Introduction to Qt/QML for Embedded Development

Based on Qt 5.6

Fundamentals of Qt for Embedded Linux

- Fundamentals of Qt for Embedded Linux
  - The Story of Qt
  - "Hello World" with QtQuick
  - Qt for Embedded Linux Overview
  - Supported Platforms
    - EGL
    - Introduction to Wayland
  - Building Qt for Embedded Linux
  - Introducing Qt Creator
  - Developing for Embedded Linux with Qt Creator
  - Practical Tips for Developers

Introduction to QtQuick

- Introduction to Qt Quick
- Composing User Interfaces
  - Graphical Elements
  - Text Elements
  - Anchor Layout
- User Interaction
  - Mouse Input
  - Touch Input
  - Keyboard Input
- Components
- Presenting Data
  - Arranging Items
  - Simple Data Models
  - Views
    - Customizing the Views
    - The Path View

QtQuick and C++

- Objects in Qt
  - Common Features of Qt's Object Model
  - Object Communication Using Signals & Slots
- Variants and Properties
  - Variants
  - Properties
- Integrating QML with C++
  - Exporting C++ Objects to QML
  - Creating New QML Elements
    - Creating Non-GUI Elements
    - Creating GUI elements
  - Using Custom Types
  - Attached Properties
  - Tips and Tricks

Supplemental Topics

QML Topics

- QtQuick Controls
- WebEngine
- Animations
- States and Transitions
- Declarative State Machines
- Painting and Effects
- The Loader Element
- Drag and Drop

Mixed C++/QML Topics

- Internationalization
- Unit Testing
- Plug-ins
- Platform, device, locale or resolution specific files
- Wayland Compositor

Supplemental Topics

C++ Topics

- The Event System
- Customized Drawing
- Resources

Model/View

- Model/View Concept
- Custom Models
- Model from C++
- Tree Models
- Value-based Models
- Proxy Models
- ObjectModel