Inria focuses on computer science & control, including, but not limited to:

- Modeling, simulation, optimization of complex dynamic systems
- Security & reliability of computing systems
- Communication & ubiquitous computing (including IoT & CPS)
- Computational sciences
- Theoretical & empirical research approaches
End-to-End Open Source IoT with RIOT

www.riot-os.org

Emmanuel Baccelli & Alexandre Abadie

on behalf of the RIOT Community
Agenda

• Which IoT are we talking about?
• Why an OS for Low-End IoT devices?
• How?
• What is RIOT?
Many Expected IoT Applications

- Building Automation
- Connected Vehicles
- Smart Homes
- Wearable technology
- eHealth
- Industrial Automation

The cloud connects various devices and systems, enabling smart homes, wearable technology, connected vehicles, building automation, and eHealth applications.
Zoom: IoT Hardware
The Internet of Things (IoT)
High-end IoT Hardware

- single-board computers
  - RaspberryPi, connected cars, smartphones...
- resources similar to average Internet devices
  - memory, computation power, network throughput...

→ Can run usual TCP/IP protocols
→ Can run usual OS such as Linux
Low-end IoT Hardware

- Smaller & cheaper smart objects
- **Low-power**
  - MCUs & radios

Energy: Milliwatt instead of Watt
CPU: Megahertz instead of Gigahertz
Memory: Kilobytes instead of Gigabytes

→ Cannot run TCP/IP protocols as is
→ Cannot run usual OS such as Linux
Zoom: the Internet

• How can the Internet work at such large scale?
• How does it allow quick-paced progress?

organic growth, made possible by combining:

1. Open standard protocol specifications
2. Open source development platforms
3. Extremely flexible algorithms
Agenda

• Which IoT are we talking about?

• **Why an OS for Low-End IoT devices?**

• How?

• What is RIOT?
Why a software platform for Low-end IoT devices?

- Linux, Android... bare-metal?

- But as IoT software evolves...
  - more complex pieces, e.g. an IP network stack
  - evolution of application logic

- ...non-portable IoT software slows innovation

  90% of IoT soft. should be hardware-independent

  → this is achievable with a good software platform
  (but not if you develop bare-metal)
Why Open Source?

- **Faster innovation** by spreading IoT software dev. costs
- **Long-term IoT software robustness & security**
- **Trust, transparency & protection of IoT users’ privacy**
- **Less garbage** with less IoT device lock-down
Agenda

• Which IoT are we talking about?
• Why an OS for Low-End IoT devices?
• How?
• What is RIOT?
How to achieve a good software platform?

- Experience (e.g. with Linux) points towards:
  - open source
  - free core
  - driven by a grassroots community

- But technically, departure from Linux is needed
Main Challenges of an OS for IoT

Subj. to low-end IoT device resource constraints:

- Kernel performance
- System-level interoperability
- Network-level interoperability
- Trust
Software platform on low-end IoT devices?

• The good news:
  – no need for advanced GUI (simple shell is enough!)
  – no need for high throughput performance (kbit/s)
  – no need to support dozens of concurrent applications

• The bad news:
  – kBytes of memory!
  – typically no MMU!
  – extreme energy efficiency must be built-in!
Software platform on low-end IoT devices

- Contiki
- RIOT
- TinyOS
- mbedOS (ARM)
- Zephyr (Intel)
- LiteOS (Huawei)
- ...
- ... and closed-source alternatives

Agenda

• Which IoT are we talking about?
• Why an OS for Low-End IoT devices?
• How?

• What is RIOT?
RIOT: an OS that fits IoT devices

Internet

Memory ~ 16kB

Memory ~ 8kB

Memory ~ 100kB
RIOT: an OS that fits IoT devices

- RIOT is the combination of:
  - needed memory & energy efficiency to fit IoT devices
  - functionalities of a full-fledged operating system
    - Advanced, consistent APIs across 32-bit, 16-bit, 8-bit hardware
    - Full-featured, extensible network stacks
    - Large pool of 3rd party software libraries
Agenda

• Which IoT are we talking about?
• Why an OS for Low-End IoT devices?
• How?
• What is RIOT?

  - Aspect 1: kernel performance
  - Aspect 2: system-level interoperability
  - Aspect 3: network-level interoperability
  - Aspect 4: trust
Challenge: Constrained Devices

ENERGY
- Milliwatt instead of Watt

CPU
- Megahertz instead of Gigahertz

Memory
- Kilobytes instead of Gigabytes
RIOT Kernel Performance

• **Micro-kernel** architecture (contrary to Linux)
  - Ultra-low memory footprint
  - Tickless scheduler → energy efficiency
  - Deterministic O(1) scheduler → real-time
  - Low latency interrupt handler → reactivity

Result (on ARM 32-bit)
- around 1kB of RAM
- less on 16-bit

Fits on low-end IoT devices
RIOT Kernel Performance

• Typical approach: event-driven, single stack

• RIOT approach: **multi-threading**
  - minimal Thread Control Block (TCB)
  - minimized stack usage
  - Ultra-efficient Inter-Process Communication (IPC)
  - (Note: multithreading is optional)

Result (on 16-bit at 8MHz)
- Min. TCB: 8 bytes
- Min. Stack Size: 96 bytes
- Up to 16,000 Messages/s with IPC
  (-= 10,000 Packets/s for 802.15.4)

Fits on low-end IoT devices
Agenda

• Which IoT are we talking about?
• Why an OS for Low-End IoT devices?
• How?
• What is RIOT?
  • Aspect 1: kernel performance
  • Aspect 2: system-level interoperability
  • Aspect 3: network-level interoperability
  • Aspect 4: trust
Challenge: Interoperability

- System-level interoperability
  - Hardware-independent IoT software
  - Usability of third-party software, well-known tools
RIOT System-level Interoperability

• Typical approach: specialized programming
  Exotic Paradigms & Programming Languages

• RIOT approach: standard programming
  ANSI C, no macro ‘magic’, standard multi-threading

Result: Low Porting Effort
  ✓ Emulation Support: RIOT as a Process
  ✓ Third-Party Library Package
  ✓ Third-Party Development Tools

<table>
<thead>
<tr>
<th>Package</th>
<th>Diff Size</th>
<th>Overall</th>
<th>Relative</th>
</tr>
</thead>
<tbody>
<tr>
<td>libcoap</td>
<td>639 lines</td>
<td></td>
<td>6.3 %</td>
</tr>
<tr>
<td>libfixmath</td>
<td>34 lines</td>
<td></td>
<td>0.2 %</td>
</tr>
<tr>
<td>lwip</td>
<td>767 lines</td>
<td></td>
<td>1.3 %</td>
</tr>
<tr>
<td>micro-ecc</td>
<td>14 lines</td>
<td></td>
<td>0.8 %</td>
</tr>
<tr>
<td>relic</td>
<td>24 lines</td>
<td></td>
<td>&lt;0.1 %</td>
</tr>
</tbody>
</table>
RIOT : system-level Interoperability

- Consistent, powerful API on 8-bit, 16-bit, 32-bit – 60+ IoT boards/devices supported, various radios...

With a simple application over a IPv6/6LoWPAN stack in RIOT, 95% of the code is hardware-independent and/or reusable.
Agenda

• Which IoT are we talking about?
• Why an OS for Low-End IoT devices?
• How?
• What is RIOT?
  • Aspect 1: kernel performance
  • Aspect 2: system-level interoperability
  • Aspect 3: network-level interoperability
  • Aspect 4: trust
Challenge: Interoperability

- **System-level interoperability**
  - Hardware-independent IoT software
  - Usability of third-party software, well-known tools

- **Network-level interoperability**
  - End-to-end connectivity per default
  - Device-to-device connectivity
Network-level Challenge: The IoT today looks mostly like this.
Network-level Challenge:
The IoT we want looks more like that.
The IoT we want is... the Internet!
Internet building blocks: Challenged by IoT...

... because of resource constraints on IoT devices
  - Memory, CPU, energy

... because of low-power communication characteristics
  - Lossy / duty cycles
  - Super-small frames
  - Spontaneous wireless architecture

→ Adapted standard IoT protocols needed
Standard IoT protocols? On the way!
Work in progress at IETF, IEEE, W3C, OMA...

New specs for link layer technologies
- Low-power radios, PLC, BACnet
- IEEE 802.15.4, Z-Wave, BLE, LoRa (and IEEE 802.11)
- More to come...

New specs for network layer protocols
- Fitting IoT requirements and interoperable with IP
- 6TiSCH, 6LoWPAN, RPL, OLSRv2, AODVv2
- More to come...

New specs for application layer protocols
- Fitting IoT requirements and interoperable with web
- CoAP, LwM2M, MQTT-SN, CBOR
- Security with DTLS, OSCOAP
- More to come...

New network paradigms
- Content-centric networking for IoT
- More to come...
Network Stacks Available with RIOT

- **default (6LoWPAN) stack**
  - GNRC: in-house stack

- **3rd-party packages**
  - lwIP stack
  - uIP (emb6) stack
  - Thread (OpenThread) stack
  - ...

- **experimental stacks**
  - CCN-lite
  - NDN-RIOT
Agenda

• Which IoT are we talking about?
• Why an OS for Low-End IoT devices?
• How?
• What is RIOT?
  • Aspect 1: kernel performance
  • Aspect 2: system-level interoperability
  • Aspect 3: network-level interoperability
  • Aspect 4: trust
Some level of trust with IoT?

Combining RIOT & Linux, IoT is possible with

- End-to-end open source
- End-to-end secure & open communication standards
- From anywhere in the Internet all the way to (low-end) IoT devices
Some level of trust with IoT?

- Started in 2013 as a research project
  - “soft” copyleft license (LGPLv2.1)
  - Grassroots open source community

- 130+ code contributors world-wide
  - Makers, academics, SME, bigger industrial
  - RIOT Foundation (for-common-good organization)
  - RIOT Summit: gathering community yearly
RIOT in a nutshell

Free, open-source platform for portable IoT software

RIOT offers a platform functionally equivalent to Linux, based on:

- open-source,
- open-access protocol specs,
- community-driven dev.

<table>
<thead>
<tr>
<th>Closed- &amp; Open-source IoT Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Other network stacks</td>
</tr>
<tr>
<td>CoAP, LWM2M, MQTT-SN, MQTT, AMQP</td>
</tr>
<tr>
<td>RPL, UDP, RPL, IPv6, 6LoWPAN</td>
</tr>
</tbody>
</table>

Micro Kernel

Open-Source Drivers

Closed-Source Drivers

Peripherals (including network interfaces)

Lowpower MCU + radio or wired communications

Third-party software

Hardware
RIOT in Action

http://riot-demo.inria.fr

- COTS low-end IoT hardware
- Web server dashboard
- COAP/IPv6/6LoWPAN
- End-to-end open source
- End-to-end open communication standards
RIOT in Action

RIOT supported on open access testbeds: https://www.iot-lab.info
Closing words

- We’re seeing a lot of activity on OS for low-end IoT

- We’re seeing huge progress in IoT on:
  - connectivity
  - interoperability

- Main IoT challenges ahead:
  - Large scale orchestration/management
  - Security (avoid IoT botnets armageddon?)
  - Privacy (reconciled with IoT and big data?)
Thanks!

News: https://twitter.com/RIOT_OS
For cooperation questions: riot@riot-os.org
For developer questions: devel@riot-os.org
Contributing: https://github.com/RIOT-OS/RIOT
Support & discussions on IRC: irc.freenode.org #riot-os