Battery Life Modelling 101

Validating Battery Models with P1125

A key part of a successful battery powered product is ensuring that customers are happy with the battery life or run time. There are several parameters that can be altered during the design phase to help optimize battery life but knowing how much specific events or product features add/affect your overall battery life is key. Time can be wasted optimizing an event/wakeup or state that contributes little to overall battery life. These same techniques can also be used for root causing shorter than expected battery life as well.

The starting point is to create a model that predicts the expected battery life.

Identify Product Power States

Start by identifying the power states of your product. We will assume that our product has a few distinctive states based on a typical IoT type device.

- Deep Sleep: This is the state where the product consumes the least amount of current, generally where there is no activity in sensors or processors.
- Sensor Read: This is the state where a sensor or sensors are read on a periodic timer, for example temperature.
- Radio/Cloud Update: This is the state where the product turns on a radio and sends data to the cloud.

The power distribution design, how your product electrically connects the various points of load(s) to the battery, will determine what current that battery will actually deliver for each of these states. Generally, the battery is not directly connected to ICs, and must go through DC/DC conversion, either with switched mode power supplies (SMPS) or with linear regulators, or a combination of both. It is beyond the scope of this short brief to examine such architectures.

For the purpose of this brief, it is assumed the engineer, based on the architecture of her design, has determined the following battery currents[1] for the three states of the product,

- Deep Sleep: 200uA
- Sensor Read: 2mA for 0.5 seconds, every minute (or 0.0167 Hz)
- Radio Update: 150mA for 10 seconds, every 4 hours (or 0.00007 Hz)

These values are determined by reviewing design IC/module datasheets, which can take some time in interpreting. For example, embedded microcontrollers generally have several standby modes, with each mode having different properties. The software and hardware designers optimize the selection of modes based on product requirements.

[1] All currents that make up the load must be "battery referred".

Battery is a bucket of Charge

The battery is a bucket of charge that gets consumed by the product to do work. Batteries are typically specified in "mAhr", milli-Amp hours. Sometimes this is also called the "1C discharge rate", or the maximum amount of current (1C) that can be (safely) discharged. This specification is the amount of current (rate of charge) that the battery can supply for 1 hour at which time it will be fully discharged.

How much charge (water) is in a 1000 mAhr battery (bucket)?

$$I = Q/t$$
$$Q = I \cdot t$$
$$Q = 1.0\ Amps\ \cdot 3600\ seconds$$
$$Q = 3600\ Coulombs$$

Note: mAhrs is basically coulombs but in an hour based instead of seconds based. Or alternatively,

$$Q = mAh * 3.6$$

Therefore 1000mAhr is 3600 Coulombs.

Battery Voltage

The battery voltage also plays a part in battery life, but it is generally a second order effect for a well architected product. Different types of battery chemistry have different voltage profiles.

As mentioned above, the electrical design usually involves SMPS and/or linear regulator devices, have an efficiency that varies with input voltage, which is the battery voltage. Datasheets for such devices will typically show plots of efficiency versus load current at various input (battery) voltages. Real products generally do not have DC loads, so although these plots will profile the IC (SMPS or linear regulator), they do not profile your product.

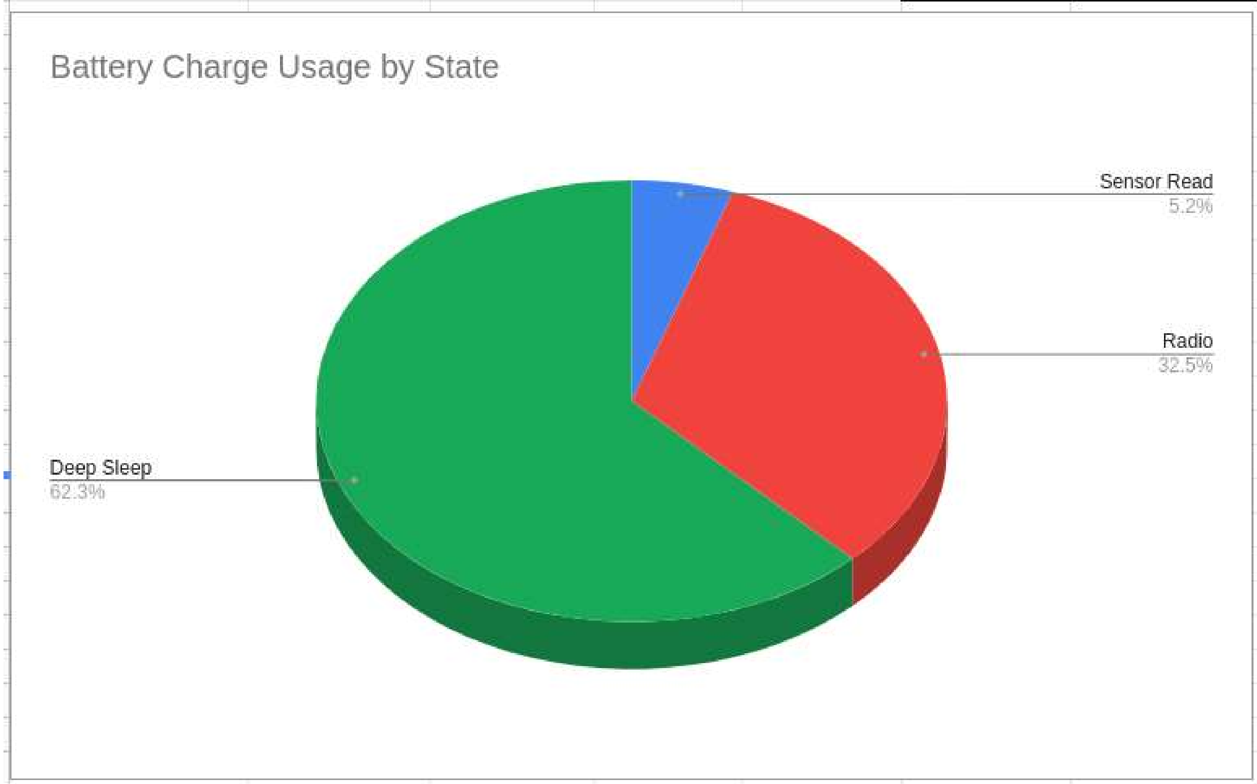
The P1125 script examples include a script that sweeps the input (battery) voltage while making a long running mAhr plot, and then plots a mAhr versus Battery Voltage plot.

Product Power Usage Pie Chart

Create a simple spreadsheet that models the charge usage of your product. Eventually the model can evolve to include power conversion efficiency of your architecture as well as more states, and or complicated usage of those states. Some of these ideas will be covered in a future whitepaper.

This is the kind of breakdown you want to see from your pie chart - where the battery charge is being used by percent for each state. Here we used a 1000 mAhr battery, and the Days were increased until the total reached 3600 Coulombs. Here the green cells are inputs and the other cells are calculated values. When determining battery life it is useful to work in seconds so goals of days or weeks need to be converted into seconds.

Days	130	0.36	Years			
Seconds	11232000					
Dynamic States	**Current (Amps)**	**Duration (s)**	**Coulombs / Activation**	**Activation Freq (Hz)**	**# Activations**	**Total Coulombs**
Sensor Read	0.002000	0.5	0.001000	0.016667	187200	187.200
Radio	0.150000	10	1.500000	0.000069	780	1170.000
Static States	**Current (Amps)**					**Total Coulombs**
Deep Sleep	0.000200					2246.400
					Total	**3603.600**

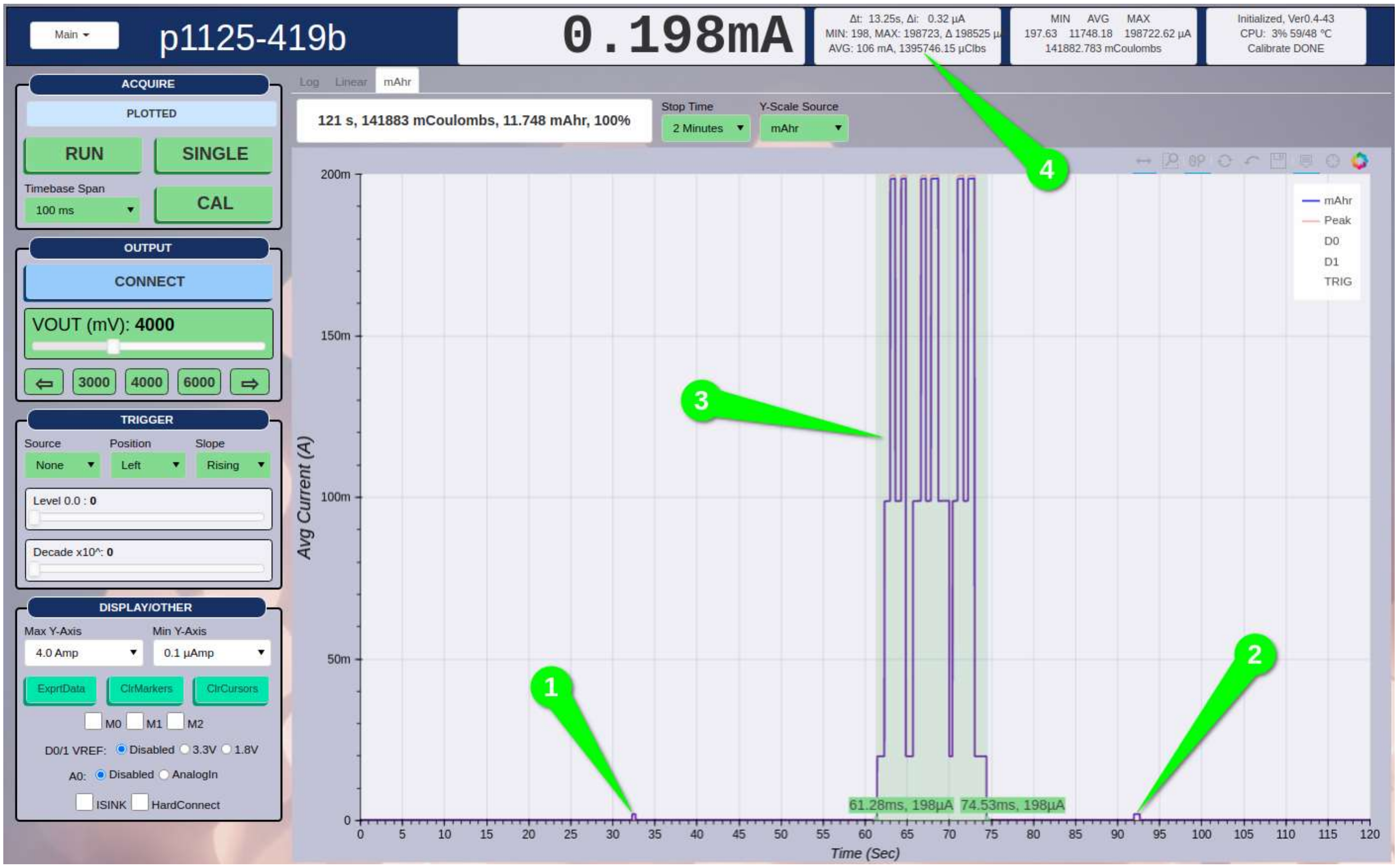


Given this breakdown, you can focus on the biggest consumers in order to increase battery life.

P1125 Measurements

For “fast” events, we use the Log/Lin plot tabs. Here is a (made up) plot for looking at the Sensor Read state [2], which also shows the Standby State [1 & 3]. Note that the measured total Coulombs for this event is 1.12 mClbs [4], which is shown with the plot Crosshair tool.

Measuring the long running Radio/Cloud event [3] the mAhr plot tab is used. Here we can see the Sensor Read events [1] and [2], which occur every minute. The Coulombs measured for the Radio/Cloud event is shown [4] using the crosshair tools to select the region of interest.



Note the mAhr plot is showing 11.748 mAhr, which only represents this 2min window. Since the repeat cycle for the Radio/Cloud event is 4 hours, one would want to run the product for >4 hours to get the correct mAhr answer. The P1125 does not support >2 hour mAhr plots via the GUI. Instead there is a script available for running very long plots.

Summary

Create a simple spreadsheet to model your product power consumption states in order to determine battery life. The model will direct your design efforts to increase battery life.

The P1125 can validate power states with its unique ability to measure a wide dynamic range of current for short and long time spans.