Choosing the right Embedded Linux platform for your next project

Laszlo Agocs
Senior Software Engineer
The Qt Company, Oslo, Norway

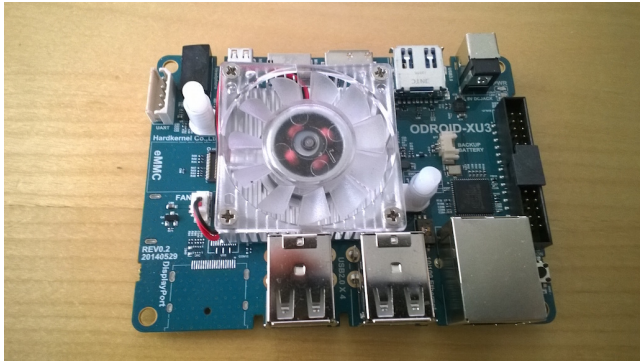
Qt WORLD SUMMIT 2015

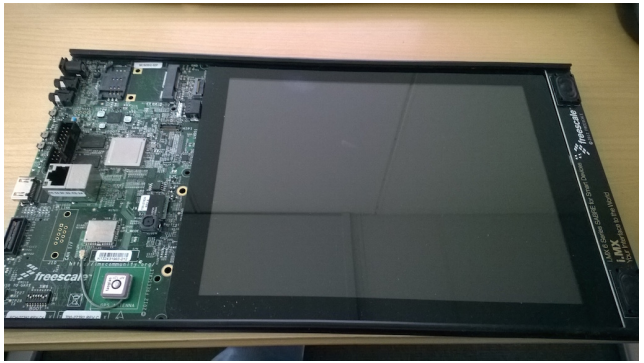
Agenda

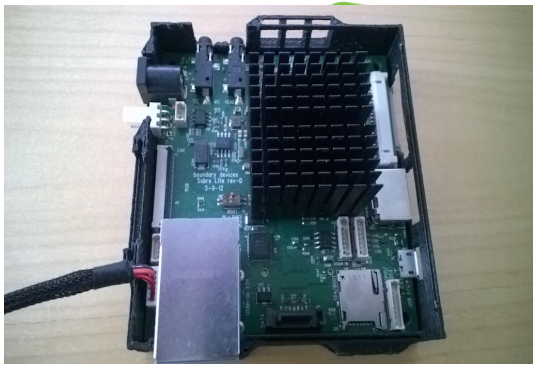
1. What is “embedded” in this context?
2. What do we typically want to build?
3. Why is it hard without Qt?
4. What does Qt expect?
    - Graphics
    - Input
    - Multimedia
    - Software stack

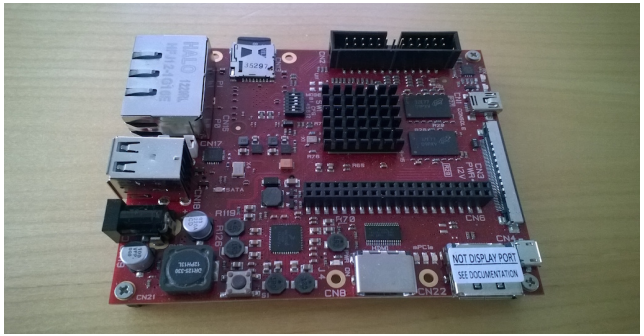
What do we mean by Embedded?

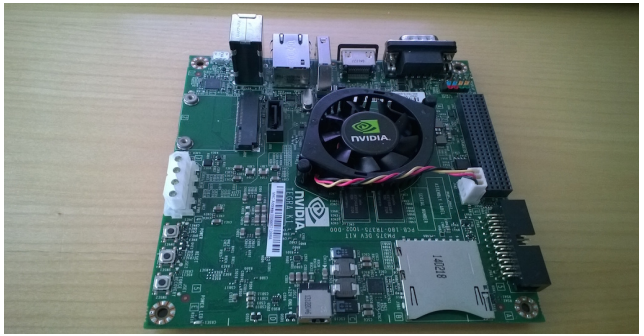
- At least 500 MHz CPU, 256 MB RAM
  - These specs do not mean much in practice. It all depends on what you want to build.
- (GPU with OpenGL ES 2.0 and 128 MB memory)
- Microcontrollers and similar are out of question.


ODROID-XU3
eMMC
DisplayPort

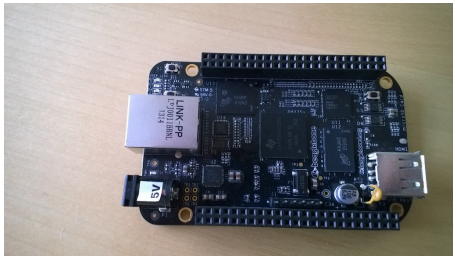

freescale

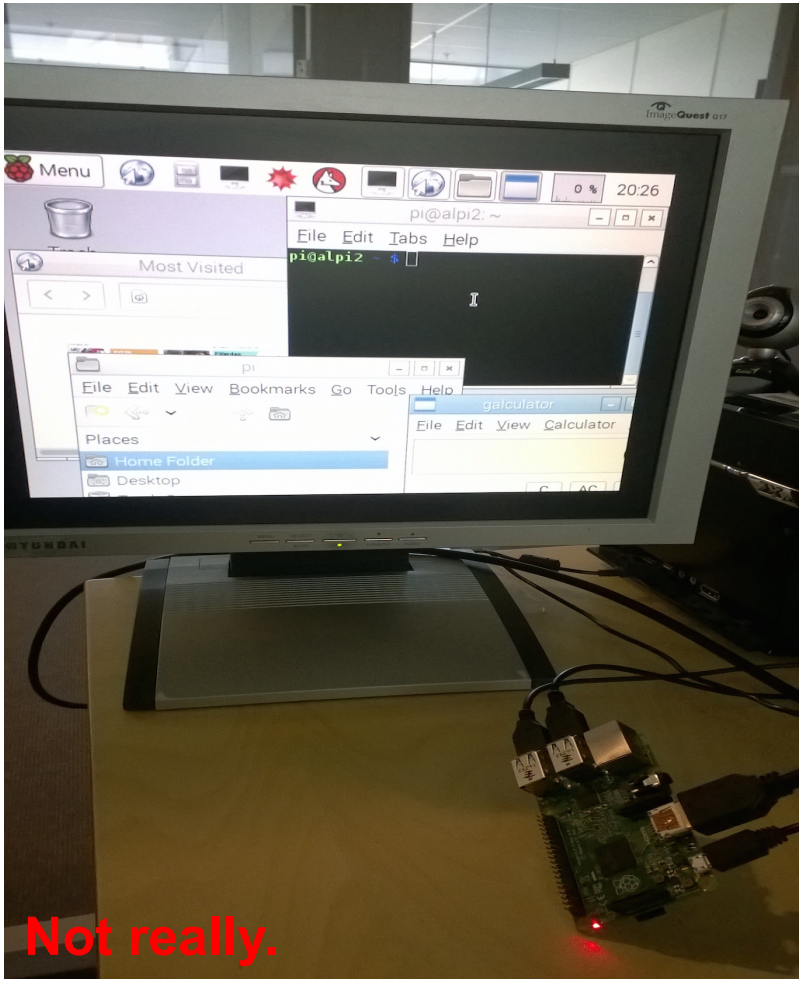


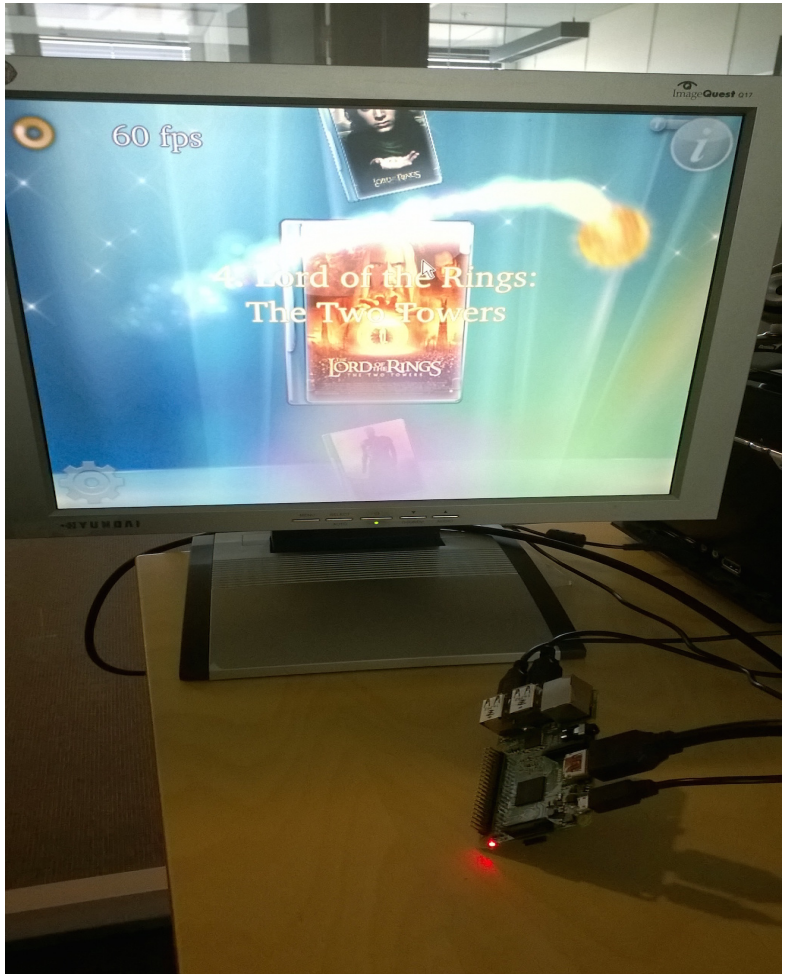

NOT DISPLAY PORT
SEE DOCUMENTATION

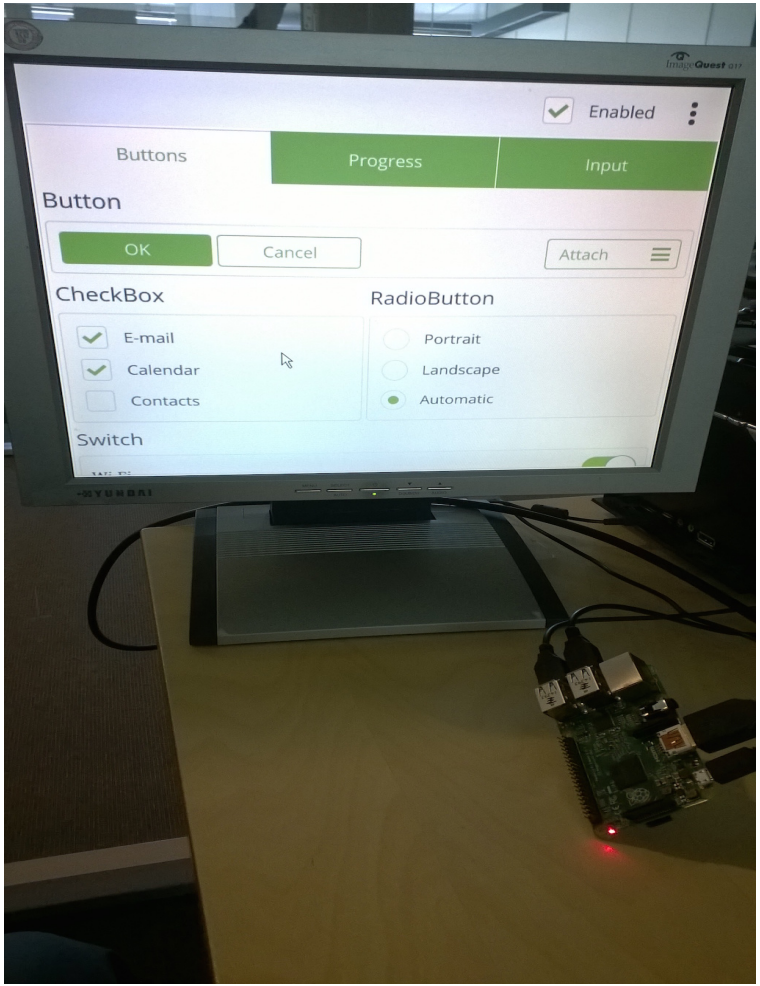

NVIDIA

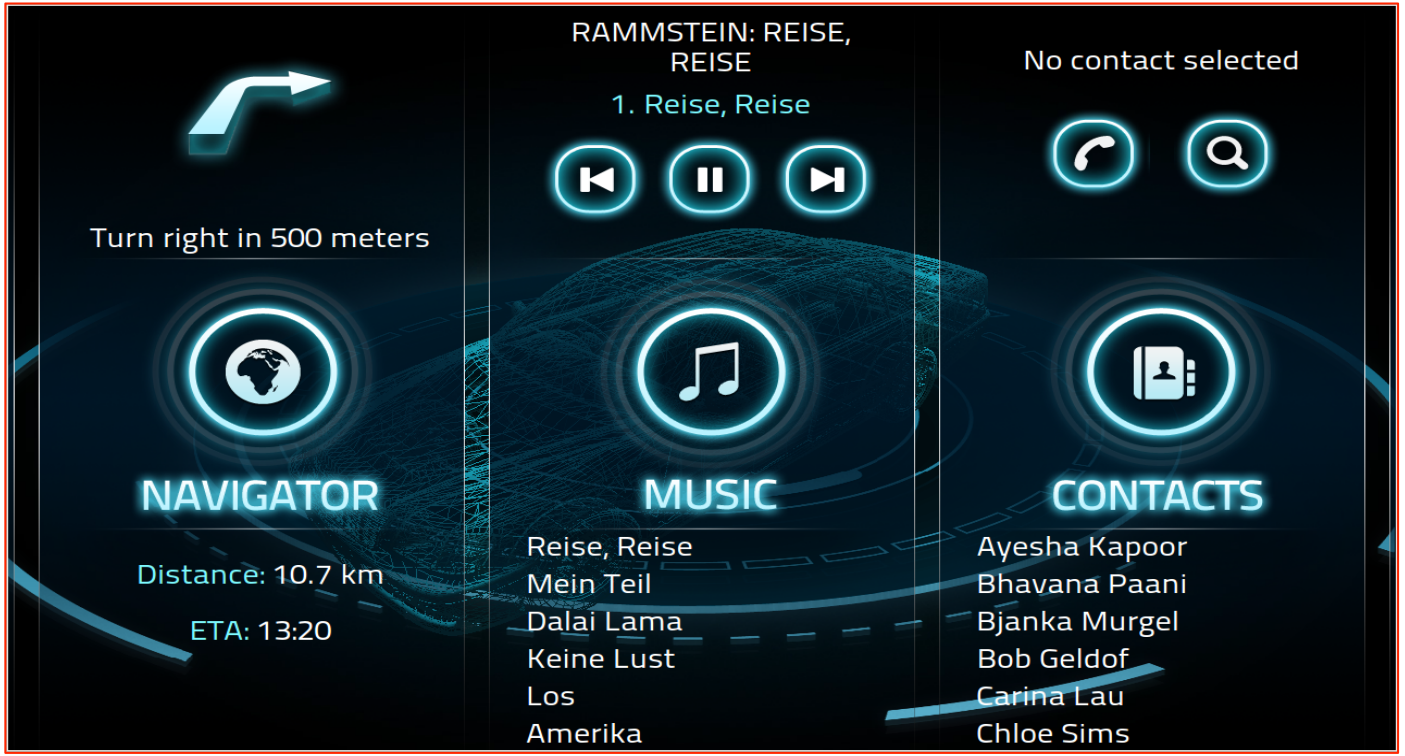

HDMI

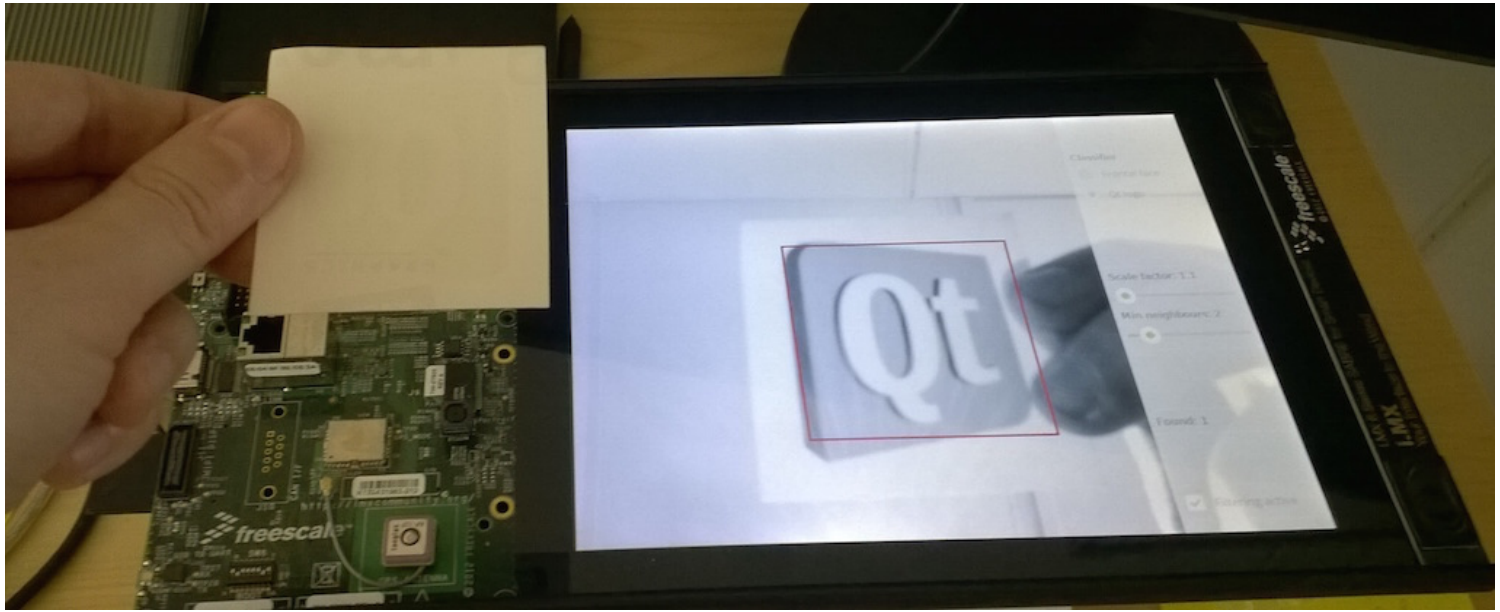



Menu
0 %
20:26
pi@alpi2: ~
File Edit Tabs Help
pi@alpi2 ~ $
Most Visited
File Edit View Bookmarks Go Tools Help
Places
Home Folder
Desktop
calculator
File Edit View Calculator
Not really.


60 fps
4. Lord of the Rings:
The Two Towers


Enabled
Buttons
Progress
Input
Button
OK
Cancel
Attach
CheckBox
E-mail
Calendar
Contacts
RadioButton
Portrait
Landscape
Automatic
Switch


RAMMSTEIN: REISE, REISE
1. Reise, Reise
No contact selected
Turn right in 500 meters
NAVIGATOR
Distance: 10.7 km
ETA: 13:20
MUSIC
Reise, Reise
Mein Teil
Dalai Lama
Keine Lust
Los
Amerika
CONTACTS
Ayesha Kapoor
Bhavana Paani
Bjanka Murgel
Bob Geldof
Carina Lau
Chloe Sims


freescale
Qt
freescale
i.MX

Why is this hard?

- Cross-compilation
- A full windowing system may not be available or not desirable
- User interface controls completely customized
  - No native look and feel. No ready-made widgets.
- EGL, OpenGL, and friends
  - Driver (vendor) specific bits are often inevitable.

Why is Qt great on Embedded?

- QPA
  - Platform plugins for running with and without windowing systems.
  - Comes with platform/vendor-specific glue for many boards.
- Qt Quick
- Qt Quick Controls
- Qt Multimedia
- Cross-compilation

Graphics

Platform plugins for Embedded Linux

- linuxfb: SW-only rendering, no OpenGL.
- eglfs: Windowing system-less, full-screen EGL/OpenGL ES platform.
  - Multiple backends.
- xcb: The full, windowed X11 platform.
  - Multiple backends.
- wayland
  - Multiple backends.

1. Does the device have a GPU?

If not, then Qt Quick 2D Renderer or the legacy QWidgets are the only option.
Limited performance, animation and visual effects.

2. Do you need a full windowing system like X11 or Wayland?

If yes, are you sure? In many cases this is overkill.

Check if accelerated OpenGL is really available.

3. Does the platform plugin have the necessary backend?

eglfs backends

- *“Could not create the egl surface: error = 0x300b”*

- Backends in Qt 5.6:
  - Vivante (i.MX6)
  - Mali (ODROID)
  - Broadcom (Raspberry Pi)
  - DRM/KMS/GBM (Intel on Mesa)
  - X11
  - Some do not need a backend (Beaglebone)
  - Others will usually need a new backend.

eglfs challenges

- "*OpenGL windows cannot be mixed with others*"
  - eglfs is not a compositor, except for software-rendered QWidget windows.
- Multiple displays
  - Plain EGL has no way to control this. Platform/vendor-specific approaches.

eglfs challenges

- *“Supports OpenGL ES”* is **not** sufficient to make an informed decision on whether you will get smooth 60 FPS accelerated UIs with a particular hardware and software stack.

Wayland with Qt Wayland & Compositor

- Confirm there is official, stable, maintained Wayland support for the board.
- Multiple backends: wayland-egl, brcm-egl, xcomposite-glx/egl, libhybris
- Systems with custom buffer sharing mechanisms may need a new one.
  - *“Supports Wayland”* is **not** sufficient to make an informed decision.
- Test on the boards. With Qt. Do not assume it will work just because it works on your desktop PC or because it works with (a perhaps patched) Weston.

Input

Input

- Keyboard, mouse, (multi)touch, tablet/pen

  Easy with a windowing system (X11, Wayland)
  - But not always, esp. touch and pen.

  Up to Qt's own backends without a windowing system
  - **libinput** (new in 5.5, default in 5.6 when present)
  - **evdevkeyboard**, **evdevmouse**, **evdevtouch**

  Dependencies
  - Hotplugging + dynamic showing and hiding of the mouse cursor need **libudev**.

Input

- tslib: a touchscreen access library
  - Qt's tslib backend only provides mouse events (no QTouchEvent)
  - For resistive, single-touch screens where the driver does no MT protocol.
  - Calibration
- Voice: Qt Speech
  - Tech preview in Qt 5.6

Multimedia

Multimedia

- Video playback needs HW acceleration for GStreamer.
- GStreamer 0.10 is being phased out.
  - 1.0 is now the default in Qt 5.6.
- Getting Qt Multimedia running **efficiently** can be tricky with new hardware and drivers.
  - If camera/video is essential, prefer known-good boards.

Software stack

Software stack

- Custom system images vs. Yocto, Buildroot, and similar.
- Prefer the latter for maintainable and extensible generation of system images and SDKs.

- *Is there an **actively maintained**, **official** Yocto layer for my board?*

Software stack

- How does Qt get in there?
- configure/make/install is still (and going to be) the official way.
- Random Qt builds with hacked makespecs, unknown configuration, features silently disabled are not necessarily ideal.
- Yocto integration improvements on-going.

Summary

Hardware, drivers, software stack, and application design go hand in hand and affect each other heavily.

Test early, test often.
With Qt.
On the target device.

Thank you!

http://www.qt.io