

MoWeMiS

Mobiel Weer & Milieu Station

Chris Dries & Chris Steyaert
Maakplaats Balen & Andromeda Mol

Met steun van VVS en de bibliotheek Balen



Inhoud

Inleiding.....	1
Raspberry Pi voorbereiden.....	2
OS installatie.....	2
Software installatie.....	3
MQTT:.....	3
Python:	3
Mousepad editor:.....	3
Node-Red:.....	4
Autostart bij bootup	4
Maria DB op Pi:	4
Visual studio code:	5
bibliotheeken installeren:	5
Grafana:	5
De hardware	7
Principe schema	7
Raspberry Pi 4:.....	7
Wemos D1 (Arduino clone):	8
5V voeding Arduino:.....	8
BME680 Sensor:	8
MQ135 gas sensor:.....	9
DS18B20 temperatuur sensor:	9
Aansturen ventilatoren:	10
Samenbouwen van de componenten:	10
De source code	12
C++ op Wemos D1:.....	12
Bibliotheeken:	12
Declaraties:.....	12
Initialisatie:	12
Subroutines:	13
Python code:.....	14

Node Red:.....	14
Voorbeelden	15
De database:.....	16
Bijlage A technische termen:.....	17
Bijlage B Inline URL lijst:	17
Bijlage C Componentenlijst:	18
Bijlage D Softwarelijst:	18
Bijlage E Internet data bronnen:	18
Bijlage F Principe schema:	19

Inleiding

Om de bouwhandleiding te volgen is geen voorkennis vereist, maar enige technische kennis is wel een groot pluspunt.

Met behulp van sensoren worden een aantal weer en milieu gerelateerde variabelen gemeten. Die worden aangevuld met zowel weer als milieu data van het internet. De verzamelde gegevens worden als een geheel op een website weergegeven en in een database opgeslagen. Alle gebruikte software is opensource.

De sensoren worden gecontroleerd door een Wemos D1 die zijn data doorgeeft via MQTT voor centrale verwerking door een Raspberry Pi 4. De Raspberry Pi haalt met Python scripts gegevens van het internet. Met Node-Red worden de sensor en internet data dan gecombineerd tot één dashboard dat aan de gebruiker gepresenteerd wordt als een website op de Pi.

Als bouwproject is er zowel een praktische elektronische dimensie als een software dimensie. Het resultaat geeft naast het plaatselijk weer ook milieu parameters weer.

Van het internet kunnen, afhankelijk van de locatie van de gebruiker, de volgende parameters weergegeven worden: O3, NO2, CO, SO2, PM10, PM2.5, BC, C6H6 en NH3.

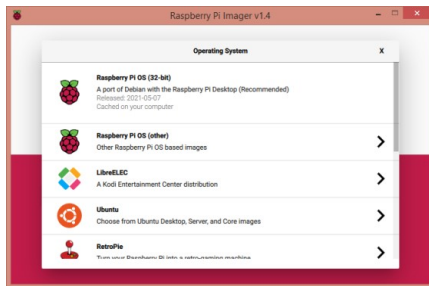
De sensoren meten: luchtdruk, relatieve vochtigheid, temperatuur, luchtkwaliteit, VOC (Kooldioxide, ammoniak, stikstofoxiden, benzeen, alcohol en rook) en optioneel PM10, en PM2.5 trillingen (aardschokken). De mogelijkheid onderzoeken om aardschokken te meten maakt deel uit van het project.

Als het meetstation operationeel is geeft het niet enkel een schat aan actuele informatie alle gewenste parameters kunnen ook opgeslagen worden in de database om nadien gebruikt te worden, bijvoorbeeld om trends te bekijken.

Raspberry Pi voorbereiden

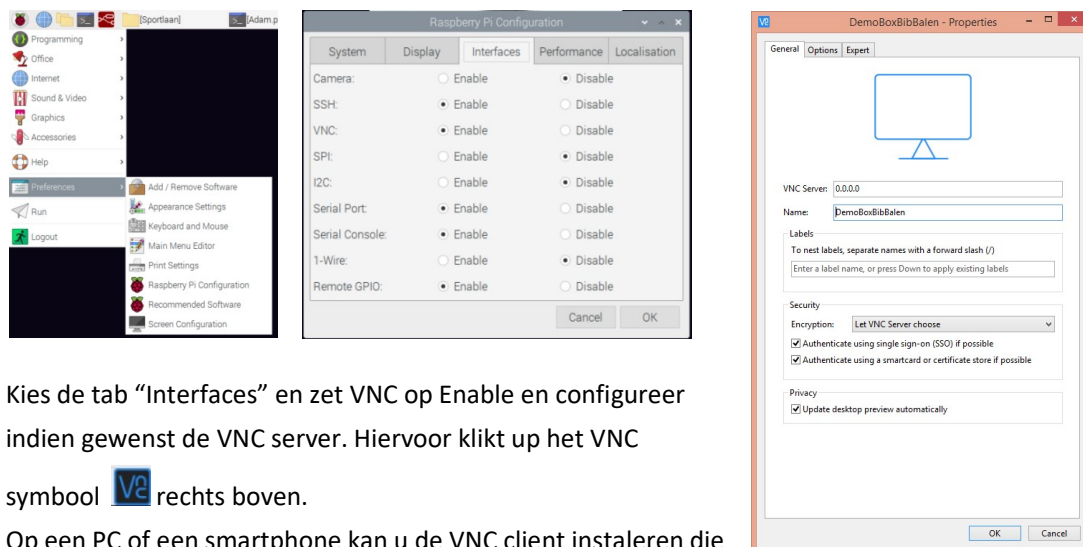
OS installatie


Afhankelijk of de micro-SD kaart al dan niet voorgeïnstalleerd is moet bij levering moet het OS nog op de micro SD kaart gekopieerd worden. Een gemakkelijke en veilige manier is om de [Raspberry Pi Imager](#) daarvoor te gebruiken. Voor dit project kies je Raspberry OS (32bit) bijgenaamd Rasbian. Ubuntu is mogelijk als alternatief maar wordt in deze handleiding niet gebruikt.



Vervolgens kies je de SD kaart in je PC waarop je het OS wil installeren. En klikt op “write”. Na afloop kan je de SD kaart in je Raspberry Pi (Pi) plaatsen, de Pi aansluiten en spanning geven. Het OS boot op. Tijdens de eerste boot krijg je een aantal vragen om je Pi te configureren naar jouw wensen. Dit proces valt niet in de scope van deze handleiding meer informatie over het configuratie proces vind je [hier](#)¹. Zet tijdens dit proces de VNC server actief. Zo kan je nadien de Pi benaderen vanaf uw thuis computer en hoeft je niet steeds een toetsenbord, muis en monitor met de Pi te verbinden. Mocht je de optie gemist hebben kan het nadien ook nog.

Kies Raspberry Pi configuration



Kies de tab “Interfaces” en zet VNC op Enable en configureer indien gewenst de VNC server. Hiervoor klikt up het VNC symbool  rechts boven.

Op een PC of een smartphone kan u de VNC client installeren die je verbinding laat maken met uw Pi. (U heeft de login gegevens van de PI nodig.)

¹ <https://raspberrypi-guide.github.io/getting-started/raspberry-pi-configuration>

Software installatie

Rasbian is een Linux gebaseerd OS. Deze handleiding is geen Linux cursus. Je kan de Linux commando's uit de handleiding kopiëren en plakken. Meer informatie over Linux vind je op het internet en in de bibliotheek.

De commando's staan steeds in deze vorm **opdracht**.

Enkele Linux commando's als voorbeeld:

Lijst de bestanden op in de actieve folder: **ls**

Test je internet connectie: **ping vrt.nu**

Voor je bijkomende software installeert is het, zeker bij een voorgeïnstalleerd image, best practice om eerst het OS te updaten en upgraden.

sudo apt update

sudo apt upgrade

Vervolgens wordt de nodige software geïnstalleerd.

MQTT:

<https://pimylifeup.com/raspberry-pi-mosquitto-mqtt-server/>

sudo apt install mosquitto mosquitto-clients

verifiëren met:

sudo systemctl status mosquitto

Mogelijk luisterd je Mosquitto server enkel op zijn locale poort (127.0.0.1). Om dat aan te passen moet je de config file aanpassen.

Een werkend voorbeeld (op Rasbian)

```
# Place your local configuration in /etc/mosquitto/conf.d/
#
# A full description of the configuration file is at
# /usr/share/doc/mosquitto/examples/mosquitto.conf.example

#allow connection from network
listener 1883 0.0.0.0

#allow anonymus
allow_anonymous true

pid_file /run/mosquitto/mosquitto.pid

persistence true
persistence_location /var/lib/mosquitto/

log_dest file /var/log/mosquitto/mosquitto.log

include_dir /etc/mosquitto/conf.d
```

Python:

<https://projects.raspberrypi.org/en/projects/generic-python-install-python3#linux>

sudo apt install python3 idle3

Mousepad editor:

<https://github.com/codebrainz/mousepad>

```
sudo apt-get install -y mousepad
```

Node-Red:

<https://nodered.org/docs/getting-started/raspberrypi>

De installatie gebeurt door een script

```
bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)
```

Autostart bij bootup

```
sudo systemctl enable nodered.service
```

Maria DB op Pi:

<https://pimylifeup.com/raspberry-pi-mysql/>

```
sudo apt install mariadb-server
```

Nu de MySQL-serversoftware op de Raspberry Pi is geïnstalleerd, moeten we deze beveiligen door een wachtwoord in te stellen voor de "root"-gebruiker. MySQL is standaard geïnstalleerd zonder wachtwoord, wat betekent dat u de MySQL-server kunt benaderen zonder enige authenticatie. Voer, indien gewenst, het volgende commando uit om het MySQL-beveiligingsproces te beginnen:

```
sudo mysql_secure_installation
```

Volg gewoon de prompts om een wachtwoord in te stellen voor de root gebruiker en om uw MySQL installatie te beveiligen. Voor een veiligere installatie moet je "Y" antwoorden op alle prompts als er gevraagd wordt om "Y" of "N" te antwoorden. Deze prompts zullen functies verwijderen die het iemand makkelijker maken om toegang tot de server te krijgen. Zorg ervoor dat je het wachtwoord dat je tijdens dit proces instelt opschrijft, aangezien we het zullen moeten gebruiken om toegang te krijgen tot de MySQL server en om databases en gebruikers aan te maken.

Als je nu toegang wilt krijgen tot de MySQL-server van uw Raspberry Pi en wijzigingen in uw databases wilt aanbrengen, kan je het volgende commando invoeren:

```
sudo mysql -u root
```

om een user aan te maken gebruikt u:

```
CREATE USER 'user'@'ip adres' IDENTIFIED BY 'paswoord';
```

Geef alle rechten aan deze user:

```
GRANT ALL PRIVILEGES ON *.* TO 'user'@'ip adres' IDENTIFIED BY 'paswoord';
```

Controleer of het gelukt is:

```
SHOW GRANTS FOR 'user'@'ip adres';
```

Oplijsten alle gebruikers:

```
SELECT User, host FROM mysql.user;
```

Het laatste wat we moeten doen om zowel onze MySQL database als onze gebruiker af te ronden is het flushen van de privilege tabel. Zonder de privilege tabel te flushen, zal de nieuwe gebruiker geen toegang hebben tot de database.

We kunnen dit doen door het volgende commando uit te voeren.

```
FLUSH PRIVILEGES;
```

Er zijn twee verschillende manieren om uit de MYSQL commando regel te stappen, de eerste is door

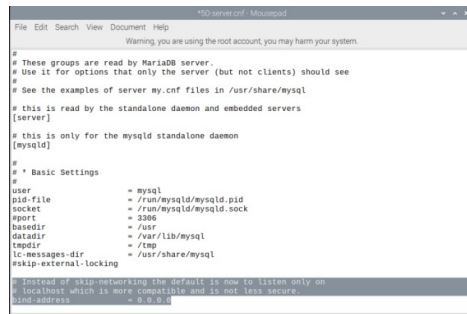
"quit;" te typen in de MySQL interface. De andere manier om uit de MYSQL command line te gaan is door op CTRL + D te drukken.

Om het mogelijk te maken om de database van op andere computers te beheren moet het bind adres op 0.0.0.0 gezet worden.

sudo mousepad

open het bestand etc/mysql/mariadb.conf.d/??-server.cnf

Zet het bind adres op 0.0.0.0



Visual studio code:

<https://code.visualstudio.com/docs/setup/raspberry-pi>

sudo apt install code

sudo apt upgrade code

bibliotheken installeren:

als pip3² niet geïnstalleerd is: **sudo apt-get install python-pip3**

pip3 install paho-mqtt

pip3 install beautifulsoup4

pip3 install requests

pip3 install gpiozero (enkel als u de Raspberry Pi IO gaat gebruiken bv voor de koelventilator)

sudo pip3 install mysqlclient

Grafana:

Volg de handleiding op: <https://grafana.com/docs/grafana/latest/setup-grafana/installation/debian/>

Start de server met system:

sudo systemctl daemon-reload

sudo systemctl start grafana-server

sudo systemctl status grafana-server

Configure the Grafana server to start at boot:

sudo systemctl enable grafana-server.service

standaard login: admin pasw: admin

Laatste update, afsluiten en herstarten

sudo apt update

² Pip3 is de package installer voor Python3. Je gebruikt pip3 om pakketten uit de Python Package Index en andere bibliotheken te installeren.

```
sudo apt upgrade
```

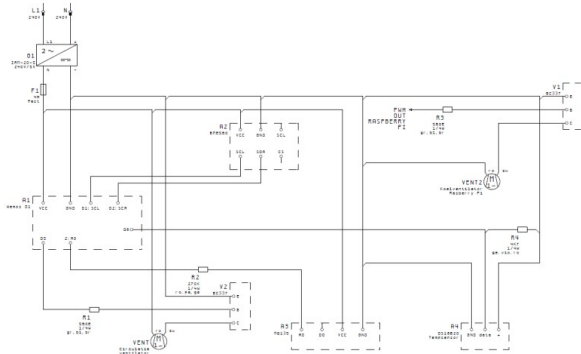
```
sudo reboot
```

De hardware

Om hardware te kunnen gebruiken moet je sommige componenten berekenen. Er is geen hogere wiskunde nodig voor de meeste berekeningen. De berekeningen zijn opgenomen in de handleiding, de onderliggende theorie niet.

Principe schema

Zie ook Bijlage F



Raspberry Pi 4:

De Pi heeft zelf ook een GPIO connector die desgewenst sensoren kan aansturen. In deze handleiding is er echter voor gekozen om alle sensoren aan te sturen via de Wemos D1 zodat de Pi ergens een vast plaatsje kan krijgen evt met andere randapparatuur. De sensor box is mobiel en kan eender waar geplaatst worden waar je Wifi bereik en spanning hebt. De GPIO van de Pi wordt wel gebruikt als u opteert om de Pi actief te gaan koelen. Dan wordt 1 pin gebruikt als PWM uitgang ([Pulsbreedtemodulatie](#)³) die via een transistor eindtrap de ventilator stuurt. Dit kan met de ingebouwde functie van de Pi of met eigen software.



Specifications

Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
2GB, 4GB or 8GB LPDDR4-3200 SDRAM (depending on model)
2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
Gigabit Ethernet
2 USB 3.0 ports; 2 USB 2.0 ports
Raspberry Pi standard 40 pin GPIO header (fully backwards compatible with previous boards)
2 x micro-HDMI ports (up to 4Kp60 supported)
2-lane MIPI DSI display port
2-lane MIPI CSI camera port
4-pole stereo audio and composite video port
H.265 (4Kp60 decode), H.264 (1080p60 decode, 1080p30 encode)
OpenGL ES 3.1, Vulkan 1.0
Micro-SD card slot for loading operating system and data storage
5V DC via USB-C connector (minimum 3A*)
5V DC via GPIO header (minimum 3A*)
Power over Ethernet (PoE) enabled (requires separate PoE HAT)
Operating temperature: 0 – 50 degrees C ambient

* A good quality 2.5A power supply can be used if downstream USB peripherals consume less than 500mA in total.

Raspberry Pi GPIO Header

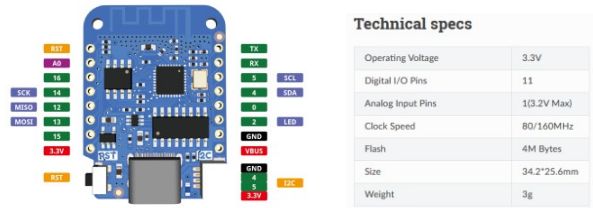


Pin#	NAME	NAME	Pin#
01	3.3v DC Power	DC Power 5v	02
03	GPIO-2 (SDA1, I2C)	DC Power 5v	04
05	GPIO-3 (SCL1, I2C)	Ground	06
07	GPIO-4 (GPICLK0)	(TXD0, UART) GPIO14	08
09	Ground	(RXD0, UART) GPIO15	10
11	GPIO17	(PWM0) GPIO18	12
13	GPIO27	Ground	14
15	GPIO22	GPIO23	16
17	3.3v DC Power	GPIO24	18
19	GPIO10 (SPI0_MOSI)	Ground	20
21	GPIO-9 (SPI0_MISO)	GPIO25	22
23	GPIO11 (SPI0_CLK)	(SPI0_CE0_N) GPIO18	24
25	Ground	(SPI0_CE1_N) GPIO17	26
27	GPIO-0 (SDA0, I2C)	(SCL0, I2C) GPIO11	28
29	GPIO-5	Ground	30
31	GPIO-6	(PWM0) GPIO12	32
33	GPIO13 (PWM1)	Ground	34
35	GPIO19	GPIO19	36
37	GPIO26	GPIO20	38
39	Ground	GPIO21	40

³ <https://nl.wikipedia.org/wiki/Pulsbreedtemodulatie>

Wemos D1 (Arduino clone):

De Wemos D1 is gebouwd rond een [ESP-8266EX](#) microcontroller. In dit project controleert de D1 alle sensoren. Hierbij worden zowel de analoge ingang als een aantal van de diverse communicatie mogelijkheden van de D1 gebruikt (WiFi, one wire, I²C).



Technical specs	
Operating Voltage	3.3V
Digital I/O Pins	11
Analog Input Pins	1(3.2V Max)
Clock Speed	80/160MHz
Flash	4M Bytes
Size	34.2*25.6mm
Weight	3g

5V voeding Arduino:

Je kan de D1 voeden met een Type-C USB voeding maar omdat we voor de sensoren ook 5V nodig hebben is het handiger om een onafhankelijke 5V voeding te gebruiken. De Mean Well power module is een AC/DC convertor (220V_{AC}/5V_{DC}) 4A met een ingebouwde beveiliging is een compacte oplossing.

BME680 Sensor:

De BME680 wordt geproduceerd door Bosch en heeft de volgende sensoren aan boord:

Luchtdruksensor:

Meetbereik: 300 - 1100hPa

Kleinste resolutie: 20bit

Relatieve nauwkeurigheid: ± 0.6 hPa

Temperatuursensor:

Meetbereik: 0-65°C

Kleinste resolutie: 20bit

Nauwkeurigheid: ± 1.0 °C

Vochtigheidssensor:

Meetbereik: 0 - 100%RH

Kleinste resolutie: 16bit

Nauwkeurigheid: ± 3 %

Luchtkwaliteit sensor:

Meetbereik: 0 - 500 IAQ

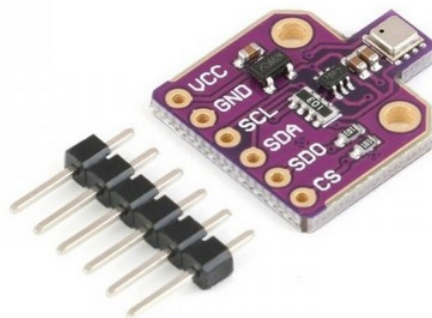
Kleinste resolutie: 1 IAQ index

Nauwkeurigheid: ± 15 %

Bedrijfstemperatuur: -40 - 85°C

Om het gebruik van de BME680 gemakkelijk te maken gebruiken we een BME680 die samen met rand componenten gemonteerd is op een breakout board om zo een module te vormen.

Verschillende fabrikanten gebruiken verschillende printplaten. Alle versies zijn in principe bruikbaar zolang u zich houdt aan de aansluitingen van uw board.



De BME680 communiceert met de Wemos D1 via I²C via de standaard IO kanalen D₁ en D₂. De voeding komt van de AC/DC convertor. In de source code wordt de Adafruit_Sensor.h bibliotheek gebruikt om met de sensor te werken.

MQ135 gas sensor:

Ook de MQ135 is op een breakout board gemonteerd.

De relevante specs:

Voedingsspanning: 5V DC

Signaalspanning (digitale pin): 5V

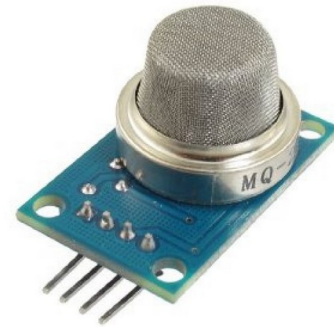
Uitgangsspanning (analoge pin): 0 - 5V

Detecteerbare gassen: Kooldioxide, ammoniak, stikstofoxiden, benzeen, alcohol en rook

Minimale voorverwarmingstijd: 24 uur

De MQ135 heeft een analoge ingang 0-5V_{DC} de Wemos D1 heeft een analoge ingang 0-1V_{DC}.

De A0 ingang van de Wemos D1 mini is verbonden met de ADC ingang van de ESP-12 via een spanningsdeler van 3,2/1, bestaande uit een weerstand van 220kΩ en 100kΩ. deze schakeling zit in de D1 om analoge spanningen 0-3.3V_{DC} te digitaliseren. Met een geschikte voorschakelweerstand kan je deze spanningsdeler aanpassen zodat die voor hogere spanningen werkt. In ons geval om 5V te meten heb ik een 270 kΩ 1/4W weerstand gekozen. Daarmee word de spanningsdeler $(220+270)/100$ of 490k/1k. In principe is dit 10kΩ te weinig, je kan dit oplossen door een tweede weerstand in serie te zetten of, als je een aantal weerstanden van 490k hebt, er eentje te kiezen die aan de hoge kant van zijn tolerantie zit (E12 reeks $\pm 10\%$ $\rightarrow 490k = 441k$ tot 554k).



DS18B20 temperatuur sensor:

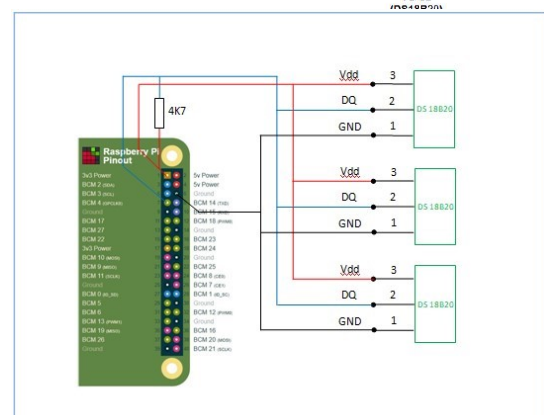
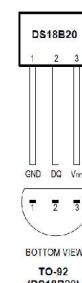
De DS18B20 is een discreet component, het maakt geen deel uit van een breakout board maar kan, samen met 1 extra weerstand per uitleeseenheid, op zichzelf gebruikt worden.

De temperatuur range is -55°C tot 125°C.

De communicatie met de D1 verloopt via het door Dallas Semiconductors ontwikkelde

“one wire” protocol. Het is in principe een tweedraads bus systeem. 1 draad wordt gebruikt voor de communicatie met de diverse componenten op de bus, de tweede draad is enkel een aarde. Elk 1-Wire component heeft een uniek 16 bits adres dat wordt gebruikt om data uit de sensor halen. Als je meerdere DS18B20 sensoren wil gebruiken kan dat. Je hebt maar 1 afsluitweerstand voor de bus nodig en de DQ data pinnen worden in serie met de D1 verbonden.

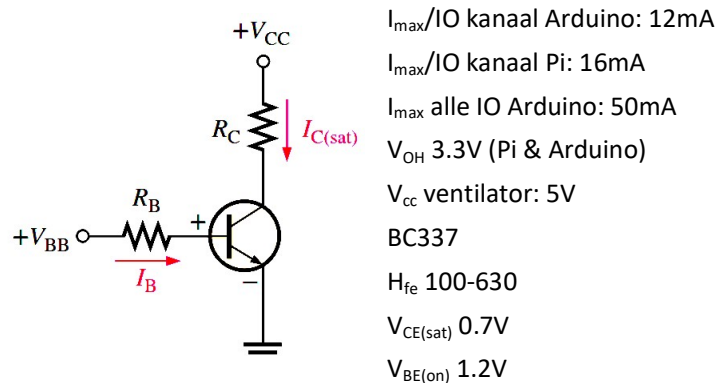
De demo opstelling gebruikt 1 DS18B20 om de



omgevingstemperatuur te meten (de temperatuur sensor in de BME680 meet de temperatuur in de sensor box, niet de omgevingstemperatuur)

Aansturen ventilatoren:

De demo heft twee ventilatoren, eentje om de Pi te kolen en de andere om de sensorbox te spoelen. De D1 en de Pi kunnen niet voldoende stroom leveren via hun uitgangen om de ventilatoren rechtstreeks aan te sturen. De sturing gebeurt via een transistor [BC337](#) als vermogen trap. Het principe schema van een transistor als schakelaar. Rc is de load, in ons geval de ventilator. Rb de basis weerstand beperkt de stroom die de uitgang van de Pi of de D1 moeten leveren.



$$R_{Bmin} = (V_{cc} - V_{CE}) / I_B = (5 - 0.7) / 0.010 = 430 \Omega$$

Voor weerstanden van de E12 reeks is de dichtbijliggende hogere waarde 470Ω om wat marge in te bouwen heb ik gekozen voor **Rb = 560Ω**.

$$I_B = U_{RB} / R_B = (U_{OH} - V_{BE}) / R_B = (3.3 - 1.2) / 560 = 0.00375 \text{ (4mA)}$$

$$P_{RB} = U_{RB} \cdot I_B = (2.1) \cdot 0.00375 = 0.008W \rightarrow \text{weerstand van } \mathbf{1/4W} \text{ zijn ruim voldoende.}$$

Samenbouwen van de componenten:

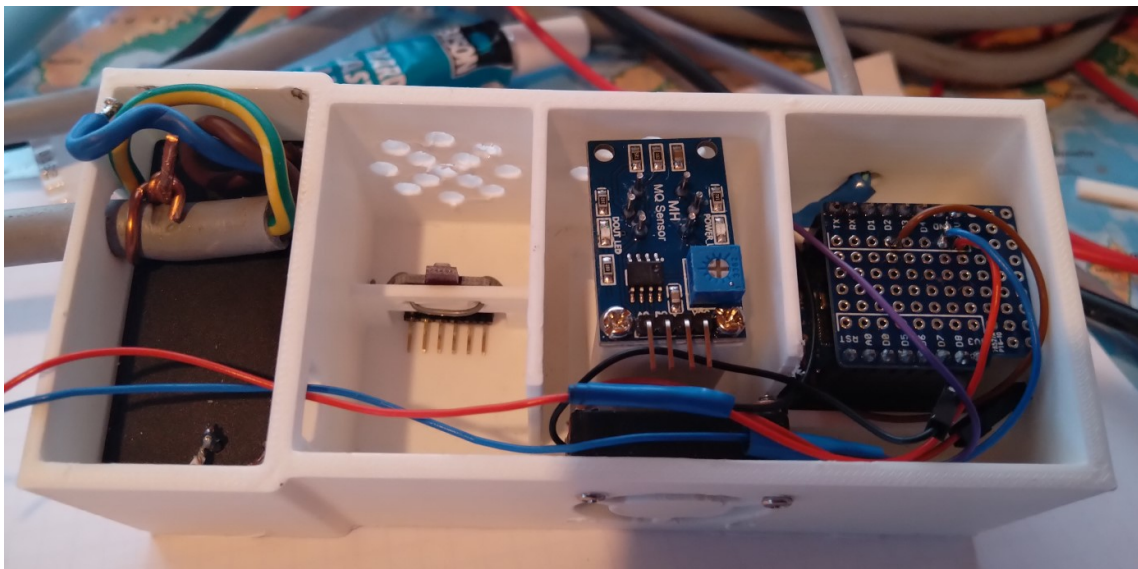
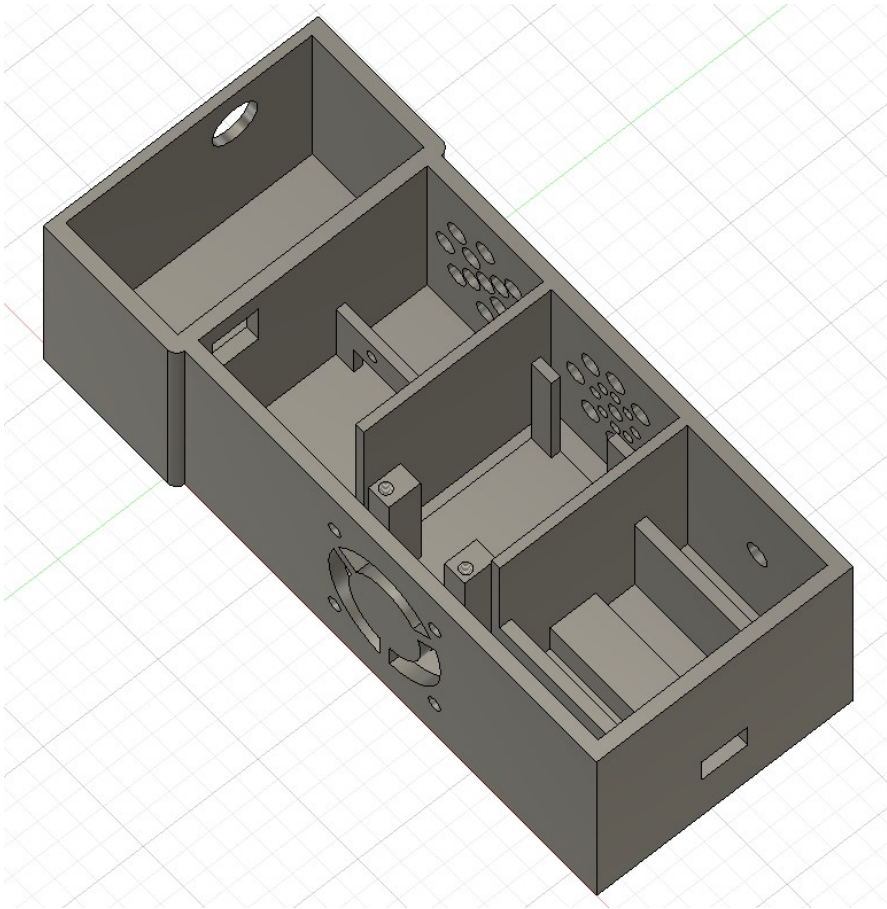
Het samenbouwen vraagt enige handigheid. De demo box is gebouwd op een 3D geprint frame dat netjes in de gebruikte omkasting past. Je kan dit frame gebruiken als dezelfde breakout boards hebt en dezelfde omkasting. Het 3D design is met Fusion 360 gemaakt, je vind het bestand op GitHub.

De componentenlijst omvat een post klein materiaal, dit zijn bv de nodige bouten en moeren, connectoren, draadjes, de weerstanden, experimenteer printje ...

Wat kabelbinders, isolatie tape, krimpkous, solduur, ... bij de hand hebben kan handig zijn.

aanbevolen gereedschap: (kleine) schroevendraaiers, knip-, strip- en langebektang, soldeerbout, multimeter, lat, winkelhaak, boormachine, boren, ...

Bouw foto's:



De source code

C++ op Wemos D1:

De source code is beschikbaar op GitHub.

Het zou te ver leiden om elke regel hier te bespreken. De software wordt inline gedocumenteerd.

De programma secties worden hier wel voorgesteld.

Bibliotheken:

Bibliotheken bevatten source code die door andere (of door u) geschreven is en die door het programma gebruikt gaat worden.

```
1 #include <OneWire.h>
2 #include <DallasTemperature.h>
3 #include <Pinger.h>
4 #include <ESP8266WiFi.h>
5 #include <SPI.h>
6 #include <Ethernet.h>
7 #include <PubSubClient.h>
8 #include <elapsedMillis.h>
9 #include <Wire.h>
10 #include <SPI.h>
11 #include <Adafruit_Sensor.h>
12 #include <Adafruit_BME680.h>
13
```

LET OP, onder Windows 11 kan het zijn dat het OS niet toelaat dat er bibliotheken bij geïnstalleerd worden. Je kan je instellingen aanpassen onder “Virus & threat protection settings”

Declaraties:

In de sectie declaraties worden alle (globale)variabelen die het programma nodig heeft aangemaakt (gedecclareerd)

```
14 //*****Declaraties*****
15 #define ONE_WIRE_PIN D3
16 OneWire oneWire(ONE_WIRE_PIN);
17 DallasTemperature sensors(&oneWire);
18 DeviceAddress Thermometer;
19 Adafruit_BME680 bme; // I2C
20 WiFiClient wifi_client;
21
22 // DEFINE HERE THE KNOWN NETWORKS
23 const char* KNOWN_SSID[] = {"WiFi-2.4-09A8 Accespoint", "WiFi-2.4-09A8"};
24 const char* KNOWN_PASSWORD[] = {"d95uGKd72jY", "d95uGKd72jY"};
25 const int KNOWN_SSID_COUNT = sizeof(KNOWN_SSID) / sizeof(KNOWN_SSID[0]); // number of known networks
26
27 // Ethernet and MQTT related objects
28 EthernetClient ethClient;
29 PubSubClient mqttClient(ethClient);
30 Pinger pinger;
31 int pingResult;
32
33 // global var
34 int NetworkModus = 0; //0 geen netwerk, 1 WiFi SSID, 2 meerdere SSID
35 int deviceCount = 0;
36 float tempC1;
37 float tempC2;
38 char buffer[12]; //the ASCII of the integer will be stored in this char array
39 int beat = 0;
40 int connectionAttemped = 0;
41 int loopCounter = 0;
42 int firstLoop = 0;
43 int mobSensHoldPreBreath = 0;
44 int mobSensBreath = 0;
45 int mobSensHoldPostBreath = 60;
46 int Flush = 1;
47 int mobSensBaseTimeMultiplier = 1;
48 const char* server = "192.168.1.43"; // Make sure to leave out the http and slashes!
49
```

Initialisatie:

De parameters waarmee het programma aan de slag gaat worden hier ingesteld. De delay(10) op het einde pauzeert het gebruikers programma voor 10 msec. Dit geeft het OS van de D1 de kans om de initialisatie correct uit te voeren voordat het programma hervat.


```

50 //*****Initialisatie*****
51 // Set up oversampling and filter initialization
52 bme.setTemperatureOversampling(BME680_OS_8X);
53 bme.setHumidityOversampling(BME680_OS_2X);
54 bme.setPressureOversampling(BME680_OS_4X);
55 bme.setIIRFilterSize(BME680_FILTER_SIZE_3);
56 bme.setGasHeater(320, 150); // 320°C for 150 ms
57
58 delay(10);
59

```

Subroutines:

Een subroutine is een stukje code dat door het hoofdprogramma aangeroepen wordt om specifieke dingen uit te voeren. Je hoeft maar 1 keer te schrijven en kan het zoveel als nodig aanroepen.

Een subroutine definitie start met op te geven welk type variabele terug gegeven word (int voor een integer getal, float voor een komma getal, void voor niets, ... Dan een naam (geen spaties) dan tussen ronde haakjes het type en de naam van wat het programma kan doorgeven aan de subroutine.

De volgende subroutines worden gebruikt:

void wait(int seconds) : wacht lus

void subscribeReceive(char topic, byte* payload, unsigned int length)* : ontvangt MQTTdata

void reconnect(): (her)conecteert de MQTT cliënt

void setup(): is een standaard subroutine van Arduino, deze routine wordt 1 keer uitgevoerd als de Arduino opstart. In het sensorbox programma wordt hier eerste de WiFi connectie en vervolgens de MQTT verbinding opgezet.

void loop(): de hoofdloop van een Arduino programma. Hier worden de metingen gedaan en de resultaten verzonden.

Python code:

De source code is beschikbaar op GitHub.

Het zou te ver leiden om elke regel hier te bespreken. De software wordt inline gedocumenteerd.

Er zijn 3 Python scripts die elke een deel van het werk doen.

hetWeer.py

Dit script haalt de online weer informatie op van het internet en verzend het WAN adres van de internet verbinding naar een Email adres. Dit WAN adres is nodig als je via het internet en portforwarding je Pi wil benaderen van buiten je eigen netwerk.

onlineEvironData.py

Verwerkt de data van de sensor box en haalt de milieu data van het internet

cpuFan.py

Dit script is optioneel, het stuurt de koelventilator van de Pi aan. Je kan opteren om zonder koeling te werken, de ingebouwde controle routine te gebruiken of deze. Het voordeel van *cpuFan.py* is dat je zelf met de parameters kan spelen tot de koeling doet wat jezelf prefereert.

Node Red:

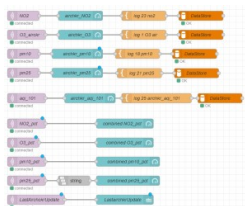
De Node Red flow is beschikbaar op Github.

Je moet enkele nodes bij installeren, dat doe je via het menu Manage Pallet.

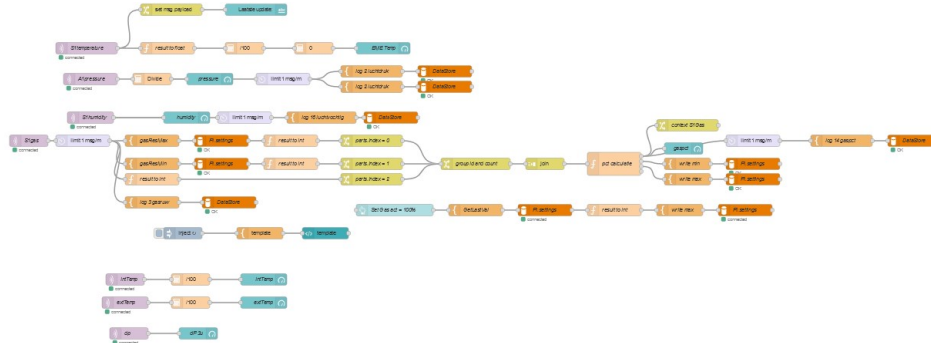
De ontbrekende nodes zijn:

- node-red-node-mysql
- node-red-dashboard
- node-red-contrib-calc
- node-red-contrib-hourglass
- node-red-node-email

In Node Red is programmeren beperkt tot het strikt noodzakelijke. Je koppelt functies aan mekaar en bouwt op die manier de applicatie backend. Meer informatie over Node-Red vindt je online. De front end is een website waarvan je tot op zekere hoogte de lay-out kan bepalen. Sommige flows zijn tamelijk simpel zoals die ik gebruik voor de aircheckr dat die via MQTT binnenkomt in de database te zetten en te visualiseren.

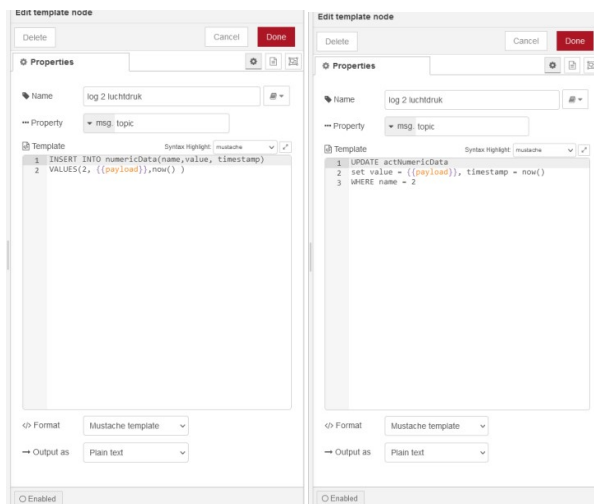


Andere stukken zijn wat uitgebreider zoals bij de BME 680 sensor data die eerst nog wat bewerkt wordt.

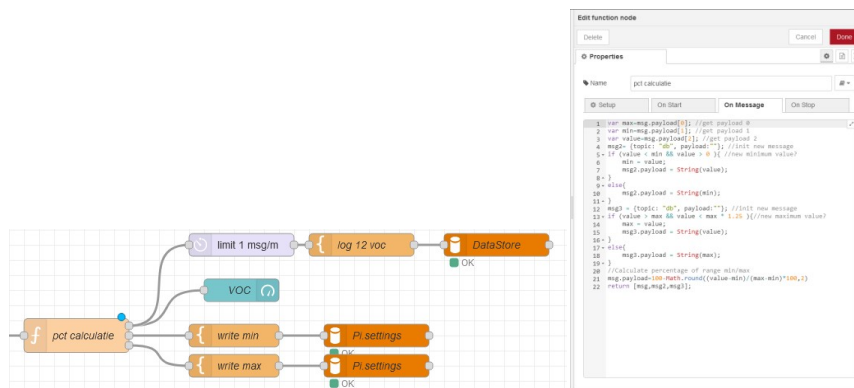


Voorbeelden

Een nieuwe waarde naar de database sturen of een waarde updaten.



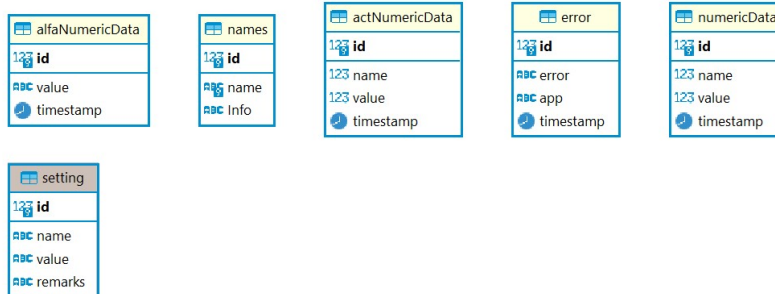
Berekening en logging van percentuele waarde VOC (Volatile Organic Compound)



De database:

MariaDB is een relationele database. Om het zo simpel mogelijk te houden heb ik de relationele functionaliteit hier niet geïmplementeerd.

Het ER diagramma:



De SQL opdrachten vallen buiten de scope van deze handleiding.

Bijlage A technische termen:

SI prefixes v17-18

Prefix	Name	Symbol	Base 10	Decimal	English word		Adoption ^(a) (1)	Etymology	
					Short scale	Long scale		Language	Source word
	yotta	Y	10 ²⁴	1 000 000 000 000 000 000 000 000	septillion	quadrillion	1991	Latin	eight ^(b) (2)
	zetta	Z	10 ²¹	1 000 000 000 000 000 000 000 000	sextillion	trillion	1991	Latin	seven ^(b) (2)
	eka	E	10 ¹⁸	1 000 000 000 000 000 000 000 000	quintillion	trillion	1975	Greek	six
	peta	P	10 ¹⁵	1 000 000 000 000 000 000 000 000	quadrillion	billion	1975	Greek	five ^(b) (2)
	tera	T	10 ¹²	1 000 000 000 000 000 000 000 000	trillion	billion	1960	Greek	four ^(b) (2) monder
	giga	G	10 ⁹	1 000 000 000 000 000 000 000 000	billion	million	1960	Greek	giant
	mega	M	10 ⁶	1 000 000 000 000 000 000 000 000	million		1873	Greek	great
	kilo	k	10 ³	1 000 000 000 000 000 000 000 000	thousand		1795	Greek	thousand
	hecto	h	10 ²	100 000 000 000 000 000 000 000	hundred		1795	Greek	hundred
	deca	da	10 ¹	10 000 000 000 000 000 000 000	ten		1795	Greek	ten
			10 ⁰	1	one			--	
	deci	d	10 ⁻¹	0.1	tenth		1795	Latin	ten
	centi	c	10 ⁻²	0.01	hundredth		1795	Latin	hundred
	milli	m	10 ⁻³	0.001	thousandth		1795	Latin	thousand
	micro	μ	10 ⁻⁶	0.000 001	millionth		1873	Greek	small
	nano	n	10 ⁻⁹	0.000 000 001	billionth	millionth	1960	Greek	dwarf
	pico	p	10 ⁻¹²	0.000 000 000 001	trillionth	billionth	1960	Spanish	peak, a little bit
	femto	f	10 ⁻¹⁵	0.000 000 000 000 001	quadrillionth	billionth	1964	Danish	fifteen, Fermi ^(b) (2)
	atto	a	10 ⁻¹⁸	0.000 000 000 000 000 001	quintillionth	trillionth	1964	Danish	eighteen
	zepto	z	10 ⁻²¹	0.000 000 000 000 000 000 001	sextillionth	trillionth	1991	Latin	seven ^(b) (2)
	yocto	y	10 ⁻²⁴	0.000 000 000 000 000 000 000 001	septillionth	quadrillionth	1991	Latin	eight ^(b) (2)

AC duid op wisselspanning/stroom, DC op gelijkspanning/stroom

U of V duid op een spanning in Volt bv 5V_{DC}, 230V_{AC}

I staat voor stroom in Ampere (A) I_B de stroom naar de basis van de transistor I_{load} de stroom door de belasting. (10mA = 0.01A)

Ω staat voor Ohm, de eenheid van elektrische weerstand het “Ω” teken kan wegelaten worden. 1kΩ = 1 kilo Ohm = 1000Ω, 10E = 10 Ω, 1M = 1 mega Ω, ...

De E12 weerstanden serie in Ω:

- 10, 12, 15, 18, 22, 27, 33, 39, 47, 56, 68, 82,
- 100, 120, 150, 180, 220, 270, 330, 390, 470, 560, 680, 820,
- 1 k, 1.2 k, 1.5 k, 1.8 k, 2.2 k, 2.7 k, 3.3 k, 3.9 k, 4.7 k, 5.6 k, 6.8 k, 8.2 k,
- 10 k, 12 k, 15 k, 18 k, 22 k, 27 k, 33 k, 39 k, 47 k, 56 k, 68 k, 82 k,
- 100 k, 120 k, 150 k, 180 k, 220 k, 270 k, 330 k, 390 k, 470 k, 560 k, 680 k, 820 k,
- 1 M, 1.2 M, 1.5 M, 1.8 M, 2.2 M, 2.7 M, 3.3 M, 3.9 M, 4.7 M, 5.6 M, 6.8 M, 8.2 M,
- 10 M

Bijlage B Inline URL lijst:

Raspberry pi imager: <https://www.raspberrypi.com/news/raspberry-pi-imager-imaging-utility/>

Raspberry pi First boot: <https://raspberrypi-guide.github.io/getting-started/raspberry-pi-configuration>

BC337: <https://www.onsemi.com/pdf/datasheet/bc337-d.pdf>

Pulsbreedtemodulatie: <https://nl.wikipedia.org/wiki/Pulsbreedtemodulatie>

esp8266: <https://www.espressif.com/en/products/socs/esp8266>

Bijlage C Componentenlijst:

- Raspberry Pi 4 Model B 8GB
- Raspberry Pi USB-C Voeding - 5.1V 3A
- Micro SD 128Gb
- Arduino Wemos D1 Mini V4
- BME680 Sensor
- MQ-135 Gas Sensor
- DS18B20 Temperature Sensor
- Optioneel SDS011 Dust Sensor
- Optioneel ADXL345 Digital 3-axis Accelerometer
- IRM-20-5
- kleinmateriaal
- HighPi Raspberry Pi 4 Case
- Box for Arduino
- Fan 30x30x7mm - 5V
- 230V Power supply cable
- Pi HDMI to micro HDMI Cable

Bijlage D Softwarelijst:

- [Raspbian](#) OS voor Raspberry PI
- [Arduino IDE](#) Ontwikkel omgeving Arduino
- [Node-red](#) Dashboard
- [Grafana](#) Data visualisatie
- [MariaDB](#) Database
- [Eclipse Mosquitto](#) MQTT server
- [Dbeaver](#) Database cliënt
- [Visual studio code](#) Ontwikkelomgeving Python

Bijlage E Internet data bronnen:

- openweathermap.org
- <https://www.aircheckr.com>
- <https://www.irceline.be/nl>
- <https://www.irceline.be/tables/actair/actair.php?lan=nl>

Bijlage F Principe schema:

