

Code for nitrogen assessment and prescription

```
#include "AS726X.h"
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <NewPing.h>
#include <Keypad.h>

AS726X sensor;

LiquidCrystal_I2C lcd(0x27, 20, 4);

const int TRIGGER_PIN = 3;
const int ECHO_PIN = 4;

NewPing ultrasonicSensor(TRIGGER_PIN, ECHO_PIN);

const int buttonPin = 2;

float ndviValue, ndreValue, rviValue;

int growthStage = 0;

bool growthStageEntered = false;

const byte ROWS = 4; // Four rows
const byte COLS = 3; // Three columns
char keys[ROWS][COLS] = {
  {'1', '2', '3'},
  {'4', '5', '6'},
  {'7', '8', '9'},
  {'*', '0', '#'}
};

byte rowPins[ROWS] = {5, 6, 7, 8}; // Connect to the row pinouts of the keypad
byte colPins[COLS] = {9, 10, 11}; // Connect to the column pinouts of the keypad
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

void setup() {
  Wire.begin();
  sensor.begin();
```

```
lcd.begin();
lcd.backlight();
lcd.setBacklight(HIGH);
pinMode(buttonPin, INPUT_PULLUP);
lcd.setCursor(0, 0);
lcd.print("N Level Estimator");
lcd.setCursor(0,1);
lcd.print("Place This Device");
lcd.setCursor(0,2);
lcd.print("Over Crop Canopy");
lcd.setCursor(0,3);
lcd.print("Lets Start");
delay(5000);
lcd.clear();
lcd.print("Enter Growth Stage:");
lcd.setCursor(0, 1);
}
void loop()
{
if (!growthStageEntered) {
char key = keypad.getKey();
if (key != NO_KEY) {
if (key >= '0' && key <= '9') {
growthStage = growthStage * 10 + (key - '0');
lcd.print(key);
} else if (key == '#') {
growthStageEntered = true;
lcd.clear();
lcd.print("Growth Stage: ");
```

```

lcd.print(growthStage);

lcd.setCursor(0, 1);
}
}
delay(100); // Add a small delay to prevent rapid key presses
return; // Exit loop until growth stage is entered
}
unsigned int distanceCm = ultrasonicSensor.ping_cm();
sensor.takeMeasurements();
if (sensor.getVersion() == SENSORTYPE_AS7263)
{
float a = sensor.getCalibratedR();
float b = sensor.getCalibratedS();
float c = sensor.getCalibratedT();
float d = sensor.getCalibratedU();
float e = sensor.getCalibratedV();
float f = sensor.getCalibratedW();
float m = (a/(a+b+c+d+e+f));
float n = (b/(a+b+c+d+e+f));
float o = (c/(a+b+c+d+e+f));
float p = (d/(a+b+c+d+e+f));
float q = (e/(a+b+c+d+e+f));
float r = (f/(a+b+c+d+e+f));
float x = ((d - b) / (d + b));
lcd.clear();
lcd.setCursor(0,0);
lcd.print("NDVI=");
lcd.print(x);
lcd.setCursor(11,0);

```

```

lcd.print("RVI=");
lcd.print(d/b);

lcd.setCursor(0,1);
lcd.print(" NDRE=");
lcd.print((d-c)/(d+c));
lcd.setCursor(0,2);
lcd.print(" Distance=");
lcd.print(distanceCm);
lcd.print(" cm");
if (digitalRead(buttonPin) == LOW)
{
float ndviSum = 0, ndreSum = 0, rviSum = 0;
int samples = 1000;
for (int i = 0; i < samples; i++)
{
ndviSum += x;
ndreSum += ((d-c)/(d+c));
rviSum += (d/b);
}
ndviValue = ndviSum / samples;
ndreValue = ndreSum / samples;
rviValue = rviSum / samples;
lcd.clear();
lcd.print("Average NDVI: ");
lcd.print(ndviValue);
lcd.setCursor(0, 1);
lcd.print("Average NDRE: ");
lcd.print(ndreValue);
lcd.setCursor(0, 2);

```

```
lcd.print("Average RVI: ");
lcd.print(rviValue);
lcd.setCursor(0, 3);
float Nitrogen Dose = 21.64 - 4.03 * ndviValue - 0.497 * ndreValue - 2.023 * rviValue +
0.003 * growthStage;
if (Nitrogen Dose < 2) {
lcd.print("N not required");
} else {
lcd.print("N Dose:");
lcd.setCursor(8, 3);
lcd.print(Nitrogen Dose);
lcd.setCursor(14, 3);
lcd.print("kg/ac");
}
delay(5000); // Debounce delay
}}}
```