**Project Manual: Intelligent Smart Precision Agriculture with IoT and Machine Learning**

**1. Introduction**

This document provides a comprehensive guide to building a Smart Precision Agriculture System using the ESP32 microcontroller, multiple environmental sensors, a cloud infrastructure based on Firebase and ThingSpeak, and a Machine Learning model deployed through a Flask API. The system automates irrigation in a hydroponic/soil-based farming setup by monitoring temperature, humidity, soil moisture, pH, and light intensity and making intelligent decisions.

**2. Components Required**

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Component** | **Quantity** |
| 1 | ESP32 Dev Board | 1 |
| 2 | DHT11 Sensor (Temp & Humidity) | 1 |
| 3 | Soil Moisture Sensor | 1 |
| 4 | LDR (Light Sensor) | 1 |
| 5 | pH Sensor Module | 1 |
| 6 | 5V Relay Module | 1 |
| 7 | Submersible Pump | 1 |
| 8 | DC-DC Buck Converter | 1 |
| 9 | 16x2 LCD Display with I2C Interface | 1 |
| 10 | Breadboard & Jumper Wires | As needed |
| 11 | 5V SMPS or Battery Pack | 1 |

**3. Hardware Connections**

|  |  |
| --- | --- |
| **Sensor/Module** | **ESP32 GPIO Pin** |
| DHT11 | GPIO16 |
| Soil Moisture | GPIO32 |
| LDR | GPIO33 |
| pH Sensor | GPIO34 |
| Relay Module | GPIO14 |
| I2C SDA (LCD) | GPIO21 |
| I2C SCL (LCD) | GPIO22 |

Ensure proper voltage levels and grounding. Use opto-isolated relay for pump control.

**4. Software Setup**

**4.1 Arduino IDE Configuration:**

* Install ESP32 board URL from Preferences.
* Add libraries: FirebaseESP32, ThingSpeak, DHT, LiquidCrystal\_I2C, HTTPClient, ArduinoJson.

**4.2 Firebase Setup:**

* Create a Firebase project.
* Enable Realtime Database in test mode.
* Obtain database URL and Legacy Token.

**4.3 ThingSpeak Setup:**

* Create a channel with 5 fields.
* Obtain Channel ID and Write API Key.

**5. Machine Learning Model Deployment**

**5.1 Data Preparation:**

* Collect data from sensors and store in CSV.
* Preprocess data and normalize values.

**5.2 Model Training:**

* Use LSTM or Decision Tree classifier.
* Target: Binary classification (water/no water).

**5.3 Deployment:**

* Save model as .h5 or .tflite.
* Use Flask with pyngrok to expose via HTTPS.
* Implement /predict endpoint accepting JSON.

**6. Integration Flow**

1. ESP32 reads all sensor values.
2. Sends data to Firebase and ThingSpeak.
3. Makes a POST request to Flask ML API.
4. Receives a prediction.
5. Activates pump for 5 seconds if prediction is "water".
6. Also triggers pump every 15 minutes as a fallback.
7. LCD shows live sensor readings.

**7. Final Features**

* Dual-pump activation logic: time-based and AI-based.
* Real-time data visualization.
* Cloud data storage for performance evaluation.
* Mobile-accessible dashboards.
* Cost-effective and energy-efficient setup.

**8. Deployment Tips**

* Use external 5V power for pump.
* Calibrate pH and soil sensors before use.
* Shield electronics from water exposure.
* Regularly update ML model with new data.

**9. Troubleshooting**

|  |  |
| --- | --- |
| **Issue** | **Solution** |
| WiFi Not Connecting | Check SSID/password, reboot ESP32 |
| Firebase/ThingSpeak Failures | Recheck API keys and database URL |
| LCD Not Displaying | Verify I2C address (0x27 or 0x3F) |
| Pump Not Turning Off | Confirm GPIO logic level and pump relay type |

**10. Conclusion**

This project serves as a robust, scalable prototype for intelligent agricultural automation. By combining IoT and ML, it demonstrates the transformative potential of modern technologies in addressing food security and sustainability challenges. This manual provides all necessary instructions to replicate the system for academic or personal innovation purposes.