



CO₂-wachter

luchtkwaliteit bewaken – doe het zelf!

Mathias Claußen, Ton Giesberts en
Luc Lemmens (Elektor)
ontwerp: Florian Schäffer (Duitsland)

Hoge CO₂-concentraties kunnen problemen veroorzaken in slecht geventileerde ruimten. Deze CO₂-wachter biedt daarvoor een oplossing. Het ontwerp, gebouwd met through-hole componenten, waarschuwt de gebruiker met drie LED's en een akoestisch alarm wanneer het tijd is voor frisse lucht. Gerber-bestanden van de print en de ontwerpbestanden van het frontpaneel kunnen gratis worden gedownload.

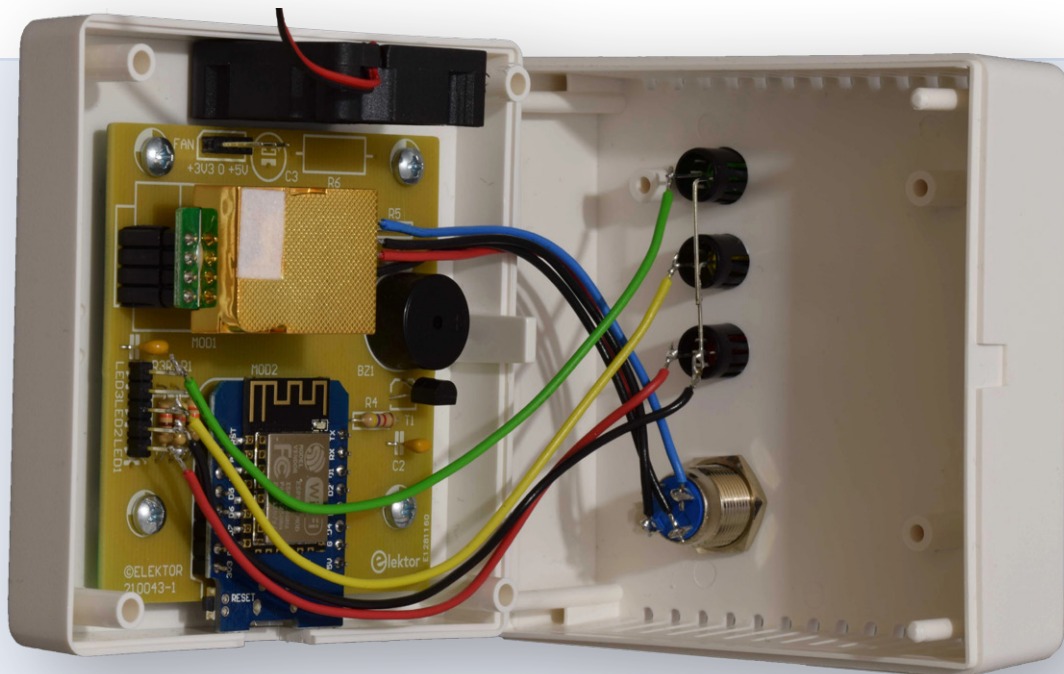


De schakeling

Het hart van het schema van **figuur 1** is MOD2, een Wemos D1 Mini, een op de ESP8266EX gebaseerde WiFi-module met 4 MB flash-programmageheugen. Deze verzamelt gegevens van een MH-Z19C CO₂-sensor (MOD1) via een software-UART, stuurt drie LED's aan om het CO₂-niveau te tonen (LED1...3), en bedient tevens zoemer BZ1, die als akoestisch alarm dient wanneer het CO₂-niveau onaanvaardbaar hoog is. De CO₂-wachter wordt gevoed via de micro-USB connector van de Wemos-module, met behulp van een standaard 5V_{DC}-netadapter. Deze voedingsspanning wordt via een interne diode en zekering naar pen 9 van MOD2 gevoerd en naar een ingebouwde 3,3V-spanningsregelaar. De uitgang van deze LDO wordt gebruikt voor de interne voeding van de Wemos-module zelf, en is ook verbonden met pen 8 van de module. Deze 3,3 V kan worden gebruikt voor het voeden van een miniatur-ventilator aangesloten op K1, zoals we later zullen zien. De 5V-uitgang van MOD2 wordt gebruikt voor de voeding van de CO₂-sensor en van een extra LED in schakelaar S1 om aan te geven dat de WiFi-verbinding actief is. Deze voeding kan ook worden aangesloten op K1 (via R6) als alternatieve voeding voor de ventilator. Met drukknop S1 kan de CO₂-sensor worden geactiveerd om de automatische kalibratie te starten.

Deze CO₂-wachter is een doe-het-zelf project voor de bewaking van de binnenlucht en de beoordeling van de luchtkwaliteit op basis van de kooldioxideconcentratie (*Indoor Air Quality*, IAQ). Het apparaat maakt gebruik van een niet-dispersieve infrarood-sensor (NDIR) en geeft de luchtkwaliteit in vijf bereiken weer met behulp van drie LED's. Bovendien klinkt een alarm als de CO₂-concentratie onaanvaardbaar hoog is. Het is mogelijk om de gemeten waarden via WiFi (CO₂-concentratie en temperatuur) naar het ThingSpeak IoT-analyseplatform te sturen en de gegevens grafisch te evalueren.

Er zijn veel factoren die de luchtkwaliteit beïnvloeden, maar de CO₂-concentratie in een ruimte wordt vrijwel onmiddellijk gevoeld. Een te hoge concentratie van dit gas veroorzaakt vermoeidheid, concentratieverlies of zelfs misselijkheid, om nog maar te zwijgen van de blijvende negatieve effecten als de CO₂-concentratie in uw omgeving veel te hoog is. Dit laatste is (hopelijk!) niet erg waarschijnlijk in een woning of kantoor, maar deze CO₂-wachter geeft in elk geval tijdig een indicatie om een ruimte te ventileren.



Figuur 3. De ventilator is geïnstalleerd.

zich aan de onderkant van de print bevinden, is het relatief eenvoudig om componenten te desolderen als een fout is gemaakt. Begin met het monteren van de laagste componenten (weerstand en diode), gevolgd door de condensatoren, transistor, zoemer en headers (busstrips voor de CO₂-sensor MOD1, optionele pinheaders voor de schakelaar en LED's). Om de D1 Mini-module (MOD2) te monteren, soldeert u eerst de twee achtpolige male headers aan die zijde van de module waar de USB-connector en de meeste andere componenten zich bevinden (zie **figuur 3**). Wij raden aan om voor MOD2 geen busstrips op de print te gebruiken, omdat er bij het insteken en uittrekken van de USB-connector enige kracht op de module wordt uitgeoefend. Het beste is om de D1 Mini op de printplaat te solderen. Bovendien kan het gebruik van busstrips de luchtstroom door de ventilatieopeningen van de behuizing belemmeren.

Wanneer de print met vier schroeven in de onderste helft van de behuizing is gemonteerd en de ventilator is vastgelijmd (zie hieronder), kan de sensormodule in de bijbehorende bussen worden gestoken. **Raak nooit de bovenkant van de sensor aan**, zoals in de datasheet staat [2]: "Vermijd elke druk uit welke richting dan ook op de vergulde plastic kamer tijdens solderen, installeren en gebruik. Raak vooral nooit de luchtopening (wit stoffilter) aan".

De vier- en vijfpolige busstrips voor de sensor worden waarschijnlijk geleverd als één 10-pins SIL busstrip. Knip deze bij de vijfde pen door om afzonderlijke vier- en vijfpolige busstrips te krijgen.

Vorbereiden van de behuizing

Gebruik een kopie van het frontpaneel-ontwerp als sjabloon voor de juiste posities van de vier gaten (drie LED's en de schakelaar), en gebruik een centerponsje om de middelpunten aan te geven. Op de Elektor Labs-pagina van dit project [3] staan enkele tips en trucs om beschadiging van de behuizing te voorkomen bij het boren van de gaten. Let bij dit soort kastjes vooral op de lange pinnen waarmee de behuizing wordt gesloten; die kunnen breken als ze niet goed worden behandeld! Er moet een inkeping worden gemaakt in één kant van de onderste helft van de behuizing om toegang te krijgen tot de micro-USB connector van de D1 Mini-module, die dient als de voeding voor de CO₂-wachter. Zorg ervoor dat de uitsparing groot genoeg is voor de

USB-plug van uw voeding. Plaats de D1 Mini-module op de print om de juiste plaats voor de inkeping aan te geven. U hoeft de module met de bijbehorende male headers nog niet op de print te solderen. Als de module op de print is geplaatst en de print met wellicht twee schroeven tijdelijk in de behuizing is vastgezet, kan de plaats van de inkeping worden afgetekend. Deze zit iets uit het midden.

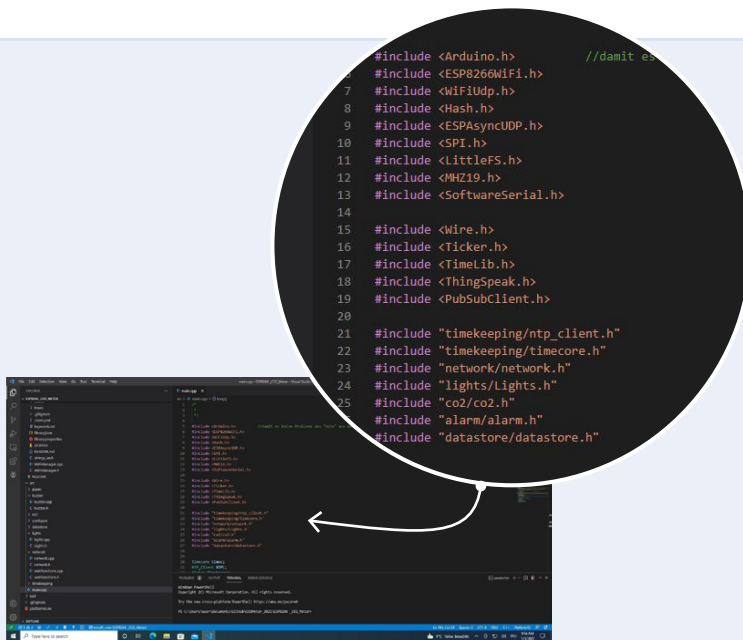
Er hoeven geen gaten te worden gemaakt voor de luchtstroom van de ventilator; deze wordt met superlijm of epoxykit aan de onderzijde van de behuizing vastgelijmd (**figuur 3**). Om resonantie te voorkomen mag de ventilator het deksel van de behuizing niet raken. De ventilator moet lucht uit de behuizing blazen. Voordat u de ventilator monteert, controleert u de richting van de luchtstroom door hem aan te sluiten op een voeding van 3,3...5 V of controleert u of er een pijl op de zijkant van het frame van de ventilator staat die de richting van de luchtstroom aangeeft. Let goed op de polariteit van de aansluitingen. Een verkeerde polariteit kan de ventilator beschadigen. (In moderne exemplaren zit vaak wat elektronica).

Bedrading van de onderdelen op het frontpaneel

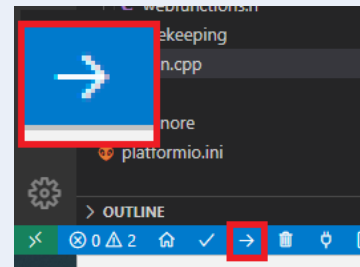
Voor het aansluiten van de LED's zijn slechts vier draden nodig. De massaverbinding van de print naar drie indicatie-LED's in het frontpaneel kan gemeenschappelijk worden uitgevoerd, soldeer gewoon de drie kathodes van de LED's aan één enkele draad.

Zorg ervoor dat er geen draden op de antenne van MOD2 liggen wanneer het deksel van de behuizing op het onderste gedeelte wordt geplaatst. Houd ze zo ver mogelijk van de Wemos-module vandaan, maar maak ze niet langer dan nodig. De draden naar de schakelaar kunnen 8,5 cm lang zijn en de draden naar de LED's ongeveer 12 cm. Dan kunt u de bovenkant van de behuizing verwijderen en deze op zijn kant naast de onderzijde leggen zonder de bedrading te hoeven verwijderen. Er zijn geen schroeven nodig om de behuizing te sluiten; vier lange plastic pennen van het deksel passen in de vier lange montagebussen.

Natuurlijk kunt u uw eigen frontpaneel ontwerpen. U kunt bijvoorbeeld een aparte blauwe LED en schakelaar gebruiken. Maar let op de plaatsing van de LED's en de schakelaar; ze mogen niet te dicht bij de modules komen.



Figuur 4. De code is in PlatformIO IDE geladen.



Figuur 5. Knop voor het uploaden van de firmware.

De firmware in de Wemos laden

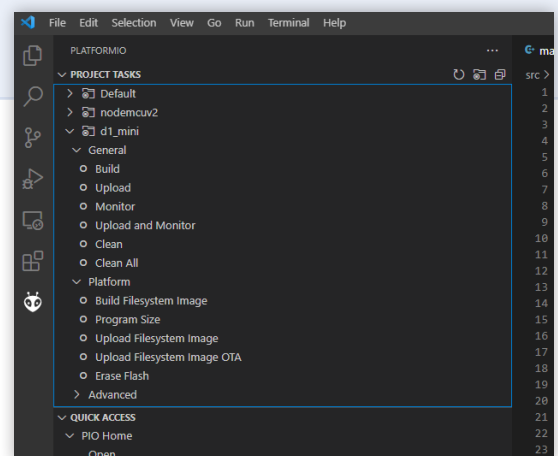
De firmware voor de ESP8266 op de Wemos-module is geschreven met het Arduino Framework, dus u hebt de sketch en een set bibliotheken nodig. Voor de ontwikkeling van de code van deze CO₂-wachter worden PlatformIO en Visual Studio Code gebruikt. De bibliotheken, het project en de upload-instellingen zijn gecombineerd, zodat het compileren van de code niet zo'n probleem zou mogen zijn.

Om Visual Studio Code en PlatformIO te installeren, kunt u de aanwijzingen [4] volgen die PlatformIO beschikbaar stelt. Na de installatie moet u de broncode uit de Elektor GitHub Repository halen. Als u niet zeker weet hoe u aan de code komt, kunt u de video bekijken die onze collega Clemens Valens [5] heeft gemaakt. En als u meer wilt weten over GitHub en het gebruik van Repositories, kijk dan eens naar het Elektor Webinar over GitHub [6]. Nadat u de code hebt en deze opent met de PlatformIO IDE, moet u iets zien zien als **figuur 4**. Sluit de Wemos aan op uw computer en druk op de knop voor het uploaden (**figuur 5**). PlatformIO zal alle benodigde bibliotheken, tools, board support packages downloaden en beginnen met compileren en aansluitend uploaden.

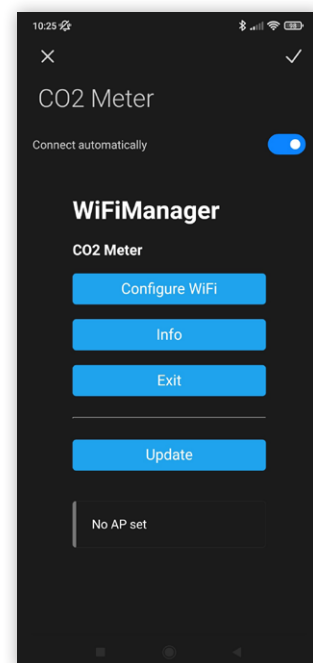
PlatformIO heeft nu de firmware gebouwd en geflashed. Zoals bij de meeste ESP8266-projecten zijn we er echter nog niet. Omdat de firmware via een webserver geconfigureerd kan worden, moeten we in een tweede stap ook de webpagina's naar de ESP8266 schrijven. Klik in Visual Studio Code op het kleine 'mierenkopje'; er moet dan een lijst met taken verschijnen (**figuur 6**). Klik eerst op *Build Filesystem Image* en daarna op *Upload Filesystem Image*. Dit zal de inhoud voor de webserver naar de ESP8266 schrijven. Als u foutmeldingen krijgt in de stijl van "can't access COM port", controleer dan of u niet ergens een seriële terminal hebt geopend die de poort blokkeert. Daarna is de firmware klaar om geconfigureerd en gebruikt te worden.

WiFi- en cloud-connectiviteit

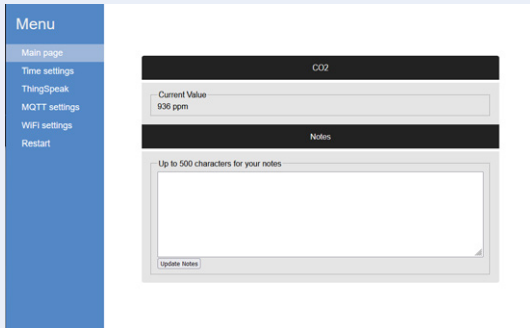
Als de ESP8266 op uw Wemos-module geen verbinding kan maken met een WiFi-netwerk, zal hij een access point starten met de naam "CO2 Meter". Maak er verbinding mee en open een browser die naar <http://192.168.4.1> wijst. U zou de WiFi Manager configuratie-interface moeten zien (**figuur 7**). Klik op *Configure WiFi* en selecteer uw netwerk;



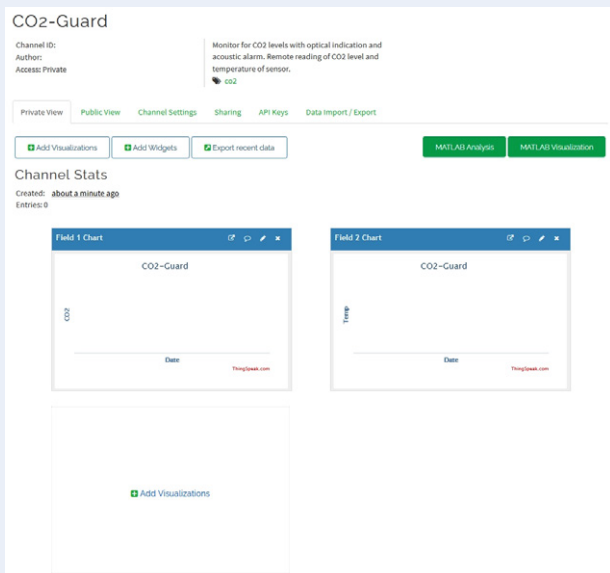
Figuur 6. De kleine 'mierenkop' toont meer opties.



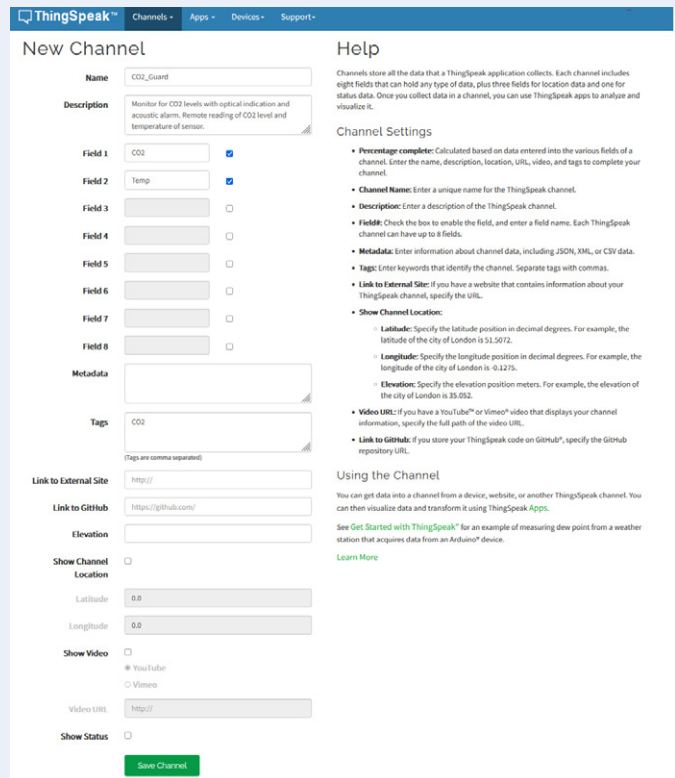
Figuur 7. WiFi-instellingen.



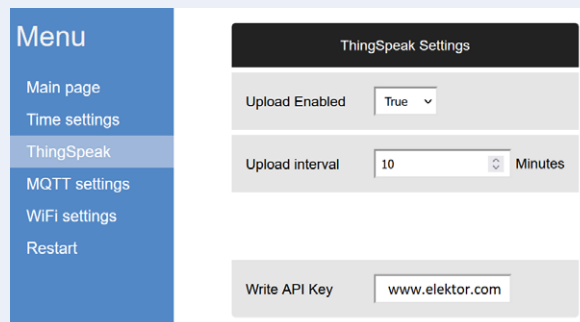
Figuur 8. Web-interface van de CO₂-wachter.



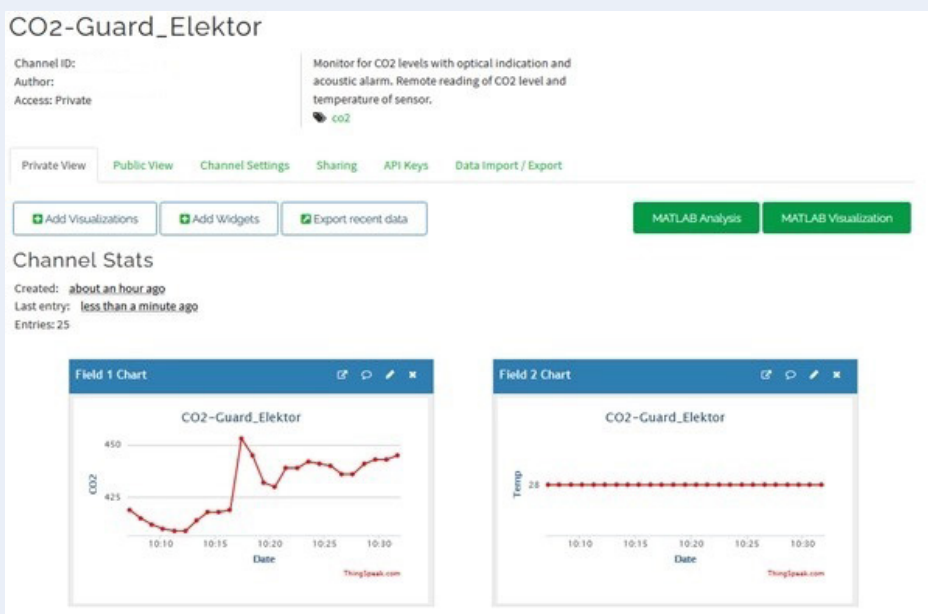
Figuur 10. Nieuw aangemaakt ThingSpeak-kanaal.



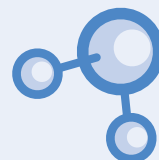
Figuur 9. ThingSpeak-instellingen.



Figuur 11. ThingSpeak-instellingen.



Figuur 12. Binnenkomende data.





pas de gewenste instellingen toe voordat u op save klikt. De ESP8266 zal nu proberen verbinding te maken met uw netwerk. Als hij een verbinding tot stand kan brengen, kunt u binnen uw netwerk <http://co2guard.local> gebruiken om toegang te krijgen tot de web-interface (**figuur 8**). Vanaf hier kunt u de CO₂-wachter gaan configureren. U kunt de CO₂-wachter desgewenst verbinden met een webgebaseerde oplossing voor gegevensopslag zoals ThingSpeak of een MQTT-broker zoals die wordt gebruikt in Node-RED. Met de laatste kunt u nog meer automatisering aan de sensor toevoegen – bijvoorbeeld een bericht naar uw mobiele telefoon sturen als het CO₂-niveau een bepaalde waarde overschrijdt. Om met ThingSpeak aan de slag te gaan, beschrijven we in het kort hoe een account kan worden ingericht.

ThingSpeak: maak een MathWorks-account aan

Als u alleen de actuele CO₂-concentratie wilt monitoren in de ruimte waarin u zich bevindt, dan hebt u geen WiFi-verbinding nodig en al helemaal geen toegang tot een ThingSpeak-account – de LED's (en in extreme gevallen de zoemer) geven dan een duidelijke indicatie van de luchtkwaliteit, althans: wat de CO₂ betreft. Maar als u bijvoorbeeld het verloop van de gemeten waarden wilt loggen en als functie van de tijd wilt analyseren, of van een grotere afstand wilt kunnen zien hoe hoog de concentratie is, dan ligt een verbinding met ThingSpeak via WiFi voor de hand. U kunt dan ook de temperatuur volgen die ook door de sensormodule wordt gemeten, maar op geen enkele manier op de CO₂-wachter zelf wordt getoond.

Om een online-kanaal aan te maken voor uw metingen, gaat u naar de ThingSpeak-website [7] en kiest u *Get Started For Free*. Op de volgende pagina kunt u een MathWorks-account aanmaken (als u dat nog niet hebt). Hierna maakt u een nieuw kanaal aan en voert u de benodigde gegevens in zoals in **figuur 9**. Klik op *Save Channel*. Er verschijnt een overzicht van het zojuist aangemaakte kanaal zoals in **figuur 10**.

Onder de tab *API Keys* moeten we de *Write API Key* kopiëren voor later gebruik. Houd deze voor uzelf en laat hem aan niemand anders zien. Ga naar de web-interface van de CO₂-wachter en selecteer de ThingSpeak-optie in het zijmenu. Er zal een pagina met de instellingen voor ThingSpeak verschijnen. Voer de *API Key* in, selecteer een upload-interval en schakel ThingSpeak in (**figuur 11**). U gaat nu terug naar de MathWorks ThingSpeak-site. Onder het menu *Channels* kiest u *My Channels* en vervolgens de tab *Private View*. Na korte tijd zou u daar de eerste metingen moeten zien zoals in **figuur 12**.

Om uw gegevens met anderen te delen, kunt u afzonderlijke gebruikers met meekijkrechten toevoegen, of u kunt het kanaal publiek maken zodat iedereen uw gegevens kan bekijken. Afzonderlijke gebruikers krijgen een e-mail op het e-mailadres dat u kunt invoeren op het tabblad *Sharing*. Als u gebruik maakt van een gratis (niet betaald) ThingSpeak-plan, is het delen van data beperkt tot maximaal drie andere gebruikers. Voor meer details over het delen van data kunt u het MathWorks-helpcenter [8] raadplegen.

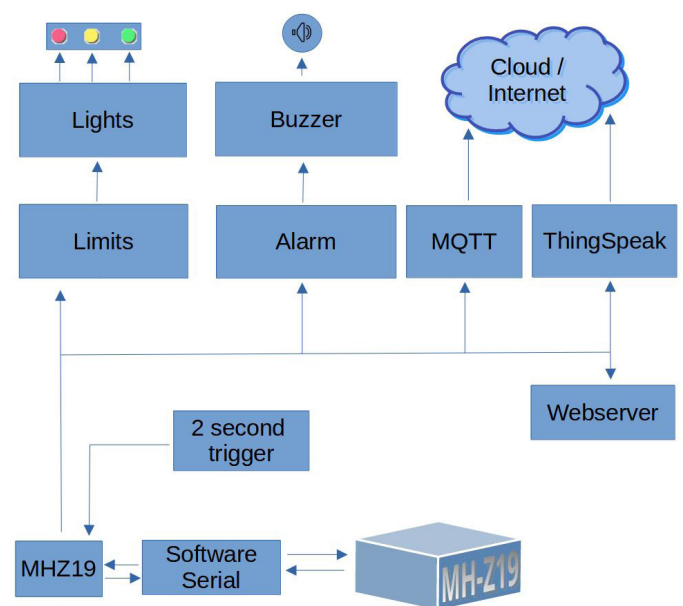
MQTT-verbinding

MQTT is een veelzijdige manier om gegevens te distribueren naar andere systemen voor opslag of verdere verwerking. U hebt geen

externe service nodig om gegevens te verzamelen of te verwerken, en u kunt de gegevens binnen uw eigen netwerk houden. Een van de tools die MQTT-connectiviteit bieden is Node-RED (bijvoorbeeld op een Raspberry Pi). We hebben in het verleden al veel over installaties voor Node-RED geschreven, en als u een snelle installatiegids voor de Raspberry Pi wilt, kijk dan eens op de Node-RED website [9]. Voor MQTT binnen de web-interface van de CO₂-wachter kunt u een server en een *topic* opgeven waar data gepubliceerd moet worden (**figuur 13**). De gegevens zelf worden verzonden als JSON-string, met de objecten "CO2Value" en "Temp", die de CO₂-concentratie in ppm en de temperatuur in °C weergeven. Als u meer wilt weten over Node-RED en de Raspberry Pi, werp dan eens een blik in het boek van Dogan Ibrahim, *"Programmeren met Node-RED"* (Elektor 2020) [10].

KiCad PCB-ontwerpbestanden

De print is oorspronkelijk ontworpen in Altium, en de bijbehorende Gerber-bestanden zijn gebaseerd op dit beproefde ontwerp. Dat is vrij eenvoudig, zoals u al aan het schema kunt zien. Aangezien dit niet zo complex is (zie figuur 1), is het gereconstrueerd in KiCad. Naast het schema moest ook het printontwerp worden overgezet, maar omdat we de laatste stabiele versie van KiCad 5 gebruikten, was er nog geen native (geïntegreerde) importmogelijkheid voor Altium-boards beschikbaar. Een truc is om de corresponderende Gerber-bestanden te gebruiken en deze in KiCad te importeren. De Gerber-viewer in KiCad kan printbestanden genereren die u kunt gebruiken in de PCB layout tool. Dit levert niet alleen de omtrek van de print maar ook de footprints en montagegaten aan de hand waarvan we de componenten kunnen plaatsen. Op deze manier kunnen PCB's in KiCad worden



Figuur 13. MQTT-instellingen.



geïmporteerd, maar het verdient de voorkeur om KiCad versie 6 en de native Altium board-importmogelijkheid te gebruiken als u met Altium-bestanden te maken hebt.

Maar waarom al deze moeite? Ten eerste kunnen anderen dan direct met de KiCad-bestanden werken en ze indien gewenst wijzigen. Ook is het bij sommige PCB-services mogelijk een KiCad-print te laten vervaardigen door het gewoon te slepen en neer te zetten (bijvoorbeeld Aisler) [11]. U hoeft niet naar Gerber te exporteren en boorbestanden te genereren en daar een ZIP-bestand van te maken. Als uw printleverancier echter Gerber- en boorbestanden nodig heeft, kunt u die nog altijd in KiCad genereren volgens de specificaties die uw leverancier vereist.

Als u uw eigen behuizing rondom de print wilt ontwerpen, heeft de combinatie van KiCad en FreeCad zich ruimschoots bewezen, zoals blijkt uit ons kerstproject [12]. En het printen van een behuizing kan voor sommigen handiger zijn dan het boren van gaten in een prefab-exemplaar. En als u thuis een 3D-printer hebt staan die stof staat te verzamelen, waarom zou u die dan niet gebruiken? Het betekent ook dat u één onderdeel minder hoeft te bestellen. De 3D-printbare behuizing is op het moment van schrijven van dit artikel nog niet af, maar er wordt wel aan gewerkt. En dat betekent dat u geduld moet hebben of toch wat gaten zult moeten boren...

Een eigen ontwerp

Het bouwen van een CO₂-sensor voor thuis of op het werk is niet extreem moeilijk, en zoals u weet zijn er al veel CO₂-meters ontwikkeld, commercieel maar ook voor zelfbouw. Met de MH-Z19 sensor die in dit project wordt gebruikt, krijgt u een heel behoorlijk en betaalbaar instrument. De ESP8266 is een beproefde WiFi-microcontroller die niet alleen kan worden geprogrammeerd met het Arduino Framework, maar in de

huidige samenstelling ook ondersteuning biedt voor Home Assistant. Wilt u uw eigen sensor bouwen? Dan moet u ongeveer € 30 uitgeven voor de sensor zelf (in Europa). De schakelaar, Wemod D1 Mini, print, LED's, behuizing, ventilator komen daar nog bij (plus porto), dus u komt in totaal uit op ongeveer € 60 aan onderdelen. Aan de andere kant heeft het bewaken van de CO₂-concentratie in uw omgeving veel voordelen. Als u de CO₂-concentratie kent, kunt u ervoor zorgen dat er voldoende frisse lucht komt. ◀

210043-03

Vragen of opmerkingen?

Hebt u technische vragen of opmerkingen naar aanleiding van dit artikel? Stuur een e-mail naar de redactie van Elektor via redactie@elektor.com.

Een bijdrage van

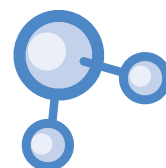
Oorspronkelijk hardware-ontwerp: **Florian Schäffer**

Ontwerp en tekst: **Mathias Claußen, Ton Giesberts, Luc Lemmens**

Redactie: **Jens Nickel, C. J. Abate**

Vertaling: **Eric Bogers**

Layout: **Harmen Heida**



Software

Bij de software die oorspronkelijk door de ontwerper op het Elektor Labs-platform werd gepost, was een aantal zaken voor verbetering vatbaar. Zo zijn hard-coded inloggegevens voor uw WiFi-netwerk alleen zinvol als u nooit de netwerknaam of het wachtwoord verandert waarmee het apparaat is verbonden. Ook is het niet erg veilig om uw inloggegevens ergens in een aantal codebestanden op te slaan. Hetzelfde geldt voor de ThingSpeak API Key. Als u deze sleutel moet veranderen, betekent dit dat u de hele code opnieuw moet compileren en uploaden naar het apparaat.

Wat het Elektor lab hard gecodeerd heeft gelaten (als #defines in /configure/configure.h) is de pin mapping, alsmede de CO₂-drempelwaarden voor de drie LED's en de buzzer. In het bestand /configure/configure.h staan alle hard gecodeerde instellingen op één plaats en kunt u ze wijzigen indien nodig. Als u instellingen in dit bestand wijzigt, moet u de firmware uiteraard opnieuw compileren en uploaden naar de ESP8266.

Binnen de software is de datastroom vrij eenvoudig. Om de twee seconden worden de sensorwaarden opgevraagd en geretourneerd naar verschillende code-modules. Voor de lokale bewaking van het CO₂-niveau hebben we de limits-klasse, die uiteindelijk de drie LED's zal aansturen, en we hebben de alarm-klasse, die de zoemer zal laten klinken als een bepaald CO₂-niveau wordt overschreden. Deze klassen zijn op een generieke manier opgebouwd, zodat het gemakkelijk is om de aansturing van de LED's te veranderen voor een andere weergave van de meetwaarden. Hetzelfde geldt voor de zoemer. Als u een actieve zoemer wilt gebruiken of een geluidsmodule, dan is deze klasse flexibel genoeg om dit te realiseren.

Voor de cloud-connectiviteit zijn waar mogelijk bestaande bibliotheken gebruikt, met name voor ThingSpeak. Dit maakt de gegevensuitwisseling met deze dienst een stuk eenvoudiger dan bij het ESP32-weerstation [13] dat we een paar jaar geleden publiceerden [14].



ONDERDELENLIJST

Weerstanden:

R1,R2,R3 = 330 Ω, 0,25 W, 5%

R4 = 4k7, 0,25 W, 5%

R5 = 0 Ω (draadbrug, zie tekst)

R6 = 0 Ω (draadbrug voor 5 V, of laat weg, zie tekst)

R7 = 1k5, 0,25 W, 5%

Condensatoren:

C1,C2 = 100 n/50 V, 10%, X7R, steek 5 mm

C3 = 100 µ/25 V, optioneel, niet gemonteerd, zie tekst

Halfgeleiders:

D1 = BAT85, DO-35

T1 = BC327, PNP, TO-92

LED1 = LED rood, 5 mm

LED2 = LED geel, 5 mm

LED3 = LED groen, 5 mm

Overig:

MOD1 = 1x10 SIL-header, female, gedeeld in 1x4 en 1x5 (zie tekst)

MOD1 = MH-Z19C, versie 400-5000 ppm, pinnen aan onderzijde

MOD2 = Wemos D1 Mini met 2 stuks 1x8 SIL-header

S1 = momentschakelaar met ingebouwde LED (blauwe ring)

Behuizing PP73BL, Supertronic

BZ1 = DC-buzzer, 5 V, steek 5 of 7,62 mm, Ø 14 mm max

krimpkous, binnendiameter 2,4 mm, 20 cm
LED-houder voor 5mm-LED (LED1...3), zwart
ventilator 5 VDC, 30x30x10,5 mm
12-polige male SIL-header
print 210043-1 v2.0

Dun soepel montagedraad

voor aansluiting LED's en schakelaar:

montagedraad, 0,25 mm², zwart (2x8,5 + 12 cm)

montagedraad, 0,25 mm², rood (8,5 + 12 cm)

montagedraad, 0,25 mm², geel (12 cm)

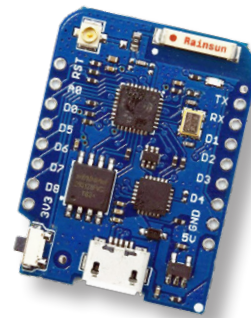
montagedraad, 0,25 mm², groen (12 cm)

montagedraad, 0,25 mm², blauw (8,5 cm)



GERELATEERDE PRODUCTEN

- WeMos D1 mini Pro – ESP8266 based WiFi Module (SKU 19185)
www.elektor.nl/19185
- PCB V2.1 with Components (excluding WeMos D1 and MH-Z19)
www.elektormagazine.de/aisler/co2guard
- H. Henrik Skovgaard, IoT Home Hacks with ESP8266 (SKU 19158)
www.elektor.nl/19158
- D. Ibrahim, Programming with Node-RED (E-book) (SKU 19225)
www.elektor.nl/19225



WEBLINKS

- [1] Elektor GitHub-repository: <https://github.com/ElektorLabs/210043-CO2-Guard>
- [2] Winsen MHZ-19 datasheet: www.winsen-sensor.com/d/files/infrared-gas-sensor/mh-z19b-co2-ver1_0.pdf
- [3] "CO2 Guard," op Elektor Labs: www.elektormagazine.nl/labs/co2-guard
- [4] PlatformIO installatie-handleiding: <https://platformio.org/install/ide?install=vscode>
- [5] "How to Code (Download) or Clone Something From GitHub," Elektor TV: www.youtube.com/watch?v=X5e3xQBeqf8
- [6] "GitHub Webinar: My Way Into GitHub," Elektor TV: www.youtube.com/watch?v=j_LgvVhBdwQ
- [7] MathWorks ThingSpeak: <http://thingspeak.com>
- [8] MathWorks ThingSpeak Help: <https://bit.ly/ChannelPropertiesMATLAB>
- [9] Node-RED installatie-handleiding: <https://nodered.org/docs/getting-started/raspberrypi>
- [10] D. Ibrahim, Programming with Node-RED, Elektor 2020: www.elektor.com/programming-with-node-red-e-book
- [11] Aisler BV: <http://aisler.net>
- [12] M. Claußen, "DIY Christmas Fireplace: A 3D Puzzle with PCBs, LEDs, and Raspberry Pi Pico," ElektorMagazine.com: www.elektormagazine.com/articles/diy-christmas-fireplace-3d-puzzle-pcb-leds-raspberry-pi-pico
- [13] "ESP32 Weatherstation" op Elektor Labs: www.elektormagazine.nl/labs/esp32-weather-station-180468
- [14] R. Aarts, "ESP32-Weerstation," Elektor Magazine januari/februari 2019: www.elektormagazine.nl/magazine/elektor-72/42284