SmolPhone

a smartphone with energy limits

Joseph Paturel[1], Clément Quinson[2], Martin Quinson[1], Simon Rokicki[1]

1: Univ. Rennes, IRISA, Inria, France — 2: MotionLab ML GmbH, Germany.

Grenoble, 15 février 2024

SmartPhone evolution

Nokia 3310 (2000)

16 MB storage
100 MHz ARM7
13 kg eq.CO2

Iphone 3GS (2009)

32 GB storage
600 MHz ARM8 + GPU
55 kg eq.CO2

Iphone 14 Pro (2022)

Up to 1TB storage
6 cores + 5GPU + NN+Img
116 kg eq.CO2

- Modern smartphone outperform recent laptops
- Battery life: only feature to steadily decrease, despite tripled capacity

SmartPhone evolution

Nokia 3310 (2000)

16 MB storage
100 MHz ARM7
13 kg eq.CO2

Iphone 3GS (2009)

32 GB storage
600 MHz ARM8 + GPU
55 kg eq.CO2

Iphone 14 Pro (2022)

Up to 1TB storage
6 cores + 5GPU + NN+Img
116 kg eq.CO2

- Modern smartphone outperform recent laptops
- Battery life: only feature to steadily decrease, despite tripled capacity

Exponential growth vs. Finite resources and finite needs

- How could we do otherwise? Do we really need all this?
- Can we go for low-tech mobile computing?

What are the low-techs? (1/2)

Definition by Valérie Laforest (DR in Environmental and Organizational Eng.)

- Low pressure on resources: Resource efficient and recycling/reuse
- Avoid sophistication: simplify, accessible by non-knowledgeable people
- Foster social links: collaborative, participative, locally adapted
- Target non-superfluous needs: As defined by Manfred Max-Neef
  Needs are limited; satisfiers vary a lot but shouldn't be the focus

What are the low-techs? (1/2)

Definition by Valérie Laforest (DR in Environmental and Organizational Eng.)

- Low pressure on resources: Resource efficient and recycling/reuse
- Avoid sophistication: simplify, accessible by non-knowledgeable people
- Foster social links: collaborative, participative, locally adapted
- Target non-superfluous needs: As defined by Manfred Max-Neef
  Needs are limited; satisfiers vary a lot but shouldn't be the focus

Definition by ADEME

- Maximize social utility; Reduce complexity; Maintenance over replacement
- Reduce environmental impact to not overpass local and planetary limits
- Question needs toward essential ones.
- Accessible solutions: empowers broader audience w/ understanding and usage

What are the low-techs? (2/2)

Definition by the Low-tech Lab

- Accessible: buildable and repairable with no advanced tooling/knowledge
- Useful: not futile. Addressing fundamental needs.
- Durable: ecological (efficient, reuse), reparable.

Low-tech inspired by a vast literature

- Illich: *Tools for Conviviality*, 1973.
- Schumacher: *Small is beautiful*, 1973.
- Bookchin: *Toward a Liberatory Technology*, 1965.
- ...

Some initiatives toward practical applications

- Some fablabs, Low-tech lab, L'atelier paysan, etc.

Low-tech computing?

Previous definitions are not adapted to computing

- Are computers doomed as a large technosystem? cf. "Héritage et Fermeture"
- Resilient systems (efficient, durable, reusable, easy-to-use, fault tolerent)?
- Can we avoid rebound effects and expert's dictatorship?

Low-tech computing?

Previous definitions are not adapted to computing

- Are computers doomed as a large technosystem? cf. "Héritage et Fermeture"
- Resilient systems (efficient, durable, reusable, easy-to-use, fault tolerent)?
- Can we avoid rebound effects and expert's dictatorship?

ESOS project (lead by Insa Rennes)

- Sustainable, Open and Sovereign Electronic
- Bottom-up approach to the problems induced by the computing technosystem

Low-tech computing?

Previous definitions are not adapted to computing

- Are computers doomed as a large technosystem? cf. "Héritage et Fermeture"
- Resilient systems (efficient, durable, reusable, easy-to-use, fault tolerent)?
- Can we avoid rebound effects and expert's dictatorship?

ESOS project (lead by Insa Rennes)

- Sustainable, Open and Sovereign Electronic
- Bottom-up approach to the problems induced by the computing technosystem

The SmolPhone project (this work)

- Attempt toward a useful, durable and accessible mobile computing
- Top-down: Simplify hardware to the point where capabilities must be reduced

Low-tech computing?

Previous definitions are not adapted to computing

- Are computers doomed as a large technosystem? cf. "Héritage et Fermeture"
- Resilient systems (efficient, durable, reusable, easy-to-use, fault tolerent)?
- Can we avoid rebound effects and expert's dictatorship?

ESOS project (lead by Insa Rennes)

- Sustainable, Open and Sovereign Electronic
- Bottom-up approach to the problems induced by the computing technosystem

The SmolPhone project (this work)

- Attempt toward a useful, durable and accessible mobile computing
- Top-down: Simplify hardware to the point where capabilities must be reduced
- A device offering some smartphone-like features but lasting days on a charge
  - Increase battery life of a reduced set of features
- Long term goals (unrealistic now): lasting 10 years; hackable by non-specialists
- Non-goals: cheaper device; sell devices or services

Energy trade-offs

Typical smartphone consumption (from literature)

- CPU: 3000 mW
- Screen: OLED 800 mW
- Cellular: 600 mW idle / 1200 mW TX (4G – LTE Cat4)
- Wifi: 80 mW idle / 120 mW TX

Energy trade-offs

Typical smartphone consumption (from literature)

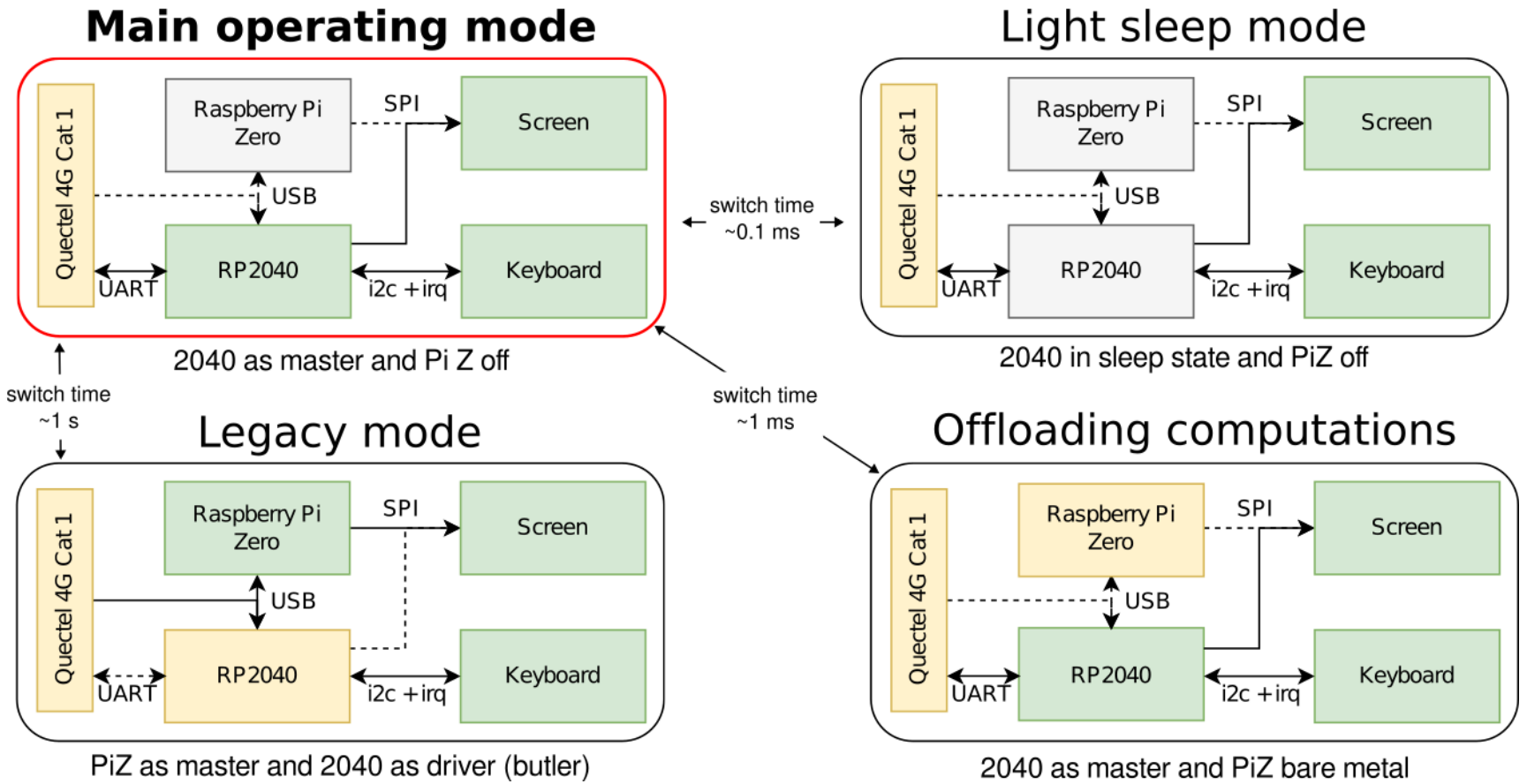
- CPU: 3000 mW
- Screen: OLED 800 mW
- Cellular: 600 mW idle / 1200 mW TX (4G – LTE Cat4)
- Wifi: 80 mW idle / 120 mW TX

Smolphone envisioned hardware

- Energy-efficient computing: micro-controllers
  - RP2040 (Cortex M0): 100 mW / core @133MHz; Fast sleep mode at 0.4 mW
  - Speed comparable to Pentium II (1997 – $\approx$ 50W) but 264kB RAM, 2MB flash
- Energy-efficient screen
  - OLED: 3 mW/cm^2 (black) to 20 mW/cm^2 (bright white)
  - eInk is bi-stable, but inefficient updates (10 mW/cm^2 at 2 Hz)
  - Memory LCDs: no refresh $\rightsquigarrow$ 2 μW/cm^2 (monochrome, fast)
- Energy-efficient cellular network
  - LTE Cat M1: 200 mW (TX 10kbps) / 5G: 3000 mW (TX 100Mbps)

On-board computation offloading

- Tiny-small design: most operations on a RP2040 microcontroler;
  - Offload heavy computations to a Raspberry Pi Zero on board (bare metal)
  - Pass full control to Linux on Pi Zero for legacy application



- Multikernel: Harness compute power; offload TCP, filesystem to other chips
- Compile-time verif with Rust to alleviate lack of MMU & runtime safety

Online infrastructure

Remote rendering

- HTML5 cannot be rendered on 2040
- Render in the cloud before download, to not start the PiZ
- Do not offload anything to the cloud (extra work hardly efficient)

Online point of presence

- Turn off data plan aggressively
- Online proxy sends text messages when a message arrives
- Maybe useless with LTE M1 hardware?

Junkyard computing

- Reuse existing hardware (e.g. your old phone)
- Reduce carbon impact
- Data self-hosting improves privacy

Designing a smartphone with energy limits

Low-power mobile device

- Memory LCD + keyboard: 0.5mW monochrome (from 800mW OLED)
- LTE M1 cellular: 200mW @10kbps (from 1200mW 4G or 3000mW 5G)
- Processing: RP2040 100mW per busy core + 600mW Pi Zero (from 3000mW)

Device with smartphone-like features but lasting days on a charge

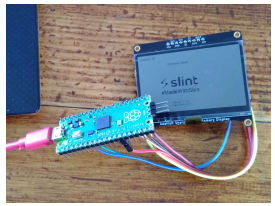
- Tiny-small design on board + multi-kernel to spread OS functions on chips
- Cloud-assisted: Rendering in smart proxy + online point of presence

Prospective applications

- Phone, Text messaging, DAV calendar, todo notes, podcasts: RP2040
- MyAndroidApp: Pi Zero with WayDroid; Passkey instead of banking app
- GPS navigation: Tile rendering on Pi Zero, navigation on RP2040
- Instant messaging: Matrix proxy server in cloud, interactions on RP2040
- HTML pages: Rendering in cloud, interactions on RP2040
- Redefining smartphones: features removed (video), but offline OSM / WP

SmolPhone current state

Prototyping and exploration since maybe one year



Ongoing internship: on-board offloading

- GPS navigation as a target application (with mepo)
- Goal: first-hand metering of the different modes' consumption + SW prototype

Proposed internship: SmolNet

- Cloud rendering, deported asynch graphical interface and simplified HTML
- No student yet, so we are slowly exploring by ourselves

Inria Action Exploratoire (accepted on 02/14)

- HW engineer for 2 years: Puzzle prototype + A5 devboard + better form factor
- Embedded software engineer for 2 years:
  - Base software (phone, Text messaging, DAV calendar, todo notes, podcasts)
  - Online infrastructure toward self-hosting and junkyard computing

Conclusion

Low-techs as an appealing future

- Resource efficient, accessible by novices, participative, non-superfluous needs
- Social utility, low complexity, long maintenance, essential needs, accessible
- Accessible, useful and durable.

Special challenges to low-tech computing

- Rebound effect, expert dictatorship, technosystem as a ruinous ruin

SmolPhone vision & research programs

- Low power hardware limiting the applications by design
- Architecture ideas: Tiny-small and onboard offloading, multi-kernel
- Cloud ideas: smart proxy & online presence w/ junkyard computing
- Other HW ideas: intermitent computing, energy harvesting
- Other empowering ideas: computing in the small (WP), simpler systems
- Other design ideas: foster local interactions and participative communities
- (insert your dream here)