SmolPhone

a smartphone with energy limits

Joseph Paturel, Martin Quinson, Simon Rokicki
Univ. Rennes, IRISA, Inria, France.

Beg Meil, October 11. 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 smartphones 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 smartphones 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 would it mean?

What are the low-techs?

Definition by ADEME

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

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.

Definition by Stéphane Crozat

- environnemental sustainability, social responsibility and technical conviviality

Some initiatives toward practical applications

- Some fablabs, Low-tech lab, L'atelier paysan, etc (but none in ICT).

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 tolerant)?
- 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 tolerant)?
- 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 tolerant)?
- 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 tolerant)?
- 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
- Smartphone with increased battery life at the cost of a reduced set of features
  - Reconsider classical design choices

The SmolPhone project

Research-action in post-growth computing

- Practical goal: low-tech smartphone with a one-week battery life
  - Not optimizing but reconsidering design choices
- Long term (unrealistic) goals: lasting 10 years; hackable by non-specialists
- Non-goals:
  - Cheaper device
  - Business plan on selling devices or services
  - Nostalgia or retrocomputing: need GPS, WhatsApp/Signal, web access, etc.
- Inspirations:
  - UXN: tiny but convivial VM (64kb of working memory but lovely assembly)
  - gemini: debloated HTML (web of hypertexts without inline links or images)
  - oulipo: writing movement using formal constraints to boost the creativity

Going under 100mW on average: hardware side

- Battery on FairPhone5 or iPhone 15: $\approx$ 4200mAh = 16.25Wh = **97mW·week**

Going under 100mW on average: hardware side

- Battery on FairPhone5 or iPhone 15: $\approx$ 4200mAh = 16.25Wh = **97mW·week**

Typical smartphone consumption (Galaxy S3 – 2017)

- CPU: 80mW idle / 3000 mW full
- Screen: OLED 800 mW — 3 mW/cm² (black) to 20 mW/cm² (bright white)
- Cellular 4G: 600 mW idle / 1200 mW TX
- Wifi: 80 mW idle / 120 mW TX

Going under 100mW on average: hardware side

- Battery on FairPhone5 or iPhone 15: $\approx$ 4200mAh = 16.25Wh = **97mW·week**

Typical smartphone consumption (Galaxy S3 – 2017)

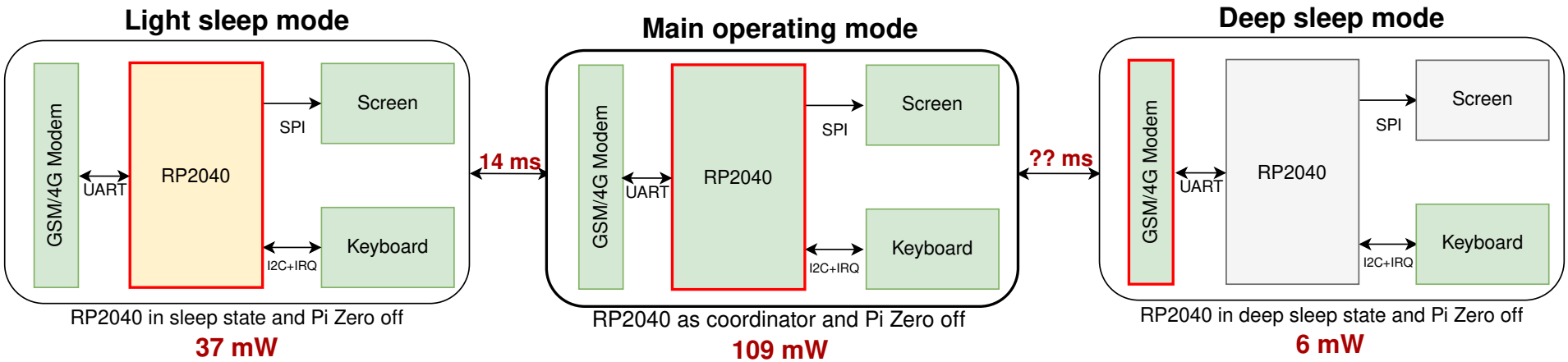
- CPU: 80mW idle / 3000 mW full
- Screen: OLED 800 mW — 3 mW/cm² (black) to 20 mW/cm² (bright white)
- Cellular 4G: 600 mW idle / 1200 mW TX
- Wifi: 80 mW idle / 120 mW TX

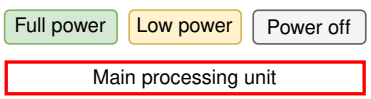
Smolphone envisioned hardware

- Energy-efficient computing: micro-controllers (RP2040: 100mW peak)
  - Speed comparable to Pentium II (1997 – $\approx$ 50W) but 264kB RAM, 2MB flash
- Energy-efficient screen
  - eInk is bi-stable, but inefficient updates (10 mW/cm² at 2 Hz)
  - Memory LCDs: no refresh $\rightsquigarrow$ 2 μW/cm² (monochrome, fast)
- Energy-efficient cellular network
  - 4G LTE Cat 1: 1µW idle, 250 mW TX (10kbps)
  - 5G: <1µW idle, 3000 mW TX (100Mbps)

Building a smartphone on that hardware

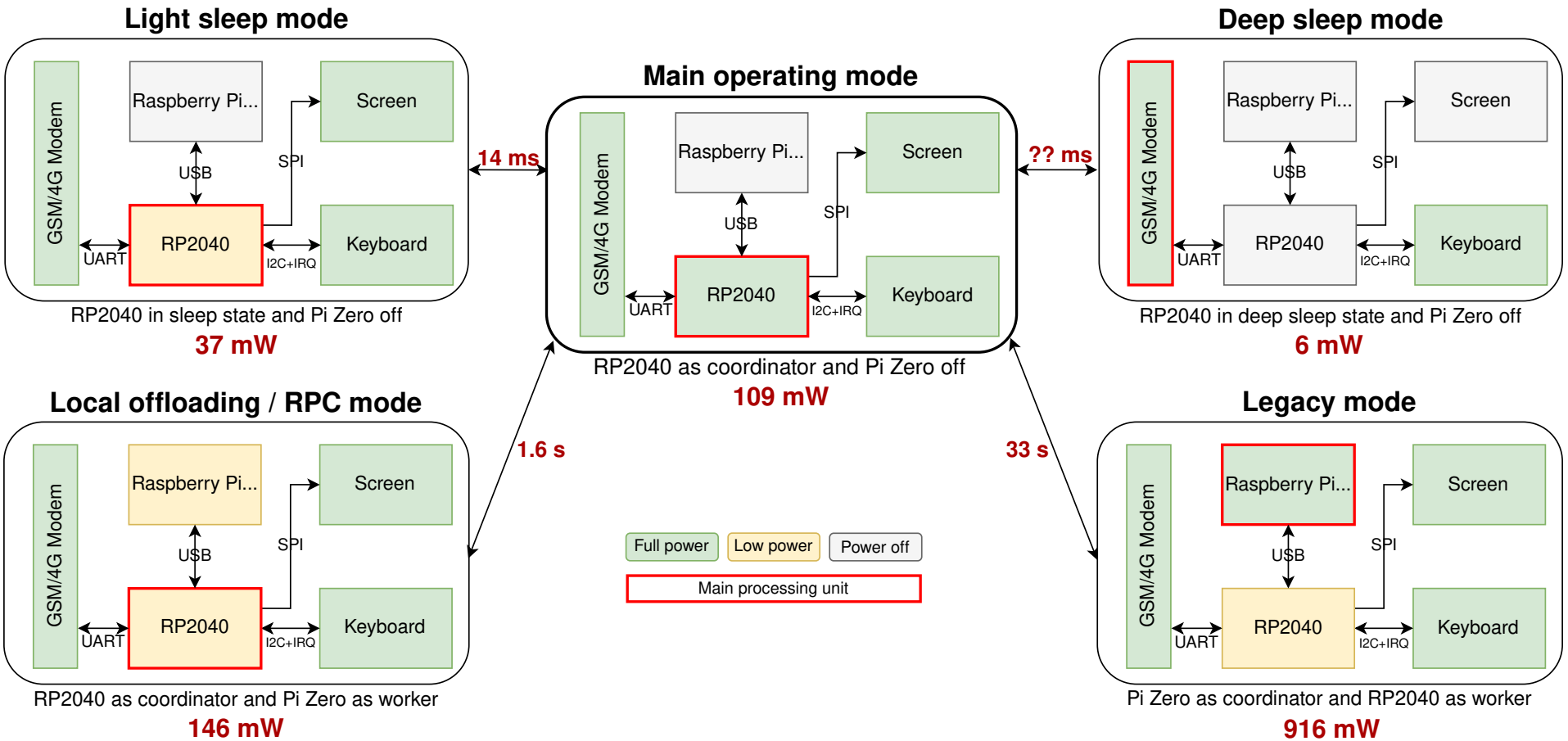
- Run on a RP2040 microcontroler
- Light sleep mode dozen times a second; deep sleep whenever possible





Building a smartphone on that hardware

- Run most operations on a RP2040 microcontroler
- Light sleep mode dozen times a second; deep sleep whenever possible



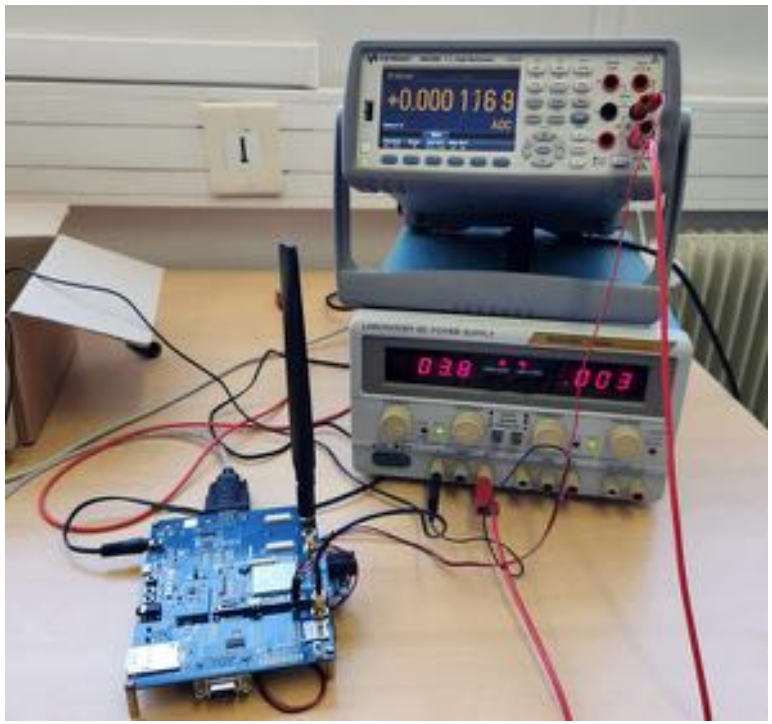
- Tiny-small design: add a Pi Zero for heavy computations
- Offload simple computations to PiZ bare metal (GPS tile)
- Pass full control to Linux on Pi Zero for legacy application

Quectel E912U-GL

- LTE Cat 1 are low-power modems intended for IoT
- Power Saving Mode: power off; back on RX or interrupt. $\approx$ 1µW
- Discontinuous reception (eDRX): off for 60s when no data is expected. $\approx$ 1µW

Preliminary measurements

- Data TX or RX: 250mW
- Voice call: 500mW
- Idle: 100mW
- Sleep: 40mW
- PSM: 1µW ?
- Text message: 0.1mWh



- More work needed to characterize this device, and explore others

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 Signal 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

Software stack

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

Smol is beautiful

- Applications should be scripted for conviviality (UXN targets 64kb of RAM)
- Aggressively prioritize simplicity over features (inspired by DuskOS)

Redefining smartphones

- Some features are removed: video, IA and neural networks
- Some features are added: offline OSM and wikipedia, easy extensions
- What can I remove from your smartphone before you stop using it?
- What crazy application you'd want?

Designing a smartphone with energy limits

Low-power mobile device

- Memory LCD + keyboard: 0.5mW monochrome (from 800mW OLED)
- LTE M1 cellular: 250mW @10kbps (from 1200mW 4G or 3000mW 5G)
- Processing: RP2040 100mW per busy core + 900mW Pi Zero (from 3000mW)

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

- Tiny-small design on board
- Cloud-assisted: Rendering in smart proxy + online point of presence

Other crazy ideas

- Multikernel: Harness compute power; offload TCP, filesystem to other chips
- Noisy algorithms: long-lasting device through soft robust to transient faults
- Data-over-Voice-over-GSM, intermitent computing, energy harvesting

Why would I want such phone?

Designing a smartphone with energy limits

Low-power mobile device

- Memory LCD + keyboard: 0.5mW monochrome (from 800mW OLED)
- LTE M1 cellular: 250mW @10kbps (from 1200mW 4G or 3000mW 5G)
- Processing: RP2040 100mW per busy core + 900mW Pi Zero (from 3000mW)

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

- Tiny-small design on board
- Cloud-assisted: Rendering in smart proxy + online point of presence

Other crazy ideas

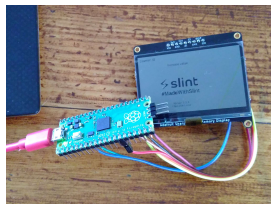
- Multikernel: Harness compute power; offload TCP, filesystem to other chips
- Noisy algorithms: long-lasting device through soft robust to transient faults
- Data-over-Voice-over-GSM, intermitent computing, energy harvesting

Why would I want such phone?

- Trade power (let the world obey) for might (do things by yourself) [Damasio]
- Really yours to fiddle with: opening it won't void the warranty

SmolPhone current state

Prototyping and exploration since maybe one year



Past internships

- Aloïs Rautureau: On-board offloading (metering the modes' consumption)
- Israel Kafando: Metering the 4G modem, and modem workbench

Ongoing internship: SmolNet

- Aurel Hamon: Cloud rendering, deported asynch GUI and simplified HTML

Future work: Inria Action Exploratoire

- HW engineer for 2 years: Puzzle prototype + A5 devboard + better form factor
- Victorien Elvinger: software engineer for 2 years
  - Scriptable convivial framework (between uxn and DuskOS)
  - Base software (phone, text messaging, DAV calendar, todo notes, podcasts)
  - Online infrastructure toward self-hosting and junkyard computing

My research interests

My research interests

2005-2015: Performance in HPC and Cloud infrastructures

- One of the main authors of the SimGrid framework
- Accurate modeling of the performance of distributed systems
- Timings and energy; Unmodified MPI apps or C/C++/Python prototypes

My research interests

2005-2015: Performance in HPC and Cloud infrastructures

- One of the main authors of the SimGrid framework
- Accurate modeling of the performance of distributed systems
- Timings and energy; Unmodified MPI apps or C/C++/Python prototypes

2015-2025: Correction of asynchronous parallel program

- Integrated a Software Model Checker in SimGrid, verifying unmodified apps
- SotA reduction algorithms, partially copes with multithreaded apps

My research interests

2005-2015: Performance in HPC and Cloud infrastructures

- One of the main authors of the SimGrid framework
- Accurate modeling of the performance of distributed systems
- Timings and energy; Unmodified MPI apps or C/C++/Python prototypes

2015-2025: Correction of asynchronous parallel program

- Integrated a Software Model Checker in SimGrid, verifying unmodified apps
- SotA reduction algorithms, partially copes with multithreaded apps

2025-. . . : Post-growth distributed infrastructures

- We need less bloat, in a fluctuating world.
  - Simplicity over modularity; Robustness over performance and control
- Not forking off my research agenda, just another perspective on same objects
- Since tech is not neutral, what is it that you want to foster?

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

The SmolPhone project

- Low power hardware limiting the applications by design
- Constrains unleash creativity, toward many original research projects

We are scientists

- We are not going to change the world, but we can at least work towards it