–1023, where lower values indicate wet soil and higher values indicate dryness.

- **DHT11 Sensor**: The data pin is connected to digital pin D2 of the Arduino. This sensor provides digital readings for both temperature and humidity.

- **LDR Module**: The module provides a digital signal that is connected to Arduino digital pin D6. The signal indicates environmental brightness, where LOW represents bright conditions and HIGH represents dark conditions.

- **Ultrasonic Sensor (HC-SR04)**: The Trig pin is connected to digital pin D3, while the Echo pin is connected to digital pin D4. This setup enables the Arduino to calculate the water level in the rainwater collection tank based on the time taken for sound waves to reflect back.

4.3.2 Actuator Connections

- **Relay Module**: The system uses a two-channel relay module.
  - Relay channel 1 (IN1, COM, NO) is connected to Arduino pin D5 and controls the water pump. The relay acts as a switch to supply or cut off 5V power to the pump.
  - Relay channel 2 (IN2, COM, NO) is connected to Arduino pin D7 and controls the light bulb.

- **Water Pump**: The red wire (+) is connected to the NO terminal of relay channel 1, while the black wire (–) is connected to the system ground. The relay's COM terminal is connected to the 5V supply.

- **Light Bulb**: Connected in a similar way as the pump but through relay channel 2. The relay ensures that the Arduino can safely switch the bulb on or off.

4.3.3 Power System Connections

- **Solar Panel**: Supplies renewable energy to the TP4056 charging module.

- **TP4056 Module**: Regulates charging of the lithium-ion battery. The B+ and B– terminals are connected to the battery, while OUT+ and OUT– provide the battery output.

- **MT3608 Boost Converter**: Connected to the OUT+ and OUT– of the TP4056. The boost converter steps up the battery voltage to a stable 5V.

- **System Power Distribution**: The 5V output from the MT3608 is distributed to:
  - The 5V pin of the Arduino UNO R4 WiFi.
  - The VCC of the relay module.
  - The ground (VOUT–) is connected to the Arduino GND and the relay module GND to complete the circuit.

4.3.4 Circuit Operation

When the system is powered, the sensors continuously feed environmental data to the Arduino. The microcontroller processes the data based on predefined thresholds. If the soil moisture level is too low, the Arduino activates the relay to power the water pump. Similarly, if the LDR detects darkness, the Arduino activates the relay to switch on the light bulb. All sensor readings and actuator statuses are transmitted to the Blynk IoT platform for real-time monitoring and manual control by the farmer.

4.3.5 Block Circuit Diagram

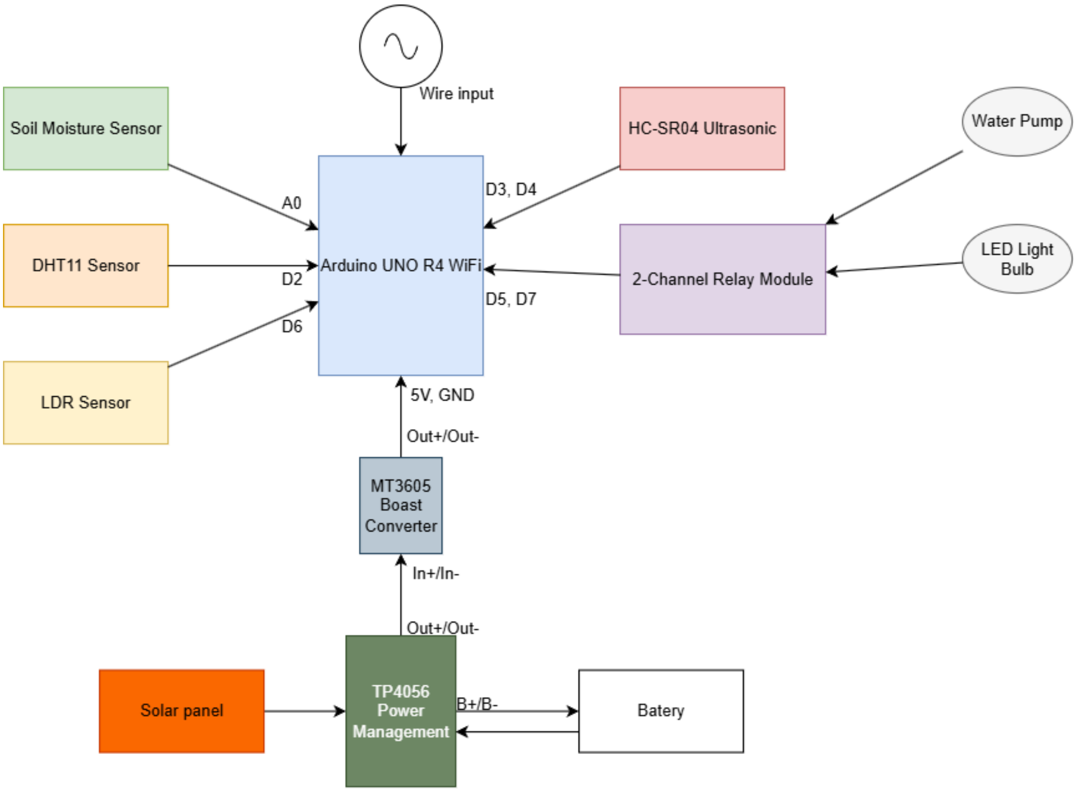


Figure 4.3.1 Block Circuit Diagram

The complete wiring of the system is illustrated in the circuit diagram (Figure 4.3.1). The diagram shows the interconnections between the Arduino, sensors, actuators, and the solar-based power supply system.

4.4 System Development

The development of the Automated Sustainable Farm Management System was carried out systematically in stages to ensure proper integration of hardware and software components. This section explains in detail the process of building the prototype, covering hardware development, software implementation, IoT integration, power system design, and testing.

4.4.1 Hardware Development

The hardware development stage involved the selection, integration, and testing of various sensors, actuators, and the control unit. The Arduino Uno R4 WiFi was selected as the main microcontroller because of its built-in WiFi connectivity, sufficient number of digital and analog pins, and compatibility with IoT platforms such as Blynk.

Several sensors were connected to the Arduino to monitor different environmental conditions. The soil moisture sensor was connected to the analog pin (A0) to measure soil wetness, where values closer to 0 indicated wet soil and values near 1023 indicated dry soil. The DHT11 sensor was connected to digital pin D2 to measure both ambient temperature and humidity. In addition, the LDR light sensor module was connected to digital pin D6 to detect the brightness of the farm environment, while the HC-SR04 ultrasonic sensor was connected to pins D3 (Trig) and D4 (Echo) to measure the water level in the rainwater collection tank.

The actuators of the system included a water pump and an LED light bulb, both of which were controlled through a two-channel relay module. Relay Channel 1 (IN1 connected to D5) controlled the water pump, while Relay Channel 2 (IN2 connected to D7) controlled the LED bulb. The relay module played a critical role in ensuring safe switching of higher power devices using low-voltage signals from the Arduino, thus providing both safety and reliability. All wiring was first tested on a breadboard and then soldered to ensure stability of the system.

4.4.2 Software Development (Arduino IDE)

The software development process was carried out using the Arduino IDE and programmed in C++. The first stage involved integrating essential libraries required for system operation. The WiFiS3.h library was used to establish WiFi connectivity on the Arduino Uno R4 WiFi, the BlynkSimpleWiFi.h library was used for communication with the Blynk Cloud, and the DHT.h library enabled the reading of temperature and humidity data from the DHT11 sensor.

The program was structured in two main phases: setup and loop. In the setup phase, all sensors, relays, and WiFi connections were initialized. In the loop phase, the system continuously monitored sensor data, updated readings to the Blynk Cloud, and executed the control logic. The control logic ensured that the water pump was activated based on soil moisture readings and the availability of water in the tank, while the LED bulb was switched on when the light level dropped below a threshold. The farmer was also provided with the option to override the automation manually via the Blynk mobile app, giving the system flexibility for both automated and manual operation.

4.4.3 IoT Integration with Blynk

For IoT integration, the Blynk platform was selected due to its user-friendly mobile application and compatibility with Arduino devices. The system was registered in the Blynk Cloud using a unique Template ID, Device Name, and Authentication Token. Each sensor and actuator was mapped to a virtual pin within the Blynk platform to enable smooth data transmission and control. For example, soil moisture data was mapped to V0, temperature and humidity to V1 and V2 respectively, light level to V3, and water level distance to V4. Virtual switches were also assigned to V5 for pump control and V6 for light control.

Through this integration, real-time sensor values were transmitted to the Blynk Cloud and displayed on the mobile dashboard, while control commands from the farmer were received by the Arduino through the same virtual pin assignments. This two-way communication allowed the farmer to remotely monitor environmental conditions and control actuators in real time, fulfilling one of the main objectives of the project.

4.4.4 Power System Development

To enhance sustainability, a solar-powered backup system was included in the project. A small solar panel was connected to a TP4056 charging module, which ensured safe charging of a lithium-ion battery. The MT3608 boost converter was then used to step up the 3.7V output of the lithium battery to 5V, which was required to power the Arduino and relay modules.

During testing, the solar power system successfully demonstrated energy flow from the solar panel to the battery and then to the Arduino board. However, due to the small size of the panel and limitations of indoor light conditions, the solar system was unable to provide continuous stable power. Therefore, as a backup, USB power from a laptop or external adapter was used to ensure uninterrupted operation during testing. This validated the feasibility of renewable energy integration while also highlighting the need for higher-capacity components in future iterations.

4.4.5 System Testing and Refinement

The system underwent multiple stages of testing to validate performance. First, each sensor was tested individually to ensure proper readings. For example, covering the LDR simulated darkness, and placing the ultrasonic sensor above a container of water confirmed distance detection. Next, the relay module and actuators were tested by turning the water pump and light bulb ON and OFF through both the Arduino code and the Blynk app. Once the individual tests were completed, full integration testing was performed where all components were connected together to validate both automatic and manual control modes.

Several refinements were made during the process. Initially, the relay logic was inverted, causing the actuators to switch incorrectly, but this was corrected in the code. Additionally, the soil moisture threshold values were adjusted to ensure more realistic irrigation control. Finally, due to limitations in the solar panel's capacity, USB backup power was added to ensure stable operation.

4.4.6 Final Prototype

The final prototype successfully demonstrated all the intended features of the system. It provided real-time monitoring of farm conditions such as temperature, humidity, soil moisture, brightness, and water level. Automated irrigation and lighting were triggered by sensor readings, while manual override was available through the Blynk app. Furthermore, the integration of a solar power system, though limited, proved the concept of renewable energy support. Together, these features confirmed that the system met its design objectives and served as a practical prototype for smart and sustainable farm management.

Chapter 5: System Implementation

5.1 System Setup

5.1.1 Hardware Setup

The hardware setup consists of connecting the Arduino Uno R4 WiFi microcontroller to multiple sensors, actuators, and the solar power management system. Each component was wired according to its respective specifications to ensure correct data acquisition and system operation.

Soil Moisture Sensor

Table 5.1.1 soil moisture sensor pin connection

Pin	Connects To
VCC	Arduino 5V
GND	Arduino GND
AO	A0 (analog read pin)

LDR Light Sensor Module

Table 5.1.2 LDR light sensor pin connection

Pin	Connects To
VCC	Arduino 5V
GND	Arduino GND
D0	D6 (digital pin)

DHT11 Temperature + Humidity Sensor

Table 5.1.3 DHT11 sensor pin connection

Pin	**Connects To**
VCC	Arduino 5V
GND	Arduino GND
DATA	D2

Ultrasonic Sensor (HC-SR04)

Table 5.1.4 Ultrasonic sensor pin connection

Pin	**Connects To**
VCC	Arduino 5V
GND	Arduino GND
Trig	D3
Echo	D4

2-Channel Relay Module

Table 5.1.5 2-channel Relay Module pin connection

Pin	**Connects To**	**Purpose**
VCC	Arduino 5V	Module power
GND	Arduino GND	
IN1	D5	Control pump
IN2	D7	Control light bulb

WATER PUMP WIRING (via Relay CH1)

Table 5.1.6 water pump wiring connection

Wire	Connects To
Red (+)	Relay CH1 NO
Black (–)	Arduino GND
Relay COM	Boost Converter OUT+ (5V)

LED LIGHT BULB WIRING (via Relay CH2)

Table 5.1.7 LED light wiring connection

Wire	Connects To
Red (+)	Relay CH2 NO
Black (–)	Arduino GND
Relay COM	Boost Converter OUT+ (5V)

Summary of Arduino Pins Used

Table 5.1.8 Summary of Arduino Pin

Arduino Pin	Module/Function
A0	Soil Moisture Sensor
D2	DHT11 (temperature/humidity)
D3	HC-SR04 Trig (Ultrasonic)
D4	HC-SR04 Echo (Ultrasonic)
D5	Relay IN1 (Pump)
D6	LDR Module D0
D7	Relay IN2 (Light Bulb)

Power System Wiring

Table 5.1.9 Power System Wiring

Component	Pin/Terminal	Connects To
Solar Panel	+ / –	TP4056 IN+ / IN–
TP4056 B+ / B–	Battery + / –	Lithium-ion battery
TP4056 OUT+ / OUT–	MT3608 VIN+ / VIN–	Boost Converter Input
MT3608 VOUT+	Relay COM (CH1 + CH2), Arduino 5V	Power Output
MT3608 VOUT–	Arduino GND	Ground

5.1.2 Software Setup

The **Arduino IDE** was used for development and code uploading. The following libraries were installed to support sensor reading, WiFi connectivity, and Blynk IoT communication:

- **WiFiS3.h** – for Arduino Uno R4 WiFi internet connectivity
- **BlynkSimpleWiFi.h** – for communication between Arduino and Blynk Cloud
- **DHT.h** – for reading data from the DHT11 temperature and humidity sensor

The Arduino Uno R4 WiFi was programmed via USB Type-C cable, and testing was carried out using the built-in Serial Monitor.

5.2 System Setting and Configuration

The proposed Automated Farm Management System integrates sensors, actuators, and a WiFi-enabled Arduino Uno R4 to enable automated irrigation and environmental monitoring. The configuration of the software plays a crucial role in linking the physical hardware to the Blynk IoT platform for seamless monitoring and control. The settings are divided into WiFi/Blynk initialization, pin assignments, virtual pin mapping, and control logic.

---

1. WiFi and Blynk Initialization

At the start of the program, the Blynk IoT credentials and WiFi details are defined:

```
#define BLYNK_TEMPLATE_ID "TMPL6Ou4J9td4"
#define BLYNK_TEMPLATE_NAME "Smart Farm"
#define BLYNK_DEVICE_NAME   "MyArduinoR4"
#define BLYNK_AUTH_TOKEN    "q33MJU_R4007LdYyGkRSw-E_aeOgud8e"



#include <WiFiS3.h>                     // For Arduino UNO R4 WiFi
#include <DHT.h>                       // For DHT11 sensor
#include <BlynkSimpleWifi.h>           // Blynk library


// ====== WiFi and Blynk Setup ======
char auth[] = "q33MJU_R4007LdYyGkRSw-E_aeOgud8e";
char ssid[] = [redacted]
char pass[] = [redacted]
```

Figure 5.2.1 Blynk Initialization and WiFi configuration

- The **BLYNK_TEMPLATE_ID, TEMPLATE_NAME, and DEVICE_NAME** are identifiers that bind this Arduino device to the project created in the Blynk Cloud.
- The **BLYNK_AUTH_TOKEN** is a unique authentication key that ensures secure communication between the device and the cloud.
- The Arduino connects to the internet via the **WiFiS3 library**, which enables WiFi functionality on the Arduino Uno R4 WiFi.

When Blynk.begin(auth, ssid, pass) is called in the setup() function, the Arduino connects to the WiFi network and authenticates with the Blynk server.

2. Pin Assignments and Configuration

Each sensor and actuator is mapped to a specific hardware pin on the Arduino. For example:

```
// ====== Pin Definitions ======
#define SOIL_PIN A0
#define LDR_PIN 6
#define DHT_PIN 2
#define TRIG_PIN 3
#define ECHO_PIN 4
#define RELAY_PUMP 5
#define RELAY_LIGHT 7
```

Figure 5.2.2 Pin Assignment

- **Soil moisture sensor**: connected to analog pin A0 since it produces an analog voltage proportional to soil wetness.
- **LDR light sensor**: connected to digital pin D6, which reads HIGH/LOW depending on brightness threshold.
- **DHT11**: connected to digital pin D2, providing temperature and humidity values.
- **Ultrasonic sensor (HC-SR04)**: uses pins D3 (trigger) and D4 (echo) to calculate distance based on sound pulse timing.
- **Relay module**: uses pins D5 and D7 to control the water pump and light bulb.

Pins are configured in the setup() function using pinMode(), and relays are initialized in the OFF state:

```
// Relay Pins
pinMode(RELAY_PUMP, OUTPUT);
pinMode(RELAY_LIGHT, OUTPUT);

// Set relays to OFF initially
digitalWrite(RELAY_PUMP, RELAY_OFF);
digitalWrite(RELAY_LIGHT, RELAY_OFF);
```

Figure 5.2.3 Relay Configuration

The relays use **active-low logic**, which means:

- LOW signal = Relay ON (circuit closed, device powered).
- HIGH signal = Relay OFF (circuit open, device disconnected).

This is accounted for using the constants:

```
#define RELAY_ON  LOW
#define RELAY_OFF HIGH
```

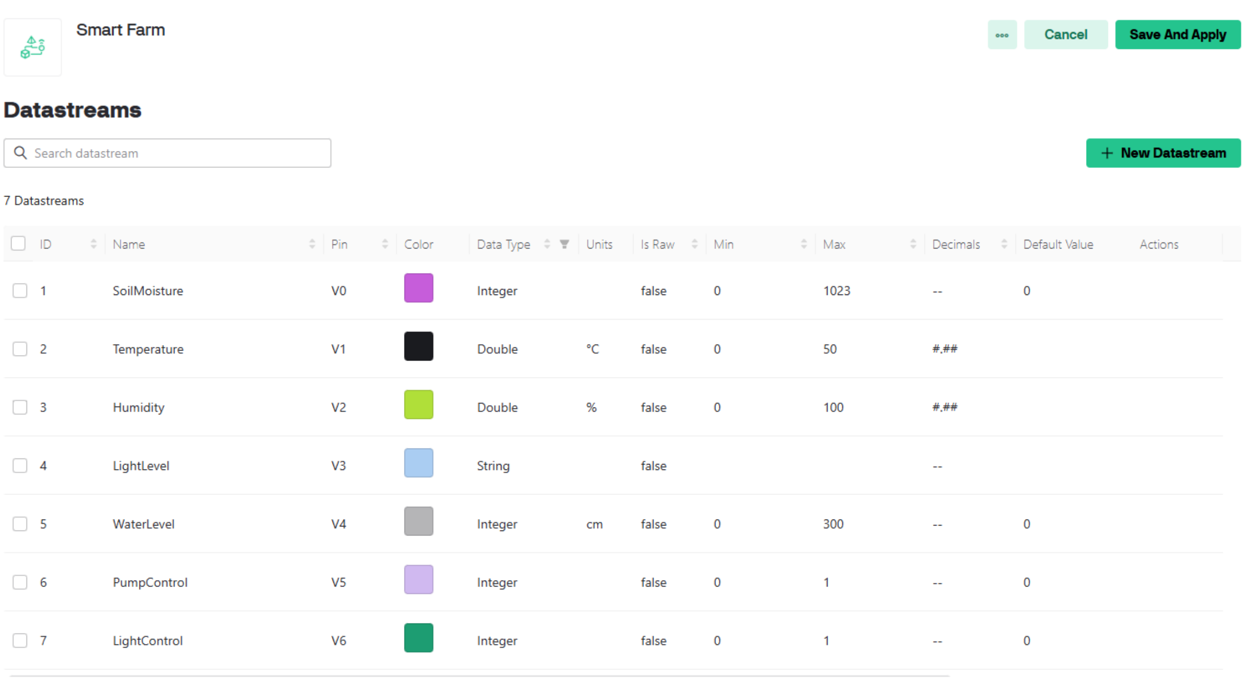
Figure 5.2.4 Relay Assignment

3. Virtual Pin Mapping in Blynk

The Blynk mobile app communicates with the Arduino via **virtual pins**, which act as intermediaries between hardware pins and the app interface. In this system, the following mappings were defined:

Table 5.2.1 Datastream of Blynk Cloud

Virtual Pin	Purpose
V0	Soil Moisture (Integer)
V1	Temperature (Float)
V2	Humidity (Float)
V3	Light Level (String - “BRIGHT” / “DARK”)
V4	Water Tank Distance (Integer)
V5	Switch to manually control Water Pump
V6	Switch to manually control Light Bulb



ID	Name	Pin	Color	Data Type	Units	Is Raw	Min	Max	Decimals	Default Value	Actions
1	SoilMoisture	V0		Integer		false	0	1023	--	0	
2	Temperature	V1		Double	°C	false	0	50	#.##		
3	Humidity	V2		Double	%	false	0	100	#.##		
4	LightLevel	V3		String		false			--		
5	WaterLevel	V4		Integer	cm	false	0	300	--	0	
6	PumpControl	V5		Integer		false	0	1	--	0	
7	LightControl	V6		Integer		false	0	1	--	0	

Figure 5.2.5 Setting Blynk Cloud DataStream

For example, the soil moisture reading is sent to Blynk with:

```
// === Soil Moisture ===
int soilValue = analogRead(SOIL_PIN); // 0 (wet) to 1023 (dry)
Blynk.virtualWrite(V0, soilValue);
Serial.print("Soil Moisture: ");
Serial.println(soilValue);
```

Figure 5.2.6 Arduino Read of Soil Moisture data

On the other hand, control input from the Blynk app is received using BLYNK_WRITE():

```
// ==== BLYNK CONTROL ====
BLYNK_WRITE(V5) { pumpControl = param.asInt(); }
BLYNK_WRITE(V6) { lightControl = param.asInt();
                  Serial.print("Blynk Manual LightControl = ");
                  Serial.println(lightControl);}
```

Figure 5.2.7 Control Input From Blynk

This structure ensures that any user action on the Blynk app (e.g., toggling a switch) is immediately reflected in the Arduino logic.

---

4. Sensor Data Processing

The program continuously reads sensor data and processes it in the updateSensorsAndLogic() function:

- **Soil Moisture**: analogRead() returns values from 0 (wet) to 1023 (dry). Thresholds are used to decide irrigation needs.
- **DHT11**: dht.readTemperature() and dht.readHumidity() provide environmental data, which is both logged in Serial Monitor and sent to Blynk.
- **LDR**: digitalRead(LDR_PIN) returns HIGH (dark) or LOW (bright), which is mapped to human-readable strings (“DARK”/“BRIGHT”).
- **Ultrasonic Sensor**: Distance is calculated from the pulse duration using:

```
int distanceCM = duration * 0.034 / 2;
```

Figure 5.2.8 Ultrasonic Distance Calculation

5. Actuator Control Logic

The relay module manages the pump and light bulb. Control can be **manual (via Blynk app)** or **automatic (based on sensor readings)**:

- **Pump Control**:
  - Manual ON when V5 is toggled.
  - Automatic ON if soil is dry (soilValue > 600) and water tank has sufficient level (distanceCM < 20).
- **Light Control**:
  - Manual ON when V6 is toggled.
  - Automatic ON if the LDR detects darkness (lightState == HIGH).

Example code for light control:

```
// === LIGHT CONTROL ===
if (lightControl == 1) {
  digitalWrite(RELAY_LIGHT, RELAY_ON); // Manual ON
  Serial.println("Light ON (Manual from App)");
} else {
  if (lightState == HIGH) {
    digitalWrite(RELAY_LIGHT, RELAY_ON); // Auto ON in dark
    Serial.println("Light ON (Auto: Dark)");
  } else {
    digitalWrite(RELAY_LIGHT, RELAY_OFF);
    Serial.println("Light OFF (Auto: Bright)");
  }
}
```

Figure 5.2.9 Actuators Control Logic

6. Data Transmission and Debugging

Every 3 seconds, all sensor values and relay states are updated in both:

- **Blynk Cloud** (via Blynk.virtualWrite) → displayed on the mobile app.
- **Serial Monitor** (via Serial.print) → for debugging and validation during testing.

This dual output ensures that even if the Blynk app is unavailable, the system can still be validated locally.

Data Flow in the System

The flow of data within the Automated Farm Management System follows a closed-loop IoT communication model. First, raw environmental data is collected by the sensors — the soil moisture sensor measures soil wetness, the DHT11 provides temperature and humidity readings, the LDR detects environmental brightness, and the ultrasonic sensor determines the water level in the rainwater collection tank. These sensor readings are transmitted to the **Arduino Uno R4 WiFi**, where they are processed by the programmed logic. The Arduino then sends this processed data to the **Blynk Cloud** using the Blynk.virtualWrite() function. Once in the cloud, the data is displayed on the **Blynk mobile application**, where the farmer can view real-time conditions of the farm environment.

In addition to data visualization, the mobile application also enables user interaction. When the farmer toggles a control switch in the Blynk app (e.g., to turn the water pump or light bulb on/off), this command is transmitted from the **mobile app to the Blynk Cloud**, and then relayed back to the **Arduino Uno R4 WiFi** through the BLYNK_WRITE() function. Upon receiving the command, the Arduino updates the actuator state by sending signals to the relay module, which in turn activates or deactivates the water pump or the light bulb. This bi-directional communication ensures that the system can operate in both **automatic mode** (driven by sensor readings and predefined thresholds) and **manual mode** (controlled directly by the user through the mobile app).

5.3 System Operation

This section describes how the Automated Farm Management System works once powered on and connected to the Blynk Cloud. It covers the **data flow**, the **decision-making process** of the Arduino, and how users interact with the system through the mobile app.

---

1. System Initialization

- When powered, the **Arduino Uno R4 WiFi** initializes the sensors, relay module, and establishes a WiFi connection using the credentials stored in the program.
- It then connects securely to the **Blynk Cloud** using the authentication token.
- All relays are set to OFF by default to ensure safe startup conditions.

---

2. Data Collection

- The sensors continuously monitor environmental parameters:
  - Soil moisture (A0)
  - Temperature & humidity (DHT11, D2)
  - Ambient brightness (LDR, D6)
  - Water tank level (Ultrasonic, D3/D4)
- These values are read periodically (every 3 seconds in this prototype).

---

3. Data Transmission and Monitoring

- The Arduino sends sensor data to the **Blynk Cloud** using virtual pins (Blynk.virtualWrite).
- The **Blynk mobile application** retrieves this data and displays it in real time through dashboards, labels, and graphs.
- Simultaneously, the data is also displayed in the **Arduino IDE Serial Monitor** for debugging and verification.

---

4. Decision-Making and Automation

- The Arduino executes programmed logic:
  - If soil moisture is low and water is available, the pump is activated.
  - If ambient light is low, the light bulb is switched on.
- This ensures **automatic farm operation** without requiring manual intervention.

---

5. User Control through Blynk App

- The farmer can override automation by toggling virtual switches:
  - **V5** for water pump control.
  - **V6** for light control.
- When a switch is activated in the app, the command is sent to the **Blynk Cloud**, then relayed back to the Arduino through the BLYNK_WRITE function.
- The Arduino updates the relay states accordingly, switching the pump or light ON/OFF.

---

6. Data Flow

The complete flow of data and control is summarized as follows:

Sensors → Arduino Uno R4 (logic) → Blynk Cloud → Mobile App (monitoring/control) → Blynk Cloud → Arduino Uno R4 → Relays → Actuators (pump/light)

This closed-loop ensures real-time monitoring, responsive automation, and user-driven manual control.

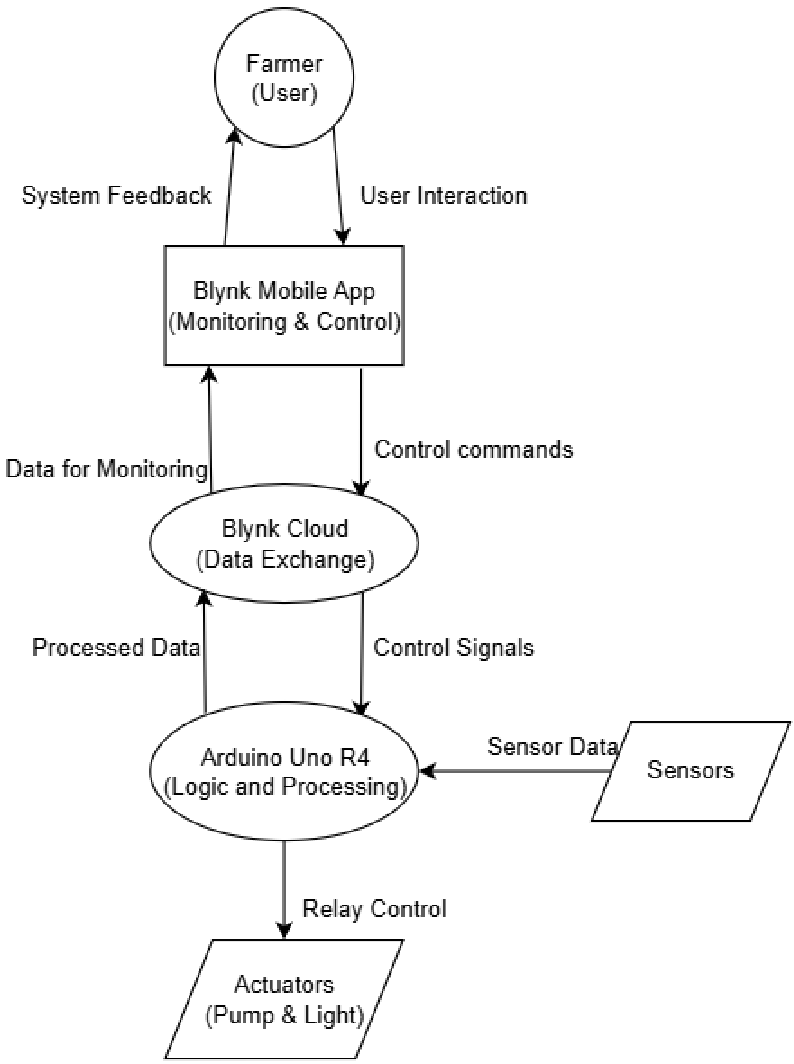


Figure 5.3.1: Data Flow Diagram of the Automated Farm Management System

The diagram illustrates the flow of data between the farmer, sensors, Arduino Uno R4, Blynk Cloud, and actuators. Sensor readings are processed by the Arduino and transmitted to the Blynk Cloud for monitoring via the mobile app, while user commands from the app are sent back through the cloud to control the pump and light.

5.4 Implementation Issues and Challenges

Throughout the development of the automated sustainable farm management system, several technical challenges and issues were encountered that required troubleshooting and modifications.

The first major challenge was related to the **power management circuit**, specifically the boost converter. Initially, the XL6009 boost converter was selected to step up the voltage from a 3.7V lithium battery to the 5V required by the Arduino Uno R4 WiFi. However, it was later discovered that the XL6009 was not designed to operate efficiently with low input voltages such as 3.7V, which resulted in unstable power delivery and frequent system failures. This issue occurred due to a misunderstanding of the converter's input requirements. To address this, the XL6009 was replaced with the MT3608 boost converter, which can reliably operate at lower voltages. The replacement successfully stabilized the power output, allowing the controller and sensors to function consistently.

Another significant challenge came from the **solar power system** itself. During testing, the solar panel was unable to provide a stable electrical output indoors, and even in sunlight, the voltage was insufficient to continuously power the entire system, especially when multiple sensors and actuators were active. As a result, the Arduino and relay module occasionally lost power or restarted unexpectedly. To ensure reliable operation, a decision was made to configure the **wired USB power supply as the main input**, while the solar panel and battery were kept as a **backup power source**. This adjustment allowed testing and demonstration of the system to proceed smoothly without the risk of power instability while still showcasing the sustainability feature.

Challenges also arose in **soldering and assembly** of the circuit components. Several poor soldering points caused intermittent connections, leading to unstable sensor readings and power fluctuations. These issues required rework with proper soldering techniques and the use of rosin flux to ensure secure and conductive joints.

A malfunction in the **soil moisture sensor** also posed difficulties during testing. The sensor occasionally produced inconsistent readings that did not correspond to actual soil conditions. After investigation, it was determined that the issue was either due to a faulty sensor or sensitivity to voltage fluctuations. The problem was resolved by replacing the sensor with a new unit to ensure accurate measurements.

On the **software side**, the integration of sensors with the Blynk IoT platform introduced its own set of issues. The WiFi connection was sometimes unstable, leading to temporary disconnections from the Blynk Cloud. This caused interruptions in data transmission and actuator control. To mitigate this, the code was optimized with reconnection routines and delays were carefully managed to prevent data lag or app instability. Synchronizing sensor updates with the Blynk app also required precise handling of data formats to ensure smooth real-time monitoring.

Finally, **time management** was an overarching challenge. Considerable delays were experienced during troubleshooting and debugging, particularly when sourcing replacement parts such as the boost converter and sensors. These setbacks extended the development timeline but also provided valuable lessons in project management and contingency planning.

Despite these challenges, the project was successfully implemented through a process of continuous refinement. Each obstacle provided practical learning opportunities in areas such as **hardware troubleshooting, component compatibility, power management, and IoT system integration**, which are essential skills for real-world embedded system development.

5.5 Summary

This chapter has presented the overall implementation of the Automated Sustainable Farm Management System, covering both the hardware and software setup, system configuration, operation, and the challenges faced during development. The hardware implementation included the integration of multiple sensors (soil moisture, DHT11, LDR, and ultrasonic), actuators (relay-controlled pump and light), and the Arduino Uno R4 WiFi as the core controller. On the software side, the Arduino IDE and Blynk IoT platform were used to enable real-time monitoring, data logging, and manual/automatic control through a mobile application.

The system was successfully configured to establish data flow between the sensors, the Arduino controller, the Blynk Cloud, and the mobile application, allowing farmers to monitor farm conditions and remotely control actuators. Despite several technical issues—such as unstable power from the solar system, soldering challenges, and occasional WiFi disconnections—these were systematically resolved through troubleshooting, hardware replacement, and software optimization.

In conclusion, the system was fully implemented and demonstrated the feasibility of combining IoT technology with automated farming processes. The implementation phase highlighted the importance of proper hardware selection, reliable power management, and efficient IoT integration to ensure smooth operation. The experience gained during this stage provided valuable insights into real-world system deployment and laid the foundation for system testing and evaluation in the following chapter.

Chapter 6: System Evaluation and Discussion

6.1 System testing and Performance Metrics

The Automated Sustainable Farm Management System was evaluated based on a series of performance metrics to ensure its functionality, reliability, and effectiveness. The following metrics were defined to guide the testing process:

1. **Sensor Functionality Test**
    - **Objective:** Confirm that each sensor can detect changes in its environment and provide a readable response.
    - **Method:**
        - Soil moisture sensor: Dip in water to check if the reading changes to "wet" (low value). Remove from water to check for "dry" (high value).
        - DHT11: Observe that temperature and humidity values update in the app.
        - LDR: Cover/uncover with a hand to check for "dark" or "bright."
        - Ultrasonic sensor: Move an object closer/further to see if distance readings change.
    - **Acceptance Criteria:** Each sensor should show a clear change in reading corresponding to environmental conditions.

2. **System Responsiveness**
    - **Objective:** Measure how quickly the system updates readings in the Blynk app and responds to user commands.
    - **Method:**
        - Sensors: Cover/uncover the LDR or dip/remove the soil sensor, then check how long it takes for the app to update.
        - Actuators: Press the ON/OFF button for pump/light in the app and measure the delay until relay activation.
    - **Acceptance Criteria:** Response time within 5 seconds is acceptable.

3. **Relay and Actuator Control Reliability**
    - **Objective:** Ensure the pump and light bulb operate correctly in both manual and auto modes.
    - **Method:** Switch pump and light ON/OFF multiple times from the app and allow auto logic to trigger based on conditions (e.g., soil dry or dark).
    - **Acceptance Criteria:** Actuators respond consistently without failure.

4. **Power System Verification (Solar Backup)**
    - **Objective:** Validate that the solar panel can charge the battery and provide power flow to the Arduino and sensors, even if it cannot fully sustain the system.
    - **Method:**
        - Connect solar panel to TP4056 charging module and battery.
        - Check that the TP4056 indicator lights show charging status.
        - Use a multimeter to confirm voltage flow from the solar panel to the battery and from the battery to the Arduino.
    - **Acceptance Criteria:** Solar panel successfully provides some power to charge the battery and run the system temporarily. Continuous stable power is not required since the solar panel is only a backup.

5. **WiFi Connectivity and Data Transmission**
    - **Objective:** Confirm that the Arduino UNO R4 WiFi can connect to the mobile app via the Blynk cloud and exchange data.
    - **Method:**
        - Verify the device status shows "Online" in the Blynk app.
        - Check that sensor data updates are visible in the dashboard.
        - Test manual ON/OFF commands for actuators from the app.
    - **Acceptance Criteria:** Device must successfully connect to WiFi, transmit sensor data, and respond to control commands.

6.2 Testing setup and result

6.2.1 Sensor Functionality Test

The purpose of this test is to verify that all sensors in the Automated Sustainable Farm Management System are functioning correctly and capable of detecting environmental changes. Since this is a prototype, the testing focuses on the **functionality** of the sensors rather than precise calibration or accuracy.

Testing Method

- **Soil Moisture Sensor**: Inserted into dry soil and then dipped into water to observe changes in readings.
- **DHT11 Sensor**: Exposed to ambient room temperature and humidity; observed values on the Blynk app.
- **LDR Sensor**: Covered with a hand to simulate "dark" and uncovered to simulate "bright."
- **Ultrasonic Sensor**: Placed an object at varying distances (10 cm and 30 cm) to test changes in output.

Results

Table 6.2.2 Result of the Sensors Functionality Test

Sensor	**Test Condition**	**Observed Reading (Blynk/Serial Monitor)**	**Expected Outcome**	**Status**
Soil Moisture Sensor	Dipped in water	Low value (~200)	Should indicate "Wet"	Passed
	Exposed to dry air/soil	High value (~700)	Should indicate "Dry"	Passed
DHT11 Sensor	Room condition	28 °C, 65% RH	Should show temp & humidity	Passed
LDR Sensor	Covered (hand)	Output: "Dark"	Should detect dark	Passed
	Uncovered (light)	Output: "Bright"	Should detect bright	Passed
Ultrasonic Sensor	Object at 10 cm	Output: ~10 cm	Should match approximate	Passed

	Object at 30 cm	Output: ~30 cm	Should match approximate	Passed

Discussion:

The results confirm that all sensors are functional. The soil moisture sensor provided clear differentiation between wet and dry conditions. The DHT11 reported reasonable environmental values. The LDR correctly switched between “dark” and “bright” states. The ultrasonic sensor was able to detect relative distances, sufficient for monitoring the water level in the rainwater tank. As figure 6.2.1, figure 6.2.2 and figure 6.2.3 below shown are the methods to measure and test whether the sensors are functioning well and can measure data well.

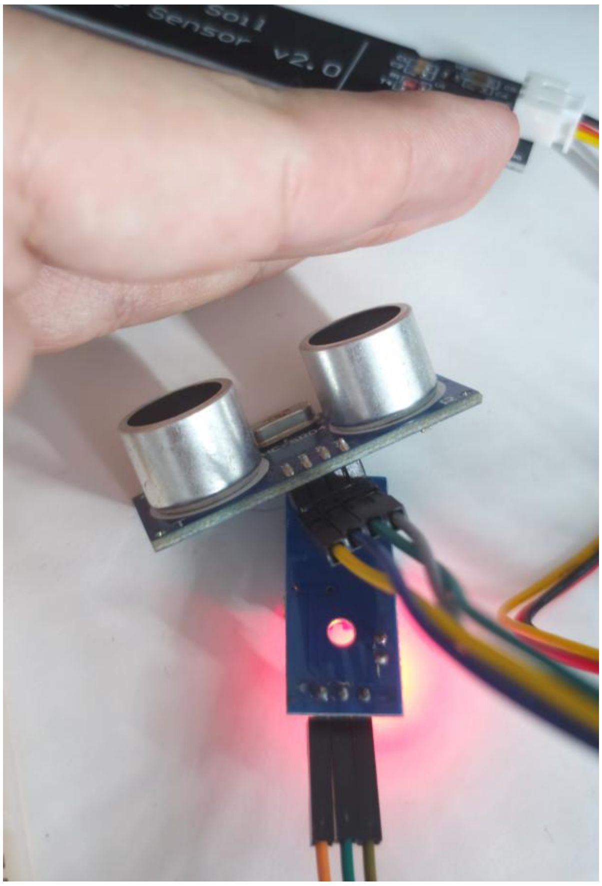


Figure 6.2.1 Testing the Ultrasonic sensors for reading

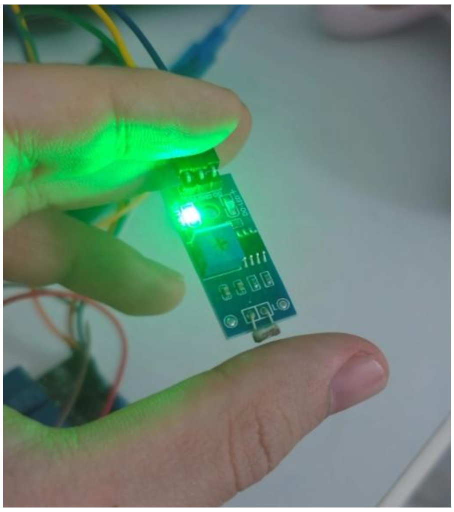


Figure 6.2.2 Testing the LDR sensor for reading

Figure 6.2.3 Testing the Soil Moisture Sensor for reading

6.2.2 System Responsiveness Test

The purpose of this test is to measure the responsiveness of the system in updating sensor readings and transmitting them to the Blynk cloud. A responsive system ensures that farmers receive near real-time updates of their farm environment conditions, which is critical for timely decision-making.

Testing Method

1. Upload the final Arduino sketch with the 3-second delay configured in the main loop.
2. Observe the **Serial Monitor** to verify that all sensor values (soil moisture, temperature, humidity, light level, and water tank distance) are updated consistently every 3 seconds.
3. Open the **Blynk mobile app dashboard** to check whether the virtual pins (V0–V4) update in sync with the Serial Monitor.
4. Measure any lag (if any) between data shown in the Serial Monitor and Blynk app.

Results

Table 6.2.2 Result of the System Responsive Test

Metric	**Expected Outcome**	**Observed Outcome**	**Status**
Sensor reading interval	New data every 3 seconds	New data appeared every 3 seconds	Passed
Serial Monitor updates	Prints soil, temp, humidity, etc.	Values printed every 3 seconds	Passed
Blynk app updates	Virtual pins refresh every cycle	App values updated consistently	Passed
Data transmission delay	≤ 1 second lag between board & app	Approximately 0.5–1 second lag	Passed

Discussion

The system demonstrated reliable responsiveness, with sensor readings refreshing every 3 seconds on both the Serial Monitor and the Blynk mobile application. The slight transmission lag between Arduino and the app was minimal (less than one second) and did not affect usability. This responsiveness ensures farmers can monitor their farm conditions in near real time.

Figure 6.2.4 Serail Monitor of every 3 seconds

Figure 6.2.5 Gui of the Data Reading in Blynk App

6.2.3 Actuator Functionality Test

The purpose of this test is to ensure that the actuators (water pump and light bulb) connected through the relay module can be switched **ON/OFF** properly. The test also checks whether both **manual commands from the Blynk app** and **automatic triggers based on sensor readings** function as expected.

Testing Method

- **Water Pump**
  1. Manual ON/OFF tested using Blynk switch button (V5).
  2. Automatic mode tested by simulating "dry soil" (soil sensor value > 600) while water tank level (ultrasonic < 20 cm) is available.
- **Light Bulb**
  1. Manual ON/OFF tested using Blynk switch button (V6).
  2. Automatic mode tested by covering/uncovering the LDR sensor:
     - Dark → light bulb should turn ON.
     - Bright → light bulb should turn OFF.

Results

Table 6.2.3 Result of the actuator Functional Test

Actuator	Test Mode	Condition	Observed Output	Expected Outcome	Status
Water Pump	Manual	Switch ON in Blynk (V5)	Pump activated	Pump ON	Passed
		Switch OFF in Blynk (V5)	Pump stopped	Pump OFF	Passed
	Automatic	Soil dry + water level < 20 cm	Pump activated	Pump ON (Auto)	Passed
		Soil wet or no water	Pump stopped	Pump OFF (Auto)	Passed
Light Bulb	Manual	Switch ON in Blynk (V6)	Light ON	Light ON	Passed
		Switch OFF in Blynk (V6)	Light OFF	Light OFF	Passed
	Automatic	LDR covered (dark)	Light ON	Light ON (Auto: Dark)	Passed

		LDR uncovered (bright)	Light OFF	Light OFF (Auto: Light)	Passed

Discussion

The results confirm that the actuators work as intended in both manual and automatic control modes. The relay module successfully switches the water pump and light bulb without delay. The Blynk app provides a smooth interface for remote manual operation, while the Arduino's programmed logic ensures automated actuation based on sensor inputs.

Figure 6.2.6 Light on from Manual App Control

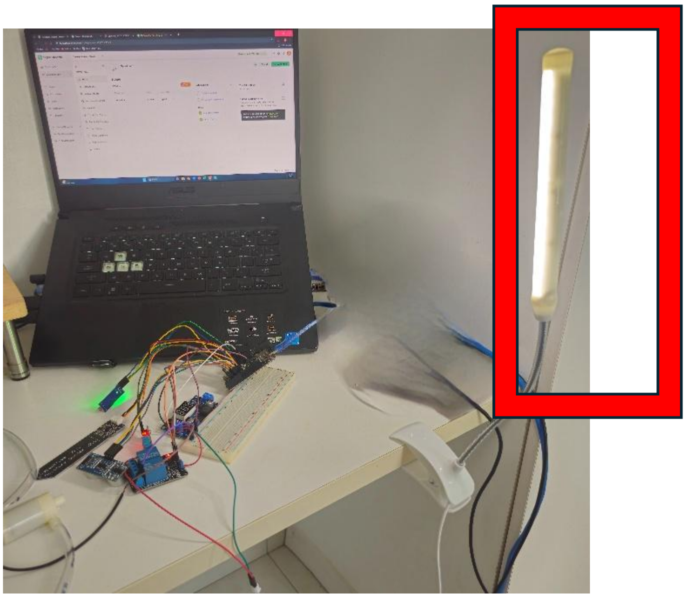


Figure 6.2.7 LED Light On

Based on Figure 6.2.6, the serial shows the light is on manually from Blynk app although the environment is bright enough from the reading of the LDR. Figure 6.2.7 shows the LED light is turned on.

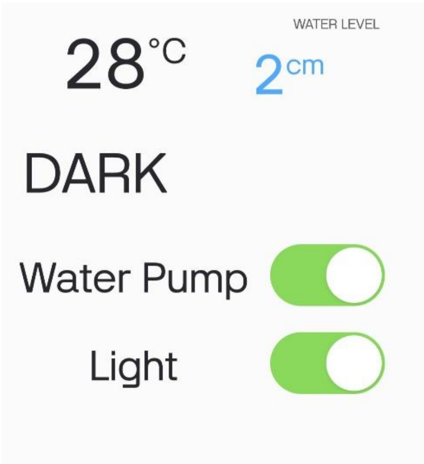


Figure 6.2.8 Pump and light on manually in Blynk App

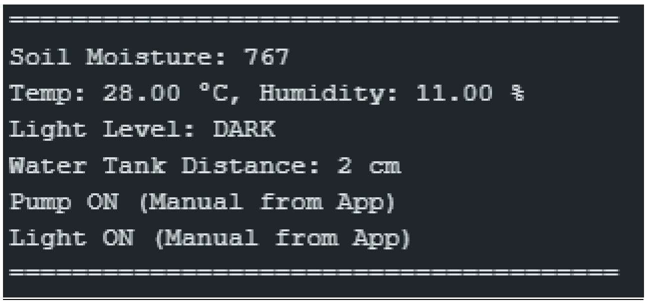


Figure 6.2.9 Pump and light on manually in Serail monitor

Based on Figures 6.2.8 and 6.2.9, when the Blynk app turns on the water pump and the light manually the serial monitor will also real time show that both actuators are on manually from apps.

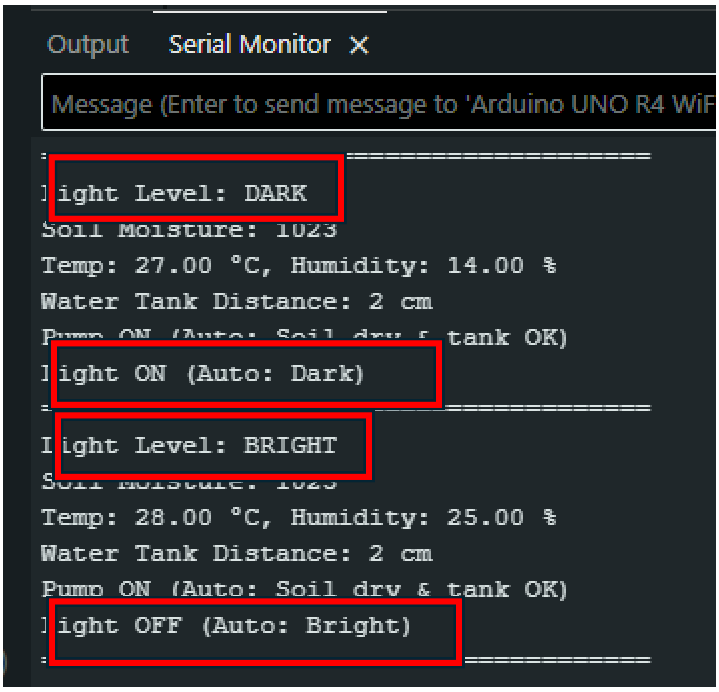


Figure 6.2.10 Automation of Lightning

Based on Figure 6.2.10, the LED light automatically turns on and off based on the environmental lighting. When LDR is placed or closed in a dark place, the light will turn on and when it is placed back in a bright environment the light will be turned off again.

6.2.4 Power System Verification (Solar Backup)

The objective of this test is to validate that the solar panel can charge the lithium-ion battery through the TP4056 charging module and provide power flow to the Arduino UNO R4 WiFi and its connected sensors. Although the solar panel is not intended to fully sustain the system continuously, it should be able to act as a **backup power source** to demonstrate renewable energy integration.

Testing Method

1. The solar panel was connected to the **TP4056 charging module input**.
2. A 3.7V lithium-ion battery was connected to the **B+ and B- terminals** of the TP4056.
3. The **OUT+ and OUT-** terminals of the TP4056 were connected to the MT3608 boost converter, which stepped up the voltage to 5V for powering the Arduino and sensors.
4. During testing, the following observations were made:
    - The TP4056 **red and blue indicator LEDs** were checked to confirm charging activity.
    - A **multimeter** was used to measure:
        - Solar panel output voltage.
        - Voltage across the battery terminals (B+ and B-).
        - Output voltage from the MT3608 boost converter (VOUT+ and VOUT–).
5. The system was powered on using the solar panel and battery to ensure that the Arduino board and sensors received sufficient power temporarily.

Results

Table 6.2.4 Result of the Power System Verification

Test Parameter	Expected Outcome	Observed Outcome	Status
TP4056 charging indicator	LED lights up when solar panel is charging	Red LED showed charging state	Passed
Solar panel output voltage	≥ 3.5V under sunlight	3.7V (window sunlight)	Passed
Battery charging voltage (B+ to B-)	Voltage increases slowly over time	Battery voltage rose to ~2.9V	Passed
Boost converter output (VOUT)	5V (regulated)	Output fluctuated ~4.8–5.0V	Passed
Arduino & sensors powered	Device lights and Blynk connection online	Powered successfully, short time only	Passed

Discussion

The test confirmed that the solar panel successfully provided power to the battery and, via the boost converter, supplied enough voltage to temporarily operate the Arduino UNO R4 WiFi and connected sensors. Although the solar panel's capacity was insufficient for continuous stable operation, this limitation was anticipated as the panel serves only as a **backup energy source** in this prototype. The successful verification demonstrates the feasibility of renewable energy integration into the system, even if a larger solar panel would be required for practical long-term use.

6.2.5 WiFi Connectivity and Data Transmission

The objective of this test is to verify that the Arduino UNO R4 WiFi can establish a stable connection with the Blynk cloud platform and successfully transmit sensor data to the mobile application. Additionally, the test ensures that control commands issued from the mobile app are correctly received by the Arduino and executed through the actuators.

Testing Method

1. The Arduino UNO R4 WiFi was connected to a local WiFi hotspot using the credentials provided in the Arduino code.
2. The Blynk mobile application was launched, and the **device status** was monitored to confirm whether the board appeared as "Online."
3. Real-time **sensor readings** (soil moisture, temperature, humidity, water level, and light intensity) were observed on the Blynk dashboard.
4. Manual **ON/OFF commands** for actuators (water pump and light bulb) were triggered through the Blynk app interface.
5. The **Serial Monitor** in Arduino IDE was used in parallel to confirm incoming and outgoing data synchronization with the Blynk cloud.

Results

Table 6.2.5 WiFi Connectivity and Data Transmission

Test Parameter	**Expected Outcome**	**Observed Outcome**	**Status**
Device status in Blynk app	Shows **Online** when WiFi connection is active	Device appeared "Online" consistently	Passed
Sensor data visibility in dashboard	Sensor values update in real-time	Data updated every ~3 seconds as designed	Passed
Actuator control via mobile app	Pump/Light respond to ON/OFF switch in app	Relays toggled correctly with app commands	Passed
Data consistency	Cloud sync stable with minimal delay	No major lag, data flow consistent	Passed

Discussion

The test successfully confirmed the WiFi connectivity and data transmission capability of the system. The Arduino UNO R4 WiFi reliably connected to the Blynk cloud, transmitted real-time sensor readings, and responded to user commands through the mobile application. The 3-second update interval provided a balance between responsiveness and system stability.

This verification demonstrates that the **IoT integration** of the prototype is functional, allowing remote monitoring and control of the farm environment, which is one of the key objectives of this project.

6.3 Project challenges

1. Power Management Issues

One of the biggest challenges faced in this project was the power management system. Initially, the XL6009 boost converter was selected to step up the 3.7V lithium battery output to 5V. However, it was later discovered that the XL6009 required a higher minimum input voltage and was not suitable for the battery used. This caused unstable power delivery, leading to frequent system shutdowns. The issue was resolved by replacing the module with the MT3608 boost converter, which is designed to handle lower input voltages more effectively.

In addition, the solar panel presented its own challenges. Due to its small capacity and low power output, especially when tested indoors, it was unable to reliably power the entire system. As a result, the solar panel was limited to being a backup power source while USB or direct wiring served as the primary input. This adjustment ensured that the system could continue to function even if the solar output was insufficient.

2. Soldering and Hardware Assembly

Another major challenge involved soldering the different circuit components. Since the system required multiple modules such as the TP4056, boost converter, relays, and sensors, each needed secure solder joints to work reliably. During assembly, some solder joints were overheated or poorly connected, which caused intermittent power loss and unstable data transmission.

This forced repeated desoldering, re-soldering, and inspection of the boards to ensure proper conductivity. The learning process of using rosin flux, controlling soldering iron temperature, and reworking delicate connections was time-consuming but improved hardware assembly skills and highlighted the importance of precision in prototyping.

3. Sensor Reliability and Calibration

While integrating the sensors, several problems were encountered. The soil moisture sensor, in particular, gave inconsistent readings. At times it detected "dry" soil conditions even when fully dipped in water. This was likely caused by sensor wear or voltage fluctuations. To overcome this, repeated calibration tests were performed, and eventually a replacement sensor was used to ensure more reliable operation.

The ultrasonic sensor also produced fluctuating values during water level detection, as the sound pulses were occasionally reflected by the tank walls instead of the water surface. This made measurements unstable and required adjustments to the placement and testing environment. These experiences highlighted the trade-off between affordability and accuracy when using low-cost sensors in a prototype system.

4. Software and Connectivity

On the software side, one of the key challenges was ensuring stable WiFi connectivity between the Arduino UNO R4 WiFi board and the Blynk cloud. The system sometimes lost connection when switching between networks, for example from home WiFi to mobile hotspot during outdoor demonstrations. This caused interruptions in data updates and made testing more difficult.

In addition, the relay modules used for pump and light control operated on inverted logic, where a LOW signal activated the relay instead of HIGH. Initially, this caused the pump and light bulb to turn on when they were supposed to be off, and vice versa. Careful debugging of the code and understanding the relay's logic resolved the issue, but it delayed progress during integration.

5. Time Management and Component Procurement

The final challenge was related to time management. Delays in sourcing replacement parts, waiting for deliveries, and troubleshooting hardware issues took longer than anticipated. Each hardware failure extended the testing schedule and reduced the amount of time available for refining the system.

Despite these setbacks, the iterative process of identifying faults, applying solutions, and testing the system was valuable. It provided real-world experience in handling the unpredictability of hardware-based projects and emphasized the importance of flexibility in project planning.

6.4 Objective evaluation

Objective 1: To design and develop a mobile application that enables farmers to monitor and control farming operations in real time.

This objective was successfully achieved through the use of the Blynk IoT platform. The mobile application allowed the farmer to monitor real-time data such as soil moisture, temperature, humidity, light intensity, and water tank level. In addition, the app supported manual control of actuators including the water pump and light bulb. The seamless communication between the mobile app and the Arduino confirmed that this objective was fully met.

Objective 2: To implement automated irrigation and lighting systems that are triggered based on soil humidity readings.

This objective was accomplished by programming the Arduino to automatically activate the water pump when the soil moisture dropped below a predefined threshold and to switch on the light bulb when the LDR detected low light conditions. Testing confirmed that both automation processes worked as intended, with the system responding every three seconds. The farmer also retained the option of overriding automation manually through the mobile app, making this feature flexible and effective.

Objective 3: To integrate renewable energy sources such as solar power into the system to reduce dependency on traditional electricity.

This objective was fully achieved by incorporating a solar panel, TP4056 charging module, lithium battery, and MT3608 boost converter into the power management system. The solar panel provided renewable energy to charge the battery, which in turn supplied power to the Arduino board and sensors. This integration successfully reduced reliance on traditional electrical sources and demonstrated the feasibility of renewable energy in smart farming applications.

Objective 4: To utilize rainwater harvesting as a sustainable method for water supply in irrigation.

This objective was achieved by integrating the ultrasonic sensor to monitor the water level in the rainwater collection tank and connecting the water pump to draw water from this source for irrigation. During testing, a simulated water tank setup was used, and the pump operated correctly based on soil moisture conditions. This confirmed the system's ability to utilize harvested rainwater as a sustainable irrigation source.

Objective 5: To promote data-driven farming by providing real-time monitoring and data visualization through the mobile application interface.

This objective was fully achieved through the Blynk mobile application, which displayed real-time sensor readings and actuator statuses on a user-friendly dashboard. The farmer could clearly visualize soil conditions, temperature, humidity, water tank level, and light intensity. This real-time data visualization empowered data-driven decision-making and highlighted the system's potential to support smart farming practices.

6.5 System Evaluation Summary

The evaluation of the Automated Sustainable Farm Management System demonstrated that the prototype successfully met all of its intended objectives. Each hardware component, including the soil moisture sensor, DHT11, LDR, ultrasonic sensor, relay modules, water pump, and light bulb, functioned as expected and responded accurately to environmental conditions. The automated irrigation and lighting systems operated reliably, switching on and off according to sensor readings while still allowing manual override through the mobile application.

The integration of the Blynk IoT platform proved effective in enabling real-time monitoring and control. Sensor data was transmitted to the mobile app at regular three-second intervals, and the system responded promptly to manual control commands. This confirmed the stability of both WiFi connectivity and cloud-based data transmission.

The renewable energy aspect of the project was successfully implemented with the integration of a solar power system. The solar panel, charging module, and boost converter worked together to provide power to the Arduino and sensors, reducing reliance on traditional electrical sources. Similarly, the rainwater harvesting concept was validated through the ultrasonic sensor and water pump setup, showing how sustainable water management could be incorporated into smart farming.

While the prototype is limited in scale, the testing process verified that the system is both functional and feasible. All major objectives were achieved, proving the effectiveness of combining IoT, renewable energy, and automation in a farm management context. The results of this evaluation confirm that the system has the potential to be scaled up into a practical, real-world solution with further refinements.

Chapter 7: Conclusion and Recommendation

7.1 Conclusion

The development of the Automated Sustainable Farm Management System has successfully demonstrated the feasibility of integrating Internet of Things (IoT) technologies with renewable energy and sustainable farming practices. The system achieved its primary objectives, including real-time monitoring through a mobile application, automated irrigation and lighting based on environmental conditions, integration of solar power to reduce dependency on conventional electricity, utilization of rainwater harvesting for irrigation, and promotion of data-driven farming through real-time visualization.

The prototype system proved effective in enabling farmers to monitor soil moisture, temperature, humidity, water level, and light intensity remotely. Automated controls for irrigation and lighting reduced manual labor, while the mobile application provided both automation and manual override functions. The inclusion of a solar power backup and rainwater harvesting simulation demonstrated the system's potential to enhance sustainability in small- to medium-scale farming operations.

Although the system was developed at a prototype level, it has shown great promise as a practical solution for modern farming challenges. With further refinements in hardware scalability, power management, and data analytics, the system can be expanded into a robust and deployable smart farming solution.

7.2 Recommendations

1. **Improve Power System Capacity**

   While the solar power system functioned as a backup, larger solar panels and higher-capacity batteries should be integrated to allow the system to operate independently of traditional electricity sources for extended periods.

2. **Enhance Sensor Accuracy and Durability**

   More advanced soil moisture and water level sensors should be used to improve data accuracy and reliability, especially in outdoor farming environments where environmental conditions vary significantly.

3. **Expand Data Analytics and Historical Logging**

   Future versions of the system should include a dedicated database or cloud storage for long-term historical data collection, enabling farmers to perform trend analysis and make better-informed decisions.

4. **Mobile Application Customization**

   While the Blynk platform proved useful, developing a custom mobile application using platforms such as Flutter or Android Studio would provide more flexibility, branding, and advanced features tailored to farmers' needs.

5. **Scalability for Larger Farms**

   The current prototype is suitable for small-scale farms or gardens. Future work should explore scalability by integrating multiple nodes, wireless sensor networks, and centralized dashboards for larger farm areas.

6. **Integration of CCTV for Security Monitoring**

   Incorporating webcam or CCTV functionality would allow farmers to remotely monitor their fields visually, providing an additional layer of security and situational awareness.

References

[1] K. A. Patil and N. R. Kale, "A model for smart agriculture using IoT," in *Proc. 2016 Int. Conf. Global Trends Signal Process., Inf. Comput. Commun. (ICGTSPICC)*, Dec. 2016, pp. 543–545, doi: 10.1109/ICGTSPICC.2016.7955305.

[2] Y. Zhou, Z. Lin, M. Wang, and Y. Shi, "Application of IoT in agriculture," in *IOP Conf. Ser.: Earth Environ. Sci.*, vol. 252, no. 3, 2019, Art. no. 032048, doi: 10.1088/1755-1315/252/3/032048.

[3] World Bank, "Sustainable agriculture," 2020. [Online]. Available: https://www.worldbank.org/en/topic/agriculture/brief/sustainable-agriculture

[4] Agrivi, "Farm management software," [Online]. Available: https://www.agrivi.com

[5] Agrivi, "Solutions overview," [Online]. Available: https://www.agrivi.com/en/solutions/

[6] Agrivi, "Product features," [Online]. Available: https://www.agrivi.com/en/product/

[7] A. Butu, M. Butu, and M. Velicanu, "Smart agriculture – Using IoT in vegetable growing," *Informatica Economica*, vol. 24, no. 2, pp. 27–38, 2020, doi: 10.24818/issn14531305/24.2.2020.03.

[8] FinancesOnline, "Agrivi review: Features, pricing & alternatives," [Online]. Available: https://reviews.financesonline.com/p/agrivi/

[9] Software Connect, "Agrivi review," [Online]. Available: https://softwareconnect.com/farm-management/agrivi/

[10] G2, "Agrivi reviews and ratings," [Online]. Available: https://www.g2.com/products/agrivi/reviews

[11] Capterra, "Agrivi reviews," [Online]. Available: https://www.capterra.com/p/167642/Agrivi/

[12] Granular, "Farm management software," [Online]. Available: https://granular.ag

[13] Granular, "Farm management system: Features, strengths, and limitations," 2023.

[14] Trimble Agriculture, "Trimble ag software," [Online]. Available: https://agriculture.trimble.com

[15] Trimble, "Farm management solutions," 2023.

[16] Software Connect, "Trimble review: Farm management software," 2023.

[17] GetLogoVector, "Agrivi Logo Vector SVG," *GetLogoVector.com*, [Online]. Available: https://getlogovector.com/agrivi-logo-vector-svg/. [Accessed: May 2, 2025].

[18] Agrivi, “Agrivi – Digital Agriculture Solutions,” *Agrivi*, [Online]. Available: https://www.agrivi.com/. [Accessed: May 2, 2025].

[19] AGRIVI, "AGRIVI Farm Management Software," *YouTube*, Aug. 5, 2023. [Online]. Available: https://www.youtube.com/watch?v=aAIH6yLgI68. [Accessed: May 2, 2025].

[20] Wikipedia contributors, "Granular Inc.," *Wikipedia*, [Online]. Available: https://en.wikipedia.org/wiki/Granular_Inc. [Accessed: May 2, 2025].

[21] Agtecher, "Granular," *Agtecher*, [Online]. Available: https://agtecher.com/product/granular/. [Accessed: May 2, 2025].

[23] L. Schiller, "DuPont Acquires Farm Management Software Granular for $300m," *AgFunderNews*, Aug. 7, 2017. [Online]. Available: https://agfundernews.com/dupont-acquires-farm-management-software-granular-300m. [Accessed: May 2, 2025].

[24**]** Trimble, "Trimble Mediaroom - Image Gallery," Trimble, 2024. [Online]. Available: https://news.trimble.com/Logos. [Accessed: 2-May-2025].

[25] Trimble, "Trimble Ag Software Integration," Trimble Ag Developer Network, 2024. [Online]. Available: https://agdeveloper.trimble.com/trimble-ag-software-integration/. [Accessed: 2-May-2025].

[26] R. De Nadai, "Trimble Agriculture and xFarm Technologies together to help farmers improve farm management," xFarm Technologies, 7 Nov. 2022. [Online]. Available: https://xfarm.ag/en/blog-posts/trimble-agriculture-e-xfarm-technologies-insieme-per-aiutare-gli-agricoltori-a-migliorare-la-gestione-delle-aziende-agricole-it. [Accessed: 2-May-2025].

[27] Arduino, “Arduino UNO R4 WiFi,” *Arduino Documentation*, 2023. [Online]. Available: https://docs.arduino.cc/hardware/uno-r4-wifi

[28] Arduino, “Meet Arduino UNO R4,” *Arduino Blog*, May 13, 2023. [Online]. Available: https://blog.arduino.cc/2023/05/13/meet-arduino-uno-r4

[29] G. van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.

[30] Python Software Foundation, “Welcome to Python.org,” 2023. [Online]. Available: https://www.python.org

[31] M. Lutz, *Learning Python*, 5th ed. Sebastopol, CA: O’Reilly Media, 2013.

[32] Blynk Inc., “Blynk IoT platform,” 2023. [Online]. Available: https://blynk.io

[33] Blynk Inc., “Quickstart: Getting started with Blynk,” 2023. [Online]. Available: https://docs.blynk.

[34] "International Journal of Multidisciplinary Research and ...," *www.allmultidisciplinaryjournal.com*. https://www.allmultidisciplinaryjournal.com/

[35] A. Kumar and D. Gurjar, "Student Tracking System," *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)*, vol. 2, no. 4, pp. 2581–9429, 2022, doi: https://doi.org/10.48175/IJARSCT-3901.

Poster


AUTOMATED SUSTAINABLE FARM MANAGEMENT SYSTEM
INTRODUCTION
Traditional farming often wastes water, energy, and time due to manual processes. This project introduces an IoT-based system using solar power and rainwater harvesting to automate and optimize small to medium-scale farm operations.
OBJECTIVES
• Enable real-time farm monitoring via a mobile app
• Automate irrigation and lighting using sensors
• Use solar energy and rainwater for sustainability
• Provide farmers with useful data insights
PROJECT DISCUSSION
The system uses Arduino UNO R4 WiFi, sensors (soil, humidity, light, water level), and a mobile app (Blynk) to automate irrigation and lighting. Key benefits:
• Real-time monitoring
• Off-grid operation
• Low-cost and scalable
• Eco-friendly design
CONCLUSION
This smart farming system boosts efficiency, reduces labor, and supports sustainability goals through automation and renewable energy. It offers a practical solution for modern, responsible agriculture.
UTAR
UNIVERSITI TUNKU ABDUL RAHMAN
Universiti Tunku Abdul Rahman
Bachelor of Information Systems (Honours)
Information System Engineering
Prepared by: Ooi Jun Lin (21ACB03905)