

Citizen Science – *IoT für die Sinne*

Projektthema:

Prototypische Entwicklung eines IoT-Geräts, welches die Sinne Hören und/ oder Sehen einbindet/ unterstützt/ ersetzt.

Die Idee:

Ein Lärmimmisionsmessgerät – Wir schaffen Aufmerksamkeit für Lärmbelastung!

- Ein Sensor nimmt die Lautstärke der Umwelt war
- Die mit Hilfe des Sensors gesammelten Daten werden in einem Dashboard visualisiert ausgewertet.
- Passanten können mittels Barcode auf das Dashboard über ihr Smartphone zugreifen

Dieses Dokument erläutert
die Vorgehensweise bei der Einrichtung
des Prototypen.

Benötigte Hardware:

- IoT Octopus
- Sound Level Meter
- LoraWAN Antenne
- Featherwing
- Adapterkabel für den analogen Anschluss – Sound Level Meter

Benötigte Accounts (kostenfrei):

- The Things Network
- InfluxDB
- Grafana

Konzept:

- Mikrofon des Sound Level Meter nimmt einen Wert auf
- Wert wird an den IoT Octopus übertragen und in dB(A) umgerechnet
- Wert wird via LoraWAN an das The Things Network übertragen
- Von dort mit Hilfe von NodeRed auf an InfluxDB weitergeleitet
- Das Grafana Dashboard greift auf InfluxDB zu und stellt Daten grafisch dar
- QR Code für Erreichung des Dashboards auf dem Plakat



Hilfreiche Links

<https://www.umwelt-campus.de/fileadmin/Umwelt-Campus/IoT-Werkstatt/octopus/Quickstart.pdf>

<https://www.umwelt-campus.de/fileadmin/Umwelt-Campus/IoT-Werkstatt/octopus/QuickstartMacOS.pdf>

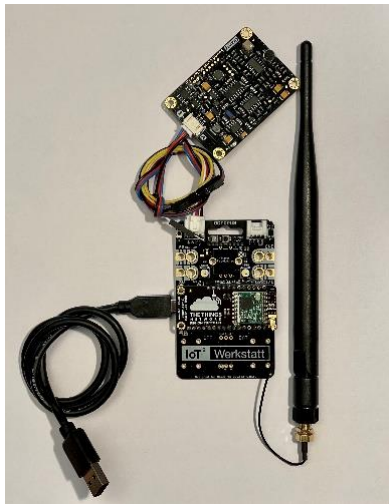
<https://www.thethingsnetwork.org/>

<https://www.influxdata.com/dashboards/>

<https://grafana.com/>

Umsetzung:

1. Zusammenbau der Hardware



- Das Wichtigste zuerst: Die Hardware Komponenten IMMER erst anschließen und danach könnt ihr das Gerät mit eurem PC verbinden, wenn ihr das nicht beachtet schmoren euch ggf. die Kabel an!
- Der Sound Level Meter wird an den Analogen Anschluss des IOT-Octopus geschlossen.
- Im Bild sind 2 verschiedenfarbige Kabel zu sehen – da der Sound Level Meter andere Anschlüsse hat wird einen Adapter benötigt, der dazwischengeschaltet wird
- Zwischen dem Sound Level Meter und dem IOT Octopus wird das schwarze Kabel an das schwarze, das rote an das rote und das blaue an das gelbe Kabel angeschlossen, damit der Sound Level Meter seine Daten übertragen kann.
- Das Featherwing wird mittig auf den IOT-Octopus eingeklickt

- Die LoraWAN-Antenne wird mittels des Kabels an das Featherwing angeschlossen
- Zuletzt wird dann für eine Stromversorgung oder die Programmierung das Mini-USB-B-Netzteil angeschlossen und kann mit einer Batterie/PowerBank oder einem PC verbunden werden

2. The Things Network (TTN) Konto einrichten

UM die vom IOT-Device gesendeten Daten zu empfangen muss ein TTN Konto eingerichtet werden.

Ist dies geschehen, kann in der Console eine Application angelegt werden. Der Name ist selbst zu wählen. Wenn die Application erfolgreich gespeichert wurde, muss ihr ein Device hinzugefügt werden. Wurde dieses auch erfolgreich angelegt, erhält man die Zugangsdaten, also die entsprechenden Schlüssel, um über LoraWAN von einem Gerät aus Daten an das TTN Device zu senden.

3. Arduino programmieren

Es wurde der Standardcode zum Einrichten der LoraWAN Verbindung verwendet und entsprechend an unsere Anforderungen angepasst.

Vorbedingung zum Ausführen des Codes:

- Ein TTN Konto mit einem eingerichteten Device
- Um eine Verbindung herzustellen müssen die entsprechenden Schlüssel aus dem TTN im Device eingetragen werden (DEVEUI, APPEUI, APPKEY).
- Der Sound Level Meter muss an den analogen Anschluss des IOT Octopus angeschlossen werden.

Wenn diese Bedingungen erfüllt sind, kann der Code auf den IOT Octopus hochgeladen werden und er ist funktionsfähig.

Die Schlüssel aus dem Device des TTN müssen wie in der folgenden Abbildung eingetragen werden:

```
static const u1_t PROGMEM DEVEUI[8]={
    0x1E,0x9F,0x07,0xDF,0x[REDACTED]};
void os_getDevEui (u1_t* buf) {
    memcpy_P(buf, DEVEUI, 8);
}

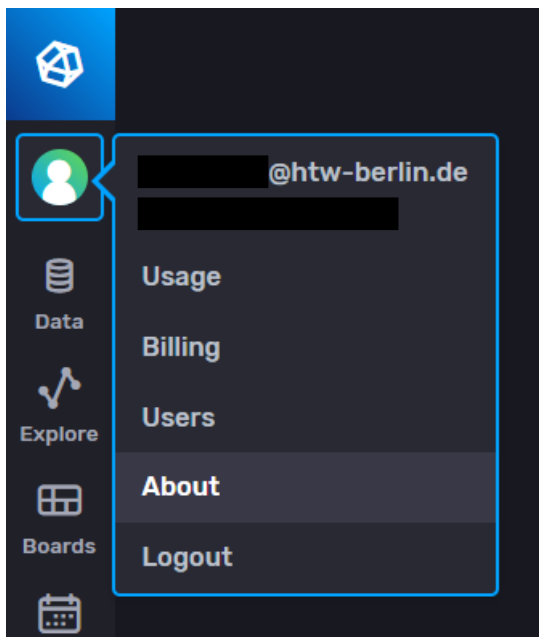
static const u1_t PROGMEM APPEUI[8]={
    0xC0,0x7A,0x03,0xD0,0x7E,[REDACTED]};
void os_getArtEui (u1_t* buf) {
    memcpy_P(buf, APPEUI, 8);
}

static const u1_t PROGMEM APPKEY[16]={
    0xA6,0xE5,0x77,0xC4,0x59,0x91,0x1C,0xFA,0x5F,0xF8,0x4A,0x[REDACTED]};
void os_getDevKey (u1_t* buf) {
    memcpy_P(buf, APPKEY, 16);
};|
```

Unter folgendem Link gibt den vollständigen Quellcode: https://github.com/kayaloe/mwv_iot

4. InfluxBD einrichten

Als erstes erfolgt die Registrierung bei InfluxDB. Danach erhält man die die entsprechende Organization ID im Menü unter „About“.



Im Menü „Data“ ist es uns möglich den Bucket und Tokens zu erstellen

5. Verbindung von TTN zu InfluxDB: NodeRed

Vorbedingung zum Ausführen des Codes:

- Ein TTN Konto mit einem eingerichteten Device und Zugangsdaten
- Ein Influx Konto mit Zugangsdaten
- Installation von NodeRED auf einem beliebigen Device [NodeRED-Erstinstallation](#)

Die Einrichtung einer Decoder-Funktion im TTN ist essentiell, damit die Daten von NodeRED erfasst und weiterverarbeitet werden können.

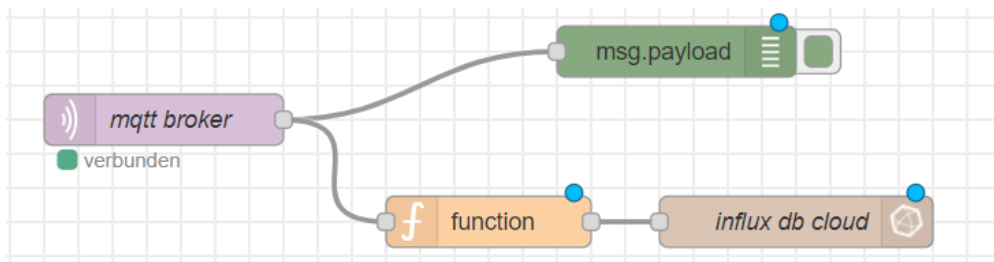
decoder converter validator encoder

```
function Decoder(bytes, port) {  
  var value = bytes[0];  
  var value1 = bytes[1];  
  return {  
    value: value,  
    value1: value1  
  }  
}
```

Um die Daten aus TTN in der InfluxDB Datenbank zu erfassen, wird Node-Red verwendet. Dazu haben wird ein Flow angelegt, in diesen wird ein MQTT Broker eingesetzt, um die Daten abzurufen. Diesem MQTT Broker folgt eine Payload-Funktion und abschließend die Verbindung zur InfluxDB-Cloud.

Mit folgendem Code lassen sich alle Nodes auf einmal importieren:

```
[{"id":"40a0ad5c.e3def4","type":"tab","label":"dB_measurement","disabled":false,"info":"","id":"371fabb4.ea5ed4","type":"mqtt-broker","name":"*TTNAPPLICATIONID*","broker":"eu.thethings.network","port":"1883","tls":"","clientid":"","usetls":false,"compatmode":false,"keepalive":"60","cleansession":true,"birthTopic":"","birthQos":"0","birthPayload":"","closeTopic":"","closeQos":"0","closePayload":"","willTopic":"","willQos":"0","willPayload":"","id":"6ed717ab.628f18","type":"influxdb","hostname":"127.0.0.1","port":"8086","protocol":"http","database":"database","name":"","usetls":false,"tls":"","influxdbVersion":"2.0","url":"https://westeurope-1.azure.cloud2.influxdata.com","rejectUnauthorized":true},{id":"d1505ce0.75035","type":"mqtt in","z":"40a0ad5c.e3def4","name":"mqtt broker","topic":"+/devices+/up","qos":"2","datatype":"json","broker":"371fabb4.ea5ed4","x":150,"y":300,"wires":[["8537753a.37ec38","18c13adb.87bc3d"]]},{"id":"8537753a.37ec38","type":"debug","z":"40a0ad5c.e3def4","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType":"msg","statusVal":"","statusType":"auto","x":450,"y":240,"wires":[],"id":"fe2e7b89.85b888","type":"influxdb out","z":"40a0ad5c.e3def4","influxdb":"6ed717ab.628f18","name":"influx db cloud","measurement":"+/devices+/up","precision":"","retentionPolicy":"","database":"database","precisionV18FluxV20":"ms","retentionPolicyV18Flux":"","org":"*ORGANIZATIONID*","bucket":"*BUCKETNAME*","x":600,"y":360,"wires":[],"id":"18c13adb.87bc3d","type":"function","z":"40a0ad5c.e3def4","name":"","func":"var msg7 = {};\n\nmsg7.payload = [{\"sound_level_meter_db\":\nmsg.payload.payload_fields.value}];\n\nreturn\nmsg7;\n","outputs":1,"noerr":0,"initialize":"","finalize":"","x":370,"y":340,"wires":[["fe2e7b89.85b888"]]]]
```



Nach dem Import aller Nodes sollte zuerst die der Anwendung zugrundeliegende TTN Application ID hinterlegt werden. Unter „Server“ sind Zugangsdaten und der EU Server des things networks, sowie der Port zu speichern. Unter „Topic“ lassen sich mit „+/devices+/up“ die Daten aller Devices unter dieser ID abholen. Im Punkt „Output“ wird festgelegt, dass ein JSON weitergegeben wird.

mqtt in Node bearbeiten

Löschen Abbrechen Fertig

Properties

Server *ServerNameTTN*

Topic +/devices+/up

QoS 2

Output a parsed JSON object

Name mqtt broker

Bevor die Daten an die InfluxDB weitergeleitet werden, wird noch die entsprechende Payload-Funktion konfiguriert:

function Node bearbeiten

Löschen Abbrechen Fertig

Properties

Name Name

Setup Funktion Close

```
var msg7 = {};
msg7.payload = [{"sound_level_meter_db": msg.payload.payload_fields.value}];
return msg7;
```

Nun wird noch die Node für die Verknüpfung zu InfluxDB konfiguriert. Unter „Server“ sollte zunächst die URL und der Token bekanntgegeben werden. Im Anschluss werden die Organization ID und der entsprechende Bucket aus InfluxDB eingetragen.

6. Verknüpfung von InfluxDB und Grafana

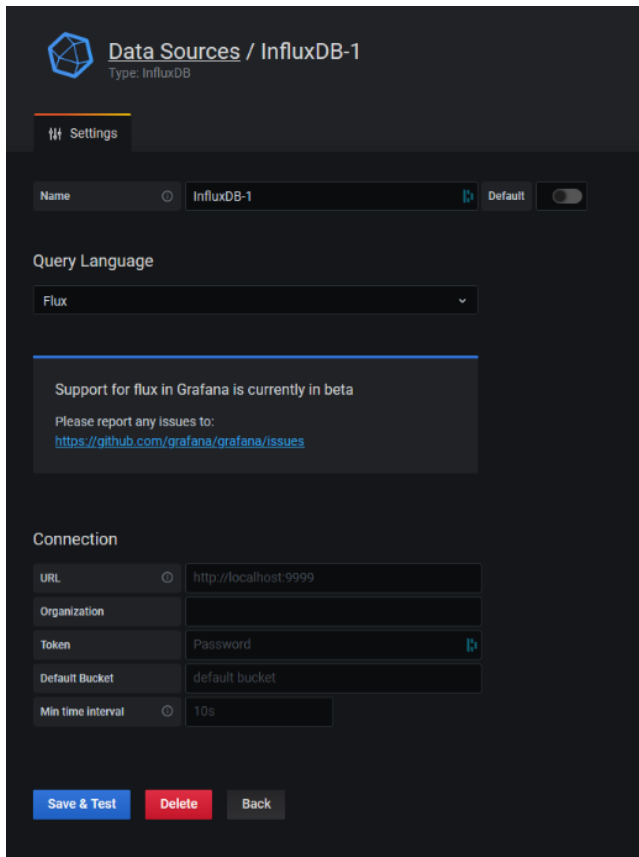
Damit die erfassten Daten auf einem öffentlichen Dashboard geteilt werden können müssen von InfluxDB die Sensordaten an Grafana weitergeleitet werden. Da zum Stand Januar 2021 die InfluxDB Dashboards nicht veröffentlicht werden können und die Funktionalität gegenüber von Grafana auch noch deutlich eingeschränkt ist.

Es wird empfohlen sich einen Grafana Cloud Account zu erstellen, sodass das Hosting durch den Anbieter kostenlos übernommen wird.

Zur Verknüpfung von InfluxDB und Grafana muss das Menü Configuration (Zahnrad-Symbol) > Data Sources geöffnet werden. In der oberen rechten Ecke befindet sich der Button „Add Data source“. Anschließend öffnet sich ein Menü in dem InfluxDB als Datenquelle ausgewählt werden muss. Es müssen alle Felder spezifiziert werden.

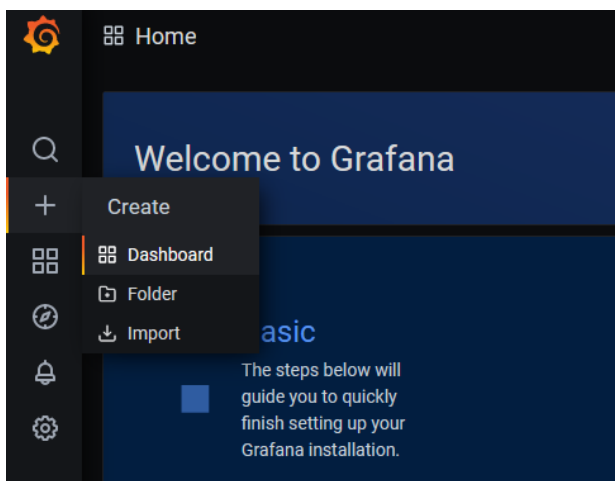
1. Als Query Language wird Flux empfohlen.
2. Als URL ist die Server-Domain in InfluxDB Cloud zu spezifizieren. Die zu spezifizierende Domain steht zwischen `https://... /` und dem nächsten Slash und kann variieren zwischen einzelnen Konten.
3. Die OrganizationID steht in InfluxDB cloud unter Nutzerkonto > User > About > Common Ids und ist in Grafana in das gleichnamige Feld zu übertragen.
4. Ebenfalls muss der für NodeRED generierte Token wiederverwendet werden und unter dem Punkt Token spezifiziert werden.
5. Der in InfluxDB Cloud spezifizierte Bucket-Name muss unter dem Punkt Bucket eingetragen werden.

6. Unter „min time interval“ ist die Abfragegeschwindigkeit spezifiziert, diese muss nicht weiter spezifiziert werden kann aber verringert werden, wenn das Dashboard öfter die Daten aus InfluxDB Cloud abrufen soll.

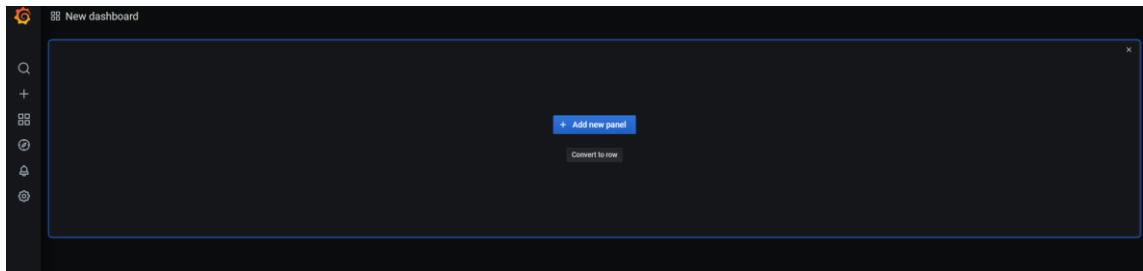


7. Dashbaord erstellen - Grafana

Ein Dashboard in Grafana kann über das Dashboard-Menü erstellt werden, siehe Bild.



Über das Feld „Add new panel“ kann eine neue Ausgabe erstellt werden.



Unter dem Reiter Query muss nun InfluxDB als Datenquelle ausgewählt werden. Es öffnet sich ein Feld für die Flux Query Abfrage. Folgende Abfrage wurde verwendet:

```
from(bucket: "BUCKET_NAME")
  |> range(start: v.timeRangeStart, stop:v.timeRangeStop)
  |> filter(fn: (r) =>
    r._measurement == "DEVICE_NAME" and
    r._field == "MEASURE_VALUE"
  )
```

Device_Name: Entspricht dem in NodeRED spezifizierten Device_Namen vom TTN.

Measure_Value: Entspricht dem in NodeRED spezifizierten Werte_Name der Influx übermittelt wird

Zur Bearbeitung des Panels können auf der rechten Seite unter dem Reiter ‚Panel‘ weitere Einstellungen vorgenommen werden, wie beispielsweise das Festlegen des Titels des Panels.