

Kurzanleitung DigiSelfTrans

Pegel-Selbstbau, Hard- und Software, V1.2 (25.08.2026)

1. Hardwaredesign

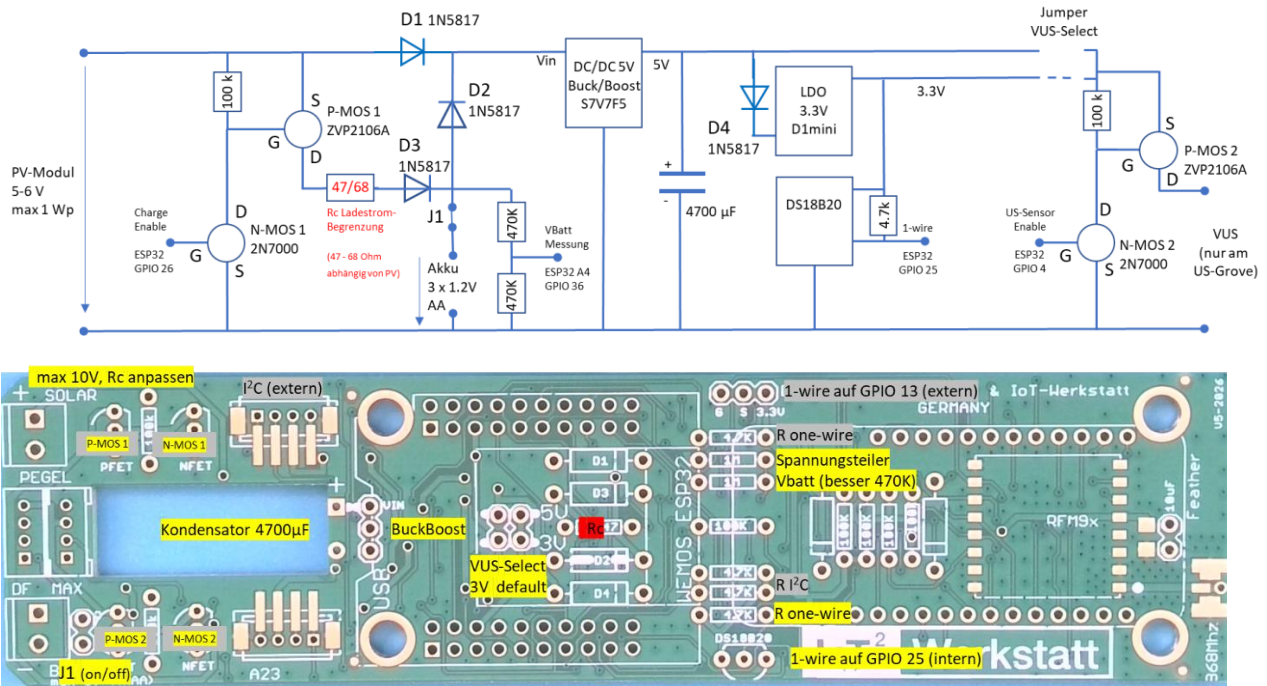
Akku/PV
Batteriebetrieb
Teilelisten

2. Software

Betrieb
Hardwaretest

3. Gehäuse

1 Hardwaredesign



Randbedingungen:

- LoRaWAN und ESP32 (D1mini) und IoT²-Werkstatt als einfachen Einstieg
- Lithium-Ionen-Akku zu gefährlich für die Arbeit mit Jugendlichen
- Nutzung von NiMh-Akkus, mit PV oder alternative Bestückung mit Batterie
- Erweiterungsmöglichkeiten (externes 1-wire, externes I²C)
- US-Sensor automatisch abschaltbar (Energieeffizienz)

Allgemeine Designentscheidungen

Die Versorgung des Ultraschallsensors wird über eine MOSFET Kombination und GPIO 4 vom ESP32 gesteuert. Damit ist der Sensor die meiste Zeit abgeschaltet und wir erreichen wenige µA Ruhestrom (mit den ESP32 Modulen von AZ-Delivery). ESP-Module aus anderen Quellen können abweichende Verbräuche aufweisen (nachmessen).

Über einen Jumper (VUS-Select) könnte der Maxbotix Sensor zukünftig auch mit 5 V betrieben werden und damit eine erhöhte Reichweite erzielen. Da der ESP32 ein 3.3V I/O-System besitzt, sind 5 V am seriellen Eingang allerdings außerhalb der Spezifikation. Hier wäre dann ein Spannungsteiler an der seriellen Leitung sinnvoll.

Der Spannungsteiler an GPIO36 ermöglicht die Messung der Batteriespannung im Betrieb. Hier teilen zwei Widerstände die Batteriespannung von max. 4.5 V auf den 3 V Bereich des ESP32 herunter. Die Dimensionierung ist sehr hochohmig gewählt (1 M bis 470 K), um den kontinuierlichen Strombedarf im DeepSleep zu minimieren. In der Praxis bewährt haben sich 470 kOhm (entgegen der Platinenbeschriftung von 1 MOhm, 1 MOhm hat in der Praxis bei hohen Außentemperaturen zu Messartefakten geführt).

Über den Jumper J1 kann die Batterie im Bedarfsfalle (Wartung) von der Platine getrennt werden.

Auf dem Board sind zwei weitere Grove-Schnittstellen vorgesehen. Die obere Schnittstelle ermöglicht den Anschluss von I²C-Sensoren. Bei Bedarf sind entsprechende Abschlusswiderstände (4.7k) bestückbar. Beim aktuell verwendeten ESP32-Package V3.2 hat der I²C Anschluss Timing Probleme mit 10 MHz CPU-Frequenz. Bei Nutzung müsste die CPU-Taktfrequenz erhöht werden.

Die untere Schnittstelle ermöglicht den Anschluss analoger Sensoren (GPIO 34 und GPIO 39).

Oben gegenüber des internen DS18B20 Temperatursensors befindet sich eine weitere 1-Wire Schnittstelle (GPIO 13). Hier kann z.B. ein externer Temperaturfühler angeschlossen werden. Dazu ist auch der entsprechende Widerstand 4.7 K zu bestücken.

Die D1mini ESP32 Module verfügen oft über eine Power-LED, die auch im Deepsleep noch mehrere mA Strom benötigt. Ein kurzer Kniff mit dem Fingernagel oder einem Schraubenzieher entfernt diesen für uns unnützen Energieverbraucher. Details [hier](#).

[LoRa-Shield Adafruit RFM95W](#). Die zum Betrieb notwendige Diodenlogik findet sich [hier](#).

Der ausführliche Schaltplan findet sich [hier](#).

Zusätzlich kann die Platine nach dem Löten mit Kunststoff-Lack (z.B. [Kontakt Chemie Plastik 70](#)) vor Feuchtigkeitseinwirkung geschützt werden. Steckverbinder, Klemmen, Antenne, ESP-Modul, USB-Buchse, Reset-Knopf vorher abkleben (oder nur die Leiterbahnseite besprühen).

Vor Inbetriebnahme: Sichtprüfung der Dioden/MOSFETs, Kurzschlussprüfung, Versorgung ohne ESP32, Hardwaretest mit USB (s. Testprogramm), Strommessung ohne USB, LoRa-Join-Test, Referenzmessung mit definiertem Abstand, Gehäuse-Dichtigkeits- und Kondensationstest.

Zum Test der Hardware nach dem Löten, gibt es [hier](#) ein eigenes Testprogramm (ohne LoRa und ohne Batteriespannungsüberwachung Tiefentladung). Siehe auch den entsprechenden Abschnitt in diesem Dokument.

Designentscheidungen PV/Akku Version

Handelsübliche NiMh-Schaltungen nicht für „Load Sharing“ / Power-Path-Betrieb geeignet. Wir benötigen gleichzeitiges solares Laden und Betreiben der Microcontroller Deshalb eigne Ladeschaltung mit Power-Path, d.h. ggf. Betrieb komplett über Solar.

PV-Modul ist in weiten Bereichen flexibel wählbar. Dabei sind Module mit maximal 10 V Leerlaufspannung möglich. Begrenzt wird dies durch Buck/Boost S7V7F5 von Pololu (Vin = 2.7 bis 11.8 V). Achtung: PV-Module haben im Leerlauf bei hoher Sonneneinstrahlung auch mehr als die angegebene Spannung, unbedingt ins Datenblatt schauen).

Unsere Wahl: [6 V, 1 Watt Peak](#), also max. 167 mA.

Maximal 1 Wp wurde gewählt, um das System auch bei Fehlbestückung (z. B. Diode D2 falsch herum) möglichst sicher zu gestalten. Denn dann würde die PV direkt auf die Akkus wirken, d.h. bei 1 W und 6 V würde der Ladestrom bei Sonne ca. 170 mA betragen. Spätestens nachts sollte das allerdings auffallen, denn das System würde dann nicht aus dem Akku gespeist. Das wäre kein sicherer Dauerzustand (bitte Bestückung vorher prüfen).

Buck/Boost, um PV auch bei geringer Solareinstrahlung optimal zu nutzen.

<https://www.pololu.com/product/2119>

Akkus sollen möglichst robust, sicher (bewusst keine Lilon) und einfach beschaffbar sein. Unsere Wahl fiel auf den für diese Anwendung besonders geeigneten Varta Solar Akku AA mit 800 mAh:

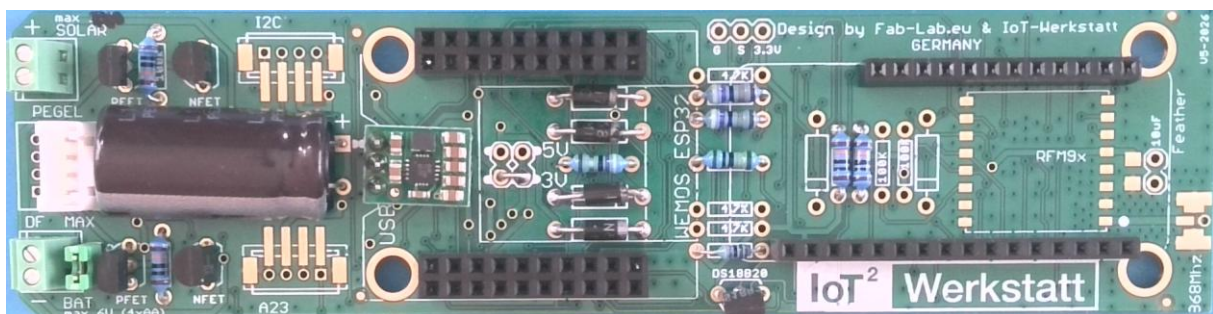
<https://www.varta-ag.com/de/konsument/produktkategorien/akkus/solar-aa-800-mah>

Hier kommt es nicht auf möglichst hohe Kapazität (mAh) an, sondern auf die Fähigkeit zur Dauerladung (Trickle Charge). Dies ist lt. [Datenblatt](#) mit 0.03-0.05 C = 24 - 40 mA möglich. Gehen wir von ca. 6.4 V Spannung (6.7 V Leerlauf minus 0.3 V über Diode D3) und vollen Batterien (3 x 1.5 V) aus, so fließt im Ladezweig über D3 ein durch R_c begrenzter Ladestrom. $I_c = (U_{PV} - U_{Batt}) / R_c = (6.4 - 4.5 \text{ V}) / 47 \text{ Ohm} = 40 \text{ mA}$ bzw. $(6.4 - 4.5 \text{ V}) / 68 \text{ Ohm} = 28 \text{ mA}$. Zusätzlich verhindert eine softwaregestützte Temperaturabschaltung eine Ladung bei sehr starker Sonne (Gehäusetemperatur über 30 Grad), Frost (Temperatur < 0 Grad Celsius) oder eine Batteriespannung von mehr als 4.4 V. Dazu dient die MOSFET 1 Stufe im Schaltplan, angesteuert über GPIO 26 des ESP32. Damit halten wir die im Datenblatt des Akkus angegebenen erlaubten Ladetemperaturen (0...45 Grad) ein.

Bei der Montage unbedingt auf richtige Polung achten. Bei Wartung vor Ort die PV abdecken und Akku über Jumper J1 abschalten. Alte Akkus gehören ins Recycling.

Bei anderen PV-Modulen und Akkus müssen diese Grenzwerte und der Ladewiderstand neu berechnet werden. In Abhängigkeit von Leerlaufspannung den Ladewiderstand R_c anpassen (vergrößern), Solar-Akkus nutzen (Trickle-Charge Angaben beachten). Auf keinen Fall andere Zelltypen als die vorgesehenen NiMh Akkus verwenden (keine Lilon, keine Batterien).

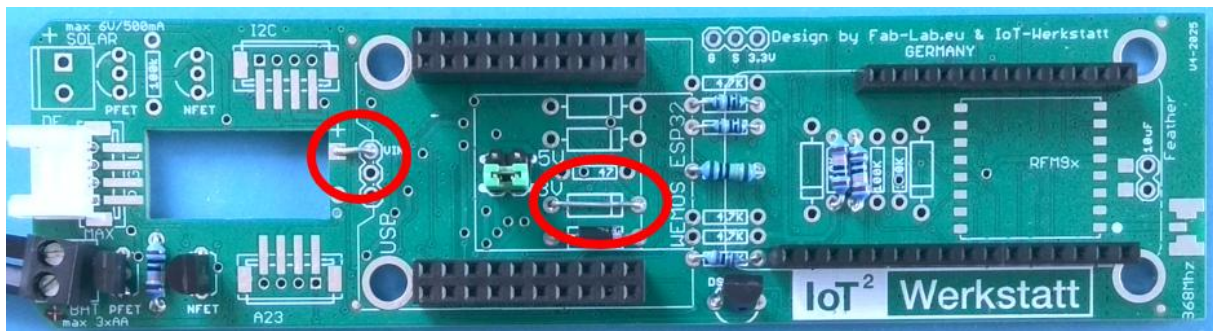
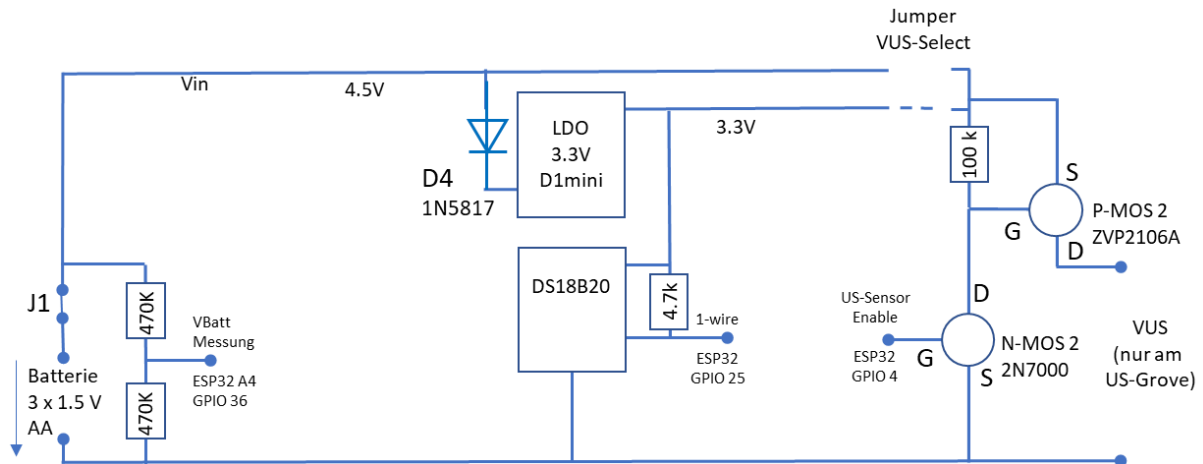
Die Diodenlogik D1 und D2 sorgt dafür, dass wenn die Spannung des Solarmoduls größer als die Batteriespannung ist, das System praktisch komplett aus der PV versorgt wird (Power-Path). Das sorgt für weniger Lade- / Entladezyklen der Akkus und ermöglicht ggf. einen Neustart auch bei niedrigem Akkustand.



Bestückung für Solarbetrieb

Designentscheidungen Batteriebetrieb

Vereinfachtes Setup, denn im Batteriebetrieb entfallen die Akku- und Solarkomponenten. Entsprechende Bauteile werden nicht bestückt bzw. durch zwei Drahtbrücken ersetzt. Diode D4 bleibt.



Bestückung Batteriebetrieb

Als Energieversorgung eignen sich AA Lithium Primärbatterien mit 1.5 V z. B. „[Ultra Lithium](#)“ [von Varta](#) aus dem Baumarkt. Theoretische Lebensdauer je nach Abtaste mehrere Jahre. Vorteil gegenüber normalen Alkali-Mangan Zellen ist der erhöhte Temperaturbereich im Winter (-20 bis 60 Grad Celsius, [Datenblatt](#)). Notfalls funktionieren auch gute Alkali-Mangan Zellen.

Teileliste (PV/Akku)

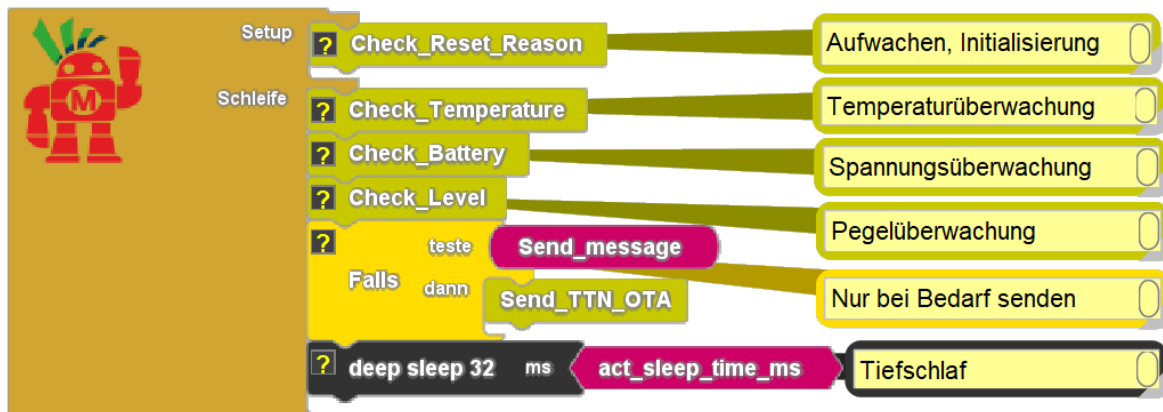
Anzahl	Bauteil	Nutzung / Quelle (Beispiel)
1	US - Sensor	Maxbotix 7368 oder 7369 (je nach Reichweite) DF Robot https://botland.de/ultraschall-abstandssensoren/15716-ultraschall-abstandssensor-a01nyub-28-750cm-wasserdicht-dfrobot-sen0313-6959420915750.html
1	ESP32 D1mini	https://www.az-delivery.de/products/esp32-d1-mini
1	Adafruit LoRa Radio FeatherWing – RFM95W 900 MHz	https://www.adafruit.com/product/3231
1	Buck/Boost Pololu S7V7F5	https://www.pololu.com/product/2119
1	Grove Buchse und ggf. Kabel (Maxbotix)	https://www.seeedstudio.com/Grove-Universal-4-pin-connector.html https://www.seeedstudio.com/Grove-4-pin-Female-Jumper-to-Grove-4-pin-Conversion-Cable-5-PCs-per-PAck.html
3	Varta Solar Akku AA, 800 mAh	https://www.varta-ag.com/de/konsument/produktkategorien/akkus/solar-aa-800-mah
2	MOSFET 2N 7000 DIO	Ladelogik und Abschaltung US-Sensor
2	MOSFET ZVP 2106 A	Ladelogik und Abschaltung US Sensor
5	Widerstand 100 kOhm	Pullup
1	Widerstand 47 oder 68 Ohm, 0.5 W	Ladestrom
2	Widerstände Metall 470 kOhm	Spannungsteiler Batteriespannung
1	Widerstand 4.7 kOhm	1-wire Pullup
6	Schottky 1N5817	Diodenlogik und Adafruit-LoRa Feather
1	DS18B20	1-wire Temperatur
1	Buchsenleiste 2,54mm 1x12	Feather-Wing
1	Buchsenleiste 2,54mm 1x16	Feather-Wing
2	2x10pol.-Stiftleiste	ESP32 D1-Mini
1	Elko, radial, 4700 µF, 10 V, 105°	https://www.reichelt.de/de/de/shop/produkt/elko_radial_4700_f_10_v_105_low_esr_rm_5-359290
2	Leiterplattenklemme, 2 polig, RM 2,54 mm	Solar/Batterieanschluss
1	Batteriehalter 3 AA	
1	Solarzelle 5-6V, <1Wp	
1	Kabel PV (optional)	https://www.pollin.de/p/wasserdichtes-steckverbindungsset-stecker-kupplung-1-m-2-polig-452375

Teileliste (Batteriebetrieb)

Anzahl	Bauteil	Nutzung / Quelle (Beispiel)
1	US - Sensor	Maxbotix 7368 oder 7369 (je nach Reichweite) DF Robot https://botland.de/ultraschall-abstandssensoren/15716-ultraschall-abstandssensor-a01nyub-28-750cm-wasserdicht-dfrobot-sen0313-6959420915750.html
1	ESP32 D1mini	https://www.az-delivery.de/products/esp32-d1-mini
1	Adafruit LoRa Radio FeatherWing - RFM95W 900 MHz	https://www.adafruit.com/product/3231
1	Grove Buchse und ggf. Kabel (Maxbotix)	https://www.seeedstudio.com/Grove-Universal-4-pin-connector.html https://www.seeedstudio.com/Grove-4-pin-Female-Jumper-to-Grove-4-pin-Conversion-Cable-5-PCs-per-Pack.html
3	Varta Lithium Ultra AA	https://www.varta-ag.com/de/konsument/produktkategorien/batterien/lithium/ultra-lithium-aa
1	MOSFET 2N 7000 DIO	Abschaltung US-Sensor
1	MOSFET ZVP 2106 A	Abschaltung US-Sensor
4	Widerstand 100 kOhm	Pullup
2	Widerstände Metall 470 kOhm	Spannungsteiler Batteriespannung
1	Widerstand 4.7 kOhm	1-wire Pullup
1	DS18B20	1-wire Temperatur
3	Schottky 1N5817	Diodenlogik und Adafruit Lora-Feather
1	Buchsenleiste 2,54mm 1x12	Feather-Wing
1	Buchsenleiste 2,54mm 1x16	Feather-Wing
2	2x10pol.-Stiftleiste	ESP32 D1-Mini
1	Batteriehalter 3 AA	

2 Software V6

Das Ardublock-Programm V6 findet sich [hier](#).



Randbedingungen:

- Nutzung der IoT²Werkstatt mit ESP32, Programmierung nicht nur für Nerds.
- Konsequente Verwendung der Energiesparmöglichkeiten des ESP32 und unserer Platine.

Hauptprogramm Loop

Das Programm läuft in einer Endlosschleife, führt Messungen durch (Check_xxx) und entscheidet in Abhängigkeit von den Messergebnissen, ob Daten über LoRa (Send_TTN_OTA) an die Cloud gesendet werden müssen. Anschließend geht die CPU für eine bestimmte Zeit (act_sleep_time_ms) in den energieeffizienten Tiefschlaf (Deepsleep). Eine interne Uhr sorgt für regelmäßiges Aufwachen und Fortsetzung des Messablaufs.

Unterprogramm Check_Reset_Reason

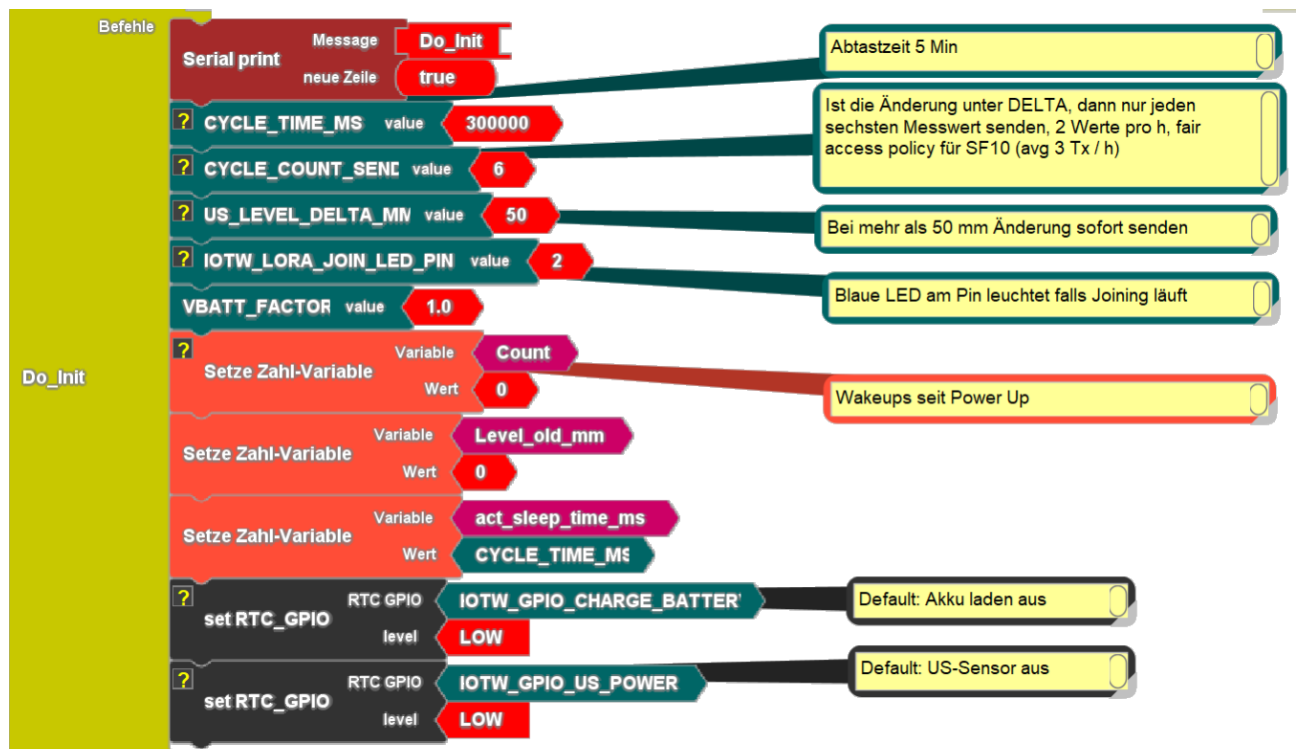
Das erste Unterprogramm (Check_Reset_Reason) überprüft den Aufwachgrund. Im Normalfall wird dies der Deepsleep-Reset sein und das Programm in der Wiederholungsschleife eine weitere Messung durchführen. Die Anzahl der Messungen wird gezählt (count) und kann später in der Cloud abgerufen werden. Ein Reset von Count deutet auf Hardwareprobleme hin (Feuchtigkeit, Batterie, Akku).



Nur beim allerersten Aufruf nach einem Stromausfall oder einem Reset über den Reset-Taster initialisiert sich das Programm (Do_Init).

Unterprogramm Do_Init

Hier können verschiedene Anpassungen an die Randbedingungen vor Ort vorgenommen werden.



Der wichtigste Parameter ist **CYCLE_TIME_MS**. Dieser legt die Zykluszeit der normalen Messung fest. In unserem Fall wird alle 5 Minuten (= 300.000 ms) ein Messvorgang ausgelöst. Nur wenn bei der Messung ein **US_LEVEL_DELTA** (hier von mehr als 50 mm) zur letzten Datenübertragung in die Cloud festgestellt wird, erfolgt eine sofortige Benachrichtigung über LoRa. Andernfalls erfolgt die Übertragung nur alle **CYCLE_COUNT_SEND** Messungen. In unserem Fall wird also mindestens alle $5 * 6 = 30$ Minuten gesendet, es sei denn, der Wasserstand hat sich um mehr als 50 mm verändert. Dies ist ein Kompromiss aus Geschwindigkeit (alle 5 Minuten melden) und Energieverbrauch (bei Stillstand nur alle 30 Minuten senden).

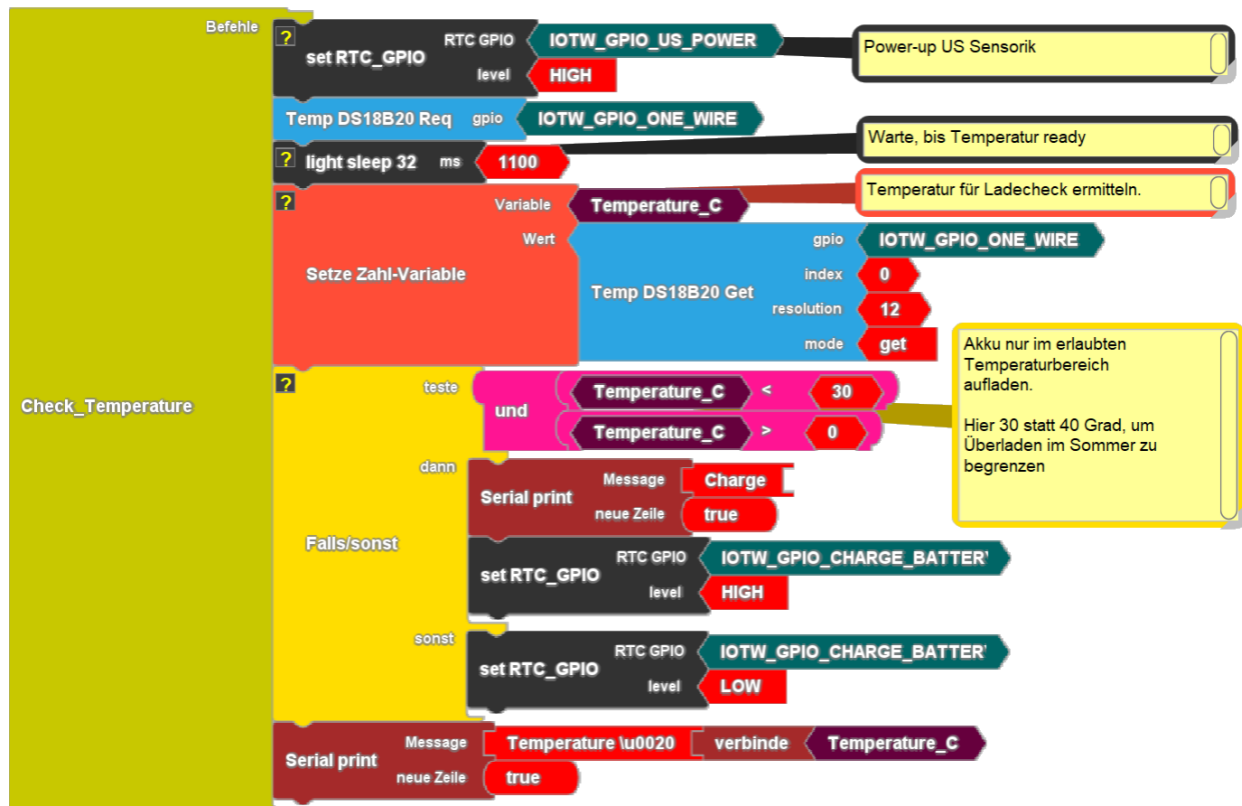
Über **IOTW_LORA_JOIN_LED_PIN** kann eine ggf. vorhandene LED auf dem D1miniESP32-Board getriggert werden (bei den verwendeten AZ-Delivery Boards ist das die blaue LED an GPIO 2). Diese leuchtet dann bei der OTA-Initialisierung (benötigt aber auch Energie). Dieses Feature erleichtert die Inbetriebnahme am Standort (LoRa-Gateway gesehen?).

Der **VBATT_FACTOR** könnte für besondere Anwendungen eine noch genauere Kalibrierung der Batteriespannung ermöglichen. Hier nutzen wir zwei hochohmige Widerstände als Spannungsteiler, wodurch sich eine gewisse Messungenauigkeit (ca. 5%) ergibt.

Die beiden **setRTC_GPIO** Blöcke sorgen für einen definierten Zustand der beiden GPIOs für die MOSFET Schaltung (Laden Akku, US Sensor aktivieren). RTC-GPIOs werden auch im Deepsleep auf den gewünschten Pegel gehalten.

Unterprogramm Check_Temperature

Hier erfolgt die Temperaturmessung und eine Entscheidung, ob der Akku geladen werden darf.



Der verwendete 1-wire Temperatursensor benötigt je nach Temperaturlösung eine Wandlungszeit von ca. 1.1 Sekunden. In dieser Zeit hat die CPU eigentlich nichts zu tun und kann in den energieeffizienten Lightsleep-Zustand wechseln. Vorher setzen wir noch den GPIO für die Ansteuerung des Ultraschallsensors, so dass dieser in der Zeit seine Initialisierung durchlaufen kann.

Nach Abfrage des Temperatursensors erfolgt die Entscheidung, ob sich die Gehäusetemperatur in einem für die Akkuladung erlaubten Bereich befindet. Entsprechend wird ggf. das MOSFET für den Trickle-Charge Zweig freigeschaltet. Dieser Zustand wird dann auch im DeepSleep bis zur nächsten Temperaturmessung beibehalten. Die 30 Grad sind defensiv gewählt und treten üblicherweise nur im Hochsommer bei direkter Sonnenbestrahlung auf. Sie schützen den Akku vor zu langer Überladung.

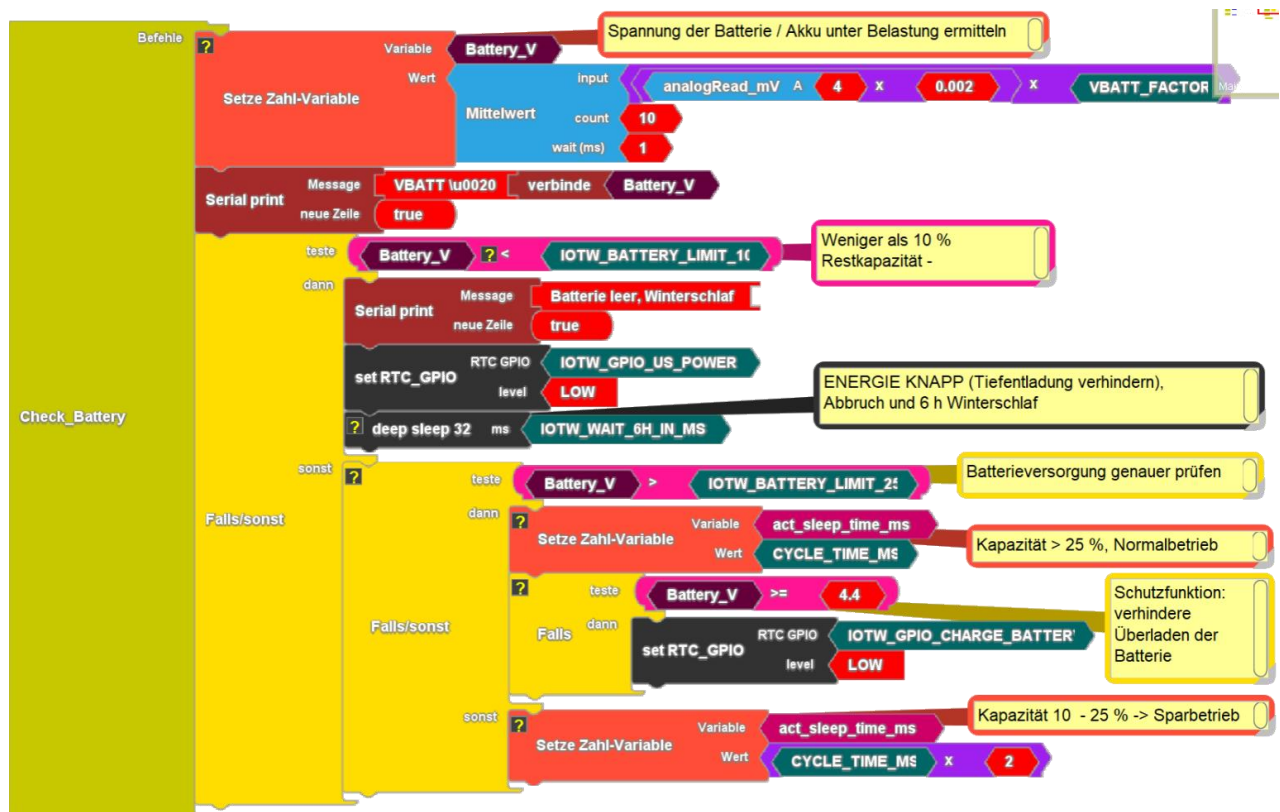
Ein eventuell vorhandener externer 1-wire Sensor könnte über



Initiiert und nach der Wartezeit analog ausgelesen werden.

Unterprogramm Check_Battery

Hier wird die Akkuspannung gemessen und entsprechende Aktionen eingeleitet.



Ein ganz wichtiger Punkt ist die Lebensdauer des Gesamtsystems. Neben der Temperaturkontrolle spielt hier der Ladezustand des Akkus eine zentrale Rolle. Wir messen die Akkuspannung unter Belastung über einen Spannungsteiler und glätten diesen Messwert durch Mittelwertbildung (10 Messungen in 1 ms Abstand). Über den Faktor rechnen wir die gemessenen mV und die Teilung durch den Spannungsteiler auf die tatsächliche Spannung (in V) um.

Anschließend orientieren wir uns an der Spannungskurve eines NiMh Akkus. In `arduino_pins.h` der Makey on Battery finden sich dazu folgende Predefinitions:

```
#define IOTW_BATTERY_LIMIT_25 3.6 // 3xAA NiMH, Spannung bei 25 % Restkapazität
```

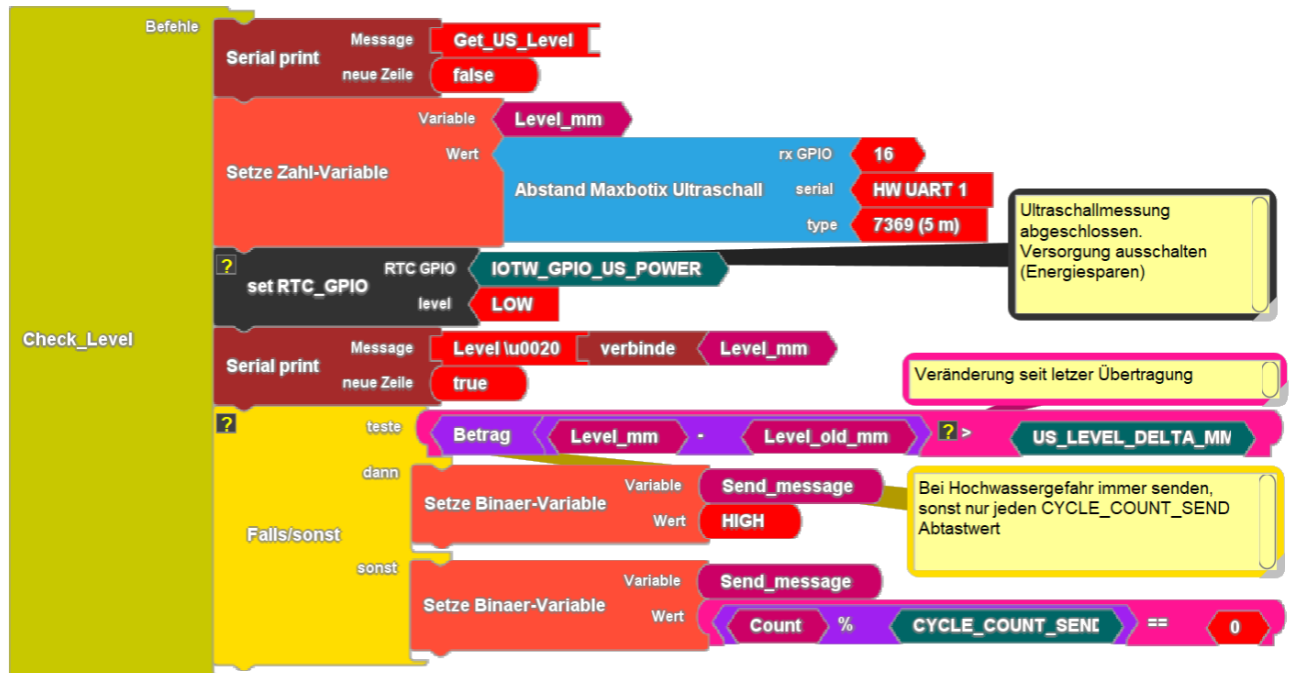
```
#define IOTW_BATTERY_LIMIT_10 3.3 // 3xAA NiMH, Spannung bei 10 % Restkapazität
```

Je nach gemessener Spannung verdoppeln wir die Abtastdauer bzw. gehen sofort für 6 Stunden in den Tiefschlaf. Im Tiefschlaf verbraucht unsere Schaltung dann nur wenige μA und überdauert so eine längere Dunkelfase, ohne den Akku tief zu entladen.

Eine zusätzliche Überwachung stoppt die Trickle-Charge Ladefunktion des Akkus, falls 4.4 V überschritten wird. Dies ist keine echte „Akku voll“ Erkennung, sondern eher eine zusätzliche Schutzfunktion für die Akkus. Uns geht es im System nicht darum, möglichst schnell 100 % Ladezustand zu erzeugen, sondern die Lebensdauer des Akkus zu maximieren

Unterprogramm Check_Level

Hier werden wir den Ultraschallsensor aus.



Der beim Start der Temperaturmessung aktivierte Ultraschallsensor wird ausgewertet. Hierzu erfolgt eine serielle Kommunikation auf GPIO 16. Je nach gewähltem Sensor gibt es verschiedene Messbereiche (5 m bzw. 10 m bei Maxbotix).

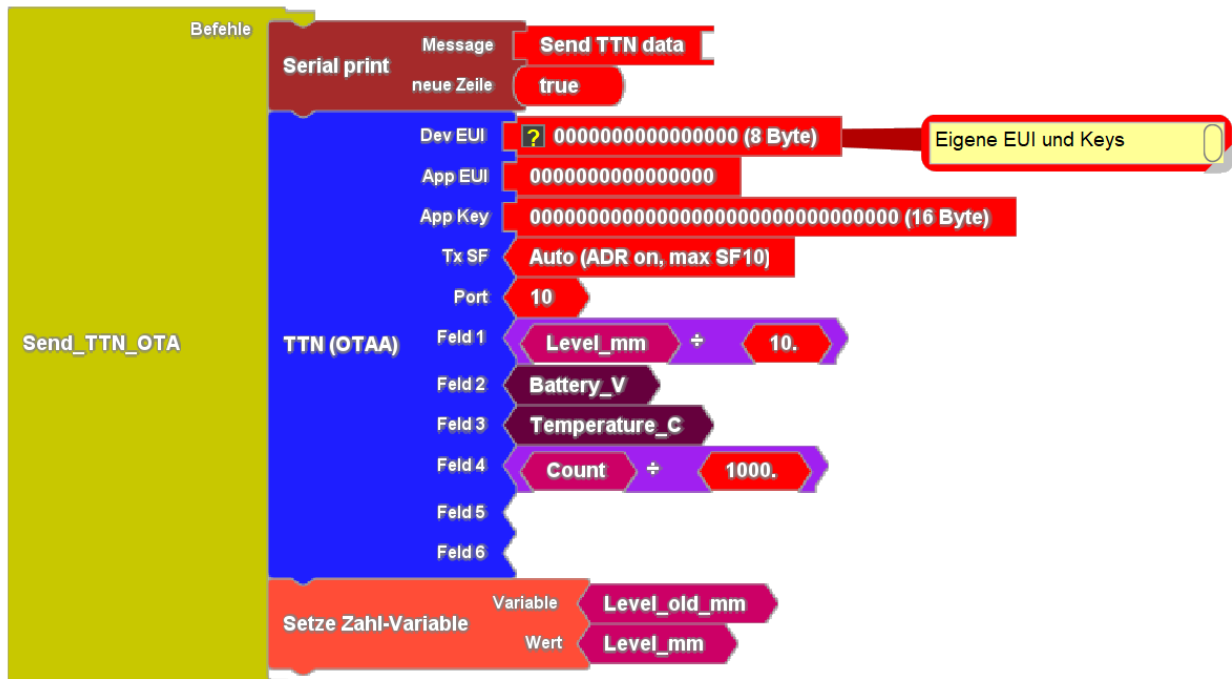
Wird der kostengünstigere DFRobot Sensor verwendet, ist das entsprechende Blöckchen (aus Externe Interfaces) zu nutzen.



Je nach Messwert und dem zuletzt übermittelten Pegelstand wird hier entschieden, ob es einer sofortigen Meldung per LoRaWAN erfordert. Nur alle **CYCLE_COUNT_SEN** Messungen wird zwingend gesendet.

Unterprogramm Send_TTN_OTA

Hier erfolgt die Übertragung der Messdaten in die Cloud



Dazu sind die eigenen EUIs (aus der TTN-Cloud) anzugeben. Der „SpreadingFactor SF“ beeinflusst Reichweite und Übertragungsdauer und wird hier automatisch gewählt.

<https://avbentem.github.io/airtime-calculator/ttn/eu868>

Die Konfiguration des TTN Netzwerkes wird in einem gesonderten [Tutorial](#) behandelt.

Testsoftware für die Hardware

Teste die korrekte Funktion der Hardware (ohne LoRa). Das Programm findet sich [hier](#).



Da das finale System aus Energiespargründen im Betrieb möglichst wenige Testausgaben generiert und die langen Zykluszeiten zusätzlich eine Kontrolle erschweren, gibt es ein vereinfachtes Testprogramm für die Hardware. Denn nichts ist ärgerlicher, als mit einer kalten Lötstelle an der Brücke zu stehen.

Dieses Programm kann auch ohne LoRa-Feather und PV genutzt werden. Ultraschallsensor anschließen (bei DF Robot ist das Programm zu modifizieren). Batterie / Akku anschließen. Dann sollte die Ausgabe etwa wie folgt aussehen:

```
-----  
Temperature  24.44  
Level  500  
Mit ChargeVBATT  4.42  
Ohne ChargeVBATT  4.43  
-----
```

Durch ändern des Abstands vom Ultraschallsensor können wir die Funktion testen. Mit einem Finger auf den 1-wire Temperatursensor checken wir auch dessen korrekte Funktion.

Wenn wir nun die Batterie statt an den Batterieklemmen, an die Klemmen für die PV anschließen, können wir auch den Ladezweig testen. Jetzt sehen wir die Batteriespannung nur, wenn der Ladezweig aktiv ist.

```
-----  
Temperature  23.87  
Level  500  
Mit ChargeVBATT  4.35  
Ohne ChargeVBATT  1.02  
-----
```

Wer ein Strommessgerät verfügt (z.B. [Power Profiler Kit II von Nordic](#)), der kann auch den Energiebedarf des Gesamtsystems beobachten. Dabei muss die USB-Verbindung zum PC unbedingt getrennt werden. Andernfalls stören die USB-Ströme eine sinnvolle Messung.

Unsere Deepsleep-Pause sollte einen Bedarf von ca. 100 μA anzeigen. Andernfalls ist vielleicht die Power-On LED auf dem Modul noch aktiv?



3 Gehäuse

Gehäuse möglichst schattig montieren, helles Gehäuse. Solarmodul mit Abstandshaltern hinterlüftet montieren, idealerweise abgesetzt Richtung Süden. Durch tägliche starke Temperaturwechsel entsteht ein Druckwechsel im Gehäuse, der Feuchte durch Dichtungen ziehen kann. Die Druckausgleichsschraube sollte seitlich oder unten sitzen, nicht auf einer waagerechten Oberseite, und eine Tropfschleife an allen von oben kommenden Kabeln ergänzt werden. Version mit Wasserrohr findet sich [hier](#). Unbedingt auf korrekte Position der Dichtungen achten.

Teileliste Wasserrohr

AF0601.068.050 PVC-U Verschraubung, 2fach Klebemuffe 50mm, 2 1/4"	https://www.pvc-welt.de/PVC-U-Verschraubung-2fach-Klebemuffe_1
AF0604.068.050 PVC-U Verschraubung Klebemuffe x Aussengewinde 50mm	https://www.pvc-welt.de/PVC-U-Verschraubung-Klebemuffe-x-Aussengewinde_2
AF0816.000.050 PVC-U Kappe Innengewinde incl. Flachdichtung 1 1/2"	https://www.pvc-welt.de/PVC-U-Kappe-Innengewinde-incl-Flachdichtung
AFA203.050.025 PVC-U Reduzierring Achtkant, Klebestutzen x Innengewinde, Typ A 50mm x 3/4"	https://www.pvc-welt.de/PVC-U-Reduzierring-Achtkant-Klebestutzen-x-Innengewinde-Typ-A
3D-Druck: Adapter für China-Sensor - INNODIG	3D-Modell verlinken
Rohr	https://www.pvc-welt.de/PVC-U-Rohr-Grau-50-x-24mm-PN-10
PVC-Kleber	https://www.pvc-welt.de/PVC-Weltde-PVC-U-Kleber-und-Reiniger
Druckausgleichsschraube	https://www.reichelt.de/de/de/shop/produkt/druckausgleichselement_m12x1_5-285020
Kabeldurchführung	https://www.reichelt.de/de/de/shop/produkt/kabelverschraubung_m12_grau-284013
Schelle SZ0100.047.051 47 - 51mm 20 M8 M10 47 - 51	https://www.pvc-welt.de/Schraubrohrschele-Stahl-verzinkt-mit-Gummieinlage-DIN-4109_2

Alternativ: Wasserdichtes Rechteckgehäuse (Bildquelle DRK Vulkaneifel). Dieser Ansatz hat sich in der Praxis bewährt (einfacherer Zugang bei Wartung vor Ort).

