Mango-IMX6Q GPIO 8 개 제어하기
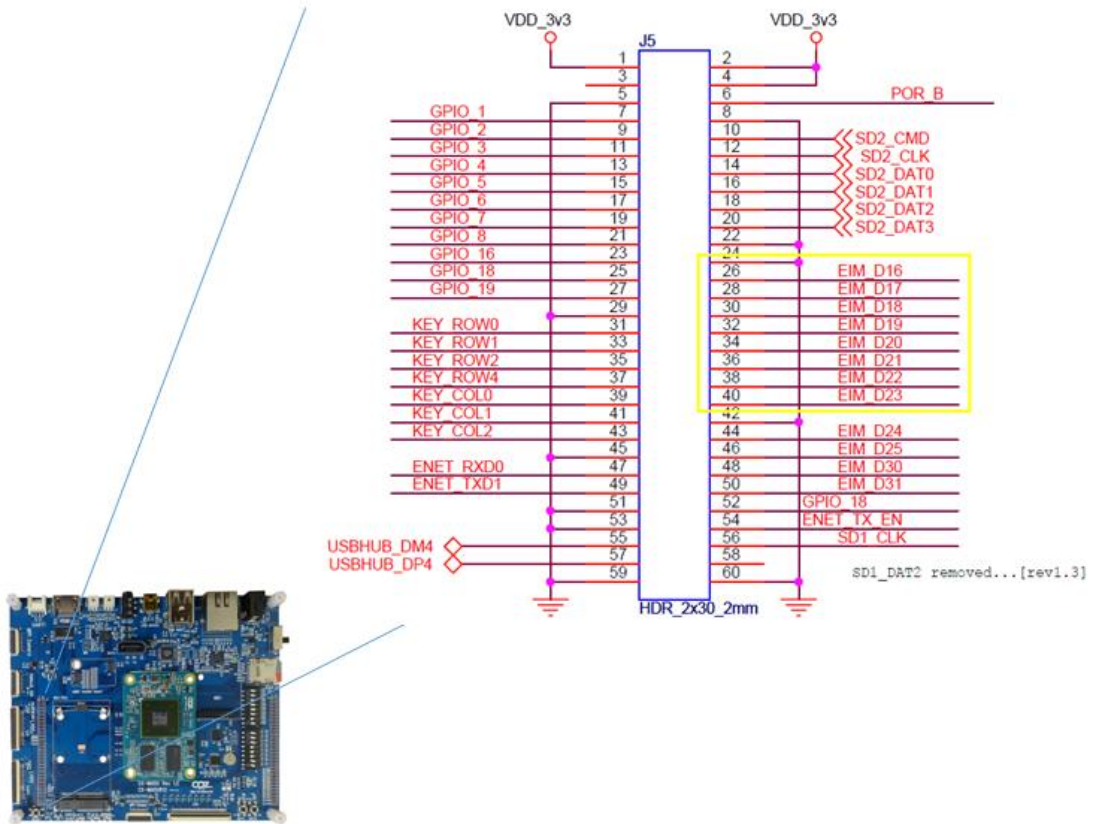
Document History

Revision	Date	Change note
Init	2017-01-25	전종인

1. 핀맵

J5 커넥터에서 EIM_D16 ~ EIM_D23 까지 8개 할당함.

2. 커널 수정

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

```
&iomuxc {
        pinctrl-names = "default";
        pinctrl-0 = <&pinctrl_hog>;

        imx6qdl-sabresd {
                pinctrl_hog: hoggrp {
```

```
                    fsl,pins = <
                        MX6QDL_PAD_NANDF_D0__GPIO2_IO00 0x80000000
                        MX6QDL_PAD_NANDF_D1__GPIO2_IO01 0x80000000
                        MX6QDL_PAD_NANDF_D2__GPIO2_IO02 0x80000000
                        MX6QDL_PAD_NANDF_D3__GPIO2_IO03 0x80000000
                        MX6QDL_PAD_GPIO_0__CCM_CLKO1      0x130b0
                        MX6QDL_PAD_ENET_TXD1__GPIO1_IO29 0x80000000
                        MX6QDL_PAD_ENET_CRS_DV__GPIO1_IO25 0x80000000
                        MX6QDL_PAD_EIM_CS1__GPIO2_IO24 0x80000000
                        MX6QDL_PAD_ENET_RXD0__GPIO1_IO27 0x80000000
                        MX6QDL_PAD_EIM_A25__GPIO5_IO02 0x80000000
                        MX6QDL_PAD_EIM_D16__GPIO3_IO16 0x80000000
                        MX6QDL_PAD_EIM_D17__GPIO3_IO17 0x80000000
                        MX6QDL_PAD_EIM_D18__GPIO3_IO18 0x80000000
                        MX6QDL_PAD_EIM_D19__GPIO3_IO19 0x80000000
                        MX6QDL_PAD_EIM_D20__GPIO3_IO20 0x80000000
                        MX6QDL_PAD_EIM_D21__GPIO3_IO21 0x80000000
                        MX6QDL_PAD_EIM_D22__GPIO3_IO22  0x80000000
                        MX6QDL_PAD_EIM_D23__GPIO3_IO23  0x80000000
```

3. Android 수정

device/fsl/sabresd_6dq/mango_gpio.sh 파일 추가

```
#!/system/bin/sh

#<gpio3_16,EIM_D16>
echo 80 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio80/direction
chmod 777 /sys/class/gpio/gpio80/direction
chmod 777 /sys/class/gpio/gpio80/value
#<gpio3_17,EIM_D17>
echo 81 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio81/direction
chmod 777 /sys/class/gpio/gpio81/direction
chmod 777 /sys/class/gpio/gpio81/value
```

```
#<gpio3_18,EIM_D18>
echo 82 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio82/direction
chmod 777 /sys/class/gpio/gpio82/direction
chmod 777 /sys/class/gpio/gpio82/value
#<gpio3_19,EIM_D19>
echo 83 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio83/direction
chmod 777 /sys/class/gpio/gpio83/direction
chmod 777 /sys/class/gpio/gpio83/value

#<gpio3_20,EIM_D20>
echo 84 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio84/direction
chmod 777 /sys/class/gpio/gpio84/direction
chmod 777 /sys/class/gpio/gpio84/value
#<gpio3_21,EIM_D21>
echo 85 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio85/direction
chmod 777 /sys/class/gpio/gpio85/direction
chmod 777 /sys/class/gpio/gpio85/value

#<gpio3_22,EIM_D22>
echo 86 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio86/direction
chmod 777 /sys/class/gpio/gpio86/direction
chmod 777 /sys/class/gpio/gpio86/value
#<gpio3_23,EIM_D23>
echo 87 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio87/direction
chmod 777 /sys/class/gpio/gpio87/direction
chmod 777 /sys/class/gpio/gpio87/value
```

서비스 등록
"device/fsl/sabresd_6dq/init.rc"

```
service mango_gpio /system/bin/mango_gpio.sh
    class main
```

```
    user root
    group root
    oneshot
```

파일 시스템에 포함

"device/fsl/imx6/imx6.mk"

```
#mango gpio CRZ_icanjji crazyboys 20170125
PRODUCT_COPY_FILES +=    ₩
        device/fsl/sabresd_6dq/mango_gpio.sh:system/bin/mango_gpio.sh ₩
```

4. Android 어플 예제 첨부

mango_standard.h파일에서 gpio번호 할당

```
#define GPIO_1 80
#define GPIO_2 81
#define GPIO_3 82
#define GPIO_4 83
#define GPIO_5 84
#define GPIO_6 85
#define GPIO_7 86
#define GPIO_8 87
```

명령으로 GPIO 번호 알아 본다.

EIM_D16(GPIO3_16) ~ EIM_D23(GPIO3_23) 임

cat /sys/kernel/debug/gpio 명령 실행 결과

```
GPIOs 0-31, platform/209c000.gpio, 209c000.gpio:
 gpio-4    (?                       ) out lo
 gRTL871X: cfg80211_rtw_scan(wlan0)
pio-5    (?                     ) out lo
 gpio-17   (?            RTL871X: SetHwReg8188E:(HW_VAR_CHECK_TXBUF)TXBUF Empty(1) in 0 ms
        ) out lo
 gpio-18       RTL871X:    SetHwReg8188E(wlan0):    [HW_VAR_MACID_SLEEP]    macid=0,    org
reg_0x48c=0x00000000
(?                     ) out lo
```

```
gpio-19   (?                    ) out lo
gpio-25   (phy-reset        ) out lo
gpio-30   (?                    ) out lo

GPIOs 32-63, platform/20a0000.gpio, 20a0000.gpio:
gpio-62   (?                    ) out lo
gpio-63   (?                    ) out lo

GPIOs 64-95, platform/20a4000.gpio, 20a4000.gpio:
 gpio-91   (Power Button      ) in   hi
 gpio-92   (Volume Up          ) in   hi

GPIOs 96-127, platform/20a8000.gpio, 20a8000.gpio:

GPIOs 128-159, platform/20ac000.gpio, 20ac000.gpio:

GPIOs 160-191, platform/20b0000.gpio, 20b0000.gpio:
 gpio-171 (?                    ) out lo

GPIOs 192-223, platform/20b4000.gpio, 20b4000.gpio:
```

mango_standard.c에 아래와 같이 추가합니다.

```
JNIEXPORT void JNICALL Java_android_gpio_sample_GpioActivity_gpioexport
  (JNIEnv *env, jobject thiz)
{

        gpio_export(GPIO_1);
        gpio_dir_out(GPIO_1);
        LOGD("gpioexport 1");

        gpio_export(GPIO_2);
        gpio_dir_out(GPIO_2);
        LOGD("gpioexport 2");

        gpio_export(GPIO_3);
        gpio_dir_out(GPIO_3);
        LOGD("gpioexport 3");
```

```
        gpio_export(GPIO_4);
        gpio_dir_out(GPIO_4);
        LOGD("gpioexport 4");

        gpio_export(GPIO_5);
        gpio_dir_out(GPIO_5);
        LOGD("gpioexport 5");

        gpio_export(GPIO_6);
        gpio_dir_out(GPIO_6);
        LOGD("gpioexport 6");

        gpio_export(GPIO_7);
        gpio_dir_out(GPIO_7);
        LOGD("gpioexport 7");

        gpio_export(GPIO_8);
        gpio_dir_out(GPIO_8);
        LOGD("gpioexport 8");



}

JNIEXPORT void JNICALL Java_android_gpio_sample_GpioActivity_gpiorelay
   (JNIEnv *env, jobject thiz,jint out)
{
        gpio_set_value(GPIO_1,out);
        LOGD("GPIO_1 %d ",out);
```