

```

// Includen van alle benodigde libraries
#include <Arduino.h>
#include <WiFi.h>
#include <HTTPClient.h>
#include <ArduinoJson.h>
#include <DHT.h>
#include <ESP32Servo.h>

// Definieren van alle datapins van alle componenten
#define DHTPIN 14
#define DHTTYPE DHT11
#define servopin 15
#define echoPin 26
#define grondPin 33
#define trigPin 27
#define lichtPin1 32
#define lichtPin2 35
#define lichtPin3 34
#define greenpin 2
#define redpin 4

// magic number handlers
const int procentHulpwaarde = -1;
const int nulWaarde = 0;
const int potDiepte = 30;
const int grensWaardepostUpdate = 10;
const int procentRekenwaarde = 100;
const int sensorRekenwaarde = 4095;
const int timerDelaywaarde = 1000;

// Declaratie variabelen
String strluchtvochtigheid;
String strtemperatuur;
String strgrondvochtigheid;
String strlichtsterkte1;
String strlichtsterkte2;
String strlichtsterkte3;
String strwaterniveau;

double dblluchtvochtigheid = nulWaarde;
double dbltemperatuur = nulWaarde;
double lichtSterkte1;
double lichtSterkte2;
double lichtSterkte3;

int distance;
int sensorValueGrond;
int postCounter;

long duration;
unsigned long lastTime = nulWaarde;
unsigned long timerDelay = timerDelaywaarde;

// Aanroepen van benodigdheden voor miscellaneous functies binnen de code
DHT dht(DHTPIN, DHTTYPE);
StaticJsonDocument<200> doc;
Servo myservo;

// Declaratie van WiFi netwerken die gebruikt kunnen worden

//const char* ssid = "schenktol";
//const char* password = "TLE143TLE";

```

```

//const char* ssid = "wjavandertol";
//const char* password = "WT3030wt";

const char* ssid = "12connect";
const char* password = "";

// Declaratie van de eindbestemming
const char* serverName = "http://smartpot.nealgeilen.nl/api/addData";

//const char* serverName = "https://collect2.com/api/02f27a5d-b70b-4cc2-b451-
fd9e89be984f/datarecord/";

void setup()
{
  Serial.begin(115200);
  myservo.attach(servopin);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(greenpin, OUTPUT);
  pinMode(redpin, OUTPUT);
  //pinMode(redPin, OUTPUT);
  //pinMode(greenPin, OUTPUT);

  dht.begin();

  WiFi.begin(ssid, password);
  Serial.println("Connecting");

  while(WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("Verbonden met WiFi, met Ip adres: ");
  Serial.println(WiFi.localIP());
}

void loop()
{
  // Ultrasonic sensor
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  // Natuurkundige berekening voor de hoeveelheid water in het reservoir
  distance = duration * 0.034 / 2;

  if (distance < 5)
  {
    digitalWrite(greenpin, HIGH);
    digitalWrite(redpin, LOW);
  }

  if (distance > 15)
  {
    digitalWrite(greenpin, HIGH);
    digitalWrite(redpin, HIGH);
  }
}

```

```

}

if (distance > 25)
{
    digitalWrite(greenpin, LOW);
    digitalWrite(redpin, HIGH);
}
if ((millis() - lastTime) > timerDelay)
{

    if(WiFi.status()== WL_CONNECTED)
    {
        // Ultrasonic sensor
        digitalWrite(trigPin, LOW);
        delayMicroseconds(2);
        digitalWrite(trigPin, HIGH);
        delayMicroseconds(10);
        digitalWrite(trigPin, LOW);
        duration = pulseIn(echoPin, HIGH);
        // Natuurkundige berekening voor de hoeveelheid water in het reservoir
        distance = duration * 0.034 / 2;

    if (distance < 5)
    {
        digitalWrite(greenpin, HIGH);
        digitalWrite(redpin, LOW);
    }

    if (distance > 15)
    {
        digitalWrite(greenpin, HIGH);
        digitalWrite(redpin, HIGH);
    }

    if (distance > 25)
    {
        digitalWrite(greenpin, LOW);
        digitalWrite(redpin, HIGH);
    }

    //Lees alle sensoren uit

    // DHT sensor
    dblluchtvochtigheid = dht.readHumidity();
    dbltemperatuur = dht.readTemperature();

    // Grondvochtigheids sensor
    sensorValueGrond = analogRead(grondPin);

    // Ultrasonic sensor
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    // Natuurkundige berekening voor de afstand
    distance = duration * 0.034 / 2;

    // Lichtsterkte sensoren
    lichtSterkte1 = analogRead(lichtPin1);
    lichtSterkte2 = analogRead(lichtPin2);

```

```

lichtSterkte3 = analogRead(lichtPin3);

// Print alle waardes uit (voor debugging)
Serial.print("Distance: ");
Serial.print(distance);
Serial.println(" cm");
Serial.print("Temperatuur: ");
Serial.println(dbltemperatuur);
Serial.print("Luchtvochtigheid: ");
Serial.println(dblluchtvochtigheid);
Serial.print("Grondvochtigheid: ");
Serial.println(sensorValueGrond);
Serial.print("Lichtsterkte sensor 1: ");
Serial.println(lichtSterkte1);
Serial.print("Lichtsterkte sensor 2: ");
Serial.println(lichtSterkte2);
Serial.print("Lichtsterkte sensor 3: ");
Serial.println(lichtSterkte3);

HTTPClient http;
http.begin(serverName);

//procentuele omrekening van alle getallen
sensorValueGrond = (((sensorValueGrond - sensorRekenwaarde) *
procentHulpwaarde) * procentRekenwaarde) / sensorRekenwaarde;
distance = ((distance * procentRekenwaarde) / potDiepte) - procentRekenwaarde;
lichtSterkte1 = (lichtSterkte1 * procentRekenwaarde) / sensorRekenwaarde;
lichtSterkte2 = (lichtSterkte2 * procentRekenwaarde) / sensorRekenwaarde;
lichtSterkte3 = (lichtSterkte3 * procentRekenwaarde) / sensorRekenwaarde;

// Omzetten van alle getallen naar strings, zo kunnen ze worden meegegeven in de
JSON string
strtemperatuur = strtemperatuur + dbltemperatuur;
strluchtvochtigheid = strluchtvochtigheid + dblluchtvochtigheid;
strwaterniveau = strwaterniveau + distance;
strgrondvochtigheid = sensorValueGrond + strgrondvochtigheid;
strlichtsterkte1 = strlichtsterkte1 + lichtSterkte1;
strlichtsterkte2 = strlichtsterkte2 + lichtSterkte2;
strlichtsterkte3 = strlichtsterkte3 + lichtSterkte3;

if (sensorValueGrond < 50)
{
    myservo.write(180);
    delay(1000);
    myservo.write(90);
}

// Toevoegen van headers aan de HTTP Post, zo kunnen we gebruik maken van
gebruikersnamen en wachtwoorden
http.addHeader("Content-Type", "application/json");
http.addHeader("X-AUTH-TOKEN", "TEST");
http.addHeader("X-AUTH-ID", "wajdhlawkjhdlawjkdhawkjdh");
http.addHeader("Temperatuur", strtemperatuur);
http.addHeader("Luchtvochtigheid", strluchtvochtigheid);
http.addHeader("Grondvochtigheid", strgrondvochtigheid);
http.addHeader("Waterniveau", strwaterniveau);
http.addHeader("Lichtsterkte1", strlichtsterkte1);
http.addHeader("Lichtsterkte2", strlichtsterkte2);
http.addHeader("Lichtsterkte3", strlichtsterkte3);

// HTTP Post

```

```

        int httpResponseCode = http.POST("{ \"Temperatuur\": "+strtemperatuur+" ,
        \"/>

```