

Eindopdracht week 7: freestyle IoT project

Koolmonoxide melder



Project beschrijving

We hebben de opdracht gekregen om zelf een projectje te ontwerpen met onze eigen hardware en de blynk app die we aan de praat hebben gekregen in het vak Sensoren en Interfacing. Dus ben ik direct al mijn sensor afgegaan om te kijken welke sensoren mij het interessantste leken en ben ik op een paar ideetjes gekomen. Uiteindelijk heb ik gekozen om mijn koolmonoxide sensor te gebruiken voor een koolmonoxide melder van te maken. Ik heb de waardes van deze sensor omgevormd en laat deze weergegeven op de LCD display en mijn blynk app. Om de max waarde van koolstofmonoxide aan te passen gebruik ik lokaal een potentiometer maar kan deze apart ook instellen via een slider op de blynk app. Als de max waardes van de potentiometer of de slider op de blynk app wordt overschreden door de koolmonoxide sensor zal er een rood ledje beginnen branden tezamen met een ledje op de app. Ook heb ik via een andere leuke functie in mijn programma er voor gezorgd dat ik een melding krijgen op mijn gsm. Deze toepassing kan gebruikt worden in ieder huis dat centrale verwarmingsketel bevat.

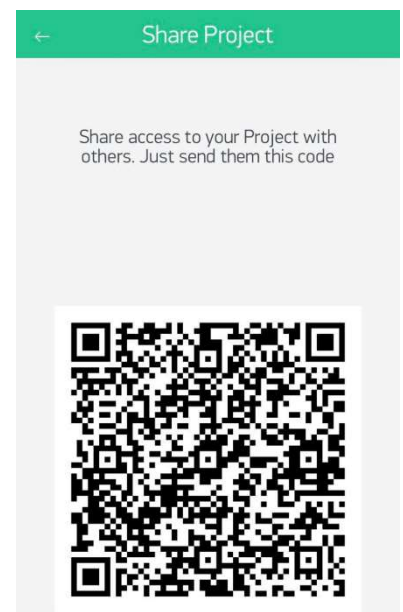
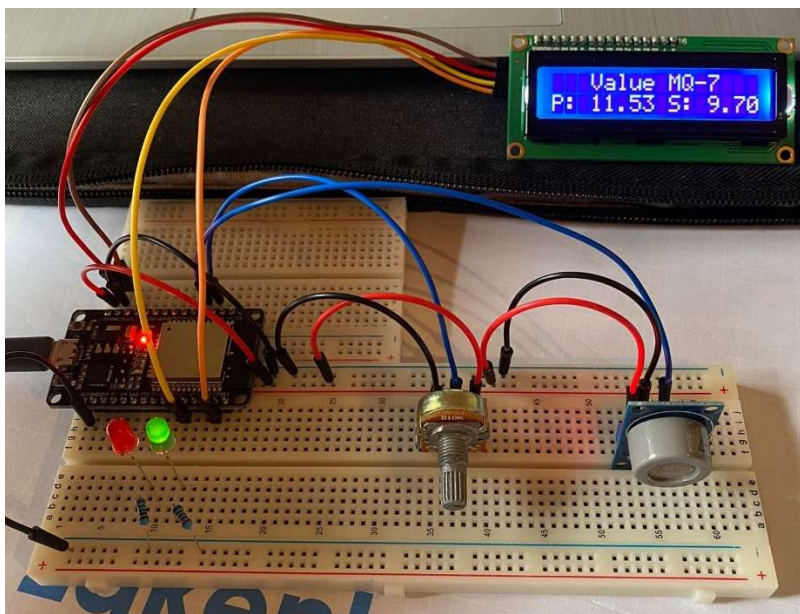
Waarom de koolmonoxide melder?

Jaarlijks overlijden in België gemiddeld 10 tot 15 personen door koolmonoxidevergiftiging. Enkele honderden mensen belanden in het ziekenhuis. Een koolmonoxidemelder kan je op tijd waarschuwen **en daarmee je leven redden!**



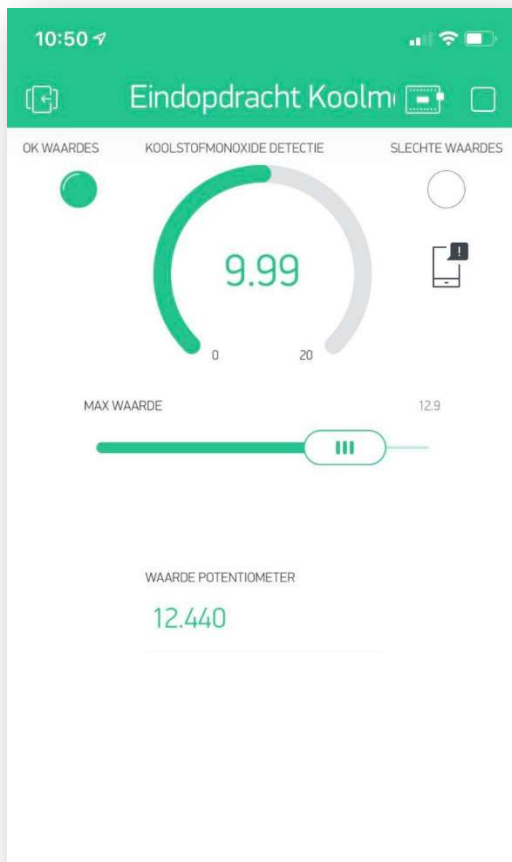
Link naar mijn GitHub pagina --> <https://github.com/SenneOpDeBeeck> <--

Link naar mijn Youtube demonstratie video --> <https://youtu.be/DgAC7FkSgdU> <--

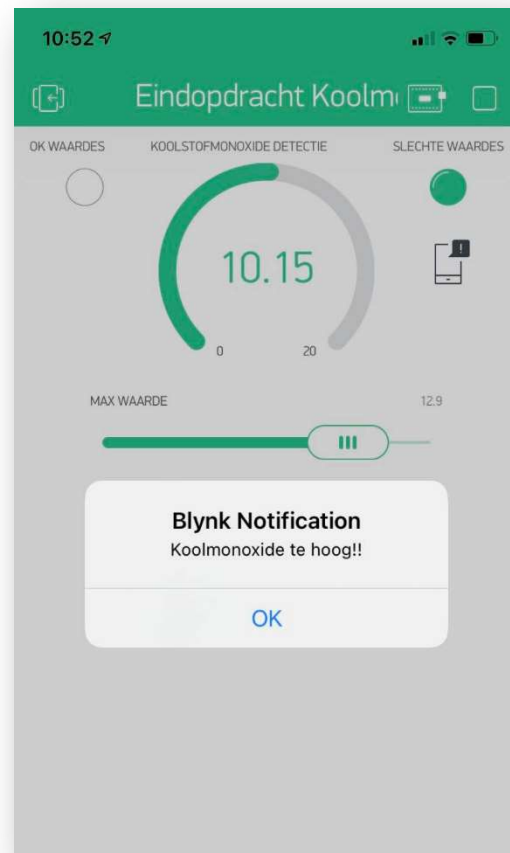


Display van de Blynk app:

Goede waarden



Slechte waarden



Arduino ide code

```
//////////////////////////////////// Library //////////////////////////////////////
#include <Wire.h>                                // Library
#include <LiquidCrystal_I2C.h>                    // Library
LiquidCrystal_I2C lcd(0x27, 16, 2);              // Library

#define BLYNK_PRINT Serial                       // Verbinding met blynk
#include <WiFi.h>                                 // Wifi instellingen
#include <WiFiClient.h>                           // Wifi instellingen
#include <BlynkSimpleEsp32.h>                     // Verbinding met blynk

char auth[] = "N8DhZE6ErggllhC59IX44Okpi-fz_85-"; // Code voor Blynk app
```

```
char ssid[] = "telenet-0BA24";           // Wifi instellingen
char pass[] = "aWdYfkF68ane";           // Wifi instellingen

#define LedR 18                           // Rode Led
#define LedG 19                           // Groene Led

WidgetLED led1(V1);                       // Led op blynk
WidgetLED led2(V2);                       // Led op blynk

int gewensteWaarde;                       // Memory voor blynk slider
float Val = 0;                            // Memory voor blynk slider

BlynkTimer timer;                        // Blynk functie

void sendSensor()
{
    float MQ7_volt;                       // Sensor MQ-7 waarde
    float MQ7_Value;                     // Sensor MQ-7 waarde

    float potVolt;                       // Potentiometer waarde
    float potWaarde;                     // Potentiometer waarde

    float potValue = analogRead(34);      // Potentiometer waardes uitlezen
    potVolt = map(potValue, 300, 4095, 500, 1500); // Potentiometer waardes aanpassen
    potWaarde = potVolt / 100;            // Potentiometer waardes aanpassen

    MQ7_Value = analogRead(35);           // Sensor MQ-7 waardes uitlezen
    MQ7_volt = MQ7_Value / 100;           // Sensor MQ-7 waardes omvormen

    Serial.print("sensor_MQ7 = ");        // Weergeeft op de serial print: "sensor_volt = "
```

```
Serial.println(MQ7_volt);           // Geeft de waarde van sensor mq-7 op de serial
print                               //

Serial.print("Potentiometer value = "); // Weergeeft op de serial print:
"Potentiometer value = "

Serial.println(potWaarde);          // Geeft de waarde van de potentiometer op de
serial print

lcd.clear();                        // LCD display leegmaken

lcd.setCursor(3, 0);                // LCD display van sensor MQ-7

lcd.print("Value MQ-7");            // LCD display van sensor MQ-7

lcd.setCursor(0, 1);                // LCD display van de potentiometer

lcd.print("P: ");                  // LCD display van de potentiometer

lcd.setCursor(03, 1);               // LCD display van de potentiometer

lcd.print(potWaarde);               // LCD display van de potentiometer


lcd.setCursor(9, 1);                // LCD display van sensor MQ-7

lcd.print("S: ");                  // LCD display van sensor MQ-7

lcd.setCursor(11, 1);               // LCD display van sensor MQ-7

lcd.print(MQ7_volt);                // LCD display van sensor MQ-7


Blynk.virtualWrite(V7, MQ7_volt);   // Geeft de waarde van de MQ7 sensor op de
blynk app                            //

Blynk.virtualWrite(V8, potWaarde);   // Geeft de waarde van de potentiometer op
de blynk app                          //

delay(1000);                        // Delay tussen de regels


if (MQ7_volt >= potWaarde || MQ7_volt >= Val ) // Wanneer de ingestelde waarde word
overschreden door sensor MQ-7 gaat het alarm af.

{

digitalWrite(LedG, LOW);            // Groene led wordt uitgezet

digitalWrite(LedR, HIGH);           // Rode led wordt aangezet
```

```
Blynk.notify("Koolmonoxide te hoog!! ");          // Blynk notificatie
led1.off();                                       // Led op blynk
led2.on();                                       // Led op blynk

} else
{
    digitalWrite(LedR, LOW);                     // Rode led wordt aangezet
    digitalWrite(LedG, HIGH);                     // Groene led wordt aangezet
    led1.on();                                    // Led op blynk
    led2.off();                                   // Led op blynk
}
}

#define gewensteWaarde V5                        // Virtual pin5 voor de slider

BLYNK_WRITE(gewensteWaarde)                     // Functie voor blynk slider
{
    int gt = param.asInt();                       // blynk slider uitlezen
    Val = gt;                                     // blynk slider uitlezen
    Serial.print("Gewenste waarde ingesteld op: "); // blynk slider uitlezen
    Serial.println(gt);                           // blynk slider uitlezen
}

//////////////////////////////////// Void Setup //////////////////////////////////////

void setup()
{
    Serial.begin(115200);                         // Verbinding maken met de seriële monitor
    Serial.print("Connecting to ");               // Verbinding maken met de wifi
    Serial.println(ssid);                         // Verbinding maken met de wifi
```

```
WiFi.begin(ssid, pass); // Verbinding maken met de wifi
int wifi_ctr = 0; // Verbinding maken met de wifi
while (WiFi.status() != WL_CONNECTED) { // Verbinding maken met de wifi
    delay(500); // Delay tussen de regels
    Serial.print("."); // Verbinding maken met de wifi
}
Serial.println("WiFi connected"); // Verbinding maken met de wifi
Blynk.begin(auth, ssid, pass, "server.wyns.it", 8081); // Verbinding maken met Blynk

Stippelijn(); // Toont een stippenlijn op de seriële monitor door de
fuctie "stippelijn"

Serial.println("Eindopdracht: KoolmonoxideMelder op Blynk!!"); // Weergeeft op de serial print:
"Eindopdracht: MultiMelder"

Stippelijn(); // Toont een stippenlijn op de seriële monitor door de
fuctie "stippelijn"

delay(1000); // Delay tussen de regels

Serial.println("**Sensoren zijn aan het opwarmen*"); // Weergeeft op de seriële monitor:
"**Sensoren zijn aan het opwarmen*"

delay(1000); // Delay tussen de regels

lcd.begin(); // Lcd display start scherm
lcd.backlight(); // Lcd display start scherm
lcd.setCursor(1, 0); // Lcd display start scherm
lcd.print("Welcome to my"); // Lcd display start scherm
lcd.setCursor(2, 1); // Lcd display start scherm
lcd.print("End project"); // Lcd display start scherm
delay(10000); // Lcd display start scherm
lcd.clear(); // Lcd display start scherm
lcd.setCursor(0, 0); // Lcd display start scherm
lcd.print("...Preparing..."); // Lcd display start scherm
lcd.setCursor(2, 1); // Lcd display start scherm
lcd.print("Wait please"); // Lcd display start scherm
```

```
delay(5000);           // Lcd display start scherm  
lcd.clear();           // Lcd display start scherm
```

```
timer.setInterval(1000L, sendSensor);           // Sensor waardes doorsturen naar blynk
```

```
pinMode(LedR, OUTPUT);           // Aansluiting Rode led
pinMode(LedG, OUTPUT);           // Aansluiting Groene
```

}

```
//////////////////////////////////// Void Loop //////////////////////////////////////
```

```
void loop()
{
  Blynk.run();           // Blynk app starten
  timer.run();           // Acties uitvoeren op de blynk app
}
```

//////////////////////////////////// Functie //////////////////////////////////////

[illegible]