Eco

A Hardware-Software Co-design for In Situ Power Measurement on Low-end IoT Systems

7th International Workshop on Energy Harvesting & Energy-Neutral Sensing Systems (ENSsys 2019)
November 10, 2019 - New York

Michel Rottleuthner ✉ **michel.rottleuthner@haw-hamburg.de**
Thomas C. Schmidt ✉ t.schmidt@haw-hamburg.de
Matthias Wählisch ✉ m.waehlisch@fu-berlin.de

Agenda

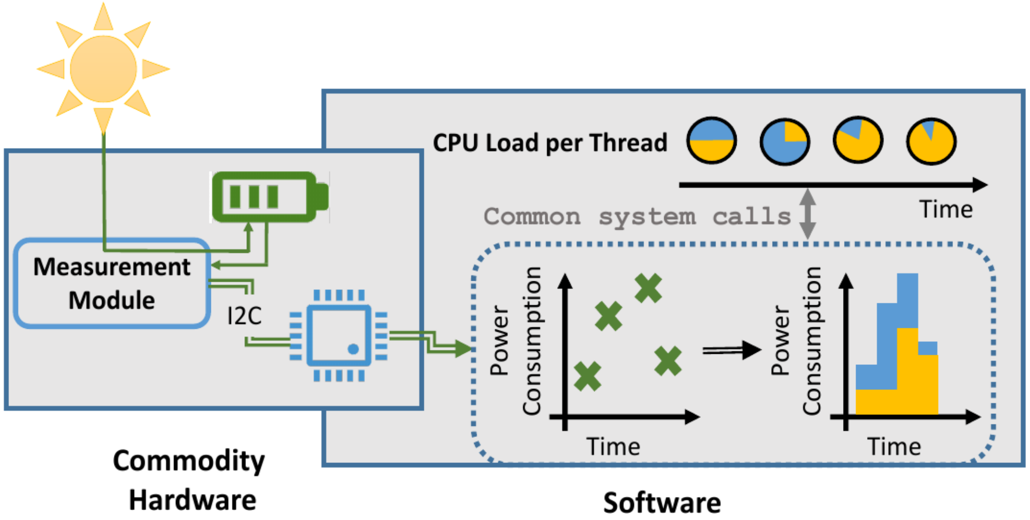
- Background on in situ power measurement
- Eco overview and software integration
- Evaluation of Eco
- Conclusion & outlook

Background

In situ power measurement & related work

Background

- Energy harvesting
- Energy management
  - Assessment
  - Attribution
  - Allocation
  - Prediction



Q1: “Can we do this with commodity components in a generic way?”

Q2: “How much overhead is introduced by that?”

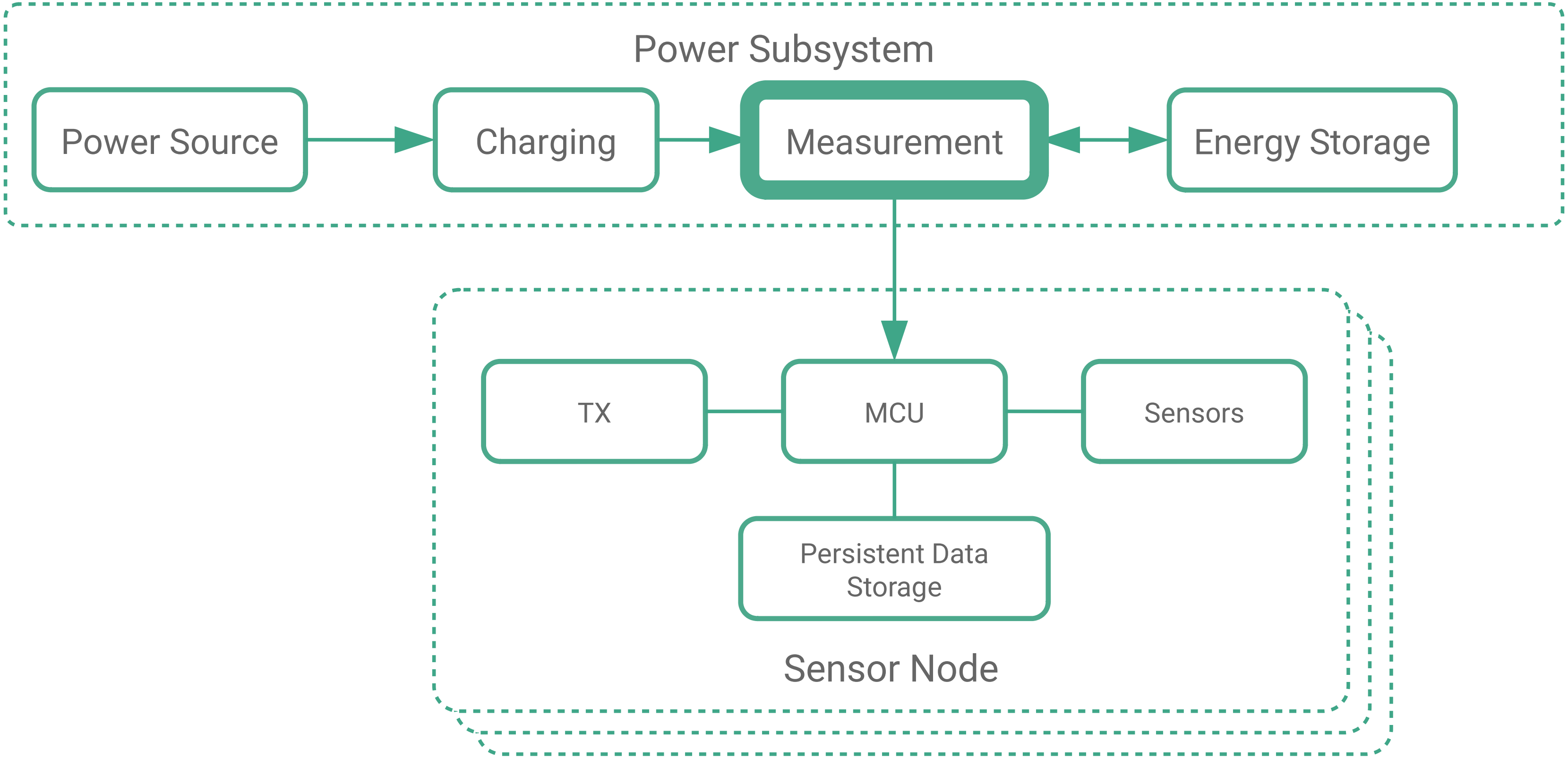
Background - Related Work

Solution	**Description**	CPU overhead investigation	Platform independent attribution	Bidirectional & voltage
SPOT [13]	custom design; shunt voltage to frequency	✗	✗	✗
iCount [9]	SPOT principle; frequency from inductor	✗	✗	✗
Nemo [24]	additional MCU; tailored for TelosB	✓	✗	✗
Eco	commodity components; generic software	✓	✓	✓

Eco

Architecture and Module Overview

Eco – System Overview



Power Subsystem

Power Source

Charging

Measurement

Energy Storage

TX

MCU

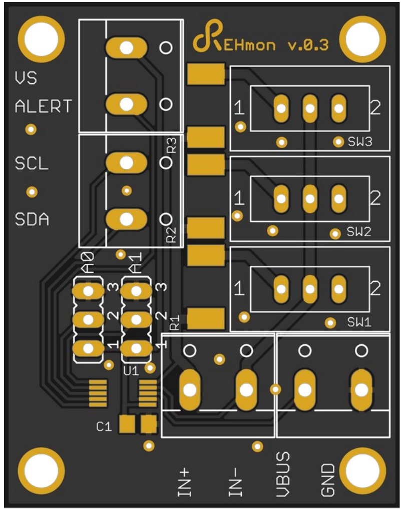
Sensors

Persistent Data Storage

Sensor Node

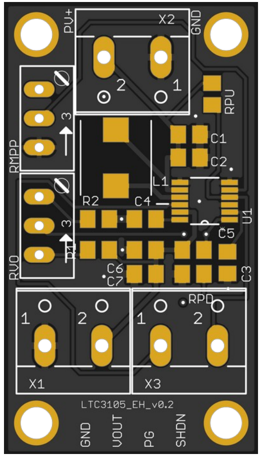
Eco – Measurement Module

- Bi-directional shunt monitor (INA226)
- Measures current and voltage
- I²C interface
- Current range selectable e.g. {40, 100, 500} mA
- Calibration
- Configurable conversion time and averaging
- Interrupt features for unattended operation



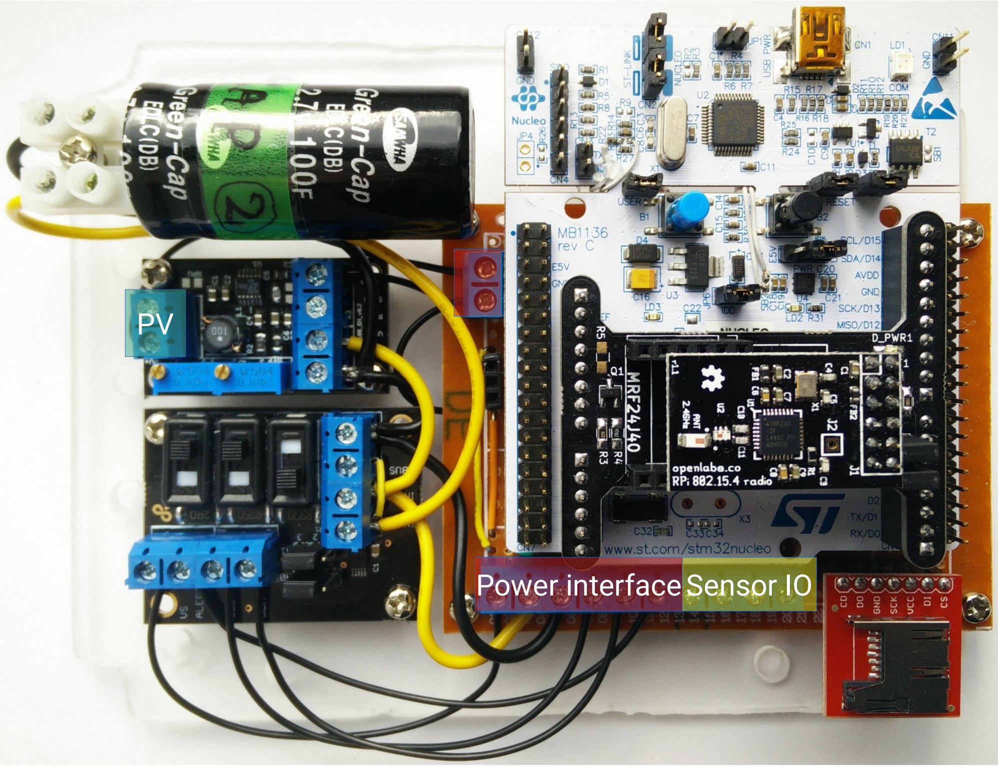
Eco – Charging Module

- 400 mA DC-DC step-up converter (LTC3105)
- Input working range: 225 mV – 5 V
- Suitable for various super caps and batteries
  - Vout: 2.2 V – 5.1 V
- Flexible adjustment for different PV-cells
  - MPPC ≤ 5 V



Eco – Test System Orchestration

- MCU: plain nucleo-l476rg
- Green-Cap (100 F 2.7 V)
- Interface board (Morpho)
- µSD-Card storage (SPI)

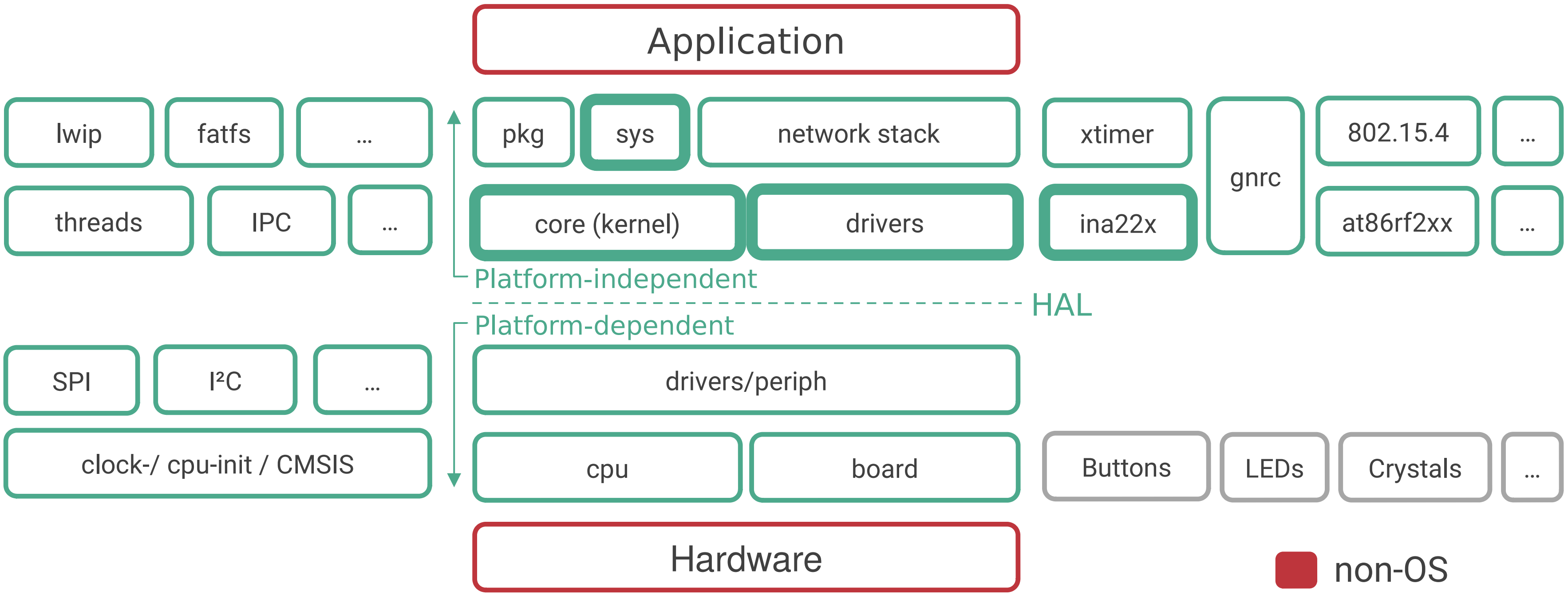


Eco

Software Integration

Eco – Software Integration

- Implemented on RIOT
- Explicit tracing
  - Measure consumption of code sections
  - Collect trace or aggregate
- Thread measurement
  - Continuous sampling
  - Attribute consumption to threads: new `es` command



Application

lwip
fatfs
...
threads
IPC
...

pkg
sys
network stack
core (kernel)
drivers

xtimer
ina22x
gnrc
802.15.4
...
at86rf2xx
...

Platform-independent

HAL

Platform-dependent

SPI
I²C
...
clock-/ cpu-init / CMSIS

drivers/periph
cpu
board

Buttons
LEDs
Crystals
...

Hardware

>> 100 different boards

🟥 non-OS
🟩 SW-module
⬜ config

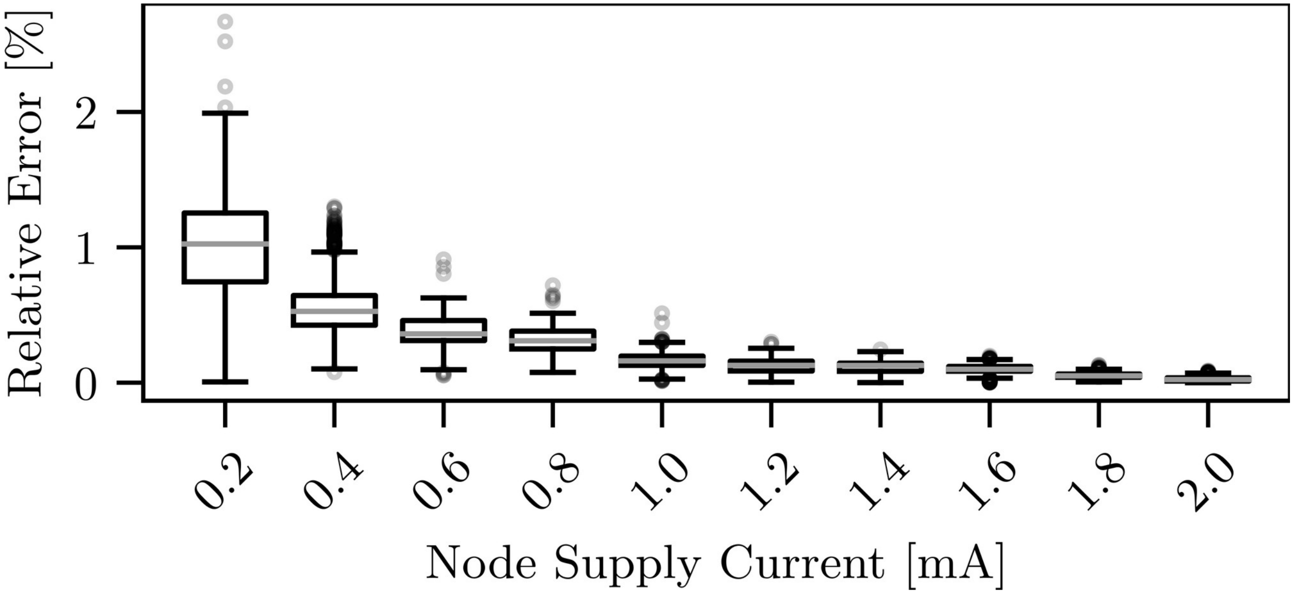
Evaluation of Eco

Methodology and Results

Evaluation of Eco - Methodology

- Measurement accuracy
  - Employ highly accurate reference (Keithley DMM7510)
  - Vary current by applying voltage to fixed resistor (Siglent SPD3303C)
- Overhead
  - Consumption: measurement module & communication bus
  - CPU-utilization: impact of using generic I²C communication as interface
  - Use `ps` command of RIOT to measure CPU-time and context switches

Measurement Accuracy

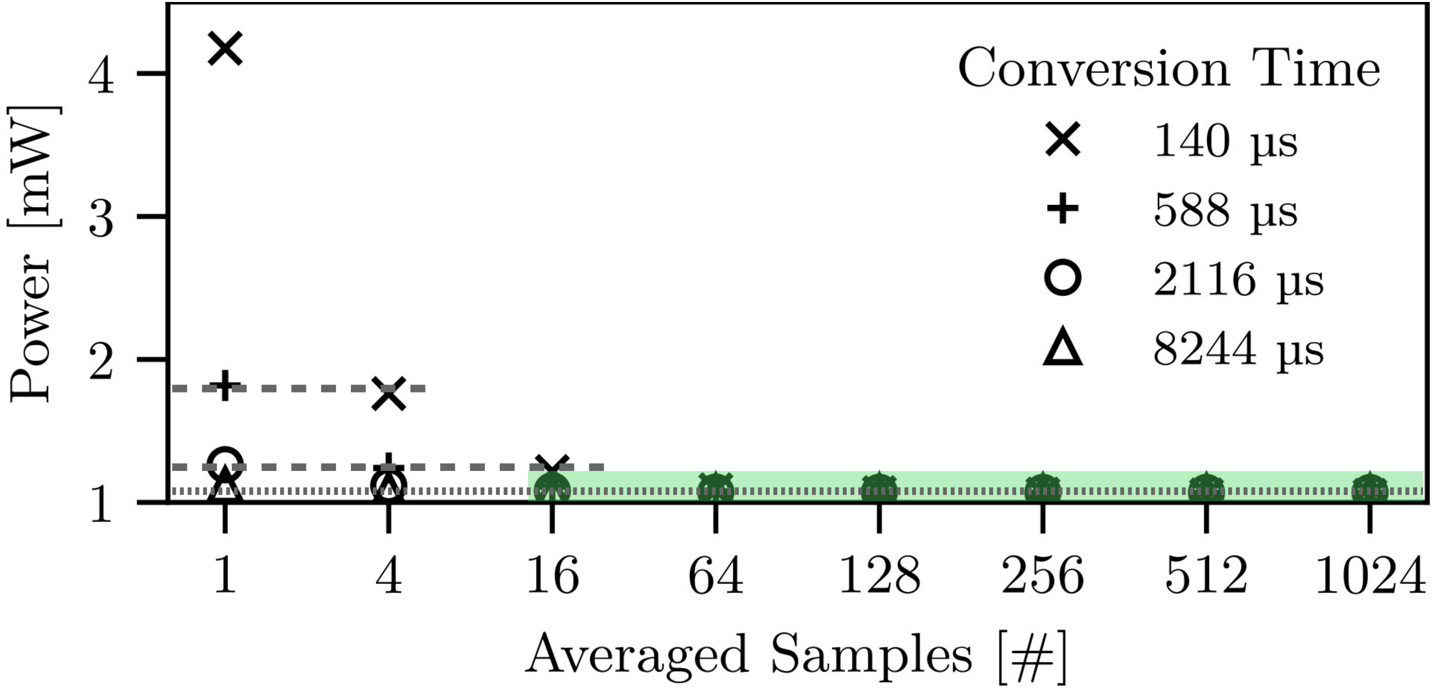


- 2 Ω shunt, 1.25 µA LSB
- 1 % median error @ 200 µA close to the LSB resolution

Power Overhead Contributers

- Shunt resistor losses
- Measurement module sampling
- I²C communication

Sampling & Communication Consumption

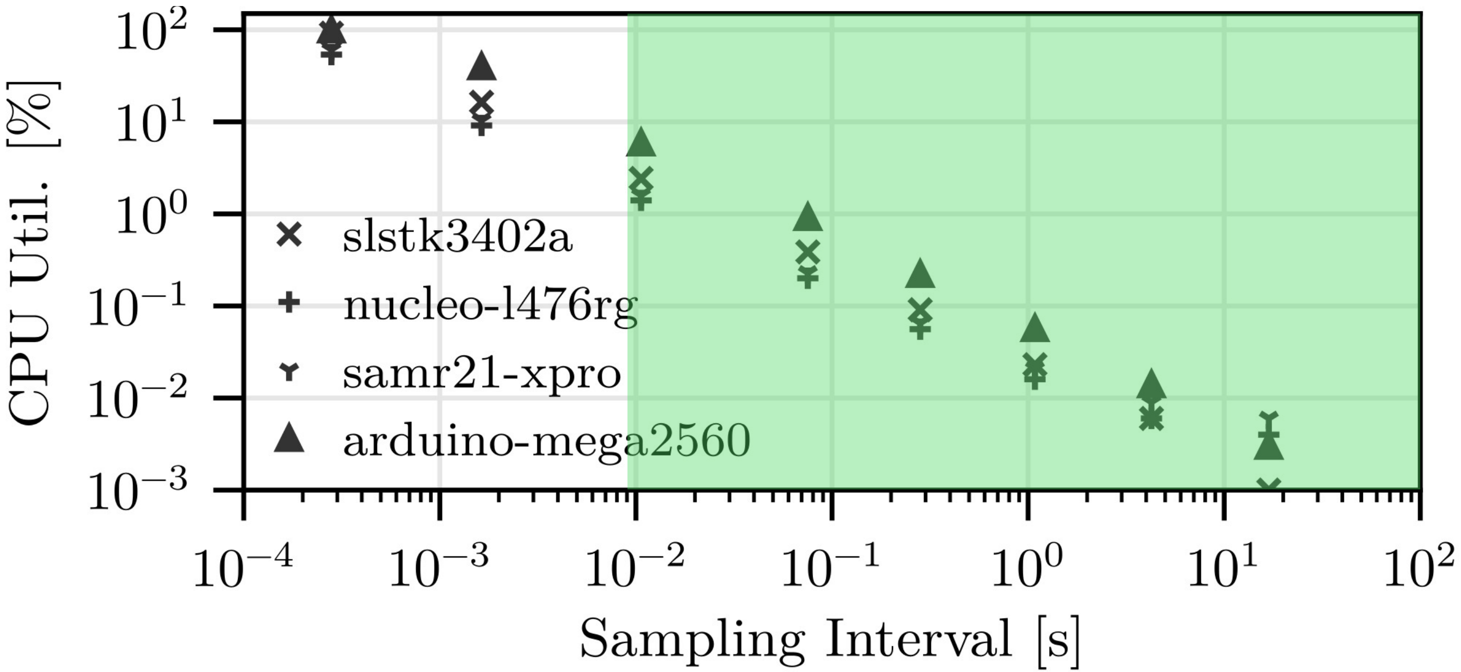


┅ Baseline consumption with active sampling ~1 mW

- - No significant difference between more averaging / sampling

🟩 Communication has no significant impact up to 588 µs @16 avg.

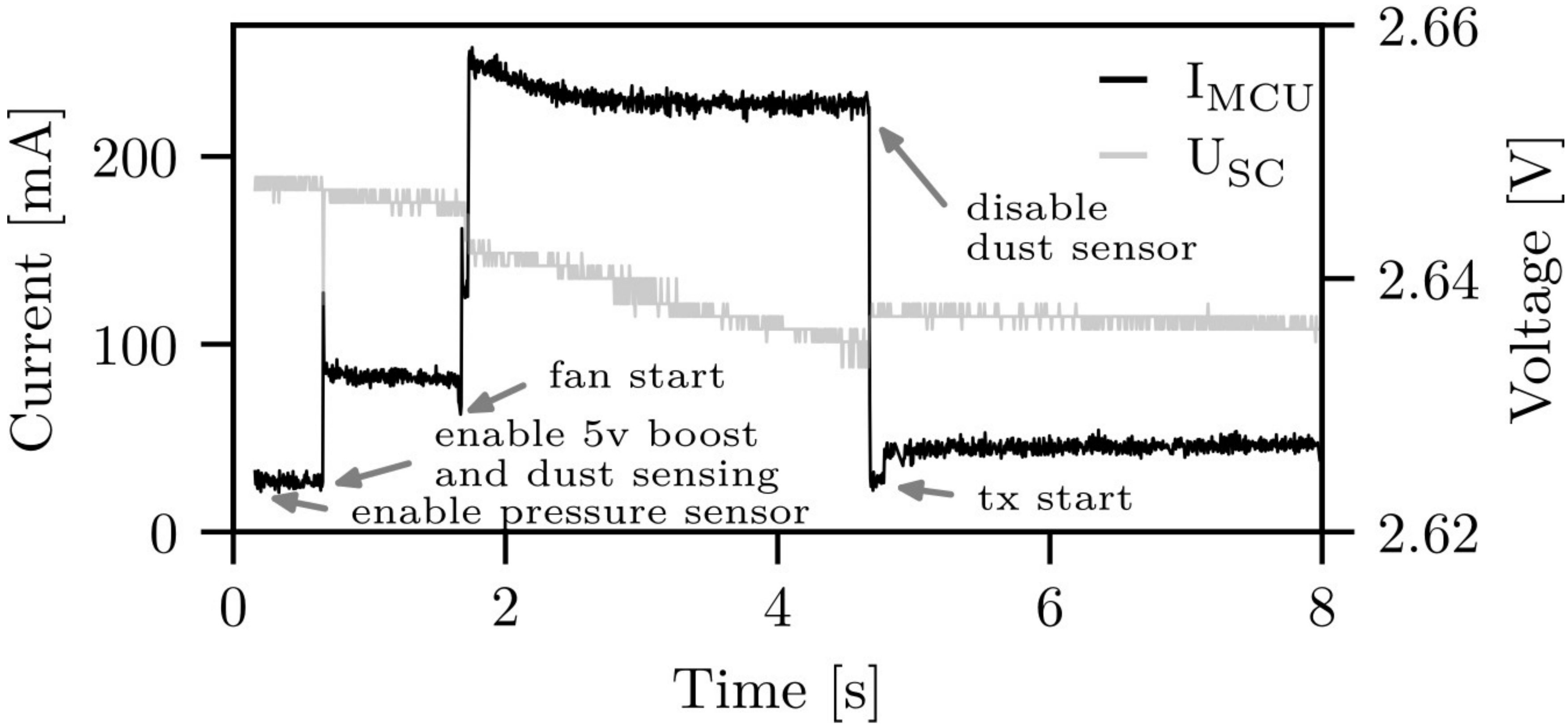
Cross-Platform CPU-Utilization



- Pessimistic scenario (separately read U and I, calculate P)
- Sampling every 10 ms possible on all platforms with CPU utilization between 1.4 % and 6 %

🟩 Communication consumption insignificant

Application Example: Active Dust Sensor



Conclusion

- Consumption of the communication-bus needs to considered
    - Can be the dominant factor for very fast sampling
    - Negligible for reasonable self-measurement data rates
- Power measurement as generic system service
    - Compatible to virtually any MCU
    - CPU-overhead low enough even for very constrained devices
- Already usable in deployments

Outlook

- Dynamic sampling rate
- Improve thread attribution with state tracking
- Add layer for allocation & prediction

Discussion

Thank you for your attention!

Questions?

For more detailed comparison see our technical report: https://arxiv.org/abs/1909.10609