Zephyr RTOS: Under 1KB of RAM

Parthiban

Linumiz

whoami

- Parthiban, Linumiz GmbH
  - Embedded Linux and Zephyr RTOS development, consulting, training
  - Embedded Linux development: BSP, u-boot, Linux Kernel, Yocto Project, Buildroot
  - Zephyr: SoC, Board support, drivers
  - www.linumiz.de
- Living in **Berlin**, Germany

Linumiz and its offering

Linumiz is a domain-agnostic embedded software engineering and consulting company specializing in Linux and Zephyr RTOS. Our expertise spans board bring-up, Board Support Package (BSP) development, device driver implementation, and over-the-air (OTA) software updates, primarily for ARM-based systems but not limited to them. We empower businesses with tailored embedded solutions, ensuring reliability, scalability, and seamless hardware-software integration.

Embedded Linux

- Board support package
- Driver Development
- Board Bring up
- Custom Linux build
- Application Development

Zephyr RTOS

- Board porting
- Firmware Development
- Driver development
- Update management
- Upstream activities

Consulting

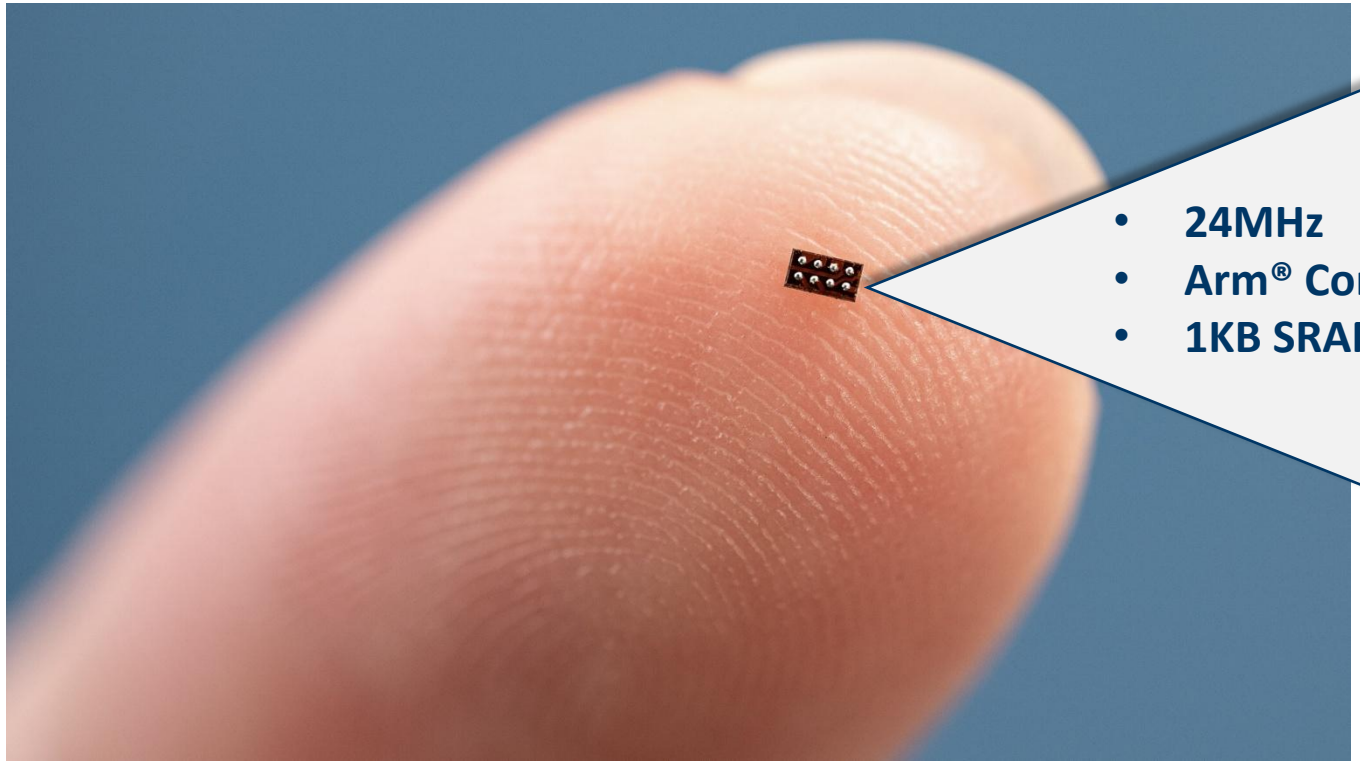
- HW recommendation
- Architecture design
- System development
- Support for specific issues
- Support Services

Device security

- Static analysis
- Dynamic analysis
- Device hardening
- Root of Trust (RoT)
- Security recommendations

Linumiz

World's smallest MCU runs Zephyr RTOS!

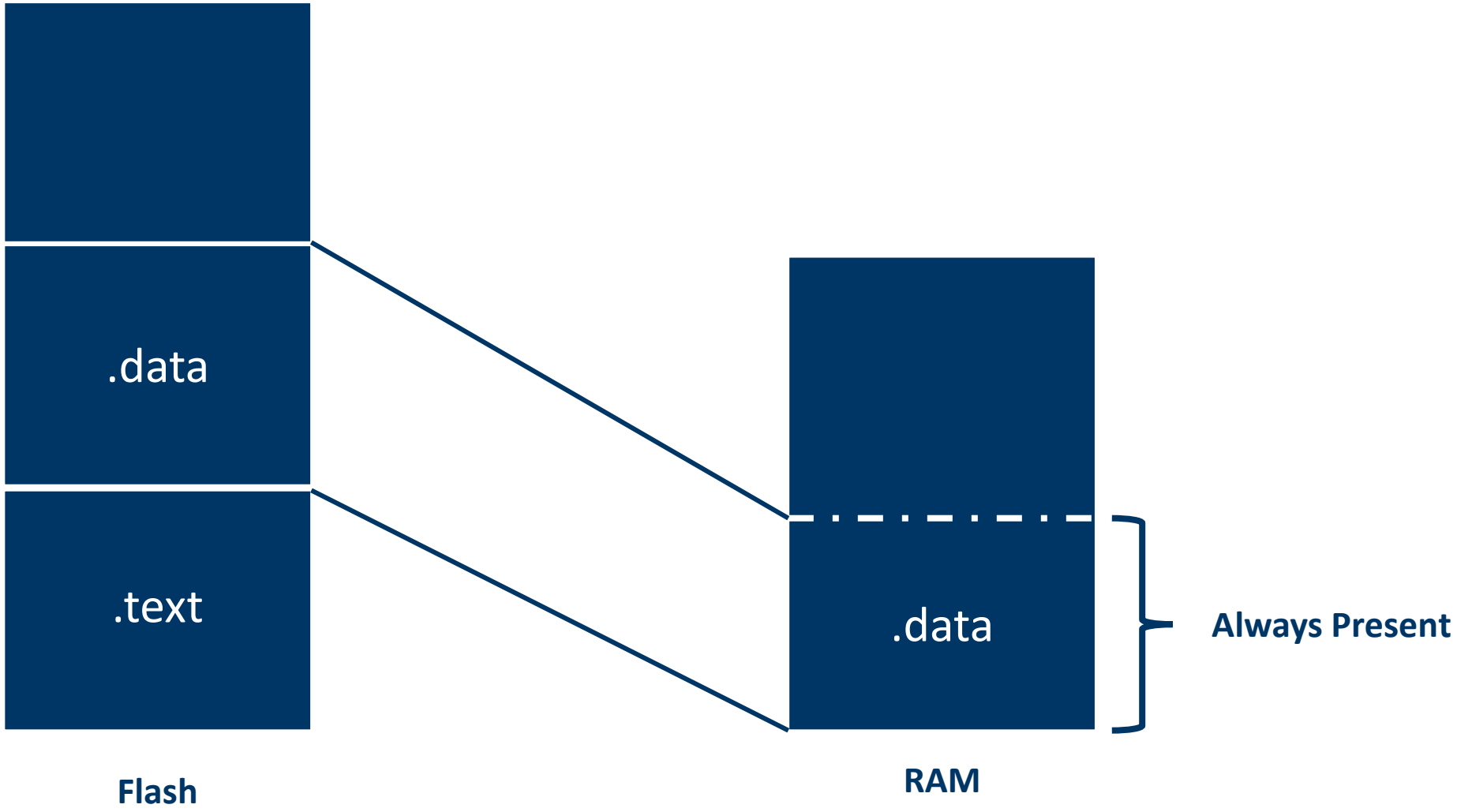


- 24MHz
- Arm® Cortex®-M0+
- 1KB SRAM

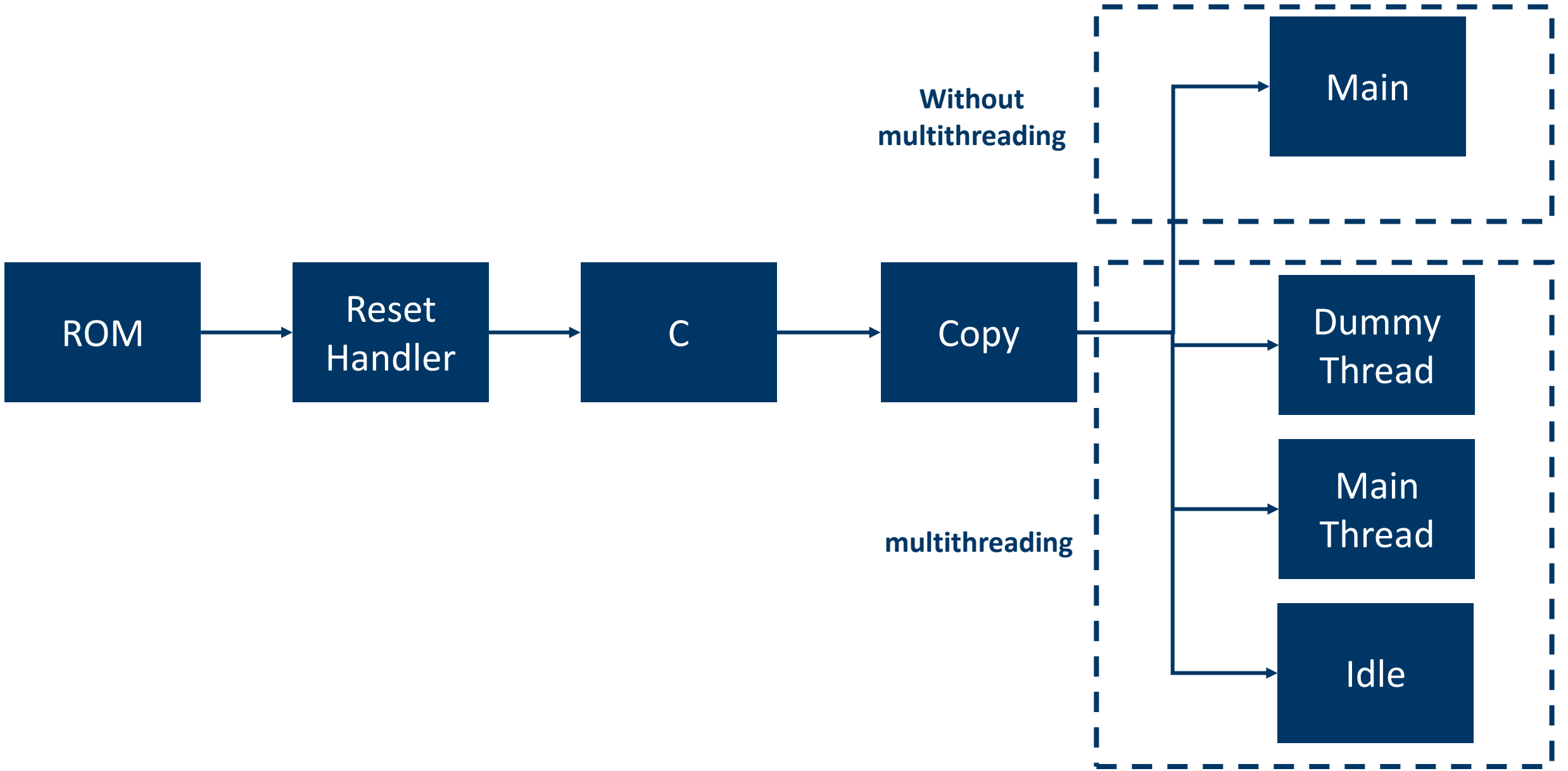
Zephyr®

Ref: https://www.ti.com/about-ti/newsroom/news-releases/2025/2025-03-11-ti-introduces-the-world-s-smallest-mcu--enabling-innovation-in-the-tiniest-of-applications.html

Memory map



Boot-Flow


Without multithreading
Main
ROM
Reset Handler
C
Copy
multithreading
Dummy Thread
Main Thread
Idle

Static threads

Main thread

Dummy thread

Idle

```
Path                                                                        Size        %  Address    Section
================================================================================================================
Root                                                                        4505  100.00%  -
├── (hidden)                                                                   0    0.00%  -
├── (no paths)                                                              3700   82.13%  -
│   ├── _kernel                                                               32    0.71%  0x20000364 bss
│   ├── _sw_isr_table                                                        256    5.68%  0x00003050 sw_isr_table
│   ├── _thread_dummy                                                        112    2.49%  0x20000248 bss
│   ├── stdout                                                                 4    0.09%  0x00003150 tbss
│   ├── z_idle_threads                                                       112    2.49%  0x20000168 bss
│   ├── z_interrupt_stacks                                                  2048   45.46%  0x20000398 noinit
│   ├── z_main_stack                                                        1024   22.73%  0x20000c98 noinit
│   └── z_main_thread                                                        112    2.49%  0x200001d8 bss
```

RAM Usage

```
[149/149] Linking C executable zephyr/zephyr.elf

Memory region         Used Size  Region Size  %age Used
           FLASH:       13567 B       256 KB      5.18%
             RAM:        4248 B        32 KB     12.96%
        IDT_LIST:           0 GB        32 KB      0.00%
```

RAM Usage (Cont.,)

drivers	381	8.59%	-
clock_control	2	0.05%	-
clock_control_mspm0.c	2	0.05%	-
__devstate_dts_ord_35	2	0.05%	0x20000159 device_states
gpio	335	7.55%	-
gpio_mspm0.c	335	7.55%	-
__devstate_dts_ord_25	2	0.05%	0x2000015f device_states
__devstate_dts_ord_26	2	0.05%	0x2000015d device_states
__devstate_dts_ord_71	2	0.05%	0x2000015b device_states
gpio_mspm0_data_a	12	0.27%	0x20000304 bss
gpio_mspm0_data_b	12	0.27%	0x200002f8 bss
gpio_mspm0_data_c	12	0.27%	0x200002ec bss
gpioa_pincm_lut	124	2.80%	0x200000bc datas
gpiob_pincm_lut	128	2.89%	0x2000003c datas
gpioc_pincm_lut	40	0.90%	0x20000014 datas
init_irq	1	0.02%	0x20000158 datas
serial	28	0.63%	-
uart_mspm0.c	28	0.63%	-
__devstate_dts_ord_58	2	0.05%	0x20000163 device_states
__devstate_dts_ord_62	2	0.05%	0x20000161 device_states
uart_mspm0_data_0	12	0.27%	0x2000014c datas
uart_mspm0_data_1	12	0.27%	0x20000140 datas
timer	16	0.36%	-
cortex_m_systick.c	16	0.36%	-
announced_cycles	4	0.09%	0x20000314 bss
cycle_count	4	0.09%	0x20000318 bss
last_load	4	0.09%	0x2000031c bss
overflow_cyc	4	0.09%	0x20000310 bss

RAM Usage (Cont.,)

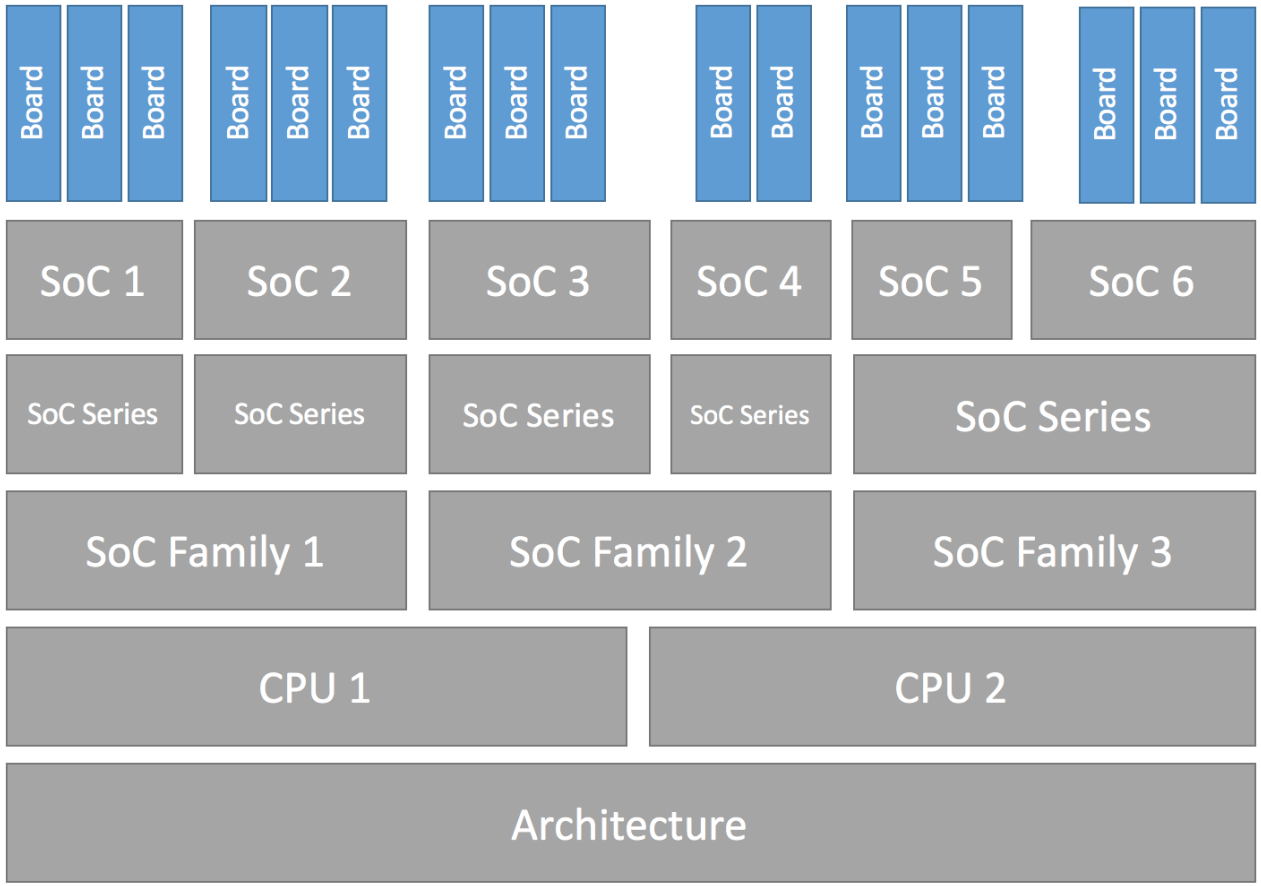
```
├── kernel                                  314   7.08%  -
│   ├── init.c                              257   5.79%  -
│   │   ├── z_idle_stacks                   256   5.77%  0x20000b58 noinit
│   │   └── z_sys_post_kernel                 1   0.02%  0x20000350 bss
│   ├── timeout.c                            20   0.45%  -
│   │   ├── announce_remaining                4   0.09%  0x2000034c bss
│   │   ├── curr_tick                         8   0.18%  0x200002d0 bss
│   │   └── timeout_list                      8   0.18%  0x20000138 datas
│   └── timeslicing.c                        37   0.83%  -
│       ├── pending_current                   4   0.09%  0x20000340 bss
│       ├── slice_expired                     1   0.02%  0x20000351 bss
│       ├── slice_max_prio                    4   0.09%  0x20000344 bss
│       ├── slice_ticks                       4   0.09%  0x20000348 bss
│       └── slice_timeouts                   24   0.54%  0x200002b8 bss
```

Fake it and Make it

```
sram0: memory@20000000 {
        compatible = "mmio-sram";
        reg = <0x20000000 0x800>;
};
```

Porting Zephyr

- ARCH
- SoC
- Drivers
  - Timer
  - GPIO
  - Pin Controller
  - Clock
  - UART
- Devicetree
- Board



First run

```
int main(void)
{
        while (true) {
                printf("Hello World! %s\n", CONFIG_BOARD_TARGET);
                k_msleep(1000);
        }

        return 0;
}
```

First run

```
int main(void)
{
        while (true) {
                printf("Hello World! %s\n", CONFIG_BOARD_TARGET);
                k_msleep(1000);
        }

        return 0;
}
```

First run

```
int main(void)
{
        while (true) {
                printf("Hello World! %s\n", CONFIG_BOARD_TARGET);
                k_msleep(1000);
        }

        return 0;
}
```

Missing systick

First run

```
int main(void)
{
        while (true) {
                printf("Hello World! %s\n", CONFIG_BOARD_TARGET);
                k_msleep(1000);
        }

        return 0;
}
```

Missing systick

Roll of system timer

Revamping timer

```
void sys_clock_announce(int32_t ticks);

void sys_clock_set_timeout(int32_t ticks, bool idle);

uint32_t sys_clock_elapsed(void);

void sys_clock_idle_exit(void);

void sys_clock_disable(void);

uint32_t sys_clock_cycle_get_32(void);
```

Real Kernel Size

- sleep (~300 bytes)
- mutex, semaphore, spinlock
- IPC

Thank You

Linumiz