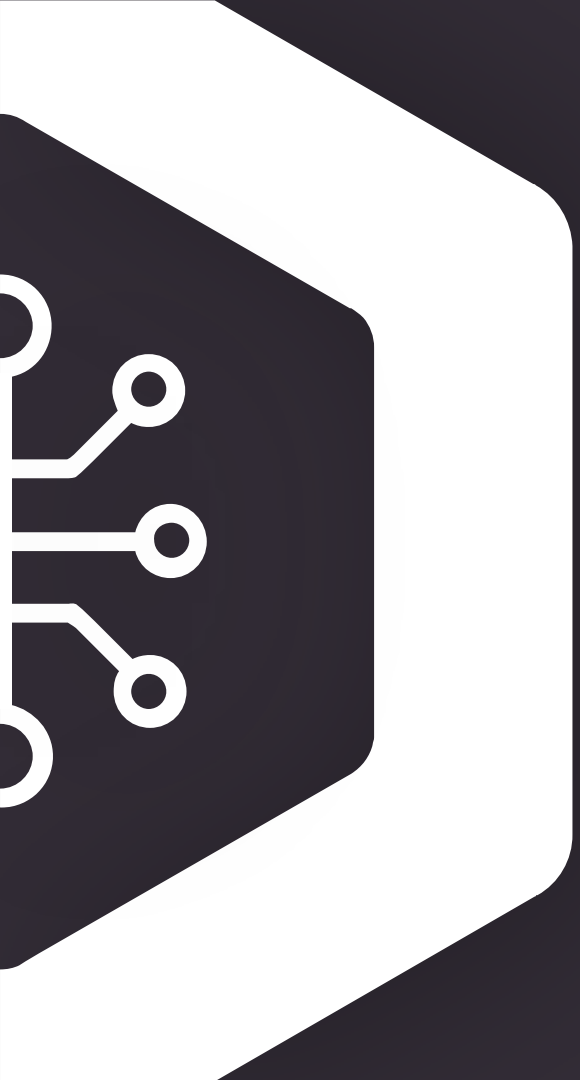


Enabling the Silicon Labs EFR family in Zephyr - a case study from a port's lifecycle

Embedded Open Source Summit, Prague, 2023-06-29

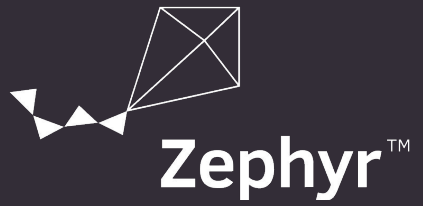
Karol Gugala kgugala@antmicro.com
Anders Pettersson Anders.Pettersson@silabs.com

antmicro

antmicro

SILICON LABS AND RTOS's

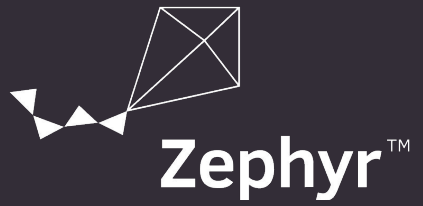
- FreeRTOS and Micrium OS
- In Simplicity Studio and SDK's
- Multiple examples provided
- Easy to get started
- Amazon adds additional "libraries"



antmicro

SILICON LABS AND ZEPHYR

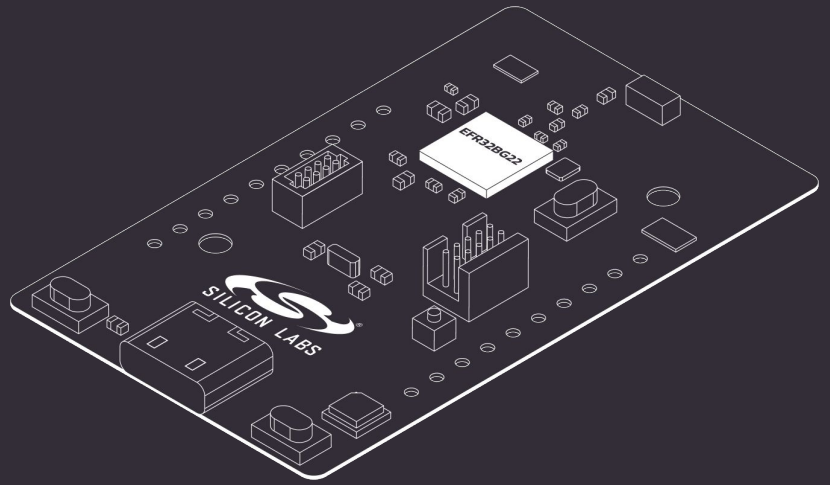
- Joined Zephyr project 2021
- Silver member
- Engaging Antmicro
- First packages available now



antmicro

EFR32 SERIES 2

- Cortex-M33 Cores, Peripherals, Zephyr BLE
- Target to support full peripheral set
- BG22 Thunderboard
- xG24 Development kit
- BG27 Development kit



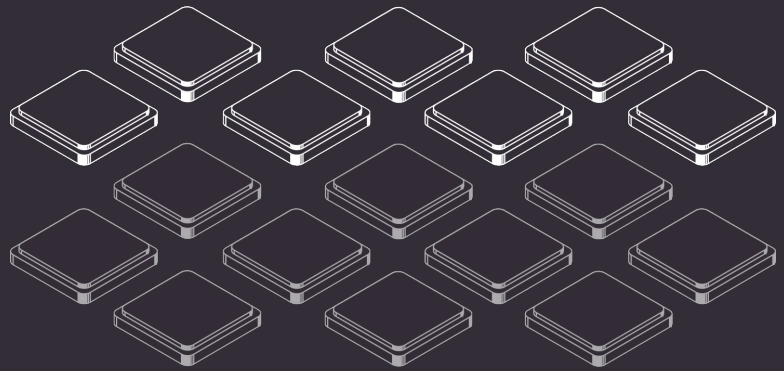
ANTMICRO AND ZEPHYR

- Platinum member of the project
- Providing support services for Zephyr
- Maintaining RISC-V Zephyr port
- Working on improving Zephyr testing with Renode
- Zephyr dashboard collects tests results for all the platforms available in Zephyr
- Renodepedia aims to collect and organize the data about all the hardware and software

antmicro

HOW TO ADD A NEW SOC FAMILY TO ZEPHYR

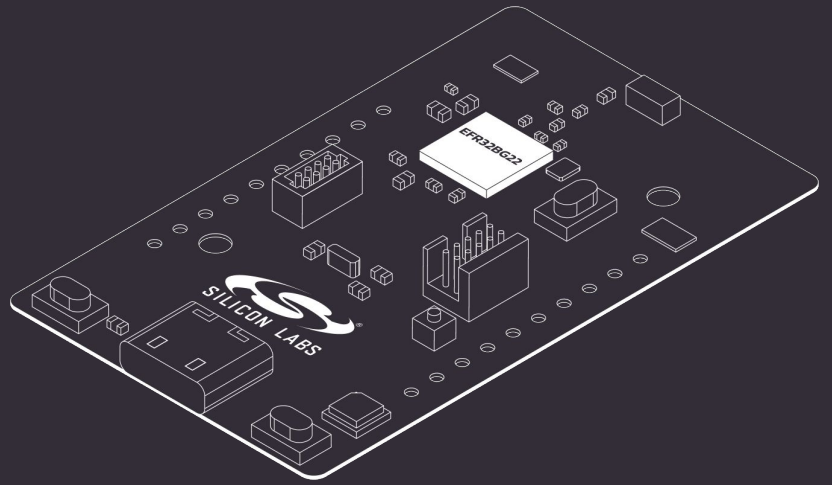
- Figure out the correct place for a new SoC
  - Do not duplicate the code - propose abstraction layers to cover the CPU/SoC/Boards division
- Update HAL if needed, take care of the resulting breakage
- Update documentation, reflecting the broader support
- Prepare hardware flashing scripts
  - This may require adding a new backend in west



antmicro

TAKING OVER EXISTING CODE BASE

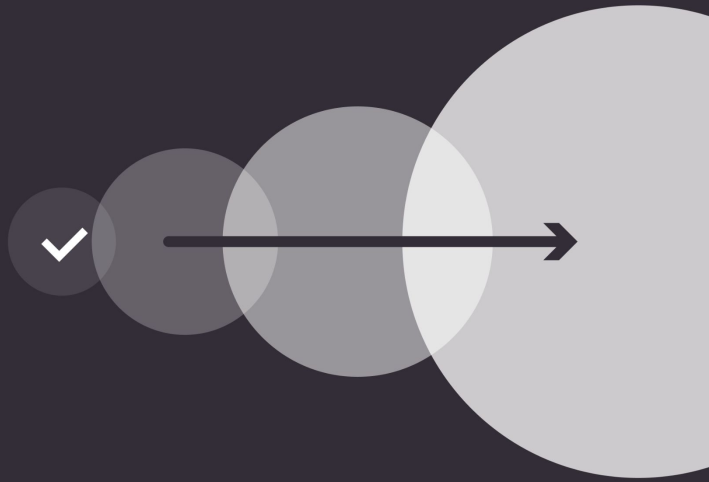
- When we started working on the EFR32 family there was an initial, WIP port
- The code needed some focused attention to adapt it to Zephyr standards
  - Divide into smaller chunks
  - Take care of licensing
  - Improve the code style in general
  - Adapt to new Zephyr APIs



antmicro

START SMALL

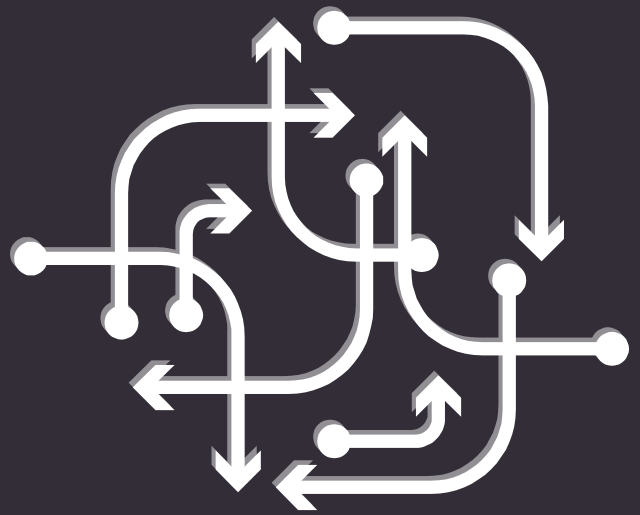
- The Initial PR added the following functionalities:
  - Pinctrl driver
  - UART driver modifications
  - EFR32BG22 SoC support
  - EFR32BG22-SLTB010A board support
- Once this has been merged, we could proceed with more functionalities



antmicro

PRIORITIZATIONS IN FACE OF CUSTOMER DEMAND

- The project had specific objectives in terms of boards to be supported, and limitations in terms of resources that could be assigned
- Of course, with so many users of the SiLabs silicon, appetites are high for supporting boards already in products etc.
- A lot of the time it is about (joint) explaining and managing expectations
- This is hard, but necessary to enable buy-in



antmicro

GETTING LICENSES IN ORDER

- On several occasion, HAL code needed to be updated to match the Zephyr requirements
- Needed to get rid of "Silicon Labs End User License Agreement" portion and move it over to zlib
- This needs internal approvals and time
- This activity was mostly about cleanup, due diligence

antmicro

TESTING IN SIMULATION

- Porting SW to new HW often requires starting with a clean state platform without any drivers at all
  - Limited debug options available (no serial output etc)
- You may have no, or limited access to hardware
  - You may not have a way to program your board
- This can be addressed with simulation - Renode supports EFR32 SoCs
- This also lets you add tests to avoid regressions - Renode Robot tests are integrated in Zephyr

antmicro

TRY IT YOURSELF

- It's super easy to get the renode simulation up and running
- Use the `renode-run` tool to quickly setup the required tools, get the binaries and run the demo
- Simply copy paste the code from this slide to your Linux terminal and you'll get Zephyr shell sample running on simulated EFR32BG22

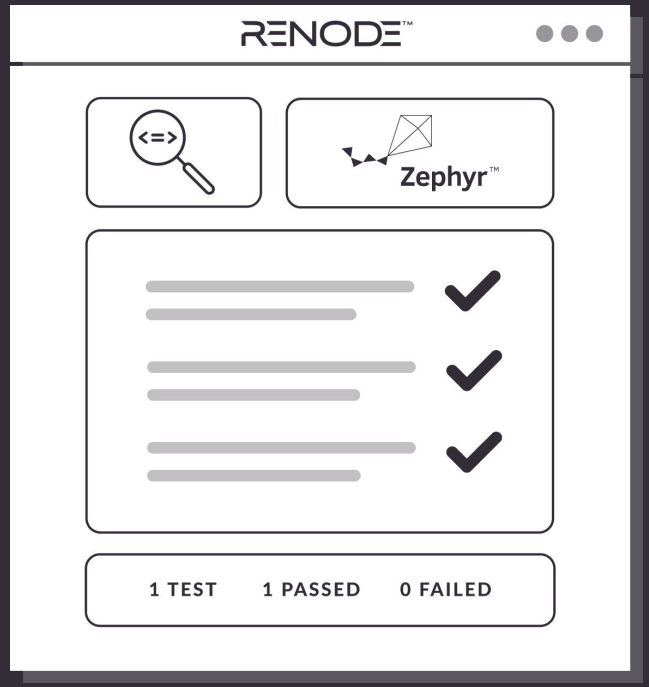
```
pip3 install --user --upgrade
git+https://github.com/antmicro/renode-run.git

renode-run demo -b efm32hg_slstk3400a
shell_module
```

antmicro

RENODE DASHBOARD

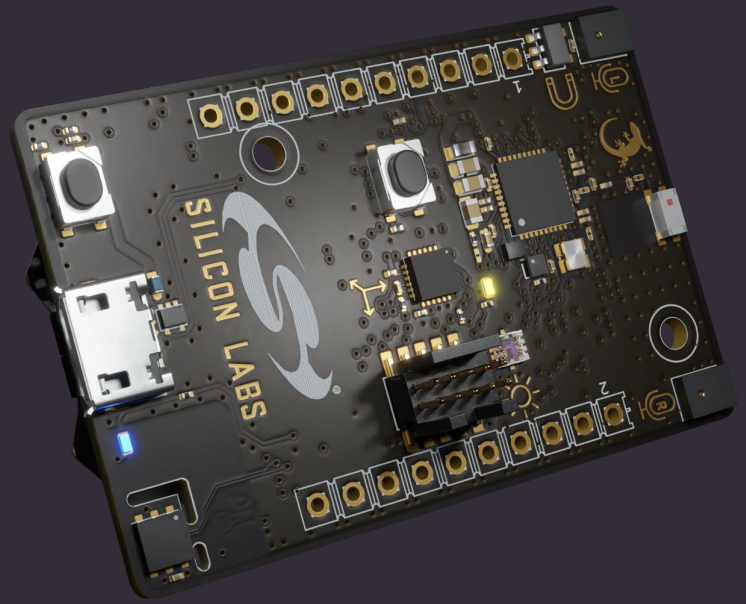
- Renode dashboard collects all the boards that are available in Zephyr, builds examples for them and run tests in Renode
- Since we already had support for EFR32 SoCs, the corresponding pages popped up automatically



antmicro

LOOKS ALSO MATTER

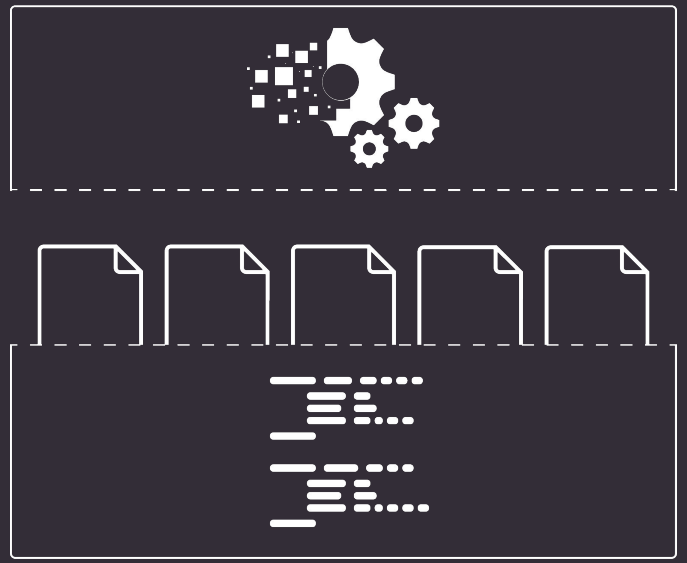
- Silicon Labs actually shares gerber files for their boards
- While we were at it, we also used our open source hardware design flow to create nice looking 3D visualizations - useful for marketing purposes



antmicro

EXTENDING TO COMPLETE SUPPORT

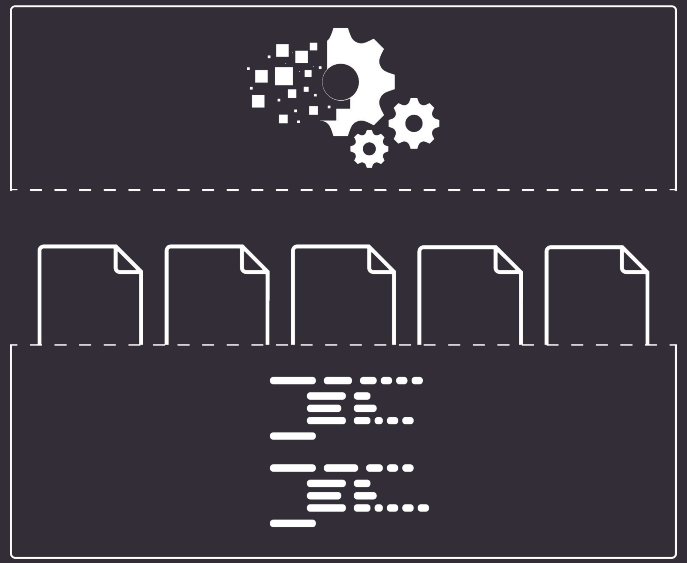
- Opening one big PR to add support for an entire board ends up with the first review saying "please split this"
- The best approach is to add all the drivers one by one
  - This way the PRs will trigger maintainers responsible for the subsystems
- Adding a driver may require HAL updates
  - HAL updates should always point to a corresponding Zephyr PR
- Once one driver is merged you often have to rebase the PRs for the rest



antmicro

HOW THIS WORKS IN PRACTICE

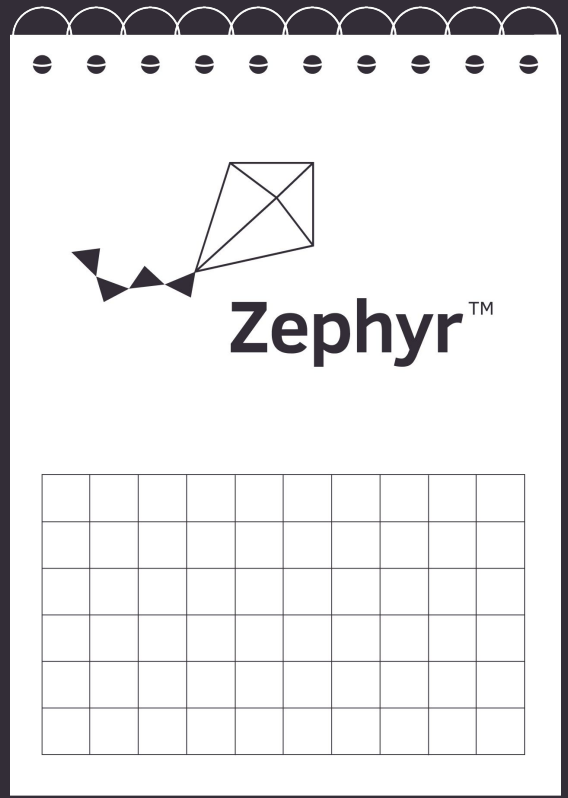
- You can take a look at a few PRs enabling subsequent functionalities of the BG22 board:
  - SPI along with SPI transaction sample
  - Watchdog Timer
  - I2C driver
  - Entropy driver
  - Bluetooth Low Energy
- Many of them already existed in Zephyr, just needed to be tweaked and enabled for the new platforms



antmicro

FITTING IN THE ZEPHYR RELEASE CYCLE

- Zephyr has a well defined release cycle
  - The dates are available on GitHub
  - Each release is preceded by a "feature freeze" period, where only bug fixes are merged into zephyr code base
- Landing large pieces of code has a big chance of hitting feature freeze
  - Plan for that
  - Know the schedule
  - Attend TSC meetings to be on top of things



antmicro

WIRELESS SUPPORT

- The EFR32 xG family has a radio block enabling wireless communication: BLE, Zigbee, Thread etc., BLE was highest priority
- Zephyr has an open source BLE stack
- But wireless SoC vendors (like SiLabs) often provide binary libraries for handling the radio
- Took care to minimize the proprietary bit to low level radio interface handling to make it "as open as (currently) possible"

antmicro

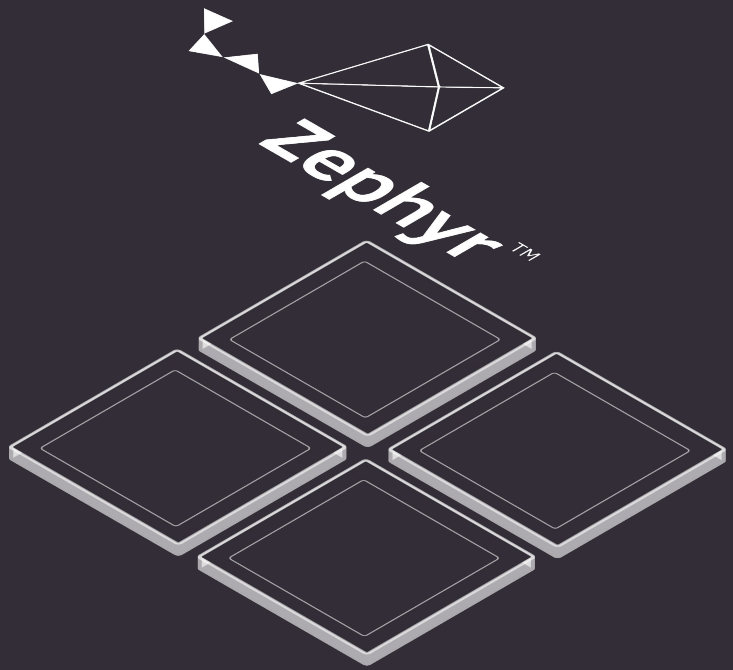
WIRELESS SUPPORT

- Zephyr defines a procedure for adding blobs
- Needs approval by the TSC
- Once this is complete, the code base can be extended to link against the blob and enable wireless communication

antmicro

ZEPHYR VS BARE METAL

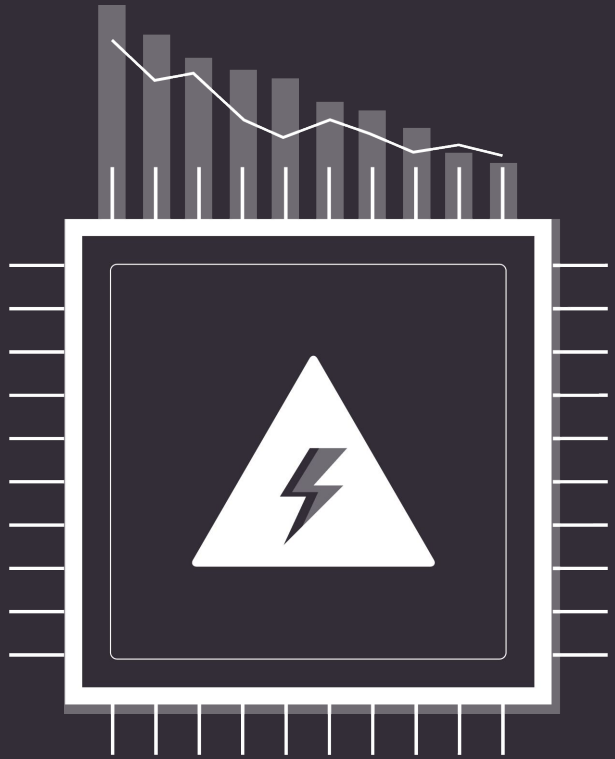
- One of the considerations when migrating from custom bare metal code is how it will impact application code:
  - Code size
  - Power efficiency
  - Portability - how hard it'll be to migrate existing logic to Zephyr



antmicro

POWER MANAGEMENT

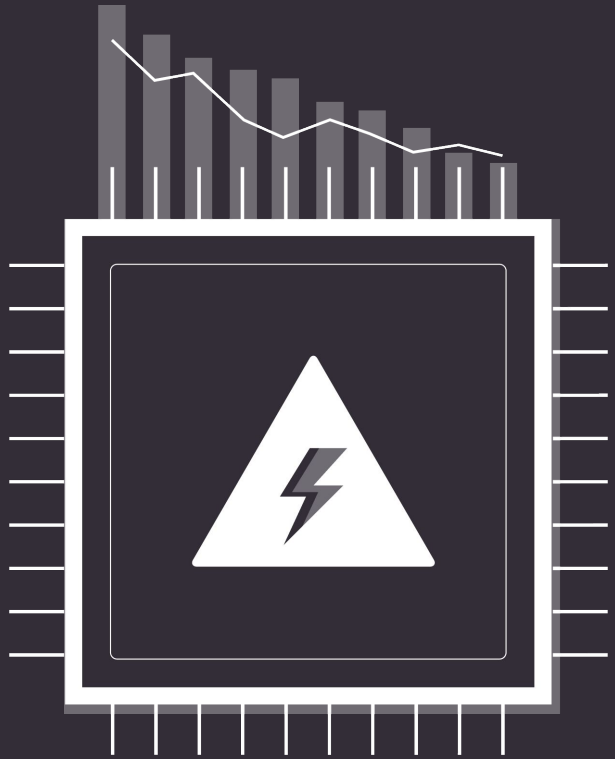
- Zephyr's power management is generic and applies to all devices
  - It assumes the device will go to the lowest available PM state if there is nothing scheduled
  - Each platform should implement a PM driver
- Mapping custom PM flow to generic Zephyr power management may require adding more sophisticated logic in the PM driver



antmicro

MINIMIZING POWER LEAKAGE

- Initial comparison showed that the bare metal code draws 97uA while Zephyr port 1860 uA
- Enabling PM in the default config reduced current draw to ~170uA
- Introducing custom, partial wake up functionality got us to the required level



antmicro

WHAT NEXT

- More boards
  - Target to use automated board config files
- New families
  - Next family is SiW917 WiFi Soc
- More Connectivity
  - Matter, WiSun, Cloud etc.

THANK YOU
FOR YOUR ATTENTION!