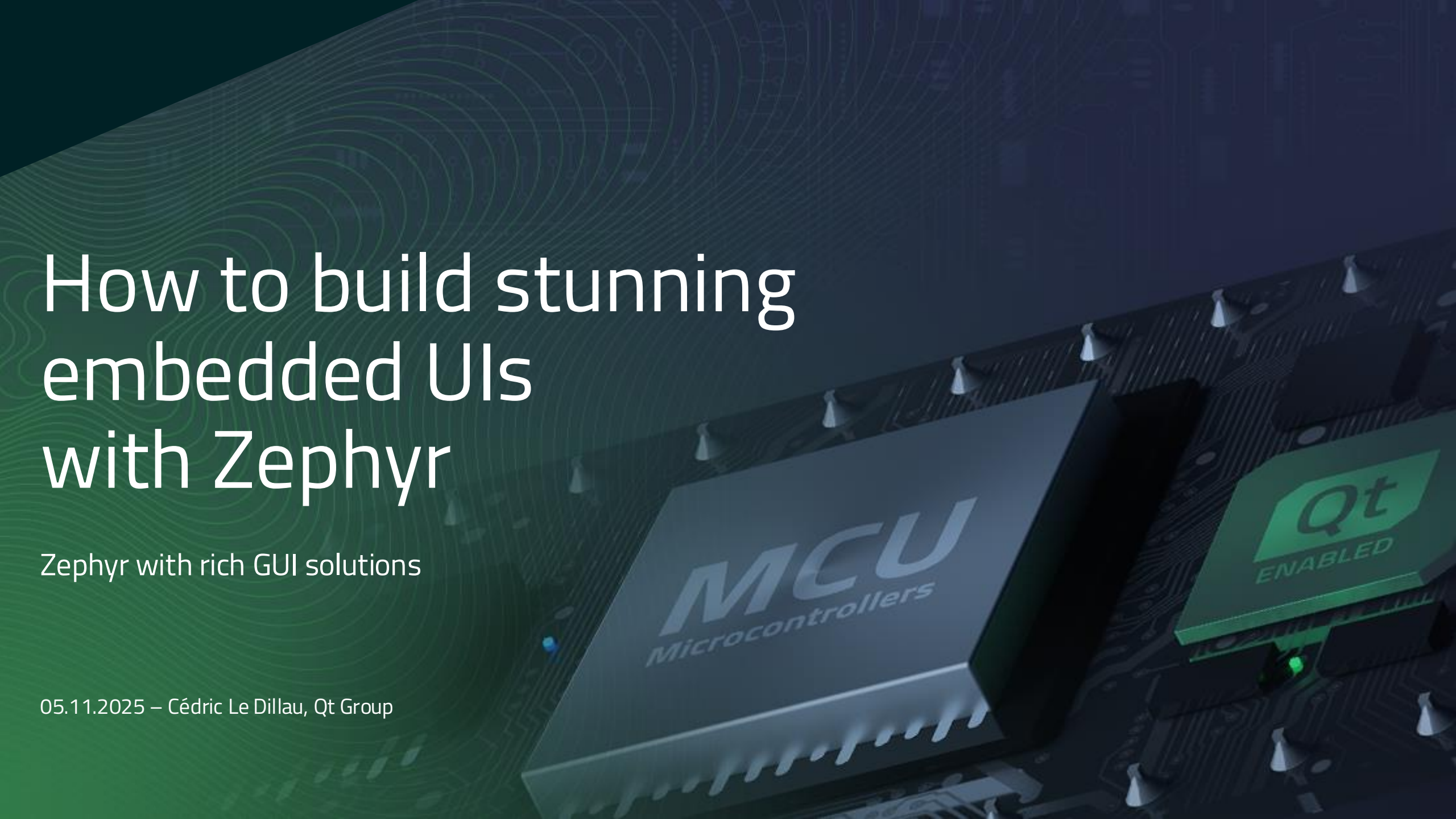
How to build stunning embedded UIs with Zephyr

Zephyr with rich GUI solutions



05.11.2025 – Cédric Le Dillau, Qt Group

Qt for Zephyr

Agenda

1. Stunning UI?
2. Qt Quick Ultralite – the engine inside Qt for MCUs
3. Workflow & Dataflow
4. Roadmap
5. Key takeways

Zephyr®

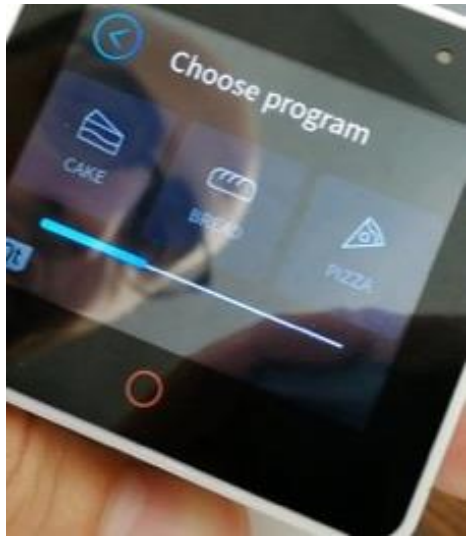
Solutions Engineering Manager at Qt Group, EMEA.

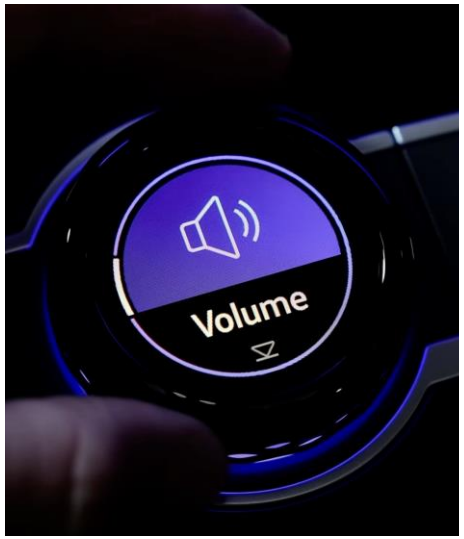
Worked in different industries for over x10 years, including electronic, signal processing, broadcast television, broadband communications, security, and OTA upgrade, for embedded systems.

For today's topic:

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

Examples of UI

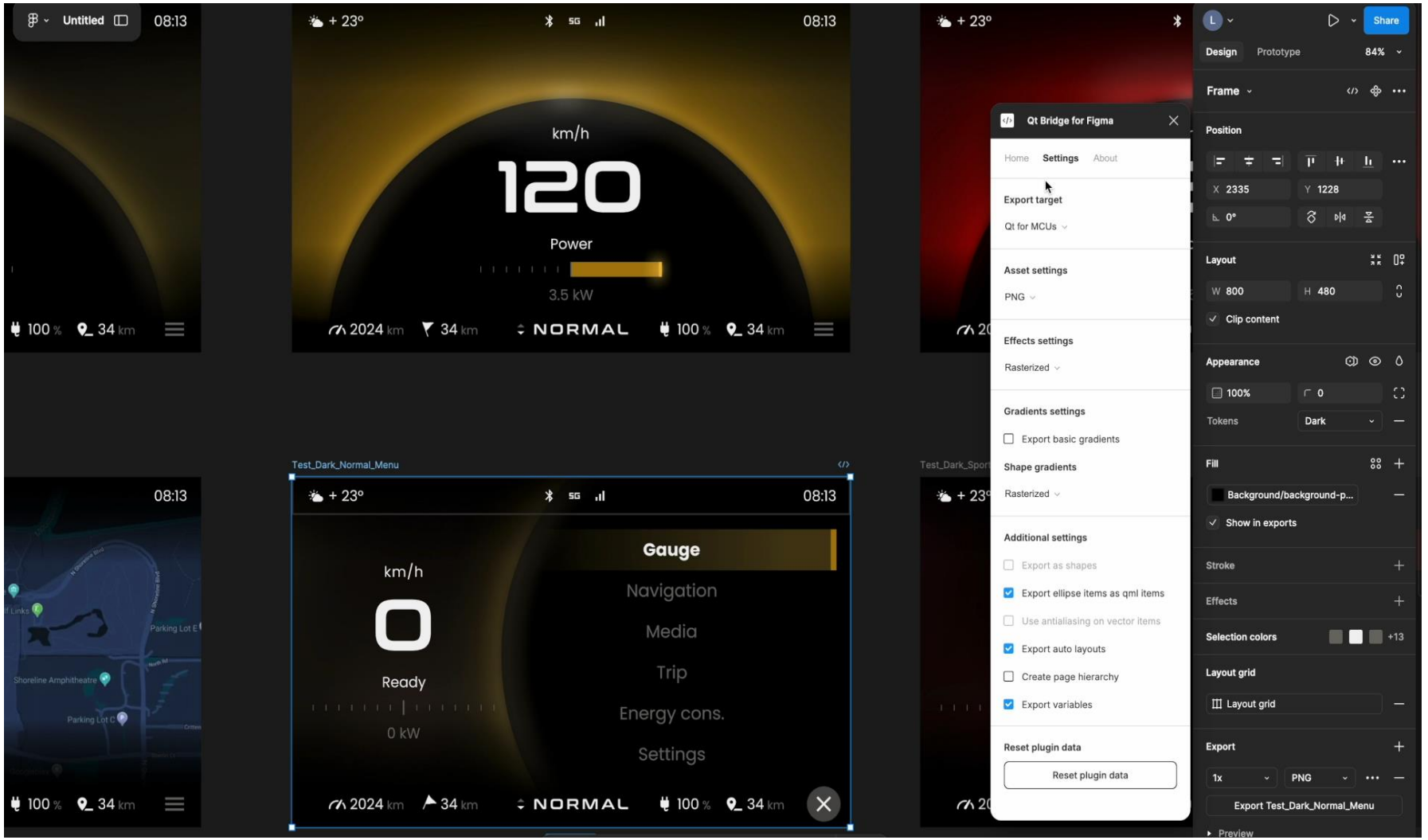




Zephyr™ Project

Qt Development

Qt Figma Bridge for MCUs



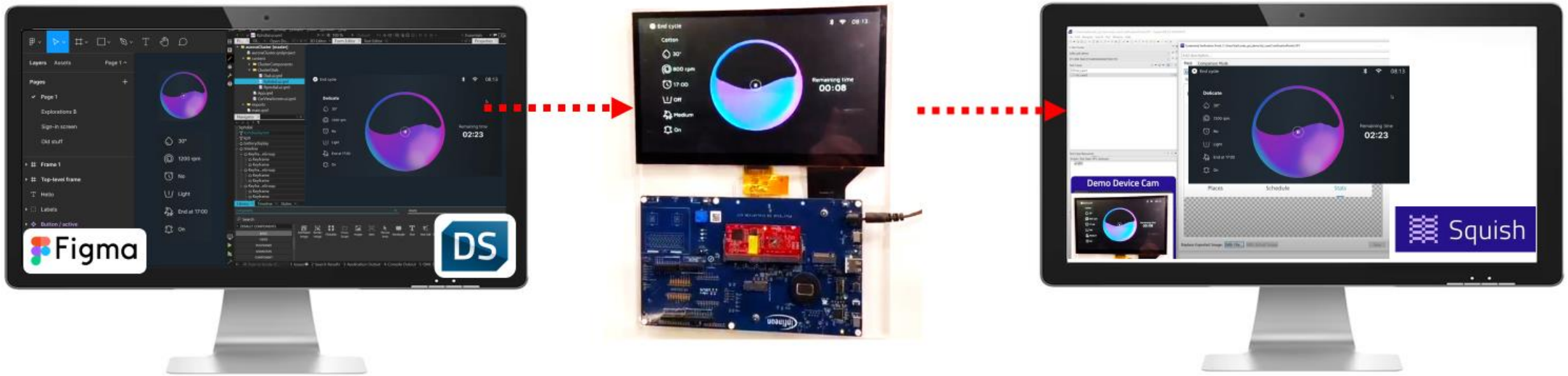
Complete product creation with Qt for MCUs

We help to reduce time to market massively

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

② *"Whole design runs just in 7MB"*

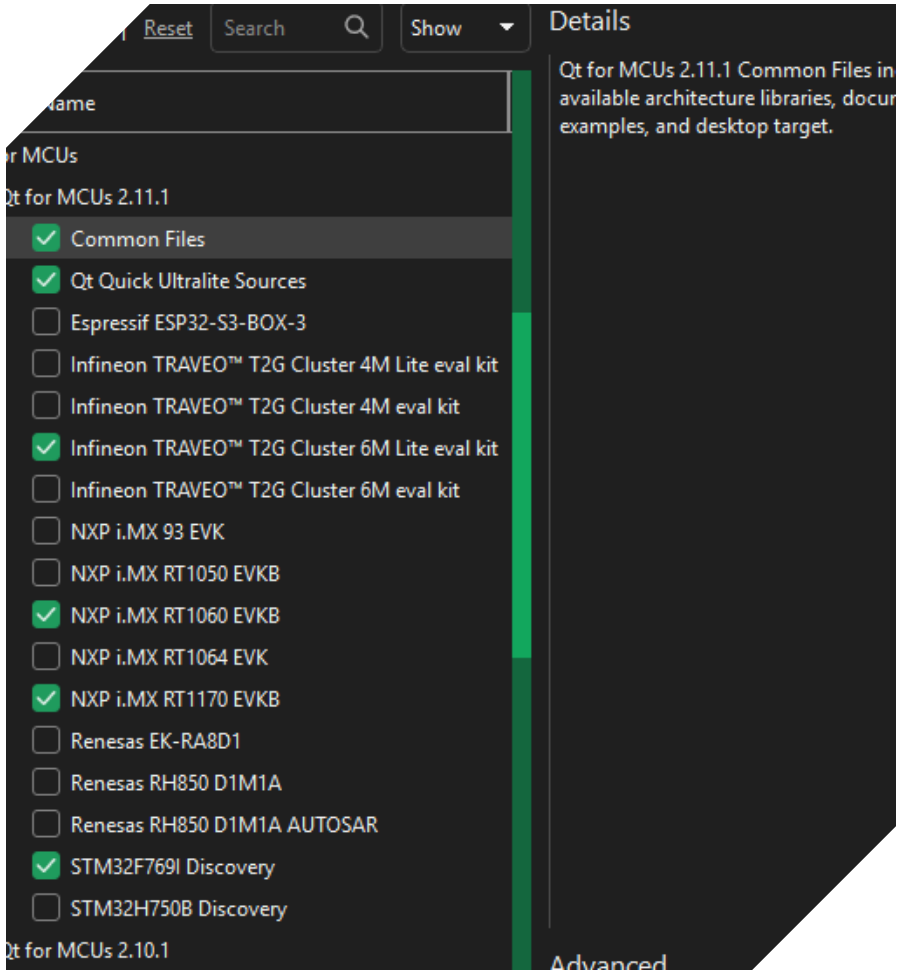
③ *"Run multi-lingual test automation on MCU"*



Reduction of dev time by 50%

Reduction of testing time by 30%

Getting started with Qt



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:

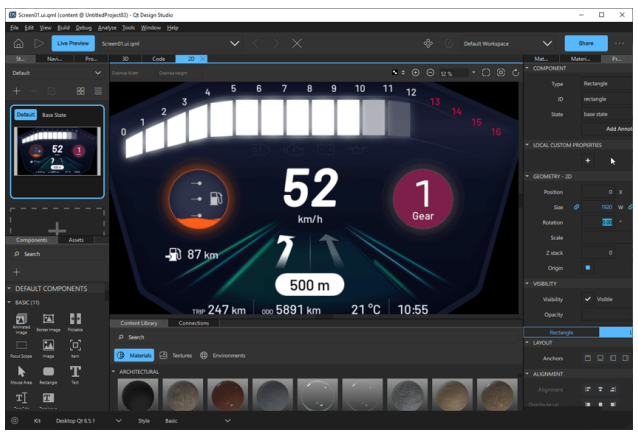
Using Qt Ultralite with Zephyr

Zephyr™ Project

Qt Development

Workflow

- **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
Squish GUI Tester

Flash image

Build the Zephyr™
project

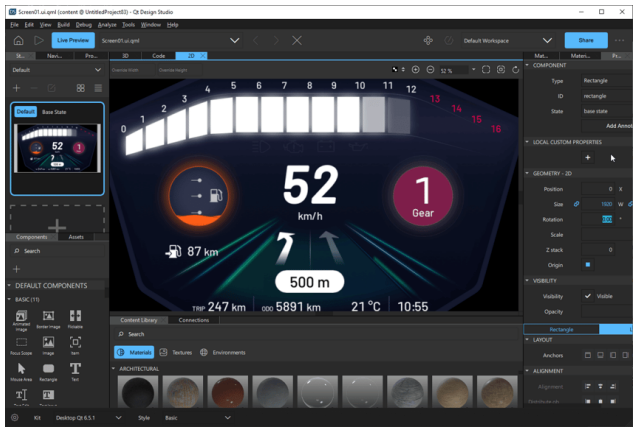
[Using Qt Ultralite with Zephyr]

Zephyr™ Project

Qt Development

Workflow

- **Design**
- Components
- Modules

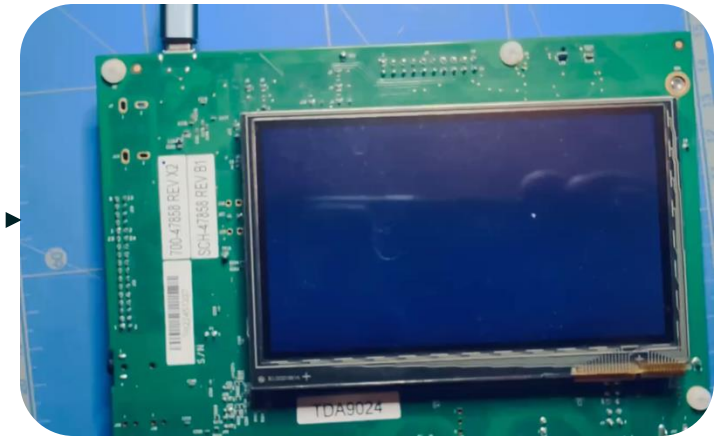


Develop new QUL GUI application

Add backend (QUL + Zephyr)

Export as Cmake project

Port QUL To your platform



Run UI/UX tests Squish GUI Tester

Flash image

Build the Zephyr™ **project**

[Using Qt Ultralite with Zephyr]

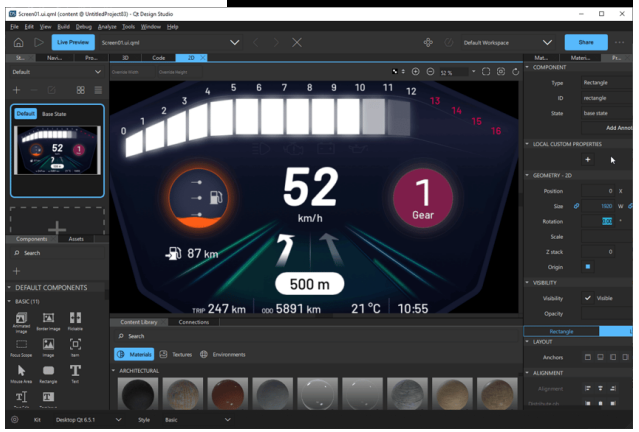
Zephyr™ Project

Qt Development

Dataflow

Add backend
(QUL + Zephyr)

- **Design**
- Components
- Modules



{properties}

Qt Ultralite Application
UI/UX

Signals

Signals
Slots

Backend
(msgq)

Backend
(EventQueue)

Zephyr RTOS

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

```cpp
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;
```
C++

Interactive demo?

Summary of the interactive demo:
- 2 QR Codes:
  - One pointing to unsecure demo (**without** url)
  - One pointing to secure demo (with url)
- Both applications are same Qt/WebAssembly app with different parameters
- Qt for MCUs application on demo board uses Zephyr OS backend to send MQTT messages and control display of last slides of this presentation.

ZephyrOutro/index.html

Qt Zephyr Roadmap

Now

- Zephyr support released since **Qt for MCUs 2.9** (2.12.1 in Nov)
- 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 (Squish) on Zephyr

Zephyr Community Plans

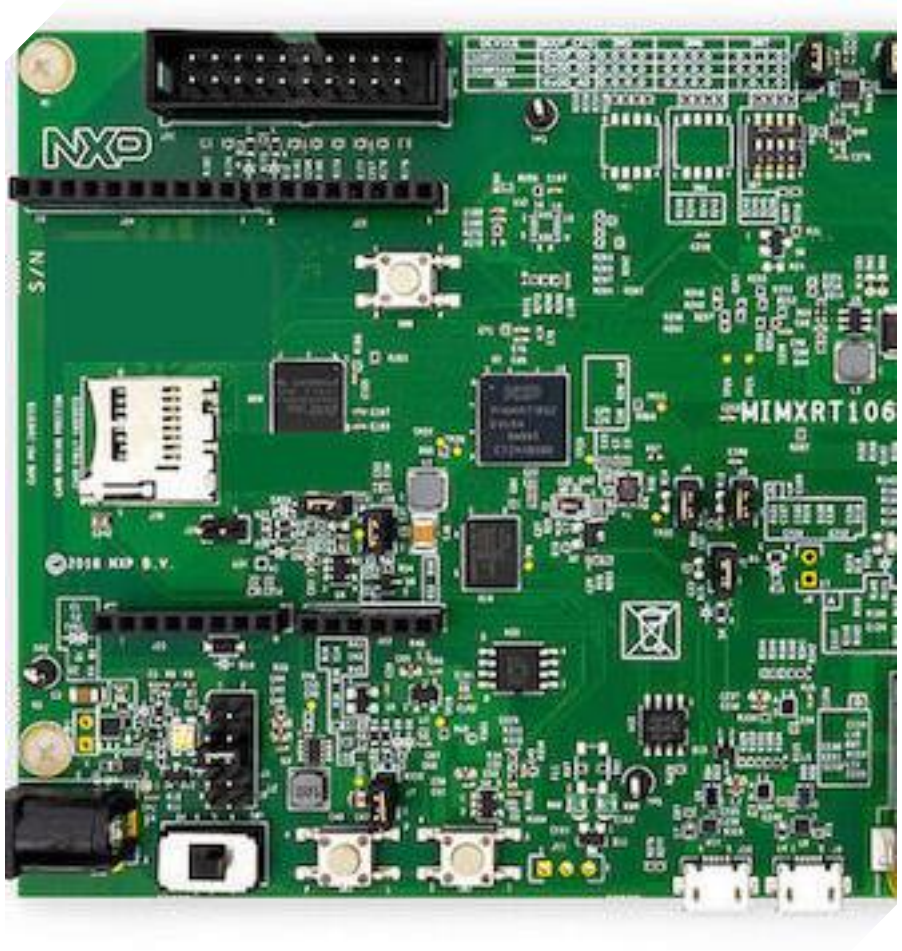
- Qt Group is a Zephyr project silver member and has a voting right in the **Technical Steering Community**
- Engage in technical discussion and steer features/fixes that affect Qt for MCUs performance/integration
- Define, propose, and contribute a **graphics standard** on Zephyr and plans to open source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

qt-demos.herokuapp.com/static/ZephyrOutro/index.html

Zephyr™ Project

Qt Development

Target hardware/software



NXP RT1060 EVKB

- **ARM Cortex M7**
- **600 MHz**
- **256 Mbit SDRAM**
- **64 Mbit QSPI Flash**
- **512 Mbit Hyper Flash**
- **Ethernet**
- **WiFi/BT: IW416**

Versions:

- ❑ Qt for MCUs 2.12.1
- ❑ Zephyr v4.1.0 / SDK 0.17.0
- ❑ Squish/QUL 9.1.1

Key takeaways

- Qt for MCUs for Zephyr™ is bringing accelerated graphics to Zephyr™
- UI is easier to design using Figma and Qt 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
- Try on an officially supported platforms and then customize it your own developments and choices.

- Porting steps are well documented, and you can follow them along.
  https://doc.qt.io/QtForMCUs/
  https://doc.qt.io/QtForMCUs/qtul-using-with-zephyr.html

- Safety, Security and Maintenance are our priorities.

Thank you!

Now, let's have a chat!

Cédric Le Dillau, Qt Group
cedric.le.dillau@qt.io

Credits: diver photo from CC-BY-SA Soljaguar