Linux Plumbers Conference 2011

OSWALD: Lessons from and for the Open Hardware Movement

Tim Harder

OSU Open Source Lab

September 8, 2011



Inception of the OSWALD and the project's timeline

- Concepts/ideas leading to the project
 - OSU Platforms for Learning
 - ▶ TekBots program for ECE students
- Paired with Beaversource
 - Combines coding and social networking Elgg + Trac
 - Easy introduction to open source communities
- Timeline
 - Project started in June 2008
 - Release first prototypes in December 2008
 - ► First deployment in April 2009
 - ► Large-scale deployment in October 2009
 - Project stalls mainly due to hardware issues in 2010



Design considerations

- Price
- Flexibility
- Manufacturability
- Openness



BeagleBoard used as a model platform

OSWALD hardware

- ARM processor
- ▶ 128 MB RAM
- ▶ 256 MB NAND
- ▶ 3.5" 320×240 display
- DVI out (up to 1024x768)
- Resistive touchscreen
- Touchpad
- ► 3-axis accelerometer

- 5-way rocker switch
- Speaker
 - Microphone
 - Built-in battery



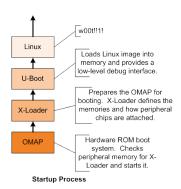
(a) CORE.0 board



(b) OSWALD



OSWALD software



- X-Loader and U-Boot bootloaders
- Linux kernel: OMAP tree + OSWALD patches
- Custom distribution: Radix
- Other platforms: Gentoo, Ångström,
 Poky, MeeGo, and Android
- Code repositories: code.oregonstate.edu/git
- Wiki: beaversource.oregonstate. edu/projects/cspfl

Composition of Radix

- Built using OpenEmbedded
- Lightweight and flexible
- Matchbox desktop
- GCC, Java (via jamvm), Python, Perl
- SDK available
- ► Emulation possible via gemu branch



Default Radix desktop

Applications available on Radix

Figure: Applications



(a) epdfview



(c) pcmanfm



(b) gpicview



(d) xournal

- Focus on lightweight applications
- Examples:
 - ► PDF viewer: epdfview
 - Image viewer: gpicview
 - File manager: pcmanfm
 - Drawing/writing: xournal
- Patched to work better within OSWALD environment
- Plenty of other software: mplayer, octave, etc



Gaming

Figure: Various games and emulators running on the OSWALD



(a) Opentyrian



(b) Freedoom



(c) Numpty Physics



(d) Snes9x



(e) Dosbox



(f) ScummVM

A hands-on approach to computer science

- Freshmen students acquire devices on entry into the CS program
- Devices are integrated into courses throughout the curriculum to enhance labs and lectures
- Help emulate real world development and programming environments for students

A hands-on approach to computer science

- Mainly used in introductory classes for a few terms
- Programming basics in Java, data structures, and a few other classes
- ▶ OSU Robotics club: used in the 2010 Mars rover design



In-house development

- Great for small projects and limited deployments
- Allows for better control over the platform experts close at hand
- Hard to fix problems on complex platforms quickly
- Disastrous when core development team moves on

Shoestring budget planning

- Bad hardware batches can quickly kill small projects
- Hard to communicate with 3rd party manufacturers
- Difficult to plan through hardware revisions

Finding external support

- ▶ Nearly impossible to find a vendor neutral solution
- Companies enjoy promising future devices
- Generally slow, similar to the grant process

Sponsors











Small development team

- Allowed for more experimental and reactive system building
- Communication network was quick and easy to navigate
- Hard to spend development time working on training new developers
- Difficult to spend time on important side projects — GUI development and theming, keyboard improvement, etc
- Building a community



Radix virtual keyboard

GUI hassles

Figure: Display problems



(a) GTK dialog



(b) File menu

- Small resolutions not supported well in many current GUI toolkits (desktop-centric apps are mainly to blame)
- Generally requires extensive theming and toolkit rework
- Most companies/projects create their own themes or toolkits for mobile interfaces

Prepare a solid software foundation, i.e. get someone else to do the work

- ▶ In the best case, use a well-tested platform with plenty of 3rd party support and documentation
- Outsource majority of development and documentation efforts to a community or larger project
- Focus efforts on creating educational content and related content modules

Poor classroom experiences are costly

- ► Teachers are very risk averse
- Mistakes and rushed deployments are extremely expensive
- Hard to regain entry and confidence in the classroom environment
- Students are turned off by poor experiences

Required infrastructure for successful deployments

- Have necessary support in place and trained before large deployments
- ▶ More lead-in time for professors and TAs for Q&A sessions and training before classroom usage
- Documentation and related support information a must
- Additional resources: debugging environments, basic IDE support, SDKs, and VMs

Extremely high barrier to entry for newcomers

- Average incoming student has minimal exposure to working with Linux let alone in embedded development environments
- Deploy devices from the top down
- Test runs with advanced/older students and trickle down as the project matures

OSWALD's future

- Current status: extremely small amount of working devices left in stock
- Project suspended while searching for funds and future platform
- ► Looking and interested in any ideas or suggestions towards possible devices

Searching for the perfect platform

- Information sharing, interoperability, user-centered design, and collaboration around open source education
- Well-supported 3rd party device with a relatively open platform
- Dream goal: Help create a platform used by multiple universities to develop open curriculum around open source and embedded development

Requirements for an open, educational platform

- Hardware: Rugged case, decent battery life, and well-supported chipset family
- Software: OS flexibility (no jailbreaking required)
- ▶ Environment: Support standard programming environments (Java and C/C++ for current classes)
- Connectivity: Wifi and USB host mode required
- Desired features: Easily modifiable case and rich set of I/O devices

Acknowledgements

Developers

Ben Goska, Kevin Kemper, Corbin Simpson, and all other unmentioned contributors

Project managers

Carlos Jensen, Don Heer, and others

Sponsors







