

Power-Management des Heidi-Trackers

Der Arduino wird von 2 Akkus versorgt. Das ist einerseits die reguläre Bordspannungsversorgung und andererseits ein Stütz-Akku, der ab einer Bordspannung von $U_{\text{batt}} < \text{ca. } 3,8\text{V}$ dafür sorgt, dass die Spannungseinbrüche, verursacht durch die hohen Spitzenströme des GSM-Moduls, nicht zu einem Brown-Out des Arduino führen. Beide Versorgungsstränge sind über eine Stromweiche, 2 Schottky-Dioden, zusammengeführt. Diese Schaltung sorgt dafür, dass beide Akkus beim Entladen immer das gleiche Spannungsniveau haben.

Der Arduino bezieht also seine Versorgungsspannung über je 1 Schottky-Diode Typ 1N5817. Diese hat einen Spannungsverlust von etwa $0,3\text{V}$ in Durchlassrichtung, das heißt am Arduino kommt $U_{\text{batt}} - 0,3\text{V}$ an.

Der ESP32 arbeitet bis etwa $2,6\text{V}$ korrekt, soweit keine internen Verbraucher wie GPIO's zusätzlichen Strom benötigen. Sobald sich der Strombedarf erhöht, kann es zu einem Brownout-Reset kommen (ein Puffer-Kondensator ändert daran nichts).

Im Bereich unter $2,6\text{V}$ ist das Verhalten des ESP32 unbestimmt – er bootet im Kreis.

Der Tiefenentladeschutz der Lademodule schaltet die Versorgung erst bei $U_{\text{batt}} < 2,4\text{V}$ ab und erst bei $U_{\text{batt}} \geq 3,0\text{V}$ wieder zu.

Folgende Spannungsgrenzen sind fest eingestellt:

- $U_{\text{batt}} \geq 3,6\text{V}$ – normale Funktion
- $3,6\text{V} > U_{\text{batt}} \geq 3,5\text{V}$ – verdoppelte Zeiten zwischen den regulären Datenübertragungen, Alarmer normal
- $3,5\text{V} > U_{\text{batt}} \geq 3,4\text{V}$ – keine regulären Datenübertragungen, Alarmer normal
- $U_{\text{batt}} < 3,4\text{V}$ – deep sleep 15 Minuten
- $U_{\text{batt}} < 3,3\text{V}$ – deep sleep 60 Minuten

Ziel des Power-Management ist es, bei Unterversorgung der Akkus so lange wie möglich Alarmer zu senden und später eine Versorgungsspannung des Arduino von unter $2,6\text{V}$ zu vermeiden.

Ab $3,5\text{V}$ Batteriespannung hat der Akku nur noch sehr wenig Energiegehalt. Deshalb wird die reguläre Übertragung der Daten eingestellt. Alarmer werden ab $3,4\text{V}$ nicht mehr abgesetzt, da die Spannung dabei so stark einbricht, dass die Übertragung nicht sicher gewährleistet ist.

Um das relativ schnelle Entladen des Akkus bei $U_{\text{batt}} < 2,8\text{V}$ auf $U_{\text{batt}} < 2,4\text{V}$ zu vermeiden, kann eine Schmitt-Tigger-Schaltung mit entsprechender Hysterese für eine Abschaltung des Arduino ab $U_{\text{batt}} < 3,0\text{V}$ eingebaut werden.

Extremes Unterversorgungs-Szenario:

Pos	U _{batt}	Betriebszustand	Ladezustand
1	< 3,6 V	Strom sparen durch Verdopplung der Zeitspanne zwischen den Datenübertragungen	
2	< 3,5 V	Einstellung der Datenübertragungen (messungen laufen weiter), nur noch Alarme werden abgesetzt (mit Übertragung des Standorts)	
3	< 3,4 V	keine Aktivitäten mehr, alle 15 Minuten Überprüfung des Batteriezustandes	
4	< 3,3 V	keine Aktivitäten mehr, alle 60 Minuten Überprüfung des Batteriezustandes	
5	< 3,0 V	Der ESP32 löst bei der Überprüfung des Batteriezustandes einen Brown-Out-Reset aus, beim nächsten Bootvorgang wird der Brown-Out erkannt und ein Deep-Sleep von 60 Minuten ausgesöst	
6	< 2,8 V	Der ESP32 bootet im Kreis und verbraucht dabei dauerhaft 30 mA, der Akku wird weiter entladen	
7	< 2,4 V	Der Laderegler des Akkus schaltet alle Lasten ab	
8	> 2,4 V	Der Laderegler des Akkus schaltet alle Lasten weiterhin ab	
9	> 3,0 V	Der Laderegler des Akkus schaltet alle Lasten wieder zu → Position 5 – der ESP32 bootet zumindest kontrolliert	
10	> 3,3 V	→ Position 4	
..	..	usw.	

Discharge, capacity: Samsung INR18650-35E 3500mAh (Pink)

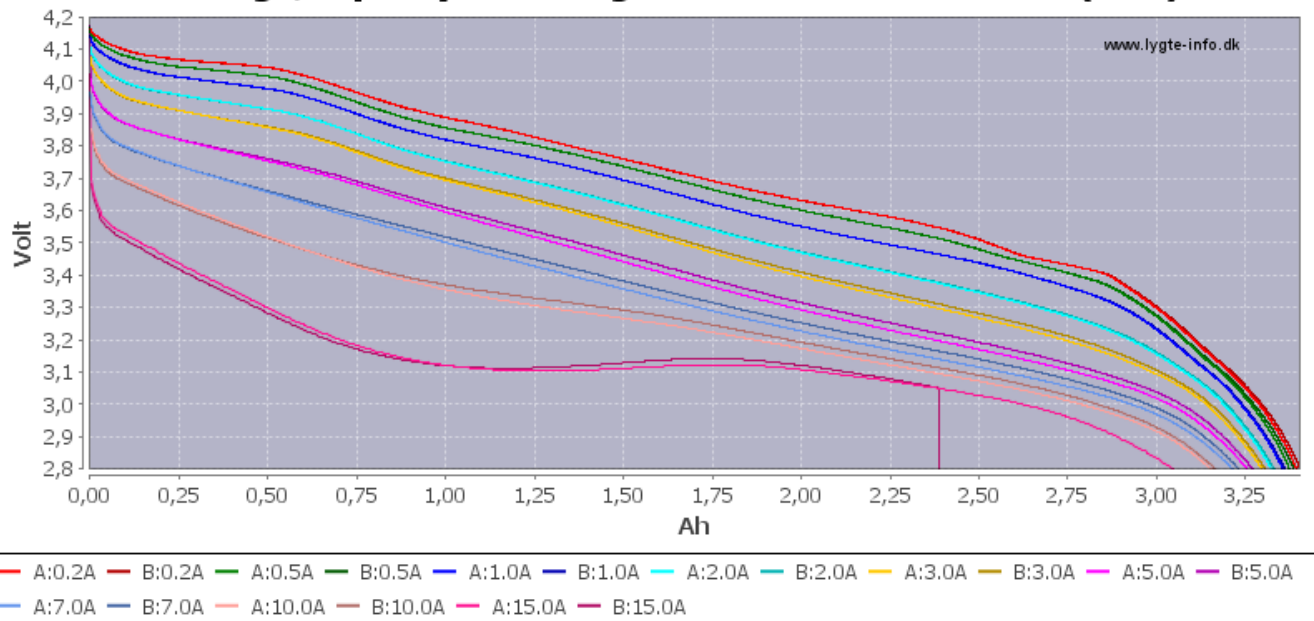


Abbildung 1: Quelle: <https://lygte-info.dk>