



Oefeningen chapter 3

Rapport

IoT Advanced ITFactory

Frederik Crauwels

Academiejaar 2021-2022

Campus Geel, Kleinhoefstraat 4, BE-2440 Geel

INHOUDSTAFEL

Contents

INHOUDSTAFEL	3
1 OEFENINGEN CHAPTER 3 SQL	4
1.1 Exercise SQL – Light & temperature.....	4
1.1.1 Oefening 1	4
1.1.2 Oefening 2 (1): Lees data van database en toon op lokale webserver:	7
1.1.3 Oefening 3 (2): @home en dynDNS (NGROK).....	8
1.1.4 Foto's van de opstelling (voor alle oefeningen):	16

1 OEFENINGEN CHAPTER 3 SQL

1.1 Exercise SQL – Light & temperature

1.1.1 Oefening 1

Voor de eerste oefening meten we simpelweg licht en temperatuur. Ik heb echter een DHT22 i.p.v. de standaard temperatuur meter. Deze is veel nauwkeuriger en heeft ook een vochtigheidsmeter (vorig jaar aangekocht, aanbevolen door IoT Essentials leerkracht).

Voor deze oefening heb ik een uitgebreid python bestand geschreven aangezien we soortgelijke oefeningen vorig jaar gemaakt hebben – nu met de SQL extentie. **Foto's van de setup zelf zie je op de laatste pagina's.**

```
Chapter3_3.py > readadc
1  import cgitb ; cgitb.enable()
2  import spidev
3  import time
4  import busio
5  import digitalio
6  import board
7  import requests
8  from adafruit_bus_device.spi_device import SPIDevice
9  import adafruit_dht
10 import RPi.GPIO as GPIO
11 from tabulate import tabulate
12 from datetime import datetime
13 import MySQLdb
14
15
16 # Initialize SPI bus
17 spi = busio.SPI(board.SCK, MOSI=board.MOSI, MISO=board.MISO)
18
19 # Initialize control pins for adc
20 cs0 = digitalio.DigitalInOut(board.CE0) # chip select
21 adc = SPIDevice(spi, cs0, baudrate= 1000000)
22
23 #temperature and humidity
24 dhtDevice = adafruit_dht.DHT22(board.D4, use_pulseio=False)
25
26 # read SPI data 8 possible adc's (0 thru 7)
27 def readadc(adcnum):
28     if ((adcnum > 7) or (adcnum < 0)):
29         return -1
30     with adc:
31         r = bytearray(3)
32         spi.write_readinto([1,(8+adcnum)<<4,0], r)
33         time.sleep(0.00005)
34         adcout = ((r[1]&3) << 8) + r[2]
35         return adcout
36
37 # Initiate the DB
38 database = MySQLdb.connect(host="localhost", user="pi", passwd="raspberry",db="mydb")
39
40 #database select
41 cursor = database.cursor()
42
43
44 try:
45     while True:
46         try:
47             #light sensor
48             tmp0 = readadc(1)
49             light = round((tmp0 / 1023) * 100)
50
51             #temperature and humidity
52             temperature_c = dhtDevice.temperature
```

```

13 temperature_f = round((temperature_c * 9 / 5) + 32)
14 humidity = dhtDevice.humidity
15
16 #table with data
17 time.sleep(1)
18 print()
19 print(f"Amount of light: {light, 'N'}, Temperature in C°: {temperature_c, 'C'}, Temperature in Fahrenheit: {temperature_f, 'F'}, Humidity: {humidity, 'N'}, [What is being displayed, 'The amount', 'Data type']")
20 cursor.execute(INSERT INTO Licht_temp(Temperature, Humidity, Light) VALUES(%s, %s, %s)", (temperature_c, humidity, light))
21 database.commit()
22 # Sometimes our (temperature and humidity) will run into errors.
23 # This will prevent the program from stopping.
24 # and the program will continue to run afterwards.
25 except RuntimeError as error:
26     print()
27     print("An error has occurred.")
28     print(error.args[0])
29     print("A new table will be printed soon.")
30     print()
31     time.sleep(1)
32     continue
33 except KeyboardInterrupt:
34     print()
35     print("The program has been interrupted.")
36     print("This is the end of the program!")
37 # At the end you can stop the program with Ctrl+C
38 # and it will display an "end of program" message.

```

17/10 extra toepassing:

Automatisch verwijderen van de database gegevens van de openstaande sessie. De lijst wordt ontzettend lang; na afsluiten van het programma (KeyboardInterrupt) wordt de huidige DB leeggemaakt:

```

except KeyboardInterrupt:
    #Clean up DB when exiting the python loop
    cursor.execute("DELETE FROM Licht_temp")
    database.commit()
    GPIO.cleanup()
    print()

```

Nu krijg je dus na afsluiten een clean page:

<https://8b30-84-197-9-22.ngrok.io>

Temperature	Humidity	Light (%)
-------------	----------	-----------

Met dit script krijg je in de CLI een mooie representatie van de data:

```

pi@raspberrypi:~/IotAdvanced $ python3 Chapter3_3.py
What is being displayed      The amount  Data type
-----
Amount of light:             26          %
Temperature in C°:           21.2         ° , C
Temperature in Fahrenheit:    70           F
Humidity:                     69.3         %

What is being displayed      The amount  Data type
-----
Amount of light:             26          %
Temperature in C°:           21.2         ° , C
Temperature in Fahrenheit:    70           F
Humidity:                     69.3         %

What is being displayed      The amount  Data type
-----
Amount of light:             26          %
Temperature in C°:           21.2         ° , C
Temperature in Fahrenheit:    70           F
Humidity:                     72.2         %
^C
The program has been interrupted.
This is the end of the program!_

```

Tot zo ver het script en de CLI weergave van dit script.

In het bovenstaande script zicht ook al een verbinding naar de SQL server waar deze data ook binnenkomt (behalve Fahrenheit):

1.1.2 Oefening 2 (1): Lees data van database en toon op lokale webserver:

Zie vorige oefening voor uitwerking. Het enige wat ons nog te wachten staat is een PHP webpagina die de data dynamisch inleest vanuit de database en deze ziet er zo uit:

```
html > index.php
1  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
2  <html>
3  <body>
4  <?php
5
6  $hostname = "localhost";
7  $username = "pi";
8  $password = "raspberry";
9  $db = "mydb";
10
11 $dbconnect=mysqli_connect($hostname,$username,$password,$db);
12
13 if ($dbconnect->connect_error) {
14     die("Database connection failed: " . $dbconnect->connect_error);
15 }
16
17 ?>
18
19 <table border="1" align="center">
20 <tr>
21     <td>Temperature</td>
22     <td>Humidity</td>
23     <td>Light (%)</td>
24 </tr>
25
26 <?php
27
28 $query = mysqli_query($dbconnect, "SELECT * FROM Licht_temp")
29     or die (mysqli_error($dbconnect));
30
31 while ($row = mysqli_fetch_array($query)) {
32     echo
33     "<tr>
34         <td>{$row['Temperature']}</td>
35         <td>{$row['Humidity']}</td>
36         <td>{$row['Light']}</td>
37     </tr>\n";
38 }
39
40 ?>
41
42 </table>
43 </body>
44 </html>
```

← → ↻ ⚠ Not secure | 192.168.60.254

21	71	29
21	71	29
21	71	29
21	71	29
21.2	70.7	29
21.2	70.7	29
21.2	70.7	29
21.2	70.7	29
21.2	70.7	29
21.2	70.7	29
21.2	70.7	29
21.2	70.7	29
21.2	70.7	29
21.2	70.7	29
21.2	70.7	29
21.2	70.7	29
21.2	70.7	29
21.2	70.7	29
21.2	70.7	29
21.2	70.7	29
21.2	70.7	29
21.2	69.3	26
21.2	72.2	26

1.1.3 Oefening 3 (2): @home en dynDNS (NGROK).

Port-forwarding problemen: Na het bekijken van deze opdracht kan ik in eerste instantie niet aanraden om port forwarding te gebruiken in de huidige context. Alsnog worden er vandaag meer dan voldoende alternatieven aangeboden om **zonder port forwarding** de opdracht succesvol uit te voeren en testen.

- **Security:** Port forwarding is veilig, enkel en alleen als je sterke wachtwoorden / logins gebruikt en alles consistent up to date houdt. Daarnaast moet je ook voorzien dat jouw netwerk veilig wordt gesteld door behulp van een firewall of recente router (met firewall ingebouwd).
 - **Login:** In deze (en IoT Essentials) hebben wij gebruik gemaakt van een zéér standaard login: username: pi, password: xxx. Alleen al in deze context is port forwarding een absolute no-go. Deze login bevat basic credentials zonder extra beveiligingen. Daarnaast is er ook geen 2FA of OAUTH ter beschikking waardoor extra security measures niet beschikbaar zijn. Er bestaan botnets / scrapers die actief op zoek gaan en scannen naar open poorten (waaronder port 80).
 - **Consistent upgraden:** Ik kwam er deze avond nog eens zelf achter dat ik mijn PI niet meer recent heb geupdate + geupgrade. De cursus raadt consistent aan om te **updaten, maar niet om te upgraden**. Een dist-upgrade wordt bijna nooit uitgevoerd, wat weer extra risico's meebrengt indien je jouw raspberry pi openlijk toegankelijk maakt.

- **Netwerkveiligheid:** het thuisnetwerk is vaak ook niet extra beveiligd met ACLs of een eigen firewall, bijvoorbeeld. Vaak wordt dit door de ISP gedaan.
In mijn geval is dat telenet, die daarom ook heeft besloten om port forwarding niet meer toe te laten voor iedereen en bepaalde poorten automatisch af te sluiten (voor eigen veiligheid, uiteraard). Telenet blokkeert vandaag de dag standaard alle poorten van 0 tot 1023, waaronder dus ook poort 80 en poort 443. **Zonder een Business internet (met vast IP) abonnement kan je daar dus niet aan,** en met een goede reden ook... security!
- **Kotstudent:** als kotstudent heb je simpelweg geen toegang tot de router. Ook als student op locatie in de campus heb je dat niet. Daarom wordt deze opdracht onzettend hard gelimiteerd, wat zeker niet zo hoeft te zijn (of blijven). Als je dan nog enkel een klassiek telenet abonnement hebt thuis loop je tegen veel lampen aan en wordt de opdracht onuitvoerbaar.

Om deze bovenstaande redenen heb ik een alternatieve oplossing gevonden die specifiek gemaakt is om bepaalde toepassingen te **testen**. Ik ga hierbij gebruik maken **van een secure tunnel**. Deze tool is **NGROK**. Het nadeel is dus dat je zowel je NAT als je firewall omzeilt (die waarschijnlijk toch onbeheerd zijn thuis en waar 99% van de mensen geen idee van heeft). NGROK kan nog **veel meer** aanbieden dan enkel de mogelijkheid om deze opdracht succesvol uit te voeren zonder onnodige security risks aan te gaan. Daarnaast is het ook een vrijblijvende oplossing, het is gratis met een trial periode. Ook **PiTunnel** zou hier een alternatief kunnen zijn.

Ook hier weer blijft weer opnieuw dezelfde boodschap: its never safe.

[Abusing Ngrok: Hackers at the End of the Tunnel \(huntress.com\)](https://www.huntress.com/abusing-ngrok-hackers-at-the-end-of-the-tunnel)

Meer documentatie over NGROK: [ngrok - secure introspectable tunnels to localhost](https://ngrok.com/docs)

1.1.3.1 De opdracht

Alles van de bovenstaande oefeningen is opnieuw van toepassing zonder enige veranderingen. Index.php is de standaard go-to webpagina die je krijgt als je op het IP adres gaat zoeken. De data en het script lopen automatisch vanaf je deze opent in de CLI (zie *bovenstaande pagina's dus voor de bestanden*).

Zo krijg je dus het volgende:

1. Installatie van NGROK is uitgevoerd, je hebt het standaard NGROK bestand nu op jouw raspberry pi.

```
pi@raspberrypi:~/ngrok $ ls -all
total 25384
drwxr-xr-x  2 pi pi    4096 Oct 17 05:45 .
drwxr-xr-x 40 pi pi    4096 Oct 17 05:46 ..
-rwxr-xr-x  1 pi pi 25983364 May  4 16:20 ngrok
pi@raspberrypi:~/ngrok $
```

2. Nu ga je gebruik maken van deze tool om de tunnel op te starten:

```
./ngrok http 80
pi@raspberrypi:~/ngrok $ ./ngrok http 80
```

Dit maakt een speciale tunnel voor jou, via NGROK, om de poort 80 open te stellen; jouw webpagina dus. Indien je een trial account hebt gaat deze URL er altijd anders uit zijn. Je krijgt geen statisch IP, dus moet je even kijken op de website zelf om het IP te achterhalen, of je kijkt via de CLI.

CLI:

```

pi@raspberrypi: ~/ngrok
File Edit Tabs Help
ngrok by @inconshreveable (Ctrl+C to quit)

Session Status      online
Account             Frederik Crauwels (Plan: Free)
Version             2.3.40
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           http://7911-84-197-9-22.ngrok.io -> http://localhost:80
Forwarding           https://7911-84-197-9-22.ngrok.io -> http://localhost:80

Connections          ttl    opn    rt1    rt5    p50    p90
0                  0      0.00   0.00   0.00   0.00

```

NGROK dashboard:

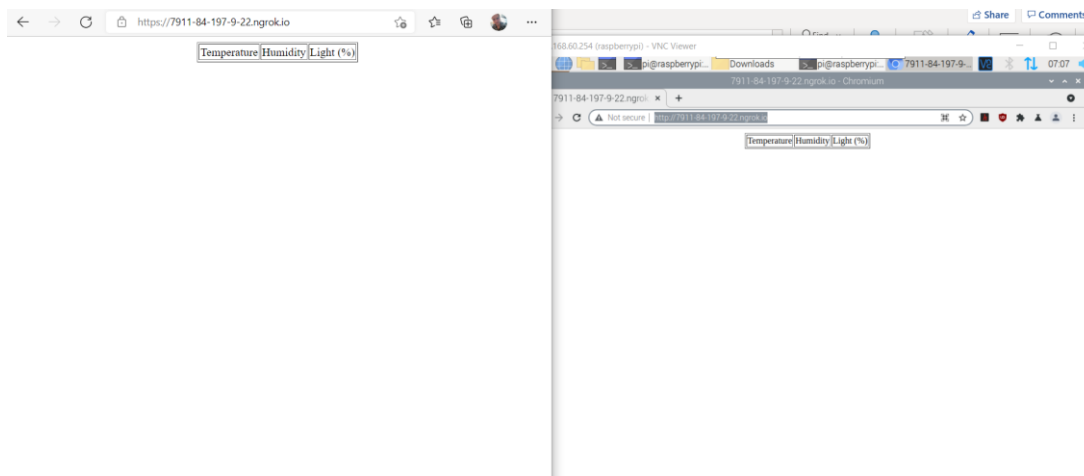
Online Tunnel Endpoints

Filter tunnels...

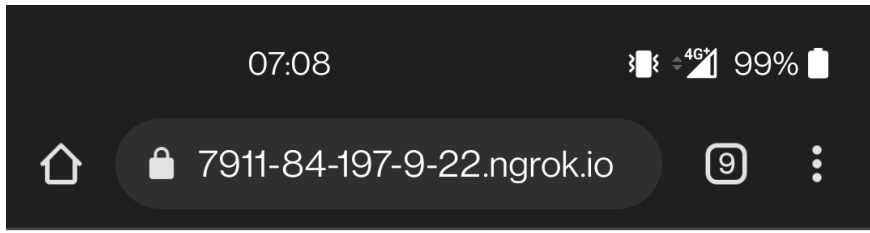
Region	URL	Client IP	Established
US	http://7911-84-197-9-22.ngrok.io	84.197.9.22	<1m ago
US	https://7911-84-197-9-22.ngrok.io	84.197.9.22	<1m ago

NGROK opent altijd zowel poort 80 als poort 443 voor jouw toepassingen.

- Kijk nu de link na in jouw internetbrowser. Waar je maar wil, wanneer je maar wil:

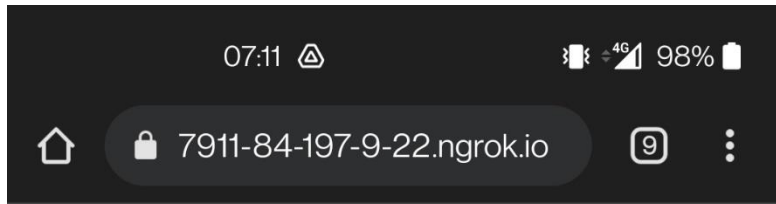


Op host staat een HTTPS browser open, op pi staat een HTTP browser open.
GSM via 4G (buiten mijn local netwerk):



Temperature	Humidity	Light (%)
-------------	----------	-----------

[illegible]

GSM (4G) – buiten lokale netwerk:

Temperature	Humidity	Light (%)
19.6	65.7	1
19.6	65.7	1
19.6	66.4	1
19.6	66.4	1
19.6	66.5	1
19.6	66.5	1
19.6	66.5	1
19.6	66.5	1
19.6	66.5	1
19.6	66.5	1
19.6	65	1
19.6	65	1
19.5	66.4	1
19.5	66.4	1
19.5	66.5	1
19.5	66.5	1
19.5	66.4	1
19.5	66.4	1
19.5	66.3	1
19.5	66.3	1
19.5	66.4	1
19.5	66.4	1
19.5	66.4	1
19.5	66.4	1
19.5	66.4	1
19.5	66.4	1
19.6	66.4	1
19.6	66.4	1
19.6	66.4	1
19.6	66.4	1
19.6	66.4	1
19.5	66.3	1
19.5	66.3	1
19.5	66.3	1
19.5	65	1
19.5	65	1
19.5	66.3	1
19.5	66.3	1
19.5	66.4	1
19.5	66.4	1
19.6	66.4	1
19.6	66.4	1
19.6	66.5	1
19.6	66.5	1
19.5	66.4	1
19.5	66.4	1
19.5	66.4	1
19.5	66.4	1
19.6	66.4	1
19.6	66.4	1
19.6	66.4	1
19.6	66.4	1
19.5	66.4	1

Hierbij kan je ook meekijken wanneer er iemand connecteert, bijvoorbeeld:

```

pi@raspberrypi: ~/ngrok
File Edit Tabs Help

ngrok by @inconshreveable

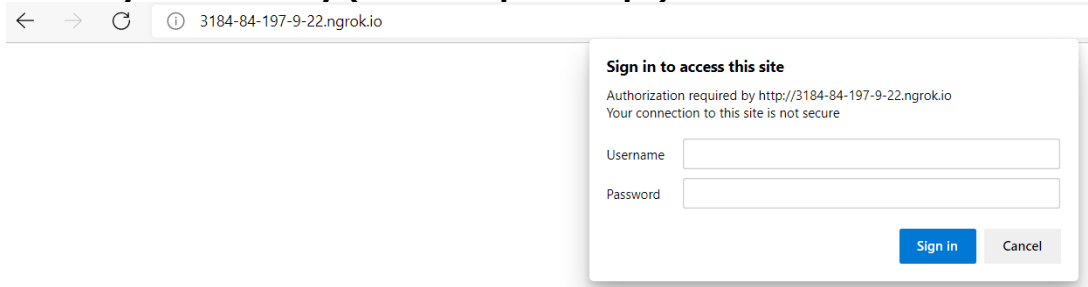
Session Status      online
Account             Frederik Crauwels (Plan: Free)
Version             2.3.40
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           http://7911-84-197-9-22.ngrok.io -> http://localhost:80
Forwarding           https://7911-84-197-9-22.ngrok.io -> http://localhost:80

Connections
  ttl    opn    rt1    rt5    p50    p90
  10      0     0.00   0.00   5.14   8.06

HTTP Requests
-----
GET /                200 OK
GET /                200 OK
GET /                200 OK
GET /                200 OK
GET /favicon.ico     404 Not Found
GET /                200 OK
GET /                200 OK
GET /                200 OK
GET /favicon.ico     404 Not Found
GET /                200 OK

```

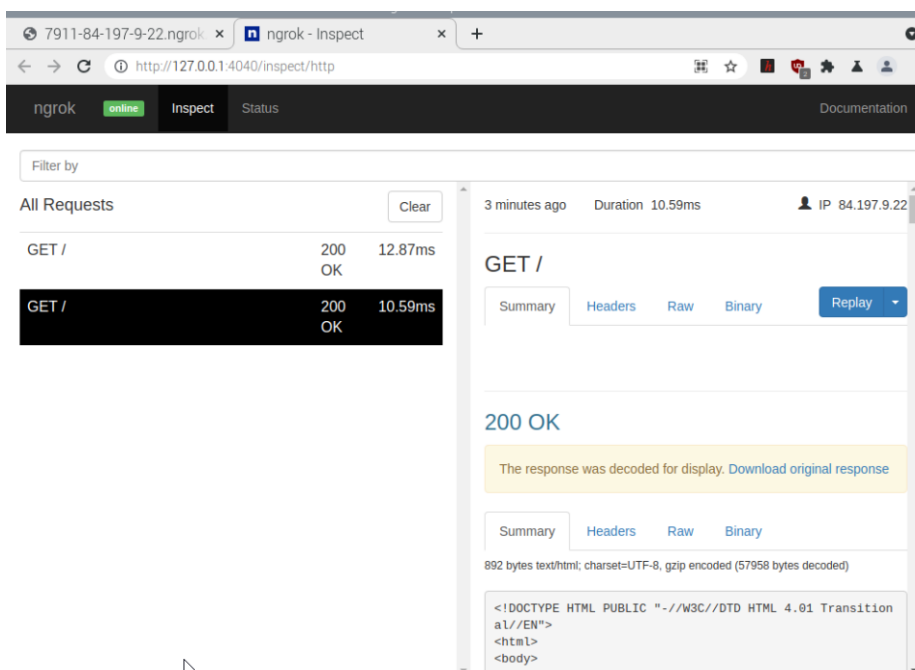
Je kan ook al basic authentication voor de webpagina meegeven, een extra layer in security (zowel http als https):



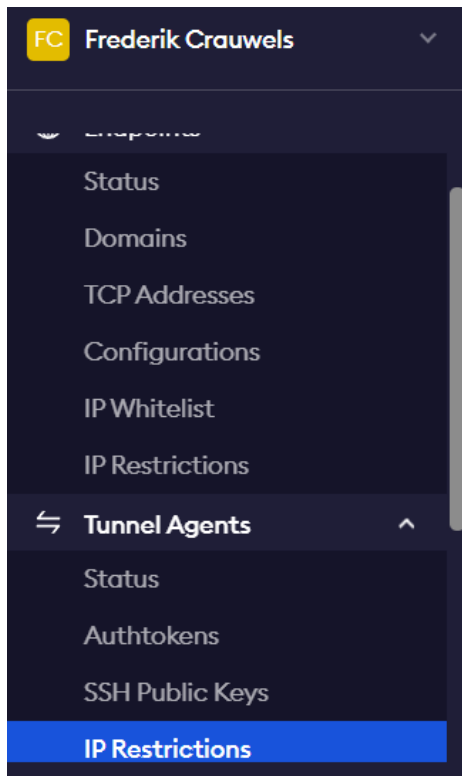
Geen goed wachtwoord:

401 Unauthorized

Lokaal heb je ook toegang tot het NGROK dashboard:



Dit, en vele andere mogelijkheden maken van NGROK een belangrijke en handige tool voor huidige en toekomstige IoT projecten. Extra beveiliging kan je ook toevoegen, indien je hier toegang tot hebt (momenteel beta / onbetaald account). Zo heb je meer controle over jouw toepassingen. **Safety first:**



1.1.4 Foto's van de opstelling (voor alle oefeningen):

Houd geen rekening met de opstelling aan de linkerkant met de potentiometers. Dit is voor een andere oefening (volgend hoofdstuk).

