

## Schritt 1: Home Assistant öffnen

1. Öffne dein **Home Assistant Dashboard**.
  2. Stelle sicher, dass du als Administrator angemeldet bist.
- 

## Schritt 2: ESPHome Add-on installieren

1. Gehe zu **Einstellungen** (das Zahnradsymbol).
  2. Wähle **Add-ons, Backups & Supervisor** aus.
  3. Klicke auf **Add-on Store**.
  4. Suche nach **ESPHome**.
  5. Klicke auf **ESPHome** und dann auf **Installieren**.
  6. Warte, bis die Installation abgeschlossen ist.
- 

## Schritt 3: ESPHome Device Builder hinzufügen

1. Nach der Installation von ESPHome, gehe zurück zum **Add-on Store**.
  2. Suche nach **ESPHome Device Builder**.
  3. Klicke auf **Installieren** und warte, bis die Installation abgeschlossen ist.
- 

## Schritt 4: ESPHome Device Builder konfigurieren

1. Öffne das **ESPHome Device Builder**-Add-on.
  2. Folge den Anweisungen auf dem Bildschirm, um dein erstes Gerät zu konfigurieren.
  3. Du kannst ein vorgefertigtes Gerät auswählen oder ein benutzerdefiniertes Gerät erstellen.
- 

## Schritt 5: Gerät mit Home Assistant verbinden

1. Sobald dein Gerät konfiguriert ist, erhältst du eine YAML-Datei.
  2. Lade diese YAML-Datei in das **ESPHome**-Add-on hoch.
  3. Flash dein ESP-Gerät mit der erstellten Firmware.
  4. Gehe zurück zu **Home Assistant** und füge das Gerät über **Integrationen** hinzu.
- 

## Schritt 6: Überprüfen

1. Überprüfe in **Home Assistant**, ob dein Gerät korrekt erkannt wurde.
2. Du kannst jetzt Automatisierungen und Steuerungen für dein Gerät erstellen.

---

## Hinweis:

- Stelle sicher, dass dein ESP-Gerät (z. B. ESP32 oder ESP8266) mit demselben Netzwerk wie Home Assistant verbunden ist.
- Wenn du Probleme hast, überprüfe die Logs in **ESPHome** und **ESPHome Device Builder**.

---

## Schritt-für-Schritt-Anleitung: ESP32-S3 mit ESPHome Device Builder und Home Assistant einrichten

---

### Vorbereitung

1. **Besorge dir einen ESP32-S3** (oder ein ähnliches ESP-Gerät).
2. **Stelle sicher, dass du ein USB-Kabel hast**, das Daten übertragen kann (nicht nur zum Laden).

---

### Schritt 1: Home Assistant öffnen

1. Öffne dein **Home Assistant Dashboard**.
2. Stelle sicher, dass du als **Administrator** angemeldet bist.

---

### Schritt 2: ESPHome Add-on installieren

1. Gehe zu **Einstellungen** (das Zahnradsymbol).
2. Wähle **Add-ons, Backups & Supervisor** aus.
3. Klicke auf **Add-on Store**.
4. Suche nach **ESPHome**.
5. Klicke auf **ESPHome** und dann auf **Installieren**.
6. Warte, bis die Installation abgeschlossen ist.

---

### Schritt 3: ESPHome Device Builder hinzufügen

1. Gehe zurück zum **Add-on Store**.
2. Suche nach **ESPHome Device Builder**.
3. Klicke auf **Installieren** und warte, bis die Installation abgeschlossen ist.

---

## Schritt 4: ESPHome Device Builder konfigurieren

1. Öffne das **ESPHome Device Builder**-Add-on.
2. Klicke auf **NEW DEVICE** (unten rechts).
3. Klicke auf **Continue**.
4. Gib deinem Gerät einen Namen (z. B. jbd-bms-ble).
5. Klicke auf **Next**.
6. Wähle das Gerät **ESP32-S3** aus.
7. Klicke auf **Skip**.
8. Klicke auf **Edit**, um den Quellcode (\*.yaml) zu öffnen.

```
× battery.yaml
1  esphome:
2    name: battery
3    friendly_name: Battery
4
5  esp32:
6    board: esp32-s3-devkitc-1
7    framework:
8      type: arduino
9
10 # Enable logging
11 logger:
12
13 # Enable Home Assistant API
14 api:
15   encryption:
16     key: "Ch9vFQ1MSt57cBfLUXNtDSTAUxAVMR/+bMNGYw08="
17
18 ota:
19   - platform: esphome
20     password: "01a52379ccc61049e8e1663017416bbf"
21
22 wifi:
23   ssid: !secret wifi_ssid
24   password: !secret wifi_password
25
26 # Enable fallback hotspot (captive portal) in case wifi connection fails
27 ap:
28   ssid: "Battery Fallback Hotspot"
29   password: "VpezChAKKJEq"
30
31 captive_portal:
32
```

9.

---

## Schritt 5: Quellcode anpassen

1. Gehe zur GitHub-Seite: [esphome-jbd-bms](#).
2. Wähle eine Beispiel-Datei aus, z. B.:
  - [esp32-ble-example-multiple-devices.yaml](#)
  - [esp32-ble-example.yaml](#)
3. Kopiere die folgenden Abschnitte aus dem GitHub-Code in deinen eigenen Code:
  - **Zeile 5 bis 8:** esp32: ...
  - **Zeile 14 bis 16:** api: ...
  - **Zeile 18 bis 20:** ota: ...

```

1 platform:
2   name: battery
3   friendly_name: Battery
4
5   esp32:
6     board: esp32-s3-devkitc-1
7     framework:
8       type: arduino
9
10    # Enable logging
11    logger:
12
13    # Enable Home Assistant API
14    api:
15      encryption:
16        key: "Ch9v0fQ1M5t57cBfLUXNiTDSTAUxAVNR/+bWNGYwo8="
17
18    ota:
19      platform: esphome
20      password: "01a52379ccc61049e8e1663017416bbf"
21
22    wifi:
23      ssid: !secret wifi_ssid
24      password: !secret wifi_password
25
26    # Enable fallback hotspot (captive portal) in case wifi connection fails
27    ap:
28      ssid: "Battery Fallback Hotspot"
29      password: "VpezCNAKXJEQ"
30
31    captive_portal:
32
33
34
35    substitutions:
36    name: jbd-bms-ble
37    device_description: "Monitor and control a Xiaoxiang Battery Management System (JBD-BMS) via BLE"
38    external_components_source: github://syssi/esphome-jbd-bms@main
39    mac_address: 70:3e:97:07:c0:3e
40
41    esphome:
42      name: ${name}
43      comment: ${device_description}
44      min_version: 2024.6.0
45      project:
46        name: "syssi.esphome-jbd-bms"

```

4. Überschreibe die entsprechenden Zeilen in deinem Code.

```

54
55 logger:
56   level: DEBUG
57
58 # If you use Home Assistant please remove this `mqtt` section and uncomment the `api` component!
59 # The native API has many advantages over MQTT: https://esphome.io/components/api.html#advantages-over-mqtt
60 #mqtt:
61 # broker: !secret mqtt_host
62 # username: !secret mqtt_username
63 # password: !secret mqtt_password
64 # id: mqtt_client
65
66 api:
67   encryption:
68     key: "Ch9v0fQ1M5t57cBfLUXNiTDSTAUxAVNR/+bWNGYwo8="
69

```

5.

6. Lösche den ursprünglichen GitHub-Code, nachdem du die benötigten Abschnitte kopiert hast.

```

1 esphome:
2   name: battery
3   friendly_name: Battery
4
5   esp32:
6     board: esp32-s3-devkitc-1
7     framework:
8       type: arduino
9
10    # Enable logging
11    logger:
12
13    # Enable Home Assistant API
14    api:
15      encryption:
16        key: "Ch9v0fQ1M5t57cBfLUXNiTDSTAUxAVNR/+bWNGYwo8="
17
18    ota:
19      platform: esphome
20      password: "01a52379ccc61049e8e1663017416bbf"
21
22    wifi:
23      ssid: !secret wifi_ssid
24      password: !secret wifi_password
25
26    # Enable fallback hotspot (captive portal) in case wifi connection fails
27    ap:
28      ssid: "Battery Fallback Hotspot"
29      password: "VpezCNAKXJEQ"
30
31    captive_portal:
32
33
34
35    substitutions:
36    name: jbd-bms-ble
37    device_description: "Monitor and control a Xiaoxiang Battery Management System (JBD-BMS) via BLE"
38    external_components_source: github://syssi/esphome-jbd-bms@main
39    mac_address: 70:3e:97:07:c0:3e
40
41    esphome:
42      name: ${name}
43      comment: ${device_description}
44      min_version: 2024.6.0
45      project:
46        name: "syssi.esphome-jbd-bms"

```

7.

8. Klicke auf **Save**, um die Änderungen zu speichern.

**Schritt 6: Bluetooth-Adresse der Batterie hinzufügen**

1. Lade die **BAT-BMS-App** auf dein Smartphone herunter.
2. Öffne die App und suche nach der Bluetooth-Adresse deiner Batterie (z. B. JBD 70:3E:97:07:C0:3E).

3. Trage diese Adresse in deinen Quellcode ein.

```
1 substitutions:
2 name: jbd-bms-ble
3 device_description: "Monitor and control a Xiaoxiang Battery Management System (JBD-BMS) via BLE"
4 external_components_source: github://syssi/esphome-jbd-bms@main
5 mac_address: 70:3e:97:07:c0:3e
6
7 esphome:
8 name: ${name}
9 comment: ${device_description}
```

- 4.
5. Klicke auf **Save**, um die Änderungen zu speichern.

### Schritt 7: Firmware auf den ESP32-S3 flashen

1. Verbinde den **ESP32-S3** über USB mit deinem Home Assistant-PC.
2. Klicke im ESPHome Device Builder auf **Install**.
3. Wähle **Plug into computer running ESPHome Device Builder**.
4. Wähle den USB-Port aus (z. B. /dev/ttyUSB0).

```
Install battery.yaml
INFO ESPHome 2024.12.4
INFO Reading configuration /config/esphome/battery.yaml...
INFO Updating https://github.com/syssi/esphome-jbd-bms.git@main
INFO Generating C++ source...
INFO Compiling app...
Processing jbd-bms-ble (board: esp32-s3-devkit-1; framework: arduino; platform: platformio/espressif32@5.4.0)
-----
Library Manager: Installing esphome/noise-c @ 0.1.6
INFO Installing esphome/noise-c @ 0.1.6
Unpacking [#####] 100%
Library Manager: noise-c@0.1.6 has been installed!
INFO noise-c@0.1.6 has been installed!
Library Manager: Resolving dependencies...
INFO Resolving dependencies...
Library Manager: Installing esphome/ibutton@1.1.0
INFO Installing esphome/ibutton @ 1.1.0
Unpacking [#####] 100%
Library Manager: ibutton@1.1.0 has been installed!
INFO ibutton@1.1.0 has been installed!
Hardware: ESP32S3 240MHz, 320KB RAM, 8MB Flash
toolchain-riscv32-esp @ 8.4.0-202312-patch5
toolchain-riscv-esp32 @ 8.4.0-202312-patch5
Dependency Graph
|-- WiFi @ 2.0.0
|-- ESP8266 @ 2.0.0
|-- Update @ 2.0.0
|-- noise-c @ 0.1.6
```

- 5.
6. Warte, bis der Flashing-Prozess abgeschlossen ist.

```
Install battery.yaml
Flash [#####] 70.3% (100.0 KiB) (100.0 KiB) (100.0 KiB)
Writing jbd-bms-ble-ble-firmware.bin
Creating esp32s3 image...
Successfully created esp32s3 image.
esp32_create_binfile_bin[["platformio/jbd-bms-ble/firmware.bin"], ["platformio/jbd-bms-ble/firmware.bin"]]
Write 1000000 bytes to file /config/esphome/jbd-bms-ble/firmware.bin ready to flash to flash to offset 0x0
esp32_copy_to_bin[["platformio/jbd-bms-ble/firmware.bin"], ["platformio/jbd-bms-ble/firmware.bin"]]
===== [#####] 100.0% (100.0 KiB) (100.0 KiB) (100.0 KiB) =====
INFO Successfully compiled program.
Applying 0x...
Serial port /dev/ttyUSB0
Connecting...
Chip is ESP32-S3 (QFN55) (revision v0.2)
Features: WiFi, BLE, embedded PSRAM (8M (8M))
Crystal is 40MHz
MAC: 70:3e:97:07:c0:3e
Uploading data...
Merging stub...
Stub uploading...
Changing baud rate to 460800
Change...
Configuring flash size...
Auto-detected flash size: 8MB
Flash will be erased from 0x00000000 to 0x0000ffff...
Compressed 1000000 bytes to 800000...
Write 1000000 bytes (800000 compressed) at 0x00000000 in 20.4 seconds (effective 39.2 Kbit/s)...
Hash of data verified.
Merging: Merge file at 0x1 protected with a hash checksum, to not changing the flash size setting, use the --flash_size_wear option instead of --flash_size...
Flash checksum
Compressed 1000000 bytes to 800000...
Write 1000000 bytes (800000 compressed) at 0x00000000 in 0.5 seconds (effective 257.3 Kbit/s)...
Hash of data verified.
Compressed 800000 bytes to 344...
Write 800000 bytes (344 compressed) at 0x00000000 in 0.1 seconds (effective 456.7 Kbit/s)...
Hash of data verified.
Compressed 344 bytes to 47...
Write 344 bytes (47 compressed) at 0x00000000 in 0.1 seconds (effective 573.6 Kbit/s)...
Hash of data verified.
Merging...
Send requesting via UART pin...
INFO Successfully uploaded program.
INFO Starting log upload from /config/esphome/jbd-bms-ble/firmware.bin with baud rate 115200
[11:30:11] ESP32-S3 [esp32s3] 20241227
[11:30:11] Serial: /dev/ttyUSB0
[11:30:11] [11:30:11] (POWERON), boot: boot (SPI_FLASH_BOOT)
[11:30:11] [11:30:11]
[11:30:11] mode: 0x0, clock: 40.0
```

- 7.

### Schritt 8: Verbindung überprüfen

1. Klicke auf **Logs**, um zu überprüfen, ob die Verbindung erfolgreich war.
2. Wenn **"no connected"** angezeigt wird, schließe die BAT-BMS-App auf deinem Smartphone, da nur ein Gerät gleichzeitig verbunden sein kann.

- Überprüfe, ob die Daten der Batterie angezeigt werden.

---

## Schritt 9: Gerät in Home Assistant hinzufügen

- Gehe in Home Assistant zu **Einstellungen > Geräte und Dienste**.
- Suche nach deinem neu installierten Gerät (z. B. jbd-bms-ble).
- Klicke auf **Hinzufügen**.
- Überprüfe die Daten in der **Übersicht**.

5.

Übersicht	
jbd-bms-ble max cell voltage	3,255 V
jbd-bms-ble max voltage cell	12
jbd-bms-ble min cell voltage	3,249 V
jbd-bms-ble min voltage cell	1
jbd-bms-ble nominal capacity	104,70 Ah
jbd-bms-ble operation status	Unbekannt
jbd-bms-ble operation status bitmask	3
jbd-bms-ble power	259,7 W
jbd-bms-ble short circuit error count	Unbekannt
jbd-bms-ble software version	2,8
jbd-bms-ble state of charge	9 %
jbd-bms-ble temperature 1	22,7 °C
jbd-bms-ble temperature 2	25,2 °C
jbd-bms-ble temperature 3	22,3 °C
jbd-bms-ble temperature 4	22,0 °C
jbd-bms-ble temperature 5	21,9 °C
jbd-bms-ble temperature 6	22,2 °C
jbd-bms-ble total voltage	52,05 V

  

jbd-bms-ble bms0 average cell voltage	3,0472 V
jbd-bms-ble bms0 balancing	Aus
jbd-bms-ble bms0 charging	Ein
jbd-bms-ble bms0 delta cell voltage	0,0110 V
jbd-bms-ble bms0 device model	Unbekannt
jbd-bms-ble bms0 discharging	Ein
jbd-bms-ble bms0 errors	
jbd-bms-ble bms0 operation status	Unbekannt
jbd-bms-ble bms0 power	226,2 W
jbd-bms-ble bms0 state of charge	2 %
jbd-bms-ble bms0 total voltage	48,75 V
jbd-bms-ble bms1 average cell voltage	3,0465 V
jbd-bms-ble bms1 balancing	Aus
jbd-bms-ble bms1 charging	Ein
jbd-bms-ble bms1 delta cell voltage	0,0120 V
jbd-bms-ble bms1 device model	Unbekannt
jbd-bms-ble bms1 discharging	Ein
jbd-bms-ble bms1 errors	
jbd-bms-ble bms1 operation status	Unbekannt
jbd-bms-ble bms1 power	222,3 W
jbd-bms-ble bms1 state of charge	2 %
jbd-bms-ble bms1 total voltage	48,74 V

---

## Hinweise

- WLAN-Daten korrigieren:** Falls du beim ersten Einrichten einen Fehler gemacht hast, kannst du die WLAN-Daten unter **SECRETS** korrigieren. Vergiss nicht, die Änderungen zu speichern.
  - USB-Kabel:** Verwende ein USB-Kabel, das Daten übertragen kann, sonst funktioniert das Flashing nicht.
-