Memfault

An Empowering Endeavor: Predicting and Improving Device Battery Life

Tyler Hoffman
Co-Founder & Head of Developer Experience

Tyler Hoffman

Co-Founder & Engineer, Memfault

- Passion: developer tools and infrastructure for embedded engineers and companies
- Previously a Firmware Engineer @ Pebble & Fitbit
- Split time between writing firmware and building internal services to help monitor millions of devices
- Can find my thoughts and content on Memfault's Interrupt blog (interrupt.memfault.com)

pebble Memfault fitbit

Today's Agenda

- The Importance of Battery Life
- Why batteries are *hard*
- What factors into battery life
- Viewing battery life of a device in Memfault
- Understanding fleet-wide battery life in Memfault
- Best practices and recommendations

The Importance of Battery Life

Battery Life in Hardware Products

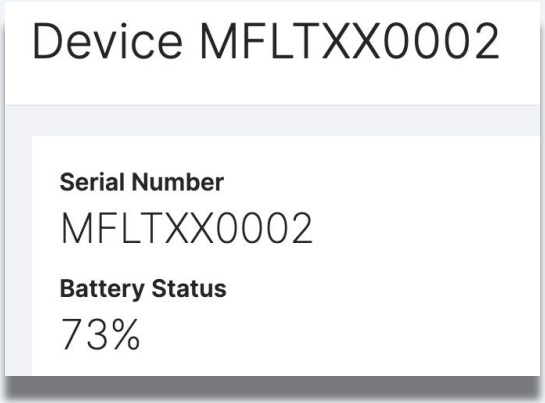
- **Critically important to customers**
- If it's out of power, it's non-functional
- Pebble Watch
  - 7 days on the box,
    - ~130mAh battery
  - #1 support issue from customers
  - We spent **a lot of engineering time** on battery life

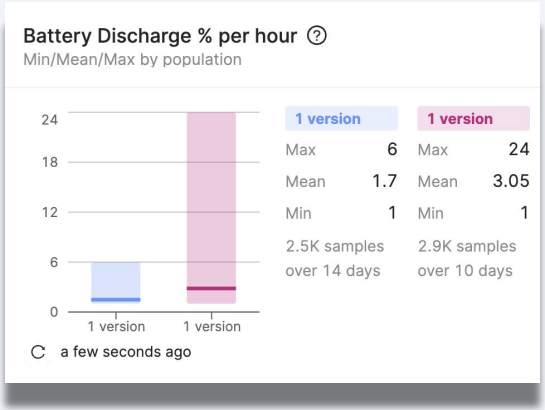
Our customers' top priority

- 60% of Memfault customers are tracking battery-related metrics for their devices
- Half of these customers have alerts set up for high battery drain

Companies care about:

- battery life (e.g. 7+ days)
- battery health (0-100%)
- battery safety (fires, high power draw)
- devices with low battery (non rechargeable devices that last years)



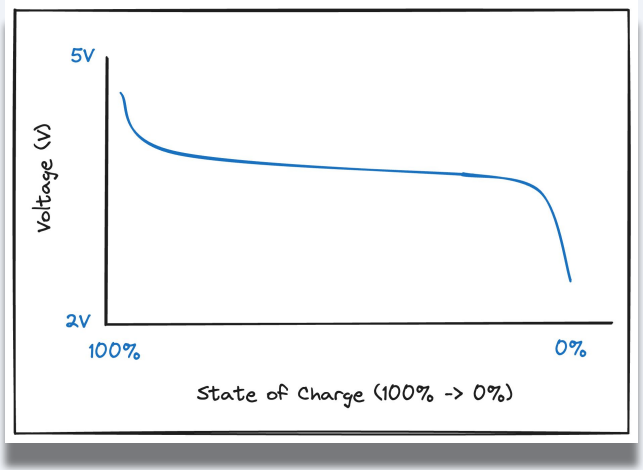


Memfault

Why batteries are *hard*

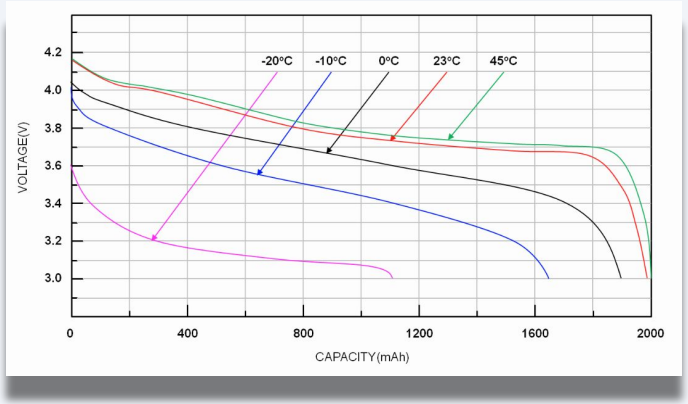
Voltage -> State of Charge

- The battery's State of Charge (SoC) is often calculated from voltage (V) to a %.
- A reliable SoC measurement requires an accurate battery curve
- For Li-ion batteries, the curve between 80% - 20% is **painfully** flat
- There are "standard" battery curves for each technology of battery, but each battery needs a custom curve in reality.
- You also need a charge curve!



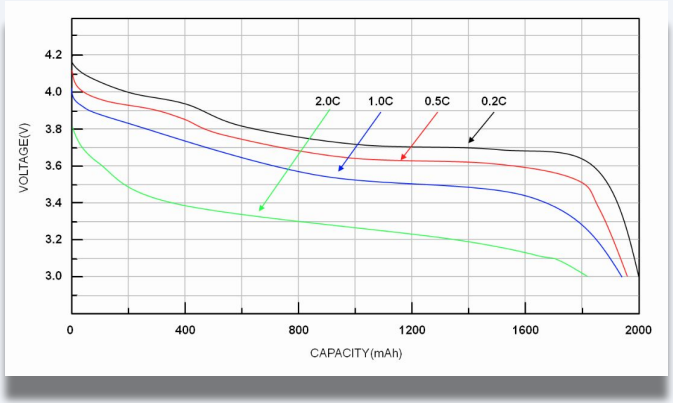
Batteries don’t behave consistently

Battery performance changes with temperature



- Under cold temperatures, performance is reduced
- **When in doubt - test devices at the top of mountains, in the ocean, and in a sauna.**

Batteries behave differently under load



- When under current draw, batteries return different voltage (V) readings
- **Measure V during known or expected current draw**

More difficult for low power devices

- Battery fuel gauges are amazing!
  - State of charge (SoC)
  - State of health (SoH)
  - Tracks the power in and out of batteries
  - Can provide accurate discharge rates
- They consume power
- We had one on our Pebble watches - did not use it
- Many devices opt not to include or use one

What factors into battery life

Poll #1

Do you collect metrics about the battery and battery life of your devices?

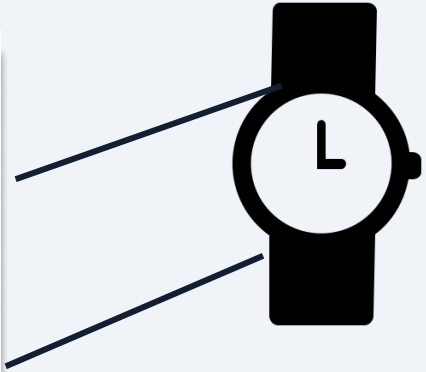
A. *Yes*

B. *No*

C. *Device doesn’t have a battery.*

Yet.

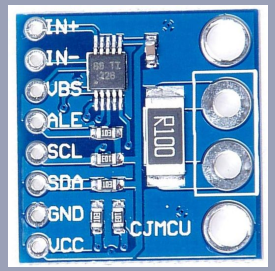
Measuring Power Consumption - Desk



Using a multimeter

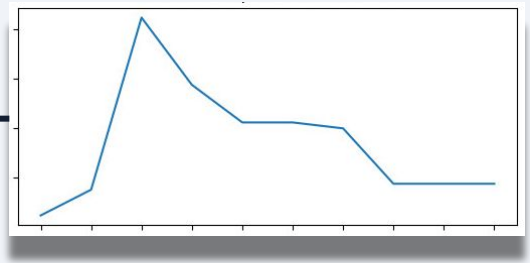
Measuring Power Consumption - Desk

Onboard



TI
INA-226

USB

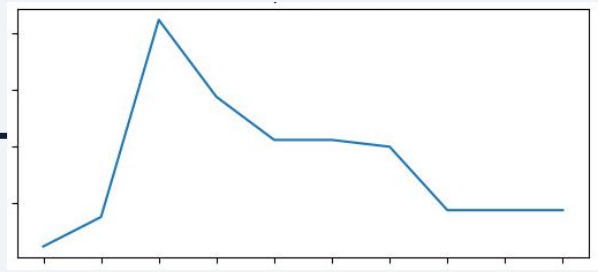


Using an onboard power monitor

Measuring Power Consumption - Device

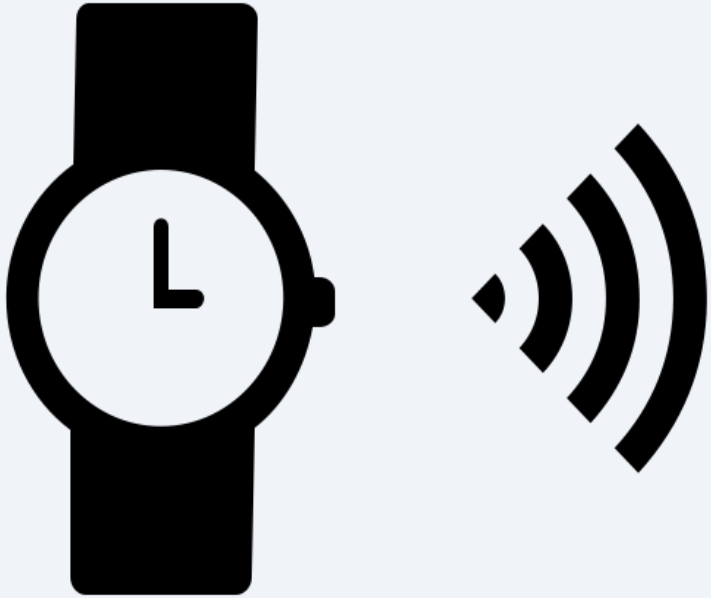
Integrated Fuel Gauge

USB



Counting coulombs using a fuel gauge

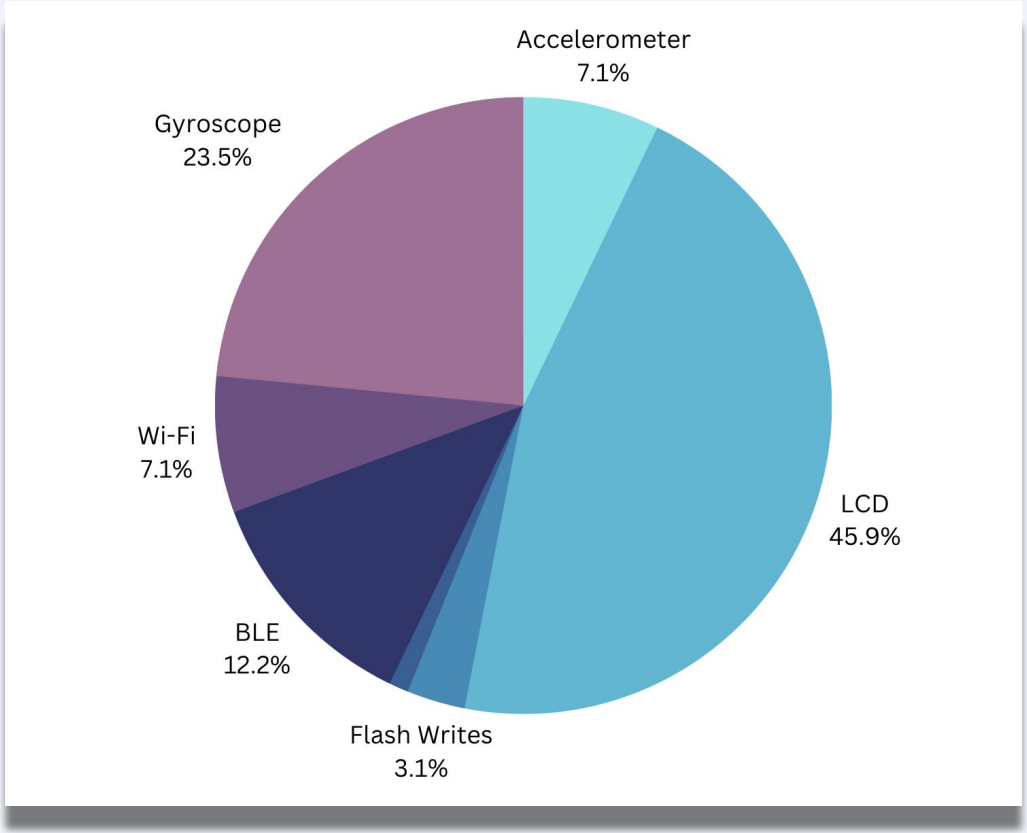
Measuring Power Consumption - *Device*



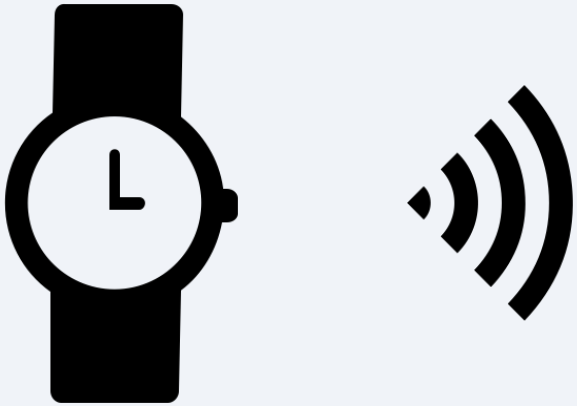
(without a fuel gauge)

Measuring Power Consumption - *Device*

We want a distribution of where power is being spent



Measuring Power Consumption - *Remotely*



```
Example App Booting
Metric ID: 1 -- Value: 15000  # elapsed ticks
Metric ID: 2 -- Value: 8531   # Main Task ticks
Metric ID: 3 -- Value: 8223   # Timer Task ticks
Metric ID: 4 -- Value: 14     # Timer Task count
Metric ID: 5 -- Value: 3593   # "sensor on" ticks
Metric ID: 6 -- Value: 4696   # heap-free low watermark
```

Metrics of course

Metrics

Uncover trends on anything that is numerical

- Task runtimes
- Peripheral utilization and power states
- CPU state tracking (sleep, running)
- Connectivity radio status

```
Example App Booting
Metric ID: 1 -- Value: 15000  # elapsed ticks
Metric ID: 2 -- Value: 8531   # Main Task ticks
Metric ID: 3 -- Value: 8223   # Timer Task ticks
Metric ID: 4 -- Value: 14     # Timer Task count
Metric ID: 5 -- Value: 3593   # "sensor on" ticks
Metric ID: 6 -- Value: 4696   # heap-free low watermark
```

Power profiling

- Run with peripheral powered off
- Measure power of peripheral **disabled**
- Instrument & exercise peripheral
- Measure power of **busy peripheral**
- Take the delta of power consumption

Estimate power draw by time or by usage

*µAh per **external flash sector erase***

*µAh per **BLE minute scanning***

*µAh per **second LCD backlight on***

*µAh per **second vibe motor***

Metrics to track for power profiling

1

Connectivity

- Radio utilization (bytes, packets, time on/off)
- Radio strength, disconnections

2

Peripherals

- GPS, accelerometer, gyroscope, cameras, compass, LED's, etc.
- Utilization, power settings, storage read/write activity

3

CPU metrics

- Task runtimes
- CPU mode tracking (sleep, deep sleep, running)
- Time blocked to detect thrashing

4

Extra Information

- Operating temperature, environment temperature
- Battery cycle count, battery health

Battery life of a device in Memfault

Poll #2

How difficult do you find debugging battery life issues to be?

A. *Very difficult*

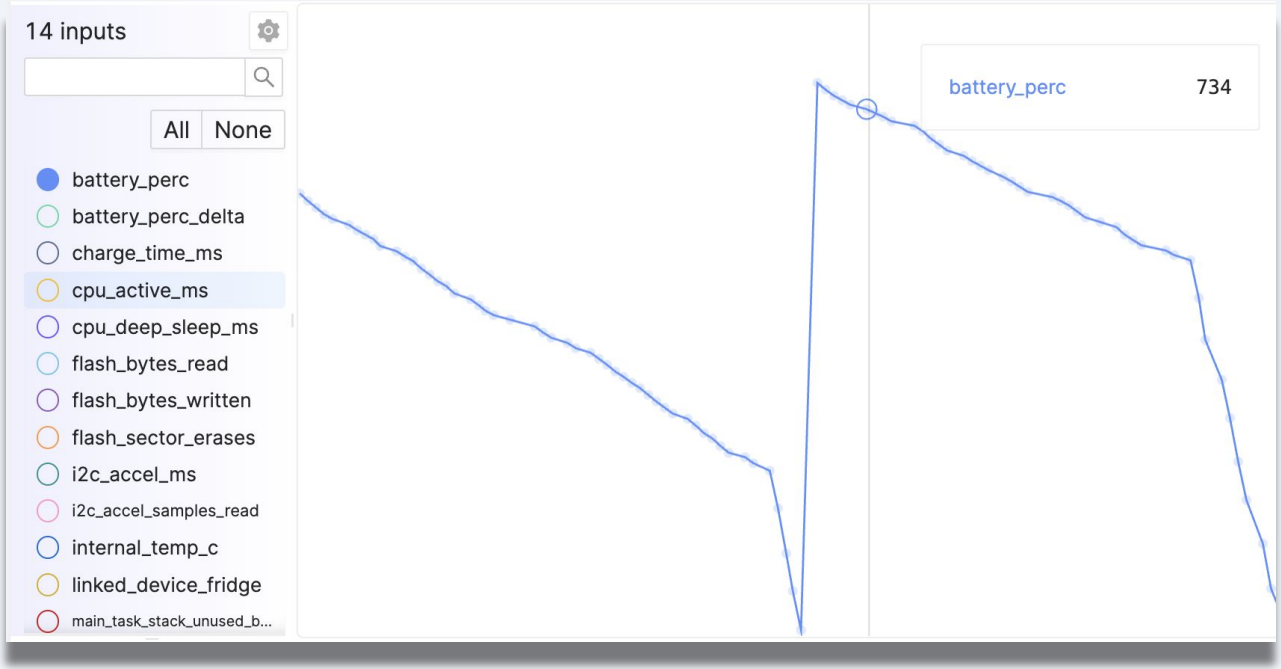
B. *Somewhat difficult*

C. *Easy*

D. *Very easy*

E. *Device is powered. I'm lucky!*

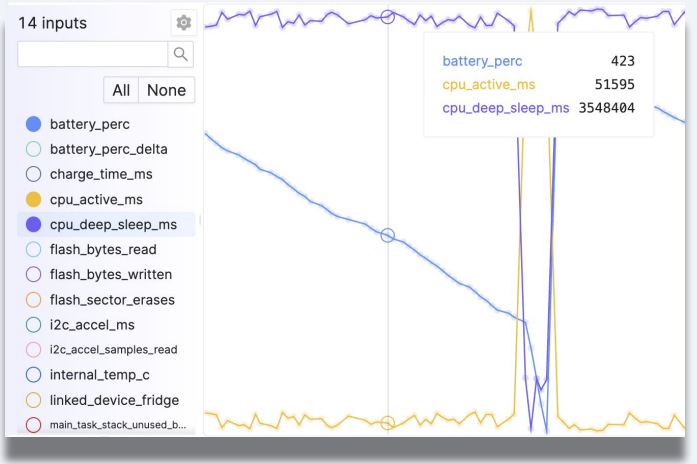
Battery on the Timeline

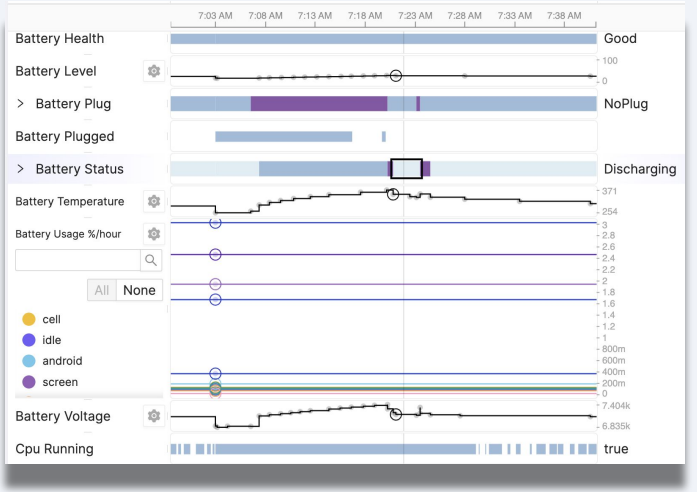


Demo - Correlating battery issues with metrics

Using Timeline for Battery Life

- Debugging battery issues on a single device
- The Timeline view of battery and related metrics is great for Customer Support teams
- Can correlate customer issues and complaints to potential power issues
- **Coming soon for Android:** State of Health
  - Current capacity / theoretical capacity





Understanding fleet-wide battery life in Memfault

Projected Duration of Battery Life

We want to know the **projected battery life** (typically in days)

or

Typical Approach

- Thousands of devices
- Send the battery’s SoC in % sporadically with a timestamp
- Store this metric data warehouse
- Run a SQL query on the data
- Profit?

Typical Approach

- Thousands of devices
- Send the battery's SoC in % sporadically with a timestamp
- Store this metric data warehouse
- Run a SQL query on the data
- Profit?

Issues with approach

- Dropped data ruins the calculations
- Timestamps aren't always accurate
- Devices lose time (all the time)
- Charging devices makes the calculation even more difficult
- The SQL query is **hard** if not impossible
- Doing this for millions of devices will cost $$$ in data warehouse

Recording State of Charge (SoC)

Device A	Device B	Device C	Device D
75%	23%	92%	5%
72%	21%	89%	2%
67%	19%	85%	**10%**
34%	19%	**100%**	7%
78%	**21%**	97%	5%
50%	**100%**	94%	**15%**
100%	92%	92%	12%

bold = % increase (battery was charged)

Compute projected battery life using SoC → ***Difficult***

Better Approach

- Infinite number of devices
- Send the battery’s SoC drop in % at regular intervals
- Compute a simple average

Regularly send battery SoC Drop

Device A - % battery drop per hour

80 - 77 = 3%
72 - 67 = 5%
25 - 23 = 2%
23 - 74 = -51% | ***DISCARD***
50 - 48 = 2%
10 - 7 = 3%

3% average battery drop per hour

Discard readings where charger & power event took place

Compute projected battery life using SoC Drop → ***Easier!***

Projected Battery Life Calculation

$$Projected\ Hours\ of\ Battery\ Life = \frac{100}{Average\ SoC\ drop\ per\ hour}$$

$$\frac{100}{1.1\%\ drop\ per\ hour} = \sim 90\ hours\ of\ battery\ life$$

Tracking SoC Drop with Memfault

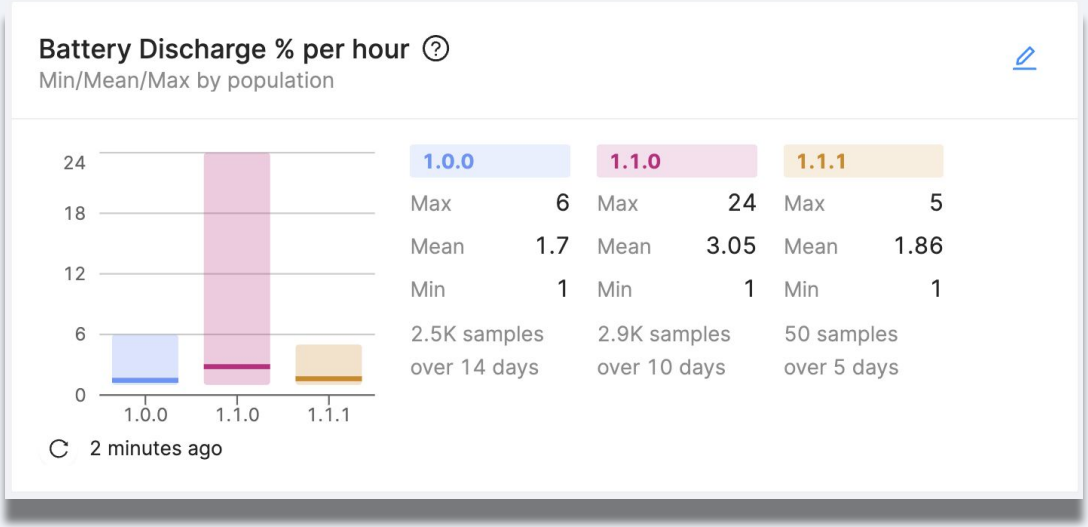
MCU & Linux

- Capture a metric **Battery_ChargeLevelDrop**, value from 0-10,000, which represents the battery's state of charge drop within a heartbeat
- E.g. 87% to 84% over the course of the heartbeat interval, report **300**.
- Chart **Battery_ChargeLevelDrop**

Android

- It's automatic!
- Chart the metric **battery_discharge_rate_pct_per_hour_avg**

Spotting Battery Regressions



Battery life expectancy was ~2x shorter in 1.1.0!

Best Practices › Tracking Battery Life

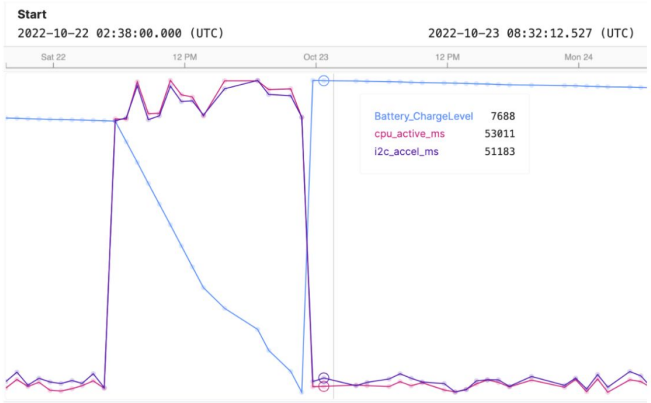
Tracking Battery Life with Memfault

Memfault believes that the expected and the actual battery life of devices are two of the most important reliability metrics for devices that are not connected directly to power sources. Unfortunately, understanding and predicting battery life trends and regressions with thousands to millions of devices in the field is *hard*.

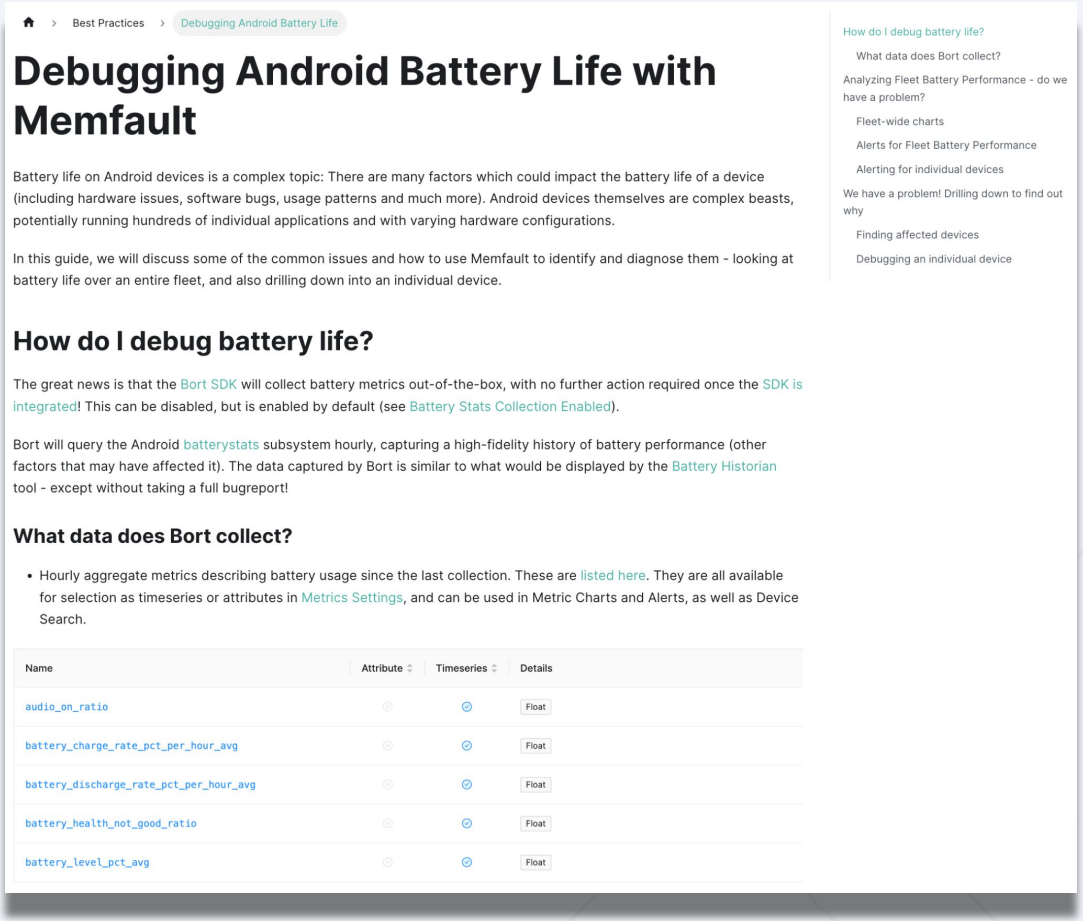
In this guide, we will walk through the benefits of tracking battery life with Memfault, how to understand and predict battery life for a single device as well as across an entire fleet of devices, and finally talk through common pitfalls when monitoring battery life.

Tracking Battery Life with Memfault

By using Memfault's device SDKs and the heartbeat metrics, the difficulties of monitoring battery life will become a problem of the past. Memfault provides two main benefits when it comes to monitoring battery life: the ability to view a single device's battery status over time, and the ability to understand battery trends across an entire fleet of devices.



https://docs.memfault.com/docs/best-practices/metrics-for-battery-life/



Best Practices › Debugging Android Battery Life

Debugging Android Battery Life with Memfault

Battery life on Android devices is a complex topic: There are many factors which could impact the battery life of a device (including hardware issues, software bugs, usage patterns and much more). Android devices themselves are complex beasts, potentially running hundreds of individual applications and with varying hardware configurations.

In this guide, we will discuss some of the common issues and how to use Memfault to identify and diagnose them - looking at battery life over an entire fleet, and also drilling down into an individual device.

How do I debug battery life?

The great news is that the Bort SDK will collect battery metrics out-of-the-box, with no further action required once the SDK is integrated! This can be disabled, but is enabled by default (see Battery Stats Collection Enabled).

Bort will query the Android batterystats subsystem hourly, capturing a high-fidelity history of battery performance (other factors that may have affected it). The data captured by Bort is similar to what would be displayed by the Battery Historian tool - except without taking a full bugreport!

What data does Bort collect?

- Hourly aggregate metrics describing battery usage since the last collection. These are listed here. They are all available for selection as timeseries or attributes in Metrics Settings, and can be used in Metric Charts and Alerts, as well as Device Search.

Name	Attribute	Timeseries	Details
audio_on_ratio			Float
battery_charge_rate_pct_per_hour_avg			Float
battery_discharge_rate_pct_per_hour_avg			Float
battery_health_not_good_ratio			Float
battery_level_pct_avg			Float

How do I debug battery life?
- What data does Bort collect?

Analyzing Fleet Battery Performance - do we have a problem?
- Fleet-wide charts
- Alerts for Fleet Battery Performance
- Alerting for individual devices

We have a problem! Drilling down to find out why
- Finding affected devices
- Debugging an individual device

https://docs.memfault.com/docs/best-practices/android-battery-debugging

Best Practices

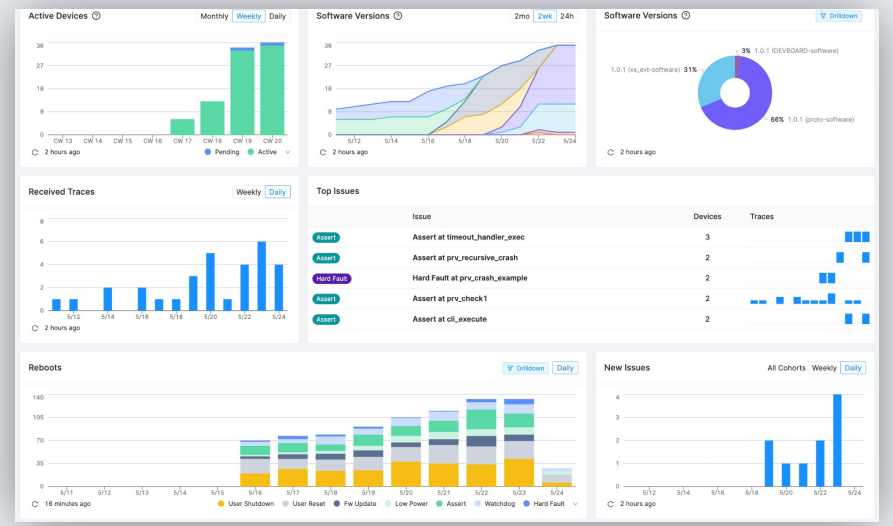
Tips from us at Memfault

- Develop a battery curve ASAP to go from V to SoC %
  - Improve it over time
- Record both SoC and SoC drop
  - SoC for customer support
  - SoC drop for fleet-wide aggregations
- Ignore data points when a charger was connected - ruins your averages of SoC drop
- Ship firmware frequently - monitor always

Memfault

Memfault

- Single tool for all teams
- Dashboards & metrics
- Diagnosing crashes
- Device logs and metadata
- OTA distribution and deployments
- Custom Alerts



Trusted By

See more at https://memfault.com/customers/

“Memfault gives us the hard data to be confident in the reliability of our firmware and proactively take action, resolving issues before our users are impacted.”

Raman Thapar
Director of Engineering,

What We Covered Today

- The Importance of Battery Life
- Why batteries are *hard*
- What factors into battery life
- Viewing battery life of a device in Memfault
- Understanding fleet-wide battery life in Memfault
- Best practices and recommendations

Thank You!

- Learn more about Memfault: memfault.com
- Find me on:
  - LinkedIn: https://www.linkedin.com/in/tyhoff/
  - Twitter: https://twitter.com/ty_hoff
- Read my posts on Interrupt (interrupt.memfault.com)

Battery Life Resources:

- Best Practices: Tracking Battery Life with Memfault
- Best Practices: Debugging Android Battery Life with Memfault
- Interrupt: Understanding Battery Performance of IoT Devices

Tyler Hoffman

Co-Founder & Engineer, Memfault

Memfault

Memfault