CEC Continuing Education Center

DesignNews

Mastering Zephyr RTOS

DAY 1 : Welcome to Zephyr RTOS

Sponsored by

DigiKey

BENINGO EMBEDDED GROUP

informa markets

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.

CEC Continuing Education Center

BENINGO EMBEDDED GROUP

DigiKey

THE SPEAKER

Jacob Beningo

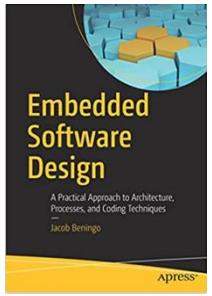
Jacob@beningo.com

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

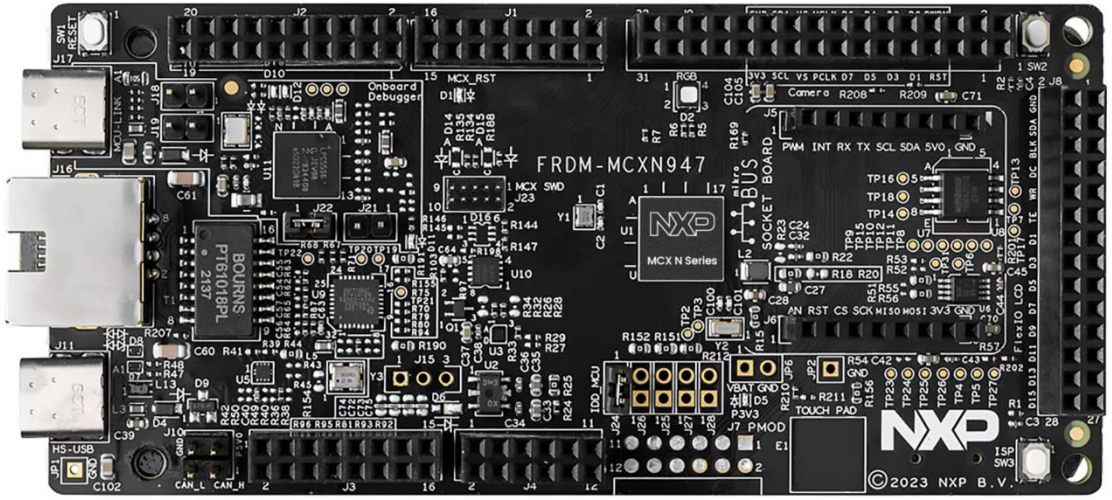
Visit **www.beningo.com** to learn more

This week's topics:

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

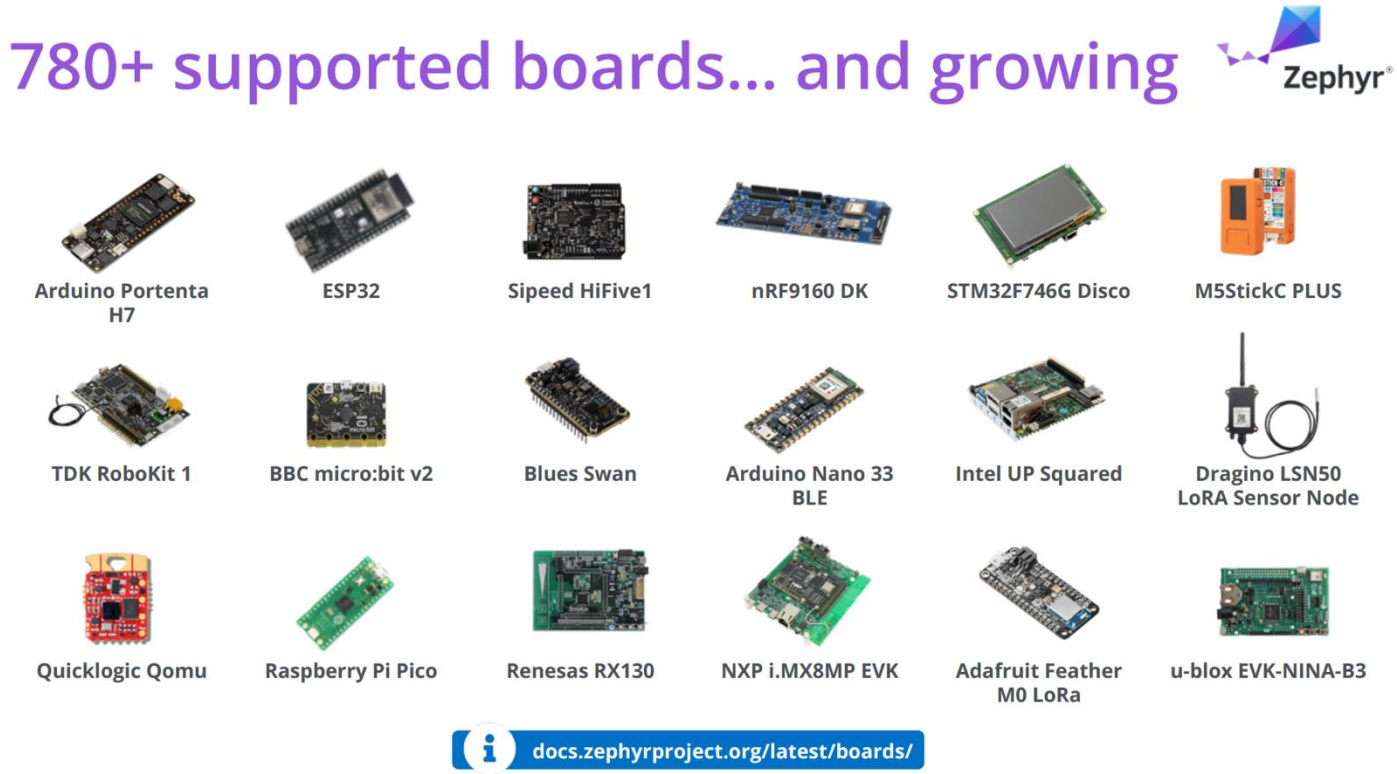
NXP FRDM-MCXN947

- MCX-N947 Dual Arm Cortex-M33 microcontroller running at 150 MHz
- 2MB dual-bank on chip Flash
- 512 KB RAM
- External Quad SPI flash over FlexSPI
- USB high-speed (Host/Device) with on-chip HS PHY. HS USB Type-C connectors
- 10x LP Flexcomms each supporting SPI, I2C, UART
- 2x FlexCAN with FD, 2x I3Cs, 2x SAI
- 1x Ethernet with QoS
- On-board MCU-Link debugger with CMSIS-DAP
- Arduino Header, FlexIO/LCD Header, SmartDMA/Camera Header, mikroBUS



Zephyr Board Support

https://docs.zephyrproject.org/latest/boards/index.html#



Zephyr RTOS

01

What is Zephyr?

An open-source RTOS (Apache 2.0)

- Linux Foundation supported
- Built with safety and security in mind
- Lightweight kernel with supporting services
  - Connectivity stacks like Wi-Fi, BLE, USB, MQTT, etc
- Developer Friendly
  - Logging, tracing, built-in-shell, Windows/Linux/macOS
- Ships with SBOMs (Software Bill of Materials)
- Supports more than 780 boards

- Additional Resources: Zephyr Project Overview Document

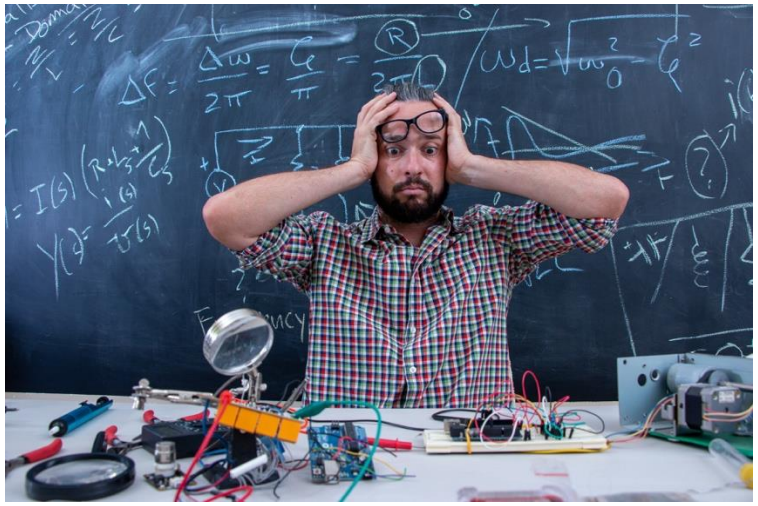
The Zephyr Community

- Hardware Engineers

Can get up and running quickly without being software experts.

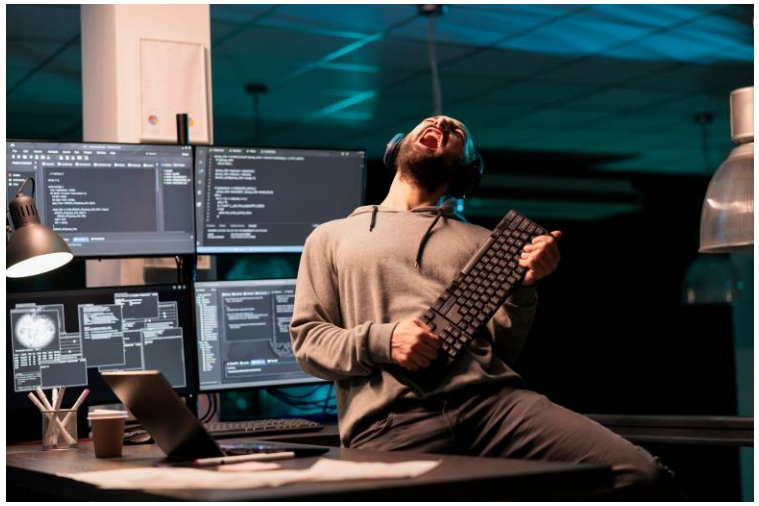
(Board Check-out)

- Firmware Engineers



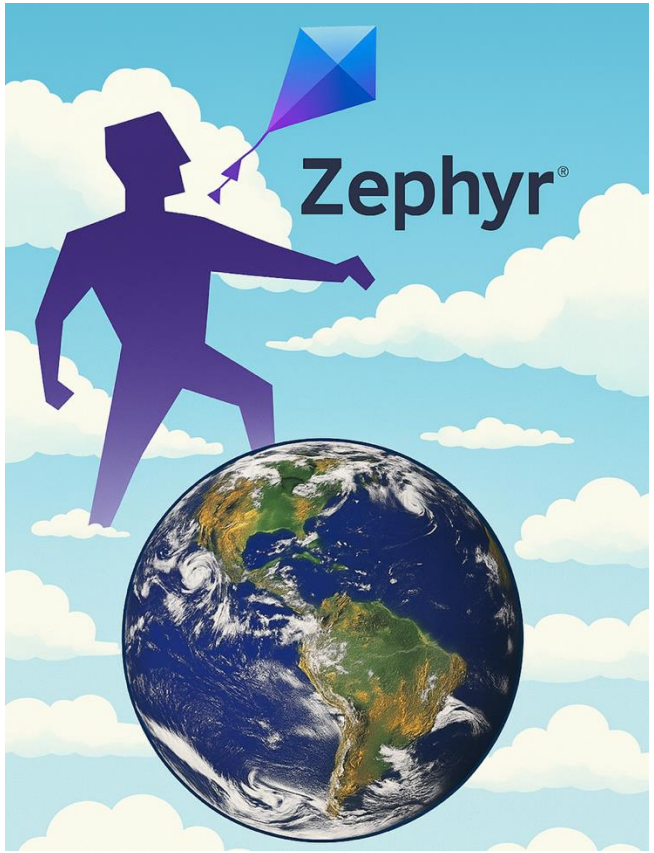
Work up and down the software stack including low-level including DTS, drivers, subsystems, etc.

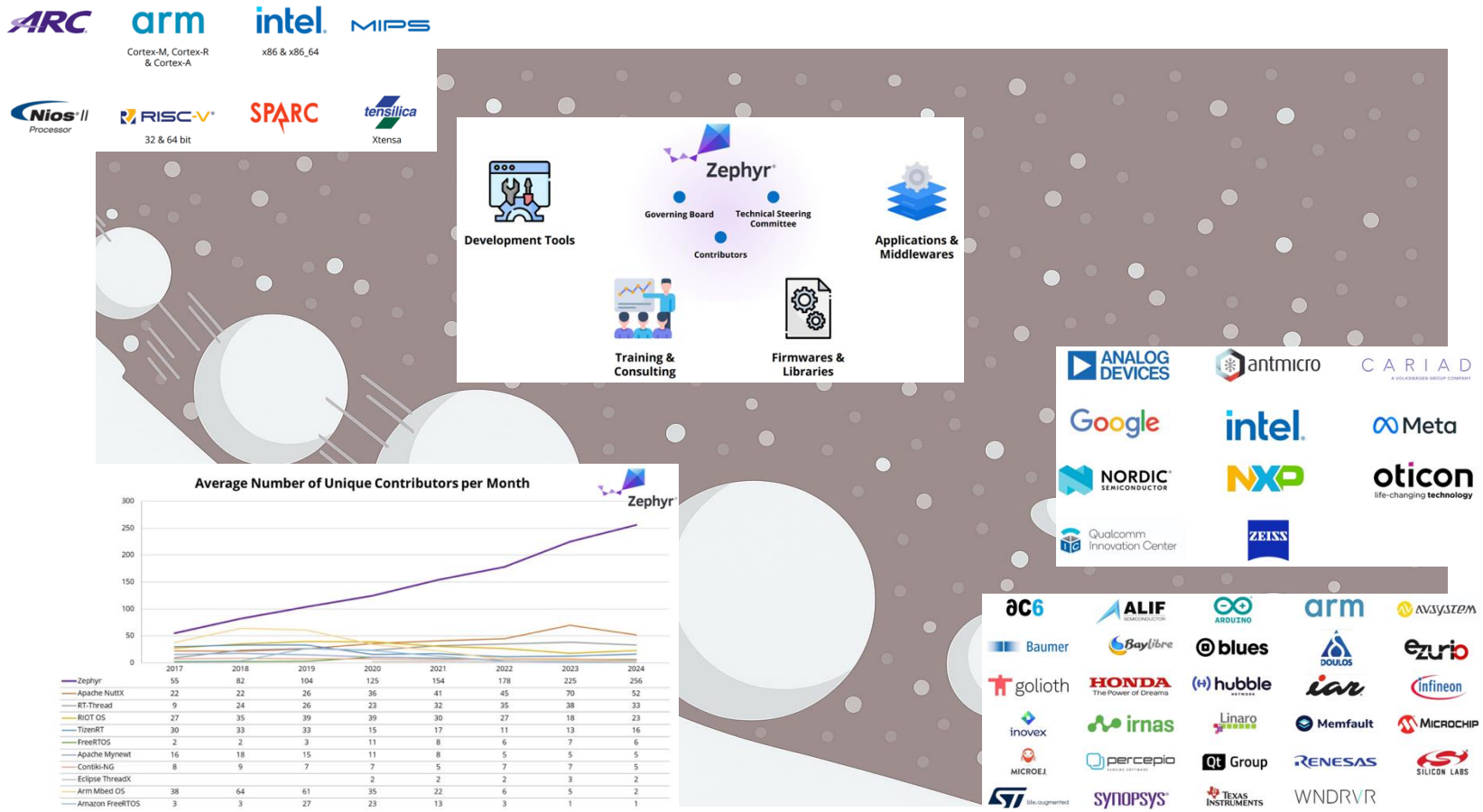
- Software Engineers



Treat Zephyr like a Linux system and can easily get up to speed developing real-time systems.

Zephyr Will Rule the World . . .





Audience POLL Question

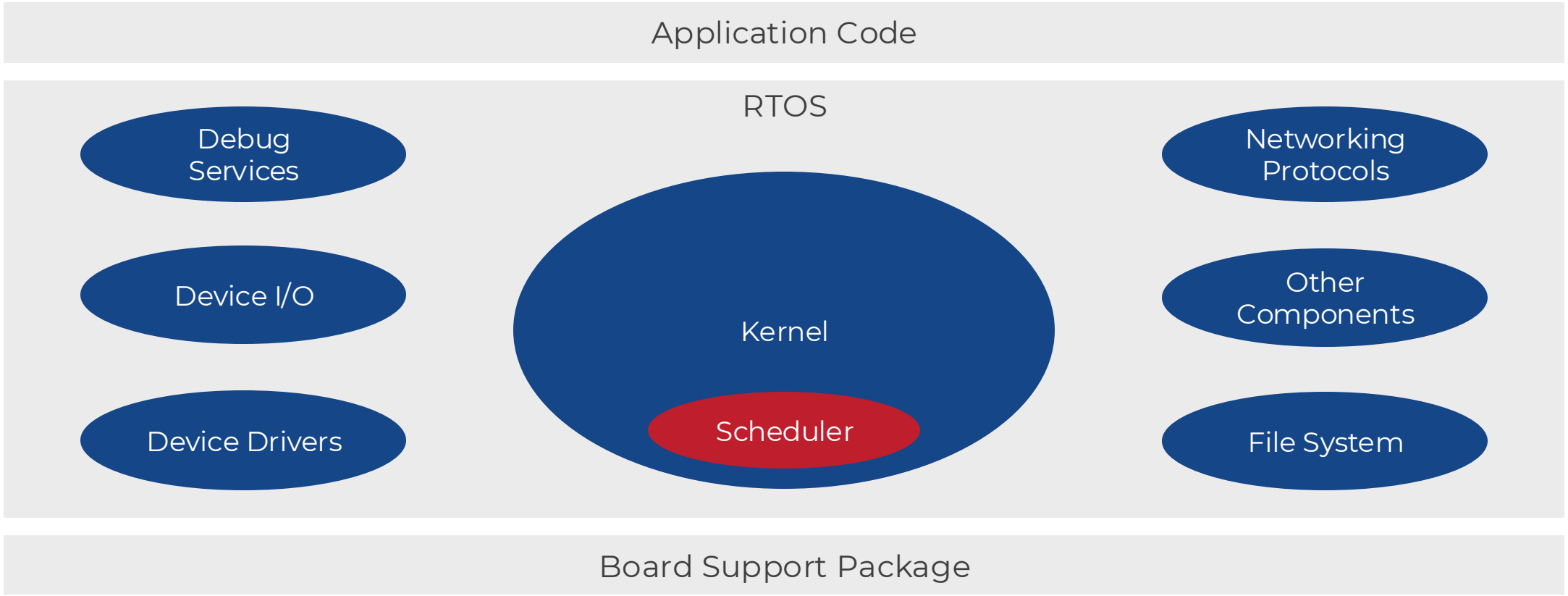
Which of the following best describes you?

a) Never used Zephyr before
b) Have installed and run “Hello World”
c) Played around a bit beyond the basics
d) Using Zephyr daily for work

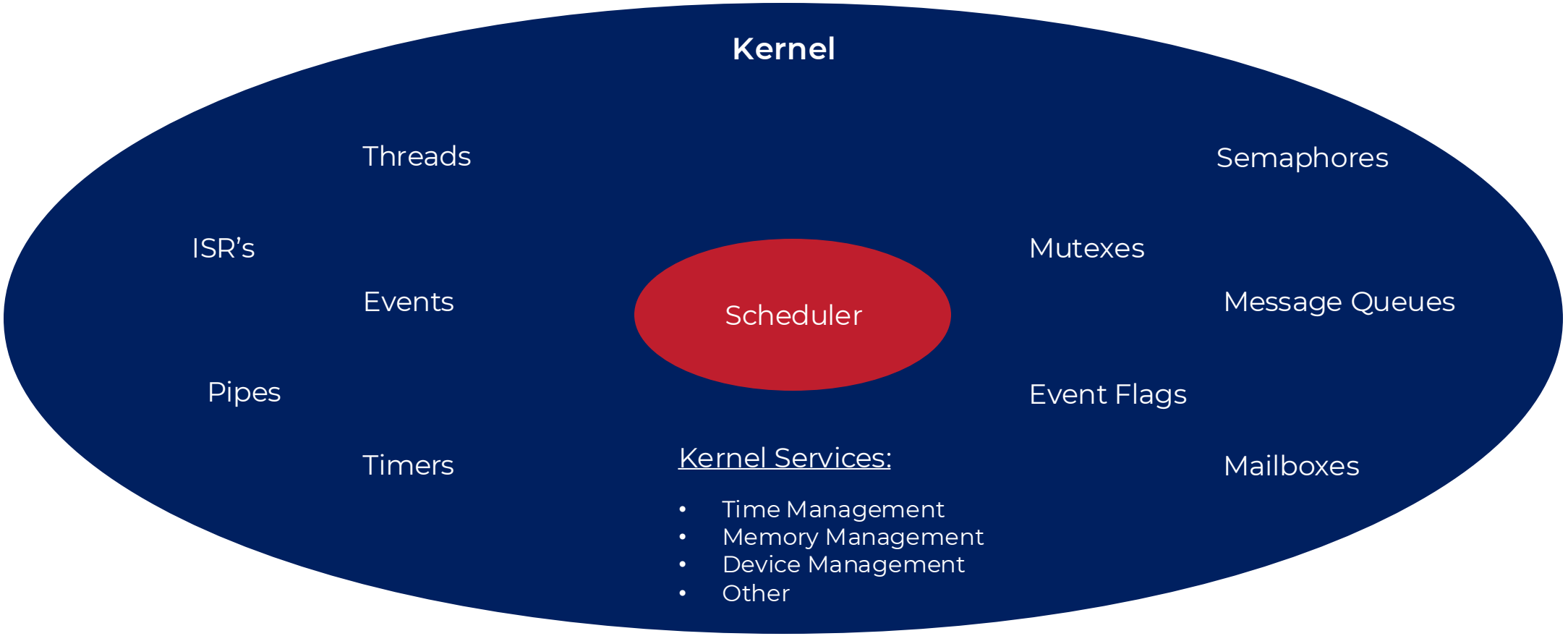
The Zephyr Kernel

01

Real-time Operating Systems

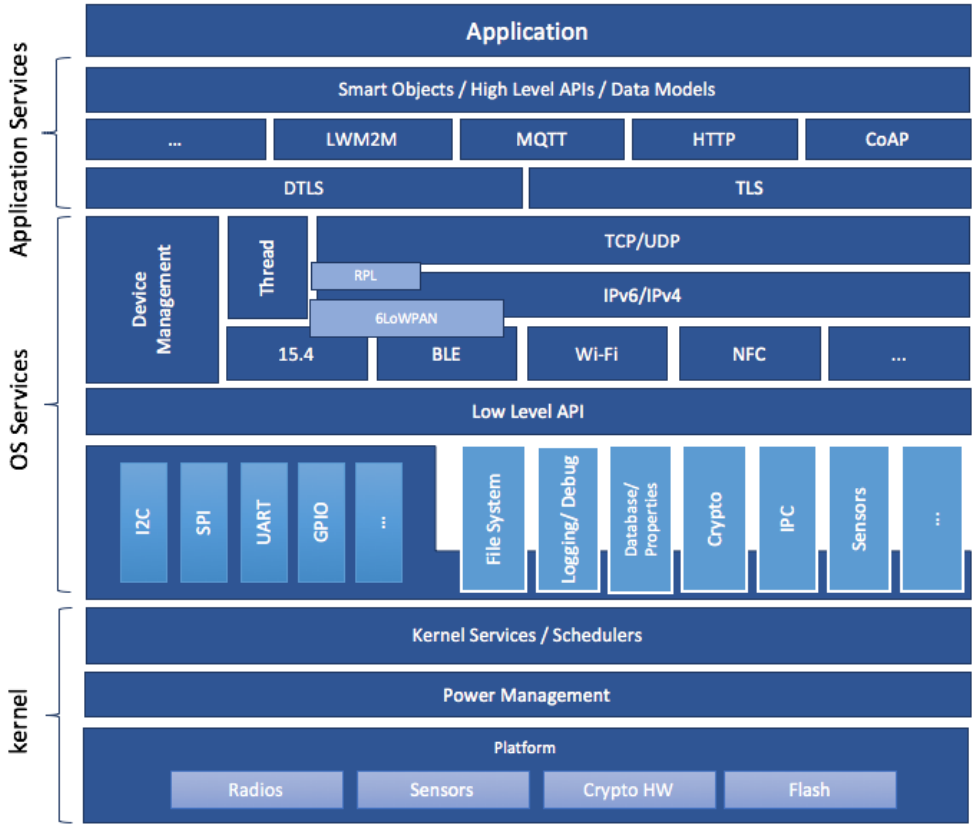


What is a Kernel

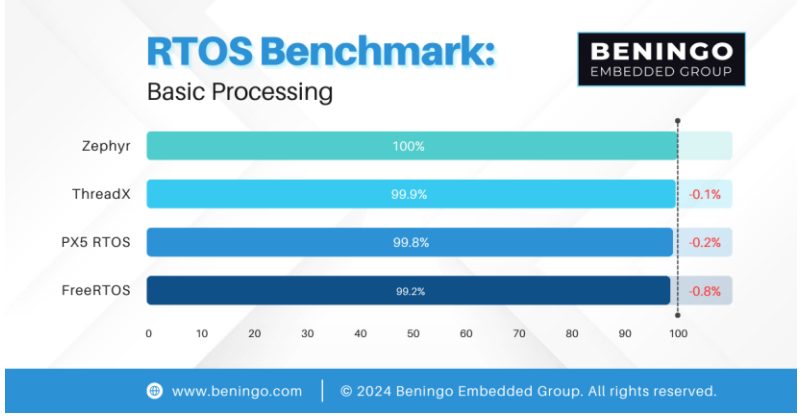


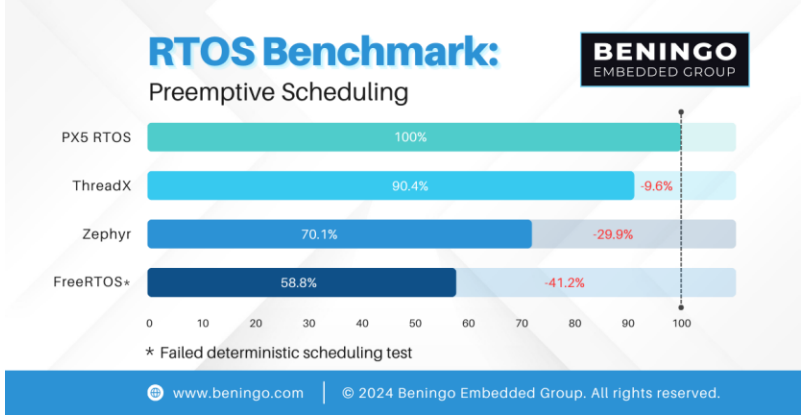
Zephyr Architecture

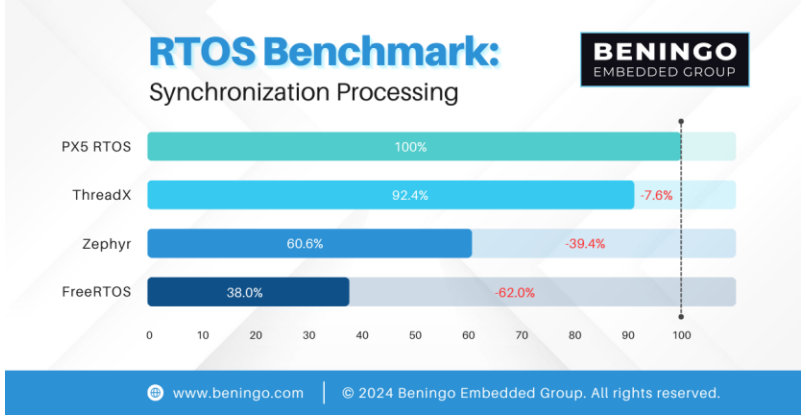
- Kernel
  - Light-weight, real-time kernel
  - Thread Management
  - Power Management
- OS-Services
  - Memory Management
  - Connectivity Stacks
  - Device Drivers
- Application Services
  - Logging and Tracing
  - Internet Protocols

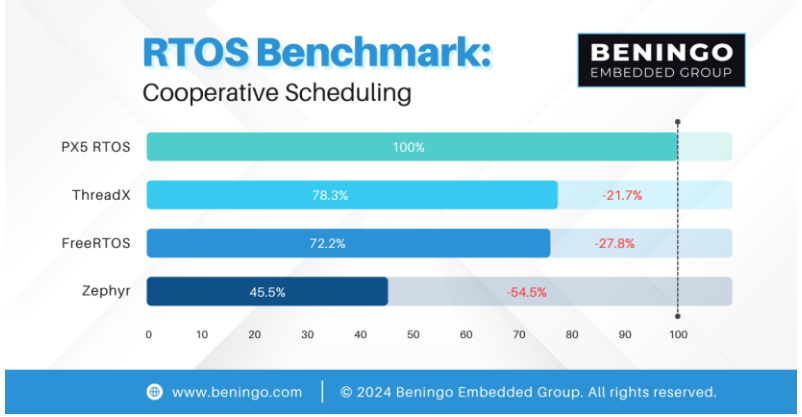


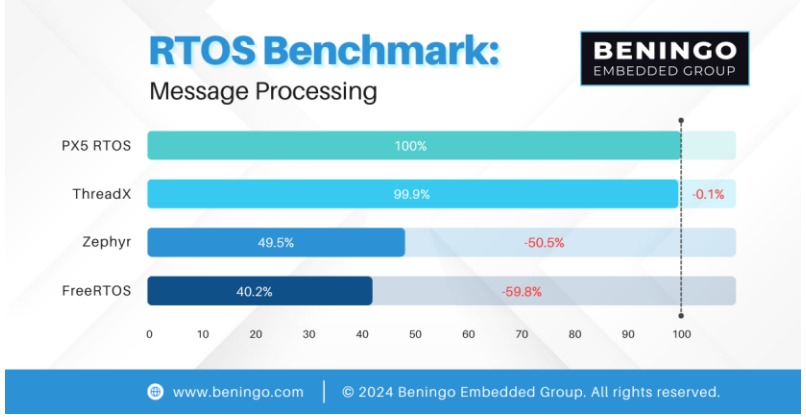
Kernel Performance – 2024 Zephyr 3.7

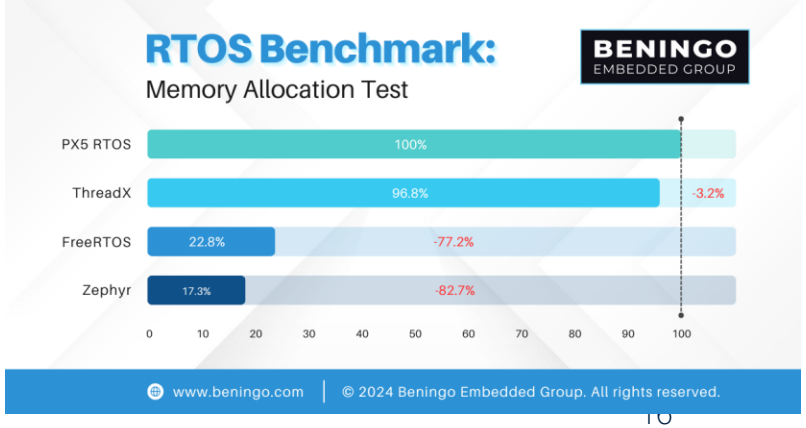










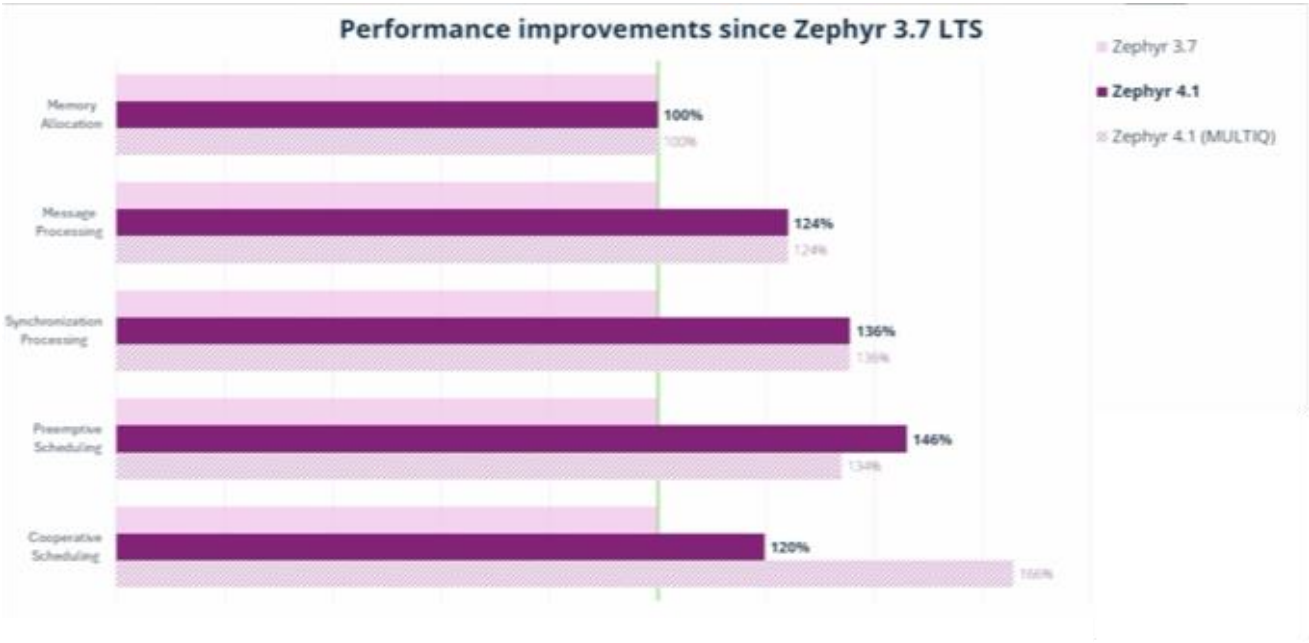


Kernel Performance – Zephyr 4.1

Major Performance Increases

- 46% Faster Preemptive Scheduling
- 36% Faster Synchronization
- 24% Faster Message Processing

Experimental MUTLIQ scheduling



source: https://www.youtube.com/watch?v=TOIwI9XrHZM

Audience POLL Question

How important is kernel performance to your application?

a) Critical – We have hard real-time constraints, and even microseconds matter.
b) Important – Latency and responsiveness affect our user experience or power efficiency.
c) Somewhat important – We care, but the kernel isn't usually our bottleneck.
d) Not important – Our app is simple, and kernel overhead isn’t a concern.

Installing Zephyr

03

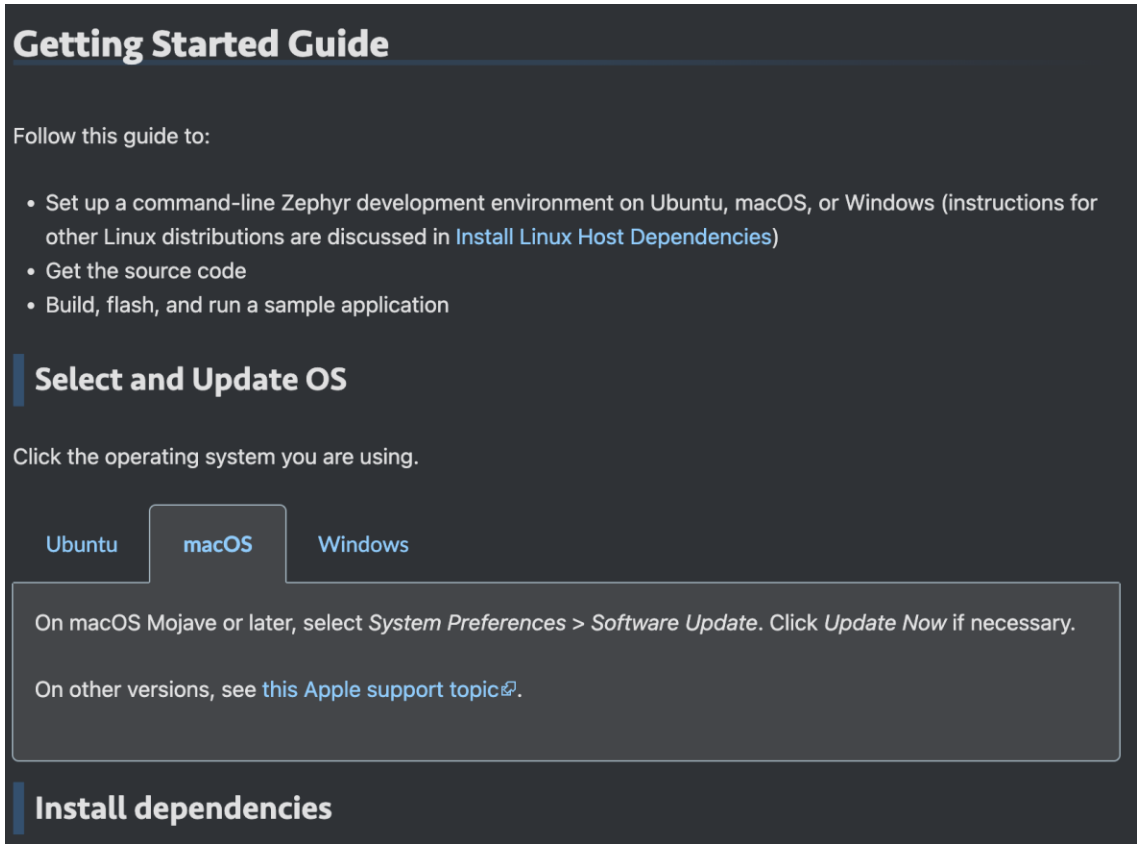
Zephyr is Self-Contained

- Use “West” to download everything
- Support for:
  - CPUs
  - Boards
  - Bootloader
  - Subsystems
  - Peripherals
- Drivers are vetted
  - Thriving open-source community with maintainers
  - Zephyr Discord Channel

Installing Zephyr with West

https://docs.zephyrproject.org/latest/develop/getting_started/index.html

- Select OS
- Install Dependencies
  - Cmake
  - Python
  - Devicetree Compiler
- Get Zephyr and install Python Dependencies
- Install the Zephyr SDK
- Build the Blinky Sample



Installing Zephyr with West - Windows

Tool	Min. Version
CMake	3.20.5
Python	3.10
Devicetree compiler	1.4.6

cd $Env:HOMEPATH python -m venv

zephyrproject\.venv

zephyrproject\.venv\Scripts\Activate.ps1

Install and Configure West

pip install west

west init zephyrproject

cd zephyrproject

west update

west zephyr-export

west packages pip --install

Install Zephyr SDK

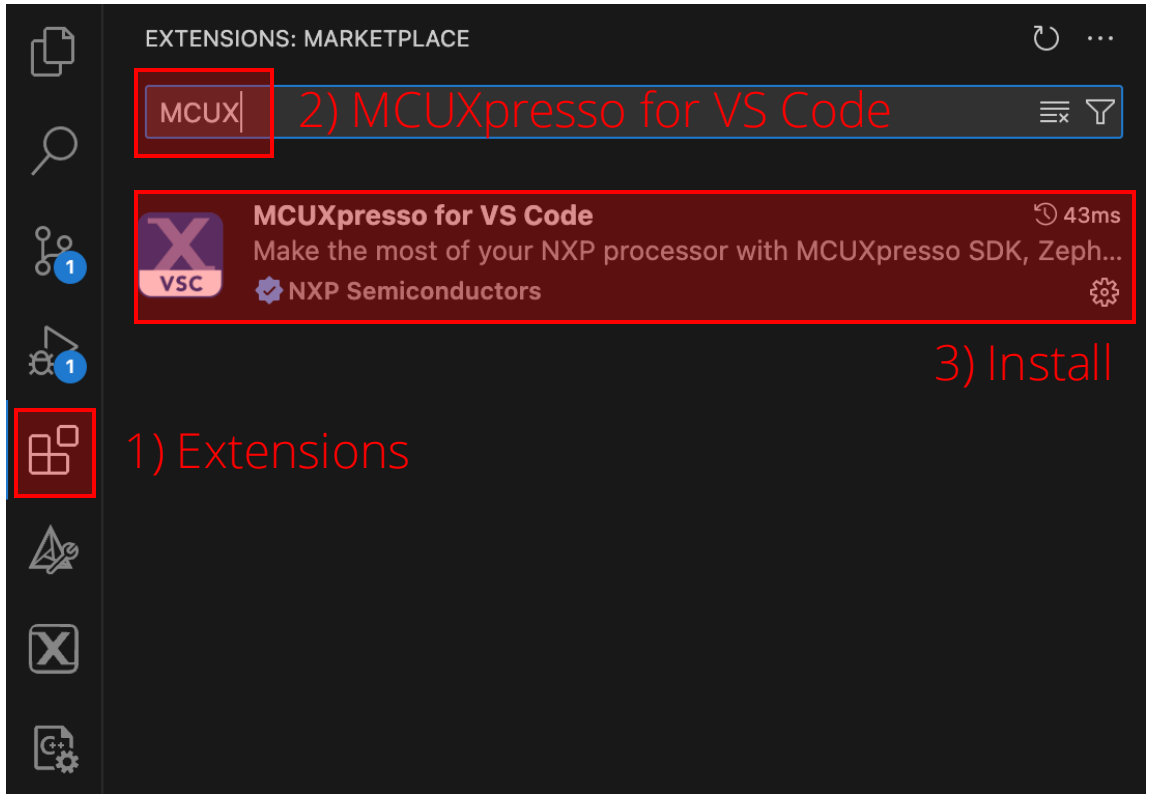
cd $Env:HOMEPATH\zephyrproject\zephyr

west sdk install

Installing Zephyr with Vendor Tools

Visual Studio Code offers an integrated environment to streamline development.

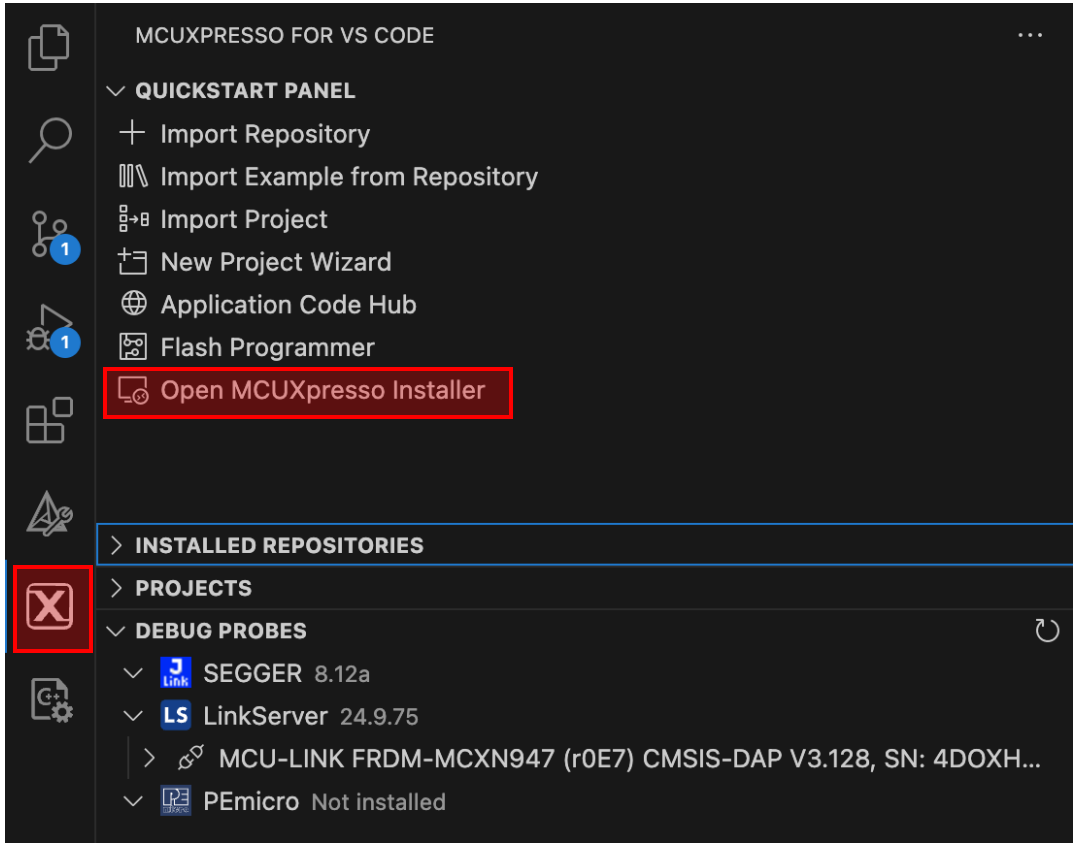
- Customizable with extensions
- Cross-Platform
- Simplified build tools
- Terminal integration
- etc



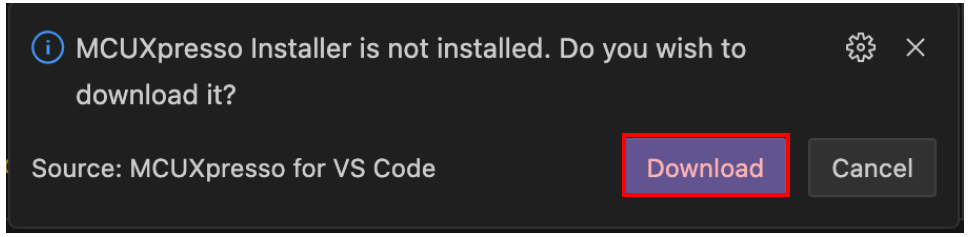
Installing Zephyr with Vendor Tools

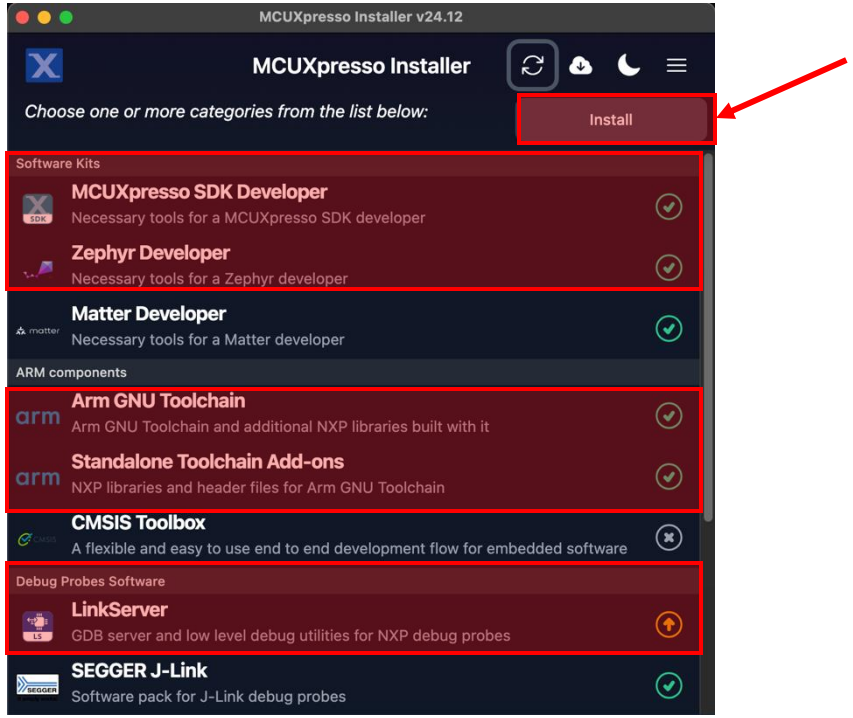
- Installing MCUXpresso Tools

1)



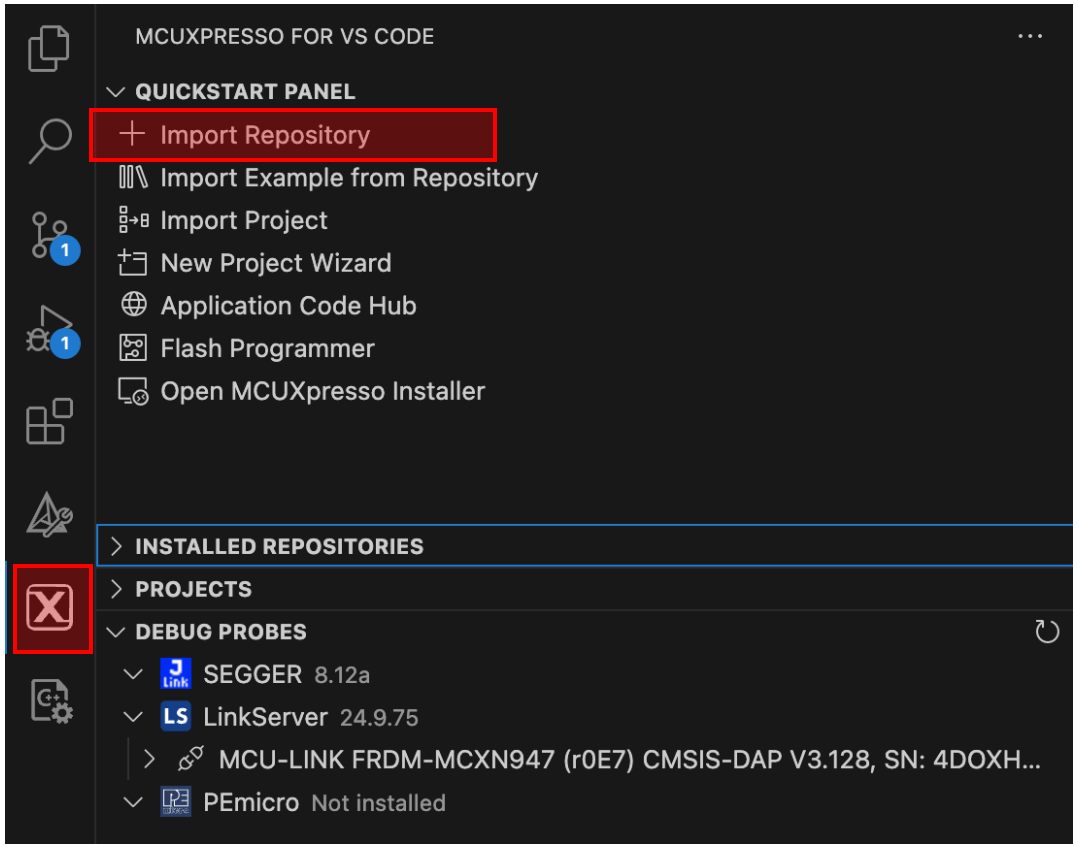
2)

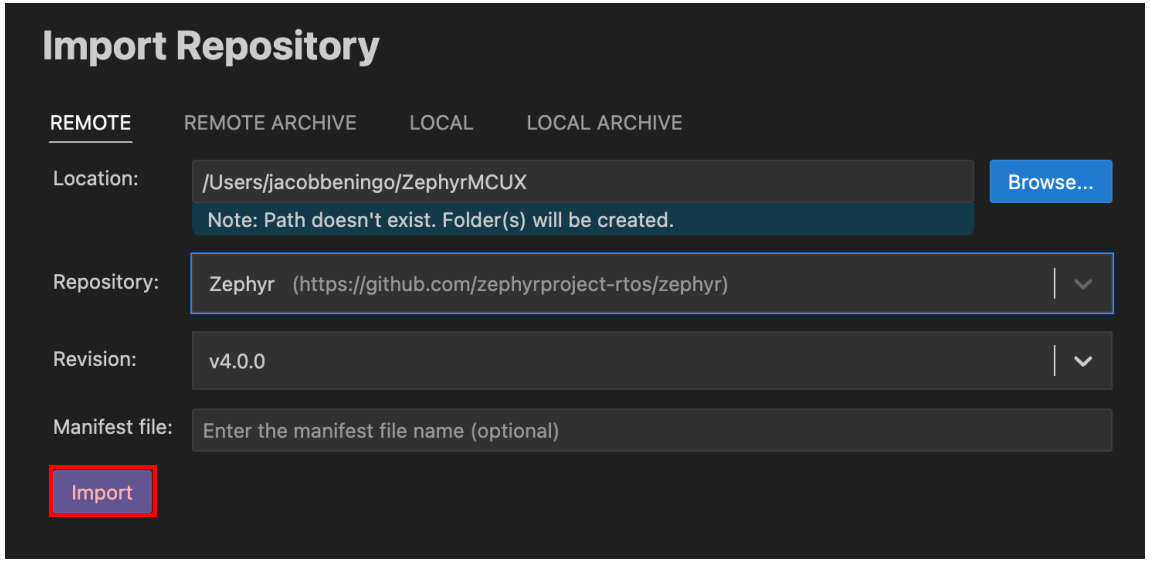




Installing Zephyr with Vendor Tools – Importing Zephyr

1)





Audience POLL Question

Which Zephyr workflow to you think you prefer?

a) Using vendor-provided tools to install and manage
b) Directly using west and the command line
c) I'm not sure yet, too new at this
d) I like to use both

Next Steps

04

Going Further

Download the extra resources:

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

Zephyr Documentation:

- Zephyr Getting Started Documentation
- Getting Started with Zephyr RTOS: Hello Blinky! (Webinar)

Downloadable Resources:

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

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

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

CEC Continuing Education Center

DesignNews

Thank You

Sponsored by

DigiKey

BENINGO EMBEDDED GROUP

informa markets