







BATTERY-FREE GAME BOY

Team: Jasper de Winkel, Vito Kortbeek, Josiah Hester, Przemysław Pawełczak
Paper: https://doi.org/10.1145/3411839
Venue: ACM IMWUT / UbiComp 2020
Code: https://github.com/tudssl/engage
Press: CNET, The Wall Street Journal, Mashable, Hackaday, The Verge, Gizmodo, Engadget, PCMag, The Register, Tech Times, Nintendo Life, Daily Mail, The Independent, r/gadgets, Naked Gaming, Seeker

This project originates from the question: can we game-on-the-go without batteries? Batteries add size, weight, bulk, cost and especially inconvenience because of constant recharging—to any device. Energy can be generated by mashing buttons while gaming, and readily available energy from sunlight is all around us, so why not use this energy for battery-free mobile gaming! Significant challenges in software resiliency and efficiency, hardware operation and energy usage first need to be solved, but would represent a fundamental advancement over non-interactive (and not very fun) battery-free devices that currently exist.

Battery-free devices can make phone calls, monitor greenhouses, even track eyes, but so far immersive interactive experiences without batteries have not been explored. This is a critical gap in the research around battery-free devices, as these types of reactive, interactive, and screen-focused systems are a significant portion of the current and anticipated smart systems.

In this project we focus specifically on this ignored part of the battery-free device ecosystem, mobile gaming, and use this application to elucidate the essential challenges that must be explored to get us to a future where reactive and user facing applications can also be battery-free.

THE BASIC PROBLEM: POWER FAILURES





Dynamic energy harvesting causes voltage fluctuations which cause frequent power failures. What would typically happen if a battery was removed from a Game Boy and replaced with solar panels is shown in this image. The game would play until energy is lost (i.e. at line 185) and then restart at the loading screen. *Intermittent computing* techniques seek to make it such that after the power failure, line 186 is then executed proceeding from the exact system state as before the failure. If this works, then play can resume at the exact frame where play ended (so that the Tetris block will be in the same place).

DESIGN: EMULATING A GAME BOY AND CHECKPOINTING



We designed the Energy Aware Gaming (ENGAGE) platform as a proof by demonstration that interactive systems are possible with pure energy harvesting and intermittent power. Four key ideas are discussed in the paper that make the system work:

- 1. Gather energy from gaming actions like button presses and from environmental energy
- 2. Track and checkpoint minimal state at the system level to enable faster "save games" and restore
- 3. Use processor emulation to play retro games to provide the nostalgia of a classic platform
- 4. Speedup intermittent computing by leveraging modern 32-bit ARM Cortex M4 microcontrollers and external FRAM. Specifically an Ambiq Apollo3 Blue ARM Cortex-M4 MCU, and Fujitsu MB85RS4MT 512 KB FRAM.



THE TEAR DOWN

The platform is built around an ARM Microcontroller (MCU) architecture, a departure from classic intermittent computing devices using Texas Instruments MSP430FRxxxx MCUs. No ARM MCUs we are aware of contain on-chip fast, byte-addressable non-volatile memory—such as FRAM—serving as a main memory. Only slow and energy-expensive FLASH memory is present. A fast, byte addressable non-volaitile memory is crucial to enable saving and restoring state in spite of power failures. To enable this, we equipped our battery-free console with an external dedicated FRAM IC. Our choice of MCU was the Ambiq Apollo3 Blue ARM Cortex-M4 operating at a clock frequency of 96 MHz, chosen for its best-inclass energy efficiency. The Apollo3 runs the Game Boy emulator and MPatch software. The FRAM is an Fujitsu MB85RS4MT 512 KB FRAM connected through SPI to the MCU providing a fast (enough) and durable method of non-volatile storage for patch checkpoints.

Energy Harvesting. We extract energy from two sources: (i) button presses of a regular Game Boy, using mechanical off-the-shelf button press harvesters, and (ii) a set of solar panels attached to the front of the Game Boy chassis. The selected buttons were ZF AFIG-0007 energy harvesting switches, used in simple scenarios like BLE messages on click in previous applications. Six of these kinetic harvesting switches are used for the the D-pad (four switches) and "A", "B" buttons. The energy from the buttons is rectified, boosted, and stored in small capacitors. Solar panels are managed by a Texas Instruments BQ25570 harvester/power management chip which has mediocre Maximum Power Point tracking ability b" ; reat cold-start and a dual boost / buck converter in a small package. Everything is stored in a 3.3

Cartridges. To be able to read game cartridges, a cartridge connector is placed on the back of the

Call Linges. To be able to read game callinges, a callinge connector is placed on the back of the platform. The cartridge interfaces with the MCU using Semtech SX1503 I/O expanders, in this case the enders also translate the 3V system voltage to 5V logic required by the cartridge. **Display.** It was incredibly difficult to source a large display that had reasonable power requirements, a excentre fires first bandy minimal with and with a citation of the second second external destination of the second secon displays because of the slow refresh rate, not suitable for gaming, and the complex driving circuitry. **Jasper de Winkel** at Delft University of Technology (TU Delft) Instead we use a Japan Display LPM013M126A LCP, which measures 26.02 × 27.82 mm. The display is is **Vito Kortbeek** at Delft University of Technology (TU Delft) smaller than the original Game Boy screen (47× 43 mm) but has a better resolution of 176 × 176 **Josiah Hester** at Northwestern University compared to 100× 144 pixels. Przemysław Pawełczak at Delft University of Technology (TU Delft) Summary. As shown below, the main components are: (A) Ambig Apollo3 Blue ARM Cortex-M4 MCU, (B) Fuiitsu MB85RS4MT 512 KB FRAM. (C)_ZF AFIG-0007 energy harvesting switch. (D) Semiconductor Northwestern University **Delft**

Can ENGAGE play all Gameboy games?

Yes! It is running a custom emulator which allows it to impersonate the Game Boy and play any Game Boy game (not the Color versions though). That said, just like with any emulator, not *every* title may work, and some bugs may prevent some games from loading some parts. More importantly, some of the games are harder to play because of the need to constantly power the device with button presses, since some games you would not press buttons very often

When the energy source is changed from solar to button-mashing, how long is the Gameboy temporarily off?

We actually harvest from both energy sources at the same time! We developed a circuit that manages this internally.

How do the buttons generate power?

The buttons generate power by moving a small but powerful magnet inside a tightly wound wire coil. The change in the magnetic field generates power. When you press the button (and when you release) it moves the magnet through the coil, this energy is then siphoned into a capacitor for immediate use by the Game Boy hardware to support all activities. This is a straightforward application of <u>Faraday's law</u>. Because of advances in manufacturing over the past decade the magnet and coil are so small that they can fit inside a button that is acceptable to a user.

Can I buy one?

Unfortunately not. But you can build one! The ENGAGE is a research platform, a proof by demonstration that sustainable smart devices are possible, useful, maybe even fun. All designs, hardware, and code to make your own ENGAGE is publicly available free and open source on <u>Github</u>.

ACKNOWLEDGMENTS

This research was supported by Netherlands Organisation for Scientific Research, partly funded by the Dutch Ministry of Economic Affairs, through TTW Perspective program ZERO (P15-06) within Project P1 and P4, and by the National Science Foundation through grants CNS-1850496 and CNS-2032408. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science F_{i} and the National Science foundation.