Power management of the Heidi tracker

The Arduino is powered by 2 batteries. On one hand the regular board voltage supply and on the other hand a backup battery, which prevents a brown-out of the Arduiono due to voltage drops caused by the high peak currents of the GSM module from a board voltage of U_{batt} < approx. 3.8V. Both supply lines are connected via a current switch, 2 Schottky diodes. This circuit leads to a equal voltage discharging of both batteries. .

So the Arduino gets its supply voltage via 1 Schottky diode of type 1N5817 each. This diode has a voltage loss of about 0.3V in forward direction, which means that U_{batt} - 0.3V arrives at the Arduino.

The ESP32 works correctly down to about 2.6V, as far as no internal consumers like GPIO's need additional current. As soon as the power consumption increases just a little bit, a brownout reset may occur (a big capacitor does not change this bevavior).

In the range below 2.6V the behavior of the ESP32 is undetermined - it boots in a circle.

The deep discharge protection of the solar loader modules cuts the load from the batterie at $U_{batt} < 2.4V$ and switches it on again at $U_{batt} \geq 3.0V$.

The following voltage limits are fixed:

• $U_{batt} \geq 3{,}6V$	– normal function
• $3{,}6V > U_{batt} \geq 3{,}5V$	– doubled times between regular data transmissions, alarms normal
• $3{,}5V > U_{batt} \geq 3{,}4V$	– no regular data transmissions, alarms normal
• $U_{batt} < 3{,}4V$	– deep sleep for 15 minutes
• $U_{batt} < 3{,}3V$	– deep sleep for 60 minutes

The aim of the power management is to be able to send alarms as long as possible when the batteries are underpowered and to avoid a supply voltage of the Arduino below 2.6 V later on.

From 3.5 V battery voltage the battery has only very small energy content. Therefore the regular transmission of data is stopped. Alarms are not sent from 3.4 V, because the voltage drops may cause that the transmission fails.

To avoid the relatively fast discharge of the battery at the range from Ubatt < 2.8 V to Ubatt < 2.4 V, a Schmitt-trigger circuit with appropriate hysteresis can be installed for a cut off of the Arduino from Ubatt < 3.0V.

Extreme low-supply scenario:

Pos	U_{batt}	Operating status	Charging status
1	< 3,6 V	save power by doubling the time between data transfers	
2	< 3,5 V	data transmissions are stopped (measurements continue), only alarms are sent (with transmission of the location)	
3	< 3,4 V	no more activities, check battery status every 15 minutes	
4	< 3,3 V	no more activities, check battery status every 60 minutes	
5	< 3,0 V	the ESP32 triggers a brown-out reset when checking the battery status, the brown-out is detected during the next boot process and a deep sleep of 60 minutes is set.	
6	< 2,8 V	the ESP32 boots in a circle and consumes 30 mA permanently, the battery is further discharged	
7	< 2,4 V	the charge controller of the battery cuts off all loads	
8	> 2,4 V	the charge controller of the battery keeps all loads cut off.	
9	> 3,0 V	charge controller of the battery turns back on all loads → position 5 - the ESP32 boots at least in a controlled manner	
10	> 3,3 V	→ position 4	
..	..	etc.	

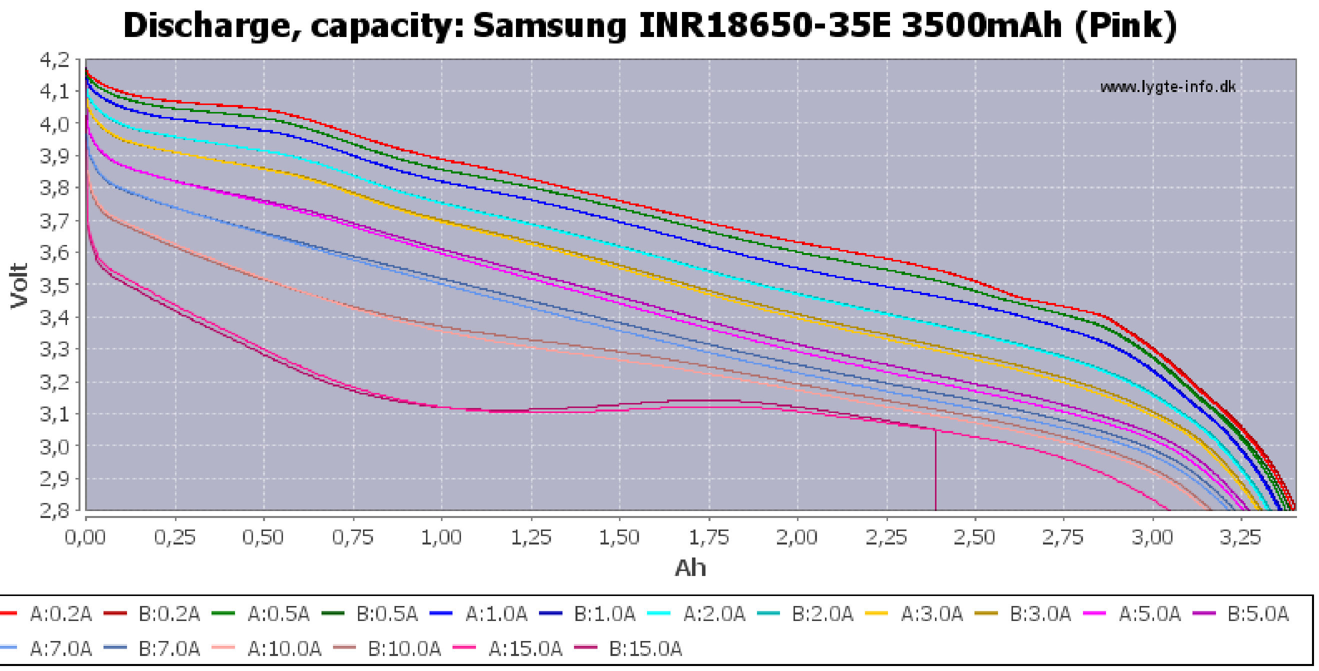

Abbildung 1: source: https://lygte-info.dk