Redundant Booting with U-Boot

Welcome to the Redundancy Theater Playhouse

Thomas Rini

Overview

- Historically how redundancy has been developed and implemented
- What we have today
  - And have had for a while
- What we hope to have soon
- Sample use cases
  - Deployed product
  - Developers
- Example

Historically...

- One-off custom creations
- Hacks to U-Boot

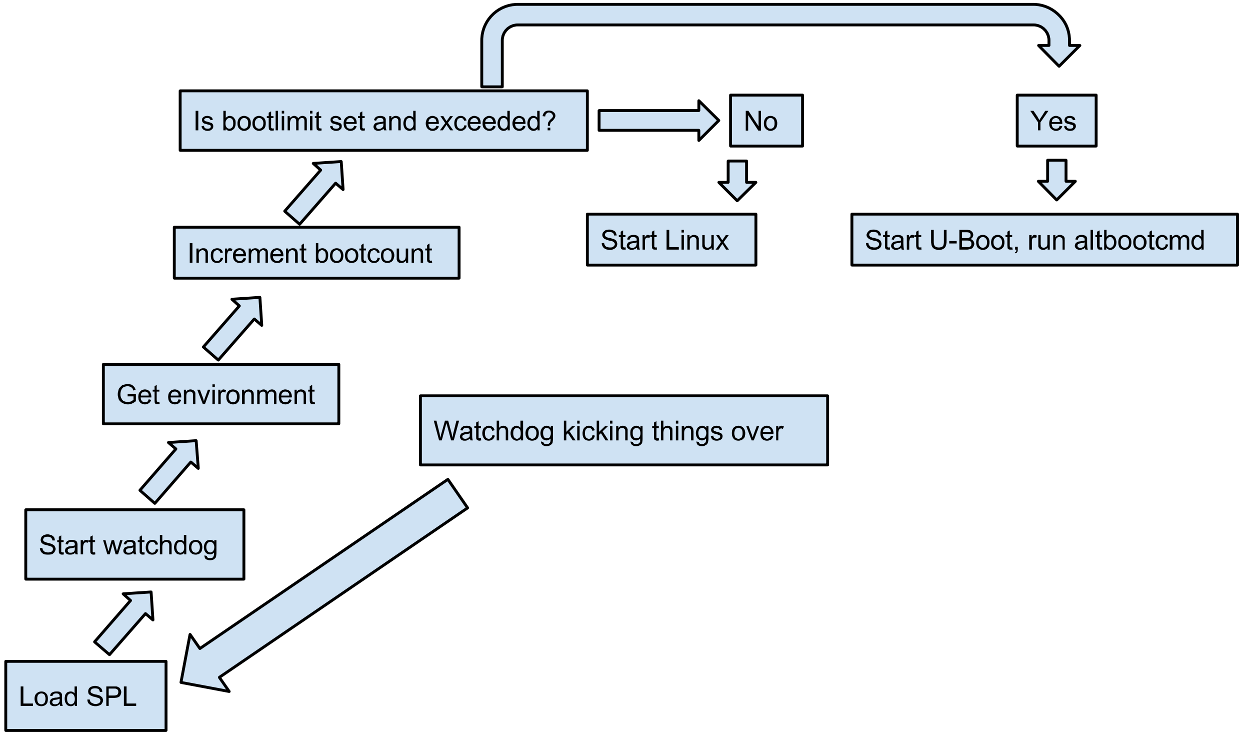
- And sometimes, working with upstream and leveraging existing features

Features Available Now (v2014.10)

- Redundant U-Boot environment
- Hardware Watchdogs
- Boot counting
  - Requires a “good” location to work with
- Cryptographic image signature checking
  - Software only, currently
- TPM (Trusted Platform Module) support
- “Falcon Mode”, aka SPL boots OS

Some of this is relatively new, much of it is not. We're working to address needs and enhance support, as developers come forward.

How it looks


Is bootlimit set and exceeded?
No
Yes
Increment bootcount
Start Linux
Start U-Boot, run altbootcmd
Get environment
Watchdog kicking things over
Start watchdog
Load SPL

Features In Progress

- Linux Kernel side of Boot counting
  - Have to clear the counter once the system decides it's up and stable
  - Work in Progress: http://goo.gl/ES0tYf
- Hardware / ROM Cryptographic image signature checking
- “Falcon Mode” enhancements
- SPL / Bootcount enhancements using environment
  - Work in Progress: https://github.com/trini/u-boot/tree/v2014.10-plus-spl-bootcount
- More FS support
  - Read from extN not just FAT

Features that have been talked about

- Wider environment use in Falcon Mode
  - Today “FAT” supports environment saying what to load, but not “raw” modes
- zImage support and/or FIT image support
  - Today only legacy uImages are supported
  - Both of these would require additional work to know where to put the payloads (in some cases)
- Integrate altbootcmd into Falcon Mode
  - Today when bootcount is exceeded we fall back to full U-Boot, but in some cases we may not need to.

Sample Use Cases: Deployed Products

- ROM provides a level of security and redundancy
- SPL OS boot provides quick path into the Linux kernel
- Redundant environment is consulted for where to find what to boot, how many times to try
- A fail-safe alternative exists as backup (failed upgrades, etc)
- Watchdog is enabled to reset the board when things have gone bad
- Images are cryptographically signed
- A TPM is enabled to allow for only trusted upgrades to happen

Sample Use Cases: Developer

- SPL OS boot enabled, environment consulted for what Linux kernel to be booting and testing
- Watchdog enabled, relatively short timeout set
  - Catch failure quicker
- Bootlimit is set low, just 1 or 2
  - Again, catch failure quicker
- Environment again points to a known working backup image to use, for when things fail
- fw_setenv in Linux to point at new test images / etc

Development cycle improved, fallback available without external hardware.

Example

- BeagleBone Black
  - Watchdog Support
  - SPL OS boot enabled
  - Bootlimit is set, clearable from the kernel
  - Persistent Environment (redundant)
    - fw_setenv/getenv supports this