Smart Plant Monitoring System Using ESP32

Abstract :-

1	This Project proposes a smart plant monitoring system integrated with an ESP32 microcontroller and a suite of sensors including soil moisture, ultrasonic distance, DHT11 for temperature and humidity, and an LCD display. The system enables real-time monitoring of crucial environmental parameters essential for plant growth and health. Leveraging Bluetooth connectivity, users can remotely access and control the system via a mobile application, allowing for seamless interaction and management.
2	Commands are transmitted wirelessly through serial Bluetooth communication, enabling users to retrieve sensor readings and adjust system settings conveniently. The integration of multiple sensors ensures comprehensive monitoring capabilities, facilitating informed decision-making for optimal plant care. The system's embedded design, coupled with Bluetooth control, offers a versatile and user-friendly solution for smart plant management in both indoor and outdoor environments.

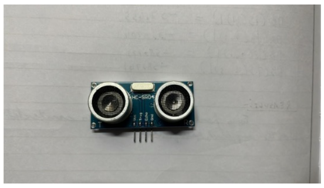
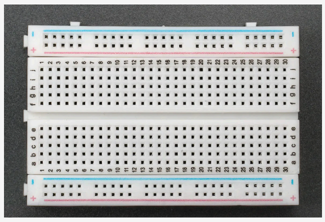
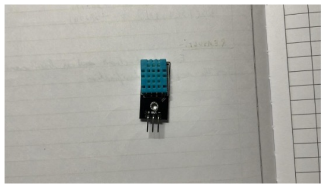
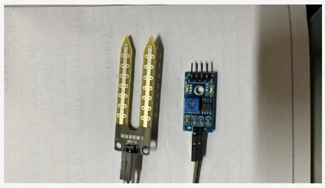
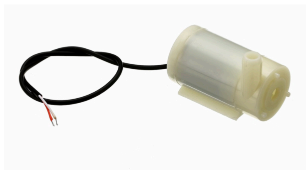
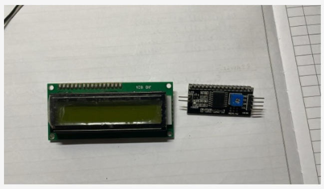
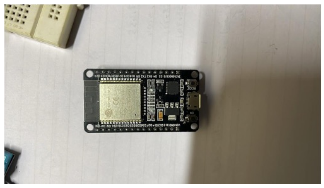
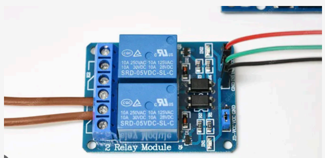
	Key Functionalities:
1	Real-time Monitoring: The system continuously monitors soil moisture levels, temperature, humidity, and ambient light conditions to provide accurate and up-to-date information about the plant's environment.
2	Bluetooth Control: Users can remotely access the system via a mobile application over Bluetooth connection, allowing them to view sensor readings and adjust settings from anywhere.
3	Sensor Fusion: By integrating multiple sensors, the system provides a holistic view of the plant's environment, enabling users to identify potential issues and take proactive measures.
4	Data Logging: The system logs sensor data over time, enabling users to track environmental trends and make informed decisions about plant care strategies.
5	Customizable Alerts: Users can set customizable thresholds for sensor readings, triggering alerts when conditions deviate from optimal levels, ensuring timely intervention.
6	Energy Efficiency: The system is designed to minimize power consumption, prolonging battery life for extended monitoring periods in remote or off-grid locations.

	Use Cases :-
1	Home Gardening: Individuals can use the system to monitor the health of plants in their gardens or indoor spaces, ensuring optimal growing conditions and maximizing yield.
2	Agricultural Monitoring: Farmers can deploy the system in agricultural fields to monitor soil moisture levels, temperature, and humidity, helping them optimize irrigation and fertilizer usage while minimizing water waste.
3	Research and Education: Educational institutions and research facilities can use the system to study plant physiology and environmental factors affecting plant growth, providing valuable insights for academic research and student projects.
4	Greenhouse Management: Commercial greenhouse operators can leverage the system to monitor and control environmental conditions, optimizing crop growth and maximizing profitability.
5	Urban Agriculture: Urban gardeners and community organizations can use the system to monitor plant health in urban environments, promoting sustainable food production and green spaces in cities.

Introduction :-

1	In the domain of modern agriculture, the evolution of embedded systems has brought about significant advancements, reshaping conventional farming techniques and fostering more sustainable and efficient approaches to plant cultivation. At the forefront of this transformation lies the Smart Embedded Plant Monitoring System. This system embodies the fusion of embedded systems technology with plant monitoring functionalities, presenting unparalleled insights and control over environmental conditions crucial for crop health.
2	Central to the Smart Embedded Plant Monitoring System's functionality is its integration of Bluetooth connectivity, serving as a conduit for seamless communication between the system and various display interfaces. This integration not only facilitates real-time data transmission to local display units but also enables remote access to critical information via mobile devices like smartphones and tablets. This dual capability empowers farmers and growers with the flexibility to monitor and manage their crops from any location, at any time, thereby enhancing productivity and optimizing yield.
3	Moreover, the Smart Embedded Plant Monitoring System features a dedicated LCD display, providing immediate feedback on essential parameters for plant growth and development. This on-site display ensures that vital information, including soil moisture levels, ambient temperature, and light intensity, remains readily accessible to users, even in environments where mobile connectivity may be limited or unavailable.
4	In this project, we delve into the design and functionality of the Smart Embedded Plant Monitoring System, exploring its potential to revolutionize plant care practices by offering a comprehensive, integrated solution for monitoring, analyzing, and optimizing crop conditions. Through the seamless integration of Bluetooth connectivity, LCD display, and mobile compatibility, this system promises to empower farmers and growers with the tools they need to achieve greater efficiency, sustainability, and success in modern agriculture.

Input Specifications (Components): -

1. Ultrasonic Distance Sensor	Rs 190	
2. Breadboard	Rs 60	
3. Temperature and Humidity Sensor (DHT11)	Rs 60	
4. Soil Moisture Sensor	Rs 35	
5. Water Pump	Rs 40	
6. LCD Display	Rs 120	
7. ESP-32 wroom	Rs 500	
8. Relay	Rs 60	

Working Principle: -

The Smart IoT Plant Monitoring Sensor, incorporating the listed components, is designed to provide real-time monitoring of various environmental parameters crucial for plant health. Here's how it works:

1. **Ultrasonic Distance Sensor:**
    - The ultrasonic distance sensor is used to measure the distance between the sensor and the surface of the soil in the plant pot. This distance can indicate the water level in the pot, as the soil absorbs moisture, affecting the distance measured by the sensor.
2. **Breadboard:**
    - The breadboard serves as a platform for connecting and interconnecting the various electronic components of the system. It provides a convenient and temporary way to prototype and test circuits before final assembly.
3. **Temperature and Humidity Sensor:**
    - The temperature and humidity sensor measures the ambient temperature and humidity in the plant's environment. This data is crucial for understanding the microclimate around the plant and ensuring optimal growing conditions.
4. **Soil Moisture Sensor:**
    - The water pump is integrated into the system to provide water to the plant when the soil moisture level drops below a certain threshold, typically set at 30%. When the soil moisture sensor detects that the moisture level has fallen below this threshold, it triggers the water pump to supply water to the plant, ensuring that it remains adequately hydrated.
5. **Water Pump:**
    - The flame sensor is included to detect any potential fire hazards in the vicinity of the plant. While not directly related to plant health, it adds an extra layer of safety to the monitoring system, especially in environments where fire risks may be present.
6. **LCD Display:**
    - The LCD serves as the user interface, providing real-time feedback on the sensor readings. It typically displays information such as soil moisture levels, temperature, humidity, and any alerts or notifications generated by the system.
7. **ESP-32:**
    - The ESP-32 is a microcontroller with built-in Wi-Fi and Bluetooth capabilities. It serves as the brain of the monitoring system, collecting data from the various sensors, processing it, and transmitting it wirelessly to a central hub or cloud-based platform for further analysis and visualization. Additionally, it can receive commands or configuration updates wirelessly, allowing for remote control and monitoring of the sensor system.

Overall, the Smart IoT Plant Monitoring Sensor continuously monitors key parameters such as soil moisture, temperature, humidity, and potential fire hazards, providing growers with actionable insights to optimize plant care and ensure healthy growth. The integration of the ESP-32 enables seamless connectivity and remote monitoring capabilities, enhancing the system's versatility and usability.

Code :-

```
// Include libraries for LCD display, DHT sensor, and Bluetooth communication
#include <LiquidCrystal_I2C.h>
#include "DHT.h"
#include "BluetoothSerial.h"

// Error message if Bluetooth is not enabled during compilation
#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to and enable it
#endif

// Bluetooth serial object and variables for received data
BluetoothSerial SerialBT;
int received;
char receivedChar;

// Define constants for characters used in Bluetooth commands
const char SM = '1';
const char GW = '2';
const char TH = '3';

// Define pin connected to the LED
const int LEDpin = 23;

// Define pin connected to the Pump Relay
const int relay = 26;

// Define pin connected to the DHT sensor
#define DHT11PIN 4

// Create a DHT object for interacting with the sensor
DHT dht(DHT11PIN, DHT11);

// Define pins used for the ultrasonic sensor
const int trigPin = 5;
const int echoPin = 18;

// Define constants for speed of sound and conversion factor (optional)
#define SOUND_SPEED 0.034
#define CM_TO_INCH 0.393701

// Variables used for distance measurement
long duration;
float distanceCm;

// Variables used for soil moisture reading
int _moisture, sensor_analog;
const int sensor_pin = A0;

// Define LCD dimensions
int lcdColumns = 16;
int lcdRows = 2;

// Create a LiquidCrystal object for controlling the LCD display
LiquidCrystal_I2C lcd(0x27, lcdColumns, lcdRows);
```

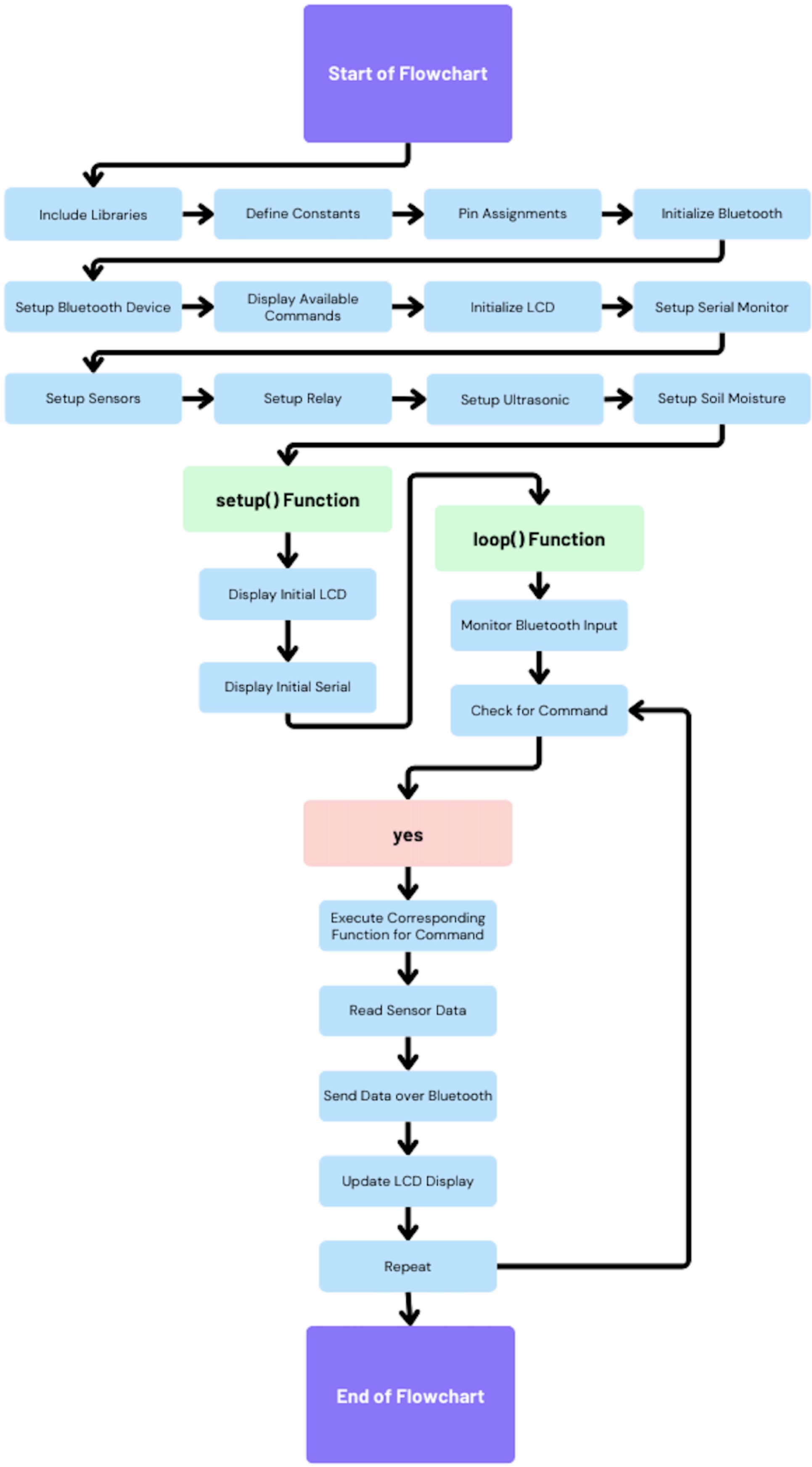
```
// Define static message and scrolling message for the LCD
String messageStatic = "IOT Project";
String messageToScroll = "SMART PLANT MONITER";

// Function to scroll text on the LCD display
void scrollText(int row, String message, int delayTime, int lcdColumns) {
  for (int i = 0; i < lcdColumns; i++) {
    message = " " + message;
  }
  message = message + " ";
  for (int pos = 0; pos < message.length(); pos++) {
    lcd.setCursor(0, row);
    lcd.print(message.substring(pos, pos + lcdColumns));
    delay(delayTime);
  }
}

// Function to read temperature and humidity from the DHT sensor
void read_DHT(){
  String humi = (String)dht.readHumidity();
  String temp = (String)dht.readTemperature();
  messageToScroll = "Temperature: " + temp + "'c ; Humidity: " + humi;
  SerialBT.println(messageToScroll);
  Serial.println(messageToScroll);
  scrollText(1, messageToScroll, 300, lcdColumns);
}

// Function to read soil moisture value
void read_SoilM(){
  sensor_analog = analogRead(sensor_pin);
  _moisture = ( 100 - ( (sensor_analog / 4095.00) * 100 ) );
  messageToScroll = "Moisture = " + (String)_moisture + "%";
  SerialBT.println(messageToScroll);
  Serial.println(messageToScroll);
  scrollText(1, messageToScroll, 300, lcdColumns);
}

//Functon to mesure soil moisture and control water Pump through erelay
void soilM_for_waterpump(){
  sensor_analog = analogRead(sensor_pin);
  _moisture = ( 100 - ( (sensor_analog / 4095.00) * 100 ) );
  if (_moisture<30){
    digitalWrite(relay, LOW);
  }
  else if(_moisture>50){
    digitalWrite(relay, HIGH);
  }
}
// Function to measure distance using the ultrasonic sensor
void read_US(){
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
```

```
  duration = pulseIn(echoPin, HIGH);
  distanceCm = 100-(duration * SOUND_SPEED/2);
  messageToScroll = "The hight of the plant " + (String)distanceCm;
  SerialBT.println(messageToScroll);
  Serial.println(messageToScroll);
  scrollText(1, messageToScroll, 300, lcdColumns);
}

// Function to initialize components (LCD, serial, Bluetooth, etc.)
void setup(){
  lcd.init();
  lcd.backlight();
  Serial.begin(115200);
  pinMode(relay, OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  dht.begin();
  SerialBT.begin("IOT_project");
  SerialBT.println("Use to following Numbers to control the device\n 1: soil moisture\n 2: hight of the
plant \n 3: humidity and Temperature");
  scrollText(1, messageToScroll, 250, lcdColumns);
}

// Function that runs repeatedly to monitor, display, and control
void loop(){
  soilM_for_waterpump();
  lcd.setCursor(0, 0);
  lcd.print(messageStatic);
  SerialBT.println("ready to recive command: ");
  receivedChar = (char)SerialBT.read();
  if (Serial.available()) {
    SerialBT.write(Serial.read());
  }
  if (SerialBT.available()) {
    SerialBT.print("Received:");
    SerialBT.println(receivedChar);
    Serial.print("Received:");
    Serial.println(receivedChar);

    if(receivedChar == SM) {
      read_SoilM();
    }
    if(receivedChar == GW) {
      read_US();
    }
    if(receivedChar == TH) {
      read_DHT();
    }
  else{
      SerialBT.print("the command ID is not found");
    }
    SerialBT.print("wait till the device is ready to recive");
  }
  soilM_for_waterpump();
  delay(10);
}
```

Algorithm :-

1. Include necessary libraries for LCD display, DHT sensor, Bluetooth communication, and define constants and pin assignments for components.
2. Initialize Bluetooth serial communication and setup Bluetooth device name. Display a message indicating the available commands.
3. Initialize components such as LCD, DHT sensor, and set pin modes for the relay, ultrasonic sensor, and soil moisture sensor.
4. Implement functions to read data from sensors:
   - read_DHT(): Reads temperature and humidity from the DHT sensor, sends data over Bluetooth, and scrolls the message on the LCD.
   - read_SoilM(): Reads soil moisture level, sends data over Bluetooth, and scrolls the message on the LCD.
   - soilM_for_waterpump(): Reads soil moisture level and controls the water pump relay based on moisture threshold.
   - read_US(): Measures the height of the plant using the ultrasonic sensor, sends data over Bluetooth, and scrolls the message on the LCD.
5. Setup function (setup()):
   - Initialize LCD, serial communication, and Bluetooth.
   - Display initial messages on the LCD and serial monitor.
6. Main loop function (loop()):
   - Continuously monitor Bluetooth serial for incoming commands.
   - If a command is received, execute corresponding functions to read sensor data or perform actions.
   - Send data over Bluetooth and display messages on the LCD accordingly.
   - Repeat the process continuously.

Flow Chart:-



Start of Flowchart
Include Libraries
Define Constants
Pin Assignments
Initialize Bluetooth
Setup Bluetooth Device
Display Available Commands
Initialize LCD
Setup Serial Monitor
Setup Sensors
Setup Relay
Setup Ultrasonic
Setup Soil Moisture
setup() Function
Display Initial LCD
Display Initial Serial
loop() Function
Monitor Bluetooth Input
Check for Command
yes
Execute Corresponding Function for Command
Read Sensor Data
Send Data over Bluetooth
Update LCD Display
Repeat
End of Flowchart

Result: -

Equipped with the Smart Embedded Plant Monitoring System, growers gain precise control over crucial factors influencing plant well-being. Real-time insights into soil moisture, temperature, humidity, and potential hazards empower growers to take timely actions, optimizing growing conditions. By upholding optimal environmental parameters, growers foster robust plant growth, resulting in enhanced yields and superior crop quality. Furthermore, the ESP-32's remote monitoring capabilities afford growers the convenience of overseeing their plants from any location, thereby streamlining plant management processes and boosting efficiency.

References: -

- R. Zhang, Z. Ren, J. Sun, W. Tang, D. Ning, Y. Qian "Method for monitoring the cotton plant vigor based on the WSN technology" Comput Electron Agric, 133 (2017), pg. 68-79.
- P.C. Robert "Precision agriculture: a challenge for crop nutrition management" Progress in Plant Nutrition: Plenary Lectures of the XIV International Plant Nutrition Colloquium, Springer, Dordrecht (2002), pg. 143-149.
- J. Mateo-Sagasta, S.M. Zadeh, H. Turral, J. Burke "Water Pollution from Agriculture: A Global Review" Executive summary (2017)
- A. Muller, C. Schader, N. El-Hage Scialabba, et al. "Strategies for feeding the world more sustainably with organic agriculture" Nat Commun, 8 (1) (2017), pg. 1-13
- H. Auernhammer "Precision farming—the environmental challenge" Comput Electron Agric, 30 (1-3) (2001), pg. 31-43.
- V.I. Adamchuk, J.W. Hummel, M.T. Morgan, S.K. Upadhyaya "On-the-go soil sensors for precision agriculture" Comput Electron Agric, 44 (1) (2004), pg. 71-91.
- H. Chhipa "Nano fertilizers and nanopesticides for agriculture" Environ Chem Lett, 15 (1) (2017), pg. 15-22.
- E. Raj, M. Appadurai, K. Athiappan "Precision farming in modern agriculture" Smart Agriculture Automation Using Advanced Technologies, Springer, Singapore (2021), pg. 61-87.
- J. Billingsley "Global positioning systems (GPS) for agriculture: an overview" Robot Autom Improv Agric (2019), pg. 101-106.
- R.P. Sishodia, R.L. Ray, S.K. Singh "Applications of remote sensing in precision agriculture: a review" Rem Sens, 12 (19) (2020), pg. 31-36.