

Mango-IMX6Q Android 9.0 포팅가이드

<http://www.mangoboard.com/>

<http://cafe.naver.com/embeddedcrazyboys>

Crazy Embedded Laboratory

Document History

Revision	Date	Change note
Init	2020-09-03	전종인

1.	개발 환경	5
2.	문서 및 소스 다운로드	5
2.1.	안드로이드 Pie 9.0, 커널,u-boot소스 다운로드 하기	6
2.2.	U-boot, kernel, 안드로이드 컴파일 하기	7
2.2.1.	Android 컴파일	7
2.2.2.	U-boot 컴파일	7
2.2.3.	커널 컴파일	8
3.	U-boot 디바이스 드라이버 포팅	8
3.1.	U-boot 수정	8
3.1.1.	RAM 살리기	8
3.1.2.	PMIC 제거 에러 잡기	9
3.1.3.	SD4 detect 하기	11
3.1.4.	이더넷 포팅 하기	14
3.1.5.	HDMI 확인 하기	15
3.1.6.	Display 포팅하기	16
3.1.7.	U-boot에서 로고 변경하기	20
4.	커널 디바이스 드라이버 포팅	22
4.1.	커널에서 불필요한 디바이스 삭제	22
4.1.1.	Mango-IMX6Q에 없는 디바이스 config 없애기	22
4.1.2.	PMIC 없으므로 LDO로 변경하기	25
4.2.	Display , 터치 드라이브 포팅	25
4.2.1.	LCD Power EN, Backlight GPIO 포팅	25
4.2.2.	7인치 감압식 LCD 드라이버 포팅	30
4.2.3.	10.1인치 정전식 LCD 드라이버 포팅	33
4.2.4.	정전식 드라이버 포팅하기	33
4.2.5.	7인치 정전식 LCD 드라이버 포팅	42
4.2.6.	PMIC 없으므로 LDO로 변경하기	43
4.3.	USB Host 확인	43
4.4.	eMMC 모듈 테스트	44
4.5.	SDHC4 포팅	47
4.6.	Watchdog 포팅	49
4.7.	LED 포팅	50
4.8.	USB Device 포팅 및 테스트	51
4.9.	키버튼 드라이버 포팅하기	52
4.10.	RTC 포팅	53
4.11.	이더넷 포팅	54
4.12.	PCIe 드라이버 포팅 및 테스트	58
4.13.	오디오 코덱 WM8960 포팅	60

4.14.	SATA 포팅	62
5.	Android 포팅	62
5.1.	불필요한 디바이스 제거	62
5.2.	Bootargs 설정하기	63
5.3.	정전식 터치 동작 시키기	63
5.4.	에러 디버깅	63
5.4.1.	파일 시스템 Mount 못하는 문제	63
6.	Android 이미지 Write하기	66
6.1.	이미지 틀 다운로드 해 보자.	66
6.1.1.	스크립트로 이용하여 이미지 Write하기	66

1. 개발 환경

리눅스 PC 우분투 16.04 64bit 운영체제에서 개발을 한다.

빌드에 필요한 Package는 <https://source.android.com/setup/build/initializing> 참조해서 설치를 해야 한다.

Mango-IMX6Q 보드는 i.MX6 Quad를 사용하고 있다.

회로도, 하드웨어 매뉴얼이 필요하다.

회로도는 help@crz-tech.com으로 요청하면 된다.

하드웨어 매뉴얼은 <http://crztech.iptime.org:8080/Release/mango-imx6q/Doc/Hardware/>에서 다운로드 받으면 된다.

다른 자료는 <http://crztech.iptime.org:8080/Release/mango-imx6q/>에서 다운로드 받으면 된다.

2. 문서 및 소스 다운로드

https://www.nxp.com/design/software/embedded-software/i-mx-software/android-os-for-i-mx-applications-processors:IMXANDROID?&tab=Documentation_Tab

링크에서 android_p9.0.0_2.2.0-ga_docs.zip 파일을 다운로드 받는다.

다운로드 받으려면, nxp 홈페이지에 회원가입해야 한다.

압축을 풀면 Android_Release_Notes.pdf를 보면 i.MX6Quad Platform을 지원한다고 되어 있다.

관련 된 소스와 문서는 아래와 같이 정의가 되어 있다.

Android source code package	<ul style="list-style-type: none"> • <code>imx-p9.0.0_2.2.0-ga.tar.gz</code>: i.MX Android proprietary source code package to enable the Android platform on i.MX-based boards. For example, Hardware Abstraction Layer implementation and hardware codec acceleration.
Documents	<p>The following documents are included in <code>android_p9.0.0_2.2.0-ga_docs.zip</code>:</p> <ul style="list-style-type: none"> • <i>Android™ Quick Start Guide (AQSUG)</i>: A document that explains how to run the Android platform on an i.MX board using prebuilt images. • <i>Android™ User's Guide (AUG)</i>: A document describing procedures for configuring and building this release package. • <i>Android™ Release Notes (ARN)</i>: A document that introduces key updates and known issues in this release. • <i>i.MX Android™ Extended Codec Release Notes (IMXACRN)</i>: A document that provides the extended codec information. • <i>i.MX Android™ Camera Issues on the SDP Platform (ACOI)</i>: A document that describes the camera issues on the SDP platform. • <i>i.MX Graphics User's Guide (IMXGRAPHICUG)</i>: A document that describes GPU 2D API, Tools, Memory, and Application programming guidelines.
Prebuilt images	<p>You can test the Android platform with a prebuilt image on i.MX board before building any code:</p> <ul style="list-style-type: none"> • <code>android_p9.0.0_2.2.0-ga_image_6qsabresd.tar.gz</code>: Prebuilt images with NXP extended features for the SABRE-SD board. The extended features include more multimedia format support. • <code>android_p9.0.0_2.2.0-ga_image_6qsabreauto.tar.gz</code>: Prebuilt images with NXP extended features for the SABRE-AI board. The extended features include more multimedia format support. • <code>android_p9.0.0_2.2.0-ga_image_6sxsabresd.tar.gz</code>: Prebuilt images with NXP extended features for the i.MX 6SoloX SABRE-SD board.

이제 포팅 할 준비가 되었다.

2.1. 안드로이드 Pie 9.0, 커널,u-boot소스 다운로드 하기

https://www.nxp.com/design/software/embedded-software/i-mx-software/android-os-for-i-mx-applications-processors:IMXANDROID?&tab=Design_Tools_Tab

링크에서

`imx-p9.0.0_2.2.0-ga.tar.gz`를 다운로드 받는다.

해당 소스를 리눅스 PC 작업할 디렉토리에 다운로드 받는다.

압축을 푼다.

이제부터 `android_p9.0.0_2.2.0-ga_docs` 디렉토리에 `Android_User's_Guide.pdf` 파일을 보면서 컴파일을 하면 된다.

```
$ tar xf imx-p9.0.0_2.2.0-ga.tar.gz
```

아래와 같이 실행을 한다.

```
$ mkdir ~/bin
```

```
$ curl https://storage.googleapis.com/git-repo-downloads/repo > ~/bin/repo
```

```
$ chmod a+x ~/bin/repo
```

```
$ export PATH=${PATH}:~/bin
$ source ~/imx-p9.0.0_2.2.0-ga/imx_android_setup.sh
```

소스를 다운로드를 한다. 약 3시간 걸린다.

다운로드가 완료가 되면, android_build 디렉토리가 생성이 되면서 다운로드 된 파일을 볼 수 있다.

```
$ cd android_build/
$ ls
Android.bp  SCR-p9.0.0_2.2.0-ga.txt  bootable  compatibility  developers  external
kernel      packages  prebuilts  test  vendor
EULA.txt    art      bootstrap.bash  cts  development  frameworks
libcore     pdk      sdk  toolchain
Makefile    bionic  build  dalvik  device  hardware
libnativehelper  platform_testing  system  tools
```

커널 소스는 vendor/nxp-opensource/kernel_imx/ 디렉토리에 있다.

u-boot 소스는 vendor/nxp-opensource/u-boot-imx/ 디렉토리에 있다.

2.2. U-boot, kernel, 안드로이드 컴파일 하기

안드로이드, 커널, u-boot 컴파일 해 보자.

2.2.1. Android 컴파일

Mango-IMX6Q 보드와 유사한 sabresd 보드 이름으로 컴파일 한다.

```
$ cd android_build
$ source build/envsetup.sh
$ lunch sabresd_6dq-eng
$ make -j 9 2>&1 | tee build-log.txt
```

“lunch sabresd_6dq-eng” 하면 보드는 sabresd_6dq이고, eng는 개발자 옵션을 넣어서 컴파일 하는 것이다. 좀 더 자세한 내용은 user guide를 살펴보기 바란다.

컴파일 시간이 약 1시간 이상 걸린다. 리눅스 PC 성능은 RAM 16GB 이상, 하드디스크는 300GB 이상 있는 것이 좋다.

컴파일이 완료되면 아래와 같이 메시지가 나옵니다.

```
#### build completed successfully (01:37:58 (hh:mm:ss)) ####
```

2.2.2. U-boot 컴파일

```
$ cd android_build
$ source build/envsetup.sh
```

```
$ lunch sabresd_6dq-eng
$ make bootloader -j4
```

컴파일 결과 아래와 같이 메시지가 나옵니다.

```
#### build completed successfully (02:27 (mm:ss)) ####
```

2.2.3. 커널 컴파일

```
$ export MY_ANDROID=/home/icanjji/work/imx6q/android-work/android9.0-work/android_build
경로는 각자 환경에 맞게 세팅한다.
$ cd ${MY_ANDROID}/vendor/nxp-opensource/kernel_imx
$ echo $ARCH && echo $CROSS_COMPILE
# Make sure you have those 2 environment variables set
# If the two variables have not set, please set the as:
$ export ARCH=arm
$ export CROSS_COMPILE=${MY_ANDROID}/prebuilts/gcc/linux-x86/arm/arm-linux-androideabi-4.9/bin/arm-linux-androideabi-
# Generate ".config" according to default config file under arch/arm/configs.
# to build the kernel Image for i.MX 6Quad, 6QuadPlus, 6DualLite, 6Solo, 6SoloLite,
6SoloX ,7Dual and 7ULP
$ make imx_v7_android_defconfig
$ make KCFLAGS=-mno-android
```

컴파일 하면 된다.

3. U-boot 디바이스 드라이버 포팅

3.1. U-boot 수정

u-boot를 Mango-IMX6Q에 맞게 수정해야 한다.

순서는 RAM-> Serial(Debug) -> Display(HDMI,10.1인치) -> 이더넷

3.1.1. RAM 살리기

Mango-IMX6Q는 512MB*4EA DDR3 장착되어 있다.

RAM 설정 파일은 board/freescale/mx6sabresd/mx6q_4x_mt41j128.cfg에 정의가 되어 있다.

Sabresd 보드는 1GB RAM이므로 설정 값을 변경한다.

같은 Sabreauto 보드에 설정값이 2GB RAM이다. 복사를 해 온다. 만약 같은 것이 없으면, 메모리 데이터 시트와 i.MX6Q RAM 설정 부분을 보면서 모두 맞추어야 한다.

board\freescaler\mx6sabreauto\wimximage.cfg 파일에 RAM 설정 부분을 가지고 와서 똑같이 맞춘다.
컴파일 해서 RAM이 제대로 살아났는지 확인 해 보자.

테스트 결과 RAM은 2GB로 인식을 했다.

```
U-Boot 2018.03-dirty (Sep 04 2020 - 20:10:34 +0900)

CPU:   Freescale i.MX6Q rev1.3 996 MHz (running at 792 MHz)
CPU:   Extended Commercial temperature grade (-20C to 105C) at 36C
Reset cause: POR
Model: Freescale i.MX6 Quad SABRE Smart Device Board
Board: MX6-SabreSD
DRAM:  2 GiB
```

3.1.2. PMIC 제거 에러 잡기

아래와 같이 에러가 발생을 한다.

```
U-Boot 2018.03-dirty (Sep 04 2020 - 20:10:34 +0900)

CPU:   Freescale i.MX6Q rev1.3 996 MHz (running at 792 MHz)
CPU:   Extended Commercial temperature grade (-20C to 105C) at 25C
Reset cause: POR
Model: Freescale i.MX6 Quad SABRE Smart Device Board
Board: MX6-SabreSD
DRAM:  2 GiB
read error from device: 89f18f10 register: 0x0!read error from device: 89f18f10 register: 0x3!PMIC:
PFUZE100! DEV_ID=0xffffffffb REV_ID=0xffffffffb
read error from device: 89f18f10 register: 0x21!write error to device: 89f18f10 register: 0x21!read error
from device: 89f18f10 register: 0x24!write error to device: 89f18f10 register: 0x24!read error from
device: 89f18f10 register: 0x2f!write error to device: 89f18f10 register: 0x2f!read error from device:
89f18f10 register: 0x32!write error to device: 89f18f10 register: 0x32!read error from device: 89f18f10
register: 0x0!Not supported, id=11
initcall sequence 8ff9eb4c failed at call 17804cbc (err=-22)
### ERROR ### Please RESET the board ###
```

Mango-IMX6Q에서는 PFUZE100(PMIC) 사용하지 않는다.

board/freescale/mx6sabresd/mx6sabresd.c 파일을 수정

power_init_board함수를 호출 하는데

안드로이드 전체 이미지를 Write하기에는 시간이 많이 걸리므로, u-boot만 컴파일 후

Linux 이미지를 사용한다.

<http://crztech.iptime.org:8080/Release/mango-imx6q/linux/kernel-4.1.15/20190321/mango-imx6q-kernel4.1.15-linux-10.1-image-20190321.tgz>

다운로드 받아서, sd에 Write한다. U-boot.imx 파일을 복사한 후 아래 명령으로 Write하면 된다.

```
$ sudo ./sdwriter-ubuntu16.04 sdb imx6q bin
```

u-boot 소스에 Device tree가 있다.

arch/arm/dts/imx6qdl-sabresd.dtsi 파일 수정해야 한다.

```
#if 0 //crazyboys 20200907
    pmic: pfuze100@08
...
#endif
```

PMIC 막으니까 아래와 같이 에러가 발생을 한다.

```
initcall sequence 8ff9eb4c failed at call 17804cbc (err=-19)
### ERROR ### Please RESET the board ###
```

찾아 보면 아래와 같다.

```
common/board_f.c 파일에
void board_init_f(ulong boot_flags)
=>
lib/initcall.c:                printf("initcall sequence %p failed at call %p (err=%d)\n",
```

common/board_f.c 파일에

static const init_fnc_t init_sequence_f[] 에 순서가 정해져 있다. Pfuze 부분을 삭제 했으니 에러가 발생한 것 같다.

찾았다. common/board_r.c 파일에서

```
//crazyboys 20200907 power_init_board,
```

막으면 된다. 아래와 같이 console까지 간다.

```
U-Boot 2018.03-dirty (Sep 07 2020 - 14:16:16 +0900)
```

```
CPU:   Freescale i.MX6Q rev1.3 996 MHz (running at 792 MHz)
```

```
CPU:   Extended Commercial temperature grade (-20C to 105C) at 33C
```

```
Reset cause: POR
```

```
Model: Freescale i.MX6 Quad SABRE Smart Device Board
```

```
Board: MX6-SabreSD
```

```
MANGO-IMX6Q-DRAM: 2 GiB
```

```
MMC:   FSL_SDHC: 0, FSL_SDHC: 1, FSL_SDHC: 2
```

Loading Environment from MMC... *** Warning - bad CRC, using default environment

Failed (-5)

No panel detected: default to Hannstar-XGA

Display: Hannstar-XGA (1024x768)

In: serial

Out: serial

Err: serial

flash target is MMC:2

Net: No ethernet found.

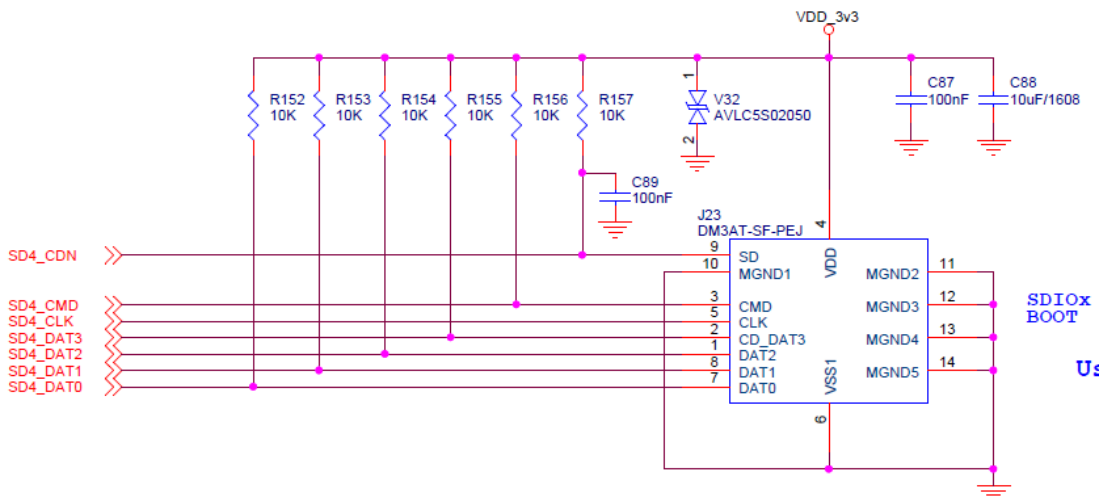
Loading Environment from MMC... *** Warning - bad CRC, using default environment

Failed (-5)

에러가 나오는 것은 SD가 제대로 detec가 되지 않아서 이다.

3.1.3. SD4 detect 하기

CX-IMX6Q 회로도를 보면 SD4를 SD 부팅으로 사용한다.



board/freescale/mx6sabresd/mx6sabresd.c 소스에서 수정을 한다.

```
static iomux_v3_cfg_t const usdhc4_pads[] = {
    IOMUX_PADS(PAD_SD4_CLK__SD4_CLK | MUX_PAD_CTRL(USDHC_PAD_CTRL)),
    IOMUX_PADS(PAD_SD4_CMD__SD4_CMD | MUX_PAD_CTRL(USDHC_PAD_CTRL)),
    IOMUX_PADS(PAD_SD4_DAT0__SD4_DATA0 | MUX_PAD_CTRL(USDHC_PAD_CTRL)),
    IOMUX_PADS(PAD_SD4_DAT1__SD4_DATA1 | MUX_PAD_CTRL(USDHC_PAD_CTRL)),
```

```

IOMUX_PADS(PAD_SD4_DAT2__SD4_DATA2 | MUX_PAD_CTRL(USDHC_PAD_CTRL)),
IOMUX_PADS(PAD_SD4_DAT3__SD4_DATA3 | MUX_PAD_CTRL(USDHC_PAD_CTRL)),
#if 0 //crazyboys 20200907 remove it
IOMUX_PADS(PAD_SD4_DAT4__SD4_DATA4 | MUX_PAD_CTRL(USDHC_PAD_CTRL)),
IOMUX_PADS(PAD_SD4_DAT5__SD4_DATA5 | MUX_PAD_CTRL(USDHC_PAD_CTRL)),
IOMUX_PADS(PAD_SD4_DAT6__SD4_DATA6 | MUX_PAD_CTRL(USDHC_PAD_CTRL)),
IOMUX_PADS(PAD_SD4_DAT7__SD4_DATA7 | MUX_PAD_CTRL(USDHC_PAD_CTRL)),
#endif
};

```

아래 cd 핀 수정, SD4는 cd pin을 사용하지 않음

```

int board_mmc_getcd(struct mmc *mmc)
{
    struct fsl_esdhc_cfg *cfg = (struct fsl_esdhc_cfg *)mmc->priv;
    int ret = 0;

    switch (cfg->esdhc_base) {
    case USDHC2_BASE_ADDR:
        ret = !gpio_get_value(USDHC2_CD_GPIO);
        break;
    case USDHC3_BASE_ADDR:
        //ret = !gpio_get_value(USDHC3_CD_GPIO);
        ret = 1; /* eMMC/uSDHC3 is always present */
        break;
    case USDHC4_BASE_ADDR:
        ret = 1; /* uSDHC4 is always present */
        break;
    }

    return ret;
}

```

arch/arm/dts/imx6qdl-sabresd.dtsi 수정

```

pinctrl_usdhc4: usdhc4grp {
    fsl,pins = <
        MX6QDL_PAD_SD4_CMD__SD4_CMD            0x17059
        MX6QDL_PAD_SD4_CLK__SD4_CLK            0x10059
        MX6QDL_PAD_SD4_DAT0__SD4_DATA0         0x17059
        MX6QDL_PAD_SD4_DAT1__SD4_DATA1         0x17059
    >
};

```

```

MX6QDL_PAD_SD4_DAT2__SD4_DATA2      0x17059
MX6QDL_PAD_SD4_DAT3__SD4_DATA3      0x17059
#endif
>;
};

&usdhc4 {
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_usdhc4>;
    bus-width = <4>;
    non-removable;
    no-1-8-v;
    keep-power-in-suspend;
    status = "okay";
};

```

include/configs/mx6sabresd.h 파일 수정

```
#define CONFIG_SYS_MMC_ENV_DEV      2      /* crazyboys 20200907 SDHC3-> SDHC4 */
```

```
#define CONFIG_MMCROOT              "/dev/mmcblk3p2" /* crazyboys 20200907 SDHC3 -
> SDHC4 */
```

이미지를 Write 후 environment를 설정을 한다.

```
setenv mmcdev 2
```

전체 설정을 하려면 <http://crztech.iptime.org:8080/Release/mango-imx6q/linux/kernel-4.1.15/20190321/HowTo.txt> 파일을 열어서 설정을 한다.

```

=> mmcinfo
Device: FSL_SDHC
Manufacturer ID: 3
OEM: 5344
Name: SC16G
Bus Speed: 50000000
Mode : SD High Speed (50MHz)
Rd Block Len: 512

```

SD version 3.0

High Capacity: Yes

Capacity: 14.8 GiB

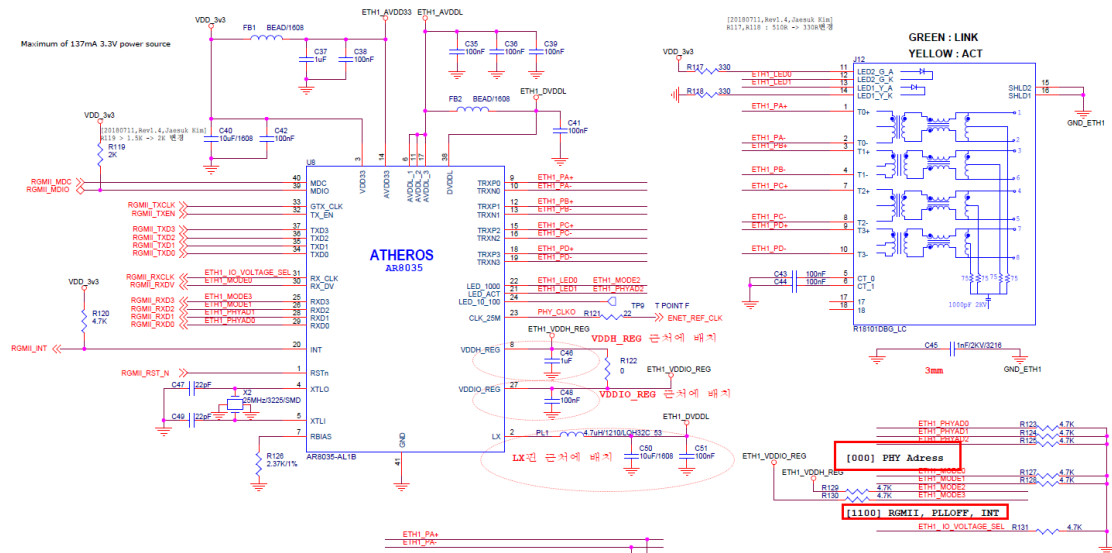
Bus Width: 4-bit

Erase Group Size: 512 Bytes

확인 할 수 있다.

3.1.4. 이더넷 포팅 하기

Mango-IMX6Q는 PHY칩을 Athros8035를 사용한다. PHY Address는 0을 사용한다.



include/configs/mx6sabre_common.h 파일에서 수정

```
#define IMX_FEC_BASE                ENET_BASE_ADDR
#define CONFIG_FEC_XCV_TYPE         RGMII
#ifdef CONFIG_DM_ETH
#define CONFIG_ETHPRIME             "eth0"
#else
#define CONFIG_ETHPRIME             "FEC"
#endif
#define CONFIG_FEC_MXC_PHYADDR     0 /* crazyboys 20200907 1->0 */
```

테스트 방법 MAC address를 지정을 해야 한다.

```
=> setenv ethaddr 04:32:F4:FE:D5:1C
=> saveenv
=> dhcp fec
```

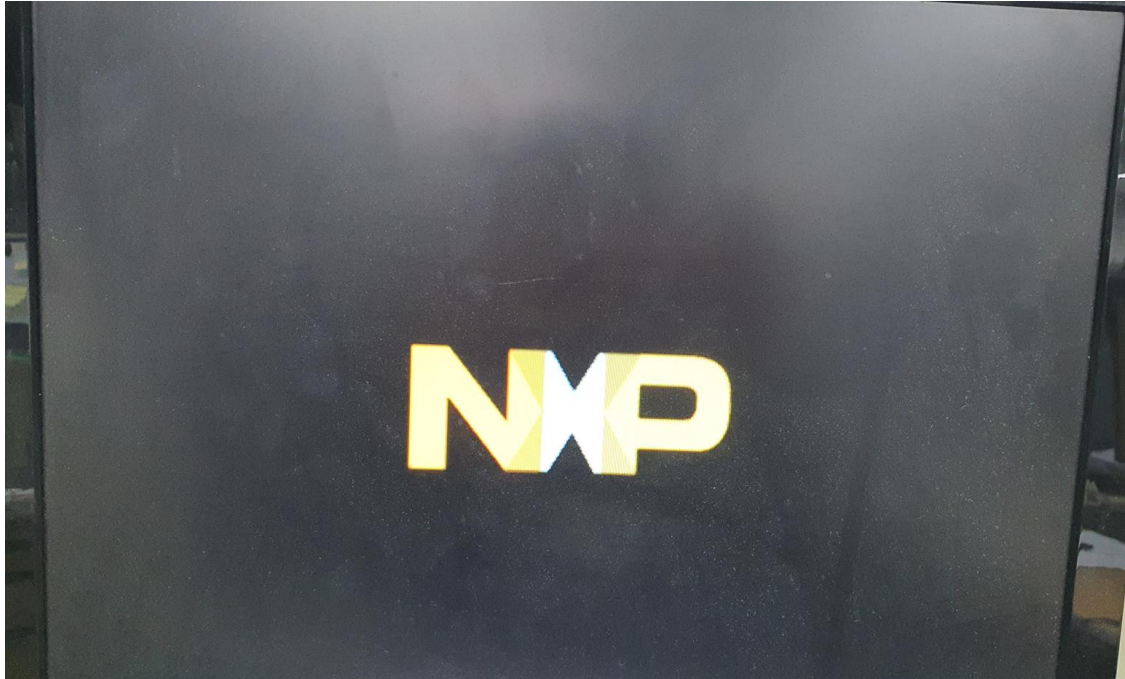
```
BOOTP broadcast 1
BOOTP broadcast 2
BOOTP broadcast 3
DHCP client bound to address 192.168.100.215 (1354 ms)
*** Warning: no boot file name; using 'C0A864D7.img'
Using FEC device
TFTP from server 0.0.0.0; our IP address is 192.168.100.215; sending through gateway 192.168.100.1
Filename 'C0A864D7.img'.
Load address: 0xfec
Loading: *
TFTP error: 'File not found' (1)
Not retrying...
```

이더넷이 ip를 할당 받으면 된다.

3.1.5. HDMI 확인 하기

```
=> setenv panel HDMI
=> saveenv
Saving Environment to MMC... Writing to MMC(2)... OK
=> reset
```

HDMI 포트에 LCD를 연결하면 아래와 같이 화면이 나온다.



3.1.6. Display 포팅하기

회로도를 보면 LCD Power Enable, Backlight GPIO가 있다.

구분	Pad Name	GPIO
LCD_BL_CTRL	SD1_DAT2	GPIO1_19
DISP0_PWR_EN	ENET_TXD0	GPIO1_30

IMX6DQRM-Reference Manual.pdf 데이터시트를 보면 아래와 같이 정의가 되어 있다.

LCD_BL_CTRL과 DISP0_PWR_EN을 모두 High해야 LCD가 밝혀져서 화면이 표시가 된다.
 정확히 말하면 LCD_BL_CTRL 핀은 PWM 신호로 제어하는 것이 맞으나, u-boot에서 화면
 밝기를 조절하지 않고, 커널에서 조절하는 것으로 하겠다.
 DISP0_PWR_EN핀은 LCD에 Power를 공급할 때 제어하는 핀이다.

board/freescale/mx6sabresd/mx6sabresd.c파일에서 아래와 같이 정의를 한다.

```
#define DISP0_PWR_EN    IMX_GPIO_NR(1, 30)//crazyboys 20200907
#define DISP0_BACKLIGHT_EN IMX_GPIO_NR(1, 19) //crazyboys 20200907
```



```

static void enable_backlight(void)
{
    SETUP_IOMUX_PADS(bl_pads);
    gpio_request(DISPO_PWR_EN, "Display Power Enable");
    gpio_direction_output(DISPO_PWR_EN, 1);
#if 1 //crazyboys 20200907
    gpio_request(DISPO_BACKLIGHT_EN, "Display Backlight Enable");
    gpio_direction_output(DISPO_BACKLIGHT_EN, 1);
#endif
}

```

LCD에 따라서 타이밍 값을 아래와 같이 수정한다.

static struct display_info_t const displays[] 배열에 아래와 같이 추가한다.

```

}}, {
    .bus    = 0,
    .addr   = 0,
    .pixfmt = IPU_PIX_FMT_BGR24,
    .detect = NULL,
    .enable = enable_rgb,
    .mode   = {
        .name          = "MANGO-CAP7", //crazyboys 20151005
        .refresh       = 60,
        .xres          = 1024, //800,
        .yres          = 600, //480,
        .pixclock      = 51200, //29850,
        .left_margin   = 60,
        .right_margin  = 60,
        .upper_margin  = 5, //23,
        .lower_margin  = 5, //10,
        .hsync_len     = 200, 10,
        .vsync_len     = 25, //10,
        .sync          = 0,
        .vmode         = FB_VMODE_NONINTERLACED
    }
}}, {
    .bus    = 0,
    .addr   = 0,

```

```

.pixfmt = IPU_PIX_FMT_BGR24,
.detect = NULL,
.enable = enable_rgb,
.mode = {
    .name          = "MANGO-PRESS10.4", //crazyboys 20151005
    .refresh       = 60,
    .xres          = 800, //800,
    .yres          = 600, //480,
    .pixclock      = 20000, //51200, //29850,
    .left_margin   = 60,
    .right_margin  = 60,
    .upper_margin  = 5, //23,
    .lower_margin  = 5, //10,
    .hsync_len     = 200, 10,
    .vsync_len     = 25, //10,
    .sync          = 0,
    .vmode         = FB_VMODE_NONINTERLACED
}}, {
.bus      = 0,
.addr     = 0,
.pixfmt = IPU_PIX_FMT_BGR24,
.detect = NULL,
.enable = enable_rgb,
.mode = {
    .name          = "MANGO-CAP10.1", //crazyboys 20151005
    .refresh       = 60,
    .xres          = 1280, //800,
    .yres          = 800, //480,
    .pixclock      = 13468, //51200, //29850,
    .left_margin   = 68,
    .right_margin  = 60,
    .upper_margin  = 8, //23,
    .lower_margin  = 8, //10,
    .hsync_len     = 2,
    .vsync_len     = 2, //10,
    .sync          = 0,
    .vmode         = FB_VMODE_NONINTERLACED
}

```

```

}}, {
    .bus      = 0,
    .addr     = 0,
    .pixfmt   = IPU_PIX_FMT_BGR24,
    .detect   = NULL,
    .enable   = enable_rgb,
    .mode     = {
        .name          = "MANGO-PRESS7", //crazyboys 20151005
        .refresh       = 60,
        .xres          = 800,
        .yres          = 480,
        .pixclock      = 37879,
        .left_margin   = 46,
        .right_margin  = 16,
        .upper_margin  = 23,
        .lower_margin  = 7, //10,
        .hsync_len     = 1,
        .vsync_len     = 1, //10,
        .sync          = 0,
        .vmode         = FB_VMODE_NONINTERLACED
    }
}

```

u-boot에서 검증한다.

```

=> setenv panel MANGO-PRESS7
=> saveenv
Saving Environment to MMC...
Writing to MMC(2)... done
=> reset
resetting ...

```

리부팅하고 u-boot에서 프롬프트에서 멈추면 로그가 나온다.

<10.1" LCD>

```

setenv panel MANGO-CAP10.1
saveenv
reset

```

<7" 1024x600 LCD>

```

setenv panel MANGO-CAP7
saveenv
reset

```

U-Boot 2018.03-dirty (Sep 07 2020 - 17:41:30 +0900)

```
CPU: Freescale i.MX6Q rev1.3 996 MHz (running at 792 MHz)
CPU: Extended Commercial temperature grade (-20C to 105C) at 45C
Reset cause: POR
Model: Freescale i.MX6 Quad SABRE Smart Device Board
Board: MX6-SabreSD
MANGO-IMX6Q-DRAM: 2 GiB
fec_phy_reset
MMC: FSL_SDHC: 0, FSL_SDHC: 1, FSL_SDHC: 2
Loading Environment from MMC... OK
Display: MANGO-CAP7 (1024x600)
IPU DMFC NORMAL mode: 1(0~1), 5B(4,5), 5F(6,7)
panel size = 1024 x 600
pixel clk = 19531000Hz
crz pixel_fmt=861030210
IPU DMFC DP HIGH RES: 1(0,1), 5B(2~5), 5F(6,7)
```

아래와 같이 에러 발생 시 수정 방법은 아래와 같다.

```
Display: MANGO-PRESS10.4 (800x600)
mxcfb: Error initializing panel.
```

"IPU_PIX_FMT_BGR24" 정의가 되어 있는데, 제대로 동작을 하지 않는다.

```
.pixfmt = IPU_PIX_FMT_BGR24,
```

drivers/video/ipu_disp.c 파일에서 "IPU_PIX_FMT_BGR24" 추가한다.

```
static int ipu_pixfmt_to_map(uint32_t fmt)
{
    switch (fmt) {
        case IPU_PIX_FMT_GENERIC:
        case IPU_PIX_FMT_RGB24:
        case IPU_PIX_FMT_BGR24: //crazyboys 20200907
```

3.1.7. U-boot에서 로고 변경하기

<https://cafe.naver.com/embeddedcrazyboys/40199>

글을 참조해서 변경 한다.

```
tools/logos/
디렉토리에
```

freescale.bmp 파일을 변경하면 됩니다.

해당 파일은

tools/Makefile 에서 보면

아래와 같이 컴파일이 되어 진다.

```
# Use board logo and fallback to vendor
```

```
ifneq $(wildcard $(srctree)/$(src)/logos/$(BOARD).bmp),)
```

```
LOGO_BMP= $(srctree)/$(src)/logos/$(BOARD).bmp
```

```
else
```

```
ifneq $(wildcard $(srctree)/$(src)/logos/$(VENDOR).bmp),)
```

```
LOGO_BMP= $(srctree)/$(src)/logos/$(VENDOR).bmp
```

```
endif
```

```
endif
```

아래 명령이 실행되면서 만들어 진다.

```
$(LOGO_H):      $(obj)/bmp_logo $(LOGO_BMP)
```

```
    $(obj)/bmp_logo --gen-info $(LOGO_BMP) > $@
```

[출처] [u-boot 로고 이미지 변경하기 \(Embedded Crazyboys\)](#) | 작성자 [머털도사](#)

7인치 감압식 LCD를 사용하면

해상도가 800x480이므로 800x480에 8bit에 BMP 파일을 만든다.



QRZ TECHNOLOGY

4. 커널 디바이스 드라이버 포팅

4.1. 커널에서 불필요한 디바이스 삭제

커널에서 Mango-IMX6Q에서 사용하지 않는 Device부터 삭제하고, 디바이별로 포팅하는 것이 좋다.

4.1.1. Mango-IMX6Q에 없는 디바이스 config 없애기

arch/arm/boot/dts/imx6qdl-sabresd.dtsi 파일을 수정하면 된다.

Device tree는 공부를 해야 한다.

```
#if 0 //crazyboys 20200907
    hannstar_cabc {
        compatible = "hannstar,cabc";
        lvds0 {
            gpios = <&gpio6 15 GPIO_ACTIVE_HIGH>;
        };
        lvds1 {
            gpios = <&gpio6 16 GPIO_ACTIVE_HIGH>;
        };
    };
#endif
#if 0 //crazyboys 20200907
    codec: wm8962@1a {
        compatible = "wlf,wm8962";
```

```

    reg = <0x1a>;
    clocks = <&clks IMX6QDL_CLK_CKO>;
    DCVDD-supply = <&reg_audio>;
    DBVDD-supply = <&reg_audio>;
    AVDD-supply = <&reg_audio>;
    CPVDD-supply = <&reg_audio>;
    MICVDD-supply = <&reg_audio>;
    PLLVDD-supply = <&reg_audio>;
    SPKVDD1-supply = <&reg_audio>;
    SPKVDD2-supply = <&reg_audio>;
    gpio-cfg = <
        0x0000 /* 0:Default */
        0x0000 /* 1:Default */
        0x0013 /* 2:FN_DMICCLK */
        0x0000 /* 3:Default */
        0x8014 /* 4:FN_DMICDAT */
        0x0000 /* 5:Default */
    >;
    amic-mono;
};

mma8451@1c {
    compatible = "fsl,mma8451";
    reg = <0x1c>;
    position = <0>;
    vdd-supply = <&reg_sensor>;
    vddio-supply = <&reg_sensor>;
    interrupt-parent = <&gpio1>;
    interrupts = <18 8>;
    interrupt-route = <1>;
};

ov564x: ov564x@3c {
    compatible = "ovti,ov564x";
    reg = <0x3c>;
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_ipu1_2>;
    clocks = <&clks IMX6QDL_CLK_CKO>;

```

```

        clock-names = "csi_mclk";
        DOVDD-supply = <&vgen4_reg>; /* 1.8v */
        AVDD-supply = <&vgen3_reg>; /* 2.8v, on rev C board is VGEN3,
                                   on rev B board is VGEN5 */
        DVDD-supply = <&vgen2_reg>; /* 1.5v*/
        pwn-gpios = <&gpio1 16 1>; /* active low: SD1_DAT0 */
        rst-gpios = <&gpio1 17 0>; /* active high: SD1_DAT1 */
        csi_id = <0>;
        mclk = <24000000>;
        mclk_source = <0>;
    };
#endif
#if 0 //crazyboys 0907
    egalax_ts@04 {
        compatible = "eeti,egalax_ts";
        reg = <0x04>;
        pinctrl-names = "default";
        pinctrl-0 = <&pinctrl_i2c2_egalax_int>;
        interrupt-parent = <&gpio6>;
        interrupts = <8 2>;
        wakeup-gpios = <&gpio6 8 0>;
    };

    max11801@48 {
        compatible = "maxim,max11801";
        reg = <0x48>;
        interrupt-parent = <&gpio3>;
        interrupts = <26 2>;
        work-mode = <1>;/*DCM mode*/
    };

    pmic: pfuze100@08 {
        compatible = "fsl,pfuze100";
        reg = <0x08>;
    };
#endif
#if 0
    ov564x_mipi: ov564x_mipi@3c { /* i2c2 driver */
        compatible = "ovti,ov564x_mipi";

```



```

reg = <0x3c>;
clocks = <&clks 201>;
clock-names = "csi_mclk";
DOVDD-supply = <&vgen4_reg>; /* 1.8v */
AVDD-supply = <&vgen3_reg>; /* 2.8v, rev C board is VGEN3
                        rev B board is VGEN5 */
DVDD-supply = <&vgen2_reg>; /* 1.5v*/
pwn-gpios = <&gpio1 19 1>; /* active low: SD1_CLK */
rst-gpios = <&gpio1 20 0>; /* active high: SD1_DAT2 */
csi_id = <1>;
mclk = <24000000>;
mclk_source = <0>;

};
#endif

```

4.1.2. PMIC 없으므로 LDO로 변경하기

arch/arm/boot/dts/imx6qdl-sabresd.dtsi 파일 수정

```

&reg_arm {
    /delete-property/ vin-supply;
};

&reg_pu {
    /delete-property/ vin-supply;
};

&reg_soc {
    /delete-property/ vin-supply;
};

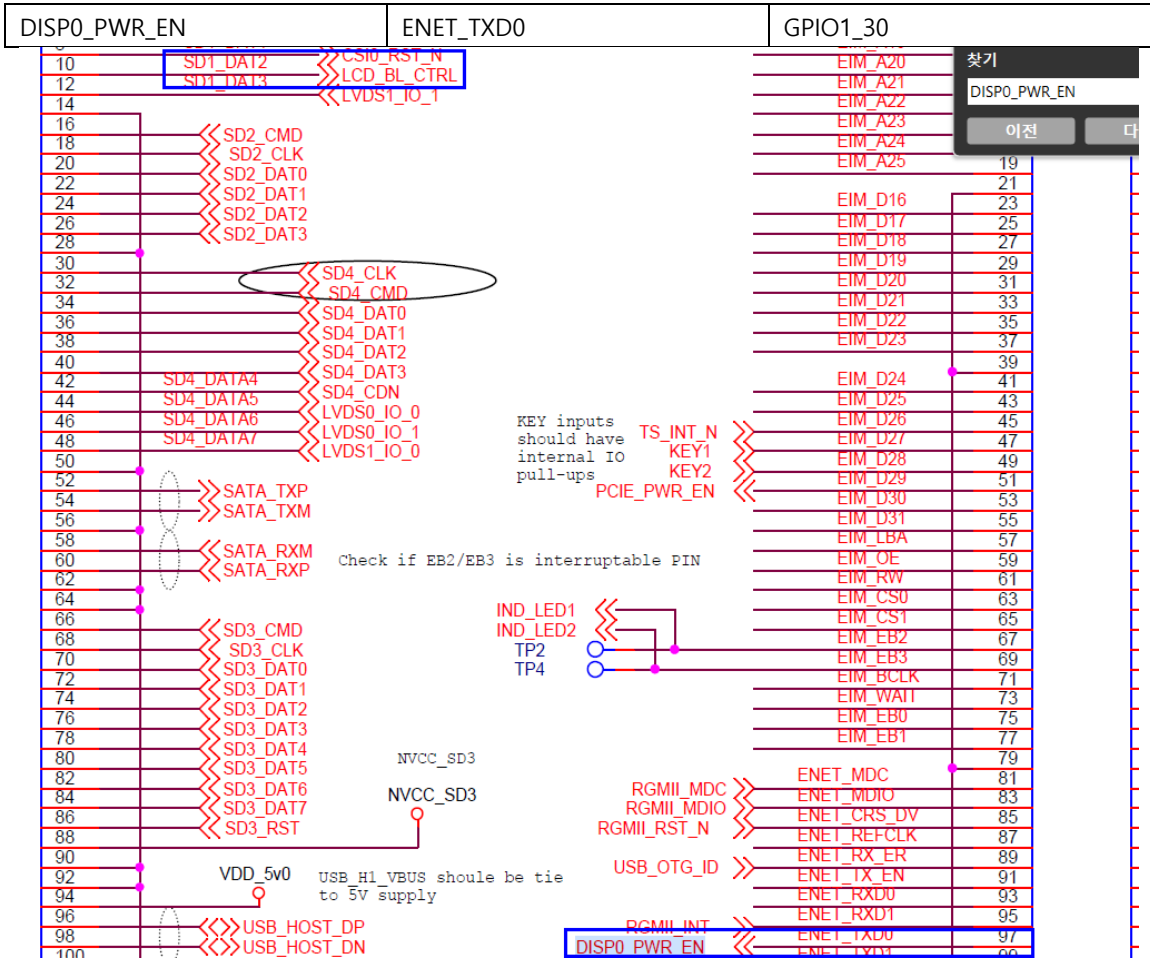
```

4.2. Display , 터치 드라이브 포팅

4.2.1. LCD Power EN, Backlight GPIO 포팅

LCD 화면을 디스플레이하기 위해서는 아래 회로도에 GPIO 핀을 컨트롤해야 한다.

구분	Pad Name	GPIO
LCD_BL_CTRL	SD1_DAT2	GPIO1_19



arch/arm/boot/dts/imx6qdl-sabresd.dtsi 파일에서 아래와 같이 수정을 한다.

```

mxcfb1: fb@0 {
    compatible = "fsl,mxc_sdc_fb";
    disp_dev = "lcd";
    interface_pix_fmt = "BGR24";
    //interface_pix_fmt = "RGB24";
    mode_str = "MANGO-CAP7";
    default_bpp = <24>;
    int_clk = <0>;
    late_init = <0>;
    status = "disabled";
};

lcd@0 {
    compatible = "fsl,lcd";

```

```

ipu_id = <0>;
disp_id = <0>;
default_ifmt = "BGR24";
pinctrl-names = "default";
pinctrl-0 = <&pinctrl_ipu1>;
power_en_gpio = <&gpio1 30 0>; /* Power EN */
backlight_ctl_gpio = <&gpio1 19 0>; /* Backlight CTRL */
status = "okay";

```

Default_ifmt가 BGR24인 것은 LCD와 연결하는 핀맵이 B->G->R로 연결되어 있다.

"power_en_gpio"와 "backlight_ctl_gpio"는
drivers/video/fbdev/mxc/mxc_lcdif.c 파일에 추가를 한다.

```

//crazyboys 20200908
#include <linux/gpio.h>
#include <linux/of_gpio.h>

static int mxc_lcdif_probe(struct platform_device *pdev)
{
    int ret;
    struct pinctrl *pinctrl;
    struct mxc_lcdif_data *lcdif;
    struct mxc_lcd_platform_data *plat_data;
//crazyboys 20200908
    struct device_node *np = pdev->dev.of_node;
    enum of_gpio_flags flags;
    unsigned int pwr_gpio,backlight_gpio;
    int status=-1;
//end
    ...
//crazyboys 20200908
    #if 1
        pwr_gpio = of_get_named_gpio_flags(np, "power_en_gpio", 0, &flags);
        if(pwr_gpio == -EPROBE_DEFER){
            dev_err(&pdev->dev, "LCD Power Enalbe Gpio fail\n");
            return pwr_gpio;
        }

        status = devm_gpio_request_one(&pdev->dev,pwr_gpio,GPIOF_OUT_INIT_HIGH,NULL);

```

```

if(status < 0 )
{
    dev_err(&pdev->dev, "failed to request gpio %d: %d\n",pwr_gpio,status);
    return status;
}

backlight_gpio = of_get_named_gpio_flags(np, "backlight_ctl_gpio", 0, &flags);
if(backlight_gpio == -EPROBE_DEFER){
    dev_err(&pdev->dev, "Backlight Ctrl Gpio fail\n");
    return backlight_gpio;
}

status = devm_gpio_request_one(&pdev->dev,backlight_gpio,GPIOF_OUT_INIT_HIGH,NULL);
if(status < 0 )
{
    dev_err(&pdev->dev, "Backlight Ctrl GPIO Request Fail status %d\n",status);
    return status;
}
#endif

```

probe하면서 GPIO를 High한다.

PWM으로 Backlight를 조정할 수 있다.

IMX6DQRM-Reference Manual.pdf 파일을 보면 PWM2_OUT으로 사용 할 수 있다.

SD1_DAT2	ALT0	SD1_DATA2	HYS - ENABLED	SW_PAD_CTL_PAD_SD1_DATA2
	ALT1	ECSPI5_SS1	PUS - 100K_OHM_PU	
	ALT2	GPT_COMPARE2	PUE - PULL	
	ALT3	PWM2_OUT	PKE - ENABLED	
	ALT4	WDOG1_B		

Table continues on the next page...

Chapter 4 External Signals and Pin Multiplexing

Table 4-1. Pin Assignments (continued)

Pad Name	Mode	Signal	Pad Settings	Pad/Group Registers
	ALT5	GPI01_IO19	ODE - DISABLED	
	ALT6	WDOG1_RESET_B_DEB	SPEED - MEDIUM DSE - 40_OHM SRE - SLOW	

arch/arm/boot/dts/imx6qdl-sabresd.dtsi 파일 수정

```
backlight {
    compatible = "pwm-backlight";
    //pwms = <&pwm2 0 50000>;
    pwms = <&pwm2 0 400000>;//2.5KHz
    //pwms = <&pwm2 0 40000>;//25KHz
    //pwms = <&pwm2 0 4000>;//250KHz
    brightness-levels = <
        0 1 2 3 4 5 6 7 8 9
        10 11 12 13 14 15 16 17 18 19
        20 21 22 23 24 25 26 27 28 29
        30 31 32 33 34 35 36 37 38 39
        40 41 42 43 44 45 46 47 48 49
        50 51 52 53 54 55 56 57 58 59
        60 61 62 63 64 65 66 67 68 69
        70 71 72 73 74 75 76 77 78 79
        80 81 82 83 84 85 86 87 88 89
        90 91 92 93 94 95 96 97 98 99
        100
    >;
    default-brightness-level = <100>;
}
```

```
        status = "okay";  
};
```

```
pinctrl_pwm2: pwm2grp {  
    fsl,pins = <  
        MX6QDL_PAD_SD1_DAT2_PWM2_OUT    0x1b0b1  
    >;  
};  
  
&pwm2 {  
    pinctrl-names = "default";  
    pinctrl-0 = <&pinctrl_pwm2>;  
    status = "okay";  
};
```

테스트 방법

```
# echo 10 > /sys/class/backlight/backlight/brightness => 어둡게  
# echo 100 > /sys/class/backlight/backlight/brightness => 밝게
```

4.2.2. 7인치 감압식 LCD 드라이버 포팅

드라이버 포팅을 위해서 7인치 감압식 LCD(AT070TN94) 데이터 시트, Mango-IMX6Q 회로도가 필요하다. LCD 드라이버에 타이밍을 보기 위해서 7인치 감압식 LCD 데이터 시트가 필요하다. 데이터시트를 보면 타이밍이 아래와 같이 나와 있다.

Item	Symbol	Values			Unit	Remark
		Min.	Typ.	Max.		
Horizontal Display Area	thd	-	800	-	DCLK	
DCLK Frequency	fclk	26.4	33.3	46.8	MHz	
One Horizontal Line	th	862	1056	1200	DCLK	
HS pulse width	thpw	1	-	40	DCLK	
HS Blanking	thb	46	46	46	DCLK	
HS Front Porch	thfp	16	210	354	DCLK	

Item	Symbol	Values			Unit	Remark
		Min.	Typ.	Max.		
Vertical Display Area	tvd	-	480	-	TH	
VS period time	tv	510	525	650	TH	
VS pulse width	tvpw	1	-	20	TH	
VS Blanking	tvb	23	23	23	TH	
VS Front Porch	tvfp	7	22	147	TH	

타이밍 표를 보고, 타이밍을 맞추면 된다.

DCLK Frequency를 26.4MHz로 했을 때 타이밍 값이다.

drivers/video/fbdev/mxc/mxc_lcdif.c 파일에 static struct fb_videomode lcdif_modedb[] 배열에 추가하면 된다.

```
"MANGO-PRESS7", //name
    60, //refresh
    800, //xres
    480, //yres
    37879, //pixclock(ns)
    46, //left_margin(HBP)
    16, //right_margin(HFP)
    23, //upper_margin(VBP)
    7, //lower_margin(VFP)
    1, //hsync_len
```

```

1, //vsync_len
FB_SYNC_CLK_LAT_FALL, //sync
FB_VMODE_NONINTERLACED | FB_VMODE_ODD_FLD_FIRST, //vmode
0, //flag

```

터치 드라이버 포팅 감압식 LCD이다.

중간에 tsc2007 칩이 있다.

arch/arm/boot/dts/imx6qdl-sabresd.dtsi 파일에 추가

```

touchscreen: tsc2007@4a {
    compatible = "ti,tsc2007";
    reg = <0x4a>;
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_mango_ts>;
    interrupt-parent = <&gpio3>;
    interrupts = <26 0>;
    gpios = <&gpio3 26 GPIO_ACTIVE_LOW>;
    ti,x-plate-ohms = <660>;
    linux,wakeup;
};

```

커널 드라이버는 drivers/input/touchscreen/tsc2007_core.c 사용하면 된다.

테스트 방법은 아래와 같이 하면 된다. 리눅스 파일 시스템에서 테스트 했다.

```

# ts_calibrate
xres = 800, yres = 480
Took 7 samples...
Top left : X = 337 Y = 534
Took 7 samples...
Top right : X = 3727 Y = 590
Took 8 samples...
Bot right : X = 3683 Y = 3496
Took 7 samples...
Bot left : X = 314 Y = 3519
Took 5 samples...
Center : X = 2045 Y = 2030
-23.375244 0.207105 0.002348
-21.038757 -0.000633 0.128980
Calibration constants: -1531920 13572 153 -1378796 -41 8452 65536
[CRZ] drivers/input/touchscreen/tsc2007_core.c (193) tsc2007_stop:
# ts_test

```


4.2.3. 10.1인치 정전식 LCD 드라이버 포팅

10.1인치 타이밍은 아래와 같이 맞춘다.

drivers/video/fbdev/mxc/mxc_lcdif.c 파일 수정

```
"MANGO-CAP10.1",          //name
    60,                    //refresh
    1280,                  //xres
    800,                   //yres
    13468,                 //pixclock(ns)
    68,                    //left_margin(HBP)
    60,                    //right_margin(HFP)
    8,                     //upper_margin(VBP)
    8,                     //lower_margin(VFP)
    2,                     //hsync_len
    2,                     //vsync_len
    0, //sync
    FB_VMODE_NONINTERLACED, //vmode
    0, //flag
```

테스트 결과

```
# fbset

mode "1280x800-64"
    # D: 74.250 MHz, H: 52.660 kHz, V: 64.376 Hz
    geometry 1280 800 1280 800 24
    timings 13468 68 60 8 8 2 2
    accel false
    rgba 8/16,8/8,8/0,0/0
endmode
```

4.2.4. 정전식 드라이버 포팅하기

7인치 정전식 LCD와 10.1인치 정전식 LCD에 사용하는 터치 드라이버이다.

정전식 터치를 사용한다.

CX-IMX6Q 회로도를 보면, I2C 2번 bus를 사용한다.

인터럽터 핀과 터치 리셋 핀이 있다.

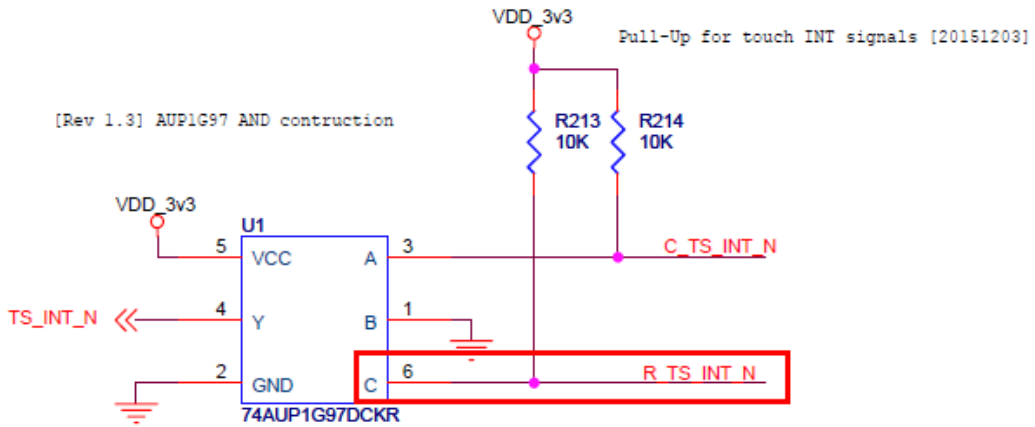
구분	핀 이름	GPIO	
CT_RST_N	SD1_CMD	GPIO1_IO18	
C_TS_INT_N	EIM_D26	GPIO3_IO26	

우선 핀을 device tree에서 정의를 한다.

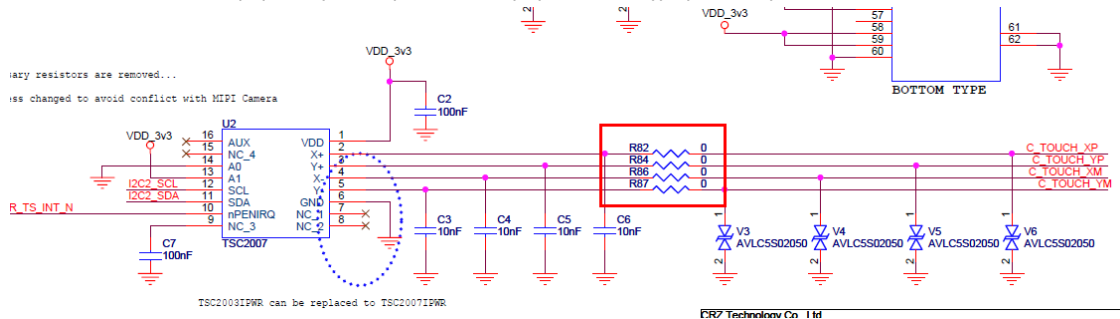
arch/arm/boot/dts/imx6qdl-sabresd.dtsi 파일 수정

```
pinctrl_mango_ts: mango_ts {
    fsl,pins = <
        /* Reset*/
        MX6QDL_PAD_SD1_CMD_GPIO1_IO18 0x17059
        /* Interrupt */
        MX6QDL_PAD_EIM_D26_GPIO3_IO26 0x17059
    >;
};
```

U1 부품 6번 PIN이 LOW를 유지한다.



원인은 TSC2007 칩에서 인터럽트가 발생을 하여 LOW로 유지를 한다.



CR-LIF2-CT101R001 LCD보드에서 TSC2007 과 연결되는 C_TOUCH_XP, C_TOUCH_YP, C_TOUCH_XM, C_TOUCH_YM 핀이 Ground로 되어 있어, 터치가 눌린 것으로 인식한다.

정전식 Touch를 사용할 경우 TSC2007 칩에 IRQ Disable을 하여 사용하도록 수정

COMMAND BYTE

Table 2. Command Byte Definition (Excluding the Setup Command)⁽¹⁾

BIT	NAME	DESCRIPTION
D7-D4	C3-C0	All Converter Function Select bits as detailed in Table 3, except for the setup command ('1011').
D3-D2	PD1-PD0	00: Power down between cycles. PENIRQ enabled. 01: A/D converter on. PENIRQ disabled. 10: A/D converter off. PENIRQ enabled. 11: A/D converter on. PENIRQ disabled.
D1	M	0: 12-bit (Lower speed referred to as the 2MHz clock). 1: 8-bit (Higher speed referred to as the 4MHz clock).
D0	X	Don't care.

커널 imx6qdl-sabresd.dtsi 파일 수정한다.

```
mango-ts@38 {
    compatible = "mango,mango-ts";
    reg = <0x38>;
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_mango_ts>;
    interrupt-parent = <&gpio3>;
    interrupts = <26 0>;
    //resets = <&mango_ts_reset>;
    reset-gpios = <&gpio1 18 GPIO_ACTIVE_LOW>;
};

tsc2007@4a {
    compatible = "ti,tsc2007";
    reg = <0x4a>;
    //pinctrl-names = "default";
    //pinctrl-0 = <&pinctrl_mango_ts>;
    //interrupt-parent = <&gpio3>;
    //interrupts = <26 0>;
    //gpios = <&gpio3 26 GPIO_ACTIVE_LOW>;
    // ti,x-plate-ohms = <660>;
    // linux,wakeup;
};
```

터치 드라이버는 <http://crztech.iptime.org:8080/Release/mango-imx6q/Android6.0/20171204/mango-imx6qxf-android6.0-10.1inch-sd-wifi-5M-MIPI-src-20171204.tgz> 에서 터치 드라이버를 추출한다.

mango_ft5x06_ts.c , mango_ft5x06_ts.h 파일이다.

컴파일 시 에러가 발생을 한다.

```
drivers/input/touchscreen/mango_ft5x06_ts.c:1419:13: error: 'struct file' has no member named
'f_dentry'
drivers/input/touchscreen/mango_ft5x06_ts.c:2097:1: error: type defaults to 'int' in declaration of
'MODULE_DEVICE_TABLE' [-Werror=implicit-int]
drivers/input/touchscreen/mango_ft5x06_ts.c:2147:1: error: type defaults to 'int' in declaration of
'module_init' [-Werror=implicit-int]
drivers/input/touchscreen/mango_ft5x06_ts.c:2151:20: error: expected declaration specifiers or '...'
before string constant
    MODULE_DESCRIPTION("FocalTech ft5x0x TouchScreen driver");
                        ^
drivers/input/touchscreen/mango_ft5x06_ts.c:2152:16: error: expected declaration specifiers or '...'
before string constant
    MODULE_LICENSE("GPL");
```

<디버깅>

```
drivers/input/touchscreen/mango_ft5x06_ts.c:1419:13: error: 'struct file' has no member named
'f_dentry'
    => Include/linux/fs.h 파일에 정의가 되어 있지 않다. 아래와 같이 수정한다.
    struct file {
        union {
            struct llist_node    fu_llist;
            struct rcu_head      fu_rcuhead;
        } f_u;
        struct path              f_path;
#define f_dentry              f_path.dentry //crazyboys 20200910
```

drivers/input/touchscreen/mango_ft5x06_ts.c 파일에서 include 파일 수정.

드라이버가 3.x 커널에서 사용하던 거라, 커널 4.14로 되면서 많이 변경이 되었네요.

```
#include <linux/module.h>
#include <linux/init.h>
#include <linux/i2c.h>
#include <linux/interrupt.h>
#include <linux/input.h>
#include <linux/irq.h>
#include <linux/gpio.h>
#include <linux/delay.h>
#include <linux/slab.h>
#include <linux/bitops.h>
```

```
#include <linux/input/mt.h>
#include <linux/of_gpio.h>
#include <linux/reset.h>
#include <linux/fs.h>
```

drivers/input/touchscreen/Makefile 파일 추가

```
obj-$(CONFIG_MANGO_TOUCH_FT5x06_101INCH) += mango_ft5x06_ts.o
obj-$(CONFIG_MANGO_TOUCH_FT5x06_7INCH) += mango_ft5x06_ts.o
```

drivers/input/touchscreen/Kconfig 파일에 아래 내용 추가

```
config MANGO_TOUCH_FT5x06_7INCH
    tristate "crztech mango 7 inch FT5x06 PCAP touch "
    depends on INPUT_TOUCHSCREEN
    default n
    help
        Say Y here to enable the driver for the 10.1" FT5x06 PCAP touchscreen on the Mango
        board.
        If unsure, say N. To compile this driver as a module, choose M here: the module will be
        called s3c_ts.
config MANGO_TOUCH_FT5x06_7INCH_90_ROT
    tristate "90 Rotation mango 7 inch FT5x06 PCAP touch "
    depends on INPUT_TOUCHSCREEN
    default n
    help
        Say Y here to enable the driver for the 7" FT5x06 PCAP touchscreen on the Mango board.
        If unsure, say N. To compile this driver as a module, choose M here: the module will be
        called s3c_ts.
config MANGO_TOUCH_FT5x06_7INCH_270_ROT
    tristate "270 Rotation mango 7 inch FT5x06 PCAP touch "
    depends on INPUT_TOUCHSCREEN
    default n
    help
        Say Y here to enable the driver for the 7" FT5x06 PCAP touchscreen on the Mango board.
        If unsure, say N. To compile this driver as a module, choose M here: the module will be
        called s3c_ts.
config MANGO_TOUCH_FT5x06_101INCH
    tristate "crztech mango 10.1 inch FT5x06 PCAP touch "
    depends on INPUT_TOUCHSCREEN
```

```

default n
help
    Say Y here to enable the driver for the 10.1" FT5x06 PCAP touchscreen on the Mango
board.
    If unsure, say N. To compile this driver as a module, choose M here: the module will be
called s3c_ts.
config MANGO_TOUCH_FT5x06_10INCH_90_ROT
    tristate "90 Rotation mango 10.1 inch FT5x06 PCAP touch "
    depends on INPUT_TOUCHSCREEN
    default n
    help
        Say Y here to enable the driver for the 10.1" FT5x06 PCAP touchscreen on the Mango
board.
        If unsure, say N. To compile this driver as a module, choose M here: the module will be
called s3c_ts.
config MANGO_TOUCH_FT5x06_10INCH_270_ROT
    tristate "270 Rotation mango 10.1 inch FT5x06 PCAP touch "
    depends on INPUT_TOUCHSCREEN
    default n
    help
        Say Y here to enable the driver for the 10.1" FT5x06 PCAP touchscreen on the Mango
board.
        If unsure, say N. To compile this driver as a module, choose M here: the module will be
called s3c_ts.

```

커널 Configuration에서 CONFIG_MANGO_TOUCH_FT5x06_101INCH=y 추가한다.

tsc2007.c 파일에서 수정 한다.

drivers/input/touchscreen/Kconfig 파일에 아래 내용 추가

```

config MANGO_TSC2007_IRQ_DISABLE
    tristate "TSC2007 based touchscreens irq disable"
    depends on I2C
    help
        Say Y here if you have a TSC2007 based touchscreen irq disable.

    If unsure, say N.

```

To compile this driver as a module, choose M here: the module will be called tsc2007.

"tsc2007_core.c" 파일 수정

```
static void tsc2007_stop(struct tsc2007 *ts)
{
    int err;//crazyboys 20160517
    //MANGO_DBG("#n");
    ts->stopped = true;
    mb();
    wake_up(&ts->wait);

    disable_irq(ts->irq);
#ifdef CONFIG_MANGO_TSC2007_IRQ_DISABLE//crazyboys 20160517
    err = tsc2007_xfer(ts, ADC_ON_12BIT);
#endif

static int tsc2007_open(struct input_dev *input_dev)
{
    struct tsc2007 *ts = input_get_drvdata(input_dev);
    int err;

    ts->stopped = false;
    mb();

    enable_irq(ts->irq);
#ifdef CONFIG_MANGO_TSC2007_IRQ_DISABLE //crazyboys 20200910
    /* Prepare for touch readings - power down ADC and enable PENIRQ */
    err = tsc2007_xfer(ts, PWRDOWN);
    if (err < 0) {
        tsc2007_stop(ts);
        return err;
    }
#endif
    return 0;
}

static int tsc2007_probe(struct i2c_client *client,
```

```

        const struct i2c_device_id *id)
{
    const struct tsc2007_platform_data *pdata =
        dev_get_platdata(&client->dev);
    struct tsc2007 *ts;
    struct input_dev *input_dev;
    int err;

    if (!i2c_check_functionality(client->adapter,
                                I2C_FUNC_SMBUS_READ_WORD_DATA))
        return -EIO;

    ts = devm_kzalloc(&client->dev, sizeof(struct tsc2007), GFP_KERNEL);
    if (!ts)
        return -ENOMEM;
#ifdef CONFIG_MANGO_TSC2007_IRQ_DISABLE //crazyboys 20200910
    if (pdata)
        err = tsc2007_probe_pdev(client, ts, pdata, id);
    else
        err = tsc2007_probe_dt(client, ts);
    if (err)
        return err;
#endif

#ifdef CONFIG_MANGO_TSC2007_IRQ_DISABLE
    err = devm_request_threaded_irq(&client->dev, ts->irq,
                                   tsc2007_hard_irq, tsc2007_soft_irq,
                                   IRQF_ONESHOT,
                                   client->dev.driver->name, ts);

    if (err) {
        dev_err(&client->dev, "Failed to request irq %d: %d\n",
                ts->irq, err);
        return err;
    }
#endif
#ifdef CONFIG_MANGO_TSC2007_IRQ_DISABLE
}
#endif
}
#endif

```



```

/* power down the chip (TSC2007_SETUP does not ACK on I2C) */
err = tsc2007_xfer(ts, PWRDOWN);
if (err < 0) {
    dev_err(&client->dev,
            "Failed to setup chip: %d\n", err);
    return err;    /* chip does not respond */
}
err = input_register_device(input_dev);
if (err) {
    dev_err(&client->dev,
            "Failed to register input device: %d\n", err);
    return err;
}
err = tsc2007_iio_configure(ts);
if (err) {
    dev_err(&client->dev,
            "Failed to register with IIO: %d\n", err);
    return err;
}
}
#endif

```

에러 발생

```

[CRZ] drivers/input/touchscreen/tsc2007_core.c (342) tsc2007_probe:
imx6q-pinctrl 20e0000.iomuxc: pin MX6Q_PAD_SD1_CMD already requested by 20e0000.iomuxc;
cannot claim for 1-0038
imx6q-pinctrl 20e0000.iomuxc: pin-210 (1-0038) status -22
imx6q-pinctrl 20e0000.iomuxc: could not request pin 210 (MX6Q_PAD_SD1_CMD) from group
mango_ts on device 20e0000.iomuxc
mango-ts 1-0038: Error applying setting, reverse things back
mango-ts: probe of 1-0038 failed with error -22
snvs_rtc 20cc000.snvs:snvs-rtc-lp: rtc core: registered 20cc000.snvs:snvs-rtc-lp as rtc0

```

MX6Q_PAD_SD1_CMD 이중으로 define되어있다는 것이다.

중복으로 선언 된 부분을 삭제하면 된다.

커널 메시지에서 아래와 같이 나오면, 인식이 된 것이다.

```

[CRZ] drivers/input/touchscreen/mango_ft5x06_ts.c (1792) ft5x06_i2c_ts_probe_dt:
input: mango-ts as /devices/virtual/input/input1

```

```
CRZ ft5x0x_ts_probe reset timer start
-----[FTS] Firmware version = 0x61
-----[FTS] report rate is 0Hz.
-----[FTS] touch threshold is 500.
```

테스트 방법과 확인 방법은 아래와 같다.

```
# cat /proc/bus/input/devices
I: Bus=0018 Vendor=0000 Product=0000 Version=0000
N: Name="mango-ts"
P: Phys=
S: Sysfs=/devices/virtual/input/input1
U: Uniq=
H: Handlers=kbd event0
B: PROP=0
B: EV=b
B: KEY=400 0 4 0 0 0 0 c0000 0 0 0
B: ABS=2650000 1000000
```

4.2.5. 7인치 정전식 LCD 드라이버 포팅

커널 Configuration 한다.

```
CONFIG_MANGO_TOUCH_FT5x06_7INCH=y
CONFIG_MANGO_TSC2007_IRQ_DISABLE=y
```

타이밍은 아래와 같이 세팅을 한다.

```
"MANGO-CAP7", //name
    60, //refresh
    1024, //xres
    600, //yres
    19531, //pixclock(ns)
    60, //left_margin(HBP)
    60, //right_margin(BFP)
    5, //upper_margin(VBP)
    5, //lower_margin(VFP)
    200, //hsync_len
    25, //vsync_len
    0, //sync
    FB_VMODE_NONINTERLACED, //vmode
    0, //flag
```

커널 로그에서 터치가 인식이 되면 아래와 같이 나오면 정상이다.

```
[CRZ] drivers/input/touchscreen/tsc2007_core.c (342) tsc2007_probe:
[CRZ] drivers/input/touchscreen/tsc2007_core.c (193) tsc2007_stop:
[CRZ] drivers/input/touchscreen/mango_ft5x06_ts.c (1828) ft5x0x_ts_probe: [FTS] ft5x0x_ts_probe,
driver version is 3.0.
[CRZ] drivers/input/touchscreen/mango_ft5x06_ts.c (1792) ft5x06_i2c_ts_probe_dt:
[CRZ] drivers/input/touchscreen/mango_ft5x06_ts.c (1865) ft5x0x_ts_probe: tsdata->reset_pin=18
usb 1-1: new high-speed USB device number 2 using ci_hdrc
input: mango-ts as /devices/virtual/input/input1
CRZ ft5x0x_ts_probe reset timer start
usb 1-1: New USB device found, idVendor=05e3, idProduct=0608
usb 1-1: New USB device strings: Mfr=0, Product=1, SerialNumber=0
usb 1-1: Product: USB2.0 Hub
hub 1-1:1.0: USB hub found
hub 1-1:1.0: 4 ports detected
-----[FTS] Firmware version = 0x13
-----[FTS] report rate is 100Hz.
-----[FTS] touch threshold is 200.
```

“fbset” 명령으로 확인 시 아래와 같다.

```
mode "1024x600-60"
    # D: 51.201 MHz, H: 38.096 kHz, V: 59.993 Hz
    geometry 1024 600 1024 600 32
    timings 19531 60 60 5 5 200 25
    accel false
    rgba 8/16,8/8,8/0,8/24
endmode
```

```
# cat /sys/class/graphics/fb0/mode
U:1024x600p-60
```

4.2.6. PMIC 없으므로 LDO로 변경하기

4.3. USB Host 확인

arch/arm/boot/dts/imx6qdl-sabresd.dtsi 아래와 같이 정의가 되어 있으면 된다.

```
&usbh1 {
```

```
vbus-supply = <&reg_usb_h1_vbus>;  
status = "okay";  
};
```

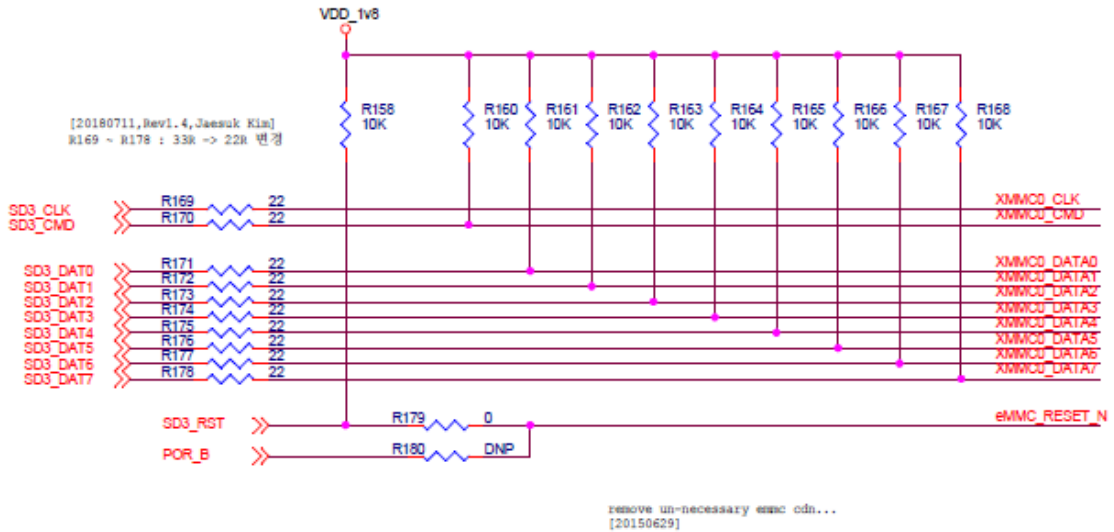
확인 방법

USB Storage를 꽂아서 확인하면 된다.

```
usb 1-1.2: new high-speed USB device number 3 using ci_hdrc  
usb 1-1.2: New USB device found, idVendor=05e3, idProduct=0723  
usb 1-1.2: New USB device strings: Mfr=3, Product=4, SerialNumber=2  
usb 1-1.2: Product: USB Storage  
usb 1-1.2: Manufacturer: Generic  
usb 1-1.2: SerialNumber: 000000009451  
usb-storage 1-1.2:1.0: USB Mass Storage device detected  
usb-storage 1-1.2:1.0: Quirks match for vid 05e3 pid 0723: 8000  
scsi host0: usb-storage 1-1.2:1.0  
scsi 0:0:0:0: Direct-Access    Generic  STORAGE DEVICE   9451 PQ: 0 ANSI: 0  
sd 0:0:0:0: [sda] 15628288 512-byte logical blocks: (8.00 GB/7.45 GiB)  
sd 0:0:0:0: [sda] Write Protect is off  
sd 0:0:0:0: [sda] No Caching mode page found  
sd 0:0:0:0: [sda] Assuming drive cache: write through  
sda: sda1
```

4.4. eMMC 모듈 테스트

Mango-IMX6Q에서는 USDHC3이 eMMC로 사용한다.



arch/arm/boot/dts/imx6qdl-sabresd.dtsi 파일에서 cd-gpios, wp-gpios 핀을 사용하지 않으므로 아래와 같이 수정한다. HS200 모드로 동작을 시킨다.

```
&usdhc3 {
    pinctrl-names = "default", "state_100mhz", "state_200mhz";
    pinctrl-0 = <&pinctrl_usdhc3>;
    pinctrl-1 = <&pinctrl_usdhc3_100mhz>;
    pinctrl-2 = <&pinctrl_usdhc3_200mhz>;
    bus-width = <8>;
    //cd-gpios = <&gpio2 0 GPIO_ACTIVE_LOW>;
    //wp-gpios = <&gpio2 1 GPIO_ACTIVE_HIGH>;
    //sd-uhs-sdr104;
    mmc-hs200-1_8v;
    keep-power-in-suspend;
    enable-sdio-wakeup;
    status = "okay";
};
```

eMMC_RESET_N 핀은 SD3_RST 핀에 연결되어 있다.

```
pinctrl_usdhc3: usdhc3grp {
    fsl,pins = <
        MX6QDL_PAD_SD3_CMD__SD3_CMD        0x17059
        MX6QDL_PAD_SD3_CLK__SD3_CLK        0x10059
        MX6QDL_PAD_SD3_DAT0__SD3_DATA0     0x17059
        MX6QDL_PAD_SD3_DAT1__SD3_DATA1     0x17059
        MX6QDL_PAD_SD3_DAT2__SD3_DATA2     0x17059
    >
```

```

        MX6QDL_PAD_SD3_DAT3__SD3_DATA3      0x17059
        MX6QDL_PAD_SD3_DAT4__SD3_DATA4      0x17059
        MX6QDL_PAD_SD3_DAT5__SD3_DATA5      0x17059
        MX6QDL_PAD_SD3_DAT6__SD3_DATA6      0x17059
        MX6QDL_PAD_SD3_DAT7__SD3_DATA7      0x17059
        MX6QDL_PAD_SD3_RST__SD3_RESET       0x17059
//crazyboys 20200910
    };
};

pinctrl_usdhc3_100mhz: usdhc3grp100mhz {
    fsl,pins = <
        MX6QDL_PAD_SD3_CMD__SD3_CMD         0x170b9
        MX6QDL_PAD_SD3_CLK__SD3_CLK         0x100b9
        MX6QDL_PAD_SD3_DAT0__SD3_DATA0      0x170b9
        MX6QDL_PAD_SD3_DAT1__SD3_DATA1      0x170b9
        MX6QDL_PAD_SD3_DAT2__SD3_DATA2      0x170b9
        MX6QDL_PAD_SD3_DAT3__SD3_DATA3      0x170b9
        MX6QDL_PAD_SD3_DAT4__SD3_DATA4      0x170b9
        MX6QDL_PAD_SD3_DAT5__SD3_DATA5      0x170b9
        MX6QDL_PAD_SD3_DAT6__SD3_DATA6      0x170b9
        MX6QDL_PAD_SD3_DAT7__SD3_DATA7      0x170b9
    >;
};

pinctrl_usdhc3_200mhz: usdhc3grp200mhz {
    fsl,pins = <
        MX6QDL_PAD_SD3_CMD__SD3_CMD         0x170f9
        MX6QDL_PAD_SD3_CLK__SD3_CLK         0x100f9
        MX6QDL_PAD_SD3_DAT0__SD3_DATA0      0x170f9
        MX6QDL_PAD_SD3_DAT1__SD3_DATA1      0x170f9
        MX6QDL_PAD_SD3_DAT2__SD3_DATA2      0x170f9
        MX6QDL_PAD_SD3_DAT3__SD3_DATA3      0x170f9
        MX6QDL_PAD_SD3_DAT4__SD3_DATA4      0x170f9
        MX6QDL_PAD_SD3_DAT5__SD3_DATA5      0x170f9
        MX6QDL_PAD_SD3_DAT6__SD3_DATA6      0x170f9
        MX6QDL_PAD_SD3_DAT7__SD3_DATA7      0x170f9
    >;
};

```

```
};
```

인식이 되는지 확인 한다.

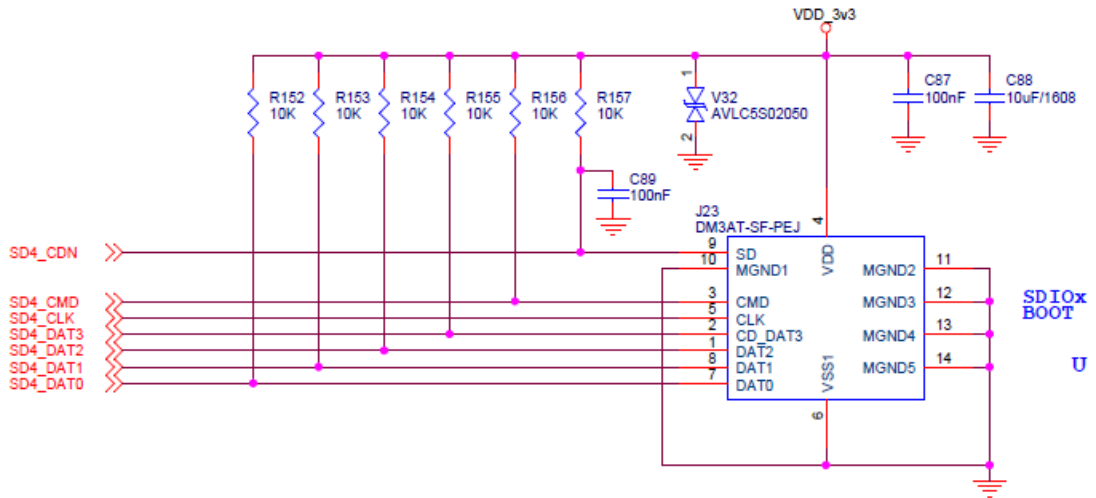
eMMC 모듈 장착부분은 하드웨어 매뉴얼을 참조한다.

```
mmc1: SDHCI controller on 2194000.usdhc [2194000.usdhc] using ADMA
mmc2: SDHCI controller on 2198000.usdhc [2198000.usdhc] using ADMA
sdhci-esdhc-imx 219c000.usdhc: Got CD GPIO
mmc3: SDHCI controller on 219c000.usdhc [219c000.usdhc] using ADMA
mmc2: new DDR MMC card at address 0001
mxv_vpu 2040000.vpu_fsl: VPU initialized
mxv_hdmi_cec soc:hdmi_cec@00120000: HDMI CEC initialized
mmcblk2: mmc2:0001 8GND3R 7.28 GiB
Galcore version 6.2.4.190076
mmcblk2boot0: mmc2:0001 8GND3R partition 1 4.00 MiB
mmcblk2boot1: mmc2:0001 8GND3R partition 2 4.00 MiB
mmcblk2rmpb: mmc2:0001 8GND3R partition 3 512 KiB
mmcblk2: p1 p2 p3 < p5 p6 p7 p8 p9 > p4
```

확인 방법

```
# mount -t debugfs none /sys/kernel/debug
# imx-sdma 20ec000.sdma: external firmware not found, using ROM firmware
cat /sys/kernel/debug/mmc2/ios
clock:          198000000 Hz
actual clock:   198000000 Hz
vdd:           21 (3.3 ~ 3.4 V)
bus mode:       2 (push-pull)
chip select:    0 (don't care)
power mode:     2 (on)
bus width:      3 (8 bits)
timing spec:    9 (mmc HS200)
signal voltage: 1 (1.80 V)
driver type:    0 (driver type B)
# random: crng init done
```

4.5. SDHC4 포팅



SD4_CDN 핀이 있다. GPIO2_IO12 사용한다.

SD4_DATA4	ALT1	SD4_DATA4
	ALT2	UART2_RX_DATA
	ALT5	GPIO2_IO12

Micro SD 카드 Slot에 Micro SD 카드를 삽입하여 테스트 한다.

arch/arm/boot/dts/imx6qdl-sabresd.dtsi 파일 수정

```
pinctrl_usdhc4: usdhc4grp {
    fsl,pins = <
        MX6QDL_PAD_SD4_CMD__SD4_CMD            0x17059
        MX6QDL_PAD_SD4_CLK__SD4_CLK            0x10059
        MX6QDL_PAD_SD4_DAT0__SD4_DATA0         0x17059
        MX6QDL_PAD_SD4_DAT1__SD4_DATA1         0x17059
        MX6QDL_PAD_SD4_DAT2__SD4_DATA2         0x17059
        MX6QDL_PAD_SD4_DAT3__SD4_DATA3         0x17059
        MX6QDL_PAD_SD4_DAT4__GPIO2_IO12        0x13069 /*
crazyboys 20200910 cd pin */
```



```

#if 0 //crazyboys 20200910
                                MX6QDL_PAD_SD4_DAT4__SD4_DATA4      0x17059
                                MX6QDL_PAD_SD4_DAT5__SD4_DATA5      0x17059
                                MX6QDL_PAD_SD4_DAT6__SD4_DATA6      0x17059
                                MX6QDL_PAD_SD4_DAT7__SD4_DATA7      0x17059
#endif

```

```

&usdhc4 {
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_usdhc4>;
    bus-width = <4>;
    cd-gpios = <&gpio2 12 GPIO_ACTIVE_LOW>;
    no-1-8-v;
    keep-power-in-suspend;
    status = "okay";
};

```

4.6. Watchdog 포팅

arch/arm/boot/dts/imx6qdl-sabresd.dtsi 파일 수정

```

&gpc {
    fsl,ldo-bypass = <0>;/* CRZ_icanjji crazyboys 20161201 */
    fsl,wdog-reset = <1>; /* crazyboys 20160225 watchdog select of reset source */
    pu-supply = <&reg_pu>; /* ldo-bypass:use pu_dummy if VDDSOC share with VDDPU */
};

&wdog1 {
    status = "okay";
};

&wdog2 {
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_wdog>;
    fsl,ext-reset-output;
    status = "disabled";
};

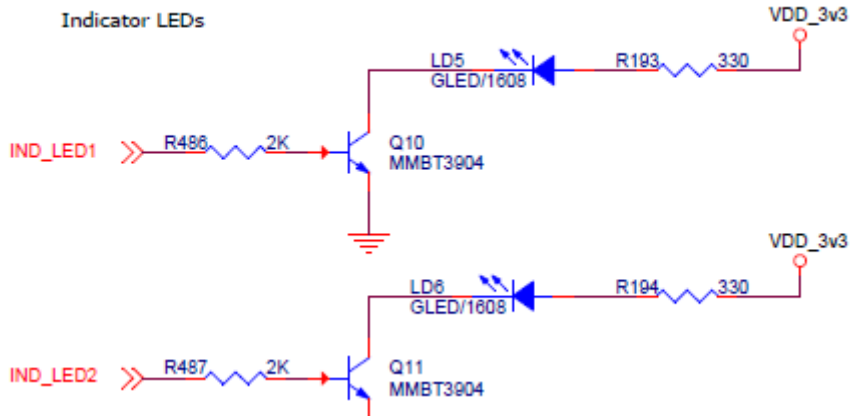
```

Watchdog 테스트 방법

```
cat > /dev/watchdog
Hit return once
Wait 60 seconds
Watchdog reset.
```

4.7. LED 포팅

Mango-IMX6Q LED는 2개가 있다.



IND_LED1	EIM_EB2	GPIO2_IO30
IND_LED2	EIM_EB3	GPIO2_IO31

arch/arm/boot/dts/imx6qdl-sabresd.dtsi 파일에 추가 한다.

```
leds {
    compatible = "gpio-leds";
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_gpio_leds>;

    mango-led1 {
        label = "MANGO-LED1";
        gpios = <&gpio2 30 GPIO_ACTIVE_LOW>;
        linux,default-trigger = "heartbeat";
        default-state = "off";
        status = "okay";
    };
};
```

```

        mango-led2 {
            label = "MANGO-LED2";
            gpios = <&gpio2 31 GPIO_ACTIVE_HIGH>;
            //linux,default-trigger = "heartbeat";
            default-state = "on";
            status = "okay";
        };
};

pinctrl_gpio_leds: ledgrp {
    fsl,pins = <
        /* LED1 By crazyboys */
        MX6QDL_PAD_EIM_EB2__GPIO2_IO30 0x1b0b1
        /* LED2 By crazyboys */
        MX6QDL_PAD_EIM_EB3__GPIO2_IO31 0x1b0b1
    >;
};

```

커널 Configuration은

에러 발생

```

[CRZ] drivers/leds/leds-gpio.c (239) gpio_led_probe:
[CRZ] drivers/leds/leds-gpio.c (164) gpio_leds_create:
leds-gpio: probe of leds failed with error -16

```

Gpio가 이중으로 정의가 되어 있다.

```

reg_sensor: regulator@4 {
    compatible = "regulator-fixed";
    reg = <4>;
    regulator-name = "sensor-supply";
    regulator-min-microvolt = <3300000>;
    regulator-max-microvolt = <3300000>;
    //gpio = <&gpio2 31 0>; //crazyboys 20200915
    startup-delay-us = <500>;
    enable-active-high;
};

```

4.8. USB Device 포팅 및 테스트

arch/arm/boot/dts/imx6qdl-sabresd.dtsi 파일 수정

```

&usb0tg {

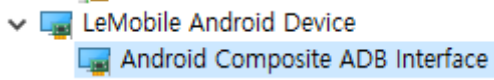
```

```

vbus-supply;//crazyboys 20200915 = <&reg_usb_otg_vbus>;
pinctrl-names = "default";
pinctrl-0 = <&pinctrl_usbotg>;
disable-over-current;
srp-disable;
hnp-disable;
adp-disable;
status = "okay";
};

```

장치관리자에서 아래와 같이 인식이 되면 된다.



4.9. 키버튼 드라이버 포팅하기

버튼이 2개 있다.

```

pinctrl_gpio_keys: gpio_keysgroup {
    fsl,pins = <
        /* KEY1 By crazyboys */
        MX6QDL_PAD_EIM_D27__GPIO3_IO27 0x17059

        /* KEY2 By crazyboys */
        MX6QDL_PAD_EIM_D28__GPIO3_IO28 0x17059
    >;
};

```

```

gpio-keys {
    compatible = "gpio-keys";
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_gpio_keys>;

    power {
        label = "Power Button";
        gpios = <&gpio3 27 GPIO_ACTIVE_LOW>;
        wakeup-source;
        linux,code = <KEY_POWER>;
    };
    back {

```

```

        label = "Back key";
        gpios = <&gpio3 28 GPIO_ACTIVE_LOW>;
        linux,code = <KEY_BACK>;
    };

};

```

확인 방법

```

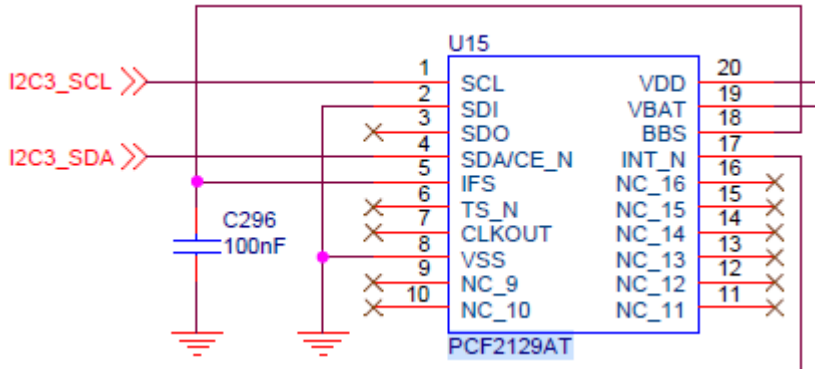
# cat /proc/bus/input/devices

I: Bus=0019 Vendor=0001 Product=0001 Version=0100
N: Name="gpio-keys"
P: Phys=gpio-keys/input0
S: Sysfs=/devices/soc0/gpio-keys/input/input2
U: Uniq=
H: Handlers=kbd event1
B: PROP=0
B: EV=3
B: KEY=40000000 100000 0 0 0

```

4.10.RTC 포팅

PCF2129 RTC칩이 탑재되어 있다. I2C는 3번 버스를 사용한다.



드라이버는 drivers rtc/rtc-pcf2127.c 에 있다.
 arch/arm/boot/dts/imx6qdl-sabresd.dtsi 추가 한다.

```

&i2c3 {
    clock-frequency = <100000>;
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_i2c3>;

```

```
status = "okay";

rtc@51 {
    compatible = "nxp,pcf2129";
    reg = <0x51>;
};
```

커널 Configuration은 아래와 같이 한다.

```
CONFIG_RTC_DRV_PCF2127=y
```

에러 발생 시

```
rtc-pcf2127-i2c 2-0051: hctosys: unable to read the hardware clock
```

아래와 같이 시간을 한번 설정하면 된다.

```
hwclock --systohc -u
```

다시 부팅하면 커널 로그에서 시간이 설정된 것을 볼 수 있다.

```
rtc-pcf2127-i2c 2-0051: setting system clock to 1970-01-01 00:08:12 UTC (492)
```

컴파일 후 확인 한다.

```
$ cd /sys/class/i2c-dev/i2c-2/device/2-0051
# cat modalias
i2c:pcf2129
```

4.11. 이더넷 포팅

Mango-IMX6Q는 이더넷 칩은 AR8035이다.

RGMII 모드로 동작을 한다.

```
CONFIG_AT803X_PHY=y
```

인터럽트 핀은 ENET_RXD1(GPIO1_IO26) , 인터럽트 핀을 사용하지 않고, polling 방식으로 구동을 한다.

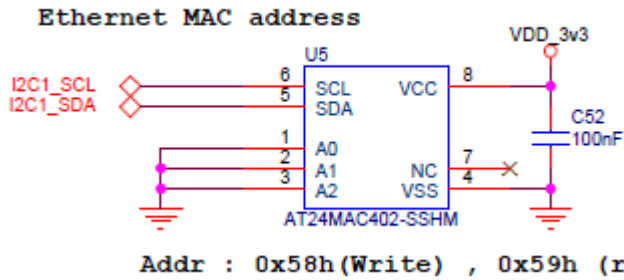
RESET 핀은 ENET_CRSDV(GPIO1_IO25)

```
&fec {
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_enet>;
    phy-mode = "rgmii";
    phy-reset-gpios = <&gpio1 25 GPIO_ACTIVE_LOW>;

    status = "okay";
```

```
};
```

MAC 칩이 있다.



drivers/net/ethernet/freescale/fec_main.c 파일 수정한다.

```
static int mac_i2c_rxdata(struct i2c_client *client, unsigned char addr, unsigned char *data, int length)
{
    int err = -EINVAL;
    unsigned char addr_buff[1];
    addr_buff[0] = addr & 0xff;
    struct i2c_msg msgs[] = {
        {
            .addr = client->addr,
            .flags = 0,
            .len = 1,
            .buf = addr_buff,
        },
        {
            .addr = client->addr,
            .flags = 0x0001, //I2C_M_RD,
            .len = length,
            .buf = data,
        },
    };
};

MANGO_DBG("addr = 0x%x", client->addr);

if (i2c_transfer(client->adapter, msgs, 2) < 0) {
    printk(KERN_ERR "mt9p111_i2c_rxdata failed!\n");
    return -EIO;
}
```

```
    }

    return 0;
}
```

```
    i2c_adap = i2c_get_adapter(0);
    memset(&i2c_info, 0, sizeof(struct i2c_board_info));
    client = i2c_new_device(i2c_adap, &my_dev_info[0]);

#ifdef CONFIG_24AA025E48 //24AA..
    client->addr = 0x50;
    mac_i2c_rxdata(client,0xfa,mac_addr,6);

#else //AT24MAC402 Chip
    client->addr = 0x58;
    mac_i2c_rxdata(client,0x9A,mac_addr,6);

#endif

    printk("[CRZ] Mac Addr: %02X:%02X:%02X:%02X:%02X:%02X\n",
           mac_addr[0], mac_addr[1], mac_addr[2], mac_addr[3], mac_addr[4],
           mac_addr[5]);

    i2c_put_adapter(i2c_adap);

    memcpy(iap, mac_addr, 6);
```

```
static const unsigned short normal_i2c[] = { 0x50, I2C_CLIENT_END };
static struct i2c_board_info my_dev_info[] __initdata = {
    {
        I2C_BOARD_INFO("ksz8081_mac",0x50),
        .platform_data= NULL,
    },
};
```

이더넷에서 dhcp로 ip를 할당 받지 못하는 문제가 발생을 한다.
AR8035에서 나오는 Clock은 125MHz가 나온다.

```
# ethtool eth0
Settings for eth0:
    Supported ports: [ TP MII ]
```



```

Supported link modes:  10baseT/Half 10baseT/Full
                        100baseT/Half 100baseT/Full
                        1000baseT/Full

Supported pause frame use: Symmetric
Supports auto-negotiation: Yes
Advertised link modes: 10baseT/Half 10baseT/Full
                        100baseT/Half 100baseT/Full
                        1000baseT/Full

Advertised pause frame use: Symmetric
Advertised auto-negotiation: Yes
Link partner advertised link modes: 10baseT/Half 10baseT/Full
                                     100baseT/Half 100baseT/Full
                                     1000baseT/Full

Link partner advertised pause frame use: Symmetric
Link partner advertised auto-negotiation: Yes
Speed: 1000Mb/s
Duplex: Full
Port: MII
PHYAD: 0
Transceiver: external
Auto-negotiation: on
Supports Wake-on: g
Wake-on: d
Link detected: yes

```

원인이 무엇인지 찾고 있다. 의심되는 부분은 irq가 POLL로 동작을 한다는 것이다.

```

Atheros 8035 ethernet 2188000.ethernet-1:00: attached PHY driver [Atheros 8035 ethernet]
(mii_bus:phy_addr=2188000.ethernet-1:00, irq=POLL)

```

커널 드라이버에서 configuration에서
CONFIG_AT803X_PHY is not set
로 해야 한다. AT8035 드라이버 문제가 있다.

```

# udhcpc -ieth0
udhcpc: started, v1.26.2
[CRZ]      drivers/net/phy/phy_device.c      (1816)      phy_probe:      phydrv->flags=0x0,
phy_interrupt_is_valid(phydev)=0
[CRZ] arch/arm/mach-imx/mach-imx6q.c (153) ar8035_phy_fixup:
[CRZ] drivers/net/phy/phy_device.c (892) phy_attached_print: phydev->irq=-1
Generic    PHY    2188000.ethernet-1:00:  attached    PHY    driver    [Generic    PHY]

```

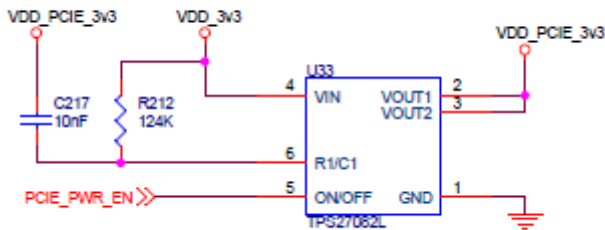
```

(mii_bus:phy_addr=2188000.ethernet-1:00, irq=POLL)
IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
udhcp: sending discover
fec 2188000.ethernet eth0: Link is Up - 100Mbps/Full - flow control rx/tx
IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
udhcp: sending discover
udhcp: sending select for 192.168.100.219
udhcp: lease of 192.168.100.219 obtained, lease time 7200
deleting routers
adding dns 168.126.63.1
adding dns 168.126.63.2

```

4.12. PCIe 드라이버 포팅 및 테스트

PCIE_PWR_EN(EIM_D29,GPIO3_IO29) 핀은 High로 해야 한다.



```

/* delete reg_pcie: regulator@3 {
    compatible = "regulator-fixed";
    reg = <3>;
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_pcie_reg>;
    regulator-name = "MPCIE_3V3";
    regulator-min-microvolt = <3300000>;
    regulator-max-microvolt = <3300000>;
    gpio = <&gpio3 29 0>;
    regulator-always-on;
    enable-active-high;
};
pinctrl_pcie_reg: pciereggrp {
    fsl,pins = <
        MX6QDL_PAD_EIM_D29__GPIO3_IO29    0x1b0b0 /* crazyboys
20200916 */

```

```
        >;
    };
*/
```

PCIE_RST_B(GPIO_17, GPIO7_12) 핀이다.

```
pinctrl_pcie: pciegrp {
    fsl,pins = <
        MX6QDL_PAD_GPIO_17_GPIO7_IO12 0x1b0b0
    >;
};
```

```
&pcie {
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_pcie>;
    power-on-gpio = <&gpio3 29 0>;
    reset-gpio = <&gpio7 12 GPIO_ACTIVE_LOW>;
    status = "okay";
};
```

커널 Configuration을 한다.

```
CONFIG_PCI=y
CONFIG_PCI_DOMAINS=y
CONFIG_PCI_DOMAINS_GENERIC=y
CONFIG_PCI_SYSCALL=y
CONFIG_PCIEPORTBUS=y
CONFIG_HOTPLUG_PCI_PCIE=y
CONFIG_PCIEAER=y
CONFIG_PCI_MSI=y
CONFIG_PCI_MSI_IRQ_DOMAIN=y
CONFIG_PCIE_DW=y
CONFIG_PCI_IMX6=y
```

로그를 보면 아래와 같이 나오면 정상이다.

```
imx6q-pcie 1ffc000.pcie: no reserved region node.
[CRZ] drivers/pci/dwc/pci-imx6.c (2361) imx_pcie_probe:
imx6q-pcie 1ffc000.pcie: 1ffc000.pcie supply epdev_on not found, using dummy regulator
OF: PCI: host bridge /soc/pcie@1ffc000 ranges:
OF: PCI:   IO 0x01f80000..0x01f8ffff -> 0x00000000
OF: PCI:   MEM 0x01000000..0x01efffff -> 0x01000000
```

```

imx6q-pcie 1ffc000.pcie: Link up, Gen1
imx6q-pcie 1ffc000.pcie: PCI host bridge to bus 0000:00
pci_bus 0000:00: root bus resource [bus 00-ff]
pci_bus 0000:00: root bus resource [io 0x0000-0xffff]
pci_bus 0000:00: root bus resource [mem 0x01000000-0x01efffff]
PCI: bus0: Fast back to back transfers disabled
PCI: bus1: Fast back to back transfers disabled
pci 0000:00:00.0: BAR 8: assigned [mem 0x01000000-0x015fffff]
pci 0000:00:00.0: BAR 0: assigned [mem 0x01600000-0x016fffff]
pci 0000:00:00.0: BAR 6: assigned [mem 0x01700000-0x0170ffff pref]
pci 0000:01:00.0: BAR 2: assigned [mem 0x01000000-0x013fffff 64bit]
pci 0000:01:00.0: BAR 0: assigned [mem 0x01400000-0x01407fff 64bit]
pci 0000:00:00.0: PCI bridge to [bus 01-ff]
pci 0000:00:00.0: bridge window [mem 0x01000000-0x015fffff]
pcieport 0000:00:00.0: Signaling PME with IRQ 308
pcieport 0000:00:00.0: AER enabled with IRQ 308

```

아래 명령으로 확인 가능하다.

```

# lspci -k
00:00.0 Class 0604: 16c3:abcd pcieport
01:00.0 Class 0280: 14e4:4355

```

4.13. 오디오 코덱 WM8960 포팅

Wm8960 코덱을 사용한다.

arch/arm/boot/dts/imx6qdl-sabresd.dtsi 추가

```

sound {
    compatible = "fsl,imx6q-sabresd-wm8960",
                "fsl,imx-audio-wm8960";
    model = "wm8960-audio";
    cpu-dai = <&ssi2>;
    audio-codec = <&codec>;
    asrc-controller = <&asrc>;
    hp-det = <3 0>;
    codec-master;
    gpr = <&gpr>;
    audio-routing =
        "Headphone Jack", "HP_L",
        "Headphone Jack", "HP_R",

```

```

        "Ext Spk", "SPK_LP",
        "Ext Spk", "SPK_LN",
        "Ext Spk", "SPK_RP",
        "Ext Spk", "SPK_RN",
        "Main MIC", "MICB",
        "CPU-Playback", "ASRC-Playback",
        "Playback", "CPU-Playback",
        "ASRC-Capture", "CPU-Capture",
        "CPU-Capture", "Capture";
    mux-int-port = <2>;
    mux-ext-port = <3>;
    //hp-det-gpios = <&gpio1 9 1>;
    //mic-det-gpios = <&gpio1 9 1>;
    status="okay";
};
sound-hdmi {
    compatible = "fsl,imx6q-audio-hdmi",
                "fsl,imx-audio-hdmi";
    model = "imx-audio-hdmi";
    hdmi-controller = <&hdmi_audio>;
};

```

I2C는 1번 채널을 사용한다.

Slave address는 0x1a이다.

```

codec: wm8960@1a {
    compatible = "wlf,wm8960";
    reg = <0x1a>;
    clocks = <&clks IMX6QDL_CLK_CKO>;
    clock-names = "mclk";
    clock-frequency= <12000000>;
    wlf,shared-lrclk;
};

```

I2S slave를 사용한다.

```

&ssi2 {
    fsl,mode = "i2s-slave";//crazyboys 20200916
    status = "okay";
};

```

소스는

```
sound/soc/fsl/imx-wm8960.c
```

클럭 부분 수정을 한다.

```
drivers/clk/imx/clk-imx6q.c
```

```
imx_clk_set_parent(clk[IMX6QDL_CLK_CKO2_SEL], clk[IMX6QDL_CLK_OSC]);
    //crazyboys 20200916 imx_clk_set_parent(clk[IMX6QDL_CLK_CKO], clk[IMX6QDL_CLK_CKO2]);
#if 1//crazyboys 20200916
    imx_clk_set_parent(clk[IMX6QDL_CLK_CKO], clk[IMX6QDL_CLK_OSC]);
    clk_set_rate(clk[IMX6QDL_CLK_CKO],24000000);//crazyboys 20190718 24MHz audio
    clk_set_rate(clk[IMX6QDL_CLK_CKO2],12000000);//crazyboys 20190621 12MHz hdmi
#endif
```

4.14. SATA 포팅

arch/arm/boot/dts/imx6q-sabresd.dts 파일에 추가 한다.

```
&sata {
    status = "okay";
};
```

커널 Configuration은 아래와 같이 하면 된다.

```
CONFIG_SATA_AHCI=y
CONFIG_SATA_AHCI_PLATFORM=y
CONFIG_AHCI_IMX=y
```

5. Android 포팅

5.1. 불필요한 디바이스 제거

device/fsl/imx6dq/sabresd_6dq/BoardConfig.mk 파일 수정

```
BOARD_HAS_SENSOR := false
SENSOR_MMA8451 := false
BOARD_HAVE_BLUETOOTH_BCM := false
```

5.2. Bootargs 설정하기

device/fsl/imx6dq/sabresd_6dq/BoardConfig.mk 파일에서 설정을 한다.

10.1인치 LCD 설정을 아래와 같이 한다.

```
BOARD_KERNEL_CMDLINE := console=ttymx0,115200 init=/init video=mxafb0:dev=lcd,MANGO-CAP10.1,if=BGR24,bpp=32 video=mxafb1:dev=hdmi,1920x1080M@60,if=BGR24,bpp=24 video=mxafb2:off video=mxafb3:off vmlloc=256M androidboot.console=ttymx0 consoleblank=0 androidboot.hardware=freescale cma=320M galcore.contiguousSize=33554432 loop.max_part=7
```

5.3. 정전식 터치 동작 시키기

Android 정전식 터치를 동작하게 하기 위해서는 디바이스 드라이버 이름과 같은 idc가 있어야 한다.

```
# cat /proc/bus/input/devices
I: Bus=0018 Vendor=0000 Product=0000 Version=0000
N: Name="mango-ts"
P: Phys=
S: Sysfs=/devices/virtual/input/input1
U: Uniq=
H: Handlers=kbd event0
B: PROP=0
B: EV=b
B: KEY=400 0 4 0 0 0 0 c0000 0 0 0
B: ABS=2650000 1000000
```

"mango-ts.idc"가 vendor/usr/idc/mango-ts.idc 디렉토리에 존재를 해야 한다.

device/fsl/imx6dq/ProductConfigCommon.mk 파일에서 추가를 한다.

```
device/fsl/common/input/eGalax_Touch_Screen.idc:$(TARGET_COPY_OUT_VENDOR)/usr/idc/mango-ts.idc ₩
```

5.4. 에러 디버깅

5.4.1. 파일 시스템 Mount 못하는 문제

```
init: init first stage started!
```

```
init: Using Android DT directory /proc/device-tree/firmware/android/
init: [libfs_mgr]fs_mgr_read_fstab_default(): failed to find device default fstab
init: bool android::init::FirstStageMount::InitRequiredDevices(): partition(s) not found in /sys, waiting for
their uevent(s): boot, system, vbmeta, vendor
init: Wait for partitions returned after 10010ms
init: bool android::init::FirstStageMount::InitRequiredDevices(): partition(s) not found after polling
timeout: boot, system, vbmeta, vendor
init: Failed to mount required partitions early ...
init: Reboot start, reason: reboot, rebootTarget: bootloader
init: Reboot ending, jumping to kernel
imx-sdma 20ec000.sdma: external firmware not found, using ROM firmware
ci_hdrc ci_hdrc.1: remove, state 4
```

위의 메시지에서 가장 중요한 메시지는 imx-sdma 20ec000.sdma: external firmware not found, using ROM firmware

이다.

커널 메시지를 보면 아래와 같이 출력이 된다.

```
mmc2: SDHCI controller on 2198000.usdhc [2198000.usdhc] : eMMC
```

```
mmc3: SDHCI controller on 219c000.usdhc [219c000.usdhc] : SD4
```

```
mmc1: SDHCI controller on 2194000.usdhc [2194000.usdhc] using ADMA
```

```
mmc2: SDHCI controller on 2198000.usdhc [2198000.usdhc] using ADMA
```

```
sdhci-esdhc-imx 219c000.usdhc: Got CD GPIO
```

```
mmc3: SDHCI controller on 219c000.usdhc [219c000.usdhc] using ADMA
```

```
mmc2: new DDR MMC card at address 0001
```

```
mxc_vpu 2040000.vpu_fsl: VPU initialized
```

```
mxc_hdmi_cec soc:hdmi_cec@00120000: HDMI CEC initialized
```

```
mmcblk2: mmc2:0001 8GND3R 7.28 GiB
```

```
Galcore version 6.2.4.190076
```

```
mmcblk2boot0: mmc2:0001 8GND3R partition 1 4.00 MiB
```

```
mmcblk2boot1: mmc2:0001 8GND3R partition 2 4.00 MiB
```

```
mmcblk2rpm: mmc2:0001 8GND3R partition 3 512 KiB
```

```
mmcblk2: p1
```

```
mmc3: host does not support reading read-only switch, assuming write-enable
```

```
mmc3: new high speed SDHC card at address aaaa
```

```
mmcblk3: mmc3:aaaa SC16G 14.8 GiB
```

```
random: fast init done
```

```
mmcblk3: p1 p2
```


안드로이드 소스를 보면 마운트하는 것을 아래 파일이다.

device/fsl/imx6dq/sabresd_6dq/fstab.freescale 파일 수정

```
/devices/soc0/soc/2100000.aips-bus/219c000.usdhc/mmc_host*    auto    auto    defaults
voldmanaged=sdcard:auto,encryptable=userdata
```

안드로이드가 부팅 되면서 아래 코드를 수행한다.

```
init.rc:    mount_all /vendor/etc/fstab.freescale
```

커널도 수정을 해야 한다.

arch/arm/boot/dts/imx6qdl-sabresd.dtsi 파일 수정

```
android {
    compatible = "android,firmware";
    fstab {
        compatible = "android,fstab";
        system {
            compatible = "android,system";
            /* sd card node which used if androidboot.storage_type=sd */
dev_sd = "/dev/block/platform/soc0/soc/2100000.aips-bus/219c000.usdhc/by-name/system";
            /* emmc node which used if androidboot.storage_type=emmc */
dev_emmc = "/dev/block/platform/soc0/soc/2100000.aips-bus/2198000.usdhc/by-name/system";
            type = "ext4";
            mnt_flags = "ro,barrier=1,inode_readahead_blks=8";
            fsmgr_flags = "wait,avb";
        };
        vendor {
            compatible = "android,vendor";
            /* sd card node which used if androidboot.storage_type=sd */
dev_sd = "/dev/block/platform/soc0/soc/2100000.aips-bus/219c000.usdhc/by-name/vendor";
            /* emmc node which used if androidboot.storage_type=emmc */
dev_emmc = "/dev/block/platform/soc0/soc/2100000.aips-bus/2198000.usdhc/by-name/vendor";
            type = "ext4";
            mnt_flags = "ro,barrier=1,inode_readahead_blks=8";
            fsmgr_flags = "wait,avb";
        };
    };
};
```

androidboot.storage_type 타입은 u-boot에서 parameter로 넘겨준다.

6. Android 이미지 Write하기

6.1. 이미지 툴 다운로드 해 보자.

우선 안드로이드 툴이 필요하다.

https://www.nxp.com/webapp/sps/download/license.jsp?colCode=P9.0.0_1.0.0_GA_TOOL&appType=file1&DOWNLOAD_ID=null

android_p9.0.0_1.0.0-ga_tools.tar.gz 파일을 다운로드 받는다.

압축을 해제하면 어떻게 사용하는지 확인 해 보면, 리눅스 PC에서 사용해야하는 shell 형태로 되어 있다. 좀 더 살펴 보자. 이전에는 mfgtools를 사용했는데 변경 되었다.

6.1.1. 스크립트로 이용하여 이미지 Write하기

안드로이드 빌드 완료 후 Micro SD 카드를 Linux PC에 삽입한다.

```
$ dmesg | tail
[172161.337806] scsi host4: usb-storage 1-9:1.0
[172162.361020] scsi 4:0:0:0: Direct-Access    Generic  STORAGE DEVICE   9451 PQ: 0 ANSI: 0
[172162.361881] sd 4:0:0:0: Attached scsi generic sg3 type 0
[172162.576992] sd 4:0:0:0: [sd] 31422464 512-byte logical blocks: (16.1 GB/15.0 GiB)
[172162.577895] sd 4:0:0:0: [sd] Write Protect is off
[172162.577897] sd 4:0:0:0: [sd] Mode Sense: 03 00 00 00
[172162.578671] sd 4:0:0:0: [sd] No Caching mode page found
[172162.578675] sd 4:0:0:0: [sd] Assuming drive cache: write through
[172162.593785]  sdc: sdc1 sdc2 sdc3 sdc4 sdc5 sdc6 sdc7 sdc8 sdc9 sdc10 sdc11 sdc12 sdc13
[172162.597222] sd 4:0:0:0: [sd] Attached SCSI removable disk
```

디바이스 노드 확인한다. 디바이스 노드를 확인하지 않고, 아래 스크립트를 실행 시 Write가 되지 않는 경우도 발생하지만, 더 큰 문제는 운영체제가 삭제될 수 있다. 주의 바란다.

```
$ cd ${ANDROID_BUILD}
$ sudo ./device/fsl/common/tools/fsl-sdcard-partition.sh -f imx6q -D out/target/product/sabresd_6dq/
/dev/sdx
```

8GB micro sd 이미지가 Write가 된다.

16GB Micro sd

```
$ cd ${ANDROID_BUILD}
$ sudo ./device/fsl/common/tools/fsl-sdcard-partition.sh -f imx6q -c 14 -D out/target/product/sabresd_6dq/
```

```
/dev/sdx
```

32GB Micro Sd

```
$ cd ${ANDROID_BUILD}
```

```
$ sudo ./device/fsl/common/tools/fsl-sdcard-partition.sh -f imx6q -c 28 -D out/target/product/sabresd_6dq/  
/dev/sdx
```

userdata영역을 다시 포맷한다.

```
$ sudo mkfs.ext4 /dev/sdc11
```

아래와 같이 에러가 나오면

```
Missing simg2img app. Please run: sudo apt-get install android-tools-fsutils
```

sudo apt-get install android-tools-fsutils 실행하면 된다.

이미지 Write가 완료되면 아래와 같이 메시지가 출력된다.

```
>>>>>>>>>>>>>>> Flashing successfully completed <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
```