

BPOL

BATTERY PERFORMANCE **OPTIMISATION LIBRARY**

Facing an old problem

Low battery life time on modern day smartphones is an infamous problem we've all learned to live with.

Nowadays the words 'Can last a full day with normal usage!' can even be found on adverts. This would be unimaginable not so long ago, in a time where cell phones would last a full week without charging, and even then it was always considered a hassle.

Certainly we shouldn't go backwards? Sadly, it's not that simple. But even then, we haven't been trying our best to fight this predicament.

A hardware problem... But also a software one

It's certainly easy to blame hardware and its manufacturers. After all, we've been using pretty much the same battery technology for ages, while every other part has seen very significant and incremental improvements over the years.



Contact

Rua Alfredo Allen, 455 4200-135 Porto, Portugal

+351 220 430 300 info@fraunhofer.pt www.fraunhofer.pt

Having pocket computers with amazing screens and raw power does come at a cost - battery life. But while hardware is the main culpable party, it also comes down to how the software manages the available resources. And this, we can somehow control and certainly



Fig1. Three pillars of this project -Android library, mobile application using the library, data visualization.

improve, while waiting for big advances on battery technology.

A better insight on battery usage

This project is all about getting more and better data from everyday battery usage, making it easily available to developers.

By providing this insight, developers will be able to better understand how their software is impacting the device, therefore giving them the means to find and mitigate problems in this area.

Better knowledge, better code

The ultimate goal of this project is for developers to deliver code that is more power friendly and aware of its impact on the host system. The aim is to simplify this process by abstraction of the problem and easy access to metrics.

Main goals

- Fetching and storing relevant usage data
- Providing a simple interface to analyse the data
- Use retrieved data to make useful estimations
- Facilitate an API with commonly wanted features
- Facilitate automation of some common behaviours
- Be easily expandable and flexible



Fig2. Data visualization tool.

It's easy to be in a certain level of abstraction and thinking the operating system will manage everything for us, but we can easily find evidence that most big battery drains come from problems in the software.

Making use of gathered information

While gathering and making data available is a big point, not doing anything with that data would be missing out.

The library also provides the means to extrapolate and visualize meaningful data, such as estimations and graphics, and, on an Application Programming Interface (API) level, provides configurable call-backs for important events – like if battery expectancy estimations are below the project requirements. Configuring an application with behaviours that listen to these events, like dimming the screen or disabling data synchronization, comes naturally.

The advantages of automation

Having important events being tracked by the library, automating battery saving commands becomes much simpler, and the library facilitates them.

This will allow developers to save a lot of time, providing not only better code and battery usage, but also letting developers focus on other things and reduce costs.



Fig3. Snapshot of the developed test application.