**Uitleg code:**

**RaspberryPi 4:**

import time

from gpiozero import LED

import paho.mqtt.client as mqtt

from RPLCD import CharLCD

import RPi.GPIO as GPIO

from RPLCD.i2c import CharLCD

from time import sleep

import re

from typing import NamedTuple

**Hier gaan we de benodigde modules importeren om het project te doen werken.**

* Time: Hiermee gaan we tijd toevoegen zodat we delays enz mogelijk kunnen maken.
* Gpiozero: We gebruiken dit om de RaspberryPi GPIO-pinnen te bedienen.
* Paho.mqtt.client: Voor communicatie met mqtt brokers, hiermee kunnen we publiceren en subscriben.
* RPLCD: Voor de aansturing van lcd schermen op de RaspberryPi.
* CharLCD: Om een LCD met I²C aan te sturen.
* Typing: ondersteund het annoteren voor typen in python.

# LCD

lcd = CharLCD(i2c\_expander='PCF8574', address=0x27, port=1,

cols=16, rows=2, dotsize=8, #Voor lcd 16x2

charmap='A02',

auto\_linebreaks=True,

backlight\_enabled=True)

**De code initialiseert een LCD scherm met een I²C expander ven het type ‘PCF8574’ met het adres van 0x27 op de I²C bus. Het scherm is ingesteld op een 16x2 lcd maar kan aangepast worden naar keuzen. De rest zijn ingesteld op de standaard waarden.**

# WiFi-gegevens

SSID = "\*\*\*\*\*\*\*\*\*\*\*"

PASSWORD = "\*\*\*\*\*\*\*\*\*\*"

**Hier vul je de WIFI gegevens in die je gaat gebruiken.**

# MQTT-gegevens

mqtt\_server = "192.\*\*\*.\*.\*\*\*"

mqtt\_port = 1883

mqtt\_user = "\*\*\*\*\*\*\*"

mqtt\_password = "\*\*\*\*\*\*"

client\_id = "\*\*\*\*\*\*\*"

topic = "esp32/#"

topic1 = "esp32/kamer1/temp"

topic2 = "esp32/kamer2/temp"

**De MQTT-gegevens zijn voor met welke MQTT-broker we verbinding gaan maken. De poort geeft aan waar hij zal luisteren voor inkomende verbindingen.**

**De topics zijn onderwerpen waar we specifiek gaan naar luisteren.**

# LED

A1 = LED(17)

V1 = LED(27)

A2 = LED(23)

V2 = LED(24)

**De vaststelling voor welke led aan welke pin hangt.**

# Callback-functie voor het ontvangen van berichten

def on\_message(client, userdata, message):

topic = message.topic

**In de callback-functie wordt eerst het onderwerp (topic) van het ontvangen bericht opgeslagen in de variabele topic. Dit stelt de code in staat om te reageren op berichten op basis van het onderwerp waarop ze zijn ontvangen.**

lcd.clear

if topic == topic1:

lcd.cursor\_pos = (0, 0)

lcd.write\_string("Temp1: {}C".format(message.payload.decode()))

**Wanneer de topic van het binnenkomende bericht gelijk is aan topic1 die op voorhand is aangemaakt zal de lcd op de eerste rij de informatie van het binnenkomende bericht printen.**

elif topic == topic2:

lcd.cursor\_pos = (1, 0)

lcd.write\_string("Temp2: {}C".format(message.payload.decode()))

**Wanneer de topic van het binnenkomende bericht gelijk is aan topic2 die op voorhand is aangemaakt zal de lcd op de tweede rij de informatie van het binnenkomende bericht printen.**

if message.payload.decode() == "airco1-AAN":

print("A1".format(message.payload.decode()))

A1.on()

V1.off()

**Als het binnenkomende bericht gelijk is aan airco1-AAN. Zal het bericht A1 weergegeven worden op het scherm en zal led A1 aan gaan en V1 uit.**

if message.payload.decode() == "verwarming1-AAN":

print("V1")

V1.on()

A1.off()

**Als het binnenkomende bericht gelijk is aan verwarming1-AAN. Zal het bericht V1 weergegeven worden op het scherm en zal led V1 aan gaan en A1 uit.**

if message.payload.decode() == "airco2-AAN":

print("A2".format(message.payload.decode()))

A2.on()

V2.off()

**Als het binnenkomende bericht gelijk is aan airco2-AAN. Zal het bericht A2 weergegeven worden op het scherm en zal led A2 aan gaan en V2 uit.**

if message.payload.decode() == "verwarming2-AAN":

print("V2")

V2.on()

A2.off()

**Als het binnenkomende bericht gelijk is aan verwarming2-AAN. Zal het bericht V2 weergegeven worden op het scherm en zal led V2 aan gaan en A2 uit.**

# Verbinding maken met MQTT-broker

client = mqtt.Client(client\_id)

client.on\_message = on\_message

client.username\_pw\_set(mqtt\_user, mqtt\_password)

client.connect(mqtt\_server, mqtt\_port)

**Dit stuk maakt verbinding met een MQTT-broker met de MQTT client**

* client = mqtt.Client(client\_id): Hier wordt een MQTT-client geïnstantieerd met een opgegeven client-ID.
* client.on\_message = on\_message: Hier wordt de callback-functie on\_message ingesteld, zodat deze wordt opgeroepen telkens wanneer er een bericht wordt ontvangen.
* client.username\_pw\_set(mqtt\_user, mqtt\_password): Hier worden de gebruikersnaam en het wachtwoord ingesteld voor authenticatie bij de MQTT-broker.
* client.connect(mqtt\_server, mqtt\_port): Hier maakt de client verbinding met de opgegeven MQTT-broker via het opgegeven adres en poortnummer.

# Abonneren op het opgegeven topic

client.subscribe(topic1)

client.subscribe(topic2)

client.subscribe(topic)

**Hier abonneren we op onze topics.**

# Loop voor het luisteren naar berichten

client.loop\_forever()

**Hierdoor blijft de MQTT-client actief.**