Developing Embedded Devices with Qt

The Qt Company
Andy Nichols

Embedded Conference Finland 2017
Writing Software for Application Processors can be Overwhelming
Desktop and Mobile platforms are less overwhelming

- Many of the hard decisions are already made:
  - Operating System
  - Native API’s
  - Available Toolchains
  - IDE and Tooling
Microcontrollers put you in Control
(but not more than you can handle)

› Bare metal programming
› You choose the OS
› You build the OS
› You could write the OS!
› You can write or at least see every line of code that runs on the device.
Application Processors

 › The control of a Microcontroller
 › The features of a Desktop
 › More features, more problems
Pick an Operating System / Kernel

› Linux  
› Android  
› QNX  
› Integrity  
› VxWorks  
› *BSD
Decide what will be in your Stack

› Want graphics?
  › You need a graphics tool kit like Cairo

› Want to render scalable text?
  › You need Freetype

› What about non-trivial text layouts?
  › You need harfbuzz

› Do you want to play sounds?
  › You need ALSA

› What if I’m not on linux?
  › You need other things than above!
Decide on a Toolchain

› Needs to be able to build not just your kernel and your Application, but also everything else on your system.
› Can I cross compile from Windows?
› Good support for debugging and profiling
Platform Complexity

 › Unlikely to manage every line of code
   › Lots of Shared Libraries that will make up stack
   › Operating Systems are feature rich

 › Impractical to write every line of code
   › Won’t get the most value out of hardware
   › Not all hardware will have open specs
     › GPU
     › Wireless devices
Qt makes Application Processor projects more manageable.
The Qt Framework is Middleware

› No matter what OS you pick, the Qt APIs will be the same
› Qt becomes the “native” platform
   › How do I show text
   › How do I request data from a database
   › How do I read from a Bluetooth sensor
   › How do I play an alarm
   › The answer is there is a Qt API for that!
Qt Tooling is your SDK
Qt Creator
Qt provides APIs to make application development cycles shorter.
Qt UI Offering – Choose the Best of All Worlds

Qt Quick

C++ on the back, declarative UI design (QML) in the front for beautiful, modern touch-based User Experiences.

Qt Widgets

Customizable C++ UI controls for traditional desktop look-and-feel. Also good for more static embedded UIs for more limited devices / operating systems.

Web / Hybrid

Use HTML5 for dynamic web documents, Qt Quick for native interaction.
Rapid Workflow with Qt Quick

Declarative UI Design
Stunningly Fluent Modern User Interfaces, written with QML. Ideal for rapid UI prototyping.

Imperative Logic
Power of Cross-Platform Native Qt/C++

Core
Processes, Threads, IPC, Containers, I/O, Strings, Etc.

Network
HTTP
FTP
SSL

Sql
SQL & Oracle Databases

I/O, Strings, Etc.

+ Direct Hardware Access

Designer

Developer
Qt for Device Creation Developer Offering

**Embedded solutions**
- Virtual keyboard
- Qt Quick 2D Renderer
- Utilis, Wifi, etc.

**Software Stack**
- OTA solution
- Boot to Qt stack

**Embedded tooling**
- Device emulator
- Build your own stack
- Remote debugging

**Qt Toolkit**
- Qt Add-ons
- New Qt Add-ons (Charts, Data Visualization)
- Qt Essentials

**Embedded Platforms**
- eLinux
- QNX
- WEC
- VxWorks
Boot to Qt Software Stack

› Immediate Embedded Prototyping
› Kick-start to Embedded Projects
› Full Customization through the Yocto Project tooling
Qt Virtual Keyboard

› Full onscreen keyboard solution
› Customized styling
› Handwriting Support
› Asian input methods
# Qt Framework Overview

## Add-Ons
- Canvas 3D
- Active Qt
- X11, Windows, Mac Extras
- Android Extras
- XML & XML Patterns
- Charts
- Graphical Effects
- Print Support
- Image Formats
- SVG
- Qt Quick 2D renderer
- NFC
- Sensors
- Positioning
- Bluetooth
- Data Visualization
- Concurrent
- Serial Port
- D-Bus
- Purchasing
- Qt 3D
- WebEngine
- WebSockets
- WebChannel

## Essentials
- GUI
- Widgets
- Network
- SQL
- Quick Controls
- Quick Dialogs
- Quick Layouts
- Quick
- Test
- QML

## Desktop & mobile platforms
- Windows
- Mac
- Linux Desktop
- Android
- iOS
- WinRT

## Development Tools
- Qt Creator
  - Cross-platform IDE
- Qt Designer
  - GUI Designer
- Qt Linquist
  - I18N Toolset
- Qt Assistant
  - Documentation Tool
- moc, uic, rcc
  - Build Tools
- Qt Quick Compiler
- Qt Visual Studio Add-In
- qmake
  - Cross-platform Build Tool
- CPU usage analyzer
- QML Profiler
- Clang static analyzer
- Qt Assistant
  - Documentation Tool
- Qt Quick Compiler
- Qt Visual Studio Add-In
How can you integrate Qt into your Project?

› Source Code
› Linux
  › Yocto Project
  › Buildroot
  › Timesys

› Android
› QNX
› VxWorks
› Integrity
Thank you!

Check out our website at qt.io for more info