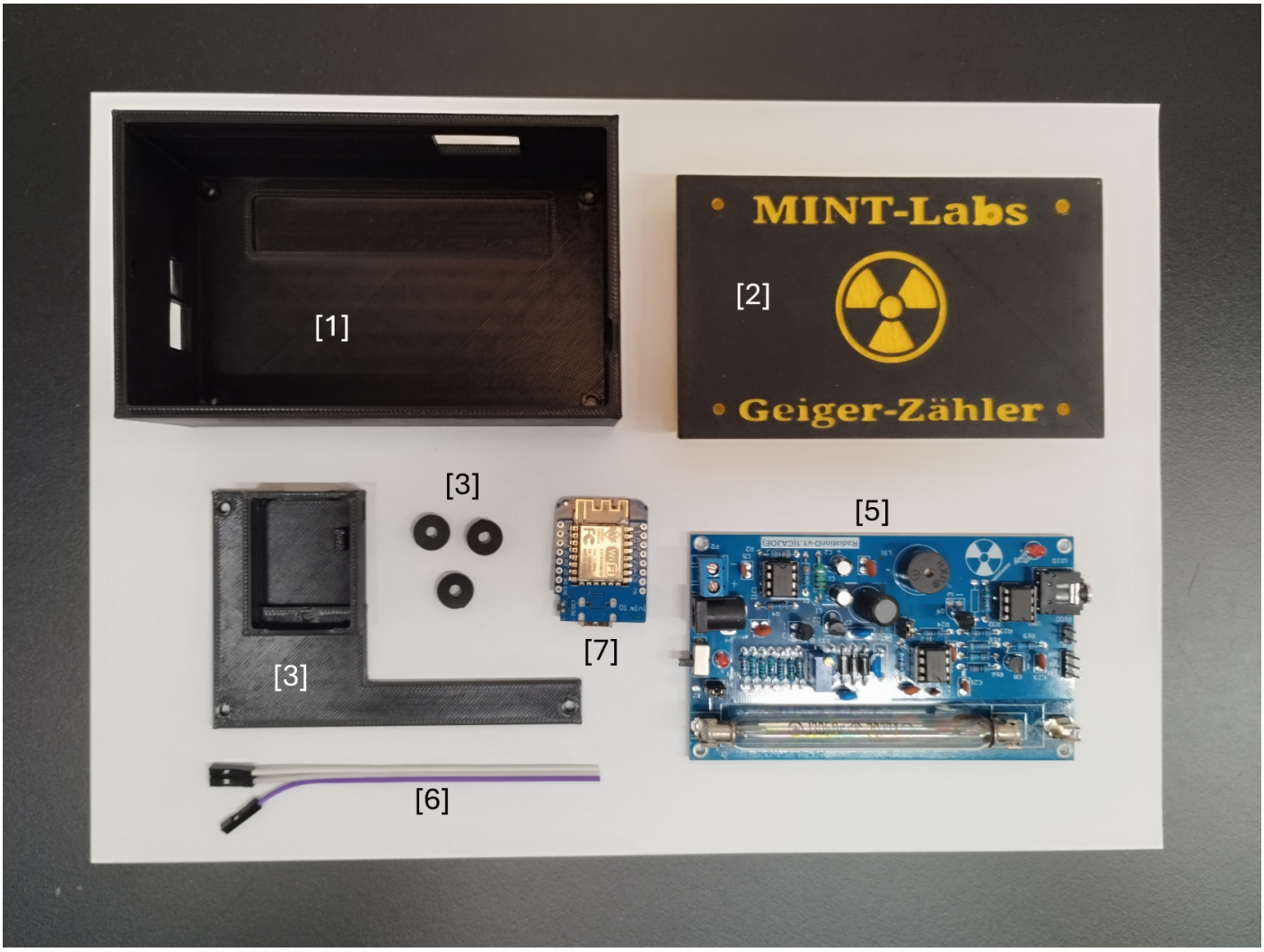
Geiger-Zähler

In den MINT-Labs sind drei Arduino-basierte Geiger-Zähler vorhanden. Infos zu diesen finden sich im MakeMagazin HIER. Ein Geiger-Zähler ist exakt wie in der Anleitung vom MakeMagazin beschrieben gebaut, inklusive Batterie-Case und LCD. Die zwei weiteren sind ohne LCD und ohne Batterie, sondern für den statischen Gebrauch im Haus gedacht. Einer davon hängt im Erdgeschoss direkt vor der "Brodelküche", einer hängt im UG vor dem Eingang zur "um:welt". Diese beiden sind auch im SmartHome eingebunden (Home Assistant) und sollen perspektivisch über MQTT ihre Werte auch online zur Verfügung stellen.

Aufbauanleitung:



1. Teile aus dem 3D-Drucker

Stück	Name	Bildnummer
1	Geigerzaehler_body_kurz.stl	[1]
1	Geiger-Zähler_Deckel_MINT-Labs.stl	[2]
1	Geigerzaehler_D1frame.stl	[3]
3	Geigerzaehler_DISTANZ.stl	[4]

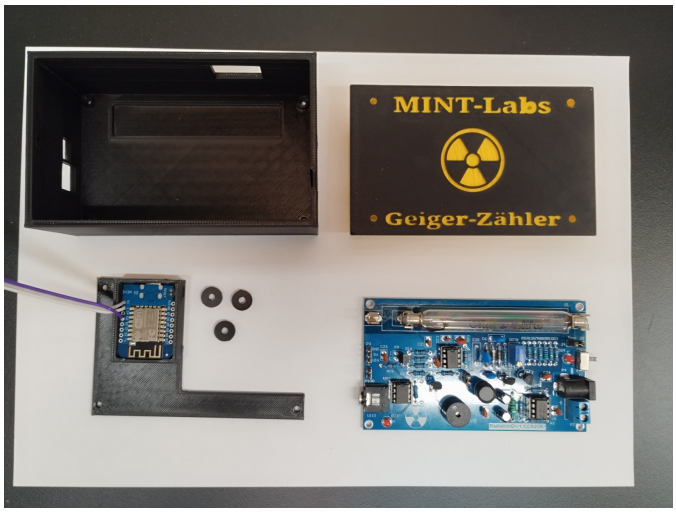
2. Elektronik-Teile

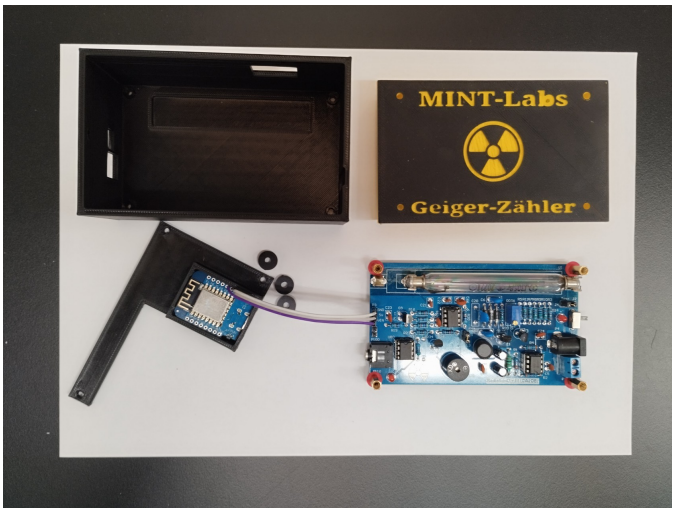
Stück	Name	Bildnummer
1	Geigerzähler mit Zählrohr	[5]
1	Kabel, dreipolig, einseitig mit Dupont-Stecker	[6]
1	ESP8266 D1 Mini	[7]

3. Sonstige Teile

Stück	Name	Bildnummer
1	USB-Kabel *(liegt Geigerzähler bei)*	[8]
4	Distanzbolzen *(liegt Geigerzähler bei)*	[9]
4	Mutter, M3 *(liegt Geigerzähler bei)*	[10]
9	Schraube, M3 x 8mm	[11]

4. Aufbau



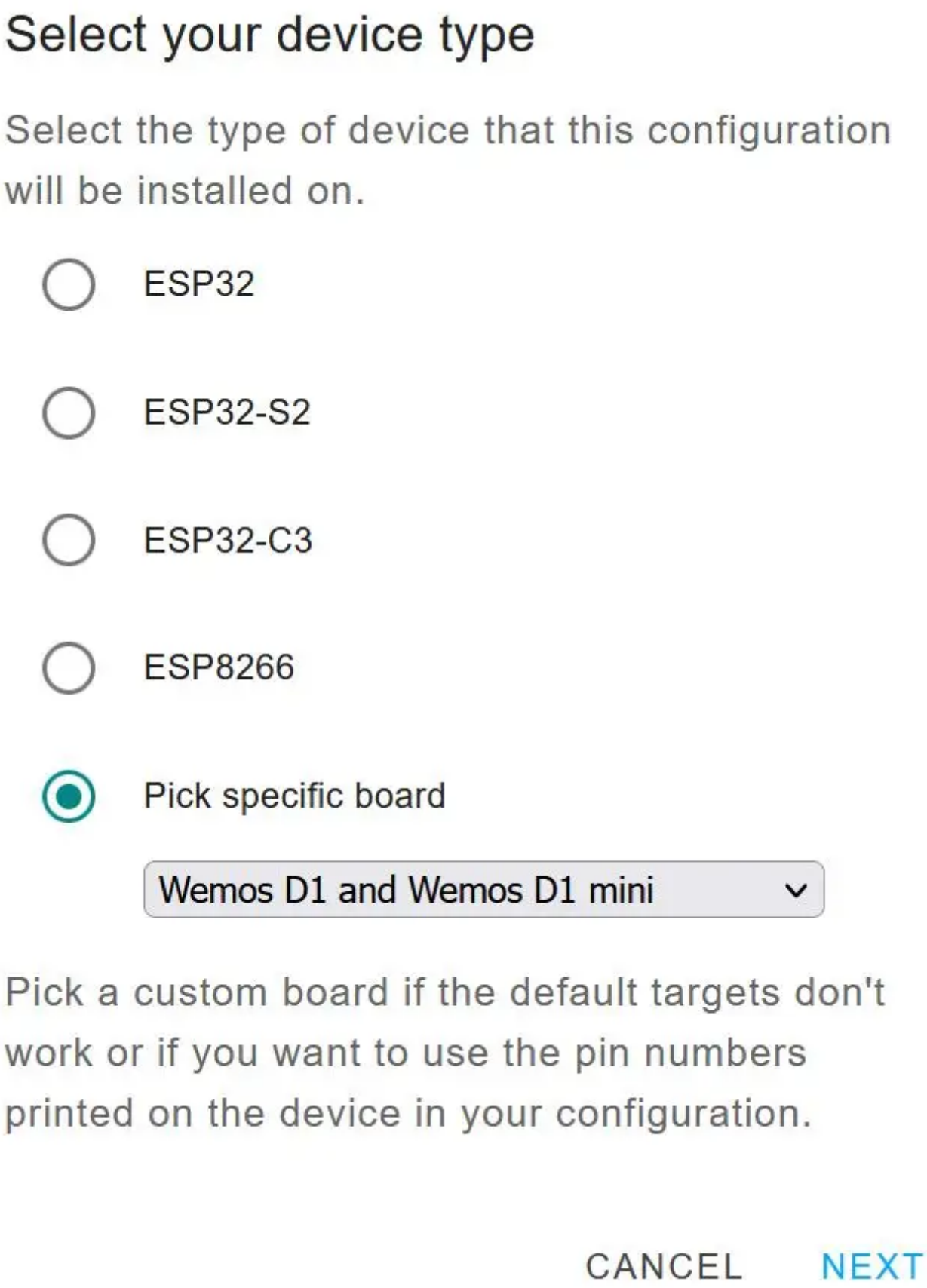


1. Bild: Dreipoliges Kabel an D1-Board anlöten (GND, 5V und D3)
2. Bild: Dreipoliges Kabel mit Dupont-Stecker an Geiger-Board stecken, Polung beachten: GND->GND, 5V->5V, D3->VIN; D1-Board vorsichtig in vorgesehene Box vom "Geigerzaehler_D1frame.stl" klemmen; die vier Distanzbolzen mit den vier M3-Muttern am Geiger-Board befestigen
3. Bild: "Geigerzaehler_D1frame.stl" mit den drei "Geigerzaehler_DISTANZ.stl" und den drei M3-8mm Schrauben am "Geiger-Zähler_Deckel_MINT-Labs.stl" befestigen
4. Bild: Geiger-Board in Box einlegen, Bauteile-Seite geht nach unten, die Bolzen stehen dann direkt auf den Löchern
5. Bild: Mit vier M3-8mm-Schrauben das Geiger-Board in der Box befestigen
6. Bild: die zwei verbleibenden Schrauben M3-8mm seitlich in die Box schrauben, um den Deckel mit der Box zu verbinden

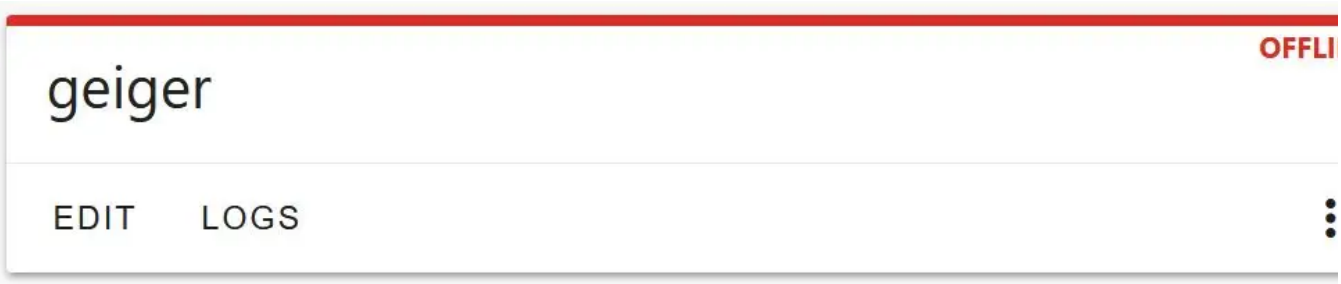
5. Programmierung

Siehe auch HIER. Einzig der Teil mit dem LCD wird anders gemacht als dort beschrieben:

1. Wir schließen den ESP an einen Rechner an und installieren über die ESPhome-Oberfläche des Home Assistant das ESPhome und geben dem Board einen Namen.
2. Im weiteren Verlauf wählen wir "Specific Board" und dann Wemos D1 mini



3. Somit haben wir einen neuen ESP als Gerät angelegt. Wir installieren es aber noch nicht sondern wählen zunächst "Skip" und dann "Edit" um den Code manuell zu bearbeiten



4. Als Code geben wir ein:
5.
```
sensor:
 - platform: pulse_counter
   pin: D3
   unit_of_measurement: 'mkSv/Hour'
   name: 'Ionizing Radiation Power'
   count_mode:
     rising_edge: DISABLE
     falling_edge: INCREMENT
   update_interval: 60s
   accuracy_decimals: 3
   id: my_doze_meter
   filters:
     - sliding_window_moving_average: # 5-minutes moving average (MA5) here
         window_size: 5
         send_every: 5
     - multiply: 0.0067 # SBM20 tube conversion factor of pulses into mkSv/Hour

 - platform: integration
   name: "Total Ionizing Radiation Doze"
   unit_of_measurement: "mkSv"
   sensor: my_doze_meter # link entity id to the pulse_counter values above
   icon: "mdi:radioactive"
   accuracy_decimals: 5
   time_unit: min # integrate values every next minute
   filters:
     - multiply: 0.00009
```

6. Nun noch "install" klicken und schon wird die Firmware kompiliert und auf den ESP gespielt. In Home Assistant steht nun eine Entität "*sensor.total_ionizing_radiation_doze*" zur Verfügung.

Revision #3
Created 3 December 2024 16:12:20 by Fabian
Updated 3 December 2024 17:07:19 by Fabian