The Fedora ARM Project

Jon Masters <jcm@redhat.com>
Principal Software Engineer
Red Hat

LinuxCon Japan 2012

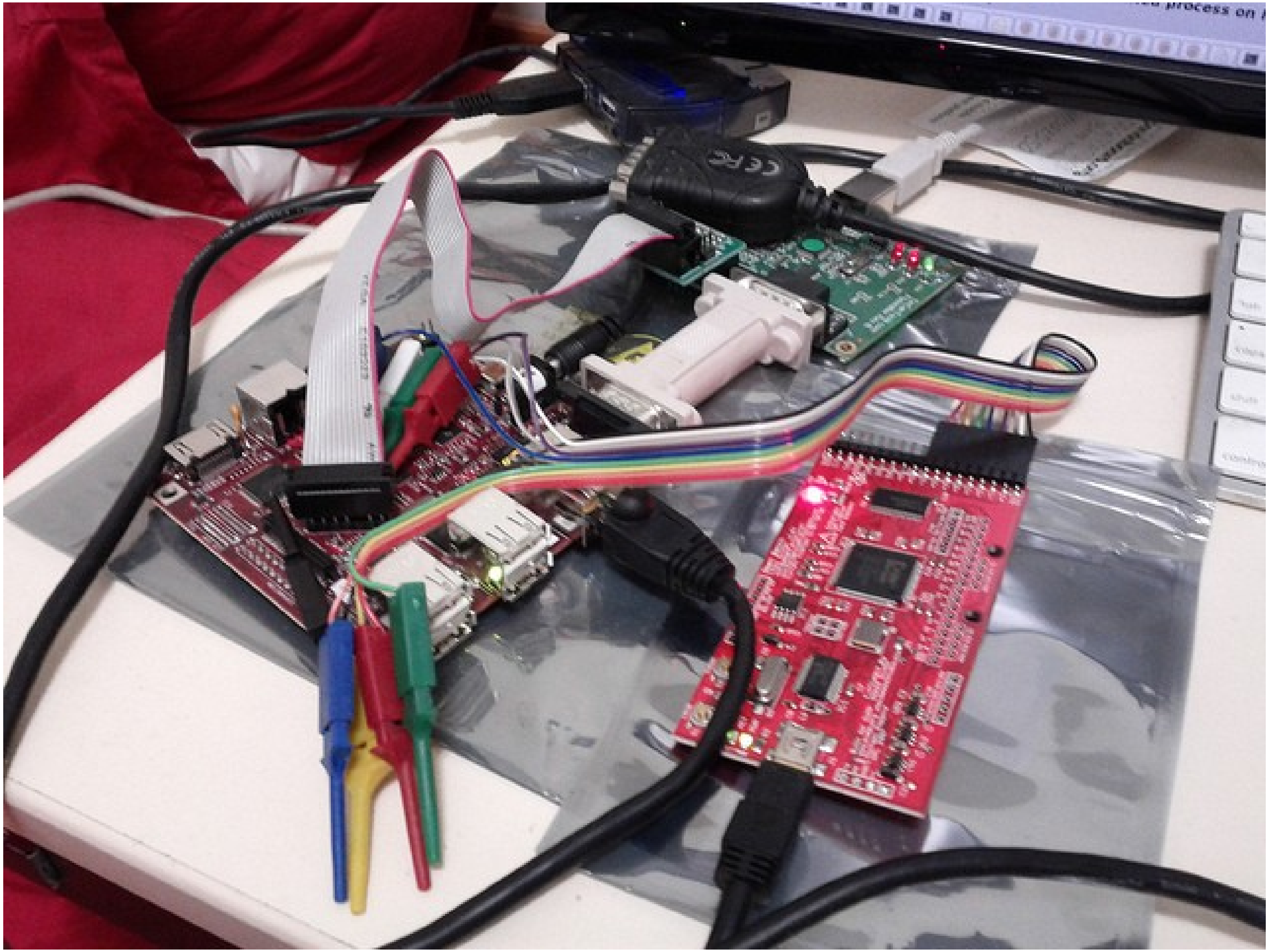

Image: BeagleBoard-xM powered by TI OMAP3, connected to Open Logic Sniffer

What is the Fedora ARM project?

- http://fedoraproject.org/wiki/Architectures/ARM
- A community effort to port Fedora to ARM Architecture
  - Goal is feature parity with x86 (“Primary Architecture”)
- Fedora supports ARMv5 and ARMv7 32-bit systems
  - Including low-cost PandaBoard, TrimSlice, etc.
  - Successfully tested on first generation Calxeda servers
  - F17 Beta is now available, final version coming soon
- Will support 64-bit ARM systems
  - Collaborating within the community on ARMv8|AArch64

Where do I get it?

- http://fedoraproject.org/wiki/Architectures/ARM
- Download pre-built images from our website
  - Many popular targets supported, more coming soon
- Embedded boards
  - Easy option is to “dd” onto a memory card and boot
  - Can also download filesystem images (tarballs)
- Server systems
  - Full installer support has been tested, and coming soon

#fedora-arm (Freenode) or arm@lists.fedoraproject.org

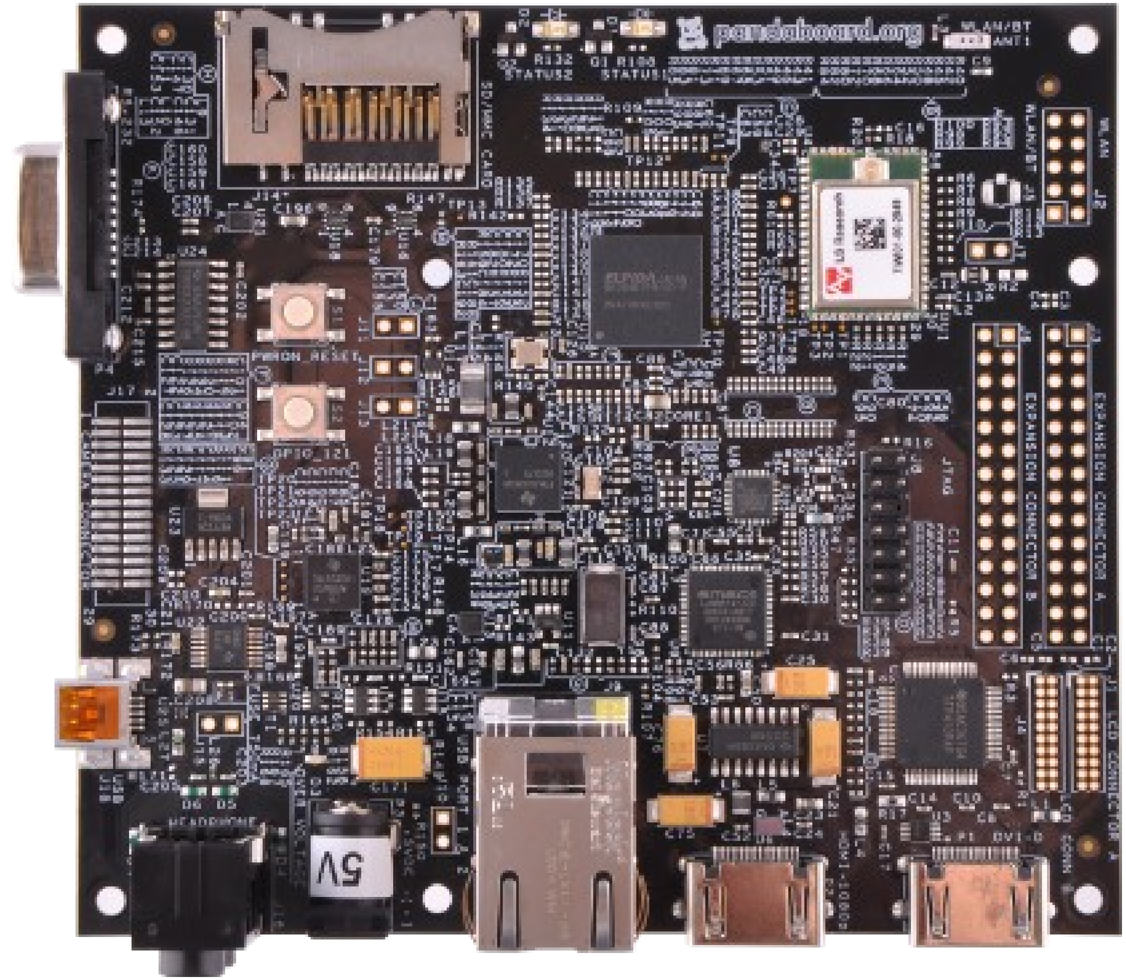

Image: PandaBoard powered by TI OMAP4 (source: http://www.pandaboard.org/)

How can I help?

- Try out our images without any risk or commitment
  - Report bugs (bugzilla.redhat.com, IRC, email, etc.)
- Help us to fix problems in existing Fedora packages
  - Problems with optional architecture features
  - Assumptions from other architectures (memory model)
  - Documentation and helping out newcomers is helpful
- Kernel developers can test bigger configurations
  - The “defconfig” option is only a minimum
- Port your software to ARM Architecture

Why does the Fedora ARM Project exist?

- The “post PC” era will be driven by low energy at scale
- Modern cellphones, tablets are replacing PC desktops
  - Multi-core, multi-GHz processors, with many GB RAM
  - Additional power provided by Cloud based services
- ARM helps power future “Hyperscale” Cloud systems
  - Densely packed Server-on-Chip heterogeneous designs
  - Failure-In-Place model of use (like display pixels)
- New architectures are fun to work with :)

Image: TrimSlice powered by Nvidia Tegra2 (source: http://www.trimslice.com/)

ARM Technology 101

- 4 Billion ARM cores shipped in Q1 2012
    - ARM pioneered the fabless IP-based licensing model
- ARM uses System-on-Chip (SoC) technology
    - Components traditionally on many chips integrated
    - IO Controllers, functional offload, co-processors
    - New heterogeneous features and “big.LITTLE”, etc.
- Many SoCs can be combined as independent systems
    - Scale-out rather than scale-up

ARM Technology 101

- ARM has nearly 30 years of history
  - Dating back to “BBC” school computing (see now rPi)
- Numerous architecture iterations
  - ARMv5 and ARMv7 current 32-bit
  - Cortex-A19|A15 (dual/quad clusters) core
  - 32-bit (40-bit physical addressing with A15+)
  - ARMv8 (64-bit) has been announced
- Most active Linux architecture by far

Image: OLPC XO-1.75 powered by Marvell Sheeva (source: www.laptop.org/)

ARM Technology 101

- Taming the ARM “zoo” is important for success
- Differentiated value through System-on-Chip
  - Allows novel solutions to problems
  - Improves power efficiency
  - A nightmare for Linux distributions
- Software platform diversity
  - ARMv7 “hard float” as a 32-bit standard
  - Options exist around ISA (T32 vs. A32)
  - Can optimize for SIMD, uArch choices, etc.

ARM Technology 101

- Standardizing the platform is essential
  - Linux and non-Linux vendors must collaborate
- Define a basic compute platform for the SoC
  - No value in differentiating over UARTS, etc.
  - Require certain core CPU features
- Enumerate installed devices within a system
  - DeviceTree and ACPI
- Provision using standardized tooling
  - UEFI for boot and OS installation
  - IPMI for remote control of systems

Fedora Technology 101

- Fedora is a Free and Open Source Linux distribution
  - Does not ship with proprietary drivers or firmware
  - Uses upstream Linux kernel on ARM and non-ARM
- Build system is build-on-target using “Koji”
  - Koji drives “mock”, which drives “rpmbuild”
  - Distribution is not cross-compiled (not embedded)
- ARM is a Secondary Architecture. Aims to be Primary
  - Proposal for the procedure for promotion adopted

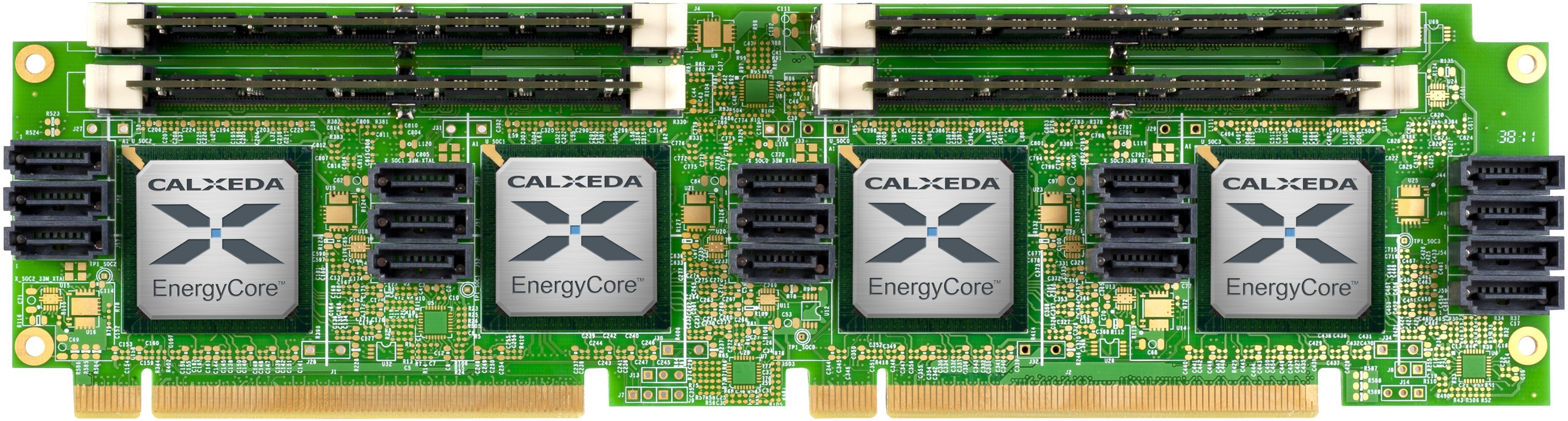

Image: Calxeda EnergyCard powered by Calxeda EnergyCore (source: calxeda.com/)

The past of the Fedora ARM Project

- The Fedora ARM Project is several years old
- Current version of the project is a collaboration
  - Initially lead by Seneca College (Toronto, Canada)
  - Members: Seneca, Fedora Community, Red Hat, You
- First modern release was Fedora 13 based
  - Fedora 14 onwards used by OLPC (XO-1.75)
  - Fedora 15 is still in active use (e.g. on builders)

The past of the Fedora ARM Project

- Fedora 15 was our first full ARM Architecture bootstrap
  - Treated ARMv7 (armv7hl, ARMv7-VFPv3-D16) as new
  - Intentional trial run preparing for ARMv8
- We maintain two ARM Architecture ports (versions)
  - ARMv5 “soft float” for older systems
  - ARMv7 “hard float” for newer systems
- Both use the “EABI”, but not the same ABI

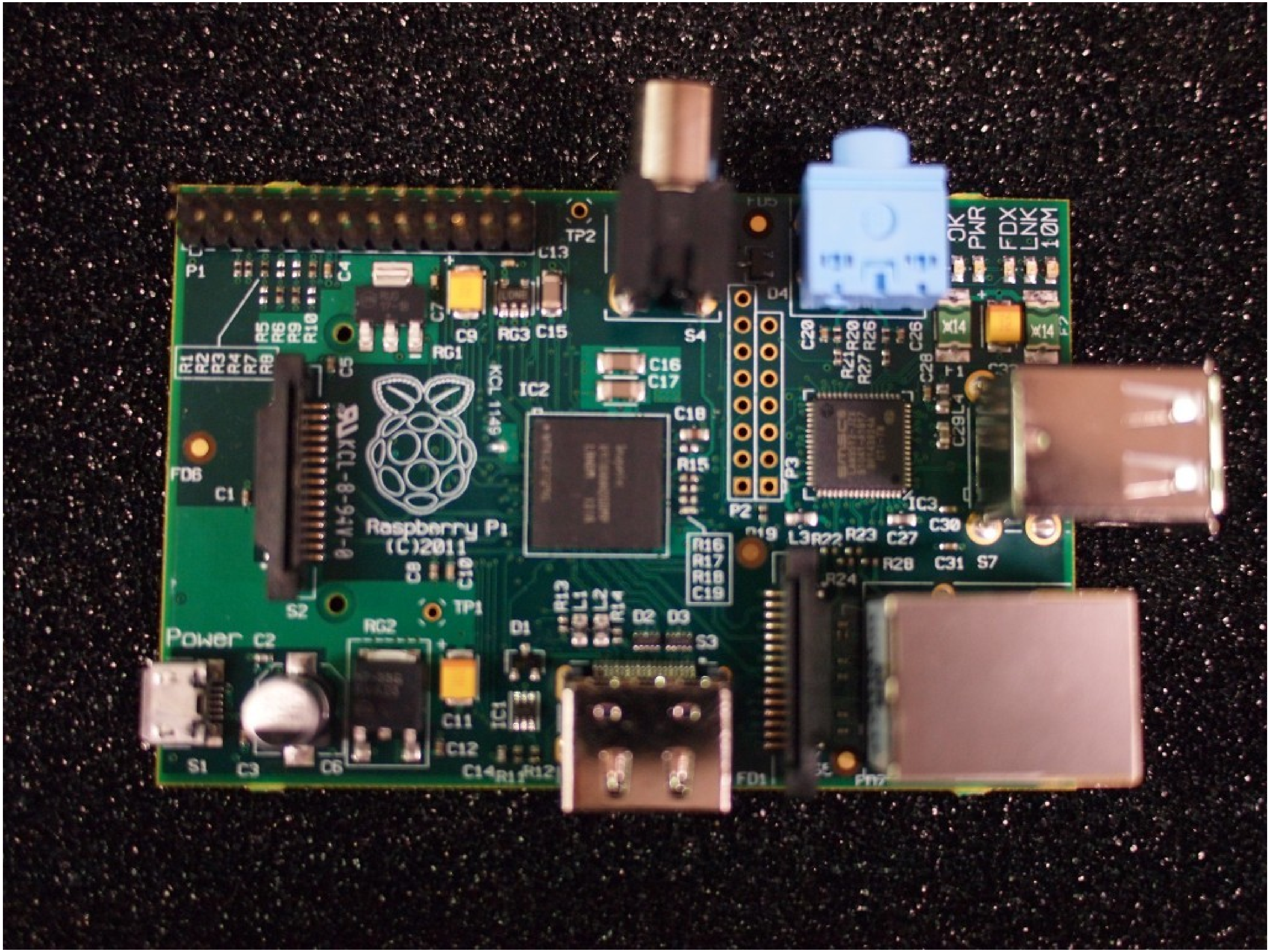

Image: Raspberry Pi powered by Broadcom BCM2835 (source: raspberrypi.org/)

The present of the Fedora ARM Project

- The current release is Fedora 17 (Beta)
- Nearly 100% software package parity with x86
  - A few remaining issues (e.g. ARMv5 atomics)
  - Installer solutions is being worked on
- Support for many devices, including:
  - TrimSlice (Nvidia Tegra)
  - BeagleBoard/BeagleBone/PandaBoard (TI OMAP)
  - Raspberry Pi (work in progress)
  - Many others (including first server systems)

The future of the Fedora ARM Project

- The Fedora ARM Project is just beginning
- We will support more devices over the coming year
  - Improvements in 3D graphics and tablets
  - Better support for ARM server systems
- We will support ARMv8 64-bit ARM systems
  - Working with key organizations, such as Linaro
  - Co-ordinating on standards

Image: Atlas, the Titan who supported the Heavens (source: wikipedia.org/)

The Fedora ARM Project for ARMv8

- ARMv7 “hard float” was in preparation for ARMv8
- Multi-stage bootstrap process is required
  - Stage 1 is cross-compilation
  - Stage 2 is native rebuild
  - Stage 3 is first rpmbuild
  - Stage 4 is full “mock”
  - Stage 5 is full “koji”
- Collaboration with other vendors is important
  - Need to avoid fragmentation