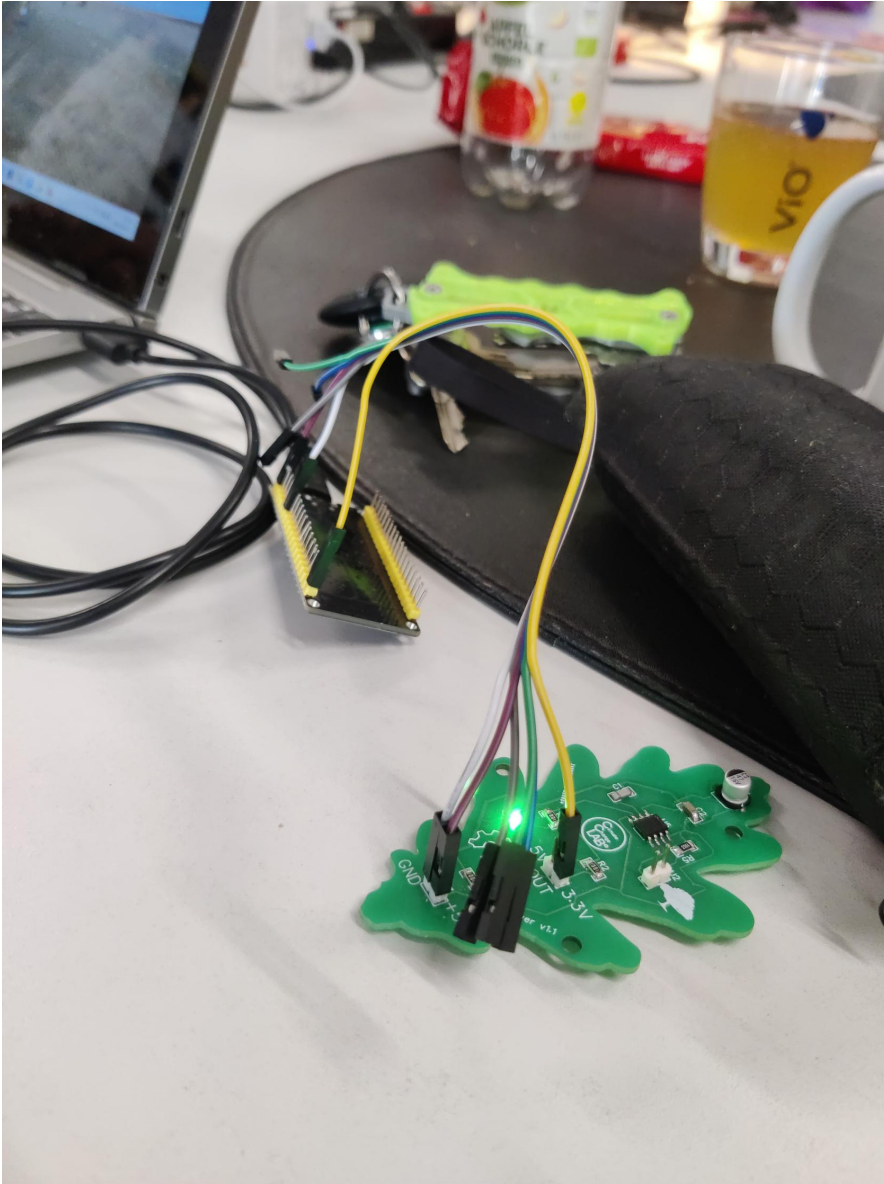


# Arduino IDE und ESP32 flashen

Deutsche Dokumentation der Originalquelle:

[http://wikifab.org/wiki/Capteur\\_BioData\\_pour\\_ESP32](http://wikifab.org/wiki/Capteur_BioData_pour_ESP32)



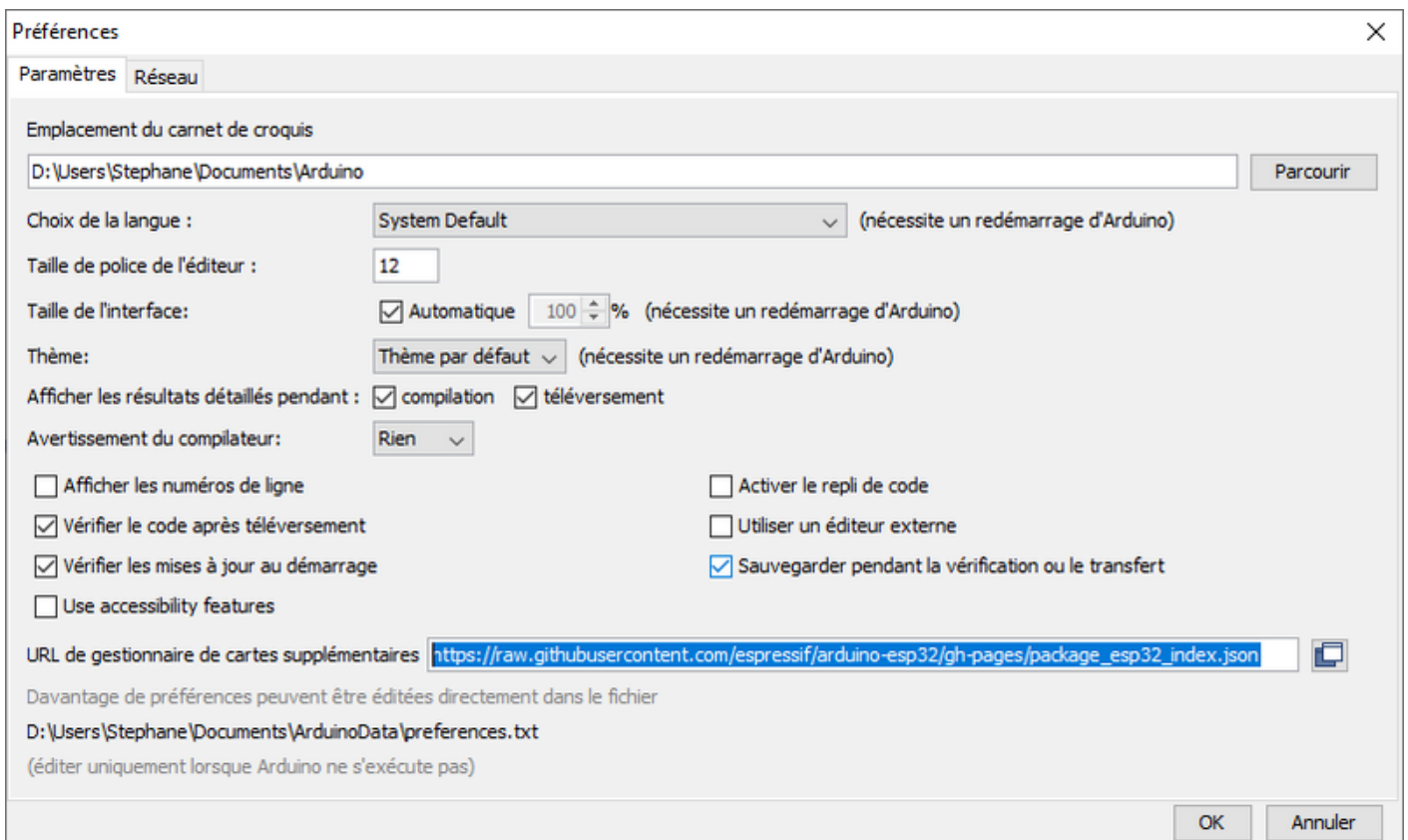
## Arduino IDE

Wir arbeiten mit der IDE Version 2.3.6.. Siehe <https://www.arduino.cc/en/software>. Wir installieren außerdem `python3-serial`, um die USB-Verbindung über die IDE zuzulassen:

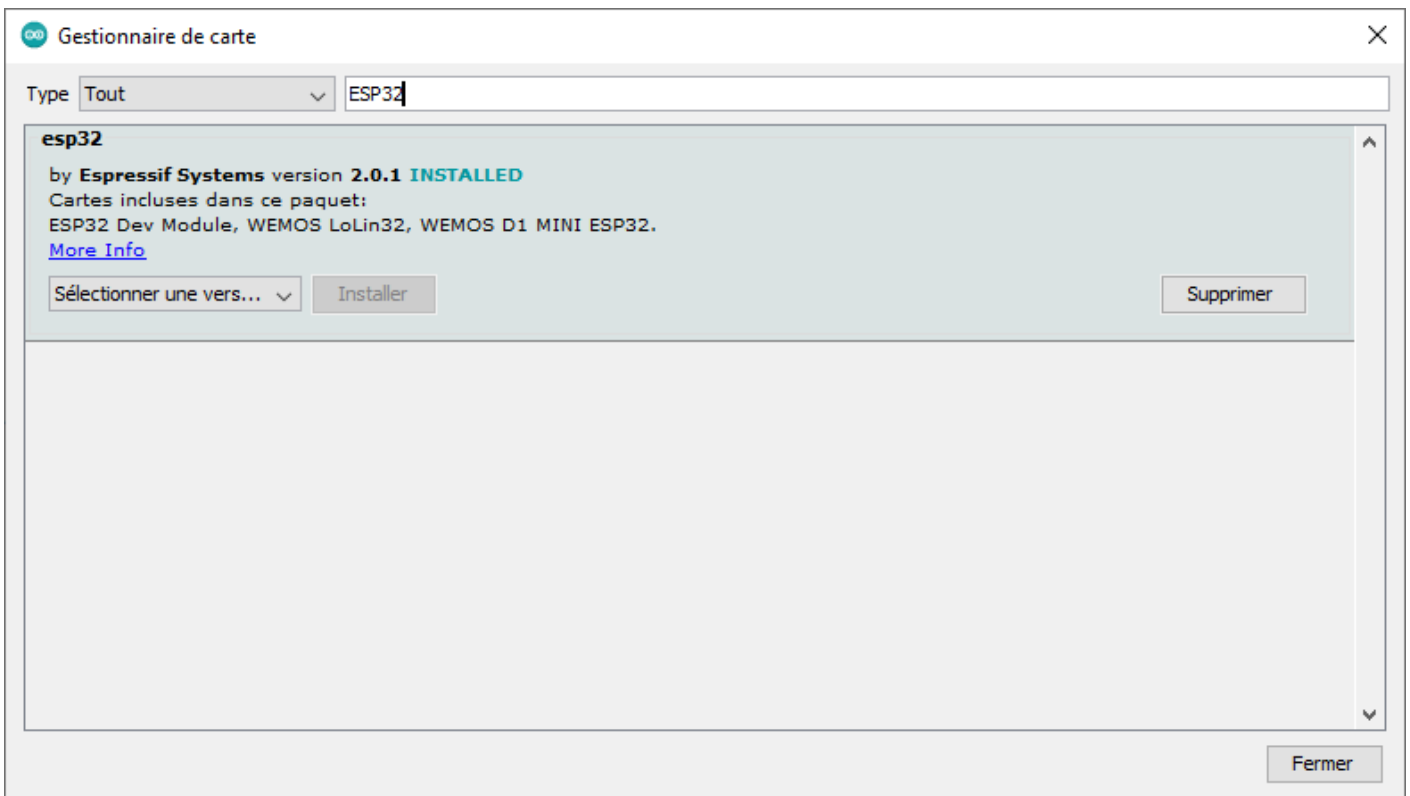
```
sudo apt install python3-serial
wget https://downloads.arduino.cc/arduino-ide/arduino-ide_2.3.6_Linux_64bit.AppImage
chmod +x arduino-ide_2.3.6_Linux_64bit.AppImage
mv arduino-ide_2.3.6_Linux_64bit.AppImage /opt/
```

## ESP32 Boards

Fügen Sie den ESP32 zur Arduino-IDE hinzu, indem Sie die Zeile [https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\\_esp32\\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json) hinzufügen:



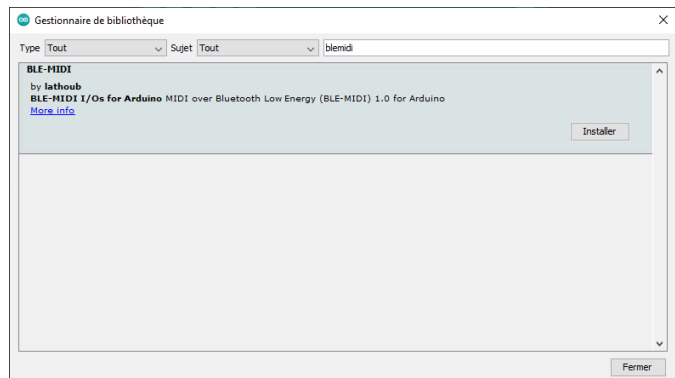
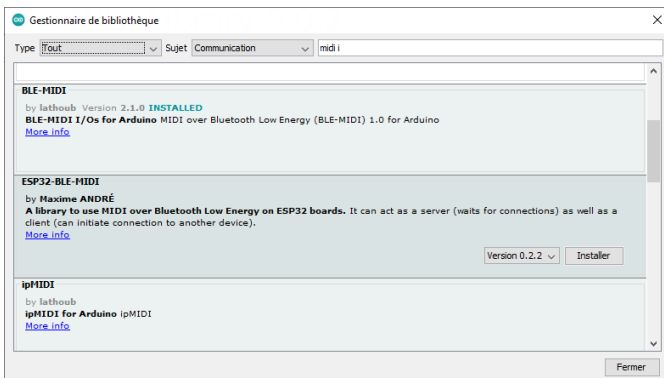
Laden Sie die ESP32-Erweiterung mit dem Board-Manager herunter:

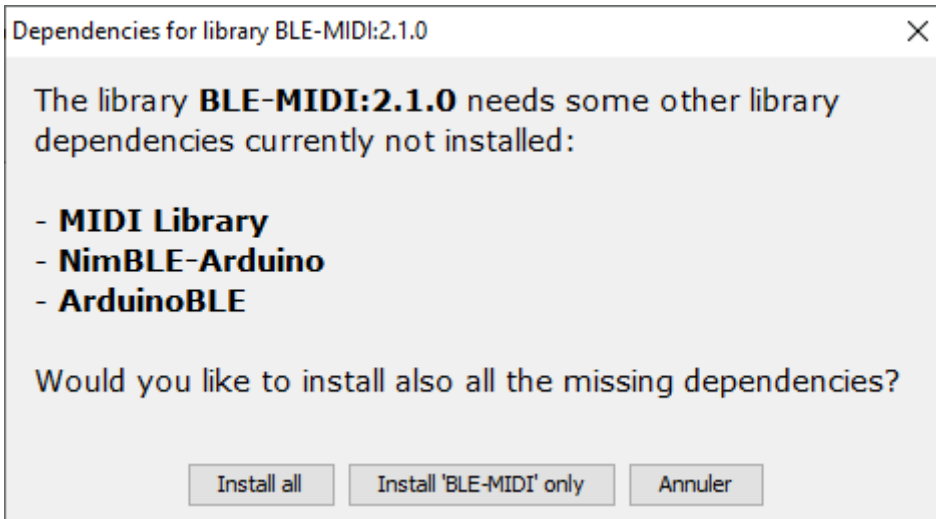


## Bibliothèques installer

Notwendige Bibliotheken und Versionen:

- NimBLE-Arduino 1.4.3
- ESP32-BLE-MIDI 0.3.2
- BLE-MIDI 2.2
- ArduinoBLE 1.4.1





## Kompilieren und Hochladen

Außerdem manuell das Projekt in deinen Library-Ordner (z.B. `~/Arduino/libraries/MidiFlower-0.2.0/`) entpacken

- die Originalversion des Hackathons:  
<https://github.com/crocsg/MidiFlower/releases/download/v0.1/midiflower.zip>
- **Die überarbeitete Version:**  
<https://github.com/crocsg/MidiFlower/archive/refs/tags/v0.2.0.zip>

**Kompilieren Sie den Sketch für einen ESP32 und laden Sie ihn auf den ESP32 hoch. (Möglicherweise müssen Sie die Taste „Boot“ drücken, um den Upload durchzuführen. Siehe Dokumentation des ESP32). Der Sketch ist hier verfügbar:**

## USB-Anschluss

Falls auf über USB nicht auf den Controller zugegriffen werden kann, fehlen u.U. die passenden Berechtigungen. Dies lässt sich wie folgt beheben:

```
sudo usermod -a -G dialout $(whoami)
```

Falls dies nicht ausreicht:

```
sudo vim /etc/udev/rules.d/99-usb-serial.rules
```

```
KERNEL=="ttyACM[0-9]*", SYMLINK+="ttyUSB%n", GROUP="dialout", MODE="0660"
```

```
sudo udevadm control --reload-rules
sudo udevadm trigger
```

Falls dies immer noch nicht ausreicht ein dreckiger Trick:

```
sudo chmod 666 /dev/ttyUSB0
```

## Modul-Auswahl, Partitionierung, Hochladen

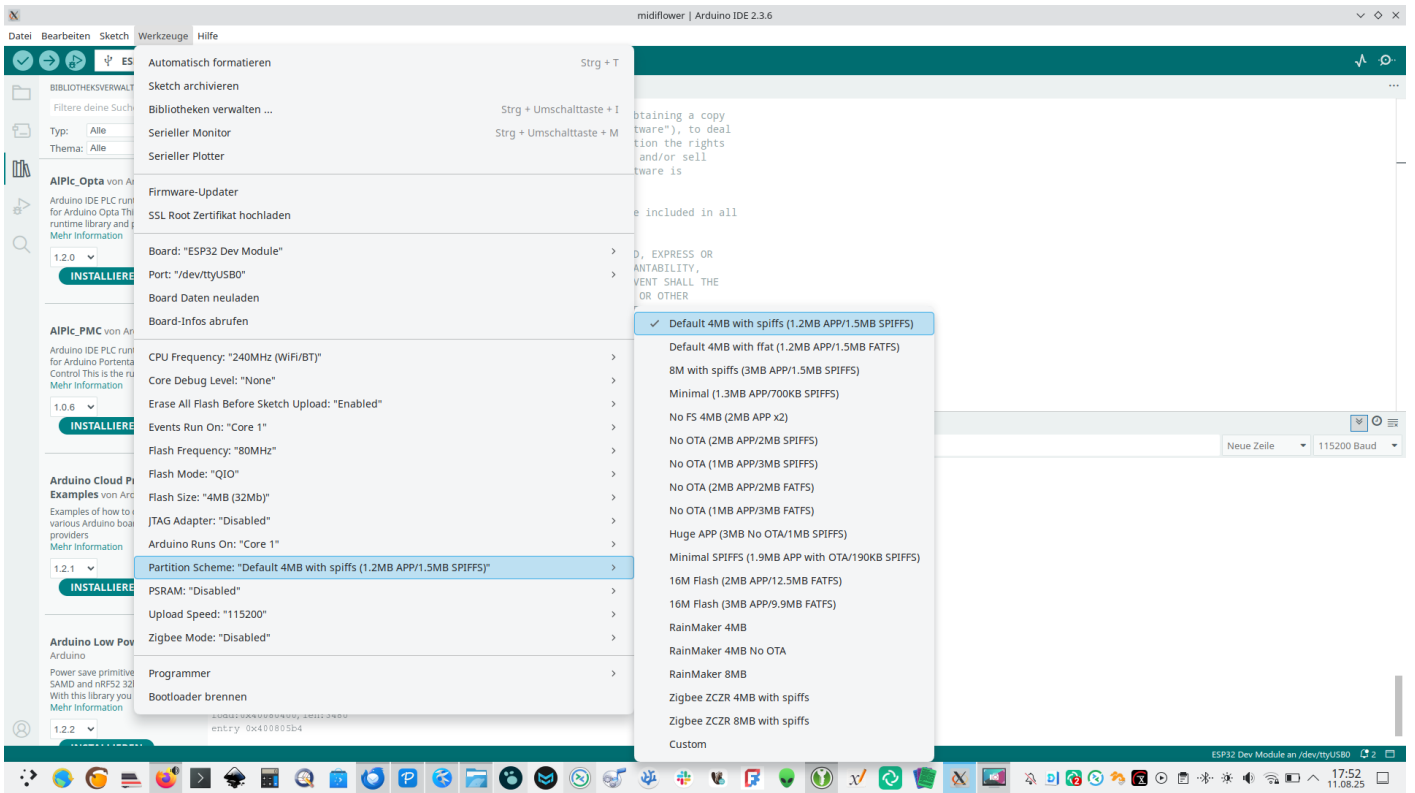
[Unser Modul](#) hat folgende Specs:

- Stromversorgungsspannung (USB): 5 V
- Logiklevel: 3.3 V
- Benötigter Betriebsstrom: min. 500 mA
- SoC: ESP32 WROOM 32
- Taktfrequenzbereich: bis 240 MHz
- RAM: 512 kB
- SPIFFS: 4 MB
- I/O Pins: 34
- Schnittstellen: SPI, I2C, I2S, CAN, UART
- Wi-Fi Protokolle: 802.11 b/g/n (802.11n bis zu 150 Mbps)
- Wi-Fi Frequenz: 2.4 GHz - 2.5 GHz
- Bluetooth: V4.2 - BLE und Classic Bluetooth
- Drahtlose Antenne: PCB
- Abmessungen: 56 x 28 x 13 mm

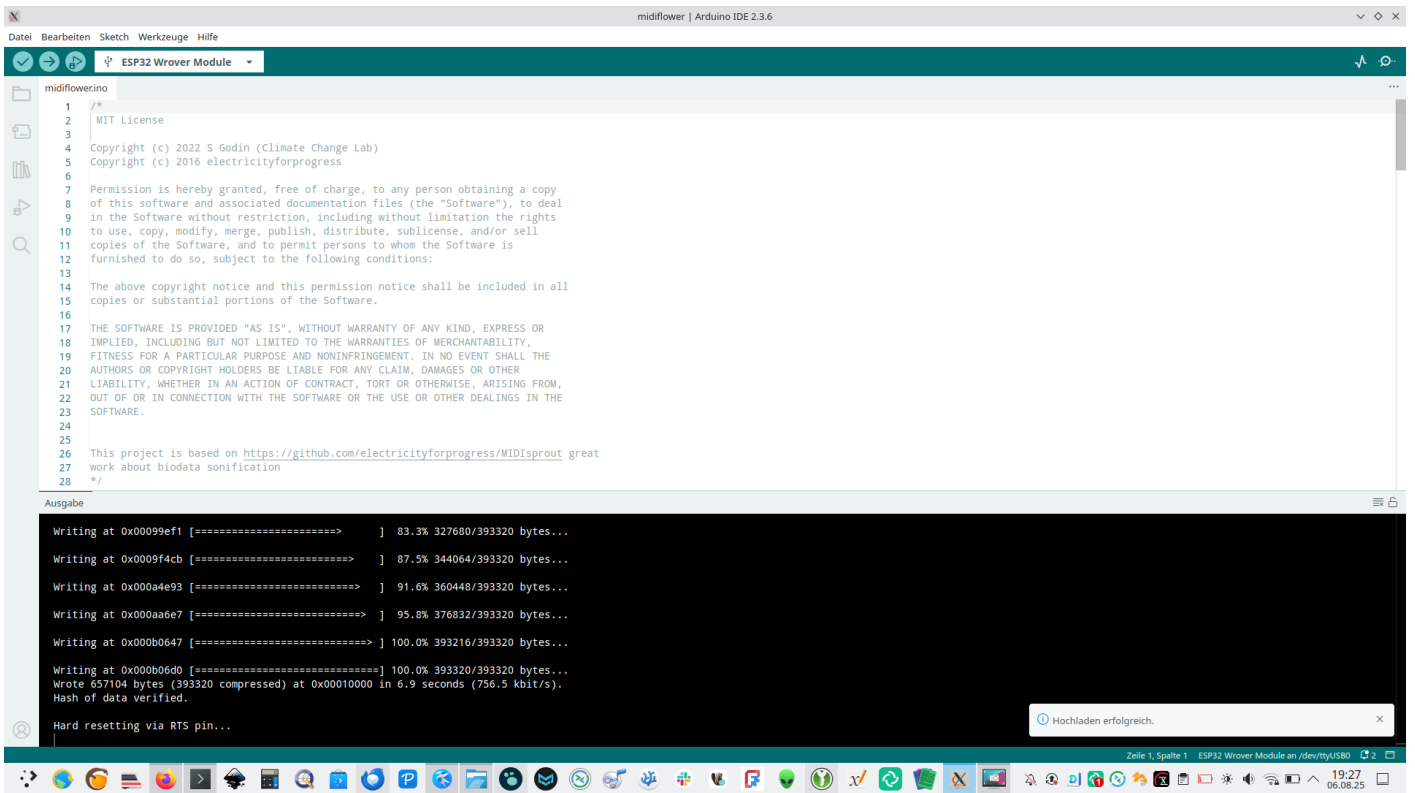
Folgendes Partitionierungsschema ist für's Flashen wichtig: `No OTA (2MB App / 2MB SPIFFS)`

Folgendes Partitionierungsschema ist für's Flashen wichtig: `Default 4MB with spiffs`

`(1.2MB APP/1.5MB SPIFFS)`



Wir selbst nutzen für unseren Flashing Vorgang ein **ESP32 Dev Kit C** von AZ-Delivery. Dies entspricht in Arduino IDE dem Entwicklerboard **ESP32 Wrover Module** oder **ESP32-WROOM-DA**. Wir flashen bei einer Baudrate von 115200.



## Flashing überprüfen

Über den Serial Monitor unter Tools können wir bei angeschlossenen USB-Port überprüfen, was die Ausgabe des Controllers ist. Mit Baudrate 115200 erhalten wir u.U. folgenden Output:

```
st:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0030,len:4980
load:0x40078000,len:16612
load:0x40080400,len:3480
entry 0x400805b4
E (229) esp_core_f0[]0000 Incorrect size of core dump image: 1
```

u.U. stimmt hier das Partitionierungsschema nicht. Überprüfen und neu kompilieren!

## Hinweise und Referenzen

Dieses Tutorial wurde dank der Arbeit von Sam Cusumano [electricityforprogress](https://github.com/electricityforprogress)

<https://github.com/electricityforprogress/MIDIsprout> erstellt. Die Originalarbeit von Sam unterliegt der Open-Source-Lizenz „MIT Licence“.

---

Version #15

Erstellt: 2025-08-07 11:36:26 CEST von Mario Voigt

Zuletzt aktualisiert: 2026-01-07 14:29:25 CET von Mario Voigt