RIOT Hackathon: Power Management / Energy

Robert, Michel, Raphael

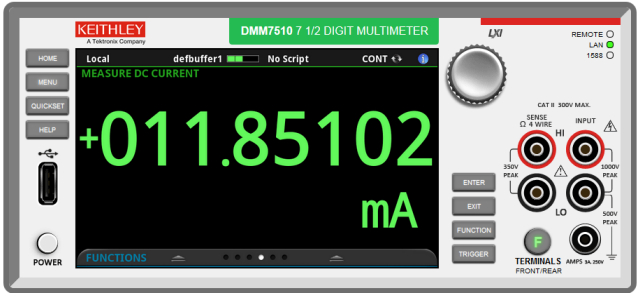
July 2017

STM32L4 – state

- RTC implemented (PR #7420[1])
  ⇒ Used by test application (wake-up source!)
- 4 Power modes identified that we will implement
- Next step: Use power mode
  ⇒ Requires clock recovery (stmclk module) **WIP**
- Implemented test application

[1] https://github.com/RIOT-OS/RIOT/pull/7420

STM32L4 (cont.)



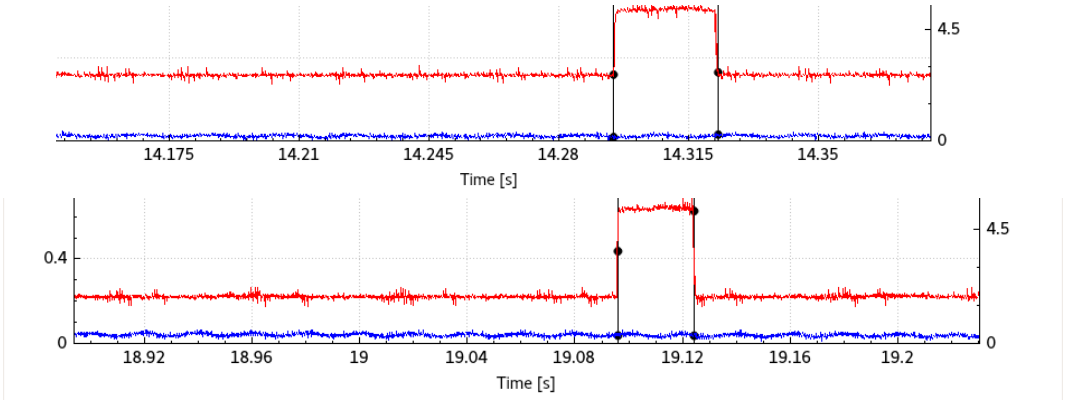
Running

Shutdown

atmega – state

- Finally Watchdog Interrupt is working (used as wakeup source)
- `atmega_common`
  - Using `pm_block`/`pm_unblock` in UART
- Implemented PM for INGA
  - `pm_set` in `cpu/atmega1284p` and use `pm_layered` module
  - Allowing 8 sleep modes, all implemented
    $\Rightarrow$ `pm_layered` changed, might be inefficient (warning is thrown for other MCUs)
    $\Rightarrow$ to be discussed
- Tests for
  - Wakeup from button or WDTI
  - Wakeup from `msg_send`

atmega (cont.)


4.5
0
14.175
14.21
14.245
14.28
14.315
14.35
Time [s]
4.5
0.4
0
18.92
18.96
19
19.04
19.08
19.12
19.16
19.2
Time [s]

Problems & Goals

Problems

- STM32L4
  - Clock tree complex
    $\Rightarrow$ Which clocks need to be recovered and how?
  - Changing between sleep modes
- atmega_common
  - Erroneous interfaces (no `uart_poweron` / `uart_poweroff`)

Goals

- `atmega_common`: Disable all peripherals by default
- General test application for both STM32L4 and ATmega
  $\Rightarrow$ Find use cases for measurements
- Rudimentary documentation in the RIOT Wiki