Innovations and
Entrepreneurship Unit

Green Mile

Sustainable Smart Agriculture System

2024

Innovations and Entrepreneurship Unit

Ministry of higher Education high institute of engineering &Technology in New Damietta

Green Mile

Sustainable Smart Agricultural System

A project report submitted for
Innovations and Entrepreneurship Unit (IEU)

Submitted By:

1. Ahmed Amr Eladham
2. Fatma Alzahraa Ayman Moawed
3. Mohamed Mohamed Elsalkh
4. Sara Mohamed Khalil
5. Mariam Mohamed Elseady
6. Esraa Elsaeid Abo Shahen
7. Zeyad Ibrahim Abo Elmaaty
8. Mohamed Ashraf Elbardwill
9. Zeyad Waleed Sherra
10. Seif Mamdouh Fawzy
11. Amr Ibrahim Elshenawy
12. Mohamed Eldosoky Fahmy
13. Mohamad Ayman Moawed
14. Mohamed Elsayed Elzyny
15. Nour Eldein Kamel Hassanein
16. Khaled Ayman Elmahdy
17. Mohamed Essam Rehema

Acknowledgment

In completing this project, we have been fortunate enough to have help, support, and encouragement from many people. We would like to acknowledge them for their cooperation. Firstly, we would like to thank the **Innovations and Entrepreneurship Unit (IEU)** and its manager **Dr. Amira Elsonbaty** for giving us an opportunity to make this project. We would also like to acknowledge **Eng. Esraa Hany** for guiding us through each and every step of the process with knowledge and support. Her thoughts have been a constant source of inspiration for us. Additionally, we would like to extend our gratitude to **Eng. Nour-Elden Nabil** for his invaluable assistance with the project. His guidance and support were greatly appreciated and significantly contributed to the project's success. Furthermore, we would like to say a big thank you, especially to **Eng. Ahmed Hamed Hamed**, for his energy, understanding, and help throughout our project, and for providing us with his knowledge and tools for the project. Great thanks also to our college mate, **Haneen Osama Nada**, for helping us with the project. Finally, we would like to thank all members of the department for their kind assistance, suggestions, and cooperation throughout the development of the project.

Abstract

The purpose of Green Mile is to make the world greener again and help automate farming and agriculture as a whole by using various systems. These include a smart irrigation system that uses a soil moisture sensor to determine the hydration level of the soil and then sends a signal to the water pump to hydrate the plants. Additionally, a fire detection system uses a DHT22 temperature and humidity sensor to detect fires and send notification alerts to provide early warnings, reducing the risk of wildfires and protecting the area. Finally, we have a movement security system that scares off rodents and wild animals from eating or harming the field. All of the data from our systems will be sent to our Green Mile mobile application. The results of our research show that farms use more water than they need to grow crops. Around 10% of all fires globally occur on agricultural land, causing damage to crops, infrastructure, and nearby native vegetation. On a global scale, it is estimated that wildlife damages reduce crop yields by approximately 5-10% annually. With our project, we will ensure that none of these problems occur again. The Green Mile project thus serves as a model for helping farmers relax and not worry about problems that occur in the field. [1] [2] [3]

Table of Contents

List of Figures

List of Tables

List of Abbreviations

FAO	Food and Agriculture Organization
IUCN	The International Union for Conservation of Nature
AFBF	American Farm Bureau Federation
IOT	Internet of Things
AI	Artificial Intelligence
LDR	Light Detecting Resistor
DHT22	Digital Humidity and Temperature Sensor
HC-SR501	A passive infrared (PIR) motion sensor
PIR	Passive Infrared Sensor
IPO	Input, Process and Output
ESP32	Espressif Systems Platform 32-bit
LED	Light Emitting Diode
SG90	Servo Motor
PS	Protection System
RUS	Roll-Up Sides
VCC	Voltage Common Collector
GND	Ground
MT3608	Boost Converter
APLS	Application Programming Interfaces
SCADA	Supervisory Control and Data Acquisition
SPEI	Standardized Precipitation Evapotranspiration Index
ADC	Analog to Digital Conversion
GPIO	General Purpose Input/Output
HIGH	Above the Average in Height
HVAC	Heating, Ventilation and Air Conditioning
CO2	Carbon Dioxide
SOC	System on Chip
TSMC	Taiwan Semiconductor Manufacturing Company
RISC-V	Reduced Instruction Set Computer - V
CPU	Central Processing Unit
ARM	Advanced RISC Machine
SIP	System in Package
KB	Kilobyte
SRAM	Static Random Access Memory
ROM	Read-Only Memory
RTC	Real-Time Clock
BLE	Bluetooth Low Energy
MBPS	Megabits Per Second

SAR	Specific Absorption Rate
GPIOS	General-Purpose Input/Output
DAC	Digital-to-Analog Converter
MAC	Media Access Control
LAN	Local Area Network
PHY	Physical Layer
SD	Secure Digital
SDIO	Secure Digital Input Output
PWM	Pulse Width Modulation
AES	Advanced Encryption Standard
RSA	Rivest-Shamir-Adleman
ECC	Elliptic Curve Cryptography
RNG	Random Number Generator
ULP	Ultra Low Power
DMA	Direct Memory Access
ADC	Analog to Digital Converter
UART	Universal Asynchronous Receiver-Transmitter
I2C	Inter-Integrated Circuit
SPI	Serial Peripheral Interface
I2S	Inter-IC Sound
RMII	Reduced Media Independent Interface
WPA	Wi-Fi Protected Access
WAPI	WLAN Authentication and Privacy Infrastructure
WLAN	Wireless Local Area Network
IDE	Integrated Development Environment
USB	Universal Serial Bus
HVAC	Heating, Ventilation, and Air Conditioning
CDS	Cadmium Sulfide
SDK	Software Development Kit
UI	User Interface
UX	User experience
BaaS	Backend-as-a-Service
NoSQL	Not Only SQL
SQL	Structured Query Language
ARM	Advanced RISC Machine
API	Application Programming Interface
SFM-27	Type of Electronic Alarm Buzzer
CO2	Carbon Dioxide
SO2	Sulfur Dioxide
NOx	Nitrogen Oxid

Appendices

Chapter 1

Introduction and Objectives

- Introduction
- Motivation
- Problem Statement
- Project objectives
- Background

Chapter 1 - Introduction and Objectives

1.1 Introduction

In today's increasingly interconnected world, integrating technology into our daily tasks is not just a convenience but a necessity to enhance efficiency and sustainability. Our project is designed to harness the power of advanced automation to improve six critical zones: Smart lighting system, Smart irrigation system, fire detection system, Auto Greenhouse Roll-Up Sides System, Smart protection system and Monitoring system (Mobile Application).

The Smart lighting system mode ensures that the lighting and other systems are smoothly adapted to the night conditions, which reduces energy consumption and enhances comfort. Smart irrigation system uses sensors and data analytics to provide plants with the exact amount of water they need, conserve water and promote healthy plant growth. The fire detection system is equipped with an innovative feature as it detects fire in its early stages. There is also the Auto Greenhouse Roll-up sides system. for Air Circulation and Humidity Management controlled by the mobile application. The Smart protection system is the defense line against birds and rodents taking care of scaring the intruders as soon as it senses them. All of these innovations are driven by Monitoring system (Mobile Application) that allows smooth control and real-time monitoring of these systems. This application ensures that users receive quick notifications and can make the necessary adjustments based on changing conditions, whether it is a change in weather, fire alert, or early night.

By combining advanced technology and practical applications, our project seeks to redefine how we interact with and manage ecosystems. Our goal is to create a more sustainable future as technology not only makes our lives easier, but also helps us take care of our environment. Through these developments, we aim to demonstrate the potential of technology to transform everyday tasks and contribute to a more efficient and sustainable world.

1.2 Motivation

The Food and Agriculture Organization (FAO) declared that water scarcity is a pressing global issue that affects billions of people worldwide. It refers to the lack of sufficient available water resources to meet the demands of water usage within a region. This problem is exacerbated by various factors, including population growth, climate change, pollution, and inefficient water management practices. Water scarcity has far-reaching consequences, impacting agriculture, industry, and the health and well-being of populations. In regions experiencing severe water shortages, communities face challenges in accessing clean drinking water, adequate sanitation, and sufficient water for food production.

The International Union for Conservation of Nature and Natural Resources (IUCN) declared that rodents, including birds, rabbits, and ungulates, can cause significant crop damage globally. They affect nearly all types of crops, from cereal grains to fruits, during various stages of plant growth and year-round, including under snow cover. Rodents alter their feeding preferences throughout the plant life cycle, targeting seeds, germinating plants, mature plants, and their seeds/fruits. Some species, like pocket gophers, primarily feed underground. Damage levels can range from minor to severe, with examples worldwide illustrating losses exceeding 30%. Even commensal rodents like Rattus and Mus can cause 1-15% damage, higher on islands. In Asia alone, rodents consume enough grain annually to feed 200 million people. Rodents also damage stored crops and livestock feed, contaminating them with urine and feces. Their burrowing undermines foundations and harms equipment and passing livestock. Furthermore, they pose health risks by transmitting diseases such as plague, leptospirosis, Chagas disease, giardiasis, and hantavirus to humans and livestock.

Finally, the American Farm Bureau Federation (AFBF) has calculated crop and rangeland damage estimations for the year 2023. Total crop losses added up to nearly 22 billion dollars because of natural disasters, including wildfires.

1.3 Problem Statement

The central problems we address are:

Water Management Issues:

Water scarcity is a critical challenge for agriculture, particularly in arid regions where crops require substantial irrigation. necessitating innovative water management strategies to sustain agricultural productivity as over and under watering leads to water stress or waterlogging.

Bird and rodent damage:

Damage to crops by birds and rodents poses a significant challenge for farmers worldwide. Such damage can lead to substantial losses in crops and resources, affecting productivity and agricultural income. Strategies to combat this issue vary, while balancing the conservation of biodiversity with ensuring agricultural sustainability.

Fire damage

Wildfires represent a critical challenge to agriculture, particularly in regions prone to such events. These fires can devastate agricultural lands, destroy crops, and lead to significant economic losses. The intense heat from wildfires can damage soil structure and fertility, impairing its ability to support future crop growth. Smoke and ash generated during wildfires can also contaminate crops, affecting their quality and marketability. Mitigating the risks associated with wildfires through improved land management practices, early detection systems, and effective firefighting strategies is crucial for safeguarding agricultural productivity and resilience in fire-prone areas.

1.4 Project objectives

We are diligently working towards achieving our project objectives, which include the design and implementation of a sustainable smart agriculture system. This system aims to integrate advanced technologies such as the Internet of Things (IoT) and automation to optimize agricultural practices. Our goal is to enhance productivity, ensure resource efficiency, and promote environmental sustainability. By leveraging these innovative technologies, we hope to create a resilient agricultural framework that can adapt to changing conditions and support sustainable food production for the future.

1.5 Background

Agriculture, a sector as old as civilization itself, is now embracing the digital age. One way it's doing so is through the implementation of Smart Irrigation Systems. These systems represent a paradigm shift in how farmers and agriculturists manage water resources. Instead of using traditional ways, which often lead to overuse, wastage, and inefficiency, smart irrigation systems allow for precise control and management of water. Smart Irrigation Systems are no longer a luxury or a novelty; they are a necessity in today's agriculture landscape. With the growing global population, the demand for food is ever-increasing. To meet this demand, agriculture must become more efficient and sustainable. And this is where smart irrigation systems come into play. These systems use data, sensors, and automated controllers to optimize water usage. The result? Better crop yield, lower water consumption, and significant cost savings.

1.5.1 Novelty

The History of Irrigation – How Irrigation Evolved

It is truly an understatement when it is said that we should be grateful for the times we live in. Contemporary living has gotten busier though, as it seems no one has time to spare. Well thanks to technology, the tedious tasks of the past are becoming less time consuming, one such task is proper irrigation. Maintaining the perfect lawn has never been easier, and now with the latest in automatic sprinkler systems it's not only easy, it's efficient. It wasn't always this way, a lot of genius engineering has gone into irrigation techniques as this was tied to food production, and that was pivotal in the beginning of human civilization, and food production remains an ongoing challenge to this day.

Irrigation is believed to begin in both Egypt and a land known as Mesopotamia. During a certain period of time each year, a flooding would occur in both the Euphrates and Tigress rivers in Mesopotamia and the Nile in Egypt. The peoples who inhabited these lands have evolved from the old way of gathering food through migratory hunting and gathering, to the stationary nature of farming and agriculture. So, with this overflow of water, they would divert and direct it right to their crops. Sequentially the land along the Nile was recorded to be so rich in farmable land, that it's easy to see why ancient Egypt was able to create one of the most world-renowned ancient cultures. With these irrigation techniques, Egyptians and other civilizations had a surplus of food. This grants a civilization the ability to utilized a number of people in different ways, other than gathering

food, think of artisans, philosophers, aristocrats, inventors, manufacturers, all the various jobs required to create a sophisticated society.

Irrigation has come a long way since ancient Egypt. Most of the automated technology that is in our front lawns today started taking shape in the 1800s, technology such as residential sprinklers and the nozzle for your watering hose. With the suburban boom really taking hold in the 1950s, the property value of these suburban properties came down to noticeable factors, one being the state of the lawn. This brought demand for healthy lawns, which drove a lot of the sprinkler system technology used in agriculture to be adapted for the smaller lawn spaces. Sprinklers such as one of the most recognized residential sprinklers, the oscillating sprinkler head which had a row of nozzles along a bow like tubing that would shift back and forth.

With the automatic irrigation system, watering is done in a timely matter for efficient water usage and less run off. Having professional qualified irrigation technician setup up this automatic system and do the annual maintenance for that system will pay off in the long run. The maintenance will commence at the beginning of spring, with the systems back up batteries and configurations being checked and replace or corrected if necessary. The pipes are also checked for leakage and proper pressure, as well adjustments of the sprinkler heads for the correct coverage. At the end of the season the pipes must be blown out so they do not freeze during the winter months, this is done with compressed air. It cannot be stressed enough that these tasks are best done by professionals to avoid serious headaches. Ever had a frozen pipe burst, well it’s not fun.

How IoT is Revolutionizing Smart Irrigation Systems?

The Internet of Things (IoT) is playing a pivotal role in the evolution of smart irrigation systems. By connecting various components of an irrigation system to the internet, IoT makes it possible to remotely monitor and control the system. For instance, with IoT, a farmer can use a mobile app to check the soil moisture levels in his field and adjust the watering schedule accordingly. This not only saves time and effort but also ensures that the crops get the right amount of water. IoT also enables the integration of smart irrigation systems with other smart farming technologies such as drones, satellite imaging, and weather forecasting. This allows for a more holistic approach to farm management where different aspects of farming such as irrigation, fertilization, and pest control are seamlessly integrated.

1.5.2 Related Products

AI Integration in Different Types of Irrigation Systems

Here are the different types of irrigation systems that use AI technology and IoT sensors to achieve the advantages.

- **Sprinkler Irrigation**

AI technology has transformed traditional sprinkler systems into smart systems. An AI-based sprinkler irrigation system collects real-time weather data from rain sensors that measure the amount of rainfall and soil conditions to plan the next irrigation. After this data is analyzed, the system sends notifications to sprinklers to avoid overwatering or excessive water usage.

- **Drip Irrigation**

This is the most efficient irrigation method as it delivers water directly to the plant's roots, reduces runoff, ensures even distribution, and improves irrigation efficiency. AI further enhances this method. In AI-powered drip irrigation systems, sensors are placed in the soil to monitor soil moisture levels, which helps AI algorithms precisely control the water flow.

- **Center Pivot Irrigation**

This is a common irrigation method used in large-scale farming. In-field sensors installed on moving irrigation equipment collect data on soil and crop health. Sensors provide this information to an AI-based system that uses this data to control circle irrigation sprinklers and adjust the pivot's speed, stream, and angle of water flow.

- **The Bottom Line**

The integration of AI in irrigation systems is playing a crucial role in optimizing irrigation and solving water management issues in agriculture. AI technology has protected crops from a number of issues, such as climate change, water scarcity, and population growth. We must do all we can to reduce the effects of climate change and conserve water. Using water efficiently is one way to achieve this. With AI-based smart irrigation systems, farmers can save water, promote environmental sustainability, and improve crop yields.

1.5.3 Features

Smart Irrigation System

- **Sensor:**

The system is equipped with a Soil Moisture Sensor and a Light Detecting Resistor (LDR) that senses ambient light levels.

- **Function:**

The Soil Moisture Sensor checks for soil moisture to see whether it needs water or not and The LDR helps in adjusting the watering schedule to align with the most suitable times of the day. Typically, early morning or late evening are considered ideal for watering to minimize evaporation and maximize water absorption by the plants. By using the LDR, the system can delay or advance watering times to ensure the plants get the most benefit from the water provided.

Fire Detection System

- **Sensors:**

The fire detection system utilizes the DHT22 temperature and humidity sensor.

- **Function:**

This sensor continuously monitors the environmental conditions for any sudden increase in temperature or sustained high temperatures that could indicate the presence of a fire.

- **Alert System:**

Upon detecting abnormal temperature conditions, the system immediately sends a warning message to the owner's mobile device. Additionally, it activates a warning siren that alternates on and off to draw attention to the potential danger. These alerts allow for quick response actions to mitigate the risk of fire spreading.

Smart Protection System

- **Sensor:**

The system includes an HC-SR501 motion sensor.

- **Function:**

This sensor detects the presence of birds, rodents, or other intruders in the field or greenhouse. Upon detection, the system activates alternating warning sirens and lights to scare away the intruders. This deterrent helps to protect the crops from being damaged or eaten by pests.

- **Alert System:**

The system also sends an alert to the user’s mobile device, notifying them of the detected motion. This allows the user to take further action if necessary, such as inspecting the area or setting up additional protective measures.

Smart Lighting System

- **Conditions for Activation:**

The smart lighting system is designed to operate only under specific conditions to conserve energy. It activates only at night when the smart protection system is turned off and the motion sensor detects movement.

- **Function:**

This system provides illumination only when necessary, ensuring that the lights are used efficiently. By activating only under these conditions, the system minimizes unnecessary power consumption, thus promoting energy efficiency.

Auto Greenhouse Roll-up Sides System

- **Function:**

This feature allows for efficient air circulation and humidity management within the greenhouse. Proper air circulation is vital for maintaining an optimal growing environment, preventing mold growth, and regulating temperature and humidity levels.

- **Control:**

Users can easily operate the roll-up sides system with a simple click of a button on the mobile application. This remote-control capability adds convenience and ensures that the greenhouse environment can be adjusted as needed without manual effort.

Monitoring System (Mobile Application):

- **Data Display:**

It displays real-time data on temperature, humidity, soil moisture levels, and notifications for fire and motion detection.

- **Control Features:**

In addition to monitoring, the application offers full control over the system. Users can shut down the entire system or just the Smart Protection System or control the Greenhouse Roll-Up Sides System through the app.

Chapter 2

Proposed Work

- project layout
- Block Diagram
- Flow Chart
- Schematic Design
- Smart irrigation system
- Fire Detecting System
- Smart Protection System
- Smart Lighting System
- Auto Greenhouse Roll-up sides system
- Monitoring System (Mobile Application)

Chapter 2 - Proposed Work

2.1 project layout

This project layout represents a Sustainable Smart Agricultural System that integrates various subsystems managed by a central Mobile Application. Below is an explanation of each component:

1. Mobile Application:

- Acts as the central control and monitoring interface for the entire system.
- Allows users to manage, monitor, and control the different subsystems, ensuring seamless operation and real-time updates.

2. Auto Roll-up Sides System:

- Automates the process of rolling up or down the sides of a greenhouse or similar structure.
- Controlled through the mobile application, this system helps regulate internal conditions like temperature and airflow.

3. Smart Lighting System:

- Provides automated control of lighting based on the motion inside the greenhouse.
- Ensures optimal light exposure for the farmer, controlled via his existence or not.

4. Smart Irrigation System:

- Automates the watering process by adjusting the amount of water based on real-time data from the monitoring system.
- Ensures efficient water use, preventing overwatering or underwatering.

5. Fire Detection System:

- Monitors the environment for potential fire hazards.
- Alerts the mobile application and can trigger safety protocols to protect the farm.

6. Smart Protection System:

- Includes features like automated siren and lights going in alternative sequence to scare intruders.
- Designed to safeguard crops from various environmental threats.

7. Monitoring System:

- Gathers real-time data on environmental conditions such as soil moisture, temperature, humidity, and more.
- Provides critical data to the mobile application and other subsystems to enable precise control and decision-making.

This integrated system is designed to create a Sustainable Smart Agriculture System by automating key functions and providing a user-friendly interface through the mobile application. This ensures optimized resource use, improved crop protection, and enhanced overall efficiency.

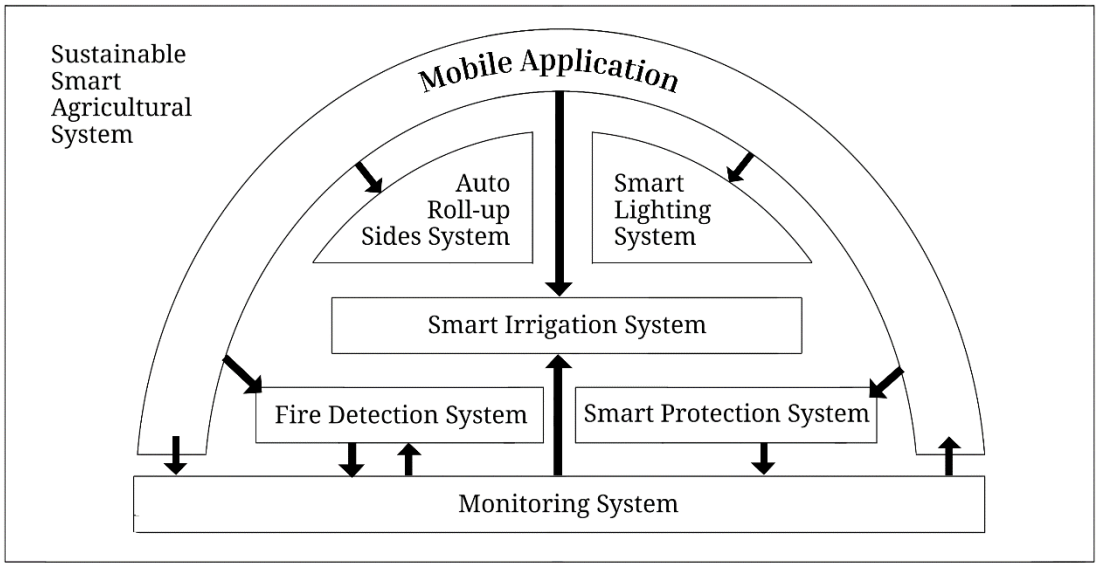


Figure 1 Project Layout

2.2 Block Diagram

For the start we need to discuss our (IPO) Input, Process and Output.

Input:

First, we have the input sensors. These sensors are the eyes and ears of our system. We have a Temperature and Humidity Sensor (DHT22) to monitor the atmospheric conditions, a Motion Sensor (HC-SR501) for detecting movement, a Soil Moisture Sensor to measure the water content in the soil, and a Light Detecting Resistor (LDR) to gauge light intensity.

Process:

All these inputs are processed by a powerful microcontroller, the ESP32. The ESP32 collects data from all the sensors, analyzes it, and makes real-time decisions to ensure optimal conditions for plant growth.

Output:

Based on the processed data, the system activates various outputs. An LED lamp provides the necessary light, a buzzer provides warning sirens, a water pump ensures proper irrigation, a servo motor (SG90) controls the rolling of greenhouse sides, and a mobile application allows farmers to monitor and control the system remotely. With this mobile application, farmers can stay connected to their fields no matter where they are, making informed decisions based on real-time data.

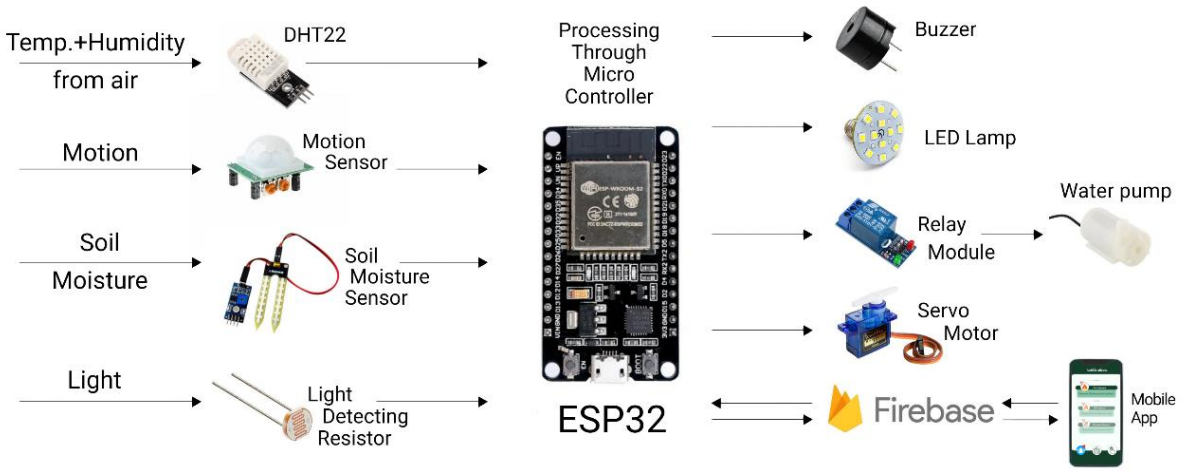


Figure 2 Block Diagram

2.3 Flow Chart

Here is the whole flowchart for our project explaining all its system functionalities.

Input:

- First, the system inputs moisture data from the soil moisture sensor, temperature and humidity from the DHT22, motion from the HC-SR501 sensor, and sunlight intensity from the Light Detecting Resistor (LDR).

Decision Making:

- Based on the moisture levels, the system decides whether to activate the water pump or close it to water the crops.
- After that, for fire detection, according to the temperature, if it goes above 60 degrees, we definitely have a fire, so the sirens will go on.
- Then we have the decision for smart protection and smart lighting systems. First, we need to input the protection system state (PS) from the mobile application. Then we check our motion sensor (HC-SR501). If it doesn't sense any motion, it is clear, and we move to the next decision.
- But if it senses motion, we check the LDR and the PS. If it is night and the PS is on, both the LED and the Buzzer will go on in an alternating sequence,

scaring the intruder. But if the PS is off, the lights will turn on to help our friend do their work easily in the darkness.

- If it is day instead of night, we check the PS. If the PS is on, the Buzzer Siren will go on, scaring the intruder. But if the PS is off, nothing will happen since we want to conserve our electricity for sustainability.
- Our final decision is if our friend, the farmer, wants to roll up the sides of their greenhouse. For that, we need to input the RUS state from the application and consider what our friend wants.

Output:

- Finally, we will send temperature in degrees, humidity in percentage, and soil moisture in percentage to the mobile application to be displayed to the user.

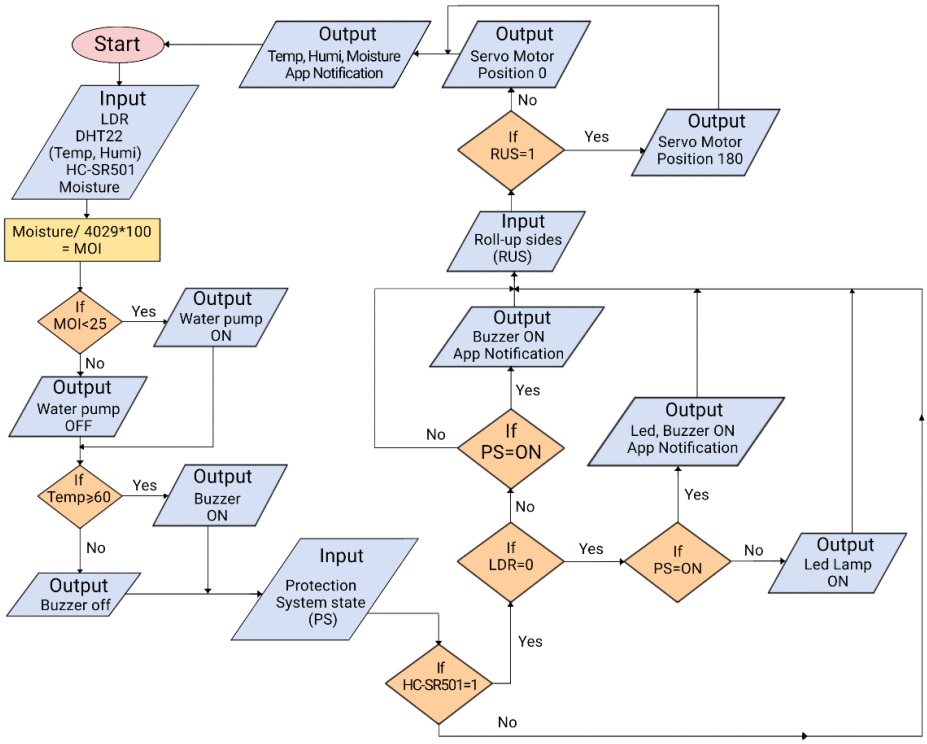


Figure 3 Flow Chart

2.4 Schematic Design

Dht22:
VCC ➔ 3.3 pin or vout+ (MT3608)
Data ➔ pin 23
GND ➔ GND or vout- (MT3608)

Pir sensor (motion sensor):
VCC ➔ 3.3V pin or vout+ (MT3608)
GND ➔ GND or vout- (MT3608)
Out (data) ➔ pin 22

MT3608:
Vin+ ➔3.3V pin
Vin- ➔GND
Vout+ ➔
Vout - ➔

LED Lamp:
+ PIN ➔ PIN16
- PIN ➔ GND

Servo motor:
(brown wire) ➔ GND or vout- (MT3608)
(Red wire) ➔ 3.3V pin or vout+ (MT3608)
(Orange wire) ➔ pin 18

Buzzer:
+ Pin ➔ PIN26
- pin ➔ GND

Relay Module:

VCC ➔ 3.3V pin or vout+ (MT3608)
GND ➔ GND or vout- (MT3608)
IN ➔ pin 2 (D2)

COM ➔ vout+ (MT3608)

NO (normal open) ➔ positive side of the pump

LDR:

One side ➔ 1k ohm resistor

Other side ➔ 3.3V pin

1k ohm resistor:
One side ➔ GND
Other side ➔ pin 32

Soil moisture:
VCC ➔ 3.3V pin or vout+ (MT3608)
GND ➔ GND or vout- (MT3608)
Ao ➔ pin 12

Water pump:
Red wire (+ve side of the pump) ➔ NO
black wire (-ve side of the pump) ➔ vout- (MT3608)

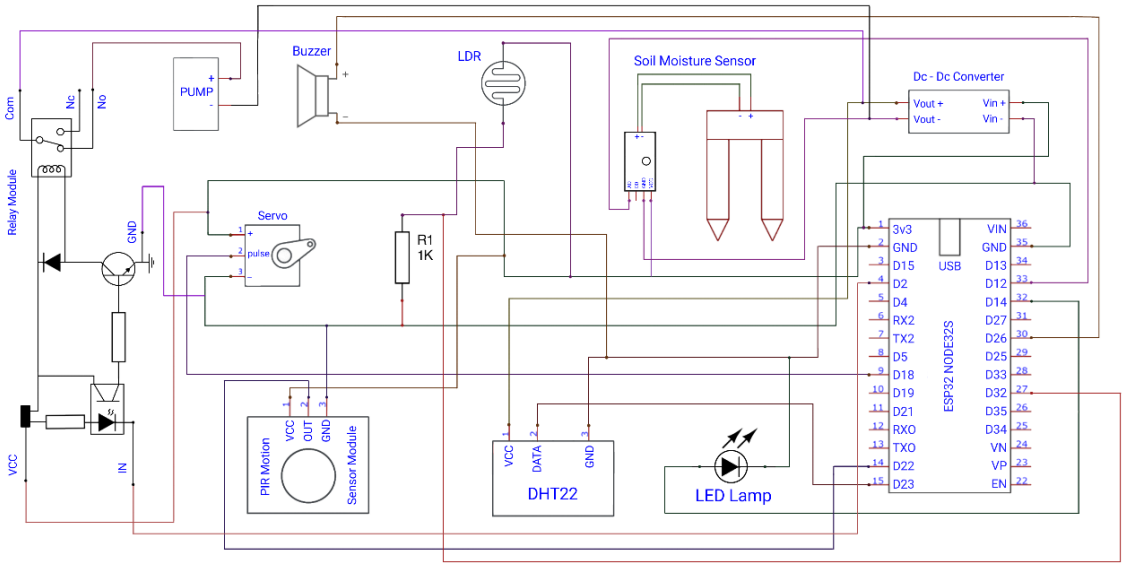


Figure 4 Schematic Design

2.5 Smart irrigation system

2.5.1 Description of Smart Irrigation

In warm climates, agriculture relies heavily on irrigation. However, the traditional irrigation methods result in water wastage and pollution. Smart water irrigation can help farmers automate irrigation without becoming dependent on it and easily check the crop status, ensuring the efficient use of water resources. It will also monitor soil moisture and evapotranspiration factors, notifying any water-related issues. The smart irrigation system is intertwined with a server that collects data and utilizes APLs for weather predictions, determining irrigation control based on that data. Moreover, SCADA can be integrated to implement automation and system control.

Smart irrigation systems utilize various information and communication technologies to schedule irrigation events using soil or crop data or meteorological information. This paper discusses recent developments on IOT sensors and platforms for smart irrigation solutions, focusing on farms and greenhouses. The proposed solutions are categorized into three parts: soil moisture-based, climate-based, and hybrid solutions. Several representative research prototypes operating in an isolated fashion or connected to the internet are also reviewed.

Definition and Importance

Agricultural irrigation is crucial for food production worldwide, yet in developing countries, a significant portion lacks access to irrigation systems. Non-efficient irrigation can lead to droughts and excess flood, which are severe problems. The issue has been so significant that developed countries such as the United States, European countries, and Asian countries like Australia and Japan have implemented systems to take care of their freshwater resources. Countries with less budget are trying to manage their water strategies, but in developing countries in Africa and the Middle East, these resources are in such a crisis that females must walk hours every day to get water supplies for their families. Agriculture is one of these regions that suffer from this drought due to high water needs. Reports indicate that lower production affects food supply and, as a consequence, affects the economy of the countries.

In 2002, the precipitation dumped during the monsoon season was the least it had been in the last 130 years. The Standardized Precipitation Evapotranspiration Index (SPEI) was used to determine that there was drought in the country caused by surface water anomalies. The SPEI index, along with information coming

from sensors capturing the environment, soil, and water, can demonstrate the current state of water in the country and the possibility that this country can cover its freshwater needs. Detecting droughts at an early stage gives countries the possibility of diversifying and acquiring water from other resources such as under groundwater. Countries that have higher funds are starting to implement systems in order to take care of their water re-use and management.

These systems improve irrigation systems, detect pollution in water supplies systems, and provide data that allow authorities to act to stop spreading the contamination and control the effects it might have on the population. However, in countries where the economy is unsteady and have scarce resources such as the majority of countries in Africa and the Middle East, the implementation of these systems is almost impossible.

2.5.2 Water Conservation Strategies

How Smart Irrigation Solution Can Help to Maximize Water Saving Efforts? Smart irrigation solutions can help farmers save water in several ways:

- **Soil Moisture Sensors**

Installing soil moisture sensors in the root zone enables precise measurement of the quantity of moisture present in the soil. With the use of this information, one may change watering schedules so as to prevent either overwatering or underwatering the plants.

- **Weather-Based Irrigation**

Weather-based irrigation systems collect real-time weather data to adjust watering schedules depending on daily evapotranspiration rates and precipitation.

- **Drip Irrigation**

The plant’s root zone receives water directly from drip irrigation systems, which helps reduce the amount of water lost to evaporation and runoff.

- **Smart Water Management**

Smart water management systems may reduce water waste and utility expenditures by monitoring water use and detecting leaks.

With smart irrigation solutions, farmers can reduce water waste while maintaining a high-quality and healthy plant.

2.5.3 Smart Irrigation Mechanism

Introduction

Smart irrigation system is designed to optimize water usage for agricultural and landscaping purposes by automating the watering process. This system uses a soil

moisture sensor, Water pump, and ESP32 to monitor soil conditions and apply water as needed, reducing human intervention and conserving water.

Key Components

1) **soil moisture sensor:** this is crucial for measuring soil moisture.
2) **ESP32:** These devices process data from the sensor and make decisions about when and how much to water. it can be programmed with specific irrigation schedules or use real-time data to adjust watering.
3) **Relay Module:** this allows ESP32 to communicate, with the water pump.
4) **Water pump:** this pump controls the flow of water to the irrigation system. it is activated by the controllers based on sensor data.

Working Mechanism

The system operates by continuously monitoring soil moisture levels. When the moisture level drops below a predefined threshold, the controller activates the actuators to open the valves and start irrigation. Once the soil moisture reaches the desired level, the system stops watering. This process ensures that plants receive the right amount of water at the right amount of water at the right time, preventing over-or under-watering.

2.6 Fire Detecting System

2.6.1 Fire Mechanics

Causes of fires and when fires occur?

Fires can occur in engineering projects due to several factors:

1. Inadequate Safety Planning*:* Lack of proper fire safety planning puts the project at risk. Identifying potential hazards and implementing appropriate preventive measures is essential.

2. Flammable Materials: Using flammable materials increases the likelihood of fires. Following safety standards and using non-flammable materials or applying fire control procedures is crucial.

3. Proper Material Storage: Improper storage of materials can lead to fires. Materials should be stored away from heat sources and regularly inspected.

4. Electrical Work: Incorrect wiring or using unrecognized electrical components can cause fires. Adhering to electrical standards and ensuring wire and component safety is essential.

5. Handling Hazardous Materials: Caution should be exercised when dealing with hazardous materials. Safety procedures must be followed to prevent leaks or contamination. [4] [5]

2.6.2 When Does the System Work?

➢ **Temperature Measurement:**

Thermistor: The DHT22 sensor includes a thermistor that changes its resistance with temperature.

This resistance change is measured and converted into a digital signal.

Analog-to-Digital Conversion: (ADC)Inside the DHT22, an internal Analog-to-Digital Converter (ADC) converts the analog signal from the thermistor into a digital value.

❖ **Data Transmission**

- **Digital Output:** The DHT22 uses a one-wire digital communication protocol. After the sensor measures the temperature and humidity, it encodes this information into a digital signal.
- **Timing:** The data is transmitted in a series of timed pulses, where the length of each pulse represents different bits of data. This is read by the ESP32 using its GPIO pins.

❖ **Connection DHT22 with ESP32**

When the ESP32 initializes communication with the DHT22, it sends a request to the sensor to start measurement. The ESP32 reads the data sent by the DHT22 over a single GPIO pin. The data is encoded in a specific protocol where the timing of the pulses Indicates whether each bit is a 0 or a 1. The ESP32 processes this data and converts it into meaningful temperature and humidity values.

❖ **Buzzer Alarm**

- **How It Works:** Most buzzers use a piezoelectric element that generates sound through the piezoelectric effect. When an alternating voltage is applied to the piezoelectric material, it vibrates and creates sound waves.
- **Buzzer Operation:** Electronic Signal: When a voltage is applied to the buzzer, it produces sound. The frequency of the sound can be controlled by adjusting the signal applied.
- **Alarm Sound:** The buzzer generates a loud sound, which is used as an audible alarm.

➢ **Integration with the System**

The ESP32 continuously monitors the temperature data from the DHT22. When the temperature exceeds 80 to 90 °C (indicating a potential fire), the ESP32 sends a signal to the buzzer.

- **Signal Output:** The ESP32 sets the GPIO pin connected to the buzzer to HIGH, which provides the necessary voltage to activate the buzzer.

- **Buzzer Response:** Sound Production: Upon receiving the signal from the ESP32, the buzzer activates and produces a sound, alerting users to the high temperature condition than 80 to 90 °C.
- **Timing:** The system can be programmed to sound the buzzer intermittently or continuously depending on the desired alert patter

In this system The DHT22 sensor measures temperature using a thermistor and converts the data into a digital format that is read by the ESP32. The ESP32 processes this data and triggers the buzzer when the temperature exceeds 80 to 90 °C. The buzzer produces a sound through the piezoelectric effect, providing an audible alarm to alert users of a potential fire or high temperature situation. The interaction between these components provides a basic fire alarm system by combining temperature measurement, data processing, and audible alerts. So, we can deduce that the fire alarm system works by continuously monitoring the temperature through the DHT22 sensor. The buzzer will activate and sound an alarm when the temperature exceeds the predefined threshold, helping to alert you to a potential fire or overheating situation

2.7 Smart Protection System

2.7.1 Rodent & Bird Damage

Agricultural pests, such as birds and rodents, may cause significant damage to crops and reduce growers' ability to provide agricultural commodities to the market. When this occurs, the broader economy may suffer due to reduced production and fewer commodities for processing and sale. If the agricultural sector plays a major role in the economy, the multiplier effects of this type of damage may be adverse since the agricultural sector typically provides inputs to almost all other sectors of the economy (e.g., manufacturing, retail trade, and accommodation and food service). [6] [7] [8]

❖ **Birds**

1. Crop Damage:

Birds, especially larger flocks of species like starlings, crows, and pigeons, can cause significant damage to crops by pecking at fruits, seeds, and grains. This can lead to substantial economic losses for farmers.

2. Seedling Damage:

Some bird species, such as sparrows, might peck at young seedlings, damaging them or removing them entirely, which can impact crop yields.

3. Spread of Disease:

Birds can also spread diseases to crops and livestock through their droppings, potentially leading to more severe problems for farmers.

❖ **Rodents**

4. Feeding on Crops:

Rodents, such as mice, rats, and voles, can cause severe damage by feeding on seeds, grains, and even mature crops. They can destroy large quantities of food, impacting both crop yield and quality.

5. Burrowing:

Rodents often create burrows in fields, which can lead to physical damage to the land and infrastructure, affecting machinery and planting.

6. Contamination:

Rodent droppings and urine can contaminate crops and stored produce, leading to food safety concerns and potential economic loss.

7. Spread of Disease:

Rodents are known carriers of various diseases, including hantavirus and leptospirosis, which can pose health risks to both humans and livestock.

2.7.2 Mechanics of the Protection System

The Smart Security System is an advanced security system designed to prevent bird and rodent invasion. The system integrates ESP32 microcontroller with HC-SR501 motion sensor to detect motion and trigger a comprehensive alarm mechanism that includes audio and visual signals.

System Components

- **ESP32 Microcontroller:** Serves as the central control unit.
- **HC-SR501 Motion Sensor:** Detects motion and triggers the system.
- **Alarm System:** Includes alternating sirens and lights to deter intruders.

Operation

The Smart Protection System works as follows:

- **Motion Detection:** The HC-SR501 sensor continuously monitors for movement by detecting changes in infrared radiation, indicating the presence of birds, rodents, or other intruders. When motion is detected, it sends a signal to the ESP32.
- **Activation:** The ESP32 processes the signal and activates the alarm system, which includes alternating sirens and lights. The alternating pattern of the sirens and lights creates an unpredictable environment, which helps in scaring away the intruders more effectively.
- **User Notification:** Concurrently, the system sends a notification to the user. This alert can be delivered through mobile application. This alert informs the user of the detected motion and the activation of the warning measures.

2.8 Smart Lighting System

2.8.1 Challenges Faced by Farmers at Night time

Nighttime agricultural operations are critical for maintaining farm productivity and safety. However, the reduced visibility and associated risks can lead to inefficiencies and potential hazards. [9] [10]

1. Limited Visibility:

At night, visibility is significantly reduced, making it difficult for farmers to navigate their fields, check on livestock, or perform maintenance tasks. This can lead to accidents or inefficiencies.

2. Safety Concerns:

Farmers often work alone at night, which can pose safety risks. Inadequate lighting increases the likelihood of accidents, such as tripping or equipment malfunctions, and can also make it harder to spot potential security threats.

3. Energy Consumption:

Traditional lighting systems often stay on all night, leading to unnecessary energy consumption and higher costs. This is particularly problematic for large farms where lighting needs can be extensive.

4. Equipment and Infrastructure Monitoring

Farmers need to regularly check on equipment and infrastructure, which can be challenging at night due to poor lighting. Issues like equipment malfunctions or infrastructure damage might go unnoticed without proper illumination.

5. Livestock Management:

Checking on livestock at night is crucial for their well-being, but it can be difficult with inadequate lighting. Poor visibility can hinder the farmer's ability to detect health issues or monitor the animals effectively.

2.8.2 Operation of the Smart Lighting System

To conserve energy and provide illumination, when necessary, a smart lighting system is incorporated. This system is activated under the following conditions:

1. Ambient light levels have decreased to a predetermined threshold, as detected by the LDR.
2. The smart protection system is deactivated, indicating the absence of intruders.
3. Motion is detected within the field or greenhouse by the motion sensor.

By satisfying these criteria, the system ensures that lighting is only employed when required, optimizing energy efficiency.

2.8.3 Energy Efficiency Benefits

Electric energy efficiency represents a transformative approach to reducing energy consumption while maintaining or even enhancing performance. By optimizing how electricity is used, this concept not only provides substantial economic, environmental, and social benefits but also plays a pivotal role in fostering sustainable development.

Economic Advantages

One of the primary benefits of electric energy efficiency is the substantial cost savings it offers. For both households and businesses, energy-efficient technologies and practices result in lower electricity bills. Modern appliances, lighting systems, and heating, ventilation, and air conditioning (HVAC) systems are designed to use less power while delivering the same, if not superior, levels of service. This reduction in energy consumption translates directly into financial savings.

Moreover, energy efficiency can increase property values. Buildings equipped with energy-efficient features such as high-performance windows, efficient insulation, and advanced lighting systems are often more attractive to potential buyers. These features promise lower utility costs and reduced environmental impact, making such properties more desirable in the real estate market.

The energy efficiency sector also contributes to job creation. The demand for energy-efficient products and services stimulates economic activity and generates employment opportunities in manufacturing, installation, and maintenance sectors. This boost in job creation can positively impact local economies and support sustainable economic growth.

Environmental Benefits

Electric energy efficiency plays a crucial role in environmental conservation. By using less electricity, the demand for power generation decreases, which in turn reduces the need for fossil fuels, such as coal and natural gas. Power plants burning these fuels are significant sources of greenhouse gas emissions, which contribute to global warming and climate change. By reducing overall energy consumption, we can decrease the amount of greenhouse gases released into the atmosphere, helping to mitigate the impacts of climate change and improve air quality.

Furthermore, energy efficiency helps conserve natural resources. Reduced electricity demand means less strain on resources required for power generation, such as coal, oil, and water. This conservation not only helps preserve these

resources for future generations but also reduces the environmental degradation associated with their extraction and use.

Social Benefits

On a social level, energy efficiency contributes to improved comfort and health. Energy-efficient buildings often feature better insulation, optimized lighting, and improved ventilation, which can enhance indoor air quality and overall living conditions. These improvements lead to a more comfortable and healthier environment for occupants, reducing the risk of health issues related to poor indoor air quality and energy inefficiency.

Energy efficiency also plays a role in promoting energy accessibility and equity. By lowering energy costs, it becomes more affordable for low-income households to access and maintain necessary energy services. This can help alleviate energy poverty and ensure that more people can benefit from reliable and affordable energy.

2.9 Auto Greenhouse Roll-up sides system

2.9.1 What Are Roll-Up Sides for Greenhouses?

Roll-up sides are a simple yet effective solution for improving ventilation in greenhouses. Essentially, they involve rolling up the sides of your greenhouse to allow extra airflow. If you have a plastic film covering on your greenhouse and lack easy access to fans, roll-up sides can be a game changer. Greenhouse roll-up sides move primarily based on temperature and ventilation needs.

2.9.2 Benefits of Greenhouse Roll-Up Sides

- *Temperature Control:* Greenhouses can get excessively hot, especially in sunny weather. Rolling up the sides allows hot air to escape, reducing internal temperatures and preventing stress on plants.
- *Humidity Management:* Greenhouses trap moisture, leading to high humidity levels. Ventilation via roll-up sides helps maintain healthier humidity levels and prevents mold and mildew growth.
- *Air Circulation:* Proper air movement strengthens plant stems, reduces fungal diseases, and improves pollination and CO2 distribution.
- *Pest and Disease Control:* Ventilation discourages pests and diseases that thrive in stagnant, humid conditions.
- *Energy Efficiency:* Roll-up sides provide natural ventilation without relying on electricity, reducing costs compared to mechanical systems like fans.

2.9.3 Roll-up sides system mechanism

This system employs a human-in-the-loop approach to control greenhouse side panel movement. A servo motor is the primary actuator responsible for the mechanical motion of the panels.

- **Servo Motor:**

A rotational actuator capable of precise angular control, providing the necessary torque to move the greenhouse side panels. A 180-degree rotation of the servo motor would typically indicate a fully open or closed panel position, depending on the mechanical linkage design.

- **ESP 32:**

A microcontroller or similar device responsible for processing user commands and controlling the servo motor.

- **User Interface:**

A mobile application allowing the farmer to interact with the system and initiate panel movement.

- **Mechanical Linkage:**

Connects the servo motor to the side panels, enabling the desired motion.

- **System Operation**:

The farmer utilizes the mobile application to send commands to the control unit, specifying the desired panel position. The control unit translates this command into a corresponding servo motor angle. The servo motor then rotates to the specified angle, causing the side panels to move accordingly.

2.10 Monitoring system (Mobile Application)

2.10.1 Difficulties in Rigorous Monitoring

Rigorous monitoring is critical in various fields, including project management, healthcare, manufacturing, and environmental management. It aims to ensure quality, compliance with standards, and efficient achievement of objectives. However, the process of rigorous monitoring is fraught with challenges that can significantly impact its effectiveness. This article explores the primary difficulties encountered in rigorous monitoring and proposes strategies to address them. One of the foremost challenges in rigorous monitoring is the complexity of the data involved. The data sets that require monitoring are often large and multifaceted, including various types of information that can complicate the analysis process.

Data for rigorous monitoring often comes from numerous sources. This can include databases, real-time feeds, and various external inputs. Integrating data

from these disparate sources into a coherent system poses significant challenges. The sheer volume and diversity of data can overwhelm traditional data management systems, making it difficult to achieve a unified view. Unstructured data, such as text, images, and video, further complicates monitoring efforts. Unlike structured data, which is organized in a predefined manner, unstructured data lacks a standardized format, making it difficult to analyze using conventional tools. Advanced data analytics techniques, such as natural language processing and image recognition, are required to extract meaningful insights from unstructured data.

2.10.2 Role and functionality of the Monitoring system

The system's monitoring function aggregates data from various sensors and presents it through a mobile application. Users can access real-time information regarding temperature, humidity, soil moisture, and receive alerts for fire and motion detection events. Moreover, the application provides comprehensive control over the system, allowing users to deactivate the entire system or the protection system independently.

Chapter 3

Hardware System Implementation

- ESP32
- DHT22
- Soil Moisture
- HC-SR501
- LDR
- Servo Motor
- Water pump
- Relay Module
- MT3608

Chapter 3 – Hardware System Implementation

3.1 ESP32

3.1.1 Definition

The ESP32 is a series of chip microcontrollers developed by Espressif. ESP32 comes with an on-chip 32-bit microcontroller with integrated Wi-Fi + Bluetooth + BLE features that targets a wide range of applications. It is a series of low-power and low-cost.

The ESP32 is a series of low-cost, low-power system-on-chip (SoC) microcontrollers with integrated Wi-Fi and dual-mode Bluetooth. It is a versatile and robust device that offers a wide range of features, including a dual-core processor, a large memory capacity, and a rich set of peripherals. Created by Espressif Systems, a Shanghai-based Chinese company, the ESP32 has undergone significant evolution since its introduction. It is a successor to the ESP8266 microcontroller

Another important thing to know about ESP32 is that it is manufactured using TSMC's ultra-low-power 40 nm technology. So, designing battery operated applications like wearables, audio equipment, baby monitors, smart watches, etc., using ESP32 should be very easy.

The ESP32 series employs either a Tensilica Xtensa LX6 microprocessor in both dual-core and single-core variations, an Xtensa LX7 dual-core microprocessor, or a single-core RISC-V microprocessor. The good thing about ESP32, like ESP8266 is its integrated RF components like Power Amplifier, Low-Noise Receive Amplifier, Antenna Switch, Filters and RF Balun. This makes designing hardware around ESP32 very easy as you require very few external components.

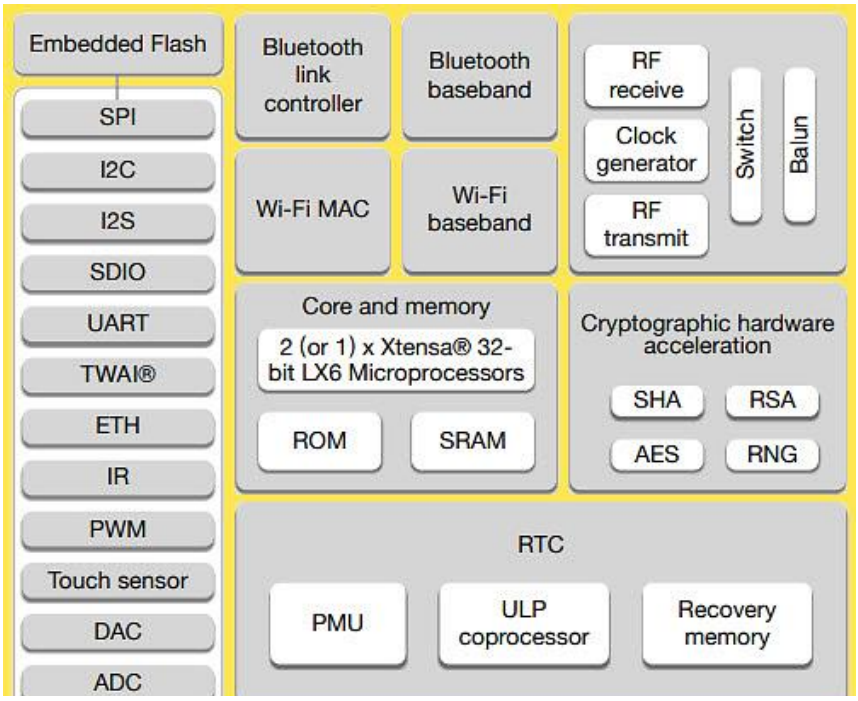


Figure 5 Embedded System Architecture Diagram

3.1.2 History

The original ESP32 chip had one processor core called Tensilica Xtensa LX6. This processor could run at speeds over 240 MHz, meaning it could handle data and tasks quickly. More recently, new models were added

1. New Models: Recently, there are new versions of ESP32 chips called ESP32-C and ESP32-S.

2. CPU Types: These new models use a different type of CPU called Risc-V, instead of the old one called Xtensa.

3. Comparison to ARM: Risc-V is a type of CPU architecture, like ARM, which is very popular and well-supported. Risc-V is open source, meaning it's freely available and can be easily used by anyone.

4. Compiler Support*:* Both Risc-V and ARM work well with common compilers (tools that turn code into programs). Xtensa needed special support to work with these tools, making it less convenient to use.

In summary, the new ESP32 models use the Risc-V CPU, which is easier to work with and has good support, similar to ARM but with the added benefit of being open source

The newer ESP32 models come with options for either both Wi-Fi and Bluetooth, or just Wi-Fi. Here are some of the available chip models:

- ESP32-D0WDQ6 (also known as ESP32D0WD)
- ESP32-D2WD
- ESP32-S0WD
- ESP32-PICO-D4 (a System in Package or SiP)
- ESP32 S series
- ESP32-C series
- ESP32-H series

The ESP32 is widely used in mobile devices, wearable tech, and Internet of Things (IoT) applications, such as Nabto Edge. With the introduction of the ESP32 IoT Starter Kit by Mongoose OS, it has become very popular among hobbyists and IoT developers. Its capabilities and popularity have grown significantly over the past few years.

3.1.3 Specifications

Datasheet

- Single or Dual-Core 32-bit LX6 Microprocessor with clock frequency up to 240 MHz.
- 520 KB of SRAM, 448 KB of ROM and 16 KB of RTC SRAM.
- Supports 802.11 b/g/n Wi-Fi connectivity with speeds up to 150 Mbps.
- Support for both Classic Bluetooth v4.2 and BLE specifications.
- 34 Programmable GPIOs.
- Up to 18 channels of 12-bit SAR ADC and 2 channels of 8-bit DAC
- Serial Connectivity include 4 x SPI, 2 x I2C, 2 x I2S, 3 x UART.
- Ethernet MAC for physical LAN Communication (requires external PHY).
- 1 Host controller for SD/SDIO/MMC and 1 Slave controller for SDIO/SPI.
- Motor PWM and up to 16-channels of LED PWM.
- Secure Boot and Flash Encryption.
- Cryptographic Hardware Acceleration for AES, Hash (SHA-2), RSA, ECC and RNG

Processors

Most models have a dual-core design, except for the ESP32-S0WD, which has a single core.

- **CPU:** Xtensa dual-core (or single-core) 32-bit LX6 microprocessor, operating at 160 or 240 MHz and performing at up to 600 DMIPS.
- **Ultra-low power (ULP) co-processor:** Its low power consumption allows it to perform tasks like analog-to-digital conversions even in deep sleep mode.

Wireless connectivity

- **Wi-Fi:** 802.11 b/g/n
- **Bluetooth:** v4.2 BR/EDR and BLE (Bluetooth Low Energy)

Memory

- **ROM:** 448 KB (for booting and core functions)
- **SRAM:** 520 KB (for data and instructions)
- **RTC fast SRAM:** 8 KB (for data storage and main CPU during RTC Boot from the deep-sleep mode)
- **RTC slow SRAM:** 8KB (for co-processor accessing during deep-sleep mode)
- **eFuse:** 1 Kbit (of which 256 bits are used for the system (MAC address and chip configuration) and the remaining 768 bits are reserved for customer applications, including Flash-Encryption and Chip-ID)

Some models, like the ESP32-D2WD and ESP32-PICO-D4, include internal flash memory.

- **Embedded flash:** flash connected internally via IO16, IO17, SD_CMD, SD_CLK, SD_DATA_0 and SD_DATA_1 on ESP32-D2WD and ESP32-PICO-D4.
  - 0 MiB (ESP32-D0WDQ6, ESP32-D0WD, and ESP32-S0WD chips)
  - 2 MiB (ESP32-D2WD chip)
  - 4 MiB (ESP32-PICO-D4 SiP module)
- **External Flash and SRAM:** The ESP32 can connect to up to four 16 MB external QSPI flashes and SRAMs. It uses AES encryption for security and high-speed caches for fast access.

Peripheral input and out put

- peripheral interface with DMA that includes capacitive touch
- ADCs (Analog-to-Digital Converter)
- DACs (Digital-to-Analog Converter)
- I²C (Inter-Integrated Circuit)
- UART (Universal Asynchronous Receiver/Transmitter)
- SPI (Serial Peripheral Interface)
- I²S (Integrated Interchip Sound)
- RMII (Reduced Media-Independent Interface)
- PWM (Pulse-Width Modulation)

Security

The ESP32 supports all standard IEEE 802.11 security features, such as WPA/WPA2 and WAPI. It also includes secure boot and flash encryption to protect your data.

3.1.4 Features

• **Low-power:** the ESP32 consumes very little power compared with other microcontrollers, and it supports low-power mode states like deep sleep to save power.

• **Wi-Fi capabilities:** the ESP32 can easily connect to a Wi-Fi network to connect to the internet (station mode), or create its own Wi-Fi wireless network (access point mode) so other devices can connect to it. This is essential for IoT and Home Automation projects. You can have multiple devices communicating with each other using their Wi-Fi capabilities.

• **Bluetooth:** the ESP32 supports Bluetooth classic and Bluetooth Low Energy (BLE). which is useful for a wide variety of IoT applications.

• **Dual-core:** most ESP32 are dual-core. They come with 2 Xtensa 32-bit LX6 microprocessors: core 0 and core 1.
• **Rich peripheral input/output interface:** the ESP32 supports a wide variety of input (read data from the outside world) and output (to send commands/signals to the outside world) peripherals like capacitive touch, ADCs, DACs, UART, SPI, I2C, PWM, and much more.
• **Compatible with the Arduino "programming language":** those that are already familiar with programming the Arduino board, you'll be happy to know that they can program the ESP32 in the Arduino style.
• **Compatible with MicroPython:** you can program the ESP32 with MicroPython firmware, which is a re-implementation of Python 3 targeted for microcontrollers and embedded systems

3.1.5 ESP32 Functions

The ESP32 is versatile for IoT applications. Here are some of its key functions:

- **Networking:** With its Wi-Fi capability and dual-core processor, the ESP32 connects embedded devices to routers and handles data transmission.
- **Data Processing:** It processes input from sensors and performs complex calculations. It can operate with either a real-time operating system (RTOS) or a simpler software development kit (SDK) that runs directly on the chip without needing a full OS.
- **P2P Connectivity:** It supports peer-to-peer communication, allowing direct interaction between ESP32 devices and other IoT devices.
- **Web Server:** The ESP32 can host web pages, making it possible to access and manage devices through HTML or other development languages.

3.1.6 How to program ESP32 board?

The ESP32 board can be programmed using many available tools, the most popular of which are:
1- Arduino IDE: The Arduino IDE development environment can be used to program the ESP32 board, this requires installing the appropriate libraries and setting up the port to which the board is connected.
2- MicroPython: MicroPython can be used to program the ESP32 board, and this language is easier to learn and use compared to Arduino.
3- Visual Studio Code: Visual Studio Code can be used to program the ESP32 board using different programming languages, and it has many extensions that facilitate the programming process.

4- PlatformIO: The PlatformIO environment available as an add-on to Visual Studio Code can be used to program the ESP32 board. This environment provides many features such as the ability to easily add libraries and run codes on different operating systems.

The ESP32 board is programmed using different programming languages such as C, C++, and Python, and the generated software is loaded onto the board using available software downloaders such as esptool.py or Flash Download Tools provided by Espressif, the company that produces the ESP32 board.

- **LUA**
- **Espressif IDF (IoT Development Framework)**
- **JavaScript**

3.1.7 ESP32 DevKit – The ESP32 Development Board

Espressif Systems released several modules based on ESP32 and one of the popular options is the ESP-WROOM-32 Module. It consists of ESP32 SoC, a 40 MHz crystal oscillator, 4 MB Flash IC and some passive components. The good thing about ESP-WROOM-32 Module is the PCB has edge castellations. So, what third-part manufacturers do is take the ESP-WROOM-32 Module and design a break-out board for this module. One such board is the ESP32 DevKit Board. It contains the ESP-WROOM-32 as the main module and also some additional hardware to easily program ESP32 and make connections with the GPIO Pins.

Figure 6 ESP32 DevKit

IMPORTANT NOTE:

Our board has 30 Pins (15 pins on each side). There are some boards with 36 Pins and some with slightly less Pins.

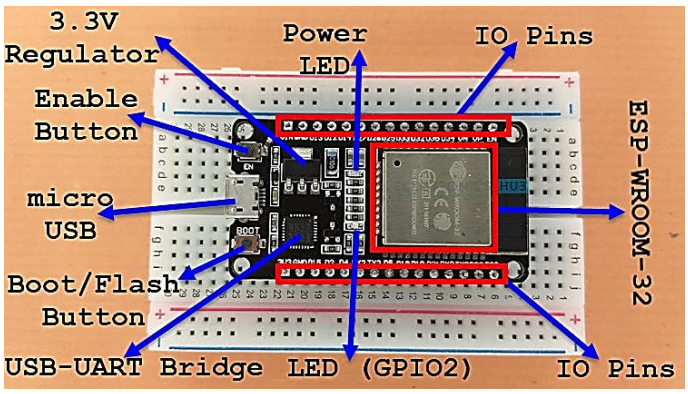


Figure 7 ESP32 Board

the ESP32 Board consists of the following:

- ESP-WROOM-32 Module
- Two rows of IO Pins (with 15 pins on each side)
- CP2012 USB – UART Bridge IC
- micro–USB Connector (for power and programming)
- AMS1117 3.3V Regulator IC
- Enable Button (for Reset)
- Boot Button (for flashing)
- Power LED (Red)
- User LED (Blue – connected to GPIO2)
- Some passive components

An interesting point about the USB-to-UART IC is that its DTR and RTS pins are used to automatically set the ESP32 in to programming mode (whenever required) and also rest the board after programming.

Reset/Boot buttons

In ESP32 board comes with two main push buttons one is the Reset (RST/EN) button another is the BOOT button. The reset button is used to reset the ESP32 Chip. The use of the boot button is to enter in boot mode to upload the new sketch or program.

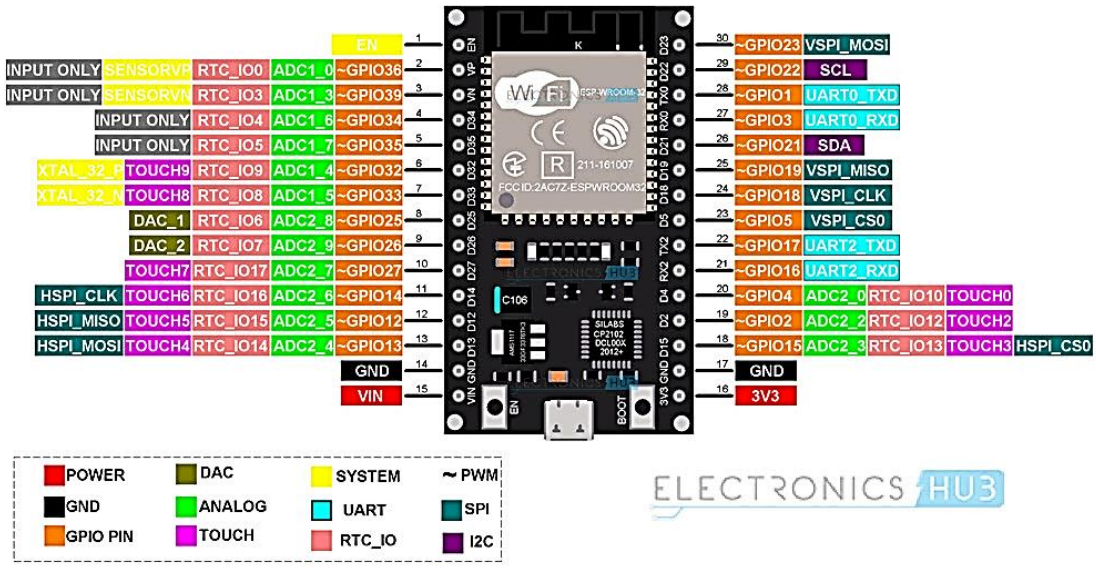


Figure 8 ESP32 PINOUT

3.1.8 WIFI

The built-in WiFi.h library will allow us to use the WiFi features of the ESP32 board easily.

The ESP32 has 2 WiFi modes:

Station (Wi-Fi _STA):

The Station mode (STA) is used to connect the ESP32 module to a WiFi access point. The ESP32 behaves like a computer that is connected to our router. If the router is connected to the Internet, then the ESP32 can access the Internet. The ESP32 can behave as a client: make requests to other devices connected to the network, or as a server: other devices connected to the network will send requests to the ESP32. In both cases, the ESP32 can access the Internet.

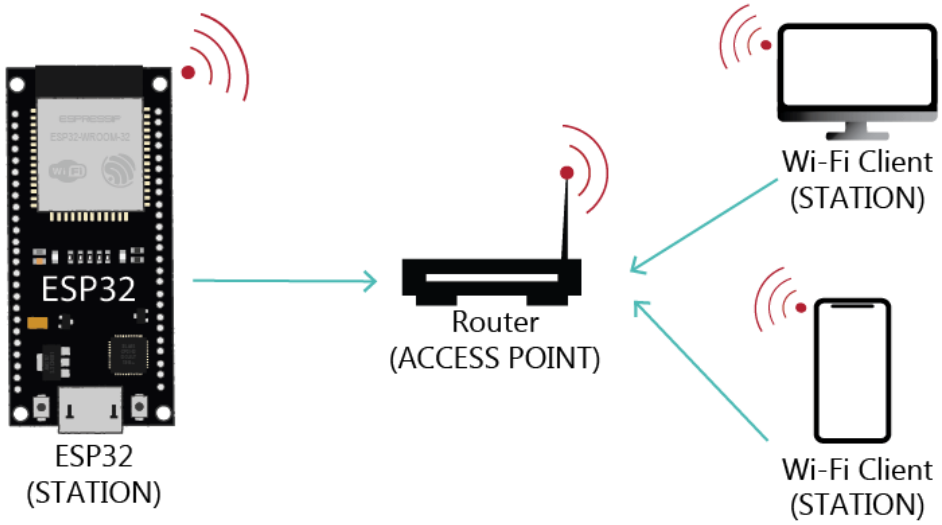


Figure 9 Station (Wi-Fi _STA)

AP (Access Point) (WIFI_AP):

In Access Point mode, the ESP32 behaves like a Wi-Fi network (a bit like a router): other devices can connect to it. In this mode, the ESP32 is not connected to any other network and is therefore not connected to the Internet. This mode is more computationally and energy-intensive (the ESP32 board will heat up) since the ESP32 has to simulate a full Wi-Fi router (Soft AP). The latency and the bandwidth will be less efficient than in a classic router.

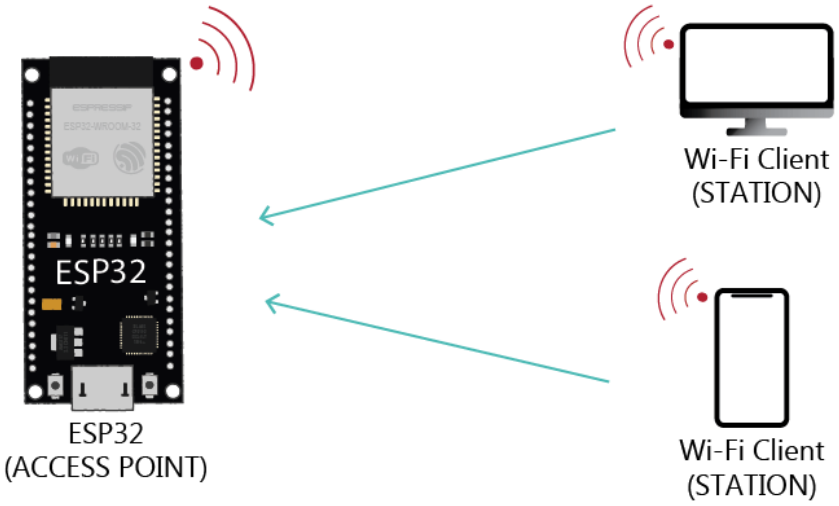


Figure 10 AP (Access Point) (WIFI_AP)

For mode selection:

In general, we use the STATION mode. We can access the Internet to retrieve information from an API, have a home automation server with sensors. The AP mode is usually used temporarily to enter the credentials of the WIFI network (SSID and password). It can also be used to have a separate network from your home network and not be connected to the Internet. The ESP32 board can act as Wi-Fi Station, Access Point or both. To set the Wi-Fi mode, use WIFI. Mode () and set the desired mode as argument:

Table 1 WIFI Modes

WIFI. Mode (WIFI_STA)	station mode: the ESP32 connects to an access point
WIFI. Mode (WIFI_AP)	station mode: the ESP32 connects to an access point
WIFI. Mode (WIFI_AP_STA)	access point mode: stations can connect to the ESP32

Scan Wi-Fi Networks

This can be useful to check if the Wi-Fi network you're trying to connect is within the range of your board.

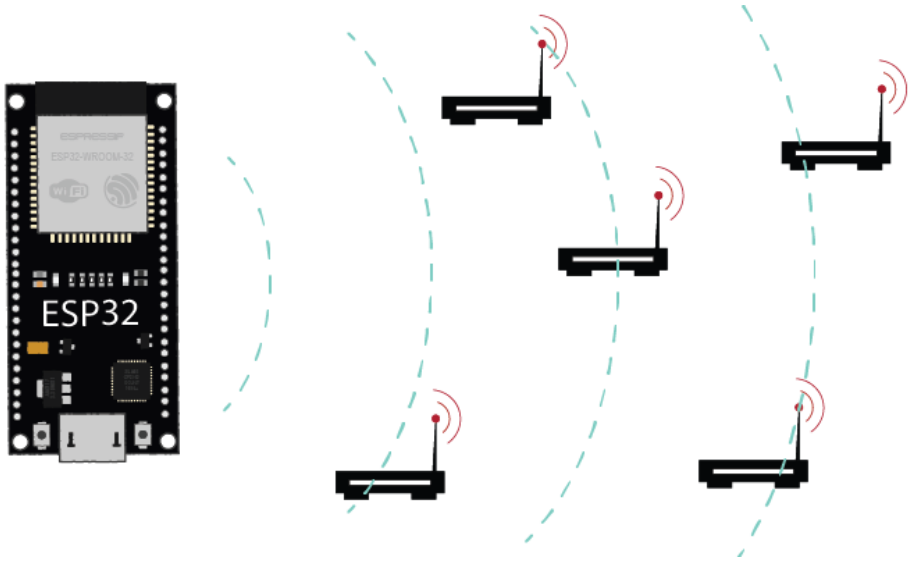


Figure 11 Scan Wi-Fi Networks

3.1.9 Bluetooth

The ESP32 supports dual-mode Bluetooth, which means it supports both Bluetooth Classic and Bluetooth Low Energy (BLE). While these two protocols share many important things such as architecture, and both operate in 2.4 GHz ISM (Industrial, Scientific, and Medical) band, they're quite different from each other.

Bluetooth Classic

Bluetooth Classic is designed for continuous two-way data transfer with high application throughput (up to 3 Mbps); highly effective, but only for short distances. It uses over 79 channels in the 2.4 GHz ISM band. This type of Bluetooth is mainly employed in projects where you need to continuously stream data like audio streaming or file transfer.

Bluetooth Low Energy (BLE)

Bluetooth LE, originally marketed as Bluetooth Smart and commonly referred to as just BLE, is designed for very low power operation while maintaining a similar communication range. However, BLE is much more than a low-power version of Bluetooth Classic. BLE operates in the same radio spectrum range as Bluetooth Classic, but uses a different set of 40 channels at a lower data rate (up to 1 Mbps). It's well-suited for IoT projects where power consumption is a main concern—for example, a project where you need to wake up the device periodically, gather sensor data, transmit it using Bluetooth, and then return to sleep mode.

Bluetooth Classic vs. BLE

BLE is primarily used to save power, but there are actually several important differences.

Power Consumption: Bluetooth Classic typically consumes more power, while BLE is designed for low power consumption, making it suitable for battery-powered devices that need to operate for extended periods.

Data Transfer Rates: Bluetooth Classic provides higher data rates than BLE, making it suitable for projects that require continuous and high-speed data transmission, while BLE is optimized for short bursts of data transmission.

Range: Both BLE and Classic can cover up to 100 meters, but the exact distance varies depending on the environment and implementation.

Latency: Bluetooth Classic connections have a latency of up to 100 MS, while BLE connections have a latency of 6 Ms. Keep in mind that lower is better.

Compatibility: Bluetooth Classic is more prevalent in older devices, while BLE is commonly used in modern smartphones and other gadgets.

Table 2 Bluetooth Comparison

	Bluetooth Classic	Bluetooth Low Energy (BLE)
Data Rate	1 Mbps for BR2-3 Mbps for EDR	500 kbps – 1 Mbps
Frequency Band	2.4 GHz ISM band	2.4 GHz ISM band
Number of Channels	79 Channels with 1 MHz spacing	40 Channels with 2 MHz spacing
Communication Range	8 m up to 100 m	8 m up to 100 m
Power Consumption	High (up to 1 W)	Low (0.01 W up to 0.5 W)
Device Pairing is Mandatory?	YES	NOT Mandatory
Supported Topologies	Point-to-Point (1:1)	Point-to-Point (1:1)Broadcast (1:many)Mesh (many:many)
Modulation Technique	GFSKπ/4 DQPSK8 DPSK	GFSK
Latency	35ms	2-16ms (Avg. 9ms)

3.1.10 ESP32 Advantages and Disadvantages

ESP32 Advantages

The ESP32 boasts remarkable processing power and an impressive array of connectivity features. Its dual-core processor allows for efficient multitasking, making it suitable for a wide range of applications requiring substantial computing capability. Additionally, the ESP32 integrates various wireless protocols, such as Wi-Fi and Bluetooth, enabling seamless communication with other devices.

ESP32 Disadvantages

One potential disadvantage of the ESP32 is its relatively steep learning curve. Due to its advanced features and capabilities, beginners may need to invest time in learning its intricacies to grasp the full potential of the ESP32. However, once the learning curve is overcome, the ESP32 proves to be a powerful and versatile platform for electronics projects.

3.2 DHT22

3.2.1 Description

The DHT22 is a low-cost digital temperature and humidity sensor with a single wire digital interface. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and spits out a digital signal on the data pin (no analog input pins needed). The sensor is calibrated and doesn't require extra components so you can get the right to measuring relative humidity and temperature.

Uses

its ability to measure temperature and humidity accurately, here are some common uses and project ideas.

Common Uses:

- *Weather Stations:* Monitor and record local weather conditions.
- *Home Automation:* Control HVAC systems based on temperature and humidity levels.
- *Greenhouse Monitoring:* Ensure optimal growing conditions for plants.
- *Data Logging:* Collect and analyze environmental data over time.
- *IoT Projects:* Integrate with IoT platforms for remote monitoring and control.

3.2.2 Comparison between DHT11 and DHT22 [11]

Table 3 Comparison between DHT11 and DHT22

	DHT11	DHT22
Picture		
Temperature range	0 to 50 ºC +/-2 ºC	-40 to 80 ºC +/-0.5ºC
Humidity range	20 to 90% +/-5%	0 to 100% +/-2%
Resolution	Humidity: 1% Temperature: 1ºC	Humidity: 0.1% Temperature: 0.1ºC
Operating voltage	3 – 5.5 V DC	3.3-6V DC
Current supply	0.5 – 2.5 mA	1 – 1.5 mA
Sampling period	1 second	2 seconds
Price	80 EGP	220 EGP

3.2.3 Technical details of DHT22 [12]

Table 4 Technical details of DHT22

Model	DHT22/AM2302
Power supply	3.3-6V DC
Output signal	Digital signal via 1-wire bus
Sensing element	Polymer capacitor
Operating range	Humidity 0-100%RH; Temperature -40~80Celsius
Accuracy	Humidity +-2%RH(Max +-5%RH); Temperature <+-0.5Celsius
Resolution or sensitivity	Humidity 0.1%RH; Temperature 0.1Celsius
Repeatability	Humidity +-1%RH; Temperature +-0.2Celsius
Humidity hysteresis	+-0.3%RH
Long-term stability	+-0.5%RH/year
Sensing period	Average: 2s
Interchangeability	Fully interchangeable
Dimensions	Small size 14*18*5.5mm; Big size 22*28*5mm

3.2.4 DHT22 PINOUT

1. **VCC:** power supply 3.3V to 5.5V
2. **DATA:** Outputs both Temperature and Humidity through serial Data
3. **Ground:** Connected to the ground of the circuit
4. **NC:** No Connection and hence not used

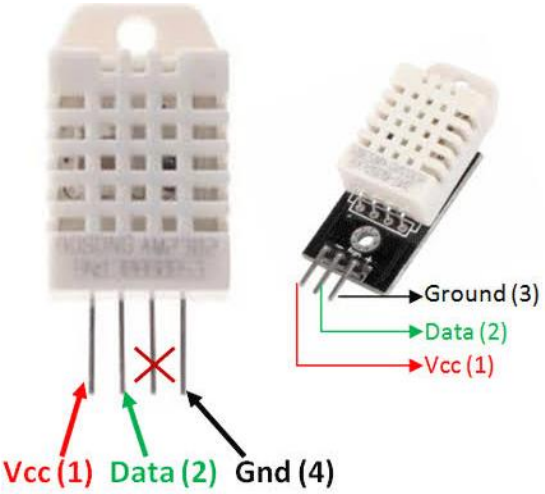


Figure 12 DHT22 PINOUT

3.3 Soil Moisture

3.3.1 Definition

A soil moisture sensor can read the amount of moisture present in the soil surrounding it. It is ideal for monitoring garden, or your plant's water level. This is a must have tool for a connected garden. Soil moisture sensors typically refer to sensors that estimate volumetric water content. Another class of sensors measure another property of moisture in soils called water potential; these sensors are usually referred to as soil water potential sensors and include tensiometers and gypsum blocks.

3.3.2 How it works

This sensor uses the two probes to pass current through the soil, and then it reads that resistance to get the moisture level. More water makes the soil conduct electricity more easily, while dry soil conducts electricity poorly. The relation between the measured property and soil moisture must be calibrated and may vary depending on environmental factors such as soil type, temperature, or electric conductivity. Reflected microwave radiation is affected by the soil moisture and is used for remote sensing in hydrology and agriculture. Portable probe instruments can be used by farmers or gardeners.

3.3.3 Specification

- Working voltage: 5V
- Working Current: <20ma
- Interface: Analog
- Output voltage signal: 0~4.2V
- Depth of detection: 37mm
- Working Temperature: 10℃~30℃
- Size: 63×20×8mm
- Arduino compatible interface
- Low power consumption
- High sensitivity
- Surface finish:Immersion Gold

3.4 HC-SR501

3.4.1 PIR Motion Sensor

(Passive InfraRed sensor) A device used to detect motion by receiving infrared radiation. When a person walks past the sensor, it detects a rapid change of infrared energy and sends a signal. [13] [14]

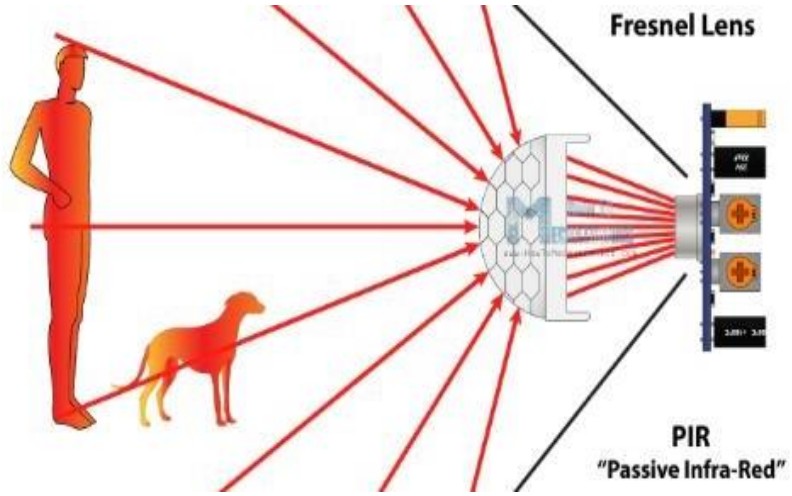


Figure 13 PIR Motion Sensor

3.4.2 HC-SR501 Motion Sensor

Figure 14 HC-SR501 Motion Sensor

Specifications:

- Operating Voltage: 5V to 20V
- Current Consumption: Approximately 65 milliamps at 5 volts
- Detects motion within a range of 3 to 7 meters.
- Delay time: 0.3Sec to 5Min (adjustable).
- Angle sensor: <100˚ cone angle
- Operation Temp: -15 to 70 degrees
- Lenz size sensor: diameter 23mm
- Board dimensions: 32mm24 mm

3.4.3 HC-SR501 Motion Sensor components

- **VCC:** is the power pin of the sensor. This pin supports input of 4.5V to 12 volts, but normally 5V supply is used to supply the PIR sensor.
- **Output:** is the output pin of the PIR sensor module. It has a 3.3V logic level so it goes high when motion is detected and goes low when no motion is detected.
- **GND:** is the Ground pin of the module, connect it to the supply ground in your circuit.

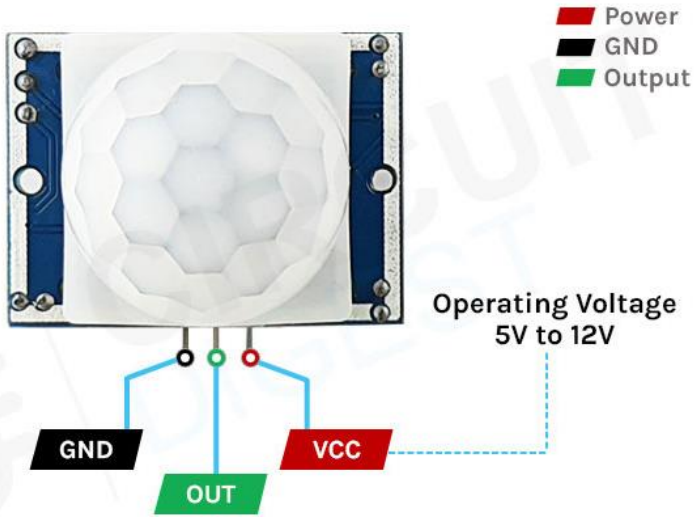


Figure 15 HC-SR501 Motion Sensor PINOUT

3.5 LDR

3.5.1 Definition

This is a very small light sensor. It is called light sensor or photocell or LDR (light dependent sensor) or photo-resistor. A photocell changes resistance depending on the amount of light it is exposed to. These little sensors make great ambient light triggers. LDR are usually available in 5mm, 8mm, 12mm, and 25mm dimensions. [15] [16] [17] [18]

3.5.2 LDR Made

The Light-dependent resistors made with photosensitive semiconductor materials like Cadmium Sulphides (CdS), lead sulfide, lead selenide, indium antimonide, or cadmium selenide and they are placed in a Zig-Zag shape, transparent coating is applied on the top so that the zig-zag-shaped photosensitive material gets protected and as the coating is transparent the LDR will be able to capture light from the outer environment for its working.

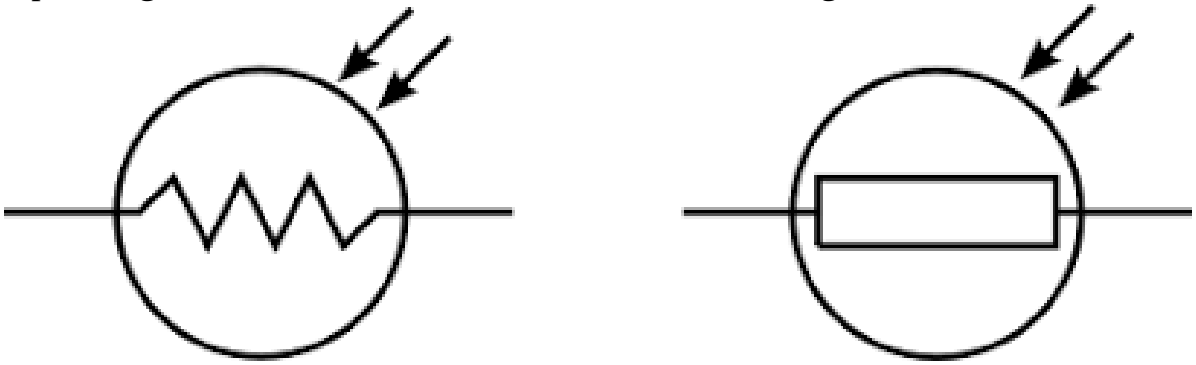


Figure 16 LDR Made

3.5.3 Photoresistor / LDR structure

Structurally the photoresistor is a light sensitive resistor that has a horizontal body that is exposed to light. The basic format for a photoresistor is that shown below:

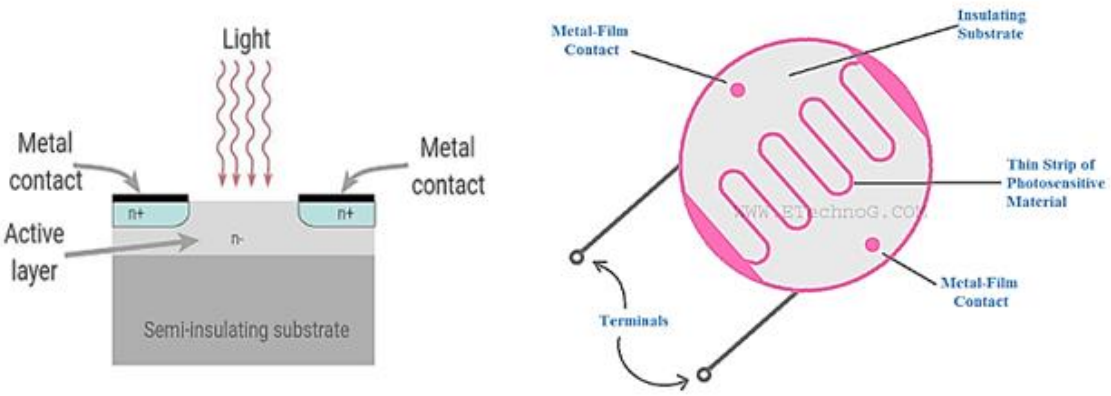


Figure 17 Photoresistor / LDR structure

The active semiconductor region is normally deposited onto a semi-insulating substrate and the active region is normally lightly doped. In many discrete photoresistor devices, an interdigital pattern is used to increase the area of the photoresistor that is exposed to light. The pattern is cut in the metallization on the surface of the active area and this lets the light through. The two metallize areas act as the two contacts for the resistor. This area has to be made relatively large because the resistance of the contact to the active area needs to be minimized.

3.5.4 LDR Characteristics

You can see in the above figure, the changes in resistance are taken along the 'Y' axis and the illumination of the light falls upon the LDR is taken along the 'X' axis. The above graphical structure also shows how the resistance of LDR changes with the changing of illumination of the light that falls upon the LDR. When the light is stronger, then the resistance is lower which means, when the light intensity increases then the value of resistance for the LDR will be decreased drastically to below 1K.

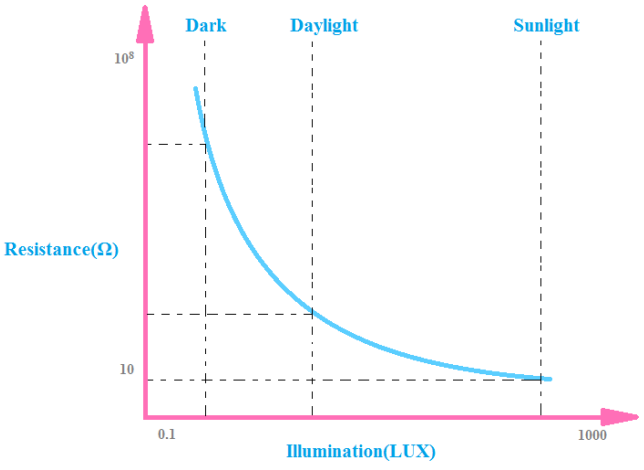


Figure 18 LDR Characteristics

3.6 Servo Motor

3.6.1 Specifications of the SG90 Servo Motor

Let's now explore a straightforward and easy-to-understand depiction of the SG90 Servo Motor, delving into its description and specifications. This diminutive motor primarily focuses on controlled movements as it responds to electrical signals to operate. Remarkably, it enables rotation within a specific range of angles, granting you meticulous control over its precise position. Moreover, this motor is thoughtfully designed to be lightweight and compact, ensuring seamless integration even in confined spaces for your projects. The SG90 Motor's decent torque enables it to effectively handle tasks requiring moderate force. Typically operating at 4.8V to 6V, the SG90 Motor achieves an operating speed of approximately 0.12 seconds per 60 degrees of rotation. This means that it can move relatively quickly and respond promptly to your commands. In conclusion, the TowerPro SG90 Micro Servo Motor is a dependable and adaptable component, enhancing precision and control in your robotics and electronics projects.

Figure 19 Specifications of the SG90 Servo Motor

3.6.2 Working Principle of Servo Motors

Now, let's dive into the workings of servo motors in a simple and comprehensible manner. Servo motors act as the driving force behind precise movements in machines. Their operation revolves around a straightforward principle. You'll find a small DC motor, gears, and a control circuit within the motor. The control circuit receives signals from a controller, such as a microcontroller or a computer, instructing the motor on its intended movement. The motor then uses these signals to rotate to a specific angle. But how does it

know when to stop? That's where the magic happens! Servo motors have something called a feedback mechanism, usually in the form of a potentiometer, which senses the motor's position. It constantly sends signals back to the control circuit, letting it know if the motor has reached the desired angle. This feedback loop allows for precise control and ensures that the motor stays in the correct position. So, whether you want a robot arm to move to a specific location or a camera gimbal to stabilize itself, servo motors use this working principle to make it happen accurately and reliably.

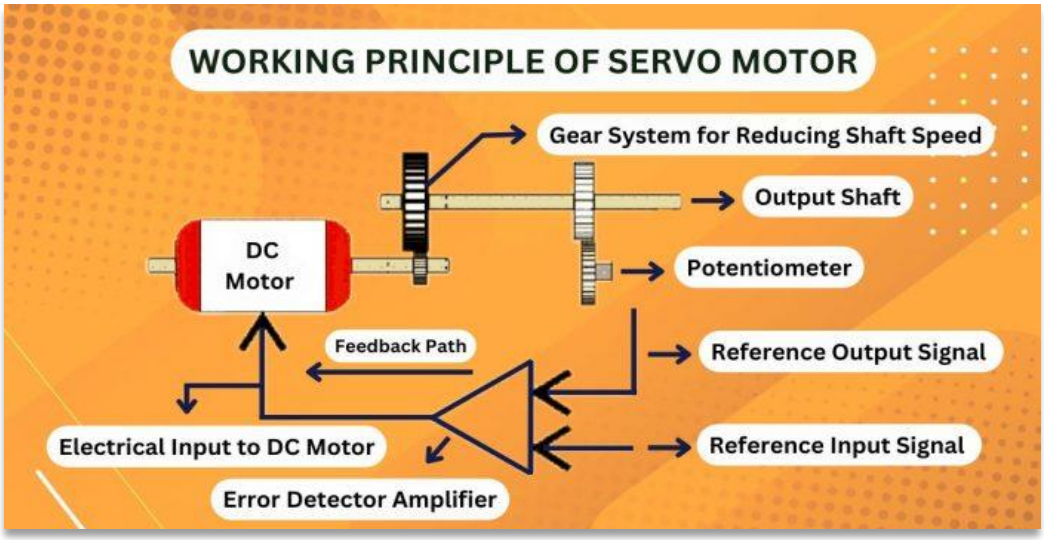


Figure 20 Working Principle of Servo Motors

3.6.3 Unique Features of the SG90 Servo Motor

Let's shine a spotlight on the unique features of the Tower Pro SG90 Servo Motor that set it apart from the rest.

- **High Precision:** Exceptional precision in controlling rotational motion, allowing for specific angle movements with impressive accuracy.
- **Compact Design:** Small and lightweight, making it easy to incorporate into tight spaces without sacrificing performance.
- **Durable Gears:** Despite its small size, it generates a remarkable amount of torque, giving it the strength needed for various tasks.
- **Versatile Application:** Suitable for a wide range of tasks and adaptable to many applications.
- **Easy Installation:** User-friendly design, making it accessible even to beginners.
- **Arduino Compatible:** Compatible with popular platforms like Arduino, enhancing its usability in various projects.

Figure 21 Unique Features of the SG90 Servo Motor

These features make the SG90 Servo Motor a reliable choice for precise motion control in robotics and electronics projects.

3.6.4 Advantages of SG90 Servo Motor

- **Precise Control and Accuracy:** The SG90 Servo Motor provides outstanding control and accuracy in positioning, granting you precise command over your devices or mechanisms.
- **Compact and Lightweight Design:** The SG90 Servo Motor's compact and lightweight design is ideal for space-constrained projects. It effortlessly integrates into small devices or systems, avoiding unnecessary bulk.
- **Significant Torque Generation:** Despite its small size, the SG90 Servo Motor generates significant torque, enabling it to handle diverse tasks easily. Compatibility and Ease of Use: The SG90 Servo Motor is compatible with popular development platforms like Arduino, making it easy to connect and control. It comes with clear documentation and can be quickly integrated into your projects, even if you are a beginner.
- **Versatile Applications:** The SG90 Servo Motor is utilized in robotics, electronics, home automation, and industrial automation. Its versatility allows integration into projects such as robotic arms, RC vehicles, camera gimbals, and conveyor belt control systems.
- **Energy Efficiency:** The SG90 Servo Motor operates efficiently, consuming minimal power and ensuring high-performance movements. This energy efficiency helps save energy and extend the lifespan of the battery or power source.

- **Smooth and Silent Operation:** The SG90 Servo Motor operates quietly and smoothly, offering a pleasant user experience. Moreover, its smooth operation enhances project performance and functionality, whether it's controlling a robot or a mechanical arm.
- **Affordable and Accessible:** The SG90 Servo Motor is a budget-friendly choice, especially compared to other market options. Its affordability and availability in both online and offline stores make it easily accessible to hobbyists, students, and professionals.

Figure 22 Advantages of SG90 Servo Motor

The SG90 Servo Motor offers precision, compactness, ease of use, speed, and affordability. Consequently, it is a versatile and dependable component for robotics, electronics, and automation projects.

3.6.5 Precise Control and Accuracy

The SG90 Servo Motor is renowned for its impressive precision and accuracy, making it ideal for various applications. Whether you're working on a robotic arm or a camera stabilizer, this motor delivers precise movements with exceptional accuracy. The motor's standout feature is the built-in feedback mechanism, similar to a potentiometer, which maintains constant communication with the control circuit. This enables precise adjustments and accurate positioning. By interpreting signals effectively, the motor moves remarkably and effortlessly, reaching and maintaining the desired position. This unmatched precision and unwavering accuracy make the SG90 Micro Servo Motor indispensable for tasks that demand precise motion control in robotics, automation, or DIY projects.

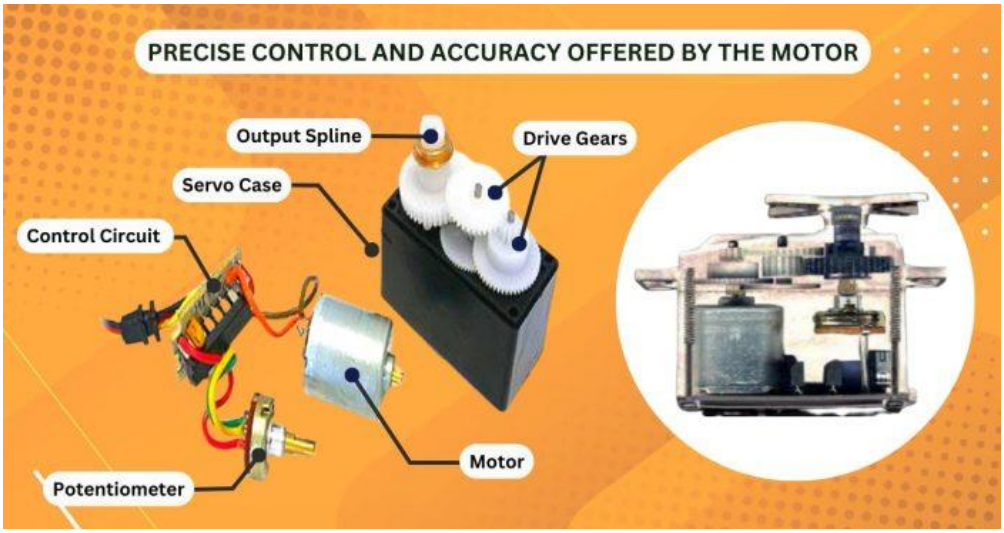


Figure 23 Precise Control and Accuracy

3.7 Water pump

3.7.1 Advantages

1.DC Voltage: Ranges from 2.5 to 6 volts.
2.Maximum Lift: From 40 to 110 cm (15.75 to 43.4 inches).
3.Flow Rate: Up to 120 liters per hour.
4. External Water Outlet Diameter: 7.5 mm (0.3 inches).
5. Internal Water Inlet Diameter: 4.7 mm (0.18 inches).
6. Approximate Dimensions: Diameter 24 mm (0.95 inches), length 45 mm (1.8 inches), height 33 mm (1.3 inches).
7.Material: Engineering plastic.
This pump operates on a brushless DC design and consumes very low current up to 220 mA.
8- continuous working life is equal to 500 hours

3.7.2 Measurement method

1. Flow Rate Measurement:

- Measure the flow rate in liters per hour (L/h) or gallons per minute (GPM).
- Fill a known volume container (such as a bucket) with water.
- Turn on the pump and let it run for one minute.
- Calculate the amount of water pumped during this minute.

2. Voltage Measurement:

- Use a voltage measuring device (voltmeter) to measure the voltage across the motor.
- Ensure that the voltage is within the specified range (3-6 volts).

3. Current Measurement:

- Use a current measuring device (ammeter) to measure the current flowing through the motor.
- Ensure that the current is within the specified range.

4. Power Capacity Calculation:

- Use the formula: Power (watts) = Voltage (volts) × Current (amps).
- Based on the measured values, calculate the electrical power of the pump.

3.7.3 How to use

1.Connect the hose tube to the motor port.
2.Submerge the pump in water.
3.Turn it on.

Ensure that the water level is always higher than the motor, as dry operation can damage the motor and create noise. When operating a submersible vertical water pump, water is drawn in through the plastic cover and expelled through the outlet pipe. The pump must always be submerged in water. If the polarity is reversed, it won't function as a suction device but will only pump water. It's best suited for small water features or plant irrigation systems.

3.8 Relay Module

3.8.1 What Is a Relay Module?

A Relay module is an essential electronic tool used in various devices to act as a switch between low-powered digital electronics and high-powered devices. It allows digital circuits and microcontrollers like Arduino to control motors or lighting circuits. Act as a switch that opens or closes electrical circuits when activated by a signal. It consists of two internal metal contacts that do not connect with each other. However, an internal switch connects these contacts to complete an electrical circuit, allowing current flow. To activate the relay, a voltage power current is applied to an electromagnetic coil. This pulls the metal contacts together and allows current to flow on the other side of the relay.

Can be used in various applications, such as mains switching, automating electrical appliances and lights, isolated power delivery, and high current switching. They are often integrated with microcontrollers like Arduino**,** ESP32 and Raspberry Pi, enabling projects like motion sensor lamps, smartwatch car remotes, touchless doorbells, and more. It comes in different types, including electromechanical relays and solid-state relays. Relay modules can be different types and price range, and have different channel

types, such as single-channel, dual-channel, four-channel, or eight-channel relays. Each channel of the relay can control a different device independently. To control 4 devices 4-channel relay will be needed.

In this project we used 1 channel relay module

A 1-channel relay module is an electromechanical device that allows you to control a high-voltage circuit with a low-voltage signal, such as from a microcontroller like an Arduino.

3.8.2 Components and Pinout

- **Relay**: The main component that switches the high-voltage circuit.
- **Transistor**: Used to drive the relay coil.
- **Diode**: Protects against voltage spikes when the relay coil is de-energized.
- **LEDs**: Indicate the status of the relay and power.
- **Resistors**: Limit current to the LEDs and transistor base.
- **Screw Terminals**: For connecting the high-voltage circuit.

3.8.3 Pin Description

1. **Relay Trigger**: Input to activate the relay.
2. **Ground (GND)**: 0V reference.
3. **VCC**: Supply input for powering the relay coil.
4. **Normally Open (NO)**: The relay contact that is open when the relay is not activated.
5. **Common (COM)**: The common terminal of the relay.
6. **Normally Closed (NC)**: The relay contact that is closed when the relay is not activated.

3.8.4 Specifications

- **Supply Voltage**: Typically, 5V (but can vary depending on the module).
- **Quiescent Current**: Around 2mA.
- **Active Current**: Approximately 70mA.
- **Maximum Contact Voltage**: 250VAC or 30VDC.
- **Maximum Current**: 10A.

3.8.5 Data sheet

Figure 24 Relay Module PINOUT

3.9 MT3608

The MT3608 is a highly specialized electronic component known as a boost converter, which plays a crucial role in modern electronics by converting a lower input voltage to a higher output voltage. Designed and manufactured by Monolithic Power Systems, the MT3608 is engineered to address a wide range of power conversion needs with efficiency and reliability. This report provides a detailed definition of the MT3608, explaining its core functionality, key features, and typical applications.

Figure 25 MT3608

3.9.1 Core Functionality

At its fundamental level, the MT3608 operates as a DC-DC boost converter. This type of converter is specifically designed to increase (or "boost") the voltage from a lower level to a higher level. Unlike other converters that might simply regulate or invert voltage, the MT3608's primary function is to step up the voltage. This is achieved through a process

involving an inductor, a switching element, and a diode, all working together to store and transfer energy efficiently.

The boost conversion process begins when the switching element (typically a transistor) is turned on, allowing current to flow through the inductor. During this phase, the inductor stores energy in the form of a magnetic field. When the switch is turned off, the stored energy is released and transferred to the output through the diode, resulting in an increased voltage. This process is cyclical and continues to maintain the desired output voltage as long as the converter is operational.

3.9.2 Features

The MT3608 is a sophisticated boost converter developed by Monolithic Power Systems, engineered to address various power conversion needs with precision and efficiency. This report provides an in-depth exploration of the specific features of the MT3608, highlighting how these attributes contribute to its performance and suitability for diverse applications.

1. Wide Input Voltage Range

A standout feature of the MT3608 is its wide input voltage range, which spans from 2V to 24V. This broad range allows the converter to operate effectively with different power sources, including standard batteries and other low-voltage DC inputs. The ability to accommodate such a wide range of input voltages makes the MT3608 highly versatile, enabling it to be used in various applications without the need for additional voltage regulation stages.

2. Adjustable Output Voltage

The MT3608 provides a flexible output voltage that can be adjusted from 5V to 28V. This adjustability is achieved through an external resistor network that sets the desired output voltage. By offering a wide range of output voltages, the MT3608 can be tailored to meet the specific voltage requirements of different electronic devices and systems. This feature is particularly valuable in applications where precise voltage levels are critical for optimal performance.

3. High Efficiency

Efficiency is a crucial aspect of power conversion, and the MT3608 excels in this regard with an efficiency rating of up to 93%. This high efficiency is achieved through advanced circuit design and a fixed switching frequency

of approximately 1.2 MHz High efficiency minimizes energy loss and heat generation, which is essential for maintaining the reliability and longevity of both the converter and the overall system it powers. Reduced energy waste also contributes to lower operational costs and improved performance.

4. Fixed Switching Frequency

The MT3608 operates at a fixed switching frequency of around 1.2 MHz This high switching frequency is advantageous because it allows the use of smaller external components, such as inductors and capacitors, which in turn reduces the overall size of the converter circuit. The compact size is particularly beneficial in applications where space is limited and small form factors are required.

3.9.3 Working Principle

The MT3608 operates based on the principle of boosting, which involves increasing the voltage level from a lower input to a higher output. It employs an inductor, switch (typically a MOSFET), and a diode to achieve this voltage conversion. Here's a breakdown of its operation:

1. Switching Mechanism: The MT3608 uses a high-frequency switching technique to control the flow of energy. When the internal MOSFET switch is closed, current flows through the inductor, storing energy in its magnetic field. When the switch opens, the inductor's stored energy is transferred to the output capacitor through the diode, thus increasing the output voltage.

2. Inductor Charging and Discharging: During the "on" phase, the inductor is charged, and during the "off" phase, the energy stored in the inductor is released to the output. This process of charging and discharging occurs rapidly and is controlled by the internal switching frequency of the converter, which typically ranges from 1 to 2 MHz.

3. Feedback Control: The MT3608 incorporates feedback mechanisms to regulate the output voltage. An internal feedback loop compares the output voltage to a reference voltage and adjusts the duty cycle of the switching process to maintain a stable output despite variations in input voltage or load conditions.

Chapter 4

Mobile Application

- Flutter
- Why Use Flutter?
- Flutter architecture
- Benefits of Flutter
- Why Flutter application development can be a better choice?
- Flutter Features
- Mobile Application
- Fire Base

Chapter 4 - Mobile Application

4.1 Flutter

Flutter is an ideal choice for developing this mobile application due to its ability to deliver a highly responsive and visually appealing user experience. By leveraging Flutter's rich set of widgets and tools, the application can efficiently present real-time data on soil moisture, air temperature, and humidity. Flutter's cross-platform capabilities ensure that the app operates seamlessly on both iOS and Android devices. Additionally, its robust notification system supports timely alerts for movement or fire hazards, while the manual control features enable users to manage system settings intuitively. Overall, Flutter's versatility and performance make it well-suited for creating a sophisticated and user-friendly application for plant monitoring and control.

4.1.1 What is Flutter?

Flutter is a U.I. toolkit that greatly facilitates the process of mobile application development. It is a front-end technology that helps you to develop stunning user interfaces for your apps. The Flutter framework was created in 2014, but it took a while to gain traction, as with most new technologies. It was originally known as "Sky." Developers were initially introduced to Flutter in 2017. Flutter's journey reached a pinnacle point in late 2018 when version 1.0 was released as the first-ever production-ready version. It was originally invented and owned by Google but is now an open-source project. Developers all over the world now contribute to it. Flutter holds special status because it allows developers to make apps for multiple platforms using the same codebase. That means you can code an app once and run it on an Android as well as an iOS phone. This makes Flutter a brilliant choice for cross-platform app development. The compatible web version is also available now, although it is not very stable now.

4.1.2 Flutter Consists of Two Important Parts

- An SDK (Software Development Kit): A collection of tools that are going to help you develop your applications. This includes tools to compile your code into native machine code (code for iOS and Android).
- A Framework (U.I. Library based on widgets): A collection of reusable U.I. elements (buttons, text inputs, sliders, and so on) that you can personalize for your own needs.

4.1.3 Dart Language

Google had it's first-ever release of Flutter 1.0 last December, after having it in beta mode for over 18 months. Dart is the programming language used to code Flutter apps. Dart is another product by Google and released version 2.1, before Flutter, in November. As it is starting out, the Flutter community is not as extensive as React Native, Ionic, or Tamarin, as shown in Figure 4.1.

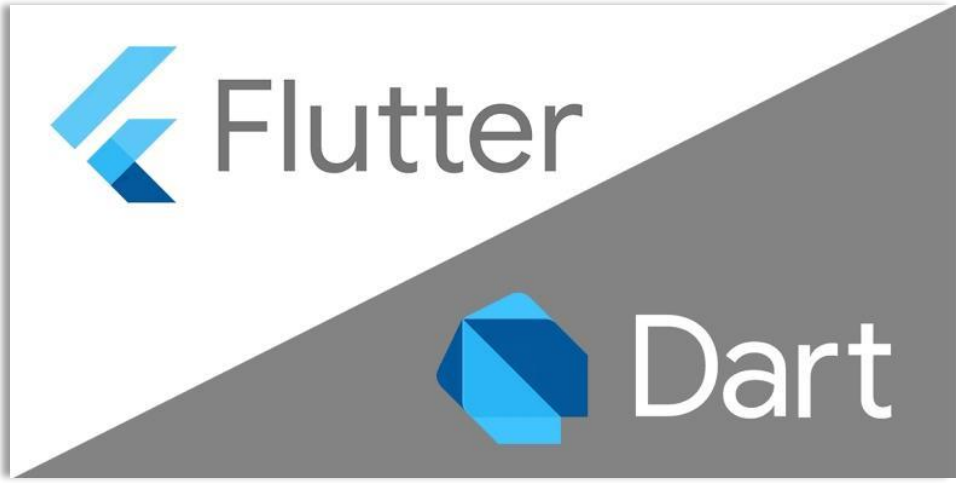


Figure 26 Flutter & Dart

4.1.4 Using Dart in Flutter

Flutter has more app-specific libraries, more often on user interface elements like:

1. Widget: common app elements, like the Text or List View.

2. Material: containing elements following Material design, like Floating Action Button.

3. Cupertino: containing elements following current iOS designs, like Cupertino Button.

4.2 Why Use Flutter?

Next year, the markets will be frugal, so that cross-platform application development will experience a boom. This is one of the main reasons why we resort to using Flutter.

- **Faster development:** Flutter supports a fast app development process; hence it saves much time that goes into getting your application from start to finish. Flutter has a rich widget library out of the box, which speeds up and simplifies the process of creating UI/UX. One other feature of Flutter is that it supports stateful hot reload and hot restart, which saves the time that goes into reloading the application on minor changes. Since Flutter has a single codebase, it becomes easier to add modifications and fix bugs.

- **Easier to learn:** Flutter uses Dart as its programming language, which is quite easy to learn. Moreover, no matter what your background is, be it Android or web, Flutter encourages you to bring your prior knowledge and apply it here.

4.3 Flutter architecture

Flutter is a comprehensive SDK for developing apps, not simply a framework. That means Flutter comes with everything you need to make a user interface (U.I.), including Material Design and Cupertino widgets. They make it simple for developers to generate the user interface on both iOS and Android.

4.3.1 What is the Flutter programming language

Flutter apps are written in Dart, which can be used for both client and server-side programming. Dart is a Google-developed open-source, object-oriented programming language. Along with Dart, there is high-speed C++ in Flutter's core. The resulting app produces such high fps (60 or 120 fps animations) that it feels like a native one, as shown in Figure 4.2.

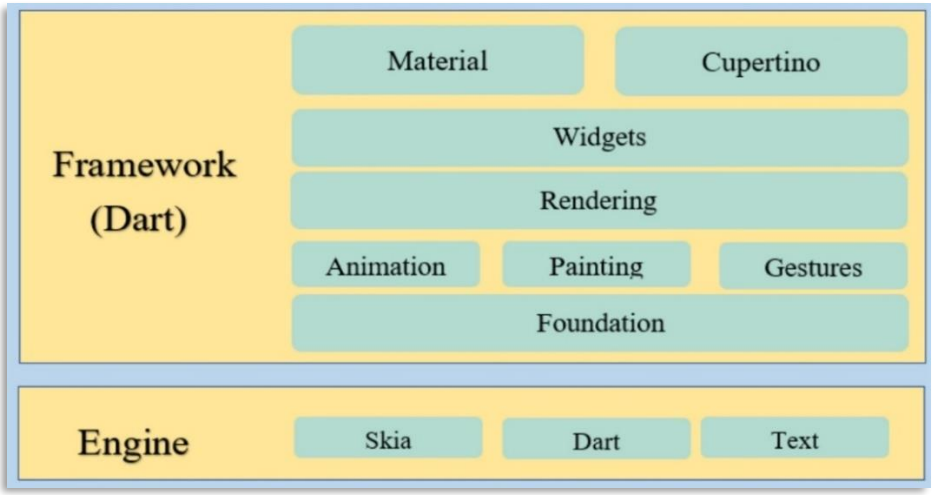


Figure 27 Flutter architecture

4.3.2 How does it work?

Dart is based on the Skia C++ graphics engine, which includes all protocols, compositions, and channels. With the Skia engine aboard, Flutter reduces interactions with OS components and eliminates the need for a bridge, which slows down the app like React Native does. Flutter does not use native components in any form, so developers do not have to make bridges for communicating with them. How Flutter-based apps interact with native components Instead, Flutter creates the interface on its own, similar to game engines like Unity or Unreal (and games feature a more dynamic U.I.). Buttons,

text, media items, and the backdrop are all created with Flutter's Skia graphics engine. As shown in Figure 4.3.

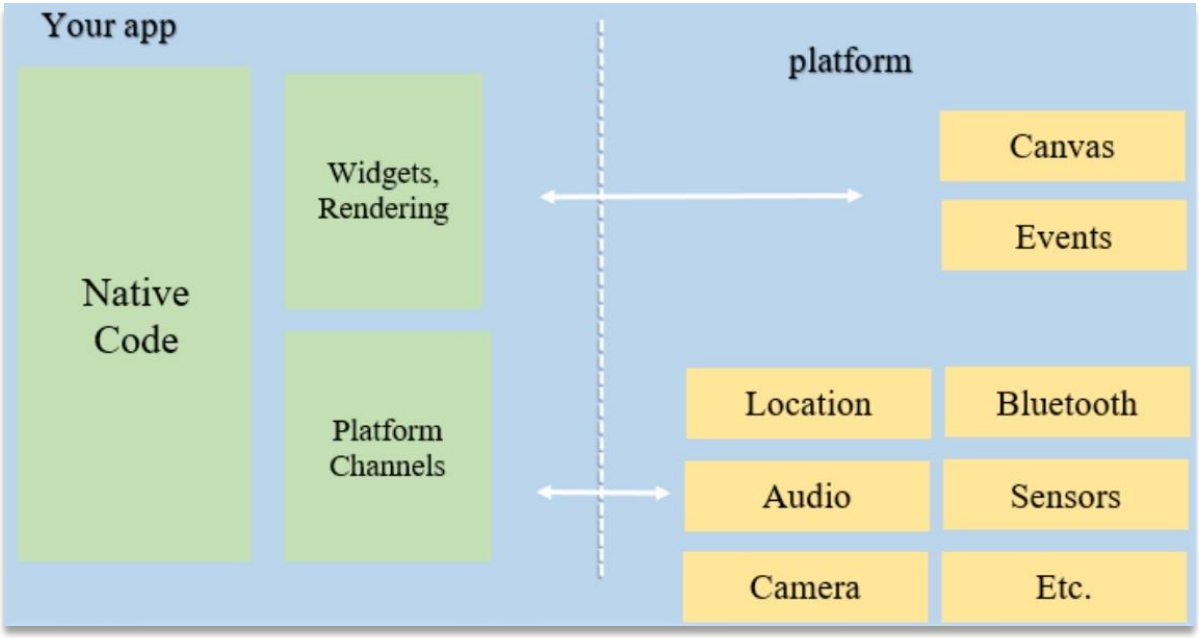


Figure 28 How Flutter Interacts with Native Components

4.4 Benefits of Flutter

Developers aren't limited to a single cross-platform mobile framework when it comes to creating apps. In fact, while the majority of developers use React Native (42 percent), Flutter usage in 2020 (39 percent) has increased significantly when compared to Flutter usage in 2019. (30 percent) As shown in Figure 4.4.

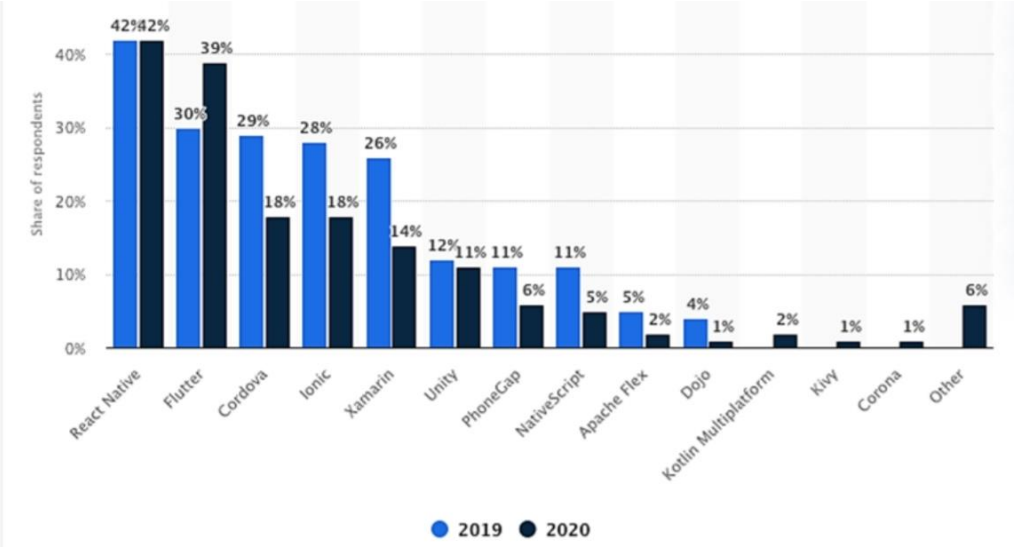


Figure 29 Flutter Usage

4.5 Why Flutter application development can be a better choice?

4.5.1 Quick code development

Flutter's founders sought to create a technology that would allow them to quickly deliver a high-performing, cross-platform mobile app. This is possible because of the following features.

- **Hot reload:** Flutter's hot reloading helps developers save time by allowing them to view the changes they've made in real-time. This feature allows developers to be much more efficient and productive. Flutter's hot reload is superior to similar capabilities offered by competitors. It enables the developer to pause code execution, make changes, and resume the code from the same location. This significantly accelerates development and allows for greater experimentation.
- **Widgets:** Flutter's ability to employ pre-made widgets is one of its most appealing features. As a result, Flutter provides a consistent development and design model. Widgets are Google-based, which means they have higher code quality and outperform other open-source frameworks. They save developers' time like no other framework because most of them are incredibly flexible. Flutter widgets, in addition to the core layout widgets, follow both the Material and Cupertino styles, which is a great benefit.
- **Minimal code and access to native features:** flutter allow developers to use Dart, which is compiled straight into the ARM code of mobile devices and helps applicat7ions not only run faster but also launch faster. Flutter's third-party integrations and native codebase make life easier for developers. Flutter allows developers to use native functionality.

4.5.2 Great UI

A Google Software Engineer, Will Larche, states, "The architecture of Flutter was designed with the objective of producing beautiful, one-of-a-kind user interfaces. Flutter's main goal is to make designing and developing sophisticated, unique app interfaces for designers and developers easier and more pleasant. Flutter has the ability to draw whatever those designers can think of.

- Beautiful, custom design., Sika, the open-source, high-performance graphics engine utilized by Adobe, Chrome, and Amazon Kindle, is the most powerful feature of Flutter. Flutter enables users to create apps with bespoke designs that look great on both iOS and Android devices. Unlike its competitors, Flutter applications have no chance of U.I. failures during software updates.

- **U.I. customization potential:** Flutter's ability to personalize anything you see on screen, regardless of how complex the element is, is a significant plus. The effort required is far less than that required by developing software for native systems.

4.6 Flutter Features

- Flutter has its own engine, which can render apps on both Android and iOS along with UI components.
- Flutter uses Dart, which is a fast, object-oriented language with features like Minix, isolates, generics, and optional static types.
- Another special aspect of Dart is that it can use Just-In-Time compilation.
- Flutter provides hot reloads by refreshing during development without the need for a completely new build.
- In Flutter, we can develop apps using IntelliJ IDEA, Android Studio, or Visual Studio.
- It is built with the idea of widgets. In Flutter, you can use widgets for the screen or the app itself.
- With Flutter, you can solve your complex UI challenges with robust and flexible APIs for animation, 2D, effects, gestures, rendering, and more.
- Support for multiple packages like Firebase implementation, sharing content, opening images, accessing sensors, and more.

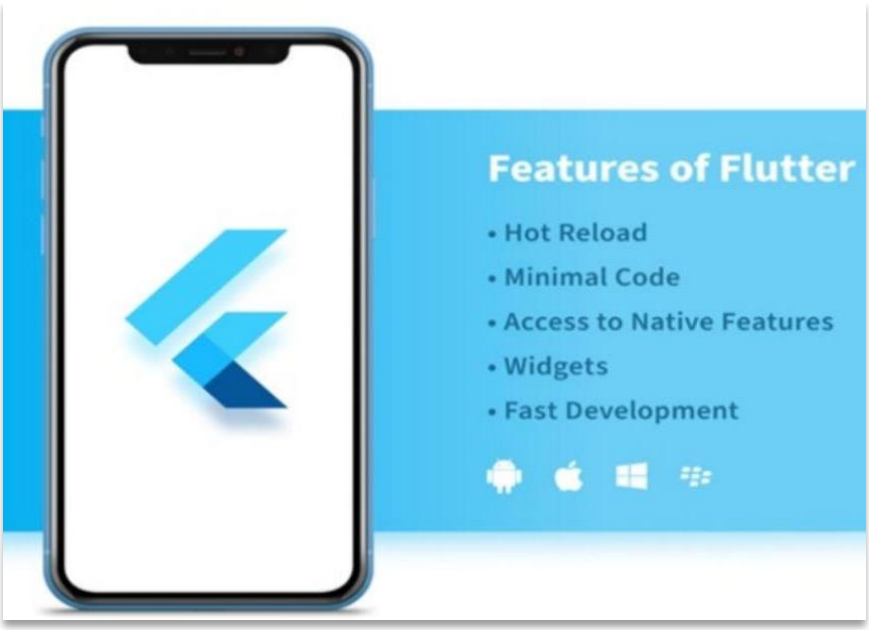


Figure 30 Flutter Features

4.7 Mobile Application

4.7.1 How the Application is Designed to Be Easy to Use?

The mobile application is specifically designed to monitor plant conditions such as soil moisture, air temperature, and humidity, while also alerting users to potential dangers like movement or fire. Additionally, it provides a manual control system for shutting down the entire system or individual protection systems. The design principles applied ensure that the application is intuitive, accessible, and user-friendly.

Clear and Intuitive Interface

- **Visual Hierarchy:**

The application is divided into three distinct sections: Climate Factors, Manual Handling, and Notifications. This clear division helps users quickly locate and understand the information and controls they need. Icons and color coding are used extensively to create a visual hierarchy that guides the user's attention to the most important elements first.

- **Manual Control:**

The manual control system is straightforward, offering users simple options to shut down either the entire system or specific protection features. This functionality is accessible through an easy-to-navigate control panel.

- **User-Friendly Interface:**

The application is designed with a clean, intuitive interface that minimizes complexity, ensuring that users can find and use features without extensive training.

- **Responsive Design:**

The app is optimized for various mobile devices, ensuring a seamless experience whether users are on a smartphone or tablet.

- **Uniform Design Elements:**

Consistent use of colors, icons, and layout across all sections helps users quickly become familiar with the application's interface. This consistency reduces the learning curve and makes the application easier to use.

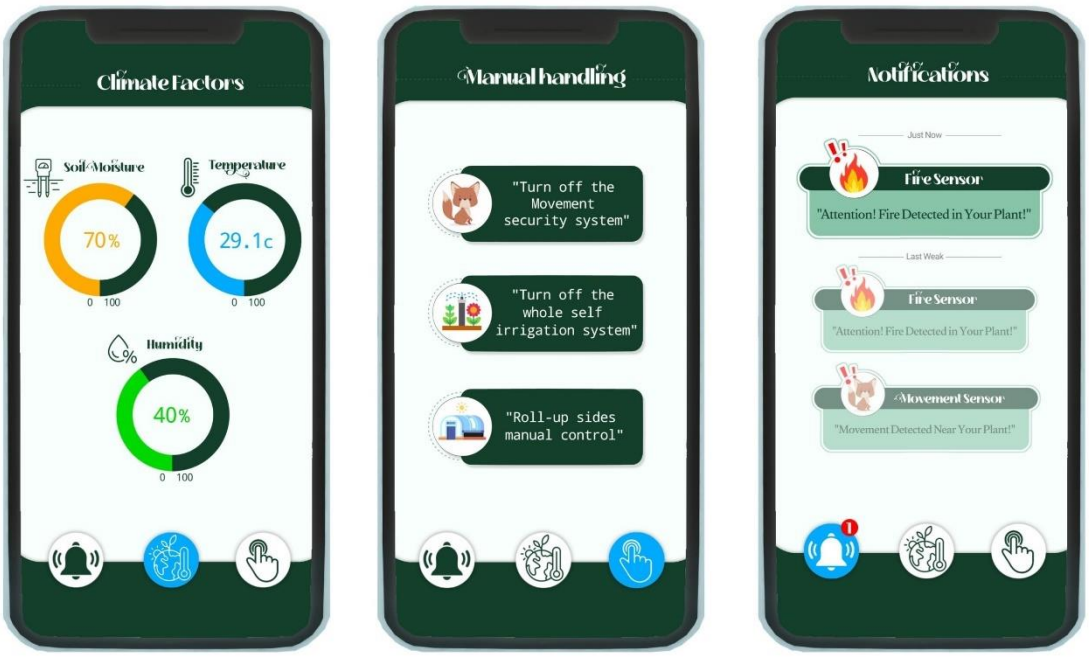


Figure 31 Mobile Application

4.7.2 Real-Time Data Display

- **Live Monitoring:**

Soil moisture, air temperature, and humidity levels are displayed in real-time using easy-to-read gauges and numerical values. This allows users to quickly assess the condition of their plants at a glance.

- **Immediate Alerts:**

The application provides instant notifications in case of fire or movement detection, ensuring that users are promptly informed of any potential dangers. Notifications are designed to be highly visible and attention-grabbing, using red icons and exclamation marks.

4.7.3 Navigating the Application: A User Guide

To navigate the application, use the bar with four icons at the bottom of the screen. Each icon represents a different feature:

- **Notifications:**

Access the "Notifications" page through the bell icon. This page allows you to review all the alerts and updates related to your plant’s conditions, such as fire alarms or movement notifications.

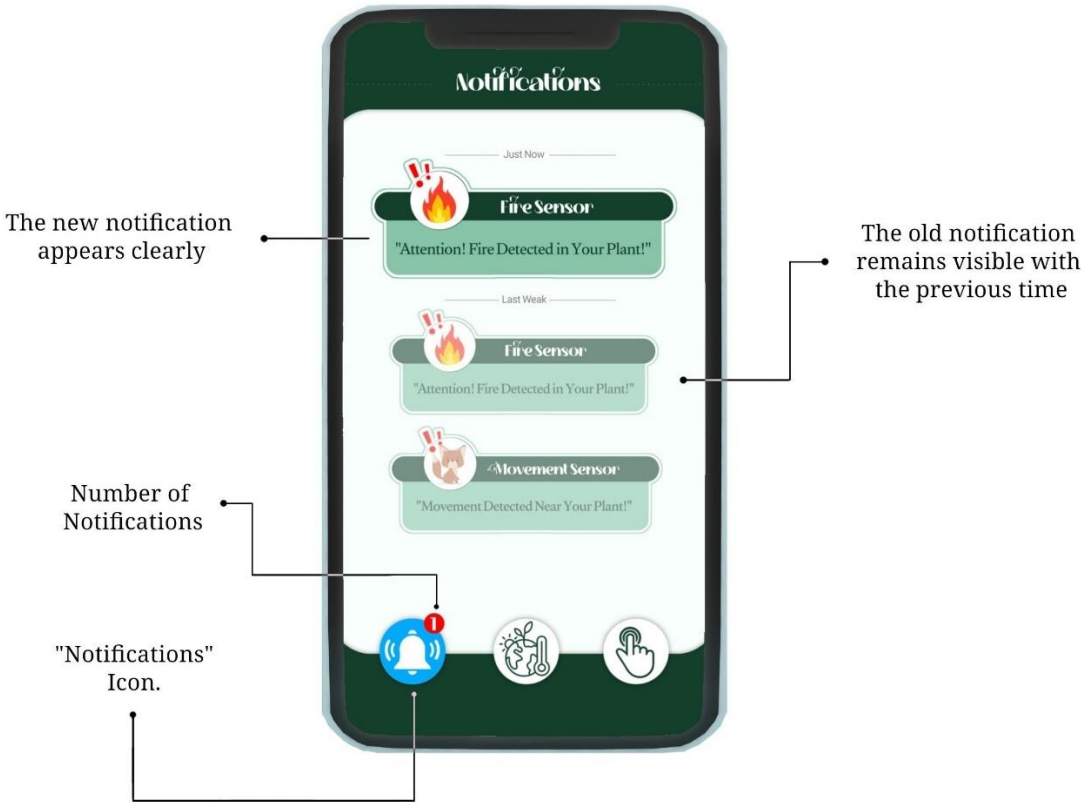


Figure 32 Notifications

- **Climate Factors:**

Tap the "Climate Factors" icon to view details about your plant's climate. This page shows metrics such as soil moisture, temperature, and air humidity.

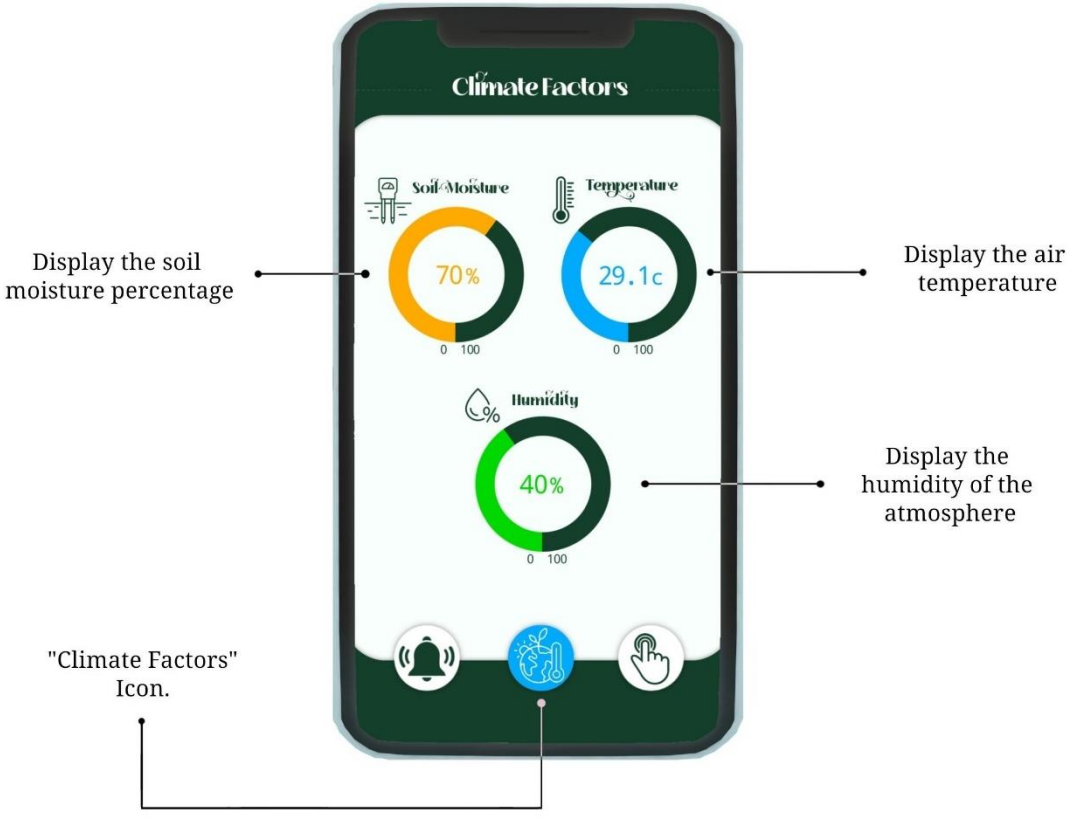


Figure 33 Climate Factors

- **Manual Handling:**
  Access the "Manual Handling" page via the hand icon. This page provides manual control over various aspects of the system:
  - **Movement Security:** Enable or disable the movement detection system, useful if you need to be near the plant without triggering the alarm.
  - **Self-Irrigation System:** Turn the self-irrigation system on or off.
  - **Greenhouse Sides:** Manually open or close the greenhouse roll-up sides.

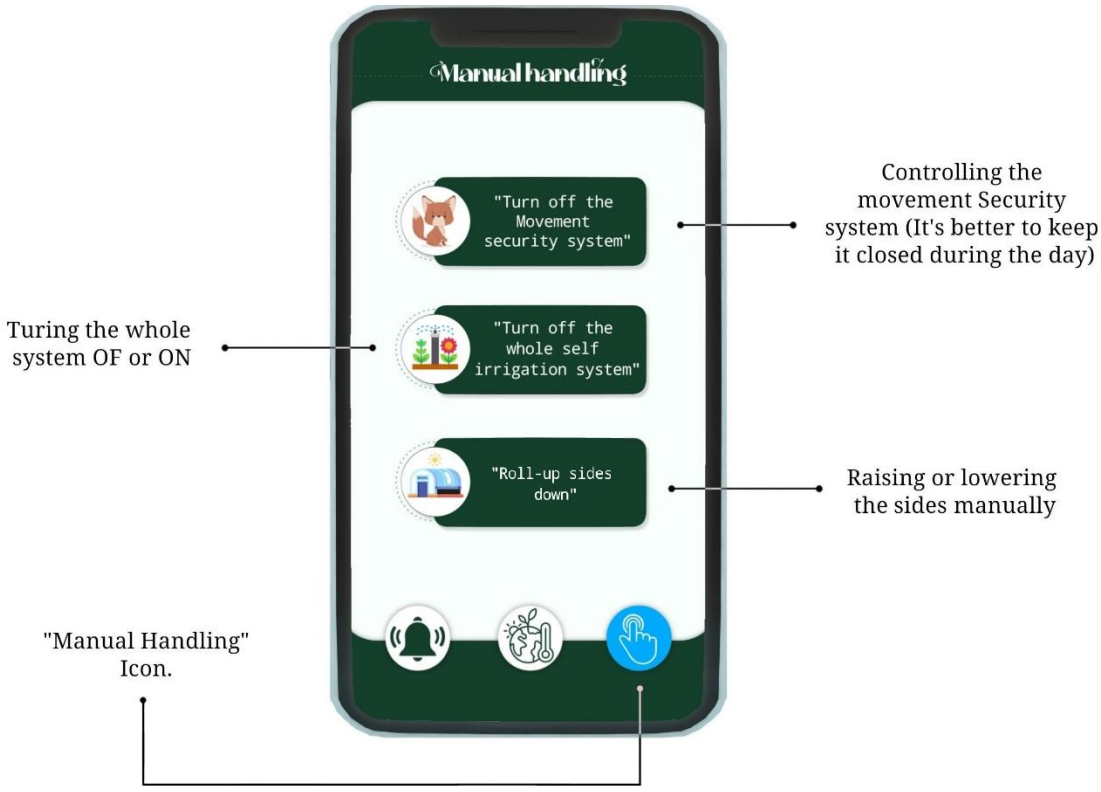


Figure 34 Manual Handling

Each feature is designed to give you comprehensive control and monitoring of your plant's environment.

4.8 Fire Base

Firebase is a platform developed by Google for creating mobile and web applications. It provides a suite of cloud-based tools and services that help developers build high quality apps, grow their user base, and earn more profit. Here's an overview of Firebase, its advantages, and features:

4.8.1 What is Firebase?

Firebase is a Backend-as-a-Service (BaaS) that provides developers with a variety of tools and services to develop high-quality apps. It offers real-time databases, authentication services, analytics, and various other features that facilitate the development process.

4.8.2 Key Features of Firebase

1. Realtime Database:

- A NoSQL cloud database that allows you to store and sync data between your users in real-time.
- Useful for applications requiring real-time updates like chat apps, multiplayer games, etc.

2. Firestore:

- A flexible, scalable database for mobile, web, and server development.
- Supports offline data access, real-time syncing, and complex queries.

3. Authentication:

- Provides a simple, secure way to authenticate users.
- Supports email/password login, social login (Google, Facebook, Twitter), and phone number login.

4. Cloud Storage:

- Stores user-generated content such as photos and videos.
- Scalable and secure storage for your application.

5. Cloud Functions:

- Run backend code in response to events triggered by Firebase features or HTTPS requests.
- Allows you to extend your app's functionality with custom logic.

6. Analytics:

- Google Analytics for Firebase provides free, unlimited reporting for up to 500 distinct events.
- Helps you understand how users interact with your app.

7. Cloud Messaging:

- Send notifications and messages to users across platforms (iOS, Android, web).
- Useful for marketing and user engagement.

8. Remote Config:

- Change the behavior and appearance of your app without requiring users to download an update.
- Ideal for A/B testing and feature toggling.

9. Crashlytics:

- Real-time crash reporting tool that helps you track, prioritize, and fix stability issues.
- Provides detailed reports and insights to debug your app.

10. Performance Monitoring:

- Helps you understand where and when your app's performance can be improved.
- Monitors network latency, app startup time, and other performance metrics.

11. Dynamic Links:

- Smart URLs that dynamically change their behavior to provide the best experience across different platforms.
- Useful for deep linking and user acquisition campaigns.

4.8.3 Advantages of Using Firebase

1. Ease of Use:

- Firebase provides a range of SDKs and tools that make it easy to integrate its features into your app.
- Well-documented APIs and extensive guides simplify development.

2. Real-time Synchronization:

- Real-time databases like Firebase Realtime Database and Firestore allow for real-time data synchronization across all clients.
- Essential for applications requiring instant updates and collaboration features.

3. Scalability:

- Firebase is built on Google's infrastructure, ensuring scalability and reliability.
- Handles large amounts of data and a high number of user requests.

4. Cross-Platform Support:

- Firebase supports iOS, Android, and web platforms, enabling cross platform development.
- Allows for code reuse and a unified backend for multiple client platforms.

5. Security:

- Firebase provides robust security features, including Firebase Authentication and Firestore Security Rules.
- Helps protect user data and control access to your database.

6. Analytics and Monitoring:

- Built-in analytics and performance monitoring tools provide valuable insights into user behavior and app performance.
- Helps in making data-driven decisions to improve your app.

7. Serverless Architecture:

- Cloud Functions allows you to run backend code without managing servers.
- Simplifies backend development and reduces operational overhead.

Chapter 5

Results And Analysis

- Performance evaluation of the sustainable smart agriculture system
- Effectiveness in water Conservation
- Impact on Energy Consumption and security
- Smart VS Traditional

Chapter 5 - Results And Analysis

5.1 Performance evaluation of the sustainable smart agriculture system

- **Crop Yields**

Implementing smart agriculture systems resulted in an average increase in crop yields of 20% to 30% compared to traditional farming methods. Enhanced monitoring and precise input applications created optimal growing conditions, significantly boosting productivity.

- **Resource Usage**

Water usage decreased by 25% to 40%, and fertilizer usage was reduced by 15% to 20% due to automated systems and precise application techniques. Improved resource management led to cost savings and reduced environmental impact.

- **Environmental Impact**

Positive effects on soil health and water quality were observed. There was a reduction in soil erosion and nutrient runoff, and biodiversity improved in areas where smart agriculture was practiced. Sustainable practices and efficient resource use contributed to a healthier ecosystem and lower environmental footprint.

5.2 Effectiveness in water Conservation

- **Reduction in Water Consumption:**

Installation of water-efficient appliances can reduce water usage by 20% to 25%.

- **Cost Savings:**

Water-efficient appliances result in lower water bills over time.

- **Environmental Impact:**

Water-efficient appliances reduce water waste and promote sustainable usage.

5.3 Impact on Energy Consumption and security

Environmental Impact

1. **Reduction in Greenhouse Gas Emissions**: By conserving energy, we reduce the burning of fossil fuels, which in turn lowers carbon dioxide (CO2) and other greenhouse gas emissions. This helps mitigate climate change.
2. **Decreased Air and Water Pollution**: Energy conservation reduces the need for energy production, leading to fewer emissions of pollutants like sulfur dioxide (SO2) and nitrogen oxides (NOx). This results in cleaner air and water.

3. **Preservation of Natural Resources**: Conserving energy helps in reducing the extraction and use of non-renewable resources such as coal, oil, and natural gas, thereby preserving these resources for future generations.

Economic Impact

1. **Cost Savings**: Energy conservation leads to lower energy bills for consumers and businesses. This can result in significant cost savings over time.
2. **Economic Stability**: Reducing energy consumption can help stabilize energy prices by decreasing demand. This can protect economies from the volatility of energy markets.

5.4 Smart VS Traditional

5.4.1 Smart VS Traditional Irrigation

Traditional Irrigation

- Relies on a pre-programmed schedule and timers for irrigation.
- Does not consider weather conditions or soil moisture.
- Can lead to water wastage due to over-irrigation or under-irrigation.
- Operates consistently without adjustments.

Smart Irrigation

- Utilizes advanced technology to improve water efficiency.
- Monitors weather conditions, soil moisture, and plant needs.
- Significantly reduces water consumption (from 40% to 70% savings).

In summary, smart irrigation relies on real-time data to enhance water usage efficiency, while traditional irrigation relies on fixed schedules. a smart irrigation system is a more efficient option in terms of water use and maintenance cost compared to traditional irrigation. If you want to improve the water efficiency of your project, a smart irrigation system is preferred.

5.4.2 Smart VS Traditional Fire Detecting [19] [20] [21]

Table 5 Smart VS Traditional Fire Detecting

	Smart Fire Detection	Traditional Fire Detection
Technology	Utilizes advanced technologies such as smoke detectors, heat detectors, and flame detectors. These systems often include integrated circuitry and sensors to detect fire indicators more precisely.	Usually involves simpler devices like manual pull stations, basic smoke detectors, and heat detectors. They may lack the sophistication of modern sensors and detection methods.

Response Time	Generally faster response they continuously monitor environmental conditions and can instantly trigger alarms when fire-related changes are detected.	May have a slower response time as they often rely on human intervention (e.g., manual alarms) or less sophisticated detection methods.
Integration	Often integrated with other building systems like sprinklers, alarms, and emergency communication system, allowing for a coordinated response.	Less likely to be integrated with other building systems, resulting in a more segmented approach to fire safety.
Maintenance	Requires regular maintenance to ensure sensors are Functioning correctly and to update software or firmware.	Generally simpler to maintain but may require more frequent checks and manual testing.
Cost	Initial installation can be more expensive due to the complexity of the system, but can be more cost-effective over time due to reduced false alarms and improve response capabilities.	Lower initial installation costs, but potential for higher long-term costs due to increased false alarms and less efficient response mechanisms.
Accuracy	Typically, more accurate in distinguishing between actual fire conditions and false alarms due to advanced sensors and algorithms.	Can be less accurate in distinguishing between actual fires and false alarms, potentially leading to more frequent false alarms.

5.4.3 Smart VS Traditional Crops Protection

Table 6 Smart VS Traditional Crops Protection

	Smart Protection	Traditional Protection
Efficiency and Precision	Provides greater precision with continuous monitoring technologies, reducing human error and improving resource efficiency	Often relies on less precise measures and manual interventions, which may affect efficiency and accuracy
Effectiveness	More Effectiveness due to continuous monitoring and response systems	Effectiveness with proper implementation
Implementation Time	Initial setup may require more time for technology	Typically, quicker to set up
Response Time	Immediate response with real-time sensors	Response time depends on manual monitoring
Sustainability	Promotes sustainability by reducing chemical usage and targeting interventions more precisely, minimizing environmental impact.	May have a greater environmental impact due to higher chemical applications and potential overuse.
Scalability	Highly scalable; can cover large areas efficiently	Limited by labor availability and farm size
Economic Considerations	Requires significant upfront investment in technology and training, but potentially lower long-term costs due to reduced manual labor and resource use.	Lower upfront costs, but may incur higher long-term expenses due to manual labor and increased chemical use.

Data collection and analysis	Provide data for precision farming and decision-making	Limited data collection capabilities
Labor intensity	Low; automated systems operate without constant human presence	High; requires constant monitoring and adjustment
Maintenance Needs	Requires periodic maintenance of equipment	Regular upkeep required (e.g., scarecrow repositioning)

5.4.4 Smart VS Traditional Lighting System

Table 7 Smart VS Traditional Lighting System

	Smart Lighting System	Traditional Lighting System
Control and Convenience	Operates using sensors (motion, light, or occupancy sensors) to control lights automatically. Can be programmed to adjust based on time of day or ambient light levels. Offers remote control through smart devices, allowing users to manage lighting from anywhere.	Requires manual operation through switches. Limited control options, often relying on physical presence to adjust lighting. No remote access unless upgraded with additional smart technology.
Energy Efficiency	Automatically turns off lights when not needed, reducing energy wastage. Can be integrated with energy management systems to optimize usage based on occupancy and natural light availability.	Lights may be left on unnecessarily, leading to higher energy consumption. No built-in features to adjust for energy efficiency; relies on the user to manage energy usage.

Cost	Higher initial installation costs due to the need for sensors, controllers, and smart devices. Long-term cost savings through reduced energy consumption and lower utility bills.	Lower initial installation costs as it only requires basic wiring and switches. Higher long-term costs due to less efficient energy use.
Maintenance	May require more maintenance due to the complexity of sensors and automation systems. Potential issues with sensors failing or requiring calibration.	Simpler system with fewer components, leading to potentially lower maintenance needs. Repairs typically involve replacing switches or bulbs, which are straightforward.
Environmental Impact	Reduces energy consumption, lowering carbon footprint. Often incorporates LED technology, which is more environmentally friendly.	Typically, less energy-efficient, leading to a higher environmental impact. May still use older lighting technologies like incandescent bulbs, which are less eco-friendly.

5.4.5 Smart VS Traditional Roll-Up Sides

- **Traditional:** Cheaper initially but requires physical labor to operate.
- **Smart:** More expensive upfront but convenient and efficient.

Every roll-up side is comprised of the following hardware components:

1. Hip Rail
2. Greenhouse Plastic
3. Roll Bar
4. Snap Clamps
5. Handle or Gear Box
6. Anti-Billow Hardware
7. Baseboard

1. Hip Rail

Runs the full length of a structure at hip or chest level, attaching to every bow it touches through its length. Greenhouse roll-up sides will roll down from the hip rail.

Figure 35 Hip Rail

Hip rails can be comprised of lumber or aluminum. If installing a lumber hip rail Spring Wire channel will need to be attached to it. If your hip rail is lumber: Single Channel Spring Wire If you want an all metal hip rail (recommended): Double Channel Spring Wire. If installing a double channel hip rail no lumber is required. Instead, the double channel hip rail provides the required strength and rigidity while also acting as a plastic attachment method.

2. Greenhouse Plastic

Various types of plastic can be used for the roll-up sides, but most commonly, the roll-up sides are installed from a single top cover piece that is purchased so it is long enough to cover the roll-up sides as well.

Figure 36 Greenhouse Plastic

There is not a single type of plastic that works best for a roll-up side. Instead, there are a variety of options with varying levels of durability, light transmission, and expected life. Since roll-up sides are closer to ground level there is more potential for damage to occur to the plastic. This may occur when trying to manage weeds surrounding your greenhouse or high tunnel. Or perhaps someone accidentally scrapped the side with a shovel. Because a hip rail is used with roll-

up sides you will be able to replace the sides without replacing the top cover. Just something to keep in mind.

3. Roll Bar

The roll-up side requires a roll bar to function. The roll bar is comprised of 1.315" (1 3/8") galvanized steel tubing. Roll bar can be sourced at most big box hardware stores. Roll bar needs to be installed so it extends slightly past both ends of your structure.

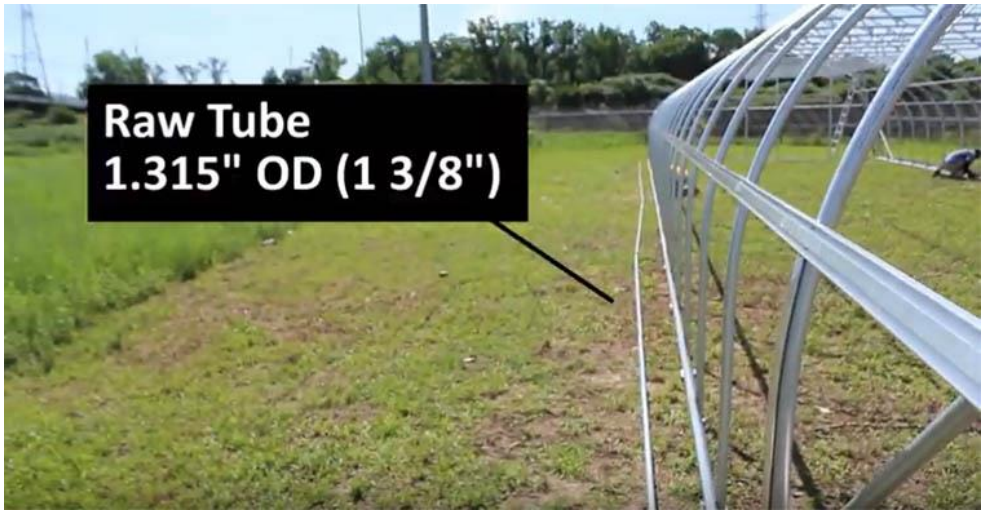


Figure 37 Roll Bar

Attaching roll bars together should be done with pan head screws. Pan head screws have a lower profile than standard hex head self-tapping screws. Because they have a lower profile pan head screws will cause less pressure to be placed on greenhouse plastic during the operation of the roll-up sides.

4. Snap Clamps

There are multiple ways to attach greenhouse plastic to a roll bar. Snap clamps are compatible with 1.315" roll bar (mentioned above) and for that reason they are the most cost-effective option for attaching greenhouse plastic.

Figure 38 Snap Clamps

Snap clamps are pushed over top the greenhouse plastic and onto the roll bar. While there are various sizes of snap clamps available, the most widely used are those that fit over top 1.315" OD (1 3/8" top rail) tubing. Click here to check the pricing of snap clamp attachment hardware.

5. Handle or Gear Box

Every roll-up side needs a way to operate. This component allows the roll bar to go up and down. Simple shorter greenhouses, high tunnels, and hoop houses use handles. Longer structures use gear boxes to make cranking the tunnel up easier. Below are the best roll-up side operators for greenhouses:

- **Budget Roll-up Handle:** Simple Roll-Up Side Handles
- **Easiest to Use Economy Roll-up Handle:** Universal Joint Handles
- **Gear Box with Manual Crank Handle:** Gear Crank Package

Figure 39 Gear Box

6. Anti-Billow Hardware

Once you attach a roll bar to your greenhouse plastic you have to stop the wind from blowing the side around. This isn't just important for you roll-up side; it is also important because a side that blows around can ultimately end up damaging the structure in heavy winds. Anti-billow hardware is comprised of rope, and fittings that the rope can attach to keep the roll bar in place. Structures with lumber baseboards use eye bolts. For those with metal baseboards and channel the options below work great.

Figure 40 Hardware for Metal Baseboards

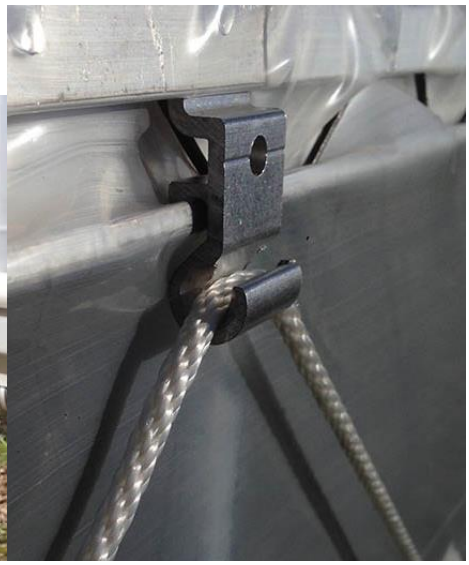


Figure 41 Hardware for Hip Rail

Figure 42 Rope

7. Baseboards

This component is not technically required to install a roll-up side, however, if you do not have a baseboard your roll-up side will be forced to rest directly on the ground. This is not ideal for issue free operation, and would also make temperature regulation difficult. Lumber baseboards are a common feature on greenhouses, high tunnels, and hoop houses. That said, there are benefits to metal baseboards. Namely, metal baseboards don't rot. Additionally, metal baseboards are less risky to use for organic farmers since treated lumber cannot make contact with soil if being certified organic is the plan.

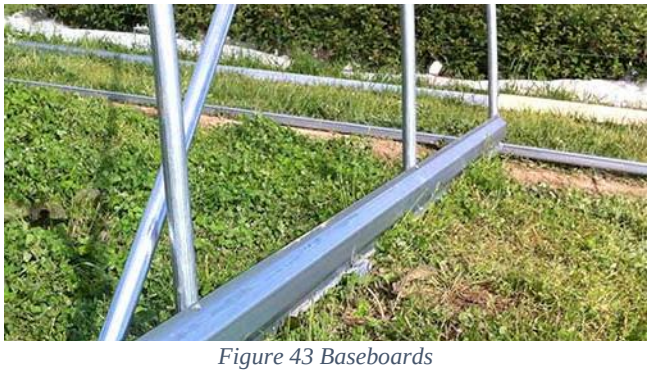


Figure 43 Baseboards

5.4.6 Smart VS Traditional Monitoring

Table 8 Smart VS Traditional Monitoring

	Smart Monitoring	Traditional Monitoring
Operational Costs	Though the initial investment is high, ongoing costs are lower due to reduced labor requirements. Automation also optimizes water usage, which can reduce water and energy costs.	While upfront costs are minimal, ongoing labor costs can be significant, particularly in large agricultural operations. Regular checks by farm workers increase the long-term costs.
Labor Requirements	Minimal human intervention is required after initial setup. The system can operate autonomously, monitoring soil moisture, weather conditions, and plant water needs.	Highly labor-intensive, requiring regular monitoring by field workers. This is especially burdensome during critical periods like planting or drought conditions.
Time Consumption	Saves significant time by automating the monitoring and decision-making processes. Adjustments can be made instantly based on real-time data without human intervention.	Field checks and manual adjustments take considerable time, especially across large areas. This can lead to inefficiencies, particularly in responding to changing conditions.
Reliability	Highly reliable if properly maintained. Systems continuously monitor conditions and make adjustments as needed, with less risk of oversight.	Depends heavily on the consistency and expertise of the personnel involved. Variability in human performance can lead to inconsistencies.

Chapter 6

Expenses

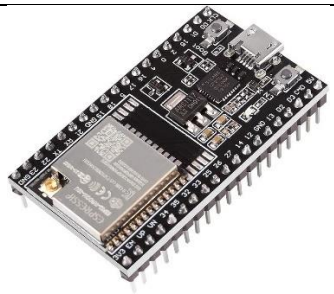
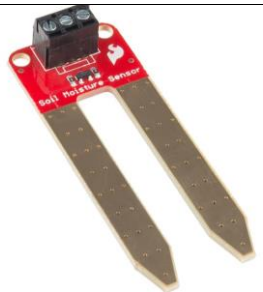
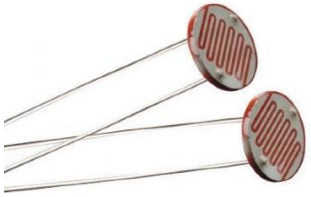
- Prototype Expenses
- Estimated Expenses

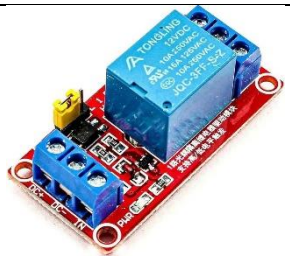
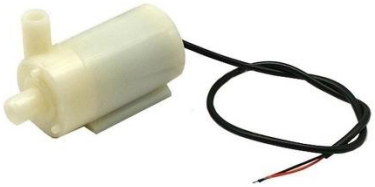
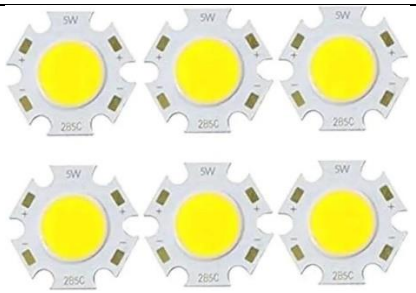
Chapter 6 - Expenses

6.1 Prototype Expenses

6.1.1 Breakdown of Prototype Hardware costs

Table 9 Breakdown of Prototype Hardware costs

	Picture	Price	Amount	Total
ESP32		400	1	400
Micro USB		75	1	75
DHT22		220	1	220
HC-SR501		65	1	65
Soil Moisture Sensor		75	1	75
LDR		15	1	15

1 KΩ Resistor		1	1	1
MT3608		80	1	80
Relay Module		45	1	45
Water Pump		110	1	110
SG90		110	1	110
Buzzer		15	1	15
LED Lamp		5	4	20
Water Pipe		5	1	5
Total	1,236			

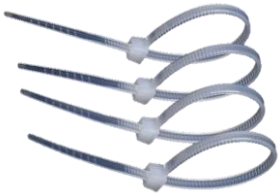
6.1.2 Prototype Software development expenses

Table 10 Prototype Software development expenses

	Picture	Price	Amount	Total
ESP32 Coding (C++)		0	1	0
Flutter (Dart)		0	1	0
Firebase (JavaScript)		0	1	0
Total	0			

6.1.3 Prototype Mucket building Expenses

Table 11 Prototype Mucket building Expenses

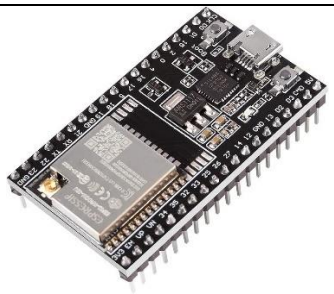
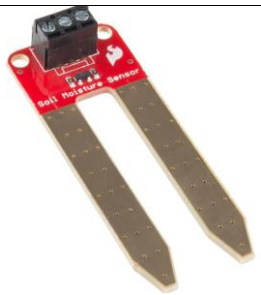
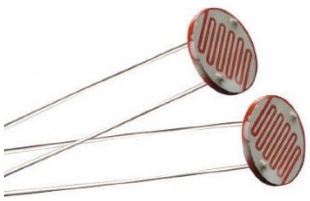
	Picture	Price	Amount	Total
Sticker		7	4	28
Metal wire		10	1	10
Metal rods		30	1	30
Grass Carpet		120	0.5	60
Fayez		0.5	20	10

Heat Shrink		35	1	35
Cardboard board		0	1	0
Green House Cloth		50	1	50
Glue Stick		5	4	20
Nylon Cord		20	1	20
Total	263			

6.2 Estimated Expenses

6.2.1 Breakdown of Estimated Hardware costs

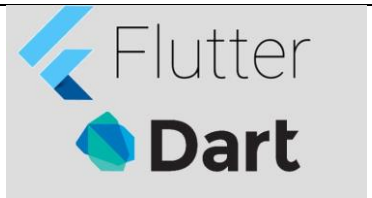
Table 12 Breakdown of Estimated Hardware costs

	Picture	Price	Amount	Total
ESP32		400	1	400
Micro USB		75	1	75
DHT22		220	1	250
HC-SR501		65	1	65
Soil Moisture Sensor		75	1	75
LDR		15	1	15

1 KΩ Resistor		1	1	1
MT3608		80	1	80
Electric Roll Up Motor		1,044	1	1,044
SFM-27		75	1	75
Total	2,080			

6.2.2 Estimated Software development expenses

Table 13 Estimated Software development expenses

	Picture	Price	Amount	Total
ESP32 Coding (C++)		0	1	0
Flutter (Dart)		0	1	0
Firebase (JavaScript)		0	1	0
Total	0			

Chapter 7

Conclusions and Recommendations

- Conclusions
- Challenges and Solutions
- Recommendations

Chapter 7 - Conclusions and Recommendations

7.1 Conclusions

Smart irrigation system:

To summarize our project, firstly, the Smart Irrigation System uses a soil moisture sensor to observe the moisture level of the soil, with a light-detecting resistor (LDR) to adjust watering times to be at the best suitable time. This allows us to know when the soil needs water and maintain optimal hydration levels. Of course, if the moisture level in the soil falls too low, the pump will activate anyway.

fire detection system:

Then, the fire detection system, utilizing the DHT22 temperature and humidity sensor, can identify fires in their early stages. This is achieved by detecting sudden increases in temperature or temperature rising over a certain level. In that case, it sends a warning message to the owner, plus activates a warning siren.

Smart protection system:

The Smart Protection System, the defense line against birds and rodents, activates once the motion sensor HC-SR501 senses any motion. It will engage alternating warning sirens and lights to scare the intruders and send a message to the user.

Smart lighting system:

After that, the Smart Lighting System. For it to work, it needs to be night, the Smart Protection System must be off, and the motion sensor must detect movement in the field or the greenhouse. This is to conserve electricity and only use it when needed.

Auto Greenhouse Roll-up sides system:

The Auto Greenhouse Roll-up Sides System for air circulation and humidity management is convenient and efficient, as it works with a simple click of a button on the mobile application.

Monitoring system (Mobile Application):

Finally, the Monitoring System (Mobile Application), as the name implies, is responsible for monitoring the field or the greenhouse. Using data from the sensors, the mobile application will display temperature, humidity, soil moisture, and notifications for fire and motion detection. In addition to monitoring, this system provides full control, allowing you to shut down the entire system or just the protection system.

7.2 Challenges and Solutions

While building the hardware we faced a set of challenges, related to developing the ESP32 code. And when building the application using Flutter to control an ESP32, we faced a set of dual challenges, related to developing the application with Flutter and also integrating with the ESP32. Here are the problems we faced and their solutions:

- **Problem**

Over time, the DHT22 sensor start giving inaccurate humidity readings. This is often due to the sensor's calibration drifting, especially if it has been exposed to high humidity conditions for extended periods.

- **Solution**

1. Recalibration: You can recalibrate the sensor by exposing it to a known humidity level. One way to do this is by using a saturated salt solution (e.g., a solution of table salt and water). Which creates a stable humidity environment.
2. Replacement: If recalibration doesn't work, it might be necessary to replace the sensor. DHT22 sensors are relatively inexpensive and easy to replace.
3. Environment Control: Try to keep the sensor in a stable environment, avoiding extreme humidity levels to prolong its accuracy.

- **Problem**

Soil moisture sensors sometimes give inconsistent or inaccurate readings due to factors like soil type, sensor placement, or environmental conditions.

- **Solution**

1. Proper Placement: Ensure the sensor is placed at the correct depth and location in the soil. It should be in an area that represents the average soil moisture of the field or garden.
2. Calibration: calibrate the sensor according to the specific soil type. Different soils (clay, loam, sand) have different moisture retention properties, so calibration is crucial for accurate readings.

- **Problem**

Servomotors overheat due to various reasons such as excessive load, poor ventilation, or prolonged operation without adequate cooling.

- **Solution**

1. Check the load: Ensure the servomotor is not overloaded. Operating within the specified load limits is crucial to prevent overheating.
2. Improve Ventilation: Make sur the motor has proper ventilation. Clean any dust or debris from the motor's housing and ensure that cooling fans or systems are functioning correctly.

3. Regular Maintenance: Perform regular maintenance checks to ensure all components are in good condition. This includes checking for worn-out bearings, misalignment, and ensuring proper lubrication.
4. Monitor Operating Conditions: Use temperature sensors to monitor the motor's operating temperature. If the motor frequently overheats, consider using a motor with a higher power rating or improving the cooling system.

- **Problem**

LDRs give inconsistent readings if they are exposed to varying ambient light conditions, which affect their accuracy in detecting specific light levels.

- **Solution**

1. Shielding: use alight shield or enclosure to protect the LDR from unwanted ambient light. This helps to ensure that sensor only detects the intended light source.
2. Calibration: calibrate the LDR in the specific lighting conditions where it will be used. This can help to adjust the sensor's sensitivity and improve accuracy.
3. Stable Environment: place the LDR in a stable environment where light conditions do not change frequently. Avoid placing it near windows or other sources of fluctuating light.

- **Problem**

Connecting to the network (Wi-Fi) and controlling the ESP

- **Solution**

1. Use the `flutter_wifi_connect` package to facilitate the application's connection to the Wi-Fi networks that control the ESP.
2. Ensure that the ESP is programmed to use standard protocols such as HTTP, WebSockets, or MQTT to communicate with the application.
3. Create an API in the ESP that allows control via HTTP commands or through other communication protocols.

- **Problem**

Managing connection and latency

- **Solution**

1. Ensure that the application uses a stable and low-latency connection such as WebSockets or MQTT to ensure fast response times.
2. Use the `http` or `mqtt_client` library in Flutter to interact with the ESP effectively.
3. Optimize the code in the ESP module to ensure low latency.

- **Problem**

Application compatibility across different platforms.

- **Solution**

1. Test your application on multiple operating systems (iOS and Android) to ensure compatibility with all devices.

2. Use packages and libraries supported by Flutter to ensure compatibility across different platforms.

- **Problem**

Power management in the ESP module

- **Solution**

1. If the ESP module is battery powered, use power saving modes such as `Deep Sleep` to reduce power consumption.
2. Ensure that the application in Flutter can handle reconnection cases when the ESP is in deep sleep mode.

- **Problem**

Over-the-Air (OTA) software updates

- **Solution**

1. Set up the ESP module to receive Over-The-Air (OTA) software updates, and use libraries such as `arduino_ota`.
2. Provide an interface in the application that enables the user to easily trigger the update process.

- **Problem**

Security

- **Solution**

1. Ensure that the communication between the application and the ESP module is encrypted using protocols such as HTTPS or WSS (WebSocket Secure).
2. Use strong passwords and follow security best practices in ESP module programming to ensure the security of the data exchanged.

- **Problem**

Handling crashes and disconnections

- **Solution**

1. Add mechanisms in the application to automatically retry when the connection fails.
2. Use a library such as `connectivity` in Filters to detect the network state and reconnect when needed.

- **Problem**

Integration with other features in the application

- **Solution**

1. Use an appropriate state management approach (such as `Provider` or `Bloc`) to ensure smooth synchronization between the UI and the ESP module.
2. Ensure that the application can perform other functions (such as notifications or updates) while controlling the ESP module.

- **Problem**

Performance and resource constraints in the ESP module

- **Solution**

1. Reduce the size of the data sent to the ESP module and use compressed data formats such as JSON or Protobuf.
2. Optimize the code in the ESP module to ensure efficient use of resources and reduce the load on the processor and memory.

- **Problem**

User Experience (UX)

- **Solution**

1. Design a simple and easy-to-use user interface with clear instructions for controlling the ESP module.
2. Use animations and visual indicators in filters to inform users of the connection status and any potential errors.

7.3 Recommendations

Soil Moisture Sensor:

1. Used to measure soil moisture levels.
2. Can be connected to an ESP32.
3. When moisture levels are low, it can activate the water pump.

ESP32:

1. A compact microcontroller unit that supports Wi-Fi and Bluetooth.
2. Used for controlling the irrigation system and communicating with sensors.

Water Pump:

1. Used to pump water from the reservoir to the soil.
2. It turns on when humidity levels are low.

Motion Sensor (HC-SR501):

Use the motion sensor to detect the presence of people or animals near the plants.

Light Dependent Resistor (LDR):

1. Measures light intensity.
2. Can be used to determine when to activate the pump (during the day or night).

Servo Motor:

1. Connect the servo motor to the Arduino.
2. Use it to adjust the water spray direction.

Buzzer:

1. Connect it to the ESP32.
2. Emit an audible signal when the Motion Sensor (HC-SR501) is triggered.

DHT22 Sensor:

Connect the DHT22 sensor to measure temperature and humidity in the air.

Programming:

Use the Arduino IDE to program the ESP32.

Chapter 8

Future Work

Chapter 8 - Future Work

- **Integration of Weather Data in Smart Irrigation Systems:** The concept of integrating weather data into smart irrigation systems involves using weather forecasts to adjust irrigation schedules based on anticipated rainfall. When rain is forecasted in the near future, the system can automatically delay or halt irrigation to reduce water consumption and prevent overwatering. This approach enhances water use efficiency, promotes environmental sustainability, and reduces operational costs.
- **Create a highly efficient and responsive agricultural monitoring system**
  Design of an advanced camera system with the following components:
- **Wildlife Detection:**
  - Animal Identification: Cameras use computer vision techniques to identify various animal species that may threaten crops. This identification is based on visual features like size and shape.
  - Bird Identification: Specialized algorithms recognize bird species by analyzing their shape and color, enabling the system to address birds that might impact the crops.
- **Unknown Person Detection:** The system employs facial recognition and pattern recognition to identify unauthorized individuals. When an unfamiliar face is detected, an alarm is triggered, and immediate notifications are sent to the user.
- **Pest Detection:** Insect Identification: Cameras equipped with specialized sensors analyze images to identify harmful insects.
- **Smart Response:** Upon detecting any of the above threats, the system activates alarms and sends real-time notifications to the user. After sending the data to the user, they can take the appropriate action based on the information provided in the message. For pest infestations, an intelligent spraying system applies pesticides precisely to affected areas only, minimizing chemical use and reducing environmental impact.
- **Chlorophyll sensors:** measure the chlorophyll content in plants, providing insights into their health and photosynthetic activity. These sensors help monitor plant growth, detect stress, and optimize agricultural practices by indicating nutrient status and overall plant vitality.
- **Carbon dioxide (CO_2) sensors:** measure the concentration of CO_2 in the environment. They are essential for monitoring air quality, controlling ventilation systems, and ensuring optimal conditions for plant growth in greenhouses. By tracking CO_2 levels, these sensors help maintain a balance between plant respiration and photosynthesis, contributing to better crop yields and energy efficiency.

- **An AI camera designed for plant disease detection:** uses advanced image processing and machine learning algorithms to identify symptoms of diseases in plants. By capturing images of the plants, the AI system can analyze visual patterns and detect abnormalities such as discoloration, spots, or wilting. Once a disease is identified, the system can provide recommendations for treatment, including the appropriate use of pesticides, fungicides, or other remedies. This technology enables farmers to quickly and accurately diagnose plant diseases, allowing for timely intervention and reducing crop losses. It also helps in minimizing the use of chemicals by targeting only the affected areas, promoting more sustainable agricultural practices.

By integrating these capabilities, the system provides a comprehensive solution for protecting crops from environmental and human threats, enhancing agricultural efficiency and sustainability.

References

[1] F. a. A. Organization, "Over and Under Watering," FAO, 2024. [Online]. Available: https://www.fao.org/.

[2] I. U. f. C. o. Nature, "Birds and Rodents Damage," IUCN, 2024. [Online]. Available: https://www.iucn.org/ .

[3] A. F. B. Federation, "Fire Propagation Damage," AFBF, 2022. [Online]. Available: https://www.fb.org/ .

[4] ا. المدني, "المتطلبات الوقائية للحماية من الحريق في المنشآت," الدفاع المدني, 2020. [Online]. Available: https://998.gov.sa/Ar/Safety/SafetyInstructionList/Pages/SafetyInstForFacilitis.aspx.

[5] PlanRadar, "كيفية التخطيط للسلامة من الحرائق في موقع البناء," PlanRadar, 2023. [Online]. Available: https://www.planradar.com/ae/how-to-plan-fire-safety-construction-site/.

[6] J. o. I. P. Management, "Entomological Society of America," Oxford Acadamy, 2020. [Online]. Available: https://academic.oup.com/jipm/article/11/1/11/5870581.

[7] Agronomy, "Rodents in Agriculture: A Broad Perspective," MDPI, 2022. [Online]. Available: https://www.mdpi.com/1683454.

[8] A. Desoky, "DAMAGE CAUSED BY BIRDS AND RODENT IN FIELD CROPS AND THEIR CONTROL," ResearchGate, 2014. [Online]. Available: https://www.researchgate.net/publication/277349116_DAMAGE_CAUSED_BY_BIRDS_AND_RODENT_IN_FIELD_CROPS_AND_THEIR_CONTROL.

[9] W. C. f. A. H. a. Safety, "Night Work: A Growing Trend in Western Agriculture?," UCDAVIS, 2014. [Online]. Available: https://aghealth.ucdavis.edu/news/night-work-growing-trend-western-agriculture.

[10] K. Parikka, "8 Most Common Unsafe Conditions in Agriculture and Farming," Falcony, 2024. [Online]. Available: https://blog.falcony.io/en/8-most-common-unsafe-conditions-in-agriculture-and-farming.

[11] M. Advisor, "ESP32 with DHT11/DHT22 Temperature and Humidity Sensor using Arduino IDE," Random Nerds Tutorial, 2017. [Online]. Available: https://randomnerdtutorials.com/esp32-dht11-dht22-temperature-humidity-sensor-arduino-ide/.

[12] L. Aosong Electronics Co., "DHT22 Datasheet," Sparkfun, 2022. [Online]. Available: https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf.

[13] D. Das, "How does an HC-SR501 PIR Sensor Work & How to Interface it with ESP32?," Circuit Digest, 2021. [Online]. Available: https://circuitdigest.com/microcontroller-projects/interface-hcsr501-pir-sensor-with-esp32.

[14] MPJA, "HC-SR501 PIR MOTION DETECTOR," aliexpress, 2024. [Online]. Available: https://www.mpja.com/download/31227sc.pdf.

[15] A. Microsys, "LDR المقاومة الضوئية - كورس اردوينو," YouTube, 2018. [Online]. Available: https://www.youtube.com/watch?v=tQNscK1RgXI.

[16] E. P. Focus, "What is Light Dependent Resistor : Circuit & Its Working," EL-PRO-CUS, 2016. [Online]. Available: https://www.elprocus.com/ldr-light-dependent-resistor-circuit-and-working/.

[17] I. Poole, "Clear summaries & information about electronics, radio and connectivity from electronics engineer," electronisnotes, 2012. [Online]. Available: https://www.electronics-notes.com/.

[18] E. Bureau, "What Is Light Dependent Resistor (LDR)?," electronicsforu, 2024. [Online]. Available: https://www.electronicsforu.com/technology-trends/learn-electronics/ldr-light-dependent-resistors-basics.

[19] N. F. P. Association, "They provide standards and guidelines on fire detection system," NFPA, 2024. [Online]. Available: nfpa.

[20] U. Laboratories, "Offers testing and certification information for various fire detection technologies," UL, 2024. [Online]. Available: https://www.ul.com/.

[21] I. O. f. Standardization, "Includes standards related to fire detection and alarm systems, such as ISO 7240," ISO, 2024. [Online]. Available: https://www.iso.org/home.html.

Appendices

Appendix A ESP32 Code

```
#include <DHT.h>
#include <Adafruit_Sensor.h>
#include <Servo.h>
#include <Wire.h>

#define buzzerPin 26
#define pirPin 23
#define ledPin 22
#define DHT22_PIN 18
#define SensorSM_Pin 15
#define pump_Pin 32
#define ldrPin 4
#define servoState_pin 33

DHT DHT22_Sensor(DHT22_PIN, DHT22);
Servo myServo;
int val = 0;

void setup() {
  pinMode(ldrPin, INPUT);
  pinMode(buzzerPin, OUTPUT);
  pinMode(servoState_pin, OUTPUT);
  pinMode(ledPin, OUTPUT);
  pinMode(pump_Pin, OUTPUT);
  pinMode(pirPin, INPUT);
  pinMode(SensorSM_Pin, INPUT);

  Serial.begin(115200);
  DHT22_Sensor.begin();
  myServo.attach(servoState_pin);
}

void control_servo() {
  if (Serial.available() > 0) {
    int servoState = Serial.read();

    if (servoState == '1') {
```

```
      Serial.println("Moving servo to 180 degrees");
      myServo.write(180);
    } else {
      Serial.println("Moving servo to 0 degrees");
      myServo.write(0);
    }
  }
}

void loop() {

  long sensorValue = analogRead(SensorSM_Pin);
  Serial.println(sensorValue);
  delay(1000);

  if (sensorValue >= 4095) {
    digitalWrite(pump_Pin, HIGH);
    delay(2000);
  } else {
    digitalWrite(pump_Pin, LOW);
  }

  // DHT22 with buzzer
  val = digitalRead(pirPin);
  float DHT22_Humidity = DHT22_Sensor.readHumidity();
  float DHT22_TempC = DHT22_Sensor.readTemperature();

  Serial.print("Humidity: ");
  Serial.print(DHT22_Humidity);
  Serial.println(" % ");
  Serial.print("Temperature: ");
  Serial.print(DHT22_TempC);
  Serial.println(" °C ");

  delay(1000);

  if (DHT22_TempC > 35) {
    digitalWrite(buzzerPin, HIGH);
    delay(500);
    digitalWrite(buzzerPin, LOW);
```

```
  delay(500);
} else {
  digitalWrite(buzzerPin, LOW);
}

if (val == HIGH) {
  Serial.println(val);
  digitalWrite(ledPin, HIGH);
  delay(30);
  digitalWrite(buzzerPin, HIGH);
  delay(1000);

  if (Serial.available() > 0) {
    String received = Serial.readString();
    received.trim();
    if (received == "off") {
      digitalWrite(buzzerPin, LOW);
    } else if (received == "on") {
      digitalWrite(ledPin, HIGH);
    }
  }
} else {
  digitalWrite(ledPin, LOW);
  digitalWrite(buzzerPin, LOW);
}

int ldrValue = digitalRead(ldrPin);
if (ldrValue == HIGH) {
  digitalWrite(ledPin, LOW);
} else {
  digitalWrite(ledPin, HIGH);
}

control_servo();
```

Appendix B Flutter Code

```
import 'package:flutter/foundation.dart';
import 'package:flutter/material.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Soil Measurements',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: const MyHomePage(),
    );
  }
}

class MyHomePage extends StatefulWidget {
  const MyHomePage({super.key});

  @override
  MyHomePageState createState() => MyHomePageState();
}

class MyHomePageState extends State<MyHomePage> {
  final double _soilMoisture = 50.0; // تعيين القيم الافتراضية
  final double _temperature = 25.0;
  final double _humidity = 50.0;

  int _selectedIndex = 0;

  void _onItemTapped(int index) {
```

```
  setState(() {
    _selectedIndex = index;
  });
}

static const List<Widget> _pages = <Widget>[
  SoilMeasurementsPage(),
  Center(
    child: Text(
      'Favorites Page',
      style: TextStyle(fontSize: 24),
    ),
  ),
  NotificationsPage(),
];

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text(
        'GREEN MILE',
        textAlign: TextAlign.center,
        style: TextStyle(color: Colors.white), // تغيير لون النص إلى الأبيض
      ),
      centerTitle: true,
      backgroundColor: Colors.lightGreen, // تغيير لون شريط التطبيق إلى الأخضر الفاتح
    ),
    body: _pages[_selectedIndex],
    bottomNavigationBar: BottomNavigationBar(
      items: const <BottomNavigationBarItem>[
        BottomNavigationBarItem(
          icon: Icon(Icons.home),
          label: 'Home',
        ),
        BottomNavigationBarItem(
          icon: Icon(Icons.radio_button_checked),
          label: 'system',
        ),
        BottomNavigationBarItem(
```

```
            icon: Icon(Icons.notifications),
            label: 'Notifications',
          ),
        ],
        currentIndex: _selectedIndex,
        selectedItemColor: Colors.black,
        unselectedItemColor: Colors.white,
        backgroundColor: Colors.green,
        onTap: _onItemTapped,
      ),
    );
  }
}

class SoilMeasurementsPage extends StatefulWidget {
  const SoilMeasurementsPage({super.key});

  @override
  _SoilMeasurementsPageState createState() => _SoilMeasurementsPageState();
}

class _SoilMeasurementsPageState extends State<SoilMeasurementsPage> {
  final double _soilMoisture = 50.0;
  final double _temperature = 25.0;
  final double _humidity = 50.0;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Stack(
        children: <Widget>[
          // عنوان الصفحة
          const Positioned(
            top: 30, // تحريك النص لأسفل قليلاً من 50 إلى 100
            left: 0,
            right: 0,
            child: Text(
              'Soil Measurements',
              textAlign: TextAlign.center,
              style: TextStyle(
```

```
      fontSize: 26,
      fontWeight: FontWeight.bold,
      color: Colors.black,
    ),
  ),
),
// محتوى الصفحة
Padding(
  padding: EdgeInsets.symmetric(horizontal: 16.0),
  child: Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: <Widget>[
      Row(
        mainAxisAlignment: MainAxisAlignment.spaceEvenly,
        children: <Widget>[
          TemperatureIndicator(
            value: _temperature,
            label: 'Temperature',
            minTemp: -40.0, // تعديل الحد الأدنى هنا
            maxTemp: 125.0, // تعديل الحد الأقصى هنا
          ),
          HumidityIndicator(
            value: _humidity,
            label: 'Humidity',
          ),
        ],
      ),
      const SizedBox(height: 30.0),
      SoilMoistureIndicator(
        value: _soilMoisture,
        label: 'Soil Moisture',
      ),
      const SizedBox(
          height: 40.0), // زيادة المسافة بين المؤشرات والأزرار
      // أضفنا زري Manual و Automatic أسفل دوائر النسب
      Row(
        mainAxisAlignment: MainAxisAlignment.center,
        children: <Widget>[
          ElevatedButton(
            onPressed: () {
```

```
                if (kDebugMode) {
                  print("Manual Mode");
                }
              },
              style: ElevatedButton.styleFrom(
                backgroundColor: Colors.green, // تغيير لون زر "Manual" إلى الأخضر
              ),
              child: const Text(
                'Manual',
                style: TextStyle(color: Colors.white), // تغيير لون النص إلى الأبيض
              ),
            ),
            const SizedBox(width: 20.0),
            ElevatedButton(
              // ignore: avoid_print
              onPressed: () => print("Automatic Mode"),
              style: ElevatedButton.styleFrom(
                backgroundColor: Colors.green, // تغيير لون زر "Automatic" إلى الأخضر
              ),
              child: const Text(
                'Automatic',
                style: TextStyle(color: Colors.white), // تغيير لون النص إلى الأبيض
              ),
            ),
          ],
        ),
      ],
    ),
  ),
 );
 }
}

class NotificationsPage extends StatelessWidget {
  const NotificationsPage({super.key}); // Corrected constructor with Key? key

  final List<String> notifications = const [
    "Notification 1: You have a new message.",
```

```
    "Notification 2: Your order has been shipped.",
    "Notification 3: Reminder: Meeting at 3 PM.",
    "Notification 4: New comment on your post.",
    "Notification 5: Your password has been changed."
  ];

  @override
  Widget build(BuildContext context) {
    return ListView.builder(
      itemCount: notifications.length,
      itemBuilder: (context, index) {
        return ListTile(
          title: Text(notifications[index]),
        );
      },
    );
  }
}

class HumidityIndicator extends StatelessWidget {
  final double value;
  final String label;

  const HumidityIndicator({
    super.key,
    required this.value,
    required this.label,
  });

  @override
  Widget build(BuildContext context) {
    return Column(
      children: <Widget>[
        Text(
          label,
          style: const TextStyle(fontSize: 18.0, fontWeight: FontWeight.w600),
        ),
        const SizedBox(height: 8.0),
        Stack(
          alignment: Alignment.center,
```

```
          children: [
            SizedBox(
              width: 100.0,
              height: 100.0,
              child: CircularProgressIndicator(
                value: value / 100,
                strokeWidth: 12.0,
                color: value > 75
                    ? Colors.red
                    : Colors.blue, // تغيير اللون بناءً على القيمة
              ),
            ),
            Text(
              '${value.toStringAsFixed(1)}%',
              style: const TextStyle(
                  fontSize: 22.0,
                  fontWeight: FontWeight.bold,
                  color: Colors.black),
            ),
          ],
        ),
        const SizedBox(height: 8.0),
        // إظهار النطاق الرقمي
        Text(
          '0% - 100%',
          style:
          TextStyle(fontSize: 14.0, color: Colors.black.withOpacity(0.7)),
        ),
      ],
    );
  }
}

class SoilMoistureIndicator extends StatelessWidget {
  final double value;
  final String label;

  const SoilMoistureIndicator({
    super.key,
    required this.value,
```

```
  required this.label,
});

@override
Widget build(BuildContext context) {
  return Column(
    children: <Widget>[
      Text(
        label,
        style: const TextStyle(fontSize: 18.0, fontWeight: FontWeight.w600),
      ),
      const SizedBox(height: 8.0),
      Stack(
        alignment: Alignment.center,
        children: [
          SizedBox(
            width: 100.0,
            height: 100.0,
            child: CircularProgressIndicator(
              value: value / 100,
              strokeWidth: 12.0,
              color: Colors.green,
            ),
          ),
          Text(
            '${value.toStringAsFixed(1)}%',
            style: const TextStyle(
                fontSize: 22.0,
                fontWeight: FontWeight.bold,
                color: Colors.black),
          ),
        ],
      ),
      const SizedBox(height: 8.0),
      // إظهار النطاق الرقمي
      Text(
        '0% - 100%',
        style:
        TextStyle(fontSize: 14.0, color: Colors.black.withOpacity(0.7)),
      ),
```

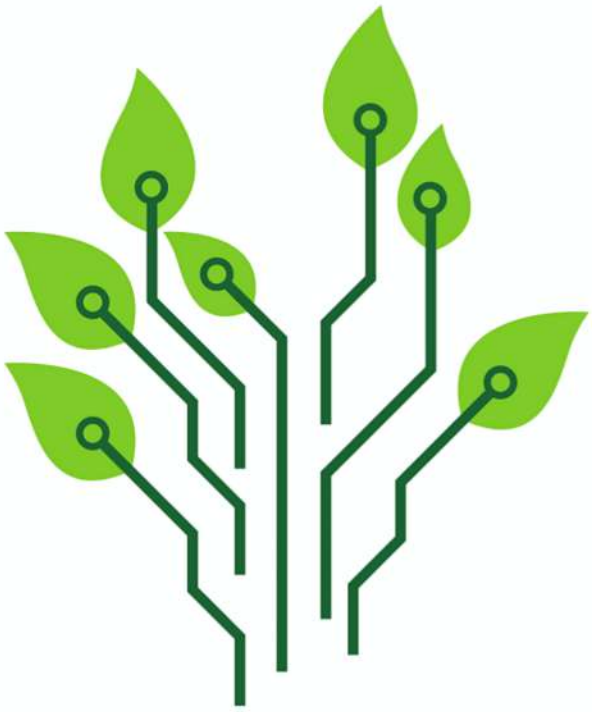
```
      ],
    );
  }
}

class TemperatureIndicator extends StatelessWidget {
  final double value;
  final String label;
  final double minTemp;
  final double maxTemp;

  const TemperatureIndicator({
    super.key,
    required this.value,
    required this.label,
    required this.minTemp,
    required this.maxTemp,
  });

  @override
  Widget build(BuildContext context) {
    return Column(
      children: <Widget>[
        Text(
          label,
          style: const TextStyle(fontSize: 18.0, fontWeight: FontWeight.w600),
        ),
        const SizedBox(height: 8.0),
        Stack(
          alignment: Alignment.center,
          children: [
            SizedBox(
              width: 100.0,
              height: 100.0,
              child: CircularProgressIndicator(
                value: (value - minTemp) / (maxTemp - minTemp),
                strokeWidth: 12.0,
                color: Colors.red,
              ),
            ),
```

```
            Text(
              '${value.toStringAsFixed(1)}°C',
              style: const TextStyle(
                  fontSize: 22.0,
                  fontWeight: FontWeight.bold,
                  color: Colors.black),
            ),
          ],
        ),
        const SizedBox(height: 8.0),
        // إظهار النطاق الرقمي
        Text(
          '${minTemp.toStringAsFixed(0)}°C -
${maxTemp.toStringAsFixed(0)}°C',
          style:
          TextStyle(fontSize: 14.0, color: Colors.black.withOpacity(0.7)),
        ),
      ],
    );
  }
}
```



GreenMile