CHAPTER 1 : INTRODUCTION

Traditional farming is facing unknown challenges in meeting the growing global demand for food, water by factors such as climate change, water scarcity, and population growth. In response, there is a pressing need for agricultural systems that are not only productive but also sustainable. The emergence of IoT technologies offers a solution by enabling the creation of smart agriculture systems capable of monitoring, analyzing, and optimizing various farm parameters in real-time(figure1). The life cycle of smart agriculture made to specify all natural issues with artificial technology .This paper presents an overview of such a system, designed to address the complexities of modern agriculture while making environmental atmosphere and economic viability.[1]

With the growth of world population, according to the UN Food and Agriculture Organization, the world will need to produce 70% more food in 2050, shrinking agricultural lands, and depletion of finite natural resources, We need to enhance farm yield has become critical. Limited availability of natural resources such as fresh water and fertile land along with slowing yield trends in several staple crops, have further aggravated the problem. Another concern over the farming industry is the shifting structure of agricultural workforce. Moreover, agricultural labor in most of the countries has declined**[2].** As a result of the declining agricultural workforce, adoption of internet connectivity solutions in farming practices has been triggered, to reduce the need for manual labor. IoT solutions are focused on helping farmers close the supply demand gap, by ensuring high yields, profitability, and protection of the environment.[3]

Precision agriculture, leveraging IoT technology, optimizes resource usage for enhanced crop yields and reduced operational expenses. This approach integrates special equipment, wireless connections,software used.

Based on a survey by BI Intelligence, it was anticipated that the adoption of IoT devices in agriculture would reach 75 million by 2020, with a 20% annual growth rate. Concurrently, the global smart agriculture market is expected to triple by 2025, reaching $15.3 billion, compared to just over $5 billion in 2016.

Figure-1.1:This figure shows the life cycle of smart Agriculture

1.1 PURPOSE

To identify plant monitoring to obtain the best method of smart agriculture system and to monitor few parameters for the growth of plants.

• To develop an algorithm in-plant monitoring system such as temperature, humidity, automatic irrigation system, detect the motion activity and rain status.

• To develop Smart Monitoring Plant Using IoT System.

The project focuses on utilizing the ESP32 as an environmental monitoring system, specifically for plant care. It incorporates a soil moisture detector to regulate plant watering based on moisture levels. Acting as the central control server, the ESP32 orchestrates the entire system, connecting to essential sensors for maintaining an ideal plant environment. Plant health status is monitored through Blynk apps, with an optimal pH range of 5.5 to 6.5 for plant thriving. The ESP32 gathers data from sensors, stores it, and facilitates integration of hardware and software components for real-time plant status monitoring via the Blynk server(figure[2], ensuring prompt assessment of plant health.**[4]**

Figure 1.2:This figure shows about plant in blink server

Despite considerable government efforts, small-scale farmers have not reaped the benefits, resulting in a significant gap between them and large-scale farmers, leading to an imbalance. To address this disparity, systematic planning is essential, requiring detailed regional information. In many countries like India, where farming sustains a majority of the population and contributes significantly to the national income, outdated agricultural practices persist despite the widespread availability of modern technology. Small farmers continue to rely on traditional methods such as manual seed distribution, limited cropping patterns, and outdated cultivation systems. The irregularity of monsoons and uneven water availability throughout the year further compound these challenges.**[5]**

CHAPTER 2 :REQUIRED SPECIFICATION

Hardware components:

1.Node MCU

2.DHT 11

3.Relay module

4.Rain Sensor

5.Soil moisture Sensor

6.PIR Sensor

7.pump

8.Bread board

9.Battery

10.UV Light

11.A Glass Box

Software components:

1.Windows 11

2.Arduino ide

3.Blink App

2.1 NODE MCU ESP8266

The NodeMCU development board showcases the ESP-12E module housing the ESP8266 chip, featuring a Tensilica Xtensa 32-bit LX106 RISC microprocessor. This versatile chip operates at a clock frequency range of 80MHz to 160MHz, supporting real-time operating systems, and offers 128 KB of RAM and 4MB of Flash memory for ample storage. With built-in Wi-Fi and Bluetooth capabilities, it's an ideal choice for Internet of Things (IoT) projects and can be powered via its Micro USB jack or the VIN pin.offering versatility in power sources. Additionally, it supports multiple communication interfaces including UART, SPI, and I2C, further enhancing its adaptability for various project requirements.

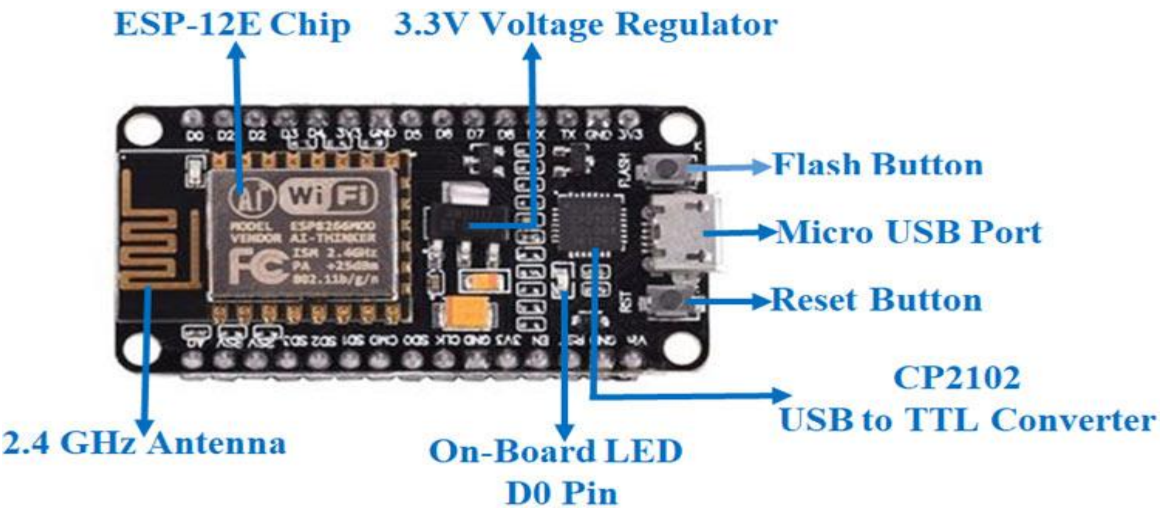


Figure 2.1:NodeMC

2.2 NodeMCU ESP8266 Specifications & Features

- Microcontroller used : Tensilica 32-bit RISC CPU Xtensa LX106
- Range of operating voltage : 3.3V
- Input Voltage Range: 7-12V
- Digital Input/OutputPins (DIO): 16
- Analog Input Pins in : 1
- UART: 1
- SPI: 1
- I2C: 1
- Flash Memory: 4 MB
- SRAM: 64 KB
- Clock Speed: 80 MHz
- USB-TTL based on CP2102 onboard, enabling plug and play functionality.

Pin Category	**Name**	**Description**
Power	Micro-USB, 3.3V, GND, Vin	**Micro-USB:** NodeMCU can be powered through the USB port **3.3V:** Regulated 3.3V can be supplied to this pin to power the board **GND:** Ground pins **Vin:** External Power Supply
Control Pins	**EN, RST**	The pin and the button resets the microcontroller
Analog Pin	A0	Used to measure analog voltage in the range of 0-3.3V
GPIO Pins	GPIO1 to GPIO16	NodeMCU has 16 general purpose input-output pins on its board
SPI Pins	SD1, CMD, SD0, CLK	NodeMCU has four pins available for SPI communication.
UART Pins	TXD0,RXD0, TXD2, RXD2	NodeMCU has two UART interfaces, UART0 (RXD0 & TXD0) and UART1 (RXD1 & TXD1). UART1 is used to upload the firmware/program.
I2C Pins		NodeMCU has I2C functionality support but due to the internal functionality of these pins, you have to find which pin is I2C.

This NodeMCU consists of total 30 pins and each pin has its own specification.

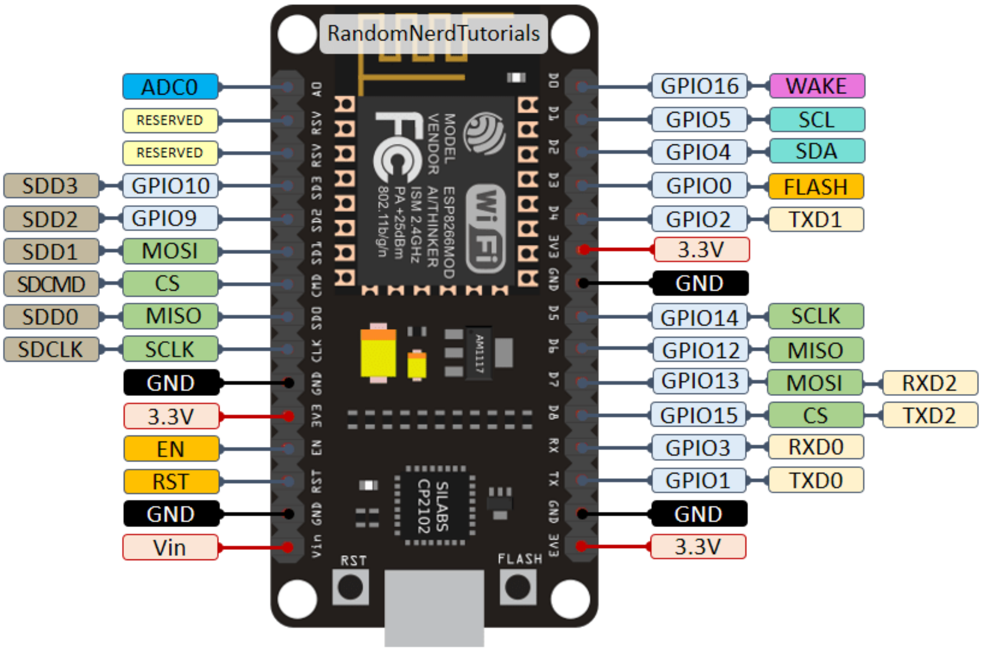


Fig2.2: ESP8266 NodeMCU pinout

You can power the NodeMCU/ESP8266 and its peripherals directly through the VIN pin, which is regulated by the onboard regulator on the NodeMCU module.**[6]** Alternatively, you can supply regulated 5V to the VIN pin. The 3.3V pins output from the onboard voltage regulator can be used to power external components.

Applications

- Prototyping of IoT device.
- Low power battery operated applications
- Network projects. Projects requiring multiple I/O interfaces with Wi-Fi and Bluetooth functionalities.

2.3 DHT11

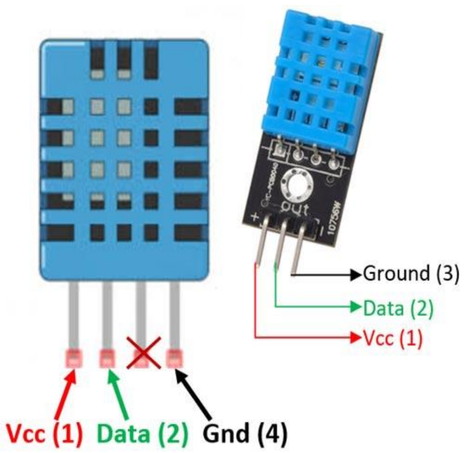
DHT11 Sensor measures temperature and humidity, two critical parameters in agriculture. By deploying these sensors throughout a farm or greenhouse, farmers can monitor the microclimate in different areas. This data can help optimize growing conditions for various crops, ensure proper humidity levels for seed germination, and prevent issues like mold or fungal growth. Different areas within a farm or greenhouse may have distinct microclimates due to factors like shading, topography, or proximity to water sources. DHT11 sensors can help identify these variations, allowing farmers to adjust planting schedules, crop selection, and cultivation practices accordingly. By continuously monitoring humidity with DHT11 sensors, farmers can implement preventive measures such as adjusting ventilation, spacing plants, or applying fungicides when necessary

This sensor does have 4 pins:

1) Vcc : Power supply ranges from 3.5V to 5.5V.

2) Data : Outputs both Temperature and Humidity range

3) Ground: made a connection between ground and the circuit



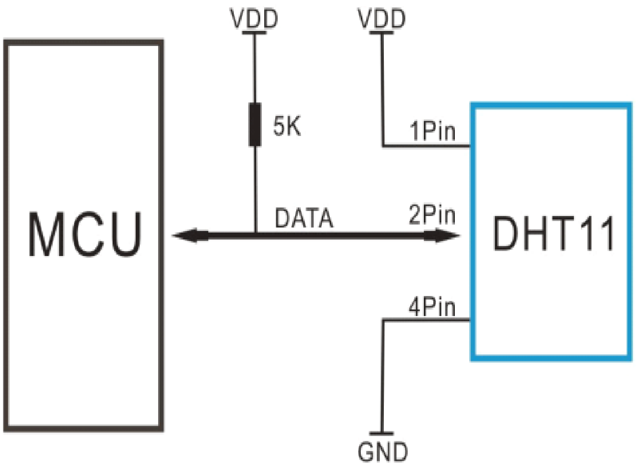


Figure 2.3:Temperature Sensor connection

Figure2.4:NodeMCU with the DHT11

As we can shown, The data pin is connected to an I/O pin of the MCU and a 5K pull-up Resistor is utilized. This data pin transmits both the temperature and humidity values as serial data.. Through this connection DHT11 values get read by the NodeMCU

As The data pin is linked to an I/O pin of the MCU and is accompanied by a 5K pull-up resistor, and it transmits both temperature and humidity values as serial data.. Through this connection DHT11 values get read by the NodeMCU.

DHT11 Specifications:

- Operating Voltage range: 3.5V to 5.5V
- Operating current point: 0.3mA (measuring) 60uA (standby)
- Outputs serial data
- Has a Temperature ranges 0°C to 50°C
- Humidity Range of 20% to 90%
- Resolution bits based for Temperature and Humidity both are 16-bit
- The Accuracy(perfection) is ±1°C and ±1%

DHT11 Applications:

- Measure temperature and humidity
- Weather stations
- HVAC system
- Home automation system

2.4 Relay Module:

A relay is a type of electromechanical switch that gets activated by an electrical signal and consists of an electromagnet coil and one or more sets of contacts, which allows current to flow through the contacts when the coil generates a magnetic field. Relays are commonly employed to control high-power devices with low-power signals from microcontrollers or other electronic circuits.

Figure 2.5: Relay module

Pin Configuration:

Relay modules typically have several pins for connecting to external circuits and power sources. Here's a typical pin configuration:

VCC: pin is used to supply power. It's usually connected to a power source.

GND: This pin is the ground connection for the relay module. It's connection to the ground (0Volts) of the power supply.

Signal Input: This pin is used to control the relay. Applying a voltage or logic signal to this pin activates the relay.

Normally Open (NO) and Normally Closed (NC) Contacts: These pins are the switching contacts of the relay. They are usually labelled as NO and NC and are connected to the load (device) that you want to control.

COM (Common): This pin is common to both the NO and NC contacts. It's typically connected to one side of the power supply or load.

Optional Pins: Some relay modules may have additional pins for features like indicator LEDs, protection diodes, or additional control options.

Applications of Relay module:

Home automation

Industrial automation

Safety system

Security system

2.5 Soil Moisture sensor

The soil moisture sensor measures the moisture content in the soil by checking the volume of water present and estimating it through coefficients. It detects the water content and sends analog signals displayed digitally.

Soil moisture sensors transmit soil condition information to Arduino for processing and display, aiding farmers in monitoring moisture levels and drainage to prevent waterlogging and salinity issues. Real-time data on soil moisture levels at different depths allows farmers to make informed irrigation decisions, ensuring optimal water supply for crops. Farmers can use this information to determine when and how much to irrigate, ensuring that crops receive the right amount of water at the right time.

Figure 2.6:Soil moisture sensor

Pin configuration:

As shown figure8

VCC (or VDD): This is the power supply pins. It is typically connected to a Range of 3.3V or 5V pin on the microcontroller of power source.

GND (or Ground): This is the ground pin based on the soil moisture sensor. It is connected to the ground or GND pin of the microcontroller or Powersource.

Analog Output: This pin provides an analog voltage output that corresponds to the moisture level detected by the sensor.

It is usually connected to one of the analog input pins (e.g., A0, A1, etc.) on a microcontroller.

Digital Output (Optional): Some soil moisture sensors also have a digital output pin that provides a binary syatem (usually HIGH or LOW) based on a preset moisture threshold.

This pin can be connected to a digital input pin on a r for digital readings.

Application of soil moisture sensor:

Precision irrigation

Crop Management

Environment monitoring

Water resource management

2.6 PIR SENSOR

PIR sensors, which stands for Passive Infrared sensors, are devices In security systems, automatic lighting systems, and other applications requiring motion detection, passive infrared sensors detect motion by measuring changes in infrared radiation levels within their field of view.

Figure 2.7:PIR sensor

Pin Configuration:

As shown in figure-9

VCC or VDD: This pin is for the power supply. It's where you connect the positive terminal of your power source, usually 3.3V or 5V, depending on the specifications of your PIR sensor.

GND: This pin is for the ground connection. Connect the negative terminal of your power source (ground) to this pin.

OUT or SIGNAL: This pin outputs a digital signal when motion is detected. It usually sends a HIGH (or 1) signal when motion is detected and very low (or 0) when there's no motion.

CIRCUIT :

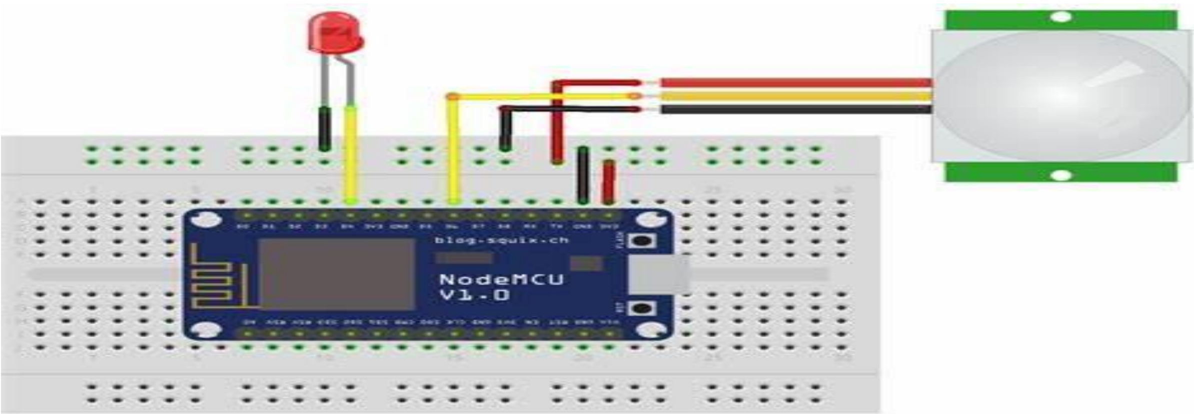


Figure-2.8 Node mcu connection with PIR sensor

2.7 RAIN SENSOR:

A rain sensor, also known as a rain switch or rain detector, is a device used to detect the presence of rain or moisture. It's commonly used in automated weather systems, irrigation systems, and automotive applications such as automatic windshield wipers. The basic principle behind most raindrop sensors involves detecting changes in conductivity or capacitance caused by the presence of water. When raindrops fall on the sensor surface, they alter its electrical properties, which can be measured by the sensor circuitry. **[8]**

Figure-2.9:Rain Sensor

There are different types of raindrop sensors available, ranging from simple ones that trigger when they detect moisture to more sophisticated models that can measure rainfall intensity or accumulate rainfall data over time.

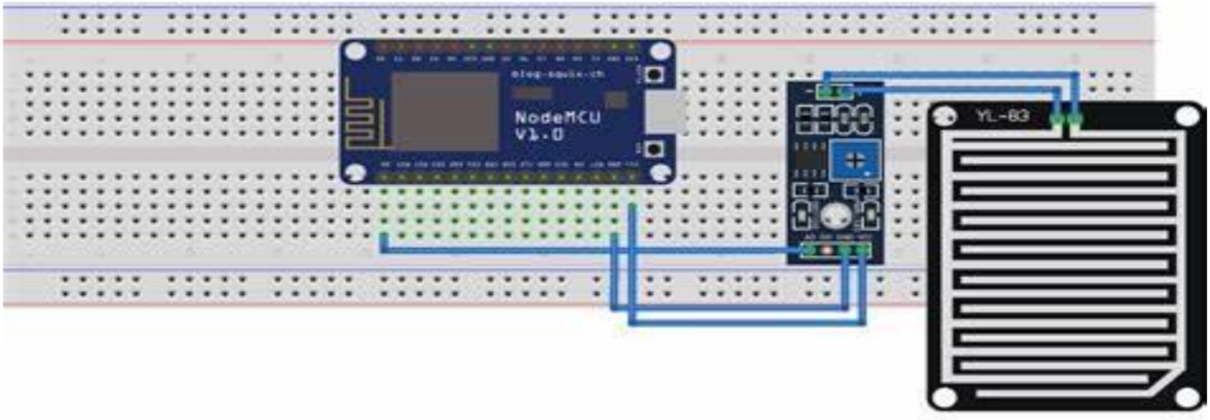


Figure-2.10:Rain sensor connections with Node MCU

Pin configuration:

VCC (Power Supply): This pin which are used to provide power to the rain sensor. It is usually connected to a positive voltage source range of, such as +5V or +3.3V, depending on the sensor's specifications.

GND (Ground): This pin is the ground connections for the rain sensor which is helpful for detecting the rain. It should be connected to the ground terminal of your power supply or microcontroller.

Analog Output: Many rain sensors feature an analog output pin. This pin provides an analog voltage or current signal that varies based on the intensity of rainfall. The output voltage may increase as the amount of rain detected increases.

Digital Output: Some rain sensors also offer a digital output pin. This pin typically provides a digital signal, such as a logic high or logic low to indicate the presence or absence of rain. The digital output may be triggered when a certain threshold of rainfall is detected.

Heater Control (Optional): In some advanced rain sensor models, there may be a pin for controlling an internal heating element. This heating element helps prevent the accumulation of dew or frost on the sensor's surface, ensuring accurate rain detection in various weather conditions.

Adjustment/Control (Optional): Certain rain sensors may feature pins for adjusting sensitivity or threshold levels. These pins allow you to customize the sensor's response to different rainfall intensities or environmental conditions.

2.8 BATTERY :

In this project may involve mobile applications for monitoring and management tasks. Tablets, smartphones, or handheld devices used by farmers or agricultural technicians require battery power to operate in the field. batteries play a crucial role in providing power to various components and devices used for monitoring, automation, and data collection. Here's how batteries are typically used in smart agriculture projects:.

Figure-2.11:Battery

Batteries enable the deployment of remote monitoring methods and control systems in agricultural settings. Farmers can use smartphones, laptops, or computer to access real-time data and control irrigation systems, fertilization equipment, and other machinery from anywhere, thanks to battery-powered communication.**[9]**

2.10 USB CABLE:

In an IoT project, a USB cable serves several crucial functions. It provides a reliable power supply to IoT devices, ensuring continuous operation. It facilitates data transfer between the IoT device and other components, such as sensors or controllers. Additionally, it is used for initial device setup, firmware updates, and debugging. The USB cable's versatility makes it an essential component in the connectivity and maintenance of IoT systems.

Figure 2.12

2.10 BREAD BOARD

Breadboards are essential tools for building and testing electronic circuits without soldering. They are especially useful for prototyping and experimentation. Here are some reasons why breadboards matter:

Prototyping and Testing: Breadboards allow for the quick assembly and disassembly of circuits, making them ideal for prototyping new designs and testing different configurations before committing to a final design.

No Soldering Required: Breadboards eliminate the need for soldering, which saves time and allows for easy modifications and troubleshooting. This is particularly useful for beginners who may not be comfortable with soldering.

Figure-2.13:BreadBoard

2.11 JUMPER WIRES

Jumper wires play a crucial role in electronics prototyping and circuit building by providing flexible wire connections with connectors at each end, allowing for easy component connection on a breadboard or between various points on a circuit. These wires are available in a variety of lengths, colors, and types, including male-to-male, male-to-female, and female-to-female options.

jumper wires are essential components used in breadboarding and prototyping electronic circuits. They play a crucial role in connecting different parts of the circuit and ensuring properties

Figure:2.14 jumper wires

CHAPTER 3:METHODOLOGY

This document outlines the research methodology employed in the study, detailing the data collection equipment, procedures, and data analysis processes. It also serves as a reference for addressing any future concerns and evaluating the project's overall strategy.

Key hardware components utilized in the study include a soil moisture sensor connected to the NodeMCU ESP32 microcontroller, along with additional sensors such as a rain sensor, PIR sensor, and DHT11 sensor.

The Arduino IDE software is employed to program the microcontroller, enabling it to collect and process data from the sensors. Additionally, Blynk apps are utilized to facilitate the transfer of information and data obtained by the system, ensuring seamless communication with farmers or relevant organizations.

This comprehensive approach ensures the efficient collection, analysis, and dissemination of valuable agricultural data, ultimately contributing to improved decision-making and resource management in agricultural practices.

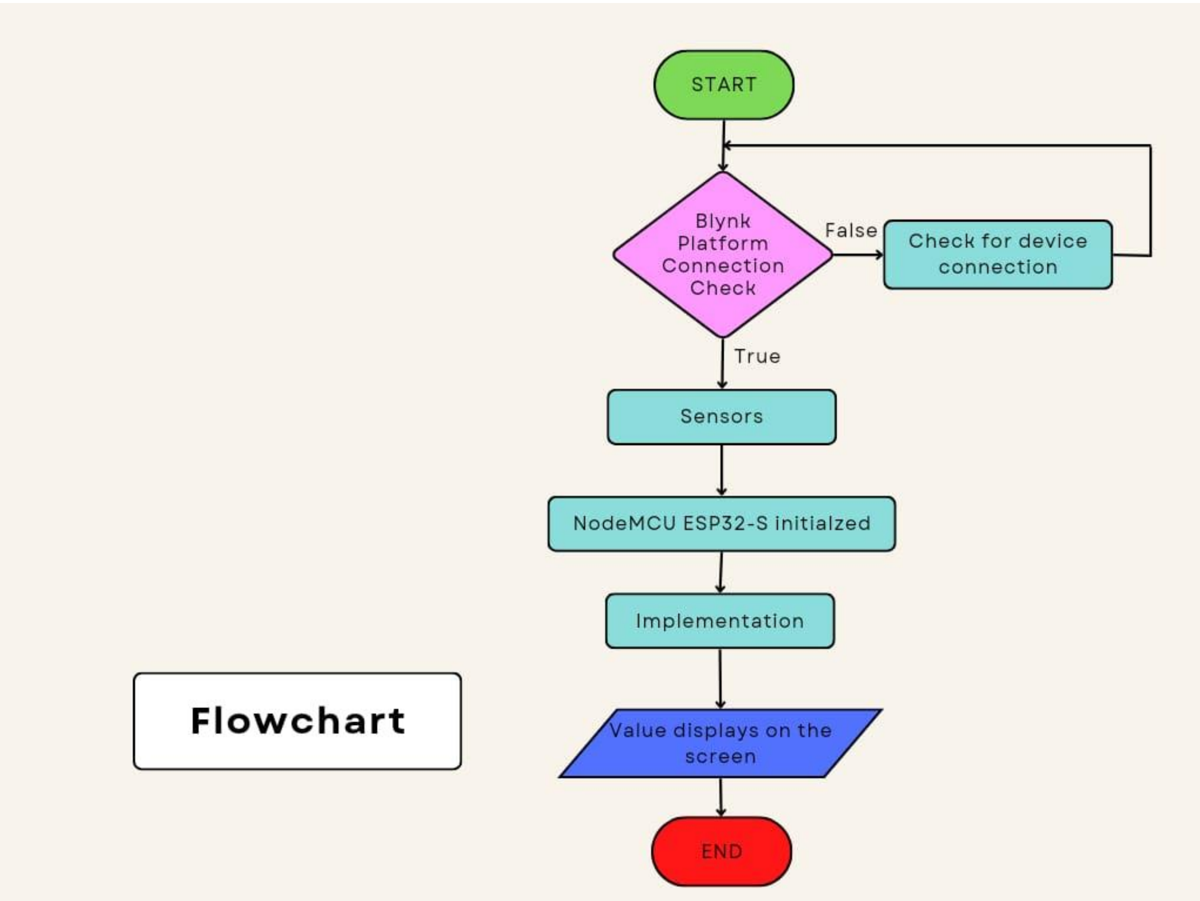


Figure-3.1 Flow chart along with the BLINK platform

3.1 BLOCK DIAGRAM

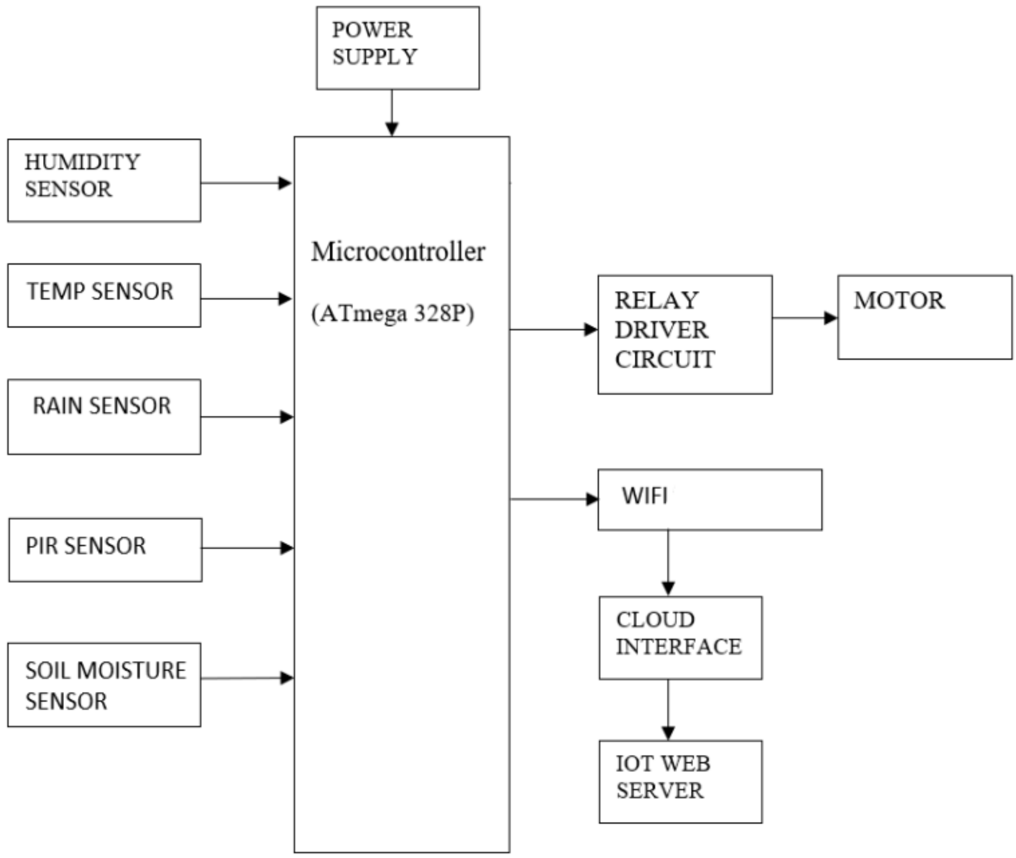


Figure-3.2: Hardware Setup With MicroController

Here's a structured outline of the methodology:

- Needs Assessment and Planning
- System Design
- Hardware Implementation
- Software Development
- Field Testing and validation
- Data Analysis
- Evaluation and verification

CHAPTER 4:IMPLEMENTATION

4.1.1 LAUNCH ARDUINO IDE

- Download and install the Arduino IDE from the Arduino website.
- For Windows: Click the Arduino icon on the desktop or find it in the Start Menu.
- For macOS: Open the Applications folder and double-click the Arduino app.
- For Linux: Run arduino from the terminal after installing.
- Start creating or uploading sketches to your Arduino board

4.1.2 Create a new project……To create a new project, select File → **New**

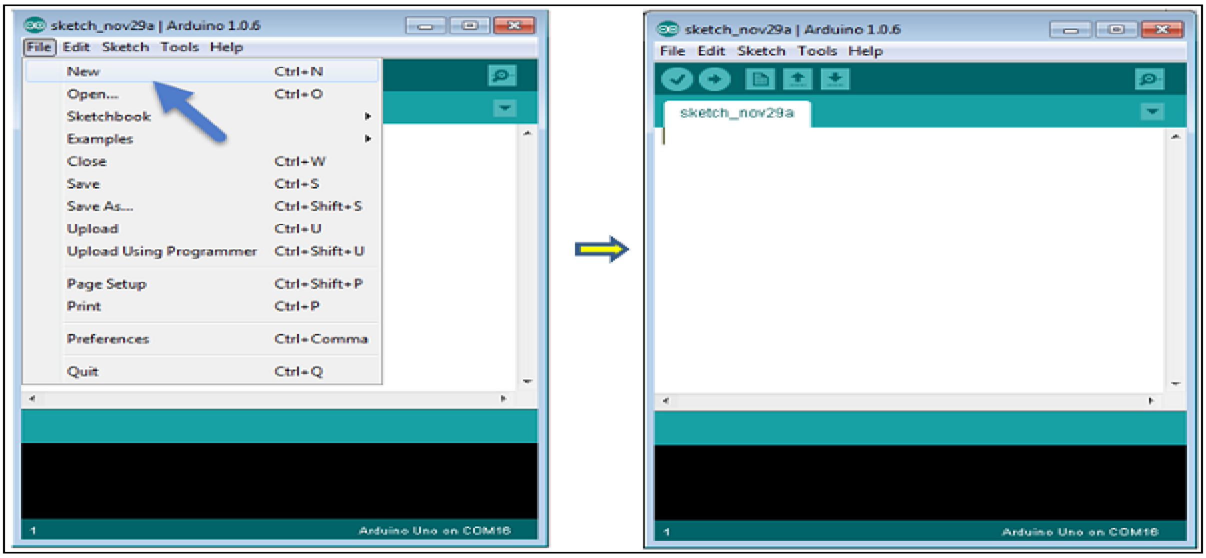


Figure-4.1.1: Create New project

4.1.3 Select your Arduino Board

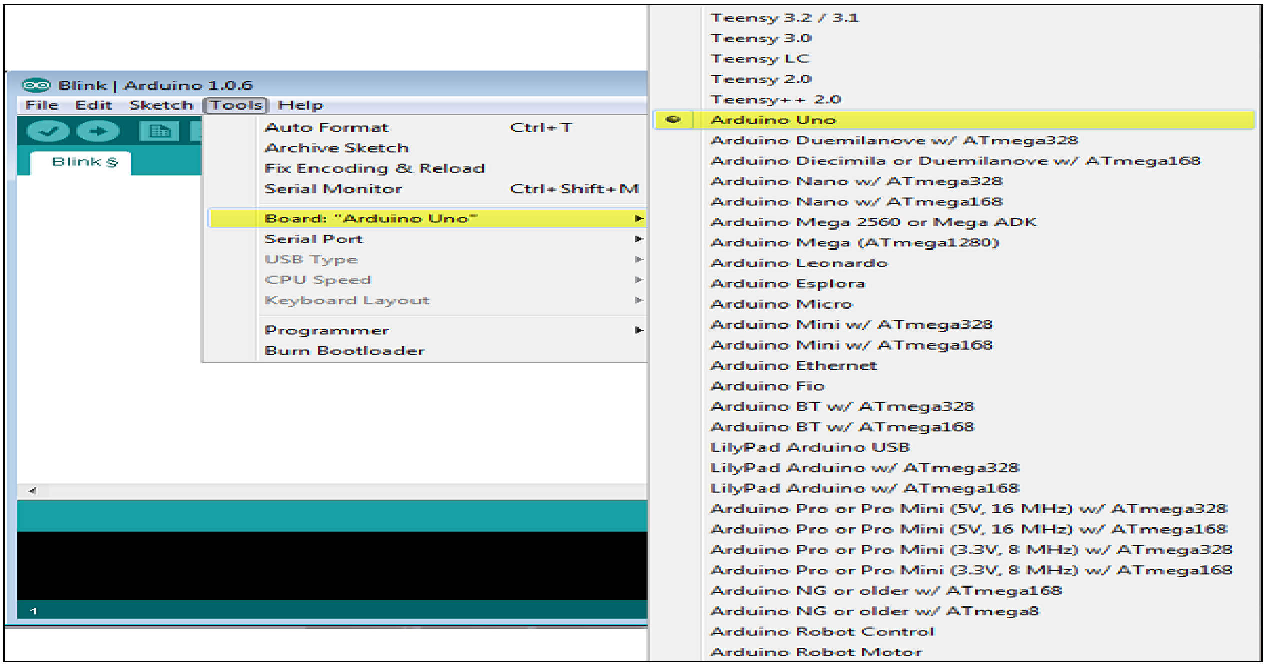


Figure-4.1.2 Select Your arduino IDE

We must avoid any errors when orm while uploading a program to the board, we must select the correct Arduino board name, which holds same with the board connected to your pc.

Go to Tools in th arduino → select your board and make the use of arduino

To set up your Arduino board, navigate to

Tools → Serial Port and choose the appropriate port,

typically COM3 or higher, ensuring that the board is connected.

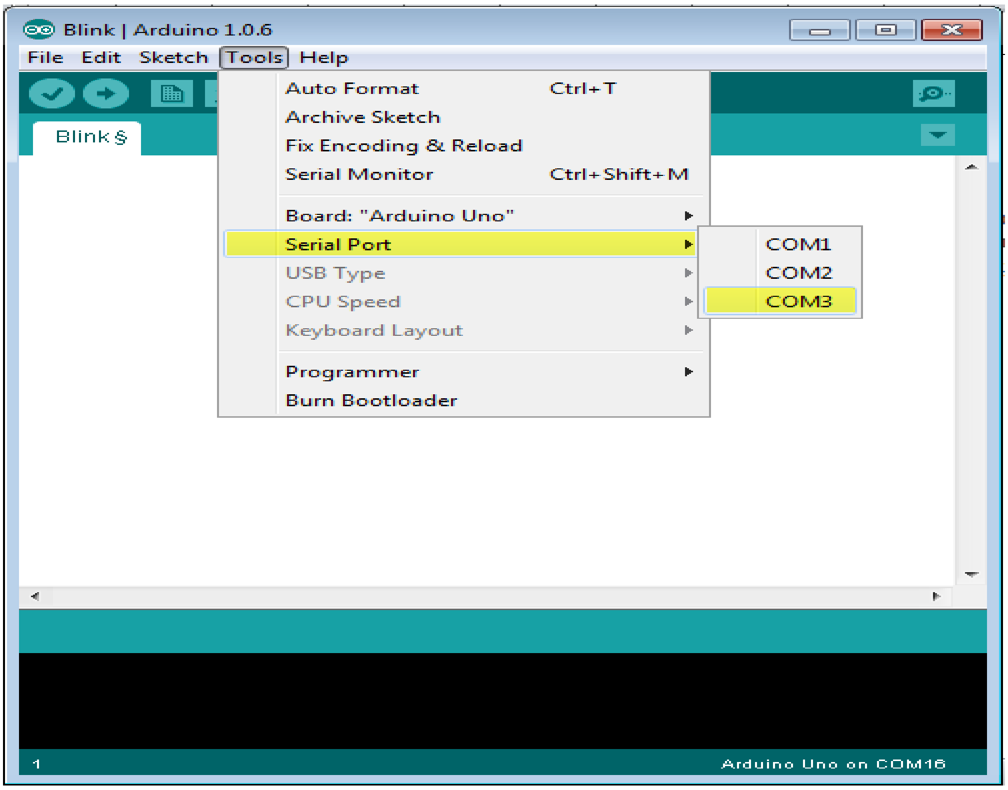


Figure-4.1.3: Selection of serial port

Upload the program

Figure-4.1.4.: for uploading the program

A – helpful to verify if there is any compilation error.

B − Used to upload a program to the Arduino board.

C – It is used to create a new sketch.

D − Used to directly open an example sketch.

E – made to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

4.2 BLYNK APP SETUP

4.2.1

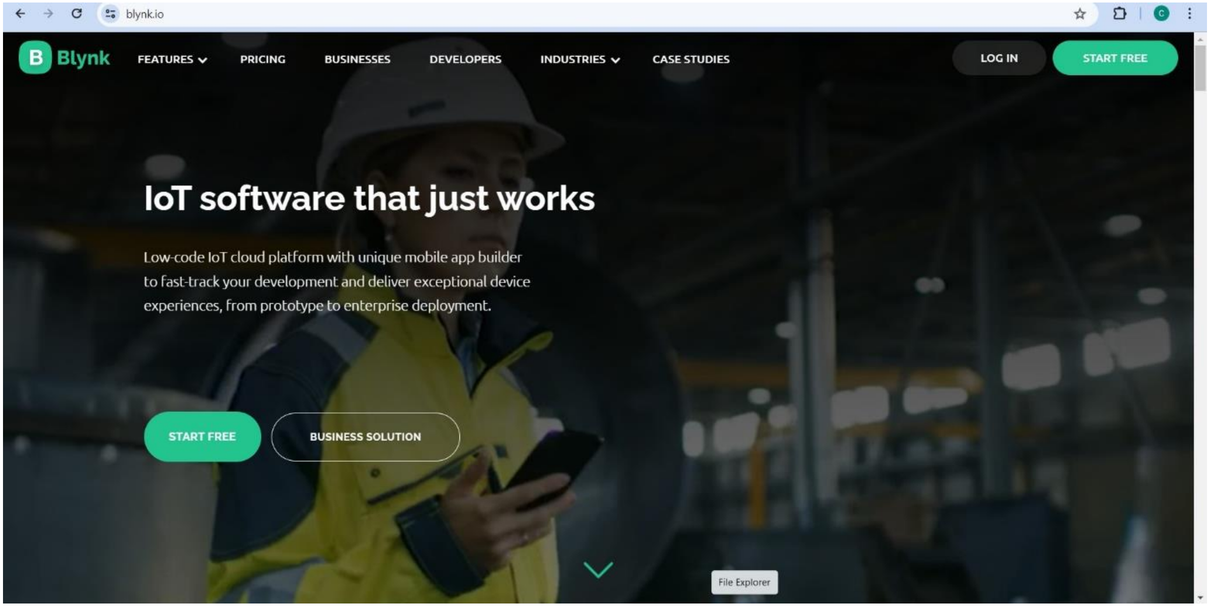


Figure-4.2.1 Blink software

Download and install the Blynk app from the Google Play Store or Apple App Store.

Open the Blynk app on your smartphone.

Create a new account or log in with your existing credentials.

Start a new project by selecting the "+" icon and choosing your device.

Add widgets to your project and configure them as needed.**[11]**

4.2.2

Trying to sign up or if already have a account try to login the page .

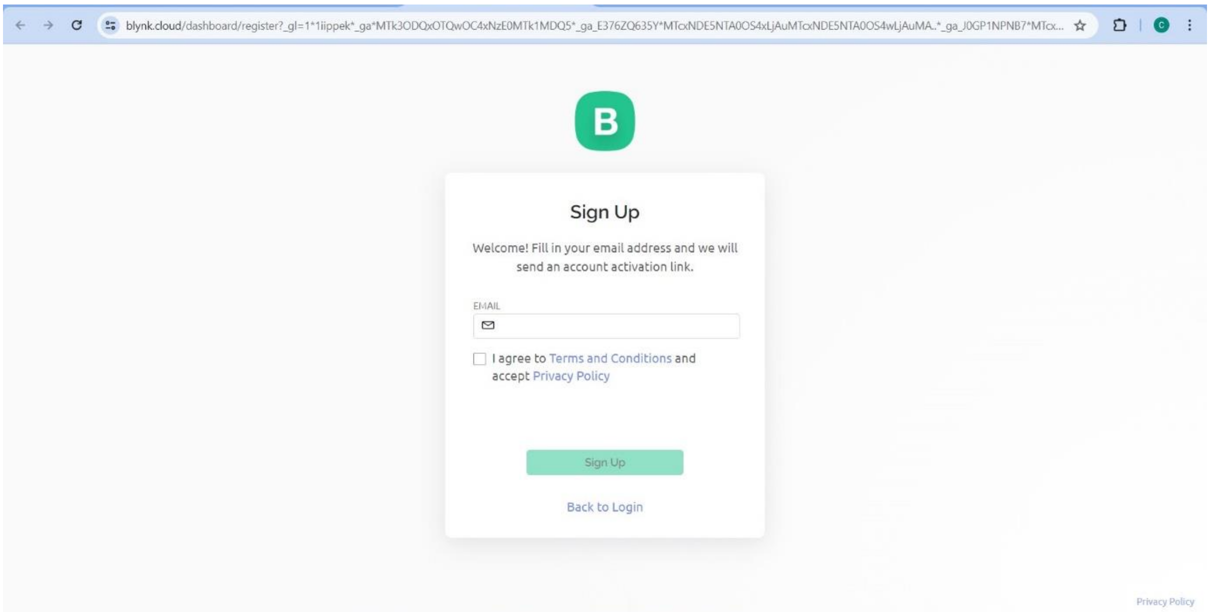


Figure-4.2.2: Sign up with devices

4.2.3

BLYNK CONSOLE ALONG WITH THE DEVICES

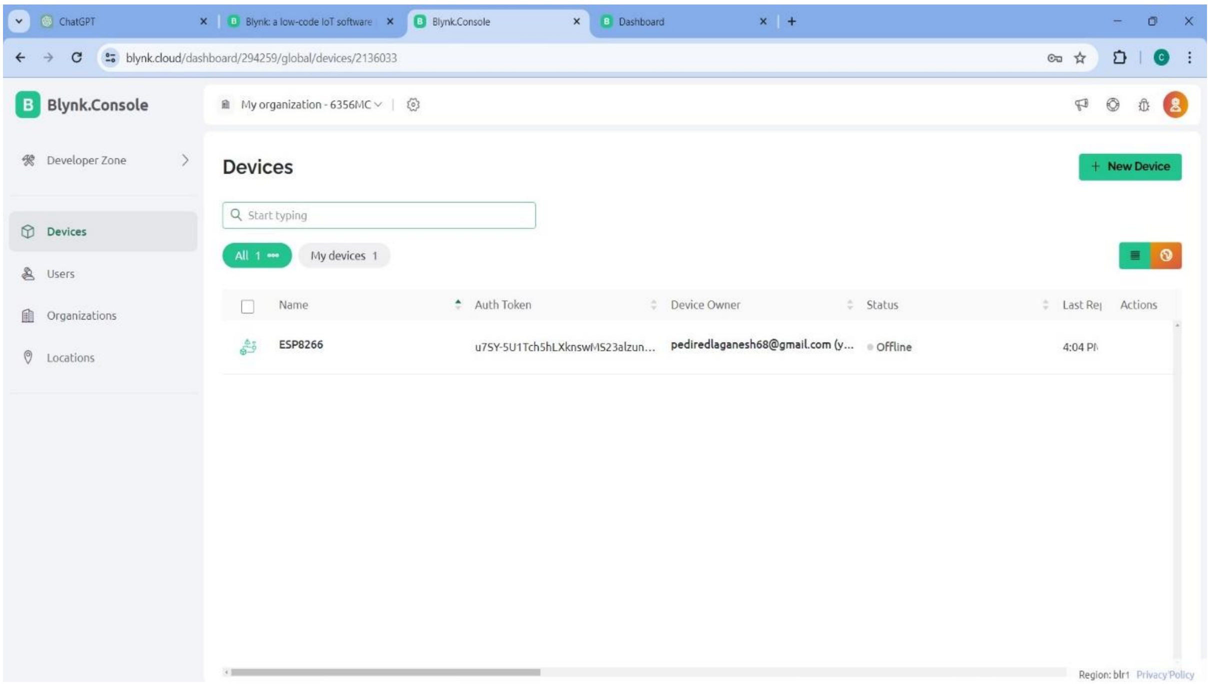


Figure-4.2.3: Dash Board

4.2.4 WEB DASH BOARD

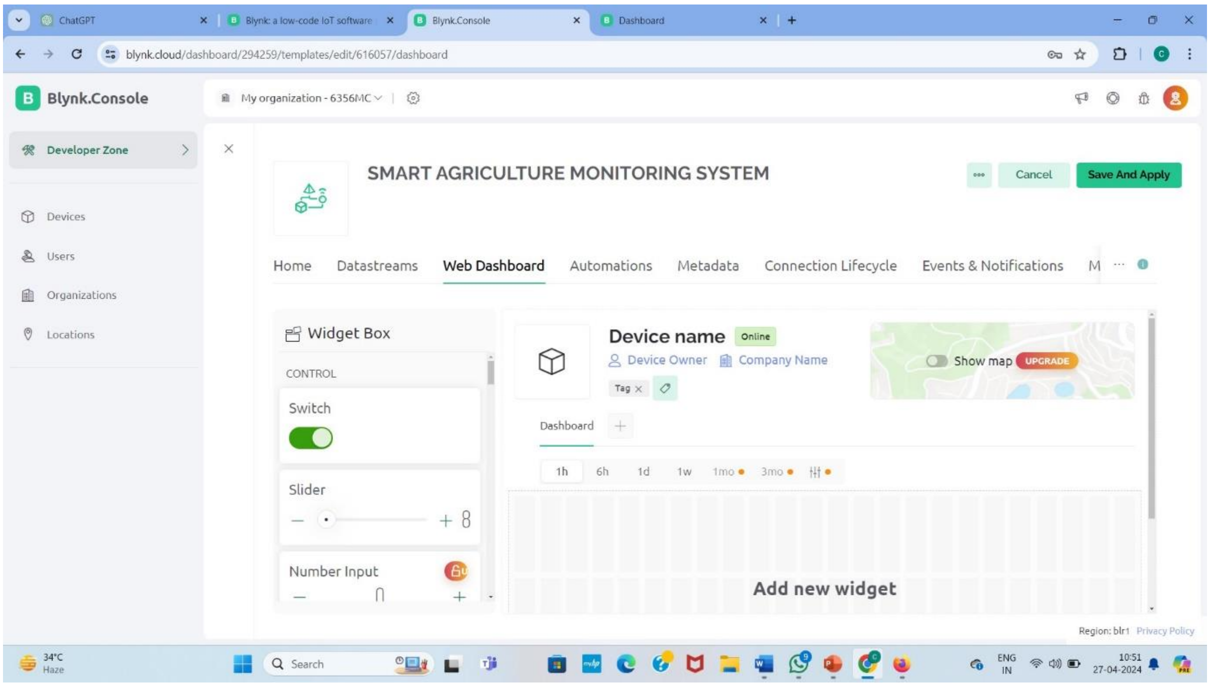


Figure-4.2.4: Web DashBoard

We can get the device name and control the project with a particular switch

4.2.5

DATASTREAMS

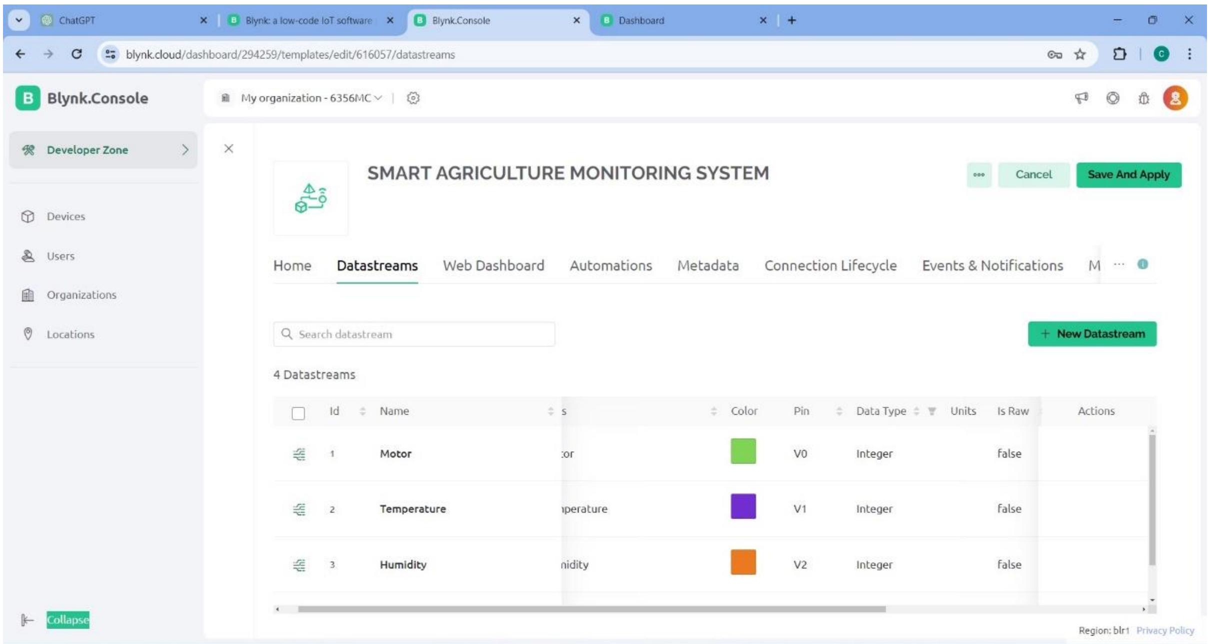


Figure-4.2.5:Data Streams

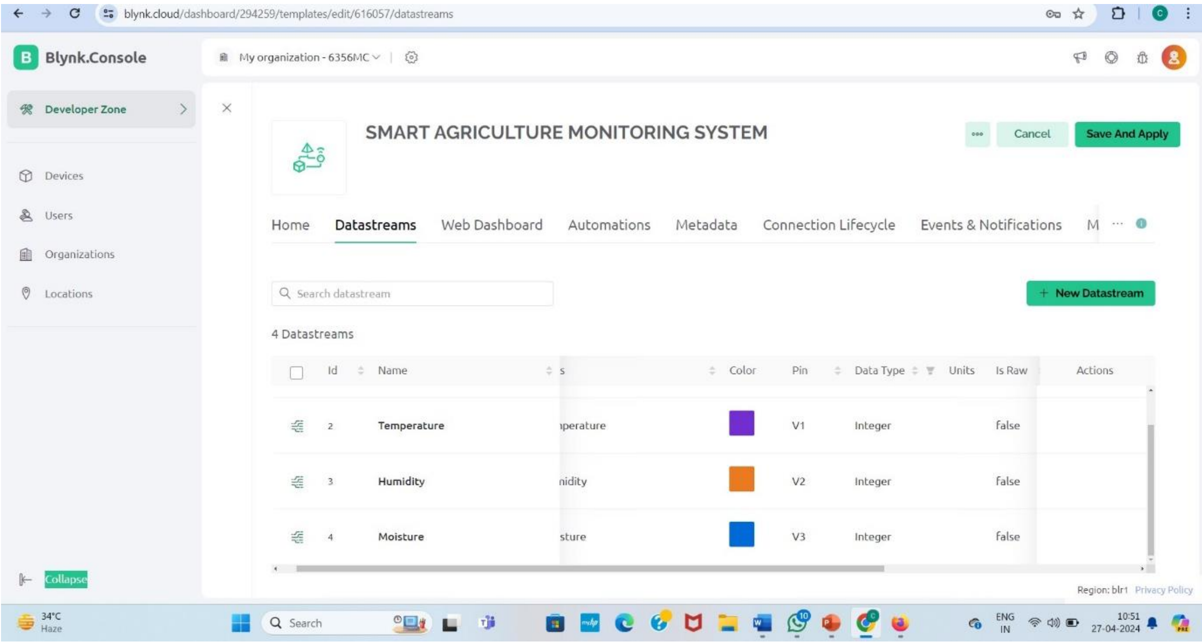


Figure-4.2.6

Here we get the components we use and data type used and the action taken can be found.You can configure these data streams in the Blynk app dashboard to visualize data through widgets like graphs, gauges, and displays. Blynk also allows for setting thresholds and notifications based on the data stream value

4.2.6 AUTOMATIONS

In the Blynk app, data streams facilitate the communication between your hardware devices and the Blynk cloud, enabling real-time data monitoring and control.**[12]**

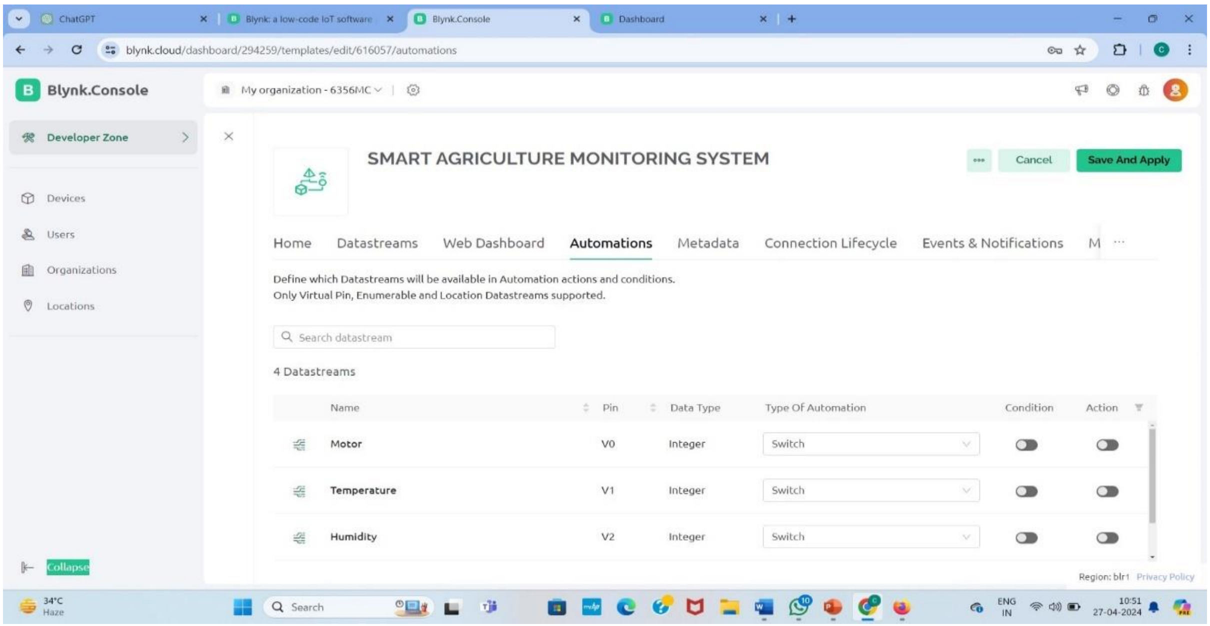


Figure-4.2.7:Automations

4.2.7 **CONNECTION LIFE CYCLE**

1. Initialization: The microcontroller and sensors power on and initialize.

2. Network Connection: The microcontroller connects to the internet and the Blynk cloud server using an authentication token.

3. Data Collection and Transmission: Sensors collect data, which is processed and sent to the Blynk cloud via virtual pins.

4. Data Visualization and Control: Users view real-time data on the Blynk app and control actuators based on the data.

5. Disconnection and Reconnection: The microcontroller attempts to reconnect and resume data transmission if the connection is lost.**[13]**

4.2.8 **EVENTS AND NOTIFICATIONS**

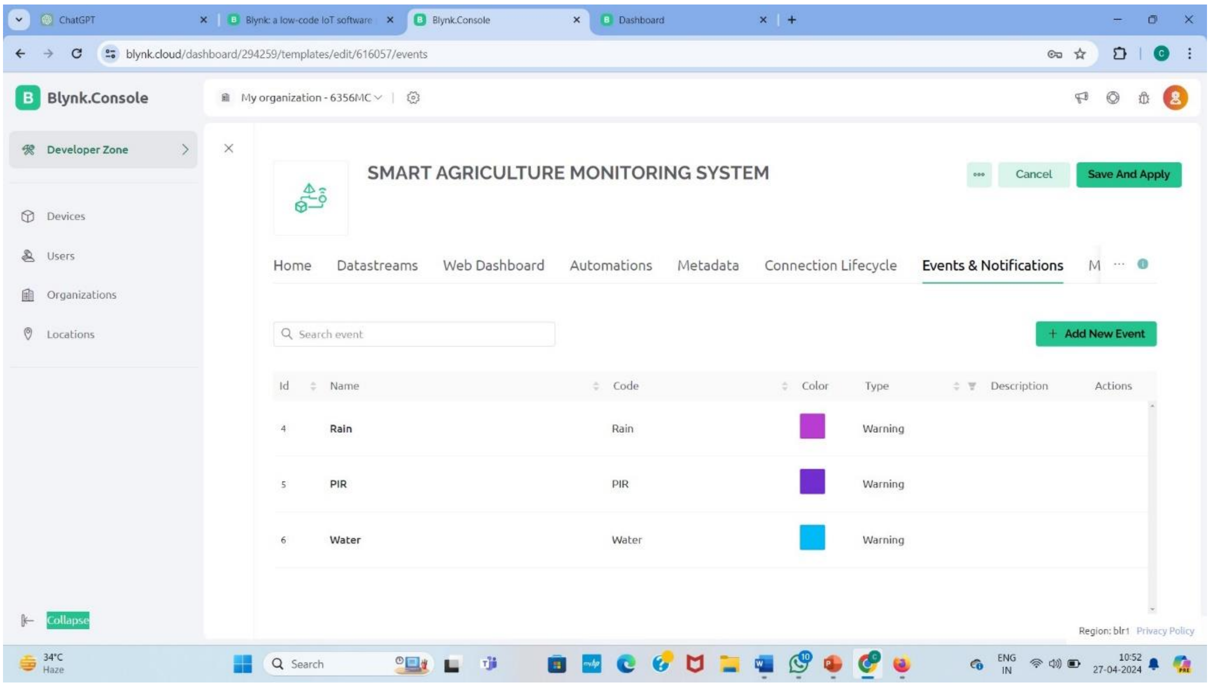


Figure-4.2.8: Events & Notifications

Users can configure specific conditions, such as sensor thresholds, to trigger alerts when those conditions are met. Real-time push notifications are sent to the user's smartphone to alert them of critical events like high soil moisture or low battery levels. Additionally, users can set up email alerts for detailed notifications and logs, and SMS notifications for urgent matters. The Blynk app also displays notifications directly within the interface, allowing users to quickly respond to changes and events.

4.2.9 **NOTIFICATIONS LIMITS**

These notifications can be customized to alert users about critical conditions, maintenance reminders, or routine updates.(figure-25) Email notifications can also be set up for detailed reports and logs, ensuring important information is communicated effectively. Additionally, in-app notifications provide real-time updates directly within the Blynk app, allowing users to monitor and respond to changes promptly.

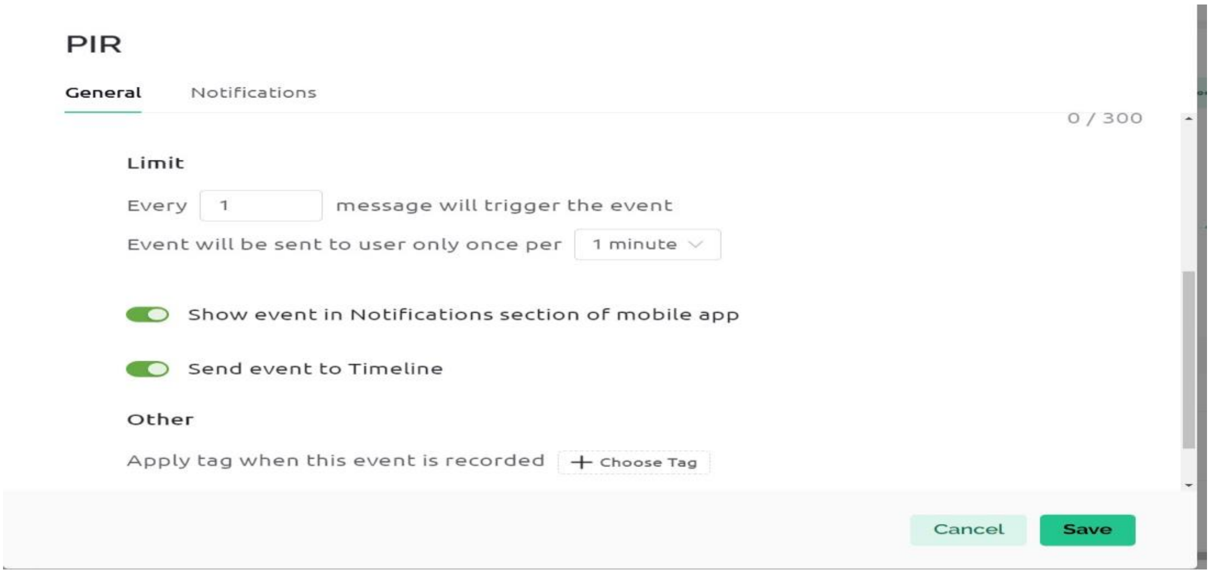


Figure-4.2.9 Notifications Limits

general notifications help users stay updated on their IoT project's status and events. Users can receive push notifications for various triggers, such as sensor readings crossing predefined thresholds or device connectivity changes. These notifications can be customized to alert users about critical conditions, maintenance reminders, or routine updates. Email notifications can also be set up for detailed reports and logs, ensuring important information is communicated effectively.

Add Notification Widget:

1. In the Blynk app, tap on the "+" icon to add a new widget.
2. Select the "Notification" widget.
3. Configure the Notification Widget:
4. Set up the conditions for notifications.
   For example, set up thresholds for soil moisture levels to send notifications when the soil is too dry or too wet.

NOTIFICATIONS:

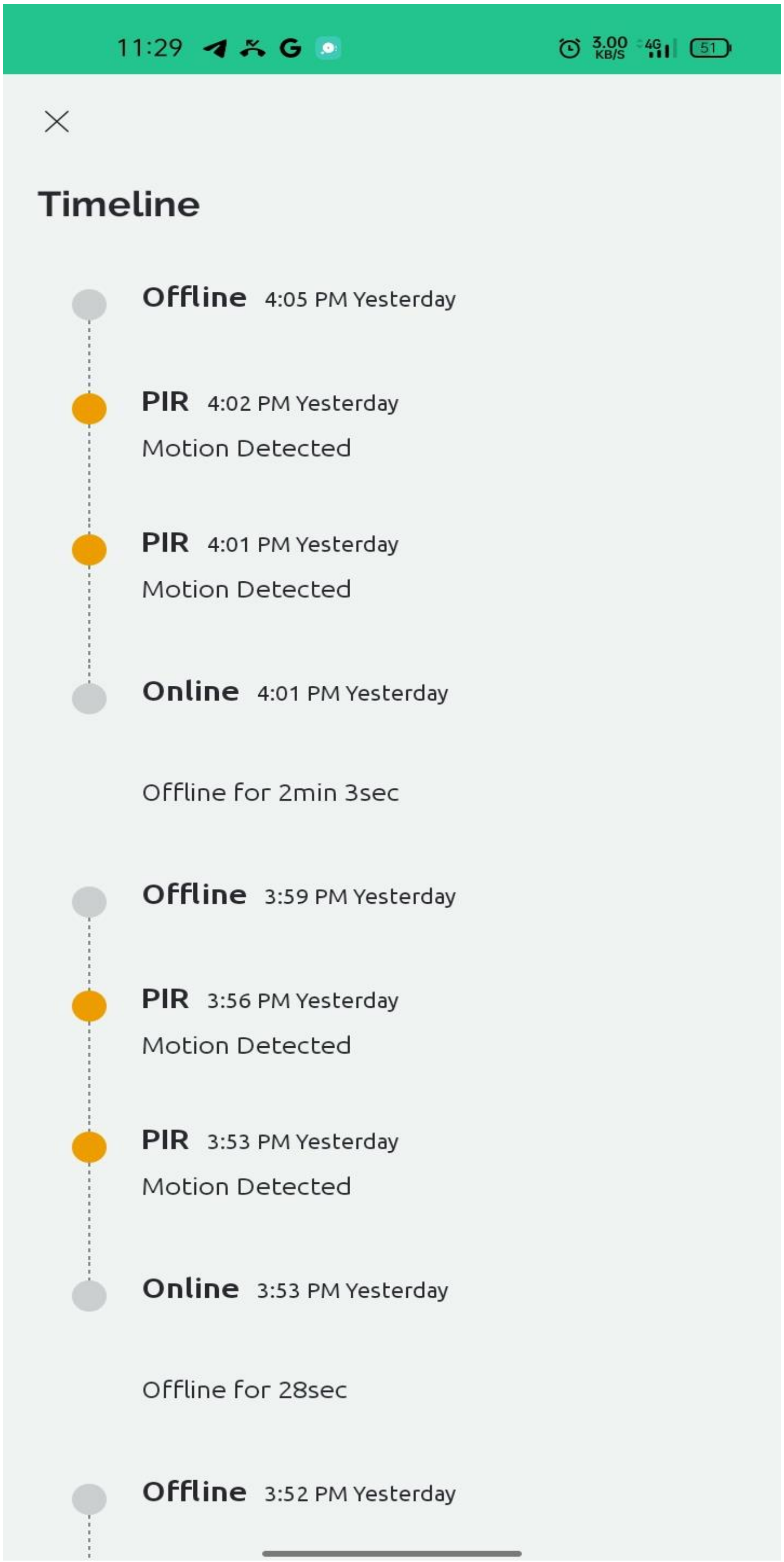


Figure -4.2.10 results

5.CIRCUIT DIAGRAM AND LIVE PICTURE

5.1 CIRCUIT DIAGRAM

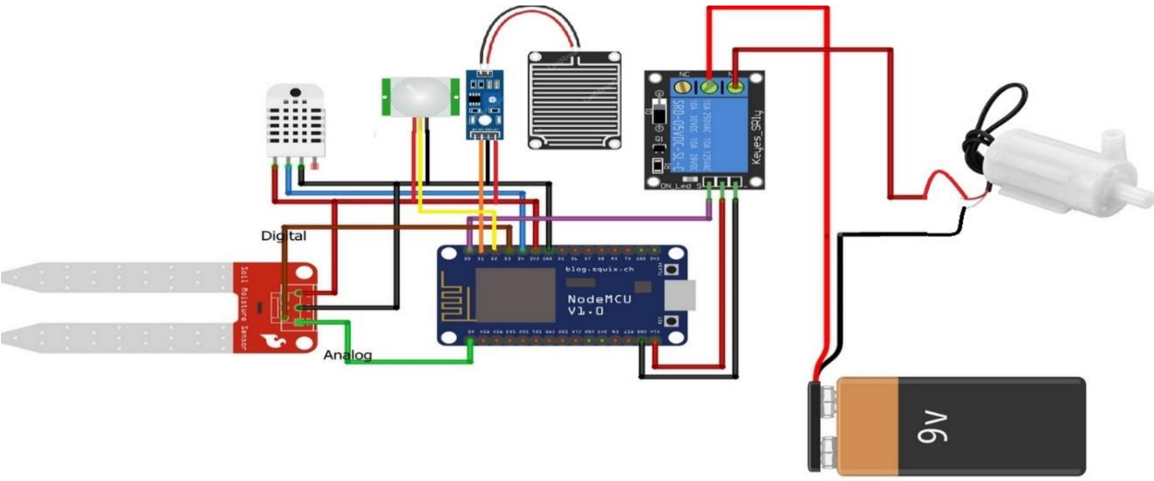


Figure-5.1 Circuit Diagram

5.2 LIVE PICTURE

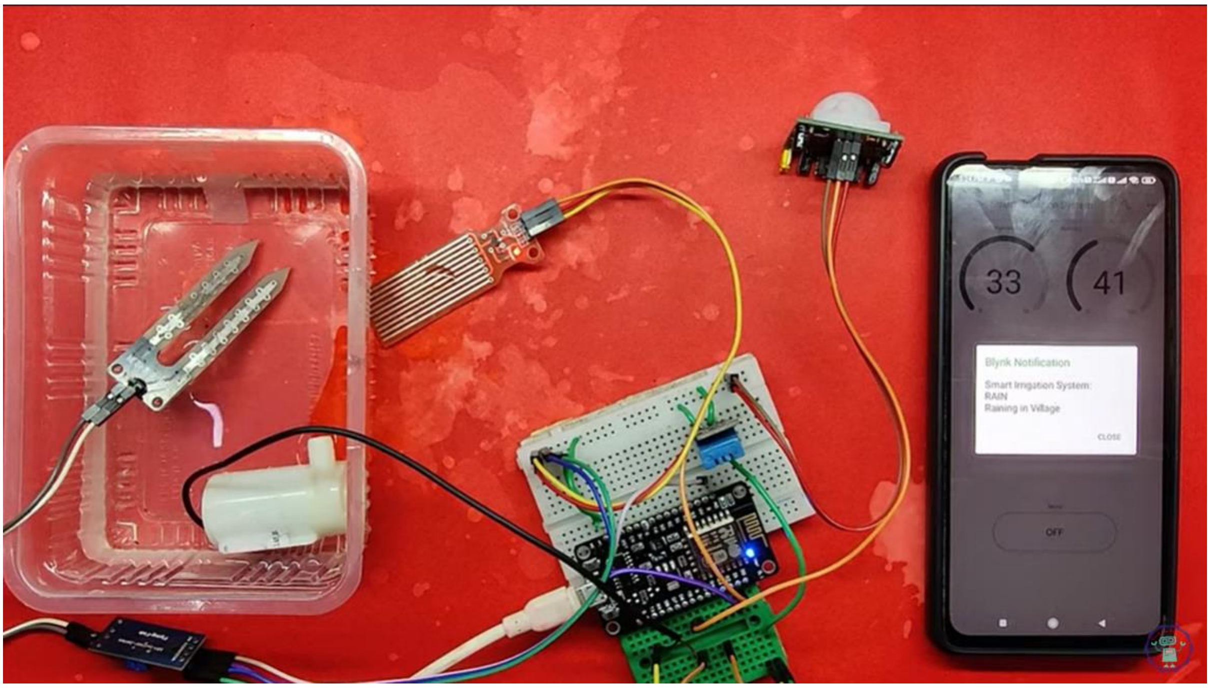


Figure-5.2 :It shows about Live Picture

6.RESULTS

6.1 RESULTS

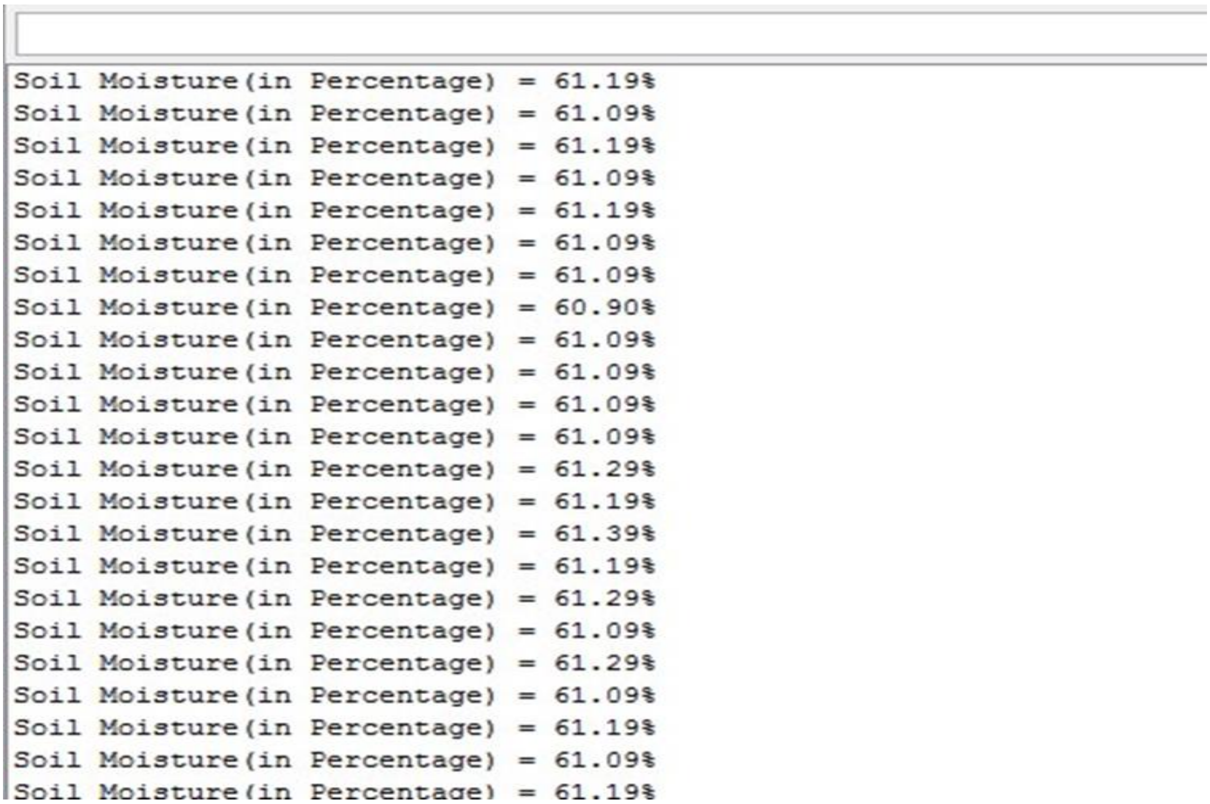


Figure-6.1 Results In Serial Monitor

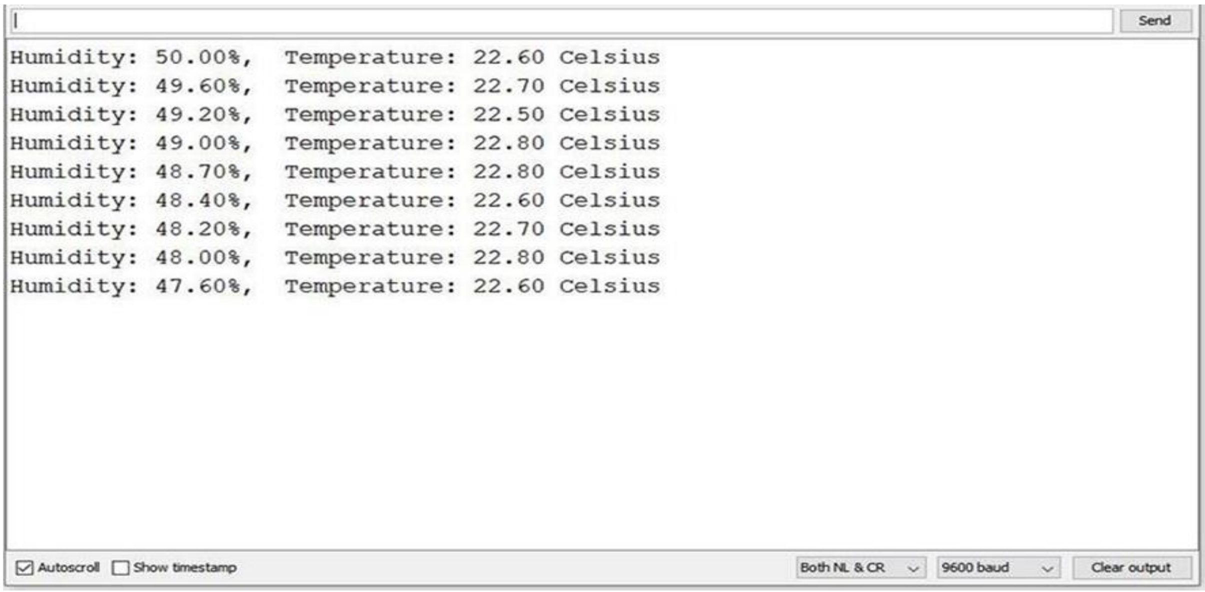


Figure -6.2 Results of Humidity and Temperature

6.2.IN BLINK APP RESULT

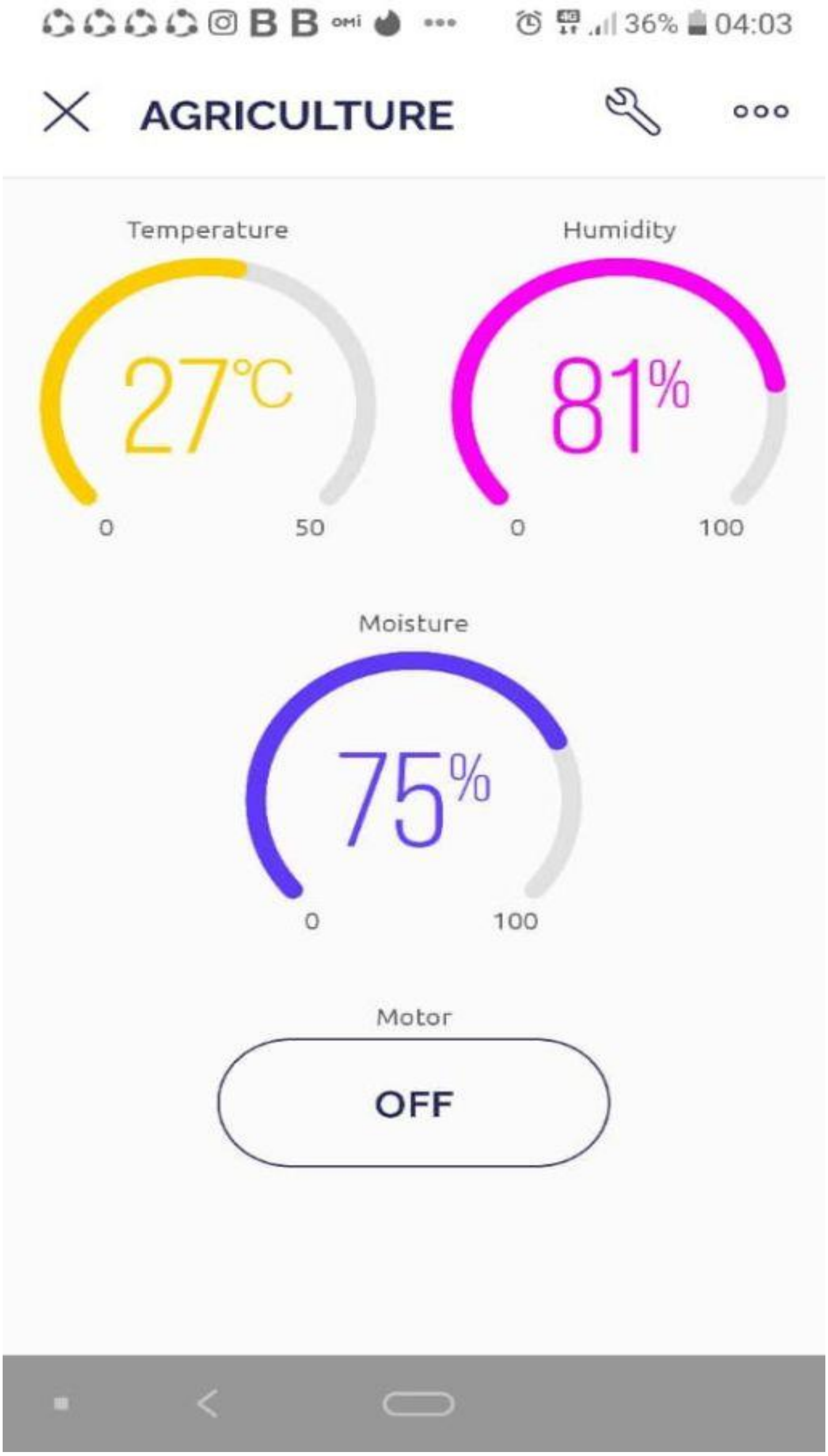


Figure-2..1 Results In Blink APP

7.FUTURE SCOPE

In other words, for successful IoT software development, it's essential to prioritize testing to ensure a high-quality end product and mitigate potential faults arising from complexity or hardware-software interactions. Continuous monitoring throughout development is crucial to catch and address issues promptly, minimizing errors and failures. Management should take accountability and utilize analytics to assess risks and maintain quality standards, especially regarding security. Lastly, choosing a competent IoT solution provider is paramount for achieving desired outcomes and ensuring the success of IoT applications.**[14]**

For the purpose of avoiding errors and failures, continuous monitoring is essential during the development of Internet of Things software.

Management must accept accountability. Analytics must guarantee quality and evaluate software risks because any carelessness in the creation of IoT applications can result in security issues that can be very expensive to fix

8.CONCLUSION

In conclusion, the development of a smart agriculture system signifies a transformative shift in the agricultural sector, leveraging advanced technologies to enhance productivity, efficiency, and sustainability. By integrating Internet of Things (IoT) devices, sensors, and automated systems, farmers can monitor and manage their crops and livestock with unprecedented precision. This leads to optimized resource usage, reduced environmental impact, and improved yield quality. The real-time data analytics and predictive insights provided by smart agriculture systems empower farmers to make informed decisions, adapt to changing conditions, and ultimately increase their profitability. As technology continues to evolve, the potential for smart agriculture to revolutionize traditional farming practices and contribute to global food security becomes increasingly evident. Embracing these innovations will be crucial for meeting the demands of a growing population and ensuring a sustainable future for agriculture.

REFERENCES

1.Smart Agriculture: A Sustainable Approach Using IoT and Data Analytics" by Mritunjay Singh, Nibedita Ray-Bennett, and Tuan Dinh Le. This book discusses the integration of IoT and data analytics in agriculture to enhance sustainability and efficiency.

2.Internet of Things and Data Analytics Handbook" edited by Hwaiyu Geng. This handbook provides comprehensive coverage of IoT applications, including smart agriculture, and explains how data analytics can be used to improve agricultural practices.

3."Internet of Things (IoT) for Smart Agriculture: Opportunities and Challenges" by M. R. Abdmeziem, D. Tandjaoui, and I. Romdhani. This paper explores the opportunities and challenges of using IoT in agriculture and provides case studies of successful implementations.

4. IEEE Symposium on Industrial Electronics & Applications

5.IEEE Xplore Digital Library: Search for papers and articles on smart agriculture. IEEE Xplore

6. https://www.nodemcu.com/index_en.html

7. https://builtin.com/articles/iot-devices#:~:text=IoT%20devices%20are%20hardware%20that%20connect%20wirelessly%20to%20the%20internet,or%20even%20a%20smart%20refrigerator.

8.ResearchGate: A network where researchers publish and share their work. Search for smart agriculture systems. ResearchGate

9.SpringerLink: Provides access to scientific documents, including books and journal articles on smart agriculture. SpringerLink

10. https://www.arduino.cc/en/software

11. https://blynk.io/

12.https://www.edrawmax.com/app/ai-diagram/?needShareGive=true*nativeId=56d8c5a16687470fb151359d03871bf0*sharerId=5620771

13. https://www.nifa.usda.gov/grants/programs/precision-geospatial-sensor-technologies-programs/precision-agriculture-crop-production

14. https://www.javatpoint.com/iot-in-agriculture

SOURCE CODE

```
// Project Name : Smart Agriculture Monitoring System Using UV Light

// Platform : Blynk Mobile App

#define BLYNK_TEMPLATE_ID "TMPL39IOQY6aX"

#define BLYNK_TEMPLATE_NAME "SMART AGRICULTURE MONITORING
SYSTEM"

#define BLYNK_AUTH_TOKEN "u7SY-5U1Tch5hLXknswMS23alzun6cva"

#define BLYNK_PRINT Serial

#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

#include "DHT.h"

char auth[] = BLYNK_AUTH_TOKEN;

// Set password to "" for open networks.

char ssid[] = "GS";  // type your wifi name

char pass[] = "12341234";  // type your wifi password

#define DHTPIN  D4

#define rainSensor D1

#define moistureSensor D3

#define analogreading A0

#define PIR D2

#define motor D0

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);
```

```
BlynkTimer timer;
int moistureState =0;
int lastmoisState=0;
int rainState = 0;
int lastRainState = 0;
int MotorState=0;
int lastPIRState = 0;
int PIRState=0;
unsigned long old =0;
unsigned long current =0;
int interval=0;
BLYNK_CONNECTED() {
  Blynk.syncVirtual(V0);
}
BLYNK_WRITE(V0)
{
MotorState = param.asInt();
if (MotorState==1)
{
digitalWrite(motor,HIGH);
Serial.println("Motor ON");}
else{
digitalWrite(motor,LOW);
```

```
Serial.println("Motor OFF");}}
void Reading()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  moistureState  = digitalRead(moistureSensor);
  PIRState  = digitalRead(PIR);
  delay(10);
  Serial.println("PIRSTATTE: "+String(PIRState));
  rainState = digitalRead(rainSensor);
  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
Blynk.virtualWrite(V2, h);  //V2 is for Humidity
  Blynk.virtualWrite(V1, t);  //V1 is for Temperature
int i = analogRead(analogreading);
i=4095-i;
i= map(i, 0, 4095, 0, 100);
// Serial.println ("AnalogReading: " + String (i) );
Blynk.virtualWrite(V3, i);
 if (moistureState == 1 && lastmoisState == 0) {
  Serial.println("Blynk Notification, needs water, send notification");
```

```
  Blynk.logEvent("WATER","Water your plants");
  lastmoisState = 1;
  delay(1000);}
else if (moistureState == 0 && lastmoisState == 1) {
  Serial.println("Water is sufficient");
  lastmoisState=0;
  }
  else if (rainState == 1 && lastRainState == 0) {
  Serial.println("Blynk notification: Its Raining!");
  Blynk.logEvent("RAIN","Raining in Village");
  lastRainState = 1;
  delay(1000); }
 else if (rainState == 1 && lastRainState == 1) {
  Serial.println("has not been watered yet");
delay(1000);
  Blynk.virtualWrite(V0,  LOW);
  }
else if (PIRState == 1 && lastPIRState == 0) {
  Serial.println("Blynk Notification,PIR Object detected");
 Blynk.logEvent("PIR","Motion Detected");
  lastPIRState = 1;
  delay(1000);}
if (rainState == 0) {
```

```
  Serial.println("Its Raining Continuously!");
   lastRainState = 0;  }
 else if (moistureState == 1 && lastmoisState == 1) {
 Serial.println("NO RAIN"); }
current=millis();
interval=(current-old)/1000;
  if (interval > 60 && PIRState==0)
  {
   interval=0;
   old=current;
    lastPIRState = 0;
    Serial.println("PIR Reactivating...");
    delay(3000);
  }
}
void setup()
{
 Serial.begin(115200);
 Serial.println();
  delay(2000);
 Serial.println("Pleae wait.... DHT Sensor activation");
 dht.begin();
 delay(3000);
```

```
  Serial.println("Please wait for Blynk Server connection");
  pinMode(moistureSensor, INPUT);
  pinMode(rainSensor, INPUT);
  pinMode(PIR, INPUT);
  pinMode(analogreading,INPUT);
  pinMode(motor,OUTPUT);
  Blynk.begin(auth, ssid, pass);
  Blynk.virtualWrite(V0,  MotorState);
  timer.setInterval(1000L, Reading);
}
void loop()
{
  Blynk.run();
  timer.run(); }
```

COURSE OUTCOME (COs) ATTAINMENT

➢ **Expected Course Outcomes (COs):**

(Refer to COs Statement in the Syllabus)

____

____

____

____

➢ **Course Outcome Attained:**

How would you rate your learning of the subject based on the specified COs?

1 2 3 4 5 6 7 8 9 10

LOW **HIGH**

➢ **Learning Gap (if any):**

____

____

____

____

➢ **Books / Manuals Referred:**

____

____

____

____

Date: **Signature of the Student**

➢ **Suggestions / Recommendations:**

(By the Course Faculty)

____

____

____

____

Centurion
UNIVERSITY
Shaping Lives...
Empowering Communities...

CENTURION UNIVERSITY OF TECHNOLOGY AND MANAGEMENT,

ANDHRA PRADESH

CAMPUSES:

Bhubaneswar Campus

Vizianagaram Campus
Tekkali Village,
Nelimarla Mandal,
Vizianagaram Pin: 535003