

Projectdossier: Energiemanagement en home automation

Integratieproject 2024-2025

Inhoud

Inleiding.....	3
Componenten.....	4
Firebeetle ESP-32-E Microcontroller	4
Raspberry Pi 4 Model B	5
IIC & UART NFC Module	6
PIR Motion Sensor	7
SCD 40 (Temperature, Humidity & CO2 Sensor)	9
Gravity: Analog AC Current Sensor (20A):.....	11
Servomotor:	13
RGB led:	14
Transformator RS-25-5:	15
Schakelaar Automatische zekering	16
Bronnen	23

Inleiding

Dit dossier beschrijft een innovatief Home Automation- en Energiemanagement-systeem dat huishoudelijke processen automatiseert en optimaliseert. Het project omvat taken zoals verlichting, ventilatie (automatisch raambeheer), temperatuurregeling, het openen van de voordeur via een NFC-module en real-time energieverbruikmonitoring. Met sensoren en actuatoren, aangestuurd door een ESP32-microcontroller en geïntegreerd met een Raspberry Pi, stuurt het systeem data naar het Blockbax-platform via Node-Red voor overzichtelijke monitoring en visualisatie.

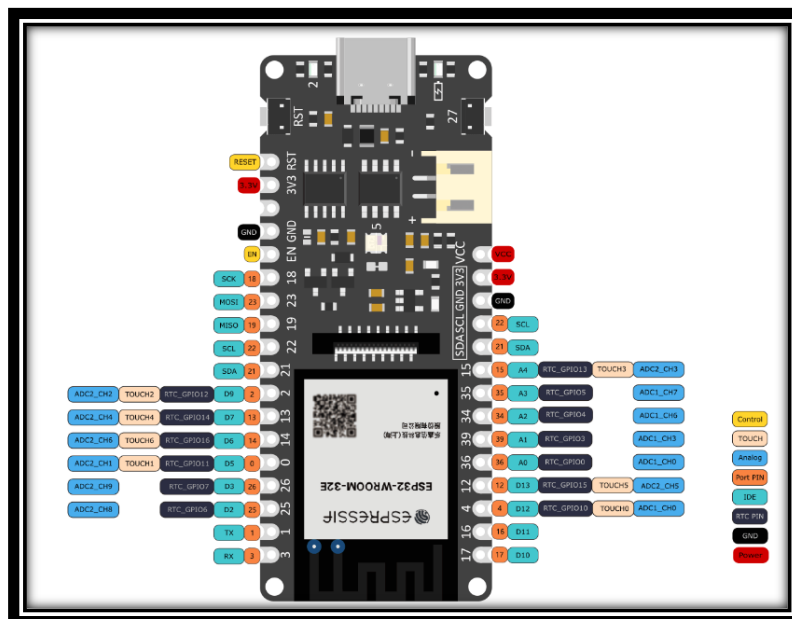
Componenten

Firebeetle ESP-32-E Microcontroller

Functie: De ESP32 dient als centrale unit in het systeem. Het zorgt voor de communicatie tussen de sensoren, actuatoren en het IoT-platform (Blockbox) via Wi-Fi.

Waarom?: Ingebouwde Wi-Fi om te connecteren en gebruik van veel documentatie (niet zoals een arduino).

Pinschema Microcontroller:



```
void setup()
{
  Wire.begin();
  Serial.begin(115200);
  delay(50);

  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(PIR, INPUT);
  pinMode(CURRENTSENSOR, INPUT);

  Serial.println("Initializing System\n");

  initSCD4X();
  initNFC();
  initWiFi();
}
```

Hier wordt er zowel setup als loop getoond, hier worden sensoren, de mqtt- en WiFiClient geïnitieerd.

```

mqttClient.setServer(RPI_ADDRESS,
1883);
mqttClient.setCallback(callback);
door.attach(D10);
nfcDelay, sensorDelay, doorTimer =
millis();
}
int val = 0;
void loop()
{
  while (!mqttClient.connected())
  {
    reconnect();
  }
  mqttClient.loop();

  if (millis() - nfcDelay >= 2000)
  {
    readNFC();
  }
  if (millis() - sensorDelay >= 5000)
  {
    readSCD4X();
    readACCurrent();
    sensorDelay = millis();
  }

  if (doorTimer + 5000 <= millis())
  {
    // Close door after 5 sec
    door.write(0);
  }

  readPIRSensor();
}

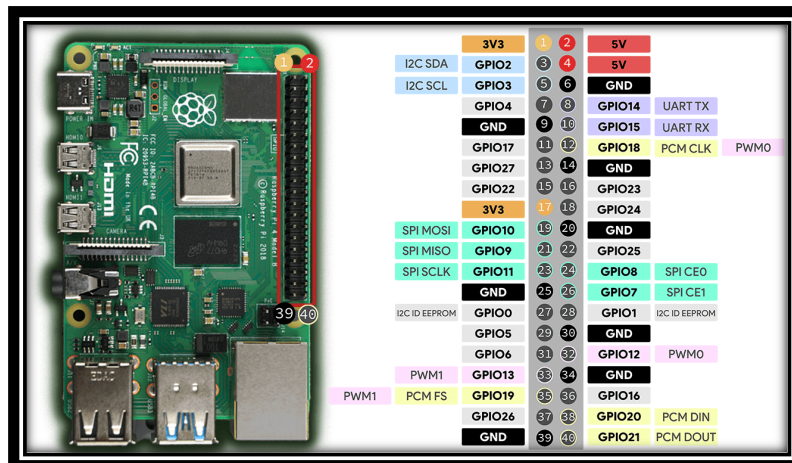
```

U kunt hieronder makkelijk Timestamps veranderen in het config.h bestand om bepaalde lezingen te vertragen/versnellen.

Raspberry Pi 4 Model B

Functie: Verwerkt de binnenkomende data via Node-RED en voert de logica uit voor acties en beslissingen.

Pinschema Raspberry Pi 4 Model B:



IIC & UART NFC Module

Functie: Leest NFC-tags.

Toepassing: Wordt gebruikt als slot voor de voordeur, waarbij enkel toegelaten tags toegang verlenen.

Waarom?: Efficiënter en veiliger dan een normale sleutel, en wordt tegenwoordig steeds vaker gebruikt en gevraagd in ons technologische “tijdperk”.

Gebruik: deze NFC gaan we gebruiken samen met de servo zodat als er gescanned wordt, dat de deur (servo) opengaat.

Code:

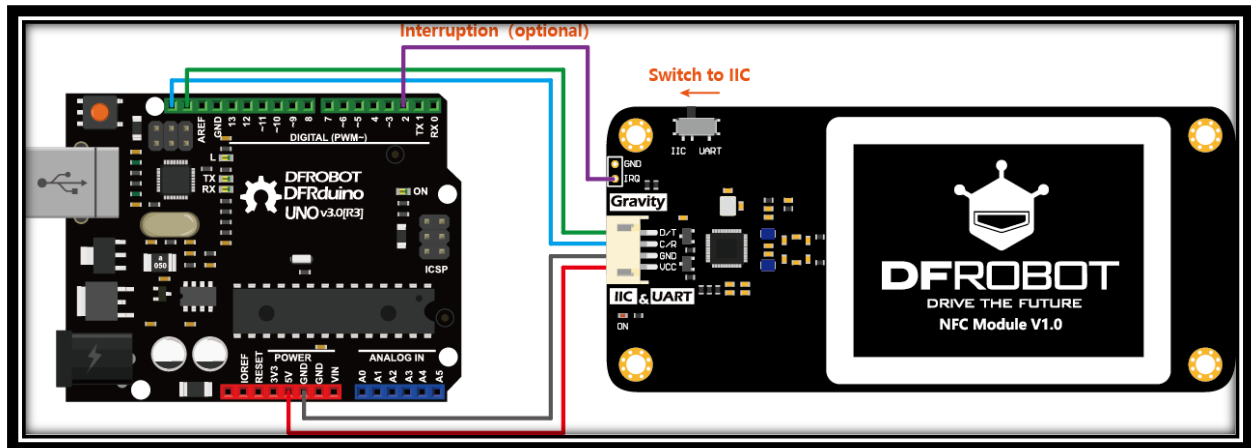
```
#define BLOCK_SIZE 16
#define PN532_IRQ 2
#define POLLING 0

void readNFC()
{
  nfc.readData(dataRead,
    READ_BLOCK_NO);
  mqttClient.publish(MQTT_NFC, (char
    *)dataRead);
}
```

BLOCK_SIZE: De grootte van elk blok data dat van de NFC-kaart wordt gelezen.
PN532_IRQ en POLLING: Hardware-instellingen voor de NFC-module.

De NFC-tag, wordt hier gelezen en doorgestuurd naar het Node-RED platform via MQTT-communicatie.

Aansluitingschema:



Library: DFRobot_PN532.h

PIR Motion Sensor

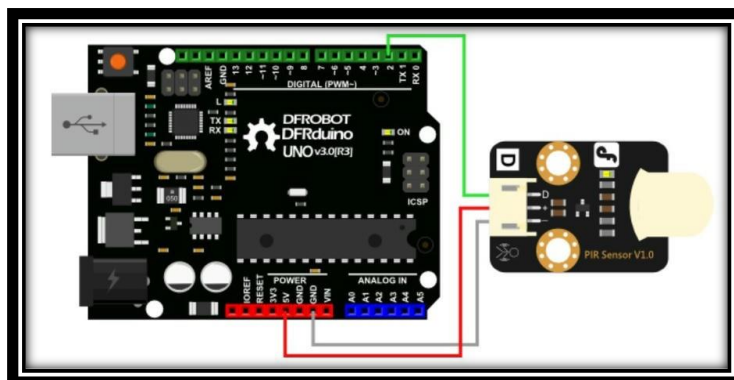
Functie: Detecteert menselijke aanwezigheid of beweging.

Toepassing: Schakelt verlichting automatisch aan en uit afhankelijk van beweging in de ruimte.

Waarom?: Wij waren normaal van plan om een Human Presence sensor te gebruiken maar was een beetje uit de scope van het project en was ook een beetje prijzig in vergelijking met de PIR sensor.

Gebruik: Wij gaan deze combineren met een LED zodat de lichten aan gaan als er beweging is.

Voorbeeld Aansluitschema:



Library: geen

SCD 40 (Temperature, Humidity & CO2 Sensor)

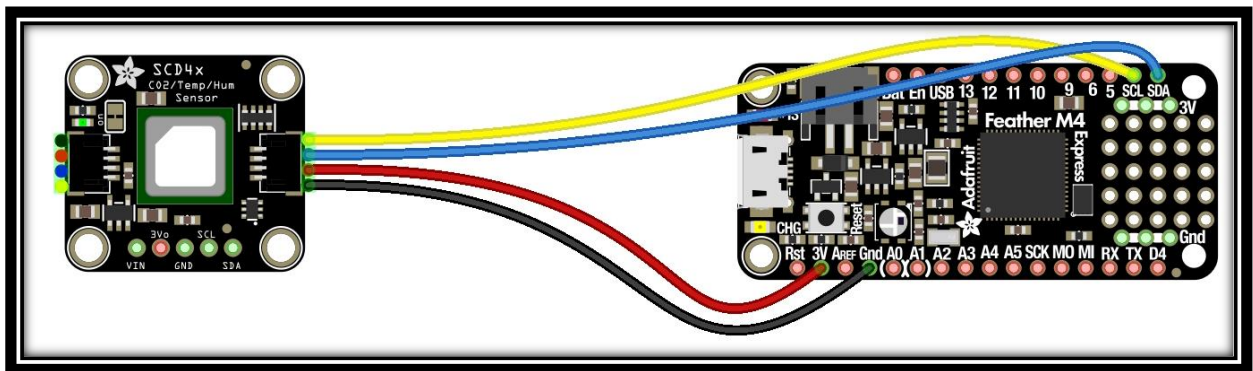
Functie: Meet CO₂-concentratie, temperatuur en luchtvochtigheid.

Interface: I2C (adres 0x62)

Waarom?: Je kan zowel Temperatuur, CO² als luchtvochtigheid meten (3 in 1), da

Gebruik: deze sensor gaan we combineren met een ander sensor zodat als bv. De temperatuur hoog is, dat het raam open gaat.

Toepassing: Regelt het verluchten of verwarmen van een kamer bij overschrijding van CO₂-waarden of temperatuurgrenzen.



Library: SensirionI2CScd4x

Code: De sensor communiceert met I2C, dus wordt aangesloten met SDA – SCL pinnen.

```
SensirionI2CScd4x scd4x;
```

```
void readSCD4X()
{
```

```
    uint16_t co2;
    float temperature;
    float humidity;
    scd4x.readMeasurement(co2,
        temperature, humidity);
```

Dit object communiceert met de SCD4X CO₂-sensor via I2C.

In deze functie worden de temperatuur-, CO₂-, en vochtigheidswaarden gelezen.

```
// Send Data to MQTT Broker  
itoa(co2, msgStr, 10);  
mqttClient.publish(MQTT_CO2,  
msgStr);  
  
itoa(temperature, msgStr, 10);  
mqttClient.publish(MQTT_TEMP,  
msgStr);  
  
itoa(humidity, msgStr, 10);  
mqttClient.publish(MQTT_HUMIDITY,  
msgStr);  
}
```

In dit stukje code worden de metingen, doorgestuurd via MQTT naar het Node-RED platform, waar de hoofdlogica plaatsbevindt.

Gravity: Analog AC Current Sensor (20A):

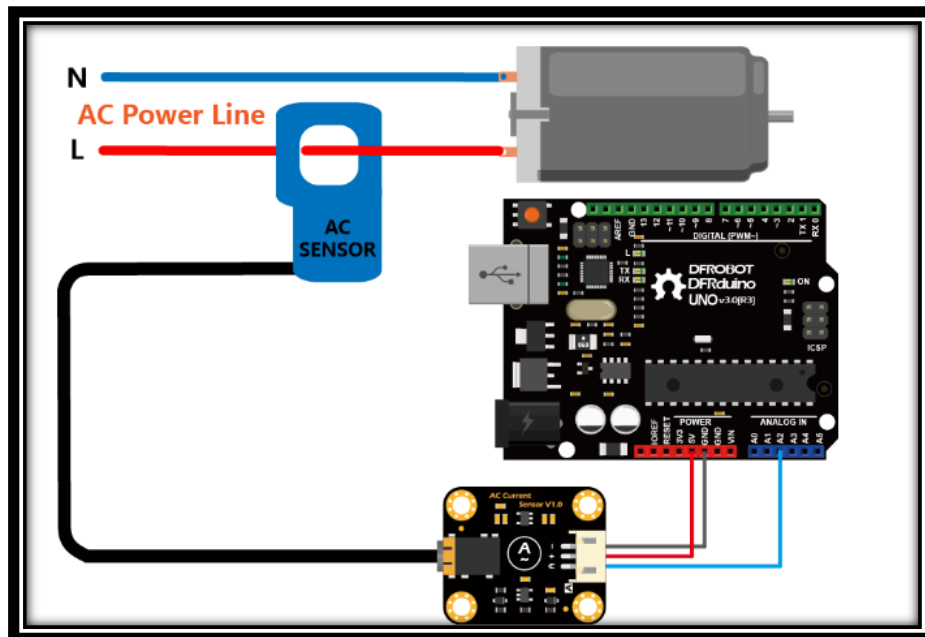
Functie: Meet de stroomsterkte in het systeem.

Toepassing: Meet het energieverbruik en geeft data door voor energiemonitoring.

Waarom?: verplicht in ons project aangezien energiemanagement aanwezig moet zijn. Daarnaast is deze makkelijk te installeren en configureren.

Gebruik: we gaan deze clippen aan de power line zodat wij stroom kunnen meten en laten weergeven op blockbax.

Aansluiting schema:



Library: geen

```
void readACCurrent()
{
  for (int i = 0; i < 5; i++) {
    peakVoltage +=
    analogRead(CURRENTSENSOR);
    delay(1);
  }
  peakVoltage = peakVoltage / 5;
```

De piekspanning wordt gemeten door meerdere keren de waarde van de stroom-sensor via AnalogRead.

```
voltageVirtualValue = peakVoltage *  
0.707;
```

```
ACCurrntnValue = voltageVirtualValue *  
ACTectionRange;
```

```
char msgStr[50];  
itoa(ACCurrntnValue, msgStr, 10);  
mqttClient.publish(MQTT_CURRENT,  
msgStr);
```

De piekspanning wordt vermenigvuldigd met 0,707 om de effectieve waarde (RMS) te berekenen. Verander deze waarde in de in de virtuele spanningswaarde als het nodig is.

De gelezen waarden, worden doorgestuurd naar het Node-RED platform via MQTT.

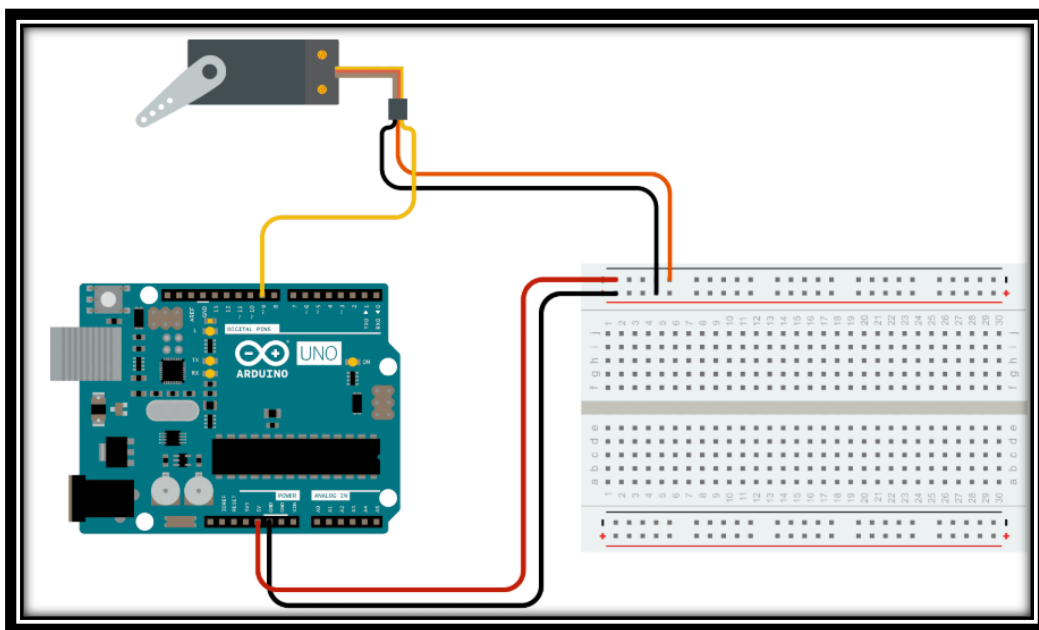
Servomotor:

Functie: Voert acties uit, zoals het openen en sluiten van het raam en ook voor de deur.

Toepassing: Opent of sluit het raam/ deur.

Waarom?: We gebruiken een servomotor voor het imiteren van een sterkere motor om het raam te openen, aangezien een servomotor in de scope van de opdracht beter past. (goedkoper, makkelijk te gebruiken)

Aansluitingschema:



Library: ESP32Servo

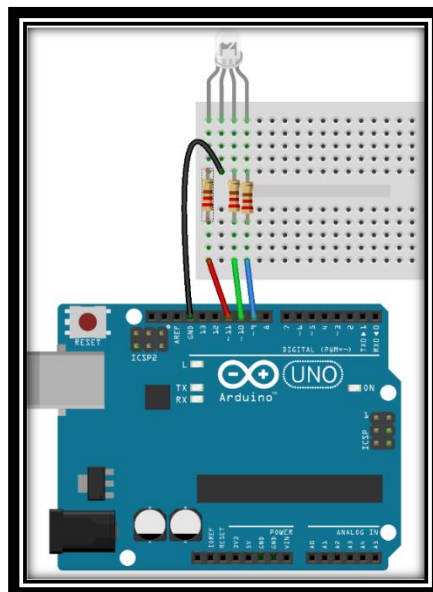
RGB led:

Functie: Rood Groen Blauw verlichting

Toepassing: Een gekleurd visueel beeld geven wanneer er iemand gedetecteerd wordt via de PIR-sensor en een signaal voor de NFC als wordt gescanned met een ander NFC.

Gebruik: We gaan deze gebruiken samen met de NFC, als er scanned met de juiste NFC, dan gaat de LED groen worden, anders wordt het rood. (Blauwe kleur gaan we niet gebruiken).

Aansluitingschema:



Code: to be continued

Hier moet gedeelte code staan voor als Pieter een ander kleur wilt.

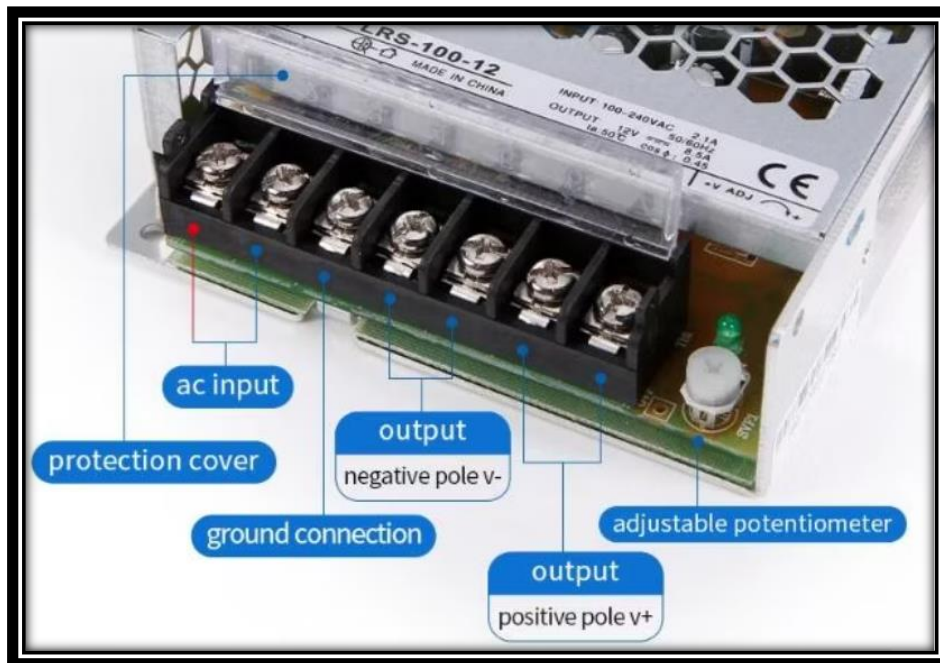
Transformator RS-25-5:

Functie: Het input voltage omzetten naar een lagere bruikbare volt.

Toepassing: Hogere voltage omzetten naar een lagere voltage om het ESP en dergelijke onderdelen te laten werken op de juiste voltage.

Waarom?: we willen niet dat onze componenten kapot gaan door overload.

Aansluitschema:



Schakelaar Automatische zekering

Functie: Het elektriciteit beveiligen.

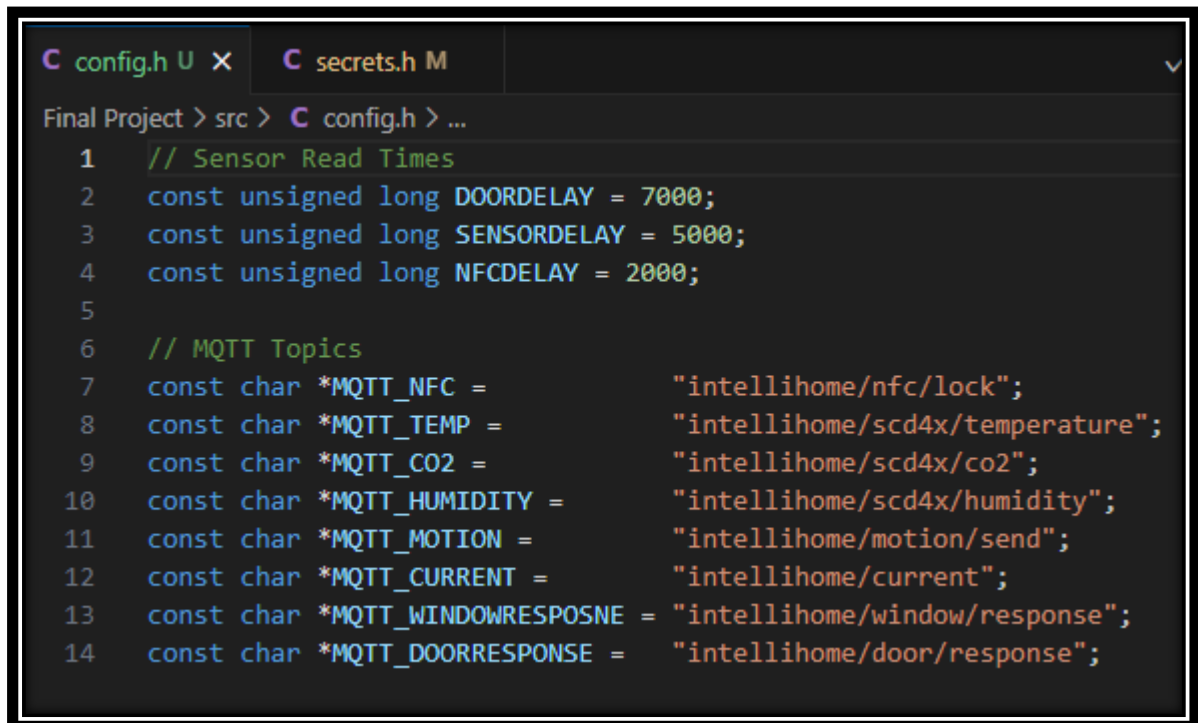
Waarom?: De automaat onderbreekt het elektrische circuit als door kortsluiting of overbelasting een te hoge stroom in de installatie ontstaat.

Toepassing: Hoofdbeveiliging van de elektriciteit dat binnenstroomt naar de transformator.



Gebruikshandleiding

ESP32 Code

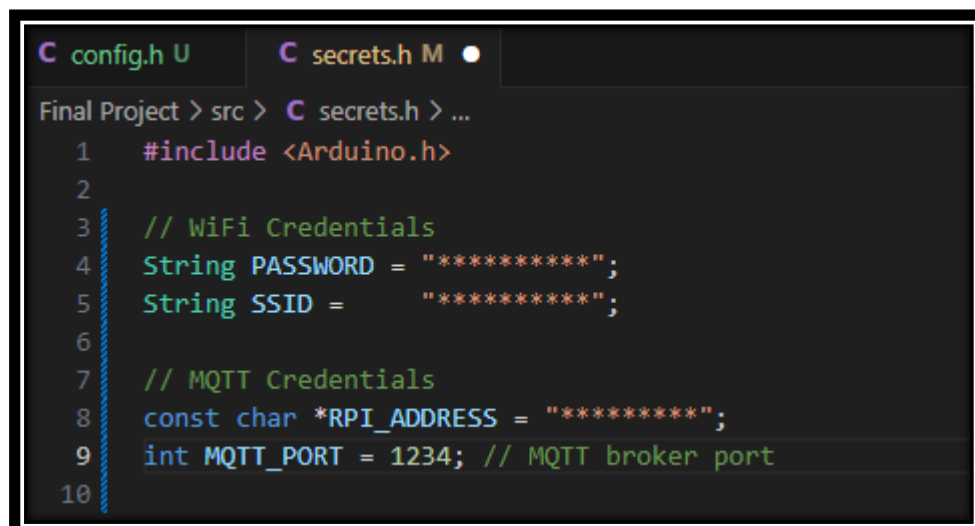


```

C config.h U x C secrets.h M
Final Project > src > C config.h > ...
1 // Sensor Read Times
2 const unsigned long DOORDELAY = 7000;
3 const unsigned long SENSORDELAY = 5000;
4 const unsigned long NFCDELAY = 2000;
5
6 // MQTT Topics
7 const char *MQTT_NFC = "intellihome/nfc/lock";
8 const char *MQTT_TEMP = "intellihome/scd4x/temperature";
9 const char *MQTT_CO2 = "intellihome/scd4x/co2";
10 const char *MQTT_HUMIDITY = "intellihome/scd4x/humidity";
11 const char *MQTT_MOTION = "intellihome/motion/send";
12 const char *MQTT_CURRENT = "intellihome/current";
13 const char *MQTT_WINDOWRESPOSNE = "intellihome/window/response";
14 const char *MQTT_DOORRESPONSE = "intellihome/door/response";

```

In het config.h bestand kan u alle sensortimings aanpassen voor moest u een sensor meer of minder vaak willen uitlezen.



```

C config.h U C secrets.h M ●
Final Project > src > C secrets.h > ...
1 #include <Arduino.h>
2
3 // WiFi Credentials
4 String PASSWORD = "*****";
5 String SSID = "*****";
6
7 // MQTT Credentials
8 const char *RPI_ADDRESS = "*****";
9 int MQTT_PORT = 1234; // MQTT broker port
10

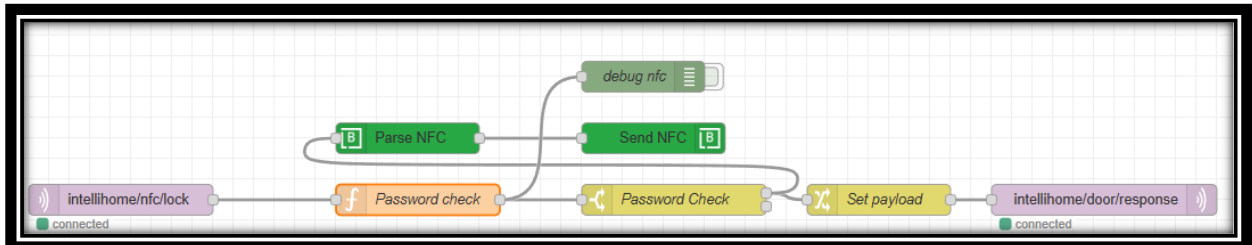
```

Vervolgens kan u in het secrets.h bestand de credentials voor de WiFi en de MQTT instellingen aanpassen naar jouw benodigdheden.

Node-RED config

Toevoegen van NFC Gebruiker

Voor het toevoegen van een gebruiker moeten we een wachtwoord en naam toevoegen aan onze Node-RED flow.



Hier drukken we vervolgens op de “Password check” node en voegen we in de 2 arrays een entry toe.

Edit function node

Delete

Cancel

Done

Properties

Name

Password check

Setup

On Start

On Message

On Stop

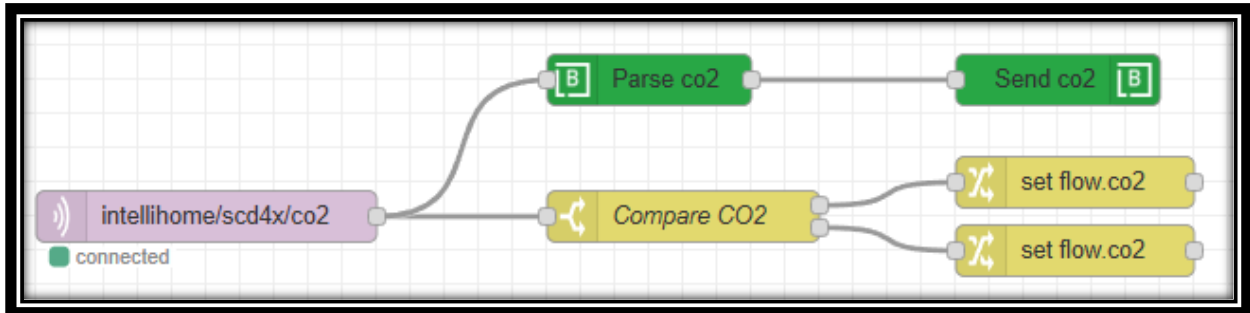
```

1  const keys = ["rDSNSLQQcLIE", "wVWwGzwySmeN", "nbRGkJGRm2jV"];
2  const fam = ["Mama", "Papa", "Kind"];
3  let passwd = msg.payload;
4
5  if(keys.includes(passwd)){
6      msg.unlocked = true;
7      let index = keys.findIndex(element => element = passwd);
8      msg.user = fam[index];
9  } else {
10     msg.unlocked = false;
11  }
12
13  return msg;

```

Drempelwaarden

Voor het aanpassen van temperatuur, vochtigheid en co2 waardes kijken we naar onderstaande flow.



Hier doen we vervolgens de “Compare CO2” node open en vervangen we de nummer door de gewenste drempelwaarde. Dit is hetzelfde voor temperatuur, vochtigheid en CO2.

Edit switch node

Delete Cancel Done

Properties

Name: Compare CO2

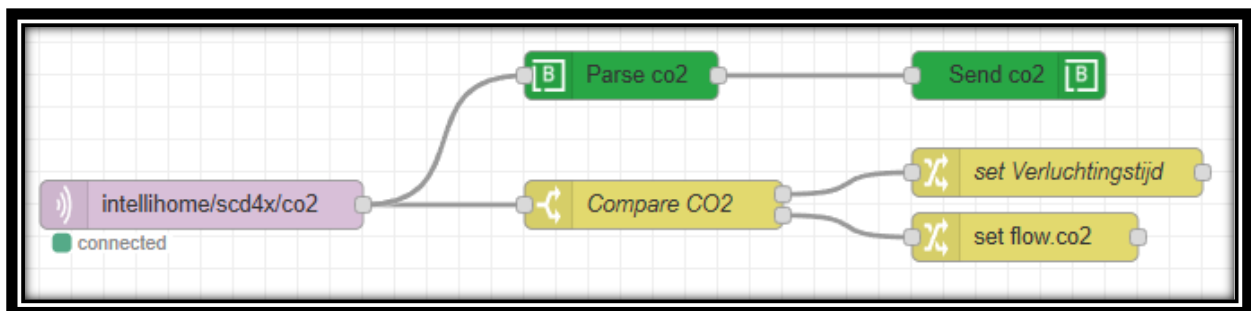
Property: msg.payload

Rule 1: \geq 2000 → 1

Rule 2: $<$ 2000 → 2

Raam verluchtingstijd

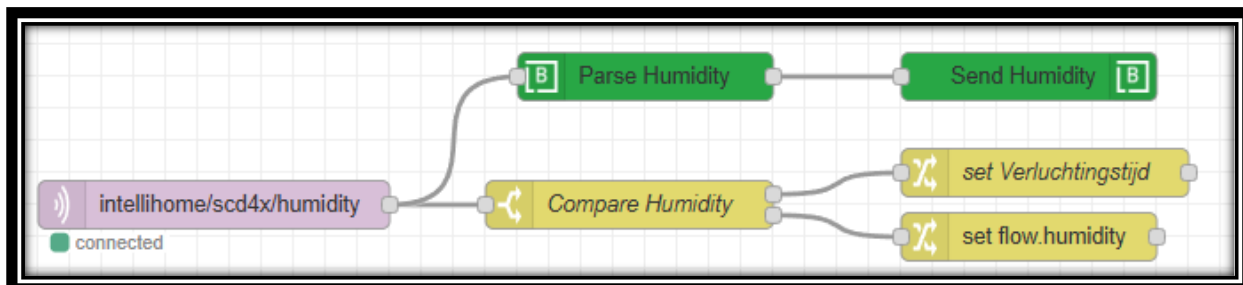
Ten slotte kijken we naar de verluchtingstijd van het raam. Om deze aan te passen naar wens kijken we naar volgende flow. Deze is steeds hetzelfde voor temperatuur, vochtigheid en co2.



Vervolgens openen we de “set Verluchtingstijd” node en passen we hier aan naar de gewenste tijd in seconden.

Blockbax configuratie

Voor het versturen van de data naar het Blockbax platform kijken we naar een van de Blockbax nodes.



Hier geven we de correcte ingestion ID, timestamp, metingwaarde en data type.

Delete
Cancel
Done

Properties

Name
Parse Humidity

☒ Use Ingestion ID

Ingestion ID
a_z kdg-test-type-groep-1\$humidity

Date
Generating Timestamp

Value
msg. payload

Data type
Data type Number

Ten slotte kiezen we naar welk project en via welke methode we deze data willen versturen.

The screenshot shows a dialog box titled "Edit blockbax-send node". At the top, there are three buttons: "Delete", "Cancel", and "Done". Below the buttons is a tab labeled "Properties" with a gear icon. To the right of the tab are three icons: a gear, a document, and a monitor. The main area of the dialog contains the following fields and options:

- Name:** A text field containing "Send Humidity".
- Project:** A dropdown menu showing "intlhome", with a pencil icon and a plus icon to its right.
- Use inbound connector:** An unchecked checkbox.
- Auto create subjects:** An unchecked checkbox.
- Type:** A dropdown menu showing "MQTT".
- Batch period:** A text field containing "1" and a dropdown menu showing "Second(s)".

Bronnen

09-01-2025:

Illustratie voorblad:

https://stock.adobe.com/be_fr/images/home-automation-icon-trendy-modern-flat-linear-vector-home-automation-icon-on-white-background-from-thin-line-smart-home-collection/237285028

ESP32-E:

https://wiki.dfrobot.com/FireBeetle_Board_ESP32_E_SKU_DFR0654

RPi B M4:

<https://randomnerdtutorials.com/raspberry-pi-pinout-gpios/>

NFC Module:

<https://wiki.dfrobot.com/Gravity:%20I2C%20&%20UART%20NFC%20Module%20SKU:%20DFR0231-H>

PIR-Module:

https://wiki.dfrobot.com/PIR_Motion_Sensor_V1.0_SKU_SEN0171

SCD40:

<https://www.adafruit.com/product/5187>

<https://learn.adafruit.com/adafruit-scd-40-and-scd-41?view=all>

Analog current meter:

https://wiki.dfrobot.com/Gravity_Analog_AC_Current_Sensor_SKU_SEN0211

servomotor:

<https://www.dfrobot.com/product-255.html>

rgb led:

<https://components101.com/diodes/rgb-led-pinout-configuration-circuit-datasheet>

automatisch zekering, en transformator:

<https://doemarkt.be/doe-het-zelf/elektriciteit/automatische-zekeringen/automatische-zekering-2p-6a>

<https://a5250379796602ad.en.made-in-china.com/product/ZsnmyplKZQrF/China-25W-RS-25-5-12V-2A-24V-5V-Single-Output-AC-DC-Switching-Power-Supply.html>

10-01-2025

kdg logo:

<https://belgianstudentleague.be/schools/karel-de-grote-hogeschool/>