Smart Plant System

SSC  
Vlissingen

Projectleider : Rutger Haaze

Projectnummer : 1

Datum : 25-11-2019

Versie :

**Inhoud**

Inhoud

[**1. Inleiding** 3](#_Toc26429328)

[1.1 Algemeen 3](#_Toc26429329)

[1.2 Inhoud van technisch ontwerp 3](#_Toc26429330)

[**2. Fysiek ontwerp** 4](#_Toc26429331)

[2.1 Planoverzicht 4](#_Toc26429332)

[2.2 Opsomming te verrichte activiteiten 5](#_Toc26429333)

[2.3 Benodigd gereedschap 5](#_Toc26429334)

[**3. Inrichting** 6](#_Toc26429335)

[3.1 Installatieoverzicht 6](#_Toc26429336)

[3.2 Assemblatie 6](#_Toc26429337)

[**4. Testplan** 8](#_Toc26429338)

[4.1 Waarom wordt er getest 8](#_Toc26429339)

[4.2 Wanneer testen 8](#_Toc26429340)

[4.3 Wat wordt getest 8](#_Toc26429341)

[4.4 Waar wordt getest 8](#_Toc26429342)

[4.5 Wie gaat testen 8](#_Toc26429343)

[**5. Handleiding gebruik van Smart Plant System** 9](#_Toc26429344)

# **Inleiding**

## Algemeen

De klant wil tomaten kweken zoals hij ze in Italië heeft gezien. Het systeem dat dit zal

gaan doen, zal worden gebouwd tijdens het project en zal de volgende

functionaliteiten hebben:

* Ingebouwd waterreservoir.
* Sensor voor het water niveau in het reservoir.
* Sensor voor het meten van de grondvochtigheid.
* Ingebouwde groeilamp met timer.
* LED die aangeeft wanneer het systeem werkt en het water reservoir bijgevuld moet worden.

In dit technisch ontwerp zal vastgesteld worden hoe de gekozen oplossing zal worden getest.

## 1.2 Inhoud van technisch ontwerp

Het technisch ontwerp beschrijft de technische aspecten van het project, zoals hoe de Arduino UNO zal worden gebruikt om alles aan te sturen en op welke manier het systeem de variabele meet. In hoofdstuk 2 wordt het plan om het prototype te bouwen uitgelegd. Verder worden de voor- en nadelen benoemd en zijn de kosten gedocumenteerd.

# **2. Fysiek ontwerp**

## 2.1 Planoverzicht

Danny Esseling wil een systeem dat automatisch de plant van zijn behoefte voorziet. Ook tijdens langere periodes zonder menselijk toezicht. Hiervoor zal de projectgroep een prototype bouwen dat beschikt over een intern reservoir, waterlevel sensor, Soil moisture sensor, een waterpomp en een groeilamp.

Voor dit project zullen de volgende spullen nodig zijn om een werkend prototype te bouwen.

Onderdelen die bestelt moeten worden

|  |
| --- |
| Onderdeel |
| Arduino UNO |
| Soil moisture sensor |
| Float switch |
| Submersible Pomp |
| USB cable A to B |
| LED Kweekstrip |
|  |
| Totaal: |

Onderdelen die al aanwezig zijn in het SSC:

|  |
| --- |
| Transistor |
| 1k en 4,7k weerstand |
| LED lamp |
| Jumper kabels |
|  |
|  |
|  |
| Totaal: |

De onderdelen van de plantenpot zelf zullen worden getekend door Tristan Labruyère geprint door een 3D printer. Omdat de afdeling ICT zelf beschikt over een geschikte 3D printer zullen ze daar worden geprint. Zodra alle onderdelen binnen zijn en de onderdelen van de plantenpot geprint zijn zal alles in elkaar worden gezet door projectgroep.

## 2.2 Opsomming te verrichte activiteiten

* Ontwerpen 3D onderdelen.
* 3D ontwerp laten tekenen in CAD software bij Werktuigbouwkunde.
* 3D onderdelen printen.
* Elektronica en andere hardware bestellen.
* Elektronica solderen en andere hardware assembleren.
* Javascript code schrijven met alle functionaliteiten.
* Javascript code uploaden en testen.
* Elektronica inbouwen in het systeem.
* Elektronica testen.
* Smart plant system opleveren.

2.3 Benodigd gereedschap

* Lijmpistool
* Stanley mes
* Plastic lijm
* Soldeer ijzer

# **3. Inrichting**

In dit hoofdstuk staat beschreven welke software er geïnstalleerd zal worden en hoe de opbouw van het prototype er uit zal zien.

## 3.1 Installatieoverzicht

Voor een succesvolle testomgeving zijn de volgende onderdelen nodig.

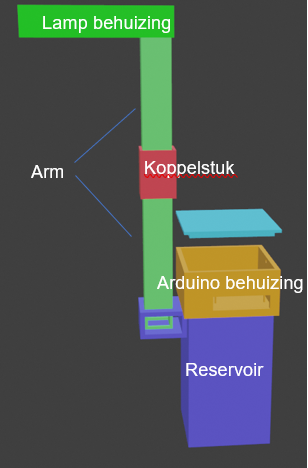
**Hardware:**

* Arduino UNO
* Soil moisture sensor
* Water pomp
* LED kweeklicht
* Water level sensor
* Montage kabel
* Jumper kabels
* LED Lampje
* Weerstand 1.0K en 4.7k
* Transistor

**Software:**

* Arduino IDE

## 3.2 Assemblage

Voor het “brein” van de plantenpot hebben we gekozen voor een Arduino UNO. Dit computerplatform is uitermate geschikt voor kleine automatisering oplossingen. Doordat de software opensource is zullen de kosten van het project lager blijven. Daarnaast is uitbreiden erg simpel en vrijwel altijd mogelijk.

De Arduino UNO zal worden gemonteerd in een 3D geprinte behuizing van waaruit de bedrading zal lopen naar de rest van de elektronica.

De 3D onderdelen kunnen worden geprint met een Laag dikte van 0.15mm.

Tijdens het printen oriënteren we de onderdelen dusdanig dat er geen of zo weinig mogelijk filler word gebruikt door de printer.

De bestanden van de 3D onderdelen staan in de [**Smart Plant System**](https://scalda365.sharepoint.com/sites/SSCIOT/Gedeelde%20%20documenten/Projecten/Projecten_Lopend/Smart%20Plant/Smart%20Plant%20Systeem) Map in de sharepoint van de Kennisgroep IOT.

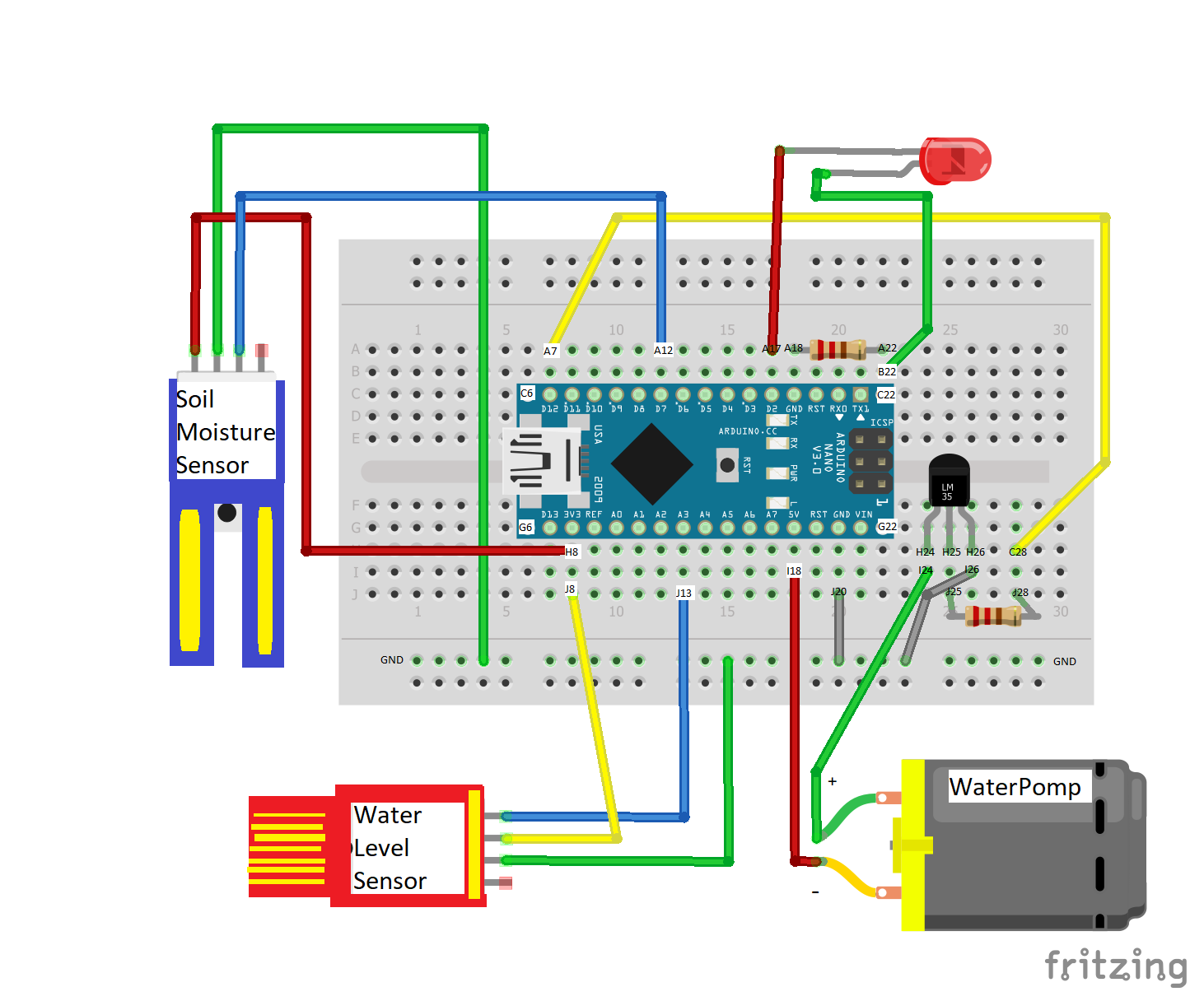
RGB LED STRIP

+

-

DIG

## 3.3 Wiring schema



4.7K

1K

Do

+

-

Do

+

-

Do = Digital Output

Di = Digital Input

# 3.4 Java-Code

De Java-Code bestaat uit verschillende code reeksen die samengevoegd zijn om tot de uiteindelijke code te komen. Ook zijn er 2 Libraries gebruikt om de functionaliteiten uit te breiden.

De uitleg naar de werking van de code laten we doen door de oorspronkelijke makers.

**De code staat in de Kennisgroep Sharepoint in de map** [**Smart Plant System**](https://scalda365.sharepoint.com/sites/SSCIOT/Gedeelde%20%20documenten/Projecten/Projecten_Lopend/Smart%20Plant/Smart%20Plant%20Systeem)

**De Bronnen voor de gebruikte code:**

De code voor het water geven en waterreservoir: <https://www.youtube.com/watch?v=T_tpKoNCVYw>

De code voor het aansturen van LED strips: <https://www.norwegiancreations.com/2018/01/programming-digital-rgb-led-strips-with-arduino-and-the-fastled-library/>

De code om Timer-loops te maken:

<https://www.forward.com.au/pfod/ArduinoProgramming/TimingDelaysInArduino.html>

**Gebruikte Libraries:**

FastLED(voor de LED aansturing code):

<http://fastled.io/>

ElapsedMillis(voor de interne Timer):

<https://playground.arduino.cc/Code/ElapsedMillis/>

**Uitleg Java-code:**

De dubbele slash “//” staat voor een comment. Een comment word door de code niet mee gelezen. Voor het leesgemak zijn alle comments //geel gekleurd.

De comments leggen uit wat de functie is van de regel of reeks.

Verschillende onderdelen van de java code worden van elkaar onderscheiden door het gebruik van { “Left curl bracket”(begin sub onderdeel) en } “Right curl bracket”(einde van sub onderdeel).

Zie bronnen en bijbehorende video’s voor verdere uitleg over de code.

# Uitleg Java-Code

\*/

// **Deze constanten worden niet veranderd, het zijn de namen voor de pins die in gebruik zijn:**

const int ledPin = 2; // Digitale output pin waar de LED aan verbonden is.

const int pumpPin = 12; // Digitale output pin waar de water pomp aan verbonden is

const int waterLevelPin = A3; // Analoge pin waar de water level sensor aan verbonden is

const int moistureSensorPin = 7; // Digitale input pin waar de grondvochtigheid mee gemeten word

const unsigned long eventInterval = 20000; //de intervallen waarop de led strip aan moet gaan

unsigned long previousTime = 0; // stel hier de actuele tijd in (voorbeeld (1 uur smiddags is 13 uur = 3600000(seconden per uur) x 13 (uur)

// These are the values to edit:

double checkInterval = 1800; //Tijdsintervallen waarop de grondvochtigheids word gecheckt - het is standaard op een half uur ingestelt = 900000

int waterLevelThreshold = 380; // Drempelwaarde waarop de LED gaat knipperen om voor laag laag water niveau te waarschuwen – verander dit op basis van de test

int emptyReservoirTimer = 90; // Hoe lang de LED gaat knipperen als de watertank bijgevuld moet worden standaard staat dit op = 30mins

int amountToPump = 300; // Hoe lang de pomp aan gaat als de plant water nodig heeft

// Global temp values

int sensorWaterLevelValue = 0; // Hier word de waarde van de waterlevel sensor tijdelijk opgeslagen

int moistureSensorValue = 0; // Hier word de waarde van de grondvochtigheids sensor tijdelijk opgeslagen

void setup() {

// de startup code die een enkele keer runt staat hier onder:

Serial.begin(9600);

pinMode(ledPin, OUTPUT);

pinMode(pumpPin, OUTPUT);

pinMode(moistureSensorPin, INPUT);

FastLED.addLeds<WS2812, DATA\_PIN, RGB>(leds, NUM\_LEDS); //Haalt de LED info op uit fastLED library

//De led knippert 5 keer om te laten zien dat hij aan staat en dat de code werkt:

for (int i=0; i <= 4; i++){

digitalWrite(ledPin, HIGH);

delay(300);

digitalWrite(ledPin, LOW);

delay(300);

}

delay(2000);

digitalWrite(ledPin, HIGH); // De LED word aan gezet

}

void loop() {

// De code die herhaaldelijk moet draaien staat hier onder:

sensorWaterLevelValue = analogRead(waterLevelPin); //Lees de waarde van de waterlevel sensor

Serial.print("Water level sensor value: "); //print het naar de monitor

Serial.println(sensorWaterLevelValue);

if (sensorWaterLevelValue < waterLevelThreshold){ //check if we need to alert you to a low water level in the tank

for (int i=0; i <= emptyReservoirTimer; i++){

digitalWrite(ledPin, LOW);

delay(1000);

digitalWrite(ledPin, HIGH);

delay(1000);

}

}

else {

digitalWrite(ledPin, HIGH);

delay(checkInterval); //wait before checking the soil moisture level

}

// check soil moisture level

moistureSensorValue = digitalRead(moistureSensorPin); //read the moisture sensor and save the value

Serial.print("Soil moisture sensor is currently: ");

Serial.print(moistureSensorValue);

Serial.println(" ('1' means soil is too dry and '0' means the soil is moist enough.)");

if (moistureSensorValue == 1){

//pulse the pump

digitalWrite(pumpPin, HIGH);

Serial.println("pump on");

delay(amountToPump); //keep pumping water

digitalWrite(pumpPin, LOW);

Serial.println("pump off");

delay(800); //delay to allow the moisture in the soil to spread through to the sensor

}

}

void loop() {

// De code die herhaaldelijk moet draaien staat hier onder:

sensorWaterLevelValue = analogRead(waterLevelPin); //Lees de waarde van de waterlevel sensor

Serial.print("Water level sensor value: "); //print het naar de monitor

Serial.println(sensorWaterLevelValue);

if (sensorWaterLevelValue < waterLevelThreshold){ //check of het waterniveau voldoende is

for (int i=0; i <= emptyReservoirTimer; i++){

digitalWrite(ledPin, LOW);

delay(1000);

digitalWrite(ledPin, HIGH);

delay(1000);

}

}

else {

digitalWrite(ledPin, HIGH);

delay(checkInterval); //zorgt voor een Delay voordat het Soil moisture level gecheckt word

}

for(int i=0; i<NUM\_LEDS; i++){ //Dit is de code waarin de RGB kleur word gedefinieerd.

leds[i] = CHSV(160, 255, 128); //Deze rij definieerd de kleur die pulseerd tijdens elke Soil moisture check

FastLED.show();

delay(100); //dit is de delay tussen elke pulse

leds[i] = CHSV(200,255,255); //Deze rij definieerd de kleur die blijft branden

FastLED.show();

}

// check Soil moisture level

moistureSensorValue = digitalRead(moistureSensorPin); //leest de Soil moisture sensor af en slaat de waarde op

Serial.print("Soil moisture sensor is currently: ");

Serial.print(moistureSensorValue);

Serial.println(" ('1' means soil is too dry and '0' means the soil is moist enough.)");

if (moistureSensorValue ==1){

//zet de pomp aan

digitalWrite(pumpPin, HIGH);

Serial.println("pump on");

delay(amountToPump); //blijf water pompen

digitalWrite(pumpPin, LOW);

Serial.println("pump off");

delay(800); //wachten op nieuwe reading van de Soil moisture sensor

}

}

Flowchart over de werking van de code:

Afbeelding met tekst

Automatisch gegenereerde beschrijving

**4. Testplan**

## 4.1 Waarom wordt er getest

Er zal worden getest of al de functionaliteiten van het Smart Plant System goed werken en op de juiste manier informatie doorgeven aan elkaar.

## 4.2 Wanneer testen

Zodra het prototype in elkaar staat zal het testen beginnen, dit kan door vertraging binnen het project omtrent de bestelling van de benodigde onderdelen tot na de review week duren.

## 4.3 Wat wordt getest

-Er zal worden getest of de hoeveelheid water en grondvochtigheid die gemeten word goed word geïnterpreteerd door de code.

-Er zal worden getest of het reservoir waterdicht is.

-Er zal worden getest of de LED lamp gaat branden bij het signaal dat het reservoir leeg is.

-Er zal worden getest of de Soil moisture sensor over langere tijdsintervallen correct blijft meten.

-Er zal worden getest of de groeilamp voldoende licht geeft en wat de optimale tijd is om de groeilamp aan te hebben staan.

-Er zal worden getest of de code naar behoren werkt.

-Er zal worden getest of de interne timers naar behoren werken

## 4.4 Waar wordt getest

We zullen de software testen tijdens werktijden in de IOT ruimte in Scalda Vlissingen.

De hardware zal worden getest tijdens werktijden in het Cisco LAB in Scalda Vlissingen.

## 4.5 Wie gaat testen

De leden van projectgroep Smart Plant Pot zullen de software en hardware gaan testen.

## 4.6 Wanneer zijn de test geslaagd?

De test zullen geslaagd zijn als het geheel in goede orde samenwerkt en de volgende componenten functioneel zijn.

**Java Code:**

-De code compiled en loopt zonder Errors

-Periodieke check of grondvochtigeheid voldoende is.

-Waterpomp activeren wanneer nodig.

-Periodieke check of er voldoende water in het reservoir zit.

-Waarschuwingslamp aan bij laag waterniveau.

-Pomp uit bij laag waterniveau

-NachtTijdTimer werkt naar behoren

-ResterendDaglichtTimer werkt naar behoren

**Hardware:**

-Groeilamp gaat aan en uit op timer

-Rerservoir is waterdicht.

-Waterpomp pompt voldoende water.

-Alle sensoren werken naar behoren

-LED lampen werken naar behoren

**5. Handleiding gebruik van Smart Plant System**

In deze handleiding worden de functionaliteiten van de plant system beschreven en hoe hij gebruikt moet worden.

Tijdens het opnieuw opstarten van de Arduino(bij verplaatsing of stroomuitval):

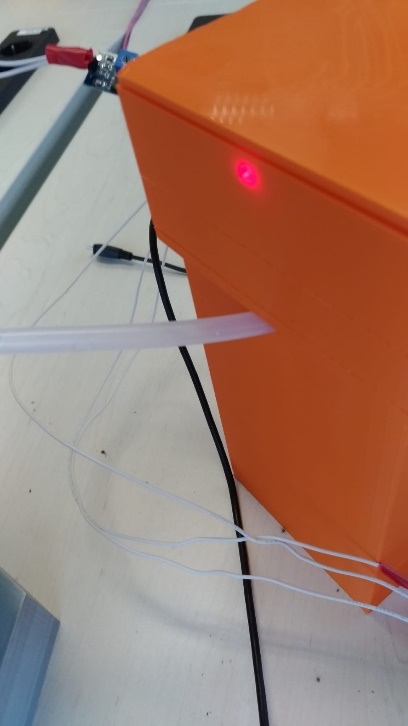
* De interne Timers van de Arduino zijn gebaseerd op een initiële input tijdens het uploaden van de code. Dus als de Arduino uit heeft gestaan zal de daglicht timer opnieuw ingesteld moeten worden.

Dus stel dat je om 2 uursmiddags de code upload. Dan moet er bij de Currenttime 14uur in miliseconden worden ingevoerd (50400000 milliseconden).

const unsigned long eventInterval = 20000; //de intervallen waarop de led strip aan moet gaan

unsigned long previousTime = 0; // stel hier de actuele tijd in (voorbeeld (1 uur smiddags is 13 uur = 3600000(seconden per uur) x 13 (uur)

Dagelijks gebruik en checks:



* Er moet dagelijks gekeken worden of het reservoir nog voldoende water bevat.

De planten systeem zal zelf aangeven wanneer er onvoldoende water aanwezig is in het reservoir door middel van het knipperende **Rode LED lampje** aan de bovenkant van de behuizing.

* Als het rode lampje knipperd OF als het vrijdag is moet het reservoir worden bijgevuld worden door de behuizing er af te halen en het reservoir bij te vullen tot onder de bovenkant van de water level sensor(Let goed op dat het water niet op de elekronica komt!).

Seizoen gebonden checks:

* De interne Timers van de Arduino zijn gebaseerd op een initiële input tijdens het uploaden van de code.

Dus stel dat je de Code upload naar de Arduino in de winter waar het rond 5 uur al donker begint te worden en de plant meer licht nodig heeft dan de zon kan geven, en je wilt dat de plant 3 uur extra licht krijgt per dag. dan moet de waarde voor de **Delay** verander worden in 10800000 miliseconden.

if (currentTime - previousTime >= eventInterval) {

for(int i=0; i<NUM\_LEDS; i++){ //Dit is de code waarin de RGB kleur word gedefinieerd.

leds[i] = CHSV(160, 255, 128); //Deze rij definieerd de kleur die pulseerd tijdens elke soil moisture check

FastLED.show();

delay(100); //dit is de delay tussen elke pulse

leds[i] = CHSV(100,100,100); //Deze rij definieerd de kleur die blijft branden

FastLED.show();

Serial.print("If statement is JA ");

}

previousTime = currentTime;

delay(10800000);

}