CODE:

WOKWI SAVE SHARE

sketch.ino ● | diagram.json | npksensor.chip.json | npksensor.chip.c | moisturesensor.chip.json | moisturesensor.chip.c | libraries.txt | Library Manager

```
#define BLYNK_AUTH_TOKEN "rctDfVwCBZ7PERy8wMtJPRZivww5Zx7W"
#define BLYNK_TEMPLATE_ID "TMPL3tEgZbd6z"
#define BLYNK_TEMPLATE_NAME "CROP AND PEST MONITORING IOT"

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include "DHTesp.h"

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels
#define OLED_RESET -1    // Reset pin # (or -1 if sharing Arduino reset pin)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

char ssid[] = "Wokwi-GUEST";
char pass[] = "";

const int DHT_PIN = 27;
DHTesp dhtSensor;

const int pHSensorPin = 32;
// Calibration parameters (you need to calibrate your sensor)
const float acidVoltage = 2032.44;  // voltage at pH 4.0
const float neutralVoltage = 1500.0; // voltage at pH 7.0

// Potentiometer is connected to GPIO 34 (Analog ADC1_CH6)
const int potPin = 34;

// variable for storing the potentiometer value
int potValue = 0;
```

WOKWI SAVE SHARE

sketch.ino ● | diagram.json | npksensor.chip.json | npksensor.chip.c | moisturesensor.chip.json | moisturesensor.chip.c | libraries.txt | Library Manager

```

#define ncom 3 //  number of commands.
char commar[ncom] = {0x1, 0x3, 0x5}; // Actual commands
// Response Strings can be stored like this
char respar[ncom][30] = {"Phosphorous value is: ", "Potassium value is: ", "Nitrogen value is: "};
uint8_t rtValue[ncom]; // Store the return values from the custom chip in here. you can use the same
//values to forward to the IOT part.

BlynkTimer timer;

void displayTitle();
void displayTeamMembers();
void displayAllSensorReadings(unsigned long duration);
void displayThankYou();
void sendData();

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  Serial2.begin(15200, SERIAL_8N1, 16, 17); //initialize the custom chip communication line.

  // Initialize Blynk
  Serial.println("Connecting to Blynk...");
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  while (!Blynk.connected()) {
    Serial.print(".");
    delay(500);
  }
  Serial.println("");
  Serial.println("Blynk connected.");

  dhtSensor.setup(DHT_PIN, DHTesp::DHT22);
```

WOKWI SAVE SHARE

sketch.ino ● | diagram.json | npksensor.chip.json | npksensor.chip.c | moisturesensor.chip.json | moisturesensor.chip.c | libraries.txt | Library Manager

```
void setup() {
  timer.setInterval(2000L, sendData); // Send data every 2 seconds

  Serial.println("Hello, ESP32!");

  display.begin(SSD1306_SWITCHCAPVCC, 0x3C); // Initialize with the I2C address 0x3C
  display.clearDisplay();                    // Clear the buffer
  display.display();                         // Display

  // Display title for 5 seconds
  displayTitle();
  delay(5000);

  // Display team members for 5 seconds
  displayTeamMembers();
  delay(5000);

  // Display all sensor readings for 10 seconds
  displayAllSensorReadings(10000);

  // Display "Thank you" message
  displayThankYou();
}

void displayTitle() {
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 10);
  display.println("CROP NUTRITION MONITORING IOT");
  display.println("System Using IoT");
  display.display();
```

WOKWI SAVE SHARE

sketch.ino ● diagram.json npksensor.chip.json npksensor.chip.c moisturesensor.chip.json moisturesensor.chip.c libraries.txt Library Manager

```
88  void displayTitle() {
96  }
97
98  void displayTeamMembers() {
99    display.clearDisplay();
100   display.setTextSize(1);
101   display.setTextColor(SSD1306_WHITE);
102   display.setCursor(0, 10);
103   display.println("Team Members:");
104   display.println("Lingeswaran B");
105   display.println("Krishna M");
106   display.println("Mukesh D");
107   display.println("Udhaya Kumar S");
108   display.println("Venkatesh S");
109   display.display();
110 }
111
112 void displayAllSensorReadings(unsigned long duration) {
113   float h = dhtSensor.getHumidity();
114   float t = dhtSensor.getTemperature();
115   int potValue = analogRead(potPin);
116   display.clearDisplay();
117   display.setTextSize(1);
118   display.setTextColor(SSD1306_WHITE);
119   display.setCursor(0, 10);
120   display.println("All Sensor Readings:");
121   display.print("Temperature: ");
122   display.println(t, 2);
123   display.print("Humidity: ");
124   display.println(h, 2);
125   display.print("Soil Moisture: ");
126   display.println(potValue);
```

WOKWI SAVE SHARE

sketch.ino ● diagram.json npksensor.chip.json npksensor.chip.c moisturesensor.chip.json moisturesensor.chip.c libraries.txt Library Manager

```
112 void displayAllSensorReadings(unsigned long duration) {
126   display.println(potValue);
127   display.display();
128   delay(duration);
129 }
130
131 void displayThankYou() {
132   display.clearDisplay();
133   display.setTextSize(1);
134   display.setTextColor(SSD1306_WHITE);
135   display.setCursor(0, 20);
136   display.println("Thank you!");
137   display.display();
138 }
139
140 void sendData() {
141   float t = dhtSensor.getTemperature();
142   float h = dhtSensor.getHumidity();
143
144   if (isnan(h) || isnan(t)) {
145     Serial.println("Failed to read from DHT sensor!");
146     return;
147   }
148
149   Serial.print("Humidity: ");
150   Serial.print(h);
151   Serial.print("%\t");
152   Serial.print("Temperature: ");
153   Serial.print(t);
154   Serial.println(" °C");
155
156   int analogValue = analogRead(pHSensorPin);
```

WOKWI SAVE SHARE

sketch.ino ● diagram.json npksensor.chip.json npksensor.chip.c moisturesensor.chip.json moisturesensor.chip.c libraries.txt Library Manager

```
140 void sendData() {
157
158   // Convert the analog value to voltage
159   float voltage = analogValue * (3.3 / 4095.0); // 3.3V reference, 12-bit ADC
160
161   // Convert the voltage to pH value and moisture
162   float pHValue = (voltage * 14.0) / 3.3; // Declare pHValue here
163   potValue = analogRead(potPin);
164   Serial.println("Moisture: " + String(potValue));
165   Serial.print("pH Value: ");
166   Serial.println(pHValue, 1);
167
168   for (uint8_t i = 0; i < ncom; i++) {
169     Serial2.print((char)commar[i]); // send the command stored in ncom array through serial2
170     if (Serial2.available()) { //if serial2 data is there
171       rtValue[i] = Serial2.read(); // read serial2
172       Serial2.flush(); // flush serial2, very important. otherwise extra bits may interfere with communication
173       Serial.print(respar[i]); // print the response array to the console.
174       Serial.println(rtValue[i]); // print the return value with newline at console
175     }
176   }
177
178   //send data to blynk
179   Blynk.virtualWrite(V0, t);  //Temperature
180   Blynk.virtualWrite(V1, h);  //Humidity
181   Blynk.virtualWrite(V2, potValue); //soil Moisture
182   Blynk.virtualWrite(V4, rtValue[0]); //Phosphorous
183   Blynk.virtualWrite(V3, rtValue[2]); //Nitrogen
184   Blynk.virtualWrite(V5, rtValue[1]); //Potassium
185 }
186
187 void loop() {
```

WOKWI SAVE SHARE

sketch.ino ● | diagram.json | npksensor.chip.json | npksensor.chip.c | moisturesensor.chip.json | moisturesensor.chip.c | libraries.txt | Library Manager

```
186
187  void loop() {
188    // put your main code here, to run repeatedly:
189    Blynk.run();
190    timer.run();
191  }
192
```

OUTPUT: