

```

4 #include <Adafruit_SSD1306.h>
5
6 #define SCREEN_WIDTH 128    // OLED display width, in pixels
7 #define SCREEN_HEIGHT 64    // OLED display height, in pixels
8 #define OLED_RESET -1       // Reset pin # (or -1 if sharing Arduino reset pin)
9 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
10
11 #define RE 8
12 #define DE 7
13
14 //const byte code[]= {0x01, 0x03, 0x00, 0x1e, 0x00, 0x03, 0x65, 0xCD};
15 const byte nitro[] = {0x01,0x03, 0x00, 0x1e, 0x00, 0x01, 0xe4, 0x0c};
16 const byte phos[] = {0x01,0x03, 0x00, 0x1f, 0x00, 0x01, 0xb5, 0xcc};
17 const byte pota[] = {0x01,0x03, 0x00, 0x20, 0x00, 0x01, 0x85, 0xc0};
18
19 byte values[11];
20 SoftwareSerial mod(2,3);
21
22 void setup() {
23     Serial.begin(9600);
24     mod.begin(9600);
25     pinMode(RE, OUTPUT);
26     pinMode(DE, OUTPUT);
27
28     display.begin(SSD1306_SWITCHCAPVCC, 0x3C); //initialize with the I2C addr 0x3C (128x64)
29     delay(500);
30     display.clearDisplay();
31     display.setCursor(25, 15);
32     display.setTextSize(1);
33     display.setTextColor(WHITE);
34     display.println(" NPK Sensor");
35     display.setCursor(25, 35);
36     display.setTextSize(1);
37     display.print("Initializing");
38     display.display();
39     delay(3000);
40 }
41
42 void loop() {
43     byte val1, val2, val3;
44     val1 = nitrogen();
45     delay(250);
46     val2 = phosphorous();
47     delay(250);
48     val3 = potassium();
49     delay(250);
50
51
52     Serial.print("Nitrogen: ");
53     Serial.print(val1);

```

```

54 Serial.println(" mg/kg");
55 Serial.print("Phosphorous: ");
56 Serial.print(val2);
57 Serial.println(" mg/kg");
58 Serial.print("Potassium: ");
59 Serial.print(val3);
60 Serial.println(" mg/kg");
61 delay(2000);
62
63 display.clearDisplay();
64
65
66 display.setTextSize(2);
67 display.setCursor(0, 5);
68 display.print("N: ");
69 display.print(val1);
70 display.setTextSize(1);
71 display.print(" mg/kg");
72
73 display.setTextSize(2);
74 display.setCursor(0, 25);
75 display.print("P: ");
76 display.print(val2);
77 display.setTextSize(1);
78 display.print(" mg/kg");
79
80 display.setTextSize(2);
81 display.setCursor(0, 45);
82 display.print("K: ");
83 display.print(val3);
84 display.setTextSize(1);
85 display.print(" mg/kg");
86
87 display.display();
88 }
89
90 byte nitrogen(){
91   digitalWrite(DE,HIGH);
92   digitalWrite(RE,HIGH);
93   delay(10);
94   if(mod.write(nitro , sizeof(nitro))==8){
95     digitalWrite(DE,LOW);
96     digitalWrite(RE,LOW);
97     for(byte i=0;i<7;i++){
98       // Serial.print(mod.read(),HEX);
99       values[i] = mod.read();
100      Serial.print(values[i],HEX);
101     }
102     Serial.println();
103   }

```

```

104     return values[4];
105 }
106
107 byte phosphorous() {
108     digitalWrite(DE,HIGH);
109     digitalWrite(RE,HIGH);
110     delay(10);
111     if(mod.write(phos,sizeof(phos))==8){
112         digitalWrite(DE,LOW);
113         digitalWrite(RE,LOW);
114         for(byte i=0;i<7;i++){
115             // Serial.print(mod.read(),HEX);
116             values[i] = mod.read();
117             Serial.print(values[i],HEX);
118         }
119         Serial.println();
120     }
121     return values[4];
122 }
123
124 byte potassium() {
125     digitalWrite(DE,HIGH);
126     digitalWrite(RE,HIGH);
127     delay(10);
128     if(mod.write(pota,sizeof(pota))==8){
129         digitalWrite(DE,LOW);
130         digitalWrite(RE,LOW);
131         for(byte i=0;i<7;i++){
132             // Serial.print(mod.read(),HEX);
133             values[i] = mod.read();
134             Serial.print(values[i],HEX);
135         }
136         Serial.println();
137     }
138     return values[4];
139 }

```