



Eclipse ThreadX

A recipe for success with four simple ingredients

Frédéric Desbiens
Senior Manager — Embedded and IoT
November 11, 2025

Photo by [Becca Tapert](#) on [Unsplash](#)



Remembrance Day 2025



Canadian National Vimy Memorial — Vimy, France

Photo credit: Commonwealth war graves commission

In Flanders Fields

*In Flanders fields, the poppies blow
Between the crosses, row on row,
That mark our place; and in the sky
The larks, still bravely singing, fly
Scarce heard amid the guns below.*

*We are the Dead. Short days ago
We lived, felt dawn, saw sunset glow,
Loved and were loved, and now we lie,
In Flanders fields.*

*Take up our quarrel with the foe:
To you from failing hands we throw
The torch; be yours to hold it high.
If ye break faith with us who die
We shall not sleep, though poppies grow
In Flanders fields.*

John McCrae — December 8, 1915

About Me

Frédéric Desbiens

Senior Manager — Embedded and IoT

B.Ed., B.Sc.A, MBA

Developer, Architect, Product Manager...

Oracle, Cisco, Pivotal...

Published author; Frequent Speaker

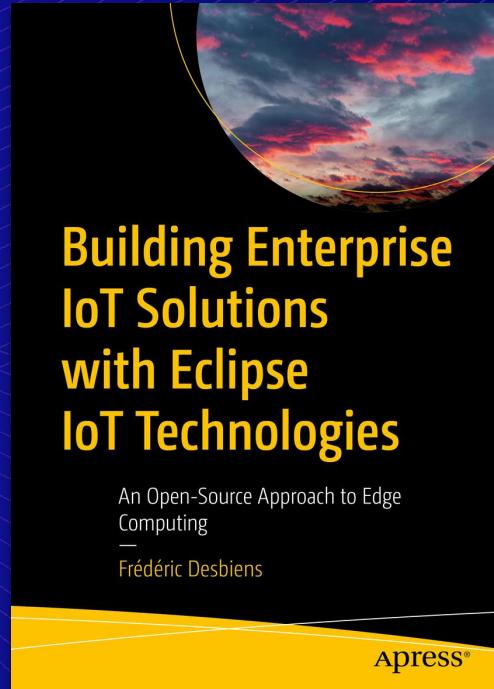
<https://linkedin.com/in/fredericdesbiens>



A Comprehensive Overview

A comprehensive overview of the open-source IoT and Edge Computing platforms available at the Eclipse Foundation

ISBN: 978-1484288818

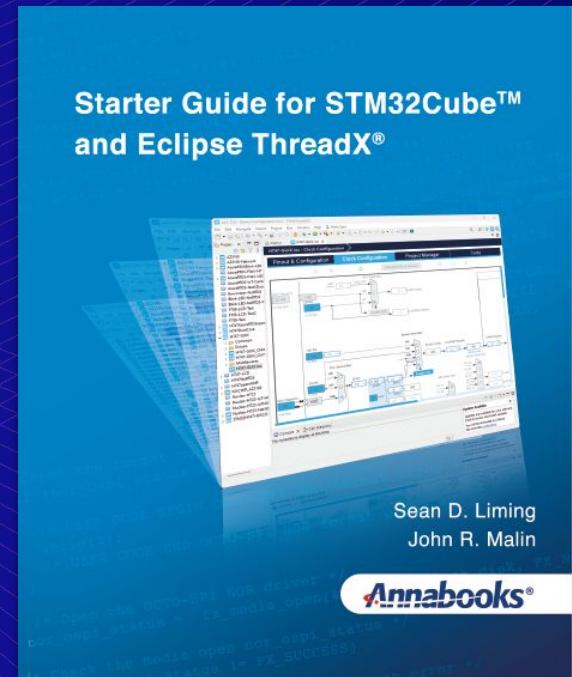


Suggested Reading

A deep dive into the STM32Cube tools
from STMicroelectronics and Eclipse
ThreadX

Published in August 2024

ISBN: 979-8-9854172-3-4



Meet Eclipse ThreadX

A Small and Fast Open Source Real-Time Operating System Certified for Safety-Critical Applications

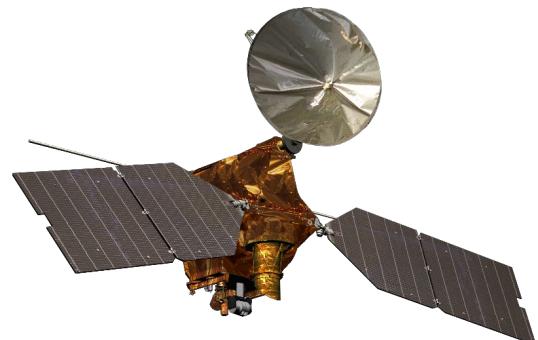


ECLIPSE
THREADX

12 Billion*

devices worldwide run ThreadX

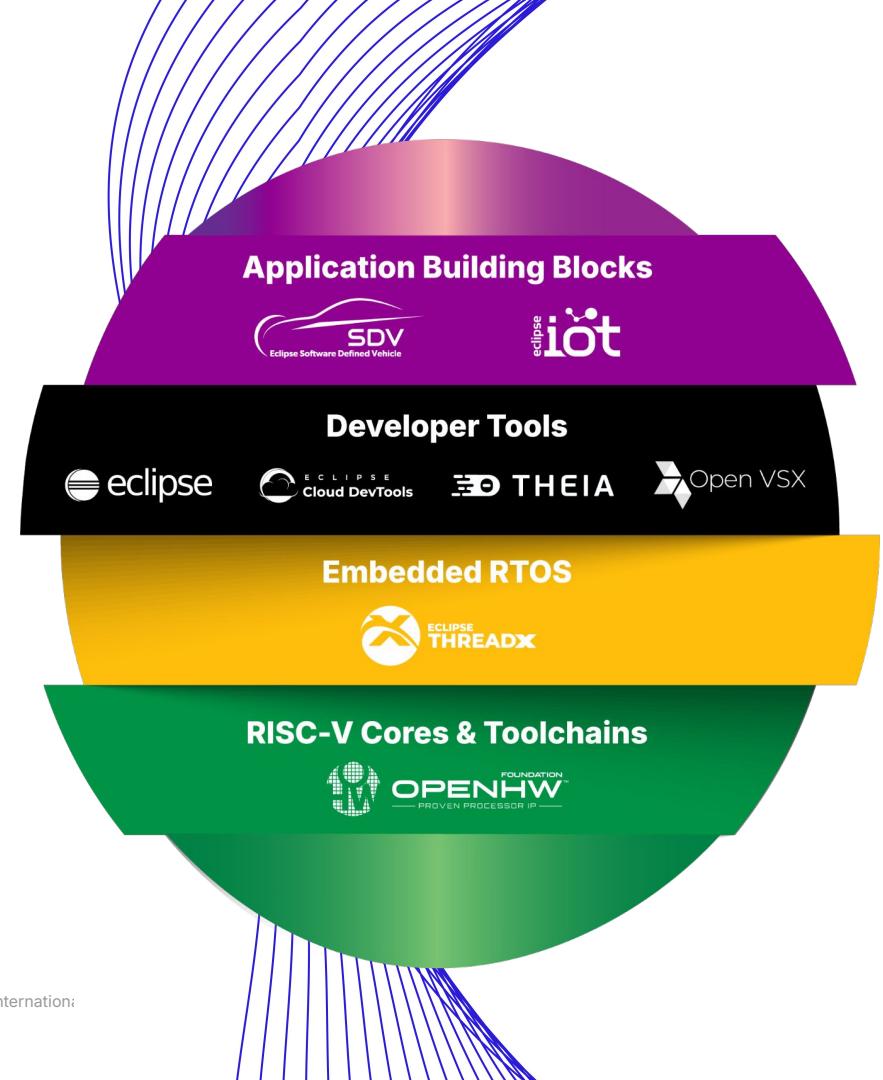
* ... and counting



The Eclipse RISC-V Ecosystem

The Only Comprehensive
Open Source RISC-V Stack for
Embedded Applications

Vendor-Neutral
Community-Driven
Sustainable



What is a RTOS?

A Real-Time Operating system is system software that provides services and manages processor resources for applications. These resources include processor cycles, memory, peripherals, and interrupts.

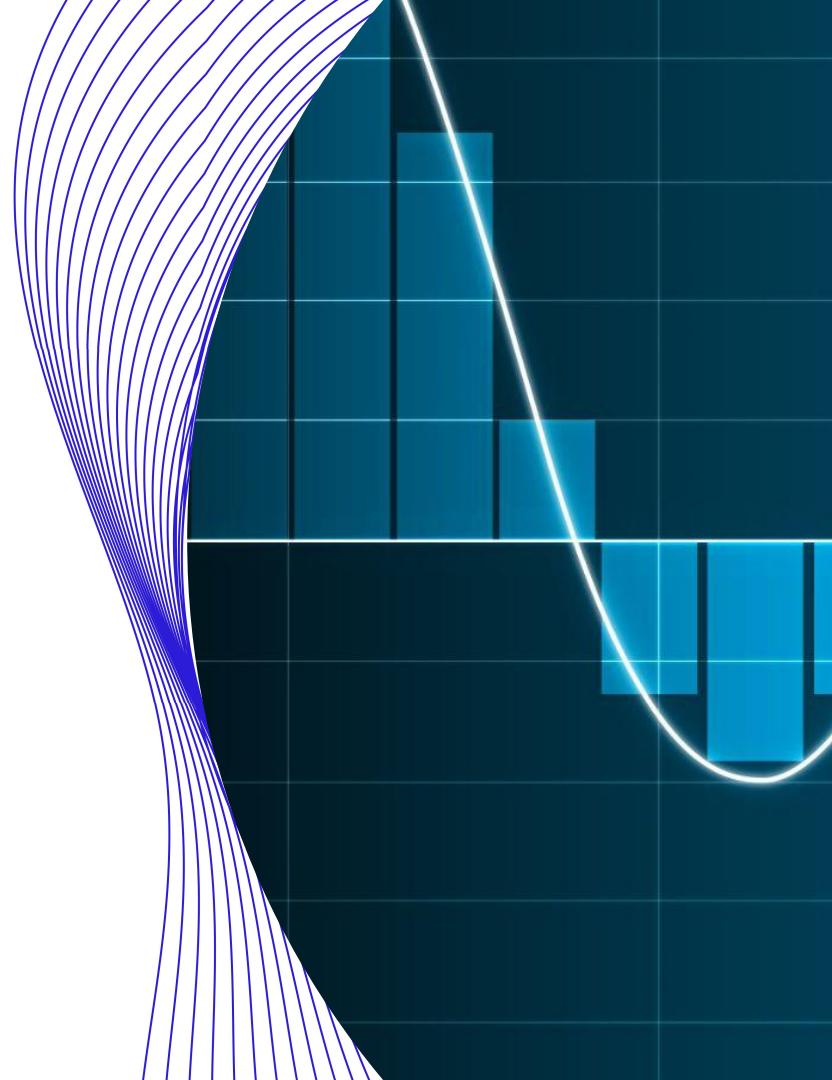
The main purpose of a Real-Time Operating System is to allocate processing time among various duties the embedded software must perform.



Photo by Emily Merton on [Unsplash](#)

RTOS Characteristics

- Small and fast
- Dedicated purpose
- Real-time, deterministic processing requirements
- Hardware target: Typical MCU
 - < 512KB memory
 - < 200MHz
 - 32-bit
- MMU not required



What Developers Want



The ability to execute multiple threads concurrently

typically with a single processing core for fast, real-time processing.



Flexible ways to execute threads

based on priority and system conditions and temporarily suspend tasks as needed.



Full capabilities in spite of constrained resources

with a small memory footprint.

ThreadX History



1997

ThreadX

Express Logic

Proprietary

2019

Azure RTOS

Microsoft

Open code with restrictive
license

2023

Eclipse ThreadX

Eclipse Foundation

MIT License

ThreadX Deep Dive

Eclipse ThreadX System Components



ThreadX

High-performance real-time operating system
(SIL 4, ASIL D, Med. Class C)



NetX Duo

IPv4/IPv6 stack with TLS/DTLS support
(SIL 4, ASIL D, Med. Class C)



FileX

Embedded FAT filesystem with optional fault tolerance



GUIX

Design environment and runtime for 2D user interfaces
(SIL 4, ASIL D, Med. Class C)



USBX

USB Stack providing host, device and on-the-go support
(SIL 4, ASIL D, Med. Class C)



TraceX

Graphical view of real-time events enabling analysis of system-level behavior

Safety Certifications



FUNKTIONALE SICHERHEIT
GEPRÜFT
FUNCTIONAL SAFETY
APPROVED



Certified by SGS-TÜV Saar
for use in safety-critical
systems, according to
IEC-61508 SIL 4, IEC-62304
SW Safety Class C, ISO
26262 ASIL D and EN 50128



Compliant with all
"required" and "mandatory"
rules of MISRA-C:2004 and
MISRA C:2012



Certified by UL for
compliance with UL 60730-1
Annex H, CSA E60730-1
Annex H, IEC 60730-1 Annex
H, UL 60335-1 Annex R, IEC
60335-1 Annex R, and UL
1998



ThreadX

- **Small:** 2KB minimal footprint
- **Fast:** Sub-microsecond context switches
- **Easy:** Consistent API
- **Safe:** SIL 4, ASIL D
- **Advanced:** Preemption-threshold, event chaining, auto scaling, modules with memory protection

ThreadX API

Thread Services

Messaging Queues

Counting Semaphores

Mutexes

Event Flags

Block Memory Pools

Byte Memory Pools

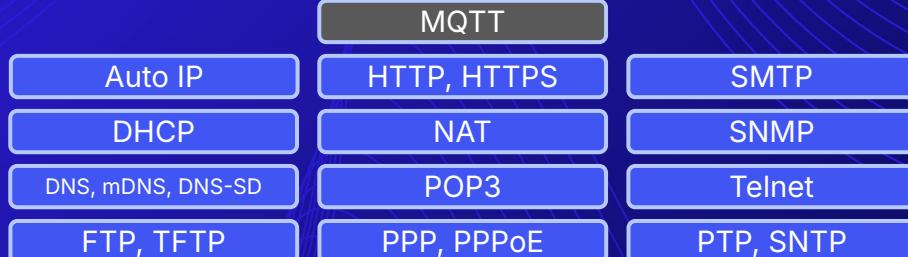
Application Timers

ThreadX Core Scheduler

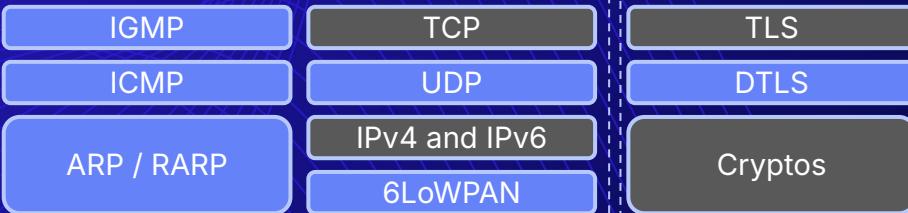
NetX Duo

- **Small:** 50KB Device-to-Cloud
- **Fast:** Near wire speed, minimal CPU usage
- **Security:** Extensive pen testing, EAL 4+, FIPS 140-2
- **Easy:** Consistent API
- **Safe:** SIL 4, ASIL D
- **Advanced:** Extensive components, zero copy, auto scaling

Addons



Core



Security

Ethernet, Wi-Fi, Cellular, 802.15.4, ...

ThreadX

FileX

- **Small:** 9KB minimal footprint
- **Fast:** Direct data writes, cache optimized for speed
- **Easy:** Consistent API
- **Advanced:** Support for FAT 12/16/32, fault tolerance, extensive caching support, cluster pre-allocation, NAND/NOR wear leveling (LevelX), auto scaling



FileX API

Media Services

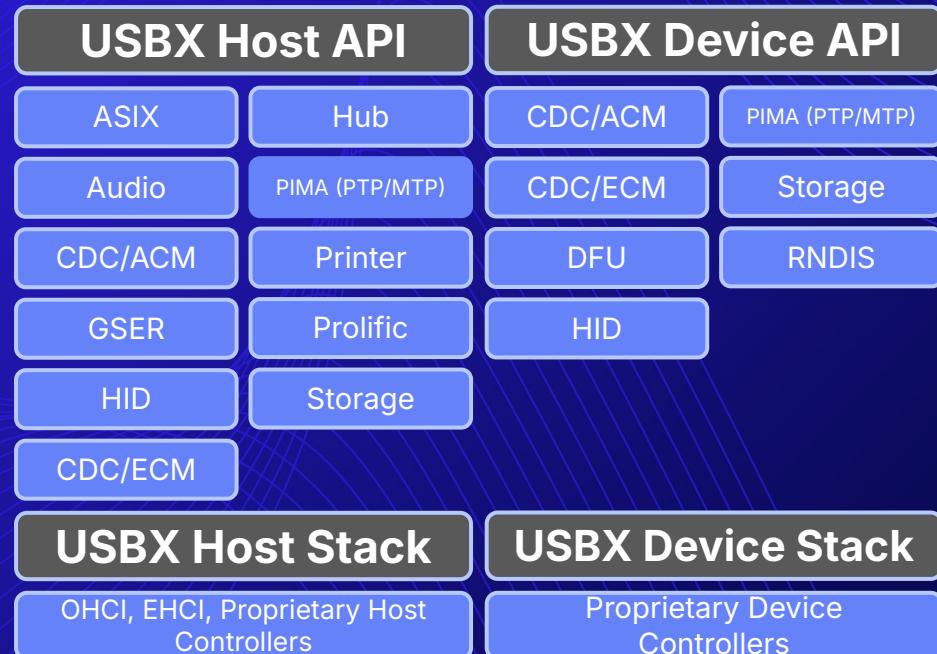
Directory Services

File Services

LevelX (NOR/NAND), RAM Disk, USBX, SD CARD, ...

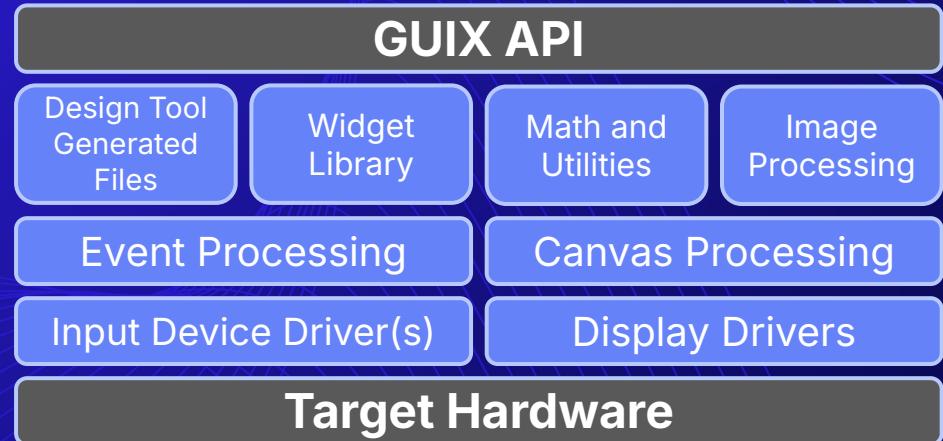
USBX

- **Small:** footprint of 8.5KB for devices and 12KB for hosts
- **Fast:** Leverages DMA, minimal function call layering
- **Easy:** Consistent API, device/host controller integration
- **Safe:** SIL 4, ASIL D
- **Advanced:** Comprehensive class support

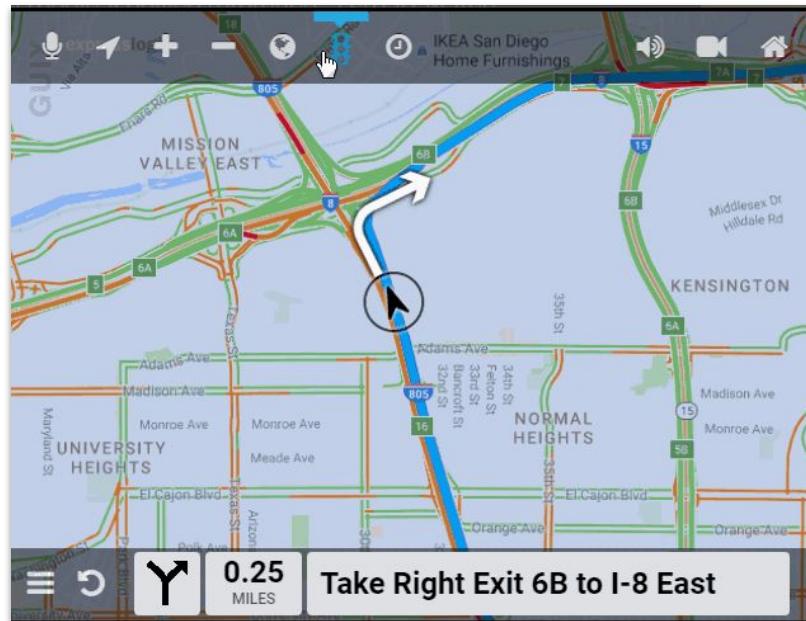
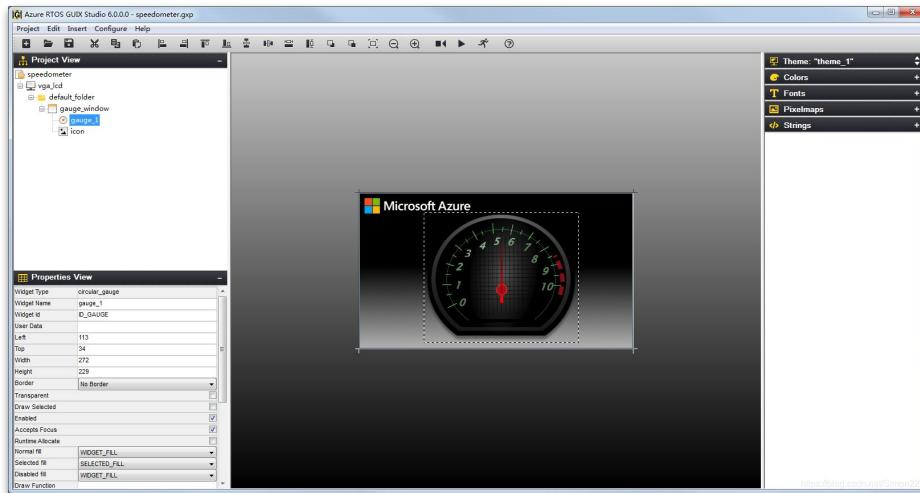


GUIX

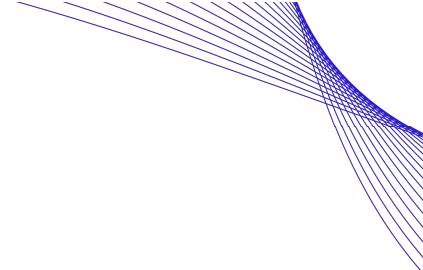
- **Small:** 13KB basic support
- **Fast:** Written in C, optimized clipping, drawing, and event handling
- **Easy:** Consistent API, graphics accelerator integration
- **Safe:** SIL 4, ASIL D
- **Advanced:** WYSIWYG design tool, extensive widget collection, auto scaling



Eclipse ThreadX GUIX Studio



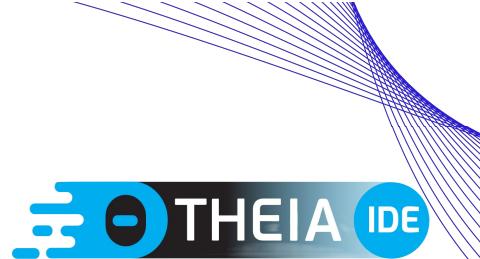
Getting Started



Developer Tools

- Linux, MacOS, or Windows workstation
- Appropriate C compiler
 - GNU Arm Embedded Toolchain works for most targets
- Required tooling
 - CMake
 - Ninja
- ThreadX has no third-party dependencies

How About an IDE?



Bring your own AI...

... or no AI at all!

The screenshot shows a C/C++ development environment with the following details:

- File Explorer:** Shows the project structure under "C:\tx_apl".
- Code Editor:** Displays the file "common\tx.h" with code related to ThreadX API definitions.
- Build Log:** Shows the output of the CMake build process, indicating successful compilation of "tx_port.h" and other files, and linking of "tx_thread.o" and "tx_thread.o" into "tx_apl.exe".



The Starter Application

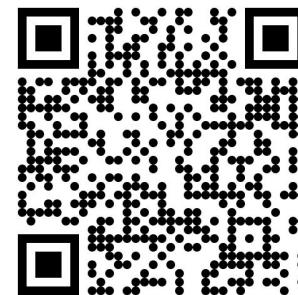
```
iot-devkit
|-- MXChip
|   '-- AZ3166
|       |-- app
|           |-- startup
|           '-- stm32cubef4
|       |-- cmake
|       |-- lib
|           |-- mxchip_bsp
|           |-- netxduo
|           |-- stm32cubef4
|           |-- threadx
|               '-- wiced_sdk
|       '-- scripts
|-- cmake
`-- shared
    |-- lib
    |   |-- jsmn
    |   |-- netxduo
    |   '-- threadx
    '-- src
```

Main source repository

Third-party libraries and app-specific files

NetX Duo and ThreadX as Git submodules

<https://github.com/eclipse-threadx/iot-devkit>



The Starter Application: main files in “app”

CMakeLists.txt
board_init.c
board_init.h
cloud_config.h
cmsis_utils.h
console.c
main.c
screen.c
screen.h
snntp_client.c
snntp_client.h
threadx_pnp_info.h
wwd_networking.c
wwd_networking.h

Add WiFi
credentials here

Start coding
here

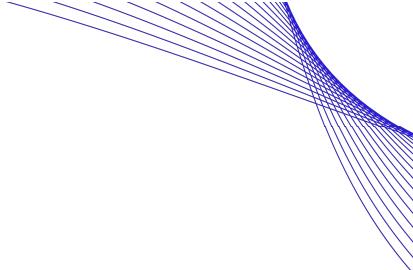
Call to Action

Eclipse ThreadX is a mature RTOS...
... but a growing open source project

- **Get the code!**
<https://github.com/eclipse-threadx>
- **Read the docs!**
<https://github.com/eclipse-threadx/rtos-docs-asciidoc>
- **Stay informed!**
<https://accounts.eclipse.org/mailing-list/threadx>
- **Join the ThreadX Alliance!**
- **Contributors and committers wanted!**



What You Can Do



Potential Contributions (1 / 2)

- ThreadX
 - Ports to new architectures, platforms, SOCs
 - Add modules support to more ports (MPU / MMU needed)
- NetX Duo
 - Upgrade protocol support (MQTT v5)
 - Restore support for CoAP / LwM2M
 - Next-generation protocols (Eclipse uProtocol, Eclipse zenoh, etc.)

Potential Contributions (2 / 2)

- USBX
 - Expand class support
 - Add support for additional controllers
- FileX / LevelX
 - Implement alternatives to FAT (littleFS?)
- GUIX
 - Replace current Windows-only tools with cross-platform alternative
 - Replace GUIX! (LVGL?)



Frédéric Desbiens

frederic.desbiens@eclipse-foundation.org

<https://linkedin/in/fredericdesbiens>