

Gegevens in een database opslaan

In een eerdere handleiding heb ik laten zien hoe je een database opzet en hoe je gegevens uit de database op een webpagina kunt tonen.

Rest ons een beschrijving van hoe we sensoren kunnen uitlezen en hoe we die waarden in de database kunnen opslaan.

Samen met de eerdere handleidingen heb je zo een overzicht over hoe je een webapplicatie die de gegevens van de sensoren laat zien, in elkaar kunt zetten

Voor het coderen van deze functionaliteit maken we gebruik van python. Tijdens de installatie van MySQL hebben we ook een python-mysql module geïnstalleerd zodat we met behulp van python code ook de database kunnen vullen.

Maak op de Raspberry Pi in de map 'weerstation' een nieuw bestand aan en noem dat 'meting.py'. Vul dit bestand vervolgens met onderstaande code:

```
#!/usr/bin/python
import grovepi
import math
import MySQLdb

dht_sensor_port = 7 # sensor is aangesloten op poort 7
dht_sensor_type = 0 # we gebruiken de blauwe sensor

[ temp,hum ] = grovepi.dht(dht_sensor_port,dht_sensor_type)

if hum is not None and temp is not None:
    db = MySQLdb.connect("localhost", "pi", "admin", "mydb")
    curs = db.cursor()
    try:
        sqlline = "insert into metingen values(NOW(), {0:.0f}, {1:.0f});".format(temp, hum)
        curs.execute(sqlline)
        curs.execute ("DELETE FROM metingen WHERE tijd < NOW() - INTERVAL 180 DAY;")
        db.commit()
        print "Meting toegevoegd aan database"
    except MySQLdb.Error as e:
        print "Fout: de database wordt teruggezet" + str(e)
        db.rollback()
else:
    print "Het is niet gelukt de sensor uit te lezen!"
```

Deze code is eventueel te downloaden via <http://guuskoning.nl/wp-content/uploads/2019/03/weerstation.zip>

In de code wordt eerst de sensor uitgelezen en gecontroleerd of deze geldige waarden bevat. Is dit het geval dan worden die waarden opgeslagen in de tabel 'metingen'. Tevens worden waarnemingen ouder dan 180 dagen uit de database verwijderd. Het idee is dat die gegevens verouderd zijn en er geen behoefte meer is om die in te zien.

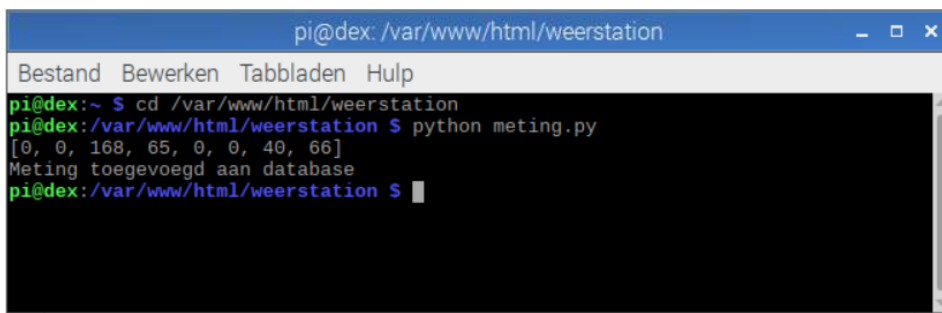
Willen we de code kunnen uitvoeren, dan moeten we het bestand 'meting.py' executie rechten geven. Dit doe je door in een terminal het volgende commando uit te voeren:

chmod +x meting.py

Daarna kun je het programma uitvoeren met het commando:

python meting.py

Als het goed is worden de temperatuur en de luchtvochtigheid nu in de database opgeslagen.



```
pi@dex: /var/www/html/weerstation
Bestand  Bewerken  Tabbladen  Hulp
pi@dex:~ $ cd /var/www/html/weerstation
pi@dex:/var/www/html/weerstation $ python meting.py
[0, 0, 168, 65, 0, 0, 40, 66]
Meting toegevoegd aan database
pi@dex:/var/www/html/weerstation $
```

Met phpMyAdmin kunnen we even in de database zelf kijken.

Zoals je ziet is er één waarneming opgeslagen.

Willen we regelmatig een waarneming doen, dan kunnen we steeds handmatig het programma 'meting.py' uitvoeren, maar veel handiger is het om dit te automatiseren.

In het besturingssyteem van de Raspberry Pi is er een speciale tabel waarin alle taken staan die met een bepaalde regelmatig op de achtergrond gedraaid moeten worden.

Met het commando **crontab -e** open je deze tabel en kun je er een commando aan toevoegen. Voer dit commando uit en voeg aan de tabel de volgende regel toe:

`*/5 * * * * sudo python /var/www/html/weerstation/meting.py`

Deze regel zorgt er voor dat de meting om de vijf minuten wordt uitgevoerd. Sla het bestand op en wacht een tijdje voordat je weer in de database gaat kijken. Afhankelijk van hoelang je gewacht hebt moet je nu meerdere metingen in de database zien verschijnen.

Ik heb een kwartier gewacht en je ziet dat er om de vijf minuten een meting wordt gedaan.