Activity 6**: Automated Plant Care System**

This activity is how to build a self-sustaining plant monitoring and watering system. It combines a moisture sensor, temperature/humidity sensor, and a relay module to automate plant care. The system will display real-time data on an LCD and can be controlled via a smartphone using Bluetooth.

**1. Learning Outcomes**

* **Integrated Sensing:** Learn to read data from multiple sensors (DHT11, Soil Moisture) simultaneously.
* **Actuator Control:** Control an actuator (relay and pump) based on sensor data.
* **Display & HMI:** Display real-time sensor readings on an LCD using the I2C protocol.
* **Wireless Control:** Use the HC-05 Bluetooth module to send commands and receive data from the system via a smartphone app.

**2. Materials**

* **ESP32 Development Board**
* **DHT11 Sensor Module**
* **Soil Moisture Sensor Module**
* **Single Relay Module**
* **16x2 LCD with I2C Backpack**
* **HC-05 Bluetooth Module**
* **Breadboard**
* **Jumper Wires**
* **5V Water Pump & Power Supply (not listed, but required for a functional system)**
* **Small Bucket/Container (for water)**

**3. Circuit Connections**

This project requires careful wiring to connect multiple modules to the ESP32.

* **16x2 LCD (I2C):**
  + **VCC** to **ESP32 3.3V** or **5V**
  + **GND** to **ESP32 GND**
  + **SDA** to **ESP32 GPIO21**
  + **SCL** to **ESP32 GPIO22**
* **DHT11 Sensor:**
  + **VCC** to **ESP32 3.3V**
  + **GND** to **ESP32 GND**
  + **DATA** to **ESP32 GPIO4**
* **Soil Moisture Sensor:**
  + **VCC** to **ESP32 3.3V**
  + **GND** to **ESP32 GND**
  + **AO (Analog Output)** to **ESP32 GPIO34**
* **Single Relay Module:**
  + **VCC** to **ESP32 5V** (or external 5V power supply)
  + **GND** to **ESP32 GND**
  + **IN** to **ESP32 GPIO2**
* **HC-05 Bluetooth Module:**
  + **VCC** to **ESP32 5V**
  + **GND** to **ESP32 GND**
  + **TXD** to **ESP32 GPIO16** (RX2)
  + **RXD** to **ESP32 GPIO17** (TX2)

**4. Setting Up the Arduino IDE**

1. **Install ESP32 Board Manager:** Make sure the ESP32 package is installed in your Arduino IDE.1
2. **Install Libraries:**
   * **LiquidCrystal\_I2C:** Go to Sketch > Include Library > Manage Libraries and search for "LiquidCrystal\_I2C." Install the one by **Frank de Brabander**.
   * **DHT sensor library:** Search for "DHT sensor library" and install it by **Adafruit**.
   * **Adafruit Unified Sensor:** This is a dependency for the DHT library. Install it as well.

**5. Arduino Code**

This code reads from the sensors, controls the relay, and communicates via Bluetooth. The system will automatically water the plant if the soil moisture falls below a certain threshold.

C++

#**include** <LiquidCrystal\_I2C.h>

#**include** <DHT.h>

#**include** "HardwareSerial.h"

// Set the LCD address and dimensions

LiquidCrystal\_I2C lcd(0x27, 16, 2);

// DHT11 sensor pin and type

#**define** DHT\_PIN 4

#**define** DHT\_TYPE DHT11

DHT dht(DHT\_PIN, DHT\_TYPE);

// Soil moisture sensor pin

#**define** MOISTURE\_PIN 34

// Relay module pin

#**define** RELAY\_PIN 2

// HC-05 Bluetooth serial communication

HardwareSerial SerialBT(2);

// Define a threshold for watering (adjust as needed, 0 = dry, 4095 = wet)

#**define** MOISTURE\_THRESHOLD 1500

void setup() {

Serial.begin(115200);

SerialBT.begin(9600); // Bluetooth default baud rate

// Initialize LCD

lcd.init();

lcd.backlight();

// Initialize DHT sensor

dht.begin();

// Set relay pin as output and start off

pinMode(RELAY\_PIN, OUTPUT);

digitalWrite(RELAY\_PIN, HIGH); // Assuming a LOW signal activates the relay

lcd.setCursor(0, 0);

lcd.print("Smart Garden");

lcd.setCursor(0, 1);

lcd.print("System Ready");

delay(2000);

}

void loop() {

// Read sensor data

float humidity = dht.readHumidity();

float temperature = dht.readTemperature();

int moisture = analogRead(MOISTURE\_PIN);

// Check if sensor readings are valid

if (isnan(humidity) || isnan(temperature)) {

Serial.println("Failed to read from DHT sensor!");

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Sensor Error");

return;

}

// Display data on LCD

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("T:");

lcd.print(temperature);

lcd.print("C H:");

lcd.print(humidity);

lcd.print("%");

lcd.setCursor(0, 1);

lcd.print("Moisture:");

lcd.print(moisture);

// Check if soil is dry and water the plant

if (moisture < MOISTURE\_THRESHOLD) {

Serial.println("Soil is dry! Watering...");

SerialBT.println("Soil is dry! Watering...");

lcd.setCursor(0, 1);

lcd.print("Watering...");

digitalWrite(RELAY\_PIN, LOW); // Turn on the pump

delay(5000); // Run pump for 5 seconds

digitalWrite(RELAY\_PIN, HIGH); // Turn off the pump

delay(2000); // Wait for moisture to be absorbed

}

// Handle Bluetooth commands

if (SerialBT.available()) {

char command = SerialBT.read();

if (command == 'W') {

Serial.println("Manual water command received!");

SerialBT.println("Manual watering initiated.");

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Manual Water");

digitalWrite(RELAY\_PIN, LOW);

delay(5000);

digitalWrite(RELAY\_PIN, HIGH);

}

}

// Send sensor data to Bluetooth

String dataString = "T:" + String(temperature) + ",H:" + String(humidity) + ",M:" + String(moisture);

SerialBT.println(dataString);

delay(5000); // Wait before the next reading cycle

}

**6. Running the Project**

1. **Modify and Upload Code:** Change the I2C address for your LCD if needed (common addresses are 0x27 or 0x3F). Upload the code to your ESP32.
2. **Connect Power:** Power your ESP32.
3. **Bluetooth Connection:** On your smartphone, enable Bluetooth and pair with the HC-05 module (the default pairing code is 1234 or 0000). Use a Bluetooth Terminal app to send the character 'W' to manually water the plant or to view the incoming data string.
4. **Test the System:**
   * Submerge the soil moisture sensor in water and see the moisture value increase.
   * Let the sensor dry out, and watch as the relay clicks and a message appears on the LCD.
   * Use your phone's Bluetooth terminal to send 'W' and manually activate the relay.