

"Automated Test System - build blocks"

Łukasz Majewski

Samsung R&D Institute Poland

Embedded Linux Conference Europe
Dublin, 05-10-2015

Outline

- 1 Introduction
- 2 System description
- 3 Building blocks
- 4 Summary
- 5 Discussion

Introduction

A few words about me

- Embedded systems programmer
- Using u-boot since 2008
- "Blame me" for:
 - Device Firmware Upgrade (DFU)
 - Trats/Trats2 devices

Presentation goal

Presentation goal

Sharing knowledge and experience after building simple and cost effective setup for automated u-boot and kernel tests with using open source SW and off-the-shelf HW.

Testing our u-boot

- 2011: Build test infrastructure (u-boot-CI)

Testing our u-boot

- 2011: Build test infrastructure (u-boot-CI)
- 2014: USB DFU/USB Mass Storage test scripts in mainline

Testing our u-boot

- 2011: Build test infrastructure (u-boot-CI)
- 2014: USB DFU/USB Mass Storage test scripts in mainline
- 2014: Boot and flashing tests - HardWare Test (HWT) v1

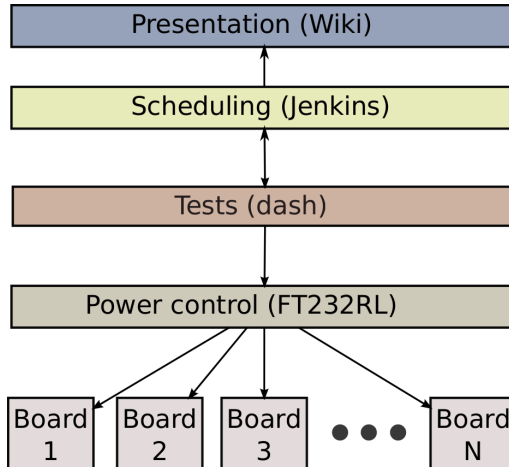
Testing our u-boot

- 2011: Build test infrastructure (u-boot-CI)
- 2014: USB DFU/USB Mass Storage test scripts in mainline
- 2014: Boot and flashing tests - HardWare Test (HWT) v1
- 2015: HWT v2

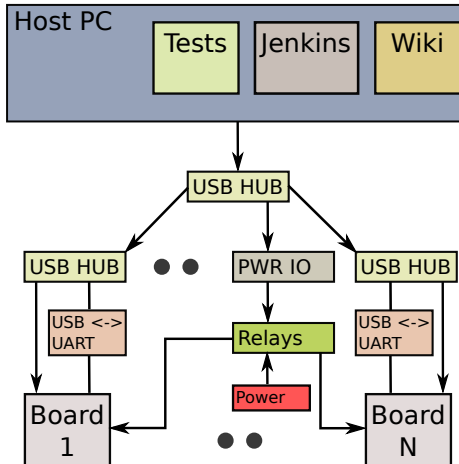
Automated Test System



Automated Test System - overview

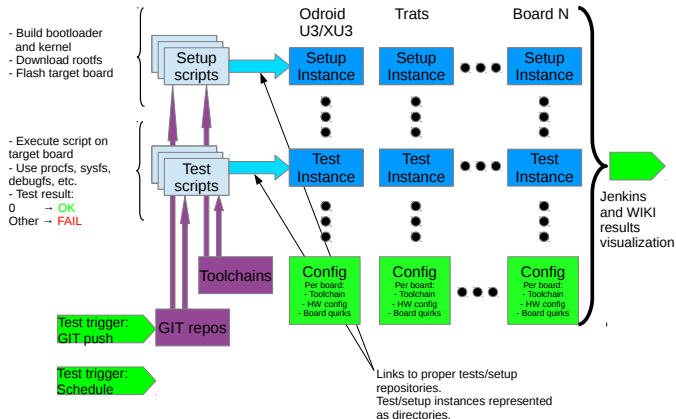


Automated Test System - block diagram



System description

Architecture



Key design decisions

- Why: New vs. Reuse

Key design decisions

- Why: New vs. Reuse
- SW: Python vs. Shell

Key design decisions

- Why: New vs. Reuse
- SW: Python vs. Shell
- Console: Python (pexpect) vs. expect (CLI program)

Key design decisions

- Why: New vs. Reuse
- SW: Python vs. Shell
- Console: Python (pexpect) vs. expect (CLI program)
- Flashing: Ethernet (TFTP) vs. USB (DFU)

Key design decisions

- Why: New vs. Reuse
- SW: Python vs. Shell
- Console: Python (pexpect) vs. expect (CLI program)
- Flashing: Ethernet (TFTP) vs. USB (DFU)
- Scheduling: Jenkins

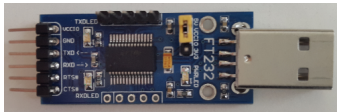
Key design decisions

- Why: New vs. Reuse
- SW: Python vs. Shell
- Console: Python (pexpect) vs. expect (CLI program)
- Flashing: Ethernet (TFTP) vs. USB (DFU)
- Scheduling: Jenkins
- Presentation: Wiki

Building blocks

Power control

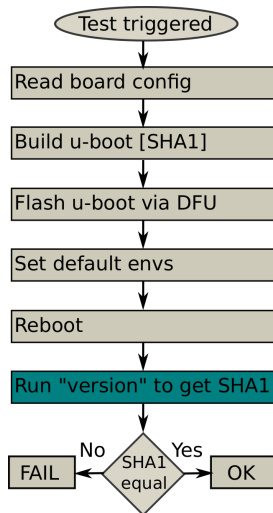
- FTDI FT232RL
 - **Note:** https://github.com/lmajewski/HWT_io_ctl.git
- Relays
 - **Note:** GPIO-controlled relays module for Arduino
- USB HUB
 - **Note:** External and stable power supply



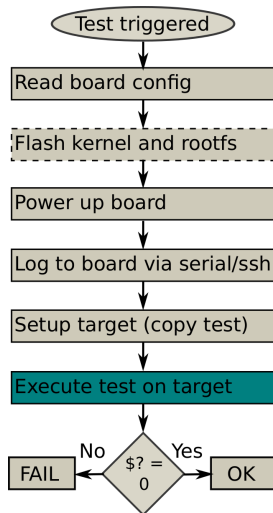
Host PC

- Ubuntu Long Term Support (LTS)
- USB udev configuration
- Runs:
 - Dash tests
 - Jenkins client
 - Repository mirroring
 - MediaWiki

Host PC - Test U-boot booting



Host PC - Test kernel



Host PC - Jenkins [optional for simple setup]

- Responsible for scheduling tests
- Provides more sophisticated functionality to the test setup:
 - Gerrit push trigger
 - Repository/branch change trigger
- For simple test system it can be omitted - cron managed test triggering with results uploaded to wiki

Host PC - Wiki

2015-09-14 03:21

Scenario	Status	Repository	Branch	Image	Log
KSFW_HWT_HQ_SCENARIO_1	OK	linux-arm64-exynos	exynos-next-4.1	tizen-mobile_20150914.1_mobile-wayland-arm64-n4.tar.gz [1]	[2]
KSFW_HWT_HQ_SCENARIO_2	OK	linux-arm64-exynos	exynos-next-4.1	tizen-mobile_20150914.1_mobile-wayland-arm64-n4.tar.gz [3]	[4]
KSFW_HWT_HQ_SCENARIO_3	OK	linux-arm64-exynos	exynos-next-4.1	tizen-mobile_20150914.1_mobile-wayland-arm64-n4.tar.gz [5]	[6]
KSFW_HWT_HQ_SCENARIO_4	OK	linux-arm64-exynos	exynos-next-4.1	tizen-mobile_20150914.1_mobile-wayland-arm64-n4.tar.gz [7]	[8]
KSFW_HWT_HQ_SCENARIO_5	OK	linux-arm64-exynos	exynos-next-4.1	tizen-mobile_20150914.1_mobile-wayland-arm64-n4.tar.gz [9]	[10]
KSFW_HWT_HQ_SCENARIO_6	OK	linux-arm64-exynos	exynos-next-4.1	tizen-mobile_20150914.1_mobile-wayland-arm64-n4.tar.gz [11]	[12]
KSFW_HWT_HQ_SCENARIO_8	OK	linux-arm64-exynos	exynos-next-4.1	tizen-mobile_20150914.1_mobile-wayland-arm64-n4.tar.gz [13]	[14]
KSFW_HWT_HQ_SCENARIO_9	OK	linux-arm64-exynos	exynos-next-4.1	tizen-mobile_20150914.1_mobile-wayland-arm64-n4.tar.gz [15]	[16]
KSFW_HWT_HQ_SCENARIO_10	OK	linux-arm64-exynos	exynos-next-4.1	tizen-mobile_20150914.1_mobile-wayland-arm64-n4.tar.gz [17]	[18]
KSFW_HWT_HQ_SCENARIO_11	OK	linux-arm64-exynos	exynos-next-4.1	tizen-mobile_20150914.1_mobile-wayland-arm64-n4.tar.gz [19]	[20]
KSFW_HWT_HQ_SCENARIO_12	FAILED	linux-arm64-exynos	exynos-next-4.1	tizen-mobile_20150914.1_mobile-wayland-arm64-n4.tar.gz [21]	[22]
KSFW_HWT_HQ_SCENARIO_13	OK	linux-arm64-exynos	exynos-next-4.1	tizen-mobile_20150914.1_mobile-wayland-arm64-n4.tar.gz [23]	[24]
KSFW_HWT_HQ_SCENARIO_14	OK	linux-arm64-exynos	exynos-next-4.1	tizen-mobile_20150914.1_mobile-wayland-arm64-n4.tar.gz [25]	[26]

Minimal HW costs

Device	Price [EUR]	Remarks
FT232RL	10	
Relays	5	
USB HUB	40	External power
Host PC	0	Old reused
Cables/connectors	10	
Power supply	0	Included with target board
Total	65	

Summary

Solved problems

- Problem: Breaking the default boot

Solved problems

- Problem: Breaking the default boot
 - Solution: Use C/C++ with nanosleep()

Solved problems

- Problem: Breaking the default boot
 - **Solution:** Use C/C++ with nanosleep()
- Problem: Jenkins logs readability

Solved problems

- Problem: Breaking the default boot
 - Solution: Use C/C++ with nanosleep()
- Problem: Jenkins logs readability
 - Solution: Concatenate logs and link them to Wiki

Solved problems

- Problem: Breaking the default boot
 - **Solution:** Use C/C++ with nanosleep()
- Problem: Jenkins logs readability
 - **Solution:** Concatenate logs and link them to Wiki
- Problem: Target power control

Solved problems

- Problem: Breaking the default boot
 - Solution: Use C/C++ with nanosleep()
- Problem: Jenkins logs readability
 - Solution: Concatenate logs and link them to Wiki
- Problem: Target power control
 - Solution: HW rework to also control VUSB

Solved problems

- Problem: Breaking the default boot
 - **Solution:** Use C/C++ with nanosleep()
- Problem: Jenkins logs readability
 - **Solution:** Concatenate logs and link them to Wiki
- Problem: Target power control
 - **Solution:** HW rework to also control VUSB
- Problem: Lack of persistent USB configuration

Solved problems

- Problem: Breaking the default boot
 - Solution: Use C/C++ with nanosleep()
- Problem: Jenkins logs readability
 - Solution: Concatenate logs and link them to Wiki
- Problem: Target power control
 - Solution: HW rework to also control VUSB
- Problem: Lack of persistent USB configuration
 - Solution: Test system connectivity managed by udev rules, which use USB device serial numbers

Found bugs:

- u-boot: SD card timeout for writing data
- u-boot: Wrong error status passing from FAT to upper layers
- Linux: "rootwait" problem in Trats/Trats2

Conclusion

- YES, it was worth to build this setup.

Conclusion

- **YES**, it was worth to build this setup.
- Benefits:
 - Tizen.org: build, boot and functionality tests
 - u-boot-dfu: build, boot (u-boot and kernel), USB: Mass Storage, DFU, THOR

Discussion