

Datenabruf und Auswertung

Vorbereitungsworkshop für das Messsystem LoRaMINT

Matthias Ruf

Datenabruf und Auswertung • Ablauf Workshoptag 5

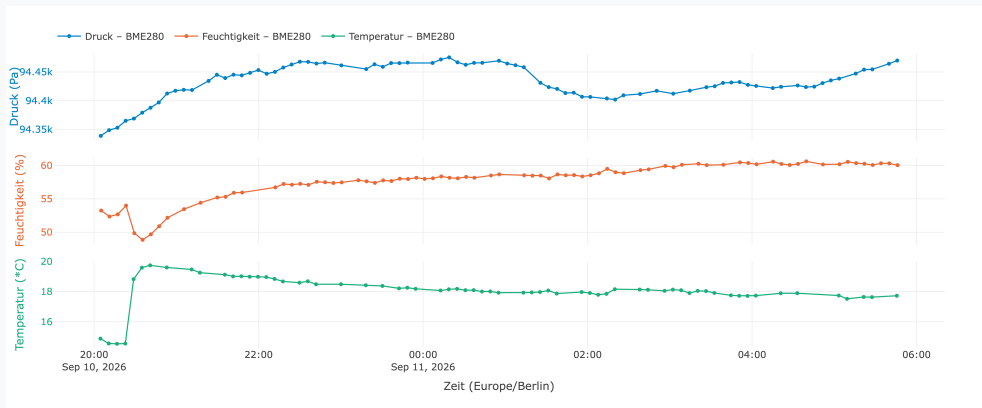
SFZ Südwürttemberg • Standort Ochsenhausen

Workshoptag 5

1

Was die Nacht gebracht hat

Eine Nacht in drei Kurven



Eine Node, eine Nacht: Druck, Feuchte und Temperatur

Was da zu sehen ist

Ein Fehlstart

Drei Punkte bei 15 Grad,
dann ein Sprung auf 19,7.

Was da gemessen
wurde, war noch nicht
der Raum.

Dann wird es kühler

Von 19,7 auf 17,7 Grad
über die ganze Nacht.

Zwei Grad. Drinnen geht
es langsam.

Die Feuchte gegenläufig

Temperatur runter,
Feuchte hoch: von 50
auf 60 Prozent.

Beide Kurven gehören
zusammen.

Diese Seite zeichnet für euch

Gerät, Messgröße und Zeitraum auswählen, fertig. Heute baut ihr dasselbe Bild selbst, und dann bestimmt ihr, was darauf steht.

2

Zurück auf den Laptop

Das Board bleibt heute aus

Was bleibt

Variablen, Listen,
Schleifen,
Verzweigungen,
Funktionen,
Dictionaries.

Alles von Tag 1 bis 3 gilt.

Was wegfällt

machine, Pin, I2C und
das Speichern aufs
Board.

Diese Namen gibt es hier
nicht.

Was dazukommt

Bibliotheken, die auf
dem Board keinen Platz
hatten.

Zum Fragen und zum
Zeichnen.

Ein Programm von gestern läuft hier nicht

Sobald `machine` darin vorkommt, endet es im `ImportError`. Kein Fehler von euch, nur ein anderer Computer.

Thonny umstellen

Vier Tage MicroPython, ab jetzt nicht mehr

1. Unten rechts in der Statusleiste auf die Interpreter-Anzeige klicken
2. In der Liste "Lokales Python 3" anklicken



Dieselbe Liste wie an Tag 1, diesmal die obere Zeile

3

Wie ein Programm fragt

Eine Adresse, die antwortet

Bisher habt ihr die Messwerte *angesehen*.
Jetzt sollen sie in ein Programm.

Dafür gibt es eine zweite Adresse, neben
der für Menschen:

```
loramint.de/api/v1/measurements
```

Ruft sie im Browser auf. Es kommt kein
Bild, keine Seite, nur Text.

Analogie

Eine **API** ist eine Adresse, die
Text zurückgibt, statt eine Seite
anzuzeigen.

Gemacht für Programme, nicht
für Augen.

Was zurückkommt

```
{
  "data": [
    {
      "measurand": "Temperatur",
      "value": "13.4",
      "unit": "*C",
      "location": "Raum 101"
    }
  ]
}
```

Analogie

Ein **JSON**-Objekt ist ein Dictionary, das durch die Leitung passt.

Geschweifte Klammern, Schlüssel, Doppelpunkt, Wert. Das kennt ihr seit Tag 3. Neu ist nur, dass es hier als Text ankommt.

Die eckigen Klammern darin sind eine Liste, auch die kennt ihr.

Anfrage und Antwort

Die Anfrage sagt

wohin die Adresse

was die Filter: Messgröße, Ort,
Zeitraum

wer der Nachweis

Die Antwort bringt

eine Zahl hat es geklappt?

die Daten wenn ja

200 Alles in Ordnung, die Daten sind dabei

401 Der Nachweis fehlt oder stimmt nicht

404 Diese Adresse gibt es nicht

Der Nachweis gehört nach oben

Eure Messwerte stehen nicht für jeden offen. Das Programm muss sich ausweisen, mit einem **Token**.

Er reist in der **Kopfzeile** der Anfrage, nicht in der Adresse:

```
Authorization: Bearer <token>
```

Analogie

Ein **Token** ist ein Ausweis für Programme statt für Menschen.

Nicht an die Adresse hängen

Was in der Adresse steht, landet im Verlauf des Browsers und in jedem Protokoll. Ein Ausweis hat dort nichts verloren.

4

Die eigenen Werte holen

Mehr Zeilen braucht es nicht

```
1 # Die Messwerte der Nacht holen.  
2  
3 import requests  
4  
5 TOKEN = "hier steht dein Token"  
6  
7 antwort = requests.get(  
8     "https://www.loramint.de/api/v1/measurements",  
9     headers={"Authorization": "Bearer " + TOKEN},  
10    params={"measurand": "Temperatur", "per_page": 5},  
11 )  
12  
13 print(antwort.status_code)
```

Was auf dem Bildschirm steht

print(daten)

```
{'data': [  
  {'id': '0dd5eff0-2ad1-449e-82d3-2bf3afbd6576', 'deviceEui': 'A84041C79184DB77',  
    'measurand': 'Temperatur', 'unit': '*C', 'datatype': 'float', 'sensor': 'BME280',  
    'location': 'NWT-Vor', 'value': '24.248624801635742', 'timeMethod': 'server',  
    'recordedAt': '2026-09-09T15:58:47.852Z', 'createdAt': '2026-09-09T15:58:47.852Z'},  
  {'id': '2c401b0d-ad91-438f-97b1-0bfc7a40a357', 'deviceEui': 'A8404145A184DB83',  
    'measurand': 'Temperatur', 'unit': '*C', 'datatype': 'float', 'sensor': 'BME280',  
    'location': 'Raum 101', 'value': '21.392559051513672', 'timeMethod': 'server',  
    'recordedAt': '2026-09-02T23:25:10.155Z', 'createdAt': '2026-09-02T23:25:10.155Z'}],  
'pagination': {'page': 1, 'per_page': 2, 'total': 165419,  
  'total_pages': 82710, 'has_next': True}}
```

die zwei Teile • Feldnamen • Werte • Klammern und Kommas

Die Antwort hat zwei Teile

`data`

Die Liste der Messwerte. Jeder Eintrag ist ein Dictionary mit `measurand`, `value`, `unit`, `location` und `recordedAt`.

`pagination`

Wie viele es insgesamt sind und auf wie viele Seiten sie verteilt wurden.

Die Antwort kommt seitenweise

Ohne Angabe zwanzig Werte auf einmal, höchstens hundert. Oben stehen fünf, weil fünf auf den Bildschirm passen. Wer mehr will, setzt `per_page` hoch und blättert mit `page`.

Woher der Token kommt

`loramint.de/management/tokens`

Feld	Was hineingehört
Name	wofür er da ist
Gehört der Gruppe	wem die Messwerte gehören
Laufzeit in Tagen	höchstens 360

Genau einmal

Der Wert erscheint nur beim Anlegen.
Gespeichert ist nur seine Prüfsumme.

Haben und Dürfen sind zweierlei

Erst die Freigabe der Datengruppe
macht ihn nützlich, und sie lässt sich
zurücknehmen.

Jetzt ihr

Einzelarbeit am Arbeitsblatt

Arbeitsblatt 5 • Aufgabe 1

Holt eure eigenen Messwerte. Danach steht eine Nacht auf eurem Bildschirm, geholt von eurem Programm.

401

Der Token stimmt nicht.

404

Die Adresse stimmt nicht.

Leere Liste

Der Filter passt auf nichts. Ort und Messgröße prüfen.

5

Rechnen mit echten Zahlen

Eine Falle von Tag 3, noch einmal

```
>>> messwert["value"]  
'13.4'  
>>> messwert["value"] + 1  
TypeError
```

Die Anführungszeichen verraten es:
Der Messwert kommt als **Text**, nicht als
Zahl.

Genau wie gestern

An Tag 3 lieferte
`sensor_bme.values` ebenfalls Text
mit Einheit, und ihr seid auf
`read_compensated_data`
ausgewichen.

Hier gibt es kein Ausweichen.
`float()` macht aus dem Text eine
Zahl.

Aus der Antwort eine Liste

```
1 # Aus der Antwort eine Liste von Zahlen machen.
2
3 messwerte = daten["data"]
4 messwerte.reverse()          # die API antwortet neueste zuerst
5
6 werte = []
7 for messwert in messwerte:
8     werte.append(float(messwert["value"]))
9
10 print("Anzahl: ", len(werte))
11 print("Mittel: ", round(sum(werte) / len(werte), 1))
12 print("Höchster:", max(werte))
13 print("Tiefster:", min(werte))
```

Jetzt ihr

Einzelarbeit am Arbeitsblatt

Arbeitsblatt 5 • Aufgabe 2

Mittelwert, Höchst- und Tiefstwert eurer Nacht. Die Funktionen dafür kennt ihr seit Tag 1.

Der Weg dahin

Aus der Antwort wird eine Liste, aus dem Text eine Zahl, und dann rechnet ihr wie gewohnt.

Wenn ihr fertig seid

Sieht ein Wert unmöglich aus? Dann ist er es vielleicht auch.

6

Aus Zahlen wird ein Bild

Vier Zeilen für ein Diagramm

```
1 # Aus Zahlen wird ein Bild.  
2  
3 import matplotlib.pyplot as plt  
4  
5 plt.plot(werte)  
6  
7 plt.title("Temperatur über Nacht")  
8 plt.xlabel("Messung")  
9 plt.ylabel("Grad Celsius")  
10  
11 plt.show()
```

Was auf ein Diagramm gehört

Ein Titel

Was ist zu sehen, und von wann?

Zwei Achsen

Beide beschriftet, senkrecht mit der Einheit.

Die Kurve

Und sonst nichts, was nicht gebraucht wird.

Ohne Einheit behauptet ein Bild etwas, das es nicht zeigt

Eine Kurve, die von 20 auf 13 fällt, sagt ohne Achsenbeschriftung nicht, ob es um Grad, Prozent oder Stunden geht. Das ist derselbe Gedanke wie beim Dictionary an Tag 3: Eine Zahl allein sagt nichts.

Jetzt ihr

Erst gemeinsam, dann allein

Arbeitsblatt 5 • Aufgabe 3

Zeichnet eure Nacht. Erst die Kurve, dann Titel und Achsen, dann das Speichern.

Das Ziel für alle

Eine beschriftete Kurve, gespeichert als Datei.

Wenn ihr fertig seid

Es gibt Ausbaustufen. Sie kommen später dran.

7

Schluss und Rückmeldung

Fünf Tage

Am Montag

Ein Board, das eine LED anschalten konnte.

Dazwischen

Kontrollstrukturen, ein Sensor, eine Funkstrecke, ein Netz.

Heute

Eine Messstation, die von allein misst, sendet und sich auswerten lässt.

Und danach

Die Boards bleiben da. Was ihr hier gebaut habt, ist der Anfang eines eigenen Projekts am SFZ, nicht das Ende eines Kurses.

Kahoot 7

Abschlussfeedback

So läuft es

Ein letztes Mal **kahoot.it** und die PIN von der Wand. Diesmal ohne richtige Antworten: Es geht um eure Rückmeldung.

Dazu der Bogen

Auf Papier und anonym. Dort steht das, was in einer Runde von fünf Leuten niemand sagt.

Und die Kärtchen

Die Strichlisten aller fünf Tage kommen zurück.

8

Zeigen und weiterbauen

Zwei Minuten, kein Vortrag

Jede und jeder zeigt das eigene Diagramm

Am Beamer, zwei Minuten. Eine Sache, auf die ihr stolz seid, und ein Satz dazu, was das Bild zeigt.

Auch eine Lücke zählt

Wenn die Kurve abbricht, erzählt das etwas. Zeigt es trotzdem.

Wer früher weg muss

zeigt zuerst.

Ausbaustufen

Für alle, die noch Zeit haben

- a Eine zweite Messgröße dazu, mit eigener Achse
- b Nacht und Tag im Diagramm farbig hinterlegen
- c Höchst- und Tiefstwert beschriften
- d Zwei Orte übereinanderlegen und vergleichen
- e Das Bild als Datei speichern, mit Titel und Beschriftung

Freiwillig, und zwar wirklich

Das sind keine Aufgaben, die jemand nachholen müsste. Wer jetzt geht, geht mit einem fertigen Ergebnis.