

Smart crop Advisor

Objective:

To develop a smart IoT system that monitors soil conditions and suggests suitable crops to enhance agricultural productivity.

SmartCrop Advisor – Project Overview

SmartCrop Advisor is an IoT-based smart agriculture system designed to assist farmers in selecting suitable crops by continuously monitoring key soil and environmental parameters. The system integrates multiple sensors to measure soil nutrients (Nitrogen, Phosphorous, Potassium), soil pH, soil moisture, temperature, and humidity. These parameters play a crucial role in determining soil fertility and crop suitability.

The hardware setup consists of an Arduino Uno microcontroller connected to an NPK sensor, soil moisture sensor, pH sensor, and a DHT11 sensor. The sensors collect real-time data from the soil and surrounding environment, which is processed by the Arduino. Based on predefined threshold values and rule-based logic, the system analyzes the collected data and suggests an appropriate crop for cultivation.

SmartCrop Advisor helps reduce guesswork in farming by providing data-driven crop recommendations, improving crop yield and efficient resource utilization. The project demonstrates the practical application of IoT technology in agriculture and highlights how sensor-based monitoring can support smart farming practices.

```
#include <SoftwareSerial.h>
#include <Wire.h>
#include "DHT.h"

// ----- RS485 NPK Sensor -----
#define RE 8
#define DE 7

SoftwareSerial mod(2, 3); // RS485
```

```
// ----- DHT11 Sensor -----  
  
#define DHTPIN 4  
  
#define DHTTYPE DHT11  
  
DHT dht(DHTPIN, DHTTYPE);  
  
  
// ----- PH Sensor -----  
  
#define PH_PIN A0  
  
float calibration = 0.0;  
  
float phValue = 0;  
  
  
// ----- Soil Moisture Sensor -----  
  
#define SOIL_PIN A1  
  
int soilMoisture = 0;  
  
String moistureStatus;  
  
  
// ----- Crop Suggestion -----  
  
String cropSuggestion;  
  
  
void setup() {  
    Serial.begin(9600);  
  
    mod.begin(9600);  
  
  
    pinMode(RE, OUTPUT);  
    pinMode(DE, OUTPUT);
```

```
dht.begin();  
randomSeed(analogRead(0));  
  
delay(500);  
Serial.println("System Initialized...");  
}  
  
void loop() {  
  
// ----- NPK Readings (Simulated) -----  
byte nitrogenVal = nitrogen();  
byte phosphorousVal = phosphorous();  
byte potassiumVal = potassium();  
  
// ----- DHT11 Readings -----  
float humidity = dht.readHumidity();  
float temperature = dht.readTemperature();  
  
// ----- pH Sensor Reading -----  
int phSensorValue = analogRead(PH_PIN);  
float voltage = phSensorValue * (5.0 / 1023.0);  
phValue = 7 + ((2.5 - voltage) / 0.18);  
phValue = phValue + calibration;  
  
// ----- Soil Moisture Reading -----
```

```
soilMoisture = analogRead(SOIL_PIN);

if (soilMoisture > 700)
    moistureStatus = "Dry";
else if (soilMoisture > 400)
    moistureStatus = "Moderate";
else
    moistureStatus = "Wet";

// ----- Crop Suggestion Logic -----
if (moistureStatus == "Wet" && temperature > 20 && pHValue >= 5.5 && pHValue <= 7.5) {
    cropSuggestion = "Rice";
}

else if (moistureStatus == "Moderate" && temperature >= 15 && temperature <= 25) {
    cropSuggestion = "Wheat";
}

else if (moistureStatus == "Dry" && temperature > 25) {
    cropSuggestion = "Cotton";
}

else if (pHValue >= 6.0 && pHValue <= 7.0) {
    cropSuggestion = "Vegetables";
}

else {
    cropSuggestion = "Soil not suitable for major crops";
}
```

```
// ----- Display Output -----  
  
Serial.println("----- Sensor Data -----");  
  
Serial.print("Nitrogen: "); Serial.print(nitrogenVal); Serial.println(" mg/kg");  
  
Serial.print("Phosphorous: "); Serial.print(phosphorousVal); Serial.println(" mg/kg");  
  
Serial.print("Potassium: "); Serial.print(potassiumVal); Serial.println(" mg/kg");  
  
  
Serial.print("Humidity: "); Serial.print(humidity); Serial.println(" %");  
Serial.print("Temperature: "); Serial.print(temperature); Serial.println(" °C");  
  
  
Serial.print("Soil Moisture: "); Serial.print(soilMoisture);  
Serial.print(" ("); Serial.print(moistureStatus); Serial.println(")");  
  
  
Serial.print("pH Value: "); Serial.println(phValue, 2);  
  
  
Serial.print("Suggested Crop: ");  
Serial.println(cropSuggestion);  
  
  
Serial.println("-----");  
delay(3000);  
}  
  
  
// ----- NPK Functions (Simulation) -----  
  
byte nitrogen() {  
    return random(10, 101);  
}
```

```
}
```

```
byte phosphorous() {
```

```
    return random(5, 101);
```

```
}
```

```
byte potassium() {
```

```
    return random(20, 201);
```

```
}
```