Individual Project

IT essentials 1ITF

Tristan Berkmans

1ITF-8
R0784105

CAMPUS

Geel

Technology
Elektronics-ICT / Applied informatics

IT essentials

Course unit: IT essentials

Educational activity: IT essentials

First tier

Academiejaar 2020-2021

Table of Contents

1 Inleiding

Voor het vak IT-essentials kreeg ik de opdracht om een temperatuurmeter met lichtsensor te maken. Ik gebruik hiervoor het programma Visual Studio Code.

Ik stelde een stappenplan op waarin ik mijn opstelling schematisch heb weergegeven. Daarna worden de nodige materialen opgesomd. Daarna wordt er getoond hoe de kabels aangesloten dienen te worden. Vervolgens is er een blokschema met onder andere de geschreven code. In dit blokschema sluit ik de kabels aan en laat ik het lampje branden. Daarna ga ik met WIFI verbinden en laat ik de twee sensors werken. Hierna maak ik een grafiek met de temperatuur in Thingspeak en dan geef ik wat uitleg over Thingspeak. Vervolgens activeer ik Deepsleep. Dit is de ESP helemaal uitschakelen en vervolgens terug opstarten. Tenslotte doe ik nog een webdisplay met behulp van een HTML- en CSS-pagina.

Zodra alles werkte, heb ik het geheel geïntegreerd in een zelfgemaakte case. Deze hangt op in mijn game-room.

In het besluit beoordeel ik het project.

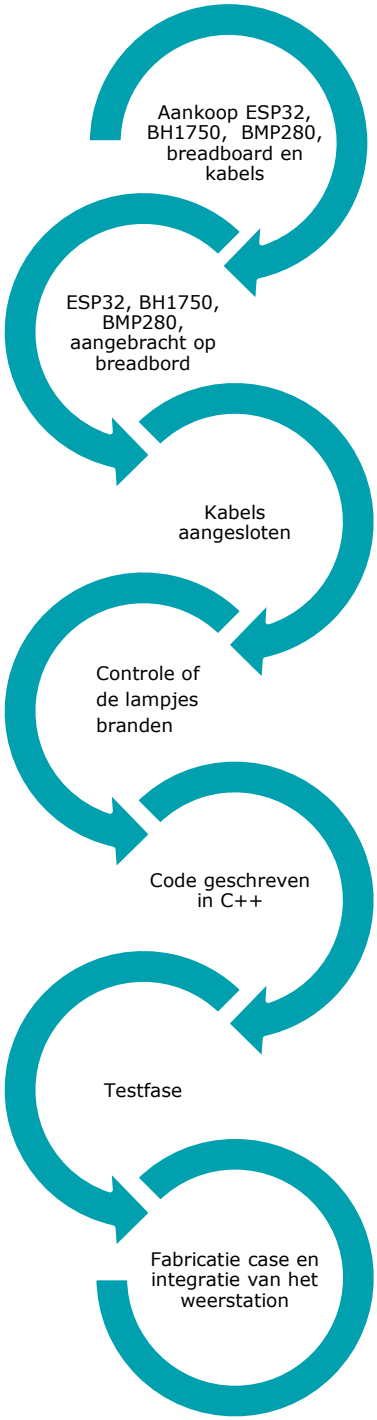
2 YouTube-link naar de video

https://youtu.be/K_4_BSnhJXM

3 ESP32

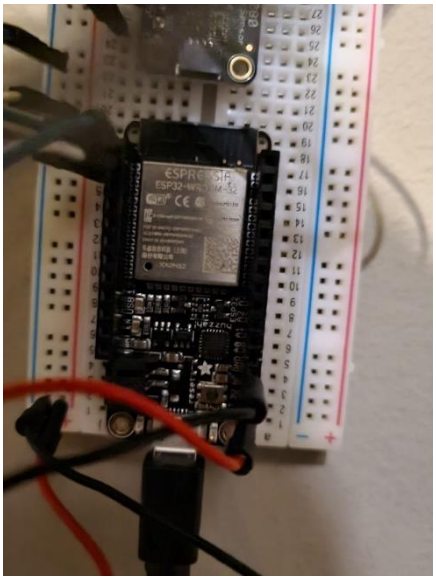
3.1 Schematische weergave van de opstelling

Voordat ik begon aan dit project, heb ik de verschillende stappen genoteerd in een schema.

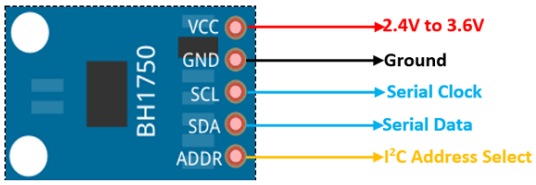


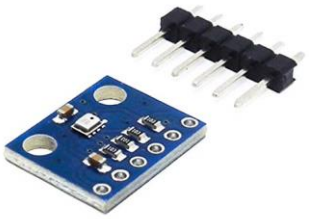
3.2 Nodige materialen

Ik heb een ESP32 gekocht op school. Dit is een budgetserie van laag-vermogen microcontrollers met geïntegreerde Wi-Fi en dual-mode Bluetooth.



Ik heb ook een BH1750 gekocht op Thomas More. Deze lichtsensor heeft een groot bereik en kan gebruikt worden om de backlight van een LCD scherm bij te stellen aan de hand van het licht dat aanwezig is.

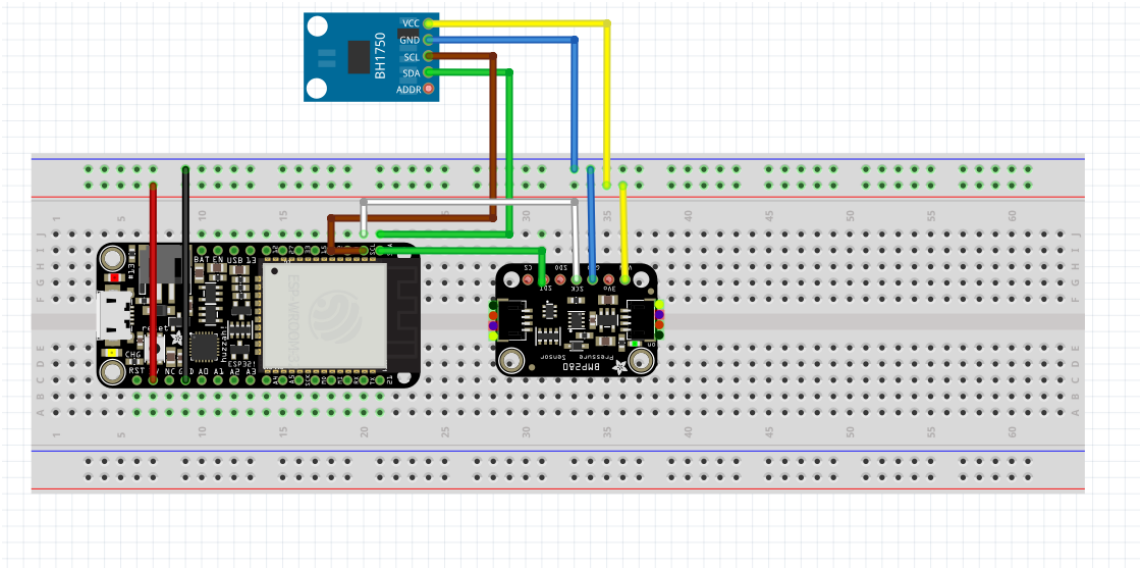




Daarnaast heb ik ook een BMP280 aangeschaft. Dit is een chip die nog zuiniger en preciezer de luchtdruk en temperatuur meet. Dankzij de ingebouwde I2C en SPI interfaces zijn de data makkelijk uit te lezen.

Verder heb ik kabels en en een breadboard aangekocht.

3.3 Montage



De ondernomen stappen zijn de volgende:

Stap 1: Ik heb ESP32, BH1750 en BMP280 op het breadboard bevestigd met de bijgeleverde pinnetjes.

Stap 2: Ik heb de GND aangesloten van de ESP32 op de min van mijn breadboard. Ik gebruikte hiervoor de ZWARTE KABEL.

Stap 3: Ik heb de 3.3 Volt aangesloten op de plus van mijn breadboard. Ik gebruikte hiervoor de RODE KABEL.

Stap 4: Daarna heb ik de BH1750 en de BMP280 verbonden met een BLAUWE KABEL naar de GND.

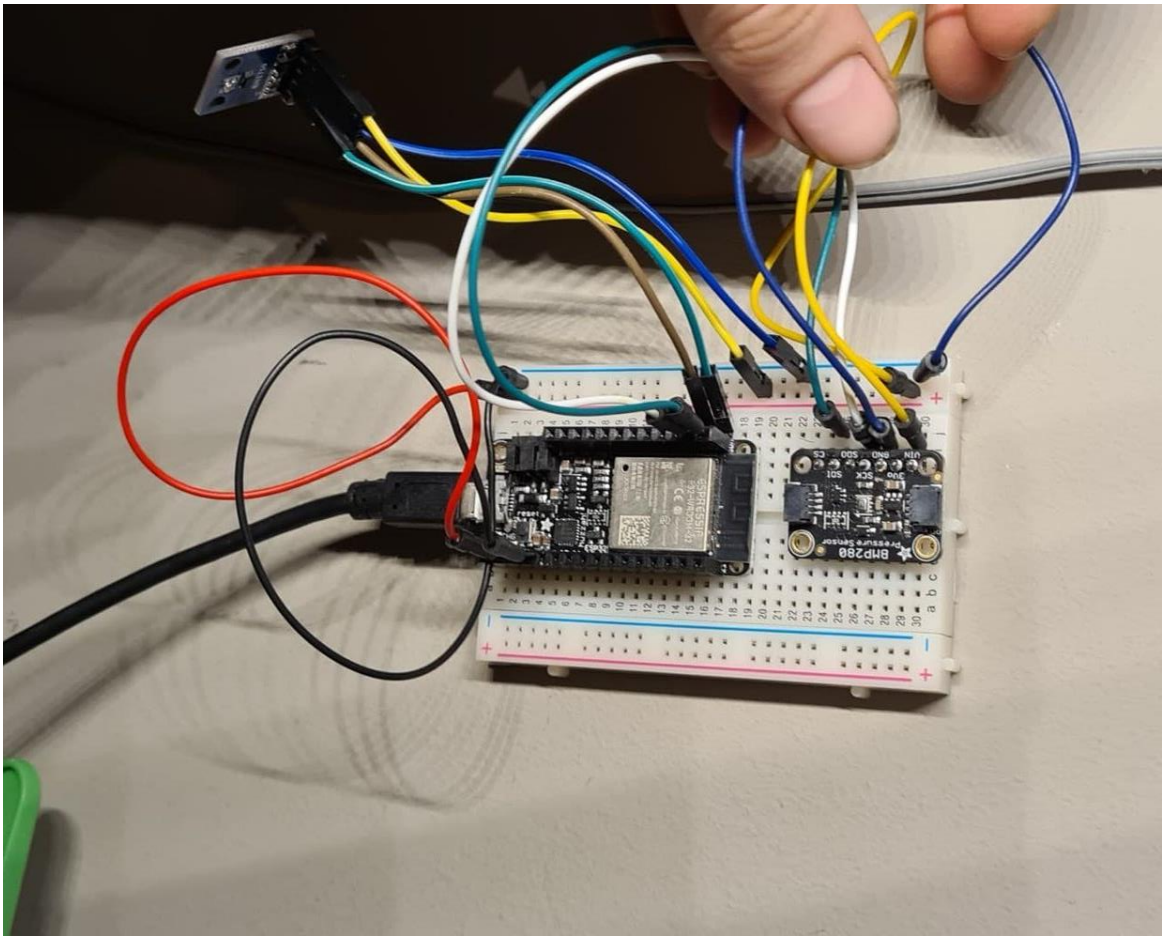
Stap 5: Vervolgens heb ik de BH1750 en de BMP280 verbonden met een GELE KABEL naar de 3.3 Volt.

Stap 6: De GROENE KABEL wordt aangesloten aan de SDA van de BH1750 en de BMP280 wordt ook met de GROENE KABEL aangesloten op SDI. Deze komen beiden van de ESP32 SDA/23.

Stap 7: De BMP280 wordt aangesloten met een WITTE KABEL vanuit de SDI en deze gaat naar SCL/22 van de ESP32.

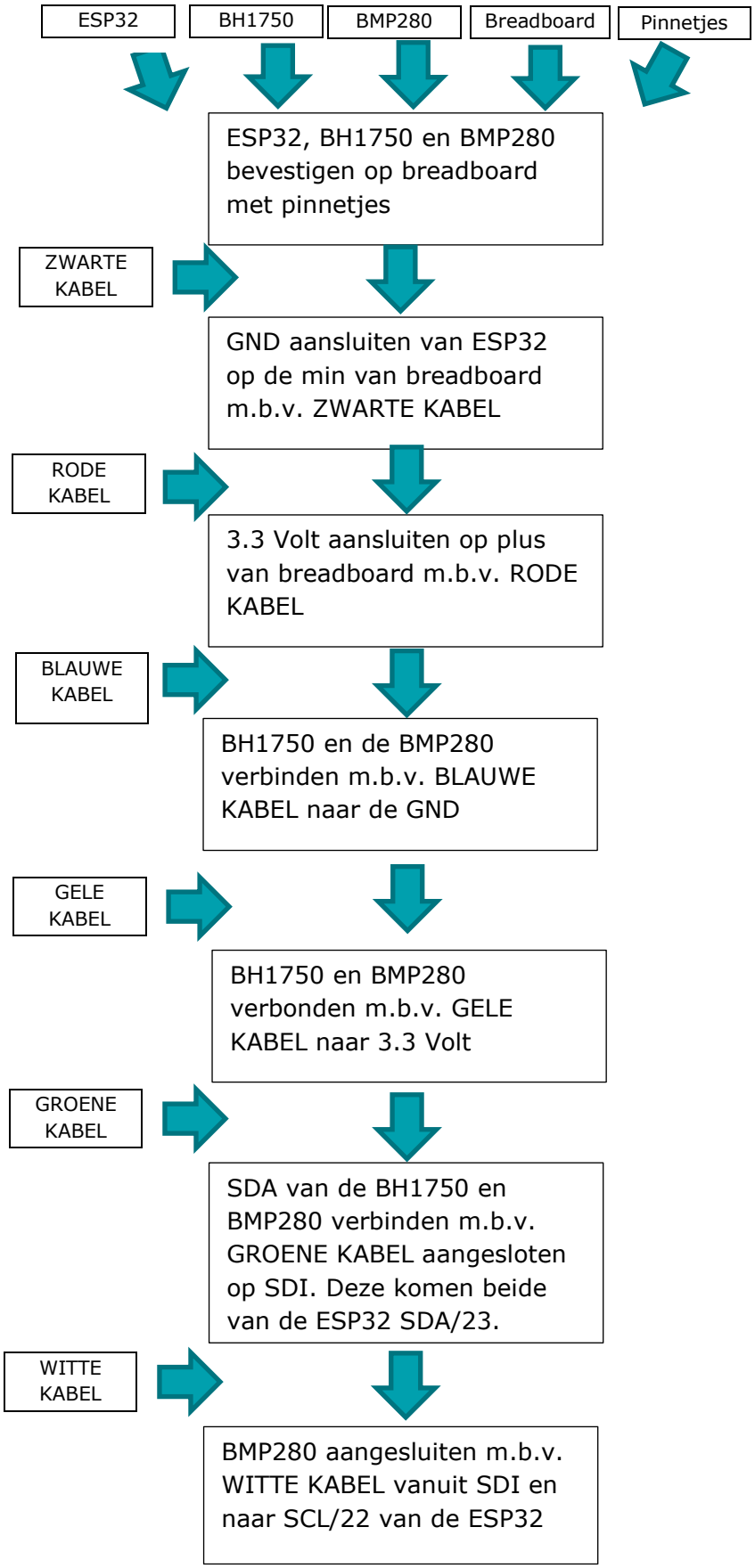
Stap 8: De BH1750 wordt aangesloten met een BRUINE KABEL vanuit de SCL en deze gaat naar SCL/22 van de ESP32.

Stap 9: Ik heb de micro-USB kabel aangesloten op de ESP32 en daarna heb ik de kabel aangesloten op de USB-poort van mijn laptop. Ik controleer of het lampje brandt. Dit is het geval. Ik concludeer daaruit dat alles goed is aangesloten.



3.4 Blokschema

3.4.1 Kabels aansluiten



BRUINE KABEL

BH1750 aansluiten m.b.v. BRUINE KABEL vanuit de SCL en naar SCL/22 van ESP32

Micro-USB kabel

Micro-USB kabel aansluiten op ESP32 en op de USB-poort van laptop.

3.4.2 Rode led

```
#include <Arduino.h>
int counter;
int delayaa;
void setup() {
  Serial.begin(9600);
  Serial.println("Seriele verbinding succes");
  pinMode(13, OUTPUT);
  counter = 0;
}

void loop() {
  if (counter % 2 == 0) {
    delayaa=500;
  }
  else{
    delayaa=2000;
  }
     digitalWrite(13, HIGH);
  delay(delayaa);
  digitalWrite(13, LOW);
  delay(delayaa);
  counter++;
  Serial.println("licht is"+String(counter));
}
```

Run code!

Brandt het rode lampje?

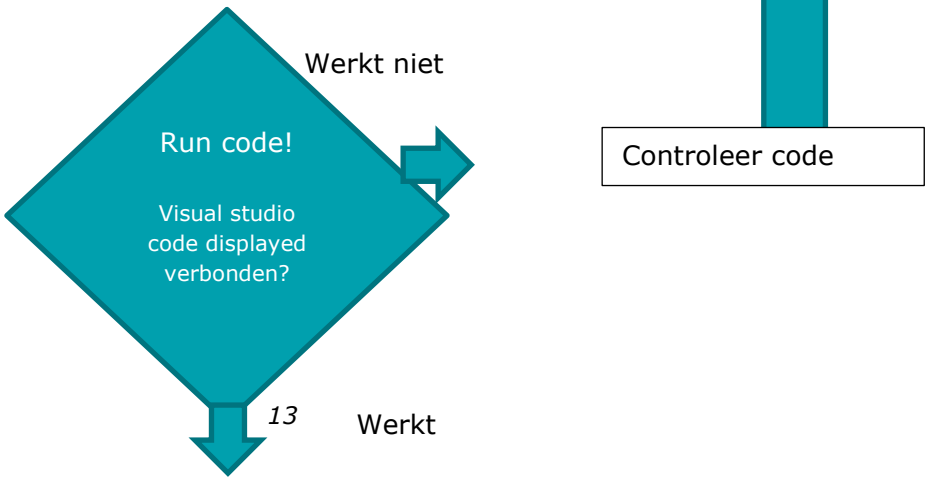
Werkt niet

Controleer code

Werkt

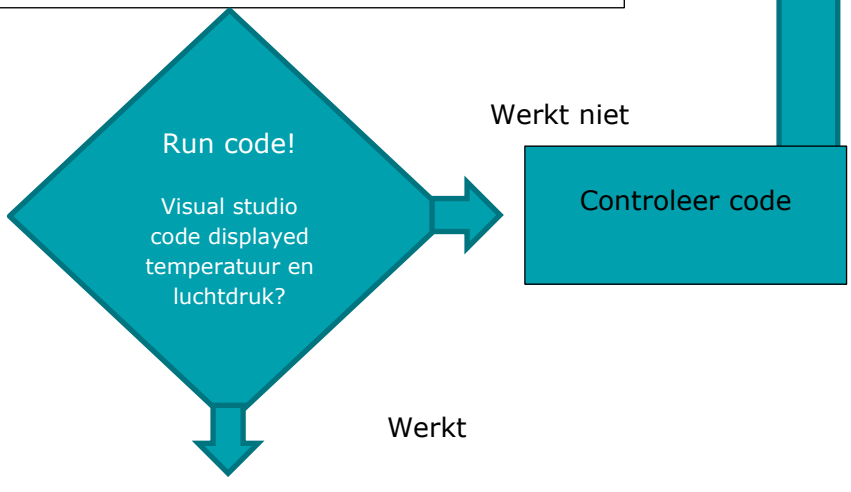
3.4.3 Met WIFI verbinden

```
#include "WiFi.h" // ESP32 WiFi include
#include "WiFiConfig.h" // My WiFi configuration.

const char *SSID = "Your SSID";
const char *WiFiPassword = "Your Password";

void ConnectToWiFi()
{

  WiFi.mode(WIFI_STA);
  WiFi.begin(SSID, WiFiPassword);
  Serial.print("Connecting to "); Serial.println(SSID);

  uint8_t i = 0;
  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.print('.');
    delay(500);

    if ((++i % 16) == 0)
    {
      Serial.println(F(" still trying to connect"));
    }
  }

  Serial.print(F("Connected. My IP address is: "));
  Serial.println(WiFi.localIP());
}


void setup()
{
  Serial.begin(9600);

  ConnectToWiFi();
}
```



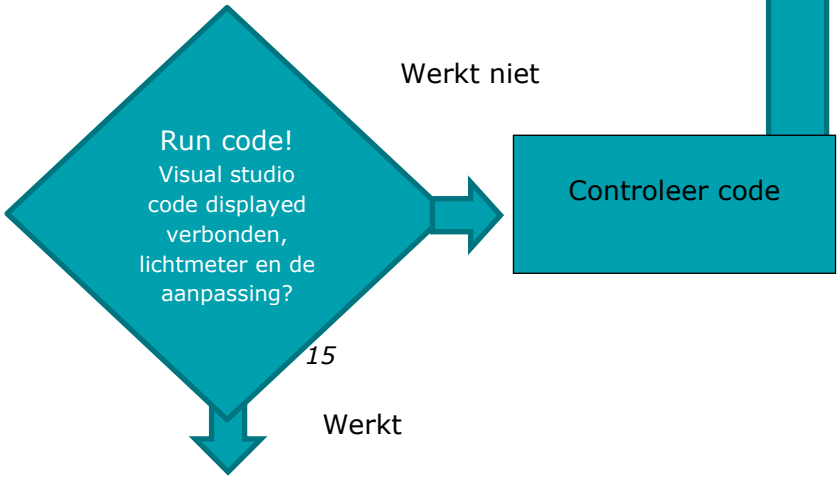
3.4.4 Temperatuur en luchtdruk BMP280

```
#include <Arduino.h>
#include <Adafruit_BMP280.h>
Adafruit_BMP280 bmp;
void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);
  Serial.begin(9600);
  Serial.println();
 if (!bmp.begin(0x77))
  {
    Serial.println("Could not find a valid BMP280 sensor");
    while (true)
    {
    }
  }

  bmp.setSampling(Adafruit_BMP280::MODE_NORMAL
                  Adafruit_BMP280::SAMPLING_X2
                  Adafruit_BMP280::SAMPLING_X16
                  Adafruit_BMP280::FILTER_X16
                  Adafruit_BMP280::STANDBY_MS_500);
void loop()
{
  digitalWrite(LED_BUILTIN, HIGH);
  String suffix = "";
  suffix += String(bmp.readTemperature());
  suffix += "°C ";
  suffix += String(bmp.readPressure() / 100);
  suffix += " hPa"
  String ssid = "#ITFactory " + suffix;
  Serial.println(""+ssid);
  digitalWrite(LED_BUILTIN, LOW);
  delay(10000);
```



3.4.5 Licht sterkte meten met de BH1750

```
#include <Wire.h>
#include <BH1750.h>
BH1750 lightMeter;
void setup(){
  Serial.begin(9600);
  Wire.begin();
  lightMeter.begin(BH1750::ONE_TIME_HIGH_RES_MODE);
  Serial.println(F("BH1750 One-Time Test"));
}
void loop() {
  float lux = lightMeter.readLightLevel(true);
  Serial.print(F("licht: "));
  Serial.print(lux);
  Serial.println(F(" lx"));
  if (lux < 0) {
    Serial.println(F("Error fout gededecteerd"));
  }
  else {
    if (lux > 40000.0) {
      //verkort de meettijd - nodig in direct zonlicht
        Serial.println(F("er is veel licht in de kamer"));
      }
    else {
        if (lux > 10.0) {
          // typische lichte omgeving
            Serial.println(F("kamer met normaal aantal licht is"));
          }
      Else {
            Serial.println(F("kamer met weinig licht in"));

}
  }
```

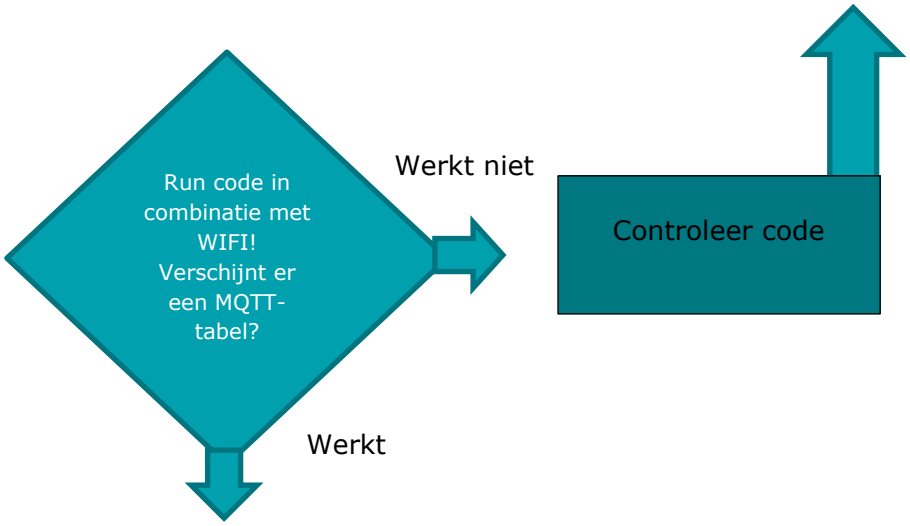


3.4.6 Thingspeak temperatuur met MQTT

3.4.6.1 code

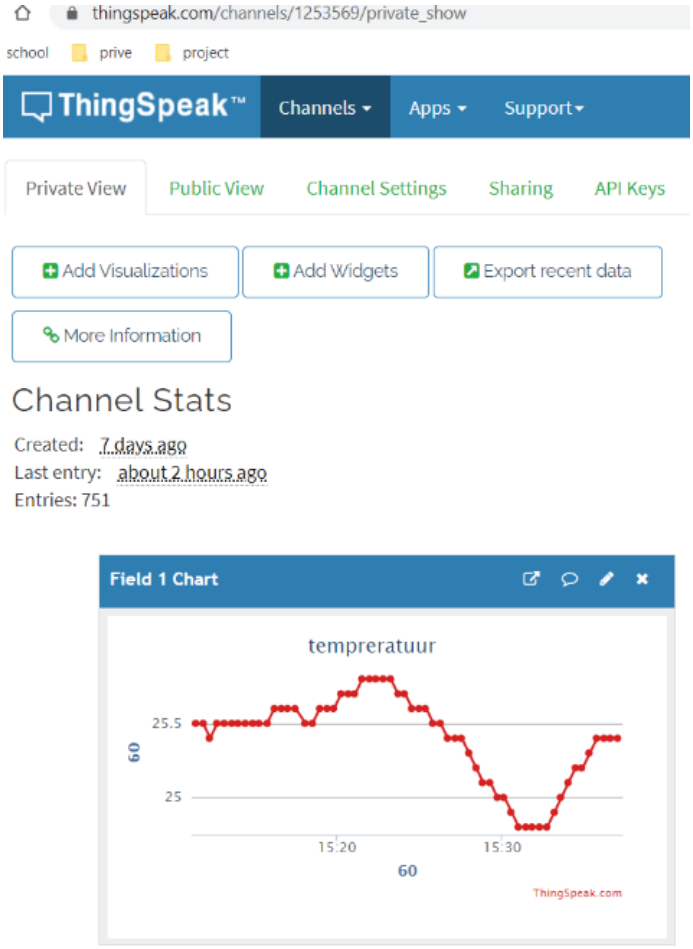
```
// gebruik vie thingspeak.com je eigen user, pwd, writeAP
I, readAIP en channelID
const char* mqtt_server ="mqtt.thingspeak.com";
const char* mqtt_user = "I5H5WIXESS7YVZ0C";
const char* mqtt_pwd = "MSZ6WC1JC4XAMMQ7";

char writeAPI[] = "RG8B5P5ALUPEN2KB";
char readAIP[] = "7BG5IG20U3ZHJE49";
long channelID = 1253569;
String OTopicStr = "channels/"+String( channelID ) +"/pub
lish/"+String(writeAPI);
const char* Otop = OTopicStr.c_str();
void reconnect() {
    // blijf in loop tot dat internet verbonden is
    while (!Client.connected()) {
      Serial.print("proberen te verbinden met MQTT...");
      // probeer te verbinden
      if (client.connect("ESP32Client",mqtt_user,mqtt_pwd
)) {
        Serial.println("verbonden");
      }
      else {
        Serial.print("mislukt, rc=");
        Serial.print(client.state());
        Serial.println("probeer achter 5 seconden");
        // wacht 5 seconden voor dat je opnieuw probeert
        delay(5000);
      }
    }
  }
void setup(){
  client.setServer(mqtt_server, 1883);
  client.setCallback(callbackMQ);
}
Loop{
  if (!client.connected()) {
    reconnect();
  }
  client.loop();

  char ctemp[7];
  String Pay= String("field1=")+ctemp;

dtostrf(temperatureesp,6,1,ctemp);
  char attri[100];
  Pay.toCharArray(attri,100);
  client.publish(Otop,attri);
delay(20000);
}
```

T



3.4.6.2 Thingspeak

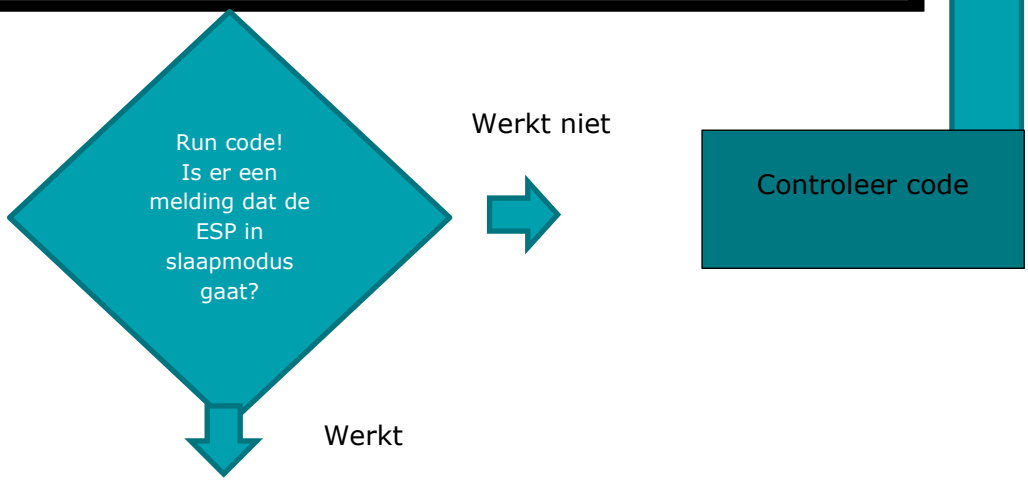
Nu nog ewven kort over thinkspeak.com in Hier aan de zij kant ziet u het Op de foto rechts staat het gewenste eindresultaat van de visualisatie. Werken met Thingspeak is zeer leerijk. Het heeft me wel enige tijd gekost om het te laten werken. Maar toen het eenmaal werkte, was ik heel erg tevreden over het resultaat!



ThingSpeak biedt rechtstreekse visualisaties van data die door de apparaten naar ThingSpeak zijn doorgegeven. Met de mogelijkheid om MATLAB®-code in ThingSpeak uit te voeren, kan men, zodra de data binnenkomen, online analyses en verwerking van de gegevens uitvoeren. ThingSpeak wordt gebruikt voor prototyping en proof of concept IoT-systemen die analyse vereisen.

3.4.7 Deepsleep

```
#include <PubSubClient.h>


#define uS_TO_S_FACTOR 1000000
#define TIME_TO_SLEEP  60
Setup{
  server.begin();
   esp_sleep_enable_timer_wakeup(TIME_TO_SLEEP * uS_TO_S_FACTOR);
  Serial.println("Setup ESP32 to sleep for every " + String(TIME_TO_SLEEP) + " Seco
nds");
  delay(60000);
  Serial.println("Going to sleep now");
  delay(1000);
  Serial.flush();
  esp_deep_sleep_start();
  Serial.println("This message will never be printed");
}
void print_wakeup_reason(){
  esp_sleep_wakeup_cause_t wakeup_reason;
  wakeup_reason = esp_sleep_get_wakeup_cause();
  switch(wakeup_reason)
  {
    case ESP_SLEEP_WAKEUP_TIMER : Serial.println("Wakeup caused by timer"); break;
```

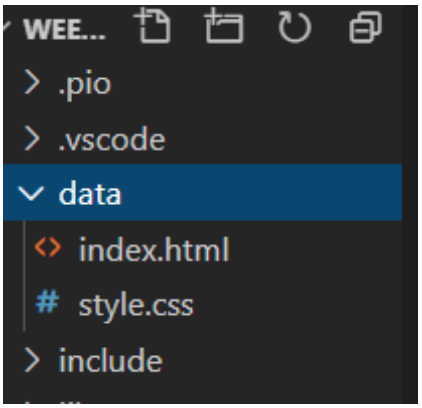


3.4.8 Web display

3.4.8.1 code

```
#include <PubSubClient.h>
#include "WiFi.h"
#include "ESPAsyncWebServer.h"


String processor(const String& var){
  Serial.println(var);

  if (var == "LIGHT") {
    return String(lightMeter.readLightLevel());
  }

  if (var == "TEMPERATURE") {
    return String(bmp.readTemperature());
  }

  if (var == "PRESSURE") {
    return String(bmp.readPressure()/100);
  }
  return String();

}

server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send(SPIFFS, "/index.html", String(), false, processor);
  });

  // Route voor het laden style.css file
  server.on("/style.css", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send(SPIFFS, "/style.css", "text/css");
  });

  // Route om het lampje te zetten naar hoog
  server.on("/on", HTTP_GET, [](AsyncWebServerRequest *request){
    digitalWrite(ledPin, HIGH);
    request->send(SPIFFS, "/index.html", String(), false, processor);
  });

  // Route om het lampje naar laag te zetten
  server.on("/off", HTTP_GET, [](AsyncWebServerRequest *request){
    digitalWrite(ledPin, LOW);
    request->send(SPIFFS, "/index.html", String(), false, processor);
  });
```

3.4.8.2 HTML page



- Maak een map aan en noem deze data.
- Daarna maak je twee nieuwe bestanden aan index.hml en style.css.
- De html-code kan je zelf kiezen.
- De CSS-code kan je dit ook zelf kiezen.
- In terminal moet je dit runnen “pio run -t uploadfs”

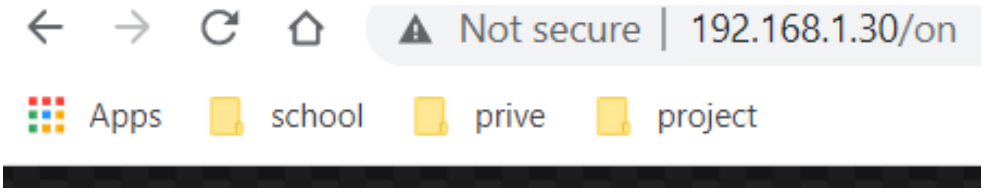
```
<!DOCTYPE html>
<html>

<head>
    <title>ESP32 Web Server</title>
    <meta naam=" viewport" content=" Width=devide-width, initial-scale1">
    <link rel="icon" href="data:,">
    <link rel="stylesheet" type="text/css" href="style.css">
</head>

<body>
    <div>
        <h1>ESP 32 Web display</h1>
        <p>temperatuur is %TEMPERATURE% graden celsuis</p>
        <p>luchtdruk is %PRESSURE% hpa</p>
        <p>Lichtsterkte is %LIGHT% lux</p>
        <p><a href="/on"><button class="knop">rood lampje aan!</button></a></p>
        <p><a href="/off"><button class="knop knop2">rood lampje uit!</button></a></p>
    </div>
</body>

</html>
```

3.4.8.3 CSS page

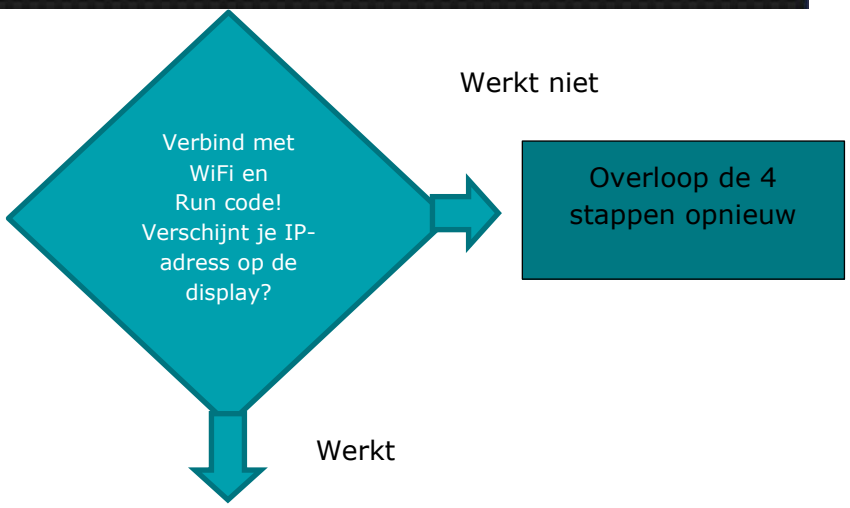
```
html {font-family: Verdana, Geneva, Tahoma, sans-serif;
    display: inline-block;
    margin: auto;
    text-align: center;
}body {background-size: 1000px;
    background:
        linear-gradient(27deg, #151515 5px, transparent 5px) 0 5px,
        linear-gradient(207deg, #151515 5px, transparent 5px) 10px 0px,
        linear-gradient(27deg, #222 5px, transparent 5px) 0px 10px,
        linear-gradient(207deg, #222 5px, transparent 5px) 10px 5px,
        linear-gradient(90deg, #1b1b1b 10px, transparent 10px),
        linear-gradient(#1d1d1d 25%, #1a1a1a 25%, #1a1a1a 50%, transparent 50%, transparent 75%, #242424 75%, #242424);
    background-color: #131313;
    background-size: 20px 20px;
}h1 {color: white;
    padding: auto;
    text-transform: uppercase;
}p {font-size: 2rem;
    color: white;
    font-weight: bold;
    text-transform: uppercase;
}div {min-width: 300px;
    background-color: #fff;
    background:
        radial-gradient(hsl(0, 100%, 27%) 4%, hsl(0, 100%, 18%) 9%, hsla(0, 100%, 20%, 0) 9%) 0 0,
        radial-gradient(hsl(0, 100%, 27%) 4%, hsl(0, 100%, 18%) 8%, hsla(0, 100%, 20%, 0) 10%) 50px 50px,
        radial-gradient(hsla(0, 100%, 30%, 0.8) 20%, hsla(0, 100%, 20%, 0)) 50px 0,
        radial-gradient(hsla(0, 100%, 30%, 0.8) 20%, hsla(0, 100%, 20%, 0)) 0 50px,
        radial-gradient(hsla(0, 100%, 20%, 1) 35%, hsla(0, 100%, 20%, 0) 60%) 50px 0,
        radial-gradient(hsla(0, 100%, 20%, 1) 35%, hsla(0, 100%, 20%, 0) 60%) 100px 50px,
        radial-gradient(hsla(0, 100%, 15%, 0.7), hsla(0, 100%, 20%, 0)) 0 0,
        radial-gradient(hsla(0, 100%, 15%, 0.7), hsla(0, 100%, 20%, 0)) 50px 50px,
        linear-gradient(45deg, hsla(0, 100%, 20%, 0) 49%, hsla(0, 100%, 0%, 1) 50%, hsla(0, 100%, 20%, 0) 70%) 0 0,
        linear-gradient(-45deg, hsla(0, 100%, 20%, 0) 49%, hsla(0, 100%, 0%, 1) 50%, hsla(0, 100%, 20%, 0) 70%) 0 0;
    background-color: #300;
    background-size: 100px 100px;
    padding: 20px;
    margin: 20px;
}.knop {display: flex;
    background-color: red;
    border: none;
    border-radius: 4px;
    color: white;
    padding: 16px 40px;
    font-size: 30px;
    margin: auto;
    cursor: pointer;
    text-decoration: none;
}.knop:hover {background-color: orange;
    color: dimgrey;}
.knop2 {background-color: lightblue;}
```

3.4.8.4 Display met IP

Nu kan je via je IP-adres van de ESP je webdisplay zien.



Hieronder zie je een voorbeeld van mijn webdisplay.

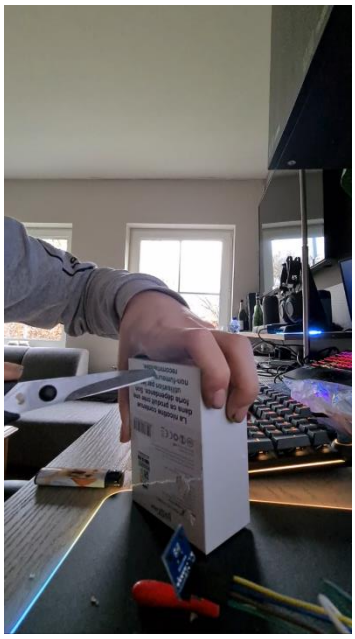


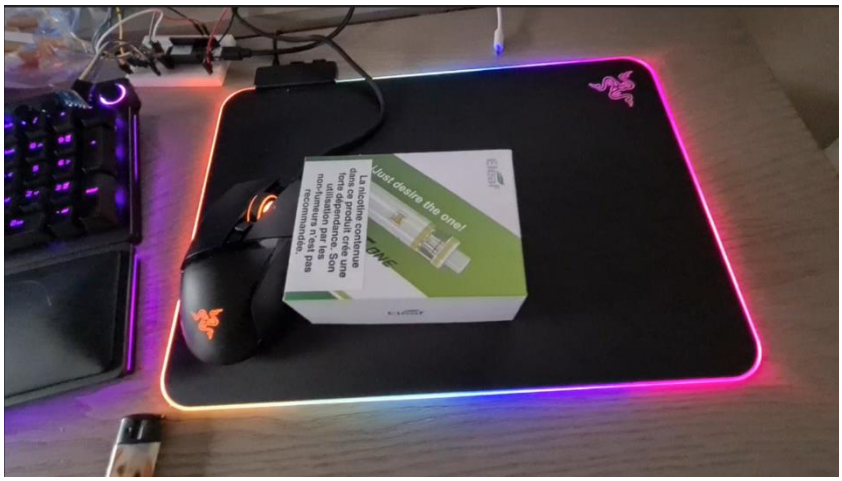
Voeg alle codes samen en je hebt het hele project.

3.5 Fabricatie en integratie van het weerstation

3.5.1 Case

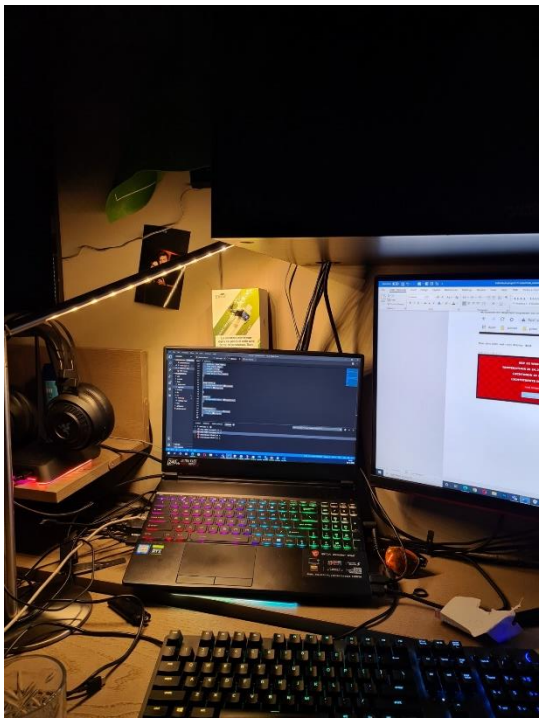
Mijn case heb ik gemaakt uit mijn vape-doosje dat ik een aantal jaren geleden gekocht heb. Ik heb hier mijn ESP32 in verwerkt en ik heb erop gelet dat mijn lichtsensor nog altijd licht krijgt.





3.5.2 Integratie

Ik heb mijn case vastgemaakt aan de muur in mijn game room en aangesloten. Zodat ik altijd weet hoe warm het is van zodra ik mijn computer aanzet. Ik ga verder experiementeren met de ESP32. Ik heb een gevoel dat er nog veel meer mogelijkheden zijn.



4 Besluit

4.1 Project beschrijving

Ik vind het een heel leerrijk project en het leukste vind ik het aansluiten van de componenten. Ik heb er me de laatste drie weken verdiept in het onderwerp. Ik heb veel bijgeleerd zoals werken met C++ en Visual Studio code. Daarnaast heb ik regelmatig opzoekwerk vericht in verband met codes. Natuurlijk liep niet alles meteen van een leien dakje en heb ik ook helemaal vast gezeten. Ik ben van mening dat dit project netjes en naukeurig afgehandeld is en ik heb er een goed gevoel bij.

4.2 Project evaluatie

Programma	***werkt***
Temperatuur sensor	**ja**
Licht sensor	**ja**
Webdisplay van ESP (temperatuur, druk, lichtsterkte)	**ja**
Lampje aan en uit (Webdisplay)	**ja**
Verbinden met mqtt (Thingspeak)	**ja**
Temperatuur tabel (thingspeak)	**ja**
Tabel van lichtsterkte en druk (Thingspeak)	**Nee**