How to build stunning embedded UIs with Zephyr

Zephyr with rich GUI solutions

28.10.2025 – Cédric Le Dillau

Qt for Zephyr

Agenda

1. Creating UI?
2. Qt Quick Ultralite – the engine inside Qt for MCUs
3. Workflow & Dataflow
4. Building a Qt Zephyr project
5. Roadmap
6. Wrapping up

Solutions Engineering Manager at Qt Group, EMEA.

Worked in different industries for over x10 years, including electronic, signal processing, broadcast television, broadband communications, security, and OTA upgrade, for embedded systems.

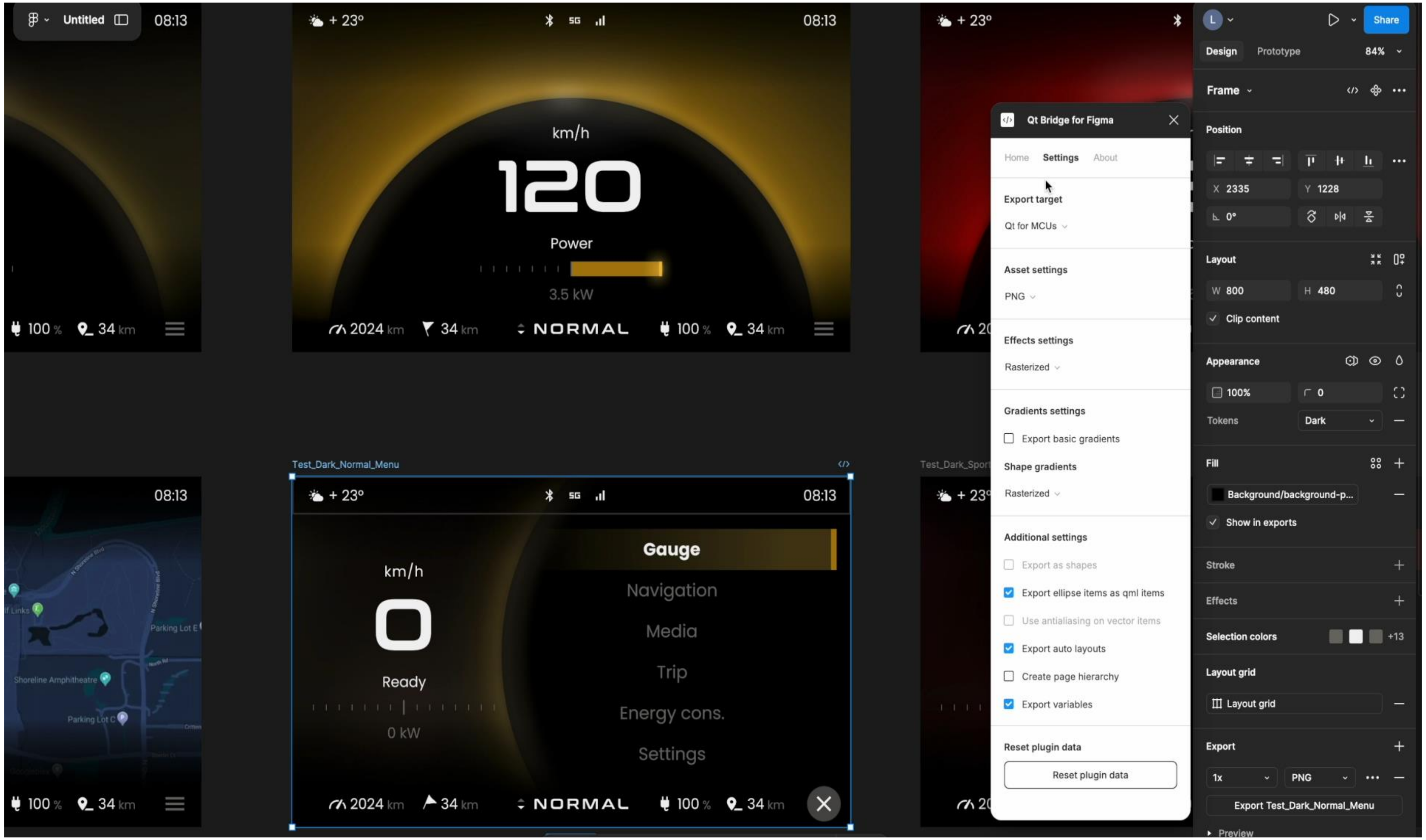
For today’s topic:

- Working with the integration of Qt for MCUs using Zephyr RTOS.
- Bridging the gaps of *tools*

QtWS26

Examples of UI

Qt Figma Bridge for MCUs



Qt Bridge for Figma

Home | Settings | About

Export target
Qt for MCUs

Asset settings
PNG

Effects settings
Rasterized

Gradients settings
□ Export basic gradients

Shape gradients
Rasterized

Additional settings
□ Export as shapes
☑ Export ellipse items as qml items
□ Use antialiasing on vector items
☑ Export auto layouts
□ Create page hierarchy
☑ Export variables

Reset plugin data
Reset plugin data

Design | Prototype | 84%

Frame

Position
X 2335 | Y 1228
0°

Layout
W 800 | H 480
✓ Clip content

Appearance
100% | 0
Tokens: Dark

Fill
Background/background-p...
✓ Show in exports

Stroke

Effects

Selection colors +13

Layout grid
Layout grid

Export
1x | PNG
Export Test_Dark_Normal_Menu
Preview

Qt Bridge for Figma for MCU

What is it?

- Qt Bridge now has a configurable export target as Qt for MCU.
- Once selected, MCU-compatible features will be highlighted in the bridge, and warnings will appear for incompatible items to export.

Why do you need it?

- Rapid hands-off for lightweight software for microcontrollers
- Enhanced scalability from design phase

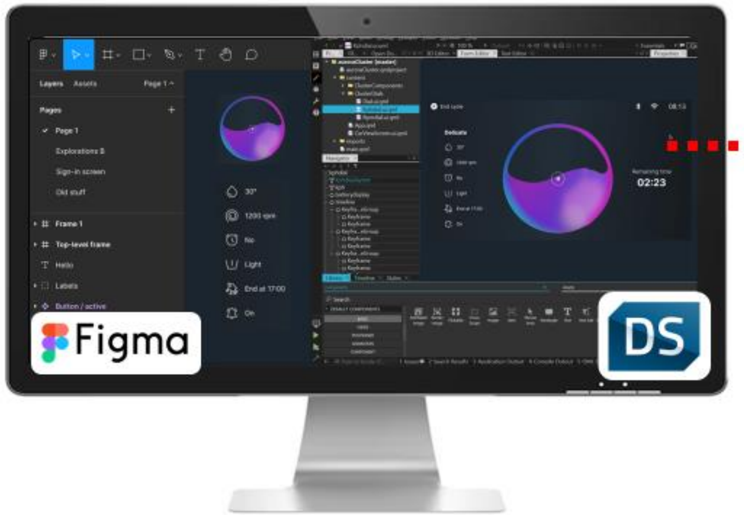
Complete product creation with Qt for MCUs

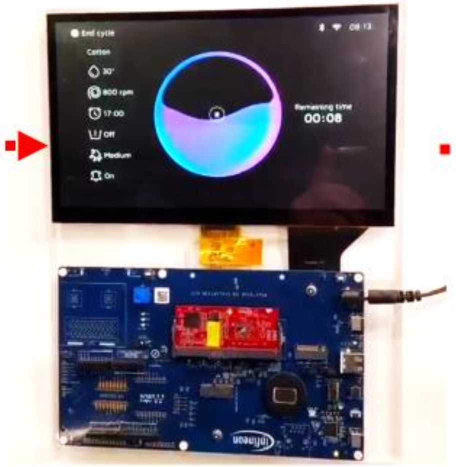
We help to reduce time to market massively

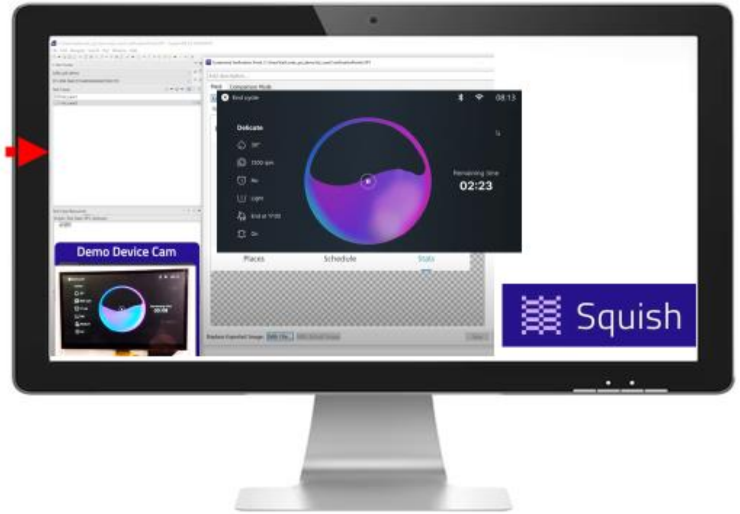
① *"Design to prototype in minutes on real device"*

② *"Whole design runs just in 7MB"*

③ *"Run multi-lingual test automation on MCU"*







Reduction of dev time by 50%

Reduction of testing time by 30%

Zephyr™ Project

Qt Development

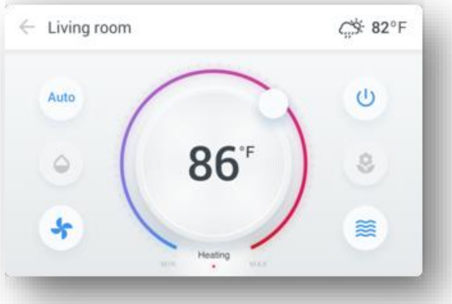
Qt Quick Ultralite – the engine inside Qt for MCUs

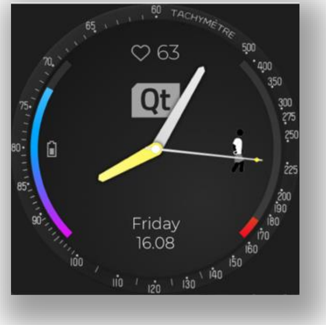
Key features

- Less code with QML declarative language
- Low memory requirements (200 KB RAM min.)
- Wide range of supported hardware platforms

- Hardware-accelerated graphics
- Extensive image format support
- Best-in-class font rendering powered by Monotype Spark

- Library of stylable UI controls and 2D shapes for smartphone-like UX
- Simple yet powerful animation framework
- 2.5D effects





Target hardware

NXP RT1060 EVKB

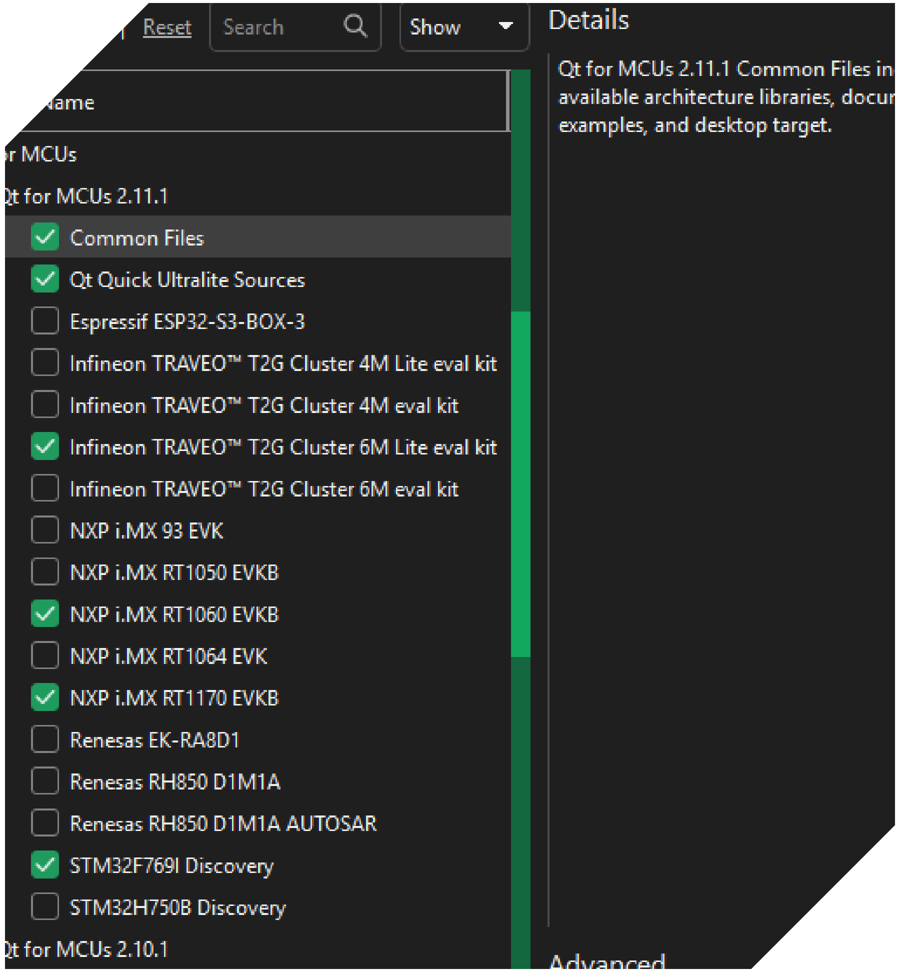
- **ARM Cortex M7**
- **600 MHz**
- **256 Mbit SDRAM**
- **64 Mbit QSPI Flash**
- **512 Mbit Hyper Flash**
- **Ethernet**

Versions:
□ Qt for MCUs 2.12.1
□ Zephyr v4.1.0 / SDK 0.17.0
□ Squish/QUL 9.1.1

Zephyr™ Project

Qt Development

Getting started with Qt



Reset | Search | Show

Details

Qt for MCUs 2.11.1 Common Files in
available architecture libraries, docu
examples, and desktop target.

Name

r MCUs

Qt for MCUs 2.11.1

- ☑ Common Files
- ☑ Qt Quick Ultralite Sources
- ☐ Espressif ESP32-S3-BOX-3
- ☐ Infineon TRAVEO™ T2G Cluster 4M Lite eval kit
- ☐ Infineon TRAVEO™ T2G Cluster 4M eval kit
- ☑ Infineon TRAVEO™ T2G Cluster 6M Lite eval kit
- ☐ Infineon TRAVEO™ T2G Cluster 6M eval kit
- ☐ NXP i.MX 93 EVK
- ☐ NXP i.MX RT1050 EVKB
- ☑ NXP i.MX RT1060 EVKB
- ☐ NXP i.MX RT1064 EVK
- ☑ NXP i.MX RT1170 EVKB
- ☐ Renesas EK-RA8D1
- ☐ Renesas RH850 D1M1A
- ☐ Renesas RH850 D1M1A AUTOSAR
- ☑ STM32F769I Discovery
- ☐ STM32H750B Discovery

Qt for MCUs 2.10.1

Advanced

Download Qt Online Installer (Qt Maintenance Tool)

- **Select Qt Creator**
- **Select one of the MCU kits**
- **Select Qt Quick Ultralite Source**

Open Qt for MCUs Zephyr page:

Using Qt Ultralite with Zephyr

Zephyr™ Project

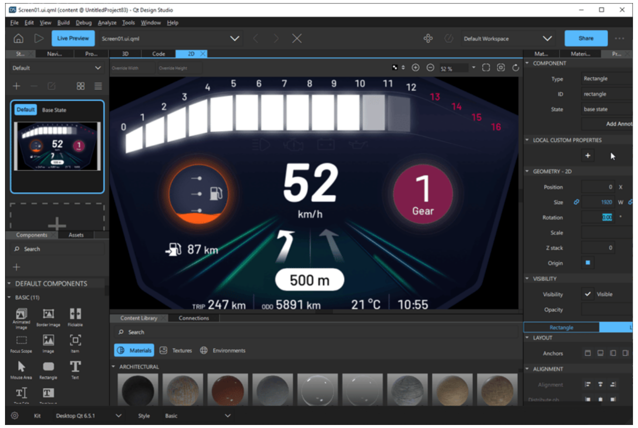
Qt Development

Workflow

Port QUL
To your platform

- **Design**
- Components
- Modules

Develop new QUL
GUI application

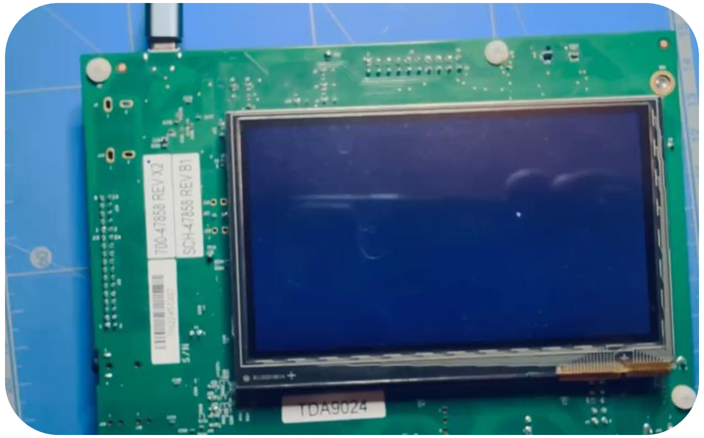


Add backend
(QUL + Zephyr)

Run UI/UX tests

Export as Cmake
project

Using Qt Ultralite with Zephyr



Flash image

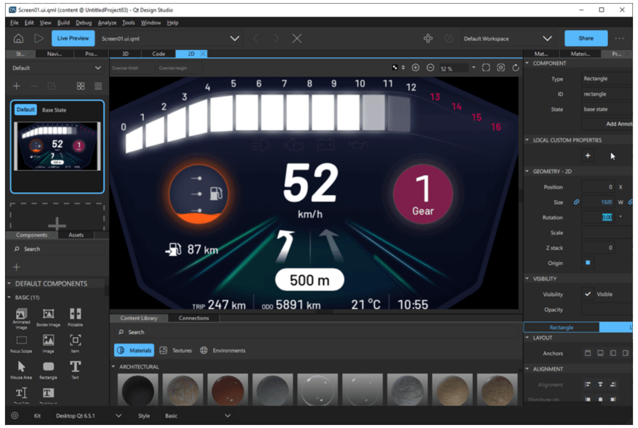
Build the Zephyr™
project

Zephyr™ Project

Qt Development

Workflow

- ○ **Design**
- ○ Components
- ○ Modules



Port QUL
To your platform

Develop new QUL
GUI application

Add backend
(QUL + Zephyr)

Export as Cmake
project

Run UI/UX tests

Flash image

Build the Zephyr™
project

Using Qt Ultralite with Zephyr

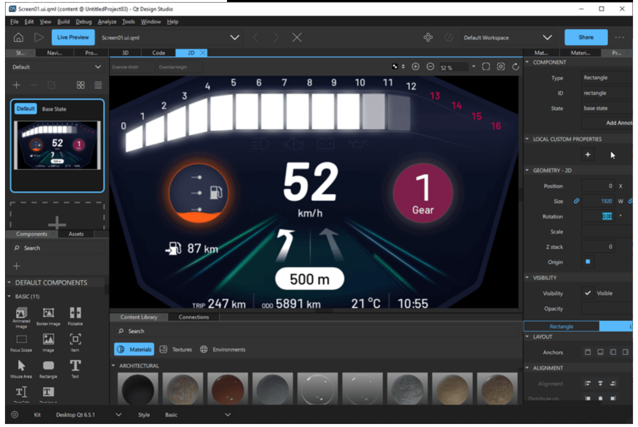
Zephyr™ Project

Qt Development

Dataflow

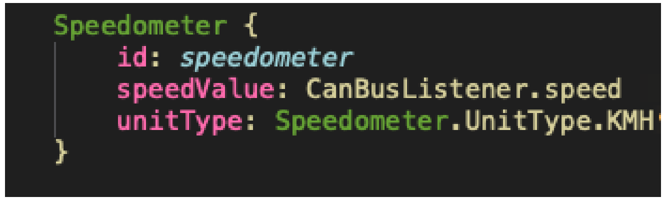
Add backend
(QUL + Zephyr)

- **Design**
- Components
- Modules



{properties}

Qt Ultralite Application UI/UX

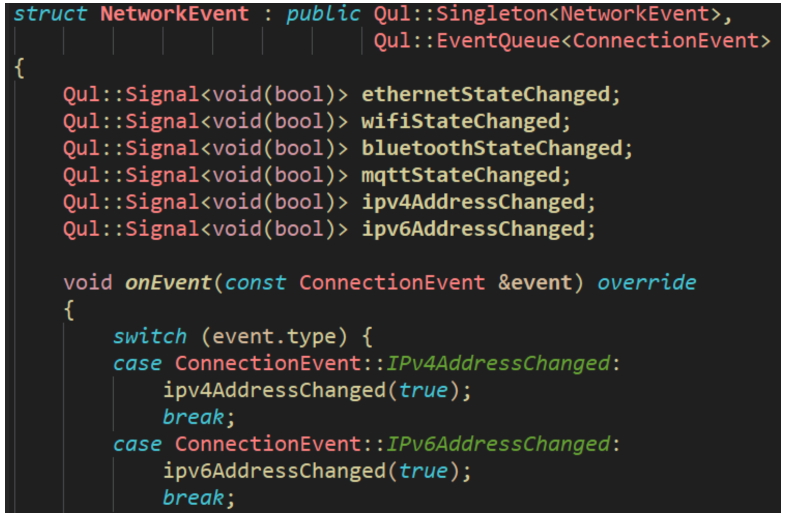


```
Speedometer {
    id: speedometer
    speedValue: CanBusListener.speed
    unitType: Speedometer.UnitType.KMH
}
```

Signals

Signals
Slots

Backend
(EventQueue)



```
struct NetworkEvent : public Qul::Singleton<NetworkEvent>,
                             Qul::EventQueue<ConnectionEvent>
{
    Qul::Signal<void(bool)> ethernetStateChanged;
    Qul::Signal<void(bool)> wifiStateChanged;
    Qul::Signal<void(bool)> bluetoothStateChanged;
    Qul::Signal<void(bool)> mqttStateChanged;
    Qul::Signal<void(bool)> ipv4AddressChanged;
    Qul::Signal<void(bool)> ipv6AddressChanged;

    void onEvent(const ConnectionEvent &event) override
    {
        switch (event.type) {
        case ConnectionEvent::IPv4AddressChanged:
            ipv4AddressChanged(true);
            break;
        case ConnectionEvent::IPv6AddressChanged:
            ipv6AddressChanged(true);
            break;
```

Zephyr RTOS

Zephyr

Zephyr™ Project

Qt Development

Qt Zephyr Roadmap

Now

- Zephyr support released since **Qt for MCUs 2.9**
- Integration with **PXP APIs on NXP platforms** (RT 1060/64) for better graphics performance

2025

- Improved Zephyr + Qt Tooling integration and tool-chain workflow
- More platform support (NXP RW612, Infineon PSoC Edge)
- New IoT/Connected demo
- DeviceLink on Zephyr

Zephyr Community Plans (2025)

- Qt Group is a Zephyr project silver member and has a voting right in the **Technical Steering Community**
- Engage in technical discussion and steer features/fixes that affect Qt for MCUs performance/integration
- Define, propose, and contribute a **graphics standard** on Zephyr and plans to open source HW adaptation layer
- Collaboration with **hardware vendors**

Zephyr™ Project

Qt Development

Wrapping up

- Qt for MCUs for Zephyr is bringing accelerated graphics to Zephyr
- UI is easier to design using Figma and Design Studio
- UI testing with Squish is now possible on MCUs

- Using Qt on MCUs is easy and you won’t be locked-in to a MCU vendor or a specific board
- You can try on one of the officially supported platforms and then customize it for MCU of your own choice.
- The steps for porting are well documented, and you can follow them along.

- All questions and suggestions are welcome!

Thank you!

Now, let’s have a chat!

Qt
for MCUs

Zephyr®

cedric.le.dillau@qt.io

osman.chohan@qt.io

Credits: diver photo from CC-BY-SA Soljaguar