

Erste Schritte auf dem ESP32

Vorbereitungsworkshop für das Messsystem LoRaMINT

Matthias Ruf

Motivation und Einführung • Ablauf Workshoptag 1

SFZ Südwürttemberg • Standort Ochsenhausen

Workshoptag 1

1

Warum wir hier sind

Eine Frage, die man nicht nachschlagen kann

Wie kalt wird es nachts wirklich im Schulgarten? Wie feucht ist es im Gewächshaus?

Solche Fragen spezifischen Fragen lassen sich nicht durch Google lösen. Man muss **selbst messen**.

Was ist dafür nötig?

Ein Gerät, das misst. Eine Verbindung, die die Werte zum Speicherort bringt. Und schließlich der Speicherort selbst.



WarnMe – ein Frühwarnsystem für Hochwasser, gebaut am SFZ

Das gibt es hier schon

Schülerprojekte am SFZ Ochsenhausen



WarnMe

Sensorik, die Messwerte auswertet und selbstständig Alarm schlägt.



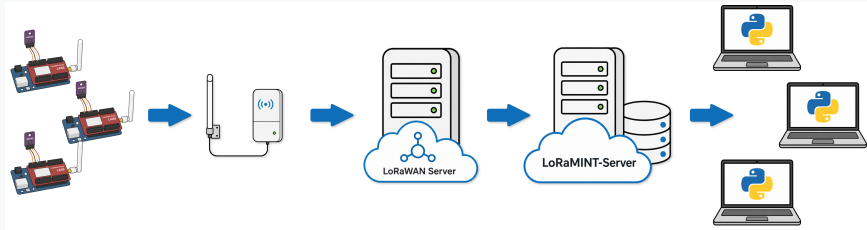
CO₂-Checker

Luftqualität im Klassenzimmer messen und sichtbar machen.

Beide verbinden **Programmieren** und **echte Sensorik**. Beide Projekte arbeiteten mit realen Zahlen.

LoRaMINT

Ein Messsystem für die MINT-Begabtenförderung



Datenfluss im LoRaMINT-System: mehrere Sensor-Nodes senden über ein LoRaWAN-Gateway an den Server, der die Werte an das LoRaMINT-Backend weiterreicht – dort liegen sie zum Abruf bereit

Die Stationen im Einzelnen

Erfassen

Node: ESP32 mit Sensor und LA66-Shield. Misst und sendet.

Gateway: nimmt das Funksignal auf.

Sammeln

The Things Network: der LoRaWAN-Server.

Backend: speichert die Werte und bietet eine API an.

Auswerten

Endgerät: ruft die Werte über die API ab – zum Beispiel mit einem Python-Programm auf dem Laptop.

Warum Funk statt WLAN

LoRaWAN reicht über Kilometer und kommt mit sehr wenig Strom aus. Dafür sind die Nachrichten winzig. Das passt genau zu Messwerten: Eine Temperatur ist eine kurze Zahl, kein Video.

Zwei Stellen, an denen ihr eingreift

Werte hineingeben

Ihr baut eine eigene Node: Sensor anschließen, Messwerte auslesen, über LoRaWAN verschicken.

Programmiert wird sie in **MicroPython**.

Werte herausholen

Ihr holt die gesammelten Messreihen über die API zurück, rechnet damit und stellt sie als Diagramm dar.

Dafür genügt **Python** auf dem Laptop.

MicroPython und Python unterscheiden sich in dem, was wir hier brauchen, praktisch nicht. Was ihr an den ersten Tagen lernt, gilt in der Regel für beide Sprachdialekte.

Was ihr dafür können müsst

- Variablen und grundlegende Datentypen
- Operatoren und einfache Berechnungen
- Listen und Dictionaries
- Schleifen und Verzweigungen
- Funktionen aufrufen, Bibliotheken einbinden

Das ist die Liste für den ganzen Workshop. Ihr müsst davon nichts mitbringen. Genau dafür sind die ersten drei Tage da.

Die fünf Workshoptage

1. **Erste Schritte auf dem ESP32** – Hardware, Umgebung, Variablen, Datentypen, Operatoren
2. **Steuerung des Programmablaufs** – Listen, Schleifen und Verzweigungen
3. **Sensorik und Bibliotheken** – Sensorschaltung aufbauen und auslesen, Dictionaries
4. **Einspeisen ins LoRaMINT-System** – LoRaWAN, Node registrieren, Langzeitmessung
5. **Datenabruf und Auswertung** – Abruf über die API, Rechnen, Diagramme

Der heutige Tag

1. Kurzer Wissenstest – ohne Punkte, ohne Rangliste
2. Die Hardware in der Projektbox
3. Board anschließen, Entwicklungsumgebung, erstes Programm
4. Die Onboard-LED ansteuern
5. Variablen, Datentypen, Operatoren, Funktionen aufrufen

Gleich: ein paar Fragen

Es geht nur darum, wo ihr gerade steht. Das Quiz dient dazu, dass ich mich die nächsten Tage auf euern Wissensstand einstellen kann. Wer etwas nicht weiß, ist trotzdem richtig hier.

2

Wo steht ihr gerade?

Kahoot 1

Zehn Fragen, ohne Punkte

So läuft es

Öffnet **kahoot.it** und gebt die PIN ein, die gleich an der Wand steht. Zwei Fragen zu eurer Erfahrung, acht zu Python und Hardware.

Es gibt nichts zu gewinnen

Keine Punkte, keine Rangliste. Ich will nur wissen, wo ihr steht.

Nichts gewusst?

Völlig in Ordnung, dann einfach raten oder die Zeit ablaufen lassen.

3

Die Hardware

Was liegt da vor euch?

Sammeln im Plenum

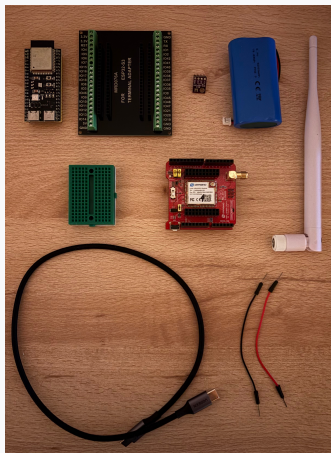
Packt die Projektbox aus

Legt die Teile vor euch hin. Wir gehen sie gemeinsam durch: Was ist das, und wofür könnte es gut sein?

Arbeitsblatt 1 • Ergebnissicherung

Tragt die Namen der Teile in die Übersicht auf dem Arbeitsblatt ein.

Was in der Projektbox liegt



Heute

ESP32-S3, Terminal-Adapter, USB-C-Kabel.

Ab Tag 3

Breadboard, Sensor, Jumperkabel.

Ab Tag 4

LA66-Shield, Antenne, Akku.

4

Das Board zum Laufen bringen

Was wir in den nächsten dreißig Minuten machen

1. Das Board an den Laptop anschließen
2. Thonny einrichten, damit es mit dem Board spricht
3. In der Shell den ersten Befehl abschicken
4. Ein Programm schreiben, speichern und laufen lassen

Zusammen, nicht vorneweg

Wir machen jeden Schritt gemeinsam. Wenn bei euch etwas anders aussieht als vorne:
Hand heben, nicht weiterklicken.

Der ESP32-S3

Ein ganzer Rechner in Streichholzschachtelgröße

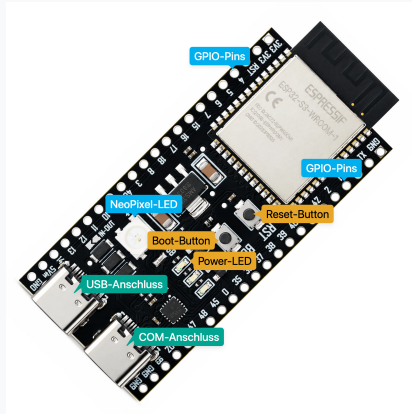
Auf dem Board sitzt ein Mikrocontroller: Rechenwerk, Speicher und Anschlüsse auf einer Platine. Kein Bildschirm, keine Tastatur – er tut nur das, was im Programm steht.

Über die **GPIO-Pins** an den Rändern gibt er Signale aus und nimmt welche entgegen. Dort hängen später die Sensoren.

Schon drauf

Die **NeoPixel-LED** – damit arbeitet ihr gleich im Anschluss, ganz ohne Verkabelung. Und zwei USB-C-Buchsen, von denen nur eine die richtige ist.

Was wo sitzt



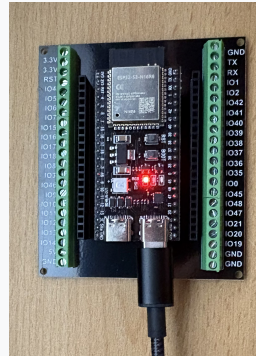
Von den beiden USB-C-Buchsen brauchen wir den COM-Anschluss

Anschließen

1. USB-C-Kabel in den **COM-Anschluss** stecken – von den beiden Buchsen die **rechte**
2. Anderes Ende an den Laptop
3. Die rote Power-LED auf dem Board leuchtet

Nicht jedes Kabel kann Daten

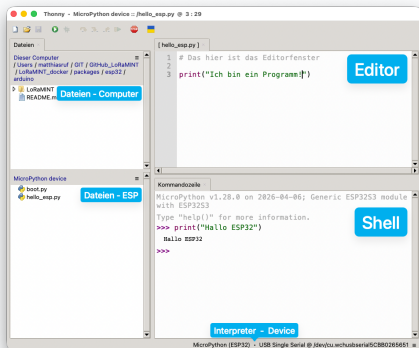
Manche USB-C-Kabel sind reine Ladekabel: Das Board bekommt Strom, taucht am Rechner aber nirgends auf. Nehmt die Kabel aus der Projektbox.



So sitzt das Kabel richtig

Thonny

Die Entwicklungsumgebung



Editor

Hier steht euer Programm. Es läuft erst, wenn ihr es startet.

Kommandozeile

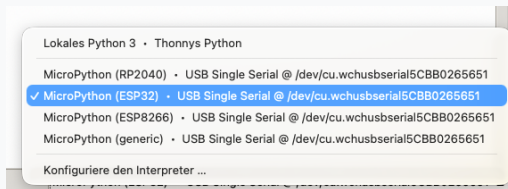
Hier gilt jede Zeile sofort. Gut zum Ausprobieren.

Dateien

Zwei Bereiche übereinander: euer Laptop und das Board.

Thonny mit dem Board verbinden

1. Unten rechts in der Statusleiste auf die Interpreter-Anzeige klicken
2. In der Liste **MicroPython (ESP32)** anklicken



Gebraucht wird der Eintrag mit dem Port, der beim Einstecken dazukam

Kein passender Eintrag dabei

Kabel prüfen, Board ab- und wieder anstecken, Liste neu öffnen. Hilft das nicht: Hand heben.

Der erste Befehl

In der Kommandozeile, unten im Fenster

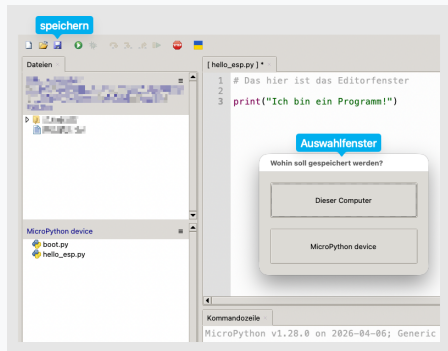
```
>>> print("Hallo LoRaMINT!")  
Hallo LoRaMINT!  
>>> 2 + 3  
5
```

Woran ihr merkt, dass es geklappt hat

Die Antwort erscheint sofort unter eurer Eingabe – und sie kommt nicht vom Laptop, sondern vom Board am Kabel.

Laptop oder Board?

Die wichtigste Frage beim Speichern



Dieser Computer

Die Datei liegt auf dem Laptop.

MicroPython device

Die Datei liegt auf dem Board, auch ohne Kabel.

`main.py` ist besonders

Startet von selbst, sobald das Board Strom bekommt.

Ein Programm, Zeile für Zeile

```
1 # Erstes Programm auf dem ESP32
2 # Workshop Tag 1, Schritt 1.4 - der Programmablauf ist sequenziell:
3 # Zeile fuer Zeile, von oben nach unten, genau einmal.
4
5 print("Hallo LoRaMINT!")
6
7 messstelle = "Schulgarten"
8 temperatur = 18.4
9
10 print("Messstelle:", messstelle)
11 print("Temperatur:", temperatur, "Grad Celsius")
```

Sequenziell

Eins nach dem anderen

Das Board arbeitet das Programm **von oben nach unten** ab. Jede Zeile genau einmal, in der Reihenfolge, in der sie dasteht. Es überlegt nicht, es springt nicht, es fängt nicht vorne an, wenn hinten etwas fehlt.

Was daraus folgt

Eine Variable muss belegt sein, **bevor** sie benutzt wird. Ein Wert muss gemessen sein, **bevor** er gesendet wird.

Was das leichter macht

Wenn ein Programm nicht tut, was es soll, könnt ihr es mit dem Finger nachfahren. Es gibt nur einen Weg hindurch.

Zwei Zeilen vertauscht

reihenfolge.py

```
print("Temperatur:", temperatur)

temperatur = 18.4
```

Was das Board dazu sagt

NameError: name 'temperatur' isn't defined

In Zeile 1 gibt es temperatur noch nicht – die Belegung kommt erst danach. Für uns steht sie sichtbar auf derselben Folie, für das Board liegt sie in der Zukunft.

Fehlermeldungen lesen

Eine Fehlermeldung ist keine Strafe, sondern die genaueste Auskunft, die ihr bekommen könnt. Sie nennt fast immer beides: **wo** es klemmt und **was** nicht stimmt.

Lest sie von unten nach oben – die letzte Zeile sagt, was passiert ist.

Drei, die euch heute begegnen

SyntaxError – verschrieben, meist Klammer oder Anführungszeichen

NameError – Name gibt es (noch) nicht

IndentationError – Einrückung passt nicht

Arbeitsblatt 1 • Der erste Aufgabenblock

Aufgabe 1 – Board anschließen, Thonny auf MicroPython (ESP32) einstellen und in der Kommandozeile `print("Hallo")` ausprobieren.

Aufgabe 2 – Das Programm von der Codefolie abtippen, als `erstes-programm.py` **auf dem Board** speichern und starten. Messstelle und Temperatur auf eigene Werte ändern.

Aufgabe 3 (Zusatz) – Zwei Zeilen absichtlich vertauschen, die Fehlermeldung notieren und in eigenen Worten erklären.

5

Die LED auf dem Board

Bringt die LED zum Leuchten

Einzelarbeit am Arbeitsblatt

Arbeitsblatt 1 • Aufgabenblock zur NeoPixel-LED

Die farbige LED sitzt schon auf dem Board – ihr braucht kein einziges Kabel dafür. Erst überhaupt leuchten, dann in eurer Farbe, dann blinken.

Wenn ihr feststeckt

Fangt mit der Einstiegsaufgabe an und meldet euch.

Wenn ihr fertig seid

Die Spezialaufgabe am Ende des Blocks wartet.

6

Variablen und Datentypen

Werte merken und benennen

Einzelarbeit am Arbeitsblatt

Arbeitsblatt 1 • Aufgabenblock zu Variablen und Datentypen

Ein Messwert muss irgendwo hin, bevor man mit ihm weiterarbeiten kann. Genau dafür sind Variablen da – und der Typ entscheidet, was drin stehen darf.

Vier Typen begegnen euch

Ganze Zahl, Kommazahl, Wahrheitswert, Text.

Probiert es aus

Die Kommandozeile antwortet sofort – schneller als jede Erklärung.

7

Operatoren und Rechnen

Mit Werten rechnen

Einzelarbeit am Arbeitsblatt

Arbeitsblatt 1 • Aufgabenblock zu Operatoren

Aus Messwerten werden erst dann Aussagen, wenn man mit ihnen rechnet: umrechnen, vergleichen, zusammenzählen.

Nicht nur Zahlen

Auch Texte lassen sich verbinden – und Vergleiche liefern wahr oder falsch.

Wenn ihr fertig seid

Die Spezialaufgabe rechnet mit echten Messwerten.

8

Funktionen aufrufen

Funktionen benutzen

Einzelarbeit am Arbeitsblatt

Arbeitsblatt 1 • Aufgabenblock zu Funktionen

`print` habt ihr längst benutzt. `print` ist eine **Funktion**: Man ruft sie beim Namen und gibt ihr in Klammern mit, womit sie arbeiten soll.

Heute nur benutzen

Was in den Klammern wirklich passiert, klären wir an Tag 3.

Wenn ihr fertig seid

Sucht im Arbeitsblatt nach einer Funktion, die ihr noch nicht kennt.

9

Was hängengeblieben ist

Kahoot 2

Lernstandskontrolle

So läuft es

Noch einmal **kahoot.it** und die PIN von der Wand. Diesmal geht es um das, was wir heute gemacht haben.

Was ihr heute geschafft habt

Ein Board angeschlossen, eine Entwicklungsumgebung eingerichtet, ein eigenes Programm geschrieben – und eine LED dazu gebracht, das zu tun, was ihr wolltet.