



HAL
open science

Quantifying the tiny-Small design of the SmolPhone

Aloïs Rautureau, Joseph Paturel, Martin Quinson, Simon Rokicki

► **To cite this version:**

Aloïs Rautureau, Joseph Paturel, Martin Quinson, Simon Rokicki. Quantifying the tiny-Small design of the SmolPhone. ICT4S 2024 - 10th International Conference on ICT for Sustainability, Jun 2024, Stockholom, Sweden. pp.1-4, 2024. hal-04589322

HAL Id: hal-04589322

<https://inria.hal.science/hal-04589322>

Submitted on 27 May 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Quantifying the tiny-Small design of the SmolPhone

Aloïs Rautureau^{1,†}, Joseph Paturel^{1,†}, Simon Rokicki^{1,†} and Martin Quinson^{1,†}

¹Univ. Rennes, IRISA, Inria, CNRS.

Abstract

The SmolPhone project explores the design space of modern smartphones with energy limits, increasing the battery lifetime possibly at the price of a reduced feature set. The energy consumption of classical smartphones can roughly be divided in three main parts: the processing units, the screen, and the network connectivity. This paper evaluates the practical feasibility of the SmolPhone vision, which involves heterogeneous multiprocessors and low-power technologies for the screen and network. We identify several usage modes according to the provided features, and quantify the energy consumption of the processing units in each mode, as well as the time required to switch between modes.

Keywords

Sustainable Computing, Mobile Computing, Frugal Design

1. Introduction

The SmolPhone [1] project is an action research on the topic of low-tech in the domain of IT systems. Its practical aspects consist in designing a sort of low-tech smartphone offering some services of a classical smartphone with a one-week battery lifetime. The goal is not to optimize a typical smartphone but rather to reconsider the smartphone design space, exploring unusual architectures and evaluating the set of features that should be part of long-lasting smartphones.

The current work evaluates the practical feasibility of the “tiny-Small” hardware design, which consists in the use of two compute units on the same board: a microcontroller plays the role of the tiny unit with low power and tiny performance, while a small processor provides some more performance at the price of higher power consumption. The main idea of the tiny-Small design is to rely as much as possible on the tiny compute node, and dynamically pass the control to the other compute node on need. The rest of this paper is organized as follows. Section 2 introduces several usage modes that we have identified and discusses the capabilities offered to the applications in each mode. Section 3 describes our first prototype while Section 4 quantifies the power consumption in each mode and the switching time between these modes. Section 5 concludes on the potential feasibility of a low-tech smartphone leveraging the tiny-Small design.

2. Usage modes of a tiny-Small device

Adapting the processor mode to the applications needs is a classical idea to reduce the power

ICT4S'24: Symposium on ICT for Sustainability, June 24–28, 2024, Stockholm, Sweden.

[†]These authors contributed equally.

 0009-0007-8507-2167 (A. Rautureau); 0009-0008-6120-2147 (J. Paturel); 0000-0002-0195-096X (S. Rokicki); 0000-0001-7408-054X (M. Quinson)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

consumption without hindering the user experience. Computers adapt the processor frequency to that extend. The tiny-Small design pushes this idea further thanks to the use of two processing units. The goal is to rely as much as possible on a low-power microcontroller (MCU) with advanced power management features. Such unit is so limited that it can only run simple and dedicated applications, while classical application relying on an operating system are too demanding for such a tiny device. Our design is thus completed with a small processor (CPU) running Linux applications on need. Overall, we identified five main usage modes for our tiny-Small design, each exhibiting different power and performance profile.

To save energy, we strive to rely on the MCU as much as possible. To that extend, we plan to implement a dedicated version of the simple applications of a smartphone such as calls, text messaging, agenda, notepad or epub reader. This enables a **main operating mode** where the MCU is the main processing unit and the CPU is completely turned off.

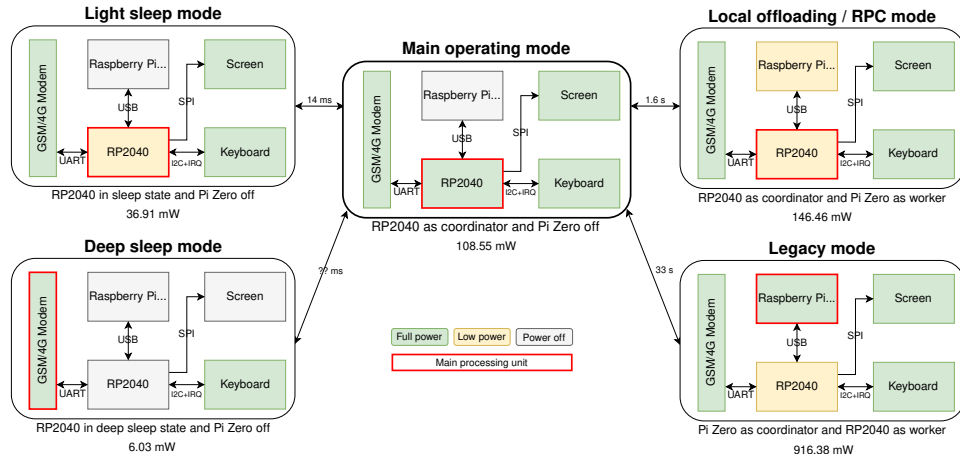
When the interactions with the user allow, the consumption can be further reduced using the power management capabilities of the MCU, which include two sleep modes. The MCU can switch very quickly to a **light sleep mode** where the compute cores are switched off while the internal clocks remain active. The MCU can then be woken up by any interruption. The MCU can also enter a **deep sleep mode** where the chip is almost completely shut down, including the internal clocks. It can then only be woken up by interrupts from external sources, such as a button or a notification from the modem. We intend to put the MCU in light sleep mode several times per second when the interaction with the user permits (leaving the screen and keyboard active), while the device will switch to the deep sleep mode when the screen is turned off, e.g. at night or when the device returns in a pocket.

Performance critical and resource hungry codes that do not fit on the MCU are executed on the CPU. It can be either in **legacy mode** when the applications require complete Linux environment, or a lighter **local offloading mode** where the applications run on the CPU directly, possibly within an unikernel environment where the application is relieved from the hardware handling by a dedicated library. This mode is well adapted to resource hungry applications which are often used on a smartphone such as video rendering, photography or GPS maps rendering. These applications are common enough on a smartphone to justify the development of unikernel implementations to benefit from the energy savings and fast switching time enabled by the local offloading mode. Any application for which no unikernel implementation exists should however be executed in legacy mode, despite its resource burden.

3. First tiny-Small prototype

We implemented a prototype using a Raspberry 2040 (RP2040) as a MCU, and a Raspberry Pi Zero (PiZ) as a CPU. We complemented this design with a 2.9" LCD 2.0 display (Sharp matrix display) featuring low power consumption and high reactivity [2]. Adding a 4G LTE modem, a Wi-Fi module, and a small mechanical keyboard is still future work [1].

The legacy mode uses the 2024-03-15 release of Raspberry Pi OS Lite distribution with the Bluetooth and WiFi connectivity disabled on the PiZ. The local offloading mode is implemented using the 2024.03.0 release of the Barebox bootloader, while the main operating mode is implemented in Rust using the Slint UI library for convenience.



To switch from the main operating mode to the offloading mode, the RP2040 wakes up the PiZ and calls the procedure to execute by running a command in the Barebox bootloader. The RP2040 then waits for the PiZ to return the command's result over USB. Switching to the legacy mode works similarly, using the Barebox boot loader to start the operating system. The application can use the screen and keyboard through an USB connection, for which the RP2040 acts as a driver. The control is automatically returned to the RP2040 when Linux is shut down.

4. Experimental evaluation

To evaluate the power consumption of each usage mode and the switching delays, the RP2040 cycled through each mode, turning a GPIO high to indicate when it switched to the next one. In the active modes, a matrix multiplication application is used to keep the processing unit busy. We used a Rigol MSO8204A oscilloscope to accurately measure the switching time between modes, while a Keysight 34465A digital multimeter was used to measure the power consumption of each mode. The reported values are presented in the figure above.

These values include the screen consumption (0.44 mW on average, peaking at 0.86 mW when drawing). No GSM modem, WiFi chip nor keyboard was included in these experiments.

5. Conclusion

In this work, we evaluated the applicability of the tiny-Small design for a low-power smartphone. Combining a MCU and a CPU allows to both dramatically reduce the power consumption for basic applications, while allowing the use of more resource hungry applications. Experimental evaluations show that this design could allow about 50 hours of full MCU usage on a tiny 1000 mAh battery, and probably twice this amount when leveraging the light sleep mode.

In the future, we plan to extend this prototype with a LTE Cat1 modem for the GSM and data connectivity [3], and measure the power consumption in the energy saving modes of such device. We also plan to develop some basic applications on the MCU and build a full hardware prototype to conduct some experiments in real settings.

References

- [1] J. Paturel, C. Quinson, M. Quinson, S. Rokicki, SmolPhone: a smartphone with energy limits, in: IGSC 2023 - 14th International Green and Sustainable Computing, C. Mani Krishna and Michele Magno, Toronto, Canada, 2023, p. 4. URL: <https://inria.hal.science/hal-04156447>.
- [2] T. Grosse-Puppenthal, S. Hodges, N. Chen, J. Helmes, S. Taylor, J. Scott, J. Fromm, D. Sweeney, Exploring the design space for energy-harvesting situated displays, in: Proceedings of the 29th Annual Symposium on User Interface Software and Technology, UIST '16, Association for Computing Machinery, New York, NY, USA, 2016, p. 41–48. URL: <https://doi.org/10.1145/2984511.2984513>. doi: 10.1145/2984511.2984513.
- [3] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, O. Spatscheck, A close examination of performance and power characteristics of 4g lte networks, in: Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services, MobiSys '12, Association for Computing Machinery, New York, NY, USA, 2012, p. 225–238. URL: <https://doi.org/10.1145/2307636.2307658>. doi:10.1145/2307636.2307658.