Mainline u-boot for Tizen - “war stories”

Łukasz Majewski

Samsung R&D Institute Poland

Embedded Linux Conference Europe
Düsseldorf, 13-10-2014
Outline

1. Introduction
2. "War stories"
3. Cooperation with community
4. Discussion
Introduction
A few words about me

- Embedded systems programmer
- Using u-boot since 2008
- "Blame me" for:
  - PMIC framework (v1/v2)
  - Device Firmware Upgrade
  - trats2/trats
  - OneNAND
A few words about Tizen

- Open source, Linux based OS
- Samsung Linux Platform + MeeGo + BADA API
- Developed by community with main support from Samsung and Intel
- Multiple profiles based on Tizen Common:
  - Tizen In-Vehicle Infotainment
  - Tizen Mobile
  - Tizen TV
  - Tizen Wearable
Why do we need u-boot?

- Providing reliable way of booting Linux on a variety of boards by using Flattened Image Tree
- Updating on-board persistent memory (DFU, THOR, Mass Storage)
- Recovering to the "default" state
- Facilitating development
Why do we need u-boot?

- Providing reliable way of booting Linux on a variety of boards by using Flattened Image Tree
- Updating on-board persistent memory (DFU, THOR, Mass Storage)
- Recovering to the "default" state
- Facilitating development

**GOAL**: Mainline u-boot on Tizen devices!
Samsung - history of involvement

- Boards:
  - SLP: smdkc100 (Q3'09), goni (Q2'10) and universal c210 (Q1'11)
  - Tizen: trats (Q1'12), trats2 (Q3'12) and odroidU3/X2 (Q3'14)
- PMIC framework (Q3’11)
- Porting UDC driver to u-boot (Q4’11)
- USB Gadget infrastructure (Q2’12)
- DFU (Q3’12)
- GUID Partition Table (Q4’12)
- Mass Storage (Q1’13)
- Universally Unique ID (Q2’14)
Samsung - contribution chart
Present situation

- **Mainline - full functional support**
  - trats, trats2 and odroid U3/X2 development boards supported
Present situation

- **Mainline - full functional support**
  - trats, trats2 and odroid U3/X2 development boards supported

- **tizen.org - minor enhancements**
  - One u-boot for trats2 and odroid U3/X2
  - Special animations which appear when charging battery on trats2
USB Gadget

Problem:

- No easy way to develop USB gadgets
Problem:
- No easy way to develop USB gadgets

Solution:
- Reuse established Linux composite API (2.6.36)
- Reuse available Linux gadgets (like Mass Storage)
L1 cache

Motivation:

- Reduce time needed for start of a system
L1 cache - Problem

Program code

int i
int tab[56]

Main memory

int k

Cache L1 (line size 64B)

Łukasz Majewski
Mainline u-boot for Tizen - “war stories”
L1 cache - Problem

```c
int i
0x100
int tab[56]
Program code
0x13C
Main memory
0x104
int k
0x140
Cache L1 (line size 64B)
```
L1 cache - Problem

Program code

```c
int i
int tab[56]
int k
```

Main memory

Cache L1 (line size 64B)
L1 cache - Problem

Program code

```
int i
int tab[56]
```

Main memory

```
0x100 0x104
```

Cache L1 (line size 64B)

```
0x0 0x13C
```

DATA

```
0x0 0x100
```

i, k - corrupted

```
0x104 0x140
```
L1 cache - Solution

- `memalign()`
- `*_CACHE_ALIGN_BUFFER()`
L1 cache - Solution

- `memalign()`
- `*_CACHEALIGNBUFFER()`

Outcome:
- Normal u-boot operation (booting kernel) - speedup \( \sim 16\% \)
Power Management framework (PMIC)

Motivation:

- Provide means to charge and monitor battery status
Power Management framework (PMIC)

Motivation:
- Provide means to charge and monitor battery status

PMIC v2 features:
- Simple hierarchy
- Support for I2C, SPI, Big/Little endian, byte, half-word and word transfers
- Reduce power consumption at charging (reduce CPU and bus CLK, disable LDOs)
Power Management framework (PMIC)

Motivation:
- Provide means to charge and monitor battery status

PMIC v2 features:
- Simple hierarchy
- Support for I2C, SPI, Big/Little endian, byte, half-word and word transfers
- Reduce power consumption at charging (reduce CPU and bus CLK, disable LDOs)

PMIC v2 limitations:
- One instance per PMIC device
- No support for driver model
Power Management framework (PMIC)

Motivation:
- Provide means to charge and monitor battery status

PMIC v2 features:
- Simple hierarchy
- Support for I2C, SPI, Big/Little endian, byte, half-word and word transfers
- Reduce power consumption at charging (reduce CPU and bus CLK, disable LDOs)

PMIC v2 limitations:
- One instance per PMIC device
- No support for driver model

Solution:
- PMIC v3.
USB transmission speed

Motivation:
- Considerable time was needed to download data via USB.
USB transmission speed

Motivation:
- Considerable time was needed to download data via USB.

Solution:
- Thorough profiling
- Remove unnecessary cache operations in the s3c_hsocg UDC
- Allow burst transfers for non EP0 endpoints
- Zero copy approach between UDC driver and gadgets
USB transmission speed

Motivation:
- Considerable time was needed to download data via USB.

Solution:
- Thorough profiling
- Remove unnecessary cache operations in the s3c_hsocg UDC
- Allow burst transfers for non EP0 endpoints
- Zero copy approach between UDC driver and gadgets

Outcome:
- Transmission speed improved from 9 MiB/s to 27 MiB/s
Cooperation with community
Social interaction

- Promptly reply to e-mails
- Even when you are buried in work give a firm deadline when you will review patches
- "Keep calm and be patient" - do not get easily angry
Code review

- Send patches early (as RFC) to receive more feedback
- Provide detailed documentation
- Include test scripts for your code
- Prepare patches for mailing list by using patman and buildman
Benefits

- Cooperation with the community does work in the long term
- Code developed by one company is reviewed, tested and enhanced by others
  - DFU: community contribution: 37%
  - PMIC v2: community contribution: 68%
- Less effort is needed to maintain common code
Future plans

- Send patches for multiplatform u-boot
- Development of PMIC v3
- Complete switch to driver model
Discussion
Thank you!

Łukasz Majewski
Mainline u-boot for Tizen - “war stories”