HARDWARE

Components used :

* **ESP32 Microcontroller**  
  The ESP32 is the central controller of the system, responsible for collecting data from various sensors, processing it, and sending it to a Flask server via HTTP requests. It also controls the water pump and other peripherals.
* **Moisture Sensor**  
  The moisture sensor detects the moisture content of the soil. When the soil moisture drops below a certain threshold, it signals the ESP32 to activate the water pump and irrigate the plants.
* **Water Pump**  
  The water pump is used to irrigate the plants when triggered by the ESP32. The pump is controlled using a relay to ensure it is powered on or off based on the moisture levels in the soil.
* **5V Relay**  
  The relay acts as an intermediary switch that allows the ESP32 to control high-power devices like the water pump. When the moisture level is low, the ESP32 sends a signal to the relay to turn the water pump on, and when the moisture level is sufficient, it turns the pump off.
* **GSM Module**  
  The GSM module is used to send SMS alerts. It is particularly useful for notifying the user if there's an issue with the system, such as low moisture levels, high temperature, or when a sensor malfunctions.
* **Rain Sensor**  
  The rain sensor detects rain and sends a signal to the ESP32 when rain is detected. If the rain sensor is activated, it prevents the water pump from activating, avoiding unnecessary irrigation when it’s raining.
* **DHT11 Sensor**  
  The DHT11 sensor measures the temperature and humidity levels of the environment. This data can be used to monitor the weather conditions, helping to optimize irrigation timing.
* **pH Sensor**  
  The pH sensor measures the acidity or alkalinity of the soil, which is important for plant health. The ESP32 uses the pH data to alert the user when the soil pH is outside the optimal range for plant growth.
* **Breadboard and Jumper Wires**  
  The breadboard is used to prototype the circuit connections between the components, and jumper wires are used for connecting the sensors to the ESP32.

**Working:**

This system is designed to monitor various environmental parameters such as soil moisture, temperature, humidity, pH, and rainfall, while also controlling the irrigation system (water pump) based on the sensor data. The system communicates with a Flask-based server over HTTP, sending real-time sensor data for monitoring and control.

**Sensor Connections:**

The sensors are connected to the ESP32 via GPIO pins. The connections are made using a breadboard for ease of prototyping and jumper wires to make the necessary connections. Each sensor is connected to a specific GPIO pin on the ESP32, and the data from these sensors is read periodically.

* **Rain Sensor**: Connected to a GPIO pin to monitor rainfall. If it detects rain, the system prevents irrigation.
* **DHT11**: Connected to a GPIO pin to monitor the temperature and humidity levels.
* **pH Sensor**: Connected to an analog pin on the ESP32 to measure the soil pH levels.
* **Moisture Sensor**: Connected to an analog pin to measure the soil moisture levels.
* **Relay**: The relay is connected to a GPIO pin and is used to control the water pump. It acts as a switch for turning the pump on or off based on the moisture level.

**Data Collection and Transmission:**

1. **Sensor Data Acquisition**:
   * The ESP32 reads data from the connected sensors, including the moisture level, temperature, humidity, pH level, and rain detection status.
2. **HTTP Request to Flask Server**:
   * The sensor data is sent to a Flask server through HTTP requests. The ESP32 makes periodic requests to the server, passing the sensor readings as parameters.
3. **Flask Server**:
   * The Flask server receives the sensor data and processes it. It can display the data on a web page, provide real-time updates, or take actions based on predefined thresholds (e.g., sending an alert if the moisture level is low).

**Control of Water Pump (Irrigation):**

1. The system continuously monitors the moisture level from the moisture sensor.
2. If the moisture level is below a specified threshold, the ESP32 sends a signal to the relay to activate the water pump.
3. The water pump irrigates the plants by dispensing water from the reservoir.
4. If the moisture level is sufficient, the pump is turned off by sending a signal to the relay.
5. The rain sensor ensures that irrigation does not occur during rainfall, thus saving water and preventing overwatering.

**SMS Alerts via GSM Module:**

1. The GSM module is connected to the ESP32 to send SMS alerts.
2. If the moisture level falls below the threshold, or if there is a problem with one of the sensors, the ESP32 will send an SMS message to the user with a status update or alert.
3. The GSM module ensures that the user is notified even when they are not actively monitoring the system.

**Conclusion:**

This project integrates multiple environmental sensors with an ESP32 microcontroller to create a smart irrigation system. It continuously monitors environmental conditions like soil moisture, temperature, humidity, pH, and rainfall, and automatically controls the irrigation process using a water pump and relay. It also sends SMS alerts to the user and provides remote monitoring through a Flask web interface. This system is ideal for smart gardening and precision agriculture applications, ensuring that plants receive optimal care with minimal human intervention.