VŠB TECHNICKÁ UNIVERZITA OSTRAVA

VSB TECHNICAL UNIVERSITY OF OSTRAVA

www.vsb.cz

Low Power Techniques in IoT Systems

...because every µA counts

Martin Lysek

VSB – Technical University of Ostrava

martin.lysek@vsb.cz

September 30, 2025

VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE | DEPARTMENT OF TELECOMMUNICATIONS

What We'll Cover

- **Why power consumption matters** in IoT devices
- **Software techniques** to reduce energy usage
- **Hardware strategies** for efficient design
- **Real-world examples** supported by own data
- **Measuring power consumption** with right tools

Reasons to care about power consumption

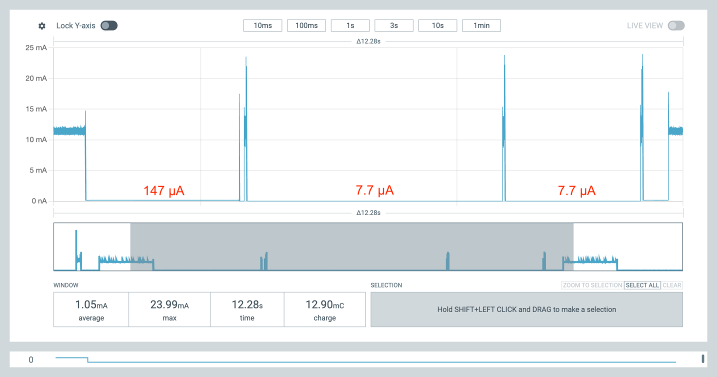
- More lifetime per battery
- Fewer truck-rolls → lower OPEX
- Smaller battery → smaller BOM → smaller device
- Easier to predict lifetime (energy budgeting)
- Some devices are physically unreachable (buried / sealed)

Sleep Modes

- Key mechanism to reduce average current in IoT devices
- MCU is mostly sleeping — not mostly computing
- Wake-up only for short tasks (measure / compute / transmit)
- Biggest power savings come from extending sleep time
- Goal: **"race to sleep"** $\rightarrow$ finish work fast and go back to sleep

Measurements

- Light-sleep: short sleeps with fast wake (keeps context)
- Deep sleep (timer): lowest power, periodic scheduled wake-up
- Deep sleep (interrupt): wake only when something actually happens



Battery Life

- Device: ESP32-H2-MINI-1 DevKit
- Battery: 3000 mAh (18650)
- TX burst: 200 mA for 0.1 s every 10 minutes
- no-sleep wait (delay loop) $\rightarrow$ ~10 mA avg $\rightarrow$ ~12.5 days
- light-sleep wait (147 μA) $\rightarrow$ ~0.18 mA avg $\rightarrow$ ~1.9 years
- deep-sleep wait (7.7 μA) $\rightarrow$ ~0.041 mA avg $\rightarrow$ ~8.3 years

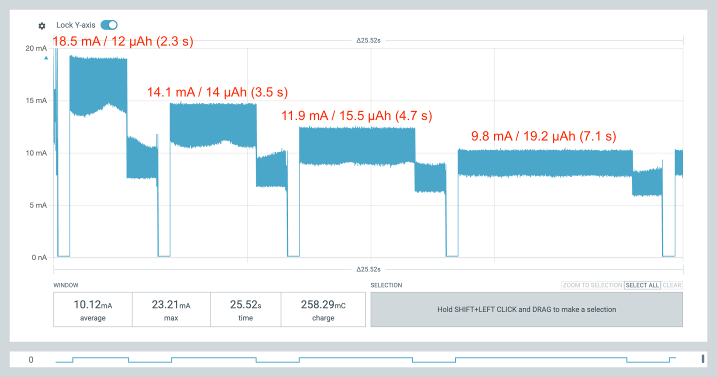
$$I_{\text{avg}} = \frac{I_{\text{wake}} \cdot t_{\text{wake}} + I_{\text{sleep}} \cdot t_{\text{sleep}}}{T_{\text{cycle}}}$$

Clock Scaling

- Lower clock frequency reduces instantaneous current (datasheet)
- Real world $\rightarrow$ task takes longer at lower clock
- Result: total energy per task can be **higher**
- Frequency scaling is only truly effective with voltage scaling.
- Race-to-sleep usually gives bigger savings than underclocking

Measurements

- Same exact workload, only CPU clock changed
- Tested frequencies: 96 / 64 / 48 / 32 MHz
- Divided by Light-sleep



Battery Life

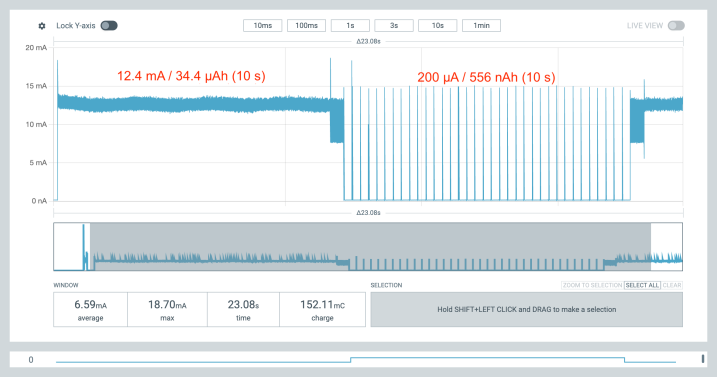
- Device: ESP32-H2-MINI-1 DevKit
- Battery: 3000 mAh (18650)
- Duty cycle: same task every 10 minutes
- Between tasks: deep-sleep 7.7 µA
- **96 MHz:** $\sim$0.082 mA avg $\rightarrow$ $\sim$1524 days
- **64 MHz:** $\sim$0.083 mA avg $\rightarrow$ $\sim$1506 days
- **48 MHz:** $\sim$0.084 mA avg $\rightarrow$ $\sim$1488 days
- **32 MHz:** $\sim$0.086 mA avg $\rightarrow$ $\sim$1453 days

Interrupt-Driven Operation

- Polling wastes power by "actively waiting"
- Interrupts allow MCU to sleep until something actually happens
- Ideal for sensors with rare or sporadic events (Accelerometers)
- Waiting is almost always dominant in battery life $\rightarrow$ optimize the wait
- Interrupt-driven wait = near-zero energy between events
- Same principle applies to I^2C / SPI / UART

Measurements

- Same signal with same time window
- Polling in while vs interrupt
- Light-sleep in interrupt mode



Battery Life

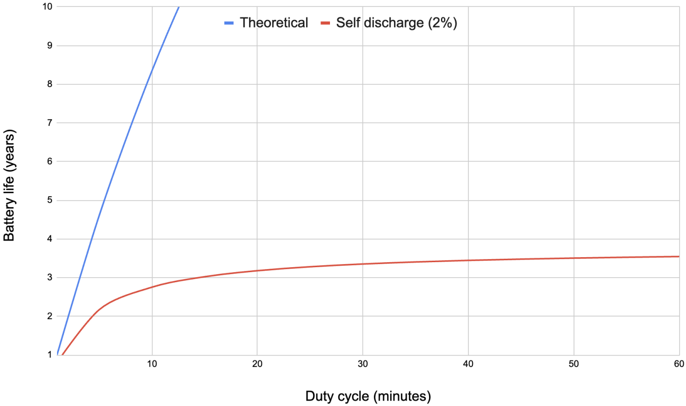
- Device: ESP32-H2-MINI-1 DevKit
- Battery: 3000 mAh (18650)
- Polling wait (12.4 mA) $\rightarrow$ $\sim$10.1 days
- Interrupt wait (0.2 mA) $\rightarrow$ $\sim$625 days
- Measured periodic signal is not great example, but already a huge win for interrupts
- With truly random events the gain is even larger

Duty Cycle

- First question: what sampling frequency do we actually need?
- Most IoT devices are not CPU bound — they are **wait bound**
- The fewer wake-ups we need, the lower the average current
- Duty cycle is the single best lever to stretch battery lifetime
- Always combine with deep/light sleep between events

Battery Life

- Battery: 3000 mAh (18650)
- TX burst: 200 mA for 0.1 s
- Battery life is not unlimited (self discharge, discharge limit)



Power Gating

- Peripherals often draw significant idle current even when "not used"
- Turning a block fully OFF is often more effective than any low-power mode
- **But** some chips have extremely low sleep currents → gating may not be worth it
- Simplest approach: P–MOSFET high–side load switch controlled by MCU GPIO
- More advanced: PMIC / load–switch ICs with enable pins (no external FET needed)
- This applies to radios, storage, displays, RFID, GNSS, sensors, audio codecs, etc.

Power Gating – Example

- Attendance system, battery powered
- RFID reader (RC522) draws e.g. ~15 mA idle
- Working hours: 07:00–20:00 (13 hours)
- Outside this window RC522 is fully OFF by MOSFET
- always ON: $24 \cdot 15$ mAh = **360 mAh / day**
- gated OFF outside working hours: $13 \cdot 15$ mAh = **195 mAh / day**

85% increase in battery lifetime using power gating

Efficient Power Supply

- DC/DC topology choice matters as much as firmware
- LDOs are simple and cheap but waste $(V_{in} - V_{out}) \cdot I$ as heat
- Buck converters (step-down) keep efficiency high over wide load range
- Example: 12V $\rightarrow$ 3.3V @ 100 mA
- LDO efficiency: $\frac{3.3}{12} = 27.5\% \rightarrow \sim$72.5% wasted
- Buck efficiency: typically 85–95%
- Battery lifetime difference here $\sim$3×–4×
- At ultra-low load (tens of μA) the DC/DC I_q can dominate $\rightarrow$ LDO can be better

Communication – Energy Cost

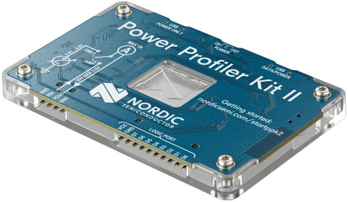
- Radio is usually the single most expensive event in the whole duty cycle
- Acknowledge based protocols cost more (TX + RX + turnaround)
- RX (listening) is often **more expensive** than TX
- Typical TX currents:
  - BLE 2.4 GHz TX: 12–20 mA
  - Zigbee 2.4 GHz TX: 25–35 mA
  - LoRa sub-GHz TX: 40–120 mA
- Wi-Fi association + DHCP + TLS handshake can cost more energy than the payload itself
- Mesh networks are expensive → they require many RX windows

Why Measure Power Consumption?

- Power optimization is only effective if measurable
- Real consumption often differs from datasheet values
- Helps identify energy bottlenecks (e.g. Wi-Fi, sensors)
- Enables battery life estimation and validation
- Essential for low-power IoT design

Power Profiler Kit II (Nordic)

- Measures current from 200 nA to 1 A
- USB-powered, integrates with Nordic Power Analyzer
- Ideal for sleep mode analysis
- Resolution: 100 nA
- Sampling rate: up to 100 ksps



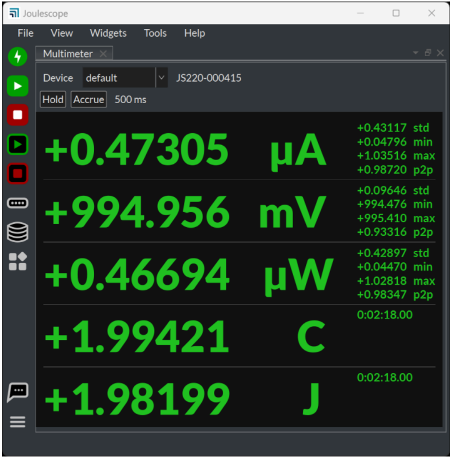
PPK2 – Example Output

- Graph shows current spikes during wake-up
- Sleep mode visible as flat low-current region
- Enables precise duty cycle tuning

JouleScope

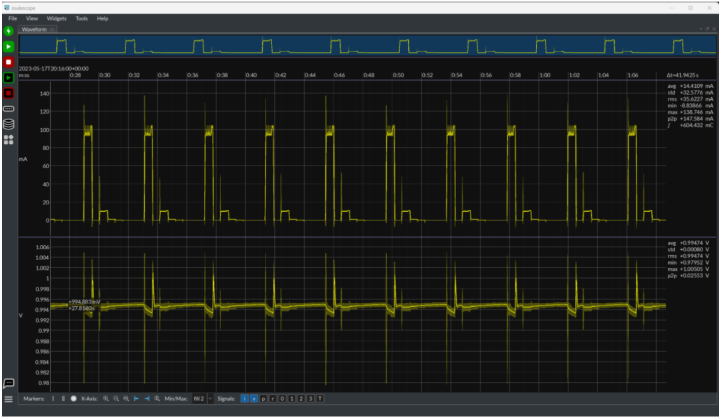
- High-precision current and voltage measurement
- Range: 1 nA to 10 A
- Measures energy directly (Joules)
- Ideal for full device profiling
- Software with real-time graphs and statistics

JouleScope – Example Output



JouleScope – Example Output

- Real-time current and voltage graphs
- Energy accumulation over time
- Exportable data for analysis



PPK2 vs. JouleScope

Feature	PPK2	JouleScope
Current Range	200 nA – 1 A	1 nA – 10 A
Voltage Range	0 - 5 V	0 – 15 V
Sampling rate	100 ksps	1 Msps
Energy Measurement	No	Yes
Software	Nordic Power Analyzer	JouleScope UI
Price	96 USD	499 USD

References

Espressif Systems
ESP32-H2 Series Datasheet
https://www.espressif.com/en/products/socs/esp32-h2

Nordic Semiconductor
Power Profiler Kit II (PPK2) Documentation
https://www.nordicsemi.com/Products/Development-hardware/Power-Profiler-Kit-2

Jetperch LLC
Joulescope Precision DC Energy Analyzer
https://www.joulescope.com/

https://github.com/lysek01/esp32-h2-power-profiling/

Thank you for your attention

Martin Lysek

VSB – Technical University of Ostrava

martin.lysek@vsb.cz

September 30, 2025