Introduction

What is RIOT

- **operating system** for microcontrollers
  - **microkernel architecture** ⇒ require very low resources
  - **real-time** and **multi-threaded**
  - comes with **in-house networking stacks**

What is RIOT

- **operating system** for microcontrollers
  - **microkernel architecture** ⇒ require very low resources
  - **real-time** and **multi-threaded**
  - comes with **in-house networking stacks**

- **open-source**: https://github.com/RIOT-OS/RIOT
  - free software platform
  - **world-wide community** of developers

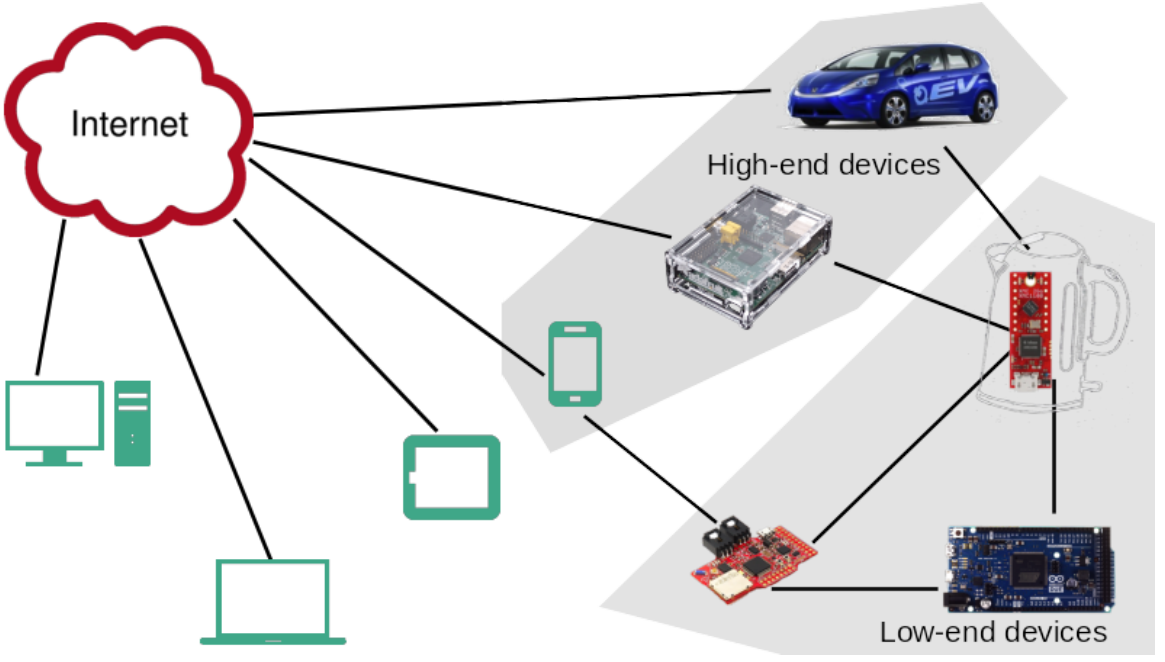
What is RIOT

- **operating system** for microcontrollers
  - **microkernel architecture** ⇒ require very low resources
  - **real-time** and **multi-threaded**
  - comes with **in-house networking stacks**

- **open-source**: https://github.com/RIOT-OS/RIOT
  - free software platform
  - **world-wide community** of developers

- **easy to use** and **reuse**
  - Standard programming in C
  - Standard tooling
  - **API is independent** from the hardware

RIOT in the IoT world



⇒ RIOT is designed for low-end devices
(kB RAM, MHz, mW)

History of the project

- 2013: Inria, FU Berlin and HAW founded RIOT
  - stemed from a French-German research project
  - kernel evolved from FireKernel

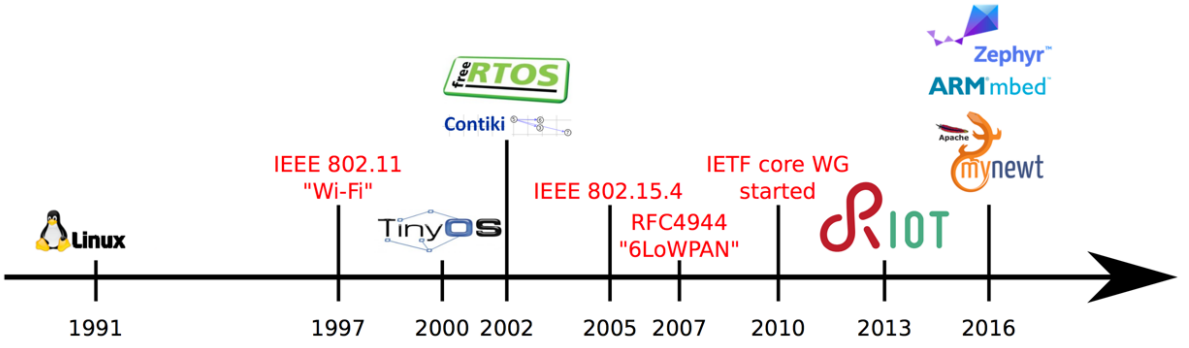
History of the project

- 2013: Inria, FU Berlin and HAW founded RIOT
  - stemed from a French-German research project
  - kernel evolved from FireKernel
- The community today:
  - So far, **+200** different contributors to the master branch
  - Academics: Berkeley, UCLA, MIT, AIT, TZI, etc
  - Industrial: Cisco, Samsung, ImgTec, Fujitsu, Thalès
  - SME: Zolertia, OTAKeys, Mesotic, Eistec, We-sens
  - Member of the EdgeXFoundry initiative

History of the project

- 2013: Inria, FU Berlin and HAW founded RIOT
  - stemed from a French-German research project
  - kernel evolved from FireKernel
- The community today:
  - So far, **+200** different contributors to the master branch
  - Academics: Berkeley, UCLA, MIT, AIT, TZI, etc
  - Industrial: Cisco, Samsung, ImgTec, Fujitsu, Thalès
  - SME: Zolertia, OTAKeys, Mesotic, Eistec, We-sens
  - Member of the EdgeXFoundry initiative
- Annual RIOT Summit: https://summit.riot-os.org

Competitors

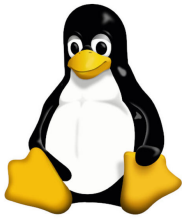


Reference: O. Hahm et al. "Operating Systems for Low-End Devices in the Internet of Things: A survey," IEEE Internet of ThingsJournal, 2016.

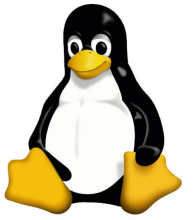
- requires less memory & adapts to a wider range of architectures
- fosters an open-source philosophy more akin to Linux
- provides more integrated high-level functionalities

The RIOT philosophy & community

- RIOT is free-software, licensed under **LGPLv2.1**
- The community takes inspiration from Linux

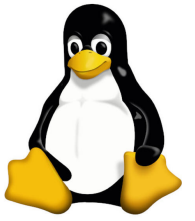


The RIOT philosophy & community



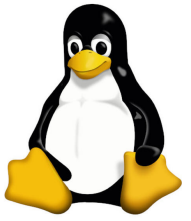
- RIOT is free-software, licensed under **LGPLv2.1**
- The community takes inspiration from Linux
- Use standards whenever possible
  (C-ANSI, standard tools, standard protocols, standard procedures)
- Follow **POSIX** standards

The RIOT philosophy & community



- RIOT is free-software, licensed under **LGPLv2.1**
- The community takes inspiration from Linux
- Use standards whenever possible
  (C-ANSI, standard tools, standard protocols, standard procedures)
- Follow **POSIX** standards
- Avoid code duplication, easy to program, increase **portability**, modularity
- Vendor & Technology **independence**

The RIOT philosophy & community



- RIOT is free-software, licensed under **LGPLv2.1**
- The community takes inspiration from Linux
- Use standards whenever possible
  (C-ANSI, standard tools, standard protocols, standard procedures)
- Follow **POSIX** standards
- Avoid code duplication, easy to program, increase **portability**, modularity
- Vendor & Technology **independence**
- Decisions and orientations are taken by a **grass-root community**

Ecosystem & community processes

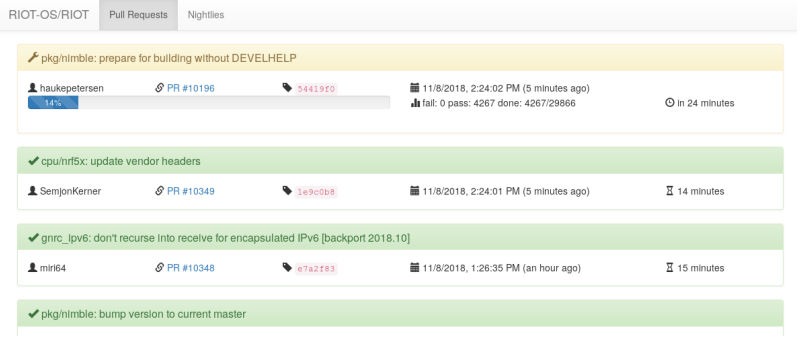
- Standard tooling and build system: **Makefiles**, **OpenOCD**, **GDB**

Ecosystem & community processes

- Standard tooling and build system: **Makefiles**, **OpenOCD**, **GDB**
- Distributed and fast CI, Murdock: https://ci.riot-os.org

  ⇒ **Build and run** all test/example applications

  ⇒ **Static tests** (Cppcheck, Coccinelle, etc)

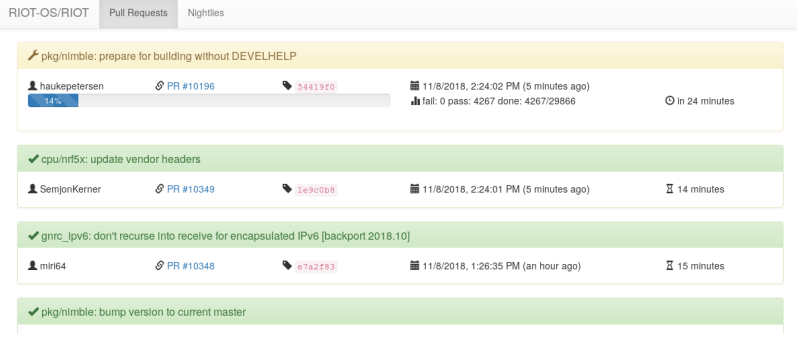


Ecosystem & community processes

- Standard tooling and build system: **Makefiles**, **OpenOCD**, **GDB**
- Distributed and fast CI, Murdock: https://ci.riot-os.org

  ⇒ **Build and run** all test/example applications

  ⇒ **Static tests** (Cppcheck, Coccinelle, etc)



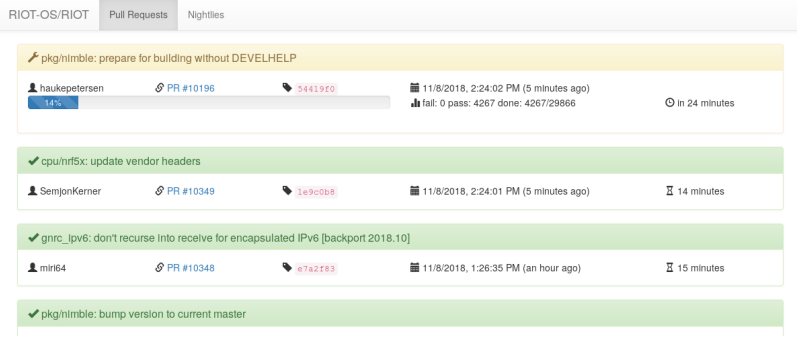
- **Online documentation** ⇒ https://doc.riot-os.org

Ecosystem & community processes

- Standard tooling and build system: **Makefiles**, **OpenOCD**, **GDB**
- Distributed and fast CI, Murdock: https://ci.riot-os.org

  ⇒ **Build and run** all test/example applications

  ⇒ **Static tests** (Cppcheck, Coccinelle, etc)



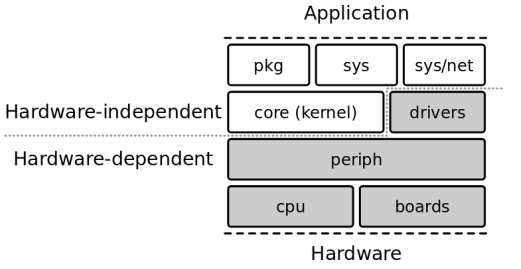
- **Online documentation** ⇒ https://doc.riot-os.org
- In-depth **code reviews**
- Stable **release every 3 months**: . (ex: 2018.07, 2018.10, etc)

Technical overview

Long story short: paper in IEEE Internet of Things Journal
Preprint available: http://riot-os.org/files/2018-IEEE-IoT-Journal-RIOT-Paper.pdf

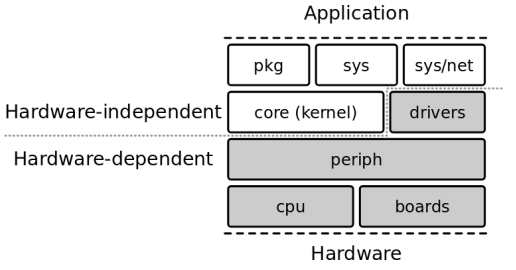
OS characteristics

- **Micro-kernel** based architecture: modular approach



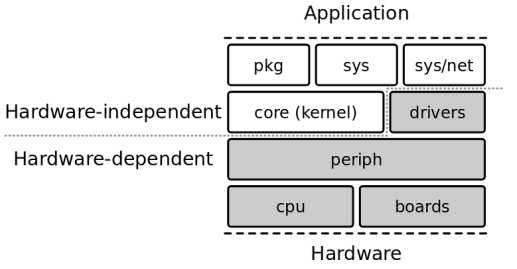
OS characteristics

- **Micro-kernel** based architecture: modular approach
- Small footprint
  ⇒ **2.8kB** RAM, **3.2kB** ROM on 32-bit Cortex-M

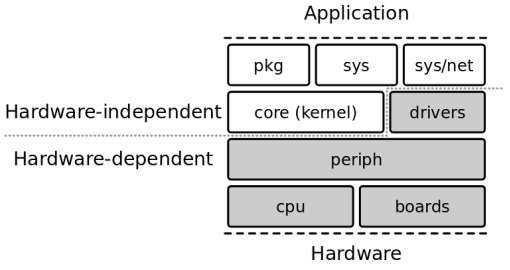


OS characteristics

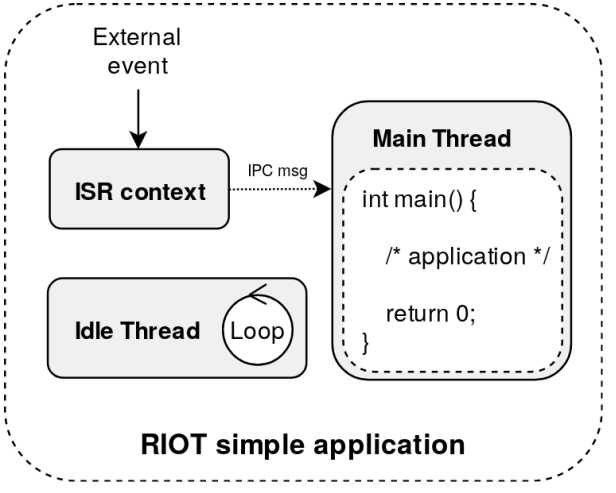
- **Micro-kernel** based architecture: modular approach
- Small footprint
  ⇒ **2.8kB** RAM, **3.2kB** ROM on 32-bit Cortex-M
- **Real-Time** scheduler
  - ⇒ fixed priorities preemption with O(1) operations
  - ⇒ tickless scheduler



OS characteristics



- **Micro-kernel** based architecture: modular approach
- Small footprint
  ⇒ **2.8kB** RAM, **3.2kB** ROM on 32-bit Cortex-M
- **Real-Time** scheduler
  - ⇒ fixed priorities preemption with O(1) operations
  - ⇒ tickless scheduler
- **Multi-Threading** and IPC:
  - Separate thread contexts with separate thread memory stack
  - Minimal thread control block (TCB)
  - Thread synchronization using mutexes, semaphores and messaging

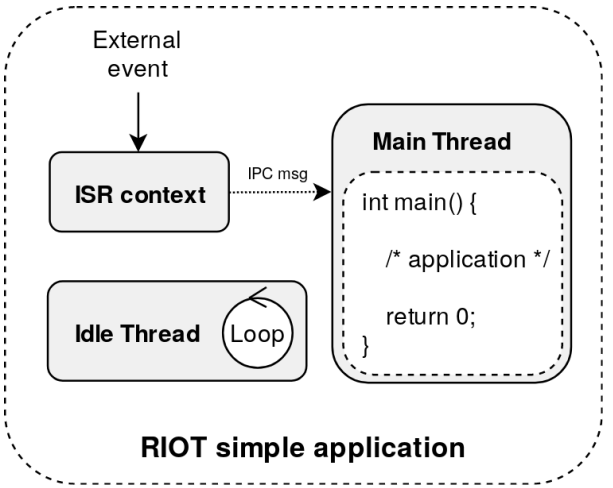


Multi-Threading

2 threads by default:

- the `main` thread: running the `main` function
- the `idle` thread:
  - lowest priority
    ⇒ fallback thread when all other threads are blocked or terminated
  - switches the system to low-power mode

The ISR context handles external events and notifies threads using IPC messages

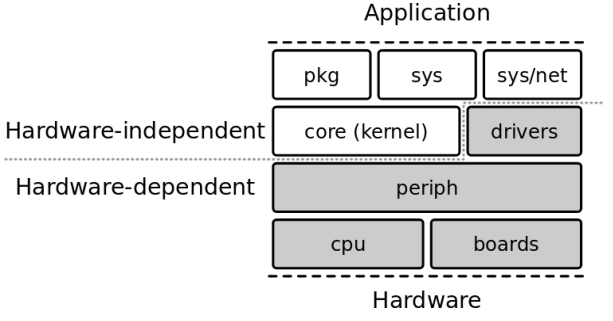


Hardware abstraction layer

- Divided in 3 blocks: boards, cpus, drivers
- CPUs are organized as follows:
  architecture (ARM) > **family** (stm32) > **type** (stm32l4) > **model** (stm32l476rg)
- Generic API for cpu peripherals (gpio, uart, spi, pwm, etc)

  ⇒ same API for all architectures
- Only based on vendor header files (CMSIS) ⇒ implementation from scratch

  ⇒ less code duplication, more efficient, more work
- One application ⇒ one board ⇒ one cpu model



Hardware support overview

- **Hardware abstraction layer:** support for 8/16/32 bit, ARM, AVR, MIPS
- Supported vendors: Microchip, NXP, STMicroelectronics, Nordic, TI, ESP, RISC-V, etc
- **Large list of sensors and actuators** supported (e.g drivers)
- *native* board: **run RIOT as process on your computer**
- **+100 boards supported**

NXP MICROCHIP NORDIC SEMICONDUCTOR TEXAS INSTRUMENTS

A modular OS

Features are provided as modules ⇒ **only build what's required**

- System libraries: **xtimer**, **shell**, crypto, etc

A modular OS

Features are provided as modules ⇒ **only build what's required**

- System libraries: **xtimer**, **shell**, crypto, etc
- Sensors and actuators
- Display drivers, filesystems, etc

A modular OS

Features are provided as modules ⇒ **only build what's required**

- System libraries: **xtimer**, **shell**, crypto, etc
- Sensors and actuators
- Display drivers, filesystems, etc
- Embedded interpretors: Javascript, Micropython, LUA

A modular OS

Features are provided as modules ⇒ **only build what's required**

- System libraries: **xtimer**, **shell**, crypto, etc
- Sensors and actuators
- Display drivers, filesystems, etc
- Embedded interpretors: Javascript, Micropython, LUA
- High-level network protocols: CoAP, MQTT-SN, etc

A modular OS

Features are provided as modules ⇒ **only build what's required**

- System libraries: **xtimer**, **shell**, crypto, etc
- Sensors and actuators
- Display drivers, filesystems, etc
- Embedded interpretors: Javascript, Micropython, LUA
- High-level network protocols: CoAP, MQTT-SN, etc
- External packages

Package	Overall Diff Size	Relative Diff Size
ccn-lite	517 lines	1.6 %
libfixmath	34 lines	0.2 %
lwip	767 lines	1.3 %
micro-ecc	14 lines	0.8 %
spiffs	284 lines	5.5 %
tweetnacl	33 lines	3.3 %
u8g2	421 lines	0.3 %

Useful system libraries

- **xtimer**
  - high-level timer subsystem that provides full abstraction from the hardware timer
  - Can set callbacks, put a thread to sleep, etc
- **shell**
  - provides interactive command line interface
  - useful for interactive debugging or examples
- **Others:** crypto, fmt, math, etc

External packages

- RIOT can be extended with external packages
- Integrated (and eventually patched) on-the-fly while building an application
- Easy to add: just requires 2 `Makefiles`
- Example of packages: lwIP, Openthread, lvgl, loramac, etc

Package	Overall Diff Size	Relative Diff Size
ccn-lite	517 lines	1.6 %
libfixmath	34 lines	0.2 %
lwip	767 lines	1.3 %
micro-ecc	14 lines	0.8 %
spiffs	284 lines	5.5 %
tweetnacl	33 lines	3.3 %
u8g2	421 lines	0.3 %

Network stacks

IP oriented stacks ⇒ designed for Ethernet, WiFi, 802.15.4 networks

- **GNRC**: the in-house 802.15.4/6LowPAN/IPv6 stack of RIOT

Network stacks

IP oriented stacks ⇒ designed for Ethernet, WiFi, 802.15.4 networks

- **GNRC**: the in-house 802.15.4/6LowPAN/IPv6 stack of RIOT
- **Thread**: 802.15.4 IPv6 stack provided by the ThreadGroup

Network stacks

IP oriented stacks ⇒ designed for Ethernet, WiFi, 802.15.4 networks

- **GNRC**: the in-house 802.15.4/6LowPAN/IPv6 stack of RIOT
- **Thread**: 802.15.4 IPv6 stack provided by the ThreadGroup

openthread
released by Nest

- **OpenWSN** (experimental): a deterministic MAC layer implementing the IEEE 802.15.4e TSCH protocol

Network stacks

IP oriented stacks ⇒ designed for Ethernet, WiFi, 802.15.4 networks

- **GNRC**: the in-house 802.15.4/6LowPAN/IPv6 stack of RIOT
- **Thread**: 802.15.4 IPv6 stack provided by the ThreadGroup

- **OpenWSN** (experimental): a deterministic MAC layer implementing the IEEE 802.15.4e TSCH protocol

- Other IPv6 stacks:
  - **lwIP**: full-featured network stack designed for low memory consumption
  - **emb6**: A fork of Contiki network stack that can be used without proto-threads

Other network support

- In-house Controller Area Network (**CAN**)

Other network support

- In-house Controller Area Network (**CAN**)
- **BLE** stack support: NimBLE

Other network support

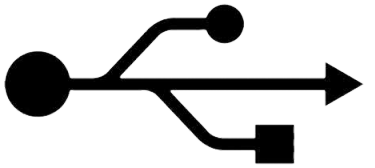
- In-house Controller Area Network (**CAN**)
- **BLE** stack support: NimBLE
- **LoRaWAN** stack ⇒ Compliant with LoRaWAN 1.0.2

Other network support

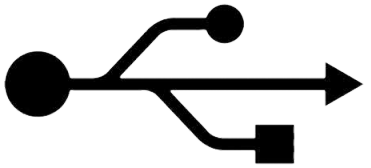
- In-house Controller Area Network (**CAN**)
- **BLE** stack support: NimBLE
- **LoRaWAN** stack ⇒ Compliant with LoRaWAN 1.0.2
- **SigFox** support for ATA8520e modules

Other important features

- Full featured USB stack (CDC-ACM, CDC-ECM, etc)



Other important features

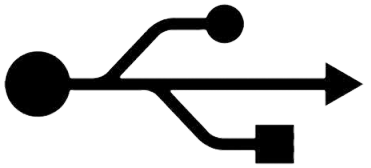


- Full featured USB stack (CDC-ACM, CDC-ECM, etc)

- Standard and secure software update implementation

https://datatracker.ietf.org/wg/suit/about/

Other important features



- Full featured USB stack (CDC-ACM, CDC-ECM, etc)

- Standard and secure software update implementation

https://datatracker.ietf.org/wg/suit/about/

Coming soon

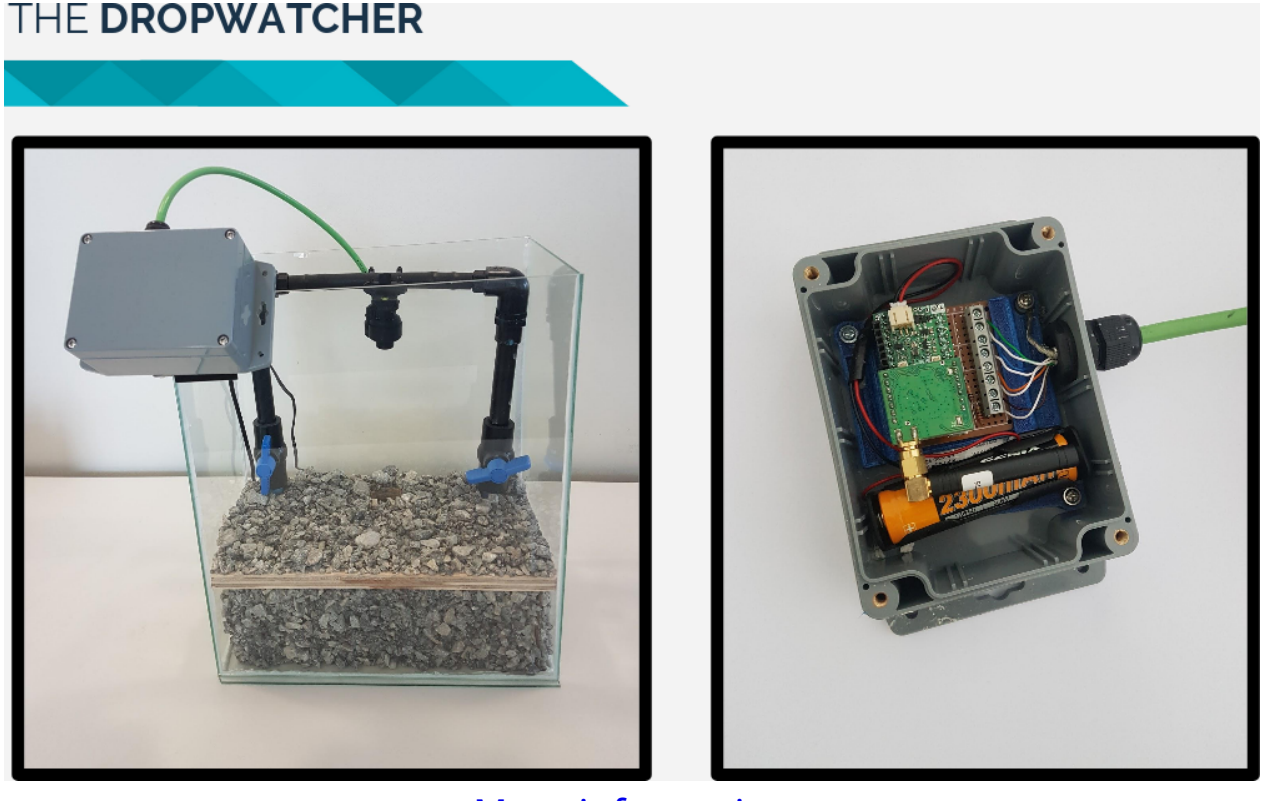
- NB-IoT support

- More advanced configuration with Kconfig

RIOT in action

IoT deployments using RIOT

Telefonica Chile: LoRa devices in a mine



More information

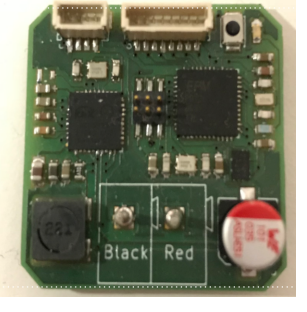
IoT deployments using RIOT

Algramo: Automatic cereal dispenser

More information

IoT deployments using RIOT

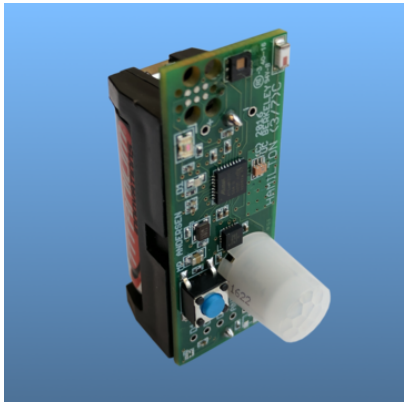
Home automation using KNX

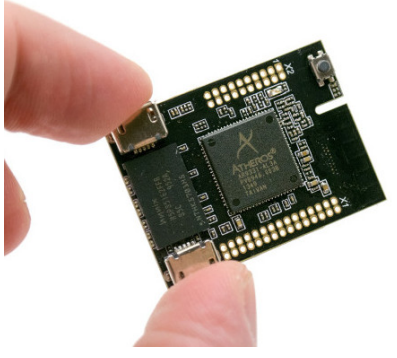


[More information]

IoT products & services using RIOT

- Environment monitoring: Hamilton IoT (USA), Unwired Devices (Russia)





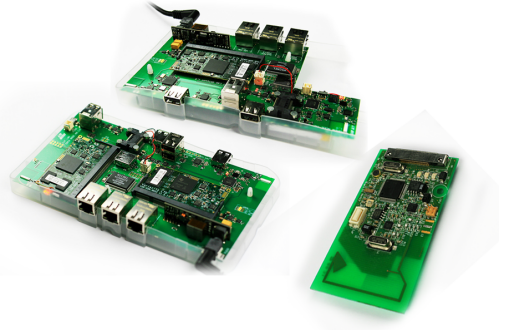
- On-Board diagnostics for connected cars: OTAKeys (Continental)
- Design Office: Eistec (Sweeden), Mesotic (France)

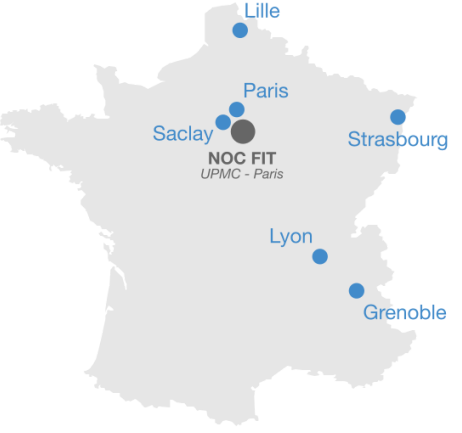
RIOT on FIT/IoT-LAB large scale testbed

https://www.iot-lab.info

IoT-LAB is a **large scale experimentation testbed**

- Can be used for **testing wireless communication** networks on **small devices**
- Can be used for **learning** IoT programming and **communication protocols**
- Can be used for testing software platforms on **heterogeneous hardware**





Learn RIOT

- RIOT Tutorials

  https://github.com/RIOT-OS/Tutorials

- RIOT online course

  https://github.com/riot-os/riot-course

Summary

- Generalities on RIOT: history, community, users
- A technical overview:
  - OS charactéristics
  - hardware support
  - libraries
  - network stack
  - import external libraries via packages
- The RIOT ecosystem: standard tools, CI, documentation
- Companies using RIOT

Back to the course