# Soil Nutrient Measurement using Arduino Nano & Soil NPK Sensor



## JUNE 30 ’22

**School of Interdisciplinary Engineering & Sciences (SINES) Authored by: Javaria Ishtiaq**

## Supervisor: Dr. Shahzad Younis

**Contents**

[Acknowledgements 2](#_bookmark0)

[Overview 2](#_TOC_250000)

[NPK SENSOR 3](#_bookmark1)

[SPECIFICATIONS 3](#_bookmark2)

[PINOUT 3](#_bookmark3)

[MAX485 TTL to RS-485 Interface Module 4](#_bookmark4)

[SPECIFICATIONS 4](#_bookmark5)

PINOUT 4

[OLED 5](#_bookmark6)

[SPECIFICATIONS 5](#_bookmark7)

PINOUT 5

[CODE 5](#_bookmark8)

[Source Code 5](#_bookmark9)

[Procedure to upload the code to Arduino Nano 8](#_bookmark10)

[Interfacing the Sensor with the Arduino 10](#_bookmark11)

[DEMO VIDEO 12](#_bookmark12)

[PROBLEMS & THEIR SOLUTIONS 13](#_bookmark13)

[CONCLUSION 15](#_bookmark14)

[REFERENCES 16](#_bookmark15)

# Acknowledgments

This project is carried out by Javaria Ishtiaq under the supervision of Dr. Shahzad Younis for the Summer Internship 2022 in SINES. I took most of the information and help from the internet. Reporting any error or discrepancy found in the text is appreciated.

# Overview

The project is used to determine the soil nutrients with the help of an NPK sensor and displays it on the Arduino monitor as well as on OLED. We used RS485 as the communication modbus and Arduino Nano as a microcontroller. The sensor when dipped in the soil measures the quantity of the three important nutrients of the soil which are N (nitrogen), P (phosphorous), and K (potassium).

## NPK SENSOR

The soil NPK sensor is suitable for detecting the content of nitrogen, phosphorus, and potassium in the soil, and judging the fertility of the soil by detecting the content of N, P, and K in the soil. The sensor doesn’t require any chemical reagent. Since it has High measurement accuracy, fast response speed, and good interchangeability, it can be used with any microcontroller. You cannot use the sensor directly with the microcontroller as it has a Modbus Communication port. Hence you need any Modbus Module like RS485/MAX485 and connect the sensor to the microcontroller. The sensor operates on 9-24V & power consumption is very low. While talking about the accuracy of the sensor, it is up to within 2%. The nitrogen, phosphorous & potassium measuring resolution is up to 1mg/kg (mg/l).

**SPECIFICATIONS**

1. Power: 9V-24V
2. Measuring Range: 0-1999 mg/kg (mg/l)
3. Operating Temperature: 5-45 °C
4. Resolution: 1mg/kg
5. Precision: ±2% F.S.
6. Output Signal: RS485

**7.** Baud Rate: 2400/4800/9600

**8.** Protection Class: IP68

**PINOUT**

**Thread color:** Connection **Brown:** Power supply 9~12V **Black:** Negative power GND **Yellow:** 485-A

**Blue:** 485-B



## MAX485 TTL to RS-485 Interface Module

The MAX485 TTL To RS485 Module converts TTL signal to RS485 for long-range, high data rate error-prone differential communication. RS485 at its core with 2 wires allows half-duplex data transmission. RS485 at its core with 2 wires allows half-duplex data transmission.

**SPECIFICATIONS**

1. Uses differential signaling for noise immunity
2. Distances up to 1200 meters
3. Speeds up to 2.5Mbit/Sec
4. Multi-drop supports up to 32 devices on the same bus
5. Red power LED
6. 5V operation

**PINOUT**

Arduino Connections

**RO:** Serial RX pin D2

**RE:** Digital output pin D8 **DE:** Digital output pin D7 **DI:** Serial TX pin D3

**GND:** Ground on the microcontroller

**VCC:** 5V on the microcontroller

Sensor Connections

**B:** Blue wire of the sensor

**A:** Yellow wire of the sensor



## OLED

An organic light-emitting diode (OLED or organic LED), also known as an organic electroluminescent (organic EL) diode is a light-emitting diode (LED) in which the emissive electroluminescent layer is a film of organic compound that emits light in response to an electric current.

**SPECIFICATIONS**

1. High-resolution at 128x64 pixels
2. Lower power consumption: only 0.06W with normal use
3. Power supply AC 3V-5V, working very well with Arduino
4. Working temperature: -30 degrees to 70 degrees Celsius
5. Driver IC SSD1306

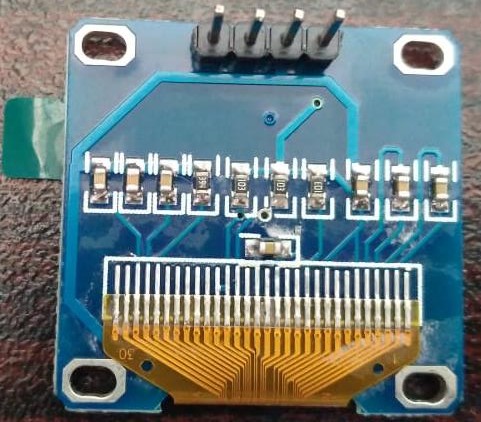
**PINOUT**

**SDA:** Pin A4

**SCL:** Pin A5

**GND:** Ground on the microcontroller

**VCC:** 3V on the microcontroller



## CODE

To start the work first download the required libraries and add them to the Arduino IDE.

1. Adafruit SSD1306 Library
2. Adafruit GFX Library

**Source Code**

#include <SoftwareSerial.h> #include <Wire.h>

#include <Adafruit\_GFX.h> //Adafruit GFX and SSD1306 library is for OLE display #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 RE and DE pins of the sensor #define DE 7

//const byte code[]= {0x01, 0x03, 0x00, 0x1e, 0x00, 0x03, 0x65, 0xCD}; //this works as a command to retrieve the data from using modbus from the and picker sensor

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); //Tx and Rx pin as pin 2 and 3

void setup() {

Serial.begin(9600); //initialize serial begin and mode begin mod.begin(4800);

pinMode(RE, OUTPUT);// keeping RE and DE as outputs 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; //we assign three variables as value 1, value 2 and value 3 val1 = nitrogen(); //assigning value 1 as nitrogen

delay(250);

val2 = phosphorous(); //assigning value 2 as phosphorous delay(250);

val3 = potassium(); //assigning value 3 as potassium delay(250);

Serial.print("Nitrogen: "); //to display value in serial moniter and on display 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); //enabling DE and RE as high digitalWrite(RE,HIGH);

delay(10);

if(mod.write(nitro,sizeof(nitro))==8){ //retrieve the value of nitrogen from the sensor digitalWrite(DE,LOW);

digitalWrite(RE,LOW); for(byte i=0;i<7;i++){

//Serial.print(mod.read(),HEX);

values[i] = mod.read(); //modbus command

Serial.print(values[i],HEX); //print the value of data in hexadecimal and convert it into decimal

}

Serial.println();

}

return values[4];

}

byte phosphorous(){ //for 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(){ //for 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();

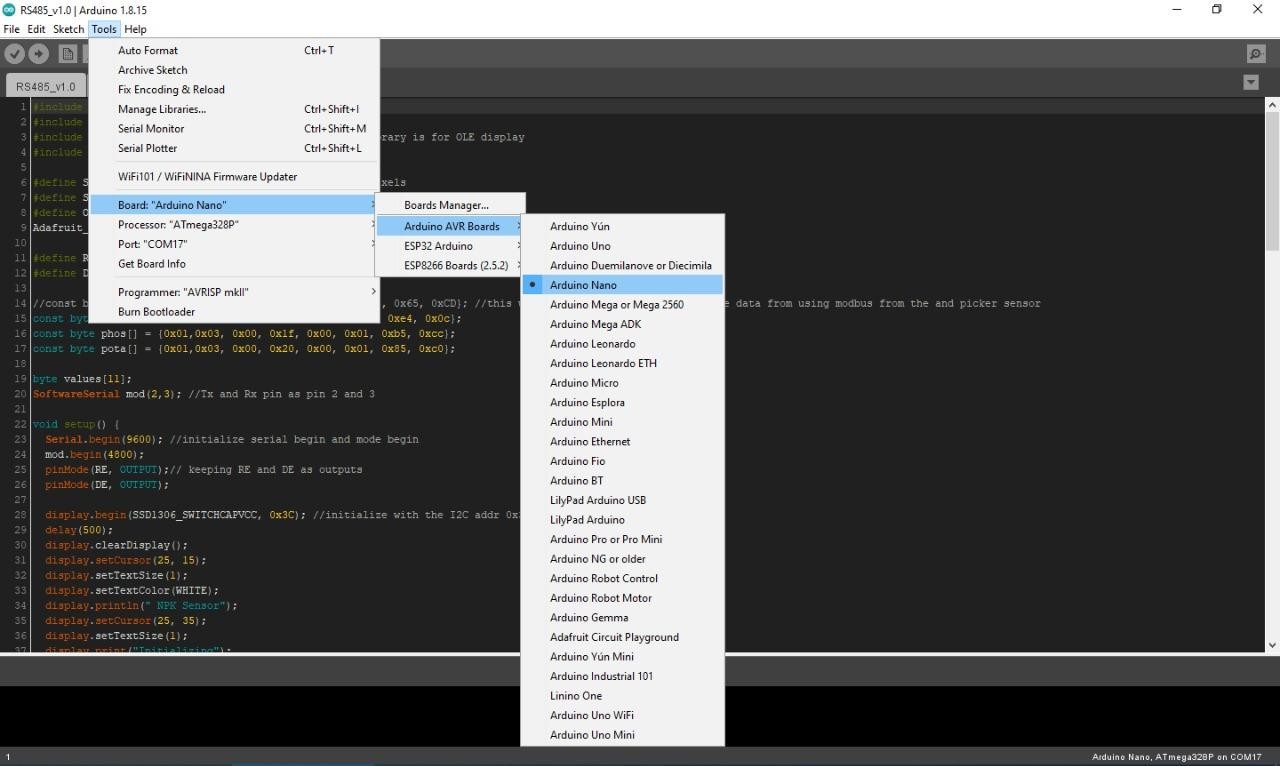
}

return values[4];

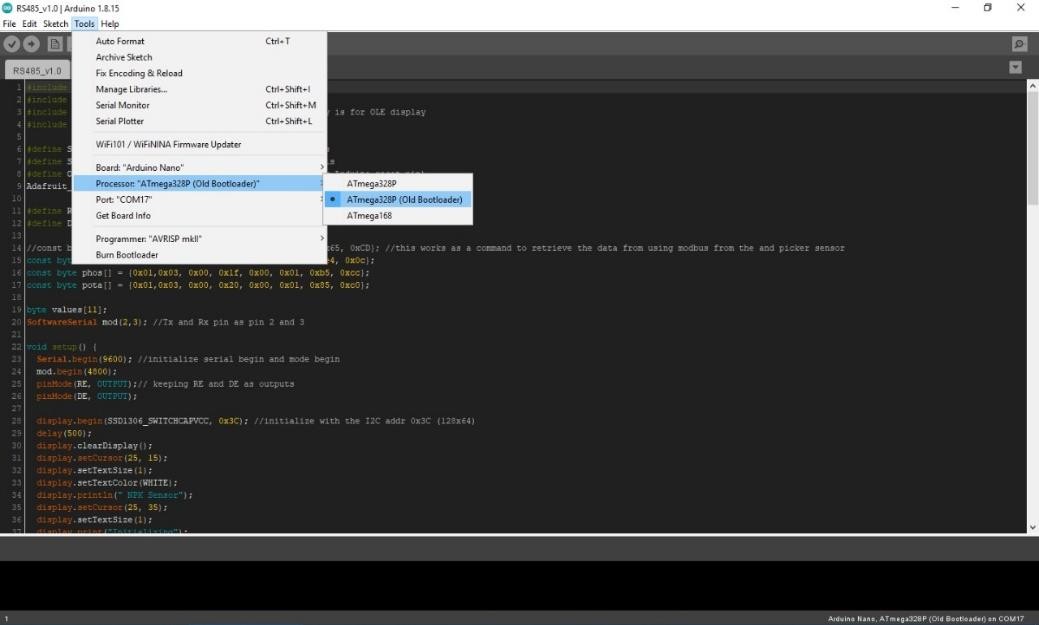
}

**Procedure to upload the code to Arduino Nano**

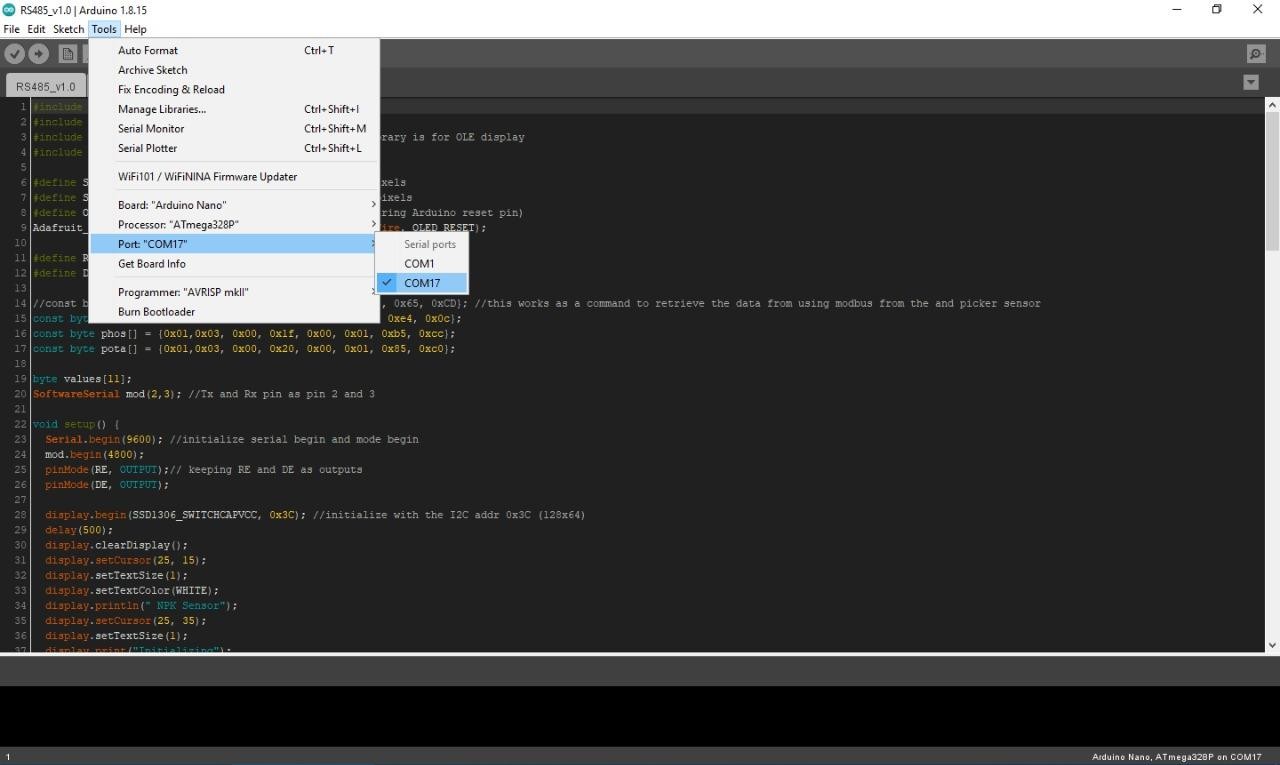
Select the board from Tools > Board > Arduino AVR Board > Arduino Nano.



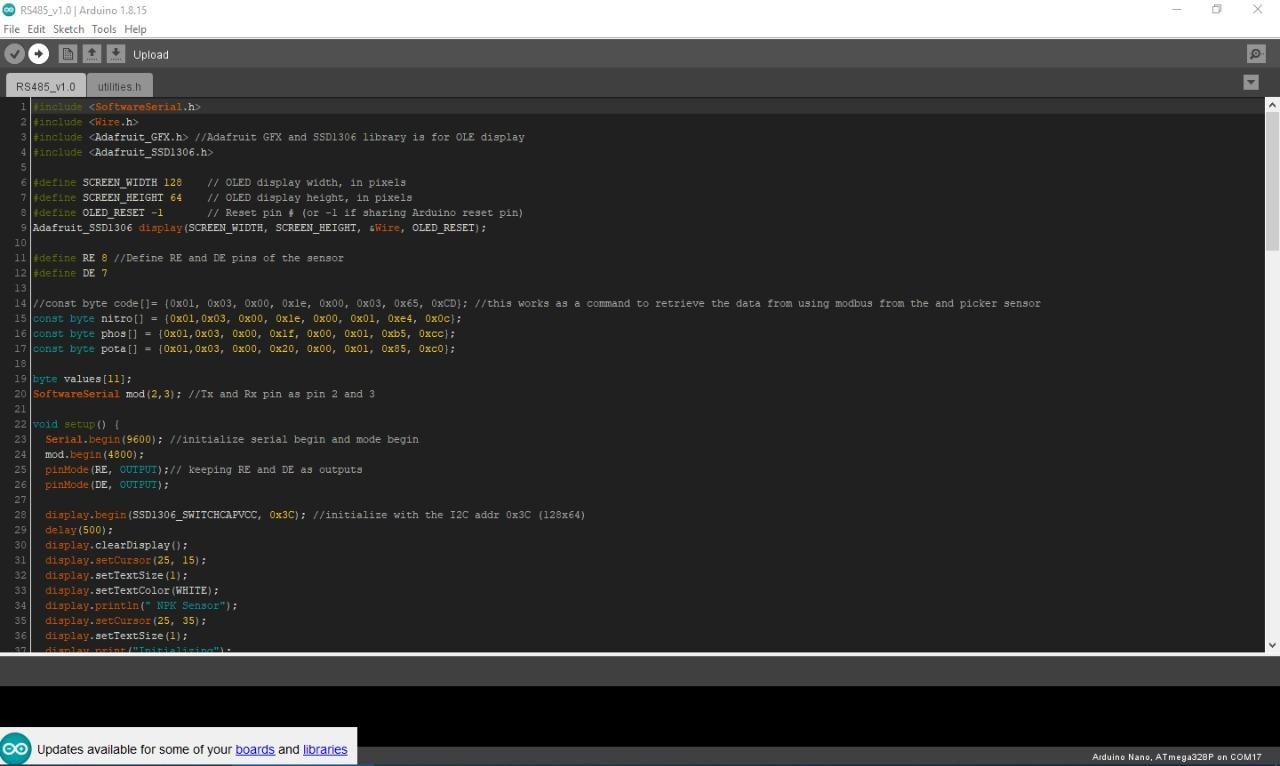
After selecting the board select the processor from Tools > Processor > ATmega328P (Old Bootloader).



Insert the USB Cable. Make sure that you have selected the right port.

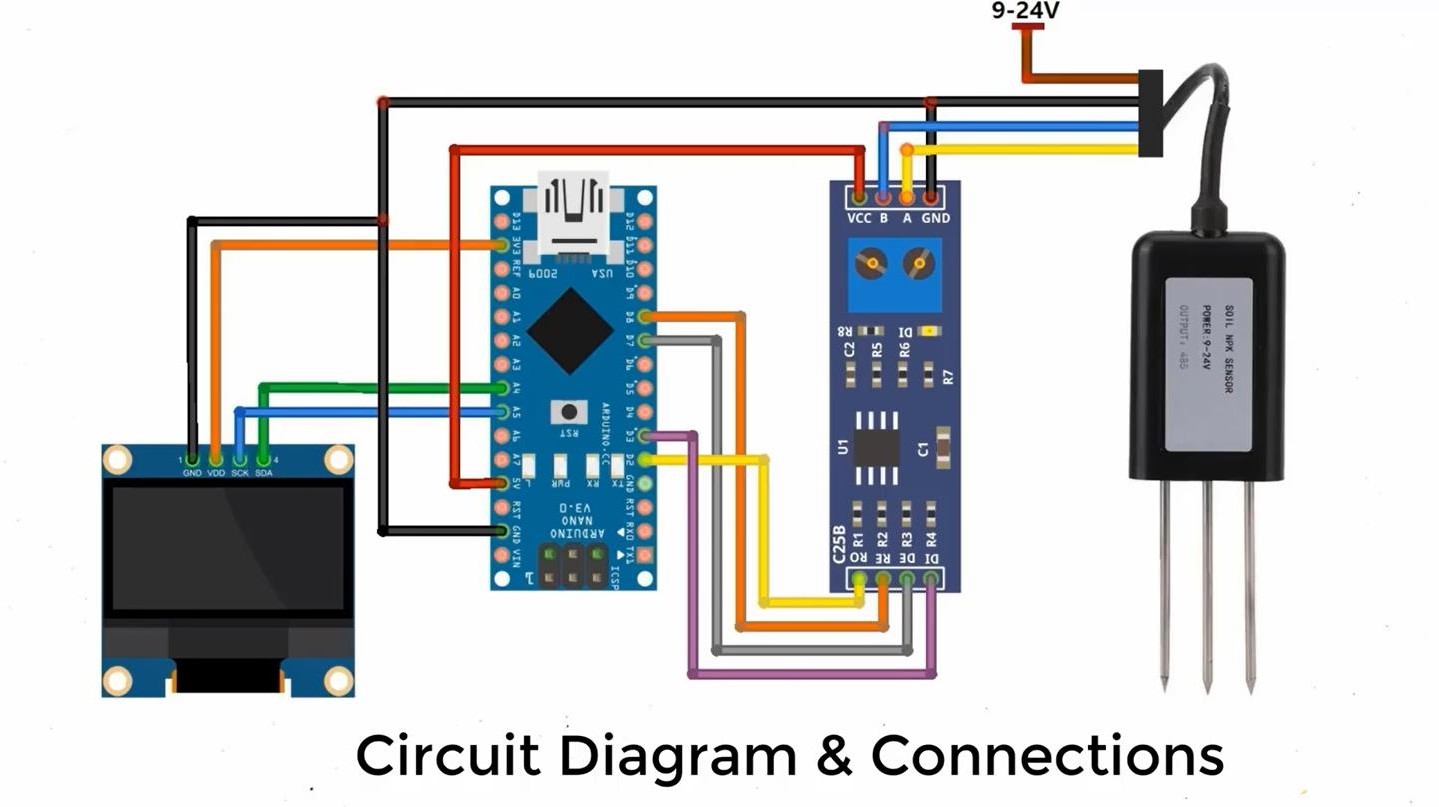


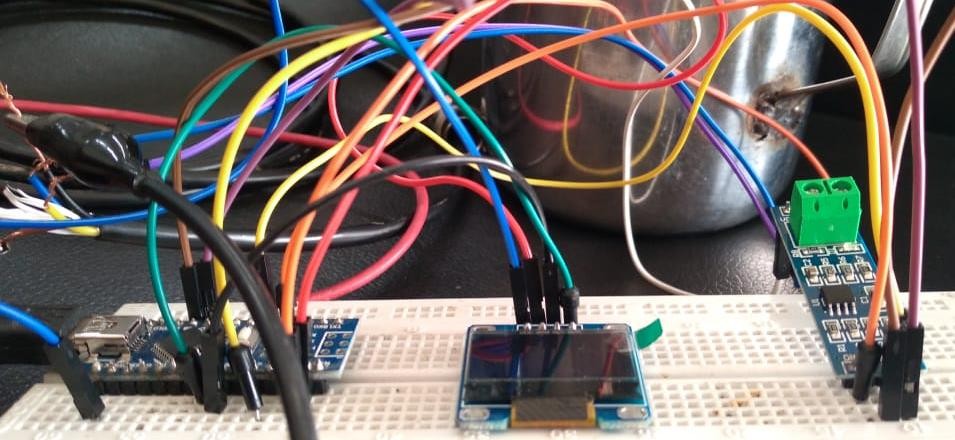
Upload the code now.

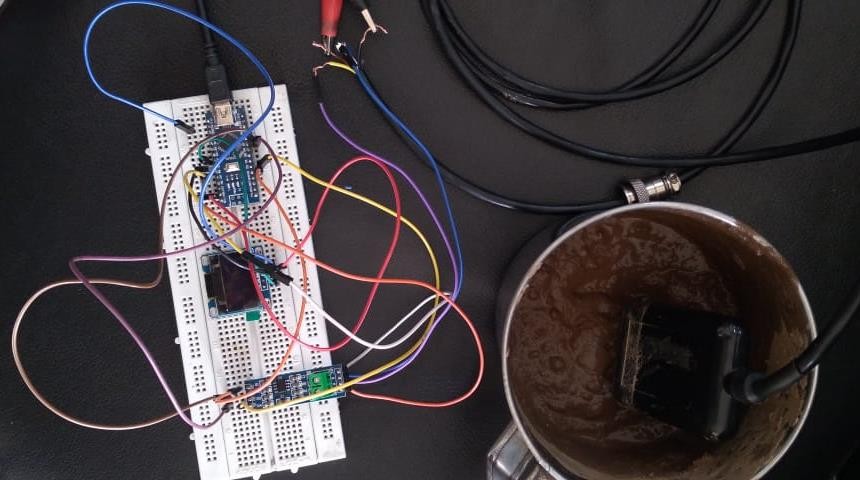
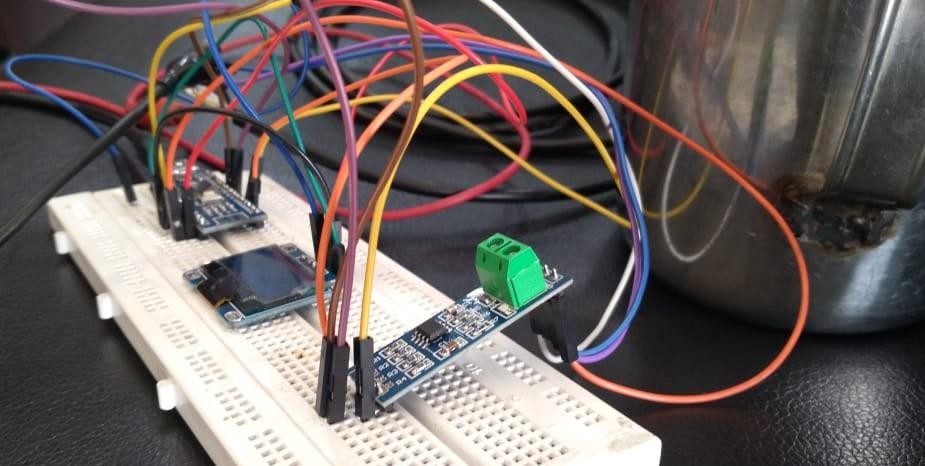


## Interfacing the Sensor with the Arduino

Just make the necessary connections as mentioned above or you can see it from the picture below.



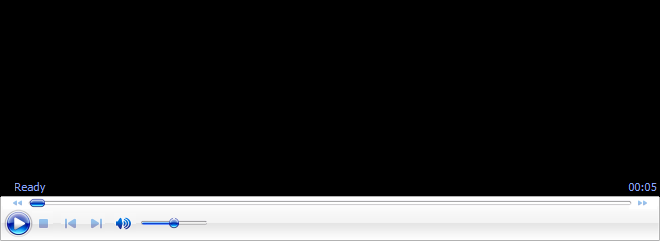




For the sensor, connect its VCC with a 9V or 12V source. I gave it a 9V source using DC Power Supply and connected its ground to the DC Power Supply ground.



## DEMO VIDEO



## PROBLEMS & THEIR SOLUTIONS

1. **Uploading the code to the Arduino Nano**

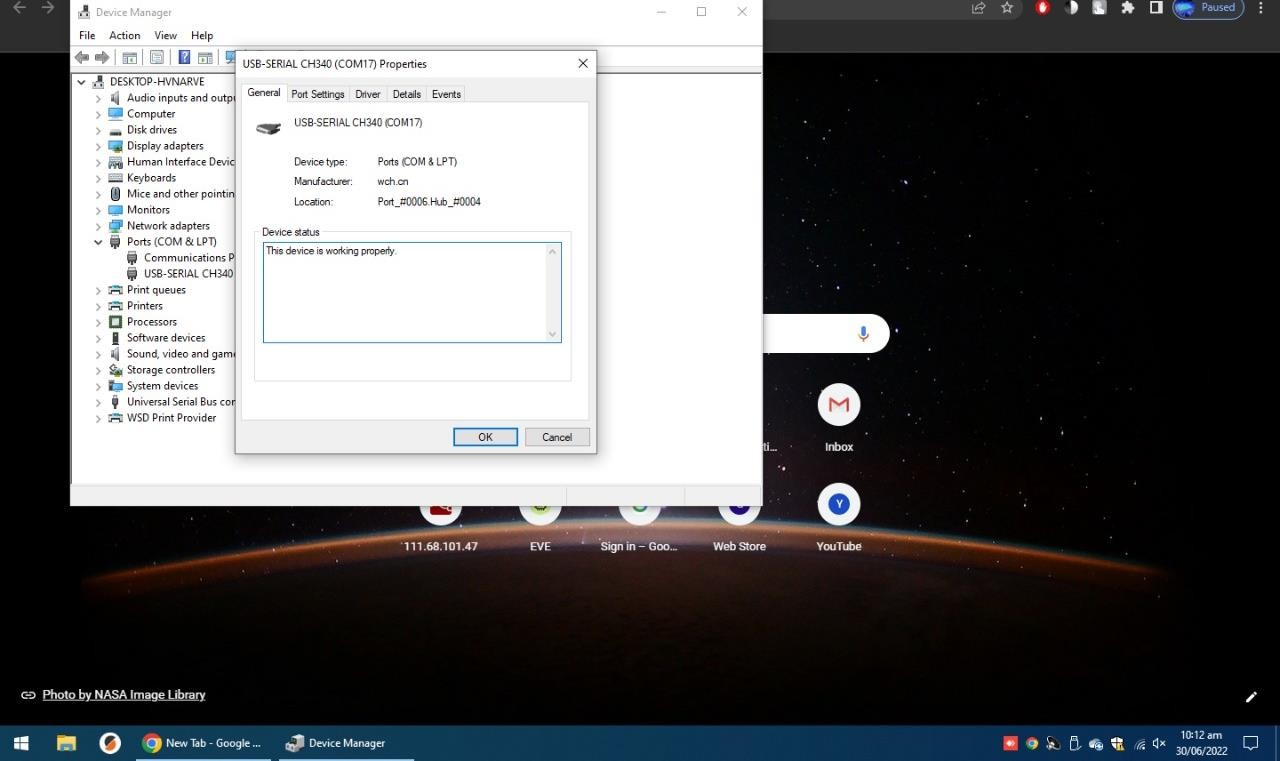
If you are unable to upload the code to your Arduino Nano first make sure that you have selected the right port. You can do so by plugging the cable into the laptop and connecting it with the Arduino Nano. Bring your cursor to the bottom of the windows logo and right-click it. There you will see an option of Device Manager.



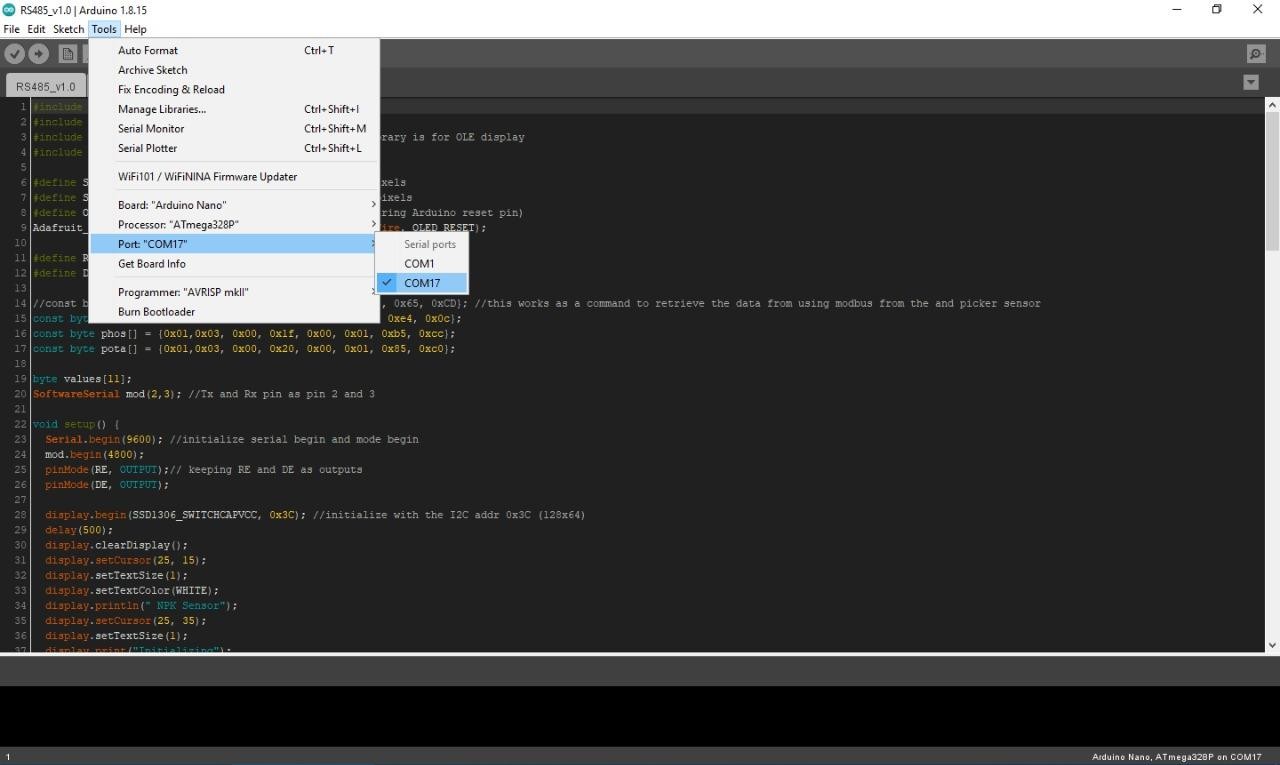
Select it and go to the options of Ports (COM &LPT)



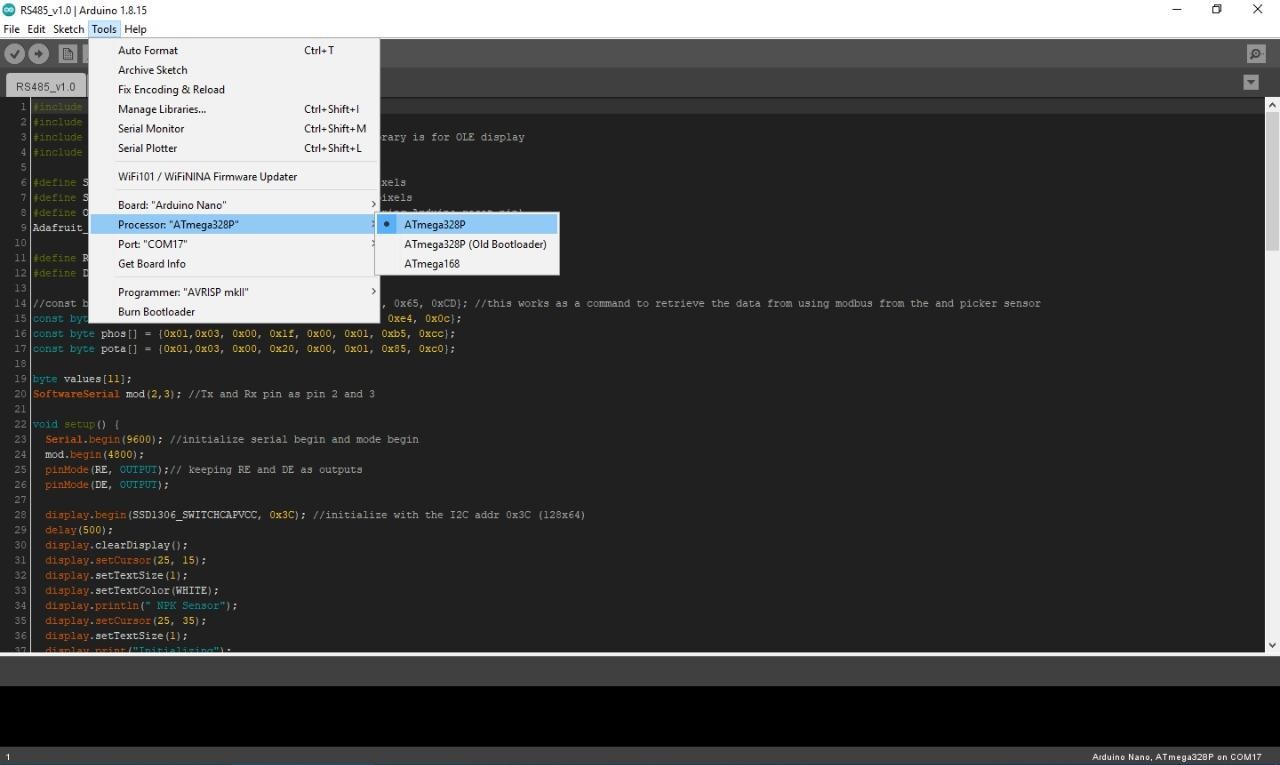
Now click on the port to which you have inserted your USB cable in. In my case it was USB-SERIAL CH340 (COM17).



Now go back to Arduino IDE and select that port.



If you are still unable to upload the code. Then try this from the processors option select ATmega328P. I first tried uploading the code using this processor but it did not happen so I had to choose the Old Bootloader.



1. **The display shows 255 or 0 as output for all N, P & K values**
   1. To resolve this issue make sure that you have connected the DI, DE, RE, and RO pins according to the code or as the connections are shown.
   2. The OLED should be provided VCC of 3V or 5v.
   3. MAX485 should be provided a VCC of 5v and common ground.
   4. The NPK Sensor needs 9V or 12V or more. Supply appropriate voltage to its VCC pin. The sensor won’t operate at 5V or anything lesser than 9V.

## CONCLUSION

I made this project just to understand the communication between Arduino and a sensor via RS485. If you don’t get the desired, correct or same answers every time then this also happened to me. The space which remains hollow in this project is to get correct and right measurements every time.

## REFERENCES

I took help from

[https://how2electronics.com/measure-soil-nutrient-using-arduino-soil-npk-](https://how2electronics.com/measure-soil-nutrient-using-arduino-soil-npk-sensor/?nowprocket=1) [sensor/#:~:text=The%20soil%20nutrient%20content%20can,soil%20to%20increase%20](https://how2electronics.com/measure-soil-nutrient-using-arduino-soil-npk-sensor/?nowprocket=1) [crop%20fertility.](https://how2electronics.com/measure-soil-nutrient-using-arduino-soil-npk-sensor/?nowprocket=1)