

# 10. UV Strahlung

## Hintergrund

UV-Strahlung (ultraviolette Strahlung) ist eine Strahlung im Bereich 100 bis 400 nm. Dieser Bereich wird weiter unterteilt in UVA- (315–400 nm), UVB- (280–315 nm) und UVC-Strahlung (100–280 nm). Beim Durchgang der Sonnenstrahlung durch die Erdatmosphäre werden nahezu die gesamte UVC- und etwa 90 % der UVB-Strahlung absorbiert. Daher besteht die die Erdoberfläche erreichende UV-Strahlung der Sonne zu einem großen Teil aus UVA mit einem geringen UVB-Anteil<sup>1)</sup>.

Der **UV-Index** ist eine durch die Weltgesundheitsorganisation (WHO) und der Weltorganisation für Meteorologie (WMO) international standardisierte Maßzahl für die Stärke der UV-Strahlung der Sonne. Dieser wird aus der gemessenen UV-Strahlung errechnet. Der UV-Index soll dem Menschen helfen, die Sonnenstrahlung einzuschätzen und mit geeigneten Maßnahmen sich vor den gefährlichen Strahlung zu schützen. Je höher der Indexwert ist, desto schädlicher ist die Strahlung. Der UV-Index ist unabhängig vom Hauttyp<sup>2)</sup>.

Der Deutsche Wetterdienst veröffentlicht eine tagesgenaue Vorhersage für den UV-Index auf seiner Webseite: <https://www.dwd.de/DE/leistungen/gefahrenindizesuvi/gefahrenindexuvi.html>.

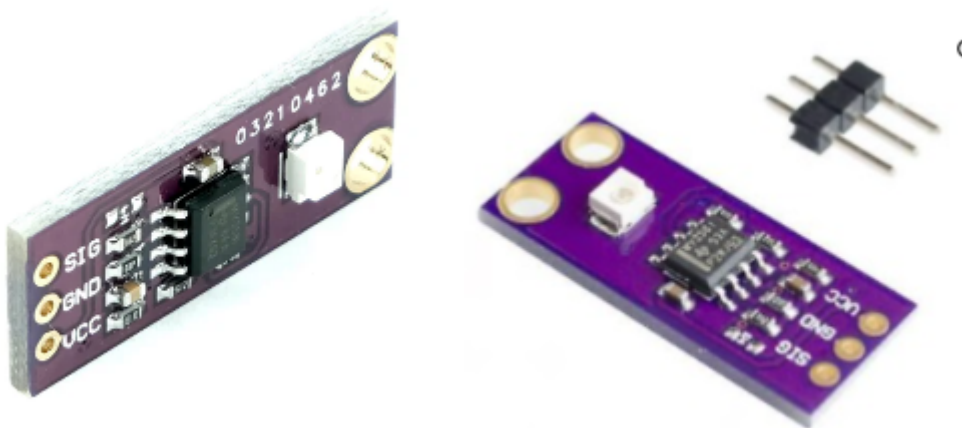
In unserer Wetterstation verwenden wir einen Sensor, der als Analogwert den UV-Index ausgibt, sodass eine manuelle (sehr komplizierte) Berechnung entfällt.

## Einkaufsliste

Komponente	Preis	Bezugsquelle (Beispiel)
(Analog-Digital-Wandler ADS1115 - nur 1x notwendig für alle Sensoren!)	(6 EUR)	<a href="#">BerryBase</a>
GUVA S12S UV-Sensor	7 EUR	<a href="#">Amazon</a>
<b>GESAMT</b>	<b>7 EUR</b>	

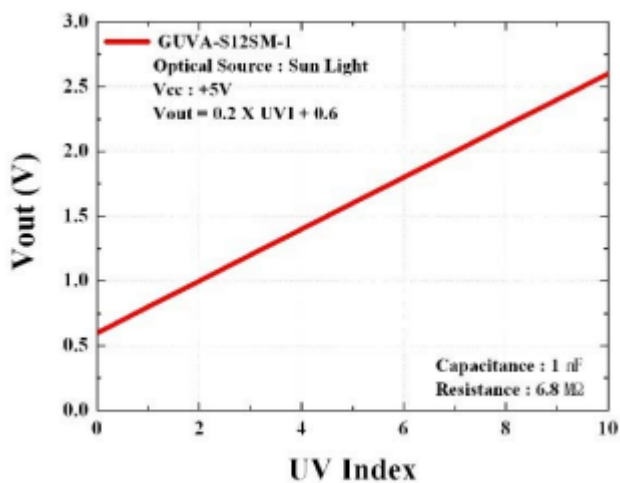
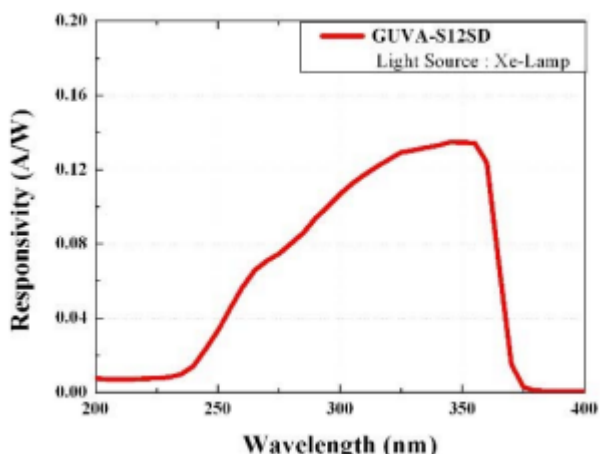
## Sensorauswahl

Wir verwenden den **UV-Sensor GUVA S12S**, den es von verschiedenen Herstellern und in verschiedenen Bauarten als Sensor-Modul zum Anschluss an Raspberry/Arduino gibt. Unser Gehäuse ist vorbereitet für die längliche Bauform mit zwei Bohrungen am oberen Ende - wenn ihr das Gehäuse/die Platine nicht ändern wollt, achtet darauf, dass ihr diese weit verbreitete Bauform bestellt.



### Längliche Bauart

Der Sensor misst im relevanten Wellenlängenbereich der UVA- und UVB-Strahlung (280 - 400 nm), die Ausgabe des Sensors ist linear zum UV-Index:



Ich habe den Sensor gegen die FOSHK-Wetterstation meines Nachbarn verglichen und auch noch einmal gegen die Angaben zum UV-Index des DWD verglichen (nur Tageswerte - ich habe 12 Uhr mittags im Sommer verglichen). Die Werte stimmen sehr gut überein.

## Technische Daten:

- Versorgungsspannung 2,5 V - 5 V
- Hohe Empfindlichkeit
- Erfassungsbereich: 240nm-370nm
- Winkel: 130 grad
- Größe: 11mm × 27mm
- Ausgang: 0-1V (entspricht UV Index 1-10).

## Datenblatt:

<https://cdn-shop.adafruit.com/datasheets/1918guva.pdf>

## Gehäuse

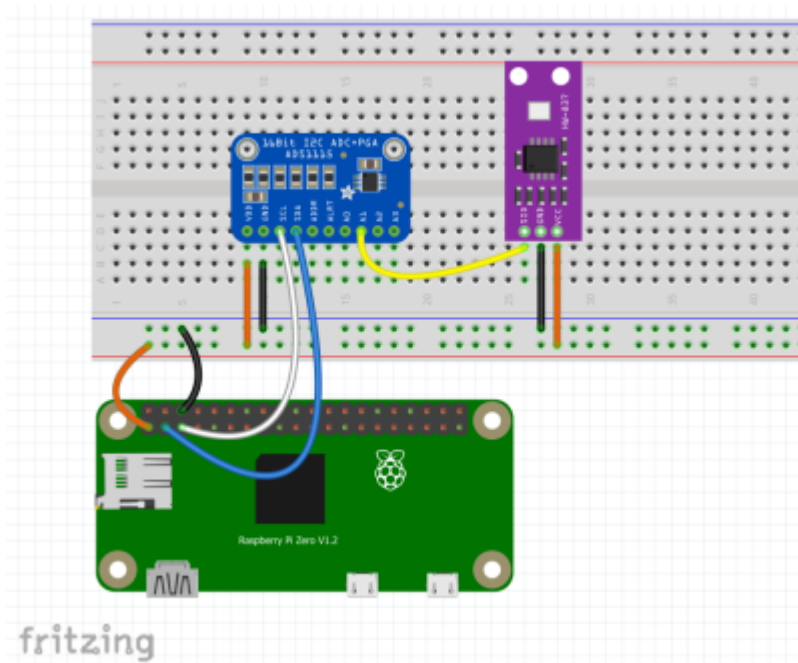
Wir verwenden das gleiche Gehäuse [wie der Helligkeitssensor](#). Es ist kein separates Gehäuse notwendig.

## Zusammenbau und Montage

Das Modul wird auf den entsprechenden Platz auf der Platine gesetzt, siehe [Anleitung zum Helligkeitssensor](#).

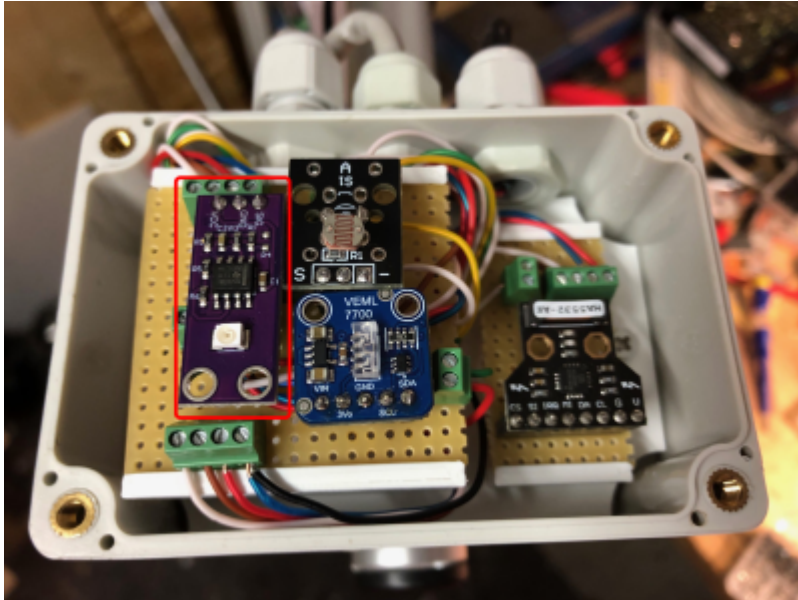
## Anschluss

Der GUA S12S wird über den ADS1115 Spannungswandler am Raspberry angeschlossen, hier zur Übersicht erst einmal auf dem Breadboard - aber Achtung! Es existieren sehr viele verschiedene Bauarten und Platinen! Schaut Euch die Beschriftung eurer Platine genau an!



Auf der Platine in der Wetterstation wird der Sensor auf die gleiche Platine wie der Helligkeitssensor gesetzt. Der Anschluss ist dazu bereits komplett vorbereitet. Es ist ansonsten nichts weiter notwendig. Siehe Kapitel [9. Helligkeit](#)

Fertig angeschlossen:



## Software Allgemein

Der Sensor gibt je nach UV-Intensität eine analoge Spannung aus. Die ausgegebene Spannung entspricht dabei dem UV-Index: **UV-Index**  $\approx$  **Ausgangsspannung** / **0.1**. Die gemessene Spannung rechnen wir später in den UV-Index um.

## Software ESP32 (ESPEasy)

Der Sensor wird über das Device Analog input - internal eingebunden.

### Task Settings:

- Name: intanalog\_p1
- Enabled: Ja
- Analog Pin: ADC1 ch3 / GPIO-39
- Attenuation: 2.5 dB
- Oversampling: Use Current Sample
- Apply Factory Calibration: Nein
- Calibration enabled: Ja
- Point 1:  $0 \triangleq 0.000$
- Point 2:  $1357 \triangleq 1357.000$
- Send to Controller 1 (MQTT): Ja

- Intervall: 5



Main Config Controllers Hardware **Devices** Tools

---

**Task Settings**

Device: Analog input - internal ? ⓘ

Name:

Enabled:

---

**Data Source**

Remote Unit:  [Unknown Unit Name]

*Note: 0 = disable remote feed, 255 = broadcast*

---

**Device Settings**

Analog Pin:  ←

*Note: Do not use ADC2 pins with WiFi active*

Attenuation:

Oversampling:

---

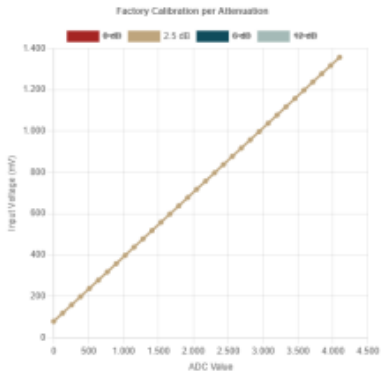
**Factory Calibration**

Apply Factory Calibration:

*Note: When checked, reading is in mV*

Factory Calibration Type: Calibration Line Fitting

Calibration Curve:



Current ADC raw value: 0

Attenuation @0 dB: Range / Step: 75 ... 1039 [mV] / 0.235 [mV]

Attenuation @2.5 dB: Range / Step: 78 ... 1357 [mV] / 0.312 [mV]

Attenuation @6 dB: Range / Step: 107 ... 1877 [mV] / 0.432 [mV]

Attenuation @12 dB: Range / Step: 142 ... 3134 [mV] / 0.731 [mV]

---

**Two Point Calibration**

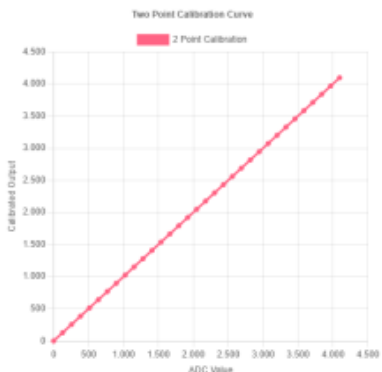
Calibration Enabled:

Point 1:  ±

Point 2:  ±

*Note: Input float values will be stored as int, calibration values will be adjusted accordingly*

Two Point Calibration:



Current: 0 [raw] ± 0.0000

Minimum: 0 [raw] ± 0.0000

Maximum: 4095 [raw] ± 4095.0000

Step Size: 0.9998

---

**Multipoint Processing**

Multipoint Processing Enabled:

Nr Multipoint Fields:

Split-Paste Multipoint Fields:

*Note: When checked, a set of tab, space or newline separated values can be pasted at once.*

---

**Data Acquisition**

Single event with all values:

*Note: Unchecked: Send event per value. Checked: Send single event (intanalog\_p1@AI) containing all values*

---

Send to Controller (Home Assistant (openHAB) MQTT, enabled)

---

Interval:  [sec]

### Values #1:

- Name: uv
- Formula:  $\%value\% * 1.357 / 4095$
- Decimals: 4

#	Name	Formula	Decimals	Stats	Hide	Axis
1	uv	$\%value\% * 1.357 / 4095$	4	<input type="checkbox"/>	<input type="checkbox"/>	L1

## Software Raspberry (Multi-IO Plugin)

Der Sensor wird über ein Sensormodul ads1x15 eingebunden, unter dem dann 1 Sensor Input für den UV-Index angelegt wird. Das gleiche Modul nutzen wir auch noch für 2 weitere Sensor Inputs: [Dämmerungssensor](#) und [Windrichtungsmesser](#). Es muss nur einmal für alle 3 Sensoren angelegt werden!

### Sensor Modul:

- Name: ads1115\_1
- Chip Address: 0x48
- Type: ADS1115
- Pins: Pin 0, Pin 1, Pin 2
- Gain: 1

The screenshot shows a configuration window titled "Add/Edit Sensor Module: ads1x15". It contains the following fields and options:

- Name: ads1115\_1
- Chip Address: 0x48
- Type: ADS1115
- Pins: Pin 0, Pin 1, Pin 2 (all checked), Pin 3 (unchecked)
- Gain: 1

At the bottom, there are three buttons: Save, Cancel, and Scan Bus.

### Sensor Input: UV-Index

- Name: uv
- Type: Voltage
- Pin: Pin 1
- Polling Intervall: 5



Add/Edit Sensor Input: ads1x15

Module Name	ads1115_1
Name	UV
Type	Voltage
Pin	Pin 1
Polling Interval	5

1)

Deutscher Wetterdienst:

<https://www.dwd.de/DE/service/lexikon/Functions/glossar.html?nn=103346&lv2=102828&lv3=788144>

2)

Deutscher Wetterdienst:

<https://www.dwd.de/DE/service/lexikon/Functions/glossar.html?nn=103346&lv2=102828&lv3=102866>

From:

<https://wiki.loxberry.de/> - **LoxBerry Wiki - BEYOND THE LIMITS**

Permanent link:

[https://wiki.loxberry.de/howtos\\_knowledge\\_base/loxberry\\_wetterstation/10\\_uv\\_sensor](https://wiki.loxberry.de/howtos_knowledge_base/loxberry_wetterstation/10_uv_sensor)

Last update: **2025/01/02 21:27**