State of the U-Boot

Thomas Rini, Konsulko Group
History, in brief

- Separate projects of PPCBoot and ARMBoot, prior to November 2002.
- Merged, renamed to U-Boot, added x86
- Since then added more than 10 other architectures
- Wolfgang Denk as head custodian for over 10 years
- Tom Rini as head custodian since September 2012
- Lots more on Wikipedia
Community

- Over 20 companies and 110 individual developers every release in the last year
- A number of talks at various industry conferences
- Contributing back up to the Linux Kernel when we share code
Architecture and SoC support

- 32bit ARM
  - Atmel, Rockchip, Texas Instruments, NXP i.MX, Allwinner, Xilinx, UniPhier, Tegra, Marvell, STM32
- 64bit ARM
  - NXP Layerscape, Allwinner, Xilinx, UniPhier, Tegra, Marvell
- MIPS (Boston, Malta, etc)
- x86 (32 and 64bit, Baytrail, Broadwell, Quark, etc)

... and this is of course an incomplete list
Important Features

- SPL, Falcon Mode
- Cryptographic image support
  - Proprietary (TI, NXP) and not (FIT images)
- Generic distribution boot support
  - Fedora, Debian, others now, FreeBSD in progress
- EFI application support
Testing / CI

- travis-ci.org
- test.py
- tbot
- Coverity
- board farms
Testing / CI (Travis CI)

- Provides run-time-limited automated build and test instances.
- Able to build 97% of possible boards
- 10 QEMU-based test.py runs and sandbox
- Anyone can connect with their github and test prior to submission
Testing / CI (test.py)

- Based on pytest framework
- Works on real hardware, QEMU and sandbox
  - Target local and Target/Host tests
- We also have test/fs/fs-test.sh
  - FAT and ext2/3/4 tests
“tbot is a tool for executing testcases on boards”
Falls somewhere in between Jenkins and test.py
Heiko Schocher has a good video demonstration on youtube titled “tbot git bisect demo”
  - https://www.youtube.com/watch?v=zfpjp3DLsx4
Testing / CI (coverity)

- Community instance under “Das U-boot”
  - Limited to building for a single configuration, so sandbox
  - 45 defects in the last year
- Various vendors with commercial instances
Testing / CI (board farms)

- DENX
- Various private companies
My board farm
FlashAir, YKUSH and Relays

- FlashAir WiFi enabled SD cards
  - See more at http://konsulko.com/?p=1419
- YKUSH
  - See more at https://www.yepkit.com/products/ykush
- Relay
  - See more at http://www.robot-electronics.co.uk/htm/ethoo8tech.html
buildman, not MAKEALL

- The venerable MAKEALL script was retired in July 2016
- The replacement, buildman, was introduced in April 2013
- More flexible
  - Multiple architectures in a single command
  - Describe what to build in regex form
  - Size comparison
New tool for creating a functional output from one or more binaries

Uses device tree syntax to describe the output.

For example:
- x86 describes where to place U-Boot and various required other firmware entries
- Allwinner describes where to place SPL and then U-Boot in a single binary file
- aarch64 can use this to describe where to place ATF, U-Boot, etc.
Kbuild / Kconfig

- Kbuild, the make system from the Linux kernel, has been fully implemented for about 3 years.
- Kconfig transition, in progress since then.
  - Implementation is in-sync with v4.10
  - Emphasis on having logic in Kconfig files to ensure reasonable and minimal defconfig files
  - Start making use of the new imply keyword
In progress

- Driver Model and Device Tree
  - Including SPL
  - Including figuring out how to deal with the extremely resource constrained systems (smartweb, Ci20)

- Device Tree
  - Live tree
  - Being able to pass our tree to Linux
Near term goals

- Finish Kconfig migration this calendar year
- SPL + Linux and kexec? Happy to help!
- More test.py tests
- Strike up the conversation with kernelci.org again
- Find more time for stackoverflow questions
- Expand Coverity coverage