

User Guide

Tizen on SDTA7D

Ver.1.0

Revision History

– Release Notes

Date	Version	Release Notes
2019.03.29	Ver.1.0	First version released based on SDTA7D

– Contact

- Website: <https://www.sigma-delta.tech>
- Email: sales@sigma-delta.tech
- Github Link: <https://github.com/SigmaDeltaTechnologiesInc>

Table of Contents

1. Build for SDTA7D	4
1.1. Source list for SDTA7D	4
1.2. Install packages	4
1.3. GBS config file & Kickstart file	4
1.4. GBS build & Create image	5
2. Fusing boot image (u-boot, kernel)	6
2.1. Fusing Bootloader	6
2.2. Fusing Kernel	8
3. Fusing Tizen platform image, install RAMDISK and install firmware	11
3.1. Tizen platform image fusing with SDTA7D	11
3.2. Copy ramdisk image to boot partition and set ramdisk address variable on u-boot	11
4. Testing Tizen	12
4.1. Wi-Fi test	12

1. Build for SDTA7D

1.1. Source list for SDTA7D

1.1.1. List of sources

- u-boot
- kernel
- building-blocks
- peripheral-bus
- peripheral-io
- smarthing-plugin
- systemd
- wlandrv-plugin-tizen-ap6335
- bluetooth-firmware-bcm

1.1.2. Github list for each SDTA7D sources

- <https://github.com/SigmaDeltaTechnologiesInc/u-boot.git>
- <https://github.com/SigmaDeltaTechnologiesInc/linux-stable.git>
- <https://github.com/SigmaDeltaTechnologiesInc/building-blocks.git>
- <https://github.com/SigmaDeltaTechnologiesInc/peripheral-bus.git>
- <https://github.com/SigmaDeltaTechnologiesInc/peripheral-io.git>
- <https://github.com/SigmaDeltaTechnologiesInc/systemd.git>
- <https://github.com/SigmaDeltaTechnologiesInc/smarthing-plugin.git>
- <https://github.com/SigmaDeltaTechnologiesInc/wlandrv-plugin-tizen-ap6335.git>
- <https://github.com/SigmaDeltaTechnologiesInc/bluetooth-firmware-bcm.git>

1.2. Install packages

```
$ cd ~/Tizen-sdta7d
```

- Clone source for SDTA7D

```
$ git clone https://github.com/SigmaDeltaTechnologiesInc/u-boot.git
$ git clone https://github.com/SigmaDeltaTechnologiesInc/linux-stable.git
$ git clone https://github.com/SigmaDeltaTechnologiesInc/building-blocks.git
$ git clone https://github.com/SigmaDeltaTechnologiesInc/peripheral-bus.git
$ git clone https://github.com/SigmaDeltaTechnologiesInc/peripheral-io.git
$ git clone https://github.com/SigmaDeltaTechnologiesInc/systemd.git
$ git clone https://github.com/SigmaDeltaTechnologiesInc/smarthing-plugin.git
$ git clone https://github.com/SigmaDeltaTechnologiesInc/wlandrv-plugin-tizen-ap6335.git
$ git clone https://github.com/SigmaDeltaTechnologiesInc/bluetooth-firmware-bcm.git
```

1.3. GBS config file & Kickstart file

- Prepare GBS config file and Kickstart file (please request to your partner)
.gbsSdta7d.conf
tizen-unified_iot-boot-armv7l-sdta7d.ks

tizen-unified_iot-headless-2parts-armv7l-sdta7d.ks

1.4. GBS build & Create image

```
$ sudo gbs -c ~/.gbsSdta7d.conf build -A armv7l --include-all

$ sudo mic cr loop tizen-unified_iot-boot-armv7l-sdta7d.ks --pack-to=tizen-unified_iot-boot-armv7l-
sdta7d.tar.gz --logfile=mic_build.log -A armv7l

$ sudo mic cr loop tizen-unified_iot-headless-2parts-armv7l-sdta7d.ks --pack-to=tizen-unified_iot-
headless-2parts-armv7l-sdta7d.tar.gz --logfile=mic_build.log -A armv7l
```

- the final image at ~/Tizen-sdta7d/mic-output

tizen-unified_iot-boot-armv7l-sdta7d.tar.gz

tizen-unified_iot-headless-2parts-armv7l-sdta7d.tar.gz

2. Fusing boot image (u-boot, kernel)

Fusing newly built kernel, u-boot and tizen platform image fusing (first image)

2.1. Fusing Bootloader

2.1.1. prepare imx_usb_loader

```
$ cd ~/tools
$ git clone https://github.com/boundarydevices/imx_usb_loader
$ cd ~/tools/imx_usb_loader/
$ make
```

2.1.2. change to USB boot mode(configure by Jump on board)

- DAP Station for SDTA7D : jumper setting
 - x00
 - x00
 - x00
 - x00

2.1.3. power on board and check lsusb

```
$ lsusb | grep Freescale
Bus 00x Device 00y: ID 15a2:0076 Freescale Semiconductor, Inc
```

2.1.4. Load the SPL binary via USB

```
$ cd ~/Tizen-sdta7d/mic-output
$ tar zxvf tizen-unified_iot-boot-armv7l-sdta7d.tar.gz
modules.img
u-boot.img
SPL
sdta7d.dtb
zImage

$ tar zxvf tizen-unified_iot-headless-2parts-armv7l-sdta7d.tar.gz
system-data.img
rootfs.img
ramdisk.img
ramdisk-recovery.img
$ sudo ~/tools/imx_usb_loader/imx_usb SPL
```

2.1.5. Load the u-boot.img binary via USB

```
$ sudo ~/tools/imx_usb_loader/imx_usb u-boot.img
```

2.1.6. Using any key, break into U-Boot Console

- U-Boot console
SERIAL_CONSOLE_U_BOOT
Loading Environment from MMC... OK
In: serial
Out: serial
Err: serial
Net: FEC0
Hit any key to stop autoboot: 0

2.1.7. Use the default environment variables

- U-Boot console

```
=> env default -f -a  
=> saveenv
```

2.1.8. Configure mmc 0 to boot from the data partition, and disable access to boot partitions

- U-Boot console

```
=> mmc partconf 0 0 7 0
```

2.1.9. Run the DFU agent so we can flash the new images using dfu-util tool

- Please refer <http://dfu-util.sourceforge.net/> to install dfu-util tool
- U-Boot console

```
=> dfu 0 mmc 0
```

- If you have error, you must create partition by (14)
- Ubuntu host machine console

```
$ cd ~/tizen-imx7/u-boot  
$ sudo dfu-util -D SPL -a spl  
$ sudo dfu-util -R -D u-boot.img -a u-boot
```

2.1.10. Change to EMMC boot mode(configure by Jump on board)

- DAP Station for SDTA7D : jumper setting
X00
00X
X00
X00

2.1.11. Reset

2.1.12. Change to USB Mass Storage mode

- U-Boot console

```
=> ums 0 mmc 0
```

2.1.13. Verify USB Mass Storage device

- Ubuntu host machine console

```
$ cd ~/
$ lsusb | grep Netchip
```

- Bus 00x Device 00y: ID 0525:a4a5 Netchip Technology, Inc. Pocketbook Pro 903

2.1.14. Create first partition for DFU fail

- DFU needs a partition table to operate. Let's create one as a temp, just to get things loaded:
- First run on the SDTA7D:

```
=> ums 0 mmc 0
```

- Next, on your development machine: (replace DISK with /dev/sdX, look at lsblk and use the new driver that's the SDTA7D.)

```
$ sudo dd if=/dev/zero of=${DISK} bs=1M count=10

$ sudo sfdisk ${DISK} <<-__EOF__
$ 1M,,L,*
$ __EOF__

$ sudo mkfs.ext4 -L rootfs ${DISK}1
```

- Now in u-boot kill the “ums 0 mmc 0” and rerun the “dfu 0 mmc 0.”

2.2. Fusing Kernel

2.2.1. Check USB mass storage device on host ubuntu machine

- Ubuntu host machine console

```
$ lsblk
  NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
  sdb           8:16   1    3.6G  0 disk
  çuçwsdb4     8:20   1     1K  0 part
  çuçwsdb2     8:18   1   128M  0 part
  çuçwsdb5     8:21   1     1G  0 part
  çuçwsdb3     8:19   1     2G  0 part
  çuçwsdb1     8:17   1    32M  0 part
  ç|çwsdb6     8:22   1   410M  0 part
  sda           8:0     0 160.1G  0 disk
  çuçwsda2     8:2     0    4.5G  0 part
  ç|çwsda1     8:1     0 155.6G  0 part /
```

2.2.2. export variable

- Check version number on config/kernel.release : 4.20.6-0-sdta7d
- Check usb storage by lsblk : /dev/sdc

```
$ export kernel_version=4.20.6-0-sdta7d
$ export DISK=/dev/sdc
```


2.2.3. Erase partition

- Ubuntu host machine console

```
$ sudo dd if=/dev/zero of=${DISK} bs=1M count=10
```

2.2.4. Install new Bootloader

- Ubuntu host machine console

```
$ sudo dd if=SPL of=${DISK} seek=1 bs=1k  
$ sudo dd if=u-boot.img of=${DISK} seek=69 bs=1k
```

2.2.5. Partitioning eMMC

- Ubuntu host machine console

```
$ sudo ./mkpart.sh ${DISK}
```

- Recommend change boot partiton to FAT16 from ext4, because ext4 partion is very slow in boot time
- You can use gparted Ubuntu tool

2.2.6. Mount boot partition

- Ubuntu host machine console

```
$ sudo mount ${DISK}1 /media/hbahn/boot/
```

- It will be automatically mounted

2.2.7. Setup etlinux.conf

- Ubuntu host machine console

```
$ sudo mkdir -p /media/hbahn/boot/extlinux/  
$ sudo sh -c "echo 'label Linux ${kernel_version}' > /media/hbahn/boot/extlinux/extli-  
nux.conf"  
$ sudo sh -c "echo '      kernel /boot/vmlinuz-${kernel_version}' >> /media/hbahn/boot/ex-  
tlinux/extlinux.conf"  
$ sudo sh -c "echo '      append root=/dev/ram rw' >> /media/hbahn/boot/extlinux/extlinux.  
conf"  
$ sudo sh -c "echo '      fdt dir /boot/dtbs/${kernel_version}/' >> /media/hbahn/boot/ext-  
linux/extlinux.conf"  
$ sudo sh -c "echo '      initrd /boot/ramdisk.img' >> /media/hbahn/boot/extlinux/extli-  
nux.conf"
```

2.2.8. Copy Kernel Image

- Ubuntu host machine console

```
$ sudo mkdir -p /media/hbahn/boot/boot  
$ sudo cp -v zImage /media/hbahn/boot/boot/vmlinuz-${kernel_version}
```

2.2.9. Copy Kernel Device Tree Binaries

- Ubuntu host machine console

```
$ sudo mkdir -p /media/hbahn/boot/boot/dtbs/${kernel_version}/  
$ sudo cp sdta7d.dtb /media/hbahn/boot/boot/dtbs/${kernel_version}/
```

2.2.10. Copy kernel modules

- Ubuntu host machine console

```
$ sudo dd if=modules.img of=${DISK}2 bs=1M
```

3. Fusing Tizen platform image, install RAMDISK and install firmware

3.1. Tizen platform image fusing with SDTA7D

- Ubuntu host machine console

```
$ sudo dd if=rootfs.img of=${DISK}3 bs=1M  
$ sudo dd if=system-data.img of=${DISK}5 bs=1M
```

3.2. Copy ramdisk image to boot partition and set ramdisk address variable on u-boot

- Ubuntu host machine console

```
$ sudo cp ramdisk.img /media/hbahn/boot/boot
```

4. Testing Tizen

4.1. Wi-Fi test

- root:~> wifi_manager_test
 - 1 Wi-Fi init and set callbacks
 - 3 Activate Wi-Fi device
 - 9 Scan request
 - b Get AP list
 - c Connect

Copyright © 2019 Sigma Delta Technologies Inc. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means – electronic, mechanical, photocopying, recording or otherwise – without the permission of Sigma Delta Technologies.

Trademarks and trade names may be used in this document to refer to either the entities claiming the marks and names or their products. Sigma Delta Technologies disclaims any proprietary interest in the trademarks and trade names other than its own.

This document is subject to change without notice.

Sigma Delta Technologies Inc.

706-1, Gasan Digital 1-ro 19,
Geumcheon-gu, Seoul
Korea 08594

Website: <https://www.sigma-delta.tech>

Email: admin@sigma-delta.tech

Technical support: support@sigma-delta.tech

Tel: (+82)2-2629-4701

Please send any comments to 'support@sigma-delta.tech'