

Dokumentation – Projekt Influx

1 Inhaltsverzeichnis

1. Einführung.....	2
1.1 Voraussetzungen.....	2
1.2 Zielsetzung.....	2
2 Influx Datenbank.....	2
2.1 Beschreibung.....	2
3 Ablaufbeschreibung.....	3
3.1 InfluxDb installieren.....	3
3.1.1 Repository hinzufügen.....	3
3.1.2 Installieren und starten.....	3
3.1.3 Verbindung aufbauen.....	3
3.1.4 Testdaten einfügen.....	3
3.1.5 Datenbank erstellen.....	3
3.2 NodeRed Anpassungen.....	4
3.2.1 NodeRed starten.....	4
3.2.2 InfluxDb Schnittstelle hinzufügen.....	4
3.3 Temperatur-Skript Anpassungen.....	5
4 Testen.....	7
5 Fazit.....	8

1. Einführung

1.1 Voraussetzungen

Für eine erfolgreiche Umsetzung des Projekts benötigt man:

- Fertig installierte SD-Karte mit Mosquitto und Bibliotheken zum Auslesen der Sensordaten des DHT22-Sensors und MQTT mit Python
- Alle bisherigen Skripte um Temperatursensor auszulesen und via MQTT die Daten zu übermitteln

1.2 Zielsetzung

Die Daten von der Temperaturmessung sollen in einer Influx Datenbank gespeichert werden.

Die gespeicherten Daten sollen grafisch dargestellt werden.

2 Influx Datenbank

2.1 Beschreibung

InfluxDB ist eine Open-Source-Datenbank, die speziell für die Speicherung und Abfrage von Zeitreihendaten konzipiert ist. Sie wird oft für Metriken, Ereignisse, Sensordaten oder andere Daten verwendet, die über einen Zeitraum hinweg erfasst werden.

InfluxDB ist eine beliebte Wahl für die Überwachung von Systemen, die Analyse von IoT-Daten und die Echtzeit-Analyse von Daten.

3 Ablaufbeschreibung

3.1 InfluxDb installieren

3.1.1 Repository hinzufügen

Zu Beginn muss das Archiv von InfluxDb hinzugefügt werden.

```
paddi@vin:~$ curl -s https://repos.influxdata.com/influxdata-archive_compat.key > influxdata-archive_compat.key
echo '393e8779c89ac8d58f81f942f9ad7fb82a25e133faddaf92e15b16e6ac9ce4c influxdata-archive_compat.key' | sha256sum -c && cat influxdata-archive_compat.key | gpg --dearmor | sudo tee /etc/apt/trusted.gpg.d/influxdata-archive_compat.gpg > /dev/null
echo 'deb [signed-by=/etc/apt/trusted.gpg.d/influxdata-archive_compat.gpg] https://repos.influxdata.com/debian stable main' | sudo tee /etc/apt/sources.list.d/influxdata.list
```

Abb. 1: Repository hinzufügen

3.1.2 Installieren und starten

Nun kann InfluxDb installiert und der Service daraufhin gestartet werden.

```
paddi@vin:~$ sudo apt-get update && sudo apt-get install influxdb
sudo service influxdb start
```

Abb. 2: Installieren und starten

3.1.3 Verbindung aufbauen

Über die Hostadresse wird die Verbindung aufgebaut.

```
paddi@paddispi:~$ influx
Connected to http://localhost:8086 version v1.11.8
InfluxDB shell version: v1.11.8
```

Abb. 3: Verbinden

3.1.4 Testdaten einfügen

Zur prüfung der Installation werden Testdaten eingefügt.

```
paddi@paddispi:~$ curl https://s3.amazonaws.com/noaa.water-database/NOAA_data.txt -o NOAA_data.txt
```

Abb. 4: Testdaten runterladen

```
paddi@paddispi:~$ influx -precision rfc3339 -database NOAA_water_database
```

Abb. 5: Testdaten einspeisen

```
paddi@paddispi:~$ influx -precision rfc3339 -database NOAA_water_database
Connected to http://localhost:8086 version v1.11.8
InfluxDB shell version: v1.11.8
> show measurements
name: measurements
name
--
average_temperature
h2o_feet
h2o_ph
h2o_quality
h2o_temperature
>
```

Abb. 6: Testdaten anzeigen

3.1.5 Datenbank erstellen

Es wird eine Datenbank mit dem Namen „messungen“ angelegt.

```
paddi@paddispt:~$ influx
Connected to http://localhost:8086 version v1.11.8
InfluxDB shell version: v1.11.8
> create database messungen
```

Abb. 7: Datenbank erstellen

3.2 NodeRed Anpassungen

3.2.1 NodeRed starten

Nachdem die Datenbank aufgesetzt ist kann, falls nicht bereits gemacht, NodeRed gestartet werden.

```
paddi@paddispt:~$ node-red start
```

Abb. 8: NodeRed starten

3.2.2 InfluxDb Schnittstelle hinzufügen

In NodeRed wird eine Schnittstelle für InfluxDb bereitgestellt.

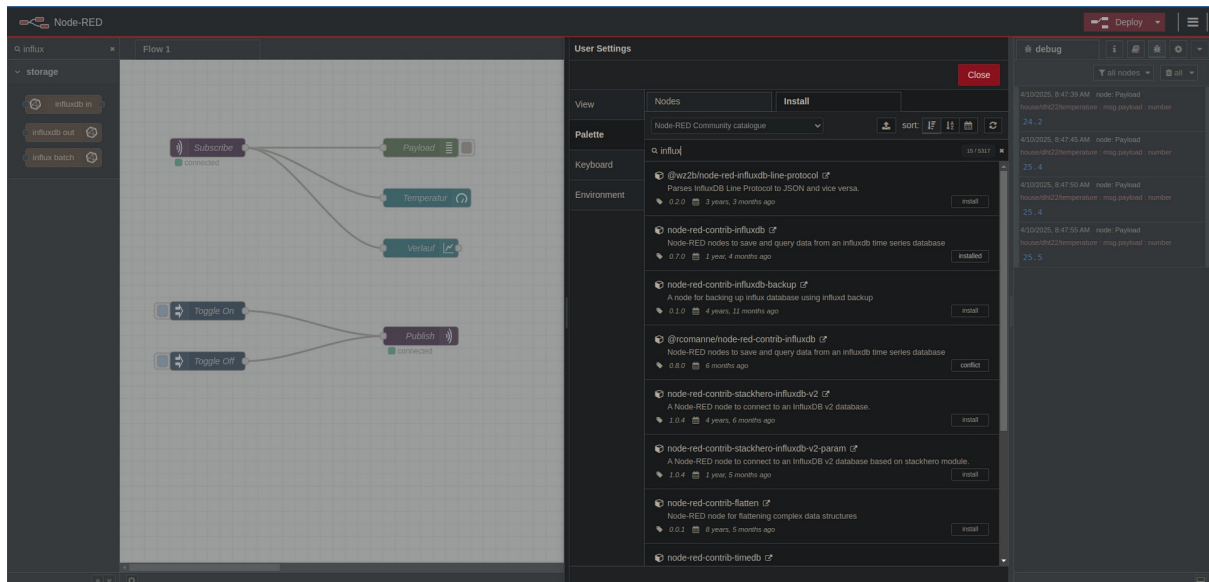


Abb. 9: Schnittstelle in NodeRed

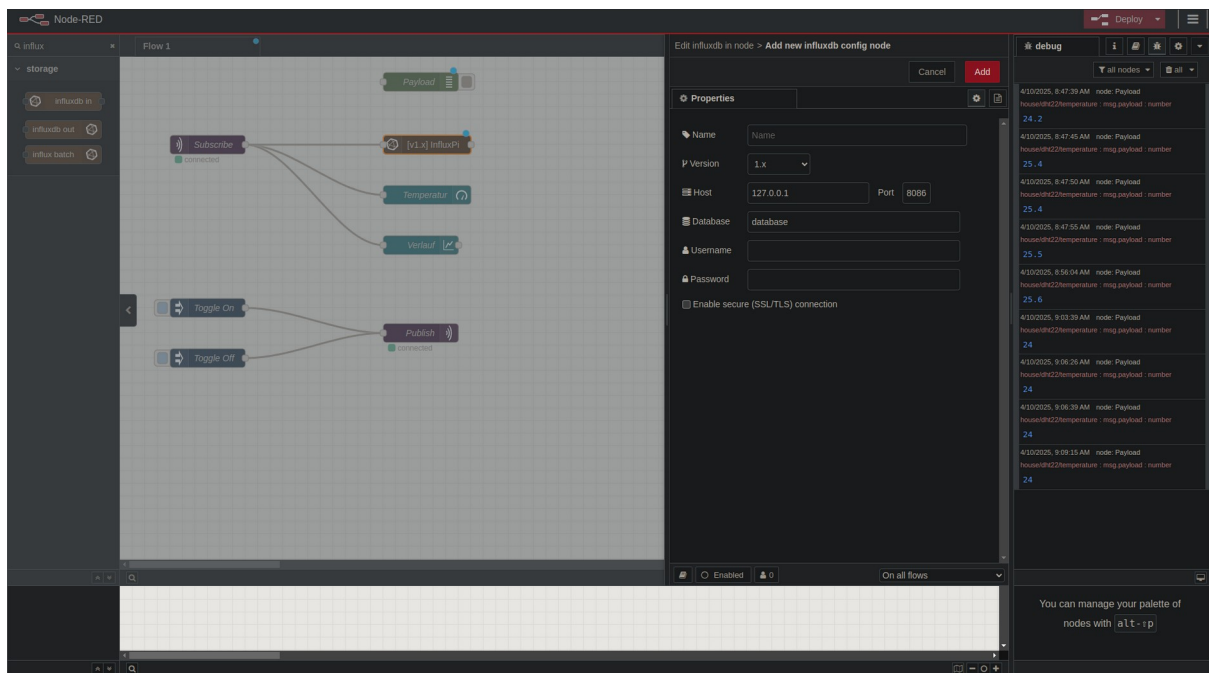


Abb. 10: Verbindungsaufbau mit InfluxDb

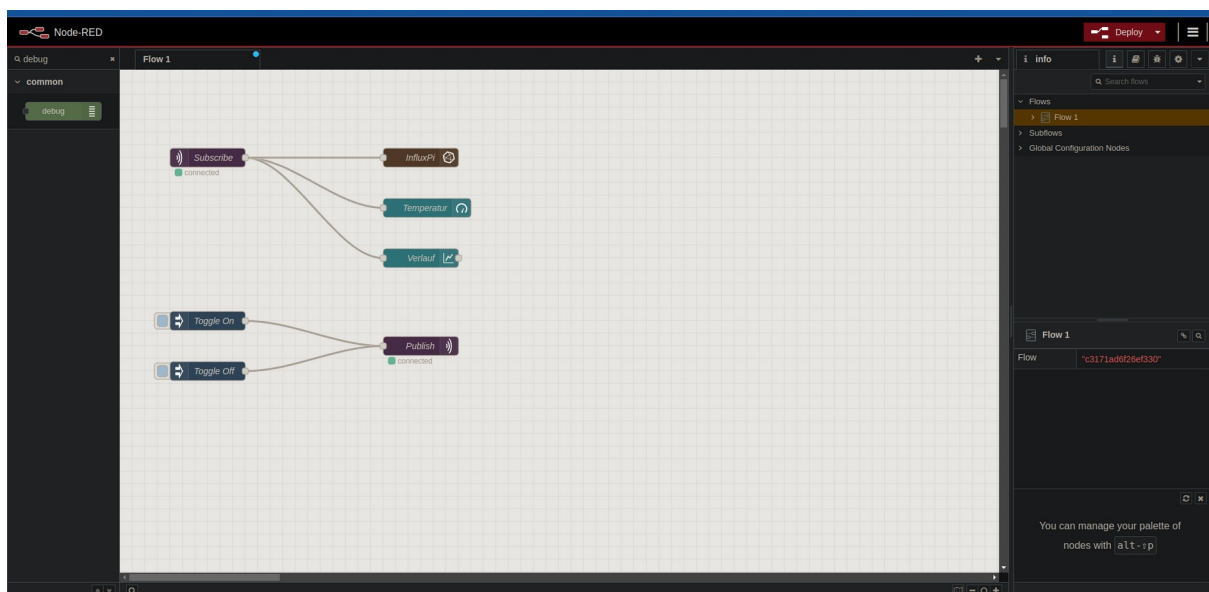


Abb. 11: InfluxDb Schnittstelle bereitgestellt

3.3 Temperatur-Skript Anpassungen

Nachdem die Schnittstelle in NodeRed bereitgestellt ist, wird das Temperatur-Skript angepasst, sodass die Daten in einem JSON-Format ausgegeben werden. Dies ermöglicht ein direktes schreiben der Daten in die InfluxDb Datenbank „messungen“.

```
GNU nano 7.2 dht22-mqtt.py
print("connect failed");

Connected = False #global variable for the state of the connection

#MQTT Connection
client=mqtt.Client("control1")
#client.username_pw_set(username, password)
#client.tls_set(cert_reqs=ssl.CERT_NONE) #no client certificate needed
#client.tls_insecure_set(insecure)
client.on_connect=on_connect
client.connect(broker, port=port)
client.loop_start()
print("hallo")
while True:
    try:
        #print("hallo")
        temperature, humidity = dhtDevice.temperature, dhtDevice.humidity
        data = json.dumps({"temperature":temperature, "humidity":humidity})
        #client.publish("{}temperature".format(publish_topic), "{:.1f}".format(temperature), qos)
        #nur zum Test
        #client.publish("test", "123")
        #client.publish("{}humidity".format(publish_topic), "{}".format(humidity), qos, retain_message)
        client.publish("{}sensor_daten".format(publish_topic), data)
        print(data)
        #print("{}temperature".format(publish_topic), "{:.1f}".format(temperature))
    except RuntimeError as error:
        print(error.args[0])
        time.sleep(2.0)
    except KeyboardInterrupt:
        client.disconnect()
        client.loop_stop()
        time.sleep(5)
```

Abb. 12: Daten als JSON asugeben

4 Testen

Sind die Anpassungen erfolgreich, werden die Daten von der Temperaturmessung automatisch in die entsprechende Datenbank geschrieben.

Mit einem Select-Befehl können diese angezeigt werden.

```
node@paddlapi: /src/iot/001-temperatur $ influx
Connected to http://localhost:8086 version v1.11.8
InfluxDB shell version: v1.11.8
> use messungen
Using database messungen
> select * from sensor_data
name: sensor_data
time                humidity temperature
----                -
1744270177882351465 32.6      23.9
1744270183126693935 32.2      24
1744270195639331205 33.2      23.9
1744270200909163889 32.4      24
1744270206157473859 32.2      23.9
1744270211417328389 32.1      24
1744270216674024128 32.1      24
174427021933459639 32      24
1744270227194464793 32      24
1744270232461411826 32      24
1744270237726738247 32      24
1744270242976753353 31.8      24
1744270255493489392 31.9      24
1744270260754852377 31.9      24
1744270266013130269 32.6      24
1744270271276314922 32.6      24
1744270276537362569 32.4      24
1744270281792143774 33.2      24
>
```

Abb. 14: Messungsdaten anzeigen

Die Tests waren erfolgreich. InfluxDb wurde korrekt in NodeRed integriert.



5 Fazit

Ohne großen Aufwand lassen sich mit NodeRed und InfluxDb Daten speichern, sodass diese für zukünftige Auswertungen verwendet werden können. Daher sollte dies bei jedem Projekt implementiert werden.