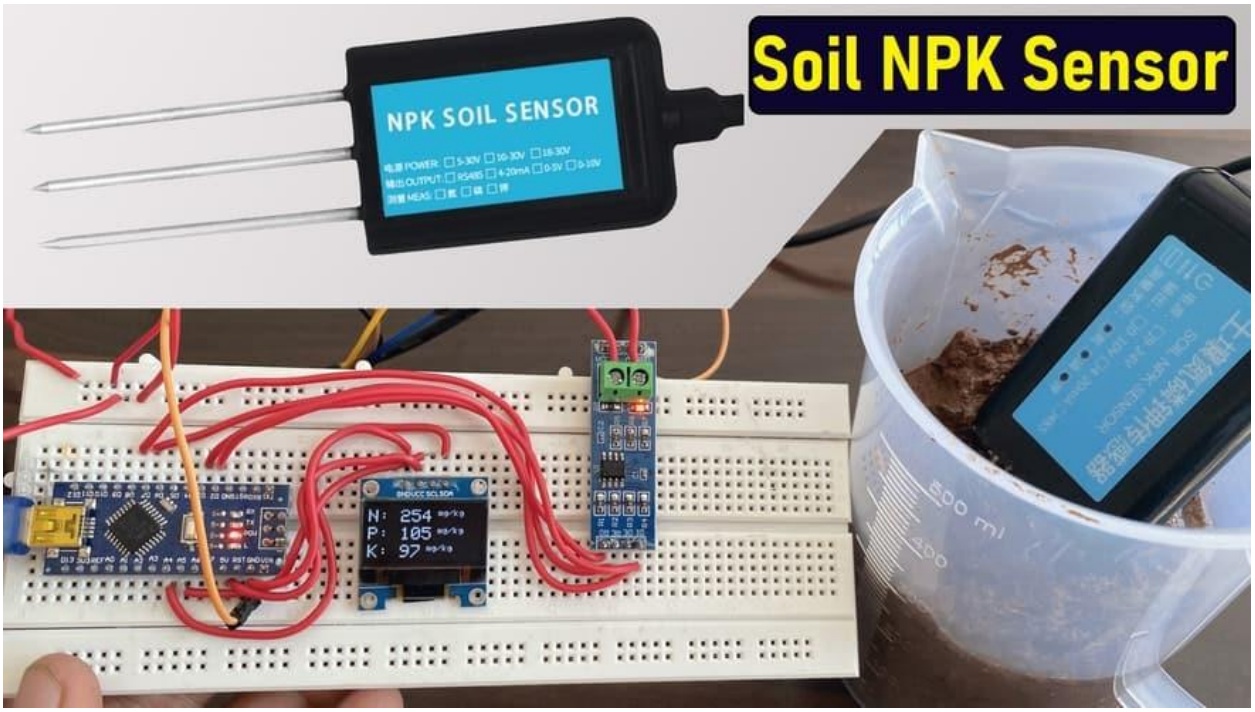
Measure Soil Nutrient using Arduino & Soil NPK Sensor



So, here we will use **Future Electronics Egypt soil NPK sensor** to detect the soil nitrogen, phosphorous & Potassium in a soil. The Soil NPK sensor is a **low cost, quick responsive, high precision & portable** Sensor that works with **Modbus RS485 (RS485 UART TTL converter)**. The advantage of this sensor over a traditional detection method is that it gives very **fast measurement** & data are **highly accurate**. All you need is to insert its probe in soil and get the reading using Arduino. So, let's learn in detail about the interfacing of Soil NPK Sensor with Arduino.

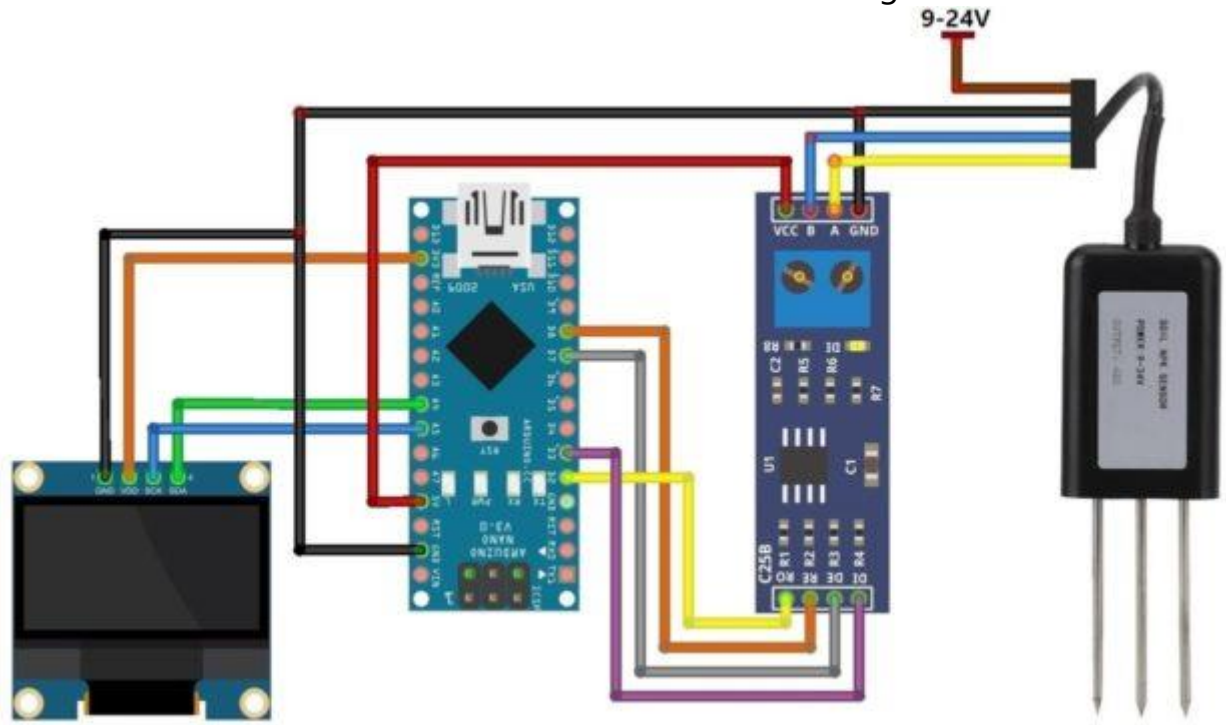
Bill of Materials

The components required for making a device that can help you in studying the Soil Nutrient Content is given below. You can purchase almost all the components from the Amazon.

S.N.	COMPONENTS NAME	DESCRIPTION	QUANTITY
1	Arduino Board	Arduino Nano or UNO	1
2	NPK Sensor	Soil NPK Sensor	1
3	OLED Display	0.96" I2C OLED Display	1
4	Modbus Module	MAX485 Modbus	1
5	Power Supply	9V - 12V DC Supply	1
6	Connecting Wires	Jumper Wires	10
7	Breadboard	-	1

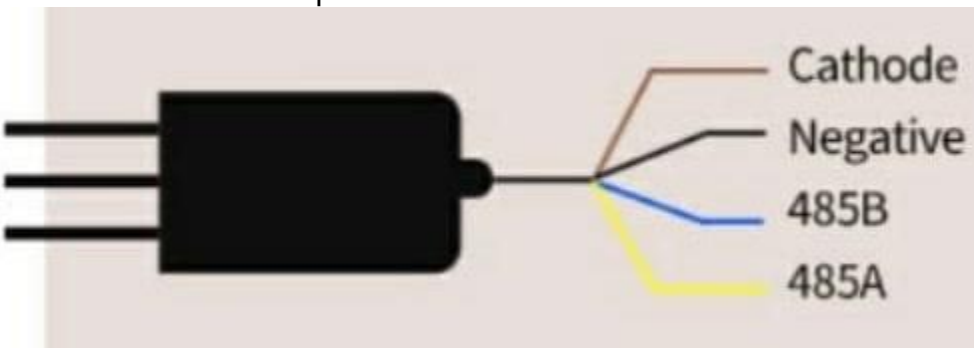
Interfacing Soil NPK Sensor with Arduino

Now, let us **interface the Soil NPK Sensor with the Arduino** Nano Board using the **MAX485 Modbus Module**. Check the connection diagram below.

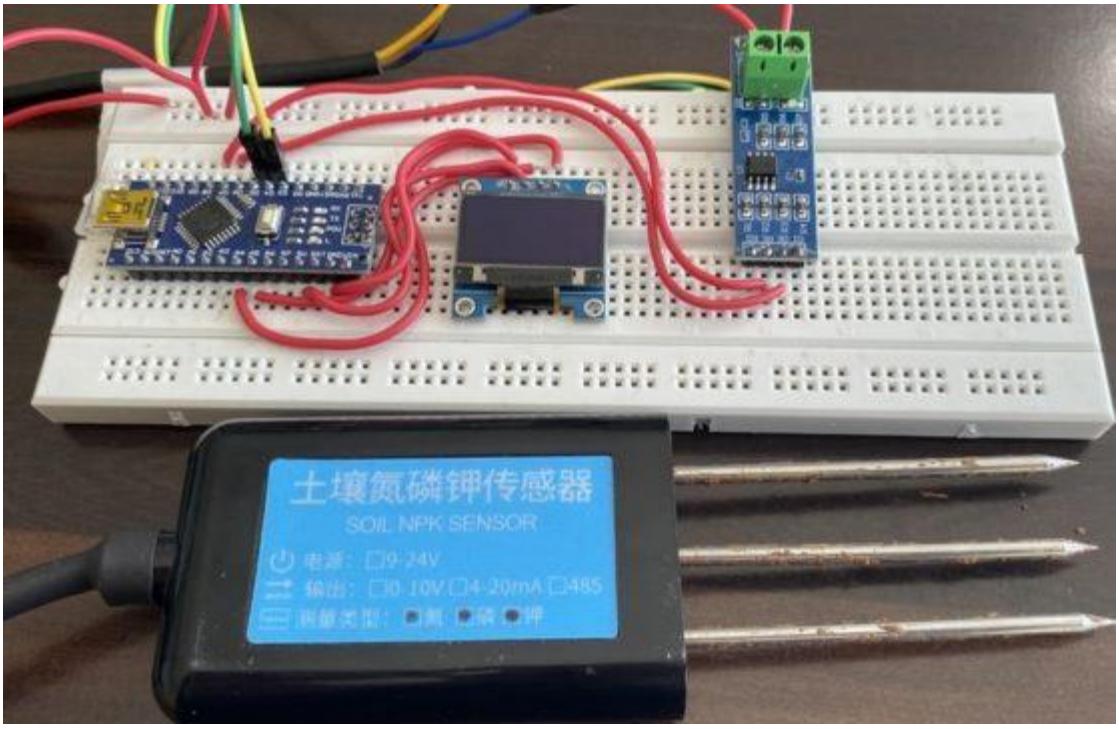


Connect the **R0 & DI** pin of from the Modbus to **D2 & D3** Arduino using Software Serial. Similarly, we have to enable **DE & RE** high. To do this connect the DE & RE Pins to the **D7 & D8** pin of Arduino. The NPK Sensor has 4 wires.
The **brown** one is VCC which needs a **9V-24V Power Supply**. The GND pin which

is **black** in color. So connect it to the GND of Arduino. The **Blue wire** which is the B pin is connected to the B pin of MAX485 & the **Yellow Wire** which is the A pin is connected to the A pin of MAX485.



The **0.96" SSD1306 OLED Display** is an I2C Module. Connect the OLED Display VCC & GND pins to **3.3V** & GND of Arduino. Similarly, connect its **SDA & SCL** pins to the A4 & A5 of Arduino. You can follow the circuit diagram & assemble the circuit on a breadboard or make a custom PCB.



Arduino Code/Program

The source code for **interfacing Soil NPK Sensor with Arduino** & retrieving Soil Nutrient value from the Sensor via Modbus command is given below. You can send the command and retrieve the value in **HEX Code**. The HEX code needs to be converted into **Decimal** to get the Measured **Soil Nutrient content data**. Since we are using OLED Display to display the Soil Nutrient values (**Nitrogen, Phosphorous & Potassium**) in **mg/kg**, you will need OLED Library. Download the following OLED Library and add it to the Arduino IDE.

1. **Adafruit SSD1306 Library**: **Download**
2. **Adafruit GFX Library**: **Download**

Here is the complete source code. Compile the code & upload it to the Arduino Nano Board.

```
#include <SoftwareSerial.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.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);

#define RE 8
#define DE 7

const byte nitro[] = {0x01,0x03, 0x00, 0x1e, 0x00, 0x01, 0xe4, 0x0c};
const byte phos[] = {0x01,0x03, 0x00, 0x1f, 0x00, 0x01, 0xb5, 0xcc};
const byte pota[] = {0x01,0x03, 0x00, 0x20, 0x00, 0x01, 0x85, 0xc0};

byte values[11];
SoftwareSerial mod(2,3);

void setup() {
  Serial.begin(9600);
  mod.begin(9600);
  pinMode(RE, OUTPUT);
  pinMode(DE, OUTPUT);

  display.begin(SSD1306_SWITCHCAPVCC, 0x3C); //initialize with the I2C addr 0x3C (128x64)
  delay(500);
  display.clearDisplay();
  display.setCursor(25, 15);
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.println(" NPK Sensor");
  display.setCursor(25, 35);
  display.setTextSize(1);
  display.print("Initializing");
  display.display();
  delay(3000);
}
```

```
void loop() {
  byte val1,val2,val3;
  val1 = nitrogen();
  delay(250);
  val2 = phosphorous();
  delay(250);
  val3 = potassium();
  delay(250);


  Serial.print("Nitrogen: ");
  Serial.print(val1);
  Serial.println(" mg/kg");
  Serial.print("Phosphorous: ");
  Serial.print(val2);
  Serial.println(" mg/kg");
  Serial.print("Potassium: ");
  Serial.print(val3);
  Serial.println(" mg/kg");
  delay(2000);

  display.clearDisplay();


  display.setTextSize(2);
  display.setCursor(0, 5);
  display.print("N: ");
  display.print(val1);
  display.setTextSize(1);
  display.print(" mg/kg");

  display.setTextSize(2);
  display.setCursor(0, 25);
  display.print("P: ");
  display.print(val2);
  display.setTextSize(1);
  display.print(" mg/kg");

  display.setTextSize(2);
  display.setCursor(0, 45);
  display.print("K: ");
  display.print(val3);
  display.setTextSize(1);
  display.print(" mg/kg");

  display.display();
}

byte nitrogen(){
  digitalWrite(DE,HIGH);
  digitalWrite(RE,HIGH);
  delay(10);
  if(mod.write(nitro,sizeof(nitro))==8){
    digitalWrite(DE,LOW);
    digitalWrite(RE,LOW);
    for(byte i=0;i<7;i++){
    //Serial.print(mod.read(),HEX);
    values[i] = mod.read();
    Serial.print(values[i],HEX);
    }
    Serial.println();
```

```
  }
  return values[4];
}

byte phosphorous(){
  digitalWrite(DE,HIGH);
  digitalWrite(RE,HIGH);
  delay(10);
  if(mod.write(phos,sizeof(phos))==8){
    digitalWrite(DE,LOW);
    digitalWrite(RE,LOW);
    for(byte i=0;i<7;i++){
    //Serial.print(mod.read(),HEX);
    values[i] = mod.read();
    Serial.print(values[i],HEX);
    }
    Serial.println();
  }
  return values[4];
}

byte potassium(){
  digitalWrite(DE,HIGH);
  digitalWrite(RE,HIGH);
  delay(10);
  if(mod.write(pota,sizeof(pota))==8){
    digitalWrite(DE,LOW);
    digitalWrite(RE,LOW);
    for(byte i=0;i<7;i++){
    //Serial.print(mod.read(),HEX);
    values[i] = mod.read();
    Serial.print(values[i],HEX);
    }
    Serial.println();
  }
  return values[4];
}
```