

# Temperatur-, relative Luftfeuchte-, Strahlungsintensitätslogger

## Anleitung

Erstellt von Daniel Hahn, MUT  
Stand: 5. Juni 2025

### Inhaltsverzeichnis

<b>1. Spezifikationen .....</b>	<b>2</b>
<b>2. Aufbau.....</b>	<b>3</b>
<b>3. Aufladen .....</b>	<b>3</b>
<b>4. RTC-Batteriewechsel.....</b>	<b>4</b>
<b>5. Messintervall setzen .....</b>	<b>4</b>
<b>6. Erwartete Laufzeit .....</b>	<b>4</b>
<b>7. Verbesserungen .....</b>	<b>4</b>

## 1. Spezifikationen

### SHT31 Temperatur- und Luftfeuchtesensor

- Hersteller: Sensirion AG
- Messunsicherheit:  $\pm 0.3\text{ °C}$  (Temperatur),  $\pm 2\%$  rF (Luftfeuchtigkeit)
- Kalibrierung erfolgte werkseitig durch den Hersteller; Bestätigung durch Vergleichsmessung dreier Logger (s. GitHub daniel.hahn: logger\_cal.txt)

### DS1302 Real-Time Clock

- Hersteller: Anzado GmbH
- Genauigkeit:  $\pm 20\text{ ppm}$  bei  $25\text{ °C}$  (entspricht ca.  $\pm 1\text{ Minute/Monat}$ )
- Kalibrierung durch manuelle Zeitsetzung beim Flashen im Arduino-IDE Sketch

### ESP32 NodeMCU Development Board

- Hersteller: Espressif Systems
- Chipsatz: ESP-WROOM-32
- USB-Serial-Chip: CP2102
- Kommunikationsschnittstellen: SPI, I2C, UART
- Weitere Funktionen: 12-bit ADC, PWM, Wi-Fi, Bluetooth

### Strahlungsintensitätssensor (durch Solarzelle)

- Typ: 1 V, 0.2 W, Größe:  $40 \times 40 \times 2\text{ mm}$
- Verbindung: Über Lastwiderstand an GND
- Messprinzip:

$$\text{Strahlungsintensität } \left( \frac{\text{W}}{\text{m}^2} \right) = \left( \frac{U_{PV}}{U_{STC}} \right) \times G_{STC}$$

Mit:

$U_{PV}$ : gemessene Spannung des Solarpanels

$U_{STC}$ : maximale Spannung bei  $1000\text{ W/m}^2$  (1.0 V)

$G_{STC}$ : Referenz-Strahlungsintensität ( $1000\text{ W/m}^2$ )

### Stromversorgung

- Akku: Li-Ion (Panasonic NCR18650), Kapazität: 3450 mAh
- Laderegler: CHB-LIT1A von BerryBase
- 5V DC/DC Konverter von Anzado GmbH

## Mikro-SD-Kartenmodul: MSD-AADP

- Hersteller: BerryBase
- Unterstützt FAT16/32, bis 32 GB
- Kommunikation über SPI

Weitere Informationen sind den jeweiligen Datenblättern der Hersteller zu entnehmen.

Skripte für die Mikrocontroller der jeweiligen Logger wurden mit der Arduino IDE (Arduino 2024) entwickelt und können in GitHub (daniel.hahn) eingesehen werden.

## 2. Aufbau

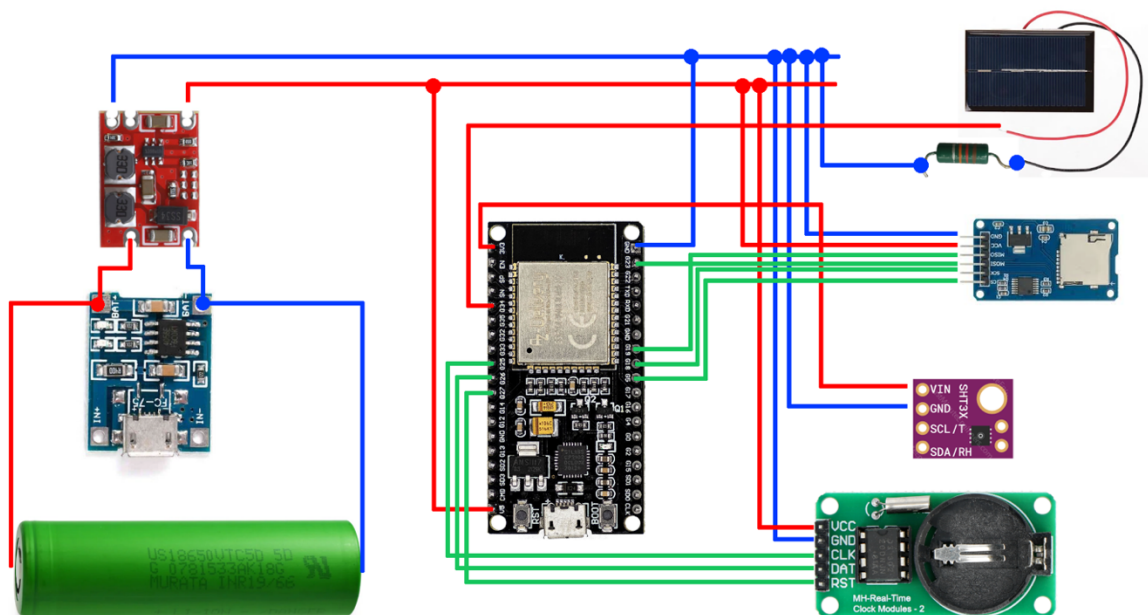


Figure 1: Schematischer Aufbau des (Temperatur-, relative Luftfeuchte-, Strahlungsintensitäts-) Loggers; Grüne Linien: Datenleitungen, Blau: Erdungen, Rot: 3,3V bzw. 5V

## 3. Aufladen

1. Logger ausschalten
2. Per MicroUSB mit 5V DC laden
3. Ladevorgang: blaues Licht
4. Fertig geladen: grünes Licht

## 4. RTC-Batteriewechsel

1. Batterie austauschen
2. Codezeile in setup() unkommentieren -> `setRTCFromCompileTime();`
3. Flashen des MCU
4. Codezeile in setup() kommentieren -> `//setRTCFromCompileTime();`
5. Erneutes Flashen des MCU

## 5. Messintervall setzen

1. **Intervall in Sekunden** in Codezeile verändern ->  
`esp_sleep_enable_timer_wakeup(SekundenULL * FaktorULL);`  
`esp_sleep_enable_timer_wakeup(3600ULL * 1000000ULL);`
2. Flashen des MCU

## 6. Erwartete Laufzeit

Stromaufnahme im aktiven Zustand  
(Messung, Schreiben auf SD-Karte):

ca. 100mA für ca. 5s

Stromaufnahme im DeepSleep Zustand:

ca. 0.1mA für Intervall

→ Laufzeit:            Akkukapazität/ Stromaufnahme pro Stunde  
                          3500mAh/ 0,239mAh/h = 14 Tage

→ Reale Laufzeit:    für Logger ohne Strahlungsintensität: 11d  
                          für Logger mit Strahlungsintensität: 4d

## 7. Verbesserungen

Mögliche Verbesserungen der Datenlogger sind dem Bericht zur Projektstudie (Earthship-Simulation, Hahn 2025) zu entnehmen.