CEC Continuing Education Center

DesignNews

Mastering Zephyr RTOS

DAY 5 : Debugging, Logging, and Best Practices

Sponsored by

DigiKey

BENINGO EMBEDDED GROUP

informa markets

©2025 Beningo Embedded Group, LLC. All Rights Reserved.

Webinar Logistics

- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Group Chat’ by maximizing the chat widget in your dock.

©2025 Beningo Embedded Group, LLC. All Rights Reserved.

CEC Continuing Education Center

BENINGO EMBEDDED GROUP

DigiKey

THE SPEAKER

Jacob Beningo

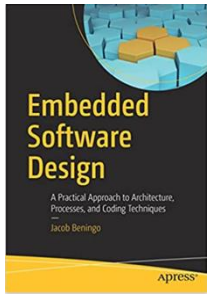
Jacob@beningo.com

jacobbeningo

Beningo Embedded Group – CEO / Founder

Focus: Software Architecture, Processes, and Dev Skills

At Beningo Embedded Group, we believe everyone deserves the skills to confidently advance their careers, meet deadlines, and deliver quality embedded systems. We provide modern strategies, insights, and hands-on training to equip developers and teams with the tools they need to succeed.



EMBEDDED SOFTWARE ACADEMY BY BENINGO

Visit **www.beningo.com** to learn more

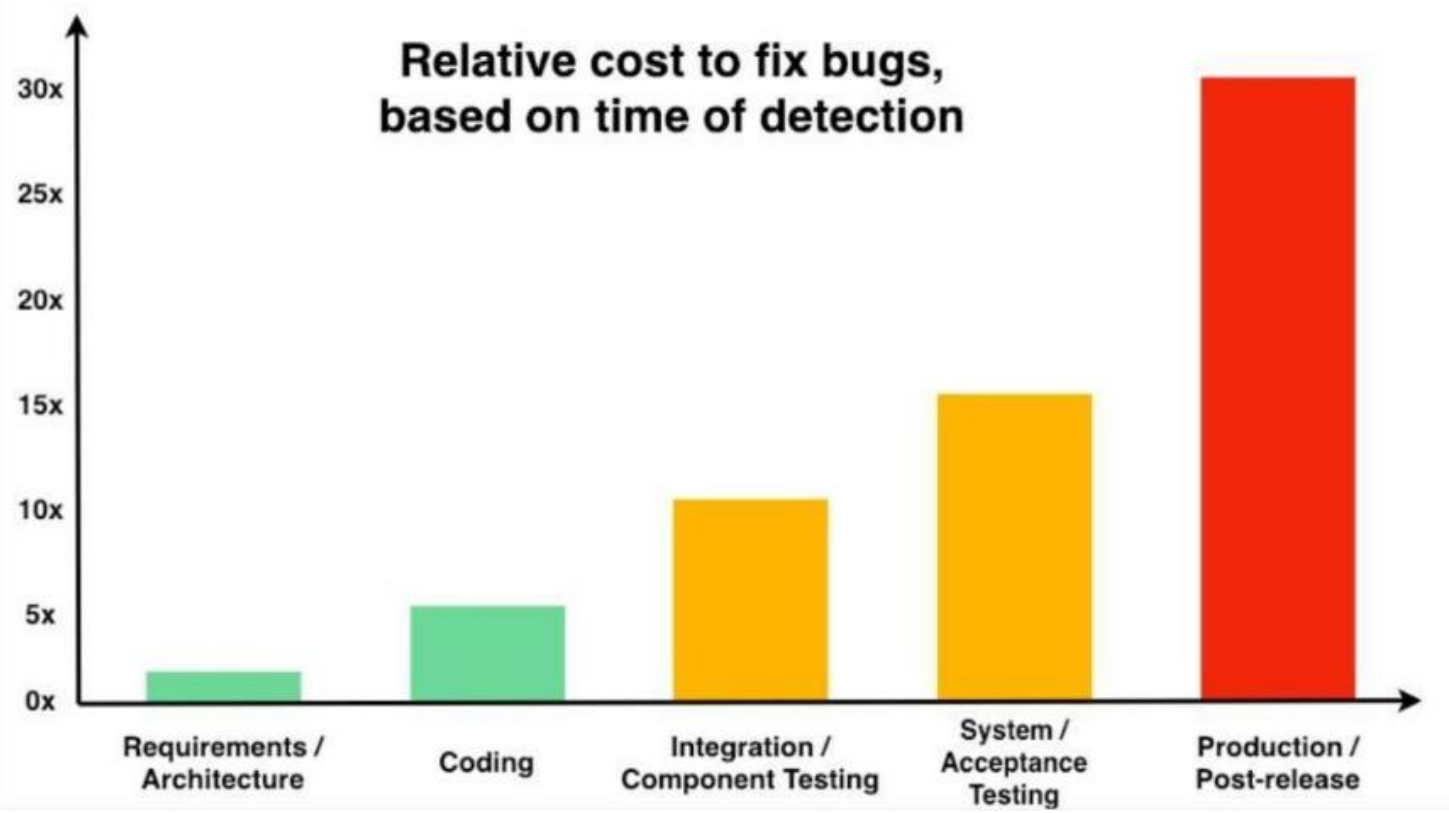
©2025 Beningo Embedded Group, LLC. All Rights Reserved.

Debugging

01

©2025 Beningo Embedded Group, LLC. All Rights Reserved.

Bugging is Expensive!

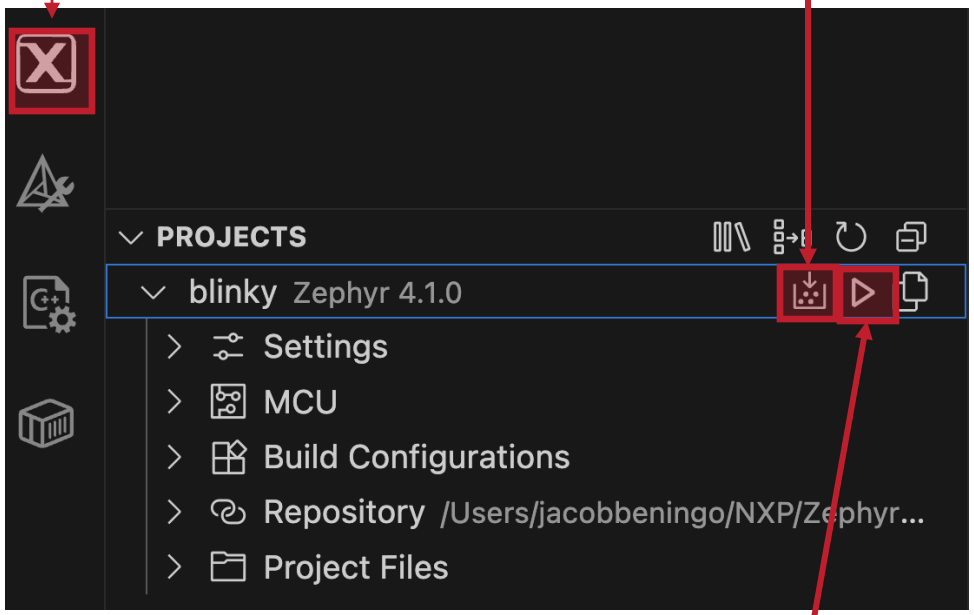


©2025 Beningo Embedded Group, LLC. All Rights Reserved.

Debug Using VS Code

MCUXpresso for VS Code Extension

Compile



Debug

synchronization > .vscode > {} launch.json > Launch Targets > {} Debug with SEGGER J-Link

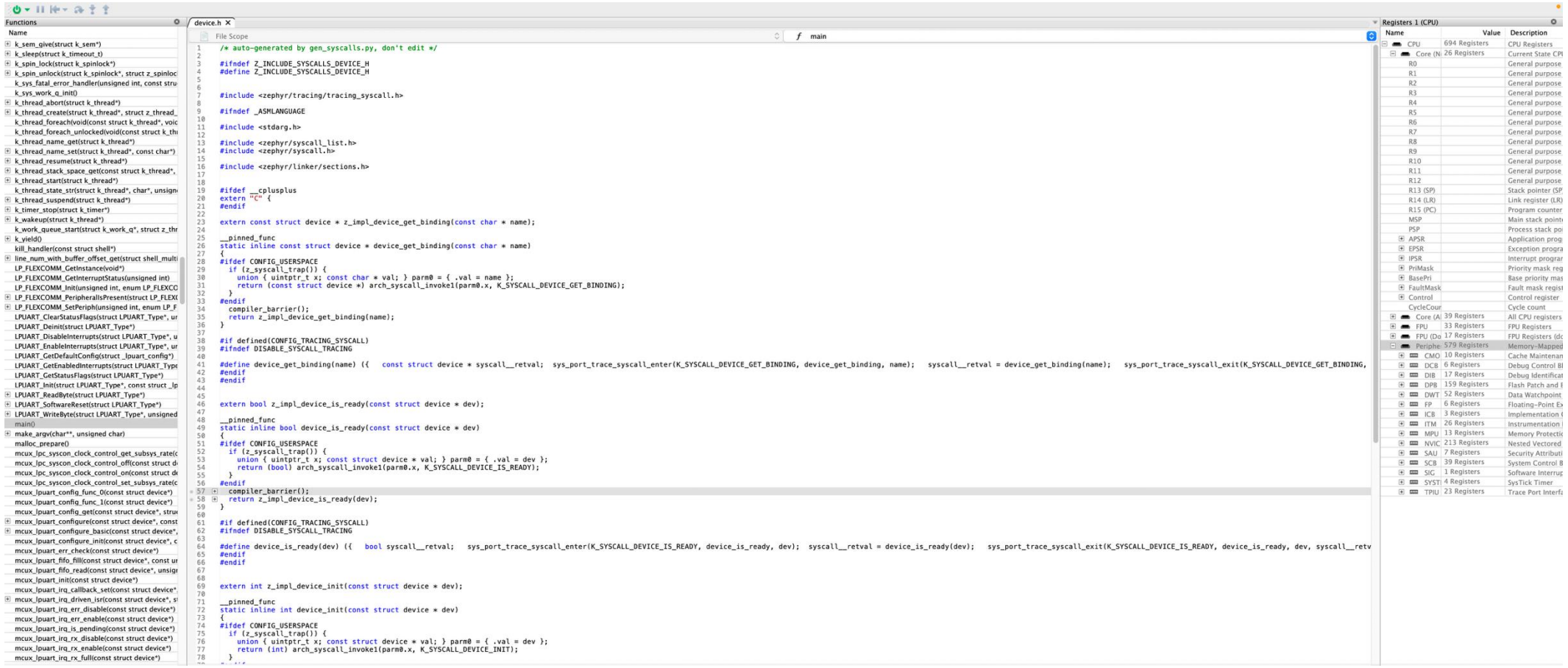
```
{
  "configurations": [
    {
      "type": "mcuxpresso-debug",
      "name": "Debug with SEGGER J-Link",
      "request": "launch",
      "cwd": "${workspaceFolder}",

      "executable": "${workspaceFolder}/build/zephyr/zephyr.elf",
      "stopAtSymbol": "main",
      "probeSerialNumber": "505100646",
      "isAttach": false,
      "probeType": "SEGGER",
      "skipBuildBeforeDebug": false,
      "skipPreFlashActions": false,

      "gdbInitCommands": [
        "set remotetimeout 600",
        "set debug-file-directory",
        "set non-stop off"
      ],
```

blinky > src > C main.c > main(void)

```
61
49      while (1) {
50          gpio_pin_set_dt(&led0, led_state);
51          gpio_pin_set_dt(&my_led, led_state);
52
53          printk("Blink! LED0 and myLED state: %s\n", led_state ?
54
55          led_state = !led_state;
56          k_msleep(SLEEP_TIME_MS);
57      }
58
59      return 0;
60  }
```


©2025 Beningo Embedded Group, LLC. All Rights Reserved.

SEGGER Ozone



©2025 Beningo Embedded Group, LLC. All Rights Reserved.

Audience POLL Question

What is your preference for debugging your Zephyr RTOS applications?

a) Using raw gdb with west debug
b) Using Visual Studio Code
c) Using SEGGER Ozone
d) Using run-time capabilities like logs, the shell, etc

©2025 Beningo Embedded Group, LLC. All Rights Reserved.

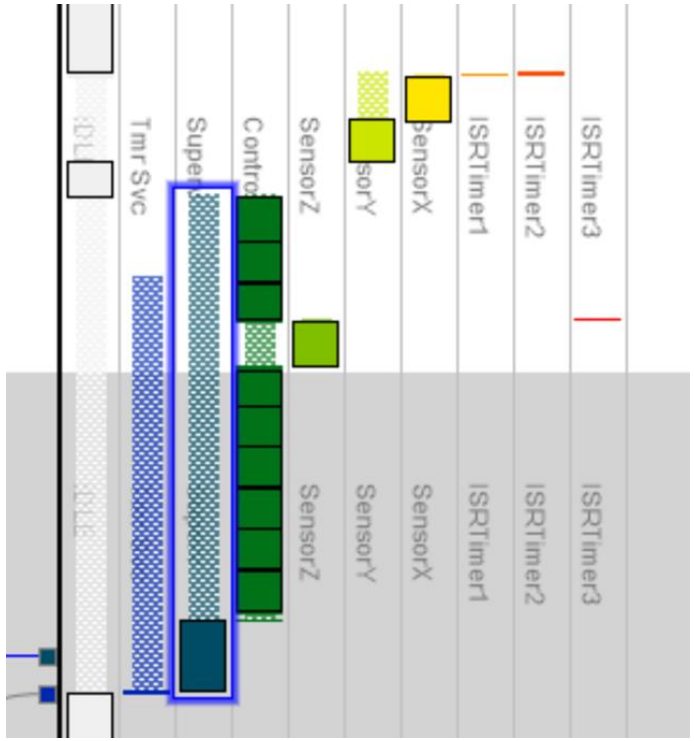
Debug Trace

02

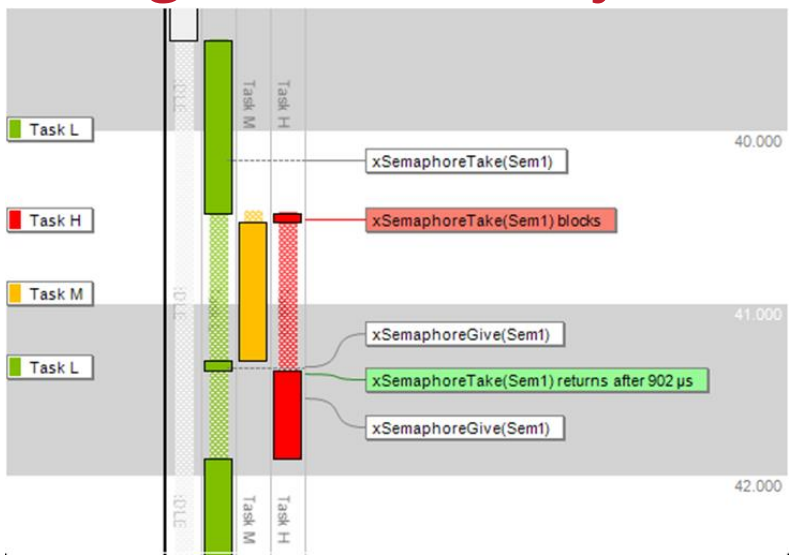
©2025 Beningo Embedded Group, LLC. All Rights Reserved.

The Power of Tracing

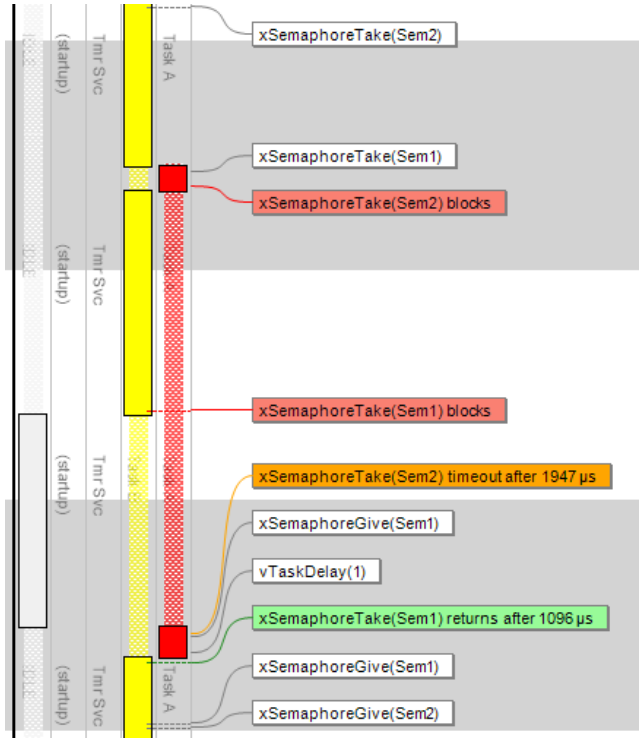
Danger #1 – Task



Danger #2 – Priority

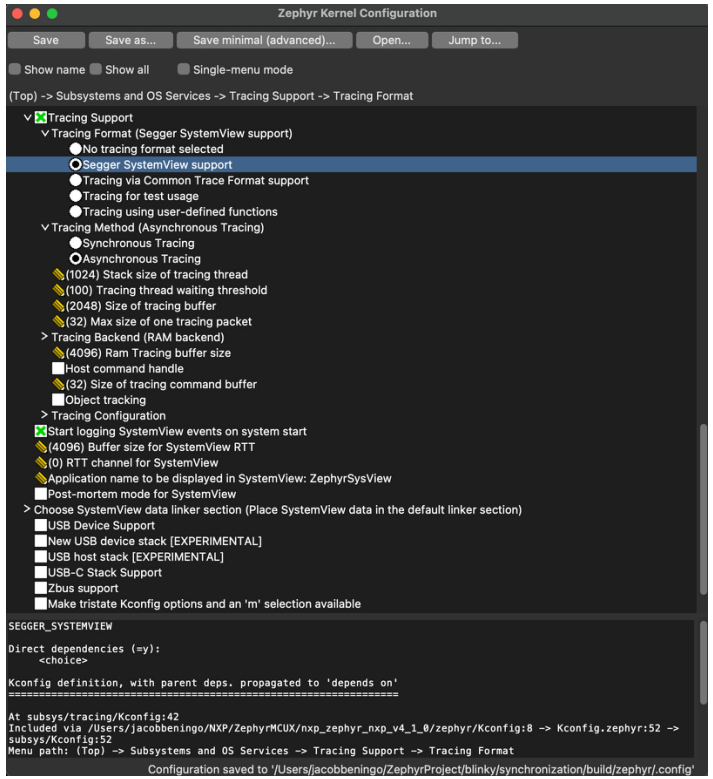


Danger #3 - Deadlock

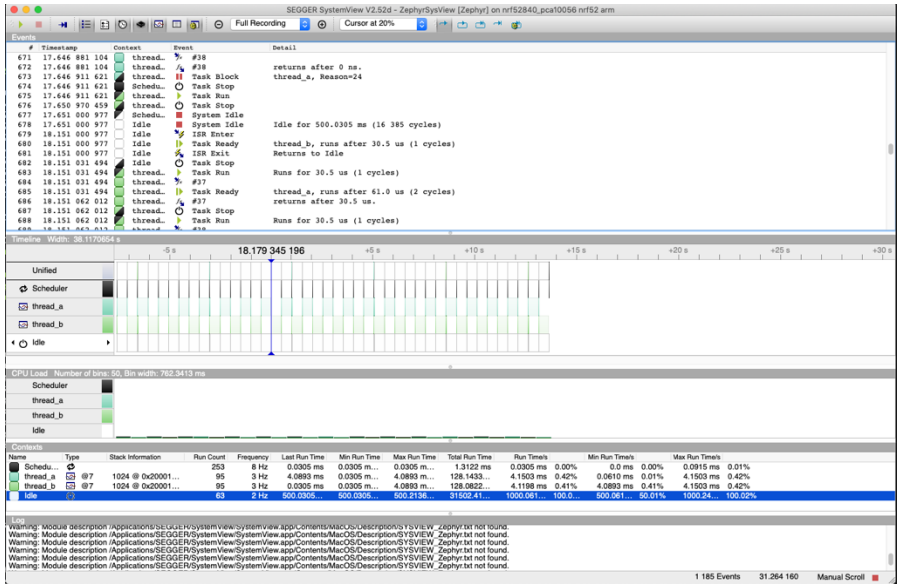


©2025 Beningo Embedded Group, LLC. All Rights Reserved.

SEGGER SystemView

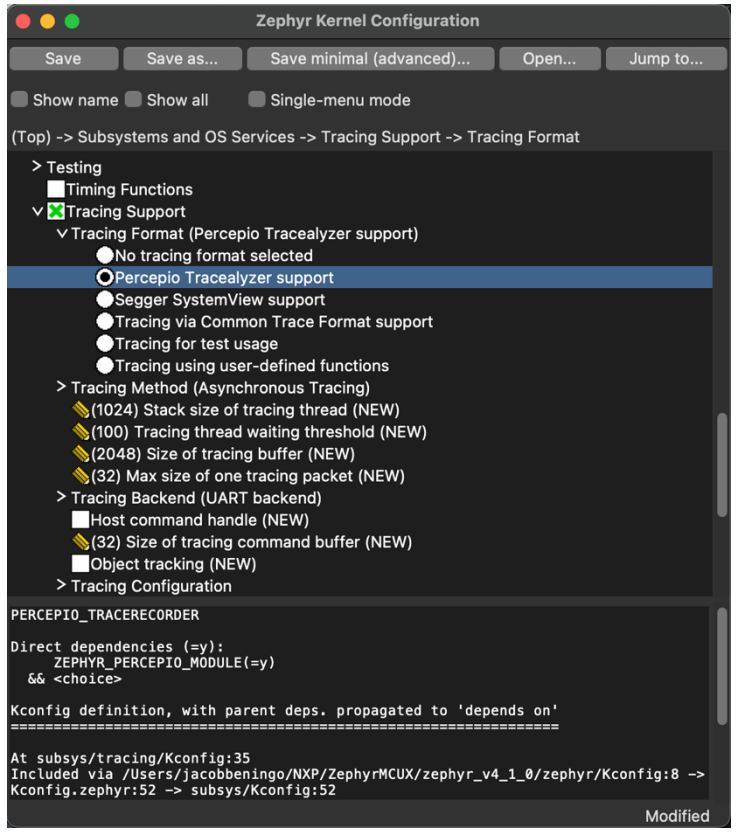



```
CONFIG_STDOUT_CONSOLE=y
CONFIG_THREAD_NAME=y
CONFIG_SEGGER_SYSTEMVIEW=y
CONFIG_USE_SEGGER_RTT=y
CONFIG_TRACING=y
```

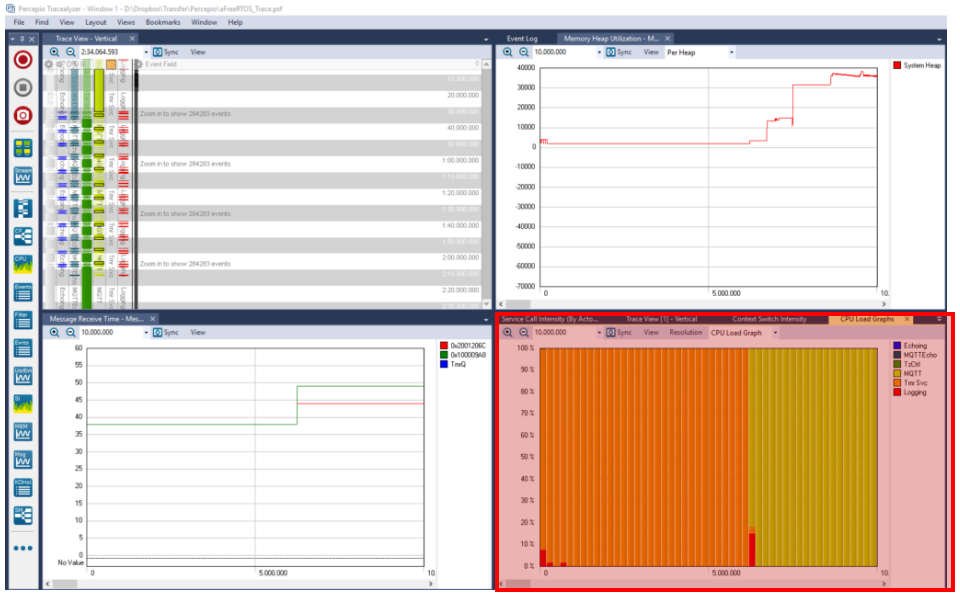


©2025 Beningo Embedded Group, LLC. All Rights Reserved.

Percepio TraceView




```
CONFIG_TRACING=y

CONFIG_PERCEPIO_TRACERECORDER=y
```



©2025 Beningo Embedded Group, LLC. All Rights Reserved.

Audience POLL Question

Which of the following best describes the purpose of tracing in an embedded system running Zephyr RTOS?

a) To encrypt communication between tasks and peripherals
b) To monitor and record system events for debugging and performance analysis
c) To configure GPIO pins dynamically during runtime
d) To allocate stack memory for kernel threads

©2025 Beningo Embedded Group, LLC. All Rights Reserved.

Logging

03

©2025 Beningo Embedded Group, LLC. All Rights Reserved.

What is logging?

A structured way to report system messages like errors, warnings, info, and debug output.

Key Features in Zephyr Logging:

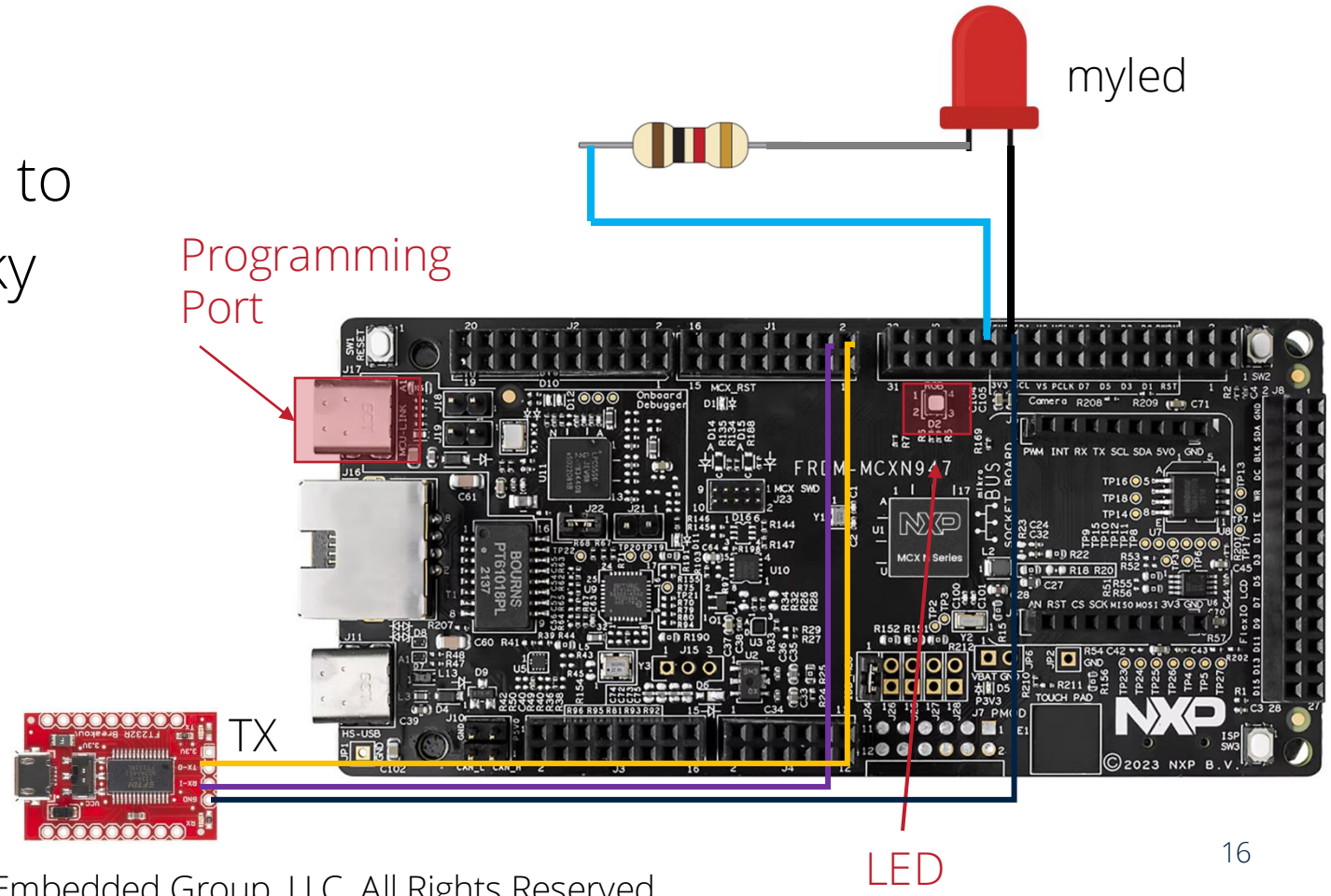
- LOG_ERR(), LOG_WRN(), LOG_INF(), LOG_DBG() macros
- Module-based: LOG_MODULE_REGISTER(main)
- Adjustable log levels per module
- Deferred or immediate logging modes
- Output backends: UART, RTT, memory

©2025 Beningo Embedded Group, LLC. All Rights Reserved.

Goals and Hardware Setup

The purpose of this example is to modify the samples/basic/blinky example to:

- Display log info on USART2
- Keep the Shell on USART4
- Explore Log messages



©2025 Beningo Embedded Group, LLC. All Rights Reserved.

Enabling Logging

app.overlay

```
/ {
    chosen {
        zephyr,console = &flexcomm2_lpuart2;
        zephyr,shell-uart = &flexcomm4_lpuart4;
        zephyr,uart-mcumgr = &flexcomm2_lpuart2;
    };

    leds {
        compatible = "gpio-leds";

        my_led: my_led {
            gpios = <&gpio1 22 GPIO_ACTIVE_LOW>;
            label = "My Custom LED";
        };
    };

    aliases {
        myled = &my_led;
    };
};
```

prj.conf

```
# Required for printk on UART2 (console)
CONFIG_SERIAL=y
CONFIG_PRINTK=y
CONFIG_CONSOLE=y
CONFIG_UART_CONSOLE=y

# Logging configuration - use UART2 for logs
CONFIG_LOG=y
CONFIG_LOG_BACKEND_UART=y
CONFIG_LOG_MODE_DEFERRED=y
CONFIG_LOG_MODE_IMMEDIATE=n
CONFIG_LOG_DEFAULT_LEVEL=3
CONFIG_LOG_BACKEND_UART_AUTOSTART=y

# Disable RTT backend if you don't need it
CONFIG_LOG_BACKEND_RTT=n

# Shell configuration - use UART4 for shell
CONFIG_SHELL=y
CONFIG_SHELL_BACKEND_SERIAL=y
CONFIG_DEVICE_SHELL=y
CONFIG_KERNEL_SHELL=y

# Shell configuration
CONFIG_SHELL_PROMPT_UART="shell>"
CONFIG_SHELL_CMD_BUFF_SIZE=128
CONFIG_SHELL_HISTORY_BUFFER=512
CONFIG_SHELL_VT100_COLORS=n
CONFIG_SHELL_VT100_COMMANDS=n
CONFIG_SHELL_CMDS=y
CONFIG_THREAD_MONITOR=y

# Important: Disable shell log backend to prevent logs from appearing on shell
CONFIG_SHELL_LOG_BACKEND=n

# Disable colored log output
CONFIG_LOG_BACKEND_FORMAT_TIMESTAMP=y
CONFIG_LOG_BACKEND_SHOW_COLOR=n
```


©2025 Beningo Embedded Group, LLC. All Rights Reserved.

The Code

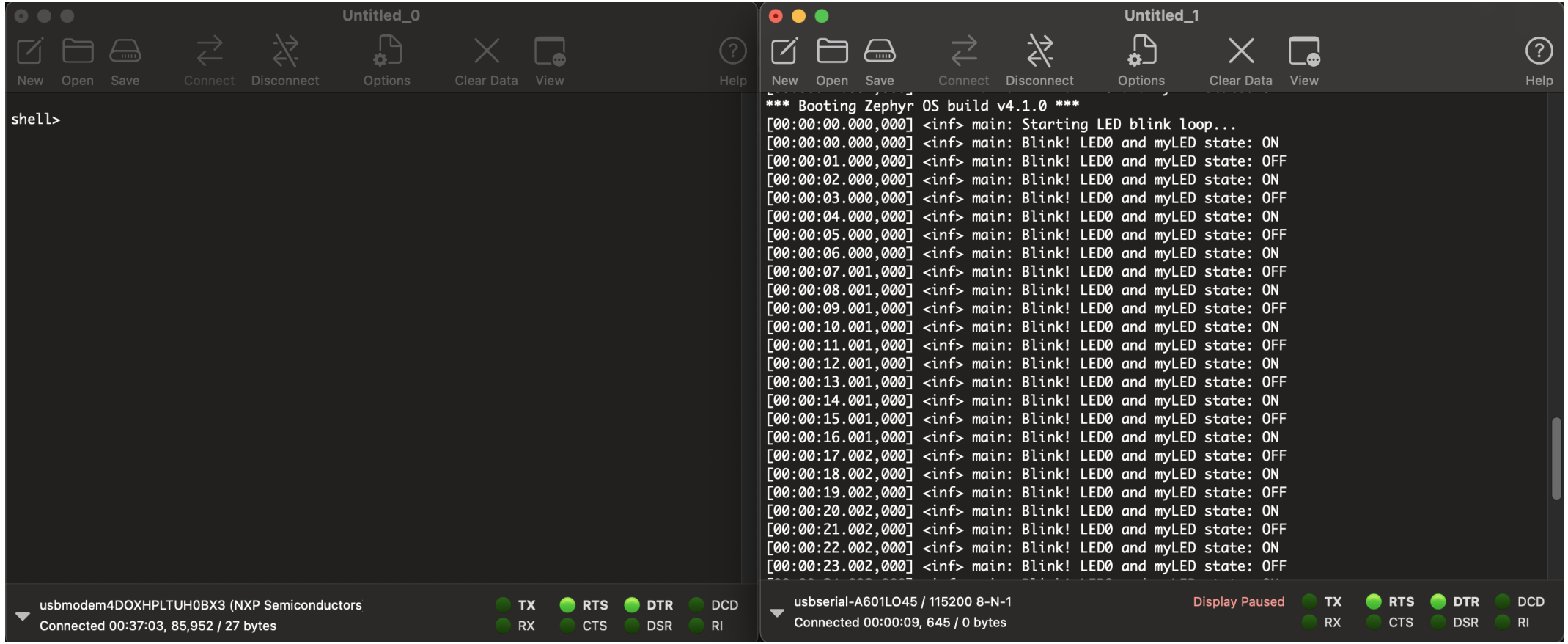
```c
#include <zephyr/kernel.h>
#include <zephyr/drivers/gpio.h>
#include <zephyr/logging/log.h>
#include <zephyr/shell/shell.h>

LOG_MODULE_REGISTER(main);  // Register the logging module
```

```c
int main(void)
{
    int ret;
    bool led_state = true;

    if (!gpio_is_ready_dt(&led0)) {
        LOG_ERR("led0 device not ready");
        return 0;
    }

    if (!gpio_is_ready_dt(&my_led)) {
        LOG_ERR("myLED device not ready");
        return 0;
    }
```

```c
    LOG_INF("Starting LED blink loop...");

    while (1) {
        gpio_pin_set_dt(&led0, led_state);
        gpio_pin_set_dt(&my_led, led_state);

        LOG_INF("Blink! LED0 and myLED state: %s", led_state ? "ON" : "OFF");

        led_state = !led_state;
        k_msleep(SLEEP_TIME_MS);
    }

    return 0;
```


©2025 Beningo Embedded Group, LLC. All Rights Reserved.

Result



©2025 Beningo Embedded Group, LLC. All Rights Reserved.

Audience POLL Question

What is the most important configuration to ensure your log messages don't end up on all your UART devices?

a) CONFIG_LOG_MODE_DEFERRED=y
b) CONFIG_SHELL_LOG_BACKEND=n
c) CONFIG_SHELL_LOG_BACKEND=n
d) CONFIG_LOG_BACKEND_RTT=n

©2025 Beningo Embedded Group, LLC. All Rights Reserved.

Next Steps

04

©2025 Beningo Embedded Group, LLC. All Rights Reserved.

Going Further

Download the extra resources:

- https://beningo.short.gy/jVGeiDNZephyrLP

Zephyr Documentation:

- Tracing Documentation
- Logging Documentation

Downloadable Resources:

- West Cheat Sheet (High-Resolution)
- RTOS Performance Guide
- Application Code Examples
- Zephyr Docker Container
- VS Code Debug Launch Script

©2025 Beningo Embedded Group, LLC. All Rights Reserved.

Additional Resources

Please consider the resources below:

- Jacob's Blogs
- Jacob's CEC courses
- Embedded Software Academy

- Embedded Bytes Newsletter
  - http://bit.ly/1BAHYXm

Consulting | Coaching | Training

www.beningo.com

©2025 Beningo Embedded Group, LLC. All Rights Reserved.

Next Steps

- Welcome to Zephyr RTOS
- Build Systems, Kconfig, and Device Tree
- Threads, Scheduling, and RTOS Primitives
- Drivers, Peripherals, and Customization
- Debugging, Logging, and Best Practices

©2025 Beningo Embedded Group, LLC. All Rights Reserved.

CEC Continuing Education Center

DesignNews

Thank You

Sponsored by

DigiKey

BENINGO EMBEDDED GROUP

informa markets