

# Externe Datenkommunikation mit Calliope mini

Dr. Thomas Meinike

Hochschule Merseburg  
Fachbereich Wirtschaftswissenschaften  
und Informationswissenschaften



116. MNU-Bundeskongress

27.03.2026

# Einstieg

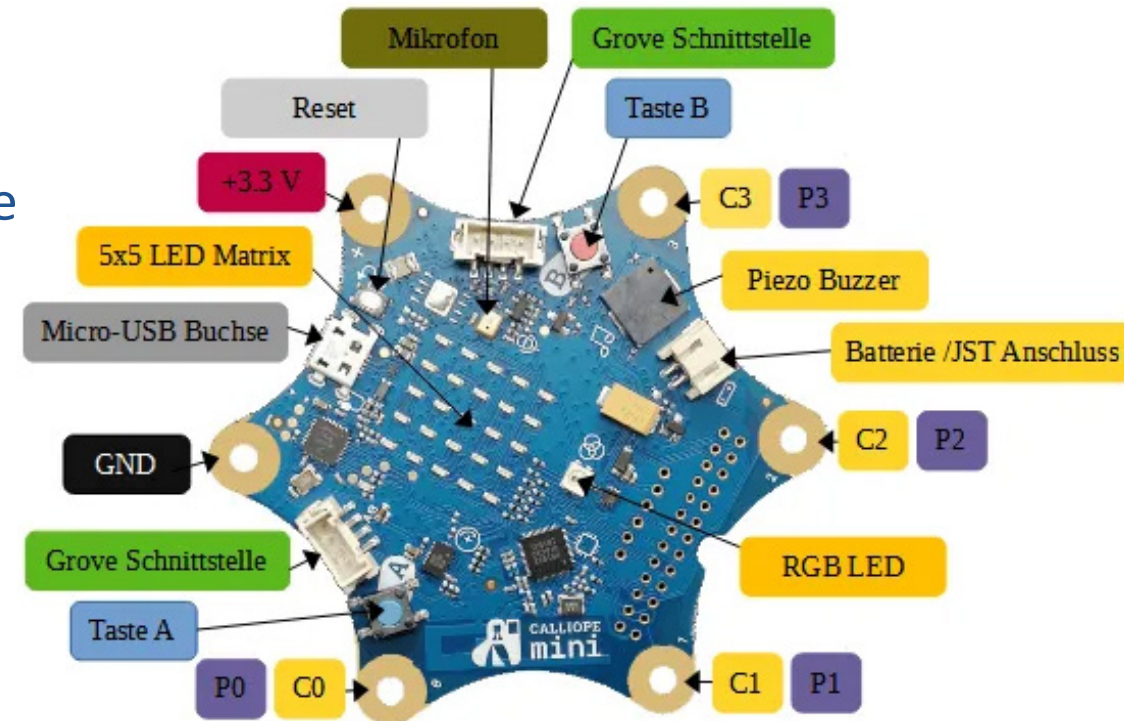


# Einstieg <sup>1/5</sup>

- Calliope mini (1/2/3) kommen seit 2017 auch an der Hochschule Merseburg zum Einsatz, u. a. in Kursen des Autors im Rahmen der Web-Entwicklung mit JavaScript.
- Abfrage und Auswertung von internen sowie externen Sensordaten bieten vielfältige Ansätze zur praktischen Umsetzung von Programmierideen.
- Zur Erstellung der Calliope-spezifischen \*.hex-Programme kommt die Entwicklungsumgebung MakeCode unter Nutzung von JavaScript zum Einsatz.
- Browser-seitig wird ebenfalls JavaScript im Kontext von HTML, CSS und SVG verwendet.
- Grundlegende Kenntnisse zu Aufbau und Nutzung der Geräte werden vorausgesetzt.
- Im Folgenden wird nur auf die zum Verständnis wesentlichen Komponenten eingegangen.

# Einstieg <sup>2/5</sup>

- Im Projekt verwendete Calliope-Basiskomponenten
  - LED-Matrix mit 25 roten LEDs, einzeln ansprechbar
  - Frei programmierbare RGB-LEDs (eine bei C1/C2 bzw. drei bei C3)
  - Tasten A und B zur Steuerung von Programmaktionen
  - Grove-Schnittstellen A0 (I<sup>2</sup>C) und A1 (UART) für externe Peripherie wie Sensoren, Displays, WLAN-Modul
  - Interne Sensoren für Temperatur, Lichtintensität und Geräuschpegel



Bildquelle: <https://draeger-it.blog/vorstellung-des-calliope-mini/>

# Einstieg <sup>3/5</sup>

- Im Projekt verwendete Peripherie für Grove-Anschlüsse

1 | SCD40-Sensor (CO<sub>2</sub>, Temperatur, Luftfeuchtigkeit)

2 | Ultrasonic Ranger (Abstandsmessung)

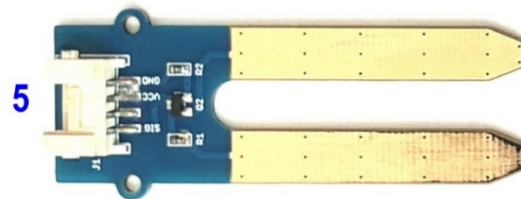
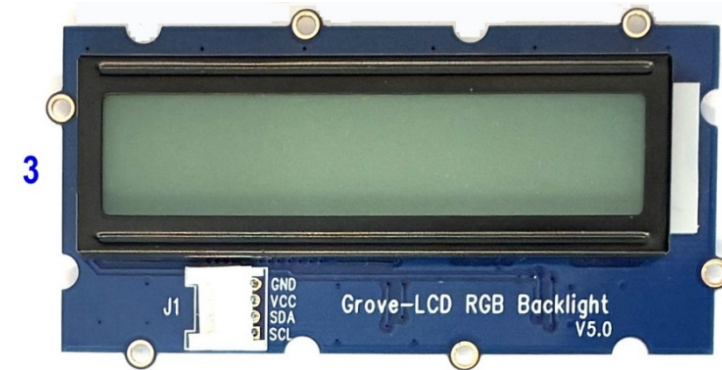
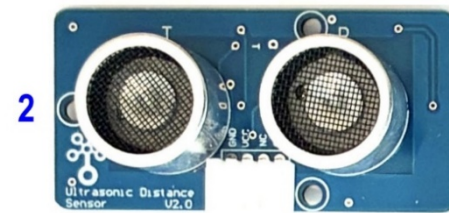
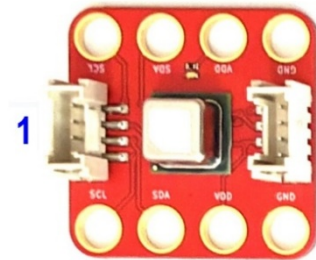
3 | 16x2-Display LCD RGB Backlight

4 | Light Sensor

5 | Moisture Sensor

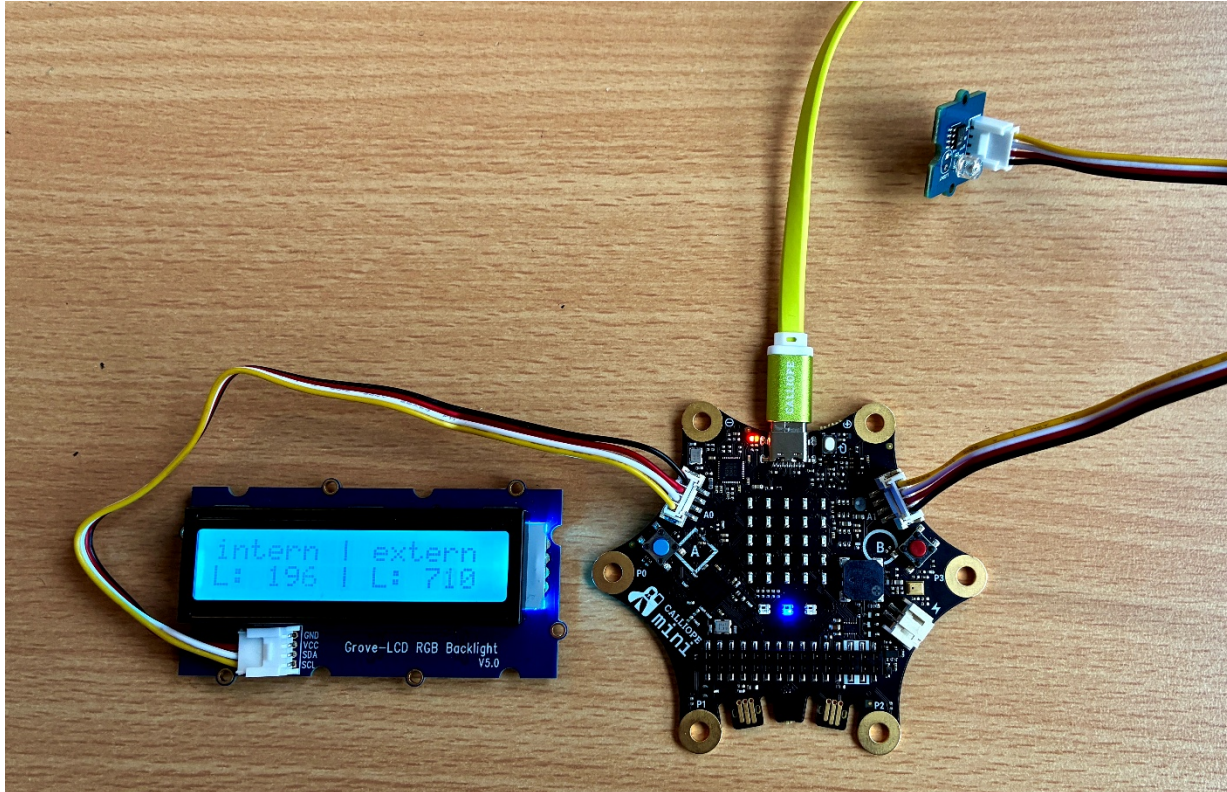
6 | WLAN-Modul ESP8266

7 | 4-Digit-Display

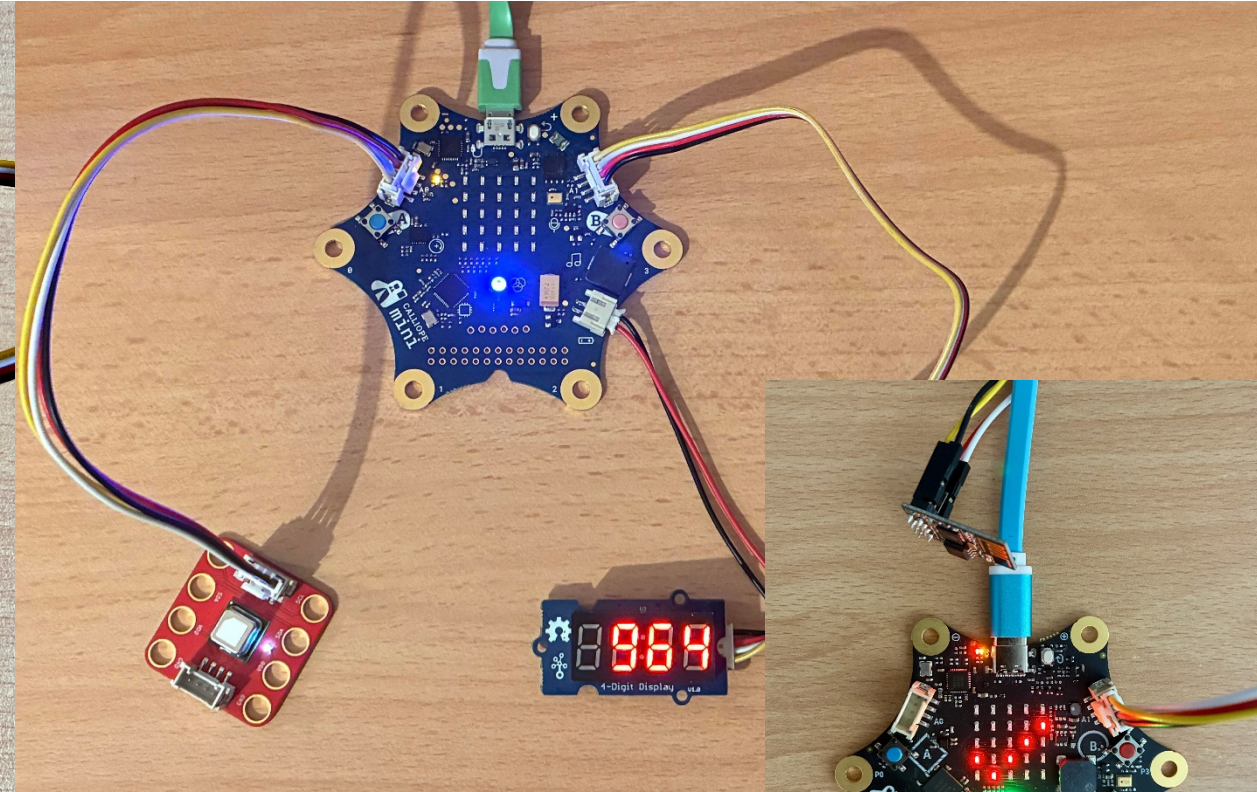


# Einstieg 4/5

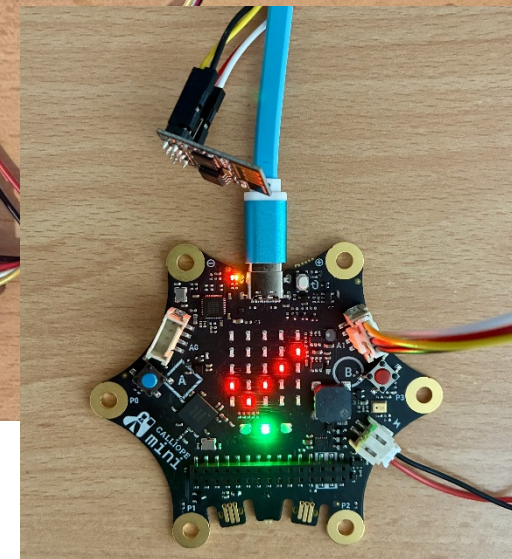
- Beispiel-Projekte mit Peripherie



Externe Lichtsensordaten an 16x2-Display



CO<sub>2</sub>-Messung (ppm) an 4-Digit-Display



Interne Sensordaten an WLAN-Modul

- Programmierung mit der MakeCode-Umgebung und zusätzlichen Erweiterungen

The screenshot displays the Microsoft MakeCode for Calliope mini web editor. The interface is divided into several sections:

- Workspace:** Shows a Calliope mini board with a play button overlay.
- Block Palette:** Located on the left, it contains various categories of blocks: Grundlagen, Eingabe, Musik, LED, C02-Sensor SCD40 (highlighted), Schleifen, Logik, Variablen, Mathematik, Motoren, and Erweiterungen (highlighted).
- Code Editor:** The center area shows JavaScript code for a button click event. The code is as follows:

```
1 basic.forever(function() {
2   // eigener Code für dauerhafte Aktionen ...
3 })
4
5 input.onButtonEvent(Button.A, input.buttonEventClick(), function() {
6   // eigener Code für Aktionen nach Drücken der A-Taste ...
7 })
8
9 function CO2_Messung(): number {
10  // eigene Funktion ...
11  return SCD40.get_co2()
12 }
```
- Extensions Panel:** Opened on the right, it shows a search for 'SCD40' and a list of categories: RGB, Klima, Motor, Lichter und Bildschirm, Anzeige, Software, Jaocad, KI, IoT, Audio, Wissenschaft, and Netzwerk. The 'C02-Sensor-SCD40' extension is highlighted.
- Footer:** A teal bar at the bottom contains a 'Herunterladen' button, the project name 'Calli\_Test', and control buttons for undo, redo, and zoom.

# Daten seriell senden und empfangen



# Daten seriell senden und empfangen <sup>1/8</sup>

- Calliope mini können seit Version 1 seriell kommunizieren

## ✓ Initialisierung

- Standardwert 115200 Baud, 9600 erwies sich als ausreichend.

```
serial.redirectToUSB()  
serial.setBaudRate(9600)
```

## ✓ Daten senden

- Einzelne Werte oder mehrere z. B. als JSON-Strings.

```
let JSONstr: string = "..."  
serial.writeLine(JSONstr)
```

## ✓ Daten empfangen

- Einzelne Werte oder ebenfalls zeilenweise.

```
let received: string = serial.readLine()
```

serial.

- writeValue
- writeLine
- writeString
- writeNumber
- writeNumbers
- readLine
- readUntil
- readString
- onDataReceived
- redirect
- redirectToUSB
- delimiters

# Daten seriell senden und empfangen <sup>2/8</sup>

- Daten für Browser-Verarbeitung als JSON-String ausgeben (1)

```
let send: boolean = false
```

```
input.onButtonEvent(Button.A, input.buttonEventClick(), function() {  
  let L: number = 0  
  let S: number = 0  
  let T: number = 0  
  let F: number = 0  
  
  serial.redirectToUSB()  
  serial.setBaudRate(9600)  
  send = true  
})
```

```
input.onButtonEvent(Button.B, input.buttonEventClick(), function() {  
  send = false  
})
```

- ✓ Taste A initialisiert und startet den Sendevorgang, Taste B beendet ihn.
- ✓ Verwendet werden drei interne Sensoren und ein externer für Feuchtigkeit.

# Daten seriell senden und empfangen <sup>3/8</sup>

- Daten für Browser-Verarbeitung als JSON-String ausgeben (2)

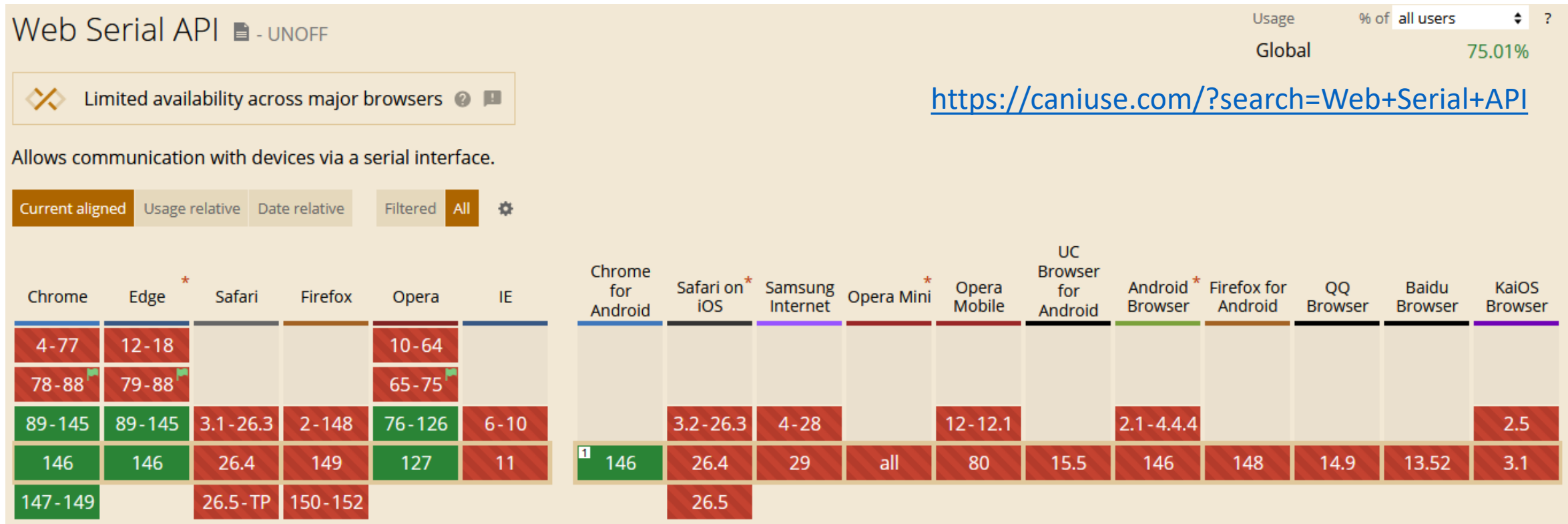
```
basic.forever(function() {  
  if(send) {  
    L = input.lightLevel() // 0 (dark) to 255 (bright)  
    S = input.soundLevel() // 0 (silent) to 255 (loud)  
    T = input.temperature() // °C  
    F = input.moisture() // 0 to 100 % (eigene Erweiterung des input-Namespaces)  
  
    JSONstr = '{ "L" : ' + L + ', "S" : ' + S + ', "T" : ' + T + ', "F" : ' + F + ' }'  
  
    serial.writeLine(JSONstr)  
  }  
  basic.pause(3000)  
})
```

- ✓ Es wird alle drei Sekunden ein Datenpaket gesendet, bis das Senden über die Taste B abgebrochen wird.
- ✓ Ein typischer JSON-String hat die Form { "L" : 58, "S" : 93, "T" : 25, "F" : 81 } mit den Komponenten **L**icht, **S**ound, **T**emperatur und **F**euchtigkeit.

# Daten seriell senden und empfangen <sup>4/8</sup>

- Daten im Browser über die Web Serial API verarbeiten (1)

- ✓ Die WSA ist eine aktuell nur in Chromium-Browsern etablierte JavaScript-DOM-Schnittstelle.



- ✓ Über die WSA können Daten gelesen und auch gesendet werden.

# Daten seriell senden und empfangen 5/8

- Daten im Browser über die Web Serial API verarbeiten (2)
  - ✓ Calliope mini wird über einen COM-Port verbunden → WSA übernimmt.

```
window.addEventListener("load", function() {  
  // Initialisierungen  
  
  // Serielle Verbindung  
  if(navigator.serial) { readSerialData(); }  
  else { // Web Serial API nicht unterstützt. }  
  
  // Daten abfragen und verarbeiten  
  async function readSerialData() {  
    try {  
      // Anfrage zur Auswahl des seriellen Geräts  
      port = await navigator.serial.requestPort();  
  
      if(port.connected) {  
        // Port mit Parametern für Calliope mini öffnen  
        await port.open({ baudRate: 9600, bufferSize: 64 });  
  
        // Daten UTF-8-kodiert empfangen  
        reader = port.readable.getReader();  
        const textdecoder = new TextDecoder("UTF-8");
```

```
while(true) {  
  const { value, done } = await reader.read();  
  if(done) { reader.releaseLock(); break; }  
  else if(value) {  
    // Daten (value) auswerten  
    const buffer = textdecoder.decode(value);  
    const line = buffer.trim();  
    processData(line);  
  }  
}  
}  
}  
catch(error) { // Fehlermanagement }  
finally { reader.releaseLock(); }  
});  
  
// JSON-Daten als Tabelle und SVG-Graphen ausgeben  
function processData(jsonstr) { ... }  
});
```

datenverdrahten.de möchte eine Verbindung mit einem seriellen Anschluss herstellen.

mbed Serial Port (COM3) - Gekoppelt

Verbinden

Abbrechen


# Daten seriell senden und empfangen <sup>6/8</sup>


- Daten im Browser über die Web Serial API verarbeiten (3)
  - ✓ Darstellung der Ergebnisse tabellarisch und pro Sensorwert ein Graph als SVG-Polyline.

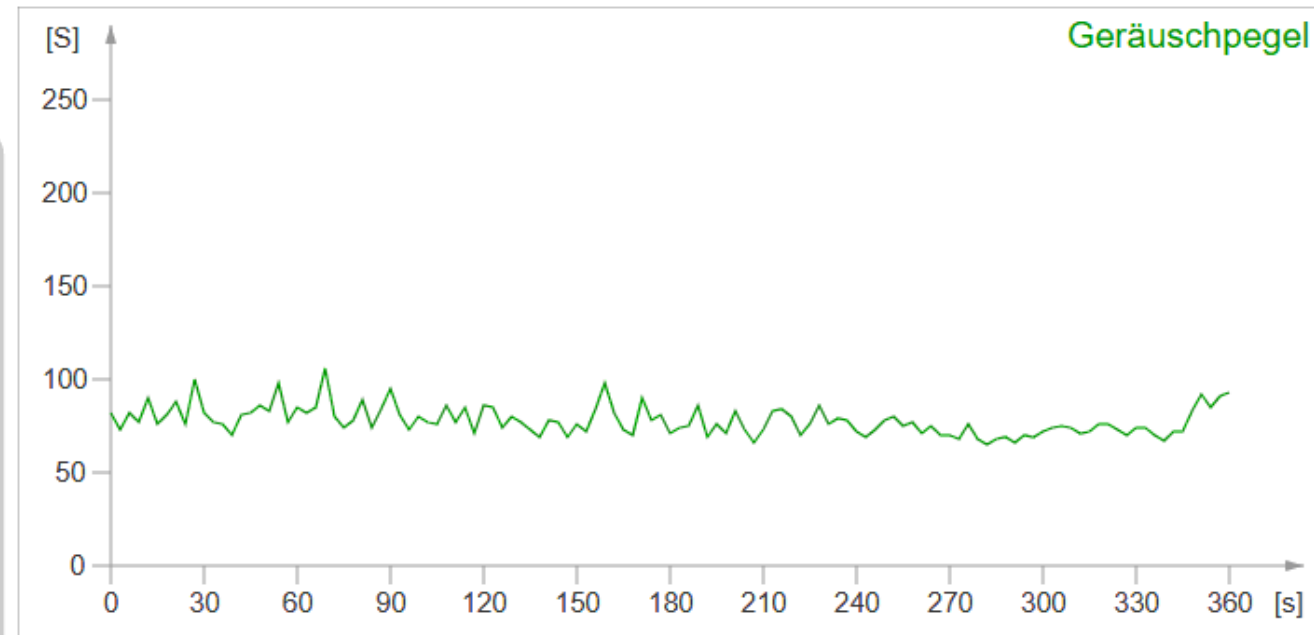
## Serielle Verbindung mit Calliope mini

[Programm für den Calliope mini zum Senden von Daten](#)

Mit Port verbinden und Daten abfragen

**Daten empfangen**  120 (360 s) **Verbindung trennen**

JSON-String	{ "L" : 58, "S" : 93, "T" : 25, "F" : 81 }	
Licht (0-255)	58	<input type="checkbox"/>
Sound (0-255)	93	<input checked="" type="checkbox"/>
Temperatur (°C)	25	<input type="checkbox"/>
Feuchtigkeit (%)	81	<input type="checkbox"/>



- ✓ Aus diesem Prototyp wurden weitere Projekte abgeleitet, u. a. zum längeren CO<sub>2</sub>-Monitoring.

# Daten seriell senden und empfangen 7/8

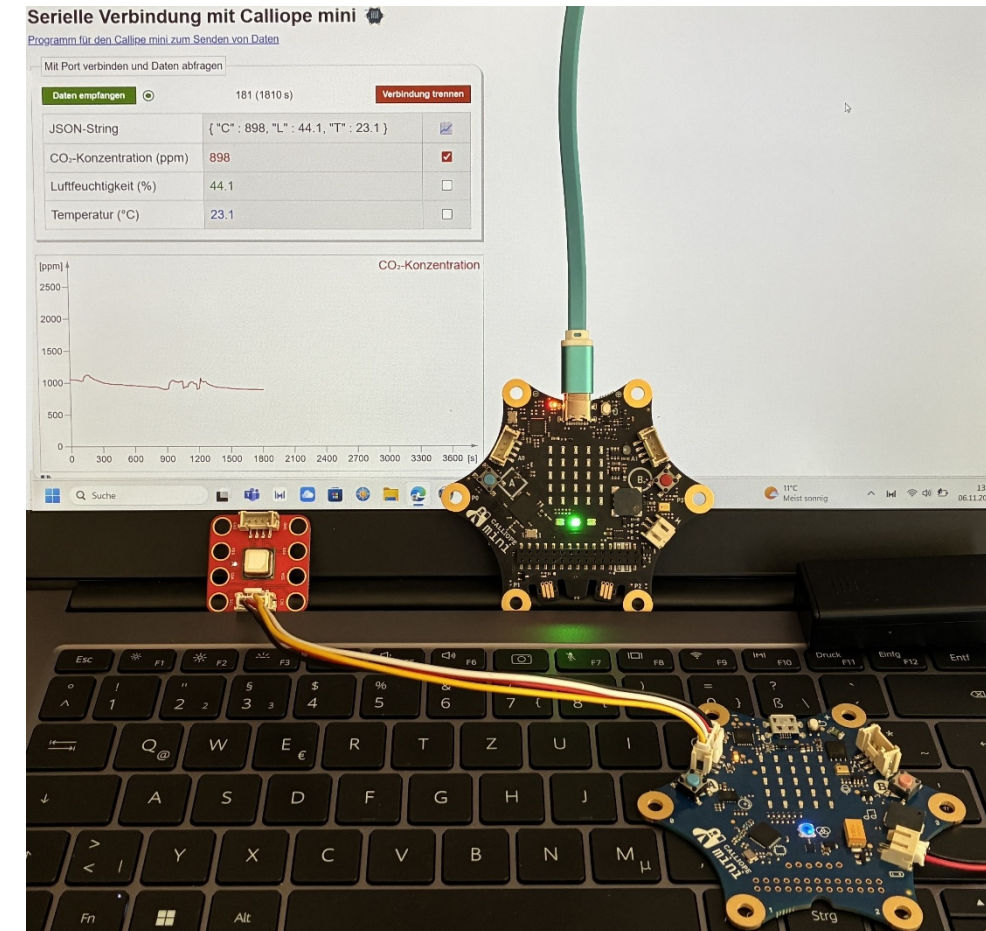
- Daten im Browser über die Web Serial API verarbeiten (4)

→ Live-Demo

- ✓ Erweitertes Szenario für CO<sub>2</sub>-Messung mit zwei über Funk (Bluetooth) verbundenen Calliope mini.
- ✓ Gerät A sammelt die Daten vom SCD40-Sensor und sendet diese periodisch an Gerät B, welches über die WSA mit dem Browser kommuniziert.

```
radio.setGroup(23)
radio.setTransmitPower(7)
// ...
function sendData(): void {
  let C: number = SCD40.get_co2() // ppm
  let L: number = SCD40.get_relative_humidity() // %
  let T: number = SCD40.get_temperature(SCD40.SCD40_T_UNIT.C) // °C
  let D: string = C + "," + L + "," + T
  radio.sendString(D)
}

// ...
radio.onReceivedString(function(receivedString: string) {
  // Auf Gerät B den empfangenen Datensatz verarbeiten
})
```

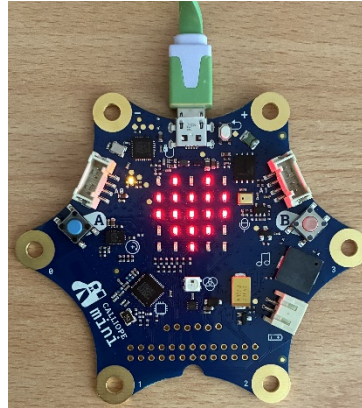


# Daten seriell senden und empfangen 8/8

- Daten im Browser über die Web Serial API verarbeiten (5) → Live-Demo
  - ✓ Daten aus dem Browser-Kontext an Calliope mini senden, Beispiel Ansteuerung der LED-Matrix.
  - ✓ Jede Belegung/Nichtbelegung der 25 im HTML-Formular vorgesehenen Check-Boxen sendet einen String der Form "0,0,1". Die ersten beiden Werte stehen für die LED-Position, hier oben links und der dritte Wert 1|0 bedeutet Ein- bzw. Ausschalten.
  - ✓ Zusätzlich kann das erzeugte Muster als Code kopiert und weiterverwendet werden.

```
// Daten vom Browser via WSA senden
// ...
writer = port.writable.getWriter();
const textencoder = new TextEncoder("UTF-8");
const data = textencoder.encode("0,0,1\n");
writer.write(data);
// ...
```

```
// Auf dem Calliope mini werden die übertragenen Positionen
// mittels led.plot(x,y) bzw. led.unplot(x,y) gesetzt.
```



## Serielle Verbindung mit Calliope mini

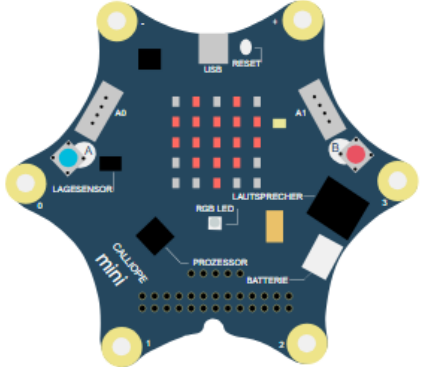
[Programm für den Calliope mini zum Empfangen von Daten](#)

Mit Port verbinden und Daten senden

**Daten senden**  Calliope mini ist verbunden. **Verbindung trennen**

LED-Matrix

<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**Reset**  **Kopieren**

Calliope-Code

```
basic.showLeds(`
. # . # .
# # # # #
# # # # #
. # # # .
. . # . .
`)
```

# Daten über WLAN senden

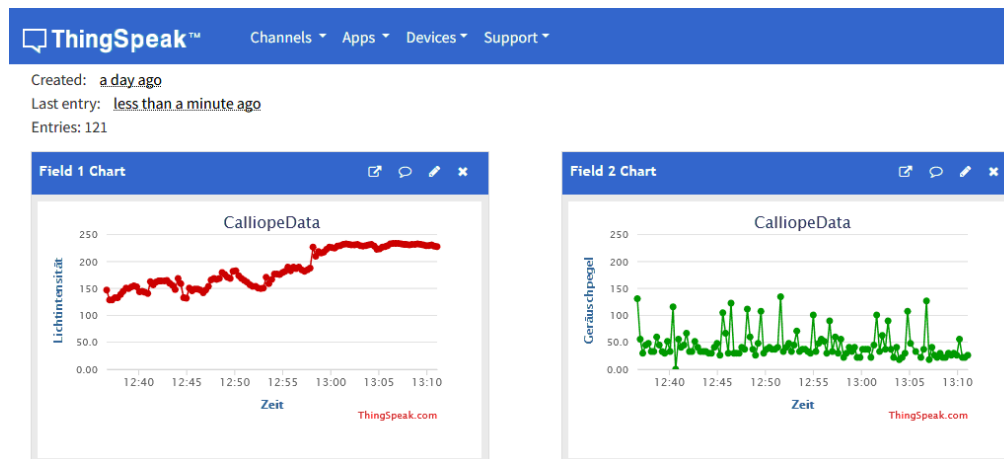


# Daten über WLAN senden <sup>1/3</sup>

- Datenübertragung mit ESP8266-Modul an die IoT-Plattform ThingSpeak
  - ✓ An ThingSpeak lassen sich Daten über WLAN senden. Dieser Dienst gehört zu MathWorks, bekannt als Hersteller der Software MATLAB. Ggf. vorhandene Konten sind kompatibel.
  - ✓ Es lassen sich diverse Diagramme und Widgets zur Visualisierung der übertragenen Daten einrichten.
  - ✓ Eine MakeCode-Erweiterung von Cytron Technologies ermöglicht die Kommunikation.

```
// Initialisierung
esp8266.init(SerialPin.P17, SerialPin.P16, BaudRate.BaudRate115200)
if(esp8266.isESP8266Initialized()) { esp8266.connectWiFi("ssid", "password") }

// Daten einlesen und senden
let L: number = input.lightLevel() // 0 (dark) to 255 (bright)
let S: number = input.soundLevel() // 0 (silent) to 255 (loud)
let T: number = input.temperature() // °C
esp8266.uploadThingspeak("writeApiKey", L, S, T, 0, 0, 0, 0)
if(esp8266.isThingspeakUploaded()) {
  // Erfolgsmeldung
  basic.pause(15000)
}
```



# Daten über WLAN senden <sup>2/3</sup>

- Erweiterung des ESP8266-Moduls für eigene Ziele (1)

→ Live-Demo

- ✓ Code der Cytron-Erweiterung wurde analysiert und mit Anpassungen im Namespace esp8266 ergänzt, um eigene Web-Server anzusprechen. Auf dem Zielsystem empfängt ein PHP-Skript die Daten als CSV-Datei.

```
const server: string = "example.net"
const script: string = "/calliope/calliope_csv.php"
// ...
namespace esp8266 {
  // ...
  // Mit dem Server verbinden, ggf. zurück
  if(sendCommand("AT+CIPSTART=\"TCP\", \"\" + server + "\",80", "OK", 10000) == false)
  return
  // Daten für GET-Request vorbereiten (hier wiederum Licht, Sound, Temperatur)
  let params: string = "?L=" + L + "&S=" + S + "&T=" + T
  let data: string = "GET " + script + params
    data += " HTTP/1.1\r\n"
    data += "Host: " + server + "\r\n"

  // Daten senden
  sendCommand("AT+CIPSEND=" + (data.length + 2))
  sendCommand(data)
  // ...
}
```

# Daten über WLAN senden <sup>3/3</sup>

- Erweiterung des ESP8266-Moduls für eigene Ziele (2)

→ Live-Demo

- ✓ PHP-Code zum Speichern der empfangenen Daten in einer CSV-Datei. Diese lässt sich dann weiterverarbeiten, etwa in eine HTML-Tabelle umformen.

```
<?php
$D = date("Y-m-d H:i:s", time());
$L = $_GET["L"];
$S = $_GET["S"];
$T = $_GET["T"];

if(!empty($D) && !empty($L) && !empty($S) && !empty($T)) {
    $file = fopen("data.csv", "a");
    fwrite($file, "$D;$L;$S;$T\n");
    fclose($file);
    echo "Ok";
}
else {
    echo "Error";
}
?>
```

## Calliope-Daten (WLAN)

Datum/Zeit	Licht	Sound	Temp
2026-03-22 19:24:07	43	111	26
2026-03-22 19:24:33	43	72	26
2026-03-22 19:24:57	45	75	26
2026-03-22 19:25:16	44	136	26
2026-03-22 19:25:37	44	72	26
2026-03-22 19:25:54	49	86	26
2026-03-22 19:26:11	48	104	26
2026-03-22 19:26:28	43	125	26
2026-03-22 19:26:45	42	96	26
2026-03-22 19:27:09	48	97	26

# Fazit und Ausblick



# Fazit und Ausblick

- Der Einplatinencomputer Calliope mini ist und bleibt speziell unter Nutzung zusätzlicher Peripherie-Komponenten ein attraktives Gerät für kreative Projekte im Rahmen der schulischen und universitären Ausbildung.
- Die Möglichkeit der Nutzung von JavaScript (oder auch Python) erweitert den Horizont zur Umsetzung eigener Ideen.
- Neuere Entwicklungen im Bereich der Web-Technologien wie Web Serial API (oder auch WebUSB API) schaffen zusätzliche Anreize zu deren Erkundung mit einem etablierten System.
- Die folgende Ressource liefert detaillierte Beschreibungen aller Projekte mit Code-Download: Meinike, T.: Externe Datenkommunikation mit Calliope mini – Serielle Übertragung von Daten und mittels WLAN. [https://datenverdrahten.de/PDF/Calliope\\_mini\\_Datenkommunikation.pdf](https://datenverdrahten.de/PDF/Calliope_mini_Datenkommunikation.pdf)

# Danke für die Aufmerksamkeit und viel Spaß bei eigenen Projekten!

