Hands-on Workshop: Developing on Quick Start Board for i.MX Processors

APF-CON-T0701

Rui Yang

May 2013
Topics

• Get i.MX materials from Freescale website
  - i.MX6 Series Processor / Reference Solutions Material

• Setup building environment and compiling
  - Android / Linux solution

• Download compiled images into target board

• Design Consideration
  - Hardware Design
Get i.MX materials from Freescale website
Get i.MX materials from Freescale website

- From the following web address:
  - [http://www.freescale.com/webapp/sps/site/homepage.jsp?nodeId=018rH3ZrDR](http://www.freescale.com/webapp/sps/site/homepage.jsp?nodeId=018rH3ZrDR)
- You can get all i.MX application processors’ family list:

You can get detailed information about each i.MX application processors after clicking into it.

This section use i.MX 6 Series Processors for example.

**i.MX Applications Processors**
- i.MX Product Selector
- i.MX Family Comparison Table
  - i.MX 6 Series Processors
  - i.MX53 Processors
  - i.MX51 Processors
  - i.MX50 Processors
  - i.MX37 Processors
  - i.MX35 Processors
  - i.MX31 Processors
  - i.MX28 Processors
  - i.MX27 Processors
  - i.MX25 Processors
  - i.MX23 Processors
  - i.MX21 Processors
  - i.MXS Processors
i.MX6 Series and Reference Solutions

• Check available reference solutions for i.MX6:

Choose suitable processor for your production in i.MX 6 Series:
i.MX6Q Reference Solutions

• Select proper reference solutions for your production
1. Select i.MX6Q in family tree and click into it;
2. Choose and press “Documentation” in option bar;
3. Get i.MX6Q Application Processor related material;
1. Select “SABRE Platform for Smart Devices”, and choose “Software & Tools”;
2. Hardware Material (Schematic, Gerber, OrCAD..etc);
3. Manufacturing tool;
4. Software Material (Android, Linux..etc)
Setup Building Environment and Compiling
Using Android Solution
Preparation

Download Ubuntu 10.04 (Lucid) 64-bit Desktop from:
http://releases.ubuntu.com/lucid/

The Sun JDK is no longer in Ubuntu's main package repository. Download latest Oracle/Sun JDK 6 binary release from:
Setup Building Environment

Install the Ubuntu 10.04 64 Bit Desktop.

Install the following packages for essential Android build, uImage
And uboot format support, building mtd-util, file comparison, storage partition:

```bash
$ sudo apt-get install git-core gnupg flex bison gperf build-essential \   zip curl zlib1g-dev libc6-dev lib32ncurses5-dev ia32-libs \   x11proto-core-dev libx11-dev lib32readline5-dev lib32z-dev \   libgl1-mesa-dev g++-multilib mingw32 tofrodos python-markdown \   libxml2-utils xsltproc
$ sudo apt-get install uboot-mkimage
$ sudo apt-get install uuid-dev liblzo2-dev
$ sudo apt-get install meld gparted
```

Install and setup jdk-6u43-linux-x64.bin, then verify Java version.

```bash
$ sudo chmod +x jdk-6u43-linux-x64.bin
$./jdk-6u43-linux-x64.bin
$ echo 'export PATH=Your Path/jdk1.6.0_43/bin:$PATH' >> ~/.bashrc
$ java -version
java version "1.6.0_43"
Java(TM) SE Runtime Environment (build 1.6.0_43-b01)
Java HotSpot(TM) 64-Bit Server VM (build 20.14-b01, mixed mode)
```

**NOTE:** You can get detailed install guide in “buildenv.pdf” from our Disty.
Get Source Code (Android/Kernel)

Get Android source code from Goodle repo, then retrieve ALSA source code:

```bash
$ cd ~
$ mkdir myandroid
$ cd myandroid
$ curl https://dl-ssl.google.com/dl/googlesource/git-repo/repo > ./repo
$ chmod a+x ./repo
$ ./repo init -u https://android.googlesource.com/platform/manifest -b android-4.0.4_r1.1
$ cp /opt/imx-android-13.4.1/code/13.4.1/default.xml ./repo/manifests/default.xml
$ ./repo sync
$ cd myandroid/external
$ git clone git://android.git.linaro.org/platform/external/alsa-lib.git
$ cd myandroid/external
$ git clone git://android.git.linaro.org/platform/external/alsa-utils.git
$ cd myandroid/hardware
$ git clone git://android.git.linaro.org/platform/hardware/alsa_sound.git
```

Get Kernel source code from Freescale’s git:

```bash
$ cd myandroid
$ git clone git://git.freescale.com/imx/linux-2.6-imx.git kernel_imx
$ cd kernel_imx
$ git checkout imx-android-13.4.1
```

**NOTE: Assume you unpack “imx-android-13.4.1.tar.gz” to “/opt/imx-android-13.4.1/”**
Get Source Code (U-Boot/i.MX Patch)

Get U-Boot source code from Freescale’s git:

```bash
$ cd myandroid/bootable
$ mkdir bootloader
$ cd bootloader
$ git clone git://git.freescale.com/imx/uboot-imx.git uboot-imx
$ cd uboot-imx
$ git checkout imx-android-13.4.1
```

Add patch code for i.MX:

```bash
$ cd myandroid/bootable
$ cd ~/myandroid
$ source /opt/imx-android-13.4.1/code/13.4.1/and_patch.sh
$ c_patch /opt/imx-android-13.4.1/code/13.4.1 imx_13.4.1
```

If everything is OK, "c_patch" will generate the following output to indicate successful patch:

```
******************************************************************
Success: Now you can build the Android code for FSL i.MX platform
******************************************************************
```

**NOTE:** You can get detailed steps from “/imx-android-13.4.1/doc/Android_User_Guide.pdf”
Build Images

Build U-Boot image (i.MX 6Quad SABRE SD for example):

$ cd ~/myandroid/bootable/bootloader/uboot-imx
$ export ARCH=arm
$ export CROSS_COMPILE=~//myandroid/prebuilt/linux-x86/toolchain/arm-eabi-4.4.3/bin/arm-eabi-
$ make distclean
$ make mx6q_sabresd_android_config
$ make

Build Kernel image:

$ cd ~/myandroid/kernel_imx
$ make distclean
$ make imx6_android_defconfig
$ make uImage

Build boot.img (uImage + uRamdisk):

$ cd ~/myandroid
$ source build/envsetup.sh
$ lunch sabresd_6dq-user
$ make bootimage

Build Android Image

$ cd ~/myandroid
$ source build/envsetup.sh
$ lunch sabresd_6dq-user
$ make
Useful Tips:

- We provide reference virtual build host running on VMware-player-5.0.0, all the building environment are ready.
- Uncompress ubunut_64-bit_1004_20121126_android.7z to PC, Install
- VMware-player-5.0.0 and open this image.

Username: **vmuser**
Password: **vmuser**
Download Android Images into Target Board
Using SABRE Platform for Smart Devices
Compiled Images

- After building, you can get the following images located in “myandroid/out/target/product/sabresd_6dq”:

  - **u-boot-6dl.bin/u-boot-6q.bin**: bootloader, start offset is 0, max size is 1MB;
  - **boot.img**: android image which stores kernel and ramdisk together, partition name is “Boot”, start offset is 8MB, max size is 8MB;
  - **recovery.img**: boot.img format, which stores kernel and ramdisk, partition name is “Recovery”, start offset is following “Boot”, max size is 8MB;
  - **system.img**: android EXT4 system files, partition name is “System”, start offset is following “Recovery”, max size is 512MB;
Download Preparation

• Setup download environment as following:
Download Images by Mfgtools

1. Copy such four images into “\Mfgtools-Dir\Profiles\MX6Q Linux Update\OS Firmware\files\android”;

2. Change the SABRE SD SW6 (boot) to 00001100 (from 1-8 bit) to enter USB OTG download mode;

3. Power on the board. Using USB cable on the SABRE SD OTG port, connect your Windows PC to SABRE SD;

4. MfgTool will detect SABRE board, the status will change as following:
Download Images by MfgTools

5. Click “Start” to start image downloading;

6. During the downloading process, the status bar of MfgTool will show the downloading status;

7. The downloading is complete when MfgTool show “green progress bar” as following:

8. Change Boot Switch (SW6) to 11100110 (from 1-8 bit), make SABRE boot from eMMC;

**NOTE:** You can get detailed steps from “/imx-android-13.4.1/doc/ Android_Quick_Start_Guide.pdf”
1. There are three hardware displays supported in SDP: two LVDS display panels and HDMI output, you can set different U-Boot environment parameters for display output as following:

**LVDS Display Single Display** on LVDS1 display:
U-Boot > setenv bootargs console=ttymxc0,115200 androidboot.console=ttymxc0 vmalloc=400M init=/init video=mxcfb0:dev=ldb,LDB-XGA,if=RGB666,bpp=16 video=mxcfb1:off video=mxcfb2:off fbmem=10M fb0base=0x27b00000

**HDMI Display Single Display**:
U-Boot > setenv bootargs console=ttymxc0,115200 androidboot.console=ttymxc0 vmalloc=400M init=/init video=mxcfb0:dev=hdmi,1920x1080M@60 video=mxcfb1:off video=mxcfb2:off fbmem=28M

**LVDS&HDMI Display Dual Display** enable LVDS1 and HDMI output dual display feature:
U-Boot > setenv bootargs console=ttymxc0,115200 init=/init rw video=mxcfb0:dev=ldb,LDBXGA, if=RGB666,bpp=16 fb0base=0x27b00000 video=mxcfb1:dev=hdmi,1920x1080M@60 fbmem=10M,28M video=mxcfb2:off vmalloc=512M androidboot.console=ttymxc0

**LVDS&HDMI&LVDS Display Triple Display** enable LVDS1, HDMI output, and LVDS0 triple display:
U-Boot > setenv bootargs console=ttymxc0,115200 init=/init rw video=mxcfb0:dev=ldb,LDBXGA, if=RGB666,bpp=16 fb0base=0x27b00000 video=mxcfb1:dev=hdmi,1920x1080M@60 fbmem=10M,28M vmalloc=512M androidboot.console=ttymxc0

2. You can follow below steps to copy files into “/system” partition:

$ mount -t ext4 -o rw,remount /dev/block/mmcblk0p5 /system
$ busybox cp /mnt/sdcard/files /system/media/
$ mount -t ext4 -o ro,remount /dev/block/mmcblk0p5 /system
Setup Building Environment and Compiling
Using Linux Solution
**LTIB Introduction**

- Linux Target Image Builder (LTIB) is a tool created by Freescale that is used to build Linux target images, composed of a set of packages:
  - A mechanism to deliver Linux board support packages (BSPs)
  - A wrapper around tool chains and standard Linux commands (cp, make, objcopy, tar, gcc, …)

- LTIB Packages for i.MX6Q SABRE Board:
  - Toolchain for the ARM® Cortex™- A9 CPU
  - Linux Kernel 3.0.35
  - Uboot 2009.08
  - Base tools: BusyBox, Dropbear, …
  - and many more …
Setup LTIB Environment

Install the Ubuntu 10.04 64 Bit Desktop.

Install the following packages:

```bash
$ sudo apt-get install gettext libgtk2.0-dev rpm bison m4 libfreetype6-dev
$ sudo apt-get install libdbus-glib-1-dev liborbit2-dev intltool
$ sudo apt-get install ccache ncurses-dev zlib1g zlib1g-dev gcc g++ libtool
$ sudo apt-get install uuid-dev liblzo2-dev
$ sudo apt-get install tcl dpkg
$ sudo apt-get install ia32-libs libc6-dev-i386 lib32z1
```

This package is used for uImage and uboot format support:

```bash
$ sudo apt-get install uboot-mkimage
```

These two packages are recommended to help the daily work, “meld” for file comparison and “gparted” for storage partition:

```bash
$ sudo apt-get install meld gparted
```

**NOTE:** You can get detailed steps from “Setting_up_LTIB_Host_L3.0.35_1.1.0.pdf”
Install and Configure LTIB

Install LTIB package, not as root, in a location such as /home/user/:

```
$ tar zxvf L3.0.35_1.1.0_121218_source.tar.gz
$ ./L3.0.35_1.1.0_121218_source/install
```

Configure and build LTIB:

```
$ cd <LTIB directory>
$ ./ltib -m config
```

1). The LTIB menu will appear as following;
2). If it has not been selected already, select:
   --- Choose the platform type
      Selection (imx6q) --->
3). Exit
4). Save

5). Select
   --- Choose your board for u-boot board (mx6q_sabresd) --->
6). Using the spacebar, select:
   [*] Configure the Kernel
7). Don’t exit yet...
Install and Configure LTIB

8). Select Package List;
   A Linux system is comprised of two main entities:
   -- Kernel
   -- File System
9). The Package List is what produces the File System.

10). Kernel Configuration;
11). Text messages will scroll by until the Kernel Configuration menu appears;
12). Exit from the kernel config;
13). The kernel now builds.

After LTIB building, you can get the U-boot and Kernel images:
Download Linux Images into Target Board
Using SABRE Platform for Smart Devices
Download Images by Mfgtools

• You should prepare Rootfs for Mfgtools download as following:

   $ cd <your rootfs directory>
   $ sudo -s
   $ tar -cjf ../rootfs.tar.bz2 ./*

• Copy “u-boot.bin”, “uImage”, “rootfs.tar.bz2” into i.MX6Q Linux MFGtool profile folder.
  - (~\Mfgtools-Rel-1.1.0_121218_MX6Q_UPDATER\Profiles\MX6Q Linux Update\OS Firmware\files)

• Change the SABRE SD SW6 (boot) to 00001100 (from 1-8 bit) to enter USB OTG download mode, then download Linux images by Mfgtoos at the same way as Android;

• Change Boot Switch (SW6) to 01000010 (from 1-8 bit), make SABRE boot from SD card slot3;
Download Images in Linux Host – (1)

Insert one SD card into your Linux host PC, and it will recognize your SD. In this example, the device node assigned is “/dev/sdb”:

$ cat /proc/partitions

<table>
<thead>
<tr>
<th>major</th>
<th>minor</th>
<th>#blocks</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>0</td>
<td>78125000</td>
<td>sda</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>75095811</td>
<td>sda1</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>1</td>
<td>sda2</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>3028221</td>
<td>sda5</td>
</tr>
<tr>
<td>8</td>
<td>32</td>
<td>488386584</td>
<td>sdc</td>
</tr>
<tr>
<td>8</td>
<td>33</td>
<td>488386552</td>
<td>sdc1</td>
</tr>
<tr>
<td>8</td>
<td>16</td>
<td>3921920</td>
<td>sdb</td>
</tr>
<tr>
<td>8</td>
<td>18</td>
<td>3905535</td>
<td>sdb1</td>
</tr>
</tbody>
</table>

Copying Boot Loader Image into SD card:

$ sudo dd if=u-boot-mx6q-sabresd.bin of=/dev/sdb bs=512 seek=2 skip=2 conv=fsync

Copying Kernel Image into SD card:

$ sudo dd if=uImage of=/dev/sdb bs=512 seek=2048 conv=fsync
Create a partition for Root File System:

```bash
$ sudo umount /dev/sdb
$ sudo fdisk /dev/sdb
  u [switch the unit to sectors instead of cylinders]
  d [repeat this until no partition is reported by the 'p' command ]
  n [create a new partition]
  p [create a primary partition]
  1 [the first partition]
  16384 [starting at offset sector #16384, i.e. 8MB, which leaves enough space for the kernel, the boot loader and its configuration data]
  <enter> [using the default value will create a partition that spans to the last sector of the medium]
  w [this writes the partition table to the medium and fdisk exits]
$ sudo mkfs.ext4 /dev/sdb1
```

Copy target file system into the partition in SD card:

```bash
$ mkdir /home/user/mountpoint
$ sudo mount /dev/sdb1 /home/user/mountpoint
$ gunzip rootfs.ext2.gz
$ mount -o loop -t ext2 rootfs.ext2 /home/user/rootfs
$ cd /home/user/rootfs
$ sudo cp -rpa [A-z]* /home/user/mountpoint
$ sudo umount /home/user/mountpoint
```

Insert this SD card into SABRE board slot3 and boot from it.
Modify U-Boot Environment

Create U-Boot environment commands to send display out through LVDS connected to DISP0 (default) from SD card:

```
U-Boot > setenv loadaddr 0x10800000
U-Boot > setenv bootargs_base 'console=tty1mxc0,115200'
U-Boot > setenv bootargs_base_lvds 'video=mxcfb0:dev=ldb,LDB-XGA,if=RGB666'
U-Boot > setenv bootargs_mmc 'root=/dev/mmcblk1p1 rootwait rw ip=none rootfstype=ext4'
U-Boot > setenv bootargs ${bootargs_base} ${bootargs_base_lvds} ${bootargs_mmc}
U-Boot > setenv bootcmd_mmc 'mmc dev 2;mmc read ${loadaddr} 0x800 0x2000;bootm'
U-Boot > setenv bootcmd 'run bootcmd_mmc'
U-Boot > saveenv
```

For SDP, the LVDS is connected to DISP1, thus, modify the command as follows:

```
U-Boot > setenv bootargs_base_lvds 'video=mxcfb0:dev=ldb,LDB-XGA,if=RGB666
ldb=sin1'
```

Need to add “rootfstype=xxx” to eliminate the time to determine the file system type of the root file system.

**NOTE:** refer to “SABRE_SD_Release_Notes_L3.0.35_1.1.0.pdf” for the details about the “Kernel Boot Parameters”.
Ubuntu Booting on SDP

- After download, the Ubuntu boot from SD card slot3 on SABRE Platform for Smart Devices as following:
Hardware Design Consideration
### Power Management

- You should consider the following parts in power design:
  1. Voltage range of each power domain;
  2. Maximal current consuming of each power domain;
  3. System power up sequence;

- The following are typical Max Power Measurement Results on SD Board:

<table>
<thead>
<tr>
<th>Supply Domain</th>
<th>Voltage (V)</th>
<th>Linux - ER1205 - on SD Board¹</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>P (mW)</td>
</tr>
<tr>
<td>VDDARM_IN</td>
<td>1.37</td>
<td>2068.7</td>
</tr>
<tr>
<td>VDDSOC_IN</td>
<td>1.37</td>
<td>1555</td>
</tr>
<tr>
<td>VDDHIGH_IN</td>
<td>2.78</td>
<td>236.3</td>
</tr>
<tr>
<td>Total Power (without DDR3 I/O + Memories)</td>
<td></td>
<td>3860</td>
</tr>
<tr>
<td>DDR3 I/O +³ Memories</td>
<td>1.5</td>
<td>1995</td>
</tr>
<tr>
<td>Total Power</td>
<td></td>
<td>5855</td>
</tr>
</tbody>
</table>

Footnote:

¹ These measurements were taken on a specific SD Board and may vary depending on the board and environmental conditions.
## i.MX6Q Power Domain

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Usage</th>
<th>Voltage</th>
<th>Max Current</th>
<th>Generated</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDDARM_IN</td>
<td>ARM Core Power</td>
<td>1.05~1.5V, 1.375V</td>
<td>2500 mA</td>
<td>PF0100 SW1A/B</td>
<td>1</td>
</tr>
<tr>
<td>VDDARM23_IN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VDDSOC_IN</td>
<td>IP Power (VPU, GPU...)</td>
<td>1.275~1.5V, 1.375V</td>
<td>1750 mA</td>
<td>PF0100 SW1C</td>
<td>1</td>
</tr>
<tr>
<td>VDDHIGH_IN</td>
<td>Internal Regulator</td>
<td>2.8~3.3V, 2.8V</td>
<td>100 mA</td>
<td>PF0100 VGEN5</td>
<td>---</td>
</tr>
<tr>
<td>VDD_SNVS_IN</td>
<td>Backup Battery</td>
<td>2.8~3.3V, 3.0V</td>
<td>400 μA</td>
<td>PF0100 VSNVS</td>
<td>0</td>
</tr>
<tr>
<td>USB_OTG_VBUS</td>
<td>USB Supply Voltages</td>
<td>4.4~5.25V, 5.0V</td>
<td>600 mA</td>
<td>PF0100 SWBST</td>
<td>---</td>
</tr>
<tr>
<td>USB_H1_VBUS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NVCC_DRAM</td>
<td>DDR I/O supply</td>
<td>LPDDR2: 1.14<del>1.3V, 1.2V DDR3: 1.425</del>1.575V, 1.5V DDR3_L: 1.283~1.45V, 1.35V</td>
<td>2500 mA</td>
<td>PF0100 SW3A/B</td>
<td>---</td>
</tr>
<tr>
<td>NVCC_RGMII</td>
<td>RGMII I/O Power</td>
<td>1.15~2.625V, 1.5V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NVCC_EIM0,1,2</td>
<td>GPIO Power Supply</td>
<td>1.65~3.6V, 1.8/ 2.8/ 3.3 V</td>
<td>2000 mA</td>
<td>PF0100 SW2</td>
<td>---</td>
</tr>
<tr>
<td>NVCC_ENET</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NVCC_GPIO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NVCC_LCD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NVCC_NANDF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NVCC_SD2/3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NVCC_JTAG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**i.MX6Q Internal Regulators**

- i.MX6 series contain 7 internal regulators, it simplify the power supply scheme of the system;
- The following domains are supplied by internal regulators:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Usage</th>
<th>Voltage</th>
<th>Generated</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVCC_LVDS2P5</td>
<td>LVDS</td>
<td>2.25~2.75V, 2.5V</td>
<td>i.MX VDDHIGH_CAP</td>
</tr>
<tr>
<td>NVCC_MIPI</td>
<td>MIPI</td>
<td>2.25~2.75V, 2.5V</td>
<td>i.MX VDDHIGH_CAP</td>
</tr>
<tr>
<td>HDMI_VP</td>
<td>HDMI Supply Voltages</td>
<td>0.99~1.3V, 1.1V</td>
<td>i.MX VDDSOC_CAP</td>
</tr>
<tr>
<td>HDMI_VPH</td>
<td></td>
<td>2.25~2.75V, 2.5V</td>
<td>i.MX VDDHIGH_CAP</td>
</tr>
<tr>
<td>PCIE_VP</td>
<td>PCIe Supply Voltages</td>
<td>1.023~1.3V, 1.1V</td>
<td>i.MX VDDSOC_CAP</td>
</tr>
<tr>
<td>PCIE_VPH</td>
<td></td>
<td>2.325~2.75V, 2.5V</td>
<td>i.MX VDDHIGH_CAP</td>
</tr>
<tr>
<td>PCIE_VPTX</td>
<td></td>
<td>1.023V~1.3V, 1.1V</td>
<td>i.MX VDDSOC_CAP</td>
</tr>
<tr>
<td>SATA_VP</td>
<td>SATA Supply Voltages</td>
<td>0.99~1.3V, 1.1V</td>
<td>i.MX VDDSOC_CAP</td>
</tr>
<tr>
<td>SATA_VPH</td>
<td></td>
<td>2.25~2.75V, 2.5V</td>
<td>i.MX VDDHIGH_CAP</td>
</tr>
</tbody>
</table>
i.MX6 SDP Power Connectivity
Typical Power Requirements

System Power Rails

<table>
<thead>
<tr>
<th>Voltage</th>
<th>Rail Name</th>
<th>Block</th>
<th>Generated By</th>
<th>Current Capacity (mA)</th>
<th>NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.0</td>
<td>PMIC_5V</td>
<td>USB</td>
<td>PF0100 SWBST</td>
<td>600</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LVPD1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>HDMI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AUX_5V</td>
<td>SATA</td>
<td>MAX8815</td>
<td>1000</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LVPD0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.3</td>
<td>GEN_3V3</td>
<td>EMMC</td>
<td>PF0100 SW2</td>
<td>2000</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SD3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>NOR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SATA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LVPD5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>HDMI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MIPI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>mPCIe</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SENSORS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VGEN6_3V3</td>
<td>ETH</td>
<td>PF0100 VGEN6</td>
<td></td>
<td>2000</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.15</td>
<td>AUX_3V15</td>
<td>EXP HDR</td>
<td>PF0100 SW4</td>
<td>1000</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TOUCH</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>GPS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.8</td>
<td>VDDHIGH_1V</td>
<td>IMX6</td>
<td>PF0100 VGEN5</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>VGEN3_2V3</td>
<td>CAMERA</td>
<td>PF0100 VGEN3</td>
<td></td>
<td>100</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.5</td>
<td>GEN_2V5</td>
<td>SATA</td>
<td>IMX6</td>
<td>TBD</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>HDMI</td>
<td>VDDHH(lCHAP)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MIPI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>mPCIe</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.8</td>
<td>GEN_1V8</td>
<td>AUDIO</td>
<td>PF0100 VGEN4</td>
<td>350</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CAMERA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ACC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VGEN2_1V5</td>
<td>CAMERA</td>
<td>PF0100 VG2N2</td>
<td></td>
<td>250</td>
<td></td>
</tr>
<tr>
<td>VGEN1_1V5</td>
<td>GPS</td>
<td>PF0100 VG2N1</td>
<td></td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>DDR_1V5</td>
<td>DDR</td>
<td>PF0100 SW34/B</td>
<td></td>
<td>3500</td>
<td></td>
</tr>
<tr>
<td>1.375</td>
<td>VDDCORE</td>
<td>ARMCORE</td>
<td>PF0100 SW14/B</td>
<td>7500</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>VDDSOC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>VDD50C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.75</td>
<td>VREFDDR</td>
<td>DDR</td>
<td>PF0100 VREFDDR</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

Total System Current Requirements: 4454
### Power Connections for i.MX6 Series:

<table>
<thead>
<tr>
<th></th>
<th>i.MX 6Quad</th>
<th>i.MX 6Dual</th>
<th>i.MX 6DualLite</th>
<th>i.MX 6Solo</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SH1</strong></td>
<td>Shorted</td>
<td>Open</td>
<td>Shorted</td>
<td>Shorted</td>
</tr>
<tr>
<td><strong>SH2</strong></td>
<td>Shorted</td>
<td>Open</td>
<td>Open</td>
<td>Shorted</td>
</tr>
<tr>
<td><strong>R1</strong></td>
<td>Open</td>
<td>Shorted</td>
<td>Open</td>
<td>Open</td>
</tr>
<tr>
<td><strong>R2</strong></td>
<td>Open</td>
<td>Open</td>
<td>Shorted</td>
<td>Open</td>
</tr>
</tbody>
</table>
Capacitor Placement

• One 22 μF bulk capacitor should be connected to each of these on-chip LDO regulator outputs:
  - VDDARM_CAP
  - VDDARM23_CAP
  - VDDSOC_CAP
  - VDDPU_CAP

• A 22 μF bulk capacitor must be placed as near as possible with pins/vias. **The distance should be less than 50mil between bulk cap and VDDxx_CAP pins.** Decoupling capacitors such as 0.1 μF or 0.22 μF should also be used.

• It is highly recommended that the user places the decoupling and bulk capacitors of the power domains on the bottom layer of the hardware design, directly underneath the associated package contacts.
Related Materials:

- **AN4397.pdf:**
  - Common Hardware Design for i.MX 6Dual/6Quad and i.MX 6Solo/6DualLite

- **AN4509.pdf:**
  - i.MX 6Dual/6Quad Power Consumption Measurement

- **IMX6DQ6SDLHDG.pdf:**
  - Hardware Development Guide for i.MX 6Quad, 6Dual, 6DualLite, 6Solo
  - Families of Applications Processors

- **IMX6DQCEC.pdf:**
  - i.MX 6Dual/6Quad Applications Processors for Consumer Products Data Sheet