

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 환경4
- 2. 문서 및 소스 다운로드.....4
 - 2.1. 안드로이드 Pie 9.0, 커널,u-boot소스 다운로드 하기..... 5
 - 2.2. U-boot, kernel, 안드로이드 컴파일 하기..... 6
 - 2.2.1. Android 컴파일.....6
 - 2.3. 이미지 툴 다운로드 해 보자.....21
 - 2.3.2. 소소제목.....21

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"> • imx-p9.0.0_2.2.0-ga.tar.gz: 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 android_p9.0.0_2.2.0-ga_docs.zip:</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"> • android_p9.0.0_2.2.0-ga_image_6qsabresd.tar.gz: Prebuilt images with NXP extended features for the SABRE-SD board. The extended features include more multimedia format support. • android_p9.0.0_2.2.0-ga_image_6qsabreauto.tar.gz: Prebuilt images with NXP extended features for the SABRE-AI board. The extended features include more multimedia format support. • android_p9.0.0_2.2.0-ga_image_6sxsabresd.tar.gz: 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.gz를 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/uboot-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.

2.3. U-boot 수정

u-boot를 Mango-IMX6Q에 맞게 수정해야 한다.

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

2.3.4. 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\freescale\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

2.3.5. 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=0xfffffff REV_ID=0xfffffff

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.ipstime.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.

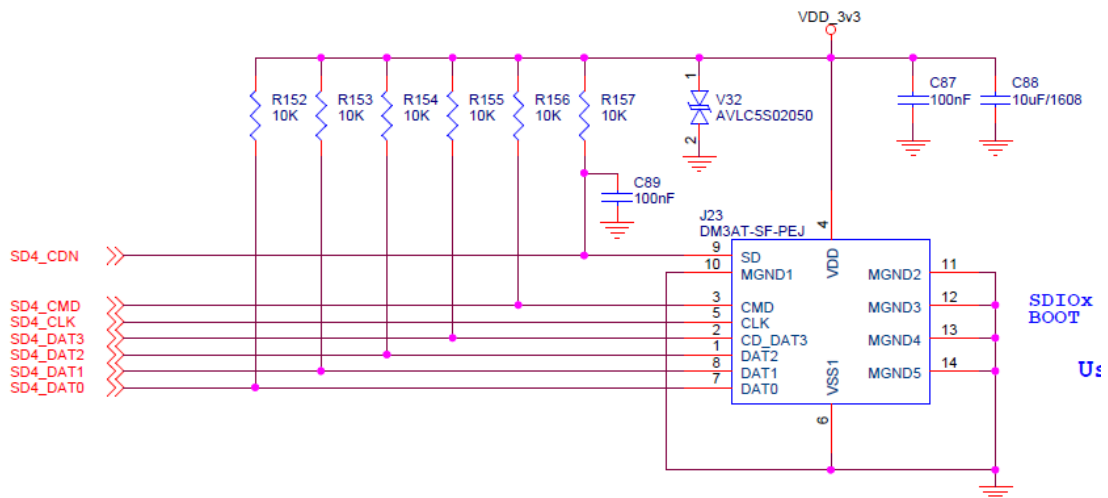
Loading Environment from MMC... *** Warning - bad CRC, using default environment

Failed (-5)

에러가 나오는 것은 SD가 제대로 detec가 되지 않아서 이다.

2.3.6. 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)),
    #endif
};
```

```

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

        #if 0 //crazyboys 20200907
        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>;
    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

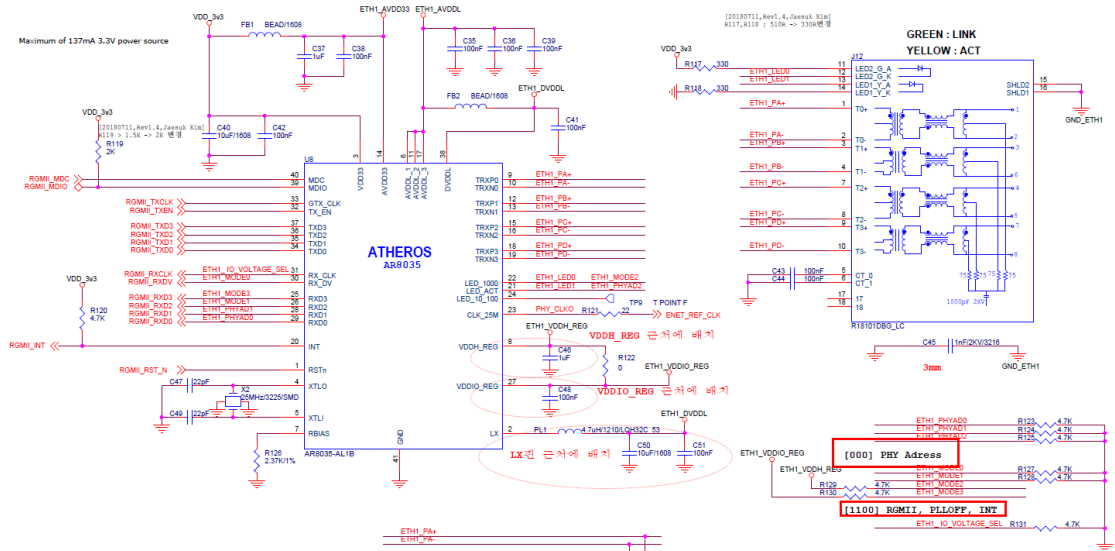
```

Erase Group Size: 512 Bytes

확인 할 수 있다.

2.3.7. 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)
```

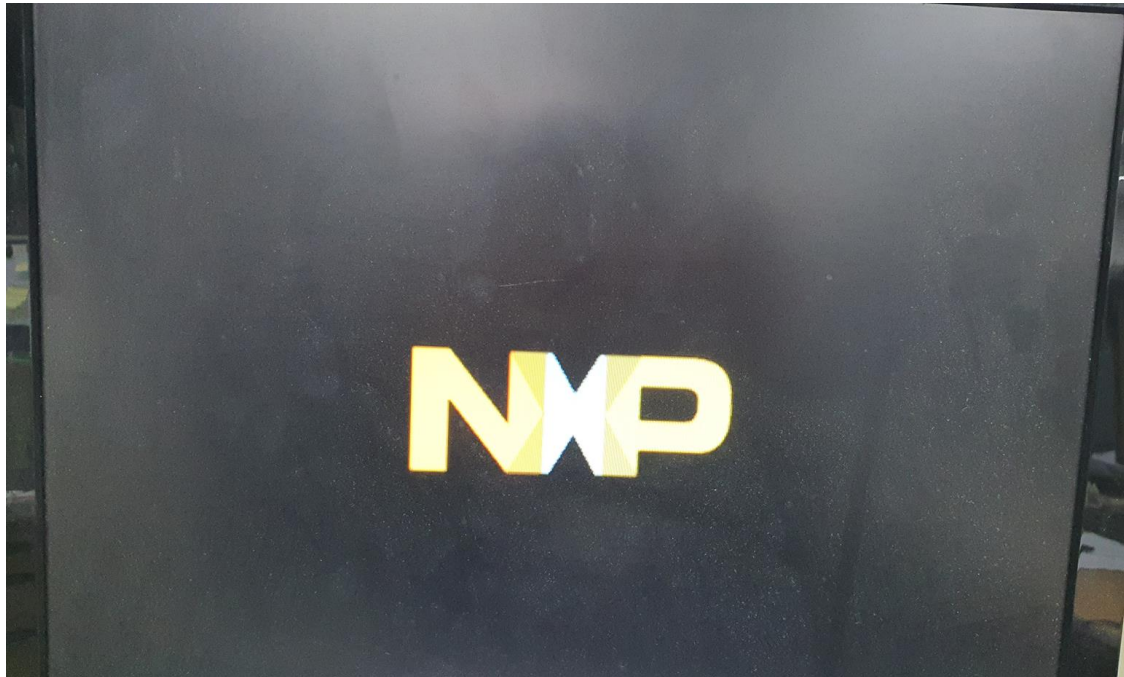
```
*** Warning: no boot file name; using 'C0A864D7.img'  
Using FEC device  
TFTP from server 0.0.0.0; our IP address is 192.168.100.215; sending through gateway 192.168.100.1  
Filename 'C0A864D7.img'.  
Load address: 0xfec  
Loading: *  
TFTP error: 'File not found' (1)  
Not retrying...
```

이더넷이 ip를 할당 받으면 된다.

2.3.8. 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.



2.3.9. 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(DISP0_PWR_EN, "Display Power Enable");
    gpio_direction_output(DISP0_PWR_EN, 1);
    #if 1 //crazyboys 20200907
        gpio_request(DISP0_BACKLIGHT_EN, "Display Backlight Enable");
        gpio_direction_output(DISP0_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 로고가 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)
mxafb: 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
```

2.3.10. 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

2.3.11. 이더넷 포팅 하기

2.3.12. 이더넷 포팅 하기



2.4. 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.

2.4.13.소소제목

