IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A PERSONALIZED PLANT MONITORING SOLUTION WITH ESP32

B. SRIVIDYA
Electronics and Communication Engineering
HITAM college
Hyderabad, India

P. Gopi Krishna
Electronics and Communication Engineering
HITAM college
Hyderabad, India

G. Shashikanth
Electronics and Communication Engineering
HITAM college
Hyderabad, India

I. Giri Naveen Sai
Electronics and Communication Engineering
HITAM college
Hyderabad, India

***Abstract*__**The Internet of Things (IoT) is revolutionizing the agriculture industry, empowering farmers to address significant challenges. From livestock monitoring and conservation to plant & soil monitoring, IoT offers solutions that improve the quality, quantity, sustainability, and cost-effectiveness of agricultural production. Today's farms, regardless of size, can leverage IoT to remotely monitor sensors that detect soil moisture, crop growth, and pests. This information allows for smart control of connected harvesters and irrigation equipment. This project specifically focuses on monitoring soil parameters like moisture, temperature, and electrical conductivity. Automated irrigation based on these parameters is controlled by a microcontroller. The user receives text message notifications for any deviations from expected values, ensuring complete system health. Additionally, the project incorporates plant pest detection for comprehensive monitoring Greenhouses, controlled environments for plant cultivation, are another area where IoT can significantly impact. This project aims to design a simple, low-cost Arduino-based system for continuous monitoring and control of environmental parameters. This ensures optimal conditions for plant growth and yield.

Keywords— Internet of things (IOT), Arduino IDE, Microcontroller, moisture sensors, ESP32.

I. INTRODUCTION

The Internet of Things (IoT) is swiftly reshaping our reality, seamlessly connecting devices and erasing the boundaries between the physical and digital realms. This extensive network extends its influence into the agricultural domain, where embedded sensors and actuators in farming equipment and even livestock gather and transmit real-time data through the internet. This opens up a wealth of possibilities, facilitating automation, intelligent decision-making, and a deeper comprehension of the environment.

At the forefront of this agricultural transformation, the ESP8266 IoT Automatic Irrigation System serves as a prime illustration of the revolutionary potential of this technology. Departing from conventional irrigation methods, this system adopts a data-centric approach, utilizing real-time information on soil moisture, temperature, and pest activity to fine-tune irrigation schedules, reduce water consumption, and optimize crop yields. This evolution toward smart agriculture transcends mere technological progress; it signifies a fundamental shift in our perspective on food production.

Harnessing the capabilities of IoT allows us to meet the escalating demand for food while upholding responsible resource management and environmental preservation. Embracing automation, data-driven insights, and targeted interventions, smart agriculture charts the course for a future where sustainable practices and technological ingenuity seamlessly integrate into the framework of agricultural processes. This vision presents a compelling resolution to responsibly feed the world and secure a flourishing future for generations to follow.

II. REVIEW FOR LITERATURE

The current scenario, marked by declining water tables, depletion of rivers and tanks, and unpredictable environmental conditions, underscores the critical necessity for the judicious utilization of water resources. In India, approximately 35% of land is consistently irrigated, with two-thirds relying on monsoons for water supply. Irrigation serves as a pivotal solution, diminishing dependence on monsoons, enhancing food security, and boosting agricultural productivity, thereby creating more employment opportunities in rural areas. Farmers grapple with challenges in the watering system, particularly determining the appropriate amount and timing of water supply. Overwatering not only damages crops but also results in water wastage. To circumvent such issues, maintaining an optimal water level in the soil is imperative.

This study introduces a system comprising humidity sensors, moisture sensors, and temperature sensors strategically positioned in the plant's root zone. The gateway unit, implemented through ESP8266, manages the sensor data and transmits it to an Android application. This application is designed to measure approximate values from temperature sensors, humidity sensors, and moisture sensors programmed into a microcontroller, ultimately regulating the quantity of water supplied.

III. RELATED WORK

Existing research in the field focuses on using sensor technologies, such as humidity, moisture, and temperature sensors, to optimize agricultural irrigation. Studies explore strategic sensor placement in the plant root zone for real-time monitoring of soil conditions. Integration of gateway units like ESP8266 is also studied for efficient data collection and transmission. Additionally, mobile applications are developed to enable remote monitoring and control of irrigation systems, empowering farmers to make informed decisions based on sensor data. Overall, related work emphasizes leveraging technology for sustainable and efficient water management in agriculture.

IV. METHODOLOGY ON PRESENT INVESTIGATION

4.1 Proposed Methodology

FC-28 sensor monitors soil moisture, triggering the SSR to activate the Submersible Pump for precise irrigation. Blynk app allows remote monitoring and control, optimizing water usage and crop yield.

4.2 Components list

1.ESP32 DevKitM-1

2.FC-28 Soil Moisture Sensor

3.BME280 Temperature Humidity Pressure Sensor,

4.Solid State Relay (SSR)

5.Submersible Water Pump

6.PIR Sensor

4.3 Software Requirement

1.Blynk Application

2.Arduino IDE

4.2.1 ESP32 DevKitM-1

ESP32-DevKitM-1 is an ESP32-MINI-1/1U-based development board produced by Espressif. Most of the I/O pins are broken out to the pin headers on both sides for easy interfacing .Users can either connect peripherals with jumper wires or mount ESP32-DevKitM-1 on a breadboard.

Fig-4.2.1: ESP 32 DevKitM-1

4.2.2. FC-28 Soil Moisture Sensor

The FC-28 Soil Moisture Sensor is a simple breakout for measuring the moisture in soil and similar materials. The soil moisture sensor is straight forward to use. The two large, exposed pads function as probes for the sensor, together acting as a variable resistor.

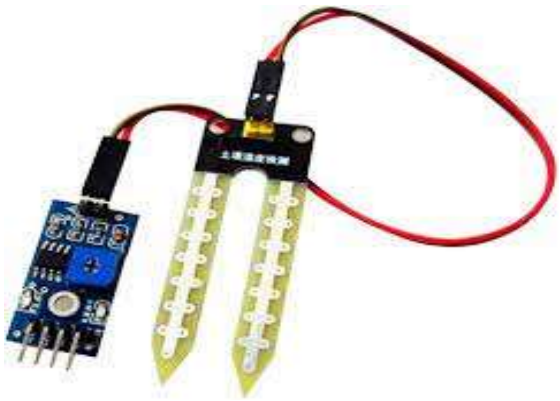


Fig-4.2.2: FC-28 Soil Moisture Sensor

4.2.3. BME280 Temperature Humidity Pressure Sensor

The BME280 is a humidity sensor especially developed for mobile applications and wearables where size and low power consumption are key design parameters. The unit combines high linearity and high accuracy sensors and is perfectly feasible for low current consumption, long-term stability and high EMC robustness.

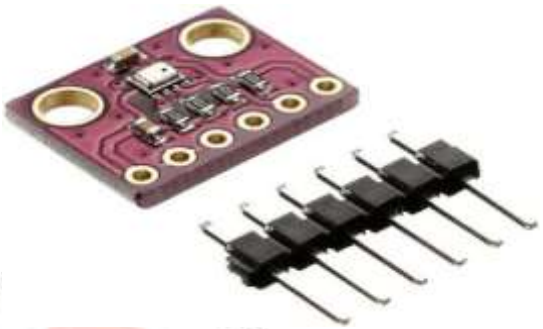


Fig-4.2.3: BME280 Temperature Humidity Pressure Sensor

4.2.4 Solid State Relay (SSR)

Solid State Relays (SSRs) are electronic switching devices that use semiconductor components to perform the same function as traditional electromechanical relays. This title could serve as a starting point for a comprehensive discussion about SSRs, covering their advantages over traditional relays, how they operate, and various applications in different industries.

Fig-4.2.4: Solid State Relay (SSR)

4.2.5. Submersible Water Pump

Submersible water pumps designed for direct current (DC) operation. Topics that could be covered include the design principles of DC submersible pumps, their various applications such as in agriculture, aquaculture, or domestic water systems, and a discussion on the efficiency and advantages of using DC-powered pumps in underwater settings.

Fig-4.2.5: Submersible water pump

4.2.6 PIR Sensor

A **passive infrared sensor** (**PIR sensor**) is an electronic sensor that measures infrared (IR) light radiating from objects in its field of view. They are most often used in PIR-based motion detectors. PIR sensors are commonly used in security alarms and automatic lighting applications.

PIR sensors detect general movement, but do not give information on who or what moved. For that purpose, an imaging IR sensor is required.

PIR sensors are commonly called simply "PIR", or sometimes "PID", for "passive infrared detector". The term *passive* refers to the fact that PIR devices do not radiate energy for detection purposes. They work entirely by detecting infrared radiation (radiant heat) emitted by or reflected from objects.

Fig-4.2.6: PIR Sensor

4.3.1. Blynk Application

Blynk is a comprehensive software suite that enables the prototyping, deployment, and remote management of connected electronic devices at any scale.

Whether it's personal IoT projects or commercial connected products in the millions, Blynk empowers users to connect their hardware to the cloud and create iOS, Android, and web applications, analyze real-time and historical data from devices, remotely control them from anywhere, receive important notifications, and much more.

Fig-4.3.1: Blynk Application

4.3.2. Arduino IDE

Arduino IDE is an open-source software, designed by Arduino.cc and mainly used for writing, compiling & uploading code to almost all Arduino Modules.

Fig-4.3.2: Arduino IDE

V. CONNECTION MECHANISM

A personalized plant monitoring solution utilizing the ESP32 microcontroller entails a sophisticated connection mechanism to seamlessly integrate various components. The initial step involves connecting plant monitoring sensors—such as soil moisture, temperature, humidity, and light sensors—to the designated GPIO pins of the ESP32. Attention to detail is crucial during the wiring process, ensuring proper pin configuration and avoiding conflicts. A reliable power supply, be it batteries, a power adapter, or alternative sources, is then connected to sustain continuous operation. Communication protocols, such as I2C, SPI, or UART, are implemented to facilitate data transfer between the ESP32 and the sensors. The heart of the system lies in the control mechanism, where programming logic is employed to process sensor data, make informed decisions, and execute actions based on the monitored information. Leveraging the ESP32's built-in wireless capabilities, whether Wi-Fi or Bluetooth, enables remote monitoring and control of the plant system. Microcontroller programming, utilizing languages like Arduino or Micro Python, is essential for efficient system operation. For user interaction, a web-based dashboard or mobile app can be designed as a user interface. Rigorous testing, sensor calibration, and security considerations are imperative to ensure the robustness and reliability of the personalized plant monitoring solution.

VI.BLOCK DIAGRAM

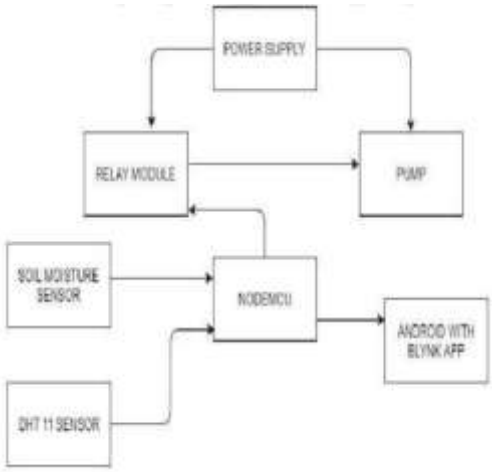
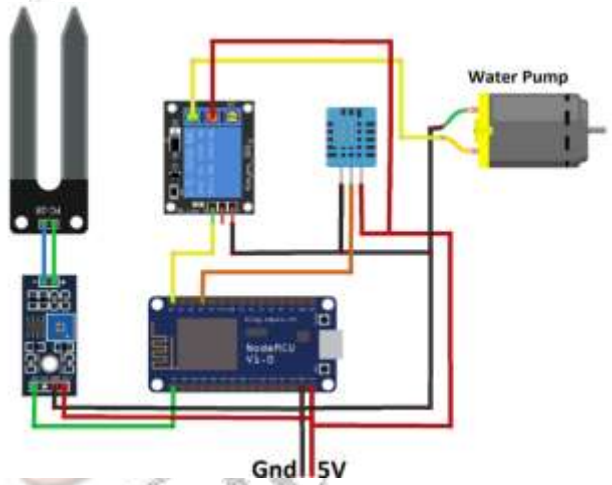


Fig-6.1 Block Diagram

VII.CIRCUIT DIAGRAM



VIII.SOURCE CODE

```
// Blynk Smart Plant Monitoring System

/* Connections

Relay. D3

Btn.  D7

Soil.  A0

PIR.  D5

SDA.  D2

SCL.  D1

Temp.  D4

*/


//Include the library files
```

```
#define BLYNK_TEMPLATE_ID "TMPL3BA2AWggD"
#define BLYNK_TEMPLATE_NAME "Smart Plant"
#define BLYNK_AUTH_TOKEN "Be2TiAAaGtBOMTh1ZOe0lEBOUUsxa7ge"
#include <LiquidCrystal_I2C.h>
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>

//Initialize the LCD display
LiquidCrystal_I2C lcd (0x27, 16,2);
char auth[] = "Be2TiAAaGtBOMTh1ZOe0lEBOUUsxa7ge"; // Enter your Blynk Auth token
char ssid[] = "GT NEO 3T"; //Enter your WIFI SSID
char pass[] = "gns12345"; //Enter your WIFI Password

DHT dht(D4, DHT11);//(DHT sensor pin,sensor type) D4 DHT11 Temperature Sensor
BlynkTimer timer;

//Define component pins
#define soil A0 //A0 Soil Moisture Sensor
#define PIR D5 //D5 PIR Motion Sensor
int PIR_ToggleValue;

void checkPhysicalButton();
int relay1State = LOW;
int pushButton1State = HIGH;
#define RELAY_PIN_1 D3 //D3 Relay
#define PUSH_BUTTON_1 D7 //D7 Button
#define VPIN_BUTTON_1 V12
//Create three variables for pressure
double T, P;
char status;
void setup() {
 Serial.begin(9600);
 lcd.begin();
 lcd.backlight();
 pinMode(PIR, INPUT);

 pinMode(RELAY_PIN_1, OUTPUT);
 digitalWrite(RELAY_PIN_1, LOW);
 pinMode(PUSH_BUTTON_1, INPUT_PULLUP);
 digitalWrite(RELAY_PIN_1, relay1State);
 Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
 dht.begin();
lcd.setCursor(0, 0);
 lcd.print(" Initializing ");
 for (int a = 5; a <= 10; a++) {
  lcd.setCursor(a, 1);
  lcd.print(".");
  delay(500);
 }
lcd.clear();
 lcd.setCursor(11, 1);
 lcd.print("W:OFF");
 //Call the function
 timer.setInterval(100L, soilMoistureSensor);
 timer.setInterval(100L, DHT11sensor);
 timer.setInterval(500L, checkPhysicalButton);
}
//Get the DHT11 sensor values
void DHT11sensor() {
 float h = dht.readHumidity();
 float t = dht.readTemperature();
 if (isnan(h) || isnan(t)) {
  Serial.println("Failed to read from DHT sensor!");
```

```
  return;
 }
 Blynk.virtualWrite(V0, t);
 Blynk.virtualWrite(V1, h);

 lcd.setCursor(0, 0);
 lcd.print("T:");
 lcd.print(t);
lcd.setCursor(8, 0);
 lcd.print("H:");
 lcd.print(h);

}
//Get the soil moisture values
void soilMoistureSensor() {
 int value = analogRead(soil);
 value = map(value, 0, 1024, 0, 100);
 value = (value - 100) * -1;
 Blynk.virtualWrite(V3, value);
 lcd.setCursor(0, 1);
 lcd.print("S:");
 lcd.print(value);
 lcd.print(" ");

}
//Get the PIR sensor values
void PIRsensor() {
 bool value = digitalRead(PIR);
 if (value) {
                          Blynk.logEvent("Motion
Detection","WARNNG!  Motion  Detected!");
//Enter your Event Name
  WidgetLED LED(V5);
  LED.on();
 } else {
  WidgetLED LED(V5);
  LED.off();
 }
 }

BLYNK_WRITE(V6)
{
 PIR_ToggleValue = param.asInt();
}


BLYNK_CONNECTED() {
 // Request the latest state from the server
 Blynk.syncVirtual(VPIN_BUTTON_1);
}

BLYNK_WRITE(VPIN_BUTTON_1) {
 relay1State = param.asInt();
 digitalWrite(RELAY_PIN_1, relay1State);
}
void loop() {
  if (PIR_ToggleValue == 1)
  {
  lcd.setCursor(5, 1);
  lcd.print("M:ON ");
    PIRsensor();
   }
   else
   {
  lcd.setCursor(5, 1);
  lcd.print("M:OFF");
  WidgetLED LED(V5);
  LED.off();
   }

if (relay1State == HIGH)
{
```

```
lcd.setCursor(11, 1);
lcd.print("W:ON ");
}
else if (relay1State == LOW)
{
  lcd.setCursor(11, 1);
  lcd.print("W:OFF");
  }
Blynk.run();//Run the Blynk library
timer.run();//Run the Blynk timer

}
```

IX.RESULTS

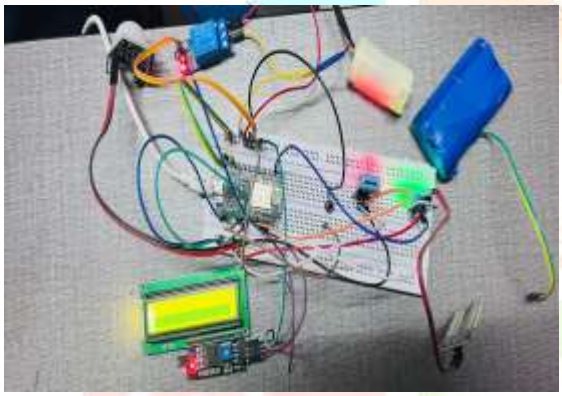


Fig-9.1 Working Model

Fig-9.2 display

Fig-9.3 App results

X.CONCLUSION

The personalized plant monitoring solution using the ESP32 microcontroller offers a tailored approach to smart agriculture and plant care. By integrating sensors, wireless communication, and data analytics, it enables real-time monitoring and informed decision-making for optimized plant growth and environmental management. This innovative solution showcases the potential of IoT technologies in enhancing sustainability, productivity, and resilience in plant ecosystems, highlighting the convergence of technology, agriculture, and environmental stewardship in a digitalized world.

XI.REFERENCES

[1] www.keil.com

8GSM4beginers,December.2000.

[2] http://en.Wikipedia.org/wiki/GSM

[3] Control and Communication Challenges in Networked Real-Time Systems By John Baillieul,Fellow IEEE , and Panos J. Antsaklis , Fellow IEEE .

[4] Panicle Rice Mite Program Manual by U.S. Department of Agriculture.