

Bouw je eigen CO2 meter: deel 2

In deel 2 worden indicatie ledjes voor de lucht kwaliteit toegevoegd en worden de meetwaarden over een korte periode gemiddeld om een rustigere weergave te bereiken.

De CO2 waarden

De CO2 meter meet de hoeveelheid koolstofdioxide in de lucht en dit wordt uitgedrukt in ppm (parts per million).

Een waarde van ongeveer 400 ppm wordt in de buitenlucht gemeten. In huis en gebouwen zijn door menselijke activiteit de waarden hoger. Algemeen wordt gesteld dat CO2 waarden van 0 – 800 ppm veilige waarden zijn. Indien de CO2 meter een waarde aangeeft van 800 – 1200 ppm wil dit zeggen dat extra ventilatie vereist is om de situatie naar een goed niveau te brengen. Als bovengrens voor een toelaatbare waarde voor CO2 gehalte in een binnenruimte wordt 1200 ppm gesteld. Bij CO2 waarden van 1200 ppm en hoger is het dringend nodig om maatregelen te treffen.

We gaan onze CO2 meter met 3 ledjes (groen, geel en rood) uitbreiden om de verschillende grenswaarden van de lucht kwaliteit aan te geven.

- **Groen** bij minder dan 800 ppm CO2, de ruimte is goed geventileerd
- **Geel** tussen de 800 en 1200 ppm CO2, ventileren is gewenst
- **Rood** bij meer dan 1200 ppm CO2, ventileren is noodzakelijk.

Sketch met ledjes uitbreiding: Sketch CO2 meting_02

```
/*  
Sketch CO2_meting_02 versie 1.0  
Meting van CO2, temperatuur en vochtigheid met sensor SCD30  
De weergave is op een 4 x 20 I2C LCD en op de seriële monitor  
LEDjes toegevoegd voor grenswaarden CO2  
Het oorspronkelijk programma is gemaakt door Nathan Seidle van SparkFun Electronics  
Aangepast voor Coderdojo  
*/  
  
#include <Wire.h>           // laad de I2C bibliotheek  
#include <LiquidCrystal_I2C.h> // bibliotheek voor het LCD scherm  
#include "SparkFun_SCD30_Arduino_Library.h" // laad de SCD30 bibliotheek  
  
LiquidCrystal_I2C lcd(0x27, 20, 4); // start LCD scherm en lcd is gebruiksnaam  
SCD30 Sensor;           // het SCD30 board krijgt gebruiksnaam Sensor  
  
float temperatuur, vochtigheid; // variabelen  
int CO2waarde;  
int index;  
int ledPins[] = {4, 5, 6};  
  
byte graden[8] = {           // graden teken
```

```

B00010,
B00101,
B00101,
B00010,
B00000,
B00000,
B00000,
B00000
};

void emptyline()          // lege lijn routine
{
  lcd.print ("          ");
}

void setup() {
  for (index = 0; index <= 3; index++)
  {
    pinMode(ledPins[index], OUTPUT);
  }
  Serial.begin(115200);
  Serial.println("SCD30 programma met LCD uitlezing");
  Wire.begin();

  if (Sensor.begin() == false)
  {
    Serial.println("Sensor niet gevonden.....");
    while (1);
  }

  lcd.init();           // initialiseer het LCD scherm
  lcd.backlight();      // zet de backlight aan
  lcd.createChar(0, graden); // het symbool graden staat nu op positie 0
  lcd.clear();          // wis het scherm
  lcd.setCursor(0, 0);  // zet de cursor op positie 1, regel 0
  lcd.print (" CO2-Temp-Vocht ");
}
//De SCD30 geeft data om de 2 seconden

void loop() {
  if (Sensor.dataAvailable())
  {
    CO2waarde = Sensor.getCO2();
    lcd.setCursor(0, 1); // zet de cursor op positie 1, regel 1
    emptyline();
    lcd.setCursor(0, 1); // zet de cursor op positie 1, regel 1
    lcd.print ("CO2 :"); lcd.print(CO2waarde); lcd.print (" ppm");
    if (CO2waarde >= 1200)
    {
      digitalWrite(ledPins[0], HIGH); // GEVAAR
      digitalWrite(ledPins[1], LOW);
      digitalWrite(ledPins[2], LOW);
    }
  }
}

```

```

}
else if (CO2waarde >= 900)
{
  digitalWrite(ledPins[0], LOW);
  digitalWrite(ledPins[1], HIGH); //opletten
  digitalWrite(ledPins[2], LOW);
}
else
{
  digitalWrite(ledPins[0], LOW);
  digitalWrite(ledPins[1], LOW);
  digitalWrite(ledPins[2], HIGH); //veilig
}
Serial.print("co2 waarde(ppm):");
Serial.print(CO2waarde);

temperatuur = Sensor.getTemperature();
lcd.setCursor(0, 2); // zet de cursor op positie 1, regel 2
emptyline();
lcd.setCursor(0, 2); // zet de cursor op positie 1, regel 2
lcd.print ("Temp : "); lcd.print(temperatuur, 1); lcd.write(0); lcd.print ("C");
Serial.print(" temp(C):");
Serial.print(temperatuur, 1);

vochtigheid = Sensor.getHumidity();
lcd.setCursor(0, 3); // zet de cursor op positie 1, regel 3
emptyline();
lcd.setCursor(0, 3); // zet de cursor op positie 1, regel 3
lcd.print ("Vocht: "); lcd.print(vochtigheid, 1); lcd.print (" % ");
Serial.print(" vocht(%):");
Serial.print(vochtigheid, 1);
Serial.println();
}
else
  Serial.println("wacht op nieuwe meting");

delay(500);
}

```

De weergave van de metingen is nogal onrustig. We maken nu twee aanpassingen aan het programma:

- 1) De getoonde sensorwaarde wordt alleen vervangen als er een verandering heeft plaats gevonden.
- 2) De gemeten CO2 waarde wordt berekend uit de laatste 10 metingen.

Wat meer uitleg bij de aanpassingen van het programma zal in een later stadium worden beschreven.

Sketch met middeling van CO2 waarden: Sketch CO2 meting_03

```

/*
Sketch CO2_meting_03 versie 1.0
Meting van CO2, temperatuur en vochtigheid met sensor SCD30
De weergave is op een 4 x 20 I2C LCD en op de seriële monitor
LEDjes voor grenswaarden CO2
Ververs sensorwaarden op het display alleen indien waarde veranderd is
De CO2 waarde wordt berekend uit de laatste 10 metingen
Het oorspronkelijk programma is gemaakt door Nathan Seidle van SparkFun Electronics
Aangepast voor Coderdojo
*/

#include <Wire.h>           // laad de I2C bibliotheek
#include <LiquidCrystal_I2C.h> // bibliotheek voor het LCD scherm
#include "SparkFun_SCD30_Arduino_Library.h" // laad de SCD30 bibliotheek

LiquidCrystal_I2C lcd(0x27, 20, 4); // start LCD scherm en lcd is gebruiksnaam
SCD30 Sensor;           // het SCD30 board krijgt gebruiksnaam Sensor

float temperatuur, vorigeTemperatuur, vochtigheid, vorigevochtigheid; // variabelen
int CO2waarde, vorigeCO2waarde;
int index, som, tempint, vochtint;
int ledPins[] = {4, 5, 6};
int co2array [11];

byte graden[8] = {        // graden teken
  B00010,
  B00101,
  B00101,
  B00010,
  B00000,
  B00000,
  B00000,
  B00000
};

void emptyline()          // lege lijn routine
{
  lcd.print ("          ");
}

void setup()
{
  for (index = 0; index <= 3; index++)
  {
    pinMode(ledPins[index], OUTPUT);
  }
  Serial.begin(115200);
  Serial.println("SCD30 programma met LCD uitlezing");
  Wire.begin();

  if (Sensor.begin() == false)

```

```

{
  Serial.println("Sensor niet gevonden.....");
  while (1);
}

lcd.init();          // initialiseer het LCD scherm
lcd.backlight();    // zet de backlight aan
lcd.createChar(0, graden); // het symbool graden staat nu op positie 0
lcd.clear();        // wis het scherm
lcd.setCursor(0, 0); // zet de cursor op positie 1, regel 0
lcd.print (" CO2-Temp-Vocht ");
}
//De SCD30 geeft data om de 2 seconden

void loop()
{
  if (Sensor.dataAvailable())
  {
    CO2waarde = Sensor.getCO2();
    if (CO2waarde > 3100)
      CO2waarde = 3100;
    som = 0;
    for (index = 9; index > -1; index--)
    {
      co2array[index + 1] = co2array[index];
      som = som + co2array[index];
    }
    co2array[0] = CO2waarde;
    CO2waarde = (som) / 10;

    if (CO2waarde != vorigeCO2waarde)
    {
      lcd.setCursor(0, 1); // zet de cursor op positie 1, regel 1
      emptyline();
      lcd.setCursor(0, 1); // zet de cursor op positie 1, regel 1
      lcd.print ("CO2 : "); lcd.print(CO2waarde); lcd.print (" ppm");
    }

    vorigeCO2waarde = CO2waarde;

    if (CO2waarde >= 1200)
    {
      digitalWrite(ledPins[0], HIGH); // GEVAAR
      digitalWrite(ledPins[1], LOW);
      digitalWrite(ledPins[2], LOW);
    }
    else if (CO2waarde >= 900)
    {
      digitalWrite(ledPins[0], LOW);
      digitalWrite(ledPins[1], HIGH); //opletten
      digitalWrite(ledPins[2], LOW);
    }
  }
}

```

```

else
{
  digitalWrite(ledPins[0], LOW);
  digitalWrite(ledPins[1], LOW);
  digitalWrite(ledPins[2], HIGH); //veilig
}
Serial.print("co2 waarde(ppm):");
Serial.print(CO2waarde);

temperatuur = Sensor.getTemperature();
temperatuur = temperatuur * 10;
tempint = temperatuur;
temperatuur = tempint / 10.;
if (temperatuur != vorigetemperatuur)
{
  lcd.setCursor(0, 2);    // zet de cursor op positie 1, regel 2
  emptyline();
  lcd.setCursor(0, 2);    // zet de cursor op positie 1, regel 2
  lcd.print ("Temp : "); lcd.print(temperatuur, 1); lcd.write(0); lcd.print ("C");
}
vorigetemperatuur = temperatuur;
Serial.print(" temp(C):");
Serial.print(temperatuur, 1);

vochtigheid = Sensor.getHumidity();
vochtigheid = vochtigheid * 10;
vochtint = vochtigheid;
vochtigheid = vochtint/10.;
if (vochtigheid != vorigevochtigheid)
{
  lcd.setCursor(0, 3);    // zet de cursor op positie 1, regel 3
  emptyline();
  lcd.setCursor(0, 3);    // zet de cursor op positie 1, regel 3
  lcd.print ("Vocht: "); lcd.print(vochtigheid, 1); lcd.print (" % ");
}
vorigevochtigheid = vochtigheid;

Serial.print(" vocht(%):");
Serial.print(vochtigheid, 1);
Serial.println();
}
else
  Serial.println("wacht op nieuwe meting");

delay(2000);
}

```