

Diese Zeilen brauchst du immer

hilft bei

Aufgabe 2 und 3

verwendet

```
from loramint import LoRaMINT, MintValue
```

```
lora = LoRaMINT()  
lora.join()
```

Sie stehen in **jedem** eigenen Programm, immer ganz oben und immer nur einmal. `join` gilt weiter, auch nach einem Neustart; beim zweiten Mal kommt es sofort zurück.

Zwischen zwei Nachrichten liegen **10 Sekunden**. Wer früher sendet, bekommt `False` zurück.

Den Sensor auslesen

hilft bei

Aufgabe 2

verwendet

```
from machine import Pin, I2C  
import bme280  
  
i2c = I2C(0, sda=Pin(10), scl=Pin(11))  
sensor_bme = bme280.BME280(i2c=i2c)  
  
temperatur, druck, feuchte = sensor_bme.read_compensated_data()
```

Reihenfolge: Temperatur, Druck, Feuchte. Der Druck kommt in Pascal, `druck / 100` ergibt hPa.

Der zweite Sensor hängt am selben `i2c`:

```
import bh1750  
sensor_lux = bh1750.BH1750(i2c)  
lux = sensor_lux.luminance(bh1750.BH1750.ONCE_HIRES_1)
```

Vom Messwert zur Nachricht

hilft bei

Aufgabe 2

verwendet

In dieser Reihenfolge, sonst verschickst du eine Zahl mit zu vielen Stellen:

```
temperatur, druck, feuchte = sensor_bme.read_compensated_data()  
  
messwert = MintValue(round(_____, 1), "____", "_____",  
                      "_____", "____")  
  
lora.sendValue(messwert)
```

Die fünf Angaben: **Wert, Einheit, Ort, Messgröße, Sensor**. Gerundet wird *vor* dem Senden.

ID
S041



Diese Zeilen brauchst du immer

Hilfskärtchen • Workshoptag 4

ID
S042



Den Sensor auslesen

Hilfskärtchen • Workshoptag 4

ID
S043



Vom Messwert zur Nachricht

Hilfskärtchen • Workshoptag 4

Auf den Knopf warten

hilft bei
Aufgabe 1, Experte

verwendet

Ein Knopf ist ein Eingang. Gedrückt heißt 0, losgelassen 1.

```
from machine import Pin

knopf = Pin(0, Pin.IN, Pin.PULL_UP)

while knopf.value() == ____:
    ____
```

Die Idee: Solange nichts gedrückt ist, soll das Programm auf der Stelle treten. Ein Schleifenrumpf darf nicht leer bleiben, aber es gibt ein Wort für "tu nichts".

Erst nach der Schleife wird gesendet.

Die Schleife bauen

hilft bei
Aufgabe 3

verwendet

```
import machine

<hier steht alles, was einmal passiert>

while True:
    <messen>
    <senden>
    machine.lightsleep(60000)
```

Die Zahl sind **Millisekunden**. Über der Schleife stehen die `import`-Zeilen, `i2c`, der Sensor, `lora` und `join` **einmal**. Wer sie hineinzieht, meldet sich jede Minute neu an. Was sich wiederholen soll, steht eingerückt darunter, das Warten am Ende gehört dazu.

Wenn nichts ankommt

hilft bei
allen Aufgaben

verwendet

Was du siehst	Was zu tun ist
<code>check_connection → False</code>	Drähte prüfen. Grün und blau vertauscht? Sie gehören über Kreuz.
<code>join → False</code>	Sitzt die Antenne? Noch einmal versuchen, nicht neu starten.
<code>sendLog → False</code>	Zu schnell gesendet. Zehn Sekunden warten.
<code>ImportError</code>	Die Bibliothek liegt nicht auf dem Board. Bei <code>loramint</code> ist es ein ganzer Ordner.
<code>TypeError</code>	Text und Zahl zusammen. <code>str()</code> macht aus einer Zahl Text.
Status-Fenster leer	Erst warten. Bis etwas ankommt, vergehen Sekunden.

ID
S044



Auf den Knopf warten

Hilfskärtchen • Workshoptag 4

ID
S045



Die Schleife bauen

Hilfskärtchen • Workshoptag 4

ID
S046



Wenn nichts ankommt

Hilfskärtchen • Workshoptag 4